

# ESA

## Exam of Skill Assessment

### 「ESA-VerilogDesign」技能試験 問題 (選択式問題) サンプル

問題番号	問1～問40
試験時間	80分

#### 注意事項

1. 解答用紙の記入に当たっては、次の指示に従ってください。指示に従わない場合には、採点されません。
  - (1) 所属名欄  
所属名を記入してください。
  - (2) 従業員番号、氏名欄  
未記入や誤記入のない様に正しく記入して下さい。
  - (3) 解答欄  
文字は採点者にわかるように記入して下さい。
2. 乱丁・落丁がないか、試験開始直後に確認して下さい。もしある場合は速やかに申し出て下さい。
3. 論理合成に関する問題については、問題文に特に指定がない限り Synopsys 社の Design Compiler™ 2000.11-SP2 の使用を前提とします。また、論理合成の結果に関する問題については、TSMC 社 0.25  $\mu$ m スタンドセル(SAGE™)用のライブラリ slow.db を使用し、論理合成時に特別なタイミング制約を与えない、通常のコンパイル結果を正解とします。

指示があるまで開いてはいけません。  
問題に関する質問にはお答えできません。

問1から問40まで「1~5」の5者択一選択式です。

問1 次の演算式と等価の演算式はどれか。

```
wire      A;
wire [3:0] B;
wire      Y;

assign Y = A & (~& B);
```

- 1 . assign Y = A ~& B
- 2 . assign Y = A ~&& B;
- 3 . assign Y = A &~& B;
- 4 . assign Y = A & ~( B[0] & B[1] & B[2] & B[3] );
- 5 . assign Y = A & ( B[0] ~& B[1] ~& B[2] ~& B[3] );

問2 入力 SEL, D0, D1 が次の値の時、演算式の出力 DOUT はどのような値になるか。

```
SEL ... 1'bx
D0  ... 4'b0011
D1  ... 4'b1001

wire [3:0] D0, D1, DOUT;
wire      SEL;
assign DOUT = (SEL==1'b0) ? D0: D1;
```

- 1 . 4'b0011
- 2 . 4'b1001
- 3 . 4'bxxxx
- 4 . 4'bx0x1
- 5 . シミュレータに依存するので確定できない。

**問3** FFを生成する always 文では、次の RTL 記述のように、代入文に遅延を設定したほうがよい。遅延を設定することの理由として適切なものはどれか。

```
parameter DELAY = 1;
always @(posedge CLK) begin
    Q <= #DELAY DATA;
end
```

1. 適切な遅延値を指定すれば、RTL の論理合成が不要になるため。
2. 論理合成後のゲートレベル・シミュレーションでも、ここで設定した遅延値が有効になるため。
3. 論理合成ツールに FF だということを正しく認識させるため。
4. RTL シミュレーションの際、0 時間での遅延によるレーシングの問題を防止するため。
5. 論理合成後の回路のゲートレベル遅延時間を想定すると、論理合成ツールの制約条件として有効になるため。

問4 case文で記述した次の記述のうち、論理合成を行ったときにラッチが生成されるのはどれか。その組み合わせを答えよ。

< 共通部分 >

```
module SEL8( A, SEL, Y );
input  [7:0] A;
input  [2:0] SEL;
output Y;
reg    Y;

    always @(A or SEL) begin

        // A ~ C の case 文

    end

endmodule
```

<p>A.</p> <pre>case (SEL)   3'b000 : Y = A[0];   3'b001 : Y = A[1];   3'b010 : Y = A[2];   3'b011 : Y = A[3];   3'b100 : Y = A[4];   3'b101 : Y = A[5];   3'b110 : Y = A[6]; endcase</pre>	<p>B.</p> <pre>case (SEL)   3'b000 : Y = A[0];   3'b001 : Y = A[1];   3'b010 : Y = A[2];   3'b011 : Y = A[3];   3'b100 : Y = A[4];   3'b101 : Y = A[5];   3'b110 : Y = A[6];   3'b111 : Y = A[7]; endcase</pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

C.

```
case (SEL)
  3'b000 : Y = A[0];
  3'b001 : Y = A[1];
  3'b010 : Y = A[2];
  3'b011 : Y = A[3];
  3'b100 : Y = A[4];
  3'b101 : Y = A[5];
  3'b110 : Y = A[6];
  default : Y = 1'bX;
endcase
```

- 1 . A
- 2 . B
- 3 . A , B
- 4 . A , C
- 5 . A , B , C

問5 8進カウンタの動作を検証するために次のようなテストベンチ記述を用意した。選択肢の中で、動作結果がシミュレータに依存しない安全な記述はどれか。

< テストベンチ記述 >

```

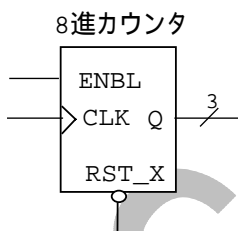
module CNT8_TEST;
parameter CYCLE=10000;
parameter HALF_CYCLE=5000;
parameter STB=1000;

CNT8 CNT8(CLK,RST_X,ENBL,Q);

always begin // クロック発生
    CLK=1'b1; #HALF_CYCLE;
    CLK=1'b0; #HALF_CYCLE;
end
...
// 選択肢の記述
...
endmodule

```

< 検証対象の回路 >



真理値表

RST_X	CLK	ENBL	Q
0	X	X	0
1		0	$Q_0$
1		1	$Q_0+1$

注:  $Q_0$  はクロック立ち上がり直前の Q の値を示す

注: x は don't care を示す

```

module CNT8(CLK,RST_X,ENBL,Q);
input  CLK,RST_X,ENBL;
output [2:0] Q;
reg    [2:0] Q;
parameter DELAY=1;

always @(posedge CLK or negedge RST_X)
    if(RST_X==1'b0)
        Q <= #DELAY 3'd0;
    else if(ENBL==1'b1)
        Q <= #DELAY Q + 1;

endmodule

```

## &lt; 選択肢 &gt;

1. 

```
initial begin
    RST_X=1'b1; ENBL=1'b0;
    @(posedge CLK) RST_X=1'b0;
    @(posedge CLK) RST_X=1'b1;
    @(posedge CLK) ENBL=1'b1;
    ...
end
```
2. 

```
initial begin
    RST_X=1'b1; ENBL=1'b0;
    @(posedge CLK) #STB RST_X=1'b0;
    @(posedge CLK) #STB RST_X=1'b1;
    @(posedge CLK) #STB ENBL=1'b1;
    ...
end
```
3. 

```
initial begin
    RST_X=1'b1; ENBL=1'b0;
    #CYCLE RST_X=1'b0;
    #CYCLE RST_X=1'b1;
    #CYCLE ENBL=1'b1;
    ...
end
```
4. 

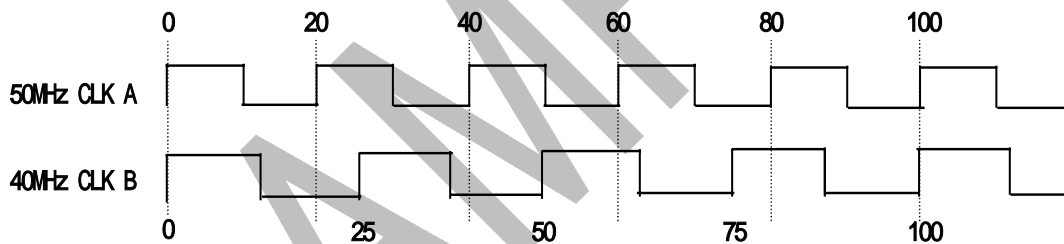
```
initial begin
    RST_X=1'b1; ENBL=1'b0;
    #STB;
    #CYCLE RST_X=1'b0;
    #CYCLE RST_X=1'b1;
    #CYCLE ENBL=1'b1;
    ...
end
```
5. 

```
initial begin
    RST_X=1'b1; ENBL=1'b0;
    #CYCLE RST_X=1'b0;
    #CYCLE RST_X=1'b1;
    #STB;
    #CYCLE ENBL=1'b1;
    ...
end
```

問6 単相同期回路において、クロック周期が  $20.0\text{ns}$ 、FF のセットアップタイムが  $1.2\text{ns}$ 、FF のホールドタイムが  $0.8\text{ns}$ 、クロックスキューが最大  $2.0\text{ns}$  のとき、各 FF 間で許される回路及び配線の遅延量は何  $\text{ns}$  か。

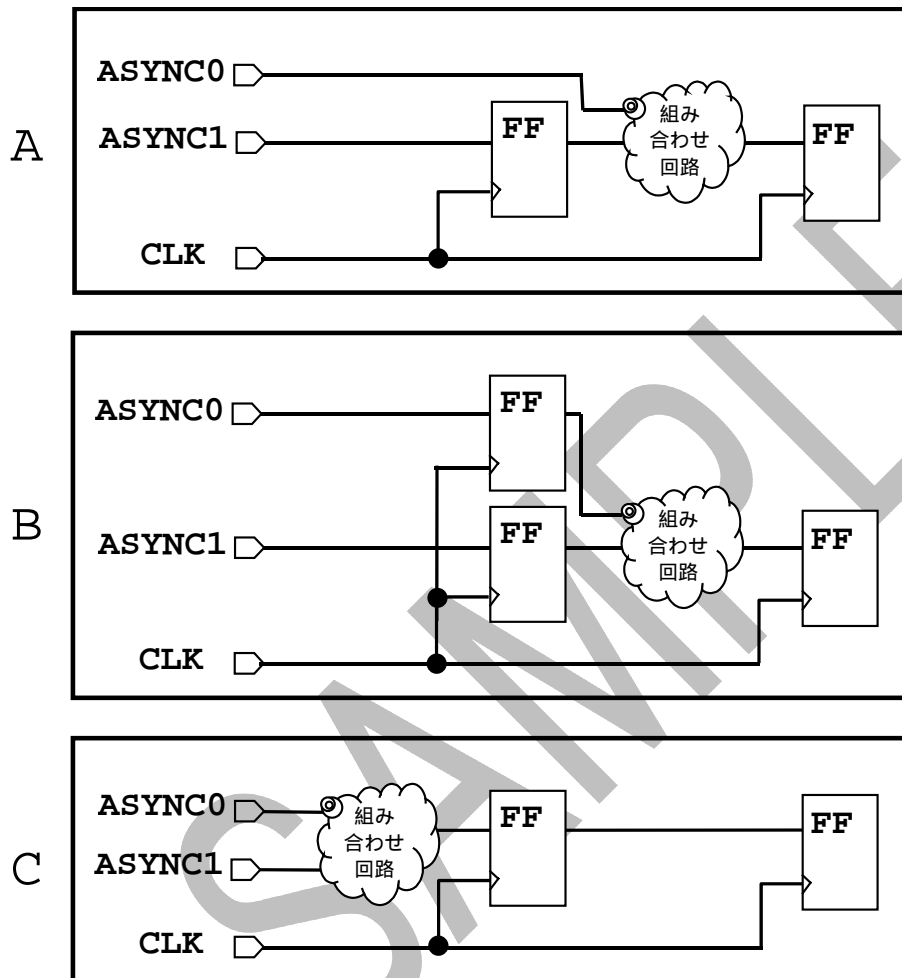
- 1 .  $16.0\text{ns}$
- 2 .  $16.8\text{ns}$
- 3 .  $17.2\text{ns}$
- 4 .  $18.0\text{ns}$
- 5 .  $18.8\text{ns}$

問7  $50\text{MHz}$  のクロック A で動作する FF から、 $40\text{MHz}$  のクロック B で動作する FF に渡る、非同期クロック間のパスがある。このパスに次のような波形の信号を論理合成ツールにタイミング制約として与えたとき、A ~ B 間は何  $\text{ns}$  の制約がかかっていることになるか。2つの FF はクロックの立ち上がりで動作するものとし、制約はセットアップタイム、ホールドタイム、クロックスキュー、などクロック周期以外の要因は全て無視するものとする。



- 1 .  $5\text{ns}$
- 2 .  $20\text{ns}$
- 3 .  $25\text{ns}$
- 4 .  $100\text{ns}$
- 5 . 無限大

問 8 次の図の後段の FF の出力には、メタステーブルが発生する可能性がある。その発生確率を順に並べた正しい組み合わせはどれか。ただし、クロック CLK の周波数は 50MHz で動作しているものとし、入力 ASYNC0, ASYNC1 には、CLK とは異なるクロックで駆動される FF からデータが非同期で入力されるものとする。



発生確率  
高 低

1. A, B, C
2. A, C, B
3. B, A, C
4. C, A, B
5. C, B, A

問9 100万ゲート規模のASICを、 $0.18\mu\text{m}$ あるいは $0.14\mu\text{m}$ の半導体プロセスで作成する場合、論理合成後ではタイミング制約を満たしていたパスが、レイアウト後にタイミングエラーを多量に発生することが問題になる。レイアウト後のタイミングエラーを少なくするための正しい論理合成の方針はどれか。

1. すべてのセルのドライブ能力を強化する。
2. クロック周期のタイミング制約に、実際の周期より10%くらい少ない値を採用する。仮にその周期でタイミング違反があっても、実際の周期で解析を行いタイミングを満たしていれば構わない。
3. クロック周期のタイミング制約に、実際の周期より10%くらい少ない値を採用する。その周期で解析を行った結果、タイミング違反になるパスは、RTL記述を修正し、論理段数が少なくなるようにする。
4. タイミングエラーをなくすよりも、面積を削減する。
5. `wire_load`(配線遅延)モデルを見直し、配線遅延値を大きく見積もる。