論理回路

第10回 多状態順序回路の設計

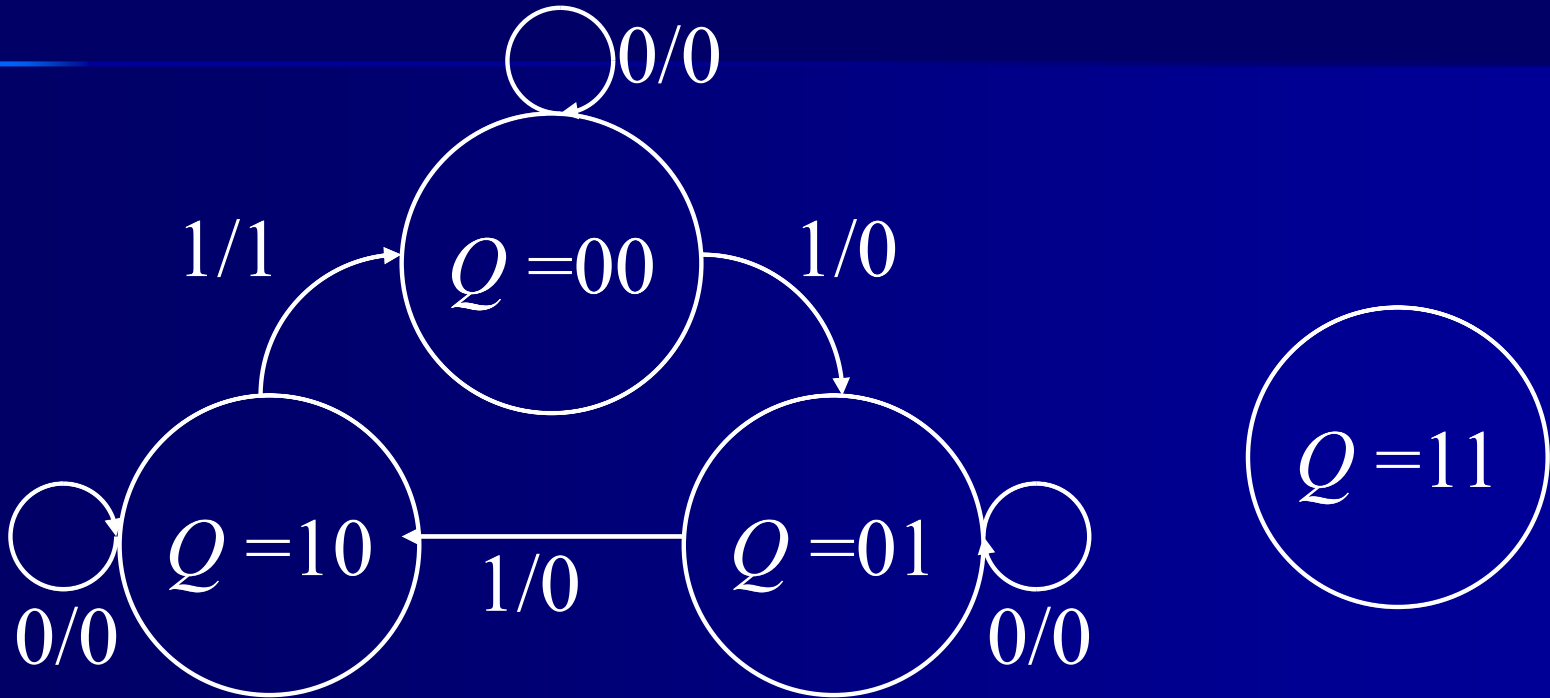
http://www.info.kindai.ac.jp/LC
E館3階E-331 内線5459
takasi-i@info.kindai.ac.jp

不完全指定論理関数と完全指定論理関数

2^n個の状態を持つ(n個のFFを持つ)論理関数に対して

- 定義: 完全指定論理関数
  - 全ての状態に対する状態遷移関数,出力関数が定義されている論理関数
- 定義: 不完全指定論理関数
  - ある状態に対する状態遷移関数,出力関数が定義されていない論理関数

不完全指定論理関数

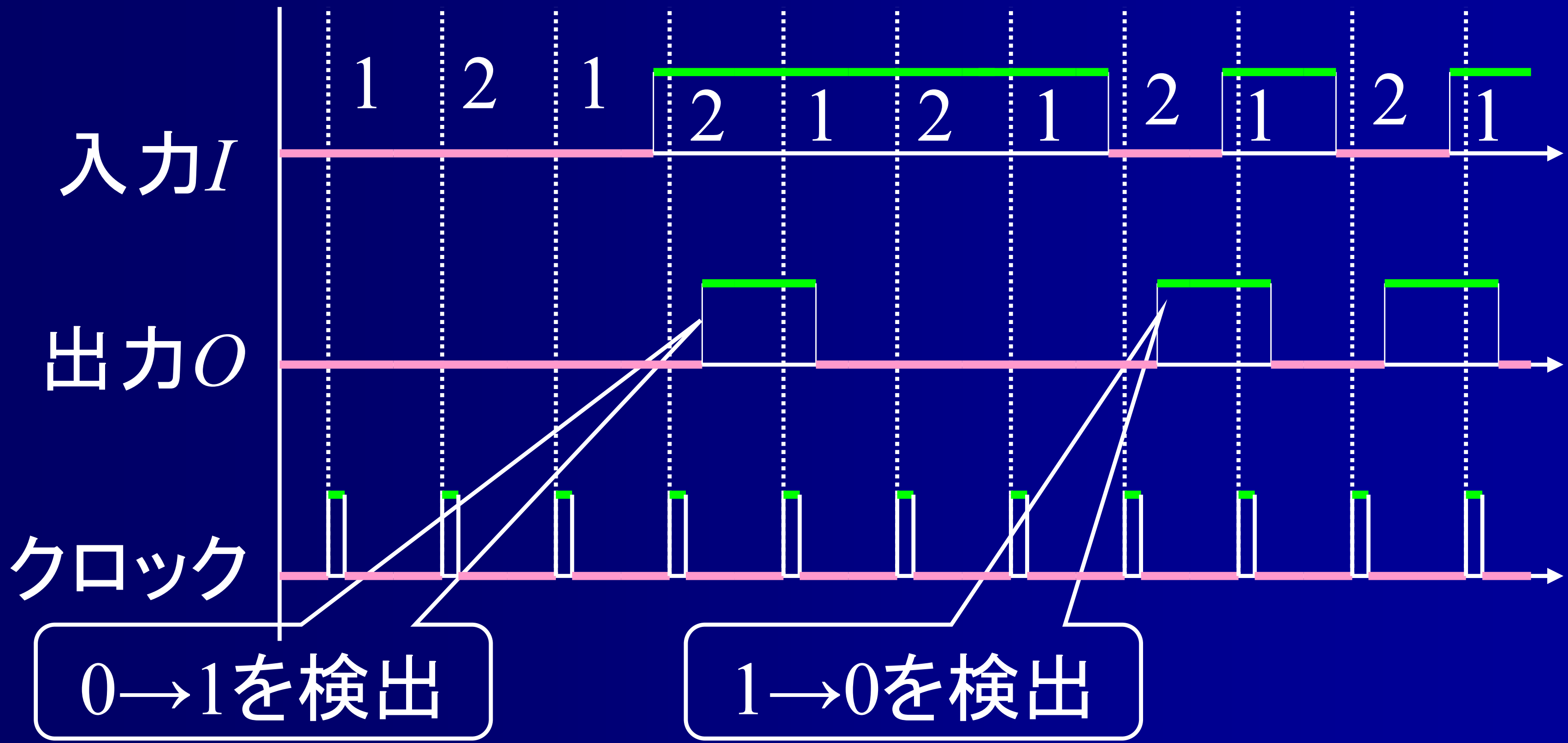


状態$Q=11$は状態遷移関数,
出力関数が未定義

状態11はドントケア

2ビット列比較回路の設計

- 連続して入力された 2 ビットが同じなら 0、異なれば 1を出力する
- 1ビット目入力時は0を出力



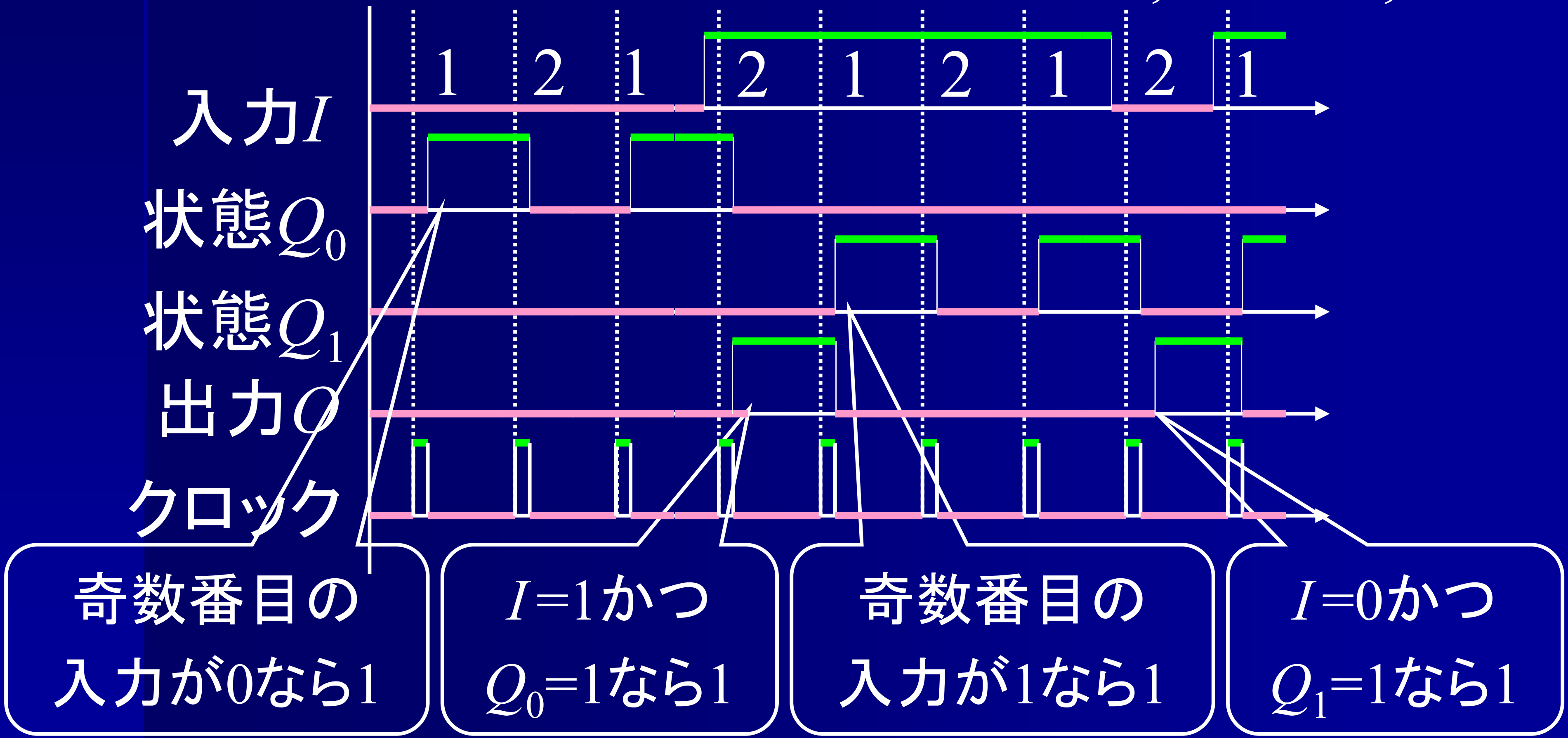
同期式回路の設計

1. 入力($I_1, I_2, \ldots I_m$),出力($O_1, O_2, \ldots O_n$), 状態($Q_1, Q_2, \ldots Q_k$)を決める
2. 状態遷移図を描く
3. 状態遷移表を作成する
4. 拡大入力要求表を作成する
5. FFの入力条件式を求める
6. 出力関数を求める
7. 回路図を描く

1.入力,出力,状態の決定

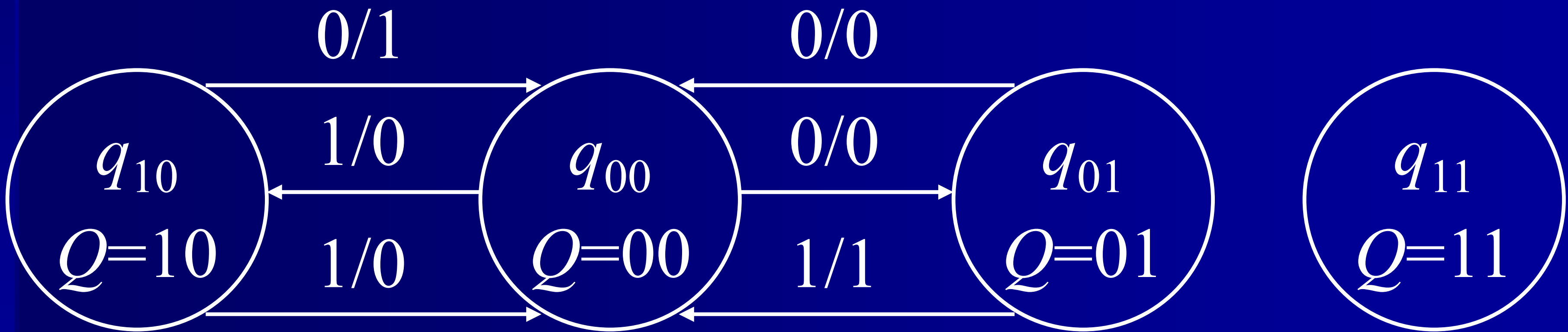
- 入力が奇数番目か偶数番目か？
- 奇数番目の入力が0か1か？

⇒ FF2個で記憶可能　1入力, 1出力, 2FF



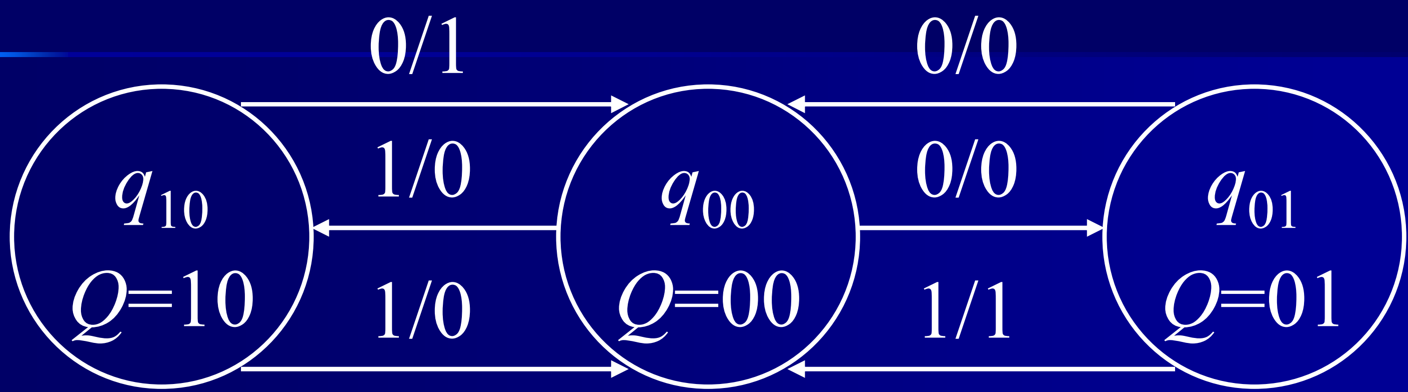
2.状態遷移図を描く

- 状態Q_0：奇数番目の入力が0なら1
- 状態Q_1：奇数番目の入力が1なら1
- 出力O：$Q_0=1$ かつ $I=1$ または
  $Q_1=1$ かつ $I=0$ ならば 1 を出力



状態11はドントケア

3.状態遷移表を作成する



I	Q_1	Q_0	Q_1^+	Q_0^+	O
0	0	0	0	1	0
1			1	0	0
0	0	1	0	0	0
1			0	0	1

石水隆1

I	Q_1	Q_0	Q_1^+	Q_0^+	O
0	1	0	0	0	1
1			0	0	0
0	1	1	-	-	-
1			-	-	-

スライド 8

石水隆1　　石水隆, 2020/06/28

4.拡大入力表を作る

I	Q_1	Q_0	Q_1^+	Q_0^+	O	D_1	D_0
0	0	0	0	1	0	0	1
1			1	0	0	1	0
0	0	1	0	0	0	0	0
1			0	0	1	0	0
0	1	0	0	0	1	0	0
1			0	0	0	0	0
0	1	1	-	-	-	-	-
1			-	-	-	-	-

5.FFの入力条件式,
6.出力関数を求める

I	Q_1	Q_0	O	D_1	D_0
0	0	0	0	0	1
1			0	1	0
0	0	1	0	0	0
1			1	0	0
0	1	0	1	0	0
1			0	0	0
0	1	1	-	-	-
1			-	-	-

$$D_1 = I \cdot \overline{Q_1} \cdot \overline{Q_0}$$

$$D_0 = \overline{I} \cdot \overline{Q_1} \cdot \overline{Q_0}$$

$$O = \overline{I} \cdot Q_1 + I \cdot Q_0$$

D_1

I \ Q_1Q_0	00	01	11	10
0			-	
1	1		-	

D_0

I \ Q_1Q_0	00	01	11	10
0	1		-	
1			-	

O

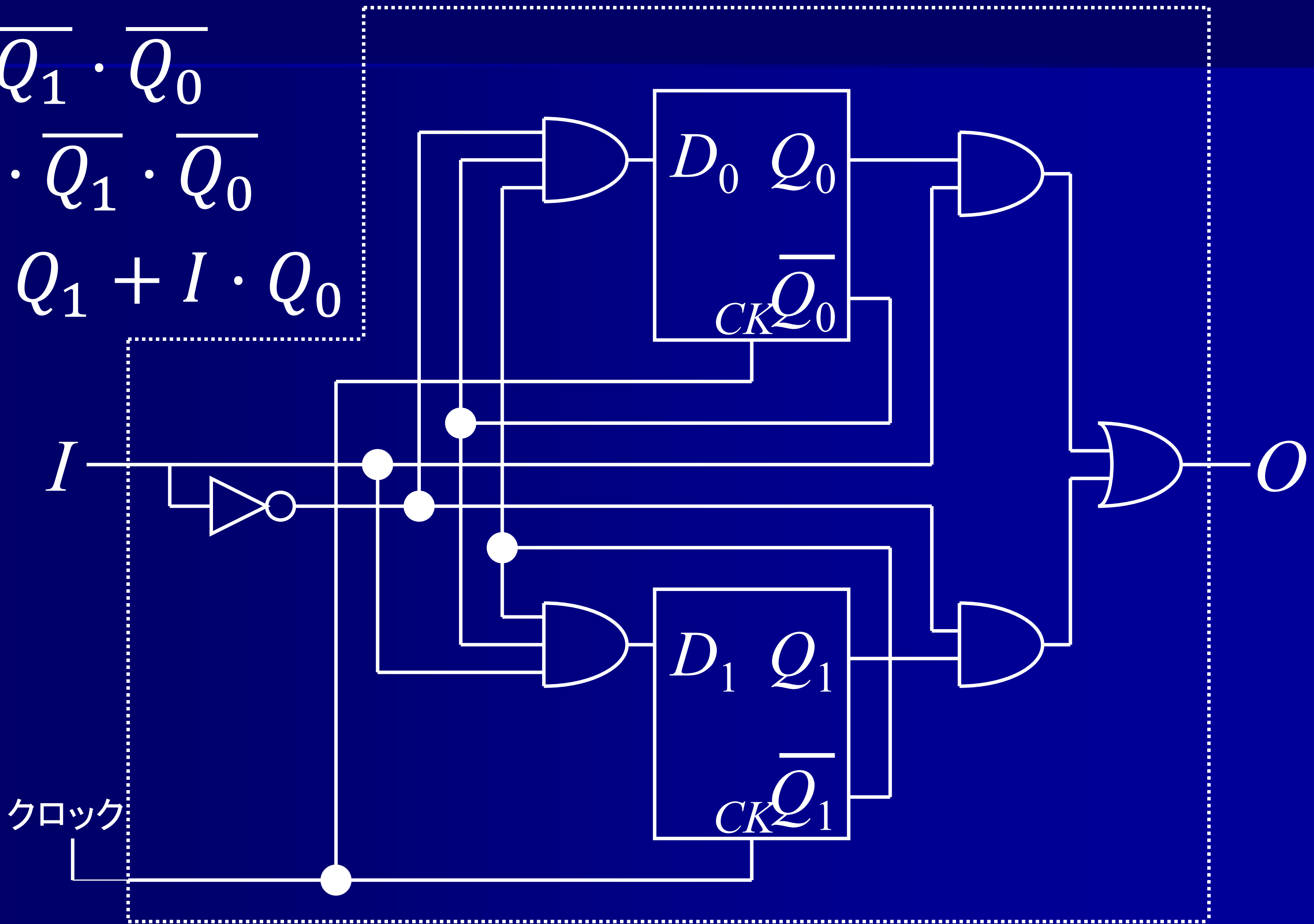
I \ Q_1Q_0	00	01	11	10
0			-	1
1		1	-	

7.回路図を描く

$$D_1 = I \cdot \overline{Q_1} \cdot \overline{Q_0}$$

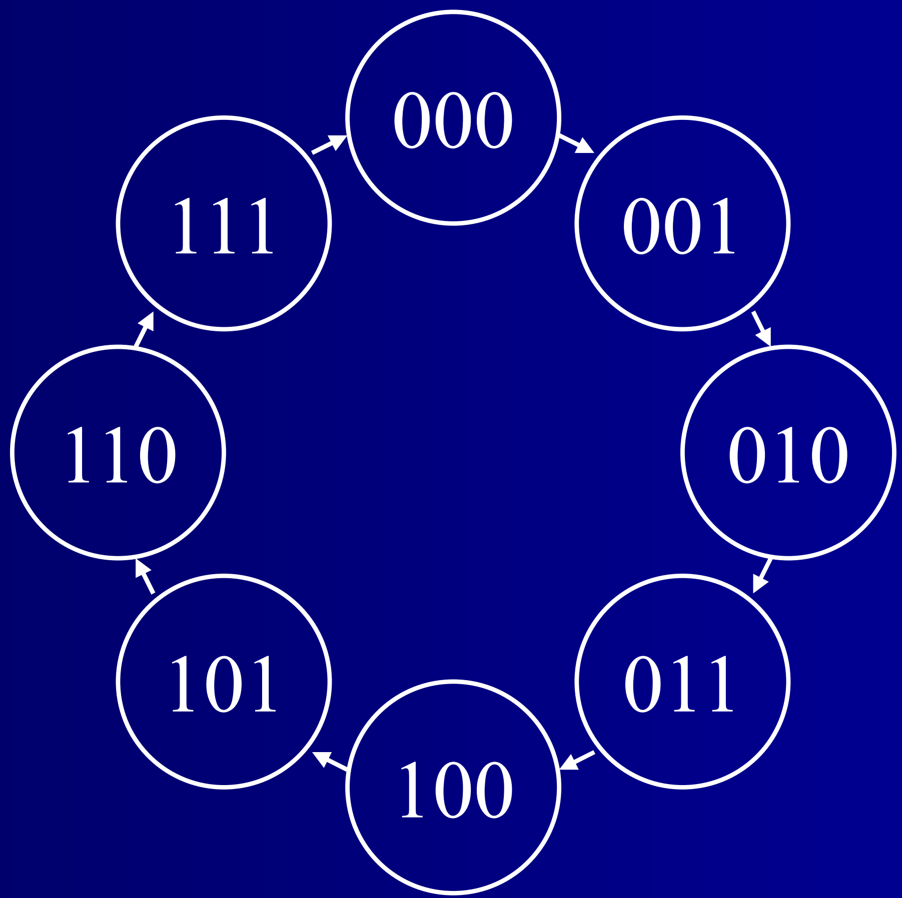
$$D_0 = \overline{I} \cdot \overline{Q_1} \cdot \overline{Q_0}$$

$$O = \overline{I} \cdot Q_1 + I \cdot Q_0$$



同期式8進分周器(カウンタ)の設計

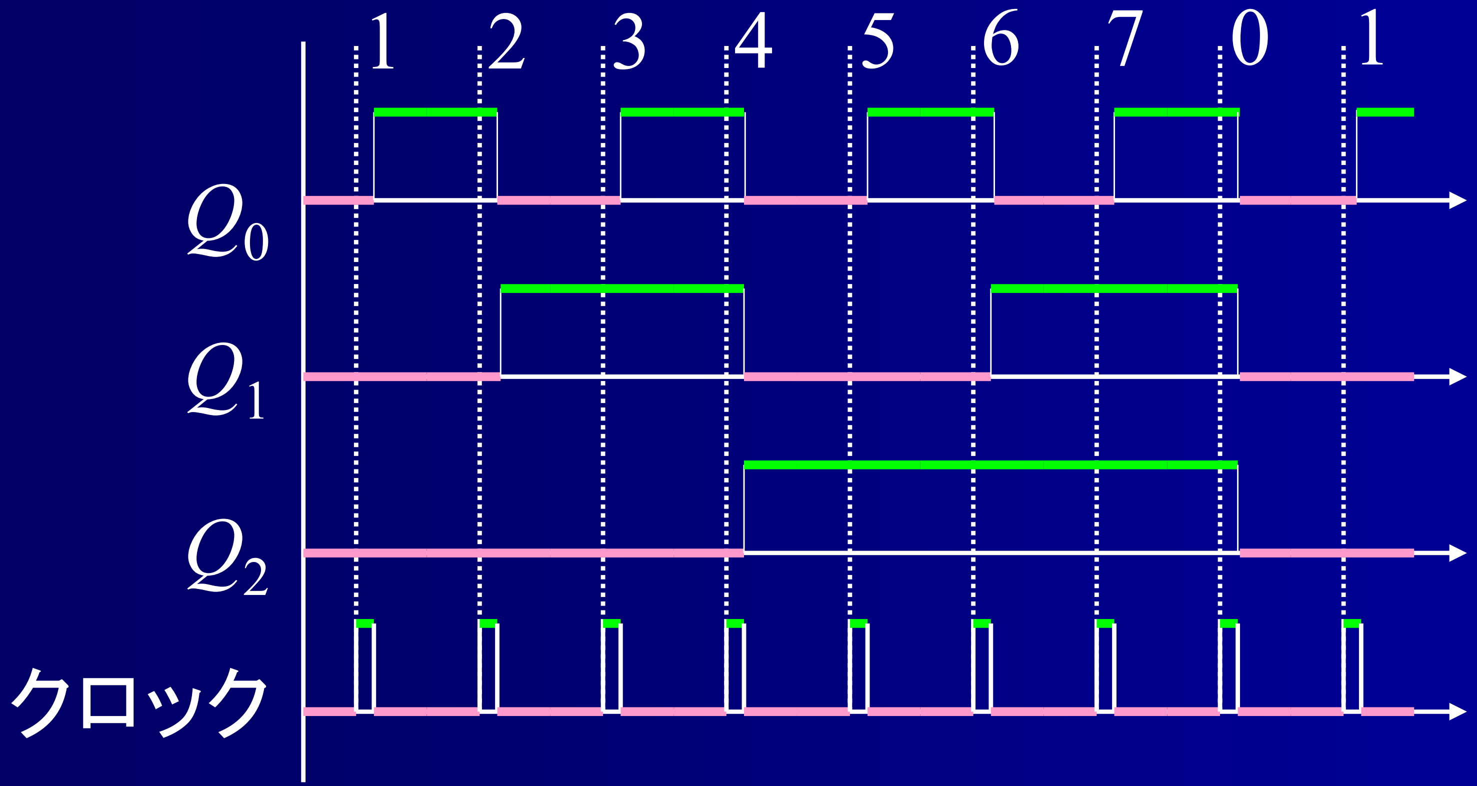
- 3ビット状態(Q_2, Q_1, Q_0)が 000→001→…→111→000 と遷移する回路



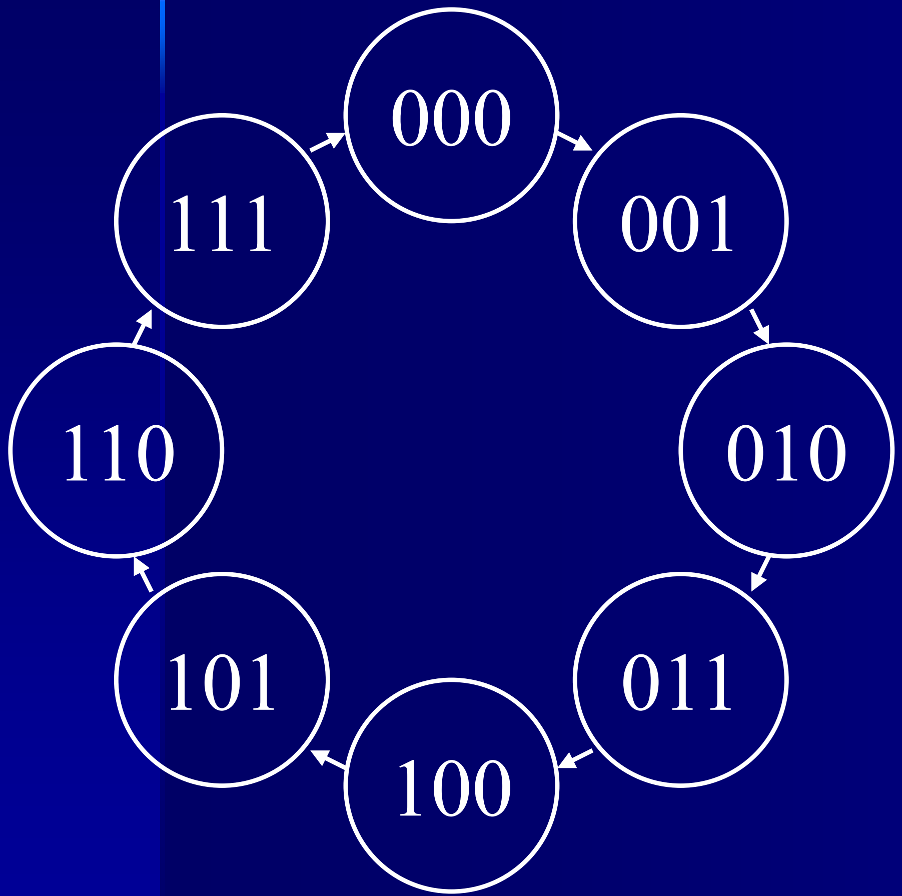
1.入力,出力,状態の決定

- 入力,出力は無し
- 3ビットであるのでFF3個で記憶可能

0入力, 0出力, 3FF



2.状態遷移図,3.状態遷移表を描く



Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

4.拡大入力表を作る

Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	J_2K_2	J_1K_1	J_0K_0
0	0	0	0	0	1	0 -	0 -	1 -
0	0	1	0	1	0	0 -	1 -	- 1
0	1	0	0	1	1	0 -	- 0	1 -
0	1	1	1	0	0	1 -	- 1	- 1
1	0	0	1	0	1	- 0	0 -	1 -
1	0	1	1	1	0	- 0	1 -	- 1
1	1	0	1	1	1	- 0	- 0	1 -
1	1	1	0	0	0	- 1	- 1	-1

5.FFの入力条件式を求める

Q_2	Q_1	Q_0	J_2K_2	J_1K_1	J_0K_0
0	0	0	0 -	0 -	1 -
0	0	1	0 -	1 -	- 1
0	1	0	0 -	- 0	1 -
0	1	1	1 -	- 1	- 1
1	0	0	- 0	0 -	1 -
1	0	1	- 0	1 -	- 1
1	1	0	- 0	- 0	1 -
1	1	1	- 1	- 1	-1

J_2K_2

Q_2 \ Q_1Q_0	00	01	11	10
0	0 -	0 -	1 -	0 -
1	- 0	- 0	- 1	- 0

J_1K_1

Q_2 \ Q_1Q_0	00	01	11	10
0	0 -	1 -	- 1	- 0
1	0 -	1 -	- 1	- 0

J_0K_0

Q_2 \ Q_1Q_0	00	01	11	10
0	1 -	- 1	- 1	1 -
1	1 -	- 1	- 1	1 -

$J_2 = Q_1 \cdot Q_0 \quad K_2 = Q_1 \cdot Q_0$

$J_1 = Q_0 \quad K_1 = Q_0$

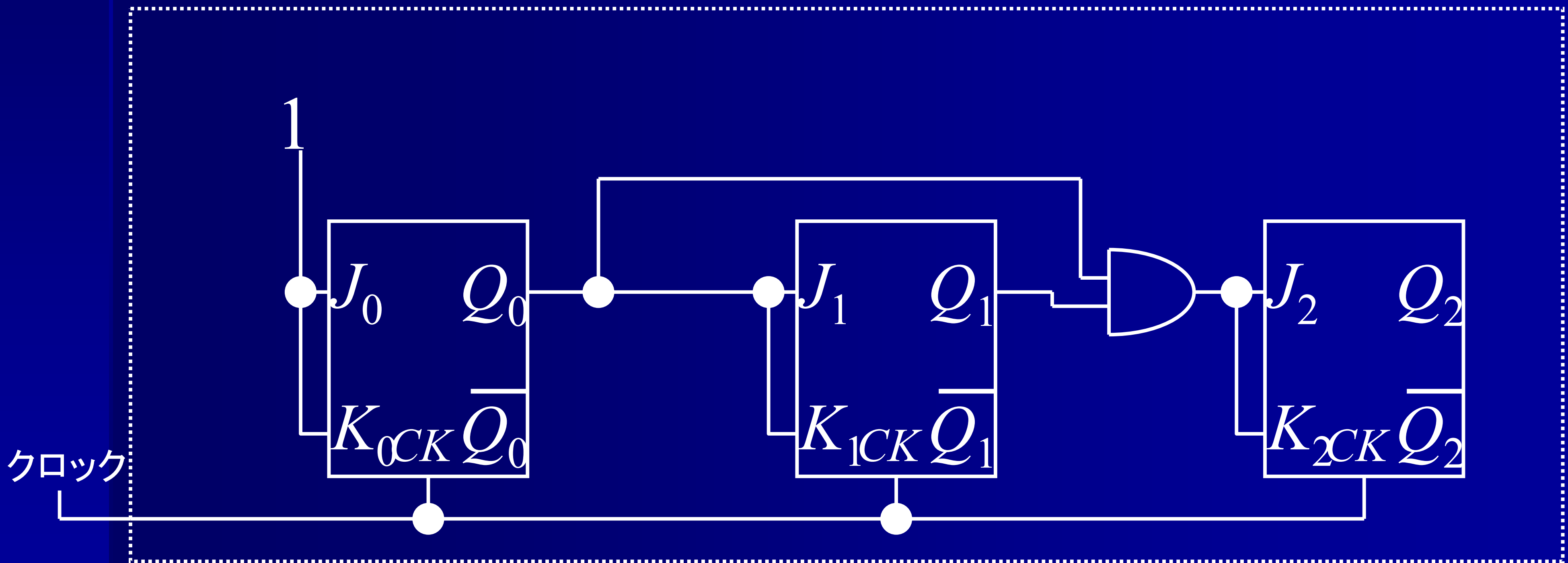
$J_0 = 1 \quad K_0 = 1$

7.回路図を描く

$$J_2 = Q_1 \cdot Q_0 \quad K_2 = Q_1 \cdot Q_0$$

$$J_1 = Q_0 \quad K_1 = Q_0$$

$$J_0 = 1 \quad K_0 = 1$$

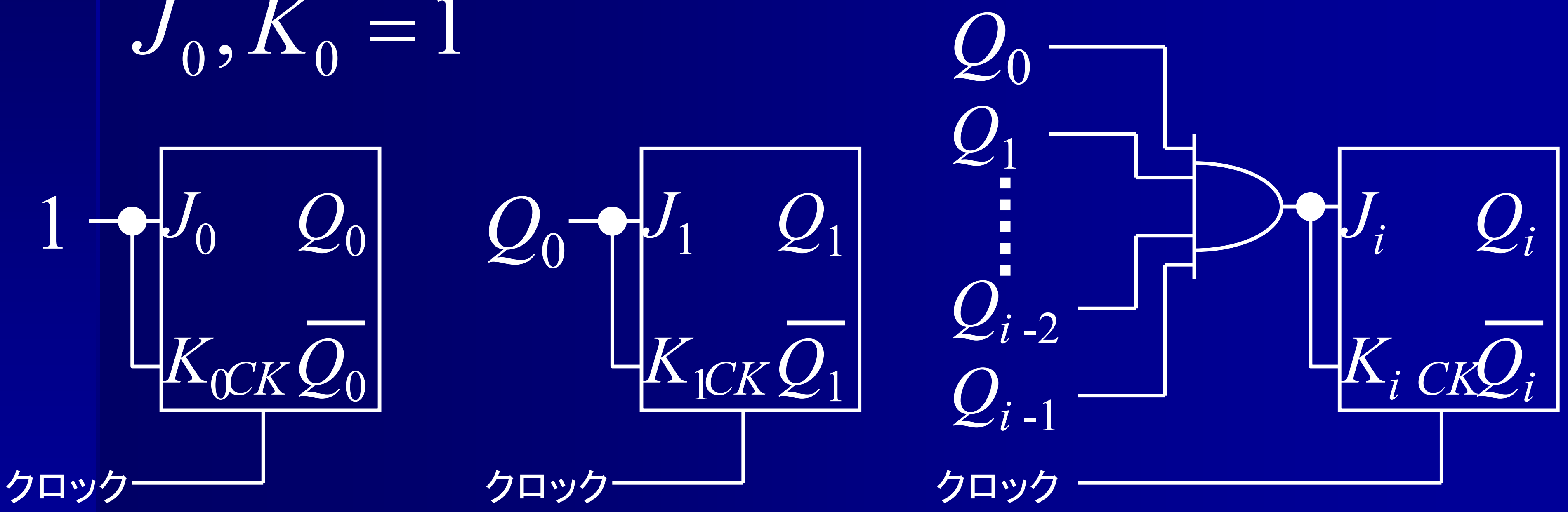


同期式2^n進分周器

$$J_i, K_i = Q_{i-1} \cdot Q_{i-2} \cdot \ldots \cdot Q_1 \cdot Q_0 \quad (2 \le i \le n)$$

$$J_1, K_1 = Q_0$$

$$J_0, K_0 = 1$$

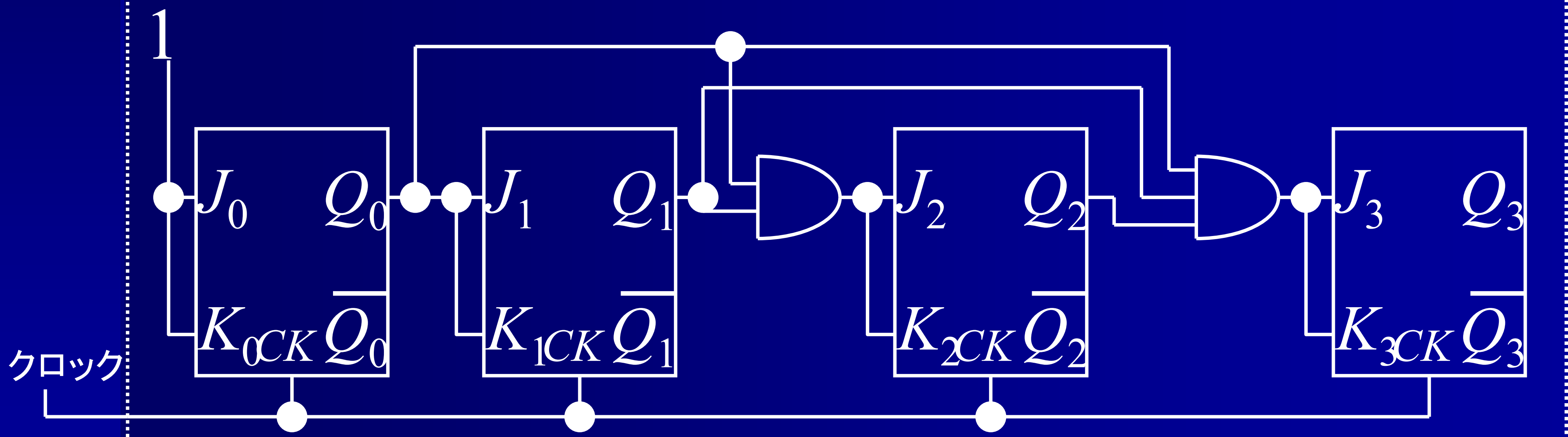


0ビット目のJKFF 1ビット目のJKFF iビット目のJKFF

同期式16進分周器

$$J_3, K_3 = Q_2 \cdot Q_1 \cdot Q_0 \quad J_2, K_2 = Q_1 \cdot Q_0$$

$$J_1, K_1 = Q_0 \qquad J_0, K_0 = 1$$

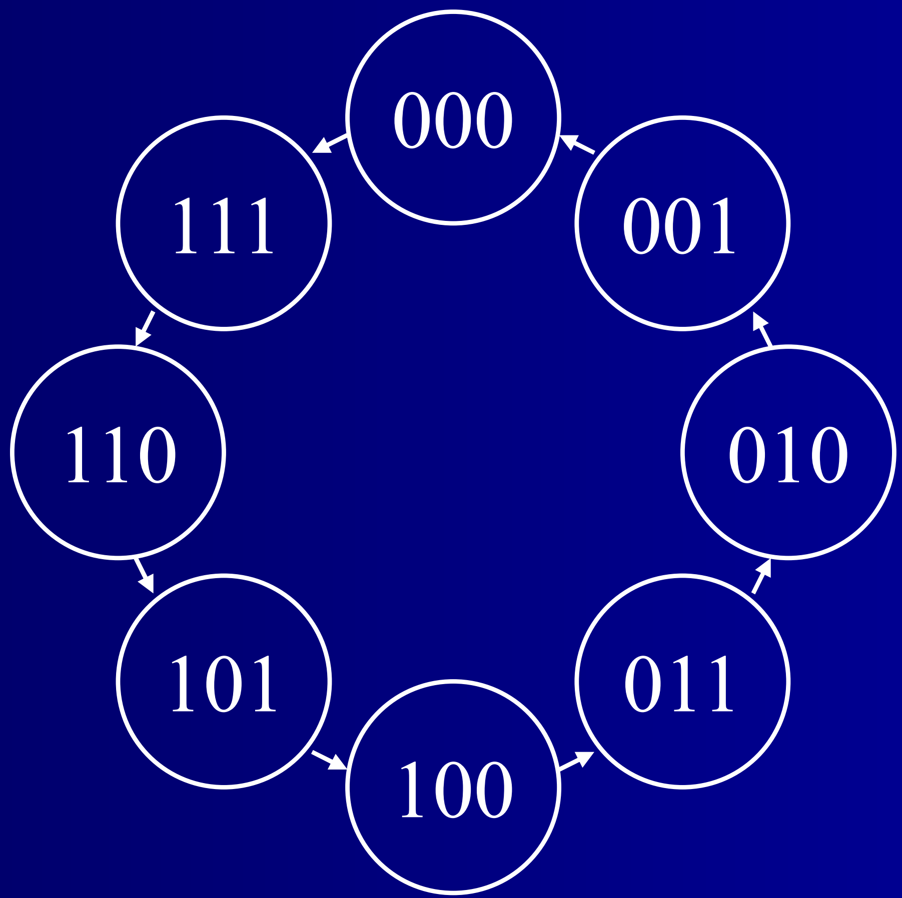


様々なカウンタ

- n ビット2進カウンタ (2^{n}進カウンタ)
- n ビット2進減算カウンタ
- グレイコードカウンタ
- ジョンソンカウンタ
- リングカウンタ
- BCDカウンタ (10進カウンタ)

2^n進減算カウンタ

- 1 ずつ減らす遷移をする回路



同期式2^n進減算カウンタ

加算カウンタ

$$J_i, K_i = Q_{i-1} \cdot Q_{i-2} \cdot \ldots \cdot Q_1 \cdot Q_0 \quad (2 \le i \le n)$$

$$J_1, K_1 = Q_0$$

$$J_0, K_0 = 1$$

減算カウンタ

$$J_i, K_i = \overline{Q_{i-1}} \cdot \overline{Q_{i-2}} \cdot \ldots \cdot \overline{Q_1} \cdot \overline{Q_0} \quad (2 \le i \le n)$$

$$J_1, K_1 = \overline{Q_0}$$

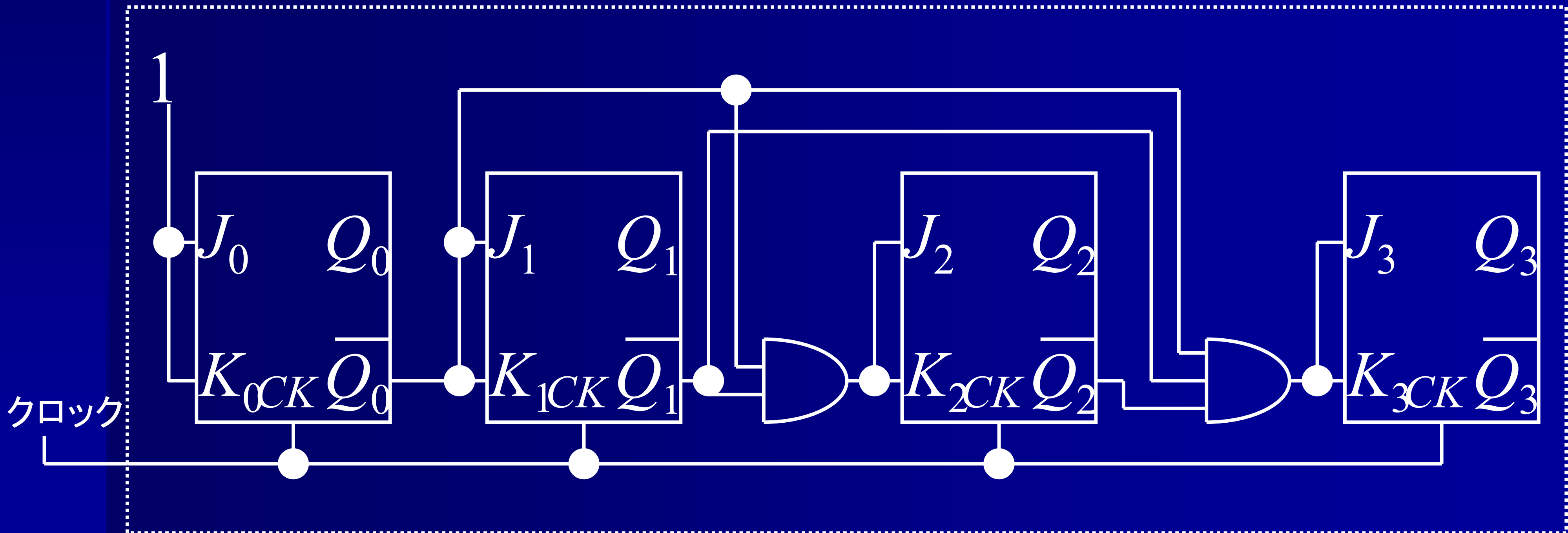
$$J_0, K_0 = 1$$

加算カウンタの Q の代わりに $\overline{Q}$ を用いると減算カウンタになる

同期式16進減算カウンタ

$$J_3, K_3 = \overline{Q_2} \cdot \overline{Q_1} \cdot \overline{Q_0} \qquad J_2, K_2 = \overline{Q_1} \cdot \overline{Q_0}$$

$$J_1, K_1 = \overline{Q_0} \qquad J_0, K_0 = 1$$



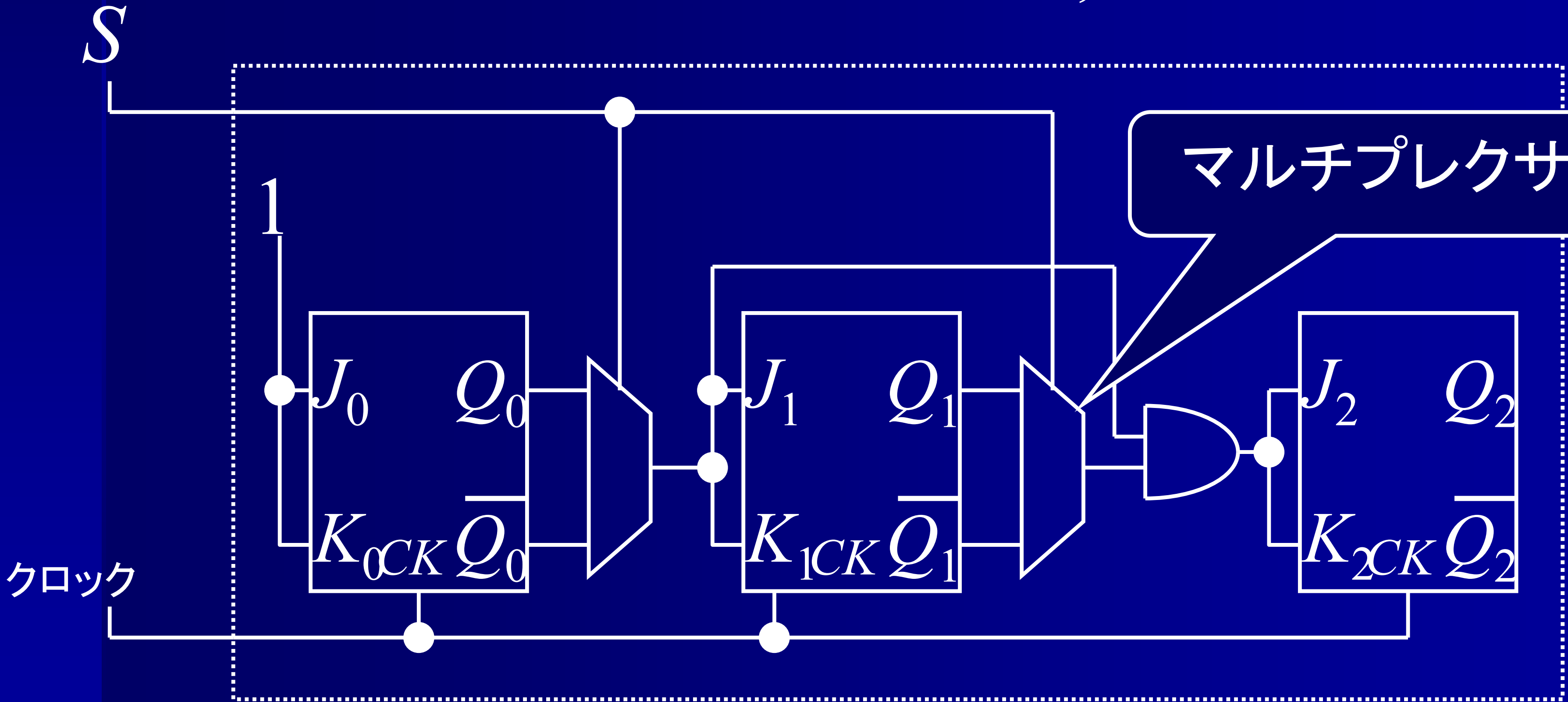
同期式2^n進加減算カウンタ

$$J_i, K_i = \overline{S}\cdot(Q_{i-1}\cdot\ldots\cdot Q_0) + S\cdot(\overline{Q_{i-1}}\cdot\ldots\cdot\overline{Q_0})\ (2\le i\le n)$$

$$J_1, K_1 = \overline{S}\cdot Q_0 + S\cdot\overline{Q_0}$$

$$J_0, K_0 = 1$$

$S=0$ のとき加算, $S=1$ のとき減算



グレイコード

- カルノー図で用いる2進数列
  - 隣り合う数は1ビットのみ異なる
- n ビット2^n状態の完全指定論理関数
  - 2ビット 00,01,11,10
  - 3ビット 000,001,011,010,110,111,101,100
  - 4ビット 0000,0001,0011,0010,0110,0111,0101,0110, 1100,1101,1111,1110,1010,1011,1001,1000

グレイコードの作り方

2ビット
↓
3ビット

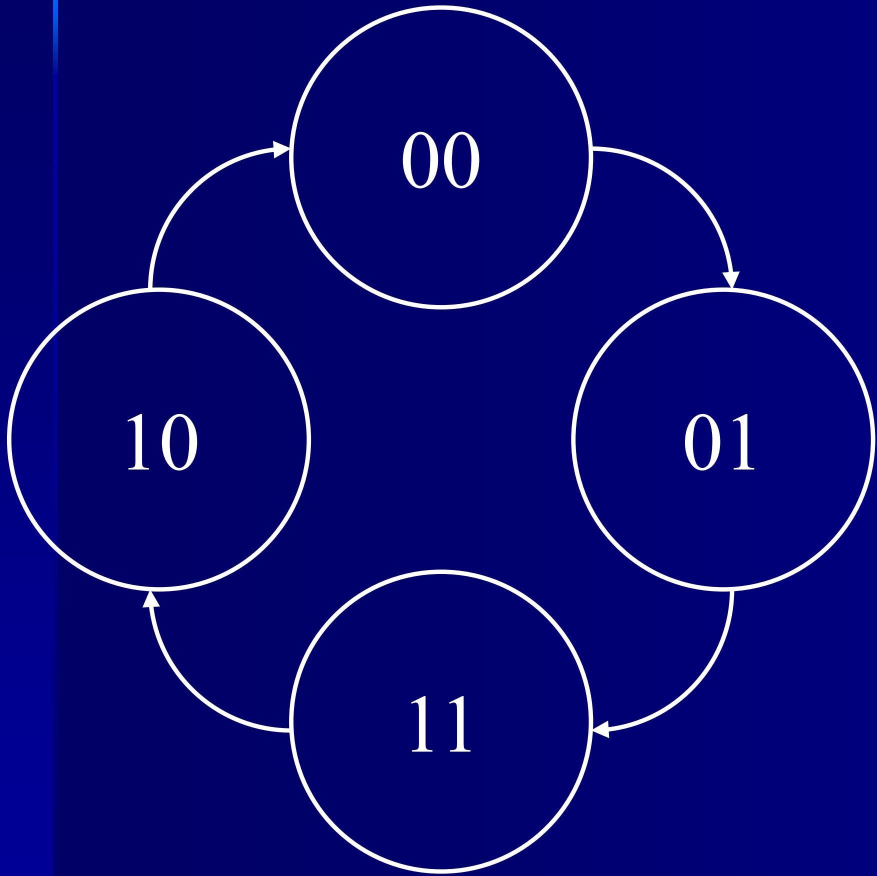
2ビット				2ビット逆順			
00	01	11	10	10	11	01	00
先頭に0				先頭に1			
000	001	011	010	110	111	101	100

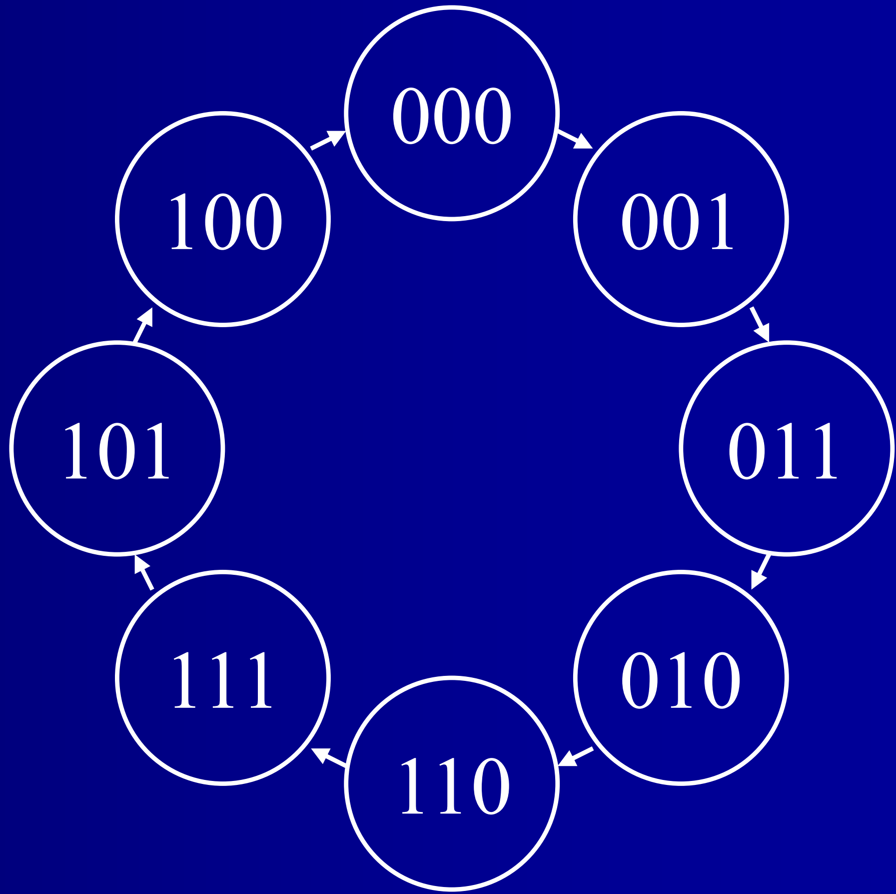
3ビット
↓
4ビット

3ビット							
000	001	011	010	110	111	101	100
先頭に0							
0000	0001	0011	0010	0110	0111	0101	0100

3ビット逆順							
100	101	111	110	010	011	001	000
先頭に1							
1100	1101	1111	1110	1010	1011	1001	1000

グレイコードカウンタ





3.状態遷移表を作る
4.拡大入力要求表を作る

Q_1	Q_0	Q_1^+	Q_0^+	J_1K_1	J_0K_0
0	0	0	1	0 -	1 -
0	1	1	1	1 -	- 0
1	0	0	0	- 1	0 -
1	1	1	0	- 0	- 1

5. FFの入力条件式を求める

Q_1	Q_0	J_1K_1	J_0K_0
0	0	0 -	1 -
0	1	1 -	- 0
1	0	- 1	0 -
1	1	- 0	- 1

$J_1\ K_1$

Q_0 \ Q_1	0	1
0	0 -	- 1
1	1 -	- 0

$J_0\ K_0$

Q_0 \ Q_1	0	1
0	1 -	0 -
1	- 0	- 1

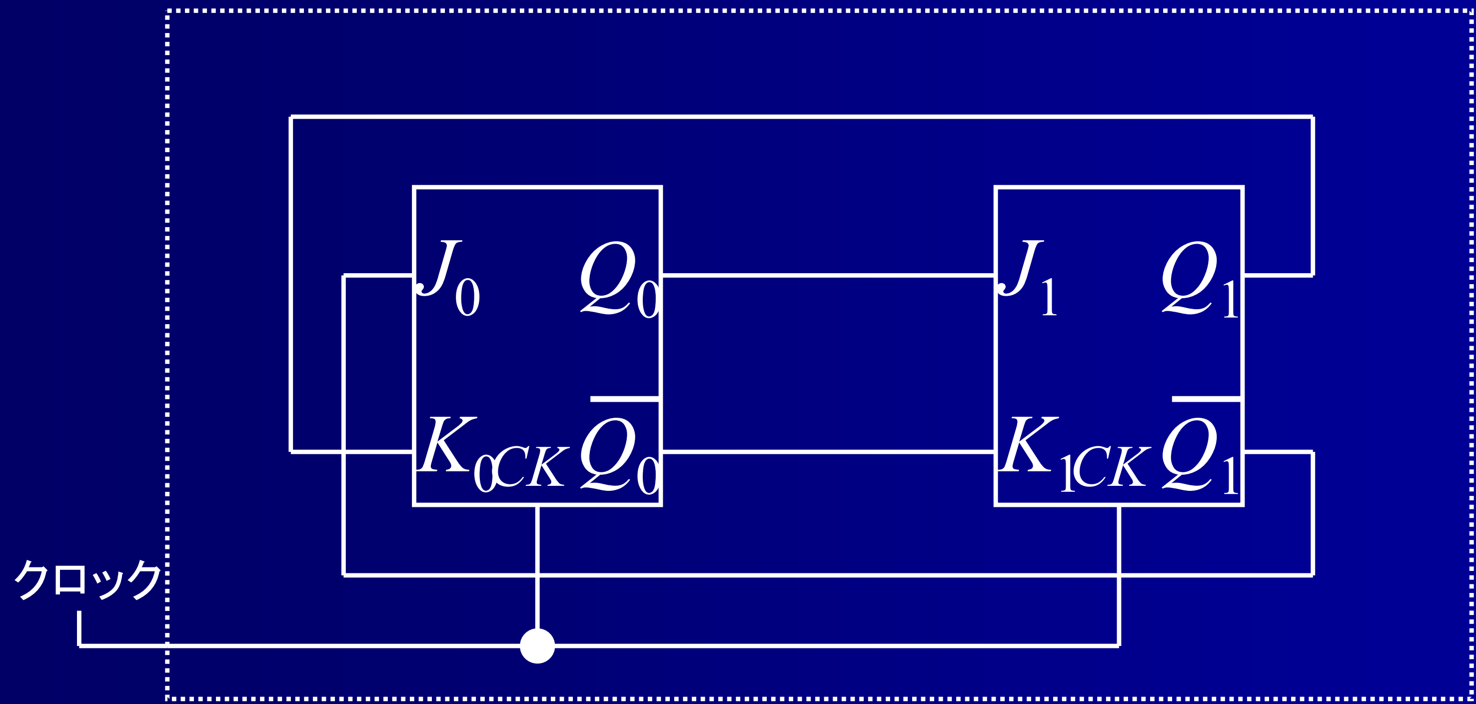
$$J_1 = Q_0 \quad K_1 = \overline{Q_0}$$

$$J_0 = \overline{Q_1} \quad K_0 = Q_1$$

7.回路図を描く

$$J_1 = Q_0 \quad K_1 = \overline{Q_0}$$

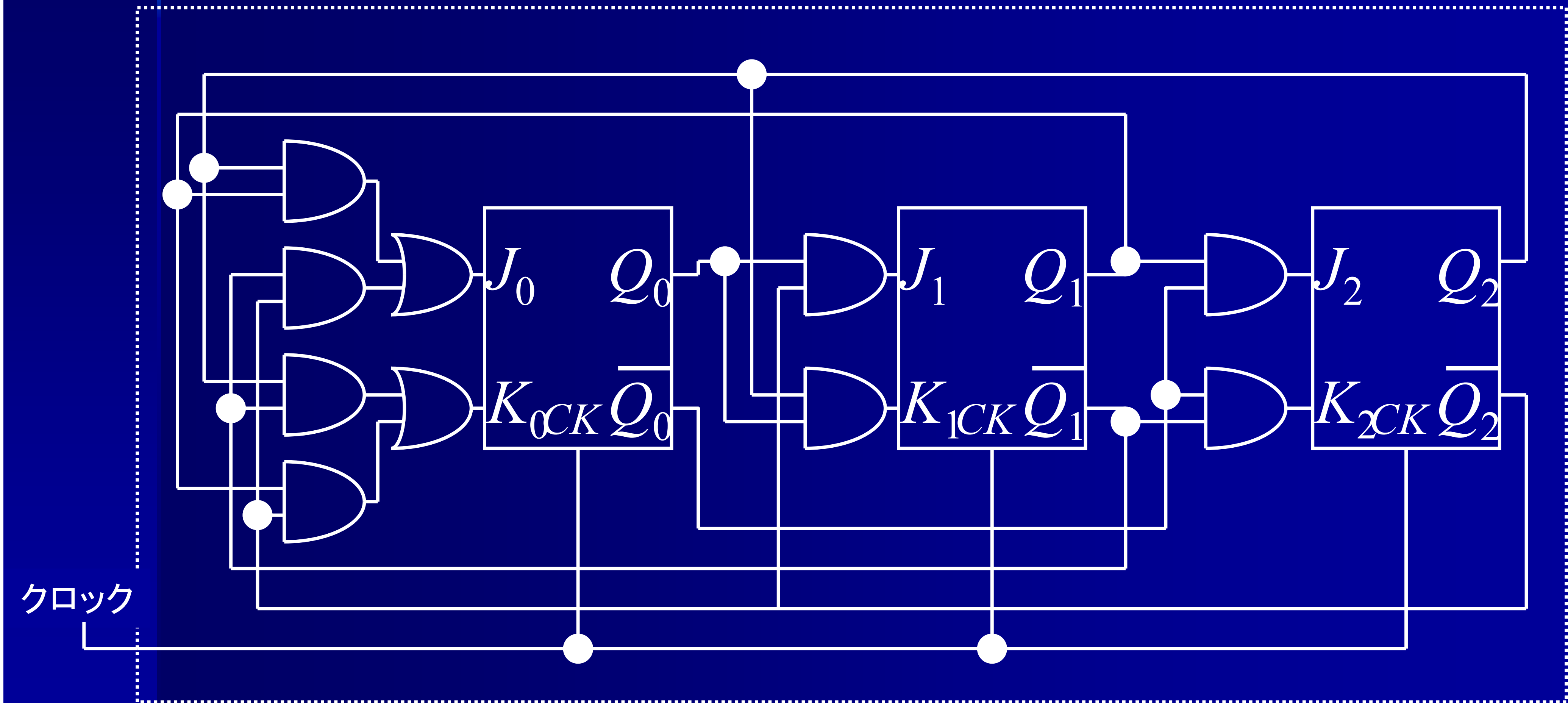
$$J_0 = \overline{Q_1} \quad K_0 = Q_1$$



3ビットグレイコードカウンタ

$$J_2 = Q_1 \cdot Q_0 \quad J_1 = \overline{Q_2} \cdot Q_0 \quad J_0 = \overline{Q_2} \cdot \overline{Q_1} + Q_2 \cdot Q_1$$

$$K_2 = \overline{Q_1} \cdot \overline{Q_0} \quad K_1 = Q_2 \cdot Q_0 \quad K_0 = \overline{Q_2} \cdot Q_1 + Q_2 \cdot \overline{Q_1}$$



2^n進→グレイコード変換

- 2^n進からグレイコードへの変換

$$\boldsymbol{b} = (b_{n\text{-}1}, b_{n\text{-}2}, \ldots, b_1, b_0)$$

$$\boldsymbol{g} = (g_{n\text{-}1}, g_{n\text{-}2}, \ldots, g_1, g_0)$$

$$g_{n\text{-}1} = b_{n\text{-}1}$$

$$g_i \quad = b_{i+1} \oplus b_i \ (0 \leqq i < n\text{-}1)$$

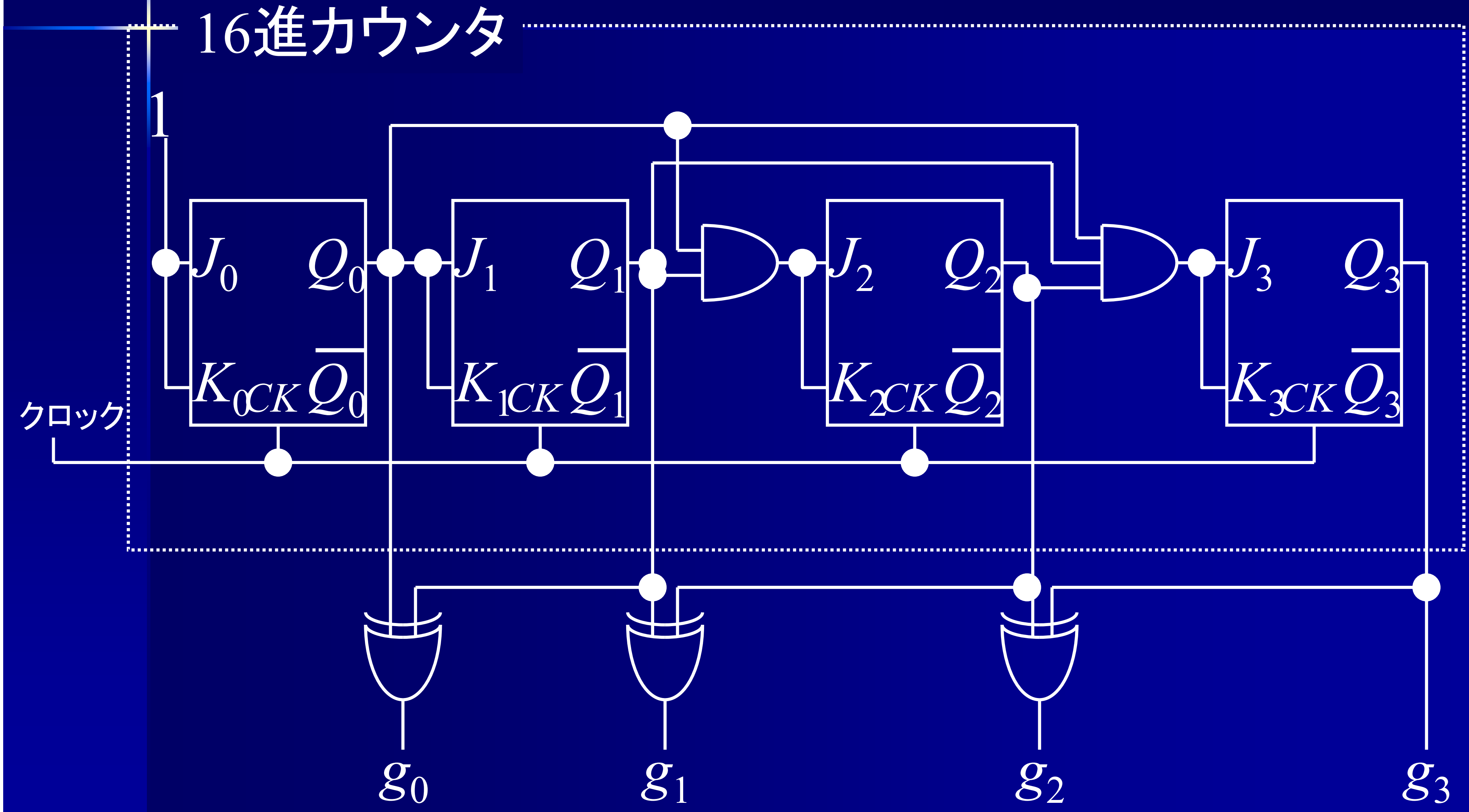
$\boldsymbol{b} = 01110$ (14)

$\boldsymbol{g} = 01001$

グレイコードカウンタは直接作るより2^n進カウンタから変換した方が簡単

(※) グレイコード→2^n進は難しい

4ビットグレイコードカウンタ



グレイコードカウンタの利点

1ビット　2ビット　1ビット　3ビット　1ビット　2ビット　1ビット　3ビット

000→001→010→011→100→101→110→111→000

1ビット　1ビット　1ビット　1ビット　1ビット　1ビット　1ビット　1ビット

000→001→011→010→110→111→101→100→000

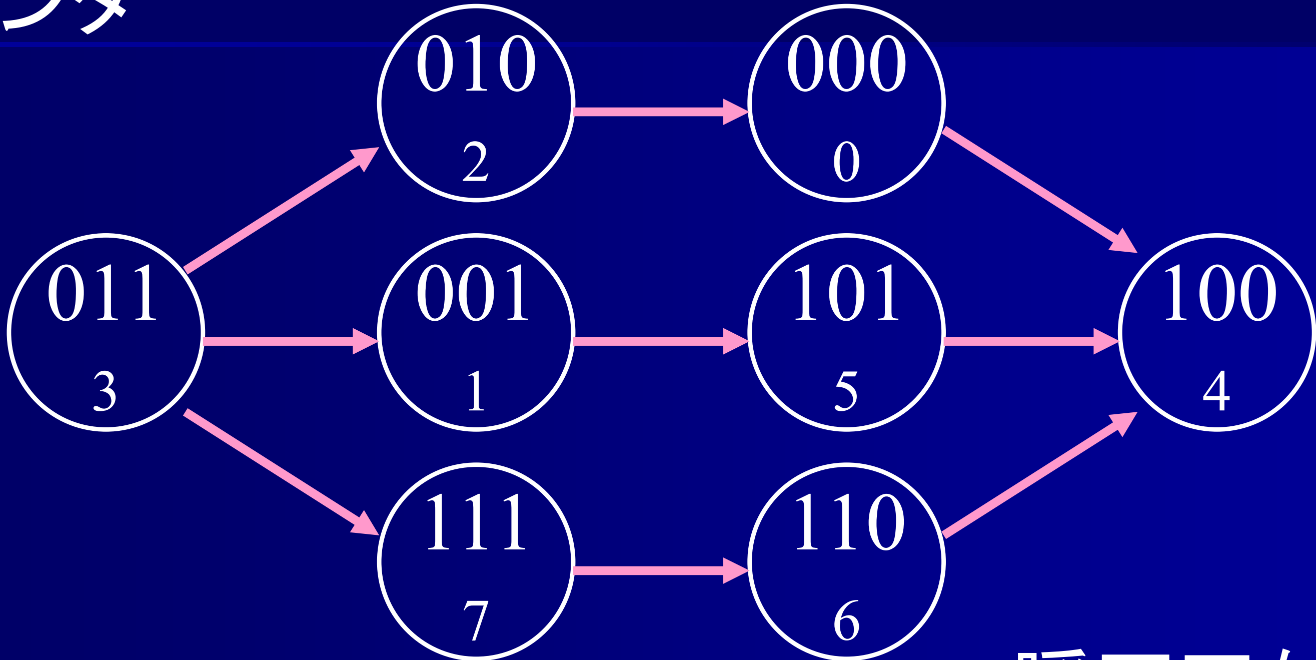
増加時に変化するビットは 1 ビットのみ

↓

変化途中に不正な値が現れない

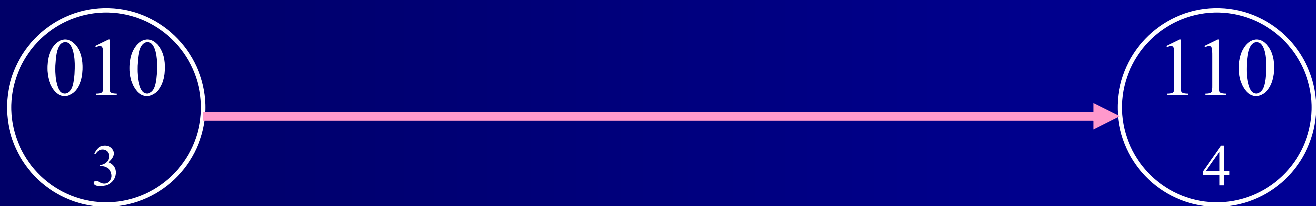
グレイコードカウンタの利点

8進カウンタ



一瞬不正な値が出る

グレイコードカウンタ



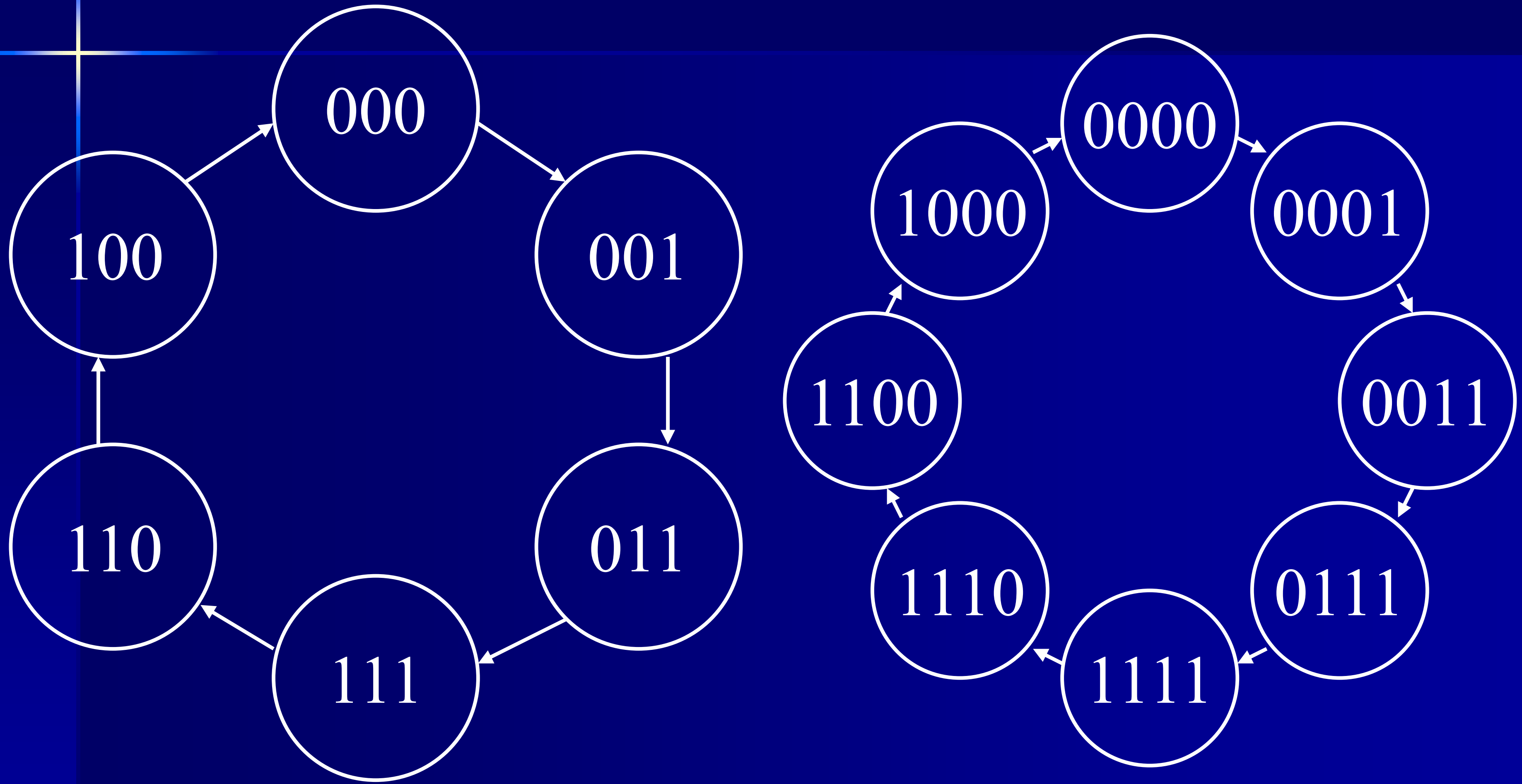
不正な値は出ない

ジョンソン(Johnson)カウンタ

- nビット$2n$状態の不完全指定論理関数
  - 2ビット 00,01,11,10
  - 3ビット 000,001,011,111,110,100
  - 4ビット 0000,0001,0011,0111,1111,1110,1100,1000

右から順に0を1に変える								
0000	0001	0011	0111	1111	1110	1100	1000	0000
				右から順に1を0に変える				

ジョンソンカウンタ



010,101はドントケア

0010,0100,0101,0110,1001,
1010,1011,1101はドントケア

3.状態遷移表を作る
4.拡大入力要求表を作る

$Q_2 Q_1 Q_0$	$Q_2^+ Q_1^+ Q_0^+$	$D_2 D_1 D_0$
0 0 0	0 0 1	0 0 1
0 0 1	0 1 1	0 1 1
0 1 0	-	-
0 1 1	1 1 1	1 1 1
1 0 0	0 0 0	0 0 0
1 0 1	-	-
1 1 0	1 0 0	1 0 0
1 1 1	1 1 0	1 1 0

5. FFの入力条件式を求める

Q_2	Q_1	Q_0	D_2	D_1	D_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	-	-	-
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	-	-	-
1	1	0	1	0	0
1	1	1	1	1	0

$$D_2 = Q_1 \quad D_1 = Q_0 \quad D_0 = \overline{Q_2}$$

D_2

Q_2 \ Q_1Q_0	00	01	11	10
0			1	-
1		-	1	1

D_1

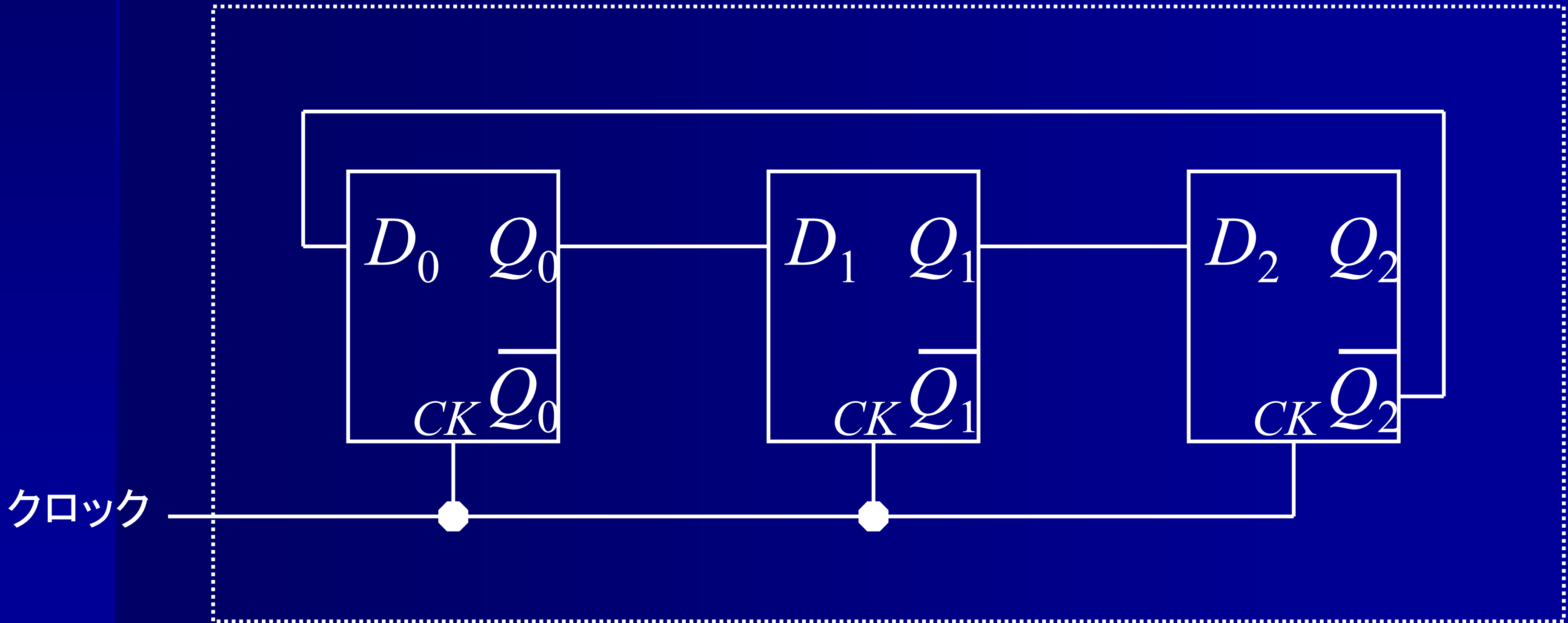
Q_2 \ Q_1Q_0	00	01	11	10
0		1	1	-
1		-	1	

D_0

Q_2 \ Q_1Q_0	00	01	11	10
0	1	1	1	-
1		-		

7.回路図を描く

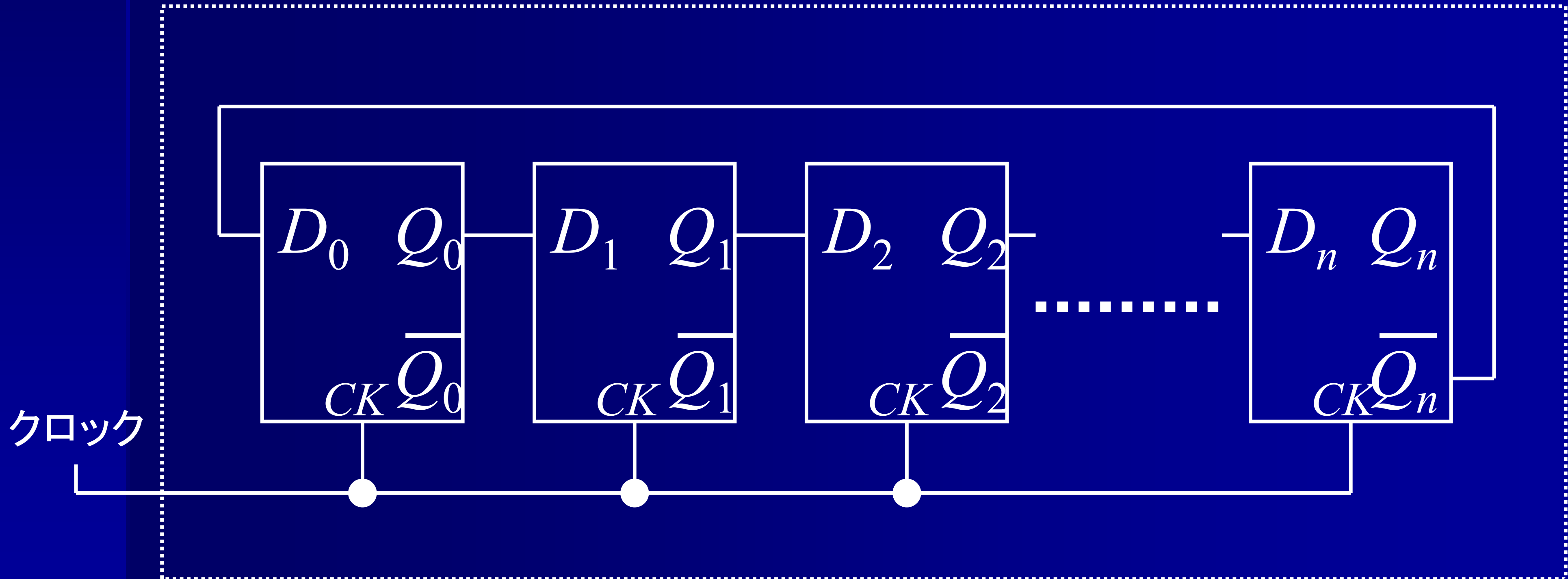
$$D_2 = Q_1 \quad D_1 = Q_0 \quad D_0 = \overline{Q_2}$$



n ビットジョンソンカウンタ

$$D_i = Q_{i-1} \quad (1 \le i \le n)$$

$$D_0 = \overline{Q_n}$$

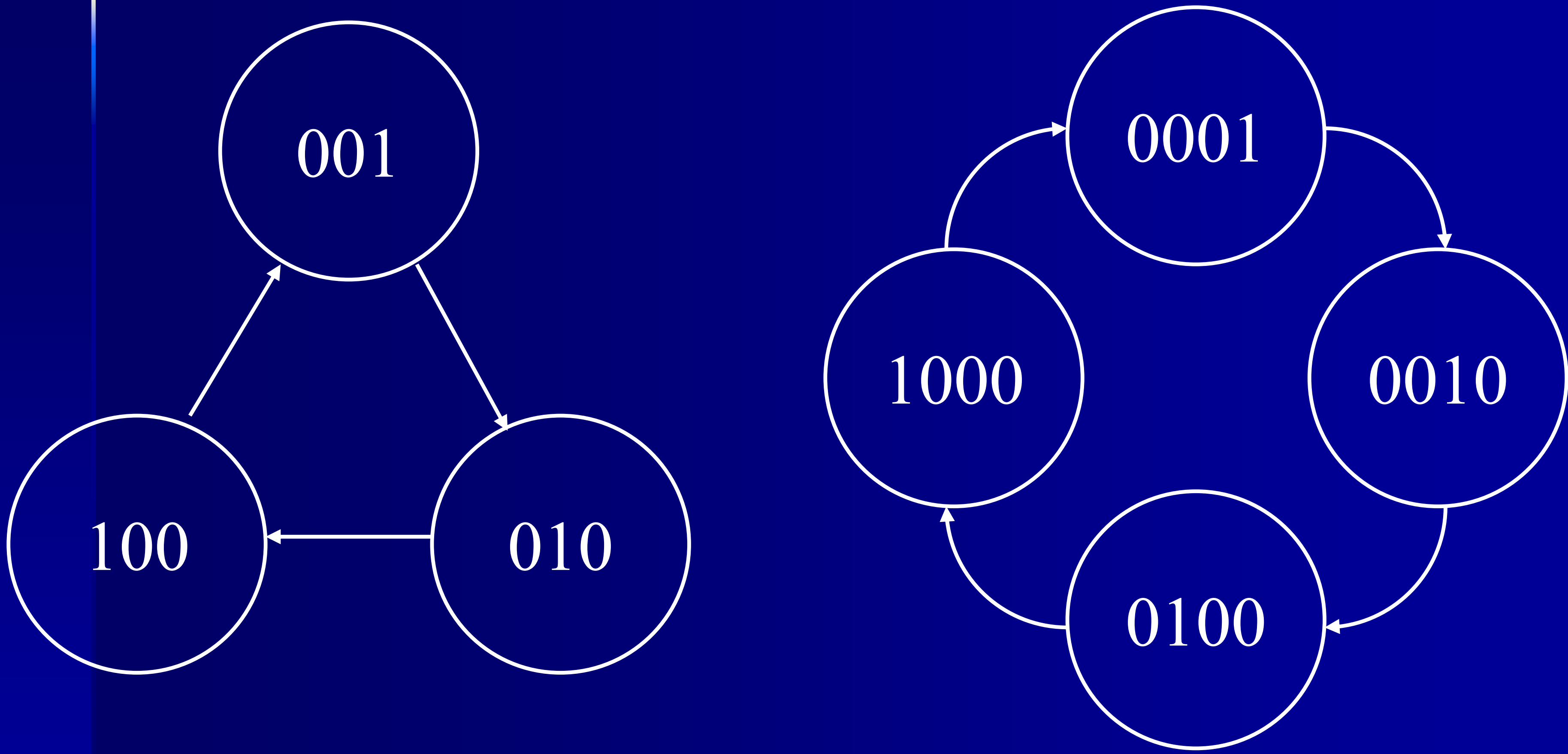


リング(ring)カウンタ

- *n*ビット*n*状態の不完全指定論理関数
  - 2ビット 01,10
  - 3ビット 001,010,100
  - 4ビット 0001,0010,0100,1000

000001	000010	000100	001000	010000	100000
1を左にずらす					

リングカウンタ



3.状態遷移表を作る
4.拡大入力要求表を作る

$Q_2 Q_1 Q_0$	$Q_2^+ Q_1^+ Q_0^+$	$D_2 D_1 D_0$
0 0 0	-	-
0 0 1	0 1 0	0 1 0
0 1 0	1 0 0	1 0 0
0 1 1	-	-
1 0 0	0 0 1	0 0 1
1 0 1	-	-
1 1 0	-	-
1 1 1	-	-

5. FFの入力条件式を求める

Q_2	Q_1	Q_0	D_2	D_1	D_0
0	0	0	-	-	-
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	-	-	-
1	0	0	0	0	1
1	0	1	-	-	-
1	1	0	-	-	-
1	1	1	-	-	-

D_2

Q_2 \ Q_1Q_0	00	01	11	10
0	-		-	1
1		-	-	-

D_1

Q_2 \ Q_1Q_0	00	01	11	10
0	-	1	-	
1		-	-	-

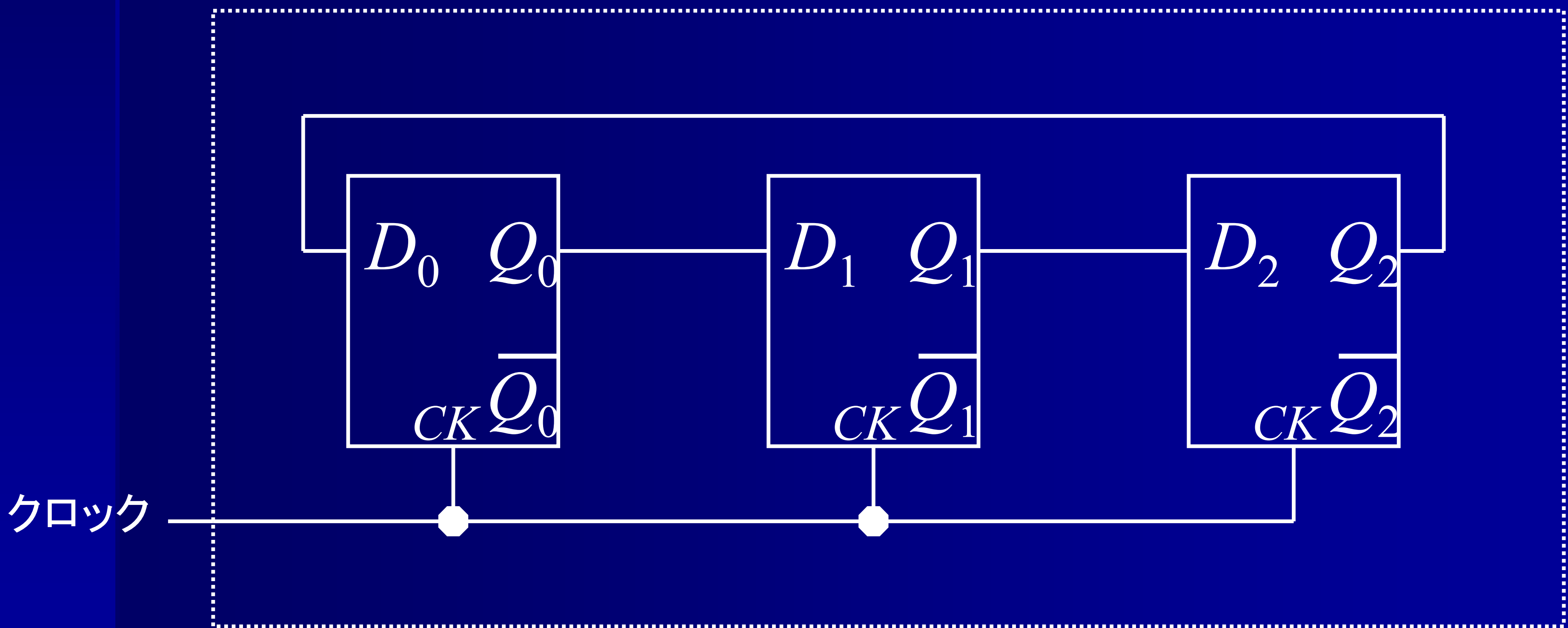
D_0

Q_2 \ Q_1Q_0	00	01	11	10
0	-		-	
1	1	-	-	-

$$D_2 = Q_1 \quad D_1 = Q_0 \quad D_0 = Q_2$$

7.回路図を描く

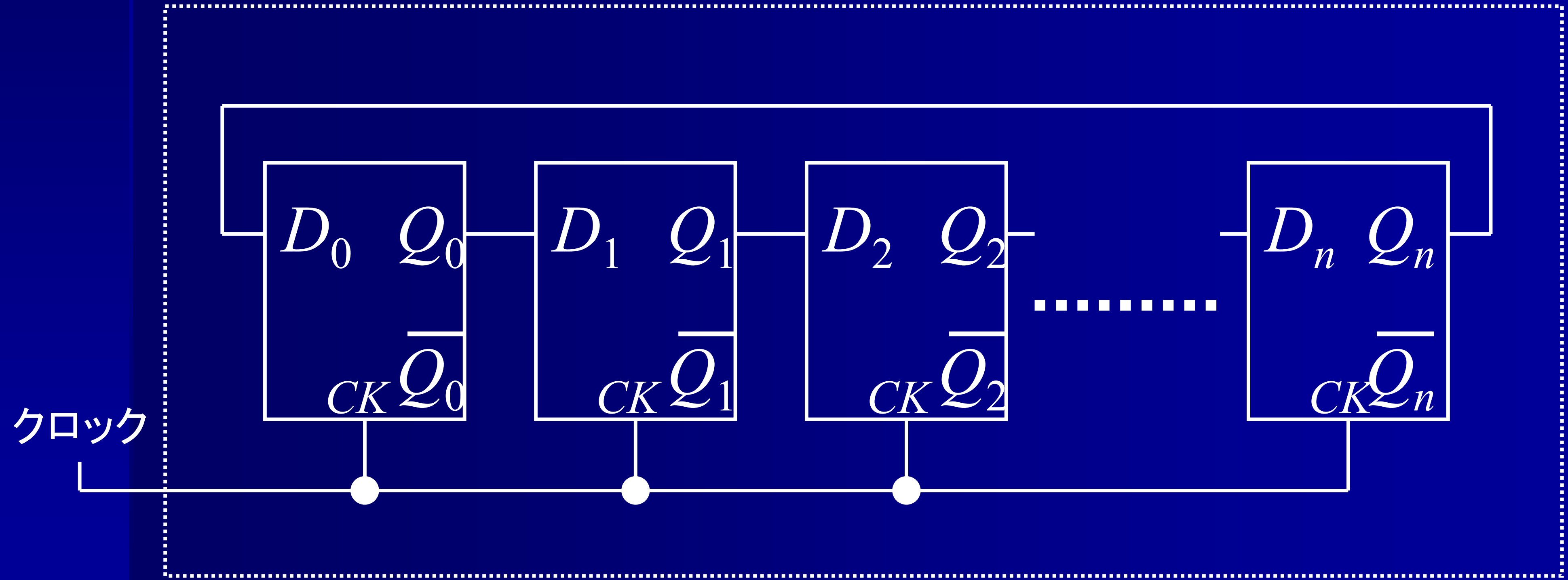
$$D_2 = Q_1 \quad D_1 = Q_0 \quad D_0 = Q_2$$



n ビットリングカウンタ

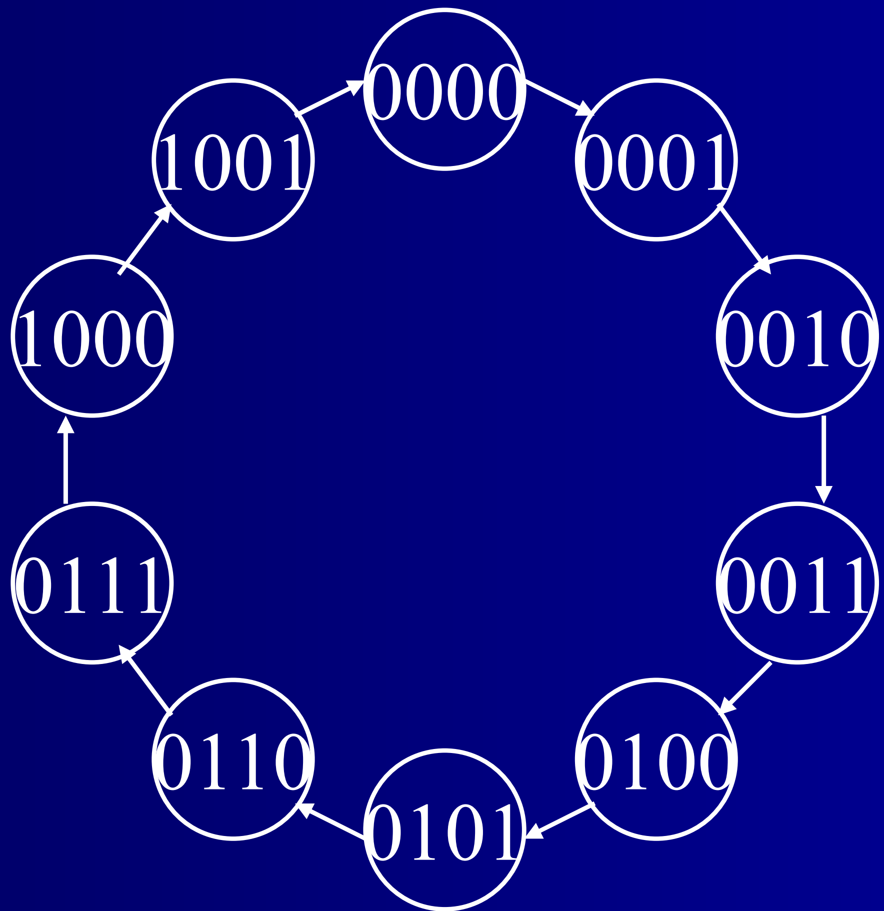
$$D_i = Q_{i-1} \quad (1 \le i \le n)$$

$$D_0 = Q_n$$



BCD (Binary Coded Decimal)カウンタ

- 2進コード化10進数カウンタ
- 4ビット10状態の不完全指定論理関数



1010～1111は
ドントケア

3.状態遷移表,4.拡大入力表を作る

$Q_3\,Q_2\,Q_1\,Q_0$	$Q_3^+Q_2^+Q_1^+Q_0^+$ $D_3\,D_2\,D_1\,D_0$	$Q_3\,Q_2\,Q_1\,Q_0$	$Q_3^+Q_2^+Q_1^+Q_0^+$ $D_3\,D_2\,D_1\,D_0$
0 0 0 0	0 0 0 1	1 0 0 0	1 0 0 1
0 0 0 1	0 0 1 0	1 0 0 1	0 0 0 0
0 0 1 0	0 0 1 1	1 0 1 0	-
0 0 1 1	0 1 0 0	1 0 1 1	-
0 1 0 0	0 1 0 1	1 1 0 0	-
0 1 0 1	0 1 1 0	1 1 0 1	-
0 1 1 0	0 1 1 1	1 1 1 0	-
0 1 1 1	1 0 0 0	1 1 1 1	-

5. FFの入力条件式を求める

D_3

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00				
01			1	
11	-	-	-	-
10	1		-	-

D_2

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00			1	
01	1	1		1
11	-	-	-	-
10			-	-

D_1

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00		1		1
01		1		1
11	-	-	-	-
10			-	-

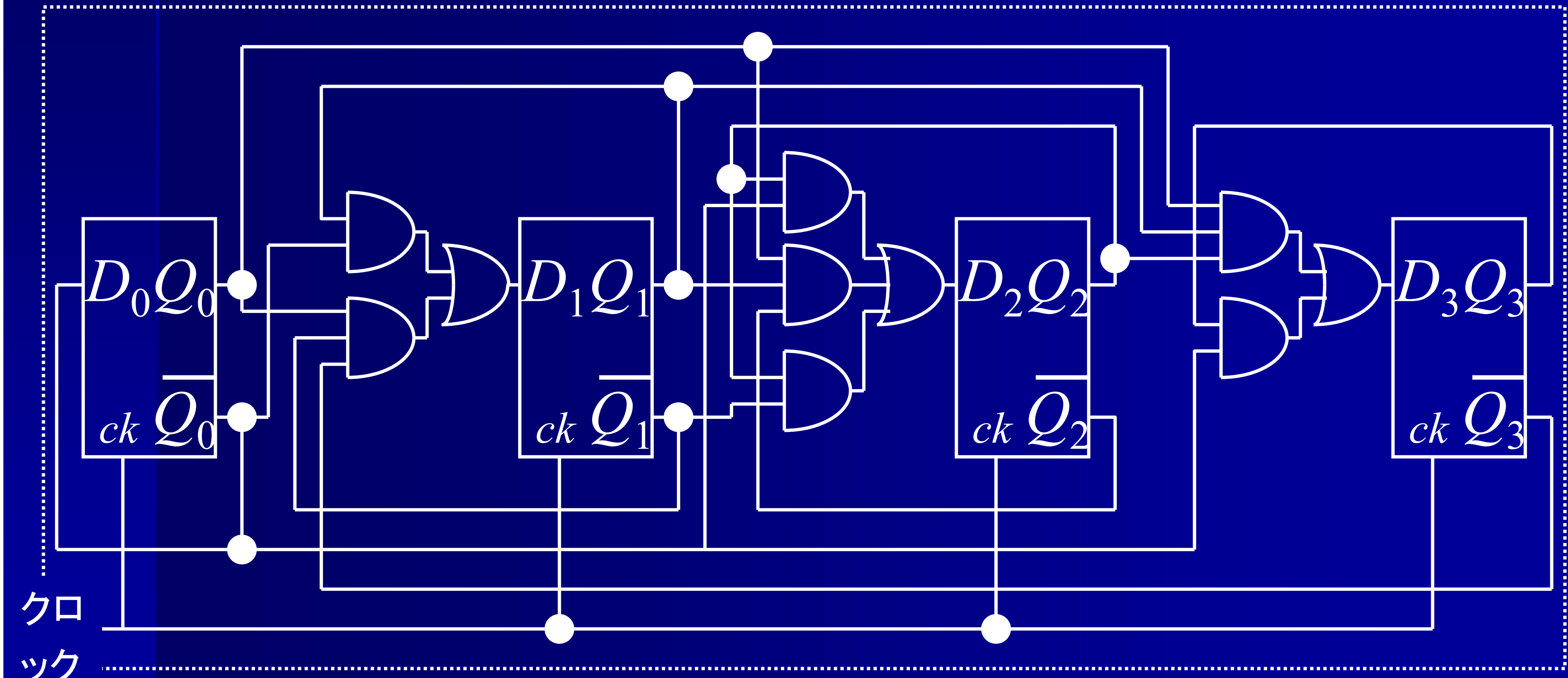
D_0

Q_3Q_2 \ Q_1Q_0	00	01	11	10
00	1			1
01	1			1
11	-	-	-	-
10	1		-	-

7.回路図を描く

$$D_3 = Q_3\overline{Q_0} + Q_2Q_1Q_0 \quad D_2 = Q_2\overline{Q_1} + Q_2\overline{Q_0} + \overline{Q_2}Q_1Q_0$$

$$D_1 = \overline{Q_3}\,\overline{Q_1}Q_0 + Q_1\overline{Q_0} \quad D_0 = \overline{Q_0}$$



4.拡大入力表を作る(JKFF)

$Q_3\, Q_2\, Q_1\, Q_0$	$J_3K_3\ J_2K_2\ J_1K_1\ J_0K_0$	$Q_3\, Q_2\, Q_1\, Q_0$	$J_3K_3\ J_2K_2\ J_1K_1\ J_0K_0$
0 0 0 0	0- 0- 0- 1-	1 0 0 0	-0 0- 0- 1-
0 0 0 1	0- 0- 1- -1	1 0 0 1	-1 0- 0- -1
0 0 1 0	0- 0- -0 1-	1 0 1 0	-
0 0 1 1	0- 1- -1 -1	1 0 1 1	-
0 1 0 0	0- -0 0- 1-	1 1 0 0	-
0 1 0 1	0- -0 1- -1	1 1 0 1	-
0 1 1 0	0- -0 -0 1-	1 1 1 0	-
0 1 1 1	1- -1 -1 -1	1 1 1 1	-

5. FFの入力条件式を求める

J_3K_3

Q_1Q_0 \ Q_3Q_2	00	01	11	10
00	0-	0-	0-	0-
01	0-	0-	1-	0-
11	-	-	-	-
10	-0	-1	-	-

J_2K_2

Q_1Q_0 \ Q_3Q_2	00	01	11	10
00	0-	0-	1-	0-
01	-0	-0	-1	-0
11	-	-	-	-
10	0-	0-	-	-

J_1K_1

Q_1Q_0 \ Q_3Q_2	00	01	11	10
00	0-	1-	-1	-0
01	0-	1-	-1	-0
11	-	-	-	-
10	0-	0-	-	-

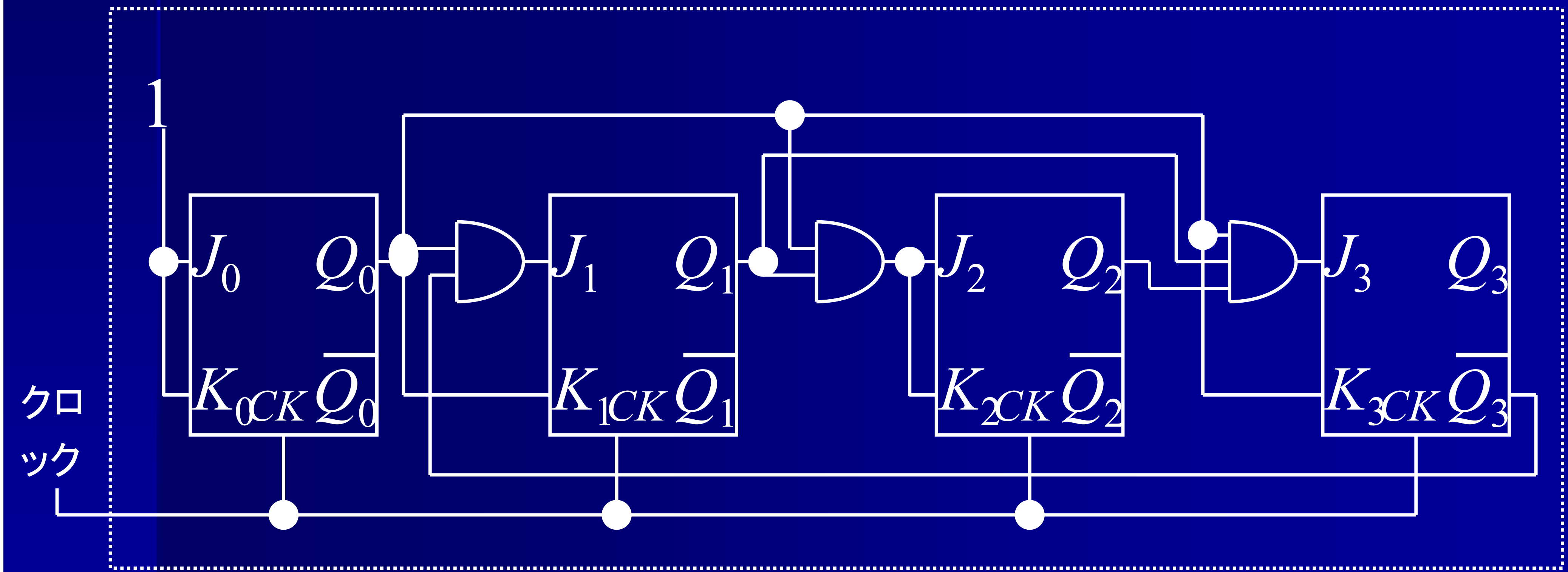
J_0K_0

Q_1Q_0 \ Q_3Q_2	00	01	11	10
00	1-	-1	-1	1-
01	1-	-1	-1	1-
11	-	-	-	-
10	1-	-1	-	-

7.回路図を描く

$$J_3 = Q_2 \cdot Q_1 \cdot Q_0 \quad J_2 = Q_1 \cdot Q_0 \quad J_1 = \overline{Q_3} \cdot Q_0 \quad J_0 = 1$$

$$K_3 = Q_0 \quad K_2 = Q_1 \cdot Q_0 \quad K_1 = Q_0 \quad K_0 = 1$$

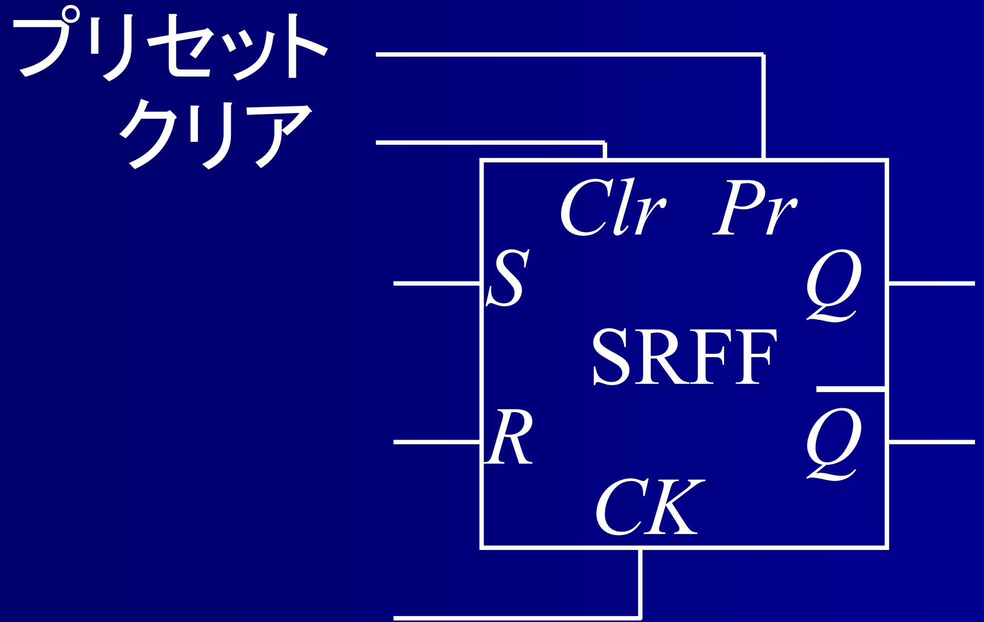


プリセット,クリア付フリップフロップ

- プリセット,クリア付フリップフロップ
  - 通常の入力(SR,D,T,JK)に加え、Preset信号PrとClear信号Clrを入力

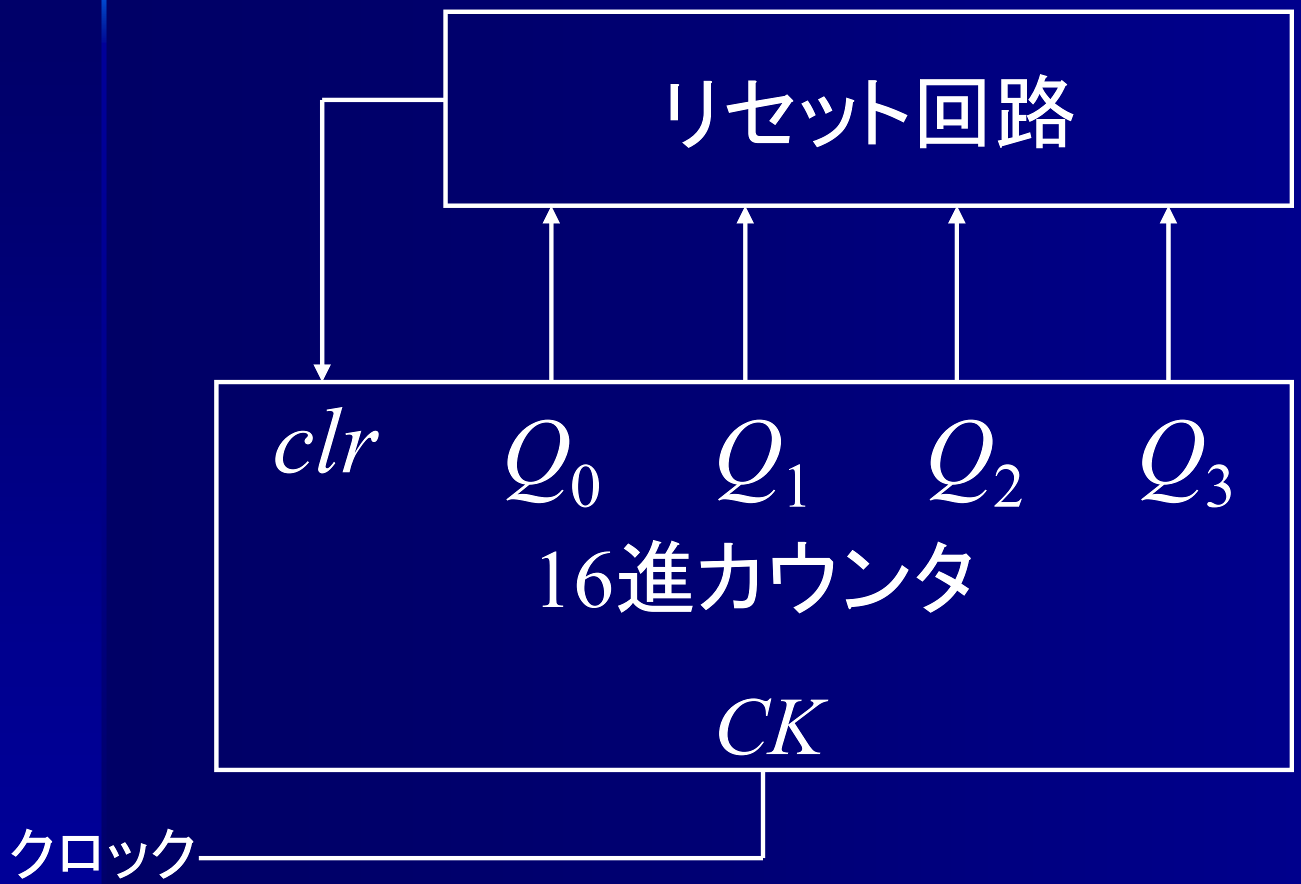
Preset信号でクロックに関係無く1にセット

Clear信号でクロックに関係無く0にリセット



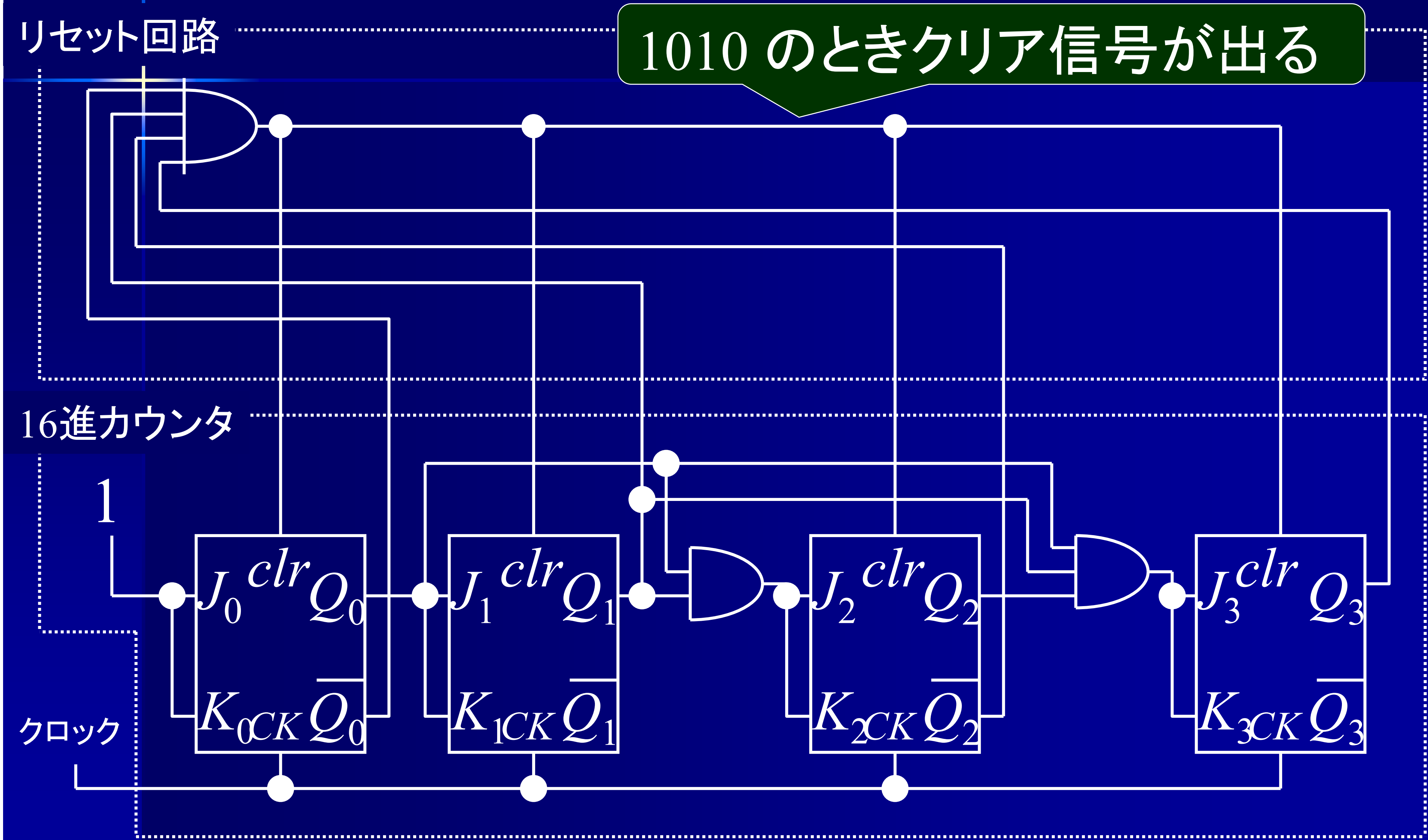
クリア信号付FFを用いたBCDカウンタ

- 16進カウンタ+リセット回路

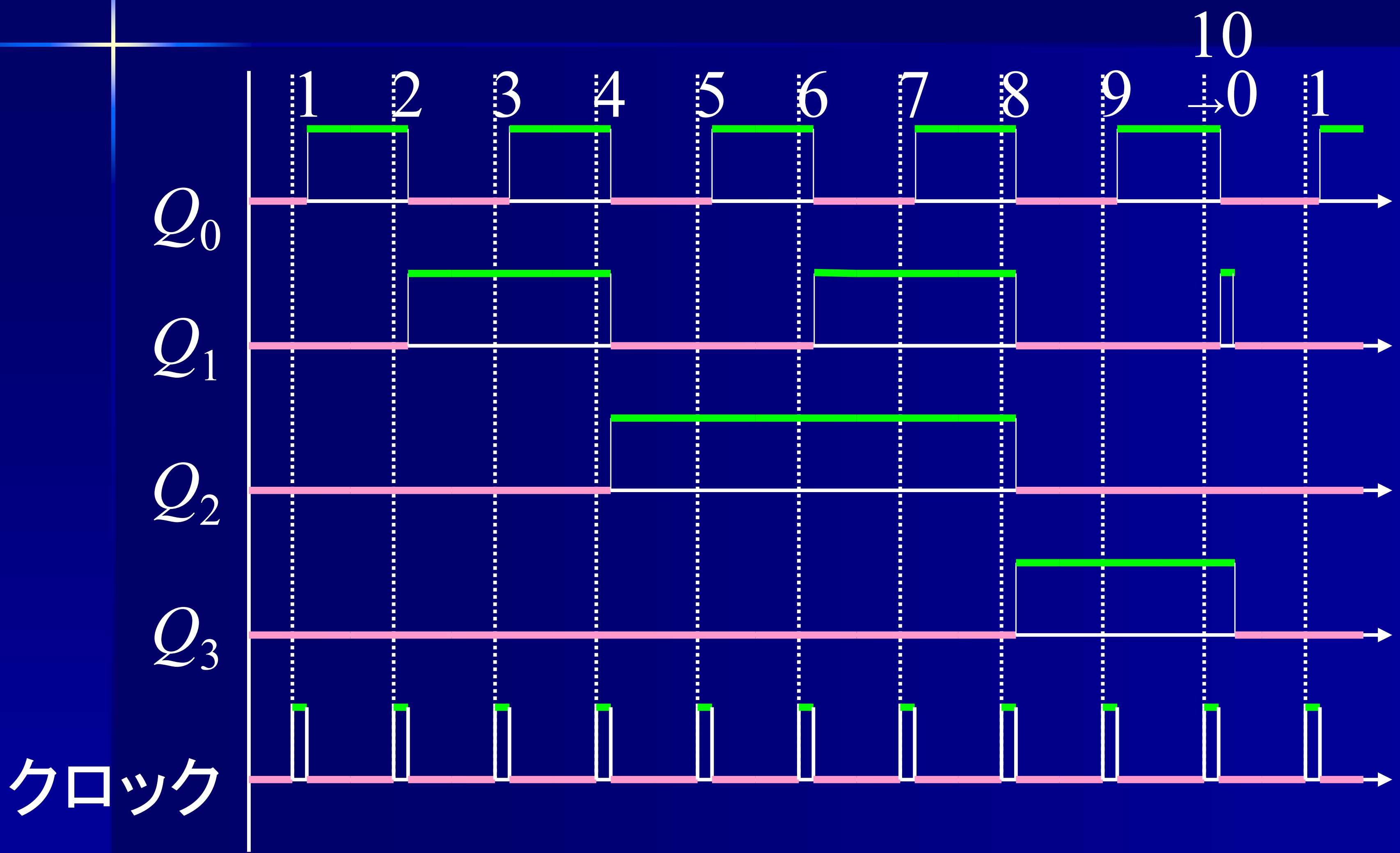


1010になったときにクリア信号を出す

クリア信号付FFを用いたBCDカウンタ



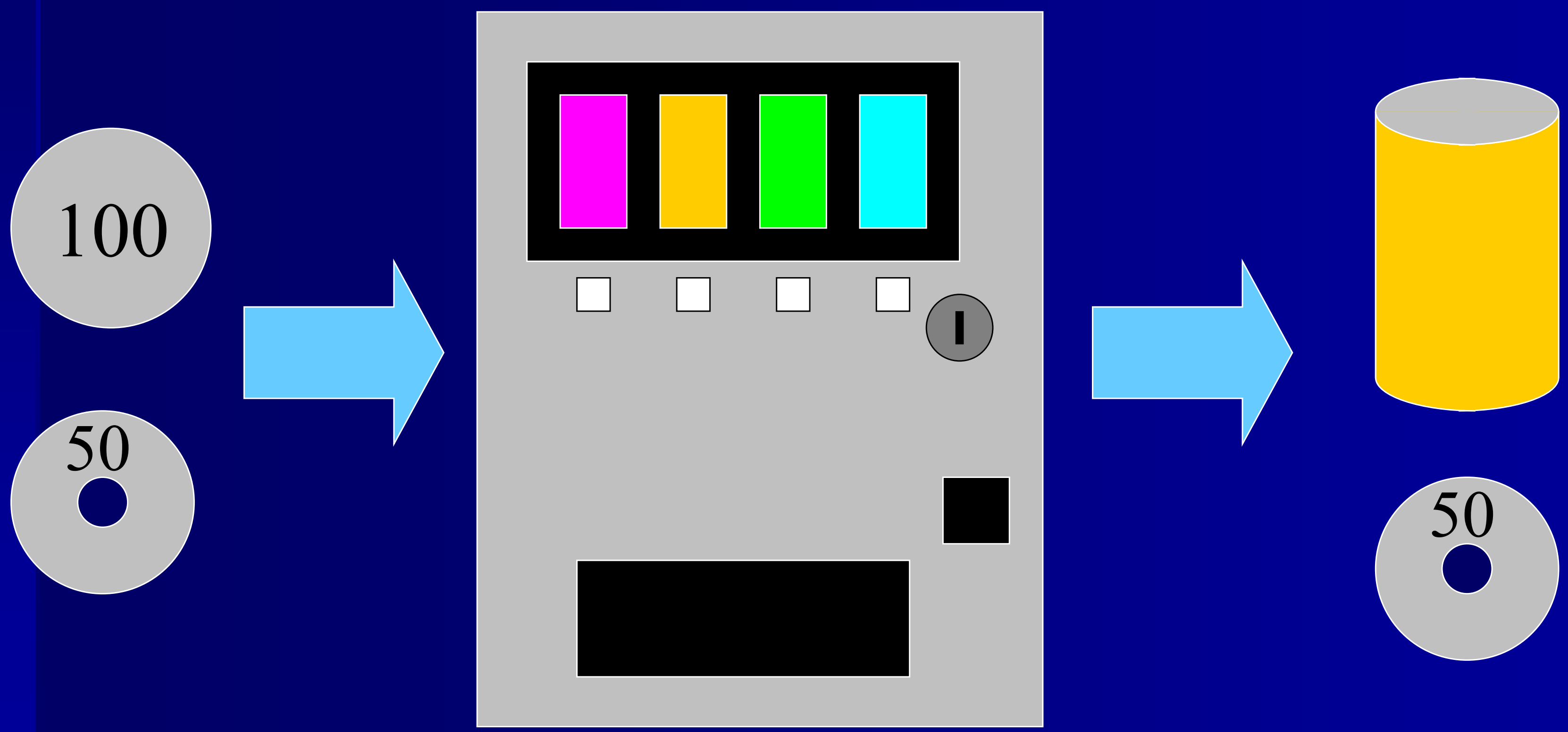
クリア信号付FFによる BCDカウンタの遷移



一瞬 1010 になった後 0000 にクリアされる

自動販売機の設計

- 100円投入されると商品を出す
  - コイン投入口は100円用と50円用の2つ
  - お釣り返還口は50円用が1つ



自動販売機の動作

投入金額	自動販売機の動作
0円	次の投入待ち
50円	次の投入待ち
100円	商品を出して0円に
150円	商品とお釣り50円を出して0円に
200円	商品とお釣り50円を出して50円に

必要な状態は「0円が投入された」
「50円が投入された」の2通り

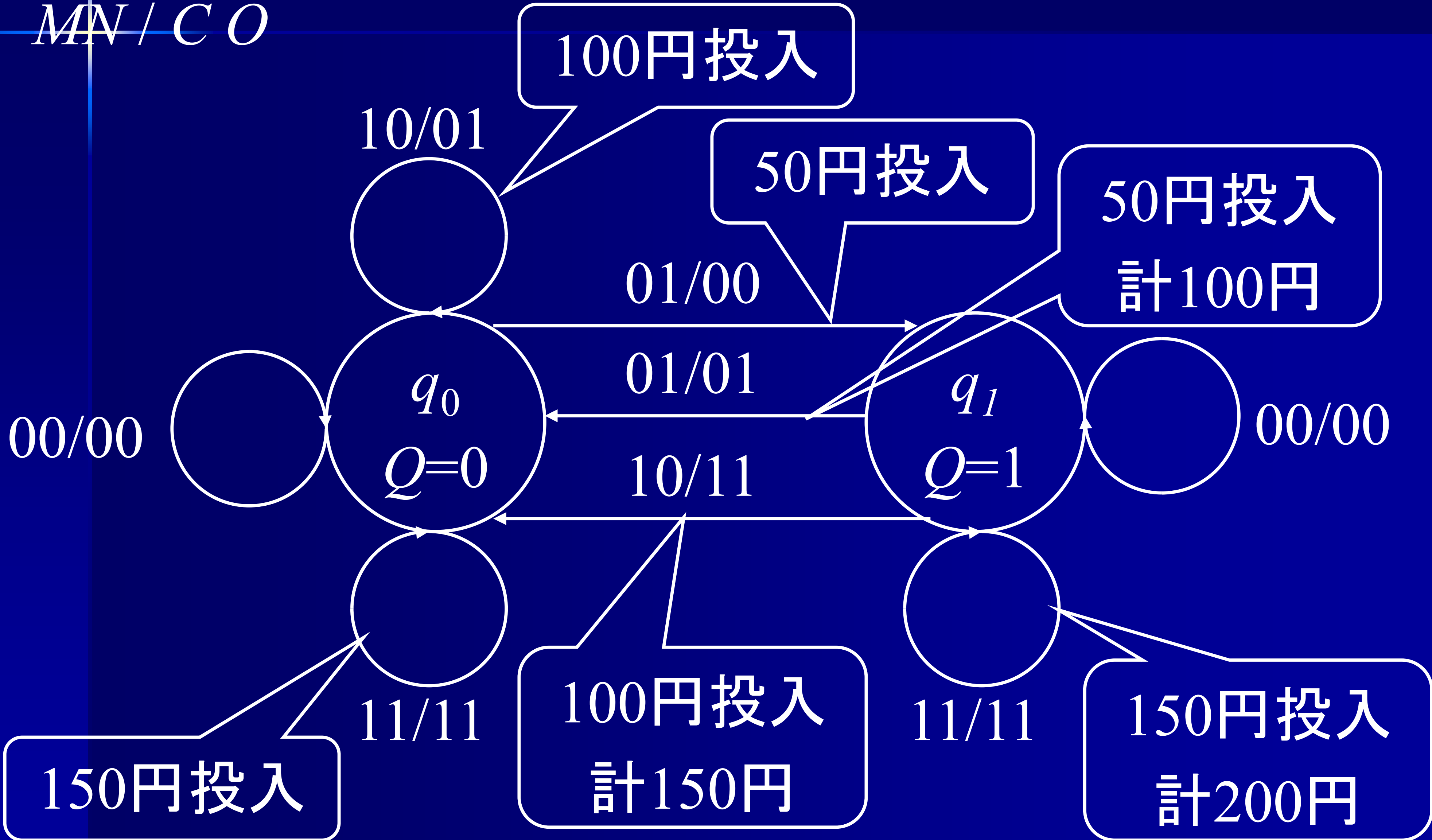
1. 入力,出力,状態を決める

入力	N	50円を投入
	M	100円を投入
出力	O	商品を出す
	C	お釣り50円を返却
状態	q_0	0円投入された
	q_1	50円投入された

2入力2出力1FF

2.状態遷移図を描く

$MN \,/\, C\,O$



3.状態遷移表を作る
4.拡大入力要求表を作る

M	N	Q	Q^+	C	O	D	JK
0	0	0	0	0	0	0	0 -
0	1		1	0	0	1	1 -
1	0		0	0	1	0	0 -
1	1		0	1	1	0	0 -
0	0	1	1	0	0	1	- 0
0	1		0	0	1	0	- 1
1	0		0	1	1	0	- 1
1	1		1	1	1	1	- 0

5.FFの入力条件式,6.出力関数を求める

C

Q \ MN	00	01	11	10
0			1	
1			1	1

O

Q \ MN	00	01	11	10
0			1	1
1		1	1	1

5.FFの入力条件式,6.出力関数を求める

D

Q \ MN	00	01	11	10
0		1		
1	1		1	

$$C = M \cdot N + M \cdot Q$$

$$O = M + N \cdot Q$$

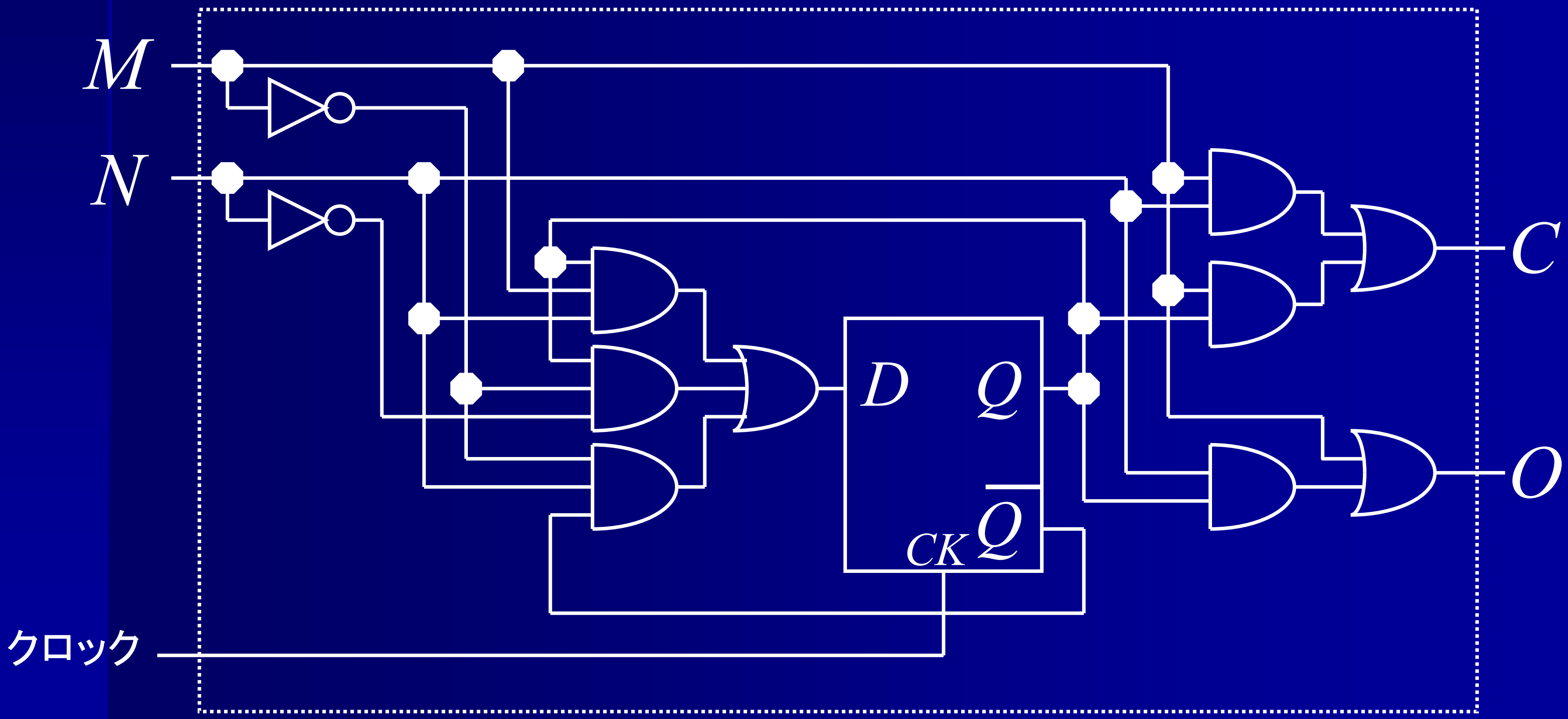
$$D = \overline{M} \cdot \overline{N} \cdot Q + \overline{M} \cdot N \cdot \overline{Q} + M \cdot N \cdot Q$$

7.回路図を描く

$$C = M \cdot N + M \cdot Q$$

$$O = M + N \cdot Q$$

$$D = \overline{M} \cdot \overline{N} \cdot Q + \overline{M} \cdot N \cdot \overline{Q} + M \cdot N \cdot Q$$



J

Q \ MN	00	01	11	10
0		1		
1	-	-	-	-

K

Q \ MN	00	01	11	10
0	-	-	-	-
1		1		1

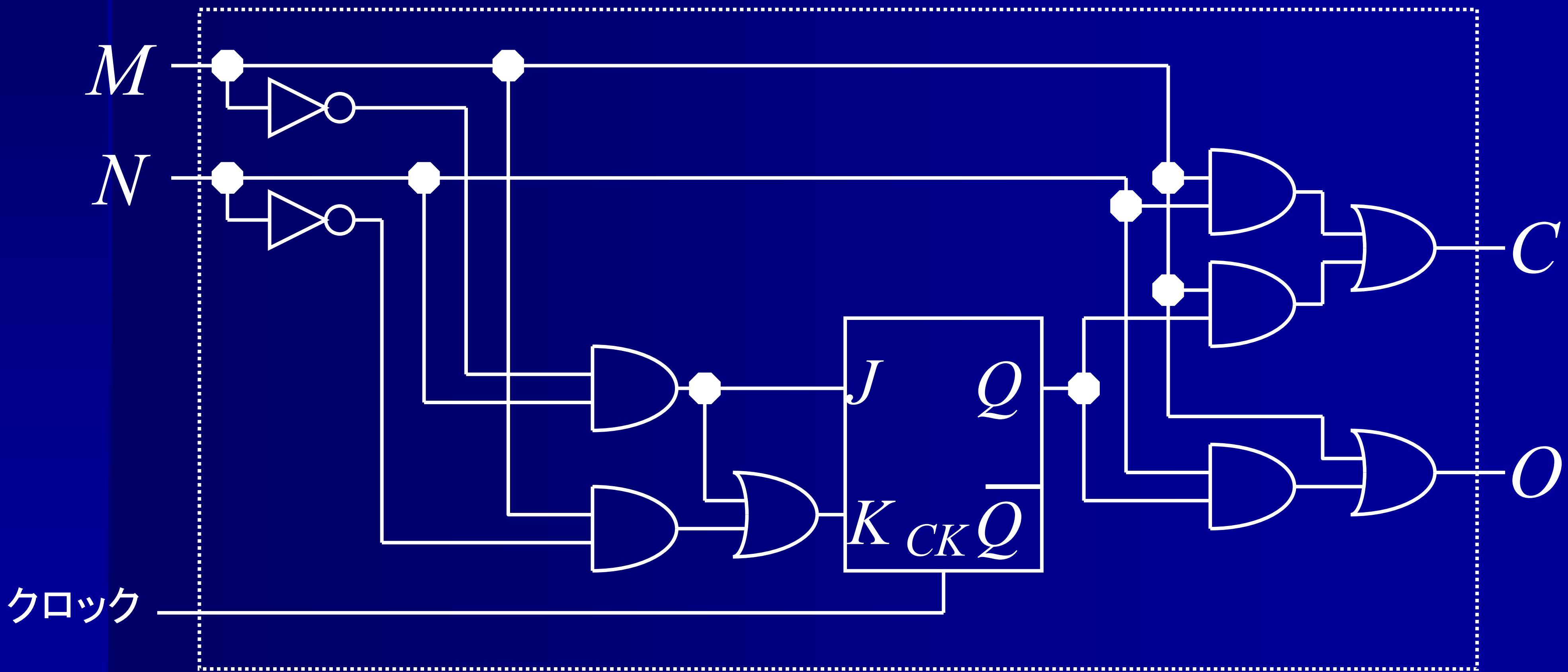
$$C = M \cdot N + M \cdot Q \qquad O = M + N \cdot Q$$

$$J = \overline{M} \cdot N \qquad K = \overline{M} \cdot N + M \cdot \overline{N} = M \oplus N$$

7.回路図を描く

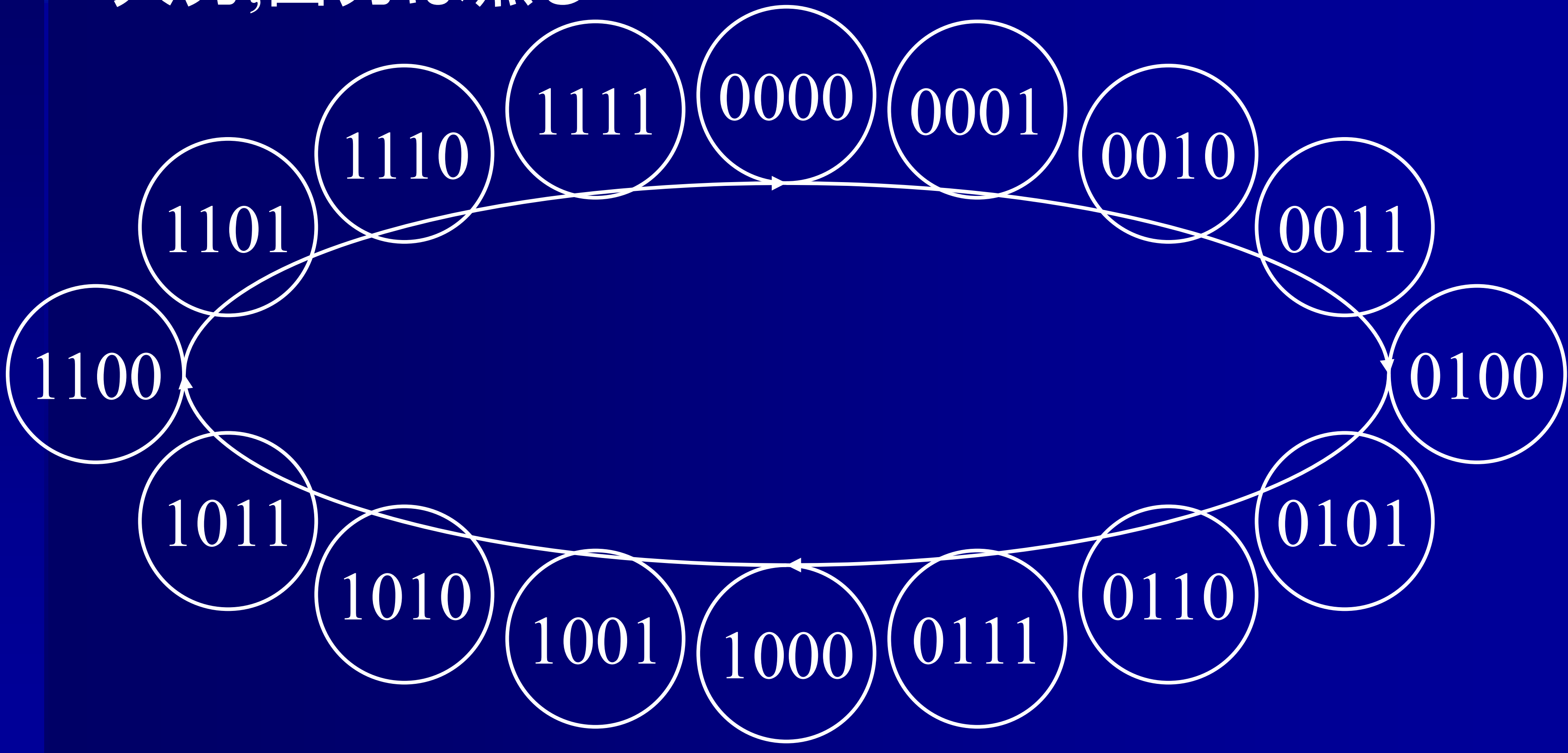
$$C = M \cdot N + M \cdot Q \qquad O = M + N \cdot Q$$

$$J = \overline{M} \cdot N \qquad K = \overline{M} \cdot N + M \cdot \overline{N} = M \oplus N$$



演習問題：カウンタの設計

- TFFを用いて同期式16進カウンタを設計せよ
  - 入力,出力は無し



Q_3	Q_2	Q_1	Q_0	Q_3^+	Q_2^+	Q_1^+	Q_0^+	T_3	T_2	T_1	T_0
0	0	0	0	0	0	0	1				1
0	0	0	1	0	0	1	0			1	1
0	0	1	0	0	0	1	1				1
0	0	1	1	0	1	0	0		1	1	1
0	1	0	0	0	1	0	1				1
0	1	0	1	0	1	1	0			1	1
0	1	1	0	0	1	1	1				1
0	1	1	1	1	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1				1
1	0	0	1	1	0	1	0			1	1
1	0	1	0	1	0	1	1				1
1	0	1	1	1	1	0	0		1	1	1
1	1	0	0	1	1	0	1				1
1	1	0	1	1	1	1	0			1	1
1	1	1	0	1	1	1	1				1
1	1	1	1	0	0	0	0	1	1	1	1

T_3

Q_1Q_0 / Q_3Q_2	00	01	11	10
00				
01			1	
11			1	
10				

T_2

Q_1Q_0 / Q_3Q_2	00	01	11	10
00			1	
01			1	
11			1	
10			1	

T_1

Q_1Q_0 / Q_3Q_2	00	01	11	10
00		1	1	
01		1	1	
11		1	1	
10		1	1	

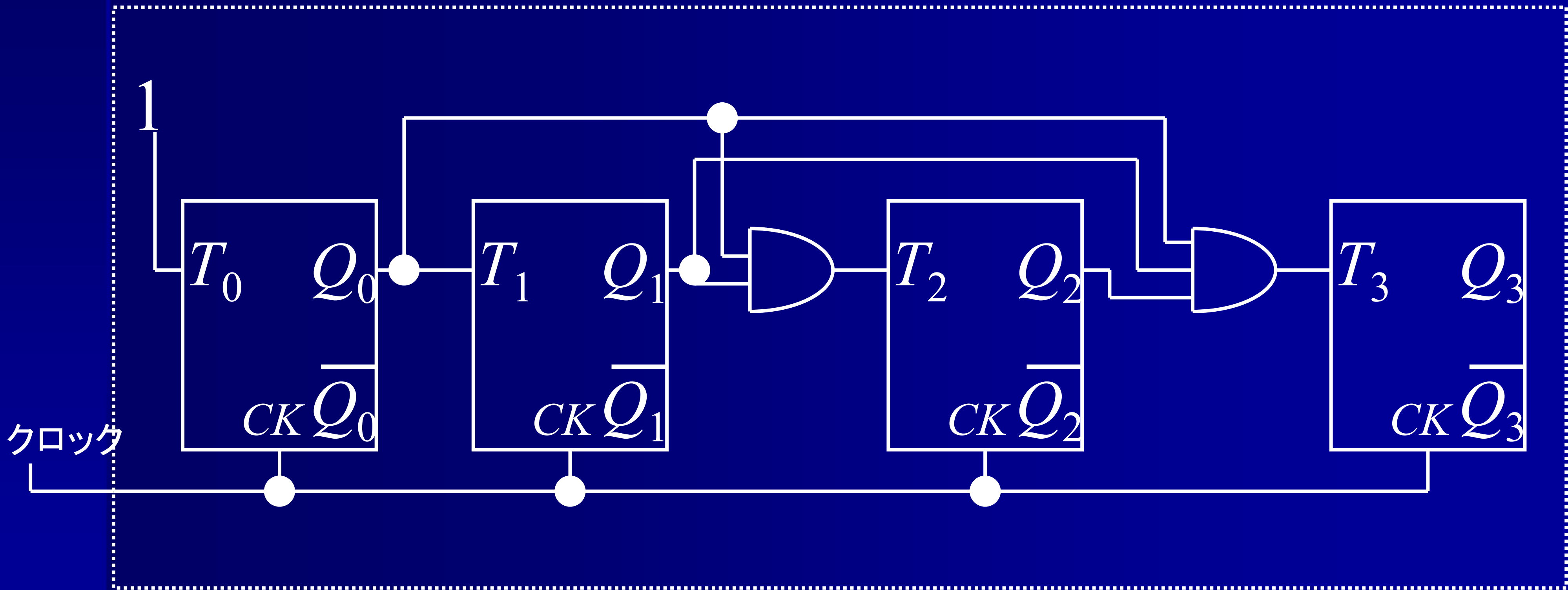
T_0

Q_1Q_0 / Q_3Q_2	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

TFFを用いた同期式16進分周器

$$T_3 = Q_2 \cdot Q_1 \cdot Q_0 \quad T_2 = Q_1 \cdot Q_0$$

$$T_1 = Q_0 \quad T_0 = 1$$



カウンタはJKFFをTFFに置き換えるだけでよい

https://www.info.kindai.ac.jp/LC/

このページは2022年度の「論理回路」の公式ホームページです． ここに講義録，課題，レポートの提出方法他の情報を掲載します．

連絡

- Logisim実習について

  を 用いたシミュレーション実習を行います。 こちらのページを見て 各自ノートPCに Losigim をインスト

  また、以下のファイルをダウンロードしておいてください。(4/1)

  Logisim 用

  - Logisim ファイル一式 (下記の *.circ ファイルをまとめたものです)
  - (第4回講義用)：gate1.circ, gate2.circ, gate3.circ, gate4.circ, gate5.circ, gate6.circ, MP2.circ, FA.circ, FA4.circ, FAS4.circ
  - (第11回講義用)：FF.circ, BR.circ, Sft.circ, Reg.circ, SftReg.circ, Ctr16.circ, Ctr10.circ, CtrX.circ

  (注意)OS のバージョンが 11.2.3 Big Sur 以降では Logisim を使えません。その場合は、こちらのページを見てLogisim の代わりに Logisim-evolution をインストールしてください。また、以下のファイルをダウンロードしておいてください。(上記のLogisim 用

  さい) (4/1)

  Logisim-evolution 用

  - Logisim-evolution ファイル一式 (下記の *.circ ファイルをまとめたものです)
  - (第4回講義用)：gate1.circ, gate2.circ, gate3.circ, gate4.circ, gate5.circ, gate6.circ, MP2.circ, FA.circ, FA4.circ, FAS4.circ
  - (第11回講義用)：FF.circ, BR.circ, Sft.circ, Reg.circ, SftReg.circ, Ctr16.circ, Ctr10.circ, CtrX.circ

- 出席について

  単位取得には原則として全ての授業に出席する必要があります。 やむを得ず欠席する場合はその翌週までに必ず欠席届を出してください。 欠席届無しの欠席が複数回ある場合は履修の意思無しと見做して不受扱いにします。

  オンライン授業では、当日 GoogleClassrooom から出席カードが提出がされていれば出席扱いにします。

- 課題について

問題：自動販売機の設計

- 100円投入されると商品を出す
  - コイン投入口は100円,50円共通用が1つ
    (一度にコイン1枚しか投入できない)
  - お釣り返還口は50円用が1つ

投入金額	自動販売機の動作
0円	次の投入待ち
50円	次の投入待ち
100円	商品を出して0円に
150円	商品とお釣り50円を出して0円に

入力 $(M, N) = (1, 1)$ (150円投入) はドントケア