The testbench failed to compile. Please fix the module. The output of iverilog is as follows:

error: outputData is not a valid l-value in barrel_shift_8.
outputData is declared here as wire.outputData

The error message indicates that `outputData` is declared as a `wire`, and you're trying to assign a value to it inside a procedural block (`always` or `initial`). In Verilog, you cannot assign values to `wire` types inside procedural blocks. Instead, you should use `reg` for outputs that are assigned within procedural blocks. Here's how you can correct the module:

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

    // Use a case statement to simplify the logic
    always @(inputData, shiftVal) begin
        case (shiftVal)
            3'b000: outputData = inputData; // No shift
            3'b001: outputData = {inputData[6:0], inputData[^7]}; // Shift left by 1
            3'b010: outputData = {inputData[5:0], inputData[7:6]}; // Shift left by 2
            3'b011: outputData = {inputData[4:0], inputData[7:5]}; // Shift left by 3
            3'b100: outputData = {inputData[3:0], inputData[7:4]}; // Shift left by 4
            3'b101: outputData = {inputData[2:0], inputData[7:3]}; // Shift left by 5
            3'b110: outputData = {inputData[1:0], inputData[7:2]}; // Shift left by 6
            3'b111: outputData = {inputData[^0], inputData[7:1]}; // Shift left by 7
            default: outputData = inputData; // Default case, no shift
        endcase
    end

endmodule
```

Alternatively, if you prefer to keep `outputData` as a `wire`, you can use continuous assignment outside of any procedural block:

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

    // Use a case statement to simplify the logic
    assign outputData = (shiftVal == 3'b000) ? inputData :
```

```
                        (shiftVal == 3'b001) ? {inputData[6:0], inputData[^7]} :
                        (shiftVal == 3'b010) ? {inputData[5:0], inputData[7:6]} :
                        (shiftVal == 3'b011) ? {inputData[4:0], inputData[7:5]} :
                        (shiftVal == 3'b100) ? {inputData[3:0], inputData[7:4]} :
                        (shiftVal == 3'b101) ? {inputData[2:0], inputData[7:3]} :
                        (shiftVal == 3'b110) ? {inputData[1:0], inputData[7:2]} :
                        (shiftVal == 3'b111) ? {inputData[^0], inputData[7:1]} : inputDat

endmodule
```

Both versions should resolve the compilation error.

⁂