

```
`include "timescale.v"
```

```
module wb_aes_controller(clk,reset,wb_stb_i,wb_dat_o,wb_dat_i,wb_ack_o,  
    wb_adr_i,wb_we_i,wb_cyc_i,wb_sel_i,  
    load_o,decrypt_o,ready_i,data_o,key_o,data_i);
```

```
input    clk;
```

```
input    reset;
```

```
input    wb_stb_i;
```

```
output [31:0] wb_dat_o;
```

```
input [31:0] wb_dat_i;
```

```
output    wb_ack_o;
```

```
input [31:0] wb_adr_i;
```

```
input    wb_we_i;
```

```
input    wb_cyc_i;
```

```
input [3:0] wb_sel_i;
```

```
output    load_o;
```

```
output    decrypt_o;
```

```
output [127:0] data_o;
```

```
output [127:0] key_o;
```

```
input [127:0] data_i;
```

```
input    ready_i;
```

```
reg [31:0] wb_dat_o;
```

```
reg    wb_ack_o;
```

```
reg [127:0] data_o;
```

```
reg [127:0] key_o;
```

```

wire    load_o;

wire    decrypt_o;


reg [31:0] control_reg;
reg [127:0] cypher_data_reg;


assign load_o = control_reg[0];
assign decrypt_o = control_reg[2];


always @(posedge clk or negedge reset)
begin
    if(!reset)
    begin
        wb_ack_o<=#1 0;
        wb_dat_o<=#1 0;
        control_reg <= #1 32'h0;
        cypher_data_reg <= #1 127'h0;
        key_o <= #1 127'h0;
        data_o <= #1 127'h0;
    end
    else
    begin
        if(ready_i)
        begin
            control_reg[1] <= #1 1'b1;
            cypher_data_reg <= #1 data_i;
        end

        if(wb_stb_i && wb_cyc_i && wb_we_i && ~wb_ack_o)

```

```

begin
wb_ack_o<=#1 1;
case(wb_adr_i[7:0])
  8'h0:
    begin
      //Writing control register
      control_reg<= #1 wb_dat_i;
    end
  8'h4:
    begin
      data_o[127:96]<= #1 wb_dat_i;
    end
  8'h8:
    begin
      data_o[95:64]<= #1 wb_dat_i;
    end
  8'hC:
    begin
      data_o[63:32]<= #1 wb_dat_i;
    end
  8'h10:
    begin
      data_o[31:0]<= #1 wb_dat_i;
    end
  8'h14:
    begin
      key_o[127:96]<= #1 wb_dat_i;
    end
  8'h18:

```

```

begin
    key_o[95:64]<= #1 wb_dat_i;
end

8'h1C:
begin
    key_o[63:32]<= #1 wb_dat_i;
end

8'h20:
begin
    key_o[31:0]<= #1 wb_dat_i;
end
endcase
end

else if(wb_stb_i && wb_cyc_i && ~wb_we_i && ~wb_ack_o)
begin
    wb_ack_o<=#1 1;
    case(wb_adr_i[7:0])
        8'h0:
            begin
                wb_dat_o<= #1 control_reg;
                control_reg[1]<=1'b0;
            end
        8'h24:
            begin
                wb_dat_o<= #1 cypher_data_reg[127:96];
            end
        8'h28:
            begin
                wb_dat_o<= #1 cypher_data_reg[95:64];
            end
    endcase
end

```

```

        end

        8'h2C:

        begin
            wb_dat_o<= #1 cypher_data_reg[63:32];
        end

        8'h30:

        begin
            wb_dat_o<= #1 cypher_data_reg[31:0];
        end

        endcase

    end

    else

    begin

        wb_ack_o<=#1 0;

        control_reg[0]<= #1 1'b0;

    end

    end

end

endmodule

```