

Advanced Computer Architectures
Exercises

Christian Rossi

Academic Year 2023-2024

Abstract

The course topics are:

- Review of basic computer architecture: the RISC approach and pipelining, the memory hierarchy.
- Basic performance evaluation metrics of computer architectures.
- Techniques for performance optimization: processor and memory.
- Instruction level parallelism: static and dynamic scheduling; superscalar architectures: principles and problems; VLIW (Very Long Instruction Word) architectures, examples of architecture families.
- Thread-level parallelism.
- Multiprocessors and multicore systems: taxonomy, topologies, communication management, memory management, cache coherency protocols, example of architectures.
- Stream processors and vector processors; Graphic Processors, GP-GPUs, heterogeneous architectures.

Contents

| | | |
|----------|----------------------------|----------|
| 1 | Exercise session I | 1 |
| 1.1 | Exercise one | 1 |
| 1.2 | Exercise two | 1 |
| 1.3 | Exercise three | 2 |
| 1.4 | Exercise four | 3 |
| 2 | Exercise session II | 4 |
| 2.1 | Exercise one | 4 |

CHAPTER 1

Exercise session I

1.1 Exercise one

Assessing the impact of modifications on performance:

1. Substituting a hardware component with a faster alternative.
2. Incorporating multiple parallel systems for executing independent tasks.

Solution

1. By scaling up: response time decreases, while throughput increases.
2. Through scaling out: throughput experiences an increase. Response time will only escalate if a queue was present, awaiting computing resources.

1.2 Exercise two

Let's consider two CPUs: CPU1 and CPU2. CPU1 operates with a clock cycle of 2 ns , while CPU2 has an operating frequency of 700 MHz . Given the frequencies of occurrence of instructions for both CPUs:

| Operation type | Frequency | CPU1 cycle | CPU2 cycle |
|----------------|-----------|------------|------------|
| A | 0.3 | 2 | 2 |
| B | 0.1 | 3 | 3 |
| C | 0.2 | 4 | 3 |
| D | 0.3 | 2 | 2 |
| E | 0.1 | 4 | 3 |

1. Calculate the average CPI for CPU1 and CPU2.
2. Determine which CPU is the fastest.

Solution

1. The CPI (Cycle Per Instruction) is calculated as:

$$\text{CPI} = \frac{\text{clock cycles}}{\text{instruction}}$$

The average CPI is then obtained by:

$$\sum_{i=1}^n \text{CPI}_i \cdot F_i$$

Where, $F_i = \frac{I_i}{\text{instruction count}}$. For CPU1:

$$\text{CPI}_1 = 0.3 \cdot 2 + 0.1 \cdot 3 + 0.2 \cdot 4 + 0.3 \cdot 2 + 0.1 \cdot 4 = 2.7$$

For CPU2:

$$\text{CPI}_2 = 0.3 \cdot 2 + 0.1 \cdot 3 + 0.2 \cdot 3 + 0.3 \cdot 2 + 0.1 \cdot 3 = 2.4$$

2. We have that:

$$\begin{aligned} \frac{\text{EXE}_{\text{CPU1}}}{\text{EXE}_{\text{CPU2}}} &= \left(\frac{\text{IC}_1 \cdot \text{CPI}_1}{F_1} \right) \left(\frac{F_2}{\text{IC}_2 \cdot \text{CPI}_2} \right) \\ &= \frac{\text{IC}_1 \cdot \text{CPI}_1 \cdot F_2}{F_1 \cdot \text{IC}_2 \cdot \text{CPI}_2} \\ &= \frac{\text{CPI}_1 \cdot F_2}{F_1 \cdot \text{CPI}_2} \\ &= \frac{2.7 \cdot 700 \text{ MHz}}{2.4 \cdot 500 \text{ MHz}} \\ &= 1.575 \end{aligned}$$

Hence, CPU2 is approximately 1.575 times faster than CPU1.

1.3 Exercise three

The speed of image processing on FPGA is 2.86 times faster than on a CPU. The power consumption of an FPGA is 100 W, while that of the CPU is 30.85 W. We want to achieve a speedup of 2 with the addition of an FPGA.

Solution

To achieve a speedup of 2 with the addition of an FPGA, we use Amdahl's law:

$$S_{\text{overall}} = \frac{1}{(1 - F_{\text{enhanced}}) + \frac{F_{\text{enhanced}}}{S_{\text{enhanced}}}}$$

Given $S_{\text{overall}} = 2$, we solve for F_{enhanced} :

$$2 = \frac{1}{(1 - F_{\text{enhanced}}) + \frac{F_{\text{enhanced}}}{2.86}} \rightarrow F_{\text{enhanced}} = 0.768$$

Therefore, with 76.8% of the processing offloaded to the FPGA, a speedup of 2 can be achieved.

1.4 Exercise four

Consider the following assembly program:

```
LW      $1, OFF($2)
ADDI    $3, $1, 4
SUB      $4, $1, $2
ADDI    $2, $1, -8
SW      $4, OFF($2)
```

No optimizations are applied in the MIPS pipeline. The processor operates with a clock cycle of 2 ns .

- Draw the pipeline schema and highlight potential hazards.
- Illustrate the actual execution with stall cycles inserted.
- Calculate Instruction Count (IC), CPI, and MIPS.

Solution

1. The pipeline schema is:

| Instruction | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------------------------|---|---|---|---|---|---|---|---|---|
| <i>LW \$1, OFF (\$2)</i> | F | D | E | M | W | | | | |
| <i>ADDI \$3, \$1, 4</i> | | F | D | E | M | W | | | |
| <i>SUB \$4, \$1, \$3</i> | | | F | D | E | M | W | | |
| <i>ADDI \$2, \$1, -8</i> | | | | F | D | E | M | W | |
| <i>SW \$5, OFF (\$2)</i> | | | | | F | D | E | M | W |

The potential hazards are:

- Instruction 1 writes to register \$1, and instructions 2, 3, and 4 read from it.
- Instruction 2 writes to register \$3, and instruction 3 reads from it.
- Instruction 4 writes to register \$2, and instruction 5 reads from it.

2. The real execution with stall cycles inserted is:

| Instruction | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|--------------------------|---|---|----------|----------|---|----------|----------|---|---|----------|----------|----|----|----|----|
| <i>LW \$1, OFF (\$2)</i> | F | D | E | M | W | | | | | | | | | | |
| <i>ADDI \$3, \$1, 4</i> | | F | <u>S</u> | <u>S</u> | D | E | M | W | | | | | | | |
| <i>SUB \$4, \$1, \$3</i> | | | | | F | <u>S</u> | <u>S</u> | D | E | M | W | | | | |
| <i>ADDI \$2, \$1, -8</i> | | | | | | | | F | D | E | M | W | | | |
| <i>SW \$5, OFF (\$2)</i> | | | | | | | | | F | <u>S</u> | <u>S</u> | D | E | M | W |

3. The performance metrics are:

$$\begin{aligned}
 \text{IC} &= 5 \\
 \text{CPI} &= \frac{\text{CCs}}{\text{IC}} = \frac{15}{5} = 3 \\
 \text{MIPS} &= \frac{\text{clock frequency}}{\text{CPI} \cdot 10^6} = \frac{0.5 \cdot 10^9}{3 \cdot 10^6} = 166
 \end{aligned}$$

CHAPTER 2

Exercise session II

2.1 Exercise one

Consider the given program:

```
i1: add $t1, $t0, $t1
i2: add $t2, $t1, $t2
i3: subi $t0, $t2, 1
i4: sw $t0, 0x00BB($t2)
i5: beq $t0, $t2, 0x0089
```

Assuming no forwarding, register file access with read/write optimization, and control hazards solved in the instruction decode stage (ID):

1. Define all conflicts/dependencies and analyze whether they cause hazards and the theoretical amount of stalls.
2. Draw the effective pipeline schema:
3. Draw the effective pipeline schema assuming EX/EX, MEM/EX, and MEM/MEM forwarding paths are available.
4. Draw the effective pipeline schema assuming that the previous forwarding paths with also EX/ID are available.

Solution

1. The potential issues are:

| Instruction number | Instruction dependency | Register involved | Hazard | Stalls |
|--------------------|------------------------|-------------------|--------|--------|
| i2 | i1 | \$t1 | yes | 2 |
| i3 | i2 | \$t2 | yes | 2 |
| i4 | i2 | \$t2 | yes | 1 |
| i4 | i3 | \$t0 | yes | 2 |
| i5 | i2 | \$t2 | no | 0 |
| i5 | i3 | \$t0 | yes | 1 |

2. The requested pipeline schema with stalls is as follows:

| Instruction | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 | C11 | C12 | C13 | C14 | C15 |
|-------------------------------|----|----|----------|----------|----|----------|----------|----|----------|----------|-----|-----|-----|-----|-----|
| <i>add \$t1, \$t0, \$t1</i> | IF | ID | EX | M | WB | | | | | | | | | | |
| <i>add \$t2, \$t1, \$t2</i> | | IF | <u>S</u> | <u>S</u> | ID | EX | M | WB | | | | | | | |
| <i>subi \$t0, \$t2, 1</i> | | | <u>S</u> | <u>S</u> | IF | <u>S</u> | <u>S</u> | ID | EX | M | WB | | | | |
| <i>sw \$t0, 0x00BB (\$t2)</i> | | | | | | <u>S</u> | <u>S</u> | IF | <u>S</u> | <u>S</u> | ID | EX | M | WB | |
| <i>beq \$t0, \$t2, 0x0089</i> | | | | | | | | | <u>S</u> | <u>S</u> | IF | ID | EX | M | WB |

3. The requested pipeline schema with forwarding is as follows:

| Instruction | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 |
|-------------------------------|----|----|----|----|----|----------|----|----|----|-----|
| <i>add \$t1, \$t0, \$t1</i> | IF | ID | EX | M | WB | | | | | |
| <i>add \$t2, \$t1, \$t2</i> | | IF | ID | EX | M | WB | | | | |
| <i>subi \$t0, \$t2, 1</i> | | | IF | ID | EX | M | WB | | | |
| <i>sw \$t0, 0x00BB (\$t2)</i> | | | | IF | ID | EX | M | WB | | |
| <i>beq \$t0, \$t2, 0x0089</i> | | | | | IF | <u>S</u> | ID | EX | M | WB |

4. The requested pipeline schema with forwarding and EX/ID is:

| Instruction | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 |
|-------------------------------|----|----|----|----|----|----|----|----|----|
| <i>add \$t1, \$t0, \$t1</i> | IF | ID | EX | M | WB | | | | |
| <i>add \$t2, \$t1, \$t2</i> | | IF | ID | EX | M | WB | | | |
| <i>subi \$t0, \$t2, 1</i> | | | IF | ID | EX | M | WB | | |
| <i>sw \$t0, 0x00BB (\$t2)</i> | | | | IF | ID | EX | M | WB | |
| <i>beq \$t0, \$t2, 0x0089</i> | | | | | IF | ID | EX | M | WB |