BIRZEIT UNIVERSITY

Electrical and Computer Engineering

ENCS2380 – Computer Organization and Microprocessors- Fall 2022

Course Project (1)

Student Name:	Student ID NO:
Saif Al-Deen Battah.	1170986
Abdulkhaliq Injas.	1173251

Instructor: Dr. Abualseoud Hanani.

Section NO: 1.

Date: 24/1/2023.

Contents

Q1

A) Since the machine has a 3-bit opcode field → so machine support a maximum 8 instructions.

B) Since the machine has 8-bit "Src. Register. Memory Address" field:

1. Signed → -127 to +127.
2. Unsigned → 0 to 255.

C) PC → 8 bits.
IR → 16 bits.
MAR → 8 bits.
MBR → 16 bits.

Q2

Convert to machine code:

Memory	Address	Contents
10	000 000 00 00010100	Load R0, [20] (instruction)
11	000 001 00 00010101	Load R1,21
12	010 000 01 00000001	Add R0, [R1] (instruction)
13	000 001 01 00010110	load R1, [22] (instruction)
14	011 001 00 00001000	Sub R1, +8 (instruction)
15	010 000 01 00010001	Add R0,R1 (instruction)
16	001 000 00 00010111	Store R0, [23] (instruction)
20	0000000000000110	6 (data)
21	0000000000000100	4 (data)
22	0000000000001101	13 (data)
23	0000000000000000	0 (data)
24	0000000000000000	0 (data)

Attached file name "Verilog1.v" with the code.

Q3

A)

Assembly	Machine Code
Load R0, [30]	000 000 00 00011110
Load R1, [31]	000 001 00 00011111
Load R2, [32]	000 010 00 00100000
MUL R1, R2	0100 001 01 00100000
ADD R0, R1	010 000 00 00011111
SUB R0, 1	011 000 00 00000001
Load R1, [33]	000 001 00 00100001
Load R2, [34]	000 010 00 00100010
SUB R1, R2	011 001 10 00100010
DIV R0, R1	0101 000 01 00100001
Store R0, [35]	001 000 00 00100011

This assembly code performs the following operations:

1) Load the contents of memory cell address 30 into register R0 using the LOAD instruction with opcode 000, destination register R0, addressing mode 00, and memory address 30 (00011110 in binary)

2) Load the contents of memory cell address 31 into register R1 using the LOAD instruction with opcode 000, destination register R1, addressing mode 00, and memory address 31 (00011111 in binary)

3) Load the contents of memory cell address 32 into register R2 using the LOAD instruction with opcode 000, destination register R2, addressing mode 00, and memory address 32 (00100000 in binary)

4) Multiply the contents of register R1 and R2 using the MUL instruction with opcode 0100, destination register R1, addressing mode 01, and source register R2 (00100000 in binary)

5) Add the contents of register R0 and R1 using the ADD instruction with opcode 010, destination register R0, addressing mode 00, and source register R1 (00011111 in binary)

6) Subtract 1 from the contents of register R0 using the SUB instruction with opcode 011, destination register R0, addressing mode 00, and constant 1 (00000001 in binary)

7) Load the contents of memory cell address 33 into register R1 using the LOAD instruction with opcode 000, destination register R1, addressing mode 00, and memory address 33 (00100001 in binary)

8) Load the contents of memory cell address 34 into register R2 using the LOAD instruction with opcode 000, destination register R2, addressing mode 00, and memory address 34 (00100010 in binary)

9) Subtract the contents of register R2 from the contents of register R1 using the SUB instruction with opcode 011, destination register R1, addressing mode 10, and source register R2 (00100010 in binary)

10) Divide the contents of register R0 by the contents of register R1 using the DIV instruction with opcode 0101, destination register R0, addressing mode 01, and source register R1 (00100001 in binary)

11) Store the contents of register R0 in memory cell address 35 using the STORE instruction with opcode 001, destination register R0, addressing mode 00, and memory address 35 (00100011 in binary)

The above assembly code follows the mathematical order of operations and it calculates ((A + B * C - 1) / (D-E)) and stores the result in memory cell address 35.

B)

Address	Content
10	000 000 00 00011110
11	000 001 00 00011111
12	000 010 00 00100000
13	0100 001 01 00100000
14	010 000 00 00011111
15	011 000 00 00000001
16	000 001 00 00100001
17	000 010 00 00100010
18	011 001 10 00100010
19	0101 000 01 00100001
20	001 000 00 00100011

C)

Attached file name "Verilog2.v" contains the code.