RTL 설계교육

(2주차)

2022.01.28

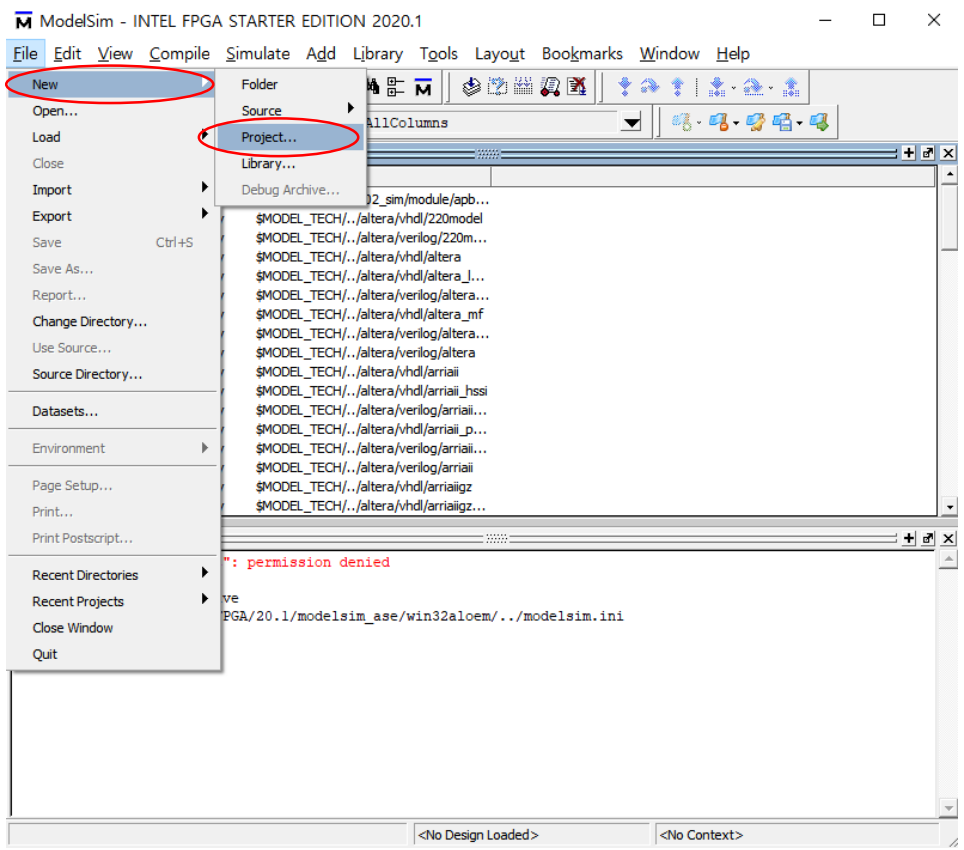
CoAsia SEMI Ltd.

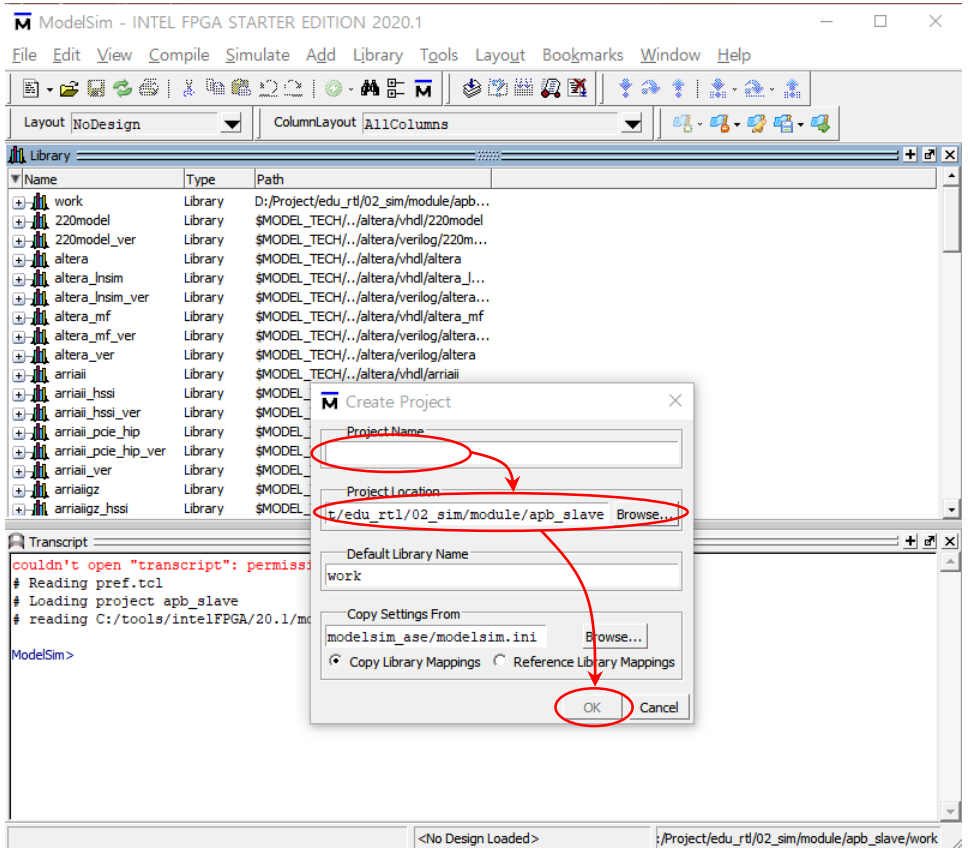
Index

1. Modelsim 사용법
2. EDA Playground 사용법
3. APB Slave Reivew
4. Module 설계 (reg_rw)
5. Module 설계 (sram_rw)
6. Module 설계 (stopwatch)

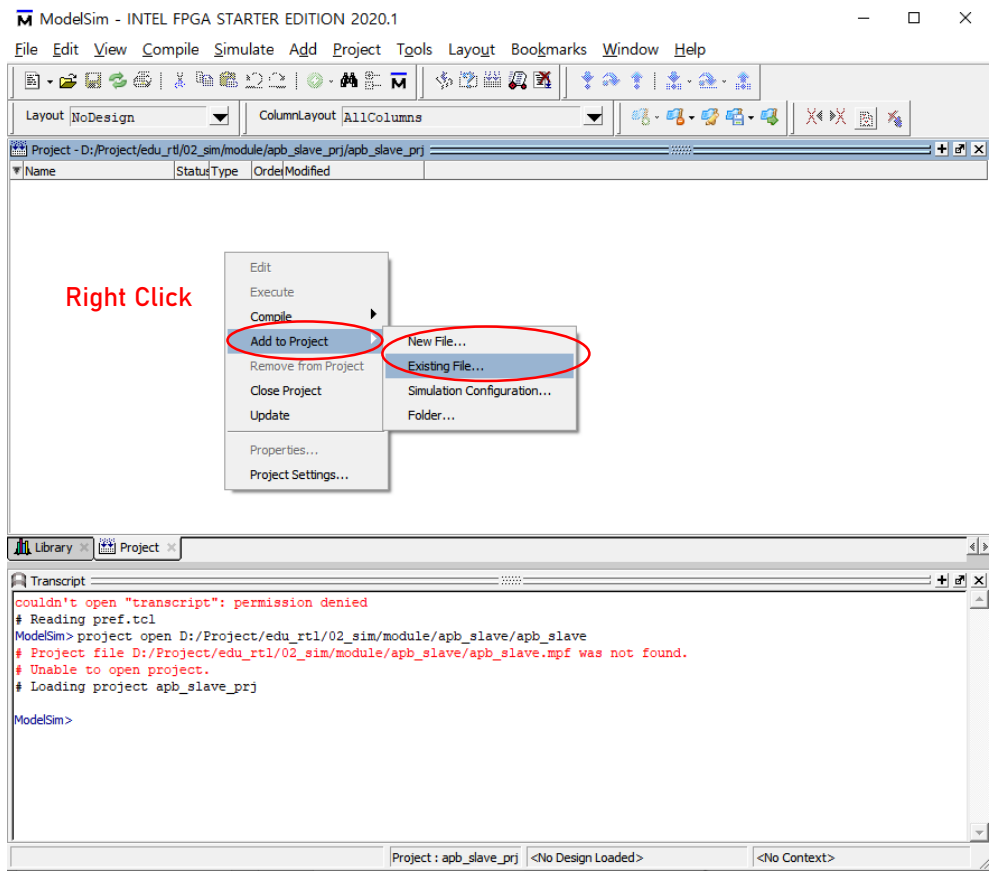
1. Modelsim 사용법

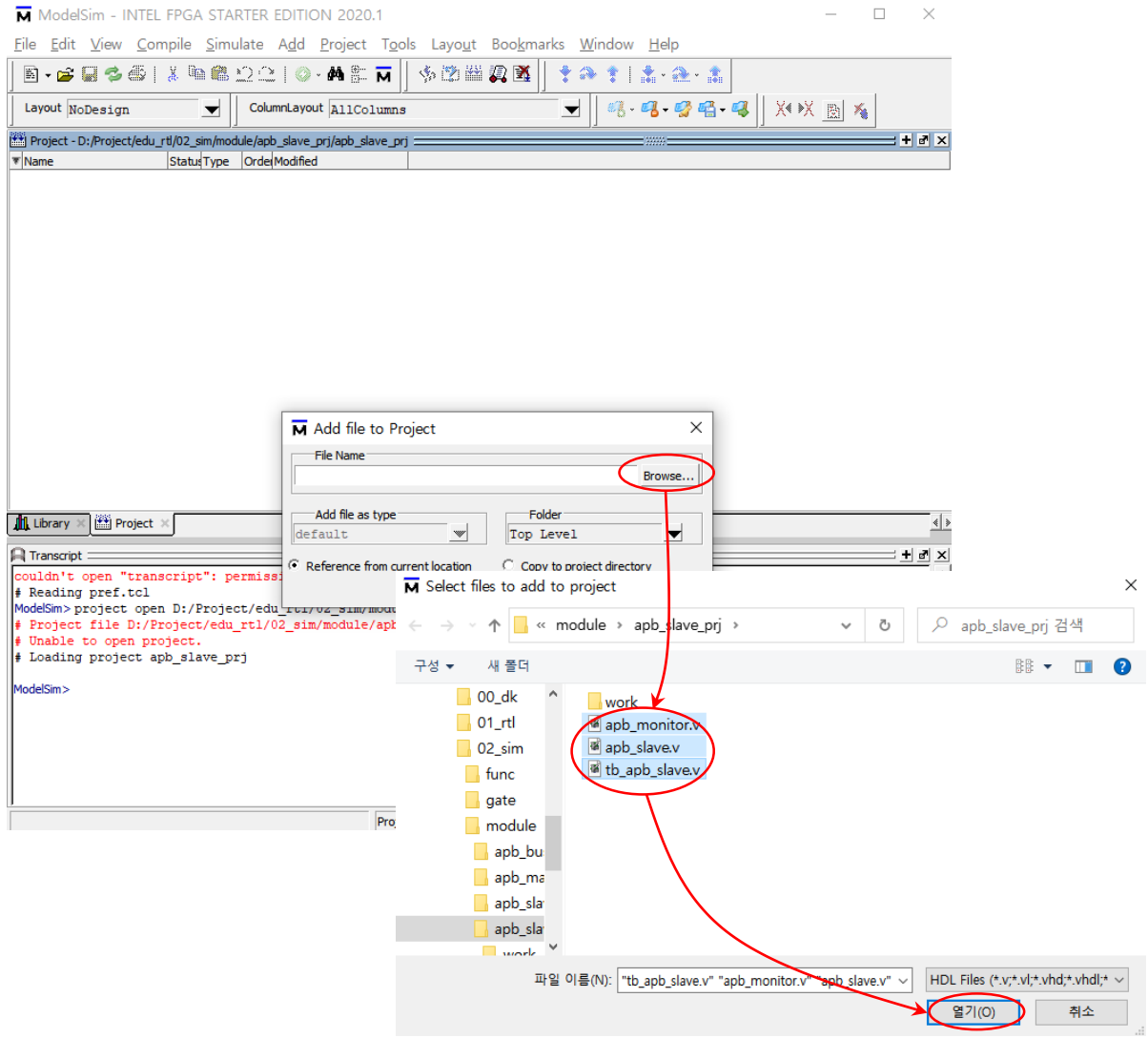
Project 생성시



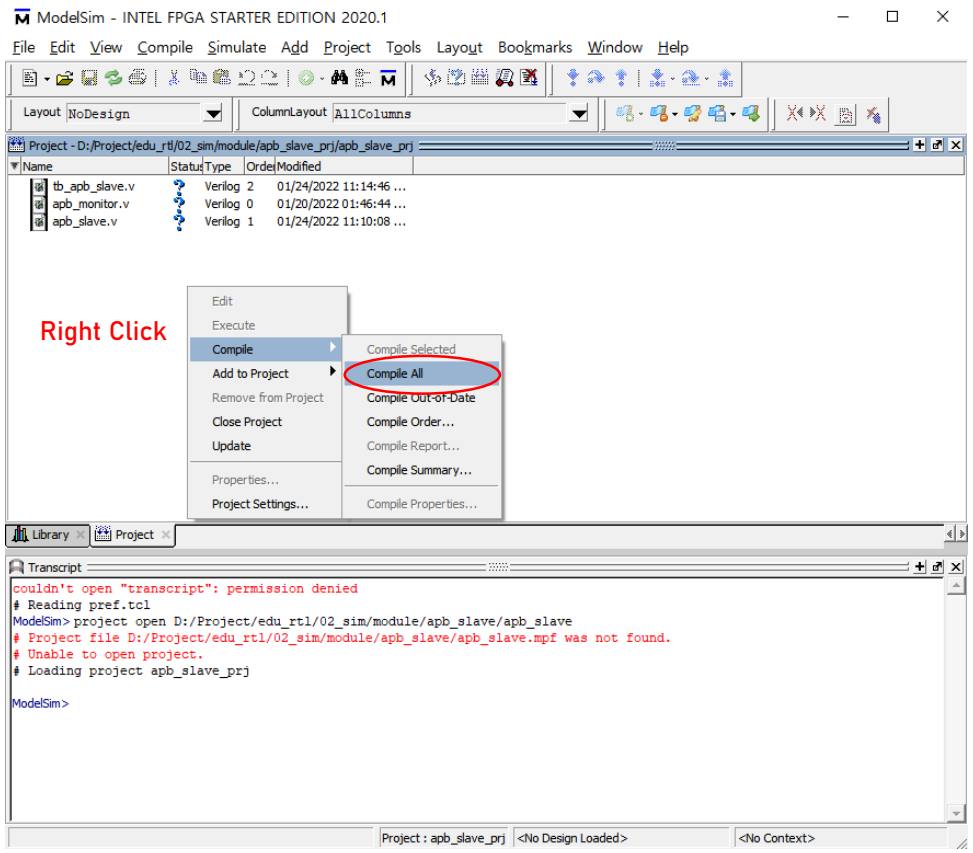


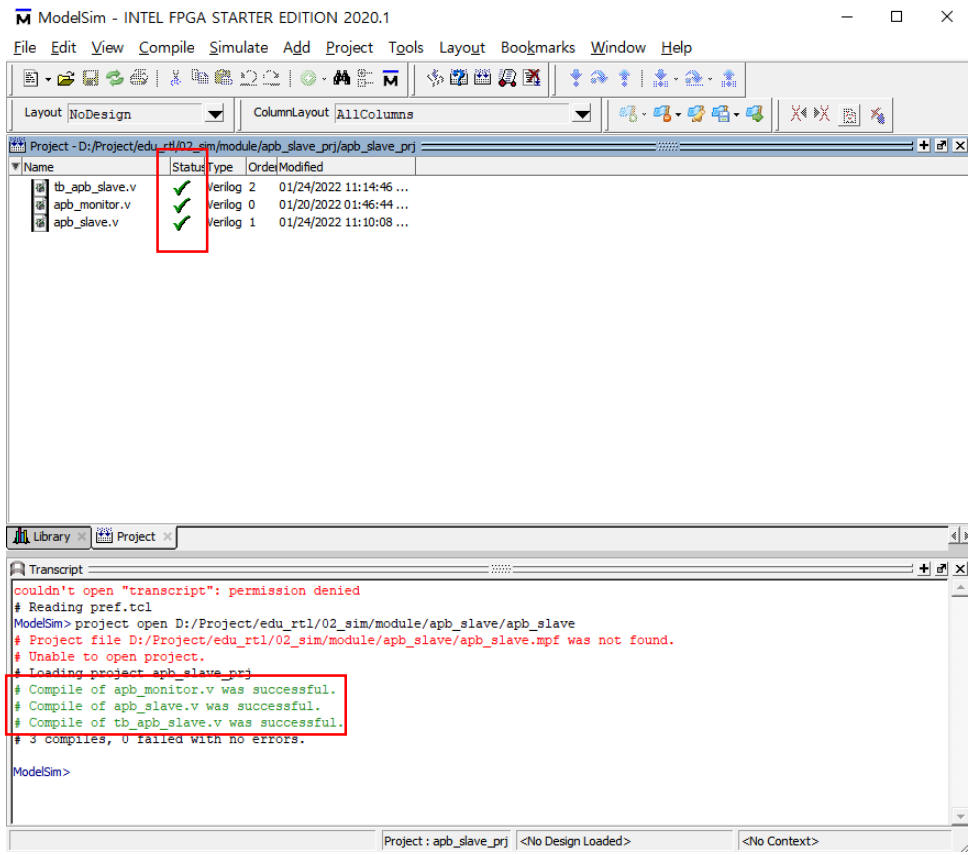
1. Modelsim 사용법



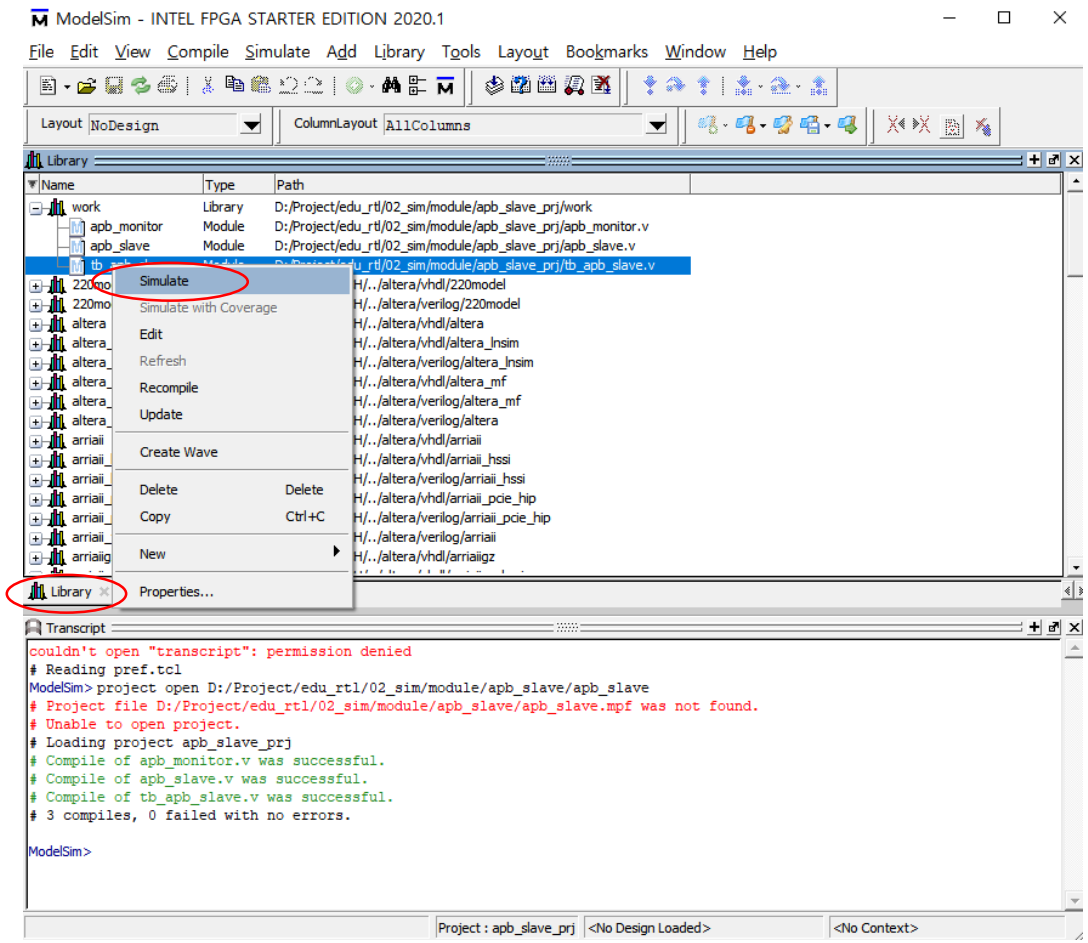


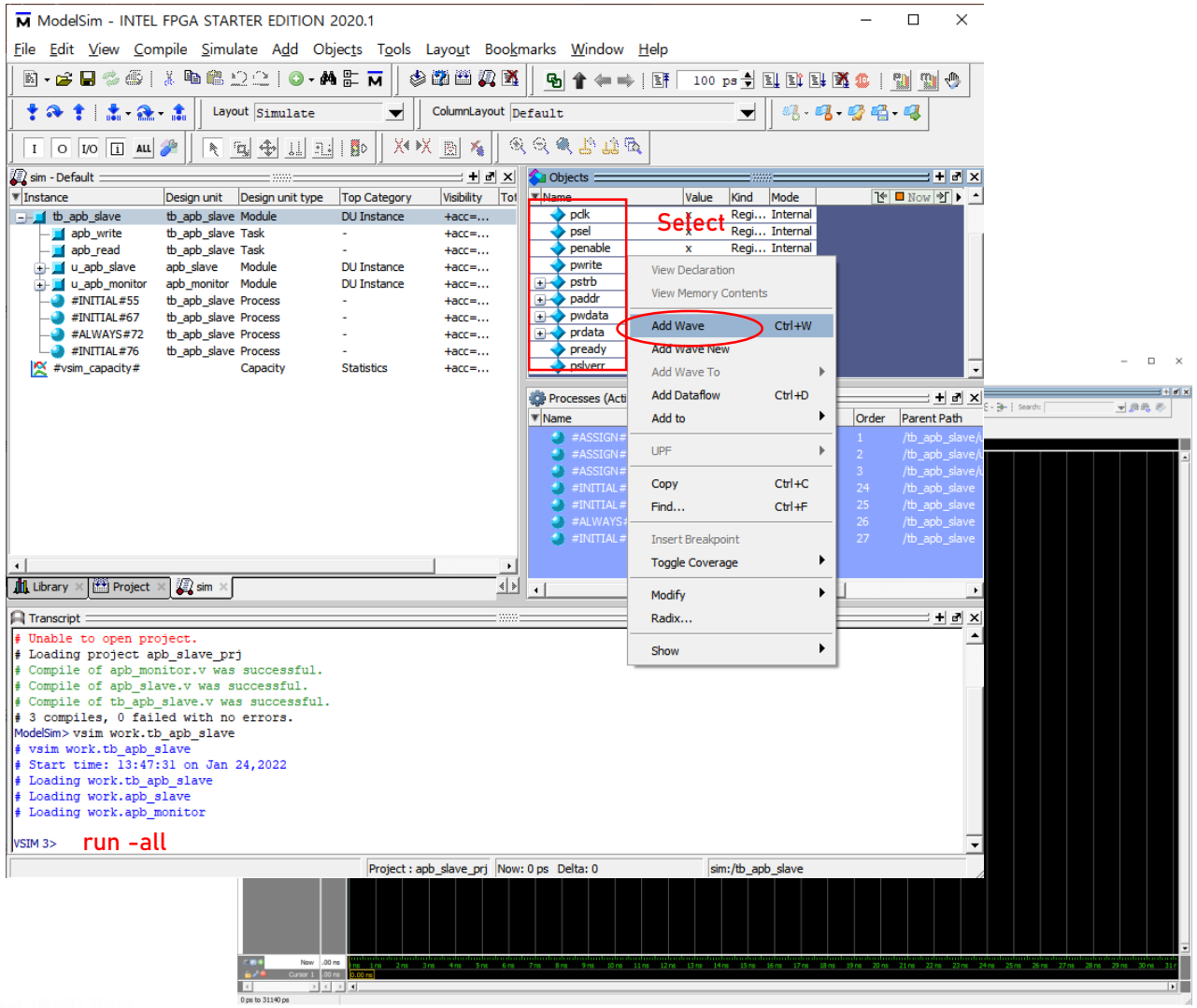
1. Modelsim 사용법



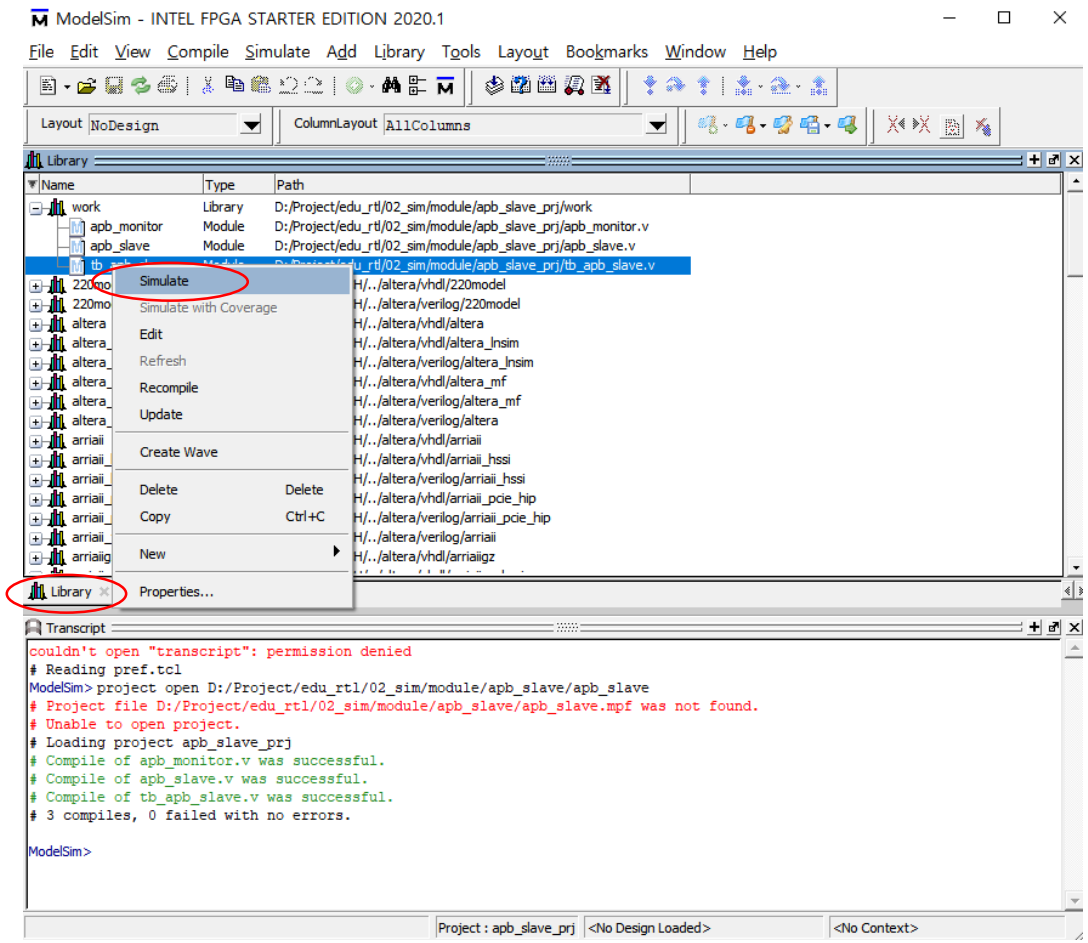


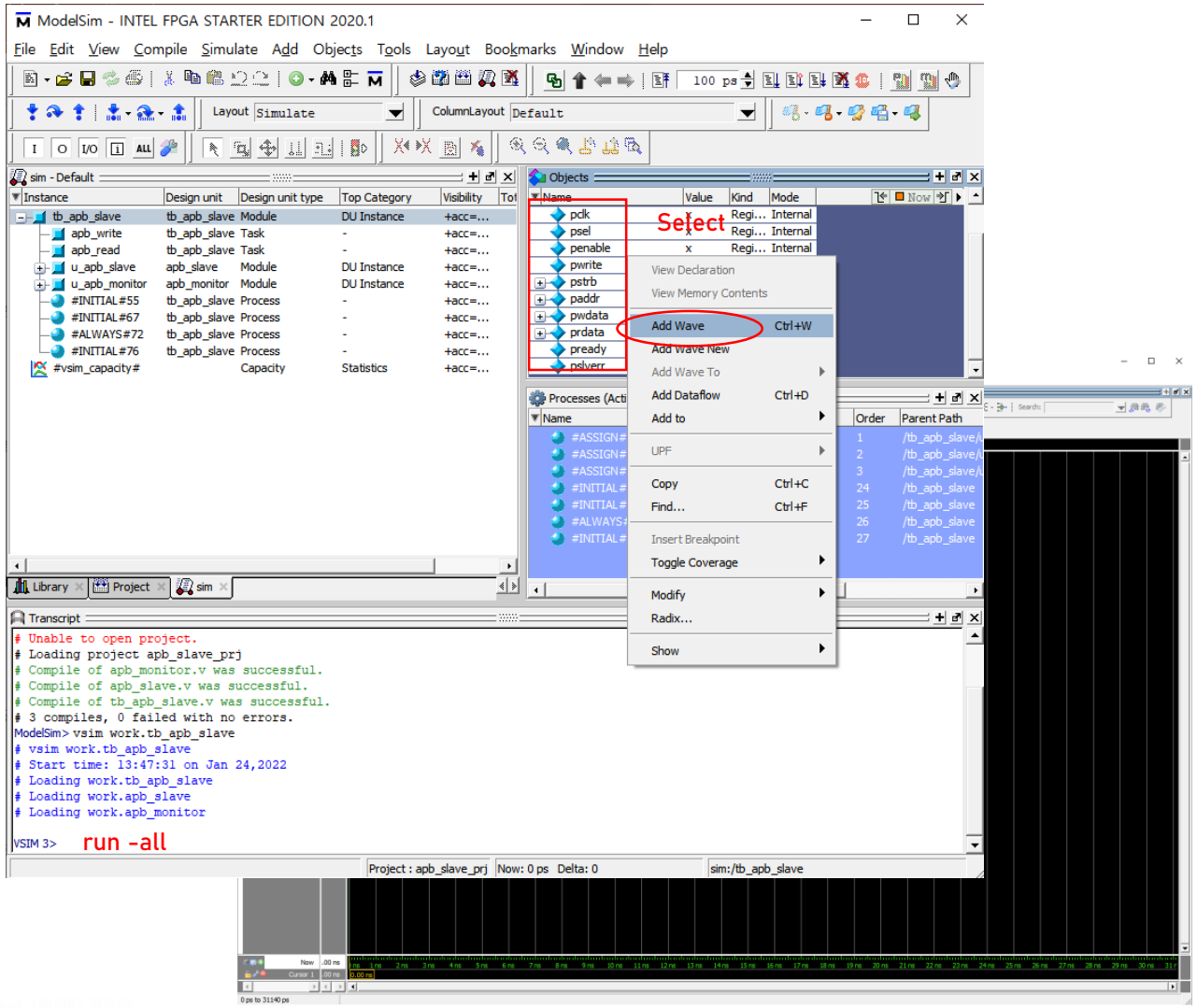
1. Modelsim 사용법





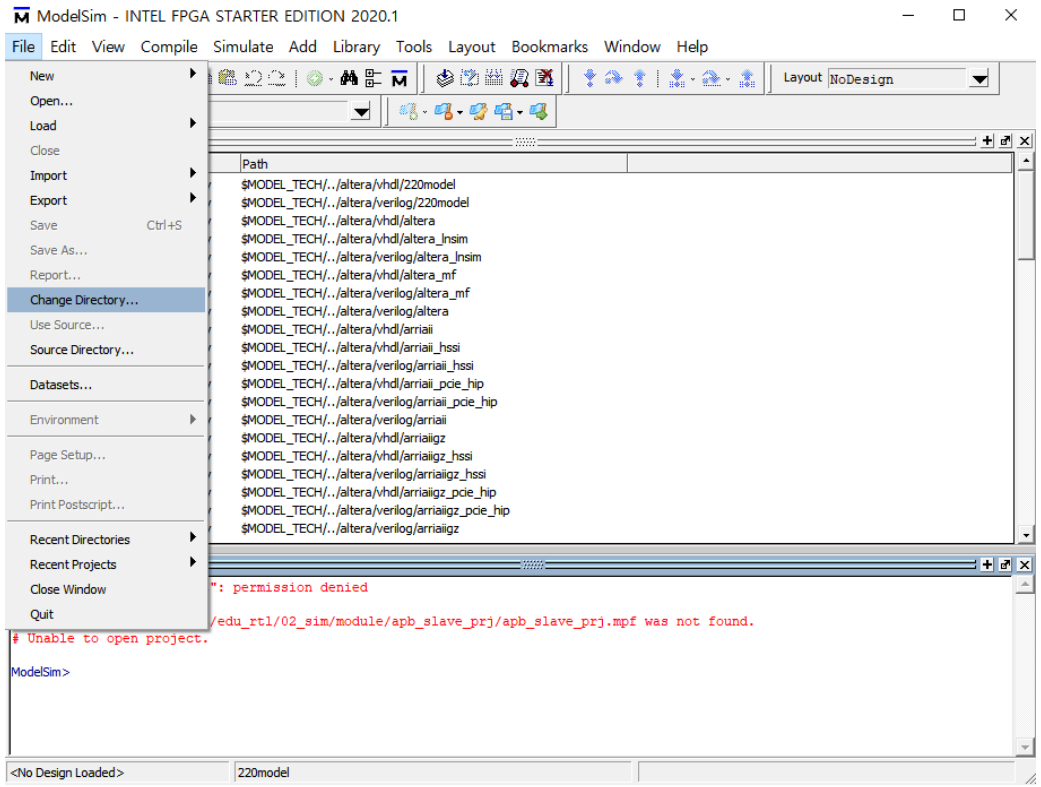
1. Modelsim 사용법

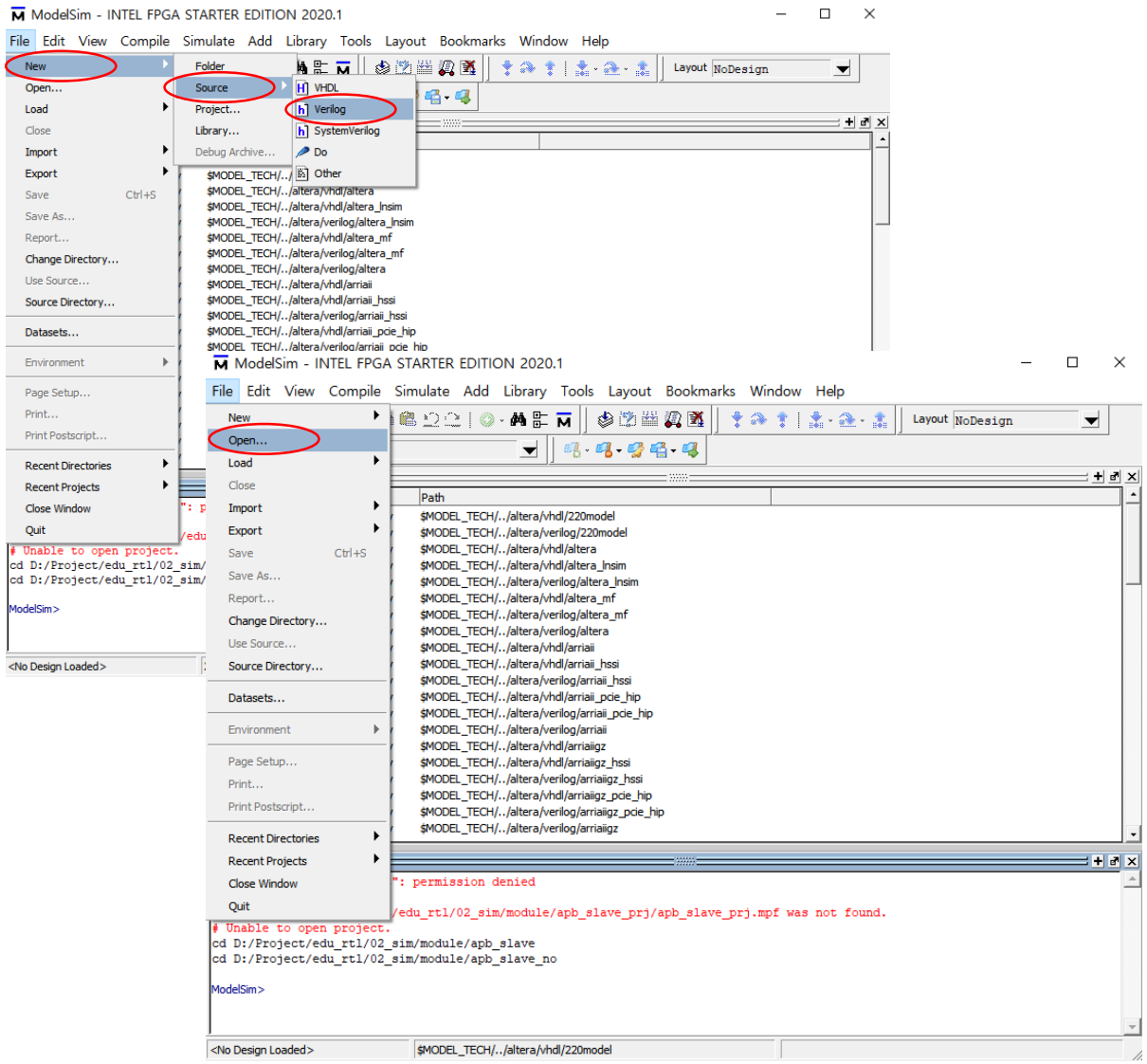




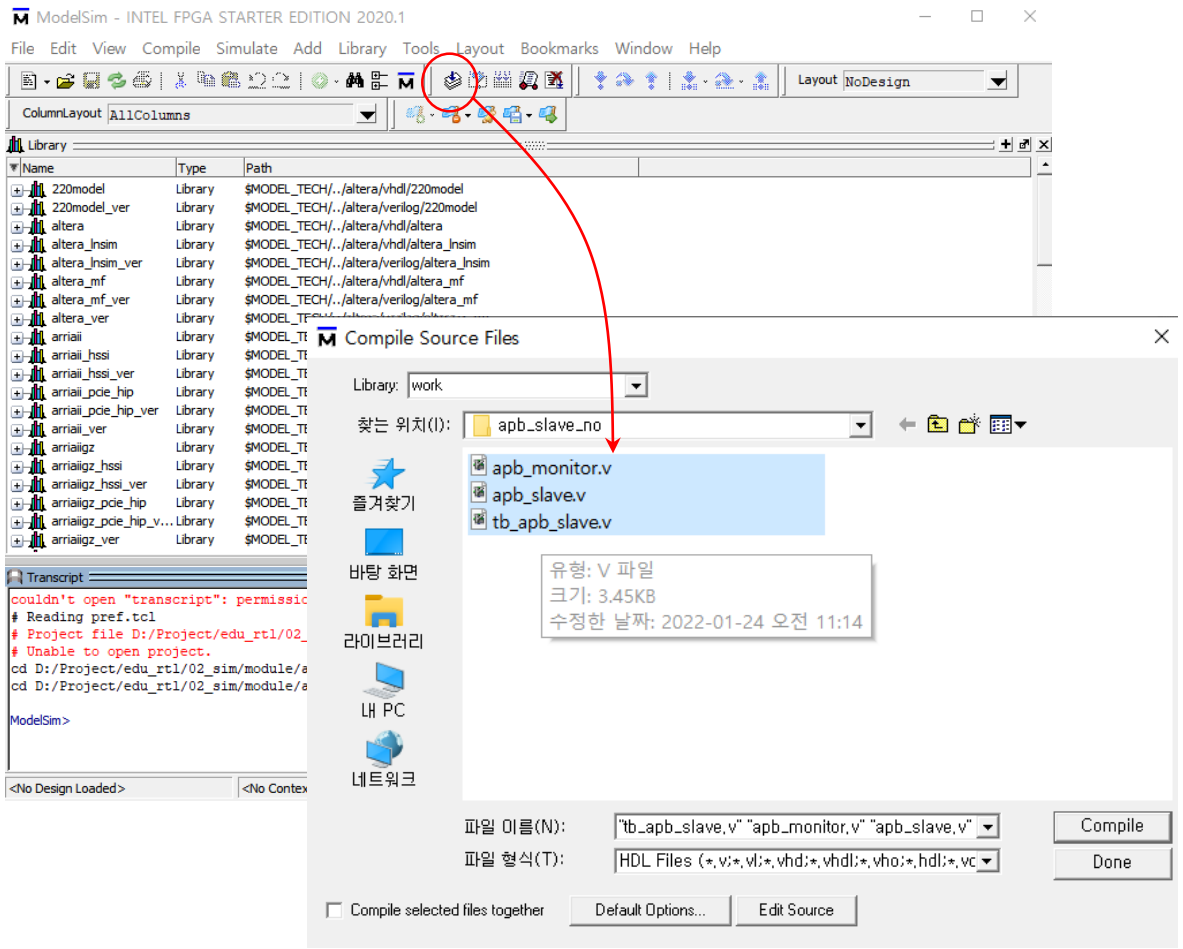
1. Modelsim 사용법

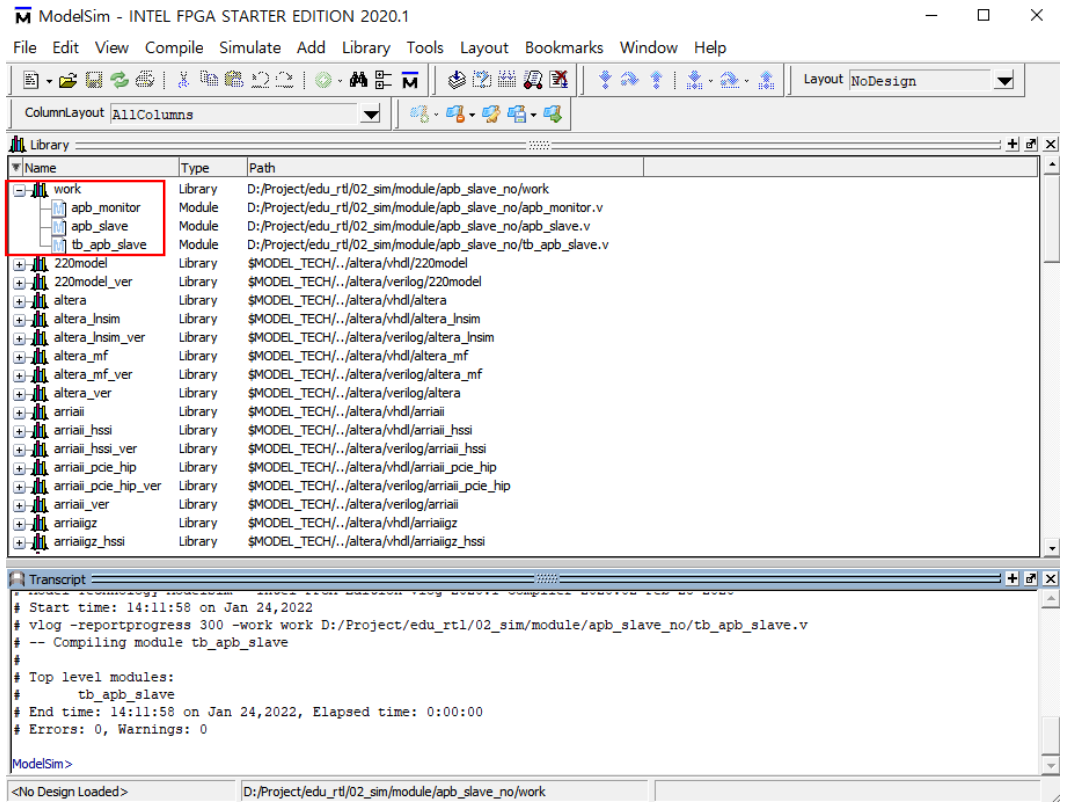
Project 미생성시





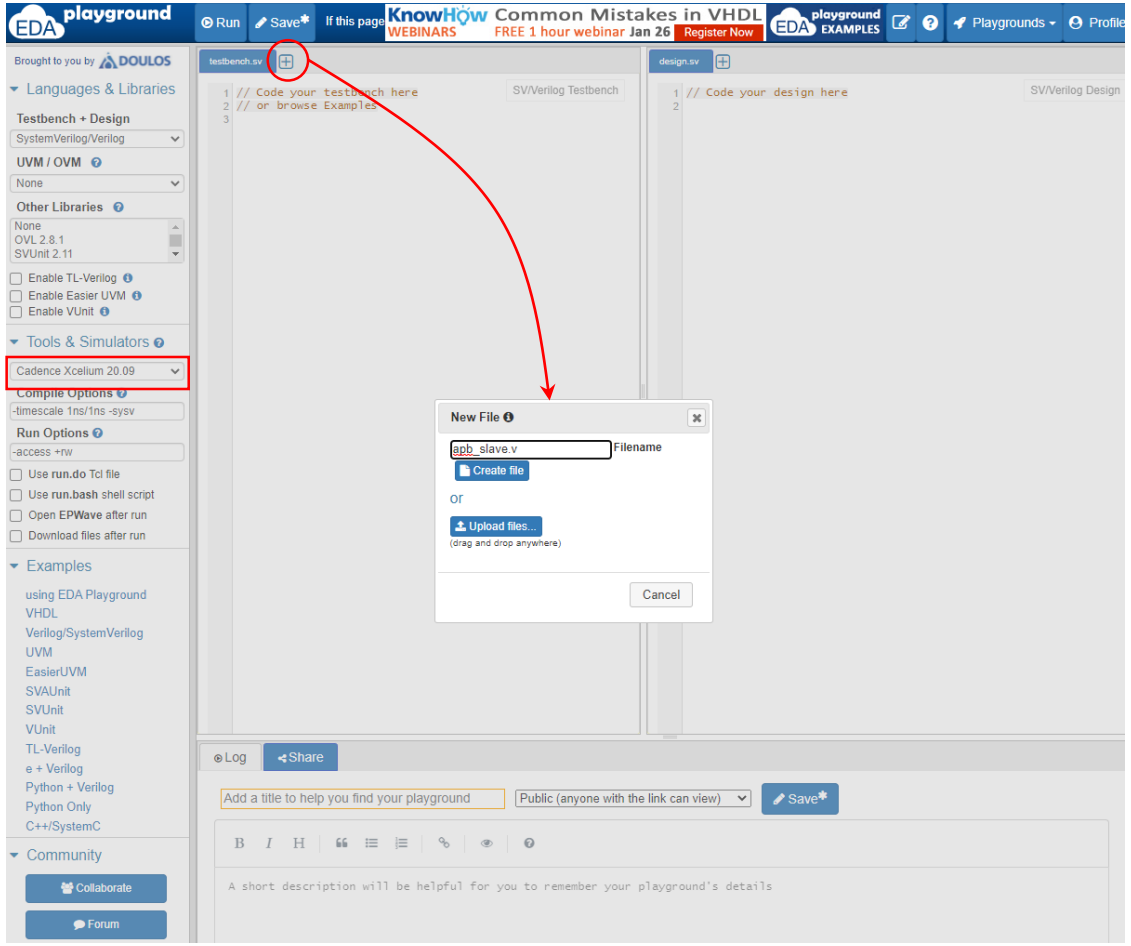
1. Modelsim 사용법

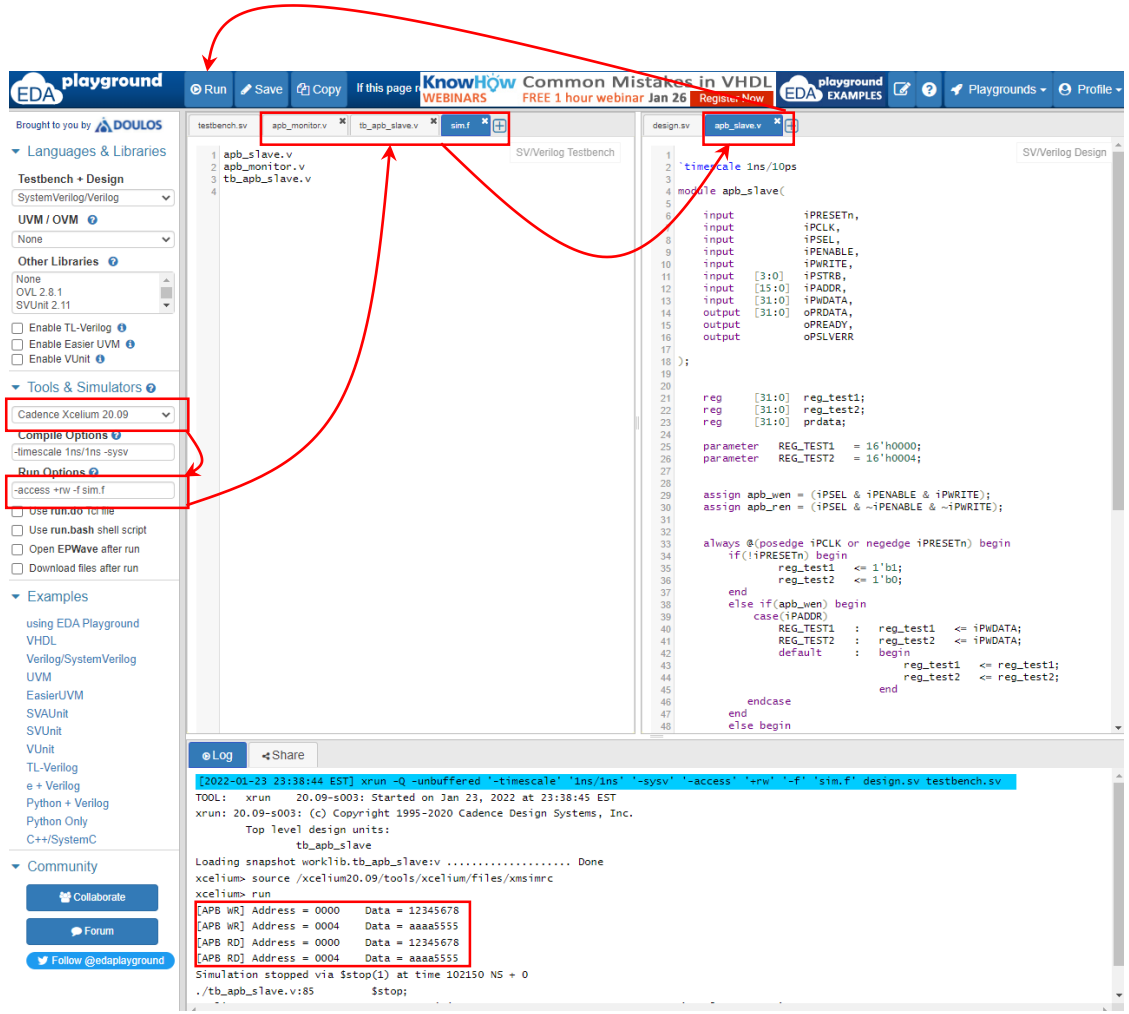




이후 과정은 Project 생성시와 같음

2. EDA Playground 사용법





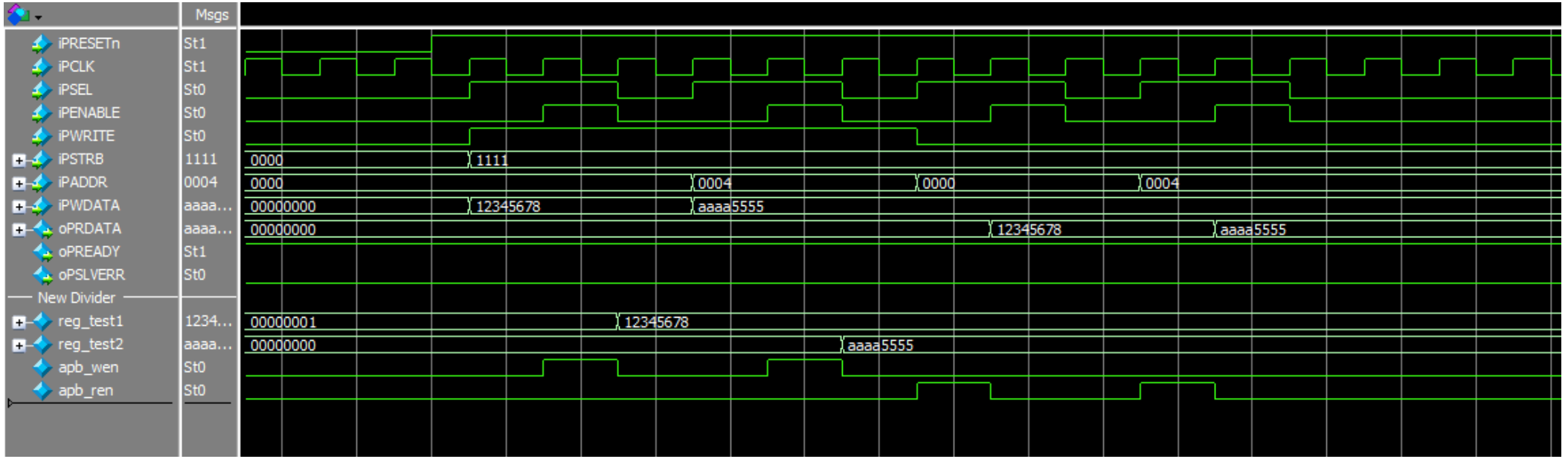
3. APB Slave

Review

```
`timescale 1ns/10ps

module apb_slave(

    input           iPRESETn,
    input           iPCLK,
    input           iPSEL,
    input           iPENABLE,
    input           iPWRITE,
    input   [3:0]   iPSTRB,
    input   [15:0]  iPADDR,
    input   [31:0]  iPWDATA,
    output  [31:0]  oPRDATA,
    output          oPREADY,
    output          oPSLVERR

);


    reg     [31:0]  reg_test1;
    reg     [31:0]  reg_test2;
    reg     [31:0]  prdata;

    parameter   REG_TEST1   = 16'h0000;
    parameter   REG_TEST2   = 16'h0004;


    assign apb_wen = (iPSEL & iPENABLE & iPWRITE);
    assign apb_ren = (iPSEL & ~iPENABLE & ~iPWRITE);
```

```
    always @(posedge iPCLK or negedge iPRESETn) begin
        if(!iPRESETn) begin
                reg_test1   <= 1'b1;
                reg_test2   <= 1'b0;
        end
        else if(apb_wen) begin
            case(iPADDR)
                REG_TEST1   :   reg_test1   <= iPWDATA;
                REG_TEST2   :   reg_test2   <= iPWDATA;
                default     :   begin
                                    reg_test1   <= reg_test1;
                                    reg_test2   <= reg_test2;
                                end
           endcase
        end
        else begin
            reg_test1   <= reg_test1;
            reg_test2   <= reg_test2;
        end
    end

    always @(posedge iPCLK or negedge iPRESETn) begin
        if(!iPRESETn)
            prdata  <= 32'h0;
        else if(apb_ren)
            case(iPADDR)
                REG_TEST1   :   prdata  <= reg_test1;
                REG_TEST2   :   prdata  <= reg_test2;
                default     :   prdata  <= 32'h0;
            endcase
        else
            prdata  <= prdata;
    end


    assign  oPRDATA     = prdata;
    assign  oPREADY     = 1;
    assign  oPSLVERR    = 0;


endmodule
```

3. APB Slave

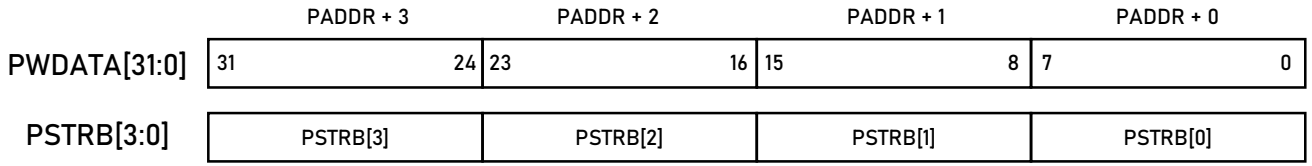



```
VSIM 7> run -all
# [APB WR] Address = 0000    Data = 12345678
# [APB WR] Address = 0004    Data = aaaa5555
# [APB RD] Address = 0000    Data = 12345678
# [APB RD] Address = 0004    Data = aaaa5555
# ** Note: $stop    : D:/Project/edu_rtl/02_sim/module/apb_slave/tb_apb_slave.v(85)
#    Time: 102150 ns  Iteration: 0  Instance: /tb_apb_slave
```

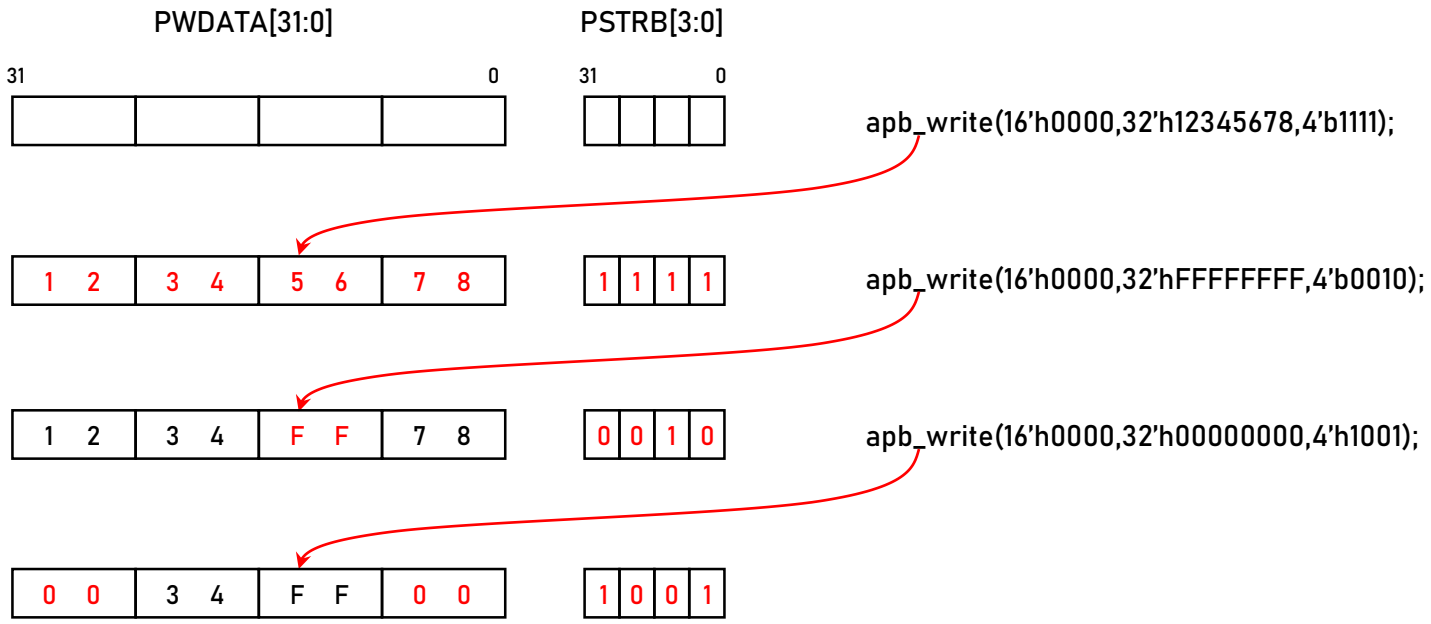
4. Module 설계 (reg_rw)

Specification

- apb_slave.v module에서 PSTRB기능 추가



- 예시



4. Module 설계 (reg_rw)

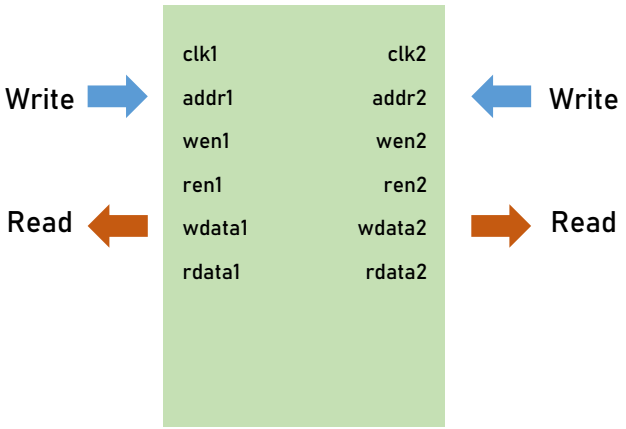
Verilog Code

```
VSIM 21> run -all
# [APB WR] Address = 0000    Data = 12345678
# [APB RD] Address = 0000    Data = 12345678
# [APB WR] Address = 0000    Data = ----ff--
# [APB RD] Address = 0000    Data = 1234ff78
# [APB WR] Address = 0000    Data = 00----00
# [APB RD] Address = 0000    Data = 0034ff00
# ** Note: $stop    : D:/Project/edu_rtl/02_sim/module/reg_rw/tb_reg_rw.v(87)
#    Time: 102750 ns  Iteration: 0  Instance: /tb_reg_rw
# Break in Module tb_reg_rw at D:/Project/edu_rtl/02_sim/module/reg_rw/tb_reg_rw.v line 87
```

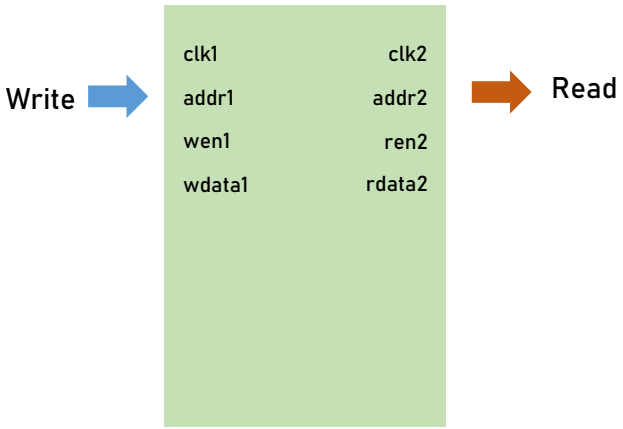
5. Module 설계 (sram_rw)

SRAM

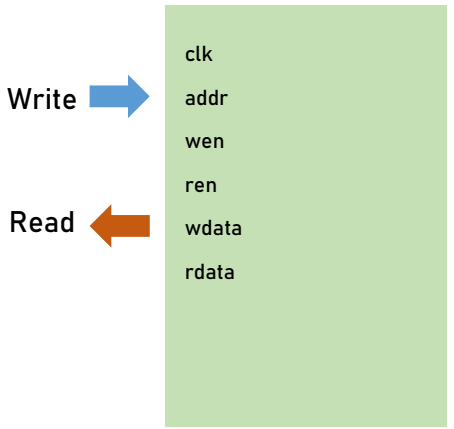
- **Dual Port SRAM**



- **Two Port SRAM**



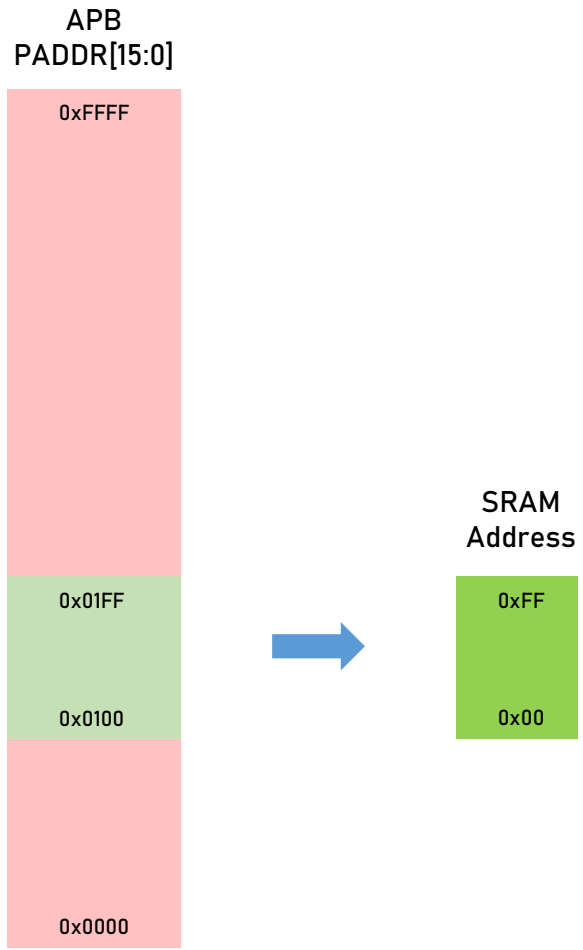
- **Single Port SRAM**



5. Module 설계 (sram_rw)

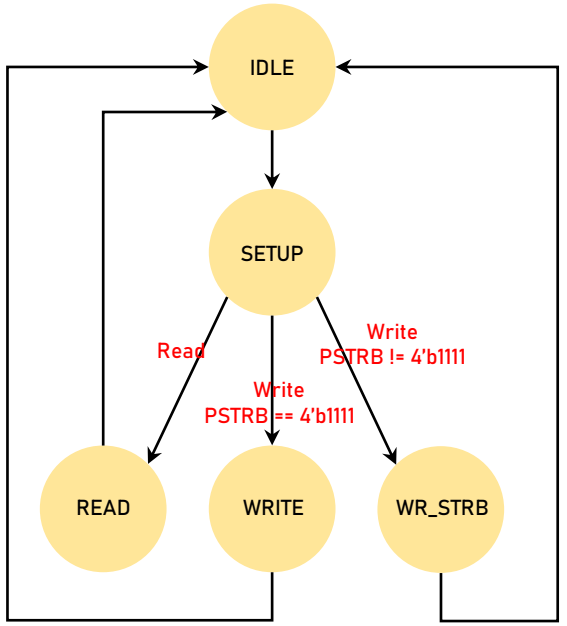
Specification

- **Address Mapping**



- **PSTRB 지원**

	PADDR + 3	PADDR + 2	PADDR + 1	PADDR + 0
PWDATA[31:0]	31 24	23 16	15 8	7 0
PSTRB[3:0]	PSTRB[3]	PSTRB[2]	PSTRB[1]	PSTRB[0]



CoAsia SEMI Ltd.

5. Module 설계 (sram_rw)

Verilog Code (Testbench)

```
`timescale 1ns/10ps

module tb_sram_rw;

    reg             presetn;
    reg             pclk;
    reg             psel;
    reg             penable;
    reg             pwrite;
    reg     [3:0]   pstrb;
    reg     [15:0]  paddr;
    reg     [31:0]  pwdata;
    wire    [31:0]  prdata;
    wire            pready;
    wire            pslverr;

    parameter       PERIOD  = (1000.0/10.0)/2;


    sram_rw             #(
    /*  parameter       */  .MEM_MSB_ADDR           (8'h10))
                        u_sram_rw(
    /*  input           */  .iPRESETn               (presetn),
    /*  input           */  .iPCLK                  (pclk),
    /*  input           */  .iPSEL                  (psel),
    /*  input           */  .iPENABLE               (penable),
    /*  input           */  .iPWRITE                (pwrite),
    /*  input   [3:0]   */  .iPSTRB                 (pstrb),
    /*  input   [15:0]  */  .iPADDR                 (paddr),
    /*  input   [31:0]  */  .iPWDATA                (pwdata),
    /*  output  [31:0]  */  .oPRDATA                (prdata),
    /*  output          */  .oPREADY                (pready),
    /*  output          */  .oPSLVERR               (pslverr));


    apb_monitor         u_apb_monitor(
    /*  input           */  .iPRESETn               (presetn),
    /*  input           */  .iPCLK                  (pclk),
    /*  input           */  .iPSEL                  (psel),
    /*  input           */  .iPENABLE               (penable),
    /*  input           */  .iPWRITE                (pwrite),
    /*  input   [3:0]   */  .iPSTRB                 (pstrb),
    /*  input   [15:0]  */  .iPADDR                 (paddr),
    /*  input   [31:0]  */  .iPWDATA                (pwdata),
    /*  input   [31:0]  */  .iPRDATA                (prdata),
    /*  input           */  .iPREADY                (pready),
    /*  input           */  .iPSLVERR               (pslverr));


    initial begin
        presetn <= 0;
        pclk    <= 0;
        psel    <= 0;
        penable <= 0;
        pwrite  <= 0;
        pstrb   <= 0;
        paddr   <= 0;
        pwdata  <= 0;
    end
```

```
    initial begin
        #(1000)
        presetn <= 1;
    end

    always #(PERIOD) begin
        pclk    <= ~pclk;
    end

    initial begin
        wait(presetn);
        wait(1000);
        apb_write(16'h1000,32'h12345678,4'b1111);
        apb_write(16'h1004,32'hffffffff,4'b1111);
        apb_write(16'h1008,32'h11111111,4'b1111);
        apb_write(16'h100a,32'haaaaaaaa,4'b1111);
        apb_write(16'h1010,32'h55555555,4'b1111);
        apb_write(16'h1014,32'haaaa5555,4'b1111);
        apb_write(16'h1018,32'h5555aaaa,4'b1111);
        apb_read(16'h1000);
        apb_read(16'h1004);
        apb_read(16'h1008);
        apb_read(16'h100a);
        apb_read(16'h1010);
        apb_read(16'h1014);
        apb_read(16'h1018);
        #(1000)
        //$finish;
        $stop;
    end
```

```
    task apb_write;
        input   [15:0]  ts_addr;
        input   [31:0]  ts_wdata;
        input   [3:0]   ts_pstrb;

        begin
            //$display("APB Write Operation");
            @(posedge pclk) psel    <= 1;
                            pwrite  <= 1;
                            paddr   <= ts_addr;
                            pwdata  <= ts_wdata;
                            pstrb   <= ts_pstrb;
            @(posedge pclk) penable <= 1;
            @(posedge pclk)
            while(!pready)  @(posedge pclk);
            psel    <= 0;
            penable <= 0;
        end
    endtask

    task apb_read;
        input   [15:0]  ts_addr;

        begin
            //$display("APB Read Operation");
            @(posedge pclk) psel    <= 1;
                            pwrite  <= 0;
                            paddr   <= ts_addr;
            @(posedge pclk) penable <= 1;
            @(posedge pclk)
            while(!pready)  @(posedge pclk);
            psel    <= 0;
            penable <= 0;
        end
    endtask


endmodule
```

5. Module 설계 (sram_rw)

Verilog Code

```
[APB WR] Address = 1000    Data = 12345678
[APB WR] Address = 1004    Data = ffffffff
[APB WR] Address = 1008    Data = 11111111
[APB WR] Address = 100a    Data = aaaaaaaa
[APB WR] Address = 1010    Data = 55555555
[APB WR] Address = 1014    Data = aaaa5555
[APB WR] Address = 1018    Data = 5555aaaa
[APB RD] Address = 1000    Data = 12345678
[APB RD] Address = 1004    Data = ffffffff
[APB RD] Address = 1008    Data = 11111111
[APB RD] Address = 100a    Data = aaaaaaaa
[APB RD] Address = 1010    Data = 55555555
[APB RD] Address = 1014    Data = aaaa5555
[APB RD] Address = 1018    Data = 5555aaaa
** Note: $stop    : D:/Project/edu_rtl/02_sim/module/sram/tb_sram_rw.v(91)
   Time: 6150 ns  Iteration: 0  Instance: /tb_sram_rw
Break in Module tb_sram_rw at D:/Project/edu_rtl/02_sim/module/sram/tb_sram_rw.v line 91
```

5. Module 설계 (sram_rw)

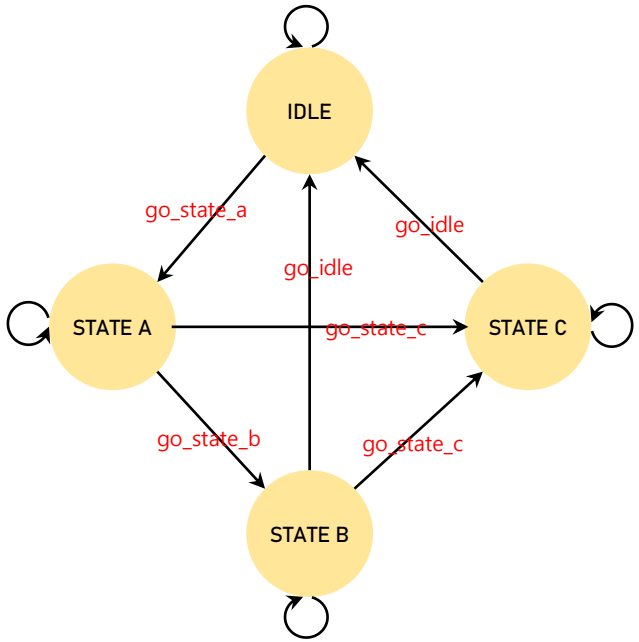
Verilog Code (PSTRB 지원)

```
# [APB WR] Address = 10f0    Data = 12345678
# [APB WR] Address = 10f0    Data = ----ff--
# [APB RD] Address = 10f0    Data = 1234ff78
# ** Note: $stop    : D:/Project/edu_rtl/02_sim/module/sram/tb_sram_rw.v(110)
#    Time: 2850 ns  Iteration: 0  Instance: /tb_sram_rw
# Break in Module tb_sram_rw at D:/Project/edu_rtl/02_sim/module/sram/tb_sram_rw.v line 110
```

6. Module 설계 (stopwatch)

State Machine

- State Machine



- State Machine Sample Code

```
parameter   IDLE    = 2'b00,
            STATE_A = 2'b01,
            STATE_B = 2'b10,
            STATE_C = 2'b11;


always @(*) begin
    case(curr_state)
        IDLE    :   begin
                        if(go_state_a)
                            next_state  <= STATE_A;
                        else
                            next_state  <= IDLE;
                    end
        STATE_A :   begin
                        if(go_state_b)
                            next_state  <= STATE_B;
                        else if(go_state_c)
                            next_state  <= STATE_C;
                        else
                            next_state  <= STATE_A;
                    end
        STATE_B :   begin
                        if(go_state_c)
                            next_state  <= STATE_C;
                        else if(go_idle)
                            next_state  <= IDLE;
                        else
                            next_state  <= STATE_B;
                    end
        STATE_C :   begin
                        if(go_idle)
                            next_state  <= IDLE;
                        else
                            next_state  <= STATE_C;
                    end
        default :   next_state  <= IDLE;
    endcase
end
```

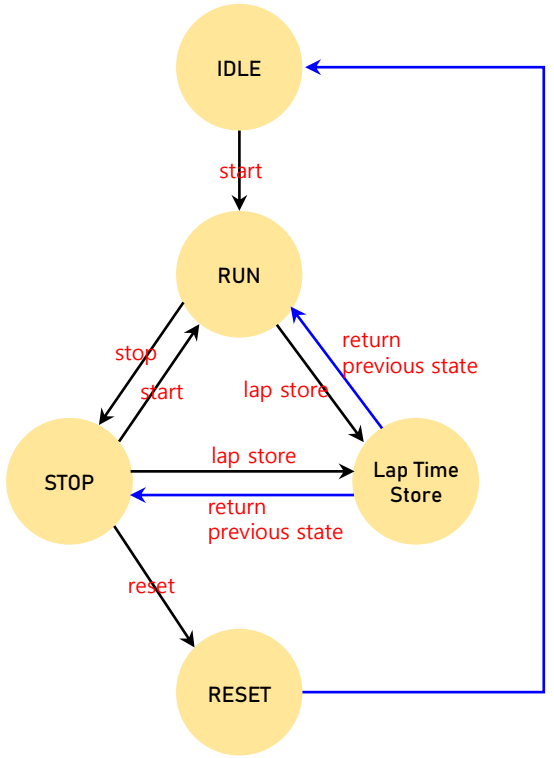
```
always @(posedge iCLK or negedge iRESETn) begin
    if(!iRESETn)
        curr_state  <= IDLE;
    else
        curr_state  <= next_state;
end

always @(posedge iCLK or negedge iRESETn) begin
    if(!iRESETn) begin
        sig_1   <= 0;
        sig_2   <= 0;
        sig_2   <= 0;
    end
    else
        case(curr_state)
            IDLE    :   begin
                            sig_1   <= 0;
                            sig_2   <= 0;
                            sig_3   <= 0;
                        end
            STATE_A :   begin
                            sig_1   <= 1;
                            sig_2   <= 0;
                            sig_3   <= 0;
                        end
            STATE_B :   begin
                            sig_1   <= 0;
                            sig_2   <= 1;
                            sig_3   <= 0;
                        end
            STATE_C :   begin
                            sig_1   <= 0;
                            sig_2   <= 0;
                            sig_3   <= 1;
                        end
            default :   begin
                            sig_1   <= sig_1;
                            sig_2   <= sig_2;
                            sig_3   <= sig_3;
                        end
        endcase
end
```

6. Module 설계 (stopwatch)

Specification

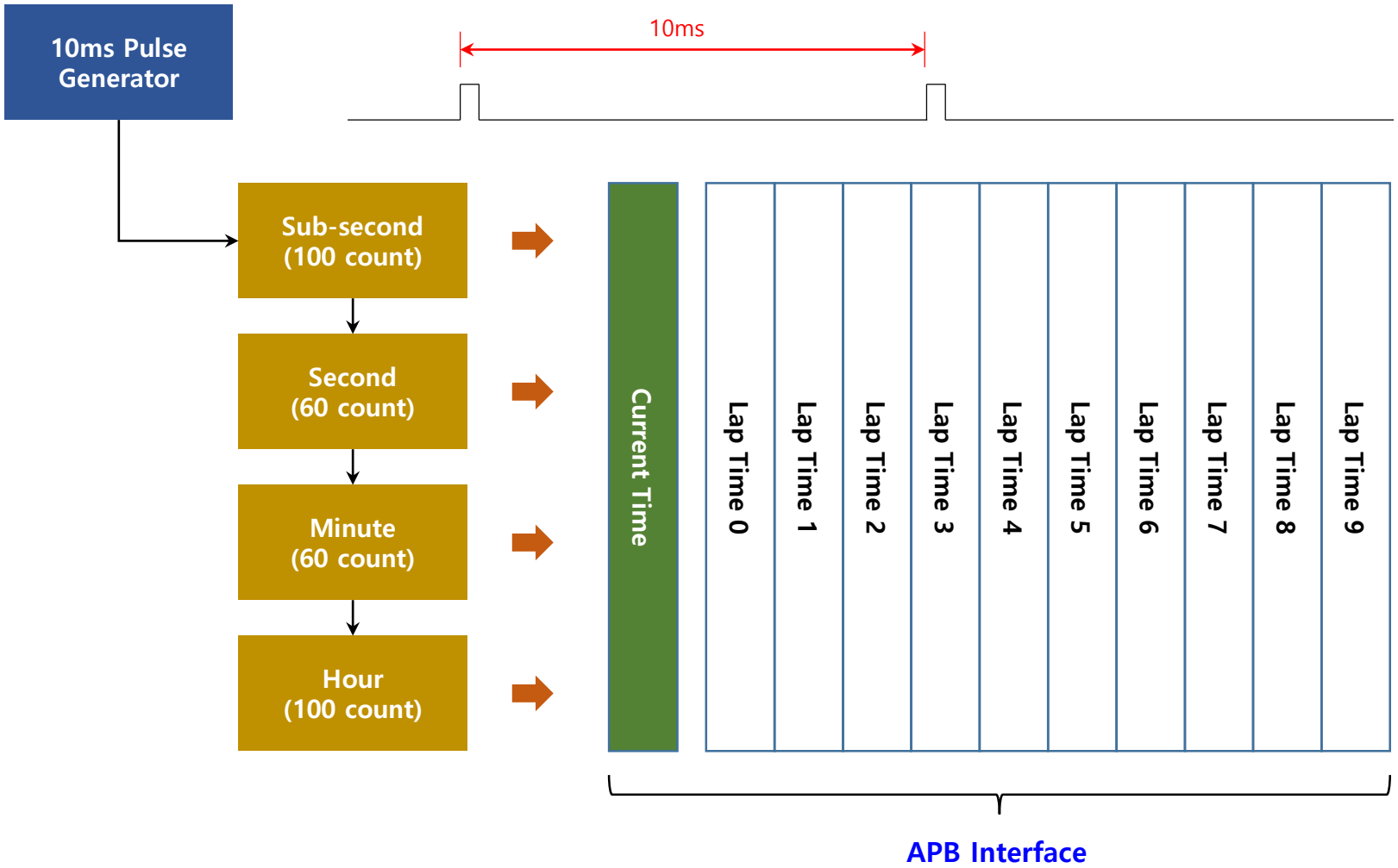
- 1/100초 분해능
- Start / Stop / Reset
- Lap Time (10개 지원)

- State Machine



5. Module 설계 (stopwatch)

Block Diagram

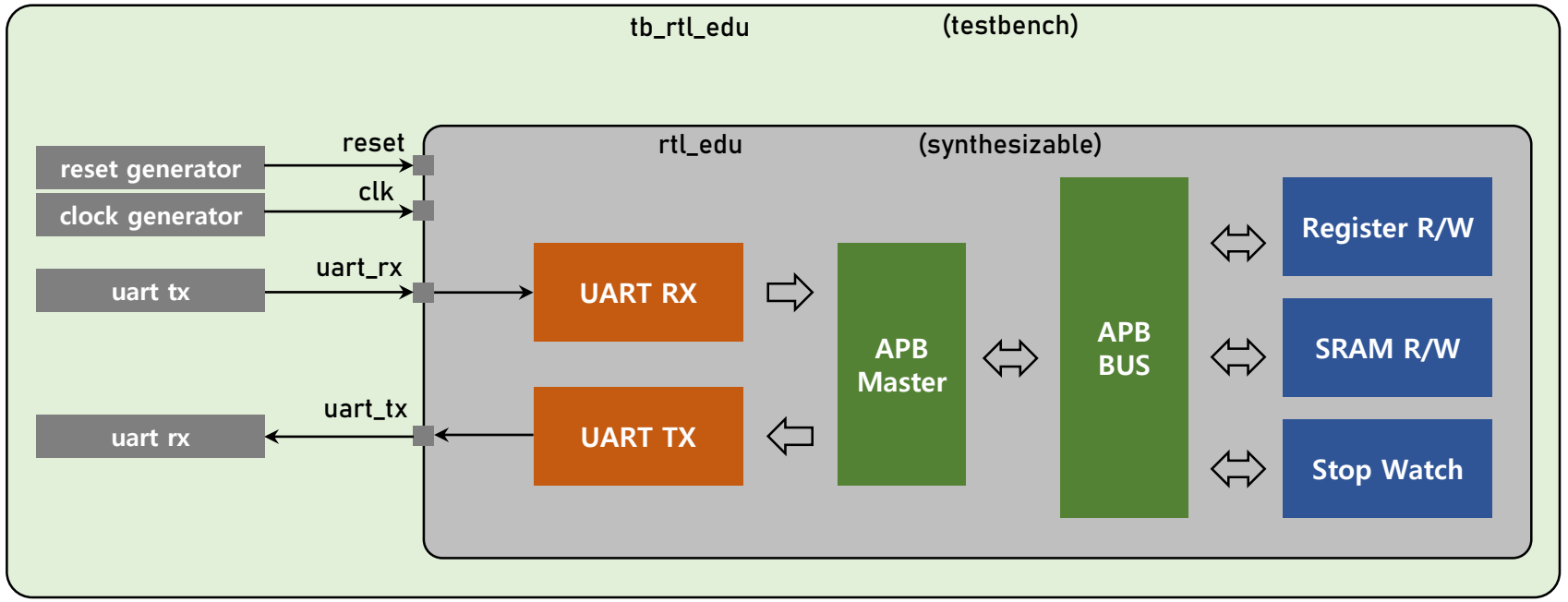


5. Module 설계 (stopwatch)

Sample Code

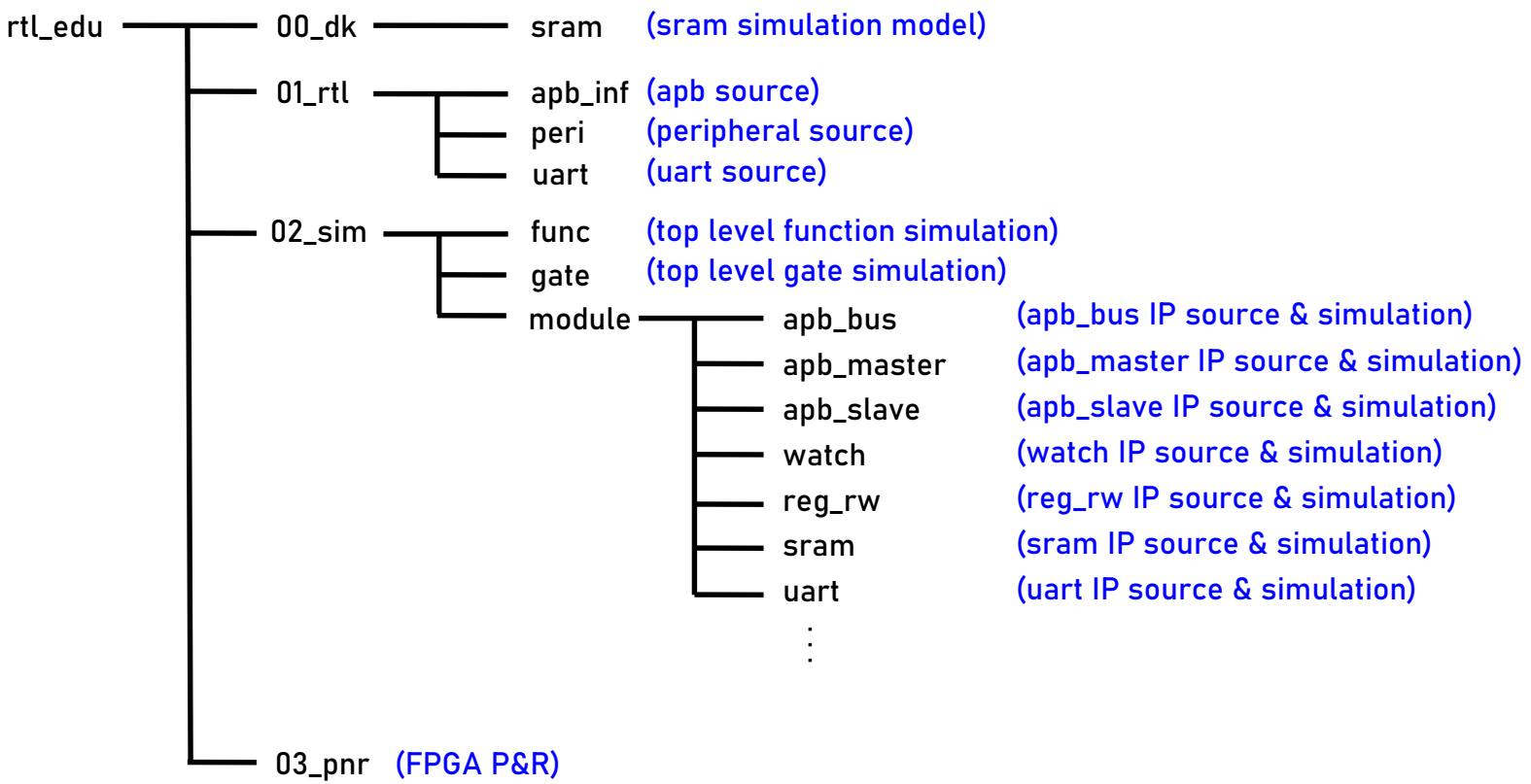
```
always @(posedge iPCLK or negedge iPRESETn) begin
    if(!iPRESETn)
        sub_sec <= 0;
    else if(curr_state == RESET)
        sub_sec <= 0;
    else if(((curr_state == RUN) | (curr_state == LAP_STORE)) & gen_10ms)
        if(sub_sec == MAX_SUB_SEC-1)
            sub_sec <= 0;
        else
            sub_sec <= sub_sec + 1;
    else
        sub_sec <= sub_sec;
end

always @(posedge iPCLK or negedge iPRESETn) begin
    if(!iPRESETn)
        sec <= 0;
    else if(curr_state == RESET)
        sec <= 0;
    else if(gen_10ms)
        if(((curr_state == RUN) | (curr_state == LAP_STORE)) & (sub_sec == MAX_SUB_SEC-1))
            if(sec == MAX_SEC-1)
                sec <= 0;
            else
                sec <= sec + 1;
        else
            sec <= sec;
    else
        sec <= sec;
end

always @(posedge iPCLK or negedge iPRESETn) begin
    if(!iPRESETn)
        min <= 0;
    else if(curr_state == RESET)
        min <= 0;
    else if(gen_10ms)
        if(((curr_state == RUN) | (curr_state == LAP_STORE)) & (sub_sec == MAX_SUB_SEC-1) & (sec == MAX_SEC-1))
            if(min == MAX_MIN-1)
                min <= 0;
            else
                min <= min + 1;
        else
            min <= min;
    else
        min <= min;
end

always @(posedge iPCLK or negedge iPRESETn) begin
    if(!iPRESETn)
        hour <= 0;
    else if(curr_state == RESET)
        hour <= 0;
    else if(gen_10ms)
        if(((curr_state == RUN) | (curr_state == LAP_STORE)) & (sub_sec == MAX_SUB_SEC-1) & (sec == MAX_SEC-1) & (min == MAX_MIN-1))
            if(hour == MAX_HOUR-1)
                hour <= 0;
            else
                hour <= hour + 1;
        else
            hour <= hour;
    else
        hour <= hour;
end
```

System Block Diagram



부록

Directory Structure



Thank you

CoAsia SEMI Ltd.