5 Arithmetic Logic Unit

I Overview

An Arithmetic Logic Unit (ALU) allows many pre-defined functions to be implemented on two binary inputs. We will look at a simple ALU that can perform four functions - three logical functions and one arithmetic function. *Select* inputs to the ALU are used to choose a particular ALU function.

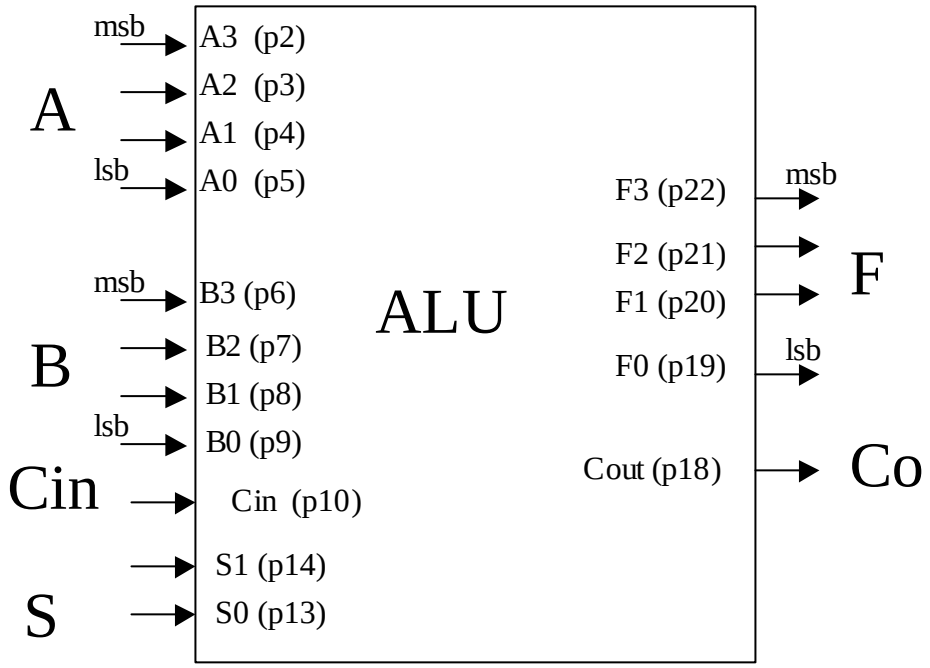
A Multiplexer (mux for short) or data selector allows the steering of 1 of many inputs to an output. Usually, a mux will have multiple input busses, and a select input can be used to steer one of the input busses to the output bus. A 2-to-1 mux will have two inputs, a 4-to-1 mux will have four inputs, etc. We will use a 2-to-1 mux that has two 4-bit inputs. This mux is also capable of other logical operations in addition to just steering one of the two 4-bit inputs to the 4-bit output.

In this lab, you will combine the mux and the ALU together to form a 'super' ALU that has more functionality than the original ALU by itself.

II. ALU

A. The TA will program one of your PLDs with the ALU function. The pinout is shown below (do not forget that Vdd is pin 24, GND is pin 12).

Figure 1



data input switches: S, Ci, A
LED: Co, F
hardwired: B

B. The truth table for the ALU is shown below (the '+' is an addition operation):

S1	S0	Function
0	0	F = A and B
0	1	F = A or B
1	0	F = A xor B
1	1	F = A + B + Cin

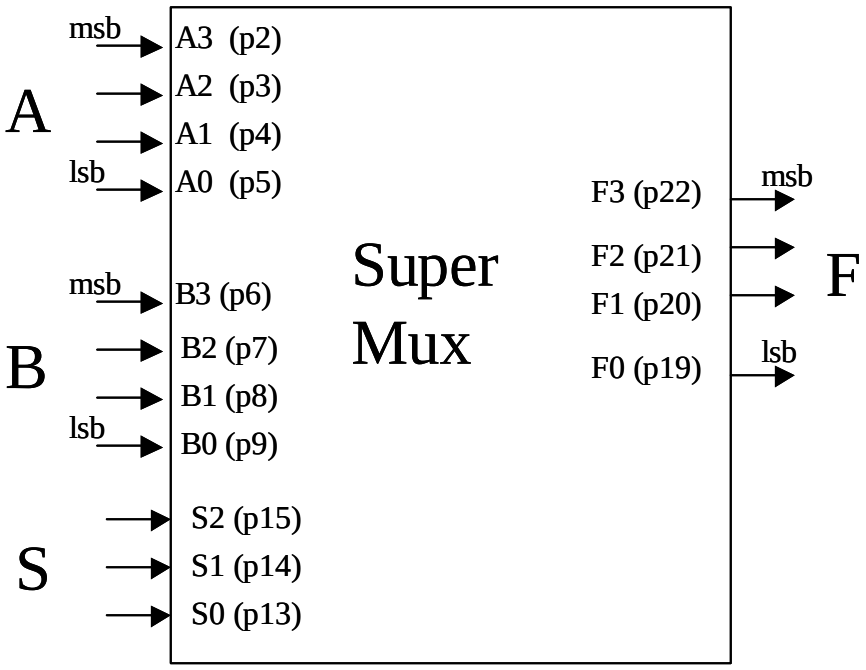
Hardwire the value of the 'B' input to the value of the last digit in your Student ID# plus 1 (i.e. if your last digit is '4', hardwire it to '5'). Apply the value of 7 to the 'A' input. Verify the circuit functionality by filling out the Table for part II in the lab data sheet.

C. Do not disconnect the circuit; it will be used in part III.

III. Data Selector

A. The TA will program one of your PLDs with the Super Mux function. The pinout is shown below (do not forget that Vdd is pin 24, GND is pin 12).

Figure 2

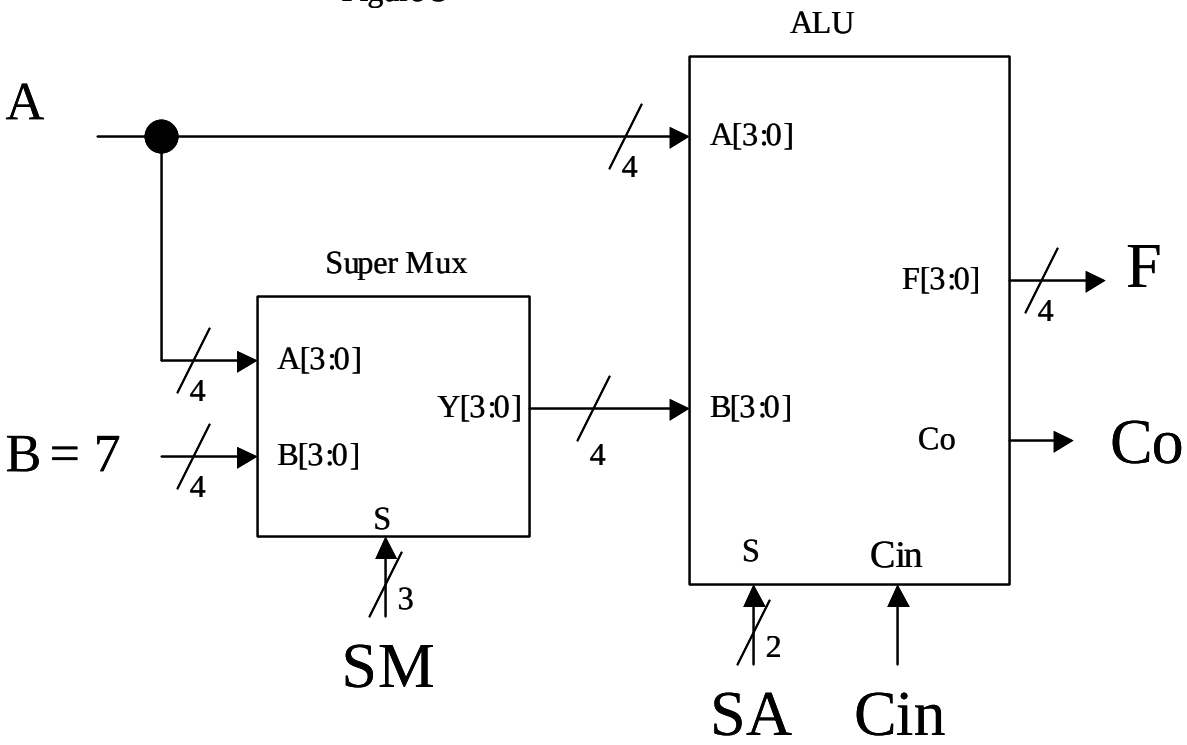


The truth table of the 'super mux' is shown below. It is called a super mux because it does more than just the 2/1 mux function.

S2	S1	S0	Function
0	0	0	F = A
0	0	1	F = not (A)
0	1	0	F = B
0	1	1	F = not (B)
1	0	X	F = "0000"
1	1	X	F = "1111"

1. Using your ALU and Super Mux PLDs, connect the circuit shown in Figure 3. This combination of the Super Mux + ALU will give us a 'Super ALU' that has more functions than the original ALU.

Figure 3



Connect the SM select lines, SA select lines, Cin to input switches. D1, D2, D3,
Hardwire the B input to the value shown.
Hardwire the 'A' input to the value of your last Student ID digit + 2 (i.e. if your last student ID digit is 8, hardwire the value to 10).

B. Use the circuit above and fill out the table in Part III of the lab data sheet using the values of A, B specified above. Have the TA check your circuit after you have it working.

PRE-LAB Data Sheet **TA Checkoff_____**

Student ID Number: _____

1. What is an ALU?

2. What is a Mux?

3. Fill out the following table to achieve the given functions. Use a value of 7 for the A input. For the 'B' input, use the value of the last digit in your Student ID# plus 1 (i.e. if your last digit is '4', hardwire it to '5'). The truth table for the ALU can found in Section II of the lab. In the table below, the '+' symbol is an addition operation.

					Expected Output				
Function	A	B	S1	S0	Co	F3	F2	F1	F0
F = A and B									
F = A or B									
F = A xor B									
F = A + B + (Cin = 0)									
F = A + B + (Cin = 1)									

4. Refer to Figure 3 in Part III. What are the required input control conditions needed to implement each of the following functions? The function must be valid for all input values of A, B (the '+' symbol is the addition operation, the '-' symbol is the subtraction operation). There can be more than one correct answer. Remember that the subtraction operation "A - B" can be implemented via 2's complement as "A + not(B) + 1".

    1) F = A + B
    2) F = A + 1
    3) F = A - B
    4) F = A and B
    5) F = A xor not (B)
    6) F = "0000" (must be valid for all combinations of A, B)
    7) F = "1111" (must be valid for all combinations of A, B)
    8) A + A
    9) A - 1
    10) Not (A) (hint: the 'xor' operation is involved in this)

In the 'func' columns for the 'ALU', 'Super Mux' write the function that corresponds to the 'S' inputs that you are using. Use the values of 'A', 'B' given in Figure 3 in Part III to fill out the 'Expected Output' columns.

Function	**ALU**				**Super Mux**				**Expected Output**				
	Cin'	S1	S0	Func	S_2	S_1	S_0	Func	Cout	F3	F2	F1	F0
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													

5. Can the function 'not(B)' be implemented for all values of 'A' (explain why or why not)? Can the function 'not (B)' be implemented for particular values of 'A' (if yes, give these values and explain the ALU, Super Mux functions needed).

6. Give three different ways of accomplishing the function F = "0000" using the circuit in Figure 3.

LAB Data Sheet **TA Checkoff_____**
Student ID Number: _____

II. ALU

					Observed Output				
Function	A	B	S1	S0	Co	F3	F2	F1	F0
F = A and B									
F = A or B									
F = A xor B									
F = A plus B plus (Cin = 0)									
F = A plus B plus (Cin = 1)									

III Super ALU = ALU + Super Mux.
Using your data from the prelab, fill out the table below. Refer to Figure 3 in Part III.

1) F = A + B
2) F = A + 1
3) F = A - B
4) F = A and B
5) F = A xor not (B)
6) F = "0000" (must be valid for all combinations of A, B)
7) F = "1111" (must be valid for all combinations of A, B)
8) A + A
9) A - 1
10) Not (A) (hint: the 'xor' operation is involved in this)

In the 'func' columns for the 'ALU', 'Super Mux' write the function that corresponds to the 'S' inputs that you are using. Use the values of 'A', 'B' given in Figure 3 in Part III to fill out the 'Observed Output' columns.

Function	**ALU**				**Super Mux**				**Observed Output**				
	Cin'	S1	S0	Func	S_2	S_1	S_0	Func	Cout	F3	F2	F1	F0
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													