

Перехідні процеси в логічних схемах

Логічні ключі характеризуються часом затримки t_3 , який пов'язаний з перехідними процесами зміни стану. Тривалість перехідних процесів не є стабільною величиною і залежить від багатьох факторів. При використанні мікросхем, в яких внутрішні затримки малі, при проектуванні цифрових пристроїв слід обов'язково враховувати тривалість перехідних процесів у лініях зв'язку та вплив параметрів ліній зв'язку на тривалість перехідних процесів в мікросхемах. Ці задачі досить складні, і їх розв'язання можливе лише з використанням спеціальних методів моделювання.

При використанні мікросхем з тривалістю зміни станів, не меншою 20 нс (ТТЛ, КМОН), перехідні процеси в лініях зв'язку не враховуються, а характер лінії зв'язку, величина та характер навантаження враховуються максимальною тривалістю затримки. Це значно спрощує процес проектування цифрових пристроїв, адже затримки можуть бути враховані відразу на стані логічного проектування. В таких ситуаціях, коли мають місце значні затримки в лініях зв'язку, вони також можуть враховуватись як дискретні затримки.

На жаль, у паспортних даних на мікросхеми подається тільки максимальна тривалість затримки. Реально ж вона може змінюватись в широких межах, а мінімальна величина затримки паспортно не визначена. Звідси витікає, що рівень виходу елемента в цифровій схемі в інтервалі часу від 0 до t_3 не визначений. Він називається *станом невизначеності* і позначається символом x_H . Такий стан елемента в схемі впливає на інші елементи, на виходах яких можуть мати місце як визначені стани "1", "0", так і невизначений стан x_H . Як результат, поведінка логічних елементів в перехідних процесах описується законами трійкової логіки з такими логічними співвідношеннями:

$$\begin{array}{lll} \text{—} & x_H + 0 = x_H ; & x_{H1} \cdot x_{H2} = x_H ; \\ x_H = x_H ; & x_H \oplus 0 = x_H ; & x_{H1} + x_{H2} = x_H ; \\ x_H \cdot 1 = x_H ; & x_H + 1 = 1 ; & x_{H1} \oplus x_{H2} = x_H \cdot \\ x_H \cdot 0 = 0 ; & x_H \oplus 1 = x_H ; & \end{array}$$

Приведені формули широко використовуються при аналізі перехідних процесів, в тому числі при побудові часових діаграм. Останні зображаються одним з двох способів, які приведені на рис. 1, *а* для елемента **2I** ($y = x_0 \cdot x_1$).

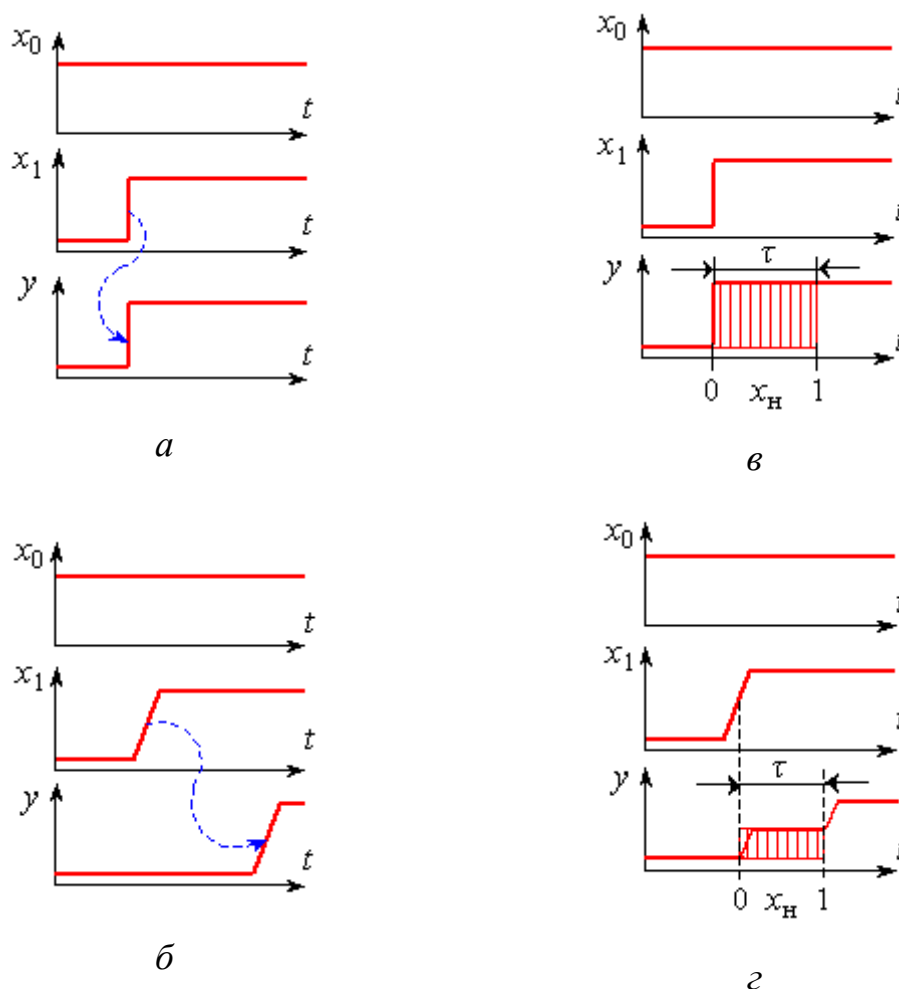


Рис. 1

У практиці аналізу схем використовуються чотири типи зображення перехідних процесів, які використовуються в залежності від необхідного рівня деталізації. Найпростіша з них – діаграма рис. 1, *а* – пояснює лише логічні зв'язки між вхідними сигналами та вихідними станами без урахування вихідних затримок.

Часова діаграма, приведена на рис. 1, *б*, враховує часові затримки, причому їх максимальне значення. Вона дає можливість оцінити максимальні значення тривалості перехідних процесів. Лінії зі стрілками показують взаємозв'язки між вхідними сигналами і вихідними станами мікросхеми.

Діаграми рис. 1, в, з враховують стани невизначеності елементів. Вони дають можливість змоделювати поведінку схеми при будь-яких комбінаціях і часових співвідношеннях внутрішніх затримок. Інтервал τ невизначеного стану x_H дорівнює величині t_3 конкретної мікросхеми.

Врахування станів невизначеності дає можливість виявити важливий ефект, який обов'язково необхідно враховувати при аналізі цифрових схем.

Розглянемо схему, що приведена на рис. 2, а, на вхід якої подається сигнал x_0 у вигляді одиночного імпульсу тривалістю T .

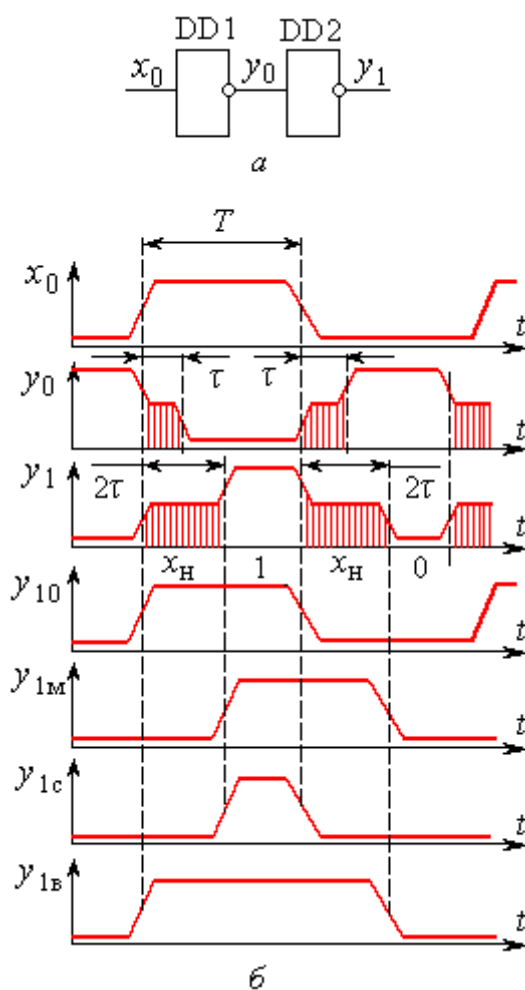


Рис. 2

Якщо тривалості інтервалів невизначеності τ при зміні станів мікросхем з “0” в “1” і з “1” в “0” однакові, то на виході y_1 матимемо сигнал, тривалість і початок якого суттєво залежить від величини затримки τ та співвідношення між затримками при зміні станів з “0” в “1” та навпаки. Якщо, наприклад, тривалість затримок $\tau=0$, то вихідний імпульс y_{10} повторюватиме вхідний.

Якщо затримки на зміну станів з “1” в “0” і навпаки однакові і максимальні, то сигнал y_{1M} матиме затримку по відношенню до вхідного на величину 2τ .

При різних співвідношеннях між затримками вихідний сигнал може бути як скороченої довжини y_{1c} , так і видовженої y_{1b} (рис. 2, б).

Звичайно, що реально сигнал y_1 може мати випадкові співвідношення між тривалістю імпульсу та паузою, випадкову затримку початку і кінця імпульсу. Як результат, імпульс, який буде пропущений через два ідентичні канали, на їх виходах може суттєво відрізнятись. Якщо ж один канал має m елементів, а другий n , то, відповідно, матимемо затримки на $m\tau$ і $n\tau$, і вихідні імпульси можуть навіть не співпадати в часі. Якщо при проектуванні схеми, що має послідовно з'єднані k елементи, необхідно одержати імпульс тривалістю t_1 , то в результаті вихідний імпульс з двостороннім допуском буде мати тривалість $t_1 \pm k\tau$.

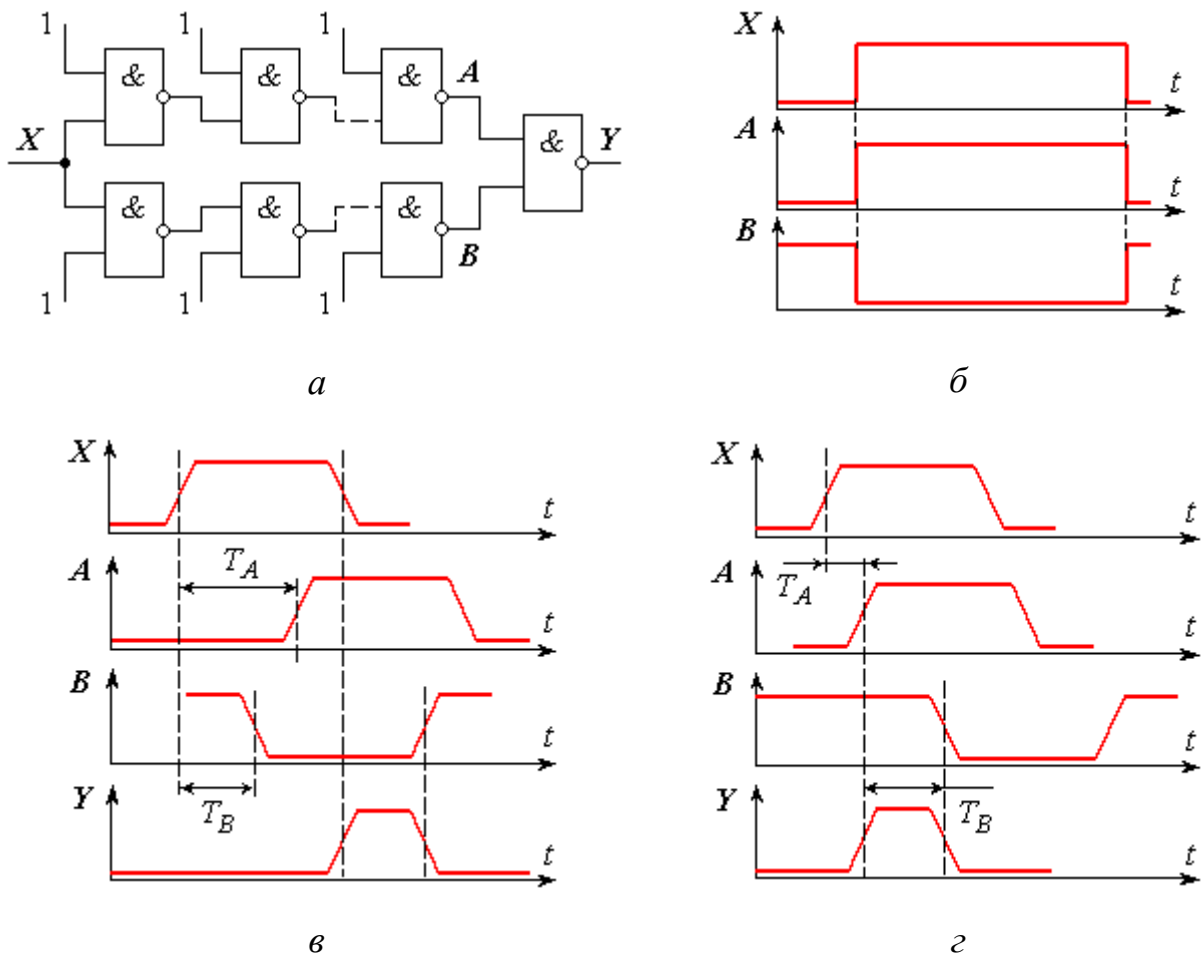


Рис. 3

Найбільші проблеми наявність затримок створює при передачі імпульсів через паралельні канали. Розглянемо схему, що приведена на рис. 3, а. Вхідний імпульс подається на два паралельних канали з елементами **I-НІ**, один з яких назвемо *A* – з парною кількістю елементів, а другий *B* – з непарною. При наявності логічних одиниць на одному з входів кожного логічного елемента обидва канали є прозорими для сигналу *x*. При відсутності часових затримок в каналах, їх вихідні сигнали *A* і *B* на вході об'єднуючого елемента **I-НІ** будуть строго протифазні і, як результат, $y = 0$ (рис. 3, б).

На рис. 3, в, з приведені приклади часових діаграм в тих ситуаціях, коли, відповідно, затримки $T_A > T_B$ і $T_B > T_A$. В обох випадках на виході *y* з'явиться сигнал, який неможливо передавати на основі алгебри Буля без врахування часових затримок. Одержаний сигнал є сигналом перешкоди, що може привести до непередбачених наслідків в послідовних схемах.

Розглянуте явище називається „гонками” або „змаганнями”. Головна проблема, яка пов'язана з гонками, полягає у тому, що проєктант не має реальних можливостей визначити момент появи можливої перешкоди та її тривалість.

Гонки називаються *критичними* або *недопустимими*, якщо хоча б один вихідний сигнал під час перехідного процесу змінюється більш ніж один раз. Критичні змагання суттєво впливають на роботу цифрових пристроїв та їх проєктування.

Розглянемо схему, що приведена на рис. 4. При $x = 1$ на обох виходах y_{τ_1} і y_{τ_2} матимемо сигнали низького рівня. Зміна *x* з “1” в “0” приведе до того, що на виході з меншим значенням τ (наприклад, $\tau_2 < \tau_1$) з'явиться одиничний сигнал раніше, ніж на іншому. Він забезпечить підтримку одиничного значення на вході DD1, внаслідок чого $y_{\tau_1} = 0$. Оскільки значення τ_1 і τ_2 невідомі, то такий перехід приведе до невизначеності станів y_{τ_1} і y_{τ_2} .

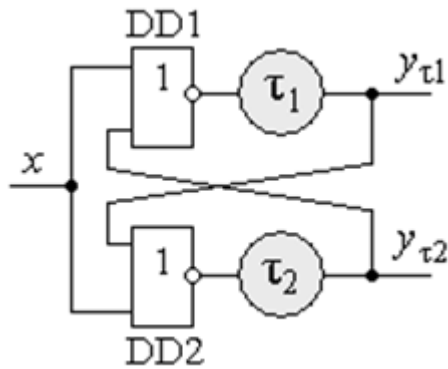


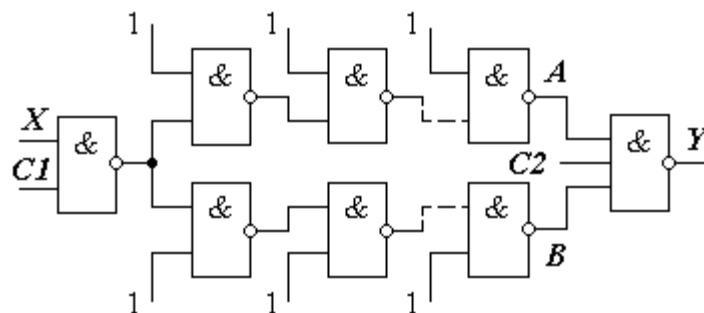
Рис. 4

На практиці використовуються три наступні способи боротьби з гонками:

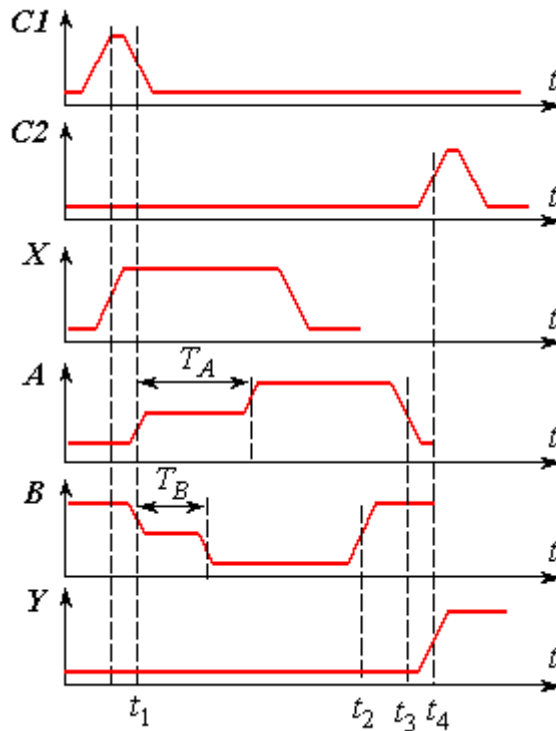
- *синхронізація;*
- *побудова протигоночних систем;*
- *врахування мінімального часу затримки.*

Синхронізація є найбільш універсальним засобом боротьби з гонками. Її суть полягає у наступному: по всьому цифровому пристрою створюється єдина система синхронізуючих сигналів. У залежності від типу елементної бази, використовуються однофазна або двофазна системи синхронізації. В останньому випадку синхронізація забезпечується двома послідовностями імпульсів *C1* та *C2*, одна з яких має затримку по відношенню до іншої на половину періоду.

Для розглянутої вище схеми з паралельними каналами (див. рис. 3, *a*) синхронна схема з двофазною синхронізацією приведена на рис. 5, *a*.



a



б

Рис. 5

За синхросигналом $C1$ сигнал x одночасно подається на два канали. Проходячи канали з різними затримками T_A і T_B , в асинхронній схемі (рис. 3, а) на виході Y в інтервалі часу $t_2 \div t_3$ з'являється перешкода. Але у синхронній схемі інформація на виході Y з'явиться лише після подачі синхроімпульсу $C2$ в момент t_4 , коли всі перехідні процеси в каналах завершаться. Важливе значення в цьому випадку має інтервал $t_1 \div t_4$ між двома синхроімпульсами, який вибирається таким, щоб перехідні процеси найбільшої тривалості могли завершитись до подачі $C2$.

У залежності від того, що виступає приймачем сигналу Y , залежить назва сигналу $C2$. В розглянутому випадку, коли сигнал Y не запам'ятовується, а $C2$ використовується лише для того, щоб в Y не з'являлась перешкода, він називається *стробом*, а процес відсікання перешкод – *стробуванням*. Якщо ж необхідно запам'ятати в схемі пам'яті, то сигнал $C2$ подається на цю схему, що робить її синхронною, а $C2$ – відповідно *синхросигналом*.

Слід зазначити, що синхронізація дає можливість суттєво спростити процес проектування цифрових схем, адже значно спрощує вирішення проблеми гонок.

Протигоночні системи – це другий практичний спосіб боротьби з гонками. Вони будуються так, що в них відсутній ризик появи на виході сигналів, не передбачених логікою роботи схеми. Прикладом подвійної схеми може бути два паралельних канали з однаковою кількістю елементів, об'єднаних елементом **АБО**; який би канал не виграв гонку, результат буде однаковим, зміниться лише момент його появи.

Для усунення гонок необхідно створювати паралельні канали передачі сигналів. Задача створення мінімально необхідної кількості паралельних каналів досить чітко формалізується при використанні карт Карно (діаграм Вейча). Сутність такої формалізації полягає в тому, щоб будь-які сусідні клітини карти Карно були охоплені щонайменше одним багатовимірним кубом. Мінімізована таким чином логічна функція буде вільною від гонок.

Практично асинхронні схеми достатньо складні в проектуванні і поступово витісняються синхронними.

Приклад 3.22. Логічна функція задана картою Карно (рис. 6). Мінімізувати функцію для отримання диз'юнктивної нормальної форми, вільної від гонок.

		x_1x_0			
		00	01	11	10
x_3x_2	00	⁰ 0	¹ 1	³ 1	² 0
	01	⁴ 0	⁵ 1	⁷ 1	⁶ 1
	11	¹² 1	¹³ 1	¹⁵ 0	¹⁴ 1
	10	⁸ 1	⁹ 1	¹¹ 0	¹⁰ 0

Рис. 6

Розв'язання. Для забезпечення диз'юнктивної форми можна об'єднати наступні групи клітинок: група № 1 – **1, 3, 5, 7**; група № 2 – **8, 9, 12, 13** і група № 3 – **6, 14**. В результаті отримаємо функцію:

$$y = \overline{x_3} \cdot x_0 + \overline{x_3} \cdot x_1 + x_2 \cdot x_1 \cdot \overline{x_0}.$$

Функція буде мінімальною, але не вільною від гонок, оскільки такі сусідні клітини, як **6, 7; 1, 9; 12, 14** не охоплюються багатовимірними кубами. Виконання умови відсутності

гонок приводить до необхідності об'єднання вказаних клітин, що може бути забезпечено двома 2-кубами (7, 6) і (12, 14) і одним 4-кубом (1, 5, 13, 9). В результаті отримаємо функцію:

$$y = \overline{x_3} x_0 + x_3 \overline{x_1} + x_2 x_1 \overline{x_0} + x_3 x_2 x_1 + x_3 x_2 \overline{x_0} + x_1 x_0,$$

яка буде вільною від гонок.

Третій спосіб – *урахування мінімального часу затримок* – знаходить використання при проектуванні таких цифрових схем, які потім виготовлятимуться на одному кристалі. Це пов'язано з тим, що проєктант паралельно зі схемотехнікою закладає й технологічні особливості виготовлення схеми з необхідними параметрами. В результаті у проєкт можна закласти необхідні часові затримки в окремих каналах або схеми. Використання таких прийомів у дискретній цифровій схемотехніці вимагає великого досвіду і уваги.

У будь-якому випадку, після проектування цифрових схем їх аналізують на предмет можливої появи перешкод в результаті гонок. На практиці з цією метою використовуються спеціальні комп'ютерні програми.

Ще одним прикладом появи перешкод завдяки гонкам є прийом сигналів з недостатньо крутими фронтами. Така ситуація має місце тоді, коли подвійний сигнал подається на паралельні канали з різними типами мікросхем. У такому випадку в деякому інтервалі рівнів вхідного сигналу для мікросхеми одного типу сигнал відповідатиме рівню логічного нуля, а для іншого – як логічної одиниці. Це приведе до появи хибних сигналів. Це явище називається „*гонками по виходу*”. Для боротьби з ним використовуються спеціальні прийоми попередньої обробки вхідних сигналів.