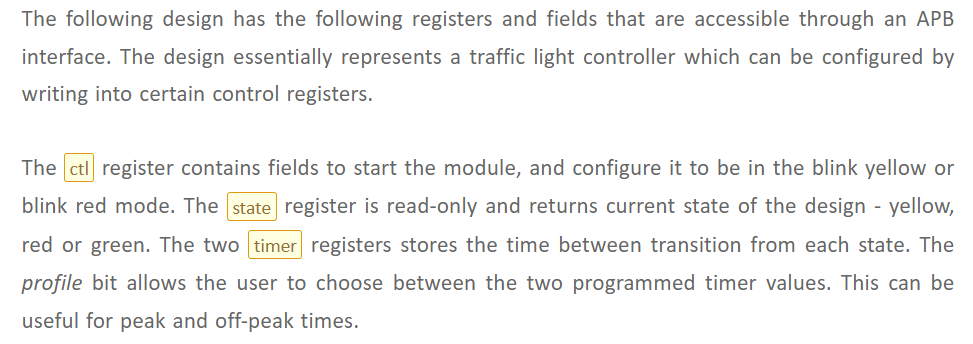
//design description



A screenshot of a computer

Description automatically generated

**Access type of register:-** Only stat reg is readonly(RO), ctl ,timer0 and timer1 registers are RW

**Design Code:-**

module traffic ( input pclk,

input presetn,

input [31:0] paddr,

input [31:0] pwdata,

input psel,

input pwrite,

input penable,

// Outputs

output [31:0] prdata);

reg [3:0] ctl\_reg; // profile, blink\_red, blink\_yellow, mod\_en RW

reg [1:0] stat\_reg; // state[1:0]

reg [31:0] timer\_0; // timer\_g2y[31:20], timer\_r2g[19:8], timer\_y2r[7:0] RW

reg [31:0] timer\_1; // timer\_g2y[31:20], timer\_r2g[19:8], timer\_y2r[7:0] RW

reg [31:0] data\_in;

reg [31:0] rdata\_tmp;

// Set all registers to default values

always @ (posedge pclk) begin

if (!presetn) begin

data\_in <= 0;

ctl\_reg <= 0;

stat\_reg <= 0;

timer\_0 <= 32'hcafe\_1234;

timer\_1 <= 32'hface\_5678;

end

end

// Capture write data

always @ (posedge pclk) begin

if (presetn & psel & penable)

if (pwrite)

case (paddr)

'h0 : ctl\_reg <= pwdata;

'h4 : timer\_0 <= pwdata;

'h8 : timer\_1 <= pwdata;

'hc : stat\_reg <= pwdata;

endcase

end

// Provide read data

always @ (penable) begin

if (psel & !pwrite)

case (paddr)

'h0 : rdata\_tmp <= ctl\_reg;

'h4 : rdata\_tmp <= timer\_0;

'h8 : rdata\_tmp <= timer\_1;

'hc : rdata\_tmp <= stat\_reg;

endcase

end

assign prdata = (psel & penable & !pwrite) ? rdata\_tmp : 'hz;

endmodule