

unité de commande

régit l'ordinateur via quelques signaux de contrôle

en se basant sur

- les instructions à exécuter
- la nature du résultat des opérations sur l'ALU

sans prendre en considération

- les données traitées
- les résultats produits

plan

- micro-opérations
 - fetch
 - indirect
 - interrupt
 - execute
- contrôle du processeur
 - interactions UC-extérieur
 - organisation
- implantation de l'UC
 - implantation matérielle
 - implantation micro-programmée

micro-opérations

étapes élémentaires composant une phase du cycle d'instruction

opérations fonctionnelles atomiques du processeur

pouvant s'exécuter en parallèle à condition

- de respecter la dépendance des données
- d'éviter les conflits

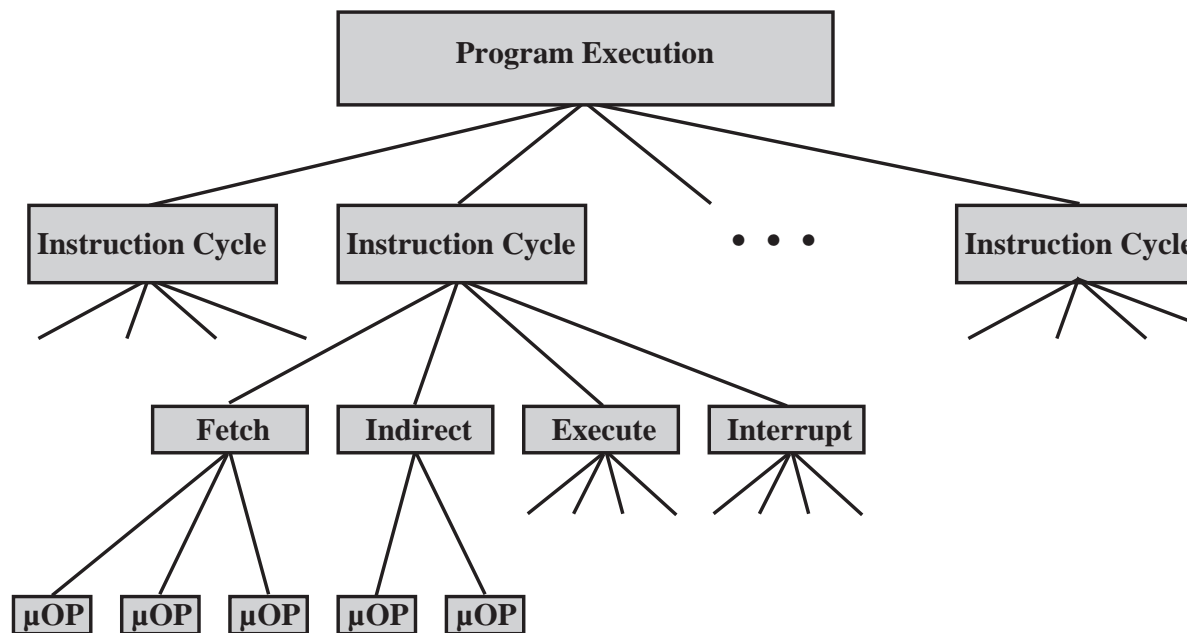


Figure 14.1 Constituent Elements of a Program Execution

exemple

CPU comprenant les registres

- PC
- IR
- MAR
- MBR

exemple

cycle de l'instruction comprenant les phases

- fetch
- indirect
- execute
- interrupt

exemple

on appelle

T	taille d'une instruction
<i>ti</i>	unités de temps de valeur égale
IR(adresse)	champs adresse du registre IR

fetch

3 étapes comprenant 4 micro-opérations

t1 $MAR \leftarrow PC$

t2 $MBR \leftarrow \text{mémoire}$

$PC \leftarrow PC + T$

t3 $IR \leftarrow MBR$

indirect

format 1 adresse

contenu de IR directement affecté pour contenir une adresse directe

t1 $MAR \leftarrow IR(\text{adresse})$

t2 $MBR \leftarrow \text{mémoire}$

t3 $IR(\text{adresse}) \leftarrow MBR(\text{adresse})$

interrupt

t1 MBR \leftarrow PC

t2 MAR \leftarrow adresse de sauvegarde

PC \leftarrow adresse de routine d'interruption

t3 mémoire \leftarrow MBR

execute

n différents opcodes nécessitent n différentes séquences de micro-opérations

exemple : ADD R1, X

t1 MAR \leftarrow IR(adresse)

t2 MBR \leftarrow mémoire

t3 R1 \leftarrow R1 + MBR

execute

exemple : ISZ X

- incrémente X de 1
- saute l'instruction suivante si le résultat est 0

t1 $MAR \leftarrow IR(\text{adresse})$

t2 $MBR \leftarrow \text{mémoire}$

t3 $MBR \leftarrow MBR + 1$

t4 $\text{mémoire} \leftarrow MBR$

si $MBR = 0$ alors $PC \leftarrow PC + T$

execute

BSA X

- sauvegarde l'adresse de l'instruction suivante dans X
- poursuit l'exécution par l'instruction située à l'adresse $X + T$

t1 $MAR \leftarrow IR(\text{adresse})$

$MBR \leftarrow PC$

t2 $PC \leftarrow IR(\text{adresse})$

mémoire $\leftarrow MBR$

t3 $PC \leftarrow PC + T$

cycle de l'instruction

le registre ICC (Instruction Cycle Code)

- contient l'état du processeur
- est modifié en fin de chaque cycle

exemple

ICC	état
00	fetch
01	indirect
10	execute
11	interrupt

contrôle du processeur

l'UC exécute 2 tâches basiques

- le séquençement des micro-opérations
- l'exécution des micro-opérations

nature des micro-opérations

- transfert de données entre registres
- transfert de données d'un registre vers une interface externe (e.g., bus système)
- transfert de données d'une interface externe vers un registre
- opération sur l'ALU en utilisant des registres pour les opérandes sources et le résultat

interactions UC-extérieur

entrées de l'UC

- horloge : une micro-opération prend un top d'horloge
- registre IR
 - identification du mode d'adressage
 - détermination du cycle execute

interactions UC-extérieur

entrées de l'UC

- registre PSW
 - détermination du statut du processeur
 - connaissance de la dernière opération de l'ALU
- signaux de contrôle du bus système
 - détection des interruptions
 - acquittements des unités externes

interactions UC-extérieur

sorties de l'UC

- signaux de contrôle internes au processeur
 - déclenchant des transferts entre registres
 - déclenchant des opérations sur l'ALU
- signaux de contrôle transportés par le bus système
 - à destination de la mémoire
 - à destination des modules d'E/S

exemple

signaux de contrôle envoyés pour un fetch

1. chargement de MAR par PC
2. simultanément
 - placement de MAR sur le bus d'adresse
 - activation d'une ligne lecture mémoire du bus de contrôle
 - chargement de MBR par le bus de données
 - incrément de PC

exemple

signaux de contrôle envoyés pour un fetch (suite)

3. un signal autorisant le chargement de IR par MBR

puis analyse de IR pour connaître le cycle suivant

exemple

signaux de contrôle envoyés pour un fetch (suite)

3. un signal autorisant le chargement de IR par MBR

puis analyse de IR pour connaître le cycle suivant

les cycles indirect et interrupt fonctionnent de manière similaire

cycle	timing	signal de contrôle
fetch	t1 : $MAR \leftarrow PC$	C_2
	t2 : $MBR \leftarrow \text{mémoire}$	C_5, C_r
	$PC \leftarrow PC + T$	
	t3 : $IR \leftarrow MBR$	C_4
indirect	t1 : $MAR \leftarrow IR(\text{adresse})$	C_8
	t2 : $MBR \leftarrow \text{mémoire}$	C_5, C_r
	t3 : $IR(\text{adresse}) \leftarrow MBR(\text{adresse})$	C_4
interrupt	t1 : $MBR \leftarrow PC$	C_1
	t2 : $MAR \leftarrow \text{adresse de sauvegarde}$	
	$PC \leftarrow \text{adresse de routine d'interruption}$	
	t3 : $\text{mémoire} \leftarrow MBR$	C_{12}, C_w

bus interne

dédié à la circulation interne des différentes informations

- données
- instructions
- adresses

relie tous les registres internes et l'ALU

accès contrôlé par les signaux de contrôle émis par l'UC

implantation de l'unité de commande

2 catégories

- implantation matérielle
- implantation micro-programmée

implantation matérielle

signaux de contrôle élaborés par un circuit combinatoire

- décodage de l'opcode en séquence de signaux de contrôle
- génération des t_i via l'horloge

implantation matérielle

correspond aux équations booléennes des signaux de sorties en fonction

- du cycle (ICC)
- du t_i (horloge)
- de l'opcode

exemple

ICC		
P	Q	cycle
0	0	fetch
0	1	indirect
1	0	execute
1	1	interrupt

$$C_5 = \overline{P}\overline{Q}t_2 + \overline{P}Qt_2 + P\overline{Q}(LOAD + ADD + AND)t_2$$

si LOAD, ADD et AND demandent une lecture dans la mémoire

implantation micro-programmée

implantation

matérielle

microprogrammée

nombre d'équations

nombre d'équations

booléennes petit

booléennes grand

RISC

CISC

micro-instructions

instructions déclenchant les micro-opérations

- décomposées en champs
- rangées dans une mémoire à une adresse précise

micro-programme = programme à base de
micro-instructions

champs des micro-instructions

- mot de contrôle correspondant à l'activation des signaux
 - 1 bit pour chaque ligne de contrôle interne
 - 1 bit pour chaque ligne de contrôle du bus de contrôle
- condition de branchement
- adresse de micro-instruction

interprétation

1. positionner les signaux de contrôle en fonction du mot de contrôle
2. si la condition indiquée par les bits de condition est fausse
 - exécuter la micro-instruction à l'adresse suivante
 - sinon
 - exécuter la micro-instruction dont l'adresse est mentionnée dans le champs adresse

lire une instruction = exécuter cette instruction

rangement

organisation en séquences dans une mémoire de contrôle

chaque séquence définit une routine correspondant à

- un sous-cycle du cycle d'instruction
- un opcode pour le cycle exécute

mémoire de contrôle

⋮	
⋮	
saut vers indirect ou execute	routine du cycle fetch
⋮	
⋮	
saut vers execute	routine du cycle indirect
⋮	
⋮	
saut vers fetch	routine du cycle interrupt
saut vers routine d'opcode	routine du cycle execute
⋮	

mémoire de contrôle

⋮ ⋮ saut vers fetch ou interrupt	routine ADD
⋮ ⋮ saut vers fetch ou interrupt	routine AND
⋮	⋮
⋮ ⋮ saut vers fetch ou interrupt	routine SUB

réalisation

implanter l'UC = exécuter le micro-programme

l'implantation nécessite

- la mémoire de contrôle (CM)
- un registre contenant l'adresse de la prochaine micro-instruction à lire (CAR)
- un registre contenant la micro-instruction (CBR)
- un séquenceur
 - chargeant le registre d'adresse
 - envoyant des ordres de lecture à la mémoire de contrôle

fonctionnement de l'UC

en un top d'horloge

1. le séquenceur envoie une commande de lecture à CM
2. la micro-instruction dont l'adresse est spécifiée par CAR est chargée dans CBR
3. génération des signaux de contrôle
4. CBR donne l'adresse de la micro-instruction suivante au séquenceur

fonctionnement de l'UC

5. le séquenceur charge une nouvelle adresse dans CAR en fonction de
 - l'adresse délivrée par CBR
 - des conditions provenant de l'ALU

format de micro-instructions

la nouvelle adresse peut être

- l'adresse courante + 1
- l'adresse de CBR
- l'adresse d'une routine correspondant à l'opcode de IR

les branchements représentent 1/4 à 1/3 des micro-instructions

format de micro-instructions

différents formats pour l'adresse

- 2 adresses
- 1 adresse
- variable

exercice corrigé

construction d'une CPU comprenant

- le registre accumulateur (ACC)
- le registre compteur ordinal (CO)
- le registre instruction qui contient l'instruction courante (RI)

exercice corrigé

- trois registres associés à la mémoire :
 - RAD contenant une adresse mémoire
 - RDE contenant un mot à écrire en mémoire
 - RDL contenant un mot lu de la mémoire

mémoire de 2^{13} mots de 16 bits

instructions

8 instructions

- 4 instructions de manipulation des données
- 4 instructions pour le contrôle de programme

format unique comprenant 2 champs

- opcode sur 3 bits
- un champs adresse sur 13 bits

instructions

LDA	000	ADR	$ACC \leftarrow (ADR)$
STA	001	ADR	$(ADR) \leftarrow ACC$
ADD	010	ADR	$ACC \leftarrow ACC + (ADR)$
NAND	011	ADR	$ACC \leftarrow ACC \text{ Nand } (ADR)$
BR	100	ADR	branchement à ADR
BRZ	101	ADR	branch. si $ACC=0$
CALL	110	ADR	appel ADR avec retour ACC
RETURN	111		branchement à ACC

signaux de contrôle

12 signaux

- 3 associés à une opération mémoire
- 5 associés à l'ALU
- 4 associés aux registres RI, CO et ACC

signaux de contrôle

- CMD 1 : chargement de RAD à partir de l'information contenue dans CO
- CMD 2 : chargement de RAD à partir de l'information contenue dans RI
- CMD 3 : chargement de RDE à partir de l'information se trouvant à la sortie de l'ALU (permet ainsi d'écrire dans la mémoire)

signaux de contrôle

- CMD 4 : type de l'opération ALU (0 pour l'addition, 1 pour le NAND)
- CMD 5 : sélectionne CO comme premier opérande de l'ALU
- CMD 6 : sélectionne le champ ADR de RI comme premier opérande de l'ALU

signaux de contrôle

- CMD 7 : sélectionne RDL comme premier opérande de l'ALU
- CMD 8 : sélectionne ACC comme deuxième opérande de l'ALU
- CMD 9 : chargement de ACC par la sortie R de l'ALU

signaux de contrôle

- CMD 10 : chargement de RI par la sortie R de l'ALU
- CMD 11 : chargement de CO par la sortie R de l'ALU
- CMD 12 : commande l'incrémentatation de CO

micro-instructions

2 formats

0	(12)	(8)	(4)	ALUR	ALUA	mémoire	format 1
9	8	7	6	5 4	3 2	1 0	

utilisé pour générer les signaux de contrôle

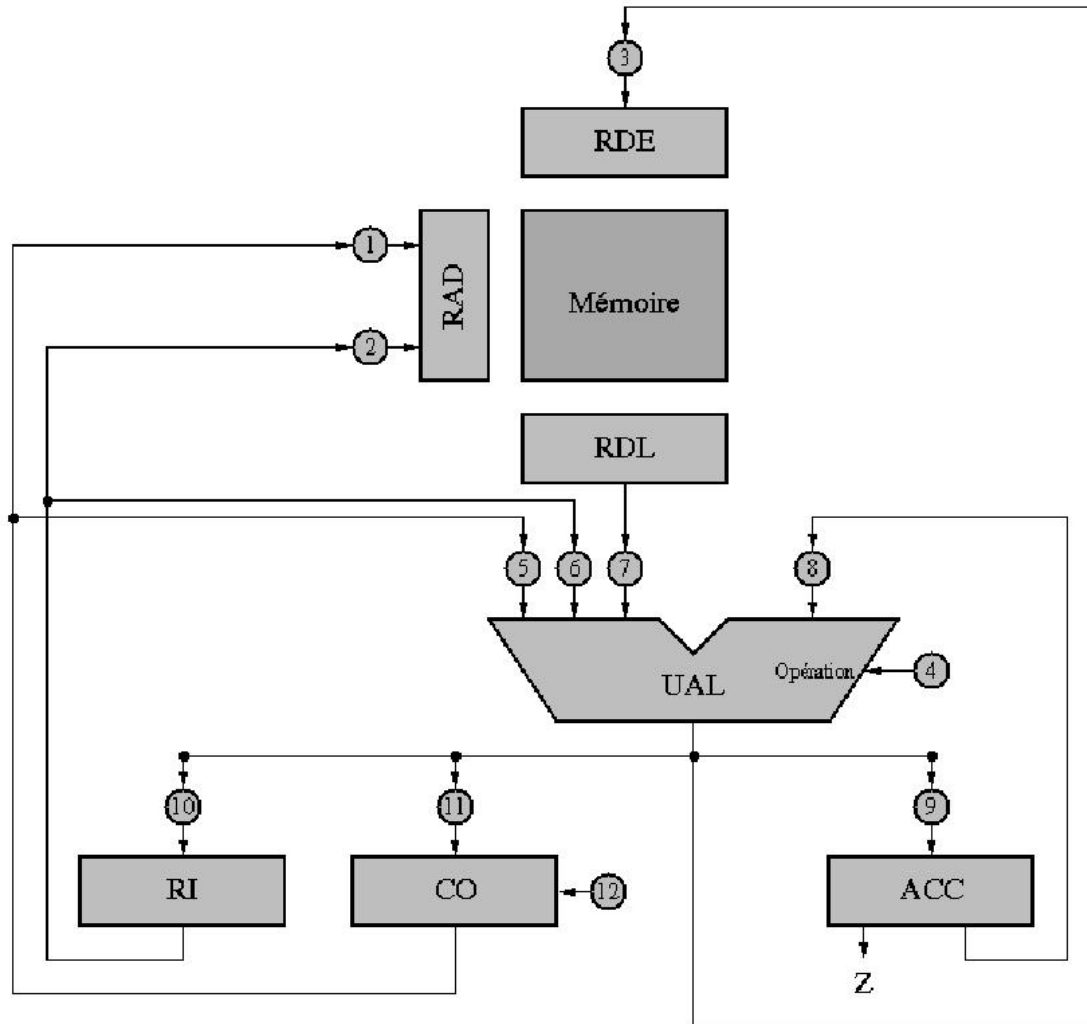
1	COND	X	adresse microinstruction	format 2
9	8 7	6	5 4 3 2 1 0	

utilisé pour les branchements dans le micro-programme

champs

	ALUR	ALUA	mémoire	cond
00	NOP	NOP	NOP	
01	(9)	(5)	(1)	microPC \leftarrow '1 COP_2 COP_1 COP_0 00'
10	(10)	(6)	(2)	microPC \leftarrow adr microinst
11	(11)	(7)	(3)	si Z alors microPC \leftarrow adr microinst sinon microPC \leftarrow 0

structure de la CPU



questions

1. dessiner le micro-contrôleur de cette machine micro-programmée
2. donner le contenu de la mémoire de micro-programme

micro-contrôleur

