module ARM(I, A, B, C, D, E, F, G, DP);

input wire [7:0] I /\* synthesis loc="2,3,4,5,6,7,8,9"\*/;

output wire A; /\* synthesis loc="14" \*/

output wire B; /\* synthesis loc="15" \*/

output wire C; /\* synthesis loc="16" \*/

output wire D; /\* synthesis loc="17" \*/

output wire E; /\* synthesis loc="18" \*/

output wire F; /\* synthesis loc="19" \*/

output wire G; /\* synthesis loc="20" \*/

output wire DP; /\* synthesis loc="21" \*/

reg [0:6] SEG7;

always @ (I) begin

casez (I)

8'b10000000: SEG7 = 7'b1110111; // A

8'b11000000: SEG7 = 7'b0110000; // 1

8'b1?100000: SEG7 = 7'b1101101; // 2

8'b1??10000: SEG7 = 7'b1111001; // 3

8'b1???1000: SEG7 = 7'b0110011; // 4

8'b1????100: SEG7 = 7'b1011011; // 5

8'b1?????10: SEG7 = 7'b1011111; // 6

8'b1??????1: SEG7 = 7'b1110000; // 7

8'b0???????: SEG7 = 7'b0111110; // U

endcase

end

assign {A,B,C,D,E,F,G} = ~SEG7;

endmodule

/\* Hexadecimal 7-Segment Decoder for 22V10 \*/

module ALPHA(I, A, B, C, D, E, F, G, DP);

input wire [0:4] I /\* synthesis loc="2,3,4,5,9"\*/;

output wire A; /\* synthesis loc="14" \*/

output wire B; /\* synthesis loc="15" \*/

output wire C; /\* synthesis loc="16" \*/

output wire D; /\* synthesis loc="17" \*/

output wire E; /\* synthesis loc="18" \*/

output wire F; /\* synthesis loc="19" \*/

output wire G; /\* synthesis loc="20" \*/

output wire DP; /\* synthesis loc="21" \*/

reg [6:0] SEG7;

always @ (I) begin

case (I)

5'b00000: SEG7 = 7'b1110111; // A

5'b00010: SEG7 = 7'b0011111; // b

5'b00100: SEG7 = 7'b1001110; // C

5'b00110: SEG7 = 7'b0111101; // d

5'b01000: SEG7 = 7'b1001111; // E

5'b01010: SEG7 = 7'b1000111; // F

5'b01100: SEG7 = 7'b1111011; // g

5'b01110: SEG7 = 7'b0110111; // H

5'b10000: SEG7 = 7'b0111100; // J

5'b10010: SEG7 = 7'b0001110; // L

5'b10100: SEG7 = 7'b0010101; // n

5'b10110: SEG7 = 7'b0011101; // o

5'b11000: SEG7 = 7'b1100111; // P

5'b11010: SEG7 = 7'b0000101; // r

5'b11100: SEG7 = 7'b0111110; // U

5'b11110: SEG7 = 7'b0111011; // y

5'b00001: SEG7 = 7'b1111110; // 0

5'b00011: SEG7 = 7'b0110000; // 1

5'b00101: SEG7 = 7'b1101101; // 2

5'b00111: SEG7 = 7'b1111001; // 3

5'b01001: SEG7 = 7'b0110011; // 4

5'b01011: SEG7 = 7'b1011011; // 5

5'b01101: SEG7 = 7'b1011111; // 6

5'b01111: SEG7 = 7'b1110000; // 7

5'b10001: SEG7 = 7'b1111111; // 8

5'b10011: SEG7 = 7'b1111011; // 9

5'b10101: SEG7 = 7'b1110111; // A

5'b10111: SEG7 = 7'b0011111; // B

5'b11001: SEG7 = 7'b1001110; // C

5'b11011: SEG7 = 7'b0111101; // D

5'b11101: SEG7 = 7'b1001111; // E

5'b11111: SEG7 = 7'b1000111; // F

endcase

end

assign {A,B,C,D,E,F,G} = ~SEG7;

endmodule

/\* Hexadecimal 7-Segment Decoder for 22V10 \*/

module LAB7(I, A, B, C, D, E, F, G, DP);

input wire [0:7] I /\* synthesis loc="2,3,4,5,6,7,8,9" \*/;

output wire A; /\* synthesis loc="14" \*/

output wire B; /\* synthesis loc="15" \*/

output wire C; /\* synthesis loc="16" \*/

output wire D; /\* synthesis loc="17" \*/

output wire E; /\* synthesis loc="18" \*/

output wire F; /\* synthesis loc="19" \*/

output wire G; /\* synthesis loc="20" \*/

output wire DP; /\* synthesis loc="21" \*/

assign DP = ~I[7];

assign G = ~I[6];

assign F = ~I[5];

assign E = ~I[4];

assign D = ~I[3];

assign C = ~I[2];

assign B = ~I[1];

assign A = ~I[0];

endmodule