Session 4

Student Name _____

Other Identification _____

Machine Language and Its Relationship to C++

In this laboratory session you will:

1. Investigate the machine language described in Appendix C of *Computer Science: An Overview*.
2. Investigate the relationship between a machine language and C++.

Your instructor will tell you which of the proposed experiments you are to perform.

In preparation for this laboratory session, you should read Chapter Two of *Computer Science: An Overview*.

The Machine Simulator

In this laboratory you will experiment with machine language concepts using the program (CP04E01) that simulates the activities of the machine described in Appendix C of *Computer Science: An Overview*. You will find a discussion of that machine in Chapter Two of that text. For your convenience, a summary of that machine language is given at the end of this laboratory.

Your instructor will tell you how to execute the simulation program on your particular system.

The simulation program (CP04E01) begins by displaying the state of the simulated machine, which consists of a rectangular presentation of all the memory cells of the machine followed by a display of the registers 0 through F, the program counter (identified on the screen as PC), and the instruction register (identified as IR) as shown below. (Note that the memory display is arranged so that the contents of the memory cell at location 5E is the value found in row 5, column E.)

```
                        Main Memory

      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
3  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
A  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
B  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
E  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
F  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

R0:00 R1:00 R2:00 R3:00 R4:00 R5:00 R6:00 R7:00  PC:00
R8:00 R9:00 RA:00 RB:00 RC:00 RD:00 RE:00 RF:00  IR:0000
```

Below this display on the screen will appear a list of the command options available, represented by the letters M, R, P, C, S, G, F, H, and Q. By typing one of these characters followed by the ENTER key you may select among the following activities:

Option	Activity
M	Change the contents of memory cells.
R	Change the contents of registers.
P	Change the contents of the program counter.
C	Clear all the memory cells or all the registers.
S	Execute a single machine cycle.
G	Execute up to 500 machine cycles or until a halt instruction is executed.
F	Obtain a list of file options for saving and retrieving programs.
H	Display a description of the available options.
Q	Terminate the simulation program.

In each case, directions will appear on the screen that will lead you through the desired activity.

The following experiment will acquaint you with the simulation program.

Experiment 4.1

Familiarize yourself with the simulation program (CP04E01) by following the steps below to enter and execute the following machine language program.

Address	Content
10	23
11	FF
12	12
13	10
14	50
15	23
16	30
17	40
18	C0
19	00

Step 1. Start the simulator and select the option M. Starting at address 10, enter the program above byte by byte. Return to the main display by entering an empty byte (i.e., by typing the ENTER key, not preceded by other symbols).

Step 2. Select the option P, and set the program counter to the value 10.

Step 3. Inspect the main display to confirm that the program has been properly placed in main memory starting at location 10 and that the program counter is set at 10. Then, select the option S to execute a single machine cycle.

What is the value of the program counter at this point? _____

What is the value in register 3 at this point? _____

Step 4. Select the option G to continue executing the program until a halt instruction is reached.

What is the value of the program counter when the simulation halts? _____

What is the value in memory cell 40 when the simulation halts? _____

Step 5. Translate the program in this experiment into English and confirm that the simulator performed the appropriate actions.

_____

_____

_____

_____

_____

Step 6. Change the contents of the memory cell at address 14 to the hexadecimal value 60. Then, execute this modified program by changing the program counter back to 10 and selecting option G from the main display. Explain the difference in the actions of this program from those of the original program.

Machine-Level Instructions

In the following experiments, you will experiment with some of the instructions found in a traditional machine language.

Experiment 4.2

In this experiment you will investigate the various arithmetic and logic operations available on the machine simulated by the program CP04E01.

Step 1. Clear the simulator's main memory and registers. Then, place the value 5C in register 1 and the value 3F in register 2. Finally, place the values 50 and 12 at memory locations A0 and A1, put the value A0 in the program counter, and execute a single machine cycle. Record the value found in register 0 and explain this result.

Step 2. One at a time, place the values 60, 70, 80, and 90 at memory location A0, reset the program counter at A0, and execute a single machine cycle. In each case record the value found in register R0 and explain the result.

Experiment 4.3

In this experiment you will investigate an application of masking.

Step 1. Execute the simulation program (CP04E01) and clear the simulated machine's main memory. Place the program below in the memory cells from address 00 to 1D. Then, place the values 6F, 87, D0, 5E, 29, and 90 in the cells from address A0 to A5.

Address	Contents	Address	Contents
00,01	20,80	10,11	54,34
02,03	11,A0	12,13	34,03
04,05	81,10	14,15	34,0B
06,07	B1,0C	16,17	20,A5
08,09	22,00	18,19	B4,1C
0A,0B	32,A0	1A,1B	B0,00
0C,0D	23,01	1C,1D	C0,00
0E,0F	14,0B		

Step 2. Run the program and record the changes in the values stored from address A0 to A5. Assuming that the patterns represent values stored in floating-point notation, state a general rule for determining which patterns are replaced by zeros. (Look at the sign bit of the original patterns.)

Step 3. Alter the program so that it replaces only the negative values with zeros and test your program. Record the changes you made below.

Experiment 4.4

In this experiment you will investigate the branch instruction provided in the machine language for the machine simulated by the program Simulator.

Step 1. Execute the simulation program (CP04E01) and clear the simulated machine's main memory. Then, place the program below in the memory cells from address F0 to FD.

Address	Contents	Address	Contents
F0	20	F7	FC
F1	00	F8	50
F2	21	F9	01
F3	01	FA	B0
F4	23	FB	F6
F5	05	FC	C0
F6	B3	FD	00

Step 2. Execute the program one step at a time by using the single step option (S) from the main display. Record the values of the program counter after each step.

Step 3. Explain the results obtained in Step 2.

Step 4. What changes should be made to the program in Step 1 if it were to be placed in memory starting at location A0? Explain your answer.

Self-Modifying Programs

In the following experiments you will investigate a self-modifying program—that is, a program that makes changes to itself.

Experiment 4.5

This experiment investigates the concept of self-modifying programs—that is, programs that modify themselves.

Step 1. Execute the simulation program (CP04E01) and clear the simulated machine's main memory. Then, place the program below in it.

Address	Contents	Address	Contents
00	23	04	B0
01	40	05	06
02	33	06	C0
03	05	07	00

Step 2. Execute the machine language program just entered and record the results. (In particular, what is the value of the program counter when execution stops?)

Step 3. Translate the machine language program in Step 1 into English and explain how the program counter obtained the value observed in Step 2.

Step 4. Clear the simulated machine's memory and place the following program in it.

Address	Contents	
00,01	2100	Initialize
02,03	2270	counters.
04,05	3109	Set origin
06,07	320B	and destination.
08,09	1000	Now move
0A,0B	3000	one byte.
0C,0D	2001	Increment
0E,0F	5101	addresses.
10,11	5202	
12,13	2333	Do it again
14,15	4010	if more
16,17	B31A	bytes to be
18,19	B004	moved.
1A,1B	2070	Adjust values
1C,1D	3071	that are
1E,1F	2079	location
20,21	3075	dependent.
22,23	207B	
24,25	3077	
26,27	208A	
28,29	3087	
2A,2B	2074	
2C,2D	3089	
2E,2F	20C0	Make sure
30,31	30A4	program
32,33	2000	will halt.
34,35	20A5	
36,37	B070	Make the big jump!

Step 5. Execute this machine language program for 500 instructions and describe the changes that have appeared in the machine's main memory.

_____

_____

_____

_____

Step 6. Continue executing the program for 500 instructions at a time and observing the changes in memory and the program counter until the program finally terminates. What did the program do?

_____

_____

_____

_____

Study the original program and explain the role of the instructions in locations 12 through 19.

Explain the role of the instructions in locations 1A through 2D.

Explain the role of the instructions in locations 2E through 33.

Machine Language Summary

The following table summarizes the machine language presented in Appendix C and Chapter Two of *Computer Science: An Overview*. Each instruction is 16 bits long and thus represented by four hexadecimal digits.

Op-code	Operand	Description
1	RXY	LOAD register R with the contents of the memory cell at address XY.
2	RXY	LOAD register R with the value XY.
3	RXY	STORE the contents of register R at memory location XY.
4	0RS	MOVE the contents of register R to register S.
5	RST	ADD the contents of registers S and T as though they were binary numbers and leave the result in register R.
6	RST	ADD the contents of registers S and T as though they represented values in floating-point notation and leave the result in register R.
7	RST	OR the contents of registers S and T and place the result in register R.
8	RST	AND the contents of registers S and T and place the result in register R.
9	RST	EXCLUSIVE OR the contents of registers S and T and place the result in register R.
A	R0X	ROTATE the contents of register R one bit to the right X times.
B	RXY	JUMP to the instruction located at memory address XY if the contents of register R equals that of register 0.
C	000	HALT

Post-Laboratory Problems

4.1. a. Translate the following C++ program into the machine language used in this laboratory session. Then execute your program using the computer simulator to confirm that it performs as expected.

```
void main(void)
{
  int value_a,
      value_b;

  value_b = -8;
  value_a = value_b + 5;
}
```

b. What changes should be made to the machine language program you wrote in Step 1 if the lines

```
int value_a,
    value_b;
```

in the C++ program were changed to

```
float value_a,
      value_b;
```

c. What complications would arise when translating the C++ program in Step 1 into our machine language if the lines

```
int value_a,
    value_b;
```

were changed to

```
float value_a;
int   value_b;
```

4.2. Translate the following C++ program into the machine language used in this laboratory session. Then execute your program using the computer simulator to confirm that it performs as expected.

```
void main(void)
{
  int Count;

 Count = 0;
 while (Count < 4)
  {
     Count++;
   }
}
```

4.3. Based on what you learned in Experiment 4.5, write a machine language program that initially starts at address 00 but repeatedly copies itself to a location 70 addresses ahead in memory and then jumps to the copy so that the program slowly crawls around and around in memory.

4.4. Design a machine language program that computes the sum of two numbers that require up to 16 bits for their binary representations. Assume that one number is stored in registers 3 and 4, while the other number is stored in registers 5 and 6. Place the sum in registers 1 and 2. For example, to add the values 6AD8 and 7254

the contents of registers 3, 4, 5, and 6 would be 6A, D8, 72, and 54, respectively, and the sum (which is DB2C) would be found in registers 1 and 2 with register 1 containing DB and register 2 containing 2C. Remember that the sum of the least significant bytes can affect the sum of the most significant bytes.

4.5. Place the patterns C3, 03, 00, 01, and D1 in the memory cells from address C0 to C4 and place the patterns 02 and C1 in the cells at address D1 and D2, respectively. Then, execute the following program.

Address	Contents	Address	Contents
70,71	20,00	7E,7F	31,73
72,73	11,C0	80,81	32,00
74,75	B1,8A	82,83	12,81
76,77	31,79	84,85	52,23
78,79	12,00	86,87	32,81
7A,7B	23,01	88,89	B0,72
7C,7D	51,13	8A,8B	C0,00

What does the program do? Explain the steps of the program.

4.6. Translate the machine language program appearing in Experiment 4.4 into the C++ language.

4.7. Write a machine language program that places the hexadecimal numbers 02, 04, 06, 08, 0A, and 0C in six consecutive memory cells starting at address A1. (See Experiment 4.3 for ideas.)

4.8. a. Write a machine language program that adds the six floating-point values stored in the memory cells from address D0 to address D5 and places the sum at address D6.

b. Revise the program in part a to add seven values starting at address D0 and store the sum at address D7.

c. What changes were required in part b? Design a generic program to add the contents of arbitrarily long sequences of memory cells.

d. Translate your answer to part c into a C++ program.