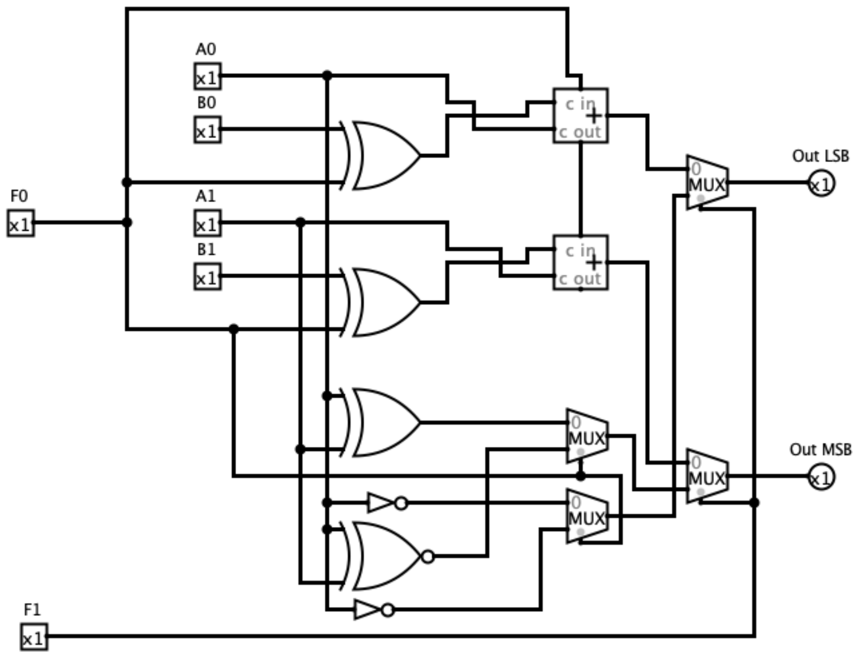
EEE3095/96S Practical 5

ARNJAM004 WLFJAD001

Addition, subtracting, incrementing and decrementing



In this circuit, there are a number of operations occurring. These are addition, subtraction, incrementing and decrementing. Because both of these operations are only dependent of F0 and F1, they can be grouped together with the use of a multiplexer in order to achieve the desired output. The two multiplexers on the furthest right of the circuit are responsible for switching between addition/subtracted and incrementing/decrementing. The other two multiplexers are for switching incrementing and decrementing. Because addition and subtraction are only dependent on F0, the switching between the two operations is done by changing the carry in (F0).

Incrementing

It is observed from a derived truth table that the MSB of the output when incrementing is simply A0 XOR A1 and the LSB on the output is NOT A0. From this, a logical circuit can be implemented.

Decrementing

From the derived truth table, it can be seen that the MSB of the output is simply A0 XNOR A1 and the LSB is also NOT A0. A similar circuit can be implemented to incrementing, the difference being the XOR is now inverted.

Multiplication and division

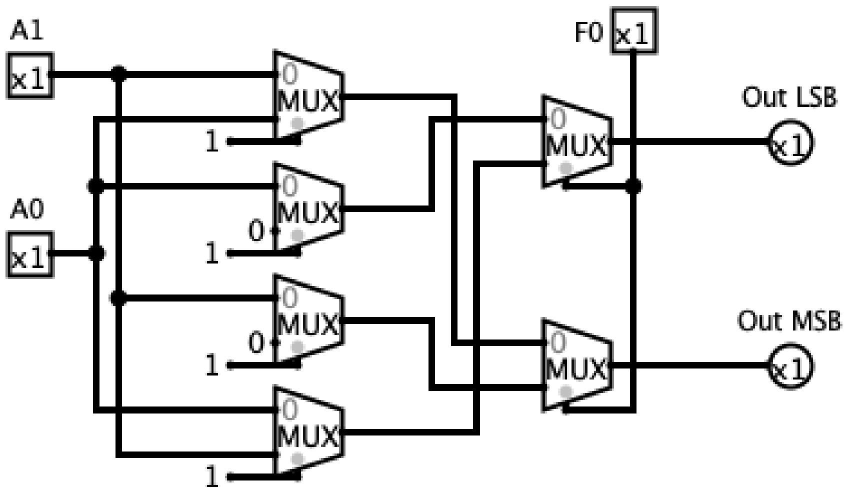
Multiply

It is observed that the multiplication of any number of bits by two is equivalent to bit shifting to the left by one.

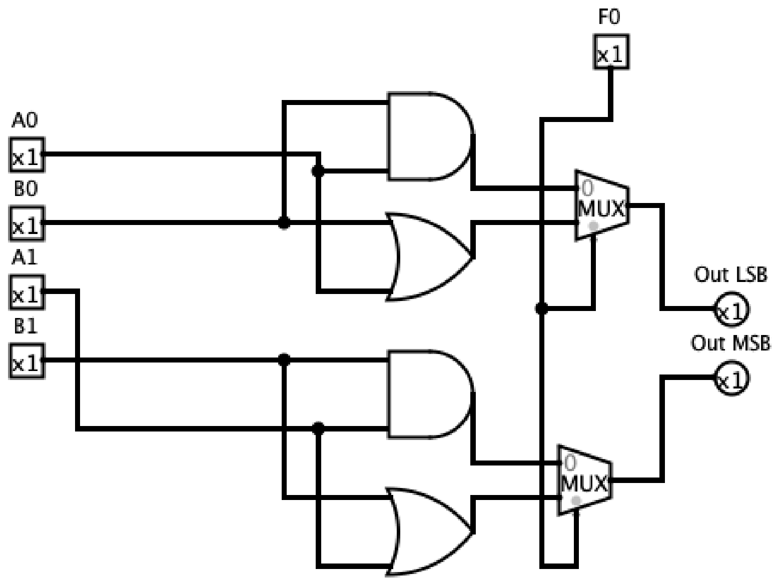
Divide

The same can be said for the case of division by two, except the bit shift will occur in the right direction.

This is implemented with the use of four multiplexers on the left most side of the circuit. The right most multiplexers are used for switching between multiply and divide operations.



Bitwise and & bitwise or



Bitwise operations

For these bitwise operations, it is noted that we are ANDing and ORing like bits of A and B. for example, A0 will be ANDed with B0 and the same with the OR operation. Two multiplexers are used in order to switch between AND and OR operations. The switching is only dependant on F0.

Bitwise and

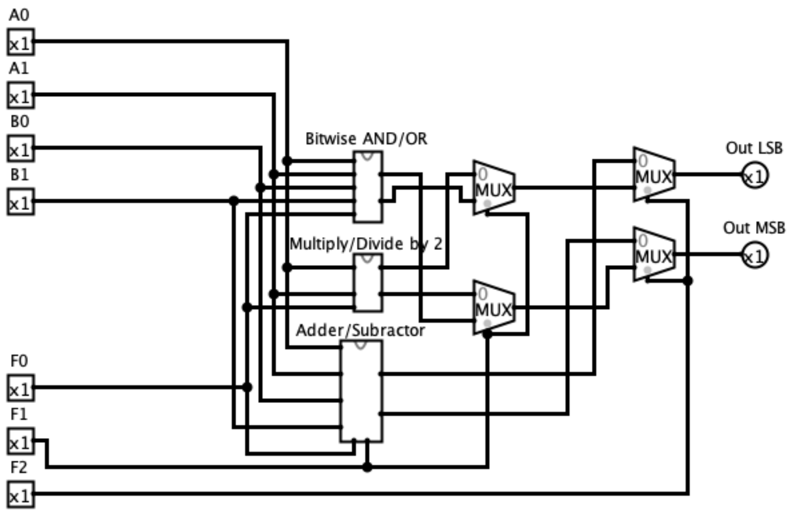
A0 will be ANDed with B0 and A1 will be ANDed with B1 to achieve the desired output

Bitwise or

A0 will be ORed with B0 and A1 will be ORed with B1 to achieve the desired output.

Full ALU

The various operations that were grouped together for spacing efficiency have been condensed into chips.


A0
x1
A1
x1
B0
x1
B1
x1
Bitwise AND/OR
Multiply/Divide by 2
Adder/Subractor
F0
x1
F1
x1
F2
x1
MUX
MUX
MUX
MUX
Out LSB
x1
Out MSB
x1