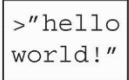
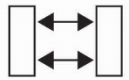
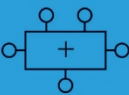
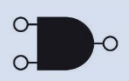
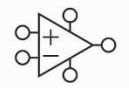
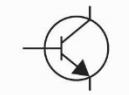
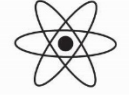


Розділ 3 - Теми

- Вступ
- Фіксатори і Тригери затримки
- Проектування синхронних логічних схем
- Скінченні автомати
- Синхронізація послідовнісних схем
- Паралелізм

Application Software	
Operating Systems	
Architecture	
Micro-architecture	
Logic	
Digital Circuits	
Analog Circuits	
Devices	
Physics	

Вступ

- Виходи послідовнісної схеми залежать як від поточних, *так і від* минулих значень на входах - схема має **пам'ять**.
- Деякі визначення:
 - **Стан:** Вся інформація про схему, необхідна для визначення її майбутньої поведінки
 - **Фіксатори і Тригери затримки:** Елементи, які зберігають один біт стану
 - **Синхроні послідовнісні схеми:** За комбінаційною схемою слідує набір тригерів.
 - Видають послідовність подій
 - Мають (короткочасну) пам'ять
 - Для зберігання інформації використовують зворотний зв'язок з виходів на входи

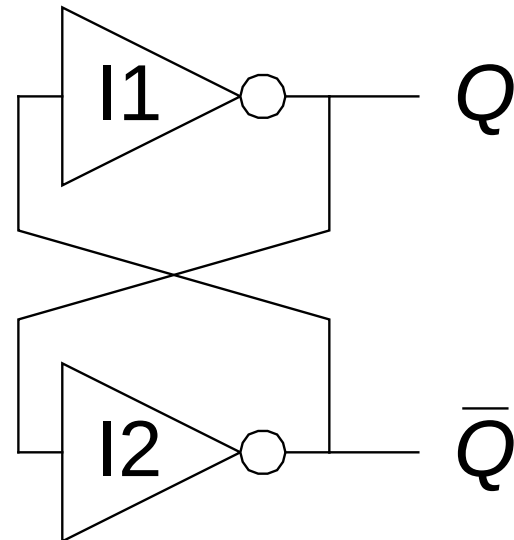
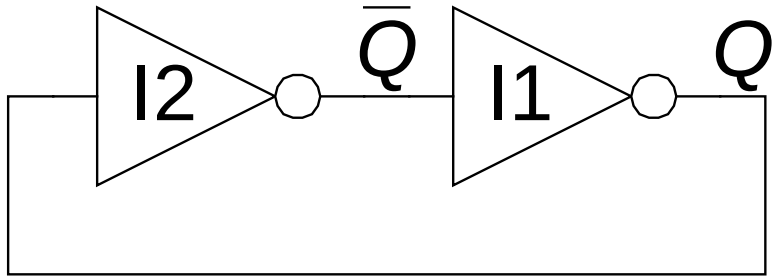
Елементи, які зберігають стан

- Стан схеми впливає на її майбутню поведінку
- Елементи, які зберігають стан схеми
 - Бістабільна схема
 - RS-тригер
 - D-фіксатор, latch (одноступеневий)
 - D-тригер затримки, flip-flop (двоступеневий)

Бістабільна схема

- Основной блок для побудови інших елементів, які зберігають стан
- Два виходи: Q , $\bar{Q}$
- Вхідів не має

Перехресно з'єднані інвертори



Аналіз бістабільної схеми

- Розглянемо два можливих випадки:

– $Q = 0$:

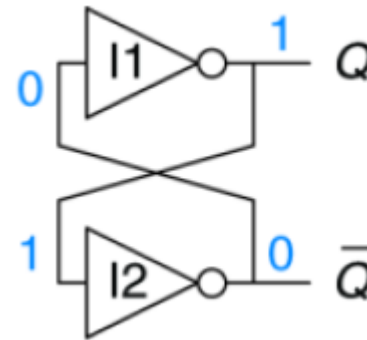
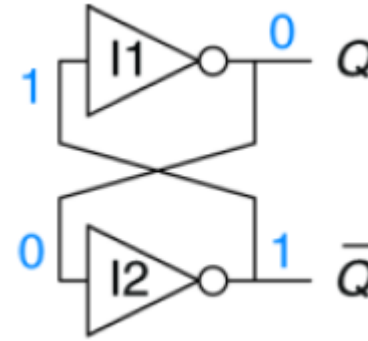
Тоді $\bar{Q} = 1$, $Q = 0$

(не має протиріччя)

– $Q = 1$:

Тоді $\bar{Q} = 0$, $Q = 1$

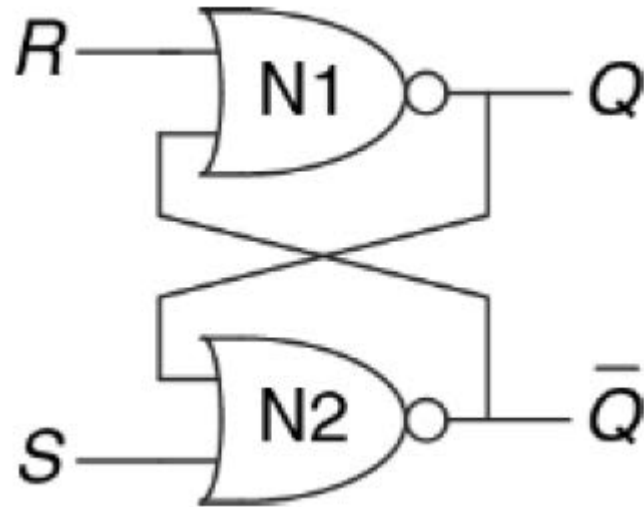
(не має протиріччя)



- Зберігає 1 біт змінної стану Q (або $\bar{Q}$)
- Відсутні входи керування станом

RS тригер

- RS-триггер
(Reset/Set,
Встановлення/Скидання)

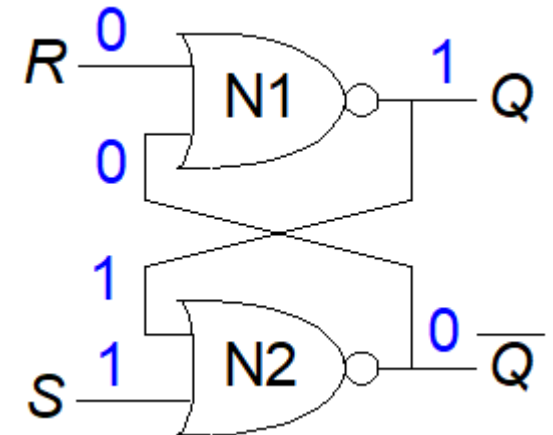
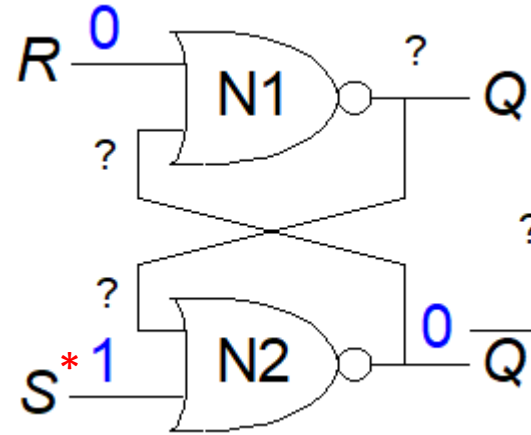


- Розглянемо чотири можливі випадки:
 - $S = 1, R = 0$
 - $S = 0, R = 1$
 - $S = 0, R = 0$
 - $S = 1, R = 1$

Аналіз RS-тригера без врахування попереднього стану

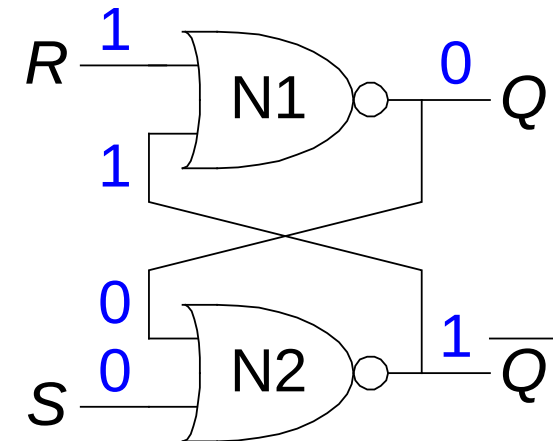
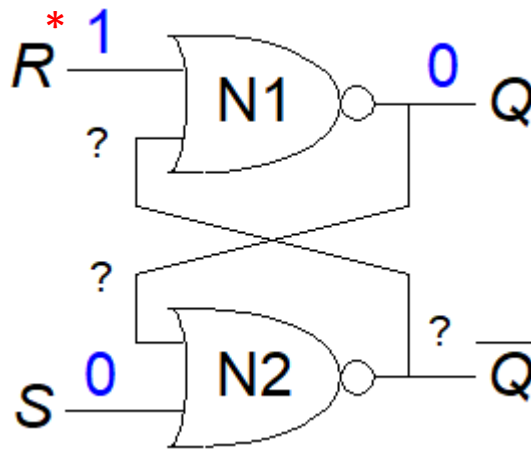
– $S = 1, R = 0$:

Тоді $Q = 1$ і $\overline{Q} = 0$



– $S = 0, R = 1$:

Тоді $\overline{Q} = 1$ і $Q = 0$

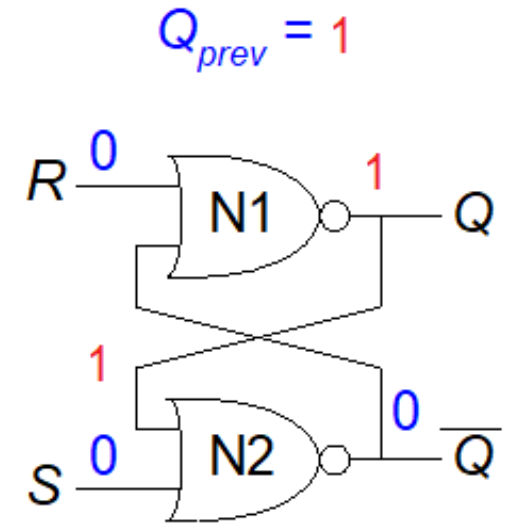
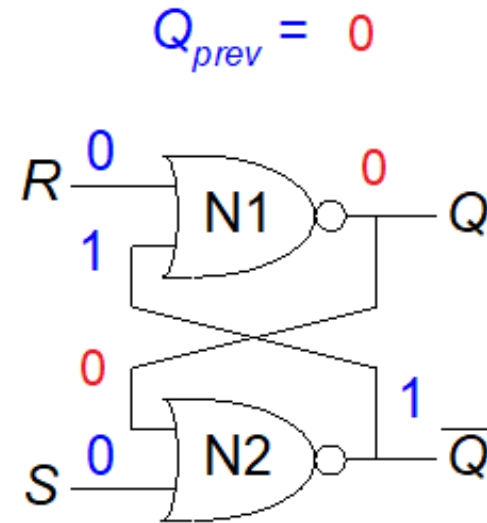


Аналіз RS-тригера з врахуванням попереднього тану

– $S = 0, R = 0$:

Тоді $Q = Q_{prev}$

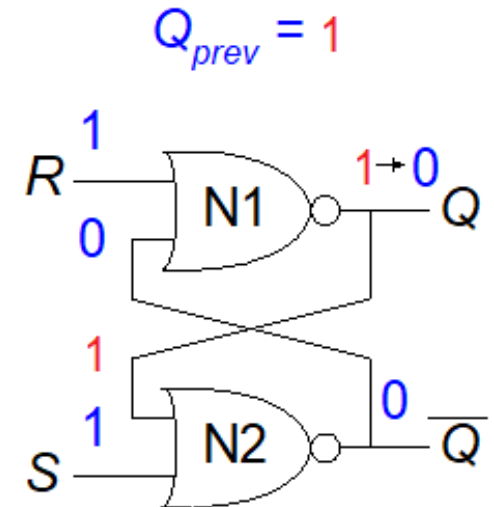
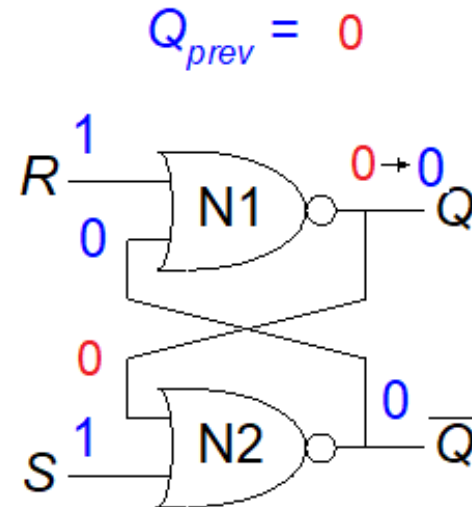
– Повторення попереднього стану



– $S = 1, R = 1$:

Тодс $Q = 0, \overline{Q} = 0$

– Заборонений стан $Q = \overline{Q}$



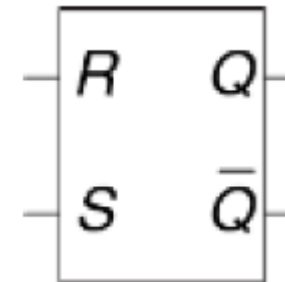
Позначення RS тригера

- RS позначає тригер із входами Reset/Set, Скидування/Встановлення
 - Зберігає один біт стану (Q)
- Значення, яке зберігається визначається входами S , R
 - **Set (Встановлення):**
Встановлює вихід в 1
($S = 1, R = 0, Q = 1$)
 - **Reset (Скидування):**
Встановлює вихід в 0
($S = 0, R = 1, Q = 0$)
- **Потрібно приймати спеціальні міри для уникнення забороненого стану ($S = R = 1$)**

Таблиця істинності RS-тригера

Case	S	R	Q	$\bar{Q}$
IV	0	0	Q_{prev}	$\bar{Q}_{prev}$
I	0	1	0	1
II	1	0	1	0
III	1	1	0	0

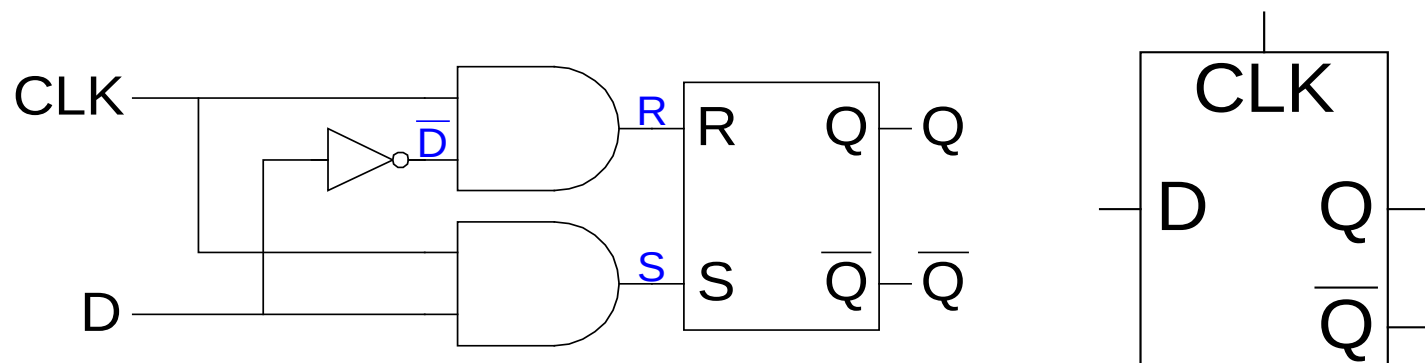
Позначення RS-тригера



D-фіксатор (latch)

- Два входи CLK , D
 - **CLK** : визначає *коли* змінюється вихід
 - **D** (вхід даних): визначає *нове значення* виходу
- Робота
 - Коли **$CLK = 1$** , D проходить на виход Q (фіксатор *відкритий*)
 - Коли **$CLK = 0$** , Q зберігає попереднє значення (фіксатор *закритий*)
- Заборонений стан $Q = \overline{Q}$ не виникає
- Фіксатор – тригер, який керується рівнем тактового сигналу
- Стан фіксатора змінюється неперервно поки **$CLK = 1$**

Внутрішня структура D-фіксатора

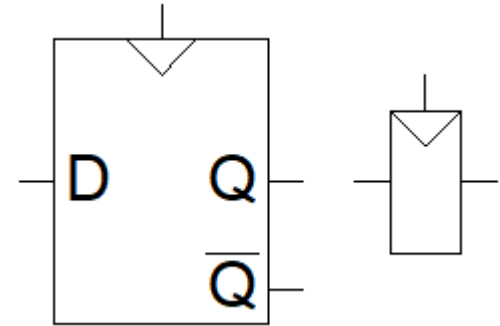


CLK	D	$\overline{D}$	S	R	Q	$\overline{Q}$
0	X	$\overline{X}$	0	0	Q_{prev}	$\overline{Q}_{prev}$
1	0	1	0	1	0	1
1	1	0	1	0	1	0

D-тригер затримки (flip-flop)

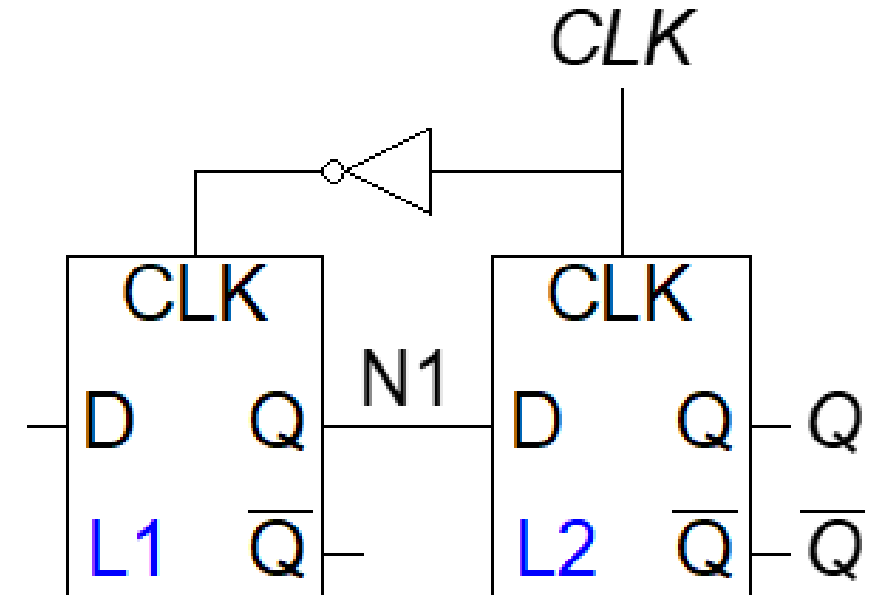
- **Входи:** CLK , D
- **Робота**
 - Фіксує значення D за *переднім* фронтом CLK
 - Коли CLK змінюється від 0 до 1,
 D проходить на вихід Q
 - В іншому випадку Q зберігає попереднє значення
 - Q змінюється тільки за переднім фронтом CLK
- D-тригер *керується фронтом* тактового сигналу

Позначення D-тригера

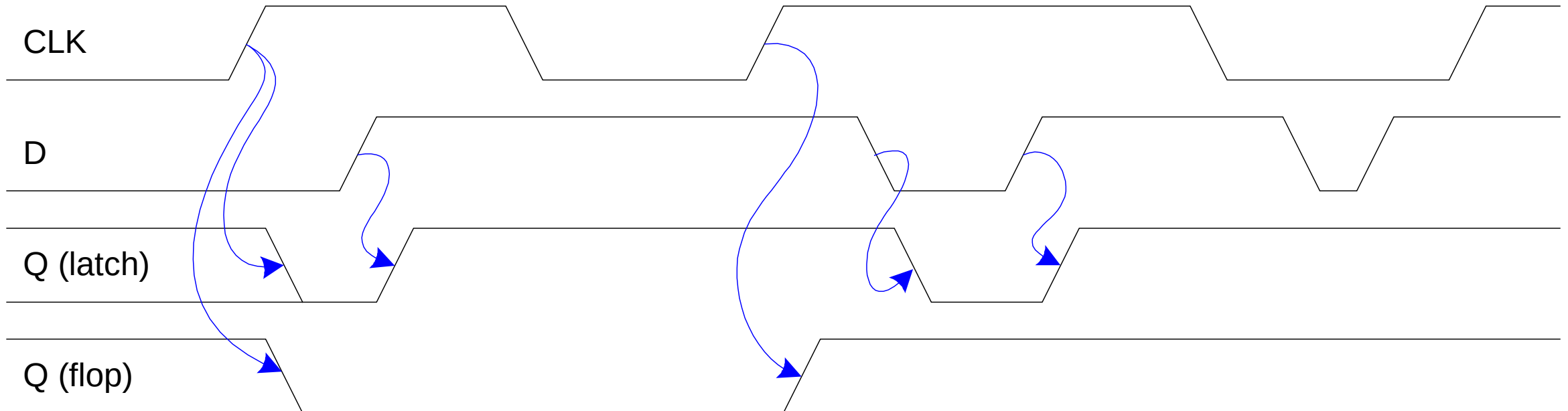
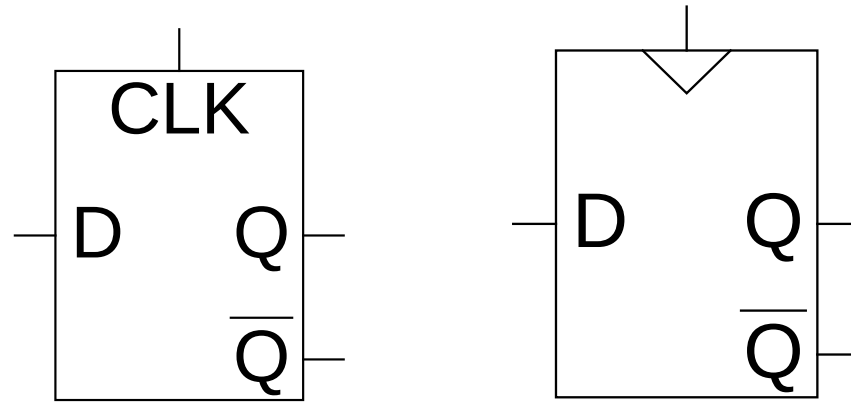


Внутрішня структура D-тригера затримки (delay)

- Два послідовно з'єднані фіксатори (L1 и L2), які керуються комплементарними тактовими сигналами
- Коли **$CLK = 0$**
 - L1 прозорий
 - L2 непрозорий
 - D проходить до N1
- Коли **$CLK = 1$**
 - L1 непрозорий
 - L2 прозорий
 - N1 проходить до Q
- Отже, за фронтом тактового сигналу (коли **CLK змінюється від 0 до 1**)
 - D проходить до Q



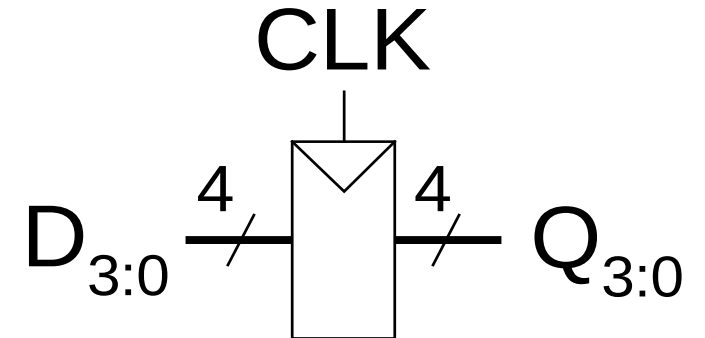
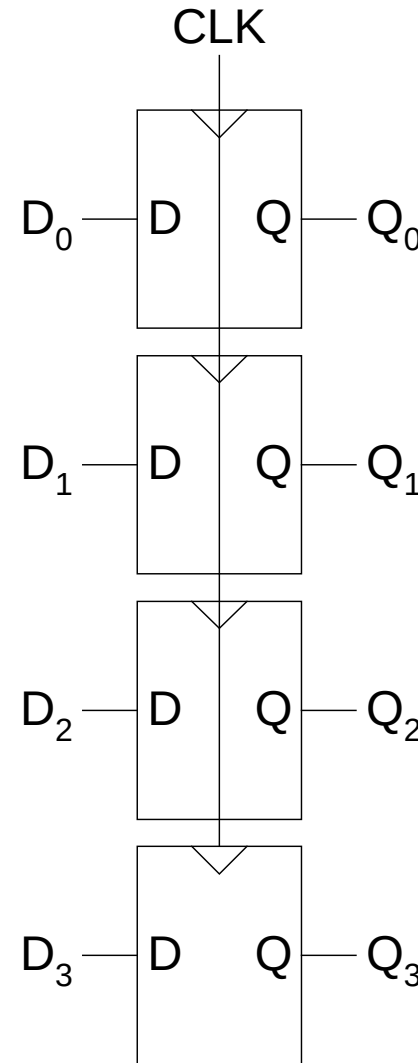
Порівняння D-фіксатора (latch) і D-тригера затримки (flop)



Регістр

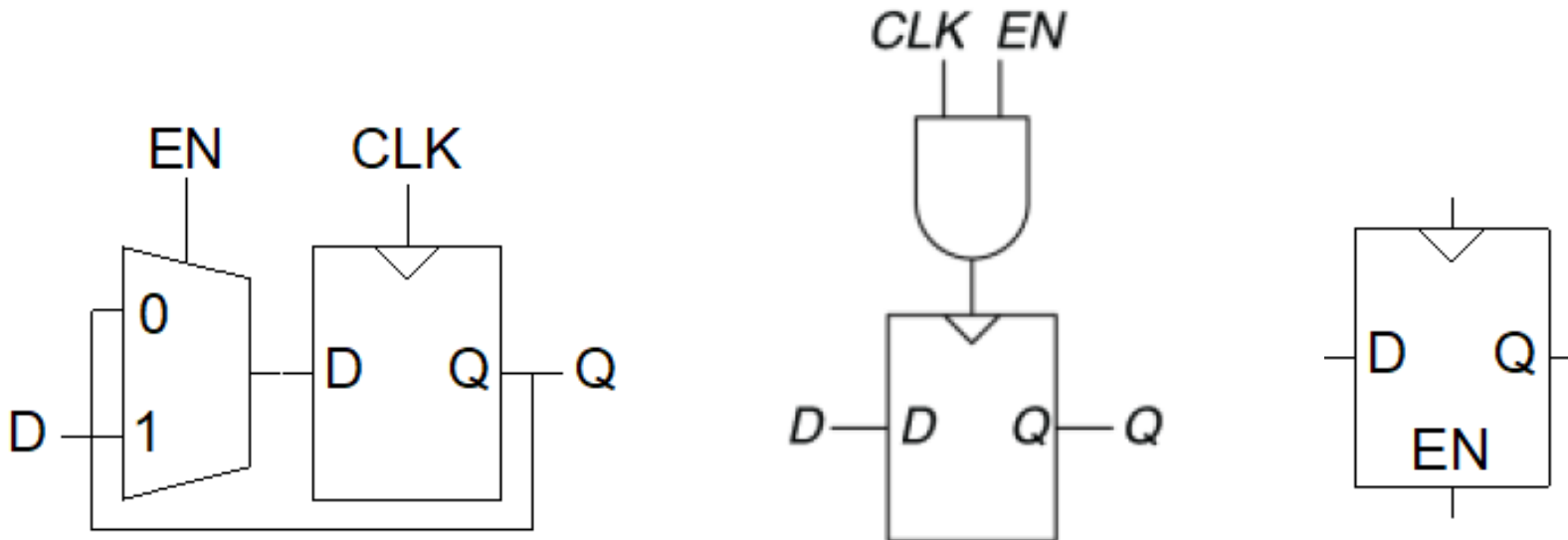
Регістр:

- N-розрядний регістр – набір з N D-тригерів із загальним тактовим сигналом
- Всі біти регістра обновлюються одночасно
- Регістр є базовим блоком послідовнісних схем



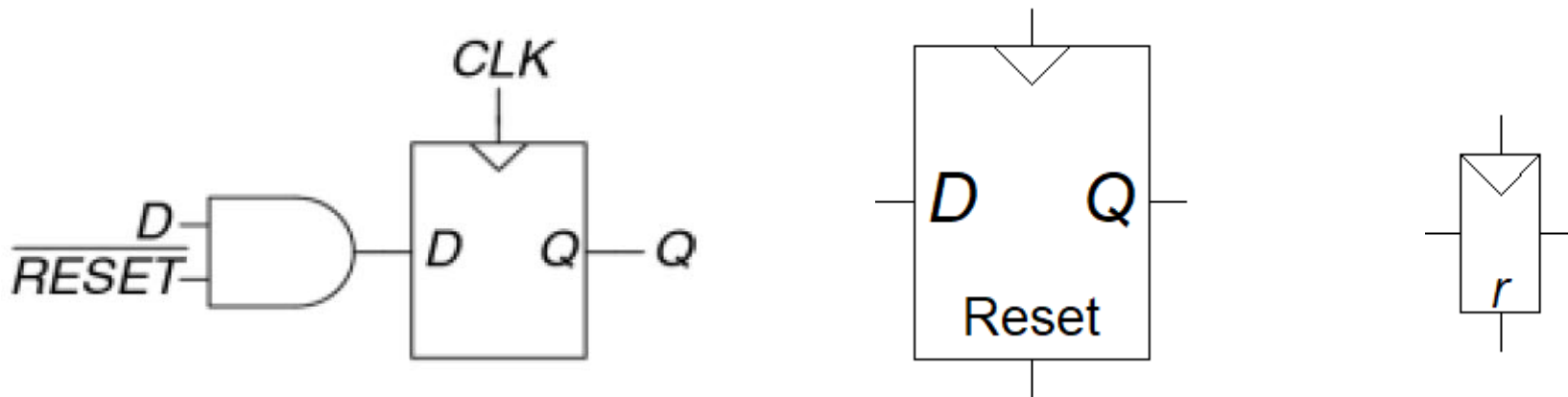
D-тригер із функцією дозволу

- **Входи:** CLK , D , EN
 - Вхід дозволу (EN) контролює, чи будуть зберігатися нові дані (D)
- **Робота**
 - $EN = 1$: D проходить на Q за фронтом тактового сигналу
 - $EN = 0$: тригер зберігає попередній стан



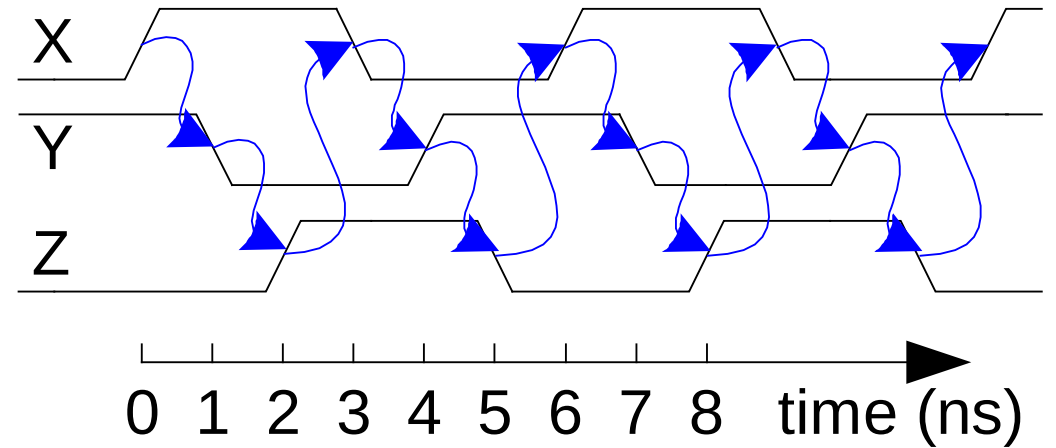
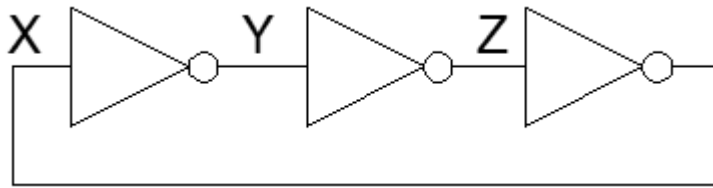
D-тригер із функцією скидування

- Входи: CLK , D , $Reset$
- Робота
 - **$Reset = 1$** : Q встановлюється в 0
 - **$Reset = 0$** : Тригер веде себе як звичайний D тригер
 - Синхронний тригер скидується синхронно по фронту сигналу CLK
 - Асинхронний тригер скидується зразу при поступанні 1 на вхід RESET не залежно від тактового сигналу



Послідовнісні цифрові схеми

- Послідовнісні схеми - всі схеми, які не є комбінаційними, тобто значення виходу залежить не тільки від поточного стану входу
- Проблемна схема:



- Вхідів не має, виходів 3
- Нестабільна схема, осцилятор
- Період залежить від величин затримок інверторів
- Схема має циклічний шлях, вихід поступає на вхід

Проектування синхронних логічних схем

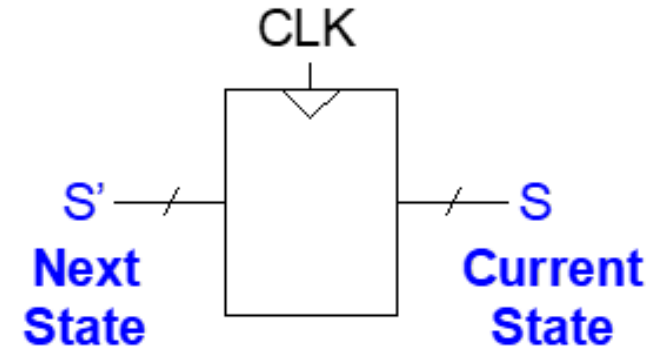
- Разривання циклічних шляхів **добавленням регістрів**
- Регістри збегігають **стан схеми**
- Стан змінюється за фронтами тактового сигналу, система **синхронізується** цим сигналом
- Правила побудови синхронних послідовнісних схем:
 - Кожний елемент схеми є або регістром, або комбінаційною схемою
 - Як мінімум один елемент схеми є регістром
 - Всі регістри тактуються єдиним тактовим сигналом
 - В кожному циклічному шляху має бути як мінімум один регістр
- Два основні типи синхронних послідовнісних схем
 - Скінченні автомати (FSM)
 - Конвейери

Скінченні автомати

- Складаються з:

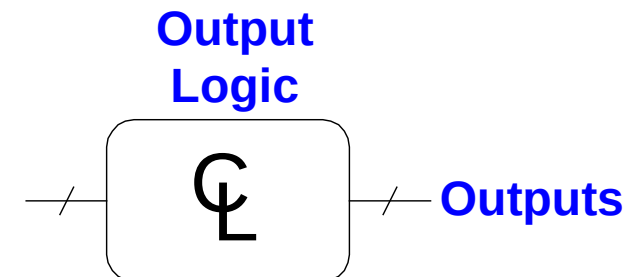
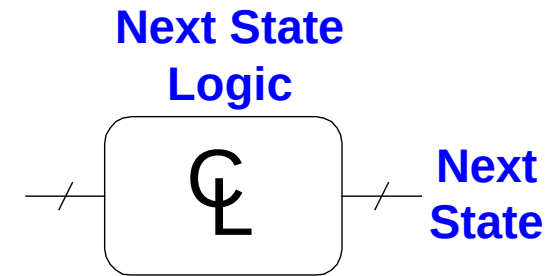
- Регістра стану**

- Зберігає поточне значення
- За фронтом тактового імпульсу завантажує наступний стан



- Комбінаційної логічної схеми**

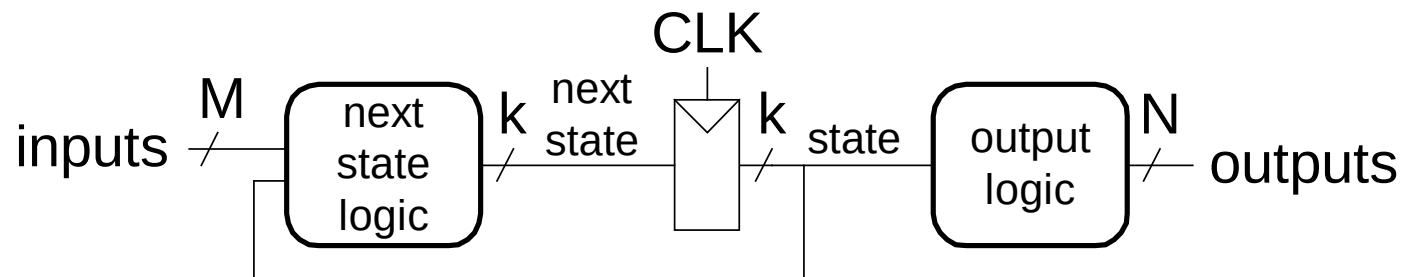
- Визначає наступний стан
- Визначає вихідні сигнали



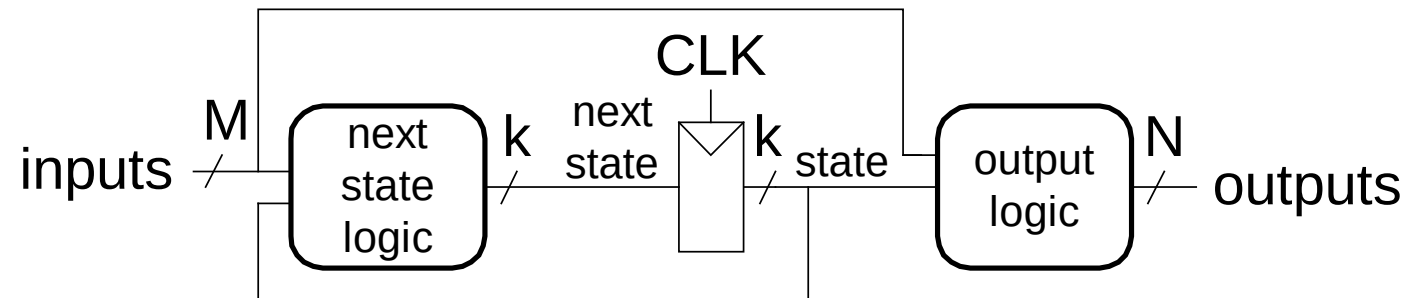
Класи скінченних автоматів

- Два класи скінченних автоматів відрізняються логікою визначення вихідних сигналів:
 - **Скінченний автомат Мура:** Вихід визначається тільки поточним станом
 - **Скінченний автомат Мілі:** Вихід визначається поточним станом і входами

Moore FSM

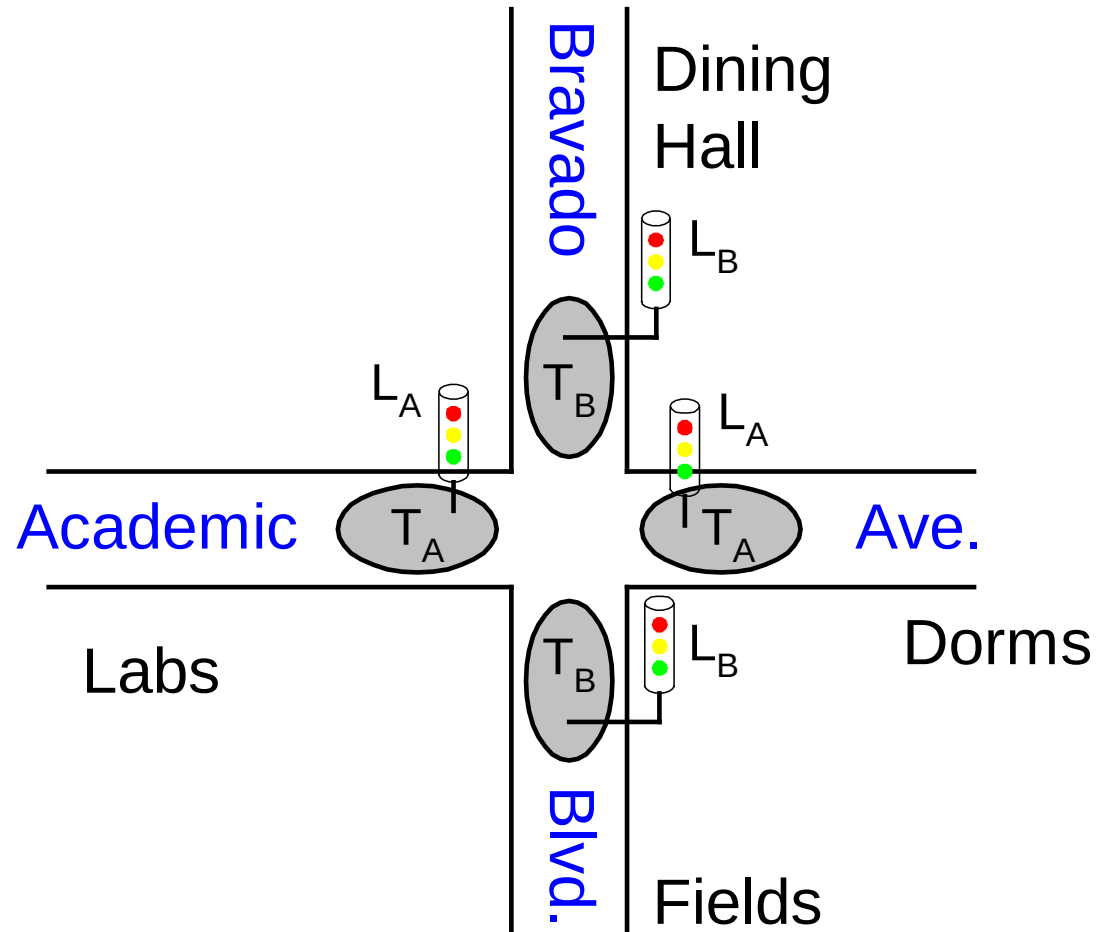


Mealy FSM



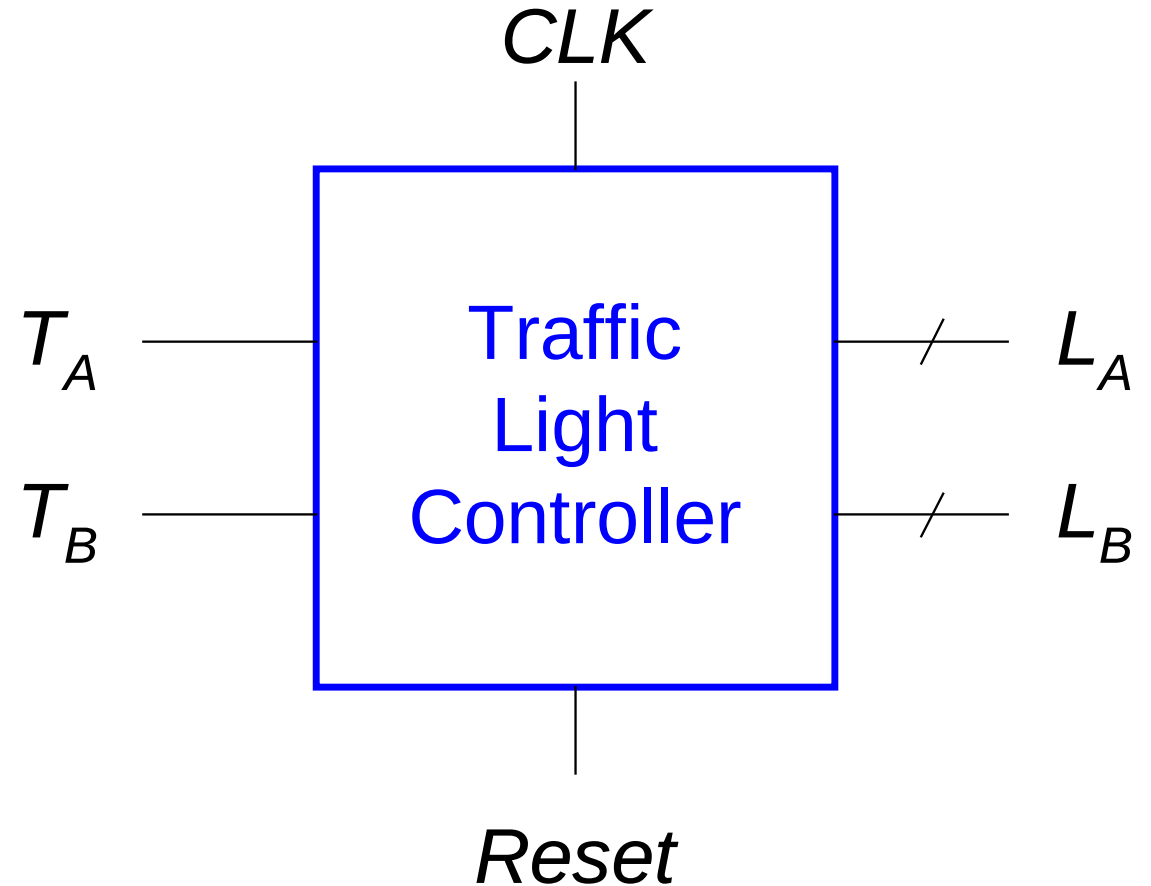
Приклад скінченного автомата

- Система керування світлофором:
 - Давачі руху: T_A , T_B (ІСТИНА, коли на вулиці є студенти)
 - Світлофори: L_A , L_B



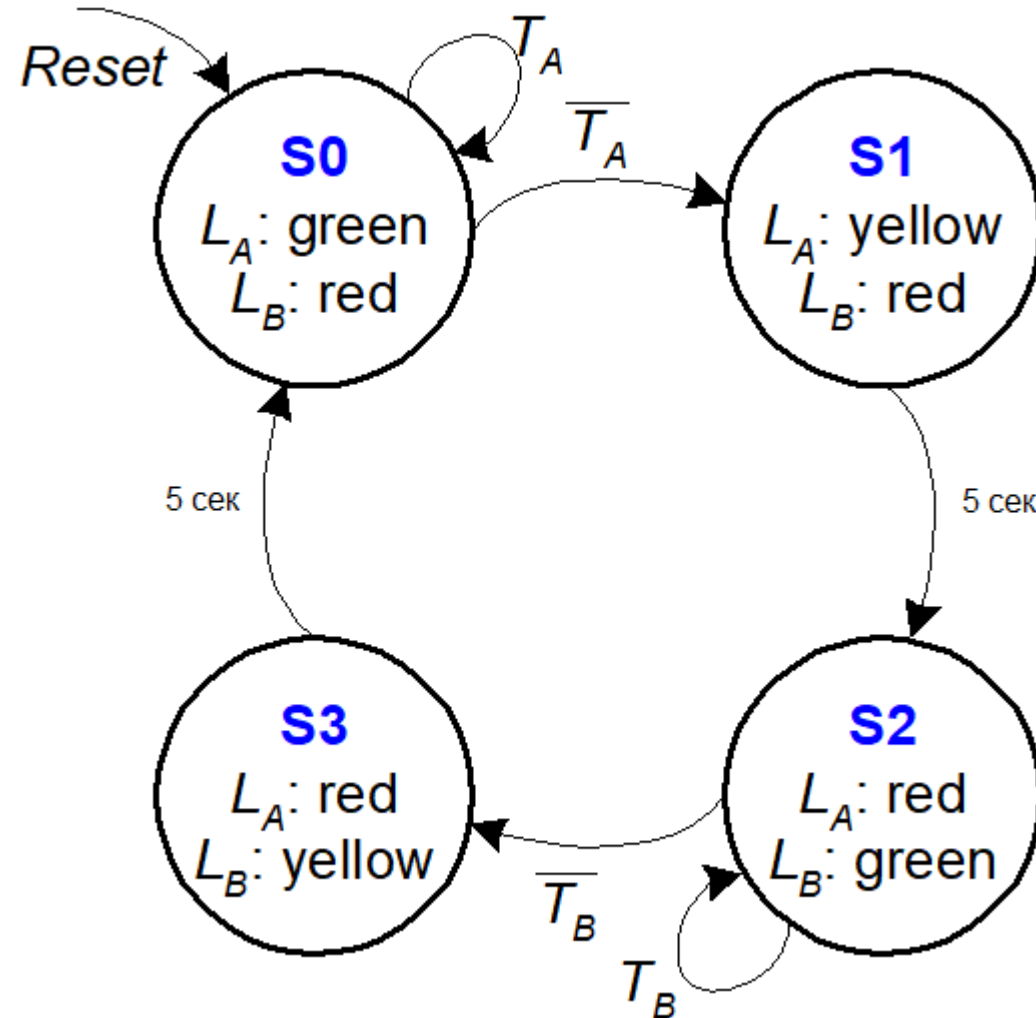
Скінченний автомат, як чорний ящик

- Входи: CLK , $Reset$, T_A , T_B
- Виходи: L_A , L_B



Діаграма переходів скінченного автомата

- **Скінченний автомат Мура:** Значення виходів вказані для кожного стану
- **Стан:** Кругжки
- **Переходи:** Дуги



Таблиця переходів скінченного автомата

Поточний стан		Входи		Наступний стан	
S_1	S_0	T_A	T_B	S'_1	S'_0
0	0	0	X	0	1
0	0	1	X	0	0
0	1	X	X	1	0
1	0	X	0	1	1
1	0	X	1	1	0
1	1	X	X	0	0

Стан	Кодування
S0	00
S1	01
S2	10
S3	11

$$S'_1 = \overline{S_1}S_0 + S_1\overline{S_0}T_B + S_1\overline{S_0}T_B = S_1 \oplus S_0$$

$$S'_0 = \overline{S_1}\overline{S_0}T_A + S_1\overline{S_0}T_B$$

Таблиця виходів скінченного автомата

Поточний стан		Виходи			
S_1	S_0	L_{A1}	L_{A0}	L_{B1}	L_{B0}
0	0	0	0	1	0
0	1	0	1	1	0
1	0	1	0	0	0
1	1	1	0	0	1

Виходи	Кодування
зелений	00
жовтий	01
червоний	10

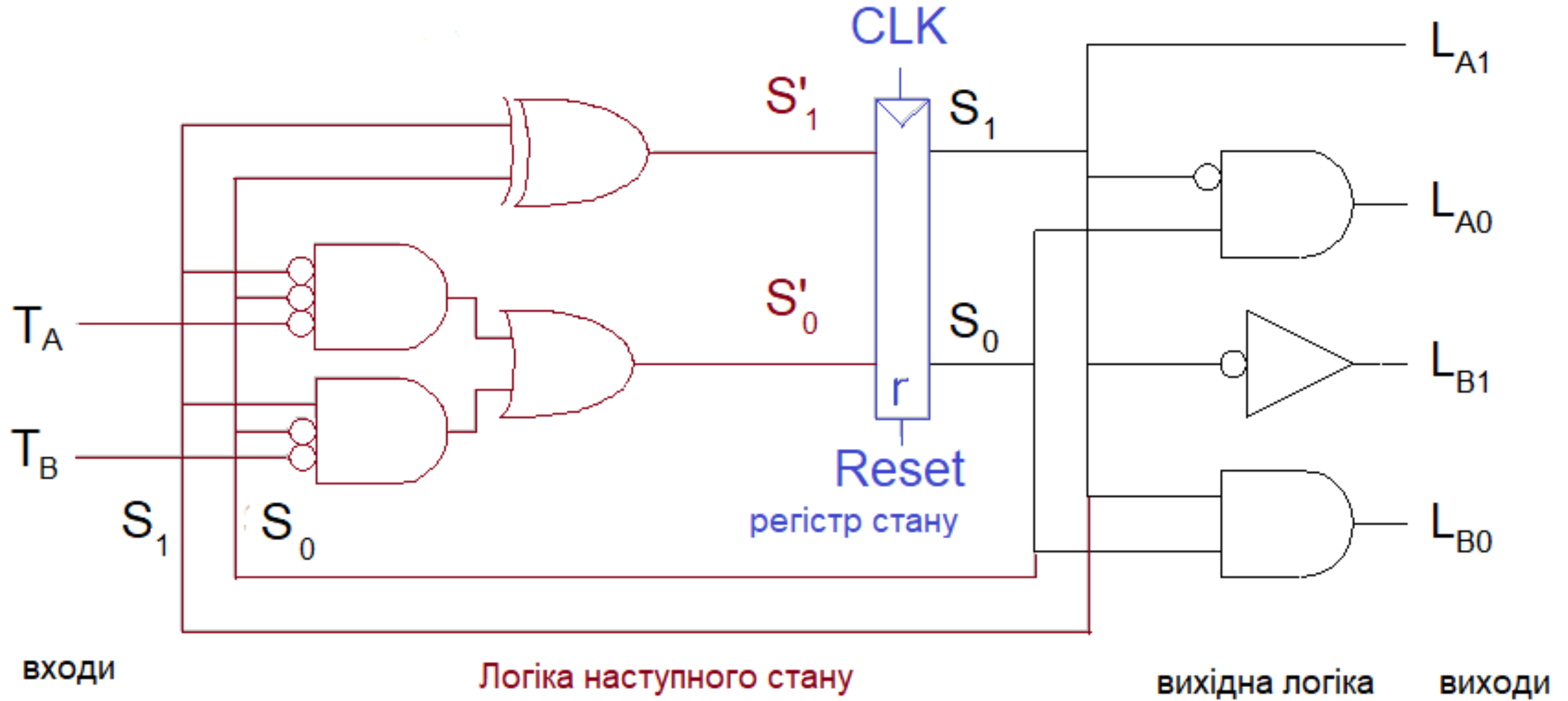
$$L_{A1} = S_1 \overline{S_0} + S_1 S_0 = S_1$$

$$L_{A0} = \overline{S_1} S_0$$

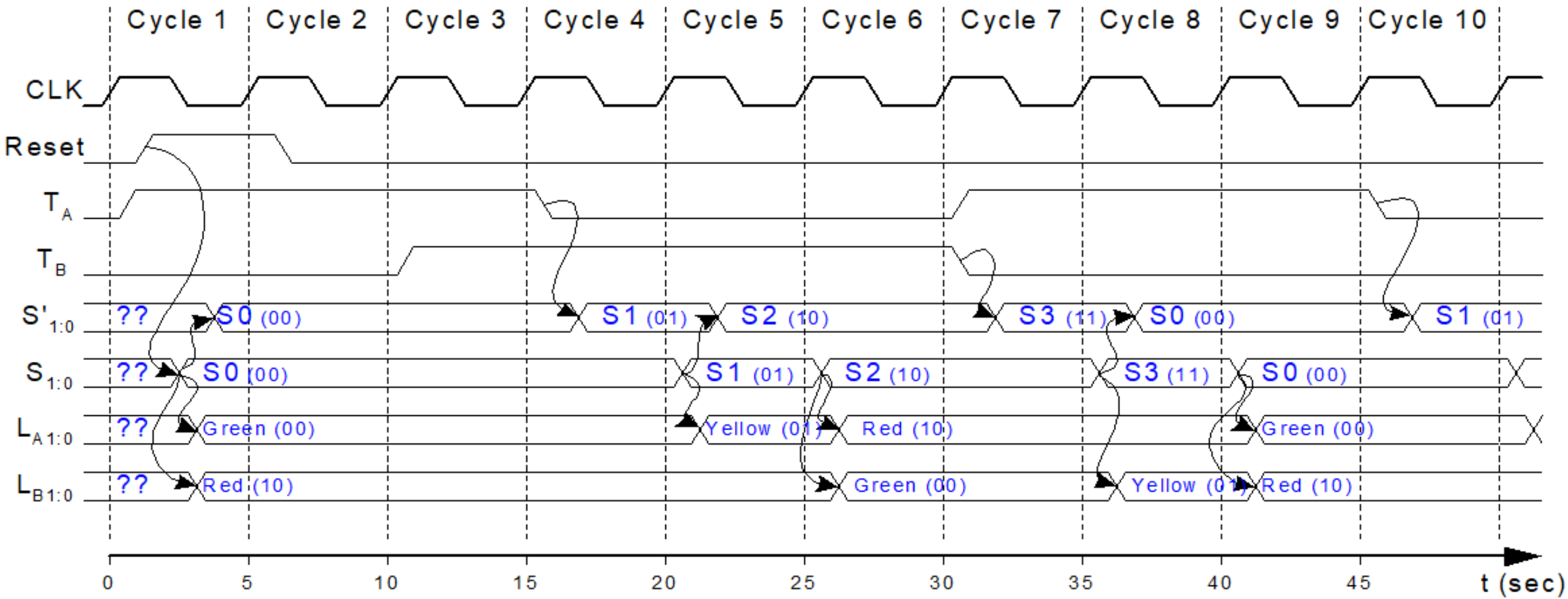
$$L_{B1} = \overline{S_1} \overline{S_0} + \overline{S_1} S_0 = \overline{S_1}$$

$$L_{B0} = S_1 S_0$$

Схема скінченного автомату світлофора



Часова діаграма скінченного автомата



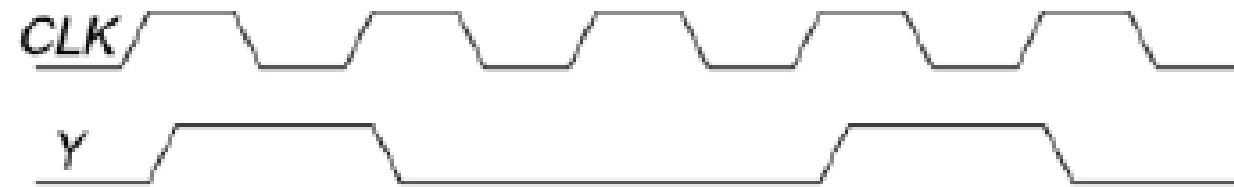
Кодування станів скінченного автомата

Важливим є вибір системи кодування станів автомата:

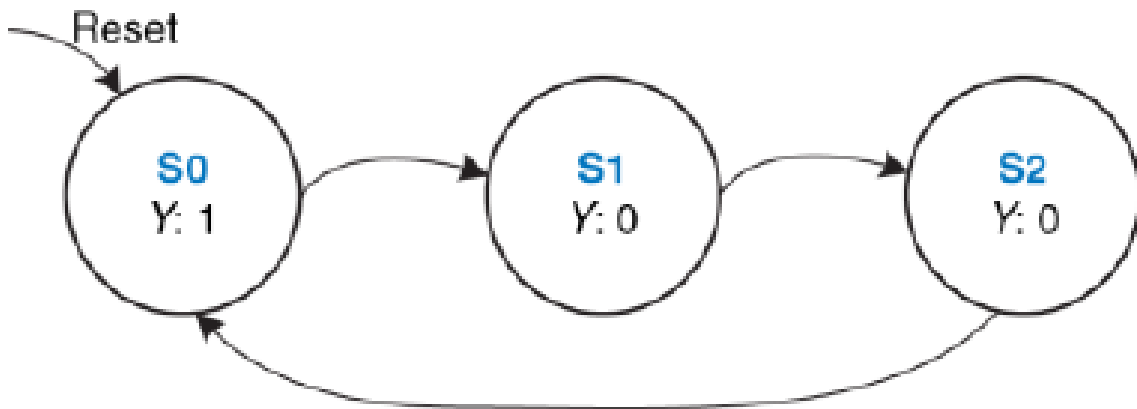
- **Двійкове кодування:**
 - Для чотирьох станів: 00, 01, 10, 11
- **Пряме унітарне (unitary code, One-hot) кодування**
 - На кожний стан виділяється один біт
 - Тільки один біт має значення ІСТИНА
 - Для чотирьох станів: 0001, 0010, 0100, 1000
 - Використовується більше тригерів
 - Часто логіка наступного стану і вихідна логіка простіша

Проектування лічильника-дільника на 3

Часова діаграма



Діаграма переходів Мура



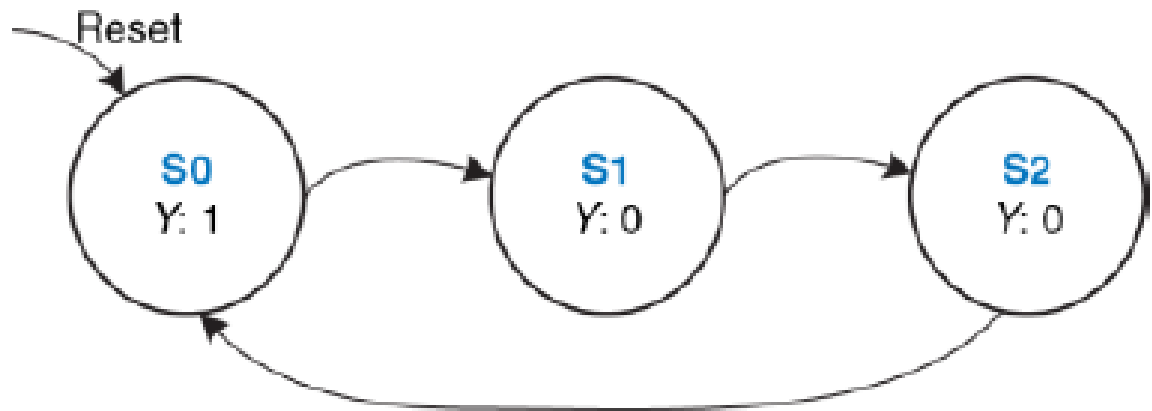
Таблиця переходів $S'_0=S_2$ $S'_1=S_0$ $S'_2=S_1$

Current State	Next State
S0	S1
S1	S2
S2	S0

Таблиця виходів $Y=S_0$

Current State	Output
S0	1
S1	0
S2	0

Пряме унітарне і двійкове кодування станів і переходів



Таблиця переходів з двійковим кодуванням

Current State		Next State	
S_1	S_0	S'_1	S'_0
0	0	0	1
0	1	1	0
1	0	0	0

Пряме унітарне і двійкове кодування станів

State	One-Hot Encoding			Binary Encoding	
	S_2	S_1	S_0	S'_1	S'_0
S0	0	0	1	0	0
S1	0	1	0	0	1
S2	1	0	0	1	0

Таблиця переходів з прямим унітарним

Current State			Next State		
S_2	S_1	S_0	S'_2	S'_1	S'_0
0	0	1	0	1	0
0	1	0	1	0	0
1	0	0	0	0	1

Схеми лічильника для різних кодувань

Схема лічильника з
двійковим кодуванням

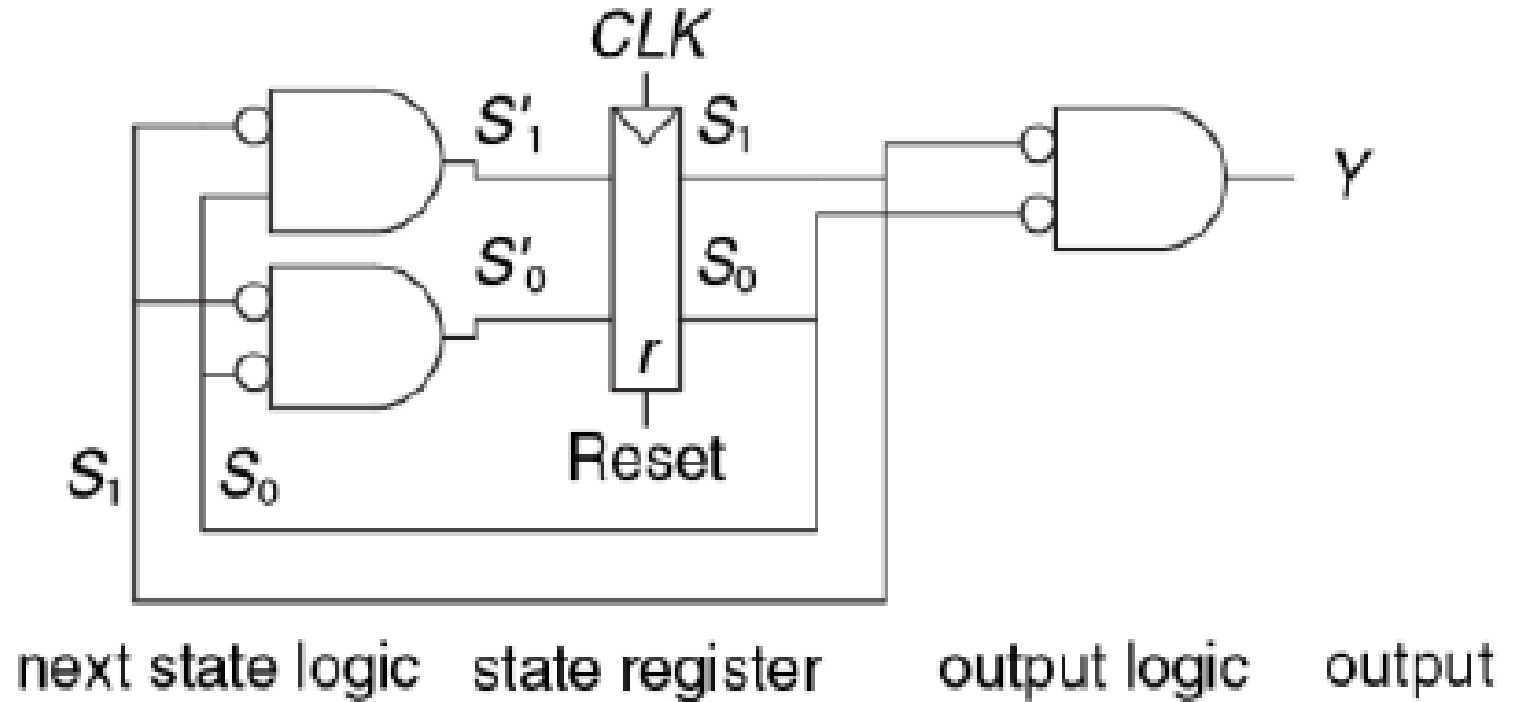
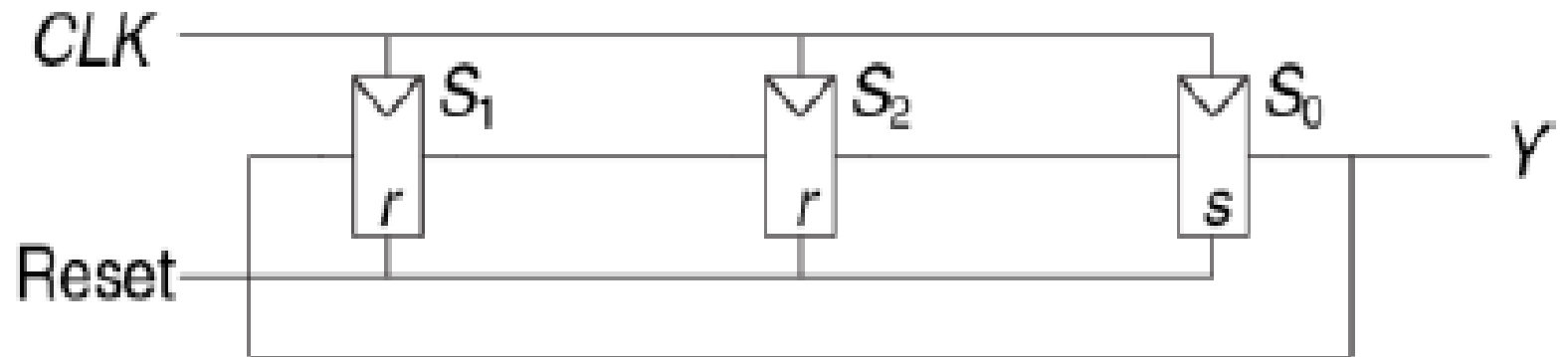


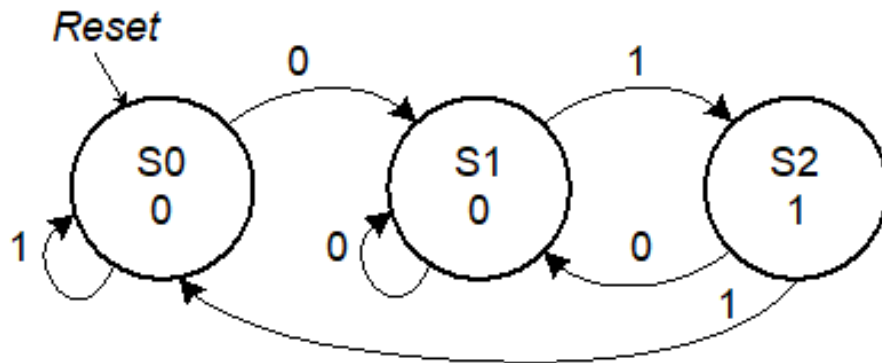
Схема лічильника з
унітарним кодуванням



Порівняння скінченних автоматів

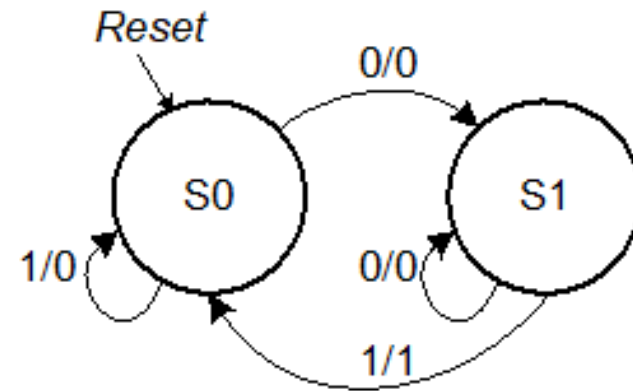
Автомати Мура і Мілі, які із потоку бітів виділяють послідовність 01

Автомат Мура:



Значення входів вказується біля дуг
Значення виходів вказуються всередині
кружків

Автомат Мілі



Значення входів/виходів задаються біля дуг

Таблиця переходів і виходів автомата Мура

Таблиця переходів автомата Мура

Current State S	Input A	Next State S'
S0	0	S1
S0	1	S0
S1	0	S1
S1	1	S2
S2	0	S1
S2	1	S0

Таблиця виходів автомата Мура

Current State S	Output Y
S0	0
S1	0
S2	1

Таблиця переходів автомата Мура з кодуванням станів

Current State S_1 S_0		Input A	Next State S'_1 S'_0	
0	0	0	0	1
0	0	1	0	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0

Таблиця виходів автомата Мура з кодуванням станів

Current State S_1 S_0		Output Y
0	0	0
0	1	0
1	0	1

Таблиця переходів і виходів автомата Мілі

Таблиця переходів і виходів автомата Мілі

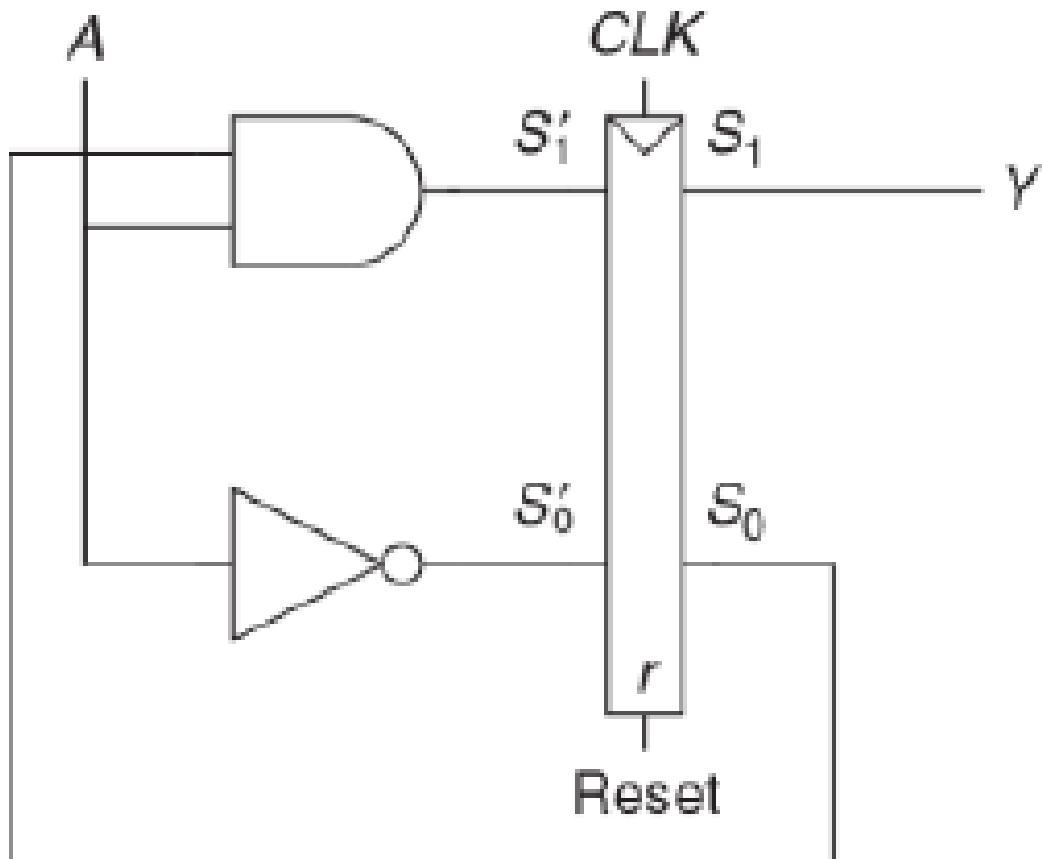
Current State S	Input A	Next State S'	Output Y
S_0	0	S_1	0
S_0	1	S_0	0
S_1	0	S_1	0
S_1	1	S_0	1

Таблиця переходів і виходів автомата Мілі з кодуванням станів

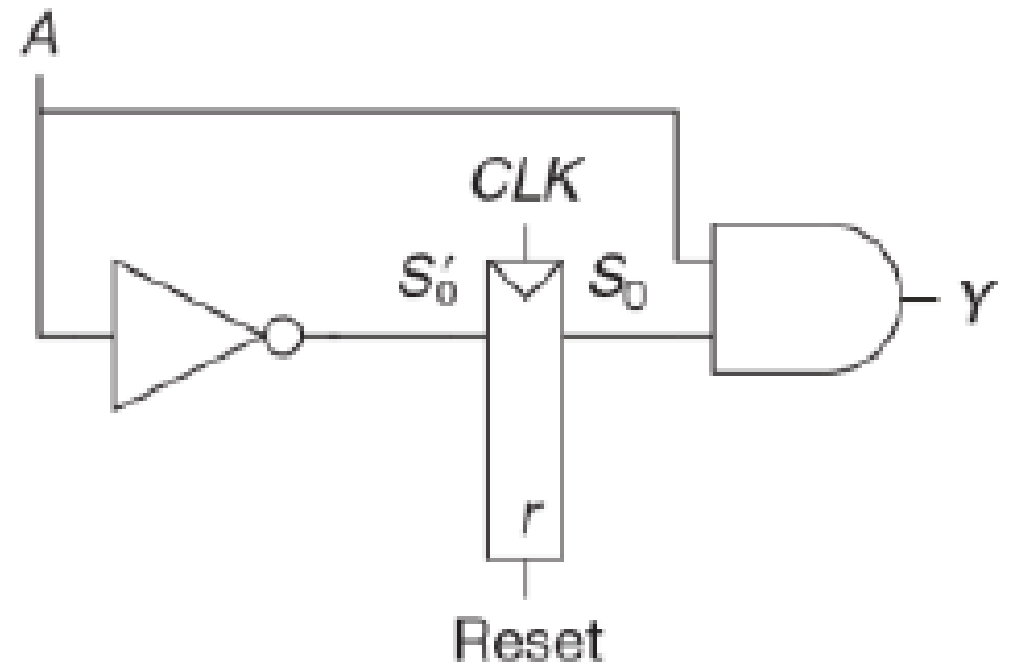
Current State S_0	Input A	Next State S'_0	Output Y
0	0	1	0
0	1	0	0
1	0	1	0
1	1	0	1

Схеми автоматів

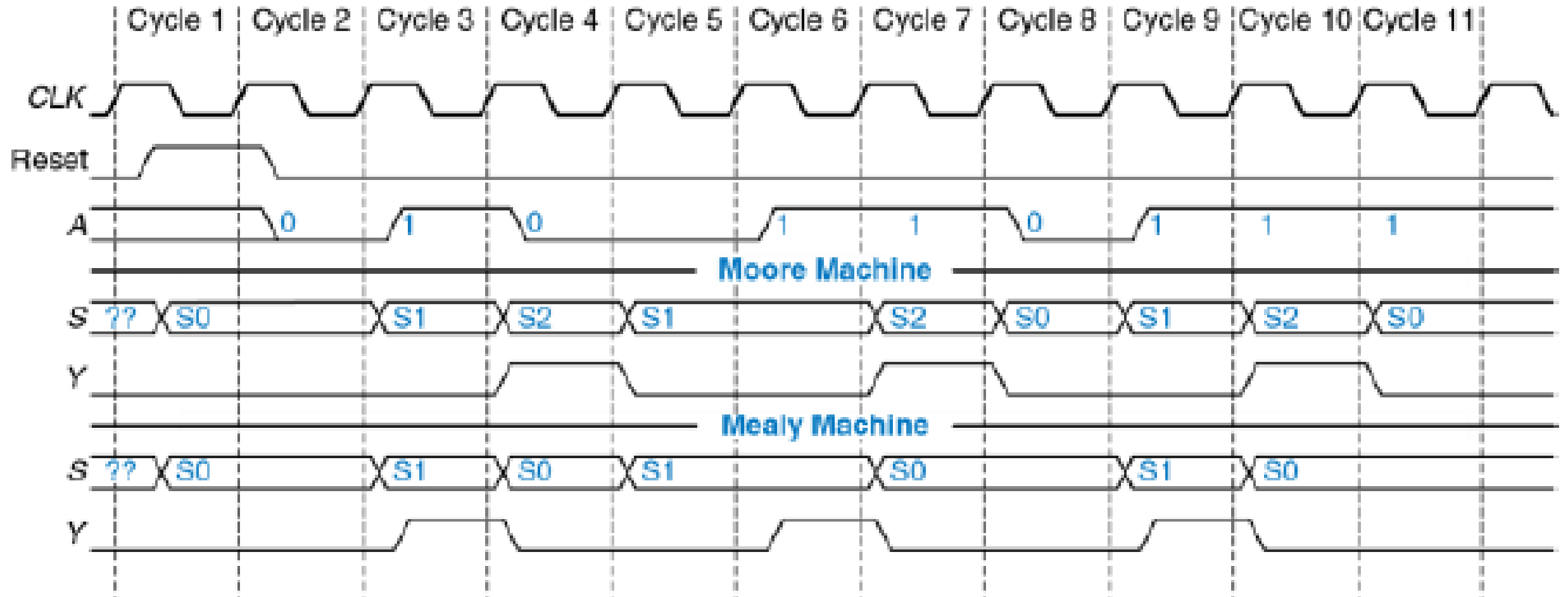
Мура



Мілі



Часова діаграма скінченних автоматів Мура і Мілі

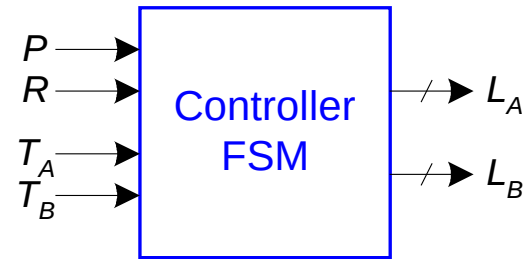


Декомпозиція скінчених автоматів

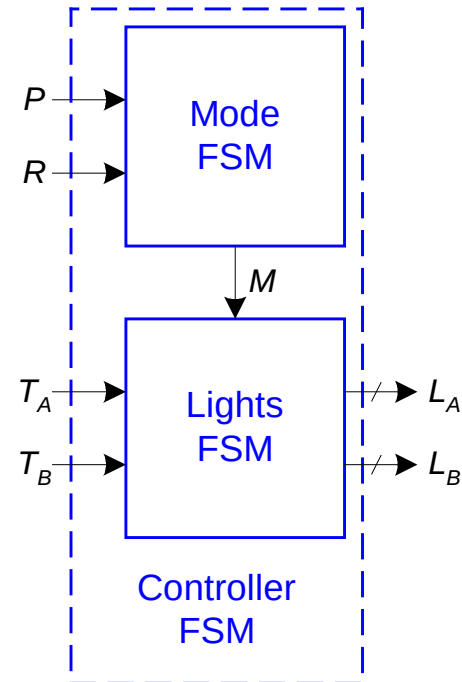
- Розділення складного скінченного автомата на більш прості взаємодіючі скінченні автомати
- Приклад: Модифікувати контролер світлофора так, щоб в ньому з'явився режим «параду»
 - Два додаткових входи P , R
 - Коли $P = 1$, автомат входить в режим парадну і світлофор на парадній вулиці залишається зеленим
 - Коли $R = 1$, автомат виходить з режиму парадну

Модифікований скінченний автомат

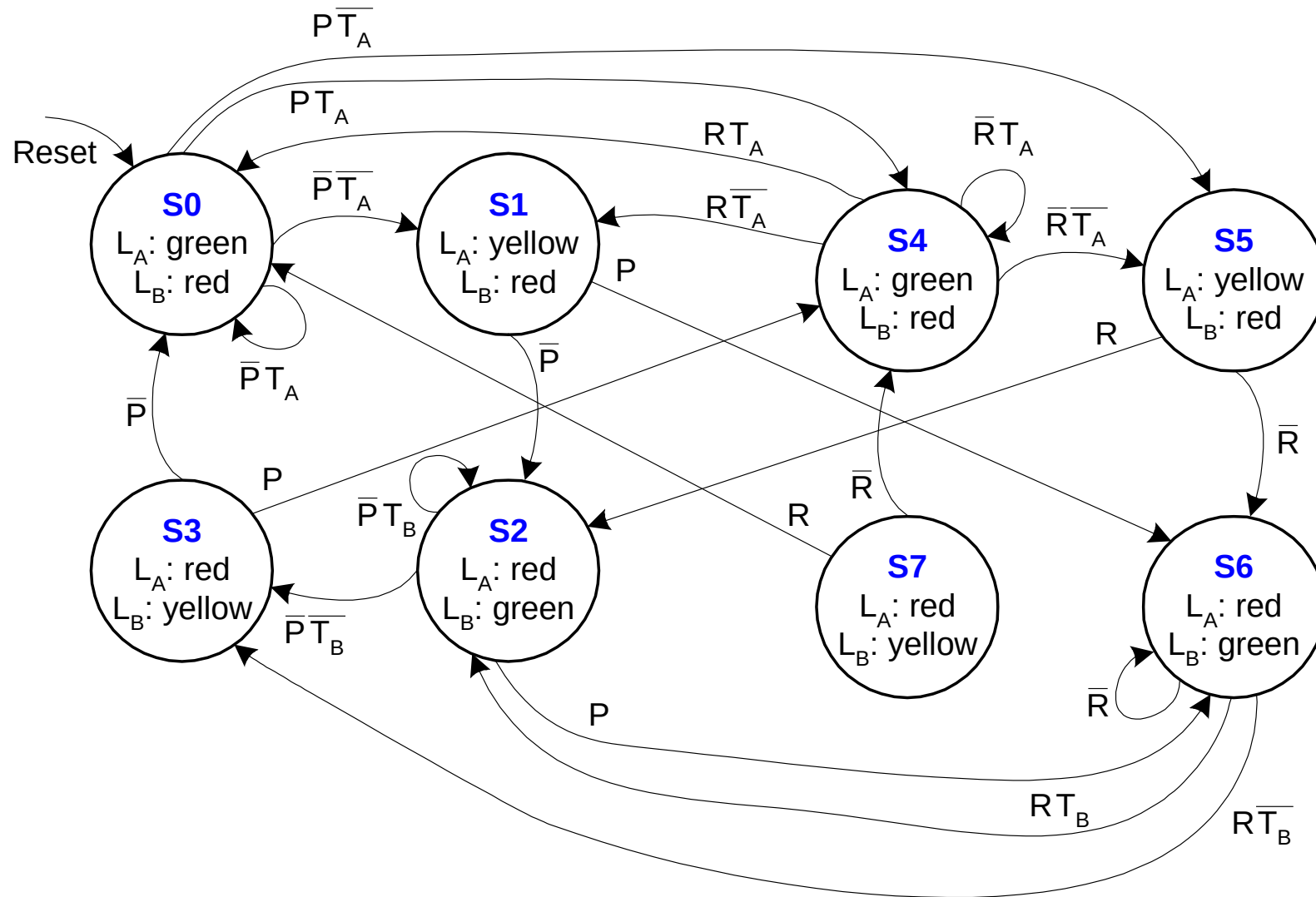
Немодульний
скінченний автомат



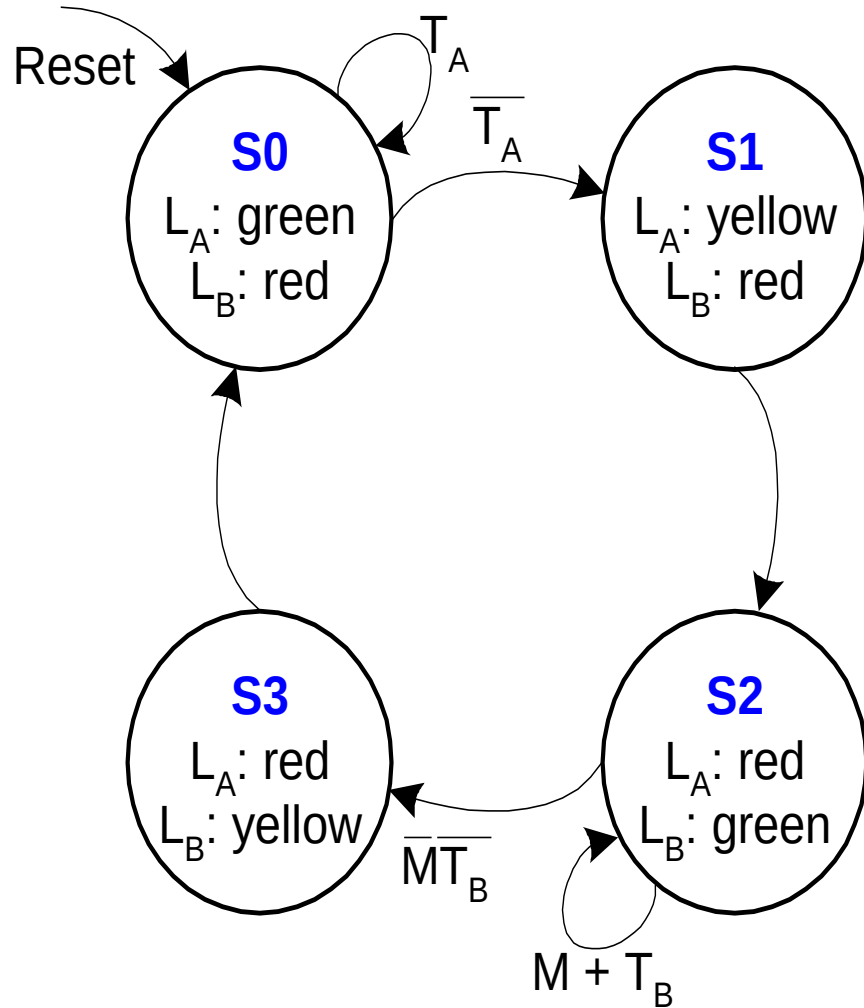
Модульний
скінченний автомат



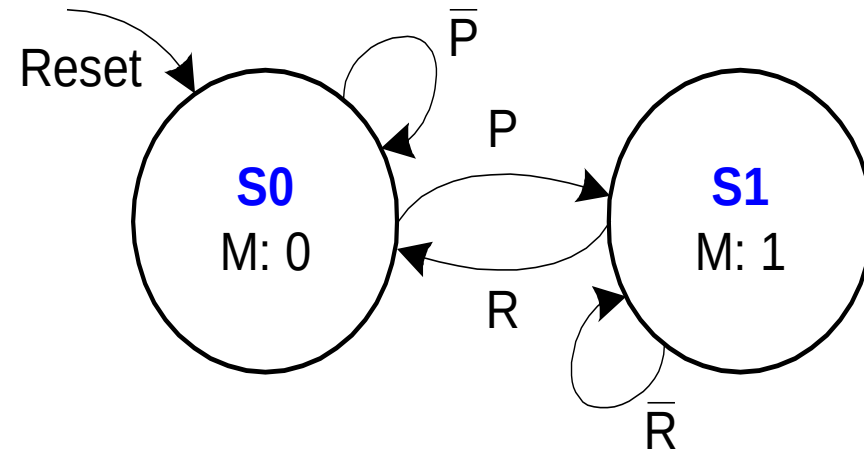
Діаграма переходів немодульного скінченного автомату



Діаграма переходів модульного скінченного автомату



Lights FSM



Mode FSM

Проектування скінченного автомата

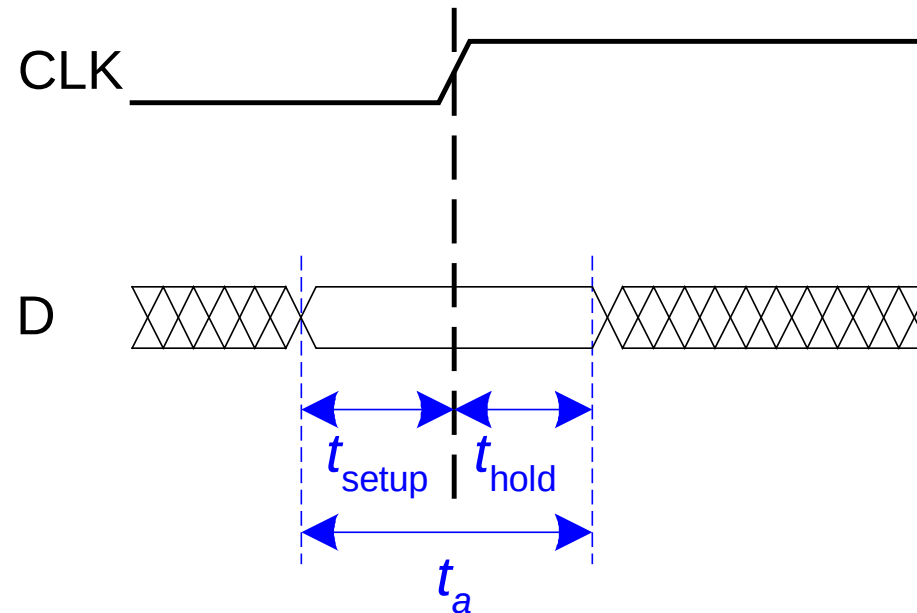
- Визначити входи і виходи
- Намалювати діаграму переходів
- Записати таблицю переходів
- Вибрати спосіб кодування станів
- Для скінченного автомата Мура:
 - Переписати таблицю переходів з врахуванням кодування станів
 - Записати таблицю виходів
- Для скінченного автомата Мілі:
 - Переписати об'єднану таблицю переходів і виходів з врахуванням кодування станів
- Записати булеві вирази логіки наступного стану і виходів
- Намалювати принципову схему

Синхронізація

- Тригер копіює сигнал з D-входу на Q-вихід за переднім фронтом тактового сигналу. Цей процес називається фіксацією (sampling) D-сигналу за фронтом тактового сигналу
- Сигнал D має бути стабільним в процесі фіксації
- Сигнал D має бути стабільним в околі фронту тактового сигналу
- Якщо це не так, може виникнути метастабільність

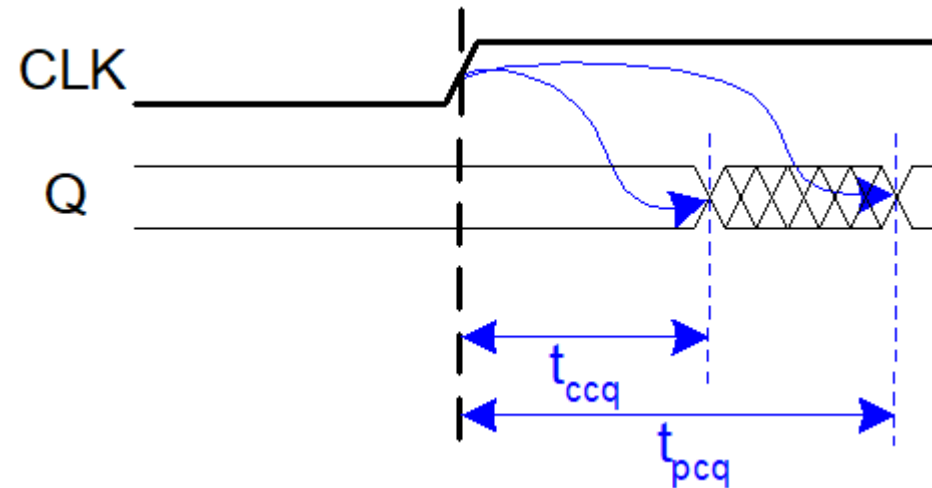
Обмеження часу зміни входів

- **Час передвстановлення (setup time):** t_{setup} = період часу *перед* фронтом тактового сигналу, впродовж якого дані повинні бути стабільними (тобто не змінюватися)
- **Час утримування (hold time):** t_{hold} = період часу *після* фронту тактового сигналу, впродовж якого дані повинні бути стабільними
- **Апертурний час:** t_a = загальний час в околі фронту тактового сигналу, впродовж якого дані повинні бути стабільними ($t_a = t_{\text{setup}} + t_{\text{hold}}$)



Обмеження часу зміни сигналів

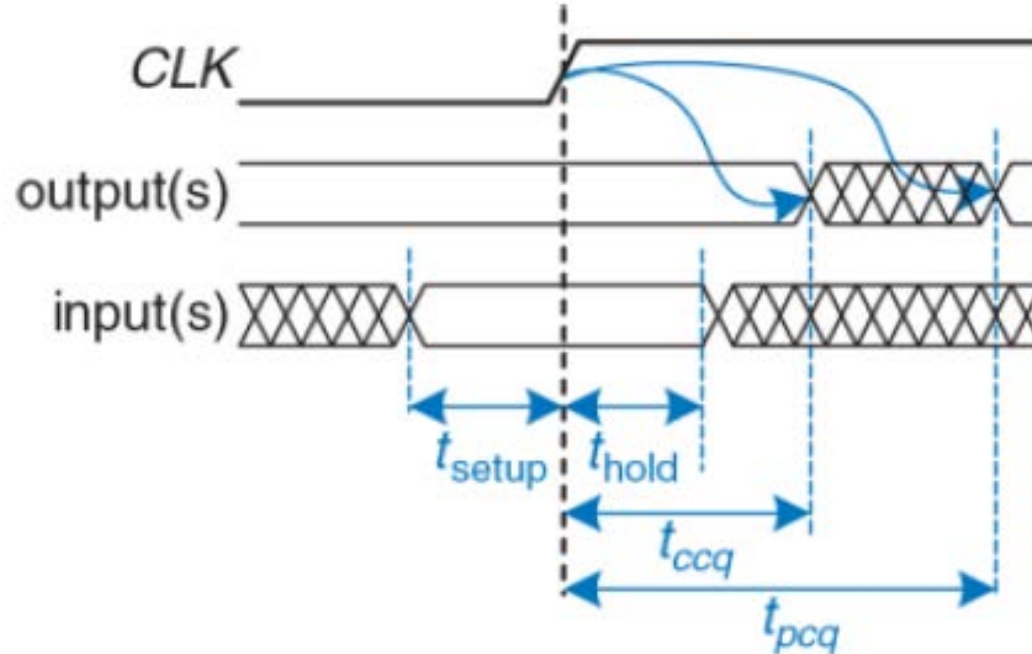
- **Затримка поширення** t_{pcq} = період часу після фронту тактового сигналу, після закінчення якого вихід Q буде гарантовано стабільним (тобто, перестане змінюватися)
- **Затримка реакції** t_{ccq} = період часу після фронту тактового сигналу, після закінчення якого Q може бути нестабільним (тобто, почати змінюватися)



Динамічна дисципліна

Динамічна дисципліна встановлює вимоги:

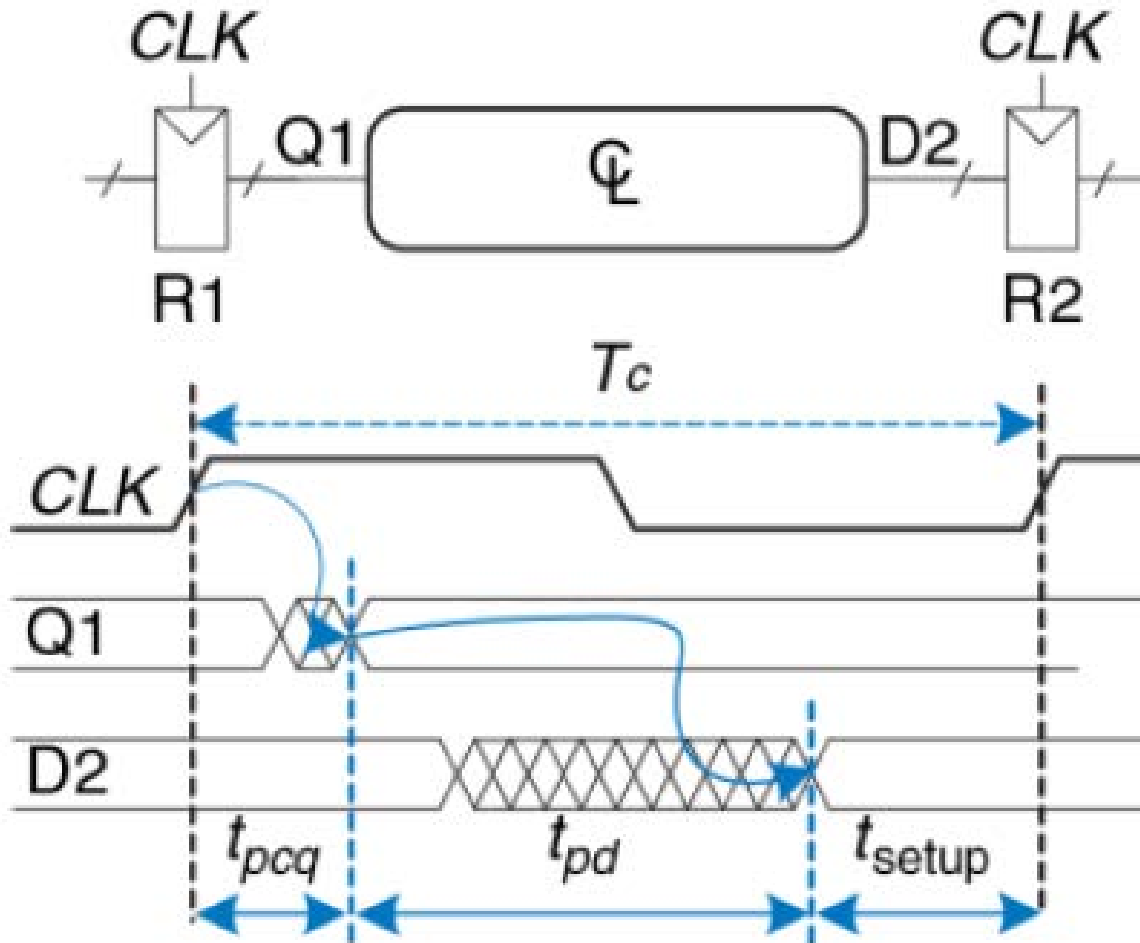
- Входи синхронної послідовної схеми повинні бути стабільними впродовж часу передвстановлення t_{setup} до фронту тактового сигналу і впродовж часу утримання t_{hold} після фронту тактового сигналу
- Так як тактові сигнали поступають з деяким розкидом фаз (clock skew), то період тактових сигналів має бути достатньо великим, щоб закінчилися перехідні процеси всіх вхідних і вихідних сигналів.



Динамічна дисципліна

- Тривалість періоду синхронізації T_c :

$$T_c \geq t_{pcq} + t_{pd} + t_{setup}$$



t_{pcq} - затримка поширення Q1

t_{pd} - затримка поширення
комбінаційної схеми CL

t_{setup} - час передвстановлення D2

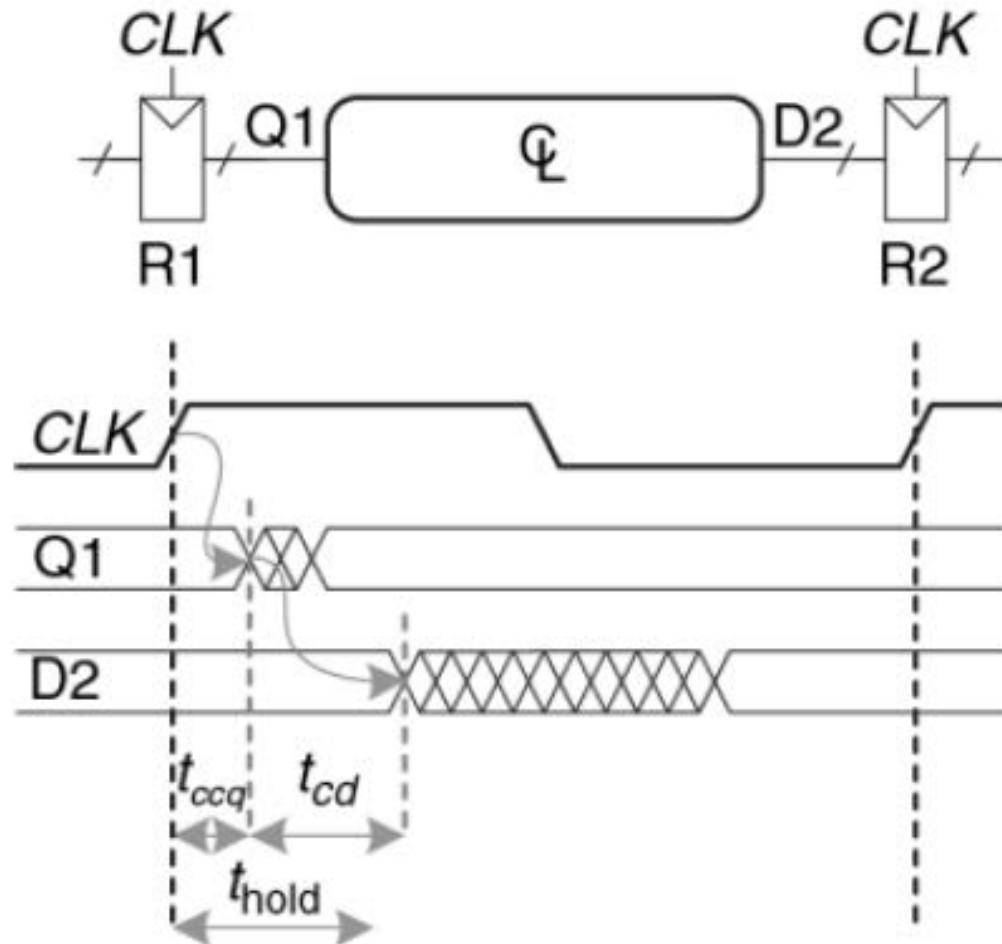
$$f_c = \frac{1}{T_c} - \text{тактова частота}$$

Проектант може змінити тільки t_{pd} , тому

$$t_{pd} \leq T_c - (t_{pcq} + t_{setup})$$

Обмеження на мінімальну затримку комбінаційної схеми

- **мінімальна** затримки комбінаційної логіки між регістрами R1 і R2
- вхід регістра R2 повинен бути стабільним впродовж часу t_{hold} після фронту тактового сигналу

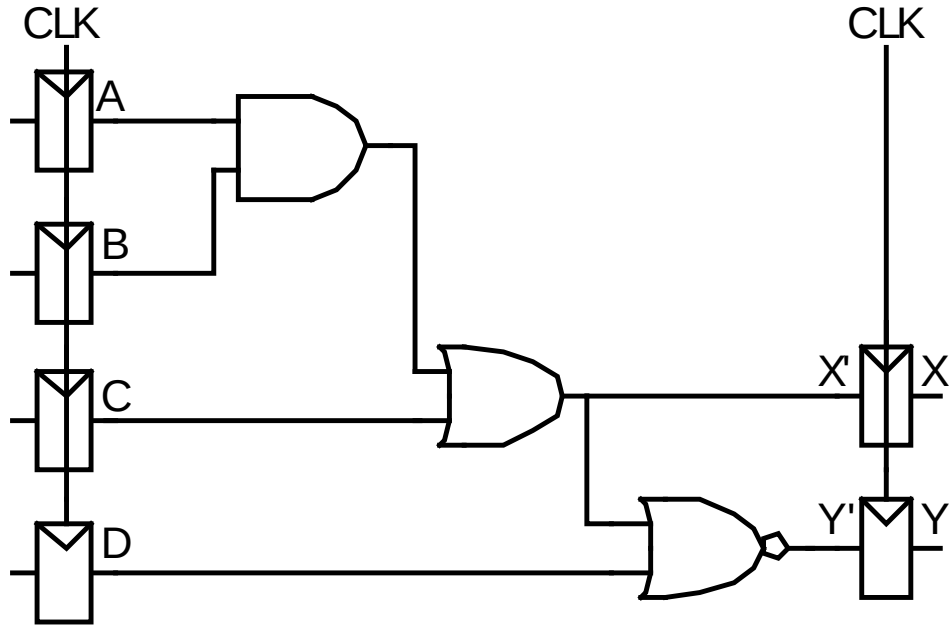


$$t_{hold} < t_{ccq} + t_{cd}$$

$$t_{cd} > t_{hold} - t_{ccq}$$

t_{cd} - мінімальна затримка
комбінаційної схеми

Часовий аналіз



$$t_{pd} = 3 \times 35 \text{ пс} = 105 \text{ пс}$$

$$t_{cd} = 25 \text{ пс}$$

Обмеження часу передвстановлення:

$$T_c \geq (50 + 105 + 60) \text{ пс} = 215 \text{ пс}$$

$$f_c = 1/T_c = 4.65 \text{ ГГц}$$

Часові характеристики

$$t_{ccq} = 30 \text{ пс}$$

$$t_{pcq} = 50 \text{ пс}$$

$$t_{\text{setup}} = 60 \text{ пс}$$

$$t_{\text{hold}} = 70 \text{ пс}$$

per gate

$$\left[\begin{array}{ll} t_{pd} & = 35 \text{ пс} \\ t_{cd} & = 25 \text{ пс} \end{array} \right.$$

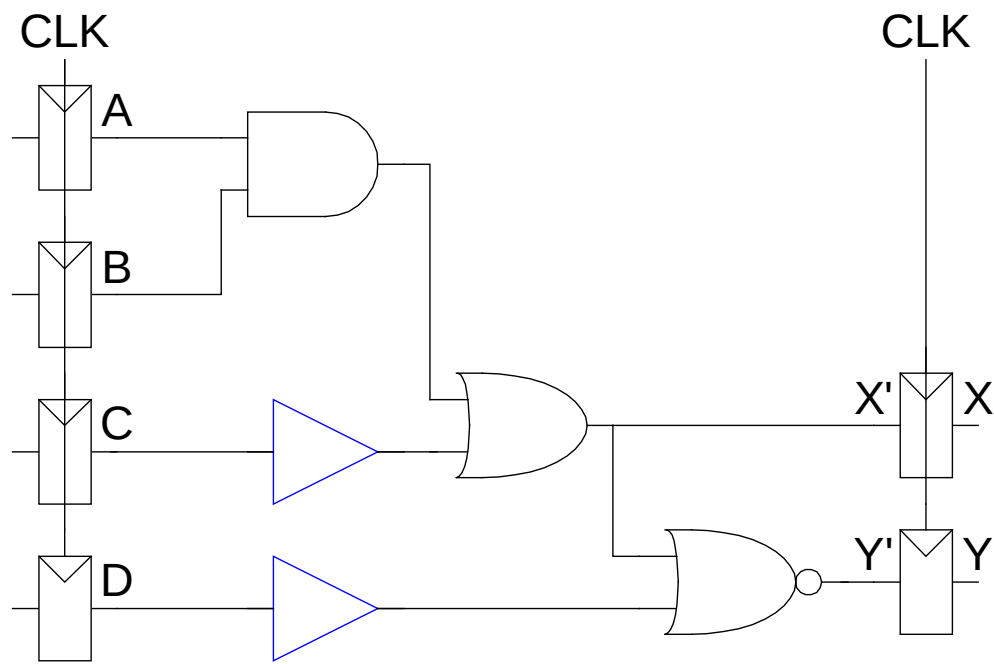
Обмеження часу утримування:

$$t_{ccq} + t_{cd} > t_{\text{hold}} ?$$

$$(30 + 25) \text{ пс} > 70 \text{ пс} ? \text{ Ні!}$$

Часовий аналіз

Додавання буфера
у найкоротший шлях



$$t_{pd} = 3 \times 35 \text{ пс} = 105 \text{ пс}$$

$$t_{cd} = 2 \times 25 \text{ пс} = 50 \text{ пс}$$

Обмеження часу передвстановлення:

$$T_c \geq (50 + 105 + 60) \text{ пс} = 215 \text{ пс}$$

$$f_c = 1/T_c = 4.65 \text{ ГГц}$$

**Часові
характеристики**

$$t_{ccq} = 30 \text{ пс}$$

$$t_{pcq} = 50 \text{ пс}$$

$$t_{\text{setup}} = 60 \text{ пс}$$

$$t_{\text{hold}} = 70 \text{ пс}$$

per gate

$$t_{pd} = 35 \text{ пс}$$

$$t_{cd} = 25 \text{ пс}$$

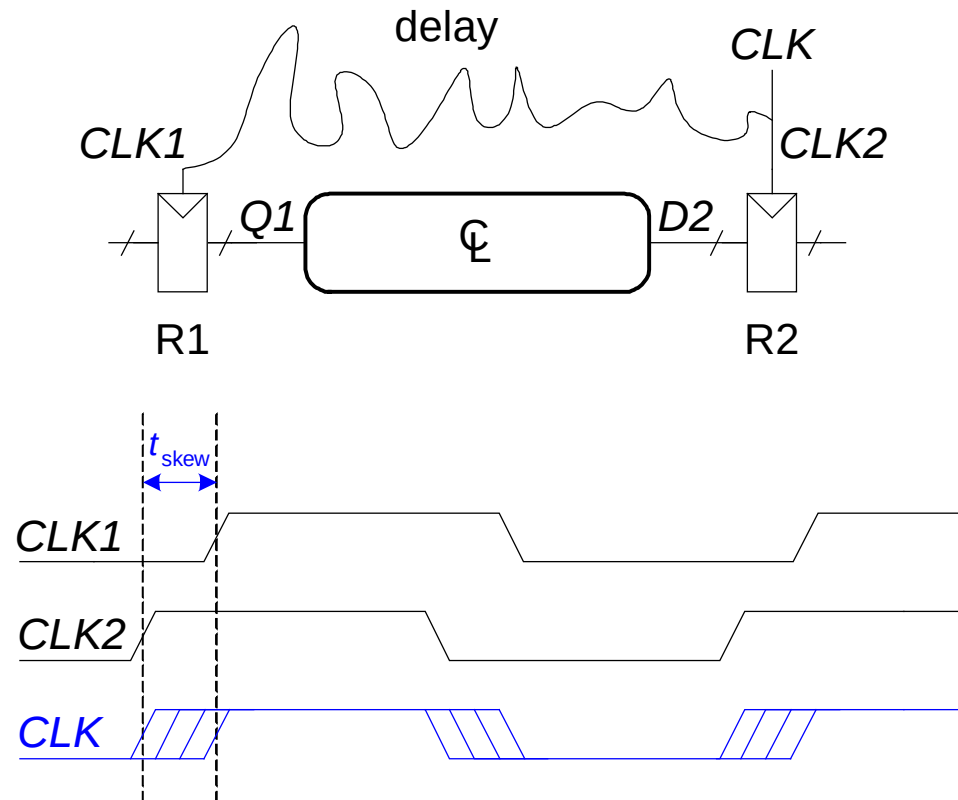
Обмеження часу утримання:

$$t_{ccq} + t_{cd} > t_{\text{hold}} ?$$

$$(30 + 50) \text{ пс} > 70 \text{ пс} ? \text{ **Так!**}$$

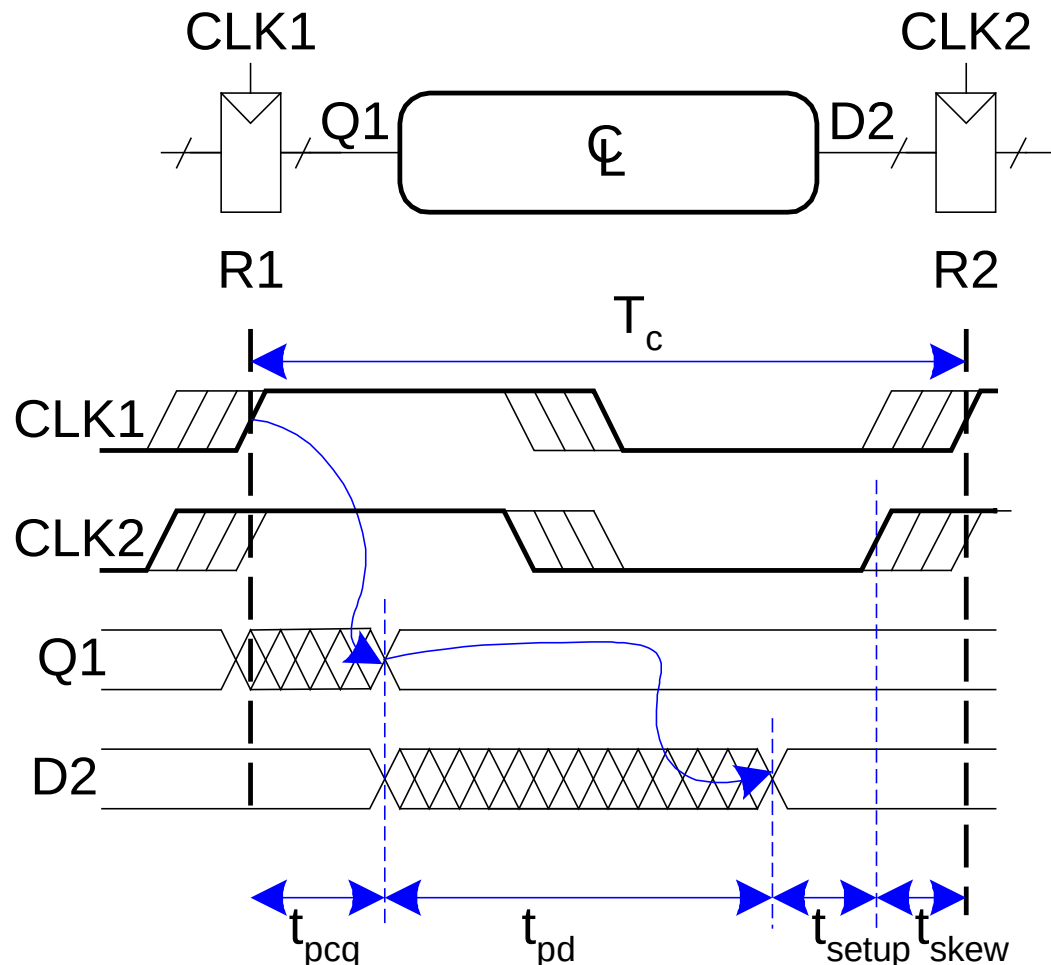
Розфазування тактових сигналів

- Тактові сигнали поступають на різні регістри схеми не одночасно
- **Розфазування**: відмінність у часі між фронтами тактових сигналів різних елементів
- Потрібно виконати **аналіз гіршого випадку**, щоб гарантувати виконання динамічної дисципліни для всіх регістрів схеми!



Обмеження часу передвстановлення з врахуванням розфазування тактових сигналів

- В найгіршому випадку CLK2 випереджають у часі CLK1

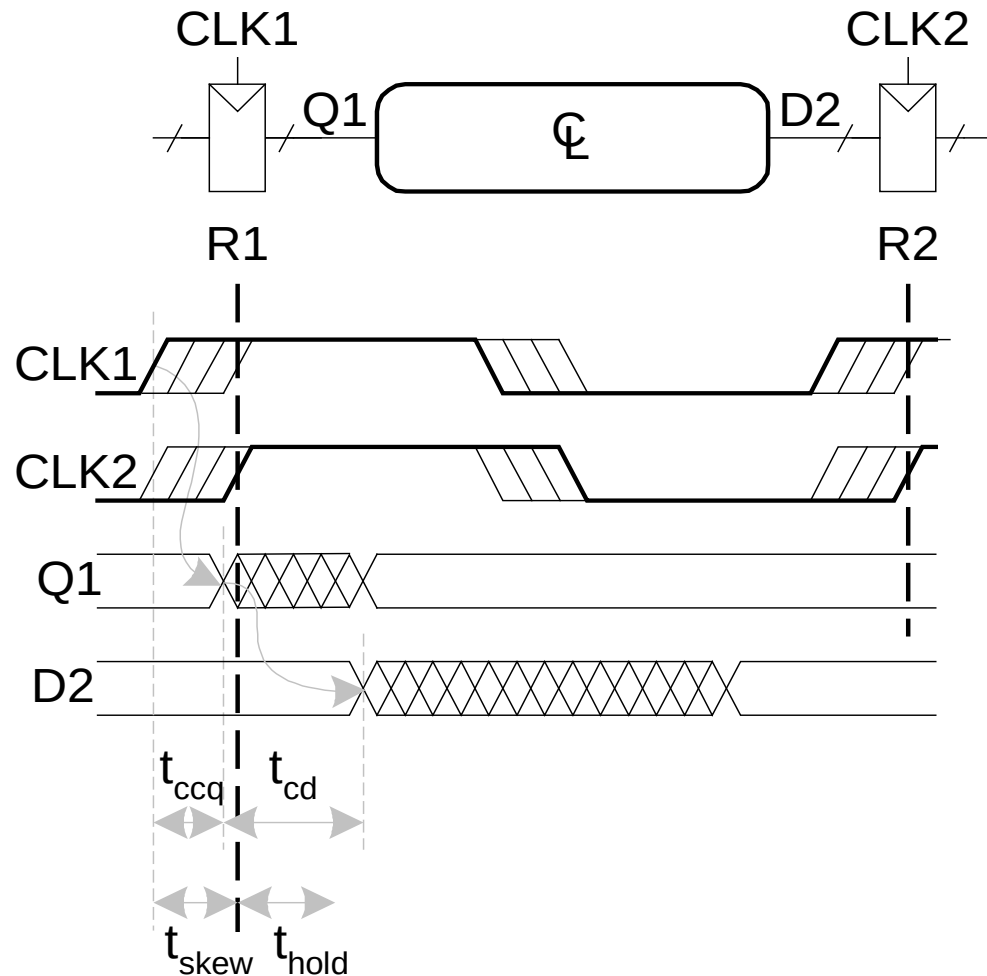


$$T_c \geq t_{pcq} + t_{pd} + t_{setup} + t_{skew}$$

$$t_{pd} \leq T_c - (t_{pcq} + t_{setup} + t_{skew})$$

Обмеження часу утримання з врахування розфазування тактових сигналів

- У гіршому випадку CLK2 відстають у часі від CLK1

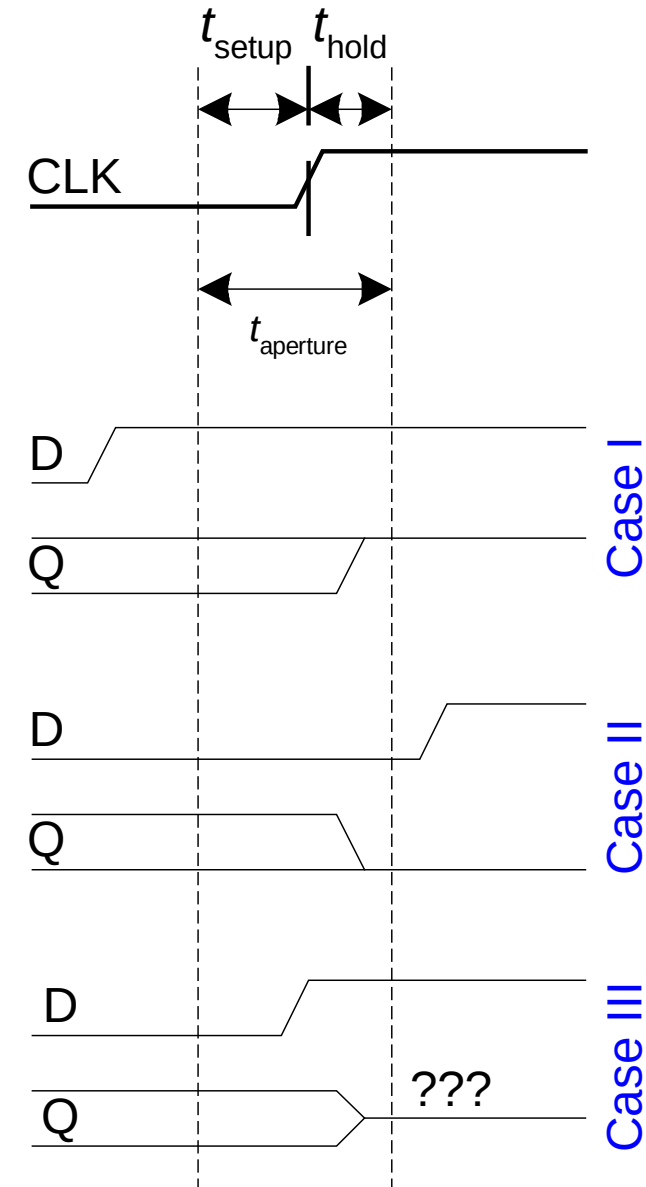
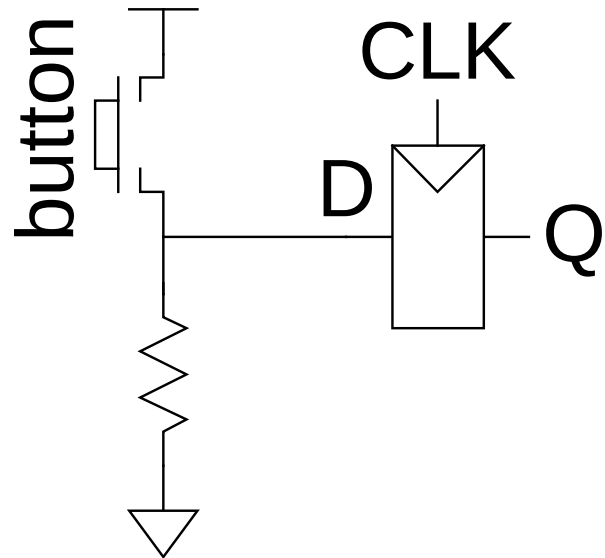


$$t_{ccq} + t_{cd} > t_{hold} + t_{skew}$$

$$t_{cd} > t_{hold} + t_{skew} - t_{ccq}$$

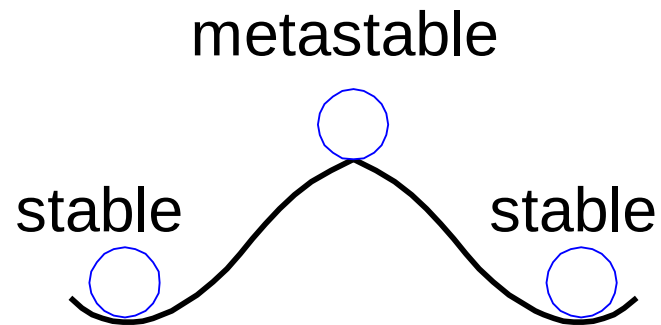
Порушення динамічної дисципліни

- Асинхронні (наприклад користувачькі) входи можуть приводити до порушення динамічної дисципліни



Метастабільність

- **Бістабільні прилади:** два стабільні стани і метастабільний стан між ними
- **Тригер** два стабільні стани (0 і 1) і один метастабільний стан
- Якщо тригер попадає у метастабільний стан, він може знаходитися в ньому необмежено довго



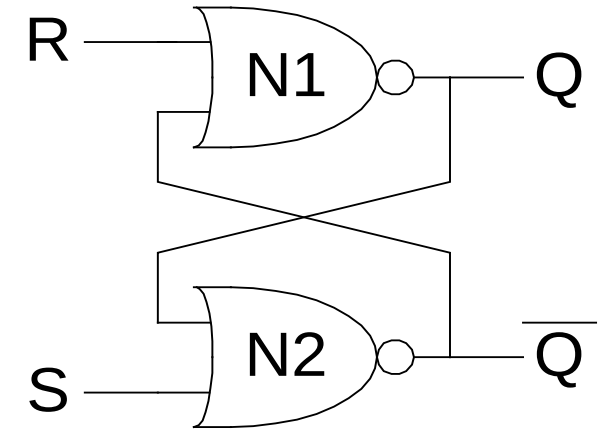
Внутрішня структура тригера

- Тригер має **зворотний зв'язок**: якщо Q знаходиться десь між 1 і 0, то перехресно з'єднані елементи встановлюють вихід в один з двох станів (1 або 0)
- Метастабільний сигнал: якщо він не прийняв коректне значення 0 або 1
- Якщо вхід тригера змінюється у випадковий момент, **ймовірність того, що його вихід Q буде в метастабільному стані через час t :**

$$P(t_{\text{res}} > t) = (T_0/T_c) e^{-t/\tau}$$

t_{res} : час переходу в 1 або 0

T_0, τ : характеристики схеми

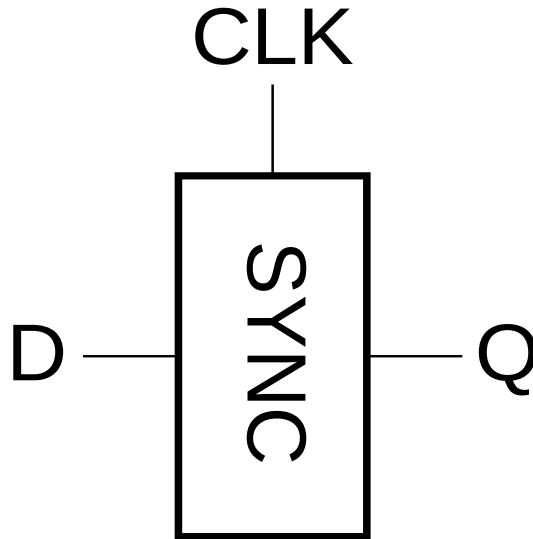


Вихід з метастабільного стану

- Інтуїтивно зрозуміло, що:
 - T_0/T_c : описує ймовірність того, вхід зміниться у невідповідний (апертурний) час
 - τ : часова константа, яка задає як швидко тригер виходить з метастабільного стану
- Якщо почекати достатно довго (t), більше як затримка поширення t_{pcq} , то з високою ймовірністю тригер перейде у стабільний стан 0 або 1

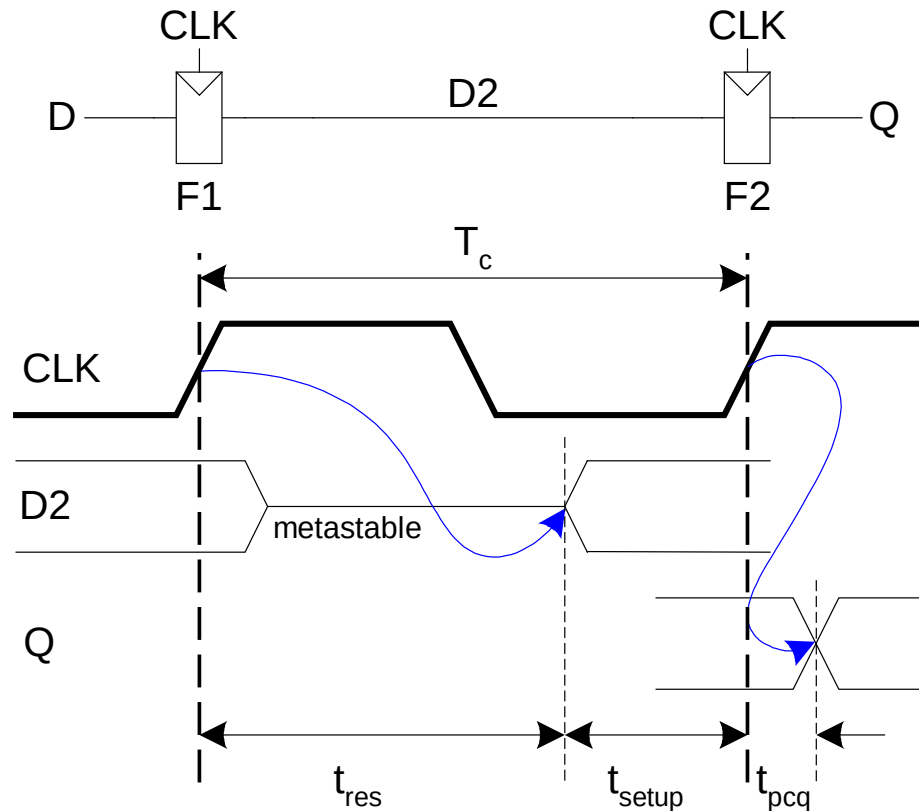
Синхронізатори

- **Наявність асинхронних входів неминуча** (інтерфейс користувача, системи з різними тактовими сигналами і т. д.)
- **Функція синхронізатора:** зробити ймовірність збою (вихід Q знаходиться в метастабільному стані) достатньо малою
- Синхронізатор не може довести ймовірність збою до 0



Внутрішня структура синхронізатора

- Синхронізатор: будується з двох послідовно з'єднаних тригерів
- Припустимо, що D змінюється під час фіксації F1
- У внутрішнього сигналу D2 є час $(T_c - t_{\text{setup}})$ для переходу в 1 або 0



Середній час напрацювання на відмову

- Якщо асинхронний вхід змінюється один раз за секунду, то ймовірність відмови за одну секунду буде дорівнювати $P(\text{failure})$

$$P(\text{failure}) = (T_0/T_c) e^{-(T_c - t_{\text{setup}})/\tau}$$

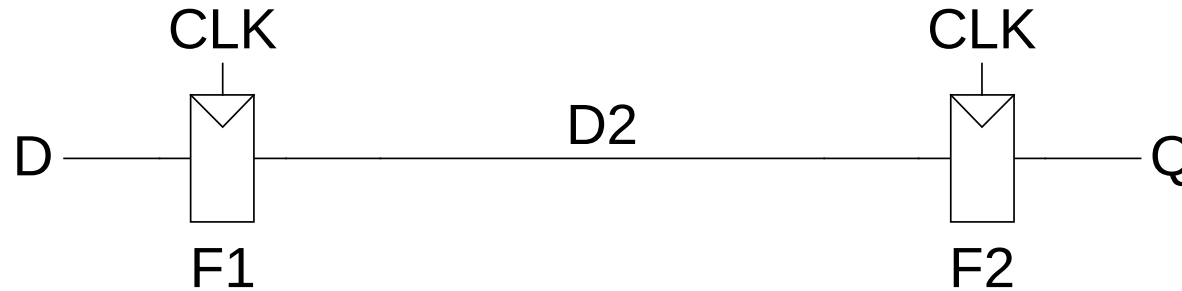
- Якщо вхід змінюється N разів за секунду, то ймовірність відмови за секунду буде:

$$P(\text{failure})/\text{сек} = N(T_0/T_c) e^{-(T_c - t_{\text{setup}})/\tau}$$

- Збій синхронізатора відбувається в середньому $1/[P(\text{failure})/\text{сек}]$
- Ця величина називається *середнім часом напрацювання на відмову* (*mean time between failures*, MTBF):

$$\text{MTBF} = 1/[P(\text{failure})/\text{сек}] = (T_c/NT_0) e^{(T_c - t_{\text{setup}})/\tau}$$

Приклад синхронізатора



- Припустимо: $T_c = 1/500 \text{ МГц} = 2 \text{ нс}$ $\tau = 200 \text{ пс}$
 $T_0 = 150 \text{ пс}$ $t_{\text{setup}} = 100 \text{ пс}$
 $N = 1$ подій за секунду
- Чому дорівнює ймовірність відмови? MTBF?
$$P(\text{failure}) = (150 \text{ пс} / 2 \text{ нс}) e^{-(1.9 \text{ нс}) / 200 \text{ пс}} = 5.6 \times 10^{-6}$$

$$P(\text{failure})/c = 10 \times (5.6 \times 10^{-6}) = 5.6 \times 10^{-5} / \text{с}$$

$$\text{MTBF} = 1/[P(\text{failure})/c] \approx 5 \text{ годин}$$

Паралелізм

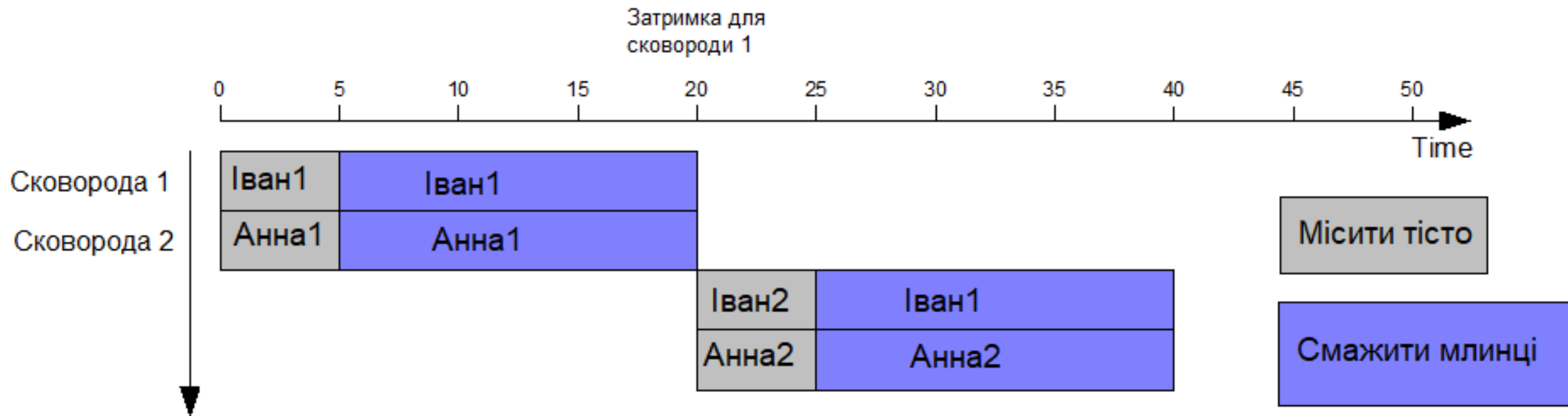
- Два типи паралелізму:
 - Просторовий паралелізм:
 - Декілька копій апаратних блоків в один і той же час виконують декілька задач
 - Часовий паралелізм:
 - розбиття задачі на декілька ступенів (конвеєризація), наприклад, збиральна лінія

Визначення паралелізму

- **Змістовий пакет (Token):** Набір вхідної інформації, який обробляється для того, щоб отримати вихідну інформацію
- **Латентність (latency):** Час проходження одного змістового пакету через всю систему з її входу на вхід
- **Пропускна здатність (throughput):** Кількість змістових пакетів, які обробляються системою в одиницю часу

Паралелізм збільшує пропускну здатність

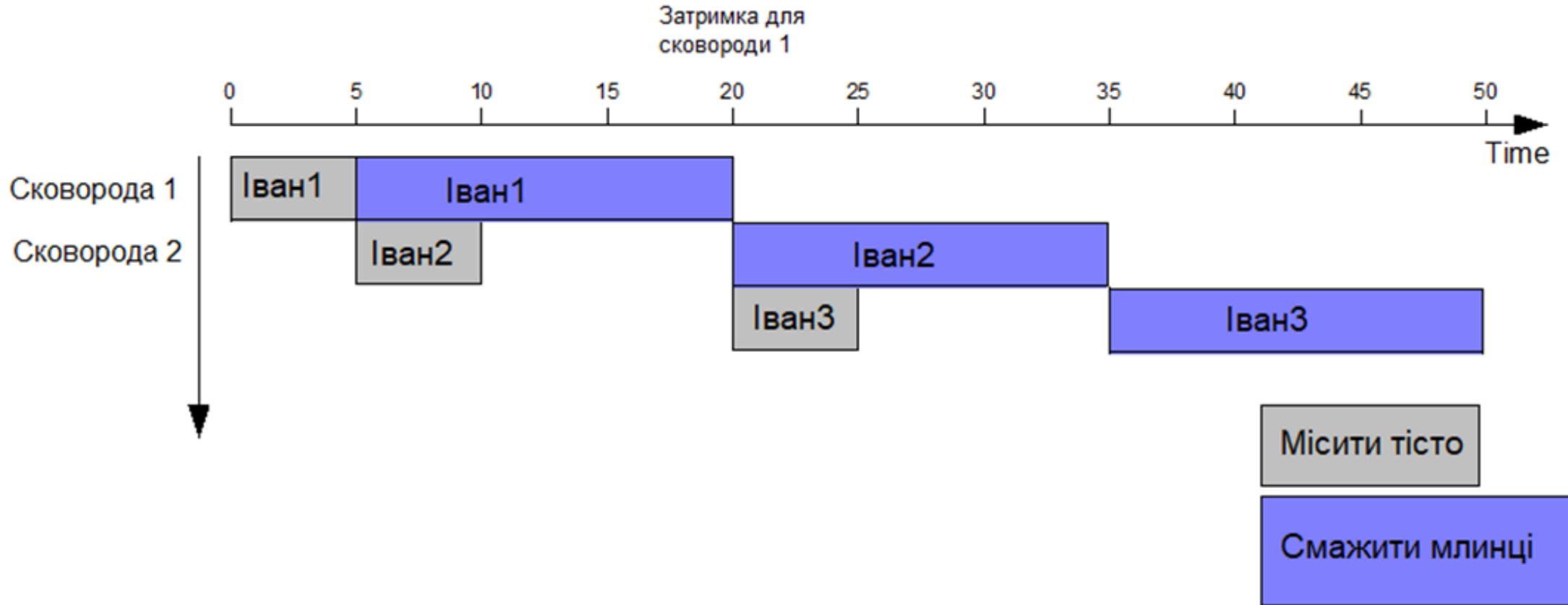
Просторовий паралелізм



Латентність = $5 + 15 = 20$ хвилин = **$1/3$ год**

Пропускна здатність = $2 \text{ сковороди} / 1/3 \text{ час} = \mathbf{6 \text{ сковорід/год}}$

Часовий паралелізм



Латентність = $5 + 15 = 20$ хвилин = **$1/3$ год**

Пропускна здатність = $1 \text{ сковорода} / 1/4 \text{ час} = \mathbf{4 \text{ сковороди/год}}$