

論理回路入門 (14)

論理回路の復習とクイズ

李 亜民

2023年1月10日(火)

課題 XIV (最終課題 (テスト): 30%)

最終課題 (テスト): 30% 成績
授業内テストとレポートの提出

公開: 2023 年 1 月 10 日 (火) 15:20

締切: 2023 年 1 月 10 日 (火) 16:50

内容: Quartus II と ModelSim を使用し、
組み合わせ回路と順序回路を
設計とシミュレーションする

レポートを提出する

締切: 2023 年 1 月 10 日 (火) 16:50

いま、もうすぐ、始まる。

論理回路入門

勉強した内容

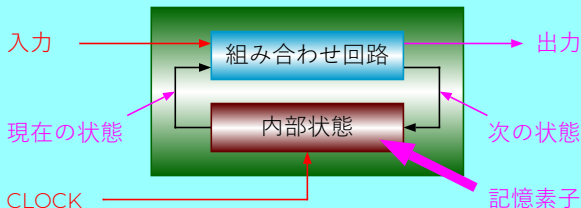
- **論理回路の基本**: ブール代数、真理値表、カルノー図、論理式の積和形と和積形、EDA ツール
- **組合せ回路**: 半加算器、全加算器、リップルキャリーアダー、キャリールックアヘッドアダー、ツリー型桁上げ先見加算器、減算器、乗算器、ウォレスツリー乗算器、マルチプレクサ、7セグメントLED、ALU、デコーダ、エンコーダ
- **順序回路**: RS ラッチ、D ラッチ、D フリップフロップ、T フリップフロップ、JK フリップフロップ、レジスタ・ファイル、状態遷移図、交通信号機制御システム、N 進カウンター

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。



基本的な論理演算

① AND (アンド)

- ▶ 論理積 (かつ)
- ▶ 『例』 $F = A \text{ AND } B = A \cdot B = A \text{ } B$

② OR (オア)

- ▶ 論理和 (または)
- ▶ 『例』 $F = A \text{ OR } B = A + B$

③ NOT (ノット)

- ▶ 否定
- ▶ 『例』 $F = \text{NOT } A = \bar{A}$

よく使われた7ゲート

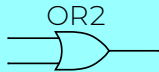
● AND

$$F = A \cdot B$$



● OR

$$F = A + B$$



● NOT

$$F = \bar{A}$$



● NAND

$$F = \overline{A \cdot B}$$



● NOR

$$F = \overline{A + B}$$



● XOR

$$F = A \oplus B$$



● XNOR

$$F = A \odot B = \overline{A \oplus B}$$



ブール代数の定理

- 1 $A \cdot A = A$ 冪等律
- 2 $A + A = A$ 冪等律
- 3 $A \cdot B = B \cdot A$ 交換律
- 4 $A + B = B + A$ 交換律
- 5 $(A \cdot B) \cdot C = A \cdot (B \cdot C)$ 結合律
- 6 $(A + B) + C = A + (B + C)$ 結合律
- 7 $A \cdot (B + C) = (A \cdot B) + (A \cdot C)$ 分配律
- 8 $A + (B \cdot C) = (A + B) \cdot (A + C)$ 分配律
- 9 $A \cdot (A + B) = A$ 吸収律
- 10 $A + (A \cdot B) = A$ 吸収律

ブール代数の定理

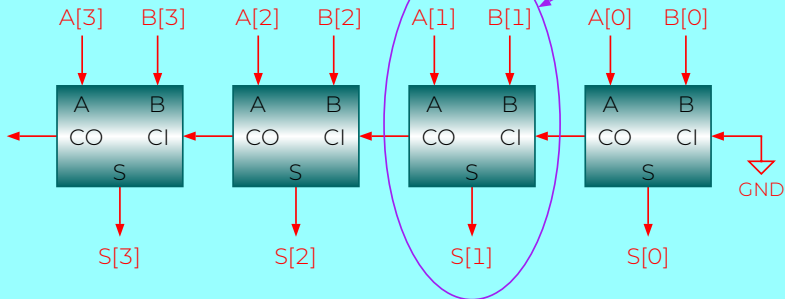
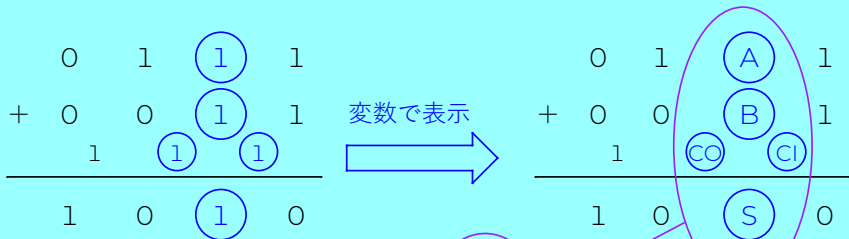
- 11 $A \cdot 0 = 0$ 支配律
- 12 $A + 1 = 1$ 支配律
- 13 $A \cdot 1 = A$ 同一律
- 14 $A + 0 = A$ 同一律
- 15 $A \cdot \bar{A} = 0$ 補元律
- 16 $A + \bar{A} = 1$ 補元律
- 17 $\overline{\bar{A}} = A$ 対合律

- 18 $\overline{A \cdot B} = \bar{A} + \bar{B}$ ド・モルガンの法則
- 19 $\overline{A + B} = \bar{A} \cdot \bar{B}$ ド・モルガンの法則

組み合わせ回路設計の手順

- ① 問題を理解する。
- ② 入力と出力信号の名前を任意に決める。
- ③ 入力のあらゆる組み合わせを列挙し，真理値表をつくる。
- ④ 真理値表から論理式をつくる。
(どの組み合わせで出力が 1 になるか。)
- ⑤ 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)。

例：全加算器回路設計



例：全加算器の真理値表

input A, B, CI; // CI: Carry In (下位桁からの繰り上がり)

output CO, S; // CO: Carry Out (上位桁への繰り上がり), S: Sum (和)

| A | B | CI | CO | S | コメント |
|---|---|----|----|---|-----------------------|
| 0 | 0 | 0 | 0 | 0 | $0 + 0 + 0 = 00$ (加算) |
| 0 | 0 | 1 | 0 | 1 | $0 + 0 + 1 = 01$ (加算) |
| 0 | 1 | 0 | 0 | 1 | $0 + 1 + 0 = 01$ (加算) |
| 0 | 1 | 1 | 1 | 0 | $0 + 1 + 1 = 10$ (加算) |
| 1 | 0 | 0 | 0 | 1 | $1 + 0 + 0 = 01$ (加算) |
| 1 | 0 | 1 | 1 | 0 | $1 + 0 + 1 = 10$ (加算) |
| 1 | 1 | 0 | 1 | 0 | $1 + 1 + 0 = 10$ (加算) |
| 1 | 1 | 1 | 1 | 1 | $1 + 1 + 1 = 11$ (加算) |

出力の CO、S は $A + B + CI$ の結果 (和) を 2 進数で表した数値。

$$00_2 = 0_{10}$$

$$01_2 = 1_{10}$$

$$10_2 = 2_{10}$$

$$11_2 = 3_{10}$$

例：全加算器の論理式

真理値表から論理式をつくる

— どの組み合わせで出力が 1 になるか

カルノー図による論理式を簡単化する (CO)

S のカルノー図

A: 0 0 1 1
B: 0 1 1 0

| | | | | |
|-------|---|---|---|---|
| Cl: 0 | | 1 | | 1 |
| 1 | 1 | | 1 | |

CO のカルノー図

A: 0 0 1 1
B: 0 1 1 0

| | | | | |
|-------|--|---|---|---|
| Cl: 0 | | | 1 | |
| 1 | | 1 | 1 | 1 |

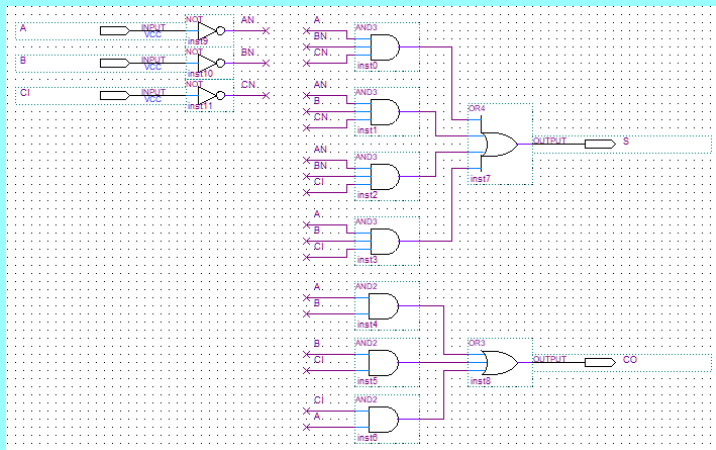
BCI AB ACI

$$S = A \bar{B} \bar{C}I + \bar{A} B \bar{C}I + \bar{A} \bar{B} CI + A B CI$$

$$CO = A B + A CI + B CI$$

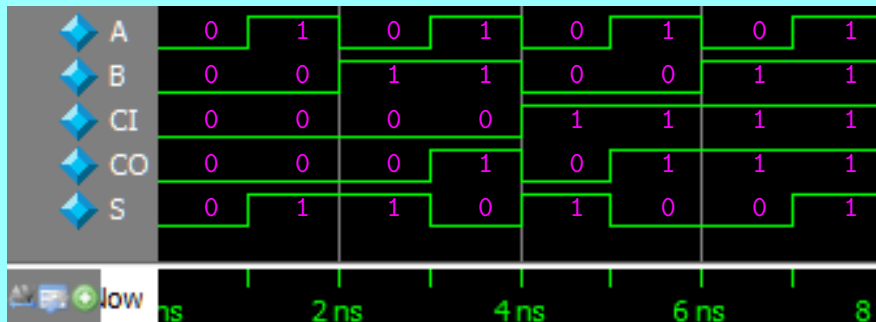
例：全加算器の回路図

$$S = A \bar{B} \bar{C} + \bar{A} B \bar{C} + \bar{A} \bar{B} C + A B C$$
$$CO = A B + A C + B C$$



例：全加算器のシミュレーション

$$S = A \bar{B} \bar{C} I + \bar{A} B \bar{C} I + \bar{A} \bar{B} C I + A B C I$$
$$CO = A B + A C I + B C I$$



↑
 $0 + 0 + 0 = 00$

↑
 $1 + 1 + 1 = 11$

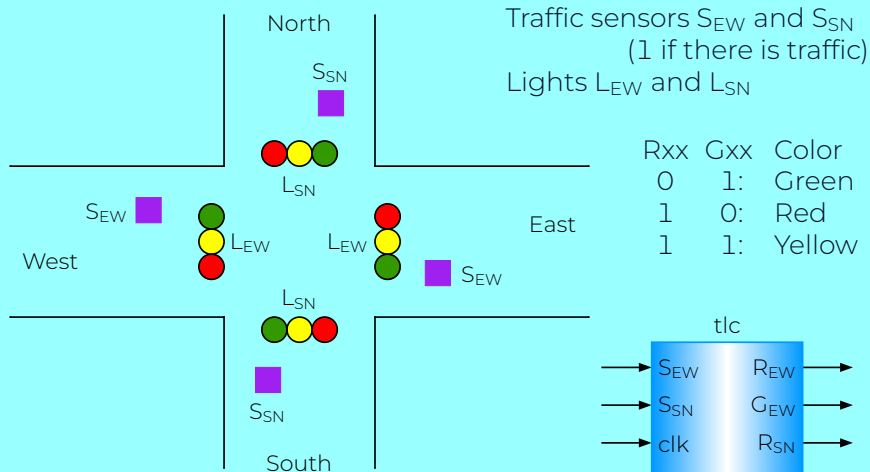
他の組合せ回路

- 1 半加算器
- 2 リップルキャリーアダー
- 3 キャリールックアヘッドアダー
- 4 ツリー型桁上げ先見加算器
- 5 減算器
- 6 乗算器
- 7 ウォレス ツリー乗算器
- 8 マルチプレクサ
- 9 7セグメント LED
- 10 ALU
- 11 デコーダ (2-4、3-8、4-16)
- 12 エンコーダ (4-2、8-3、16-4)、プライオリティエンコーダ

順序回路設計の手順

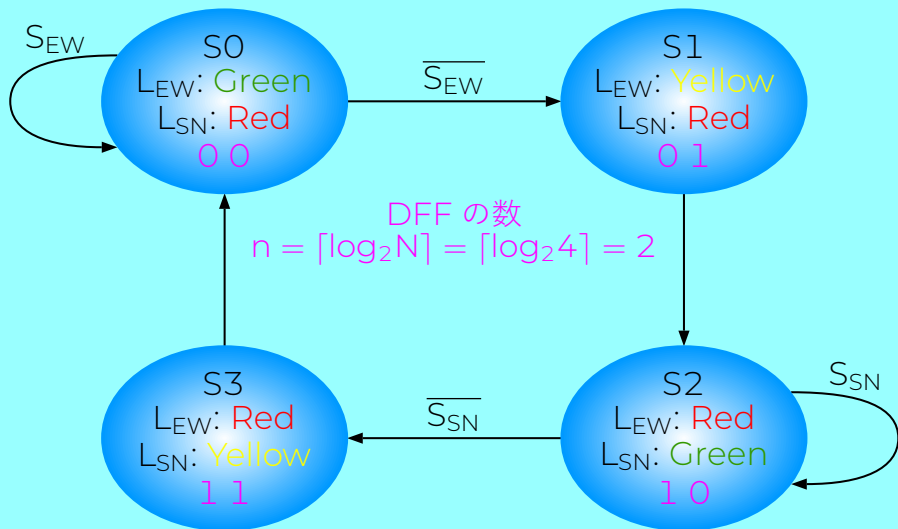
- 1 問題を理解する。
- 2 状態遷移図をつくる。
- 3 FF の数を決める ($n = \lceil \log_2 N \rceil$ 、 N は状態の数)。
- 4 各状態に n ビットの番号を付け、真理値表をつくる。
 - ▶ 次の状態の真理値表をつくる。
 - ▶ 出力関数の真理値表をつくる。
- 5 真理値表から論理式をつくる (どの条件で出力が 1 になるか)。
 - ▶ カルノー図を用いて論理式を簡単化する。
- 6 論理式から回路をつくる。
- 7 テストベンチをつくる (シミュレーションするため)。
- 8 回路をシミュレーションする (回路設計の正当性検証)。
- 9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

例：交通信号機制御システム



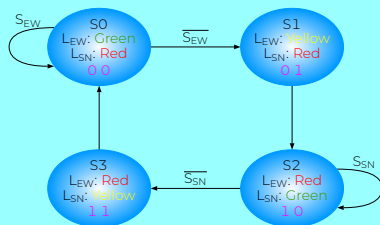
EW: East and West (東西)
SN: South and North (南北)

例：交通信号機制御（状態遷移図）



例：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



| 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 |

例：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 | 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 | 0 | 0 | 1 | 1 |
| | Q0 | 0 | 1 | 1 | 0 |
| S _{EW} S _{SN} | 00 | 1 | | | 1 |
| | 01 | 1 | | | |
| | 11 | | | | |
| | 10 | | | | 1 |

例：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 |

例：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 |

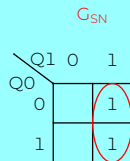
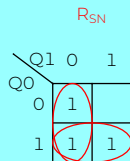
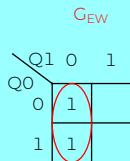
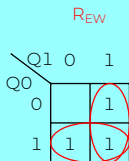
出力関数の論理式

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

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

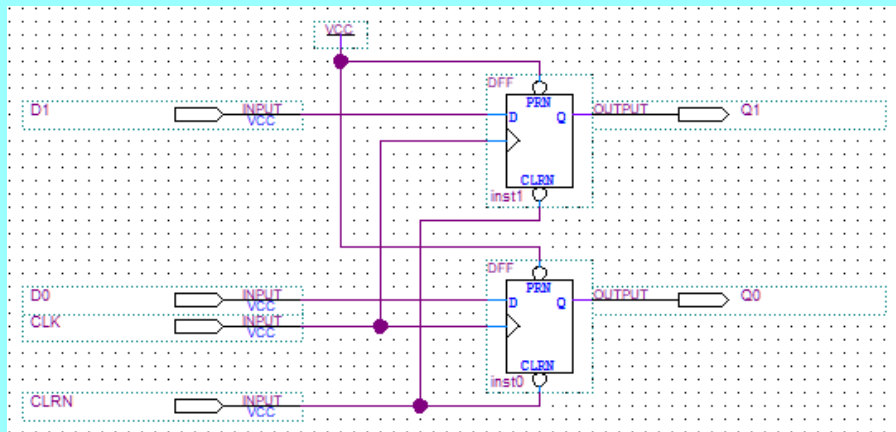
$$R_{SN} = \overline{Q1} + Q0$$

$$G_{SN} = Q1$$



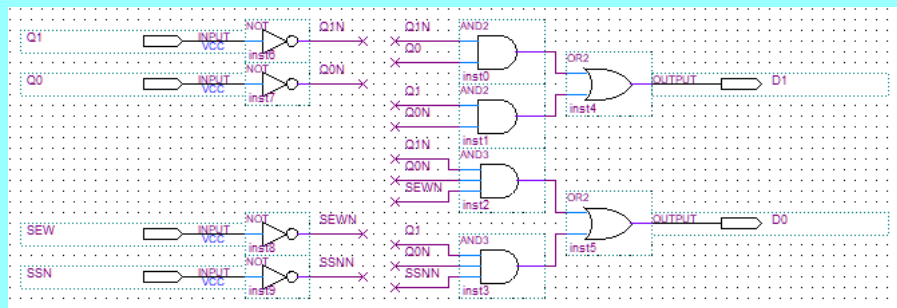
例：交通信号機制御（回路）

1. DFF の回路 tlc_dff2



例：交通信号機制御（回路）

2. 次の状態の回路 tlc_next_state

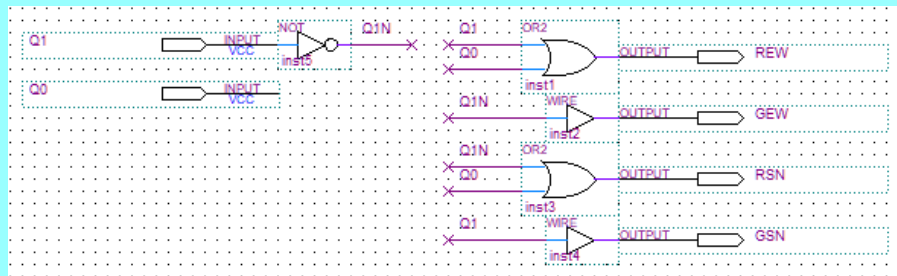


$$D1 = \overline{Q1} \cdot Q0 + Q1 \cdot \overline{Q0}$$

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

例：交通信号機制御（回路）

3. 出力関数の回路 tlc_output



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

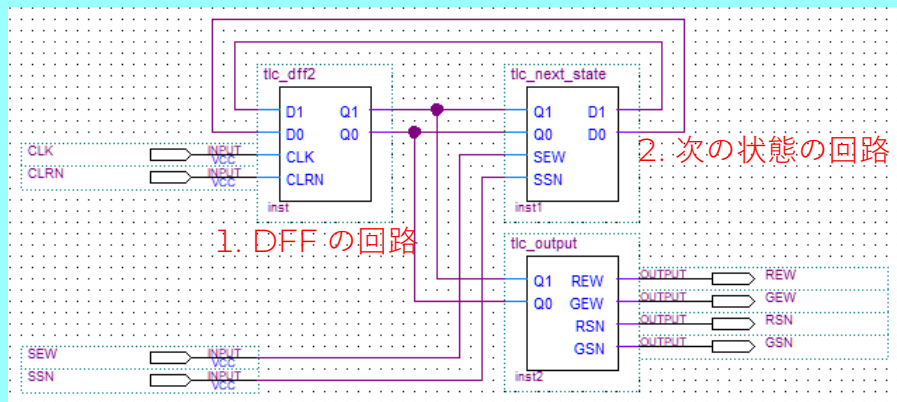
$$G_{EW} = \overline{Q1}$$

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

$$G_{SN} = Q1$$

例：交通信号機制御（回路）

全体の回路 tlc_system



3. 出力関数の回路

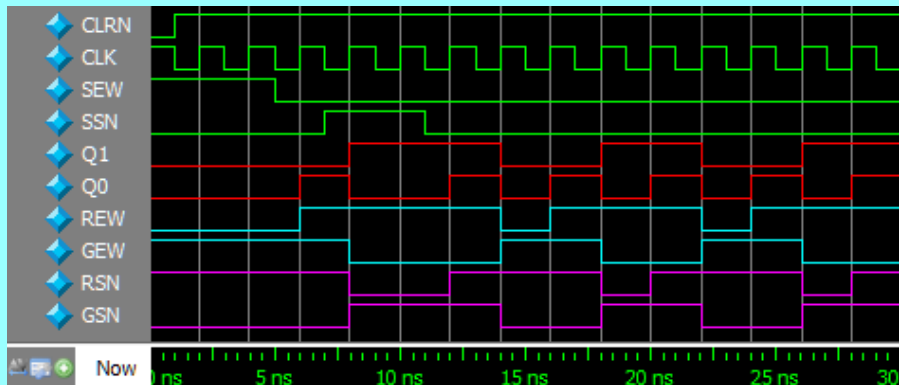
例：交通信号機制御（テストベンチ）

```
'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
```

例：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} | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |

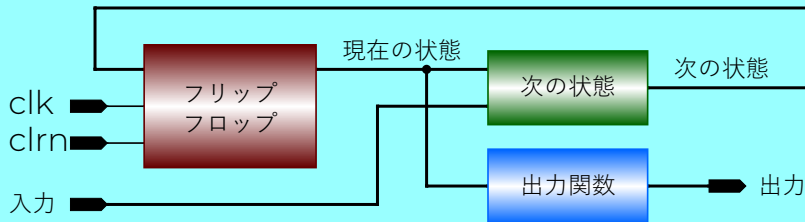
Type `add wave -r /*` in Transcript window to show all signals.

同期式N進カウンターの設計

7セグメントLEDを使い同期式N進 ($0 \sim N - 1$) カウントアップ ($u = 1$)/カウントダウン ($u = 0$) する回路をXFFを用いて設計せよ (Don't Care項を利用)。7セグメントLED: 0: 点灯 1: 消灯。

$N = 3, 4, 6, 8, 10, 16$

$X = D, JK, T$



課題 XIV (最終課題 (テスト): 30%)

最終課題 (テスト): 30% 成績
授業内テストとレポートの提出

公開: 2023 年 1 月 10 日 (火) 15:20

締切: 2023 年 1 月 10 日 (火) 16:50

内容: Quartus II と ModelSim を使用し、
組み合わせ回路と順序回路を
設計とシミュレーションする

レポートを提出する

締切: 2023 年 1 月 10 日 (火) 16:50

いま、もうすぐ、始まる。