CA Lab: Lab 10

student: Dimitri Tabatadze

June 2, 2023

Task Description

Implement memory module with E and I registers and a PC.

Solution

The module:

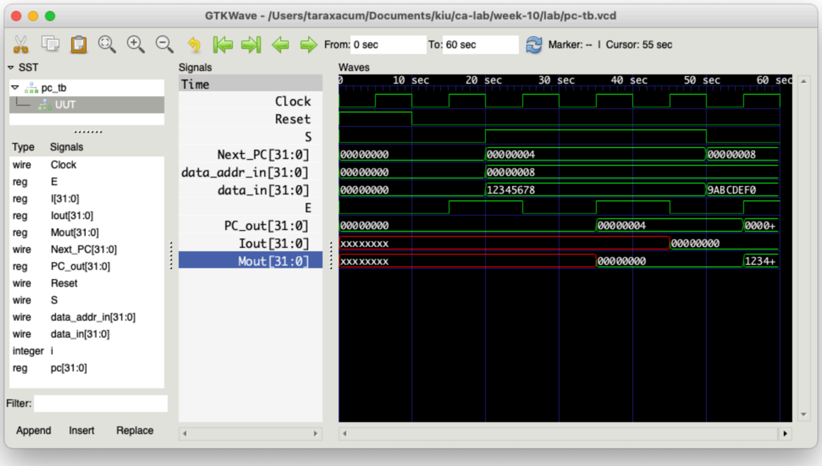
```verilog
module PC_module (
    input Clock, Reset, S,
    input [31:0] Next_PC, data_addr_in, data_in,
    output [31:0] PC_out,
    output reg [31:0] Iout, Mout,
    output reg E
);
    reg [31:0] pc;
    reg [31:0] I;
    reg [31:0] memory [0:255];

    integer i;
    initial begin
        for (i = 0; i < 32; i = i + 1) memory[1] = 32'h0;
    end

    always @(posedge Reset or posedge Clock) begin
        if (Reset) begin
            E = 0;
            pc = 0;
        end else if (!E) begin
            E = 1;
            pc = Next_PC;
            I = memory[pc >> 2];
            Mout = I;
        end else begin
            E = 0;
            if (S) begin
                memory[data_addr_in >> 2] = data_in;
            end else begin
                Mout = memory[data_addr_in >> 2];
            end
            Iout = I;
        end
    end

    assign PC_out = pc;

endmodule
```

The testbench:

```
module pc_tb();
  reg Clock;
  reg Reset;
  reg S;
  reg [31:0] Next_PC, data_addr_in, data_in;
  wire E;
  wire [31:0] PC_out, Iout, Mout;

  PC_module UUT (
    .Clock(Clock),
    .Reset(Reset),
    .S(S),
    .Next_PC(Next_PC),
    .data_addr_in(data_addr_in),
    .data_in(data_in),
    .E(E),
    .PC_out(PC_out),
    .Iout(Iout),
    .Mout(Mout)
  );

  always #5 Clock = ~Clock;

  initial begin
    $dumpfile("pc-tb.vcd");
    $dumpvars(0, pc_tb);
    Clock = 0;
    Reset = 1;
    S = 0;
    Next_PC = 32'h0;
    data_addr_in = 32'h0;
    data_in = 32'h0;

    #10;
    Reset = 0;

    #10;
    Next_PC = 32'h4;
    S = 1;
    data_addr_in = 32'h8;
    data_in = 32'h12345678;

    #30;
    Next_PC = 32'h8;
    S = 0;
    data_addr_in = 32'h8;
    data_in = 32'h9abcdef0;

    #10;
    $finish;
  end
endmodule
```

Simulation and Testing

The timing diagrams:



GTKWave - /Users/taraxacum/Documents/kiu/ca-lab/week-10/lab/pc-tb.vcd

From: 0 sec To: 60 sec Marker: -- | Cursor: 55 sec

Type	Signals
wire	Clock
reg	E
reg	I[31:0]
reg	Iout[31:0]
reg	Mout[31:0]
wire	Next_PC[31:0]
reg	PC_out[31:0]
wire	Reset
wire	S
wire	data_addr_in[31:0]
wire	data_in[31:0]
integer	i
reg	pc[31:0]

Reference

- me