EEL2020 Digital Design Lab Report

Sem II AY 2023-24

Experiment No.	:	**4**
Name	:	**Lavangi Parihar**
Roll No.	:	**B22EE044**
Partner Name (Partner Roll No.)	:	**Mansi Choudhary (B22EE045)**

Objective

Simulate and implement a BCD Adder/Subtractor circuit.
The selection of addition/subtraction operation should be done by another input. The sign (in case of
subtraction) and carry (in case of addition) should be shown on an LED, while the BCD difference or
sum should be shown on a 7-segment display. You may use Verilog operators of addition and Subtraction.

Logic Design:

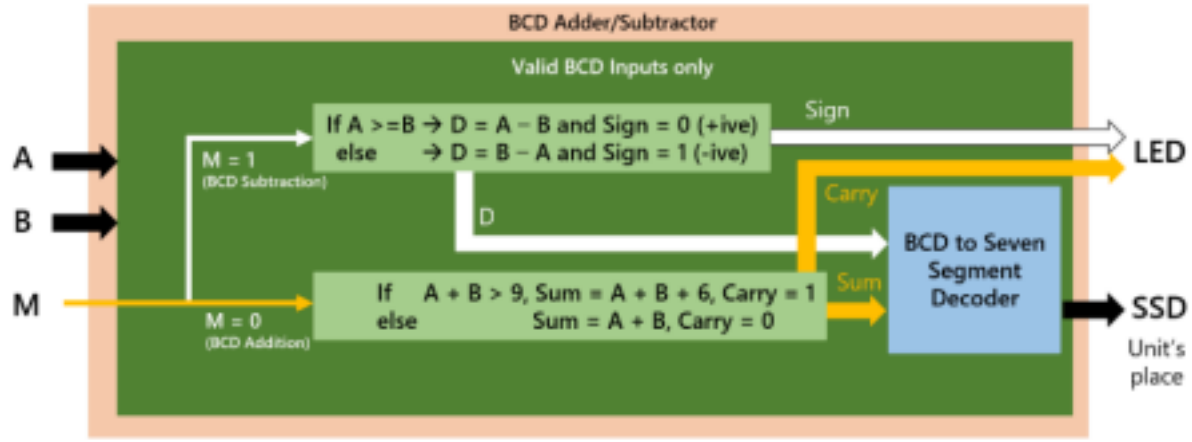


Figure 1 Module structure

Input Handling:

Validate inputs A and B as 4-bit BCD numbers.

Use mode selection input (M) to determine operation (addition/subtraction). Arithmetic Logic:

Subtraction (M = 1):

Compare A and B to find a larger number.

Subtract a smaller number from a larger one.

Determine signs based on comparison.

Addition (M = 0):

Add A and B.

Detect carry out if sum exceeds 9.

Output Generation:

Convert BCD sum/difference to 7-segment display pattern.

LED indicates carry or sign based on operation.

Control Logic:

Ensure correct operation based on mode selection.

Sequence control for proper functionality.

Source Description :

Design source:

```
module BCD_Addition_Subtraction
input [3:0] A, // 4-bit BCD input for minuend
  input [3:0] B, // 4-bit BCD input for subtrahend
  input M,       // Selection line (M=1 for subtraction, M=0 for addition)
  output reg [4:0] out_value,
  output reg [6:0] seg_out,     // Output for seven-segment display
  output reg C_sign           // Output carry or sign (Sign: 1 if negative, 0 otherwise; Carry: 1
representing)
);
reg [4:0] temp_out_value;

  // Perform subtraction or addition
  always @*
  begin
    if (A <= 4'b1001 && B <= 4'b1001)
    begin
      if (M == 1)
      begin
        if (A >= B)
        begin
          out_value = A - B;
          C_sign = 0;
        end
        else
        begin
          // Negative subtraction
          out_value = B - A;
          C_sign = 1;
        end
      end
```

```
      else
      begin
        // Addition
        temp_out_value = A + B;
        if (temp_out_value > 4'b1001)
        begin
          // Carry out if sum exceeds 9
          C_sign = 1;
          out_value = temp_out_value + 4'b0110;
        end
        else
        begin
          out_value = temp_out_value;
          C_sign = 0;
        end
      end
      $display("A = %b, B = %b, Subtraction? = %b, C_out_Sign = %b, Output value = %d", A, B, M,
C_sign, out_value);
    end
    else
      out_value = 4'b1111;
  end

  // Display the absolute value of the difference on a common anode seven-segment display
  always @*
  begin
    case (out_value[3:0])
      4'b0000: seg_out = 7'b1000000; //0111111
      4'b0001: seg_out = 7'b1111001; // Display 1
      4'b0010: seg_out = 7'b0100100; // Display 2
      4'b0011: seg_out = 7'b0110000; // Display 3
      4'b0100: seg_out = 7'b0011001; // Display 4
      4'b0101: seg_out = 7'b0010010; // Display 5
      4'b0110: seg_out = 7'b0000010; // Display 6
      4'b0111: seg_out = 7'b1111000; // Display 7
      4'b1000: seg_out = 7'b0000000; // Display 8
      4'b1001: seg_out = 7'b0010000; // Display 9
      default: seg_out = 7'b1111111; // Blank display for other values
    endcase
  end
endmodule
```

- Constraint file

The PYNQ-Z2. XDC file was updated with the following changes:

XDC File	Ports (from Verilog module)	Designation (Input/Output)	PYNQ Component Type (Button/LED/Switch etc. along with number, eg. LD01, BTN2, etc.)	PIN configuration (from the PYNQ manual)
(PYNQ-z2 board constraints file)	M	Input	SW1 M19	
	c_out_sign	Output	LED1 R14	
	seg_out[0]	Output	JB1_P W14	
	seg_out[1]	Output	JB1_N Y14	
	seg_out[2]	Output	JB2_P T11	
	seg_out[3]	Output	JB2_N T10	
	seg_out[4]	Output	JB3_P V16	
	seg_out[5]	Output	JB3_N W16	
	seg_out[6]	Output	JB4_P V12	

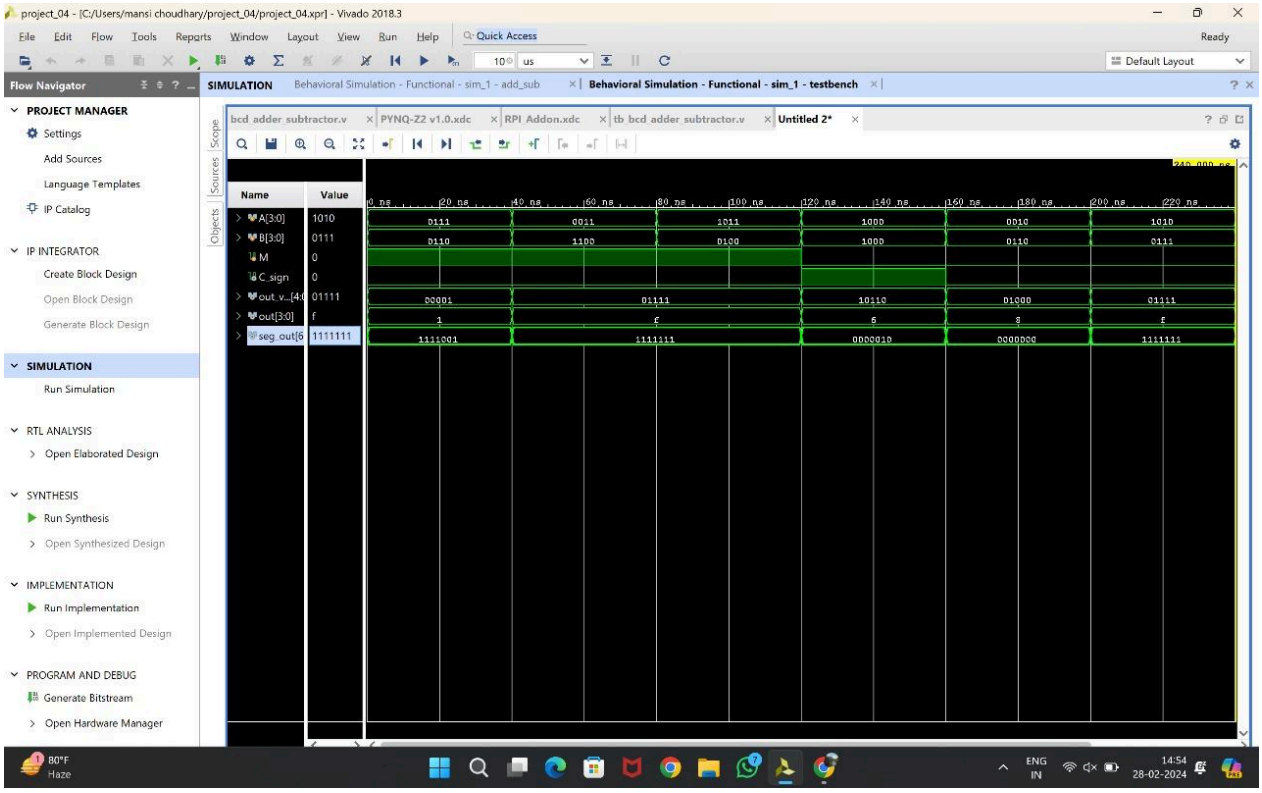
The RPI_ADDON. XDC file was updated with the following changes:

Ports	Designation	Component Type	Pin configuration
B [0]	input	get_ports (SWA)	V6
B [1]	input	get_ports (SWB)	Y6
B [2]	input	get_ports (SWC)	B19
B [3]	input	get_ports (SWD)	U7
A [0]	input	get_ports (SWE)	C20
A [1]	input	get_ports (SWF)	Y8
A [2]	input	get_ports (SWG)	A20
A [3]	input	get_ports (SWH)	W9

- Simulation source (if any)

```
 module add_sub (
   input [3:0] A,
   input [3:0] B,
   input M,
   output [6:0] seg_out,
   output c_out_sign
 );
   BCD_Addition_Subtraction addition_sub (.A(A), .B(B), .M(M), .out_value(), .seg_out(seg_out),
 .C_sign(c_out_sign));
endmodule
```

Simulation Results **(Timing diagram)**



PYNQ Working Video (to be recorded during lab session) Video [video]