Unité : Systèmes séquentiels avancés (SSA)

Architecture des processeurs

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

Reconfigurable & Embedded Digital Systems

Etienne Messerli
& collègues institut REDS
Institut REDS, HEIG-VD

30 avril 2013

Contenu de la présentation

- Introduction
  - ✓Principe de l'architecture Von Neumann
- Processeur simple: MU0 (Furber §1.3, pp. 7-13)
  - ✓conception du processeur simple MU0
  - ✓étude fonctionnement MU0
  - ✓ajout d'instruction au MU0

Architecture des processeurs

Introduction

Ordinateur

Système UNIVERSEL de traitement de l'information "binaire"

- Utilisé pour :
  - ✓Traitement de texte
  - ✓Calcul complexe
  - ✓Commande de machines
  - ✓…..

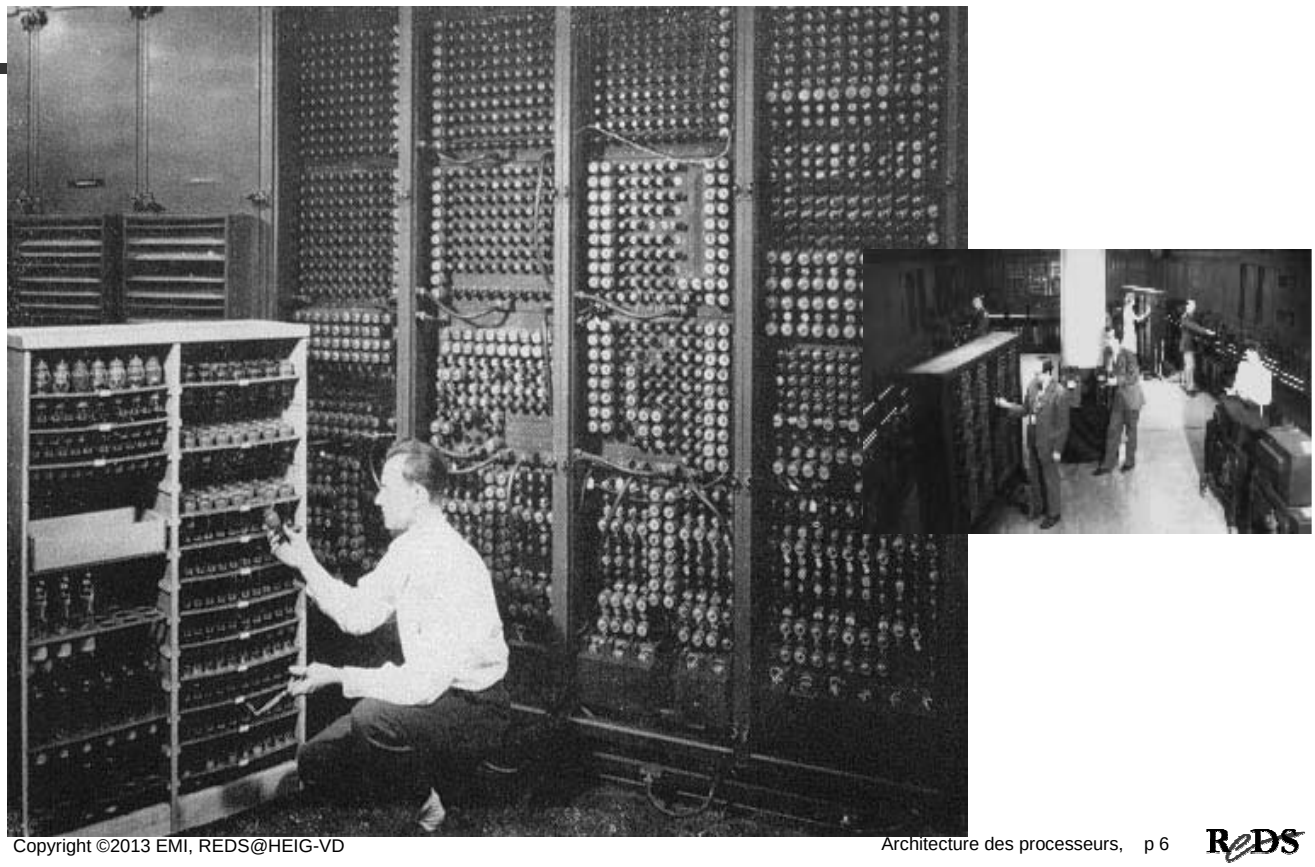
Terminologie

- Calculette simple ⇨ opérations fixes
- Calculatrice type HP48
- Ordinateur personnel PC
- Maxi ordinateur

  } ⇨ ordinateurs de puissances différentes

- Microprocesseur — partie d'un ordinateur
- Micro-contrôleur — système à processeur intégré dans une seule puce

Eniac, école de Moore, Pennsylvanie, 1946



Historique

Année	Nom	Taille m3	Puissance W	Performances (add/s)	Mém. Ko	Prix $	Rapport Perfor-/Prix	Prix $ (1991)	Rapport Perfor-/Prix
1951	UNIVAC	28,317	124'500	1'900	48	1'000'000	1	4'533'607	1
1964	IBM S360	1,699	10'000	500'000	64	1'000'000	263	3'756'502	318
1965	PDP-8	0,227	500	330'000	4	16'000	10'855	59'947	13'135
1976	Cray-1	1,642	60'000	166'000'000	32'768	4'000'000	2'842	7'675'591	51'604
1981	IBM PC	0,0283	150	240'000	256	3'000	42'105	3'702	
1991	HP 9000	0,0566	500	50'000'000	16'384	7'400	3'556'188	7'400	16'122'356
2000	PC	0,0250	150	≈1'000'000'000	256'000	1'000	≈526'315'789		

Référence [2]: Organisation et conception des ordinateurs
Patterson et Hennessy, Dunod

Site internet : http://histoire.info.online.fr/

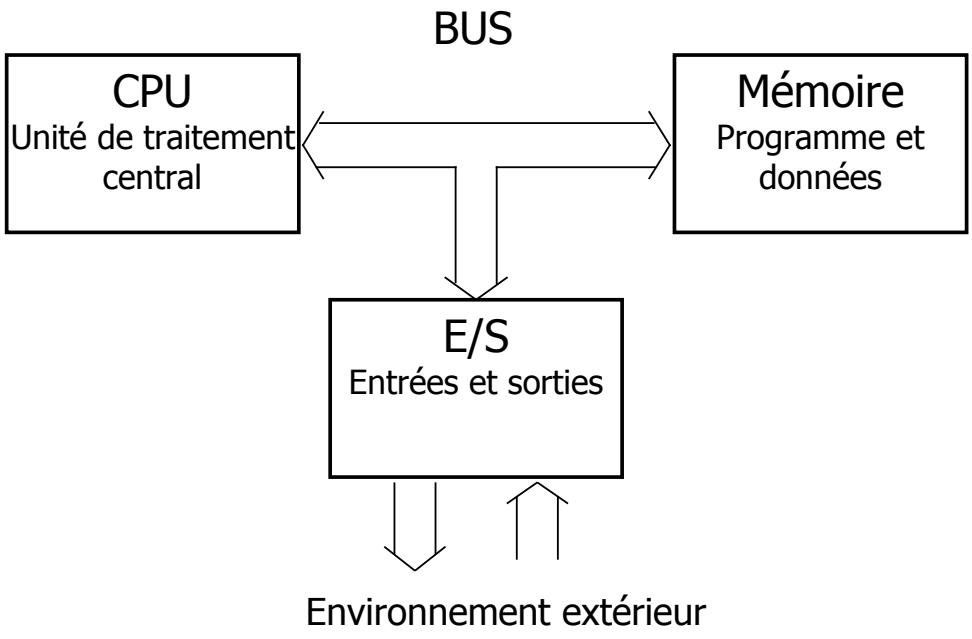
Architecture Von Neumann

- John Von Neumann, Princeton, 1946
  - ✓imagine de stocker le programme en mémoire
  - ✓pas de distinction entre la mémoire pour le programme et celle pour les données
  - ✓un seul ensemble de bus entre : CPU-Mémoire-E/S

Architecture Harvard

- Howard Aiken, université de Harvard, 1946
  - ✓Utilise un programme stocké en mémoire
  - ✓Programme et données stockés dans des mémoires séparées
  - ✓Dispose de 2 ensembles de bus, soit :
    - →un ensemble : CPU-Mémoire programme
    - →un ensemble : CPU-Mémoire donnée-E/S

Principe architecture von Neumann



Unités fonctionnelles

- CPU *Central Processing Unit*
  - ✓unité de traitement centrale
- Mémoire
  - ✓Stockage du programme et des données
- Entrées & sorties
  - ✓Connexion avec l'environnement extérieur
- Bus
  - ✓Interconnexions entre les 3 unités

CPU ...

- Maître du système
- Son rôle:
  - ✓chercher une instruction dans la mémoire
  - ✓décoder cette instruction
  - ✓exécuter l'instruction
    - →transférer des données
    - →réaliser une opération, un calcul
    - →....
  - ✓calculer l'adresse de la prochaine instruction

Instructions du CPU

- Les instructions du CPU peuvent être classées en trois catégories :
  - ✓instructions de calcul (arithmétique et logique)
  - ✓instructions de transfert de donnée
    - →interne ⇔ interne, interne ⇔ externe
  - ✓instructions de branchement (saut)

La mémoire

- Permet de stocker le programme et les données
- Plusieurs types
  - ✓mémoires volatiles (RAM, SRAM, DRAM, ..)
  - ✓mémoires non-volatiles (EPROM,EEPROM,..)
  - ✓mémoires de masses (disque dur, bande, …)
    - →celles-ci seront considérées comme des entrées/sorties

Les entrées & sortie

- Connexion avec l'environnement extérieur
  - ✓Clavier
  - ✓Ecran
  - ✓Convertisseur A/D et D/A
  - ✓Mémoires de masse (disque dur, clé USB, …)
  - ✓….

Indispensable pour le fonctionnement de l'ensemble

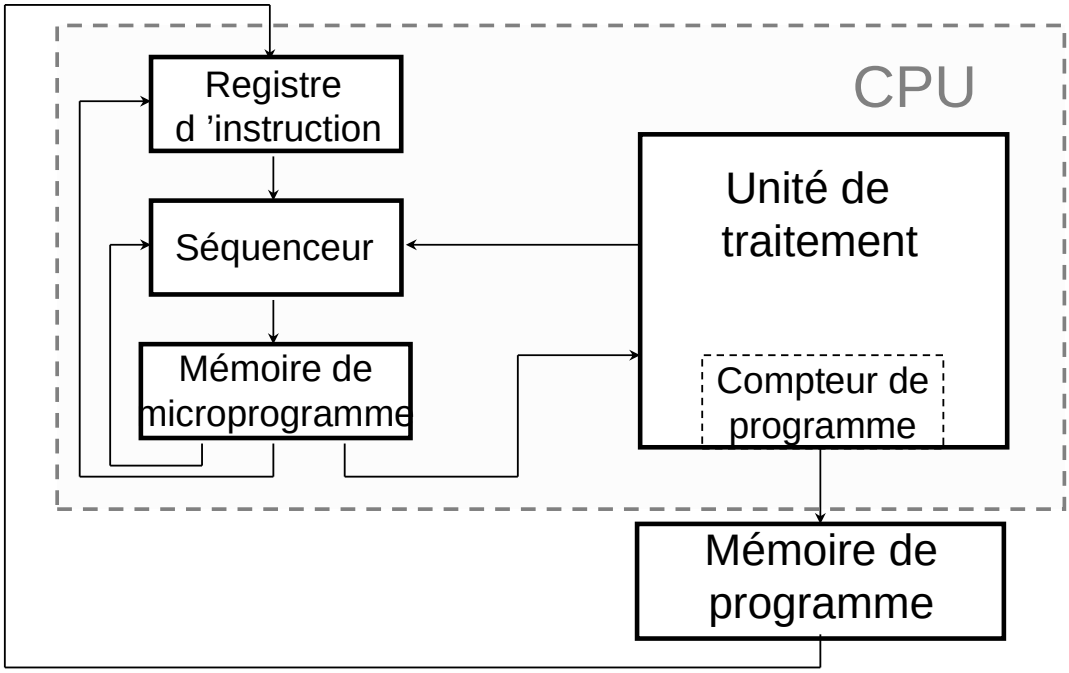
Les Bus

- Interconnecte les 3 unités principales
  - ✓Commun à tous
    ⇒ indispensable de définir un plan d'adressage
  - ✓Le CPU est le maître
  - ✓Composé de 3 parties :
    - →Bus d'adresse
    - →Bus de données
    - →Bus de contrôle

Conception CPU

- Rôle du CPU:
  - ✓tâche principalement séquentielle (voir p 11)
- CPU ⇒ MSS complexe, 2 unités :
  - ✓unité de séquencement (UC)
  - ✓unité d'exécution (UT)

CPU: schéma bloc



Fonctionnement du CPU

- 2 niveaux de programmation !
  - ✓Exécution d'une instruction du programme
  - ✓nécessite :

    l'exécution de plusieurs micro-instructions

    (un microprogramme par instruction)

Fonctionnement séquentiel du CPU

Structure CPU, MSS complexe

- Partie commande CPU (UC)
  - ✓Registre d'instruction IR
    - →souvent avec un décodeur d'instruction
  - ✓Séquenceur (UC micro-programmée ou câblée)
  - ✓Mémoire de micro-programme (si UC micro-programmée)
- Partie traitement CPU (UT)
  - ✓Unité de traitement composée de
    - →Compteur de programme PC
    - →ALU, Accumulateur
    - →Registres
    - →…..

Exécution d'une instruction

- Déterminer l'adresse du micro-programme
  Méthodes possibles:
  - ✓Directement code instruction
    →table de saut indirect
  - ✓Code instruction + quelques bits de poids faible
    →taille fixe pour tous les micro-programmes
  - ✓Utiliser un transcodeur : mémoire de *mapping*
    →taille micro-programme variable (souple)
    →même adresse de début pour différentes instructions

Exercice 1

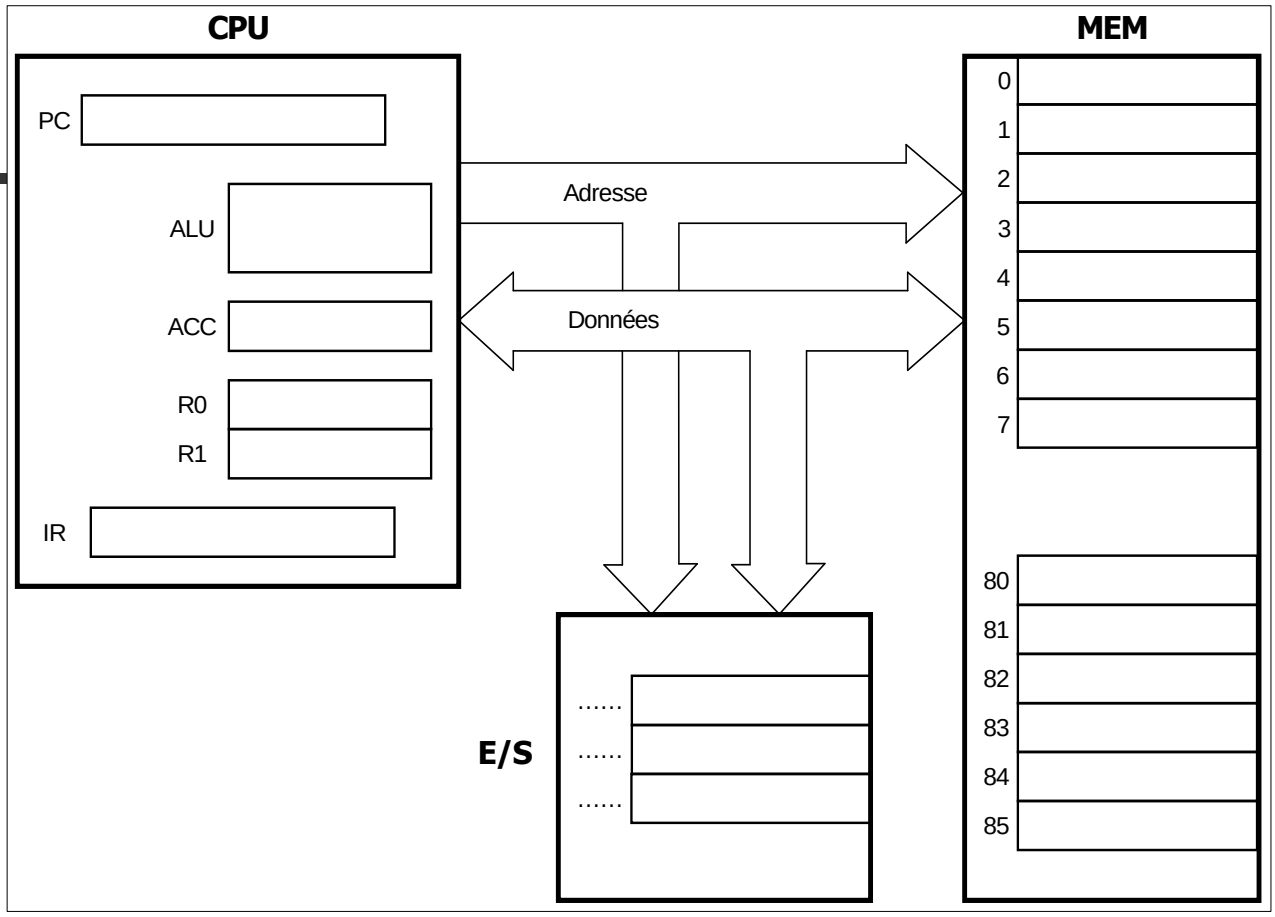
Etudier le fonctionnement interne des instructions d'un processeur

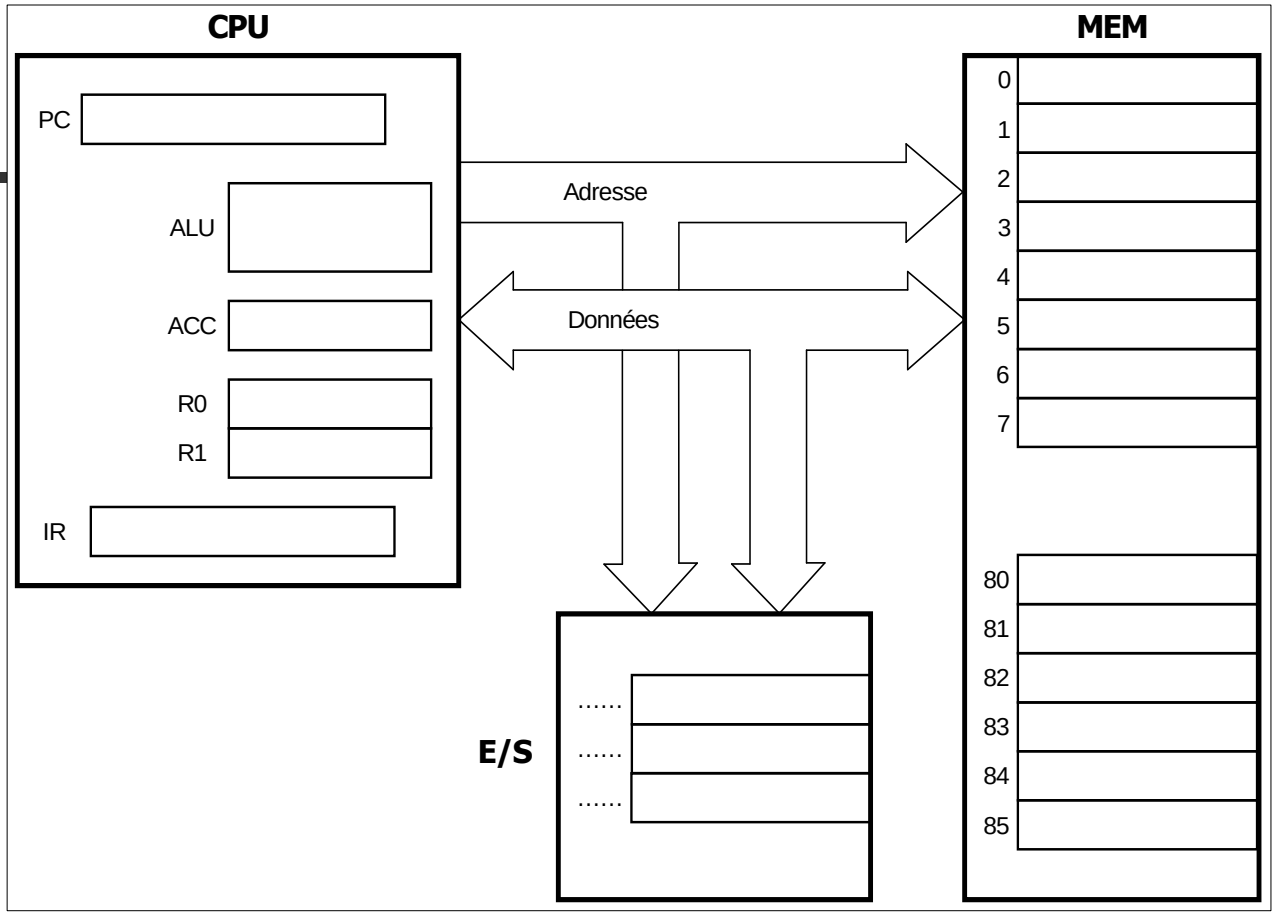
- Pour chaque instruction, indiquez les informations circulant sur les différents chemins de données (*datapath*)

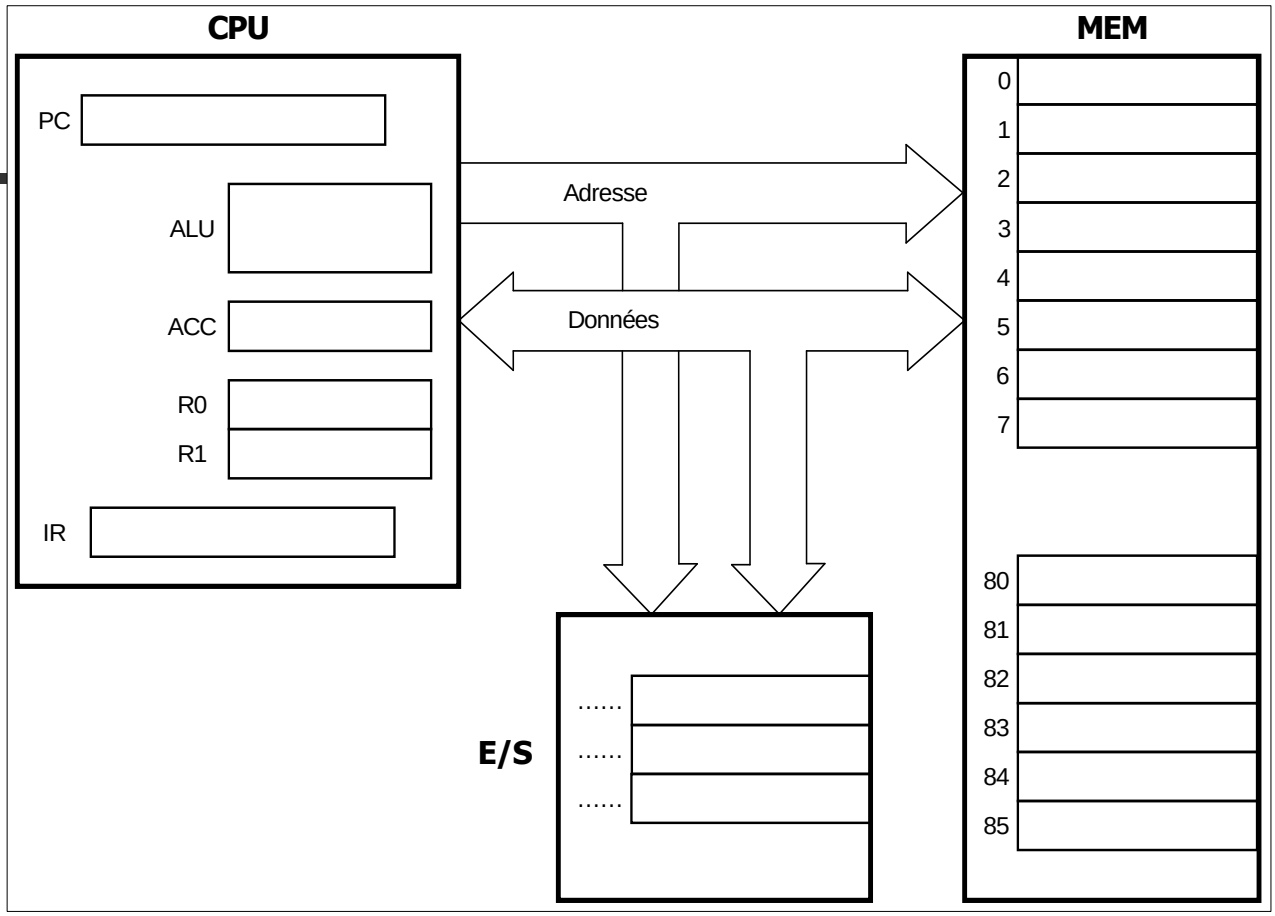
Exercice 1: énoncé

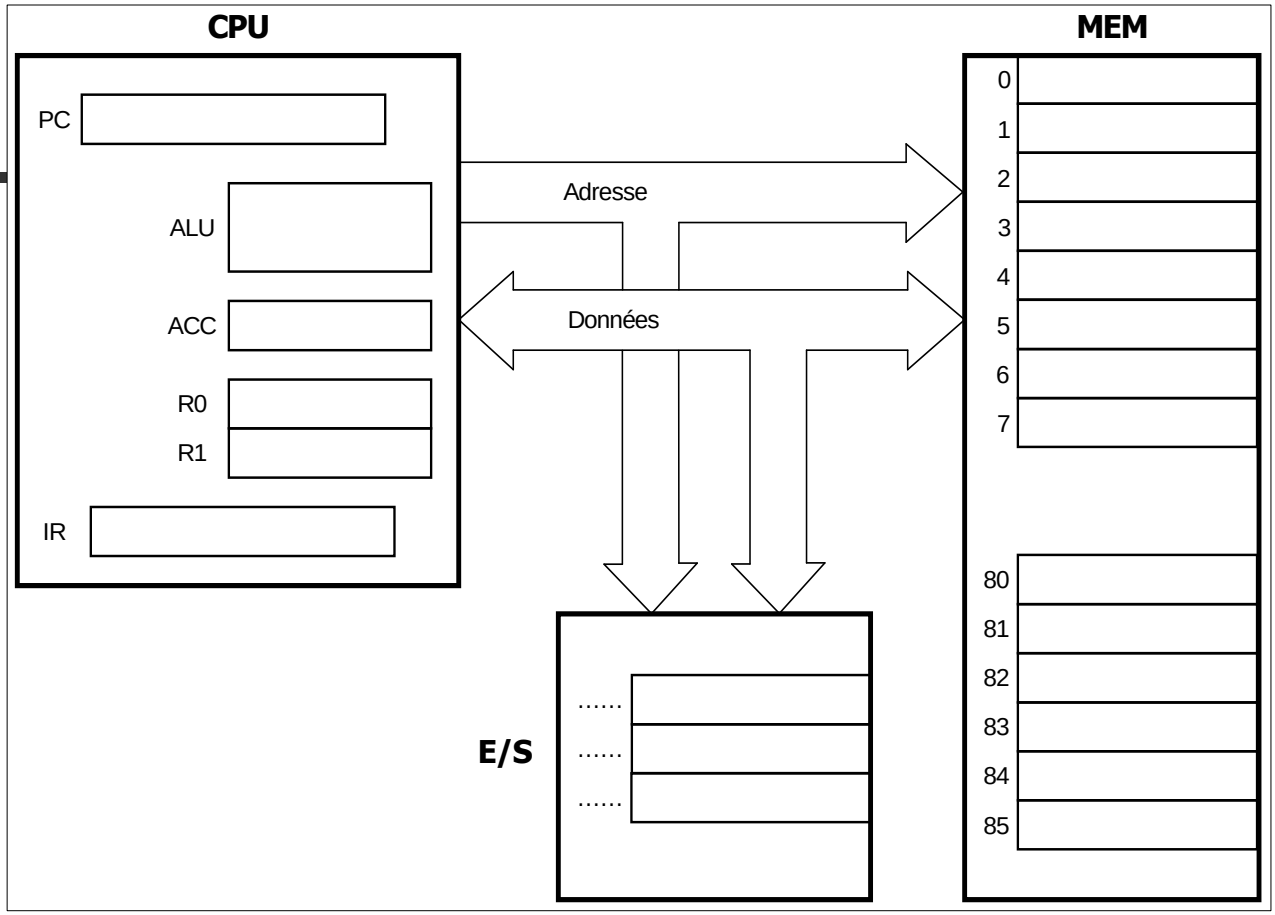
- La mémoire contient le programme suivant :

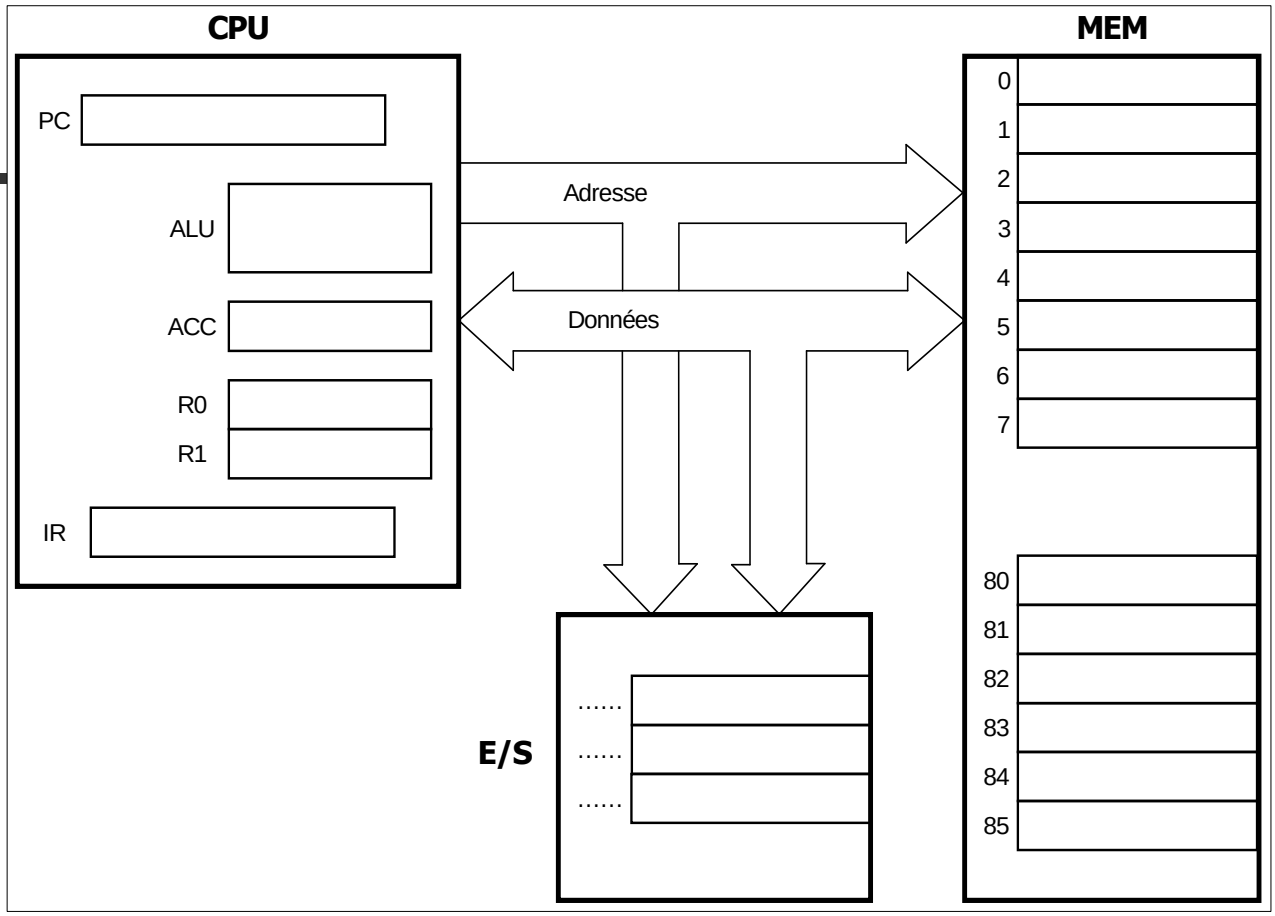
Adresse	Instruction		Description
0	Load	ACC, #25	; ACC = 25 (valeur immédiate)
1	Load	R0, ACC	; R0 = ACC
2	ADD	ACC, R0	; ACC = ACC + R0
3	Load	80,ACC	; Mem[80] = ACC (adressage direct)
4	JUMP	1	; PC = 1




CPU
MEM
PC
ALU
ACC
R0
R1
IR
Adresse
Données
E/S
0
1
2
3
4
5
6
7
80
81
82
83
84
85


CPU
MEM
PC
ALU
ACC
R0
R1
IR
Adresse
Données
E/S
0
1
2
3
4
5
6
7
80
81
82
83
84
85


CPU
MEM
PC
ALU
ACC
R0
R1
IR
Adresse
Données
E/S
.....
.....
.....
0
1
2
3
4
5
6
7
80
81
82
83
84
85


CPU
MEM
PC
ALU
ACC
R0
R1
IR
Adresse
Données
E/S
.....
.....
.....
0
1
2
3
4
5
6
7
80
81
82
83
84
85

Architecture des processeurs

Le processeur simple MU0

(§1.3, pp.7-13, Furber)

Didacticiel : ARM System Design

- Sélectionner le menu :
  - ✓Background and Introduction to the ARM
- Choisir le sous-menu :
  - ✓A Simple Processor

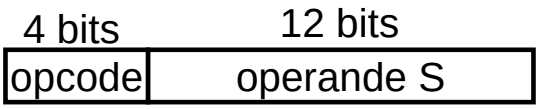
    Explications sur la conception du processeur MU0

MU0: processeur simple

- Approche par un processeur simple:
  - ✓Eléments de base:
    - →PC: compteur de programme
    - →ACC: accumulateur
    - →ALU: unité arithmétique et logique
    - →IR: registre d'instruction
    - →unité de séquencement
  - ✓machine 16 bits (données) avec 12 bits d'espace adressable

Jeu et format d'instructions

- Format des instructions :



- Sauf indication, le PC est incrémenté

Instruction		Opcode	Effect
LDA	S	0000	ACC := mem_{16}[S]
STO	S	0001	mem_{16}[S] := ACC
ADD	S	0010	ACC := ACC + mem_{16}[S]
SUB	S	0011	ACC := ACC - mem_{16}[S]
JMP	S	0100	PC := S
JGE	S	0101	if ACC >= 0 then PC := S
JNE	S	0110	if ACC /= 0 then PC := S
STP		0111	stop

Méthodologie

- Etude du processeur par l'étude de sa conception
  - ✓**datapath**: éléments qui transportent, stockent ou traitent plusieurs bits en parallèle;

    représentation: niveau RTL (register transfer level)
  - ✓**logique de contrôle**: éléments qui ne font pas partie du "datapath"

    représentation par machine d'état

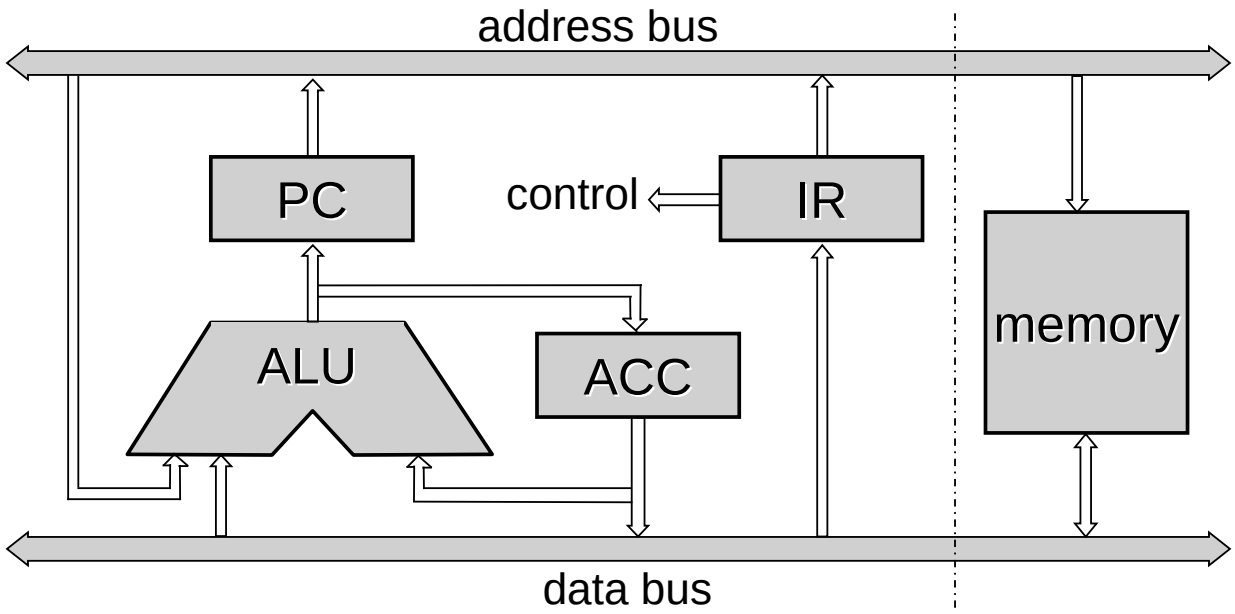
Datapath de MU0

- Nombreuses possibilités d'interconnexion
- Principe choisi ici comme guide:

  accès mémoire = facteur limitatif

  i.e.: nombre de cycles de chaque instruction = nombre d'accès mémoire
- Donc, on cherchera un datapath tel que:
  - ✓Instructions qui utilisent $mem_{16}[S]$: 2 cycles
  - ✓Instructions qui utilisent S directement: 1 cycle

Exemple de "datapath"



Exercice 2 : analyse du schéma "datapath"

- Questions sur le schéma "datapath"
  - ✓Indiquer la largeur des bus du schéma précédent
  - ✓Que doit-il se passer au démarrage ?
  - ✓Comment le PC est-il incrémenté ?
  - ✓Combien de cycles sont nécessaires pour chaque instruction du processeur MU0 ?
  - ✓Indiquer les modes d'adressage du jeux d'instructions du MU0 ?

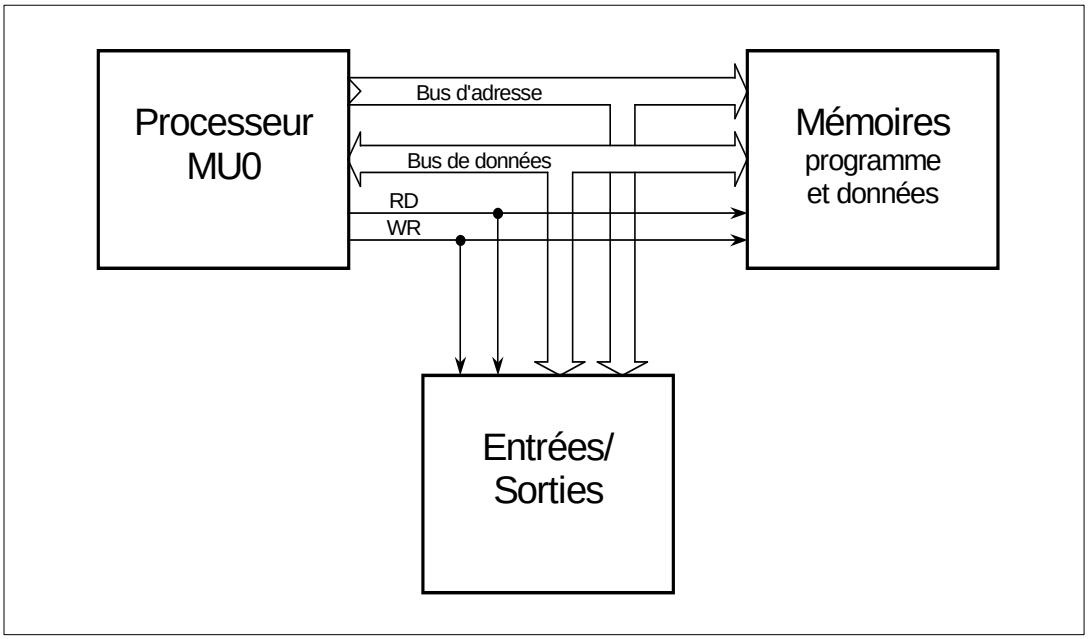
Opérations

- Convention: instruction dans IR exécutée, puis IR chargé avec instruction suivante
- Etapes, dans l'ordre:
  - ✓Selon l'instruction:
    - → Soit un opérande est lu de la mémoire, combiné avec l'ACC dans l'ALU et le résultat est écrit dans l'ACC
    - → Soit le contenu de l'ACC est placé en mémoire
    - → Soit cette étape n'a pas lieu
  - ✓Pour toutes les instruction:
    - → Chargement de la prochaine instruction à exécuter (FETCH) (adresse: du PC ou de l'instruction, via l'ALU)

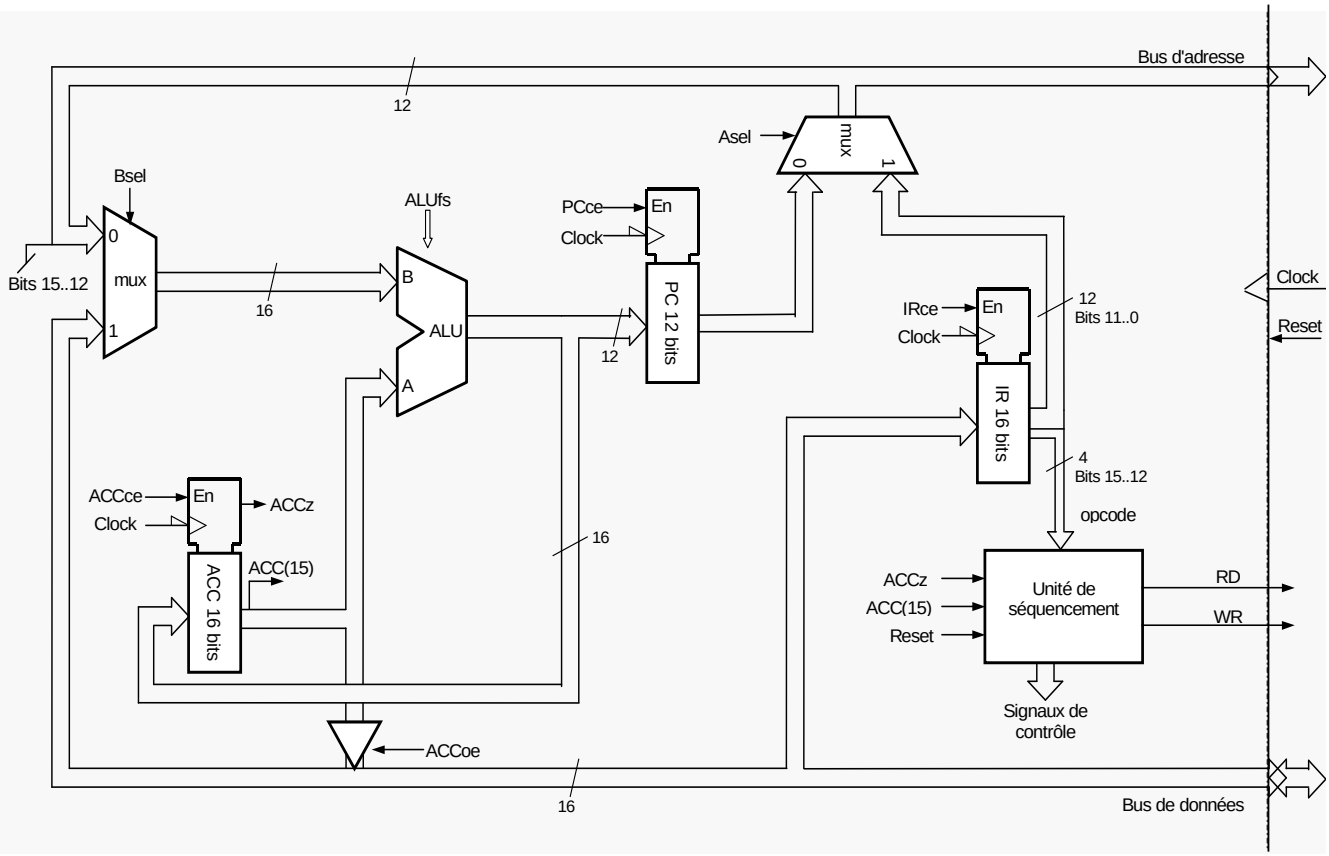
Initialisation

- Le processeur doit démarrer dans un état connu
  - ✓entrée RESET ⇒ forcer une adresse de départ connue (PC = 0x000)
  - ✓fonctionnement dans le cas du MU0:
    lors du RESET ⇒ sortie de l'ALU = 000_{16},
    valeur chargée dans le PC au prochain flanc d'horloge
  - ✓dans le cas du MU0 : RESET synchrone !!

Architecture système à processeur MU0



Dia volontairement laissé vide



Décodeur d'instruction du MU0

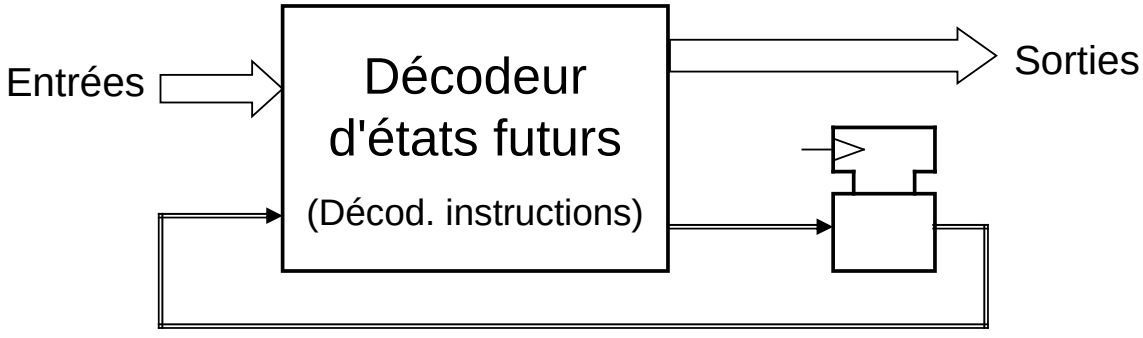
Entrées						Sorties									
Instr	Opcode	Reset	Cycle	ACCz	ACC15	Asel	Bsel	ACCce	PCce	IRce	ACCoe	ALUfs	RD	WR	Cycle +
(Reset)	xxxx	1	x	x	x	0	0	1	1	1	0	=0	1	0	0
LDA S	0000	0	0	x	x	1	1	1	0	0	0	=B	1	0	1
	0000	0	1	x	x	0	0	0	1	1	0	B+1	1	0	0
STO S	0001	0	0	x	x	1	x	0	0	0	1	x	0	1	1
	0001	0	1	x	x	0	0	0	1	1	0	B+1	1	0	0
ADD S	0010	0	0	x	x	1	1	1	0	0	0	A+B	1	0	1
	0010	0	1	x	x	0	0	0	1	1	0	B+1	1	0	0
SUB S	0011	0	0	x	x	1	1	1	0	0	0	A-B	1	0	1
	0011	0	1	x	x	0	0	0	1	1	0	B+1	1	0	0
JMP S	0100	0	x	x	x	1	0	0	1	1	0	B+1	1	0	0
JGE S	0101	0	x	x	0	1	0	0	1	1	0	B+1	1	0	0
	0101	0	x	x	1	0	0	0	1	1	0	B+1	1	0	0
JNE S	0110	0	x	0	x	1	0	0	1	1	0	B+1	1	0	0
	0110	0	x	1	x	0	0	0	1	1	0	B+1	1	0	0
STP	0111	0	x	x	x	1	x	0	0	0	0	x	0	0	0

Exercice 3 : questions sur le MU0

- Comment le Reset agit-il sur les différents blocs?
- Quel contrainte temporelle doit respecter le Reset ?
- Quel est le contenu du registre IR lorsque le signal Reset se désactive ?
- Comment l'incrémentation du PC est-elle obtenue ? Qu'elle remarque pouvez-vous faire ?
- Quel est la fonction du signal Cycle ?
- Concevoir l'unité de séquencement.
  Voir structure page suivante.

… exercice 3:

- Concevoir l'unité de séquencement (unité de commande du MU0)
  Celle-ci se décompose comme suit :



Exercice 4 :

- Faire le chronogramme de l'exécution du programme suivant après un Reset :

Adr.	Contenu MEM	
0	LDA	0xC00
1	SUB	0xC01
2	STO	0xC00
3	JNE	0
4	STP	
...		

Adr.	Contenu MEM
0xC00	0x0009
0xC01	0x0001
...	

Exercices 5 :

- Indiquer toutes les variantes possible pour fournir l'adresse d'une instruction ?
- Comment fonctionne l'instruction JMP S ?
- Comment l'opération de soustraction est-elle réalisée dans l'ALU ?

Etude fonctionnement du MU0

- Menu démarrer
  - ✓Labo systèmes numériques
    - →Embedded => MUO
    - →lancer le programme SimMU0.tcl
- Editer dans le simulateur MU0 le programme de l'exercice 4
- Simuler au niveau programmeur le programme
- Choisir le mode "Tutorial"
  - ✓étudier le fonctionnement interne du MU0 lors de l'exécution du programme

Exercice 6, 7 et 8 :

6. Sur le processeur MU0, étendre l'espace d'adressage à 20 bits à l'aide d'un registre de base de 16 bits
7. Ajouter au processeur MU0 la possibilité d'appeler des sous-routines.
   - ✓Implanter les fonctions d'appel (CALL) et de retour (RET) au sous-programme. La pile sera prévue avec un seul niveau (pas d'appel imbriqué de sous-programme)
8. Ajouter au processeur MU0 le mode d'adressage indexé (voir exercice 1.2 p. 33, Furber)

Bibliographie

[1] ARM, system-on-chip, 2000 (second edition),
Steve Furber, Addison-Wesley

[2] Organisation et conception des ordinateurs
Patterson & Hennessy, Dunod

[3]

[4] Systèmes à microprocesseur, ch. XI architecture
Serge Boada, EIVD