Exercise 5 (Shifter/Rotator (HW2 Prob1)):

- Using Dataflow/RTL Verilog, implement a shifter/rotator capable of performing left shifts/rotates on a 16-bit input (**src[15:0]**). The shift/rotate amount can be anywhere from 0 to 15-bits, and is specified by a 4-bit input (**amt[3:0]**). The result goes to a signal called **res[15:0]**. You are **not** allowed to use the << or <<< operator. Write a testbench to verify its operation. This testbench does not need to be self checking, stimulus generation only (Please ensure reasonably comprehensive coverage with stimulus. i.e. cover multiple 16-bit quantities for both shift and rotate and for multiple shift amounts).

Signal Name:	Dir:	Description:
src[15:0]	in	Input vector to be left shifted/rotated
rotate	in	If rotate is asserted then the operation is a rotate not a shift
amt[3:0]	in	Specifies the amount of the shift 0 to 15 bits
res[15:0]	out	Result of the shift

Submit as much as you completed to the dropbox. You have to finish this as part of HW2

Exercise 5 (Hints on building a shifter):

This topology is referred to as a logarithmic shifter.

This is showing a logical right shift, it should be trivial to convert this to left instead. It is a bit trickier to think of how to make it shift or rotate, but think about another set of muxes at each level that do something a bit different with the LSB(s).

src[15:0]

1'b0   [15:1]

16

16

1   0   Shift by 1 or not

shft_stg1   16

2'b00   [15:2]

16

16

1   0   Shift by 2 or not

shft_stg2   16

4'b0000   [15:4]

16

16

1   0   Shift by 4 or not

shft_stg3   16

8'h00   [15:8]

16

16

1   0   Shift by 8 or not

res[15:0]   16

1000 0000 0000 0000
8   0   0   0

0000 0000 0000 1111

0000 0000 0011 1100
0   0   3   C

1010 1111 1111 1111

1111 1111 1111 1010

1000 0000 0000 0000

000 0000 0100 0000 0000