

UNIVERSITAT DE LLEIDA
Escola Politècnica Superior
Grau en Enginyeria Informàtica
Arquitectura de Computadors

Pràctica: Jerarquia de Memòria

Sergi Sales Jové, Sergi Simón Balcells
GM3

Professorat : Concepció Roig
Data : Dimecres 24 de Abril

Índex

1	Introducció	1
2	Anàlisi de resultats	1
2.1	Opcions de millora	1
3	Conclusió	2
4	Annex	3

Llista de taules

1	Resultats de l'experiment	3
---	-------------------------------------	---

1 Introducció

Per veure el funcionament del processament segmentat es va fer un programa que realitzes una petita fórmula matemàtica composta pel sumatori d'un seguit de divisions d'exponents. El programa es va realitzar fora de classe de dues formes, una forma bàsica i una segona forma apta per aplicar la millora Delay Slot en executar-lo.

Amb els programes creats i verificats, en el laboratori es va realitzar l'execució d'aquests amb el software winmips64, que ens deixava executar-lo amb el processament segmentat amb l'opció d'afegir un parell d'opcions de millora:

- Delay slot: Permet que algunes instruccions s'executin sense els efectes de les instruccions anteriors. Això passa, per exemple, en una instrucció senzilla col·locada després d'una instrucció de salt.
- Forwarding: És una opció que optimitza l'execució del programa reduint els riscos que pot provocar el pipelining.

Els resultats de les execucions es troben en la Taula 1 de l'Annex.

2 Anàlisi de resultats

2.1 Opcions de millora

L'execució del programa es podia realitzar variant les opcions de millora que ens proposava el winmips64. Aquestes eren Delay slot i Forwarding. Aleshores com es demanava a la pràctica, vàrem executar el programa sense opcions de millora, amb només una de les dues (en els dos casos) i amb les dues alhora, com es pot veure a la Taula 1.

Si ens fixem en la primera columna, on tenim l'execució sense opcions de millora, podem veure que el CPI és més gran que 5. En principi les instruccions tenen 5 cicles per cada instrucció, i això vol dir que en executar-ho de manera segmentada acabem tenint un CPI més gran que l'execució seqüencial? Doncs no. Resulta que durant l'execució, algunes de les operacions acaben executant-se amb molts més cicles, com la divisió.

Si comparem les altres columnes de la taula, podem veure que les columnes amb menys cicles són les quals tenen activat el Forwarding. Aquestes dues columnes tenen el mateix CPI, i en l'únic que es diferencien és en els RAW stalls, i això deu ser conseqüència del Delay slot activat en la 4a columna.

Aquest fenomen pot ser perquè el programa estava ben optimitzat i no presentava quasi diferència amb la versió desenvolupada per al delay.

Una altra cosa a observar és el nombre de cicles de cada execució, que ens mostren quina d'aquestes finalitza abans que les altres. En aquest cas, amb l'opció de Forwarding els programes acaben més de pressa (acaben igual en els dos casos de Forwarding), tot i que el cas del delay slot no s'allunya tant com el cas que no té millores.

Un dels últims criteris que tindrem en compte és l'acceleració d'aquestes execucions, que es calcula amb la següent fórmula:

$$acceleracio = \frac{CPI\ ideal * profunditat\ de\ segmentacio}{CPI\ ideal + cicles\ de\ detencio}$$

Els càlculs del Speedup respecte a l'execució seqüencial per cada cas estudiat són les següents:

- Sense opcions de millora

$$acceleracio = \frac{1 * 5}{1 + \frac{144+15\ detencions}{196\ cicles}} = 2.76$$

- Forwarding

$$acceleracio = \frac{1 * 5}{1 + \frac{145\ detencions}{139\ cicles}} = 2.44$$

- Delay slot

$$acceleracio = \frac{1 * 5}{1 + \frac{111\ detencions}{152\ cicles}} = 2.89$$

- Delay slot + forwarding

$$acceleracio = \frac{1 * 5}{1 + \frac{120\ detencions}{139\ cicles}} = 2.68$$

Com es pot veure aplicant el delay slot aconseguim l'acceleració més gran, ja que és el que té l'índex de detencions/cicles més petit.

3 Conclusió

Veient els resultats obtinguts durant el laboratori, i els resultats obtinguts al fer el càlcul de l'acceleració podem decidir quina execució ens interessa més.

Creiem que la millor execució és la del Delay slot, sense Forwarding, ja que és la que aconsegueix una millor acceleració, diferenciant-se bastant de les execucions amb forwarding. A més a més, la diferència entre el CPI del delay slot amb les dues execucions amb Forwarding no es diferencien massa. Un altre dels criteris a tenir en compte en la decisió és en nombre de "stalls" que ocasionen aquestes execucions, i en aquest cas, el Delay és la que menys en genera.

Per tot el que hem comentat en el paràgraf anterior, creiem que el Delay slot, sense el Forwarding, és la millor opció, ja que és ràpid, i no genera tants conflictes com la resta.

4 Annex

	Sense opcions de millora	Forwarding	Delay Slot	Forwarding + Delay Slot
Nombre d'instruccions	37	37	37	37
Nombre de cicles	196	139	152	139
CPI	5.297	3.757	4.108	3.757
RAW stalls	144	140	111	115
WAW stalls	0	0	0	0
WAR stalls	0	4	0	4
Structural stalls	15	1	0	1

Taula 1: Resultats de l'experiment