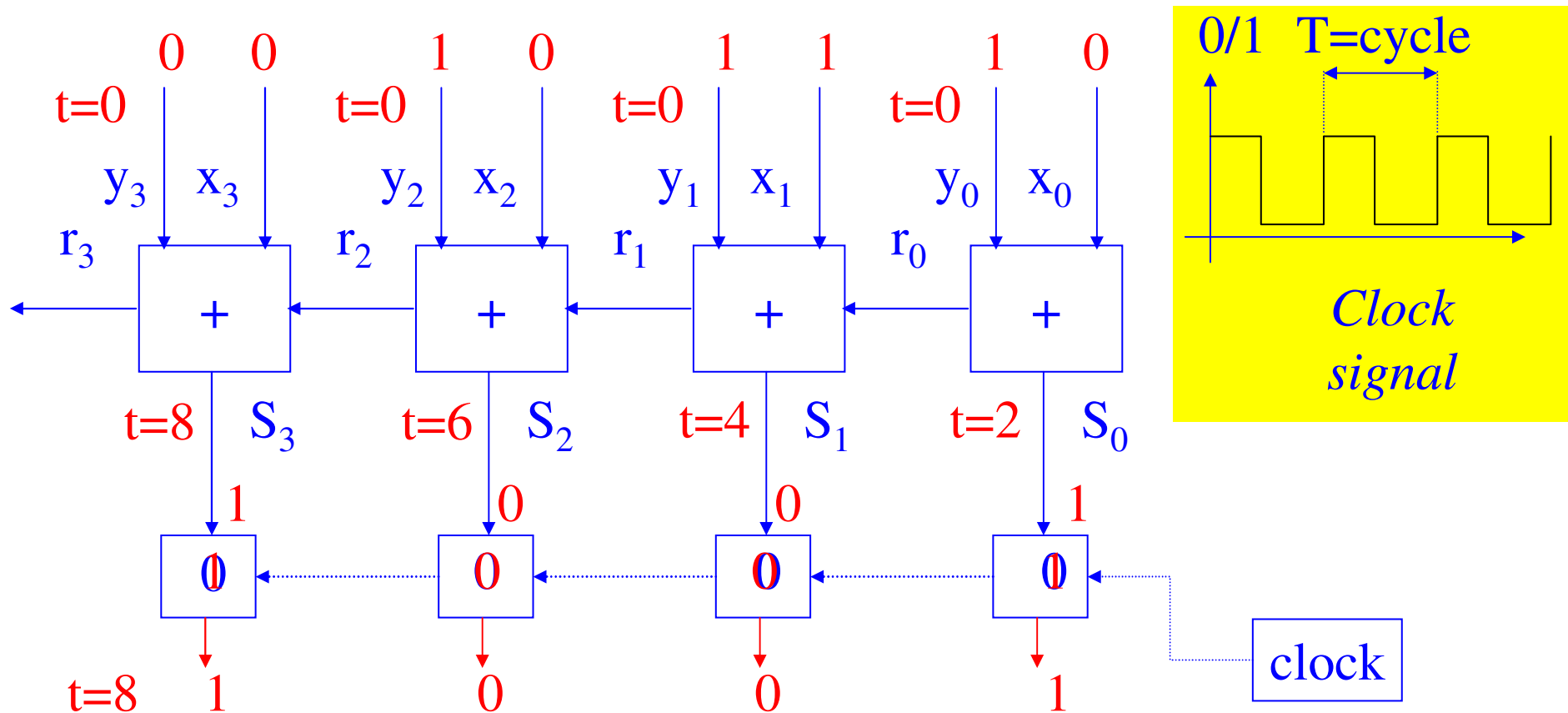


# Plan

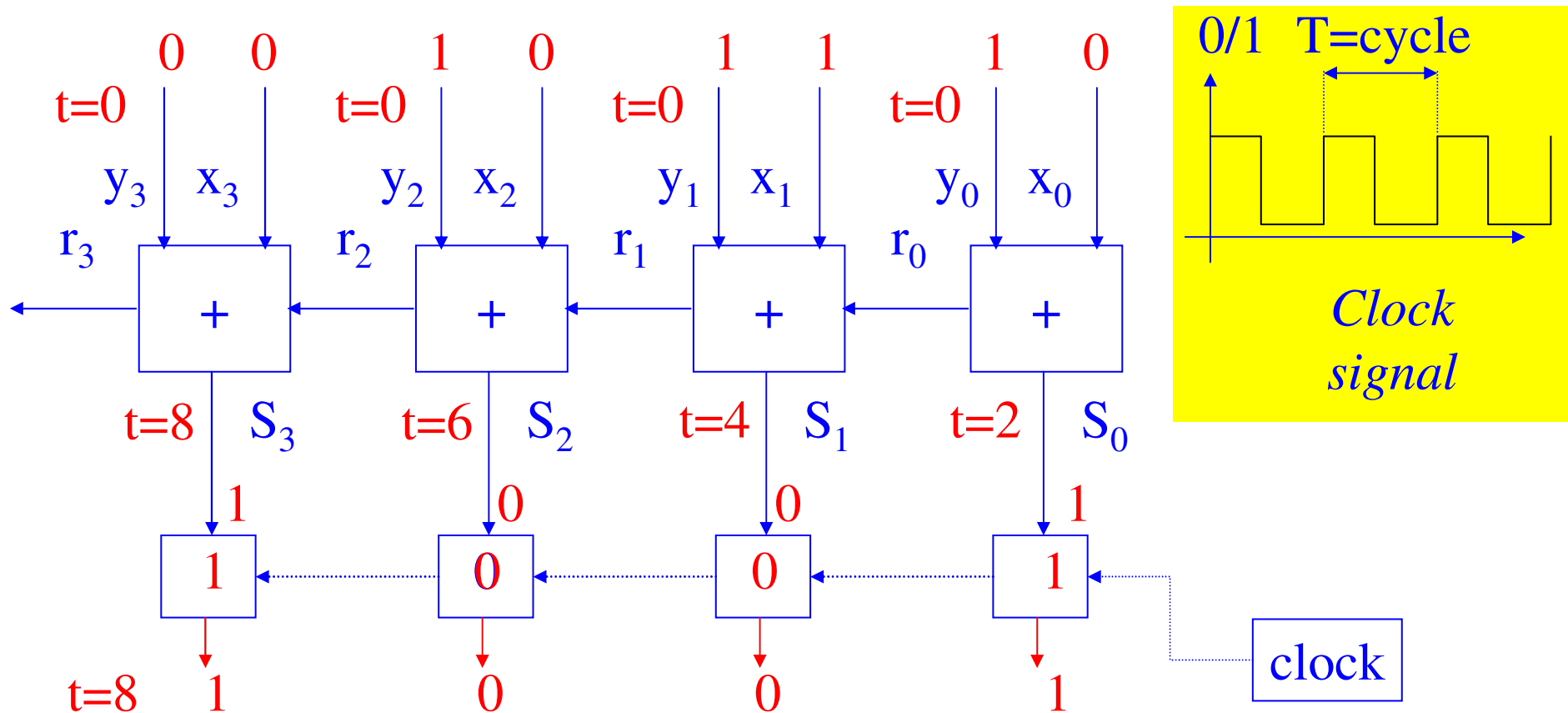
- Circuits logiques
- Représentation des nombres
- Unité Arithmétique et Logique
- **Time and memory**
- Contrôle et jonction des composants
- Evolution des ordinateurs – Historique
- Un microprocesseur simple
- Programmation d'un microprocesseur
- Système complet
- Les microprocesseurs actuels
- Exploitation de la performance des microprocesseurs

# Time



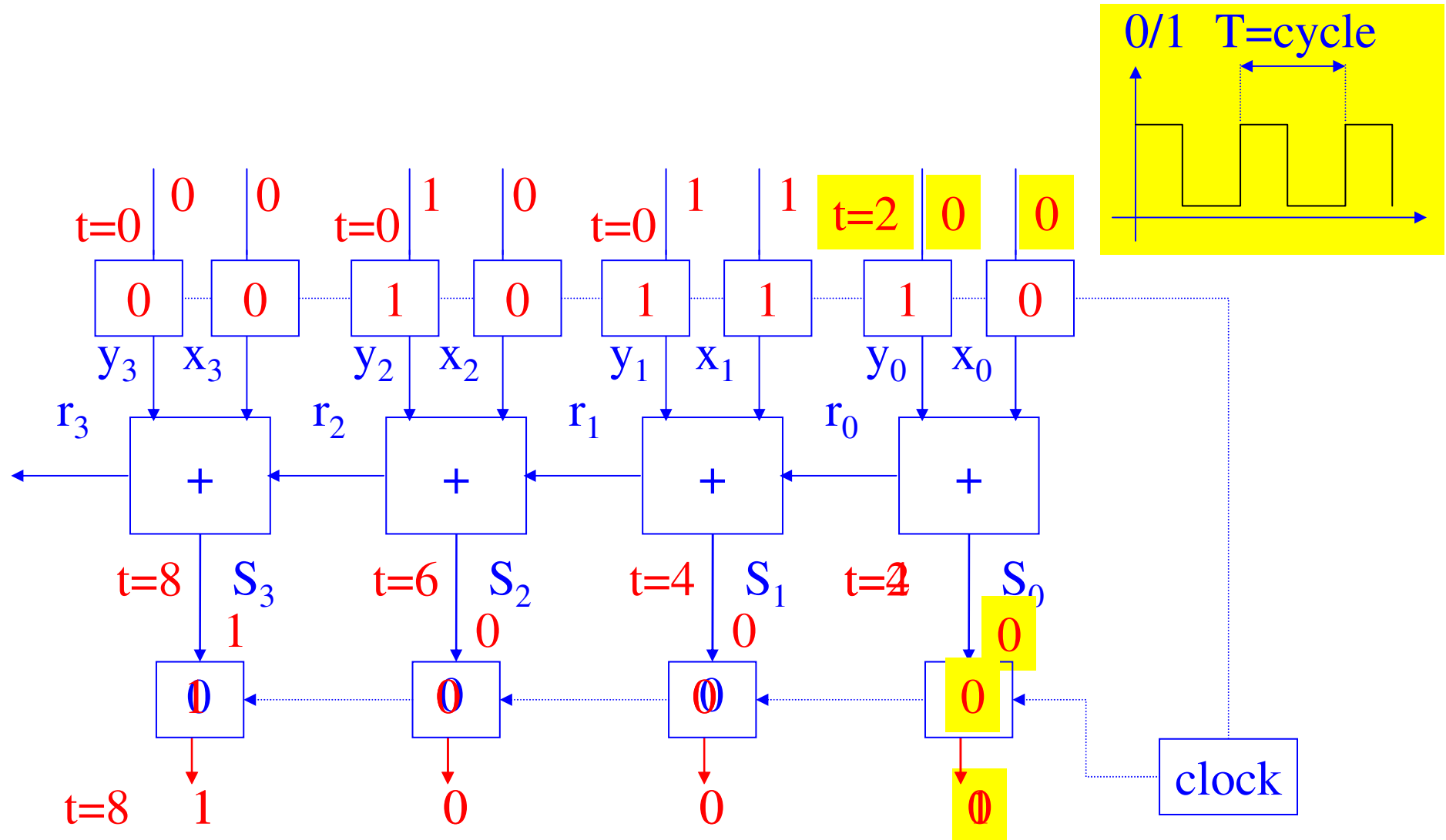
- Synchronization:
    - Wait for the output of all adders to be valid before propagating result
    - Estimate computation time to find appropriate delay
  - Insert « barriers » to synchronize outputs
- Control of « barriers »: **clock**, periodic signal

# Notion de Temps



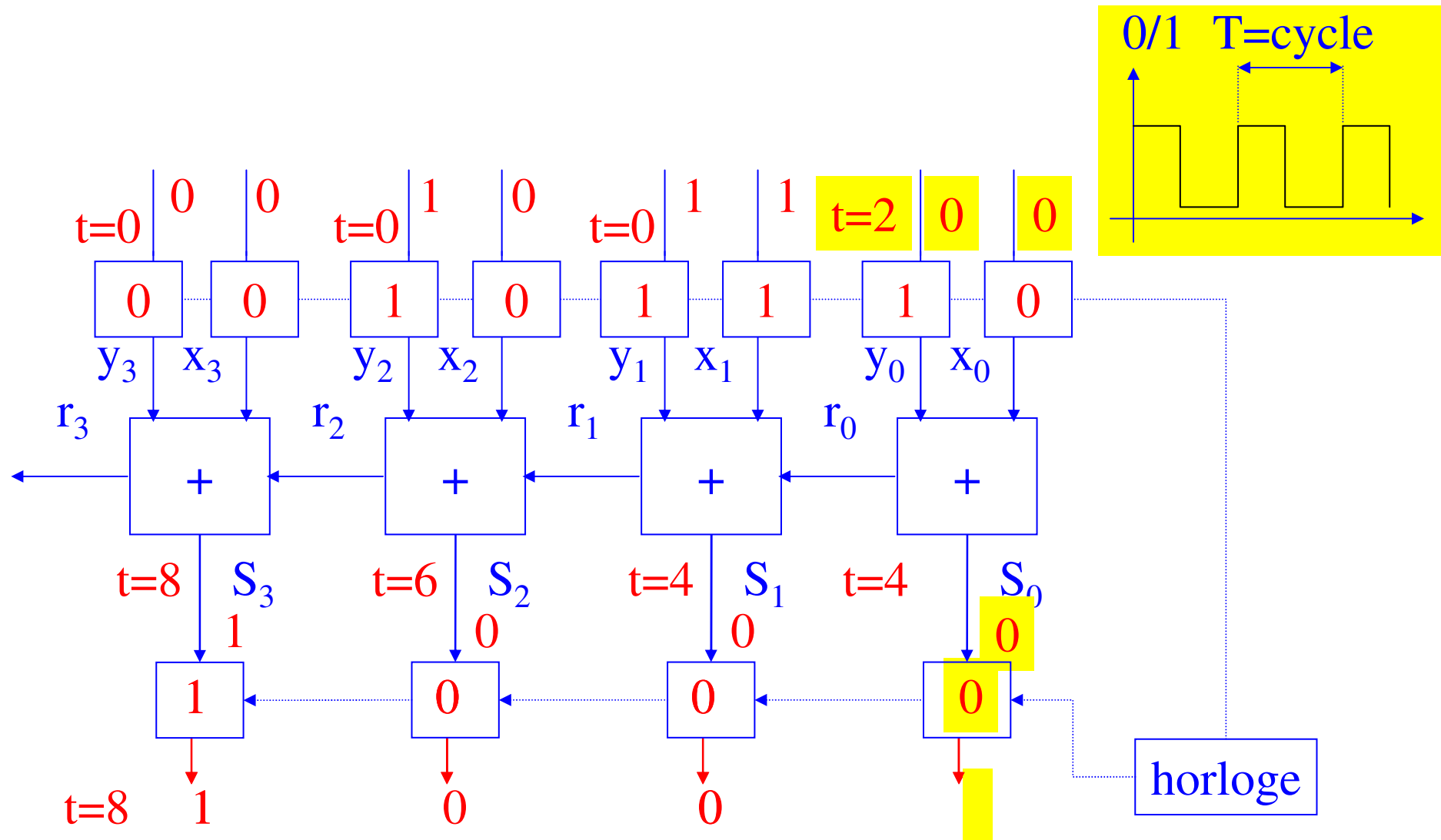
- Synchronization:
    - Wait for the output of all adders to be valid before propagating result
    - Estimate computation time to find appropriate delay
  - Insert « barriers » to synchronize outputs
- Control of « barriers »: **clock**, periodic signal

# Notion de Mémorisation



- Hold inputs during computation → memory

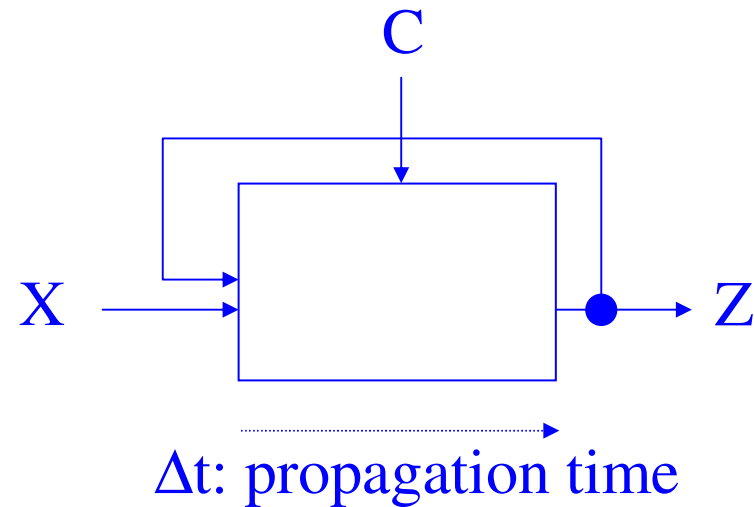
# Memory



- Maintenir les entrées pendant la durée du calcul  
→ mémorisation.

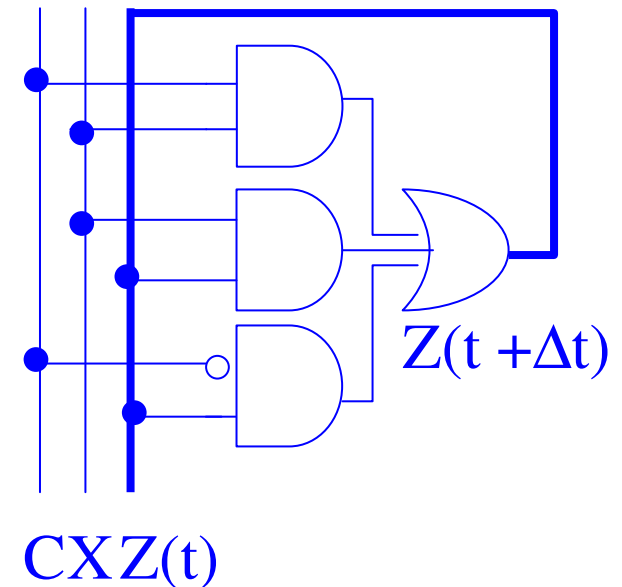
# How To Implement A Memory Circuit ?

C	X	Z(t)	Z(t+ Δt)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

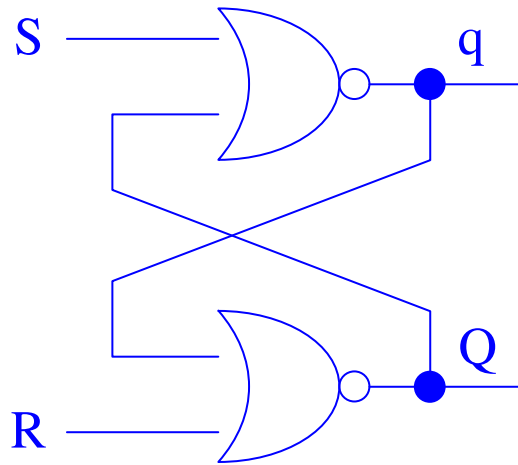


$$Z(t + \Delta t) = C'.Z(t) + X.Z(t) + C.X$$

- Barrier: input X, output Z, control using clock C:
  - C=0: memory  $\rightarrow Z(t+\Delta t) = Z(t)$
  - C=1: propagate information  $\rightarrow Z(t+\Delta t) = X$
- Loop: send output to input  $\rightarrow$  **memory**



# Latches



*Latch SR*

S	R	Q	Q <sup>+</sup>
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	<i>d</i>
1	1	1	<i>d</i>

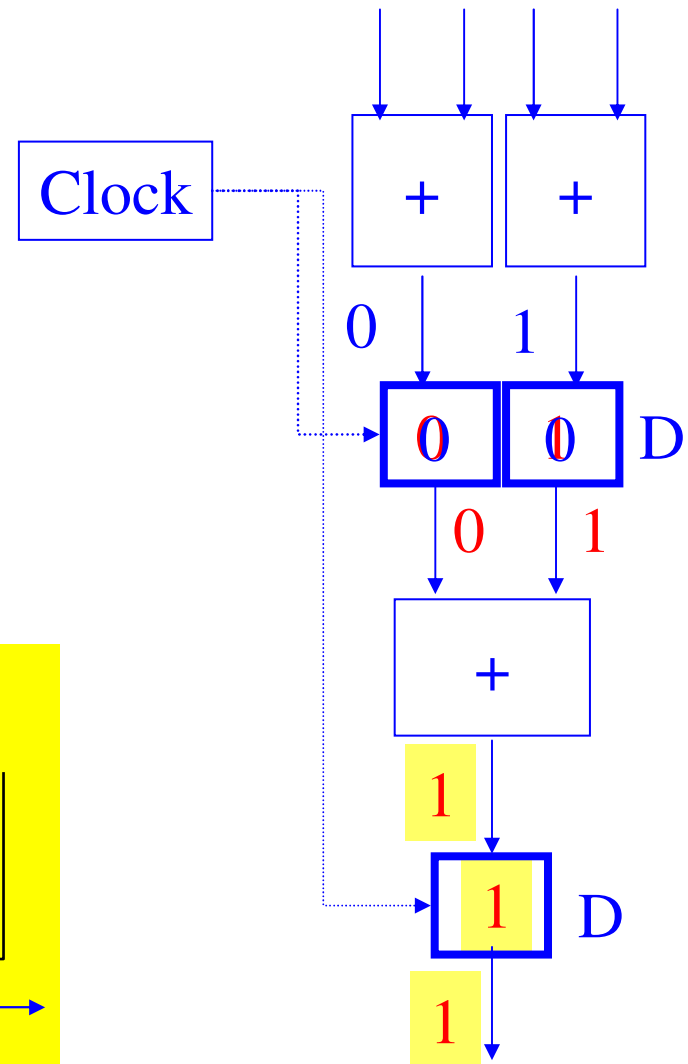
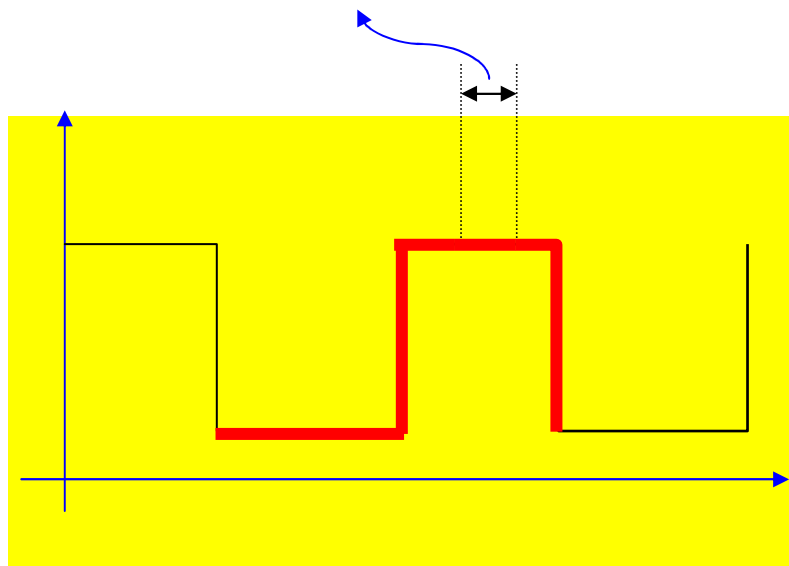
- Memorization circuits used as “building blocks”
- Different types of latches (number of control signals and behavior)
- SR= *Set/Reset*: memorize or set 1/0.



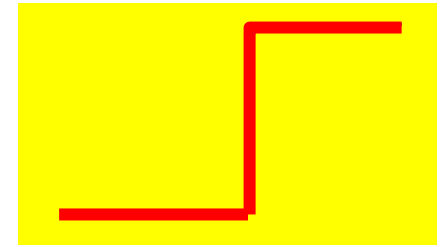
# Flips-Flops

- Operators/Computations usually in sequence
- A latch can store a result but not truly separate two operators

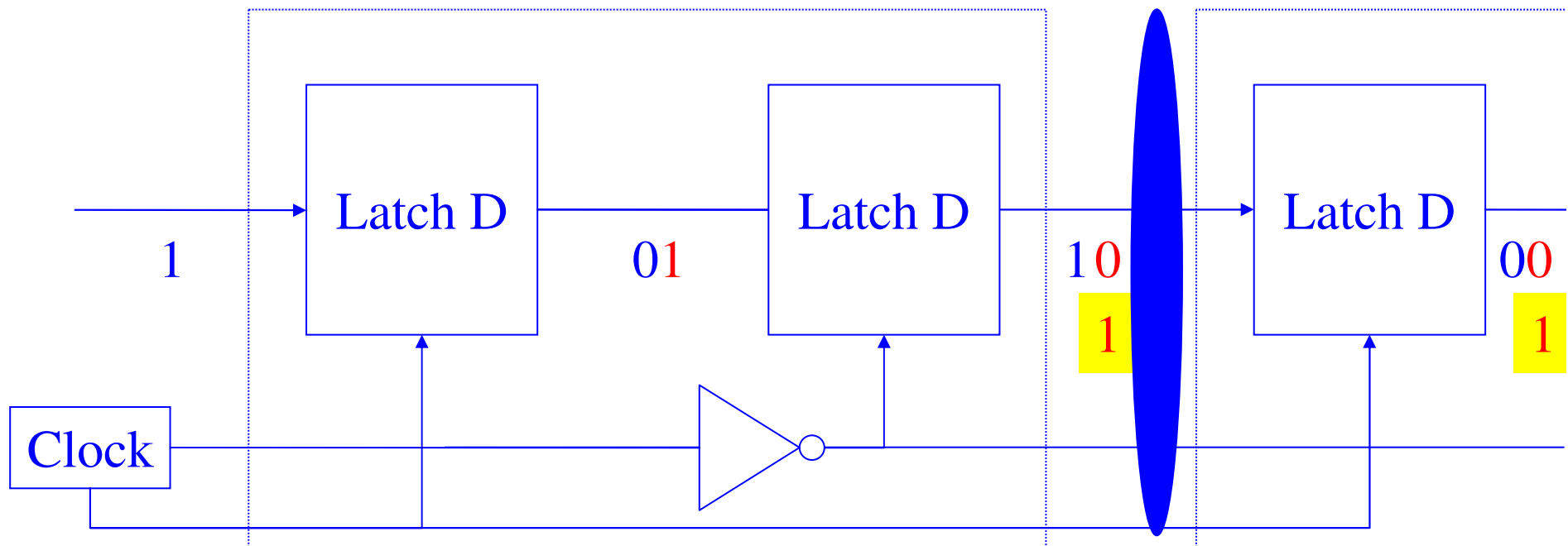
1-bit adder delay



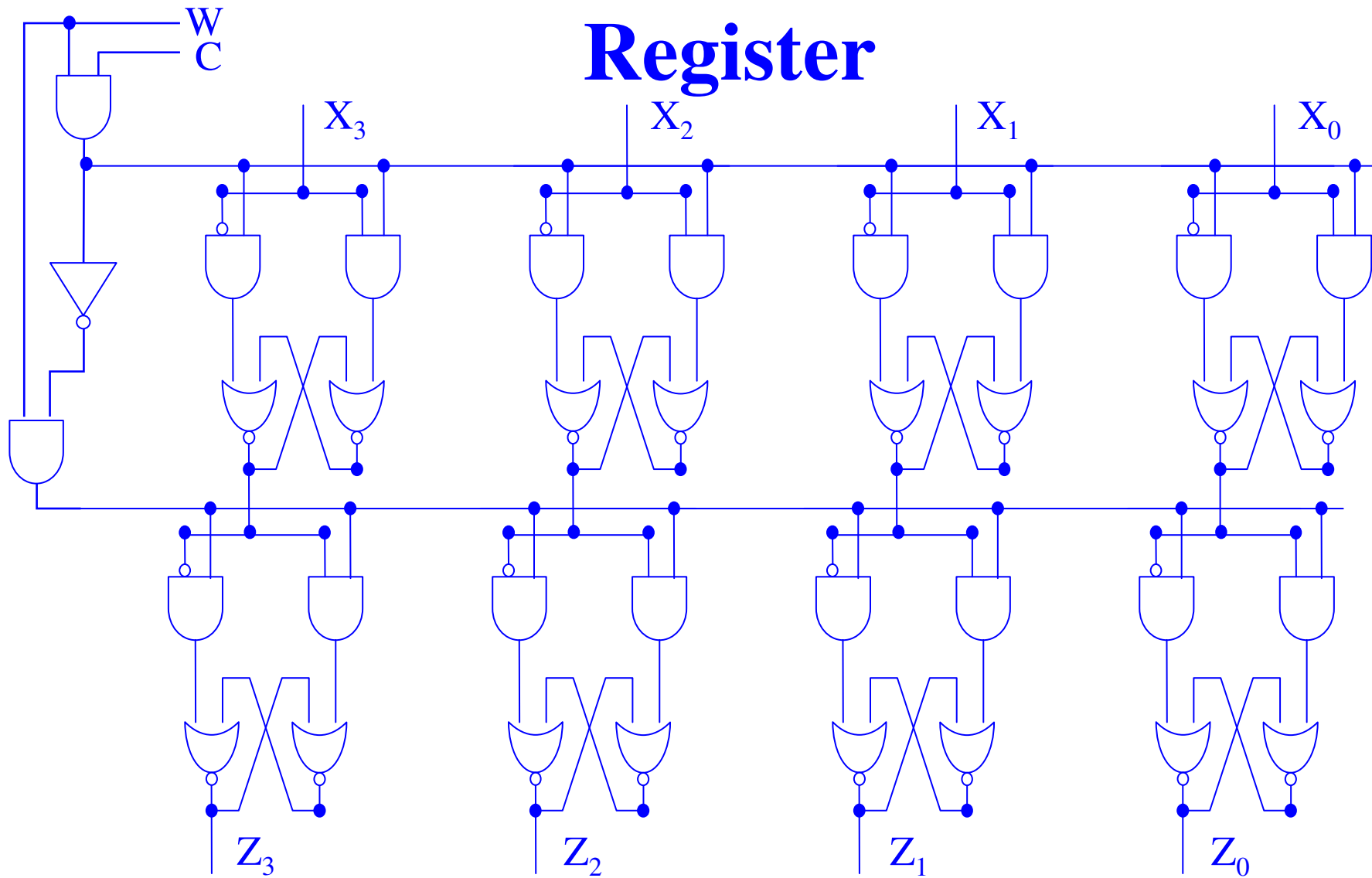
# Flips-Flops



## *Flip-Flop D*

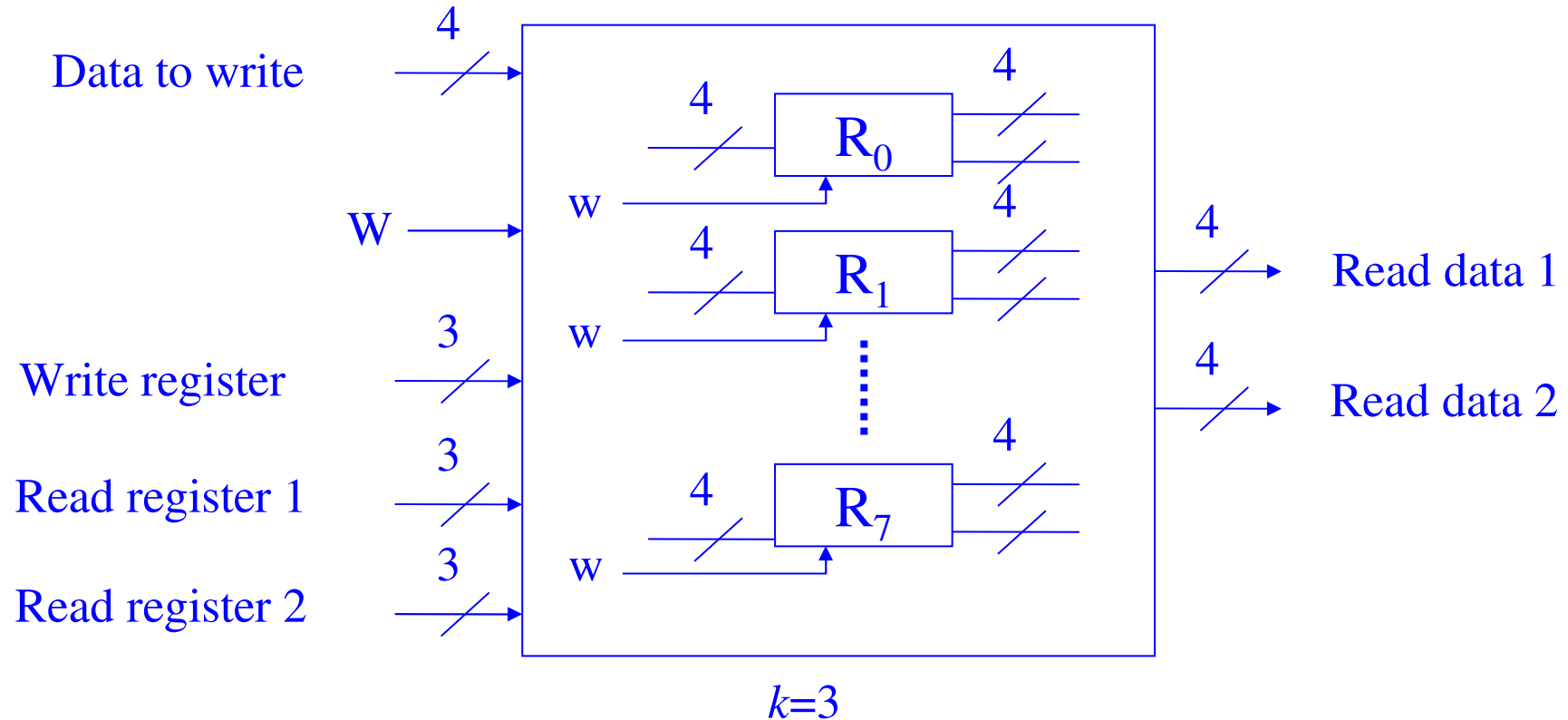


- Connect two latches, open on opposite clock phases:  
⇒ The signal can only go through one of the two latches ( $\approx$  air lock).



- $n$  flip-flops D in parallel
- Storing one word
- Read/Write; signal  $W=1 \rightarrow$  Write

# Register Bank



- Small  $2^k$  – word memory
- Usually, at least 1 write port, 2 read ports
- Access through register number ~ address ( $k$  bits).