

Master 1 - SEM

Processeurs embarqués

TD 5

Pipeline

Année 2020-2021

Pr. R. BOUDOUR



Exercice 1

Un processeur non pipeline possède un temps de cycle de 10 ns et des latches de pipeline d'une latence de 0.5 ns.

- 1) Quels sont les temps de cycle des versions pipelinées du processeur avec un pipeline de 2, 4, 8 et 16 étages si la logique de chemin de données est répartie de manière égale entre les étages de pipeline ?
- 2) Quelle est la latence de chacune des versions pipelinées du processeur ?
- 3) Combien d'étages sont requis pour atteindre un temps de cycle de 2 ns ?
- 4) Quel est le temps de cycle minimal qui peut être atteint avec un pipeline de 4 étages si de la logique supplémentaire est assignée au dernier étage afin d'équilibrer sa latence avec celles des latches contenus dans les autres étages ?

Réponse 1

- 1) $\text{Temps cycle}_{\text{pipeliné}} = \text{temps cycle}_{\text{non pipeliné}} / \text{Nombre d'étages pipeline} + \text{surcoût}$
En appliquant cette formule, on obtient : 5.5ns, 3ns, 1.75ns et 1.125ns respectivement.
- 2) Latence de chaque processeur = temps de cycle * nombre d'étages
ce qui donne 11ns, 12ns, 14ns et 18ns respectivement.
- 3) $\text{Nombre d'étages pipeline} = \text{temps cycle}_{\text{nonpipeliné}} / \text{Temps cycle}_{\text{pipeliné}} - \text{surcoût}$
L'application de la formule nous donne 6.67 étages $\approx$ 7 étages
- 4) Latence du chemin de données = 10ns + 3 * 0.5 = 11.5ns
temps de cycle = 11.5 / 4 = 2.875ns

Exercice 2

Soit un processeur non pipeliné possédant un temps de cycle de 25 ns et que le chemin de données est constitué de modules dont les latences sont respectivement de 2, 3, 4, 7, 3, 2 et 4ns (dans cet ordre). Pour implémenter le pipelining dans ce processeur, il n'est pas possible de réarranger l'ordre des modules (par exemple placer l'étage de lecture des registres avant l'étage de décodage d'instruction) ni de diviser un module en plusieurs étages de pipeline. La latence des latches du pipeline est de 1 ns.

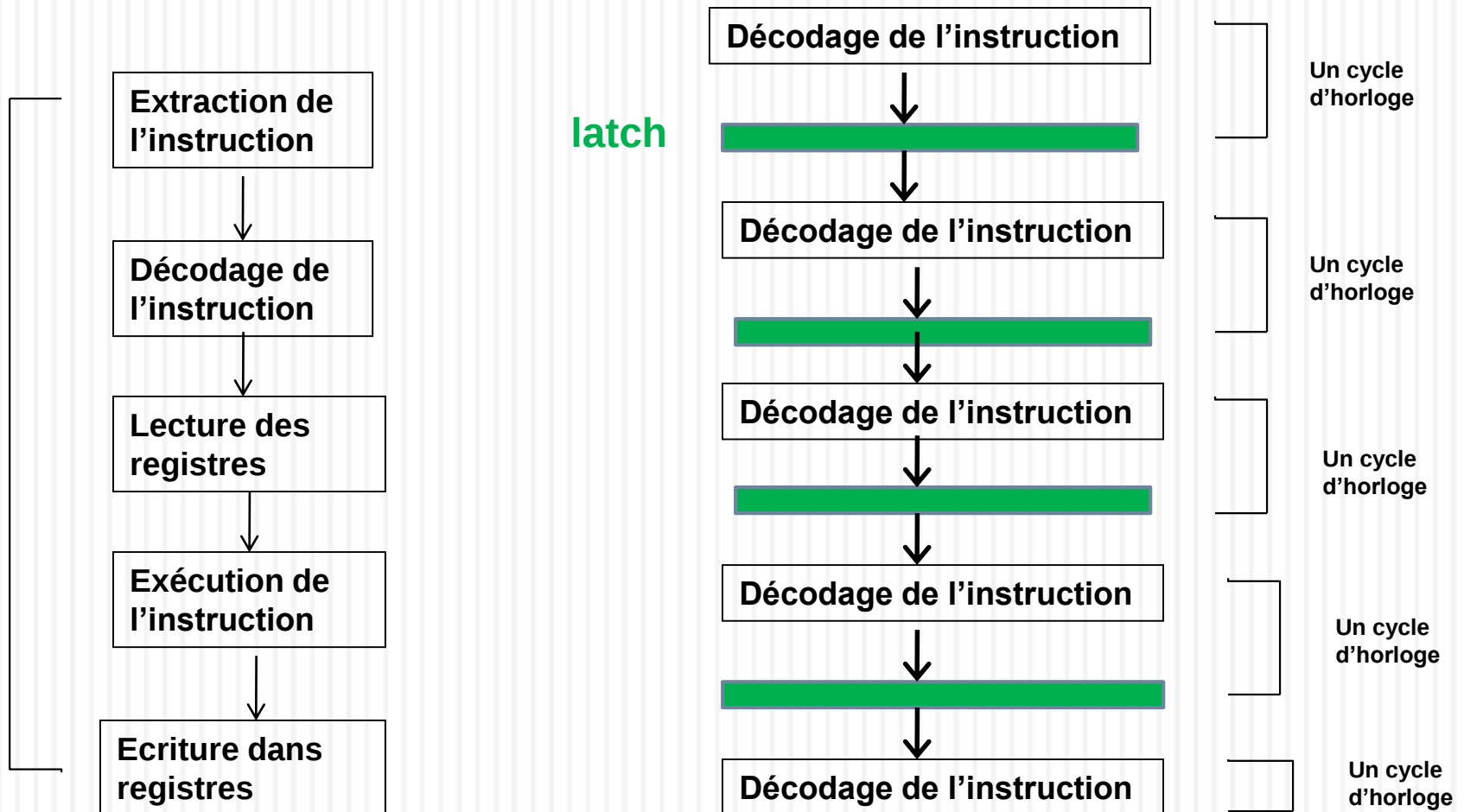
- a) Quel est le temps de cycle minimal qui peut être atteint en implémentant le pipelining sur ce processeur ?
- b) Si le processeur est divisé en un plus petit nombre d'étages qui lui permettent d'atteindre la latence minimale de la question précédente, quelle est la latence totale du pipeline ?
- c) Si le pipeline est limité à 2 étages, quel est le temps de cycle minimal ?
- d) Quelle est la latence du pipeline de la question précédente ?

Réponse 2

- a) Temps de cycle minimal = Latence module le plus long + surcoût
 $= 7 + 1 = 8 \text{ ns}$
- b) Nous regroupons dans un unique étage n'importe quel ensemble de modules adjacents en respectant la contrainte de cycle, ce qui nous donne un pipeline à 5 étages.
latence totale $= 8 * 5 = 40 \text{ ns}$
- c) Temps de cycle minimal $= 16 \text{ ns}$
- d) Latence du pipeline $= 16 * 2 = 32 \text{ ns}$

Processeurs non pipeliné et pipeliné

6



Exercice 3

7

En considérant le pipeline précédent (sans forwarding), dessiner le diagramme d'exécution pour le fragment de code suivant :

```
add r1,r2,r3
```

```
sub  r4,r5,r6
```

```
mul  r8,r9,r10
```

```
add r12,r13,r14
```

Réponse 3

8

Il n'existe aucun aléa d'instruction dans ce fragment de code, ainsi les instructions vont traverser le pipeline à la cadence d'un étage par cycle

	1	2	3	4	5	6	7	8	cycles
EI	add	sub	mul	div					
DI		add	sub	mul	div				
LR			add	sub	mul	div			
EX				add	sub	mul	div		
ER					add	sub	mul	div	

Temps d'exécution_{fragment} = 8 cycles

18/02/2021 16:30:11

Exercice 4

9

En considérant le pipeline précédent (sans forwarding), dessiner le diagramme d'exécution pour le fragment de code suivant :

```
add r1,r2,r3
```

```
sub r4,r5,r6
```

```
mul r8,r9,r4
```

```
add r12,r13,r14
```

Réponse 4

10

- Cette fois, le programme fait intervenir un aléa LAE entre l'instruction sub qui écrit dans r4 et l'instruction mul qui le lit. En conséquence, mul ne sera pas capable de lire ses registres d'entrée avant que sub n'ait terminé l'écriture dans l'étage ER, ce qui créera un blocage du pipeline.

	1	2	3	4	5	6	7	8	9	10	cycles
EI	add	sub	mul	div							
DI		add	sub	mul	div	div	div				
LR			add	sub	mul	mul	mul	div			
EX				add	sub	bulle	bulle	mul	div		
ER					add	sub	bulle	bulle	mul	div	

Temps d'exécution_{fragment} = 10 cycles

18/02/2021 16:30:11

Exercice 5

11

- Supposons que 30% des instructions soient des chargements, et qu'une fois sur deux, une instruction qui suit une instruction de chargement dépende du résultat du chargement. Si cet aléa crée un délai d'un seul cycle, de combien la machine pipelinée idéale (avec un CPI de 1), qui n'ajoute pas de délai au pipeline, est-elle plus rapide que celle avec un pipeline réel ? Ignorer toutes les suspensions autres que celles du pipeline.

Réponse 5

12

- La machine idéale est plus rapide du rapport des CPI. Le CPI pour les instructions qui suivent un chargement est de 1.5, car elles sont suspendues une fois sur deux. Parce que les chargements représentent 30% des instructions, le CPI effectif est $(0,7*1 + 0,3*1,5)$. La machine idéale est donc 1,15 fois plus rapide.

Exercice 6

13

- générer le code qui évite les suspensions de pipeline pour la séquence suivante :

$a = b + c;$

$d = e - f;$

- On suppose que les chargements ont une latence de 1 cycle.

Réponse 6

14

lw rb,b

lw rc,c

lw re,e

add ra,rb,rc

lw rf,f

sw a,ra

sub rd,re,rf

sw d,rd

instructions échangées pour éviter une suspension

sw et lw intervertis pour éviter la suspension

Les deux blocages des chargements lw rc,c vers add ra,rb,rc et lw rf,f vers sub rd,re,rf) ont été éliminés. Il existe une dépendance entre l'instruction UAL et le rangement, mais la structure du pipeline permet l'envoi du résultat. On remarquera que l'utilisation de registres différents dans les deux premières instructions était importante pour que cet ordonnancement soit correct. En particulier si e était chargée dans le même registre que b ou c, cet ordonnancement ne serait pas correct. En général, l'ordonnancement du pipeline peut augmenter le nombre de registres nécessaires.

Exercice 7

15

L'horloge du processeur est 400MHz

Le pipeline est à 6 niveaux.

Quel est le temps nécessaire pour traiter 1000 instructions, a) si le traitement n'est pas pipeliné, b) si le traitement est pipeliné ?

Réponse 7

16

$$P = 1/400 \times 10^{-6} = 2.5 \text{ ns}$$

$$T = 1000 \times 6 \times 2.5 \text{ ns} = 15 \text{ us}$$

$$T = (6+999) \times 2.5 \text{ ns} = 1005 \times 2.5 \text{ ns} = 2512.5 \text{ ns}$$

Exercice 8

17

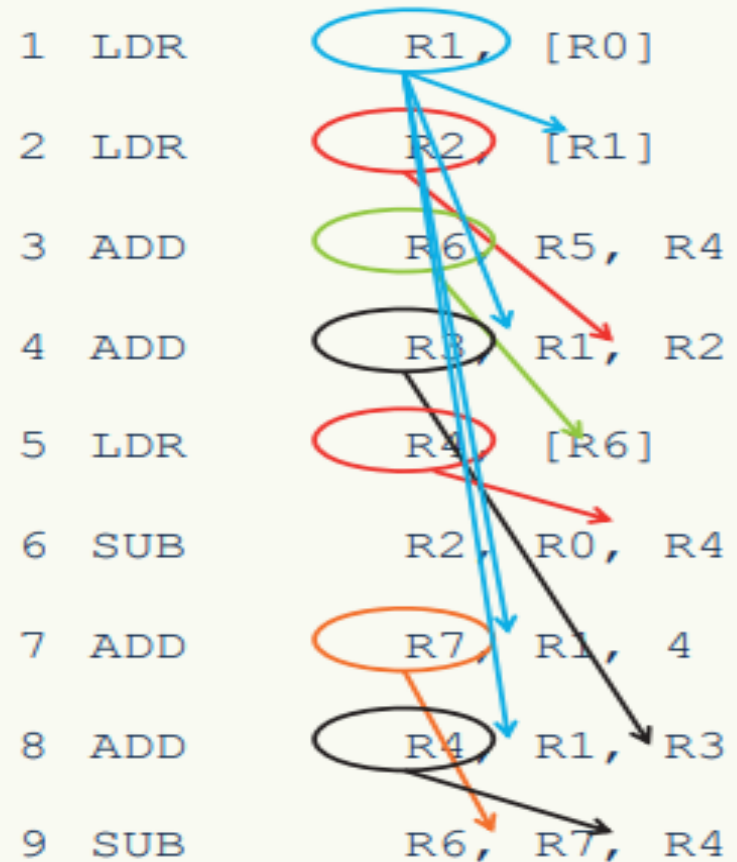
Indiquez les dépendances de types RAW, WAR et WAW

```
1  LDR    R1, [R0]
2  LDR    R2, [R1]
3  ADD    R6, R5, R4
4  ADD    R3, R1, R2
5  LDR    R4, [R6]
6  SUB    R2, R0, R4
7  ADD    R7, R1, #4
8  ADD    R4, R1, R3
9  SUB    R6, R7, R4
```

Réponse 8

18

RAW (read after write):



Réponse 8

19

WAR(write after read):

The diagram illustrates a Write After Read (WAR) hazard in the following assembly code. Arrows indicate the flow of data from the source registers of one instruction to the destination register of a subsequent instruction.

1	LDR	R1, [R0]
2	LDR	R2, [R1]
3	ADD	R6, R5, R4
4	ADD	R3, R1, R2
5	LDR	R4, [R6]
6	SUB	R2, R0, R4
7	ADD	R7, R1, 4
8	ADD	R4, R1, R3
9	SUB	R6, R7, R4

Annotations in the diagram:

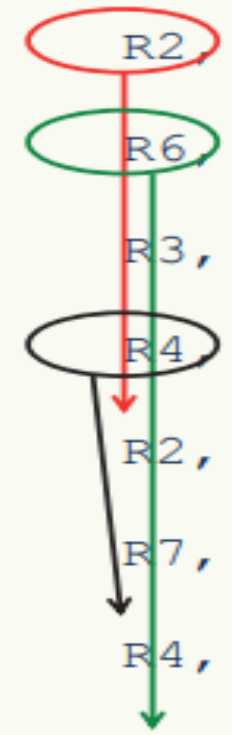
- A red oval around **R4** in instruction 3 and a red arrow pointing to **R4** in instruction 5 represent the **Read** phase of the hazard.
- A green oval around **R2** in instruction 4 and a green arrow pointing to **R4** in instruction 5 represent the **Write** phase of the hazard.
- A black oval around **R4** in instruction 5 represents the **Write** phase of the hazard.
- A blue oval around **R4** in instruction 6 and a blue arrow pointing to **R4** in instruction 8 represent the **Read** phase of the hazard.

Réponse 8

20

WAW(write after write):

1	LDR	R1, [R0]
2	LDR	R2, [R1]
3	ADD	R6, R5, R4
4	ADD	R3, R1, R2
5	LDR	R4, [R6]
6	SUB	R2, R0, R4
7	ADD	R7, R1, 4
8	ADD	R4, R1, R3
9	SUB	R6, R7, R4



Exercice 9

21

- Combien peut coûter l'aléa structurel de chargement. Supposons que les références données constituent 40% des instructions, et que le CPI idéal de la machine pipelinée est de 1 en ignorant l'aléa structurel. On suppose que la machine avec aléa structurel a une fréquence d'horloge qui est 1,05 fois plus élevée que celle de la machine sans aléa. Sans considérer d'autres dégradations de performance, quel pipeline, avec ou sans aléa structurel, est le plus rapide, et de combien ?

Réponse 9

22

- Il y a plusieurs manières de résoudre ce problème. Le plus simple est peut être de calculer le temps moyen d'une instruction sur les deux machines : D'abord, calculons l'accélération du pipeline pour chaque machine en utilisant la formule :

$$\text{Tempsmoyend'uneinstruction} = \text{CPI}_{\text{idéal}} * \text{Tempsdeccyle}$$

Puisqu'il n'y a pas de suspensions, le temps moyen d'une instruction pour la machine idéale est simplement le temps de cycle idéal.

Le temps moyen d'une instruction pour la machine avec aléas structurel est :

$$\text{Tempsmoyend'uneinstruction} = \text{CPI}_{\text{idéal}} * \text{Tempsdeccyle} = (1 + 4 * 1) * \frac{\text{tempsdecycleidéal}}{1,05}$$

- La machine sans aléa structurel est de manière évidente plus rapide. On peut utiliser le rapport des temps moyens d'exécution d'une instruction pour dire qu'elle est de 1,33 fois plus rapide.
- Pour éviter cet aléa structurel, le concepteur peut fournir un accès mémoire séparé pour les instructions, soit en divisant le cache en caches séparés pour les instructions et données

Vrai/Faux

23

- ❑ Le principe du pipeline consiste à diviser le processus d'exécution des instructions en un nombre donné d'étapes fonctionnellement indépendantes.
- ❑ L'indépendance des différents étages du pipeline permet au processeur de traiter plusieurs instructions simultanément, chacune d'entre elles se trouvant à un stade différent du cycle d'exécution.
C'est le parallélisme par anticipation.
- ❑ Un Pentium 4 est un processeur qui utilise le pipelining.

Vrai/Faux

24

- Une "dépendance" survient quand une instruction ne peut s'exécuter avant l'achèvement complet de l'instruction qui la précède.
- La dépendance de ressource est le fait que deux instructions, même à des stades d'exécution différents, sollicitent simultanément une même unité fonctionnelle (UAL, registres, mémoire cache, décodeur-séquenceur...) .
- La dépendance d'instruction est le fait que deux étages du pipeline aient besoin simultanément de la même instruction.

- CISC signifie _____?
 - **A.** Common Instruction Set Computers
 - **B.** Complex Instruction Set Compilers
 - **C.** Complex Instruction Set Computers
 - **D.** Compound Instruction Set Computer

Trouver des inconvénients pour CISC

26

- ❑ Nette séparation entre les instructions d'accès mémoire et les autres Instructions standardisées, en taille et en durée d'exécution
- ❑ beaucoup trop de codes d'instruction différents.
- ❑ taille des instructions élevée et variable (1 à 15 bytes par instruction)
- ❑ structure des instructions non standardisées: exécution complexe, peu performante
- ❑ Unité de décodage câblée, non microcodée
- ❑ architecture superpipeline, superscalaire
- ❑ Très nombreux registres à usages général
- ❑ et un jeu d'instruction réduit aux instructions simples

- Une architecture RISC typique ...
 - ▣ a des instructions qui s'effectuent en plus d'un cycle d'horloge
 - ▣ a un format d'instructions unique
 - ▣ utilise un séquenceur microprogrammé et de nombreux registres généraux
 - ▣ les 3 dernières réponses
 - ▣ aucune des 4 dernières réponses

- L'accès à la mémoire dans l'architecture RISC est limité aux instructions?
 - A. CALL et RET
 - B. PUSH et POP
 - C. ST et LD
 - D. MOV et JMP

- Quels sont les moyens permettant d'augmenter les performances d'un microprocesseur
 - Augmenter la fréquence d'horloge
 - diminuer le CPI
 - Augmenter le CPI
- Combien y'a-t-il de dépendances de données.
 - 2
 - 4
 - 6
 - 8

- Quelle dépendance est la plus critique ?
 - RAR
 - RAW
 - WAR
 - WAW
- Quelle technique est utilisée pour limiter les impacts des alias RAW ?
 - Castoming
 - Sending
 - fowading
- Quelles sont les solutions pour limiter la dépendance de contrôle entre instructions ?
 - Délai de branchement
 - Registre logique
 - Prédiction de branchement
 - Limiter le nombre de branchement

.Pipelining augmente _____ des instructions du processeur.

- A. l'efficacité**
- B. la latence**
- C. le débit**
- D. les deux A et C sont vrais.**

- Dans un ordinateur, la sortie du compilateur est appelée _____?
- A. programme
 - B. code source
 - C. linked code
 - D. code objet

- La manière dont les composants informatiques sont connectés les uns aux autres est appelée _____?
- A. mise en page informatique
 - B. l'architecture des ordinateurs
 - C. pièces d'ordinateur
 - D. matériel informatique