Jomar Pueyo UXT302

**Lab Assignment 3: BCD to Binary**

Procedure:

Convert a 12-Bit Binary Coded Decimal into a 10-Bit Binary System. The process starts when signal St is received and stores the binary sequence in a register. This is achieved by shifting bits to the right one bit at a time while being converted in between each process. If any of three 4bit sections are greater than 4’b1000, 4’b0011 are subtracted.

Observations:

While working on the assignment I found it best to separate the 12bits into three 4-bit sections. At each section we check to if any of the segments are greater that 4’b1000, if there are, we subtract 4’b0011, if not, then we shift the bits to the right into the B register. This process happens ten times for a maximum value of 999 in binary.

Results:

Code:

module bcd(bcdin,st,bin);

input [11:0] bcdin; //register A

input st; //start

output reg [9:0] bin; //register B

reg [11:0] bcd; //copy of input

reg [3:0] hundreds,tens,ones; //4bit bcd places

integer i; //FOR LOOP OR ELSE INFINITE SHIFT TO THE RIGHT (I learned my lesson)

always@\*

begin

if(st==1'b1);

begin

bcd=bcdin;

for(i=0;i<10;i=i+1)

begin

bin[i]=bcd[0];

bcd=bcd>>1;

//break bcd into 3 4bit sections

hundreds={bcd[3],bcd[2],bcd[1],bcd[0]};

tens ={bcd[7],bcd[6],bcd[5],bcd[4]};

ones ={bcd[11],bcd[10],bcd[9],bcd[8]};

//subtract 4'b0011 if greater than 4'b1000

if(hundreds>=4'b1000)

begin

hundreds=hundreds-4'b0011;

bcd[0]=hundreds[0];

bcd[1]=hundreds[1];

bcd[2]=hundreds[2];

bcd[3]=hundreds[3];

end

//subtract 4'b0011 if greater than 4'b1000

if(tens>=4'b1000)

begin

tens=tens-4'b0011;

bcd[4]=tens[0];

bcd[5]=tens[1];

bcd[6]=tens[2];

bcd[7]=tens[3];

end

//subtract 4'b0011 if greater than 4'b1000

if(ones>=4'b1000)

begin

ones=ones-4'b0011;

bcd[8]=ones[0];

bcd[9]=ones[1];

bcd[10]=ones[2];

bcd[11]=ones[3];

end

end

end

end

endmodule

Testbench:

module bcd\_tb;

// Inputs

reg [11:0] bcdin;

reg st;

// Outputs

wire [9:0] bin;

// Instantiate the Unit Under Test (UUT)

bcd uut (

.bcdin(bcdin),

.st(st),

.bin(bin)

);

initial begin

// Initialize Inputs

bcdin = 0;

st = 0;

#100

bcdin = 12'b001000100111;

st = 1'b1;

#100;

bcdin = 12'b001000100111;

st = 1'b1;

#100;

bcdin = 12'b000000000000;

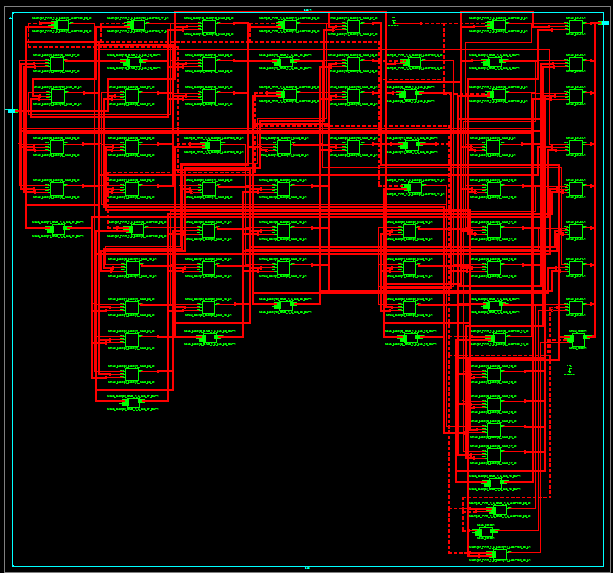
st = 1'b1;

#100;

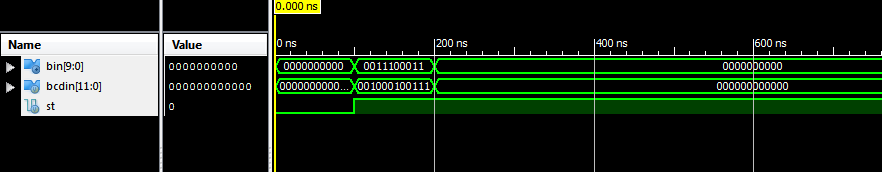
end

endmodule

Schematic:



Timing Diagram:



State Diagram:

(So) --- Start/Load ---> (S1) --- C10’/ShiftRight ---> (S2)

^ C10/Done ^ \_\_\_\_Updated Values\_\_\_\_\_\_|

Conclusion:

Successfully created a 12-Bit Binary Coded Decimal into a 10-Bit Binary System mainly using a loop of ten rotations that consisted of breaking the 12 bits into three 4-bit segments, shifting the bits to the right, checking for the segments if they were greater than 8, subtracting when necessary, and repeatedly doing so until the number converted to binary.