ICS Answer Sheet #10

Sakar Gopal Gurubacharya
s.gurubacharya@jacobs-university.de

Problem 10.1: assembly programming

a. **Convert the machine code given in hexadecimal notation into binary notation.**
b. **Write down the assembly code for the machine code. Add meaningful descriptions.**

#	Hex	Binary	Assembly code	Description
0	2e	001 0 1110	LOAD 14	Load the value of memory location 14 into the accumulator
1	b0	101 1 0000	EQUAL #0	Skip next instruction if accumulator equals to 0
2	d4	110 1 0100	JUMP #4	Jump to instruction 4 (set program counter to 4)
3	e0	111 0 0000	HALT 0	Stop execution
4	2f	001 0 1111	LOAD 15	Load the value of memory location 15 into the accumulator
5	6f	011 0 1111	ADD 15	Add the value of memory location 15 to the accumulator
6	4f	010 0 1111	STORE 15	Store the value of the accumulator in memory location 15
7	2e	001 0 1110	LOAD 14	Load the value of memory location 14 into the accumulator
8	91	100 1 0001	SUB #1	Subtract the value of 1from the accumulator
9	4e	010 0 1110	STORE 14	Store the value of the accumulator in memory location 14
10	cb	110 0 1011	JUMP 11	Jump to instruction 11 (set program counter to 11)
11	00	000 0 0000	DATA 0	Contain value of 0, indirect jump to instruction 0
12	00	000 0 0000	DATA 0	Contain value of 0, no use in program
13	00	000 0 0000	DATA 0	Contain value of 0, no use in program
14	06	000 0 0110	DATA 6	Contain value of 6, decreases down to 0
15	01	000 0 0001	DATA 1	Contain value of 1, exponentially increases to 64

c. **The program leaves a result in memory cell 15 when it halts. What is the value? Explain how the program works, either in words or by providing an equivalent program code in a higher level imperative language.**

ACC = Accumulator
PC = Program Counter
MEM [14] = Memory cell 14
MEM [15] = Memory cell 15

```
1st iteration:
ACC     :  6      1   2   6   5
PC      : 14   4  15  15  14  14  11  0
MEM[14] :  6                   5
MEM[15] :  1          2

2nd iteration:
ACC     :  5      2   4   5   4
PC      : 14   4  15  15  14  14  11  0
MEM[14] :  5                   4
MEM[15] :  2          4

3rd iteration:
ACC     :  4      4   8   4   3
PC      : 14   4  15  15  14  14  11  0
MEM[14] :  4                   3
MEM[15] :  4          8

4th iteration:
ACC     :  3      8  16   3   2
PC      : 14   4  15  15  14  14  11  0
MEM[14] :  3                   2
MEM[15] :  8         16
```

```
5th iteration:
ACC     :  2     16  32   2   1
PC      : 14   4  15  15  14  14  11  0
MEM[14] :  2                   1
MEM[15] : 16         32
6th iteration:
ACC     :  1     32  64   1   0
PC      : 14   4  15  15  14  14  11  0
MEM[14] :  1                   0
MEM[15] : 32         64

7th iteration:
ACC     :  0  HALT
PC      : 14     4  15  15  14  14  11  0
MEM[14] :  0
MEM[15] : 64

Final solution:
ACC     :  0
PC      : 14
MEM[14] :  0
MEM[15] : 64
```

At the start of each iteration, the accumulator takes whatever value is at memory cell 14. It checks this value with 0 in each iteration and the program continues if it doesn't match.

Then, the accumulator takes whatever value is at memory cell 15 and then doubles itself. At the end of each iteration the value at memory cell 15 is doubled. Each iteration follows an exponential growth with 2^n, where n is the current iteration number, n <= 6. **6 because the starting value at memory cell 14 is 6.**

Then it rereads the value at memory cell 14 and now subtracts it by 1 to have a finite loop. It is decreasing after each iteration and satisfies the EQUAL condition after 6 iterations.

The program counter changes based on the instruction given at that point.

When the program halts, the value at memory cell 15 is **64 (2^6).**

d. What happens if the value stored in memory cell 14 is changed to 10 before execution starts? Explain.

As highlighted in bold in *10.1c*, the value at memory cell 14 determines the value for 'n' in the conjecture made by observing the behavior after each iteration.

If the value stored in memory cell 14 is changed to 10 before the execution starts, it would take 10 iterations and in the 11th iteration, the program would HALT the final value at memory cell 15 would be $\mathbf{2^{10} = 1024}$

Problem 10.2: type classes (haskell)

Attempted but couldn’t do it, any hints?