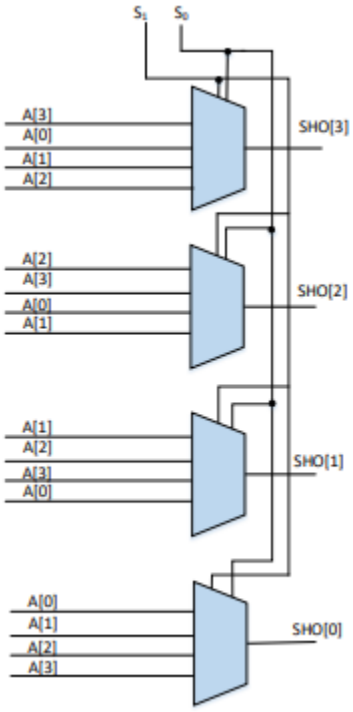
Ashaki Gumbs
Homework #2
03/09/2021

A. Write a 4-bit barrel shifter in SystemVerilog using multiplexers. This 4-bit barrel shifter can be generated with 4 multiplexers. Figure below shows the block diagram of this barrel shifter.




```
module MUX(
    input A,B,C,D,s0,s1,
    output reg SOUT
    );
always@(s0 or s1 or A or B or C
or D)begin
    case({s1, s0})
       2'b00: SOUT = A;
       2'b01: SOUT = B;
       2'b10: SOUT = C;
       2'b11: SOUT = D;
     endcase
     end
```

Figure 1 4-bit Barrel Shifter

First, I started with one, 2-bit select MUX. Then, I created the 4-bit barrel shifter in Figure 1 by instantiating 4 MUXs.

```
module Four_Bit_Barrel_Shifter(
    input s1,s0,
    input [6:0]A,
    output [3:0]S
    );
MUX M0(.s1(s1), .s0(s0), .SOUT(S[0]), .A(A[0]), .B(A[1]), .C(A[2]), .D(A[3]));
MUX M1(.s1(s1), .s0(s0), .SOUT(S[1]), .A(A[1]), .B(A[2]), .C(A[3]), .D(A[4]));
MUX M2(.s1(s1), .s0(s0), .SOUT(S[2]), .A(A[2]), .B(A[3]), .C(A[4]), .D(A[5]));
MUX M3(.s1(s1), .s0(s0), .SOUT(S[3]), .A(A[3]), .B(A[4]), .C(A[5]), .D(A[6]));


endmodule
```

Figure 2 Used to Shift the 1st 4 -Bits

Ashaki Gumbs
Homework #2
03/09/2021

```
module Four_Bit_Barrel_Shifter_More_Sig(
    input s3,s2,
    input [15:0]A,
    output [3:0]S
    );
MUX M0(.s1(s3), .s0(s2), .SOUT(S[0]), .A(A[0]), .B(A[1]), .C(A[2]), .D(A[3]));
MUX M1(.s1(s3), .s0(s2), .SOUT(S[1]), .A(A[4]), .B(A[5]), .C(A[6]), .D(A[7]));
MUX M2(.s1(s3), .s0(s2), .SOUT(S[2]), .A(A[8]), .B(A[9]), .C(A[10]), .D(A[11]));
MUX M3(.s1(s3), .s0(s2), .SOUT(S[3]), .A(A[12]), .B(A[13]), .C(A[14]), .D(A[15]));


endmodule
```

Ashaki Gumbs
Homework #2
03/09/2021

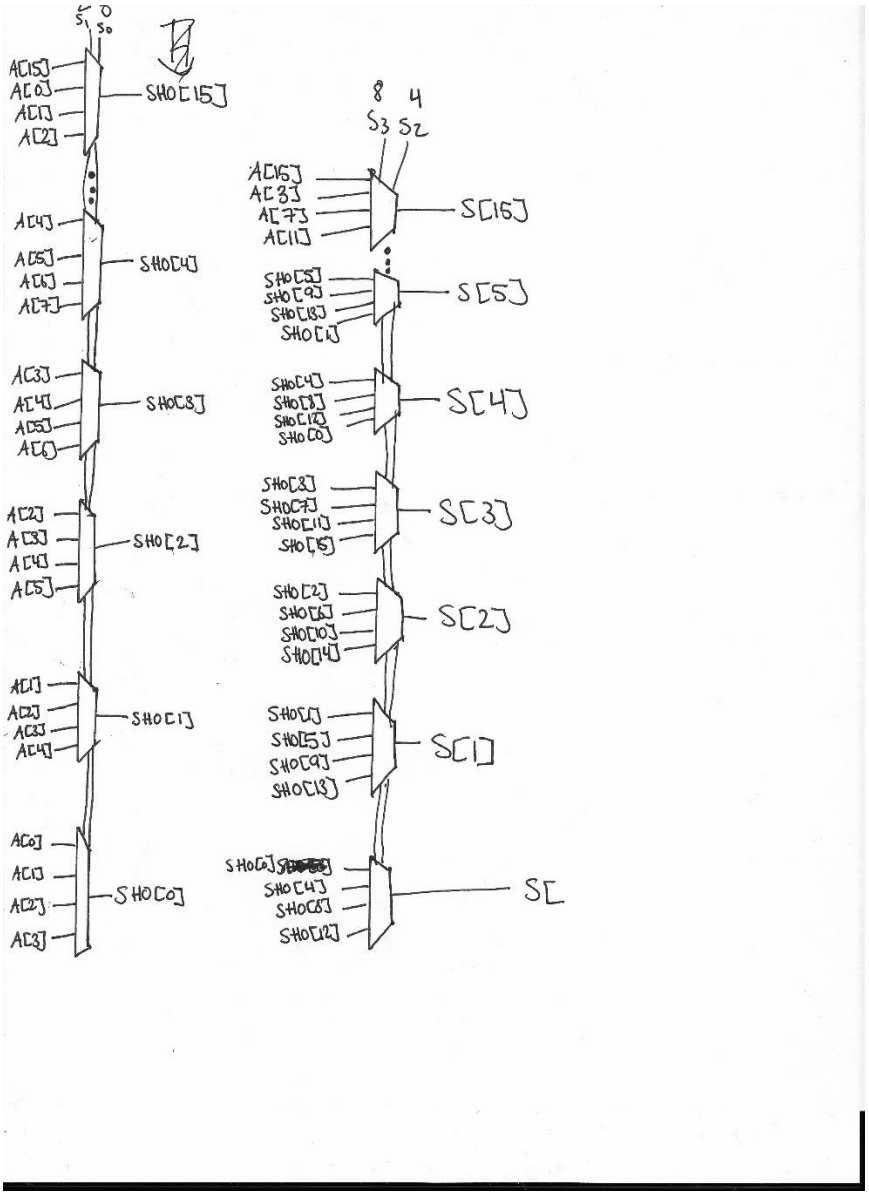
B. A 16-bit Barrel shifter can be generated using several instances of a 4-bit barrel shifter. Using this circuit write a structural description for a 16-bit barrel shifter circuit. You can use generate statement to build the iterative circuit.

```
module Top(
   input [15:0]A,
   input s3,s2,s1,s0,
   output [15:0]S
   );
wire [15:0] SHOtemp;
wire [6:0] ringaround = {A[2], A[1], A[0], A[15], A[14], A[13], A[12]};
wire [15:0] moresigin1 = {SHOtemp[15], SHOtemp[11], SHOtemp[7], SHOtemp[3], SHOtemp[14], SHOtemp[10], SHOtemp[6], SHOtemp[2],
   SHOtemp[13], SHOtemp[9], SHOtemp[5], SHOtemp[1], SHOtemp[12], SHOtemp[8], SHOtemp[4], SHOtemp[0]};
wire [15:0] moresigin2 = {SHOtemp[3], SHOtemp[15], SHOtemp[11], SHOtemp[7], SHOtemp[2], SHOtemp[14], SHOtemp[10], SHOtemp[6],
   SHOtemp[1], SHOtemp[13], SHOtemp[9], SHOtemp[5], SHOtemp[0], SHOtemp[12], SHOtemp[8], SHOtemp[4]};
wire [15:0] moresigin3 = {SHOtemp[7], SHOtemp[3], SHOtemp[15], SHOtemp[11], SHOtemp[6], SHOtemp[2], SHOtemp[14], SHOtemp[10],
   SHOtemp[5], SHOtemp[1], SHOtemp[13], SHOtemp[9], SHOtemp[4], SHOtemp[0], SHOtemp[12], SHOtemp[8]};
wire [15:0] moresigin4 = {SHOtemp[11], SHOtemp[7], SHOtemp[3], SHOtemp[15], SHOtemp[10], SHOtemp[6], SHOtemp[2], SHOtemp[14],
   SHOtemp[9], SHOtemp[5], SHOtemp[1], SHOtemp[13], SHOtemp[8], SHOtemp[4], SHOtemp[0], SHOtemp[12]};
Four_Bit_Barrel_Shifter U1(.s1(s1) ,.s0(s0),.S(SHOtemp[3:0]), .A(A[6:0]));
Four_Bit_Barrel_Shifter U2(.s1(s1) ,.s0(s0),.S(SHOtemp[7:4]), .A(A[10:4]));
Four_Bit_Barrel_Shifter U3(.s1(s1) ,.s0(s0),.S(SHOtemp[11:8]), .A(A[14:8]));
Four_Bit_Barrel_Shifter U4(.s1(s1) ,.s0(s0),.S(SHOtemp[15:12]), .A(ringaround));
Four_Bit_Barrel_Shifter_More_Sig U5(.s3(s3), .s2(s2), .S(S[3:0]), .A(moresigin1));
Four_Bit_Barrel_Shifter_More_Sig U6(.s3(s3), .s2(s2), .S(S[7:4]), .A(moresigin2));
Four_Bit_Barrel_Shifter_More_Sig U7(.s3(s3), .s2(s2), .S(S[11:8]), .A(moresigin3));
Four_Bit_Barrel_Shifter_More_Sig U8(.s3(s3), .s2(s2), .S(S[15:12]), .A(moresigin4));
endmodule
```

Ashaki Gumbs
Homework #2
03/09/2021

The 16 bit barrel shifter can be generated using 8 instances of the 4-bit barrel shifter. S0 and S1 controls the shifts by 0&2, respectively, while S3 and S2 controls the shifts by 4& 8. This results In 16 total shifts with MSB = S3, LSB = S0. My top module looks like such. Please note, the Four_Bit_Barrel_Shifter only needs 7 inputs, since inputs into the MUXs are reused in this module. This implementation isn't as straightforward when shifting S3 & S2, so 16-inputs were used.

C. Show the schematic diagram of the 16-bit barrel shifter and show the modules are connected to build circuit.



Ashaki Gumbs
Homework #2
03/09/2021

D. Write a testbench for the 16-bit barrel shifter in SystemVerilog that tests the correctness of your circuit and show the results in Modelsim

```
Top U2(.s1(s1) ,.s0(s0), .s2(s2), .s3(s3), .S(S[15:0]), .A(A[15:0]));
initial begin
A = 16'b0;
s0 = 0; s1 = 0; s2 = 0; s3 = 0;
#10
#10
A[6] = 1;
#10
s0 = 1; s1 = 0; s2 = 0; s3 = 0;
#10
s0 = 0; s1 = 1; s2 = 0; s3 = 0;
#10
s0 = 1; s1 = 1; s2 = 0; s3 = 0;
#10
s0 = 0; s1 = 0; s2 = 1; s3 = 0;
#10
s0 = 1; s1 = 0; s2 = 1; s3 = 0;
#10
s0 = 0; s1 = 1; s2 = 1; s3 = 0;
#10
s0 = 1; s1 = 1; s2 = 1; s3 = 0;
#10
s0 = 0; s1 = 0; s2 = 0; s3 = 1;
#10
s0 = 1; s1 = 0; s2 = 0; s3 = 1;
#10
s0 = 0; s1 = 1; s2 = 0; s3 = 1;
#10
s0 = 1; s1 = 1; s2 = 0; s3 = 1;
#10
s0 = 0; s1 = 0; s2 = 1; s3 = 1;
#10
s0 = 1; s1 = 0; s2 = 1; s3 = 1;
#10
s0 = 0; s1 = 1; s2 = 1; s3 = 1;
#10
s0 = 1; s1 = 1; s2 = 1; s3 = 1;
end
endmodule
```

Approach: test the 4-bit total possible combinations using S3,S2,S1,&S0.

Ashaki Gumbs
Homework #2
03/09/2021