```
`timescale 1ns / 1ps
/********************************************************************** *
* Module: Codebreaker
*
* Author: Noah Hanks
* Class: ECEN 220, Section 3, Fall 2020 - ECEN 220, Section 1, Winter 2020 * Date:
18 Nov 2020
*
* Description: Breaks the given code into an unreadable format.
*
*
**********************************************************************/
`default_nettype none


module Codebreaker(
    input wire logic clk, reset, start, done_drawing_plaintext,
    output logic[15:0] key_display,
    output logic stopwatch_run, draw_plaintext,
    output logic[127:0] plaintext_to_draw
    );

    typedef enum logic[2:0] {HOLD, DECRYPT, READABLE, DISPLAY, FINISH, ERR = 'X}
StateType;
    StateType cs, ns;
    logic enable, done;
    logic[23:0] key;
    logic[127:0] cyphertext;
    logic plaintext_is_ascii;
    assign plaintext_is_ascii = ((plaintext_to_draw[127:120] >= "A" &&
plaintext_to_draw[127:120] <= "Z") || (plaintext_to_draw[127:120] >= "0" &&
plaintext_to_draw[127:120] <= "9") || (plaintext_to_draw[127:120] == " ")) &&
                                ((plaintext_to_draw[119:112] >= "A" &&
plaintext_to_draw[119:112] <= "Z") || (plaintext_to_draw[119:112] >= "0" &&
plaintext_to_draw[119:112] <= "9") || (plaintext_to_draw[119:112] == " ")) &&
                                ((plaintext_to_draw[111:104] >= "A" &&
plaintext_to_draw[111:104] <= "Z") || (plaintext_to_draw[111:104] >= "0" &&
plaintext_to_draw[111:104] <= "9") || (plaintext_to_draw[111:104] == " ")) &&
                                ((plaintext_to_draw[103:96] >= "A" &&
plaintext_to_draw[103:96] <= "Z") || (plaintext_to_draw[103:96] >= "0" &&
plaintext_to_draw[103:96] <= "9") || (plaintext_to_draw[103:96] == " ")) &&
                                ((plaintext_to_draw[95:88] >= "A" &&
plaintext_to_draw[95:88] <= "Z") || (plaintext_to_draw[95:88] >= "0" &&
plaintext_to_draw[95:88] <= "9") || (plaintext_to_draw[95:88] == " ")) &&
                                ((plaintext_to_draw[87:80] >= "A" &&
plaintext_to_draw[87:80] <= "Z") || (plaintext_to_draw[87:80] >= "0" &&
```

```
plaintext_to_draw[87:80] <= "9") || (plaintext_to_draw[87:80] == " ")) &&
                                ((plaintext_to_draw[79:72] >= "A" &&
plaintext_to_draw[79:72] <= "Z") || (plaintext_to_draw[79:72] >= "0" &&
plaintext_to_draw[79:72] <= "9") || (plaintext_to_draw[79:72] == " ")) &&
                                ((plaintext_to_draw[71:64] >= "A" &&
plaintext_to_draw[71:64] <= "Z") || (plaintext_to_draw[71:64] >= "0" &&
plaintext_to_draw[71:64] <= "9") || (plaintext_to_draw[71:64] == " ")) &&
                                ((plaintext_to_draw[63:56] >= "A" &&
plaintext_to_draw[63:56] <= "Z") || (plaintext_to_draw[63:56] >= "0" &&
plaintext_to_draw[63:56] <= "9") || (plaintext_to_draw[63:56] == " ")) &&
                                ((plaintext_to_draw[55:48] >= "A" &&
plaintext_to_draw[55:48] <= "Z") || (plaintext_to_draw[55:48] >= "0" &&
plaintext_to_draw[55:48] <= "9") || (plaintext_to_draw[55:48] == " ")) &&
                                ((plaintext_to_draw[47:40] >= "A" &&
plaintext_to_draw[47:40] <= "Z") || (plaintext_to_draw[47:40] >= "0" &&
plaintext_to_draw[47:40] <= "9") || (plaintext_to_draw[47:40] == " ")) &&
                                ((plaintext_to_draw[39:32] >= "A" &&
plaintext_to_draw[39:32] <= "Z") || (plaintext_to_draw[39:32] >= "0" &&
plaintext_to_draw[39:32] <= "9") || (plaintext_to_draw[39:32] == " ")) &&
                                ((plaintext_to_draw[31:24] >= "A" &&
plaintext_to_draw[31:24] <= "Z") || (plaintext_to_draw[31:24] >= "0" &&
plaintext_to_draw[31:24] <= "9") || (plaintext_to_draw[31:24] == " ")) &&
                                ((plaintext_to_draw[23:16] >= "A" &&
plaintext_to_draw[23:16] <= "Z") || (plaintext_to_draw[23:16] >= "0" &&
plaintext_to_draw[23:16] <= "9") || (plaintext_to_draw[23:16] == " ")) &&
                                ((plaintext_to_draw[15:8] >= "A" &&
plaintext_to_draw[15:8] <= "Z") || (plaintext_to_draw[15:8] >= "0" &&
plaintext_to_draw[15:8] <= "9") || (plaintext_to_draw[15:8] == " ")) &&
                                ((plaintext_to_draw[7:0] >= "A" &&
plaintext_to_draw[7:0] <= "Z") || (plaintext_to_draw[7:0] >= "0" &&
plaintext_to_draw[7:0] <= "9") || (plaintext_to_draw[7:0] == " "));
    logic inc_key, key_reset;

    //assign key = 24'h79726a;
    assign cyphertext = 128'h189f2800aac06ce4a74292bffe33fd2c;
    assign key_display = key[23:8];

    decrypt_rc4 Drc4(
    .clk(clk),
    .reset(reset),
    .enable(enable),
    .key(key),
    .bytes_in(cyphertext),
    .bytes_out(plaintext_to_draw),
    .done(done));
```

```
//begins state machine for codebreaker
always_comb begin
    ns = ERR;
    enable = 0;
    stopwatch_run = 0;
    draw_plaintext = 0;
    inc_key = 0;
    key_reset = 0;
    if(reset)
        ns = HOLD;
    else
        case(cs)
        HOLD: begin
            if(start) begin
                key_reset = 1;
                ns = DECRYPT;
                end
            else
                ns = HOLD;
            end
        DECRYPT: begin
            stopwatch_run = 1;
            enable = 1;
            if(done)
                ns = READABLE;
            else
                ns = DECRYPT;
            end
        READABLE: begin
            stopwatch_run = 1;
            if(plaintext_is_ascii)
                ns = DISPLAY;
            else begin
                inc_key = 1;
                ns = DECRYPT;
                end
            end
        DISPLAY: begin
            draw_plaintext = 1;
            if(done_drawing_plaintext)
                ns = FINISH;
            else
                ns = DISPLAY;
            end
        FINISH:
            ns = FINISH;
```

```
            endcase
        end

    //assigns current state to next state
    always_ff @(posedge clk) begin
        cs <= ns;
        if(key_reset)
            key <= 0;
        else if(inc_key)
            key <= key + 1;
        end



    //assign key_display = 0;
    //assign stopwatch_run = 1;
    //assign plaintext_to_draw = {"I LOVE JAYDENNNN"};
    //assign draw_plaintext = start;



endmodule
```