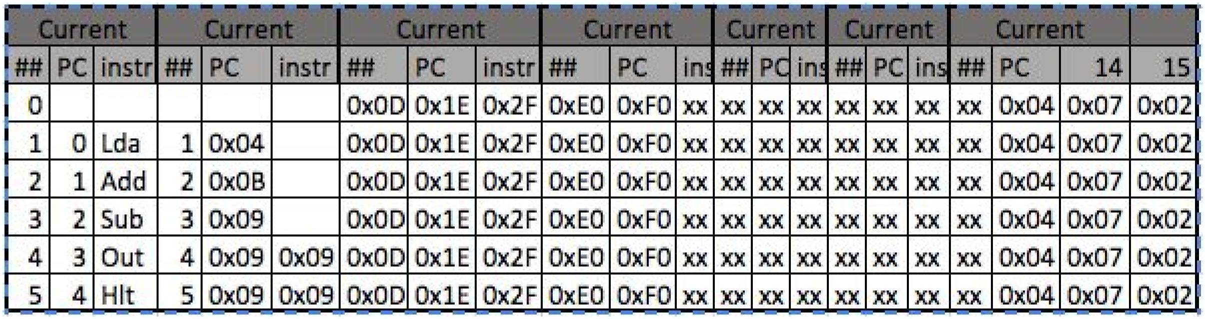
Chris Fenton
Hwk 8
Digital Logic
Collaborated with Ross and Simon

1. Mem, Acc, Out, PC
2. PC, MAR, Mem, ACC, IR, Out
3. 4
4. Instruction fetch cycle
5. It disables the clock
6. 15
7.

Instruction	Format	Meaning
Lda M	0000mmmmm	ACC<- c(M)
Add M	0001mmmmm	ACC <- (ACC + c(M)) mod 256
Sub M	0010mmmmm	ACC <- (ACC - c(M)) mod 256
Sta M	0011mmmmm	c(M) <- ACC
Jmp M	0100mmmmm	PC <- M
Jaz M	0101mmmmm	PC <- M if ACC == 0
Out	1110xxxxx	OUT <- ACC
Hlt	1111xxxxx	Halt execution

8. 

Current			Current			Current			Current			Current			Current			Current			
##	PC	instr	##	PC	instr	##	PC	instr	##	PC	ins	##	PC	ins	##	PC	ins	##	PC	14	15
0						0x0D	0x1E	0x2F	0xE0	0xF0	xx	xx	xx	xx	xx	xx	xx	xx	0x04	0x07	0x02
1	0	Lda	1	0x04		0x0D	0x1E	0x2F	0xE0	0xF0	xx	xx	xx	xx	xx	xx	xx	xx	0x04	0x07	0x02
2	1	Add	2	0x0B		0x0D	0x1E	0x2F	0xE0	0xF0	xx	xx	xx	xx	xx	xx	xx	xx	0x04	0x07	0x02
3	2	Sub	3	0x09		0x0D	0x1E	0x2F	0xE0	0xF0	xx	xx	xx	xx	xx	xx	xx	xx	0x04	0x07	0x02
4	3	Out	4	0x09	0x09	0x0D	0x1E	0x2F	0xE0	0xF0	xx	xx	xx	xx	xx	xx	xx	xx	0x04	0x07	0x02
5	4	Hlt	5	0x09	0x09	0x0D	0x1E	0x2F	0xE0	0xF0	xx	xx	xx	xx	xx	xx	xx	xx	0x04	0x07	0x02

9.

```
PC   = (t3 ^ jmp) v (t3 ^ jaz)
Inc  = t1
Amux = t3  ^ (Lda v Add v Sub v Sta)
Mar  = t1 v t3 ^ (Lda v Add v Sub v Sta)
Mst  = t4 ^ Sta
Ir   = t2
```

```
   Dmux = t4 ^ Lda
   Acc  = t4 ^ (Lda v Add v Sub)
   Out  = t3 ^ Out
   Sub  = t4 ^ Sub
   Hlt  = t3 ^ Hlt
```

10. Jaz = (Jaz ^ 0) ^ t3