

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

1

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

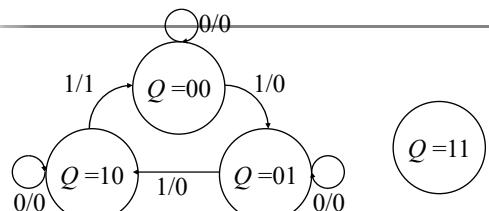
- 全ての状態に対する状態遷移関数, 出力関数が定義されている論理関数

■ 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

2

不完全指定論理関数



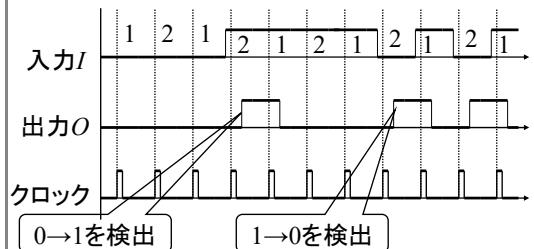
状態 $Q=11$ は状態遷移関数,
出力関数が未定義
状態 11 はドントケア

3

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



4

同期式回路の設計

1. 入力(I_1, I_2, \dots, I_m), 出力(O_1, O_2, \dots, O_n), 状態(Q_1, Q_2, \dots, 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

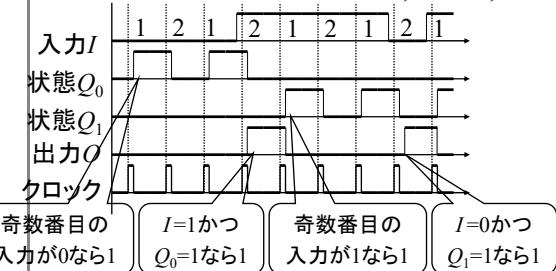
5

1. 入力, 出力, 状態の決定

■ 入力が奇数番目か偶数番目か?

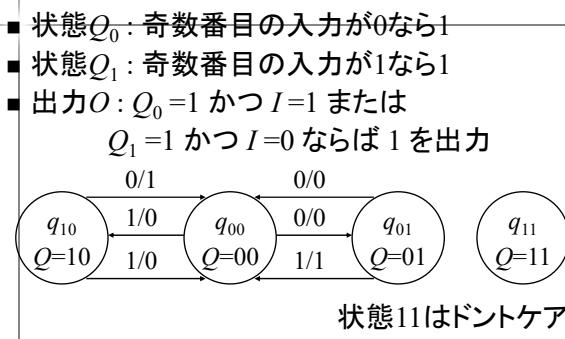
■ 奇数番目の入力が0か1か?

⇒ FF2個で記憶可能 1入力, 1出力, 2FF



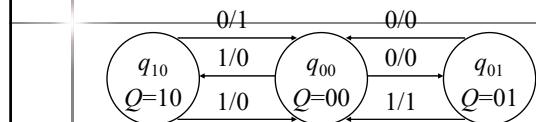
6

2. 状態遷移図を描く



7

3. 状態遷移表を作成する



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

8

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			-	-	-	-	-

9

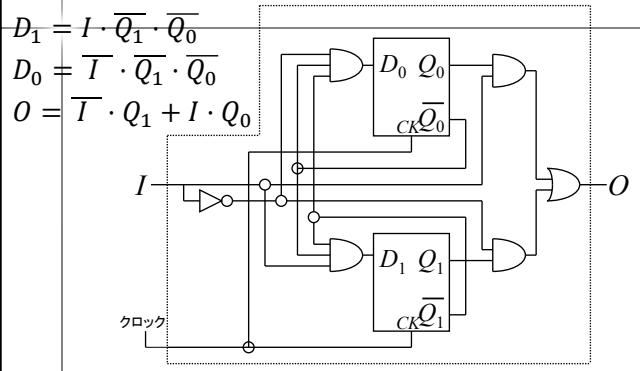
5. FFの入力条件式、 6. 出力関数を求める

I	Q_1	Q_0	O	D_1	D_0	D_1	Q_1Q_0	00	01	11	10
0	0	0	0	0	1	0				-	
1			0	1	0	1				-	
0	0	1	0	0	0	0				-	
1			1	0	0	1				-	
0	1	0	1	0	0	1		0	1		
1			0	0	0	0		0	0	0	
0	1	1	-	-	-	-		1	1	-	
1			-	-	-	-		1	1	-	

D_0	I	Q_1	Q_0	00	01	11	10
0	1	0	0	0	1		
1				0	0	0	
0	1	1	0	0	0	0	
1				1	1	-	
$D_1 = I \cdot \overline{Q_1} \cdot \overline{Q_0}$	1	0	1	00	01	11	10
$D_0 = \overline{I} \cdot \overline{Q_1} \cdot \overline{Q_0}$				0	1	-	1
$O = \overline{I} \cdot Q_1 + I \cdot Q_0$	1	1	1	0	1	1	-
				1	1	-	1

10

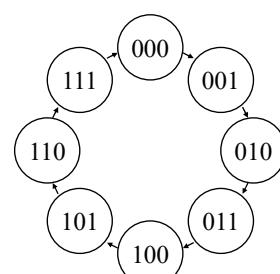
7. 回路図を描く



11

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 $000 \rightarrow 001 \rightarrow \dots \rightarrow 111 \rightarrow 000$ と遷移する回路



12

スライド 8

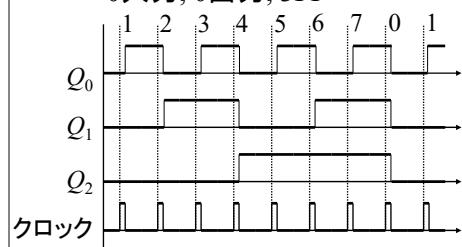
石水隆1 石水隆, 2020/06/28

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



13

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	0	1
0	1	1	1	1	0
1	0	0	1	0	1
1	0	1	1	0	1
1	1	0	0	0	0
1	1	1	0	0	0

14

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

15

5.FFの入力条件式を求める

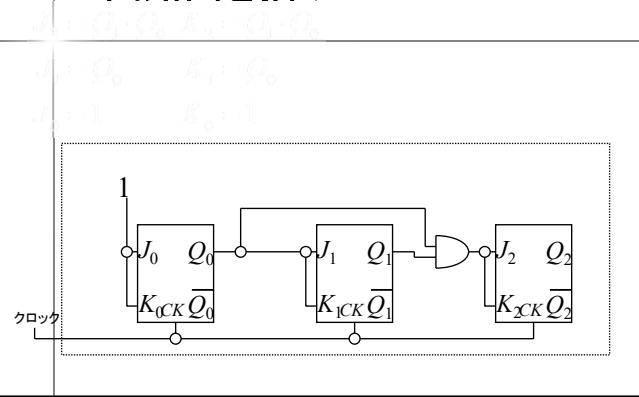
Q_2	Q_1	Q_0	J_2K_2	J_1K_1	J_0K_0	J_2K_2	Q_2Q_0	00	01	11	10
0	0	0	0 -	0 -	1 -	0	0 -	0 -	1 -	0 -	
0	0	1	0 -	1 -	- 1	1	- 0	- 0	- 1	- 0	
0	1	0	0 -	- 0	1 -	1	0 -	0 -	1 -	1 -	
0	1	1	1 -	- 1	- 1	1	0 -	1 -	- 1	- 0	
1	0	0	- 0	0 -	1 -	1	0 -	1 -	- 1	- 1	
1	0	1	- 0	1 -	- 1	1	0 -	1 -	- 1	- 1	
1	1	0	- 0	- 0	1 -	1	0 -	1 -	- 1	- 1	
1	1	1	- 1	- 1	- 1	1	1 -	1 -	- 1	- 1	

Q_2	Q_1	Q_0	J_2K_2	J_1K_1	J_0K_0	J_2K_2	Q_2Q_0	00	01	11	10
0	0	0	0 -	0 -	1 -	0	0 -	0 -	1 -	0 -	
0	0	1	0 -	1 -	- 1	1	- 0	- 0	- 1	- 0	
0	1	0	0 -	- 0	1 -	1	0 -	0 -	1 -	1 -	
0	1	1	1 -	- 1	- 1	1	0 -	1 -	- 1	- 0	
1	0	0	- 0	0 -	1 -	1	0 -	1 -	- 1	- 1	
1	0	1	- 0	1 -	- 1	1	0 -	1 -	- 1	- 1	
1	1	0	- 0	- 0	1 -	1	0 -	1 -	- 1	- 1	
1	1	1	- 1	- 1	- 1	1	1 -	1 -	- 1	- 1	

Q_2	Q_1	Q_0	J_2K_2	J_1K_1	J_0K_0	J_2K_2	Q_2Q_0	00	01	11	10
0	0	0	0 -	0 -	1 -	0	0 -	0 -	1 -	0 -	
0	0	1	0 -	1 -	- 1	1	- 0	- 0	- 1	- 0	
0	1	0	0 -	- 0	1 -	1	0 -	0 -	1 -	1 -	
0	1	1	1 -	- 1	- 1	1	0 -	1 -	- 1	- 0	
1	0	0	- 0	0 -	1 -	1	0 -	1 -	- 1	- 1	
1	0	1	- 0	1 -	- 1	1	0 -	1 -	- 1	- 1	
1	1	0	- 0	- 0	1 -	1	1 -	1 -	- 1	- 1	
1	1	1	- 1	- 1	- 1	1	1 -	1 -	- 1	- 1	

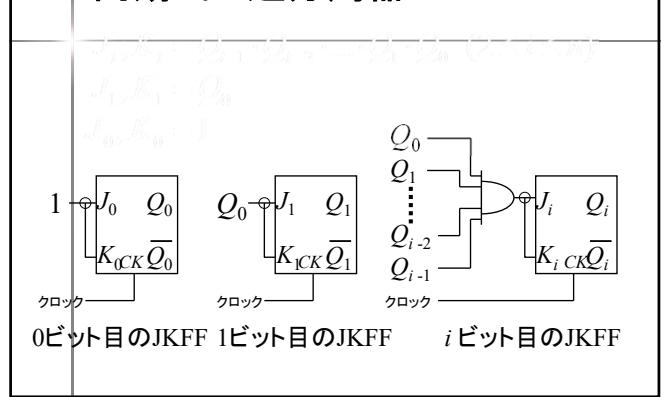
16

7.回路図を描く



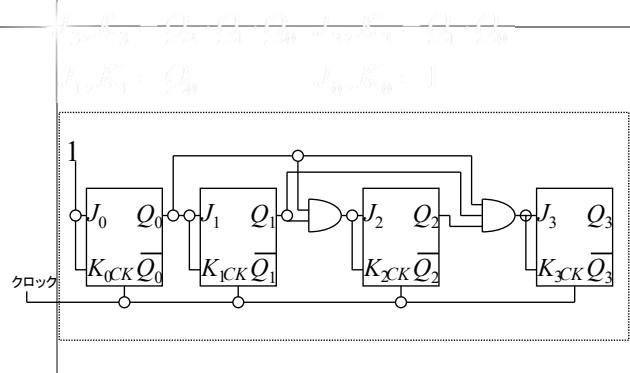
17

同期式 2^n 進分周器



18

同期式16進分周器



19

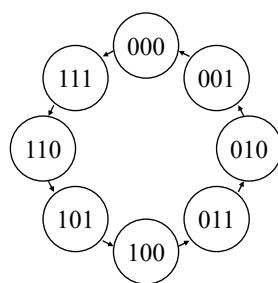
様々なカウンタ

- 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)

20

2^n 進減算カウンタ

- 1ずつ減らす遷移をする回路



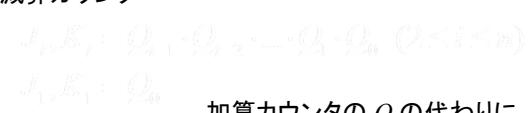
21

同期式 2^n 進減算カウンタ

加算カウンタ



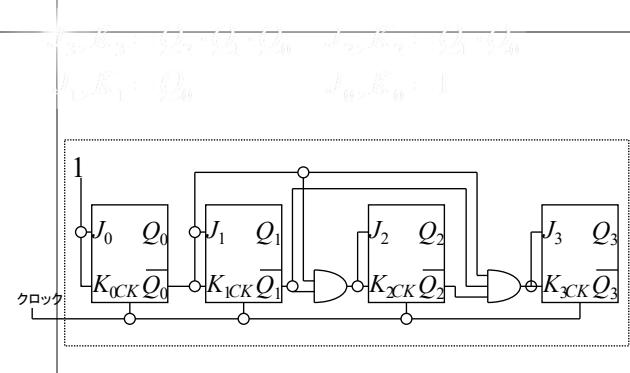
減算カウンタ



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

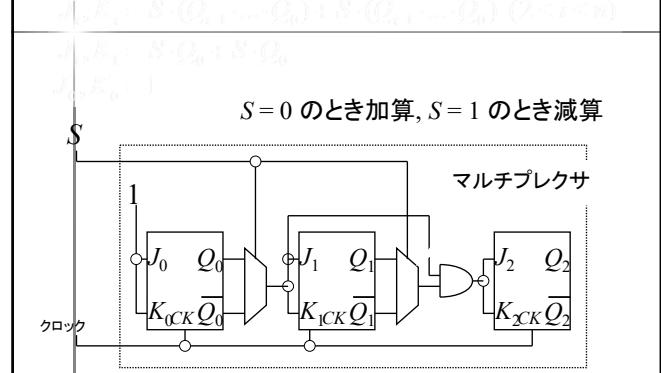
22

同期式16進減算カウンタ



23

同期式 2^n 進加減算カウンタ



24

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

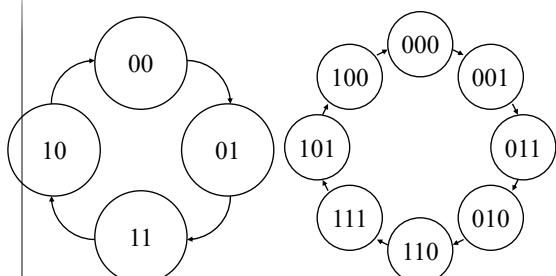
25

グレイコードの作り方

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	3ビット
000 001 011 010 110 111 101 100	000 001 011 010 110 111 101 100
先頭に0	先頭に1
0000 0001 0011 0010 0110 0111 0101 0110	0111 0101 0100
4ビット	4ビット逆順
100 101 111 110 010 011 001 000	000 001 011 010 110 111 101 100
先頭に1	先頭に0
1100 1101 1111 1110 1010 1011 1001 1000	1000 1011 1010 1001 1101 1100 1111 1110

26

グレイコードカウンタ



27

- 状態遷移表を作る
- 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

28

5. FFの入力条件式を求める

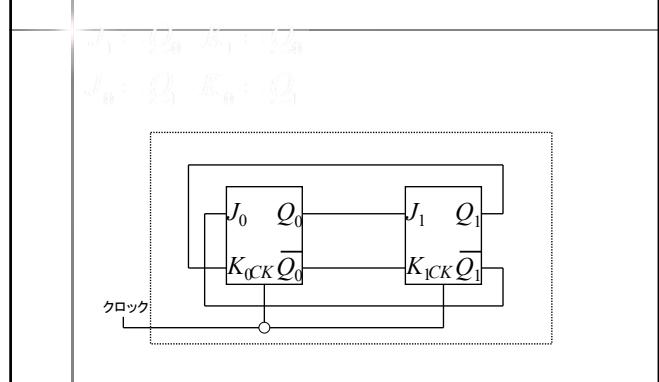
		J_1K_1	
Q_1	Q_0	J_1K_1	J_0K_0
0	0	0 -	1 -
0	1	1 -	- 0
1	0	- 1	0 -
1	1	- 0	- 1

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

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

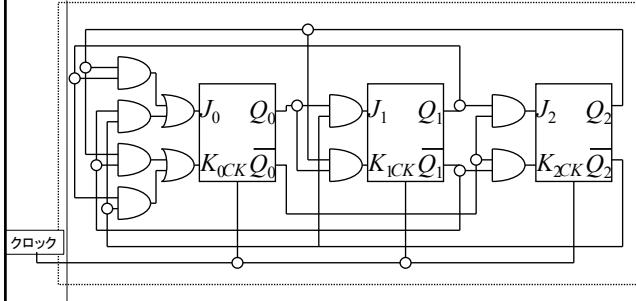
29

7.回路図を描く



30

3ビットグレイコードカウンタ



31

2ⁿ進→グレイコード変換

■ 2ⁿ進からグレイコードへの変換

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

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

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

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

$b = 01110$ (14) グレイコードカウンタは直接作るより

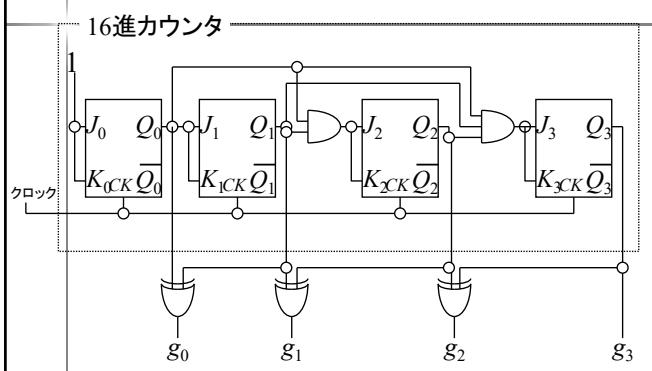
2ⁿ進カウンタから変換した方が簡単

$$g = 01001$$

(※) グレイコード→2ⁿ進は難しい

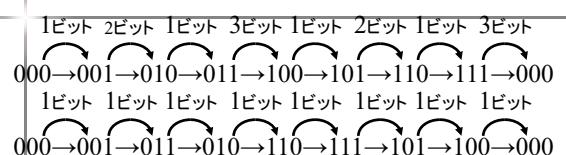
32

4ビットグレイコードカウンタ



33

グレイコードカウンタの利点



増加時に変化するビットは 1 ビットのみ

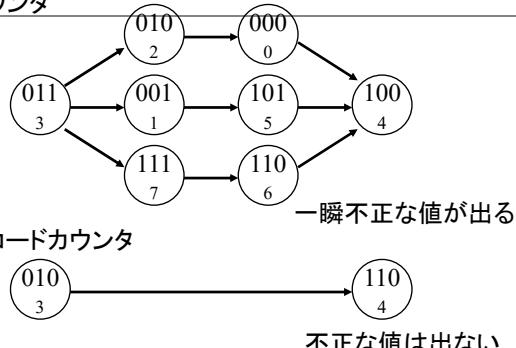


変化途中に不正な値が現れない

34

グレイコードカウンタの利点

8進カウンタ



35

ジョンソン(Johnson)カウンタ

■ n ビット2 n 状態の不完全指定論理関数

2ビット 00,01,11,10

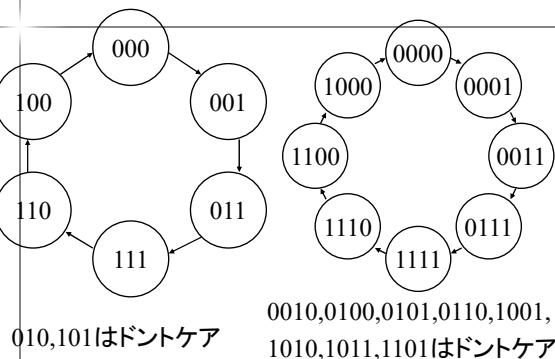
3ビット 000,001,011,111,110,100

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
						右から順に1を0に変える	

36

ジョンソンカウンタ



37

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

38

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

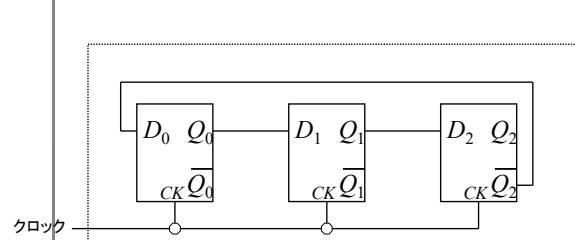
$Q_2 Q_0$	00	01	11	10
Q_2	0	1	-	-
0			1	-
1		-	1	1

$Q_2 Q_0$	00	01	11	10
Q_2	0	1	1	-
0		1	1	-
1		-	1	

$Q_2 Q_0$	00	01	11	10
Q_2	0	1	1	-
0	1	1	1	-
1		-		

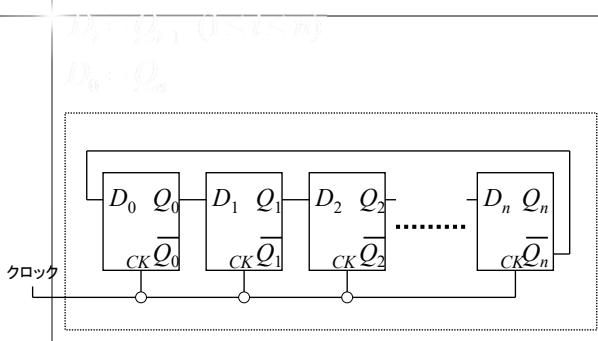
39

7.回路図を描く



40

n ビットジョンソンカウンタ



41

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

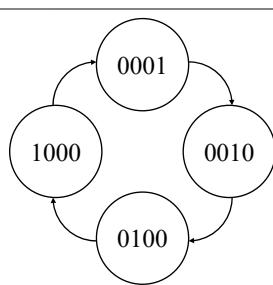
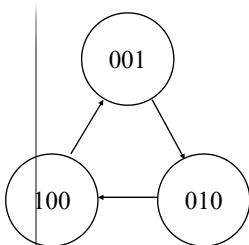
3ビット 001,010,100

4ビット 0001,0010,0100,1000

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

42

リングカウンタ



43

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	-	-

44

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	-	-	-

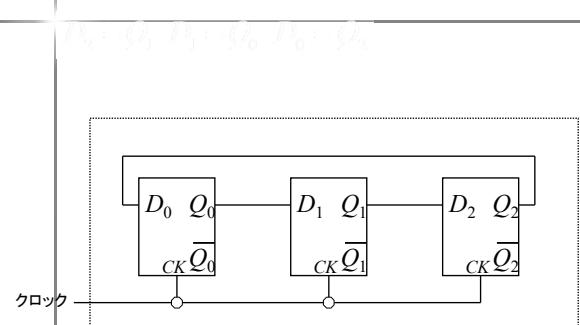
$Q_2 Q_0$	00	01	11	10
$Q_1 Q_0$	00	01	11	10
0	-		-	1
1		-	-	-

$Q_2 Q_0$	00	01	11	10
$Q_1 Q_0$	00	01	11	10
0	-	1	-	
1		-	-	-

$Q_2 Q_0$	00	01	11	10
$Q_1 Q_0$	00	01	11	10
0	-		-	
1	1	-	-	-

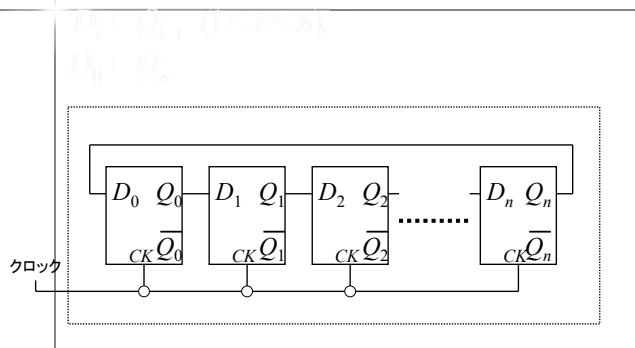
45

7.回路図を描く



46

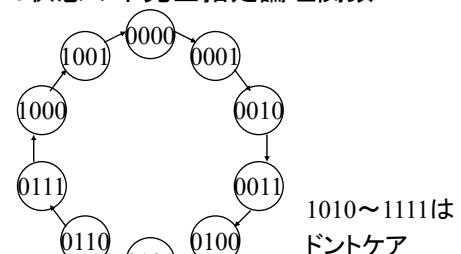
n ビットリングカウンタ



47

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



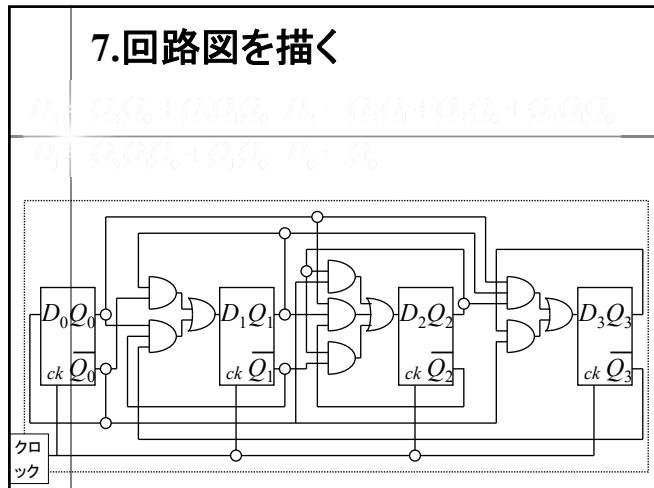
48

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	-

49

5. FFの入力条件式を求める			
D_3	D_2	D_1	D_0
$Q_1 Q_0$ $Q_2 Q_1$	00 01 11 10	00 01 11 10	00 01 11 10
00			
01		1	
11	-	-	-
10	1	-	-
$Q_1 Q_0$ $Q_2 Q_1$	00 01 11 10	00 01 11 10	00 01 11 10
00	1	1	1
01	1	1	1
11	-	-	-
10	-	-	-

50



51

4. 拡大入力表を作る(JKFF)			
$J_3 K_3 J_2 K_2 J_1 K_1 J_0 K_0$	$J_3 K_3 J_2 K_2 J_1 K_1 J_0 K_0$	$Q_3 Q_2 Q_1 Q_0$	$J_3 K_3 J_2 K_2 J_1 K_1 J_0 K_0$
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 - 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	-

52

5. FFの入力条件式を求める			
$J_3 K_3$	$J_2 K_2$	$J_1 K_1$	$J_0 K_0$
$Q_1 Q_0$ $Q_2 Q_1$	00 01 11 10	$Q_1 Q_0$ $Q_2 Q_1$	00 01 11 10
00 0 - 0 - 0 - 0 -	00 0 - 0 - 1 - 0 -	00 0 - 1 - 1 - 0 -	00 0 - 1 - 1 - 1 -
01 0 - 0 - 1 - 0 -	01 - 0 - 0 - 1 - 0 -	01 1 - - 1 - 1 -	01 1 - - 1 - 1 -
11 - - - - - -	11 - - - - - -	11 - - - - - -	11 - - - - - -
10 - 0 - 1 - -	10 0 - 0 - -	10 1 - - 1 - -	10 1 - - 1 - -
$J_3 K_3$	$J_2 K_2$	$J_1 K_1$	$J_0 K_0$
$Q_1 Q_0$ $Q_2 Q_1$	00 01 11 10	$Q_1 Q_0$ $Q_2 Q_1$	00 01 11 10
00 0 - 1 - 1 - 0 -	00 1 - - 1 - 1 -	00 1 - - 1 - 1 -	00 1 - - 1 - 1 -
01 0 - 1 - 1 - 0 -	01 1 - - 1 - 1 -	01 1 - - 1 - 1 -	01 1 - - 1 - 1 -
11 - - - - - -	11 - - - - - -	11 - - - - - -	11 - - - - - -
10 0 - 0 - - -	10 1 - - 1 - -	10 1 - - 1 - -	10 1 - - 1 - -

53

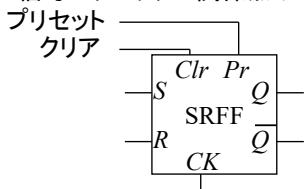
7. 回路図を描く			
$J_3 = Q_2 \cdot Q_1 \cdot Q_0$ $J_2 = Q_1 \cdot Q_0$ $J_1 = \bar{Q}_3 \cdot Q_0$ $J_0 = 1$			
$K_3 = Q_0$ $K_2 = Q_1 \cdot Q_0$ $K_1 = Q_0$ $K_0 = 1$			
J_0	J_1	J_2	J_3
$K_{CK} \bar{Q}_0$	$K_{CK} \bar{Q}_1$	$K_{CK} \bar{Q}_2$	$K_{CK} \bar{Q}_3$
1			
J_0	J_1	J_2	J_3
$K_{CK} \bar{Q}_0$	$K_{CK} \bar{Q}_1$	$K_{CK} \bar{Q}_2$	$K_{CK} \bar{Q}_3$

54

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

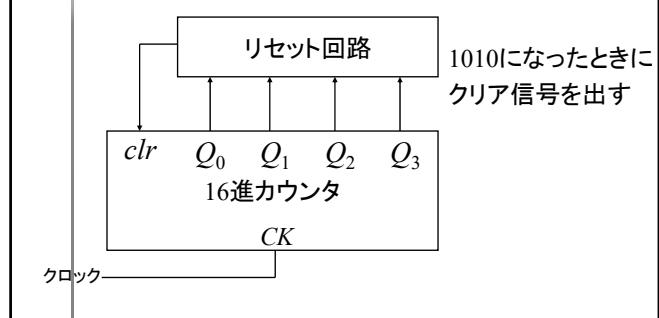
- 通常の入力(SR, D, T, JK)に加え、
Preset信号 Pr とClear信号 Clr を入力
Preset信号でクロックに関係無く1にセット
Clear信号でクロックに関係無く0にリセット



55

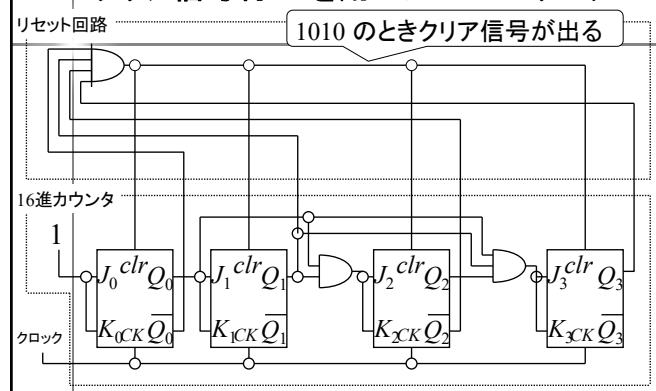
クリア信号付FFを用いたBCDカウンタ

■16進カウンタ+リセット回路



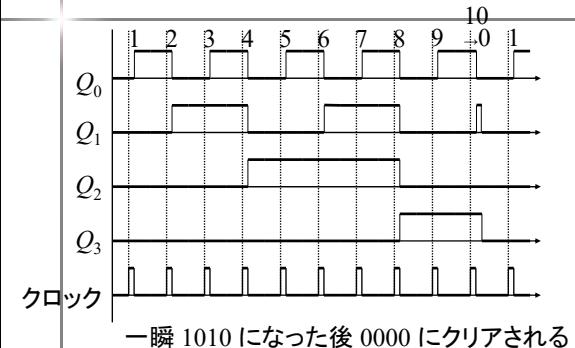
56

クリア信号付FFを用いたBCDカウンタ



57

クリア信号付FFによるBCDカウンタの遷移

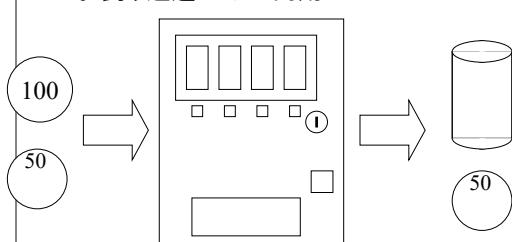


58

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



59

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

60

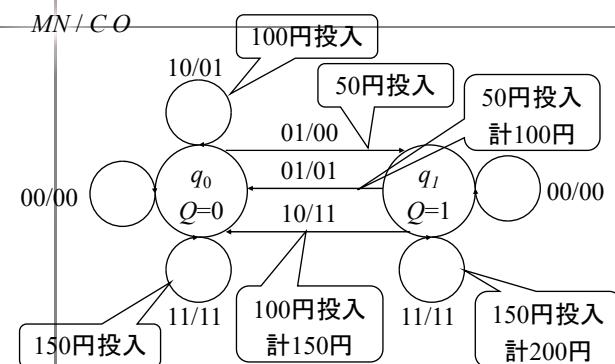
1. 入力,出力,状態を決める

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

61

2. 状態遷移図を描く



62

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

63

5. FFの入力条件式, 6. 出力関数を求める

C	$MN \backslash Q$	00	01	11	10
Q	0			1	
	1			1	1

O	$MN \backslash Q$	00	01	11	10
Q	0			1	1
	1		1	1	1

64

5. FFの入力条件式, 6. 出力関数を求める

D	$MN \backslash Q$	00	01	11	10
Q	0		1		
	1	1		1	

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

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

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

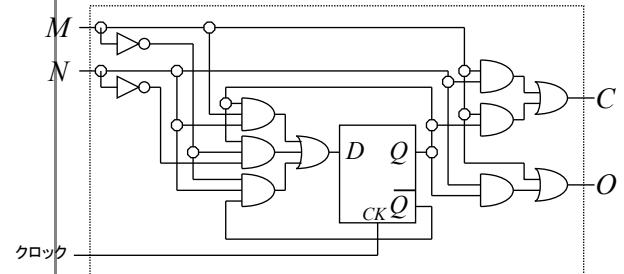
65

7. 回路図を描く

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

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

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



66

		MN		00	01	11	10
		Q	M				
J	0			1			
	1		-	-	-	-	-
K		MN		00	01	11	10
		Q	M				
K	0		-	-	-	-	-
	1			1			1

67

7.回路図を描く

$$C = M \cdot N + M \cdot \bar{Q} \quad O = M + N \cdot \bar{Q}$$

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

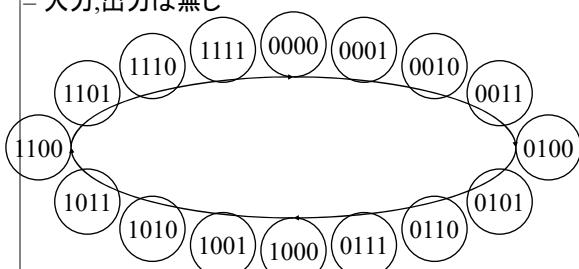
The diagram shows a logic circuit with inputs M and N , and a clock input labeled "クロック". The circuit uses an SR flip-flop with inputs J and K , and outputs Q and \bar{Q} . The output C is the product of M and N . The output O is the sum of M and the product of N and the complement of Q . The circuit is implemented as follows:

- The inputs M and N are connected to the J and K inputs of the SR flip-flop respectively.
- The clock input of the SR flip-flop is labeled "クロック".
- The output Q is connected to one input of a two-input AND gate.
- The output \bar{Q} is connected to one input of a two-input OR gate.
- The inputs of the AND gate are also connected to the M and N inputs.
- The outputs of the AND and OR gates are connected to the inputs of a two-input NOR gate, which produces the output C .
- The output Q is also connected to one input of a two-input AND gate.
- The output O is the sum of the outputs from the SR flip-flop and the AND gate, produced by a two-input OR gate.

68

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



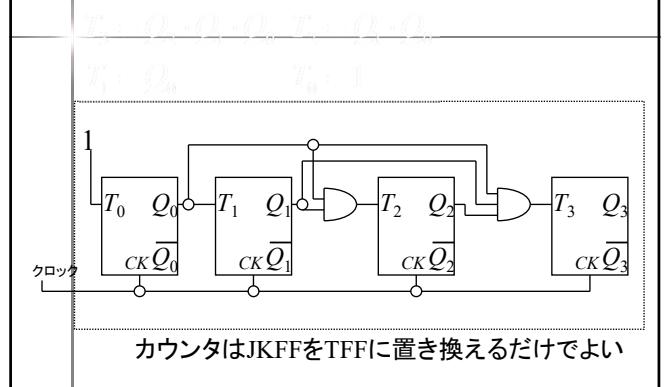
69

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	1	0	1	1	1	1				1
1	1	1	1	0	0	0	0	1	1	1	1

70

71

TFFを用いた同期式16進分周器



カウンタはJKFFをTFFに置き換えるだけでよい

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

Logisim 用

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Logisim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 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, Stl.circ, Reg.circ, SfReg.circ, Ctrl6.circ, Ctrl10.circ, CtrlX.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 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, Stl.circ, Reg.circ, SfReg.circ, Ctrl6.circ, Ctrl10.circ, CtrlX.circ

- 出題について

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複数の意思無しと見做して不受理にします。

オンライン授業では、当日 GoogleClassroom から出席カードが提出がされなければ出席扱いにします。

- 課題について

73

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

74