Homework 2 Yash Sinha

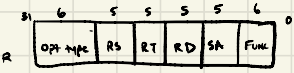
2.16 [5] <§2.5> Assume that we would like to expand the MIPS register file to 128 registers and expand the instruction set to contain four times as many instructions.

2.16.1 [5] <§2.5> How would this affect the size of each of the bit fields in the R-type instructions?

2.16.2 [5] <§2.5> How would this affect the size of each of the bit fields in the I-type instructions?

2.16.3 [5] <§§2.5, 2.10> How could each of two propose changes decrease the size of an MIPS assembly program? On the other hand, how could the proposed change increase the size of an MIPS assembly program?

a) An R type instruction has the following characteristics



# of registers in new MIPS register file = 128 bits $\Rightarrow$ $\log_2 128 = 7$ bits needed to address register

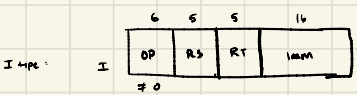
4 times as many instructions $\Rightarrow$ $\log_2 4 = 2$ bits additionally needed for opcode $\Rightarrow$ opcode = 6+2 = 8 bits

Opcode increases to 8 bits

RS, RT, RD increase to 7 bits

SA and func stay the same

b) An I type instruction has the following characteristics



As in (a), # of registers in new MIPS reg file = 128 bits $\Rightarrow$ 7 bits needed to address register

4 times as many instructions $\Rightarrow$ opcode = 8 bits

Opcode increases to 8 bits

RS, RT, RD increase to 7 bits

IMM stays the same

c) The increase in registers and size of instruction set could result in more complicated instructions being run under less instructions, which would result in less spillage, thereby, decreasing the size of the program.

However, at the same time, the size of instructions would grow due to more bits being available in the opcode and registers.

2.39 Assume for a given processor the CPI of arithmetic instructions is 1, the CPI of load/store instructions is 10, and the CPI of branch instructions is 3. Assume a program has the following instruction breakdowns: 500 million arithmetic instructions, 300 million load/store instructions, 100 million branch instructions.

2.39.1 [5] <§§1.6, 2.13> Suppose that new, more powerful arithmetic instructions are added to the instruction set. On average, through the use of these more powerful arithmetic instructions, we can reduce the number of arithmetic instructions needed to execute a program by 25%, and the cost of increasing the clock cycle time by only 10%. Is this a good design choice? Why?

2.39.2 [5] <§§1.6, 2.13> Suppose that we find a way to double the performance of arithmetic instructions. What is the overall speedup of our machine? What if we find a way to improve the performance of arithmetic instructions by 10 times?

a)

$CPI_{ar} = 1 \quad CPI_{load} = 10 \quad CPI_{branch} = 3$

$500 \cdot 10^6$ instructions $\quad 300 \cdot 10^6$ instructions $\quad 100 \cdot 10^6$ instructions

$$ET = \left[\sum_{i=0}^{3} \text{instruction count} \cdot CPI_i \right] \cdot \text{cycle time}$$

$$ET_0 = \left[\text{instr}_{ar} \cdot CPI_{ar} + \text{instr}_{load} CPI_{load} + \text{instr}_{branch} CPI_{branch} \right] \cdot \text{cycle time}_0$$

$$= \left[(500 \cdot 10^6 \cdot 1) + (300 \cdot 10^6 \cdot 10) + (100 \cdot 10^6 \cdot 3) \right] \cdot \text{cycle time}_0$$

$$= \left[500 \cdot 10^6 + 3000 \cdot 10^6 + 300 \cdot 10^6 \right] \cdot \text{cycle time}_0$$

$$= 3800 \cdot 10^6 \text{ cycles} \cdot \text{cycle time}_0$$

$$ET_1 = \left[.75 \cdot \text{instr}_{ar} CPI_{ar} + \text{instr}_{load} CPI_{load} + \text{instr}_{branch} CPI_{branch} \right] \cdot \text{cycle time}_1$$

$$= \left[.75 \cdot 500 \cdot 10^6 \cdot 300 \cdot 10^6 \cdot 10 + 100 \cdot 10^6 \cdot 3 \right] \cdot \text{cycle time}_1$$

$$= \left[375 \cdot 10^6 + 3000 \cdot 10^6 + 300 \cdot 10^6 \right] \cdot \text{cycle time}_1$$

$$= 3675 \cdot 10^6 \cdot \text{cycle time}_1$$

$$= 3675 \cdot 10^6 \cdot (1.1 \cdot \text{cycle time}_0)$$

$$= 4042.5 \cdot 10^6 \text{ cycle time}_0$$

This is not a good design choice — the new CPU will complete the instructions in $3675 \cdot 10^6$ cycles, but in terms of the first CPU which has 10% faster cycle time, it would take $4042.5 \cdot 10^6$ cycles to do what the first CPU does in $3800 \cdot 10^6$ cycles, making the second CPU a poor design choice

b)

double performance for arithmetic $=$ CPI_{ar} ↓ by $1/2$ $= CPI_{ar} \cdot 1/2 = 1 \cdot 1/2 = 1/2$

10x ↓ $= CPI_{ar}$ ↓ by $1/10$ $= CPI_{ar} \cdot 1/10 = 1 \cdot 1/10 = 1/10$

$$ET_2 = \left[(500 \cdot 10^6 \cdot 1/2) + (300 \cdot 10^6 \cdot 10) + (100 \cdot 10^6 \cdot 3) \right] \cdot \text{clocktime}_2$$

$$= \left[250 \cdot 10^6 + 3000 \cdot 10^6 + 300 \cdot 10^6 \right] \cdot \text{clocktime}_2$$

$$= 3550 \cdot 10^6 \cdot \text{clocktime}_2$$

$$\frac{ET_0}{ET_2} = \frac{3800 \cdot 10^6 \cdot \text{clocktime}_0}{3550 \cdot 10^6 \cdot \text{clocktime}_2} = 1.07 \Rightarrow$$ new CPU is 1.07x faster than the original machine

$$ET_3 = \left[(.1 \cdot 500 \cdot 10^6) + (300 \cdot 10^6 \cdot 10) + (100 \cdot 10^6 \cdot 3) \right] \cdot \text{clocktime}_3$$

$$= \left(50 \cdot 10^6 + 3000 \cdot 10^6 + 300 \cdot 10^6 \right] \cdot \text{clocktime}_3$$

$$= 3350 \cdot 10^6 \cdot \text{clocktime}_3$$

$$\frac{ET_0}{ET_3} = \frac{3800 \cdot 10^6 \cdot \text{clocktime}_0}{3350 \cdot 10^6 \cdot \text{clocktime}_3} = 1.13 \Rightarrow$$ new CPU is 1.13x faster than the original CPU

machine? What if we find a way to improve the performance of arithmetic instructions by 10 times?

2.40 Assume that for a given program 70% of the executed instructions are arithmetic, 10% are load/store, and 20% are branch.

2.40.1 [5] <§2.21> Given the instruction mix and the assumption that an arithmetic instruction requires 2 cycles, a load/store instruction takes 6 cycles, and a branch instruction takes 3 cycles, find the average CPI.

2.40.2 [5] <§§1.6, 2.13> For a 25% improvement in performance, how many cycles, on average, may an arithmetic instruction take if load/store and branch instructions are not improved at all?

2.40.3 [5] <§§1.6, 2.13> For a 50% improvement in performance, how many cycles, on average, may an arithmetic instruction take if load/store and branch instructions are not improved at all?

$P_{ar} = 70\%$ $P_{load} = 10\%$ $P_{branch} = 20\%$

a) $cycles_{ar} = 2$ $cycles_{load} = 6$ $cycles_{branch} = 3$

$CPI = .7(2) + .1(6) + .2(3)$

$= 1.4 + .6 + .6$

$CPI = 2.6$

b) If only arithmetic changed after 25% improvement $\Rightarrow$ CPI $\downarrow$ .25 $\Rightarrow$ CPI $\cdot .75$

$.7(cycles_{ar}) + .1(6) + .2(3) = 2.6 \cdot .75$

$.7 cycles_{ar} + 1.2 = 1.95$

$.7 cycles_{ar} = .75$

$cycles_{ar} = 1.07$ 1.07 cycles

c) If only arithmetic changed after 25% improvement $\Rightarrow$ CPI $\downarrow$ .50 $\Rightarrow$ CPI $\cdot .50$

$.7(cycles_{ar}) + .1(6) + .2(3) = 2.6 \cdot .50$

$.7 cycles_{ar} + 1.2 = 1.3$

$.7 cycles_{ar} = 0.1$

$cycles_{ar} = 0.14$ 0.14 cycles