

L2 Math Info – Module S4  
Corrigé Examen CLM – Juin 2013  
TOUS DOCUMENTS AUTORISÉS

**Partie 1 : Bascules D**

Soit le circuit (Figure 1) et les signaux C et D (Figure 2)

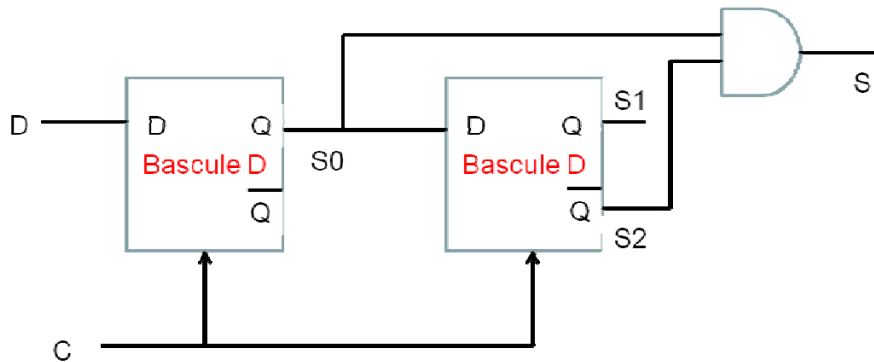


Figure 1 : Deux bascules D plus porte ET

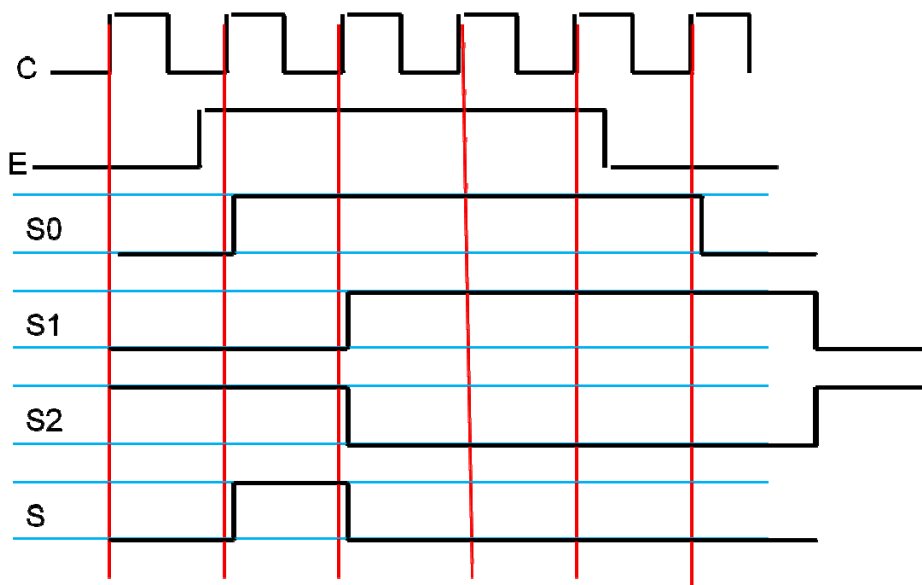


Figure 2 : Signaux appliqués sur le circuit.

Q 1) Dessiner les signaux S0, S1, S2 et S (sans tenir compte des temps de retard entre C et les sorties des bascules).

**Partie 2 : Compteurs**

Q 2) Réaliser un compteur par 8 avec des bascules D avec le codage de Gray (000, 001, 011, 010, 110, 111, 101, 100). On demande l'expression simplifiée des entrées des bascules utilisées en fonction des sorties des bascules

	Q2	Q1	Q0	D2	D1	D0
	0	0	0	0	0	1
	0	0	1	0	1	1
	0	1	1	0	1	0
	0	1	0	1	1	0
	1	1	0	1	1	1
	1	1	1	1	0	1
	1	0	1	1	0	0
	1	0	0	0	0	0

$$D_0 = \overline{Q_2} \cdot \overline{Q_1} + Q_2 \cdot Q_1 = \overline{Q_2 \oplus Q_1}$$

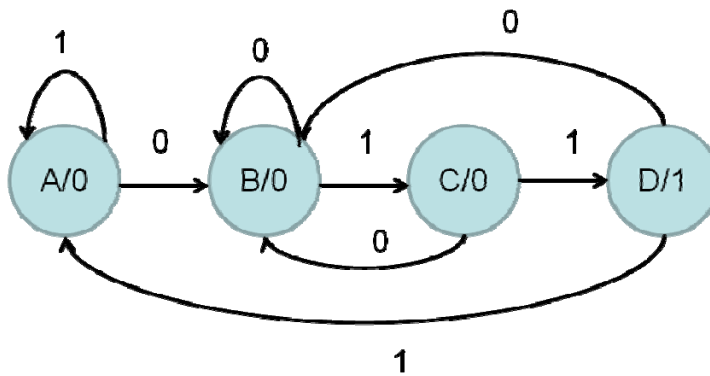
$$D_1 = \overline{Q_2} \cdot Q_0 + Q_1 \cdot \overline{Q_0}$$

$$D_2 = Q_2 \cdot Q_0 + Q_1 \cdot \overline{Q_0}$$

### Partie 3 : Reconnaissance de séquence

**Q 3) Réaliser un automate de Moore avec une entrée E et une sortie S qui reconnaît la séquence 011. On donnera**

- le graphe de transition
- le diagramme de transition
- l'implémentation avec des bascules D (donner les expressions **simplifiées** des entrées des bascules et de la sortie S de l'automate en fonction des sorties des bascules et de l'entrée E de l'automate)
- 



Séquence 011

Entree	Etat présent	Etat futur	Sortie
0	A	B	0
0	B	B	0
0	C	B	0
0	D	A	1
1	A	A	0
1	B	C	0
1	C	D	0
1	D	A	1

En codant A (00) B (01) C (10) et D (11)

	Entrée	Q1	Q0	D1	D0	Sortie
0	0	0	0	0	1	0
1	0	0	1	0	1	0
2	0	1	0	0	1	0
3	0	1	1	0	1	1
4	1	0	0	0	0	0
5	1	0	1	1	0	0
6	1	1	0	1	1	0
7	1	1	1	0	0	1

$$S = Q1Q0$$

$$D1 = \sum m(5,6) = E(Q1 \oplus Q0)$$

$$D0 = \sum m(0,1,2,3,6) = \bar{E} + Q1 \cdot \overline{Q0}$$

#### **Partie 4 : Programmation MIPS**

On considère les instructions MIPS suivantes :

SLT rd, rs, rt // rd ← 1 si rs < rt (signés) et 0 sinon  
 BNE rs, rt, offset // branchement si rs ≠ rt  
 BEQ rs, rt, offset // branchement si rs=rt  
 BGTZ rs, offset // branchement si rs > 0  
 BGEZ rs, offset // branchement si rs ≥ 0  
 BLTZ rs, offset // branchement si rs < 0  
 BLTEZ rs, offset // branchement si rs ≤ 0

**Q 4) Donner la suite des instructions MIPS permettant d'exécuter les pseudo-instructions MIPS suivantes**

- BGT r2,r3, cible // branchement si r2 > r3  
   SLT r1,r3,r2 // r1 = 1 si r3 < r2  
   BNE r1,r0, cible ou BGTZ r1, cible
- BGE r4,r5, cible // branchement si r4 ≥ r5  
   SLT r1, r4, r5 // r1 = 1 si r4 < r5 et 0 si r4 ≥ r5  
   BEQ r1,r0,cible ou BLTEZ r1, cible
- BLT r6,r7, cible // branchement si r6 < r7  
   SLT r1,r6,r7  
   BNE r1,r0, cible ou BGTZ r1, cible
- BLE r8,r9, cible // branchement si r8 ≤ r9  
   SLT r1,r9,r8 // r1 = 1 si r9 < r8 et 0 si r8 ≤ r9  
   BEQ r1,r0, cible ou BLTEZ r1, cible

**Q 5) Que fait le programme suivant ? (Donner le programme C correspondant)**

```
LUI R1, 0x 5000
ADDI R3,R0,0
ADDI R4,R0, 512
Boucle: LB R2, 0(R1)
        ADDI R1,R1,1
        BGTZ R2, Suite
        ADDI R3,R3,1
Suite :  ADDI R4,R4,-1
        BNE R4,R0, Boucle
```

Calcule le nombre de valeurs négatives ou nulles dans un tableau de 512 octets signés.

```
Char T[512], N=0 ;
For (i=0 ; i<512 ; i++)
    If (T[i] ≤ 0) N++;
```

**Q 6) Modifier le programme de la question 5 pour diminuer le nombre d'instructions utilisées.**

```
LUI R1, 0x 5000
ADDI R3,R0,0
ADDI R4,R1, 512
Boucle: LB R2, 0(R1)
        ADDI R1,R1,1
        BGTZ R2, Suite
        ADDI R3,R3,1
Suite :  BNE R1,R4, Boucle
```

**Partie 5 : Microarchitecture et temps d'exécution**

Soit le programme assembleur MIPS ci-dessous

```
LUI R1, F000H
ADDI R2, R1, 400
ADD R3,R0,R0
Boucle : LW R4,(R1)
        LW R5, 400 (R1)
        ADD R4,R4,R5
        ADD R3,R3,R4
        ADDI R1,R1,4
        BNE R1,R2, Boucle
```

Les instructions s'exécutent sur le processeur non pipeliné dont le chemin de données est présenté en Figure 3.

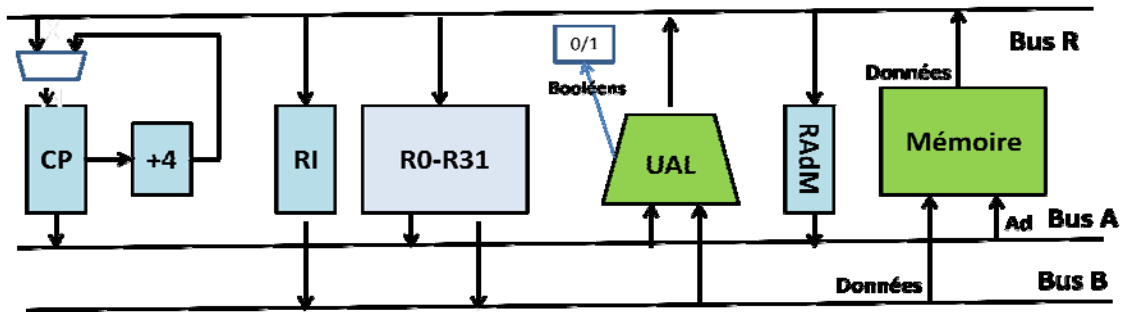


Figure 3 : Chemin de données non pipeliné

**Q 7) Donner le temps d'exécution d'une itération de la boucle en nombre de cycles d'horloge**

Type d'instructions	Nombre de cycles
Arithmétique et logique R	2
Arithmétique et logique I	2
Mémoire	3
Branchement conditionnel	3

3 instructions à 3 cycles et 3 instructions à 2 cycles

Total : 15 cycles

**Q 8) Donner le nombre d'itérations de la boucle et le temps d'exécution total du programme en nombre de cycles d'horloge. (L'instruction LUI prend 2 cycles)**

Il y a 100 itérations de la boucle.

Temps total d'exécution =  $15 * 100 + 6 = 1506$  cycles

**Q 9) Quel serait le temps d'exécution du programme ci-dessous**

```

LUI R1, F000H
ADDI R2, R1, 100
ADD R3,R0,R0
Boucle : LB R4,(R1)
          LB R5, 100 (R1)
          ADD R4,R4,R5
          ADD R3,R3,R4
          ADDI R1,R1,4
          BNE R1,R2, Boucle
    
```

LB et LW ont le même temps d'exécution. 15 cycles/itération.

Il y a 25 itérations.

Temps total d'exécution =  $15 * 25 + 6 = 375 + 6 = 381$  cycles.