

# TD Chapitre 03 : Logique Séquentielle

## Objectifs

- Comprendre le rôle de l'horloge
- Maîtriser le comportement de la DFF
- Concevoir des registres avec rétroaction
- Comprendre l'adressage de la RAM

Durée estimée : 1h30

## Exercice 1 : Comprendre l'Horloge

**Objectif :** Distinguer combinatoire et séquentiel

### 1.1 Classification

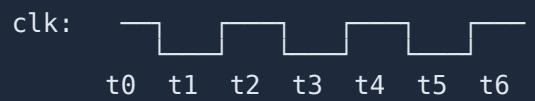
Classez ces circuits en **combinatoire** ou **séquentiel** :

Circuit	Type
Porte AND	?
Additionneur	?
Registre	?
Multiplexeur	?
RAM	?
ALU	?

► Solution

## 1.2 Front montant

Sur ce chronogramme, à quels instants ( $t_1, t_2, t_3, t_4$ ) y a-t-il un front montant ?



► Solution

## Exercice 2 : Comportement de la DFF

**Objectif:** Appliquer la règle  $q(t) = d(t-1)$

### 2.1 Chronogramme

Complétez la sortie  $q$  pour cette DFF :

clk: 

d:  $-[0]-[1]-[1]-[0]-[1]-[0]-$

q:  $-[?]-[?]-[?]-[?]-[?]-[?]-$   
t0 t1 t2 t3 t4 t5

Supposons  $q(t_0) = 0$  (valeur initiale).

► Solution

## 2.2 Question conceptuelle

Pourquoi la DFF est-elle appelée "l'atome de la mémoire" ?

► Solution

## Exercice 3 : Registre 1-bit

**Objectif :** Comprendre la rétroaction

### 3.1 Schéma

Dessinez le schéma d'un registre 1-bit utilisant :

- 1 Mux 2-vers-1
- 1 DFF

Indice : La sortie du DFF revient vers le Mux.

► Solution

### 3.2 Chronogramme du registre

Complétez la sortie **q** pour ce registre 1-bit :

```
clk:  □□□□□  
in:  -[1]-[1]-[0]-[1]-[0]-  
load: -[1]-[0]-[0]-[1]-[0]-  
q:   -[?]-[?]-[?]-[?]-[?]-  
     t0  t1  t2  t3  t4
```

Supposons  $q(t_0) = 0$ .

► Solution

## Exercice 4 : Compteur de Programme (PC)

**Objectif :** Gérer les priorités

### 4.1 Table de priorité

Le PC a 4 modes : reset, load, inc, hold. Complétez :

reset	load	inc	Action
1	x	x	?
0	1	x	?
0	0	1	?
0	0	0	?

► Solution

## 4.2 Simulation du PC

Simulez le PC avec ces signaux. PC initial = 5.

Cycle	reset	load	inc	in	PC après
0	0	0	1	-	?
1	0	0	1	-	?
2	0	1	0	10	?
3	0	0	1	-	?
4	1	0	0	-	?

► Solution

## Exercice 5 : Adressage RAM

**Objectif :** Comprendre l'adressage

### 5.1 RAM8 : Décodage d'adresse

Une RAM8 a 8 mots. L'adresse est sur 3 bits [2:0].

address	Registre sélectionné
000	?
011	?
101	?
111	?

► Solution

## 5.2 RAM64 : Adresse hiérarchique

Une RAM64 est construite avec 8 RAM8.

L'adresse est sur 6 bits [5:0].

Décomposez ces adresses :

address	RAM8	Mot dans RAM8
000101	?	?
011010	?	?
101111	?	?

► Solution

## Exercice 6 : Conception de registre

**Objectif :** Appliquer les concepts

### 6.1 Registre avec enable

Concevez un registre 1-bit avec deux signaux :

- `enable` : Si 0, la sortie est toujours 0
- `load` : Comme avant

Dessinez le schéma.

► Solution

## 6.2 Question conceptuelle

Pourquoi dit-on que les registres sont "plus rapides" que la RAM ?

► Solution

## Exercice 7 : Défi - Registre à décalage

**Objectif :** Étendre le concept

Un **shift register** décale les bits d'une position à chaque cycle.

Exemple sur 4 bits avec entrée série `sin` :

```
Cycle 0: [0][0][0][0], sin=1
Cycle 1: [1][0][0][0], sin=0
Cycle 2: [0][1][0][0], sin=1
Cycle 3: [1][0][1][0], sin=1
```

Comment construire un tel registre avec des DFF ?

► Solution

## Récapitulatif

### Compétences validées

Après ce TD, vous devez savoir :

- [ ] Distinguer combinatoire et séquentiel
- [ ] Appliquer la règle DFF :  $q(t) = d(t-1)$
- [ ] Concevoir un registre avec rétroaction
- [ ] Simuler le comportement du PC
- [ ] Décomposer une adresse RAM hiérarchique

### Prochaine étape

➡ TP Chapitre 03 : Implémenter BitReg, Register, PC et RAM

📘 Référence : Livre Seed, Chapitre 03 - Mémoire