Cours d’architecture période 4

Facteurs d’efficacité d’une architecture:

- Pipeline
- Caches
- Composants courants

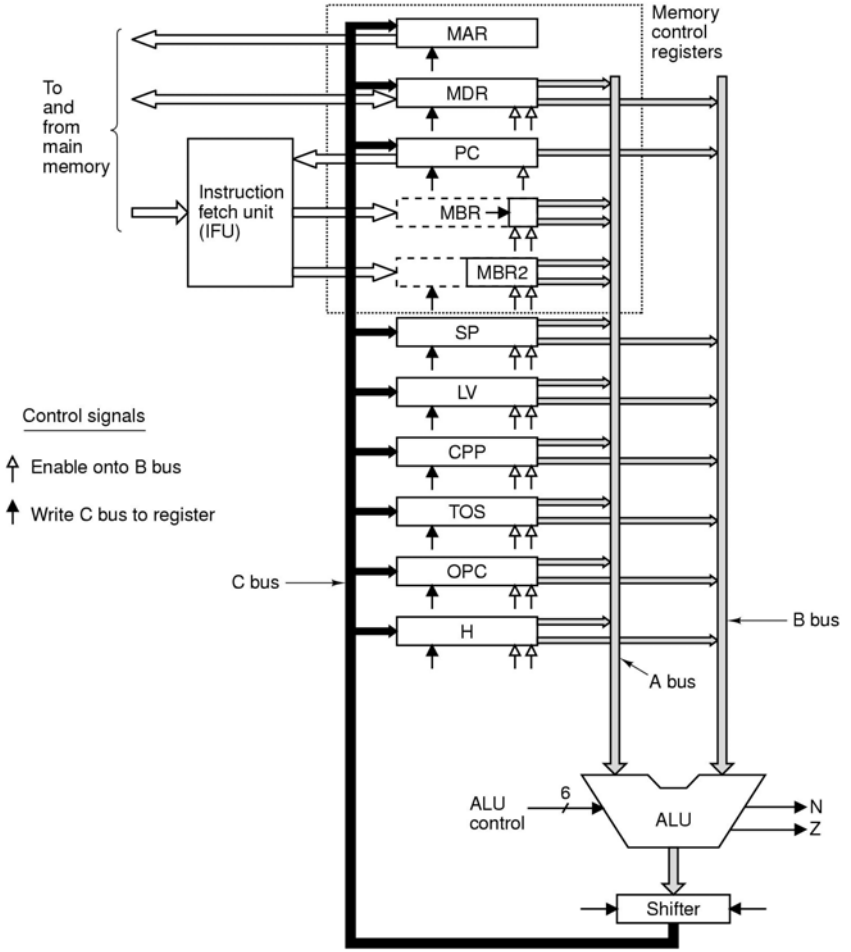
Comprendre le fonctionnement et les contraintes, le rôle du système et des compilateurs.

Micro-architecture: amélioration des performances de la Mic1

Facteurs de manque d'efficacité de Mic1:

- Le nombre d'étapes pour faire une opération arithmétique entre deux registres
- L'UAL utilisée pour les instructions comme pour les opérations sur les données

Architecture à 3 bus et unité de recherche d'instructions



Effet des 3 bus

Instruction Iload de Mic 1:

str	Opérations	description
oad1	H=LV	MBR contient index; copie de LV dans H
oad2	MAR=MBRU+H;rd	MAR=adresser variable locale à ranger
oad3	MAR=SP=SP+1	SP pointe sur nouveau sommet de pile
oad4	PC=PC+1;fetch;wr	Recherche instruction suivante, écriture nouveau sommet de pile
oad5	TOS=MDR; goto(Main1)	Mise à jour de TOS (copie du sommet de pile)
lain1	PC=PC+1;fetch;goto (MBR)	MBR contient le code de l'instruction suivante. On va chercher la suivante encore et on branche.

Instruction Iload de Mic 2:

str	Opérations	description
oad1	MAR=MBRU+LV;rd	Calcul de l'adresse de la variable et lecture
oad2	MAR=SP=SP+1	SPet MAR pointe sur nouveau sommet de pile
oad3	PC=PC+1; fetch;wr	Recherch instr suivante, écriture sommet pile
oad4	TOS=MDR	Mize à jour de TOS
oad5	PC=PC+1;fetch;goto(MB R)	Branchement à l'instrucion suivante et recherche octet suivant.

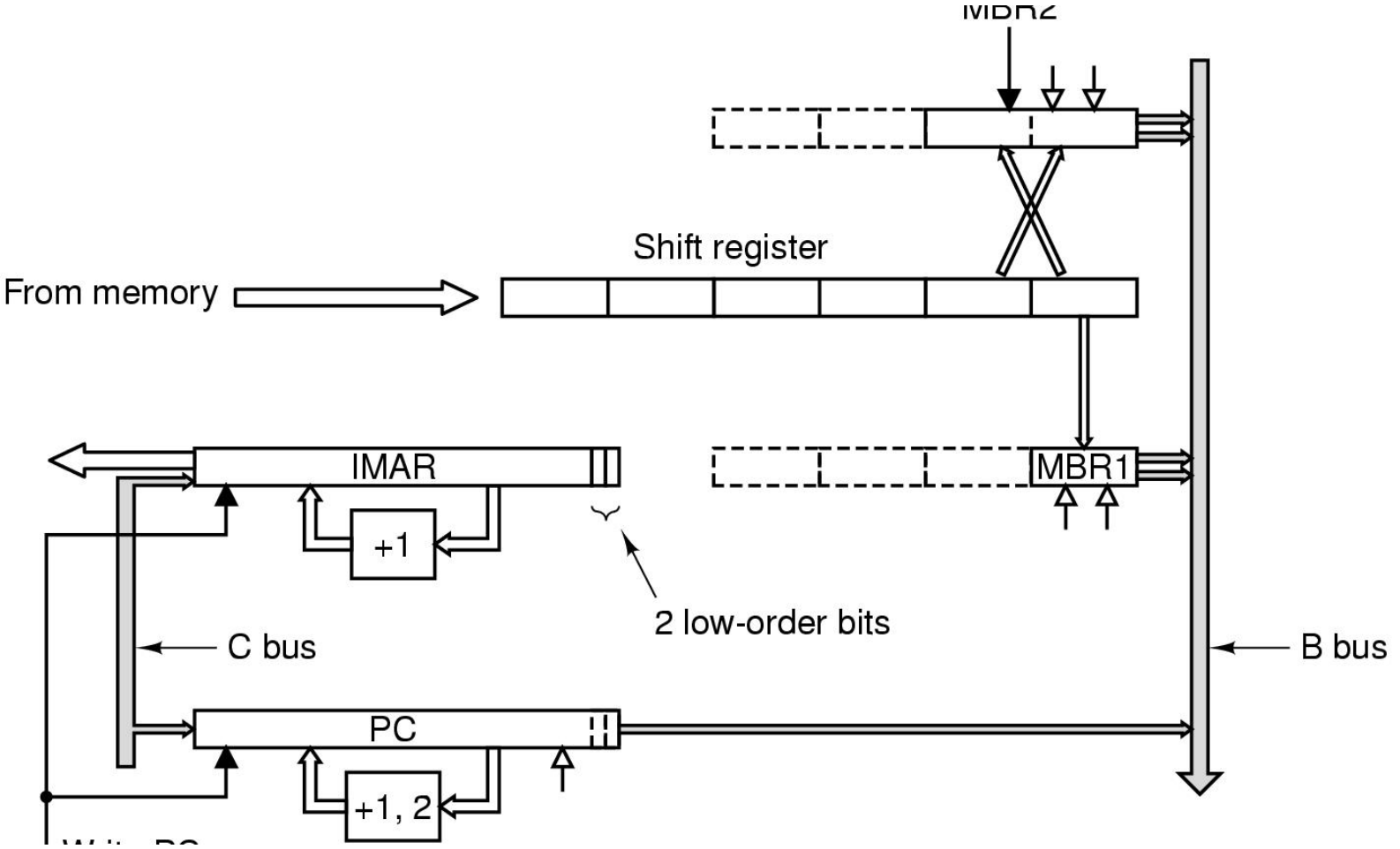
Gestion des instructions sur la MIC1

- Généralement
  - Le PC est transmis à l'UAL et incrémenté
  - Il est utilisé pour chercher les opérandes
  - Le ou les opérandes sont lus en mémoire
  - Le ou les opérandes sont écrits dans la pile éventuellement assemblés dans l'UAL avant
  - L'UAL effectue un traitement et le résultat est rangé dans un registre

Insertion d'une nouvelle unité

- Objectif:
  - Libérer l'UAL des opérations concernant la lecture et le décodage des instructions IJVM,
  - Incrémentation du PC
  - Lecture et assemblage des opérandes
- Contraintes: il faut d'autres circuits
  - Coût plus important

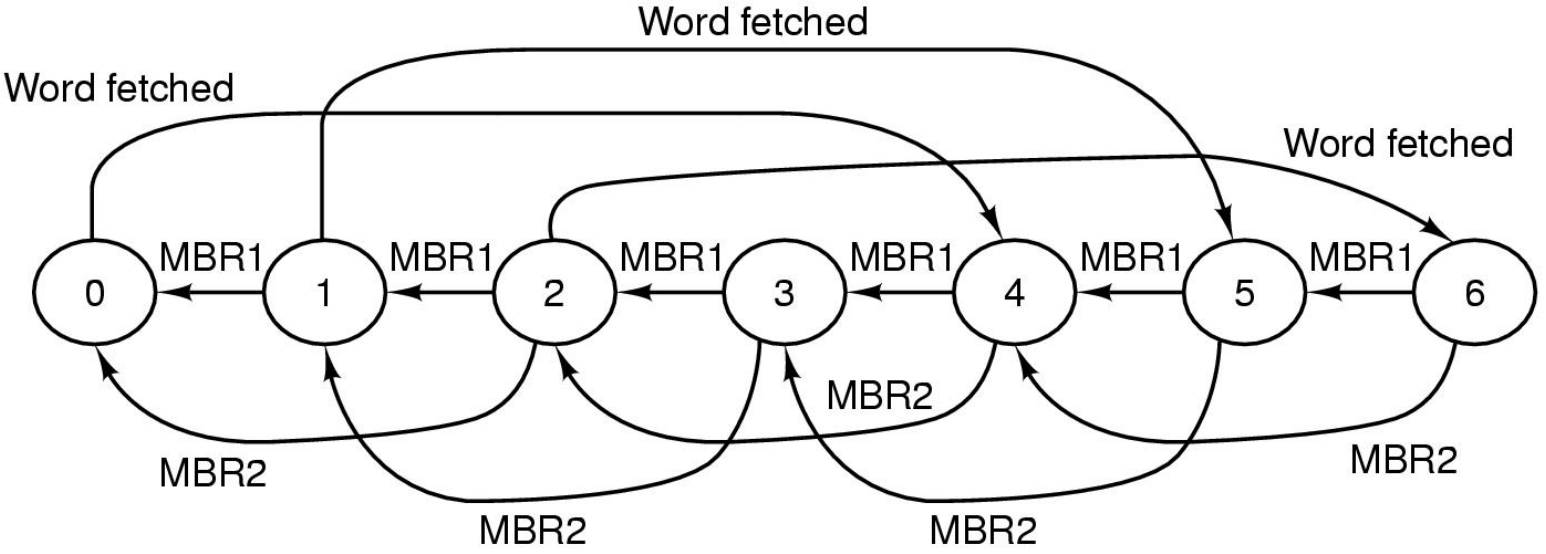
L’unité de recherche des instructions



Une unité au contrôle câblé

- Deux registres MBR:
  - MBR1 8 bits avec 2 accès au bus:
    - MBR1 (extension de signe sur 32 bits)
    - MBR1U (extension simple sur 32 bits)
  - MBR2 16 bits avec les deux types d'accès
  - Un registre à décalage de 6 octets
    - Chaque fois que MBR1 est lu par le chemin de données, il est décalé d'un octet.
    - Chaque fois que MBR2 est lu il est décalé de 2 octets
    - MBR2 et MBR2 contiennent les copies des derniers octets
    - Chaque fois que 4 octets sont libres, on lance la lecture d'un nouveau mot mémoire.
      - Objectif: avoir toujours disponible le code opération et les opérandes (Anticipation).

Automate de contrôle de l'IFU



Transitions

MBR1: Occurs when MBR1 is read
MBR2: Occurs when MBR2 is read
Word fetched: Occurs when a memory word is read and 4 bytes are put into the shift register

Gestion des adresses

- Registre IMAR compte en mots (32 bits).
  - Adresse des 4 prochains octets à charger.
  - incrémenté (de 1) à chaque chargement
- PC compte en octet
  - Adresse de l'octet suivant
  - Incrémenté de 1 ou 2 selon que MBR1 ou MBR2 a été chargé
- En cas de branchement (modification de PC)
  - PC est recopié dans IMAR avec décalage à droite (pour être sur un mot)
  - Chargement du mot et décalage pour ajuster MBR1 sur l'octet instruction.

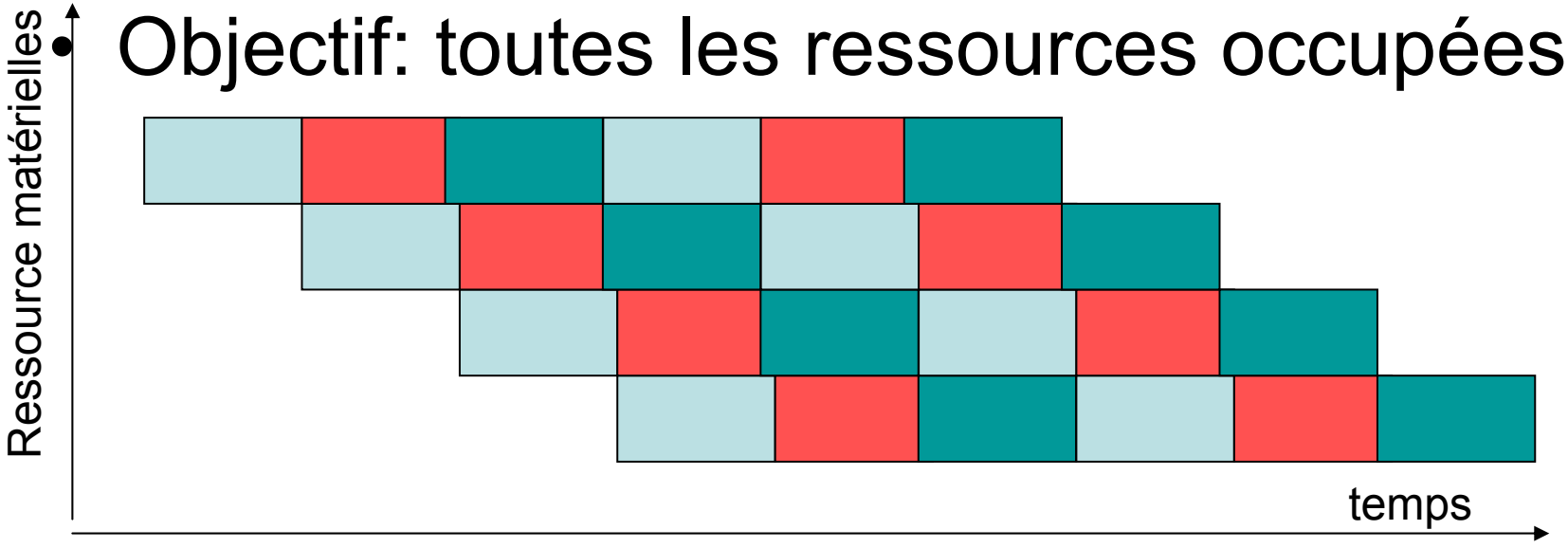
Exemples

instr	Opérations	description
iload1	MAR=LV+MBR1U ;rd	MBR1 contient l'index, pris sans extension de signe
iload2	MAR=SP=SP+1	Modification du sommet de pile
iload3	TOS=MDR ;wr ;goto (MBR1)	Mise à jour de TS et écriture

instr	Opérations	description
goto1	H=PC-1	Copie de PC (qui s'était incrémenté) dans H
goto2	PC=H+MBR2	MBR2 contient l'index, le PC est recopié dans IMAR
goto3		Attente que l'IFU fetch le nouveau mot et décale au besoin pour récupérer le nouveau code op dans MBR1
goto4	Goto(MBR1)	

Principe du pipeline:

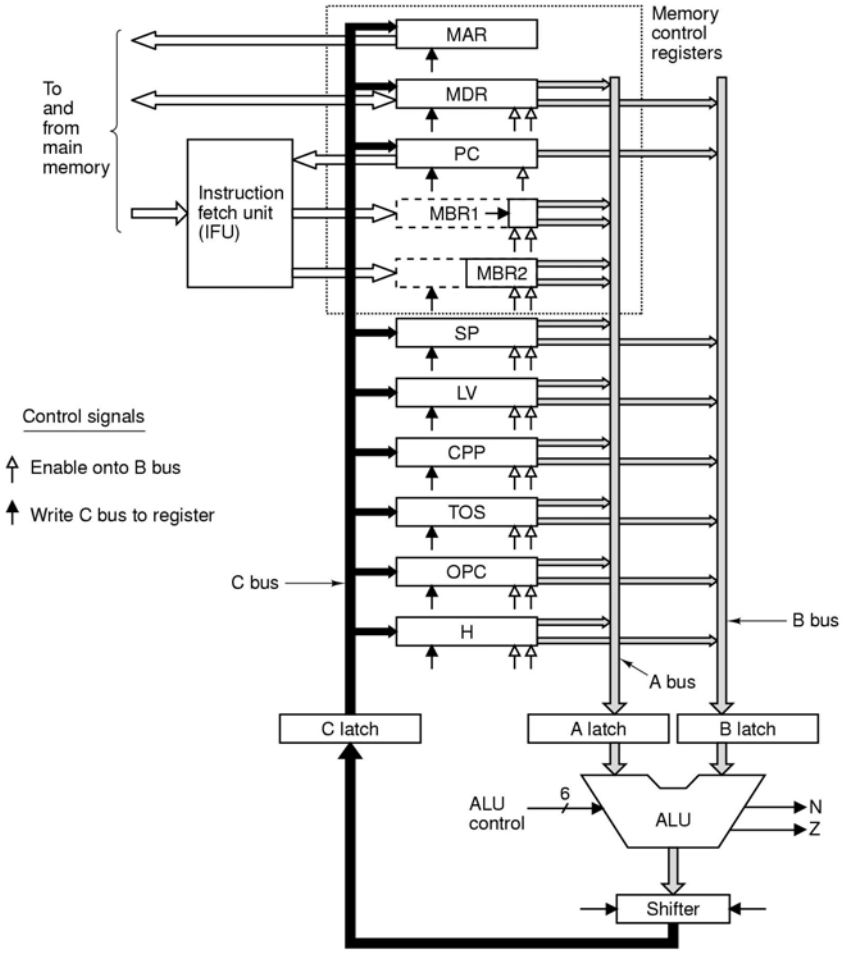
- Travail à la chaîne
- Découpage d'une tâche en sous-tâches successives
- Attribuer des ressources autonomes à chaque sous-tâche
- Objectif: toutes les ressources occupées



De Mic 2 à Mic 3

- Pipelinage du chemin de données.
  - Registres vers entrée de l'UAL
  - Traversée de l'UAL
  - Ecriture du résultat en registre
- Conséquence:
  - Registres en entrée A et B, registre en sortie R de l'UAL (ressources autonomes)
  - On peut augmenter la fréquence de l'horloge

Mic 3


Memory control registers
MAR
MDR
PC
MBR1
MBR2
SP
LV
CPP
TOS
OPC
H
To and from main memory
Instruction fetch unit (IFU)
Control signals
Enable onto B bus
Write C bus to register
C bus
B bus
A bus
C latch
A latch
B latch
ALU control
6
ALU
N
Z
Shifter

Swap de Mic 2 à Mic 3

instr	Opérations	description
Swap1	MAR=SP-1 ;rd	Lit le 2mot de la pile
Swap2	MAR=SP	Se prépar à écrire au sommet
Swap3	H=MDR ;wr	Le deuxième mot est dans MDR, lancement de l'écriture au sommet de pile
Swap4	MDR=TOS	Le sommet de pile dans MDR
Swap5	MAR=SP-1 ;wr	Ecriture dans deuxième mot
swap6	TOS=H ; goto (MBR1)	Maj TOS

	Swap1	Swap2	Swpa3	Swap4	Swap5	swap6
cycle	MAR=SP-1 ;rd	MAR=SP	H=MDR ;wr	MDR=TOS	MAR=SP-1 ;wr	TOS=H ; goto (MBR1)
1	B=SP					
2	C=B-1	B=SP				
3	MAR=C;rd	C=B				
4	MDR=mem	MAR=C				
5			B=MDR			
6			C=B	B=TOS		
7			H=C;wr	C=B	B=SP	
8			Mem=MDR	MDR=C	C=B-1	B=H
9					MAR=C;wr	C=B
10					Mem=MDR	TOS=C
11						Goto (MBR1)

Avantages

- Temps de cycle « divisé par 3 »
- Meilleur entrelacement des calculs
- 11*tc contre 6*3*tc

Pipeline et architectures RISC

- Le mécanisme du pipeline est un fondamental des architectures
  - À chargement rangement
  - À format d'instruction fixe
  - A contrôle câblé
- Exemple (DE99), jeu d'instructions

ADD, SUB	Reg-reg ou reg-Imm	Rd<-Ra op Rb, RCC modifié Rd<-Ra op ES(Imm) ; RCC modifié
AND, OR	Reg-reg Ou Reg-Imm	Rd<-Ra op Rb Rd<-Ra op ES(Imm)
CMP	Reg-Reg	Ra-Rb, RCC modifié
LD	Reg-regOu Reg-Imm	Rd<-Mem[Ra+Rb]R<-Mem[Ra+ES(Imm)]
ST	Reg-regOu Reg-Imm	Mem[Ra+Rb]<-RdMem[Ra+ES(Imm)] <-Rd
Bcond	Branchement	Si cond vraie PC-PC+ES(Imm)