perplexity

Please generate a 5-to-32 decoder design in Verilog defined as:

module decoder5_32(in[4:0], out[31:0]);
//Insert code here
endmodule

```
module decoder5_32(
    input [4:0] in,
    output [31:0] out
);

    assign out = (1 << in);

endmodule
```

However, the above code might not work as expected due to the nature of the expression `(1 << in)`, which is not directly supported in Verilog for generating a one-hot output. Here's a more accurate implementation using a case statement:

```
module decoder5_32(
    input [4:0] in,
    output [31:0] out
);

    always @(*) begin
        case(in)
            5'b00000: out = 32'b00000000000000000000000000000001;
            5'b00001: out = 32'b00000000000000000000000000000010;
            5'b00010: out = 32'b00000000000000000000000000000100;
            5'b00011: out = 32'b00000000000000000000000000001000;
            5'b00100: out = 32'b00000000000000000000000000010000;
            5'b00101: out = 32'b00000000000000000000000000100000;
            5'b00110: out = 32'b00000000000000000000000001000000;
            5'b00111: out = 32'b00000000000000000000000010000000;
            5'b01000: out = 32'b00000000000000000000000100000000;
            5'b01001: out = 32'b00000000000000000000001000000000;
            5'b01010: out = 32'b00000000000000000000010000000000;
            5'b01011: out = 32'b00000000000000000000100000000000;
            5'b01100: out = 32'b00000000000000000001000000000000;
            5'b01101: out = 32'b00000000000000000010000000000000;
            5'b01110: out = 32'b00000000000000000100000000000000;
            5'b01111: out = 32'b00000000000000001000000000000000;
            5'b10000: out = 32'b00000000000000010000000000000000;
            5'b10001: out = 32'b00000000000000100000000000000000;
```

```
            5'b10010: out = 32'b00000000000001000000000000000000;
            5'b10011: out = 32'b00000000000010000000000000000000;
            5'b10100: out = 32'b00000000000100000000000000000000;
            5'b10101: out = 32'b00000000001000000000000000000000;
            5'b10110: out = 32'b00000000010000000000000000000000;
            5'b10111: out = 32'b00000000100000000000000000000000;
            5'b11000: out = 32'b00000001000000000000000000000000;
            5'b11001: out = 32'b00000010000000000000000000000000;
            5'b11010: out = 32'b00000100000000000000000000000000;
            5'b11011: out = 32'b00001000000000000000000000000000;
            5'b11100: out = 32'b00010000000000000000000000000000;
            5'b11101: out = 32'b00100000000000000000000000000000;
            5'b11110: out = 32'b01000000000000000000000000000000;
            5'b11111: out = 32'b10000000000000000000000000000000;
            default: out = 32'b0;
        endcase
    end

endmodule
```

Alternatively, you can use a more concise approach with a loop:

```
module decoder5_32(
    input [4:0] in,
    output [31:0] out
);

    always @(*) begin
        out = 32'b0;
        if(in < 32) begin
            out[in] = 1'b1;
        end
    end

endmodule
```

This version is simpler and more efficient, as it directly sets the output bit corresponding to the input value to 1, while all other bits are set to 0.