Quinn Roemer

Professor Chris Grove

CSC 137

15 April 2020

Homework #3

6.1:

The program listed in the book is stored in the memory unit of the basic computer. Show the contents of AC, PC, and IR (in hexadecimal) at the end, after each instruction is executed

Location & Instruction	AC	PC	IR
010 CLA	0	011	7800
011 ADD 016	C1A5	012	1016
012 BUN 014	C1A5	014	4014
014 AND 017	8184	015	0017
015 BUN 013	8184	013	4013
013 HLT	8184	014	7001

NOTE: Program is listed in execution order, not actual order

6.2:

The program listed in the book is a list of instructions in hexadecimal code. The computer executes the instructions starting from address 100. What are the contents of AC and the memory word at address 103 when the computer halts?

- The contents of AC will be **FFFE**:
    - This is due to a CLA, INC, and CMA command on the register
- The contents of M[103] will be **101:**
    - This is due to a BSA 103 I instruction at location 106

6.7:

Obtain the address symbol table generated for the program of Table 6-13 during the first pass of the assembler.

ADDRESS SYMBOL	HEXADECIMAL ADDRESS
LOP	105
ADS	10B
PTR	10C
NBR	10D
CTR	10E
SUM	10F

List the translated program in hexadecimal.

Instruction	Location	Hex
LDA ADS	100	210B
STA PTR	101	310C
LDA NBR	102	210D
STA CTR	103	310E
CLA	104	7800
LOP, ADD PTR I	105	910C
ISZ PTR	106	610C
ISZ CTR	107	610E
BUN LOP	108	4105
STA SUM	109	310F
HLT	10A	7001
ADS, HEX 150	10B	0150
PTR, HEX 0	10C	0000
NBR, DEC -100	10D	FF9C
CTR, HEX 0	10E	0000
SUM HEX 0	10F	0000
ORG 150	-	-
DEC 75	150	004B
.	-	-
.	-	-
.	-	-
DEC 23	1B3	0017
END	-	-

6.21:

Write a subroutine to subtract two numbers. In the calling program, the BSA instruction is followed by the subtrahend and minuend. The difference is returned to the main program in the third location following the BSA.

Calling Program

```
BSA SUB      /Branch and save return location to subroutine SUB
HEX 17FA     /Subtrahend (to be subtracted)
HEX 2CB5     /Minuend
Hex 0        /Difference
```

Subroutine

```
SUB,    HEX 0           /Subroutine start and PC storage

        LDA SUB I       /Load into AC indirectly the value at SUB

        CMA             /Complement AC

        INC             /Increment AC

        ISZ SUB         /Increment skip if zero on SUB

        ADD SUB I       /Add the value indirectly at SUB

        ISZ SUB         /Increment skip if zero on SUB

        STA SUB         /Store the value in AC at memory location SUB

        ISZ SUB         /Increment skip if zero on SUB

        BUN SUB I       /Branch unconditionally indirectly to SUB
```

Practice:

INSTRUCTION	AC
SUB, HEX 0	-
LDA SUB I	17FA
CMA	E805
INC	E806
ISZ SUB	E806
ADD SUB I	14BB
ISZ SUB	14BB
STA SUB	14BB
ISZ SUB	14BB
BUN SUB I	14BB

#5:

Translate the listed program statements into assembly code

```
        ORG 100         /Origin of program at 100

        LDA Y           /Load Y into AC

        CMA             /Complement AC

        INC             /Increment AC

        ADD X           /Add X

        SZA             /Skip if AC zero (X == y)

        BUN AAA         /Branch unconditionally to AAA
        LDA X       /Load X into AC
        CLE         /Clear E
        CIL         /Circulate left (x * 2)
        CIL
        CIL
        STA X       /Store AC in X
        HLT         /Halt execution
AAA,    CLA         /Clear AC
LOP,    ADD X       /Add X to AC
        ISZ Y       /Increment Y and skip if zero
        BUN LOP     /Branch unconditionally to LOP
        STA SUM     /Store AC in SUM
        HLT         /Halt Execution
X,      DEC 4       /variable X
Y,      DEC -10     /variable Y
SUM,    DEC 0       /Storage location
        END         /End Program
```

Now translate the program into hex code

Instruction	*Location*	*Hex*
LDA Y	100	2114
CMA	101	7200
INC	102	7020
ADD X	103	1113
SZA	104	7004
BUN AAA	105	410D
LDA X	106	2113
CLE	107	7400
CIL	108	7040
CIL	109	7040
CIL	10A	7040
STA X	10B	3114

HLT	10C	7001
AAA, CLA	10D	7800
LOP, ADD X	10E	1113
ISZ Y	10F	6114
BUN LOP	110	410E
STA SUM	111	3115
HLT	112	7001
X, DEC 4	113	0004
Y, DEC -10	114	FFF6
SUM, DEC 0	115	0000