

=====
Part 1
=====

Sub	Acc	Dmu x	Rmu x	A	B	C	D
1	1	1	2	1	1	1	1

=====
Part 2
=====

1. Use the Register Transfer Notation discussed in the text to specify the following sequence of data transfers for the datapath you built in the previous part of this lab.

- 1) A -> Acc
- 2) add(B,Acc) -> Acc
- 3) sub(Acc,C) -> Acc
- 4) add(D,Acc) -> Acc

2. Translate each of the above RT actions into the appropriate control code for each action.

- 1) 011000000
- 2) 010010000
- 3) 110100000
- 4) 010110000

3. Initialize registers A,B,C,D to the values 0x13,0x5B,0x3A,0xF0 respectively. Translate the initial value of each of the registers into decimal.

- A. $1 \times 16 + 3 = 19$
- B. $5 \times 16 + 11 = 91$
- C. $3 \times 16 + 10 = 58$
- D. $15 \times 16 + 0 = -16$

4. Calculate by hand the values that should appear in the Acc register as you complete each of the above actions. Give your answer in both decimal and hex.

Hex	Dec
13	19
6e	110
34	52
24	36

5. Did an overflow condition happen in any of the steps of the calculation?

No.

6. What is final value in the Acc register in Decimal and in Hex?

0x24 (36base10)

7. Give a simple description of the calculation that produces the final result in the Acc register.

You feed the values from the D and the Acc registers ($52 + (-16)$) and store the result in the Acc register.

=====

Part 2

=====

Time	Register Values					Control Code					Action
	A	B	C	D	Acc	Sub	Acc	Dmux	Rmux	ABCD	
0	19	91	58	-16							Initial Reg Values
1	19	91	58	-16	19	0	1	1	00	0000	A -> Acc
2	19	91	58	-16	110	0	1	0	01	0000	add(B,Acc) -> Acc
3	19	91	58	-16	52	1	1	0	10	0000	sub(Acc,C) -> Acc
4	19	91	58	-16	110	0	1	0	11	0000	add(D,Acc) -> Acc