EE224: Digital System

ASSIGNMENT 1

Design a signed 8-bit (8 bit inputs and 16-bit output) booth encoded multiplier.

Components Used

1. Multipliexer-2 to 1 (**MUX)**: It takes two bit 8 bit inputs and depending on control pin (S) , gives out an 8 bit output.
2. Decoder: (**MUX3**): It takes the multiplicand and uses 3 control pins to determine a 9 bit output.
3. Leftshift(**Left_shift)**: It takes an 8 bit input and shifts once or doesn't depending on a control pin.
4. FullAdder(**fa16bit**): It takes two 16 bits operands and carry in inputs and performs addition/subtraction and gives out a 16bit output with carry-out.

Algorithm:

Current 2-bits	Multiplier for these	Previous MSBit	Pending Increment	Total Multiplier
00	0	0	0	0
01	+1	0	0	+1
10	-2	0	0	-2
11	-1	0	0	-1
00	0	1	+1	+1
01	+1	1	+1	+2
10	-2	1	+1	-1
11	-1	1	+1	0

Two 8 bit inputs, multiplicand and multiplier are given as input to the booth-multiplier entity. Now, the multiplier bits are scanned rightwise and depending on logic given in the above table, a 16 bit product is computed as output. The basic steps followed are:

(i) The rightmost two bits of multiplier are initially taken with value of key = '0'. This key is later used to hold MSB bit of previous pair. Thus, depending on the current two bits, the total mulitplier is selected , computed using multiplicand and added to the product (initially all bits set to zero) at rightmost bits.

(ii) The decoder takes the current 2 bits and key as the control input, multiplicand as the primary input. The total multiplier is first computed using boolean logic with certain encoding of the output multliplier magnitude (for eg, 0: 00, 1: 01, 2: 10). The sign of the multiplier is decided by the MSB of the current bit.

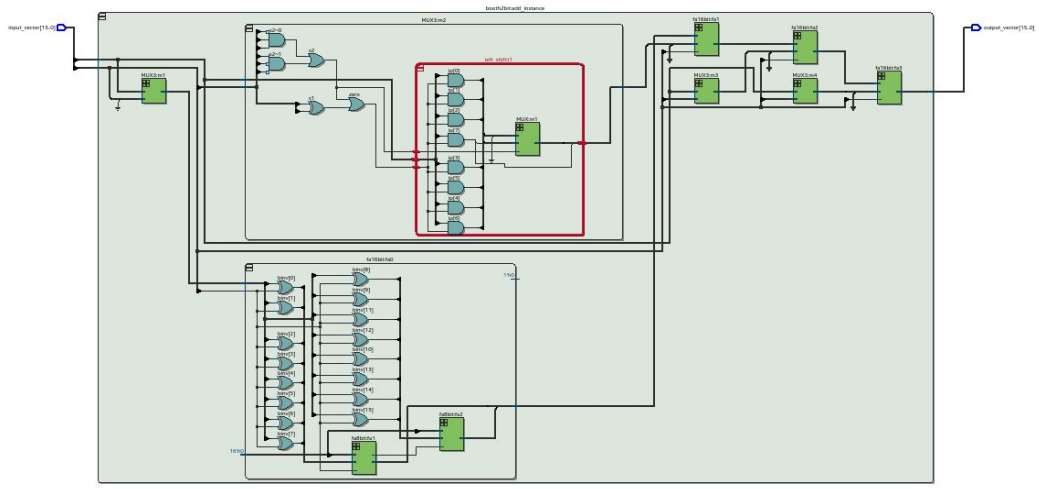
(iii) The output mulitplier magnitude is now used as control input to the left_shift block, the shifts once if input is '01', remains as it for '01' and all zero for '00'. The output is hence 9 bit. Incase shift operation isn't performed then the MSB of the output is same as that of the previous one (=0 if 8th bit is zero =1 otherwise for negative numbers).

(iv) The 9 bit output is then prefixed with all ones or zeroes depending on its signed bit and suffixed with zeros depending on relative shift required (i.e. initially no shift, in next iteration two zeros are suffixed as now for computing with next pair of multiplier bits are 2 bits ahead of previous ones).

(v) This 16 bit input is added/subtracted to the sum computed in previous iterations using a full adder. The operation of addition/subtraction is decided by the MSB of current bit taken as carry input to the adder.

(vi) The above process is repeated for next pair of bits with MSB of previous pair as a key and continued until the MSB of multiplier is reached.

Hardware:



The red-boxed is the left-shifter logic which is enclosed in another box: the decoder logic. The bottom block is the 16 bit full-adder (the logic gates are for inverting the second operand in case of subtraction and implemented as cascaded two 8 bit normal full adders).

Note: The image may not be so clear. So an image file of the above hardware logic named as '**hardware.png'** is also attached along with codes.

FILES:

1. **Booth2bit_v1.vhd:** booth multiplier architecture
2. **MUX.vhd:** multiplexer component
3. **MUX3.vhd:** decoder component
4. **Left_shift.vhd:** left-shift component
5. **Fa16bit.vhd:** 16 bit full adder
6. **Fa8bit.vhd:** 8 bit full adder
7. **Fa1bit:** 1 bit full adder
8. **DUT.vhd:** Device under test file for compilation
9. **Testbench.vhd:** Testbench
10. **Tracefile.txt:** All testcases written here
11. **OUTPUTS.txt:** The simulated output with errors if any is written here

12. **Booth2bit.qpf:** Project file
13. **Hardware.png:** hardware implementation
14. **Compilereport.png**
15. **Simulation.png:** simulated report

RESULTS:

The booth architecture was first compiled successfully. Then the testbench with all possible input combinations is added and simulated. The compilation and simulation reports are also attached with the codes.\\