

論理回路入門 (12)

有限状態機械と順序回路設計

李 亞民

2022 年 12 月 13 日 (火)

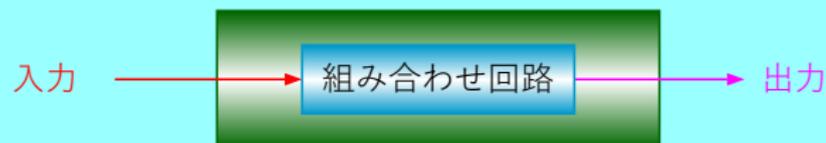
順序回路

ポイント

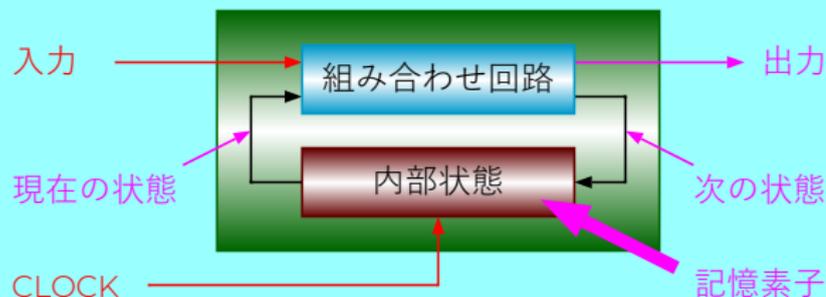
- 順序回路の基本構造
- 順序回路の分析
- Mealy 型と Moore 型順序回路
- 順序回路設計の手順
- 状態遷移図
- 次の状態の真理値表、論理式、および回路
- 出力関数の真理値表、論理式、および回路
- 交通信号機制御システム

論理回路の種類

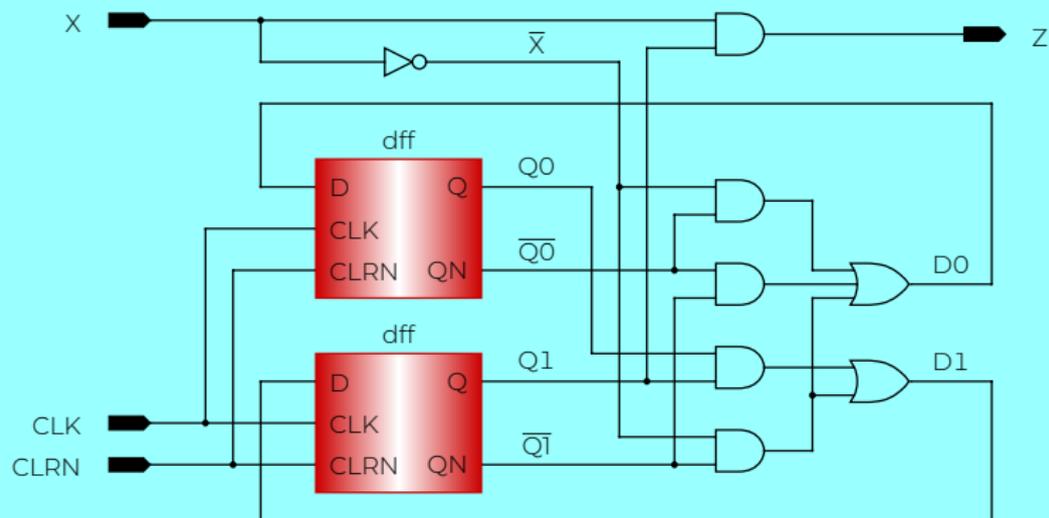
- ① 組み合わせ回路 (Combinational Circuit): 現在の入力のみで出力が決まる回路である。



- ② 順序回路 (Sequential Circuit): 内部状態と入力信号で出力が決まる回路である。有限状態機械 (Finite state machine — FSM) とも呼ばれる。



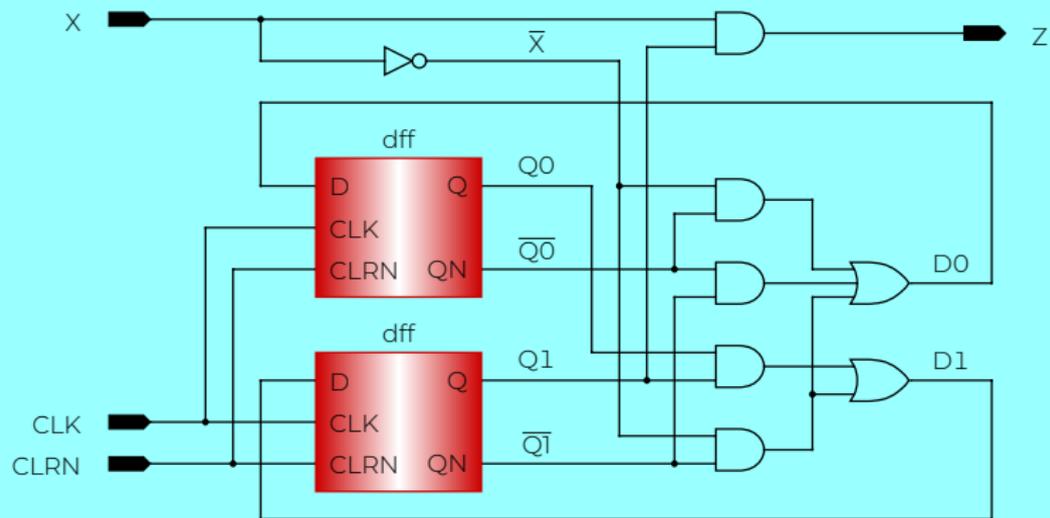
順序回路の分析



もし、こうゆうふうな回路があれば、

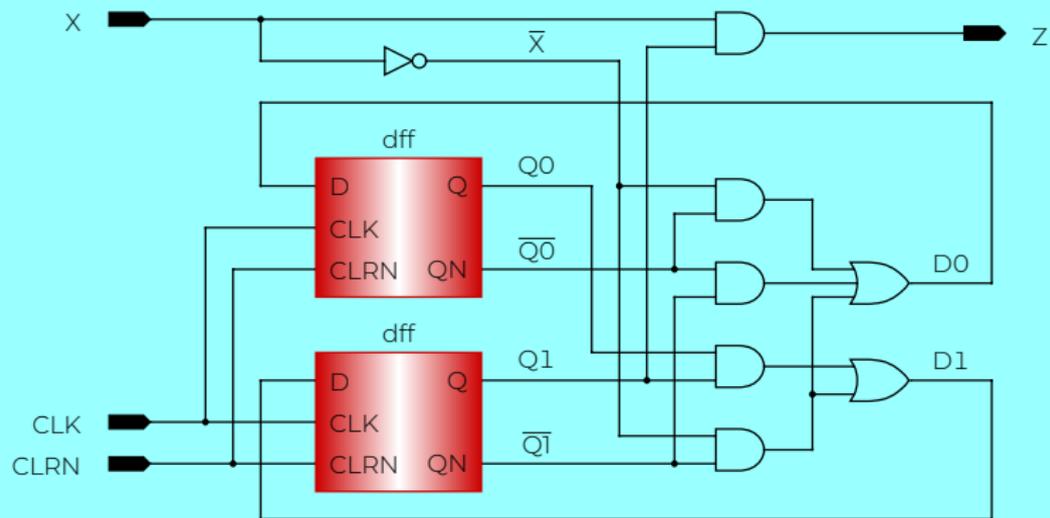
D1, D0, Z の論理式？ 真理値表？ 状態遷移図？

順序回路の分析 — 論理式



$D1 =$

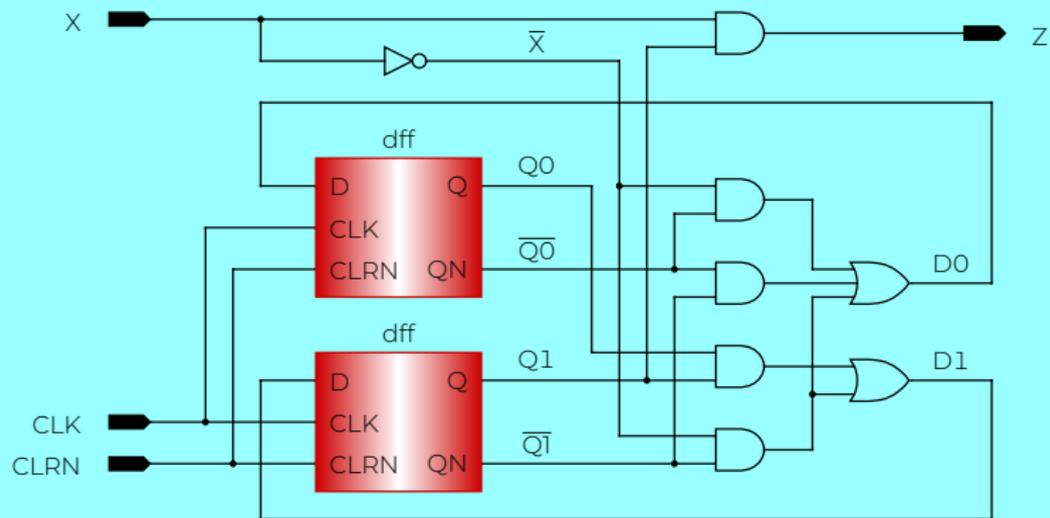
順序回路の分析 — 論理式



$$D1 = Q1 \cdot Q0 + \bar{Q1} \cdot \bar{X}$$

$$D0 =$$

順序回路の分析 — 論理式

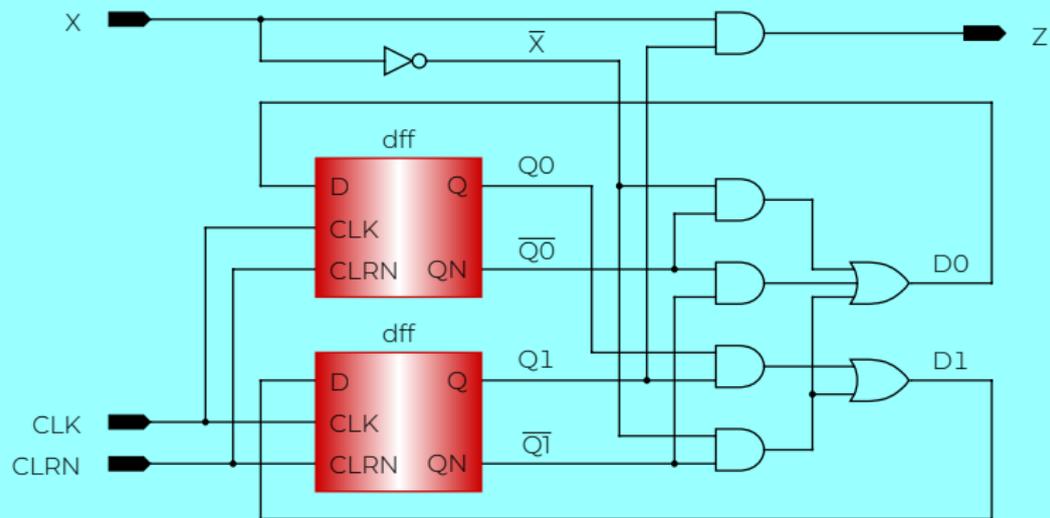


$$D1 = Q1 \cdot Q0 + \overline{Q1} \cdot \overline{X}$$

$$D0 = \overline{Q0} \cdot \overline{X} + \overline{Q1} \cdot \overline{Q0} + \overline{Q1} \cdot \overline{X}$$

$$Z =$$

順序回路の分析 — 論理式

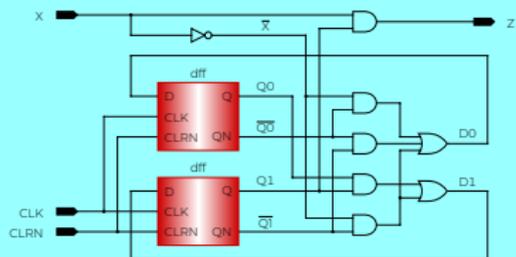


$$D1 = Q1 \cdot Q0 + \overline{Q1} \cdot \overline{X}$$

$$D0 = \overline{Q0} \cdot \overline{X} + \overline{Q1} \cdot \overline{Q0} + \overline{Q1} \cdot X$$

$$Z = Q1 \cdot X$$

順序回路の分析 — 真理値表



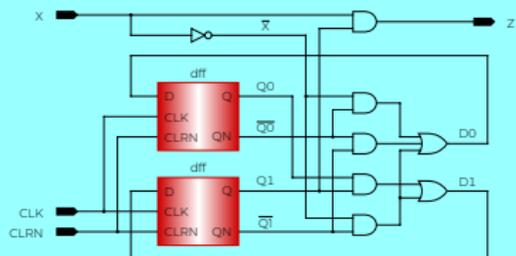
$$D1 = Q1 \cdot Q0 + \overline{Q1} \cdot \overline{X}$$

$$D0 = \overline{Q0} \cdot \overline{X} + \overline{Q1} \cdot \overline{Q0} + \overline{Q1} \cdot \overline{X}$$

$$Z = Q1 \cdot X$$

| 現在の状態 | | | 入力 | 次の状態 | | 出力 |
|-------|----|---|----|------|----|----|
| Q1 | Q0 | | X | D1 | D0 | Z |
| S0 | 0 | 0 | 0 | | | |

順序回路の分析 — 真理値表



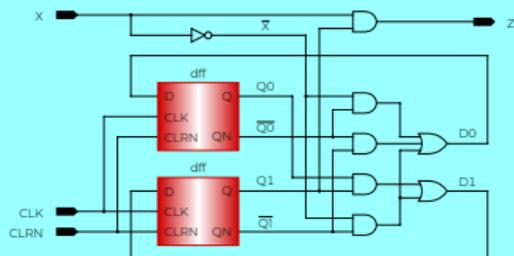
$$D1 = Q1 \cdot Q0 + \overline{Q1} \cdot \overline{X}$$

$$D0 = \overline{Q0} \cdot \overline{X} + \overline{Q1} \cdot \overline{Q0} + \overline{Q1} \cdot \overline{X}$$

$$Z = Q1 \cdot X$$

| 現在の状態 | | | 入力 X | 次の状態 | | 出力 Z |
|-------|----|----|---------|------|---|---------|
| Q1 | Q0 | D1 | | D0 | | |
| S0 | 0 | 0 | 0 | S3 | 1 | 1 |
| | | | 1 | | | |

順序回路の分析 — 真理値表



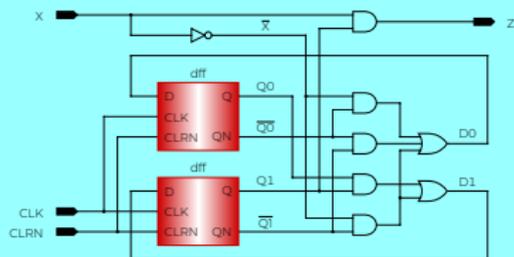
$$D1 = Q1 \cdot Q0 + \overline{Q1} \cdot \overline{X}$$

$$D0 = \overline{Q0} \cdot \overline{X} + \overline{Q1} \cdot \overline{Q0} + \overline{Q1} \cdot \overline{X}$$

$$Z = Q1 \cdot X$$

| 現在の状態 | | 入力 X | 次の状態 | | 出力 Z | |
|-------|----|---------|------|----|---------|---|
| Q1 | Q0 | | D1 | D0 | | |
| S0 | 0 | 0 | S3 | 1 | 1 | 0 |
| | | 1 | S1 | 0 | 1 | 0 |
| S1 | 0 | 1 | | | | |

順序回路の分析 — 真理値表



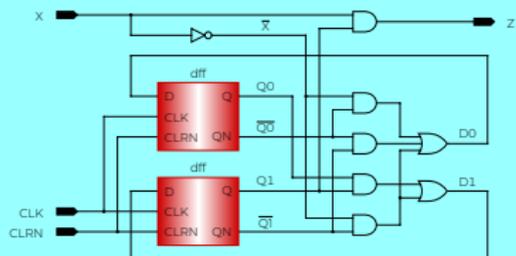
$$D1 = Q1 \cdot Q0 + \overline{Q1} \cdot \overline{X}$$

$$D0 = \overline{Q0} \cdot \overline{X} + \overline{Q1} \cdot \overline{Q0} + \overline{Q1} \cdot \overline{X}$$

$$Z = Q1 \cdot X$$

| 現在の状態 | | 入力 X | 次の状態 | | 出力 Z | |
|-------|----|---------|------|----|---------|---|
| Q1 | Q0 | | D1 | D0 | | |
| S0 | 0 | 0 | S3 | 1 | 1 | 0 |
| | | 1 | S1 | 0 | 1 | 0 |
| S1 | 0 | 0 | S3 | 1 | 1 | 0 |
| | | 1 | | | | |

順序回路の分析 — 真理値表



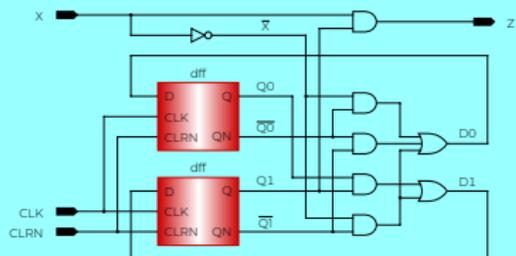
$$D1 = Q1 \cdot Q0 + \overline{Q1} \cdot \overline{X}$$

$$D0 = \overline{Q0} \cdot \overline{X} + \overline{Q1} \cdot \overline{Q0} + \overline{Q1} \cdot \overline{X}$$

$$Z = Q1 \cdot X$$

| 現在の状態 | 入力 | | 次の状態 | | 出力 |
|-------|----|----|------|--------|----|
| | Q1 | Q0 | X | D1 D0 | |
| S0 | 0 | 0 | 0 | S3 1 1 | 0 |
| | | | 1 | S1 0 1 | 0 |
| S1 | 0 | 1 | 0 | S3 1 1 | 0 |
| | | | 1 | S0 0 0 | 0 |
| S2 | 1 | 0 | 0 | | |

順序回路の分析 — 真理値表



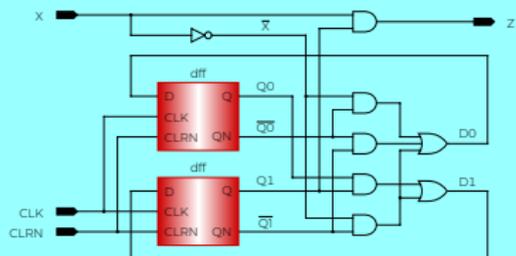
$$D1 = Q1 \cdot Q0 + \bar{Q1} \cdot \bar{X}$$

$$D0 = \bar{Q0} \cdot \bar{X} + \bar{Q1} \cdot \bar{Q0} + \bar{Q1} \cdot \bar{X}$$

$$Z = Q1 \cdot X$$

| 現在の状態 | | 入力 X | 次の状態 | | 出力 Z | |
|-------|----|---------|------|----|---------|---|
| Q1 | Q0 | | D1 | D0 | | |
| S0 | 0 | 0 | S3 | 1 | 1 | 0 |
| | | | S1 | 0 | 1 | 0 |
| S1 | 0 | 1 | S3 | 1 | 1 | 0 |
| | | | S0 | 0 | 0 | 0 |
| S2 | 1 | 0 | S1 | 0 | 1 | 0 |
| | | | | | | |
| | | 1 | | | | |

順序回路の分析 — 真理値表



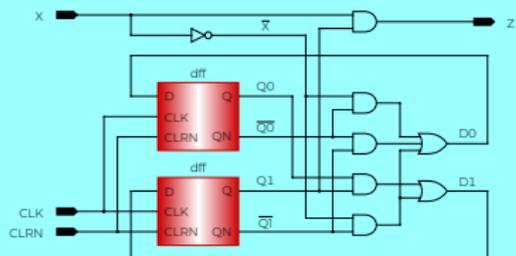
$$D1 = Q1 \cdot Q0 + \overline{Q1} \cdot \overline{X}$$

$$D0 = \overline{Q0} \cdot \overline{X} + \overline{Q1} \cdot \overline{Q0} + \overline{Q1} \cdot \overline{X}$$

$$Z = Q1 \cdot X$$

| 現在の状態 | | 入力 X | 次の状態 | | 出力 Z | |
|-------|----|---------|------|----|---------|---|
| Q1 | Q0 | | D1 | D0 | | |
| S0 | 0 | 0 | S3 | 1 | 1 | 0 |
| | | | S1 | 0 | 1 | 0 |
| S1 | 0 | 1 | S3 | 1 | 1 | 0 |
| | | | S0 | 0 | 0 | 0 |
| S2 | 1 | 0 | S1 | 0 | 1 | 0 |
| | | | S0 | 0 | 0 | 1 |
| S3 | 1 | 1 | 0 | | | |

順序回路の分析 — 真理値表



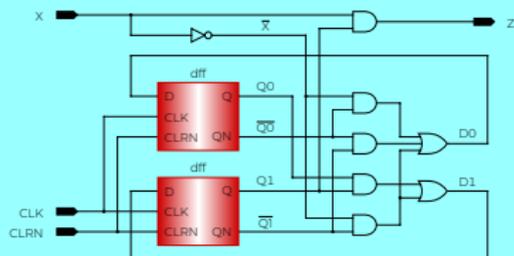
$$D1 = Q1 \cdot Q0 + \overline{Q1} \cdot \overline{X}$$

$$D0 = \overline{Q0} \cdot \overline{X} + \overline{Q1} \cdot \overline{Q0} + \overline{Q1} \cdot \overline{X}$$

$$Z = Q1 \cdot X$$

| 現在の状態 | | 入力 X | 次の状態 | | 出力 | |
|-------|----|---------|------|----|----|---|
| Q1 | Q0 | | D1 | D0 | Z | |
| S0 | 0 | 0 | S3 | 1 | 1 | 0 |
| | | | S1 | 0 | 1 | 0 |
| S1 | 0 | 1 | S3 | 1 | 1 | 0 |
| | | | S0 | 0 | 0 | 0 |
| S2 | 1 | 0 | S1 | 0 | 1 | 0 |
| | | | S0 | 0 | 0 | 1 |
| S3 | 1 | 1 | S2 | 1 | 0 | 0 |
| | | | | | | |

順序回路の分析 — 真理値表



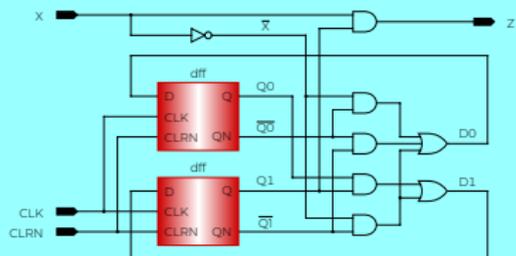
$$D1 = Q1 \cdot Q0 + \overline{Q1} \cdot \overline{X}$$

$$D0 = \overline{Q0} \cdot \overline{X} + \overline{Q1} \cdot \overline{Q0} + \overline{Q1} \cdot \overline{X}$$

$$Z = Q1 \cdot X$$

| 現在の状態 | | 入力 X | 次の状態 | | 出力 Z | |
|-------|----|---------|------|----|---------|---|
| Q1 | Q0 | | D1 | D0 | | |
| S0 | 0 | 0 | S3 | 1 | 1 | 0 |
| | | | S1 | 0 | 1 | 0 |
| S1 | 0 | 1 | S3 | 1 | 1 | 0 |
| | | | S0 | 0 | 0 | 0 |
| S2 | 1 | 0 | S1 | 0 | 1 | 0 |
| | | | S0 | 0 | 0 | 1 |
| S3 | 1 | 1 | S2 | 1 | 0 | 0 |
| | | | S2 | 1 | 0 | 1 |

順序回路の分析 — 状態遷移図



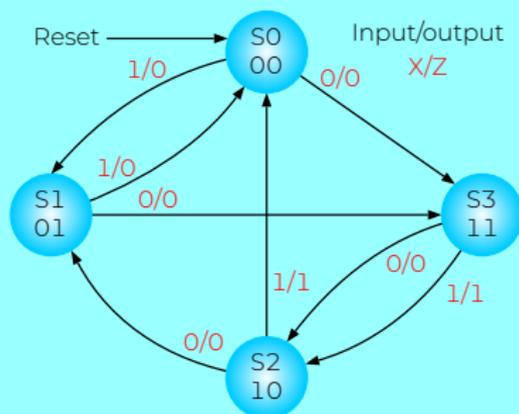
$$D1 = Q1 \cdot Q0 + \overline{Q1} \cdot \overline{X}$$

$$D0 = \overline{Q0} \cdot \overline{X} + Q1 \cdot \overline{Q0} + Q1 \cdot \overline{X}$$

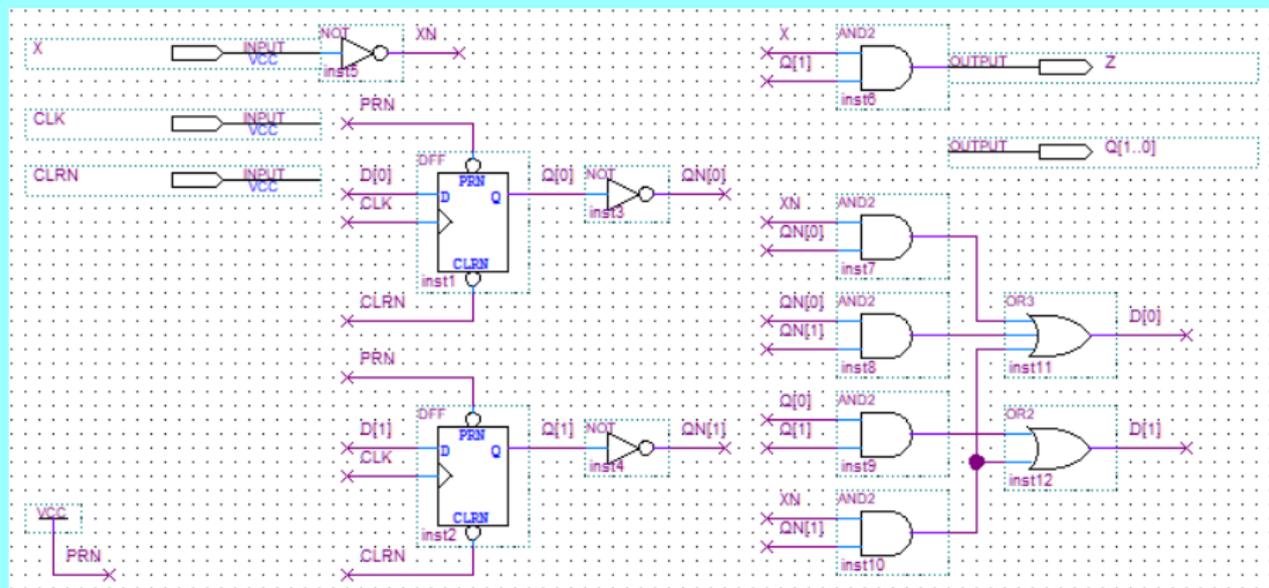
$$Z = Q1 \cdot X$$

| 現在の状態 | | 入力 X | 次の状態 | | 出力 Z | |
|-------|----|---------|------|----|---------|---|
| Q1 | Q0 | | D1 | D0 | | |
| S0 | 0 | 0 | S3 | 1 | 1 | 0 |
| | | | S1 | 0 | 1 | 0 |
| S1 | 0 | 1 | S3 | 1 | 1 | 0 |
| | | | S0 | 0 | 0 | 0 |
| S2 | 1 | 0 | S1 | 0 | 1 | 0 |
| | | | S0 | 0 | 0 | 1 |
| S3 | 1 | 1 | S2 | 1 | 0 | 0 |
| | | | S2 | 1 | 0 | 1 |

真理値表 ⇒ 状態遷移図 (分析の手順):



順序回路の実装 — 回路



$$D1 = Q1 \cdot Q0 + \overline{Q1} \cdot \overline{X}$$

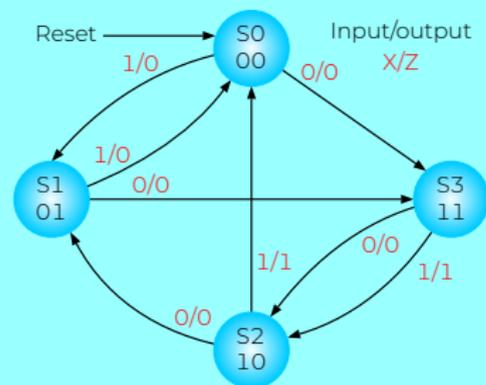
$$D0 = \overline{Q0} \cdot \overline{X} + \overline{Q1} \cdot \overline{Q0} + \overline{Q1} \cdot X$$

$$Z = Q1 \cdot X$$

順序回路の検証 — テストベンチ

```
'timescale 1ns/1ps // unit = 1 ns; accuracy = 1 ps
module seq_ex_tb;
    reg          X, CLK, CLRN;
    wire         Z;
    wire [1:0] Q;
    seq_ex i (.X(X), .CLK(CLK), .CLRN(CLRN), .Z(Z), .Q(Q));
    initial begin
        #0 CLRN = 0; CLK = 1; X = 0;
        #1 CLRN = 1;
        #6 X = 1;
        #2 X = 0;
        #2 X = 1;
        #6 X = 0;
        #3 $stop;
    end
    always #1 CLK = ~CLK;
endmodule
```

順序回路の検証 — 波形



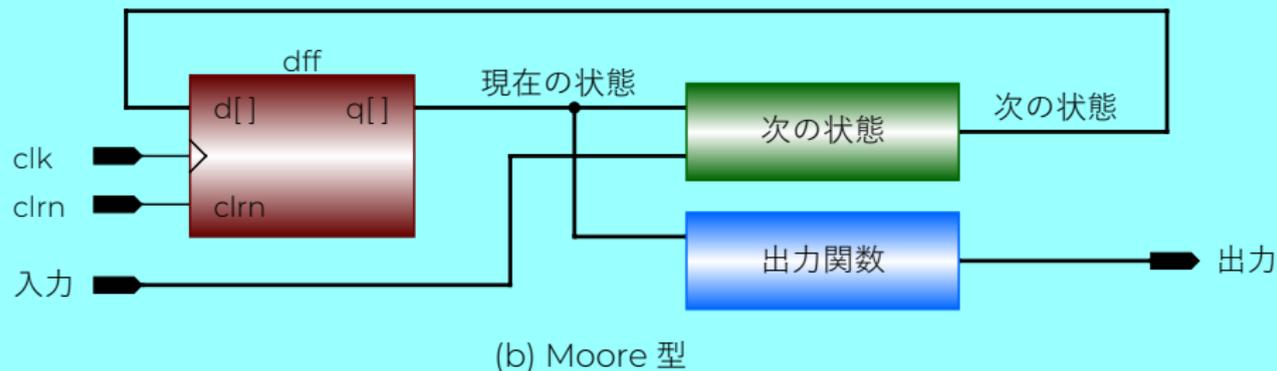
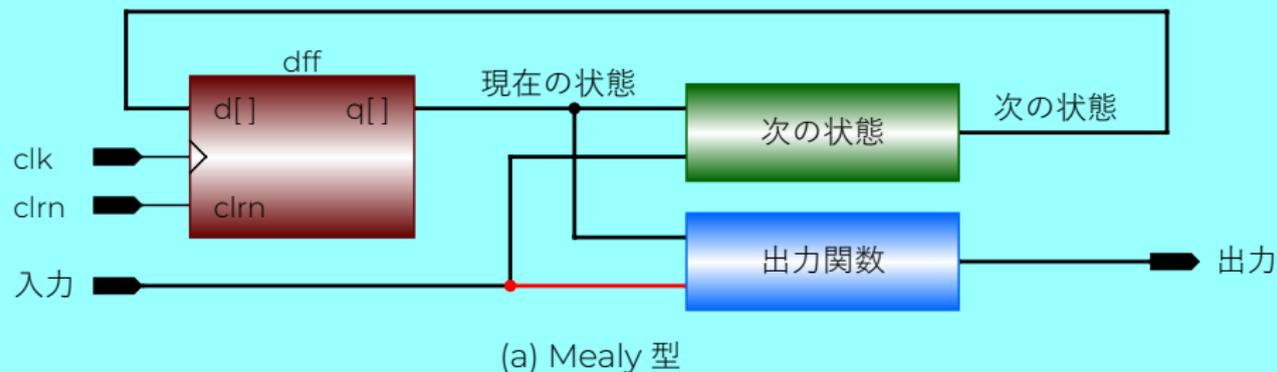
$$D1 = Q1 \cdot Q0 + \overline{Q1} \cdot \overline{X}$$

$$D0 = \overline{Q0} \cdot \overline{X} + \overline{Q1} \cdot \overline{Q0} + \overline{Q1} \cdot X$$

$$Z = Q1 \cdot X$$

テストベンチを作成するときは、
状態遷移図のすべての状態と遷移
を確認しなければならない。

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。

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。

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)。

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。

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。

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。

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)。

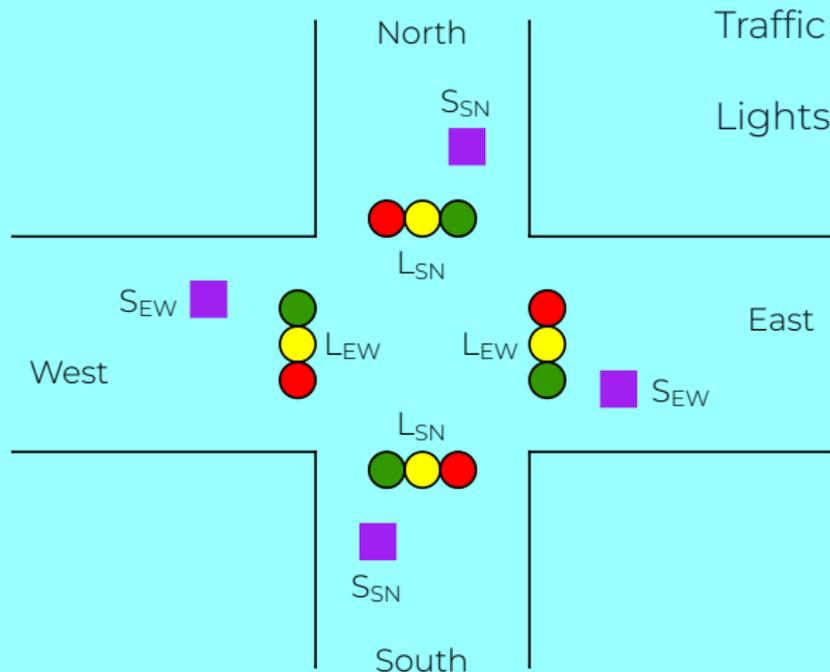
順序回路設計の手順

- 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)。

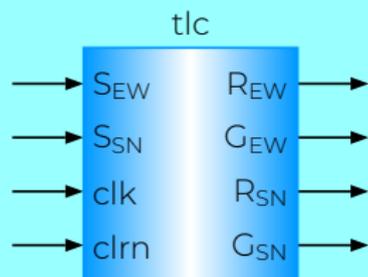
順序回路設計の手順

- 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)。

交通信号機制御システム (問題)

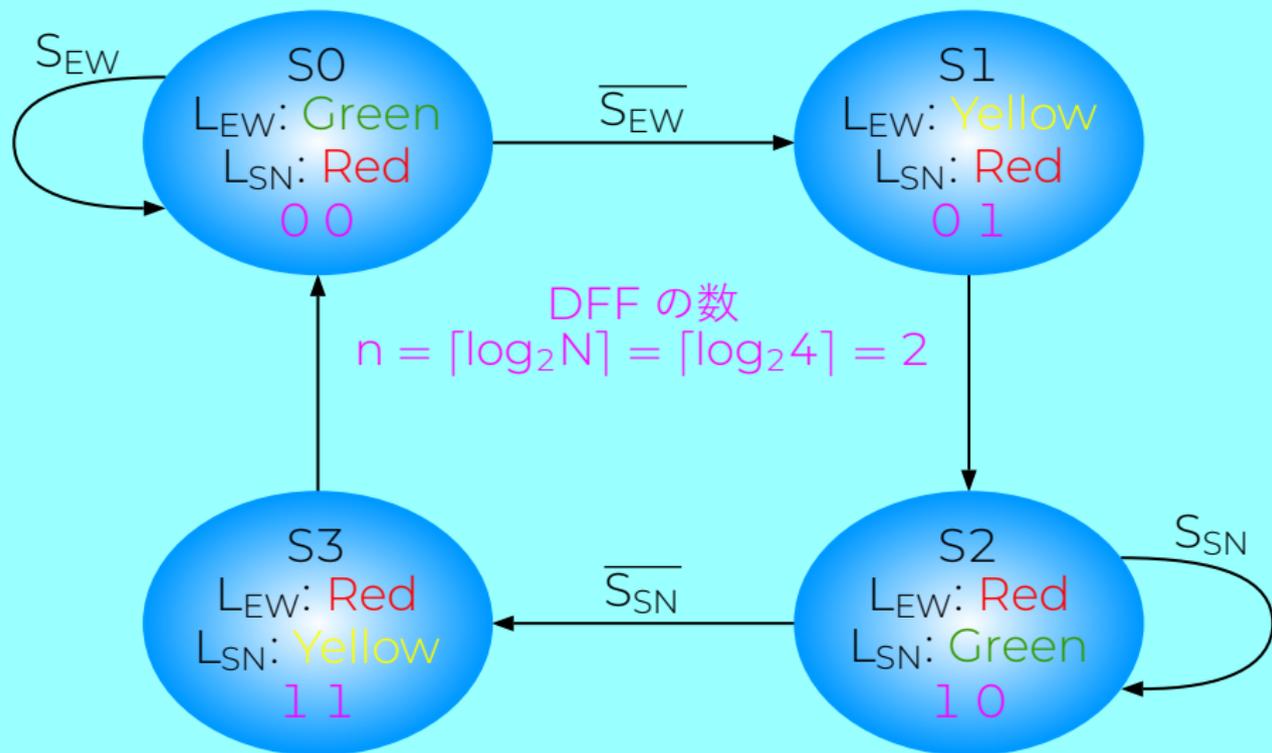


| R_{XX} | G_{XX} | Color |
|----------|----------|--------|
| 0 | 1 | Green |
| 1 | 0 | Red |
| 1 | 1 | Yellow |



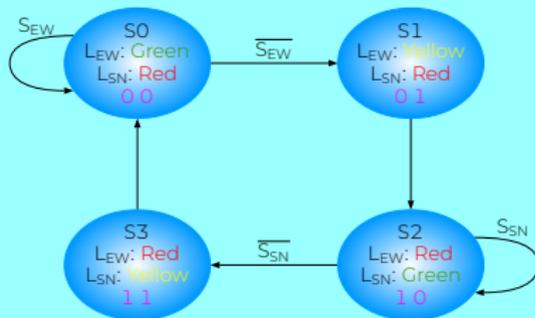
EW: East and West (東西)
SN: South and North (南北)

交通信号機制御 (状態遷移図)



交通信号機制御 (次の状態の真理値表)

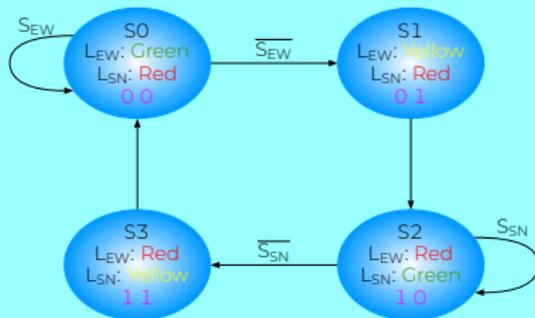
次の状態の真理値表
(状態遷移表)



| 現在の状態 | | | 入力 | | 次の状態 | |
|-------|----|---|----------|----------|------|----|
| Q1 | Q0 | | S_{EW} | S_{SN} | D1 | D0 |
| S0 | 0 | 0 | 0 | X | | |

交通信号機制御 (次の状態の真理値表)

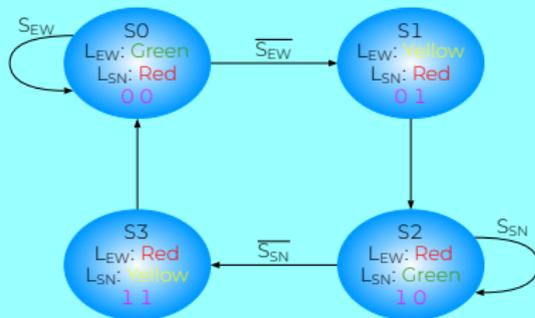
次の状態の真理値表
(状態遷移表)



| 現在の状態 | | | 入力 | | 次の状態 | | |
|-------|----|----|----------|----------|------|----|----|
| | Q1 | Q0 | S_{EW} | S_{SN} | | D1 | D0 |
| S0 | 0 | 0 | 0 | X | S1 | 0 | 1 |
| | | | 1 | X | | | |

交通信号機制御 (次の状態の真理値表)

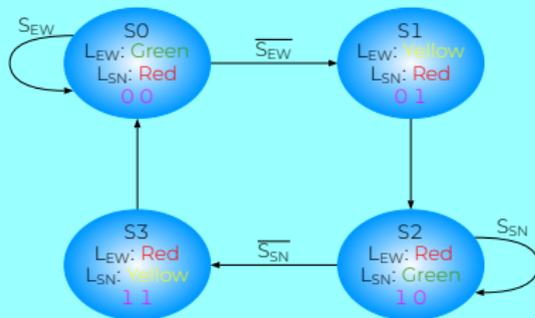
次の状態の真理値表
(状態遷移表)



| 現在の状態 | 入力 | | 次の状態 | | | | |
|-------|----|----|----------|----------|----|----|---|
| | Q1 | Q0 | S_{EW} | S_{SN} | D1 | D0 | |
| S0 | 0 | 0 | 0 | X | S1 | 0 | 1 |
| | | | 1 | X | S0 | 0 | 0 |
| S1 | 0 | 1 | X | X | | | |

交通信号機制御 (次の状態の真理値表)

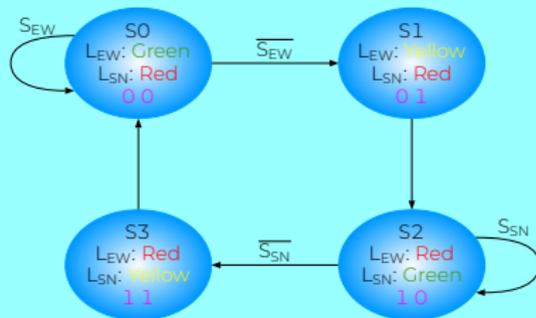
次の状態の真理値表
(状態遷移表)



| 現在の状態 | 入力 | | 次の状態 | | | | |
|-------|----|----|----------|----------|----|----|---|
| | Q1 | Q0 | S_{EW} | S_{SN} | D1 | D0 | |
| S0 | 0 | 0 | 0 | X | S1 | 0 | 1 |
| | | | 1 | X | S0 | 0 | 0 |
| S1 | 0 | 1 | X | X | S2 | 1 | 0 |
| S2 | 1 | 0 | X | 0 | | | |

交通信号機制御 (次の状態の真理値表)

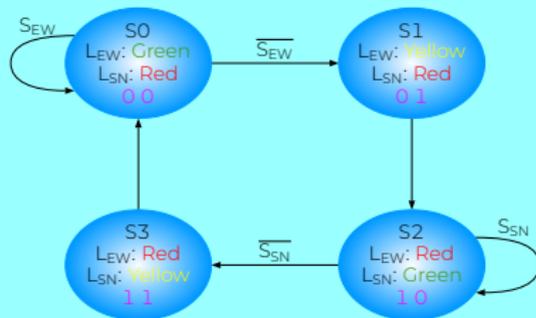
次の状態の真理値表
(状態遷移表)



| 現在の状態 | 入力 | | 次の状態 | | | | |
|-------|----|----|----------|----------|----|----|---|
| | Q1 | Q0 | S_{EW} | S_{SN} | D1 | D0 | |
| S0 | 0 | 0 | 0 | X | S1 | 0 | 1 |
| | | | 1 | X | S0 | 0 | 0 |
| S1 | 0 | 1 | X | X | S2 | 1 | 0 |
| S2 | 1 | 0 | X | 0 | S3 | 1 | 1 |
| | | | X | 1 | | | |

交通信号機制御（次の状態の真理値表）

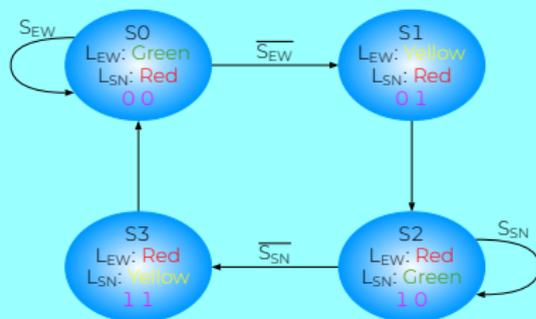
次の状態の真理値表
(状態遷移表)



| 現在の状態 | 入力 | | 次の状態 | | | | |
|-------|----|----|-----------------|-----------------|----|----|---|
| | Q1 | Q0 | S _{EW} | S _{SN} | D1 | D0 | |
| S0 | 0 | 0 | 0 | X | S1 | 0 | 1 |
| | | | 1 | X | S0 | 0 | 0 |
| S1 | 0 | 1 | X | X | S2 | 1 | 0 |
| S2 | 1 | 0 | X | 0 | S3 | 1 | 1 |
| | | | X | 1 | S2 | 1 | 0 |
| S3 | 1 | 1 | X | X | | | |

交通信号機制御 (次の状態の真理値表)

次の状態の真理値表
(状態遷移表)



| | 現在の状態 | | 入力 | | 次の状態 | | |
|----|-------|----|----------|----------|------|----|---|
| | Q1 | Q0 | S_{EW} | S_{SN} | D1 | D0 | |
| S0 | 0 | 0 | 0 | X | S1 | 0 | 1 |
| | | | 1 | X | S0 | 0 | 0 |
| S1 | 0 | 1 | X | X | S2 | 1 | 0 |
| S2 | 1 | 0 | X | 0 | S3 | 1 | 1 |
| | | | X | 1 | S2 | 1 | 0 |
| S3 | 1 | 1 | X | X | S0 | 0 | 0 |

交通信号機制御 (次の状態の論理式)

| 現在の状態 | | | 入力 | | 次の状態 | | |
|-------|----|----|-----------------|-----------------|------|----|----|
| | Q1 | Q0 | S _{EW} | S _{SN} | | D1 | D0 |
| S0 | 0 | 0 | 0 | X | S1 | 0 | 1 |
| | | | 1 | X | S0 | 0 | 0 |
| S1 | 0 | 1 | X | X | S2 | 1 | 0 |
| S2 | 1 | 0 | X | 0 | S3 | 1 | 1 |
| | | | X | 1 | S2 | 1 | 0 |
| S3 | 1 | 1 | X | X | S0 | 0 | 0 |

次の状態の論理式

$$D1 = \overline{Q1} \cdot Q0 + Q1 \cdot \overline{Q0}$$

$$D0 = \overline{Q1} \cdot \overline{Q0} \cdot \overline{S_{EW}} + Q1 \cdot \overline{Q0} \cdot \overline{S_{SN}}$$

D1

| Q1 | Q0 | 0 | 0 | 1 | 1 |
|---------------------------------|----|---|---|---|---|
| Q0 | | 0 | 1 | 1 | 0 |
| S _{EW} S _{SN} | 00 | | 1 | | 1 |
| | 01 | | 1 | | 1 |
| | 11 | | 1 | | 1 |
| | 10 | | 1 | | 1 |

D0

| Q1 | Q0 | 0 | 0 | 1 | 1 |
|---------------------------------|----|---|---|---|---|
| Q0 | | 0 | 1 | 1 | 0 |
| S _{EW} S _{SN} | 00 | 1 | | | 1 |
| | 01 | 1 | | | |
| | 11 | | | | |
| | 10 | | | | 1 |

交通信号機制御（出力関数の真理値表）

出力関数の真理値表

Color encoding: R G Color
0 1: Green
1 0: Red
1 1: Yellow

S0: L_{EW} : Green; L_{SN} : Red
S1: L_{EW} : Yellow; L_{SN} : Red
S2: L_{EW} : Red; L_{SN} : Green
S3: L_{EW} : Red; L_{SN} : Yellow

| 現在の状態 | | | 出力 | | | |
|-------|----|----|----------|----------|----------|----------|
| | Q1 | Q0 | R_{EW} | G_{EW} | R_{SN} | G_{SN} |
| S0 | 0 | 0 | 0 | 1 | 1 | 0 |
| S1 | 0 | 1 | 1 | 1 | 1 | 0 |
| S2 | 1 | 0 | 1 | 0 | 0 | 1 |
| S3 | 1 | 1 | 1 | 0 | 1 | 1 |

交通信号機制御（出力関数の論理式）

| | 現在の状態 | | 出力 | | | |
|----|-------|----|-----------------|-----------------|-----------------|-----------------|
| | Q1 | Q0 | R _{EW} | G _{EW} | R _{SN} | G _{SN} |
| S0 | 0 | 0 | 0 | 1 | 1 | 0 |
| S1 | 0 | 1 | 1 | 1 | 1 | 0 |
| S2 | 1 | 0 | 1 | 0 | 0 | 1 |
| S3 | 1 | 1 | 1 | 0 | 1 | 1 |

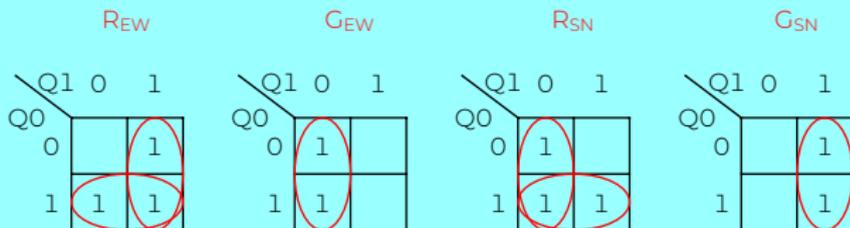
出力関数の論理式

$$R_{EW} = Q1 + Q0$$

$$G_{EW} = \overline{Q1}$$

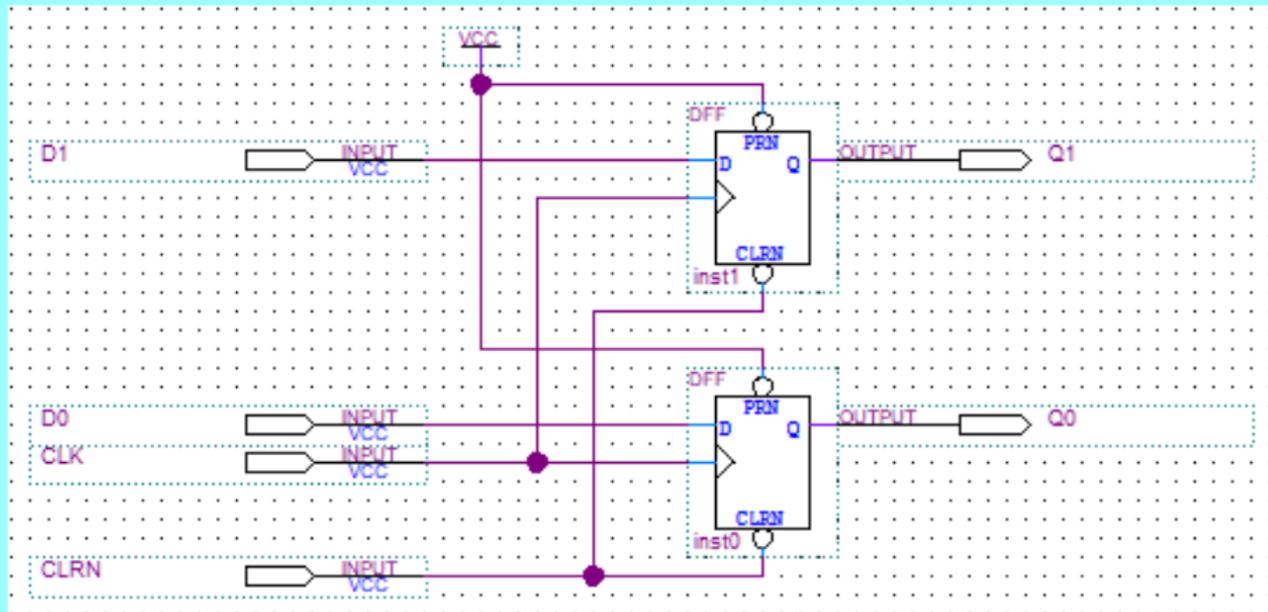
$$R_{SN} = \overline{Q1} + Q0$$

$$G_{SN} = Q1$$



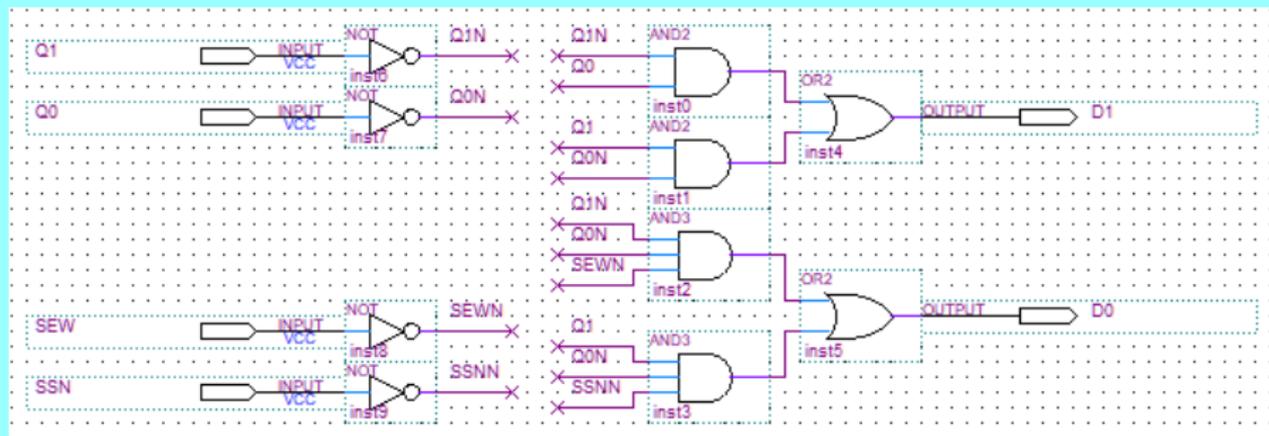
交通信号機制御（回路）

1. DFF の回路 tlc_dff2



交通信号機制御（回路）

2. 次の状態の回路 tlc_next_state

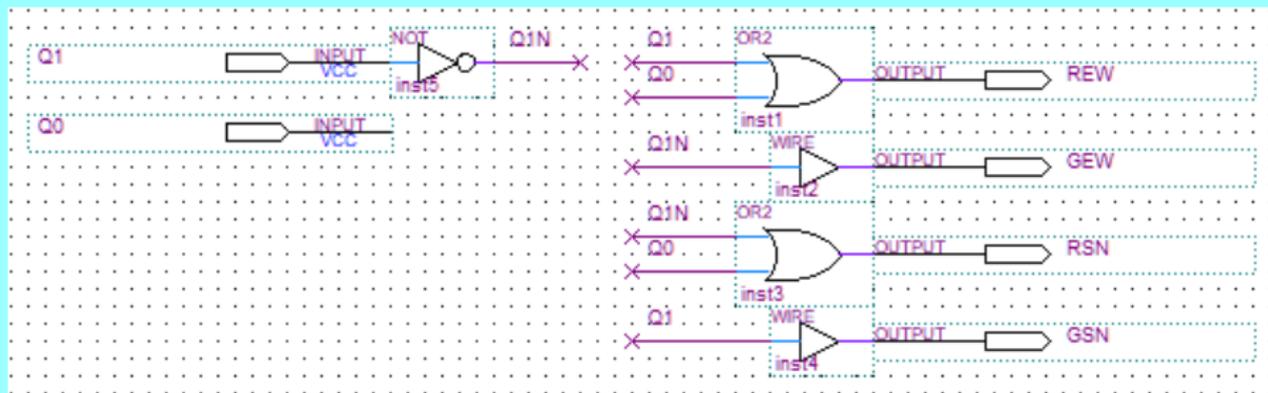


$$D1 = \overline{Q1} \cdot Q0 + Q1 \cdot \overline{Q0}$$

$$D0 = \overline{Q1} \cdot \overline{Q0} \cdot \overline{SEW} + Q1 \cdot \overline{Q0} \cdot \overline{SSN}$$

交通信号機制御（回路）

3. 出力関数の回路 tlc_output



$$R_{EW} = Q1 + Q0$$

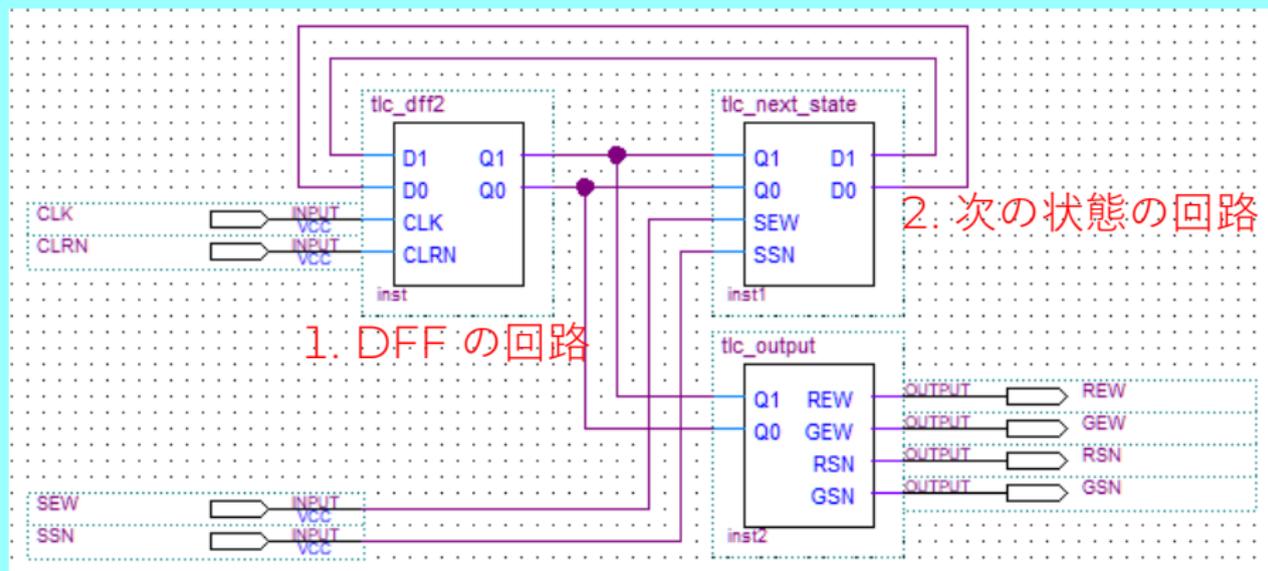
$$G_{EW} = \overline{Q1}$$

$$R_{SN} = \overline{Q1} + Q0$$

$$G_{SN} = Q1$$

交通信号機制御（回路）

全体の回路 tlc_system



3. 出力関数の回路

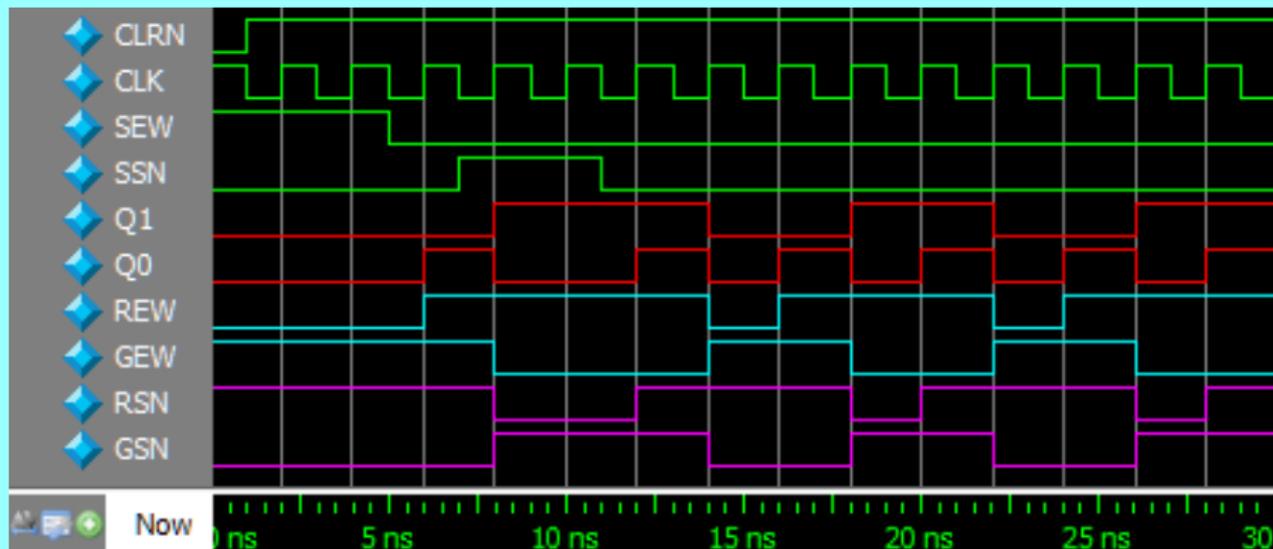
交通信号機制御（テストベンチ）

```
'timescale 1ns/1ps // unit = 1 ns; accuracy = 1 ps
module tlc_sysyem_tb;
  reg  SEW,SSN,CLK,CLRN;
  wire REW,GEW,RSN,GSN;

  tlc_system i1 (.SEW(SEW),.SSN(SSN),.CLK(CLK),.CLRN(CLRN),
                .REW(REW),.GEW(GEW),.RSN(RSN),.GSN(GSN));

  initial begin
    #0 CLK = 1; CLRN = 0; SEW = 1; SSN = 0;
    #1 CLRN = 1;
    #4 SEW = 0;
    #2 SSN = 1;
    #4 SSN = 0;
    #22 $stop;
  end
  always #1 CLK = ~CLK;
endmodule
```

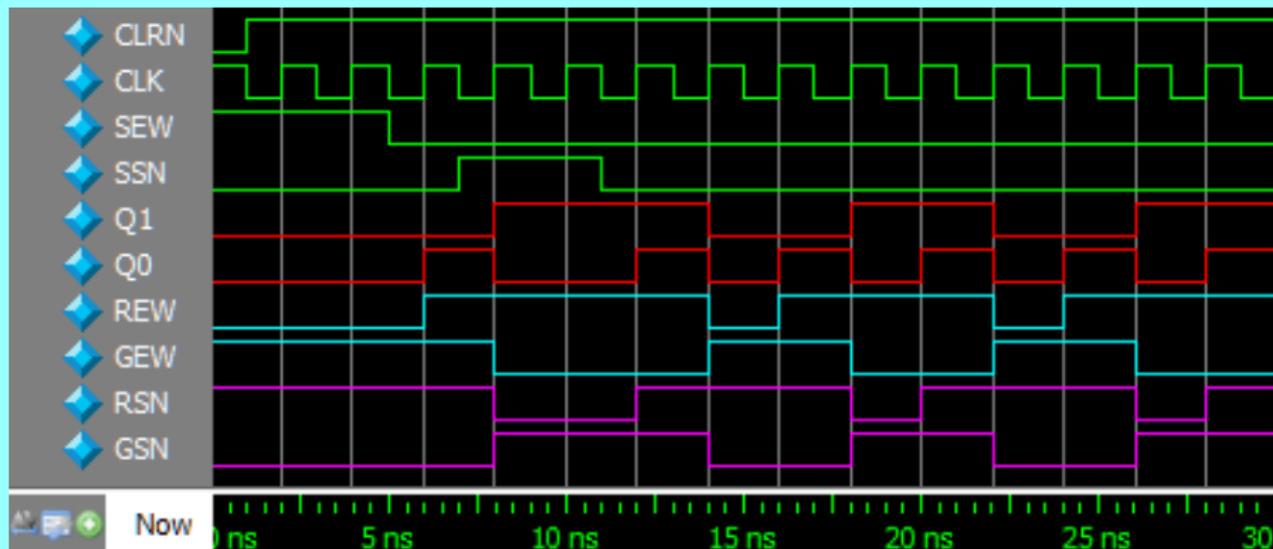
交通信号機制御 (シミュレーション)



| State | 0 | 0 | 0 | 1 | 2 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 |
|----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| L_{EW} | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| L_{SN} | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |

Type `add wave -r /*` in Transcript window to show all signals.

交通信号機制御 (シミュレーション)



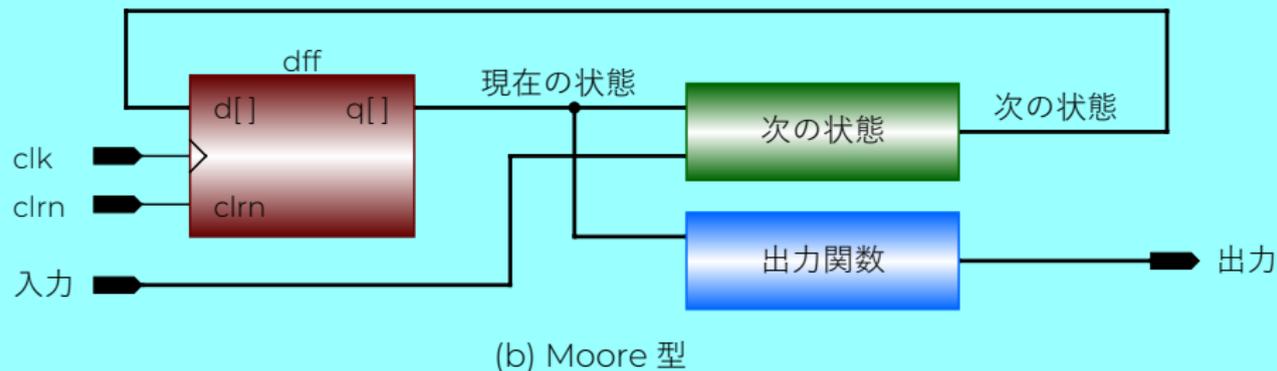
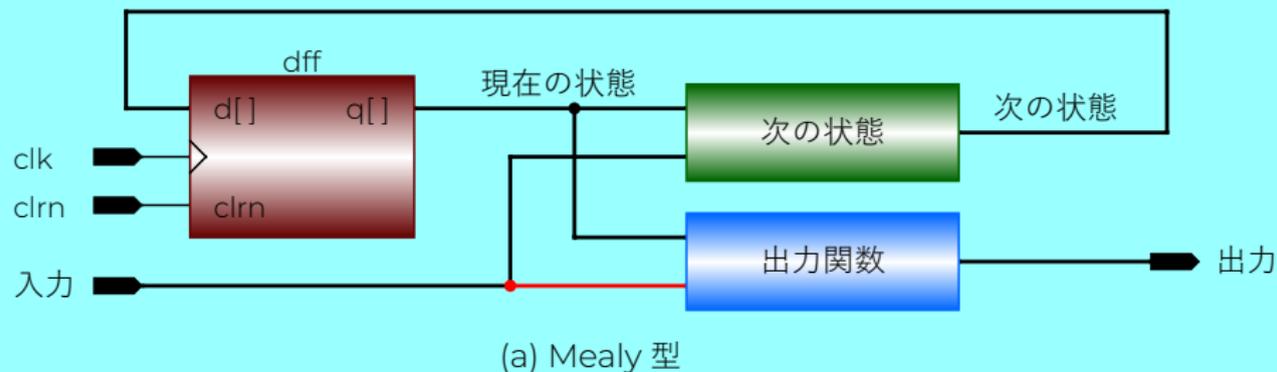
| State | 0 | 0 | 0 | 1 | 2 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 |
|-----------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| L _{EW} | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| L _{SN} | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |

この回路は良くない。車がなくても信号の色がぐるぐる変わる。

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)。

順序回路 — Mealy 型と Moore 型



順序回路のまとめ

まとめ

- 順序回路の基本構造
- 順序回路の分析
- Mealy 型と Moore 型順序回路
- 順序回路設計の手順
- 状態遷移図
- 次の状態の真理値表、論理式、および回路
- 出力関数の真理値表、論理式、および回路
- 交通信号機制御システム

課題 XII (200 点)

交通信号機制御システムを設計し動作検証シミュレーションして下さい。



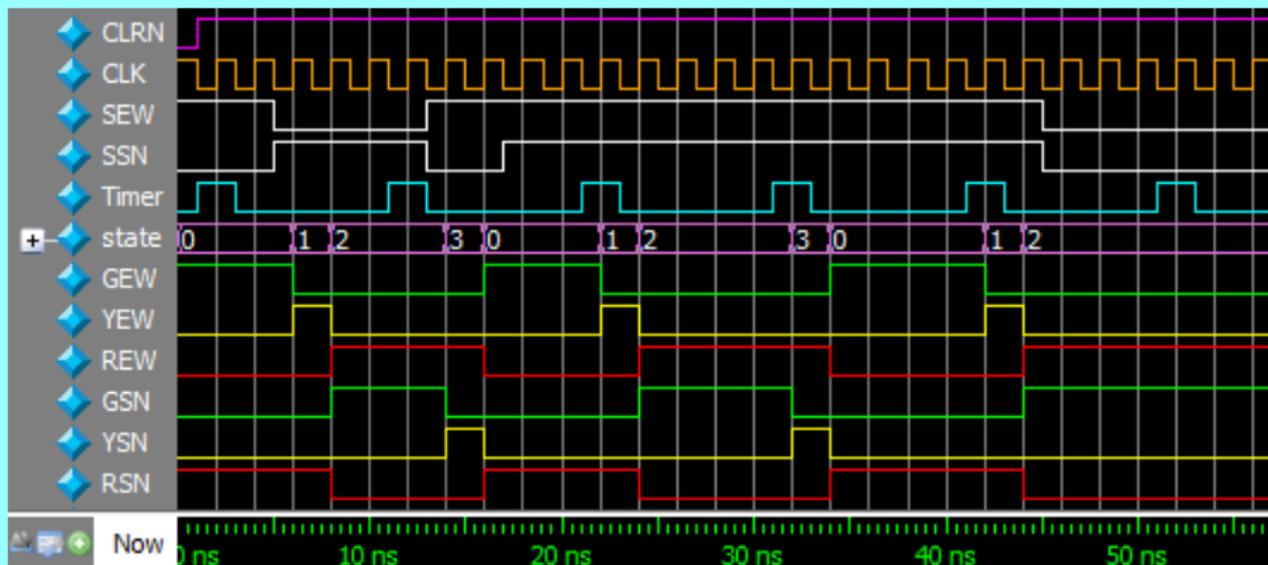
状態の割り当ては次のページの波形を参照してください。注意事項:

- (1) もし $S_{EW} = S_{SN} = 0$, 状態変化なし (Timer: ドントケア)
- (2) もし $S_{EW} \neq S_{SN}$, 状態遷移を有効にする (Timer: ドントケア)
- (3) もし $S_{EW} = S_{SN} = 1$ かつ $Timer = 1$, 状態遷移を有効にする
- (4) もし $S_{EW} = S_{SN} = 1$ かつ $Timer = 0$, 状態遷移を無効にする

テストベンチ [tl_controller_tb.v](#) を使って下さい。

レポートの必須項目: 状態遷移図、真理値表、カルノー図、論理式、回路図、波形、波形の説明。

課題 XII (200 点)



Type `add wave -r /*`
in Transcript window
to show all signals.

Timer = 0, 維持する

発展：自由練習

Verilog HDL による課題の実装

[1]. 2 ビット DFF の設計

```
module t1cdff2 (CLK, CLRN, D, Q);
    input  [1:0] D; // next state
    input          CLK, CLRN;
    output [1:0] Q; // current state
    reg  [1:0] Q; // 2-bit DFFs
    always @(posedge CLK or negedge CLRN) begin
        if (!CLRN) begin
            Q <= 0; // clear DFFs
        end else begin
            Q <= D; // save D to DFFs
        end
    end
end
endmodule
```

発展：自由練習

Verilog HDL による課題の実装

[2]. 次の状態回路の設計

```
module tlcnextstate (Q, SEW, SSN, Timer, D);
    input  [1:0] Q; // current state
    input          SEW, SSN, Timer;
    output [1:0] D; // next state
    assign D[1] =          ; // next state D[1]
    assign D[0] =          ; // next state D[0]
endmodule
```

上記コードを完成しシミュレーションして下さい。

発展：自由練習

Verilog HDL による課題の実装

[3]. 出力関数の回路の設計

```
module tlcoutput (Q, REW, YEW, GEW, RSN, YSN, GSN);
    input  [1:0] Q; // current state
    output      REW, YEW, GEW, RSN, YSN, GSN; // outputs
    assign REW =      ; // EW red
    assign YEW =     ; // EW yellow
    assign GEW =     ; // EW green
    assign RSN =     ; // SN red
    assign YSN =     ; // SN yellow
    assign GSN =     ; // SN green
endmodule
```

上記コードを完成しシミュレーションして下さい。

発展：自由練習

Verilog HDL による課題の実装

[全体]. 交通信号機制御の回路の設計

```
module tlctop (CLK, CLRN, SEW, SSN, Timer, REW, YEW, GEW, RSN, YSN, GSN);
    input      CLK, CLRN, SEW, SSN, Timer;    // inputs
    output     REW, YEW, GEW, RSN, YSN, GSN; // outputs
    wire [1:0] D;                             // internal wire
    wire [1:0] Q;                             // internal wire
    tlcdff2    i1 (                            ); // [1]. 2-bit D flip flops
    tlcnextstate i2 (                          ); // [2]. next state
    tlcoutput   i3 (                          ); // [3]. output function
endmodule
```

上記コードを完成しシミュレーションして下さい。