Représentation des nombres et arithmétique binaire

1. Conversions

1. Convertir les nombres décimaux suivants en binaire naturel puis en BCD.

37, 183, 511, 1234, 2089

2. Convertir les nombre binaires suivants en hexadécimal et en décimal.

1001100, 10001101, 1011111, 11100011

2. Nombres signés

2.1. représentation sur 4 bits en complément à 2

1. Quelles sont les valeurs extrêmes représentables ?

2. Donner les compléments à 2 des nombres binaires suivant : 0010, 1011; 0100; 1111

3. Réalisez les additions suivantes : 2 + 3; 5 + (-2); 6 + (-8); 4 + 6

2.2. Représentation sur 8 bits en complément à 2

1. Quelles sont les valeurs extrêmes représentables ?

2. Convertir en binaire les nombres suivants : 65 ; -65 ; -18

3. Effectuer les opérations suivantes en binaire et en décimal:

57h+26h ; 6Fh+32h ; 5Eh-36h ; 36h-5Eh

2.3. Conversion de format

A partir d'une représentation 4 bits on souhaite effectuer les conversions suivantes :

- représentation 4 bits non signée vers une représentation 8 bits signée
- représentation 4 bits signée vers une représentation 8 bits signée

1. Reprendre les nombres donnés au 2.1.2 et réalisez les 2 conversions de format.

2. Quelles sont finalement les opérations à effectuer pour réaliser une extension de format sur les nombres signés ?

3. Multiplications et divisions

3.1. Multiplications

1. Effectuez les multiplications des nombres non signés ci-dessous et vérifiez vos résultats en décimal.

0101× 0111 ; 0111 × 1001 ; 1101 × 1011 ;

2. Lorsque l'on multiplie 2 nombres de n bits, sur combien de bits est le résultat ?

3. Reprenez les multiplications précédentes en considérant les nombres comme des nombres signés 4 bits.

3.2. Divisions

Effectuer les divisions entières des nombres non signés suivants :

0000 1001%0000 0011 ; 0011 1101%0000 1001 ; 1001 1011 % 0000 0111 ; 1001 1011 % 1111 1001

4. Nombres flottants

- Donner la représentation flottante (IEEE 754) du nombre entier non signé : 1010 1011 1010 0011 1001 0010 0100 1001.
- Donner le plus petit et le plus grand nombre entier non signé ayant la même représentation flottante.

Algèbre de Boole et logique combinatoire

1. Tables de vérité

Donner les tables de vérité des équations booléennes suivantes :

$x_1 = a\left(b+c\right)$; $x_2 = a\bar{b}c + \bar{a}bc$; $x_3 = \left(a+b\right)\left(\bar{b}+\bar{c}\right)$; $x_4 = \overline{a+\bar{b}c}$; $x_5 = \overline{a\,\overline{(b+\bar{c})}\,d}$

2. Simplifications algébriques

Simplifier algébriquement les expressions booléennes suivantes :

$E_1 = ab\bar{c} + \bar{a}\bar{b}c + a\bar{b}\bar{c} + a\bar{b}c$

$E_2 = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc$

$E_3 = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}\bar{c}d + \bar{a}\bar{b}c\bar{d} + a\bar{b}c\bar{d} + a\bar{b}\bar{c}\bar{d} + a\bar{b}\bar{c}d + ab\bar{c}\bar{d}$

3. Tableaux de Karnaugh

Simplifier les expressions suivantes à l'aide des tableaux de Karnaugh :

$E1 = \bar{a}\bar{b}\bar{c} + \bar{a}bc + a\bar{b}\bar{c} + a\bar{b}c + abc$

$E2 = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}c\bar{d} + \bar{a}b\bar{c}d + \bar{a}bcd + a\bar{b}\bar{c}\bar{d} + a\bar{b}c\bar{d} + ab\bar{c}d$

$E2 = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}\bar{c}d + \bar{a}\bar{b}cd + \bar{a}\bar{b}c\bar{d} + a\bar{b}\bar{c}d + a\bar{b}c\bar{d} + a\bar{b}cd + ab\bar{c}d + abcd$

4. Equation booléenne

Donner l'équation booléenne la plus simple possible associée à la fonction suivante :

a b c	D
0 0 0	1
0 0 1	X
0 1 0	X
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	X
1 1 1	0

X état indifférent

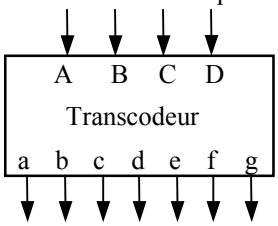
5. Opérateurs

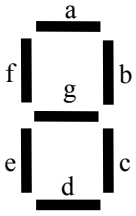
Donner le logigramme associé à l'équation $E_1 = \bar{a}\bar{b} + \bar{b}c$ en utilisant uniquement des portes *NON ET* puis *NON OU* (Opérateurs complets).

6. Transcodeur BCD 7 segments

On souhaite réaliser un transcodeur qui prend en entrée un mot BCD de 4 bits ABCD (D poids faible) et qui produit les sorties (a b c d e f g) de commande des leds de manière à afficher les symboles décimaux correspondants sur l'afficheur 7 segments.

- Dresser les tables de vérité des segments a et b uniquement.
- Simplifier les expressions logiques.
- Dresser le logigramme.





7. Additionneur

On souhaite réaliser un circuit logique qui additionne 2 mots de N bits A et B. Pour cela, on décompose cet additionneur N bits en N additionneurs 1 bit, chacun effectuant l'opération sur les bits de rang i.

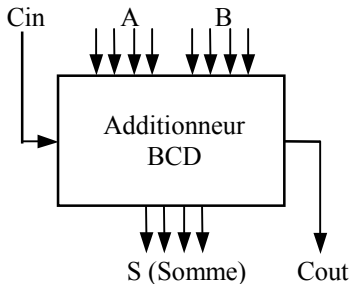
1. Quelle(s) opération(s) doit réaliser l'additionneur 1 bit de rang i ? En déduire son nombre d'entrées/sorties.
2. Donner le logigramme de cet additionneur 1 bit.
3. Donner la structure de l'additionneur N bits.

4. Quel est le problème lié à cette structure ? Comment pourrait-on y remédier ?

8. Additionneur BCD

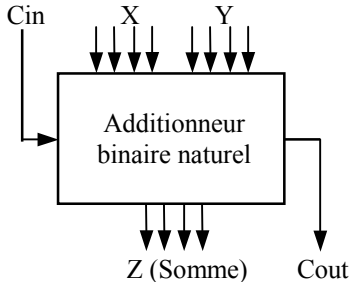
On souhaite réaliser un additionneur BCD qui effectue l'addition de deux nombres BCD **A** (A3 A2 A1 A0 avec A3 poids fort) et **B** (B3 B2 B1 B0 avec B3 poids fort). Cet additionneur devra pouvoir être réutilisable lors d'une mise en cascade pour effectuer des additions de nombres BCD codés sur plus de 4 bits.

Pour ce faire, le dispositif doit comprendre une entrée de retenue **Cin** provenant des étages de poids inférieurs et fournir une retenue **Cout** pour les étages de poids supérieurs.



0 ≤ A ≤ 9
0 ≤ B ≤ 9
Cin = 0 ou 1
0 ≤ S ≤ 9
Cout = 0 ou 1

Pour réaliser cet additionneur BCD on dispose d'additionneurs binaires naturels. Ces additionneurs effectuent la somme de deux nombres binaires et disposent de deux broches Cin et Cout pour les retenues entrante et sortante (schéma ci-dessous).



0 ≤ X ≤ F
0 ≤ Y ≤ F
Cin = 0 ou 1
0 ≤ Z ≤ F
Cout = 0 ou 1

Proposer un schéma d'additionneur BCD utilisant un ou plusieurs additionneurs binaire naturel (reprendre la forme donnée ci-dessus) et des portes logiques élémentaires.

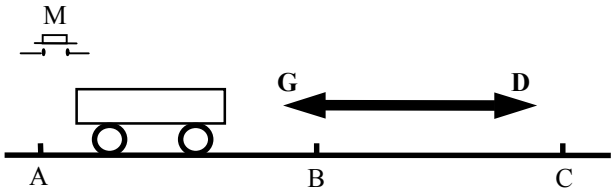
9. Multiplexeur

Réaliser la fonction logique $\bar{a}\bar{b}c + \bar{a}bc + ab\bar{c} + abc$ à l'aide d'un multiplexeur 8 vers 1.

Logique séquentielle

1. Synthèse logique

On considère un chariot pouvant se déplacer soit à droite soit à gauche. A l'origine, le chariot se trouve en A. Sur pression du bouton M, il se déplace vers la droite jusqu'au point B où il s'arrête. Sur une seconde pression, il se déplace toujours vers la droite jusqu'au point C. Une dernière pression sur le bouton M entraîne le déplacement du chariot vers la gauche du point C jusqu'au point A sans arrêt en B.



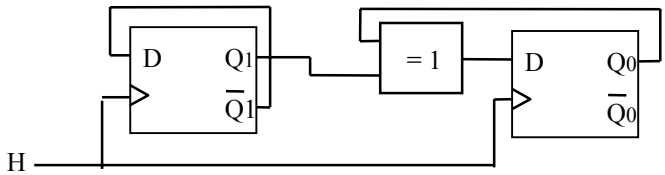
Variables booléennes :
G, **D** : Direction de marche du moteur (actionneur et capteur)
M :bouton poussoir (capteur)
A, **B**, **C** : capteurs de position

Note : l'action sur M est fugitive (Même lorsque la pression est relâchée (M = "0") le chariot ne doit pas s'arrêter mais effectuer le cycle décrit).
G et D sont les commandes des moteurs mais également des variables accessibles pour la détection du sens de déplacement. G et D ne doivent pas être actifs simultanément.

- Ecrire en pseudo langage la commande des moteurs **G** et **D** en fonction des variables dont vous disposez. (Vous utiliserez les instructions Si alors sinon et des boucles tant que)
- Ecrire les équations booléennes de **G** et **D**.

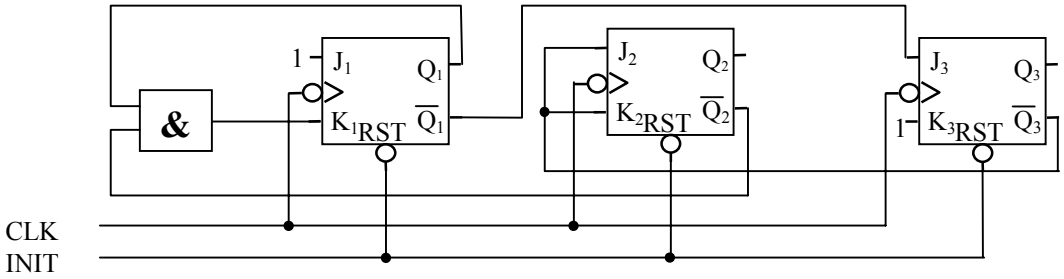
2. Bascules D

Soit le montage suivant :

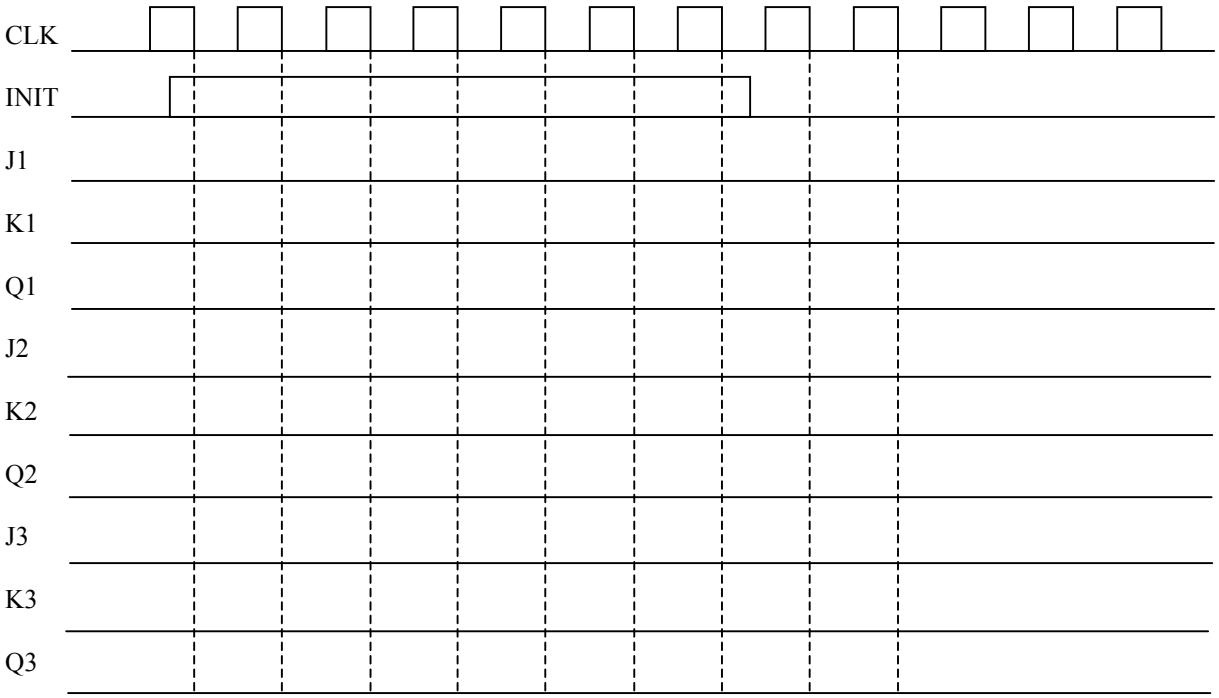


On suppose qu'a l'instant t=0, Q0 = Q1 = 0. Représentez les chronogrammes d'évolution des sorties Q1, Q0 en fonction de l'horloge H.

3. Bascules JK



Compléter le chronogramme suivant :



4. Compteurs

On dispose de bascules D actives sur front montant avec Preset et Clear asynchrones ainsi que de portes logiques élémentaires..

1. Réalisez un compteur asynchrone par 5 (5 états complets) réalisant le cycle suivant : 0, 1, 2, 3, 4, 0 …

2. Reprendre l'exercice en réalisant un compteur synchrone

3. On souhaite modifier le cycle un fois sur 2. Une fois sur 2 l'état correspondant à la valeur 4 est sauté ce qui donne pour le cycle : 0, 1, 2, 3, 4, 0, 1, 2, 3, 0 …

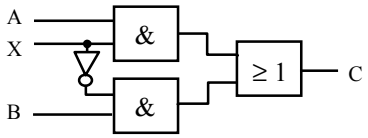
Donnez le schéma d'un système synchrone réalisant le nouveau cycle.

Descriptions VHDL

1. Descriptions systèmes combinatoires

Donner les descriptions VHDL (Entity/Architecture) des fonctions combinatoires ci après en utilisant des assignations conditionnelles ou sélectives.

1.



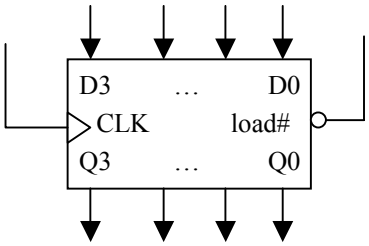
2. Décodeur 1 parmi 8 à sorties actives niveau bas

2. Bascules et registres

Donner les descriptions VHDL (Architecture) des composants synchrones suivants

1. Bascule JK active sur front descendant avec Preset et Reset asynchrone

2. Registre 4 bits à lecture/écriture parallèle synchrone

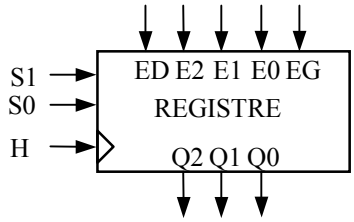


3. Registre à décalage 3 bits

On désire réaliser un registre à chargement série ou parallèle à l'aide de bascules D actives sur front montant. Ce registre dispose de 3 entrées parallèles (E3 E2 E0) d'entrées séries gauche et droite pour remplacer les bits manquants lors des décalages (respectivement ED et EG). Ce registre dispose d'une sortie parallèle 3 bits (Q2 Q1 Q0).

Le fonctionnement du circuit est défini par deux entrées de commande (S1 S0) :

- mode 00 ➔ maintien du décaleur dans l'état courant
- mode 01 ➔ décalage synchrone à gauche (utilisation de l'entrée série EG)
- mode 10 ➔ décalage synchrone à droite (utilisation de l'entrée série ED)
- mode 11 ➔ chargement parallèle d'une valeur



1. Donnez la description VHDL de ce circuit.

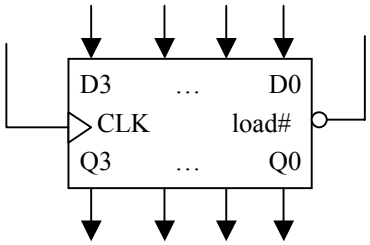
2. Pour chacune des entrées Di des bascules écrivez l'équation logique de Di en fonction des variables du système

3. Donnez le logigramme du circuit décrit

4. Compteurs/décompteurs

1. Compteur synchrone par 6

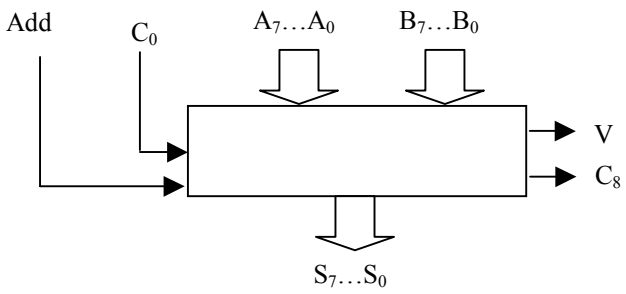
2. Compteur synchrone par 16 avec chargement synchrone d'une valeur



5. Arithmétique

Additionneur-soustracteur 8 bits
avec génération de retenue (**C_8**) et d'overflow (**V**).

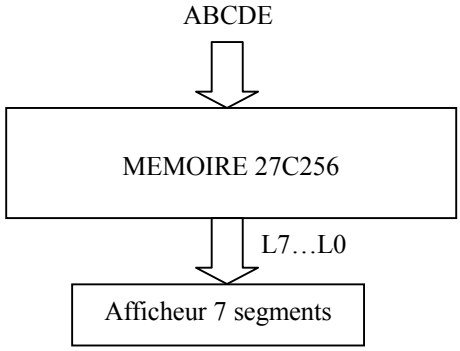
Si "Add" est à 1 : addition sinon soustraction

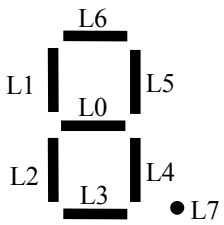


Mémoire et séquenceur

1. Réalisation d'un transcodeur BCD/7segments à l'aide d'un mémoire

On dispose d'une PROM 256 bits (documentation fournie en annexe).
On souhaite utiliser ce composant pour réaliser un transcodeur BCD/7SEG selon le schéma suivant :





ABCD est le digit BCD sur 4 bits (A poids fort et D poids faible), E correspond au point décimal de l'afficheur.

1. Rappelez le fonctionnement d'une mémoire

2. Quelle est la particularité ce cette mémoire ?

3. Quelle est l'utilité des broches $\overline{CE}$ et $\overline{OE}$?

4. Expliquez comment réaliser le câblage de la mémoire pour réaliser ce transcodeur.

5. En fonction du câblage choisi, donnez le contenu de la mémoire pour obtenir le fonctionnement désiré.

2. Séquenceur

On souhaite réaliser un séquenceur d'un automatisme séquentiel à l'aide d'une mémoire 352 bits (ROM 32x11). Le graphe de description de l'automatisme et la structure du séquenceur sont fournis en annexes.

2.1. Description :

- Lorsqu'une adresse est fournie en entrée de la mémoire, le contenu de la cellule mémoire (11 bits) référencée est disponible en sortie.
- L'adresse du mot mémoire est fournie par la sortie du compteur.
- La sortie du multiplexeur MUX est reliée au $\overline{load}$ du compteur.
- Compteur préchargeable synchrone :
  - la valeur courante du compteur est constamment disponible en sortie
  - lorsque load est mis à 0 le mot placé en entrée est chargé sur un front actif d'horloge et le compteur prend alors la nouvelle valeur après le front
  - lorsque $\overline{load}$ est à 1, la valeur courante du compteur s'incrémente d'une unité à chaque front d'horloge
  - Lorsque $\overline{clear}$ est à 0, la valeur du compteur est forcée à 0000 sur le front actif d'horloge.
- X Y et Z sont les sorties de commande du système actives niveau haut (lorsque les variables ne sont pas mentionnées elles sont au niveau logique 0, inversement lorsqu'elle sont mentionnée elle sont active donc à 1).

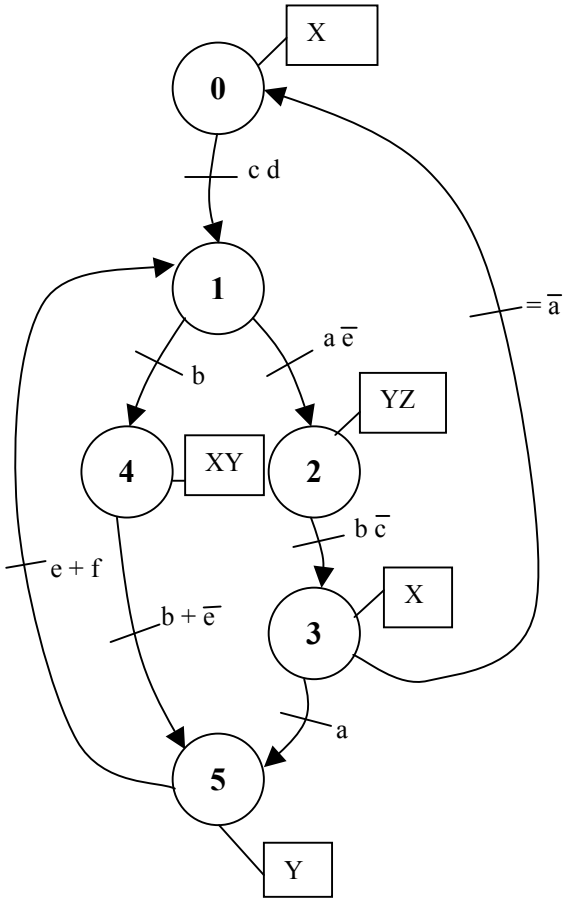
1. Expliquez la structure du mot mémoire.

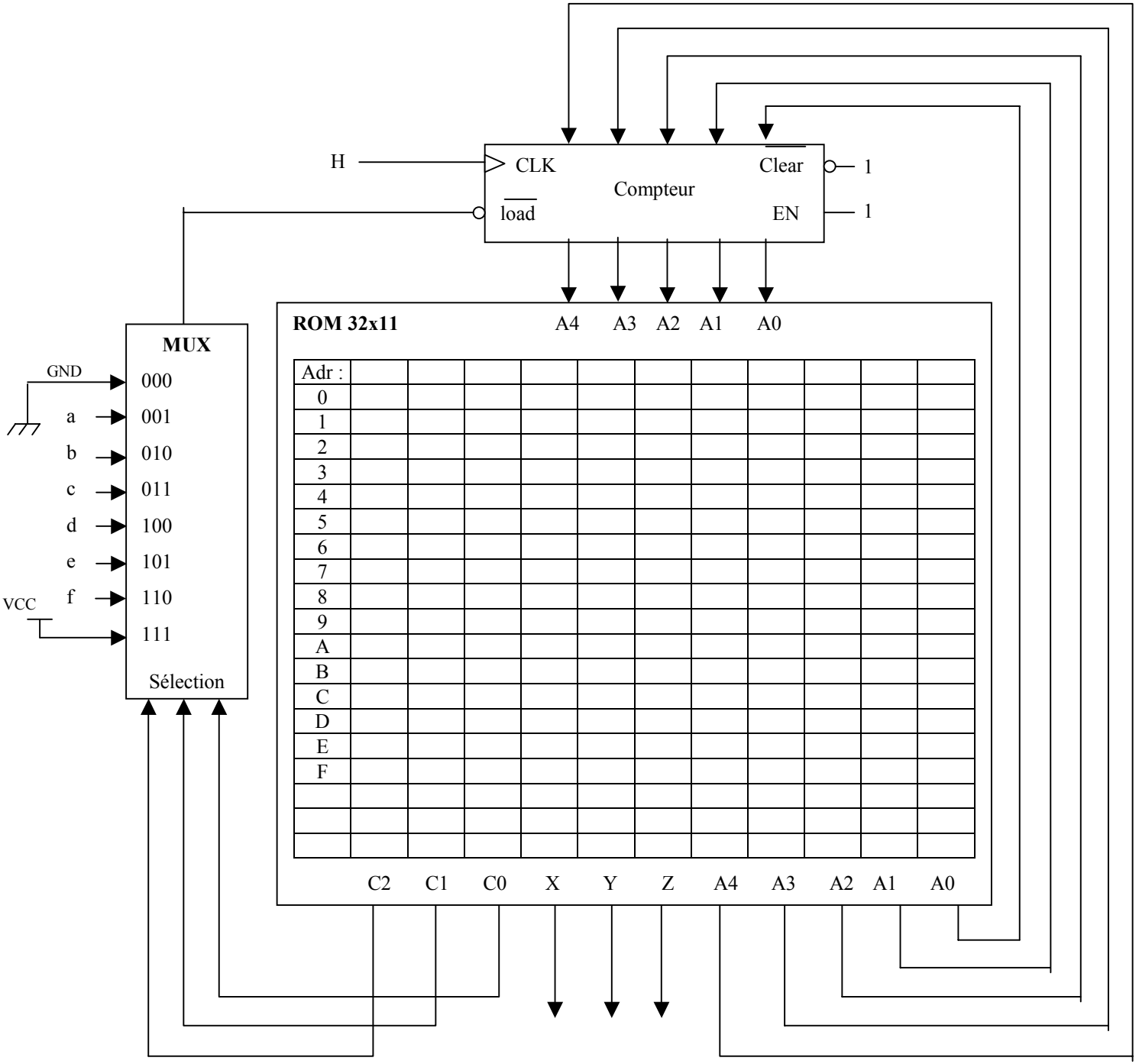
2. Donnez le contenu de la mémoire pour obtenir le fonctionnement désiré.

3. Synthétisez le même graphe de commande à l'aide de bascules D (séquenceur câblé)

4. Donnez la description VHDL de ce séquenceur

2.2. Schéma


X
0
c d
1
= ā
a ē
b
YZ
4
XY
2
b c̄
e + f
X
b + ē
3
a
5
Y

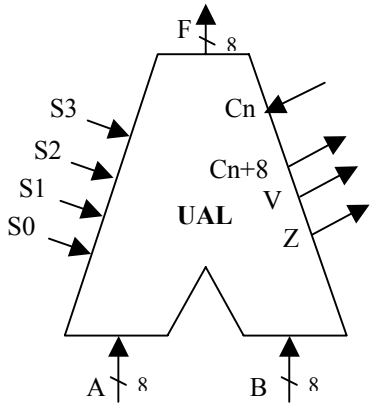

H
CLK
Clear
1
Compteur
load
EN
1
MUX
GND
000
a
001
b
010
c
011
d
100
e
101
f
110
VCC
111
Sélection
ROM 32x11
A4 A3 A2 A1 A0
Adr :
0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F
C2 C1 C0 X Y Z A4 A3 A2 A1 A0

Unité d'exécution entière

1. Etude d'une UAL

1.1. Description

On considère l'unité arithmétique et logique (UAL) 8 bits suivante :



S	Opération F	Cn+8	V	Z
0 0 0 0	A	-	-	Δ
0 0 0 1	B	-	-	Δ
0 0 1 0	non A	-	-	Δ
0 0 1 1	non B	-	-	Δ
0 1 0 0	A plus B	Δ	Δ	Δ
0 1 0 1	A plus B plus Cn	Δ	Δ	Δ
0 1 1 0	A moins B	Δ	Δ	Δ
0 1 1 1	A moins B moins Cn	Δ	Δ	Δ
1 0 0 0	B moins A	Δ	Δ	Δ
1 0 0 1	B moins A moins Cn	Δ	Δ	Δ
1 0 1 0	A+B	-	-	Δ
1 0 1 1	non (A+B)	-	-	Δ
1 1 0 0	A · B	-	-	Δ
1 1 0 1	non (A.B)	-	-	Δ
1 1 1 0	A ⊕ B	-	-	Δ
1 1 1 1	non (A ⊕ B)	-	-	Δ

- : non affecté; Δ : changement possible

Détail du fonctionnement arithmétique

- S : S3 S2 S1 S0 : Sélection opération
- Cn entrée retenue
- Cn+8 : Sortie retenue (addition sur 8 bits)
- V : indicateur de dépassement (overflow)
- Z : indique si le résultat est à zéro

Cn, n'est pas pris en considération sur les opérations logique.
Cette UAL peut également être utilisée comme comparateur (mode A moins B) :

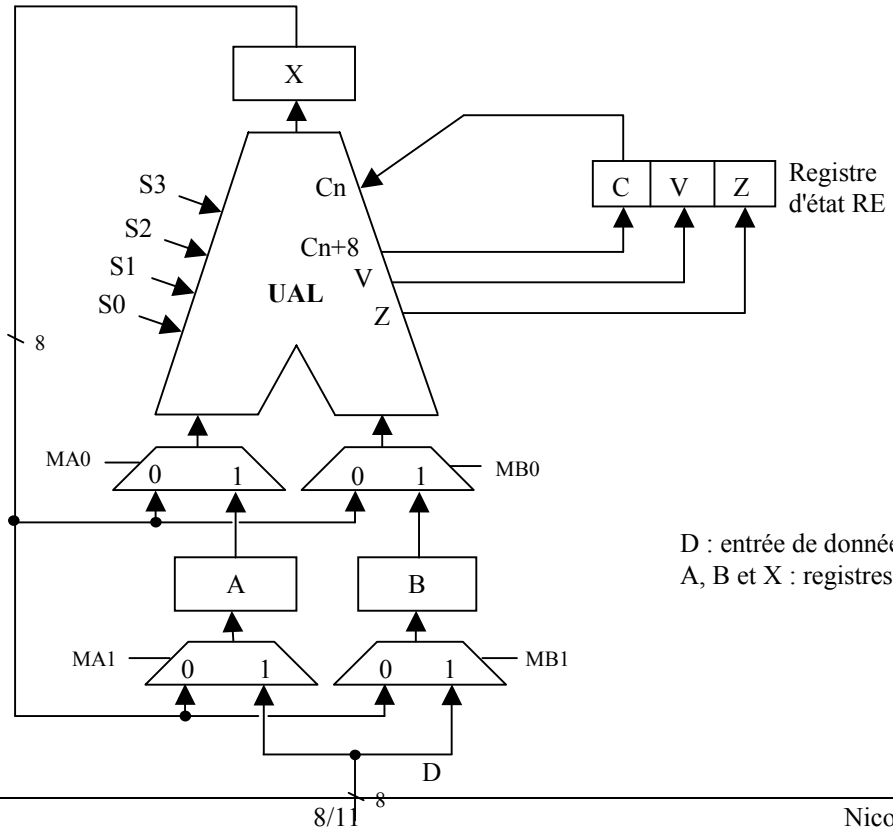
- Z = 1 si A = B
- Cn+8 est à 1 si A > B

1.2. Exemple d'utilisation

1.2.1. Description du système opératif

On considère le circit logique suivant qui utilise l'UAL décrite précédemment. Ce circuit admet 16 entrées dites entrées de commandes du circuit.

Schéma de pricipe :



D : entrée de données
A, B et X : registres 8 bits

Entrées de commande du circuit:

RA1	RA0	RB1	RB0	RX1	RX0	MA1	MA0	MB1	MB0	CLC	SEC	S3	S2	S1	S0

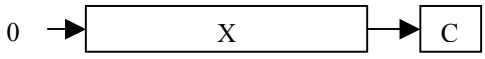
Toutes les commandes sont actives niveau haut

1) Descriptif des commandes :

1. Registres associés à l'UAL : A,B,X

Les 3 registres A, B, X associés à l'UAL sont des registres synchrones acceptant 4 types de fonctionnement en fonction des mots de commandes RA1 RA0 pour le registre A, RB1 RB0 pour B et RX1 RX0 pour X
Rx1 Rx0
0 0 : la valeur présente dans le registre est conservée
0 1 : décalage logique à gauche (poids faible vers poids fort)
1 0 : décalage logique à droite (poids fort vers poids faible)
1 1 : chargement de la valeur présente en entrée

Le décalage logique consiste à décaler bit à bit la valeur du registre soit vers la gauche soit vers la droite. Le bit perdu est alors sauvegardé dans le bit de retenu C et le nouveau bit introduit est un zéro.
Exemple de décalage logique vers la droite (RX1 RX0 = 1 0) : le poids fort est remplacé par un 0 et l'ancien poids faible est sauvegardé dans C



2. Commande des multiplexeur

En fonction des valeurs présentes sur les broches de sélection des multiplexeurs (Mx1, Mx0), une entrée est aiguillée vers la sortie. Exemple : si MA0 est à 1, c'est la sortie du registre A qui est aiguillé vers l'entrée de l'UAL, sinon c'est la sortie du registre accumulateur X.

3. Commande de l'UAL

L'entrée S (S3 S2 S1 S0) permet de sélectionner l'opération à effectuer au sein de l'UAL

4. Commande du registre d'état

CLC : Remet à zéro le bit C (CLear Carry)
SEC : Met à 1 le bit C (SEt Carry)

Note : Les opérations sur les registres sont synchrones (action sur le registre : commande + front actif d'horloge) en revanche l'UAL et les multiplexeurs sont des circuits combinatoires.

2) Etude d'un micro-programme

On suppose que les temps de réponses des différents composants logiques sont inférieurs à 1 période d'horloge. Chaque ligne correspond à une période d'horloge et le passage d'une ligne à une autre correspond à un front actif d'horloge
Compléter le tableau suivant :

Commande (hexa)	D (hexa)	A	B	X	C	V	Z
3080	7F	0	0	0	0	0	0
C200	30						
0D44	30						
0400	30						
0C45	30						
C000	43						

3) Détermination d'un micro-programme

On souhaite réaliser l'opération suivante :
chargement de A
chargement de B
comparaison de A et B
multiplication par 2 de A
soustraction de A – B et sauvegarde dans X
multiplication par 2 de X
soustraction de X – B et sauvegarde dans A

Donnez la séquence de commandes à envoyer au système. On suppose que les données sont présentes au bon moment sur le bus D.

Séquencement d'un microprocesseur simplifié

1. Description

Bus

Un bus de données de 8 bits bidirectionnel
Un bus d'adresse de 16 bits monodirectionnel

UAL

Unité arithmétique et logique avec 8 commandes possibles :

U2 U1 U0	Opération F
0 0 0	Non (A)
0 0 1	A plus B
0 1 0	B moins A
0 1 1	A moins B
1 0 0	A ⊕ B
1 0 1	A + B
1 1 0	A · B
1 1 1	Reset

Sorties UAL : C, V, Z, N

Arithmétique
C = 1 si retenue propagée
V = 1 si dépassement de capacité (overflow)
Comparaison (mode A-B) :
C = 1 si A<B, 0 si A≥B en non signé
Z = 1 si A=B
N = 1 si le résultat est négatif (nombres signés)

Registre

Les registres ont une commande de chargement ldxxx
Lorsque la commande ldxxx est active la valeur présente en entrée est chargée dans le registre
Fonction :

- Registre d'instruction : registre 8 bits qui stocke l'instruction à exécuter (les instructions sont codées sur 8 bits).
- Accumulateur A : registre 8 bits de travail associé à l'UAL
- Accumulateur B : registre 8bits de travail associé à l'UAL
- Registre d'état : registre destiné à stocker les drapeaux (bits d'état) de la machine et notamment les indicateurs d'opération de l'UAL
- Compteur de programme : registre 16 bits où est stockée la prochaine instruction à exécuter
- Registre d'adresse : registre 16 bits destiné à stocker une adresse temporaire (adresse d'une variable stockée en mémoire)

Séquenceur

Gère de manière synchrone toutes les commandes du système

1.1. Séquence de recherche (fetch)

Etablir le graphe à états correspondant à la recherche en mémoire d'une instruction. Pour chaque état donner les micro-commandes à activer.

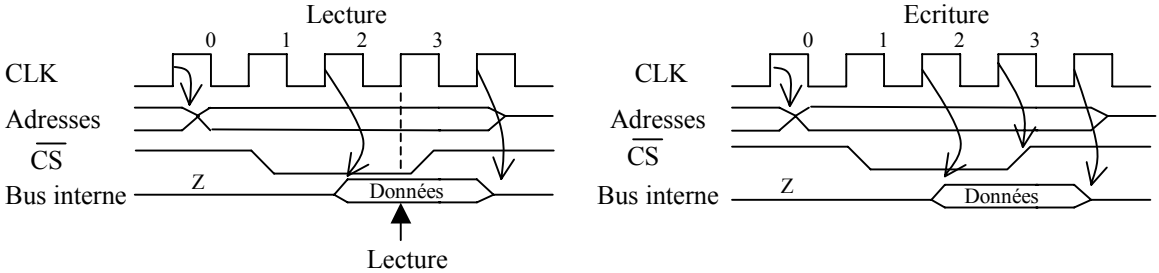
1.2. Séquence de quelques opérations

Pour chacune des instructions suivantes donnez le graphe à états correspondant :

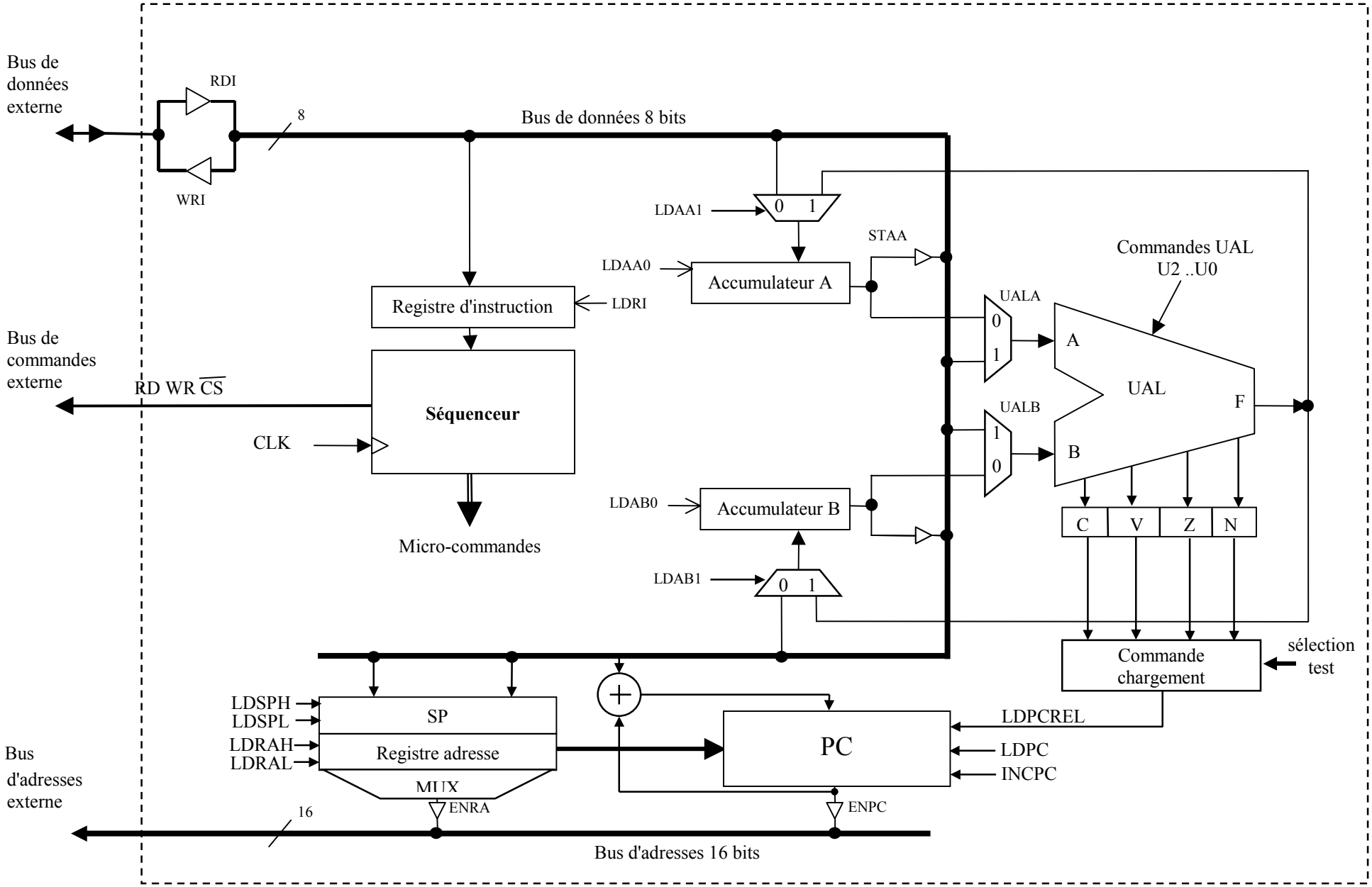
- LDAA #$11 — charge 11h dans l'accumulateur A
- ADDA #$CF — effectue l'addition de CFh avec A et stocke dans A
- STAA $A000 — Sauvegarde le contenu de l'accumulateur A dans la case mémoire d'adresse A000h

1.3. Lecture/écriture mémoire

Les lectures et écriture mémoire durent 4 cycles. Les chronogrammes de lecture écriture sont donnés ci dessous :



Note : le décodage d'une instruction s'effectue sur la dernière période d'horloge. Ainsi l'exécution ou la recherche de la suite de l'instruction s'effectue sur le cycle suivant.


Bus de données externe
RDI
WRI
8
Bus de données 8 bits
LDAA1
0 1
STAA
LDAA0
Accumulateur A
Commandes UAL
U2 ..U0
Registre d'instruction
LDRI
UALA
0
1
A
UAL
F
Bus de commandes externe
RD WR CS
Séquenceur
CLK
UALB
1
0
B
LDAB0
Accumulateur B
C
V
Z
N
Micro-commandes
LDAB1
0 1
Commande chargement
sélection test
LDSPH
LDSPL
SP
LDRAH
LDRAL
Registre adresse
PC
LDPCREL
LDPC
INCPC
Bus d'adresses externe
MUX
ENRA
ENPC
16
Bus d'adresses 16 bits