HOMEWORK 1:

```
1  `timescale 1ns / 1ps
2  module test_bench;
3    reg pclk;
4    reg psel;
5    reg pwrite;
6    reg penable;
7    reg [15:0] paddr;
8    reg [31:0] pwdata;
9
10   parameter CLK_PERIOD = 10;
11   initial begin
12     pclk = 0;
13     forever #(CLK_PERIOD / 2) pclk = ~pclk;
14   end
15
16
17   initial begin
18     psel = 1'b0;
19     pwrite = 1'b0;
20     penable = 1'b0;
21     paddr = 16'hxxxx;
22     pwdata = 32'hxxxxxxxx;
23
24     #5;
25
26     $display("INFO: Starting APB Master Simulation at time %0t", $time);
27
28     // gọi Write Task
29     @(posedge pclk);
30     $display("INFO: Calling master_write(0x1000, 0xDEADBEEF) at time %0t", $time);
31     apb_master_write(16'h1000, 32'hDEADBEEF);
32     $display("INFO: master_write finished at time %0t", $time);
33
34     repeat (2) @(posedge pclk);
35
36     // gọi Read Task
37     @(posedge pclk);
38     $display("INFO: Calling master_read(0x20A0) at time %0t", $time);
39     apb_master_read(16'h20A0);
40     $display("INFO: master_read finished at time %0t", $time);
41
42     repeat (2) @(posedge pclk);
43
```

```
43
44      // gọi Write Task khác
45      @(posedge pclk);
46      $display("INFO: Calling master_write(0xABCD, 0x12345678) at time %0t", $time);
47      apb_master_write(16'hABCD, 32'h12345678);
48      $display("INFO: master_write finished at time %0t", $time);
49
50      repeat (2) @(posedge pclk);
51
52      $display("INFO: Simulation finished at time %0t", $time);
53      $finish;
54    end
55
56    // ---- Task: Write Transfer ----
57    task apb_master_write (input [15:0] addr_in, input [31:0] data_in);
58      begin
59        // --- SETUP ---
60        paddr   <= addr_in;
61        pwdata  <= data_in;
62        pwrite  <= 1'b1;
63        psel    <= 1'b1;
64        penable <= 1'b0;
65
66        @(posedge pclk);
67        $display("TIME %0t: [WRITE] Setup Phase   -> Addr=0x%h, WData=0x%h, PSEL=1, PWRITE=1, PENABLE=0", $time, addr_in, data_in);
68
69        // --- ACCESS ---
70
71        penable <= 1'b1;
72        @(posedge pclk);
73        $display("TIME %0t: [WRITE] Access Phase  -> Addr=0x%h, WData=0x%h, PSEL=1, PWRITE=1, PENABLE=1", $time, addr_in, data_in);
74
```

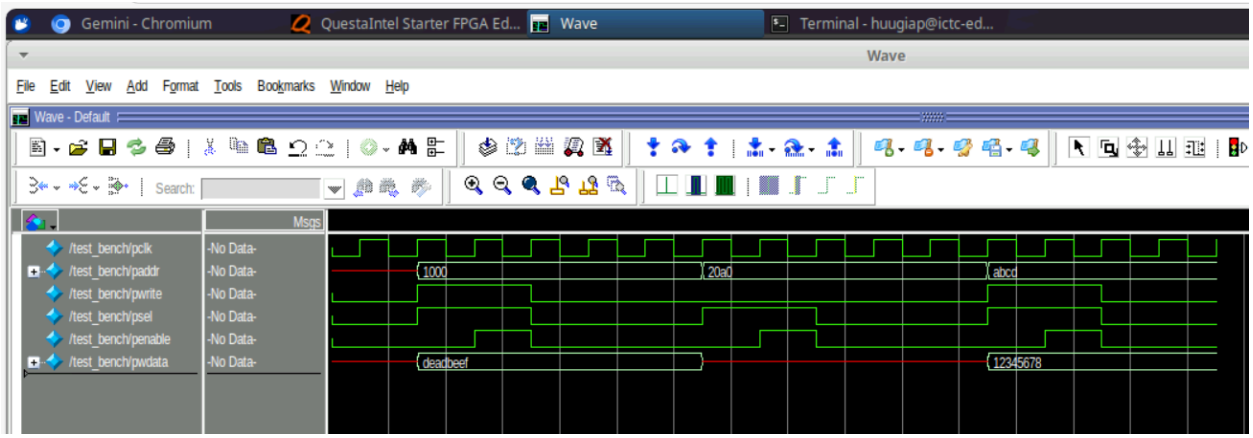
```
75        // --- END ---
76        psel    <= 1'b0;
77        penable <= 1'b0;
78        pwrite  <= 1'b0;
79        $display("TIME %0t: [WRITE] End Phase     -> PSEL=0, PWRITE=0, PENABLE=0", $time);
80      end
81    endtask
82
83    // ---- Task: Read Transfer ----
84    task apb_master_read (input [15:0] addr_in);
85      begin
86        // --- SETUP ---
87        paddr   <= addr_in;
88        pwrite  <= 1'b0;
89        psel    <= 1'b1;
90        penable <= 1'b0;
91        pwdata  <= 32'hxxxxxxxx;
92
93        @(posedge pclk);
94        $display("TIME %0t: [READ]  Setup Phase   -> Addr=0x%h, PSEL=1, PWRITE=0, PENABLE=0", $time, addr_in);
95
96        // --- ACCESS ---
97        penable <= 1'b1;
98        @(posedge pclk);
99        $display("TIME %0t: [READ]  Access Phase  -> Addr=0x%h, PSEL=1, PWRITE=0, PENABLE=1", $time, addr_in);
100
101       // --- END---
102       psel    <= 1'b0;
103       penable <= 1'b0;
104
105       $display("TIME %0t: [READ]  End Phase     -> PSEL=0, PENABLE=0", $time);
106     end
107   endtask
108
109 endmodule
```

```
huugiap@ictc-eda-ldap-1:~/10_ss10/homework1/sim$ make run
vsim -debugDB -l test_adder.log -voptargs=+acc -assertdebug -c test_bench -do "log -r /*;run -all;"
Reading pref.tcl

# 2023.3

# vsim -debugDB -l test_adder.log -voptargs="+acc" -assertdebug -c test_bench -do "log -r /*;run -all;"
# Start time: 02:02:26 on Apr 07,2025
# ** Note: (vsim-3813) Design is being optimized due to module recompilation...
# ** Note: (vsim-8611) Generating debug db.
# ** Warning: (vopt-10587) Some optimizations are turned off because the +acc switch is in effect. This will cause
# ** Note: (vsim-12126) Error and warning message counts have been restored: Errors=0, Warnings=1.
# //  Questa Intel Starter FPGA Edition-64
# //  Version 2023.3 linux_x86_64 Jul 17 2023
# //
# //  Copyright 1991-2023 Mentor Graphics Corporation
# //  All Rights Reserved.
# //
# //  QuestaSim and its associated documentation contain trade
# //  secrets and commercial or financial information that are the property of
# //  Mentor Graphics Corporation and are privileged, confidential,
# //  and exempt from disclosure under the Freedom of Information Act,
# //  5 U.S.C. Section 552. Furthermore, this information
# //  is prohibited from disclosure under the Trade Secrets Act,
# //  18 U.S.C. Section 1905.
# //
# Loading work.test_bench(fast)
# ** Note: (vsim-8900) Creating design debug database vsim.dbg.
# log -r /*
# run -all
# INFO: Starting APB Master Simulation at time 5000
# INFO: Calling master_write(0x1000, 0xDEADBEEF) at time 15000
# TIME 25000: [WRITE] Setup Phase   -> Addr=0x1000, WData=0xdeadbeef, PSEL=1, PWRITE=1, PENABLE=0
# TIME 35000: [WRITE] Access Phase  -> Addr=0x1000, WData=0xdeadbeef, PSEL=1, PWRITE=1, PENABLE=1
# TIME 35000: [WRITE] End Phase     -> PSEL=0, PWRITE=0, PENABLE=0
# INFO: master_write finished at time 35000
# INFO: Calling master_read(0x20A0) at time 65000
# TIME 75000: [READ]  Setup Phase   -> Addr=0x20a0, PSEL=1, PWRITE=0, PENABLE=0
# TIME 85000: [READ]  Access Phase  -> Addr=0x20a0, PSEL=1, PWRITE=0, PENABLE=1
# TIME 85000: [READ]  End Phase     -> PSEL=0, PENABLE=0
# INFO: master_read finished at time 85000
```

```
# Loading work.test_bench(fast)
# ** Note: (vsim-8900) Creating design debug database vsim.dbg.
# log -r /*
# run -all
# INFO: Starting APB Master Simulation at time 5000
# INFO: Calling master_write(0x1000, 0xDEADBEEF) at time 15000
# TIME 25000: [WRITE] Setup Phase   -> Addr=0x1000, WData=0xdeadbeef, PSEL=1, PWRITE=1, PENABLE=0
# TIME 35000: [WRITE] Access Phase  -> Addr=0x1000, WData=0xdeadbeef, PSEL=1, PWRITE=1, PENABLE=1
# TIME 35000: [WRITE] End Phase     -> PSEL=0, PWRITE=0, PENABLE=0
# INFO: master_write finished at time 35000
# INFO: Calling master_read(0x20A0) at time 65000
# TIME 75000: [READ]  Setup Phase   -> Addr=0x20a0, PSEL=1, PWRITE=0, PENABLE=0
# TIME 85000: [READ]  Access Phase  -> Addr=0x20a0, PSEL=1, PWRITE=0, PENABLE=1
# TIME 85000: [READ]  End Phase     -> PSEL=0, PENABLE=0
# INFO: master_read finished at time 85000
# INFO: Calling master_write(0xABCD, 0x12345678) at time 115000
# TIME 125000: [WRITE] Setup Phase   -> Addr=0xabcd, WData=0x12345678, PSEL=1, PWRITE=1, PENABLE=0
# TIME 135000: [WRITE] Access Phase  -> Addr=0xabcd, WData=0x12345678, PSEL=1, PWRITE=1, PENABLE=1
# TIME 135000: [WRITE] End Phase     -> PSEL=0, PWRITE=0, PENABLE=0
# INFO: master_write finished at time 135000
# INFO: Simulation finished at time 155000
# ** Note: $finish    : ../tb/test_bench.v(53)
#    Time: 155 ns  Iteration: 1  Instance: /test_bench
# End time: 02:02:26 on Apr 07,2025, Elapsed time: 0:00:00
# Errors: 0, Warnings: 1
```


Gemini - Chromium
QuestaIntel Starter FPGA Ed...
Wave
Terminal - huugiap@ictc-ed...
Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Search:
Msgs
/test_bench/pclk
/test_bench/paddr
/test_bench/pwrite
/test_bench/psel
/test_bench/penable
/test_bench/pwdata
-No Data-
1000
20a0
abcd
deadbeef
12345678

HOMEWORK 2:

```
1  `timescale 1ns / 1ps
2  module test_bench;
3
4    // Tín hiệu Master
5    reg pclk;
6    reg psel;
7    reg pwrite;
8    reg penable;
9    reg [15:0] paddr;
10   reg [31:0] pwdata;
11
12   // Tín hiệu Slave
13   reg pready;
14   reg [31:0] prdata;
15
16   parameter CLK_PERIOD = 10;
17   initial begin
18     pclk = 0;
19     forever #(CLK_PERIOD / 2) pclk = ~pclk;
20   end
21
22   // ---- Logic mô phỏng Slave ----
23   reg [3:0] target_delay_reg;
24   reg [3:0] current_wait_reg;
25   reg transfer_active_prev_reg;
26   reg transfer_active_now;
27
28   initial begin
29       // Khởi tạo giá trị ban đầu cho logic Slave
30       pready = 1'b0;
31       prdata = 32'hxxxxxxxx;
32       current_wait_reg = 0;
33       transfer_active_prev_reg = 0;
34       target_delay_reg = 0;
35   end
36
37   always @(posedge pclk) begin
38       transfer_active_now = psel && penable;
39
40       if (!transfer_active_now) begin
41           pready <= 1'b0;
42           current_wait_reg <= 0;
43       end else begin
```

```
    end else begin
        if (!transfer_active_prev_reg) begin
            target_delay_reg <= $urandom_range(5, 0);
            current_wait_reg <= 0;
            $display("TIME %0t: [SLAVE] Access detected. Random delay = %0d cycles.", $time, target_delay_reg);


            if (target_delay_reg == 0) begin
                pready <= 1'b1;
                $display("TIME %0t: [SLAVE] Asserting PREADY (0 cycle delay).", $time);
                if (!pwrite) begin
                    prdata <= $random;
                    $display("TIME %0t: [SLAVE] Providing PRDATA = 0x%h (valid on next edge)", $time, prdata);
                end
            end else begin
                pready <= 1'b0;
            end
        end else if (!pready) begin
            if (current_wait_reg < target_delay_reg) begin
                current_wait_reg <= current_wait_reg + 1;
            end


            if (current_wait_reg + 1 == target_delay_reg) begin
                pready <= 1'b1;
                $display("TIME %0t: [SLAVE] Asserting PREADY after %0d wait cycles (effective next edge).", $time, target_delay_reg);
                if (!pwrite) begin
                    prdata <= $random;
                    $display("TIME %0t: [SLAVE] Providing PRDATA = 0x%h (valid on next edge)", $time, prdata);
                end
            end
        end
    end
    transfer_active_prev_reg <= transfer_active_now;
end


// ---- Task: Write Transfer----
task apb_master_write (input [15:0] addr_in, input [31:0] data_in);
  begin
    // --- SETUP ---
    paddr   <= addr_in;
    pwdata  <= data_in;
```

```
    paddr   <= addr_in;
    pwdata  <= data_in;
    pwrite  <= 1'b1;
    psel    <= 1'b1;
    penable <= 1'b0;
    @(posedge pclk);
    $display("TIME %0t: [WRITE] Setup Phase    -> Addr=0x%h, WData=0x%h, PSEL=1, PWRITE=1, PENABLE=0", $time, addr_in, data_in);


    // --- ACCESS Phase + Wait States ---
    penable <= 1'b1;
    $display("TIME %0t: [WRITE] Access Phase Start -> PENABLE=1", $time);


    // --- Chờ PREADY ---
    $display("TIME %0t: [WRITE] Waiting for PREADY...", $time);
    while (!pready) begin
      @(posedge pclk);
      $display("TIME %0t: [WRITE] Still waiting... (PREADY=%b)", $time, pready);
    end
    // Pready = 1 tại sườn clock này


    $display("TIME %0t: [WRITE] PREADY detected!", $time);
    @(posedge pclk);


    // --- END ---
    $display("TIME %0t: [WRITE] End Phase     -> PSEL=0, PWRITE=0, PENABLE=0", $time);
    psel    <= 1'b0;
    penable <= 1'b0;
    pwrite  <= 1'b0;
  end
endtask


// ---- Task: Master Read Transfer ----
task apb_master_read (input [15:0] addr_in);
  reg [31:0] read_data_reg;
  begin
    // --- SETUP ---
    paddr   <= addr_in;
    pwrite  <= 1'b0;
```

```
127       psel    <= 1'b1;
128       penable <= 1'b0;
129       pwdata  <= 32'hxxxxxxxx;
130       @(posedge pclk);
131       $display("TIME %0t: [READ]  Setup Phase   -> Addr=0x%h, PSEL=1, PWRITE=0, PENABLE=0", $time, addr_in);
132
133
134       // --- ACCESS Phase + Wait States ---
135       penable <= 1'b1;
136       $display("TIME %0t: [READ]  Access Phase Start -> PENABLE=1", $time);
137
138
139       // --- Chờ PREADY ---
140       $display("TIME %0t: [READ]  Waiting for PREADY...", $time);
141       while (!pready) begin
142         @(posedge pclk);
143         $display("TIME %0t: [READ] Still waiting... (PREADY=%b)", $time, pready);
144       end
145       // Pready = 1 tại sườn clock này
146
147
148       // --- Đọc dữ liệu và In ra ---
149       read_data_reg = prdata;
150       $display("TIME %0t: [READ]  PREADY detected! Reading PRDATA = 0x%h", $time, read_data_reg);
151
152
153       @(posedge pclk);
154
155
156       // --- END ---
157       $display("TIME %0t: [READ]  End Phase     -> PSEL=0, PENABLE=0", $time);
158       psel    <= 1'b0;
159       penable <= 1'b0;
160     end
161   endtask
162
163
164   // ---- Task đợi clock (Em viết thêm) ----
165   task wait_clk_cycles (input integer num_cycles);
166       if (num_cycles > 0) begin
```

```
166       if (num_cycles > 0) begin
167           $display("INFO: Waiting for %0d idle cycles...", num_cycles);
168           repeat (num_cycles) @(posedge pclk);
169       end
170   endtask
171
172
173   // ---- Test ----
174   initial begin
175     // Khởi tạo tín hiệu Master ban đầu
176     psel = 1'b0;
177     pwrite = 1'b0;
178     penable = 1'b0;
179     paddr = 16'hxxxx;
180     pwdata = 32'hxxxxxxxx;
181
182
183     # (CLK_PERIOD * 2);
184
185
186     $display("\nINFO: ======================================================");
187     $display("INFO: Starting APB Master/Slave Simulation at time %0t", $time);
188     $display("INFO: ======================================================\n");
189
190
191     // --- 1: Write ---
192     @(posedge pclk);
193     $display("INFO: ---> Calling master_write(0x1000, 0xAAAAAAAA) at time %0t", $time);
194     apb_master_write(16'h1000, 32'hAAAAAAAA);
195     $display("INFO: ---> master_write finished at time %0t\n", $time);
196
197
198     wait_clk_cycles(3);
199
200
201     // --- 2: Read ---
202     @(posedge pclk);
203     $display("INFO: ---> Calling master_read(0x20A0) at time %0t", $time);
204     apb_master_read(16'h20A0);
205     $display("INFO: ---> master_read finished at time %0t\n", $time);
206
```

```
206
207
208     wait_clk_cycles(2);
209
210
211     // --- 3: Write (địa chỉ khác) ---
212      @(posedge pclk);
213     $display("INFO: ---> Calling master_write(0x1234, 0xBBBBBBBB) at time %0t", $time);
214     apb_master_write(16'h1234, 32'hBBBBBBBB);
215     $display("INFO: ---> master_write finished at time %0t\n", $time);
216
217
218     wait_clk_cycles(1);
219
220
221     // --- 4: Read (địa chỉ khác) ---
222      @(posedge pclk);
223     $display("INFO: ---> Calling master_read(0x5678) at time %0t", $time);
224     apb_master_read(16'h5678);
225     $display("INFO: ---> master_read finished at time %0t\n", $time);
226
227
228     wait_clk_cycles(5);
229     $display("INFO: ====================================================");
230     $display("INFO: Simulation finished at time %0t", $time);
231     $display("INFO: ====================================================");
232     $finish;
233   end
234
235
236 endmodule
237
```

```
huugiap@ictc-eda-ldap-1:~/10_ss10/homework2/sim$ make run
vsim -debugDB -l test_adder.log -voptargs=+acc -assertdebug -c test_bench -do "log -r /*;run -all;"
Reading pref.tcl

# 2023.3

# vsim -debugDB -l test_adder.log -voptargs="+acc" -assertdebug -c test_bench -do "log -r /*;run -all;"
# Start time: 01:54:16 on Apr 07,2025
# ** Note: (vsim-3812) Design is being optimized...
# ** Note: (vsim-8611) Generating debug db.
# ** Warning: (vopt-10587) Some optimizations are turned off because the +acc switch is in effect. This wil
# ** Note: (vsim-12126) Error and warning message counts have been restored: Errors=0, Warnings=1.
# //  Questa Intel Starter FPGA Edition-64
# //  Version 2023.3 linux_x86_64 Jul 17 2023
# //
# //  Copyright 1991-2023 Mentor Graphics Corporation
# //  All Rights Reserved.
# //
# //  QuestaSim and its associated documentation contain trade
# //  secrets and commercial or financial information that are the property of
# //  Mentor Graphics Corporation and are privileged, confidential,
# //  and exempt from disclosure under the Freedom of Information Act,
# //  5 U.S.C. Section 552. Furthermore, this information
# //  is prohibited from disclosure under the Trade Secrets Act,
# //  18 U.S.C. Section 1905.
# //
# Loading work.test_bench(fast)
# ** Note: (vsim-8900) Creating design debug database vsim.dbg.
# log -r /*
# run -all
#
# INFO: ==================================================
# INFO: Starting APB Master/Slave Simulation at time 20000
# INFO: ==================================================
#
```

```
# log -r /*
# run -all
#
# INFO: ====================================================
# INFO: Starting APB Master/Slave Simulation at time 20000
# INFO: ====================================================
#
# INFO: ---> Calling master_write(0x1000, 0xAAAAAAAA) at time 25000
# TIME 35000: [WRITE] Setup Phase   -> Addr=0x1000, WData=0xaaaaaaaa, PSEL=1, PWRITE=1, PENABLE=0
# TIME 35000: [WRITE] Access Phase Start -> PENABLE=1
# TIME 35000: [WRITE] Waiting for PREADY...
# TIME 45000: [SLAVE] Access detected. Random delay = 0 cycles.
# TIME 45000: [SLAVE] Asserting PREADY (0 cycle delay).
# TIME 45000: [WRITE] Still waiting... (PREADY=0)
# TIME 55000: [WRITE] Still waiting... (PREADY=1)
# TIME 55000: [WRITE] PREADY detected!
# TIME 65000: [WRITE] End Phase     -> PSEL=0, PWRITE=0, PENABLE=0
# INFO: ---> master_write finished at time 65000
#
# INFO: Waiting for 3 idle cycles...
# INFO: ---> Calling master_read(0x20A0) at time 105000
# TIME 115000: [READ]  Setup Phase   -> Addr=0x20a0, PSEL=1, PWRITE=0, PENABLE=0
# TIME 115000: [READ]  Access Phase Start -> PENABLE=1
# TIME 115000: [READ]  Waiting for PREADY...
# TIME 125000: [SLAVE] Access detected. Random delay = 0 cycles.
# TIME 125000: [SLAVE] Asserting PREADY (0 cycle delay).
# TIME 125000: [SLAVE] Providing PRDATA = 0xxxxxxxxx (valid on next edge)
# TIME 125000: [READ] Still waiting... (PREADY=0)
# TIME 135000: [READ] Still waiting... (PREADY=1)
# TIME 135000: [READ]  PREADY detected! Reading PRDATA = 0x12153524
# TIME 145000: [READ]  End Phase     -> PSEL=0, PENABLE=0
# INFO: ---> master_read finished at time 145000
#
# INFO: Waiting for 2 idle cycles...
# INFO: ---> Calling master_write(0x1234, 0xBBBBBBBB) at time 175000
# TIME 185000: [WRITE] Setup Phase   -> Addr=0x1234, WData=0xbbbbbbbb, PSEL=1, PWRITE=1, PENABLE=0
# TIME 185000: [WRITE] Access Phase Start -> PENABLE=1
# TIME 185000: [WRITE] Waiting for PREADY...
# TIME 195000: [SLAVE] Access detected. Random delay = 2 cycles.
# TIME 195000: [WRITE] Still waiting... (PREADY=0)
# TIME 205000: [WRITE] Still waiting... (PREADY=0)
# TIME 215000: [WRITE] Still waiting... (PREADY=0)
# TIME 225000: [WRITE] Still waiting... (PREADY=0)
# TIME 235000: [SLAVE] Asserting PREADY after 4 wait cycles (effective next edge).
```

```
#
# INFO: Waiting for 2 idle cycles...
# INFO: ---> Calling master_write(0x1234, 0xBBBBBBBB) at time 175000
# TIME 185000: [WRITE] Setup Phase   -> Addr=0x1234, WData=0xbbbbbbbb, PSEL=1, PWRITE=1, PENABLE=0
# TIME 185000: [WRITE] Access Phase Start -> PENABLE=1
# TIME 185000: [WRITE] Waiting for PREADY...
# TIME 195000: [SLAVE] Access detected. Random delay = 2 cycles.
# TIME 195000: [WRITE] Still waiting... (PREADY=0)
# TIME 205000: [WRITE] Still waiting... (PREADY=0)
# TIME 215000: [WRITE] Still waiting... (PREADY=0)
# TIME 225000: [WRITE] Still waiting... (PREADY=0)
# TIME 235000: [SLAVE] Asserting PREADY after 4 wait cycles (effective next edge).
# TIME 235000: [WRITE] Still waiting... (PREADY=0)
# TIME 245000: [WRITE] Still waiting... (PREADY=1)
# TIME 245000: [WRITE] PREADY detected!
# TIME 255000: [WRITE] End Phase     -> PSEL=0, PWRITE=0, PENABLE=0
# INFO: ---> master_write finished at time 255000
#
# INFO: Waiting for 1 idle cycles...
# INFO: ---> Calling master_read(0x5678) at time 275000
# TIME 285000: [READ]  Setup Phase   -> Addr=0x5678, PSEL=1, PWRITE=0, PENABLE=0
# TIME 285000: [READ]  Access Phase Start -> PENABLE=1
# TIME 285000: [READ]  Waiting for PREADY...
# TIME 295000: [SLAVE] Access detected. Random delay = 4 cycles.
# TIME 295000: [READ] Still waiting... (PREADY=0)
# TIME 305000: [READ] Still waiting... (PREADY=0)
# TIME 315000: [READ] Still waiting... (PREADY=0)
# TIME 325000: [READ] Still waiting... (PREADY=0)
# TIME 335000: [SLAVE] Asserting PREADY after 4 wait cycles (effective next edge).
# TIME 335000: [SLAVE] Providing PRDATA = 0x12153524 (valid on next edge)
# TIME 335000: [READ] Still waiting... (PREADY=0)
# TIME 345000: [READ] Still waiting... (PREADY=1)
# TIME 345000: [READ]  PREADY detected! Reading PRDATA = 0xc0895e81
# TIME 355000: [READ]  End Phase     -> PSEL=0, PENABLE=0
# INFO: ---> master_read finished at time 355000
#
# INFO: Waiting for 5 idle cycles...
# INFO: ====================================================
# INFO: Simulation finished at time 405000
# INFO: ====================================================
# ** Note: $finish    : ../tb/test_bench.v(239)
#    Time: 405 ns  Iteration: 1  Instance: /test_bench
```

```
#
# INFO: Waiting for 1 idle cycles...
# INFO: ---> Calling master_read(0x5678) at time 275000
# TIME 285000: [READ]  Setup Phase    -> Addr=0x5678, PSEL=1, PWRITE=0, PENABLE=0
# TIME 285000: [READ]  Access Phase Start -> PENABLE=1
# TIME 285000: [READ]  Waiting for PREADY...
# TIME 295000: [SLAVE] Access detected. Random delay = 4 cycles.
# TIME 295000: [READ] Still waiting... (PREADY=0)
# TIME 305000: [READ] Still waiting... (PREADY=0)
# TIME 315000: [READ] Still waiting... (PREADY=0)
# TIME 325000: [READ] Still waiting... (PREADY=0)
# TIME 335000: [SLAVE] Asserting PREADY after 4 wait cycles (effective next edge).
# TIME 335000: [SLAVE] Providing PRDATA = 0x12153524 (valid on next edge)
# TIME 335000: [READ] Still waiting... (PREADY=0)
# TIME 345000: [READ] Still waiting... (PREADY=1)
# TIME 345000: [READ]  PREADY detected! Reading PRDATA = 0xc0895e81
# TIME 355000: [READ]  End Phase      -> PSEL=0, PENABLE=0
# INFO: ---> master_read finished at time 355000
#
# INFO: Waiting for 5 idle cycles...
# INFO: ============================================================
# INFO: Simulation finished at time 405000
# INFO: ============================================================
# ** Note: $finish    : ../tb/test_bench.v(239)
#    Time: 405 ns  Iteration: 1  Instance: /test_bench
# End time: 01:54:17 on Apr 07,2025, Elapsed time: 0:00:01
# Errors: 0, Warnings: 1
```