

L2 Math Info – Module S4
Examen CLM – Mai 2013
Durée : 2H
TOUS DOCUMENTS AUTORISÉS

Partie 1 : Bascules D

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

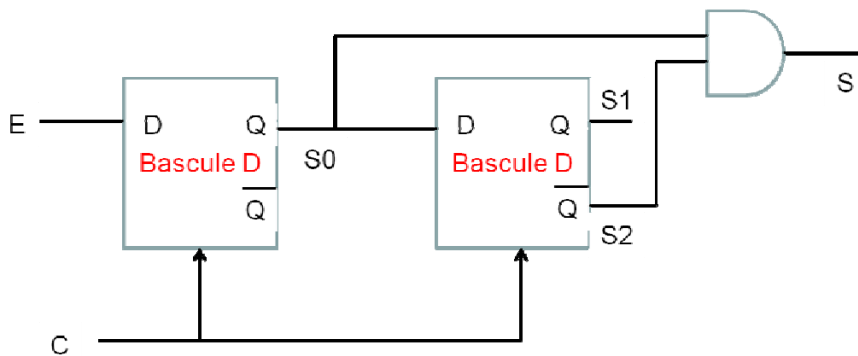


Figure 1 : Deux bascules D plus porte ET

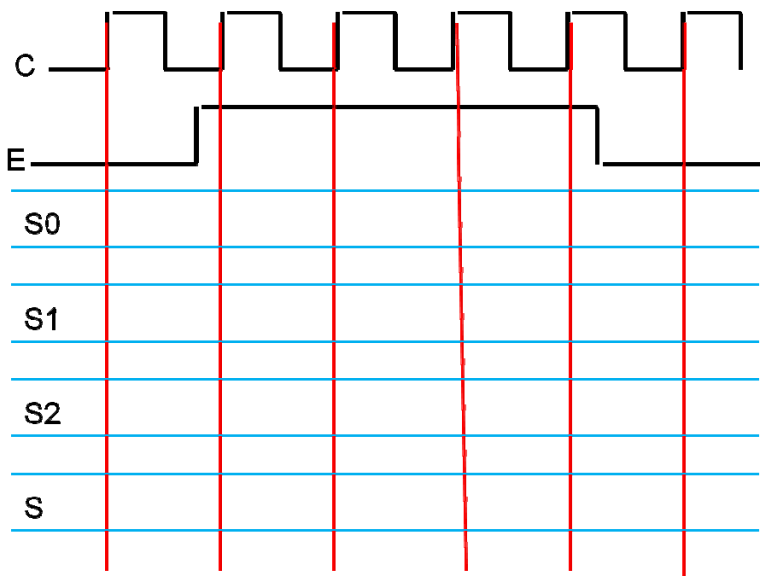


Figure 2 : Signaux appliqués sur le circuit.

Q 1) Dessiner les signaux S0, S1, S2 et S.

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

Partie 3 : Reconnaissance de séquence

Q 3) On veut réaliser un automate de Moore qui détecte chaque occurrence de la séquence 011 dans une séquence de bits. L'automate a une entrée E qui reçoit la

séquence bit à bit, et une sortie S, qui est à 1 à chaque apparition de la séquence 001. 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)

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
```

On rappelle que le registre r0 est câblé à 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 (signé)
- BGE r4,r5, cible // branchement si r4 ≥ r5 (signé)
- BLT r6,r7, cible // branchement si r6 < r7 (signé)
- BLE r8,r9, cible // branchement si r8 ≤ r9 (signé)

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

```
LUI R1, 0x5000
ADDI R3,R0,0
ADDI R4,R0,512
Loop: LB R2, 0(R1)
      ADDI R1,R1,1
      BGTZ R2, Suite
      ADDI R3,R3,1
Suite: ADDI R4,R4,-1
      BNE R4,R0,Loop
```

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

Partie 5 : Microarchitecture et temps d'exécution

Soit le programme assembleur MIPS ci-dessous

```

LUI R1, 0xF000
ADDI R2, R1, 400
ADD R3, R0, R0
Loop : LW R4, 0(R1)
      LW R5, 400 (R1)
      ADD R4, R4, R5
      ADD R3, R3, R4
      ADDI R1, R1, 4
      BNE R1, R2, Loop
    
```

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

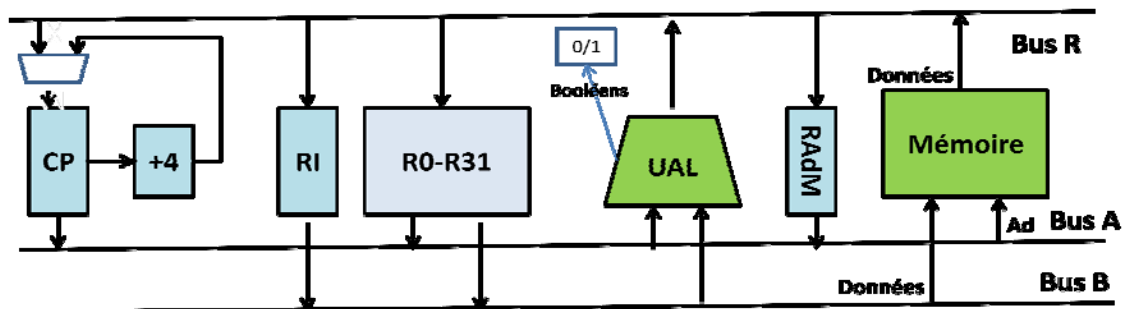


Figure 3 : Chemin de données non pipeliné

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

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)

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

```

LUI R1, 0xF000
ADDI R2, R1, 100
ADD R3, R0, R0
Loop : LB R4, 0(R1)
      LB R5, 100 (R1)
      ADD R4, R4, R5
      ADD R3, R3, R4
      ADDI R1, R1, 4
      BNE R1, R2, Loop
    
```