

L2 Math Info – Module S4 Corrigé Examen CLM – Juin 2012

TOUS DOCUMENTS AUTORISES - CALCULETTES INTERDITES

Partie 1 : Logique combinatoire

Soit un circuit combinatoire ayant 4 entrées $e_3e_2e_1e_0$ et 4 sorties $s_3s_2s_1s_0$. Les quatre bits en entrées correspondent à un nombre binaire en représentation complément à 2 (entre -8 et $+7$). Le circuit fournit en sortie le complément à 2 du nombre présent en entrée.

Q1) Donner les expressions logiques simplifiées des sorties en fonction des entrées.

	E_3	E_2	E_1	E_0	S_3	S_2	S_1	S_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	1	1
2	0	0	1	0	1	1	1	0
3	0	0	1	1	1	1	0	1
4	0	1	0	0	1	1	0	0
5	0	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	0
7	0	1	1	1	1	0	0	1
-8	1	0	0	0	1	0	0	0
-7	1	0	0	1	0	1	1	1
-6	1	0	1	0	0	1	1	0
-5	1	0	1	1	0	1	0	1
-4	1	1	0	0	0	1	0	0
-3	1	1	0	1	0	0	1	1
-2	1	1	1	0	0	0	1	0
-1	1	1	1	1	0	0	0	1

$$S_0 = E_0$$

$$S_1 = E_0 \oplus E_1$$

$$S_2 = E_2 / (E_1 + E_0) + E_2 (E_1 \cdot E_0) = (E_1 + E_0) \oplus E_2$$

$$S_3 = E_3 / (E_2 + E_1 + E_0) + E_3 (E_2 \cdot E_1 \cdot E_0) = (E_2 + E_1 + E_0) \oplus E_3$$

Partie 2 : Compteurs

Q2) Réaliser un compteur par 6 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.

	Q2	Q1	Q0	D2	D1	D0
0	0	0	0	0	0	1
1	0	0	1	0	1	0
2	0	1	0	0	1	1
3	0	1	1	1	0	0

4	1	0	0	1	0	1
5	1	0	1	0	0	0

$$D2 = \sum m(3,4)$$

$$D1 = \sum m(1,2)$$

$$D0 = \sum m(0,2,4)$$

$$D2 = Q2Q0/ + Q1Q0$$

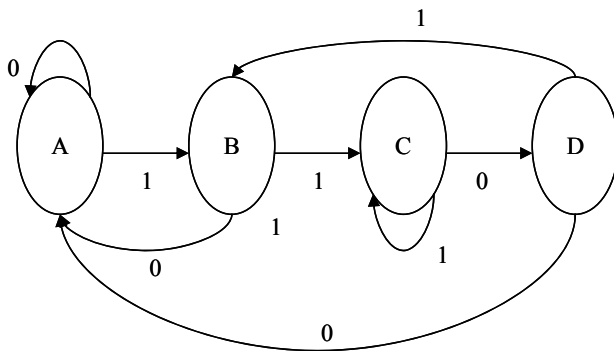
$$D1 = Q1Q0/ + Q2/.Q1Q0$$

$$D0 = Q0/$$

Partie 2 : Reconnaissance de séquence

Q3) Réaliser un automate de Moore avec une entrée E et une sortie S qui reconnaît la séquence 110. 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)



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

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

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

$$S=Q1Q0$$

$$D1 = \sum m(2,5,6) = Q1Q0' + Q2' / Q1 Q0$$

$$D0 = \sum m(2,4,7) = Q2Q1/Q0' + Q2/Q1Q0' + Q2Q1Q0$$

Partie 4 : Programmation NIOS

Q4) Que fait le programme suivant ? ? (Donner le programme C correspondant)

```

ORHI R1, R0, F000H
ORI R2, R1, 0FFCH
XOR R3, R3, R3
ADDI R4, R0, 0AH
Boucle: LDW R5, 0(R1)
        ADDI R1, R1, 4
        BLT R5, R4, Suite
        ADDI R3, R3, 1
Suite : BLE R1, R2, Boucle

```

Calcule le nombre de valeurs supérieures ou égales à 10 dans un tableau de 1024 entiers.

```

Int T[1000], N=0 ;
For (i=0 ; i<1024 ; i++)
    If (T[i] >=10) N++;

```

Un programmeur veut faire la somme des entiers 32 bits compris entre les adresses F0000000_H et F000FFFC_H.

Q5) Le programme ci-dessous est-il correct ? S'il ne l'est pas, indiquer le bogue et corriger le programme.

```

ORHI R1, R0, F000H
ADDI R2, R1, FFFCH
ADD R3, R0, R0
Boucle : LDW R4, 0(R1)
        ADD R3, R3, R4
        ADDI R1, R1, 4
        BLE R1, R2, Boucle

```

L'instruction ADDI R2, R1, FFFC effectue une extension de signe. Le résultat obtenu est donc F0000000 + FFFFFFFC soit EFFFFFFC et non F000FFFC comme souhaité.

Il faut remplacer ADDI R2,R1, FFFC par ORI R2,R1, FFFC pour obtenir le programme correct.

Partie 5 : Microarchitecture et temps d'exécution

Soit le programme assembleur NIOS ci-dessous

```

ORHI R1,R0,F000H
ADDI R2,R1,40010
XOR R3,R3,R3
Loop : LDW R4,(R1)
      LDW R5,40010(R1)
      ADD R4,R4,R5
      ADD R3,R3,R4
      ADDI R1,R1,4
      BLT R1,R2,Boucle
    
```

Les instructions s'exécutent sur le processeur non pipeliné dont le chemin de données est présenté en figure 1.

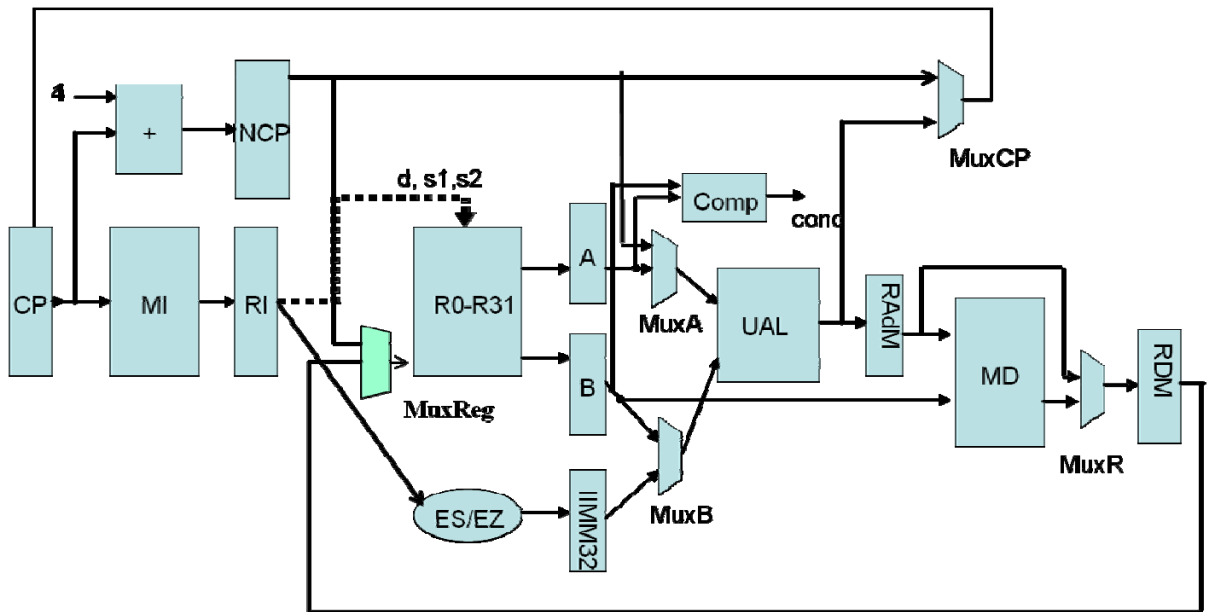


Figure 1 : Chemin de données non pipeliné

Q6) Donner le temps d'exécution d'une itération de la boucle en nombre de cycles d'horloge

Toutes les instructions prennent 5 cycles sauf BGT (3 cycles)
Total : 28 cycles

Q7) Donner le nombre d'itérations de la boucle et le temps d'exécution total du programme en nombre de cycles d'horloge.

Il y a 100 itérations de la boucle.
Temps total d'exécution = $28 \cdot 100 + 15 = 2815$ cycles

Q8) Donner le programme C correspondant au programme assembleur NIOS

```
Int T[200], i, S=0 ;  
For (i=0 ; i<100 ;i++)  
    S+= T[i] +T[i+100] ;
```