IDP EE3025

CALCULATOR

HRITIK N CHAVAN-EE16BTECH11012
SWAPNIL P JOSHI-EE16BTECH11016

INDIAN INSTITUTE OF TECHNOLOGY HYDERABAD

29 MARCH 2019

Problem Statement

- Implementation of Calculator with basic operations
  - ADDER
  - SUBTRACTOR
  - MULTIPLIER
  - DIVIDER
  - MODULUS

Introduction

- Implementing a simple calculator which performs basic operations of addition, subtraction, multiplication , division and modulus on the given 4-bit binary number.

Progress

We have implemented addition and subtraction by realising a full adder.

- Adder implemented by sequential addition of bits.
- Subtractor by using 2's complement of the number.

Listing 1: ADDER-SUBTRACTOR

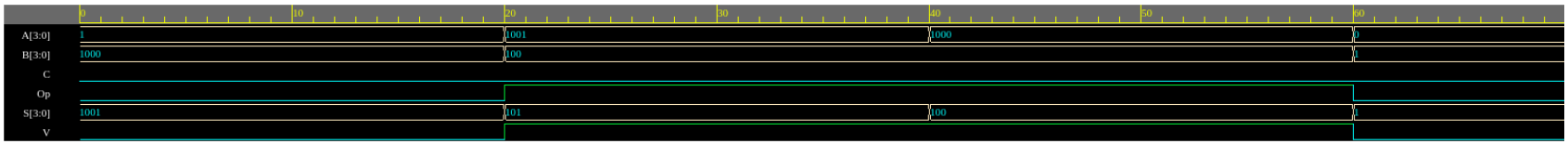
```
module ripple_carry_adder_subtractor(S, C, V, A, B,
    output [3:0] S;
    output       C;
    output       V;
    input  [3:0] A;
    input  [3:0] B;
    input        Op;

    wire         C0;
    wire         C1;
    wire         C2;
    wire         C3;
    wire         B0;
    wire         B1;
    wire         B2;
    wire         B3;
```

```
    xor(B0, B[0], Op);
    xor(B1, B[1], Op);
    xor(B2, B[2], Op);
    xor(B3, B[3], Op);
    xor(C, C3, Op);
    xor(V, C3, C2);
    full_adder fa0(S[0], C0, A[0], B0, Op);
    full_adder fa1(S[1], C1, A[1], B1, C0);
    full_adder fa2(S[2], C2, A[2], B2, C1);
    full_adder fa3(S[3], C3, A[3], B3, C2);
endmodule

module full_adder(S, Cout, A, B, Cin);
    output S;
    output Cout;
    input  A;
    input  B;
    input  Cin;
```

```
    wire   w1;
    wire   w2;
    wire   w3;
    wire   w4;
    xor(w1, A, B);
    xor(S, Cin, w1);
    and(w2, A, B);
    and(w3, A, Cin);
    and(w4, B, Cin);
    or(Cout, w2, w3, w4);
endmodule
```



- A AND B ARE THE INPUTS
- S ID THE OUTPUT(FINAL RESULT)
- C IS THE CARRY

Progress

We have implemented multiplication algorithm using Vedic Multiplier.

Listing 2: MULTIPLIER

```
module Full_Adder(input x,y,cin,output s, cout);
wire c1,c2,c3;
xor(s,x,y,cin);
and(c1,x,y),
(c2,x,cin),
(c3,y,cin);
or(cout,c1,c2,c3);
endmodule

// Multiplier

module M4bit(input [3:0] Q,input [3:0] M,output [7:
wire c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11;
wire d1,d2,d3,d4,d5,d6,d7;
wire e1,e2,e3;
wire f1,f2,f3,f4,f5,f6,f7;
wire g1,g2,g3,g4;
```

```
and(c1,M[3],Q[1]),
(c2,M[2],Q[2]),
(c3,M[1],Q[3]),
(c4,M[3],Q[0]),
(c5,M[2],Q[1]),
(c6,M[1],Q[2]),
(c7,M[2],Q[0]),
(c8,M[1],Q[1]),
(c9,M[0],Q[2]),
(c10,M[1],Q[0]),
(c11,M[0],Q[1]),
(P[0],M[0],Q[0]);

Full_Adder fa1(c1,c2,c3,d2,d1);
Full_Adder fa2(c4,c5,c6,d4,d3);
Full_Adder fa3(c7,c8,c9,d6,d5);
Full_Adder fa4(c10,c11,0,P[1],d7);
```

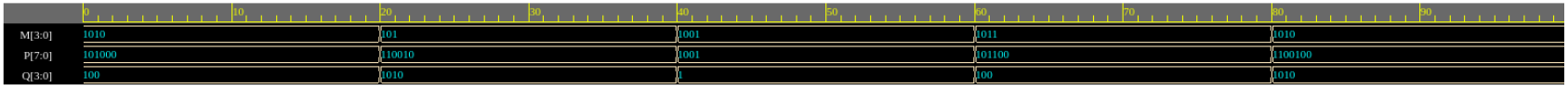
```
and(e1,M[2],Q[3]),
(e2,M[3],Q[2]),
(e3,M[0],Q[3]);

Full_Adder fa5(e1,e2,d1,f2,f1);
Full_Adder fa6(d2,d3,f5,f4,f3);
Full_Adder fa7(d4,e3,d5,f6,f5);
Full_Adder fa8(d6,d7,0,P[2],f7);

and(g1,M[3],Q[3]);

Full_Adder fa9(g1,f1,g2,P[6],P[7]);
Full_Adder fa10(f2,f3,g3,P[5],g2);
Full_Adder fa11(f4,0,g4,P[4],g3);
Full_Adder fa12(f6,f7,0,P[3],g4);

endmodule
```



- M IS THE MULTIPLICAND
- Q IS THE MULTIPLIER
- P IS THE PRODUCT

Approach

- The input numbers are given to the arduino using a keypad.
- The arduino then communicates with the raspbery pi and sends the information to be processed.
- The ico board is connected to the raspberry pi and it acts as the processing element to compute the solution.
- The solution is then transmitted to the arduino board with the help of Raspberry pi.
- Finally, the solution is displayed on the LCD screen with the help of arduino.