50.002 Computation Structure

1005398 Rohit Raghuram Murarishetti
1005427 Chiam Jia Hui
1005513 Lee Pei Xuan
1005393 Umang Gupta
1005056 Lee Cheng Xin
1005234 Ernest Lim Wei Yu
1005289 Lee Chang Zheng
1005622 Parwani Jayati

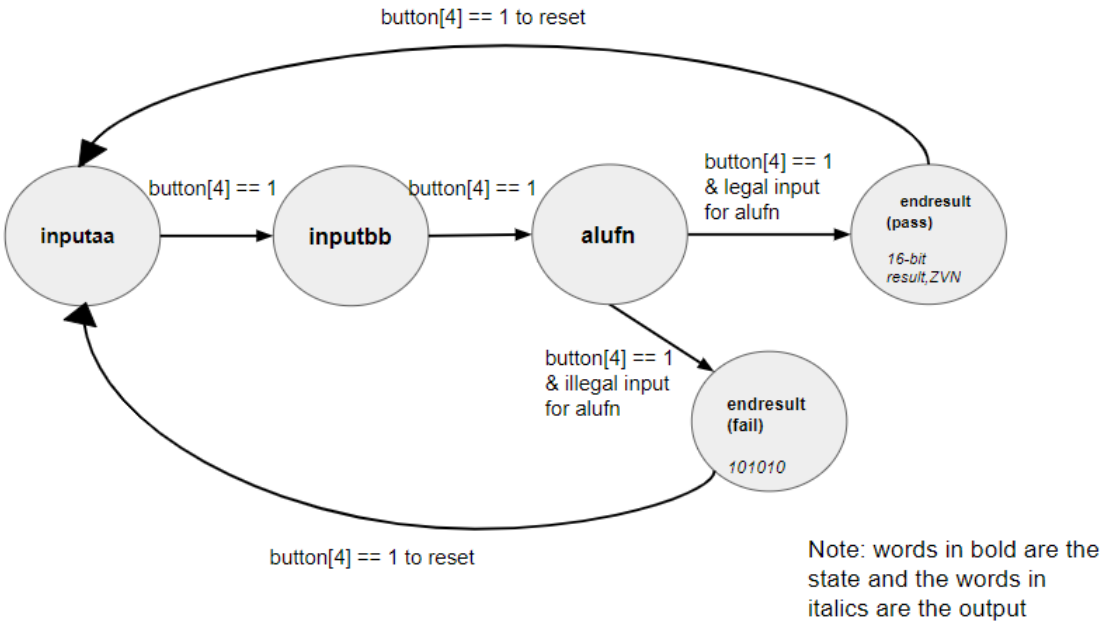
ALUFN table

Operation	**ALUFN[5:0]**	**Hex**
ADD	000000	0x00
SUB	000001	0x01
MULTIPLY	001000	0x08
DIVIDE	001001	0x09
AND	011000	0x18
OR	011110	0x1E
XOR	010110	0x16
“A” LDR	011010	0x1A
SHL	100000	0x20
SHR	100001	0x21
SRA	100011	0x23
CMPEQ	110011	0x33
CMPLT	110101	0x35
CMPLE	110111	0x37
INC	000011	0x03
DEC	000111	0x07
ERROR/ILLEGAL	All other ALUFN implementations	All other ALUFN implementations

Our implementation follows the LAB 3 ALU handout and the underlined operations are the extra features we implemented.

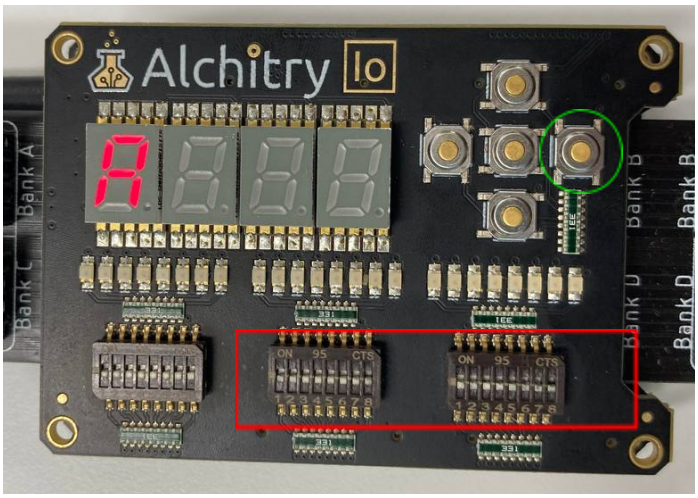
Schematics

The diagram below is a broad overview of the design of our Finite State Machine. Each state is explained in detail on the next page.

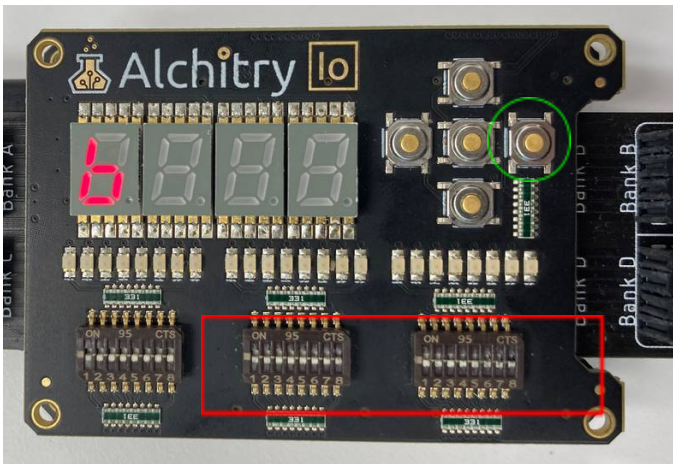


Manual Testing

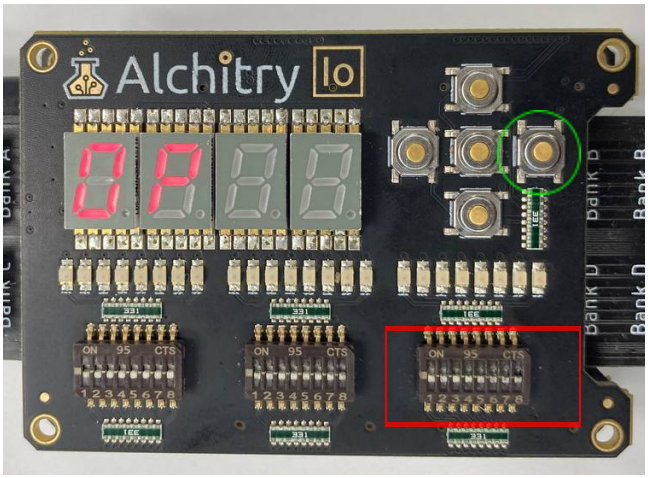
1. When the Alchitry device is powered up, A will be displayed in the first segment which indicates State “INPUTAA”.
   The user should input the 16 bits A values at this stage on io_dip[0] and io_dip[1]. After doing so, press the right button to proceed to input the values of B.



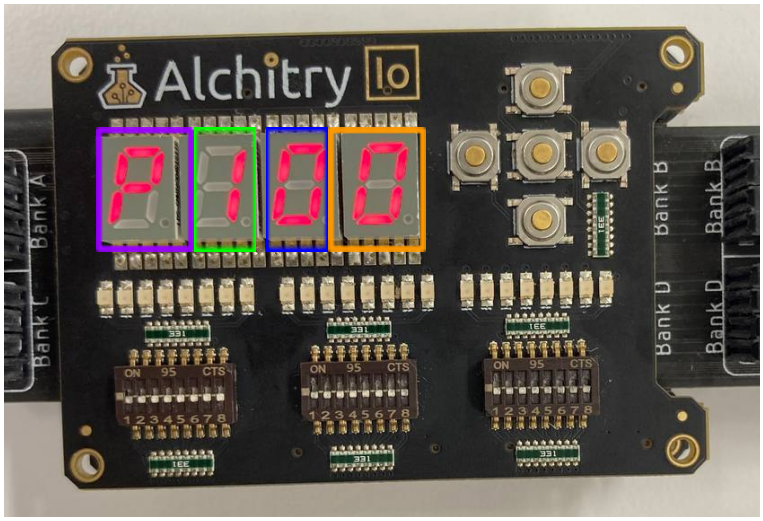
2. B will then be displayed in the first segment, which indicates State “INPUTBB”. The user should then input the 16 bits B values at this stage on io_dip[0] and io_dip[1]. Press the right button to proceed to input the values of ALUFN.



3. Next, OP will be displayed on the first and second segment, the user should input the 6 bits ALUFN values in io_dip[0][5:0].



4. If the OPCODE typed inside is invalid (not included in our ALUFN table), FAIL would be shown on the segments with an output of 101010101010 on the Io_led, otherwise the segments will display P, followed by the Z V N values as shown below. The output value of the operation between A and B will be displayed in the Io_led.



Above picture shows a valid case which returns all different outputs from ALU after computing a and b with the specific operation. Purple box indicates “PASS” which means all inputs are valid and there is a result. Green box refers to the output Z, blue box refers to the output V and orange box refers to the output N.

Above picture shows a invalid case which returns a “FAIL”. This occurs when an invalid opcode is entered.

Special Case

However, it also happens when the result of the operation of a and b is 1010101010101010. This makes 1010101010101010 a simulated error, which is unavoidable due to the design of our code, although the original alu result is correct.