ECE341 Homework No. 5 Solution

Problem No. 1

Using the same field locations for all instructions enables the processor to read the source registers of an instruction, while it is still decoding the instruction OP code. Performing the "instruction decode" and "register read" operations in parallel speeds up the stage-2 of instruction processing in the 5-stage processor hardware.

Problem No. 2

The processor executes the instruction OR R2, R2, R6

Before the instruction: R2 = 0x24B4, R6 = 0xA742, PC = 0x24A014

Register value changes in each cycle are as follows:

Cycle-1: PC = 0x24A014 + 4 = 0x24A018

Cycle-2: RA = Contents of R2 = 0x24B4

Cycle-3: RZ = (0x24B4) OR (0xA742) = 0xA7F6
RM = Contents of R6 = 0xA742

Cycle-4: RY = 0xA7F6

Cycle-5: R2 = 0xA7F6

Cycle-6: PC = 0x24A018 + 4 = 0x24A01C

Problem No. 3

(a) There are two main changes needed in the datapath of Figure 5.8 to implement the "Movz" instruction:

i. Modifications to MuxY: Since the "Movz" instruction requires the contents of "Rsrc2" to be transferred to "Rdst", the multiplexer "MuxY" needs to have another data input, which provides the contents of "Rsrc2". This data input is connected to the output of register "RM", because "RM" contains the "Rsrc2" contents read into "RB" during stage-2 of instruction processing.

ii. Propagation of comparison result: After the ALU compares the contents of "Rsrc1" with zero, the result of this comparison, which is a single bit (either "0" or "1") needs to be propagated to subsequent stages. This is accomplished by adding two 1-bit registers "RC1" and "RC2" after stage-3 and stage-4, respectively. "RC1" receives its input from the ALU and passes it to "RC2" during the next cycle. The contents of "RC2" are examined to decide if a writeback is performed during stage-5 or not.

(b)

Step	Action
1	Memory Address ← [PC], Read memory, IR ← Memory Data, PC ← [PC] + 4
2	Decode instruction, RA ← [RSrc1], RB ← [Rsrc2]
3	Compare [RA] to zero. If [RA] == 0, then RC1 ← 1, else RC1 ← 0
4	RC2 ← [RC1], RY ← [RM]
5	If [RC2] == 0, then Rdst ← [RY]

Problem No. 4

(a) Since it is assumed that the access to memory is always completed in one clock cycle, each instruction will take 5 cycles to complete on the 5-stage RISC hardware. Therefore, CPI = 5
We know that:

$$\text{Execution Time} = \frac{\text{Number of Instructions} * \text{CPI}}{\text{Clock Rate}}$$

Therefore:

$$\text{Instruction Execution Rate} = \frac{\text{Number of Instructions}}{\text{Execution Time}} = \frac{\text{Clock Rate}}{\text{CPI}} = \frac{2.5\ \text{GHz}}{5} = \frac{2.5 * 10^9}{5}$$

$= 5 * 10^8 =$ 500 million instructions per second

(b) Let us look at each stage individually:
Stage-1: 95% of instruction fetches take one cycle, 5% of fetches take 10 cycles.
Therefore CPI for stage-1 = (0.95)(1) + (0.05)(10) = 1.45
Stage-2: Each instruction decode takes one clock cycle.
Therefore CPI for stage-2 = 1
Stage-3: Each instruction execute takes one clock cycle.
Therefore CPI for stage-3 = 1
Stage-4: Load/store instructions take 2 clock cycles, all the other instructions take one cycle.
Therefore CPI for stage-4 = (0.25 + 0.15)(2) + (0.15 + 0.45)(1) = 1.4
Stage-5: Each writeback takes one clock cycle.
Therefore CPI for stage-5 = 1
Since all the instructions are executed sequentially (one at a time):
Total CPI = Sum of stage CPIs = 1.45 + 1 + 1 + 1.4 + 1 = 5.85

$$\text{Instruction Execution Rate} = \frac{\text{Clock Rate}}{\text{CPI}} = \frac{2.5\ \text{GHz}}{5.85} = \frac{2.5 * 10^9}{5.85}$$

$= 4.274 * 10^8 =$ 427.4 million instructions per second

Problem No. 5

(a)

	B-select in stage-3	*Y-select* in stage-4	*RF_write* in stage-5	*C_select* in stage-5
LOAD	1	01	1	00
AND	0	00	1	01
CALL_REGISTER	x	10	1	10

(b)

i. For instruction 1:
Contents of *RZ* = 200 + [R2] = 200 + 2000 = **2200**
Also, contents of R4 become 26 after instruction 1.

For instruction 2:
Contents of *RZ* = [R3] AND [R4] = 60 AND 26 = $(111100)_2$ AND $(011010)_2$ = $(011000)_2$ = **24**
Contents of R5 become 24 after this instruction.

ii. For instruction 1:
Contents of *RY* = Contents of memory address 2200 = **26**

For instruction 2:
Contents of *RY* = Result computed in stage-3 = **24**

For instruction 3:
Contents of *RY* = Return address = Address of next sequential instruction = 40 + 4 + 4 + 4 = **52**

iii. Instruction 3 (Call_Register) will write the current contents of register R5 to the PC. Since R5 becomes 24 after instruction 2:
Final contents of PC after the instruction sequence = **24**