國立彰化師範大學電子系硬體描述語言期中考筆試題庫

Question Bank for Midterm Exam, Hardware-Descriptive Language, Dept. of Electronics Eng., NCUE, Taiwan

Updated: 2022/09/15

Instructor: Prof. Tsung-Chu Huang

(I) Write a complete Verilog module for the Schematics 寫出完整Verilog模組(答案可能不唯一)

| No. | 電路圖(Schematics)/  規格符號(Spec Symbols) | Verilog Codes (for Reference) |
| --- | --- | --- |
| 1 |  | module AND2(Out, A, B); // gate-level  input A, B;  output Out;  and g1(Out, A, B);  endmodule  module AND2(Out, A, B); // Static RTL  input A, B;  output Out;  wire Out;  assign Out = A & B;  endmodule  module AND2(Out, A, B); // Procedural RTL  input A, B;  output Out;  reg Out;  always@(A or B)  Out <= A & B;  endmodule |
| 2 |  | module Wiring(A, B);  input [11:0] A;  output [11:0] B;  assign B = {A[6], A[7], A[8], A[9],  A[10], A[11], A[2:0], A[5:3]};  endmodule |
| 3 |  | module Ckt(F, A, B, H, K, C); // gate-level  input A, B, H, K, C;  output F;  xnor g1(X, H, K);  xor g2(Y, A, B, X);  notif1 g3(F, Y, C);  endmodule |
| 4 | Multiplexer | module MUX(A, B, C, F);  input A, B, C;  output F;  assign F = C ? A, B;  endmodule |
| 5 | Half Adder | module HA(A, B, C, S);  input A, B;  output C, S;  xor g1(S, A, B);  and g2(C, A, B);  endmodule  module HA(A, B, C, S);  input A, B;  output C, S;  assign S = A ^ B;  assign C = A & B;  endmodule  module HA(A, B, C, S);  input A, B;  output C, S;  assign {C, S} = A + B;  endmodule |
| 6 | Full Adder with 2 Half Adders | module FA(A, B, C, Co, S); // gate-level  input A, B, C;  output Co, S;  HA g1(A, B, C1, S1);  HA g2(S1, C, C2, S);  Or g3(Co, C1, C2);  // or using  // assign {Co, S} = A + B + C;  endmodule |
| 7 | 32-bit Adder with 32 Full Adders | module Adder(A, B, Ci, Co, S);  input [31:0] A, B;  input Ci;  output Co;  output [31:0] S;  genvar i;  wire [32:0] Y;  for(i=0; i<32; i=i+1) begin: Cell  FA g(A[i], B[i], Y[i], Y[i+1], S[i]);  end  assign Y[0] = Ci;  assign Co = Y[32];  endmodule |
| 8 | 32-bit Adder/Subtractor with only one 32-bit Adder, one Carry-in, and one command Sub | module AddSub(Sub, A, B, Out);  input Sub;  input [31:0] A, B;  output [32:0] Out;  assign Out = A + Sub ? ~B : B + Sub;  endmodule |
| 9 | Truth Table of F(A, B, C):   |  |  |  |  |  | | --- | --- | --- | --- | --- | | AB  C | 00 | 01 | 11 | 10 | | 0 | 1 | 1 | 0 | 0 | | 1 | 0 | 1 | 1 | 0 | | module LUT(A, B, C, F); // Look-up table  input A, B, C;  output F;  always@(\*)  case({A, B, C})  3’b000: F=1;  3’b010: F=1;  3’b011: F=1;  3’b111: F=1;  Default: F=0;  endcase  endmodule |
| 10 |  | module DFF(Clk, D, Q); // behavioral RTL  input Clk, D;  output Q;  reg Q;  always@(posedge Clk)  Q <= D;  endmodule |
| 11 | Behavioral model of JK FF | module JKFF(Clk, J, K, Q);  input Clk, J, K;  output Q;  reg Q;  always@(posedge Clk)  if(J) if(K) Q <= !Q;  else Q <= 1;  else if(K) Q <= 0;  endmodule |
| 12 | FSM of JK FF: | module JKFF(Clk, J, K, Q);  input Clk, J, K;  output Q;  reg PS, NS;  assign Q = PS;  always@(posedge Clk) PS <= NS;  always@(\*)  casex({J, K, Q})  3’b0X0: NS=0;  3’b1X0: NS=1;  3’bX01: NS=1;  3’bX10: NS=0;  endcase  endmodule |
| 13 |  | module PRNDFF(Clk, PRN, D, Q);  input Clk, PRN, D;  output Q;  reg Q;  always@(posedge Clk)  if(!PRN) Q <= 1;  else Q <= D;  endmodule |
| 14 |  | module RDFF(Clk, RstN, D, Q);  input Clk, RstN, D;  output Q;  reg Q;  always@(negedge Clk or negedge RstN)  if(!RstN) Q <= 0;  else Q <= D;  endmodule |
| 15 |  | module RTFF(Clk, Rst, T, Q); // Toggle FF  input Clk, Rst, T;  output Q;  reg Q;  always@(negedge Clk)  if(Rst) Q <= 0;  else Q <= !Q;  endmodule |
| 16 | A 50% duty 125MHz Clock Clk  Time unit: 1ns, resolution: 10ps | `timescale 1ns/10ps  module Clock(Clk);  output Clk;  reg Clk;  initial Clk = 0;  always Clk = #4 ~Clk;  endmodule |
| 17 | 10-bit Clock counter (Timer) counting up with step = 1 | module Timer(Clk, Q);  input Clk;  output [9:0] Q;  reg [9:0] Q;  always@(posedge Clk) Q <= Q + 1;  endmodule |
| 18 | 10-bit synchronously all-presettable/loadable Clock counter (Timer) counting down with step = 1 | module Timer(Clk, Rst, Load, D, Q);  input Clk, Rst, Load;  input [9:0] D;  output [9:0] Q;  reg [9:0] Q;  always@(posedge Clk)  if(Rst) Q <= 10’b1111111111;  else if(Load) Q <= D;  else Q <= Q - 1;  endmodule |
| 19 | Resettable modulo-100 count-up counter (Counting by In) | module Counter(In, Rst, Q);  input In, Rst;  output [6:0] Q;  reg [6:0] Q;  always@(posedge In)  if(Rst) Q <= 7’b0;  else if(Q==99) Q <= 7’b0;  else Q <= Q + 1;  endmodule |
| 20 |  | module Adder(A, B, Sum);  input [15:0] A, B;  output [15:0] Sum;  wire [15:0] Sum;  assign Sum = A + B;  endmodule |
| 21 |  | module OneHot(A, D);  input [1:0] A;  output [3:0] D;  assign D = 4’b0001 << A;  endmodule  module OneHot(A, D);  input [1:0] A;  output [3:0] D;  reg [3:0] D;  always@(\*)  case(A)  0: D <= 4’b0001;  1: D <= 4’b0010;  2: D <= 4’b0100;  3: D <= 4’b1000;  endcase  endmodule |
| 22 |  | module OneHot(A, W);  input [2:0] A;  output [7:0] W;  assign W = 1 << A;  endmodule |
| 23 | Oscillating Ring consisted of 1 nand gate with EN and 100 not gates | module OscRing(EN, Osc);  input EN;  output Osc;  wire [100:0] W;  nand g0(W[0], EN, W[100]);  genvar i;  generate  for(i=1; i<101; i=i+1) begin: INV  not g1(W[i], W[i-1]);  end  endgenerate  endmodule |
| 24 | Gold test: | module GoldenTest;  parameter n = 16;  parameter m = 16;  reg [n-1: 0] IN;  wire [m-1: 0] OutG, OutT;  Gold U1(IN, OutG);  CUT U2(IN, OutT);  integer i, Fail;  initial begin  Fail = 0;  for(i=0; i<(1<<n); i=i+1) begin  IN = i;  #10;  if(OutG != OutT) Fail = 1;  end  if(Fail) $display(“Failed!”);  else $display(“Passed!”);  end  endmodule |

II. TRUE OR FALSE (Mark ○ or X)

( ○ ) 1. A schematic is a graph in a structural view with elements in symbol views.

( ○ ) 2. Syntax errors can be detected by compilers.

( ○ ) 3. The event to trigger $display is itself while the events to trigger $monitor are the changes of the monitored variables.

( ○ ) 4. A combinational circuit can be mapped to a Directed Acyclic Graph (DAG) with gates as nodes.

( 🞨 ) 5. A sequential circuit with 3 D-flip-flops must have 8 states.

( 🞨 ) 6. The wildcard character “\*” in “always@\*” in Verilog means all events for a sequential circuit.

( ○ ) 7. Transitions of signals including positive and negative edges are looked as events in HDL.

( ○ ) 8. An *f*-Hz N-bit binary counter can serve as a frequency divider that provides *f*÷2*i* Hz pulse signals where *i* = 1 … N.

( ○ ) 9. Generally to speak, the SPICE netlist is also a kind of HDL.

( ○ ) 10. A general HDL can be a combination of Verilog, SystemVerilog, VHDL and SystemC.

( 🞨 ) 11. `timescale 10ps/1ns declares a time unit is 1ns.

( ○ ) 12. Verilog statement “xor g1(.o(F), A, B);” will cause a syntax error.

( ○ ) 13. bufif1 g1(F, A, C) may result in a tri-state when C=0.

( 🞨 ) 14. “reg [7:0] S; always@(\*) for(i=0; i<10; i++) S = S + X[i];” contains only one 2-input adder.

( ○ ) 15. 10’b1111110111 == -9.

( ○ ) 16. 10’b1111110111 << 3 == 10’b1110111000.

( ○ ) 17. 10’b1110111000 >> 3 == 10’b0001110111 is a logic right shift.

( ○ ) 18. 10’b1110111000 >>> 3 == 10’b1111110111 is a sign-extension right shift.

( 🞨 ) 19. wire [7:0] x; x<<3 == x\*8.

( ○ ) 20. Syntax errors are more difficult to be debugged than design errors.

( ○ ) 21. The number 16 in the chip in our experiments, EP3C16F484C6N means it’s with about 16k logic elements (LEs).

( ○ ) 22. The letter F in the chip in our experiments, EP3C16F484C6N means it’s in a flip-chip package.

( ○ ) 23. The number 484 in the chip in our experiments, EP3C16F484C6N means it’s with a 2222 pin array.

( ○ ) 24. The board for experiments, DE0 provides a 50-MHz oscillator.

( 🞨 ) 25. A logic-0 (low-voltage) signal will turn on a common-cathode (CK) LED while a logic-1 (high-voltage) will lighten common-anode (CA).

III. MULTIPLE CHOICE (Choose the best one)

( D ) 1. The purpose of rolling the behavioral-structural-physical table into a Y-char is to emphasize (A) partition (B) analysis (C) library (D) verification.

( C ) 2. Which design phase is the shortest? (A) mass production (B) prototyping chip (C) rapid prototyping (D) reverse engineering.

( C ) 3. Synthesizing a combinational circuit in a static assignment, (A) Left values must be reg (B) Right values must be in reg (C) Left values must be wires (D) Right values must be wires.

( D ) 4. Verilog statement “F=C?A:B;” can be implemented by (A) Demultiplexer (B) NOR gate (C) shifter (D) multiplexer.

( D ) 5. Which is not related to IEEE1364 compilers? (A) Modelsim (B) Xilinx’s ISE (C) Altera’s Quartus II (D) GNU Prolog.

( D ) 6. The top module in simulation phase is usually a (A) system (B) element (C) dummy module (D) testbench.

( D ) 7. Which are non-blocking assignments? (A) x<=y; y>=x; (B) x=y=z; (C) x=y; y=z; (D) x<=y; y<=z;

( D ) 8. In IEEE1364-1995, always Clk= #10 ~Clk; initial Clk=0; then Clk (A) always 0 (B) always 1 (C) duty cycle=50% (D) unknown x.

( C ) 9. #3.141592654 in `timescale 1ns/1ps means (A) 3ns (B) 3.142ps (C) 3.142ns (D) 3141.6ps.

( B ) 10. (A) 1<<3 == 4; (B) n[3:0] = -2; n[3:0]==4’hE; (C) 1’b? & 1’b? == 1’bz; (D) 4’b0011 & 4’b0101 == 4’b0111;.

( C ) 11. for loop with index i in {initial, generate, always} blocks, i should be declared as a (A) {integer, genvar, reg} (B) {integer, genvar, genvar} (C) {integer, genvar, integer} (C) {reg, genvar, integer}.

( B ) 12. Continued from the above, R[i] in the for loop act in (A) {spatial, sequential, spatial} (B) {sequential, spatial, spatial} (C) {parallel, sequential, spatial} (C) {sequential, spatial, iterative}.

( B ) 13. Which block is applied for timing/power constraints: (A) task (B) function (C) specify (C) generate.

( A ) 14. Sampling debouncer applying a frequency f for the maximum bouncing period tb and minimum pulse width tp: (A) tb < 1/f < tp (B) tb < f < tp (C) tb > 1/f > tp (D) tb > f > tp.

( B ) 15. How to call a parametered module F(I, O); parameter p; (A) F U1 (p) (I, O); (B) F (p) U1(I, O); (C) (p) F U1(I, O); (D) F U1(p, I, O);.

( C ) 16. In Quartus II, the block assignment (eg., begin A=B; B=A; end) is implemented by sourcing the right value from (A) FF input, (B) FF output, (C) First latch output, (D) Second latch output.

( D ) 17. So far, which language is seldom applied as a HDL? (A) Verilog, (B) VHDL, (C) C, (D) Python.

( B ) 18. The writing in Verilog module declaration with a line per ported-net is called (A) IP Qualification, (B) ANSI style, (C) NIST style, (D) indentation.

( A ) 19. An 8-bit 2’s complement integer n = 8'h8B is sign-extended to 11-bit x; m = n3; which is wrong? (A) m is 8-bit, (B) m is 11'h468, (C) m equals n \* 8; (D) x is 11'h78B.

( A ) 20. m = 11'h468; y=m3; y is (A) 8'h8B, (B) 11'h78B, (C) 8'h8B, (D) 11'h08B.

( A ) 20. A for loop in an initial-block will execute in (A) serial, (B) parallel, (C) advance, (D) void.

( A ) 21. Which is wrong that a for loop in a generate-block will generate elements in (A) serial, (B) parallel, (C) advance.

( C ) 22. A for loop in an always-block is done (A) in serial (B) in parallel (C) concurrent driven by event (D) with an integer index.

( A ) 23. Verilog-A is in (A) Verilog format Spice engine, (B) Spice format Verilog engine, (C) both engines, (D) both formats.

( A ) 24. So far which is the heist level HDL? (A) HLS C++, (B) SystemC, (C) SystemVerilog, (D) VHDL.

( D ) 25. The 8th bit of counter Q[7] divides the frequency by (A) 7, (B) 8, (C) 128, (D) 256 times.

IV. Questions

1. Assume each inertial delay equals to 90% of its gate delay and only the levels longer than the inertial delay can pass the gate. Draw the timing waveforms of A, B, C and D in the following code. (Note that the initial states are unknown and remember to mark gray or slash-lined for unknown states)

module test;

reg A, B;

and g1(C, A, B);

and #5 g2(D, A, B);

and #10 g3(E, A, B);

initial begin

A=0; B=1;

#10 A=1;

#5 B=0;

#10 A=0;

#10 B=1;

#5 A=1;

#15 $stop;

end

endmodule

2. Please give the most-frequent-used statement styles of Verilog codes in the rectangles according to the given examples.



3. Please draw the possibly-synthesized gate-level or block-level diagrams of the following Verilog statement blocks. Denote that it is a combinational or sequential circuits.

(1)

reg [1:0] NS, PS;

always@(posedge Clk) NS=PS;

(2)

reg PS, Out;

always@\* case(PS)

0: Out = A\*B;

1: Out = A+B;

endcase

(3)

reg PS, Out;

always@(posedge Clk) case(PS)

0: Out = A\*B;

1: Out = A+B;

endcase

(4)

wire [31:0] Out;

assign Out = A\*B;

4. Give a Go/NoGo verification testbench for Golden(output [7:0] Out, input [9:0] In) and DUT(output [7:0] Out, input [9:0] In) combinational circuits.

5. AB are two BCD digits that are declared as reg [3:0] A, B; Let concatenated AB represent a 24-hour number from 00 to 23 to count the input Clk1hr. Write the Verilog code of the 24-hour counter by BCD carrying approaches.

6. Given a clock Clk in a frequency of 12.3456MHz. Write a frequency divider, module ClkGen(Clk, Clk1ms);, to generate a clock Clk with a duty cycle of 50% in a frequency of 1kHz.

7. Write a modulo-57 counter synchronously initialized/reset from 19.

8. What is the distinction between a Moore and Mealy finite state machine?

9. Compare the for ( datatype index ) loops in the following table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Block | (serial, parallel) | Indexing datatype | In order | What are increased | Block name |
| initial | serial | integer | yes | time | no |
| generate | parallel | genvar | no | reg or module | yes |
| always@(event) | parallel (concurrent) | integer | no | Step or procedure | no |

10. The input X=0 when a button is pressed, otherwise 1. Given a 1kHz timer, it starts when the button is pressed and stop when the button is released. If the time is greater than 0.5 seconds, the output Q will be 1, otherwise 0. Draw the 3-state (idle, start, stop) state diagram with input x. Write the Verilog module Key(input Clk1kHz, input X, output Q).

11. Given a full adder, FA(input A, input B, input C, output S, output Co); write a n-bit adder, Adder, with an (n+1)-bit output Sum but without carry I/O using a generate statement with a parameter n. Instantiate a 128-bit (n=128) Adder in a module Adder\_test.

12. Develop a Verilog module of a debouncer for a push button switch that uses a de-bounce interval of 10ns. Assume the system clock frequency is 50 MHz. Draw the (button, clock, output) timing diagram in 2 cases, (clock inside bouncing interval, clock outside bouncing interval) for button signal is pressed from 1 to 0.

13. If the data type notation of an N-bit 2’s complement fixed-point integer is denoted as <N, m>, where m is the bit count of its mantissa. Write the notation result of and . What issues will occur separately for two operations if and ?

14. Except of HDLs above System-Verilog, Verilog cannot take an array, say input [7:0] X[0:15] and output [5:0] Y[1:10], as the input. How do you arrange the writing in the instantiated module and instantiating unit of your Verilog codes?

15. Give a clock signal Clk1m with a period 1 minute, write a HH:MM formatted clock in Verilog codes, Clock(Clk1m, Rst, H1, H0, M1, M0);, to show the 2-BCD Hours and 2-BCD Minutes directly using the BCD arithmetic.

16. Write an 8-bit adder with a bitwise-carry delay 0.13ns. And write a test-bench to approach the operating frequency (performance) by a timing closure method over the golden-test formal verifications.

17. Design two simple examples by specify $hold and $setup for illustrating the setup-time and hold-time violations, separately.