ASSIGNMENT - 2

DIGITAL SYSTEMS ES204

Guntas Singh Saran
guntassingh.saran@iitgn.ac.in
22110089

Nihar Shah
nihar.shah@iitgn.ac.in
22110237

PART - 1

8-bit combined Adder-Subtractor using only **Full Adders**

The module has two 8-bit inputs (**A, B**). It has a **mode** input to select between Add/Sub
It has one 8-bit output **C** and **carry_borrow_out**

VERILOG CODE

```
module FullAdder(input [2:0] A, output SUM, output carry);
// A[2] = A, A[1] = B, A[0] = C_in
xor (SUM, A[0], A[1], A[2]);
and (a1, A[2], A[1]);
and (a2, A[1], A[0]);
and (a3, A[0], A[2]);
or (carry, a1, a2, a3);
endmodule
```

```
`timescale 1ns / 1ps

module Adder_sub8bit(
 input [7:0] A,
 input [7:0] B,
 input mode,
 output [7:0] SUM,
 output [7:0] gray,
 output carry
);
wire s1, s2, s3, s4, s5, s6, s7, s8, c1, c2, c3, c4, c5, c6, c7, c8;
wire S1, S2, S3, S4, S5, S6, S7, S8, C1, C2, C3, C4, C5, C6, C7, C8;
reg [7:0] SUM1;
reg carry1;
wire [7:0] gray0;

FullAdder add1(.A({A[0], B[0] ^ mode, mode}), .SUM(s1), .carry(c1));
FullAdder add2(.A({A[1], B[1] ^ mode, c1}), .SUM(s2), .carry(c2));
FullAdder add3(.A({A[2], B[2] ^ mode, c2}), .SUM(s3), .carry(c3));
FullAdder add4(.A({A[3], B[3] ^ mode, c3}), .SUM(s4), .carry(c4));
FullAdder add5(.A({A[4], B[4] ^ mode, c4}), .SUM(s5), .carry(c5));
FullAdder add6(.A({A[5], B[5] ^ mode, c5}), .SUM(s6), .carry(c6));
FullAdder add7(.A({A[6], B[6] ^ mode, c6}), .SUM(s7), .carry(c7));
FullAdder add8(.A({A[7], B[7] ^ mode, c7}), .SUM(s8), .carry(c8));

FullAdder add9(.A({~s1, 1'b0 , 1'b1}), .SUM(S1), .carry(C1));
```

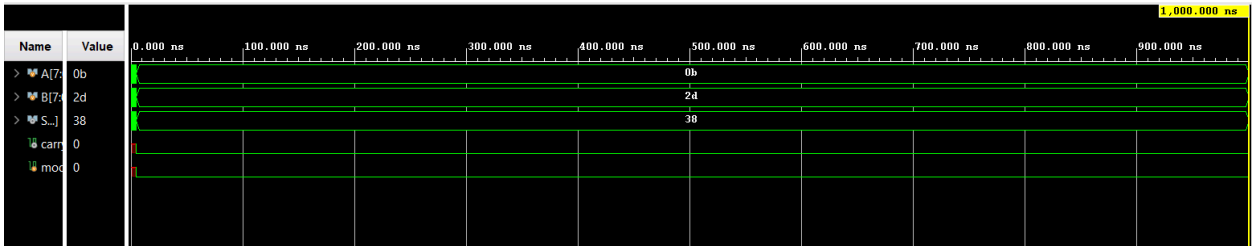
```
FullAdder add10(.A({~s2, 1'b0, C1}), .SUM(S2), .carry(C2));
FullAdder add11(.A({~s3, 1'b0, C2}), .SUM(S3), .carry(C3));
FullAdder add12(.A({~s4, 1'b0, C3}), .SUM(S4), .carry(C4));
FullAdder add13(.A({~s5, 1'b0, C4}), .SUM(S5), .carry(C5));
FullAdder add14(.A({~s6, 1'b0, C5}), .SUM(S6), .carry(C6));
FullAdder add15(.A({~s7, 1'b0, C6}), .SUM(S7), .carry(C7));
FullAdder add16(.A({~s8, 1'b0, C7}), .SUM(S8), .carry(C8));

always @* begin
 if (mode == 1 & c8 == 0) begin
   SUM1 = {S8, S7, S6, S5, S4, S3, S2, S1};
   carry1 = C8;
 end
  else begin
   SUM1 = {s8, s7, s6, s5, s4, s3, s2, s1};
   carry1 = c8;
 end
end

assign SUM = SUM1;
assign carry = carry1;

GrayConverter gray1(.binary(SUM), .gray(gray0));
assign gray = gray0;
endmodule



module GrayConverter(input [7:0] binary, output [7:0] gray);
 assign gray[7] = binary[7] ;
 assign gray[6] = binary[6] ^ binary[7];
 assign gray[5] = binary[5] ^ binary[6];
 assign gray[4] = binary[4] ^ binary[5];
 assign gray[3] = binary[3] ^ binary[4];
 assign gray[2] = binary[2] ^ binary[3];
 assign gray[1] = binary[1] ^ binary[2];
 assign gray[0] = binary[0] ^ binary[1];
endmodule
```

TESTBENCH

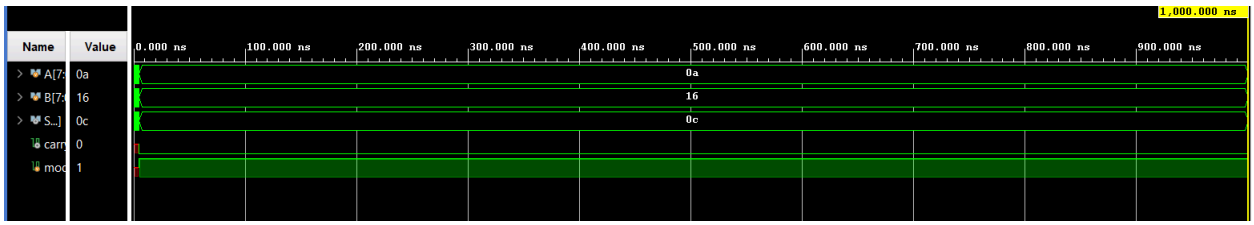
```
module AdderTest4();

reg [7:0] A, B;
wire [7:0] SUM;
wire carry;
reg mode;

wire [7:0] gray;

Adder_sub8bit adder1(.A(A), .B(B), .mode(mode),
.SUM(SUM),.gray(gray), .carry(carry));

initial begin
   // mode = 1 -> Subtract
   // mode = 0 -> Add
   A = 8'b10001010; B = 8'b10100001; mode = 1'b0; // A = 138, B = 161
 #5 A = 8'b10010001; B = 8'b01010110; mode = 1'b1; // A = 145, B = 86
 #5 A = 8'b01010110; B = 8'b10010001; mode = 1'b1; // A = 86, B = 145
 #5 A = 8'b00001011; B = 8'b00101101; mode = 1'b0; // A = 11, B = 45
 #5 A = 8'b00001010; B = 8'b00010110; mode = 1'b1; // A = 10, B = 22
 #5 A = 8'b00010110; B = 8'b00001010; mode = 1'b1; // A = 22, B = 10
end

endmodule
```

RESULTS OF SIMULATION

1. mode = 0, A = 00001011 (11), B = 00101101 (45)

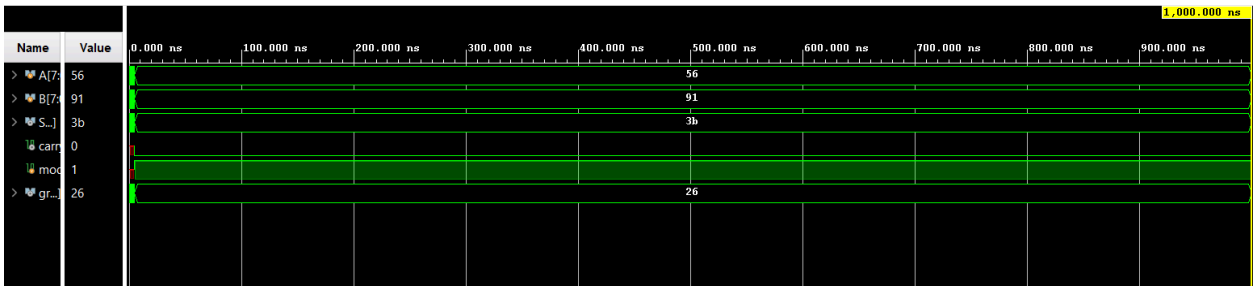


2. mode = 1, A = 00001010 (10), B = 00010110 (22)



GRAY CODE TESTING

3. mode = 1, A = 01010110(86), B = 10010001(145)



4. mode = 1, A = 00001010 (10), B = 00010110 (22)