RA FS 12 Repetitionsserie: Teil MIPS

Samuel Bucheli, Gian Calgeer, Judith Fuog

2 MIPS

2.1 Grosse Konstanten

Warum ist es nicht möglich, mit dem 32-Bit MIPS-Instruction Set eine Konstante der Länge 32 Bit mit genau einem Maschinenbefehl zu laden?

2.2 Jump

Erklären Sie den Unterschied zwischen dem Befehl `jal` und dem Befehl `j`.

2.3 MIPS ISA

Schreiben Sie den folgenden C-Programmauschnitt in Assembler um:

```
/* ...something... */
array[4] += 1638410; // Datentyp des Arrays: int
/* ...something... */
```

Nehmen Sie an, dass sich die Startadresse des Arrays im Register `$s1` befindet und dass es sich um eine 32-Bit Architektur handelt.

2.4 MIPS Multiplikation

Schreiben Sie den folgenden C-Programmauschnitt in Assembler um, verwenden Sie dabei ausschliesslich Additionsbefehle und (bedingte und unbedingte) Verzweigungen.

Nehmen Sie an, `i` stehe in Register `$s2` und `j` in Register `$s1` und dass `i` und `j` positive Ganzzahlwerte enthalten.

```
/* ...something... */
i *= j;
/* ...a bit more of something ...*/
```

2.5 `$at` Register

Wozu gibt es das Register `$at`, welches für den Assembler reserviert ist?

2.6 Noch mehr grosse Konstanten

Um einen festen 32bit Wert in ein Register zu laden, braucht es die beiden Befehle `lui` und `ori`. Kann man anstelle von `ori` auch `addi` verwenden? Begründen Sie Ihre Antwort!

2.7 Singlecycle vs. Multicycle

Erläutern Sie die Unterschiede zwischen Singlecycle und Multicycle Datapath Approach.

2.8 Sprünge

Weshalb ersetzt der Assembler in einigen Fällen den Befehl

```
1 beq $s0, $s1, L1
```

durch die Befehlssequenz

```
1 bne $s0, $s1, L2
2 j L1
3 L2:
```

2.9 Schleifen

Gegeben sei folgende for-Schleife (`i` ist ein Integer und `a` ist ein Integerarray):

```
1 for (i =0; i<3; i+=1) {a[i]=0}
```

(a) Wie lautet der Assemblercode, der vom Compiler erzeugt wird? Nehmen Sie an die Adresse vom Array `a` sei im Register `$s0` abgespeichert. Der Assemblercode muss die Schleife mithilfe eines Sprungbefehls implementieren.

(b) Welcher Befehl aus dem Assemblercode der vorherigen Teilaufgabe wird vom Compiler in den Delayslot gelegt, wenn der Code auf einer Pipeline- Architektur laufen soll?

2.10 Jump Register

Wie muss man die MIPS Single-Cycle Architektur aus der Vorlesung erweitern, damit der `jr` Befehl unterstützt wird?

2.11 Performance Berechnung

Angenommen wir haben einen Prozessor mit einer Ideal CPI von 2.0. Die Taktgeschwindigkeit betrage 2 GHz. Ein Hauptspeicherzugriff benötigt 100 ns. Der Anteil an Load/Store Befehlen betrage 25%. Die Missrate für den Instruction Cache betrage 2%, für den Data Cache 5%.

(a) Wie gross ist die Gesamt-CPI?

(b) Wie gross ist der Anteil (in %) an Memory Stalls an der gesamten Ausführungszeit?

(c) Wie verändert sich dieser Anteil, wenn der Takt auf 4 GHz erhöht wird?

(d) Nehmen Sie an, es wird zusätzlich ein Level 2 Cache hinzugefügt mit einer Zugriffszeit von 25 Takten für Treffer und Fehlzugriff. Die Fehlerzugriffsrate auf den Hauptspeicher wird dabei auf 0.5% reduziert. Welches ist die Gesamt CPI dieses System (bei einer Ideal CPI von 2.0)?

2.12 Page mode

Beschreiben Sie wie ein DRAM im Page Mode arbeitet.

2.13 Memory

Erklären Sie den Unterschied zwischen

- SRAM und DRAM
- DRAM und SDRAM

2.14 Diverses

Welche der folgenden Aussagen sind wahr?

- Forwarding wird bei write before read Abhängigkeiten eingesetzt.
- Forwarding wird bei write before write Abhängigkeiten eingesetzt.
- Register renaming wird bei write before read Abhängigkeiten eingesetzt.
- Die Ausführung eines einzelnen `sw` Befehls dauert länger in einer multi-cycle Architektur als in einer single-cycle Architektur.
- BTB heisst “Bit Transfer Buffer”.
- Die Control Unit einer Pipeline Architektur ist komplizierter als bei einer multi-cycle Architektur.
- Der Compiler-Entwurf ist bei statischer Mehrfachzuordnung (VLIW) einfacher als bei dynamischer Mehrfachzuordnung
- Der Prozessorentwurf ist bei statischer Mehrfachzuordnung (VLIW) einfacher als bei dynamischer Mehrfachzuordnung

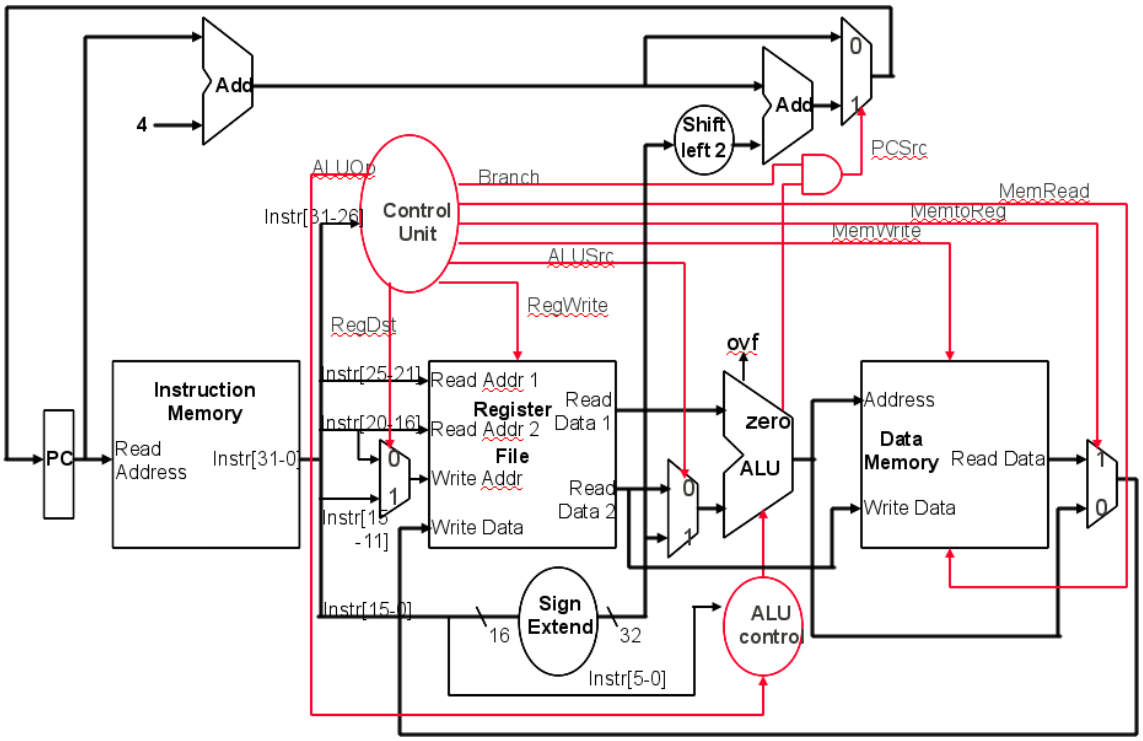
2.15 `lw` in Single Cycle Architektur

Unten ist die Schaltung einer Single Cycle Architektur abgebildet. Nehmen Sie an, es wird ein `lw` Befehl ausgeführt.

(a) Geben Sie für jeden MUX an, ob er auf 0 oder 1 steht. Geben Sie auch die Don't Care Fälle an.

(b) Weshalb braucht es für das Data-Memory ein Read Signal?

(c) Weshalb wird Instr[5-0] vor dem SignExtend abgezweigt und nicht nachher?



2.16 Caches

(a) Was bedeuted die Einführung eines L2 caches für die hit rate, hit time und miss penalty der L1 caches?

(b) Was ist besser für den L2 cache: eine grosse hit rate oder eine kleine hit time?

(c) Wie muss ein Cache aufgebaut sein, damit er "Spatial Locality" ausnutzen kann?

(d) In welchen Caches wird "Spatial Locality" eingesetzt? In welchen Caches wird "Temporal Locality" eingesetzt?

2.17 MIPS Code und Pipelining

Gegeben sei folgender MIPS Code.

```
1  lw $t0, 0($s1) # $t0 = Feldelement
2  addi $t0, $t0, 1 # Addiere 1
3  sw $t0, 0($s1) # Speichere Ergebnis
4  addi $s1, $s1, -4 # Dekrementiere Zeiger
```

Gegeben sei ein Prozessor mit einer fünfstufigen Pipeline.

(a) Wieviele Taktzyklen benötigt der Code zur Ausführung?

(b) Wird Forwarding benötigt? Falls ja, an welchen Stellen? Wird benötigt, dass Register in der ersten Hälfte des Taktes geschrieben und in der zweiten Hälfte gelesen werden? Falls ja, an welchen Stellen?

(c) Kann man den Code umstellen, so dass er weniger Takte zur Ausführung benötigt?