

ЛЕКЦІЯ 6

Комбінаційні цифрові пристрої

1. Поняття комбінаційного цифрового пристрою

Комбінаційний цифровий пристрій (combinational digital circuit) — схема, у якій поточний вектор виходів однозначно визначається поточним вектором входів. У математичній формі це записують як $Y(t)=F(X(t))$. Внутрішній стан, що зберігав би попередні значення, відсутній.

Відсутність пам'яті не означає миттєвої реакції. Після зміни входів реальна схема проходить перехідний процес, обумовлений затримками логічних елементів. Після стабілізації однаковий набір входів завжди формує однаковий набір виходів.

Комбінаційними є дешифратори, шифратори, мультиплексори, демультимплексори, компаратори, суматори та багато інших вузлів. Їх можна описати таблицями істинності, системами булевих функцій або з'єднанням стандартних модулів.

2. Методика синтезу комбінаційної схеми

1. Сформулювати умову так, щоб для кожного допустимого набору входів існував однозначний результат.
2. Визначити назви, зміст, розрядність та активні рівні входів і виходів.
3. Побудувати повну таблицю істинності або функціональну таблицю стандартного вузла.
4. Записати окрему булеву функцію для кожного виходу.
5. Мінімізувати функції, якщо це зменшує складність реалізації.
6. Побудувати схему з логічних елементів або стандартних функціональних блоків.
7. Перевірити всі вхідні набори, а також enable, valid і невикористані коди.
8. Оцінити критичний шлях, затримку та можливі короткочасні глітчi.

2.1. Приклад формалізації словесної умови

Нехай $D=1$ означає відчинені двері, $M=1$ — активний режим охорони, $K=1$ — правильний ключ розпізнано. Тривога A повинна вмикатися, коли режим охорони активний, двері відчинені, а правильного ключа немає. Отже, $A=M \wedge D \wedge \bar{K}$. Лише набір $M=1, D=1, K=0$ формує $A=1$.

3. Дешифратори

Дешифратор (decoder) перетворює n -бітний двійковий код на один із максимум 2^n виходів. В active-high моделі для кожного дозволеного коду одиницю має один вихід, а решта дорівнюють нулю. Такий формат називають one-hot — активний один сигнал із групи.

Для active-low виходів вибраний сигнал дорівнює нулю, а невибрані — одиниці. Полярність позначається інверсійними бульбашками або суфіксом $_n$ у назві. Функцію завжди потрібно читати разом з активним рівнем.

3.1. Дешифратор 2 у 4

A_1	A_0	Y_0	Y_1	Y_2	Y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Номер активного виходу дорівнює числовому значенню коду A_1A_0 . Для $10_2=2_{10}$ активується Y_2 . Виходи є мінтермами двох змінних: $Y_0=\bar{A}_1\bar{A}_0$, $Y_1=\bar{A}_1A_0$, $Y_2=A_1\bar{A}_0$, $Y_3=A_1A_0$.

3.2. Вхід дозволу

Enable (EN — дозвіл) керує всім дешифратором. Для active-high EN декодування виконується при $EN=1$, а при $EN=0$ усі active-high виходи неактивні. В active-low моделі $EN_n=0$ може бути дозволеним станом. Це визначається символом і таблицею конкретного компонента.

3.3. Каскадування

Дешифратор 3 у 8 можна побудувати з двох дешифраторів 2 у 4. Молодші біти A_1A_0 подаються на обидва блоки, а старший A_2 дозволяє лише один із них. При $A_2=0$ працює група $Y_0...Y_3$, при $A_2=1$ — група $Y_4...Y_7$. Така ієрархія збільшує розрядність, але додає затримку рівнів дозволу.

3.4. Реалізація функції через дешифратор

Нехай $F(A,B,C)=\Sigma m(1,2,6,7)$. Дешифратор 3 у 8 формує всі мінтерми $Y_0...Y_7$. Потрібна функція отримується об'єднанням виходів, індекси яких перелічені в сумі мінтермів: $F=Y_1 \vee Y_2 \vee Y_6 \vee Y_7$. Для active-low дешифратора конкретний об'єднувальний елемент змінюється відповідно до законів де Моргана.

4. Шифратори

Шифратор (encoder) виконує зворотне перетворення: за окремою активною вхідною лінією формує її двійковий номер. Для 2^n входів потрібно n кодових виходів. Простий шифратор припускає, що активний рівно один вхід.

4.1. Шифратор 4 у 2

D_0	D_1	D_2	D_3	Q_1	Q_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

Рівняння простого шифратора: $Q_1=D_2 \vee D_3$, $Q_0=D_1 \vee D_3$. Якщо одночасно активні D_1 і D_2 , обидва виходи стають одиницями та утворюють код 11, хоча D_3 не активний. Отже, без додаткового правила кілька активних входів створюють неоднозначність.

5. Пріоритетний шифратор

Пріоритетний шифратор (priority encoder) визначає код входу з найвищим пріоритетом. Нехай D_3 важливіший за D_2 , D_2 — за D_1 , а D_1 — за D_0 . Значення X у таблиці означає don't care: воно не впливає на результат, коли вже активний старший вхід.

D_3	D_2	D_1	D_0	Q_1	Q_0	V
1	X	X	X	1	1	1
0	1	X	X	1	0	1
0	0	1	X	0	1	1

D_3	D_2	D_1	D_0	Q_1	Q_0	V
0	0	0	1	0	0	1
0	0	0	0	0	0	0

Вихід V (valid — дійсність) дорівнює 1, якщо активний хоча б один вхід. Він відрізняє стан $D_0=1$, для якого код 00 є дійсним, від стану без запитів, де кодові виходи також можуть дорівнювати 00, але $V=0$.

5.1. Покроковий приклад

Для $D_3D_2D_1D_0=0111$ активні D_2 , D_1 і D_0 . Найвищий активний вхід — D_2 , тому $Q_1Q_0=10$, $V=1$. Молодші запити не відображаються в поточному коді. Після обслуговування D_2 система може скинути його та отримати код наступного запиту.

6. Мультиплексори

Мультиплексор (multiplexer, MUX) обирає один із кількох інформаційних входів і передає його на єдиний вихід. Для 2^n інформаційних входів потрібно n адресних входів вибору. MUX також називають селектором даних.

6.1. Мультиплексор 2 у 1

MUX 2 у 1 має інформаційні входи I_0 та I_1 , вхід вибору S і вихід Y . При $S=0$ вихід повторює I_0 , при $S=1$ — I_1 . Булева функція: $Y = \bar{S}I_0 \vee SI_1$. Добутки взаємовиключні, бо містять S і $\bar{S}$.

6.2. Мультиплексор 4 у 1

S_1	S_0	Вибраний вхід	Вихід
0	0	I_0	$Y=I_0$
0	1	I_1	$Y=I_1$
1	0	I_2	$Y=I_2$
1	1	I_3	$Y=I_3$

Повна функція: $Y = \bar{S}_1\bar{S}_0I_0 \vee \bar{S}_1S_0I_1 \vee S_1\bar{S}_0I_2 \vee S_1S_0I_3$. Логіка вибору утворює чотири one-hot сигнали, AND пропускають дані вибраного каналу, а OR формує спільний вихід.

7. Реалізація булевої функції через мультиплексор

Розглянемо $F(A,B,C) = \sum m(1,2,6,7)$. Для MUX 4 у 1 призначимо $S_1=A$, $S_0=B$. Для кожного фіксованого коду AB порівнюються два значення F при $C=0$ та $C=1$. Пара 00 відповідає константі 0, 11 — константі 1, 01 — C , 10 — $\bar{C}$.

AB	F при $C=0$	F при $C=1$	Підключення
00	0	1	C
01	1	0	$\bar{C}$

AB	F при C=0	F при C=1	Підключення
10	0	0	0
11	1	1	1

1. AB=00: набори 000 і 001 дають 0 та 1, тому $I_0=C$.
2. AB=01: набори 010 і 011 дають 1 та 0, тому $I_1=\bar{C}$.
3. AB=10: набори 100 і 101 дають 0 та 0, тому $I_2=0$.
4. AB=11: набори 110 і 111 дають 1 та 1, тому $I_3=1$.

Остаточне підключення: $I_0=C$, $I_1=\bar{C}$, $I_2=0$, $I_3=1$. Перевірка дає одиниці лише для наборів 001, 010, 110 і 111, тобто саме для мінтермів 1, 2, 6 і 7.

7.1. Реалізація через MUX 8 у 1

Якщо A, B і C подати на три адресні входи MUX 8 у 1, кожний I_k відповідає одному рядку таблиці. Для заданої функції I_1 , I_2 , I_6 та I_7 під'єднуються до 1, а решта — до 0. Реалізація не потребує зовнішньої логіки, але використовує більший мультиплексор.

7.2. Каскадування MUX

MUX 8 у 1 можна побудувати з двох MUX 4 у 1 та одного MUX 2 у 1. Молодші біти S_1S_0 вибирають канал усередині кожної групи, а старший S_2 обирає групу. Сигнал проходить два рівні мультиплексування, тому затримки каскадів додаються.

8. Демультимплексори

Демультимплексор (demultiplexer, DEMUX) спрямовує один інформаційний вхід D на один із кількох виходів. Адресні входи визначають маршрут. Невибрані active-high виходи дорівнюють нулю.

8.1. Демультимплексор 1 у 4

D	S_1	S_0	Y_0	Y_1	Y_2	Y_3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Рівняння: $Y_0=D\bar{S}_1\bar{S}_0$, $Y_1=D\bar{S}_1S_0$, $Y_2=DS_1\bar{S}_0$, $Y_3=DS_1S_0$. При D=1 DEMUX поводить себе як active-high дешифратор. При D=0 усі виходи нульові незалежно від адреси.

8.2. Відмінність від дешифратора

Дешифратор інтерпретує код і формує one-hot ознаку його значення. DEMUX використовує той самий код як маршрут для окремого сигналу D. Схеми близькі, але функціональна роль інтерфейсу різна. Вхід enable дешифратора часто можна використати як D для демультимплексування.

9. Цифрові компаратори

Цифровий компаратор (digital comparator) порівнює два двійкові слова та формує ознаки G (greater — більше), E (equal — рівно) і L (less — менше). Для коректної пари беззнакових чисел активний рівно один із трьох виходів.

9.1. Однорозрядний компаратор

A	B	G: A>B	E: A=B	L: A<B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

Рівняння: $G=A\bar{B}$; $E=A\odot B=A\bar{B}\vee \bar{A}B$; $L=\bar{A}B$. Символ $\odot$ позначає XNOR — еквівалентність. Для кожного рядка $GVE=L=1$, а попарні кон'юнкції виходів дорівнюють нулю.

9.2. Багаторозрядний компаратор

Порівняння починається зі старшого розряду MSB (Most Significant Bit — найстарший значущий біт). Перший різний розряд визначає результат, а молодші біти враховуються лише за рівності всіх старших пар.

Для двох бітів $E=(A_1\odot B_1)\wedge(A_0\odot B_0)$. Умова $A>B$: $G=A_1\bar{B}_1\vee(A_1\odot B_1)A_0\bar{B}_0$. Умова $A<B$ будується симетрично. Наприклад, $A=10_2$ і $B=01_2$: старші біти 1 та 0 одразу дають $A>B$, що відповідає $2>1$ у десятковій системі.

10. Комплексний приклад маршрутизації

Нехай S_1S_0 задають номер одного з чотирьох каналів, а EN дозволяє роботу. MUX 4 у 1 передає I_s на спільний вихід Y. Дешифратор 2 у 4 активує керувальний сигнал C_s для того самого каналу. При $EN=0$ Y та всі C неактивні. При $EN=1$ і $S_1S_0=10$ отримуємо $Y=I_2$ та $C_2=1$.

Приклад показує, що одна адресна шина може керувати кількома узгодженими модулями: вибирати джерело даних, периферійний пристрій і локальний сигнал дозволу.

11. Часові властивості

Комбінаційний модуль має propagation delay — затримку поширення. У каскаді затримки послідовних рівнів додаються. Після зміни адреси MUX або дешифратора виходи не слід вважати правильними, доки не мине максимальний час стабілізації критичного шляху.

Якщо кілька адресних бітів змінюються майже одночасно, різні затримки можуть утворити проміжний код і короткочасно активувати неправильний канал. Такий імпульс називають glitch, а структурну можливість його виникнення — hazard. Синхронне зчитування після стабілізації зменшує ризик поширення перехідних значень.

12. Типові помилки

- Переплутані ваги старшого та молодшого адресних бітів.
- Active-low сигнал прочитано як active-high.
- Простий шифратор використано за кількох активних входів.
- Код 00 шифратора не відрізнено від відсутності запиту через відсутній valid.

- У реалізації через MUX переплутано входи вибору та інформаційні входи.
- Перевірено лише кілька наборів замість повної таблиці.
- Не враховано затримку каскадованих блоків і глітчів.

13. Послідовність перевірки

1. Звірити назви, напрямки, розрядність і активні рівні сигналів.
2. Перевірити всі дозволені вхідні коди.
3. Окремо перевірити enable, valid, заборонені та невикористані комбінації.
4. Порівняти результати з початковою таблицею істинності.
5. Перевірити one-hot або взаємовиключність виходів, якщо це властивість вузла.
6. Оцінити критичний шлях та момент гарантованої стабілізації.
7. Проаналізувати можливість короткочасної активації неправильного каналу.

14. Терміни та скорочення

Термін	Пояснення
Combinational logic	Комбінаційна логіка без внутрішнього збереженого стану.
Decoder	Дешифратор: перетворює двійковий код на окремий вибраний вихід.
Encoder	Шифратор: перетворює активний вхід на двійковий код.
Priority encoder	Пріоритетний шифратор: кодує найважливіший активний вхід.
MUX	Multiplexer — мультиплексор, селектор одного з кількох джерел.
DEMUX	Demultiplexer — демультимплексор, розподільувач одного сигналу.
Comparator	Компаратор: формує ознаки більше, рівно та менше.
One-hot	Кодування або стан, у якому активний рівно один сигнал із групи.
Enable	Вхід дозволу роботи цифрового вузла.
Valid	Ознака дійсності сформованого коду.
MSB	Most Significant Bit — найстарший значущий біт.
Don't care, X	Значення, яке не впливає на результат у заданому рядку.
Propagation delay	Затримка поширення зміни від входу до виходу.
Glitch	Короткочасний хибний імпульс під час перехідного процесу.

15. Узагальнення

Дешифратор виконує перетворення код $\rightarrow$ one-hot, шифратор — one-hot $\rightarrow$ код, MUX обирає одне джерело, DEMUX спрямовує одне джерело до вибраного виходу, а компаратор визначає відношення між числами. Усі вузли проєктуються за однаковою методикою: специфікація, таблиця, рівняння, структура та перевірка.

Дешифратор може реалізувати функцію як суму потрібних мінтермів. MUX реалізує функцію через вибір констант, змінної або її заперечення. Ці способи є систематичними й легко перевіряються повною таблицею істинності.