

Licence Math-Info –S4
Corrigé Examen CLM – Juin 2012

TOUS DOCUMENTS AUTORISES

Partie 1 : Compteurs

Q 1) Réaliser un compteur par 3 avec des bascules D. On demande de donner les expressions simplifiées (sous forme OU de ET) des entrées D des bascules en fonction des sorties Q des bascules.

Solution 1

N	Q1	Q0	D1	D0
0	0	0	0	1
1	0	1	1	0
2	1	0	0	0

$$D1 = Q0$$

$$D0 = \overline{Q1} \cdot \overline{Q0}$$

Solution 2

N	Q1	Q0	D1	D0
0	0	0	0	1
1	0	1	1	1
2	1	1	0	0

$$D0 = \overline{Q1}$$

$$D1 = \overline{Q1} \cdot Q0$$

Q 2) On dispose d'un compteur par 16 avec une entrée de contrôle CH, quatre entrées de chargement $L_3L_2L_1L_0$ et quatre sorties $Q_3Q_2Q_1Q_0$. Lorsque $CH=0$, le compteur fonctionne comme un compteur sur une transition montante de l'horloge et lorsque $CH=1$, le compteur charge le contenu des entrées $L_3L_2L_1L_0$.

Donner le schéma logique pour réaliser un compteur par 11/12, qui compte par 11 lorsqu'une entrée $E = 1$ et par 12 lorsque l'entrée $E = 0$. On utilisera le compteur par 16 et des portes de base ET, OU et inverseurs.

Attention : un compteur par 11 compte de 0 à 10 et un compteur par 12 compte de 0 à 11.

Lorsque $E=1$, il faut charger 0000 lorsque $Q_3Q_2Q_1Q_0 = 1010$ (et éventuellement 1011)

Lorsque $E=0$, il faut charger 0000 lorsque $Q_3Q_2Q_1Q_0 = 1011$

Les états 12 à 15 sont indifférents

On a donc

$$CH = Q3.Q1.\overline{Q0}.E + Q3.Q1.Q0.\overline{E}$$

(Cette solution ne considère pas le cas où le passage de E=1 à E=0 intervient lorsque le compteur est dans l'état 1011).

$$L3L2L1L0 = 0000$$

Partie 2 : Analyse d'automate

Q 3) Que fait l'automate de la Figure 1 ?

- Donner les équations de S, D1 et D0 en fonction de l'entrée X et de Q1 et Q0
- Donner le diagramme de transition et le graphe de transition.
- Est-il de type Moore ou Mealy ?

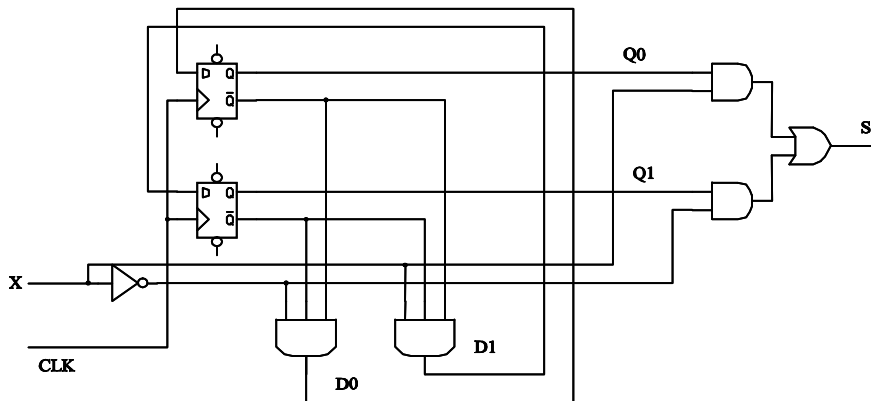
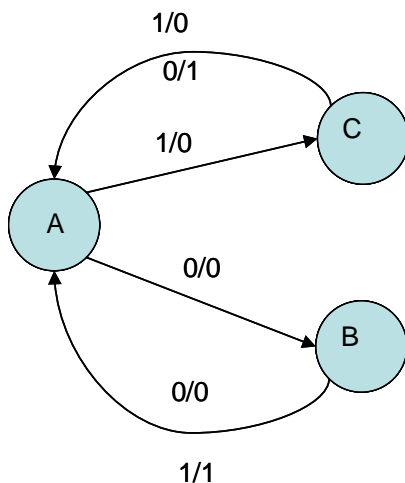


Figure 1 : Automate

$$D_0 = \overline{X} \cdot \overline{Q_1} \cdot \overline{Q_0}$$

$$D_1 = X \cdot \overline{Q_1} \cdot \overline{Q_0}$$

$$S = \overline{X} \cdot Q_1 + X \cdot Q_0$$



NB : le diagramme ci-dessus suppose que l'on part de l'état A ($Q1=0$ et $Q0 = 0$). Dans ce cas, on n'atteint jamais l'état ($Q1=1$ et $Q0=1$).

C'est un automate de Mealy (la sortie est fonction de l'état et de l'entrée X).

L'automate reconnaît une séquence (0,1) ou (1,0) sans recouvrement.

Partie 4 : INSTRUCTIONS NIOS

On considère que les registres du processeur contiennent les huit chiffres hexadécimaux suivants :

R0	0000 0000
R1	1234 5678
R2	FFFF 0000
R3	ABCD EF01
R4	FFFF FFFF
R5	8765 4321

Q 4) Donner le contenu des registres R6 à R11 (sous forme de huit chiffres hexadécimaux) après exécution des instructions suivantes.

- a) ADD R6, R2, R1 ; R6 = 12335678
- b) SUB R7, R1, R5; R7 = 8ACF1357
- c) SLLI R8,R5, 4; R8= 765 43210
- d) SRAI R9, R5,8; R9 = FF876543
- e) SRLI R10, R2, 1; R10 = 7FFF8000
- f) OR R12, R1, R3; R12 = BBFD FF79
- g) AND R13, R1, R3 ; R13 = 0204 4600

Q 5) Donner le contenu (sous forme de huit chiffres hexadécimaux) du registre CP après exécution des instructions suivantes en supposant à chaque fois que l'adresse de l'instruction est : 1000 0000_H.

- a) BGT R4,R5, +8 ; R4>R5 donc pris CP = NCP+8 = 1000 000C_H
- b) BGEU R4,R2,+12 ; R4>R2 donc pris CP =NCP+12 = 1000 0010_H
- c) BNE R0,R0, +4 ; condition fausse CP = NPC= 1000 0004_H.

Partie 5 : Programmation assembleur NIOS

Q 6) Ecrire un programme assembleur NIOS qui met à zéro toutes les cases mémoire entre les adresses 0x1000 0000 et 0x1000 00FF

```
ORHI R1,R0, 1000H
ORI R2,R1, 0100H // R2 contient l'adresse 1000 0100H
LOOP : STW R0, 0(R1)
      ADDI R1,R1,4
      BLT R1,R2, LOOP
```

Q 7) Donner le programme C correspondant au programme assembleur ci-dessous (Les adresses des entiers A et B sont aux adresses 0x1000 0000 et 0x1000 0004. Le résultat S est rangé à l'adresse 0x1000 0004). Quelle est la valeur finale de S si A=15 et B=27 ?

```
ORHI R1,R0,1000H
LDW R2,0(R1)
LDW R3,4(R1)
LOOP : BEQ R2,R3,FIN
      BGT R3,R2, SUITE
```

```

SUB R2,R2,R3
BEQ R0,R0, LOOP
SUIITE : SUB R3,R3,R2
        BEQ R0,R0, LOOP
FIN :   STW R2, 8(R1)
    
```

```

Int A,B,S ;
While (A !=B){
    If (A>B) A=A-B
    Else B=B-A ; }
S=A;
    
```

Valeur finale de S=3 (PGCD de 15 et 27)

Partie 6 : Microarchitecture et temps d'exécution

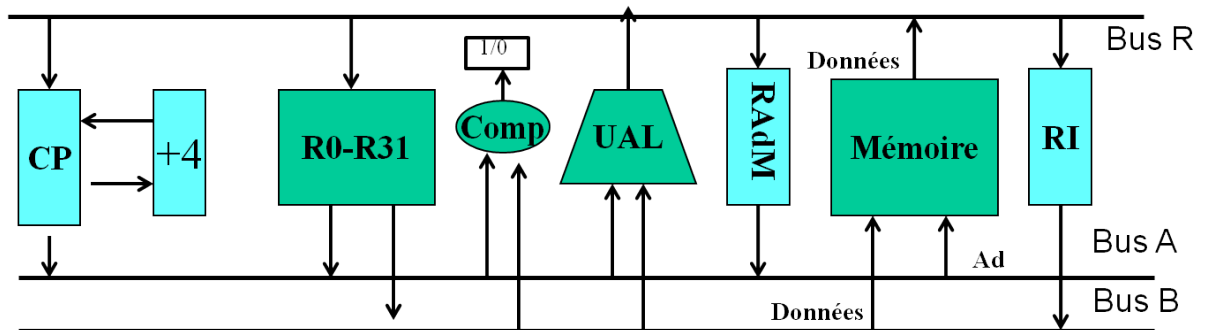


Figure 2 : Chemin de données d'une architecture non pipelinée

On suppose que les instructions NIOS s'exécutent sur le chemin de données de la Figure 2.

Soit la boucle suivante :

```

LOOP : LDW R4, 0(R1)
        LDW R5, 128(R1)
        ADD R3, R4, R5
        ADDI R1, R1, 4
        BLT R1, R2, LOOP
    
```

Q8) Donner en nombre de cycles d'horloge le temps d'exécution d'une itération de la boucle.

LDW et BGT prennent 3 cycles
ADD et ADDI prennent 2 cycles

Temps d'exécution d'une itération de boucle : 9 cycles + 4 cycles = 13 cycles.

Dans le chemin de données de la figure 2, on supprime l'additionneur +4 et on ajoute à l'UAL une commande permettant d'ajouter 4 à au contenu de l'entrée reliée au Bus A (l'entrée connectée au bus B n'étant plus utilisée) : l'UAL peut maintenant effectuer l'opération « contenu du Bus A + 4 »

Q9) Donner en nombre de cycles d'horloge le nouveau temps d'exécution d'une itération de la boucle.

Le tronc commun prend maintenant deux cycles au lieu d'1.

Temps d'exécution d'une itération de la boucle

$13+5 = 18$ cycles