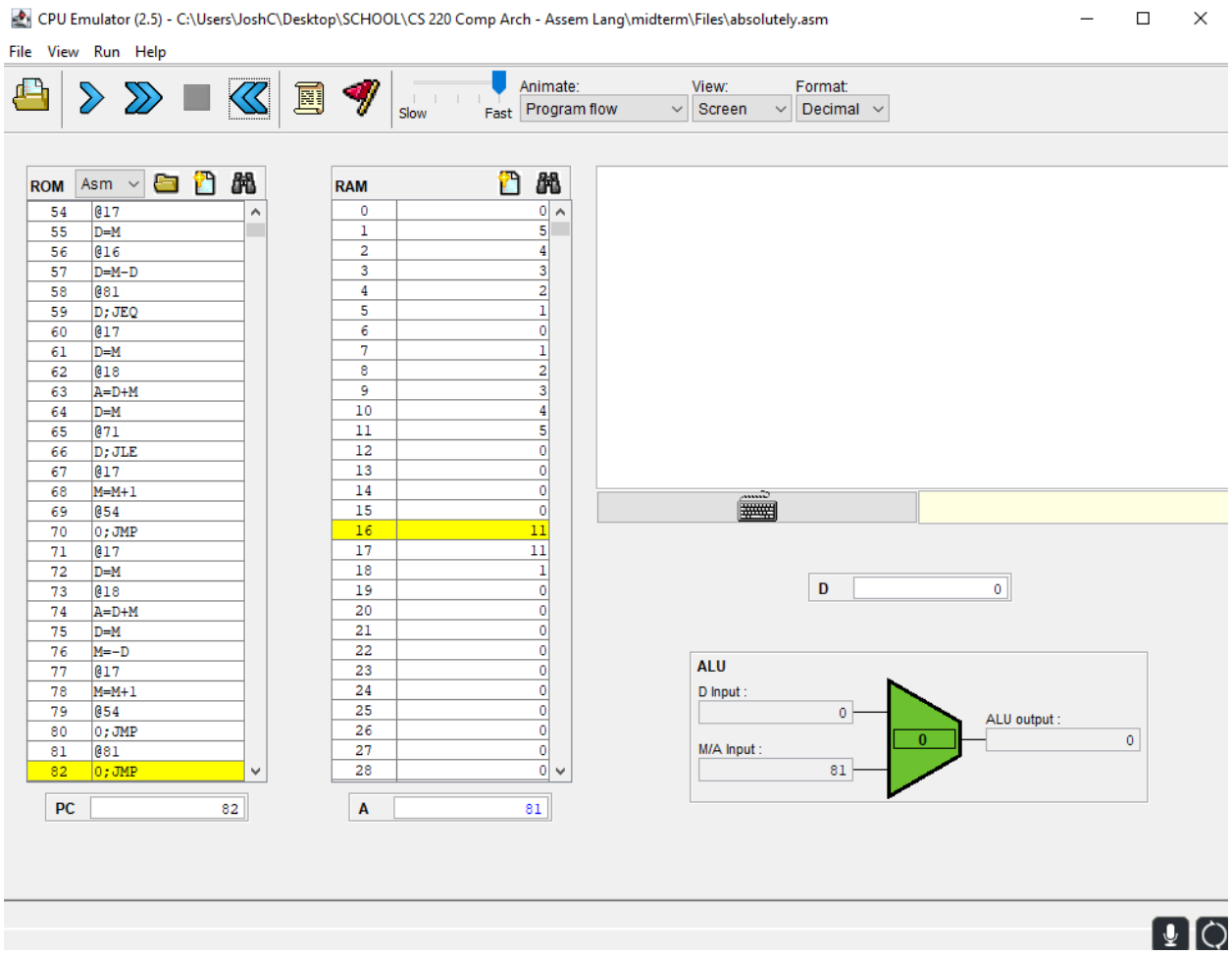



```
//initialize size = 11
@11
D=A
@size
M=D

//initialize i = 0
@i
M=0

//initialize arr
@1
D=A
@arr
M=D
```

```
//initialize array elements 1-11 [-5,-4,-3,-2,-1,0,1,2,3,4,5]
@5
D=A
@R1
M=-D

@4
D=A
@R2
M=-D

@3
D=A
@R3
M=-D

@2
D=A
@R4
M=-D

@1
D=A
@R5
M=-1

@0
D=A
@R6
M=0

@1
D=A
@R7
M=1

@2
D=A
@R8
M=D

@3
D=A
```

```
@R9
M=D

@4
D=A
@R10
M=D

@5
D=A
@R11
M=D

(LOOP)
   @i
   D=M
   @size
   D=M-D
   @END
   D;JEQ //if size - i = 0 jump to end

   @i
   D=M
   @arr
   A=M+D //go to address M+D
   D=M //store location

   @NEGATIVE
   D;JLE //if value is <= 0 jump to negative

   (POSITIVE)
         @i
         M=M+1 //increment i
         @LOOP
         0;JMP

   (NEGATIVE)
         @i
         D=M
         @arr
         A=M+D //go to address M+D
         D=M //store location
         M=-D //make value positive
```

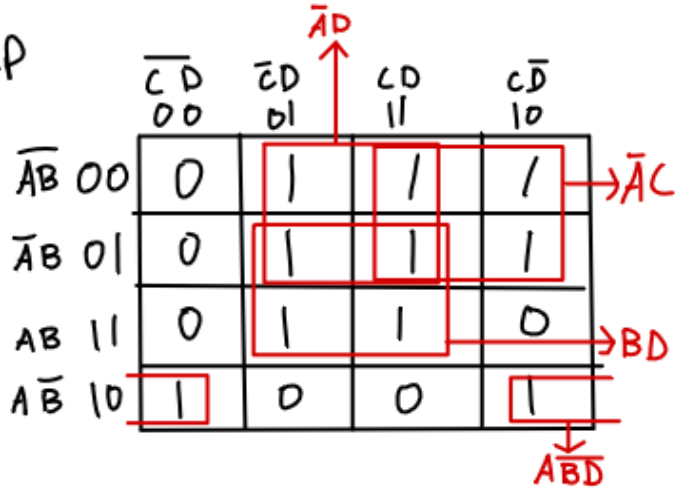
```
        @i
        M=M+1 //increment i
        @LOOP
        0;JMP

(END)
   @END
   0;JMP
```

Josh Clemens
4-1-23

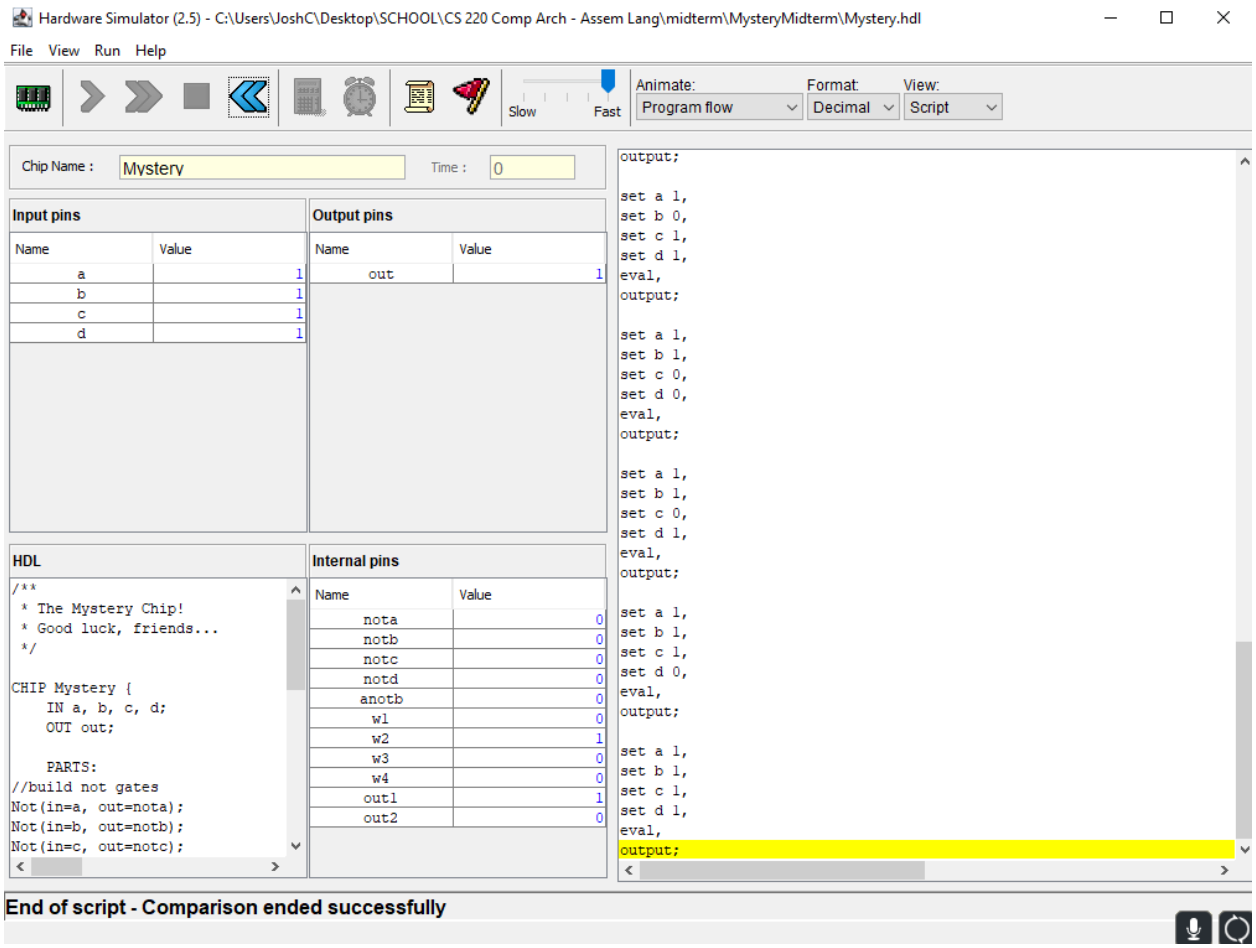
4 Variable Truth Table

A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

K-map



$F = A\bar{B}\bar{D} + BD + \bar{A}D + \bar{A}C$




```
/**
 * The Mystery Chip!
 * Good luck, friends...
 */

CHIP Mystery {
        IN a, b, c, d;
        OUT out;

        PARTS:
   //build not gates
   Not(in=a, out=nota);
   Not(in=b, out=notb);
   Not(in=c, out=notc);
   Not(in=d, out=notd);

   //first expression
   And(a=a, b=notb, out=anotb);
```

```
    And(a=anotb, b=notd, out=w1);

    //second
    And(a=b, b=d, out=w2);

    //third
    And(a=nota, b=d, out=w3);

    //fourth
    And(a=nota, b=c, out=w4);

    //combine with or gates
    Or(a=w1, b=w2, out=out1);
    Or(a=w3, b=w4, out=out2);
    Or(a=out1, b=out2, out=out);
}
```