Zach Dubinsky

CSC-270-01

2/1/22

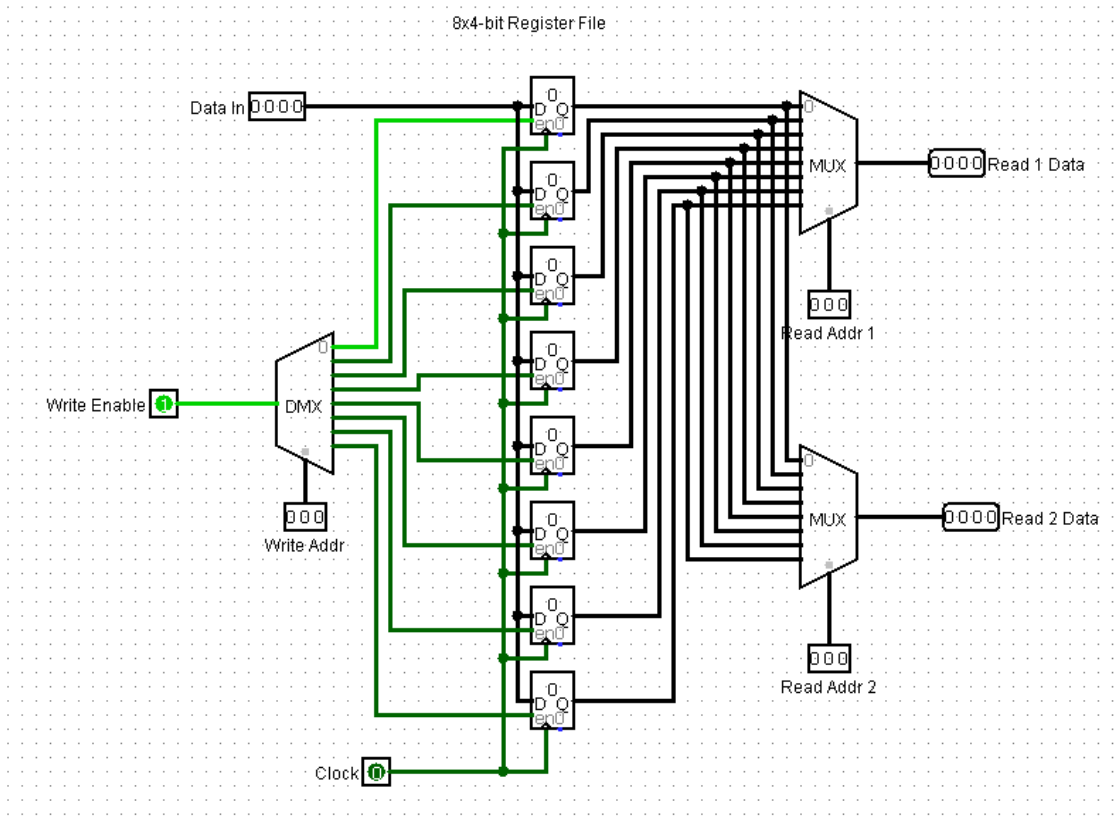
Prof.Rieffel

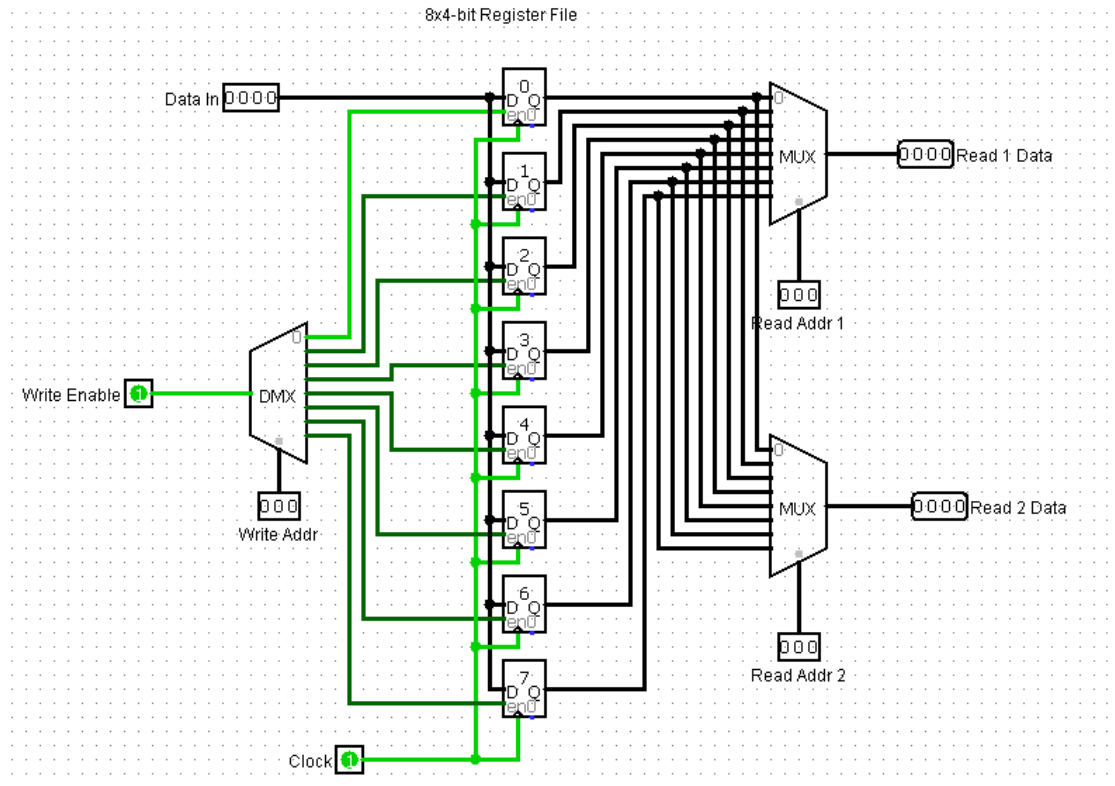
Lab 4

I affirm that all included work is my own in accordance with the class syllabus and the Union College Honor Code. [Signed: Zach Dubinsky, 2/1/22]

The following images show the correctness of the Pre-Lab4.hex program and the correctness of my circuit. I have included before and after screenshots, respectively, of my register. All forms of code for the program is included below. Throughout this report, $t0 - $t7 denotes registers zero through seven.

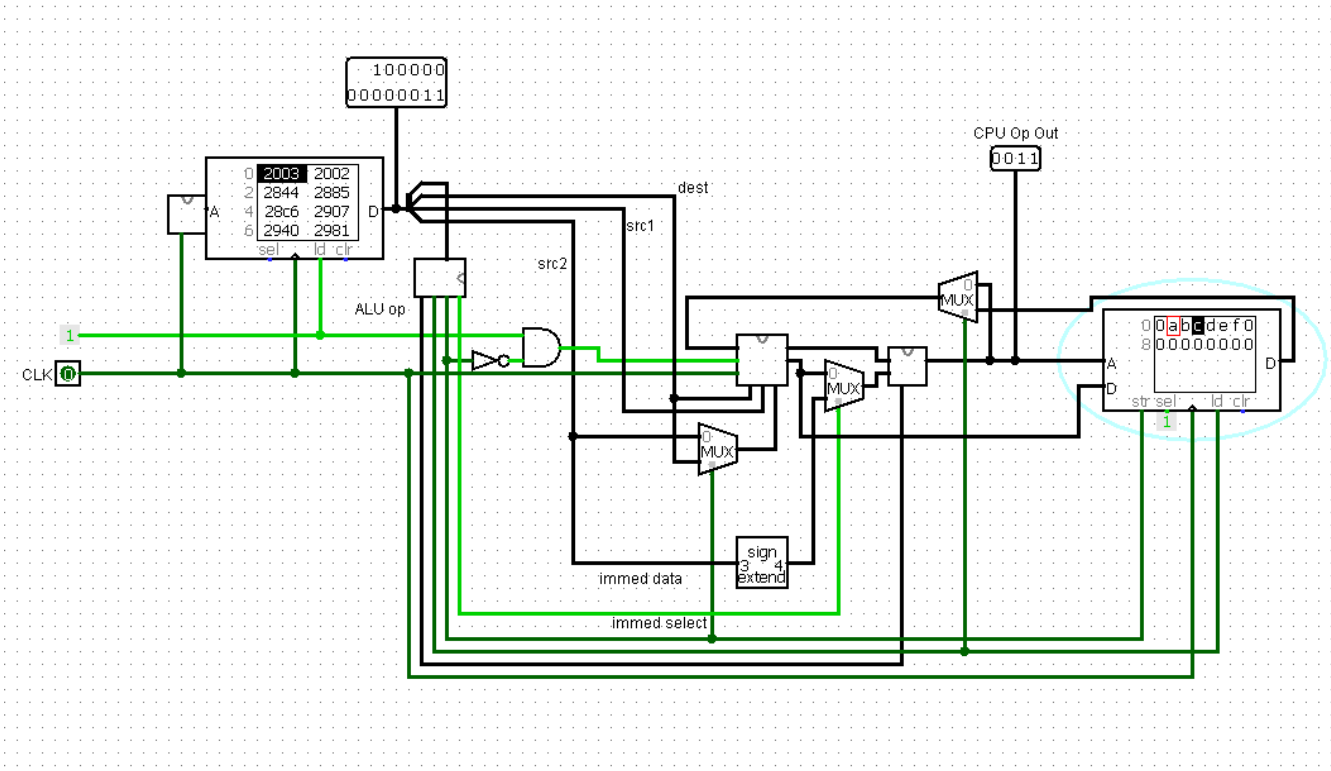
Assembly	Binary	Hexadecimal
addi $t1 $t0 1	10000001000001	2041
add $t2 $t1 $t1	00000010001001	0089
add $t3 $t2 $t1	00000011010001	00d1
add $t4 $t3 $t1	00000100011001	0119
add $t5 $t4 $t1	00000101100001	0161
add $t6 $t5 $t1	00000110101001	01a9
add $t7 $t6 $t1	00000111110001	01f1

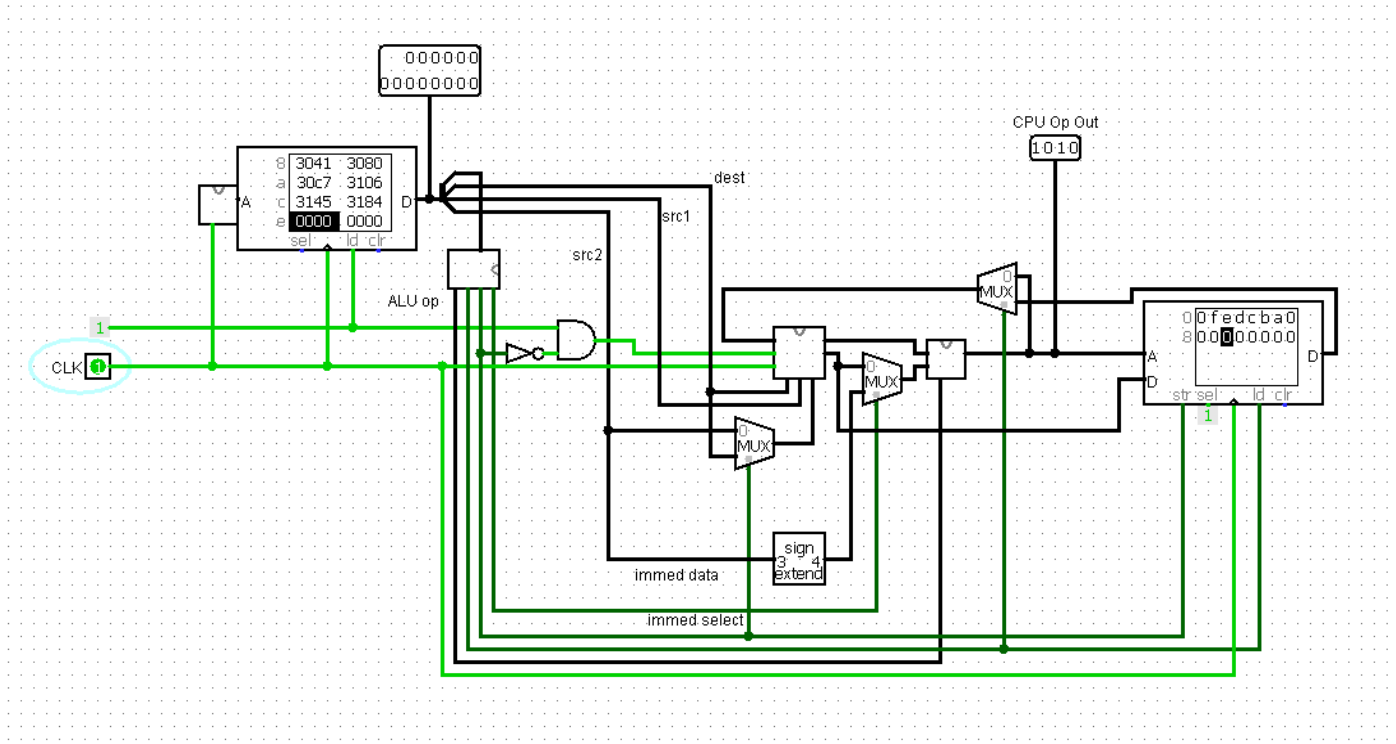

8x4-bit Register File
Data In
0000
Write Enable
DMX
000
Write Addr
Clock
MUX
0000
Read 1 Data
000
Read Addr 1
MUX
0000
Read 2 Data
000
Read Addr 2


8x4-bit Register File
Data In
0000
Write Enable
DMX
000
Write Addr
Clock
MUX
0000
Read 1 Data
000
Read Addr 1
MUX
0000
Read 2 Data
000
Read Addr 2

The following screenshots show the correctness of my Reverse.hex program. I have included before and after images, respectively, of my full circuit. Correctness is illustrated by the data memory module. All forms of code are included in the table below.

Assembly	Binary	Hexadecimal
addi $t0 $t0 3	10000000000011	2003
addi $t0 $t0 2	10000000000010	2002
lw $t1 $t0 -4	10100001000100	2844
lw $t2 $t0 -3	10100010000101	2885
lw $t3 $t0 -2	10100011000110	28c6
lw $t4 $t0 -1	10100100000111	2907
lw $t5 $t0 0	10100101000000	2940
lw $t6 $t0 1	10100110000001	2981
sw $t1 $t0 1	11000001000001	3041
sw $t2 $t0 0	11000010000000	3080
sw $t3 $t0 -1	11000011000111	30c7
sw $t4 $t0 -2	11000100000110	3106
sw $t5 $t0 -3	11000101000101	3145
sw $t6 $t0 -4	11000110000100	3184


100000
00000011
CPU Op Out
0011
0 2003 2002
2 2844 2885
4 28c6 2907
6 2940 2981
sel ld clr
A
D
dest
src1
src2
ALU op
1
CLK
MUX
sign
3 4
extend
immed data
immed select
0 0abcdef0
8 00000000
str sel ld clr


000000
00000000
CPU Op Out
1010
8 3041 3080
a 30c7 3106
c 3145 3184
e 0000 0000
sel ld clr
A
D
dest
src1
src2
ALU op
1
CLK
MUX
sign
3 4
extend
immed data
immed select
0 0fedcba0
8 00000000
str sel ld clr