CH-232-A

Answers to ICS 2020 Problem Sheet #10

Blen Daniel Assefa
bassefa@jacobs-university.de

1. A) and B)

#	Hex	Binary	Assembly Code	Description
0	2e	00101110	LOAD 14	Load the value of memory location 14 into the accumulator
1	b0	10110000	EQUAL #0	Skip next instruction if accumulator equal to 0
2	d4	11010100	JUMP #4	Jump to instruction 4 (set program counter to 4)
3	e0	11100000	HALT	Stop execution
4	2f	00101111	LOAD 15	Load the value of memory location 15 into the accumulator
5	6f	01101111	ADD 15	Add the value of memory location 15 into the accumulator
6	4f	01001111	STORE 15	Store the value of the accumulator in memory location 15
7	2e	00101110	LOAD 14	Load the value of memory location 14 into the accumulator
8	91	10010001	SUB #1	Subtract the value 1 from the accumulator
9	4e	01001110	STORE 14	Store the value of the accumulator in memory location 14
10	cb	11001011	JUMP 11	Jump to memory location 11 (Set the value of PC to the memory location number)
11	00	00000000	DATA #0	Data 0
12	00	00000000	DATA #0	Data 0
13	00	00000000	DATA #0	Data 0
14	06	00000110	DATA #6	Data 6
15	01	00000001	DATA #1	Data 1

C)

MEM = 0x2e b0 d4 e0 2f 6f 4f 2e 91 4e cb 00 00 00 06 01

1:	**PC= 0**	**IR= 0x2e**	**ACC= 0x06**	**LOAD 14**
2:	PC= 1	IR= 0xb0	ACC= 0x00	EQUAL #0
3:	PC= 4	IR= 0xd4	ACC= 0x00	JUMP #4
4:	PC= 5	IR= 0x2f	ACC= 0x01	LOAD 15
5:	PC= 6	IR= 0x6f	ACC= 0x02	ADD 15
6:	PC= 7	IR= 0x4f	ACC= 0x00	STORE 15
7:	PC= 8	IR= 0x2e	ACC= 0x06	LOAD 14
8:	PC= 9	IR= 0x91	ACC= 0x05	SUB #1
9:	PC= 10	IR= 0x4e	ACC= 0x00	STORE 14
10:	PC= 0	IR=0xcb	ACC= 0x00	JUMP 11
11:	PC= 1	IR= 0x2e	ACC= 0x05	LOAD 14
12:	PC= 2	IR=0xb0	ACC= 0x00	EQUAL #0
13:	PC= 4	IR= 0xd4	ACC= 0x00	JUMP #4
14:	PC= 5	IR= 0x2f	ACC= 0x02	LOAD 15
15:	PC= 6	IR= 0x6f	ACC= 0x04	ADD 15
16:	PC= 7	IR= 0x4f	ACC= 0x00	STORE 15
17:	PC= 8	IR= 0x2e	ACC= 0x05	LOAD 14
18:	PC= 9	IR= 0x91	ACC= 0x04	SUB #1
19:	PC= 10	IR= 0x4e	ACC= 0x00	STORE 14
20:	PC= 0	IR=0xcb	ACC= 0x00	JUMP 11
21:	PC= 1	IR= 0x2e	ACC= 0x04	LOAD 14
22:	PC= 2	IR=0xb0	ACC= 0x00	EQUAL #0
23:	PC= 4	IR= 0xd4	ACC= 0x00	JUMP #4
24:	PC= 5	IR= 0x2f	ACC= 0x04	LOAD 15
25:	PC= 6	IR= 0x6f	ACC= 0x08	ADD 15
26:	PC= 7	IR= 0x4f	ACC= 0x00	STORE 15
27:	PC= 8	IR= 0x2e	ACC= 0x04	LOAD 14
28:	PC= 9	IR= 0x91	ACC= 0x03	SUB #1
29:	PC= 10	IR= 0x4e	ACC= 0x00	STORE 14
30:	PC= 0	IR=0xcb	ACC= 0x00	JUMP 11
31:	PC= 1	IR= 0x2e	ACC= 0x03	LOAD 14
32:	PC= 2	IR=0xb0	ACC= 0x00	EQUAL #0
33:	PC= 4	IR= 0xd4	ACC= 0x00	JUMP #4
34:	PC= 5	IR= 0x2f	ACC= 0x08	LOAD 15
35:	PC= 6	IR= 0x6f	ACC= 0x016	ADD 15
36:	PC= 7	IR= 0x4f	ACC= 0x00	STORE 15
37:	PC= 8	IR= 0x2e	ACC= 0x03	LOAD 14
38:	PC= 9	IR= 0x91	ACC= 0x02	SUB #1
39:	PC= 10	IR= 0x4e	ACC= 0x00	STORE 14
40:	PC= 0	IR=0xcb	ACC= 0x00	JUMP 11
41	PC= 1	IR= 0x2e	ACC= 0x02	LOAD 14

42:	PC= 2	IR=0xb0	ACC= 0x00	EQUAL #0
43:	PC= 4	IR= 0xd4	ACC= 0x00	JUMP #4
44:	PC= 5	IR= 0x2f	ACC= 0x016	LOAD 15
45:	PC= 6	IR= 0x6f	ACC= 0x032	ADD 15
46:	PC= 7	IR= 0x4f	ACC= 0x00	STORE 15
47:	PC= 8	IR= 0x2e	ACC= 0x02	LOAD 14
48:	PC= 9	IR= 0x91	ACC= 0x01	SUB #1
49:	PC= 10	IR= 0x4e	ACC= 0x00	STORE 14
50:	PC= 0	IR=0xcb	ACC= 0x00	JUMP 11
51:	PC= 1	IR= 0x2e	ACC= 0x01	LOAD 14
52:	PC= 2	IR=0xb0	ACC= 0x00	EQUAL #0
53:	PC= 4	IR= 0xd4	ACC= 0x00	JUMP #4
54:	PC= 5	IR= 0x2f	ACC= 0x032	LOAD 15
55:	PC= 6	IR= 0x6f	ACC= 0x064	ADD 15
56:	PC= 7	IR= 0x4f	ACC= 0x00	STORE 15
57:	PC= 8	IR= 0x2e	ACC= 0x01	LOAD 14
58:	PC= 9	IR= 0x91	ACC= 0x00	SUB #1
59:	PC= 10	IR= 0x4e	ACC= 0x00	STORE 14
60:	PC= 0	IR=0xcb	ACC= 0x00	JUMP 11
61:	PC= 1	IR= 0x2e	ACC= 0x00	LOAD 14
62:	PC= 2	IR=0xb0	ACC= 0x00	EQUAL #0
63:	PC= 3	IR= 0xe0	ACC= 0x00	HALT

MEM = 0x2e b0 d4 e0 2f 6f 4f 2e 91 4e cb 00 00 00 06 0x064

Basically, at memory location 15 it doubles and at memory location 14 it subtracts 1 from the already stored value in memory location 14.

D) The program continues for another 4 loops until memory location goes to 1.