

Задачі конструкторського проектування

Автоматизація конструкторського проектування стосовно комп'ютерних систем (електронної техніки) полягає в автоматизації окремих його задач. До них відносяться:

- Компонування;
- Розміщення;
- Трасування;
- Аналізу.

Задача компонування в цьому випадку складається із задач покриття і розбиття (рис. 3.19).

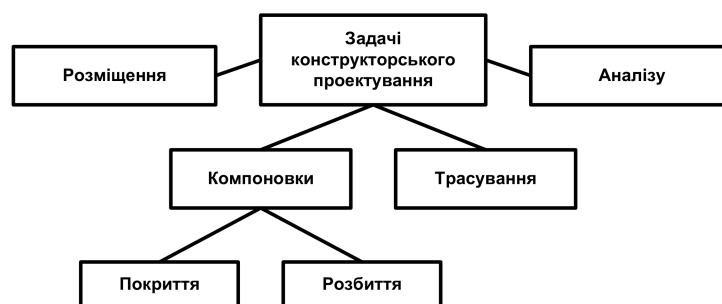


Рисунок 3.19 - Задачі конструкторського проектування

Для вирішення усього набору поставлених задач нині існує досить велика кількість спеціалізованих САПР, призначених для роботи в різних галузях виробництва. Такі САПР відрізняються один від одного різними параметрами, наприклад, мірою охоплення задач, математичними методами, закладеними алгоритми в рішення окремих задач, можливостями з точки зору кількості вживаних комплектуючих при проектуванні, об'ємів використовуваних БД, їх розрядності, зручність використання і так далі

При проектуванні електронних систем на конструкторському рівні важливе місце займає процес проектування друкованих плат, як одного з основних елементів сучасних електронних пристроїв.

Автотрасувальники

Автотрасувальники викликаються і виробляється налаштування стратегії трасування. Інформацію про особливості трасування окремих ланцюгів можна за допомогою стандартних атрибутів ввести на етапах створення принципової схеми або плати друкованої.

Програми призначені для автоматичної розводки друкованих плат з високою щільністю розміщення елементів. Ефективна при поверхневому монтажі корпусів елементів, виконаних в різних системах координат. Є

можливість розміщення провідників під різними кутами на різних шарах плати, оптимізації їх довжини і числа перехідних отворів.

Постановка задачі компоновання. Задача покриття

Задачу компоновання можна розглядати як сукупність двох взаємозв'язаних задач, – покриття і розбиття.

Формально *задача покриття* в даному випадку полягає в заповненні деякого простору, що характеризує функціональну схему, що реалізовується, окремими логічними елементами.

Початкові дані – функціональна схема з'єднань логічних елементів вузла і логічні схеми типових конструкторських елементів (модулів).

При рішенні задачі необхідно кожен логічний елемент початкової функціональної схеми реалізувати логічними елементами, що входять до складу типових модулів, з урахуванням певних вимог і обмежень.

Набори типових модулів включають два типи модулів (рис. 1.3) :

- елементні модулі, тобто що складаються з не пов'язаних логічних елементів (багатоцільових);
- функціональні модулі, де логічні елементи пов'язані в прості функціональні схеми.

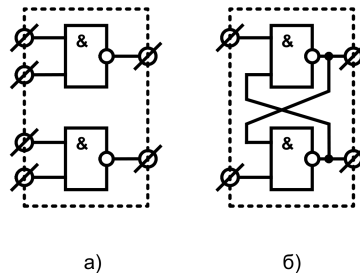


Рисунок 3.20-Типи модулів покриття

а) – елементний; б) – функціональний.

Задача покриття схеми елементними модулями, особливо для модулів з однаковими елементами зводиться до простішої задачі розбиття на підсхеми. Задача покриття функціональними модулями – складніша і вимагає розпізнавання відповідних структур у початковій схемі.

Основні показники якості покриття

Оптимізація покриття виробляється за показниками:

- Сумарна вартість модулів, що беруть участь в покритті;
- Загальне число модулів;
- Число типів використовуваних модулів;
- Кількість зв'язків між модулями;
- Кількість невживаних логічних елементів в модулі.

Можливі математичні формулювання задачі покриття. Нехай заданий набір модулів $T = (t_1, t_2, \dots, t_n)$, де n – число типів модулів в наборі. Такий набір характеризується матрицею

$$A = [a_{ij}]_{m \times n}$$

де a_{ij} – відповідає числу логічних елементів i -го типу в модулі j -го типу, m – загальне число типів логічних елементів в усіх модулях набору.

Поелементний склад заданої функціональної схеми характеризується вектором

$$B = (b_1, b_2, \dots, b_i)$$

де b_i – число логічних елементів i -го типу в схемі.

Введемо цілочисельну змінну x_j , що характеризує кількості модулів j -го типу, необхідних для покриття заданої схеми.

У простому випадку, коли ставиться задача покриття з мінімальною кількістю модулів, цільова функція набирає вигляду:

$$F = \sum_{j=1}^n x_j.$$

При обмеженнях:
$$\sum_{j=1}^n a_{ij} x_j \geq b_j, i = 1, 2, \dots, m; x_j \geq 0; j = 1, 2, \dots, n; a_{ij} > 0.$$

При мінімізації вартості:
$$F = \sum_{j=1}^n c_j x_j$$

де c_j – вартість модуля j -го типу.

У разі обліку декількох вимог вводиться додаткові критерії, наприклад,

при обліку критеріїв вартості і кількості модулів
$$F = k_1 \sum_{j=1}^n c_j x_j + k_2 \sum_{j=1}^n x_j,$$

де k_1 і k_2 – коефіцієнти важливості критеріїв.

Рішення цих задач на рівні практичної складності є дуже трудомістким процесом (навіть з урахуванням обробки на ЕОМ). Тому для їх вирішення часто використовуються наближені евристичні методи.

Простий евристичний алгоритм представляє усі модулі елементними. Для деякого логічного елементу b_i схеми вибирається один з модулів t_k . Потім розглядаються елементи b_i , пов'язані з елементом b_i і теж відповідають t_k . З усіх розглянутих b_i зберігається той, для якого число зв'язків з елементом b_i максимальне. Цим досягається мінімізація числа міжмодульних зв'язків. Потім, якщо елементів b_j пов'язаних з b_i більше немає, то розглядаються елементи b_r , які пов'язані із вже закріпленими елементами, що мають зв'язок з b_i .

Ця процедура триває до тих пір, поки усі логічні елементи початкової функціональної схеми не будуть закріплені за модулями заданого набору. У кінці роботи здійснюють парні перестановки однотипних логічних елементів різних модулів для зменшення міжмодульних зв'язків.

Задача розбиття

Початковими даними задачі є схема з'єднань конструктивних елементів на певному ієрархічному рівні конструкторського проектування.

Задача полягає в необхідності розділити початкову схему на частини так, щоб утворити конструктивні вузли наступного ієрархічного рівня з урахуванням вимог і обмежень.

Критерії якості розбиття

Якість виконання задачі розбиття аналізується по таких критеріях:

- Довжина зовнішніх зв'язків;
- Об'єм конструкції, що характеризується числом вузлів розбиття;
- Числом різних типів вузлів;
- Нерозривність функціональних призначень вузлів.

Початкову схему з'єднання елементів можна представити зваженим мульти графом $G = (X, A)$ у якому елементи – вершини вузла, а міжелементні зв'язки – ребра (рис. 3.).

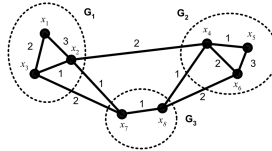


Рисунок 3.21 - Зважений мульти граф, що відбиває початкову схему

Графові G відповідає матриця суміжності $A = [a_{ij}]_{n \times n}$

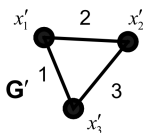
де n – кількість елементів в схемі.

Нехай задані обмеження на місткість вузлів – k і допустиме число зовнішніх виводів у вузлі – v . Вимагається розрізати початковий граф на окремі підграфи

$$G_l = (X_l, A_l)$$

де l – число підграфів $(1, 2, \dots, m)$.

Виділяємо три вузли G_1, G_2, G_3 отримуємо граф $G' = (X', A')$ де безліч вершин X' – відповідає вузлам розбиття, а безліч ребер A' визначає міжвузлові з'єднання.



Позначимо через V_l загальне число зовнішніх з'єднань l -го вузла графа G . Очевидно, що $V_l = \sum_{x_i \in X'} \sum_{x_j \notin X'} a_{ij}$.

Використовуючи матрицю рішень цей вираз можна представити як

$$V_l = \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{il} (1 - b_{jl}).$$

Для оцінки міри зв'язності схеми міжвузлових з'єднань служить величина

$$S = 0,5 \sum_{l=1}^m V_l.$$

Критерій оптимізації міжвузлових з'єднань остаточно набирає вигляду

$$F = 0,5 \sum_{l=1}^m \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{il} (1 - b_{jl})$$

Цей алгоритм застосовується для вирішення задач високої розмірності. У інших випадках використовуються наближені (евристичні) алгоритми.

Задача розміщення

Початкові дані:

- Схема з'єднань конструктивних елементів вузла;
- Конструктивні параметри елементів;
- Параметри монтажного простору.

Рішення задачі полягає у визначенні оптимального розташування елементів в заданому монтажному просторі з урахуванням вимог і обмежень. Головна задача розміщення – полегшення процесу трасування з'єднань, що йде за нею. При дротяному монтажі головний критерій оптимальності є сумарна зважена довжина з'єднань

Для друкарського монтажу окрім цього має значення взаємне розташування з'єднань і число перетинів, що враховується додатковим критерієм сумарної довжини найкоротших дерев.

Формалізація задачі розміщення характерна для випадку регулярного простору і однакових елементів.

За основу візьмемо граф $G = (X, A)$, який характеризується матрицею суміжності

$$A = [a_{ij}]_{n \times n},$$

де n – число модулів; a_{ij} – число з'єднань між модулями x_i і x_j .

Розіб'ємо площу реальної друкованої плати в координатах X, Y на m областей (позицій) як показано на рисунку.

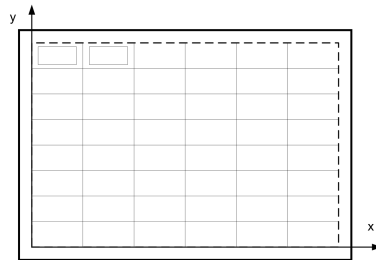


Рисунок 3.22- Рішення задачі розбиття.

Поставимо у відповідність друкованій платі граф $G_r = (P, U)$ безліч вершин – центри посадочних місць, безліч ребер – координатні грати, що зв'язують вершини графа.

Граф G_r характеризується матрицею відстаней $D = [d_{ij}]_{m(m)}$, де d_{ij} – відстань між позиціями i і j .

Введемо матрицю призначень, що характеризує результати рішення задачі розміщення, : $B = [b_{ij}]_{n(m)}$,

де $b_{ij} = 1$, якщо x_i знаходиться у позиції p_j і $b_{ij} = 0$ інакше.

Тоді сумарна зважена довжина з'єднань при довільному розміщенні модулів буде

$$F(B) = 0,5 \sum_{i=1}^n \sum_{j=1}^n d_{ij} d(x_i, x_j).$$

Задача полягає в мінімізації функціонала F при змінюваній матриці B і

обмеженнях $\sum_{i=1}^n b_{ij} = \sum_{j=1}^n b_{ij} = 1$; $b_{ok} = 1$.

В цьому випадку так само, як і в попередніх задачах часто використовуються наближені евристичні алгоритми:

- Послідовні;
- Ітераційні;

– Безперервно-дискретні.

Задача трасування

Похідні дані для рішення цієї задачі такі:

- параметри комутаційного поля;
- координати контактів кожного конструктивного елемента;
- список усіх електричних ланцюгів і перелік контактів, які відносяться до кожного окремого ланцюга.

Задача трасування полягає у з'єднанні друкованими, плівковими або навісними провідниками всіх контактів елементів відповідно до схеми з'єднань з урахуванням вимог та обмежень.

Трасування з'єднань є, як правило, завершальним етапом конструкторського проектування РЕА і полягає у визначенні ліній, що сполучають екіпотенціальні контакти елементів, і компонентів, що становлять проєктований пристрій.

Задача трасування одна з найбільш трудомістких в загальній проблемі автоматизації проектування РЕА. Це пов'язано з декількома чинниками, зокрема з різноманіттям способів конструктивно-технологічної реалізації з'єднань, для кожного з яких при алгоритмічному рішенні задачі застосовуються специфічні критерії оптимізації і обмеження. З математичної точки зору трасування – складна задача вибору з величезного числа варіантів оптимального рішення.

Основними критеріями при отриманні варіантів оптимального рішення задачі трасування є:

- відсоток реалізованих з'єднань;
- сумарна довжина провідників;
- число перетинів провідників;
- число монтажних шарів;
- число міжшарових переходів і т.д.

Часто ці критерії є взаємовиключними, тому оцінка якості трасування ведеться по домінуючому критерію при виконанні обмежень по інших критеріях або застосовують аддитивну або мультиплікативну форму оцінної функції, наприклад, наступного вигляду:

$$F = \sum_p \lambda_i f_i ,$$

де F – аддитивний критерій; λ_i – ваговий коефіцієнт; f_i – приватний критерій; p – число частинних критеріїв.

Послідовний характер трасування властивий як рішенню задачі в цілому, так і кожному етапу окремо. Всі алгоритми рішення задачі є локально-оптимальними, тобто траси, одержані в результаті роботи, оптимальні тільки на даному кроці за наявності раніше проведених трас. На подальшому кроці попередні з'єднання можуть виявитися не оптимальними.

Методи і алгоритми трасування друкованих плат

Конструкція друкованих плат (ДП) є сукупністю монтажних шарів, розташованих один під іншим.

Монтажний шар – це площина обмежених розмірів, на якій умовно нанесена сітка, що визначає місця прокладення друкарських провідників (трас). Крок такої сітки визначає обмеження на ширину друкарського провідника і відстань між ними.

Етапи трасування

Рішення задачі трасування БДП складається з наступних етапів:

- визначення переліку (списку) всіх провідників, які повинні бути прокладені між різними контактами;
- розподіл провідників після куль (розшарування);
- визначення послідовності трасування в шарі;
- власне саме трасування.

На першому етапі розв'язується питання послідовності з'єднання контактів одного ланцюга. Рішення зводиться до побудови мінімального зв'язуючого дерева.

Розглянемо граф $G = (X, A)$, де множина ребер з приписаною вагою d_{ij} характеризує всі можливі з'єднання електричного ланцюга. Граф G визначається матрицею $D = [d_{ij}] n \times n$

де n – число контактів в ланцюзі ($n \geq 3$). Необхідно визначити дерево G' , що включає всі його вершини і що має мінімальну сумарну вагу ребер.

Найбільшого поширення для цього набув алгоритм *Прима*.

На першому кроці для довільного контакту знаходиться найближча вершина і з'єднується ребром. Потім на інших $n - 2$ кроках знаходиться такий контакт, який ближче до групи вже приєднаних. У результаті для контактів x_1 (x_8 (рис. 3.) перелік з'єднань буде таким:

$[(x_1, x_2)(x_2, x_3)(x_2, x_4)(x_4, x_5)(x_5, x_6) (x_6, x_7)(x_7, x_8)]$.

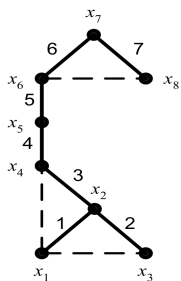


Рисунок 3.23 – Граф з'єднань за алгоритмом Прима

Другий етап може здійснюватися двома способами:

- послідовно проводяться з'єднання до заповнення чергового шару, потім здійснюється перехід до наступного.

– підраховується можливе число перетинів провідників, суміщених в одному шарі, потім проводиться розподіл між шарами.

Перший спосіб простий і не вимагає пояснень. При використанні іншого способу підрахунок можна здійснювати для двох можливих представлень з'єднання двох контактів на площині:

а) з'єднання представляються у вигляді прямолінійних відрізків, а їх перетини визначаються по рівняннях прямих ліній (рис. 3. 24);

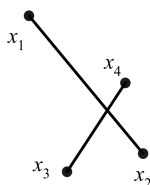


Рисунок 3. – Представлення з'єднання у вигляді прямолінійних відрізків

б) з'єднання представляються в ортогональному просторі, а їх перетини визначаються по рівняннях прямих паралельних осей координат або за фактом перекриття мінімальний прямокутників (рис. 3. 25).

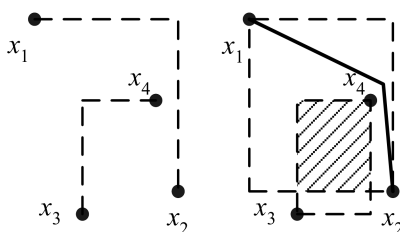


Рисунок 3. 25 – Представлення з'єднання в ортогональному просторі

У разі представлення провідників прямими відрізками етап розшарування представляється таким чином.

На площині дана система провідників (рис. 3.26, а)). Цій системі відповідає граф перетинів G (рис. 3.26, б)), де вершини – провідники, а ребра – їх перетини.

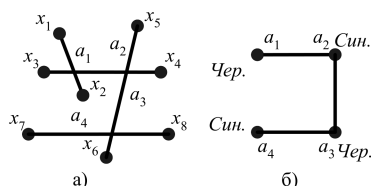


Рисунок 3.26 – Система провідників:

а) розташування на площині; б) граф перетинів.

На етапі трасування в більшості використовуваних методів уся площина плати розбивається на квадрати, розмір яких рівний допустимому розміру між провідниками і вважається, що провідники можуть проходити тільки перпендикулярно граням квадратів.

Трасою між контактами x_i і x_j називається сукупність сусідніх квадратів (що мають загальну сторону), що сполучають контакти x_i і x_j .

Алгоритми трасування

Трасування з'єднань являється, як правило, завершальним етапом конструкторського проектування РЭА і полягає у визначенні ліній, що сполучають еквіпотенціальні контакти елементів, і компонентів, що становлять проєктований пристрій.

Задача трасування – одна з найбільш трудомістких в загальній проблемі автоматизації проектування РЭА. Це пов'язано з декількома чинниками, зокрема з різноманіттям способів конструктивно-технологічної реалізації з'єднань, для кожного з яких при алгоритмічному рішенні задачі застосовуються специфічні критерії оптимізації і обмеження.

З математичної точки зору трасування – складна задача вибору з величезного числа варіантів оптимального рішення.

Одночасна оптимізація усіх з'єднань при трасуванні за рахунок перебору усіх варіантів нині неможлива. Тому розробляються в основному локально оптимальні методи трасування, коли траса оптимальна лише на цьому кроці за наявності раніше проведених з'єднань.

Задача трасування формулюється таким чином: за заданою схемою з'єднань прокласти необхідні провідники на площині (платі, кристалі і так далі), щоб реалізувати задані технічні з'єднання з урахуванням заздалегідь заданих обмежень.

Основними є обмеження на ширину провідників і мінімальні відстані між ними. Початковою інформацією для вирішення задача трасування з'єднань зазвичай є список ланцюгів, параметри конструкції елементів і комутаційного поля, а також дані по розміщенню елементів.

Критеріями трасування можуть бути відсоток реалізованих з'єднань, сумарна довжина провідників, число перетинів провідників, число монтажних шарів, число міжшарових переходів, рівномірність розподілу провідників, мінімальна область трасування і так далі. Часто ці критерії є взаємовиключними, тому оцінка якості трасування ведеться за домінуючим критерієм при виконанні обмежень по інших критеріях.

Відомі алгоритми трасування друкованих плат можна умовно розбити на три великі групи:

- 1) Хвилеві алгоритми, засновані Ли на ідеях і розроблені Ю.Л. Зиманом і Г. Г. Рябовым. Ці алгоритми набули широкого поширення в існуючих САПР, оскільки вони дозволяють легко зважати на технологічну специфіку друкарського монтажу зі своєю сукупністю конструктивних

- обмежень, Ці алгоритми завжди гарантують побудову траси, якщо шлях для неї існує;
- 2) Ортогональні алгоритми, що мають велику швидкодію, чим алгоритми першої групи. Реалізація їх на ЕОМ вимагає в 75-100 разів менше обчислень в порівнянні з хвилевими алгоритмами. Такі алгоритми застосовують при проектуванні друкованих плат з крізними металізованими отворами. Недоліки цієї групи алгоритмів пов'язані з отриманням великого числа переходів з шару на шар, відсутністю 100% - ої гарантії проведення трас, великим числом провідників, що паралельно йдуть;
 - 3) Алгоритми евристичного типу. Ці алгоритми частково засновані на евристичному прийомі пошуку шляху в лабіринті. При цьому кожне з'єднання проводиться по найкоротшому шляху, обходячи перешкоди, що зустрічаються на шляху.

Хвилевий алгоритм Лі

Цей алгоритм є класичним прикладом використання методів динамічного програмування для вирішення задач трасування друкарських з'єднань. Основні принципи побудови трас за допомогою динамічного алгоритму зводяться до наступного.

Усі осередки монтажного поля підрозділяють на зайнятих і вільних. Зайнятими вважаються осередки, в яких вже розташовані провідники, побудовані на попередніх кроках, або знаходяться монтажні виводи елементів, а також осередки, відповідні межі плати і забороненим для прокладення провідників ділянкам. Кожного разу при проведенні нової траси можна використовувати лише вільні осередки, число яких у міру проведення трас скорочується.

На безлічі вільних осередків комутаційного поля моделюють хвилю впливу з одного осередку в іншу, що сполучаються згодом загальним провідником. Перший осередок, зароджується хвиля впливів, називають джерелом, а другу – наступником хвилі. Щоб мати можливість стежити за проходженням фронту хвилі впливів, його фрагментам на кожному етапі привласнюють деякі ваги: $P_k = P_{k-1} + \varphi(f_1, f_2, \dots, f_g)$,

де P_k і P_{k-1} – ваги осередків k -го і $(k-1)$ -го фронтів; φ - вагова функція, що є показником якості проведення шляху, кожен параметр якої характеризує шлях з точки зору одного з критеріїв якості (довжини шляху, числа перетинів і тому подібне). На P_k накладають одне обмеження – ваги осередків попередніх фронтів не має бути більше ваг осередків наступних фронтів.

Фронт поширюється тільки на сусідні осередки, які мають з осередками попереднього фронту або загальну сторону, або хоч би одну загальну точку. Процес поширення хвилі триває до тих пір, поки її фронт, що розширюється, не досягне приймача або на Θ -му кроці не знайдеться жодного вільного осередку,

який міг би бути включений в черговий фронт, що відповідає випадку неможливості проведення траси при заданих обмеженнях.

Якщо в результаті поширення хвиля досягла приймача, то здійснюють "проведення шляху", яке полягає в русі від приймача до джерела по пройденим на етапі поширення хвилі осередкам, стежачи за тим, щоб значення R_k монотонно убували. В результаті отримують шлях, що сполучає ці дві точки. З опису алгоритму виходить, що усі умови, необхідні для проведення шляху, закладаються в правила приписання ваги осередкам.

Щоб виключити невизначеність при проведенні шляху для випадку, коли декілька осередків мають однакову мінімальну вагу, вводять поняття путніх координат, задаючих перевагу проведення траси. Кожен напрям кодується двійковим числом по **mod** q , де q – число сусідніх осередків, що переглядаються. При цьому ніж прийнятніше те або інший напрям, тим менший числовий код воно має. Наприклад, якщо задатися пріоритетним порядком проведення шляху згори, справа, знизу і ліворуч, то коди відповідних путніх координат будуть 00, 01, 10, і 11. Приписування путніх координат виробляють на етапі поширення хвилі. При проведенні шляху рух від осередку до осередку здійснюють по путніх координатах.

Істотними недоліками хвильового алгоритму є мала швидкодія і великий об'єм оперативної пам'яті ЕОМ, необхідний для зберігання інформації про поточний стан усіх осередків комутаційного поля, можливість побудови лише з'єднань типу "уведення-виведення". Спроби усунути вказані недоліки привели до створення ряду модифікацій хвильового алгоритму.

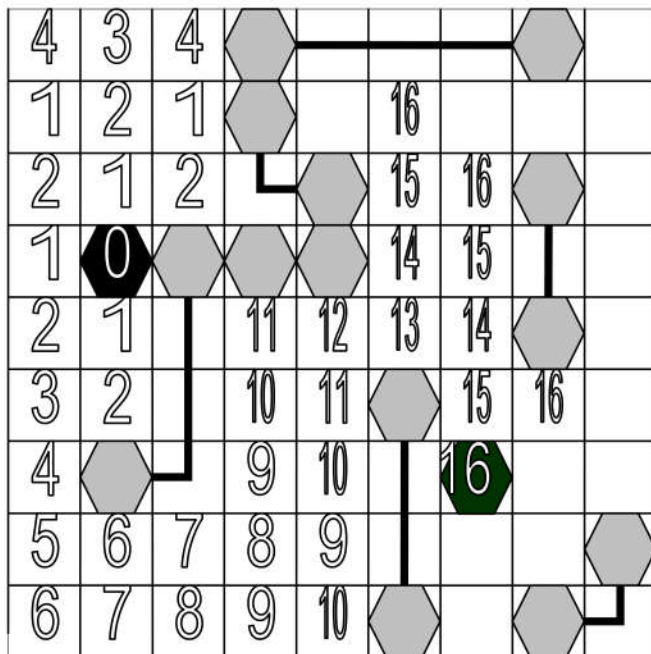


Рисунок 3.27 – Принцип проведення траси при хвильовому алгоритмі

Модифікації алгоритму Лі. Метод зустрічної хвилі

У цьому методі джерелами хвиль є обидва осередки, що підлягають електричному об'єднанню. При цьому на кожному k -му кроці по черзі будують відповідні фронти першою і другою хвиль, що поширюються з цих осередків. Процес триває до тих пір, поки який-небудь осередок з фронту першої хвилі не потрапить на фронт другої хвилі або навпаки. Проведення шляху здійснюють з цього осередку у напрямі обох джерел за правилами, описаними Лі в хвилевому алгоритмі.

Оцінимо число осередків, що переглядаються на етапі поширення хвилі, при використанні як джерела однієї або двох об'єднаних точок. Нехай відстань між цими точками R . Тоді для першого випадку у момент досягнення хвилею осередку-приймача площа проглянутої околиці має величину $Q^{(2)} \leq \pi R^2$ (знак рівності відповідає відсутності перешкод шляху поширення хвилі). Для другого випадку у момент зустрічі фронтів двох хвиль площа проглянутої околиці $Q^{(2)} \leq 2\pi(0,5R)^2 = 0,5\pi R^2$.

Таким чином, при використанні методу зустрічної хвилі площа, що переглядається, а отже, і час, що витрачається на етапі поширення хвилі, зменшуються приблизно удвічі.

Недоліком методу є необхідність виділення додаткового розряду пам'яті на кожну робочу комірку поля для зберігання інформації про приналежність її до першої або другої хвилі. Проте виграш в підвищенні швидкодії виконує вказаний недолік, тому цей метод використовують в усіх випадках, коли це дозволяє об'єм оперативної пам'яті ЕОМ.

Променевий алгоритм трасування

У цьому алгоритмі, запропонованим Л.Б. Абрайтисом, вибір осередків для визначення шляху між точками А і В, що сполучаються, виробляють по заздалегідь заданих напрямках, подібних променях. Це дозволяє скоротити число осередків, що переглядаються алгоритмом, а отже, і час на аналіз і кодування їх стану, проте призводить до зниження вірогідності знаходження шляху складної конфігурації, і ускладнює облік конструктивних вимог до технології друкованої плати.

