

4. Фреик Д.М., Галушак М.А., Межиловская Л.И. Физика и технология полупроводниковых пленок. – Львов: Высшая школа, 1988. – 152 с.

*The dependence Hall mobility on thickness and also mechanisms of their scattering in thin-film structures on the basis of tellurium are obtain. Under condition of implementation the two-layer and diffusion scattering is determined kinetic factor of near-surface layers of the thin films.*

*Key words: films, tellurium, scattering, Petritz model.*

УДК 546.28, 681.31

ББК 31.233, 32.97

Р.Б. Атаманюк, С.П. Новосядлий

### МІНІМІЗАЦІЯ АЛГОРИТМІЧНОГО ПРОЦЕСУ ПРОЕКТУВАННЯ ІНТЕГРАЛЬНИХ СХЕМ МЕТОДОМ КРЕМНІЄВОЇ КОМПІЛЯЦІЇ ЗА ЧАСОВО-ЙМОВІРНІСНИМ КРИТЕРІЄМ

*Запропоновано вдосконалення методу кремнієвої компіляції через оптимальний покрововий вибір операторів та логічних умов за критеріями часу та ймовірності правильного виконання з дотриманням КТО. Введено поняття достовірних операторів та гарантованих умов. Суть алгоритму оптимізації – у відсіканні на кожному кроці кремнієвої компіляції неперспективних варіантів реалізації операторів та логічних умов.*

*Ключові слова: кремнієва компіляція, інтегральні схеми, оптимізація, верифікація.*

Кремнієва компіляція – це метод автоматичної побудови топології кристала великих інтегральних схем (ВІС).

Кремнієвий компілятор – це генератор топології, тобто сукупність автоматичних перетворень від спеціальних властивостей ВІС до геометричного опису кристала в цілому. Кремнієвий компілятор зорієнтований на використання всіх можливостей технології ВІС, а не лише на такі окремі методи, як, наприклад, декомпозиція, метод вентильних матриць (проекткування базових матричних кристалів), метод стандартних бібліотечних елементів.

Кремнієвий компілятор – це програмна генерація топології мовою ICL – мовою інтегральних схем. Цією мовою створено, як мінімум, два кремнієві компілятори: RELAY, BRISTLE BLOCKS. Кремнієвий компілятор – це САПР ВІС, що задовольняє три умови:

- 1) ієрархічність проекту;
- 2) високий рівень мови вхідного опису проекту;
- 3) застосування параметризованих клітинок замість стандартних.

До переваг методу кремнієвої компіляції відносимо:

1. Скорочення термінів проектування ВІС.
2. Автоматичне дотримання проектних норм самими генераторами топології, так звана “коректність за побудовою”.
3. Можливість залучення до проектування САПР ВІС спеціалістів, які не є конструкторами, оскільки кремнієвий компілятор – це сукупність програмних модулів.

Одним із рівнів, через які проходить проектування ВІС, є алгоритмічний рівень. Кремнієву компіляцію можна подати як алгоритмічний процес. При проектуванні алгоритмічного процесу (АП) кремнієвої компіляції виникають задачі аналізу та синтезу. Позначимо через  $P$  імовірність безпомилкового виконання деякого процесу, через  $T$  – час його виконання. Суть задачі аналізу – у прогнозуванні показників  $P$  і  $T$  на основі інформації про структуру АП кремнієвої компіляції.

Суть задачі синтезу – у генерації такого варіанта АП кремнієвої компіляції, при якому б оптимізувалися показники  $P$  і  $T$ , а саме  $P \rightarrow \max$ ,  $T \rightarrow \min$ . Насправді  $P$  і  $T$  суттєво взаємопов’язані. Тоді дана задача розбивається на дві окремі задачі:

- 1)  $T \rightarrow \min$ ,  $P > P_{\text{доп}}$ ,
- 2)  $P \rightarrow \max$ ,  $T < T_{\text{доп}}$ .

На відміну від методів структурної надійності складних систем, надійність алгоритмічних процесів і кремнієвої компіляції зокрема забезпечується на стадії формування

топології ВІС за рахунок можливості системи виявляти та виправляти помилки при виконанні операцій.

За допомогою мови програмування ICL (Integrated Circuit Language – мови інтегральних схем) весь алгоритмічний процес кремнієвої компіляції в загальному вигляді можна записувати мовою алгоритмічних алгебр шляхом суперпозиції послідовних, паралельних, розгалужених та циклічних алгоритмічних структур. Зауважимо, що алгоритмічні процеси бувають регулярні та нерегулярні [1]. Нерегулярні зображаються графами, але кожний нерегулярний алгоритмічний процес можна звести за скінченну кількість кроків до регулярного. А тоді за допомогою апарату теорії ймовірності та теорії нечітких множин можна розв'язувати задачу надійності кремнієвої компіляції на стадії створення цього алгоритмічного процесу.

Оптимізацію топології в загальному випадку проводять зведенням до задачі лінійного програмування. Цільову функцію будують, виходячи з критеріїв якості топології:

- 1) сума гіпотетичних довжин з'єднувальних провідників має бути мінімальною;
- 2) мінімізація щільності монтажу проводиться за рахунок порожніх комірок методом “шлях деформації”;
- 3) площа топології, зайнята блоками й провідниками, повинна займати не менше 80% від усієї площі кристала, тому мінімізація топології зводиться до мінімізації незайнятої площі;
- 4) при застосуванні методу дихотомічного поділу площа блоків в обох половинах має бути однаковою, але мінімізується кількість провідників, що проходять через лінію поділу.

Система обмежень задачі лінійного програмування виникає з таких конструктивно-технологічних обмежень:

1) геометричні обмеження: а) провідники, що містяться в одному технологічному шарі, не повинні перетинатися, а відстань між ними має бути більшою від деякої заданої допустимої відстані; б) ланцюги, які відносяться до двох або більше технологічних шарів, з'єднуються між собою наскрізними вертикальними отворами, які також мають певні розміри, причому відстані між ними повинні бути більшими від деякої наперед заданої допустимої відстані, зокрема такі отвори для вертикальних з'єднань не можуть розміщуватися всередині блоків;

2) обмеження на довжину провідників: як по горизонтальних трасах, так і по вертикальних каналах провідники мають бути якнайкоротшими – ця вимога накладається для того, щоб зменшити їх опір;

3) обмеження, які пов'язані з допустимою силою струму в шинах живлення: густина струму повинна не перевищувати допустиму норму; якщо густина струму занадто велика, то від тепла може бути розплавлення, тому шини живлення мають бути досить товстими;

4) обмеження на виводи з'єднувальних провідників: рисунок з'єднань задається деревом Штейнера для того, щоб мінімізувати їх довжину;

5) обмеження на стандартизацію методу: передусім задається ширина всіх сигнальних шин та відстань між ними, що тягне за собою зниження щільності монтажу;

6) якщо маємо два технологічні шари та на одному з них провідники проходять уздовж, а на іншому – упоперек, тобто ортогонально, то в такій ґратковій структурі наскрізні вертикальні отвори мають бути розміщені у вузлах решітки.

Крім названої системи обмежень, існують також метричні та топологічні обмеження. Серед метричних назвемо такі: фіксовані розміри елементів, фіксована ширина провідників, мінімальна допустима відстань між провідниками, а також між елементами монтажного рисунка. До топологічних обмежень належать: заборона прокладення трас у зайнятих ділянках, заборона перетину різних з'єднань, вимога електричного об'єднання деяких груп виводів.

Якщо технологічні шари кристала покрити координатними сітками за методом “бітової карти”, то окремо взята клітина може бути покрита кількома елементами з різних шарів. Якщо кожній клітині поставити у відповідність її степінь покриття  $r$ , тобто кількість непорожніх технологічних шарів, якими вона покрита, то доменом називається прямокутник максимальної площі, усі клітини якого мають однаковий степінь  $r$ .

Якщо всі контури паралельні вибраним осям координат, то задача оптимізації топології, тобто побудови такого топологічного рисунка, який забезпечує мінімізацію розмірів кристала та дотримання КТО (конструктивно-технологічних обмежень), зводиться до екстремальної задачі лінійного програмування.

Два методи такого зведення (метод доменів і метод критичного шляху) та формалізація задачі планарного трасування електричних з'єднань, яка зводиться до цілочисельної задачі лінійного програмування, подано А.І. Петренком у [2].

Методом доменів у [2] деталізованому ескізу, розбитому на  $n$  доменів, ставиться у відповідність орієнтований граф, за допомогою якого мінімізується площа кристала симплекс-методом. Цього ж можна досягти також методом критичного шляху. У результаті можна досягнути максимальної щільності упаковки інтегральної схеми. Зведення до задачі лінійного програмування дозволяє оптимізувати топологію практично всіх інтегральних схем.

Застосування даних методів потребує значних об'ємів оперативної пам'яті та великих затрат часу. Тому пропонуємо вдосконалення методу кремнієвої компіляції через оптимальний покроковий вибір операторів та логічних умов за критеріями часу та ймовірності правильного виконання з дотриманням КТО.

Для аналізу надійності кремнієвої компіляції використовують такі алгоритмічні структури:

- 1) послідовна структура (композиція)  $B = A_1 \cdot A_2$ ;
- 2) розгалужена структура ( $\alpha$ -диз'юнкція)  $C = \alpha (A_1 \vee A_2)$ ;
- 3) ітеративна або циклічна структура (зворотна  $\alpha$ -ітерація)  $F = \{A\}_\alpha$ ;
- 4) робота – контроль (виявлення помилок) – виправлення  $K = A_\omega (E \vee V)$ ;
- 5) робота – вибірковий контроль (у тонких місцях – виправлення)

$$P = A_\psi (E \vee_\omega (E \vee V));$$

- 6) багатократна робота (виконання композиції  $N$  разів)  $M = A^N$ .

Для побудови ймовірнісних та часових характеристик, що пов'язані з виконанням операторів та умов, уведемо необхідні події, що пов'язані з виконанням заданих операторів та умов:

- $A^1$  – правильне, тобто безпомилкове виконання оператора  $A$ ,
- $A^0$  – неправильне виконання оператора  $A$ ,
- $\alpha^1$  – апріорна істинність логічної умови  $\alpha$ ,
- $\alpha^0$  – апріорна хибність логічної умови  $\alpha$ ,
- $\alpha^{11}$  – апостеріорна відсутність помилки I роду при виконанні умови  $\alpha$ ,
- $\alpha^{10}$  – апостеріорна наявність помилки I роду при виконанні умови  $\alpha$ ,
- $\alpha^{00}$  – апостеріорна відсутність помилки II роду при виконанні умови  $\alpha$ ,
- $\alpha^{01}$  – апостеріорна наявність помилки II роду при виконанні умови  $\alpha$ .

З урахуванням даних позначень обчислимо функції правильного виконання вказаних алгоритмічних структур:

- 1)  $B^1 = A_1^1 \cdot A_2^1$  – правильне виконання композиції;
- 2)  $C^1 = \alpha^1 \alpha^{11} A_1^1 \vee \alpha^0 \alpha^{00} A_2^1$  – правильне виконання розгалуженої структури;
- 3)  $F^1 = A^1 \alpha^{11} \vee (A^1 \alpha^{10} \vee A^0 \alpha^{00}) \cdot A^1 \alpha^{11} \vee (A^1 \alpha^{10} \vee A^0 \alpha^{00}) \cdot (A^1 \alpha^{10} \vee A^0 \alpha^{00}) \cdot \dots$  – правильне виконання циклічної структури;

- 4)  $K^1 = A^1 \omega^{11} \vee (A^1 \omega^{10} \vee A^0 \omega^{00}) \cdot V^1$  – правильне виконання: робота – контроль – виправлення;
- 5)  $P^1 = A^1 \psi^0 \vee A^1 \psi^1 \cdot (\omega^{11} \vee \omega^{10} \cdot V^1) \vee A^0 \psi^1 \omega^{00} V^1$  – правильне виконання структури робота з вибірковим контролем і виправленням;

- 6)  $M^1 = A^1 \cdot A^1 \cdot \dots \cdot A^1$  ( $N$  разів) – правильне виконання структури багатократної роботи.

Після цього можна замінити вказані оператори єдиним оператором  $B$ , або  $C$ , або  $F$ , або  $K$ , або  $P$ , або  $M$  з еквівалентними характеристиками відповідно  $p_B^1 = p_A^1 \cdot p_{A_2}^1$ ,  $t_B = t_A + t_{A_2}$  для послідовної структури.

Для структури “ $\alpha$ -диз'юнкції”:  $p_C^1 = p_\omega^1 k_\omega^{11} p_A^1 + (1 - p_\omega^1) \cdot k_\omega^{00} p_A^1$ ;

$$t_C = t_\omega + (p_\omega^1 k_\omega^{11} + (1 - p_\omega^1)(1 - k_\omega^{00})) \cdot t_A + (p_\omega^1 (1 - k_\omega^{11}) + (1 - p_\omega^1) \cdot k_\omega^{00}) \cdot t_{A_2}.$$

Структура “робота з вибірковим контролем і виправленням” замінюється одним оператором  $P$  із відповідними часово-ймовірнісними характеристиками:

$$p_P^1 = (1 - p_\psi^1) p_A^1 + p_\psi^1 \cdot (p_A^1 k_\omega^{11} + (p_A^1 (1 - k_\omega^{11}) + (1 - p_A^1) \cdot k_\omega^{00}) \cdot v^1);$$

$$t_p = t_A + P_{\psi}^1 \left( t_{\omega} + t_V \left( P_A^1 (1 - k_{\omega}^{11}) + (1 - P_A^1) \cdot k_{\omega}^{00} \right) \right).$$

Структура “багатократна робота”  $M = A^N$  замінюється одним оператором  $M$  з еквівалентними часово-ймовірнісними характеристиками:

$$P_M^1 = (P_A^1)^N; \quad t_M = N \cdot t_A.$$

Для оцінки надійності алгоритмічного процесу кремнієвої компіляції використовують такий алгоритм:

- I. Подати процес кремнієвої компіляції у вигляді суперпозицій алгоритмічних структур.
- II. Задати або обчислити часово-ймовірнісні характеристики операторів та логічних умов.
- III. Встановити лічильник алгоритмічних структур  $i:=1$ .
- IV. Знайти  $i$ -ту алгоритмічну структуру.
- V. За допомогою оператора вибору замінити дану  $i$ -ту алгоритмічну структуру одним із шести операторів  $B, C, F, K, P, M$  для вказаних типових структур.
- VI. Перевірити, чи наявні ще якісь алгоритмічні структури; якщо “так”, то повернутися до кроку IV, збільшуючи лічильник  $i:=i+1$ , якщо “ні”, то перейти до кроку VII.
- VII. Кінець.

Нехай задано початковий варіант алгоритмічного процесу кремнієвої компіляції, в якому кожний оператор на  $i$ -тому кроці може бути реалізований різними  $n$  способами. Позначимо цей оператор через  $A_i$ , а відповідну йому кількість варіантів – через  $n_i$ .

Нехай для кожної логічної умови  $\omega_j$  також існує  $n_j$  способів реалізації. Наявність різних варіантів для логічних умов пов’язана з урахуванням на кожному кроці проектування існуючих КТО. Позначимо такі набори варіантів векторами

$$A_i = \{A_{i1}, A_{i2}, \dots, A_{in}\}; \quad \omega_j = \{\omega_{j1}, \omega_{j2}, \dots, \omega_{jn}\}.$$

Нехай  $X = \{x_1, x_2, x_3, \dots, x_n\}$  – це список усіх існуючих КТО, тобто вектор керованих змінних, від якого залежить вибір номерів операторів та логічних умов. Нехай  $P(x)$  і  $T(x)$  – ймовірність та час правильного виконання кремнієвої компіляції тоді, коли керовані змінні задані вектором  $X$ . Нехай  $P_{\text{дон}}$  – найменше допустиме значення ймовірності,  $T_{\text{дон}}$  – найбільше допустиме значення часу.

Тоді в прямій постановці задача оптимізації має вигляд:  $T(x) \rightarrow \min, P(x) \geq P_{\text{дон}}$ , а у зворотній постановці задача оптимізації виглядає так:  $P(x) \rightarrow \max, T(x) \leq T_{\text{дон}}$ .

Суть алгоритму оптимізації – у відсіканні на кожному кроці кремнієвої компіляції неперспективних варіантів реалізації операторів та логічних умов. При відсіканні будемо дотримуватися таких критеріїв:

1. Якщо  $A_1$  і  $A_2$  – різні варіанти реалізації оператора  $A$  мовою інтегральних схем ICL, то реалізацію  $A_1$  вибираємо тоді й тільки тоді, коли виконуються дві такі нерівності:

$$P_{A_1}^1 > P_{A_2}^1 \quad \text{і} \quad t_{A_1} < t_{A_2},$$

де  $P^1$  – це ймовірність правильного виконання оператора  $A_1$ , а  $P_{A_2}^1$  – ймовірність правильного виконання оператора  $A_2$ .

2. Так само вибираємо логічну умову  $\omega_1$  тоді й тільки тоді, коли виконуються дві такі нерівності:

$$k_{\omega_1}^{11} > k_{\omega_2}^{11} \quad \text{і} \quad t_{\omega_1} < t_{\omega_2},$$

де  $k_{\omega_1}^{11}$  означає ймовірність відсутності помилок першого роду при контролі, а через  $k_{\omega_1}^{10}$  позначаємо ймовірність наявності помилок першого роду при контролі, через  $k_{\omega_1}^{01}$  позначаємо ймовірність відсутності помилок другого роду при контролі, а через  $k_{\omega_1}^{00}$  позначаємо ймовірність наявності помилок другого роду при контролі.

Записуємо алгоритм оптимізації процесу кремнієвої компіляції в прямій постановці:

- 1) установити лічильник алгоритмічних структур  $m:=1$ ;
- 2) знайти алгоритмічну структуру з номером  $m$ ;

3) відкинути неперспективні варіанти виконання операторів, що входять у дану структуру, залишити тільки оптимальні варіанти;

4) відкинути неперспективні варіанти виконання логічних умов, що входять у дану  $m$ -ту структуру, і залишити тільки оптимальні варіанти;

5) обчислити характеристики всіх варіантів виконання  $m$ -ї структури, ураховуючи, що ймовірності перемножуються, а часи – додаються;

6) відкинути неперспективні варіанти виконання всієї  $m$ -тої структури;

7) перевірити умову: “весь попередній процес кремнієвої компіляції замінено еквівалентним оператором”. Якщо “так”, то перейти до пункту 8, якщо “ні”, то збільшити лічильник  $m := m + 1$  і перейти до пункту 2;

8) відкинути всі варіанти процесу кремнієвої компіляції, для яких виконується нерівність  $P(x) < P_{\text{дон}}$ , а з решти варіантів вибрати той, для якого досягається  $T(x) \rightarrow \min$ ;

9) розгорнути вибраний варіант до попереднього рівня операторів та логічних умов мовою інтегральних схем ICL і записати оптимальний процес кремнієвої компіляції.

Уведемо поняття достовірних операторів та гарантованих умов – це такі оператори й умови, які в процесі кремнієвої компіляції виконуються зі ймовірністю одиниця, тобто виконуються завжди. Тоді алгоритм оптимізації у зворотній постановці виглядає так:

1) установити лічильник алгоритмічних структур  $m := 1$ ;

2) знайти структуру з номером  $m$ ;

3) відкинути неперспективні варіанти для операторів, використавши вищеназваний критерій, і зробити це для всіх операторів, що входять у  $m$ -ту структуру;

4) з усіх операторів  $m$ -тої структури, що є достовірними, відкинути ті, які перевищують обмеження за часом;

5) відкинути неперспективні варіанти для логічних умов, що входять у  $m$ -ту структуру згідно з критерієм;

6) з решти гарантованих логічних умов відкинути ті, які перевищують обмеження за часом;

7) обчислити характеристики всіх варіантів виконання  $m$ -тої структури, користуючись правилами, що ймовірність послідовного виконання операторів дорівнює добутку ймовірностей, а час послідовного виконання операторів рівний сумі часів виконання кожного оператора;

8) відкинути неперспективні варіанти виконання  $m$ -тої структури, користуючись критерієм;

9) перевірити умову: “чи замінено весь попередній процес кремнієвої компіляції еквівалентним оператором”. Якщо “так”, то перейти до пункту 10, а якщо “ні” – то збільшити лічильник  $m := m + 1$  і перейти до пункту 2;

10) відкинути варіанти процесу кремнієвої компіляції, для яких перевищено часові обмеження, тобто  $T(x) > T_{\text{дон}}$ ;

11) з решти варіантів вибрати той, для якого досягається  $P(x) \rightarrow \max$ . Цей варіант дає оптимальний набір аргументів, тобто список конструктивно-технологічних обмежень, на яких досягається максимум;

12) розгорнути вибраний варіант алгоритмічного процесу до рівня операторів та логічних умов і записати оптимальний процес кремнієвої компіляції мовою інтегральних схем ICL.

**Висновки.** Знайдено вдосконалений метод кремнієвої компіляції, тобто такий спосіб побудови алгоритмічного процесу, що забезпечує вибір оптимальної послідовності операторів за часово-ймовірнісним критерієм. Надійність кремнієвої компіляції визначається її алгоритмічним процесом.

1. Базилевич Р.П. Декомпозиционные и топологические методы автоматизированного конструирования электронных устройств. – Львов: Вища школа, Изд-во при Львов. ун-те, 1981. – 168 с.
2. Петренко А.И. и др. Автоматизация конструирования больших интегральных микросхем. – К.: Вища школа, 1983. – 312 с.
3. Мельник Р.А. Алгоритмы иерархичного моделювання площинної та просторової топології НВІС. – Львів: Вид-во ДУ “Львівська політехніка”, 1999. – 140 с.

4. Atamanyuk R.B, Novosyadly S.P. Silicic compiling as instrument of planning of topology of large-scale arrays and providing its reliability at algorithmic level // The IX International Conference "Modern problems of radio engineering, telecommunications and computer science" (Lviv – Slavske, February 28 – March 4, 2006). – Lviv, 2006. – P.604–605.
5. Атаманюк Р.Б, Новосядлий С.П. Ущільнене компоунвання топології інтегральних схем методом деформаційної ретракції // П'ята науково-технічна конференція "ПРИЛАДОБУДУВАННЯ 2006: стан і перспективи" (25 – 26 квітня 2006 р.). – К., 2006. – С.209–210.

*In this paper the silicic compiling process is described as instrument of planning of topology of large-scale arrays and providing its reliability at algorithmic level of the integrated circuit. Improvement of method of the silicic compiling through discrete optimization of topology is proposed.*

**Key words:** silicic compiling, integrated circuit, optimization, verification.

УДК 621.382

ББК 32.85

*В.І. Голота, І.Т. Козут*

### ЕЛЕМЕНТИ АРХІТЕКТУРИ МАТРИЧНО-ЕМІТЕРНОЇ НАНОМЕТРОВОЇ ЛІТОГРАФІЇ

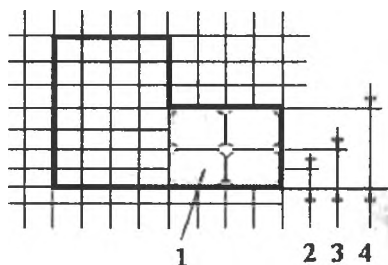
*Розглянуто сучасний стан розвитку оптично-проекційної та матрично-емітерної нанометрової літографії. Подано вихідні дані та результати розрахунків структури й основних елементів архітектури системи. Запропоновано інтегровані КНІ КМОН структури на основі базових матричних кристалів для реалізації ІС логіки, пам'яті й керування матричними випромінювачами.*

**Ключові слова:** нанометрова літографія, елемент архітектури, КНІ КМОН структури, мікроелектронні пристрої.

**Вступ.** Прогрес сучасної мікроелектроніки в значній мірі визначається можливостями технології виробництва інтегральних схем (ІС). Одним із базових технологічних процесів у виробництві ІС є літографія. У сучасних оптичних літографіях використовується ультрафіолетове (УФ), глибоке ультрафіолетове (ГУФ) та екстремальне ультрафіолетове (ЕУФ) випромінювання [1]. Літографії з УФ, ГУФ та ЕУФ випромінюванням забезпечують технологічні норми нанометрового діапазону. ЕУФ літографія має найвищу роздільну здатність і продуктивність і буде основною в масовому виробництві ІС на найближче десятиріччя. Але в зв'язку з надзвичайною складністю та високою вартістю ЕУФ літографії, недовговічністю проекційної оптики й відбиваючих масок актуальним є завдання створення альтернативних літографій. Однією з таких є матрично-емітерна (МЕ) літографія.

**Оптичні проекційні літографії.** На даний час у масовому виробництві ІС використовуються оптичні проекційні літографії з технологічними нормами 180 нм, 130 нм, 90 нм, 65 нм, а в перспективі планується освоєння – 45 нм (2007), 32 нм (2009), 22 нм (2011) [2, 3].

Технологічна норма визначає мінімальний (планарний) розмір елемента (МРЕ) топології (рис.1).



*Рис.1. Топологічний елемент: 1 – піксел; 2 – крок координатної сітки; 3 – розмір пікселя; 4 – мінімальний розмір елемента*

Так, для оптичної проекційної літографії МРЕ визначається з формули (1):

$$MRE = k_1 \frac{\lambda}{NA}, \quad (1)$$