

```

module Clock (
    input clk,
    input rst,
    output [2:0] hour,
    output [3:0] min,
    output [4:0] sec
);
    wire sec_carry, min_carry;
    Sec sec1(clk, rst, sec, sec_carry);
    Min min1(clk, rst, sec_carry, min, min_carry);
    Hour hour1(clk, rst, min_carry, hour);

```

```
endmodule
```

```

module Sec(
    input clk,
    input rst1,
    output reg [4:0] sec,
    output reg sec_carry
);
    always @(posedge clk or posedge rst1) begin
        sec_carry = 0;
        if(rst1 == 1) begin
            sec = 0;
        end
        else begin
            sec = sec + 1;
            if(sec == 20) begin
                sec = 0;
                sec_carry = 1;
            end
        end
    end
end
endmodule

```

```

module Min (
    input clk,
    input rst1,
    input sec_carry,
    output reg [3:0] min,
    output reg min_carry
);
    always @(posedge clk or posedge rst1) begin
        min_carry = 0;
        if(rst1 == 1) begin
            min = 0;
        end
        else begin
            if(sec_carry == 1) begin
                min = min + 1;
            end
            if(min == 10) begin
                min = 0;
                min_carry = 1;
            end
        end
    end
end
endmodule

```

```

module Hour (
    input clk,
    input rst1,
    input min_carry,
    output reg [2:0] hour
);
    always @(posedge clk or posedge rst1) begin
        if(rst1 == 1) begin
            hour = 0;
        end
        else begin
            if(min_carry == 1) begin
                hour = hour + 1;
            end
            if(hour == 5) begin
                hour = 0;
            end
        end
    end
end
endmodule

```

[illegible]