

Assignment for Laboratory 5
HW Instruction Set Architecture
Computer Science 240

You will be spending the next lab implementing and experimenting with a data and control path for the HW ISA (small instruction set architecture you have begun learning about in lecture). Review the notes from lecture, and answer the following questions.

1. How many instructions are there in the HW instruction set?

8 are listed.

2. How many bits are there in each instruction?

16

3. What assembly language instruction is represented by the hexadecimal value 0x0021? (each digit represents 4 bits). Describe what you expect the instruction to do.

LW R0 R2 1

Load the word at address (contents of R0 + offset 1 = address 1) in data memory into register R2.

4. What is the 16-bit binary form of the following instruction?

ADD R1 R1 R4

0010 0001 0001 0100

What are the contents of Register 1 and Register 4 after this instruction is executed?

R1 = 1, R4 = 2

5. Given the following instruction stored at address 8 in memory:

8: BEQ R5 R6 C

Assume register 5 contains FFFE, and register 6 contains FFFE and that the offset is interpreted as a signed, 4-bit, two's complement values.

After this instruction is executed, what will be the address of the next instruction?

Since the contents of R5 and R6 are equal, the branch will be taken to $PC = PC + 1 + (2 * \text{offset})$.

$PC = 8$, offset = -4, so $PC = 8 + 2 + 2 * -4 + 2$.

6. Repeat question 5, but assume that the original value of register 5 = 0003, and register 6 = 0002. What will be the address of the next instruction?

Since the contents of the registers are not equal, the PC will simply increment by 2.

$PC = 8 + 2 = 10$

7. Fill in the table for the following program:

| Address | Instruction | Operation | Rs | Rt | Rd/offset | Purpose |
|---------|-------------|-----------|----|----|-----------|--|
| 0: | 5002 | OR | R0 | R0 | R2 | initialize R2 = 0 |
| 2: | 5003 | OR | R0 | R0 | R3 | initialize R3 = 0 |
| 4: | 1220 | SW | R2 | R2 | 0 | store word at specified addr to R2 (addr = contents of R2 + offset 0) |
| 6: | 0230 | LW | R2 | R3 | 0 | load R3 w/contents of specified addr (addr = contents of R2 + offset 0) |
| 8: | 2122 | ADD | R1 | R2 | R2 | increment R2 by 1 |
| A: | 8002 | JMP | 4 | | | jump to address - = 2*offset |

Describe the result (specific values of modified registers and address locations) after allowing 18 instructions to execute:

There are 2 instructions at the beginning which are not repeated, and then the final 4 instructions repeat, or loop. So, after 18 instructions, the loop repeats 4 times.

Each time through the loop, the value in R2 is stored to an address in memory also specified by R2, the same value is then loaded into R3 from that address, and then R2 is incremented and the steps are repeated. Since R2 is 0 initially, the addresses 0,1,2,... are accessed each time through the loop, and the corresponding value is stored into the respective address. R3 is also updated each pass through the loop.

After 18 instructions, the following addresses and registers will have been modified and hold the specified values:

$M[0] = 0, M[1] = 1, M[2] = 2, M[3] = 3, R3 = 3, R4 = 2$

Does the program ever stop? No, it is an infinite loop.

