ALU Project Report

Submitted By: Colin Christie

Function Table

S_3	S_2	S_1	S_0	C-IN	Operation	Function
0	0	0	0	0	F = A + B	Add
0	0	0	0	1	F = A + B + 1	Add with carry
0	0	0	1	0	F = A + B’	Subtract with borrow
0	0	0	1	1	F = A + B’ + 1	Subtract
0	0	1	0	0	F = A	Transfer A
0	0	1	0	1	F = A + 1	Increment A
0	0	1	1	0	F = A - 1	Decrement A
0	0	1	1	1	F = A	Transfer A
0	1	0	0	X	$F = A \wedge B$	AND
0	1	0	1	X	$F = A \vee B$	OR
0	1	1	0	X	$F = A \oplus B$	XOR
0	1	1	1	X	F = A’	Compliment
1	0	X	X	X	F = shr A	Shift right A
1	1	X	X	X	F = shl A	Shift left A

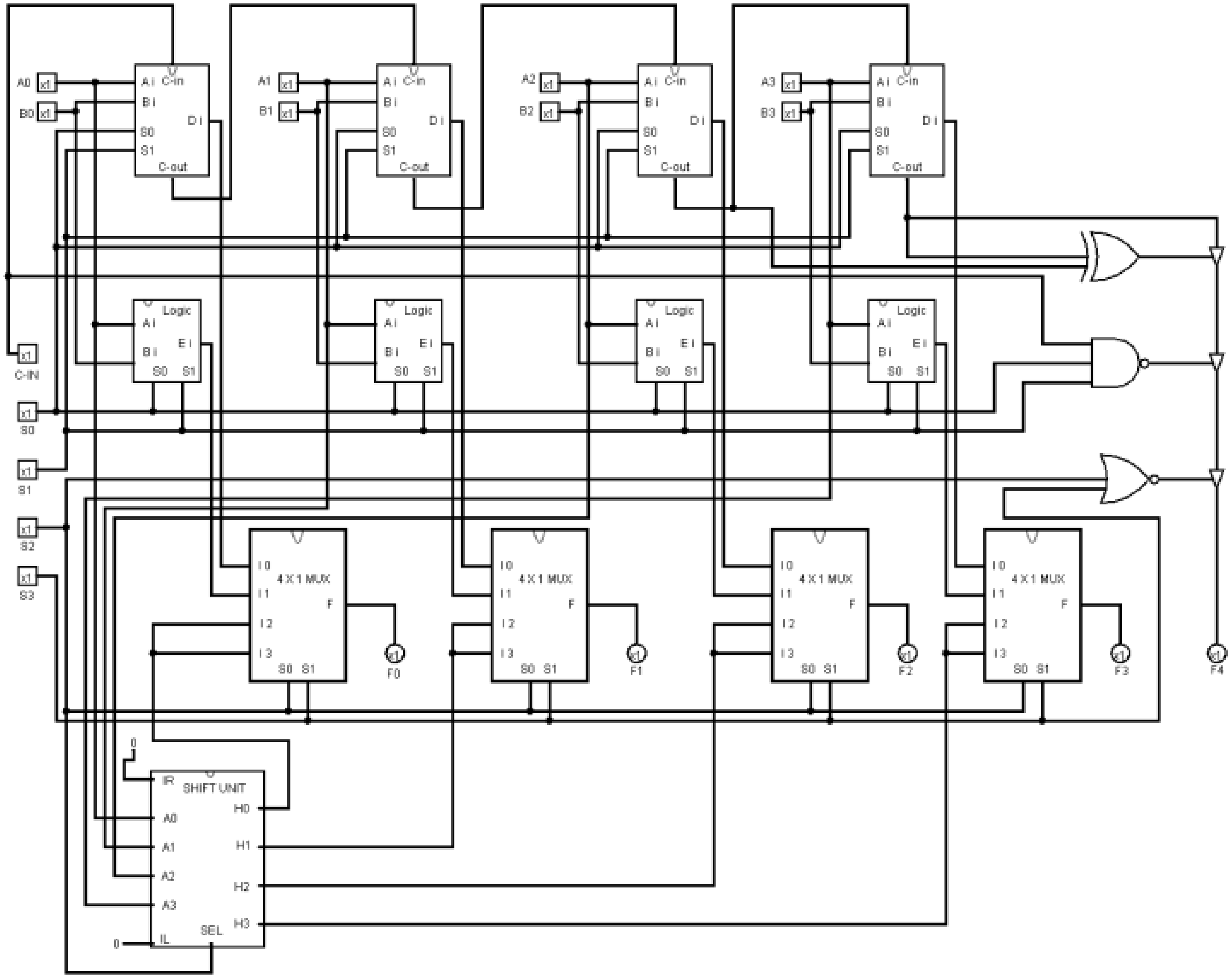
Description of Functions:

- The first 8 operations are arithmetic operations and are performed when S_3S_2 = 00
- The next 4 operations are logical operations which are performed when S_3S_2 = 01 and the C-IN has no effect on the output so it is marked with don’t care X’s
- The last 2 operations are logical shift and are performed when S_3S_2 = 10 and 11. S_1 S_0 and C-IN have no effect on the output and are marked with don’t care X’s

Please note that for all arithmetic operations the inputs and outputs are in 2’s complement form so that negative numbers can be represented. The valid range of input numbers is from +7 $(0111)_2$’s to -8 $(1000)_2$’s. The result may require a carry which is checked by an XOR condition of the carry outs of the 2 most significant bits of the arithmetic circuits. If a carry is needed F_4 is activated through a controlled buffer and the output will be a 5-bit 2’s compliment number. If a carry is not needed F_4 will be in a low impedance state and the output will be a 4-bit 2’s complement number.

For all other operations there is no assumption about whether the input is in 2’s compliment or not. The output will be a bitwise logical operation. This means that when shifting the most significant bit will be lost for a right shift, and the least significant bit will be lost for a left shift.

Arithmetic Logic Shift Unit



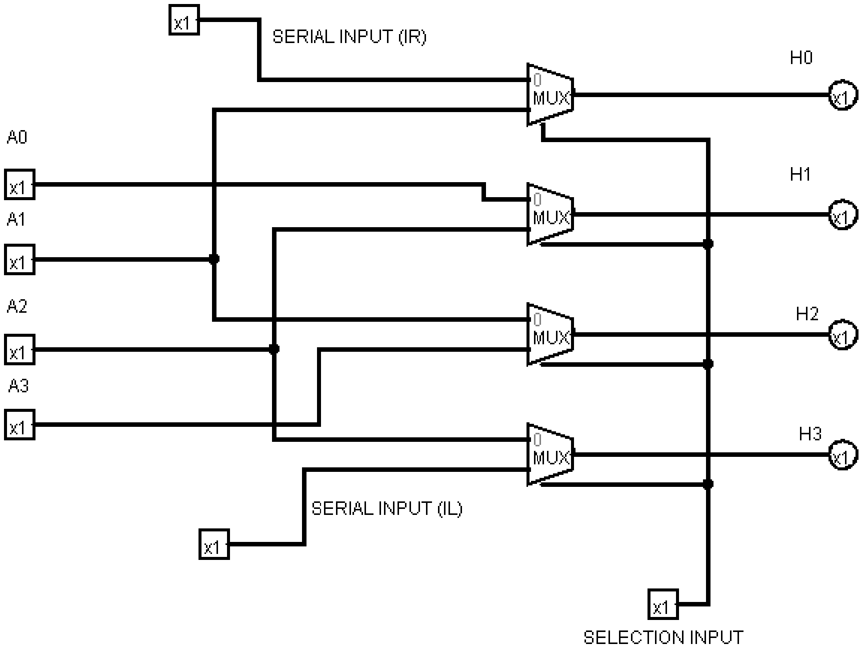
This is a diagram of the main ALU module. There are 4 arithmetic, logic and multiplexer circuits and 2 4-bit inputs A and B. The A and B inputs are both connected to the arithmetic and logic circuits. C-IN is connected to the carry in of the first arithmetic circuit and is used for selecting arithmetic operations. Both the arithmetic and logic circuits have 4x1 multiplexers in them which determine the output. S_0 and S_1 are connected to the selection bits of the logic and arithmetic multiplexers. S_0 S_1 and C-IN are also connected to a NAND gate whose output is checked by a controlled buffer so that when all three of them are activated the F_4 carry out will be disabled during the transfer A function.

S_2 and S_3 are connected to the selection bits of the 4x1 multiplexers which have the function outputs. S_2 and S_3 are also connected to a NOR gate whose output is checked by a controlled buffer so that if either S_2 or S_3 are activated the F_4 carry out will be disabled during a logic or shift operation.

The carry out of the 2 most significant bits of the arithmetic circuits are passed through an XOR gate whose output is checked by a controlled buffer so that if a carriage is required during arithmetic operations the F_4 carry out will be activated.

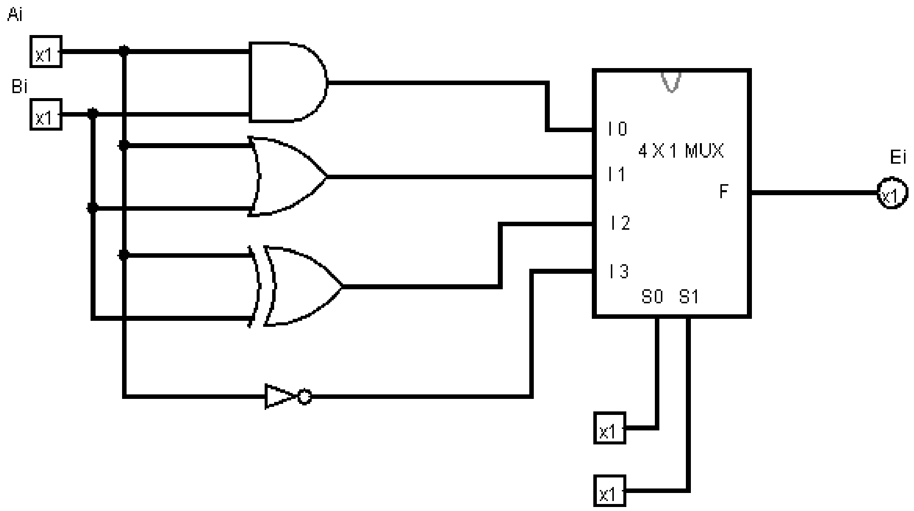
All the A inputs are also sent to the Shift unit which contains a 2x1 multiplexer whose selection bit is controlled by S_2 for a left or right shift

Shift Circuit



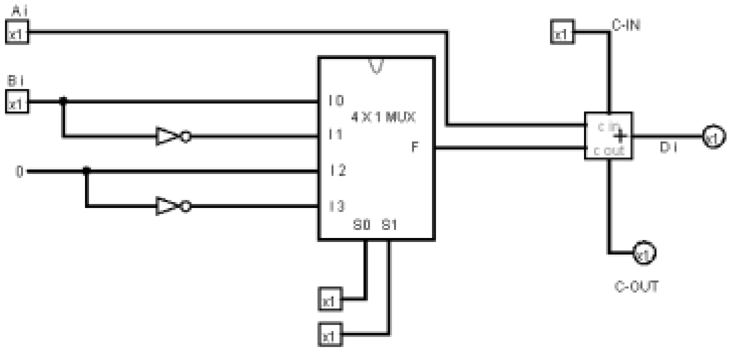
The shift circuit has 4 inputs which represent a 4-bit binary number. Each input is connected to a 2x1 MUX which has a selection input for either a right or a left shift. This type of shift circuit can perform a circular, logical, or arithmetic shift. It will be used to perform only a logical shift for this ALU so the serial inputs will have a constant 0 coming in. The A0 and A3 inputs can only go to H0 or H2 respectively. This means if there is a shift left A0 will be lost and if there is a shift right A3 will be lost. A1 and A2 can be shifted left or right without loss of bits.

Logic Circuit



The logic circuit has 2 inputs A and B which will be the input bits for a single stage in the ALU and will be used when $S_2 = 1$. It performs 4 logic operations: AND, OR, XOR, and complement. For $S_1 S_0 = 00$ E will be the AND operation. $S_1 S_0 = 01$ will perform an OR. $S_1 S_0 = 10$ will perform XOR. $S_1 S_0 = 11$ will give the complement of A.

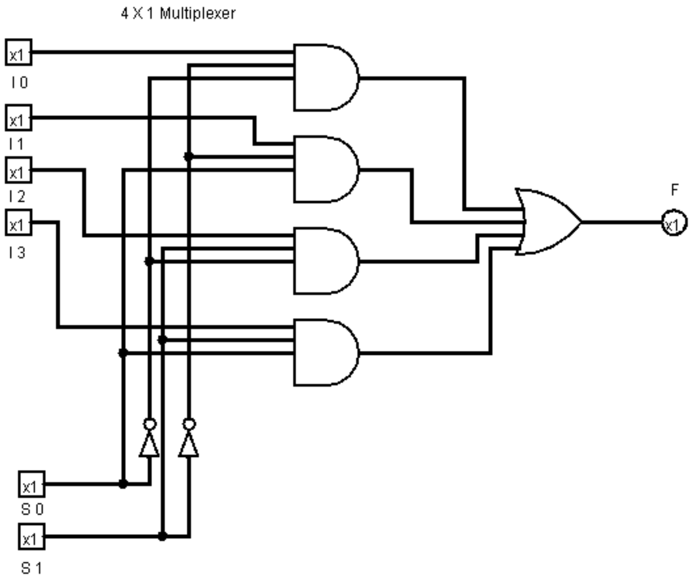
Arithmetic Circuit



The arithmetic circuit has 3 inputs, A and B which are 1-bit inputs for each stage of the ALU, and C-IN which is the carry in from the previous stage. On the first stage of the ALU C-IN is used as the 3rd selection bit for arithmetic operations. The A input is sent directly to the first input of a binary full adder while B and B's complement are sent to inputs 0 and 1 of a 4x1 MUX. Inputs 2 and 3 of the MUX are permanently set to 0 and 1 respectively and are used for increment and decrement operations. The Arithmetic circuit can perform 7 different operations.

When S_1S_0Cin = 000 an add operation is performed and the output will be the sum of A and B. When S_1S_0Cin = 001 an add with carry operation will be performed and the output will be the sum of A and B plus 1. When S_1S_0Cin = 010 the subtract with borrow operation is performed and the output will be A plus the compliment of B. When S_1S_0Cin = 011 the subtract operation is performed and the output will be A minus B. When S_1S_0Cin = 100 A is transferred directly to the output. When S_1S_0Cin = 101 A is incremented and the output will be A plus 1. When S_1S_0Cin = 110 A is decremented and the output is A minus 1. When S_1S_0Cin = 111 A is transferred directly to the output. There are 8 possible input combinations but 2 have the same output, so this circuit can perform 7 unique operations.

4-to-1 Multiplexer



The 4-to-1 multiplexer is used to select 1 of 4 input bits to output. The 2 selection bits decide which of the 4 input bits will be the output. When S_1S_0 = 00 F = I_0. When S_1S_0 = 01 F = I_1. When S_1S_0 = 10 F = I_2. When S_1S_0 = 11 F = I_3. Four AND gates are each connected to one input bit and have S_1S_0 as the other 2 inputs. One of the AND gates will always have S_1S_0 = 11, due to not gates connected to S_1S_0, so that its 3rd input bit will be able to pass through an OR gate to F which will make F equal to the value of the input. Multiplexers are used in every circuit of the ALU to control which bits are used as the outputs.