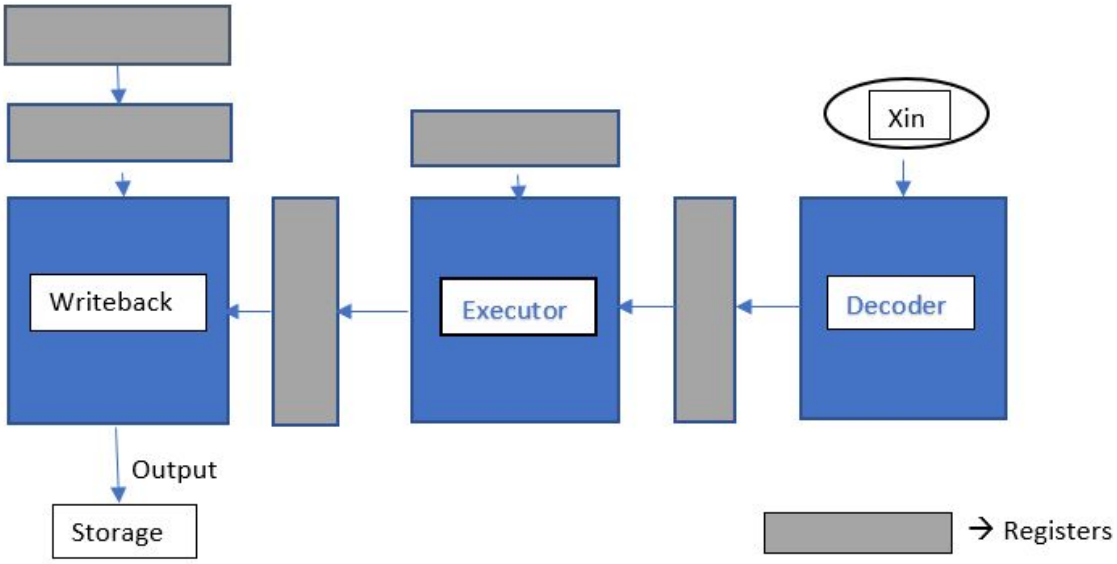
EE1193

Introduction to Hardware Description Language

Pseudo Code and Block Diagram of Pipeline Processor



Schematic Diagram for Functioning of a Pipeline Processor

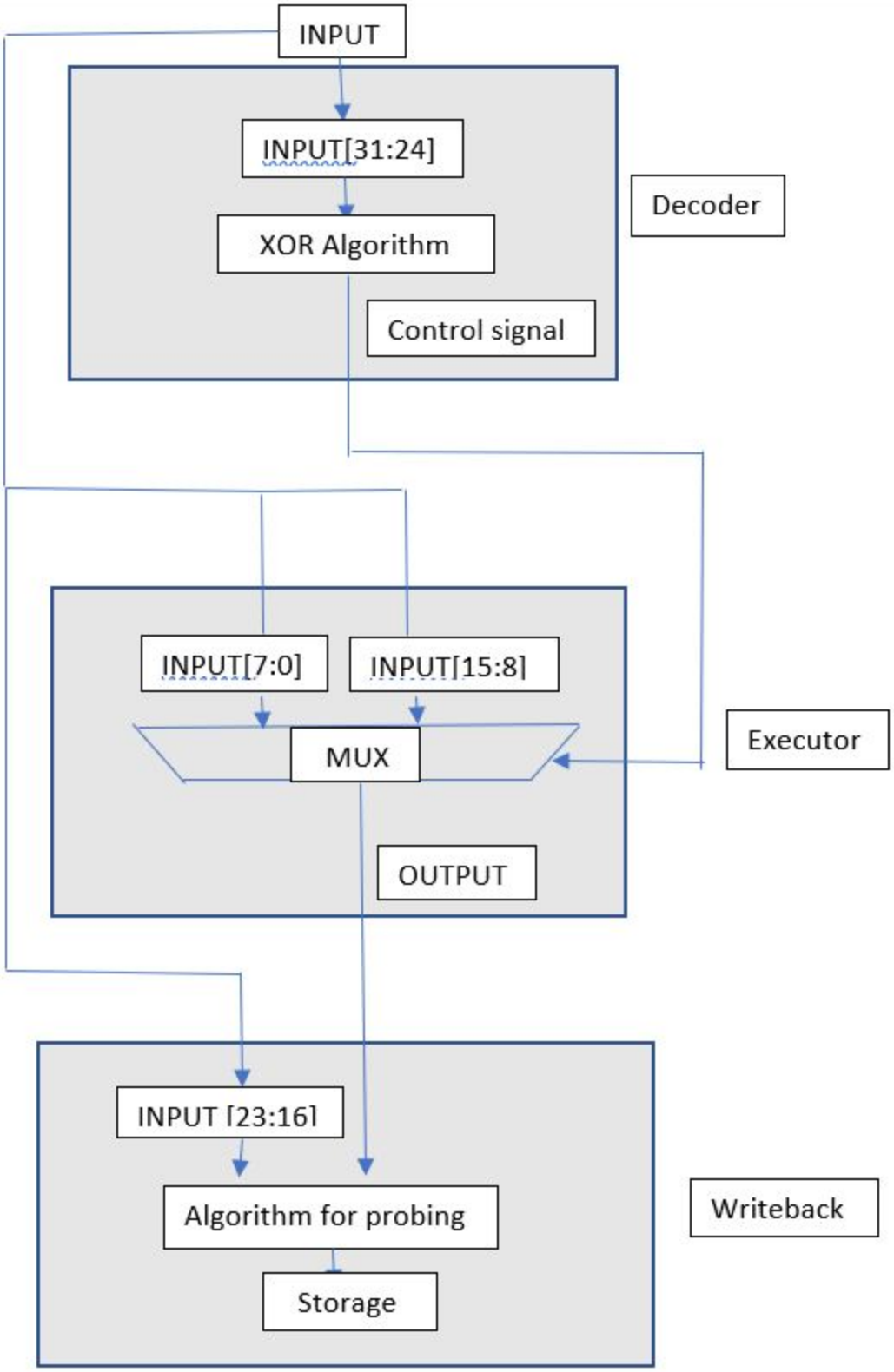
Pseudo Code

```
module Adder (output wire cout, output wire [7:0] s, input wire [7:0] x,y, input wire cin);
//This module adds 2 '8' bit numbers

wire sel[7:0];
wire c[8:0];
assign c[0] = cin;

        genvar i;
   generate
      for(i = 0; i < 8; i = i + 1)
                        begin
                                Xor s0(sel[i],x[i],y[i]);
                                Mux m0(c[i+1],y[i],c[i],sel[i]);        //Using MUX to calculate
next carry states
                                Xor z0(s[i],sel[i],c[i]);          //Final Sum
                        end
   endgenerate

assign cout = c[8];

endmodule


module proc(output wire [7:0] y, input wire [31:0] x,input wire clk);
reg sel;
wire t,cin,s;                                                   //Declaring all wires and registers
reg [7:0] a,b,z,z1,c1,i1,c,o;
wire [7:0] p;
reg [7:0] storage [255:0];                              //Memory Element for storage of the  Data

integer k;
// Decoder stage
assign s = x[31];                       // Taking a selector for the mux in the executor stage

assign cin = 0;
```

```
//Execution Stage
Adder j1(t,p,a,b,cin);

always @(sel)                                   //Changes happening at change of selector
begin
        if (sel == 0)                                   //Operation regarding value of sel
                begin
                        z = a & b;                      //"AND" Operation of 2 bytes
                end

        else
                begin
                        z = p;                          //Adding 2 bytes
                end


end

always @(posedge clk)//(Pipelining) pushing the values to registers after each clock cycles
begin
        sel <= s;               //From 's' the data goes to sel after one clock cycle
        a <= x[15:8];           //From input the data goes to a after one clock cycle
        b <= x[7:0];            //Similar to a
        z1 <= z;        //z is the output from the executor stage, data being sent to register
        o<=z1;
        c1 <= x[23:16]; // X[23:16] is the address where the data has to be stored
        k <= c1;                                //Converting Binary output to an integer
end
// Write back Stage
always@(*)
begin
        storage[k] = o;                                 //Storing output in Memory
end

assign y = storage[k];                                  //Assigning data in memory
to output wire to display

Endmodule
```

```
module test;   //Test Bench
localparam width = 32;
wire [7:0] y;              //Output
reg [width-1:0] x;      //Input
reg clk;                   //Clock
reg [7:0] store [255:0]; //Memory

  initial
        begin
                clk = 1;

        end
always #5 clk = ~clk;                          //Generating a clock of period 5
initial begin             //Inputs ; They get asssigned during rising edge of clock

        x = 98765483280;
        @(posedge clk)
#10
        x = 1234567890;
        @(posedge clk)
#10
        x = 32'b11000001111001010111001101100001;
        @(posedge clk)
#10
      x = 98765483280;
        @(posedge clk);
      @(posedge clk);
        $finish;
end

proc r1(y,x,clk);                  //Calling ModulE
initial begin
        $monitor("Output:  %b         Input: %b",y, x);
//Monitoring the Output

end

initial begin
        $dumpfile("Project.vcd");
                //Generating .vcd File
        $dumpvars;
end
endmodule
```

Block Diagram For Execution of Pipeline Processor



INSTRUCTIONS FOR EXECUTING CODE

The module takes a input executes it and produces output. The output is being stored in a memory.
So, we have to give one input at each clock cycle and output generated is printed .

Go to the directory in which code is present.
Open terminal in that directory,
type: iverilog -o sim fif_tb.v proc.v
then for simulation type : vvp sim.

Submitted by:

Akash Tadwai - ES18BTECH11019

Vamshi T - ES18BTECH11021