```
`timescale 1ns / 1ps
/**********************************************
**********************************
*
* Module: Seven_segment
*
* Author: Eric Christie
* Class: ECEN 220, Section 1, Winter 2021
* Date: 2/23/21
*
* Description: Provides the segment-level
logic for the anode and cathodes to
display the right segments
                    for the right inputs.
*
*
**********************************************
**********************************/


module seven_segment(
    input wire logic [3:0] data,
    output logic [6:0] segment);
```

```
    //internal wires
    logic CA, CB, CC, CD, CE, CF, CG;
    logic notD3, notD2, notD1, notD0;
//inverters for each data signal
    logic [3:0] A; //and gates from CA
    logic D1;
    logic [2:0] G;

    not(notD3, data[3]); //inverters
    not(notD2, data[2]);
    not(notD1, data[1]);
    not(notD0, data[0]);

    //CA - structural, gate-level,
sum-of-product
    and(A[0], notD3, notD2, notD1,
data[0]); //0001
    and(A[1], notD3, data[2], notD1,
notD0); //0100
    and(A[2], data[3], notD2, data[1],
data[0]); //1011
    and(A[3], data[3], data[2], notD1,
data[0]); //1101
    or(CA, A[0], A[1], A[2], A[3]);
```

```
    //CB - Dataflow, assign statement with
?:
    assign CB = (data == 4'h5)?1:
                (data == 4'h6)?1:
                (data == 4'hB)?1:
                (data == 4'hC)?1:
                (data == 4'hE)?1:
                (data == 4'hF)?1:
                0; //false if not one of
the above cases


    //CC - Dataflow, dataflow operators
    assign CC = (data == 4'b0010) || (data
>= 4'hC && data != 4'hD);

    //CD - Dataflow, operators
    assign CD = (data == 4'b0001) || (data
== 4'b0100) || (data == 4'b0111) || (data
== 4'b1010) || (data == 4'b1111);

    //CE - Dataflow, operators
    assign CE = ((data >= 4'd1) && (data
```

```
<= 4'd9) && (data != 4'd2) && (data !=
4'd6) && (data != 4'd8));

    //CF
    assign CF = (data == 4'h1)?1:
                (data == 4'h2)?1:
                (data == 4'h3)?1:
                (data == 4'h7)?1:
                (data == 4'hD)?1:
                0;

    //CG - structural, gate-level,
minimized equation (0 and 1)
    and(G[0], notD3, notD2, notD1); //0000
and 0001
    and(G[1], notD3, data[2], data[1],
data[0]);
    and(G[2], data[3], data[2], notD1,
notD0);
    or(CG, G[0], G[1], G[2]);


    //OUTPUT - cathodes are concatenated
into segment
```

```
    assign segment = {CG, CF, CE, CD, CC,
CB, CA};
endmodule
```