

論理回路入門（13）

順序回路 4：同期式 N 進カウンターの設計

李 亜民

2025 年 12 月 23 日 (火)

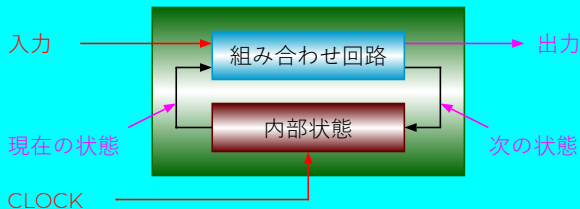
同期式N進カウンターの設計

ポイント

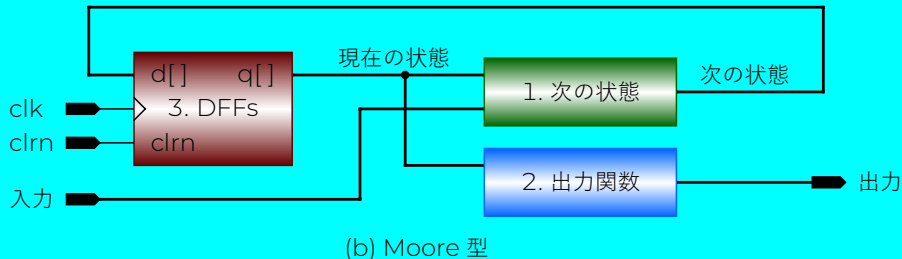
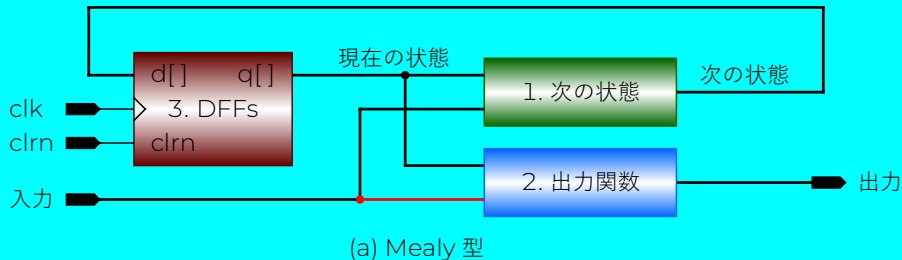
- 同期式N進カウンター
 - ▶ DFF で同期式N進カウンター
 - ▶ JKFF で同期式N進カウンター
 - ▶ TFF で同期式N進カウンター
- 非同期式カウンター
- 非同期式 FIFO

順序回路

- 順序回路は内部状態と入力信号で出力が決まる回路である。
- 順序回路の構成:
 - 1 記憶素子 (D フリップフロップなど)
 - ★ 現在の状態を記憶しておくため
 - 2 組み合わせ回路
 - ★ 状態遷移関数 (次の状態)
 - ★ 出力関数



順序回路 — Mealy 型と Moore 型



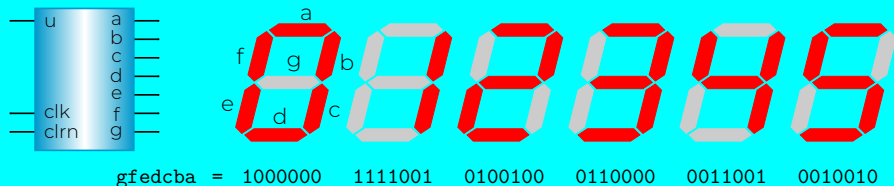
順序回路設計の手順

- 1 問題を理解する。
- 2 状態遷移図をつくる。
- 3 FF の数を決める ($n = \lceil \log_2 N \rceil$ 、 N は状態の数)。
- 4 各状態に n ビットの番号を付け、真理値表をつくる。
 - ▶ 次の状態の真理値表をつくる。
 - ▶ 出力関数の真理値表をつくる。
- 5 真理値表から論理式をつくる (どの条件で出力が 1 になるか)。
 - ▶ カルノー図を用いて論理式を簡単化する。
- 6 論理式から回路をつくる。
- 7 テストベンチをつくる (シミュレーションするため)。
- 8 回路をシミュレーションする (回路設計の正当性検証)。
- 9 与えられた回路の動作を理解する (波形の説明)。

同期式N進カウンターの設計

問題: 7セグメント LED を使いカウントアップ / カウントダウンする回路を作成しよう。

- カウント範囲: 0 ~ 5
- $u = 1$ (カウントアップ): 0, 1, 2, 3, 4, 5, 0, 1, 2, 3, ...
- $u = 0$ (カウントダウン): 0, 5, 4, 3, 2, 1, 0, 5, 4, 3, ...
- 7セグメント LED で表示する (0: 点灯; 1: 消灯)



復習：フリップフロップ — DFF

```
`timescale 1ns/1ns
module dff    (CLK, CLRN, E, D, Q);
    input     D;
    input     CLK, CLRN, E;
    output    Q;
    reg       Q;
    always @(posedge CLK or negedge CLRN) begin
        if (CLRN == 0) begin
            Q <= 0;
        end else begin
            Q <= D;
        end
    end
end
endmodule
```

DFF：CLKの立ち上がりで、 $Q_{\text{next}} = D$

[dff.v](#)

復習：フリップフロップ — JKFF

```
`timescale 1ns/1ns
module jkff (CLK, CLRN, E, J, K, Q);
    input    J, K;
    input    CLK, CLRN, E;
    output   Q;
    reg      Q;
    always @(posedge CLK or negedge CLRN) begin
        if (CLRN == 0) begin
            Q <= 0;
        end else begin
            Q <= J & ~Q | ~K & Q;
        end
    end
end
endmodule
```

JKFF : CLK の立ち上がりで、 $Q_{\text{next}} = J \overline{Q} + \overline{K} Q$

jkff.v

復習：フリップフロップ—TFF

```
`timescale 1ns/1ns
module tff  (CLK, CLRN, E, T, Q);
    input    T;
    input    CLK, CLRN, E;
    output   Q;
    reg      Q;
    always @(posedge CLK or negedge CLRN) begin
        if (CLRN == 0) begin
            Q <= 0;
        end else begin
            Q <= T & ~Q | ~T & Q;
        end
    end
end
endmodule
```

TFF: CLK の立ち上がりで、 $Q_{\text{next}} = T\overline{Q} + \overline{T}Q$

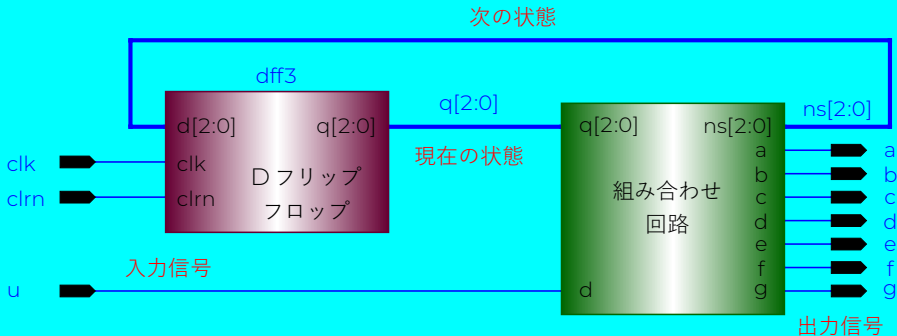
[tff.v](#)

DFF で同期式 N 進 カウンターの設計

DFF で同期式 N 進カウンターの設計

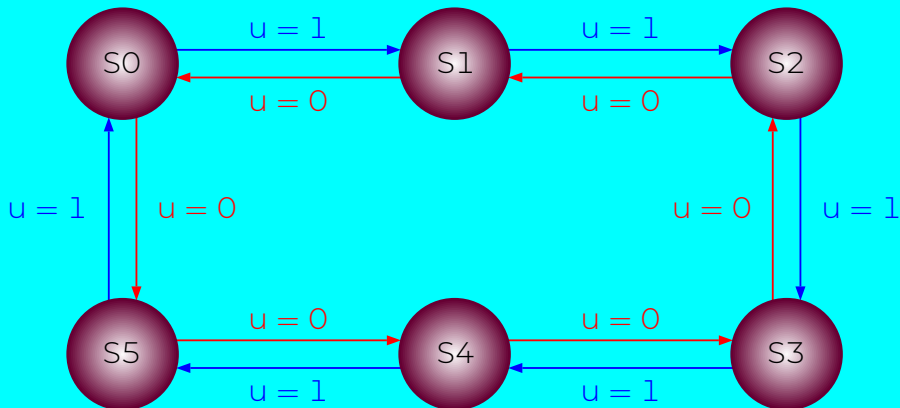
0 ~ 5 カウンター:

- 状態数: 6 (0, 1, 2, 3, 4, 5)
- 3 ビット DFF を使う ($\lceil \log_2 6 \rceil = 3$)
- カウント全体の回路:



DFF で同期式 N 進カウンターの設計

0 ~ 5 カウンター: 状態遷移図:



DFF で同期式 N 進カウンターの設計

現在の状態				入力	次の状態			
	q[2]	q[1]	q[0]	u		ns[2]	ns[1]	ns[0]
S0	0	0	0	1	S1	0	0	1
				0	S5	1	0	1
S1	0	0	1	1	S2	0	1	0
				0	S0	0	0	0
S2	0	1	0	1	S3	0	1	1
				0	S1	0	0	1
S3	0	1	1	1	S4	1	0	0
				0	S2	0	1	0
S4	1	0	0	1	S5	1	0	1
				0	S3	0	1	1
S5	1	0	1	1	S0	0	0	0
				0	S4	1	0	0
入力として	1	1	0	X	Don't care	X	X	X
存在しない	1	1	1	X	ドントケア	X	X	X

DFF で同期式 N 進カウンターの設計

q[2] 0 0 1 1
q[1] 0 1 1 0
q[0] u

00	1		x	
01			x	1
11		1	x	
10			x	1

ns[2]

q[2] 0 0 1 1
q[1] 0 1 1 0
q[0] u

00			x	1
01		1	x	
11	1		x	
10		1	x	

ns[1]

q[2] 0 0 1 1
q[1] 0 1 1 0
q[0] u

00	1	1	x	1
01	1	1	x	1
11			x	
10			x	

ns[0]

	q[2]	q[1]	q[0]	u		ns[2]	ns[1]	ns[0]
S0	0	0	0	1	S1	0	0	1
				0	S5	1	0	1
S1	0	0	1	1	S2	0	1	0
				0	S0	0	0	0
S2	0	1	0	1	S3	0	1	1
				0	S1	0	0	1
S3	0	1	1	1	S4	1	0	0
				0	S2	0	1	0
S4	1	0	0	1	S5	1	0	1
				0	S3	0	1	1
S5	1	0	1	1	S0	0	0	0
				0	S4	1	0	0
存在しない	1	1	x	x	ドントケア	x	x	x

DFF で同期式N進カウンターの設計

		q[2]	0	0	1	1
		q[1]	0	1	1	0
q[0] u						
00		1		x		
01				x	1	
11			1	x		
10				x	1	

ns[2]

		q[2]	0	0	1	1
		q[1]	0	1	1	0
q[0] u						
00				x	1	
01			1	x		
11	1			x		
10			1	x		

ns[1]

		q[2]	0	0	1	1
		q[1]	0	1	1	0
q[0] u						
00	1	1	x	1		
01	1	1	x	1		
11			x			
10			x			

ns[0]

ns[2] =

DFF で同期式 N 進カウンターの設計

		q[2]	0	0	1	1
		q[1]	0	1	1	0
q[0] u						
00	1			x		
01				x	1	
11		1	x			
10				x	1	

ns[2]

		q[2]	0	0	1	1
		q[1]	0	1	1	0
q[0] u						
00				x	1	
01		1	x			
11	1			x		
10		1	x			

ns[1]

		q[2]	0	0	1	1
		q[1]	0	1	1	0
q[0] u						
00	1	1	x	1		
01	1	1	x	1		
11			x			
10			x			

ns[0]

$$ns[2] = \overline{q[2]} \overline{q[1]} \overline{q[0]} \overline{u} + q[1] q[0] u + q[2] \overline{q[0]} u + q[2] q[0] \overline{u}$$

$$ns[1] =$$

DFF で同期式 N 進カウンターの設計

	q[2]	0	0	1	1
	q[1]	0	1	1	0
q[0] u					
00	1		x		
01			x	1	
11		1	x		
10			x	1	

ns[2]

	q[2]	0	0	1	1
	q[1]	0	1	1	0
q[0] u					
00			x	1	
01		1	x		
11	1		x		
10		1	x		

ns[1]

	q[2]	0	0	1	1
	q[1]	0	1	1	0
q[0] u					
00	1	1	x	1	
01	1	1	x	1	
11			x		
10			x		

ns[0]

$$ns[2] = \overline{q[2]} \overline{q[1]} \overline{q[0]} \overline{u} + q[1] q[0] u + q[2] \overline{q[0]} u + q[2] q[0] \overline{u}$$

$$ns[1] = \overline{q[2]} \overline{q[1]} q[0] u + q[1] \overline{q[0]} u + q[1] q[0] \overline{u} + q[2] \overline{q[0]} \overline{u}$$

$$ns[0] =$$

DFF で同期式 N 進カウンターの設計

	q[2]	0	0	1	1
	q[1]	0	1	1	0
q[0] u					
00	1		x		
01			x	1	
11		1	x		
10			x	1	

ns[2]

	q[2]	0	0	1	1
	q[1]	0	1	1	0
q[0] u					
00			x	1	
01		1	x		
11	1		x		
10		1	x		

ns[1]

	q[2]	0	0	1	1
	q[1]	0	1	1	0
q[0] u					
00	1	1	x	1	
01	1	1	x	1	
11			x		
10			x		

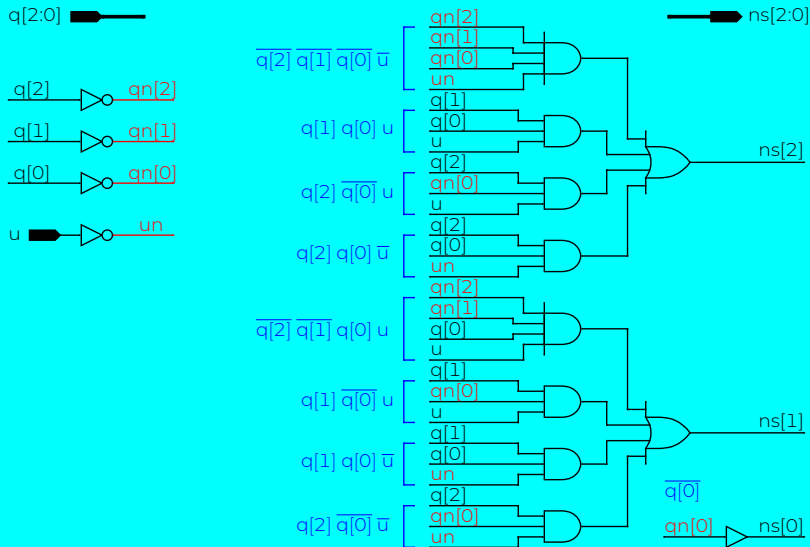
ns[0]

$$ns[2] = \overline{q[2]} \overline{q[1]} \overline{q[0]} \overline{u} + q[1] q[0] u + q[2] \overline{q[0]} u + q[2] q[0] \overline{u}$$

$$ns[1] = \overline{q[2]} \overline{q[1]} q[0] u + q[1] \overline{q[0]} u + q[1] q[0] \overline{u} + q[2] \overline{q[0]} \overline{u}$$

$$ns[0] = \overline{q[0]}$$

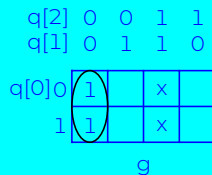
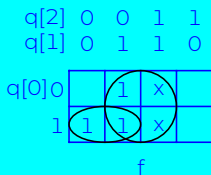
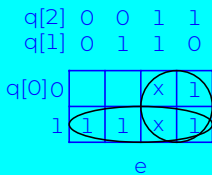
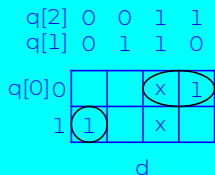
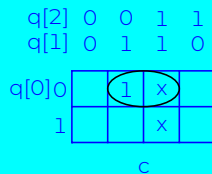
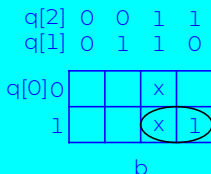
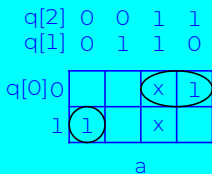
DFF で同期式N進カウンターの設計



DFF で同期式 N 進カウンターの設計

0 ~ 5 カウンター: 出力 (0: 点灯; 1: 消灯):

q[2:0]	gfedcba
000	1000000
001	1111001
010	0100100
011	0110000
100	0011001
101	0010010



DFF で同期式 N 進カウンターの設計

0 ~ 5 カウンター: 出力 (0: 点灯; 1: 消灯):

q[2:0]	gfedcba
000	1000000
001	1111001
010	0100100
011	0110000
100	0011001
101	0010010

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1		x	

a

$$a = \overline{q[2]} \overline{q[1]} q[0] + q[2] \overline{q[0]}$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	
1			x	1

b

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0	1	x	
1			x	

c

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1		x	

d

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1	1	x	1

e

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0	1	x	
1	1	1	x	

f

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	1		x	
1	1		x	

g

DFF で同期式 N 進カウンターの設計

0 ~ 5 カウンター: 出力 (0: 点灯; 1: 消灯):

q[2:0]	gfedcba
000	1000000
001	1111001
010	0100100
011	0110000
100	0011001
101	0010010

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1		x	

a

$$a = \overline{q[2]} \overline{q[1]} q[0] + q[2] \overline{q[0]}$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	
1			x	1

b

$$b = q[2] q[0]$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0	1	x	
1			x	

c

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1		x	

d

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1	1	x	1

e

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0	1	x	
1	1	1	x	

f

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	1		x	
1	1		x	

g

DFF で同期式 N 進カウンターの設計

0 ~ 5 カウンター: 出力 (0: 点灯; 1: 消灯):

q[2:0]	gfedcba
000	1000000
001	1111001
010	0100100
011	0110000
100	0011001
101	0010010

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1		x	

a

$$a = \overline{q[2]} \overline{q[1]} q[0] + q[2] \overline{q[0]}$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	
1			x	1

b

$$b = q[2] q[0]$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0	1	x	
1			x	

c

$$c = q[1] \overline{q[0]}$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1		x	

d

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1	1	x	1

e

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0	1	x	
1	1	1	x	

f

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	1		x	
1	1		x	

g

DFF で同期式 N 進カウンターの設計

0 ~ 5 カウンター: 出力 (0: 点灯; 1: 消灯):

q[2:0]	gfedcba
000	1000000
001	1111001
010	0100100
011	0110000
100	0011001
101	0010010

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1		x	

a

$$a = \overline{q[2]} \overline{q[1]} q[0] + q[2] \overline{q[0]}$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	
1			x	1

b

$$b = q[2] q[0]$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0	1	x	
1			x	

c

$$c = q[1] \overline{q[0]}$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1		x	

d

$$d = a$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1	1	x	1

e

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0	1	x	
1	1	1	x	

f

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	1		x	
1	1		x	

g

DFF で同期式 N 進カウンターの設計

0 ~ 5 カウンター: 出力 (0: 点灯; 1: 消灯):

q[2:0]	gfedcba
000	1000000
001	1111001
010	0100100
011	0110000
100	0011001
101	0010010

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1		x	

a

$$a = \overline{q[2]} \overline{q[1]} q[0] + q[2] \overline{q[0]}$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	
1			x	1

b

$$b = q[2] q[0]$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0	1	x	
1			x	

c

$$c = q[1] \overline{q[0]}$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1		x	

d

$$d = a$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1	1	x	1

e

$$e = q[2] + q[0]$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0	1	x	
1	1	1	x	

f

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	1		x	
1	1		x	

g

DFF で同期式 N 進カウンターの設計

0 ~ 5 カウンター: 出力 (0: 点灯; 1: 消灯):

q[2:0]	gfedcba
000	1000000
001	1111001
010	0100100
011	0110000
100	0011001
101	0010010

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1		x	

a

$$a = \overline{q[2]} \overline{q[1]} q[0] + q[2] \overline{q[0]}$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	
1			x	1

b

$$b = q[2] q[0]$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0	1	x	
1			x	

c

$$c = q[1] \overline{q[0]}$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1		x	

d

$$d = a$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1	1	x	1

e

$$e = q[2] + q[0]$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0	1	x	
1	1	1	x	

f

$$f = \overline{q[2]} q[0] + q[1]$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	1		x	
1	1		x	

g

DFF で同期式 N 進カウンターの設計

0 ~ 5 カウンター: 出力 (0: 点灯; 1: 消灯):

q[2:0]	gfedcba
000	1000000
001	1111001
010	0100100
011	0110000
100	0011001
101	0010010

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1		x	

a

$$a = \overline{q[2]} \overline{q[1]} q[0] + q[2] \overline{q[0]}$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	
1			x	1

b

$$b = q[2] q[0]$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0	1	x	
1			x	

c

$$c = q[1] \overline{q[0]}$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1		x	

d

$$d = a$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0		x	1
1	1	1	x	1

e

$$e = q[2] + q[0]$$

q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	0	1	x	
1	1	1	x	

f

$$f = \overline{q[2]} q[0] + q[1]$$

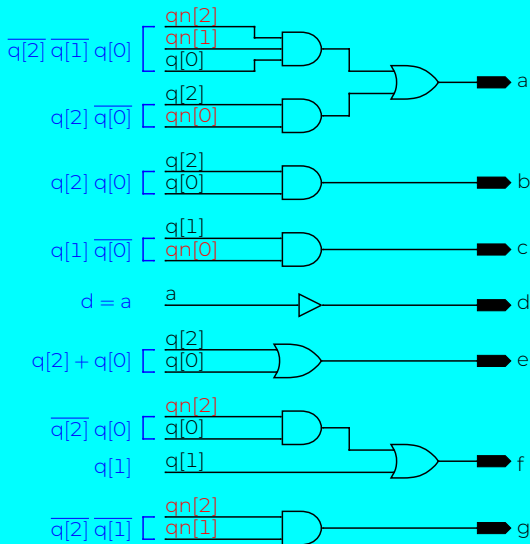
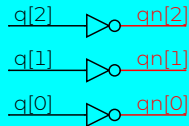
q[2]	0	0	1	1
q[1]	0	1	1	0
q[0]	1		x	
1	1		x	

g

$$g = \overline{q[2]} \overline{q[1]}$$

DFF で同期式 N 進カウンターの設計

q[2:0] 



DFF で同期式 N 進カウンターの設計

```
`timescale 1ns/1ns
module counter6_d (clk, clrn, u, a, b, c, d, e, f, g);
    input  clk, clrn, u;
    output a, b, c, d, e, f, g; // outputs

    reg [2:0] q; // current state
    wire [2:0] ns; // next state
    // 1. next state ----- 1
    assign ns[2] = ~q[2]&~q[1]&~q[0]&~u | q[1]&q[0]&u | q[2]&~q[0]&u | q[2]&q[0]&~u;
    assign ns[1] = ~q[2]&~q[1]&q[0]&u | q[1]&~q[0]&u | q[1]&q[0]&~u | q[2]&~q[0]&~u;
    assign ns[0] = ~q[0];
    // 2. outputs ----- 2
    assign a = ~q[2] & ~q[1] & q[0] | q[2] & ~q[0];
    assign b = q[2] & q[0];
    assign c = q[1] & ~q[0];
    assign d = a;
    assign e = q[2] | q[0];
    assign f = ~q[2] & q[0] | q[1];
    assign g = ~q[2] & ~q[1];
    // 3. dffs ----- 3
    always @(posedge clk or negedge clrn) begin // dffs
        if (clrn == 0) begin
            q <= 0;
        end else begin
            q <= ns;
        end
    end
endmodule
```

$$\begin{aligned} ns[2] &= \overline{q[2]} \overline{q[1]} \overline{q[0]} \overline{u} + q[1] q[0] u + q[2] \overline{q[0]} u + q[2] q[0] \overline{u} \\ ns[1] &= \overline{q[2]} \overline{q[1]} q[0] u + q[1] \overline{q[0]} u + q[1] q[0] \overline{u} + q[2] \overline{q[0]} \overline{u} \\ ns[0] &= \overline{q[0]} \\ a &= \overline{q[2]} \overline{q[1]} q[0] + q[2] \overline{q[0]} \\ b &= q[2] q[0] \\ c &= q[1] \overline{q[0]} \\ d &= a \\ e &= q[2] + q[0] \\ f &= \overline{q[2]} q[0] + q[1] \\ g &= \overline{q[2]} \overline{q[1]} \end{aligned}$$

[counter6_d.v](#)

DFF で同期式N進カウンターの設計

```
`timescale 1ns/1ns
module counter6_d_tb;
    reg  clk, clrn, u;
    wire a, b, c, d, e, f, g;

    counter6_d i0 (clk, clrn, u, a, b, c, d, e, f, g);

    initial begin
        #0  clk = 1; clrn = 0; u = 1;
        #1  clrn = 1;
        #18 u = 0;
        #25 $finish;
    end

    always #1 clk = ~clk;

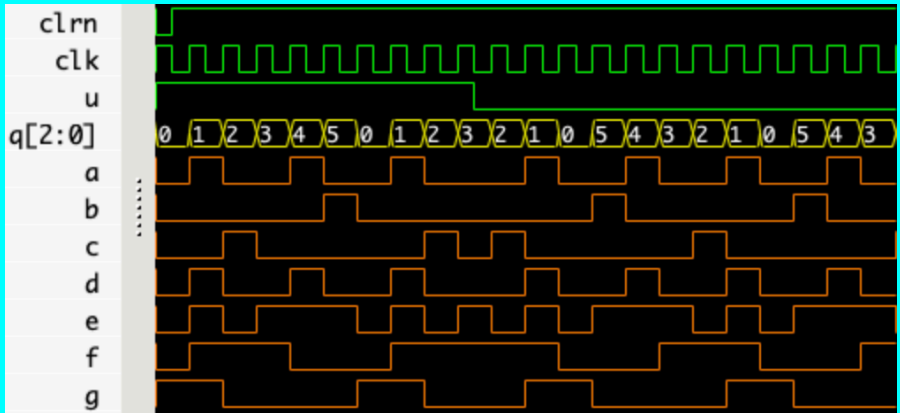
    initial begin
        $dumpfile ("counter6_d.vcd");
        $dumpvars;
    end

endmodule
```

[counter6_d_tb.v](#)

DFF で同期式 N 進カウンターの設計

0 ~ 5 カウンター: シミュレーション波形:



[counter6_d_tb.v](#) Count Up

Count Down

J K F F で同期式 N 進 カウンターの設計

JKFF で同期式 N 進カウンターの設計

- JKFF: CLK の立ち上がりで、 $Q_{\text{next}} = J \bar{Q} + \bar{K} Q \dots\dots\dots (= ns)$

ns から jk を導出する。

例： $ns = A \bar{q} + B q + C$

C を注意

$$= A \bar{q} + B q + C (\bar{q} + q)$$

$$= A \bar{q} + B q + C \bar{q} + C q$$

$$= (A + C) \bar{q} + (B + C) q$$

$$= j \bar{q} + \bar{k} q$$

▶ $j = A + C$

C は j と k の両方

▶ $k = \overline{B + C}$

の論理式に現れる

JKFF で同期式 N 進カウンターの設計

- JKFF: CLK の立ち上がりで、 $Q_{\text{next}} = J \overline{Q} + \overline{K} Q \dots\dots\dots (= ns)$

JKFF で同期式 N 進カウンターの設計

- JKFF: CLK の立ち上がりで、 $Q_{\text{next}} = J \overline{Q} + \overline{K} Q \dots\dots\dots (= ns)$
- $ns[0] = \overline{q[0]} = 1 \cdot \overline{q[0]} + 0 \cdot q[0] = j[0] \cdot \overline{q[0]} + \overline{k[0]} \cdot q[0]$
 - ▶ $j[0] = 1 \dots\dots\dots j[0] \overline{q[0]}$
 - ▶ $k[0] = \overline{0} = 1 \dots\dots\dots \overline{k[0]} q[0]$

JKFF で同期式 N 進カウンターの設計

- JKFF: CLK の立ち上がりで、 $Q_{\text{next}} = J \overline{Q} + \overline{K} Q \dots\dots\dots (= ns)$
- $ns[0] = \overline{q[0]} = 1 \cdot \overline{q[0]} + 0 \cdot q[0] = j[0] \cdot \overline{q[0]} + \overline{k[0]} \cdot q[0]$
 - ▶ $j[0] = 1 \dots\dots\dots j[0] \overline{q[0]}$
 - ▶ $k[0] = \overline{0} = 1 \dots\dots\dots \overline{k[0]} q[0]$
- $ns[1] = \overline{q[2]} \overline{q[1]} q[0] u + q[1] \overline{q[0]} u + q[1] q[0] \overline{u} + q[2] \overline{q[0]} \overline{u}$
 - ▶ $j[1] = \overline{q[2]} q[0] u + q[2] \overline{q[0]} \overline{u} \dots\dots\dots j[1] \overline{q[1]}$
 - ▶ $k[1] = \overline{\overline{q[0]} u + q[0] \overline{u} + q[2] \overline{q[0]} \overline{u}} \dots\dots\dots \overline{k[1]} q[1]$

JKFF で同期式 N 進カウンターの設計

- JKFF: CLK の立ち上がりで、 $Q_{\text{next}} = J \overline{Q} + \overline{K} Q \dots\dots\dots (= ns)$
- $ns[0] = \overline{q[0]} = 1 \cdot \overline{q[0]} + 0 \cdot q[0] = j[0] \cdot \overline{q[0]} + \overline{k[0]} \cdot q[0]$
 - ▶ $j[0] = 1 \dots\dots\dots j[0] \overline{q[0]}$
 - ▶ $k[0] = \overline{0} = 1 \dots\dots\dots \overline{k[0]} q[0]$
- $ns[1] = \overline{q[2]} \overline{q[1]} q[0] u + q[1] \overline{q[0]} u + q[1] q[0] \overline{u} + q[2] \overline{q[0]} \overline{u}$
 - ▶ $j[1] = \overline{q[2]} q[0] u + q[2] \overline{q[0]} \overline{u} \dots\dots\dots j[1] \overline{q[1]}$
 - ▶ $k[1] = \overline{\overline{q[0]} u + q[0] \overline{u} + q[2] \overline{q[0]} \overline{u}} \dots\dots\dots \overline{k[1]} q[1]$
- $ns[2] = \overline{q[2]} \overline{q[1]} \overline{q[0]} \overline{u} + q[1] q[0] u + q[2] \overline{q[0]} u + q[2] q[0] \overline{u}$
 - ▶ $j[2] = \overline{q[1]} \overline{q[0]} \overline{u} + q[1] q[0] u \dots\dots\dots j[2] \overline{q[2]}$
 - ▶ $k[2] = \overline{q[1] q[0] u + \overline{q[0]} u + q[0] \overline{u}} \dots\dots\dots \overline{k[2]} q[2]$

JKFF で同期式N進カウンターの設計

```
`timescale 1ns/1ns
module counter6_jk (clk, clrn, u, a, b, c, d, e, f, g);
    input  clk, clrn, u;
    output a, b, c, d, e, f, g; // outputs

    reg [2:0] q;    // current state
    wire [2:0] j, k; // next state
    // 1. next state ----- 1
    assign j[0] = 1;
    assign k[0] = 1;
    assign j[1] = ~q[2] & q[0] & u | q[2] & ~q[0] & ~u;
    assign k[1] = ~(~q[0] & u | q[0] & ~u | q[2] & ~q[0] & ~u);
    assign j[2] = ~q[1] & ~q[0] & ~u | q[1] & q[0] & u;
    assign k[2] = ~(q[1] & q[0] & u | ~q[0] & u | q[0] & ~u);
    // 2. outputs ----- 2
    assign a = ~q[2] & ~q[1] & q[0] | q[2] & ~q[0];
    assign b = q[2] & q[0];
    assign c = q[1] & ~q[0];
    assign d = a;
    assign e = q[2] | q[0];
    assign f = ~q[2] & q[0] | q[1];
    assign g = ~q[2] & ~q[1];
    // 3. jkffs ----- 3
    always @(posedge clk or negedge clrn) begin // jkffs
        if (clrn == 0) begin
            q <= 0;
        end else begin
            q <= j & ~q | ~k & q;
        end
    end
endmodule
```

$j[0] = 1$
 $k[0] = 1$
 $j[1] = \overline{q[2]} q[0] u + q[2] \overline{q[0]} \overline{u}$
 $k[1] = \overline{q[0]} u + q[0] \overline{u} + q[2] \overline{q[0]} \overline{u}$
 $j[2] = \overline{q[1]} \overline{q[0]} \overline{u} + q[1] q[0] u$
 $k[2] = q[1] q[0] u + \overline{q[0]} u + q[0] \overline{u}$
 $a = \overline{q[2]} \overline{q[1]} q[0] + q[2] \overline{q[0]}$
 $b = q[2] q[0]$
 $c = q[1] \overline{q[0]}$
 $d = a$
 $e = q[2] + q[0]$
 $f = \overline{q[2]} q[0] + q[1]$
 $g = q[2] \overline{q[1]}$

counter6_jk.v

JKFF で同期式 N 進カウンターの設計

```
`timescale 1ns/1ns
module counter6_jk_tb;
    reg  clk, clrn, u;
    wire a, b, c, d, e, f, g;

    counter6_jk i0 (clk, clrn, u, a, b, c, d, e, f, g);

    initial begin
        #0  clk = 1; clrn = 0; u = 1;
        #1  clrn = 1;
        #18 u = 0;
        #25 $finish;
    end

    always #1 clk = ~clk;

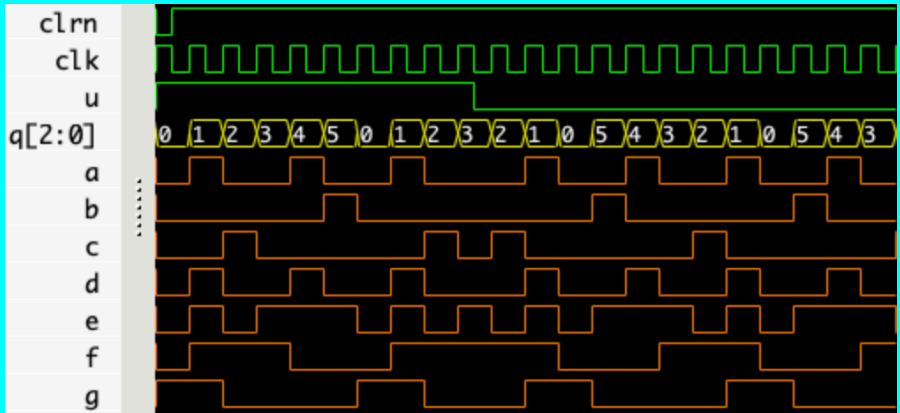
    initial begin
        $dumpfile ("counter6_jk.vcd");
        $dumpvars;
    end

endmodule
```

[counter6_jk_tb.v](#)

JKFF で同期式N進カウンターの設計

0 ~ 5 カウンター: シミュレーション波形:



[counter6_jk_tb.v](#)

Count Up

Count Down

TFF で同期式 N 進 カウンターの設計

TFF で同期式 N 進カウンターの設計

0 ~ 5 カウンター: 状態遷移表 (t = 1: Toggle)

現在の状態				入力	次の状態						
	q[2]	q[1]	q[0]	u		ns[2]	ns[1]	ns[0]	t[2]	t[1]	t[0]
S0	0	0	0	1	S1	0	0	1	0	0	1
				0	S5	1	0	1	1	0	1
S1	0	0	1	1	S2	0	1	0	0	1	1
				0	S0	0	0	0	0	0	1
S2	0	1	0	1	S3	0	1	1	0	0	1
				0	S1	0	0	1	0	1	1
S3	0	1	1	1	S4	1	0	0	1	1	1
				0	S2	0	1	0	0	0	1
S4	1	0	0	1	S5	1	0	1	0	0	1
				0	S3	0	1	1	1	1	1
S5	1	0	1	1	S0	0	0	0	1	0	1
				0	S4	1	0	0	0	0	1
入力として	1	1	0	x	Don't care	x	x	x	x	x	x
存在しない	1	1	1	x	ドントケア	x	x	x	x	x	x

TFF で同期式 N 進カウンターの設計

0 ~ 5 カウンター: 次の状態:

		q[2]	0	0	1	1
		q[1]	0	1	1	0
q[0]	u					
0	0	1		x	1	
0	1			x		
1	1		1	x	1	
1	0			x		

t[2]

		q[2]	0	0	1	1
		q[1]	0	1	1	0
q[0]	u					
0	0		1	x	1	
0	1			x		
1	1	1	1	x		
1	0			x		

t[1]

		q[2]	0	0	1	1
		q[1]	0	1	1	0
q[0]	u					
0	0	1	1	x	1	
0	1	1	1	x	1	
1	1	1	1	x	1	
1	0	1	1	x	1	

t[0]

TFF で同期式 N 進カウンターの設計

0 ~ 5 カウンター: 次の状態:

		q[2] 0 0 1 1			
		q[1] 0 1 1 0			
q[0] u					
0 0	1		x	1	
0 1			x		
1 1		1	x	1	
1 0			x		

t[2]

		q[2] 0 0 1 1			
		q[1] 0 1 1 0			
q[0] u					
0 0		1	x	1	
0 1			x		
1 1	1	1	x		
1 0			x		

t[1]

		q[2] 0 0 1 1			
		q[1] 0 1 1 0			
q[0] u					
0 0	1	1	x	1	
0 1	1	1	x	1	
1 1	1	1	x	1	
1 0	1	1	x	1	

t[0]

$$t[2] = q[1] q[0] u + q[2] \overline{q[0]} \overline{u} + q[2] q[0] u + \overline{q[2]} \overline{q[1]} \overline{q[0]} \overline{u}$$

TFF で同期式 N 進カウンターの設計

0 ~ 5 カウンター: 次の状態:

		q[2] 0 0 1 1			
		q[1] 0 1 1 0			
q[0]	u				
0	0	1		x	1
0	1			x	
1	1		1	x	1
1	0			x	

t[2]

		q[2] 0 0 1 1			
		q[1] 0 1 1 0			
q[0]	u				
0	0		1	x	1
0	1			x	
1	1	1	1	x	
1	0			x	

t[1]

		q[2] 0 0 1 1			
		q[1] 0 1 1 0			
q[0]	u				
0	0	1	1	x	1
0	1	1	1	x	1
1	1	1	1	x	1
1	0	1	1	x	1

t[0]

$$t[2] = q[1] q[0] u + q[2] \overline{q[0]} \overline{u} + q[2] q[0] u + \overline{q[2]} \overline{q[1]} \overline{q[0]} \overline{u}$$

$$t[1] = \overline{q[2]} q[0] u + q[1] \overline{q[0]} \overline{u} + q[2] \overline{q[0]} \overline{u}$$

TFF で同期式 N 進カウンターの設計

0 ~ 5 カウンター: 次の状態:

		q[2] 0 0 1 1			
		q[1] 0 1 1 0			
q[0] u					
00	1		x	1	
01			x		
11		1	x	1	
10			x		

t[2]

		q[2] 0 0 1 1			
		q[1] 0 1 1 0			
q[0] u					
00		1	x	1	
01			x		
11	1	1	x		
10			x		

t[1]

		q[2] 0 0 1 1			
		q[1] 0 1 1 0			
q[0] u					
00	1	1	x	1	
01	1	1	x	1	
11	1	1	x	1	
10	1	1	x	1	

t[0]

$$t[2] = q[1] q[0] u + q[2] \overline{q[0]} \overline{u} + q[2] q[0] u + \overline{q[2]} \overline{q[1]} \overline{q[0]} \overline{u}$$

$$t[1] = \overline{q[2]} q[0] u + q[1] \overline{q[0]} \overline{u} + q[2] \overline{q[0]} \overline{u}$$

$$t[0] = 1$$

TFF で同期式N進カウンターの設計

```
`timescale 1ns/1ns
module counter6_t (clk, clrn, u, a, b, c, d, e, f, g);
    input  clk, clrn, u;
    output a, b, c, d, e, f, g; // outputs

    reg [2:0] q; // current state
    wire [2:0] t; // next state
    // 1. next state ----- 1
    assign t[2] = q[1] & q[0] & u | q[2] & ~q[0] & ~u | q[2] & q[0] & u | ~q[2] & ~q[1] & ~q[0] & ~u;
    assign t[1] = ~q[2] & q[0] & u | q[1] & ~q[0] & ~u | q[2] & ~q[0] & ~u;
    assign t[0] = 1;
    // 2. outputs ----- 2
    assign a = ~q[2] & ~q[1] & q[0] | q[2] & ~q[0];
    assign b = q[2] & q[0];
    assign c = q[1] & ~q[0];
    assign d = a;
    assign e = q[2] | q[0];
    assign f = ~q[2] & q[0] | q[1];
    assign g = ~q[2] & ~q[1];
    // 3. tffs ----- 3
    always @(posedge clk or negedge clrn) begin // tffs
        if (clrn == 0) begin
            q <= 0;
        end else begin
            q <= t & ~q | ~t & q;
        end
    end
endmodule
```

$$t[2] = q[1] q[0] u + q[2] \overline{q[0]} \overline{u} + q[2] q[0] u + \overline{q[2]} \overline{q[1]} \overline{q[0]} \overline{u}$$
$$t[1] = \overline{q[2]} q[0] u + q[1] \overline{q[0]} \overline{u} + q[2] \overline{q[0]} \overline{u}$$
$$t[0] = 1$$

$$a = \overline{q[2]} \overline{q[1]} q[0] + q[2] \overline{q[0]}$$
$$b = q[2] q[0]$$
$$c = q[1] \overline{q[0]}$$
$$d = a$$
$$e = q[2] + q[0]$$
$$f = \overline{q[2]} q[0] + q[1]$$
$$g = \overline{q[2]} \overline{q[1]}$$

[counter6_t.v](#)

TFF で同期式N進カウンターの設計

```
`timescale 1ns/1ns
module counter6_t_tb;
    reg  clk, clrn, u;
    wire a, b, c, d, e, f, g;

    counter6_t i0 (clk, clrn, u, a, b, c, d, e, f, g);

    initial begin
        #0  clk = 1; clrn = 0; u = 1;
        #1  clrn = 1;
        #18 u = 0;
        #25 $finish;
    end

    always #1 clk = ~clk;

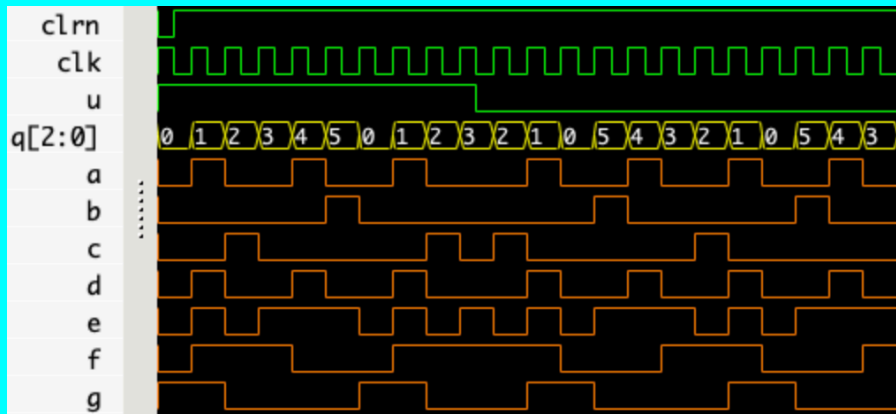
    initial begin
        $dumpfile ("counter6_t.vcd");
        $dumpvars;
    end

endmodule
```

[counter6_t_tb.v](#)

TFF で同期式 N 進カウンターの設計

0 ~ 5 カウンター: シミュレーション波形:



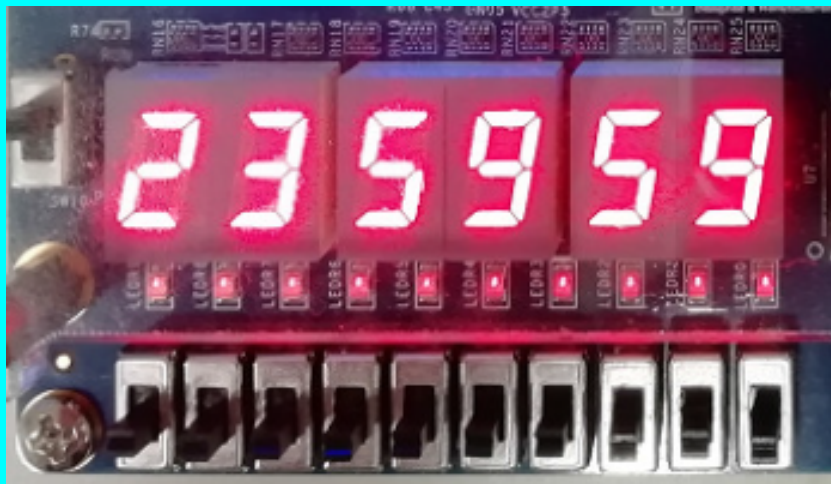
[counter6_t_tb.v](#)

Count Up

Count Down

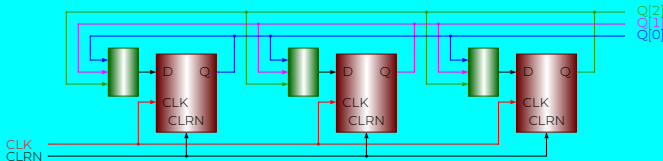
デジタル時計の実装

FPGA で実装したデジタル時計



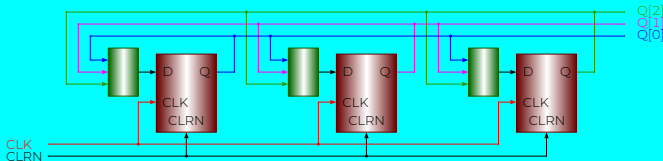
同期式と非同期式カウンタ

- 同期式 (Synchronous) とは、構成するフリップフロップの動作が同一のクロックで行われる方式のことである。

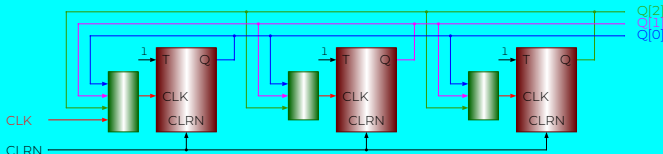


同期式と非同期式カウンタ

- 同期式 (Synchronous) とは、構成するフリップフロップの動作が同一のクロックで行われる方式のことである。

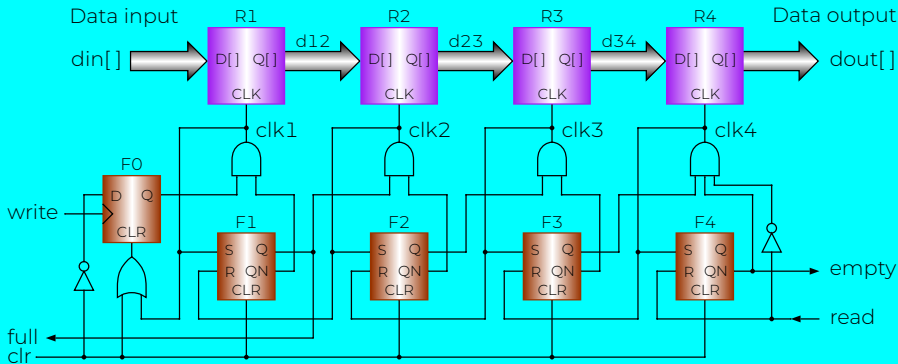


- 非同期式 (Asynchronous) とは、カウンタへの入力と各フリップフロップの出力の変化により、自身のフリップフロップの入力が決定され、その変化で動作する方式のことである。



非同期式FIFOの設計

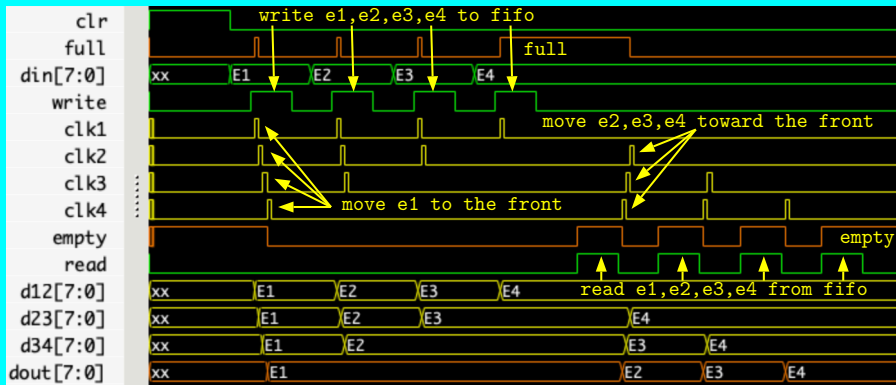
FIFO: First-In-First-Out (Queue — キュー)



R1、R2、R3、R4 : FIFO レジスタ (深さ : 4)

clk1、clk2、clk3、clk4 : 非同期 clock

非同期式FIFOの設計



write: write data to FIFO;
read: read data from FIFO.

同期式N進カウンターの設計

まとめ

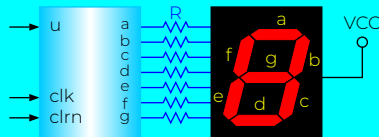
- 同期式N進カウンター
 - ▶ DFF で同期式N進カウンター
 - ▶ JKFF で同期式N進カウンター
 - ▶ TFF で同期式N進カウンター
- 非同期式カウンター
- 非同期式FIFO

課題 XIII (100 点 + 100 点)

7セグメント LED を使い同期式 10 進カウントアップ / カウントダウンする回路を、**DFF** を用いて設計し、動作検証シミュレーションして下さい (Don't Care 項を利用)。

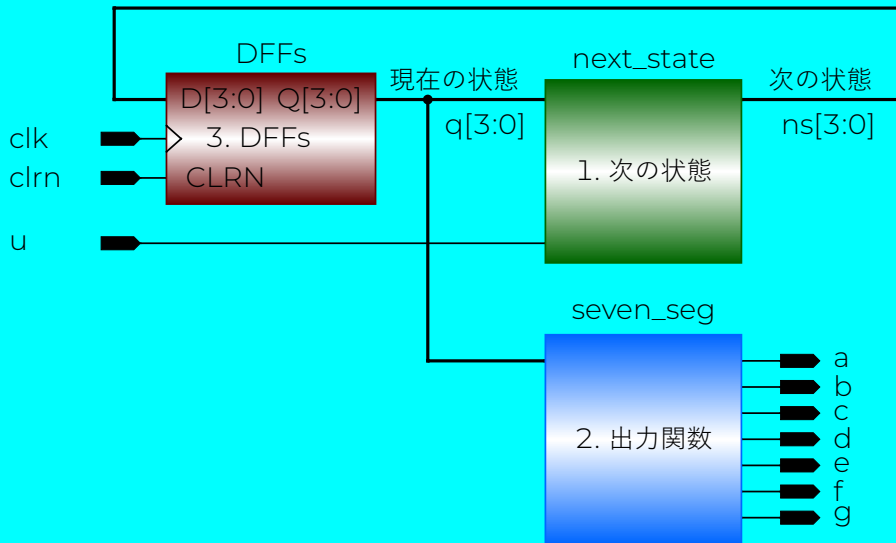
(P10 - P21 を参照)

- カウント範囲: 0 ~ 9
- $u = 1$ (カウントアップ):
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, ...
- $u = 0$ (カウントダウン):
0, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, 9, 8, 7, ...
- 7セグメント LED で表示する (0: 点灯; 1: 消灯)



注意：今回の課題については、EDA ツール (iverilog や ModelSim など) を使用しないでください。

課題 XIII (100 点 + 100 点)



課題 XIII (100 点 + 100 点)

レポートの必須項目:

- (1) 状態遷移図
- (2) 次の状態 `next_state` の真理値表
- (3) 次の状態 `next_state` 各出力信号のカルノー図
- (4) 次の状態 `next_state` の論理式
$$\begin{aligned} ns[3] &= \\ ns[2] &= \\ ns[1] &= \\ ns[0] &= \end{aligned}$$
- (5) 次の状態 `next_state` の回路図

課題 XIII (100 点 + 100 点)

(6) 出力関数 `seven_seg` の真理値表

(7) 出力関数 `seven_seg` 各出力信号のカルノー図

(8) 出力関数 `seven_seg` の論理式

a =

b =

c =

d =

e =

f =

g =

(9) 出力関数 `seven_seg` の回路図

課題 XIII (100 点 + 100 点)

(10) 予測波形

clrn

clk

u

q

a

b

c

d

e

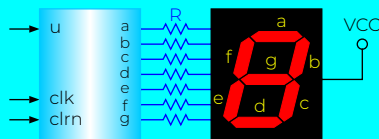
f

g

オプション (+50 点)

7セグメント LED を使い同期式 10 進カウントアップ / カウントダウンする回路を、JKFF を用いて設計し、動作検証シミュレーションして下さい (Don't Care 項を利用)。

(P22 - P27 を参照)

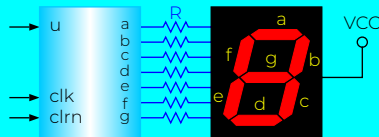


- カウント範囲: 0 ~ 9
- $u = 1$ (カウントアップ):
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, ...
- $u = 0$ (カウントダウン):
0, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, 9, 8, 7, ...
- 7セグメント LED で表示する (0: 点灯; 1: 消灯)

オプション (+50 点)

7セグメント LED を使い同期式 10 進カウントアップ / カウントダウンする回路を、TFF を用いて設計し、動作検証シミュレーションして下さい (Don't Care 項を利用)。

(P28 - P33 を参照)



- カウント範囲: 0 ~ 9
- $u = 1$ (カウントアップ):
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, ...
- $u = 0$ (カウントダウン):
0, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, 9, 8, 7, ...
- 7セグメント LED で表示する (0: 点灯; 1: 消灯)