Jona Bako
jbako@jacobs.university.de

Homework - ICS 2020 Problem Sheet #10

Problem 10.1

The following program has been written for the simple central processing unit introduced in class. The table below shows the initial content of the 16 memory cells. The first column denotes the memory address and the second column shows the memory content in hexadecimal notation.

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 3-1-4) Machine Code	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 equal to value 0
2	d4	110 1 0100	JUMP #4	Jump to instruction 4 (set program counter to 4)
3	e0	111 0 0000	HALT	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 value 1 from 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 operation defined in memory address 11
11	00	000 0 0000	Memory	No instruction/data, initialized to 0
12	00	000 0 0000	Memory	No instruction/data, initialized to 0
13	00	000 0 0000	Memory	No instruction/data, initialized to 0
14	06	000 0 0110	Memory	No instruction/data, initialized to 6
15	01	000 0 0001	Memory	No instruction/data, initialized to 1

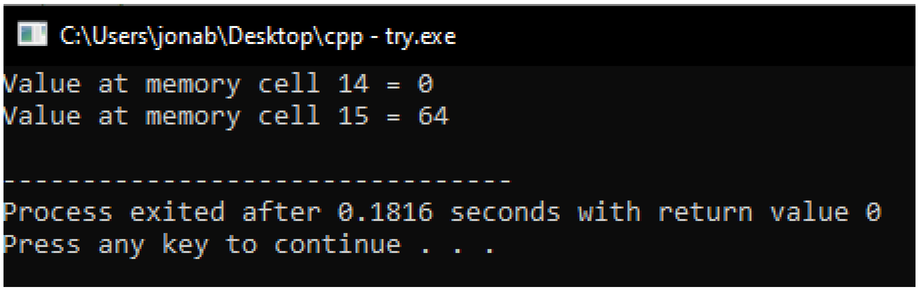
c) The program leaves a result in memory cell 15 when it halts. What is the value? Explain how the program works, either in words or by providing an equivalent program code in a higher level imperative language.

- First the value of memory location 14 (6) is loaded into the accumulator. Then since the evaluation is false, the next step is not skipped, but we jump to instruction 4. Now the value of memory location 15 (1) is loaded into the accumulator, and since according to the next instruction its value (1) is added again, now the the accumulator equals to value 2. Then, the value is stored into memory location 15. The value of memory location 14 (6), is loaded into the accumulator, and then 1 is substracted, leaving 5 to the value of accumulator, then storing this value again in memory location 14. We now jump to operation defined in memory address 11, initialized at 0, meaning we go back to start and repeat until program halts (continuousely substract 1 from value at memory cell 14, until accumulator equals to value 0, and execution stops).

Number of re-excecution of program	Value at memory cell 14 (Substract 1)	Value at memory cell 15 (Add itself)
0	6	1
1	5	1+1 = 2
2	4	2 + 2 = 4
3	3	4 + 4 = 8
4	2	8 + 8 = 16
5	1	16 + 16 = 32
6	0	Skip to memory cell 3 – HALT

- Therefore, final value stored in memory cell 15 after program halts is 32.

```
#include <iostream>
using namespace std;
int main(){
        int a = 6;
        int b = 1;
        while(1){
                if(a == 0)
                        break;
                a = a – 1;
                b = b + b;
        }
        cout<<"Value at memory cell 14 = "<<a<<endl;
        cout<<"Value at memory cell 15 = "<<b<<endl;
        return 0;
}
```



C:\Users\jonab\Desktop\cpp - try.exe

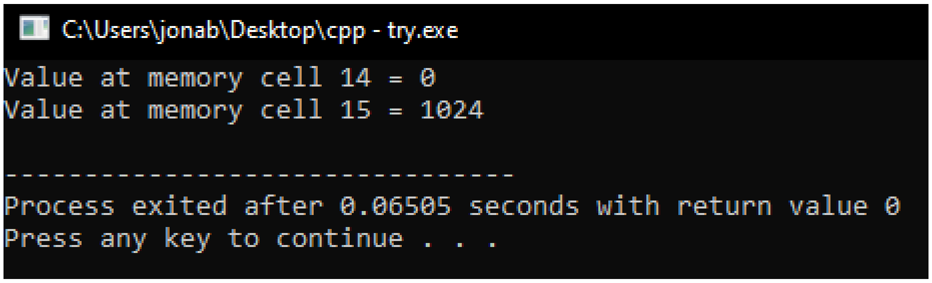
```
Value at memory cell 14 = 0
Value at memory cell 15 = 64

--------------------------------
Process exited after 0.1816 seconds with return value 0
Press any key to continue . . .
```

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

- Same steps are repeated, but this time it will take longer for the value at memory cell 14 to reach 0, and the program to halt, therefore value at memory location 15 will now be 512.

Number of re-excecution of program	Value at memory cell 14 (Substract 1)	Value at memory cell 15 (Add itself)
0	10	1
1	9	1+1 = 2
2	8	2 + 2 = 4
3	7	4 + 4 = 8
4	6	8 + 8 = 16
5	5	16 + 16 = 32
6	4	32 + 32 = 64
7	3	64 + 64 = 128
8	2	128 + 128 = 256
9	1	256 + 256 = 512
10	0	Skip to memory cell 3 – HALT

```
#include <iostream>
using namespace std;
int main(){
    int a = 10;
    int b = 1;
    while(1){
            if(a == 0)
                    break;
            a = a – 1;
            b = b + b;
    }
    cout<<"Value at memory cell 14 = "<<a<<endl;
    cout<<"Value at memory cell 15 = "<<b<<endl;
    return 0;
}
```



```
Value at memory cell 14 = 0
Value at memory cell 15 = 1024

--------------------------------
Process exited after 0.06505 seconds with return value 0
Press any key to continue . . .
```