

Giảng viên ra đề:	Ngày ra đề: 25/05/2023	Người phê duyệt:	Ngày duyệt đề
 TRƯỜNG ĐẠI HỌC BÁCH KHOA – ĐHQG-HCM Khoa Khoa học & Kỹ thuật Máy tính		THI CUỐI KỲ	Học kỳ II
			Năm học 2022-2023
		Môn học	Ngày thi: 02/06/2023
		Mã môn học	Thiết kế luận lý với HDL
		Thời lượng	70 phút
		Mã đề	2301
Ghi chú: Sinh viên ĐƯỢC sử dụng tài liệu; THU LẠI ĐỀ			

1. (L.O.1) Trong một project có nhiều file nguồn HDL, quá trình RTL Analysis được tiến hành từ file nào?

A. File constraints được đánh dấu là “target”
 B. File mã nguồn được đánh dấu là “target”
 C. File constrain được đánh dấu là “top”
☒ D. File mã nguồn được đánh dấu là “top”

Thông tin dùng chung từ câu 2 đến 3.

Biết một mạch có 2 input là A, B và 2 output là f và g. Outputs được xác định dựa vào bảng sự thật bên dưới:

A	B	f	g
0	0	1	0
0	1	1	1
1	0	0	0
1	1	1	0

2. (L.O.1) Xác định biểu thức $f(A, B)$
- ☒ A. $f(A, B) = \bar{A} + B$ B. $f(A, B) = \bar{A} \cdot \bar{B}$
 C. $f(A, B) = \bar{A} \cdot B$ D. $f(A, B) = \bar{B}$
3. (L.O.1.3) Mô tả output g bằng ngôn ngữ Verilog:

A. **assign** g = f. !B;
 B. **assign** g = !B. f;
☒ C. **assign** g = !A. B;
 D. **assign** g = A^B;

4. (L.O.1) Bước nào “Tạo ra file dùng để nạp cho FPGA”?

A. Synthesis
☒ B. Generate Bit Stream
 C. RTL Analysis
 D. Implementation

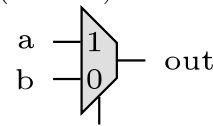
5. (L.O.1) Nguyên nhân của lỗi “multiple driver nets” là:

A. Một register được gán nhiều lần bởi nhiều giá trị khác nhau
 B. Dùng cả phép gán blocking và phép gán non-blocking lên cùng một register

C. Dùng phép gán blocking và non-blocking một cách xen lẫn **nhiều phần tử logic có điều khiển 1 tín hiệu gây ra lỗi multiple drivers net.**

- ☒ D. Một đường tín hiệu được nối với nhiều output

6. (L.O.1.1) Cho mạch như hình bên dưới.



Khi sel = 1, out = a
 Khi sel = 0, out = b

sel

Đoạn code verilog nào sau đây mô tả phần cứng trên.

A. **assign** out = ~sel.a + sel.b
☒ B. **assign** out = sel ? a : b; **câu lệnh gán điều kiện**
 C. **wire** out = a mux b; **Out sẽ dựa vào sel**
 D. A, B, C đều sai.

Thông tin dùng chung từ câu 7 đến 8.

Cho biểu thức $f(A, B, C) = \overline{(\bar{A} \cdot B)} \cdot \overline{(B \cdot C)} \cdot \overline{(C \cdot A)}$

7. (L.O.1) Xác định giá trị của f khi A = 1'b1, B = 1'b1, C = 1'b1;
 A. 0 ☒ B. 1 C. X D. Z

- ☒ 8. (L.O.1) KHÔNG biến đổi f. Biết f được tổng hợp theo công nghệ CMOS chỉ dùng cổng NAND. Xác định số lượng transistor tối thiểu để tổng hợp biểu thức trên. Có thể dùng cổng NAND với nhiều input.
1 cổng NAND cơ bản bao gồm 4 transistor
1 Cổng NAND 3 input gồm 6 transistor

A. 36 B. 30 ☒ C. 18 D. 14

Thông tin dùng chung từ câu 9 đến 10.

```
reg [2:0] a, b;
wire [3:0] sel;
always @(*) begin
  a = 3'd0;
  case (sel)
    3'd0: begin a = 3'd0; b = 3'd1; end
    3'd1: begin a = 3'd1; b = 3'd2; end
```

Select có 4 trường hợp 0,1,2,3
 ở đây mới khai báo có 3 trường hợp là 0,1,2
 Mặc định

```

3'd2: begin a = 3'd2; end
default: begin b = 2'd1; end
endcase
end

```

9. (L.O.1.2) Khi tổng hợp đoạn mô tả trên, bao nhiêu latch sẽ được sinh ra

☒ A. 1 B. 2 C. 3 D. 4

10. (L.O.1.2) Khi $sel = 3$, Xác định giá trị của a, b

A. $a = 3'b010$; $b = 3'bxxx$

B. $a = 3'b000$; $b = 3'bxx1$

C. $a = a$ cũ; $b = b$ cũ

☒ D. $a = 3'b000$; $b = 3'b001$

giá trị của b là x01
nhưng do x là giá trị nhị
phân không hợp lệ
verilog tự gán giá trị 0

Thông tin dùng chung từ câu 11 đến 12.

Cho mô tả mạch: **assign** out = $^{\wedge}(a \& b)$;.
Biết a, b, out đều là wire và lần lượt là 6, 4, 2 bit.

11. (L.O.1.3) Điều gì xảy ra khi tính toán về bên phải dấu =.

A. b sẽ mở rộng zero từ 4 bits lên 6 bits.

☒ B. b sẽ mở rộng dấu từ 4 bits lên 6 bits.

C. a sẽ giảm số bit từ 6 bits về 4 bits.

D. a, b giữ nguyên kích thước.

12. (L.O.1.3) Biết $a=20$, $b=15$. Xác định giá trị của out.

01 0100 & 00 1111 = 00 0100
Số bit 1 là lẻ \Rightarrow phép XOR = 1

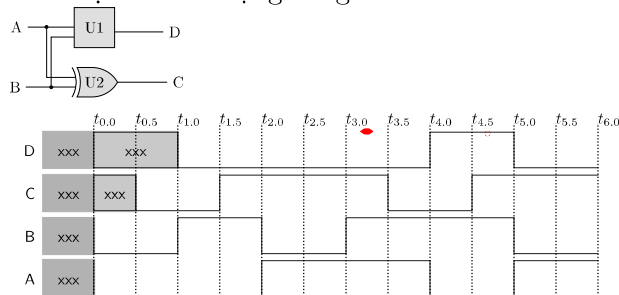
A. 0 ☒ B. 1 C. 2 D. 4

13. (L.O.1.3) Cho mô tả **assign** out = $(a === b)$; Xác định giá trị của out (1bit) khi biết $a = 3'b10x$; $b = 3'10x$.

☒ A. 1 B. 0 C. x D. z

Thông tin dùng chung từ câu 14 đến 16.

Cho mạch số và dạng sóng của nó như bên dưới:



14. (L.O.1) Dựa vào hình vẽ và sóng hiển thị. Xác định delay của cổng XOR U2.

A. 0 unit delay B. 1.0 unit delay
☒ C. 0.5 unit delay D. 1.5 unit delay

15. (L.O.1) Giả sử delay của cổng XOR là 1.5 đơn vị thời gian. Xác định giá trị của tín hiệu C tại thời điểm $t_{4.7}$

☒ A. 0 B. 1 C. X D. Z

16. (L.O.1) Biết khối U1 chỉ chứa 1 cổng duy nhất và delay của cổng đó là 1 đơn vị thời gian. Xác định cổng của khối U1. $A=1, B=1 \Rightarrow D=1$

A. OR B. NOT C. NOR ☒ D. AND

Thông tin dùng chung từ câu 17 đến 19.

```

wire sel;
(1)_____ [7:0] a, b;
(2)_____ [7:0] c;
always @(a, b, c) begin
    if (sel) c = a;
    else    c = b;
end

```

cấu trúc always
Đầu ra thay đổi khi phần trong
@() thay đổi
 \Rightarrow khi a,b,c thay đổi. Khối
always được kích hoạt \Rightarrow
tuần tự

17. (L.O.1.2) Mô tả trên sẽ tạo ra mạch?

A. Tổ hợp

☒ B. Tuần tự

C. tương tự

D. A, B, C đều sai

18. (L.O.1.2) Chọn keyword hợp lý để điền vào vị trí (1) _____ và vị trí (2) _____

A. wire, wire

☒ B. reg, reg

C. wire, reg

D. reg, wire

19. (L.O.1.3) Mô tả **always** trên có thể thay thế bằng mô tả nào dưới đây?

☒ A. **assign** c = sel ? a : b;

B. **assign** c = !sel&a | sel&b;

C. **assign** c = (sel == 0) ? a : b;

D. **assign** c = sel + a + b;

20. (L.O.1) Đối tượng nào sau đây không được sử dụng trong khối always

☒ A. Phép gán liên tục thực hiện song song (\leftarrow)

B. Câu lệnh điều kiện

C. Phép gán non-blocking thực hiện theo thứ tự (\leftarrow)

D. Khối begin-end

Thông tin dùng chung từ câu 21 đến 22.

```

reg [3:0] x, y;
always @(posedge clk) begin
    x <= x + 1'b1; x=1010
    y <= x;
end

```

Biết giá trị hiện tại của $x = 4'b1001$, $y = 4'b1011$.

21. (L.O.1.4) Xác định giá trị của y sau cạnh lên xung clk tiếp theo là:

☒ A. 4'b1001 B. 4'b1010
C. 4'b1011 D. 4'b1000

22. (L.O.1.4) Giả sử thay phép <=(non-blocking) bằng phép = (blocking). Xác định giá trị của y.

A. 4'b1001 ☒ B. 4'b1010
C. 4'b1011 D. 4'b1000

23. (L.O.1) Mô tả delay (#) sẽ được tổng hợp xuống phần cứng tương đương nào sau đây?

A. Thanh ghi.
B. Cổng luận lý.
C. Cả A và B đều được.
☒ D. Không thể tổng hợp delay.

24. (L.O.1.4) Khai báo **reg** [15:0] out. Mô tả Verilog nào sau đây hiện thực thanh ghi dịch trái 16 bit:

☒ A. out <= {out[15:1], 1'b0} xuất ra 15-->1, thêm 1 biến 0 vào bên phải
B. out <= {out[14:0], 1'b0} VD: dịch trái 4 bit 0101-->1010
C. out <= {1'b0, out[14:0]} ta lấy từ bit 1-->4
D. out <= {1'b0, out[15:1]} sau đó thêm 0 vào bên phải

Thông tin dùng chung từ câu 25 đến 26.

wire a, b, c, y, z; delay của y và z = 5
assign z = #5 y ^ c; tức là sau khi a và b thay đổi được 5 đvtg, y mới đổi
assign y = #5 a & b;

Biết tại thời điểm t: a, b, c, y, z lần lượt có giá trị là 1, 1, 1, x, x.

25. (L.O.1.3) Xác định giá trị y, z tại thời điểm t + 10.

☒ A. y=1, z=0 tại t+5 y=1
tại t+10 z=y(5)^c=1^B=0
C. y=1, z=x D. y=x, z=x

26. (L.O.1.3) Giả sử tại thời điểm t + 2, dùng force (mô phỏng) để gán y = 1'b0. Xác định giá trị y, z tại thời điểm t + 10.

A. y=0, z=0 ☒ B. y=0, z=1
C. y=1, z=0 D. y=1, z=1

Thông tin dùng chung từ câu 27 đến 29.

Cho testbench và module như đoạn mô tả bên dưới:

```
// time_unit/time_precision
`timescale 1ns /1ps
module finalexam_testbench();
  wire [3:0] c_out;
  reg [4:0] a_in, b_in;
  finalexam UUT (a_in, b_in, c_out);
```

initial begin

```
    a_in = 0; b_in = 0;
    # 1.0 a_in = 10;
    # 1.1234 a_in = 20;
```

end

endmodule

// module UUT

module finalexam(a, b, c);

input [3:0] a, b;

output [7:0] c;

assign c = {a,b};

endmodule

số bit của mỗi đầu vào là 4==> a và b có 2^4 th
tổng số TH là 2^4*2^4

27. (L.O.2) Để kiểm thử TẤT CẢ các trường hợp của **module** final_exam thì cần sinh ra bao nhiêu trường hợp khác nhau trong testbench:

A. 16 B. 32 C. 64 D. 128

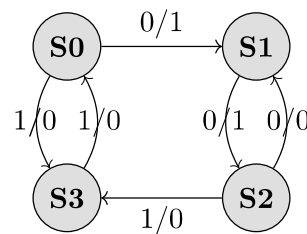
28. (L.O.2) Tại thời điểm 1.0 ns, giá trị của c_out là bao nhiêu:

tại 1ns, a vẫn nhận giá trị bằng 0 do delay 1.0 ==> a=0, b=0
A. 8'b10_xxxx B. 8'b1010_xxxx
C. 8'b1010_0000 D. 8'b0010_0000

29. (L.O.2) a_in = 20; xảy ra tại thời điểm nào

A. 1.0 ns B. 1.123 ns
C. 2.123 ns ☒ D. 1.1234 ns

Thông tin dùng chung từ câu 30 đến 32.



4 trạng thái S1,S2,S3,S4
==> cần 2 bits

nếu input là 0 thì quay về S1, 1 thì lên 3

30. (L.O.3) Có thể dùng bao nhiêu bit lưu trữ để hiện thực máy trạng thái trên.

☒ A. 2 B. 3
C. 4 D. A, B, C đều đúng

31. (L.O.3) Trạng thái kết tiếp của S2 khi input bằng 1 là.

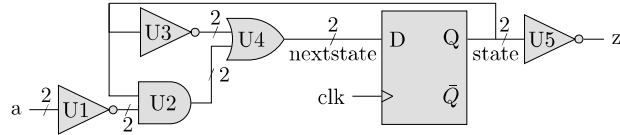
A. S0 B. S1 C. S2 ☒ D. S3

32. (L.O.3) Ở trạng thái nào, việc mô tả chuyển trạng thái là không đầy đủ. 2 trạng thái này không có đủ 2 mũi tên

A. S0, S1 ☒ B. S1,S3
C. S2, S3 D. S0, S2

Thông tin dùng chung từ câu 33 đến 36.

Cho của một máy trạng thái FMS như sau.



Cho delay của primitive như sau:

And	Or	Not	D-FF
4ns	4ns	2ns	8ns

33. (L.O.3) Mạch trên có tối đa bao nhiêu trạng thái hữu hạn.

A. 1 B. 2 C. 3 D. 4

34. (L.O.3) Dựa vào mạch mô tả. Xác định kiểu của máy trạng thái trên:

A. Máy Mealy.
B. Máy Moore.
 C. Không đủ dữ liệu để xác định.
 D. Có thể là Meal hoặc Moore.

nếu ngõ ra từ cấu trúc trực tiếp từ 1 input trở lên: mealy
 còn nếu được cấu thành chỉ thông qua FF thì là moore

35. (L.O.3) Biết $state = 2'b10$; . Xác định state ở chu kỳ tiếp theo khi $a=2'b11$;

A. 00_2 B. 10_2 C. 01_2 D. 11_2

36. (L.O.3) Xác định tần số tối đa để cho máy trạng thái trên hoạt động đúng?

A. ≈ 0.071 GHz B. ≈ 0.100 GHz
 C. ≈ 0.050 GHz D. ≈ 0.625 GHz

Thông tin dùng chung từ câu 37 đến 38.

Thiết kế một mạch có chức năng đếm số lượng sinh viên vào lớp, mạch có 5 nút nhấn, và 1 sensor cảm biến chuyển động 1 bit và hiển thị số sinh viên hiện tại ra led đơn. Biết số lượng sinh viên tối đa là 200.

37. (L.O.4) Mạch trên thuộc loại mạch nào.

A. Tổ hợp B. Tương tự(analog)
C. Tuần tự D. A, B, C đều sai

38. (L.O.4) Khai báo interface nào sau đây thỏa thiết kế trên.

A. `module stu_count(a_in, stu_out);`
 B. `module stu_count(input a, b, c, d, e, f, output [5:0] stu_out);`
 C. `module stu_count(a, b, c, d, e, f, stu_out);`
D. Khai báo A và C đều thỏa.

Thông tin dùng chung từ câu 39 đến 40.

```
module adder(y_out, a_in, b_in);
  parameter SIZE = 4;
  output [SIZE-1:0] y_out;
  input [SIZE-1:0] a_in, b_in;
  generate
    if SIZE < 8:
      slow_adder(a_in, b_in, y_out);
    else:
      faster_adder(a_in, b_in, y_out);
  endgenerate
endmodule
```

39. (L.O.1) Muốn sử dụng module adder mà output y_out có độ rộng 32 bit, thì ta khai báo thực thể của adder như thế nào?

A. `adder U0 (y, a, b);`
 B. `adder #(32) U0(y, a, b);`
 C. `adder #(.SIZE(32)) U0 (y, a, b);`
 D. B và C đều đúng.

40. (L.O.1) Khi khai báo thực thể `adder #(10) U1 (y, a, b);`. Sau quá trình "Elaboration", module adder U1 tổng hợp được chứa phần cứng nào?

A. `slow_adder`
 B. `faster_adder`
 C. Chứa cả `slow_adder` và `faster_adder`
 D. Không chứa cả `slow_adder` và `faster_adder`

— Đề kiểm tra gồm 40 câu hỏi trên 4 trang