UCSD CSE140L Spring 2014

LAB#3 Report

Demonstration Date: / /14 Student CID								
Student Name:first				M.I.	Last			
TED Submission Date & Time :								
	(FILLED BY Student BEFORE DEMO)			(*** FILLED BY TUTOR/INSTRUCTOR ***)				
	Self-test Report			Demo Reviewer Name :				
	Working	Not work	king	Demo score	Report score			
Part1:			_	/3	a)/ 1			
Part2:			_	/3	b)/ 1			
Part3:			_	/3	c)/ 2			
Part4:			_	/3	d)/ 1			
Part5:			_	/3 Subtotal	Subtotal			
				/15	/5			

TOTAL Score: ___

a) Description:

- a. In the precondition, I have everything I use in all the other parts being initialized and set all the lights to be off. I have my CID displaying.
- b. In part one, I used key[1], sw[2:0] and key[1] was for entering money. The switches are for nickel, dime, and quarter input.
- c. In part two, I used sw[9] which is a counter for the number of times coin inputs reach 35 cents.
- d. In part three, I used sw[8] & sw[4] for credit card and reset inputs.
- e. In part four, I used sw[3] for one-dollar inputs.
- f. In part five, I used all HEX's to display "Err" when an error has occurred.

b) Verilog code

```
`d`define BLANK 7'b1111111
                                                                        integer cycle = 0;
define ZERO 7'b1000000
                                                                        integer rep = 0;
define ONE 7'b1111001
                                                                        integer flag = 0;
define TWO 7'b0100100
                                                                        integer clear = 0:
define THREE 7'b0110000
define FOUR 7'b0011001
define FIVE 7'b0010010
                                                                        always @(posedge clock) begin
define SIX 7'b0000010
                                                                        state = next state:
define SEVEN 7'b1111000
                                                                        if (flag == 1) begin
define EIGHT 7'b0000000
                                                                        h3 = THRÉE;
define NINE 7'b0011000
                                                                        h2 = FIVE;
define A 7'b0001000
                                                                        clear = 1;
define b 7'b0000011
                                                                        end
define C 7'b1000110
                                                                        if (sw[9] == 1 \&\& err == 0 \&\& rep == 0) begin
define d 7'b0100001
                                                                        rep <= 1;
define E 7'b0000110
                                                                        h3 \le hex3;
define F 7'b0001110
                                                                        h2 <= hex2;
define H 7'b0001001
                                                                        h1 <= hex1;
define L 7'b1000111
                                                                        h0 \le hex0:
define r 7'b0101111
                                                                        hex3 <= `BLANK;
                                                                        hex2 <= `BLANK;
module L3C580
                                            // where 580
                                                                        hex1 <= `BLANK;
= CID
                                                                        case (counter)
                                                                        0: begin
input [9:0]sw,
                        // ten up-down switches, SW9 -
                                                                        hex0 <= `ZERO;
SW<sub>0</sub>
                                                                        end
input [3:0]key,
                        // four pushbutton swithes, KEY3 -
                                                                        1: begin
KEY0
                                                                        hex0 <= `ONE;
input clock,
                        // 24MHz clock source on Altera
                                                                        end
DE1 board
                                                                        2: begin
                        // ten Red LEDs. LEDR9 - LEDR0
output reg [9:0]ledr,
                                                                        hex0 <= `TWO;
output reg [7:0]ledg,
                        // eight Green LEDs, LEDG8 -
                                                                        end
LEDG0
                                                                        3: begin
output reg [6:0]hex3,hex2,hex1,hex0
                                            // four 7-
                                                                        hex0 <= `THREE;
segment, HEX3 - HEX0
                                                                        end
                                                                        4: begin
parameter zero = 3'b000, one = 3'b001, two = 3'b010, three
                                                                        hex0 <= `FOUR;
= 3'b011, four = 3'b100, five = 3'b101, six = 3'b110, seven =
                                                                        end
                                                                        5: begin
parameter none = 9'b000000000, nickel = 9'b000000001,
                                                                        hex0 <= `FIVE;
dime = 9'b000000010, quarter = 9'b000000100, dollar =
                                                                        end
9'b000001000, credit_card = 9'b000010000, reset =
                                                                        6: begin
9'b100000000:
                                                                        hex0 <= `SIX;
integer in;
                                                                        end
                                                                        7: begin
reg[3:0] state, next_state;
                                                                        hex0 <= `SEVEN;
                                                                        end
reg [6:0] h3, h2, h1, h0;
                                                                        8: begin
                                                                        hex0 <= `EIGHT;
integer VM = 0;
                                                                        end
reg[3:0] counter = 0;
                                                                        9: begin
integer err = 0;
                                                                        hex0 <= `NINE;
```

```
end
                                                                            hex3 <= `TWO;
                                                                            hex2 <= `ZERO;
10: begin
                                                                            hex1 <= `ZERO;
hex0 <= A;
                                                                            hex0 <= `ZERO;
end
11: begin
                                                                            h3 <= hex3;
hex0 <= b;
                                                                            h2 <= hex2;
                                                                            h1 <= hex1;
end
12: begin
                                                                            h0 \le hex0:
hex0 <= C;
                                                                            ledg[7:0] <= 0;
                                                                            ledr[9:0] <= 0;
end
13: begin
                                                                            end
                                                                            five: begin
hex0 \le d:
                                                                            hex3 <= `TWO;
hex2 <= `FIVE;
end
14: begin
                                                                            hex1 <= `ZERO;
hex0 \stackrel{\sim}{=} E;
                                                                            hex0 <= `ZERO;
end
15: begin
                                                                            h3 <= hex3;
hex0 <= F:
                                                                            h2 <= hex2;
end
                                                                            h1 <= hex1;
endcase
                                                                            h0 \le hex0;
                                                                            ledg[7:0] <= 0;
end
else if (VM == 1 && sw[9] == 0) begin
                                                                            ledr[9:0] <= 0;
                                                                            end
rep = 0;
if (err == 0) begin
                                                                            six: begin
                                                                            hex3 <= `THREE;
hex2 <= `ZERO;
case(state)
zero: begin
hex3 <= \ZERO;
                                                                            hex1 <= `ZERO;
hex2 <= `ZERO;
                                                                            hex0 <= `ZERO;
hex1 <= `ZERO;
hex0 <= `ZERO;
                                                                            h3 <= hex3;
                                                                            h2 <= hex2;
h3 <= hex3;
                                                                            h1 \le hex1;
h2 <= hex2;
                                                                            h0 \le hex0;
h1 <= hex1;
                                                                            ledg[7:0] <= 0;
h0 \le hex0;
                                                                            ledr[9:0] <= 0;
                                                                            end
cycle = 0;
ledg[7:0] <= 0;
                                                                            seven: begin
ledr[9:0] <= 0;
                                                                            cycle <= cycle + 1;
end
                                                                            case (in)
one: begin
                                                                            none: begin
hex3 <= `ZERO;
                                                                            hex3 <= h3;
hex2 <= `FIVE;
                                                                            hex2 <= h2;
hex1 <= `ZERO;
                                                                            hex1 <= h1;
hex0 <= `ZERO;
                                                                            hex0 \le h0;
h3 <= hex3;
                                                                            end
h2 <= hex2;
                                                                            nickel: begin
                                                                            hex3 <= `THREE;
h1 \le hex1;
                                                                            hex2 <= `FIVE;
h0 \le hex0;
                                                                            hex1 <= `ZERO;
ledg[7:0] <= 0;
ledr[9:0] <= 0;
                                                                            hex0 <= `ZERO;
                                                                            h3 <= hex3;
end
two: begin
                                                                            h2 <= hex2;
hex3 <= `ONE;
hex2 <= `ZERO;
                                                                            h1 \le hex1;
                                                                            h0 \le hex0;
hex1 <= `ZERO;
                                                                            end
hex0 <= `ZERO;
                                                                            dime: begin
                                                                           if (h3 == `THREE && h2 == `ZERO) begin
hex3 <= `THREE;
hex2 <= `FIVE;
h3 <= hex3;
h2 <= hex2;
h1 <= hex1;
h0 \le hex0;
                                                                            hex1 <= `ZERO;
ledg[7:0] <= 0;
                                                                            hex0 <= `FIVE;
ledr[9:0] <= 0;
                                                                            h3 <= hex3;
                                                                            h2 <= hex2;
end
three: begin
                                                                            h1 \le hex1;
hex3 <= `ONE;
                                                                            h0 \le hex0:
hex2 <= `FIVE;
                                                                            end
hex1 <= `ZERO;
                                                                            else if (h3 == `TWO && h2 == `FIVE) begin
                                                                            hex3 <= `THREE;
hex0 <= `ZERO;
                                                                            hex2 <= `FIVE;
h3 <= hex3;
h2 <= hex2;
                                                                            hex1 <= `ZERO;
h1 <= hex1;
                                                                            hex0 <= `ZERO;
h0 \le hex0;
                                                                            end
ledg[7:0] <= 0;
                                                                            else begin
ledr[9:0] <= 0;
                                                                            hex3 <= h3;
end
                                                                            hex2 \le h2;
four: begin
                                                                            hex1 \le h1;
```

```
hex0 \le h0;
                                                                         hex1 <= `NINE;
                                                                         hex0 <= `ZERO:
end
end
                                                                         end
quarter: begin
                                                                         else if (h3 == `THREE && h2 == `ZERO) begin
if (h3 == `ONE && h2 == `ZERO) begin
                                                                         hex3 <= `THREE;
                                                                         hex2 <= `FIVE;
hex1 <= `NINE;
hex3 <= `THREE;
hex2 <= `FIVE;
hex1 <= `ZERO:
                                                                         hex0 <= `FIVE:
hex0 <= `ZERO;
                                                                         end
end
                                                                         else if (h3 == `THREE && h2 == `FIVE) begin
                                                                         hex3 <= `THREE;
else if (h3 == `ONE && h2 == `FIVE) begin
                                                                         hex2 <= `FIVE:
hex3 <= `THREE:
hex2 <= `FIVE;
                                                                         hex1 <= `SIX;
hex1 <= `ZERO;
                                                                         hex0 <= `FIVE:
hex0 <= `FIVE;
                                                                         end
                                                                         else begin
else if (h3 == `TWO && h2 == `ZERO) begin
                                                                         hex3 <= h3;
hex3 <= `THREE;
hex2 <= `FIVE;
                                                                         hex2 <= h2;
                                                                         hex1 <= h1;
hex1 <= `ONE;
                                                                         hex0 \le h0;
hex0 <= `ZERO;
                                                                         end
                                                                         end
                                                                         credit_card: begin
else if (h3 == `TWO && h2 == `FIVE) begin
hex3 <= `THREE;
                                                                         hex3 <= `THREE;
                                                                         hex2 <= `FIVE;
hex2 <= `FIVE;
hex1 <= `ONE
                                                                         hex1 <= `ZERO;
                                                                         hex0 <= `ZERO;
hex0 <= `FIVE;
                                                                         end
else if (h3 == `THREE && h2 == `ZERO) begin
                                                                         default: begin
hex3 <= `THREE;
                                                                         hex3 \le h3;
hex2 <= `FIVE;
                                                                         hex2 <= h2:
hex1 <= `TWO;
                                                                         hex1 <= h1;
hex0 <= `ZERO;
                                                                         hex0 \le h0;
end
else begin
                                                                         end
hex3 <= h3;
                                                                         endcase
hex2 <= h2;
                                                                         if (cycle == 600000) begin
hex1 <= h1;
                                                                         ledg[0] <= 1;
hex0 <= h0:
                                                                         ledg[1] \ll 1;
end
                                                                         ledg[2] <= 1;
                                                                         ledg[3] <= 1;
ledg[4] <= 1;
end
dollar: begin
                                                                         ledg[5] \ll 1;
if (h3 == `ZERO && h2 == `ZERO) begin
hex3 <= `THREE;
                                                                         ledg[6] <= 1;
hex2 <= `FIVE;
                                                                         ledg[7] \ll 1;
hex1 <= `SIX;
                                                                         end
hex0 <= `FIVE;
                                                                         else if (cycle == 12000000) begin
                                                                         ledg <= 0;
end
else if (h3 == `ZERO && h2 == `FIVE) begin
                                                                         cycle <= 0;
hex3 <= `THREE;
                                                                         end
hex2 <= `FIVE;
                                                                         ledr[9:0] <= 0;
hex1 <= `SEVEN;
                                                                         end
hex0 <= `ZERO;
                                                                         default: begin
end
                                                                         end
else if (h3 == `ONE && h2 == `ZERO) begin
                                                                         endcase
hex3 <= `THREE;
                                                                         end
hex2 <= `FIVE;
                                                                         else if (err == 1) begin
hex1 <= `SEVEN;
                                                                         cycle <= 0:
hex0 <= `FIVE;
                                                                         ledg \ll 0;
                                                                         hex3 <= `E;
end
else if (h3 == `ONE && h2 == `FIVE) begin
                                                                         hex2 <= r;
hex3 <= `THREE;
                                                                         hex1 <= r;
hex2 <= `FIVE;
                                                                         hex0 <= `BLANK;
hex1 <= `EIGHT;
                                                                         state <= zero;
hex0 <= `ZERO;
                                                                         end
end
                                                                         end
else if (h3 == `TWO && h2 == `ZERO) begin
                                                                         else if (VM == 0 && sw[9] == 0) begin
hex3 <= `THREE;
hex2 <= `FIVE;
                                                                         hex3 <= `ZERO;
                                                                         hex2 <= `FIVE;
hex1 <= `EIGHT;
                                                                         hex1 <= `EIGHT;
                                                                         hex0 <= `ZERO;
hex0 <= `FIVE;
end
                                                                         end
else if (h3 == `TWO && h2 == `FIVE) begin
                                                                         end
hex3 <= `THREE;
hex2 <= `FIVE;
                                                                         always @(negedge key[1]) begin
```

	end	
if (clear == 1) begin	quarter: begin	
flag = 0;	in = quarter;	
end	next_state = five;	
if (VM == 0) begin	end	
VM = 1;	dollar: begin	
next_state = zero;	in = dollar;	
end	if (hex1 != `ZERO && he	ex1 != `ONE && hex1 != `TWO)
else if (VM == 1) begin	begin	
if (state != seven && err == 0) begin	err = 1;	
case(sw[8:0])	next_state = zero;	
none: begin	end	
. •		U boy4 'ONE II boy4 'TWO'
in = none;	`	hex1 == `ONE hex1 == `TWO)
end	begin	
nickel: begin	next_state = seven;	
in = nickel;	flag = 1;	
if (state < seven) begin	err = 0;	
if (state == six) begin	end	
counter = counter + 1;	end	
•		
next_state = seven;	credit_card: begin	
end	in = credit_card;	0)7550).
next_state = state + one;	if (hex1 == `ZERO && h	ex0 == ZERO) begin
end	next_state = zero;	
end	err = 1;	
dime: begin	end	
in = dime;		& hex0 != `ZERO) begin
if (state < six) begin	next_state = seven;	
	<u>.</u> – .	
next_state = state + two;	flag = 1;	
end	err = 0;	
else begin	end	
next_state = seven;	end	
counter = counter + 1;	reset: begin	
end	next_state = zero;	
end	end	
quarter: begin	default: begin	
in = quarter;	err = 1;	
if (state < three) begin	next_state = zero;	
next_state = state + five;	end	
end	endcase	
else begin	end	
next_state = seven;	else if (err == 1) begin	
counter = counter + 1;	next_state = zero;	
end	err = 0;	
end	end	
dollar: begin	end	
in = dollar;	end	
next_state = seven;	endmodule	
counter = counter + 1;	efine BLANK 7'b111111	1
end	`define ZERO 7'b100000	00
credit_card: begin	`define ONE 7'b111100'	1
in = credit_card;	`define TWO 7'b010010	0
next_state = seven;	`define THREE 7'b0110	
end	`define FOUR 7'b00110	
	`define FIVE 7'b0010010	
reset: begin		o .
in = reset;	`define SIX 7'b0000010	000
next_state = zero;	`define SEVEN 7'b1111	
end	`define EIGHT 7'b00000	
default: begin	`define NINE 7'b001100	0
err = 1;	`define A 7'b0001000	
next_state = zero;	`define b 7'b0000011	
end	`define C 7'b1000110	
endcase	`define d 7'b0100011	
end	`define E 7'b0000110	
else if (state == seven && err == 0) begin	`define F 7'b0001110	
case(sw[8:0])	`define H 7'b0001001	
none: begin	`define L 7'b1000111	
in = none;	`define r 7'b0101111	
end		
nickel: begin	module L3C580	// where 580
in = nickel;	= CID	
next_state = one;	(
		// ten un-down switches SMO
end	input [9:0]sw,	// ten up-down switches, SW9 -
dime: begin	SW0	// farm muchked to a sold LENG
in = dime;	input [3:0]key,	// four pushbutton swithes, KEY3 -
next_state = two;	KEY0	

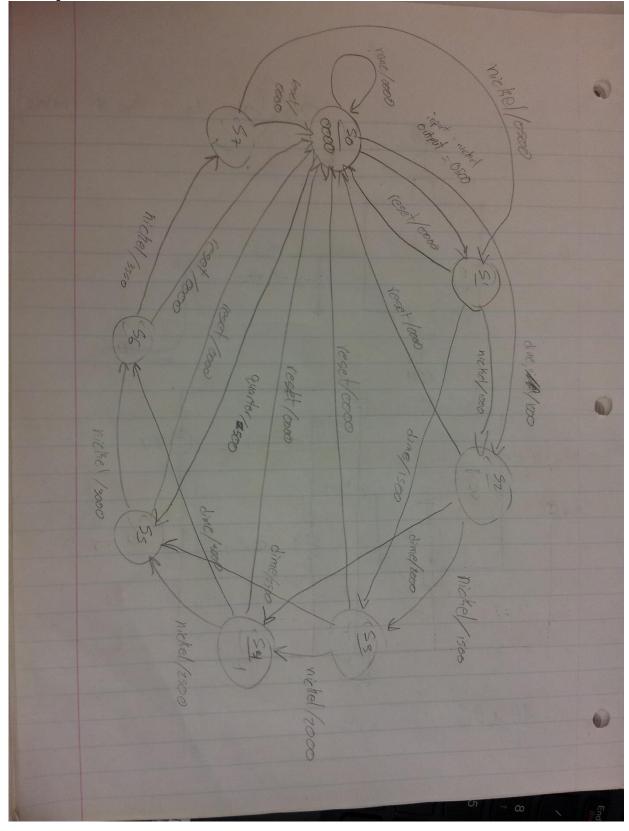
```
input clock,
                         // 24MHz clock source on Altera
                                                                          9: begin
DE1 board
                                                                          hex0 <= `NINE;
output reg [9:0]ledr,
                         // ten Red LEDs, LEDR9 - LEDR0
                                                                          end
output reg [7:0]ledg,
                         // eight Green LEDs, LEDG8 -
                                                                          10: begin
                                                                          hex0 \stackrel{\cdot}{=} A;
LEDG0
output reg [6:0]hex3,hex2,hex1,hex0
                                             // four 7-
                                                                          end
segment, HEX3 - HEX0
                                                                          11: begin
                                                                          hex0 \stackrel{-}{\leq} b;
parameter zero = 3'b000, one = 3'b001, two = 3'b010, three
                                                                          end
= 3'b011, four = 3'b100, five = 3'b101, six = 3'b110, seven =
                                                                          12: begin
                                                                          hex0 <= C;
parameter none = 9'b000000000, nickel = 9'b000000001.
                                                                          end
dime = 9'b000000010, quarter = 9'b000000100, dollar =
                                                                          13: begin
9'b000001000, credit_card = 9'b000010000, reset =
                                                                          hex0 <= d:
9'b100000000:
                                                                          end
integer in;
                                                                          14: begin
                                                                          hex0 <= E;
reg[3:0] state, next_state;
                                                                          end
                                                                          15: begin
reg [6:0] h3, h2, h1, h0;
                                                                          hex0 \le F;
                                                                          end
integer VM = 0;
                                                                          endcase
reg[3:0] counter = 0;
                                                                          end
integer err = 0;
                                                                          else if (VM == 1 && sw[9] == 0) begin
integer cycle = 0;
                                                                          rep = 0;
integer rep = 0;
                                                                          if (err == 0) begin
                                                                          case(state)
integer flag = 0;
integer clear = 0;
                                                                          zero: begin
                                                                          hex3 <= `ZERO;
hex2 <= `ZERO;
                                                                          hex1 <= `ZERO;
always @(posedge clock) begin
                                                                          hex0 <= `ZERO;
state = next_state;
                                                                          h3 <= hex3;
if (flag == 1) begin
                                                                          h2 <= hex2;
h3 = THREE;
                                                                          h1 <= hex1:
h2 = FIVE;
                                                                          h0 \le hex0;
clear = 1;
                                                                          cycle = 0;
end
                                                                          ledg[7:0] <= 0;
if (sw[9] == 1 && err == 0 && rep == 0) begin
                                                                          ledr[9:0] <= 0;
rep <= 1;
                                                                          end
h3 <= hex3;
                                                                          one: begin
h2 <= hex2;
                                                                          hex3 <= `ZERO;
h1 \le hex1;
                                                                          hex2 <= `FIVE;
h0 <= hex0:
                                                                          hex1 <= `ZERO;
hex3 <= `BLANK;
                                                                          hex0 <= `ZERO:
hex2 <= `BLANK;
                                                                          h3 <= hex3;
hex1 <= `BLANK;
                                                                          h2 <= hex2;
case (counter)
                                                                          h1 \le hex1;
0: begin
                                                                          h0 \le hex0;
hex0 <= `ZERO:
                                                                          ledg[7:0] <= 0;
end
                                                                          ledr[9:0] <= 0;
1: begin
                                                                          end
hex0 <= `ONE;
                                                                          two: begin
end
                                                                          hex3 <= `ONE:
2: begin
                                                                          hex2 <= `ZERO;
hex0 <= `TWO;
                                                                          hex1 <= `ZERO;
end
                                                                          hex0 <= `ZERO;
3: begin
                                                                          h3 <= hex3;
hex0 <= `THREE;
                                                                          h2 <= hex2;
end
                                                                          h1 \le hex1;
4: begin
                                                                          h0 \le hex0;
hex0 <= `FOUR;
                                                                          ledg[7:0] <= 0;
end
                                                                          ledr[9:0] <= 0;
5: begin
                                                                          end
hex0 <= `FIVE;
                                                                          three: begin
end
                                                                          hex3 <= `ONE;
6: begin
                                                                          hex2 <= `FIVE;
hex0 <= `SIX;
                                                                          hex1 <= `ZERO;
end
                                                                          hex0 <= `ZERO;
7: begin
                                                                          h3 <= hex3;
hex0 <= `SEVEN;
                                                                          h2 <= hex2;
end
                                                                          h1 <= hex1:
8: begin
                                                                          h0 \le hex0:
hex0 <= `EIGHT;
                                                                          ledg[7:0] <= 0;
end
                                                                          ledr[9:0] <= 0;
```

```
end
                                                                           hex2 \le h2;
four: begin
hex3 <= `TWO;
hex2 <= `ZERO;
                                                                           hex1 <= h1:
                                                                           hex0 \le h0;
                                                                           end
hex1 <= `ZERO;
                                                                           end
hex0 <= `ZERO;
                                                                           quarter: begin
                                                                           if (h3 == `ONE && h2 == `ZERO) begin
h3 <= hex3;
                                                                           hex3 <= `THREE;
h2 <= hex2:
                                                                           hex2 <= `FIVE;
h1 \le hex1;
                                                                          hex1 <= `ZERO;
hex0 <= `ZERO;
h0 <= hex0;
ledg[7:0] <= 0;
ledr[9:0] <= 0;
                                                                           end
                                                                           else if (h3 == `ONE && h2 == `FIVE) begin
end
                                                                          hex3 <= `THREE;
hex2 <= `FIVE;
five: begin
hex3 <= `TWO;
hex2 <= `FIVE;
                                                                           hex1 <= `ZERO;
hex1 <= `ZERO;
                                                                           hex0 <= `FIVE;
hex0 <= `ZERO;
                                                                           end
h3 <= hex3;
                                                                           else if (h3 == `TWO && h2 == `ZERO) begin
h2 <= hex2;
                                                                           hex3 <= `THREE;
h1 \le hex1;
                                                                           hex2 <= `FIVE;
h0 \le hex0;
                                                                           hex1 <= `ONE
ledg[7:0] <= 0;
                                                                           hex0 <= `ZERO;
ledr[9:0] <= 0;
                                                                           end
                                                                           else if (h3 == `TWO && h2 == `FIVE) begin
end
six: begin
                                                                           hex3 <= `THREE;
                                                                           hex2 <= `FIVE;
hex3 <= `THREE;
hex2 <= `ZERO;
                                                                           hex1 <= `ONE;
hex1 <= `ZERO;
hex0 <= `ZERO;
                                                                           hex0 <= `FIVE;
                                                                           end
h3 <= hex3;
                                                                           else if (h3 == `THREE && h2 == `ZERO) begin
                                                                           hex3 <= `THREE;
h2 <= hex2;
                                                                           hex2 <= `FIVE;
h1 <= hex1;
                                                                          hex1 <= `TWO;
hex0 <= `ZERO;
h0 \le hex0;
ledg[7:0] <= 0;
ledr[9:0] <= 0;
                                                                           end
end
                                                                           else begin
seven: begin
                                                                           hex3 <= h3;
cycle <= cycle + 1;
                                                                           hex2 <= h2:
case (in)
                                                                           hex1 <= h1;
none: begin
                                                                           hex0 \le h0;
hex3 \le h3;
                                                                           end
hex2 <= h2;
                                                                           end
                                                                           dollar: begin
hex1 \le h1;
                                                                          if (h3 == `ZERO && h2 == `ZERO) begin
hex3 <= `THREE;
hex0 \le h0;
end
                                                                           hex2 <= `FIVE;
nickel: begin
hex3 <= `THREE;
                                                                           hex1 <= `SIX;
hex2 <= `FIVE;
                                                                           hex0 <= `FIVE;
hex1 <= `ZERO;
                                                                           end
hex0 <= `ZERO;
                                                                           else if (h3 == `ZERO && h2 == `FIVE) begin
                                                                           hex3 <= `THREE;
hex2 <= `FIVE;</pre>
h3 <= hex3;
h2 <= hex2;
h1 <= hex1;
                                                                           hex1 <= `SEVEN;
h0 <= hex0;
                                                                           hex0 <= `ZERO:
end
                                                                           end
dime: begin
                                                                           else if (h3 == `ONE && h2 == `ZERO) begin
if (h3 == `THREE && h2 == `ZERO) begin
                                                                           hex3 <= `THREE;
hex3 <= `THREE;
                                                                           hex2 <= `FIVE;
hex2 <= `FIVE;
                                                                           hex1 <= `SEVEN;
hex1 <= `ZERO;
                                                                           hex0 <= `FIVE;
hex0 <= `FIVE;
                                                                           end
h3 <= hex3;
                                                                           else if (h3 == `ONE && h2 == `FIVE) begin
                                                                           hex3 <= `THREE;
hex2 <= `FIVE;
h2 <= hex2:
h1 <= hex1;
h0 \le hex0:
                                                                           hex1 <= `EIGHT;
                                                                           hex0 <= `ZERO;
else if (h3 == `TWO && h2 == `FIVE) begin
                                                                           end
hex3 <= `THREE;
                                                                           else if (h3 == `TWO && h2 == `ZERO) begin
hex2 <= `FIVE;
                                                                           hex3 <= `THREE;
hex1 <= `ZERO;
                                                                           hex2 <= `FIVE;
hex0 <= `ZERO;
                                                                           hex1 <= `EIGHT;
end
                                                                           hex0 <= `FIVE;
else begin
                                                                           end
hex3 <= h3;
                                                                           else if (h3 == `TWO && h2 == `FIVE) begin
```

```
hex3 <= `THREE;
hex2 <= `FIVE;
                                                                           always @(negedge key[1]) begin
hex1 <= `NINE;
hex0 <= `ZERO:
                                                                           if (clear == 1) begin
end
                                                                           flag = 0;
else if (h3 == `THREE && h2 == `ZERO) begin
                                                                           end
hex3 <= `THREE;
                                                                           if (VM == 0) begin
hex2 <= `FIVE:
                                                                           V\dot{M} = 1:
hex1 <= `NINE;
                                                                           next_state = zero;
hex0 <= `FIVE;
                                                                           end
                                                                           else if (VM == 1) begin
end
else if (h3 == `THREE && h2 == `FIVE) begin
                                                                           if (state != seven && err == 0) begin
hex3 <= `THREE;
                                                                           case(sw[8:0])
hex2 <= `FIVE;
hex1 <= `SIX;
                                                                           none: begin
                                                                           in = none;
hex0 <= `FIVE;
                                                                           end
                                                                           nickel: begin
end
else begin
                                                                           in = nickel;
hex3 <= h3;
                                                                           if (state < seven) begin
hex2 <= h2;
                                                                           if (state == six) begin
hex1 \le h1;
                                                                           counter = counter + 1;
hex0 \le h0;
                                                                           next_state = seven;
end
                                                                           end
end
                                                                           next_state = state + one;
credit_card: begin
                                                                           end
hex3 <= `THREE;
                                                                           end
hex2 <= `FIVE;
                                                                           dime: begin
hex1 <= `ZERO;
                                                                           in = dime;
hex0 <= `ZERO;
                                                                           if (state < six) begin
end
                                                                           next_state = state + two;
default: begin
                                                                           end
hex3 <= h3;
                                                                           else begin
hex2 <= h2;
                                                                           next state = seven;
hex1 <= h1;
                                                                           counter = counter + 1;
hex0 \le h0;
                                                                           end
                                                                           end
end
                                                                           quarter: begin
endcase
                                                                           in = quarter;
if (cycle == 6000000) begin
                                                                           if (state < three) begin
ledg[0] \ll 1;
                                                                           next_state = state + five;
ledg[1] <= 1;
ledg[2] <= 1;
                                                                           end
                                                                           else begin
ledg[3] <= 1;
                                                                           next state = seven;
ledg[4] <= 1;
                                                                           counter = counter + 1;
ledg[5] <= 1;
                                                                           end
ledg[6] \ll 1;
                                                                           end
ledg[7] \ll 1;
                                                                           dollar: begin
end
                                                                           in = dollar;
else if (cycle == 12000000) begin
                                                                           next_state = seven;
ledg \le 0;
                                                                           counter = counter + 1;
cycle <= 0;
                                                                           end
end
                                                                           credit_card: begin
ledr[9:0] <= 0;
                                                                           in = credit_card;
                                                                           next_state = seven;
end
default: begin
                                                                           end
                                                                           reset: begin
end
endcase
                                                                           in = reset;
                                                                           next_state = zero;
end
else if (err == 1) begin
                                                                           end
cycle <= 0;
                                                                           default: begin
ledg \ll 0;
                                                                           err = 1;
hex3 <= `E;
                                                                           next_state = zero;
hex2 <= `r;
                                                                           end
hex1 <= `r;
                                                                           endcase
hex0 <= `BLANK;
                                                                           end
state <= zero;
                                                                           else if (state == seven && err == 0) begin
end
                                                                           case(sw[8:0])
                                                                           none: begin
end
else if (VM == 0 && sw[9] == 0) begin
                                                                           in = none;
hex3 <= `ZERO;
                                                                           end
hex2 <= `FIVE;
                                                                           nickel: begin
hex1 <= `EIGHT;
                                                                           in = nickel;
hex0 <= `ZERO;
                                                                           next_state = one;
end
                                                                           end
end
                                                                           dime: begin
```

```
in = dime;
                                                                          err = 1;
next_state = two;
                                                                          end
                                                                          else if (hex1 != `ZERO && hex0 != `ZERO) begin
end
quarter: begin
                                                                          next_state = seven;
in = quarter;
                                                                          flag = 1;
                                                                          err = 0;
end
next_state = five;
end
dollar: begin
                                                                          end
in = dollar;
                                                                          reset: begin
                                                                          next_state = zero;
end
if (hex1 != `ZERO && hex1 != `ONE && hex1 != `TWO)
begin
err = 1;
                                                                          default: begin
next_state = zero;
                                                                          err = 1;
end
                                                                          next_state = zero;
else if (hex1 == `ZERO || hex1 == `ONE || hex1 == `TWO)
                                                                          end
                                                                          endcase
next_state = seven;
                                                                          end
flag = 1;
err = 0;
                                                                          else if (err == 1) begin
                                                                          next_state = zero;
end
                                                                          err = 0;
                                                                          end
end
credit_card: begin
                                                                          end
                                                                          end
in = credit_card;
if (hex1 == `ZERO && hex0 == `ZERO) begin
                                                                          endmodule
next_state = zero;
```

C) State Diagram



d) Compilation Report

Compliation	rtoport	
Flow Summary		
	Flow Status	Successful - Tue May 20 13:08:18 2014
	Quartus II Version	9.0 Build 235 06/17/2009 SP 2 SJ Web Edition
	Revision Name	L3C580
	Top-level Entity Name	L3C580
	Family	Cyclone II
	Device	EP2C20F484C7
	Timing Models	Final
	Met timing requirements	Yes
	Total logic elements	389 / 18,752 (2 %)
	Total combinational functions	382/18.752(2%)
	Dedicated logic registers	110/18.752 (<1%) 110
	Total registers Total pins	61/315(19%)
	Total virtual pins	0
	Total memory bits	0/239.616 (0%)
	Embedded Multiplier 9-bit eleme	
	Total PLLs	0/4(0%)
J		