MATERIAL INDIVIDUAL F-III

PROJE CTE



2020

Rendiment en arquitectures RISC

F-III. Estudi dels riscos de la segmentació

Arquitectura dels Computadors

Grau en Enginyeria Informàtica Dpto. Tecnologia Informàtica i Computació Universitat d'Alacant

Material individual de la Fase III

RENDIMENT EN ARQUITECTURES RISC

I. OBJECTIUS

La nova Faselll de l'assignatura se centra en l'estudi del rendiment d'arquitectures RISC, en particular se centra en l'anàlisi dels riscos de la segmentació i l'avançament d'operacions.

L'objectiu és estudiar conceptes relatius a la segmentació d'una màquina. Concretament analitzarem l'arquitectura MIPS64. Per a això, se simularà l'execució de programes davant diferents estratègies de resolución dels riscos de segmentació

Per a la realització d'aquesta pràctica s'utilitzarà l'eina WinMIPS64. Amb aquesta aplicació, a més d'escriure i executar programes en assemblador, podrem visualitzar el progrés d'execució de les instruccions a través de les diferents etapes de la ruta de dades segmentada.

En l'annex 1 es proporciona un resum del conjunt d'instruccions MIPS i en l'annex 2 es dóna una breu descripció sobre la utilització del simulador.

En el tema 4 de teoria s'estudia una arquitectura segmentada per al processador basada en el DLX. El DLX és una versió simplificada del MIPS que proporciona un model arquitectural senzill que facilita la demostració dels principis de la segmentació i en ella els únics riscos per dependència de dades que poden ocórrer són els RAW (Read after write), és a dir, riscos que sorgeixen perquè una instrucció intenta llegir un operand abans que una instrucció haja escrit en ell. És important recalcar que en aquesta pràctica no s'estudiarà el DLX i per tant en aquesta nova arquitectura sí poden aparèixer riscos WAR (Write after Read) i WAW (Write after Write) encara que no els estudiarem. Aquest simulador tampoc usa la tècnica d'implementació basada en dividir el cicle de tal manera que les lectures del banc de registres es fan en la segona meitat del cicle i les escriptures en la primera meitat.

II. DESENVOLUPAMENT DE LA PRÀCTICA

La pràctica consisteix en un exercici guiat en el qual es proposen 5 programes senzills que haureu d'introduir en el simulador WinMIPS64 per estudiar com s'executen les instruccions en una màquina segmentada, comprovar el que succeeix quan ocorre un risc de segmentació i analitzar la seva possible solució.

INTRODUCCIÓN A la SEGMENTACIÓ

En aquesta part de la pràctica es pretén que l'estudiant es familiaritze amb l'entorn del simulador, execute un programa i reflexione sobre els beneficis que aporta la segmentació a les arquitectures RISC.

Programa 1.

Introdueix el següent codi en el simulador. Per a això genera un arxiu de text amb aquest codi i gravaho amb l'extensió .s, després introdueix-ho amb el simulador a través del comando "Load" del menú "File":

```
.data
num: .word 7
num2: .word 8

.code

ld r2, num(r0)
   dadd r3, r8, r9
   dsub r10, r5, r6
   dsll r1, r4, 1
   sd r4, num2(r0)
```

- a) Indica quina funció realitza cadascuna de les instruccions del codi
- b) Indica què variables de dades usa aquest programa i on es mostren en el simulador
- c) Executeu el programa en el simulador amb l'opció **Configure/Enable Forwarding** deshabilitada. Analitzar cicle a cicle el seu funcionament amb l'opció "Single cycle" del menú Execute o bé pressionant F7 successivament. Examineu les diferents finestres que es mostren en el simulador i respondre:
 - En quin cicle del total del programa es llegeix la dada que hi ha en la variable num? a quina fase correspon de la instrucció load?
 - En quin cicle del total del programa s'escriu la dada en la variable num2? a quina fase correspon de la instrucció store?
 - En quin cicle del total del programa s'escriu una dada en el registre r10? a quina fase correspon de la instrucció de resta?
 - Després de l'execució, es produeix alguna detenció en la llera? Raona la teva resposta.
 - Quina és la mitjana de cicles per instrucció (CPI) en l'execució d'aquest programa?

RISCOS DE LA SEGMENTACIÓN PER DEPENDÈNCIA DE DADES

En aquesta part de la pràctica es pretén que l'estudiant estudie els riscos de la segmentació per dependència de dades. Es pretén que analitze la necessitat d'utilitzar maquinari d'avançament per reduir el nombre de detencions.

Programa 2.

Introdueix el següent codi en el simulador.

```
.data
   A: .word 8
   B: .word 6
   C: .word 3
   D: .word 0,0,0,0
.code
   Ld r1, A(r0)
   ld r2, B(r0)
   ld r3, C(r0)
   xor r5, r5, r5
   dadd r6, r2, r3
   dadd r7, r6, r3
   dadd r8, r7, r2
   sd r6, D(r5)
   dadd r5, r5, r1
   sd r7, D(r5)
   dadd r5, r5, r1
   sd r8, D(r5)
   daddi r9, r5, 8
   1d r10, D(r5)
   sd r10, D(r9)
halt.
```

- a) Comenta cadascuna de les línies del codi i explica breument què realitza el codi
- b) Executa el programa en el simulador amb l'opció **Configure/Enable Forwarding** deshabilitada. Analitzar pas a pas el seu funcionament, examineu les diferents finestres que es mostren en el simulador i respon:
 - Quantes detencions RAW apareixen?
 - Quines instruccions estan generant les mateixes (stalls) en la llera?
 - Explica per què es produeixen les deteccions tenint en compte les instruccions i els registres implicats. Per a això, pensa en:
 - Per què es produeix la primera detenció en el cicle 6?. Amb quin registre i instrucció?
 - És el mateix risc que es produeix en la segona detenció?. Per què tenim 2 detencions i només 1 en la primera detenció?
 - Són tots els mateixos tipus de riscos? Compara el primer risc produït amb l'última instrucció que es deté (sd)
 - Quin és la mitjana de cicles per instrucció (CPI) en l'execució d'aquest programa sota aquesta configuració?

- b) Una forma de solucionar les parades per dependència de dades és utilitzar l'avançament d'operands o Forwarding. Executa novament el programa anterior amb l'opció *Enable Forwarding* habilitada i respon:
 - Per què no es presenta cap parada en aquest cas? Explica la millora tenint en compte:
 - Com s'ha solucionat el primer risc? Des de quina unitat funcional s'ha avançat la dada per resoldre el risc que es produïa en el cicle 6?.
 - Es resol de la mateixa forma la segona detenció anterior?. Des de quina unitat s'avança?
 - Compara la solució del primer risc produït amb la de l'últim (sd)
 - Quin és la mitjana de cicles per instrucció (CPI) en aquest cas? Comparar amb l'anterior.

RISCOS DE LA SEGMENTACIÓ QUE REQUEREIXEN DETENCIONS

En aquesta part de la pràctica es pretén que l'estudiant estudie els riscos de la segmentació per dependència de dades que requereixen detencions. S'analitzarà la necessitat d'utilitzar el maquinari d'interbloqueig en determinades situacions per detenir la llera i la possibilitat d'utilitzar l'avançament per reduir el nombre de detencions. També s'analitzarà la possibilitat de reorganitzar el codi per part del compilador per eliminar els riscos que requereixen detencions.

Programa 3.

Introdueix el següent codi en el simulador. Per a això genera un arxiu de text amb aquest codi i gravaho amb l'extensió .s, després introdueix-ho amb el simulador a través del comando "Load" del menú "File":

```
.data
A: .word32 2
B: .word32 3
C: .word32 0
D: .word32 4
I: .word32 5
F: .word32 0
.code
     lw r1, A(r0)
    lw r2, B(r0)
     dadd r3, r1, r2
     sw r3, C(r0)
     lw r4, D(r0)
     lw r5, I(r0)
     dadd r6, r4, r5
     sw r6, F(r0)
halt.
```

- c) Indica quina funció realitza cadascuna de les instruccions del codi. Podries expressar el codi mitjançant dues instruccions d'un llenguatge d'alt nivell?
- d) Indica quins tipus de dades s'estan utilitzant i relaciona-ho amb les instruccions que s'utilitzen.
- e) Executeu el programa en el simulador amb l'opció **Configure/Enable Forwarding** deshabilitada. Analitzar cicle a cicle el seu funcionament amb l'opció "Single cycle" del menú Execute o ben pressionant F7 successivament. Examineu les diferents finestres que es mostren en el simulador i respondre:
 - Què ocorre en els cicles 5, 6, 13 i 14 amb les instruccions dadd?
 - Què ocorre en els cicles 8, 9, 16, 17 amb les instruccions sw?
 - Quants cicles es consumeixen en total i quants d'ells són detencions?
 - Calcula el CPI?
- f) Executeu el programa en el simulador amb l'opció *Configure/Enable Forwarding* habilitada. Analitzar cicle a cicle el seu funcionament amb l'opció "Single cycle" del menú Execute o bé pressionant F7 successivament. Examineu les diferents finestres que es mostren en el simulador i respondre:
 - Què ocorre en els cicles 6 i 11 amb les instruccions dadd? Es produeixen avançaments? Si escau indica quins.

- Es produeixen avançaments en els cicles 8 i 13 vinculats a les instruccions sw? Si escau indica quins.
- Quants cicles es consumeixen en total i quants d'ells són detencions?
- Calcula el CPI?
- c) Proposa una reorganització del codi per reduir el nombre de detencions mantenint el resultat final del programa sobre els registres i memòria.
 - Escriu el codi reorganitzat.
 - Amb l'avançament activat:
 - Quants cicles es consumeixen en total i quants d'ells són detencions?
 - Calcula el CPI?

RISCOS DE CONTROL

En aquesta part de la pràctica es pretén estudiar els riscos de control. En MIPS64 els salts condicionals (beq, beqz i bnea, etc) requereixen comprovar si dos registres són iguals o si un registre és igual a zero. En MIPS64 és possible completar aquesta decisió al final del cicle ID movent els test d'igualtat o desigualtat a aquest cicle. A més, per aprofitar aquest avançament en la decisió de salt es calculen prèviament els dos valors del PC, l'efectiu i el no efectiu. Per a això s'afegeix un sumador addicional en l'etapa ID que calcula la decisió de salt ja que d'una altra manera no estaria disponible fins a l'etapa EXE. D'aquesta manera només serà necessari un cicle de detenció en els salts.

A més, el MIPS64 prediu el salt com no efectiu. En aquest esquema es permet al maquinari que continue com si el salt no s'executés, és a dir, es realitza la cerca d'instruccions com si no ocorregués res. Si el salt és no efectiu (es determina en ID), simplement es continua amb la instrucció que s'acaba de llegir en l'etapa IF. Si resulta que el salt és efectiu es deté la segmentació, es reprèn la cerca d'instruccions en l'adreça de salt i es desfan els canvis d'estat netejant la segmentació.

En l'exemple que segueix es veurà com prediu el MIPS els salts com no efectius observant un cicle de detenció quan aquests siguin efectius. A més s'estudiarà com reduir les penalitzacions dels salts en la segmentació mitjançant els salts retardats. Amb aquesta tècnica s'executen instruccions independents del salt en els cicles de retard (delay slot). Aquestes instruccions van a executar-se sempre, independentment que el salt sigui efectiu o no, per tant s'haurà d'incloure una instrucció a continuació del salt que es tingui seguretat que es pot executar independent del resultat de la condició de salt.

Programa 4.

Introdueix el següent codi en el simulador:

```
.data
  cant: .word 8
  dades: .word 1, 2, 3, 4, 5, 6, 7, 8
  res: .word 0
.code
     dadd r1, r0, r0
     ld r2, cant(r0)
     loop: ld r3, dades(r1)
          daddi r2, r2, -1
          dsll r3, r3, 1
        sd r3, res(r1)
        daddi r1, r1, 8
        bnez r2, loop
     halt
```

- a) Indica quin és l'objectiu del codi i quin serà el resultat.
- b) Identifica els riscos per dependència de dades que poden aparèixer.
- c) Executa el programa en el simulador amb l'opció Configure/Enable Forwarding deshabilitada. Analitzar cicle a cicle el seu funcionament amb l'opció "Single cycle" del menú Execute o bé pressionant F7 successivament. Examina les diferents finestres que es mostren en el simulador i respon a les següents qüestions:
 - En quin cicle ocorre la primera parada per dependència de dades? En quina instrucció?
 - En quin cicle es produeixen dues detencions en la mateixa instrucció? A què es deu?

- Observa que ocorre amb la instrucció de salt. En quin cicle ocorre la primera parada per salt efectiu?
- Executa el programa complet i observa que l'última vegada que s'executa el salt no hi ha parada a causa que el salt ha estat no efectiu.
- Quants cicles tarda el programa a executar-se? Quants d'ells són detencions? Quin és el CPI?
- d) Executa el programa en el simulador amb l'opció **Configure/Enable Forwarding** habilitada. Analitza cicle a cicle el seu funcionament amb l'opció "Single cycle" del menú Execute o bé pressionant F7 successivament. Examina les diferents finestres que es mostren en el simulador i respon:
 - Quantes dependències de dades s'han aconseguit solucionar?
 - Quantes parades per riscos de control s'han produït?
 - Quants cicles de rellotge tarda el programa a executar-se? Quin és el CPI?
- e) Executa el programa en el simulador amb l'opció **Configure/Enable Delay Slot** habilitada. Analitzar cicle a cicle el seu funcionament amb l'opció "Single cycle" del menú Execute o bé pressionant F7 successivament. Examineu les diferents finestres que es mostren en el simulador i respon a les següents qüestions:
 - Quants cicles s'han consumit? Per què ha seguit executant-se la instrucció halt a partir del cicle 9?
 - Modifica el programa canviant la instrucció daddi r1, r1, 8 de lloc i col·locant-la després del salt (en el delay slot). Executa de nou el programa i observa què ocorre. És correcte el resultat?
 - Quantes parades per risc de control s'han eliminat? S'ha incrementat el nombre d'instruccions a executar? Quants cicles de rellotge s'han consumit?
 - Quin és el CPI?
 - Quina altra instrucció s'hagués pogut col·locar en el delay slot? Hauria estat necessari realitzar algun altre canvi en el codi?
- f) El codi del programa 5 realitza la mateixa funció que el programa 4 que acabes d'estudiar:

Programa 5.

Introdueix el següent codi en el simulador:

```
.data
  cant: .word 8
  dades: .word 1, 2, 3, 4, 5, 6, 7, 8
  res: .word 0
.code
  ld r2, cant(r0)
  dadd r1, r0, r0
  loop: beqz r2, fi
    ld r3, dades(r1)
    daddi r2, r2, -1
    dsll r3, r3, 1
    sd r3, res(r1)
    daddi r1, r1, 8
```

- Quines diferències observes?
- Executa el programa en el simulador amb l'opció Configure/Enable Delay Slot deshabilitada. En quines instruccions i en quins casos es produeix detenció per risc de control?
- Compara les estadístiques que s'obtenen en el nombre de cicles i CPI amb les obtingudes al programa 4 amb les opció Configure/Enable Forwarding habilitada i deshabilitada.
- Col·loca una instrucció vàlida en el delay slot i executa amb l'opció Configure/Enable
 Delay Slot habilitada. Quants cicles es consumeixen? Quin és el CPI?.
- Quines conclusions pots extreure en comparar el nombre de cicles i CPI dels dos programes estudiats?

III. AVALUACIÓ

A causa de la modificació en el contingut de la Fase III de les pràctiques de l'assignatura, la avaluació de la part pràctica s'establirà d'acord amb el següent criteri:

NP = 0.1x Fase I + 0.3 x NotaGrupalFaseII + 0.2 x NotaIndividualFaseII + 0.4 x NotaFaseIIIModificada