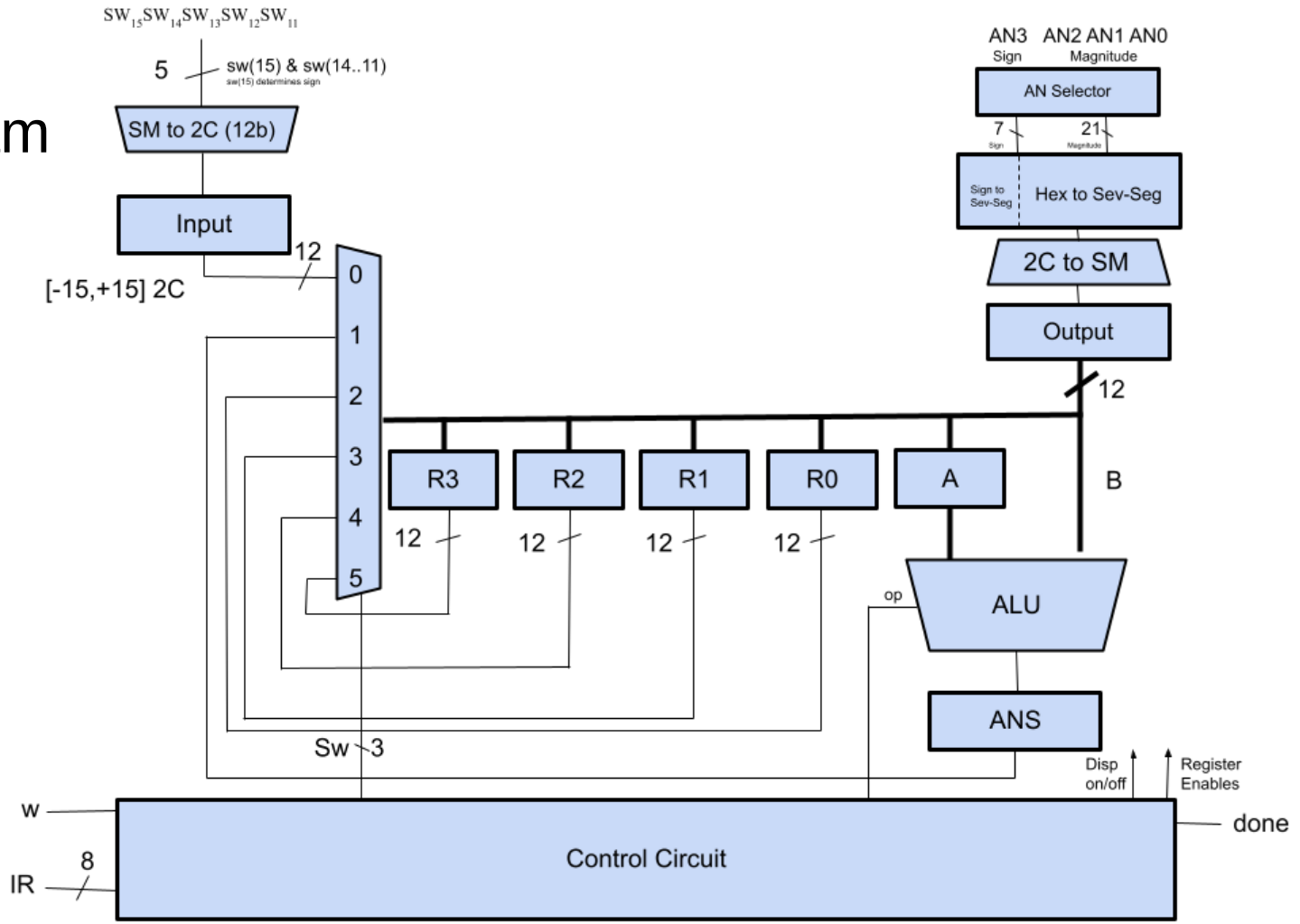
Small Microprocessor

BJ Blume, Erik Rosenkranz, Jason Rosenkranz

What our small microprocessor does

1. User inputs a number in Sign-and-Magnitude
2. User inputs the instruction code and w when ready
3. Operation is performed
4. Output is displayed on 7-segment display with sign in HEX

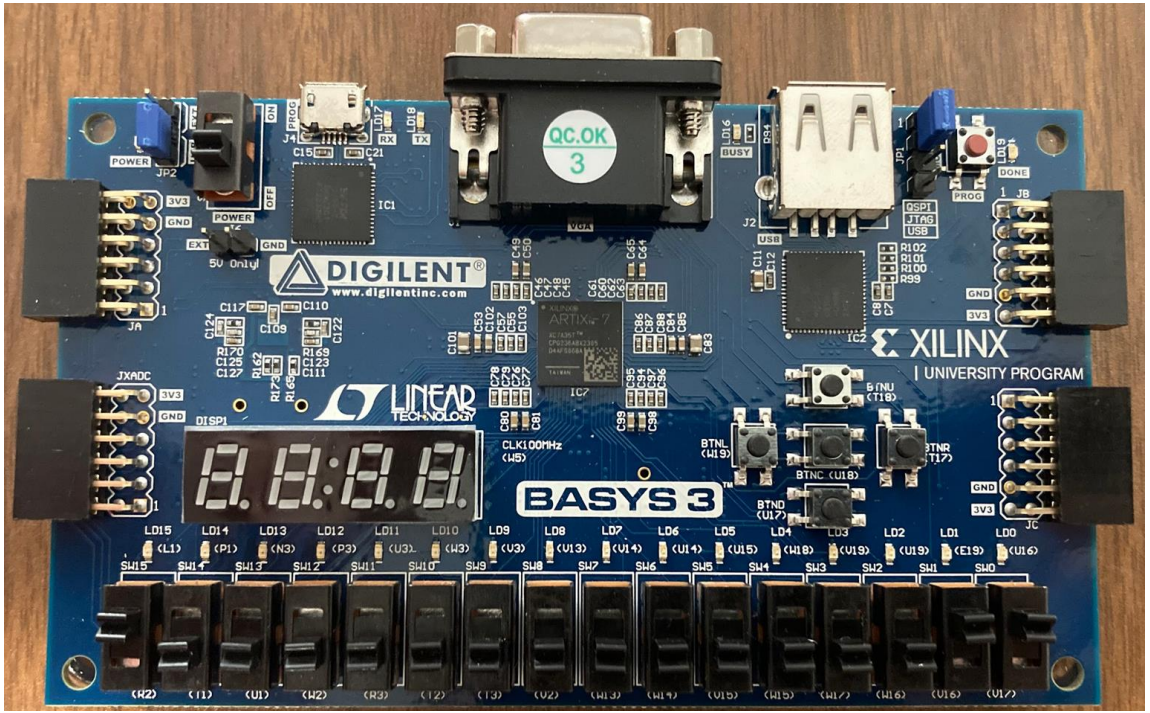
Block Diagram



Small Microprocessor Circuit Overview

- Data Formatting
- Input/Output Handling
- Design
- ALU
- Control Circuit (FSM)
- Instruction Options
- Example

Board Switches



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Sign	Input (3)	Input (2)	Input (1)	Input (0)		w	IR (7)	IR (6)	IR (5)	IR (4)	IR (3)	IR (2)	IR (1)	IR (0)	Resetn

Data Formatting

Input

- Four bit input
  - Set by four switches
- Fifth switch controls sign
  - 1 = negative, 0 = positive
- Two’s complement of input stored in the input register

Input Range: [-15, 15]

$\text{Binary (SM)}_{\text{4-bit + sign}} \rightarrow \text{Binary(2C)}_{\text{12-bit}}$

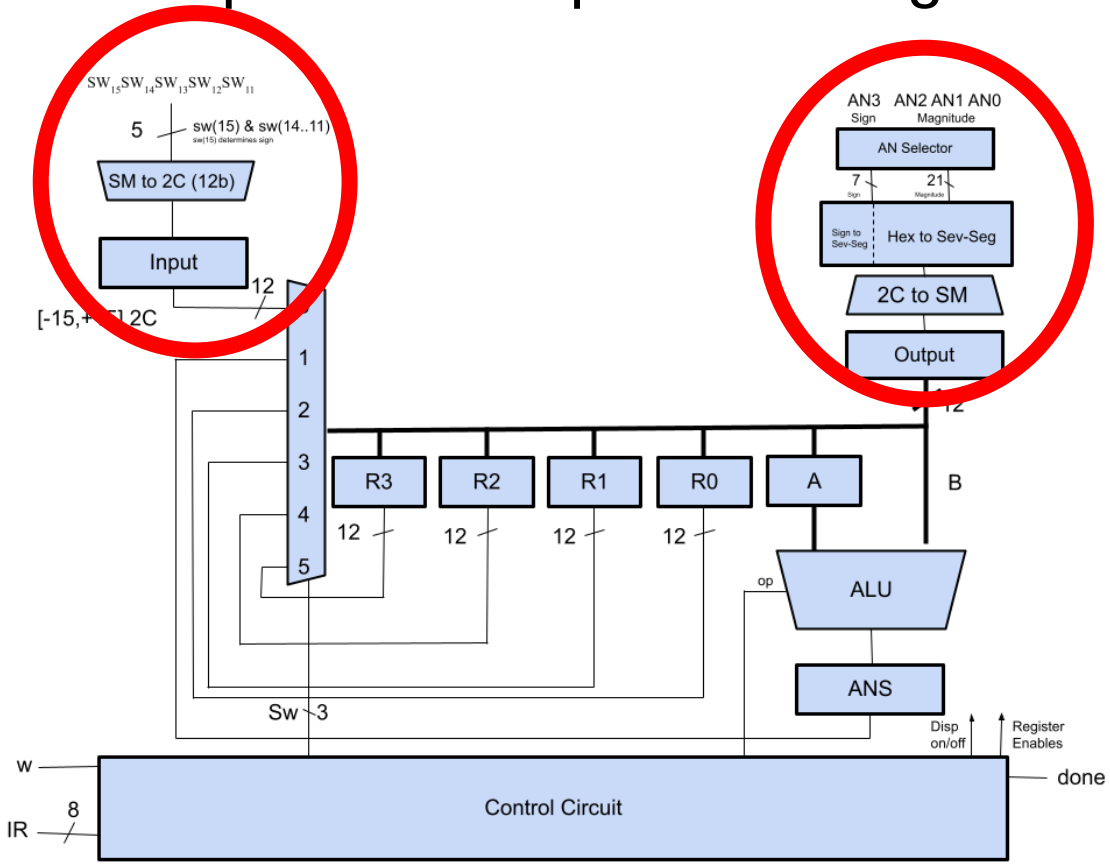
Output

- Four seven segment displays
  - First one displays sign: () or (-)
  - Hexadecimal display for others
- Output register stores pre-converted value

Output Range: [-2048, 2047]

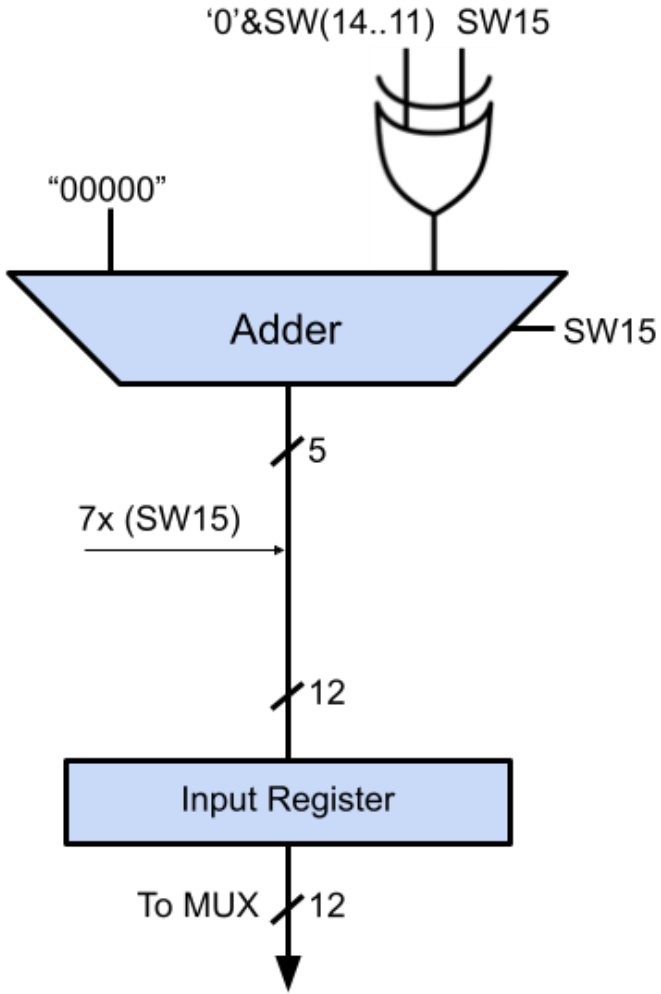
$\text{Binary(2C)}_{\text{12-bit}} \rightarrow \text{Hex (SM)}_{\text{3-bit + sign}}$

Input and Output Handling



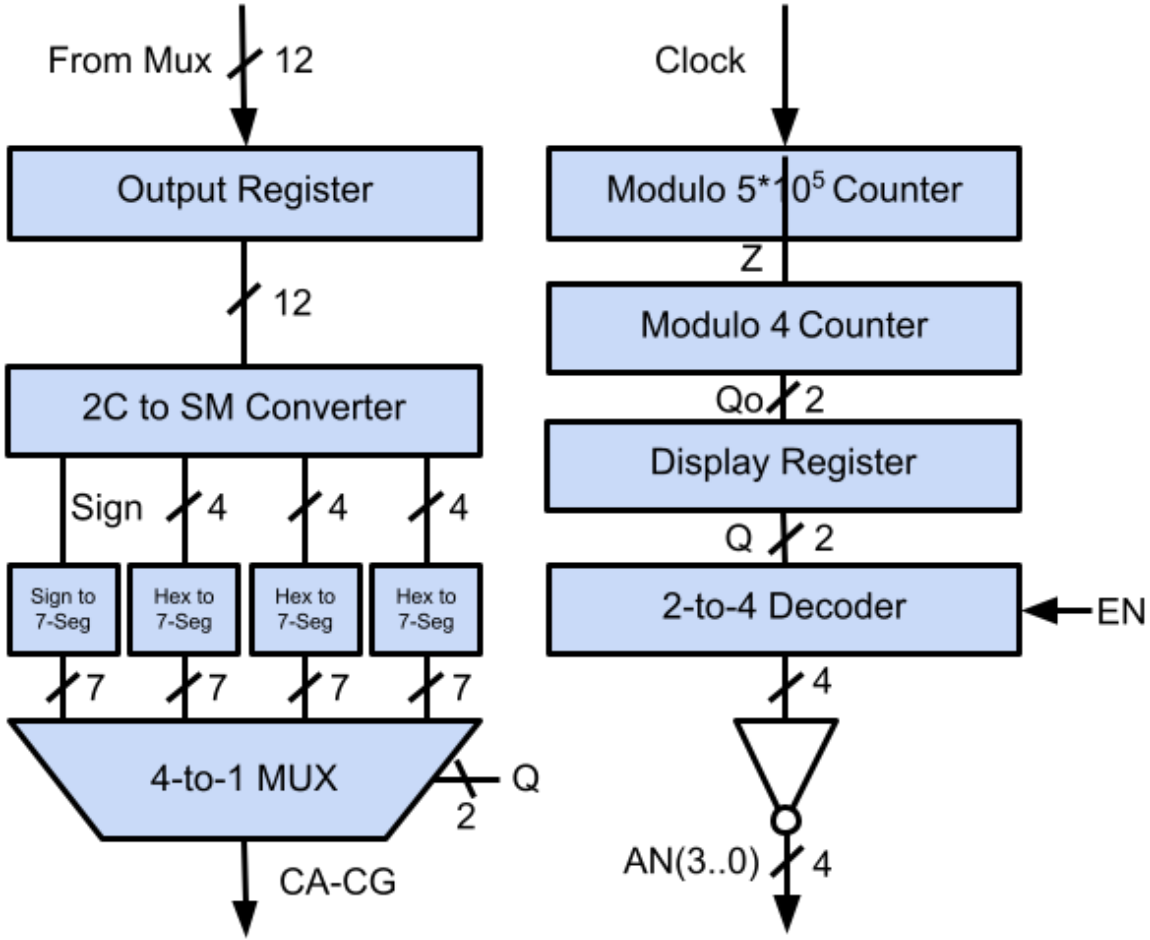
Input Module Design

- Binary input
- SW15 controls input sign
- SW 14 through 11 control magnitude

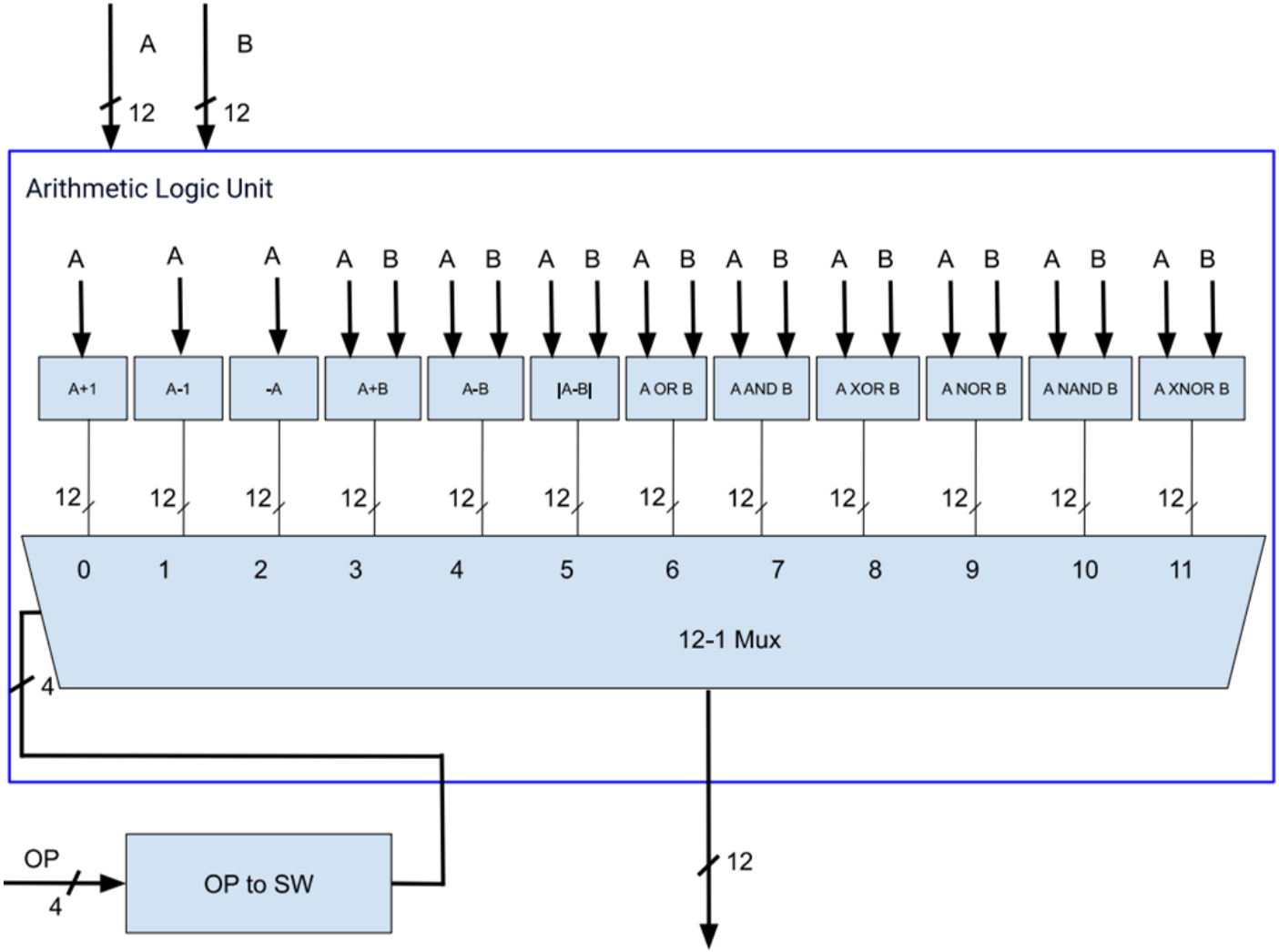


Output Module Design

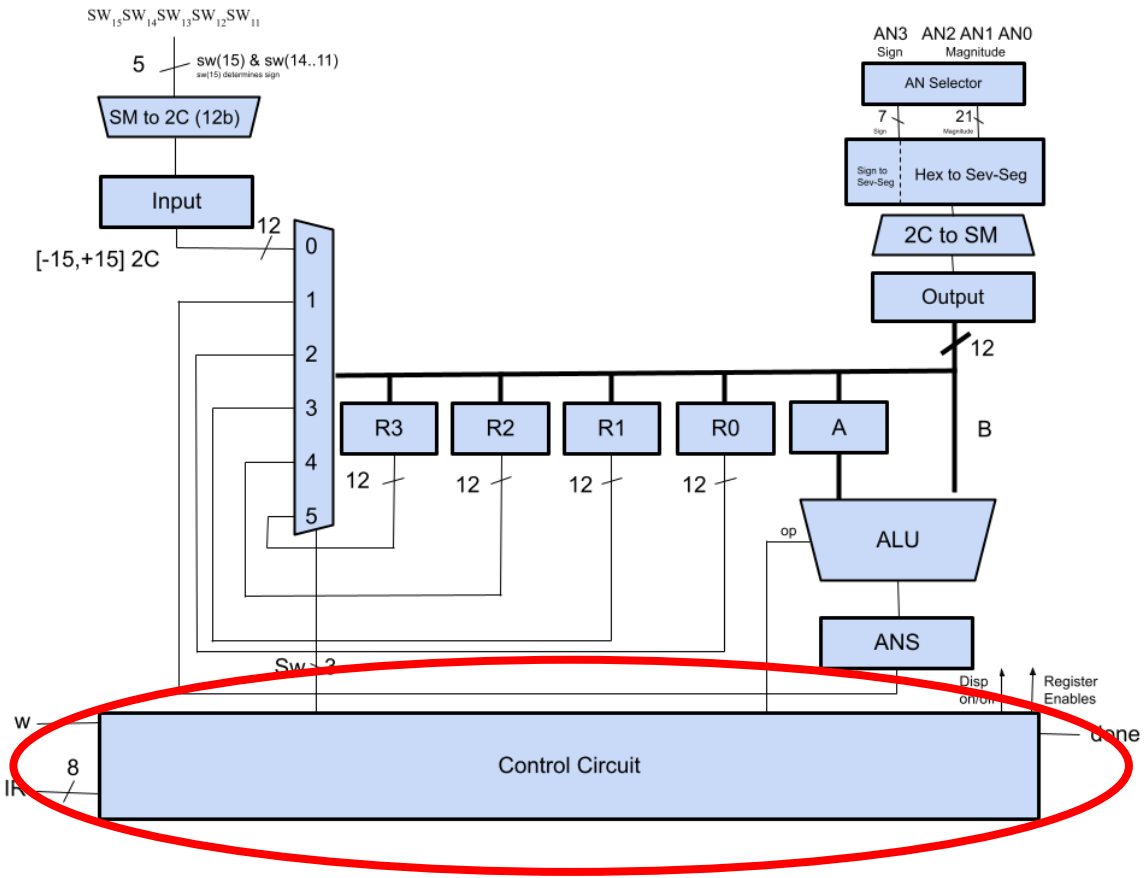
- 12-Bit 2C number sent to output register
- Modulo 4 Counter drives decoder and MUX
- Not gate for active-low setup



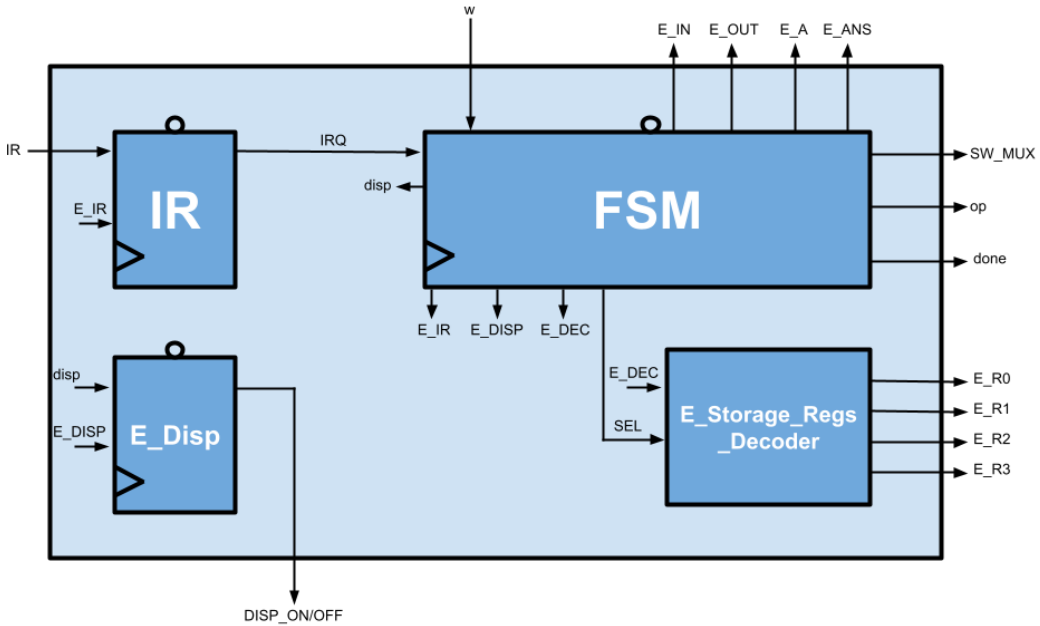
ALU


A
B
12
12
Arithmetic Logic Unit
A+1
A-1
-A
A+B
A-B
|A-B|
A OR B
A AND B
A XOR B
A NOR B
A NAND B
A XNOR B
0
1
2
3
4
5
6
7
8
9
10
11
12-1 Mux
4
OP
4
OP to SW
12

Control Circuit

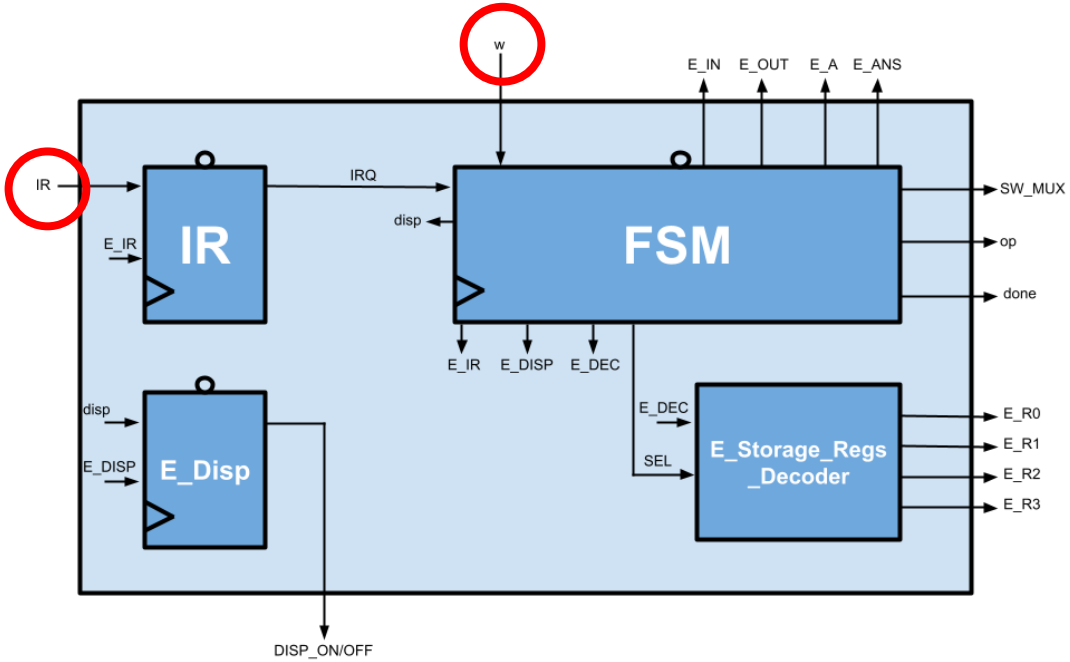


Control Circuit Overview



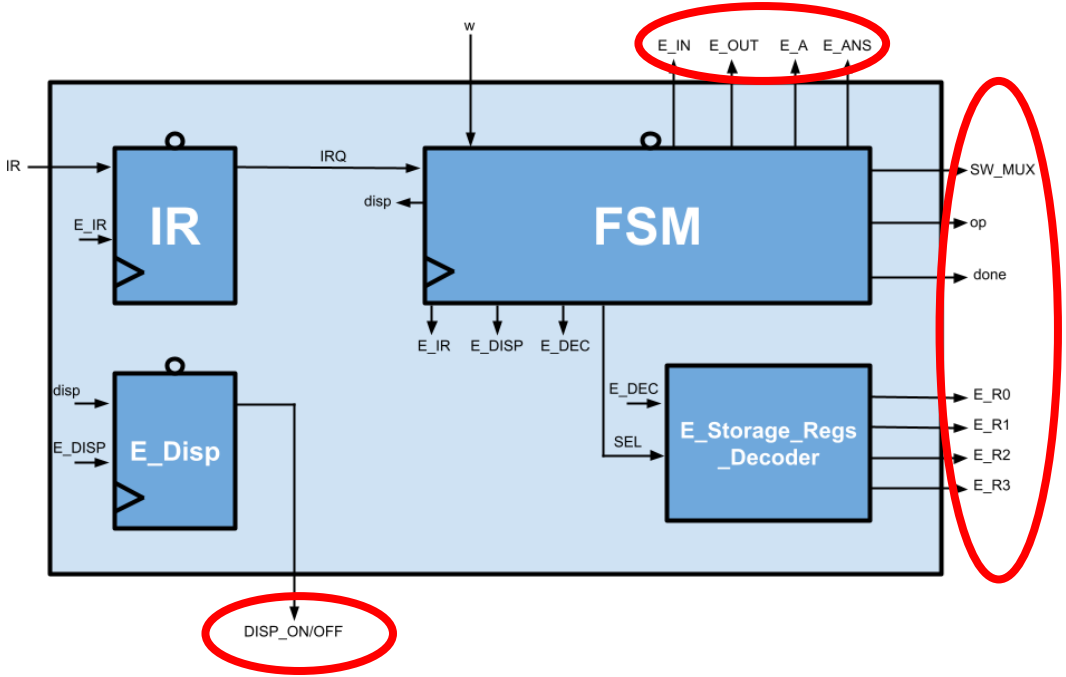
- Receives instructions from the user
- Sends operation code to ALU
- Controls flow of data
- Turns display on/off

Inputs



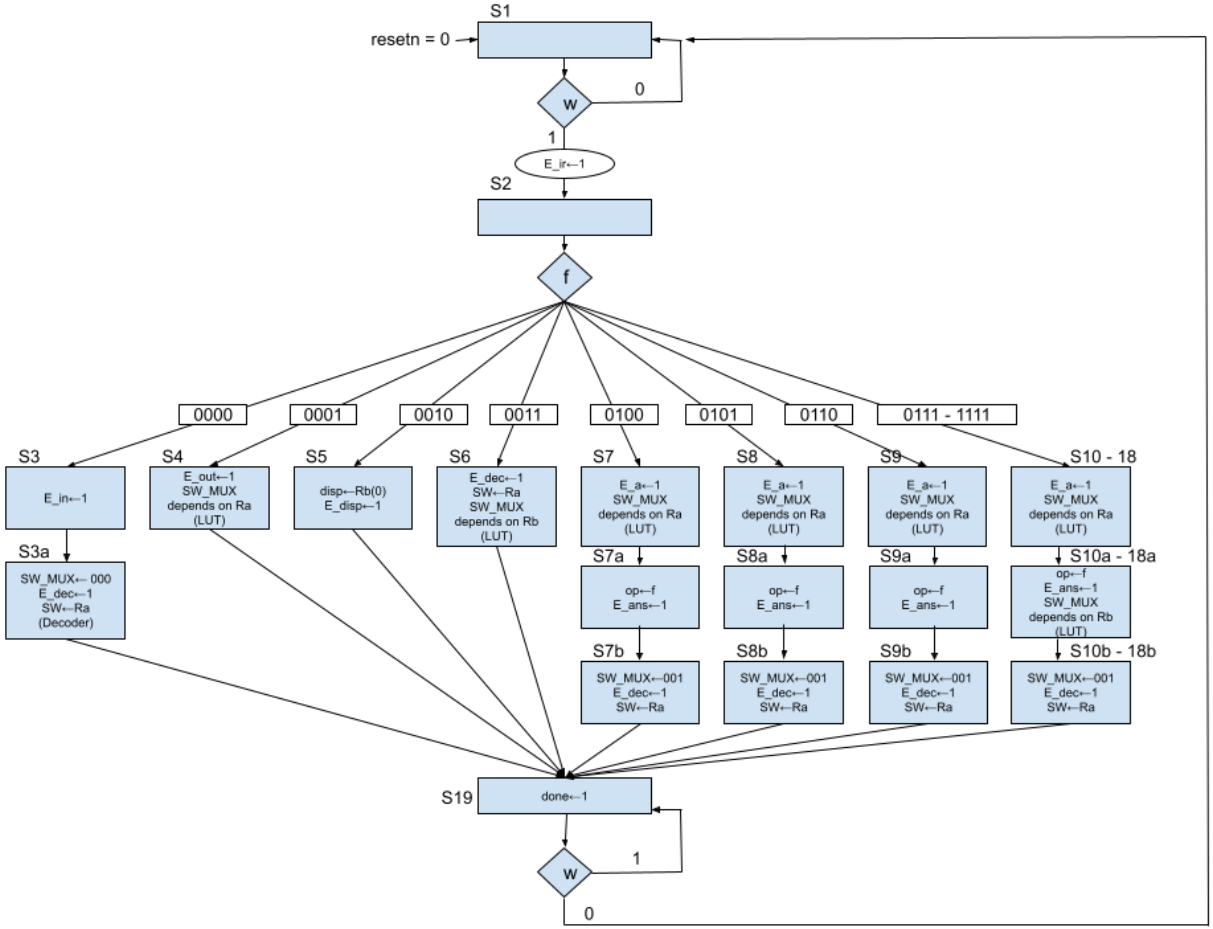
- Instruction (8 bits)
  - 16 Functions
  - 2 operands
- W (Switch)
  - Accept new instruction
  - Must be deactivated

Outputs



- Register Enables (7)
  - Control flow of data
- Mux Select (3 bits)
  - Mux has 6 inputs
- Op (4-bits)
  - Sent to ALU
- Done (LED)
- Disp_on/off (1/0)

ASM Diagram


S1
resetn = 0
w
0
1
E_ir←1
S2
f
0000
0001
0010
0011
0100
0101
0110
0111 - 1111
S3
E_in←1
S3a
SW_MUX← 000
E_dec←1
SW←Ra
(Decoder)
S4
E_out←1
SW_MUX
depends on Ra
(LUT)
S5
disp←Rb(0)
E_disp←1
S6
E_dec←1
SW←Ra
SW_MUX
depends on Rb
(LUT)
S7
E_a←1
SW_MUX
depends on Ra
(LUT)
S7a
op←f
E_ans←1
S7b
SW_MUX←001
E_dec←1
SW←Ra
S8
E_a←1
SW_MUX
depends on Ra
(LUT)
S8a
op←f
E_ans←1
S8b
SW_MUX←001
E_dec←1
SW←Ra
S9
E_a←1
SW_MUX
depends on Ra
(LUT)
S9a
op←f
E_ans←1
S9b
SW_MUX←001
E_dec←1
SW←Ra
S10 - 18
E_a←1
SW_MUX
depends on Ra
(LUT)
S10a - 18a
op←f
E_ans←1
SW_MUX
depends on Rb
(LUT)
S10b - 18b
SW_MUX←001
E_dec←1
SW←Ra
S19
done←1
w
1
0

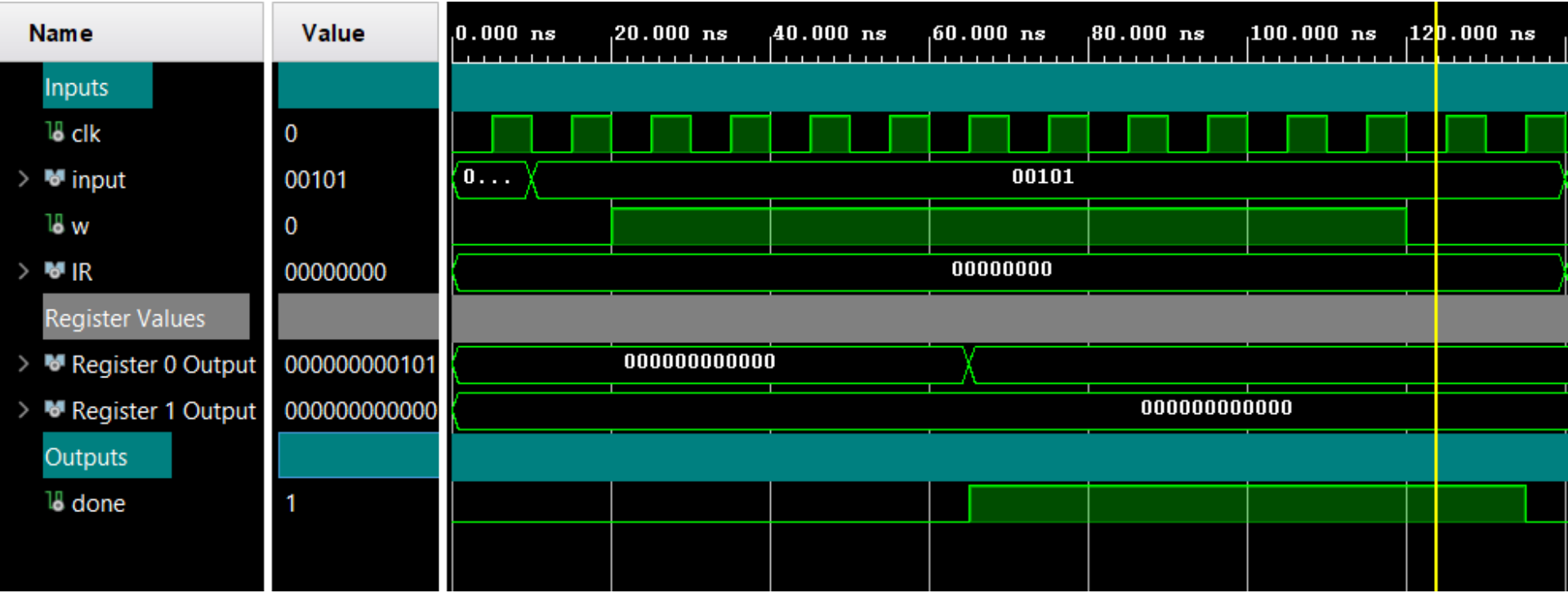
FSM Functions Table

IR Code	State	OP Code (if applicable)	Functionality
0000$R_{A1}R_{A0}$XX	S3	Not applicable	Store input into R_A
0001$R_{A1}R_{A0}$XX	S4	Not applicable	Display the value of R_A on 7-seg display
0010XXXD	S5	Not applicable	7 segment on/off (D = 1/0)
0011$R_{A1}R_AR_{B1}R_{B0}$	S6	Not applicable	Copy the value of R_B into R_A
0100$R_{A1}R_A$XX	S7	0100	$R_A=R_A+1$
0101$R_{A1}R_A$XX	S8	0101	$R_A=R_A-1$
0110$R_{A1}R_A$XX	S9	0110	$R_A=-R_A$
0111$R_{A1}R_AR_{B1}R_{B0}$	S10	0111	$R_A=R_A+R_B$
1000$R_{A1}R_AR_{B1}R_{B0}$	S11	1000	$R_A=R_A-R_B$
1001$R_{A1}R_AR_{B1}R_{B0}$	S12	1001	$R_A=\|R_A-R_B\|$

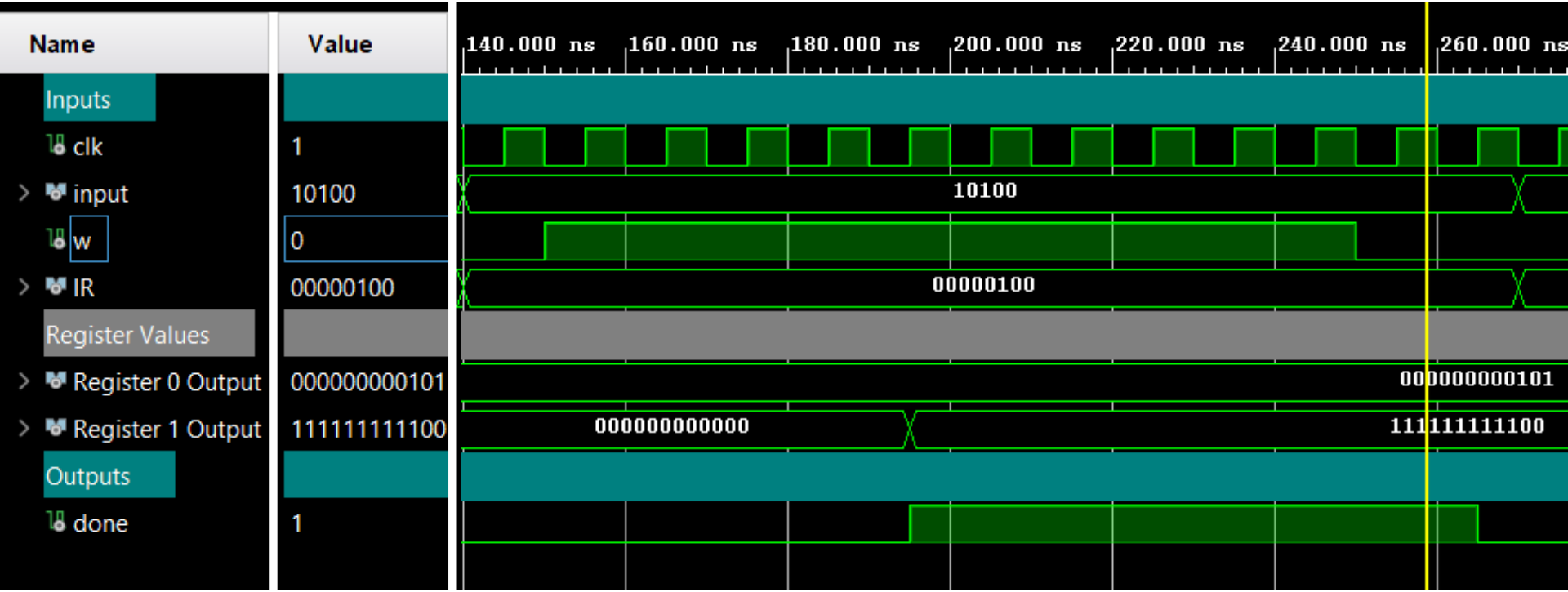
FSM Functions Table (Continued)

Bitwise Operations			
1010$R_{A1}R_AR_{B1}R_{B0}$	S13	1010	R_A=R_A OR R_B
1011$R_{A1}R_AR_{B1}R_{B0}$	S14	1011	R_A=R_A AND R_B
1100$R_{A1}R_AR_{B1}R_{B0}$	S15	1100	R_A=R_A XOR R_B
1101$R_{A1}R_AR_{B1}R_{B0}$	S16	1101	R_A=R_A NOR R_B
1110$R_{A1}R_AR_{B1}R_{B0}$	S17	1110	R_A=R_A NAND R_B
1111$R_{A1}R_AR_{B1}R_B$	S18	1111	R_A=R_A XNOR R_B

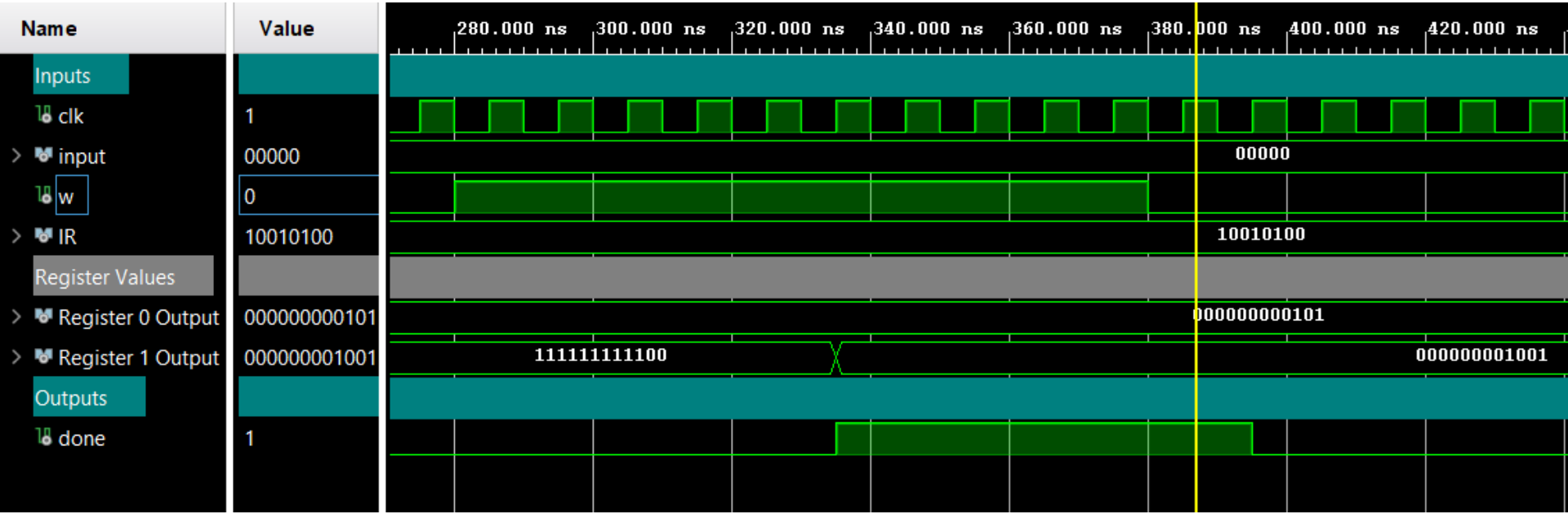
Sample Simulation: Store +5 into Register 0



Sample Simulation: Store -4 into Register 1

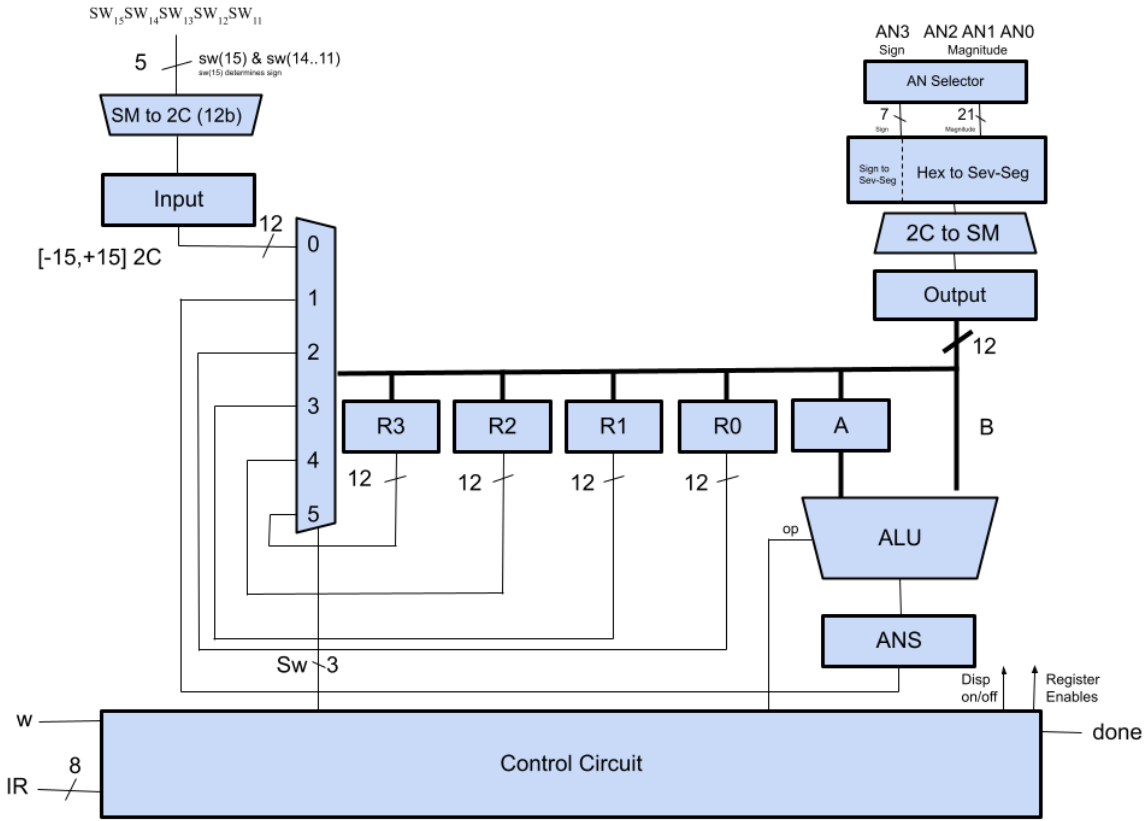


Sample Simulation: Register 1 = |Register 1 - Register 0|



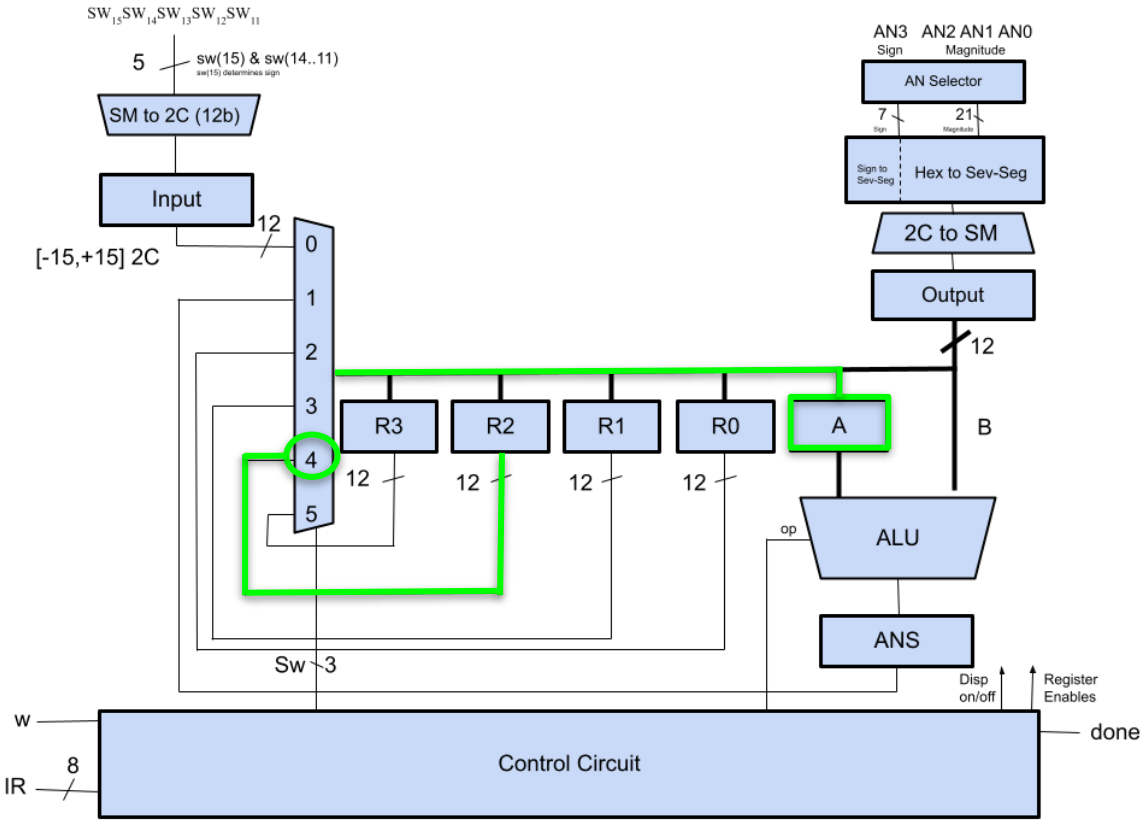
Example: Add R2 to R1

IR = 01111001
W = 1



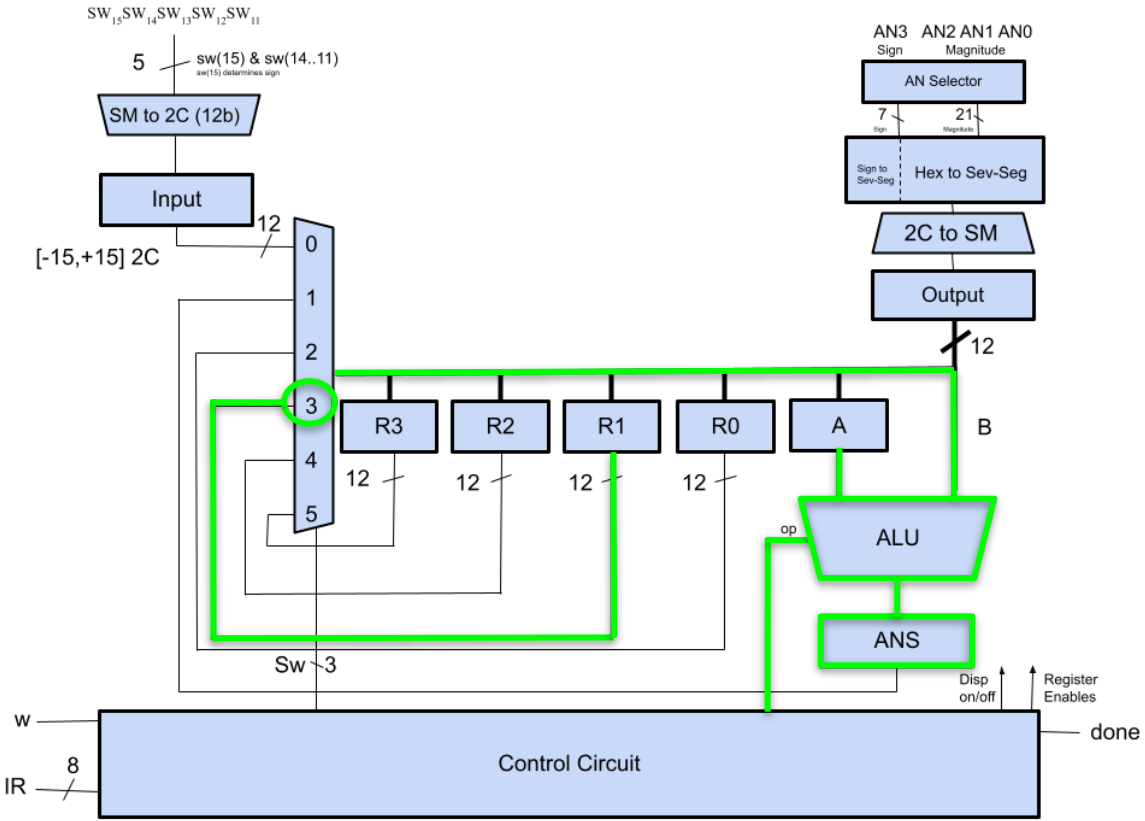
Example: Add R2 to R1

IR = 01111001
W = 1



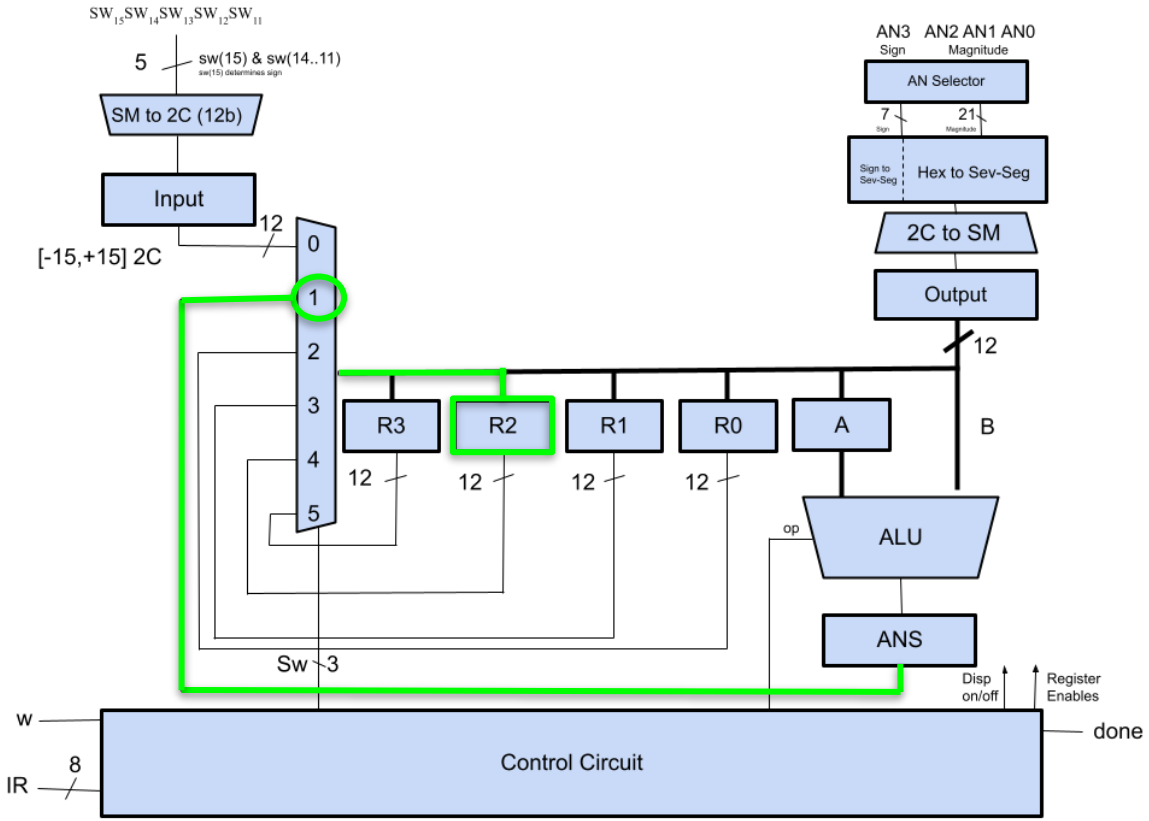
Example: Add R2 to R1

IR = 01111001
W = 1



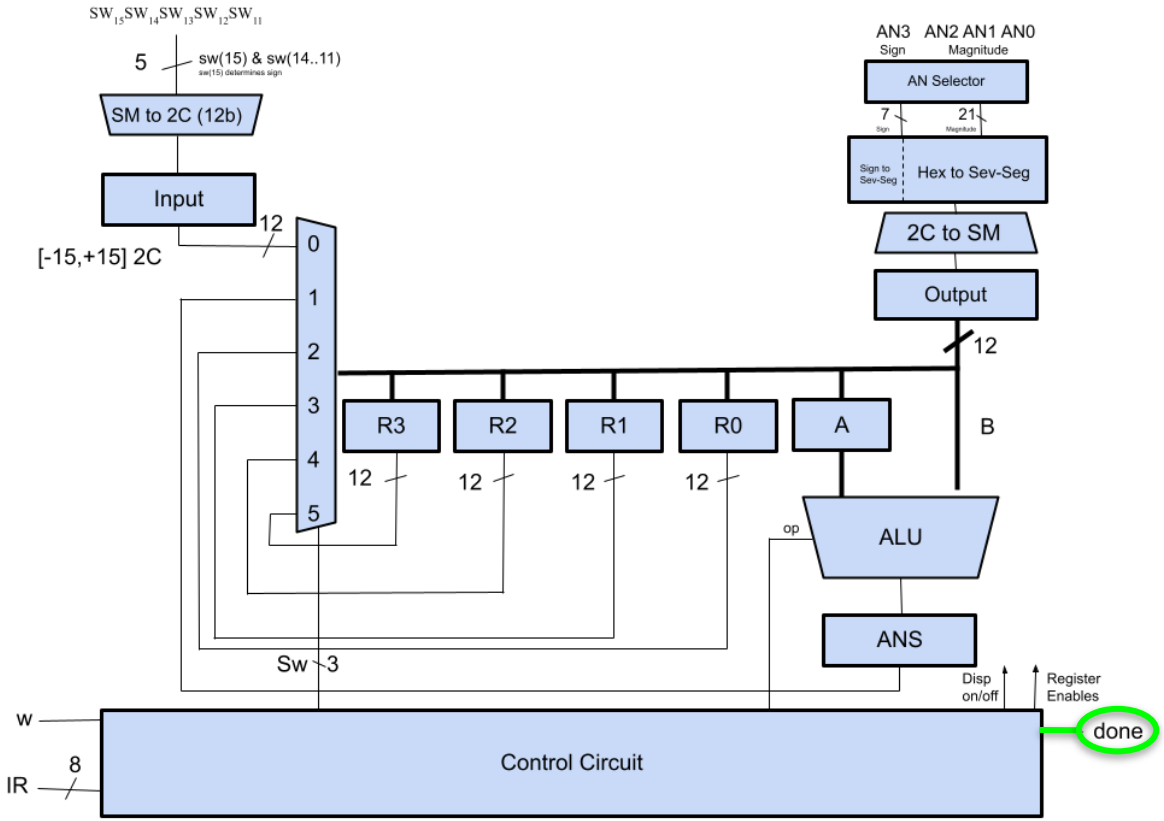
Example: Add R2 to R1

IR = 01111001
W = 1



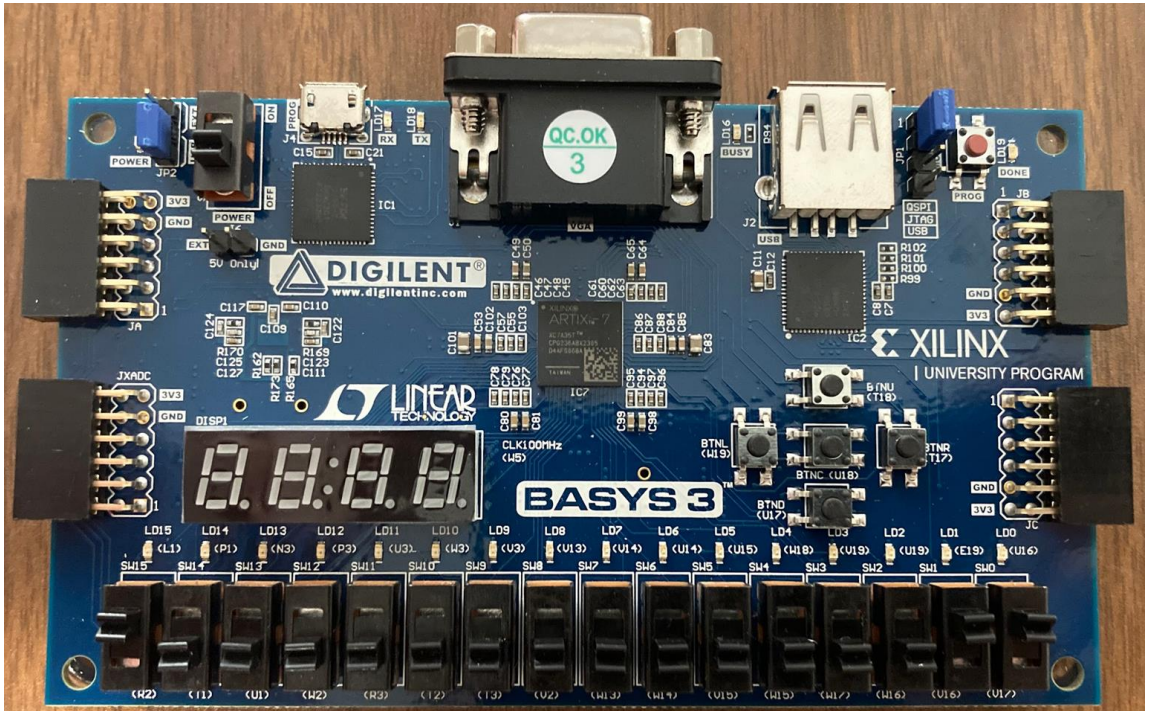
Example: Add R2 to R1

IR = 01111001
W = 1



(4 Clock ticks)

Board Switches



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Sign	Input (3)	Input (2)	Input (1)	Input (0)		w	IR (7)	IR (6)	IR (5)	IR (4)	IR (3)	IR (2)	IR (1)	IR (0)	Resetn

Demonstration