

ARCHI – Architecture des ordinateurs

Sylvain Brandel

2024 – 2025

sylvain.brandel@univ-lyon1.fr



CM 8

CIRCUITS SÉQUENTIELS

Partie 2

Compteurs, automates

Registres, RAM

Bascules

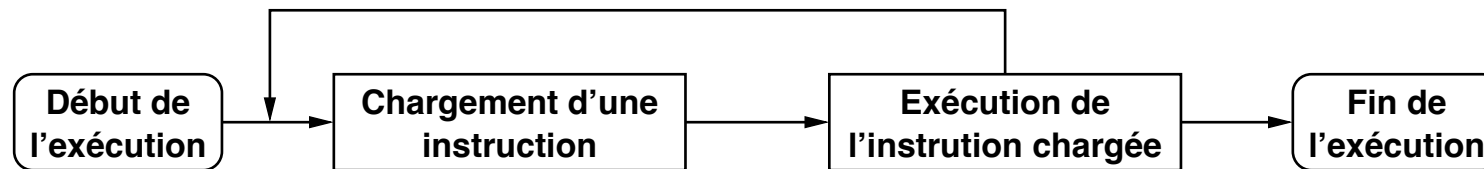
- Bistable
- Bascules transparentes
 - Bascule RS
 - Bascule *Latch* (bascule D transparente)
- Bascules opaques
 - Bascule D
 - Maître esclave
 - À commande par flanc (*Edge triggered*)
 - Bascules dérivées de la bascule D
 - T
 - JK

Compteur

- Opérateur séquentiel à n états
- Cas particulier d'un registre à décalage
 - Interdit avec des bascules transparentes
- État $i \rightarrow \text{état } i + 1 \% n$ à chaque cycle d'horloge
- Ex. Compteur par 2
- Ex. Compteur par 4
- Ex. Compteur par 4 avec remise à zéro

Automates finis

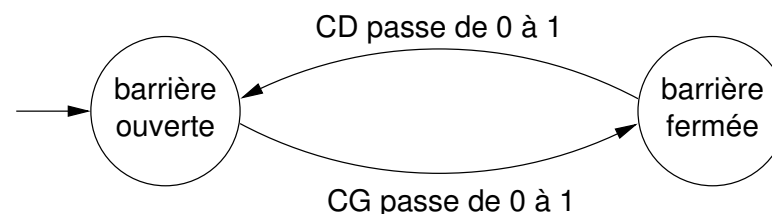
- Ex. Processeur



- Ex. Passage à niveau
 - Si un train approche, les barrières se ferment
 - Lorsque le train s'éloigne, les barrières se rouvrent
- Réaction à des événements → Asynchrone
⇒ Complicé en pratique
- Ici événements synchronisés par l'horloge → Synchrone

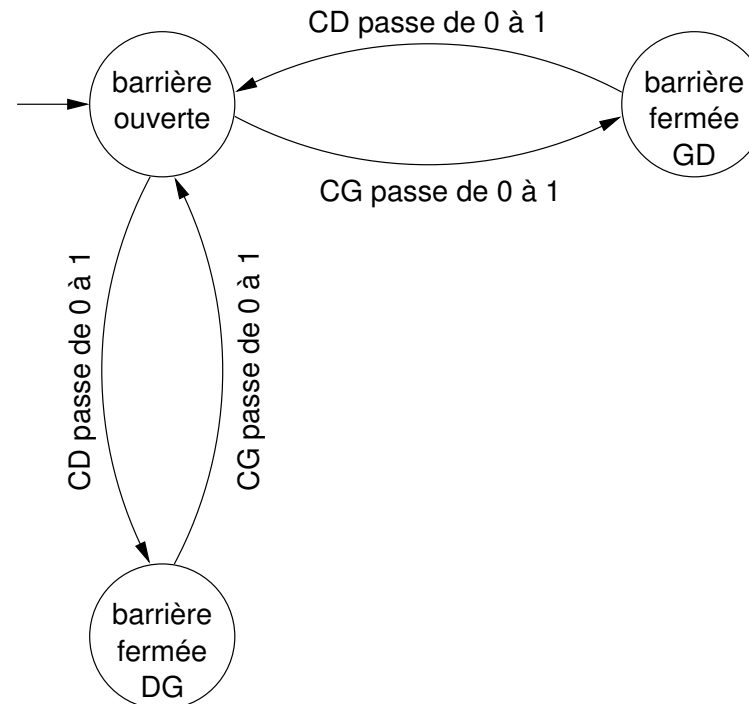
Automates finis

- Un automate à états finis est un quintuplet $(K, \Sigma, \delta, s, F)$ où
 - K : ensemble fini **d'états** (*States*)
 - Σ : ensemble fini de symboles **événements** (*Events*)
 - δ : **fonction de transition** : $K \times \Sigma \rightarrow K$
 - $s \in K$: **état initial**
 - $F \subset K$: **états acceptants** **$F = \emptyset$ ici**
- Ex. Passage à niveau
 - 2 capteurs à 300m CG et CD passent à 1 au passage d'un train
 - Si les trains ne passent que de gauche à droite
 - États : $K = \{\text{barrière ouverte}, \text{barrière fermée}\}$
 - Événements : $\Sigma = \{\text{CG passe de 0 à 1}, \text{CD passe de 0 à 1}\}$
 - État initial : $s = \text{barrière ouverte}$



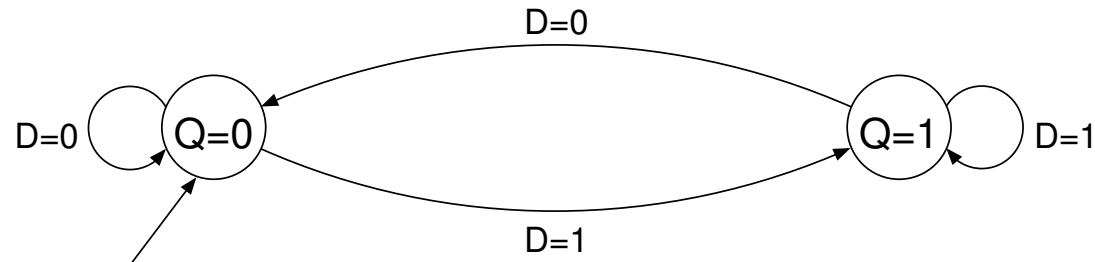
Automates finis

- Ex. Passage à niveau
 - 2 capteurs à 300m CG et CD passent à 1 au passage d'un train
 - Si les trains passent dans les deux sens
 - États : $K = \{\text{barrière ouverte}, \text{barrière fermée GD}, \text{barrière fermée DG}\}$
 - Événements : $\Sigma = \{\text{CG passe de 0 à 1}, \text{CD passe de 0 à 1}\}$
 - État initial : $s = \text{barrière ouverte}$



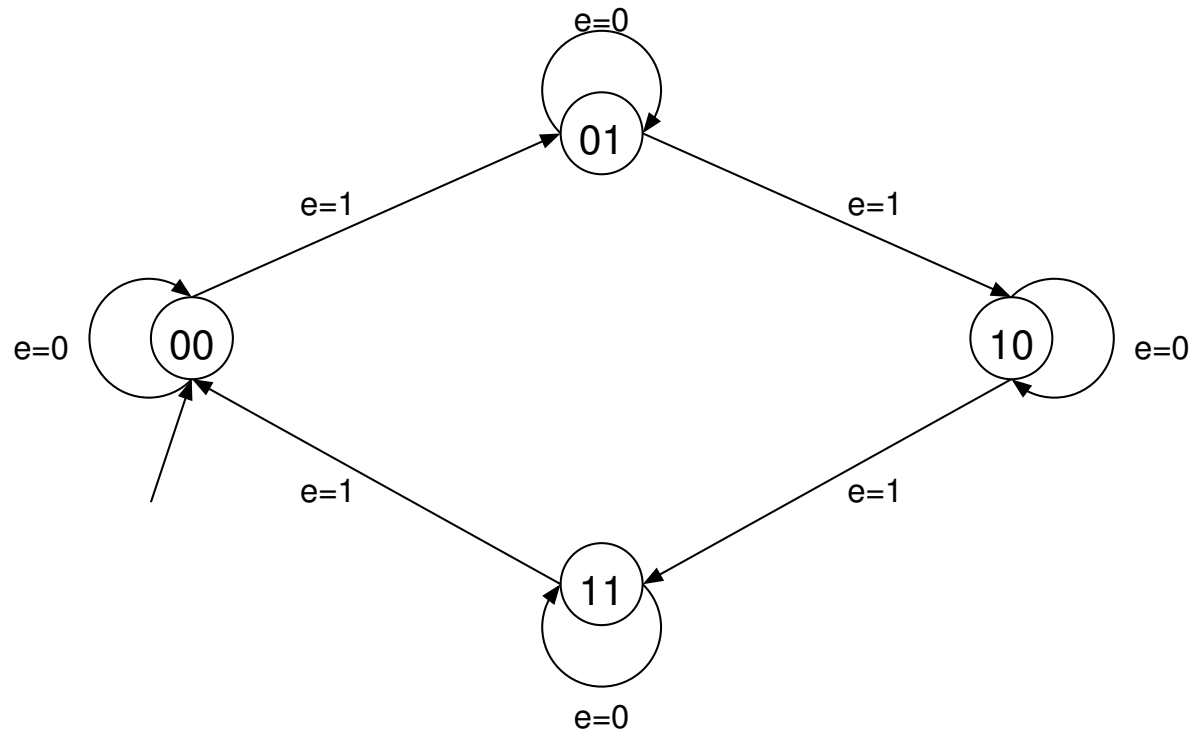
Automates finis

- Automates à états finis **synchrones** :
 - Une transition se produit à chaque **front montant** de l'horloge et est instantanée
 - Les événements sont remplacés par des **entrées** (valeur lors du front montant)
- Ex. Bascule D
 - États : $K = \{Q=0, Q=1\}$
 - Entrées : $\Sigma = \{D=0, D=1\}$



Automates finis

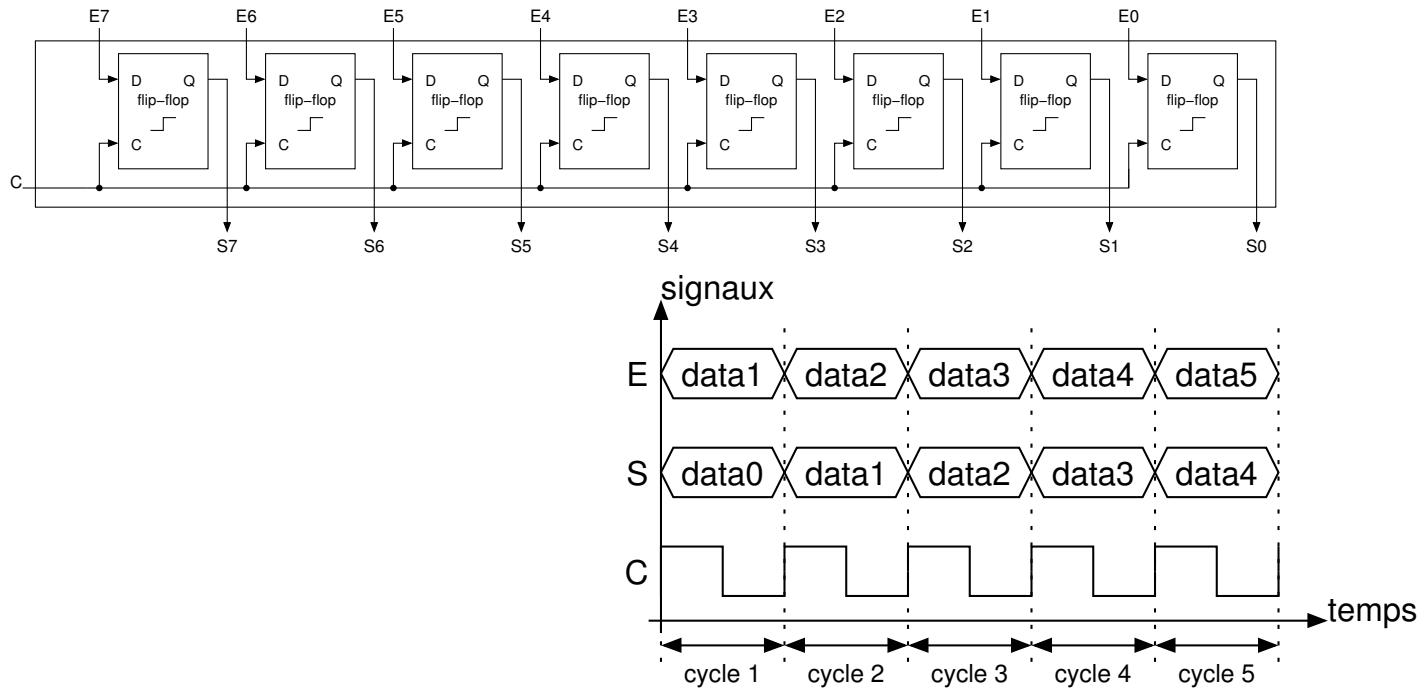
- Ex. Compteur par 4 qui s'incrémente si $e=1$, conserve sa valeur si $e=0$



- Ex. Reconnaissance de la séquence 01

Registres

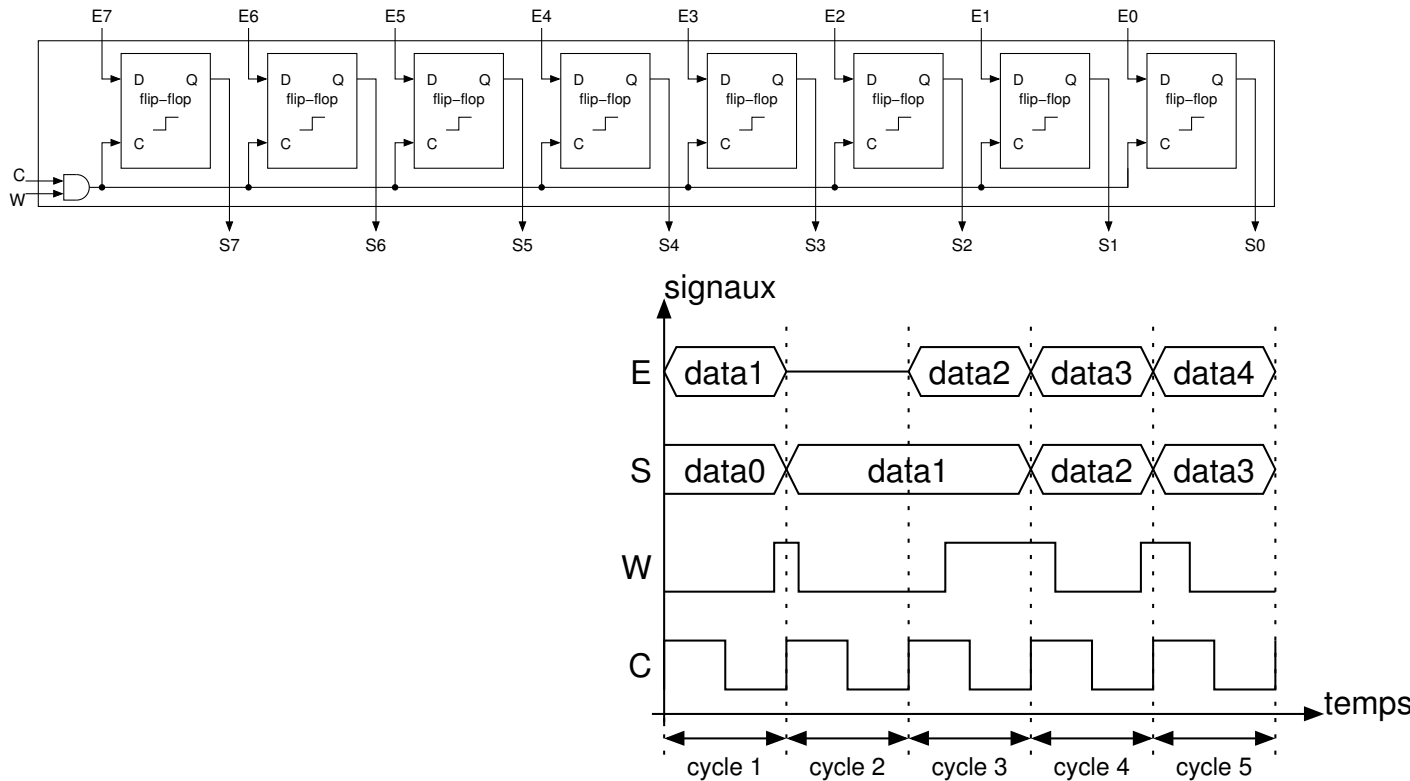
- Ex. Registre 8 bits



- E stocké en fin de cycle et disponible au cycle suivant

Registres

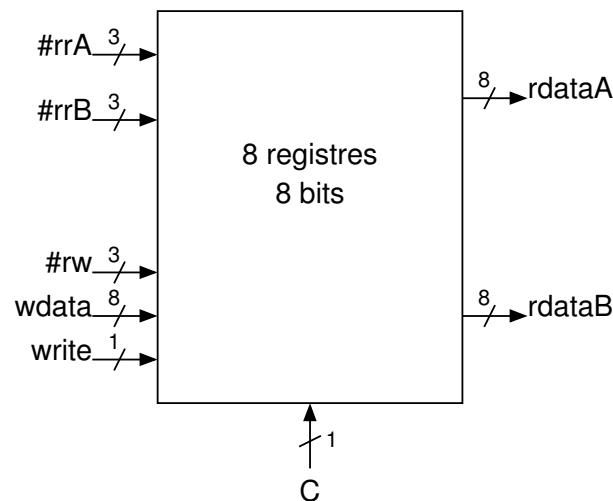
- Ex. Registre 8 bits avec mémorisation sur plusieurs cycles



- En fin de cycle
 - Si $W = 1$ alors E est stocké et disponible au cycle suivant
 - Si $W = 0$ alors S conserve la même donnée au cycle suivant

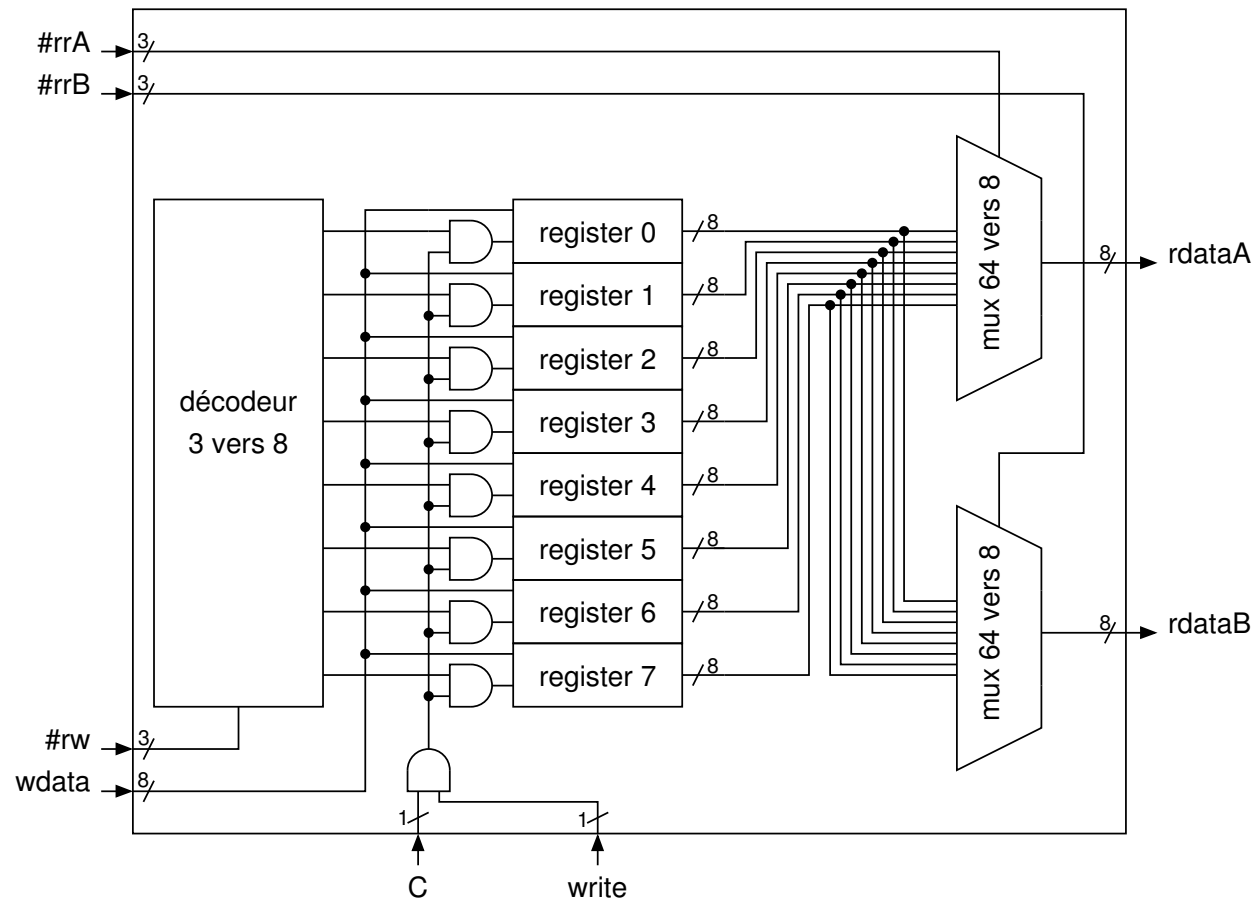
Banc de registres (*register file*)

- Lecture
 - n numéros de registres en entrée $\#rrX$, n ports $rdataX$ en sortie
 - La donnée correspondant à $\#rrX$ est maintenue sur $rdataX$
- Écriture sur le front montant
 - 1 numéro de registre $\#rw$, 1 port $wdata$, 1 horloge C et 1 signal W en entrée
 - Si sur le front montant $W=1$, la donnée $wdata$ est placée dans le registre $\#rw$
- Ex. Banc de 8 registres 8 bits avec deux ports de lecture $n = 2$



Banc de registres (*register file*)

- Ex. Banc de 8 registres 8 bits avec deux ports de lecture $n = 2$

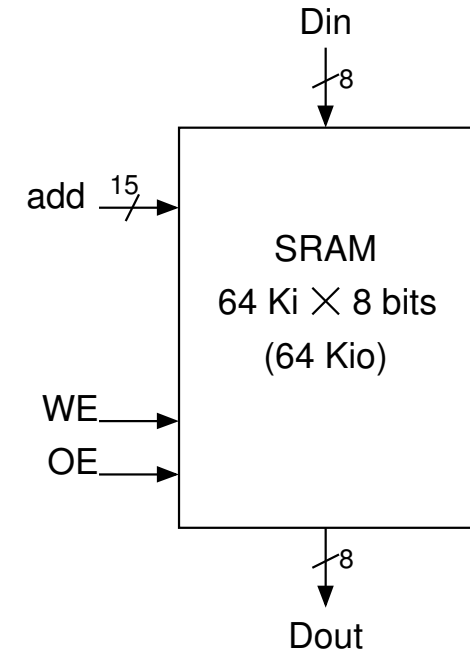


RAM

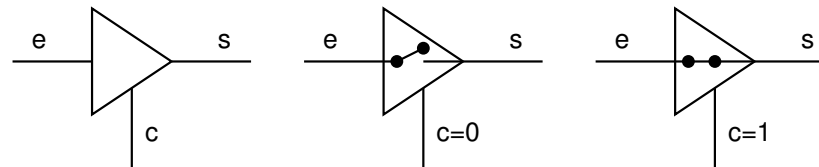
- Bancs de registres : petites mémoires
- **SRAM** (*Static RAM*) : à base de bascules
 - Données stockées tant que le système est sous tension
- **DRAM** (*Dynamic RAM*) : à base de condensateurs
 - Rafrachir la charge des condensateurs
- SRAM **plus rapide** mais **plus chère** → caches (≈ Mio)
- DRAM **moins chère** mais **plus lente** → mémoire centrale (≈ Gio)
- DRAM synchrone (cadencées par une horloge)
 - **SDRAM** (*Synchronous DRAM*) : opérations sur un front d'horloge
 - **DDR SDRAM** (*Double Data Rate SDRAM*) : sur les deux fronts d'horloge

SRAM

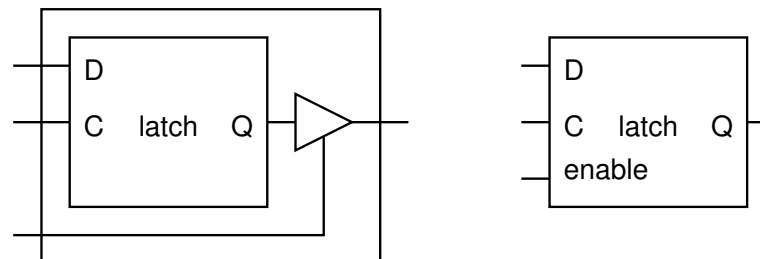
- SRAM de 2^m mots de k bits :
 - Din / Dout (k bits) : entrée / sortie de données
 - add (m bits) : adresse
 - WE (1 bit) : Write Enable
 - OE (1 bit) : Output Enable



- Buffer : interrupteur
 - $c = 1$: e est copié sur s
 - $c = 0$: rien sur s

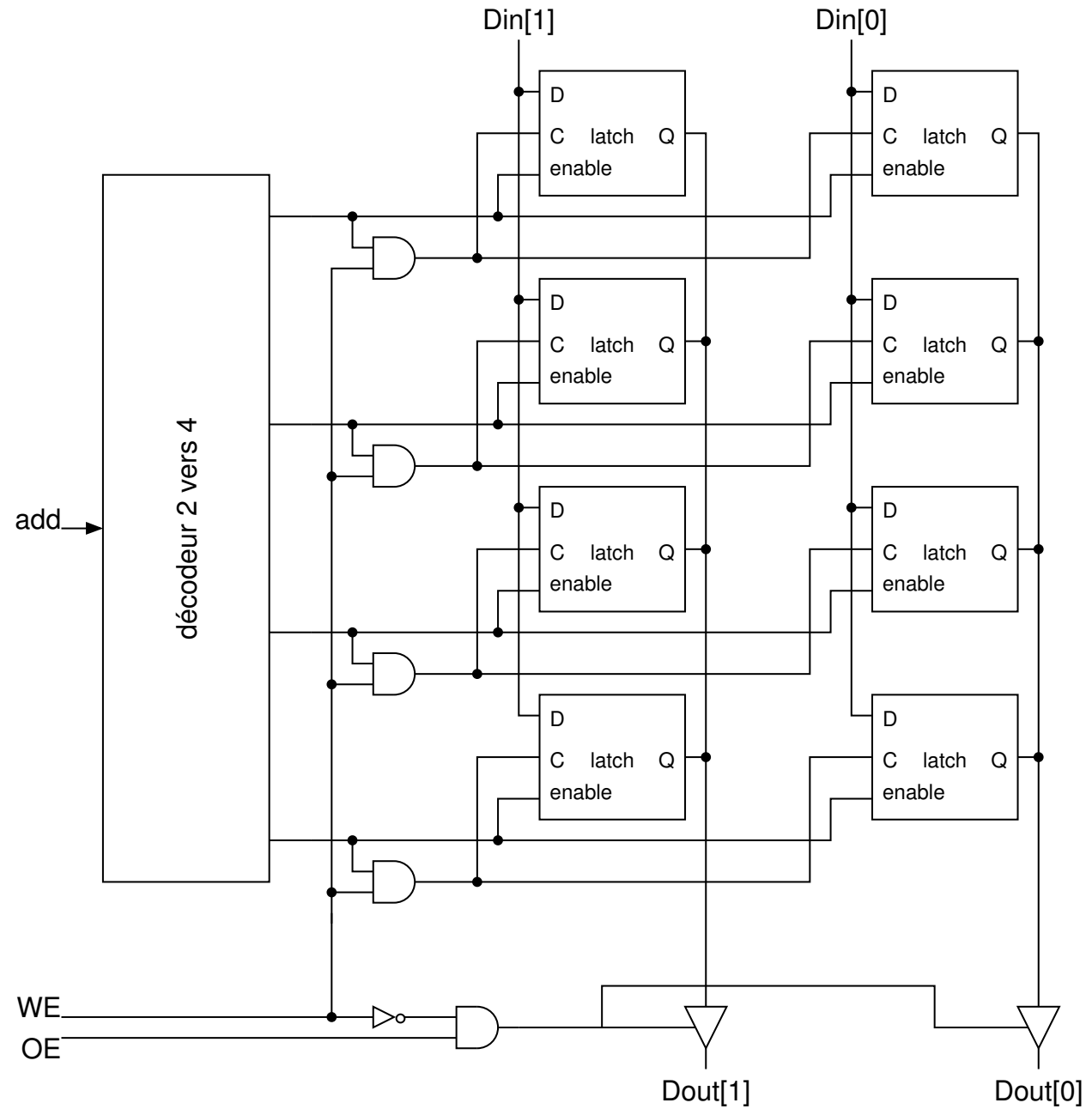


- Bascule D avec buffer



SRAM

- Ex. 4 x 2 bits



SRAM

- Ex. $2^{16} \times 8$ bits ?
- Problème : décodeur 16 vers 2^{16} compliqué à réaliser / câbler
→ Découpage en blocs
- 64Ki mots de 8 bits découpés en 8 blocs de 64Ki bits, chaque bloc stocke un bit par mot adressé
- On découpe la mémoire en blocs de 64Ki = 1024×64 bits
 - Chacun des mots de ces blocs est adressable par $\text{add}[15-6]$
 - Pour chaque mot de 64 bits on sélectionne un bit par $\text{add}[5-0]$

