Computer Architecture and Programming Languages
Jacobs University Bremen
Neeha Hammad

Course: CO20-320241
11th November, 2019

Homework 8

Problem 8.1

Solution:

(a) 25/32 = 0.78125
$0.78125 \times 2 = 1.5625 \rightarrow 1$
$0.5625 \times 2 = 1.125 \rightarrow 1$
$0.125 \times 2 = 0.25 \rightarrow 0$
$0.25 \times 2 = 0.5 \rightarrow 0$
$0.5 \times 2 = 1.0 \rightarrow 1$

This gives us: 0.11001 which can be written as 1.1001×2^{-1}

Now, exponent is -1 so $127 - 1 = 126_{10}$.

$126_{10} = 01111110_2$

sign	exp	fraction
0	01111110	10010000000000000000000

(b) $27 \div 2 = 13 \rightarrow 1$
$13 \div 2 = 6 \rightarrow 1$
$6 \div 2 = 3 \rightarrow 0$
$3 \div 2 = 1 \rightarrow 1$
$1 \div 2 = 0 \rightarrow 1$

This gives us 11011 which can also be written as 1.1011×2^4

Now, exponent is 4 so +127=131_{10}

$131 = 10000011_2$

$0.3515625 \times 2 = 0.703125 \rightarrow 0$
$0.703125 \times 2 = 1.40625 \rightarrow 1$
$0.40625 \times 2 = 0.8125 \rightarrow 0$
$0.8125 \times 2 = 1.625 \rightarrow 1$
$0.625 \times 2 = 1.25 \rightarrow 1$
$0.25 \times 2 = 0.5 \rightarrow 0$
$0.5 \times 2 = 1.0 \rightarrow 1$

This gives us: 0101101

sign	exp	fraction
0	10000011	01011010000000000000000

Problem 8.2

Solution:

(a) True

(b) False

(c) False

(d) False

(e) False

Problem 8.3
Solution:

add, \$t3, \$s0,\$s5

Problem 8.4
Solution:

(a) Since this is a J type instruction, 26 bits can be used.

(b) Reference: Upsala University Course Pages

In such a case, jr can also be used. We can temporarily store the address in a register and then just jr with that register.

By definition, jr: \$register jumps to the address in \$register most common use

Alternatively, jal (jump and link)can be used to jump anywhere. When using jal, what happens is that the initial 4 digits of the PC are concatenated along with the 26 bit address. This is left-shifted by 2 positions.

By definition, jal : Copies the address of the next instruction into the register \$ra(register 31) and then jumps to the address label.

Problem 8.5
Solution:

Class	CPI on P1	CPI on P2	Freq	(CPI on P1×Freq)/100	(CPI on P2×Freq)/100
A	1	2	60%	0.6	1.2
B	2	2	10%	0.2	0.2
C	3	2	10%	0.3	0.2
D	4	4	10%	0.4	0.4
E	3	4	10%	0.3	0.4
Sum				1.8	2.4

Total Instructions for P1: 13
Total Instructions for P2: 14

Clock Rate for P1: 4
Clock Rate for P2: 6

CPU time = (Instruction count$\times CPI)/(ClockRate)$

CPU time P1= (13 $\times 1.8)/4$

$= 5.85$

CPU time P2=(14$\times 2.4)/6$

$= 5.6$

Since time for P2 is lesser we know that P2 will finish rendering the image first.

P2 is faster than P1 with the following ratio: (5.85)/(5.6)

=1.045 (almost 4%)

Problem 8.6

Solution:

CPU time = (Instruction count$\times CPI)/(ClockRate)$

CPU Time (P1): (6 $\times(7/3))/2$
$= 7$

$CPUTime(P2) : (6 \times (5/2))/4$
$= 15/4$

Performance (P1): 1/7 = 0.149
Performance (P2): 4/15 = 0.2567

P2 has a better performance so it is faster.

P2 is faster by P1 by the following ratio: 0.267/0.1429 = 1.866

P2 is faster than P1 by this ratio. We can say that it is almost 86% faster.