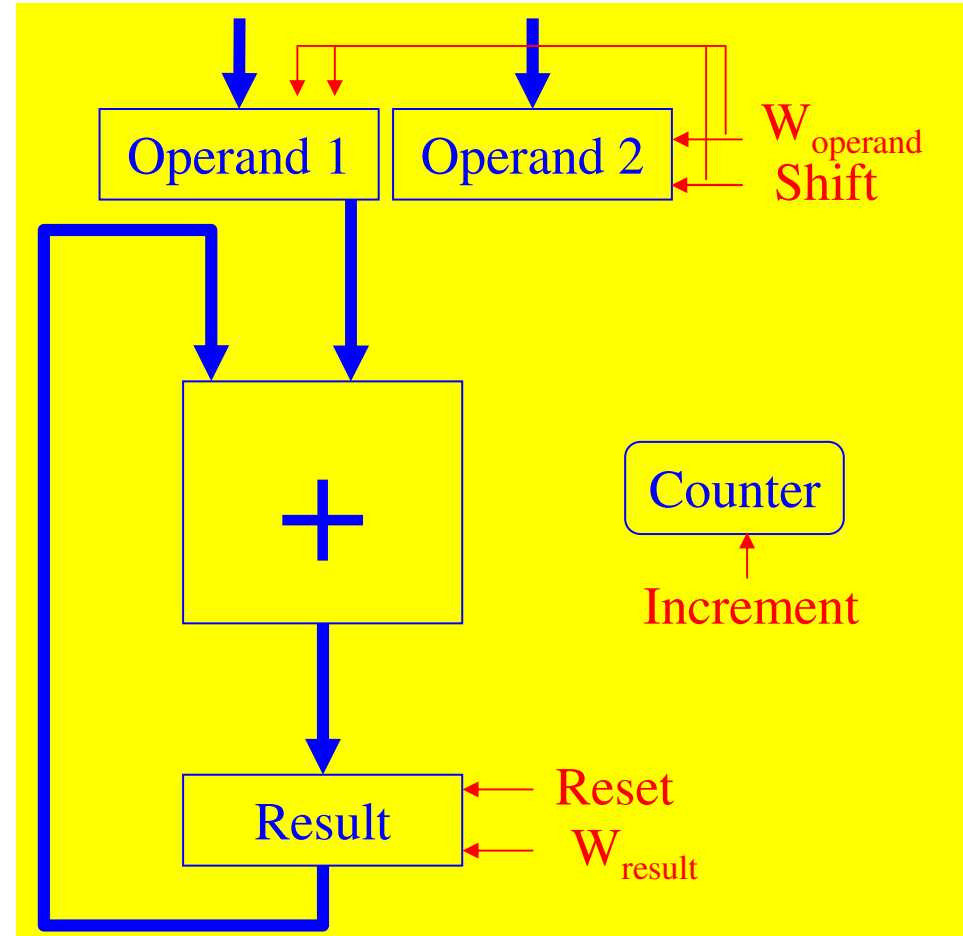
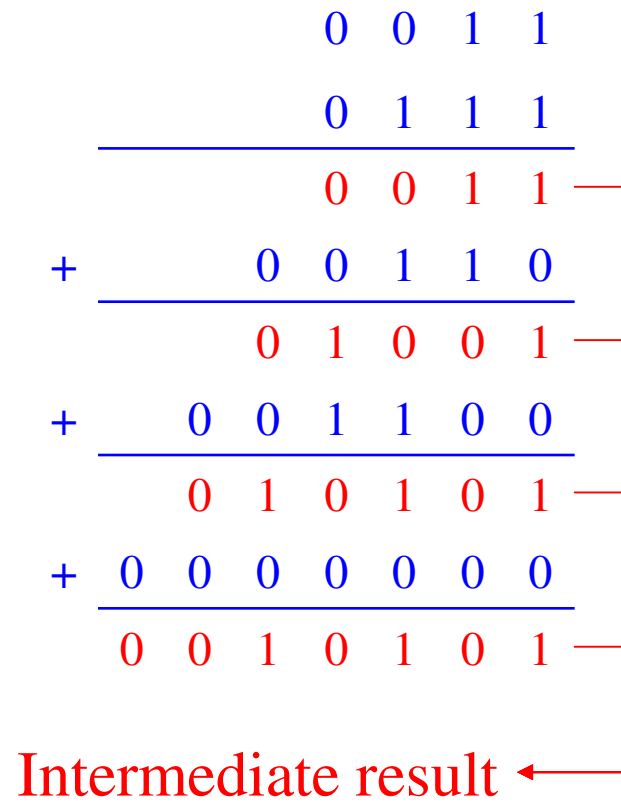


Overview

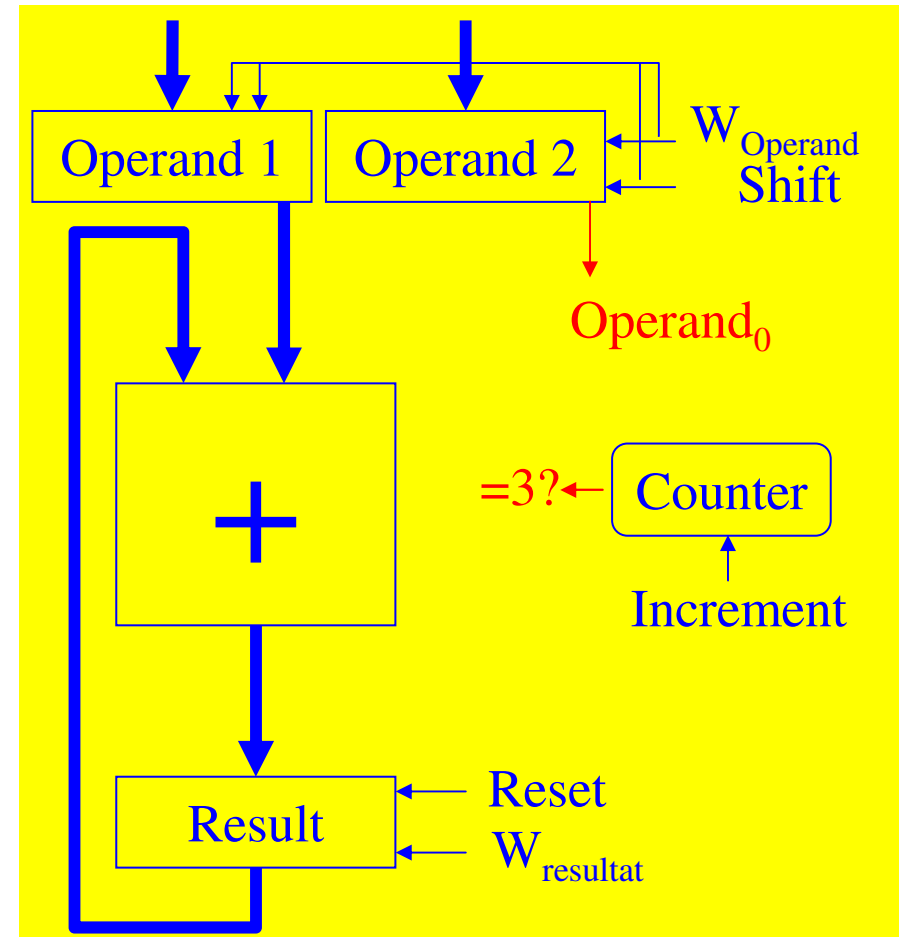
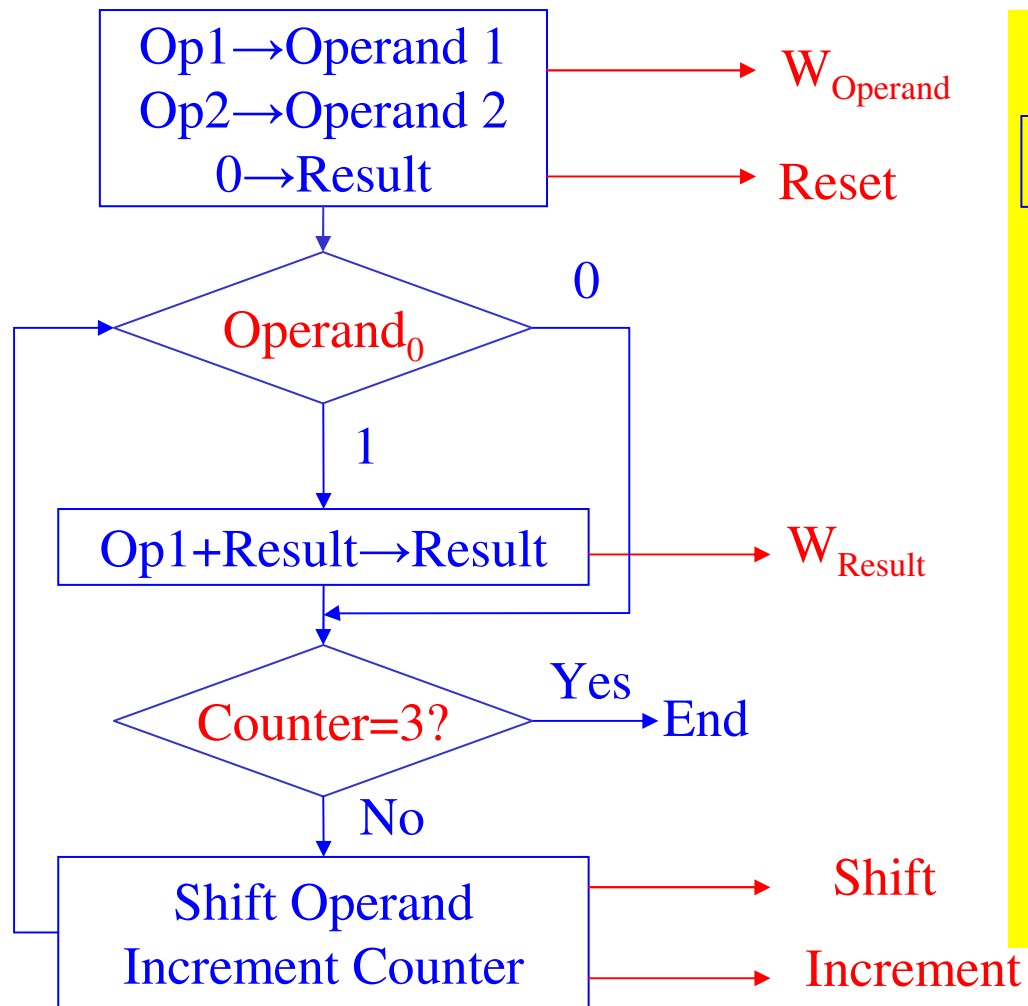
- Représentation des nombres
- Circuits logiques
- Unité Arithmétique et Logique
- Notions de temps et de mémorisation
- **Control and data paths**
- Evolution des ordinateurs – Historique
- Un microprocesseur simple
- Programmation d'un microprocesseur
- Système complet
- Les microprocesseurs actuels
- Exploitation de la performance des microprocesseurs

Control

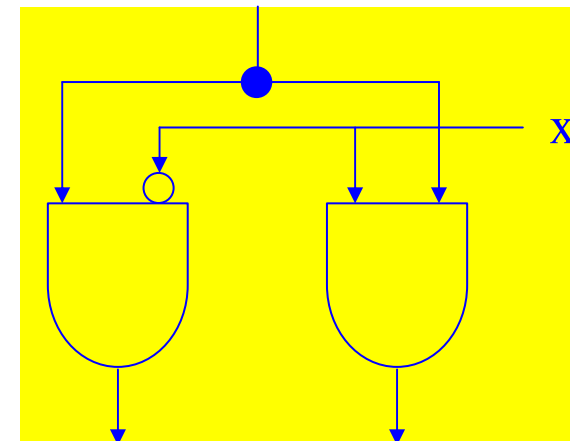
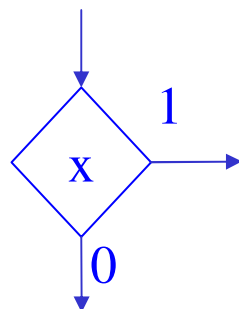
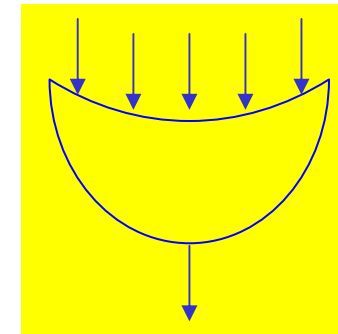
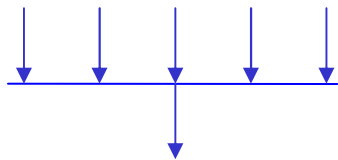
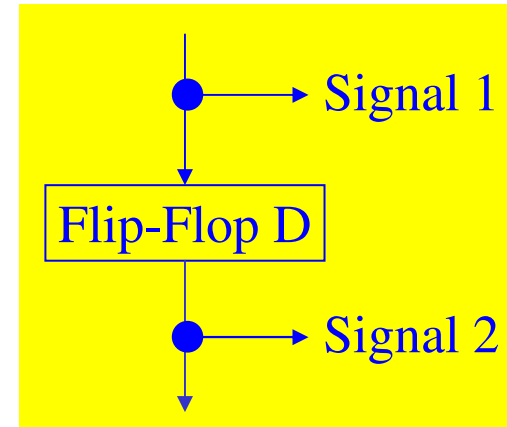
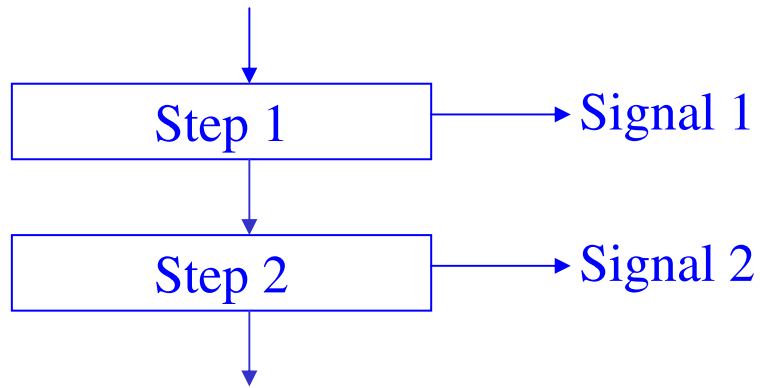


- Example with sequential multiplication

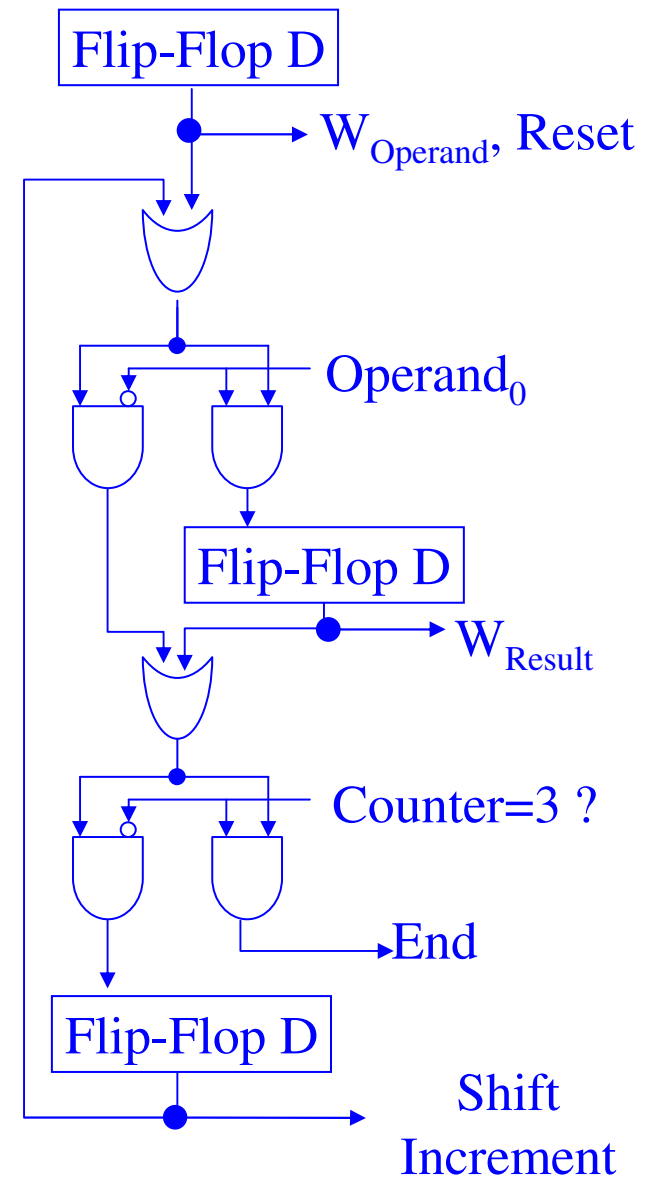
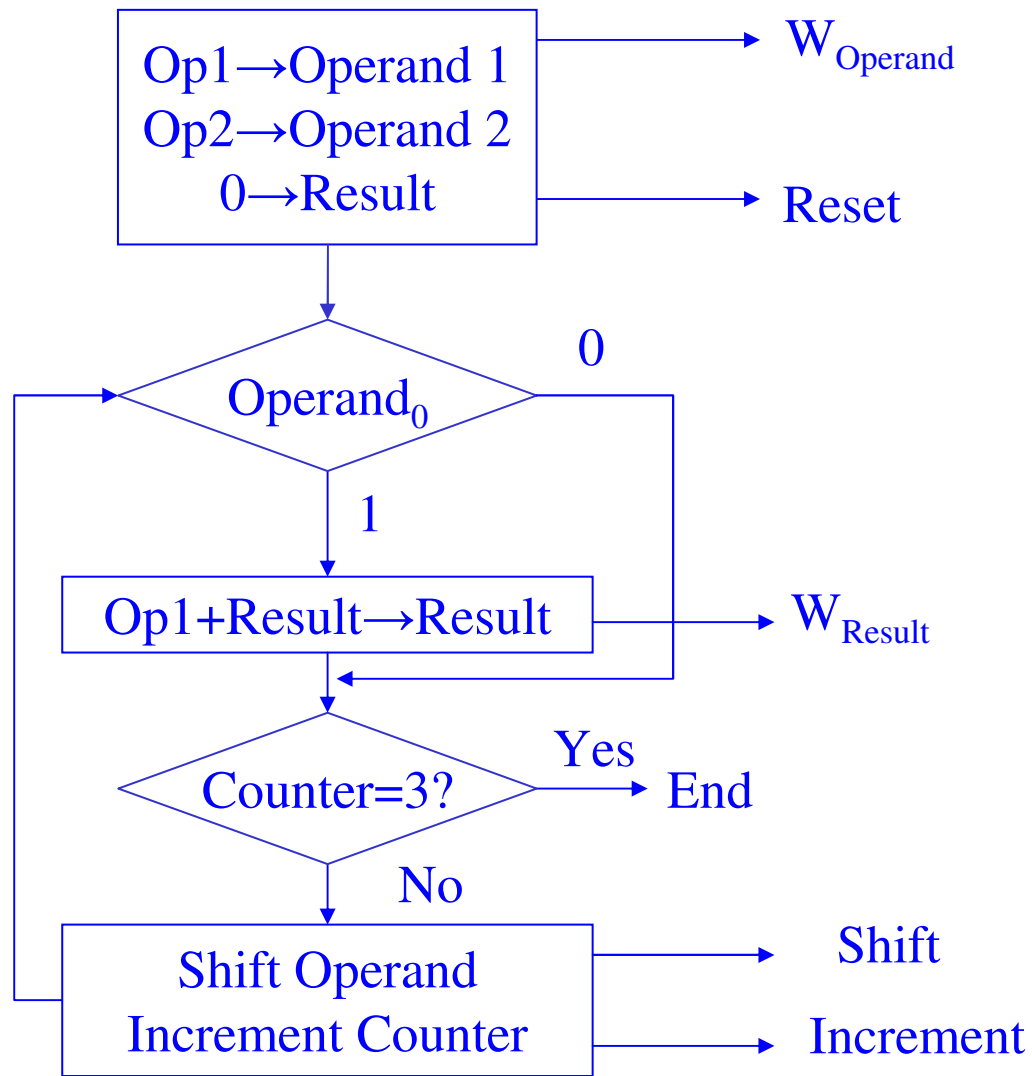
Multiplier Control



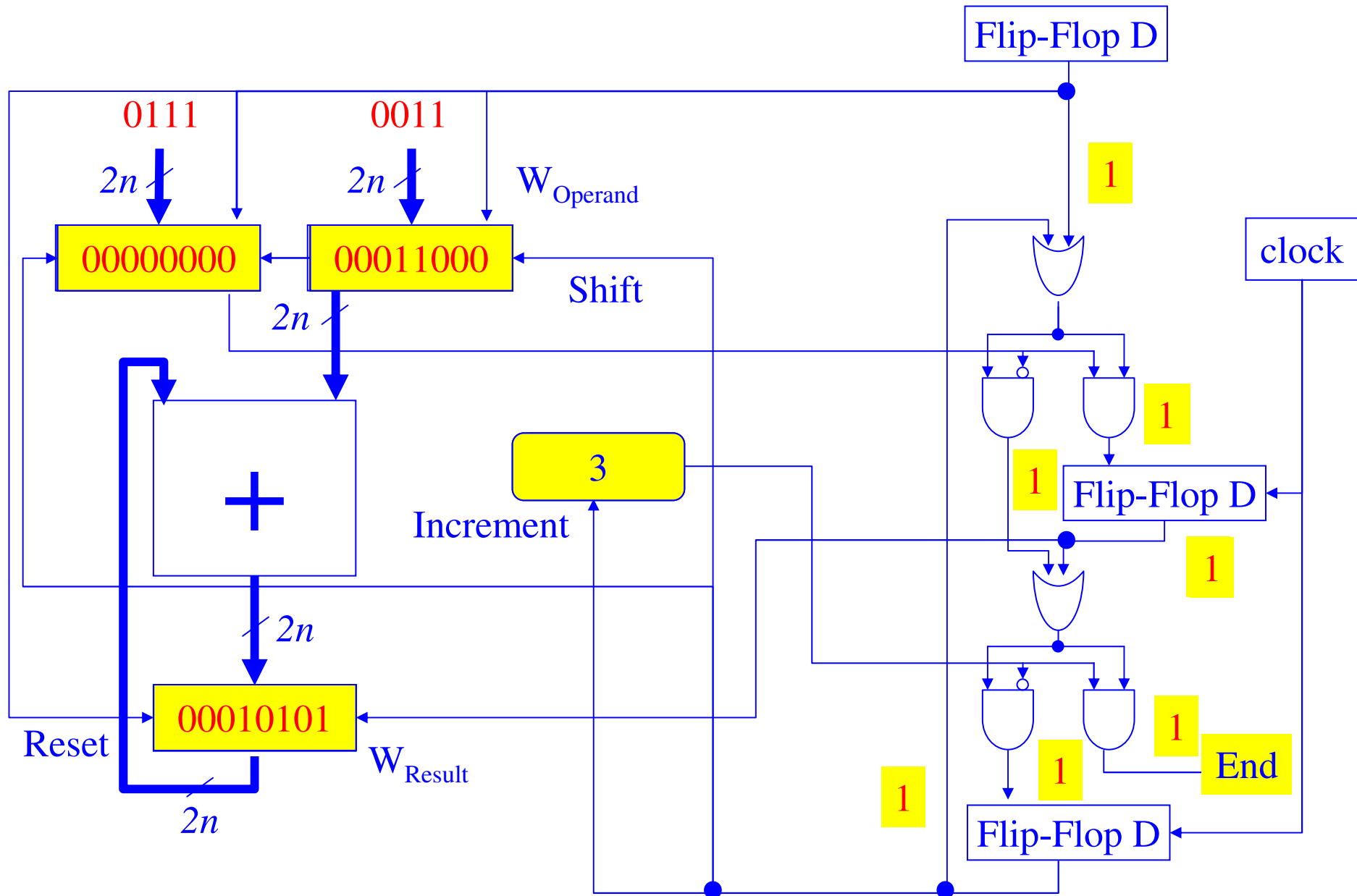
Converting a Flowchart into a Control Circuit



Multiplier Control Circuit

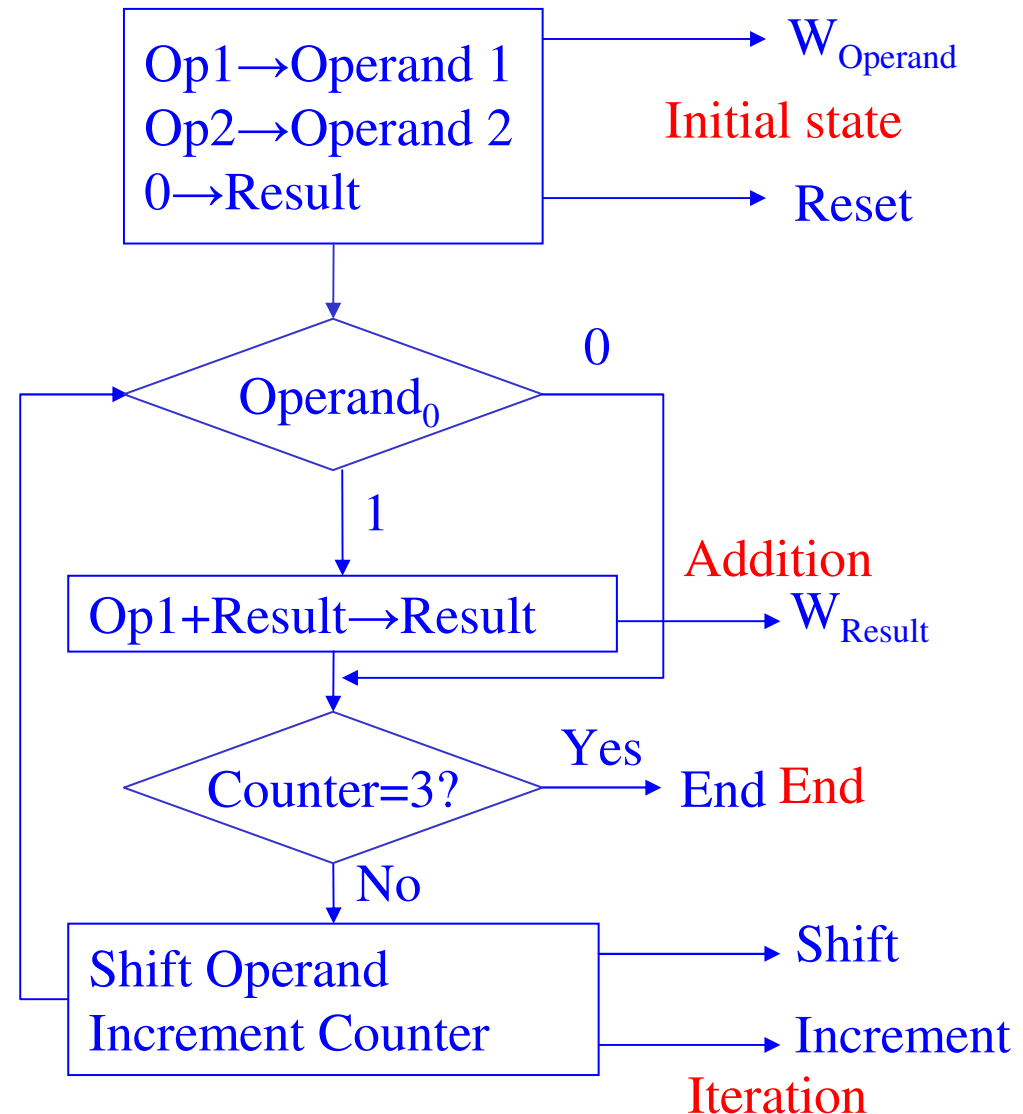


Multiplier Control Circuit



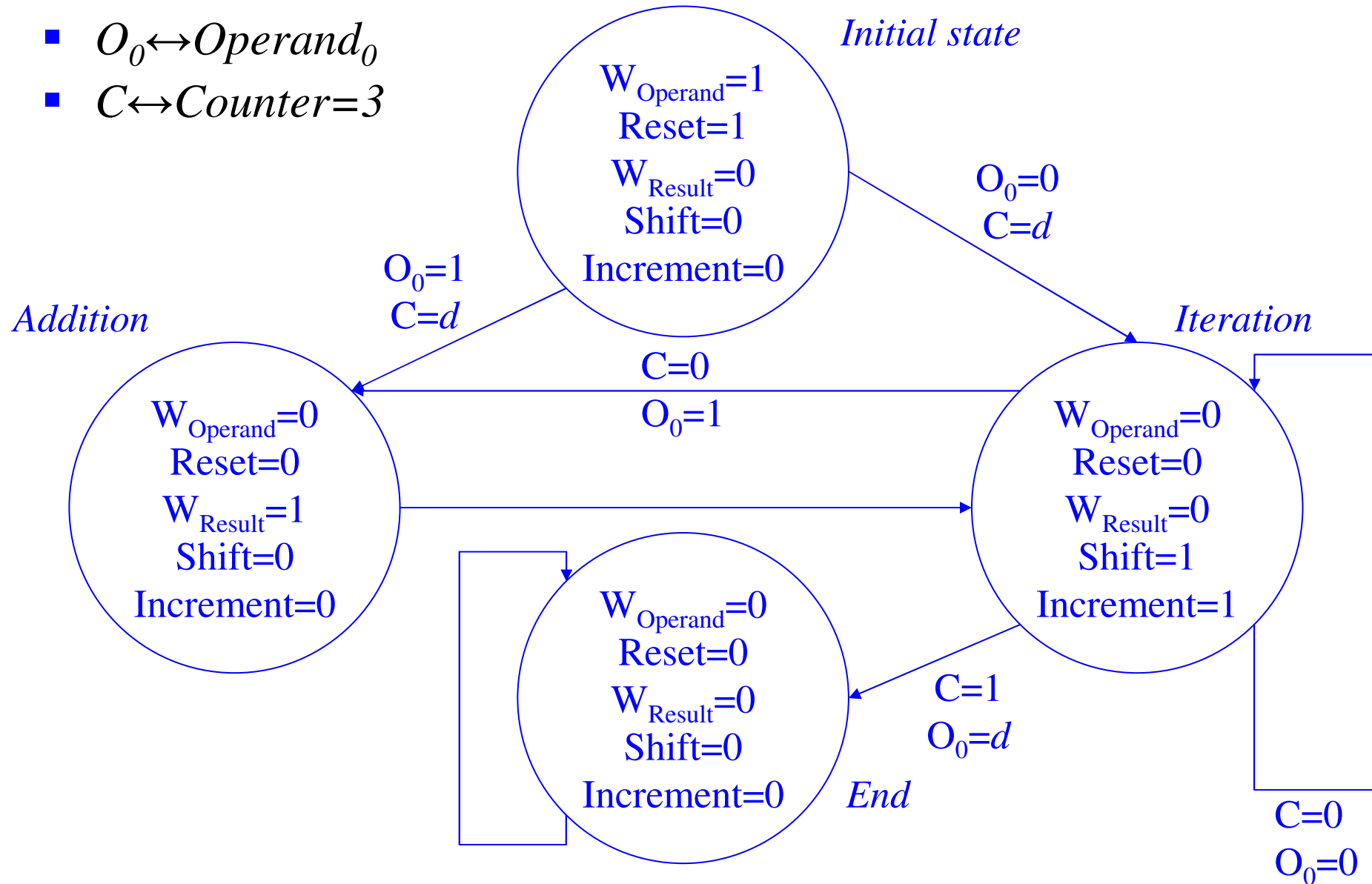
A Systematic Approach for Designing a Sequential Circuit

- Approach: convert a finite-state automaton into a circuit
- Multiplier:
 - 4 states
 - Inputs: $Operand_0$ et $Counter=3$.
 - Outputs: $W_{Operand}$, Reset, W_{Result} , Shift, Increment, End.

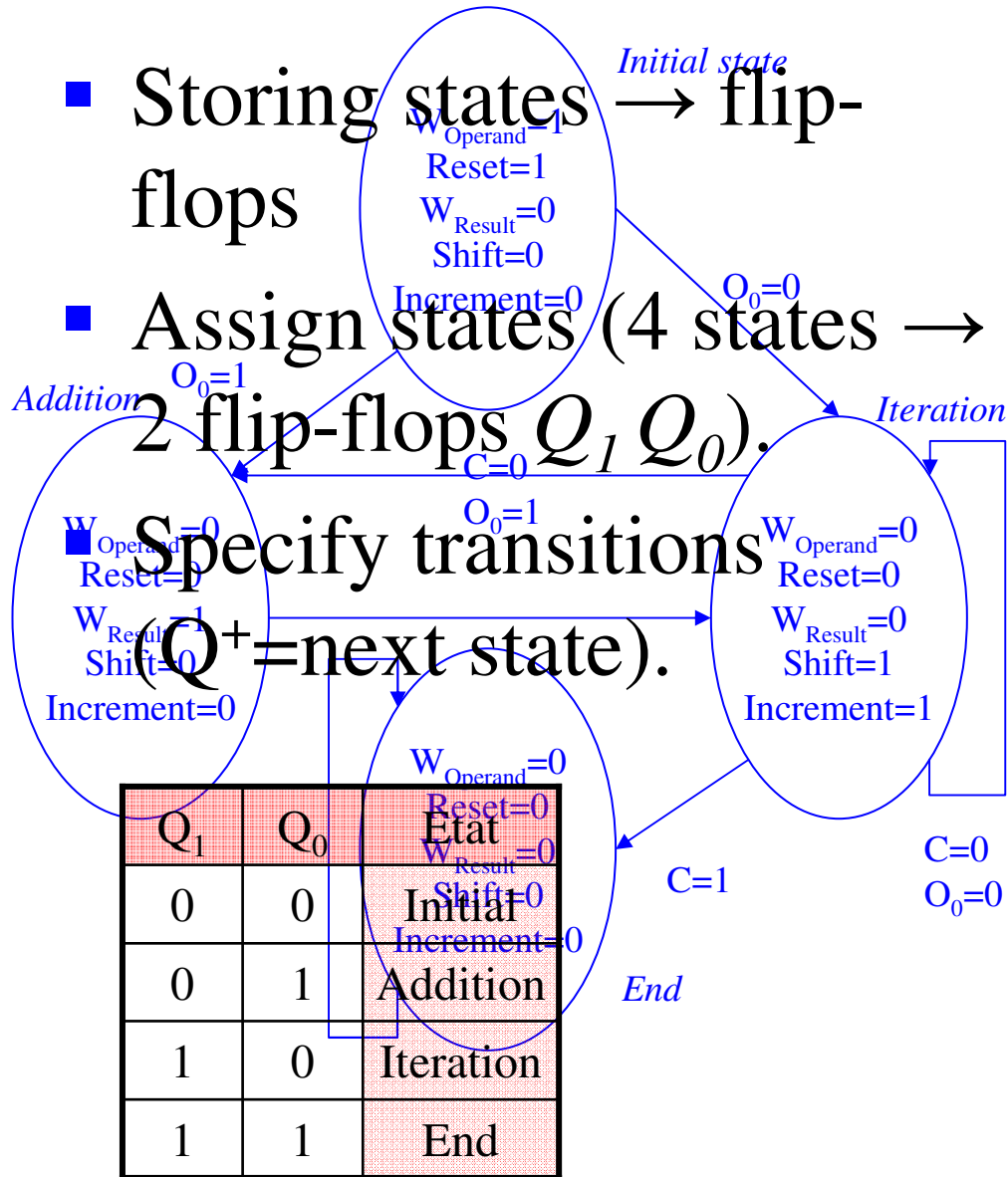


Sequential Circuit Design - Automaton

- $O_0 \leftrightarrow \text{Operand}_0$
- $C \leftrightarrow \text{Counter} = 3$



Sequential Circuit Design – Transition Table



| O_0 | C | Q_1 | Q_0 | Q_1^+ | Q_0^+ | $W_{\text{op.}}$ | Reset | $W_{\text{rés}}$ | Déc. | Inc. | End |
|-------|---|-------|-------|---------|---------|------------------|-------|------------------|------|------|-----|
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

Méthode de Conception d'un Circuit Séquentiel – Table des Transitions

- Mémorisation des états → bascules.
- Assignation des états (4 états → 2 bascules $Q_1 Q_0$).
- Description des transitions (Q^+ =état suivant).

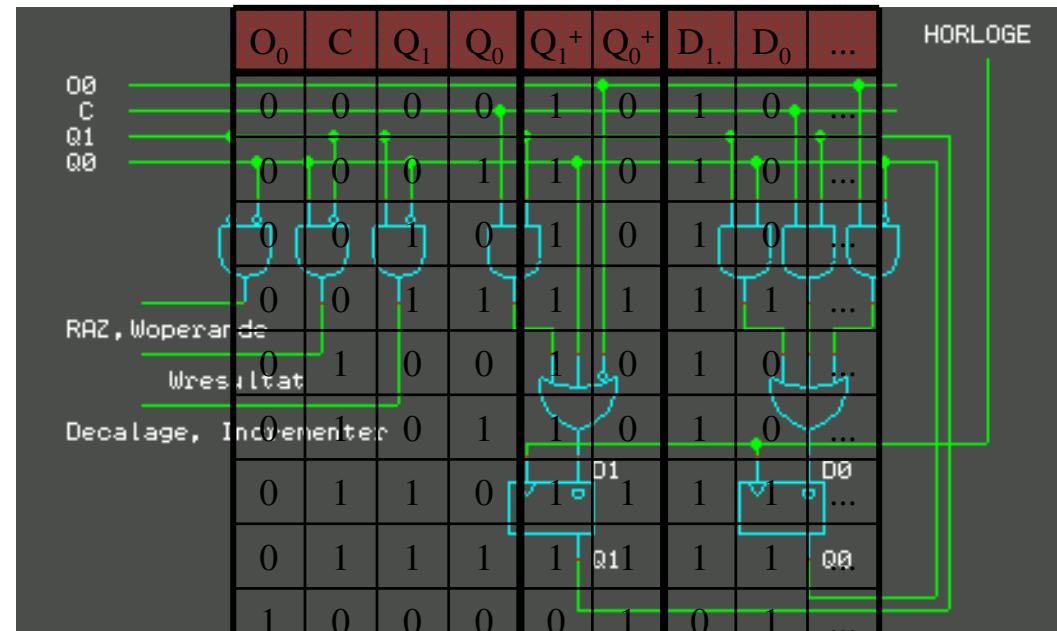
| Q_1 | Q_0 | Etat |
|-------|-------|-----------|
| 0 | 0 | Initial |
| 0 | 1 | Addition |
| 1 | 0 | Iteration |
| 1 | 1 | End |

| Q_0 | C | Q_1 | Q_0 | Q_1^+ | Q_0^+ | $W_{op.}$ | Reset | $W_{rés.}$ | Déc. | Inc. | End |
|-------|---|-------|-------|---------|---------|-----------|-------|------------|------|------|-----|
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

Sequential Circuit Design – Final Circuit

| D | Q | Q ⁺ |
|---|---|----------------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

| Q | Q ⁺ | D |
|---|----------------|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



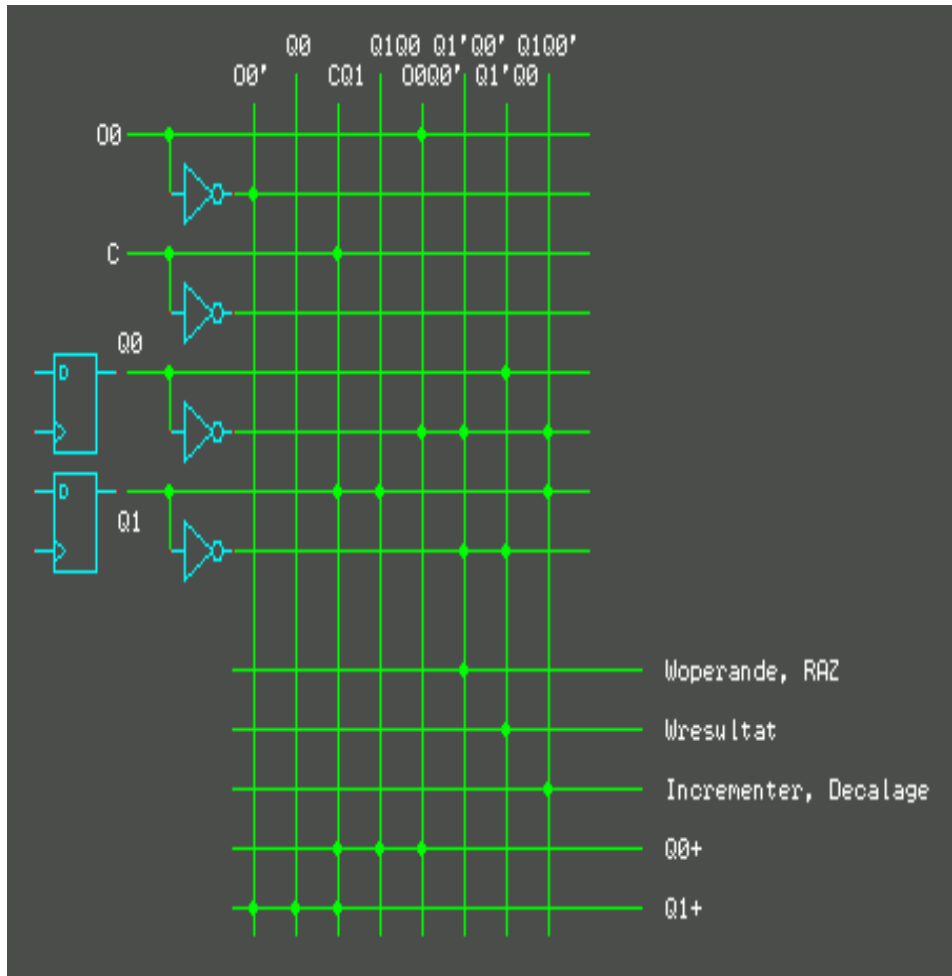
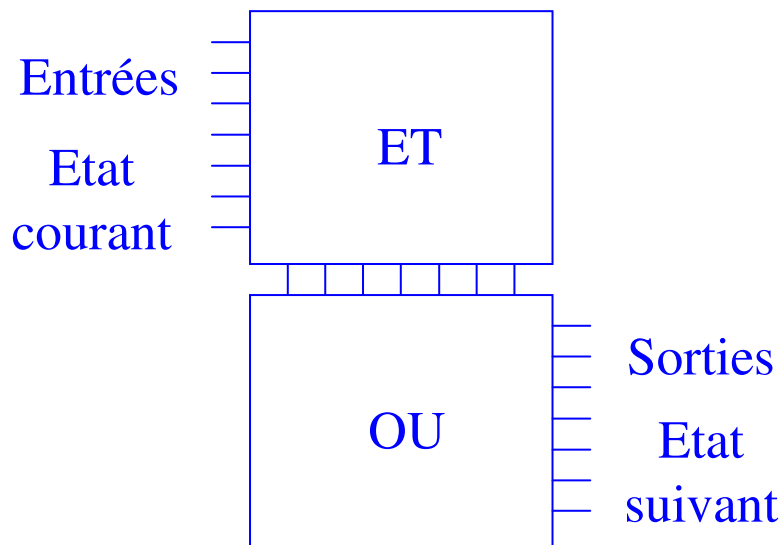
- Using D flip-flops
- Set D_1, D_0 so as to impose desired transitions
- $Q^+ = D \rightarrow$ transition table = truth table of D_1, D_0 .
- Outputs correspond to current states

| | | |
|---|---|--|
| 1 | 0 | $D_0^0 = \overline{Q_0} + Q_0 + C \cdot Q_1 \dots$ |
| 1 | 0 | $D_1^0 = \overline{Q_0} + Q_0 + C \cdot Q_1 \dots$ |
| 1 | 0 | $D_0^1 = \overline{Q_0} \cdot Q_0 + Q_1 \cdot Q_0 + C \cdot Q_1$ |
| 1 | 1 | $W_{operande} = \overline{Reset} = \overline{Q_1 \cdot Q_0}$ |
| 1 | 1 | $W_{res} = \overline{Q_1 \cdot Q_0}$ |

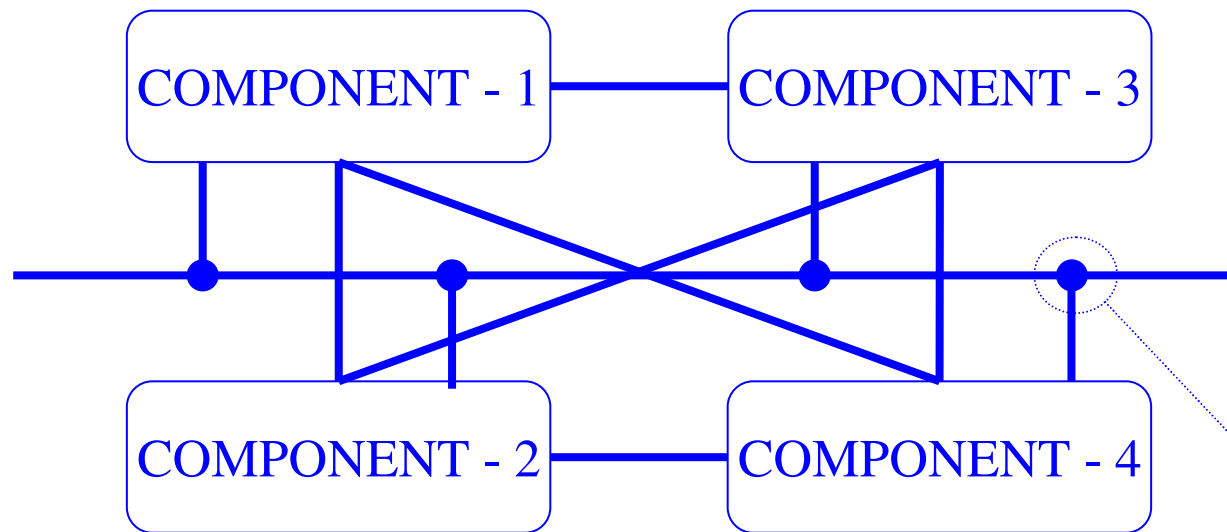
$$Shift = Increment = \overline{Q_1 \cdot Q_0}$$

Génération Rapide du Circuit

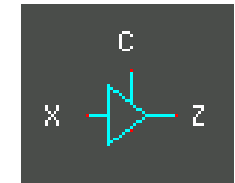
- PLA (*Programmable Logic Array*).
- Grille de connexions; liaisons entre les connexions programmables et réalisées par des diodes (\rightarrow ET et OU).
- Méthode systématique pour représenter des tables de vérité:
 - Somme (OU) de Produits (ET).



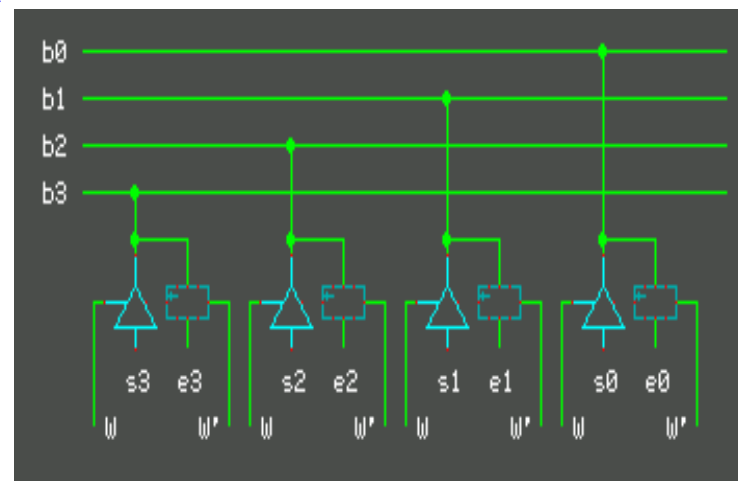
Data Paths - Bus



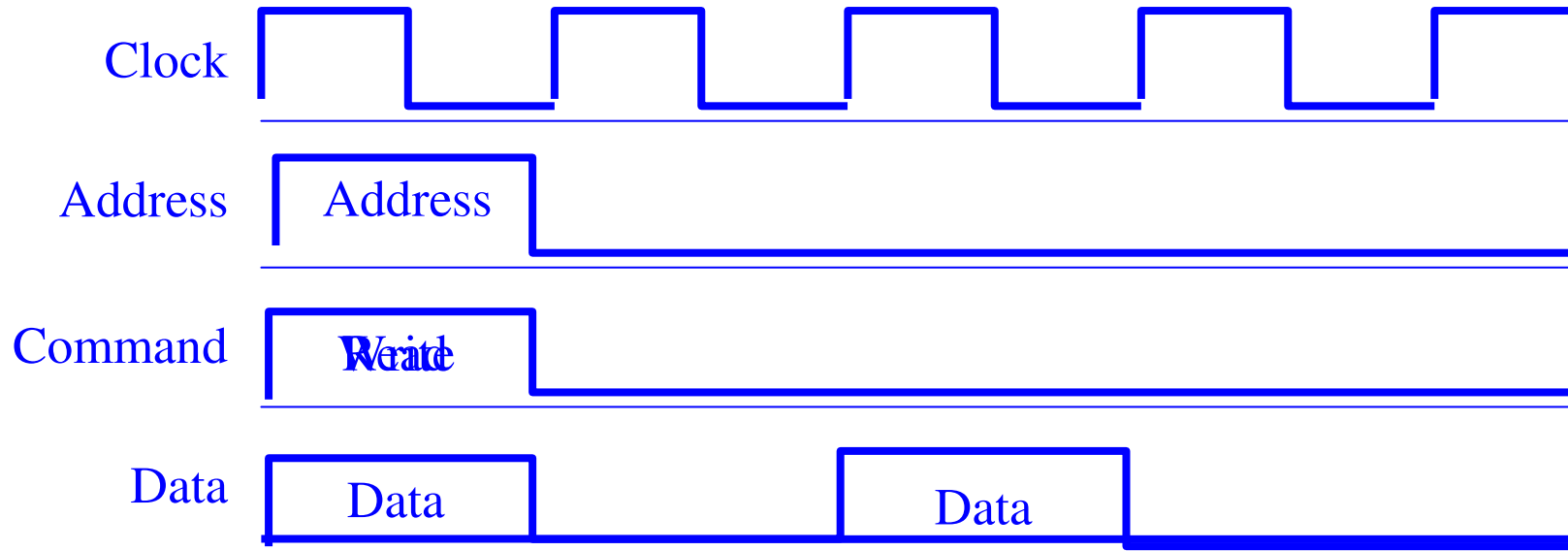
| C | X | Z |
|---|---|------|
| 0 | 0 | open |
| 0 | 1 | open |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



- Connecting a large number of components: **bus**.
- Bus = a set of n 1-bit wires
- Very cheap, but not efficient with large number of components



Bus



- Synchronous buses:
 - All components timed with same clock
 - Links indicate **address, data, command**
 - Internal buses or connection to memory
- Access protocol (priorities,...) implemented in control circuit