

## **Кафедра комп'ютерної інженерії та електроніки**

Назва дисципліни – Комп'ютерна схемотехніка

Викладач: проф. Когут Ігор Тимофійович

1. В.В. Ракитин. Интегральные схемы на КМОП-транзисторах. Москва, 2007. – 308 с. Частина 2.
2. Chorng-Wei LIAW, Leaf YEH, Ming-Jang LIN Trench Termination Design and Analysis in Low-Voltage N-Channel Trench Power Metal–Oxide–Semiconductor Field-Effect Transistor, and Chrong Jung LIN Japanese Journal of Applied Physics Vol. 47, No. 3, 2008, pp. 1507–1511
3. Компютерна схемотехніка та схемотехніка ЕОМ Конспект лекцій Київського національного авіаційного університету <http://sh.instone.com.ua/teor.php>.
4. Daniel Mlynek Design of VLSI systems. 1998. <http://emicroelectronics.free.fr/onlineCourses/VLSI/index.html> Розділ 6.
5. An Illustration of 0.1μm CMOS layout design on PC [www.microwind.net](http://www.microwind.net).

## Часть 2. Цифровые КМОП ИС

### Глава 1. Введение в проектирование цифровых КМОП ИС

В цифровых схемах используется двоичное кодирование информации – сигналы имеют два значения: логической единицы и логического нуля. В КМОП-схемотехнике за единицу принимают высокий уровень напряжения  $V_H$ , близкий к уровню напряжения питания  $V_{DD}$ , за ноль – низкий уровень напряжения  $V_L$  близкий к нулю.

Цифровые логические КМОП-элементы выполняют логические функции над входными двоичными сигналами и должны формировать на выходе двоичные сигналы.

Выполняемая логическая функция описывается таблицей истинности, которая каждой комбинации входных сигналов ставит в соответствие определенный выходной сигнал.

#### 1.1. Переключательная модель МОП-транзистора

Во многих применениях МОП-транзисторы могут рассматриваться как ключи, управляемые входными напряжениями на затворе  $G$ , рис.1.1,а.

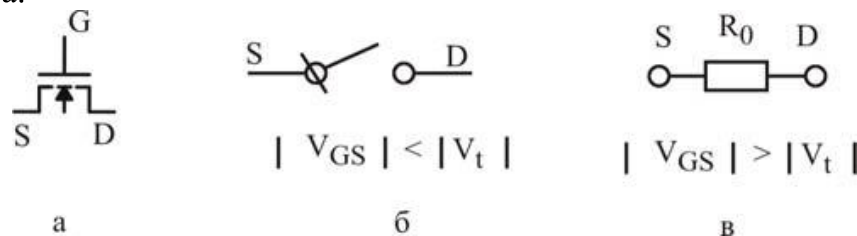


Рис.1.1. МОП-транзистор: а – условное обозначение; б – модель закрытого МОП; в – модель открытого МОП

В отсутствие входного сигнала, ключ разомкнут (рис.1.1,б). При наличии входного отпирающего напряжения подвижные носители движутся от истока  $S$  к стоку  $D$ , ключ замкнут (рис.1.1в), и величина протекающего тока определяется выходным сопротивлением прибора  $R_0$ .

В n-МОП носителями являются электроны и они движутся от истока к стоку, если сток находится под более положительным потенциалом, чем исток. Для открывания (включения) n-МОП напряжение между затвором и истоком  $V_{GS}$  должно быть больше порогового  $V_t$ .



В р-МОП носителями являются дырки, а напряжение на стоке более отрицательно, чем на истоке. Для включения р-МОП напряжение между затвором и истоком  $V_{GS}$  должно быть меньше порогового  $V_t$ .

Иногда напряжение между выходными электродами МОП меняет знак, в этих случаях можно или не различать электроды (они обычно конструктивно одинаковы, а сам МОП симметричен), или считать, что сток и исток меняются местами.

## 1.2. КМОП-ключи

Простейшая цифровая схема – это двоичный ключ, обеспечивающий управляемую передачу двоичного сигнала

Поскольку МОП-транзисторы могут быть в открытом и закрытом состоянии, то на их основе можно построить и двоичные ключи. Однако и п-МОП и р-МОП по отдельности не являются идеальными ключами по отношению к двоичным сигналам.

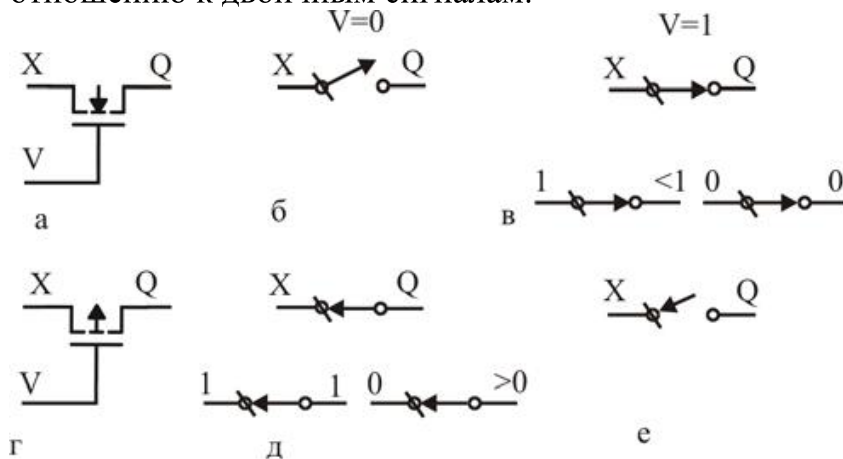


Рис.1.2. Простые ключи на МОП и условия их включения: а – п-МОП ключ; б – п-МОП выключен; в – п-МОП включен; г – р-МОП ключ; д – р-МОП включен, е – р-МОП выключен.

В п-МОП-транзисторе (рис.1.2,а) исток (электрод с меньшим потенциалом) электрически соединен со стоком, если напряжение на затворе больше напряжения на истоке на величину порогового напряжения, и разомкнут в противном случае. Возможны три ситуации:

1) если на входном затворе нулевое напряжение, то ключ между источником сигнала X и выходным узлом Q, разомкнут (рис.1.2,б) и в узле Q сохранится напряжение, которое было перед размыканием ключа;

2) если на входном затворе единичное напряжение (рис.1.2,в), а источник сигнала X поддерживает нулевое напряжение (электрод X явля-

ется истоком), то выход Q соединен со входом X. При емкостной нагрузке в точке Q на ней установится нулевой потенциал;

3) если на входном затворе единичное напряжение, а источник сигнала X поддерживает единичное напряжение (электрод X является стоком), то напряжение на выходе Q может от нулевого вырасти только до  $V_{DD} - V_t$ . Если оно уже было единичным, то оно таким и останется.

В рассмотренных ситуациях предполагается наличие источника сигнала с малым выходным сопротивлением и приемника сигнала с большим входным сопротивлением. Например, когда источник X – сток открытого МОП, а приемник Q – затвор МОП-транзистора.

Таким образом, n-МОП хорошо передает нулевое напряжение, а передача единичного напряжения происходит не полностью. Аналогично, p-МОП (рис.1.2,г) при нулевом напряжении на затворе хорошо передает единичное напряжение, а при передаче нулевого напряжения выходное напряжение может снизиться не до нулевого, а только до величины  $V_t$  (рис.1.2,д). При единичном входном напряжении p-МОП разомкнут (рис.1.2,е).

Для устранения указанных недостатков применяют КМОП-ключи, содержащие параллельно соединенные n-МОП и p-МОП (рис.1.3). Для управления такими ключами используют парафазное управление: на вход n-МОП подается управляющий сигнал V, а на вход p-МОП инвертированный –  $\bar{V}$  (рис.1.3,а).

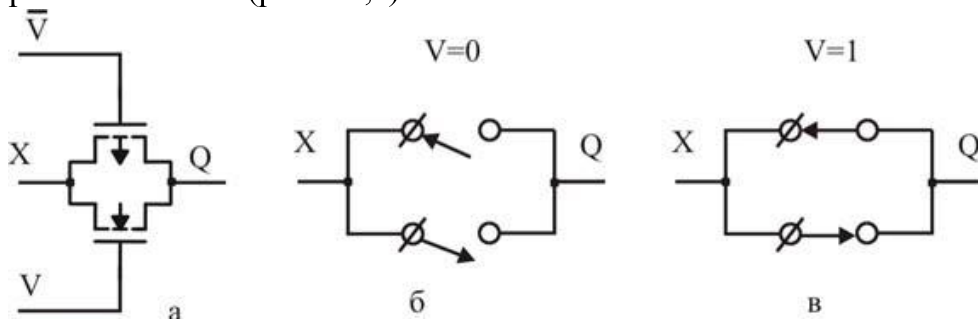


Рис.1.3. КМОП-ключ: а – схема; б – ключ выключен; в – ключ включен.

При  $V = 0$  ключ разомкнут (рис.1.3,б), оба транзистора закрыты, при  $V=1$ , оба ключа замкнуты (рис.1.3,в), причем при  $X=0$  выход Q замкнут на вход через n-МОП, а при  $X=1$  выход Q замкнут на вход через p-МОП.

### 1.3. КМОП-инверторы.

КМОП-ключи являются пассивными элементами, поскольку они не обеспечивают усиления входного (коммутируемого) сигнала. Для уси-

ления двоичных сигналов широко применяется другая простая двухтранзисторная схема – симметричный инвертор, в котором n-МОП и p-МОП включены последовательно между электродами напряжения питания (рис.1.4). В этой схеме n-МОП подсоединен истоком к нулевому потенциалу, а p-МОП истоком к положительному источнику питания. КМОП-инвертор можно рассматривать как два ключа с одним управляющим сигналом X, один из которых (n-МОП) соединяет выход Q с низким уровнем VL, а другой p-МОП – выход Q с высоким уровнем VH.

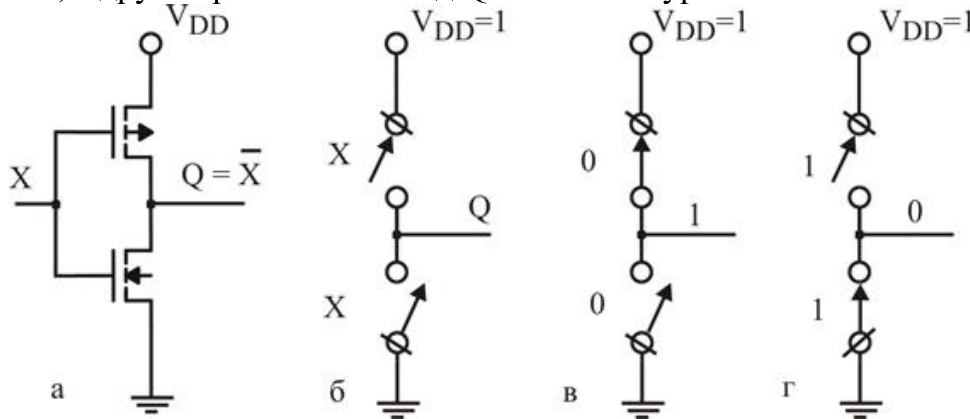


Рис.1.4. КМОП-инвертор: а – схема; б – ключевая модель инвертора; в – положение ключей при нулевом выходе; г – положение ключей при единичном выходе

При нулевом входном сигнале n-МОП закрыт, а p-МОП открыт (рис 1.4,в). На выход через открытый p-МОП поступает напряжение питания, т.е. на выходе единица.

При единичном входном сигнале ситуация прямо противоположная, p-МОП закрыт, а n-МОП открыт (рис1.3,г) и привязывает выходное напряжение к нулевому потенциалу, т.е. на выходе ноль.

В обоих случаях у КМОП-инвертора открыт только один МОП и цепь питания разомкнута. Таким образом КМОП-инверторы не потребляют ток (с точностью до малых токов утечки) в статическом состоянии, что очень важно для больших ИС.

**Квазиинверторы.** Наряду с рассмотренной схемой в качестве инверторов используют и другие схемы. Так в двухтранзисторной КМОП-схеме входной сигнал подается на затвор n-МОП, а затвор p-МОП подсоединен к нулевому потенциалу (рис.1.5,а). Нагрузочный p-МОП постоянно находится в открытом состоянии. В таком случае, при нулевом входном сигнале, n-МОП закрыт и на выходе единичное напряжение источника питания. При единичном входном сигнале оба транзистора открыты, через схему протекает ток и выходное напряжение определяется делителем, образованным МОП-транзисторами. Если сопротивление p-

МОП гораздо меньше сопротивления р-МОП, то выходное напряжение может быть достаточно близко к нулевому. Таким образом, такая схема выполняет функцию инвертора, но имеет худшие характеристики, чем КМОП-инвертор. Она потребляет ток и размах выходного сигнала не равен напряжению источника питания. Иногда такие инверторы называют квазиинверторами.

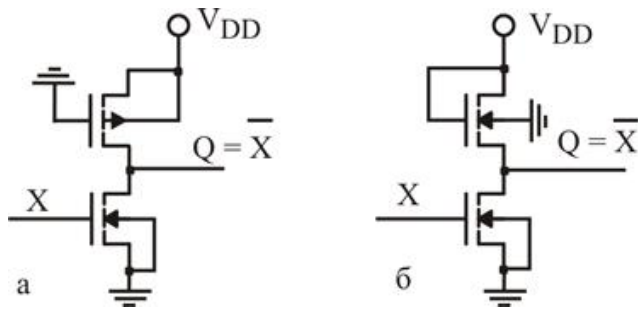


Рис. 1.5. Варианты квазиинверторов: а – р-МОП нагрузка; б – п-МОП нагрузка

Квазиинверторы могут быть построены и на транзисторах одного типа проводимости. В этом случае затвор нагрузочного транзистора подсоединяют к стоку – диодное включение МОП (рис.1.5,б). Характеристики такого квазиинвертора еще хуже, чем предыдущего, поскольку схема обладает сильной асимметрией особенно заметной в процессе переключения.

Квазиинверторы на однотипных МОП могут быть улучшены при использовании в качестве нагрузочных МОП со встроенным каналом.

#### **Достоинства КМОП-инверторов:**

- 1) в статическом состоянии цепь источника питания разомкнута, элемент не потребляет ток и не рассеивает статическую мощность;
- 2) выходное сопротивление элемента равно напряжению открытого МОП и через него выход привязан к одному из электродов источника питания. В результате с двоичные сигналы совпадают с потенциалами на электродах источника питания,  $V_H = V_{DD}$  и  $V_L = 0$ .
- 3) при одинаковых по электрическим параметрам п-МОП и р-МОП в силу симметрии схемы будут симметричны и ее характеристики.

### **1.4. Логические КМОП-элементы**

Вышеперечисленные свойства КМОП-инверторов могут быть с небольшими изменениями перенесены и на более сложные логические элементы. Практически наиболее важным является первое свойство, в силу которого статическое потребление мощности в КМОП-схемах пренебрежимо мало. Второе свойство повышает помехоустойчивость логических схем.

Среди логических элементов выделяют наборы, достаточные для синтеза произвольных логических функций, т.е. обладающие логической полнотой. Логически полными являются и отдельные элементы ИЛИ-НЕ (NOR) и И-НЕ (NAND). КМОП-вентили достаточно просто реализуют функции NOR и NAND.

Выход  $Q$  двухвходового логического элемента 2ИЛИ-НЕ зависит от входов  $X_1$  и  $X_2$

$$Q = \overline{X_1 + X_2}. \quad (1.1)$$

Рассмотрим КМОП-схему с двумя входами в которой входные сигналы поступают на р-МОП, соединенные последовательно, и на п-МОП, соединенные параллельно (рис.1.6).

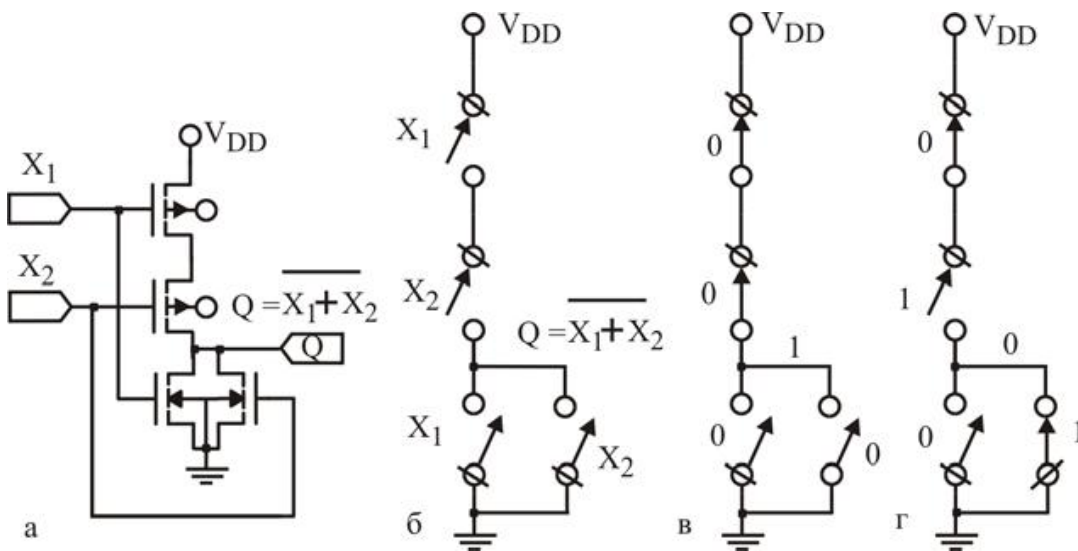


Рис.1.6. КМОП логический элемент 2ИЛИ-НЕ – NOR2: а – схема; б – ключевая модель; в – положение ключей при единичном выходе; г – положение ключей при нулевом выходе.

Если оба входа нулевые, то р-МОП включены, а п-МОП выключены, и на выходе единица. Если хотя бы один входной сигнал единичный, то цепь из двух р-МОП разомкнута, а хотя бы один п-МОП включен, и на выходе ноль. В соответствии с формулой (1.1) элемент выполняет логическую функцию 2ИЛИ-НЕ (NOR2). Легко убедиться, что он, как и инвертор, обладает первыми двумя полезными свойствами: цепь питания при всех входных сигналах разомкнута и выход привязан к источнику питания. Однако симметрия схемы нарушена. Если все транзисторы в схеме имеют одинаковое выходное сопротивление  $R_0$ , то выходное сопротивление схемы в зависимости от входных сигналов принимает одно из трех значений:  $2R_0$  при  $X_1 = X_2 = 0$ ,  $R_0$  при одном единичном сигнале,  $R_0/2$  при  $X_1 = X_2 = 1$ . Уменьшая сопротивления транзисторов, включенных по-

следовательно (р-МОП) можно снизить асимметрию схемы по выходному сопротивлению.

Для выполнения более сложных логических функций можно воспользоваться теоремой де Моргана:

$$\overline{X_1 + X_2} = \overline{X_1} \overline{X_2} ; \quad (1.2)$$

$$\overline{X_1 X_2} = \overline{X_1} + \overline{X_2} . \quad (1.3)$$

Первая формула фактически реализована в рассмотренной схеме NOR2: входные сигналы для р-МОП действуют, как проинвертированные входные, а выход п-МОП проинвертирован по отношению ко входным сигналам. Вторая формула позволит построить КМОП-схему NAND2, если п-МОП соединить последовательно, а р-МОП параллельно (рис.1.7,а).

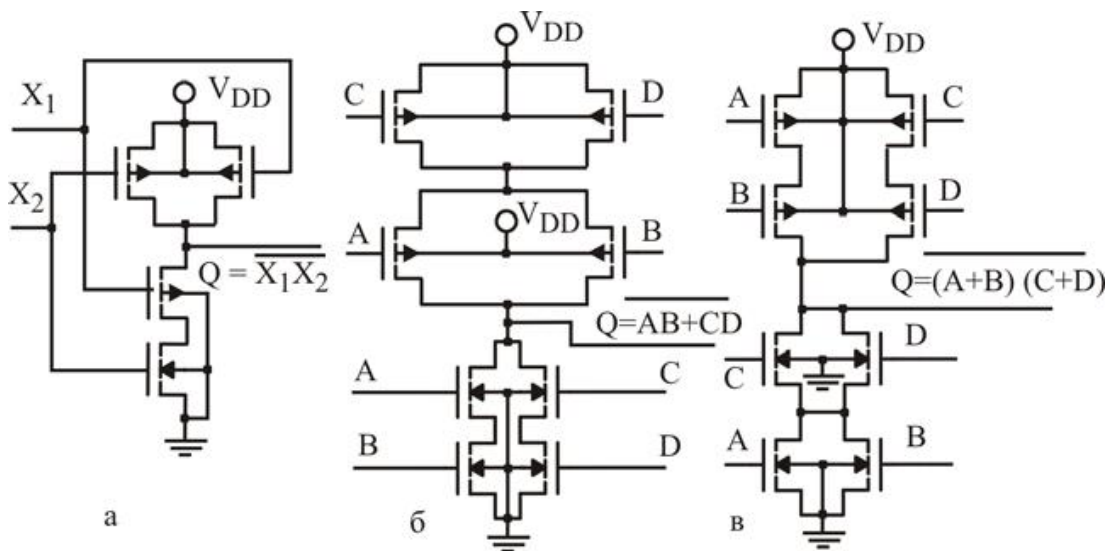


Рис.1.7. Логические КМОП-элементы: а – 2И-НЕ – NAND2; б – логическая функция задана в дизъюнктивной нормальной форме; в – логическая функция задана в конъюнктивной нормальной форме.

Обе схемы NOR2 и NAND2 состоят из двух блоков, которые разделены выходным узлом. Блок, содержащий п-МОП, размещен между выходным узлом и узлом с нулевым потенциалом. Блок, содержащий р-МОП, размещен между узлом напряжения питания и выходным узлом. Причем затворы всех МОП соединены со входами схемы.

Проанализируем два более сложных логических КМОП-элемента (рис.1.7,б и рис.1.7,в): один выполняет отрицание логической функции, заданной в дизъюнктивной нормальной форме (логическая сумма логических произведений – дизъюнкция конъюнкций), другой – отрицание логической функции, заданной в конъюнктивной нормальной форме (логическое произведение логических сумм – конъюнкция дизъюнкций). В обеих



схемах каждой цепи последовательных n-МОП, подсоединенных к некоторым входам, соответствует параллельная ветвь р-МОП, подсоединенных к тем же входам, и наоборот. Блоки КМОП-схемы, удовлетворяющие такому правилу, называют комплементарными.

Следование сформулированным правилам позволяет строить не потребляющие логические КМОП-элементы любой степени сложности. Для этого достаточно построить одну часть схемы (на n-МОП или на р-МОП) и дополнить ее комплементарной.

### 1.5. Синтез КМОП-элементов

В предыдущих разделах были приведены схемы четырех простейших КМОП-элементов: КМОП-ключа с парафазным управлением, инвертора NOT, вентиля NOR2, вентиля NAND2. Отметим, что любой из двух вентилях обладает логической полнотой, т.е. на его основе может быть реализована произвольная логическая схема. Обычно библиотека (набор стандартных элементов) гораздо обширней (рис.1.8).

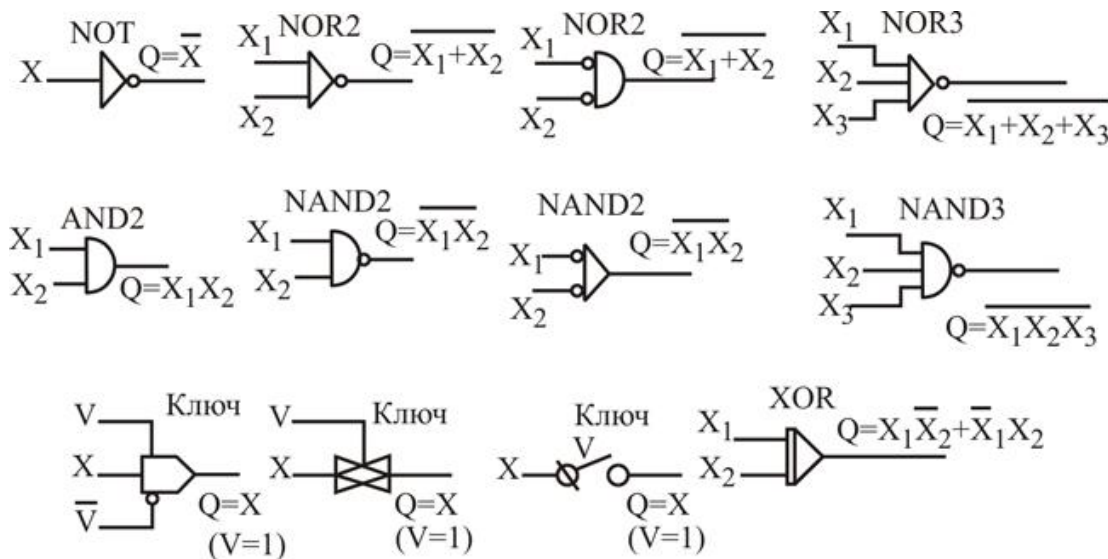


Рис.1.8. Условные обозначения логических элементов простейшей библиотеки.

Расширение библиотеки сводится к синтезу новых (часто используемых) элементов, который может проходить на различных уровнях, в том числе на транзисторном и вентильном.

**Транзисторный уровень.** Примером транзисторного синтеза может служить синтез логических схем NOR3 и NAND3. Логическая схема NOR3 получается при добавлении третьего n-МОП параллельно двум имеющимся (см. рис.1.6,а) и третьего р-МОП последовательно с двумя имеющимися. Аналогично синтезируется NAND3.

Логическая схема ИСКЛЮЧАЮЩЕГО ИЛИ (XOR) получается из схемы рис.1.7,б при  $X_1 = A = \bar{C}$ ,  $X_2 = D = \bar{B}$ .

**Вентильный уровень.** Синтез на вентильном уровне показан на примере построения КМОП-ключа с одновходовым управлением на основе КМОП-ключа с парафазным управлением (рис.1.9).

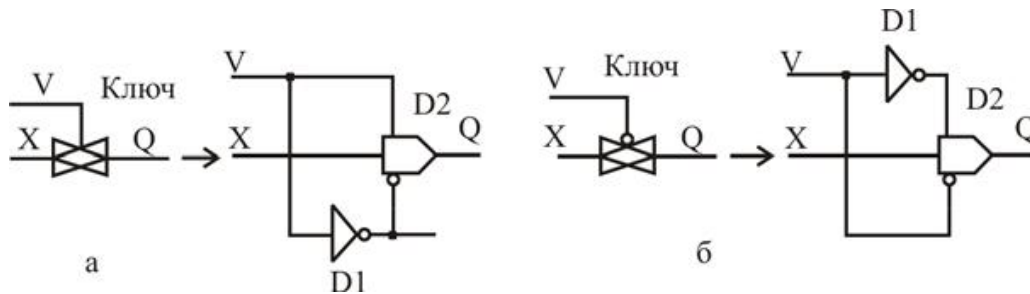


Рис.1.9. КМОП-ключи с одновходовым прямым (а) и инверсным (б) управлением

При этом одновходовое управление может быть прямым (рис.1.9,а), и инверсным (рис.1.9,б).

Даже при синтезе достаточно простых элементов возможны различные их варианты, отличающиеся как по аппаратным затратам (количеству требуемых МОП), так и по другим параметрам (быстродействию, нагрузочной способности по входу и выходу, и т.д.) Покажем это на примере схемы XOR (рис.1.10).

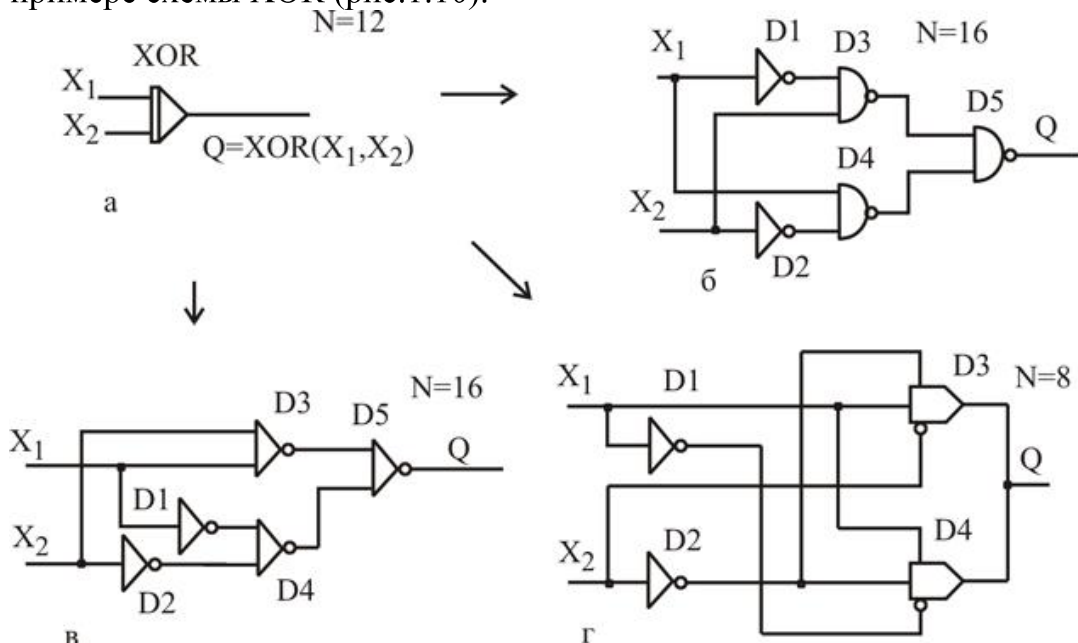


Рис.1.10. Синтез логической схемы ИСКЛЮЧАЮЩЕЕ ИЛИ (N – количество транзисторов): а – исходная схема; б – реализация в базисе NAND2; в – реализация в базисе NOR2; г – реализация с использованием КМОП-ключей.



При синтезе на транзисторном уровне требуется 12 МОП (4 МОП в двух инверторах и 8 МОП в схеме рис.1.7,б), рис.1.10,а. При синтезе XOR и в базисе NAND2 (рис.1.10,б) и в базисе NOR2 (рис.1.10,в) требуется 16 МОП, т.е. необходимо большее количество транзисторов при лучших нагрузочных способностях. При синтезе XOR с использованием КМОП-ключей требуется всего 8 МОП, но нагрузочные способности очень слабые. За счет увеличения количества транзисторов до 10 нагрузочные способности этого варианта могут быть улучшены.

## Глава 2. Инвертор

Одновходовая логическая КМОП-схема – инвертор представляет собой простейший КМОП-усилитель, работающий в режиме большого сигнала. Проведенный ранее анализ статических и динамических (частотных) характеристик простейших усилителей (см. ч.1, гл.4) нужно дополнить рассмотрением его переходных характеристик – поведением в режиме больших сигналов. Современные цифровые схемы для достижения высокого быстродействия работают с короткоканальными МОП-транзисторами, у которых длина канала порядка минимального топологического размера  $F$  и характеристики которых во многом отличаются от характеристик длинноканальных МОП-транзисторов, (рис.2.1).

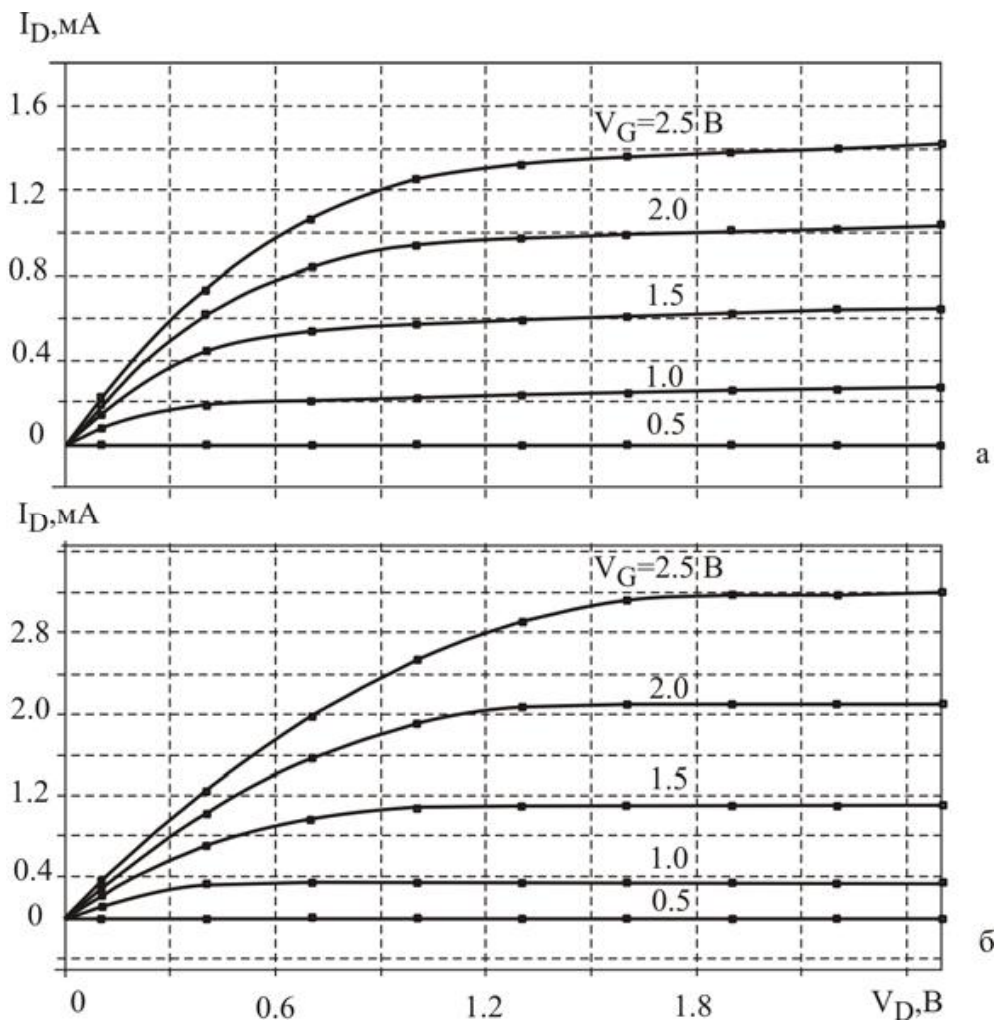


Рис.2.1. ВАХ n-МОП: а –  $W/L = 2.5/0.25$ ; б –  $W/L = 25/2.5$  (мкм).

## 2.1. Короткоканальный транзистор.

При минимальной длине канала он становится сравнимым с шириной области обеднения стокового р–n-перехода, электрическое поле в канале возрастает и в поведении транзисторов начинают оказывать влияние короткоканальные эффекты.

Основные отличия ВАХ короткоканальных транзисторов от длинноканальных, как это видно из рис.2.1, состоят в следующем:

- 1) уменьшен ток насыщения;
- 2) насыщение наступает при более низких стоковых напряжениях;
- 3) уменьшена крутизна – увеличено выходное сопротивление в линейной области;
- 4) уменьшено выходное сопротивление в насыщении.

Эти отличия обусловлены прежде всего ограничением подвижности носителей при увеличении напряжения на стоке, вызванным высоким уровнем легирования канала и большим вертикальным (создаваемым напряжением на затворе) электрическим полем. Так, для типового 0,25 мкм n-МОП подвижность носителей уменьшается в два раза уже при напряжении на стоке, равном примерно половине напряжения питания (2,5 В / 2).

Имеют место и другие проявления короткоканальных эффектов: зависимость порогового напряжения от напряжения на стоке, возрастание токов утечки, модуляция длины канала снижением барьера в канале и др.

Учет основных изменений в короткоканальных транзисторах, существенно влияющих на поведение цифровых элементов, требует усложнения ранее введенной модели МОП-транзистора. Наиболее простой способ коррекции состоит в введении нового независимого параметра – напряжения насыщения  $V_{DSat}$ .

Выходная ВАХ короткоканального транзистора записывается в виде

$$I_D = \beta(V_0 V_{min} - 1/2 V_{min}^2), \quad (2.1)$$

где  $\beta = (W/L) k = (W/L) \mu C_{ox}$  – крутизна МОП-транзистора;  $k = \mu C_{ox}$  – удельная крутизна МОП-транзистора;  $V_0 = (V_G - V_t)$  – эффективное напряжение;  $V_{min} = \min(V_0, V_D, V_{DSat})$ .

Отметим, что подвижность  $\mu$  короткоканального транзистора может отличаться от подвижности длинноканального транзистора, а значит отличаются  $k$  и  $\beta$ .

Формула (2.1) преобразуется в формулу для длинноканальных транзисторов, если положить  $V_0 = V_D = V_{DSat}$ , т.е. вернуть напряжение насыщения в точку перехода от линейного режима к режиму насыщения. Фактически у короткоканальных транзисторов насыщение происходит

раньше, чем у длинноканальных. Но оно обусловлено не перекрытием канала, а ограничением скорости носителей и отличается более плавным характером.

Видоизменение модели эквивалентно введению нового режима – режима ограничения скорости (рис.2.2).

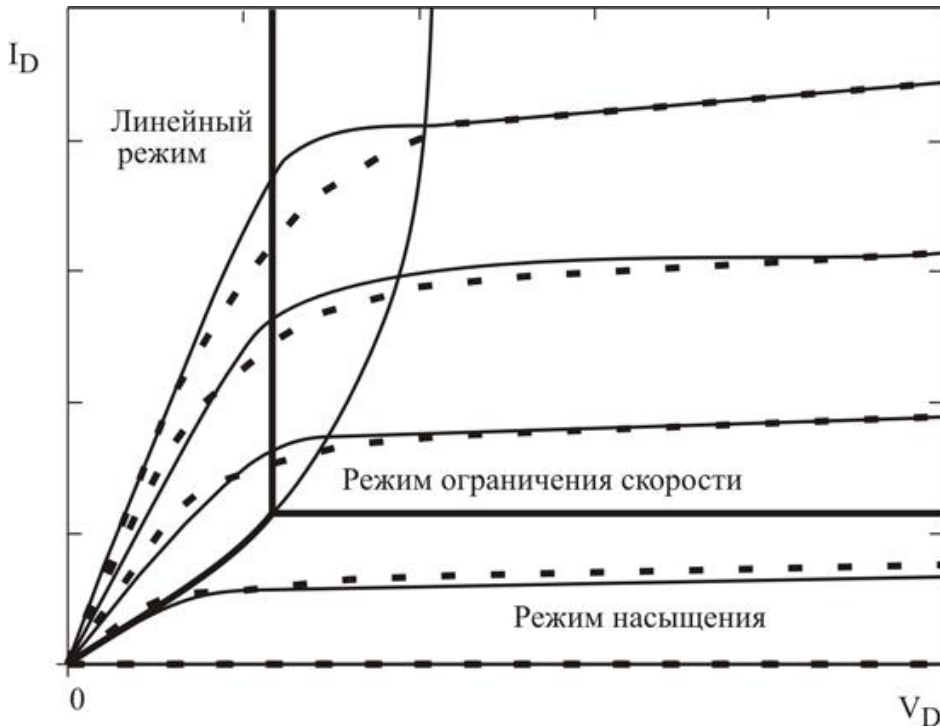


Рис.2.2 Режимы работы короткоканального транзистора.

## 2.2. Передаточная характеристика инвертора

Выходные ВАХ n-МОП и p-МОП позволяют построить передаточную характеристику инвертора на короткоканальных транзисторах (рис.2.3) аналогично тому, как это сделано ранее для инвертора на длинноканальных транзисторах (см. ч.1, гл. 4).

Передаточная характеристика позволяет определить рабочий диапазон инвертора, зависящий от предъявляемых к схеме требований.

Основными параметрами передаточной характеристики являются:

- 1) напряжение переключения;
- 2) максимальное нулевое входное напряжение;
- 3) минимальное единичное входное напряжение;
- 4) максимальное нулевое выходное напряжение;
- 5) минимальное единичное выходное напряжение.

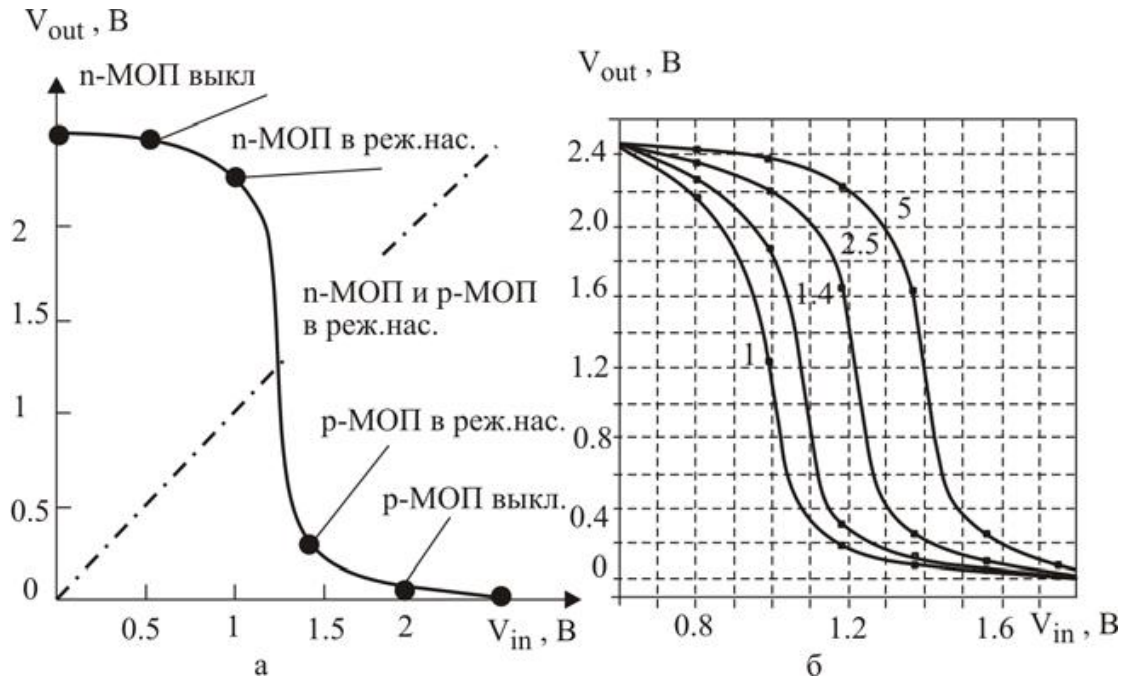


Рис.2.3. Передаточная характеристика инвертора: а – общий вид; б – при разных аспектных отношениях ( $W_p/W_n$ ).

Напряжение переключения  $V_M$  определяется из равенства токов через транзисторы, находящиеся в режиме насыщения (без учета их выходных сопротивлений):

$$k_n V_{DSatn} (V_M - V_{tn} - V_{DSatn} / 2) + k_p V_{DSatp} (V_M - V_{DD} - V_{tp} - V_{DSatp} / 2) = 0 \quad (2.2)$$

Из (2.2) находим

$$V_M = \frac{(V_{tn} + V_{DSatn} / 2) + r \cdot (V_{DD} + V_{tp} + V_{DSatp} / 2)}{1 + r}, \quad (2.3)$$

где (при равных длинах каналов n-МОП и p-МОП)

$$r = \frac{k_p V_{DSatp}}{k_n V_{DSatn}} = \frac{\mu_p}{\mu_n} \frac{W_p}{W_n}. \quad (2.4)$$

Если у n-МОП и p-МОП одинаковы по абсолютной величине пороги и напряжения насыщения, то (2.3) упрощается

$$V_M = \frac{r}{1 + r} V_{DD} + \frac{(1 - r)(V_t + V_{DSat} / 2)}{1 + r}. \quad (2.5)$$

Напряжение переключения теперь зависит от аспектного отношения транзисторов инвертора и равно половине напряжения  $V_M = V_{DD}/2$  при  $r = 1$ , т.е. когда транзисторы согласованы  $W_p/W_n = \mu_n/\mu_p$ . При этом передаточная характеристика симметрична, (см. рис.2.3,б) и выходные

сопротивления при обоих выходных сигналах одинаковы. Поскольку такие свойства улучшают помехоустойчивость и быстродействие схем, то во многих случаях используют именно согласованный инвертор, в частности, минимальный согласованный инвертор, который занимает минимальную площадь. Его n-МОП имеет аспектное отношение ( $W_{\min}/F$ ), а p-МОП –  $(\mu_n/\mu_p W_{\min}/F)$ , где  $F$  – минимальный топологический размер,  $W_{\min}$  – минимально допустимая ширина канала МОП (обычно  $W_{\min} = 2 \div 2.5 F$ ). Минимальный согласованный инвертор характеризуется аспектным отношением  $W_p/W_n = \mu_n/\mu_p$ .

В некоторых случаях используют минимальный инвертор с  $W_p/W_n = 1$ . Его n-МОП и p-МОП имеют одинаковые минимальные аспектные отношения ( $W_{\min}/F$ ), а выходные сопротивления различаются в  $\mu_n/\mu_p$  раз. Напряжение переключения минимального инвертора смещено к нулевому напряжению и равно

$V_M = V_{DD}/(\mu_n/\mu_p + 1) - [(\mu_n/\mu_p - 1)(\mu_n/\mu_p + 1)](V_t + V_{DSat}/2)$ . Так, для 0,25КМОП ( $V_{DD}=2,5$  В,  $\mu_n/\mu_p = 2,5$ ,  $V_t = 0,5$  В,  $V_{DSat} = 1$  В) у минимального инвертора  $V_M \approx 1,1$  В, а у согласованного –  $V_M \approx 1,25$  В.

Интересен вариант инвертора с гармоническим аспектным отношением, у которого  $W_p/W_n = (\mu_n/\mu_p)^{1/2}$ , а  $V_M \approx 1,2$  В.

В статическом режиме КМОП-вентиль, например NAND2 на рис.1.7,а, эквивалентен инвертору с некоторым  $r$ , зависящим от комбинации входных сигналов. Например, если в согласованной схеме NAND2 объединить входы, то  $r = 2$ . При этом  $V_M$  смещается к напряжению питания  $V_{DD}$  (рис.2.4)

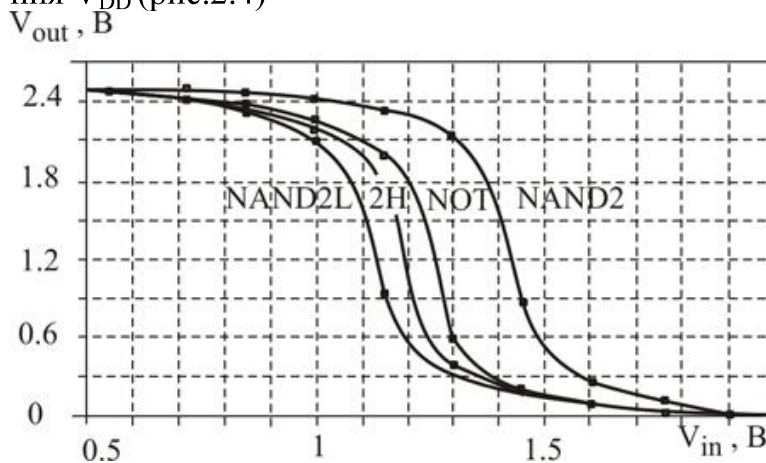


Рис.2.4. Передаточные характеристики КМОП-схем: NAND2, NOT, NAND2H, NAND2L.

Если на одном из входов NAND2 задано постоянно единичное напряжение, то передаточные характеристики сдвигаются к нулевому напряжению (кривая NAND2H на рис.2.4). Сдвиг сильнее, при изменяющемся сигнале на верхнем n-МОП (кривая NAND2L на рис.2.4), поскольку к

его истоку подключен открытый n-МОП, что эквивалентно введению отрицательной обратной связи.

**Диапазон помехоустойчивости** – диапазон входных напряжений, при которых модуль коэффициента усиления инвертора не превышает единицу и находится из условия

$$|dV_{out}/dV_{in}| < 1.$$

Для определения помехоустойчивости инвертора линеаризуем его передаточную характеристику (рис.2.5).

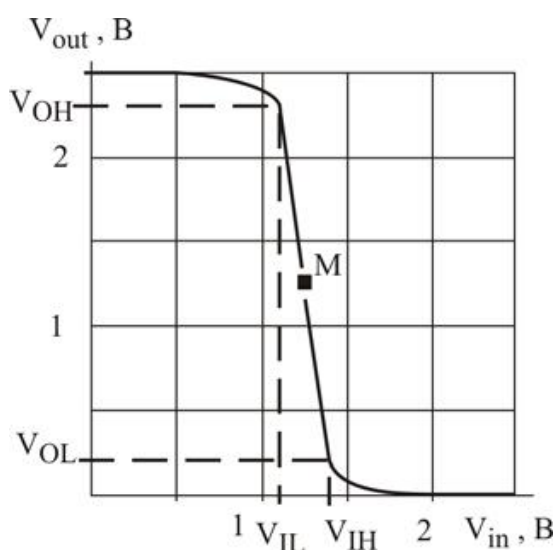


Рис.2.5. Линеаризованная передаточная характеристика инвертора

Тогда для переходной области справедливо выражение

$$V_{IH} - V_{IL} \approx -(V_{OH} - V_{OL})g \approx -V_{DD}/g, \quad (2.6)$$

где  $g$  – коэффициент усиления в точке переключения. Для его определения необходим учет выходных сопротивлений

$$g = \frac{1}{I_D(V_M)} \frac{k_n V_{DSatn} + k_p V_{DSatp}}{\lambda_n - \lambda_p} \approx \frac{1+r}{(V_M - V_{tn} - V_{DSatn}/2)(\lambda_n - \lambda_p)}.$$

Поскольку линеаризованная передаточная характеристика проходит через точку переключения  $M$ , то приближенно

$$V_{IH} = V_M - V_M/g \quad (2.7)$$

$$V_{IL} = V_M + \frac{V_{DD} - V_M}{g} \quad (2.8)$$

Запас помехоустойчивости инвертора определяется как модуль разности между входным и выходным сигналом одного двоичного уровня. Запас по нижнему (нулевому) уровню равен

$$N_{ML} = V_{IL} - V_{OL} \quad (2.9)$$

Запас по верхнему (единичному) уровню равен

$$N_{MH} = V_{IH} - V_{OH} \quad (2.10)$$

### 2.3. Быстродействие инвертора

Рассмотрим поведение инвертора при работе на емкостную нагрузку (рис.2.6). Для простоты будем полагать, что  $\mu_n/\mu_p=2$ , а инвертор согласован, т.е.  $W_p/W_n = 2$ .

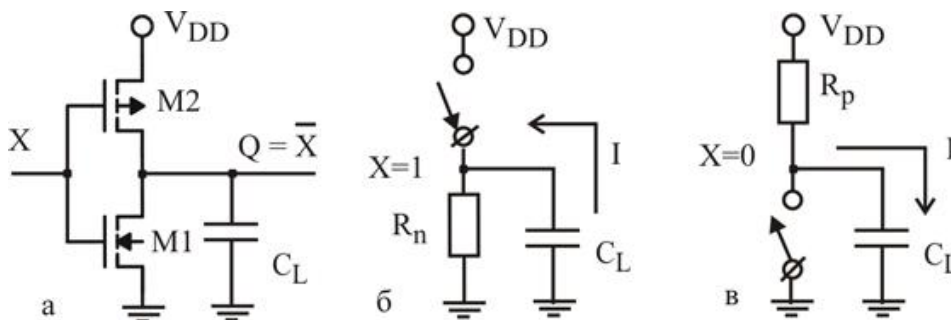


Рис.2.6. Работа инвертора на емкостную нагрузку: а – электрическая схема; б – n-МОП включен; в – p-МОП включен.

Переходные процессы, протекающие в схеме при изменении входного напряжения, носят достаточно сложный характер, поскольку происходит заряд (разряд) емкости  $C_L$  через два нелинейных элемента, n-МОП и p-МОП. Однако в первом приближении считают, что при единичном входном сигнале емкость разряжается через n-МОП, а p-МОП закрыт (рис.2.6,б). При нулевом сигнале, наоборот, емкость заряжается через p-МОП, а n-МОП закрыт (рис.2.6,в).

Наиболее важным параметром, характеризующим цифровые схемы, является время задержки элемента  $t_p$  – интервал от середины перепада входного напряжения до середины перепада выходного напряжения (рис.2.7).



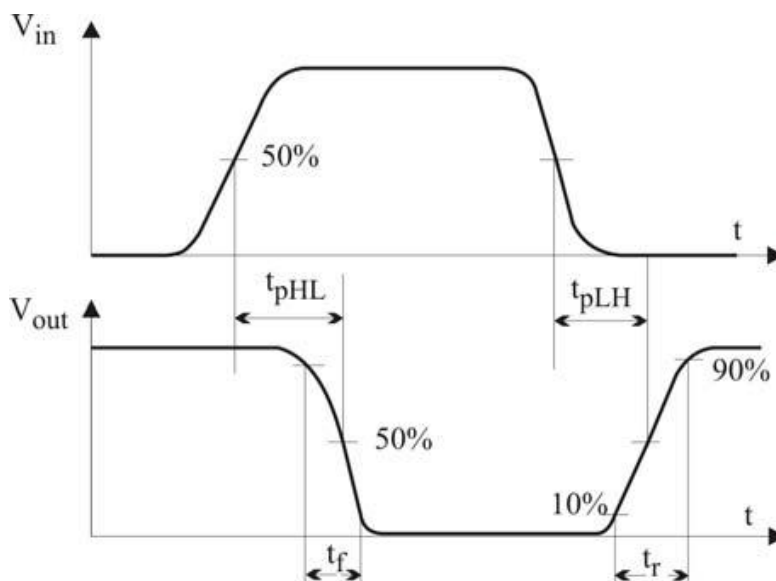


Рис.2.7. Временная диаграмма работы инвертора

Длительности (время) нарастания  $t_r$  и спада  $t_f$  выходного сигнала также имеют большое значение. Для нормальной работы КМОП-клапанов они не должны быть большими, во всяком случае не должны значительно превышать время задержки клапана

Для оценки времени задержки рассматривают упрощенный процесс – заряд емкости нагрузки  $C_L$  через некоторое эквивалентное сопротивление инвертора. Величина эквивалентного сопротивления  $R_{eq}$  зависит от режима работы. Так, если напряжение на транзисторе снижается от напряжения питания  $V_{DD}$  до  $V_{DD}/2$ , то усреднение выходного сопротивления дает

$$R_{eq} = \frac{1}{-V_{DD}/2} \int_{V_{DD}}^{V_{DD}/2} \frac{V}{I_{DSat}(1 + \lambda V)} dV \approx \frac{3}{4} \cdot \frac{V_{DD}}{I_{DSat}} \left(1 - \frac{7}{9} \cdot \lambda V_{DD}\right) \quad (2.11)$$

)

Оценка изменится незначительно, если воспользоваться средним – полусуммой выходных сопротивлений при соответствующих напряжениях

$$R_{eq} = \frac{1}{2} \cdot \left( \frac{V_{DD}}{I_{DSat}(1 + \lambda V_{DD})} + \frac{V_{DD}/2}{I_{DSat}(1 + \lambda V_{DD}/2)} \right) \approx \frac{3}{4} \frac{V_{DD}}{I_{DSat}} \left(1 - \frac{5}{6} \lambda V_{DD}\right) \quad (2.11.a)$$

Разряд емкости нагрузки  $C_L$  через эквивалентное сопротивление n-МОП до половины питания  $V_{DD}$  происходит за время

$$t_{pHL} = (\ln 2) R_{eqn} C_L, \quad (2.12)$$

а заряд от нулевого потенциала до уровня  $V_{DD}/2$  через p-МОП за время

$$t_{pLH} = (\ln 2) R_{eqp} C_L. \quad (2.12,a)$$

При согласованных n-МОП и p-МОП средняя задержка инвертора при подаче на его вход перепада напряжения величиной  $V_{DD}$  равна

$$t_p = 0,69 C_L (R_{egn} + R_{egp}) / 2 = 0,69 * 0,75 (C_L V_{DD}) / I_{DSat} = \dots\dots 0,52 (C_L V_{DD}) / I_{DSat} \quad (2.13)$$

При подаче на вход инвертора сигнала с конечной длительностью фронта ее можно учесть по-разному. Если длительность фронта  $t_s$  сравнима с временем задержки  $t_p$ , то для оценки задержки при конечном фронте  $t_{ps}$  можно воспользоваться гармоническим средним

$$t_{ps} = \left[ (t_p)^2 + (t_s/2)^2 \right]^{1/2} \quad (2.14)$$

В таком приближении при  $t_s = 2t_p$  задержка инвертора  $t_{ps}$  вырастет в  $\sqrt{2}$  раз.

Более удобно пользоваться линейным приближением  $t_{ps}$ , например, в виде

$$t_{ps} = t_p + k_{ps} t_s (1 - 2V_t / V_{DD}) \quad (2.14.a)$$

где  $k_{ps}$  – эмпирическая константа.

Некоторые цифровые схемы критичны к длительности фронтов входных сигналов и значительное возрастание времени переключения недопустимо. Поэтому при работе КМОП-схем на большую емкостную нагрузку применяют специальные формирователи (драйверы). В качестве таких схем можно использовать цепочку инверторов с возрастающим аспектным отношением.

## 2.4. Цепь инверторов

Нагрузочная емкость при работе инвертора на аналогичные инверторы складывается из выходной емкости инвертора, емкости связи и емкости входных затворов  $n$  инверторов. При согласованных инверторах входная емкость представляет собой утроенную емкость затвора n-МОП. Для емкости нагрузки имеем

$$C_L = C_{dn} + C_{dp} + nC_O = C_{dn} (1 + W_p / W_n) + C_1 + n(1 + W_p / W_n) C_g = 3C_{dn} + C_1 + 3nC_g \quad (2.15),$$

где  $C_{dn}$  – емкость стока n-МОП в инверторе;  $C_{dp}$  – емкость стока p-МОП в инверторе;  $C_g$  – емкость затвора n-МОП в инверторе;  $C_1$  – емкость соединений;  $n$  – количество инверторов в нагрузке;  $C_O$  – входная емкость инвертора.

Обозначим задержку идеального инвертора (без учета его выходной емкости и емкости связей), нагруженного на аналогичный инвертор, через

$$t_0 = 0,52(C_O V_{DD})/I_{DSat} . \quad (2.16)$$

Отметим, что  $t_0$  не зависит от размеров инверторов (ширины затворов транзисторов), так как размеры определяют в равной степени и емкости, и токи.

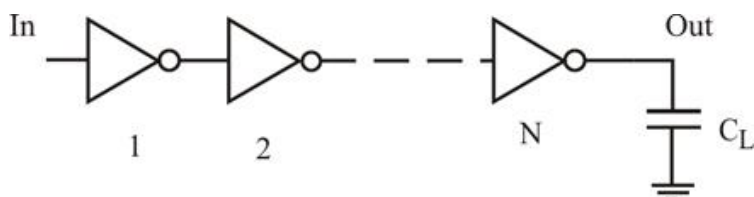


Рис.2.8. Цепочка инверторов с емкостной нагрузкой

Суммарная задержка цепочки неидеальных инверторов (рис.2.8) с коэффициентом размножения  $n$  и коэффициентом неидеальности  $\gamma = (C_{dn} + C_{dp})/C_0$  равна

$$t_{\Sigma} = \sum_1^N (\gamma + n) t_0 . \quad (2.17)$$

Из (2.17) следует линейный рост задержки с увеличением количества каскадов  $N$ .

Нагрузка, которую за время  $t_0$  успевает зарядить очередной каскад, растет по степенному закону

$$C_L = n^N C_O . \quad (2.18)$$

Количество каскадов, необходимое для перехода от входной емкости  $C_0$  к емкости нагрузки  $C_L$  вытекает из (2.18)

$$N = \log_n (C_L / C_O) = \ln(C_L / C_g) / \ln(n) \quad (2.19)$$

Подставляя (2.19) в (2.17) получаем

$$t_{\Sigma} = [\ln(C_L / C_O) / \ln(n)] (\gamma + n) t_0 \quad (2.20)$$

Функция  $t_{\Sigma}(n)$  имеет минимум, зависящий от параметра  $\gamma$ . Для идеального инвертора оптимальное  $n \approx 3$ , при  $\gamma = 1$  оптимальное  $n = 4$ . Таким образом, в КМОП-схемах, для достижения высокого быстродействия коэффициент размножения по выходу не должен превышать 4.

Цепочка из нечетного количества одинаковых инверторов, в которой выход замкнут на вход, образует кольцевой генератор (рис.2.9), с периодом генерации равным удвоенной задержке цепочки  $t_{\Sigma}$ .

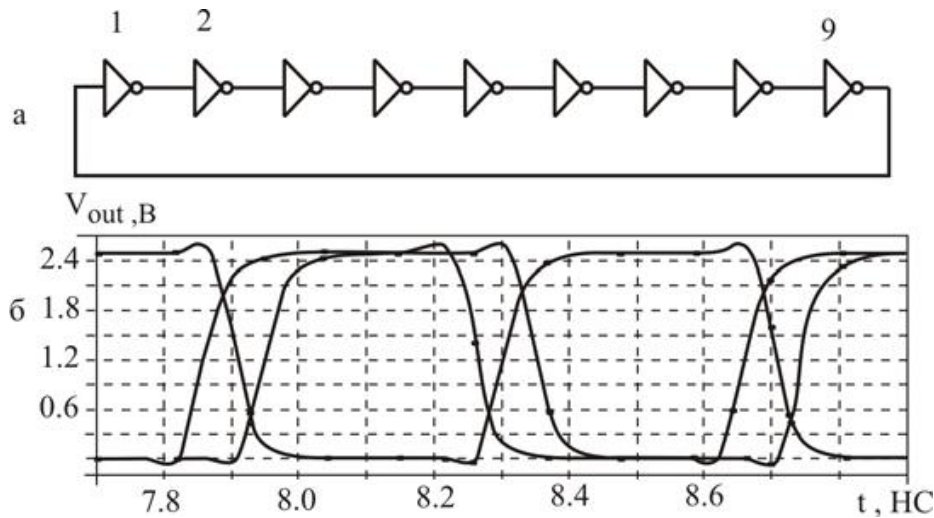


Рис.2.9. Кольцевой генератор ( $N = 9$ ): а – схема; б – временная диаграмма (форма выходного сигнала трех последовательных вентилей).

## 2.5. Потребляемая мощность

Потребляемая мощность – один из важнейших параметров инвертора. В КМОП-инверторе она складывается из трех составляющих:

$$P = P_s + P_d + P_{sc} \quad (2.21)$$

где  $P_s$  – статическая мощность;  $P_d$  – динамическая мощность;  $P_{sc}$  – мощность, обусловленная сквозными токами при переключении МОП.

Статическая мощность связана с токами утечки и сравнительно невелика.

Динамическая мощность тратится на перезарядку конденсаторов и определяет основной вклад в общее потребление

$$P_d = f C_L V^2 = f C_L V_{DD}^2, \quad (2.22)$$

где  $f$  – частота переключения;  $C_L$  – емкость нагрузки;  $V$  – перепад напряжений на емкости нагрузки.

Мощность, обусловленная со сквозными токами может составлять значительную часть общей мощности. Она зависит от длительности фронтов на входах схемы и от выходной емкости. Длинный фронт увеличивает время, в течение которого n-МОП и p-МОП открыты одновременно. Большая величина емкости ограничивает сквозной ток, так как у закрывающегося транзистора напряжение между стоком и истоком остается небольшим длительное время. Для  $P_{sc}$  справедлива оценка

$$P_{sc} \approx V_{DD} (I_{DSat} k_{sc}) [(t_t + t_r)/2] f = P_d k_{sc} \quad (2.23)$$

где  $k_{sc}$  – коэффициент, определяющий сквозной ток. Обычно  $k_{sc} = 0,1 - 0,2$ .

### 3. КМОП-вентили



Рис.3.1. Структура КМОП-вентилей

Они присоединяют выход  $Q$  либо к источнику питания  $V_{DD}$ , либо к нулевому потенциалу. В КМОП-вентилей каждая комбинация входов включает или блок  $p$  или блок  $n$ , поэтому выход однозначно определяется входными сигналами.

Поскольку  $n$ -МОП ( $p$ -МОП) включаются положительными (отрицательными) потенциалами, а присоединяют выход к нулевому (высокому) потенциалу, то КМОП-вентиль обладает встроенным инвертированием.

Логическая функция КМОП-вентилей определяется его элементарными блоками. Синтез вентилей по заданной функции задается конкретной процедурой. Можно синтезировать как простые логические выражения, так и составные.

Анализ характеристик КМОП-вентилей проводится аналогично анализу элементарного вентилей – инвертора – с учетом различных комбинаций входных сигналов.

Более того, во многих случаях поведение вентилей может рассматриваться, как поведение некоторого эквивалентного инвертора.

#### 3.1. Согласованные КМОП-вентили

В простейшем КМОП-вентилей – инверторе – двум возможным входным сигналам соответствуют проинвертированные выходные сигналы и каждому из них – определенное выходное сопротивление, зависящее от размеров  $n$ -МОП и  $p$ -МОП. В частности, у согласованного инвертора за счет выбора размеров транзисторов выходные сопротивления, а значит и задержки выходных сигналов, одинаковы (рис.3.2,а).

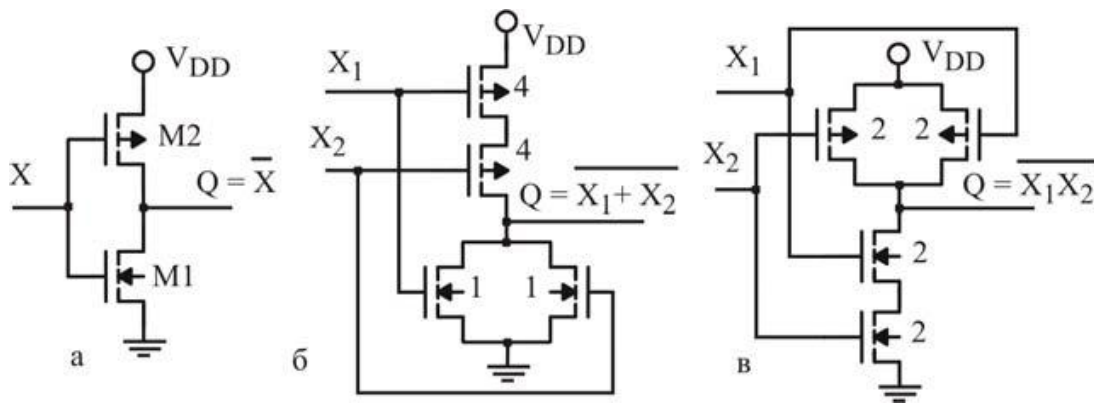


Рис.3.2. Согласованные КМОП-вентили: а – инвертор; б – NOR2; в – NAND2

В произвольных КМОП-вентилей получить такое же согласование труднее. В КМОП-вентилей выходное сопротивление может принимать различные значения даже при одинаковом выходном сигнале. Так, при нулевом выходе в схеме NOR2 открыты либо один, либо два n-МОП (рис.3.2,б), а в схеме NAND2 при единичном выходе открыт либо один, либо два p-МОП (рис.3.2,в). Однако выбором размеров транзисторов можно добиться, чтобы максимальное выходное сопротивление КМОП-вентилей не превышало заданной величины, например выходного сопротивления согласованного инвертора. Такие размеры МОП ( $W/L$ ) и приведены на рис.3.2.

Согласование КМОП-вентилей по выходному сопротивлению не означает их согласования по остальным параметрам. Такая асимметрия КМОП-вентилей проявляется и в динамике. Скорость переключения КМОП-вентилей зависит от направления (из 0 в 1 –  $0 \uparrow 1$  или из 1 в 0 –  $1 \downarrow 0$ ) и от комбинации сигналов (одновременно или в какой последовательности), что продемонстрировано в табл. 3.1, где приведены времена задержки для двух значений аспектов вентилей ( $W_p/W_n$  при одинаковых  $L=0.25$  мкм).

#### Зависимость времени задержки

#### NAND2 от входных сигналов Таблица.3.1.

Входы		Выход, Q	Время задержки, пс	
X1	X2		$W_p/W_n=2.5$	$W_p/W_n=1.5$
$0 \uparrow 1$	$0 \uparrow 1$	$1 \downarrow 0$	82(67)	73(62)
1	$0 \uparrow 1$	$1 \downarrow 0$	71	64
$0 \uparrow 1$	1	$1 \downarrow 0$	77	70
$1 \downarrow 0$	$1 \downarrow 0$	$0 \uparrow 1$	40(57)	54(81)
1	$1 \downarrow 0$	$0 \uparrow 1$	68	91
$1 \downarrow 0$	1	$0 \uparrow 1$	78	110

Примечание: В скобках для сравнения даны времена задержки инвертора

### 3.2. Параметры быстродействия вентиляей.

Для оценки быстродействия логических схем необходимы упрощенные макромодели вентиляей, параметры которых просто определить по их структуре и размерам транзисторов, а при необходимости измерить экспериментально.

На задержку вентиля  $t_p$  оказывает влияние выходная цепь – коэффициент разветвления  $N_{out}$  (количество подключенных нагрузок) (рис.3.3,а) и входная цепь – коэффициент объединения  $N_{in}$  (количество входов) (рис.3.3,б).

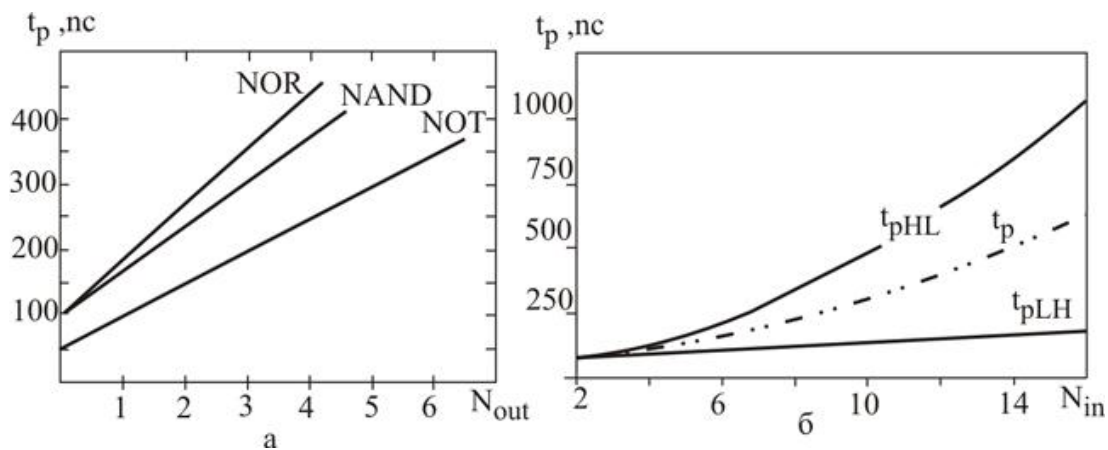


Рис.3.3. Зависимость времени задержки  $t_p$  различных вентиляей от коэффициента разветвления (а) и от коэффициента объединения (б).

Задержка линейно зависит от числа выходов КМОП, потому что каждый выход (затвор) добавляет определенную емкость и нелинейно зависит от числа входов, так как входные транзисторы включены последовательно и представляют собой распределенную структуру (особенно при большом их количестве). Так, в схеме NAND4 (рис.3.4),  $t_p \sim (C_1 + 2C_2 + 3C_3 + 4C_4)$ , поскольку каждый последующий конденсатор разряжается через возрастающее сопротивление.

В общем случае задержка КМОП-вентиля может быть представлена в виде

$$t_p = a_1 N_{in} + a_2 N_{in}^2 + a_0 N_{out}, \quad (3.1)$$

где  $a_0$ ,  $a_1$ ,  $a_2$  – феноменологические коэффициенты.

С увеличением количества входов резко растет время, что ограничивает применение многовходовых КМОП-вентиляей. Обычно используют вентили с количеством входов не более четырех.

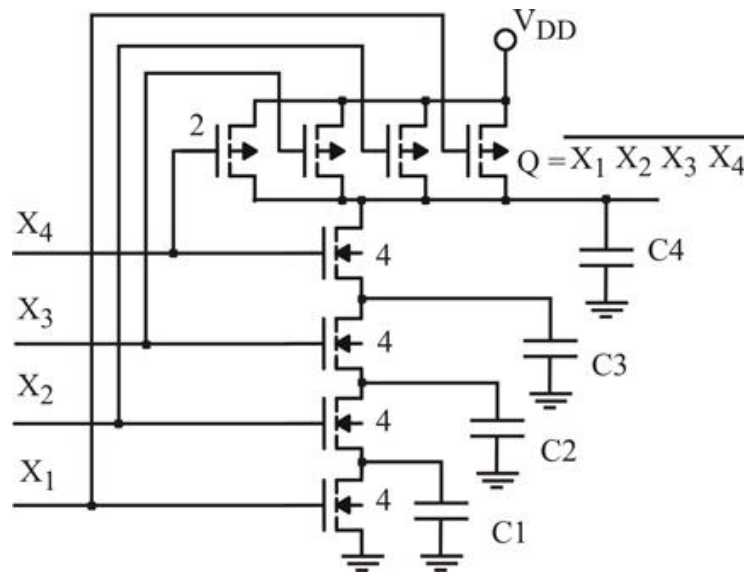


Рис.3.4. Схема многовходового вентиля – NAND4 с паразитными емкостями.

Задержку, зависящую от входной цепи, назовем внутренней (паразитной) задержкой:  $t_{pin} = a_1 N_{in} + a_2 (N_{in})^2$ , а задержку, зависящую от выходной цепи, назовем внешней (нагрузочной):  $t_{pout} = a_0 N_{out}$ . Выражение (3.1) запишем в виде

$$t_p = t_{pin} + t_{pout} \quad (3.1,a)$$

При подаче сигнала на вход согласованного вентиля сигнал на выходе появляется с задержкой, не превышающей максимальной задержки  $t_p$ . У несогласованного вентиля необходимо рассматривать две задержки: максимальную задержку включения (перехода из 0 в 1 –  $0 \uparrow 1$ ) и максимальную задержку выключения (перехода из 1 в 0 –  $1 \downarrow 0$ ).

Внешняя задержка пропорциональна величине емкости нагрузки  $C_{out}$  и может быть выражена через задержку вентиля  $t_g$ , нагруженного на внешнюю емкость, равную его входной емкости  $C_{in}$ :

$$t_p = t_{pin} + t_{pout} = t_{pin} C_{out} / C_{in} \quad (3.1,б)$$

Если входы вентиля несимметричны или имеют разные входные емкости, то и задержки по разным входам будут различны.

### 3.3. Эквивалентный инвертор

Анализ схем на произвольных вентилях удобно свести к анализу схем на инверторах – эквивалентных вентилях, для которых уже были получены оценки быстродействия, т.е. было установлено, что для цепочки КМОП-инверторов оптимальная нагрузка отдельного каскада равна четырем. С этой целью представим  $t_g$  через задержку согласованного инвертора  $t_0$ :



$$t_g = (C_{gate} R_{gate} / C_{inv} R_{inv}) t_0 = g t_0. \quad (3.2)$$

Введем безразмерные величины:  $g = (C_{gate} R_{gate} / C_{inv} R_{inv})$  – логическая нагрузка;

$h = C_{out} / C_{in}$ , – электрическая нагрузка;  $f = t_{pout} / t_0$ , – нагрузочная задержка;  $p = t_{pin} / t_0$ , – паразитная нагрузка;  $d = t_p / t_0$ , – задержка (общая). Теперь задержку вентиля (3.1,б) можно представить в виде безразмерной задержки эквивалентного инвертора

$$d = p + f = p + gh \quad (3.3)$$

Таким образом, цепочку вентиля можно заменить цепочкой эквивалентных инверторов с задержками согласно (3.3).

Цепочка из  $N$  последовательных вентилях с возрастающим аспектным отношением транзисторов, равным  $n$ , позволяет передать двоичный сигнал с емкости  $C_{in}$  на емкость  $C_{out}$

$$C_{out} \prod_{i=1}^N g_i = C_{in} n^N \quad (3.4)$$

с задержкой

$$t_d = \sum_i^N t_{di}$$

Если же сигнал в цепочке разветвляется с коэффициентом разветвления  $b = (C_{on} + C_{off}) / C_{on}$ , ( $C_{on}$  – емкость нагрузки в цепочке,  $C_{off}$  – емкость нагрузки вне цепочки), то (3.4) приобретает вид

$$C_{out} \prod_{i=1}^N g_i b_i = C_{in} n^N. \quad (3.4.a)$$

Таким образом, возрастание логической нагрузки вентилях и наличие разветвлений эквивалентно увеличению нагрузки.

### 3.4 Варианты вентилях

Параметры быстродействия вентиля (логическая нагрузка  $g$ , паразитная задержка  $p$ ) зависят от принципа положенного в основу его построения. В согласованных вентилях за основу принято равенство максимальных выходных сопротивлений (они обеспечивают одинаковые задержки переключения); в гармонических вентилях – минимальное среднее выходное сопротивление (они обеспечивают минимальную среднюю задержку переключения); в «минимальных» вентилях согласованность максимальных выходных сопротивлений с выходным сопротивлением минимального инвертора (они обеспечивают минимальную задержку переключения из единицы в ноль).

Простейшие КМОП-вентили (рис.3.5), характеризуются шириной каналов n-МОП и p-МОП.

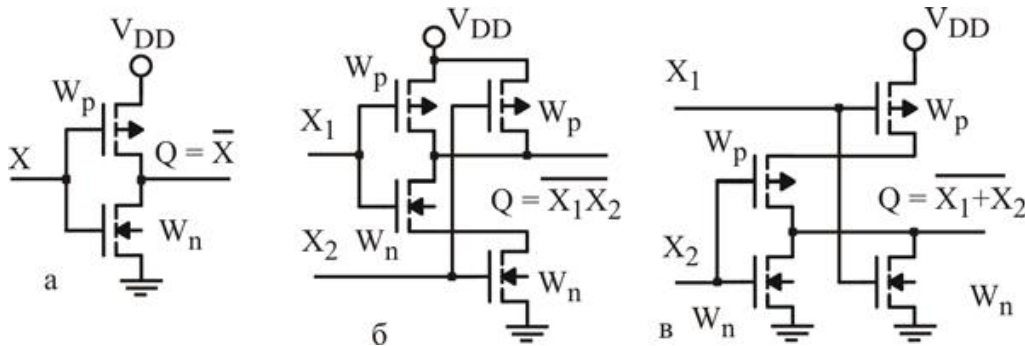


Рис.3.5. Простейшие вентили: а – NOT; б – NAND2; в – NOR2

Связь параметров быстродействия с шириной каналов простейших КМОП-вентилей приведена в табл.3.2

#### Параметры быстродействия различных вентилях

Таблица 3.2

Параметр	Согласованные			Гармонические			Минимальные		
	N O T	N A N D 2	N O R 2	N O T	N A N D 2	N O R 2	N O T	N A N D 2	N O R 2
$W_p$	2	2	4	1.41	2	2	1	1	2
$W_n$	1	2	1	1	2	1	1	2	1
$p$	1	2	2	0.8	2	4/3	2/3	4/3	4/3
$g_u$	1	4/3	5/3	1.15	4/3	2	4/3	2	2
$g_d$	1	4/3	5/3	0.81	4/3	1	2/3	1	1
$g_{avg}$	1	4/3	5/3	0.98	4/3	3/2	1	3/2	3/2

Примечание:  $g_u$  – логическая нагрузка единицы;  $g_d$  – логическая нагрузка нуля;  $g_{avg}$  – средняя логическая нагрузка.

У несогласованных вентилях выходное сопротивление, а значит, и логическая нагрузка зависит от выходного сигнала.

Паразитная задержка и логическая нагрузка увеличиваются с возрастанием сложности вентилях.

## Глава 4. Комбинационные логические схемы

Комбинационная логическая схема преобразует входные двоичные сигналы в выходные двоичные сигналы в соответствии с логической функцией. В стационарном состоянии при установившихся входных напряжениях (после окончания переходных процессов) выходные напряжения определены однозначно и не зависят от предыстории.

В отличие от вентиля логическая схема имеет несколько выходов и, как правило, не сводится к набору независимых вентиляей. Как и многие вентили, логическая схема допускает различные варианты реализации и методы синтеза, в том числе на уровнях:

- транзисторном;
- вентильном;
- блочном (из более простых логических схем).

При проектировании логических схем в качестве критериев оптимизации наиболее часто выступают:

- быстродействие;
- потребляемая мощность;
- площадь, занимаемая схемой на кристалле.

При проектировании КМОП-схем необходимо учитывать определенные особенности, в том числе:

- 1) возможность синтезировать сложные функции сопровождается ограничением числа ярусов и снижением быстродействия;
- 2) каждое инвертирование входных сигналов требует отдельного вентиля (инвертора), поэтому схемы типа NAND, NOR реализуются проще, чем схема XOR;
- 3) КМОП-схемы не допускают объединения по выходу, выполнения проводных функций.

Достоинствами комбинационных КМОП-схем (и вентиляей) являются:

- 1) отсутствие потребления в статике;
- 2) полный размах выходного напряжения;
- 3) связь выхода через открытый ключ с питанием;
- 4) высокий запас помехоустойчивости;
- 5) простота проектирования.

Комбинационные КМОП-схемы имеют и недостатки, часть которых следует из достоинств:

- 1) удвоение количества входов (n-МОП и p-МОП);
- 2) различие в параметрах транзисторов (p-МОП «слабее» n-МОП);
- 3) наличие сквозных токов при переключении;
- 4) зависимость задержки от входов и предыстории.

#### 4.1. Типовые комбинационные КМОП-схемы

**Драйверы.** В качестве драйверов (магистральных усилителей) можно использовать обычные КМОП-вентили и инверторы с большим аспектным отношением транзисторов.

В ряде случаев целесообразно объединять драйверы (и просто вентили) по выходу. В стандартных КМОП-вентилях при таком объединении могут постоянно протекать сквозные токи. Для исключения сквозных токов необходимо использовать специальные КМОП-схемы с третьим состоянием (тактируемые КМОП) (рис.4.1). В таких схемах тактовый импульс может быть введен непосредственно в блоки вентиля (рис.4.1,а), а может коммутировать входной сигнал с помощью других вентилях (рис.4.1,в).

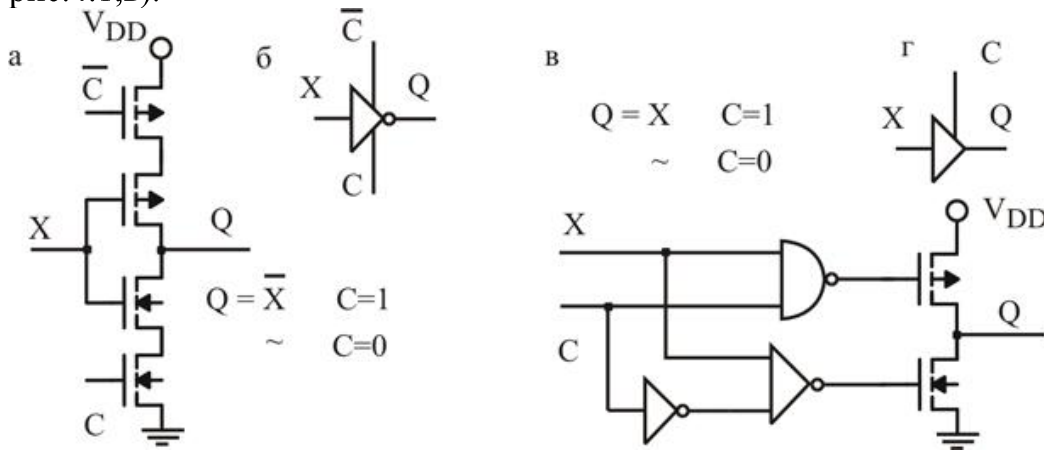


Рис.4.1. КМОП-вентили с третьим состоянием: а – с парафазным тактированием; б – его условное обозначение; в – с однофазным тактированием; г – его условное обозначение.

В обоих случаях при отсутствии сигнала управления ( $C = 0$ ) выход отсоединен от источника питания. Схема сохраняет на выходе предыдущее состояние в течение длительного времени, определяемого токами утечки. При объединении таких схем по выходу и подаче управления только на один из объединенных вентилях на общем выходе будет проинвертированный (рис.4.1,б) или прямой (рис.4.1,г) вход выбранного вентиля.

Подобные схемы используют в мультиплексорах, выходных каскадах ИС и др.

**Шифратор** преобразует код, содержащий только одну единицу, в двоичный код. Он сокращает количество сигнальных проводников по логарифмическому закону. При  $m$  входах шифратора у него не более  $M = \log_2(m)$  выходов. Поскольку в исходном коде присутствует ограниченное

количество комбинаций, то построение КМОП-шифратора упрощается. В его схемах блоки  $p$  могут не быть комплементарны блокам  $n$ . В качестве

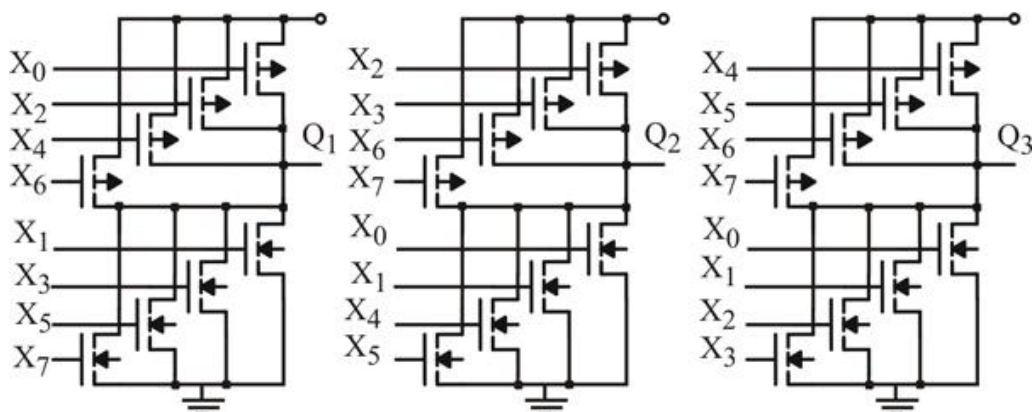


Рис.4.2. Восьмивходовой КМОП шифратор.

примера приведем восьмивходовой КМОП-шифратор (рис.4.2).

**Дешифратор** выполняет обратную операцию. Для каждой входной комбинации  $M$  разрядного числа он возбуждает (переводит в состояние единицы) одну выходную шину из  $m = 2^M$  возможных, рис.4.3.

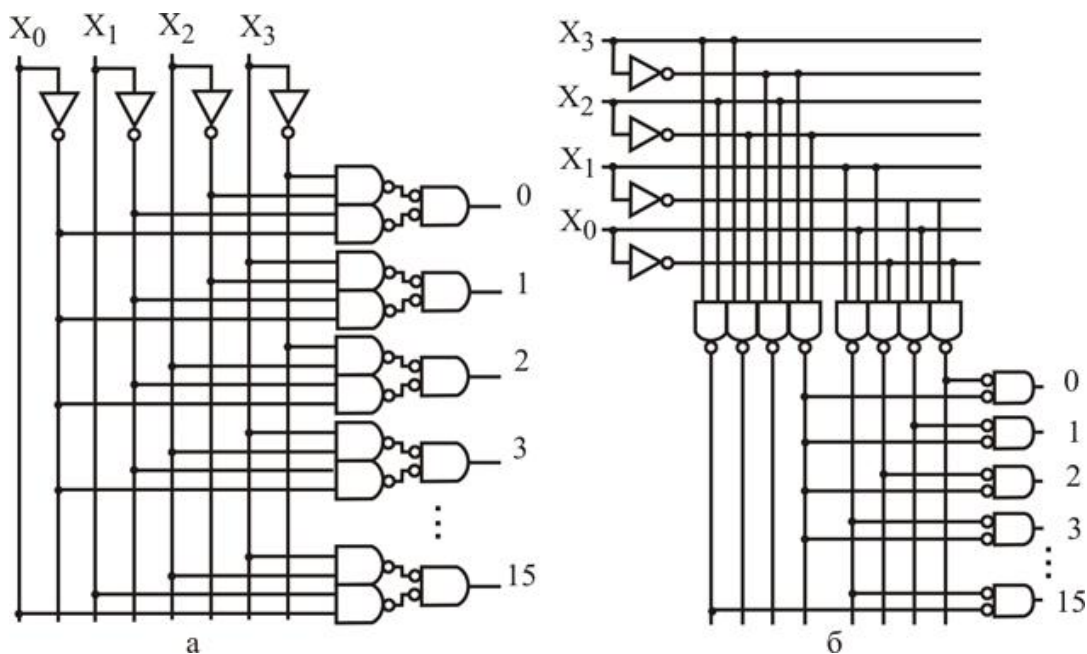


Рис.4.3. Дешифраторы: а – одноступенчатый; б – двухступенчатый

Формально каждая выходная шина является выходом  $M$ -входовой схемы И. Таких вентилях в составе дешифратора  $m = 2^M$ . При малых  $M$  подобная линейная одноступенчатая конструкция дешифратора оправда-

на. С возрастанием  $M$  аппаратные затраты растут экспоненциально, что приводит к необходимости использовать многоступенчатые дешифраторы: матричные (прямоугольные), пирамидальные, комбинированные.

На рис.4.3 приведены модификации дешифратора с  $M = 4$ .

На входах обоих дешифраторов дополнительно формируются проинвертированные сигналы.

В одноступенчатом дешифраторе (рис.4.3,а) все 16 возможных кодовых комбинаций поступает на четырехвходовые схемы И(NAND4). Схемы NAND4 построены на NAND2 и NOR2.

В двухступенчатом (матричном) дешифраторе (рис.4.3,б), введены два предварительных дешифратора на два входа. При этом у каждого из них возбужден только один выход. Выходы предварительных дешифраторов подаются на вторую ступень из схем NOR2.

**Мультиплексор (MUX)** преобразует параллельный код в последовательный (во времени) (рис.4.4). На его входы подаются сигналы  $X_1 - X_4$  и импульсы управления  $F_1 - F_4$ , следующие один за другим без перекрытия. Выход схемы  $Q$  имеет вид

$$Q = \overline{X_1 F_1} \overline{X_2 F_2} \overline{X_3 F_3} \overline{X_4 F_4} = X_1 F_1 + X_2 F_2 + X_3 F_3 + X_4 F_4.$$

На выход  $Q$  сигналы поступают последовательно со сдвигом во времени.

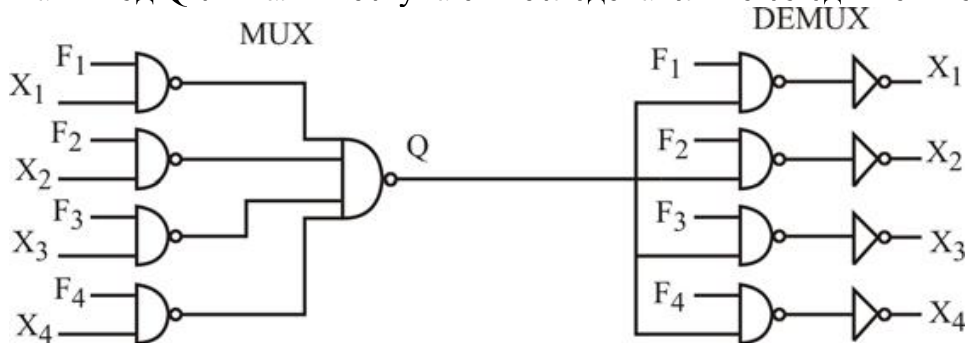


Рис.4.4. Мультиплексор – демультимплексор.

**Демультимплексор (DEMUX)** выполняет операцию обратную мультиплексированию – преобразует последовательный код в параллельный (см. рис.4.4). Входная последовательность на его входах преобразуется в выходные сигналы. Комбинационная схема демультимплексора сохраняет сдвиг сигналов во времени, введенный мультиплексором. Для их совмещения во времени требуются схемы с памятью.

## 4.2. Сумматор.

**Одноразрядный сумматор (SUM)**– одна из широко распространенных логических схем, часто являющаяся библиотечным элементом. Его логическая функция

$$S = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C} + ABC, \quad (4.1)$$

$$C_0 = A\overline{B}\overline{C} + \overline{A}B\overline{C} + \overline{A}\overline{B}C + ABC, \quad (4.2)$$

где  $S$  – выход суммы;  $C_0$  – выход переноса;  $A$  – первое слагаемое;  $B$  – второе слагаемое;  $C$  – входной перенос.

На основе этих формул можно построить одноразрядный сумматор в виде двух независимых вентилях, реализующих функции (4.1) и (4.2). Однако это будут неоптимальные схемы, так как в их составе присутствуют четырехъярусные вентили. Общее количество транзисторов в них равно  $3 \times 4 \times 2 \times 2 = 48$ .

Известны более эффективные модификации одноразрядных сумматоров (рис.4.5). Часто такие сумматоры строят на основе полусумматоров (SSUM), реализующих функции  $Q = \text{XOR}(A,B)$  и  $C = AB$  (рис.4.5,а).

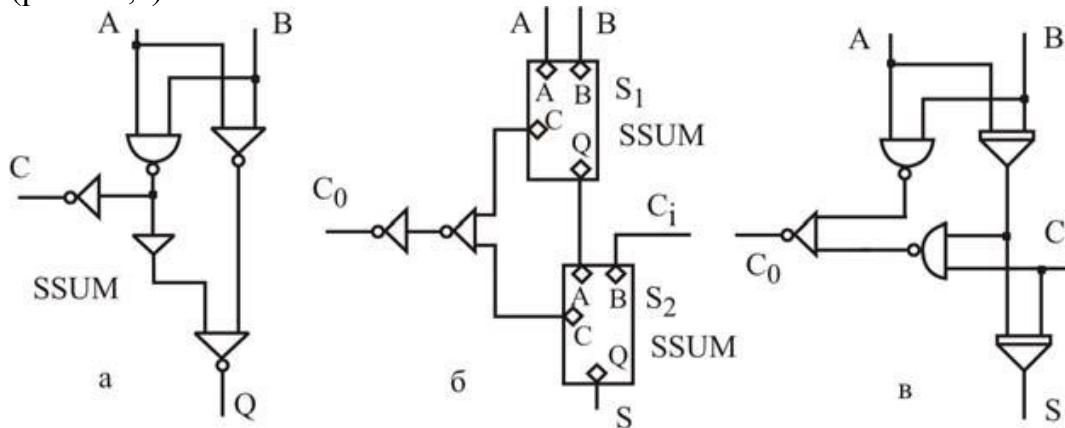


Рис.4.5. Сумматоры: а – полусумматор; б – одноразрядный сумматор на основе полусумматоров; в – одноразрядный сумматор на основе XOR.

Соединение двух полусумматоров и схемы ИЛИ (рис.4.5,б) дает одноразрядный сумматор. Фактически, в полусумматорах реализована функция XOR и даже с меньшими затратами, чем в библиотечном элементе, но она несимметрична. Кроме того, в этой модификации сумматора от входов  $A$  и  $B$  до выхода переноса  $C_0$  четыре элемента, а в схеме рис.4.5,в – таких элементов два. Поэтому, несмотря на большие затраты (44 транзистора вместо 32), схема на симметричных XOR (рис.4.5,в) более подходит для построения многоразрядных сумматоров, где задержка цепи переноса является определяющей.

Исходные выражения (4.1) и (4.2) можно преобразовать к виду

$$C_0 = A\overline{B}\overline{C} + \overline{A}B\overline{C} + \overline{A}\overline{B}C + ABC = AB + BC + AC = (A+B)(B+C)(A+C) \quad (4.3)$$



$$S = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C} + ABC =$$

$$(A + B + C)\overline{C_0} + ABC \quad (4.4)$$

Полученные выражения с точностью до инвертирования достаточно просто реализуются двумя взаимосвязанными КМОП-вентиллями. Причем в силу симметрии блок р и блок п не являются комплементарными, но сохраняют главное свойство КМОП – отсутствие тока в статике. Полный сумматор такого типа содержит всего 28 транзисторов (рис.4.6).

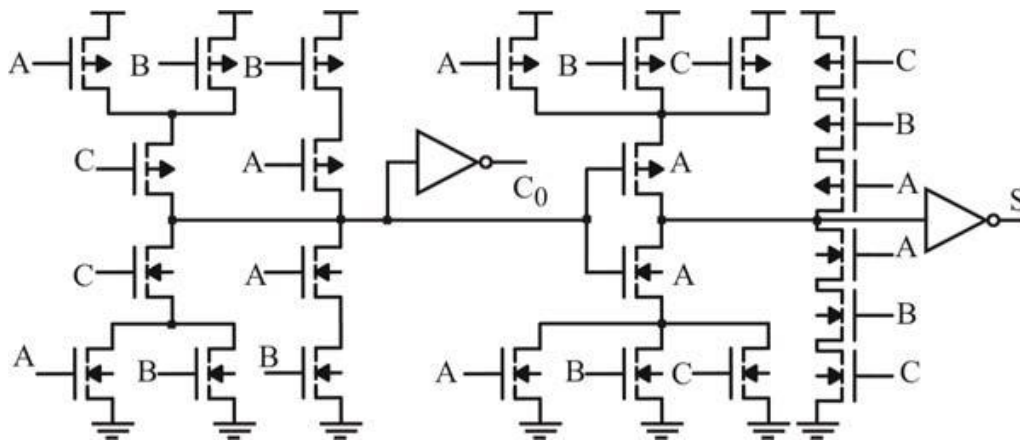


Рис.4.6 Комбинационный одноразрядный сумматор

**Параллельный сумматор** .Соединением нескольких одноразрядных сумматоров получают параллельный сумматор, в котором выход переноса  $C_0$  соединен со входом переноса следующего разряда  $C_i$ . При подаче на входы  $A$  и  $B$  многоразрядных чисел в параллельном коде результат на выходе  $S$  устанавливается после прохождения сигнала переноса. Максимально возможная длина переноса равна разрядности суммируемых чисел. Время переноса является определяющим. Поэтому важно минимизировать количество вентилях в цепи переноса. У выше рассмотренных одноразрядных сумматоров при распространении переноса он в каждом разряде проходит через два вентиля.

Преобразованием логической функции сумматора можно упростить цепь переноса. Сама функция суммы остается без изменений

$$S = \text{XOR}(C_i, P) \quad (4.5)$$

$$P = \text{XOR}(A, B) \quad (4.5,a)$$

В ранее рассмотренной модификации сумматора перенос выступал равноправно со слагаемыми. Однако в связи с тем, что именно он определяет быстродействие в параллельном сумматоре, его целесообразно выделить особо. Вводятся функция генерации переноса  $G = A \cdot B$  и функ-



ция разрешения распространения (прохождения) переноса, совпадающая с поразрядной суммой слагаемых  $P = \text{XOR}(A, B)$ . Выход переноса  $C_0 = C_i P + G$  и поскольку  $P$  и  $G$  взаимоисключающие, то дизъюнкция может быть проводной (соединением выходов). В результате схема параллельного сумматора имеет вид, показанный на рис.4.7.

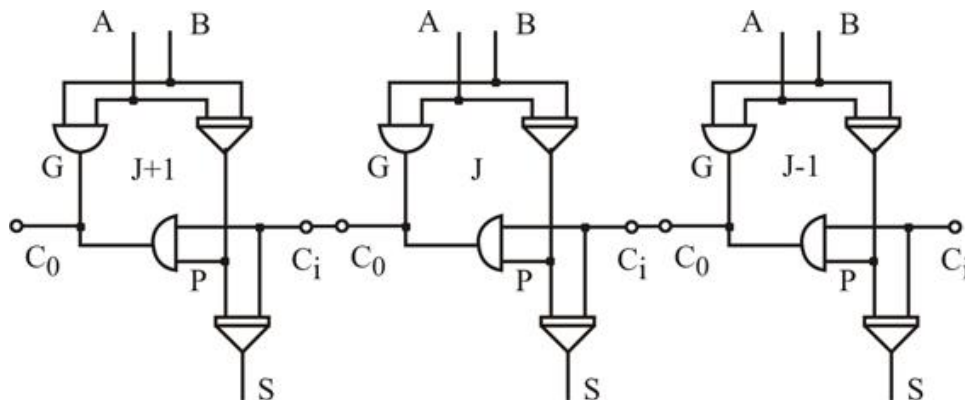


Рис.4.7 Параллельный сумматор с уменьшенной задержкой

Отметим, что в составе сумматора, в том числе в составе цепи переноса, появились вентили AND без инвертирования. Небольшим преобразованием можно перейти к инвертирующим вентилям, но исключить вентиль с двумя входами в цепи переноса невозможно, что существенно увеличивает общее время задержки сигнала переноса.

В многоразрядных параллельных сумматорах широко применяют логические методы ускорения переноса, за счет минимизации количества проходимых сигналом переноса вентилях. Один из методов – групповой перенос – предусматривает специальные цепи ускоренного переноса (рис.4.8). При совпадении разрешения переноса у нескольких соседних сумматоров сигнал переноса может обойти их по цепи ускоренного переноса.

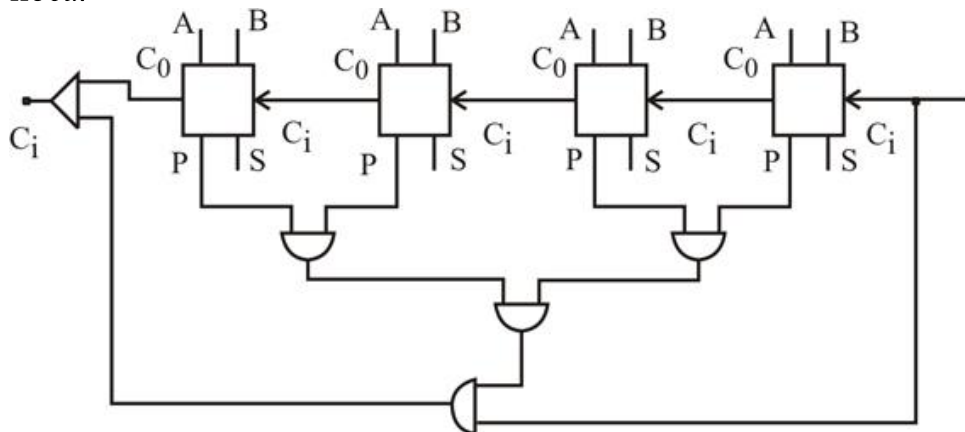


Рис.4.8. Параллельный сумматор с ускоренным переносом

В этом случае комбинационные вентили также не оптимальны, поскольку требуются двухвходовые вентили.

### 4.3. Модификации комбинационных логических схем

Один из недостатков комбинационных логических схем обусловлен «слабостью» р-МОП по сравнению с n-МОП. Поэтому понятно стремление максимально уменьшить использование р-МОП, по крайней мере, в ответственных местах схем.

Другой недостаток комбинационных логических схем заключается в невозможности инвертировать сигнал на входе вентиля. Во многих случаях наряду с прямым сигналом целесообразно иметь и инвертированный сигнал. Такие возможности предоставляют дифференциальные логические схемы, где каждый вентиль имеет прямой и инвертированный выход. Однако простое удвоение сигналов в комбинационных КМОП-вентилеях нецелесообразно, поскольку они уже содержат удвоенное количество транзисторов. Для решения проблемы предложены каскодные переключательные логические схемы.

**Каскодные переключательные логические схемы (CVSL)** или дифференциальные каскодные переключательные логические схемы представлены на рис.4.9. В составе вентиля содержится два блока n-МОП, с прямым и инверсным выходами, а его блоки р-МОП содержат по одному р-МОП, затворы которых подсоединены к выходам противоположных блоков n-МОП (рис.4.9,а).

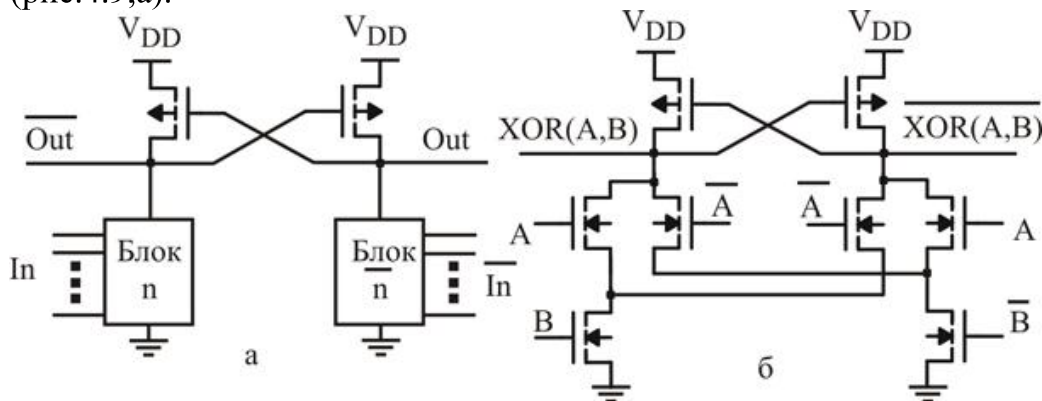


Рис.4.9. CVSL вентили: а – структура вентиля; б – XOR вентиль

В результате основное свойство КМОП – отсутствие сквозных токов сохраняется, но появляется гистерезис в передаточной характеристике. Подобная структура ускоряет переходные процессы при переключении вентиля и в некоторых случаях расширяет их функциональные возможности.

сти. Так XOR и NXOR вентиль (рис.4.9,б), в совокупности содержат всего 8 МОП.

**Тактируемые комбинационные логические схемы.** Примером тактируемого инвертора является драйвер (см. рис.4.1,а). По такому же принципу можно строить и более сложные логические схемы, рис.4.10.

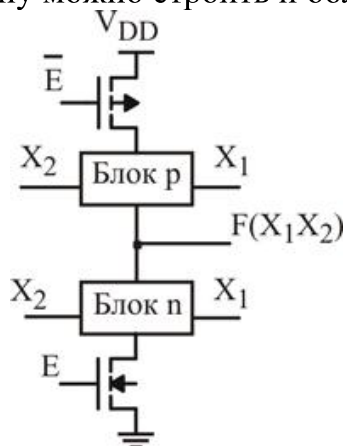


Рис.4.10. Тактируемый вентиль с двумя входами

Он находится в активном состоянии, пока на разрешающем входе Е высокий уровень. При изменении уровня разрешающего сигнала выходной потенциал некоторое время сохраняется. Такие схемы отличаются от классических комбинационных схем наличием третьего состояния, возможностью объединения по выходу. В некотором смысле они аналогичны проходным и динамическим логическими схемами.

## Глава 5. Прходные логические схемы

В проходных логических схемах применяются МОП- и КМОП-ключи. Использование ключей одного типа – однополярных ключей (проходных вентилях) позволяет исключить р-МОП, однако возникает проблема полной передачи единичного сигнала. В комплементарных КМОП-ключках (передаточных вентилях) используются n-МОП и р-МОП, но сохраняется проблема передачи сигнала по длинным цепочкам ключей.

### 5.1. Однополярные проходные логические схемы

Прходные логические схемы, использующие однополярные ключи, обладают важными преимуществами. Во-первых, в элементе всего один транзистор вместо двух в инверторе, во-вторых, у ключа три вывода вместо четырех у инвертора. Поэтому можно добиться более высокой плотности упаковки.

**Прходные вентили.** В схемах с однополярными n-МОП ключами (рис.5.1), как и КМОП-вентильях, исключаются сквозные токи. Если к некоторому узлу подключены несколько ключей, то допускается открывание только одного из них. При двух ключах, (рис.5.1,б) и (рис.5.1,в) они управляются парафазно.

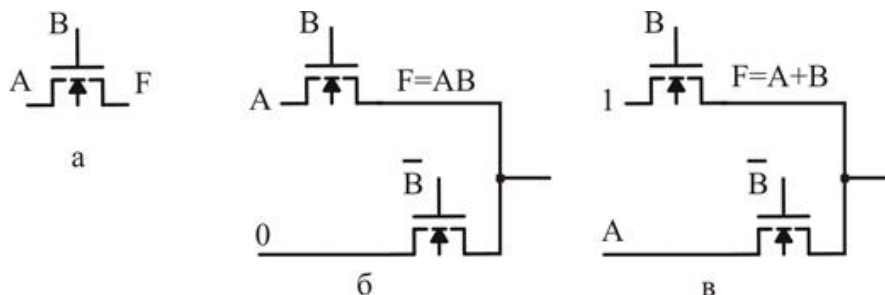


Рис.5.1. Прходные вентили: а – ключ; б – схема И; в – схема ИЛИ

Простейший ключ на одном n-МОП (рис.5.1,а) соединяет выход F со входом только при высоком уровне сигнала управления  $B = 1$ , а при  $B = 0$  потенциал выхода сохраняет состояние, т.е. обладает памятью.

Два n-МОП ключа с парафазным управлением позволяют реализовать схемы И и ИЛИ. В этом случае выход подсоединен ко входу при любом значении сигнала управления В, а логическая функция элемента зависит от сигналов на коммутируемых входах. Так, подавая на ключ с инверсным управлением нуль имеем функцию И (рис.5.1,б), а подавая на прямой вход единицу имеем функцию ИЛИ (рис.5.1,в).

Существенным отличием ключевых логических элементов от вентиля является отсутствие в них инвертирования. Иногда такое свойство упрощает построение схем. При необходимости инвертирование достигается введением дополнительных элементов.

**Дифференциальные проходные логические схемы.** Дифференциальные проходные однополярные вентили содержат прямые и инверсные входы и выходы (рис.5.2).

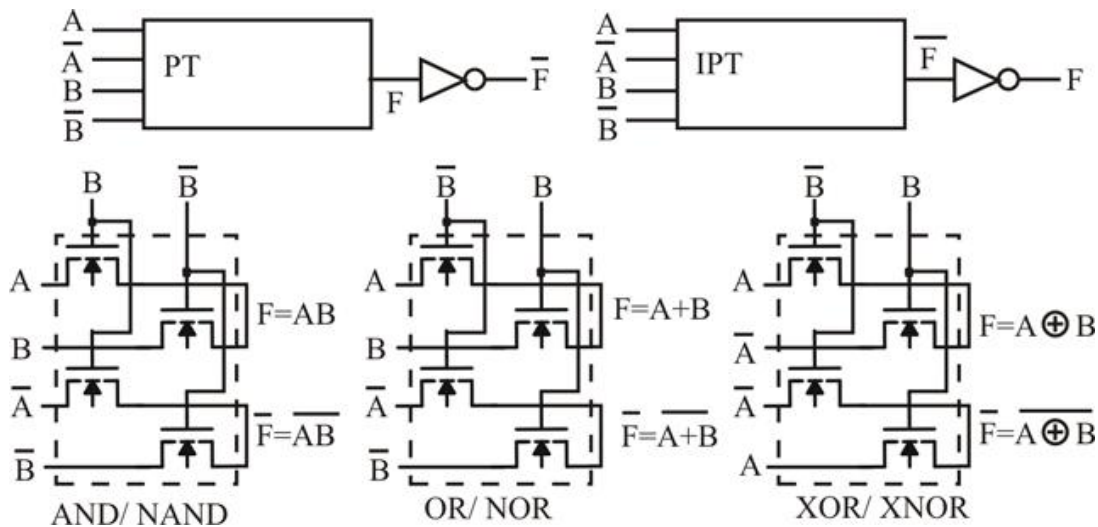


Рис.5.2. Дифференциальные проходные вентили.

С помощью четырех однополярных n-МОП ключей удастся реализовать любую прямую и инверсную симметричную логическую функцию двух переменных. Это позволяет сократить количество вентилях в составе одноразрядного сумматора.

**Одноразрядный однополярный сумматор.** Использование приведенных проходных вентилях (см. рис.5.2) определяет вид одноразрядного сумматора. Функция суммы получается из двух схем XOR

$$S = (A \oplus B) \oplus C, \quad (5.1)$$

а функция переноса из схем AND, OR и XOR

$$C_0 = AB + (A \oplus B)C \quad (5.2)$$

Вариант реализации сумматора приведен на рис.5.3. Он содержит 28 МОП.

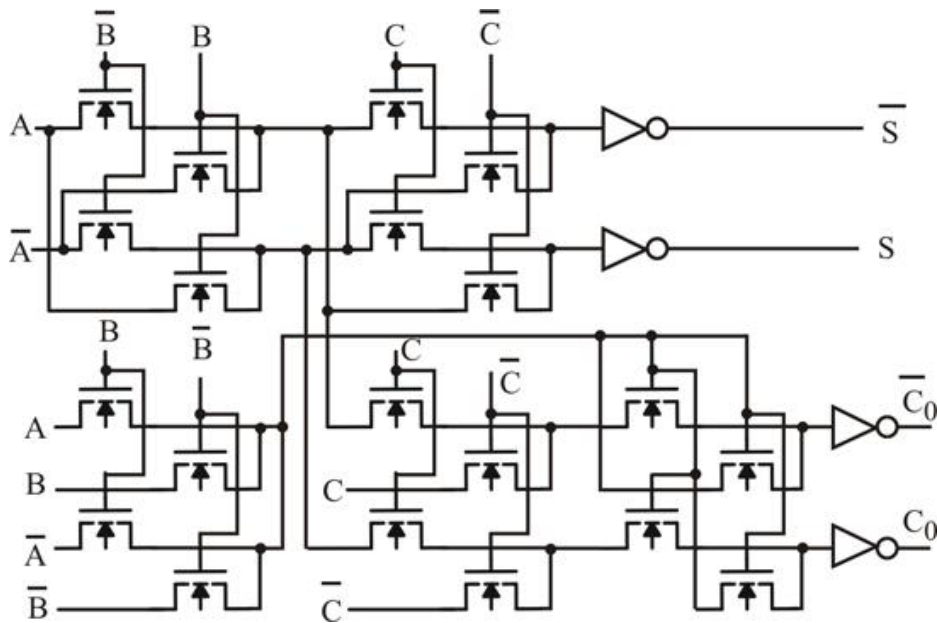


Рис.5.3. Одноразрядный сумматор на однополярных проходных элементах

**Особенности однополярных проходных логических схем.** Основным ограничением однополярных проходных вентиляей является снижение выходного перепада на выходе ключа. При использовании n-МОП ключей высокий уровень в цепочке снижается после каждого элемента на величину порогового напряжения (рис.5.4).

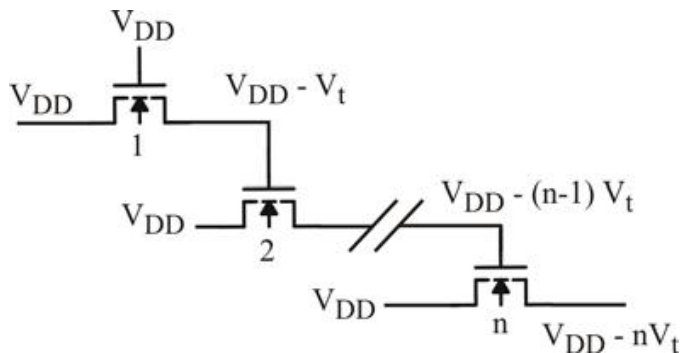


Рис.5.4 Уменьшение высокого уровня в цепочке n-МОП ключей.

**Проходные полнодиапазонные вентили.** Недостаток проходных вентиляей – неполная передача высокого уровня – может быть устранен. Для этого выход проходного вентиля подсоединяют к высокому потенциалу питания  $V_{DD}$  через p-МОП, управляемый проинвертированным сигналом самого проходного вентиля (рис.5.5,а).

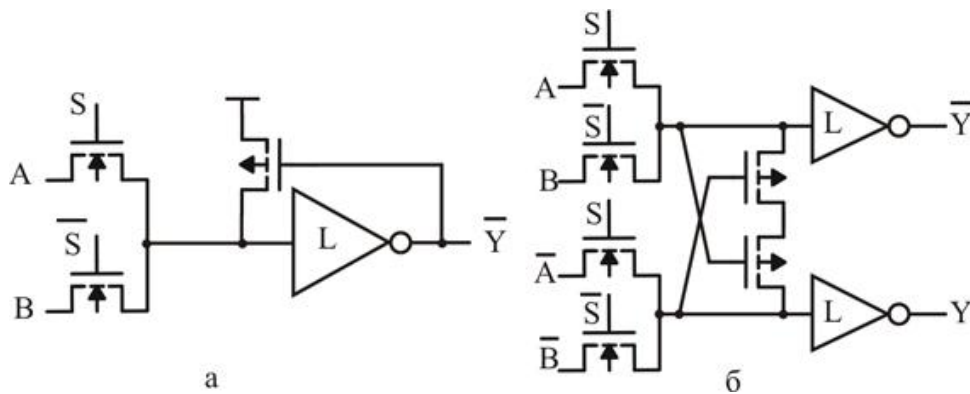


Рис.5.5. Восстановление высокого уровня в проходном (а) вентиле и дифференциальном проходном (б) вентилях.

В случае использования дифференциальных однополярных проходных вентилях, управление р-МОП осуществляется с инверсного выхода (рис.5.5,б).

## 5.2. Комплементарные проходные логические схемы

В комплементарной проходных вентилях используются КМОП-ключи, что позволяет исключить потерю перепада сигналов.

**Свойства КМОП-ключа.** КМОП-ключи состоят из двух транзисторов n-МОП и р-МОП для обеспечения полной передачи напряжения от входа A до выхода B (рис.5.6,а). При низком напряжении на управляющем входе C оба транзистора закрыты. При высоком управляющем сигнале и n-МОП и р-МОП открыты, но в зависимости от напряжения на входе они работают в разных режимах: один – как истоковый повторитель, другой – как усилитель с общим истоком. Так при  $A = 0$  (рис.5.6,б) в зависимости тока втекающего в ключ из узла B, от напряжения на разряжающейся емкости  $C_L$ , (кривая 1) дают вклад ток n-МОП (кривая 2) и ток р-МОП (кривая 3). Когда потенциал на выходе B высокий, через оба транзистора текут примерно равные токи, но когда потенциал на выходе приближается к нулевому, ток течет в основном через n-МОП. При  $A = 1$  ситуация прямо противоположная – при потенциале выхода близком к единице, ток протекает в основном через р-МОП.



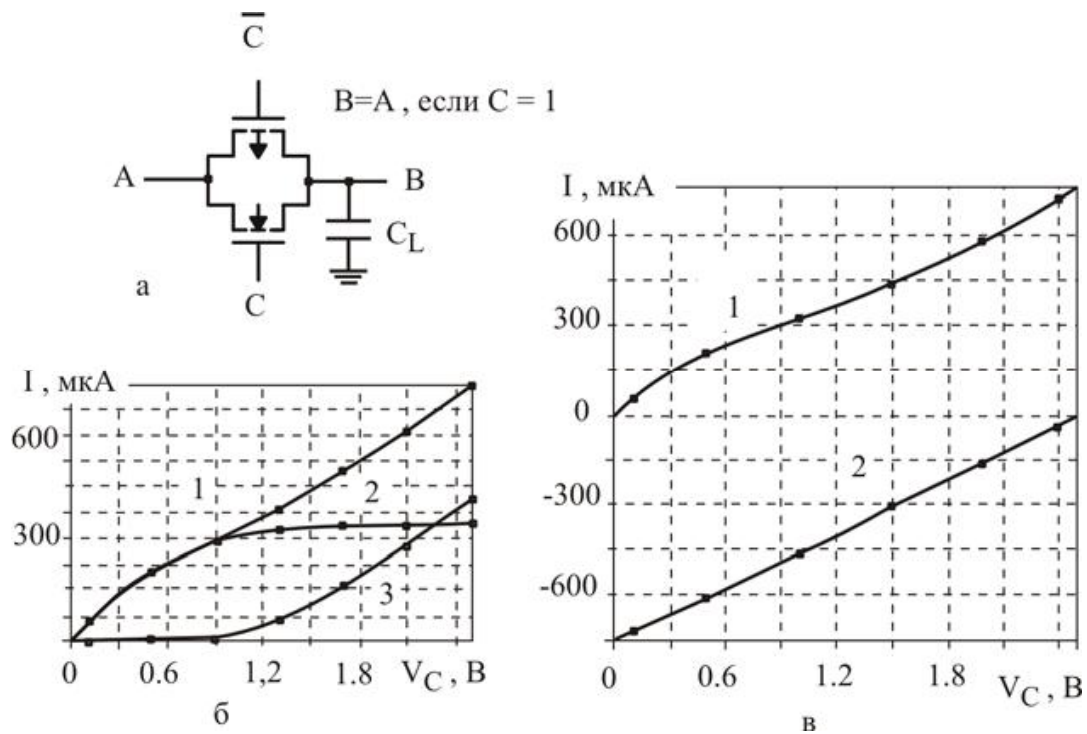


Рис.5.6. КМОП-ключ: а – электрическая схема; б – проходная ВАХ и ее составляющие при  $A = 0$ ; в – проходная ВАХ при  $A = 0$  (кривая 1) и при  $A = 1$  (кривая 2).

Выбором размеров (ширины) каналов можно выравнять токи через открытый КМОП-ключ, заряжающие и разряжающие емкость  $C_L$  выходного узла  $B$  (рис 5.6,в). В этом случае сопротивление ключа в первом приближении равно  $R_S \approx 1/(\beta V_{DSat})$ .

**Вентили на КМОП-ключах.** При построении схем, где не требуется инвертирование (мультиплексоров, демультиплексоров и т.п.) целесообразно использовать вентили на КМОП-ключах в силу их экономичности.

В мультиплексорах в каждом канале достаточно одного КМОП-ключа (рис.5.7,а).

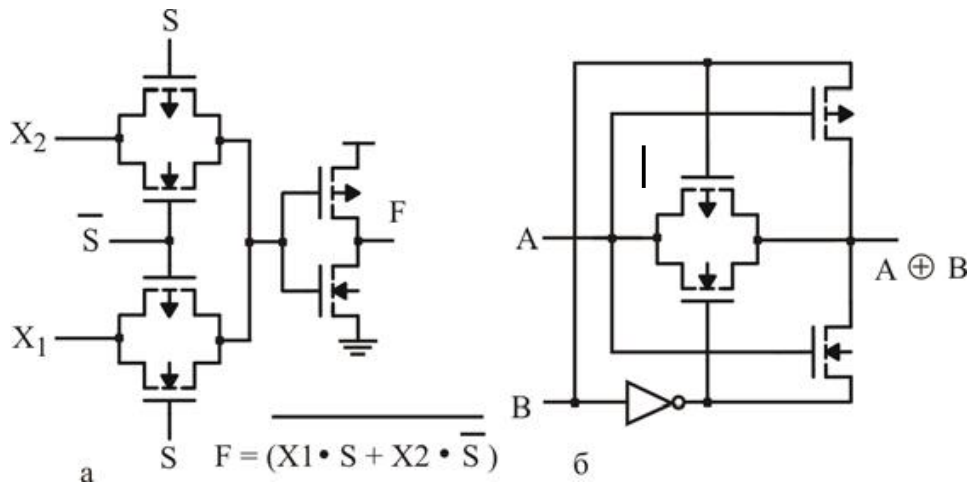


Рис.5.7. Логические схемы на КМОП-ключках: а – MUX; б – XOR

Операция ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR) также выполняется очень экономно (рис.5.7,б). Поскольку КМОП-ключи передают сигнал без усиления, то совместно с ними используется КМОП-усилитель (инвертор), необходимый и для формирования управления р-МОП, и для инвертирования сигнала в ключевых схемах.

**Одноразрядный сумматор.** Выигрыш в аппаратуре при использовании КМОП-ключей наиболее очевиден в схеме одноразрядного сумматора (рис.5.8). Для него достаточно 14 транзисторов (а с учетом выходных вентилей – 18 транзисторов).

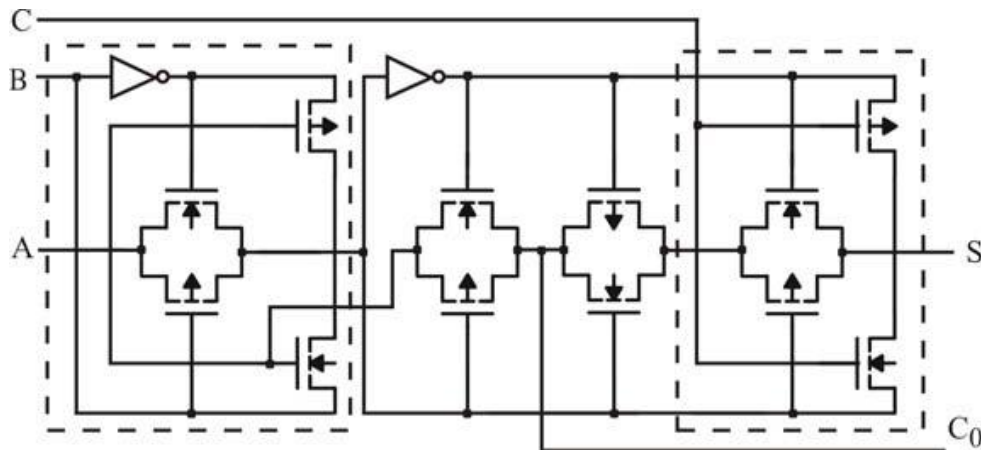


Рис.5.8. Одноразрядный сумматор на комплементарных ключках.

**Дифференциальные логические схемы на КМОП-ключках.** Как и однополярные проходные логические схемы дифференциальные схем с КМОП-ключками позволяет достаточно экономно выполнять основные логические функции (рис.5.9).

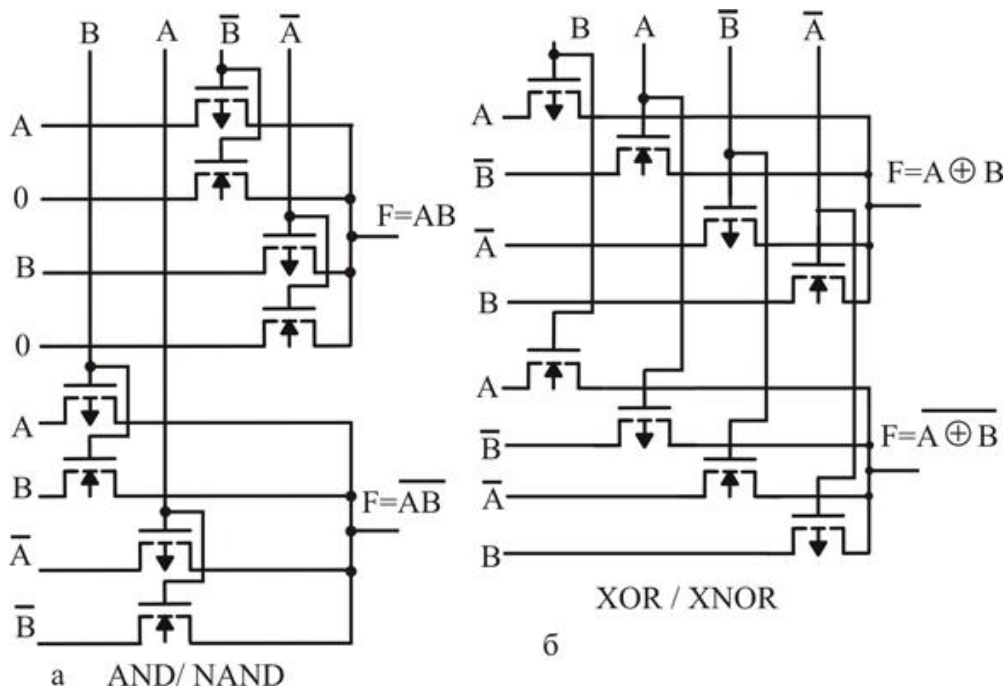


Рис.5.9. Симметричные вентили на КМОП-ключках: а – AND/NAND; б – XOR/XNOR

**Особенности проходных логических схем.** Простота однополярных и комплементарных проходных логических схем связана с отсутствием усиления. Сигнал, проходящий по цепочке вентилей (рис.5.10,а), должен заряжать емкости узлов через возрастающее сопротивление (рис.5.10,б).

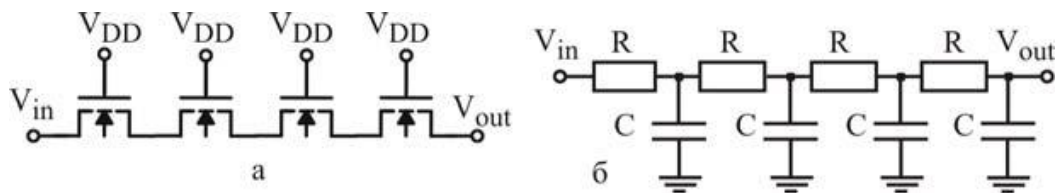


Рис.5.10. Цепь ключей: а – электрическая схема; б – эквивалентная схема

Поэтому задержка квадратично зависит от количества ключей  $N$  в цепочке  $t_0 \approx N^2 RC$ .

## Глава 6. Динамические логические схемы

Динамические логические схемы в отличие от статических используют представление информации в виде зарядов на емкостях. В статических схемах выходные сигналы вентиля сохраняются, пока не изменятся входные сигналы, поскольку они связаны однозначно. В динамических логических схемах выходные сигналы вентиля сохраняются только в течение некоторого времени – такта работы. Они связаны с входными сигналами только в определенные интервалы времени. При необходимости длительного сохранения требуется их восстановление – регенерация.

При использовании динамических логических схем необходимы тактовые сигналы, которые обеспечивают начальную установку (предустановку) потенциалов узлов, а в некоторых случаях и передачу сигналов по цепочке вентиля.

Динамические логические схемы обладают рядом достоинств:

- 1) сочетают логические функции с памятью;
- 2) используют общие тактовые сигналы для синхронизации различных блоков;
- 3) требуют меньше элементов при выполнении сложных функций;
- 4) зачастую рассеивают меньшую мощность из-за малых паразитных емкостей.

### 6.1. Динамические логические элементы

В динамических логических элементах тактовые импульсы подаются на специальные МОП-транзисторы предзаряда и чтения, а входные сигналы – на транзисторы логического блока. В отличие от комбинационных КМОП-элементов, содержащих два комплементарных логических блока, динамические элементы имеют только один логический блок.

**Динамический инвертор.** Электрическая схема динамического КМОП-инвертора и временная диаграмма его работы приведены на рис.6.1.

В режиме предзаряда низкий уровень тактового сигнала  $\phi$  открывает р-МОП предзаряда и выходной узел схемы  $Y$  заряжается до высокого уровня напряжения питания. Остальная часть схемы не мешает этому, поскольку закрыт нижний n-МОП-чтения. При повышении напряжения тактового сигнала во время перехода к рабочему режиму (режиму чтения) р-МОП предзаряда закрывается, а n-МОП чтения открывается.

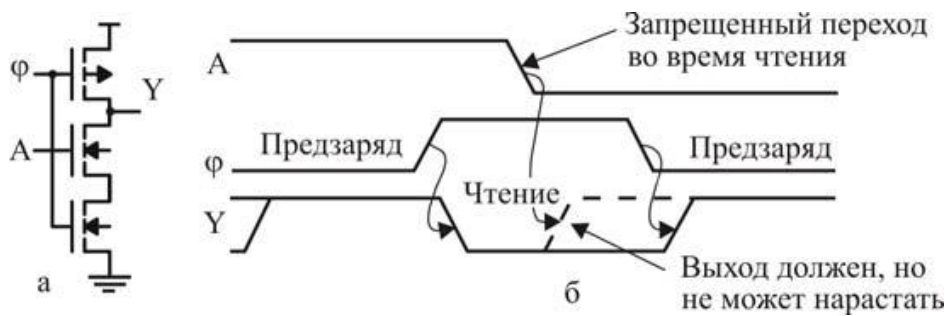


Рис.6.1. Динамический КМОП-инвертор; а – электрическая схема; б – временная диаграмма работы.

Напряжение на выходе схемы определяется входом А. Если на сигнальном входе низкий потенциал, то высокий уровень на выходе сохраняется. Если на сигнальном входе высокий потенциал, то оба n-МОП открыты, и напряжение на выходе снижается до нуля. Переход входного потенциала с низкого уровня на высокий может происходить в любой интервал времени режима чтения, и выход проинвертирует такой сигнал. Но если во время режима чтения входной сигнал с высокого уровня перейдет на низкий уровень, то выходной сигнал останется нулевым. Такой переход в режиме чтения запрещен, так как он нарушает логику работы инвертора.

Выходной потенциал динамического инвертора в режиме чтения может изменить свое состояние и перейти с высокого уровня на низкий. Поскольку такой переход является запрещенным для входов инверторов, то схема не может непосредственно в этом такте чтения работать на аналогичную схему.

Таким образом, динамический КМОП-инвертор обладает рядом особенностей, отличающих его от статического (комбинационного) инвертора:

- 1) имеет два входа: сигнальный и тактовый;
- 2) функционирует в двух режимах: предзаряда и чтения (рабочий);
- 3) информационный сигнал на выходе присутствует только в режиме чтения;
- 4) в режиме предзаряда имеет на выходе высокий потенциал;
- 5) в режиме чтения не допускается изменение входного сигнала с высокого уровня на низкий: схема не реагирует на это изменение, т.е. работает с ошибкой;
- 6) не допускает работы на аналогичную схему в такте чтения.

**Динамические вентили.** Динамические вентили строятся по аналогии с инверторами. В их составе два МОП, подсоединенные к тактовому входу  $\Phi$ : верхний р-МОП предзаряда и нижний n-МОП чтения (рис.6.2,а). Во время предзаряда р-МОП подсоединяет выход вентили к

источнику питания, а n-МОП выключен (рис.6.3). В такте чтения состояния р-МОП и n-МОП меняются, а выходной сигнал определяется комбинацией входных сигналов, подаваемых на блок.

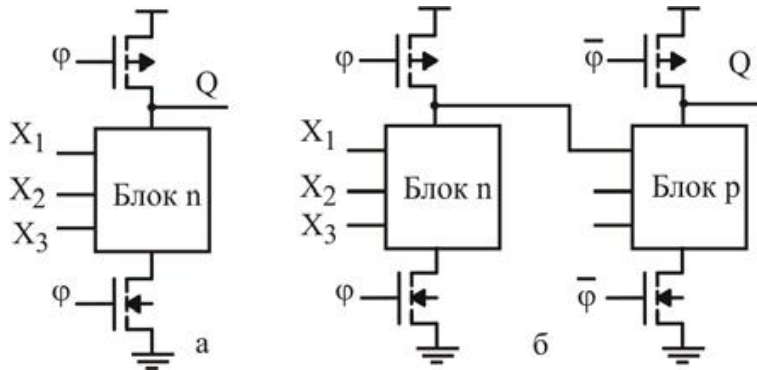


Рис.6.2. Динамические КМОП-вентили: а – n-КМОП, б – n-p-КМОП

Если все цепи блока n в режиме чтения закрыты, то на выходе остается высокое напряжение. Если хотя бы одна цепь блока n соединяет узел выхода с нулевым потенциалом, то выходной узел разряжается, и на выходе устанавливается низкий уровень.

В динамических вентилях наряду с блоком n может использоваться и блок p. Если блоки n и p используют разнополярные тактовые импульсы, то они могут взаимно работать друг на друга в одном такте чтения (рис.6.2,б).

Основное достоинство динамических КМОП-вентилей – сокращение количества транзисторов. Если в статическом вентиле с N различными входами содержится  $2N$  транзисторов, то в динамическом только  $N+2$  транзистора. Уменьшение количества транзисторов ведет к снижению входной емкости и площади, занимаемой вентиляем.

Потенциал выходного узла X динамической схемы в режиме чтения разряжается не только под действием входных сигналов, но и за счет токов утечки (рис.6.3,а). Для предотвращения такой разрядки иногда используют специальный компенсирующий р-МОП, сохраняющий высокий выходной потенциал (рис.6.3,б). Компенсирующий р-МОП управляется проинвертированным выходным сигналом. Так как ток утечки невелик, то требования к дополнительному р-МОП невысоки, и он может иметь минимально возможный размер.

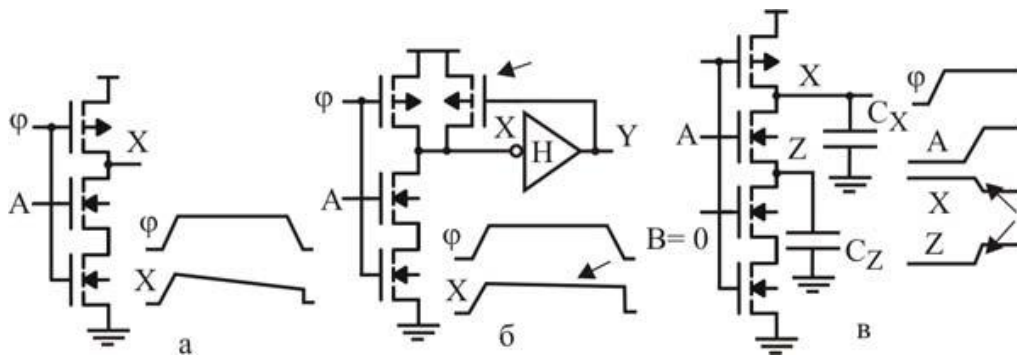


Рис.6.3 Изменение выходного потенциала в динамических КМОП-вентилей: а – за счет утечки ; б – компенсация изменения; в – за счет перераспределения заряда

Выходной сигнал динамической схемы подвержен влиянию помех за счет перераспределения заряда (рис 6.4,в). Если в режиме чтения ближайший к выходному узлу n-МОП открывается, а в целом цепь n-МОП остается закрытой, то выходной заряд емкости  $C_Z$  частично перейдет на емкость  $C_X$ . Для предотвращения такой деградации выходного сигнала необходимо уменьшать емкость узла Z или во время предзаряда привязывать к высокому потенциалу не только узел X, но и узел Z, для чего потребуется дополнительный p-МОП.

Таким образом, преимущества динамических КМОП-вентилей перед статическими следующие:

- 1) простота – меньшее количество транзисторов;
- 2) меньшее потребление тока за счет исключения сквозных токов;
- 3) меньшие входные емкости;
- 4) малая задержка нарастания выходного сигнала.

Однако у динамических вентилей немало недостатков:

- 1) выходной сигнал сформирован только в период чтения;
- 2) задержка спада больше из-за дополнительного ключа;
- 3) деградация выходного сигнала;
- 4) ограниченная минимальная тактовая частота из-за разряда выходной емкости;
- 5) чувствительность к паразитным сигналам.

## 6.2. Логические элементы типа домино

Логические элементы типа домино расширяют возможности работы динамических элементов и позволяют сочетать простоту динамических схем с гибкостью классических (статических) схем.

**Домино-вентили.** В динамических элементах в течение фазы чтения разряженный узел выхода не может быть заряжен, т.е. не может правильно реагировать на входные сигналы. Другими словами, на переходы входных сигналов накладываются ограничения, и в общем случае, вы-



ход схемы не может работать на вход аналогичной схемы в одной фазе чтения. Для устранения такой ситуации вводят дополнительный статический инвертор (рис.6.4,а).

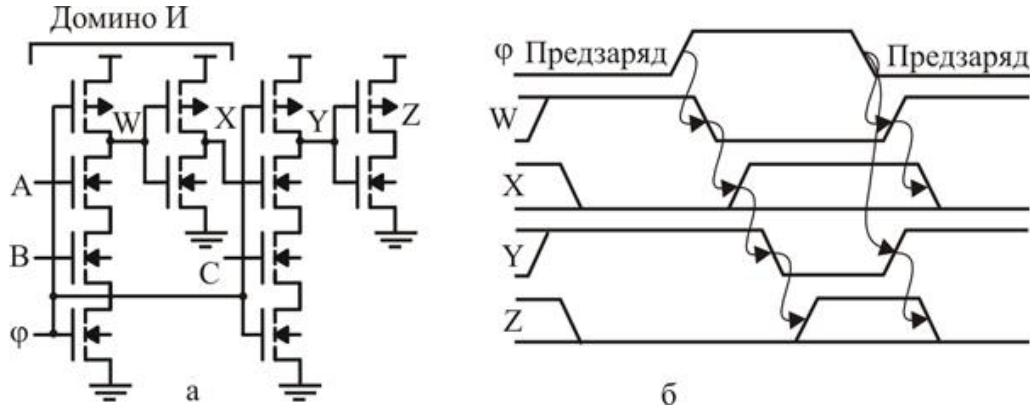


Рис.6.4. Цепочка домино вентилях: а – схема; б – временная диаграмма

Распространение сигнала по цепочке подобных вентилях происходит волнообразно (рис.6.4,б) и напоминает падение вертикально поставленных фишек домино, за что они и получили соответствующее название.

**Домино цепочка переноса.** На основе домино эффективно реализуется цепочка быстрого переноса в параллельных сумматорах (рис.6.5)

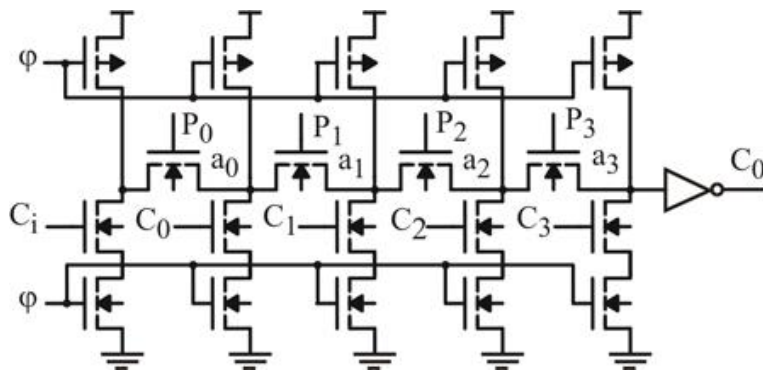


Рис.6.5. Домино цепочка переноса.

Во время низкого уровня тактового импульса φ потенциалы точек а в цепи переноса устанавливаются высокими. Наличие сигнала генерации переноса G снижает потенциал соответствующей точки до нуля. Наличие сигнала разрешения распространения переноса Р приводит к снижению потенциала вдоль цепочки – к распространению сигнала переноса.

**Дифференциальный домино-вентиль.** Домино вентили являются неинвертирующими, поскольку внутри вентиля сигнал инвертируется

дважды. Инвертирующий вентиль порождает дифференциальный элемент типа домино (рис.6.6).

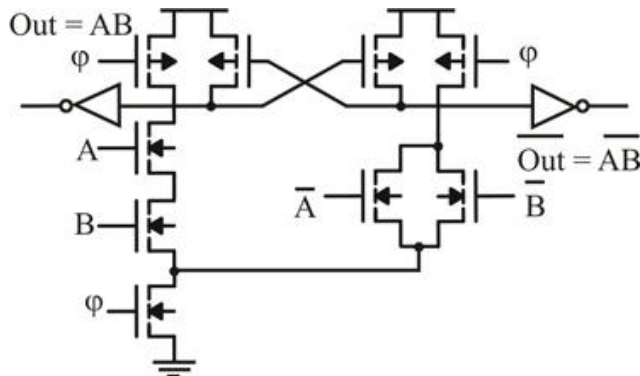


Рис.6.6. Дифференциальный домино-вентиль.

В его состав естественным образом вводится компенсирующий р-МОП-транзистор.

### 6.3. Динамические регистры сдвига

В динамических логических схемах информация передается через цепочку вентилей, которая, в частности, может быть организована как регистр сдвига.

**Динамический запоминающий элемент.** Простейший запоминающий элемент содержит входной ключ, запоминающий конденсатор и выходной усилитель (инвертор) (рис.6.7).

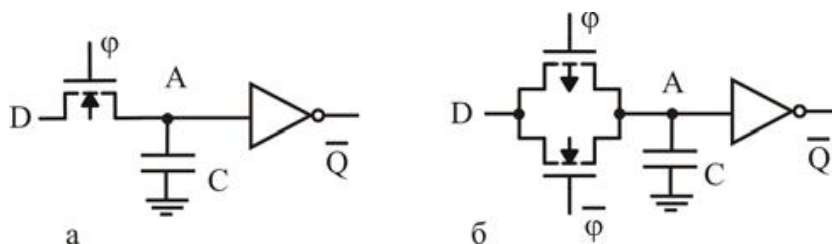


Рис.6.7. Динамический запоминающий элемент:  
а – с n-МОП ключом; б – с КМОП-ключом.

В качестве ключа используют как n-МОП ключ (рис.6.7,а), так и КМОП-ключ (рис.6.7,б).

**Однотактный регистр сдвига.** Последовательное соединение двух динамических запоминающих элементов, подсоединенных к двум фазам тактового сигнала, образует отдельный разряд регистра сдвига. Если использовать прямой и проинвертированный тактовые сигналы (пред-

полагается, что тактовые импульсы имеют быстрое нарастание и спад) (рис.6.8), то получается элемент однотокового регистра сдвига.

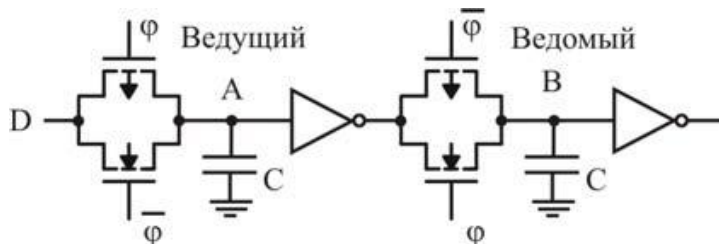


Рис.6.8. Однотоковый регистр сдвига (ведущий-ведомый)

Он работает по принципу ведущий-ведомый, а наличие двух последовательных инверторов сохраняет полярность сигнала. При низком  $\phi$  ведущий принимает новые входные данные, а ведомый сохраняет данные, принятые в предыдущем такте, и поставляет их на выход. При переходе к высокому уровню  $\phi$  вход элемента размыкается, данные от ведущего передаются к ведомому и с некоторой задержкой поступают на выход. Фактически элемент работает как задержка входного сигнала на один такт по переднему фронту тактового сигнала.

Такой регистр отличается простотой (всего 8 МОП), имеет высокое быстродействие, но чувствителен к крутизне фронта/спада тактового импульса. У регистра сдвига на тактируемых КМОП-инверторах (рис.6.9), подобный недостаток ослаблен, но за счет низкого быстродействия.

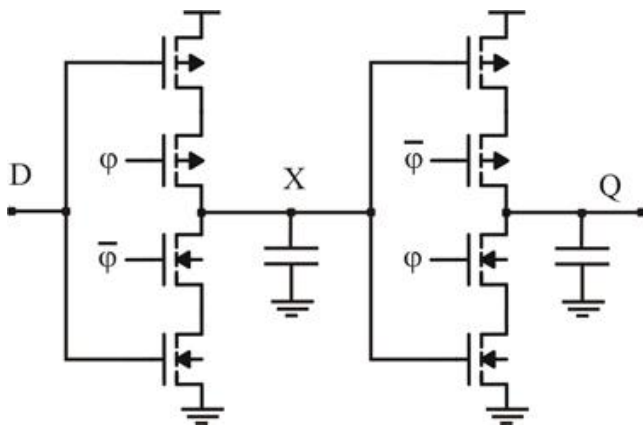


Рис.6.9. Регистр сдвига на тактируемых КМОП-инверторах

**Многотактные регистры сдвига.** Существует большое разнообразие многотактных регистров сдвига, в том числе. двух-, трех-, четырех-тактные, с неперекрывающимися и перекрывающимися тактовыми им-

пульсами. В качестве примера на рис.6.10 представлен двухтактный регистр сдвига с неперекрывающимися тактовыми сигналами.

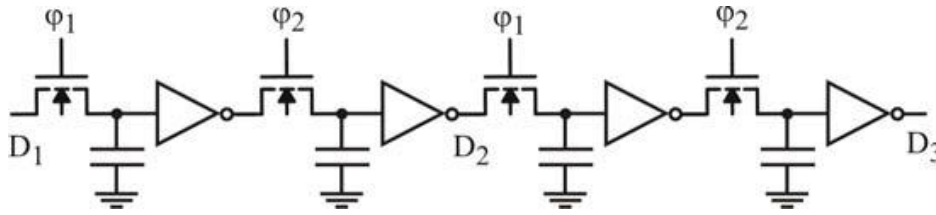


Рис 6.10. Двухтактный регистр сдвига

**Функциональные регистры сдвига.** Прохождение сигналов по регистру сдвига может сопровождаться их обработкой дополнительными логическими схемами. В частности, это могут быть схемы приема параллельных кодов в последовательный регистр сдвига.

## Глава 7. Последовательностные логические схемы

У последовательностных логических схем, в отличие от комбинационных, выходные сигналы зависят не только от входных сигналов, но и от внутреннего состояния схем, которое описывается переменными состояния.

Изменение внутреннего состояния последовательностной схемы под действием входных сигналов может происходить как синхронно с тактовым сигналом, так и в произвольные моменты времени (асинхронно).

### 7.1. Конечные автоматы и уравнения триггеров

**Конечный автомат.** При синхронной работе последовательностную схему называют конечным автоматом, и ее поведение описывают системой логических (характеристических) уравнений, зависящих от дискретного времени:

$$Y(n) = F(Q(n), X(n)); \quad (7.1)$$

$$Q(n+1) = G(Q(n), X(n)), \quad (7.2)$$

где  $Q$  – вектор внутреннего состояния;  $X$  – вектор входных переменных;  $Y$  – вектор выходных переменных;  $n$  – дискретное время (номер такта).

Для простоты характеристические уравнения перепишем в виде

$$Y = F(Q, X) \quad (7.3)$$

$$Q^+ = G(Q, X) \quad (7.4)$$

где верхний индекс “+” означает следующий момент времени.

В последовательностных схемах функция  $F$ , определяющая выходной сигнал, может зависеть от  $X$  (автомат Мура) или не зависеть (автомат Милия).

Функция  $G$ , определяющая изменение внутреннего состояния, может зависеть только от  $Q$  или только от  $X$ , что преобразует конечный автомат в генератор двоичных кодов или одноканальную задержку, соответственно.

При асинхронной работе последовательностную схему необходимо анализировать в непрерывном времени. Формулы остаются справедливыми, но приобретают другой смысл. Так,  $Q^+$  означает сдвиг на время задержки схемы  $t_0$ , т.е.

$$Q^+ = Q(t + t_0) = G(Q(t), X(t)) \quad (7.5)$$

При анализе асинхронных схем вместо булевой алгебры (исчисления высказываний) необходимо использовать логику предикатов (исчисление предикатов).

**Триггеры.** Последовательностные схемы с двумя внутренними состояниями называют триггерами. У несимметричного триггера один выход и  $Y = Q$ . У симметричного триггера два выхода: прямой и инверсный.

Простейшие триггеры имеют один вход. Возможны только два невырожденных варианта триггеров:

- 1) внутреннее состояние повторяет вход – вход поступает на выход с задержкой (в некоторых вариантах с инвертированием)

$$Q^+ = X \quad (7.6)$$

Такой триггер называют D-триггером.

- 2) вход меняет внутреннее состояние на противоположное –

$$Q^+ = \bar{X}Q + X\bar{Q} \quad (7.7)$$

Такой триггер называют T-триггером, или счетным.

Нетривиальных вариантов двухвходовых триггеров гораздо больше. Широкое распространение получили RS-триггер и JK-триггер.

Для RS-триггера

$$Q^+ = S + \bar{R}Q, \quad (7.8)$$

где R – установка нуля (восстановление); S – установка единицы (установка), причем  $SR = 0$  (одновременное присутствие сигналов установки и восстановления недопустимо, так как приводит к неопределенному состоянию).

Для JK-триггера

$$Q^+ = J\bar{Q} + \bar{K}Q. \quad (7.9)$$

На основе характеристических уравнений триггеров (7.6) – (7.9) можно построить обобщенную таблицу переходов в триггерах (табл.7.1).

**Переходы в триггерах**

**Таблица 7.1**

Переходы		D-триггер	T-триггер	RS-триггер		JK-триггер	
Q	Q <sup>+</sup>	D	T	R	S	J	K
0	0	0	0	~	0	0	~
0	1	1	1	0	1	1	~
1	0	0	1	1	0	~	1
1	1	1	0	0	~	~	0

Примечание: “~” означает независимость перехода от входного сигнала

В зависимости от способа хранения состояния триггеры разделяют на статические и динамические. В статических триггерах информация может храниться сколь угодно долго. Основой статического триггера является бистабильная ячейка, образованная перекрестным объединением инверторов. Динамические триггеры содержат логические элементы и запоминающие конденсаторы со схемами управления.

Триггеры могут быть реализованы на вентилях любых типов: комбинационных, проходных, динамических.

Во всех перечисленных случаях триггера могут быть синхронными и асинхронными.

## 7.2. Тактовая система

Синхронная работа цифровых схем требует введения тактовых импульсов, рис.7.1.

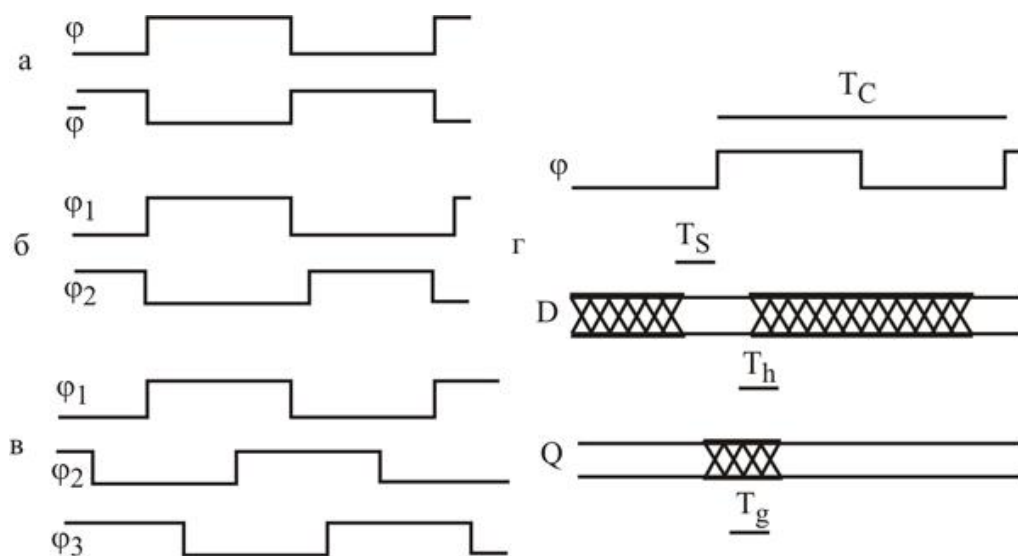


Рис.7.1. Однофазная (а) двухфазная (б) трехфазная (в) тактовые системы и временная диаграмма работы однофазного элемента, тактируемого по фронту (г)

Используются различные тактовые системы, отличающиеся количеством фаз и их взаимным перекрытием. Наиболее простой является однофазная (однотактная) система (рис.7.1,а). Фактически при использовании инвертирования в ней две фазы  $\phi$  и  $\bar{\phi}$ . Иногда, в двухфазной системе используются неперекрывающиеся импульсы  $\phi_1$  и  $\phi_2$  (рис.7.1,б). Могут быть использованы и многофазные системы с перекрытием импульсов (рис.7.1,в).



В простейшем случае применяется однофазная тактовая система (рис.7.1,г). За время одного периода (или его части) тактовых импульсов информация обрабатывается комбинационной схемой и поступает на вход данных  $D$  запоминающей схемы (например, триггера или набора триггеров – регистра). Последняя под управлением тактового сигнала  $\phi$  принимает данные, преобразует их (переходит в новое состояние) и с некоторой задержкой  $T_q$  формирует выходной сигнал  $Q$ . Для правильной работы входные данные в течение некоторого времени должны сохраняться неизменными. Так, при переключении триггера по фронту тактового импульса, данные должны быть установлены раньше фронта на время установки  $T_s$  и удерживаться в течение времени не меньшего, чем время удержания  $T_h$  после фронта  $\phi$  (рис 7.1,г).

Способ приема данных зависит от типа триггеров. Введение тактового управления триггером позволяет переключать его во время высокого или низкого уровня тактового сигнала. Такие триггеры называют одноктактными, или защелками. Во время разрешающего уровня тактового сигнала защелки ведут себя подобно асинхронным триггерам и могут переключаться многократно.

Более сложные двухтактные триггеры переключаются (тактируются) фронтом и/или спадом тактового сигнала. Для однофазных систем с двоичным тактовым импульсом  $\phi$  могут быть следующие варианты переключения триггеров (приема данных):

- 1) во время нижнего (отрицательного) уровня  $\phi$  – отрицательная защелка;
- 2) во время положительного уровня – положительная защелка;
- 3) во время нарастания импульса – переключение по нарастанию (фронту) (см. рис.7.1,г);
- 4) во время спада импульса – переключение по спаду;
- 5) во время любого перепада тактового импульса – переключение по перепаду.

Условные изображения триггеров при различных способах переключения (приема данных) приведены на рис.7.2.

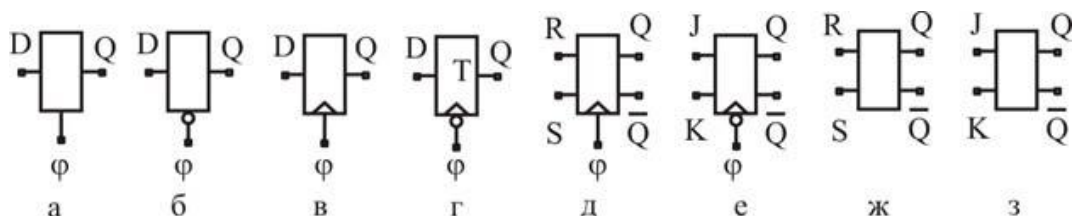


Рис.7.2. Условные обозначения триггеров: а – положительная защелка; б – отрицательная защелка; в – D-триггер с тактированием по фронту; г – T-триггер с тактированием по спаду; д – RS-триггер с тактированием по фронту; е – JK-триггер с тактированием по спаду; ж – асинхронный RS-триггер; з – асинхронный JK-триггер

### 7.3. Триггера на комбинационных вентилях

Простейший асинхронный RS-триггер может быть реализован на основе двухвходовых статических вентилей NAND2 (рис.7.3,а) или NOR2 (рис.7.3,б).

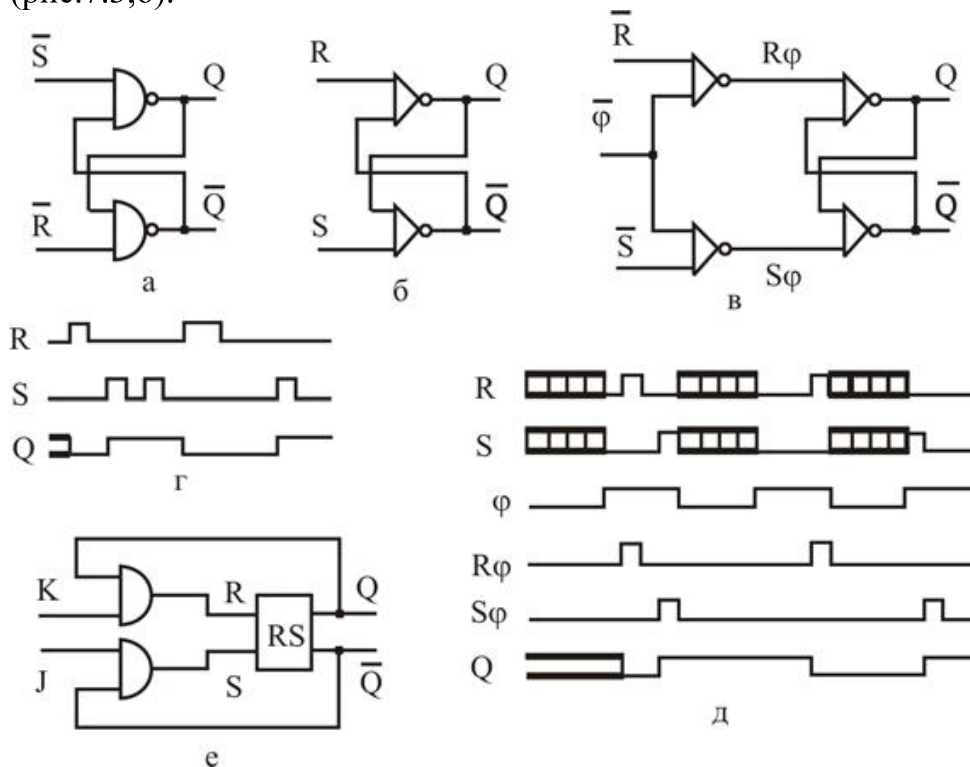


Рис. 7.3. Триггеры на комбинационных вентилях: а – NAND-RS; б – NOR-RS; в – положительная защелка; г – временная диаграмма асинхронного RS-триггера; д – временная диаграмма RS-защелки; е – асинхронный JK-триггер

В первом случае он сохраняет предыдущее состояние при высоких уровнях входов ( $R$  и  $S$ ), а во втором при низких уровнях входов.

Для перехода к синхронной работе триггера в него через вентили вводится сигнал управления  $\phi$  – разрешения изменения состояния (рис.7.3,в). В синхронном режиме работы RS-триггера сигналы  $S$  и  $R$  действуют на него только при определенном уровне сигнала управления  $\phi$ , например при положительном (рис.7.3,в), и сам триггер в этом случае является положительной защелкой.

Временные диаграммы RS-триггеров приведены на рис.7.3,г и 7.3,д.

Введением управляемой обратной связи RS-триггер преобразуется в JK-триггер (рис.7.3,е).

В современных СБИС триггеры на статических комбинационных вентилях используются ограниченно из-за большого числа компонентов и большой площади, занимаемой на кристалле.

### 7.4. Однотактные синхронные триггеры.

Однотактный синхронный D-триггер – защелка – в фазе приема передает входной сигнал D на выход Q, а в фазе хранения сохраняет выходной сигнал. У положительной защелки полярность фазы приема совпадает с полярностью тактового импульса, а у отрицательной защелки их полярности противоположны.

Динамические защелки (рис.7.4) сохраняют выходные сигналы в фазе хранения за счет зарядов на емкостях МОП-транзисторов.

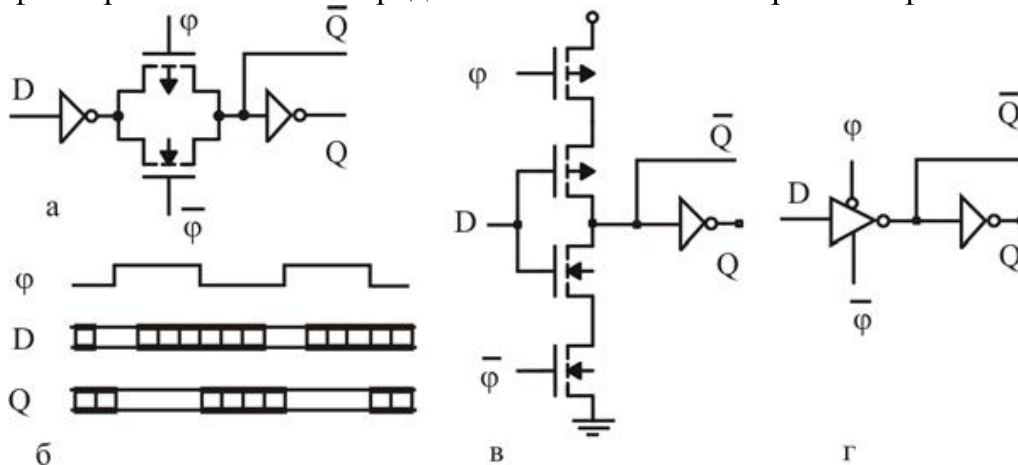


Рис.7.4. Динамические защелки: а – на проходных логических элементах; б – временная диаграмма отрицательной защелки; в, г – на тактируемых вентилях

Динамические защелки допускают различные варианты реализации, в том числе на проходных логических элементах (рис.7.4,а) и на тактируемых комбинационных вентилях (рис.7.4,в и г). Временная диаграмма работы обоих вариантов (отрицательной защелки) приведена на рис.7.4,б. У первого варианта в фазе приема открыт КМОП-ключ, у второго варианта активизирован инвертор. В результате входной сигнал поступает на прямой и инверсный выходы схемы.

В статических защелках для хранения информации используется положительная обратная связь. Для этого в ее состав вводятся дополнительные элементы, а сама обратная связь создается перекрестным соединением двух инверторов защелки в фазе хранения (рис.7.5).

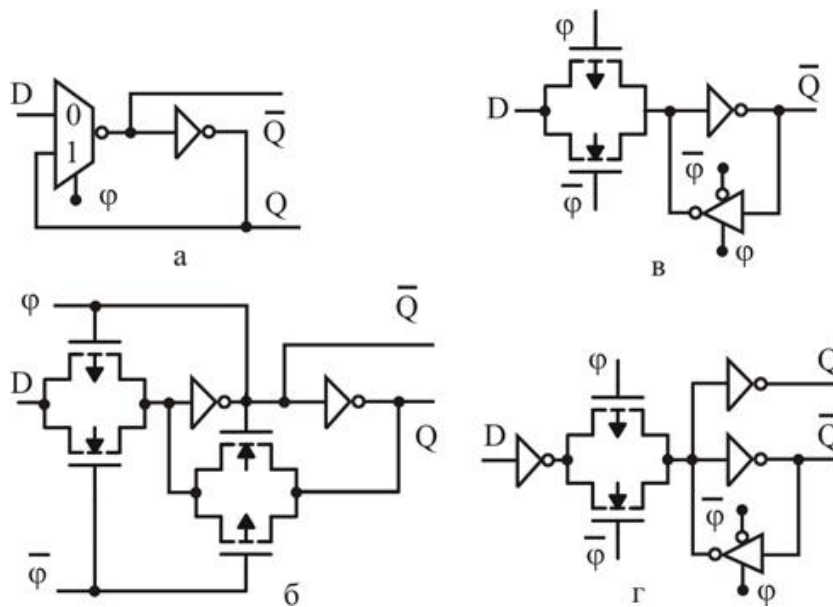


Рис.7.5. Статические КМОП защелки (а) с мультиплексором, (б) на проходных вентилях, (с) на тактируемых вентилях, (д) с входным и выходным инвертором.

Для замыкания обратной связи используют мультиплексор (рис.7.5,а), дополнительный КМОП-ключ (рис.7.5,б), тактируемый вентиль (рис.7.5,в). Для разделения входных и выходных цепей применяют специальные вентили (рис.7.5,г). Во всех случаях простая конструкция защелки позволяет иметь достаточно высокое быстродействие.

Конструктивно отрицательная и положительная защелки одинаковы, а их различие состоит в коммутации тактового сигнала (рис.7.6).

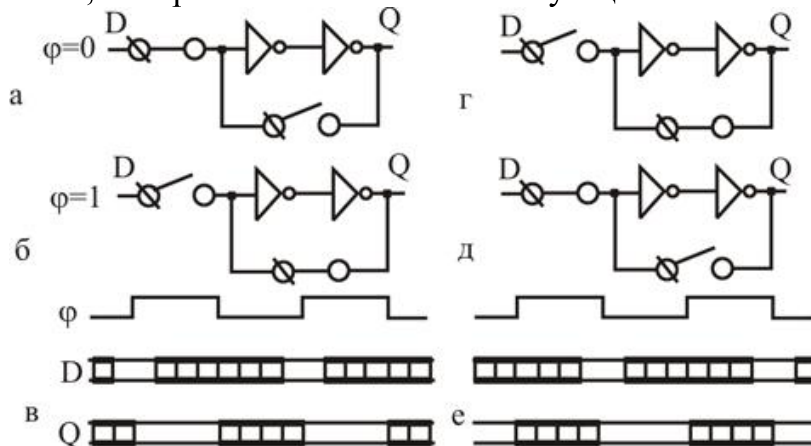


Рис.7.6. КМОП-защелки на проходных вентилях: а и б – отрицательная защелка; в – ее диаграмма работы; г и д – положительная защелка; е – ее диаграмма работы

При нижнем (нулевом) уровне тактового сигнала входной ключ отрицательной защелки включен (рис.7.6,а), а положительной защелки –

выключен (рис.7.6,г). На выход первой проходит входной сигнал, а на выходе второй – сохраняется его предыдущее значение. При верхнем (положительном) уровне тактового сигнала ситуация прямо противоположная: на выходе отрицательной защелки предыдущее состояние (рис.7.6,б), а на выходе положительной защелки входной сигнал (рис.7.6,д).

Временные диаграммы отрицательной и положительной защелок приведены на рис.7.6,в и 7.6,е, соответственно.

## 7.5.Двухтактные триггеры

Двухтактные D-триггеры состоят из разнополярных защелок, работающих по принципу ведущий – ведомый.

Динамический триггер требует для работы две фазы, но при использовании инвертирования допускает и однофазное питание (рис.7.7).

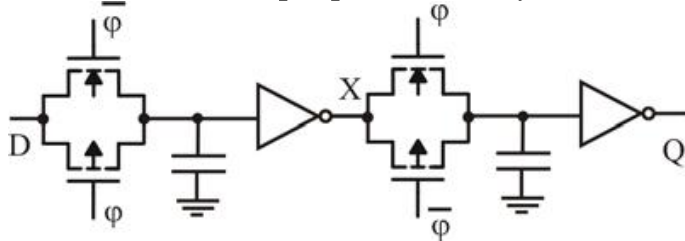


Рис.7.7. Динамический двухтактный D-триггер.

Широкое распространение получили статические двухтактные триггеры, образованные последовательным соединением разнополярных статических защелок. Так, при подсоединении к отрицательной защелке положительной получается двухтактный D-триггер, переключаемый по фронту (рис.7.8,а). Из временной диаграммы его работы (рис.7.8,б) следует, что в его внутреннем узле X во время положительного такта  $\phi$  уровень остается постоянным и именно он передается на выход.

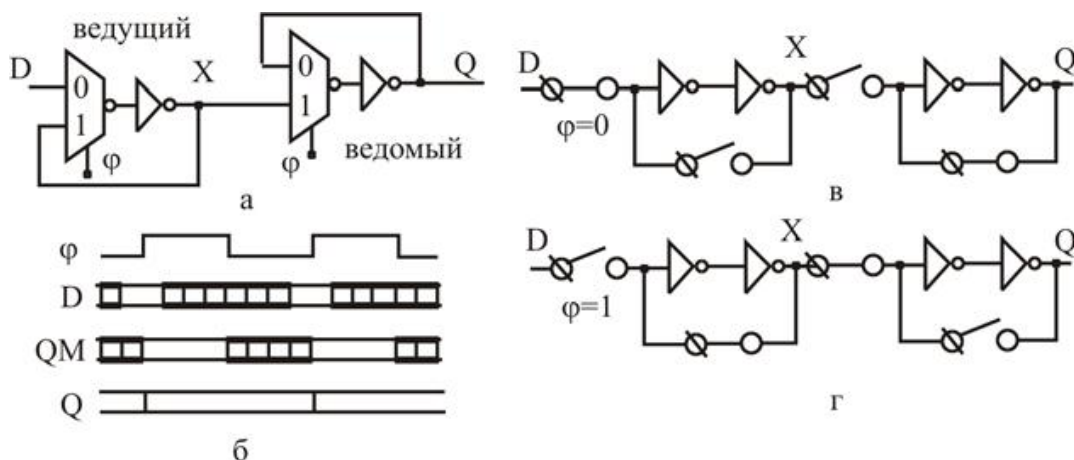


Рис .7.8. Статический двухтактный триггер с переключением по фронту, (а) схема D-триггера, (б) временная диаграмма, (с) режим приема, (д) режим хранения.

Положения ключей двухтактного D-триггера с переключением по фронту на проходных вентилях показаны на рис.7.8,в и 7.8,г. В режиме приема входной сигнал D проходит на выход первой (отрицательной) защелки, а вторая (положительная) защелка сохраняет предыдущее состояние. При переходе в режим хранения первая защелка отключается от входа, ее выход замыкается на ее вход и поступает на выход триггера Q.

Двухтактный D-триггер содержит четыре инвертора и четыре КМОП-ключа (и вспомогательный инвертор для инвертирования тактового сигнала) (см. рис.7.9,а).

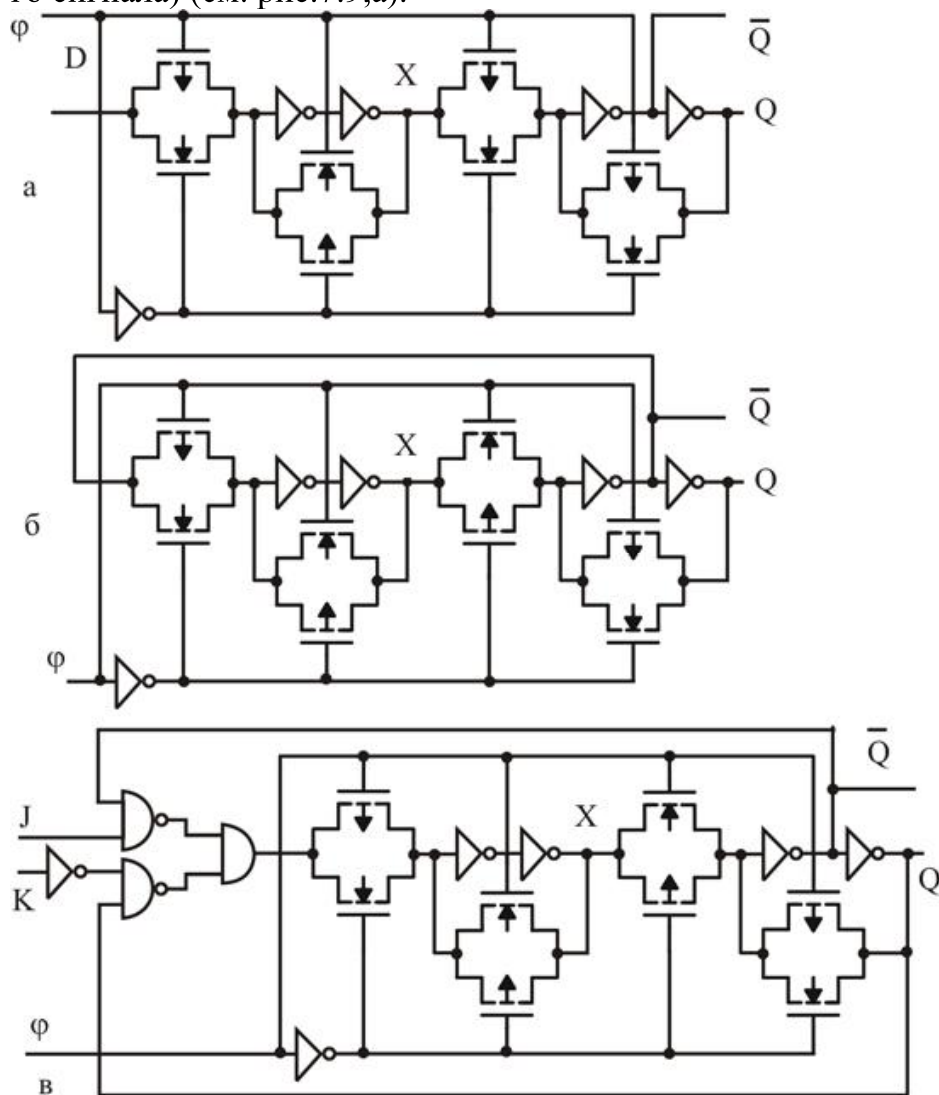


Рис.7.9. Двухтактные триггеры на проходных логических вентилях, (а) D-триггер, (б) Т-триггер, (в) JK-триггер.

На его основе могут быть построены и остальные типы триггеров, в т.ч. двухтактный Т-триггер (рис.7.9,б), JK-триггер (рис.7.9,в) и др.

## Глава 8. Полупроводниковая память

Полупроводниковая память занимает важное место в цифровых системах обработки информации. Во многих случаях ее объем и быстродействие определяют общую производительность системы. Основное отличие элементов памяти от логических элементов состоит в том, что они могут быть пассивными, т.е. работать без усиления сигнала. Это позволяет упростить их конструкцию и максимально уменьшить размеры. Как правило, запоминающие элементы объединяются в массивы (матрицы). Матрица элементов (ячеек) служит для хранения (накопления) информации. Ввод и вывод информации производятся специальными схемами управления, которые согласуют сигналы элементов памяти с внешними схемами, в том числе усиливают сигналы, считываемые с элементов памяти.

В состав управления входят схемы выборки (дешифраторы), драйверы для возбуждения шин матрицы, усилители считывания и схемы обмена данными.

Матрица накопителя (накопитель) вместе со схемами управления образует запоминающее устройство (ЗУ). Проектирование полупроводникового ЗУ тесно связано с технологическим маршрутом его изготовления. Как правило, при проектировании накопителя для повышения плотности элементов используют специальные нормы проектирования, которые учитывают специфику элементов памяти. Эти нормы могут существенно отличаться от норм проектирования цифровых логических элементов. Более того, некоторые типы полупроводниковой памяти требуют дополнительных операций в КМОП технологическом процессе.

Достоинством полупроводниковой памяти является совместимость ее технологии с технологией других элементов: логических, аналоговых и др. Недостатки полупроводниковой памяти состоят в том, что необходимо использовать несколько типов запоминающих элементов. Самые быстрые элементы на основе усилителей (триггеры) требуют постоянных источников питания. При хранении информации в виде зарядов на конденсаторах, управляемых ключами, имеет место достаточно быстрое (секунды) стекание зарядов. И, наконец, возможность длительного хранения зарядов в толще диэлектриков сопряжена с трудностями управления такими зарядами.

Современное состояние технологии памяти позволяет успешно использовать все перечисленные способы хранения информации в полупроводниковых структурах, а ее непрерывное развитие обеспечило создание ЗУ гигабитного объема. На этом пути полупроводниковые ЗУ выступают одной из главных движущих сил. Именно степень интеграции – ин-



формационная емкость кристаллов полупроводниковых ЗУ является показателем уровня технологии.

## 8.1. Архитектура ЗУ

**Классификация полупроводниковых ЗУ.** Полупроводниковые запоминающие устройства разделяются на классы по ряду признаков:

- 1) по назначению:
  - а) оперативные;
  - б) постоянные;
  - в) полупостоянные
- 2) по способу хранения информации:
  - а) статические;
  - б) динамические;
- 3) по возможности сохранения информации при отключении питания:
  - а) с сохранением информации – энергонезависимые;
  - б) без сохранения – энергозависимые;
- 4) По способу выборки:
  - а) с произвольной выборкой;
  - б) с последовательной выборкой.

В цифровых системах широко используют следующие типы (классы) полупроводниковых ЗУ:

- 1) статическое оперативное ЗУ – СОЗУ (SRAM) – в качестве сверхбыстродействующей (сверхоперативной) памяти;
- 2) динамическое оперативное ЗУ – ДОЗУ (DRAM) – в качестве оперативной (массовой) памяти;
- 3) постоянное ЗУ – ПЗУ (ROM) – в качестве постоянной памяти, программируемой в процессе изготовления;
- 4) программируемое постоянное ЗУ – ППЗУ (PROM) – в качестве постоянной памяти, однократно программируемой пользователем;
- 5) электрически репрограммируемое ЗУ – ЭРППЗУ ( $E^2$ PROM) – в качестве полупостоянной памяти, многократно перепрограммируемой пользователем.

Первые два типа памяти энергозависимые, остальные энергонезависимые.

**Организация накопителя ЗУ.** Большинство типов полупроводниковых ЗУ используют произвольную выборку. Они представляют собой матрицу элементов (ячеек) памяти, объединенных системой шин в накопитель (рис.8.1).

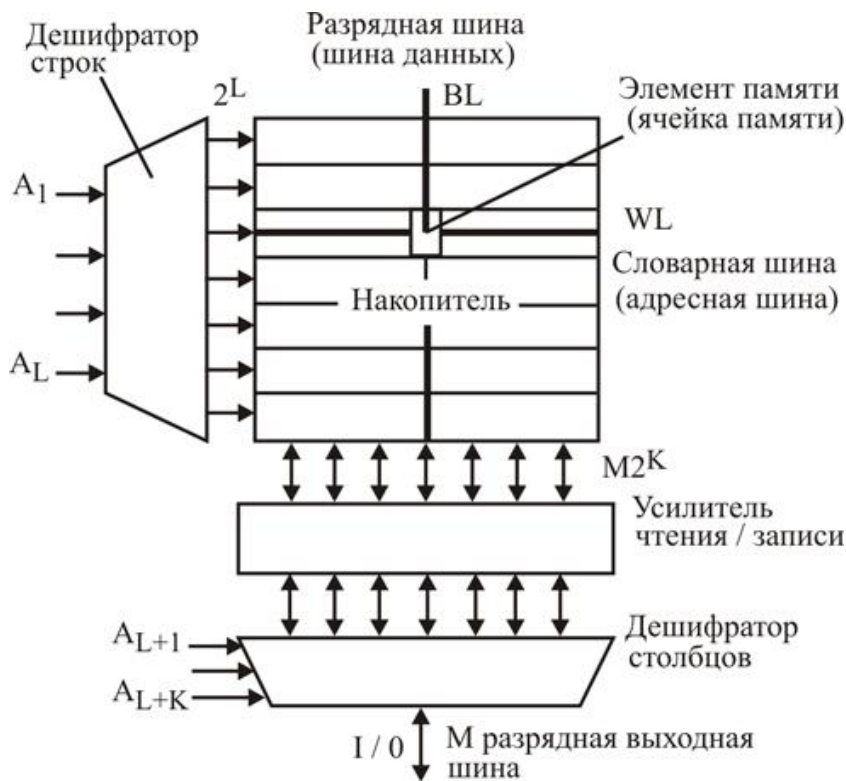


Рис.8.1. Организация ЗУ с произвольной выборкой.

Строки (слова) матрицы накопителя управляются словарными (адресными) шинами (WL). Столбцы (разряды) матрицы управляются разрядными шинами (шинами данных). Обычно одному элементу соответствует одна словарная шина и одна – две шины данных. Кроме того, в накопителе могут быть общие для всех элементов шины (питания, общего управления).

Для управления накопителем с  $2^N$  ячейками служат  $N$ -разрядный адрес и управляющие сигналы, определяющие режим работы: выборка накопителя, запись/чтение. Обмен информацией происходит по  $M$ -разрядной выходной шине данных ( $M$  может равняться и 1).

В составе адресного слова  $L$  разрядов задают адрес одной из  $2^L$  строк накопителя, а  $K$  разрядов задают адрес одной из  $2^K$  групп столбца накопителя, так что  $N = L + K$ .

Первая часть адреса  $[A_1 \dots A_L]$  поступает на дешифратор строк, а вторая часть  $[A_{L+1} \dots A_{L+K}]$  на дешифратор столбцов. Выходы дешифратора строк через драйверы возбуждают адресные шины. Выходы дешифратора столбцов выбирают  $M$  усилителей чтения (считывания)/записи. В режиме записи последние передают входной код на разрядные шины. В режиме чтения они передают сигнал с разрядных шин на выходные драйверы схемы.

Таким образом, основными схемами обрамления накопителя являются дешифраторы с драйверами и усилители считывания. Элементы дешифратора были описаны в главах 4 и 5. Усилители считывания (компараторы) будут рассмотрены в главе 9.

При большой емкости ЗУ его разбивают на блоки (рис.8.2).

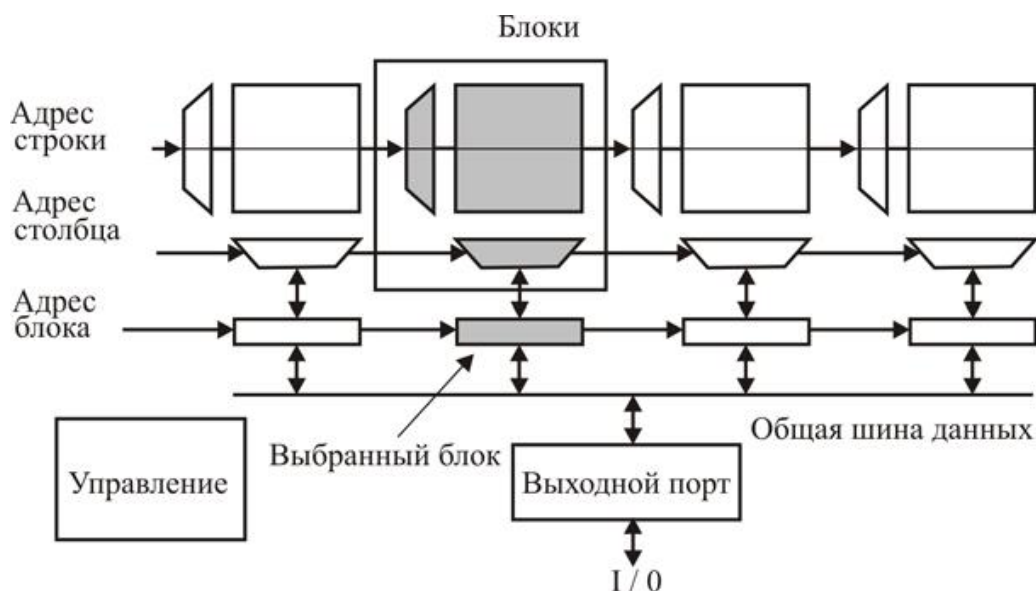


Рис.8.2. Иерархическая организация ЗУ большого объема.

Каждый блок содержит накопитель со схемами обрамления. А в составе иерархического ЗУ появляются дополнительные дешифраторы блоков и общая шина данных.

**Варианты организации ПЗУ.** Простейшей полупроводниковой памятью является ПЗУ. Постоянная информация хранится в виде присутствия или отсутствия в ячейках матрицы накопителя нелинейного элемента, например, МОП-транзистора (рис.8.3).

Возможны различные варианты считывания информации из МОП ПЗУ, но наиболее часто используются два из них: подключение n-МОП к разрядной шине по схеме NOR (рис.8.3,а) или по схеме NAND (рис.8.3,б).

Выборкой строки будем считать появление положительного (NOR) и отрицательного (NAND) импульса на одной из словарных шин. В первом варианте за «0» в ячейке памяти примем наличие n-МОП, а во втором — его отсутствие. Причем, может отсутствовать не сам транзистор, а его подсоединение (контакты) к шине.

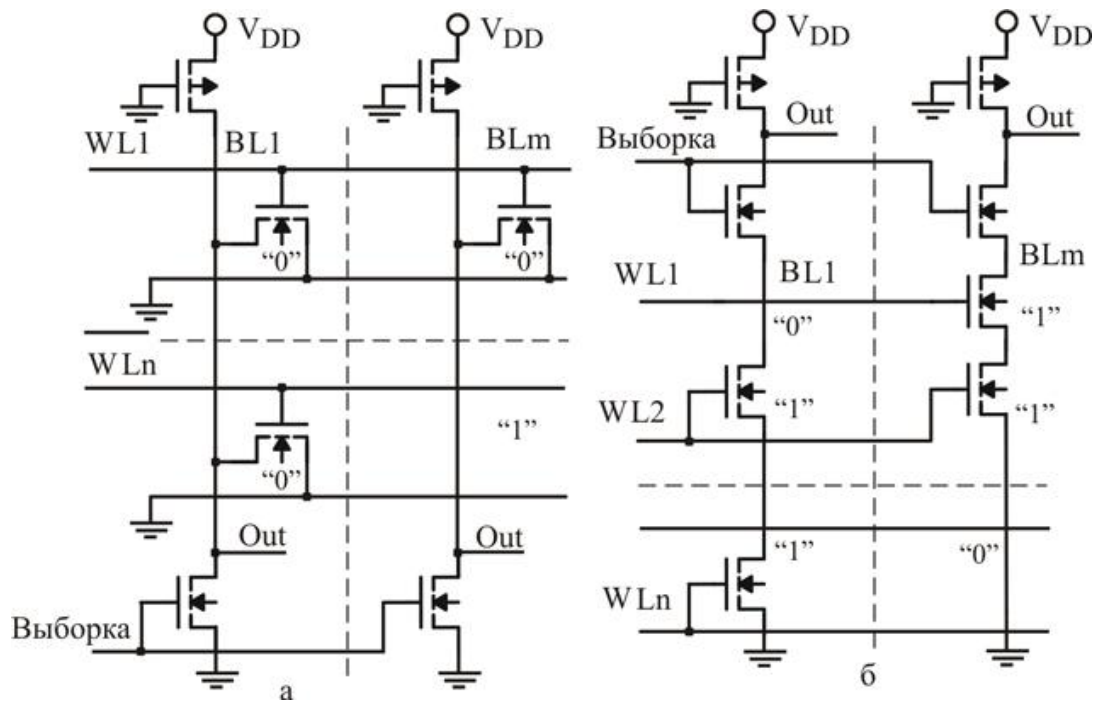


Рис.8.3. Организация ПЗУ: а – NOR; б – NAND.

В NOR ПЗУ первоначально на словарных шинах низкий потенциал, а на выходах – высокий. При возбуждении словарной шины высоким потенциалом подсоединенные к ней n-МОП открываются. В результате при подаче сигнала выборки на тех разрядных шинах, в которых у выбранного слова записаны нули, выходное напряжение понизится.

В NAND ПЗУ первоначально на словарных шинах и на выходах высокий потенциал. При возбуждении словарной шины низким потенциалом подсоединенные к ней n-МОП закрываются. При подаче сигнала выборки на тех разрядных шинах, в которых у выбранного слова записаны единицы, выходной потенциал не изменится, а у остальных понизится. Поскольку в NAND ПЗУ транзисторы включены последовательно, то у них время считывания больше, чем у NOR ПЗУ. С другой стороны, плотность элементов у NAND ПЗУ гораздо выше, чем у NOR ПЗУ.

**Временная диаграмма работы ЗУ.** Внешними сигналами ЗУ являются коды адреса и выбор режима. Для экономии количества выводов в ЗУ большого объема (в ДОЗУ) используют мультиплексирование адреса, т.е. отдельно задают адрес строки (по стробирующему сигналу RAS) и адрес столбца (по стробирующему сигналу CAS) (рис.8.4).

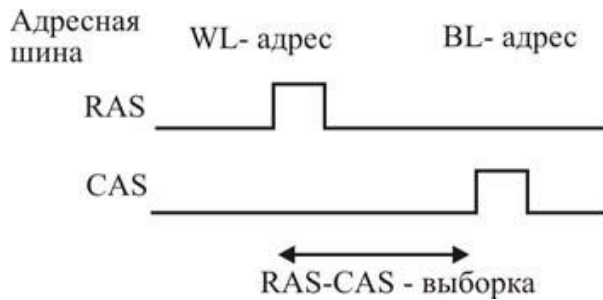


Рис.8.4. Выборка адреса: а – с мультиплексированием (ДОЗУ); б – без мультиплексирования

В быстродействующих СОЗУ, адрес подается полностью и для ускорения работы (исключения стробирования) непосредственное изменение адреса инициализирует выборку.

## 8.2. Основные типы элементов памяти

**Элементы РППЗУ.** В многократно перепрограммируемых ЗУ РППЗУ в качестве элемента памяти наиболее часто используют транзистор с плавающим затвором (рис.8.5,а).

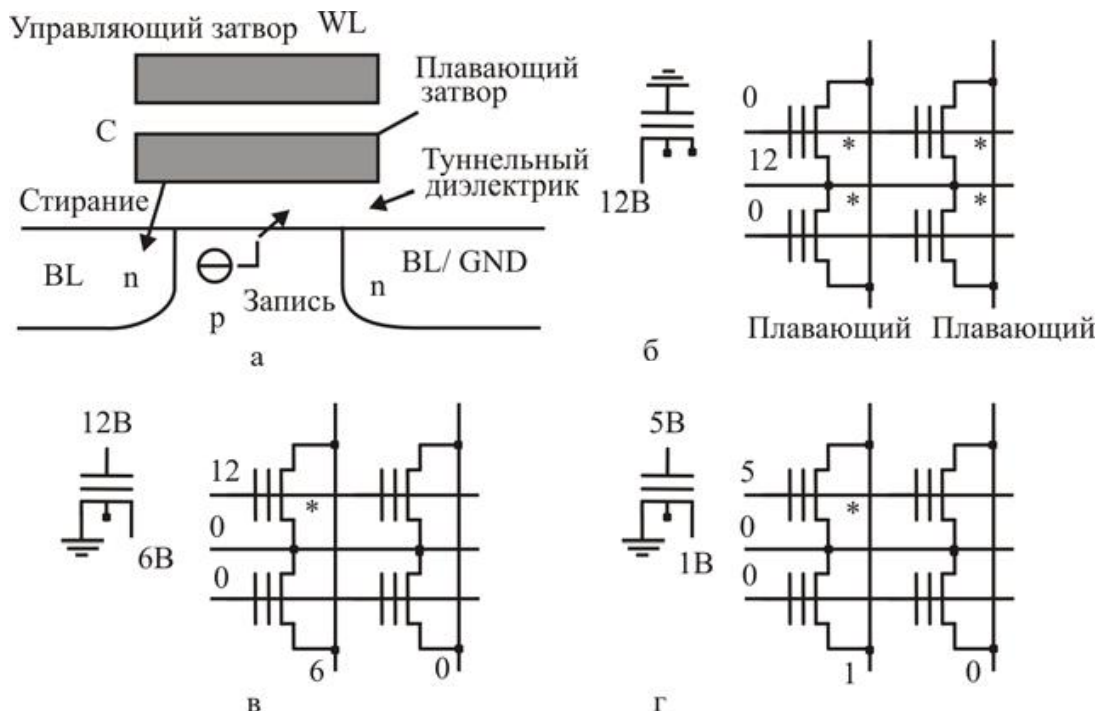


Рис.8.5. Транзистор с плавающим затвором: а – структура; б – режим стирания; в – режим записи; г – режим чтения

Он, как следует из его названия, наряду с обычным (управляющим) затвором имеет другой (плавающий) затвор. Заряд на плавающем

затворе определяет пороговое напряжение транзистора. При наличии отрицательного заряда n-МОП закрыт, а в отсутствие заряда (или при малой его величине) транзистор открыт. При считывании элемент РППЗУ аналогичен элементу ПЗУ, т.е. транзистор с плавающим затвором ведет себя как обычный транзистор, и его открывание определяется соотношением порога МОП и сигнала на затворе (рис.8.5,г).

Запись заряда на плавающий электрод (рис.8.5,в) и его удаление (стирание заряда) (рис.8.5.б) в ЭРППЗУ производятся при подаче повышенного напряжения соответствующей полярности на затвор по отношению к электродам и каналу прибора. Управление зарядом плавающего электрода может быть локальным (адресным), или общим для всего накопителя или его секции. В последнем случае конструкция накопителя существенно упрощается. ЭРППЗУ такого типа получили название ФЛЭШ ЗУ.

Программирование элементов РППЗУ происходит достаточно медленно, поскольку токи через диэлектрик невелики. Протекание токов вызывает постепенную деградацию диэлектрика, так что количество циклов перепрограммирования ограничено. Повышая напряжение программирования, запись можно ускорить, но деградация также возрастет.

Типичное время записи в транзисторах с плавающим затвором составляет единицы микросекунд, а время стирания единицы миллисекунд. Последнее для ФЛЭШ ЗУ несущественно, так как в них стирание производится большими массивами. При таких временах программирования допустимое количество циклов обычно превышает  $10^5$ .

**Элементы СОЗУ.** В статических оперативных ЗУ СОЗУ элементом памяти служит статический симметричный триггер, дополненный транзисторами выборки (рис.8.6).

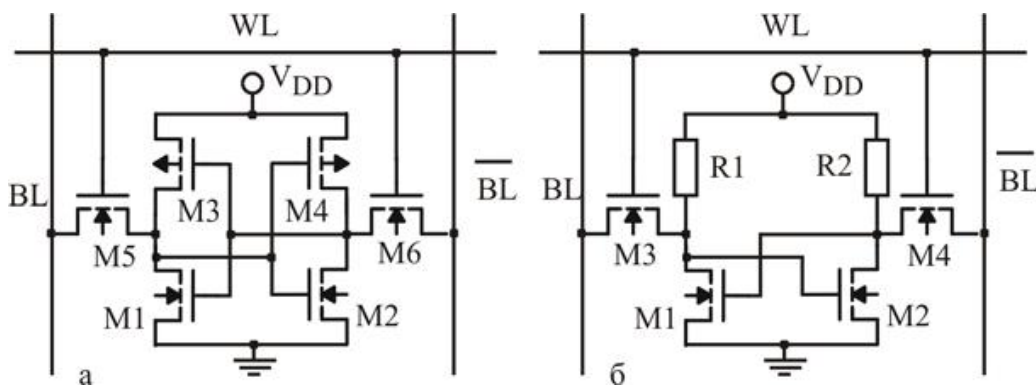


Рис.8.6. Элементы СОЗУ: а – шеститранзисторный, б – с резисторами.

Наиболее часто используется шеститранзисторный элемент (рис.8.6,а). Четырехтранзисторный триггер с резистивными нагрузками

(рис.8.6,б) имеет более простую конструкцию и занимает меньшую площадь. Однако при его изготовлении требуются дополнительные технологические операции получения высокоомных резисторов.

При хранении информации в элементе СОЗУ словарная шина находится под нулевым потенциалом, и триггер изолирован..

При выборке на словарной шине появляется высокий потенциал и плечи триггера через транзисторы выборки М5 и М6 подсоединяются к разрядным шинам.

При записи на разрядные шины подаются коды данных, подлежащих записи. Они устанавливают триггер в состояние нуля или единицы в соответствии с потенциалами на разрядных шинах.

При считывании разрядные шины переводятся в режим плавающего потенциала, и триггер через транзисторы выборки начинает их заряжать в соответствие со своим состоянием.

**Элементы ДОЗУ.** В динамических оперативных ЗУ (ДОЗУ) информация представлена в виде заряда запоминающего конденсатора  $C_S$ , подсоединенного к разрядной шине BL, через транзистор выборки М, управляемый по затвору словарной шиной WL (рис.8.7,а).

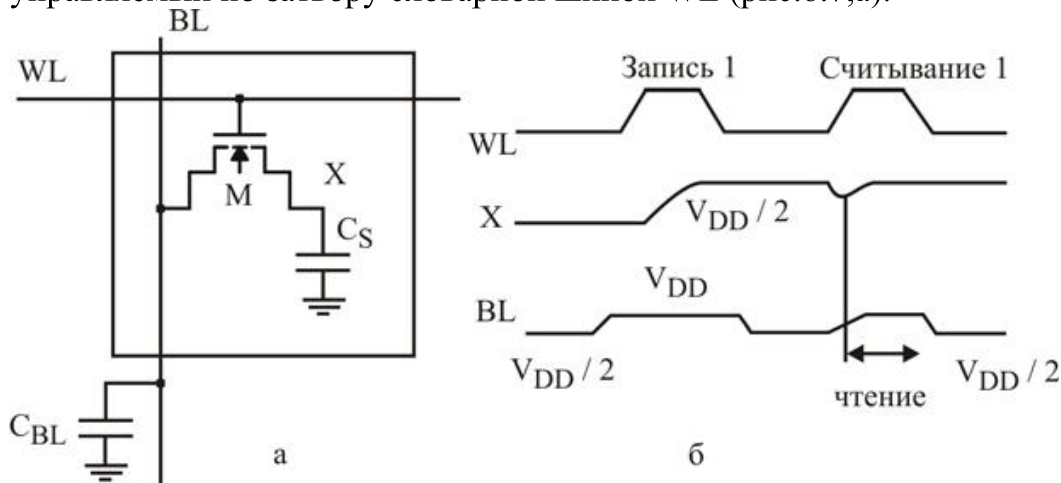


Рис.8.7. Элемент ДОЗУ: а – электрическая схема; б – временная диаграмма работы

В режиме хранения на словарной шине WL низкий потенциал и транзистор выборки закрыт (рис.8.7,б). Заряженное состояние запоминающего конденсатора единица может сохраняться в течение длительного времени (миллисекунды).

При выборке на словарную шину подают положительный импульс, который открывает транзистор выборки.

Для записи единицы на разрядную шину подают высокое напряжение записи, а для записи нуля подают нулевое напряжение.

При считывании разрядная шина переводится в режим плавающего потенциала и при открывании транзистора выборки заряд запоминаю-



щего конденсатора перезарядит разрядную шину. Поскольку емкость разрядной шины  $C_{BL}$  много больше емкости запоминающего конденсатора  $C_S$ , то сигнал считанной единицы на разрядной шине будет во столько же раз меньше, т.е.

$$\Delta V_{LB} \approx \frac{C_S}{C_{BL}} V_{DD}.$$

Обычно за счет конструкции ячейки, рис.8.8,  $C_S / C_{BL} \sim 10$ , и напряжение считывания при  $V_{DD} = 2,5 \text{ В}$ , не превышает 300 мВ.

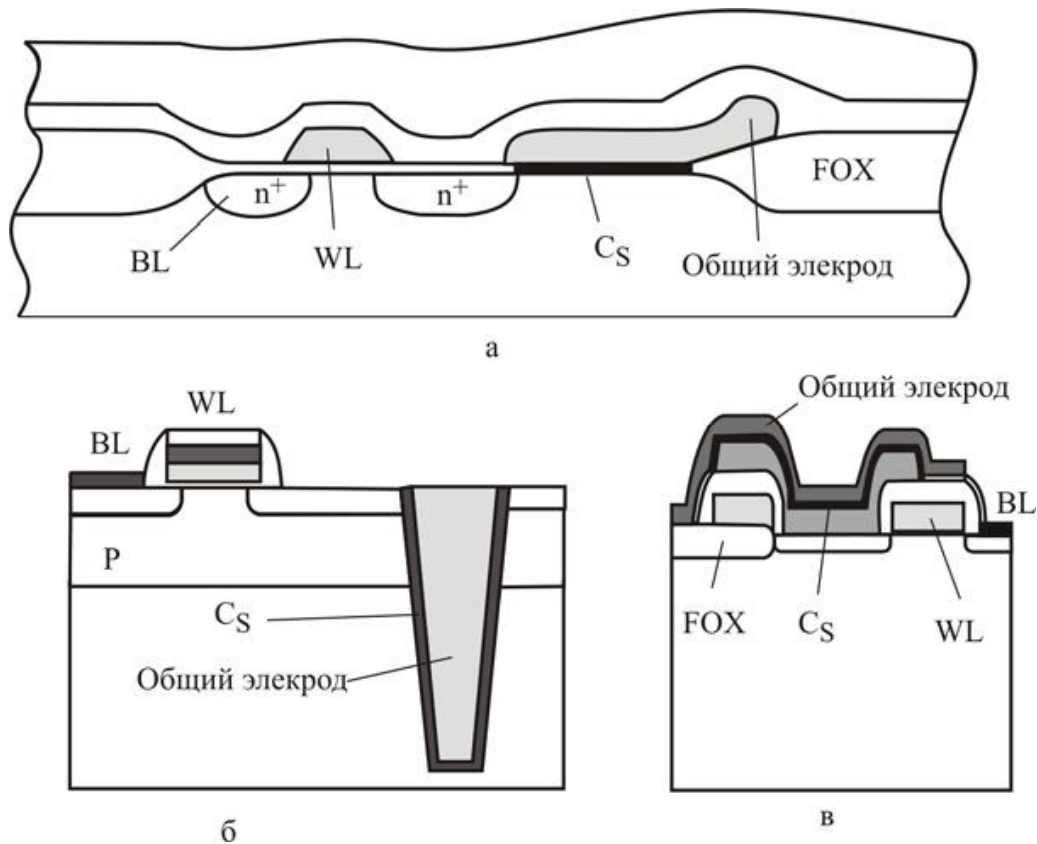


Рис.8.8. Структура элемента ДОЗУ: а – с планарным конденсатором; б – с конденсатором в канавке; в – со свернутым конденсатором.

Информация в ячейке ДОЗУ при считывании разрушается. Поэтому считанный сигнал, не снимая управление выборкой, усиливают до стандартного уровня и вновь записывают в ячейку.

Поскольку информация в ячейке ДОЗУ разрушается со временем, ее регенерируют, путем периодического считывания.

Основная проблема ДОЗУ – хранение достаточно большого заряда на малой площади элемента. Заряд не может быть очень малым так как это приведет к уменьшению сигнала считывания и снизит надежность его хранения.



Первоначально в элементах ДОЗУ использовались планарные конденсаторы, (рис.8.8,а). При увеличении емкости ЗУ необходимо было сокращать площадь ячейки памяти. Для сохранения емкости конденсатор стали делать трехмерным, углубляя в подложку (рис.8.8,б) или размещая над транзистором выборки (рис.8.8,в).

### 8.3.Параметры элементов памяти

Основными характеристическими параметрами памяти являются:

1) информационная емкость;

2) быстродействие, включая:

а) время выборки – время от разрешения (строба) выборки до появления выходного сигнала;

б) время цикла – минимальный период повторения обращения к памяти;

3) потребление.

**Сравнение параметров элементов памяти.** Сравнительные характеристики наиболее широко используемых элементов памяти (изготовленных по технологии 0,09 мкм КМОП) приведены в табл.8.2.

#### Сравнительные характеристики

элементов полупроводниковых ЗУ

*Таблица 8.2*

Параметр	СОЗУ	ДОЗУ	ФЛЭШ
Рабочее напряжение, В	1,5 – 2,5	1,5 – 2,5	10
Энерго-независимость	Нет	Нет	Да
Количество обращений	$\infty$	$\infty$	$10^5$
Площадь, мкм <sup>2</sup> )	1	0,17/0,08	0,08
Время выборки, нс	2	5/50	25/50
Ток (мкА/Мбайт)	10	100/3	< 1
Стоимость	Низкая	Высокая/ низкая	Высокая/ низкая

Примечание: значения под косой чертой соответствуют

ЗУ с упрощенным протоколом обмена

**Прогноз уменьшения размеров элементов памяти.** Запоминающие устройства прошли длинный путь развития. Являясь самым массовым продуктом интегральной технологии, они продолжают оказывать на нее стимулирующее влияние и именно их параметры прогнозируются в первую очередь. В табл.8.3. трехлетнему интервалу времени и минимальному топологическому размеру F поставлены в соответствие площади элементов основных типов ЗУ.

#### Прогнозирование уменьшения размеров элементов ЗУ

*Таблица 8.3*

Параметр	2004	2007	2010	2013	2016
Топологический размер F, нм	90	65	45	32	22
Площадь элемента ДОЗУ, 0,001 мкм <sup>2</sup>	80	28	12	6	2,5
Площадь элемента СОЗУ, 0,001 мкм <sup>2</sup>	1500	730	220	110	52
Площадь элемента ФЛЭШ, 0,001 мкм <sup>2</sup>	100	50	34	18	10

## Глава 9. Специальные схемы

### 9.1. Схемы ввода вывода

**Входной каскад.** Интегральные схемы чувствительны к перегрузкам по входу. Очень опасен электростатический разряд. Для ослабления его воздействия во входных каскадах ИС применяют специальные схемы защиты, которые не должны ухудшать параметры входного каскада; должны иметь небольшую площадь; эффективно ограничивать напряжение и токи перегрузки, возникающие при разряде; иметь максимально быстрое время срабатывания и вносить минимальное время задержки в нормальную работу ИС.

В составе схемы защиты обычно используют специальный резистор, подсоединяемый ко входу ИС (непосредственно к ее контактной площадке) и ключ, срабатывающий при повышении напряжения и закорачивающий входной сигнал через защитное сопротивление на источник питания. Ключами служат диоды, биполярные и МОП-транзисторы.

Типичный входной каскад со схемой защиты от электростатического разряда приведен на рис.9.1.

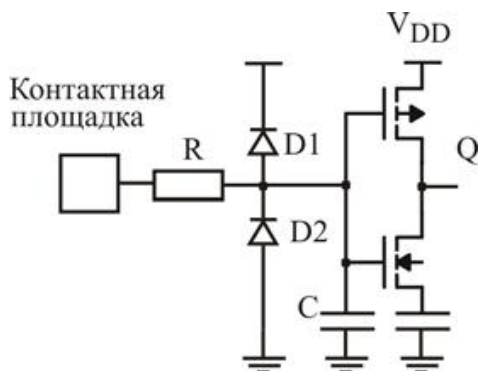


Рис.9.1 Входной каскад с защитой от электростатического разряда

**Выходной каскад.** В качестве выходного каскада ИС широко используются магистральные усилители, тактируемые парафазным сигналом разрешения вывода Е, (рис.9.2). Выход магистрального усилителя подсоединяется к контактной площадке.

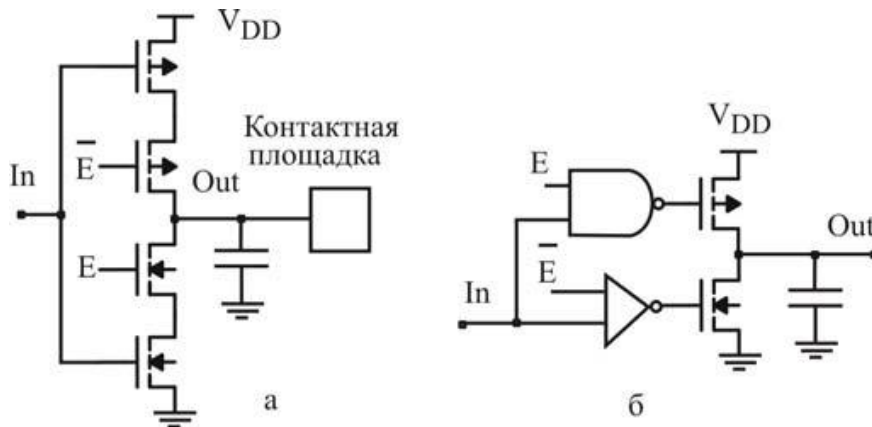


Рис.9.2. Выходной каскод: а – с тактируемым каскодом; б – с тактируемыми вентилями

Более простой выходной каскод с тактируемым каскодом (рис.9.2,а) имеет высокое выходное сопротивление, для его снижения используют более сложную схему с тактируемыми вентилями (рис.9.2,б)

Поскольку магистральные усилители допускают объединение по входу, его контактная площадка может быть и входной площадкой, т.е. в случае необходимости вход и выход могут быть совмещены.

## 9.2. Триггер Шмидта

Двоичный сигнал на входе ИС нередко сильно искажен из-за внешних реактивностей и наводок. Зачастую его переключение сопровождается значительными колебаниями уровня (дребезгом), и они могут быть настолько велики, что приводят к ложным переключениям внутренних вентилях ИС. Для исключения подобных явлений во входных каскадах используют гистерезисные схемы, в которых пороги включения и выключения разнесены, и схема после переключения некоторое время нечувствительна к изменению входного сигнала. В качестве такой схемы часто используется триггер Шмидта (рис.9.3,а).

Гистерезис в триггере Шмидта (рис.9.3,б) достигается за счет нелинейной отрицательной обратной связи. Выходной сигнал воздействует на параметры n-МОП и p-МОП-цепей инвертора и нарушает их согласование. При низком выходе открывание M6 ослабляет p-МОП-цепь и напряжение переключения инвертора сдвигается к нулевому потенциалу. При высоком выходе открывание M3 приводит к противоположному эффекту. В результате искажения входного сигнала триггера Шмидта на его выходе устраняются (рис.9.3,в).

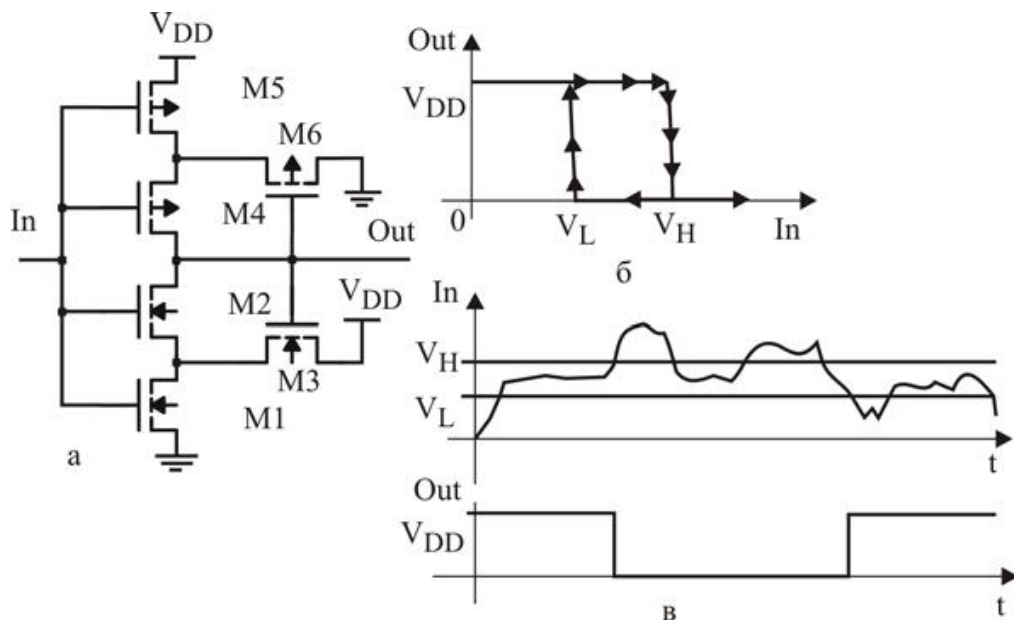


Рис.9.3. Триггер Шмидта: а – электрическая схема; б – передаточная характеристика; в – формирующие свойства

### 9.3. Компаратор

В ряде случаев уровни сигналов цифровых элементов отличаются от стандартных уровней 0 и 1. Подобная ситуация имеет место в накопителях запоминающих устройств, в которых перепад двоичных сигналов считывания с элементов памяти может быть намного меньше  $V_{DD}$ . Для восстановления стандартных уровней сигналов используются специальные элементы – компараторы.

Компаратор (рис.9.4), является пороговым элементом с резкой передаточной характеристикой

$$V_{out} = \begin{cases} V_{OH}, & V_{in} > V_{ref}; \\ V_{OL}, & V_{in} \leq V_{ref}. \end{cases}$$

где  $V_{ref}$  – опорное напряжение. В качестве опорного напряжения может использоваться как опорный потенциал, так и второй входной сигнал.

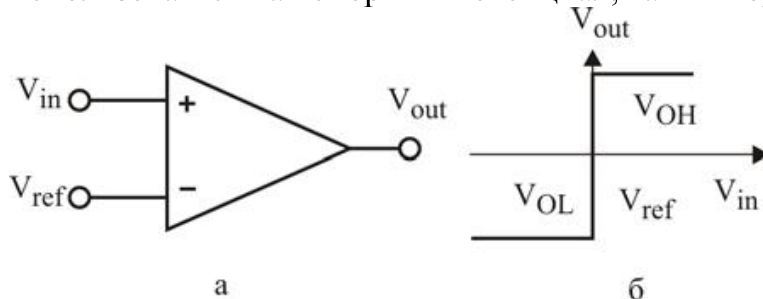


Рис.9.4. Компаратор: а – условное изображение; б – передаточная характеристика

Компараторы широко применяются в смешанных схемах (использующих как цифровые, так и аналоговые сигналы) для сравнения сигналов между собой. Наряду с прямым выходом они могут иметь и инвертирующий. Небольшой гистерезис передаточной характеристики компаратора позволяет избежать ложных переключений из-за шума входных сигналов. Чувствительность компараторов зависит от их быстродействия и обычно лежит в диапазоне от сотен микровольт до единиц милливольт.

При построении компараторов используют различные подходы, а также разные схемы: операционный усилитель с большим коэффициентом усиления, цепочку вентилях, триггер Шмидта, стробируемый триггер (защелку). Последний получил в цифровых схемах наибольшее распространение.

Компараторы на стробируемых триггерах, рис.9.5, представляют собой триггера, которые имеют два режима. При низком значении тактового импульса  $\phi$  цепь питания триггера разомкнута и его выход принудительно устанавливают в среднее состояние. В это время на входе компаратора присутствуют входные сигналы, которые вносят разбаланс в плечи триггера. При переходе тактового сигнала в высокое состояние цепь питания триггера замыкается и начинается процесс его регенеративного включения. Если исходный разбаланс превышает возможную асимметрию схемы, она перейдет в состояние, определяемой разницей входных сигналов, и будет сохранять его пока поддерживается высокий уровень  $\phi$ .

В некоторых вариантах, например в схемах усилителей ЗУ, исходное состояние и разбаланс задаются самими входными сигналами (рис.9.5,а). Стробируемые р-МОП M6 и M7 при чтении закрываются и изолируют выходы триггера от входов. В компараторах динамических ЗУ входы и выходы совмещены полностью (M6 и M7 отсутствуют) поскольку выходной сигнал триггера сразу используется для регенерации информации на входе.

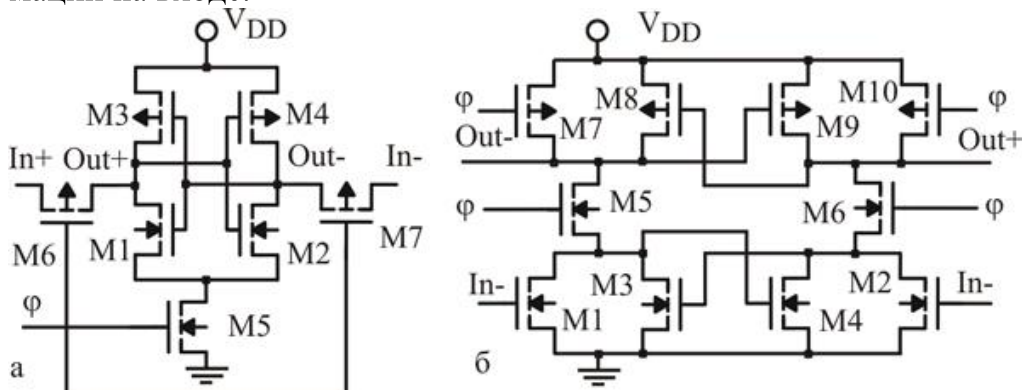


Рис.9.5. Компаратор: а – с совмещением; б – с разделением входа/выхода

В других вариантах среднее состояние триггера задается от источника питания, как показано на рис.9.5,б, где для этого введены МОП M7 и M10. Здесь разбаланс плеч триггера задается при помощи специальных входных транзисторов M1 и M2., что повышает чувствительность компаратора.

Еще большую чувствительность дает дополнительный предусилитель на входе компаратора (рис.9.6,.а). Но, так как его выходные сигналы имеют неполный перепад, он должен быть дополнен выходным RS-триггером (рис.9.6,б).

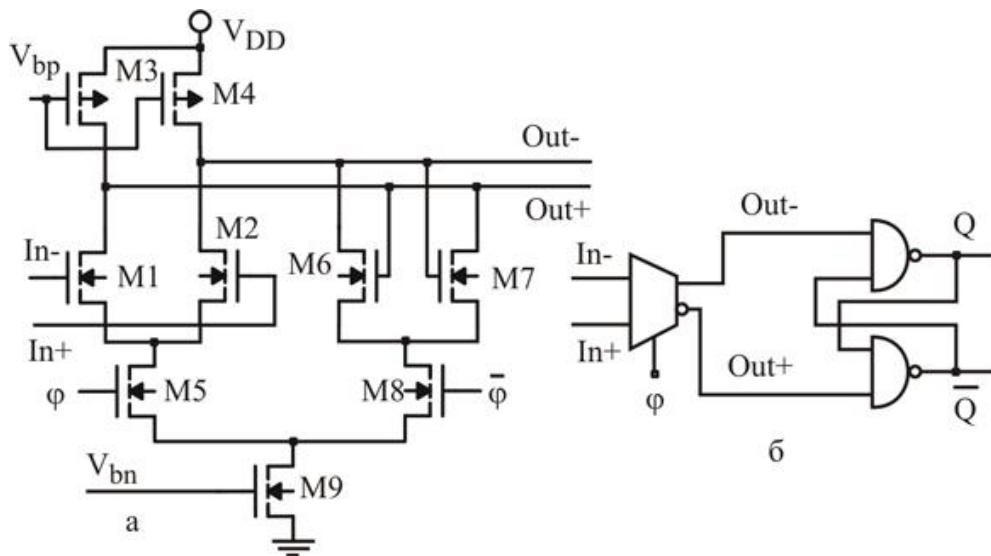


Рис.9.6. Компаратор с предусилителем: а – стробируемый предусилитель; б – блок схема-компаратора с предусилителем

## 9.4. Управляемые генераторы

В цифровых системах большое значение имеют процессы синхронизации работы различных устройств. Для организации этих процессов широко используются различные типы генераторов, в том числе управляемые генераторы. Они находят применение также в преобразователях напряжения.

**Мультивибраторы.** Простейший генератор – мультивибратор – может быть реализован на стандартных вентилях с RC элементами (рис.9.7,а).

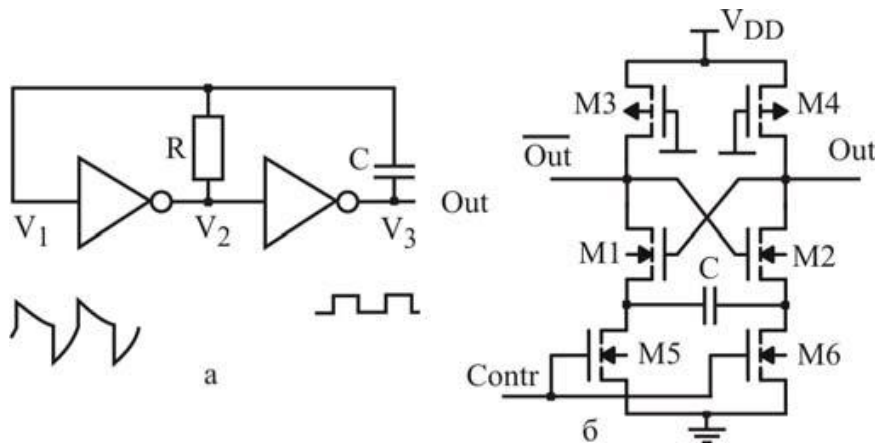


Рис.9.7. Мультивибраторы: а – на вентилях; б – симметричный

Симметричный мультивибратор, допускающий изменение частоты с помощью напряжения на управляющем входе Contr, приведен на рис.9.7,б. Подобный способ с добавлением токового зеркала эффективно применяется в ряде схем.

**Схемы с управляемой задержкой.** Инвертор с симметричным управлением от токового зеркала (рис.9.8,а), может использоваться как управляемая задержка.

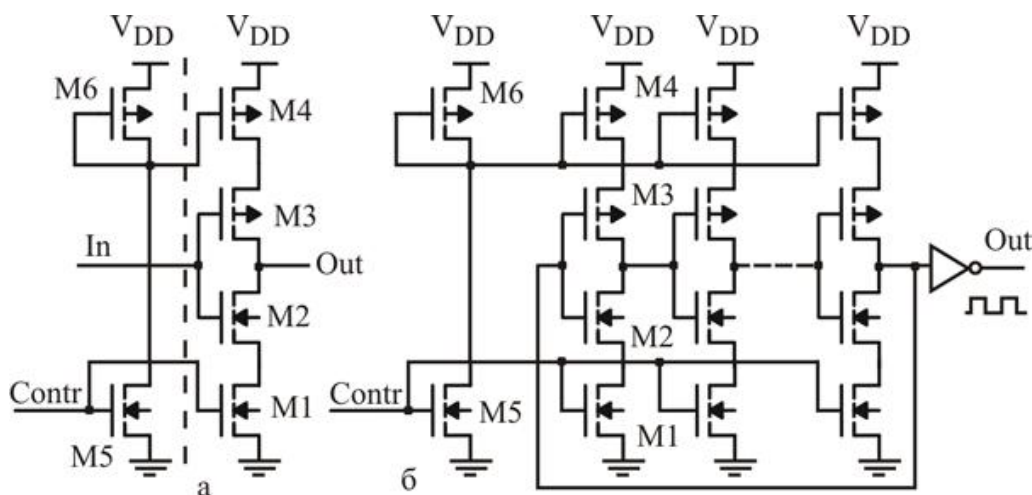


Рис.9.8. Схемы, управляемые напряжением: а – элемент задержки; б – генератор

Нечетное количество таких инверторов, замкнутых обратной связью образует управляемый напряжением кольцевой генератор (рис.9.8,б).



## 9.5. Умножитель напряжения

В некоторых случаях для работы схем требуются напряжения, превышающие напряжение питания. Такая ситуация типична при записи/стирания заряда в транзисторах с плавающим затвором, при возбуждении словарной шины динамического ЗУ и др. Использование диодно-емкостных схем с импульсным возбуждением позволяет решать подобную задачу. В качестве диодов используют n-МОП, если требуется большее положительное напряжение, и p-МОП, если необходимо отрицательное напряжение, меньшее  $V_{SS}$ .

Простейшая схема удвоения (с точностью до  $V_t$ ) напряжения представлена на рис.9.9,а..

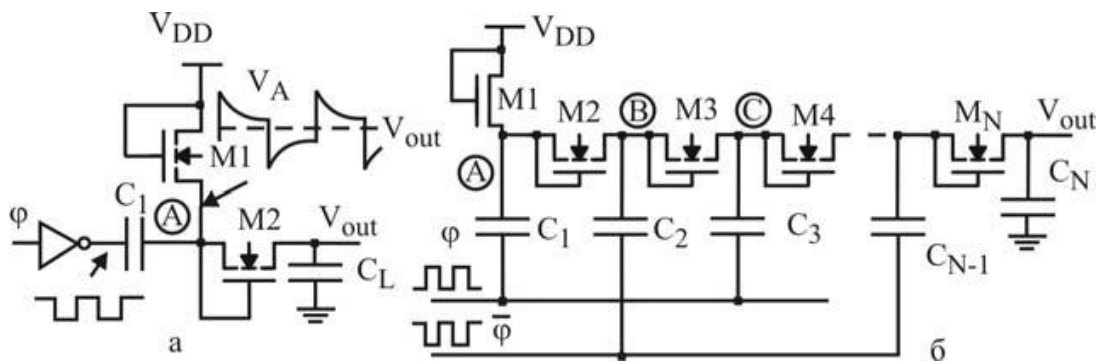


Рис.9.9. Умножитель напряжения: а – однокаскадный; б – многокаскадный

Напряжение в точке А диодом (на M1) привязано к напряжению  $V_{DD} - V_t$ . При положительном фронте с выхода инвертора потенциал в точке А вырастает и через диод M2 передается на емкость нагрузки  $C_L$ . Нарастание потенциала на выходе происходит вплоть до напряжения  $2(V_{DD} - V_t)$ .

Выход однокаскадного удвоителя может использоваться в качестве входа следующего каскада умножителя (рис.9.9,б). При двухтактном его возбуждении в точке В напряжение вырастет до  $2V_{DD} - V_t$ . В каждом последующем каскада напряжение будет вырастать на величину  $V_{DD} - V_t$ . В результате на выходе N каскадного умножителя теоретически можно получить напряжение  $(N+1)V_{DD} - NV_t$ . Реальное повышение будет меньше, особенно в тех случаях, когда МОП-транзисторы не рассчитаны на высокие рабочие напряжения.

## Глава 10. Межсоединения

В работе ИС большую роль играют проводящие межсоединения элементов. Основные электрические требования к проводникам – низкое сопротивление и способность работать при достаточно высоких плотностях тока – обеспечиваются использованием высокопроводящих материалов (алюминий, медь) в сочетании с совершенной технологией их формирования. Высокая степень интеграции современных СБИС во многом достигается за счет применения многослойной системы металлизации. Количество слоев металла в субмикронных ИС уже более шести и продолжает возрастать. Ширина проводников нижних слоев металла и линейные размеры контактов между ними порядка минимального топологического размера. Физический объем межсоединений на кристалле гораздо больше (на порядки) объема, занимаемого активными приборами – МОП-транзисторами.

Межсоединения вносят значительный вклад в основные параметры интегральной схемы: быстродействие, потребляемую мощность, шумы (помехи).

### 10.1 Электрофизические параметры межсоединений

**Сопротивления проводников.** В составе ИС имеется много различных проводящих слоев, включая полупроводниковые слои: диффузионные сток/исток, поликремниевый затвор. Однако, в силу их высокого сопротивления (50 – 400 Ом/□) они в качестве проводников не используются. Те же слои с пленками силицида имеют на порядок более низкое сопротивление (3 – 10 Ом/□) и в отдельных случаях соединяют близко расположенные элементы.

Главные межсоединения выполняются проводниками на основе высокопроводящих металлов. Отдельный проводник характеризуется длиной  $l$ , шириной  $w$  и толщиной  $t$ . В субмикронной технологии для уменьшения погонного (линейного) сопротивления межсоединений применяются проводники с большой толщиной, так что аспектное отношение  $t / w$  больше единицы (1,5 – 2). Зазор между близлежащими проводниками  $s$  и толщина диэлектрика  $h$ , на котором лежит проводник, определяют величину взаимной емкости проводника с другими проводниками.

Сопротивление проводника зависит от его размеров

$$R = \rho \frac{l}{w \cdot t} \quad (10.1)$$

где  $\rho$  – удельное сопротивление проводника.

В интегральных схемах проводники принято характеризовать удельным сопротивлением слоя  $R_{\text{кв}} = \rho / t$  (в Ом/□). В таком случае

$$R = R_{\text{кв}} l / w \quad (10.2)$$

Для технологии 0,25 КМОП удельное сопротивление слоя лежит в интервале от 0,07 до 0,04 Ом/□.

Переход из слоя в слой вносит дополнительное сопротивление – сопротивление контактов, обычно равное от 2 – 20 Ом.

Короткие проводники отличаются относительно небольшими сопротивлениями и слабо влияют на быстродействие схемы. Однако если размеры ИС достаточно велики, то в них имеются и длинные межсоединения. Длинные проводники удобно характеризовать линейным удельным сопротивлением  $R_w = \rho / (t w)$  – сопротивлением на единицу длины, которое составляет 100 – 300 Ом/мм. При длине проводника в несколько миллиметров его сопротивление становится сравнимым с сопротивлением активных приборов.

**Емкости проводников.** Длинные проводники в составе ИС имеют не только высокое сопротивление, но и большую емкость. Так как ширина протяженного проводника обычно не меняется его характеризуют линейной емкостью  $C_w$ , типичная величина которой (для технологии 0,25 КМОП) составляет 0,2 пФ/мм. Она гораздо меньше линейной емкости затвора и стока/истока, но из-за большой длины проводников их емкость сравнима, а иногда и превышает, паразитные емкости активных приборов.

Обычной является ситуация, когда и сверху, и снизу, и по бокам проводника размещены проводящие слои или другие проводники. Величины взаимных емкостей соседних проводников одного слоя того же порядка, что и емкость проводника на проводящий слой, расположенный под (или над) проводником, и должны приниматься в расчет.

## 10.2. Распространение сигналов

**Распределенная линия.** Проводник в составе ИС представляет собой распределенную RC-линию (рис.10.1,а), которую при численном анализе переходных процессов приходится разбивать на секции (рис.10.1,б).

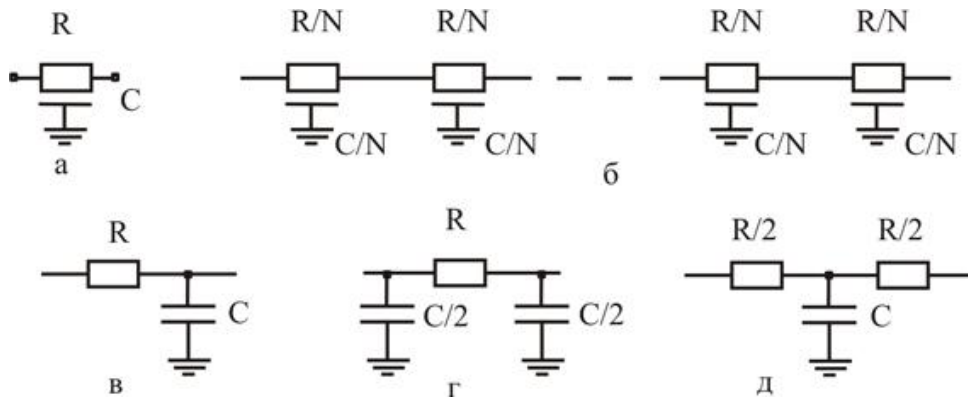


Рис.10.1 Модели распределенной RC линии: а – односекционная; б – многосекционная; в – L-модель; г –  $\pi$ -модель; д – T-модель

Для оценок пользуются простейшими приближениями (рис.10.1, в – 10.1, д). Наиболее адекватные результаты (порядка 3%) дает  $\pi$ -модель (рис.10.1, г), которая сохраняет первые два члена ряда Тейлора точного решения уравнения длинной RC-линии. В  $\pi$ -модели общая погонная емкость линии  $C$  разбивается на две одинаковые емкости  $C/2$ , включенные в общее сопротивление линии  $R$ . Она же обеспечивает максимальное удобство при оценках, так как погонная емкость учитывается путем простого увеличения выходной емкости источника сигнала и входной емкости приемника сигнала.

Согласно  $\pi$ -модели постоянная времени отрезка проводника длины  $l$ , возбуждаемого идеальным источником напряжения, равна

$$\tau = 1/2 \cdot R_w \cdot C_w \cdot l^2. \quad (10.4)$$

Если этот же проводник возбуждается источником сигнала с внутренним сопротивлением  $R_0$  и с выходной емкостью  $C_0$  и нагружен на емкость нагрузки  $C_L$ , то согласно приближению Элмора

$$\tau = R_0 \cdot (C_0 + 1 \cdot C_w / 2) + (R_0 + R_w \cdot l)(1 \cdot C_w / 2 + C_L) \quad (10.4a)$$

При согласованных источнике и нагрузке ( $R_0 = R_w \cdot l$ ,  $C_0 = C_L = 1 \cdot C_w / 2$ ) формула (10.4a) приобретает вид

$$\tau = R_w \cdot C_w \cdot l^2 + 2 \cdot R_w \cdot C_w \cdot l^2 \quad (10.4б)$$

Задаваясь величинами  $R_w = 300$  Ом/мм, и  $C_w = 0.2$  пФ/мм, имеем  $R_w C_w = 60$  пс/мм. Значит, постоянная времени проводника длиной 1 мм, возбуждаемого идеальным источником равна 30 пс, а его постоянная времени при согласованных источнике и нагрузке равна 100 пс. С увеличением длины проводника постоянная времени растет квадратично. Например, при длине проводника 3,2 мм постоянная времени превысит 1 нс.

Заметим, что для согласования источника и нагрузки на концах линии длиной 1 мм должны находиться КМОП-инверторы ( $W_p=2W_n$ ) с шириной канала n-МОП  $W_n \approx 17 \mu\text{м}$ .

### 10.3. Перекрестные помехи

Переходные процессы в системах проводников с емкостными связями могут иметь очень сложный характер. Наиболее важными последствиями взаимного влияния проводников являются изменение скорости переходных процессов и перекрестные наводки (помехи).

**Изменение задержки.** Наличие взаимной емкости  $C_{12}$  между двумя проводниками (рис.10,2.а) изменяет скорость (постоянную времени) переходных процессов и в самом проводнике.

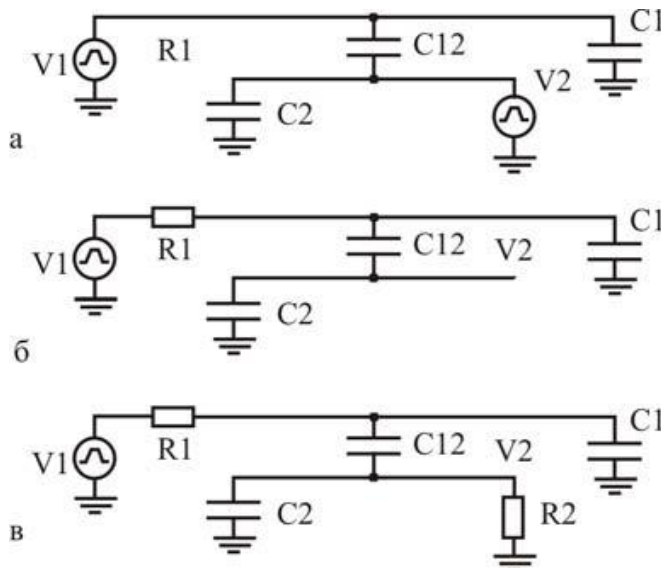


Рис.10.2. Взаимное влияние проводников с емкостной связью.

При этом постоянная времени первого проводника зависит от процессов во втором проводнике. Так, если во время изменения сигнала  $V_1$  сигнал  $V_2$  не меняется, то приращение сигнала  $\Delta V_1$  на первом проводнике заряжает суммарную емкость  $C_1 + C_{12}$ . Если же такое же приращение имеет место и на втором проводнике  $\Delta V_2 = \Delta V_1$ , то приращение сигнала  $\Delta V_1$  заряжает только емкость  $C_1$ , что ускоряет переходной процесс. При противоположном по знаку приращении сигнала на втором проводнике  $\Delta V_2 = -\Delta V_1$ , от источника  $V_1$  требуется вдвое больший ток, что замедляет переходной процесс.

**Перекрестные наводки.** Изменение напряжения на первом проводнике (источнике) из-за емкостной связи приводит к появлению сигнала (наводке) и на втором проводнике (приемнике). В тех случаях, когда

оно не предусмотрено специально, а тем более нежелательно, такой сигнал считают помехой. В сложных системах проводников при наличии большого количества сигналов помехи часто проявляются, как шумы.

Характер наводки и величина наведенного напряжения зависят от режима работы приемного проводника.

Если приемник является проводником с плавающим потенциалом, то наводка на нем определяется емкостным делителем, образованным емкостью связи  $C_{12}$  и емкостью самого проводника  $C_2$  (рис.10.2,б). В таком случае наведенный сигнал  $\Delta V_2$  зависит от сигнала источника  $\Delta V_1$

$$\Delta V_2 = \frac{C_{12}}{C_{12} + C_2} \Delta V_1 \quad (10.5)$$

Если приемник привязан к постоянному потенциалу (или к нулевому потенциалу) через сопротивление  $R_2$  (рис.10.2,в), наводка зависит не только от емкостного делителя, но и от соотношения сопротивлений источника  $R_1$  и приемника

$$\Delta V_2 = \frac{C_{12}}{C_{12} + C_2} \frac{1}{1 + k} \Delta V_1 \quad (10.5a)$$

где

$$k = \frac{R_1(C_1 + C_{12})}{R_2(C_2 + C_{12})} \quad (10.5б)$$

Если приемник – ранее включенный КМОП-вентиль, а источник – переключающийся КМОП-вентиль, то  $R_1 = 2 \div 4 R_2$ .

**Влияние наводок.** Статические КМОП-вентили помехоустойчивы. Емкостная наводка действует на них в течение переходного процесса, а по его окончании ее влияние затухает. Большие емкостные наводки увеличивают время задержки статических вентилях, но не нарушают их конечной реакции. Кроме того, они приводят к дополнительному рассеиванию мощности.

На динамические вентили наводки действуют гораздо сильнее. Емкостная наводка может изменить состояние вентиля, т.е. они потенциально помехоустойчивы.

Последовательностные схемы – триггеры – также подвержены действию помех, хотя и в меньшей степени, чем динамические вентили.

### 10.4. Повторители

Задержка сигнала по длинной линии (рис.10.3,а) пропорциональна квадрату ее длины  $l$ . При передаче сигнала на большие расстояния, особенно в быстродействующих устройствах, задержка может стать недопустимо большой. Для ее снижения естественно разбивать линию на сегменты меньшей длины (рис.10.3,б) и в каждом сегменте ставить драйвер (инвертор или буфер из двух инверторов), ускоряющий передачу сигнала. Оптимальное по суммарной задержке разбиение длинной линии зависит как от параметров проводников, так и от параметров МОП-транзисторов.

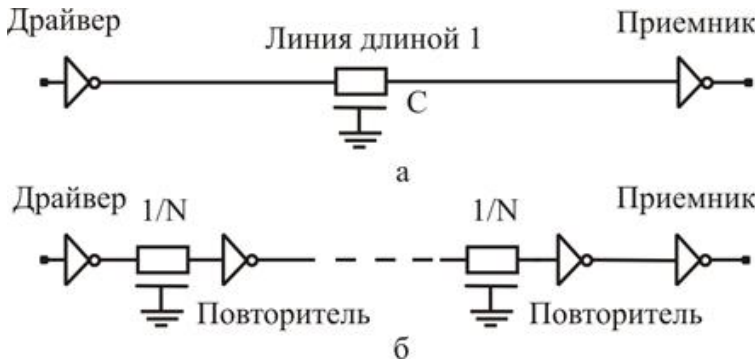


Рис.10.3. Длинная линия: а – одиночная линия; б – секционированная линия

Пусть линия длиной  $l$  разбита на  $N$  сегментов, в каждом из которых стоит драйвер с шириной затвора  $W$ . В таком случае сопротивление проводника сегмента равно  $R_w l / N$ , емкость проводника сегмента  $C_w l / N$ , выходное сопротивление драйвера  $R_g / W$ , а его входная и выходные емкости  $C_g W$ . Общая задержка сегментированной линии (по аналогии с формулой (10.4а)) равна

$$\tau_{\Sigma} = \left[ \frac{R_g}{2W} (C_g W + \frac{C_w l}{2N}) + (\frac{R_g}{2W} + \frac{R_w l}{N}) (\frac{C_w l}{2N} + C_g W) \right] N. \quad (10.6)$$

Из равенства нулю производной  $\tau_{\Sigma}$  по  $N$  можно определить оптимальное количество каскадов. Минимум задержки имеет место в случае

$$N = l \cdot \sqrt{\frac{R_w C_w}{2 R_g C_g}}. \quad (10.7)$$

Из равенства нулю производной  $\tau_{\Sigma}$  по  $W$  можно определить оптимальные размеры драйвера:

$$W = \sqrt{\frac{R_g C_w}{R_w C_g}}. \quad (10.8)$$

Подставляя (10.7) и (10.8) в (10.6) имеем значение минимальной задержки

$$\tau_{\Sigma \min} = 1 \cdot (2 + \sqrt{2}) \sqrt{R_g C_g R_w C_w} . \quad (10.9)$$

Величина задержки порядка 50 пс/мм, т.е. примерно совпадает с полученной в предыдущих оценках. Но эта задержка с возрастанием длины проводника растет линейно, а не квадратично.

Эффекты распространения (задержки) сигнала по линии существенны только для импульсов с фронтом (спадом) короче времени его распространения

$$t_r \ll 2,5 l / c,$$

где  $c$  – скорость распространения сигнала по линии.



## Глава 11. Методология проектирования

Современная кремниевая технология позволяет на одном кристалле разместить сложную микроэлектронную систему. Ее проектирование предполагает решение ряда задач с широким спектром целей, достижение которых требует определенных компромиссов. Наряду с выполнением заданных функций система должна удовлетворять ряду технических требований при используемых технологических возможностях. Общими требованиями к интегральным схемам (и системам) являются:

- уменьшение площади;
- повышение быстродействия (эффективности);
- снижение потребляемой мощности;
- упрощение тестирования;
- повышение надежности;
- снижение стоимости, состоящей из затрат на проектирование, освоение изготовления и непосредственно на производство.

С точки зрения проектировщика, существует два принципиально различных класса микроэлектронных изделий: серийные микросхемы широкого применения (например, микропроцессоры, ЗУ) и специализированные микросхемы самого разного назначения, включая системы на кристалле.

Серийные изделия, выпускаемые в массовых объемах, обладают высокой производительностью, имеют низкую стоимость, фактически являются отраслевыми стандартами. На их оптимальное проектирование тратятся значительное время и большие средства, которые окупаются за счет высокой прибыли при массовом спросе.

Специализированные микросхемы позволяют добиваться уникальных характеристик систем, которые недостижимы на стандартных микросхемах. Однако они должны разрабатываться за короткое время, при меньших затратах и с учетом объемов производства. Вследствие этого при их разработке используются различные методы ускорения (и удешевления) проектирования.

Специализированные микросхемы разрабатываются, как полузаказные, а в их составе могут быть различные блоки, в том числе:

- заказные схемы;
- на стандартных ячейках (элементах);
- на вентильных матрицах;
- с программируемыми соединениями.

Одно из основных различий при проектировании этих блоков состоит в степени свободы выбора при разработке топологии.

Для заказных схем свобода ограничивается только правилами проектирования топологии.

В блоках на стандартных ячейках ограничена их высота и месторасположение соединений.

В блоках на вентиляльных матрицах топология элементов строго задана и проектируются межсоединения (иногда только контакты между проводниками).

В схемах с программируемыми соединениями топология задана полностью, а в процессе настройки пользователь создает требуемые соединения (или устраняет ненужные).

Стандартные ячейки могут представлять собой специально или ранее разработанные типовые блоки (макроячейки), а также схемы произвольной логики.

### 11.1. Символьная топология

Процесс разработки топологии заказных схем с произвольной логикой является “узким местом” в проектировании СБИС.

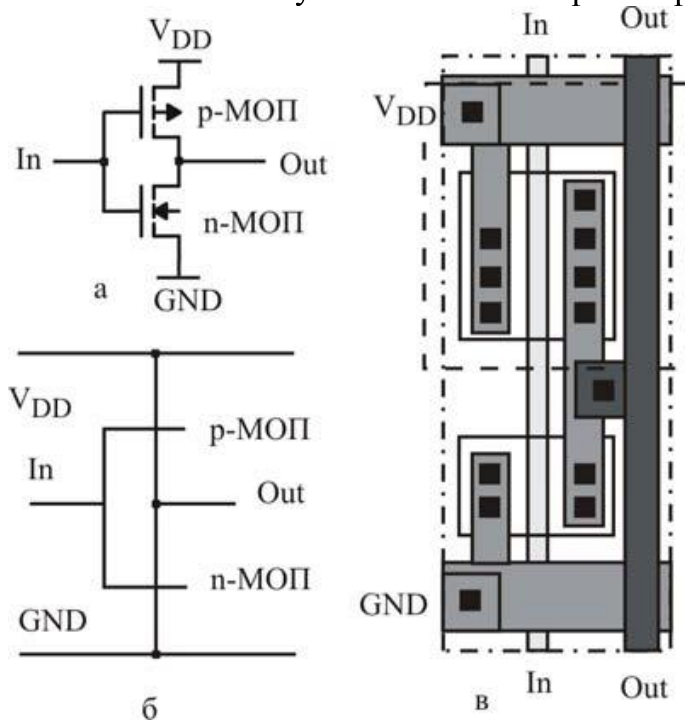


Рис.11.1. Инвертор: а – электрическая схема; б – символьная топология; в – полная топология

Для его ускорения применяют ряд методов, получивших общее название метода символьной топологии. При этом разработчик проектирует не непосредственно топологию, а ее эскиз, качественно отражающий взаимное размещение топологических элементов (рис.11.1). Полная топо-

логия реализуется при обработке символьной топологии специальной программой.

Символьная топология весьма уместна при выборе конфигурации сложных схем, поскольку позволяет достаточно просто анализировать возможные их варианты, что показано на примере инвертора на рис.11.2.

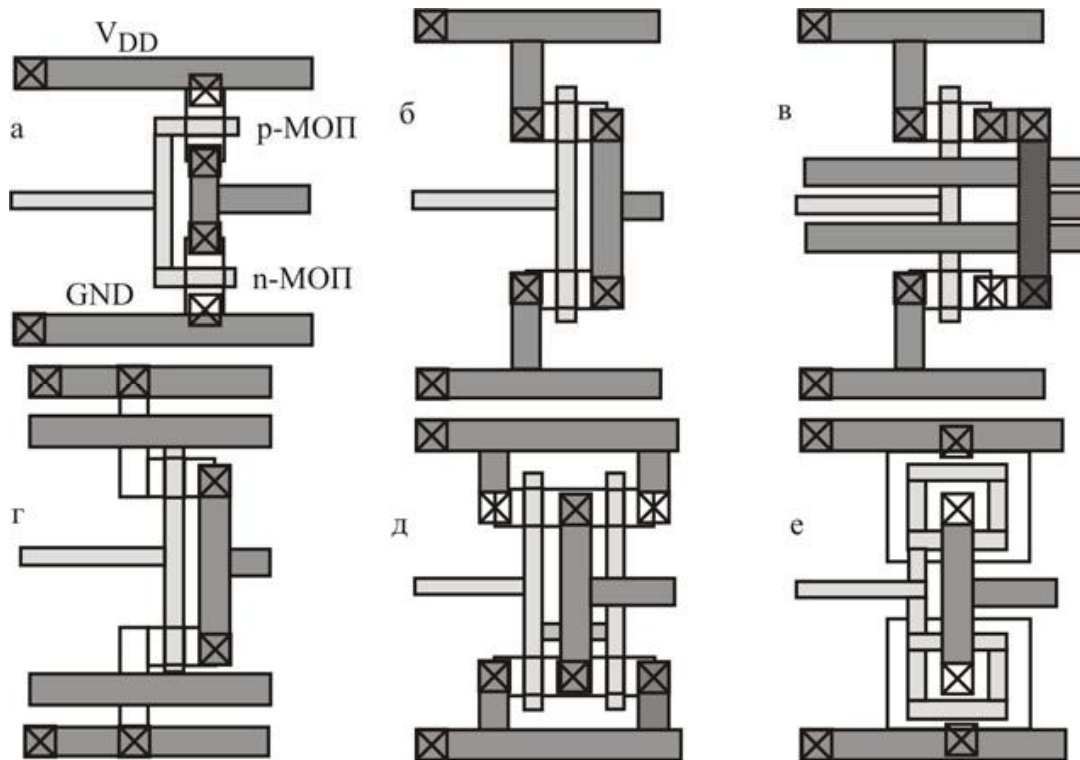


Рис.11.2. Варианты топологии инвертора

Очевидно, что n-МОП и p-МОП могут быть размещены вертикально (рис.11.2,а) и горизонтально (рис.11.2,б). Размещение других проводников над инвертором может выполняться либо между n-МОП и p-МОП (рис.11.2,в), либо между n-МОП и общей шиной GND, либо между p-МОП и шиной питания  $V_{DD}$  (рис.11.2,г). Увеличение ширины канала достигается добавлением параллельных транзисторов (рис.11.2,д) или изменением их формы (переход к кольцевым МОП) (рис.11.2,е).

Количество конфигураций сложных элементов растет очень быстро. Поэтому даже в элементах произвольной логики стремятся перейти от произвольного размещения компонентов (рис.11.3,а) к упорядоченному (рис.11.3,б).

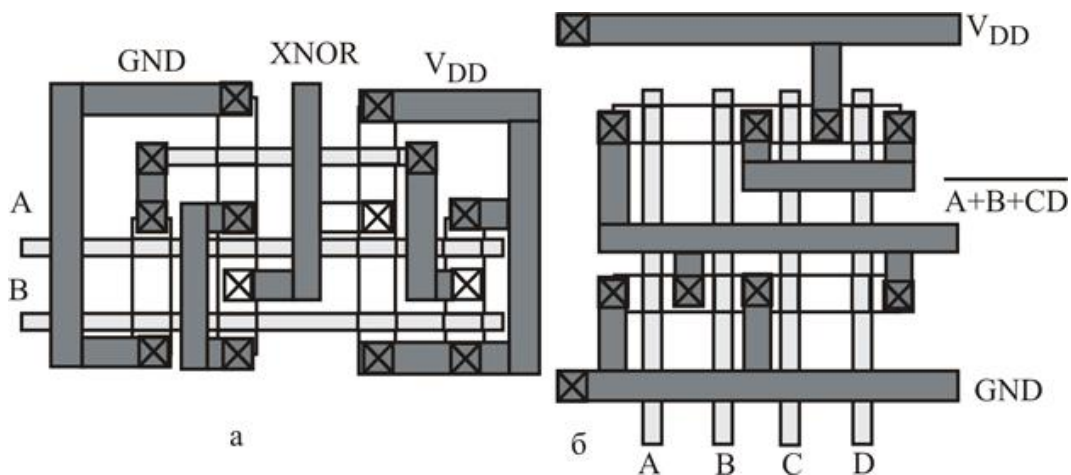


Рис.11.3. Топология вентилей: а – произвольное размещение; б – стандартные ячейки

Введение рядов для размещения элементов – переход к стандартным ячейкам – является первым шагом на этом пути.

## 11.2. Стандартные ячейки

Библиотека стандартных ячеек содержит набор схем с компактной топологией, у которых одинаковая высота, а ширина может быть различной (рис.11.4). Ограничение на высоту ячейки не препятствует оптимизации размеров транзисторов в схеме. Межсоединения ячеек должны размещаться в определенных местах, чтобы облегчить их взаимную коммутацию.

Основная проблема проектирования на стандартных ячейках заключается в определении маршрутов соединений.

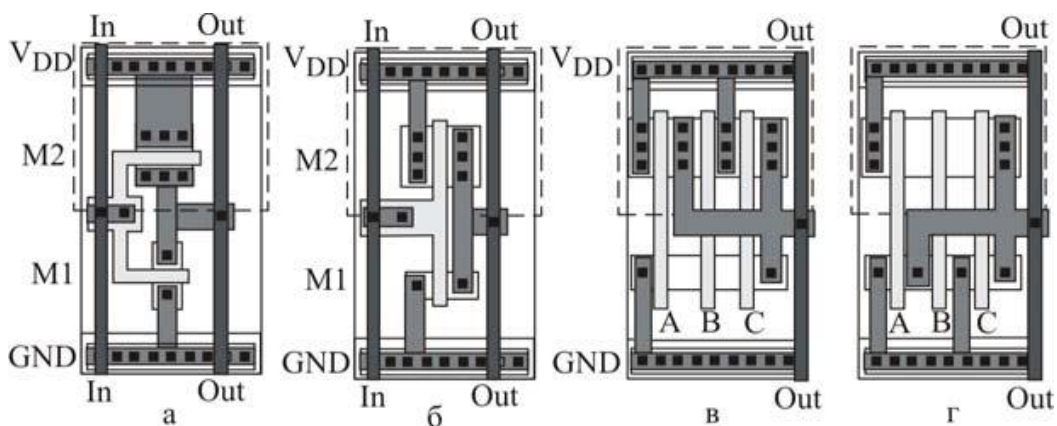


Рис.11.4. Топология стандартных ячеек.

Стандартные ячейки размещаются рядами, а зазоры между рядами используют для трасс межсоединений. В такую структуру легко вписываются и макроячейки (рис.11.5).

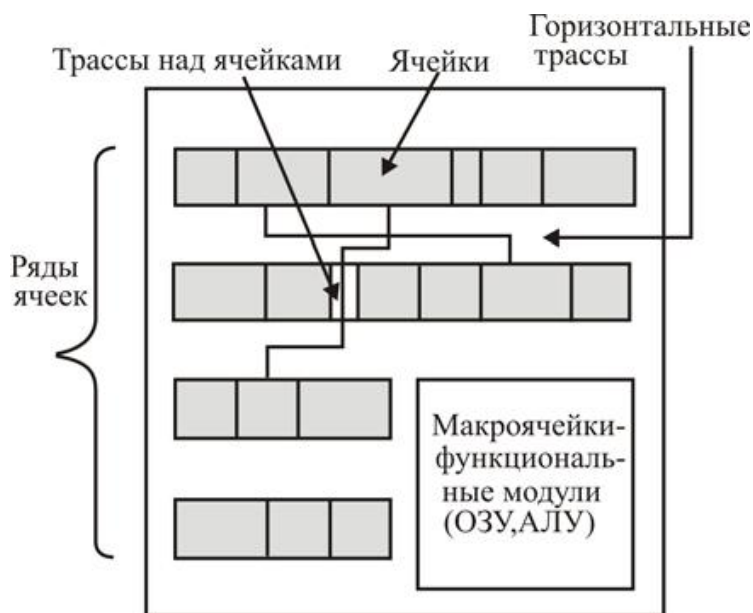


Рис.11.5. Стандартные ячейки и макроячейки

### 11.3. Вентильные матрицы.

Вентильная матрица представляет собой однородную структуру, состоящую из несоединенных между собой вентилях (или заготовок для них), расположенных в виде матрицы (рис.11.6). Вентильные матрицы, занимающие весь кристалл, называют базовыми матричными кристаллами (БМК).

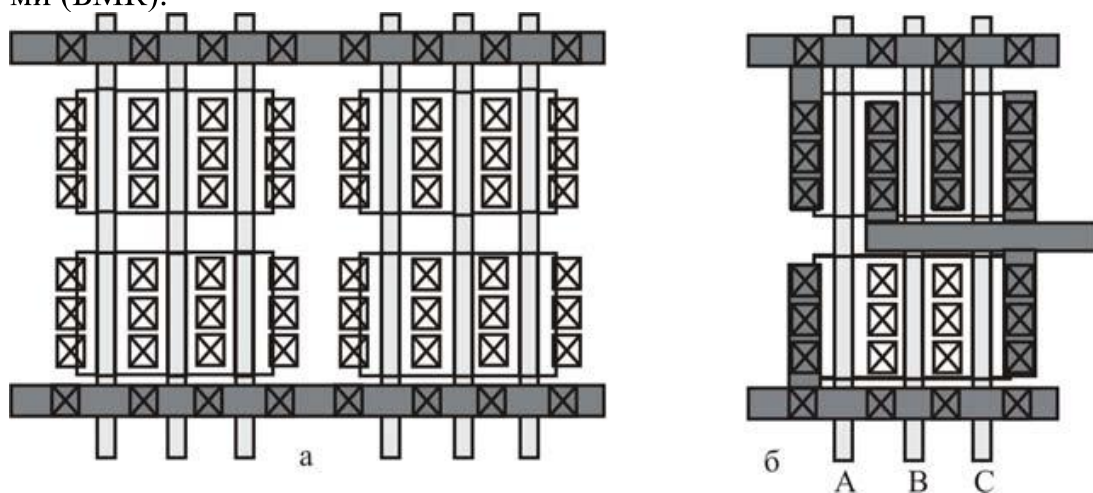


Рис.11.6. Матрица вентилях: а – без коммутации; б – с коммутацией

Иногда в виде матрицы расположены несоединенные транзисторы (рис.11.7), получившие название «море вентилях».

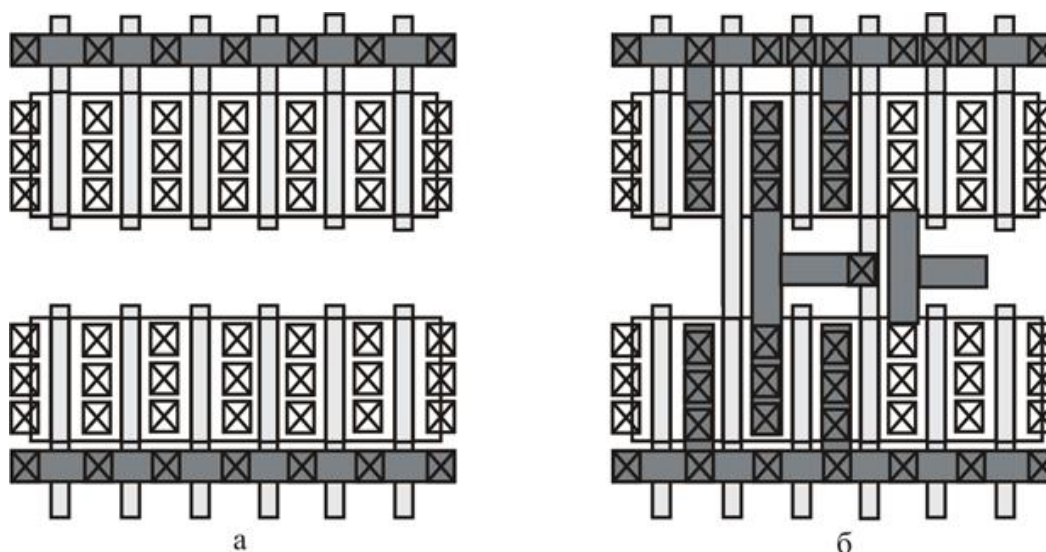


Рис.11.7. Море вентилей: а – без коммутации; б – с коммутацией

В обоих случаях необходимо задать тип вентиля подсоединением транзисторов к шинам питания и топологию межсоединений образованных вентилей.

Поскольку соединять нужно уже размещенные вентили, эффективным оказывается применение САПР, что значительно сокращает время проектирования. Таким образом, по сравнению с полным проектированием на заказ проектирование с использованием матриц вентилей является менее дорогим и более быстрым.

## 11.4. Программируемые матрицы

Однородные матричные структуры позволяют достигать высокой удельной (на единицу площади) производительности во многих задачах, где не требуются сложные итерационные алгоритмы. Кроме того, они проще в разработке и тестировании. Поэтому они находят широкое применение в интегральных схемах. К таким схемам относятся ПЗУ, в которых в качестве элемента используется всего один транзистор предельно малых размеров. Наряду с ПЗУ применяются другие матричные структуры. На рис.11.8 приведены три основные матричные структуры:

- 1) программируемая логическая матрица – ПЛМ (PLA);
- 2) постоянное запоминающее устройство – ПЗУ (PROM);
- 3) программируемая матричная логика – ПМЛ (PAL).



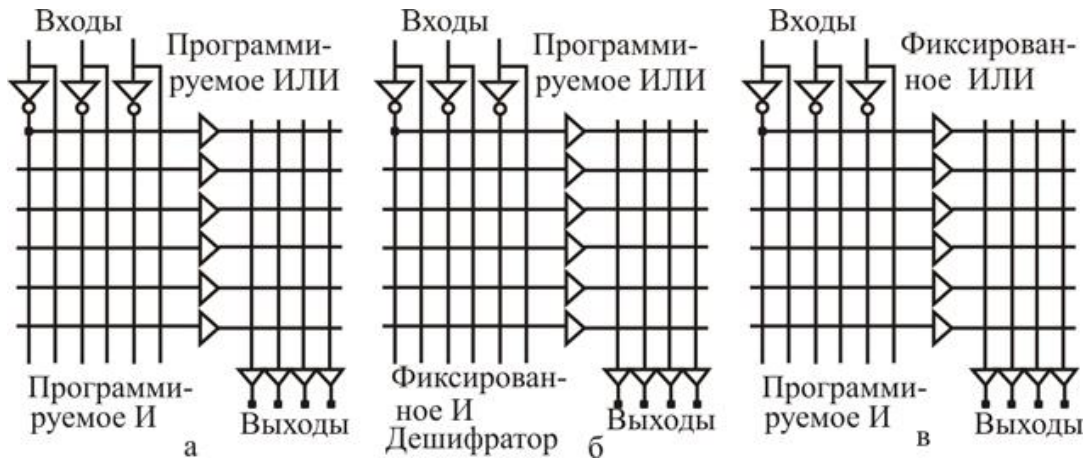


Рис.11.8. Типы программируемых матриц: а – ПЛМ; б – ПЗУ; в – ПМЛ

Каждая из структур содержит две матрицы, выполняющие проводные функции. Первая матрица выполняет функцию И, вторая – функцию ИЛИ. При этом:

в ПЛМ в обеих матрицах могут быть записаны (запрограммированы) произвольные комбинации, т.е. ее выходы выполняют произвольные логические функции над входами в стандартной форме (дизъюнкцию конъюнкций);

в ПЗУ первая матрица является дешифратором, а вторая содержит произвольные коды;

в ПМЛ в первой матрице произвольные комбинации, а в строках второй матрицы только одно соединение, т.е. она выполняет более узкие функции, чем ПЛМ тех же размеров.

### 11.5. Программируемые пользователем матрицы

Широкое распространение получили матричные структуры с программируемыми пользователем соединениями. Современная технология позволяет получать как однократные (создаваемые за счет электрического пробоя диэлектрика – “программируемые полем”) соединения, так и соединения с помощью ключей, управляемых запоминающими элементами. Причем сами элементы памяти могут быть и в виде триггеров, и в виде запоминающих МОП-транзисторов.

Программируемые полем логические матрицы содержат ряды ячеек, соединенных каналами (рис 11.9).



# Trench Termination Design and Analysis in Low-Voltage N-Channel Trench Power Metal–Oxide–Semiconductor Field-Effect Transistor

Chorng-Wei LIAW\*, Leaf YEH<sup>1</sup>, Ming-Jang LIN<sup>1</sup>, and Chrong Jung LIN

*Microelectronics Laboratory, Semiconductor Technology Application Research (STAR) Group,*

*Department of Electrical Engineering, National Tsing Hua University, Hsinchu 300, Taiwan, R.O.C.*

<sup>1</sup>*ANPEC Electronics Corporation, No. 6, Dusing 1st. Rd., Hsinchu 300, Taiwan, R.O.C.*

(Received May 18, 2007; accepted November 27, 2007; published online March 14, 2008)

In this study, a novel termination is designed in a low-voltage N-channel trench power metal–oxide–semiconductor field-effect transistor (MOSFET) to simplify the fabricating process. In a conventional trench power-MOSFET (PowerMOS), a field oxide with a metal field plate is often used for edge termination. The field oxide only exists in the termination region of the trench PowerMOS; therefore if the termination can be designed without the field oxide, field oxidation can be removed from the fabrication. Trench termination is proposed in this work to replace field oxide termination. The use of multiple trench rings leads to the desired breakdown voltage. The temperature effect in trench termination must be considered owing to the negative coefficient of the threshold voltage ( $V_{TH}$ ) of the P-type MOS (PMOS) and the temperature. This work also provides a design guideline for trench termination in the low-voltage trench PowerMOS. [DOI: 10.1143/JJAP.47.1507]

KEYWORDS: termination, trench, PowerMOS, metal field plate, body effect

## 1. Introduction

Recently, much attention has been given to low-voltage power metal–oxide–semiconductor field-effect transistor (MOSFET) (PowerMOS) because of the numerous portable devices on the market and related applications.<sup>1)</sup> The cost of fabricating a trench PowerMOS is very important. Current studies on the low-voltage trench PowerMOS have focused on the reduction of device pitch by shrinking the design rules, including trench width, source contact size and trench-to-source contact spacing.<sup>2–5)</sup> Although such engineering improves the performance, such as specific on-resistance ( $R_{ON,SP}$ ) and gate charge ( $Q_G$ ), a termination formed by the field oxide and the metal field plate between the cell region and the chip scribe line must exist to sustain the breakdown voltage.<sup>6–9)</sup> The field oxide is necessary only for the termination design and can be removed if a new termination structure without changing the fabrication process steps. The termination structure is verified by simulations and experiments with a 16 V N-channel trench PowerMOS process. The field oxidation, lithography of field oxide and field oxide etching processes can be removed to simplify the fabrication if the trench termination structure is applied. The temperature effect of trench termination on breakdown characteristics is also discussed. Chip size is an important factor when trench termination is designed in a trench PowerMOS. This work provides a trench termination design method to determine how many trenches are necessary in the termination design.

## 2. Termination Structure and Simulation

The 0.4  $\mu\text{m}$  16 V N-channel trench PowerMOS technology was used to simulate the proposed termination structure. Figure 1(a) shows the cross-sectional view of the trench termination concept. The active cell is on the left side, the bottom of the structure is the drain terminal and the scribe line is on the right side. There are three trenches between the source terminal and the scribe line. The P-well can be

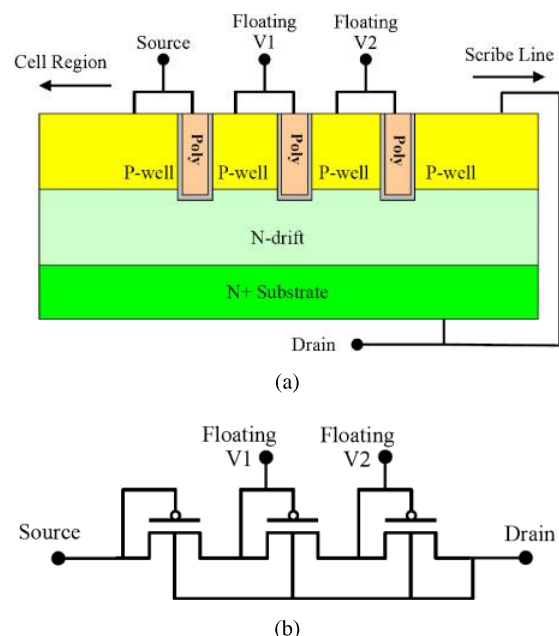


Fig. 1. (Color online) Device structure and equilibrium circuit: (a) cross-sectional view of proposed termination concept and (b) equilibrium circuit of trench termination in (a).

defined as the source or the drain of P-type MOS (PMOS), the trench bottom is the channel of PMOS, the N-drift region is the body of PMOS and the poly inside the trench is the gate terminal of PMOS. The trench poly is electrically connected to its left-side P-well silicon. This structure is similar to a three series diode-connected PMOS with the same N-body bias and is schematically shown in Fig. 1(b). If a trench termination is applied in a trench PowerMOS, the breakdown current comes from two paths: one is the source P-well to the N-drift junction avalanche breakdown leakage current, and the other is the series diode-connected PMOS conducting current from the scribe line P-well to the source P-well. The terminal names in Figs. 1(a) and 1(b) are defined for the NMOS structure; the PMOS source and drain

\*E-mail address: cwliaw@gmail.com

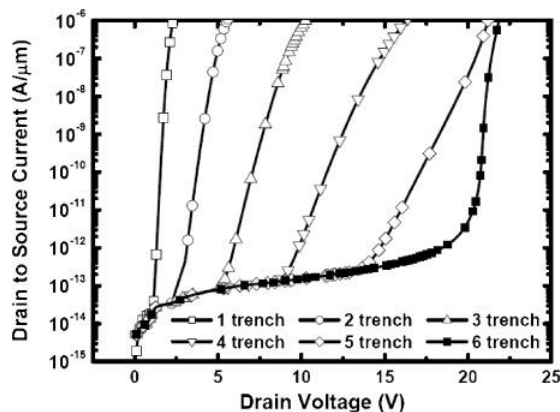


Fig. 2. Simulation results of trench terminations. The breakdown voltage increases when the number of trenches increases.

of edge termination are of opposite types from those of the NMOS. Figures 1(a) and 1(b) only show the concept of the trench termination structure; the trench numbers should be designed to block the expected voltage from the scribe line P-well to the source terminal. The blocking voltage of series PMOS is the sum of  $V_{TH}$  of each PMOS at a specific current, which is defined as breakdown current. The  $V_{TH}$  values of each PMOS are not the same because the bodies to source voltages of each PMOS are different. Different  $V_{BS}$  causes diverse body effects which results in different  $V_{TH}$ .

Figure 2 shows simulation results of six trench termination structures. The breakdown voltage increases with increasing trench numbers. In the simulation structure, the applied voltage of the P-well at the scribe line is equal to that at the drain terminal. The termination with a single trench shows the diode-connected PMOS characteristic. The  $V_{TH}$  of the PMOS is about 1.7 V. When the trench number increases to six, the conduction current of a six series diode-connected PMOS is smaller than the source P-well to the N-drift junction leakage current at a drain voltage of 16 V. The simulation result shows that the trench termination designed with six trenches can sustain over 16 V of drain voltage.

Figure 3 shows the potential contours of the six-trench termination structure when the drain voltage was 16 V. The depletion region boundary is extended smoothly from the source P-well to the scribe line P-well. The smooth potential distribution suppresses the electrical field under the trench corner and keeps the breakdown voltage as that of high as the planar P-well/N-drift junction's avalanche breakdown voltage.

### 3. Experimental Results and Discussion

A 0.4  $\mu\text{m}$ , 16 V N-channel trench PowerMOS technology was used to fabricate the trench termination structures. The Keithley 4200-SCS is used to measure the current-voltage curve of termination breakdown. Figure 4 shows the experimental results for trench termination structures. The experimental results for the six-trench termination structures are very similar to those of the simulation. In the six-trench design, three phenomena can be identified from the current-voltage ( $I$ - $V$ ) curve. When the drain voltage is lower than the value at point A, the current comes from the source P-well

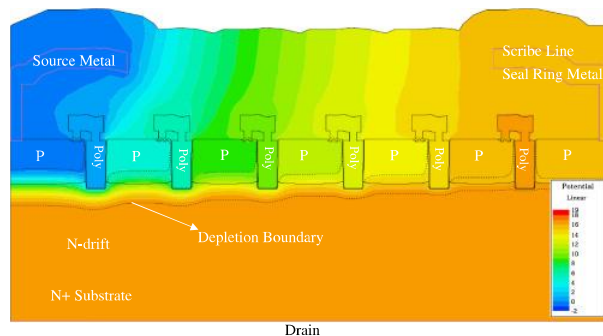


Fig. 3. (Color online) Potential contours of termination with six trenches. The depletion in the N-drift region is smoothly extended from the source to the scribe line.

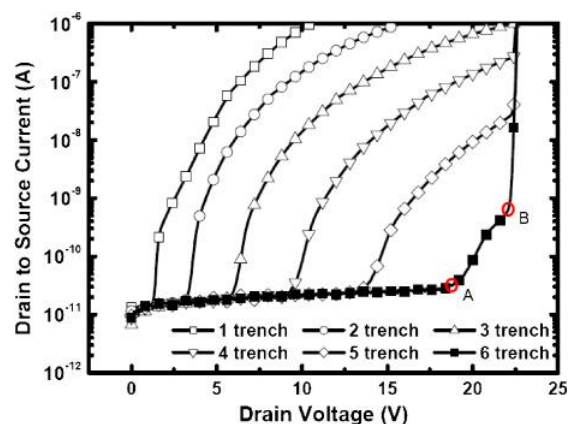


Fig. 4. (Color online) Experimental results of trench terminations. The termination with six trenches can sustain drain voltage of over 16 V.

to N-drift junction leakage current. As the drain voltage exceeds the value at point A, the six-diode-connected PMOS starts to conduct, and the current increases as drain voltage increases. When the drain voltage is larger than the value at point B, the source P-well to N-drift junction avalanche breakdown occurs and a large avalanche current flows from the drain to the source.

The limitation of trench termination is the gate oxide breakdown of two nodes. A six-trench termination testkey was designed to verify if the voltage difference between two nodes exceeds the gate oxide safe operating voltage, and the measured result is shown in Fig. 5. In the figure, the relationship between the five-floating-node voltage and the drain voltage can be observed.  $V_1$  is the first floating node from the source terminal and can be defined as the source of PMOS1. The drain and the gate of PMOS1 are shorted to ground and the body of PMOS1 is the N-drift region, which is the drain terminal of the termination. Owing to the highest reverse bias of the source to the body of PMOS1,  $V_{TH}$  of PMOS1 is the largest among the six series PMOS. If  $V_1$  is lower than the gate oxide safe operating voltage, the voltage differences of the other floating nodes are all sufficiently low in normal operation. In this design, the gate oxide thickness is 250  $\text{\AA}$  and 10 V is a safe voltage to apply on the oxide.  $V_1$  is 5 V when the drain voltage is 20 V; it is very safe to apply the trench termination structure in this technology.

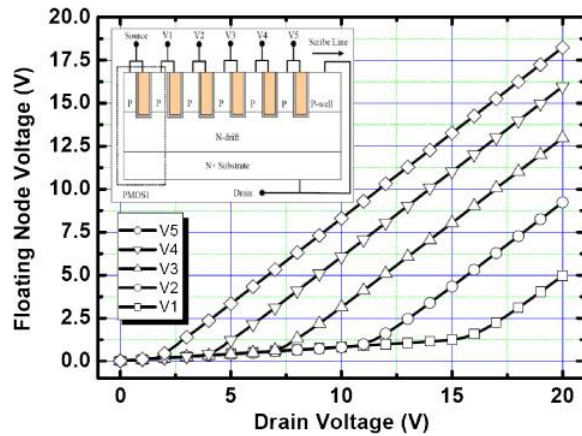


Fig. 5. (Color online) Relationship between floating node voltage and drain voltage. The voltages of all floating nodes increase when the drain voltage increases.

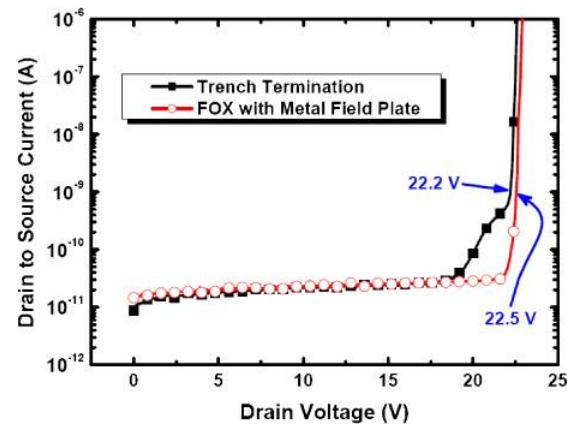


Fig. 6. (Color online) Breakdown characteristics of trench termination with six trenches and field oxide with metal plate. The breakdown voltage of trench termination is lower than that of field oxide termination.

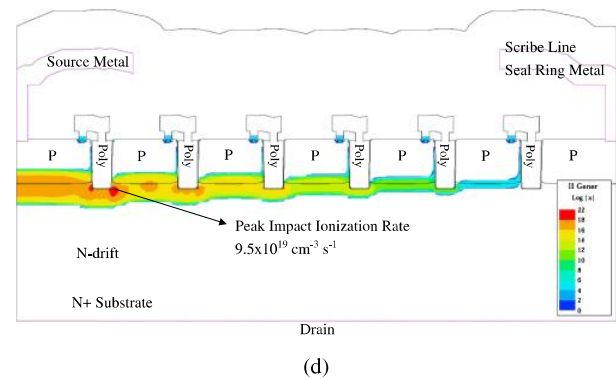
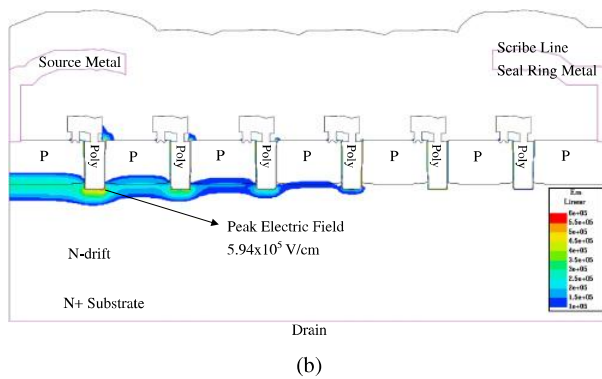
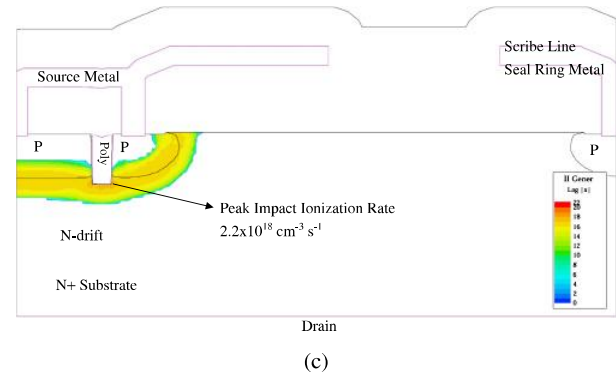
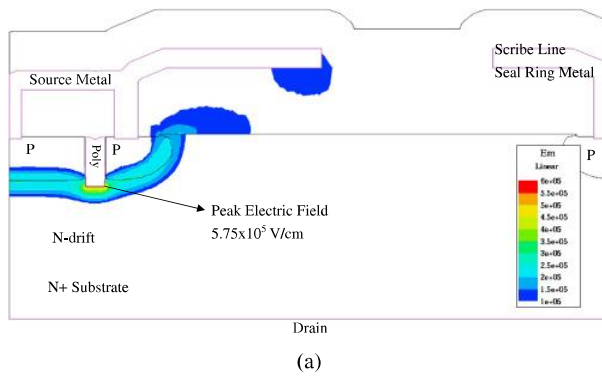


Fig. 7. (Color online) (a) Electric field of field oxide with metal field plate termination. (b) Electric field of trench termination with six trenches. (c) Impact ionization rate of field oxide with metal field plate. (d) Impact ionization rate of trench termination with six trenches.

On comparing the trench termination structure and conventional field oxide with metal field plate termination, the breakdown voltage of trench termination is found to be lower than that of conventional termination. Figure 6 shows the measured breakdown characteristic of the two termination structures. The avalanche breakdown voltages of the trench termination and the conventional termination are 22.2 and 22.5 V, respectively. The slightly lower breakdown

voltage of trench termination is caused by the series PMOS conducting current from the scribe line P-well to the source P-well. The channel hole current becomes the source P-well to N-drift junction avalanche seed current, which decreases the breakdown voltage. Figures 7(a) and 7(b) show that the electrical fields of the two termination structures are very similar. Figures 7(c) and 7(d) show the impact ionization rate contours of the two termination structures. When the



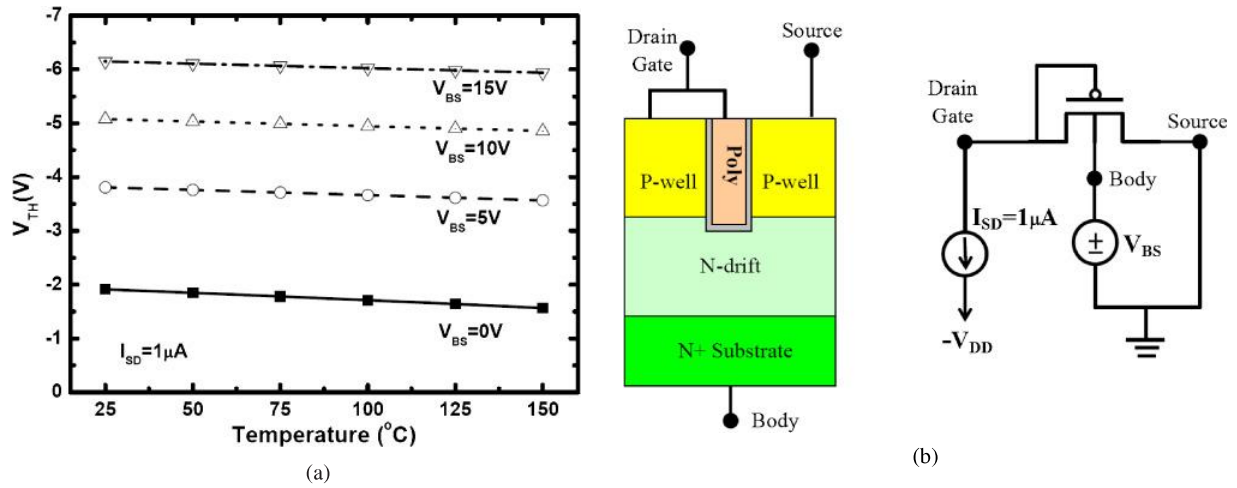


Fig. 8. (Color online) (a) Threshold voltage of PMOS with different body-to-source voltages and temperatures. (b) Device structure and measurement setup for PMOS  $V_{TH}$ .

drain voltage is 16 V, the impact ionization rate of trench termination is higher than that of conventional termination. This explains why the breakdown voltage of trench termination is slightly lower than that of conventional termination.

The temperature effect is very important to a power device. The blocking voltage of trench termination is the sum of all series PMOS  $V_{TH}$  on a specific current. It may cause a problem when the series PMOS is operated in a high-temperature environment, because of the negative coefficient of PMOS  $V_{TH}$  versus temperature. Figure 8(a) shows that the measured  $V_{TH}$  of a single PMOS is a function of temperature for different  $V_{BS}$  with  $I_{SD}$  of 1  $\mu A$ . The device structure and measurement setup are shown in Fig. 8(b). Figure 8(a) shows that  $V_{TH}$  of PMOS decreases slightly as the temperature increases for different  $V_{BS}$ . The leakage current of the series PMOS termination increases when the temperature increases as a result of decrease of  $V_{TH}$  of each PMOS. Figure 9 shows the breakdown curves for different temperatures. These  $I$ - $V$  curves can be separated into three parts. The P-well/N-drift junction leakage increases with temperature in Part I; it is the same as the normal junction leakage mechanism. In Part II, because of series PMOS conduction, the starting points move to lower drain voltage owing to the  $V_{TH}$  decrease of series PMOS when the temperature increases. Part III is dominated by the P-well/N-drift junction avalanche; the avalanche breakdown voltage increases with temperature owing to lattice scattering.

Chip size must be considered if a trench termination is designed in a trench PowerMOS. The blocking voltage is dominated by  $V_{TH}$  of PMOS; therefore, the channel width of PMOS should be considered. The channel width of PMOS is the circumference of the chip size. To design a trench termination, the only factor that needs to be considered is how many trench numbers could provide the desired blocking voltage. Figure 10(a) is the termination structure that is assumed to be designed. The PMOS closest to the scribe line is named PMOS<sub>N</sub>. PMOS<sub>N-1</sub> is the second PMOS from the scribe line toward the source region. For a 6000-

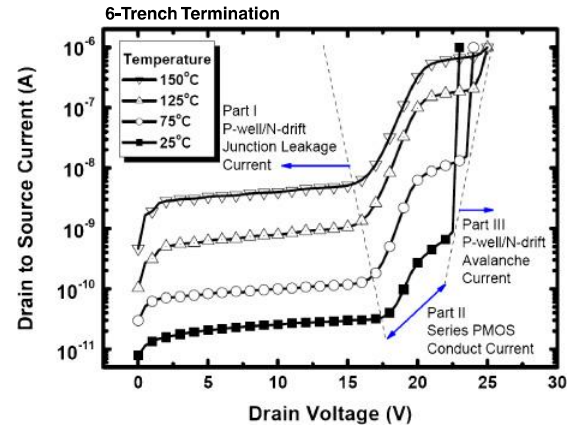


Fig. 9. (Color online) Breakdown characteristics of trench termination with six trenches at different temperatures.

$\mu m$ -circumference 16 V N-channel trench PowerMOS to be designed to have leakage current less than 10  $\mu A$  at 150 °C when  $V_{DS}$  is 16 V, Fig. 10(b) provides a design guideline to determine how many trenches should be included in the termination. Figure 10(b) is the PMOS  $V_{TH}$  measured with different  $V_{BS}$  at 150 °C when the  $I_{SD}$  was 1  $\mu A$ , and the PMOS channel width was 600  $\mu m$ . Because the circumference of the design target is ten times the PMOS channel width of the measured device, the measured current should be divided by ten to realize the desired leakage current. The source of PMOS<sub>N</sub> in Fig. 10(a) is the scribe line P-well and is shorted to the substrate. Therefore, the body-to-source voltage of PMOS<sub>N</sub> is 0 V.  $V_{TH}$  of PMOS<sub>N</sub> is  $-1.56$  V because  $V_{BS}$  is 0 V.  $V_N$  is the value of  $V_{TH(N)}$ , which is the source voltage of PMOS<sub>N-1</sub>.  $V_{BS}$  of PMOS<sub>N-1</sub> is equivalent to  $V_N$ , and  $V_{TH(N-1)}$  can be determined from the Fig. 10(b). By the same sequence,  $V_{N-X}$  can be determined. When the absolute value of  $V_{N-X}$  is larger than 16 V, the value of  $X$  is the number of trenches that should be included in the trench termination design.

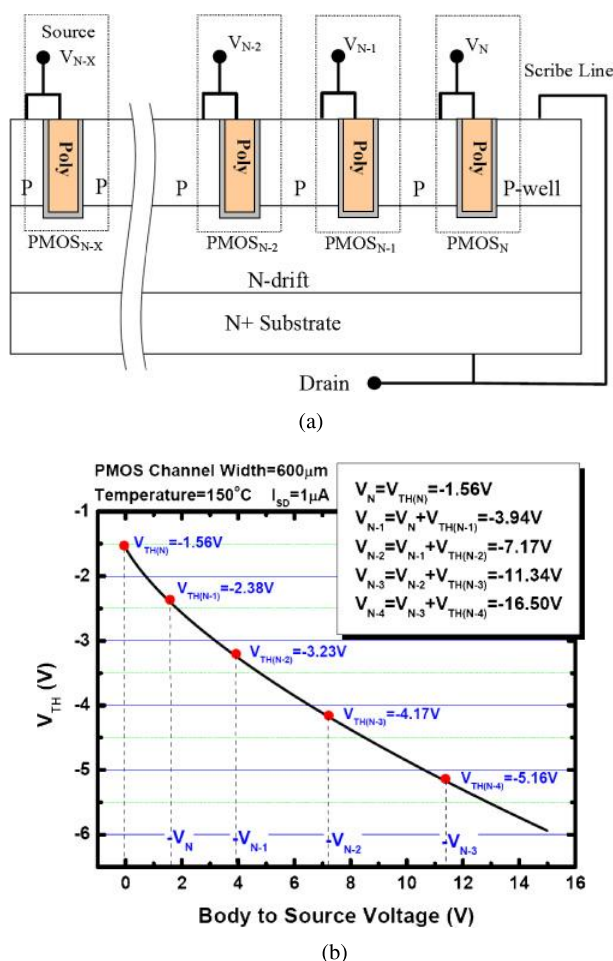


Fig. 10. (Color online) (a) The definition of trench termination structure for future design. (b)  $V_{TH}$  of all PMOS in (a). Different  $V_{TH}$  values of PMOS are caused by the different  $V_{BS}$  values.

#### 4. Conclusions

A trench termination of a 16 V trench PowerMOS has been implemented. The breakdown voltage can be designed by modifying the trench numbers without changing any process step. The field oxidation thermal budget, lithography of field oxide and field oxide etching can be removed by using trench termination in place of the metal field plate termination of the conventional field oxide. To design trench termination in technology with other voltage ratings, the gate oxide safe operating voltage should be considered at the first floating node. The temperature effect of trench termination should be considered owing to the negative coefficient of PMOS  $V_{TH}$  and temperature. The proposed trench termination design method provides an easy way of designing suitable trench termination in a trench PowerMOS.

- 1) C. Kocon, A. Challa, and P. Thorup: Proc. Int. Symp. Power Semiconductor Devices and IC's, 2006, p. 101.
- 2) M. A. A. Zandt, E. A. Hijzen, R. J. E. Hueting, and G. E. J. Koops: Proc. Circuit, Devices and Systems, 2004, p. 269.
- 3) P. Goarin, R. V. Dalen, G. Koops, and C. L. Cam: Proc. Solid-State Device Research Conf., 2006, p. 274.
- 4) I.-Y. Park, S.-G. Kim, J.-G. Koo, and J. Kim: *Electron. Lett.* **39** (2003) 1414.
- 5) S. Ono, Y. Kawaguchi, and A. Nakagawa: Proc. Int. Symp. Power Semiconductor Devices and IC's, 2003, p. 28.
- 6) M. Darwish, C. Yue, K. H. Lui, F. Giles, B. Chen, K. Chen, D. Pattanayak, Q. Chen, K. Terrill, and K. Owyang: Proc. Int. Symp. Power Semiconductor Devices and IC's, 2003, p. 24.
- 7) L. Ma, A. Amali, S. Kiyawat, A. Mirchandani, D. He, N. Thapar, R. Sodhi, K. Spring, and D. Kinzer: Proc. Int. Symp. Power Semiconductor Devices and IC's, 2003, p. 354.
- 8) S. T. Peake, R. Grover, R. Farr, C. Rogers, and G. Petkos: Proc. Int. Symp. Power Semiconductor Devices and IC's, 2002, p. 29.
- 9) A. Finney, J. Evans, P. Blair, J. Earnshaw, P. Jerred, K. Lowe, D. Mottram, N. Wolstenholme, and A. Wood: Proc. Int. Symp. Power Semiconductor Devices and IC's, 2001, p. 283.

### Вступ. Логічні елементи

Комп'ютерна схемотехніка – це науково-технічна дисципліна, яка вивчає теоретичні методи аналізу і синтезу схем комп'ютерів (електронних обчислювальних машин) і засоби їхньої технічної реалізації. Розвиток комп'ютерної схемотехніки є основою удосконалення архітектури комп'ютерів, якісного підвищення їхньої продуктивності та надійності, істотного зменшення масових та габаритних показників. Комп'ютери широко використовують у цивільній авіації та інших галузях господарства.

### Поняття елементів, вузлів і пристроїв комп'ютерної схемотехніки

Технічні засоби комп'ютерної схемотехніки залежно від функцій, які вони виконують, поділяють на елементи, функціональні вузли і пристрої, а також мікропроцесори та комп'ютери (рис. 1.1). Вони призначені для оброблення дискретної інформації і тому називаються цифровими.

Технічні засоби комп'ютерної схемотехніки в даний час основані на інтегральних мікросхемах (ІМС) різного ступеня складності.

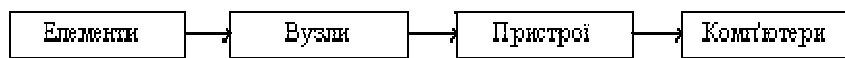


Рис.1.1. Склад технічних засобів комп'ютерної схемотехніки

Елементами в комп'ютерній схемотехніці називаються найменші неподільні мікроелектронні схеми (вироби), призначені для виконання логічних операцій або зберігання біта інформації. До елементів умовно відносяться і допоміжні схеми – підсилювачі, повторювачі, формувачі та ін.

Елементи будуються на основі двопозиційних ключів, що технічно реалізується найпростіше. Елементи з двома станами називаються двійковими.

На входах і виходах двійкового елемента діють напруги, які набувають у сталому режимі двох значень – високого УН і низького УН рівнів (індекси від англійських слів High і Low). Ці напруги відображають електричні сигнали. Сигнал з двома станами називається двійковим. Перехід елемента з одного стану в інший називається його перемиканням. На основі елементів будують типові функціональні вузли.

Елементарні дії, які виконуються в комп'ютерах за один машинний такт, називаються мікроопераціями. Наприклад, інкремент або декремент слова, зсув, інверсія, додавання та ін.

У комп'ютерах команди виконують послідовністю мікрооперацій над двійковими словами (числами). Типовими функціональними вузлами комп'ютерів називаються мікроелектронні схеми, призначені для виконання однієї або декількох мікрооперацій.

За логікою роботи функціональні вузли розподіляються на комбінаційні та послідовнісні схеми.

У комбінаційних схемах логічний стан виходів елементів залежить тільки від комбінації вхідних сигналів у даний момент часу.

До функціональних вузлів комбінаційного типу відносяться суматори, дешифратори, шифратори, мультиплексори і демультиплексори, схеми порівняння (компаратори) і контролю за парністю, кодоперетворювачі.

У послідовнісних схемах логічне значення виходів визначають як комбінацією вихідних сигналів, так і станом пам'яті схеми в даний момент часу. До функціональних вузлів послідовнісного типу відносяться регістри, лічильники, генератори чисел і керуючі автомати. На основі типових функціональних вузлів будують різноманітні пристрої комп'ютерів.

Універсальність комп'ютерів забезпечує можливість приймання і видавання інформації, її зберігання та арифметико-логічне опрацювання, а також керування усім обчислювальним процесом. Ці функції реалізуються відповідними пристроями введення, виведення, запам'ятовування, арифметико-логічними і керування.

В усіх комп'ютерах використовують генератор тактових імпульсів (ГТІ), що виробляє періодичну послідовність прямокутних імпульсів, які називаються тактовими (С). Початок кожного імпульсу С називається тактовим моментом. Часовий інтервал між двома сусідніми імпульсами С називається машинним тактом  $T_C$ . На початку кожного імпульсу С відбувається зміна інформації на входах елементів і вузлів машини (рис. 1.2).

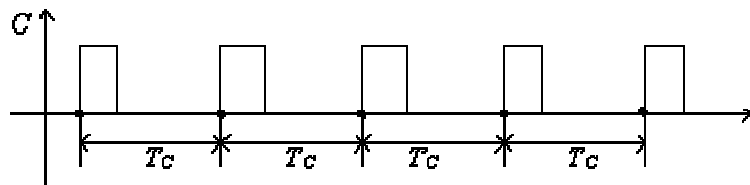


Рис. 1.2. Часова діаграма ГТІ

Частота ГТІ вимірюється десятками і сотнями мегагерц. У літературі ГТІ часто називають генераторами синхронізуючих імпульсів, а самі імпульси називають синхронізуючими або синхроімпульсами. Амплітуда і полярність імпульсу С залежить від фізичних принципів побудови машини. Принцип подачі інформації на входи елементів і вузлів у тактові моменти називається дискретизацією сигналів у часі.

У комп'ютерній схемотехніці застосовуються два основних види двійкових сигналів: потенціальні й імпульсні (рис. 1.3). Сигнал, який змінюється тільки в тактові моменти часу, називається потенціальним. Сигнал, що наростає в тактовий момент, а спадає в границях даного такту, називається імпульсним. Тривалість потенціального сигналу дорівнює або кратна тривалості машинного такту.

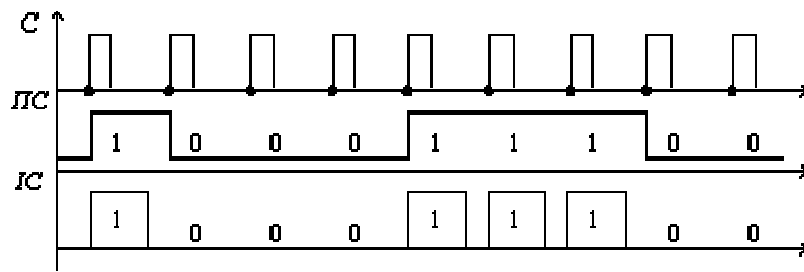


Рис. 1.3. Діаграми потенціальних (ПС) та імпульсних (ІС) сигналів

У логіці значення двійкового сигналу і відповідної змінної  $X$  кодуються символами 0 (лог. 0) і 1 (лог. 1). Напругу, що відображає символ 1, позначимо через  $U_1$ , а символ 0 – через  $U_0$ . Розрізняють два способи кодування логічних сигналів  $X_i$  потенціальними сигналами – позитивний та негативний. При позитивному кодуванні (позитивна логіка чи угода) більший рівень напруги  $U_H$  з урахуванням знака відображає лог. 1, а менший  $U_L$  – лог. 0, тобто  $X = 1$ , якщо  $U_1 = U_H$ , та  $X = 0$  при  $U_0 = U_L$  (рис. 1.4, а). При негативному кодуванні (негативна логіка чи угода) більший рівень напруги  $U_H$  з урахуванням знаку відображає лог. 0, а менший  $U_L$  – лог. 1, тобто  $X = 1$ , якщо  $U_1 = U_L$ , та  $X = 0$  при  $U_0 = U_H$  (рис. 1.4, б).

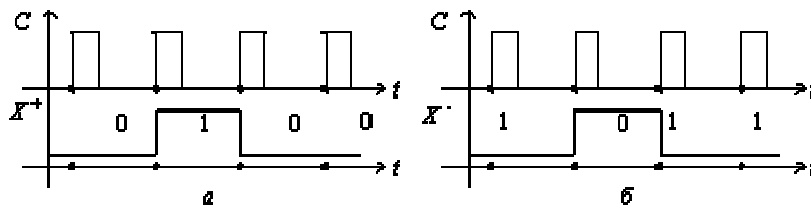


Рис. 1.4. Логічні угоди: а – позитивна  $X^+$ ; б – негативна  $X^-$

Для імпульсних сигналів розрізняють два роди кодування (рис. 1.5): перший – наявність імпульсу відображає лог. 1, відсутність – лог. 0; другий – наявність імпульсу однієї полярності відображає лог. 1, а іншої полярності – лог. 0.

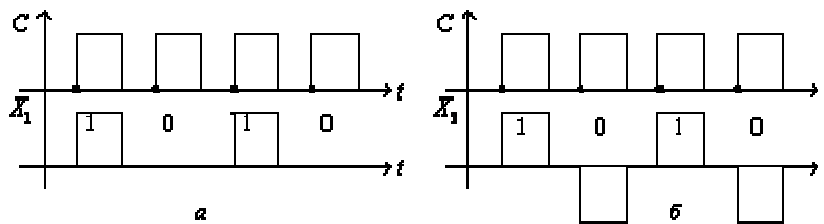


Рис. 1.5. Імпульсне кодування: а – першого роду; б – другого роду

Елементи комп'ютерів класифікуються за такими ознаками:

- використовуваними фізичними приладами;
- видом інформаційних сигналів;
- функціональним призначенням;
- конструкційно-технологічним виготовленням;
- рівнем і ступенем складності.

За типом фізичних приладів розрізняють такі елементи: побудовані на електронних лампах – перше покоління; на транзисторах – друге покоління; на ІМС малого і середнього ступеня інтеграції – третє покоління; на великих і надвеликих ІМС – четверте покоління.

За видом інформаційних сигналів виділяють:

- потенціальні елементи – використовуються тільки потенціальні сигнали;
- імпульсні елементи – використовуються тільки імпульсні сигнали;
- потенціально-імпульсні елементи – використовуються потенціальні й імпульсні сигнали.

За функціональним призначенням елементи комп'ютерної схемотехніки поділяють на такі класи:

логічні елементи, призначені для виконання логічних операцій – НЕ, І, ЧИ, НЕ І, НЕ ЧИ, НЕ І ЧИ та ін.;

елементи, які запам'ятовують – тригери, призначені для зберігання значення однієї двійкової змінної – нуля або одиниці, тобто одного біта інформації;

допоміжні елементи (підсилювачі, формувачі, перетворювачі сигналів, схеми часового узгодження, генератори імпульсів та ін.), призначені для забезпечення роботи елементів перших двох класів.

За конструкційно-технологічним виготовленням елементна база сучасної комп'ютерної схемотехніки складається з інтегральних мікросхем. Це – мікроелектронні вироби з високою щільністю упакування електрорадіоелементів (резисторів, діодів, транзисторів) і з'єднань між ними. З погляду специфікації, випробування, постачання та експлуатації ІМС розглядаються як єдине ціле. В ІМС електрорадіоелементи називаються елементами, якщо вони невіддільні від схеми, і компонентами, якщо їх можна використовувати самостійно. Надалі, щоб уникнути плутанини з елементами комп'ютерів, електрорадіоелементи називають просто компонентами.

Мікросхеми класифікують за такими головними ознаками:

- технологією виготовлення – напівпровідникові, гібридні, плівкові;
- конструкційним оформленням – корпусні та безкорпусні;
- формою оброблення інформації – аналогові, цифрові й аналого-цифрові;
- ступенем інтеграції (складності) – малі, середні, великі, надвеликі й ультравеликі;
- типом активних елементів – побудованих на біполярних і МОН-транзисторах;
- областю застосування – широкого застосування, спеціалізовані, у тому числі замовлені і напівзамовлені;
- використовуваними матеріалами – кремнієві, арсенід-галієві;
- перспективними напрямками – кріомікроелектронні, акустоелектронні, оптоелектронні, молекулярної електроніки та ін.

Набір цифрових мікросхем із спільними конструкційно-технологічними і схемотехнічними ознаками утворює серію ІМС. У комп'ютерній схемотехніці широко застосовуються цифрові напівпровідникові корпусні ІМС на основі кремнію і арсеніду галію. У напівпровідникових ІМС усі компоненти і з'єднання між ними виконані в об'ємі і на поверхні кристала площею від 4 до 100 мм<sup>2</sup>. У гібридних ІМС навісні компоненти кріпляться на поверхні діелектричної підкладки. У плівкових ІМС усі компоненти і з'єднання між ними виконані у виді тонких плівок на діелектричній підкладці.

Складність мікросхем характеризується рівнем інтеграції  $N$ , ступенем інтеграції  $K = \lg N$  і ступенем функціональної складності  $F = \lg L$ , де  $N$  – число компонентів, комп.;  $L$  – число двохходових логічних елементів (вентилів); значення десятичного логарифма округляється до більшого цілого числа. Промисловість виготовляє ІМС від першого (менше 10 комп.) до шостого (менше 1 млн комп.) і вище ступеня інтеграції.

Можливості інтегральної технології визначає щільність упакування: відношення числа компонентів до об'єму (іноді до площі) кристала. Щільність упакування в напівпровідникових ІМС складає 107 комп./см<sup>3</sup>, а для гібридних – 100–200 комп./см<sup>3</sup>.

Мала інтегральна схема (МІС) вміщує до 100 комп. включно, середня мікросхема (СІС) – 100–1000 комп., велика інтегральна схема (ВІС) – до 100000 комп., надвелика інтегральна мікросхема (НВІС) – до 1 млн комп., а ультравелика (ультра-ВІС) – до 10 млн комп. і більше. На МІС будують елементи, на СІС реалізують типові вузли, на ВІС, НВІС і ультра-ВІС забезпечують побудову мікропроцесорів і мікрокомп'ютерів.

Кожна елементарна логічна функція реалізується відповідно логічним елементом: ЧІ (диз'юнктор), І (кон'юнктор), НЕ (інвертор). Для реалізації складних функцій логічні елементи об'єднуються у логічну схему.

Функціонально повна система логічних елементів дозволяє побудувати будь-яку складну логічну схему. Такі системи утворюються такими наборами логічних елементів: 1) ЧІ, НЕ; 2) І, НЕ; 3) НЕ ЧІ; 4) НЕ І та іншими.

У технічно повній системі елементів забезпечується значення електричних параметрів двійкових сигналів, для цього використовуються допоміжні елементи – підсилювачі, повторювачі, формувачі та ін.

З урахуванням вищевикладеного можна сказати, що система елементів являє собою функціонально і технічно повний набір елементів, який використовує однакові способи представлення інформації, а також має спільні конструктивно-технологічні характеристики.

## Характеристики логічних елементів

Логічні, схемотехнічні й експлуатаційні властивості логічних елементів визначаються сукупністю характеристик і параметрів, до яких відносяться:

1) функції логічних елементів; 2) логічні угоди; 3) коефіцієнти об'єднання за входом і виходом; 4) коефіцієнт розгалуження; 5) швидкодія; 6) потужність споживання; 7) робота перемикачів; 8) вхідні й вихідні напруги і струми; 9) статична і динамічна стійкість до перешкод; 10) надійність елементів; 11) допустимі розміри механічних впливів, діапазони тиску і температури навколишнього середовища, стійкість до радіаційних впливів; 12) маса, вартість і конструктивне оформлення. У більшості випадків зазначені характеристики і параметри відносяться і до ІМС, на яких реалізовані логічні елементи.

Коефіцієнт об'єднання за входом  $N_I$  характеризує число логічних входів логічного елемента – зазвичай 1, 2, 3, 4 або 8 (рис. 1.6).

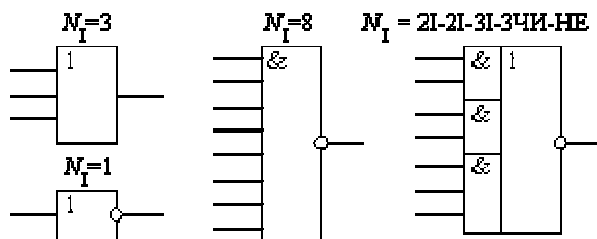


Рис. 1.6. Приклади значень коефіцієнта  $N_I$

Коефіцієнт об'єднання за виходом  $N_O$  характеризує допустиму кількість з'єднаних між собою виходів логічних елементів з метою утворення нових функцій.

Коефіцієнт розгалуження за виходом  $N_P$  характеризує навантажувальну спроможність логічного елемента, тобто максимальне число входів ідентичних схем, яке може бути одночасно залучене до виходу даного елемента без порушення його працездатності (рис. 1.7). До складу серій ІМС зазвичай входять елементи з малою навантажувальною спроможністю ( $N_P = 3 \dots 15$ ) та з великою ( $N_P = 30 \dots 50$ ).

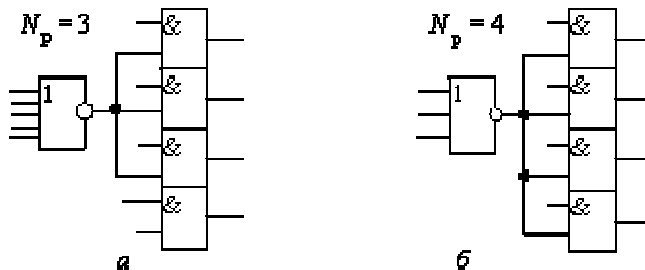


Рис. 1.7. Підключення навантаження: а –  $N_P = 3$ ; б –  $N_P = 4$

Прийняті такі визначення і буквені позначення електричних параметрів цифрових мікросхем (ДСТУ 2883-94):

вхідні  $U_I$  і вихідні  $U_O$  рівні напруг (індекси – від англійських слів Input і Output);

вхідні напруги низького  $U_{IL}$  і високого  $U_{IH}$  рівнів; для них установлюються максимальне значення низького рівня  $U_{IL\ max}$  та мінімальне значення високого рівня  $U_{IH\ min}$  (рис. 1.8, а);

вихідні напруги низького  $U_{OL}$  і високого  $U_{OH}$  рівнів; для них установлені максимальне значення низького рівня  $U_{OL\ max}$  та мінімальне значення високого рівня  $U_{OH\ min}$  (рис. 1.8, б);

вхідний  $I_I$  і вихідний  $I_O$  струми;

вхідний струм  $I_{IL}$  – при низькому рівні напруги на вході,  $I_{IH}$  – при високому;

вихідний струм  $I_{OL}$  – при низькому рівні напруги на виході, а  $I_{OH}$  – при високому;



UCC – значення напруги джерела живлення;

ICС – струм, споживаний ІМС від джерела живлення;

PCC – потужність, споживана ІМС від джерела живлення;

вхідні граничні напруги, при яких відбувається перемикання елемента:  $U_{TH}$  – найменше значення для високого рівня і

$U_{TIL}$  – найбільше значення для низького рівня.

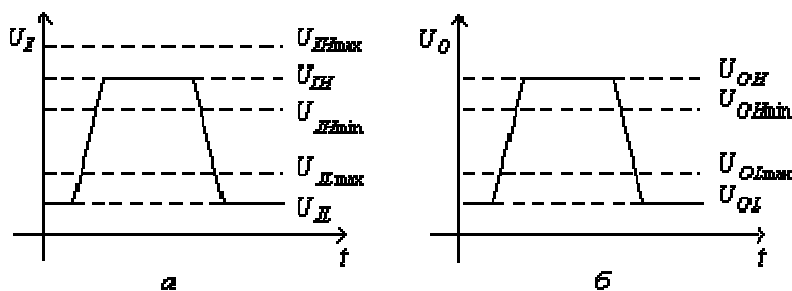


Рис. 1.8. Позначення рівнів напруги: а – вхідних; б – вихідних

Основні параметри логічних елементів визначають за допомогою вхідної, вихідної і передатної характеристик. Типові графіки цих характеристик для інвертувальних елементів транзисторно - транзисторної логіки наведені на рис. 1.9.

Вхідна характеристика логічного елемента  $I_I = f(U_I)$  – це залежність вхідного струму від зміни вхідної напруги. Струми, що втікають у схему елемента, вважають додатними, а ті, що витікають – від'ємними (рис.1.9, а). З цієї характеристики визначають вхідні струми  $I_{IL}$  для напруги  $U_{IL\ max}$  і струм  $I_{IH}$  для напруги  $U_{IH\ min}$ .

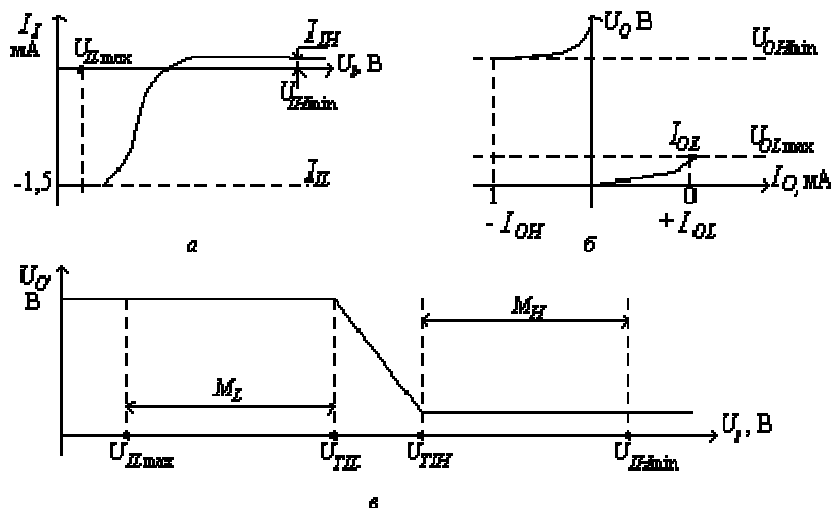


Рис. 1.9. Характеристики логічного транзисторно - транзисторного елемента: а – вхідна; б – вихідна; в – передавальна

Вихідна характеристика логічного елемента  $U_O = f(I_O)$  визначає залежність вихідної напруги від струму навантаження для станів високого і низького рівнів (рис. 1.9, б).

З цієї характеристики визначають допустимі значення струмів:  $I_{OL}$  – при низькому рівні вихідної напруги  $U_{OL\ max}$  і  $I_{OH}$  – при високому рівні напруги  $U_{OH\ min}$  (рис. 1.9, б).

Передатна характеристика  $U_O = f(U_I)$  – це залежність вихідної напруги від вхідної (рис. 1.9, в). З цієї характеристики визначають значення завадостійкості для низького рівня на вході  $M_L$  (перешкода, що відкриває) і для високого рівня на вході  $M_H$  (перешкода, що закриває):

$M_L = U_{TIL} - U_{IL\ max}$ ;  $M_H = U_{TH\ min} - U_{TH}$ .

Середня споживана потужність  $P^*CC$  елементом від джерела живлення обчислюється за формулою

$P^*CC = U_{CC} (I_{CCL} + I_{CCH}) / 2 = U_{CC} I^*CC$ ,

де  $I_{CCL}$ ,  $I_{CCH}$  – струми споживання при низькому і високому рівнях напруги на виході відповідно;  $I^*CC$  – середній струм споживання.

Сучасні елементи споживають потужність від мікроват до десятків міліватів.

Потенціальні сигнали характеризуються значенням логічного перепаду (амплітудою)  $U_M = U_H - U_L$  і тривалістю позитивного  $t_{WH}$  та негативного  $t_{WL}$  перепадів (рис. 1.10). Перепади напруг часто називають позитивними і негативними імпульсами.

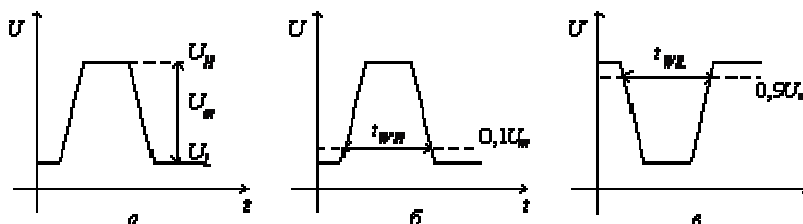


Рис. 1.10. Вимір параметрів сигналу: а – амплітуди; б, в – тривалості перепадів

Для вимірювання часових параметрів сигналу встановлюють умовні рівні в частках від амплітуди – 0,1; 0,5 і 0,9.

Швидкодію мікросхем визначають за значеннями таких тривалостей:

фронту  $t_{LH}$  і спаду  $t_{HL}$  (рис. 1.11, а);

власне вмикання  $t_{THL}$  і вимикання  $t_{TLH}$  (рис. 1.11, б); та їх затримки відповідно  $t_{DHL}$  та  $t_{DLH}$ ; затримок поширення сигналу при вмиканні  $t_{PHL}$  і вимиканні  $t_{PLH}$  (рис. 1.11, в).

Для практичних розрахунків використовують середній час затримки поширення сигналу  $t_P = (t_{PHL} + t_{PLH}) / 2$ .

Для оцінки якості елемента широко використовують узагальнений параметр – роботу перемикання  $АП = P \cdot CC \cdot t_P$ .

Якщо потужність  $P \cdot CC$  вимірюється в міліватах, а час затримки – в наносекундах, то робота перемикання АП виражається в пікоджоулях (пДж). Значення узагальненого параметра АП знаходиться в границях 0,1–200 пДж. Чим менше значення АП, тим кращі характеристики має логічний елемент.

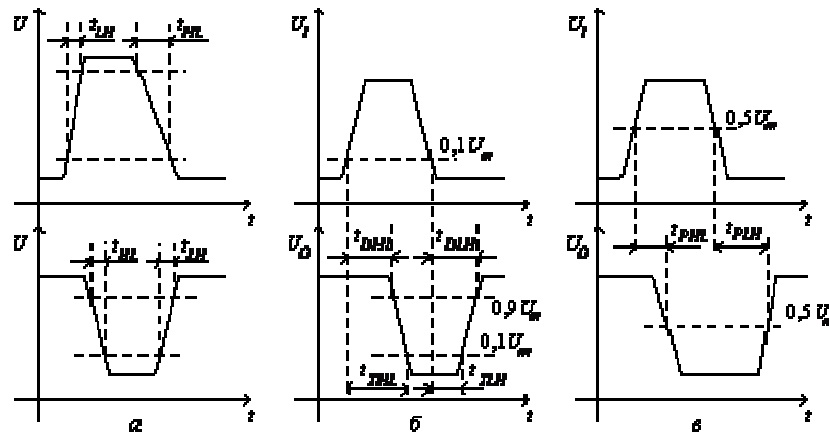


Рис. 1.11. Вимір часових параметрів сигналів: а – фронту  $t_{LH}$  і спаду  $t_{HL}$ ;

б – часу вмикання  $t_{THL}$  і вимикання  $t_{TLH}$ ; в – часу затримок розповсюдження сигналу при вмиканні  $t_{PHL}$  та вимиканні  $t_{PLH}$

Надійність ІМС характеризується трьома взаємозалежними показниками: 1) інтенсивністю відмов  $I = n / (mt)$ , де  $n$  – число відмов за час випробування, год;  $m$  – загальна кількість випробуваних мікросхем; 2) напрацюванням на відмову  $T = 1/I$ ; 3) можливістю безвідмовної роботи протягом заданого інтервалу часу  $P = \exp(-It)$ .

Для сучасних ІМС інтенсивність відмов  $I = (10^{-7} \dots 10^{-8})$ . Приймаючи, що  $I = 10^{-8}$ ,  $t = 15000$ , одержимо значення ймовірності безвідмовної роботи  $P(t) = 0,998$  або 99,8%.

### 3. 1. Визначення та призначення тригерів

Тригер – це запам'ятовуючий елемент з двома стійкими станами, зміна яких відбувається під дією вхідних сигналів. Як елемент комп'ютера, тригер призначений для зберігання одного біта інформації, тобто лог. 0 або лог. 1. Схема тригера забезпечує записування, зчитування, стирання та індикацію двійкової інформації, яка зберігається. На основі тригерів будують типові функціональні вузли комп'ютерів – регістри, лічильники, накопичувальні суматори, а також мікропрограми автоматів.

Усі різновиди тригерів являють собою елементарний автомат, який вміщує власне елемент пам'яті (ЕП) та схему керування (СхК), яка утворює вхідну логіку (рис.3.1).

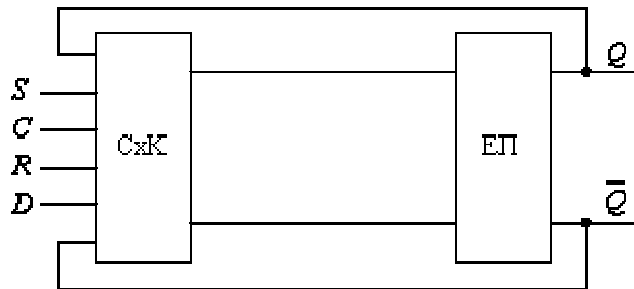


Рис.3.1. Структура тригера у вигляді ЕП і СхК

Стан тригера визначається сигналами на прямому  $Q$  та інверсному  $\bar{Q}$  виходах. При позитивному кодуванні інформації високий рівень напруги на прямому виході відображає значення лог. 1 (стан  $Q = 1$ ), а низький рівень – значення лог. 0 (стан  $Q = 0$ ). Зміна стану тригера (його перемикання) забезпечується зовнішніми сигналами й сигналами зворотного зв'язку на виході тригера, які поступають на входи СхК. Звичайно зовнішні сигнали, як і входи тригера, позначають латинськими буквами R, S, T, C, V та іншими. В найпростіших схемах тригерів окрема СхК може бути відсутньою. Оскільки функціональні властивості тригерів визначаються їхньою СхК, то назви основних входів переносяться на всю схему тригера.

### 3. 2. Класифікація тригерів

Тригери класифікують за такими ознаками: логікою функціонування (RS, JK, D, T та ін.); способом записування інформації (асинхронні й синхронні); моментом реакції на тактовий сигнал (статичні, динамічні); кількістю тактів синхронізації (одно-, дво- і тритактові); кількістю ступенів (одно- або двоступеневі тригери); складом логічних елементів (тригери на елементах НЕ І, НЕ ІІ, НЕ ІІІ та ін.).

Відповідно до логіки функціонування розрізняють такі тригери: з роздільною установкою станів "0" і "1" (RS-тригери); з одним інформаційним входом (D-тригери); з лічильним входом (Т-тригери); універсальні з роздільною установкою станів "0" і "1" (JK-тригери); комбіновані (RST-, RSJK-тригери); із складною вхідною логікою.

Входи тригерів розділяються на інформаційні (R, S, T та ін.) та керуючі (C, V). Інформаційні (логічні) входи призначені для приймання сигналів інформації, яка запам'ятовується. Назви вхідних сигналів ототожнюють з назвами входів тригера. Керуючі входи служать для керування записуванням інформації. У тригерах може бути два види керуючих сигналів: синхронізуючий (тактовий) сигнал C, який надходить до C-входу (тактового входу) і дозволяючий сигнал V, який надходить до V-входу.

За способом записування (приймання) інформації розрізняють асинхронні й синхронні (тактовні) тригери. Тригери, які не мають C-входу, називаються асинхронними (рис.3.2, а і б). В асинхронних тригерах записування інформації відбувається в будь-який момент часу при надходженні сигналів до інформаційних входів.

Тригери, які мають C-вхід, називаються синхронними. У синхронному тригері записування інформації можливе при збігу сигналів на інформаційному й синхронному входах. Цим пояснюється вища стійкість до перешкод синхронних тригерів порівняно з асинхронними.

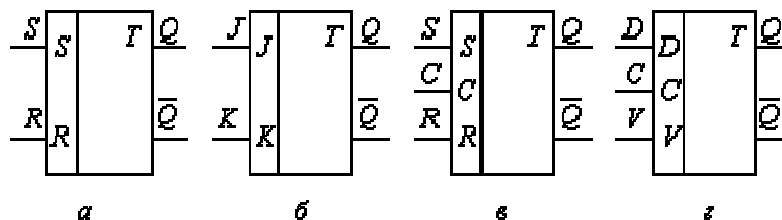


Рис.3.2. Умовні позначення тригерів: а, б – асинхронних; в, г – синхронних

До V-входів тригера надходять сигнали, які дозволяють ( $V = 1$ ) або забороняють ( $V = 0$ ) записування інформації. У синхронних тригерах з V-входом записування інформації можливе при збігу сигналів на інформаційному, C- і V- виходах (рис.3.2, г).

Залежно від кількості тактових сигналів, необхідних для формування нового стану, розрізняють одноктактові, двотактові та багатотактові тригери.

За способом керування записуванням (моментом реакції на тактовий сигнал) виділяють синхронні тригери зі статичним (за рівнем), динамічним (за фронтами) та двоступеневим керуванням. В асинхронних тригерах записування нуля і одиниці можливе у будь-який момент часу, при цьому вхідний інформаційний сигнал одночасно є й керуючим. У синхронних тригерах з керуванням за рівнем записування інформації можливе тільки впродовж тривалості тактового сигналу. При цьому тактові сигнали можуть бути прямими (змінюватися від нуля до одиниці) або інверсними (змінюватися від одиниці до нуля) (рис.3.3, а і б).

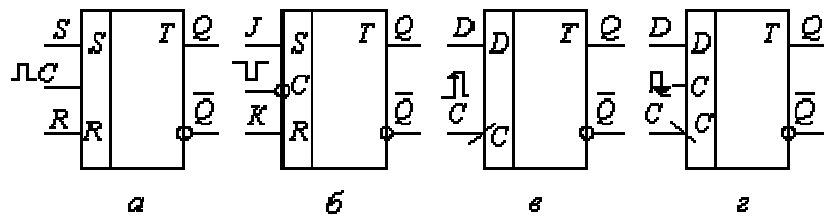


Рис.3.3. Керуючі входи тригера: а – прямий статичний; б – інверсний статичний; в – прямий динамічний; г – інверсний динамічний

При керуванні фронтами дозвіл на записування інформації дається тільки в момент перепаду тактового сигналу від нуля до одиниці (прямий динамічний вхід) або від одиниці до нуля (інверсний динамічний вхід). В інші моменти часу тригер не реагує на вхідні інформаційні сигнали незалежно від рівня тактового імпульсу (рис.3.3, в і г).

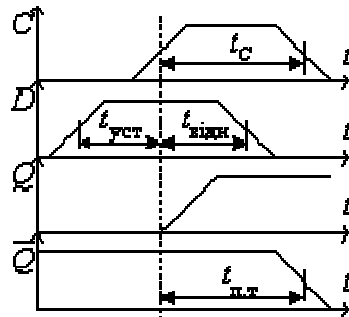


Рис.3.4. Вимірювання часових параметрів тригерів D-типу

### 3.3. Динамічні параметри тригерів

Для тригерів встановлені такі динамічні параметри, які вимірюються на рівні половини амплітуди сигналів (рис.3.4):

- мінімальна тривалість імпульсу  $t_C$  на тактовому вході;
- мінімальний час попереднього установлення сигналу на інформаційному вході  $t_{уст}$ ;
- час відновлення (фіксації)  $t_{відн}$  – мінімальний час між початком синхросигналу  $C$  і закінченням інформаційного сигналу  $D$ ; для асинхронних тригерів  $t_{відн}$  – просто тривалість вхідного сигналу;
- час перемикання тригера  $t_{п.т}$  – часовий інтервал між фронтом вхідного перемикаючого сигналу та фронтом сигналу на виході

$Q$  або  $\bar{Q}$  (який пізніше набуває нового положення). Мінімальна тривалість синхросигналу на вході тригера визначається максимальним часом перемикання тригера  $t_C \geq t_{п.т} \max$ . У двоступеневому тригері з одноктактною синхронізацією час перемикання другого ступеня визначається відносно спаду синхроімпульсу.

### 3.4. Таблиця переходів і логічні рівняння RS-тригера

RS-тригером називають запам'ятовувачий елемент з роздільними інформаційними входами для установлення його в стан "0" (R-вхід) і в стан "1" (S-вхід). Назва "RS-тригер" утворена від перших літер слів RESET (скинення) і SET (установлення).

У таблиці переходів RS-тригера (табл.3.1) прийняті позначення:  $R_t$ ,  $S_t$ ,  $Q_t$  – значення логічних змінних у момент часу  $t$  на входах  $R$ ,  $S$  і виході  $Q$ ;  $Q_{t+1}$  – стан тригера після перемикання;  $K_6$ ,  $K_7$  – невизначені коефіцієнти на тих наборах, де вхідні сигнали  $R_t$  і  $S_t$  одночасно набувають значення одиниці (заборонена комбінація сигналів).

**Таблиця 3.1**

$R_t$	$S_t$	$Q_t$	$Q_{t+1}$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	$K_6$
1	1	1	$K_7$

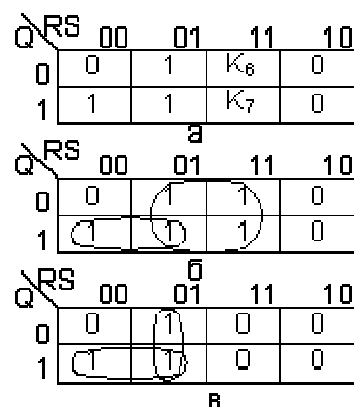


Рис.3.5. Карта Карно для RS-тригерів

Таблиці переходів відповідає карта Карно (рис.3.5, а), де значення функції  $Q_{t+1}$  для мінтермів  $R_t, S_t, \bar{Q}_t$ , і  $R_t, S_t, Q_t$  замінені невизначеними коефіцієнтами  $K_6$  і  $K_7$ . Припустивши, що комбінації вхідних сигналів  $R_t S_t = 1$  не існує, отримуємо карти Карно

для  $K_6 = K_7 = 1$  (рис.3.5, б) і  $K_6 = K_7 = 0$  (рис.3.5, в).

Із карт Карно отримуємо логічні рівняння асинхронного RS-тригера:

$$K_6 = K_7 = 1, \quad Q_{t+1} = S_t \vee \overline{R_t} Q_t, \quad (3.1)$$

$$K_6 = K_7 = 0, \quad Q_{t+1} = \overline{R_t} (S_t \vee Q_t). \quad (3.2)$$

Логічні вирази (3.1) і (3.2) визначають новий стан тригера  $Q_{t+1}$  залежно від старого стану  $Q_t$  та вхідних сигналів  $R_t$  і  $S_t$ . В подальшому для спрощення індекс  $t$  у правій частині логічного виразу опускається.

Асинхронний RS-тригер на елементах НЕ І. Перетворимо логічний вираз (2.1) до виду, зручного для реалізації на елементах НЕ І:

$$Q_{t+1} = \overline{S \vee \overline{R} \cdot Q} = \overline{\overline{S} \cdot \overline{\overline{R} \cdot Q}}. \quad (3.3)$$

Схема асинхронного RS-тригера на двох елементах НЕ І з логічними зв'язками на основі виразу (3.3) показана на рис.3.6, а. Особливістю цього тригера є інверсне керування за інформаційними входами, що відображається в умовному графічному позначенні.

Із аналізу діаграм роботи RS-тригера випливає, що елементи НЕ І в схемі перемикаються послідовно. Є інтервал часу, коли на обох виходах встановлюються однакові сигнали  $Q = 1$  і  $\overline{Q} = 1$  (рис.3.6, в, заштриховані області) – явище “ризик”.

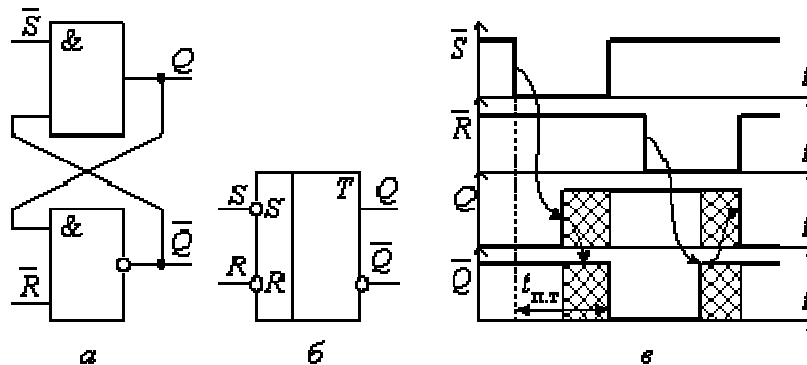


Рис.3.6. Асинхронний RS-тригер на елементах НЕ І: а – схема; б – умовне позначення; в – часові діаграми

Тривалість перемикання тригера визначається сумою затримок:  $t_{п.т} = 2t_p$ . Тривалість вхідного сигналу визначається з умови  $t_i \geq t_{п.т}$ . На практиці для надійності перемикання тригера тривалість вхідного імпульсу збільшують на одну затримку, тобто  $t_i = 3t_p$ . Максимальна і робоча частоти перемикання тригера відповідно дорівнюють  $f_{max} = 1/(2t_p)$  і  $f_p = 1/(3t_p)$ .

Асинхронний RS-тригер на елементах НЕ ЧИ. Перетворимо логічний вираз (3.2) до вигляду, зручного для реалізації на елементах НЕ ЧИ:

$$Q_{t+1} = \overline{\overline{R(S \vee Q)}} = \overline{R \vee \overline{S \vee Q}}. \quad (3.4)$$

Схема асинхронного RS-тригера на двох елементах НЕ ЧИ з логічними зв'язками на основі виразу (3.4) показана на рис.3.7, а.

Із аналізу діаграм роботи RS-тригера випливає, що елементи НЕ ЧИ в схемі перемикаються послідовно. Є інтервал часу, коли на обох виходах утворюються однакові сигнали  $Q = 0$  і  $\overline{Q} = 0$ , – явище “ризик” (рис.3.7, в). Часові параметри даного тригера

аналогічні параметрам тригера, зображеного на схемі рис.2.7, а.

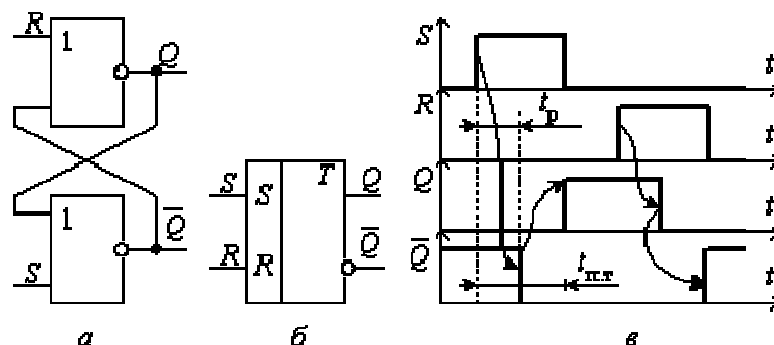


Рис.3.7. Асинхронний RS-тригер на елементах НЕ ЧИ: а – схема; б – умовне позначення; в – часові діаграми

Синхронний RS-тригер на елементах НЕ І. Для побудови синхронного RS-тригера на елементах НЕ І треба замінити в логічному виразі (3.3) змінні  $S$  і  $R$  на сполучення  $CS$  і  $CR$ , де  $C$  – синхро-сигнал:

$$Q_{t+1} = \overline{\overline{CS} \vee \overline{CR} Q}. \quad (3.5)$$

Схема синхронного RS-тригера на чотирьох елементах НЕ І з логічними зв'язками на основі виразу (3.5) показана на рис.3.8, а. Елементи D1 і D2 складають схему керування з прямими входами, а елементи D3 і D4 утворюють фіксатор (асинхронний RS-тригер).

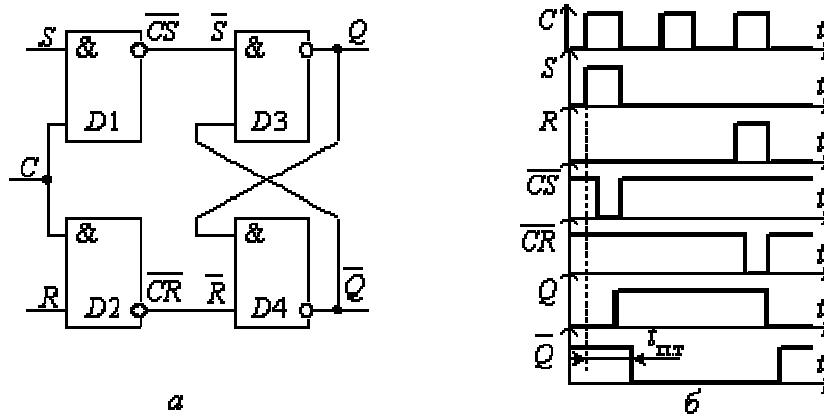


Рис.3.8. Синхронний RS-тригер на елементах HE I: а – схема; б – часові діаграми

При значенні сигналів  $CS = 1$  на виході елемента D1 встановлюється лог. 0 і тригер перемикається в стан "1". При значенні сигналів  $CR = 1$  на виході елемента D2 встановлюється лог. 0 і тригер перемикається в стан "0". Комбінація вхідних сигналів  $CSR = 1$  заборонена, оскільки призводить до невизначеного стану тригера. Із часової діаграми (рис.3.8, б) випливає, що час перемикання тригера  $t_{п.т} = 3t_{р}$ , а тривалість синхросигналу (з урахуванням запасу на одну затримку) визначається з умови  $t_C = 4t_{р}$ . Максимальна і робоча частоти перемикання тригера відповідно дорівнюють:  $f_{max} = 1/3t_{р}$  і  $f_{р} = 1/4t_{р}$ . Синхронний RS-тригер на елементах HE ЧИ. Для побудови синхронного RS-тригера на елементах HE ЧИ належить замінити в логічному виразі (3.4) змінні S і R на сполучення  $\overline{CS}$  і  $\overline{CR}$ :

$$Q_{n+1} = \overline{CR} \vee (\overline{CS} \vee Q) = \overline{C} \vee \overline{R} \vee (\overline{C} \vee \overline{S} \vee Q). \quad (3.6)$$

Схема синхронного RS-тригера на чотирьох елементах HE ЧИ з логічними зв'язками на основі виразу (3.6) показана на рис.3.9. Елементи D1 і D2 складають схему керування з інверсними входами, а елементи D3 і D4 утворюють фіксатор (асинхронний RS-тригер).

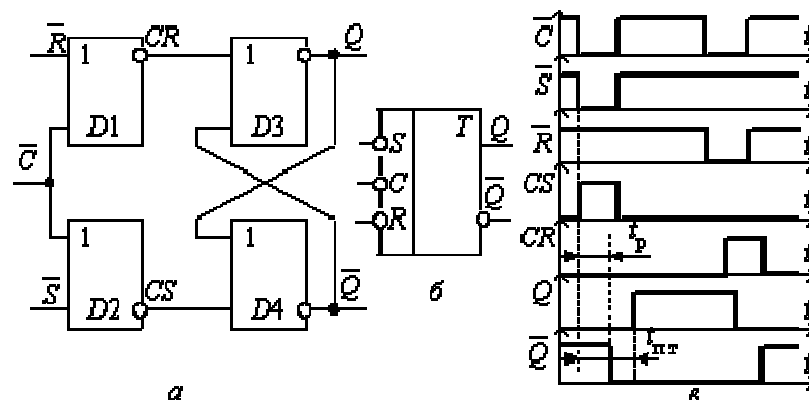


Рис.3.9. Синхронний RS-тригер на елементах HE ЧИ: а – схема; б – умовне позначення; в – часові діаграми

При значенні сигналів  $\overline{C} = 0$  і  $\overline{S} = 0$  на виході елемента D2 встановлюється лог. 1 (тобто  $CS = 1$ ) і тригер переключується в стан "1". При значенні сигналів  $\overline{C} = 0$  і  $\overline{R} = 0$  на виході елемента D1 встановлюється лог. 1 (тобто  $CR = 1$ ) і тригер переключується в стан "0". Комбінація сигналів  $\overline{C} = \overline{S} = \overline{R} = 0$  заборонена, тому що призводить до невизначеного стану тригера. Двоступеневі RS-тригери. Двоступеневі тригери будують за способом "М-S" і забезпечують поєднання двох процесів – одночасного записування нової інформації та зчитування старої. Під час дії синхроімпульсу C перший ступінь "М" (Master – основний) приймає нову вхідну інформацію, а другий ступінь "S" (Slave – допоміжний) в цей же час передає у зовнішні схеми стару інформацію. Після закінчення синхроімпульсу C інформація з першого ступеня переписується у другий ступінь. При однофазному (однотактному) обміні інформацією зв'язок між ступенями реалізується за допомогою інвертора (рис.3.10, а), забороняючих зв'язків (рис.3.10, б) або різнополярного керування (рис.3.10, в). При двотактному обміні зв'язок між ступенями забезпечується двома серіями синхросигналів – C1 і C2 (рис.3.10, г).

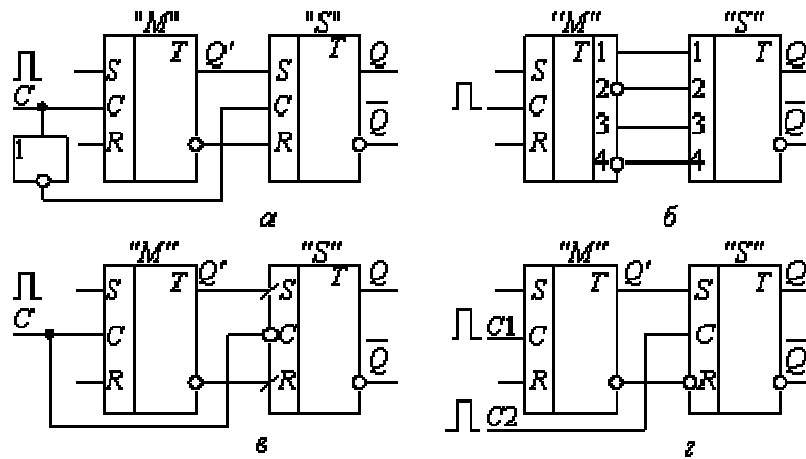


Рис.3.10. Організація зв'язку між ступенями тригера: а – з інвертором; б – із забороняючими зв'язками; в – з різнополярним керуванням; г – з двофазним обміном

## 2.5. Тригери типу JK

Тригером типу JK називається запам'ятовуючий елемент з двома стійкими станами та інформаційними входами J (аналог S) і K (аналог R), які забезпечують відповідно роздільну установку станів "1" і "0". Він функціонує подібно до RS-тригера, але при збігу сигналів  $JK = 1$  переключається в протилежний стан, тобто реалізує додавання сигналів за модулем два. Таким чином, JK-тригер не має заборонених комбінацій вхідних сигналів. Тригер типу JK є універсальним, оскільки може виконувати функції RS-тригера (при роздільному надходженні сигналів J і K), T-тригера (при одночасній подачі сигналів J і K), D-тригера (при подачі сигналу від входу J через інвертор на вхід K). Зміна станів JK-тригера наведена в табл.3.2; за допомогою карти Карно (рис.3.11) отримуємо наступне рівняння для тригера:

$$Q_{n+1} = \overline{K}_t \cdot Q_t \vee J_t \cdot \overline{Q}_t \quad (3.7)$$

Для побудови одноступеневого синхронного JK-тригера на елементах НЕ І потрібно замінити в рівнянні (3.7) змінні K і J на сполучення CK і JK, після чого виконати перетворення на основі правил подвійної інверсії та правил де Моргана:

$$Q_{n+1} = \overline{\overline{C} \cdot \overline{K} \cdot Q_t \vee C \cdot J \cdot \overline{Q}_t} = \overline{\overline{C} \cdot \overline{K} \cdot Q_t} \cdot \overline{C \cdot J \cdot \overline{Q}_t} \quad (3.8)$$

Таблиця 3.2

$K_t$	$J_t$	$Q_t$	$Q_{t+1}$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Q \ KJ	00 01 11 10			
	0	1	1	0
0	0	1	1	0
1	1	1	0	0

Рис.2.11. Карта Карно для JK-тригера

Схема одноступеневого JK-тригера з логічними зв'язками на основі рівняння (3.8) показана на рис.3.12.

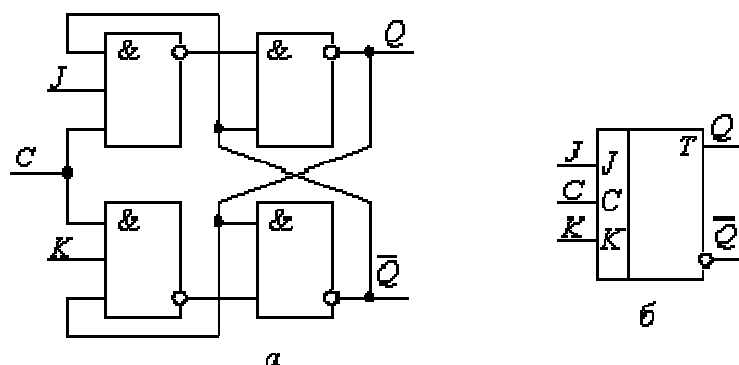


Рис.3.12. Одноступеневий JK-тригер: а – схема; б – умовне позначення

У схемотехніці потенціальних тригерів в основному застосовують двоступеневі синхронні JK-тригери на елементах НЕ І (рис.3.13). Нова інформація знімається з виходів Q основного М- ступеня, а стара – з виходів Q\* допоміжного S-ступеня.

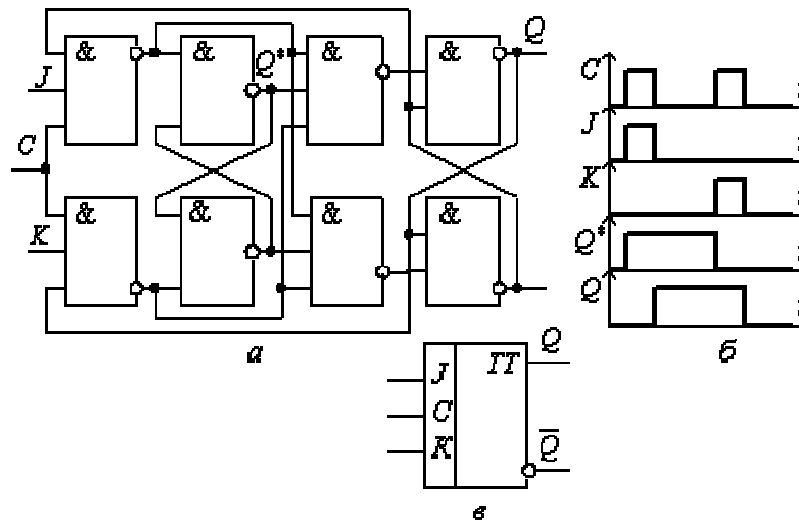


Рис.3.13. Двоступеневий JK-тригер на елементах НЕ І: а – схема; б – часові діаграми; в – умовне графічне позначення

Із часових діаграм (рис.3.13, б) слідує: при застосуванні JK-тригера для зберігання інформації сигнали на входи J і K подаються по чергово; при збігові сигналів на входах J і K реалізується лічильний тригер (на практиці ці входи часто монтажно об'єднуються). Час перемикання JK-тригера визначається сумою затримок першого і другого ступенів і дорівнює  $t_{п.т\tau} = 7t_{\tau}$ . На виході S-ступеня нова інформація з'являється після спаду сигналу C.

### 3.6. Таблиця переходів і логічні рівняння Т-тригера

Тригером типу Т називається запам'ятовуючий елемент з двома стійкими станами та одним інформаційним Т-входом. Стан Т-тригера змінюється на протилежний після кожного надходження лічильного сигналу на Т-вхід. Логіка функціонування асинхронного лічильного тригера подана таблицею переходів (табл.3.10) і описується логічним рівнянням

$$Q_{t+1} = \bar{T}_t \cdot Q_t \vee T_t \cdot \bar{Q}_t \quad (3.9)$$

Для побудови асинхронного RS-тригера на елементах НЕ І рівняння (3.9) перетворюється до вигляду, зручного для реалізації у заданому елементному базисі:

$$Q_{t+1} = \overline{\overline{\bar{T}} \cdot \overline{Q}} \vee \overline{\overline{T} \cdot \overline{\bar{Q}}} = \overline{\overline{\bar{T}} \cdot \overline{Q} \cdot \overline{\overline{T} \cdot \bar{Q}}} \quad (3.10)$$

У рівнянні (3.10) для виключення інверсії сигналу Т використовується тотожність  $\bar{T} \cdot Q = (\overline{T \cdot \bar{Q}}) \cdot Q$ . Перемикання тригера

визначається сумісною дією лічильних сигналів "Т" і зворотного зв'язку виходів Q і  $\bar{Q}$ . Для виключення "гонок" у тригері сигнали зворотного зв'язку не повинні змінюватися під час дії лічильного сигналу "Т". Затримка вихідного сигналу може здійснюватися лінією затримки (в імпульсно-потенціальній системі елементів) чи додатковим тригером (в потенціальній системі елементів). Схема одноступеневого асинхронного Т-тригера на елементах НЕ І з логічними зв'язками відповідно до рівняння (3.3) показана на рис.3.14. Сигнали з виходів елементів D1 і D2 затримуються на час  $\Delta t$ , що дорівнює тривалості лічильного сигналу на Т-вході.

Таблиця 3.3

$T_t$	$Q_t$	$Q_{t+1}$
0	0	0
0	1	1
1	0	1
1	1	0

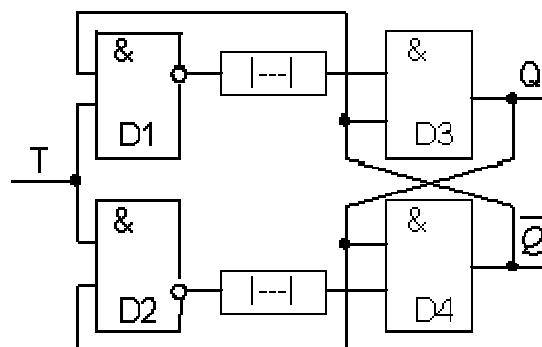


Рис.2.14. Схема одноступеневого Т-тригера

Надійні структури лічильних тригерів будують з використанням двоступеневих тригерів або тригерів з динамічним керуванням по фронту сигналу C.



### 3.7. Таблиця переходів і логічне рівняння D-тригера

Тригером типу D називається синхронний запам'ятовуючий елемент з двома стійкими станами і одним інформаційним D-входом. Закон функціонування D-тригера описується логічним рівнянням:

$$Q_{t+1} = CtDt.$$

Це рівняння показує, що після перемикавання стан D-тригера повторює значення сигналу на D-вході в тактові моменти часу. Тому в літературі D-тригери часто називають тригерами затримки (від Delay – затримка).

Схему D-тригера можна побудувати на основі синхронного RS-тригера, якщо сигнал по входу S одночасно подавати через інвертор на вхід R (рис.2.15, а). Схеми D-тригера будують також на основі самостійного логічного рівняння. Перетворимо рівняння (2.5) замінивши сигнал S на D і сигнал R на  $\bar{D}$ :

$$Q_{t+1} = \overline{C \cdot S \cdot C \cdot R} \cdot Q = \overline{C \cdot D \cdot C \cdot \bar{D}} \cdot Q. \quad (3.11)$$

Схема D-тригера на елементах НЕ І з логічними зв'язками згідно з рівнянням (3.11) показана на рис.3.15, б.

D-тригер "слідкує" за зміною сигналу на D-вході під час дії синхросигналу С і зберігає ту інформацію, яка була в момент його закінчення. RS-тригери такої властивості не мають і тому вони менше завадостійкі порівняно з D-тригерами.

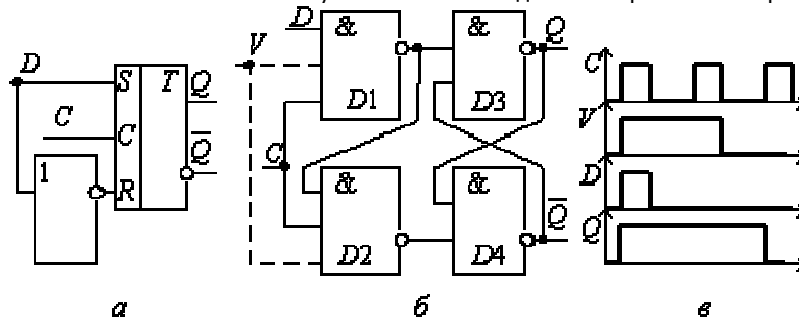


Рис.3.15. D-тригер: а – на основі RS-тригера; б – на елементах НЕ І; в – часові діаграми роботи

Для затримки інформації в D-тригері на довільне число тактів використовується дозволяючий V-вхід, як показано штриховою лінією на рис.3.15, б. Якщо  $V = 1$ , то DV-тригер функціонує як звичайний тригер затримки; якщо  $V = 0$ , то робота схеми за входами блокується і DV-тригер зберігає попередню інформацію.

Схема двоступеневого одноканального DV-тригера на елементах НЕ І із забороняючими зв'язками між ступенями показана на рис.3.16.

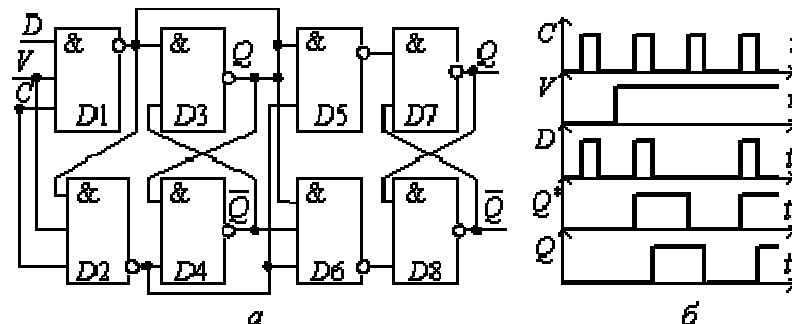


Рис.3.16. Двоступеневий DV-тригер: а – схема; б – часові діаграми

### 3.8. D-тригер з динамічним керуванням

В усіх синхронних тригерах із статичним керуванням (за рівнем) можливе фальшиве перемикання у випадку зміни сигналів на інформаційних входах під час дії синхросигналу С. Наприклад, якщо в D-тригері інформаційний сигнал змінює своє значення від одиниці до нуля до закінчення сигналу "С", то тригер може знову переключатися з одиничного в нульовий стан. Тому для надійної роботи D-тригера потрібний певний інтервал часу між фронтом синхроімпульсу С і спадом сигналу на D-вході (параметр  $t_{вдн}$ ).

У тригерах з динамічним керуванням записування інформації синхроімпульс С активний лише на короткому інтервалі часу в області фронту чи спаду. Тому D-тригери з динамічним керуванням мають високу завадостійкість. На практиці широке розповсюдження отримали D-тригери з прямим динамічним керуванням за схемою "трёх тригерів". Схема такого D-тригера з прямим динамічним керуванням показана на рис.3.17, а, а його умовне графічне зображення – на рис.3.17, б. Тут зберігання інформації здійснює основний вихідний синхронний RS-тригер (елементи D5 і D6) з інверсним керуванням, а приймання тактового і інформаційного сигналів і задання динамічного режиму роботи забезпечують два вихідних комутуючих тригери (елементи D1, D2, D3, D4).

Елемент D4 подає інверсне значення вхідного сигналу D на входи елементів D1 і D3 (рис.3.17, в). Елемент D1 повторює значення сигналу D.

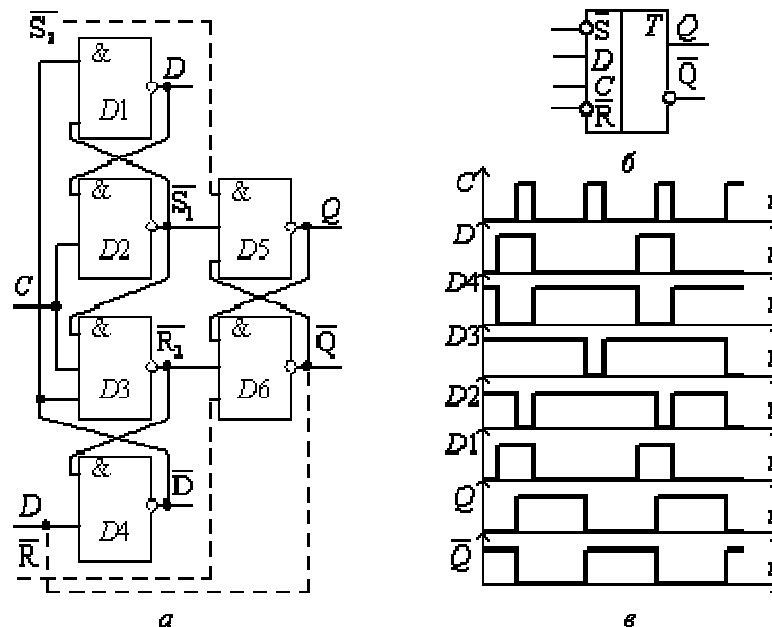


Рис.3.17. D-тригер з динамічним керуванням: а – схема; б – умовне позначення; в – часові діаграми

При  $C = D = 1$  вмикається елемент D2 і встановлює основний тригер за входом S1 в стан "1"; одночасно блокується робота елемента D3, в зв'язку з чим схема уже не реагує на зміну вхідного сигналу. При  $C = 1, D = 0$  вмикається елемент D3 і встановлює основний тригер в стан "0"; одночасно відбувається приймання нової інформації елементом D4. При  $C = 0$  забезпечується режим зберігання записаної інформації.

Час перемикання тригера за C-входом  $t_{п.т} = 3t_{р}$ .

Після закінчення сигналів на D- і C-входах починається етап відновлення, який характеризується переходом комутуючих тригерів у початковий стан за час  $3t_{р}$ . Максимальна частота перемикання D-тригера з динамічним керуванням визначається сумарним часом затримок  $f_{max} = 1/(6t_{р})$ .

<<

## Регістри

### 4.1. Загальна характеристика регістрів

Регістром називається типовий функціональний вузол комп'ютера, призначений для приймання, тимчасового зберігання, перетворення і видачі  $n$ -розрядного двійкового слова. Регістр містить регулярний набір однотипних тригерів, в кожному з яких зберігається значення одного двійкового розряду машинного слова. Найчастіше використовують тригери типів RS, JK і D (рис. 4.1).

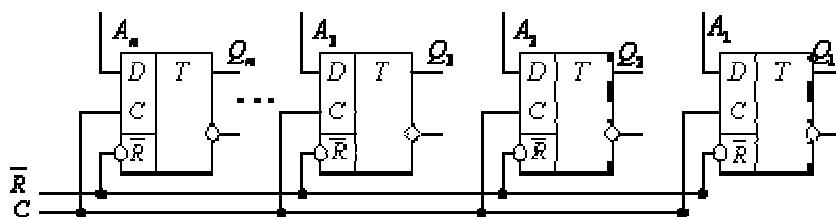


Рис. 4.1 Схема регістра на D-тригерах

Регістри, призначені тільки для приймання (записування), зберігання і передачі інформації, називаються елементарними або фіксаторами. Регістри, в яких зберігання даних поєднується з мікроопераціями зсуву, називаються зсувовими. Елементарні регістри будують на одноступеневих тригерах, а зсувові – на двоступеневих або D-тригерах з динамічним керуванням. Логічна функція регістра позначається буквами RG (register).

Регістри забезпечують зберігання команд, адреси пам'яті, результатів операцій, індексів та ін.

Регістри класифікують за такими ознаками:

способом керування записуванням – асинхронні та синхронні;

способом записування і видачі двійкових слів – паралельні, послідовні й універсальні; у паралельних регістрах записування і видача слів виконується одночасно всіма розрядами, а в послідовних – розряд за розрядом в напрямку від молодших розрядів до старших або навпаки; універсальні регістри забезпечують як паралельний, так і послідовний обмін інформацією;

числом ліній для представлення значення одного розряду слова (біта інформації) – однофазні й парафазні; при однофазному поданні значення кожного розряду слова передається по одній лінії зв'язку, а при парафазному – по двох лініях (одночасно відображається пряме та інверсне значення розряду);

числом тактів для записування слова – одно-, дво- і багатотактні;

складом мікрооперацій, які виконуються: установлювальні, записування, читання, порозрядні логічні й зсуву, а також перетворення послідовного коду в паралельний і навпаки;

напрямом зсуву – односторонні (лівий або правий зсув) і двосторонні (реверсивні);

типом тригерів, що використовуються;  
елементною структурою – потенціальні, імпульсні й потенціально-імпульсні.

#### 4.2. Установлювальні мікрооперації. Однофазний і парафазний спосіб записування інформації

Установлювальні мікрооперації служать для переключення регістрів у певний стан. Наприклад, установлення регістра в стан "0" або "1"; установлення парних розрядів у стан "0", а непарних – у стан "1"; записування в регістр деякої константи або обнуління деяких байтів та ін.

Установлювальні мікрооперації переважно використовують асинхронні входи регістра, наприклад, вхід скидання R для схеми, зображеної на рис. 4.1.

У регістрах на RS- або JK- тригерах можливий однофазний або парафазний спосіб записування інформації. При однофазному записуванні значення кожного розряду слова  $A = A_n A_{n-1} \dots A_i \dots A_1$  надходить по одній лінії зв'язку на вхід S (або J) відповідних тригерів. Після зчитування записаної інформації регістр має обнулятися по спільному R входу (рис. 4.2). Таким чином, при однофазному записуванні частота обміну інформацією зменшується, оскільки процеси введення і скидання чергуються.

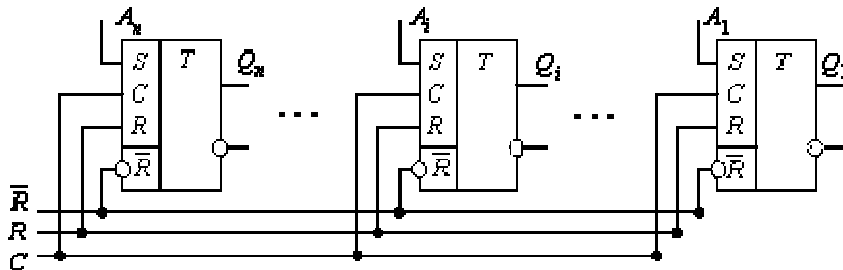


Рис.4.2 Схема регістра з однофазним записом даних

При парафазному записуванні інформації значення кожного розряду слова  $A$  передається по двох лініях зв'язку. При цьому пряме значення  $A_i$  надходить на вхід S (або J) відповідних тригерів, а інверсне значення  $\overline{A_i}$  – на вхід R (або K). У цьому випадку не потрібне попереднє скидання регістра в стан "0", тому що таку функцію виконує сигнал  $\overline{A_i}$  (рис.4.3).

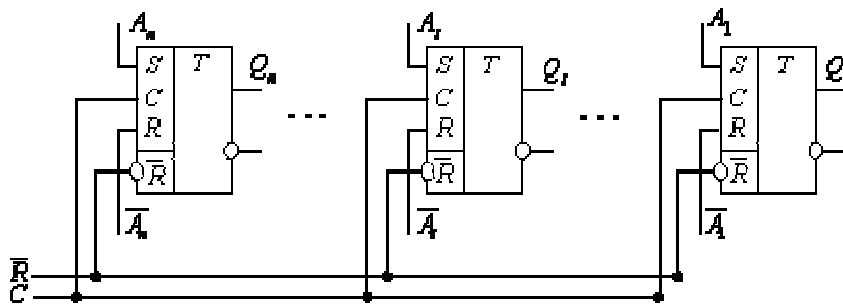


Рис.4.3 Схема регістра з парафазним записом даних

У регістрах на D-тригерах, які мають один інформаційний вхід, можливий тільки однофазний спосіб записування інформації. З приходом чергового синхроімпульсу записується нова інформація або при відсутності даних на вхідній шині стан регістра автоматично обнуляється (див. рис.4.1).  
При необхідності збереження інформації на декілька тактів у регістрах на D-тригерах потрібно використати дозволяючий V-вхід, або блокувати проходження синхроімпульсу на C-вхід.

#### 4.3. Записування інформації від двох джерел

Для записування інформації від декількох джерел (напрямків) на вході кожного тригера ставлять додаткові комбінаційні схеми, які створюють вхідну логіку регістра. Кожний напрямок має свою сукупність електричних ліній (шину), по яких передаються сигнали, що відображають значення розрядів слова. Якщо  $n$ -розрядне слово передається однофазним кодом, то шина має  $n$  ліній зв'язку, а якщо парафазним кодом – то  $2n$  ліній. Записування кожного слова ініціюється відповідним керуючим сигналом  $Y1, Y2$  та ін.

Для записування в регістр на RS-тригерах однофазним кодом слів  $A$  і  $B$  потрібно реалізувати такі порозрядні функції збудження входу  $S_i$ :

$$S_i = Y1A_i \cup Y2B_i, \quad (4.1)$$

де  $A_i$  і  $B_i$  – двійкові розряди слів  $A$  і  $B$ ;  $Y1$  і  $Y2$  – сигнали керування приймання слів  $A$  і  $B$  відповідно. Схема вхідної логіки  $i$ -го розряду регістра на основі рівняння (4.1) показана на рис.4.4, а.

Для записування в регістр на JK-тригерах парафазним кодом слів  $A$  і  $B$  потрібно реалізувати такі порозрядні функції збудження входів  $J_i$  і  $K_i$ :

$$J_i = Y1A_i \cup Y2B_i; \quad K_i = Y1\overline{A_i} \cup Y2\overline{B_i}. \quad (4.2)$$

Схема вхідної логіки  $i$ -го розряду регістра на основі рівнянь (3.2) показана на рис.4.4, б.

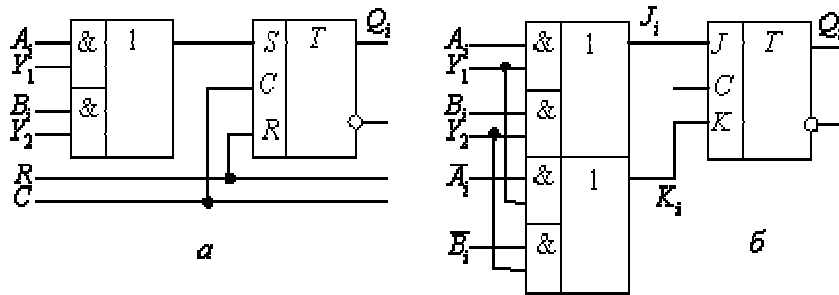


Рис. 4.4 Схеми розряду регістра із записом слів від двох джерел: а – однофазним кодом; б – парафазним кодом

#### 4.4. Зчитування інформації

Інформація, яка зберігається в регістрах, може передаватися у зовнішні схеми однофазним або парафазним способом у прямому або оберненому коді. Для реалізації мікрооперацій зчитування до виходів кожного тригера підключаються комбінаційні схеми, які створюють вихідну логіку регістра.

Схеми вихідної логіки будуються на основі таких порозрядних логічних рівнянь: для зчитування однофазним прямим або оберненим кодом

$$\text{Ші} = \text{Yпр}Q_i \cup \text{Yoб} \overline{Q_i}; \quad (4.3)$$

для зчитування парафазним прямим або оберненим кодом

$$\text{Ші}^* = \text{Yпр}Q_i \cup \text{Yпр} \overline{Q_i}; \quad \overline{\text{Ші}}^* = \text{Yoб} \overline{Q_i} \cup \text{Yoб} Q_i; \quad (4.4)$$

де Yпр і Yoб – керуючі сигнали видачі відповідно прямого або оберненого коду;  $Q_i$  і  $\overline{Q_i}$  – пряме та інверсне значення ви-

ходу  $i$ -го розряду регістра; Ші – розряд однофазної шини даних; Ші\* і  $\overline{\text{Ші}}^*$  – розряди парафазної шини даних.

Очевидно, що керуючі сигнали Yпр і Yoб не повинні збігатися в часі. Наприклад, при зчитуванні інформації парафазним оберненим кодом отримаємо:

$$\text{Yпр} = 0; \text{Yoб} = 1; \text{Ші}^* = \overline{Q_i}; \overline{\text{Ші}}^* = Q_i.$$

Схеми вихідної логіки для  $i$ -го розряду на основі рівнянь (4.3) і (4.4) показані на рис. 4.5.

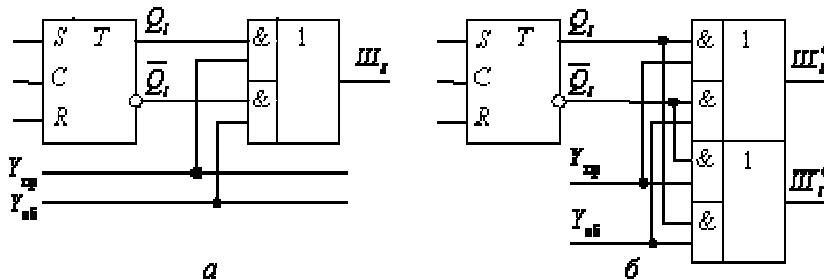


Рис. 4.5 Схеми вихідної логіки  $i$ -го розряду регістра для зчитування інформації: а – однофазним кодом; б – парафазним кодом

#### 4.5. Логічні мікрооперації в регістрах

У регістрах можуть виконуватися такі порозрядні (без перенесень) логічні мікрооперації над словами А і В: логічне додавання і множення:  $\text{RG1} := A \cup B$ ;  $\text{RG1} := A \cap B$ ; додавання за модулем два і його заперечення:  $\text{RG1} := A \oplus B$ ;  $\text{RG1} := \overline{A \oplus B}$ ; інверсія слова:  $\text{RG1} := \overline{A}$ .

Логічні мікрооперації передбачають наявність першого слова А в регістрі. З урахуванням цього логічне додавання слів А і В в регістрі на RS- або JK-тригерах з однофазним записом виконується введенням слів В без попереднього скидання. Логічне множення реалізується подаванням інверсних значень розрядів слова В на входи R (або K) тригерів регістра. Дійсно, якщо значен-

ня  $B_i = 0$ , то  $\overline{B_i} = 1$  і відповідно тригери обнуляються, що і потрібно для порозрядного логічного множення.

Мікрооперації додавання за модулем два і його заперечення реалізуються в регістрах на Т-тригерах. Спочатку записується слово А, а потім без попереднього скидання по лічильному входу вводиться слово В. Після цього на прямих виходах тригерів

фіксується результат операції  $Q = A \hat{\wedge} B$ , а на інверсних виходах –  $\overline{Q} = \overline{A \hat{\wedge} B}$ . Мікрооперація інвертування складається з подавання імпульсу на всі Т-входи тригерів регістра, в яких зберігається слово А. У підсумку на прямих виходах тригерів встанов-

люється результат згідно із співвідношенням  $Q_i = A_i \hat{\wedge} 1 = \overline{A_i}$ .

#### 4.6. Мікрооперації зсуву

Зсув – це одночасне просторове переміщення двійкового слова в розрядній сітці із збереженням порядку слідування нулів і одиниць. Регістри, призначені для виконання мікрооперацій зсуву, називаються регістрами зсуву або зсувовими.

Мікрооперації зсуву використовують у процесі виконання команд множення, ділення і нормалізації. Крім того, за допомогою зсуву здійснюється перетворення паралельного коду в послідовний або навпаки (наприклад, при обміні інформацією з магнітними стрічками і дисками).

Зсув слова може виконуватися вправо (у бік молодших розрядів) або вліво (у бік старших розрядів). Позначимо однорозрядні

мікрооперації зсуву вправо і вліво символами R і L відповідно. Розрізняють правий і лівий арифметичний (Ra, La), логічний (Rл, Lл) і циклічний (Rц, Lц) зсуви слова.

Нехай в регістрі A записано слово  $A_n A_{n-1} \dots A_2 A_1$ , де  $A_1$  – молодший розряд;  $A_n$  – старший розряд. Символічно мікрооперації зсуву записуються таким чином:

арифметичні зсуви (знаковий розряд не зсувається):

$RGA: = Ra(A) = A_n 0 A_{n-1} \dots A_2$ ;  $RGA: = La(A) = A_n A_{n-2} \dots A_1 0$ ;

логічні зсуви (одночасно зсуваються всі розряди):

$RGA: = Rл(A) = 0 A_n A_{n-1} \dots A_2$ ;  $RGA: = Lл(A) = A_{n-1} A_{n-2} \dots A_1 0$ ;

циклічні зсуви (між старшим і молодшим розрядами є кільцевий зв'язок):

$RGA: = Rц(A) = A_1 A_n A_{n-1} \dots A_2$ ;  $RGA: = Lц(A) = A_{n-1} A_{n-2} \dots A_1 A_n$ .

Арифметичні та циклічні зсуви переважно використовують при виконанні команд в процесорах, а логічні зсуви забезпечують перетворення послідовного коду в паралельний і навпаки в пристроях зв'язку з магнітними стрічками і дисками.

Зсувні регістри проєктують на двоступеневих RS- (або JK-) або D-тригерах з динамічним керуванням по фронту. Такі тригери забезпечують розділення під час процесів приймання нової інформації в кожному розряді та видачі (зсуву) старої.

Припустимо, що в реверсивному регістрі мають виконуватися паралельний запис слова A за сигналом керування  $Y_{зп}$  і зсуви інформації вліво і вправо під впливом керуючих сигналів Lл і Rл. Функції збудження Si і Ri для кожного розряду регістра зсуву на двоступеневих RS-тригерах мають вигляд:

$$S_i = Y_{зп} A_i \bar{U} R_l Q_i + 1 \bar{U} L_l \bar{Q}_i - 1; \quad R_i = Y_{зп} \bar{A}_i \bar{U} R_l \bar{Q}_{i+1} \bar{U} L_l \bar{Q}_{i-1}. \quad (4.5)$$

Схема одного розряду регістра зсуву на RS-тригерах згідно з рівнянням (3.5) показана на рис.4.6,а.

Економічна схема розряду регістра, в якій функція на R-вході тригера утвориться інвертуванням сигналу Si, показана на рис.4.6, б. Однак у цьому випадку час записування даних збільшується за рахунок затримки інвертора.

Для регістра зсуву на D-тригерах функції збудження мають вигляд:

$$D_i = Y_{зп} A_i \bar{U} R_l \bar{Q}_i + 1 \bar{U} L_l \bar{Q}_{i-1}. \quad (4.6)$$

Схема розряду регістра зсуву на D-тригерах згідно з рівнянням (4.6) показана на рис.4.6, в.

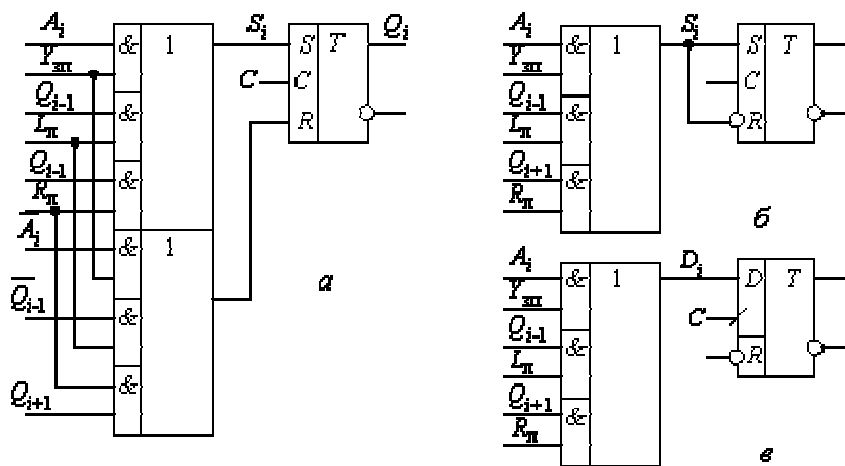


Рис.4.6 Схема розряду регістра зсуву: а – на RS-тригері; б – з інвертором на вході R; в – на D-тригері

Приклад побудови реверсивного трирозрядного регістра зсуву на D-тригерах з динамічним керуванням показаний на рис.4.7.

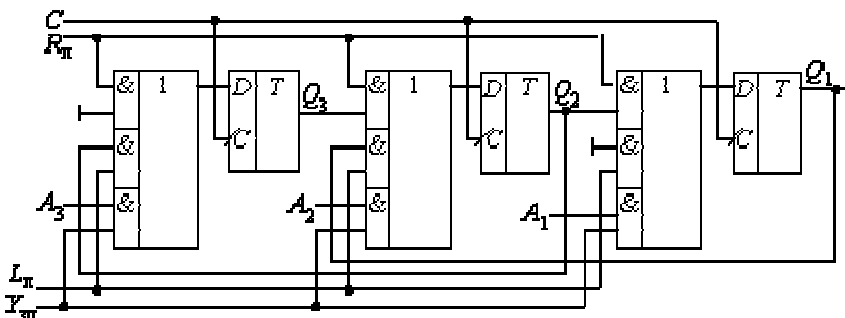


Рис.4.7 Схема реверсивного регістра зсуву на D-тригерах

Реверсивний регістр зсуву працює таким чином. При значенні сигналу  $Y_{зп}=1$  в регістр записується інформація паралельним однофазним кодом. При значенні сигналу  $Rл=1$  інформація, що зберігається, одночасно зсувається у бік молодших розрядів, при цьому розряд  $Q_3$  обнуляється. При значенні сигналу  $Lл=1$  інформація в регістрі одночасно зсувається у бік старших розрядів, при цьому розряді  $Q_1$  обнуляється. Запис і зсув інформації відбуваються по фронту синхроімпульсу.

### 3.7. Перетворення послідовного коду в паралельний і навпаки

Схема чотирирозрядного регістра зсуву вправо на JK-тригерах, яка забезпечує перетворення кодів, показана на рис.4.8, а.

Старший розряд регістра за допомогою інвертора на К-вході працює в режимі D-тригера.

Нехай від накопичувачів на магнітних дисках або стрічках на вхід регістра по лінії D поступає послідовний код слова  $A=1101$  в напрямку від молодших розрядів до старших. Значення розрядів слова поступає одночасно із синхроімпульсами, які забезпечують як приймання коду в старший розряд, так і одночасний зсув вмісту регістра вправо (рис.4.8, б). Після приходу чотирьох синхроімпульсів на виходах регістра  $Q_4 - Q_1$  встановлюється код 1101. Таким чином здійснюється перетворення послідовного коду в паралельний, яке часто називають послідовним введенням слова в регістр.

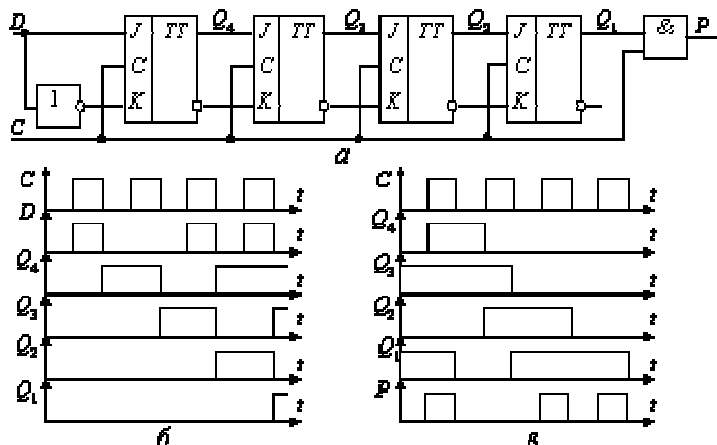


Рис.4.8 Регістр зсуву : а – схема ; б,в – перетворення послідовного коду в паралельний і навпаки

Перетворення паралельного коду в послідовний також відбувається зсувом слова, яке зберігається. Процес перетворення паралельного коду слова  $A=1101$  в послідовний в напрямку від молодших розрядів до старших за допомогою зсуву вправо показаний на рис.4.8, в.

## Лічильники

### 5.1. Загальна характеристика лічильників

Лічильником називається типовий функціональний вузол комп'ютера, призначений для лічби входних імпульсів. Лічильник являє собою зв'язаний ланцюг Т-тригерів, які утворюють пам'ять із заданим числом сталих станів (рис. 5.1).

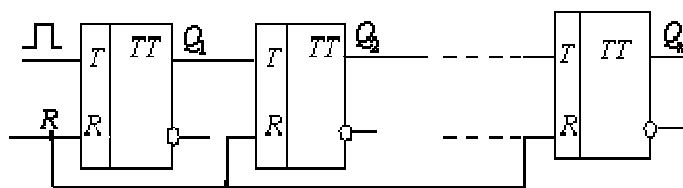


Рис.5.1. Логічна структура лічильника

Розрядність лічильника  $n$  дорівнює числу Т-тригерів. Кожний вхідний імпульс змінює стан лічильника, який зберігається до надходження наступного сигналу. Значення виходів тригерів лічильника  $Q_n, Q_{n-1}, \dots, Q_1$  відображають результат лічби в прийнятій системі числення. Логічна функція лічильника позначається буквами СТ (counter).

Список мікрооперацій лічильника вміщує попереднє встановлення в початковий стан, інкремент або декремент слова, яке зберігається, видачу слів паралельним кодом та ін.

Вхідні імпульси можуть надходити на лічильник як періодично, так і довільно розподіленими у часі. Амплітуда і тривалість лічильних імпульсів мають задовольняти технічні вимоги для серій мікросхем, які використовуються.

Лічильник є одним з основних функціональних вузлів комп'ютера, а також різних цифрових керуючих та інформаційно-вимірювальних систем. Основне застосування лічильників:

утворення послідовності адрес команд програми (лічильник команд або програмний лічильник);  
 підрахунок числа циклів при виконанні операцій ділення, множення, зсуву (лічильник циклів);  
 одержання сигналів мікрооперацій і синхронізації; аналого-цифрові перетворення і побудова електронних таймерів (годинників реального часу).

Лічильник характеризується модулем і ємністю лічби. Модуль лічби КЛЧ визначає число станів лічильника. Модуль двійкового  $n$ -розрядного лічильника визначається цілим степенем двійки  $M=2^n$ ; в лічильниках інших типів справедлива нерівність  $КЛЧ \leq M$ . Після лічби числа імпульсів  $NBX=КЛЧ$  лічильник повертається в початковий стан. Таким чином, модуль лічби, який часто називають коефіцієнтом перерахунку, визначає цикл роботи лічильника, після чого його стан повторюється. Тому число вхідних імпульсів і стан лічильника однозначно визначені тільки для першого циклу.

Ємність лічби  $N_{max}$  визначає максимальну кількість вхідних імпульсів, яку може зафіксувати лічильник при одному циклі роботи. Ємність лічби  $N_{max}=КЛЧ - 1$  за умови, що робота лічильника починається з нульового початкового стану.

У лічильниках використовуються три режими роботи: керування, накопичення і ділення. У режимі керування зчитування інформації виконується після кожного вхідного лічильного імпульсу, наприклад, в лічильнику адреси команд. У режимі накопичення головним є підрахунок заданого числа імпульсів або лічба протягом певного часу. У режимі ділення (перерахунку) основним є зменшення частоти надходження імпульсів в КЛЧ разів. Більшість лічильників може працювати в усіх режимах, проте в спеціальних лічильниках-дільниках стани в процесі лічби можуть змінюватися в довільному порядку, що дозволяє спростити схему вузла.

Лічильники класифікують за такими ознаками:

способом кодування – позиційні та непозиційні;  
 модулем лічби – двійкові, десяткові, з довільним постійним або змінним (програмованим) модулем;  
 напрямком лічби – прості (підсумовуючі, віднімальні) і реверсивні;  
 способом організації міжрозрядних зв'язків – з послідовним, наскрізним, паралельним і комбінованим переносами (позицією);  
 типом використовуваних тригерів – Т, JK, D в лічильному режимі;  
 елементним базисом – потенціальні, імпульсні та потенціально-імпульсні.

У лічильниках з позиційним кодуванням числовий вираз поточного стану лічильника визначається за формулою:

$$N = \sum_{i=1}^n r_i Q_i$$

$$Q_i = r_n Q_n + r_{n-1} Q_{n-1} + \dots + r_1 Q_1,$$

де  $r_i$  – вага  $i$ -го розряду;  $Q_i$  – значення виходу  $i$ -го розряду;  $n$  – число розрядів. Нульове значення всіх розрядів звичайно беруть як початковий стан лічильника. Всі інші стани нумерують за числом вхідних імпульсів, що надійшли.

У лічильниках з непоозиційним кодуванням (наприклад, у кодах Грея) розряди не мають постійних ваг і кожному набору станів  $Q_n, Q_{n-1}, \dots, Q_1$  приписується певна кількість вхідних імпульсів. У комп'ютерах переважно використовують лічильники з позиційним кодуванням.

За видом переходів прості лічильники (ЛЧ) розподіляються на підсумовуючі (прямої лічби) і віднімальні (зворотної лічби). У підсумовуючих лічильниках кожний доданий імпульс  $U+$  збільшує стан на одиницю, тобто реалізується мікрооперація інкремента  $Lч := Lч + 1$ . Граф переходів підсумовуючого лічильника показаний на рис.5.2, а, де вершини означають стійкі стани. Із  $M-1$ -го стану черговий сигнал  $U+$  повертає лічильник у початковий стан і видає сигнал переповнення  $P$ .

У віднімальних лічильниках кожний віднімальний імпульс  $U-$  зменшує стан на одиницю, тобто реалізується мікрооперація декремента  $Lч := Lч - 1$ . Граф переходів віднімального лічильника (рис. 5.2, б) характеризується наявністю переходів тільки в зворотному напрямку від деякого (наприклад,  $M-1$ -го) попередньо встановленого стану. Після віднімання  $M$  імпульсів лічильник видає сигнал позики  $Z$  і повертається в початковий  $M-1$ -й стан.

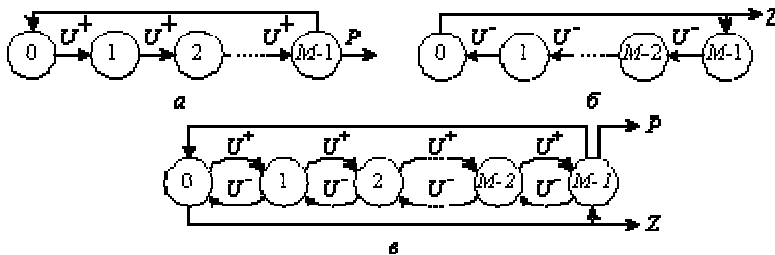


Рис.5.2. Графи лічильників: а – підсумовуючого; б – віднімального; в – реверсивного

Реверсивні лічильники мають переходи в прямому і зворотному напрямках, що дозволяє рахувати підсумовуючі та віднімальні імпульси (рис.5.2, в). У процесі лічби має виконуватися умова  $\Delta U+ + N_p \geq \Delta U-$ , де  $N_p$  – попередньо записане число. За поточним станом виходів лічильника визначається результат реверсивної лічби:

$$DN = \Delta U+ + N_p - \Delta U-.$$

До часових характеристик лічильників відносяться роздільна здатність, швидкодія і час встановлення (перемикання) коду.

Роздільна здатність  $trз$  визначається мінімальним інтервалом часу між двома вхідними імпульсами, при якому ще зберігається працездатність лічильника. Параметр  $trз$  задають часом перемикання  $tT$  першого (молодшого) тригера лічильника, тобто  $trз = tT$ , оскільки він перемикається під дією кожного вхідного імпульсу.

Швидкодія лічильника визначається максимальною частотою  $F_m$  надходження вхідних імпульсів в режимі ділення й обчислюється за формулою  $F_m = 1/tT$ . Час встановлення коду  $twст$  відраховується від початку вхідного імпульсу до моменту отримання нового стану. Даний параметр дозволяє обчислювати швидкодію лічильника в режимі керування із співвідношення  $F_m.k = 1/(twст + tзч)$ , де  $tзч$  – час зчитування інформації.

Міжрозрядні зв'язки забезпечують вироблення сигналів перенесення в старші розряди при додаванні імпульсів і сигналів позики – при відніманні. Від виду реалізації міжрозрядних зв'язків суттєво залежать параметри  $twст$  і  $F_m.k$ .

У лічильниках з послідовними перенесеннями тригери перемикаються по чергові після кожного вхідного імпульсу в напрямку від молодших розрядів до старших. Такі лічильники називаються послідовними або асинхронними. У лічильниках з паралельними перенесеннями тригери перемикаються одночасно після кожного вхідного імпульсу, такі лічильники називаються паралельними або синхронними.

## 5.2. Двійкові підсумовуючі та віднімальні лічильники

Двійкові лічильники реалізують лічбу вхідних імпульсів у двійковій системі числення.

Число розрядів  $n$  двійкового підсумовуючого лічильника для заданого модуля  $M$  знаходять із виразу  $n = \log_2 M$ . Значення поточного числа  $N+$  вхідних імпульсів  $n$ -розрядного підсумовуючого лічильника при відліку з нульового початкового стану визначають за формулою

$$N+ = \sum_{i=1}^n 2^{i-1} Q_i$$

$$Q_i = 2^{n-1} Q_n + 2^{n-2} Q_{n-1} + \dots + 2^0 Q_1,$$

де  $2^{i-1}$  – вага  $i$ -го розряду;  $Q_i \in \{0,1\}$  – логічне значення прямого виходу тригера  $i$ -го розряду. Розряди двійкового лічильника будуються на двоступеневих Т-тригерах або D-тригерах з динамічним керуванням по фронту синхросигналу (в лічильному режимі).

У двійковому підсумовуючому лічильнику перенесення  $P_i$  в сусідній старший розряд  $Q_{i+1}$  виникає в тому випадку, коли в момент надходження чергового лічильного імпульсу  $U+$  всі молодші розряди знаходяться в одиничному стані, тобто  $P_i = U+ \cdot Q_i \cdot Q_{i-1} \cdot \dots \cdot Q_1 = 1$ . Після вироблення перенесення старший розряд перемикається в стан «1», а всі молодші розряди – в стан «0».

Асинхронні підсумовуючі лічильники на двоступеневих Т-тригерах будуються так, щоб вхідні імпульси  $U+$  надходили на лічильний вхід тільки першого (молодшого) розряду. Сигнали перенесення передаються асинхронно (послідовно в часі) з прямих виходів молодших розрядів на Т-входи сусідніх старших, як показано на рис.3.14 для трирозрядного лічильника.

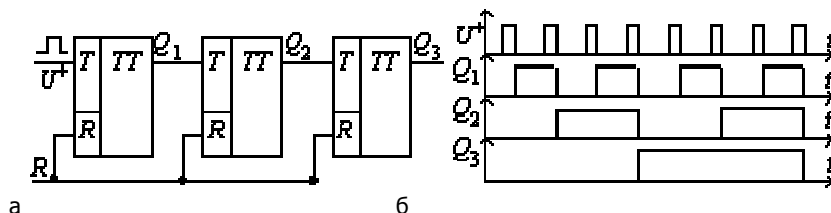


Рис.5.3. Асинхронний підсумовуючий лічильник на двоступеневих Т-тригерах: а – схема; б – часові діаграми роботи

Зміна станів тригерів відбувається за спадом лічильного імпульсу для першого розряду, а для останніх – за спадом сигналу перенесення (рис.5.3, б).

Після підрахунку семи імпульсів на виході трирозрядного лічильника установлюється двійковий код  $Q_3Q_2Q_1=111$  (тобто максимальне значення або ємність лічби). Після приходу восьмого вхідного імпульсу  $U+$  трирозрядний підсумовуючий лічильник перемикається у початковий нульовий стан послідовно (асинхронно) в часі: спочатку спадає напруга на виході  $Q_1$ , потім – на виході  $Q_2$  і т.д.

За допомогою імпульсу по входу скидання  $R$  лічильник повертається в нульовий стан у будь-який момент часу.

Часові параметри  $n$ -розрядного асинхронного лічильника на двоступеневих тригерах визначаються із співвідношень:

роздільна здатність  $tr_z = tT_1 + tT_2 = 2tT$ , де  $tT_1$  і  $tT_2$  – час перемикавання першого і другого ступенів тригера, причому  $tT_1 = tT_2 = tT$ ;

час встановлення  $tw_{ст} = tT_1 + ntT_2 \gg ntT$ ;

максимальна частота лічби в режимі ділення і керування відповідно:  $F_m = 1/2tT$ ;  $F_{m.к} \gg 1/ntT$ .

В режимі керування швидкодія асинхронного лічильника приблизно в  $n$  раз менша ніж в режимі ділення.

Схема трирозрядного асинхронного двійкового підсумовуючого лічильника на Т-тригерах з динамічним керуванням по фронту показана на рис.5.4. Лічильні імпульси  $U+$  надходять на Т-вхід тільки першого (молодшого) розряду; наступні тригери перемикаються асинхронно від сигналів перенесення з інверсних виходів сусідніх молодших розрядів.

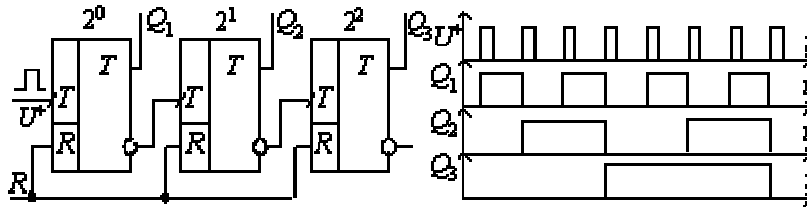


Рис.5.4. Асинхронний підсумовуючий лічильник на тригерах з динамічним керуванням по фронту: а – схема; б – часові діаграми роботи

Часові параметри асинхронного  $n$ -розрядного двійкового лічильника на тригерах з динамічним керуванням:  $tr_z = tT$ ,  $tw_{ст} = ntT$ ,  $F_m = 1/tT$ ,  $F_{m.к} \gg 1/ntT$ .

Перевагою асинхронних лічильників є простота схеми: збільшення розрядності виконується підключенням необхідного числа тригерів.

До недоліків асинхронних лічильників відносяться порівняно низька швидкодія в режимі керування та її залежність від числа розрядів, а також поява проміжних вихідних двійкових кодів у процесі послідовного перемикавання тригерів у новий стан.

Для одержання мінімального часу перемикавання лічильника використовують паралельні перенесення (рис.5.5). Для цього в кожному розряді синхронного лічильника є схема збігу, за допомогою якої аналізуються стани всіх попередніх молодших тригерів і вироблюються функції перенесення згідно з такими логічними співвідношеннями:

$P_1 = U + Q_1$ ;  $P_2 = U + Q_2Q_1$ ;  $P_3 = U + Q_3Q_2Q_1$ ;  $P_4 = U + Q_4Q_3Q_2Q_1$ .

При надходженні чергового лічильного імпульсу  $U+$  перемикаються тільки ті тригери, для яких усі попередні (молодші) розряди знаходяться в цей момент в одиничному стані.

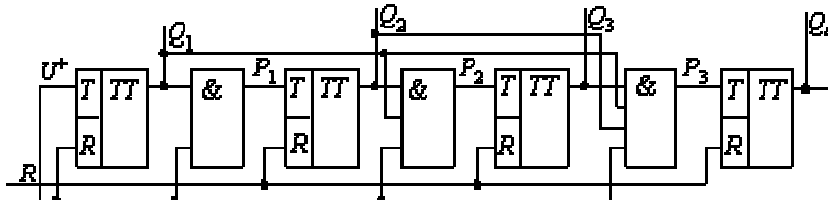


Рис.5.5. Схема підсумовуючого лічильника з паралельними перенесеннями

Час установлення синхронного лічильника не залежить від числа розрядів і дорівнює  $tw_{ст} = tTT + tP$ , де  $tTT$  – час перемикавання двоступеневого тригера;  $tP$  – час затримки розповсюдження сигналу вентилем у ланцюзі перенесення. Максимальна частота лічби в режимі керування  $F_m \gg 1/tTT$ . Таким чином, синхронні лічильники забезпечують найбільшу швидкодію в режимі керування.

При побудові багаторозрядних синхронних лічильників з'являються труднощі – зростання числа входів вентилів у ланцюзі перенесення і збільшення навантаження на виходи тригерів.

У двійковому віднімальному лічильнику кожний віднімальний імпульс  $U-$  зменшує стан на одиницю. Поточне значення кількості вхідних імпульсів  $n$ -розрядного двійкового віднімального лічильника визначається за формулою

$\delta U = N_p - N -$ ,

де  $N$  – значення коду на прямих виходах тригерів лічильника;

$N_p$  – попередньо записане початкове число; при цьому має виконуватися умова  $\delta U \geq N_p$ .

У віднімальних лічильниках сигнали міжрозрядного зв'язку називаються позиками. За правилом двійкового віднімання в момент надходження лічильного імпульсу  $U-$  позика із старшого розряду з одиничним значенням виникає за умови, що всі молодші тригери знаходяться в нульовому стані. Після цього всі вони перемикаються в стан «1», а старші – в стан «0». Сигнали позики утворюються на інверсних виходах двоступеневих тригерів або на прямих виходах тригерів з динамічним керуванням по фронту.

Параметри віднімального лічильника (модуль і ємність лічби, швидкодія) збігаються з аналогічними характеристиками підсумовуючих лічильників.

Схема трирозрядного двійкового асинхронного віднімального лічильника на двоступеневих тригерах показана на рис.5.6, а.

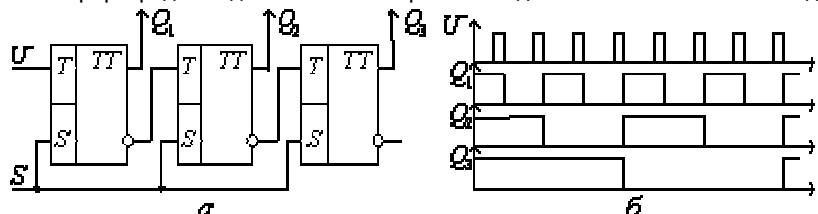




Рис.5.6. Асинхронний віднімальний лічильник на двоступеневих тригерах: а – схема; б – часові діаграми роботи

Перед початком роботи за допомогою сигналу на спільному вході S всі тригери лічильника встановлюються в стан "1", утворюючи вихідний код 111. Віднімальний імпульс U- надходить на лічильний вхід лише першого молодшого розряду, міжрозрядні сигнали позики знімаються асинхронно з інверсних виходів тригера.

Після надходження семи віднімальних імпульсів усі тригери лічильника устанавлюються в стан «0», утворюючи вихідний код 000. Восьмий віднімальний імпульс перемикає лічильник в стан 111 (за умови, що розглядається трирозрядна схема).

У віднімальному лічильнику на тригерах з динамічним керуванням по фронту сигнали позики знімаються з прямих виходів тригерів.

### 5.3. Двійкові реверсивні лічильники

Двійкові реверсивні лічильники мають переходи у двох напрямках: в прямому (при лічбі підсумовуючих сигналів U +) і в зворотному (при переліку віднімальних сигналів U -). Поточне значення різниці підрахованих імпульсів визначається із співвідношення

$$\delta U + - \delta U - = N - N_n,$$

де N – значення коду на прямих виходах тригерів лічильника;

N<sub>n</sub> – попередньо записане в лічильник початкове число. В лічбі має виконуватися умова  $\delta U - \leq N_n + \delta U + \leq 2n-1$ .

Розрізняють одноканальні та двоканальні реверсивні лічильники. В одноканальних реверсивних лічильниках підсумовуючі U + і віднімальні U - сигнали почергово надходять на спільний лічильний вхід, а напрямком лічби задається напрямком ланцюгів міжрозрядних перенесень або позик. Для перемикання міжрозрядних зв'язків у одноканальному реверсивному лічильнику потрібні додаткові керуючі сигнали.

Двоканальні реверсивні лічильники мають два лічильних входи: один для підсумовуючих імпульсів U +, другий – для віднімальних U -. Перемикання ланцюгів міжрозрядних зв'язків здійснюється автоматично лічильними сигналами: для переносів – імпульсами U +, для позики – імпульсами U -. Схема одноканального трирозрядного двійкового реверсивного лічильника показана на рис.3.18. Міжрозрядні зв'язки комутуються за допомогою логічних елементів І ЧИ.

Для задання напрямку лічбі використовують додатковий RS-тригер: з його прямого виходу знімається сигнал керування додаванням УД (вмикає ланцюги перенесення), а з інверсного виходу – сигнал керування відніманням УВ (вмикає ланцюги позики). На виходах елементів І ЧИ (які називаються "схеми реверса") виробляється сигнал Ті для лічильних входів старших розрядів:

$$T_i = Y_A Q_i \vee Y_B \overline{Q_i}, \quad i = 1, 2, 3, \dots, n.$$

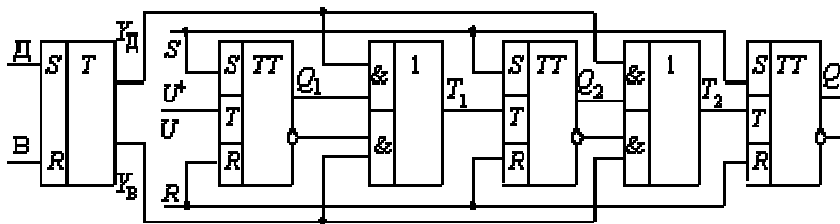


Рис.5.7. Схема одноканального реверсивного лічильника

Таким чином, якщо керуючий RS-тригер знаходиться в стані «1», то лічильник реалізує режим прямої лічби вхідних імпульсів (тобто підсумовування), в іншому випадку – забезпечує режим зворотної лічби (віднімання). В обох режимах роботи тригери перемикаються асинхронно.

Схема двоканального чотирирозрядного двійкового реверсивного лічильника показана на рис.5.8. Лічильні Т-входи в тригерах внутрішньо зв'язані схемою ЧИ.

Підсумовуючі імпульси U+ надходять на лічильний вхід першого (молодшого) розряду лічильника і одночасно – на входи всіх вентилів у ланцюзі паралельного перенесення. При цьому формуються імпульси міжрозрядних перенесень на основі логічних виразів:

$$P_1 = U^+ Q_1; P_2 = U^+ Q_1 Q_2; P_3 = U^+ Q_1 Q_2 Q_3; P_4 = U^+ Q_1 Q_2 Q_3 Q_4.$$

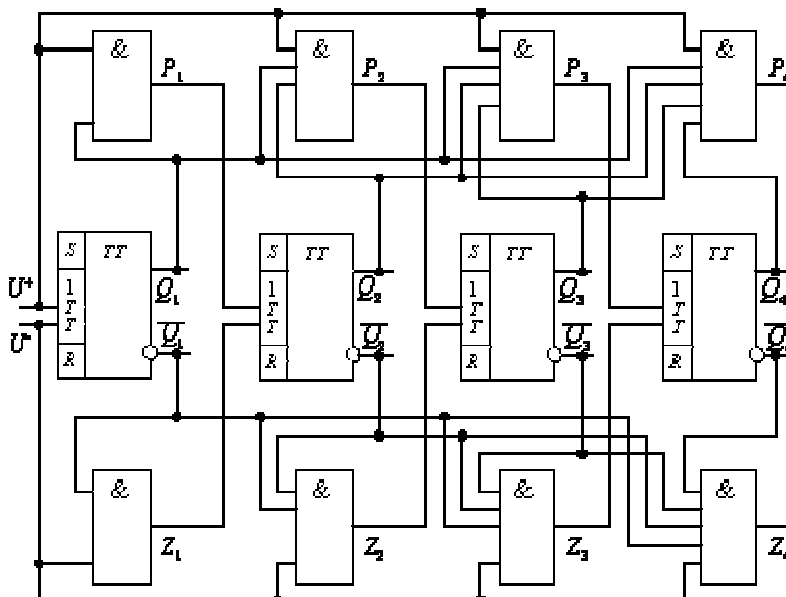


Рис.5.8. Схема двоканального реверсивного лічильника

Віднімальні імпульси  $U_-$  надходять на лічильний вхід першого розряду лічильника і одночасно на входи всіх вентилів у ланцюзі паралельних позик. При цьому формуються імпульси мікрозрядних позик на основі таких логічних виразів:

$$Z_1 = U_- \overline{Q_1}; Z_2 = U_- \overline{Q_1} Q_2; Z_3 = U_- \overline{Q_1} \overline{Q_2} Q_3; Z_4 = U_- \overline{Q_1} \overline{Q_2} \overline{Q_3} Q_4.$$

Таким чином, у двоканальних реверсивних лічильниках напрямок лічби безпосередньо задається підсумовуючими  $U_+$  або віднімальними  $U_-$  імпульсами. Забороняється одночасне надходження на входи двоканального реверсивного лічильника підсумовуючих і віднімаючих імпульсів.

На практиці з урахуванням схемотехнічних можливостей мікросхем середнього ступеня інтеграції багаторозрядні реверсивні лічильники будують у вигляді групової структури. При цьому кожна група представляється, наприклад, мікросхемою чотирирозрядного реверсивного лічильника з паралельними перенесеннями і позиками. Між групами можуть бути утворені послідовні або паралельні зв'язки.

#### 5.4. Двійково-десяткові лічильники

Двійково-десяткові лічильники реалізують лічбу імпульсів у десятичній системі числення, причому кожна десяткова цифра від нуля до дев'яти кодується чотирирозрядним двійковим кодом (тетрадою). Ці лічильники часто називають десятичними або декадними, оскільки вони працюють з модулем лічби, кратним десяти (10, 100, 1000 і т.д.).

Багаторозрядний двійково-десятковий лічильник будується на основі регулярного ланцюга декад, при цьому перша (молодша) декада має вагу 100, друга – 101, третя – 102 і т.д.

Декада будується на основі чотирирозрядного двійкового лічильника, в якому вилучається надлишкове число станів. Вилучення зайвих шести станів у декаді досягається багатьма способами:

попереднім записуванням числа 6 (двійковий код 0110);

після лічби дев'ятого імпульсу вихідний код дорівнює 1111 і десятиковий сигнал повертає лічильник у початковий стан 0110, отже, тут результат лічби фіксується двійковим кодом з надлишком 6;

блокуванням переносів: лічба імпульсів до дев'яти здійснюється у двійковому коді, після чого вмикаються логічні зв'язки блокування перенесень; з надходженням десятого імпульсу лічильник закінчує цикл роботи і повертається в початковий нульовий стан;

введенням обернених зв'язків, які забезпечують лічбу в двійковому коді й примусовим перемиканням лічильника в нульовий початковий стан після надходження десятого імпульсу.

Схема синхронного десятикового лічильника з блокуванням перенесень показана на рис.5.9. У цій схемі С-входи використовуються як лічильні. З надходженням десятого імпульсу на С-вхід молодшого розряду JK-тригера обнуляються перший і четвертий розряди і сигналом з виходу Q4 блокують перемикання другого і третього розряду.

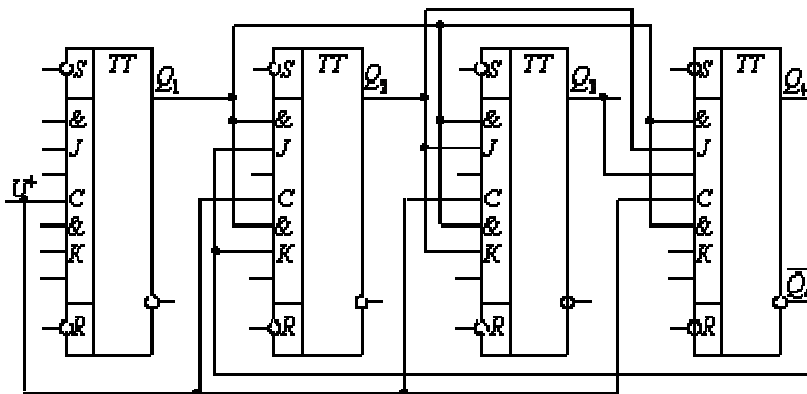


Рис.5.9. Схема десятикового лічильника на JK-тригерах

Схема п'ятирозрядного підсумовуючого двійково-десятого лічильника показана на рис.5.10.

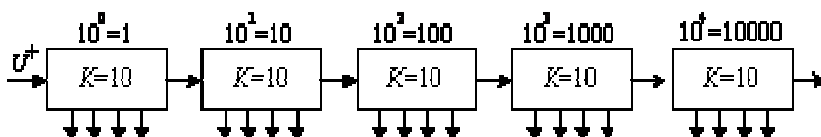


Рис.5.10. Схема п'ятирозрядного підсумовуючого двійково-десятого лічильника

Модуль даного лічильника складає  $K_{лч} = 105 = 100000$ , ємність лічби  $N_{лч} = K_{лч} - 1 = 99999$ .

Виходи тригерів кожної декади підключаються до входів дешифраторів, які забезпечують візуальну індикацію стану лічильника за допомогою різного роду світлових табло.

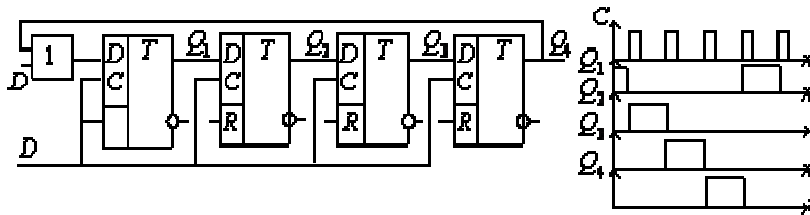
#### 5.5. Лічильники з одиничним кодуванням

При одиничному (унітарному) кодуванні стани n-розрядного лічильника розрізняються лише місцеположенням однієї одиниці, яка називається маркующим кодом; в інших розрядах записані нулі. В окремих випадках маркующий код складається з двох одиниць і називається парно-одиничним.

Лічильник з одиничним кодуванням – це ланцюг тригерів, в якому забезпечується зсув попередньо записаного маркующего коду по "кільцю" в напрямку старших розрядів (прямий підрахунок) або молодших (обернений підрахунок). Такі лічильники часто називають кільцевими (за аналогією з кільцевими регістрами зсуву).

Лічильник з одиничним кодуванням характеризується модулем КСЧ = n і ємністю лічби  $N_{лч} = n - 1$ . Таким чином, число станів кільцевого лічильника дорівнює його розрядності і є значно меншим порівняно з іншими типами лічильників. В кільцевих лічильниках кожний розряд має вагу, яка дорівнює номеру стану 0,1,2,...,(n-1). Із стану (n-1) після надходження чергового імпульсу лічильник утворює на виході сигнал закінчення циклу (переповнення) і повертається в початковий стан за допомогою кола оберненого зв'язку з виходу старшого розряду  $Q_n$  на вхід молодшого розряду  $Q_1$ . Схема чотирирозрядного кільцевого лічильника показана на рис.5.11, а. Перед початком роботи по входу D схеми ЧИ в молодший розряд лічильника записується одиниця і встановлюється початковий код  $Q_4 Q_3 Q_2 Q_1 = 0001$ . З надходженням кожного лічильного імпульсу по входу син-

хронізації одиничний код послідовно зсувається в бік старших розрядів; при цьому молодші розряди, виконані на D-тригерах з динамічним керуванням, обнуляються. Після надходження четвертого імпульсу лічильник повертається в початковий стан за допомогою сигналу з виходу Q4 на вхід схеми ЧИ.



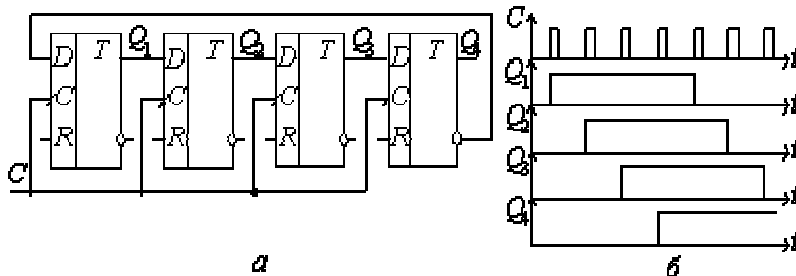
а б  
Рис.5.11. Кільцевий лічильник: а – схема; б – часові діаграми

Практичне використання кільцевих лічильників пояснюється такими його перевагами:

не потребує вихідного дешифратора, оскільки всі стани відрізняються наявністю одиниці лише в одному якому-небудь тригері; в процесі лічби завжди переключається в одиничний стан лише один тригер, що забезпечує мінімальне значення  $t_{вст}$ ; спрощується побудова схеми контролю лічильника.

Схема лічильника Джонсона будується на основі кільцевого, в якому обернений зв'язок реалізується підключенням інверсного виходу старшого розряду до входу молодшого.

Лічильник Джонсона (рис.5.12) характеризується модулем лічби  $K_{ЛЧ}=2n$  і ємністю лічби  $N_{max}=2n-1$ . Таким чином, число станів лічильника Джонсона в два рази більше аналогічного параметра кільцевого лічильника. Проте інформація на виходах лічильника Джонсона представляється не в двійковій позиційній системі числення, яка потребує додаткового перетворення.



а б  
Рис.5.12. Лічильник Джонсона: а – схема; б – часові діаграми

Як видно із часових діаграм (рис.5.12, б), в процесі лічби спочатку рухається "хвиля" одиниць, а потім – "хвиля" нулів. Дешифрація станів лічильника Джонсона здійснюється простіше порівняно з двійковими позиційними лічильниками.

## Дешифратори і шифратори

### 6.1. Загальна характеристика дешифраторів

Дешифратором називається функціональний вузол комп'ютера, призначений для перетворення кожної комбінації вхідного двійкового коду в керуючий сигнал лише на одному із своїх виходів. У загальному випадку дешифратор має  $n$  однофазних входів

(іноді  $2n$  парафазних) і  $m=2^n$  виходів, де  $n$  – розрядність (довжина) коду, який дешифрується. Дешифратор з максимально можливим числом виходів  $m=2n$  називається повним. Функціонування повного дешифратора описується системою логічних виразів вигляду:

$$F_0 = \overline{X_n} \overline{X_{n-1}} \dots \overline{X_2} \overline{X_1};$$

$$F_1 = \overline{X_n} \overline{X_{n-1}} \dots \overline{X_2} X_1;$$

$$\dots \dots \dots$$

$$F_{m-1} = X_n X_{n-1} \dots X_2 X_1,$$

де  $X_1, \dots, X_n$  – вхідні двійкові змінні;  $F_0, F_1, \dots, F_{m-1}$  – вихідні логічні функції, що являють собою мінтерми (конституенти 1)  $n$  змінних.

Індекс функції  $F_i$  визначає номер обраного виходу і відповідає десятковому еквіваленту вхідного коду. Вихід, на якому з'являється керуючий сигнал, називається активним. Якщо значення сигналу на активному виході відображається лог.1, то на решті пасивних виходів встановлюється лог.0. Двійковий код, який вміщує завжди тільки одну одиницю, а інші – нулі, називається унітарним. Тому дешифратор є перетворювачем вхідного позиційного коду в унітарний вихідний код.

У дешифраторах в інтегральному виконанні стан активного виходу часто відображається значенням лог.0, а на інших пасивних виходах устатковується лог.1. Функціонування повного дешифратора з інверсними виходами представляється системою виду:

$$L_0 = X_n \vee X_{n-1} \vee \dots \vee X_2 \vee X_1;$$

$$L_1 = X_n \vee X_{n-1} \vee \dots \vee X_2 \vee \overline{X_1};$$

$$\dots \dots \dots$$

$$L_{m-1} = \overline{X_n} \vee \overline{X_{n-1}} \vee \dots \vee \overline{X_2} \vee \overline{X_1}.$$

де  $L_0, L_1, \dots, L_{m-1}$  – вихідні логічні функції, що є макстермами (конституенти 0)  $n$  змінних.

Індекс функції  $L_i$  визначає номер вибраного виходу і відповідає десятковому еквіваленту вхідного коду. Між двома видами

$$F_i = \overline{L_i} \cdot I$$

вихідних функцій існує простий зв'язок:

Дешифратори класифікують за такими ознаками:

способом структурної організації – одноступеневі (лінійні) і багатоступеневі, в тому числі пірамідальні та прямокутні (матричні);

форматом вхідного коду – двійкові, двійково-десятькові;

розрядністю коду, який дешифрується – 2, 3, ..., n;

формою подачі вхідного коду – з однофазними і парафазними входами;

кількістю виходів – повні й неповні дешифратори;

видом вхідних стробуючих сигналів – в прямому або інверсному значеннях;

типом використовуваних логічних елементів – І, НЕ, ЧИ, НЕ І, НЕ ЧИ і т.д.

До основних характеристик дешифратора відносять: число ступенів (каскадів) дешифрації, кількість використаних логічних елементів або мікросхем, загальне число входів логічних елементів, час дешифрації і споживану потужність.

Умовні графічні позначення дешифраторів на електричних схемах показані на рис.6.1.

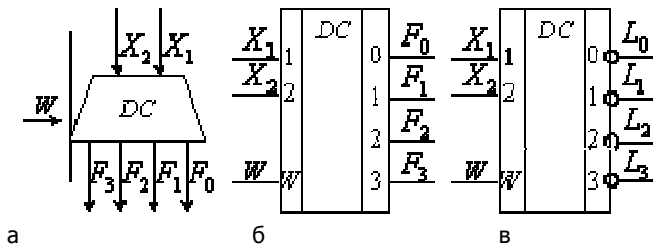


Рис. 6.1. Умовні графічні позначення дешифратора: а – на функціональних схемах; б, в – на принципіальних схемах

Логічна функція дешифратора позначається буквами DC (de-coder). Мітки лівого додаткового поля в умовному позначенні відображають десяткові ваги вхідних змінних, а мітки правого додаткового поля відповідають десятковим еквівалентам вхідних комбінацій двійкових змінних. У схему дешифраторів вбудовуються один або два стробуючих (дозволяючих) входи, наприклад, W (рис.6.1, б). За допомогою сигналу на вході W визначається момент спрацювання дешифратора; крім того, вхід W використовується для нарощування розрядності вхідного коду. На практиці повний дешифратор на n входів і m виходів для стислості називають дешифратором "з n в m" або "n ® m". Наприклад, дешифратор "з 3 у 8" – активізується одна з восьми вихідних ліній. В комп'ютерах дешифратори використовують для виконання таких операцій:

дешифрації коду операції, записаного в регістр команд процесора, що забезпечує вибір потрібної мікропрограми;

перетворення коду адреси операнда в команди в керуючі сигнали вибору заданої комірки пам'яті в процесі записування або читання інформації;

забезпечення візуалізації на зовнішніх пристроях;

реалізації логічних операцій та побудови мультиплексорів і демультиплексорів.

Використання дешифраторів для дешифрації коду операції і адреси операнда, розташованих в регістрі команд процесора, показано на рис.6.2. Дешифрація коду операції в пристрої керування (ПК) визначає тип машинної команди. Дешифрація адреси операнда в оперативній пам'яті (ОП) забезпечує доступ до вказаної комірки пам'яті для записування або зчитування даних.

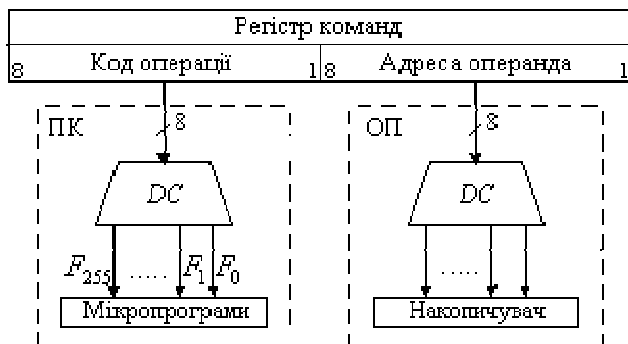


Рис. 6.2. Ілюстрація використання дешифраторів

## 6.2. Лінійні дешифратори на два входи і чотири виходи

У лінійному дешифраторі "з n в m" кожна вихідна функція  $F_i$  реалізується повністю окремим n-вхідним логічним елементом при використанні парафазного вхідного коду. Логіка роботи повних дешифраторів на два входи  $X_1, X_2$  і чотири прямих виходи  $F_0, F_1, F_2, F_3$  і чотири інверсних виходи  $L_0, L_1, L_2, L_3$  наведена в табл.4.1 і 4.2 відповідно.

Таблиця 6.1						Таблиця 6.2							
X2	X1	F0	F1	F2	F3	X2							
0	0	1	0	0	0	0	0	0	1	1	1		
0	1	0	1	0	0	0	1	1	0	1	1		
1	0	0	0	1	0	1	0	1	1	0	1		
1	1	0	0	0	1	1	1	1	1	1	0		

За даними табл.6.1 отримують систему логічних функцій в ДДНФ:

$$F_0 = \overline{X_2} \overline{X_1}; F_1 = \overline{X_2} X_1; F_2 = X_2 \overline{X_1}; F_3 = X_2 X_1. \quad (6.1)$$

Для лінійного дешифратора зі стробуючим входом W система рівнянь (4.1) набуває вигля-

$$\text{ду: } F_0 = \overline{X_2} \overline{X_1} W, F_1 = \overline{X_2} X_1 W, F_2 = X_2 \overline{X_1} W, F_3 = X_2 X_1 W. \quad (6.2)$$

Схеми лінійних дешифраторів на основі рівнянь (6.1) и (6.2) показані на рис.6.3.

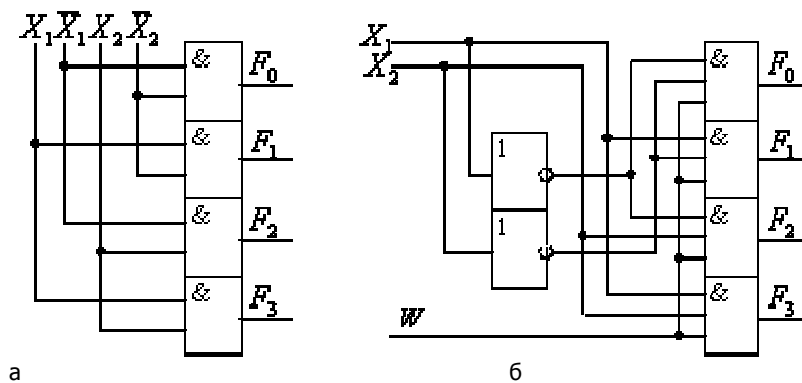


Рис. 6.3. Схеми лінійних дешифраторів на елементах І: а – з парафазними входами; б – з однофазними входами і стробуванням. У схемі, зображеній на рис. 6.3, б використовується однофазний вхідний код, оскільки інверсії змінних утворюються елементами НЕ. Якщо сигнал на стробуючому вході  $W=0$ , то робота дешифратора блокується – на всіх виходах установлюються логічні нулі незалежно від значень вхідних змінних. При  $W=1$  дешифратор функціонує згідно з табл. 6.1.

За даними табл. 6.2 записується система логічних функцій в ДКНФ:

$$L_0 = X_2 \vee X_1; L_1 = X_2 \vee \overline{X_1}; L_2 = \overline{X_2} \vee X_1; L_3 = \overline{X_2} \vee \overline{X_1}. \quad (6.3)$$

Схема лінійного дешифратора з парафазним вхідним кодом та інверсними виходами, побудована згідно з рівнянням (6.3) на елементах ЧИ, показана на рис. 6.4, а.

Для лінійного дешифратора із стробуючим  $W$  входом система керування (6.3) набуває вигляду:

$$\begin{aligned} L_0 &= X_2 \vee X_1 \vee W; L_1 = X_2 \vee \overline{X_1} \vee W; \\ L_2 &= \overline{X_2} \vee X_1 \vee W; L_3 = \overline{X_2} \vee \overline{X_1} \vee W. \end{aligned} \quad (6.4)$$

Схема лінійного дешифратора на основі рівнянь (6.4) показана на рис.6.4, б.

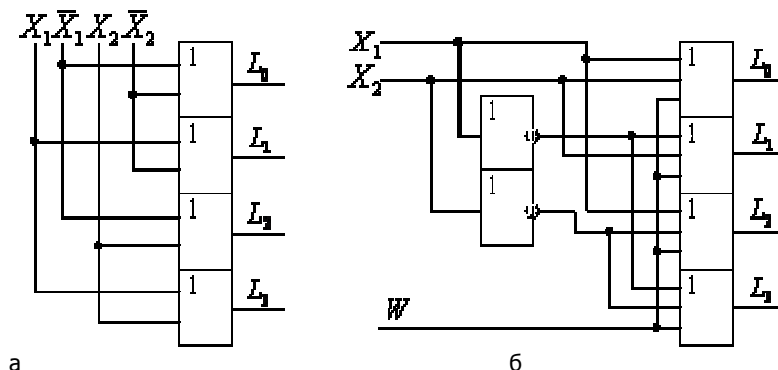


Рис. 6.4. Схема лінійних дешифраторів на елементах ЧИ: а – з парафазними входами; б – з однофазними входами і стробуванням.

Тут використовується однофазний вхідний код, оскільки інверсії змінних утворюються елементами НЕ. Якщо сигнал на стробуючому вході  $W=1$ , то робота дешифратора блокується – на всіх виходах встановлюються лог. 1 незалежно від значень вхідних змінних. При  $W=0$  дешифратор функціонує згідно з табл.6.2.

### 6.3. Пірамідальні дешифратори

У пірамідальному дешифраторі число ступенів на одиницю менше розрядності вхідного коду, тобто  $K=n-1$ . В усіх ступенях використовуються тільки двовходові логічні елементи. На першому ступені використовуються лінійні дешифратори на два входи і чотири виходи. Число логічних елементів у кожному ступені дорівнює  $M_i=2i+1$ , де  $i=1, 2, \dots, k$ . Це означає, що кожен подальший ступінь має в два рази більше елементів, ніж попередній. Вихід елемента  $i$ -го ступеня підключається до входів тільки двох елементів  $(i+1)$ -го ступеня.

Пірамідальна структура для реалізації повного дешифратора "з 3 в 8" описується системою мінтермів в виду:

$$F_0 = \overline{X_3} \overline{X_2} \overline{X_1}; F_1 = \overline{X_3} \overline{X_2} X_1; \dots F_7 = X_3 X_2 X_1.$$

Схема пірамідального дешифратора з парафазним вхідним кодом на три входи і вісім виходів показана на рис.6.5.

На першому ступені дешифруються змінні  $X_2$  і  $X_1$ , на другому ступені добавляється розряд  $X_3$ . При більшому числі розрядів дешифрованого коду, наприклад,  $n>10$ , дешифратор в  $n/4$  економічніше лінійного.

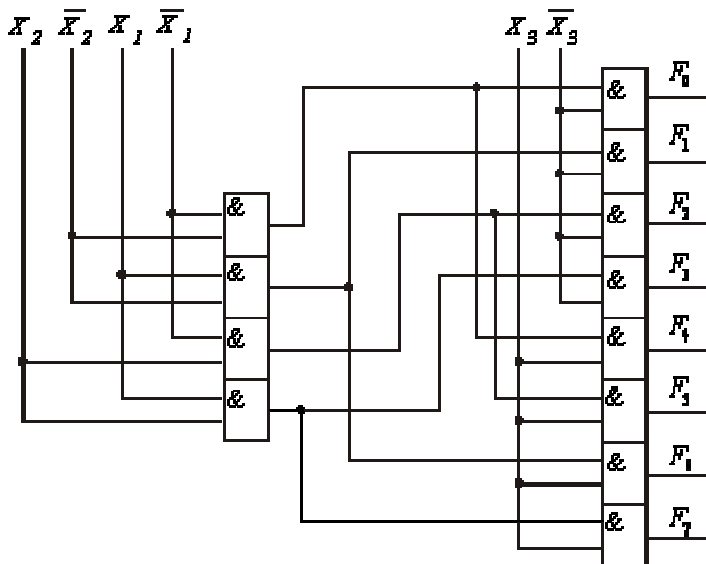


Рис. 6.5. Схема пірамідального дешифратора на три входи і вісім виходів

Основним недоліком пірамідального дешифратора є велике число ступенів, що суттєво збільшує час дешифрації коду.

#### 6.4. Прямокутні дешифратори

Прямокутний дешифратор будується за двоступеневою схемою. При цьому вхідний код розбивається на дві групи по  $n/2$  розрядів при парному  $n$ ; при непарній розрядності групи вміщують нерівне число змінних. Дві групи змінних декодуються на першому ступені двома повними лінійними (можливо і пірамідальними) дешифраторами, а на другому ступені формуються вихідні функції.

Умовно вважають, що один з дешифраторів першого ступеня формує адреси рядків матриці, а другий – адреси стовпчиків матриці. На перетині ліній рядків і стовпчиків підключається  $m=2n$  двовходових схем збігу, які утворюють другий, вихідний ступінь дешифратора. При парному  $n$  матриця вентилів квадратна, при непарному  $n$  – прямокутна. Тому такі дешифратори називаються матричними або прямокутними.

Запишемо систему вихідних функцій повного дешифратора "з 4 в 16" у вигляді таких скорочених значень:

$$\begin{aligned} F_0 &= a_0 b_0; & F_4 &= a_1 b_0; & F_8 &= a_2 b_0; & F_{12} &= a_3 b_0; \\ F_1 &= a_0 b_1; & F_5 &= a_1 b_1; & F_9 &= a_2 b_1; & F_{13} &= a_3 b_1; \\ F_2 &= a_0 b_2; & F_6 &= a_1 b_2; & F_{10} &= a_2 b_2; & F_{14} &= a_3 b_2; \\ F_3 &= a_0 b_3; & F_7 &= a_1 b_3; & F_{11} &= a_2 b_3; & F_{15} &= a_3 b_3. \end{aligned} \quad (6.5)$$

де введені дворозрядні функції  $a_i$  і  $b_i$ , які реалізуються дешифраторами рядків і стовпчиків відповідно:

$$\begin{aligned} b_0 &= X_2 X_1; & b_1 &= X_2 \bar{X}_1; & b_2 &= \bar{X}_2 X_1; & b_3 &= \bar{X}_2 \bar{X}_1; \\ a_0 &= \bar{X}_4 \bar{X}_3; & a_1 &= \bar{X}_4 X_3; & a_2 &= X_4 \bar{X}_3; & a_3 &= X_4 X_3. \end{aligned} \quad (6.6)$$

Схема прямокутного дешифратора на основі рівнянь (6.5) і (6.6) показана на рис.6.6.

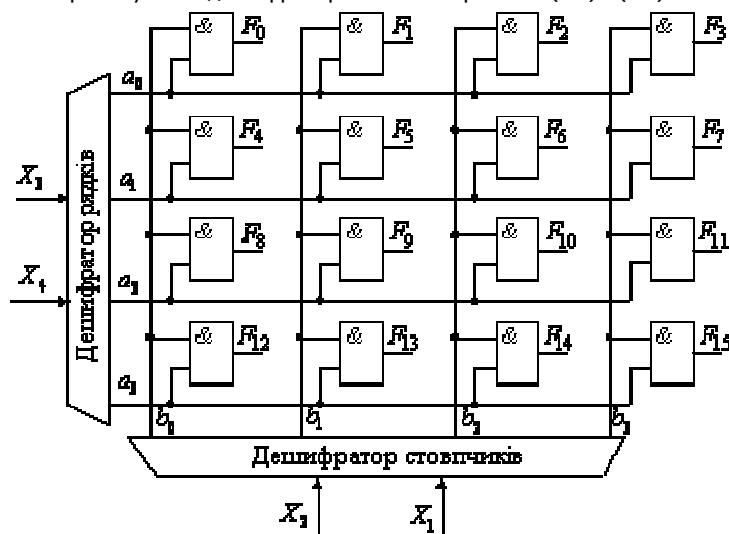


Рис. 6.6. Схема прямокутного дешифратора

При великому числі розрядів прямокутний дешифратор майже у  $n/2$  рази економічніший лінійного і у два рази – пірамідального.

#### 6.5. Багатоступеневі дешифратори.

Каскадування дешифраторів



0	1	1	1	1	1	1	1	0	1	1	1	0	1
0	1	1	1	1	1	1	0	×	1	1	0	0	1
0	1	1	1	1	1	0	×	×	1	0	1	0	1
0	1	1	1	1	0	×	×	×	1	0	0	0	1
0	1	1	1	0	×	×	×	×	0	1	1	0	1
0	1	1	0	×	×	×	×	×	0	1	0	0	1
0	1	0	×	×	×	×	×	×	0	0	1	0	1
0	0	×	×	×	×	×	×	×	0	0	0	0	1

У табл. 7.1 значення вхідних змінних праворуч від діагоналі, утвореної цифрами 1, не повинні визначати вихідний код (вони позначені хрестиком). Це пояснюється тим, що сигнал з більшим пріоритетом блокує запити з меншими пріоритетами.

Із табл. 7.1 отримуємо вирази для вихідного коду шифратора  $\overline{X_3}, \overline{X_2}, \overline{X_1}$  і функцій  $\overline{V}$  та  $\overline{P}$ , які відповідно визначають відсутність інформаційних сигналів на всіх виходах та наявність сигналу хоч би на одному вході. Для спрощення виразів використовуємо тотожність  $F_i \vee \overline{F_i} F_k = F_i \vee F_k$  та закони де Моргана:

$$\overline{X_3} = \overline{W} \vee WY_1;$$

$$X_2 = \overline{W} \vee WY_1 F_3 F_2 \vee W F_7 F_6 F_5 \vee W F_7 F_6 F_4;$$

$$X_1 = \overline{W} \vee WY_1 F_3 F_2 \vee WY_1 F_3 F_1 \vee W F_7 F_6 \vee W F_7 F_5 F_4;$$

$$\overline{P} = \overline{W} \vee WY_1 Y_2; \quad \overline{V} = \overline{W} \vee Y_1 Y_2 \overline{W} \vee Y_1 \vee Y_2;$$

$$Y_1 = F_7 F_6 F_3 F_4; \quad Y_2 = F_3 F_2 F_1 F_0.$$

На основі цих виразів побудована (рис. 7.2) схема пріоритетного шифратора "8 → 3".

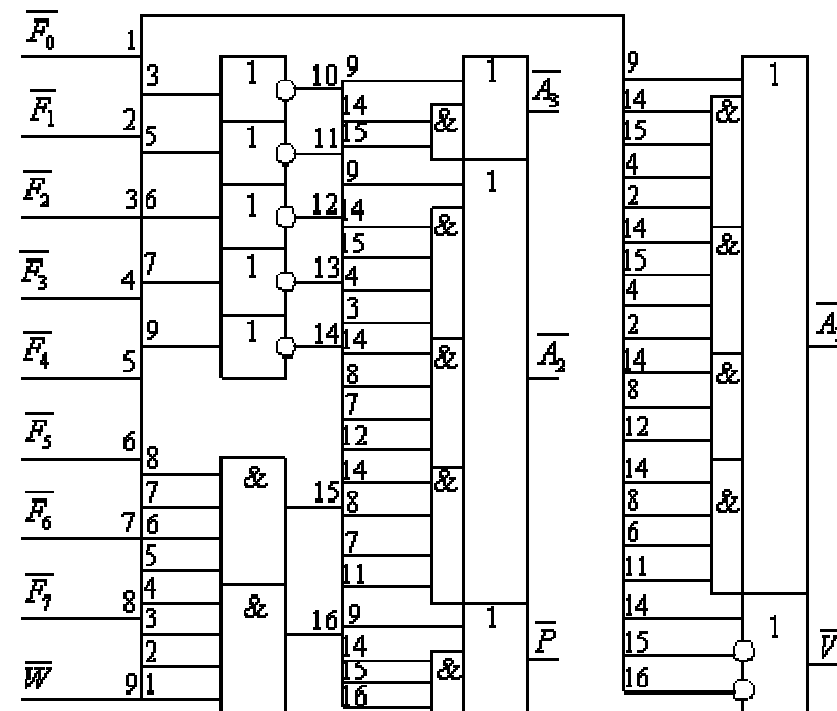


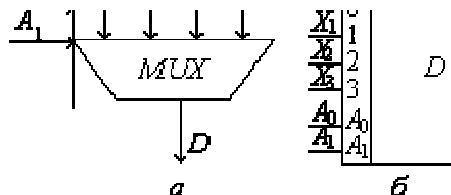
Рис. 7.2. Схема пріоритетного шифратора "8 → 3"

При  $\overline{W} = 1$  робота схеми блокується і незалежно від сигналів на входах маємо на інверсних виходах:  $\overline{X_3} \overline{X_2} \overline{X_1} = 111$ ,  $\overline{P} = 1$ ,  $\overline{V} = 1$ . Якщо, наприклад,  $\overline{F_6} = 0$  і  $\overline{F_2} = 0$ , то схема формує на виходах код номера входу із старшим пріоритетом:  $\overline{X_3} \overline{X_2} \overline{X_1} = 001$  або в прямому коді  $X_3 X_2 X_1 = 1102 = 610$ . Активний стан виходу відображається значеннями функцій  $\overline{P} = 0$  і  $\overline{V} = 1$ , які передаються в процесор, а також використовуються при каскадуванні шифраторів. Схема, зображена на рис. 7.2, є аналогом шифратора K555IB1.

### 7.3. Каскадування шифраторів



рис. 8.1. Умовне позначення мультимплектора: а – на функціональних схемах; б – на принципових схемах



Каскадування шифраторів використовується для збільшення розрядності вхідного слова. Схема каскадування двох восьмивходових шифраторів К555ІВ1 для пріоритетного обслуговування 16-розрядного слова  $\overline{F}_{15} - \overline{F}_0$  показана на рис. 7.3.

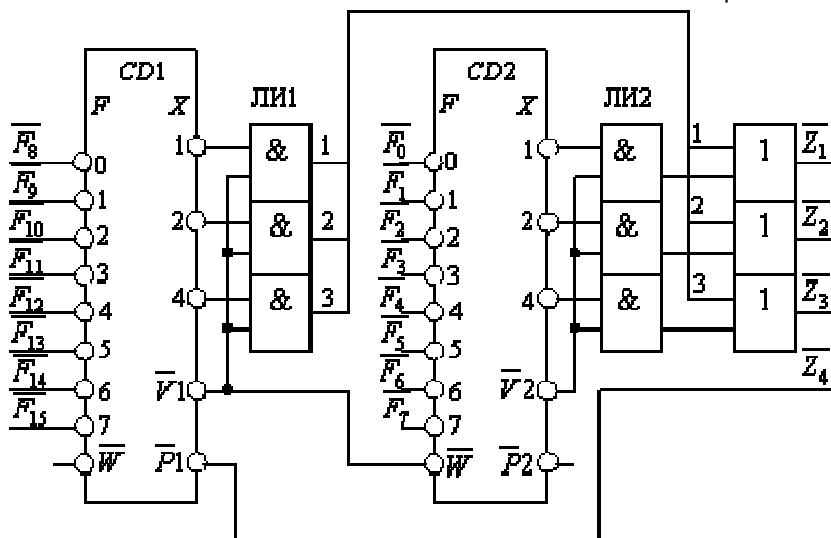


Рис. 7.3. Каскадування двох восьмивходових пріоритетних шифраторів

Розряди  $\overline{F}_{15} - \overline{F}_8$  старшого байта вхідного слова поступають на перший шифратор CD1, а розряди  $\overline{F}_7 - \overline{F}_0$  молодшого байта подаються на другий шифратор CD2. Вхід  $\overline{F}_{15}$  має найвищий пріоритет, а  $\overline{F}_0$  – найнижчий. Інформаційні виходи обох шифраторів об'єднуються за допомогою логічних елементів ЧИ, утворюючи трирозрядний інверсний код  $\overline{Z}_3\overline{Z}_2\overline{Z}_1$ . Значення старшого розряду  $\overline{Z}_4$  забезпечується безпосередньо сигналом  $\overline{F}_1$ .

Інформація з виходів першого шифратора подається на входи елементів ЧИ за допомогою схем збігу ЛІН1 при  $\overline{V}_1 = 1$  (мікросхема CD1 сприймає вхідні дані, а CD2 – блокувана). Інформація з виходів другого шифратора подається на входи елементів ЧИ за допомогою схем збігу ЛІН2 при  $\overline{V}_2 = 1$  (мікросхема CD2 сприймає вхідні дані, CD1 – блокується).

Схема працює так: коли на вході є активний сигнал із старшого байта вхідного слова, наприклад,  $\overline{F}_{14} = 0$  ( $\overline{P}_1 = 0$ ,  $\overline{V}_1 = 1$ ), то працює шифратор CD1 і на виходах елементів ЧИ формується інверсний код  $\overline{Z}_4\overline{Z}_3\overline{Z}_2\overline{Z}_1 = 0001$ , що відповідає прямому значенню  $Z_4Z_3Z_2Z_1 = 1110 = 14_{10}$ . Якщо активний вхідний сигнал відноситься до молодшого байта слова, наприклад,  $\overline{F}_6 = 0$ , то працює шифратор CD2 ( $\overline{P}_1 = 1$ ,  $\overline{V}_1 = 0$ ,  $\overline{V}_2 = 1$ ) і на інверсних виходах формується код  $\overline{Z}_4\overline{Z}_3\overline{Z}_2\overline{Z}_1 = 1001$ , що відповідає прямому числу  $Z_4Z_3Z_2Z_1 = 0110 = 6_{10}$ .

## Мультимплектори і демультимплектори

### 8.1. Загальна характеристика мультимплекторів

Мультимплексором називається функціональний вузол комп'ютера, призначений для почергової комутації (перемикання) інформації від одного з  $n$  входів на загальний вихід. Номер конкретної вхідної лінії, що підключається до виходу в кожний такт машинного часу, визначається адресним кодом  $A_0, A_1, \dots, A_{m-1}$ . Зв'язок між числом інформаційних  $n$  і адресних  $m$  входів визначається співвідношенням  $n=2^m$ . Таким чином, мультимплексор реалізує керовану передачу даних від кількох вхідних ліній в одну вихідну.

Умовне графічне позначення мультимплекторів показано на рис. 8.1. Функція мультимплекторів записується буквами MUX (multiplexor).

Мультимплектори застосовують для таких операцій: комутації як окремих ліній, так і груп ліній (шин); перетворення паралельного коду в послідовний; реалізації логічних функцій; побудови схем порівняння, генераторів кодів.

Мультимплексор символічно часто позначають: "n-1".

Логіка роботи чотиривходового мультимплексора наведена в табл. 4.4, де  $A_0, A_1$  – адресний код;  $F_0, F_1, F_2, F_3$  – виходи внутрішнього дешифратора;  $X_0, X_1, X_2, X_3$  – вхідна інформація;  $D$  – загальний інформаційний вихід.

Таблиця 8.1

A1	A0	F0	F1	F2	F3	D
0	0	1	0	0	0	$F_0X_0$

0	1	0	1	0	0	F1X1
1	0	0	0	1	0	F2X2
1	1	0	0	0	1	F3X3

На основі табл. 8.1. вираз для вихідної функції D можна представити з використанням виходів F0–F3 внутрішнього дешифратора у вигляді:

$$D = F0X0 \vee F1X1 \vee F2X2 \vee F3X3, \quad (8.1)$$

або з мінтермами адресного коду:

$$D = \overline{A_1} \overline{A_0} X0 \vee \overline{A_1} A_0 X1 \vee A_1 \overline{A_0} X2 \vee A_1 A_0 X3. \quad (8.2)$$

Схеми мультиплексорів, відповідні рівнянням (8.1) і (8.2), показані на рис. 8.2.

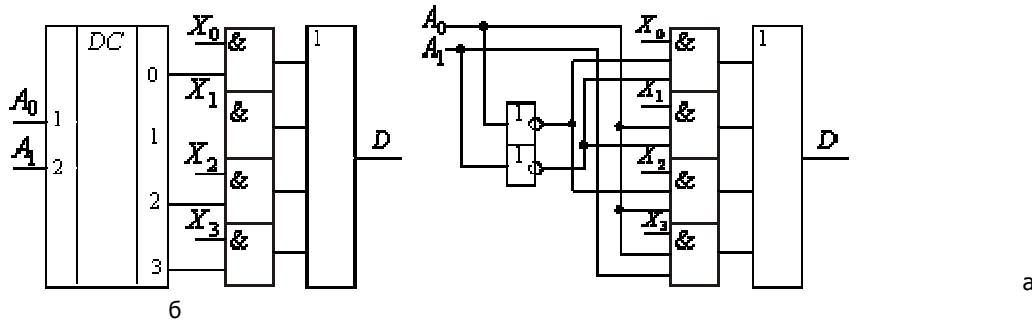


Рис. 8.2. Схеми мультиплексорів: а – з внутрішнім дешифратором; б – з адресними мінтермами

При побудові схеми мультиплексорів з внутрішнім лінійним дешифратором потрібні логічні елементи з меншим числом входів, проте при цьому збільшується час встановлення вихідного сигналу. При виключенні дешифратора швидкодія підвищується, однак потрібні схеми збігу з більшим числом входів.

## 8.2. Каскадування мультиплексорів

В інтегральному виконанні мультиплексори випускають на чотири, вісім або шістнадцять входів. Каскадування дозволяє реалізувати комутацію довільного числа вхідних ліній на базі серійних мікросхем мультиплексорів меншої розрядності.

Приклад побудови схеми мультиплексора на 16 входів на основі типових чотиривходових мультиплексорів показаний на рис. 8.3.

Молодші розряди адреси A1, A0 підключаються до адресних входів усіх мультиплексорів першого рівня, на виходах яких виробляються такі функції:

$$D'_0 = F0X0 \vee F1X1 \vee F2X2 \vee F3X3;$$

$$D'_1 = F0X4 \vee F1X5 \vee F2X6 \vee F3X7;$$

$$D'_2 = F0X8 \vee F1X9 \vee F2X10 \vee F3X11;$$

$$D'_3 = F0X12 \vee F1X13 \vee F2X14 \vee F3X15,$$

де F0 – F3 – виходи внутрішніх дешифраторів: F0 =  $\overline{A_1} \overline{A_0}$ ; F1 =  $\overline{A_1} A_0$ ; F2 =  $A_1 \overline{A_0}$ ; F3 =  $A_1 A_0$ ; X15 – X0 – вхідні змінні.

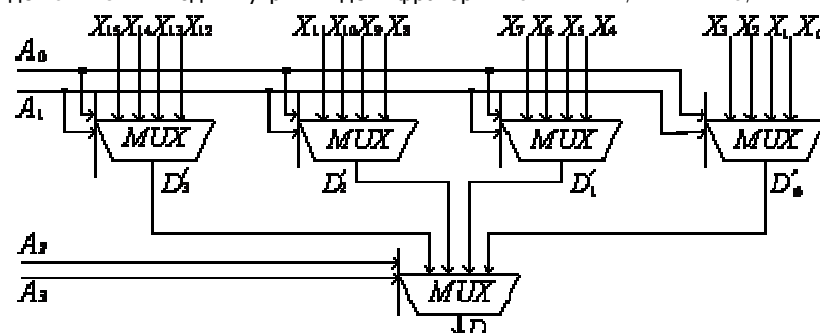


Рис. 8.3. Каскадування мультиплексорів

Старші розряди адреси A3, A2 подаються на адресні входи мультиплексора другого рівня, на виході якого формується остаточна функція

$$D = F'_0 D'_0 \vee F'_1 D'_1 \vee F'_2 D'_2 \vee F'_3 D'_3,$$

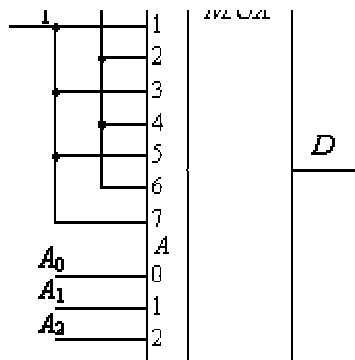
де внутрішні виходи дешифратора визначаються такими мінтермами:

$$F0 = \overline{A_3} \overline{A_2}; F1 = \overline{A_3} A_2; F2 = A_3 \overline{A_2}; F3 = A_3 A_2.$$

Нехай, наприклад, значення адреси A3 A2 A1 A0 = 10112 = 1110. При цьому на вхід другого рівня комутується змінна  $D'_2$  (оскільки  $F'_2 = 1$ ), значення якої визначається з рівнянь першого рівня:

$$D'_2 = F3 X11 = A1 A0 X11 = 1X11 = X11.$$

Рис. 8.4. Схема мультиплексора для реалізації логічної функції трьох змінних за допомогою десятичних еквівалентів мінтермів 1, 3, 5 і 7



### 8.3 Реалізація логічних функцій мультиплексорами

За допомогою мультиплексорів реалізуються логічні функції з числом змінних  $m$ , що дорівнює розрядності адресного коду. Функція, що виконується, має бути представлена в ДДНФ. При цьому змінні поступають на адресні входи, а інформаційні входи використовуються як налаштувальні – на них подаються константи нуля і одиниці залежно від функції, яка реалізується. Вихідна функція триадресного мультиплексора на вісім входів описується рівнянням:

$$D(A) = \bar{A}_2 \bar{A}_1 \bar{A}_0 X_0 \vee \bar{A}_2 \bar{A}_1 A_0 X_1 \vee \bar{A}_2 A_1 \bar{A}_0 X_2 \vee \bar{A}_2 A_1 A_0 X_3 \vee$$

$$\vee A_2 \bar{A}_1 \bar{A}_0 X_4 \vee A_2 \bar{A}_1 A_0 X_5 \vee A_2 A_1 \bar{A}_0 X_6 \vee A_2 A_1 A_0 X_7.$$

Якщо потрібно отримати логічну функцію з десятичними еквівалентами мінтермів 1, 3, 5 і 7, то на парні входи  $X_0, X_2, X_4$  і  $X_6$  необхідно подати константу "0", а на непарні  $X_1, X_3, X_5$  і  $X_7$  – константу "1". У результаті отримаємо (рис. 8.4):

$$D(A) = \bar{A}_2 \bar{A}_1 A_0 \vee \bar{A}_2 A_1 A_0 \vee A_2 \bar{A}_1 A_0 \vee A_2 A_1 A_0.$$

За допомогою додаткових логічних перетворень можна реалізувати логічні функції з числом змінних  $m+1$ , тобто на одиницю більше розрядності адресного коду мультиплексора.

Можливі й інші схемотехнічні застосування мультиплексорів. Мультиплексор може виконувати функції перетворювача  $n$ -розрядного двійкового паралельного коду в послідовний.

Для цього потрібно подати паралельний код на інформаційні входи мультиплексора і потім змінювати код на адреси в необхідній послідовності.

### 8.4 Мультиплексування шин

Мультиплексування шин – це по чергове перемикання шин (груп ліній) від кількох джерел інформації до одного приймача. Такі мікрооперації реалізуються схемами на основі мультиплексорів одиночних ліній. При виборі кількості й типу мультиплексорів враховують:

число комутуваних шин дорівнює  $2m$ , де  $m$  – довжина адресного коду;

$i$ -й номер входу всіх мультиплексорів служить для підключення розрядів певної однієї шини.

Схема мультиплексора чотирьох  $X(n), Y(n), Z(n)$  і  $S(n)$  шин показана на рис. 8.5. Для її побудови потрібно  $n$  двоадресних чотиривходових мультиплексорів, де  $n$  – довільна розрядність шин, що комутуються.

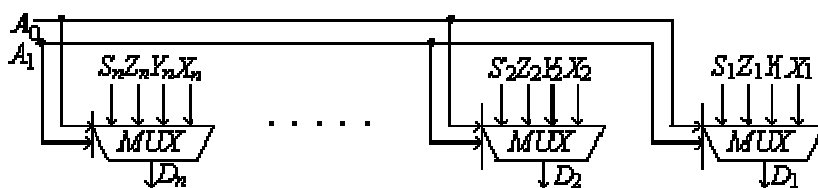


Рис. 8.5. Мультиплексор шин

### 8.5. Загальна характеристика демультимплексорів

Демультимплексором називається функціональний вузол комп'ютера, призначений для комутації (перемикання) сигналу з одного інформаційного входу  $D$  на один з  $n$  інформаційних виходів. Номер виходу, на який в кожний такт машинного часу передається значення вхідного сигналу, визначається адресним кодом  $A_0, A_1, A_2, \dots, A_{m-1}$ . Адресні входи  $m$  та інформаційні входи  $n$  пов'язані співвідношенням  $n=2m$  або  $m = \log_2 n$ .

Демультимплексор виконує функцію, обернену функції мультиплексора. Стосовно мультиплексорів і демультимплексорів користуються також терміном "селектори" даних.

В умовних графічних позначеннях (рис. 8.6) функція демультимплексора позначається буквами DMX.

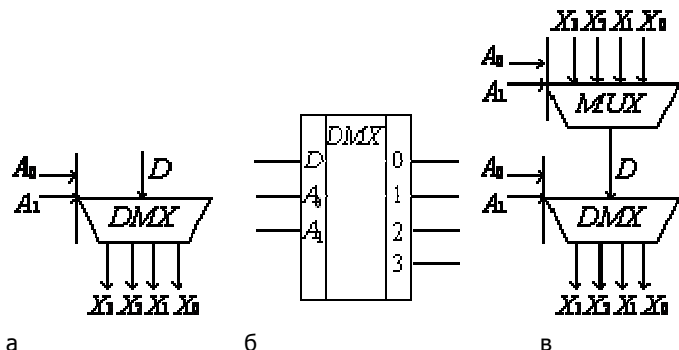


Рис. 8.6. Умовні графічні позначення демультимплексорів: а – на функціональних схемах; б – на принципових схемах; в – типове з'єднання з мультиплексором

Демультимплексори використовують для таких операцій: комутації як окремих ліній, так і багаторозрядних шин; перетворення послідовного коду в паралельний; реалізації логічних функцій та інших.

Демультимплексори часто позначають: " $1 \rightarrow n$ ".

Логіка роботи двоадресного демультимплексора на мові мікрооперацій наведена в табл. 8.2, де  $D$  – інформаційний вхід;  $F_0, F_1$ ,

F2 і F3 – виходи внутрішнього дешифратора адреси.  
Таблиця 8.2

A1	A0	F0	F1	F2	F3	X0	X1	X2	X2
0	0	1	0	0	0	F0D	–	–	–
0	1	0	1	0	0	–	F1D	–	–
1	0	0	0	1	0	–	–	F2D	–
1	1	0	0	0	1	–	–	–	F3D

За даними табл. 4.5 запишемо систему рівнянь для інформаційних виходів:

(8.3)

$$X0 = F0D = \bar{A}_1 \bar{A}_0 D; X1 = F1D = \bar{A}_1 A0D;$$

$$X2 = F2D = A1 \bar{A}_0 D; X3 = F3D = A1 A0D;$$

На основі рівнянь (8.3) побудовані схеми демультиплексорів із внутрішнім дешифратором (рис. 8.7, а) і з поєднанням адресних і вхідних змінних на тривходових елементах І (рис. 8.7, б).

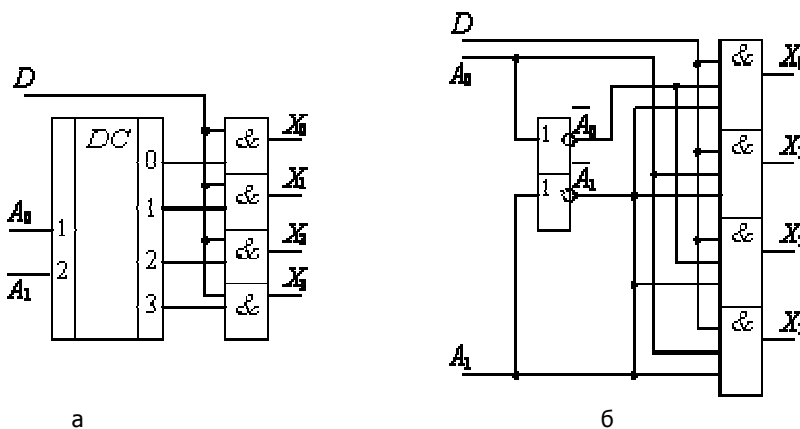


Рис. 8.7. Схема демультиплексорів: а – з внутрішнім дешифратором; б – з поєднанням адресних і вхідних змінних

Схема демультиплексора з поєднанням адресних і вхідних змінних забезпечує високу швидкодію, проте вимагає застосування логічних елементів з більшим числом входів.

#### 8.6. Каскадування демультиплексорів

Каскадування дозволяє реалізувати комутацію одного вхідного сигналу на довільне число вихідних ліній на базі серійних мікросхем меншої розрядності. Нехай потрібно реалізувати демультиплексування вхідного сигналу на  $n$  вихідних ліній, що визначаються  $m$ -розрядним адресним кодом, на базі типових мікросхем меншої розмірності виду  $1 \rightarrow n$ .

Для цього потрібно використати  $L = n/n_1$  типових демультиплексорів з числом адресних входів  $m_1 = \log_2 n_1$  кожен. Число старших адресних розрядів, що дорівнює різниці  $m - m_1$ , використовується додатковим "ведучим" демультиплексором, який розташовується у першому рівні схеми каскадування. Ведучий демультиплексор визначає почергове увімкнення одного з  $L$  демультиплексорів мікросхем другого рівня. Каскадування демультиплексорів виду  $1 \rightarrow 4$  для реалізації комутатора  $1 \rightarrow 16$  показано на рис. 8.8.

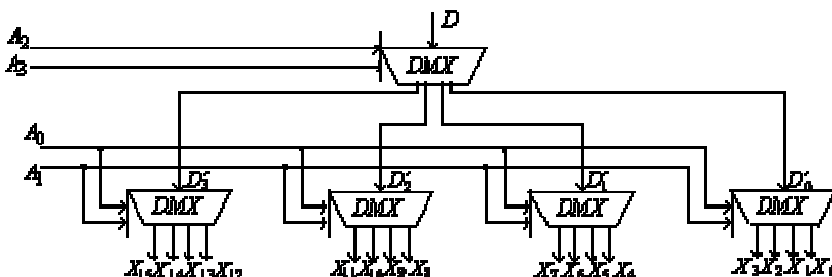


Рис. 8.8. Каскадування демультиплексорів

Нехай для схеми (рис. 8.8) адресний код  $A_3A_2A_1A_0 = 1010$  і значення вхідного сигналу  $D=1$ . Тоді на виході ведучого демультиплексора  $D_2 = A_3 \bar{A}_2 D = 1$ , а на інших виходах встановлюються нульові значення. Одиначне значення сигналу  $D_2$  передається на вихід  $X_{10}$  веденого демультиплексора згідно зі співвідношенням

$$X_{10} = A_1 \bar{A}_0 D_2 = 1.$$

Демультиплексори не випускають як самостійні вироби на інтегральних мікросхемах. Функцію демультиплексора звичайно ре-

алізують на дешифраторах, що мають входи стробування (дешифратори-демультиплексори).

### 8.7. Демультиплексування шин

Під демультиплексуванням шин розуміється почергове перемикання груп ліній від одного джерела інформації до багатьох приймачів. Такі мікрооперації реалізуються звичайно на основі демультиплексорів одиночних ліній. При виборі кількості і типу демультиплексора враховують:

число шин, які комутуються, дорівнює  $2^m$ , де  $m$  – довжина адресного коду;

кількість демультиплексорів, які використовуються, визначається розрядністю  $n$  шин, які демультиплексуються;

адресні входи всіх мультиплексорів паралельно об'єднуються.

Схема мультиплексора вхідної шини  $D(n)$  на чотири вхідні шини  $X(n)$ ,  $Y(n)$ ,  $Z(n)$  і  $S(n)$  показана на рис. 8.9.

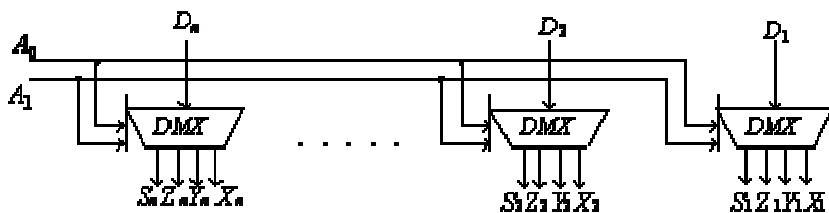


Рис. 8.9. Демультиплексор шин

Типове включення мультиплексорів і демультиплексорів для комутації вхідних і вихідних шин  $n$ -розрядних регістрів  $A$ ,  $B$ ,  $C$  і  $D$  показано на рис.8.10.

У АЛП така комутація забезпечує використання як першого операнда суматора вміст будь-якого регістра і запис результату операції в будь-який регістр, вказаний мікропрограмою команди, що виконується.

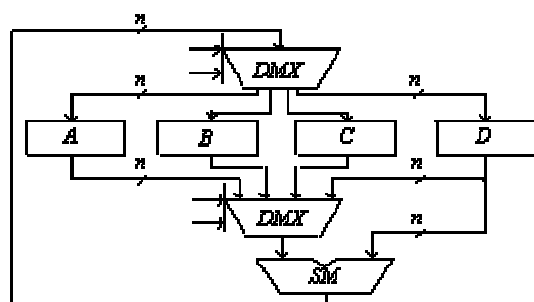


Рис. 8.10. Типова комутація вхідних і вихідних шин регістрів АЛП

## Компаратори. Схеми контролю

### 9.1. Загальна характеристика схем порівняння

Схемою порівняння (компаратором) називається функціональний вузол комп'ютера, призначений для вироблення ознак відношень між двійковими словами (числами). Ознаки відношень записуються у вигляді:

$F_i = A * K$  або  $F_i, A * K$  або  $FA * K$ ;

$F_i = A * B$  або  $P_i, A * B$  або  $FA * B$ ,

де  $A$  і  $B$  – двійкові або двійково-десяткові числа;  $K$  – двійкова константа;  $i$  – номер відношення (часто пропускається);  $*$  – операція відношення вигляду  $=$ ,  $\neq$ ,  $<$ ,  $>$ ,  $\leq$ ,  $\geq$  і т. ін.;  $F_i$  – функція, що задає результат відношення: лог.1 – якщо відношення виконується, тобто істинне, і лог.0 – якщо відношення не виконується, тобто помилкове. Функція компаратора позначається буквами COMP (comparator) або знаками  $=$ .

Основними відношеннями вважаються: «рівне»  $FA=B$ , «більше»  $PA>B$  і «менше»  $PA<B$ . Часто схеми, що реалізують відношення  $PA>B$  або  $PA<B$ , називають схемами порівняння «на більше» або «на менше». Маючи в своєму розпорядженні основні ознаки відношень, можна на їхній основі отримати ряд додаткових ознак, наприклад:

$FA \neq B = \overline{FA=B}$ ;  $FA \leq B = \overline{FA > B}$ ;  $FA \leq B = FA = B \vee FA < B$ .

Ознаки відношення використовуються як логічні умови (повідомляючі сигнали) в мікропрограмах, командах передачі керування, а також у пристроях контролю і діагностики. Після виконання кожної команди в машині автоматично формуються ознаки результатів операції. Ці ознаки, які називаються прапорами (прапорцями), вміщуються в спеціальний регістр прапорів. До прапорів звичайно відносять ознаки нульового результату, переповнення розрядної сітки, знак результату, наявність перенесень із старшого розряду суматора, парне або непарне число одиниць в результаті та ін.

Зазначимо, що формування і використання ознак (прапорців) – це основна відмінність комп'ютера від калькулятора. Тільки за допомогою прапорців машина приймає рішення про хід обчислювального процесу, тобто володіє інтелектуальними властивостями.

### 9.2. Схеми порівняння слів з константою

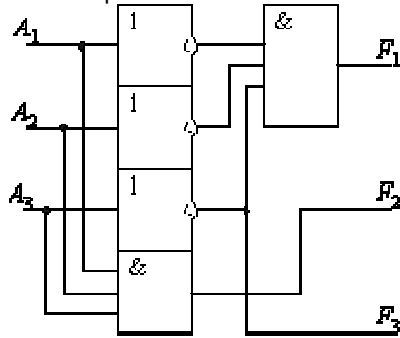
Прийmemo, що потрібно отримати ознаки відношень двійкового слова  $A=A_2A_1A_0$  з наступними заданими константами:

$F_1 := (A=000)$ ;  $F_2 := (A=111)$  і  $F_3 := (A \neq 011)$ .

На основі табл. 9.1 значення ознак відношення слова  $A$  з константами запишуться у вигляді:

$$F_1 = \overline{A_2 A_1 A_0}; F_2 = A_2 A_1 A_0; F_3 = \overline{A_3} \quad (9.1)$$

Схема порівняння слова з константою згідно з виразами (9.1) показана на рис. 9.1.



Таблиця 9.1

A2	A1	A0	F1	F2	F3	
0	0	0	1	0	1	
0	0	1	0	0	1	
0	1	0	0	0	1	
0	1	1	0	0	1	
1	0	0	0	0	0	
1	0	1	0	0	0	
1	1	0	0	0	0	
1	1	1	0	1	0	
						Рис. 9.1. Схема порівняння слова з константою

### 9.3. Схеми порівняння двійкових слів A і B

Багаторозрядні двійкові слова рівні, коли одночасно попарно рівні всі їхні розряди, тобто  $A(n) = B(n)$ , якщо  $A_i = B_i$ ,  $i = 1, 2, \dots, n$ . На основі табл. 9.2, яка задає умову рівності  $r_i$  двох  $i$ -х розрядів A і B, отримаємо:

$$r_i = \overline{A_i B_i} \vee A_i B_i = \overline{A_i \oplus B_i} = \overline{M_i} \quad (9.2)$$

де  $M_i$  – функція додавання по модулю два ("виключальне ЧИ").

Схемна реалізація функції (9.2) показана на рис. 9.2.

Ознака рівності двох  $n$ -розрядних слів  $PA=B$  визначається логічним добутком порозрядних умов  $r_i$ :

$$FA=B = r_n r_{n-1} \dots r_1 = \overline{M_n} \cdot \overline{M_{n-1}} \dots \overline{M_1}. \quad (9.3)$$

Таблиця 9.2		
$A_i$	$B_i$	$r_i$
0	0	1
0	1	0
1	0	0
1	1	1

Схема порівняння двох чотирирозрядних слів A і B згідно з виразом (9.3) показана на рис. 9.3. Схема вміщує чотири логічних елементи «виключальне ЧИ» і один кон'юнктор.

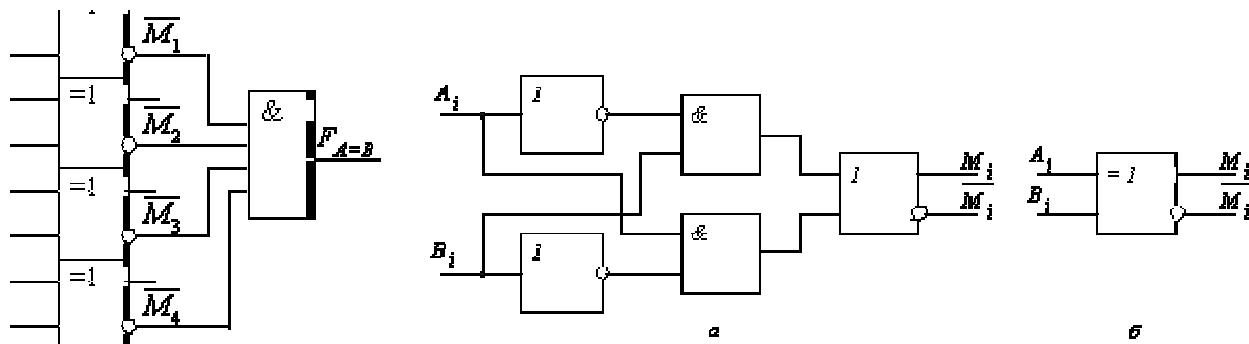


Рис. 9.2. Елемент "Виключальне ЧИ": а – схема; б – умовне позначення

Рис. 9.3. Схема порівняння двох чотирирозрядних слів А і В

При великій розрядності слів, які порівнюються, можна на першому рівні отримати ознаки для чотирирозрядних груп і на другому рівні реалізувати загальний прапор логічним множенням групових ознак. Наприклад, при розрядності порівнюваних слів  $n = 16$  отримаємо чотири групові ознаки порівняння:

$$F_{A=B}^{1,4}, F_{A=B}^{5,8}, F_{A=B}^{9,12}, F_{A=B}^{13,16}$$

де верхні індекси означають номери розрядів у групах. Тоді ознака порівняння двох 16-розрядних слів запишеться у вигляді:

$$F_{A=B} = F_{A=B}^{1,4} \cdot F_{A=B}^{5,8} \cdot F_{A=B}^{9,12} \cdot F_{A=B}^{13,16}$$

Схема порівняння двох 16-розрядних слів показана на рис. 9.4, а.

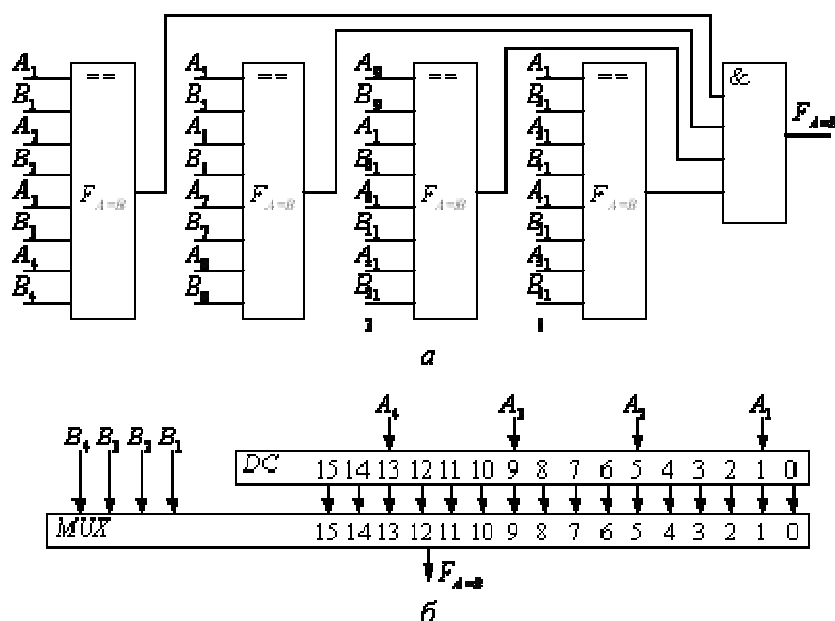


Рис. 9.4. Схема порівняння двох слів на рівність: а – групова структура; б – на основі дешифратора і мультиплексора

Порівняння може бути реалізоване і на інших схемотехнічних принципах. Схема порівняння двох чотирирозрядних чисел А і В на основі дешифратора і мультиплексора показана на рис. 9.4, б. Дешифратор виробляє одиничне значення сигналу на тому виході, номер якого також визначається десятковим еквівалентом вхідного коду. Наприклад, при  $A_4 A_3 A_2 A_1 = 0111$  логічна одиниця з'явиться на виході з номером сім. Мультиплексор підключає до виходу той вхід, номер якого також визначається десятковим еквівалентом вхідної комбінації. Якщо  $B_4 B_3 B_2 B_1 = 0111$ , то дозволяється проходження на вихід сигналу із сьомого входу. Таким чином, якщо слова А і В рівні, то формується прапор  $F_{A=B} = 1$ .

#### 9.4. Схеми порівняння двох слів «на більше»

Схема порівняння двох слів А і В «на більше» за абсолютним значенням виробляє ознаку  $F_{A>B}$  і будується за наступним алгоритмом:

аналіз нерівності слів А і В виконується послідовно в напрямку від старших розрядів до молодших;  
молодші розряди включаються в аналіз в тому випадку, коли старші розряди рівні (еквівалентні);



для отримання ознаки  $PA > B$  будеться диз'юнктивна сума порозрядних умов.

Логіка порівняння розрядів  $A$  і  $B$  наведена в табл. 9.3, де  $C_i$  – ознака  $A_i > B_i$ ;  $r_i$  – умова підключення до аналізу сусідніх молодших розрядів обох слів.

На основі табл. 9.3 отримуємо такі вирази:

$$C_i = A_i \bar{B}_i; r_i = \bar{A}_i \bar{B}_i \vee A_i B_i = \bar{A}_i \oplus \bar{B}_i = \bar{M}_i. \quad (9.4)$$

З урахуванням виразу (9.4) і алгоритму аналізу функцію ознаки  $FA > B$  представляємо у вигляді:

$$F_{A>B} = C_n \vee r_n C_{n-1} \vee \dots \vee r_n r_{n-1} \dots r_2 C_1. \quad (9.5)$$

Для порівняння двох чотирирозрядних слів «на більше» ознаку нерівності згідно з виразом (9.5) представляємо таким чином:

$$\begin{aligned} F_{A>B} &= C_4 \vee r_4 C_3 \vee r_4 r_3 C_2 \vee r_4 r_3 r_2 C_1 = \\ &= A_4 \bar{B}_4 \vee \bar{M}_4 A_3 \bar{B}_3 \vee \bar{M}_4 \bar{M}_3 A_2 \bar{B}_2 \vee \bar{M}_4 \bar{M}_3 \bar{M}_2 A_1 \bar{B}_1. \end{aligned} \quad (9.6)$$

Таблиця 9.3

$A_i$	$B_i$	$C_i$	$r_i$
0	0	0	1
0	1	0	0
1	0	1	0
1	1	0	1

Схема порівняння «на більше» двох чотирирозрядних слів  $A$  і  $B$  згідно із співвідношенням (9.6) показана на рис. 9.5.

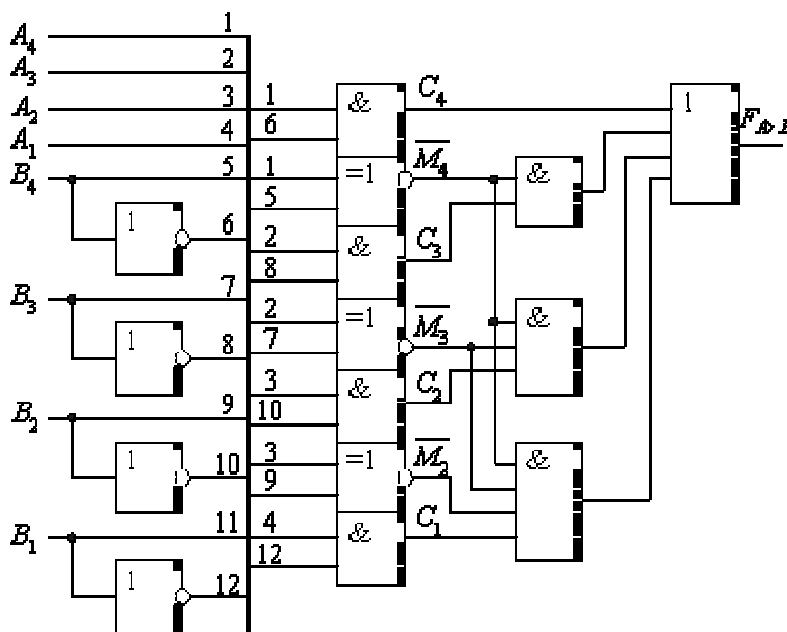


Рис. 9.5. Схема порівняння двох слів «на більше»

#### 9.5. Багаторозрядні схеми порівняння «на більше»

При реалізації схем порівняння багаторозрядних слів «на більше» виникають технічні труднощі, пов'язані з необхідністю використання вентилів з великою кількістю входів. Тому слова, що порівнюються, розбиваються на групи, які складаються, наприклад, з чотирьох розрядів. Кожна група виробляє свою ознаку нерівності  $F_{ia>b}$  і умову підключення до аналізу  $M_i$  молодшої групи згідно з виразом (9.6) і схемою (рис. 9.5). Наприклад, для  $n = 16$  маємо чотири групи, які об'єднуються згідно із співвідношенням

$$F_{A>B} = F_{A>B}^4 \vee M_{15}^4 F_{A>B}^3 \vee M_{15}^4 M_{12}^3 F_{A>B}^2 \vee M_{15}^4 M_{12}^3 M_{9}^2 F_{A>B}^1. \quad (9.7)$$

де  $F_{4A>B}$  – прапор порівняння «на більше» в найстаршій групі з розрядами  $A_{16} - A_{13}$ ,  $B_{16} - B_{13}$  і  $M_{4gr} = M_{16}M_{15}M_{14}M_{13}$  – умова для підключення до аналізу сусідньої молодшої групи;  $F_{3A>B}$  – прапор порівняння «на більше» у групі з розрядами  $A_{12} - A_{9}$ ,  $B_{12} - B_{9}$  і  $M_{3gr} = M_{12}M_{11}M_{10}M_{9}$  – умова аналізу молодшої групи;  $F_{2A>B}$  – прапор порівняння «на більше» у групі з розрядами  $A_8 - A_5$ ,  $B_8 - B_5$  і  $M_{2gr} = M_8M_7M_6M_5$  – умова підключення молодшої групи;  $F_{1A>B}$  – прапор порівняння «на більше» у групі з розрядами  $A_4 - A_1$ ,  $B_4 - B_1$ .

Схема порівняння «на більше» двох 16-розрядних слів  $A$  і  $B$  на основі рівняння (9.7) показана на рис. 9.6.

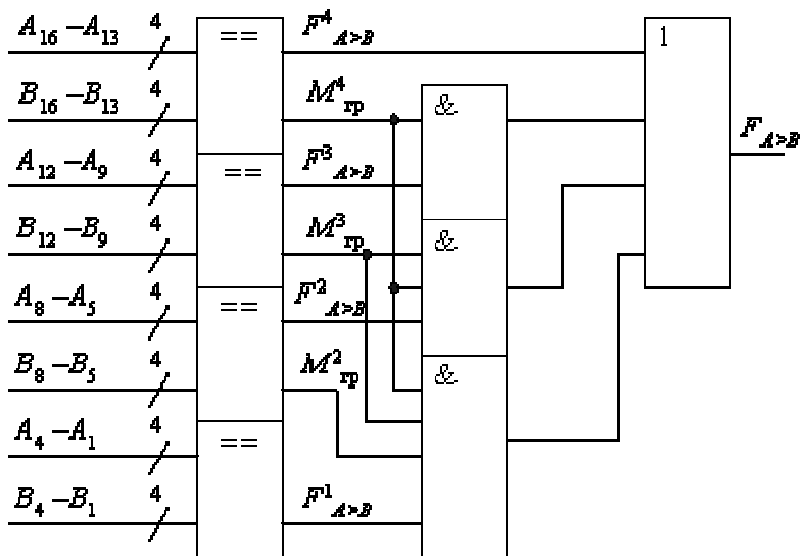


Рис. 9.6. Схема порівняння «на більше» двох 16-розрядних слів

#### 9.6. Застосування компараторів

Контроль (виявлення) і корекція (виправлення) результатів операцій є важливою умовою грамотної експлуатації машин. Контроль може бути програмним або апаратним. До апаратних методів відносяться дублювання операцій і відновлення вхідних сигналів.

Контроль операцій додавання методом дублювання реалізується двома однаковими суматорами (SM), на входи яких одночасно поступають доданки  $A(n)$  і  $B(n)$ . Обидва результати  $S1(n)$  і  $S2(n)$  поступають на входи схеми порівняння (рис. 9.7, а). Якщо обидва результати рівні, то на виході схеми порівняння значення ознаки  $FS1=S2 = 1$  і помилок немає. При нульовому значенні ознаки операцію потрібно повторити або зупинити роботу EOM.

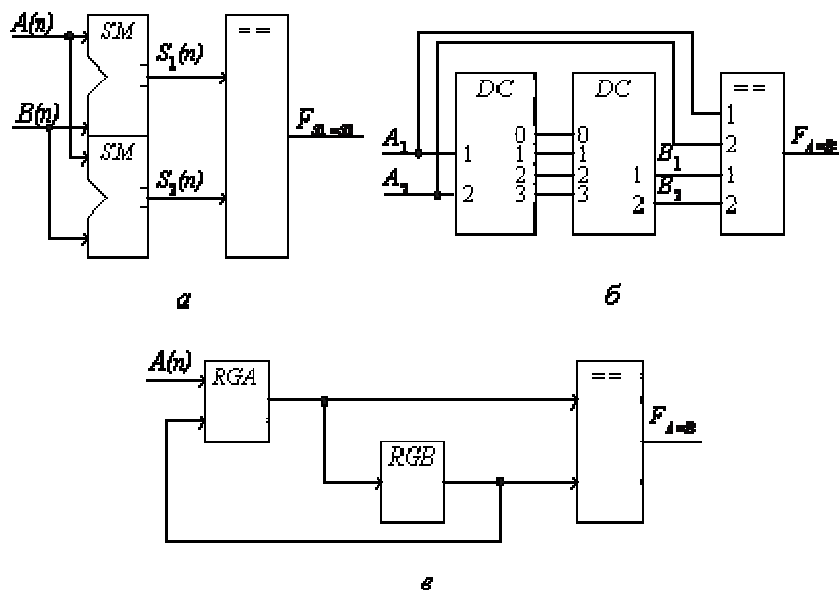


Рис. 9.7. Застосування схеми порівняння для контролю операцій

Схема контролю методом відновлення вхідних сигналів показана на рис. 9.7, б. Дворозрядне слово  $A2A1$  декодується і значення унітарного коду з виходів дешифратора поступає на входи шифратора. При правильній роботі дешифратора і шифратора вхідний код  $A2A1$  має збігатися з вихідним кодом шифратора  $B2B1$ . При цьому на виході схеми порівняння встановиться одиничне значення ознаки  $FA=B$ .

При передачі інформації з одного регістра в інший контроль правильності пересилки може здійснюватися порозрядним порівнянням вмісту цих двох регістрів. На рис. 9.7, в показаний один з варіантів контролю пересилок слів між регістрами. Після передачі інформації з регістра  $A$  в регістр  $B$  (або навпаки) проводиться порівняння їхнього вмісту. Якщо значення двох слів збігаються, то значення ознаки рівності набуває одиничного значення, інакше – виробляється сигнал помилки.

#### 9.7. Загальна характеристика схем контролю парності

У комп'ютерах широко використовується контроль парності (синонім – за паритетом або відповідністю). Цей спосіб заснований на допущенні, що в двійковому числі найчастіше виникають одиничні помилки – втрата або поява зайвої одиниці. У обох випадках число одиниць зміниться на одну. Якщо двійкове число мало непарне число одиниць, то після одиничної помилки воно виявиться парним і навпаки.

На практиці контроль парності здійснюється таким чином. Для підвищення ефективності контролю двійкове слово розбивається на частини, як правило, байти. До кожного байта додається додатковий контрольний розряд. Вміст контрольного розряду залежить від вибраного способу контролю (за парністю або непарністю). При контролі за парністю значення контрольного розряду вибирається таким, щоб загальне число одиниць у байті й контрольному біті було парним. У цьому випадку значення кон-

трольного (паритетного) біта визначається додаванням за модулем два значень розрядів байта (рис. 9.8, а):

$$F_{\text{КП}} = A_1 \oplus A_2 \oplus A_3 \oplus A_4 \oplus A_5 \oplus A_6 \oplus A_7 \oplus A_8. \quad (9.8)$$

Внаслідок операції додавання за модулем два значень розрядів байта з парним число одиниць одержуємо значення контрольного байта  $F_{\text{К.П}} = 0$ . При додаванні за модулем два значень розрядів байта з непарним числом одиниць значення контрольного байта  $F_{\text{К.П}} = 1$ .

FK.П	8	7	6	5	4	3	2	1		FK.Н	8	7	6	5	4	3	2	1
0	1	0	1	0	1	1	1	1		1	1	0	1	0	1	1	1	1
1	1	1	1	0	1	1	0	0		0	1	1	1	0	1	1	0	0
а										б								

Рис. 9.8. Контроль байта: а – за парністю; б – за непарністю

При контролі за непарністю значення контрольного біта вибирається з умови, щоб кількість одиниць у байті з урахуванням вмісту контрольного розряду була непарною. У цьому випадку значення контрольного біта набуває такого виразу:

$$F_{\text{КП}} = A_1 \oplus A_2 \oplus A_3 \oplus A_4 \oplus A_5 \oplus A_6 \oplus A_7 \oplus A_8 = \overline{F_{\text{КП}}}. \quad (9.9)$$

На практиці контроль непарності використовується частіше, оскільки фіксує повне пропадання інформації. Контроль парності (непарності) передбачає формування значень контрольних розрядів до виконання операції та перевірку байта після виконання операції з урахуванням контрольних розрядів. Наприклад, при записуванні байта в пам'ять комп'ютера одночасно автоматично формується (генерується) значення його контрольного розряду. При зчитуванні байта, що зберігається, здійснюється додавання за модулем два значень його розрядів спільно з контрольним бітом згідно з визначеним способом контролю парності або непарності. Таким чином, контроль за паритетом вимагає використання додаткових розрядів. Схеми, що забезпечують отримання значення контрольного розряду і перевірку двійкового числа за ознакою парності або непарності, називаються схемами контролю парності. Їх часто називають схемами згортки, схемами контролю за модулем два, схемами контролю за паритетом. Для отримання умови парності потрібне складання за модулем два восьмирозрядного слова, що реалізується за допомогою ступінчатого включення двохходових елементів «виключальне ЧИ»:

на першому рівні отримують функції  $F_1 - F_4$ :

$$F_1 = A_1 \wedge A_2; \quad F_2 = A_3 \wedge A_4; \quad F_3 = A_5 \wedge A_6; \quad F_4 = A_7 \wedge A_8; \quad (9.10)$$

на другому і третьому рівнях реалізуються функції:

$$F_5 = F_1 \wedge F_2; \quad F_6 = F_3 \wedge F_4; \quad M = F_5 \wedge F_6. \quad (9.11)$$

Функція  $M$  згідно з виразами (9.10) і (9.11) набуває значення лог.1 при непарному числі одиниць у вхідному байті та значення лог.0 – при парному числі одиниць у вхідному байті.

Для задання ознаки контролю вводиться керуючий сигнал  $V$ , який разом з сигналом  $M$  поступає на входи схеми «виключальне ЧИ» в четвертому рівні; на прямому й інверсному виходах цього рівня формуються пряме й інверсне значення контрольного розряду:

$$F = M \oplus V; \quad \overline{F} = \overline{M \oplus V}.$$

Логіка роботи схеми контролю, показаної на рис. 9.9, а, наведена в табл. 9.4.

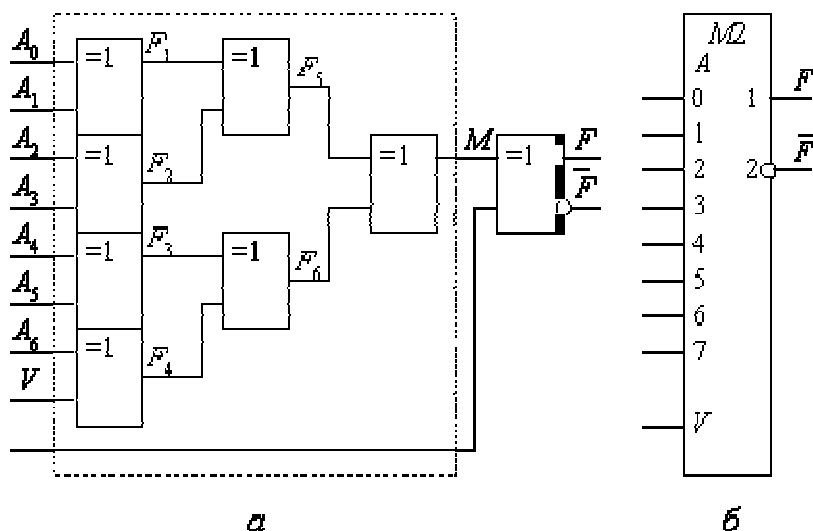


Рис. 9.9. Схеми контролю за парністю: а – ступінчате включення елемента «виключальне ЧИ»; б – умовне позначення

Таблиця 9.4

Входи A8 – A1		V	F	$\overline{F}$
На входах:	Парне число одиниць	0	0	1
	Непарне число одиниць	0	1	0
На входах:	Парне число одиниць	1	1	0

апаратними витратами: вартість однорозрядної схеми додавання визначається загальним числом логічних входів використаних елементів; вартість багаторозрядного суматора визначається загальною кількістю використаних мікросхем; споживаною потужністю суматора.

## 10.2. Однорозрядні суматори

Однорозрядним суматором називається логічна схема, яка виконує додавання значень  $i$ -х розрядів  $X_i$  та  $Y_i$  двійкових чисел з урахуванням перенесення  $Z_i$  з молодшого сусіднього розряду та виробляє на виходах функції результат  $S_i$  і перенесення  $P_i$  в старший сусідній розряд. На основі однорозрядних схем додавання на три входи та два виходи будуються багаторозрядні суматори будь-якого типу. Алгоритм роботи однорозрядного суматора відображається таблицею істинності (табл. 10.1).

На основі табл. 10.1 записується система логічних функцій для результату  $S_i$  та перенесення  $P_i$  у ДДНФ:

$$S_i = \overline{X_i} \overline{Y_i} Z_i \vee \overline{X_i} Y_i \overline{Z_i} \vee X_i \overline{Y_i} \overline{Z_i} \vee X_i Y_i Z_i; \quad (10.1)$$

$$P_i = \overline{X_i} Y_i Z_i \vee X_i \overline{Y_i} Z_i \vee X_i Y_i \overline{Z_i} \vee X_i Y_i Z_i. \quad (10.2)$$

Мінімізація функцій (10.1) та (10.2) за допомогою карт Карно показана на рис. 10.1.

Як видно з карт Карно, функція результату  $S_i$  не мінімізується, а функція  $P_i$  мінімізується зі зниженням рангу кон'юнкції та використовує тільки прямі значення змінних:

$$P_i = X_i Y_i \vee X_i Z_i \vee Y_i Z_i = X_i Y_i \vee (X_i \vee Y_i) Z_i. \quad (10.3)$$

				$X_i Y_i$					$X_i Y_i$
00	01	11	10	$Z_i$	00	01	11	10	$Z_i$
0	1	0	1	0	0	0	1	0	0
1	0	1	0	1	0	1	0	1	1

а                                      б

Рис. 10.1. Карти Карно для мінімізації функцій: а –  $S_i$ ; б –  $P_i$

При проектуванні комбінаційних однорозрядних суматорів враховують такі чинники:

схема має характеризуватися регулярністю (подібністю) структури та мінімальною вартістю, тобто мати по можливості найменше число логічних входів всіх елементів;

Таблиця 10.1

$X_i$	$Y_i$	$Z_i$	$S_i$	$P_i$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

з метою підвищення швидкодії багаторозрядного суматора потрібен мінімальний час одержання функції перенесення  $t_P = k \cdot t_P$ , де  $k$  – число послідовно увімкнених елементів від входів до виходів  $P_i$

або  $\overline{P_i}$ ;  $t_P$  – середня затримка розповсюдження сигналу одним логічним елементом в обраній серії інтегральних мікросхем; параметр  $k$  часто називають каскадністю (поверховістю) схем. Таким чином, для мінімізації часу одержання перенесення необхідно зменшити каскадність схеми та використати інтегральні мікросхеми з малим часом затримки розповсюдження сигналу;

для схем однорозрядних суматорів на основі рівнянь (10.1) і (10.2) необхідно виробляти як прямі  $P_i$ ,

так й інверсні  $\overline{P_i}$  значення функції перенесення. Така організація перенесень називається парафазною.

Для побудови схеми однорозрядного суматора на універсальних логічних елементах НЕ І рівняння (10.1) і (10.2) перетворюються на основі правил подвійної інверсії та де Моргана до такого вигляду:

$$S_i = \overline{\overline{X_i} \overline{Y_i} Z_i} \cdot \overline{\overline{X_i} Y_i \overline{Z_i}} \cdot \overline{\overline{X_i} \overline{Y_i} \overline{Z_i}} \cdot \overline{\overline{X_i} Y_i Z_i}; \quad P_i = \overline{X_i Y_i} \cdot \overline{X_i Z_i} \cdot \overline{Y_i Z_i}. \quad (10.4)$$

Схема однорозрядного суматора, побудована на елементах НЕ І відповідно до рівнянь (4.30), показана на рис. 10.2, а; її вартість, яка вимірюється числом логічних входів всіх елементів, становить 27, каскадність  $k=3$ .

Рівняння (10.1) та (10.2) можуть бути виражені через функцію «Виключальне ЧИ»:

$$S_i = (X_i \oplus Y_i) \overline{Z_i} \vee (\overline{X_i} \oplus \overline{Y_i}) Z_i = X_i \oplus Y_i \oplus Z_i; \quad (10.5)$$

$$P_i = X_i Y_i \vee (\overline{X_i} Y_i \vee X_i \overline{Y_i}) Z_i = X_i Y_i \vee (X_i \oplus Y_i) Z_i. \quad (10.6)$$

б

Схема однорозрядного суматора на елементах «виключальне ЧИ» згідно з рівняннями (10.5) і (10.6) показана на рис. 10.2, б; її вартість становить вісім входів і каскадність  $k=2$ .

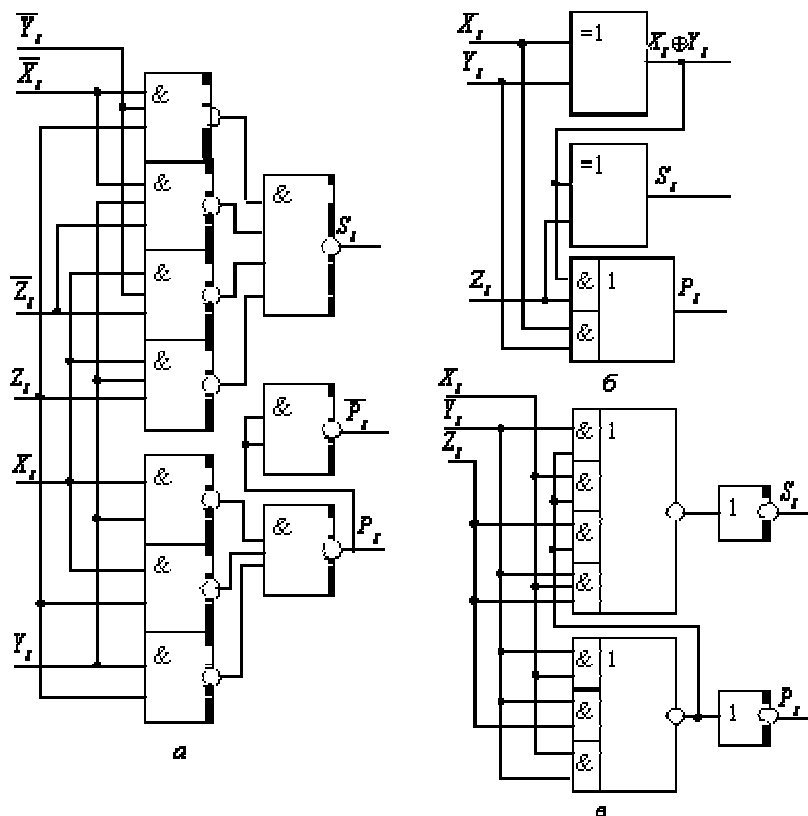


Рис. 10.2. Схеми однорозрядних суматорів: а – на елементах НЕ I; б – на елементах «виключальне ЧИ»; в – з використанням власного перенесення

Функції однорозрядного суматора – самоподвійні, тобто їхні інверсії утворюються інвертуванням значень аргументів без зміни місцезнаходження знаків диз'юнкції та кон'юнкції, наприклад, для перенесення з рівняння (10.2):

$$\bar{P}_i = X_i \bar{Y}_i \bar{Z}_i \vee \bar{X}_i Y_i \bar{Z}_i \vee \bar{X}_i \bar{Y}_i Z_i \vee X_i Y_i Z_i. \quad (10.7)$$

Помножуючи ліві та праві частини співвідношення (10.7) на макстерм  $(X_i Y_i U_i Z_i)$ , одержують:

$$\bar{P}_i X_i \vee \bar{P}_i Y_i \vee \bar{P}_i Z_i = X_i \bar{Y}_i \bar{Z}_i \vee \bar{X}_i Y_i \bar{Z}_i \vee \bar{X}_i \bar{Y}_i Z_i. \quad (10.8)$$

Після підстановки лівої частини співвідношення (10.8) в праву частину виразу (10.1) одержують рівняння для функції  $S_i$  з використанням власного перенесення:

$$S_i = \bar{P}_i X_i \vee \bar{P}_i Y_i \vee \bar{P}_i Z_i \vee X_i Y_i Z_i. \quad (10.9)$$

Схема однорозрядного суматора відповідно до рівнянь (10.9) і (10.3) показана на рис. 10.2, в; її вартість дорівнює 17 входів, каскадність  $k=2$ . Важливою властивістю цієї схеми є використання тільки прямих значень вхідних змінних і однофазного ланцюга формування перенесення  $P_i$  в старший розряд.

Напівсуматором називається логічна схема, яка виконує додавання значень  $i$ -х розрядів  $X_i$  і  $Y_i$  двійкових чисел  $X$  і  $Y$  та реалізує на виході значення результату  $M_i$  і перенесення в старший сусідній розряд  $R_i$ :

$$M_i = \bar{X}_i Y_i \vee X_i \bar{Y}_i = X_i \oplus Y_i; \quad R_i = X_i Y_i. \quad (10.10)$$

Таким чином, напівсуматор виконує лише частину завдання підсумовування в  $i$ -му розряді, оскільки не враховує перенесення з сусіднього молодшого розряду. Схема напівсуматора, побудована на основі рівнянь (10.10), показана на рис. 10.3. З рівнянь (10.5) і (10.6) виходить, що схема однорозрядного суматора може бути побудована на основі двох напівсуматорів і додаткового логічного елемента ЧИ, як показано на рис. 10.3, в.

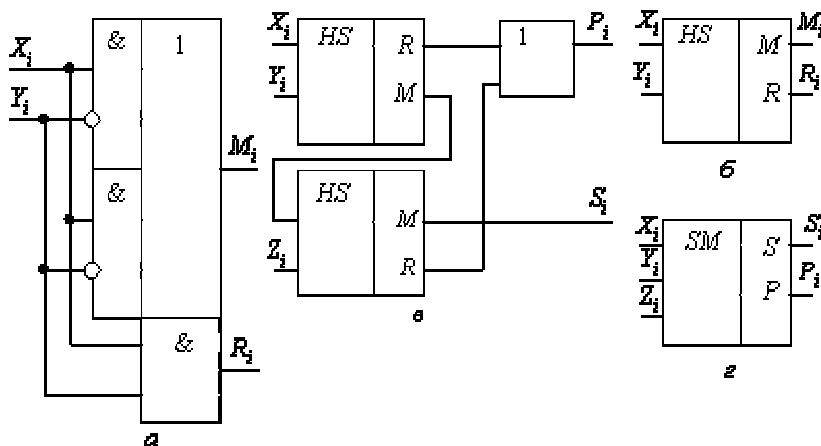


Рис. 10.3. Схеми підсумовування: а, б – напівсуматор і його умовне позначення; в, г – однорозрядний суматор і його умовне позначення

### 10.3. Послідовний багаторозрядний суматор

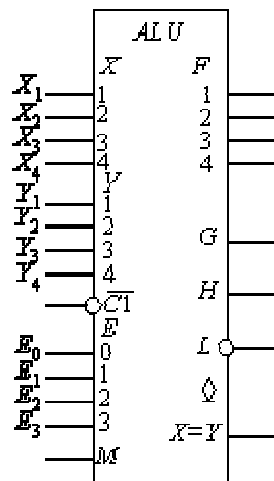


Рис. 10.6. Умовні позначення мікросхем: а – ALU 1533IP3; б – CRU 1533IP4

Послідовний двійковий багаторозрядний суматор містить: n-розрядні зсуваючі регістри операндів X і Y, регістр результату S, однорозрядний суматор SM і двоступеневий D-тригер для запам'ятовування перенесення. Усі регістри забезпечують одночасне зсування праворуч, у бік молодших розрядів (рис. 10.4).

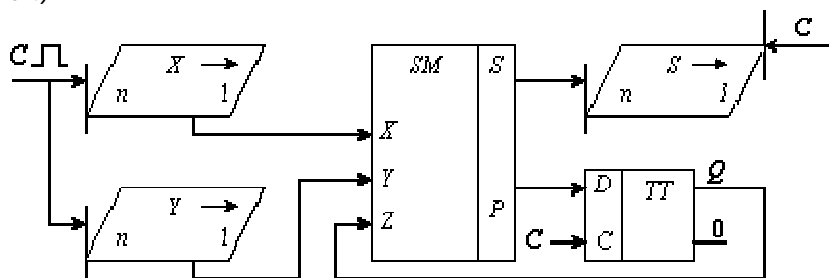


Рис. 10.4. Схема послідовного багаторозрядного суматора

У послідовному суматорі попарна подача значень розрядів  $X_i$  і  $Y_i$  починається з молодших розрядів. Утворюються значення суми  $S_i$  і перенесення  $P_i$ , які записуються відповідно в регістр результату та в тригер запам'ятовування перенесення на один такт  $T_c$ .

Послідовне додавання виконується за стільки тактів, скільки розрядів у числі. Тому час додавання  $t_S$  визначається співвідношенням:  $t_S = nT_c$ , де  $T_c$  – тривалість машинного такту.

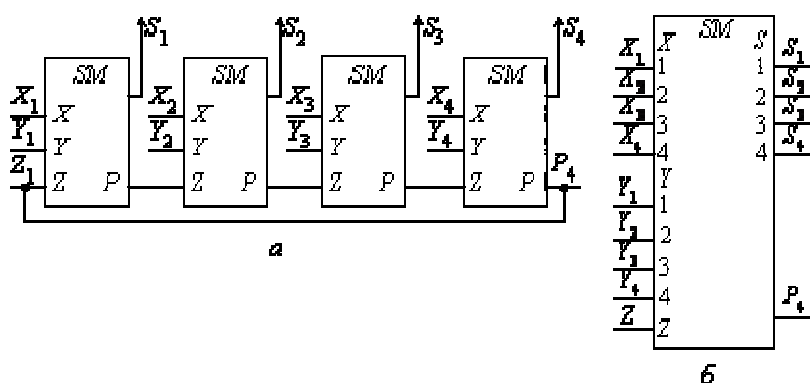
Від'ємні числа рекомендується представляти в доповняльному коді.

Послідовний суматор потребує мінімальних апаратних витрат, однак тривалість операції додавання пропорційна розрядності операндів. Тому послідовний суматор можна використовувати у відносно повільнодіючих цифрових пристроях.

#### 10.4. Паралельні багаторозрядні суматори

Паралельний багаторозрядний суматор містить n однорозрядних схем додавання, наприклад, чотири, як показано на рис. 10.5.

Рис. 10.5. Паралельний чотирирозрядний суматор: а – схема; б – умовне позначення



Значення всіх розрядів двох чисел X та Y поступають на входи відповідних однорозрядних суматорів паралельно (одночасно). В паралельних суматорах з послідовним перенесенням значення сигналу перенесення  $P_i$  передається від розряду до розряду послідовно в часі (асинхронно). При застосуванні оберненого коду перенесення з найстаршого розряду подається на вхід перенесення молодшого розряду по ланцюзі циклічного перенесення (рис. 10.5, а). При застосуванні доповняльного коду ланцюг циклічного перенесення розривається, а на вхід перенесення молодшого розряду подається логічний нуль.

У паралельних суматорах з послідовним перенесенням час додавання визначається співвідношенням:

$$t_S = (n-1)t_P + t_S,$$

де  $t_P$  – час формування перенесення в кожному розряді,  $t_S$  – час додавання в найстаршому розряді. У гіршому випадку можливий варіант, коли сигнал перенесення послідовно розповсюджується від першого до n-го розряду.

#### 10.5. Мікросхеми ALU

Промисловість випускає мікросхеми із символом функції ALU для виконання 16 арифметичних та 16 порозрядних логічних мікрооперацій залежно від вхідних сигналів настройки. У серіях ТТЛШ 530, 531, 533, 555 та 1533 вони мають позначення ИП3; в серіях ЕЗЛ 100, 500 і 700 використовують позначення ИП179.

Мікросхема ALU в серіях ТТЛШ має (рис. 10.6):

- інформаційні входи для подання двох чотирирозрядних операндів X і Y;
- входи налаштування E3–E0 для задання номера однієї з мікрооперацій;
- вхід M для задання типу мікрооперації: M=0 – арифметичні, M=1 – логічні;

- вхід перенесення  $\overline{C}_1$ , необхідний тільки при виконанні арифметичних мікрооперацій;
- виходи: результату мікрооперації S4–S1, послідовного перенесення L, генерації G, транзиту H, а також вихід з відкритим колектором від внутрішнього компаратора для вироблення ознаки рівності операндів FA=B.

Перелік арифметичних і логічних операцій, які виконують ALU, наведений у табл. 10.2. При виконанні логічних операцій перенесення між розрядами не використовується. Арифметичні операції реалізуються з урахуванням перенесень і позик. В арифме-



тичні операції включені фрагменти логічних дій. Наприклад, запис  $(X \vee Y) + X\bar{Y}$  означає, що спочатку виконується операція інверсії ( $\bar{Y}$ ), потім – логічного додавання ( $X \vee Y$ ) та логічного множення ( $X \cdot \bar{Y}$ ), а потім одержані таким чином два числа додаються арифметично з урахуванням перенесень.

Таблиця 10.2

E3	E2	E1	E0	Логіка M=1	Арифметика M=0
0	0	0	0	$\bar{X}$	$X$
0	0	0	1	$\overline{X \vee Y}$	$X \vee Y$
0	0	1	0	$XY$	$X \vee \bar{Y}$
0	0	1	1	0	-1
0	1	0	0	$\overline{XY}$	$X + X\bar{Y}$
0	1	0	1	$\bar{Y}$	$(X \vee Y) + X\bar{Y}$
0	1	1	0	$X \oplus Y$	$X - \bar{Y} - 1$
0	1	1	1	$X\bar{Y}$	$X\bar{Y} - 1$
1	0	0	0	$\bar{X} \vee Y$	$X + XY$
1	0	0	1	$\overline{X \oplus Y}$	$X + Y$
1	0	1	0	$Y$	$(X + \bar{Y}) + XY$
1	0	1	1	$X \cdot Y$	$XY - 1$
1	1	0	0	1	$X + Y$
1	1	0	1	$X \vee \bar{Y}$	$(X \vee Y) + X$
1	1	1	0	$X \vee Y$	$(X \vee \bar{Y}) + X$
1	1	1	1	$X$	$X - 1$

Мікросхема ALU виконує операцію арифметичного додавання двох чотирирозрядних операндів X і Y, якщо на входи настроювання подані сигнали E3E2E1E0=1001 та M=0. В цьому випадку мікросхема ALU виконує функцію суматора.

#### 10.6. ДВІЙКОВО-ДЕСЯТКОВІ СУМАТОРИ

Двійково-десяткові суматори використовуються для обробки масивів десяткової інформації за порівняно простими алгоритмами, оскільки при цьому вилучаються витрати часу на переведення чисел з десяткової системи числення в двійкову і навпаки. Кожна десяткова цифра  $X_i$  кодується двійковим кодом прямого заміщення "8421" (двійковою тетрадою), тобто  $X_i = X_i4X_i3X_i2X_i1$  і  $Y_i = Y_i4Y_i3Y_i2Y_i1$ . Наприклад  $X_i = 710 = 01112-10$ ,  $Y_i = 910 = 10012-10$ ; для дворозрядних десяткових чисел:  $X_iX_{i-1} = 1610 = 000101102-10$ ;  $Y_iY_{i-1} = 2810 = 001010002-10$ .

Один розряд двійково-десятькового суматора (декада) містить чотирирозрядний суматор SM1 для одержання попередньої суми в тетраді, чотирирозрядний суматор SM2 для корекції результату та логічний елемент І ЧИ для вироблення ознак корекції, як показано на рис.10.7.

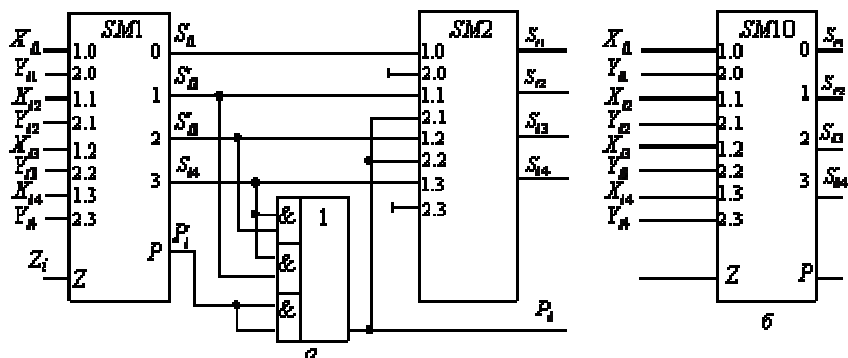


Рис.10.7. Однорозрядний двійково-десятковий суматор: а – схема; б – умовне позначення

Декада працює таким чином. Двійкові тетради десятичних цифр  $X_i = X_i4X_i3X_i2X_i1$  і  $Y_i = Y_i4Y_i3Y_i2Y_i1$  разом із перенесенням поступають на входи суматора SM1 і на його виходах утворюється попередня сума  $S_iT = S_i4S_i3S_i2S_i1$ , де  $S_iT$  – десятичний еквівалент тетради (табл. 10.3).

Таблиця 10.3

До корекції					Після корекції					Примітка
P	S4	S3	S2	S1	P?	S?4	S?3	S?2	S?1	
0	0	0	0	0	0	0	0	0	0	0?S <sub>i</sub> T?9 Корекція не потрібна
0	0	0	0	1	0	0	0	0	1	
0	0	0	1	0	0	0	0	1	0	
...	...	...	...	...	...	...	...	...	...	
0	1	0	0	1	0	1	0	0	1	10?S <sub>i</sub> T?15 Корекція потрібна: мінус 10 і перенесення в старшу декаду
1	0	0	0	0	0	1	0	1	0	
1	0	0	0	1	0	1	0	1	1	
...	...	...	...	...	...	...	...	...	...	
1	0	0	0	0	0	1	1	1	0	16?S <sub>i</sub> T?19 Корекція потрібна: плюс 6
1	0	1	0	1	0	1	1	1	1	
1	0	1	1	0	1	0	0	0	0	
1	0	1	1	1	1	0	0	0	1	
1	1	0	0	0	1	0	0	1	0	
1	1	0	0	1	1	0	0	1	1	

При цьому можливі три випадки:

1) для значення 0 ?  $S_iT$  ? 9 корекція не потрібна;

2) для значень 10 ?  $S_iT$  ? 15 потрібно відняти з попередньої суми число 10 і здійснити перенесення в старшу сусідню декаду; віднімання числа 10 в доповняльному коді відповідає додаванню за допомогою суматора SM2 до попереднього результату числа шість, тобто плюс 01102; ознакою такої корекції є одиничне значення функції корекції суми та перенесення  $F_iT = S_i4S_i3 \vee S_i4S_i2$ , (10.11)

яке реалізується елементом І ЧИ;

3) для значень 16 ?  $S_iT$  ? 19 на виході суматора SM1 виникає перенесення  $PiT$  з вагою 1610. Однак у старшій декаді його значення сприймається як 10, тому потрібно додати до попереднього результату за допомогою суматора число шість, тобто 01102.

З урахуванням рівняння (10.11) функцію корекції результату та перенесення можна записати у вигляді:

$$P_iT = P_iT \vee F_iT = P_iT \vee S_i4S_i3 \vee S_i4S_i2.$$

Таким чином, в усіх випадках, коли  $P_iT = 1$ , до попередньої суми додається число плюс 01102 і формується перенесення у старший розряд.

Схема чотирирозрядного двійково-десятькового суматора з послідовним перенесенням в тетрадах і між декадами показана на рис. 10.8. Швидкодія таких суматорів розраховують за аналогією з двійковими послідовними перенесеннями. Для двійково-десятькових суматорів можна використовувати групові структури прискорених перенесень.

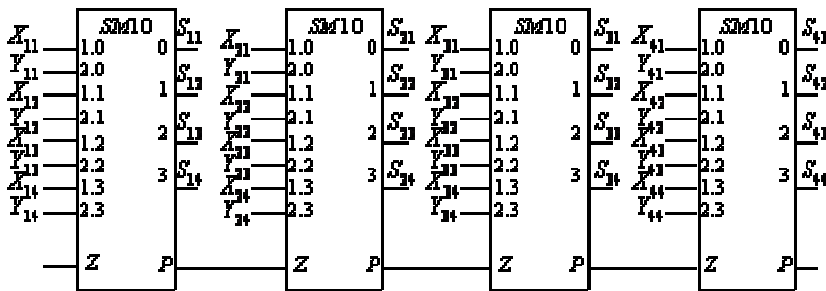


Рис. 10.8. Схема чотирирозрядного двійково-десятькового суматора.

Операція віднімання в двійково-десятьковому суматорі замінюється додаванням операндів у оберненому або доповняльному кодах. Обернений код від'ємних десятикових чисел одержують заміною кожної цифри її доповненням до дев'яти. Схема одного десятикового суматора з перетворювачами прямого коду операндів і результату в обернений код показана на рис. 10.9.

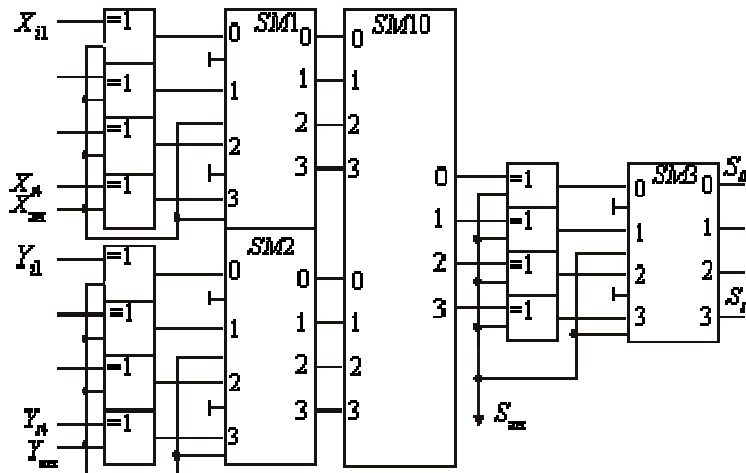


Рис. 10.9. Схема одного розряду десятикового суматора з перетворювачами прямого коду в обернений

Значення від'ємних чисел при  $X_{zn}=1$ ,  $Y_{zn}=1$ ,  $S_{zn}=1$  інвертується схемою "виключальне ЧИ"; при цьому утворюється двійковий код тетрад з надлишком шість. Корекцію результату виконують суматорами SM1, SM2 і SM3, в яких віднімання замінюється додаванням двійкової тетради з оберненим кодом числа шість, тобто плюс 10102.

## Аналогово-цифрові, цифро-аналогові перетворювачі. Кодоперетворювачі

### 12.1. Загальна характеристика цифро-аналогових перетворювачів

Цифро-аналогові перетворювачі (ЦАП) призначені для перетворення цифрової інформації в аналогову форму у вигляді напруги (іноді струму). Їх використовують у системах керування технологічними процесами, в аналогових мікропроцесорах, в дисплеях, графопобудовниках, робототехніці.

Цифро-аналогове перетворення полягає в тому, що для вхідного паралельного  $n$ -розрядного коду

$$X = X_{12} \cdot 1 + X_{22} \cdot 2 + \dots + X_{i2} \cdot i + \dots + X_{n2} \cdot n,$$

де  $X_i$  – цифри 0 або 1, а  $2-i$  – вага  $i$ -го розряду, спочатку отримують струм  $I_X$ , пропорційний значенню числа  $X$ , а потім перетворюють його у вихідну напругу. Значення струму  $I_X$  визначається сумою еталонних струмів  $I_i$ , які створюються для кожного розряду числа

$$I_X = X_{11} I_1 + X_{21} I_2 + \dots + X_{i1} I_i + \dots + X_{n1} I_n,$$

причому підсумовуються струми тільки тих розрядів, для яких  $X_i=1$ . Значення еталонів струму  $I_i$  пропорційні вазі позиції двійкового числа і зменшуються у два рази при переході від старшого  $i$ -го розряду до сусіднього молодшого з номером  $i+1$ .

### 12.2. Схеми цифро-аналогових перетворювачів

Структура ЦАП вміщує: резистивну або транзисторну матрицю для формування еталонних струмів; ключі для комутації еталонних струмів згідно з вхідним кодом до спільної точки підсумовування; операційний підсилювач (ОП) для перетворення струму  $I_X$  у вихідну напругу; допоміжні схеми для узгодження з вхідними рівнями сигналів; стабілізоване джерело опорної напруги УОП. Резистивні матриці будують або з набору двійково-зважених за номіналами резисторів, або у вигляді сходового (багато-ланкового) ланцюжка резисторів лише двох номіналів  $R-2R$ . Схема ЦАП з резистивною матрицею на основі двійково-зважених опорів виду  $R-2R-\dots-2^{n-1} R$  показана на рис. 12.1. У цій схемі опір резисторів матриці подвоюється при переході від старшого розряду до молодшого, а еталонні струми зменшуються у два рази. Наприклад, якщо для першого, найстаршого розряду взяти значення струму  $I_1=1$  мА, то для другого розряду  $I_2=0,5$  мА, для третього  $I_3=0,25$  мА і т. д.

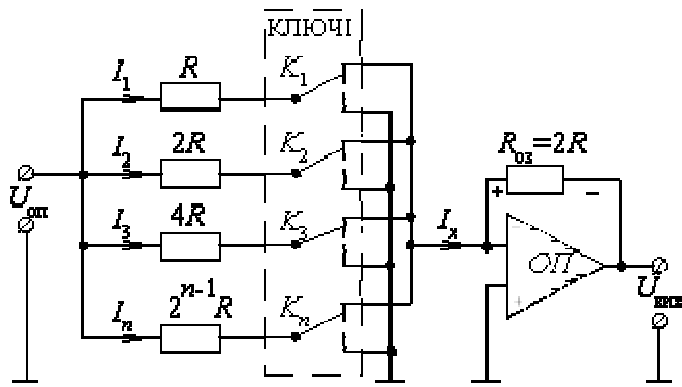


Рис. 12.1. Схема ЦАП із зваженими резисторами

Ключі  $K_1 - K_n$  керуються рівнями напруги, які відображають цифри «нуль» і «один» відповідних розрядів вхідного коду. Джере-ло опорної напруги  $U_{оп}$  найчастіше буває зовнішнім, але у деяких випадках його вбудовують у мікросхему ЦАП. На вході ОП завжди є практично нульовий потенціал, тому додавання розрядних струмів визначається співвідношенням

$$I_x = \frac{U_{оп}}{R} X_1 + \frac{U_{оп}}{2R} X_2 + \frac{U_{оп}}{4R} X_3 + \dots + \frac{U_{оп}}{2^{n-1}R} X_n =$$

$$= \frac{U_{оп}}{R} (X_1 + 2^{-1} X_2 + 2^{-2} X_3 + \dots + 2^{-(n-1)} X_n) = \frac{U_{оп}}{R} \sum_{i=1}^n X_i 2^{-(i-1)}, \quad (12.1)$$

$$I_x = (U_{оп} / R) \sum_{i=1}^n X_i \cdot 2^{-(i-1)}.$$

Напругу на виході ЦАП розраховують за формулою

$$U_{вых} = -I_x R_{оз} = -U_{оп} \sum_{i=1}^n X_i \cdot 2^{-i}, \quad (12.2)$$

де  $R_{оз} = R/2$  – опір у ланцюгу оберненого зв'язку підсилювача.

### 12.3. Двійкові коди, використовувані у цифро-аналогових перетворювачах

Введення інформації в ЦАП здійснюється, в основному, в паралельному коді. У ЦАП використовують три основних двійкових коди: прямий, зміщений і доповняльний (рис. 12.2).

Прямий код зручний при перетворенні сигналів систем стеження тому, що при переході через нуль не міняються старші розряди коду, а це дозволяє реалізувати лінійний перехід від малих позитивних до малих негативних вихідних напруг. Для перетворення позитивних і негативних кодів використовують знаковий розряд, який керує перемиканням вихідної напруги ЦАП (рис. 12.2, а).

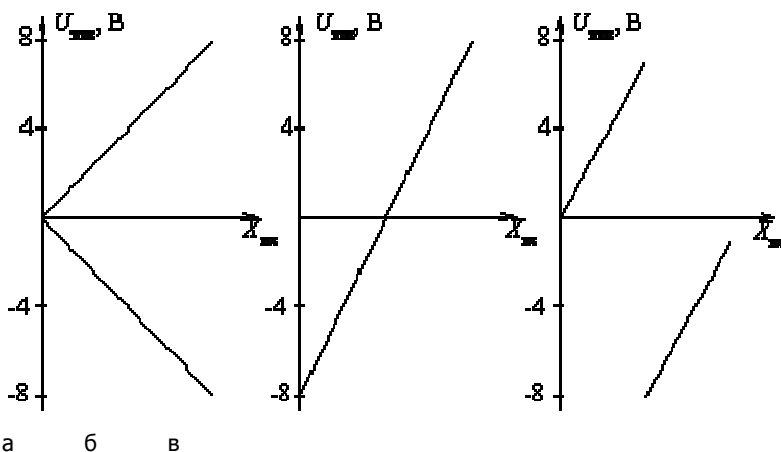


Рис. 12.2. Вихідні напруги ЦАП для кодів: а – прямого; б – зміщеного; в – доповняльного

Для виключення комутуючих елементів із схеми ЦАП використовують зміщений код, що є найпростішим (рис. 12.2, б). У доповняльному коді (рис. 12.2, в) позитивні числа перетворюються так, як і в прямому коді, а негативні – двійковим доповненням відповідного позитивного числа (інверсія всіх розрядів з подальшим додаванням одиниці в молодший розряд).

### 12.4. Основні параметри і характеристики цифро-аналогових перетворювачів

Основними параметрами ЦАП є число розрядів вхідного цифрового коду, роздільна здатність, похибки перетворення, діапазон вихідних сигналів, динамічні параметри (табл. 12.1).

Таблиця 12.1

Тип мікросхеми	Розрядність, n	Максимальне відхилення $\Delta$ , %	Час установлення туст, мкс	Технологія
K427ПА4	16	0,0015	20	КМОН
K572ПА2	12	0,025	15	Біполярна
KM1118ПА1	8	0,75	0,02	Біполярна
K1108ПА3	8	0,5	0,05	Біполярна
KM1148ПА1	10	0,75	1	Біполярна

Число розрядів n вхідного коду для різних типів ЦАП дорівнює від восьми до вісімнадцяти. Число розрядів визначає максимальну кількість кодових комбінацій на вході ЦАП, що дорівнює  $2^n$ .

Діапазон зміни вихідної напруги (без урахування знака) визначається із співвідношення (5.2) при  $X_i = 1, i = 1, 2, \dots, n$ :

$U_{\text{вих max}} = U_{\text{оп}}(2^{-1} + 2^{-2} + \dots + 2^{-n}) = U_{\text{оп}}(1 - 2^{-n})$ ,

якщо  $n = 10, U_{\text{оп}} = 10 \text{ В}$ , то  $U_{\text{вих max}} = 10 \text{ В}$ .

Роздільна здатність h характеризується мінімальним квантом вихідної напруги, який відповідає зміні вхідного коду на одиницю молодшого розряду:  $h = U_{\text{оп}}/2^n \approx 10 \text{ мВ}$  для попереднього прикладу.

Абсолютна похибка перетворення  $\Delta A$  – відхилення вихідної напруги від розрахункової в кінцевій точці характеристики перетворення. Типова похибка ЦАП не перевищує  $\pm 1/2$  молодшого розряду.

Нелінійність  $\Delta l$  – максимальне відхилення реальної характеристики перетворення від теоретичної (прямой лінії, що з'єднує точку нуля і мінімального вихідного сигналу).

Диференціальна нелінійність  $\Delta d$  – максимальне відхилення різниці двох аналогових сигналів сусідніх кодів від значення молодшого розряду.

Параметри  $\Delta A, \Delta l, \Delta d$  виражаються в частках молодшого розряду або у відсотках від повної шкали вихідної напруги.

Час установлення туст – інтервал часу від подачі вхідного коду до моменту досягнення вихідним сигналом сталого значення із заданою похибкою (зазвичай  $\pm 1/2$  молодшого розряду). Цей час визначає загальну швидкодію ЦАП.

Залежно від значень параметрів виділяють прецизійні ( $\Delta l < 0,1\%$ ) і швидкодіючі (туст  $\approx 100 \text{ нс}$ ) ЦАП.

Розрізняються ЦАП структурою резистивної матриці (зважені резистори або  $R - 2R$ ), елементною базою, функціональною повнотою, узгоджувальними пристроями, числом джерел живлення та ін.

Реалізуються ЦАП у вигляді гібридних і напівпровідникових мікросхем. Більшість мікросхем ЦАП – напівпровідникові, функціонально закінчені. Ряд ЦАП використовують зовнішні джерела опорної напруги і вихідні операційні підсилювачі.

Живлення ЦАП може здійснюватися від одного або від кількох джерел, що визначається елементною базою.

Перспективами розвитку ЦАП є підвищення швидкодії та точності, зручність узгодження з мікропроцесорами, зниження споживаної потужності.

## 12.5. Загальна характеристика аналого-цифрових перетворювачів

Аналого-цифрові перетворювачі (АЦП) призначені для перетворення аналогової інформації (звичайно у вигляді напруги) у цифровий код. Застосовують АЦП у мікропроцесорних системах, у цифрових вимірювальних приладах. Області застосування їх багато в чому аналогічні ЦАП, оскільки вони часто використовуються спільно, наприклад, в автоматизованих системах керування (АСК) (рис. 12.3).

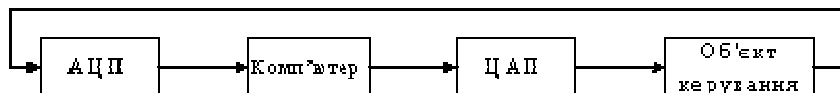


Рис. 12.3. Аналого-цифровий і цифро-аналоговий перетворювачі в контурі керування

Основними параметрами і характеристиками АЦП є:

число розрядів n вихідного коду;

роздільна здатність h – мінімальний квант вхідної напруги, за якої вихідний код змінюється на одиницю молодшого розряду;

нелінійність  $\Delta l$  – максимальне відхилення вихідного коду від розрахункового значення у всьому діапазоні шкали;

абсолютна похибка  $\Delta A$  – найбільше відхилення вихідного коду від розрахункового в кінцевій точці шкали;

час перетворення  $t_{\text{пр}}$  – інтервал від моменту початку перетворення до появи на виході сталого коду; часто замість  $t_{\text{пр}}$  швидкодія АЦП характеризується частотою перетворення;

діапазон і полярність вхідної напруги, число джерел живлення, струм споживання, можливість спільної роботи з мікропроцесорами.

У АЦП застосовуються такі методи перетворення:

послідовної лічби (з використанням ЦАП або з двотактним інтегруванням);

порозрядного кодування (послідовного двійкового наближення);

паралельної дії (зчитування);

паралельно-послідовні (комбіновані).

## 12.6. Основні параметри і характеристики аналого-цифрових перетворювачів

У АЦП використовують методи порозрядного кодування, послідовної лічби з двійковим інтегруванням і паралельного перетворення. Мікросхеми АЦП виконують за гібридною і напівпровідниковою технологією. У останні роки випускають, в основному, напівпровідникові АЦП. Основні параметри і характеристики деяких напівпровідникових АЦП наведені в табл. 12.2.

Деякі АЦП є функціонально закінченими, але більшість вимагають додаткових зовнішніх елементів: операційних підсилювачів, джерел опорної напруги, генераторів тактових імпульсів, резисторів і конденсаторів.

Таблиця 12.2

Тип	Розрядність, n	Максимальне	Час перетворення	Технологія	Примітка
-----	----------------	-------------	------------------	------------	----------

мікросхеми		відхилення, ?ЛД, %	ТП, мкс		
KP572PB3	8	± 0,75	7,5	КМОН	МК, СМ
K1107PB1	6	± 0,5	0,1	Біполярна	—
M1107PB6	10	± 1,5	0,06	Біполярна	ФЗ, СМ
K1108PB2	12	± 1	0,9	Біполярна	ФЗ

Примітка. У табл. 12.2: СМ – сумісність з мікропроцесорами; ФЗ – функціональна закінченість; МК – багатоканальність. Мікросхеми АЦП звичайно мають діапазон зміни вхідної напруги 0 – 10 В, а деякі АЦП допускають використання двополярного вхідного сигналу. Розрядність АЦП становить 6 – 12, причому ряд АЦП допускають нарощування розрядності. Швидкодія АЦП визначається, в основному, методом перетворення і елементною базою (ТТЛШ, ЕЗЛ, КМОН). Найбільшу швидкодію мають АЦП паралельної дії на базі ЕЗЛ-елементів (tпр ? 20 нс). Перетворювачі за рівними вихідних сигналів узгоджуються з ТТЛШ-, ЕЗЛ- і КМОН-мікросхемами. Більшість сучасних АЦП сумісні з мікропроцесорними пристроями. Вихідні ланцюги в таких АЦП мають три стійких стани (лог.0, лог.1 і Z).

Вихідним кодом АЦП найчастіше є двійковий. У АЦП з двійковим інтегруванням застосовується двійково-десятковий код для сполучення з індикаторами і вимірювальними приладами. Тут для представлення кожного десяткового знака використовуються чотири двійкових розряди. Використовуються також обернений і додатковий коди.

Деякі АЦП – це ВІС аналого-цифрової системи збирання даних, що включає в свій склад, крім перетворювача, багатоканальний мультиплексор, оперативний запам'ятовуючий пристрій, схеми буферів і керування.

Основними напрямками вдосконалення АЦП є:

підвищення швидкодії основних вузлів, особливо компараторів;

використання АЦП комбінованої дії;

підвищення точності перетворення, зокрема, збільшення розрядності до 16 і більше;

зниження споживаної потужності;

досягнення зручності та гнучкості застосування, особливо узгодження з мікропроцесорними пристроями.

## 12.7. Загальна характеристика перетворювачів кодів

Перетворювачем коду називається функціональний вузол комп'ютера, призначений для перетворення двійкового коду з однієї форми в іншу.

Для подання інформації використовують різноманітні двійкові та двійково-десяткові коди: прямий, обернений, доповняльний і їхні модифікації, циклічний з лишком три та інші. Існує велика кількість кодів, які забезпечують:

простоту виконання арифметико-логічних операцій;

зручність переведення чисел з десяткової системи в двійковий код;

надійність виконання заданих алгоритмів функціонування і ефективний контроль результатів обчислень;

зменшення апаратних витрат при побудові цифрових пристроїв.

Найбільш поширеними є прямий, обернений і доповняльний коди, які забезпечують представлення знака числа і заміну операції віднімання додаванням (табл. 12.3). До перетворювачів коду відносяться шифратори і дешифратори, однак за традицією ці функціональні вузли виділені в окремі самостійні класи.

Таблиця 12.3

Коди для додатних чисел				Коди для від'ємних чисел			
десятковий	прямий	обернений	доповняльний	десятковий	прямий	обернений	доповняльний
+0	0,000	0,000	0,000	–0	1,000	1,111	0,000
+1	0,001	0,001	0,001	–1	1,001	1,110	1,111
+2	0,010	0,010	0,010	–2	1,010	1,101	1,110
+3	0,011	0,011	0,011	–3	1,011	1,100	1,101
+4	0,100	0,100	0,100	–4	1,100	1,011	1,100
+5	0,101	0,101	0,101	–5	1,101	1,010	1,011
+6	0,110	0,110	0,110	–6	1,110	1,001	1,010
+7	0,111	0,111	0,111	–7	1,111	1,000	1,001

Прямий, обернений і доповняльний коди використовуються для записування знака числа, заміни операції віднімання чисел додаванням їхніх кодів, а також для визначення переповнення розрядної сітки. Для представлення знака числа у них відводиться знаковий розряд, який розташовується зліва від числа і відділяється комою. У знаковий розряд записується нуль – для позитивного числа і одиниця – для негативного.

## 12.8. Перетворювач прямого коду в обернений

У прямому двійковому коді ХПР = ХЗН Хn-1,..., Х1 один розряд, звичайно старший, відображає знак числа, інші – значення цифрових розрядів; при цьому для додатного числа ХЗН = 0, а для від'ємного ХЗН = 1. Обернений код додатного двійкового числа

збігається з прямим кодом, а для від'ємного числа цифрові розряди прямого коду інвертуються.

У процесі перетворення прямого коду в обернений значення знакового розряду  $X_{3H}$  використовується як керуючий сигнал, що забезпечує отримання такого виразу:

$$Y_i = \bar{X}_{3H} X_i \vee X_{3H} \bar{X}_i = X_{3H} \oplus X_i, \quad (12.3)$$

де  $Y_i$  – значення  $i$ -го розряду оберненого коду;  $X_i$  – значення  $i$ -го розряду додатного вхідного числа ( $X_{3H} = 0$ );  $\bar{X}_i$  – значення  $i$ -го розряду від'ємного вхідного числа ( $\bar{X}_{3H} = 1$ ).

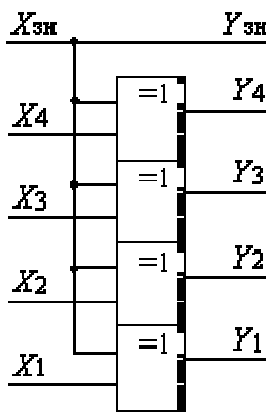


Рис. 12.4. Схема перетворювача прямого коду в обернений

Схема п'ятирозрядного перетворювача прямого коду в обернений, побудована на елементах «виключальне ЧИ» відповідно до виразу (4.21), показана на рис. 12.4.

## 12.9. Перетворювач прямого коду в доповняльний

Доповняльний код додатного двійкового числа збігається з його прямим і оберненим кодами. Доповняльний код від'ємного двійкового числа утворюється з його оберненого коду додаванням до молодшого розряду одиниці. Таким чином, операція перетворення прямого коду в доповняльний не є порозрядною і виконується значно складніше, ніж отримання оберненого коду. Відповідність між прямим і доповняльним кодами на прикладі чотирьох цифрових розрядів (беззнакових) наведена в табл. 12.4.

Знаковий розряд прямого коду використовується як керуючий сигнал: якщо  $X_{3H} = 0$ , то вихідний код повторює значення вхідного; при  $X_{3H} = 1$  реалізується перетворення згідно з табл. 12.4.

Таблиця 12.4

Прямий код				Доповняльний код				Прямий код				Доповняльний код			
X4	X3	X2	X1	Y4	Y3	Y2	Y1	X4	X3	X2	X1	Y4	Y3	Y2	Y1
0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
0	0	0	1	1	1	1	1	1	0	0	1	0	1	1	1
0	0	1	0	1	1	1	0	1	0	1	0	0	1	1	0
0	0	1	1	1	1	0	1	1	0	1	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	0	1	0	0
0	1	0	1	1	0	1	1	1	1	0	1	0	0	1	1
0	1	1	0	1	0	1	0	1	1	1	0	0	0	1	0
0	1	1	1	1	0	0	1	1	1	1	1	0	0	0	1

Карта Карно відповідно до табл. 12.4 для отримання мінімальних форм функцій перетворення прямого коду в доповняльний показана на рис. 12.5.

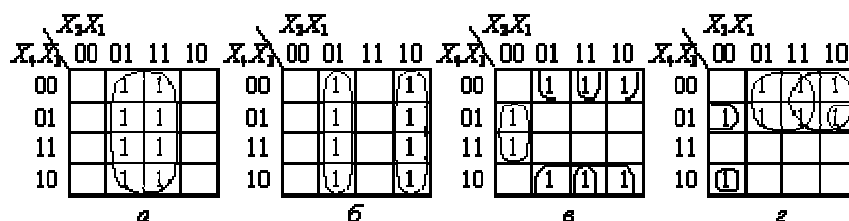


Рис. 12.5. Карта Карно для функцій перетворювача прямого коду в доповняльний: а –  $Y_1$ ; б –  $Y_2$ ; в –  $Y_3$ ; г –  $Y_4$

На основі карт Карно з врахуванням знакового розряду  $X_{3H}$  прямого коду для функцій  $Y_1, Y_2, Y_3, Y_4$ , що представляють виходи перетворювача, отримуємо:

$$Y_{\text{ЗН}} = X_{\text{ЗН}}; \quad Y_1 = X_1; \quad Y_2 = X_2 \oplus X_1 X_{\text{ЗН}};$$

$$Y_3 = X_3 \oplus (X_2 \vee X_1) X_{\text{ЗН}}; \quad Y_4 = X_4 \oplus (X_3 \vee X_2 \vee X_1) X_{\text{ЗН}}. \quad (12.4)$$

У загальному вигляді для  $Y_i$  справедливе рівняння:

$$Y_i = X_i \oplus (X_{i-1} \vee X_{i-2} \vee \dots \vee X_1) X_{\text{ЗН}}. \quad (12.5)$$

Схема перетворювача прямого коду в доповняльний на основі виразів (12.4) і (12.5) показана на рис. 12.6, а. Даний перетворювач характеризується високою швидкістю. Час встановлення вихідного коду визначається трьома затримками поширення сигналу, однак в міру зростання номера розряду лінійно зростає й необхідне число входів використовуваних елементів ЧИ.

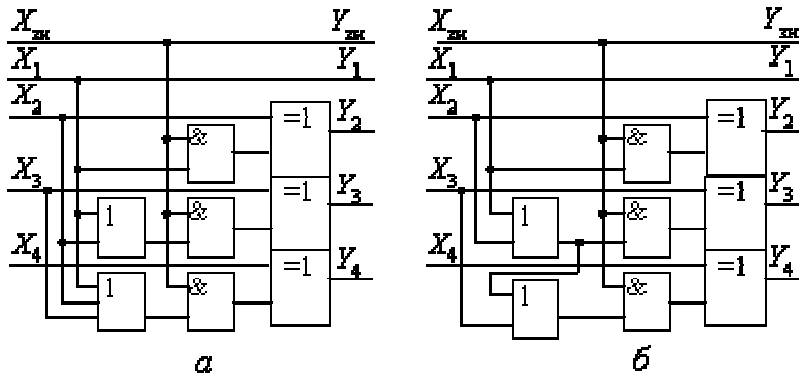


Рис. 12.6. Схеми перетворювачів прямого коду в доповняльний

Другий варіант схеми перетворювача (рис. 12.6, б) використовує тільки двовходові елементи ЧИ, при цьому диз'юнктивна сума змінних утворюється послідовним способом. У такій реалізації схема перетворювача спрощується, однак час встановлення вихідного коду істотно збільшується.

Практичне правило отримання доповняльного коду полягає в тому, що праворуч від першої одиниці (враховуючи і саму одиницю) в прямому коді числа значення розрядів – незмінні, а зліва від одиниці (крім знакового) – інвертуються. Наприклад, для прямого коду 10100100 доповняльним буде код 11011100.

Для перетворення в доповняльний код багаторозрядних двійкових чисел часто використовують переведення числа в обернений код і подальшого додавання одиниці до його молодшого розряду за допомогою суматора.

#### 12.10. Перетворювач двійкових чисел у код Грея

Код Грея утворений послідовністю двійкових чисел, в яких два будь-яких сусідніх числа відрізняються тільки одним розрядом (табл. 12.5). Перше і останнє числа вважаються сусідніми. Код Грея, який називають циклічним, відноситься до незначених двійкових кодів. Достоїнствами коду Грея є: зручність кодування кутових переміщень; простота кодувальної логіки; скорочення часу перетворення у зв'язку зі зміною значення тільки одного розряду; висока ефективність захисту від збоїв.

Недоліками коду Грея є ускладнення при виконанні арифметичних операцій і цифро-аналогових перетворень. Тому при необхідності код Грея перетворюють у двійковий код.

Таблиця 12.5

X4	X3	X2	X1	I4	I3	I2	I1	X4	X3	X2	X1	I4	I3	I2	I1
0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
0	0	0	1	0	0	0	1	1	0	0	1	1	1	0	1
0	0	1	0	0	0	1	1	1	0	1	0	1	1	1	1
0	0	1	1	0	0	1	0	0	0	1	1	1	1	1	0
0	1	0	0	0	1	1	0	1	1	0	0	1	0	1	0
0	1	0	1	0	1	1	1	1	1	0	1	1	0	1	1
0	1	1	0	0	1	0	1	1	1	1	0	1	0	0	1
0	1	1	1	0	1	0	0	1	1	1	1	1	0	0	0

За даними табл. 12.5 в клітинки карт Карно (рис. 12.7) внесено значення розрядів I1, I2, I3, I4 коду Грея.

За допомогою карт Карно отримуємо такі вирази для розрядів коду Грея:

$$I_1 = \bar{X}_2 X_1 \vee X_2 \bar{X}_1 = X_1 \oplus X_2; \quad I_2 = \bar{X}_3 X_2 \vee X_3 \bar{X}_2 = X_2 \oplus X_3;$$

$$I_3 = \bar{X}_4 X_3 \vee X_4 \bar{X}_3 = X_3 \oplus X_4; \quad I_4 = X_4. \quad (12.6)$$

Схема перетворювача прямого коду в код Грея на основі співвідношень (12.6) показана на рис. 12.8.



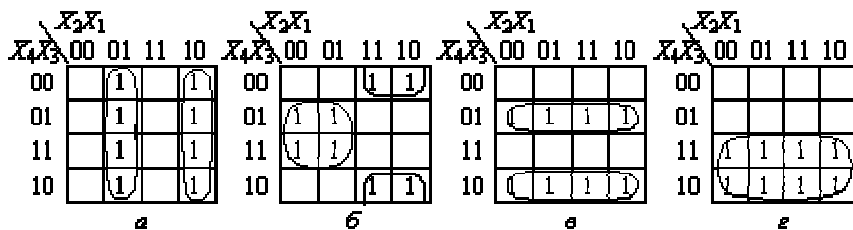


Рис. 12.7. Карта Карно для коду Грея: а – I1, б – I2, в – I3, г – I4

За аналогічною методикою, використовуючи табл. 12.5 і нові заповнення карт Карно, отримуємо обернене перетворення коду Грея в прямий код:

$$X_1 = I_1 \oplus I_2 \oplus I_3 \oplus I_4; \quad X_2 = I_2 \oplus I_3 \oplus I_4; \quad X_3 = I_3 \oplus I_4; \quad X_4 = I_4. \quad (12.7)$$

Схема перетворення коду Грея в прямий код на основі співвідношень (12.7) показана на рис. 12.9.

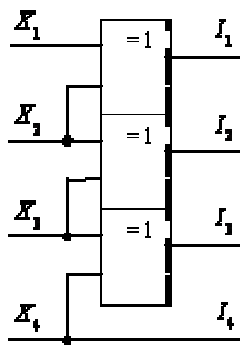


Рис.12.8. Перетворювач прямого коду в код Грея

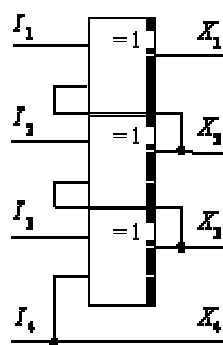


Рис.12.9. Перетворювач коду Грея в прямий код

#### 12.11. Перетворювач двійково-десяткових чисел в код семисегментного індикатора

Візуальне відображення двійково-десяткових чисел часто виконується за допомогою семисегментних індикаторів на основі електролюмінісcentних приладів, рідких кристалів або світлодіодних матриць. Кількість семисегментних індикаторів визначається розрядністю чисел, що відображаються на світловому табло – звичайно – шість і більше десяткових цифр.

Десятковий код відображуваної цифри, що виводиться з комп'ютера, поступає на вхід двійково-десятьового перетворювача, виходи якого а, б, с, ..., г підключаються до відповідних сегментів індикатора (рис. 12.10, а).

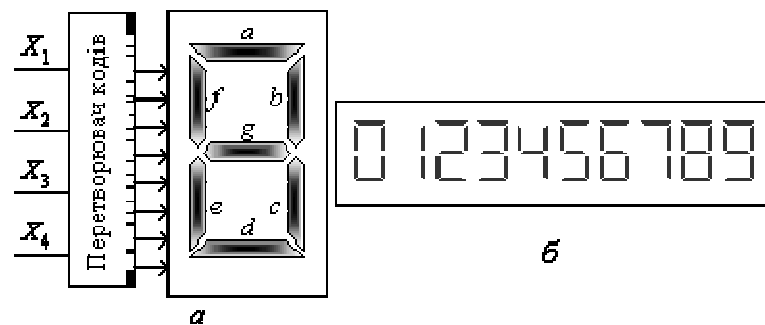


Рис. 12.10. Підключення перетворювача до індикатора (а) і відображення цифр (б)

Одиничне значення вихідного сигналу перетворювача викликає світіння сегмента, підключеного до цього виходу. Комбінації одиничних сигналів на виходах перетворювача утворюють зображення десяткової цифри в своєму розряді (рис. 12.10, б).

Відповідність між двійково-десятьовим числом і необхідними для відображення десяткової цифри наборами сегментів наведена в табл. 12.6.

Таблиця 12.6

X4	X3	X2	X1	a	b	c	d	e	f	g	X4	X3	X2	X1	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0	0	1	0	1	1	0	1	1	0	1	1
0	0	0	1	0	1	1	0	0	0	0	0	1	1	0	1	0	1	1	1	1	1
0	0	1	0	1	1	0	1	1	0	1	0	1	1	1	1	0	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0	1	1	0	0	0	1	1	1	1	1	1	1
0	1	0	0	0	1	1	0	0	1	1	1	0	0	1	1	1	1	0	1	1	1

$$g = \overline{X}_1 X_2 \vee X_2 \overline{X}_3 \vee \overline{X}_2 X_3 \vee X_4.$$

Таблица 12.7

Код Д				Код «9-Д»				Код «10-Д»				Код «Д+3»				Код «5421»			
X4	X3	X2	X1	Y4	Y3	Y2	Y1	F4	F3	F2	F1	Z4	Z3	Z2	Z1	E4	E3	E2	E1
0	0	0	0	1	0	0	1	1	0	1	0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0	1	0	0	1	0	1	0	0	0	0	0	1
0	0	1	0	0	1	1	1	1	0	0	0	0	1	0	1	0	0	1	0
0	0	1	1	0	1	1	0	0	1	1	1	0	1	1	0	0	0	1	1
0	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	0	1	0	0
0	1	0	1	0	1	0	0	0	1	0	1	1	0	0	0	0	1	0	1
0	1	1	0	0	0	1	1	0	1	0	0	1	0	0	1	0	1	1	0
0	1	1	1	0	0	1	0	0	0	1	1	1	0	1	0	0	1	1	1
1	0	0	0	0	0	0	1	0	0	1	0	1	0	1	1	1	0	0	0

1	0	0	1	0	0	0	0	0	0	0	1	1	1	0	0	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

На основі карт Карно отримуємо систему рівнянь для розрядів Y1 – Y4 оберненого Д-коду:

$$Y_1 = \bar{X}_1; \quad Y_2 = X_2; \quad Y_3 = X_2 \oplus X_3; \quad Y_4 = \bar{X}_4 \bar{X}_3 \bar{X}_2 = \bar{X}_4 \vee X_3 \vee \bar{X}_2. \quad (12.8)$$

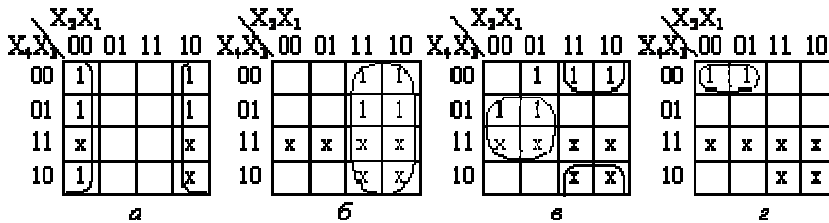


Рис. 12.12. Карти Карно для отримання оберненого Д-коду; знаком «х» помічені невизначені набори: а – Y1, б – Y2, в – Y3, г – Y4  
Із виразів (12.7) випливає, що значення другого розряду тетради прямого і оберненого Д-кодів збігаються. Схема перетворювача прямого Д-коду в обернений на основі співвідношень (12.8) показана на рис. 12.13, а. Можливий і інший спосіб побудови перетворювача Д-коду в обернений: спочатку інвертуються цифри всіх тетрад (виходить код з лишком шість) і потім відбувається віднімання з кожної тетради числа мінус 0110, що еквівалентне додаванню в доповняльному коді плюс 1010 (без урахування перенесень між тетрадами). Наприклад, A = -148, прямий Д-код  $\bar{A}_{10P} = 1\ 0001\ 0100\ 1000$ ; після інвертування маємо: АПР = 1 1110 1011 0111; після додавання плюс 1010 одержуємо значення оберненого Д-коду: АОБ = 1 1000 0101 0001.

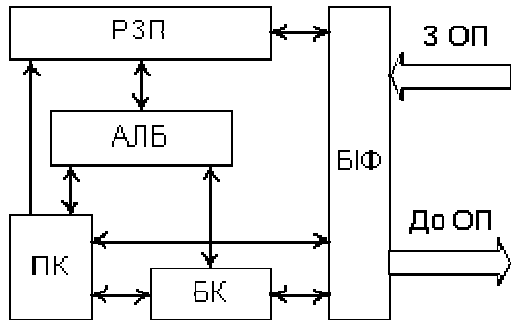


Рис. 13.1. Структура процесора

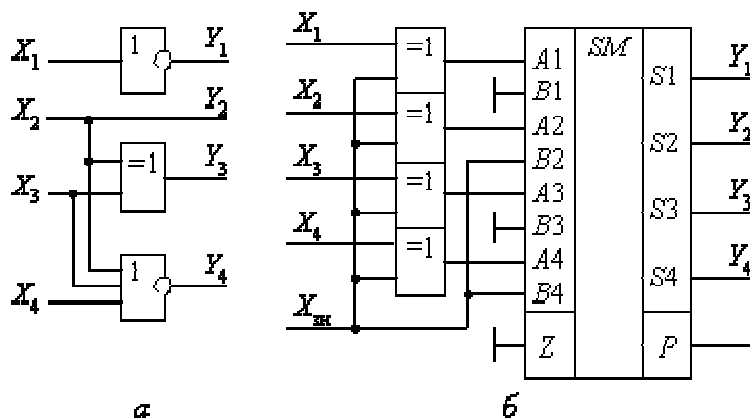


Рис. 12.13. Схеми перетворювачів Д-коду в обернений: а – на основі логічних рівнянь; б – з використанням елемента «виключальне ЧИ»

Схема перетворювача на основі інвертування і корекції тетрад показана на рис. 12.13, б. В даній схемі інвертування значень розрядів тетрад здійснюється логічними елементами «виключальне ЧИ», а корекція реалізується типовою мікросхемою чотирирозрядного комбінаційного суматора. Доповняльний Д-код тетради виходить з оберненого додаванням одиниці до молодшого розряду.

## Мікропроцесори. Великі інтегральні схеми з програмовними структурами.

### 13.1. Процесори

Універсальні комп'ютери поділяються на три функціонально зв'язані апаратні частини: процесор, пам'ять і периферійні пристрої.

Процесор – це основна функціональна частина комп'ютера, яка інтерпретує й виконує команди, тобто безпосередньо реалізує програмно-керований процес обробки даних. Процесор, який виконує в обчислювальній системі основні функції, називають центральним (ЦП). Спеціалізований процесор, призначений для керування зовнішніми пристроями (накопичувачами, дисплеями, принтерами та ін.) називають контролером.

Процесор характеризується архітектурою, до якої відносять: список арифметико-логічних операцій (система команд);

типи і формати команд і даних;  
організацію адресного простору пам'яті і периферійних пристроїв;  
способи адресації команд і даних;  
функції складових частин і структуру зв'язків з іншими пристроями машин та режими роботи.  
Процесор складається з пристроєм керування, арифметико-логічного пристрою та блоку інтерфейсу (БІФ) для з'єднання із зовнішнім середовищем – пам'яттю, периферійними пристроями (рис. 13.1).  
Оброблення даних здійснюється в АЛП, який містить арифметико-логічний блок (АЛБ), блок РЗП, блок контролю БК і місцевий блок керування при децентралізованому керуванні.  
Арифметико-логічний блок має універсальний двійковий комбінаційний суматор, двійково-десятковий суматор або схему десятикової корекції, регістри для тимчасового зберігання двох операндів і результату операцій та регістр прапорців. Для підвищення продуктивності в АЛП можуть включати спеціалізовані вузли-зсувачі, помножувачі, схеми прискореного переносу та інші пристрої. Ряд процесорів мають по два АЛП. Розрядність АЛП визначає розрядність всього процесора.  
В РЗП зберігаються початкові дані, проміжні та кінцеві результати, адреси даних, константи, які необхідні в процесі виконання команди. Всі операції в АЛП реалізуються як просторово-часові послідовності мікрооперацій над двійковими словами, кожна з яких є сукупністю булевих операцій над бітами слів. В АЛП реалізуються такі типові мікрооперації:  
передачі слів між регістрами та регістрами і пам'яттю;  
додавання двох слів, декремент (мінус 1) або інкремент (плюс 1) слова;  
арифметичні, логічні та циклічні зсуви вправо чи вліво;  
порозрядні логічні операції ЧИ, І, виключальне ЧИ та порівняння операндів;  
перетворення кодів слів – інверсія, доповнення, розширення та ін.  
Пристрій керування ПК керує процесом оброблення даних, забезпечує основні режими роботи (початкових установлень, очікування, переривання, прямого доступу до пам'яті, діагностики і контролю) та взаємодію всіх пристроїв комп'ютера. Для виконання цих функцій ПК має в своєму складі регістр і дешифратор команд, програмний лічильник для задання адреси наступної команди, блок керування та схеми синхронізації, діагностики й контролю.  
До складу процесора можуть входити спеціальні системні засоби (служба часу, засоби міжпроцесорного зв'язку, пульт керування та ін.).  
Пристрій керування послідовно зчитує код команди з пам'яті і розміщує його в регістр команд (інструкцій). Блок керування дешифрує команду і формує послідовності керуючих сигналів. Для виконання однієї мікрооперації в АЛП необхідний один керуючий сигнал. В одному машинному такті реалізується сукупність мікрооперацій – мікрокоманда. Множина мікрокоманд створює мікропрограму команди. Кожна команда має свою мікропрограму, час виконання якої називається командним циклом.  
Розрізняють апаратні, мікропрограмні та комбіновані блоки керування. Апаратні блоки керування побудовані на основі схемної логіки, а мікропрограмні – програмованої логіки (мають пам'ять мікропрограм). Комбіновані блоки керування використовують обидва способи їхньої реалізації.  
При централізованому керуванні один ПК керує процесом оброблення команд і даних у всій машині. При децентралізованому керуванні ПК формує основні керуючі сигнали, а опрацюванням даних керує місцевий блок керування, розміщений в АЛП.  
Зазначимо, що всі команди в комп'ютері реалізуються на основі принципу мікропрограмного керування, тобто виконання мікропрограм. В той же час ПК з програмовою логікою також називають мікропрограмними. Методологічно ці поняття слід розрізняти.

### 13.2. Мікропроцесори

Мікропроцесор являє собою процесор, складові частини якого мініатюризовані та розміщені в одній або декількох мікросхемах. Історія мікропроцесорів почалася в 1971 р., коли фірма Intel розробила перший у світі чотирирозрядний процесор 4004, а через рік – восьмирозрядний 8008. На основі мікропроцесорів створено нову елементну базу – мікропроцесорні засоби, на яких будують сучасні комп'ютери. У наш час терміни "процесор" і "мікропроцесор" сприймають як синоніми.  
Мікросхема, яка виконує функції мікропроцесора або його частини, називається мікропроцесорною. Сукупність мікропроцесорних та інших мікросхем, які сумісні за конструктивно-технологічним виконанням і призначені для спільного використання, називається мікропроцесорним комплектом (МПК).  
До характеристик мікропроцесорних ІМС відносяться: розміри кристала і кількість транзисторів у ньому, тип корпусу і кількість виводів.  
Найбільш важливими статичними і динамічними електричними параметрами мікропроцесорів як мікроелектронних виробів є:  
кількість джерел живлення та їхня напруга;  
струм і потужність споживання;  
кількість серій синхроімпульсів, їхня частота і амплітуда; рівні логічних сигналів;  
вхідна і вихідна ємності, навантажувальна здатність;  
час затримки розповсюдження сигналів, число операцій в секунду над операндами, які зберігаються в регістрах-акумуляторах.  
Залежно від умов експлуатації до мікропроцесорних ІМС можуть пред'являтися спеціальні вимоги щодо додержання:  
температурного діапазону роботи і режиму зберігання;  
стійкості до радіаційних і електромагнітних впливів, вібрацій та ударів;  
маси і габаритів;  
інтенсивності відмов, напрацювання на відмову, надійності функціонування.  
Мікропроцесор, призначений для використання функціонально повного набору операцій, є універсальним, інакше – спеціалізованим. За конструкцією розрізняють такі мікропроцесори:  
однокристалні – виконані у вигляді єдиної мікросхеми (рис.10.2, а);  
багатокристалні (модульні) – побудовані з мікросхем, кожна з яких виконує функції пристрою комп'ютера (рис.10.2, б);  
розрядно-нарощувальні (секційні); розрядність їхня може збільшуватися за рахунок декількох мікропроцесорних секцій, об'єднаних спільними шинами (рис.10.2, в).

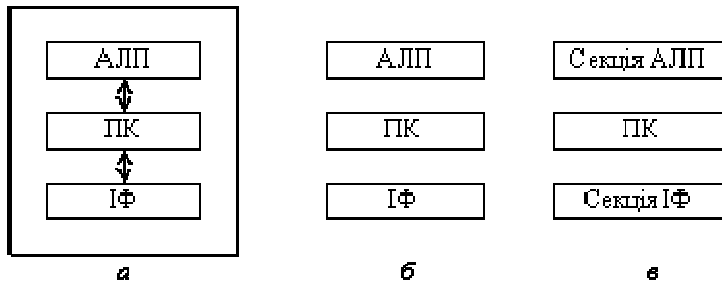


Рис. 10.2. Конструкції мікропроцесорів: а – однокристальні; б – багатокристальні; в – секційні

Усі мікропроцесори мають засоби сполучення із зовнішніми пристроями – інтерфейси (ІФ).

Для виробництва мікропроцесорів використовують усі види схемотехніки (технології): ТТЛШ, ЕЗЛ, І2Л, n-МОН, р-МОН, КМОН та ін.

Залежно від режиму роботи розрізняють такі процесори:

однопрограми (виконують одну програму) і багатопрограми (мають засоби для одночасного виконання кількох програм);

мультипроцесори (системи, в яких одночасно можуть бути активними декілька процесорів);

конвеєрні (команди виконуються послідовно рядом пристроїв, причому різні пристрої можуть одночасно обробляти відповідні частини декількох команд);

матричні (мають спеціальну архітектуру, розраховану на оброблення числових масивів);

співпроцесори (арифметичні розширювачі) – призначені для розширення списку команд ЦП; самостійно не використовуються;

периферійні – виконують функції введення-виведення інформації (асоціативні процесори, в яких характер обробки даних визначається змістом самих даних).

За видом оброблюваної інформації розрізняють цифрові (звичайні) та аналогові мікропроцесори. В аналогових мікропроцесорах на вході використовують АЦП для перетворення аналогових величин в цифровий код, а на виході – схеми ЦАП, які перетворюють цифрові дані в аналогові.

### 13.3. Мікропроцесорні комплекти

Для побудови сучасних комп'ютерів використовують мікропроцесорні засоби – комплекси апаратно і програмно сумісних ІМС різного ступеню складності (CIC, BIC, HBIC та ультра-BIC). Основою мікропроцесорних засобів є мікропроцесорні комплекти МПК і базові кристали, BIC пам'яті (рис. 13.3).

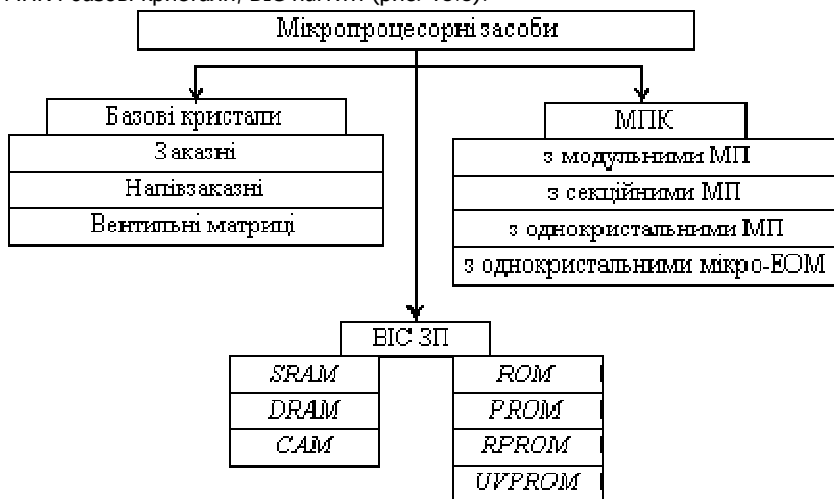


Рис.13.3. Склад мікропроцесорних засобів

Залежно від виду і типу базової програмовної BIC виділяють такі МПК:

з одним або кількома однокристальними мікропроцесорами: серії KP580, KP581, K1810, K1518, K1821, KP1828, KM1831, K1838;

з багатокристальними (модульними) мікропроцесорами: K581, K588;

із секціонованими (розрядно-модульними) мікропроцесорами: K583, K1800, K1802, K1804, K1822;

з однокристальними мікро-ЕОМ: K1813, K1816, K1820, K1827.

Однокристальні мікропроцесори мають фіксовану структуру і систему команд; призначені для побудови мікро- та міні-комп'ютерів широкого застосування. Однокристальні мікро-ЕОМ вбудовуються в технічні системи, в яких вони виконують функції контролю, керування і діагностики.

Модульні мікропроцесори орієнтовані на реалізацію конкретного класу алгоритмів (оброблення цифрових сигналів, графічні перетворення та ін.). Секціоновані мікропроцесори забезпечують побудову високопродуктивних універсальних комп'ютерів методом нарощування необхідного числа секцій.

До складу МПК входить широкий спектр спеціалізованих мікропроцесорних ІМС, в тому числі:

процесорні елементи (секції АЛП); блоки регістрів загального і спеціального призначення; багаторежимні буферні регістри

(порти, фіксатори); схеми прискорених перенесень і розширювачі обчислювальних функцій;

схеми синхронізації та генератори тактових імпульсів; блоки керування обміном з пам'яттю та формування адрес мікрокоманд;

контролери: клавіатури, відеотерміналів, накопичувачів на магнітних дисках і стрічках, введення-виведення даних, прямого доступу до пам'яті і переривань;

контролери шин і протоколів, шинні формувачі;

програмовні таймери і схеми арбітражу.

За характером роботи в часі мікропроцесори поділяються на синхронні (час виконання команд в них постійний) і асинхронні, де кожна команда реалізується за час її фактичного виконання.

Мікропроцесори мають різні пристрої керування із схемною або програмною логікою та комбіновані.

Поява мікропроцесорів – це еволюційний процес у технології виробництва мікросхем, а для обчислювальної техніки – перехід у

новий якісний стан.

До техніко-економічних переваг мікропроцесорів відносять:

високу швидкодію й надійність, малу потужність споживання, малі розміри і масу, низьку вартість; простоту технічного обслуговування, складання і розширення мікропроцесорних систем; можливість побудови потужних мікропроцесорних комплексів з паралельною обробкою інформації; реалізацію розподілених систем контролю, діагностики і керування.

Достоїнства мікропроцесорів забезпечили масове застосування їх у науці, техніці та нетрадиційних областях господарства.

Конструктивно-технологічний розвиток мікропроцесорів відбувається в таких напрямках.

Використовують нові технології, наприклад, БІКМОН, в яких комбінуються біполярні транзистори (для збільшення швидкості) та КМОН структури (для зменшення споживаної потужності та підвищення щільності компоновки). Якщо в перших мікропроцесорах відстань між сусідніми лініями дорівнювала 10 мкм, то в останніх виробках вона дорівнює 0,18 мкм.

Зростає рівень інтеграції: від 2800 транзисторів в чипах перших мікропроцесорів до 10–12 млн і більше в останніх виробках типу Itanium. За законом Мура характеристики мікросхем мають поліпшуватися в два рази кожні 18 місяців при збереженні вартості. За прогнозом експертів у 2012 р. на чипі буде розташовуватись 1,4 млрд транзисторів, які мають працювати на частоті 2700 МГц.

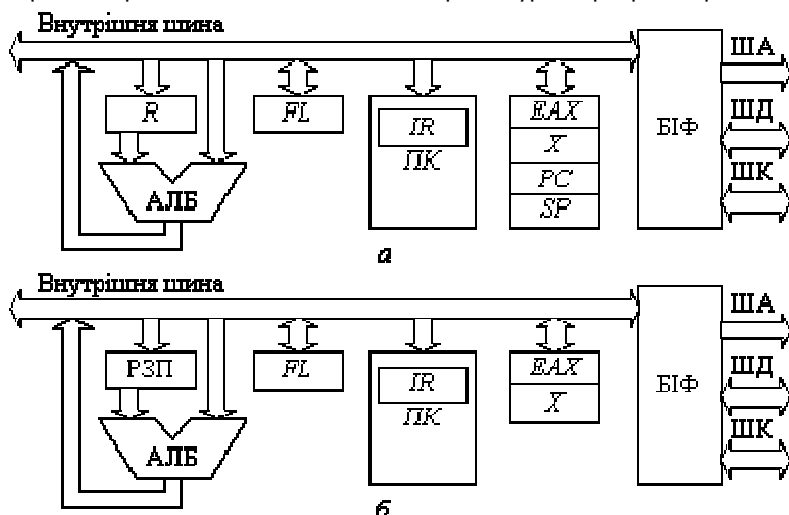
Збільшується розрядність оброблюваних даних: від чотирьох – у перших мікропроцесорах до 64 – в Itanium (зросла в 16 разів).

Прискорюється зміна поколінь мікропроцесорів. Якщо до 1994 р. нові покоління з'являлися кожні три роки, то тепер цей цикл зменшився до двох років. Вартість сучасного заводу, який випускає чипи з розмірами транзисторів 0,25 мкм становить \$ 2 млрд, а через 5 – 10 років такий комплекс буде коштувати \$ 4 млрд.

### 13.4. Архітектура мікропроцесорів

Архітектуру мікропроцесора характеризують: список команд та їхні формати; способи адресації; розрядність і ємність адресованої пам'яті; структура регістрів та їхні функції та ін.

Історично першими склалися такі основні архітектури мікропроцесорів: з акумулятором, з РЗП, зі стековою організацією та комбіновані (рис.10.4).



Структура всіх мікропроцесорів містить такі однакові пристрої, вузли і блоки, об'єднані спільною внутрішньою шиною даних: АЛБ; ПК; IR – регістр команд (інструкцій); FL – регістр ознак (прапорців); EAX – регістр виконавчої адреси; БІФ – блок інтерфейсу з вихідними шинами адреси, даних і керування.

В архітектурі мікропроцесора з акумуляторами (рис. 10.4, а) додатково використовують індексний регістр X і покажчик стека SP; регістр R виконує функції акумулятора А. При виконанні арифметико-логічних операцій перший операнд попередньо розміщується в акумуляторі, а другий – з ОП безпосередньо поступає на вхід АЛБ.

Рис. 13.4. Структура мікропроцесорів: а – з акумулятором і стеком; б – з РЗП

Результат операції розміщується в акумуляторі, а в регістр FL автоматично записуються ознаки операції.

В архітектурі мікропроцесора з РЗП (рис. 13.4, б) звичайно використовують від восьми до 16 регістрів, кожний з яких може виконувати функції акумулятора. Збільшення числа РЗП значно зменшує кількість звернень до ОП, що підвищує продуктивність комп'ютера. Але збільшення числа регістрів призводить до втрат машинного часу у випадках переривання програм: при цьому необхідно зберегти зміст РЗП в пам'яті (звичайно у стеку), а потім відновити його після обробки переривання. Тому при виборі кількості РЗП враховують цей фактор.

У мікропроцесорі зі стековою архітектурою (рис. 13.4, а) відсутні акумулятор і РЗП, а R виконує функції регістра тимчасового зберігання даних ОП на час виконання операції. Читання даних і їхнє записування у стекову пам'ять здійснюють за допомогою покажчика стека SP. Усі операції з даними виконують відповідно до польського запису: операнди розміщують у стеку в послідовності виконання над ними дій. У стек послідовно завантажують числа а, b і с (рис.13.5).

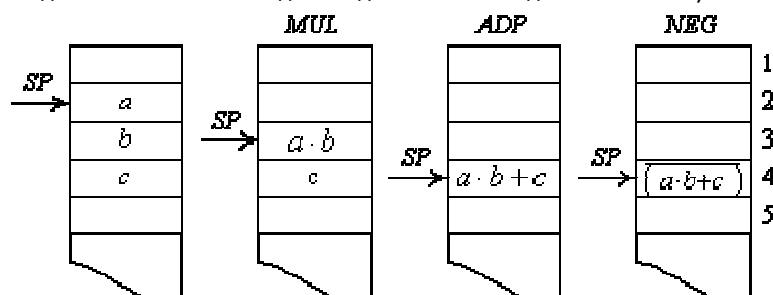
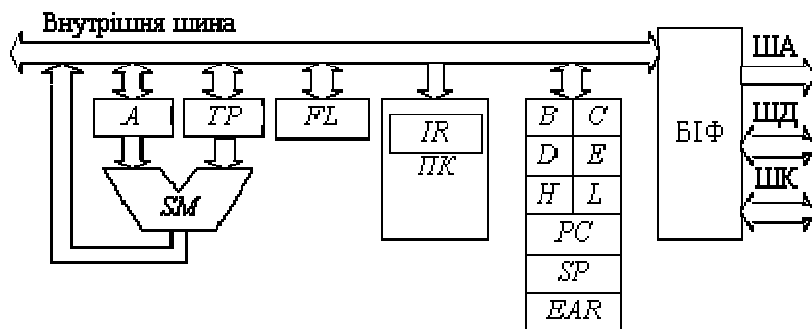


Рис. 13.5. Ілюстрація роботи стека

За командою множення MUL одержуємо добуток  $a \cdot b$ , який записується в адресі b. За командою додавання ADD маємо  $a \cdot b + c$ , який записується в адресі c. Після команди інвертування NEG в комірці за адресою c записується обернений код результату  $a \cdot b + c$ .

В мікропроцесорах з комбінованою архітектурою об'єднують значною мірою властивості архітектур з акумулятором, РЗП і стеком. До них відносять однокристальний мікропроцесор 8080 (1974 р., фірма Intel, аналог у нашій країні – КР580ВМ80А), більш досконалу його модель 8085А (аналог – КР1821ВМ85А), а також модель Z80 фірми Zilog (рис. 13.6).



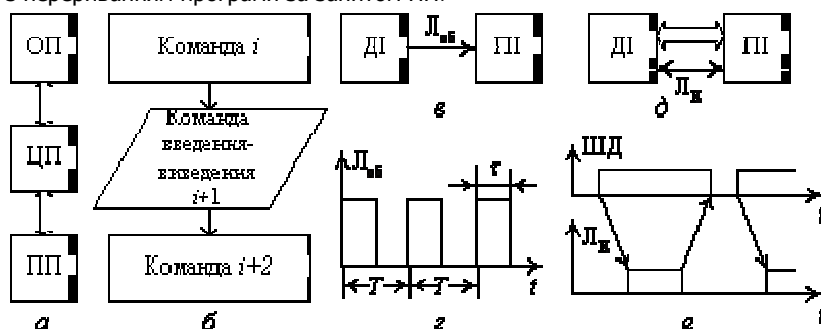
Комбінована структура містить:

блок РЗП, регістри В, С, D, Е, H, L, програмний лічильник РС, показчик стека SP, регістр виконавчої адреси ЕАР;

пристрій керування ПК з регістром команд IR.

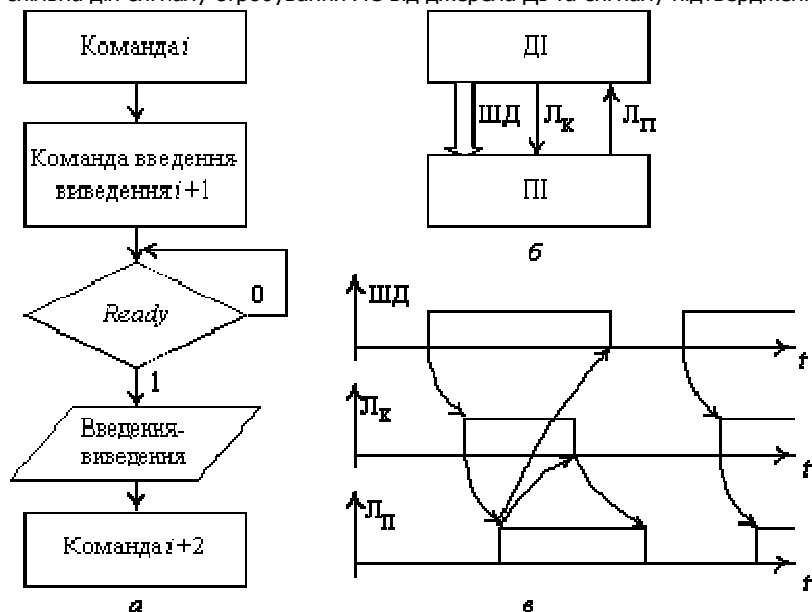
### 13.5. Програмно керований обмін інформацією

і тих ядер нашій і периферії реалізуються такі види програмно керуваного обміну інформацією: синхронний і асинхронний обміни в послідовних і паралельних двійкових кодах, які виконуються за схемою, зображеною на рис.13.7, а;



При синхронній передачі джерело інформації ДІ завжди готове до обміну відповідно до алгоритму (рис.13.7, б). Джерело інформації виставляє і утримує значення даних на лінії послідовного обміну Лоб протягом часу  $t$ , який складається з тривалості затримки розповсюдження сигналу на лінії, його розпізнавання і фіксації в реєстрі приймача ПІ (рис.13.7, в,г).

Асинхронний обмін виконується при готовності (Ready) зовнішнього пристрою до обміну даними відповідно до алгоритму (рис.13.8, а). При асинхронній передачі паралельного коду по ШД використовують метод квітування, в якому поєднуються спільна дія сигналу стробування ЛС від джерела ДІ та сигналу підтвердження приймання ЛП від приймача ПІ (рис. 13.8, б,в).



Основним недоліком синхронного і асинхронного обміну є значне завантаження процесора операціями введення–виведення, що призводить до суттєвого зменшення продуктивності комп'ютера. Тому синхронний і асинхронний обміни використовують при передачі одиночних байтів чи слів. Обмін в режимі переривання здійснюють апаратно за ініціативою зовнішнього пристрою чи програмно – командою переривання INT. Процесор, одержавши апаратний запит на переривання, закінчує поточну команду, пересилає в ОП зміст своїх регістрів і переходить на підпрограму обслуговування переривання. Після її закінчення процесор відновлює зміст своїх регістрів і продовжує виконання перерваної програми (рис. 13.9, а).

Апаратно режим переривання забезпечується контролером переривань (КПР), до якого підключаються ПП. Вихід INT контролера подається на відповідний вхід ЦП, а на ШД пересилається початкова адреса підпрограми обслуговування (рис. 13.9, б). При програмному перериванні адреса підпрограми подається в самій команді переривання INT.

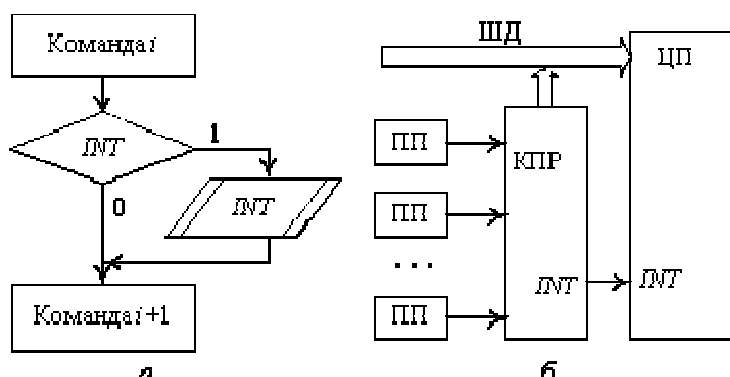


Рис.13.9. Обмін за перериванням: а – алгоритм; б – схема підключення контролера КПР

Прямий доступ до пам'яті (ПДП) використовують для швидкого обміну масивами інформації між основною пам'яттю і периферією. При цьому процесор (мікропроцесор) звільняється від безпосереднього керування операціями введення–виведення. ПДП реалізується відповідно до алгоритму (рис.13.10, а).

В міні- і мікрокомп'ютерах прямим доступом керує контролер прямого доступу до пам'яті (КПДП). Перед початком обміну процесор пересилає в КПДП таку інформацію (програмування контролера):  
початкову адресу області пам'яті, яка бере участь у обміні;  
напрямок операції обміну – введення чи виведення;  
кількість байтів, які підлягають передачі.

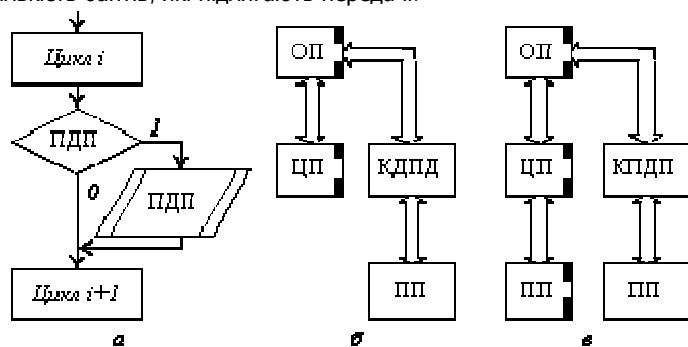


Рис.13.10. Режим ПДП: а – алгоритм; б, в – схеми підключення контролерів КПДП

Особливість режиму ПДП – обмін даними може бути між машинними циклами в команді ("Захват циклу"), а також після закінчення команди.

Контролер ПДП керує обміном даними між ОП і ПП без участі процесора (рис. 13.10, б). При необхідності в комп'ютері використовують програмно-керований обмін окремими байтами (він не вимагає програмування КПДП) і обмін масивами у режимі ПДП (рис. 13.10, в).

В універсальних комп'ютерах обмін інформацією між ОП і ПП забезпечують спеціальні пристрої - КВВ або просто канали (рис. 13.11).

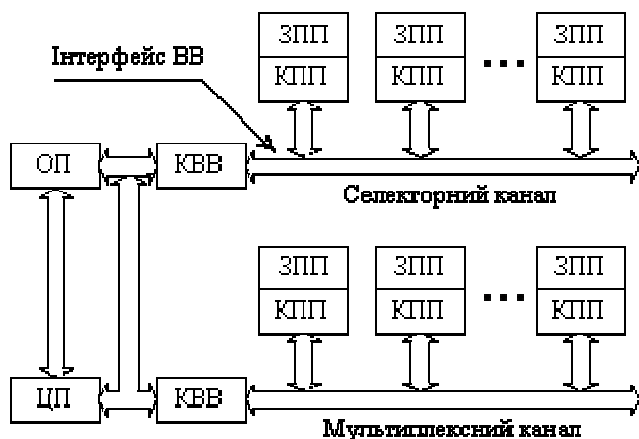


Рис.13.11. Структура комп'ютера

В каналах використовують два режими обміну інформацією: монопольний та розподілу в часі. Монопольний режим реалізують селекторним каналом, а розподілу в часі - мультиплексним каналом. Засоби каналу, які призначені для обслуговування одного



ПП, називаються підканалом.

Селекторні канали обслуговують швидкодіючі пристрої, в першу чергу ЗЗП (накопичувачі на дисках і магнітних стрічках). Селекторний канал має один підканал. Після встановлення зв'язку він монополярно обслуговує тільки один ПП: інші пристрої чекають закінчення даної програми обміну. У малих комп'ютерах використовують один селекторний канал, у великих - до шести. Мультиплексний канал (один в машині) паралельно обслуговує сотні повільно діючих ПП в режимі розподілу часу (клавіатура, принтери, перфатори та ін.).

## Пам'ять комп'ютерів. Висновки.

### 15.1. Функція пам'яті

Пам'яттю комп'ютера називається сукупність різних пристроїв, призначених для приймання, зберігання і видачі двійкової інформації. Окремий пристрій називається запам'ятовуючим (ЗП) або просто пам'яттю. Термін "запам'ятовуючий пристрій" вживають тоді, коли треба підкреслити принцип його побудови: на магнітному осерді, напівпровідниках та ін.

Термін "пам'ять" застосовують, коли вказують на функцію, яку вона виконує: основну, постійну та ін.

Пам'ять комп'ютера функціонує під керуванням операційної системи, яка розміщує масиви інформації в пам'яті, забезпечує їхній захист від несанкціонованого доступу та виконує інші функції. Продуктивність і обчислювальні можливості комп'ютера значною мірою визначаються складом і характеристиками ЗП, які застосовуються.

Пам'ять сучасних комп'ютерів класифікують за функціональним призначенням, видом носія інформації, способом організації доступу до інформації.

За функціональним призначенням пам'ять комп'ютерів поділяється на дві основні групи: зовнішню і внутрішню.

Зовнішні ЗП призначені для тривалого зберігання великих масивів інформації з ємністю до гігабайта і більше та малою швидкодією. Зовнішня пам'ять містить в собі накопичувачі на магнітних стрічках, дисках, барабанах та оптичних дисках.

Внутрішні ЗП призначені для зберігання програм і даних, які виконуються в поточний момент часу. До внутрішньої пам'яті відносяться:

надоперативні (регістрові) ЗП, які використовують регістри загального призначення процесора; вони мають невелику інформаційну ємність і швидкодію роботи процесора;

кеш-пам'ять, яка служить для зберігання копій інформації, що використовуються в поточних операціях обміну. Висока швидкодія кеш-пам'яті підвищує продуктивність комп'ютера;

оперативні, які характеризуються високою швидкодією та інформаційною ємністю до сотень мегабайт; оперативна пам'ять комп'ютерів перших поколінь будувалася на магнітних осердях, тепер же ОП реалізується на напівпровідникових ВІС ЗП. У процесі роботи інформація із зовнішньої пам'яті при необхідності переписується в оперативний ЗП (ОЗП);

постійні, які будуються на напівпровідникових ВІС. У постійну пам'ять інформація записується заздалегідь і її можна тільки прочитувати. Оперативні й постійні ЗП утворюють основну пам'ять комп'ютера;

спеціалізовані види пам'яті – багатопортові, асоціативні, відеопам'ять та ін.

За фізичним принципом побудови пам'ять комп'ютера буває:

магнітна (на осерді та плівках, на циліндричних і плоских магнітних доменах);

ультразвукова (магнітострикційна, електрострикційна);

сегнетоелектрична і голографічна (лазерна), на основі надпровідності;

напівпровідникова на ВІС і НВІС, ультра-ВІС.

Напівпровідникові ВІС ЗП в свою чергу характеризуються:

технологією виготовлення: на біполярних транзисторах (ТТЛШ, ЕЗЛ, І2Л), на МОН-структурах (р-МОН, п-МОН, КМОН); серед новітніх розробок слід відмітити ЗП, де використані ПТШ на основі арсеніду галію;

способом зберігання інформації – статичні та динамічні (у статичних ЗП елементом пам'яті є тригер, а у динамічних елемент пам'яті будують на конденсаторі і МОН-транзисторах);

енергозалежністю: розрізняють енергозалежні ВІС ЗП, в яких при відключенні джерела живлення інформація, яка зберігається, руйнується (що справедливо в цей час для більшості напівпровідникових мікросхем пам'яті), і енергонезалежні (звичайно на сегнетоелектриках), в яких інформація зберігається;

структурною організацією ВІС ЗП, яка символічно подається у вигляді  $N \times m$ , де  $N$  – кількість адресних одиниць інформації, що зберігаються;  $m$  – розрядність (організацію у вигляді  $N \times 1$  називають однорозрядною, а  $N \times m$  – словниковою, при цьому  $m > 1$ ).

Елементний базис пам'яті сучасних комп'ютерів складають мікросхеми різного ступеня інтеграції. Основою будь-якого ЗП є елемент пам'яті (ЕП) статичного або динамічного типу, призначений для записування, зберігання і зчитування одного біта інформації – цифри 0 або 1. Сукупність ЕП, які утворюють п-розрядне слово, називають коміркою пам'яті (КП). Множина КП утворює запам'ятовуючий масив, який називається матрицею  $M$  елементів пам'яті.

### 15.2. Основні параметри пам'яті

Основними операціями в пам'яті є записування і зчитування певної одиниці інформації, наприклад, байта. Ці операції називаються також зверненням до пам'яті. Пам'ять характеризується інформаційною ємністю, фізичним об'ємом, питомою ємністю і вартістю, шириною вибірки, споживаною потужністю і швидкодією.

Інформаційна ємність  $E$  являє собою максимальний об'єм даних, який може одночасно зберігатися в пам'яті. Ємність виражається в бітах, байтах ( $8 \text{ біт} = 1 \text{ байт}$ ), кілобайтах ( $2^{10} \text{ байт} = 1 \text{ Кбайт}$ ), мегабайтах ( $2^{20} \text{ Кбайт} = 1 \text{ Мбайт}$ ) і гігабайтах ( $2^{30} \text{ Мбайт} = 1 \text{ Гбайт}$ ) (при цьому потрібно врахувати, що  $2^{10} = 1024$ ).

Питома ємність визначається відношенням інформаційної ємності ЗП до його інформаційної ємності. Питома вартість – це відношення вартості ЗП до його інформаційної ємності. Ширина вибірки подається числом розрядів, які записуються в ЗП або зчитуються з нього за одне звернення.

Споживану потужність задають або для усього ЗП, або на зберігання одного біта інформації. Основними вимогами до пам'яті є максимально велика інформаційна ємність, висока швидкодія (малий час звернення:  $t_{зв} < 10 \text{ нс}$ ), мінімальна споживана потужність (менша за  $1 \text{ мВт}$  на  $1 \text{ біт}$  інформації, яка зберігається).

У наш час жоден з видів ЗП не задовольняє цих вимог повною мірою. Тому в пам'яті використовуються різні види ЗП, які розрізняються принципами побудови і своїми характеристиками.

Швидкодія ЗП вимірюється часом записування і зчитування та тривалістю відповідних їм циклів.

Час записування  $t_{WR}$  – це інтервал між моментами появи керуючого сигналу записування і установленням КП в стан, який задають вхідні сигнали. Час зчитування – це інтервал між моментами появи керуючого сигналу читання  $t_{RD}$  і даних на виході пам'яті. Мінімально допустимий інтервал між послідовними читаннями  $t_{CYR}$  і записуваннями  $t_{CYW}$  створює відповідний цикл. Тривалість циклів може перевищувати час читання чи записування, оскільки після цих операцій необхідна додаткова затримка для встановлення початкового стану пам'яті. Як тривалість циклу звернення до пам'яті беруть величину  $t_{CY} = \max(t_{CYW}, t_{CYR})$ .

### 15.3. Вхідні та вихідні сигнали мікросхеми пам'яті

Мікросхеми ОП мають типові виводи, на яких діють визначені адресні, інформаційні та керуючі сигнали (рис. 15.1, а).

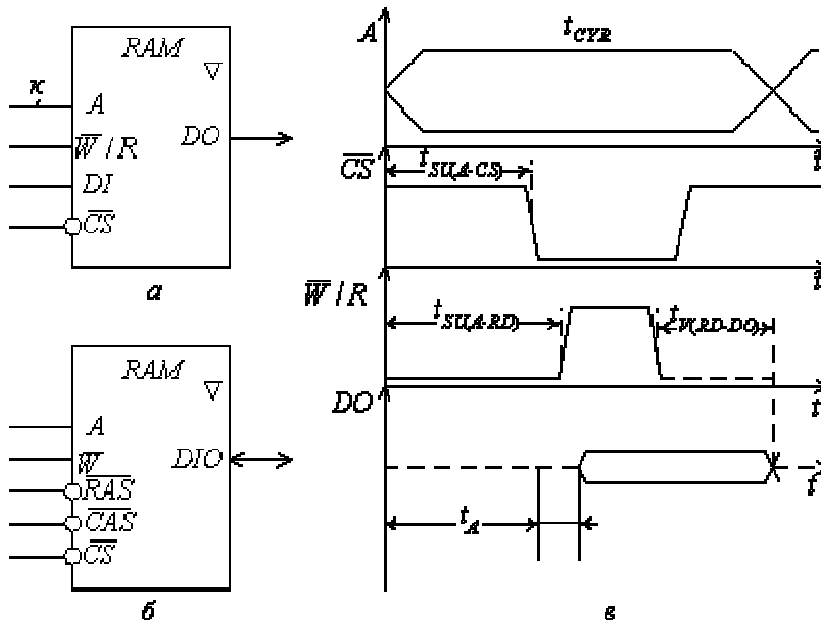


Рис. 15.1. Мікросхеми ОП: а, б – умовні графічні позначення; в – часові діаграми сигналів

Призначення виводів і сигналів на них такі:

$A$  (Address) – входи адреси, розрядність якої  $k$  визначається співвідношенням  $k = \log_2 N$ , де  $N = 2^k$  – максимально можливе число даних (біт, байт, слів), що зберігаються в пам'яті і адресуються як єдине ціле;

$DI$  (Data Input) – шина вхідних даних;

$DO$  (Data Out) – шина вихідних даних;

$\overline{W}/R$  (Write/Read) – сигнал записування даних при  $\overline{W}/R=0$  або зчитування при  $\overline{W}/R=1$ ;

$\overline{CS}$  (Chip Select) або  $\overline{CE}$  (Chip Enable) – сигнал дозволу при  $\overline{CS}(\overline{CE}) = 0$  чи заборони, якщо  $\overline{CS}(\overline{CE}) = 1$ , роботи даної мікросхеми.

Особливістю роботи динамічних ЗП є мультиплексування ША (рис. 7.1, б). Адреса, наприклад,  $A=A_{15}, A_{14}, \dots, A_0$  ділиться на старшу напівадресу  $A_x=A_{15}, A_{14}, \dots, A_8$  і молодшу  $A_y=A_7, A_6, \dots, A_0$ . Напівадреси подаються на одні й ті ж входи адреси мікросхеми пам'яті. Подавання напівадреси  $A_x$  супроводжується сигналом  $\overline{RAS}$  (Row Address Strobe), а напівадреси  $A_y$  – сигналом  $\overline{CAS}$  (Column Address Strobe). Такий спосіб адресації зменшує число виводів корпусу ІМС. Часто виводи  $DI$  і  $DO$  об'єднуються у спільний вивід  $DIO$ .

### 15.4. Часові характеристики мікросхем пам'яті

Вимоги до взаємного часового положення двох сигналів (А–В) задають такими параметрами:

часом попереднього установлення  $t_{SU}(A-B)$  сигналу А відносно сигналу В, тобто інтервалом між початками обох сигналів;

часом утримання  $t_H(A-B)$  – інтервалом часу між початком сигналу А і закінченням сигналу В;

часом зберігання  $t_V(A-B)$  – інтервалом між закінченнями сигналів А і В.

Тривалість сигналів позначається як  $t_W$  (Width – ширина).

Для ЗП характерна така послідовність сигналів у часі (рис. 15.1, в): спочатку адреса, потім вибірка мікросхеми  $\overline{CS}$ , потім строб записування–читання  $\overline{W}/R$ .

Індексом А (Access) позначають інтервали часу від появи керуючого сигналу до появи даних на виході (рис. 15.1, в).

### 15.5. Способи доступу до даних у напівпровідниковій пам'яті

У напівпровідникових ЗП виділяють адресні, послідовні й асоціативні способи доступу до даних (рис. 15.2).



Рис. 15.2. Класифікація напівпровідникових ЗП

При адресному доступі адресний код указує номер комірки пам'яті, з якою має проводитися обмін. Усі комірки в момент звернення рівнодоступні. До адресних ЗП відносяться:

RAM (Random Access Memory), українські синоніми: ОЗП (оперативний ЗП) або ЗПДВ (ЗП з довільною вибіркою);

ROM (Read Only Memory), український термін – ПЗП (постійні ЗП).

Оперативні ЗП зберігають дані, необхідні при виконанні поточної програми; вони можуть бути змінені в будь-який момент часу. Оперативні ЗП в більшості є енергозалежні.

У постійних ЗП вміст комірок або взагалі не змінюється, або змінюється рідко в спеціальних режимах.

Запам'ятовуючі пристрої RAM поділяються на статичні SRAM (Static RAM) і динамічні DRAM (Dynamic RAM). У статичних RAM елементами пам'яті є тригери. Вони зберігають свій стан, поки схема має напругу живлення і нові дані не записуються. У динамічних RAM дані зберігаються у вигляді зарядів конденсаторів, створюваних компонентами МОН-транзисторів. Саморозряд конденсаторів веде до руйнування даних, тому вони періодично (кожні 2–30 мс) мають регенеруватися. Але щільність упакування динамічних ЕП перевищує в 4–5 разів такий же показник для статичних RAM. Регенерація даних здійснюється за допомогою спеціальних контролерів. Розроблені також DRAM із внутрішніми схемами регенерації; такі ЗП називаються квазістатичними.

Статичні ОЗП розподіляють на такі типи:

асинхронні – керуючі сигнали можна задавати як імпульсами, так і рівнями;

тактовані – в них деякі сигнали мають бути обов'язково імпульсами, наприклад, сигнал дозволу роботи  $\overline{CS}$ ;

синхронні, в яких організований конвеєрний канал передачі даних, що синхронізується від тактової системи процесора.

Динамічні ЗП характеризуються найбільшою інформаційною ємністю і невисокою вартістю, тому вони використовуються як основна пам'ять комп'ютерів. Статичні ЗП в 4–5 разів дорожчі динамічних і приблизно у стільки ж разів менша їхня інформаційна ємність. Їхнім достоїнством є висока швидкодія, а типовою областю застосування – схеми кеш-пам'яті.

Постійна пам'ять типу ROM(M) програмується при виготовленні за допомогою масок, тому її називають ПЗП масочним. В подальших різновидах ROM у позначеннях є буква P (від Programmable). Це – пам'ять, що одноразово програмується користувачем – PROM (в українській термінології ППЗП – програмовані ПЗП) та багаторазово програмується – EPROM, EEPROM.

Пам'ять типу Flash по ЕП подібна до EEPROM (інакше E2PROM), але має структурні й технологічні особливості, які дозволяють виділити її в окремий тип.

У ЗП з послідовним доступом дані, що записуються, створюють чергу. Зчитування виконується слово за словом в порядку записів або навпаки. Прямий порядок зчитування використовується в буферах FIFO з дисципліною "перший прийшов – перший вийшов (First In – First Out)", а також у файлових і циклічних ЗП.

Різниця між пам'яттю FIFO і файловою ЗП полягає в тому, що у FIFO записування у пустий буфер зразу доступне для читання (тобто поступає в кінець ланцюга моделі ЗП). У файлових ЗП дані поступають у початок ланцюга і з'являються на виході після деякого числа звертань, яке дорівнює числу елементів у ланцюзі.

У циклічних ЗП слова доступні одне за одним з постійним періодом, який визначається ємністю пам'яті. До них відноситься відеопам'ять (VRAM).

Зчитування в оберненому порядку властиве стековим ЗП з дисципліною "останній прийшов – першим вийшов". Такі ЗП називаються буферами LIFO (Last In – First Out).

Час доступу до конкретної одиниці інформації, що зберігається в послідовних ЗП, є випадковою величиною. В найгіршому випадку для такого доступу треба переглянути весь об'єм інформації, що зберігається у цій пам'яті.

Асоціативний доступ реалізує пошук інформації за деякою ознакою, а не за адресою. В найбільш повній версії всі слова, які зберігаються в пам'яті, можуть одночасно перевірятися на відповідність ознаці, наприклад, на збіг визначених полів слів – тегів (від tag) за ознакою, яку задає вхідне слово (тегова адреса). На вихід передаються слова, які задовольняють ознаку. Дисципліна видавання слів, якщо тегу задовольняє декілька слів, та дисципліна записування нових даних можуть бути різними. Основна область використання асоціативної пам'яті в комп'ютерах – кешування даних.

## 15.6. Загальна характеристика кеш-пам'яті

Кеш-пам'ять (від Cache – тайник) – це засіб копіювання і зберігання блоків даних основної пам'яті типу DRAM в процесі виконання програми. Кеш-пам'ять побудована на швидкодіючих тригерних ЕП, але має невелику ємність порівняно з основною динамічною пам'яттю. Кеш зберігає обмежене число даних і тегів. Тег містить інформацію про фізичну адресу і стан даних. При кожному зверненні до основної пам'яті спеціальний контролер перевіряє за тегом наявність цієї копії в кеші. Якщо вона є, то виробляється сигнал Hit (кеш-попадання) і звернення відбувається до кеш-пам'яті (рис. 15.3).

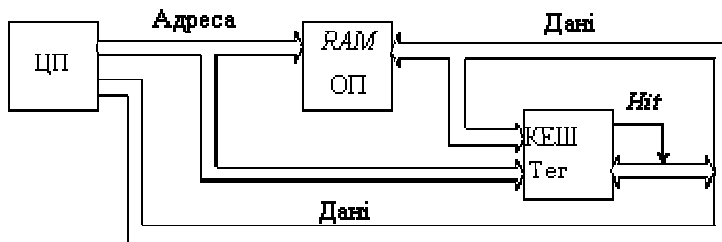


Рис. 15.3. Структура кеш-пам'яті

Якщо копії немає (кеш-промах), то сигнал Hit не виробляється і виконується читання з ОП та одночасне розміщення зчитаних даних в кеші.

Обмін з ОП може відбуватися двома способами:

перший: звернення до ОП поєднується з одночасним пошуком інформації в тегу. Звернення при попаданні до ОП анулюються; другий: звертання до ОП проводиться тільки після виявлення кеш-промаху.

У сучасних комп'ютерах кеш будують за дворівневою схемою:

первинний кеш (L1 Cache) має об'єм десятки Кбайт і вбудовується в процесор. Для підвищення продуктивності часто використовуються роздільні кеші для команд і даних (Гарвардська архітектура);

вторинний кеш (L2 Cache), звичайно встановлюють на системній платі, він має об'єм декілька Мбайт.

Більшість прикладних програм має циклічний характер і багаторазово використовує одні й ті самі дані, тому наявність кеша зменшує кількість звернень до відносно повільної ОП.

## 15.7. Загальна характеристика постійної пам'яті

Постійна пам'ять призначена для збереження програм, констант, табличних функцій та іншої інформації, яка записується заздалегідь і не змінюється в процесі поточної роботи комп'ютера. Вона застосовується також у перетворювачах кодів, знакогенераторах, у мікропрограмних пристроях керування. Загальним для всіх мікросхем постійної пам'яті є енергонезалежність, словникова організація і використання режиму зчитування як основного.

Мікросхеми постійної пам'яті розділяються на такі групи:

- ПЗП або ROM (Read Only Memory) – програмуються одноразово заводом-виробником, часто називаються масочними;
- ППЗП або PROM (Programmable ROM) – програмуються одноразово електричним способом користувачем;
- РПЗП-УФ або EPROM (Erasable PROM) – програмуються багаторазово (репрограмуються) з ультрафіолетовим стиранням і електричним записуванням;
- РПЗП-ЕС або EEPROM (Electrical EPROM) – програмуються і стираються багаторазово електричним способом.

## 15.8. Загальна характеристика флеш-пам'яті

Флеш-пам'ять (Flash Memory) використовує ЕП на транзисторах ЛІЗМОН з електричним стиранням і записуванням інформації. Вона відноситься до постійної пам'яті типу EEPROM, але ряд архітектурних і функціональних особливостей дозволили виділити флеш-пам'ять в окремий клас. Флеш-пам'ять використовує поряд з традиційними адресними і керуючими сигналами спеціальні команди. Інформація у мікросхемах флеш-пам'яті записується і зберігається в блоках визначеного розміру, іноді – призначення. При цьому стирання інформації здійснюється або для всієї пам'яті разом, або для великих блоків; це спрощує схеми ЕП. Флеш-пам'ять переважає EEPROM у тому, що не вимагає спеціальної апаратури для записування чи стирання даних.

Розрізняють такі види флеш-пам'яті:

файлова флеш-пам'ять (Flash File) – масив ЕП розділений на блоки однакового розміру (симетрична архітектура);

флеш-пам'ять з несиметричною архітектурою (Boot Block) – масив ЕП розділений на блоки різного розміру; один з блоків має апаратні засоби для захисту інформації в ньому;

флеш-пам'ять з можливістю стирання тільки всього масиву ЕП (Bulk Erase);

флеш-пам'ять з можливістю записування інформації за різних напруг програмування (Start Voltage);

пам'ять з використанням нових ЕП з чотирма станами, які зберігають по два біти (Strata Flash).

Файлова флеш-пам'ять орієнтована на заміну жорстких магнітних дисків. Такі ЗП в сотні разів зменшують споживану потужність, збільшують механічну міцність та надійність, зменшують їхні розміри і масу та на декілька порядків підвищують швидкодію при читанні даних. Мікросхеми, які заміняють магнітні диски, мають ідентичні блоки та розвинені засоби обміну інформацією. Мікросхеми файлової флеш-пам'яті фірми Intel мають інформаційну ємність 4 – 32 Мбіт, час доступу – 70 – 150 нс.

Мікросхеми Boot Block використовують однобайтову або перемикальну одно- чи двобайтову організацію і складаються з декількох блоків різного розміру. Один з блоків має додаткові апаратні засоби захисту від зміни даних; він призначений для зберігання дуже важливої інформації, яка не змінюється при модифікації даних в інших блоках. Мікросхеми Boot Block призначені для зберігання компонентів системного програмного забезпечення. Привілейований блок містить програму-завантажувач, яка записує з диска необхідні дані для ініціалізації пристроїв комп'ютера.

Мікросхема Boot Block типу 28F00BX/N, яка часто застосовується для зберігання програм базової системи введення-виведення (BIOS) в ПЕОМ, має час доступу 75–150 нс, гарантується 105 циклів стирання – програмування.

Мікросхема 28F00BX містить:

основний блок об'ємом 112 Кбайт;

два блоки параметрів ємністю по 4 Кбайт кожний;

блок-завантажувач об'ємом 4 Кбайт, стирання і програмування якого можливе тільки за особливих умов.

Основний блок і блоки параметрів захисту – рівноправні. Виділення невеликих блоків параметрів дозволяє зберігати в них ін-

формацію, яка часто змінюється.

Мікросхеми Bulk Erase мають однобайтову організацію ємності 32–256 Кбайт, час доступу 65–200 нс і являють собою єдиний масив, який стирається відразу.

#### 15.9. Загальна характеристика статичних запам'ятовуючих пристроїв

У статичних ЗП функцію запам'ятовування біта інформації виконують тригери. Вони реалізуються за будь-якою схмотехнікою – ТТЛШ, І2Л, ЕЗП, п-МОН, КМОН та іншими. Найбільш інтенсивно розвиваються ОЗП на КМОН-структурах, які при зменшенні роздільної здатності до 0,2 мкм набувають високої швидкодії. При цьому вони зберігають свої традиційні переваги – велику інформаційну ємність та дуже мале енергоспоживання – до долей мікрровольта на один ЕП.

Статичні ОЗП (SRAM) звичайно мають структуру 2DM, а при невеликій інформаційній ємності будуються за структурою 2D. Вони широко використовуються в кеш-пам'яті, яка повинна мати максимально можливу швидкодію.

Для побудови ЕП статичних ЗП широко використовують RS-тригери за схмотехнікою КМОН. Типова схема такого RS-тригера містить:

власне тригер на транзисторах VT1 і VT2 (п-тип) та на навантажувальних транзисторах VT3 і VT4 (р-тип);  
ключі вибірки на транзисторах VT5 і VT6 (рис. 15.4).

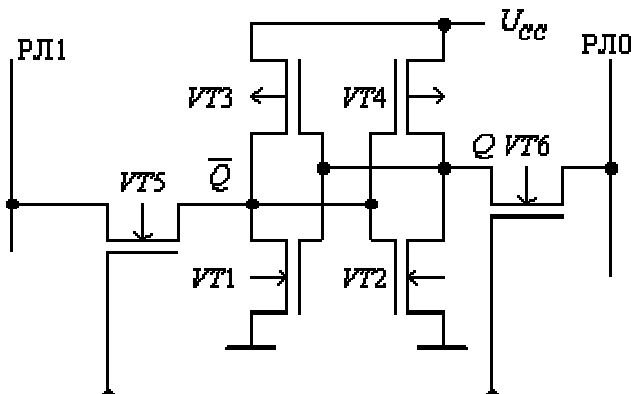


Рис. 15.4. Схема RS-тригера на КМОН структурах

З прямим Q та інверсним  $\bar{Q}$  виходами тригера через ключі вибірки пов'язані розрядні лінії записування–зчитування РЛ0 та РЛ1. В режимі зберігання транзистори VT5 і VT6 закриті. Перед записуванням даних на розрядні лінії подається високий рівень напруги. При виборі даного тригера (лінія вибору ЛВ=1) для записування одиниці встановлюють РЛ1=0, РЛ0=1, а при записуванні нуля – навпаки.

Під час зчитування даних двонаправлені ключі VT5 і VT6 відкриваються, якщо сигнал ЛВі=1. Паразитна ємність стоку закритого транзистора в тригері зменшується на мале значення  $\Delta U$ , а ємність стоку відкритого транзистора збільшується на таке саме значення. При цьому стан тригера не змінюється. Зміна сигналів на лініях Р30 і Р31 подається на входи диференційного підсилювача, який формує значення лог.0 або лог.1.

#### 15.10. Принцип побудови динамічного запам'ятовуючого елемента

У динамічній пам'яті типу DRAM інформація зберігається у вигляді зарядів на дуже малій ємності  $C_3=0,01 \dots 0,05$  пФ, яка створена між стоком і підкладкою МОН-транзистора (рис. 15.5).

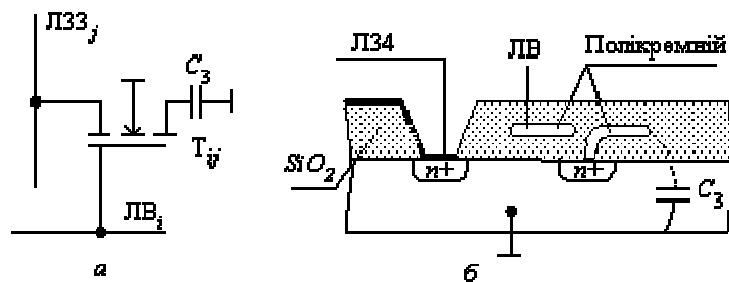


Рис. 15.5. Динамічний одностанторний ЕП: а – схема; б – топологія

Стік транзистора не має зовнішнього виводу. Для записування інформації на лінію вибірки ЛВі подається високий рівень напруги, яка відкриває транзистор Tij. Створюється провідний канал, і рівень напруги на розрядній лінії записування–зчитування ЛЗЗj визначає стан конденсатора  $C_3$ : заряджений при високому рівні (стан "1") та розряджений при низькому (стан "0")

Редакция рассчитывает на Вашу объективность, толерантность и **оперативность** и гарантирует анонимность рецензии во время переписки с автором вплоть до принятия статьи к печати. Имена рецензентов статей, принятых к печати, публикуются в журнале наряду с именами авторов. Номер журнала с публикацией направляется рецензенту.

*Рецензия может быть написана с учетом приведенной ниже канвы.*

Соответствует рубрике журнала (нужное оставить) **Электронные средства: исследования, разработки.**

### Оценка информационного потенциала (отметить)

[illegible]

### Текст рецензии

- а) показана ли актуальность темы; -да  
б) показана ли обоснованность постановки задачи;-да  
в) показана ли новизна информации;-да  
г) бесспорны ли исходные положения;- да  
д) убедительны ли доказательства; -да  
е) правомерны ли выводы; -требуют авторской доработки  
ж) возможно ли практическое применение;-да  
з) целесообразность таблиц, иллюстраций; -да  
и) строгость терминологии;-требует авторских уточнений  
к) стройность композиции рукописи,- требует авторских уточнений,  
в том числе оправданность объема.

— 1. Оценивается положительно (после авторской доработки и редактирования);

- 2. Необходимо учесть конкретные замечания, показанные в тексте рукописи (выделено желтым цветом)

**Пожелания автору по продолжению исследований (разработок):** -исследовать возможности применения предложенного метода для получения микро- и наноизображений микроэлектронных устройств (ИС, МЭМС и др)

## Chapter 6

# ARITHMETIC FOR DIGITAL SYSTEMS



- 
- [Introduction](#)
  - [Notation Systems](#)
  - [Principle of Generation and Propagation](#)
  - [The 1-bit Full Adder](#)
  - [Enhancement Techniques for Adders](#)
  - [Multioperand Adders](#)
  - [Multiplication](#)
  - [Addition and Multiplication in Galois Fields, GF\(2<sup>n</sup>\)](#)
- 

### 6.1 Introduction

Computation speeds have increased dramatically during the past three decades resulting from the development of various technologies. The execution speed of an arithmetic operation is a function of two factors. One is the circuit technology and the other is the algorithm used. It can be rather confusing to discuss both factors simultaneously; for instance, a ripple-carry adder implemented in GaAs technology may be faster than a carry-look-ahead adder implemented in CMOS. Further, in any technology, logic path delay depends upon many different factors: the number of gates through which a signal has to pass before a decision is made, the logic capability of each gate, cumulative distance among all such serial gates, the electrical signal propagation time of the medium per unit distance, etc. Because the logic path delay is attributable to the delay internal and external to logic gates, a comprehensive model of performance would have to include technology, distance, placement, layout, electrical and logical capabilities of the gates. It is not feasible to make a general model of arithmetic performance and include all these variables.

The purpose of this chapter is to give an overview of the different components used in the design of arithmetic operators. The following parts will not exhaustively go through all these components. However, the algorithms used, some mathematical concepts, the architectures, the implementations at the block, transistor or even mask level will be presented. This chapter will start by the presentation of various notation systems. Those are important because they influence the architectures, the size and the performance of the arithmetic components. The well known and used principle of generation and propagation will be explained and basic implementation at transistor level will be given as examples. The basic full adder cell (FA) will be shown as a brick used in the construction of various systems. After that, the problem of building large adders will lead to the presentation of enhancement techniques. Multioperand adders are of particular interest when building special CPU's and especially multipliers. That is why, certain algorithms will be introduced to give a better idea of the building of





multipliers. After the show of the classical approaches, a logarithmic multiplier and the multiplication and addition in the Galois Fields will be briefly introduced. Muller [Mull92] and Cavanagh [Cava83] constitute two reference books on the matter.



## 6.2 Notation Systems

### 6.2.1 Integer Unsigned

The binary number system is the most conventional and easily implemented system for internal use in digital computers. It is also a positional number system. In this mode the number is encoded as a vector of  $n$  bits (digits) in which each is weighted according to its position in the vector. Associated to each vector is a base (or radix)  $r$ . Each bit has an integer value in the range 0 to  $r-1$ . In the binary system where  $r=2$ , each bit has the value 0 or 1. Consider a  $n$ -bit vector of the form:

$$\mathbf{A} = a_{n-1}a_{n-2}\dots a_0 \quad (1)$$

where  $a_i=0$  or 1 for  $i$  in  $[0, n-1]$ . This vector can represent positive integer values  $V = A$  in the range 0 to  $2^n-1$ , where:

$$A = 2^{n-1} \cdot a_{n-1} + 2^{n-2} \cdot a_{n-2} + \dots + 2^0 \cdot a_0 \quad (2)$$

The above representation can be extended to include fractions. An example follows. The string of binary digits 1101.11 can be interpreted to represent the quantity :

$$2^3 \cdot 1 + 2^2 \cdot 1 + 2^1 \cdot 0 + 2^0 \cdot 1 + 2^{-1} \cdot 1 + 2^{-2} \cdot 1 = 13.75 \quad (3)$$

The following Table 1 shows the 3-bit vector representing the decimal expression to the right.

<b>000</b>	<b>0</b>
<b>001</b>	<b>1</b>
<b>010</b>	<b>2</b>
<b>011</b>	<b>3</b>
<b>100</b>	<b>4</b>
<b>101</b>	<b>5</b>
<b>110</b>	<b>6</b>
<b>111</b>	<b>7</b>

**Table-6.1:** Binary representation unsigned system with 3 digits

### 6.2.2 Integer Signed

If only positive integers were to be represented in fixed-point notation, then an  $n$ -bit word would permit a range from 0 to  $2^n-1$ . However, both positive and negative integers are used in computations

and an encoding scheme must be devised in which both positive and negative numbers are distributed as evenly as possible. There must be also an easy way to distinguish between positive and negative numbers. The left most digit is usually reserved for the sign. Consider the following number A with radix r,

$$A = (a_{n-1} a_{n-2} \dots a_1 a_0)_r$$

where the sign digit  $a_{n-1}$  has the following value:

$$a_{n-1} = \begin{cases} 0, & \text{if } |A| \geq 0 \\ r-1, & \text{if } |A| < 0 \end{cases}$$

for binary numbers where  $r=2$ , the previous equation becomes:

$$a_{n-1} = \begin{cases} 0, & \text{if } |A| \geq 0 \\ 1, & \text{if } |A| < 0 \end{cases}$$

The remaining digits in A indicate either the true value or the magnitude of A in a complemented form.

#### 6.2.2.1 Absolute value

000	0
001	1
010	2
011	3
100	0
101	-1
110	-2
111	-3

**Table-6.2:** binary representation signed absolute value

In this representation, the high-order bit indicates the sign of the integer (0 for positive, 1 for negative). A positive number has a range of 0 to  $2^{n-1}-1$ , and a negative number has a range of 0 to  $-(2^{n-1}-1)$ . The representation of a positive number is :

$$A = (0a_{n-2} \dots a_1 a_0)_r$$

The negatives numbers having the following representation:

$$-A = ((r-1)a_{n-2} \dots a_1 a_0)_r$$

One problem with this kind of notation is the dual representation of the number 0. The next problem is when adding two number with opposite signs. The magnitudes have to be compared to determine the sign of the result.

#### 6.2.2.2 1's complement

000	0
001	1
010	2
011	3
100	-3
101	-2
110	-1
111	-0

**Table-6.3:** binary representation signed

In this representation, the high-order bit also indicates the sign of the integer (0 for positive, 1 for negative). A positive number has a range of 0 to  $2^{n-1}-1$ , and a negative number has a range of 0 to  $-(2^{n-1}-1)$ . The representation of a positive number is :

$$A = (0a_{n-2} \dots a_1 a_0)_r$$

The negatives numbers having the following representation:

$$-A = ((r-1)\overline{a_{n-2}} \dots \overline{a_1} \overline{a_0})_r$$

One problem with this kind of notation is the dual representation of the number 0. The next problem is when adding two number with opposite signs. The magnitudes have to be compared to determine the sign of the result.

#### 6.2.2.3 2's complement

000	0
001	1
010	2
011	3
100	-4
101	-3
110	-2
111	-1

**Table-6.4:** binary representation signed in 2's complement

In this notation system (radix 2), the value of A is represented such as:

$$A = -2^{n-1}.a_{n-1} + 2^{n-2}.a_{n-2} + \dots + 2^0.a_0$$

The test sign is also a simple comparison of two bits. There is a unique representation of 0. Addition and subtraction are easier because the result comes out always in a unique 2's complement form.

#### 6.2.3 Carry Save

In some particular operations requiring big additions such as in multiplication or in filtering operations, the carry save notation is used. This notation can be either used in 1's or 2's or whatever other definition. It only means that for the result of an addition, the result will be coded in two digits

which are the carry in the sum digit. When coming to the multioperand adders and multipliers, this notion will be understood by itself.

#### 6.2.4 Redundant Notation

It has been stated that each bit in a number system has an integer value in the range 0 to r-1. This produces a digit set S:

$$S = \{0, 1, 2, \dots, r-1\}_{(4)}$$

in which all the digits of the set are positively weighted. It is also possible to have a digit set in which both positive- and negative-weighted digits are allowed [Aviz61] [Taka87], such as:

$$T = \{-l, \dots, -1, 0, +1, \dots, +l\}_{(5)}$$

where l is a positive integer representing the upper limit of the set. This is considered as a redundant number system, because there may be more than one way to represent a given number. Each digit of a redundant number system can assume the 2(l+1) values of the set T. The range of l is:

$$\left\lceil \frac{r-1}{2} \right\rceil \leq l \leq r-1 \quad (6)$$

Where:  $\left\lceil \frac{r-1}{2} \right\rceil$  is called the ceiling of .

For any number x , the ceiling of x is the smallest integer not less than x. The floor of x , is the largest integer not greater than x. Since the integer l bigger or equal than 1 and r bigger or equal than 2, then the maximum magnitude of l will be

$$\left\lfloor \frac{r}{2} \right\rfloor \quad (7)$$

Thus for r=2, the digit set is:

$$T_{r=2} = \{-1, 0, +1\}_{(8)}$$

For r=4, the digit set is

$$T_{r=4} = \{-2, -1, 0, +1, +2\}_{(9)}$$

For example, for n=4 and r=2, the number A=-5 has four representation as shown below on Table 5.

	$2^3$	$2^2$	$2^1$	$2^0$
A=	0	-1	0	-1
A=	0	-1	-1	1
A=	-1	0	1	1
A=	-1	1	0	-1

**Table-6.5:** Redundant representation of A=-5 when r=4

This multirepresentation makes redundant number systems difficult to use for certain arithmetic operations. Also, since each signed digit may require more than one bit to represent the digit, this may increase both the storage and the width of the storage bus.

However, redundant number systems have an advantage for the addition which is that it is possible to eliminate the problem of the propagation of the carry bit. This operation can be done in a constant time independent of the length of the data word. The conversion from binary to binary redundant is usually a duplication or juxtaposition of bits and it does not cost anything. On the contrary, the opposite conversion means an addition and the propagation of the carry bit cannot be removed.

Let us consider the example where  $r=2$  and  $l=1$ . In this system the three used digits are -1, 0, +1.

The representation of 1 is 10, because  $1-0=1$ .

The representation of -1 is 01, because  $0-1=-1$ .

One representation of 0 is 00, because  $0-0=0$ .

One representation of 0 is 11, because  $1-1=0$ .

The addition of 7 and 5 give 12 in decimal. The same is equivalent in a binary non redundant system to  $111 + 101$ :

$$\begin{array}{r}
 \text{carry} \quad 1 \quad 1 \\
 1 \quad 1 \quad 1 \quad 1 \\
 + \quad 1 \quad 0 \quad 1 \\
 \hline
 1 \quad 1 \quad 0 \quad 0
 \end{array}$$

We note that a carry bit has to be added to the next digits when making the operation "by hand". In the redundant system the same operation absorbs the carry bit which is never propagated to the next order digits:

$$\begin{array}{r}
 \text{carry} \quad 1 \quad 1 \\
 1 \quad 10 \quad 10 \quad 10 \\
 + \quad 10 \quad 00 \quad 10 \\
 \hline
 1 \quad 00 \quad 11 \quad 00
 \end{array}$$

The result 1001100 has now to be converted to the binary non redundant system. To achieve that, each couple of bits has to be added together. The eventual carry has to be propagated to the next order bits:

$$\begin{array}{r}
 \text{carry} \quad 1 \quad 0 \\
 1 \quad 00 \quad 11 \quad 00 \\
 \quad \downarrow \quad \downarrow \quad \downarrow \\
 1 \quad 1 \quad 0 \quad 0
 \end{array}$$



## 6.3 Principle of Generation and Propagation

### 6.3.1 The Concept

The principle of Generation and Propagation seems to have been discussed for the first time by Burks, Goldstine and Von Neumann [BGNe46]. It is based on a simple remark: when adding two numbers A and B in 2's complement or in the simplest binary representation ( $A=a_{n-1}...a_1a_0$ ,  $B=b_{n-1}...b_1b_0$ ), when  $a_i=b_i$  it is not necessary to know the carry  $c_i$ . So it is not necessary to wait for its calculation in order to determine  $c_{i+1}$  and the sum  $s_{i+1}$ .

**If  $a_i=b_i=0$ , then necessarily  $c_{i+1}=0$**

**If  $a_i=b_i=1$ , then necessarily  $c_{i+1}=1$**

This means that when  $a_i=b_i$ , it is possible to add the bits greater than the  $i$ th, before the carry information  $c_{i+1}$  has arrived. The time required to perform the addition will be proportional to the length of the longest chain  $i, i+1, i+2, i+p$  so that  $a_k$  not equal to  $b_k$  for  $k$  in  $[i, i+p]$ .

It has been shown [BGNe46] that the average value of this longest chain is proportional to the logarithm of the number of bits used to represent A and B. By using this principle of generation and propagation it is possible to design an adder with an average delay  $o(\log n)$ . However, this type of adder is usable only on asynchronous systems [Mull82]. Today the complexity of the systems is so high that asynchronous timing of the operations is rarely implemented. That is why the problem is rather to minimize the maximum delay rather than the average delay.

#### **Generation:**

This principle of generation allows the system to take advantage of the occurrences " $a_i=b_i$ ". In both cases ( $a_i=1$  or  $a_i=0$ ) the carry bit will be known.

#### **Propagation:**

If we are able to localize a chain of bits  $a_i a_{i+1}...a_{i+p}$  and  $b_i b_{i+1}...b_{i+p}$  for which  $a_k$  not equal to  $b_k$  for  $k$  in  $[i, i+p]$ , then the output carry bit of this chain will be equal to the input carry bit of the chain.

These remarks constitute the principle of generation and propagation used to speed the addition of two numbers.

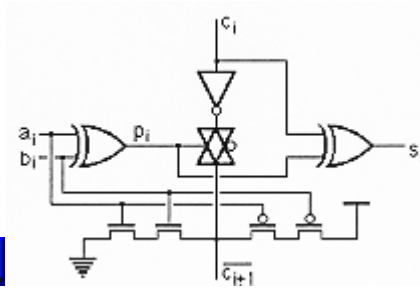
All adders which use this principle calculate in a first stage.

$$p_i = a_i \text{ XOR } b_i \quad (10)$$

$$g_i = a_i b_i \quad (11)$$

The previous equations determine the ability of the  $i^{\text{th}}$  bit to propagate carry information or to generate a carry information.

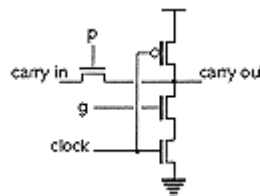
### 6.3.2 Transistor Formulation



[\[Click to enlarge image\]](#) **Figure-6.1:** A 1-bit adder with propagation signal controlling the pass-gate

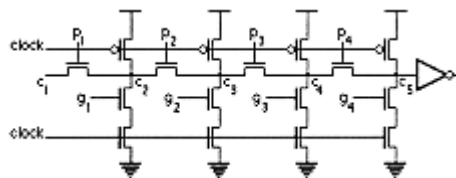
This implementation can be very performant (20 transistors) depending on the way the XOR function is built. The carry propagation of the carry is controlled by the output of the XOR gate. The generation of the carry is directly made by the function at the bottom. When both input signals are 1, then the inverse output carry is 0.

In the schematic of Figure 6.1, the carry passes through a complete transmission gate. If the carry path is precharged to VDD, the transmission gate is then reduced to a simple NMOS transistor. In the same way the PMOS transistors of the carry generation is removed. One gets a Manchester cell.



[\[Click to enlarge image\]](#) **Figure-6.2:** The Manchester cell

The Manchester cell is very fast, but a large set of such cascaded cells would be slow. This is due to the distributed RC effect and the body effect making the propagation time grow with the square of the number of cells. Practically, an inverter is added every four cells, like in Figure 6.3.



[\[Click to enlarge image\]](#) **Figure-6.3:** The Manchester carry cell

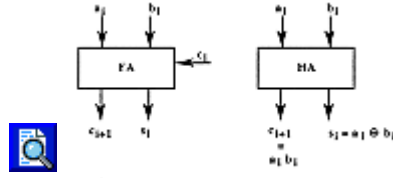


## 6.4 The 1-bit Full Adder

It is the generic cell used not only to perform addition but also arithmetic multiplication division and filtering operation. In this part we will analyse the equations and give some implementations with layout examples.

The adder cell receives two operands  $a_i$  and  $b_i$ , and an incoming carry  $c_i$ . It computes the sum and the outgoing carry  $c_{i+1}$ .

$$c_{i+1} = a_i \cdot b_i + a_i \cdot c_i + c_i \cdot b_i = a_i \cdot b_i + (a_i + b_i) \cdot c_i$$



[\[Click to enlarge image\]](#) **Figure-6.4:** The full adder (FA) and half adder (HA) cells

where

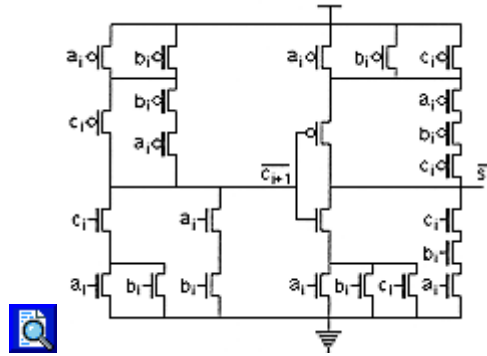
$$p_i = b_i \text{ XOR } a_i \text{ is the PROPAGATION signal (12)}$$

$$g_i = a_i \cdot b_i \text{ is the GENERATION signal (13)}$$

$$s_i = a_i \text{ XOR } b_i \text{ XOR } c_i \text{ (14)}$$

$$s_i = \overline{c_{i+1}} \cdot (a_i + b_i + c_i) + a_i \cdot b_i \cdot c_i \text{ (15)}$$

These equation can be directly translated into two N and P nets of transistors leading to the following schematics. The main disadvantage of this implementation is that there is no regularity in the nets.



[\[Click to enlarge image\]](#) **Figure-6.5:** Direct transcription of the previous equations

The dual form of each equation described previously can be written in the same manner as the normal form:

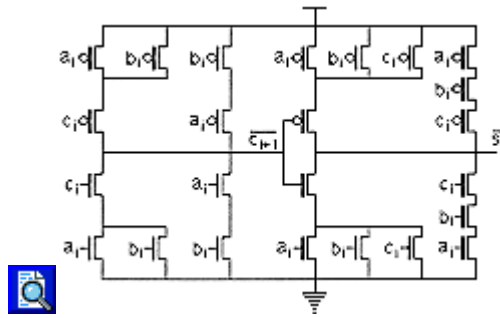
$$\begin{aligned} c_{i+1} &= a_i \cdot b_i + (a_i + b_i) \cdot c_i \text{ (16)} \\ \overline{c_{i+1}} &= (\overline{a_i} + \overline{b_i}) \cdot (\overline{c_i} + \overline{a_i} \cdot \overline{b_i}) \text{ dual of (16) (17)} \\ \overline{c_{i+1}} &= (\overline{a_i} + \overline{b_i}) \cdot \overline{c_i} + \overline{a_i} \cdot \overline{b_i} \text{ (18)} \end{aligned}$$

In the same way :

$$\begin{aligned} s_i &= \overline{c_{i+1}} \cdot (a_i + b_i + c_i) + a_i \cdot b_i \cdot c_i \text{ (19)} \\ \overline{s_i} &= (\overline{c_{i+1}} + \overline{a_i} \cdot \overline{b_i} \cdot \overline{c_i}) \cdot (\overline{a_i} + \overline{b_i} + \overline{c_i}) \text{ dual of (19) (20)} \\ \overline{s_i} &= c_{i+1} \cdot (\overline{a_i} + \overline{b_i} + \overline{c_i}) + \overline{a_i} \cdot \overline{b_i} \cdot \overline{c_i} \text{ (21)} \end{aligned}$$

The schematic becomes symmetrical (Figure 6.6), and leads to a better layout :



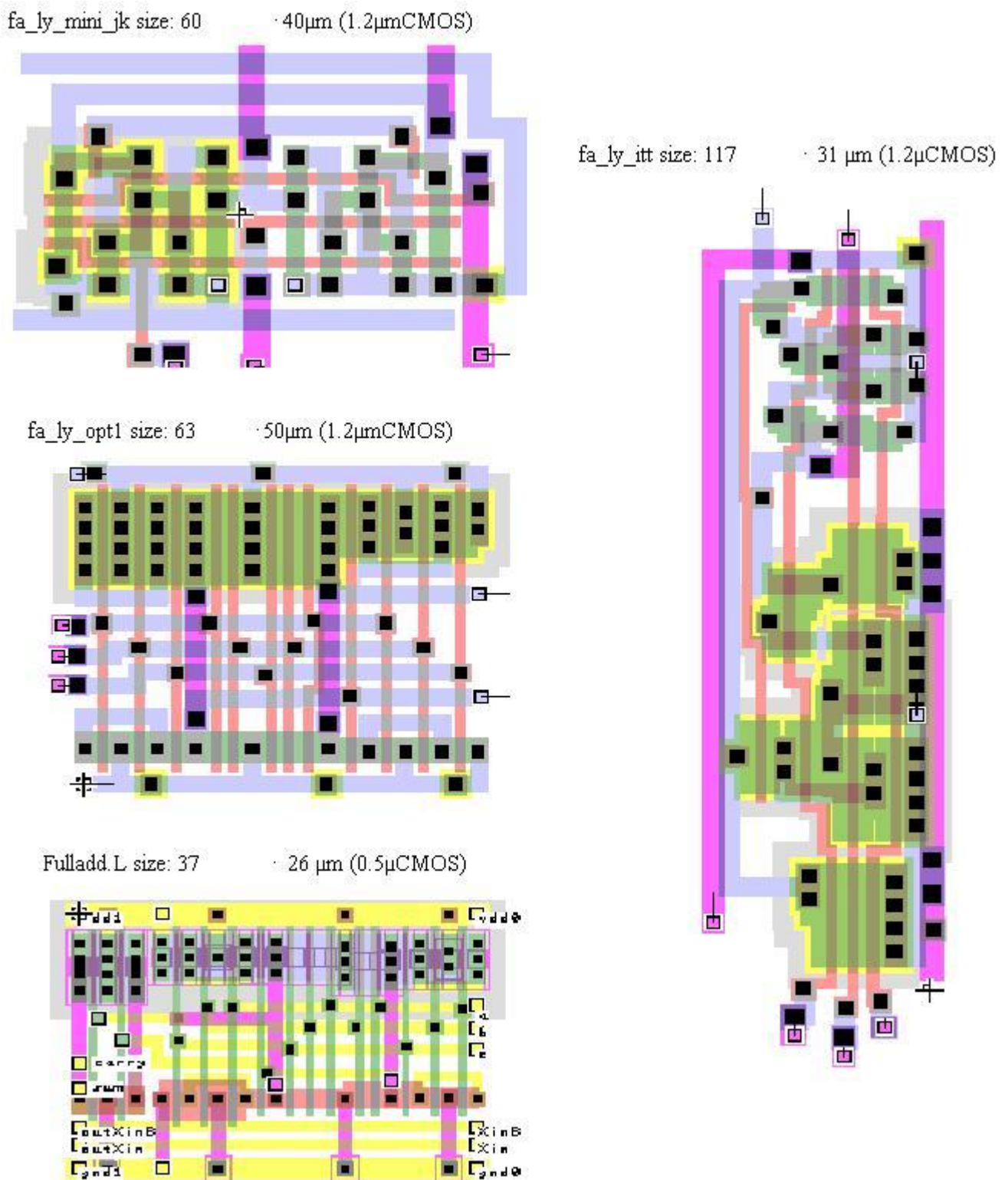


[\[Click to enlarge image\]](#) **Figure-6.6:** Symmetrical implementation due to the dual expressions of  $c_i$  and  $s_i$ .

The following Figure 6.7 shows different physical layouts in different technologies. The size, the technology and the performance of each cell is summarized in the next Table 6.

Name of cell	Number of Tr.	Size ( $\mu\text{m}^2$ )	Technology	Worst Case Delay (ns) (Typical Conditions)
fa_ly_mini_jk	24	2400	1.2 $\mu$	20
fa_ly_op1	24	3150	1.2 $\mu$	5
Fulladd.L	28	962	0.5 $\mu$	1.5
fa_ly_itt	24	3627	1.2 $\mu$	10

**Table-6.6:** Characteristics of layout cells from Figure 7.



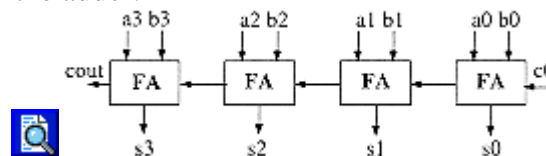
**Figure-6.7:** Mask layout for different Full Adder cells



## 6.5 Enhancement Techniques for Adders

The operands of addition are the addend and the augend. The addend is added to the augend to form the sum. In most computers, the augmented operand (the augend) is replaced by the sum, whereas the

addend is unchanged. High speed adders are not only for addition but also for subtraction, multiplication and division. The speed of a digital processor depends heavily on the speed of adders. The adders add vectors of bits and the principal problem is to speed-up the carry signal. A traditional and non optimized four bit adder can be made by the use of the generic one-bit adder cell connected one to the other. It is the ripple carry adder. In this case, the sum resulting at each stage need to wait for the incoming carry signal to perform the sum operation. The carry propagation can be speed-up in two ways. The first –and most obvious– way is to use a faster logic circuit technology. The second way is to generate carries by means of forecasting logic that does not rely on the carry signal being rippled from stage to stage of the adder.



[\[Click to enlarge image\]](#) **Figure-6.8:** A 4-bit parallel ripple carry adder

Generally, the size of an adder is determined according to the type of operations required, to the precision or to the time allowed to perform the operation. Since the operands have a fixed size, it becomes important to determine whether or not there is a detected overflow

**Overflow:** An overflow can be detected in two ways. First an overflow has occurred when the sign of the sum does not agree with the signs of the operands and the sign s of the operands are the same. In an n-bit adder, overflow can be defined as:

$$\text{Overflow} = a_{n-1}b_{n-1}\overline{s_{n-1}} + \overline{a_{n-1}}\overline{b_{n-1}}s_{n-1} \quad (22)$$

Secondly, if the carry out of the high order numeric (magnitude) position of the sum and the carry out of the sign position of the sum agree, the sum is satisfactory; if they disagree, an overflow has occurred. Thus,

$$\text{Overflow} = c_{n-1} \oplus c_{n-2} \quad (23)$$

A parallel adder adds two operands, including the sign bits. An overflow from the magnitude part will tend to change the sign of the sum. So that an erroneous sign will be produced. The following Table 7 summarizes the overflow detection

$a_{n-1}$	$b_{n-1}$	$s_{n-1}$	$c_{n-1}$	$c_{n-2}$	Overflow
0	0	0	0	0	0
0	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	0

**Table-6.7:** Overflow detection for 1's and 2's complement

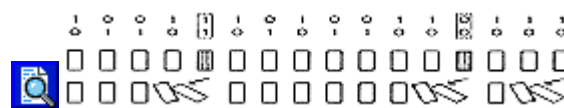
Coming back to the acceleration of the computation, two major techniques are used: speed-up techniques (Carry Skip and Carry Select), anticipation techniques (Carry Look Ahead, Brent and Kung and C3i). Finally, a combination of these techniques can prove to be an optimum for large adders.

### 6.5.1 The Carry-Skip Adder

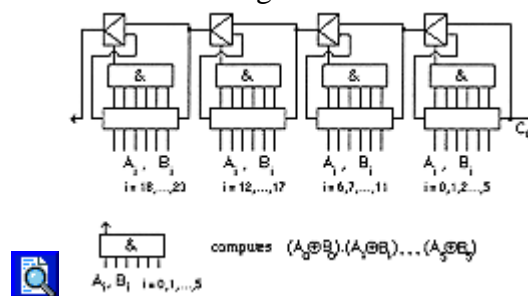
Depending on the position at which a carry signal has been generated, the propagation time can be variable. In the best case, when there is no carry generation, the addition time will only take into account the time to propagate the carry signal. Figure 9 is an example illustrating a carry signal generated twice, with the input carry being equal to 0. In this case three simultaneous carry propagations occur. The longest is the second, which takes 7 cell delays (it starts at the 4th position and ends at the 11th position). So the addition time of these two numbers with this 16-bits Ripple Carry Adder is  $7.k + k'$ , where  $k$  is the delay cell and  $k'$  is the time needed to compute the 11th sum bit using the 11th carry-in.

With a Ripple Carry Adder, if the input bits  $A_i$  and  $B_i$  are different for all position  $i$ , then the carry signal is propagated at all positions (thus never generated), and the addition is completed when the carry signal has propagated through the whole adder. In this case, the Ripple Carry Adder is as slow as it is large. Actually, Ripple Carry Adders are fast only for some configurations of the input words, where carry signals are generated at some positions.

Carry Skip Adders take advantage both of the generation or the propagation of the carry signal. They are divided into blocks, where a special circuit detects quickly if all the bits to be added are different ( $P_i = 1$  in all the block). The signal produced by this circuit will be called block propagation signal. If the carry is propagated at all positions in the block, then the carry signal entering into the block can directly bypass it and so be transmitted through a multiplexer to the next block. As soon as the carry signal is transmitted to a block, it starts to propagate through the block, as if it had been generated at the beginning of the block. Figure 6.10 shows the structure of a 24-bits Carry Skip Adder, divided into 4 blocks.



[\[Click to enlarge image\]](#) **Figure-6.10:** The "domino behaviour of the carry propagation and generation signals



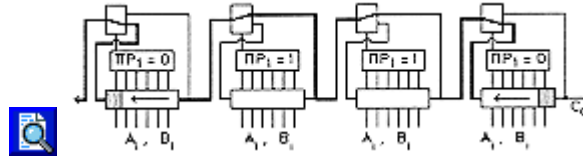
[\[Click to enlarge image\]](#) **Figure-6.10a:** Block diagram of a carry skip adder

To summarize, if in a block all  $A_i$ 's  $\neq B_i$ 's, then the carry signal skips over the block. If they are equal, a carry signal is generated inside the block and needs to complete the computation inside before to give the carry information to the next block.

## OPTIMISATION TECHNIQUE WITH BLOCKS OF EQUAL SIZE

It becomes now obvious that there exist a trade-off between the speed and the size of the blocks. In this part we analyse the division of the adder into blocks of equal size. Let us denote  $k_1$  the time needed by the carry signal to propagate through an adder cell, and  $k_2$  the time it needs to skip over one block. Suppose the  $N$ -bit Carry Skip Adder is divided into  $M$  blocks, and each block contains  $P$  adder cells. The actual addition time of a Ripple Carry Adder depends on the configuration of the input words. The completion time may be small but it also may reach the worst case, when all adder cells propagate the

carry signal. In the same way, we must evaluate the worst carry propagation time for the Carry Skip Adder. The worst case of carry propagation is depicted in Figure 6.11.



[\[Click to enlarge image\]](#) **Figure-6.11:** Worst case for the propagation signal in a Carry Skip adder with blocks of equal size

The configuration of the input words is such that a carry signal is generated at the beginning of the first block. Then this carry signal is propagated by all the succeeding adder cells but the last which generates another carry signal. In the first and the last block the block propagation signal is equal to 0, so the entering carry signal is not transmitted to the next block. Consequently, in the first block, the last adder cells must wait for the carry signal, which comes from the first cell of the first block. When going out of the first block, the carry signal is distributed to the 2<sup>nd</sup>, 3<sup>rd</sup> and last block, where it propagates. In these blocks, the carry signals propagate almost simultaneously (we must account for the multiplexer delays). Any other situation leads to a better case. Suppose for instance that the 2<sup>nd</sup> block does not propagate the carry signal (its block propagation signal is equal to zero), then it means that a carry signal is generated inside. This carry signal starts to propagate as soon as the input bits are settled. In other words, at the beginning of the addition, there exists two sources for the carry signals. The paths of these carry signals are shorter than the carry path of the worst case. Let us formalize that the total adder is made of N adder cells. It contains M blocks of P adder cells. The total of adder cells is then

$$N=M \cdot P \quad (24)$$

The time T needed by the carry signal to propagate through P adder cells is

$$T=k_1 \cdot P \quad (25)$$

The time T' needed by the carry signal to skip through M adder blocks is

$$T'=k_2 \cdot M \quad (26)$$

The problem to solve is to minimize the worst case delay which is:

$$T_{\text{worstcase}} = 2 \cdot P \cdot k_1 + (M - 2) \cdot k_2 \quad (27)$$

$$T_{\text{worstcase}} = 2 \cdot \frac{N}{M} \cdot k_1 + (M - 2) \cdot k_2 \quad (28)$$

So that the function to be minimized is:

$$f(x) = 2 \cdot \frac{N}{x} \cdot k_1 + (x - 2) \cdot k_2 \quad (29)$$

The minimum is obtained for:

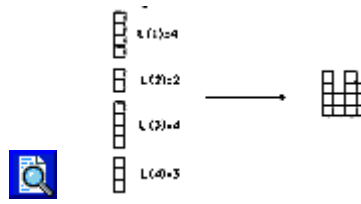
$$x_{\min} = \sqrt{2 \cdot N \cdot \frac{k_1}{k_2}} = 2(\sqrt{2 \cdot N \cdot k_1 \cdot k_2} - k_2) \quad (30)$$

## OPTIMISATION TECHNIQUE WITH BLOCKS OF NON-EQUAL SIZE

Let us formalize the problem as a geometric problem. A square will represent the generic full adder cell. These cells will be grouped in P groups (in a column like manner).

$L(i)$  is the value of the number of bits of one column.

$L(1), L(2), \dots, L(P)$  are the P adjacent columns. (see Figure 6.12)



[\[Click to enlarge image\]](#) **Figure-6.12:** Geometric formalization

If a carry signal is generated at the  $i$ th section, this carry skips  $j-i-1$  sections and disappears at the end of the  $j$ th section. So the delay of propagation is:

$$T = k_1 \cdot (L(i) + L(j)) + k_2 \cdot \sum L(p) \quad (31)$$

By defining the constant  $a$  equal to:

$$a = \frac{k_2}{k_1} \quad (32)$$

one can position two straight lines defined by:

$$y = ax + b_1 \text{ (at the left most position)} \quad (33)$$

$$y = ax + b_2 \text{ (at the right most position)} \quad (34)$$

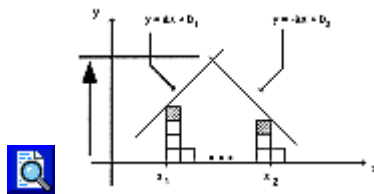
The constant  $a$  is equivalent to the slope dimension in the geometrical problem of the two two straight lines defined by equations (33) and (34). These straight lines are adjacent to the top of the columns and the maximum time can be expressed as a geometrical distance  $y$  equal to the  $y$ -value of the intersection of the two straight lines.

$$T = k_1(ax_1 + b_1 - ax_2 + b_2) + k_2(x_2 - x_1 - 2) \quad (35)$$

$$T = (k_1a - k_2)(x_1 - x_2) + k_1(b_1 + b_2) - 2k_2 \quad (36)$$

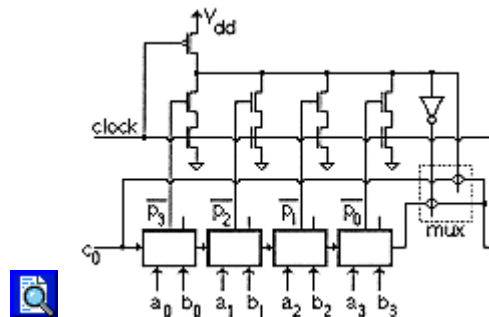
$$T = k_1(b_1 + b_2) - 2k_2 \text{ because } k_1a - k_2 = 0 \quad (37)$$





[\[Click to enlarge image\]](#) **Figure-6.13:** Representation of the geometrical worst delay

A possible implementation of a block is shown in Figure 6.14. In a precharged mode, the output of the four inverter-like structure is set to one. In the evaluation mode, the entire block is in action and the output will either receive  $c_0$  or the carry generated inside the comparator cells according to the values given to A and B. If there is no carry generation needed,  $c_0$  will be transmitted to the output. In the other case, one of the inversed  $p_i$ 's will switch the multiplexer to enable the other input.

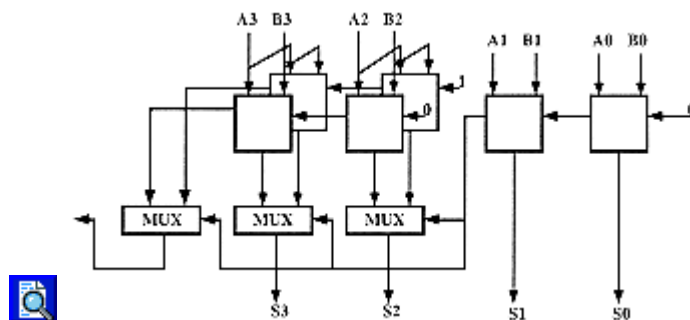


[\[Click to enlarge image\]](#) **Figure-6.14:** A possible implementation of the Carry Skip block

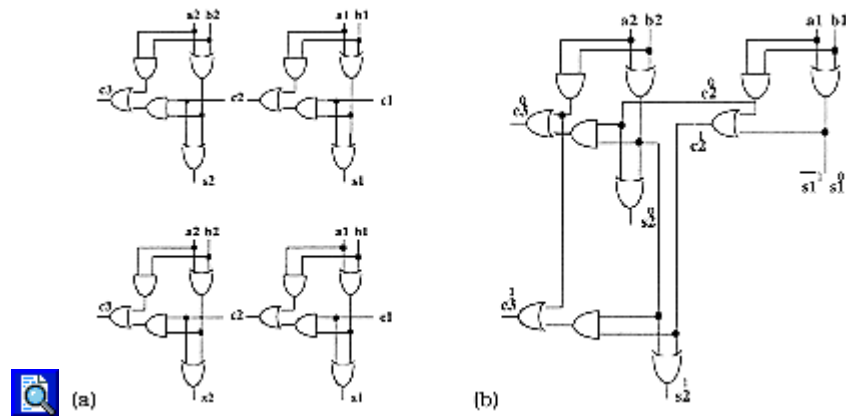
## 6.5.2 The Carry-Select Adder

This type of adder is not as fast as the Carry Look Ahead (CLA) presented in a next section. However, despite its bigger amount of hardware needed, it has an interesting design concept. The Carry Select principle requires two identical parallel adders that are partitioned into four-bit groups. Each group consists of the same design as that shown on Figure 15. The group generates a group carry. In the carry select adder, two sums are generated simultaneously. One sum assumes that the carry in is equal to one as the other assumes that the carry in is equal to zero. So that the predicted group carry is used to select one of the two sums.

It can be seen that the group carries logic increases rapidly when more high- order groups are added to the total adder length. This complexity can be decreased, with a subsequent increase in the delay, by partitioning a long adder into sections, with four groups per section, similar to the CLA adder.



[\[Click to enlarge image\]](#) **Figure-6.15:** The Carry Select adder



[\[Click to enlarge image\]](#) **Figure-6.16:** The Carry Select adder . (a) the design with non optimised used of the gates, (b) Merging of the redundant gates

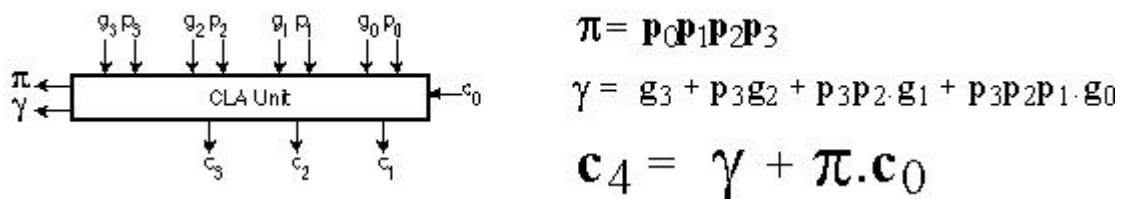
A possible implementation is shown on Figure 6.16, where it is possible to merge some redundant logic gates to achieve a lower complexity with a higher density.

### 6.5.3 The Carry Look-Ahead Adder

The limitation in the sequential method of forming carries, especially in the Ripple Carry adder arises from specifying  $c_i$  as a specific function of  $c_{i-1}$ . It is possible to express a carry as a function of all the preceding low order carry by using the recursivity of the carry function. With the following expression a considerable increase in speed can be realized.

$$C_i = G_i + G_{i-2} P_{i-1} + G_{i-3} P_{i-2} P_{i-1} + \dots + G_0 P_1 P_2 \dots P_{i-1} + C_0 P_0 P_1 P_2 \dots P_{i-1} \quad (38)$$

Usually the size and complexity for a big adder using this equation is not affordable. That is why the equation is used in a modular way by making groups of carry (usually four bits). Such a unit generates then a group carry which give the right predicted information to the next block giving time to the sum units to perform their calculation.

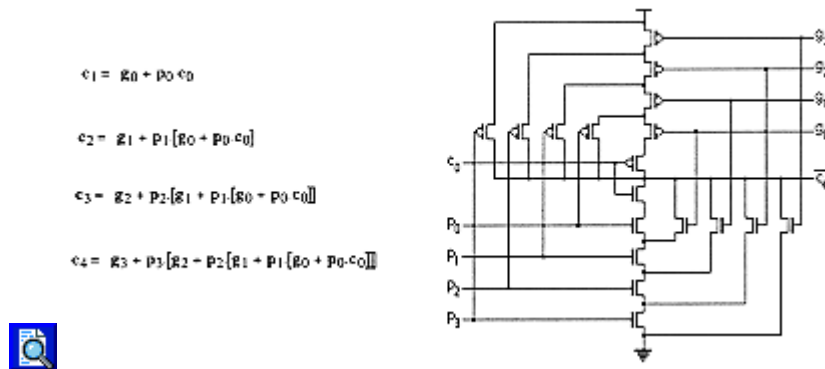


**Figure-6.17:** The Carry Generation unit performing the Carry group computation

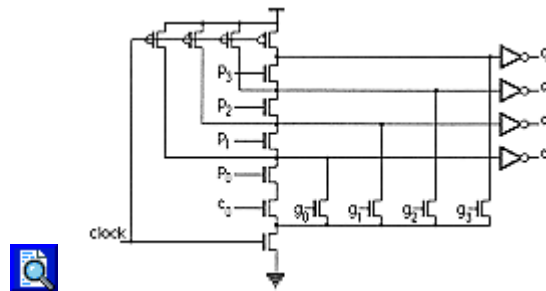
Such unit can be implemented in various ways, according to the allowed level of abstraction. In a CMOS process, 17 transistors are able to guarantee the static function (Figure 6.18). However this design requires a careful sizing of the transistors put in series.

The same design is available with less transistors in a dynamic logic design. The sizing is still an important issue, but the number of transistors is reduced (Figure 6.19).



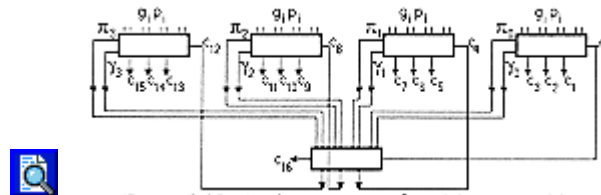


[\[Click to enlarge image\]](#) **Figure-6.18:** Static implementation of the 4-bit carry lookahead chain



[\[Click to enlarge image\]](#) **Figure-6.19:** Dynamic implementation of the 4-bit carry lookahead chain

To build large adders the preceding blocks are cascaded according to Figure 6.20.



[\[Click to enlarge image\]](#) **Figure-6.20:** Implementation of a 16-bit CLA adder

#### 6.5.4 The Brent and Kung Adder

The technique to speed up the addition is to introduce a "new" operator  $\Delta$  which combines couples of generation and propagation signals. This "new" operator come from the reformulation of the carry chain.

#### REFORMULATION OF THE CARRY CHAIN

Let  $a_n a_{n-1} \dots a_1$  and  $b_n b_{n-1} \dots b_1$  be  $n$ -bit binary numbers with sum  $s_{n+1} s_n \dots s_1$ . The usual method for addition computes the  $s_i$ 's by:

$$c_0 = 0 \quad (39)$$

$$c_i = a_i b_i + a_i c_{i-1} + b_i c_{i-1} \quad (40)$$

$$s_i = a_i ++ b_i ++ c_{i-1}, i = 1, \dots, n \quad (41)$$

$$s_{n+1} = c_n \quad (42)$$

Where ++ means the sum modulo-2 and  $c_i$  is the carry from bit position  $i$ . From the previous paragraph we can deduce that the  $c_i$ 's are given by:

$$c_0 = 0 \quad (43)$$

$$c_i = g_i + p_i c_{i-1} \quad (44)$$

$$g_i = a_i b_i \quad (45)$$

$$p_i = a_i + b_i \text{ for } i = 1, \dots, n \quad (46)$$

One can explain equation (44) saying that the carry  $c_i$  is either generated by  $a_i$  and  $b_i$  or propagated from the previous carry  $c_{i-1}$ . The whole idea is now to generate the carry's in parallel so that the  $n^{\text{th}}$  stage does not have to "wait" for the  $n-1^{\text{th}}$  carry bit to compute the global sum. To achieve this goal an operator  $\Delta$  is defined.

Let  $\Delta$  be defined as follows for any  $g, g', p$  and  $p'$  :

$$(g, p) \Delta (g', p') = (g + p \cdot g', p \cdot p') \quad (47)$$

**Lemma1:** Let  $(G_i, P_i) = (g_i, p_i)$  if  $i = 1$  (48)

$(g_i, p_i) \Delta (G_{i-1}, P_{i-1})$  if  $i$  in  $[2, n]$  (49)

Then  $c_i = G_i$  for  $i = 1, 2, \dots, n$ .

**Proof:** The Lemma is proved by induction on  $i$ . Since  $c_0 = 0$ , (44) above gives:

$$c_1 = g_1 + p_1 \cdot 0 = g_1 = G_1 \quad (50)$$

So the result holds for  $i=1$ . If  $i>1$  and  $c_{i-1} = G_{i-1}$ , then

$$(G_i, P_i) = (g_i, p_i) \Delta (G_{i-1}, P_{i-1}) \quad (51)$$

$$(G_i, P_i) = (g_i, p_i) \Delta (c_{i-1}, P_{i-1}) \quad (52)$$

$$(G_i, P_i) = (g_i + p_i \cdot c_{i-1}, P_i \cdot P_{i-1}) \quad (53)$$

$$\text{thus } G_i = g_i + p_i \cdot c_{i-1} \quad (54)$$

And from (44) we have :  $G_i = c_i$ .

**Lemma2:** The operator  $\Delta$  is associative.

**Proof:** For any  $(g_3, p_3), (g_2, p_2), (g_1, p_1)$  we have:

$$[(g_3, p_3) \Delta (g_2, p_2)] \Delta (g_1, p_1) = (g_3 + p_3 \cdot g_2, p_3 \cdot p_2) \Delta (g_1, p_1) \\ = (g_3 + p_3 \cdot g_2 + p_3 \cdot p_2 \cdot p_1, p_3 \cdot p_2 \cdot p_1) \quad (55)$$

and,

$$(g_3, p_3) \Delta [(g_2, p_2) \Delta (g_1, p_1)] = (g_3, p_3) \Delta (g_2 + p_2 \cdot g_1, p_2 \cdot p_1) \\ = (g_3 + p_3 \cdot (g_2 + p_2 \cdot g_1), p_3 \cdot p_2 \cdot p_1) \quad (56)$$

One can check that the expressions (55) and (56) are equal using the distributivity of  $\cdot$  and  $+$ .

To compute the  $c_i$ 's it is only necessary to compute all the  $(G_i, P_i)$ 's but by Lemmas 1 and 2,

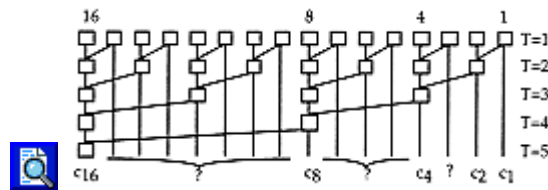
$$(G_i, P_i) = (g_i, p_i) \Delta (g_{i-1}, p_{i-1}) \Delta \dots \Delta (g_1, p_1) \quad (57)$$

can be evaluated in any order from the given  $g_i$ 's and  $p_i$ 's. The motivation for introducing the operator Delta is to generate the carry's in parallel. The carry's will be generated in a block or carry chain block, and the sum will be obtained directly from all the carry's and  $p_i$ 's since we use the fact that:

$$s_i = p_i \oplus c_{i-2} \text{ for } i=1, \dots, n \quad (58)$$

## THE ADDER

Based on the previous reformulation of the carry computation Brent and Kung have proposed a scheme to add two  $n$ -bit numbers in a time proportional to  $\log(n)$  and in area proportional to  $n \cdot \log(n)$ , for  $n$  bigger or equal to 2. Figure 6.21 shows how the carry's are computed in parallel for 16-bit numbers.



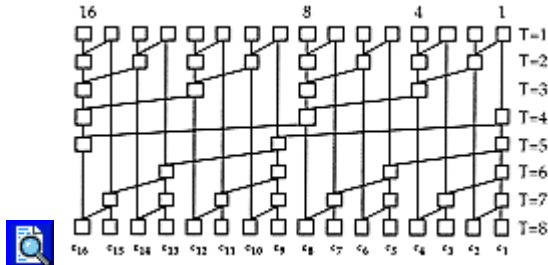
[\[Click to enlarge image\]](#) **Figure-6.21:** The first binary tree allowing the calculation of  $c_1, c_2, c_4, c_8, c_{16}$ .

Using this binary tree approach, only the  $c_i$ 's where  $i=2^k$  ( $k=0,1,...,n$ ) are computed. The missing  $c_i$ 's have to be computed using another tree structure, but this time the root of the tree is inverted (see Figure 6.22).

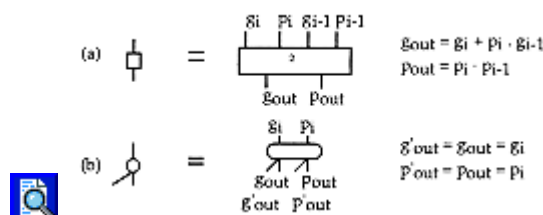
In Figure 6.21 and Figure 6.22 the squares represent a  $\Delta$  cell which performs equation (47). Circles represent a duplication cell where the inputs are separated into two distinct wires (see Figure 6.23).

When using this structure of two separate binary trees, the computation of two 16-bit numbers is performed in  $T=9$  stages of  $\Delta$  cells. During this time, all the carries are computed in a time necessary to traverse two independent binary trees.

According to Burks, Goldstine and Von Neumann, the fastest way to add two operands is proportional to the logarithm of the number of bits. Brent and Kung have achieved such a result.



[\[Click to enlarge image\]](#) **Figure-6.22:** Computation of all the carry's for  $n = 16$



[\[Click to enlarge image\]](#) **Figure-6.23:** (a) The  $\Delta$  cell, (b) the duplication cell

### 6.5.5 The C3i Adder

#### THE ALGORITHM

Let  $a_i$  and  $b_i$  be the digits of  $A$  and  $B$ , two  $n$ -bit numbers with  $i = 1, 2, ..., n$ . The carry's will be computed according to (59).

$$(G_i, P_i) = \bigtriangleup_{k=1}^1 (g_k, p_k) \quad (59)$$

with:  $G_i = c_i$  (60)

If we develop (59), we get:

$$(G_i, P_i) = (g_i, p_i) \bigtriangleup (g_{i-1}, p_{i-1}) \bigtriangleup (g_{i-2}, p_{i-2}) \bigtriangleup \dots \bigtriangleup (g_2, p_2) \bigtriangleup (g_1, p_1) \quad (61)$$

and by introducing a parameter  $m$  less or equal than  $n$  so that it exists  $q$  in  $\mathbb{N} \mid n = q.m$ , it is possible to obtain the couple  $(G_i, P_i)$  by forming groups of  $m$   $\bigtriangleup$  cells performing the intermediate operations detailed in (62) and (63).

$$(G_i, P_i) = \bigtriangleup_{k=(q-1)m+1}^1 (g_k, p_k) \dots \bigtriangleup_{k=m+1}^{2m} (g_k, p_k) \bigtriangleup_{k=1}^m (g_k, p_k) \quad (62)$$

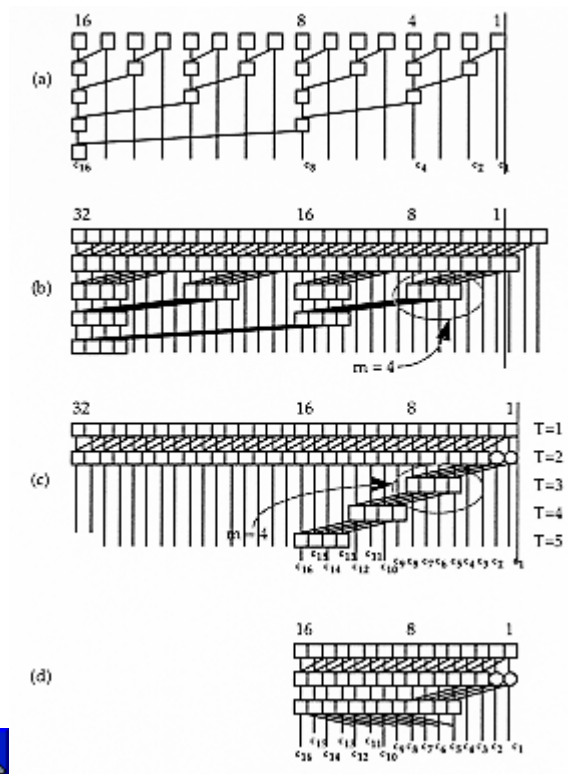
$$(G_i, P_i) = \bigtriangleup_{l=0}^i \left[ \bigtriangleup_{k=(l-1)m}^{lm} (g_k, p_k) \right] \quad (63)$$

This manner of computing the carries is strictly based on the fact that the operator  $\bigtriangleup$  is associative. It shows also that the calculation is performed sequentially, i.e. in a time proportional to the number of bits  $n$ . We will now illustrate this analytical approach by giving a way to build an architectural layout of this new algorithm. We will proceed to give a graphical method to place the  $\bigtriangleup$  cells defined in the previous paragraph [Kowa92].

## THE GRAPHICAL CONSTRUCTIONM

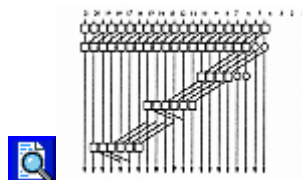
1. First build a binary tree of  $\bigtriangleup$  cells.
2. Duplicate this binary tree  $m$  times to the right ( $m$  is a power of two; see Remark1 in the next pages if  $m$  is not a power of two). The cells at the right of bit 1 determines the least significant bit (LSB).
3. Eliminate the cells at the right side of the LSB. Change the  $\bigtriangleup$  cells not connected to anything into duplication cells. Eliminate all cells under the second row of  $\bigtriangleup$  cells, except the right most group of  $m$   $\bigtriangleup$  cells.
4. Duplicate  $q$  times to the right by incrementing the row down the only group of  $m$   $\bigtriangleup$  cells left after step 3. This gives a visual representation of the delay read in Figure 6.29.
5. Shift up the  $q$  groups of  $\bigtriangleup$  cells, to get a compact representation of a "floorplan".

This complete approach is illustrated in Figure 6.24, where all the steps are carefully observed. The only cells necessary for this carry generation block to constitute a real parallel adder are the cells performing equations (45) and (46). The first row of functions is put at the top of the structure. The second one is pasted at the bottom.



[\[Click to enlarge image\]](#) **Figure-6.24:** (a) Step1, (b) Step2, (c) Step3 and Step4, (d) Step5

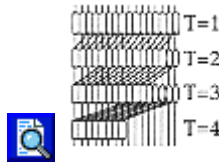
At this point of the definition, two remarks have to be made about the definition of this algorithm. Both concern the  $m$  parameter used to defined the algorithm. Remark 1 specifies the case for  $m$  not equal to  $2q$  ( $q$  in  $[0,1, \dots]$ ) as Remark 2 deals with the case where  $m=n$ .



[\[Click to enlarge image\]](#) **Figure-6.25:** Adder where  $m=6$ . The fan-out of the 11th carry bit is highlighted

**Remark 1:** For  $m$  not a power of two, the algorithm is built the same way up to the very last step. The only reported difference will concern the delay which will be equal to the next nearest power of two. This means that there is no special interest to build such versions of these adders. The fan-out of certain cells is even increased to three, so that the electrical behaviour will be degraded. Figure 6.25 illustrates the design of such an adder based on  $m=6$ . The fan-out of the  $\Delta$  cell of bit 11 is three. The delay of this adder is equivalent to the delay of an adder with a duplication with  $m=8$ .

**Remark 2:** For  $m$  equal to the number of bits of the adder, the algorithm reaches the real theoretical limit demonstrated by Burks, Goldstine and Von Neumann. The logarithmic time is attained using one depth of a binary tree instead of two in the case of Brent and Kung. This particular case is illustrated in Figure 6.26. The definition of the algorithm is followed up to Step3. Once the reproduction of the binary tree is made  $m$  times to the right, the only thing to do is to remove the cells at the negative bit positions and the adder is finished. Mathematically, one can notice that this is the limit. We will discuss later whether it is the best way to build an adder using  $m=n$ .



[\[Click to enlarge image\]](#) **Figure-6.26:** Adder where  $m=n$ . This constitutes the theoretical limit for the computation of the addition.

## COMPARISONS

In this section, we develop a comparison between adders obtained using the new algorithm with different values of  $m$ . On the plots of Figure 6.27 through Figure 6.29, the suffixes JK2, JK4, and JK8 will denote different adders obtained for  $m$  equal two, four or eight. They are compared to the Brent and Kung implementation and to the theoretical limit which is obtained when  $m$  equals  $n$ , the number of bits.

The comparison between these architectures is done according to the formalisation of a computational model described in [Kowa93]. We clearly see that BK's algorithm performs the addition with a delay proportional to the logarithm of the number of bits. JK2 performs the addition in a linear time, just as JK4 or JK8. The parameter  $m$  influences the slope of the delay. So that, the higher is  $m$ , the longer the delay stays under the logarithmic delay of (BK). We see that when one wants to implement the addition faster than (BK), there is a choice to make among different values of  $m$ . The choice will depend on the size of the adder because it is evident that a 24-bit JK2 adder (delay = 11 stages of  $\Delta$  cells) performs worse than BK (delay = 7 stages of  $\Delta$  cells).

On the other hand JK8 (delay = 5 stages of  $\Delta$  cells) is very attractive. The delay is better than BK up to 57 bits. At this point both delays are equal. Furthermore, even at equal delays (up to 73 bits) our implementation performs better in terms of regularity, modularity and ease to build. The strong advantage of this new algorithm compared to BK is that for a size of the input word which is not a power-of-two, the design of the cells is much easier. There is no partial binary tree to build. The addition of a bit to the adder is the addition of a bit-slice. This bit-slice is very compact and regular. Let us now consider the case where  $m$  equals  $n$  (denoted by XXX on our figures). The delay of such an adder is exactly one half of BK and it is the lowest bound we obtain. For small adders ( $n < 16$ ), the delay is very close to XXX. And it can be demonstrated that the delays (always in term of stages) of JK2, JK4, JK8 are always at least equal to XXX.

This discussion took into account the two following characteristics of the computational model:

- The gate of a stage computes a logical function in a constant time;
- The signal is divided into two signals in constant time (this occurs especially at the output of the first stage of  $\Delta$  cells).

And the conclusion of this discussion is that  $m$  has to be chosen as high as possible to reduce the global delay. When we turn to the comparisons concerning the area, we will take into account the following characteristics of our computational model:

- At most two wires cross at any point;
- A constant but predefined area of the gates with minimum width for the wires is used;
- The computation is made in a convex planar region.

For this discussion let us consider Figure 6.28 where we represent the area of the different adders versus the number of bits. It is obvious that for  $m$  being the smallest, the area will be the smallest as well. For  $m$  increasing up to  $n$ , we can see that the area will still be proportional to the number of bits

following a straight line. For  $m$  equal to  $n$  the area will be exactly one half of the BK area with a linear variation. The slope of this variation in both cases of BK and XXX will vary according to the intervals  $[2q, 2q+1]$  where  $q=0$ .

Here we could point out that the floorplan of BK could be optimised to become comparable to the one of XXX, but the cost of such an implementation would be very high because of the irregularity of the wirings and the interconnections. These considerations lead us to the following conclusion: to minimise the area of a new adder,  $m$  must be chosen low. This is contradictory with the previous conclusion. That is why a very wise choice of  $m$  will be necessary, and it will always depend on the targeted application. Finally, Figure 6.27 gives us the indications about the number of transistors used to implement our different versions of adders. These calculations are based on the dynamic logic family (TSPC: True Single Phase Clocking) described in [Kowa93]. When considering this graph, we see that BK and XXX are two limits of the family of our adders. BK uses the smallest number of transistors, whereas XXX uses up to five times more transistors. When  $m$  is highest, the number of transistors is highest.

Nevertheless, we see that the area is smaller than BK. A high density is an advantage, but an overhead in transistors can lead to higher power dissipation. This evident drawback in our algorithm is counterbalanced by the progress being made in the VLSI area. With the shrinking of the design rules, the size of the transistors decreases as well as the size of the interconnections. This leads to smaller power dissipation. This fact is even more pronounced when the technologies tend to decrease the power supply from 5V to 3.3V.

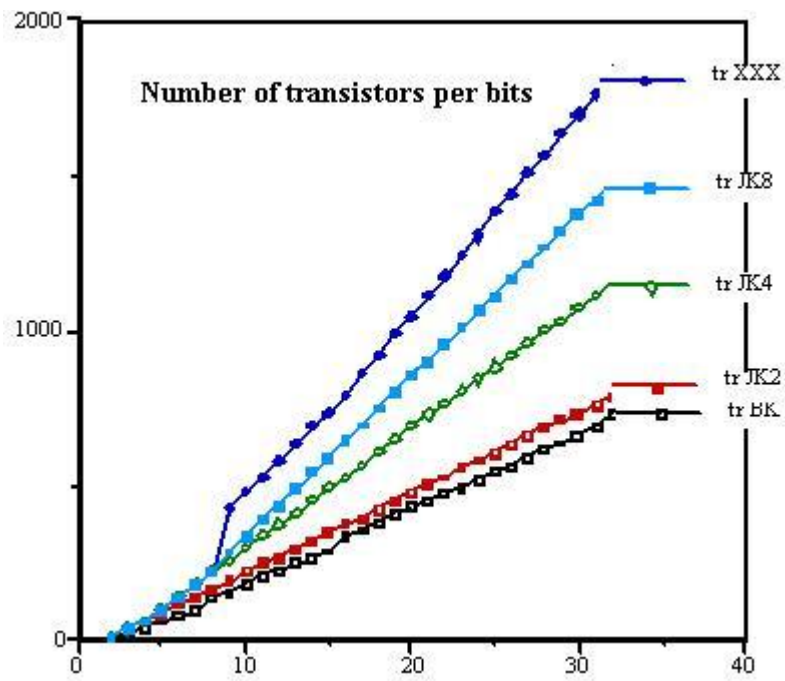
In other words, the increase in the number of transistors corresponds to the redundancy we introduce in the calculations to decrease the delay of our adders.

Now we will discuss an important characteristic of our computational model that differs from the model of Brent and Kung:

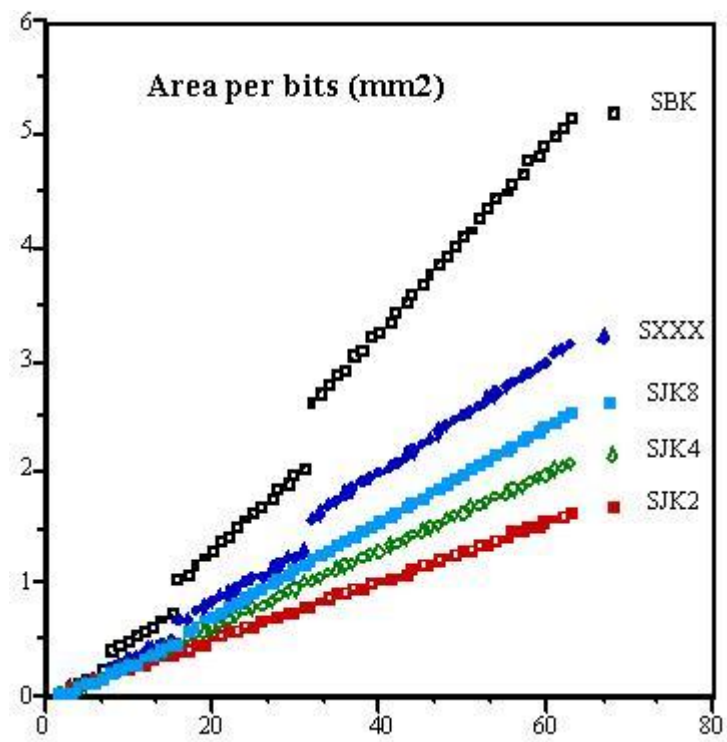
- The signal travels along a wire in a time proportional to its length.

This assumption is very important as we discuss it with an example. Let us consider the 16-bit BK adder (Figure 6.22) and the 16-bit JK4 adder (Figure 6.24). The longest wire in the BK implementation will be equal to at least eight widths of  $\Delta$  cells, whereas in the JK4 implementation, the longest wire will be equal to four widths of  $\Delta$ -cells. For BK, the output capacitive load of a  $\Delta$ -cell will be variable and a variable sizing of the cell will be necessary. In our case, the parameter  $m$  will define a fixed library of  $\Delta$ -cells used in the adder. The capacitive load will always be limited to a fixed value allowing all  $\Delta$  cells to be sized to a fixed value.



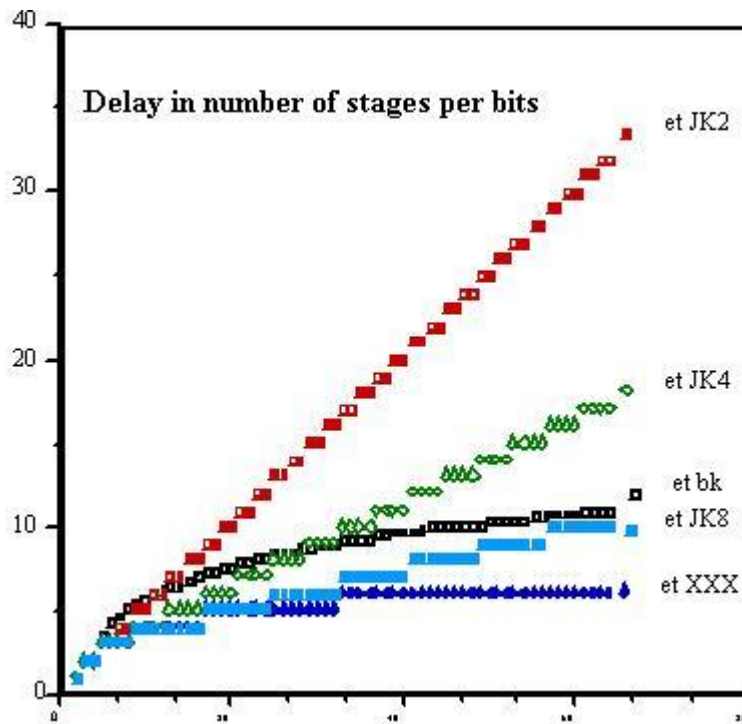


**Figure-6.27:** Number of transistors versus the number of bits



**Figure-6.28:** Area versus the number of bits





**Figure-6.29:** Delay in number of  $\Delta$  stages versus the number of bits in the adder

To partially conclude this section, we say that an optimum must be defined when choosing to implement our algorithm. This optimum will depend on the application for which the operator is to be used.



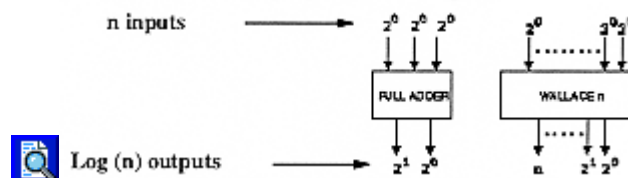
## 6.6 Multioperand Adders

### 6.6.1 General Principle

The goal is to add more than 2 operand in a time. This generally occurs in multiplication operation or filtering.

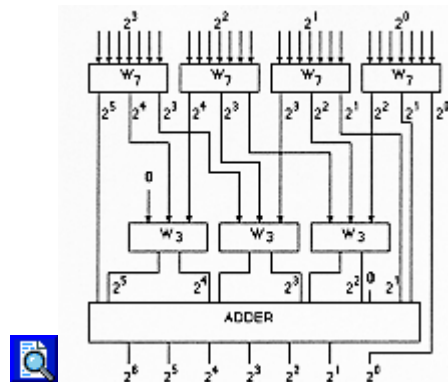
### 6.6.2 Wallace Trees

For this purpose, Wallace trees were introduced. The addition time grows like the logarithm of the bit number. The simplest Wallace tree is the adder cell. More generally, an  $n$ -inputs Wallace tree is an  $n$ -input operator and  $\log_2(n)$  outputs, such that the value of the output word is equal to the number of “1” in the input word. The input bits and the least significant bit of the output have the same weight (Figure 6.30). An important property of Wallace trees is that they may be constructed using adder cells. Furthermore, the number of adder cells needed grows like the logarithm  $\log_2(n)$  of the number  $n$  of input bits. Consequently, Wallace trees are useful whenever a large number of operands are to add, like in multipliers. In a Braun or Baugh-Wooley multiplier with a Ripple Carry Adder, the completion time of the multiplication is proportional to twice the number  $n$  of bits. If the collection of the partial products is made through Wallace trees, the time for getting the result in a carry save notation should be proportional to  $\log_2(n)$ .



[\[Click to enlarge image\]](#) **Figure-6.30:** Wallace cells made of adders

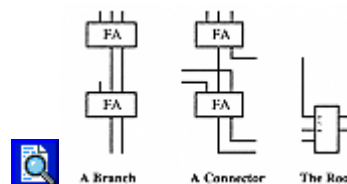
Figure 6.31 represents a 7-inputs adder: for each weight, Wallace trees are used until there remains only two bits of each weight, as to add them using a classical 2-inputs adder. When taking into account the regularity of the interconnections, Wallace trees are the most irregular.



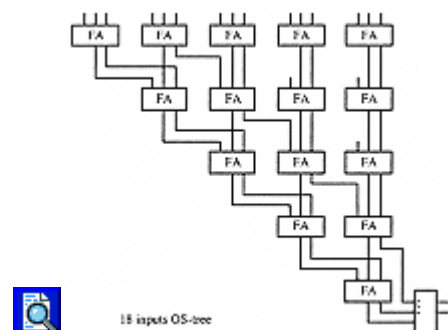
[\[Click to enlarge image\]](#) **Figure-6.31:** A 7-inputs Wallace tree

### 6.6.3 Overturned Stairs Trees

To circumvent the irregularity Mou [Mou91] proposes an alternative way to build multi-operand adders. The method uses basic cells called branch, connector or root. These basic elements (see Figure 6.32) are connected together to form n-input trees. One has to take care about the weight of the inputs. Because in this case the weights at the input of the 18-input OS tree are different. The regularity of this structure is better than with Wallace trees but the construction of multipliers is still complex.



[\[Click to enlarge image\]](#) **Figure-6.32:** Basic cells used to build OS-trees



[\[Click to enlarge image\]](#) **Figure-6.33:** A 18-input OS-tree

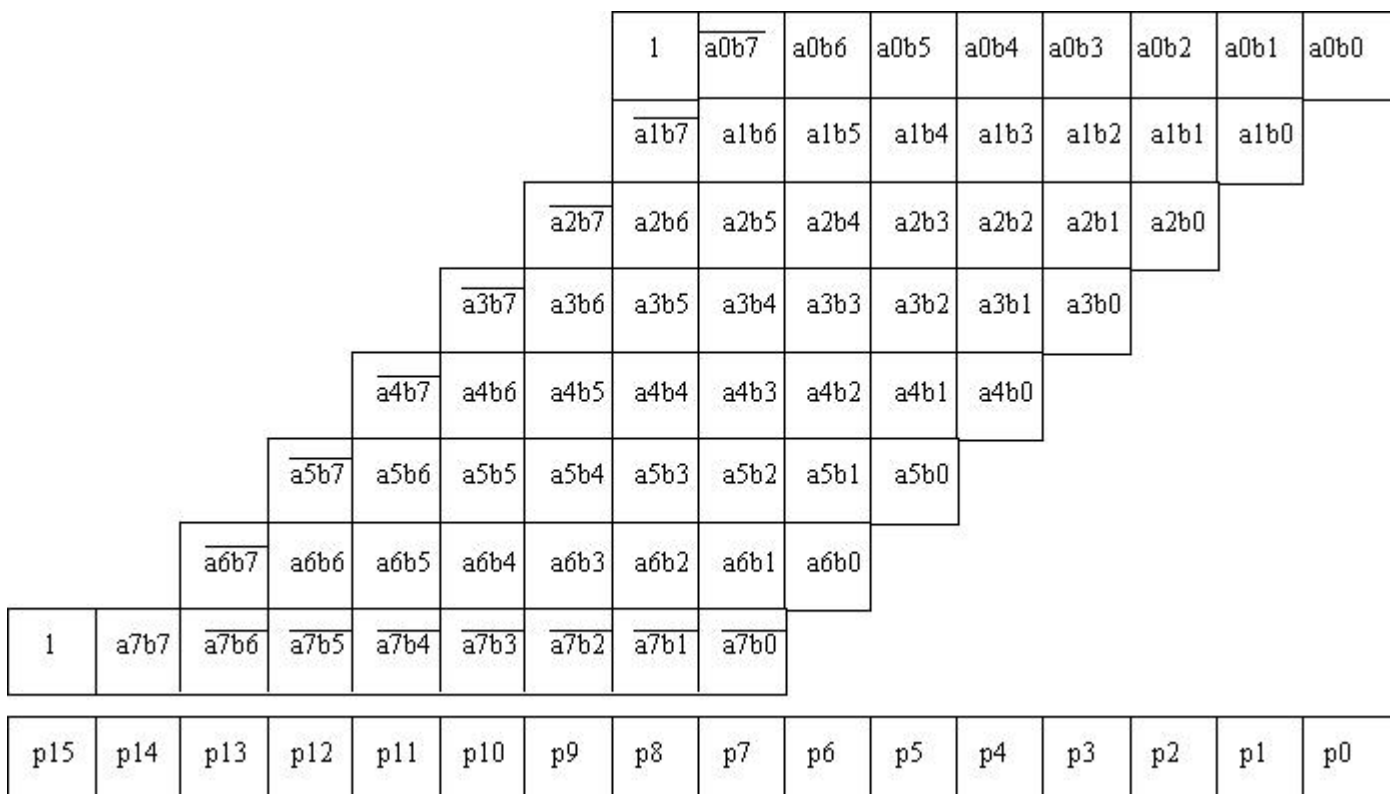


## 6.7 Multiplication

### 6.7.1 Introduction

Multiplication can be considered as a series of repeated additions. The number to be added is the multiplicand, the number of times that it is added is the multiplier, the result is the product. Each step of the addition generates a partial product. In most computers, the operands usually contain the same number of bits. When the operands are interpreted as integers, the product is generally twice the length of the operands in order to preserve the information content. This repeated addition method that is suggested by the arithmetic definition is slow that it is almost always replaced by an algorithm that makes use of positional number representation.

It is possible to decompose multipliers in two parts. The first part is dedicated to the generation of partial products, and the second one collects and adds them. As for adders, it is possible to enhance the intrinsic performances of multipliers. Acting in the generation part, the Booth (or modified Booth) algorithm is often used because it reduces the number of partial products. The collection of the partial products can then be made using a regular array, a Wallace tree or a binary tree [Sinh89].



**Figure-6.34:** Partial product representation and multioperand addition

### 6.7.2 Booth Algorithm

This algorithm is a powerful direct algorithm for signed-number multiplication. It generates a  $2n$ -bit product and treats both positive and negative numbers uniformly. The idea is to reduce the number of additions to perform. Booth algorithm allows in the best case  $n/2$  additions whereas modified Booth algorithm allows always  $n/2$  additions.

Let us consider a string of  $k$  consecutive 1s in a multiplier:

...,  $i+k$ ,  $i+k-1$ ,  $i+k-2$  , ...,  $i$ ,  $i-1$ , ...

..., 0 , 1 , 1 , ..., 1, 0, ...

where there is k consecutive 1s.

By using the following property of binary strings:

$$2^{i+k} - 2^i = 2^{i+k-1} + 2^{i+k-2} + \dots + 2^{i+1} + 2^i$$

the k consecutive 1s can be replaced by the following string

..., i+k+1, i+k, i+k-1, i+k-2, ..., i+1, i, i-1, ...  
 ..., 0, 1, 0, 0, ..., 0, -1, 0, ...  
 k-1 consecutive 0s Addition Subtraction

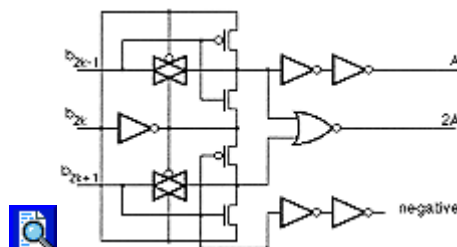
In fact, the modified Booth algorithm converts a signed number from the standard 2's-complement radix into a number system where the digits are in the set  $\{-1, 0, 1\}$ . In this number system, any number may be written in several forms, so the system is called redundant.

The coding table for the modified Booth algorithm is given in Table 8. The algorithm scans strings composed of three digits. Depending on the value of the string, a certain operation will be performed.

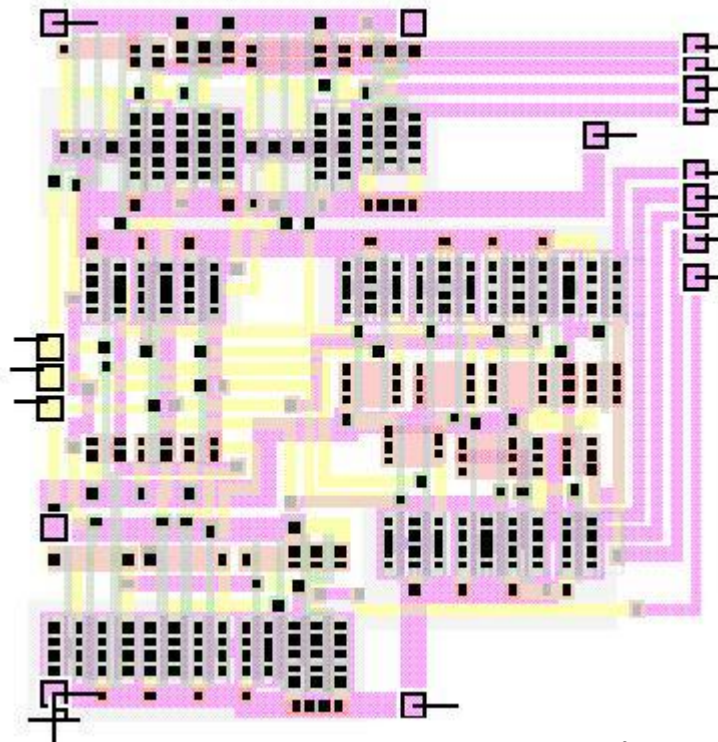
A possible implementation of the Booth encoder is given on Figure 6.35. The layout of another possible structure is given on Figure 6.36.

BIT			OPERATION	M is
$2^1$	$2^0$	$2^{-1}$		multiplied
$Y_{i+1}$	$Y_i$	$Y_{i-1}$		by
0	0	0	add zero (no string)	+0
0	0	1	add multipleic (end of string)	+X
0	1	0	add multiplic. (a string)	+X
0	1	1	add twice the mul. (end of string)	+2X
1	0	0	sub. twice the m. (beg. of string)	-2X
1	0	1	sub. the m. (-2X and +X)	-X
1	1	0	sub. the m. (beg. of string)	-X
1	1	1	sub. zero (center of string)	-0

**Table-6.8:** Modified Booth coding table.



[\[Click to enlarge image\]](#) **Figure-6.35:** Booth encoder cell



**Figure-6.36:** Booth encoder cell (layout size:  $65.70 \mu\text{m}^2$  (0.5 $\mu\text{CMOS}$ ))

### 6.7.3 Serial-Parallel Multiplier

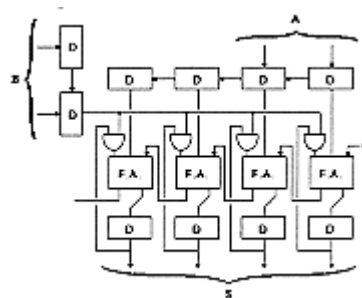
This multiplier is the simplest one, the multiplication is considered as a succession of additions. If  $A = (a_n a_{n-1} \dots a_0)$  and  $B = (b_n b_{n-1} \dots b_0)$

The product  $A.B$  is expressed as :

$$A.B = A.2^n.b_n + A.2^{n-1}.b_{n-1} + \dots + A.2^0.b^0$$

The structure of Figure 6.37 is suited only for positive operands. If the operands are negative and coded in 2's-complement :

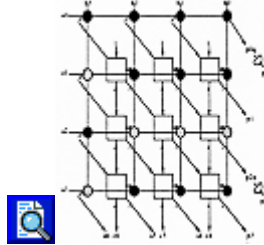
1. The most significant bit of  $B$  has a negative weight, so a subtraction has to be performed at the last step.
2. Operand  $A.2^k$  must be written on  $2N$  bits, so the most significant bit of  $A$  must be duplicated. It may be easier to shift the content of the accumulator to the right instead of shifting  $A$  to the left.



[\[Click to enlarge image\]](#) **Figure-6.37:** Serial-Parallel multiplier

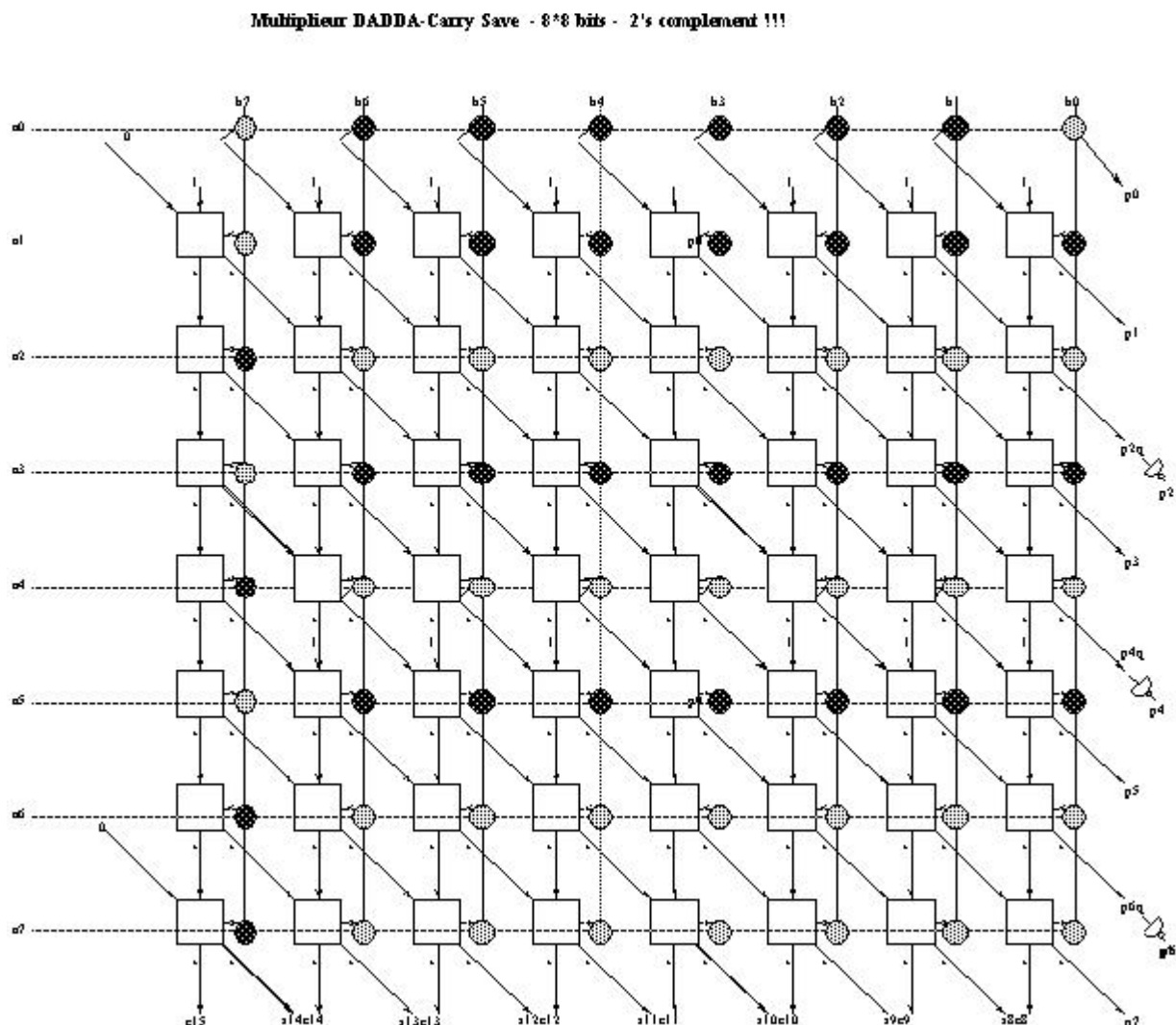
### 6.7.4 Braun Parallel Multiplier

The simplest parallel multiplier is the Braun array. All the partial products  $A \cdot b_k$  are computed in parallel, then collected through a cascade of Carry Save Adders. At the bottom of the array, the output of the array is noted in Carry Save, so an additional adder converts it (by the mean of a carry propagation) into the classical notation (Figure 6.38). The completion time is limited by the depth of the carry save array, and by the carry propagation in the adder. Note that this multiplier is only suited for positive operands. Negative operands may be multiplied using a Baugh-Wooley multiplier.



[\[Click to enlarge image\]](#) **Figure-6.38:** A 4-bit Braun Multiplier without the final adder

Figure 6.38 and Figure 6.40 use the symbols given in Figure 6.39 where CMUL1 and CMUL2 are two generic cells consisting of an adder without the final inverter and with one input connected to an AND or NAND gate. A non optimised (in term of transistors) multiplier would consist only of adder cells connected one to another with AND gates generating the partial products. In these examples, the inverters at the output of the adders have been eliminated and the parity of the bits has been compensated by the use of CMUL1 or CMUL2.





**Figure-6.40:** A 8-bit Braun Multiplier without the final adder

### 6.7.5 Baugh-Wooley Multiplier

This technique has been developed in order to design regular multipliers, suited for 2's-complement numbers.

Let us consider 2 numbers A and B :

$$\begin{aligned} A &= (a_{n-1} \dots a_0) = -a_{n-1} \cdot 2^{n-1} + \sum_{i=0}^{n-2} a_i \cdot 2^i \\ B &= (b_{n-1} \dots b_0) = -b_{n-1} \cdot 2^{n-1} + \sum_{i=0}^{n-2} b_i \cdot 2^i \end{aligned} \quad (64), (65)$$

The product A.B is given by the following equation :

$$A \cdot B = a_{n-1} \cdot b_{n-1} \cdot 2^{2n-2} + \sum_{i=0}^{n-2} \sum_{j=0}^{n-2} a_i \cdot b_j \cdot 2^{i+j} - a_{n-1} \sum_{i=0}^{n-2} b_i \cdot 2^{n+i-1} - b_{n-1} \sum_{i=0}^{n-2} a_i \cdot 2^{n+i-1} \quad (66)$$

We see that subtractor cells must be used. In order to use only adder cells, the negative terms may be rewritten as :

$$-a_{n-1} \sum_{i=0}^{n-2} b_i \cdot 2^{i+n-1} = a_{n-1} \cdot \left( -2^{2n-2} + 2^{n-1} + \sum_{i=0}^{n-2} \overline{b_i} \cdot 2^{i+n-1} \right) \quad (67)$$

By this way, A.B becomes :

$$\begin{aligned} A \cdot B &= a_{n-1} \cdot b_{n-1} \cdot 2^{2n-2} + \sum_{i=0}^{n-2} \sum_{j=0}^{n-2} a_i \cdot b_j \cdot 2^{i+j} \\ &\quad + b_{n-1} \left[ -2^{2n-2} + 2^{n-1} + \sum_{i=0}^{n-2} \overline{a_i} \cdot 2^{i+n-1} \right] \\ &\quad + a_{n-1} \left[ -2^{2n-2} + 2^{n-1} + \sum_{i=0}^{n-2} \overline{b_i} \cdot 2^{i+n-1} \right] \end{aligned} \quad (68)$$

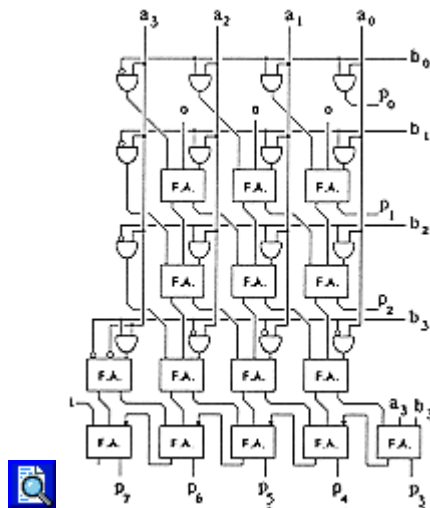
The final equation is :

$$\begin{aligned}
 A \cdot B &= -2^{2n-1} + (\overline{a_{n-1}} + \overline{b_{n-1}} + a_{n-1} \cdot b_{n-1}) \cdot 2^{2n-2} \\
 &+ \sum_{i=0}^{n-2} \sum_{j=0}^{n-2} a_i \cdot b_j \cdot 2^{i+j} + (a_{n-1} + b_{n-1}) \cdot 2^{n-1} \\
 &+ \sum_{i=0}^{n-2} b_{n-1} \cdot \overline{a_i} \cdot 2^{i+n-1} + \sum_{i=0}^{n-2} a_{n-1} \cdot \overline{b_i} \cdot 2^{i+n-1}
 \end{aligned} \tag{69}$$

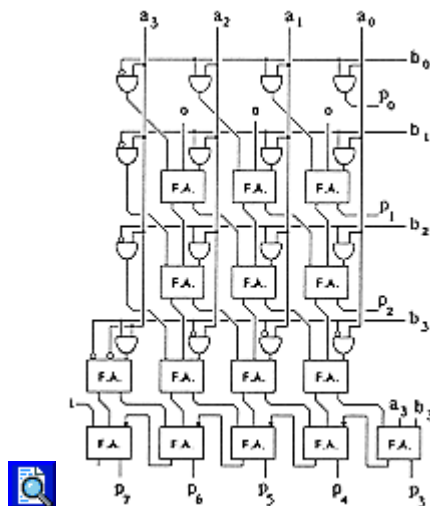
because :

$$- (b_{n-1} + a_{n-1}) \cdot 2^{2n-2} = -2^{2n-1} + (\overline{a_{n-1}} + \overline{b_{n-1}}) \cdot 2^{2n-2} \tag{70}$$

A and B are n-bits operands, so their product is a 2n-bits number. Consequently, the most significant weight is  $2^{2n-1}$ , and the first term  $-2^{2n-1}$  is taken into account by adding a 1 in the most significant cell of the multiplier.



[\[Click to enlarge image\]](#) **Figure-6.41:** shows a 4-bits Baugh-Wooley multiplier

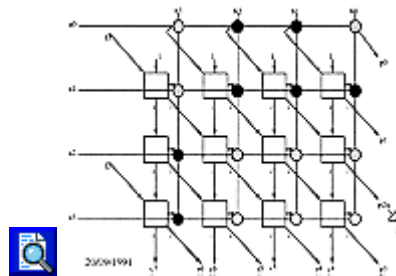


[\[Click to enlarge image\]](#) **Figure-6.41:** A 4-bit Baugh-Wooley Multiplier with the final adder



## 6.7.6 Dadda Multiplier

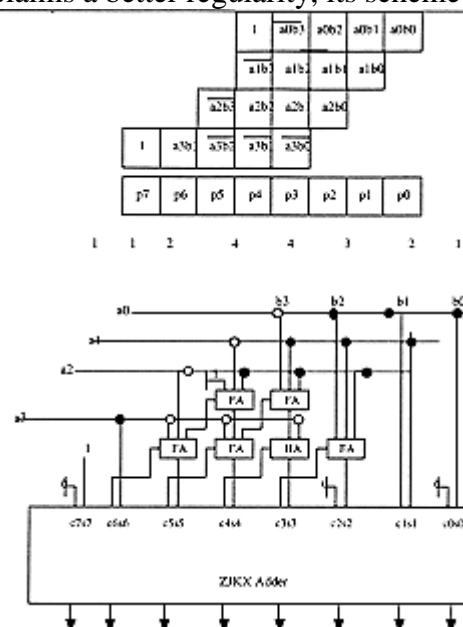
The advantage of this method is the higher regularity of the array. Signed integers can be processed. The cost for this regularity is the addition of an extra column of adders.



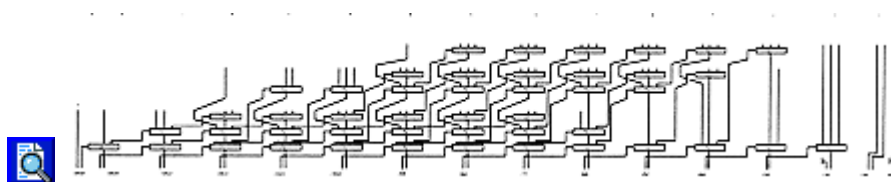
[\[Click to enlarge image\]](#) **Figure-6.42:** A 4-bit Baugh-Wooley Multiplier with the final adder

## 6.7.7 Mou's Multiplier

On Figure 6.43 the scheme using OS-trees is used in a 4-bit multiplier. The partial product generation is done according to Dadda multiplication. Figure 6.44 represents the OS-tree structure used in a 16-bit multiplier. Although the author claims a better regularity, its scheme does not allow an easy pipelining.



[\[Click to enlarge image\]](#) **Figure-6.43:** A 4-bit OS-tree Multiplier with a final adder



[\[Click to enlarge image\]](#) **Figure-6.44:** A 16-bit OS-tree Multiplier without a final adder and without the partial product cells

## 6.7.8 Logarithmic Multiplier

The objective of this circuit is to compute the product of two terms. The property used is the following equation :

$$\text{Log}(A * B) = \text{Log}(A) + \text{Log}(B) \quad (71)$$

There are several ways to obtain the logarithm of a number : look-up tables, recursive algorithms or the segmentation of the logarithmic curve [Hoef91]. The segmentation method : The basic idea is to approximate the logarithm curve with a set of linear segments.

If  $y = \text{Log}_2(x)$  (72)

an approximation of this value on the segment  $]2^{n+1}, 2^n[$  can be made using the following equation :

$$y = ax + b = (\Delta y / \Delta x) \cdot x + b = [1 / (2^{n+1} - 2^n)] \cdot x + n - 1 = 2^{-n} x + (n - 1) \quad (73)$$

What is the hardware interpretation of this formula?

If we take  $x_i = (x_{i7}, x_{i6}, x_{i5}, x_{i4}, x_{i3}, x_{i2}, x_{i1}, x_{i0})$ , an integer coded with 8 bits, its logarithm will be obtained as follows. The decimal part of the logarithm will be obtained by shifting  $x_i$   $n$  positions to the right, and the integer part will be the value where the MSB occurs.

For instance if  $x_i$  is  $(0,0,1,0,1,1,1,0) = 46$ , the integer part of the logarithm is 5 because the MSB is  $x_{i5}$  and the decimal part is 01110. So the logarithm of  $x_i$  equals  $101.01110 = 5.4375$  because 01110 is 14 out of a possible 32, and  $14/32 = 0.4375$

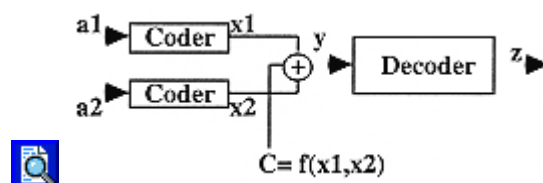
Table 9 illustrates this coding. Once the coding of two linear words has been performed, the addition of the two logarithms can be done. The last operation to be performed is the antilogarithm of the sum to obtain the value of the final product.

Using this method, a 11.6% error on the product of two binary operands (i.e. the sum of two logarithmic numbers) occurs. We would like to reduce this error without increasing the complexity of the operation nor the complexity of the operator. Since the transformations used in this system are logarithms and antilogarithms, it is natural to think that the complexity of the correction systems will grow exponentially if the error approaches zero. We analyze the error to derive an easy and effective way to increase the accuracy of the result.

Xi								Log(x)						Log(Xi)	
	7	6	5	4	3	2	1	0	Intege		Decimal part				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
3	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0
4	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0
6	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0
7	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0
8	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0
9	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0
10	0	0	0	0	1	0	1	0	0	1	1	0	0	0	0
11	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0
12	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0
13	0	0	0	0	1	1	0	1	0	1	1	0	0	0	0
14	0	0	0	0	1	1	1	0	0	1	1	0	0	0	0
15	0	0	0	0	1	1	1	1	0	1	1	1	0	0	0
16	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0
17	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0
18	0	0	0	1	0	0	1	0	1	0	0	1	0	0	0
19	0	0	0	1	0	0	1	1	1	0	0	1	1	0	0
20	0	0	0	1	0	1	0	0	1	0	0	0	1	0	0
21	0	0	0	1	0	1	0	1	1	0	0	0	1	0	0
22	0	0	0	1	0	1	1	0	1	0	0	0	1	1	0
23	0	0	0	1	0	1	1	1	1	0	0	0	1	1	0
24	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0
25	0	0	0	1	1	0	0	1	1	0	0	1	0	0	0

**Table-6.9:** Coding of the binary logarithm according to the segmentation method

Figure 6.45 describes the architecture of the logarithmic multiplier with the different variables used in the system.



[\[Click to enlarge image\]](#) **Figure-6.45:** Block diagram of a logarithmic multiplier

**Error analysis:** Let us define the different functions used in this system.

Relative error of the coder ( $<0$ ):

$$\text{The logarithm of } x: \quad x = \text{Log}_2(a) \quad (74)$$

$$\text{Approximated Log}_2: \quad \tilde{x} = 2^{-n} a + n - 1 \quad (75)$$

$$\text{Relative error:} \quad \varepsilon_{\text{cod}} = \frac{\tilde{x} - x}{x} = \frac{\tilde{x}}{x} \quad (76)$$

$$\varepsilon_{\text{cod}} = \frac{2^{-n} a + n - 1 - \text{Log}_2(a)}{\text{Log}_2(a)} \quad (77)$$

Relative error of the decoder ( $>0$ ):

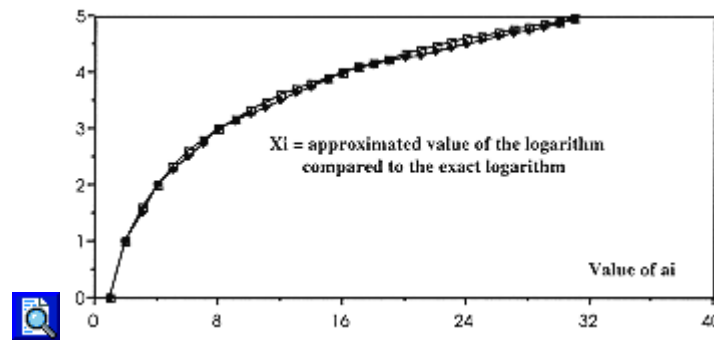
$$\text{Two to the power of } y: \quad z = 2^y \quad (78)$$

$$\text{Approximated value:} \quad \tilde{z} = 2^n (y - n + 1) \quad (79)$$

$$\text{Relative error:} \quad \varepsilon_{\text{dec}} = \frac{\tilde{z} - z}{z} = \frac{\tilde{z}}{z} \quad (80)$$

$$\varepsilon_{\text{dec}} = \frac{2^n (y - n + 1) - 2^y}{2^y}$$

The logarithm and antilogarithm curves are approximated by linear segments. They start at values which are in powers-of-two and end at the next power-of-two value. Figure 6.46 shows how a logarithm is approximated. The same is true for the antilogarithm.



[\[Click to enlarge image\]](#) **Figure-6.46:** Approximated value of the logarithm compared to the exact logarithm

By adding the unique value  $17 \cdot 2^{-8}$  to the two logarithms an improvement of 40% is achieved on the maximum error. The maximum error comes down from 11.6% to 7.0%, an improvement of 40% compared with a system without any correction. The only cost is the replacement of the internal two input adder by a three input adder.

A more complex correction system which leads to better precision but at a much higher hardware cost is possible.

In Table 10 we suggest a system which would choose one correction among three depending on the value of the input bits. Table 10 can be read as the values of the logarithms obtained after the coder for either  $a_1$  or  $a_2$ . The penultimate column represents the ideal correction which should be added to get 100% accuracy. The last column gives the correction chosen among three possibilities: 32, 16 or 0.

Three decoding functions have to be implemented for this proposal. If the exclusive -OR of  $a_2$  and  $a_3$  is true, then the added value is  $32 \cdot 2^{-8}$ . If all the bits of the decimal part are zero, then the added value is zero. In all other cases the added value is  $16 \cdot 2^{-8}$ .

This decreases the average error. But the drawback is that the maximum error will be minimized only if the steps between two ideal corrections are bigger than the unity step. To minimize the maximum error the correcting functions should increase in an exponential way. Further research could be performed in this area.

ai									log(ai)								Ideal V value	New Affected V value
	7	6	5	4	3	2	1	0	Intege				Decimal part					
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NUM	NUM
1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	32	32
4	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	16	16
6	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	32	32
7	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	8	16
8	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0	4	16
10	0	0	0	0	1	0	1	0	0	1	1	0	1	0	0	0	16	16
11	0	0	0	0	1	0	1	1	0	1	1	0	1	1	0	0	36	32
12	0	0	0	0	1	1	0	0	0	1	1	1	0	0	0	0	32	32
13	0	0	0	0	1	1	0	1	0	1	1	1	0	1	0	0	18	16
14	0	0	0	0	1	1	1	0	0	1	1	1	1	0	0	0	8	16
15	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0	2	0
16	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
17	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0
18	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	4	16
19	0	0	0	1	0	0	1	1	0	0	0	1	1	0	0	0	9	16
20	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	16	16
21	0	0	0	1	0	1	0	1	0	0	1	0	1	0	0	0	25	16
22	0	0	0	1	0	1	1	0	0	1	0	0	1	0	0	0	36	32
23	0	0	0	1	0	1	1	1	0	0	1	1	1	0	0	0	39	32
24	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	32	32
25	0	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	24	32

**Table-6.10:** A more complex correction scheme



## 6.8 Addition and Multiplication in Galois Fields, GF(2n)

The group theory is used to introduce another algebraic system, called a field. A field is a set of elements in which we can do addition, subtraction, multiplication and division without leaving the set. Addition and multiplication must satisfy the commutative, associative, and distributive laws. A formal definition of a field is given below.

### Definition

Let F be a set of elements on which two binary operations called addition "+" and multiplication ".", are defined. The set F together with the two binary operations + and . is a field if the following conditions are satisfied:

1. F is a commutative group under addition +. The identity element with respect to addition is called the zero element or the additive identity of F and is denoted by 0.
2. The set of nonzero elements in F is a commutative group under multiplication . .The identity element with respect to multiplication is called the unit element or the multiplicative identity of F and is denoted 1.
3. Multiplication is distributive over addition; that is, for any three elements, a, b, c in F:

$$a \cdot (b + c) = a \cdot b + a \cdot c$$

The number of elements in a field is called the order of the field.

A field with finite number of elements is called a finite field.

Let us consider the set {0,1} together with modulo-2 addition and multiplication. We can easily check that the set {0,1} is a field of two elements under modulo-2 addition and modulo-2 multiplication. field is called a binary field and is denoted by GF(2).

The binary field GF(2) plays an important role in coding theory [Rao74] and is widely used in digital computers and data transmission or storage systems.

Another example using the residue to the base [Garn59] is given below. Table 11 represents the values of N, from 0 to 29 with their representation according to the residue of the base (5, 3, 2).The addition and multiplication of two term in this base can be performed according to the next example:

res. to base											
N	5	3	2	N	5	3	2	N	5	3	2
0	0	0	0	10	0	1	0	20	0	2	0
1	1	1	1	11	1	2	2	21	1	0	1
2	2	2	0	12	2	0	0	22	2	1	0
3	3	0	1	13	3	1	1	23	3	2	1
4	4	1	0	14	4	2	0	24	4	0	0
5	0	2	1	15	0	0	1	25	0	1	1
6	1	0	0	16	1	1	0	26	1	2	0
7	2	1	1	17	2	2	1	27	2	0	1
8	3	2	0	18	3	0	0	28	3	1	0
9	4	0	1	19	4	1	1	29	4	2	1

**Table-6.11:** N varying from 0 to 29 and its representation in the residue number system

ADDITION in 5,3,2 residue arithmetic :

9	4,0,1	8	3,2,0
+16	1,1,0	+19	4,1,1
25	0,1,1	27	2,0,1
decimal	residue	decimal	residue
	5,3,2		5,3,2

MULTIPLICATION in 5,3,2 residue arithmetic :

7	2,1,1
* 4	4,1,0
28	3,1,0
decimal	residue
	5,3,2

The most interesting property in these systems is that there is no carry propagation inside the set. This can be attractive when implementing into VLSI these operators



---

## References

- [Aviz61] Avizienis, Signed-Digit Number Representations For Fast Parallel Arithmetic, IRE Trans. Electron. Compute., Vol EC-10, pp. 389-400, 1961.
- [Cava83] J. J. F. Cavanagh, Digital Computer Arithmetic, McGraw-Hill computer sciences series, 1983.
- [Garn59] H. L. Garner. The Residue Number System, IRE Transactions on Elec. Comput., p. 140- 147, September 1959
- [Hoe91] B. Hoefflinger, M. Selzer, F. Warkowski, Digital Logarithmic CMOS Multiplier for Very- High Speed Signal Processing, in Proc. IEEE Custom Integrated Circuit Conference, 1991, pp.16.7.1-16.7.5.
- [Kowa92] J. Kowalczyk and D. Mlynek, Un Nouvel Algorithme De Generation D'Additionneurs Rapides Dédiés Au traitement d'images, Proceeding of the Industrial Automation Conference, p. 20.9-20.13, Montreal, Québec, Canada, June 1992.
- [Kowa93] J. Kowalczyk, On the Design and implementation of algorithms for Multimedia Systems, PhD Thesis No 1188, Swiss Federal Institute of Techn., Lausanne 1994.
- [Mull82] J.M. Muller, Arithmétique des ordinateurs, opérateurs et fonctions élémentaires, Masson 1989
- [Rao74] T. R. N. Rao, Error Coding for Arithmetic Processors, New York, Academic Press, 1974
- [Sinh89] B. P. Sinha and P. K. Srimani, Fast Parallel Algorithms for Binary Multiplication and Their Implementation on Systolic Architectures, IEEE Transactions on Computers, vol. 38, No. 3, 424-431, March 1989
- [Taka87] Y. Harata, Y. Nakamura, H. Nagase, M. Takigawa and N. Takagi, A High-Speed Multiplier Using a Redundant Binary Adder Tree, IEEE Journal of Solid States Circuits, Volume SC- 22, No. 1, Pages 28-34, February 1987



---

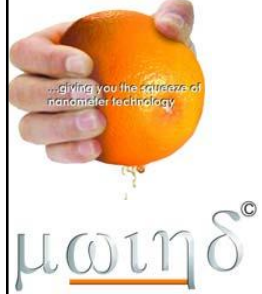
This chapter edited by D. Mlynek

a joint production of



**INTEGRATED SYSTEMS  
CENTER**





# An Illustration of $0.1\mu\text{m}$ CMOS layout design on PC





μωηδ<sup>®</sup>

# Agenda

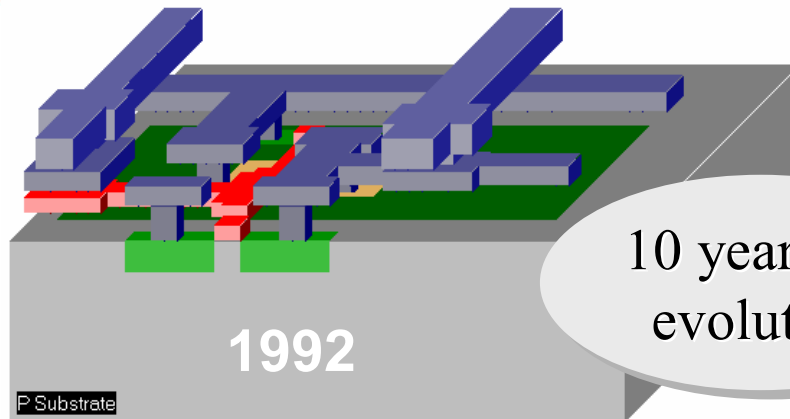
- The technology scale down
- Design trends
- The MOS device
- CMOS cell design
- Signal propagation
- Embedded Memory
- SOI

...giving you the squeeze of nanometer design technology

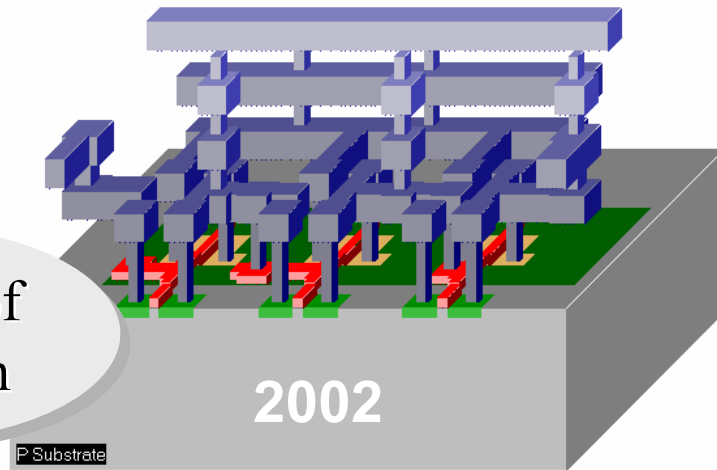
[www.microwind.net](http://www.microwind.net)



# 1. The technology scale down

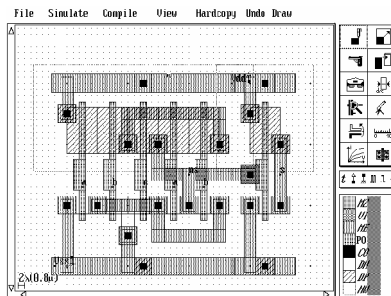


10 years of evolution



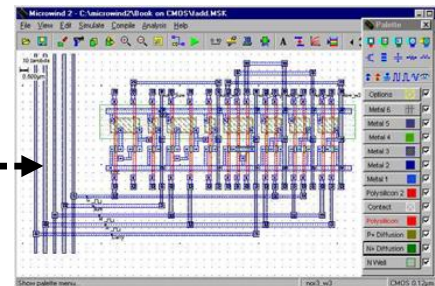
- ☺ 0.7 $\mu$ m, 2 metal layers
- ☺ Up to 100K transistors, 50MHz

- ☺ 0.12 $\mu$ m, 7 metal
- ☺ Up to 500MT, 1.5GHz



*Introduction to  $\mu$ -  
Electronics on PC*

**MSK, PROF, 3D**



*Introduction to CMOS  
design on PC*

**Microwind, Dsch**

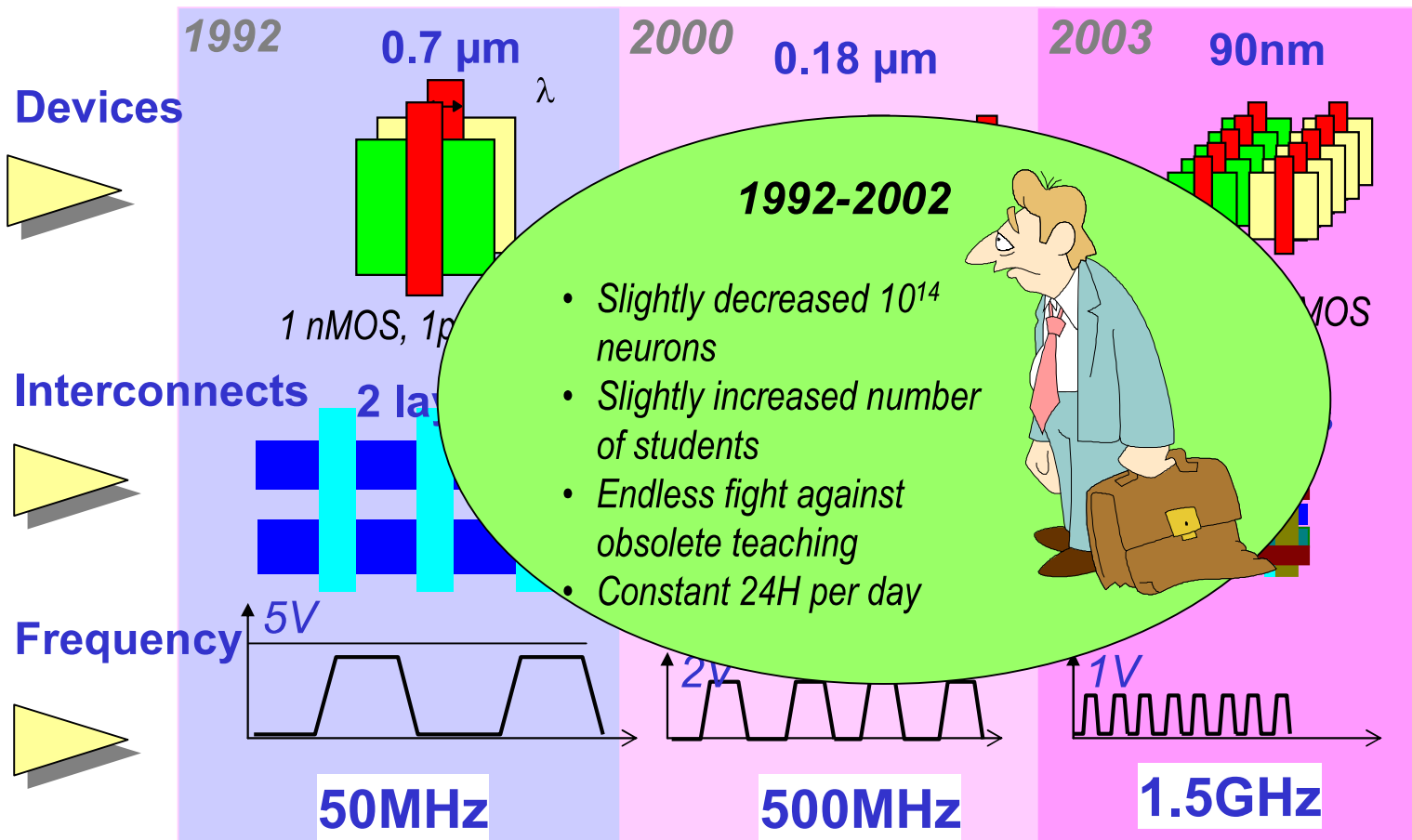
...giving you the squeeze of nanometer design technology

[www.microwind.net](http://www.microwind.net)



μωηδ<sup>©</sup>

# 1. The technology scale down

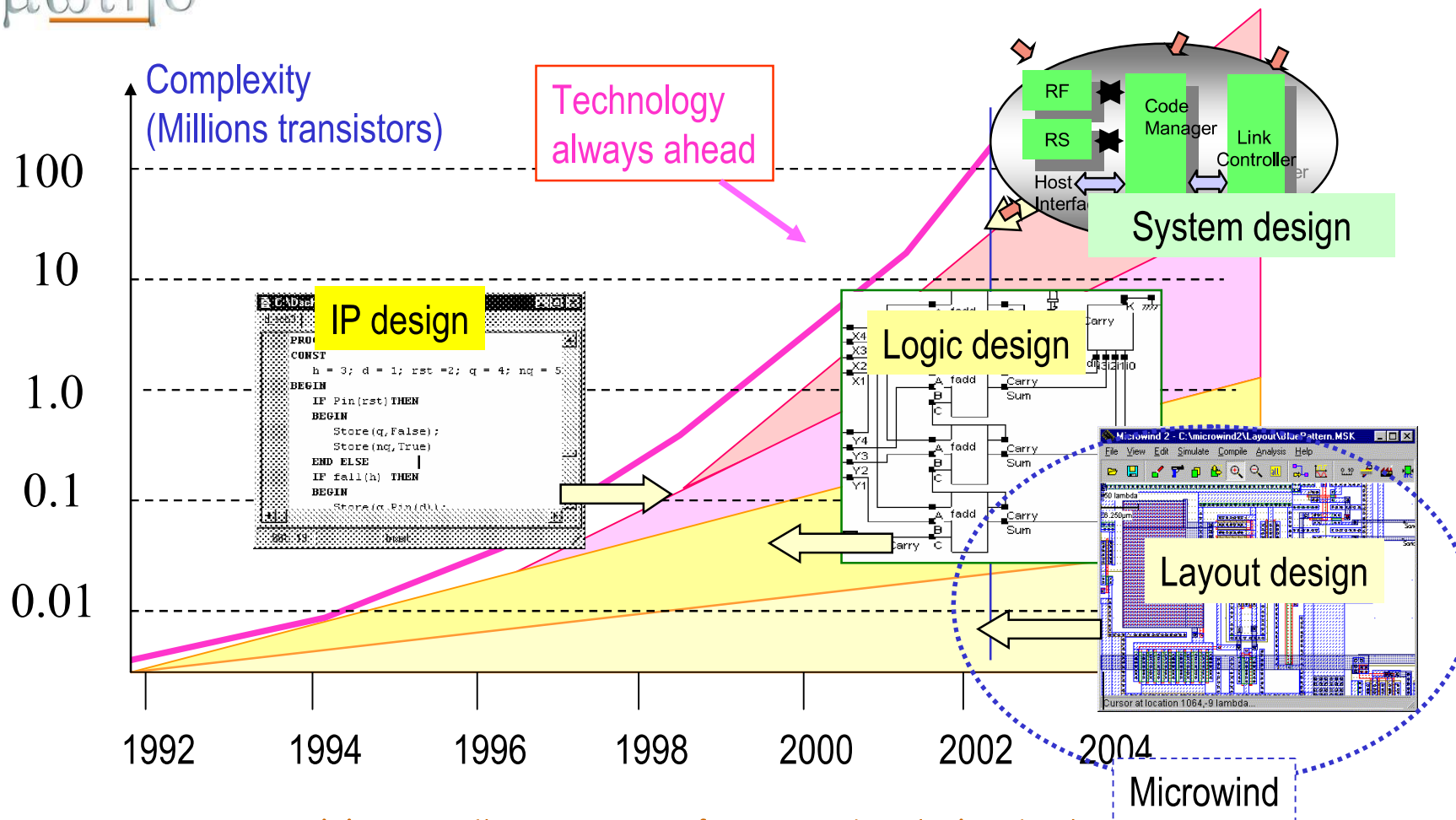


...giving you the squeeze of nanometer design technology



μωιηδ<sup>®</sup>

## 2. Design trends

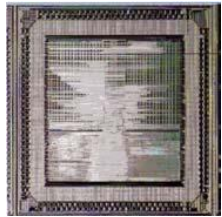


...giving you the squeeze of nanometer design technology

[www.microwind.net](http://www.microwind.net)



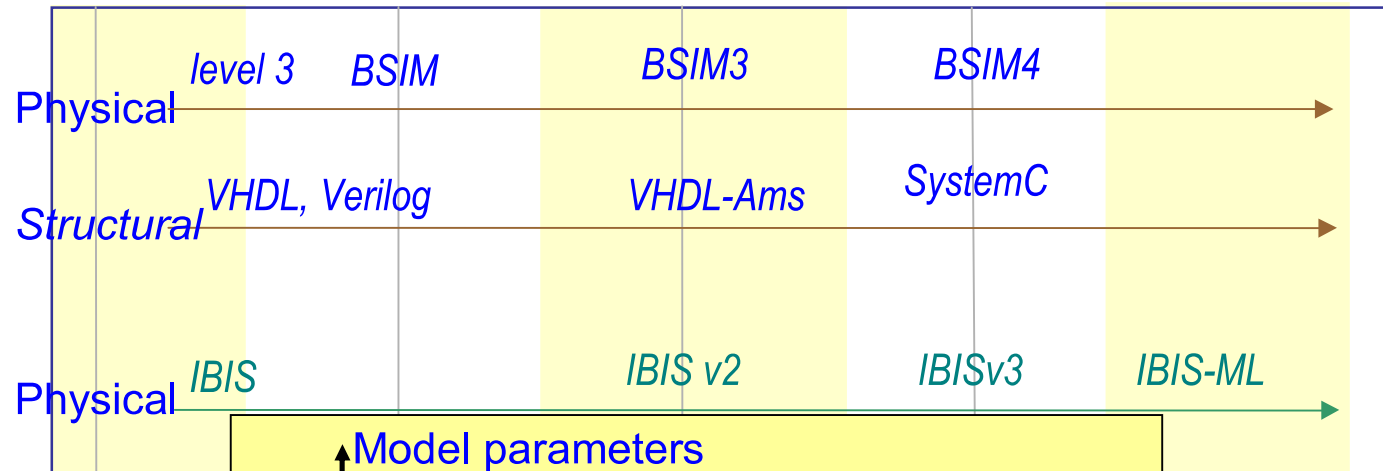
μωηδ<sup>©</sup>  
Core



Interface

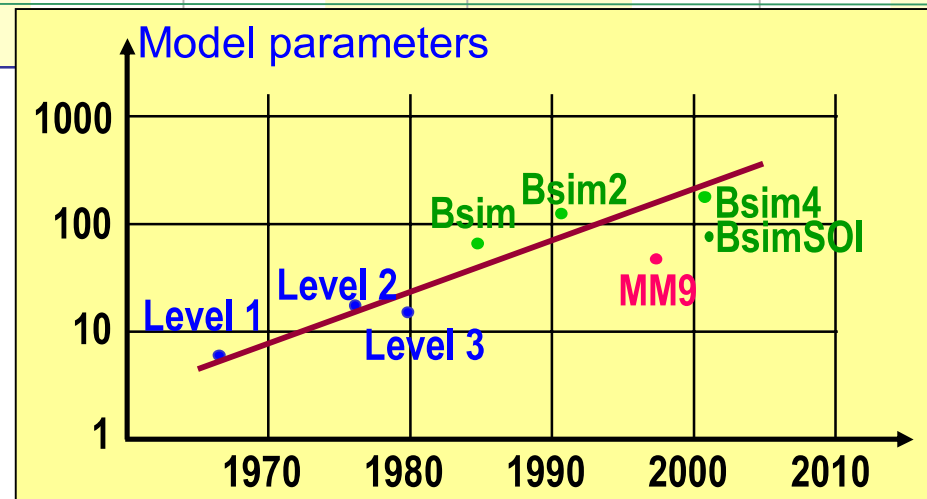


## 2. Design trends



1995

2003



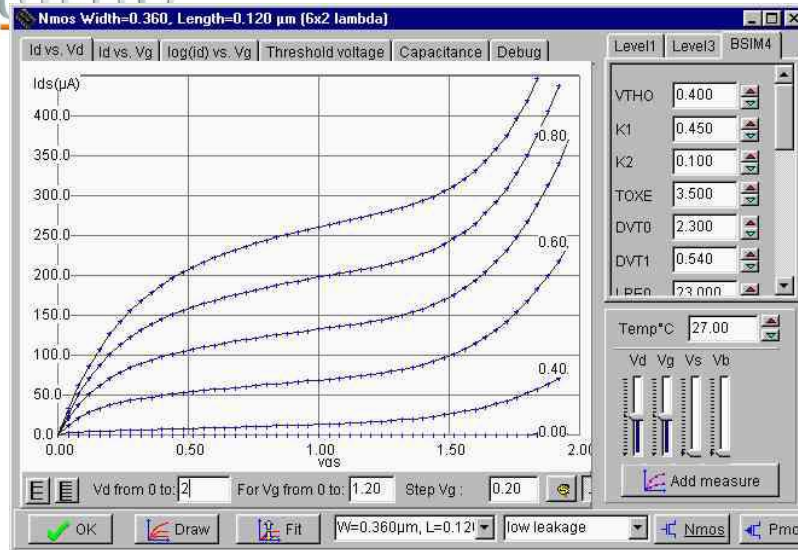
...giving you the squeeze of nanometer design technology

[www.microwind.net](http://www.microwind.net)

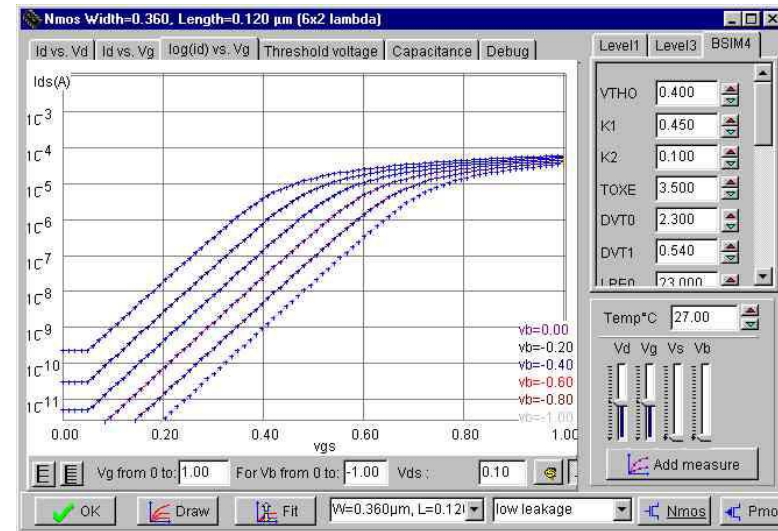


### 3. The MOS devices

microwind<sup>®</sup>



Dependence of Id vs. Length  
Impact ionization at high Vds



Important Ioff current for small Length  
Complex dependence of Vt vs. Length

$$v_{th} = V_{TH0} + K1 \cdot \sqrt{(\Phi_s - V_{bs}) - \sqrt{\Phi_s}} - K2 \cdot V_{bs} + \Delta V_{t_{SCE}} + \Delta V_{t_{NULD}} + \Delta V_{t_{DIBL}}$$

$$\Delta V_{t_{SCE}} = -\frac{0.5 \cdot DVT0}{\cosh(DVT1 \cdot \frac{L_{eff}}{l_t} - 1)} (V_{bi} - \Phi_s)$$

$$\Delta V_{t_{NULD}} = K1 \left( \sqrt{1 + \frac{LPE0}{L_{eff}}} - 1 \right) \cdot \sqrt{\Phi_s}$$

$$\Delta V_{t_{DIBL}} = -0.5 \cdot ETA0 \cdot V_{ds}$$

...giving you the squeeze of nanometer design technology

www.microwind.net

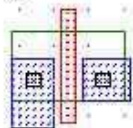


μωιηδ<sup>©</sup>

## 3. The MOS devices

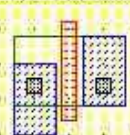
**Low Leakage**

Low leakage nMOS (default)



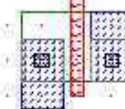
**Ultra High Speed**

Ultra high speed nMOS



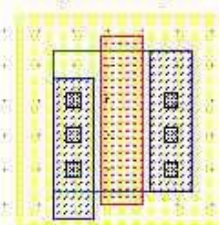
**EEProm**

EEProm MOS



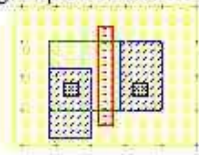
**High Voltage**

High voltage nMOS

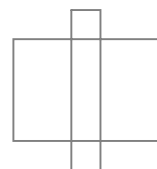


**High Speed**

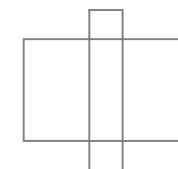
High speed nMOS device



**RF**



**MRam**



Application-oriented MOS device  
Same basic mechanism

New physical properties in EEPROM and MRam

...giving you the squeeze of nanometer design technology

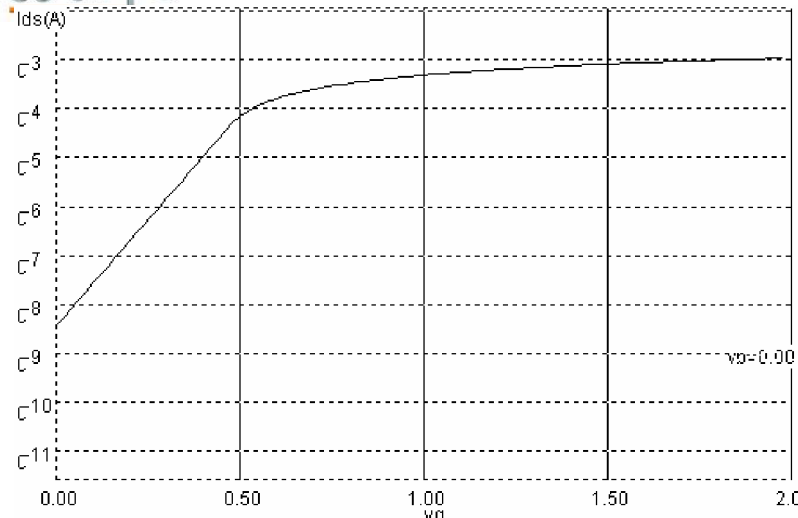
[www.microwind.net](http://www.microwind.net)





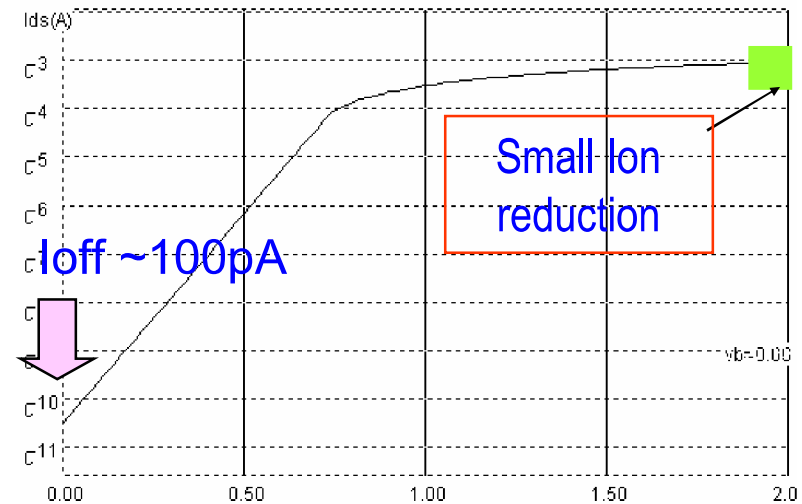
μωηδ°

### 3. The MOS devices



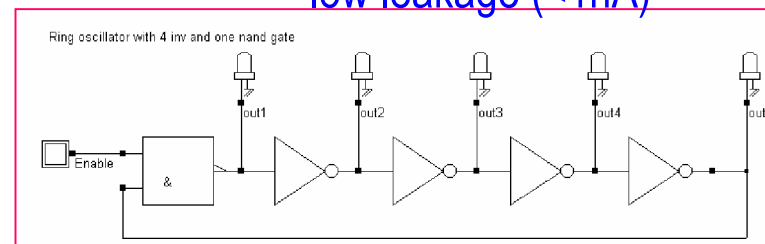
**High Speed**

High current MOS, low VT  
Shorter channel L=100nm,  
high leakage (Critical path)



**Low Leakage**

Default MOS device, high VT  
low leakage (<1nA)



...giving you the squeeze of nanometer design technology

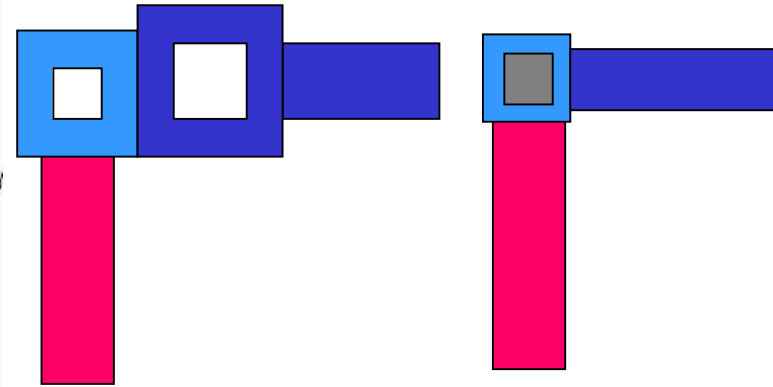
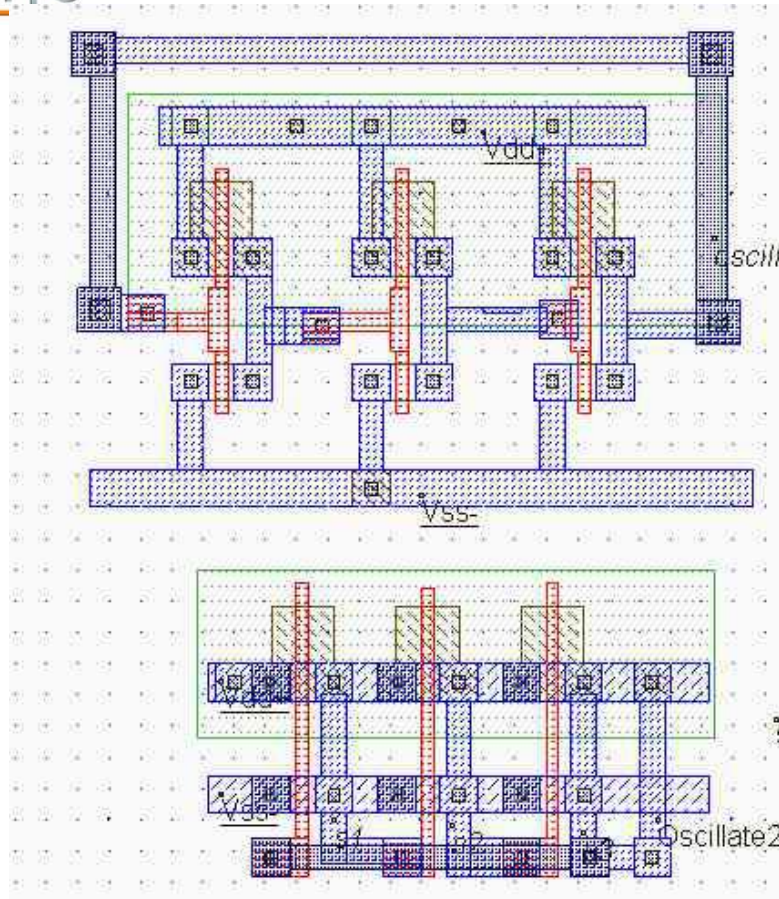
www.microwind.net





μωινδ<sup>©</sup>

## 4. CMOS cell design



Stacked vias  
Salicide/unsalicide (Large R)

*but...*

Antenna effects  
Contact parasitic effect (20  $\Omega$ )

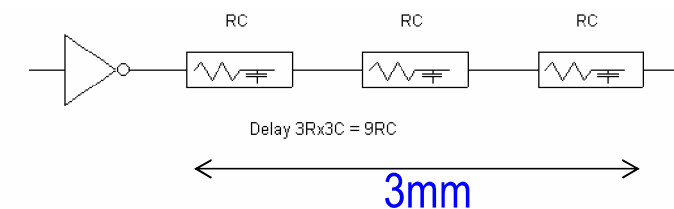
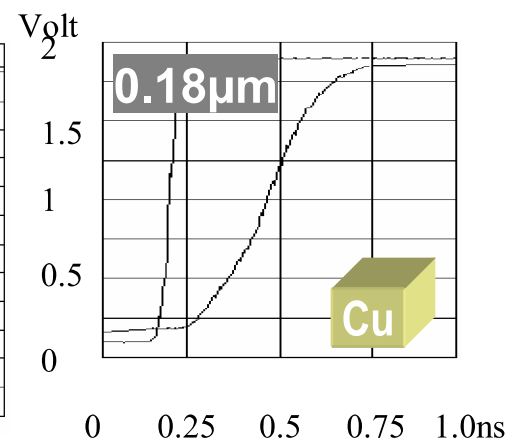
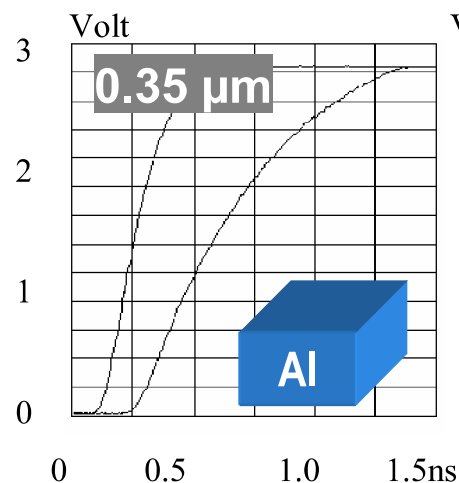
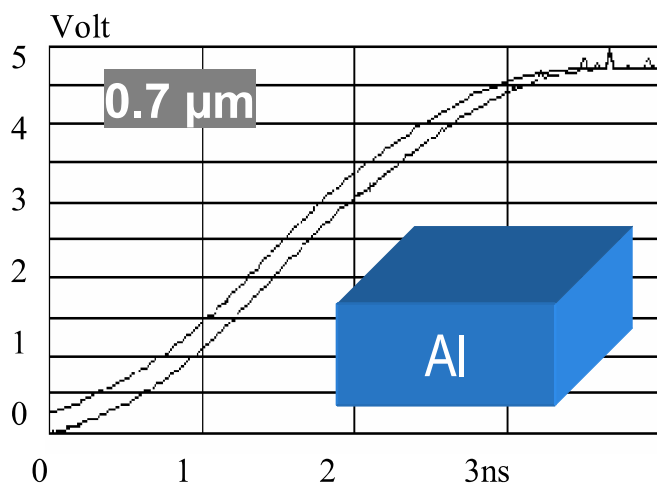
...giving you the squeeze of nanometer design technology

[www.microwind.net](http://www.microwind.net)

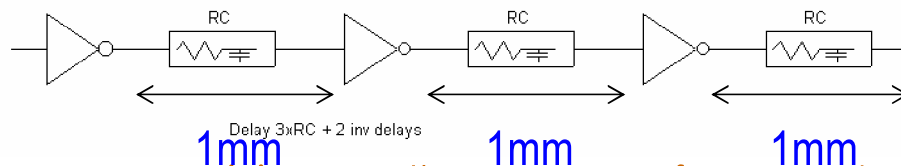


μωηδ°

## 5. Signal propagation



$3R \times 3C = 9RC$  (680ps) Repeaters help to propagate signals at long distance



$3RC + 2t_{\text{gate}}$  (380ps)

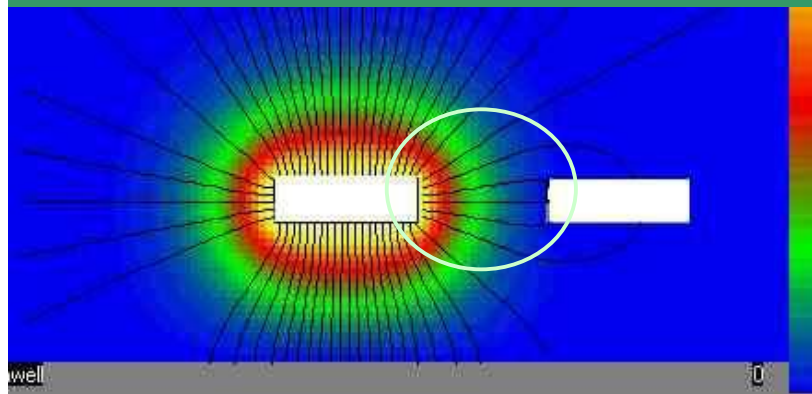
...giving you the squeeze of nanometer design technology



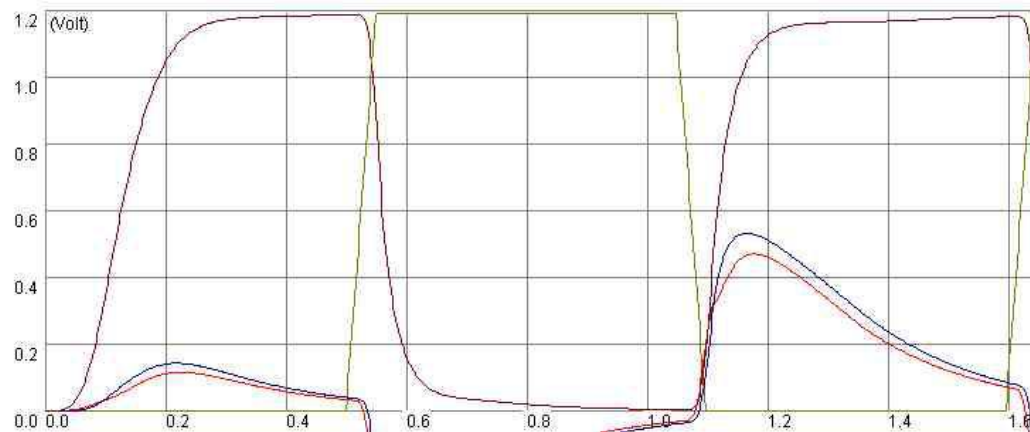
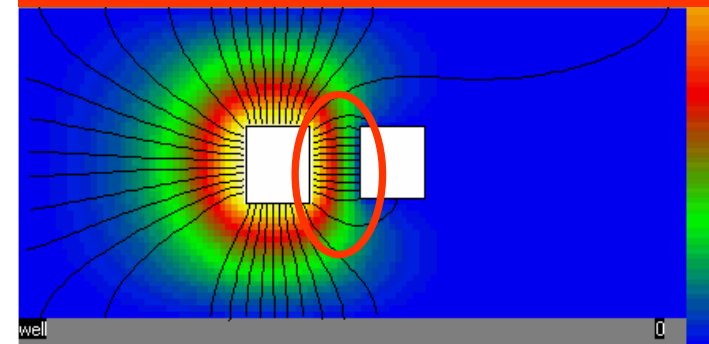
μωηδ<sup>©</sup>

## 5. Signal propagation

0.7μm Small coupling



0.12μm Strong coupling



- Very large noise, close from fault
- Low K to reduce coupling
- Long distance routing is forbidden (Critical routing length 2mm in 0.12μm)

...giving you the squeeze of nanometer design technology

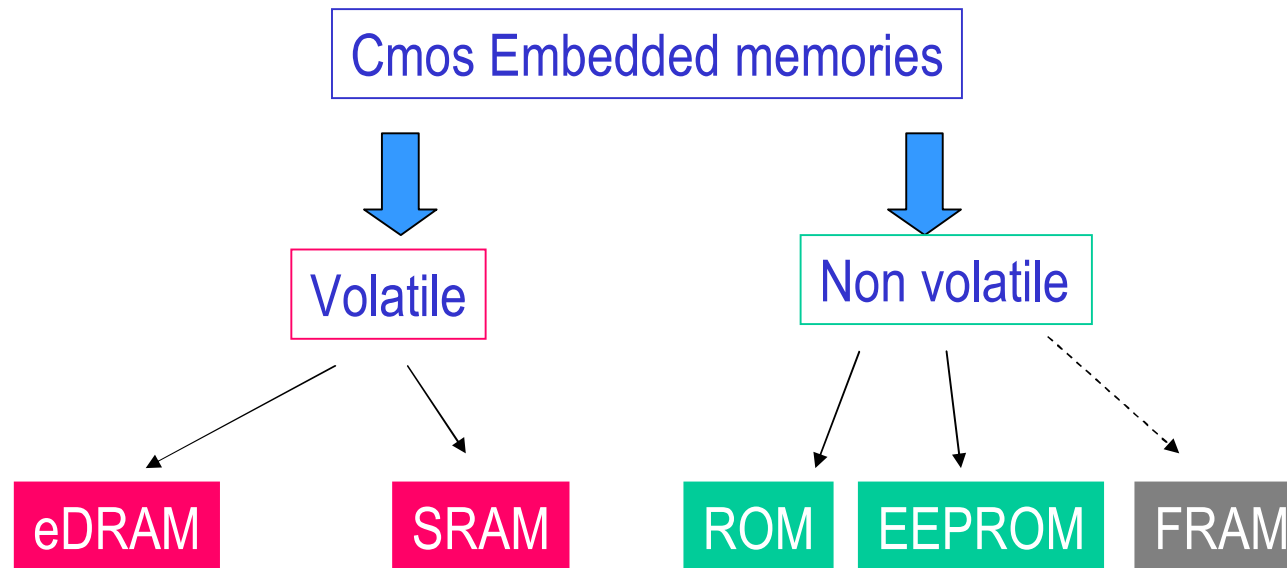
[www.microwind.net](http://www.microwind.net)



μωιηδ<sup>®</sup>

## 6. Embedded Memories

- 80% of a system-on-chip
- Bottleneck for bandwidth



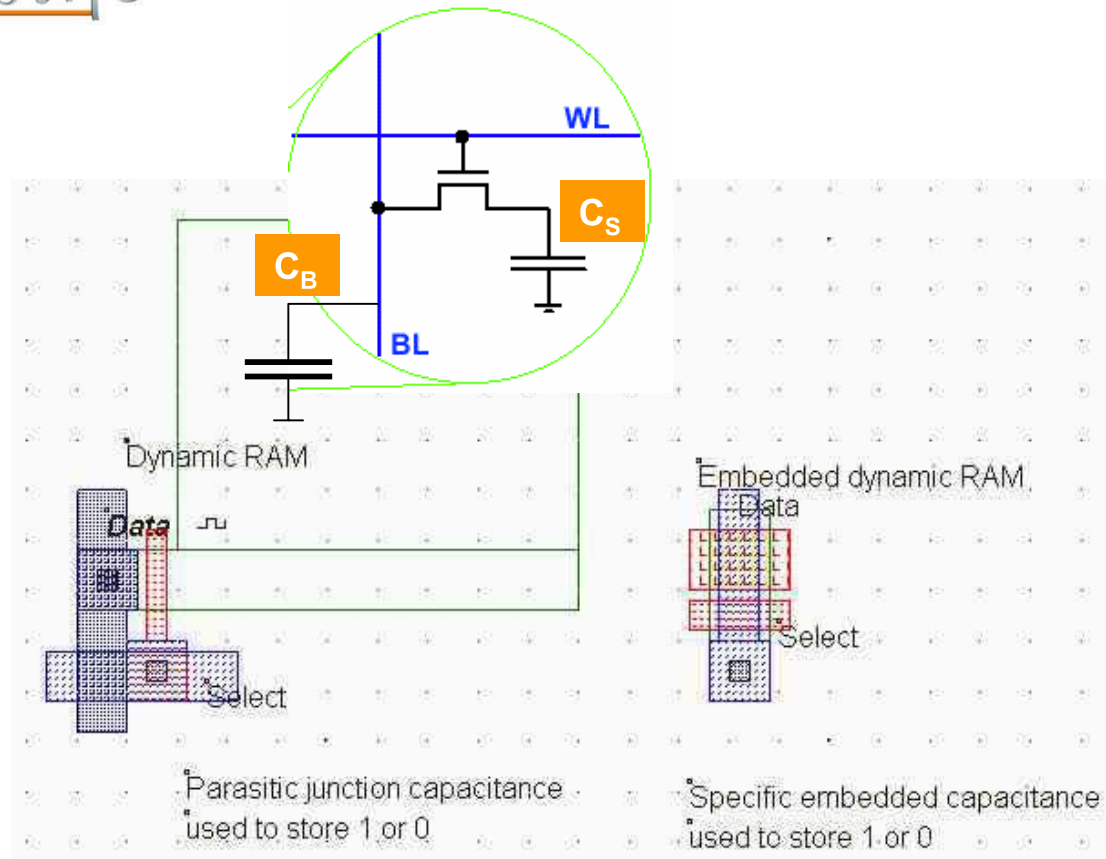
...giving you the squeeze of nanometer design technology

[www.microwind.net](http://www.microwind.net)



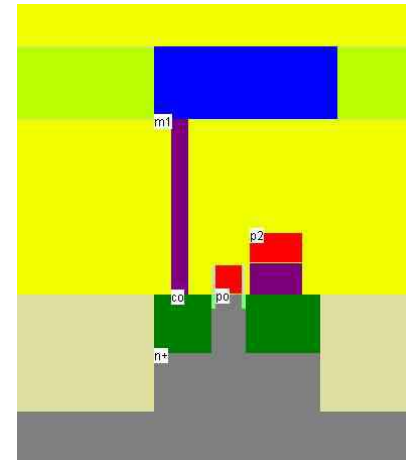
μωηδ<sup>©</sup>

## 6. Embedded Memories



Parasitic capacitance: 2fF

Specific capacitance: 3-30fF



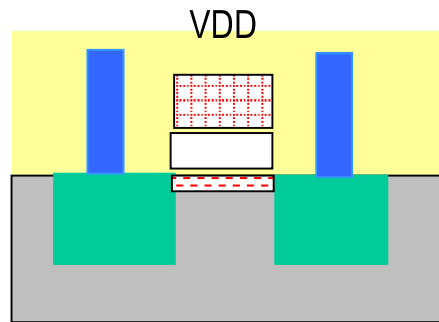
...giving you the squeeze of nanometer design technology

[www.microwind.net](http://www.microwind.net)

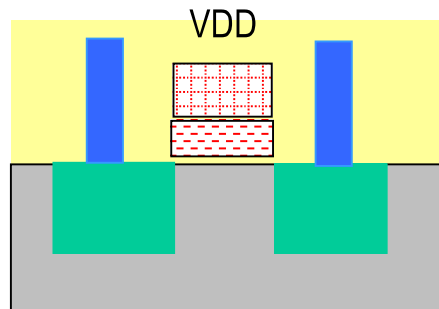


μωηδ<sup>©</sup>

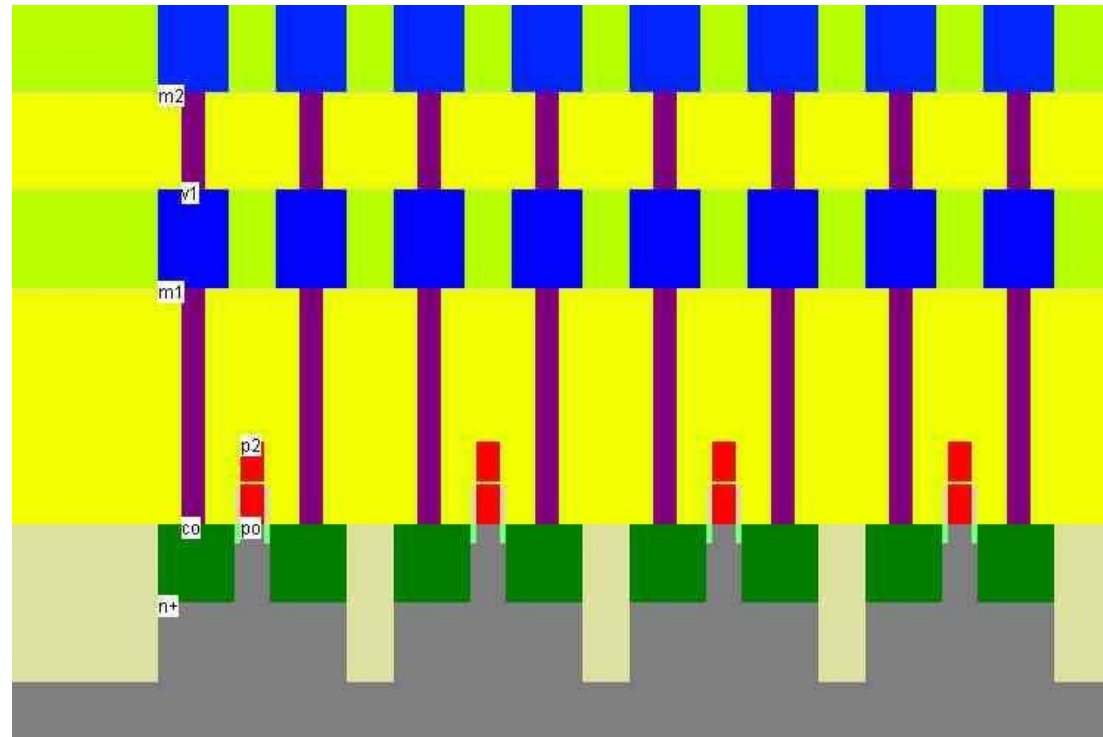
## 6. Embedded Memories



Create a small channel



Cannot create channel



Electrons injected in the floating gate by tunneling

...giving you the squeeze of nanometer design technology

[www.microwind.net](http://www.microwind.net)





μωηδ<sup>©</sup>

## 7. SOI

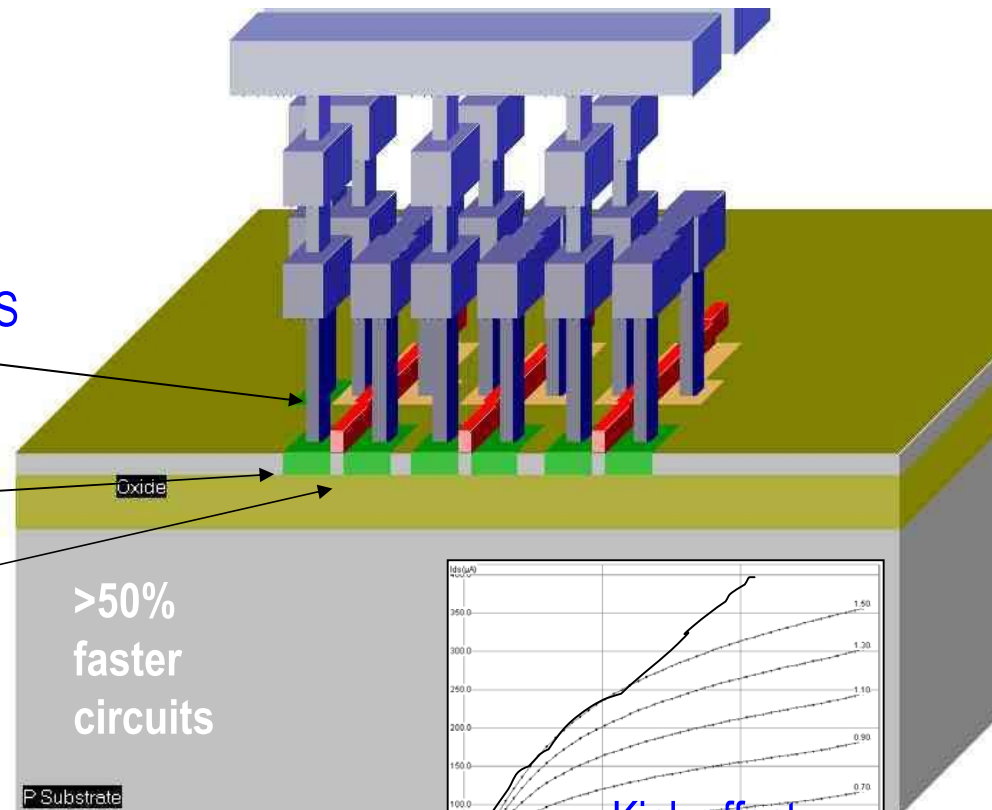
The next major evolution?

CMOS compatible

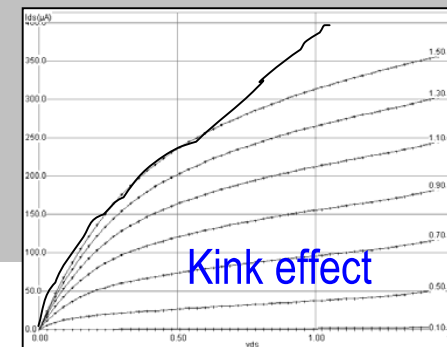
Less distance between nMOS and pMOS

Less capacitance

Less leakage

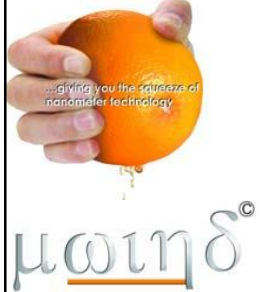


Fully or partially depleted?



...giving you the squeeze of nanometer design technology

[www.microwind.net](http://www.microwind.net)



# Conclusion

- The technology scale down has been illustrated
- Design trend towards higher levels of abstraction
- More MOS options oriented to applications in 0.1 $\mu$ m technology
- Increased interconnect layers improve density but many issues
- RC delay & crosstalk illustrated
- Embedded memories have several design styles and technological option
- Substrate below 0.1 $\mu$ m should be in SOI
- Lots of educational messages illustrated in Microwind PC tool

...giving you the squeeze of nanometer design technology

[www.microwind.net](http://www.microwind.net)