perplexity

We are generating a 32-bit Barrel Shifter with Rotation Control hierarchically in Verilog. We have generated 8-bit Barrel Shifter, 32-bit Barrel Shifter defined as follows:

```
module barrel_shift_8(
input [7:0] inputData,
input [2:0] shiftVal,
output [7:0] outputData
);
```

```
wire [7:0] shifted_1;
wire [7:0] shifted_2;
wire [7:0] shifted_4;

assign shifted_1 = (shiftVal[0]) ? {inputData[6:0], inputData[7]} : inputData;
assign shifted_2 = (shiftVal[1]) ? {shifted_1[5:0], shifted_1[7:6]} : shifted_1;
assign shifted_4 = (shiftVal[2]) ? {shifted_2[3:0], shifted_2[7:4]} : shifted_2;

assign outputData = shifted_4;
```

```
endmodule
```

```
module barrel_shift_32(
input [31:0] inputData,
input [4:0] shiftVal,
output [31:0] outputData
);
```

```
wire [31:0] shifted_8;
wire [31:0] shifted_16;

wire [7:0] block0_in, block1_in, block2_in, block3_in;
wire [7:0] block0_out, block1_out, block2_out, block3_out;

assign block0_in = inputData[7:0];
assign block1_in = inputData[15:8];
assign block2_in = inputData[23:16];
assign block3_in = inputData[31:24];

barrel_shift_8 shift_block0 (
    .inputData(block0_in),
    .shiftVal(shiftVal[2:0]),
    .outputData(block0_out)
);

barrel_shift_8 shift_block1 (
    .inputData(block1_in),
    .shiftVal(shiftVal[2:0]),
    .outputData(block1_out)
);

barrel_shift_8 shift_block2 (
    .inputData(block2_in),
    .shiftVal(shiftVal[2:0]),
    .outputData(block2_out)
);

barrel_shift_8 shift_block3 (
    .inputData(block3_in),
    .shiftVal(shiftVal[2:0]),
    .outputData(block3_out)
);

assign shifted_8 = {block3_out, block2_out, block1_out, block0_out};

assign shifted_16 = (shiftVal[4]) ? {shifted_8[15:0], shifted_8[31:16]} : shifted_8;
assign outputData = (shiftVal[3]) ? {shifted_16[23:0], shifted_16[31:24]} : shifted_16;
```

endmodule

Please include the previous module(s) in your response and use them to hierarchically generate a 32-bit Barrel Shifter with Rotation Control defined as:
module barrel_shift_32_w_rotation_ctrl(sel_left_or_right_rotate, inputData[31:0], shiftVal[4:0], outputData[31:0]);
//Insert code here
endmodule

```
module barrel_shift_8(
    input  [7:0] inputData,
    input  [2:0] shiftVal,
    output [7:0] outputData
);

    wire [7:0] shifted_1;
    wire [7:0] shifted_2;
    wire [7:0] shifted_4;

    assign shifted_1 = (shiftVal[0]) ? {inputData[6:0], inputData[7]} : inputData;
    assign shifted_2 = (shiftVal[1]) ? {shifted_1[5:0], shifted_1[7:6]} : shifted_1;
    assign shifted_4 = (shiftVal[2]) ? {shifted_2[3:0], shifted_2[7:4]} : shifted_2;

    assign outputData = shifted_4;

endmodule

module barrel_shift_32(
    input  [31:0] inputData,
    input  [4:0] shiftVal,
    output [31:0] outputData
);

    wire [31:0] shifted_8;
    wire [31:0] shifted_16;

    wire [7:0] block0_in, block1_in, block2_in, block3_in;
    wire [7:0] block0_out, block1_out, block2_out, block3_out;

    assign block0_in = inputData[7:0];
    assign block1_in = inputData[15:8];
    assign block2_in = inputData[23:16];
    assign block3_in = inputData[31:24];

    barrel_shift_8 shift_block0 (
        .inputData(block0_in),
        .shiftVal(shiftVal[2:0]),
        .outputData(block0_out)
    );

    barrel_shift_8 shift_block1 (
        .inputData(block1_in),
        .shiftVal(shiftVal[2:0]),
        .outputData(block1_out)
    );

    barrel_shift_8 shift_block2 (
        .inputData(block2_in),
        .shiftVal(shiftVal[2:0]),
        .outputData(block2_out)
    );

    barrel_shift_8 shift_block3 (
        .inputData(block3_in),
        .shiftVal(shiftVal[2:0]),
        .outputData(block3_out)
    );

    assign shifted_8 = {block3_out, block2_out, block1_out, block0_out};

    assign shifted_16 = (shiftVal[4]) ? {shifted_8[15:0], shifted_8[31:16]} : shifted_8;
    assign outputData = (shiftVal[3]) ? {shifted_16[23:0], shifted_16[31:24]} : shifted_1

endmodule

module barrel_shift_32_w_rotation_ctrl(
    input sel_left_or_right_rotate,
    input [31:0] inputData,
    input [4:0] shiftVal,
    output [31:0] outputData
);

    wire [31:0] shifted_data;
    barrel_shift_32 shifter (
        .inputData(inputData),
        .shiftVal(shiftVal),
        .outputData(shifted_data)
    );

    assign outputData = sel_left_or_right_rotate ? shifted_data : {shifted_data[shiftVal:

endmodule
```