

Informatique Générale

TD n° 11

Exercice 1 : Performance

Un processeur à 1,8 GHz exécute un programme de benchmark qui utilise un mélange de 4 types d'instruction:

Type instruction	Nombre d'instruction exécutées	Nombre de cycle / instruction
Opération entière	150000	1
Transfert mémoire	45000	2
Opération flottante	55000	2
Contrôle (sauts)	20000	2
Affichage	500	15

- Combien de cycles se programme prendra-t-il pour s'exécuter ?
 $150000 * 1 + 2 * (45000 + 55000 + 20000) + 15 * 500 = 397500$
- Quelle sera la durée d'exécution ? $397500 / (1,8 * 10^{-9})$, environ 220 μ s
- Calculez le CPI (cycle per instruction) de ce programme, défini comme le rapport entre le nombre de cycles requis pour son exécution et le nombre d'instructions.
 Il y a 270500 instructions exécutées en tout, donc $CPI = 397500 / 270500 = 1,47$

Exercice 2 : Performance encore

On considère le même processeur que précédemment, et le programme suivant,

```

p <- 0                                1+2 cycles
tant que b > 0 faire                  1+2+1 cycles (+2 si saut)
    p <- p + a                        1+3*2+1
    b <- b - 1                        1+2*2+1
fin tant que                          1+2 (saut en début de boucle)
afficher p                            1+2+15

```

- Que calcule ce programme ? Essayer de la faire fonctionner pour le deviner, en prenant par exemple $a = 3$ et $b = 5$. Évidemment $a * b$
- Chaque opération ou comparaison demande que ses opérandes (sauf les constantes) soient en registre. Pour chaque ligne, indiquez le nombre de cycles nécessaires (comptez 1 cycle pour aller chercher l'instruction en mémoire)
- De quelle variable dépend son temps d'exécution ? bien sûr b
- Combien de cycles au total se programme prendra-t-il pour s'exécuter en fonction de cette variable ? $3 + b * (4 + 8 + 6 + 3) + 2 + 18 = 23 + 21b$
 Formellement, il faut compter que le test du tantque se fait 1 fois de plus que la boucle et saute, mais ça ne change pas fondamentalement les résultats : $23 + 6 \rightarrow 29$
- Quelle sera la durée d'exécution pour $b = 1023$? $(29 + 1023 * 21) / 1,8 =$ environ 1,2 μ s

Exercice 3 : Performance encore et encore

Mêmes questions que précédemment, mais avec le programme suivant

p <- 0	1 + 2 cycles
tant que b > 0 faire	1 + 2 + 1 cycles (+ 2 si saut)
si b % 2 = 1 alors	1 + 2 + 1+1 cycles (+ 2 si saut)
p <- p + a	1 + 3*2 + 1 (que si b impair)
fin si	
b <- b / 2	1 + 2*2 + 1
a <- a * 2	1 + 2*2 + 1
fin tant que	1 + 2
afficher p	1 + 2 + 15

Comparer les performances des 2 programmes pour un b de la forme $2^n - 1$ et 2^n .

Le programme calcule aussi $a * b$, mais à « l'égyptienne ».

Nombre total de cycles, si U est le nombre de 1 dans la représentation binaire de b et Z le nombre de 0, et $b \leq 2^n$ ($n=U+Z$)

$$3 + n * (4 + 5 + 6 + 6 + 3) + 8 * U + 6 + 18 = 27 + 24n + 8U$$

Si on multiplie par 1023, $n=U=10$ donc $27+240+80=347$ cycles, à comparer aux plus de 20000 de l'autre programme. Si on multiplie par 2047, le premier aura environ le double de cycles (on double le nombre de boucles), l'autre uniquement les cycles pour 1 boucle de plus, soit près de 40000 contre moins de 600. L'un est linéaire, l'autre logarithmique.

En multipliant par 2^n , on a seulement $24(n+1) + 27 + 8$ cycles, soit 299 pour $n = 10$.

Exercice 4 : Pipeline

Dans cet exercice nous allons considérer processeur ayant un pipeline à 5 étages, chacun prenant un cycle.

1. Une instruction entre dans le pipeline au cycle i , à quel cycle aura-t-elle fini son exécution ?
elle s'exécute en $i+0, i+1, i+2, i+3, i+4$ et sera donc complètement terminée en $i+5$
2. On considère un programme de n instructions. En ignorant les *stall* dans le pipeline, calculez le nombre de cycles nécessaires pour son exécution.
 $n + (p - 1)$ si p est le profondeur du pipeline, donc ici $n + 4$
3. Quel est le CPI de ce programme sur ce processeur ?
 $n+4 / n$, tend donc vers 1 quand n tend vers $+\infty$
4. Quel serait le CPI sur un processeur n'ayant pas de pipeline ?
 $5n/n = 5$

Exercice 5 : Data Hazard

Un processeur possède un pipeline à 4 étages tel que décrit dans le cours (Fetch-Decode-Execute-Write). La lecture des données est faite lors de l'étape Execute. Un programme est composé des instructions suivantes

```
ADD R4, R1, R2
SUB R1, R2, R4
MUL R6, R1, R5
DIV R3, R2, R5
MUL R2, R3, R4
```

Ce programme n'utilise que des registres et ne fait donc pas d'accès mémoire.

1. En ignorant les *data hazard*, représentez l'exécution de ces instructions dans le pipeline en vous basant sur le cours.

ADD R4, R1, R2	F	D	E	W (R4)						
SUB R1, R2, R4		F	D	E (R4)	W (R1)					
MUL R6, R1, R5			F	D	E (R1)	W(R6)				
DIV R3, R2, R5				F	D	E	W(R3)			
MUL R2, R3, R4					F	D	E(R3)	W(R2)		

2. Refaites le même schéma en rajoutant des *stall* lorsque c'est nécessaire

ADD R4, R1, R2	F	D	E	W (R4)							
SUB R1, R2, R4		F	D	stall	E (R4)	W (R1)					
MUL R6, R1, R5			F	stall	D	stall	E (R1)	W(R6)			
DIV R3, R2, R5				stall	F	D	E	stall	W(R3)		
MUL R2, R3, R4					stall	F	D	E	stall	E(R3)	W(R2)

le stall en rouge : car on ne peut pas avoir 2 écritures registre en même temps

3. Proposez un ré-ordonnancement des instructions minimisant le temps total d'exécution.

Attention à ce que les registres gardent les mêmes valeurs quand on déplace les instructions : donc

ADD R4, R1, R2
 DIV R3, R2, R5
 SUB R1, R2, R4
 MUL R2, R3, R4
 MUL R6, R1, R5

ADD R4, R1, R2	F	D	E	W(R4)							
DIV R3, R2, R5		F	D	E	W (R3)						
SUB R1, R2, R4			F	D	E (R4)	W (R1)					
MUL R2, R3, R4				F	D	E(R3,R4)	W(R2)				
MUL R6, R1, R5					F	D	E (R1)	W(R6)			

Exercice 6 : Control Hazard

1. Quel est le coût d'une erreur de prédiction de branchement ?
 Il faut vider le pipeline, donc c'est un coût proportionnel à la profondeur.
2. Un programme est composé de 34000 instructions dont 10% sont des sauts. On dispose d'un processeur qui a un taux de succès de 90% aux prédictions de branchement. Calculez le nombre de cycles nécessaires à l'exécution de ce programme sur un processeur dont le pipeline a 5 étages.
 Le programme ayant 34000 instructions, il prendra environ 34000 cycles pour s'exécuter. 3400 instructions sont des branches. Dans 10% des cas (340 instructions), il faudra flusher le pipeline, ce qui aura un coût de 5. Donc au final ça coûtera $33660 + 340 \times 5$

3. Même question concernant un pipeline à 20 étages
Dans le cas où le pipeline est de 20 étages, c'est $33660 + 340 \cdot 20$
4. Y a-t-il un taux de succès minimum qui rendrait le processeur à 5 étages plus efficace que celui à 20 ? Commentez.
Soit x le taux de succès en %
 $Nb \text{ cycles} = 31600 + 3400 \cdot x + 3400(1-x) \cdot p$ avec p profondeur du pipeline
Il faut tracer les 2 courbes pour $p=5$ et $p=20$
Pour $x=0.5\%$
 $p=5$: 41800 cycles
 $p=20$: $31600 + 1700 + 34000 = 67300$
Si on fait le calcul pour plusieurs valeurs on s'aperçoit que c'est toujours celui avec le pipeline le plus court qui gagne. Donc pourquoi faire des pipeline long? c'est parce que cet exercice fait une hypothèse fautive. On augmente la longueur du pipeline, sans changer la fréquence. Autrement dit, une instruction prend plus de temps pour s'exécuter sur la machine à long pipeline, ce qui est le contraire de ce qu'on voudrait en allongeant le pipeline. Il faudrait donc, pour être plus réaliste, augmenter la fréquence.