Architecture des Ordinateurs

Licence Informatique de Lille

Examen Janvier 2002

Durée : 3h – tous documents autorisés

Exercice 1. *Multiplication en binaire*

On désire réaliser le circuit de multiplication d'une Unité Arithmétique et Logique 4 bits.

Question 1.1*. Quelle est la taille maximale (en bits) du résultat d'une multiplication de deux nombres de 4 bits ?*

Pour réaliser notre circuit de multiplication, nous allons utiliser la même méthode que celle utilisée pour les nombres décimaux :

```
En décimal      En binaire
    1515                 1011 (8+2+1=11)
  x  201               x  110 (4+2=6)
    1515                    0
+     0-            +   1011-
+ 3030--            +  1011--
= 304515            = 1000010 (64+2=66)
```

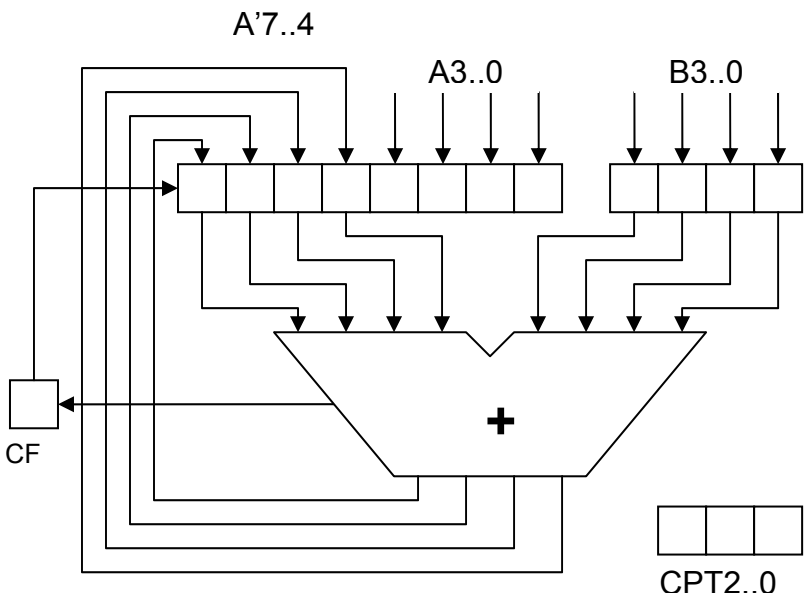
On peut noter qu'en décimal ou en binaire, on obtient les chiffres de droite à gauche. Par exemple, en décimal, le 5 de droite ne pourra pas être modifié par les additions ultérieures.

Question 1.2*. Faites les multiplications suivantes en binaire : 13 par 9 et 7 par 10.*

Question 1.3*. Donnez l'automate de l'unité de contrôle réalisant la multiplication selon la méthode expliquée ci-dessus avec le circuit dessiné ci-contre.*

Les différents signaux de commande sont :

- SET_A qui charge le registre A3..0 et qui met A'7..4 à 0
- SET_B qui charge le registre B3..0
- ADD qui charge le résultat de l'addition de A'7..4 et B3..0 dans A'7..4 et met la retenue dans la bascule CF
- DEC qui décale le registre A'7..4:A3..0 d'un bit vers la droite. Le bit A'7 copie la valeur de CF.
- RESET qui met à zéro le registre CPT2..0
- INC qui incrémente de 1 le registre CPT2..0



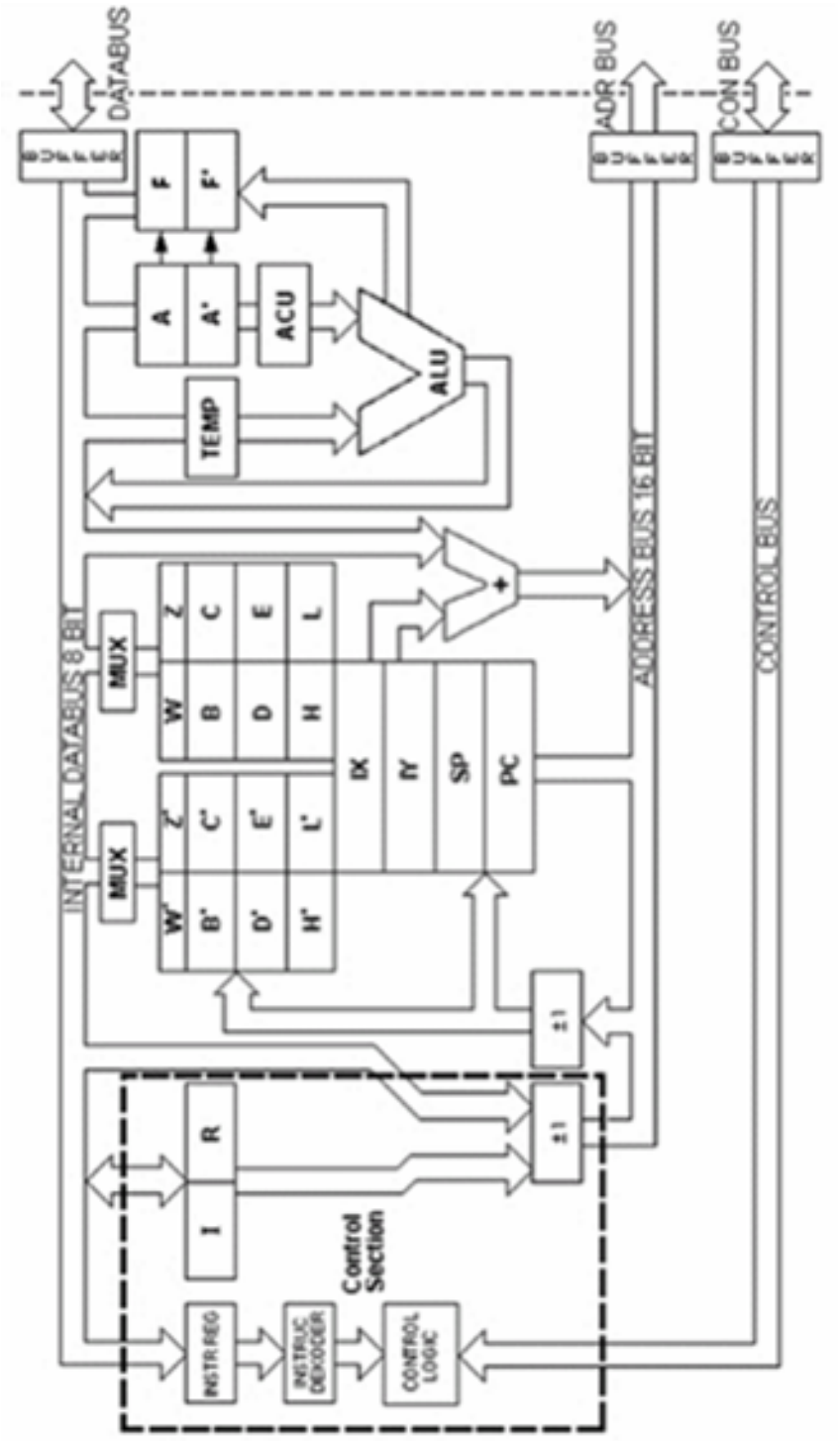
***Question 1.4**. Donnez une implémentation (Registre d'état, Gestionnaire des transistions et Générateur des signaux de contrôle) pour cet automate. On ne cherchera pas à optimiser les fonctions logiques mais on précisera les stimuli.*

Exercice 2. *Microprogrammation d'un Z80*

On considère l'architecture interne du microprocesseur Z80 inventé en 1976 et toujours utilisé de nos jours dans des appareils aussi variés que des distributeurs automatiques, des téléphones portables ou encore des consoles de jeux portables célèbres.

On peut distinguer notamment :

1. trois bus : un bus de données, un bus d'adresse et un bus de contrôle.



2. un bloc de registres (au centre) :
   - W, Z, B, C, D, E, H, L qui sont des registres 8 bits (on ignorera les registres marqués d'un prime) ;
   - IX, IY (deux registres dits d'indexation), SP (le pointeur de pile) et PC le compteur ordinal

     qui sont des registres 16 bits.

Notons que les registres 8 bits peuvent être couplés deux par deux pour faire un registre 16 bits : WZ, BC, DE, HL. Les registres W et Z sont réservés à l'usage interne du processeur, ceci signifie que l'utilisateur ne dispose pas d'instructions machine pour y accéder.

3. un bloc de calcul formé :
   - d'un accumulateur 8 bits désigné A comme à l'accoutumée ;
   - d'un registre temporaire ;
   - d'une Unité Arithmétique et Logique ;
   - d'un registre d'état noté F pour Flags.
4. un bloc de contrôle pour lequel on ne retiendra que le registre d'instruction.

Enfin, notons que le Z80 utilise la notation grand boutiste (*big endian*) pour représenter les nombres 16 bits en mémoire.

Question 2.1. *Donnez les micro-instructions de la phase 1 (lecture de l'instruction) du Z80. À chaque fois que vous introduisez une micro-instruction, donnez sa signification.*

Question 2.2. *Donnez les micro-instructions de la phase 3 pour les instructions suivantes :*

Mnémonic	Code langage machine	Commentaire
LD HL, nnnn	21 nn nn	Charge la valeur nnnn dans HL
LD HL, (nnnn)	2A nn nn	Charge la valeur stockée à l'adresse nnnn dans HL
ADD A,C	81	Addition de A et C et range dans A
JP nnnn	C3 nn nn	Saut à l'adresse nnnn
PUSH BC	C5	Met la valeur de BC dans la pile
POP HL	E1	Dépile et mets la valeur dans HL
CALL nnnn	CD nn nn	Saut à la sous-routine à l'adresse nnnn
RET	C9	Retour de sous-routine

Exercice 3. *Programmation assembleur 80x86*

On considère le squelette de programme suivant :

```
; --------------------------------
; Directives
; --------------------------------
    DOSSEG
    .MODEL small
    .STACK 200h
; --------------------------------
; Donnees
; --------------------------------
    .DATA
    cr DB 0Dh, 0Ah, '$'
    tampon DB 100 DUP (?)

; inserer ici vos donnees

; --------------------------------
; Code
; --------------------------------
    .CODE
Begin: ; point d'entrée du prog
    mov AX,@data
    mov DS,AX
    jmp debut

; --------------------------------
; procédures/fonctions
; --------------------------------
lirechaine PROC
; en entree le nombre maximal de
;   caractere dans CL
;   l'@ de la chaine dans BX
    push CX
    push BX
    push AX
    mov CH, 1
    mov AH, 1
bouclelc:
    cmp CH, CL
    jge finlc
    int 21h
    mov [BX], AL
    inc BX
    inc CH
    cmp AL, 0Dh
    jne bouclelc
finlc:
    mov BYTE PTR [BX], 24h
    pop AX
    pop BX
    pop CX
    ret
  lirechaine ENDP

isnum PROC
; en entree, l'@ de la chaine est
;    dans BX
; en retour AL vaut 13 (0D) si
;    c'est un nombre, n'importe
; quoi d'autre sinon
    push BX
bouclein:
    mov AL, [BX]
    cmp AL, 0Dh
    je finin
    cmp AL, 30h
    jl finin
    cmp AL, 39h
    jg finin
```

```
    inc BX
    jmp bouclein
finin:
    pop BX
    ret
isnum ENDP

; inserer ici vos fonctions

debut:
    mov BX, OFFSET tampon
    mov CL, 100
    call lirechaine
    call isnum
    cmp AL, 0Dh
    je est_int

; insérer vos tests

est_int:
      ; blabla
      jmp fin

fin:
    ; fin du programme
    mov AH, 4Ch
    int 21h          ; sortie du
programme
    END Begin
```

Question 3.1. *Écrivez les fonctions isfloat, ispalindrom, ispangram qui testent respectivement si une chaîne de caractères représente un « flottant », i.e. un nombre à virgule (NB. Un entier naturel est considéré comme un flottant), contient un palindrome (qui donne le même mot si on l'écrit à l'envers… ex. laval, radar ; non), contient un pangramme (qui contient toutes les lettres de l'alphabet… ex. « apportez un vieux whisky à ce petit juge blond qui fume la pipe »).*

Question 3.2. *Écrivez dans le programme principale, les différents tests : dès que l'on trouve un test vrai, on sort.*

Question 3.3. *Modifiez votre programme pour ce qu'il s'agit d'un entier alors on l'affiche à l'écran en binaire.*