

デジタル電子回路

授業開始までしばらくお待ちください。

オンライン視聴できない人へ。

オンラインで受講する人も基本的に一緒です。

自宅ネットワークの事情により、授業のストリーミング配信の視聴が困難な学生は以下の対応をしてください。

- ① この授業のスライドをよく読んで、不明な点は自分で調べるなどして、わかる範囲で内容を理解する。
- ② このページも含め、**必要な部分がすべて理解できたと思うまで以下の2ステップを繰り返す。**
 - ▶ わからない部分を e-mail 等で質問する。(宛先は `hiroyuki.kobayashi@oit.ac.jp`)
 - ▶ e-mail 等による返信をよく読んで理解する。
- ③ この資料の末尾にある課題を行い、この資料内の方法で (Google Forms で) 提出する。

授業の受講に関して

- 講義資料（スライド等）は**COMMON**に置く。
- 講義は**Google Meet**で行い、録画した講義は**Goole Drive**に置く。

<https://stream.meet.google.com/stream/1d1866da-5bff-4881-96b2-3745413fe31a>



https://drive.google.com/drive/folders/1bT-z3ICQyMYC_5Jv1L29UZYqbOhVG492

- 出席確認レポートは**Google Forms**で提出。(毎回同一 URL)

<https://forms.gle/9ruwtfJg5LQgQNpU7>

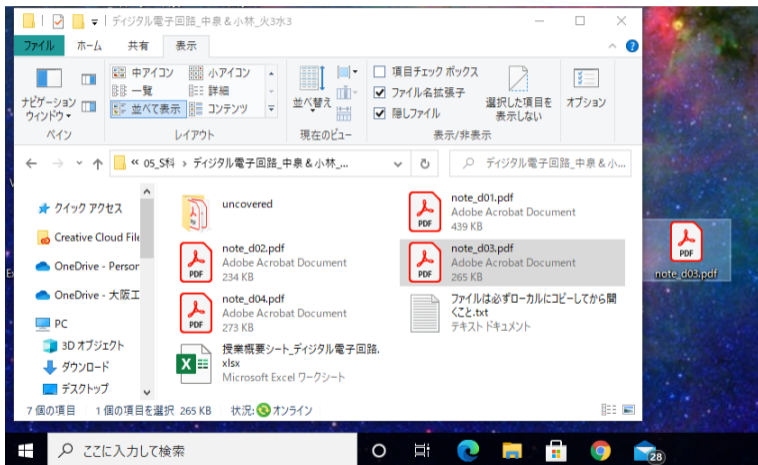


- **Slack**を補助的な連絡チャネルとする。必須ではないので使いたくなくれば使わなくてもいい。授業に関連したちょっとした（重要でない）追加説明をする。気楽な質問手段としても活用されたい。登録は大学の e-mail アドレスで行うこと。

<https://oitkobayashi.slack.com>

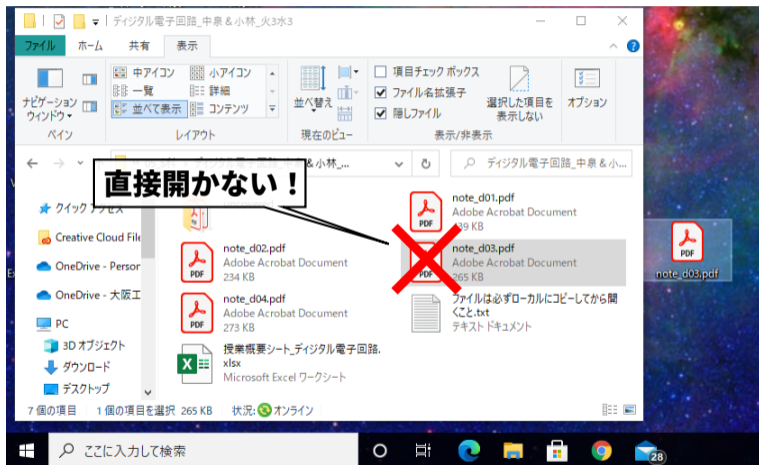
COMMON フォルダの注意事項 (全授業共通)

根源的に悪いのは Windows の仕様なのですが、ご協力ください。



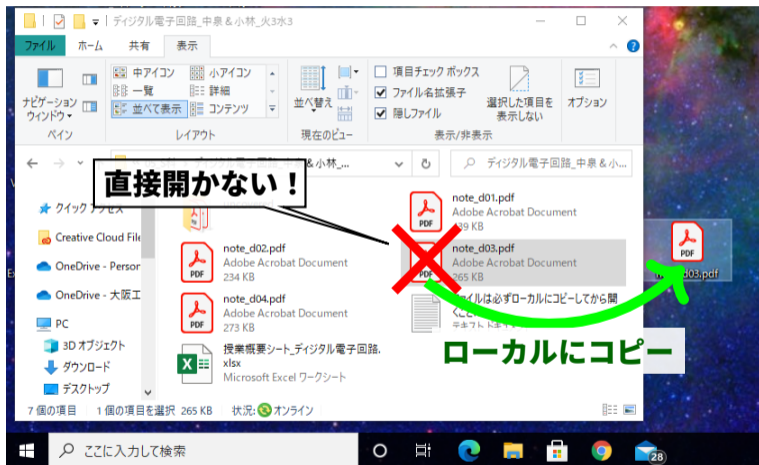
COMMON フォルダの注意事項 (全授業共通)

根源的に悪いのは Windows の仕様なのですが、ご協力ください。



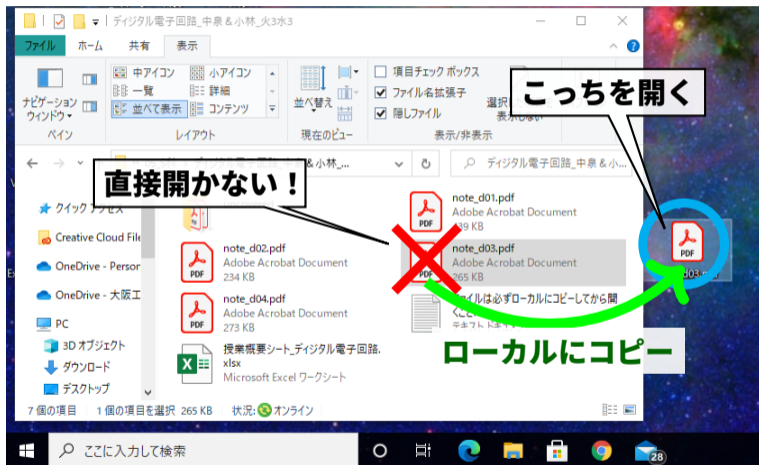
COMMON フォルダの注意事項 (全授業共通)

根源的に悪いのは Windows の仕様なのですが、ご協力ください。



COMMON フォルダの注意事項 (全授業共通)

根源的に悪いのは Windows の仕様なのですが、ご協力ください。



R/S 科デジタル電子回路

Digital Electronics



『順序回路応用 (≡ 練習問題集) 他』Google Meet

小林裕之・中泉文孝

大阪工業大学 RD 学部システムデザイン工学科・ロボット工学科




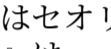
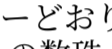
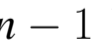


OSAKA INSTITUTE OF TECHNOLOGY

10 of 14

a L^AT_EX + Beamer slideshow

カウンタ回路

- クロックのパルス数を数える回路。
- 『』と『』とがある。
 - ▶ 『』はセオリーどおりの順序回路。**素直。**
 - ▶ 『』は  の数珠つなぎ。**単純。**
- いくつかのオマケ機能:
 - ▶ アップ/ダウン切り替え入力
 - ▶ 停止/カウント切り替え入力
 - ▶ リセット入力
- $0 \sim n - 1$ まで数えられるカウンタを『』
という。 $(n$ 進数とは関係ないので注意!!)

カウンタ回路

- クロックのパルス数を数える回路。
- 『同期カウンタ』と『非同期カウンタ』とがある。
 - ▶ 『同期カウンタ』はセオリーどおりの順序回路。**素直**。
 - ▶ 『非同期カウンタ』は 74161 の数珠つなぎ。**単純**。
- いくつかのオマケ機能:
 - ▶ アップ/ダウン切り替え入力
 - ▶ 停止/カウント切り替え入力
 - ▶ リセット入力
- $0 \sim n - 1$ まで数えられるカウンタを『 n 進カウンタ』という。 $(n$ 進数とは関係ないので注意!!)

カウンタ回路

- クロックのパルス数を数える回路。
- 『同期カウンタ』と『非同期カウンタ』とがある。
 - ▶ 『同期カウンタ』はセオリーどおりの順序回路。**素直**。
 - ▶ 『非同期カウンタ』は の数珠つなぎ。**単純**。
- いくつかのオマケ機能:
 - ▶ アップ/ダウン切り替え入力
 - ▶ 停止/カウント切り替え入力
 - ▶ リセット入力
- $0 \sim n - 1$ まで数えられるカウンタを『 』
という。 $(n$ 進数とは関係ないので注意!!)

カウンタ回路

- クロックのパルス数を数える回路。
- 『同期カウンタ』と『非同期カウンタ』とがある。
 - ▶ 『同期カウンタ』はセオリーどおりの順序回路。素直。
 - ▶ 『非同期カウンタ』は T-FF の数珠つなぎ。単純。
- いくつかのオマケ機能:
 - ▶ アップ/ダウン切り替え入力
 - ▶ 停止/カウント切り替え入力
 - ▶ リセット入力
- $0 \sim n - 1$ まで数えられるカウンタを『
』
という。(n 進数とは関係ないので注意!!)

カウンタ回路

- クロックのパルス数を数える回路。
- 『同期カウンタ』と『非同期カウンタ』とがある。
 - ▶ 『同期カウンタ』はセオリーどおりの順序回路。素直。
 - ▶ 『非同期カウンタ』は T-FF の数珠つなぎ。単純。
- いくつかのオマケ機能:
 - ▶ アップ/ダウン切り替え入力
 - ▶ 停止/カウント切り替え入力
 - ▶ リセット入力
- $0 \sim n - 1$ まで数えられるカウンタを『 n 進カウンタ』という。 $(n$ 進数とは関係ないので注意!!)

例 1: リセットつき同期式 3 進 UP カウンタ

課題: 以下のカウンタを設計せよ

- 同期式 3 進アップカウンタ ($00 \rightarrow 01 \rightarrow 10 \rightarrow 00 \rightarrow \dots$)
- 出力はカウント値 (00, 01, 10) とキャリー (桁溢れしたときに 1 になる)
- リセット入力つき (これを 1 にするといつでも 00 になる)

$$M = (S, X, Z, \delta, \omega)$$

$$S = \{00, 01, 10\}$$

$$X = \{0, 1\}$$

$$Z = \{0, 1\}$$

10

00

01

- ※ 入力のリセットのみ (リセットも不要であれば空集合でも構わない)。
- ※ カウント値出力は状態を直接出せばいいので考えなくて良い。キャリーのみ。

例 1: リセットつき同期式 3 進 UP カウンタ

課題: 以下のカウンタを設計せよ

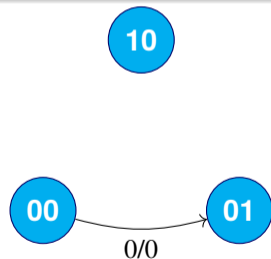
- 同期式 3 進アップカウンタ ($00 \rightarrow 01 \rightarrow 10 \rightarrow 00 \rightarrow \dots$)
- 出力はカウント値 (00, 01, 10) とキャリー (桁溢れしたときに 1 になる)
- リセット入力つき (これを 1 にするといつでも 00 になる)

$$M = (S, X, Z, \delta, \omega)$$

$$S = \{00, 01, 10\}$$

$$X = \{0, 1\}$$

$$Z = \{0, 1\}$$



- ※ 入力のリセットのみ (リセットも不要であれば空集合でも構わない)。
- ※ カウント値出力は状態を直接出せばいいので考えなくて良い。キャリーのみ。

例 1: リセットつき同期式 3 進 UP カウンタ

課題: 以下のカウンタを設計せよ

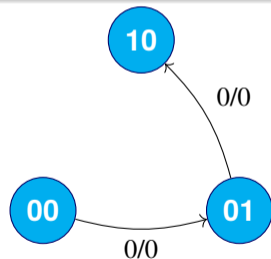
- 同期式 3 進アップカウンタ ($00 \rightarrow 01 \rightarrow 10 \rightarrow 00 \rightarrow \dots$)
- 出力はカウント値 (00, 01, 10) とキャリー (桁溢れしたときに 1 になる)
- リセット入力つき (これを 1 にするといつでも 00 になる)

$$M = (S, X, Z, \delta, \omega)$$

$$S = \{00, 01, 10\}$$

$$X = \{0, 1\}$$

$$Z = \{0, 1\}$$



※ 入力のリセットのみ (リセットも不要であれば空集合でも構わない)。

※ カウント値出力は状態を直接出せばいいので考えなくて良い。キャリーのみ。

例 1: リセットつき同期式 3 進 UP カウンタ

課題: 以下のカウンタを設計せよ

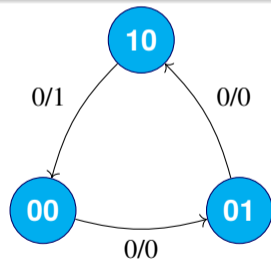
- 同期式 3 進アップカウンタ ($00 \rightarrow 01 \rightarrow 10 \rightarrow 00 \rightarrow \dots$)
- 出力はカウント値 (00, 01, 10) とキャリー (桁溢れしたときに 1 になる)
- リセット入力つき (これを 1 にするといつでも 00 になる)

$$M = (S, X, Z, \delta, \omega)$$

$$S = \{00, 01, 10\}$$

$$X = \{0, 1\}$$

$$Z = \{0, 1\}$$



※ 入力のリセットのみ (リセットも不要であれば空集合でも構わない)。

※ カウント値出力は状態を直接出せばいいので考えなくて良い。キャリーのみ。

例 1: リセットつき同期式 3 進 UP カウンタ

課題: 以下のカウンタを設計せよ

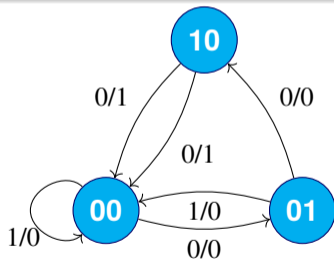
- 同期式 3 進アップカウンタ ($00 \rightarrow 01 \rightarrow 10 \rightarrow 00 \rightarrow \dots$)
- 出力はカウント値 (00, 01, 10) とキャリー (桁溢れしたときに 1 になる)
- リセット入力つき (これを 1 にするといつでも 00 になる)

$$M = (S, X, Z, \delta, \omega)$$

$$S = \{00, 01, 10\}$$

$$X = \{0, 1\}$$

$$Z = \{0, 1\}$$



※ 入力のリセットのみ (リセットも不要であれば空集合でも構わない)。

※ カウント値出力は状態を直接出せばいいので考えなくて良い。キャリーのみ。

例 1: リセットつき同期式 3 進 UP カウンタ (状態遷移表)

		$\delta(Q_1, Q_0, x)$		$\omega(Q_1, Q_0, x)$	
今の状態		次の状態		出力	
Q_1	Q_0	Q'_1 Q'_0	Q'_1 Q'_0	z	
		@ $x = 0$	@ $x = 1$	@ $x = 0$	@ $x = 1$

memo:

x リセット入力 (1= リセットする)

z キャリー出力 (1= 桁溢れ発生)

例 1: リセットつき同期式 3 進 UP カウンタ (状態遷移表)

		$\delta(Q_1, Q_0, x)$		$\omega(Q_1, Q_0, x)$	
今の状態		次の状態		出力	
Q_1	Q_0	Q'_1 Q'_0	Q'_1 Q'_0	z	
		@ $x = 0$	@ $x = 1$	@ $x = 0$	@ $x = 1$
0	0				
0	1				
1	0				

memo:

x リセット入力 (1= リセットする)

z キャリー出力 (1= 桁溢れ発生)

例 1: リセットつき同期式 3 進 UP カウンタ (状態遷移表)

		$\delta(Q_1, Q_0, x)$		$\omega(Q_1, Q_0, x)$	
今の状態		次の状態		出力	
Q_1	Q_0	Q'_1	Q'_0	z	
		@ $x = 0$	@ $x = 1$	@ $x = 0$	@ $x = 1$
0	0	0	1		
0	1	1	0		
1	0	0	0		

memo:

x リセット入力 (1= リセットする)

z キャリー出力 (1= 桁溢れ発生)

例 1: リセットつき同期式 3 進 UP カウンタ (状態遷移表)

		$\delta(Q_1, Q_0, x)$		$\omega(Q_1, Q_0, x)$	
今の状態		次の状態		出力	
Q_1	Q_0	Q'_1 Q'_0	Q'_1 Q'_0	z	
		@ $x = 0$	@ $x = 1$	@ $x = 0$	@ $x = 1$
0	0	0 1	0 0		
0	1	1 0	0 0		
1	0	0 0	0 0		

memo:

x リセット入力 (1= リセットする)

z キャリー出力 (1= 桁溢れ発生)

例 1: リセットつき同期式 3 進 UP カウンタ (状態遷移表)

		$\delta(Q_1, Q_0, x)$		$\omega(Q_1, Q_0, x)$	
今の状態		次の状態		出力	
Q_1	Q_0	Q'_1 Q'_0	Q'_1 Q'_0	z	
		@ $x = 0$	@ $x = 1$	@ $x = 0$	@ $x = 1$
0	0	0 1	0 0	0	0
0	1	1 0	0 0	0	0
1	0	0 0	0 0	1	0

memo:

x リセット入力 (1= リセットする)

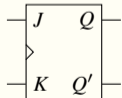
z キャリー出力 (1= 桁溢れ発生)

例 1: リセットつき同期式 3 進 UP カウンタ (δ の実現)

memo

$\delta(Q_1, Q_0, x)$

今の状態		次の状態			
Q_1	Q_0	Q'_1 @ $x = 0$	Q'_0 @ $x = 0$	Q'_1 @ $x = 1$	Q'_0 @ $x = 1$
0	0	0	1	0	0
0	1	1	0	0	0
1	0	0	0	0	0



JK-FF の励起表

$Q \rightarrow Q'$	J	K
$0 \rightarrow 0$	0	-
$0 \rightarrow 1$	1	-
$1 \rightarrow 0$	-	1
$1 \rightarrow 1$	-	0

今回は JK-FF で実現してみる。

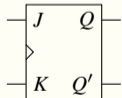
入力	今の状態		次の状態		FF 入力			
x	Q_1	Q_0	Q'_1	Q'_0	J_1	K_1	J_0	K_0

例 1: リセットつき同期式 3 進 UP カウンタ (δ の実現)

memo

$\delta(Q_1, Q_0, x)$

今の状態		次の状態			
Q_1	Q_0	Q'_1 @ $x = 0$	Q'_0 @ $x = 0$	Q'_1 @ $x = 1$	Q'_0 @ $x = 1$
0	0	0	1	0	0
0	1	1	0	0	0
1	0	0	0	0	0



JK-FF の励起表

$Q \rightarrow Q'$	J	K
$0 \rightarrow 0$	0	-
$0 \rightarrow 1$	1	-
$1 \rightarrow 0$	-	1
$1 \rightarrow 1$	-	0

今回は JK-FF で実現してみる。

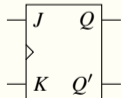
入力 x	今の状態 $Q_1 \quad Q_0$		次の状態 $Q'_1 \quad Q'_0$		FF 入力 $J_1 \quad K_1 \quad J_0 \quad K_0$			
	0	0						
	0	1						
	1	0						

例 1: リセットつき同期式 3 進 UP カウンタ (δ の実現)

memo

$\delta(Q_1, Q_0, x)$

今の状態		次の状態			
Q_1	Q_0	Q'_1 @ $x = 0$	Q'_0 @ $x = 0$	Q'_1 @ $x = 1$	Q'_0 @ $x = 1$
0	0	0	1	0	0
0	1	1	0	0	0
1	0	0	0	0	0



JK-FF の励起表

$Q \rightarrow Q'$	J	K
$0 \rightarrow 0$	0	-
$0 \rightarrow 1$	1	-
$1 \rightarrow 0$	-	1
$1 \rightarrow 1$	-	0

今回は JK-FF で実現してみる。

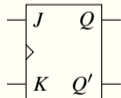
入力	今の状態		次の状態		FF 入力			
x	Q_1	Q_0	Q'_1	Q'_0	J_1	K_1	J_0	K_0
0	0	0						
1	0	0						
0	0	1						
1	0	1						
0	1	0						
1	1	0						

例 1: リセットつき同期式 3 進 UP カウンタ (δ の実現)

memo

$\delta(Q_1, Q_0, x)$

今の状態		次の状態			
Q_1	Q_0	Q'_1 @ $x = 0$	Q'_0 @ $x = 0$	Q'_1 @ $x = 1$	Q'_0 @ $x = 1$
0	0	0	1	0	0
0	1	1	0	0	0
1	0	0	0	0	0



JK-FF の励起表

$Q \rightarrow Q'$	J	K
$0 \rightarrow 0$	0	-
$0 \rightarrow 1$	1	-
$1 \rightarrow 0$	-	1
$1 \rightarrow 1$	-	0

今回は JK-FF で実現してみる。

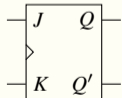
入力	今の状態		次の状態		FF 入力			
x	Q_1	Q_0	Q'_1	Q'_0	J_1	K_1	J_0	K_0
0	0	0	0	1				
1	0	0	0	0				
0	0	1	1	0				
1	0	1	0	0				
0	1	0	0	0				
1	1	0	0	0				

例 1: リセットつき同期式 3 進 UP カウンタ (δ の実現)

memo

$\delta(Q_1, Q_0, x)$

今の状態		次の状態			
Q_1	Q_0	Q'_1 @ $x = 0$	Q'_0 @ $x = 0$	Q'_1 @ $x = 1$	Q'_0 @ $x = 1$
0	0	0	1	0	0
0	1	1	0	0	0
1	0	0	0	0	0



JK-FF の励起表

$Q \rightarrow Q'$	J	K
$0 \rightarrow 0$	0	—
$0 \rightarrow 1$	1	—
$1 \rightarrow 0$	—	1
$1 \rightarrow 1$	—	0

今回は JK-FF で実現してみる。

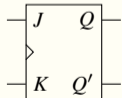
入力	今の状態		次の状態		FF 入力			
x	Q_1	Q_0	Q'_1	Q'_0	J_1	K_1	J_0	K_0
0	0	0	0	1	0	—		
1	0	0	0	0				
0	0	1	1	0				
1	0	1	0	0				
0	1	0	0	0				
1	1	0	0	0				

例 1: リセットつき同期式 3 進 UP カウンタ (δ の実現)

memo

$\delta(Q_1, Q_0, x)$

今の状態		次の状態			
Q_1	Q_0	Q'_1 @ $x = 0$	Q'_0 @ $x = 0$	Q'_1 @ $x = 1$	Q'_0 @ $x = 1$
0	0	0	1	0	0
0	1	1	0	0	0
1	0	0	0	0	0



JK-FF の励起表

$Q \rightarrow Q'$	J	K
$0 \rightarrow 0$	0	—
$0 \rightarrow 1$	1	—
$1 \rightarrow 0$	—	1
$1 \rightarrow 1$	—	0

今回は JK-FF で実現してみる。

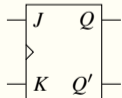
入力	今の状態		次の状態		FF 入力			
x	Q_1	Q_0	Q'_1	Q'_0	J_1	K_1	J_0	K_0
0	0	0	0	1	0	—		
1	0	0	0	0	0	—	0	—
0	0	1	1	0				
1	0	1	0	0	0	—		
0	1	0	0	0			0	—
1	1	0	0	0			0	—

例 1: リセットつき同期式 3 進 UP カウンタ (δ の実現)

memo

$\delta(Q_1, Q_0, x)$

今の状態		次の状態			
Q_1	Q_0	Q'_1 @ $x = 0$	Q'_0 @ $x = 0$	Q'_1 @ $x = 1$	Q'_0 @ $x = 1$
0	0	0	1	0	0
0	1	1	0	0	0
1	0	0	0	0	0



JK-FF の励起表

$Q \rightarrow Q'$	J	K
$0 \rightarrow 0$	0	–
$0 \rightarrow 1$	1	–
$1 \rightarrow 0$	–	1
$1 \rightarrow 1$	–	0

今回は JK-FF で実現してみる。

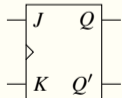
入力	今の状態		次の状態		FF 入力			
x	Q_1	Q_0	Q'_1	Q'_0	J_1	K_1	J_0	K_0
0	0	0	0	1	0	–		
1	0	0	0	0	0	–	0	–
0	0	1	1	0	1	–		
1	0	1	0	0	0	–		
0	1	0	0	0			0	–
1	1	0	0	0			0	–

例 1: リセットつき同期式 3 進 UP カウンタ (δ の実現)

memo

$\delta(Q_1, Q_0, x)$

今の状態		次の状態			
Q_1	Q_0	Q'_1 @ $x = 0$	Q'_0 @ $x = 0$	Q'_1 @ $x = 1$	Q'_0 @ $x = 1$
0	0	0	1	0	0
0	1	1	0	0	0
1	0	0	0	0	0



JK-FF の励起表

$Q \rightarrow Q'$	J	K
$0 \rightarrow 0$	0	—
$0 \rightarrow 1$	1	—
$1 \rightarrow 0$	—	1
$1 \rightarrow 1$	—	0

今回は JK-FF で実現してみる。

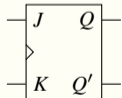
入力	今の状態		次の状態		FF 入力			
x	Q_1	Q_0	Q'_1	Q'_0	J_1	K_1	J_0	K_0
0	0	0	0	1	0	—	1	—
1	0	0	0	0	0	—	0	—
0	0	1	1	0	1	—		
1	0	1	0	0	0	—		
0	1	0	0	0			0	—
1	1	0	0	0			0	—

例 1: リセットつき同期式 3 進 UP カウンタ (δ の実現)

memo

$\delta(Q_1, Q_0, x)$

今の状態		次の状態			
Q_1	Q_0	Q'_1 @ $x = 0$	Q'_0 @ $x = 0$	Q'_1 @ $x = 1$	Q'_0 @ $x = 1$
0	0	0	1	0	0
0	1	1	0	0	0
1	0	0	0	0	0



JK-FF の励起表

$Q \rightarrow Q'$	J	K
$0 \rightarrow 0$	0	-
$0 \rightarrow 1$	1	-
$1 \rightarrow 0$	-	1
$1 \rightarrow 1$	-	0

今回は JK-FF で実現してみる。

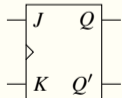
入力 x	今の状態 $Q_1 \quad Q_0$		次の状態 $Q'_1 \quad Q'_0$		FF 入力 $J_1 \quad K_1 \quad J_0 \quad K_0$			
0	0	0	0	1	0	-	1	-
1	0	0	0	0	0	-	0	-
0	0	1	1	0	1	-	-	1
1	0	1	0	0	0	-		
0	1	0	0	0			0	-
1	1	0	0	0			0	-

例 1: リセットつき同期式 3 進 UP カウンタ (δ の実現)

memo

$\delta(Q_1, Q_0, x)$

今の状態		次の状態			
Q_1	Q_0	Q'_1 @ $x = 0$	Q'_0 @ $x = 0$	Q'_1 @ $x = 1$	Q'_0 @ $x = 1$
0	0	0	1	0	0
0	1	1	0	0	0
1	0	0	0	0	0



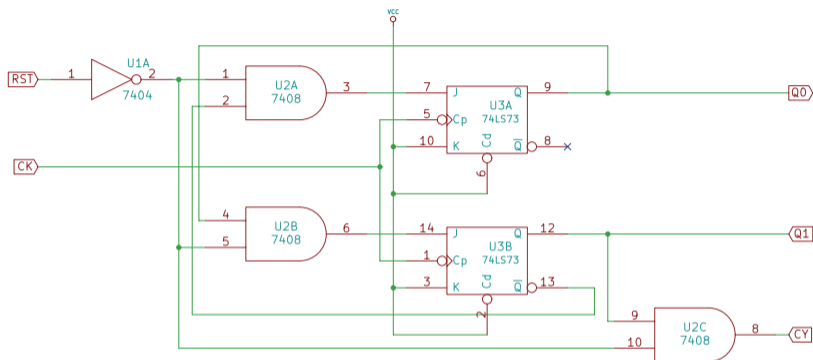
JK-FF の励起表

$Q \rightarrow Q'$	J	K
$0 \rightarrow 0$	0	—
$0 \rightarrow 1$	1	—
$1 \rightarrow 0$	—	1
$1 \rightarrow 1$	—	0

今回は JK-FF で実現してみる。

入力	今の状態		次の状態		FF 入力			
x	Q_1	Q_0	Q'_1	Q'_0	J_1	K_1	J_0	K_0
0	0	0	0	1	0	—	1	—
1	0	0	0	0	0	—	0	—
0	0	1	1	0	1	—	—	1
1	0	1	0	0	0	—	—	1
0	1	0	0	0	—	1	0	—
1	1	0	0	0	—	1	0	—

例 1: リセットつき同期式 3 進 UP カウンタ (設計例)

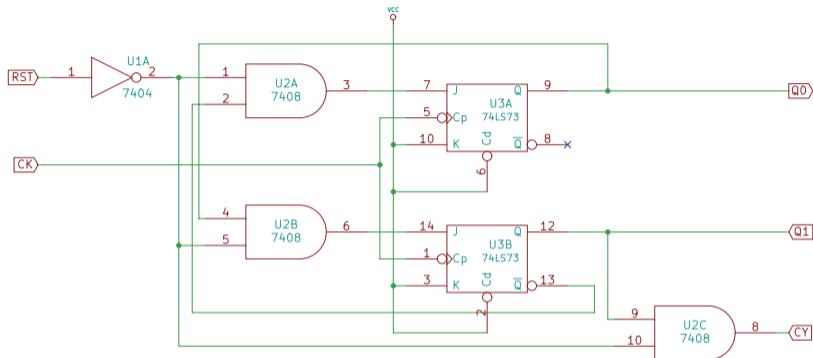


$$J_1 = \quad , K_1 =$$

$$J_0 = \quad , K_0 =$$

$$Z =$$

例 1: リセットつき同期式 3 進 UP カウンタ (設計例)



$$J_1 = \bar{x} \cdot Q_0, K_1 = 1$$

$$J_0 = \bar{x} \cdot \bar{Q}_1, K_0 = 1$$

$$Z = \bar{x} \cdot Q_1$$

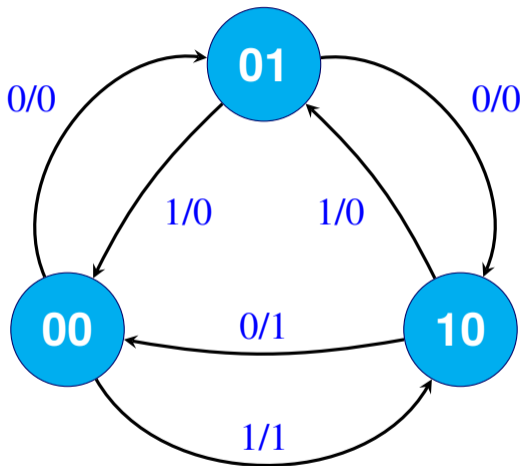
練習: 同期式 3 進 UP/DOWN カウンタ

今度は自分でやってみよう。

課題: 以下のカウンタを設計せよ

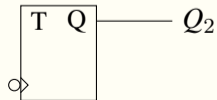
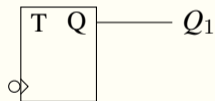
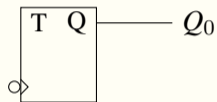
- 同期式 3 進カウンタ ($00 \leftarrow \rightarrow 01 \leftarrow \rightarrow 10 \leftarrow \rightarrow 00 \leftarrow \rightarrow \dots$)
 - 出力はカウント値 (00, 01, 10) とキャリー (桁溢れしたときに 1 になる)
 - UP/DOWN 切り替え入力 (0 のとき UP、1 のとき DOWN) つき。
リセットなし。
-
- 入力は UP/DOWN のみ。
 - カウント値出力は状態を直接出せばいいので考えなくて良い。キャリーのみ。
 - キャリーは $10 \rightarrow 00$ のときも $00 \rightarrow 10$ のときも発生するものとする。

参考: 状態遷移図 (こんなモノ、できるだけ見ない)

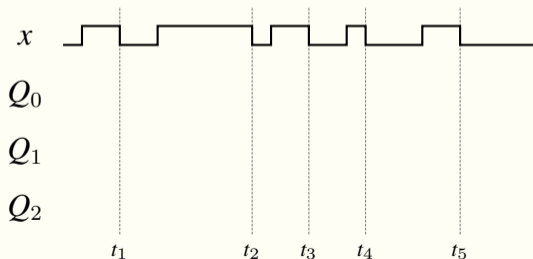


話は変わって、非同期式カウンタ

非同期式 2^n 進カウンタ (8進の例)

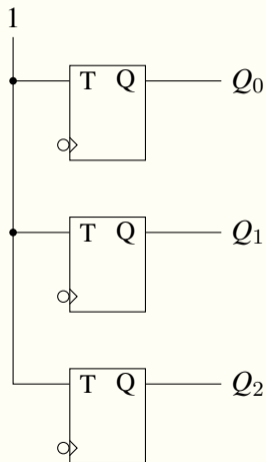


- **T-FF を数珠つなぎ**すれば良い。とても単純・お手軽。
- Q_0, Q_1, \dots の変化のタイミングが _____ ので注意。
ちなみに同期式の場合は _____。
- クロック信号を同期に使わない、いわば裏技的なテクニック。

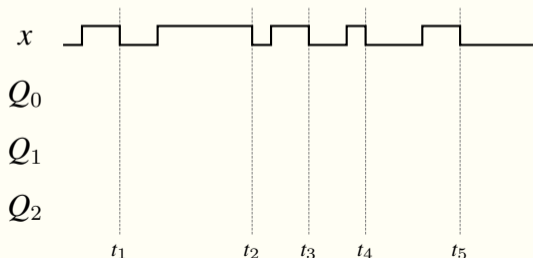


話は変わって、非同期式カウンタ

非同期式 2^n 進カウンタ (8進の例)

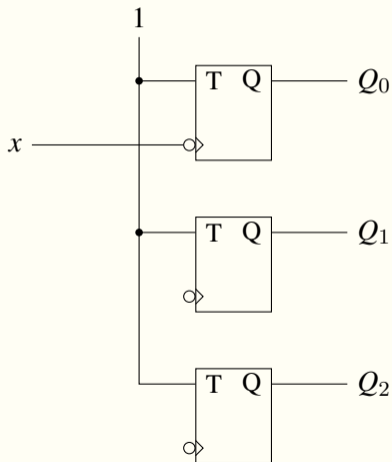


- **T-FF を数珠つなぎ**すれば良い。とても単純・お手軽。
- Q_0, Q_1, \dots の変化のタイミングが _____ ので注意。
ちなみに同期式の場合は _____。
- クロック信号を同期に使わない、いわば裏技的なテクニック。

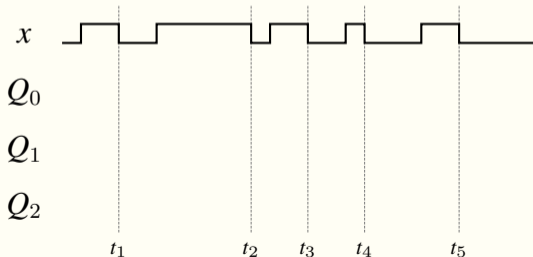


話は変わって、非同期式カウンタ

非同期式 2^n 進カウンタ (8進の例)

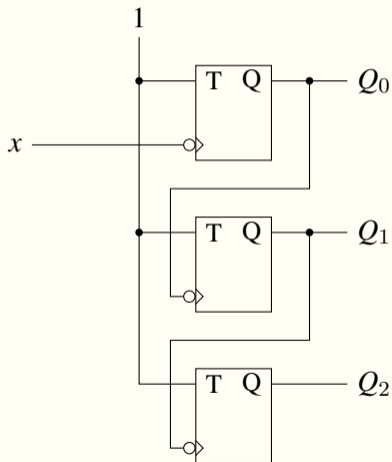


- **T-FF を数珠つなぎ**すれば良い。とても単純・お手軽。
- Q_0, Q_1, \dots の変化のタイミングが _____ ので注意。
ちなみに同期式の場合は _____。
- クロック信号を同期に使わない、いわば裏技的なテクニック。

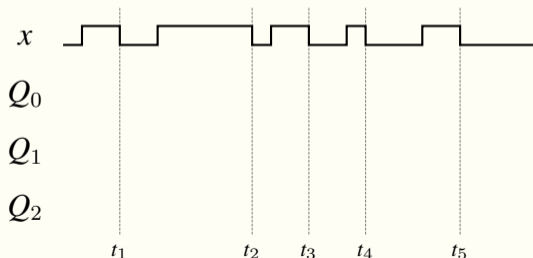


話は変わって、非同期式カウンタ

非同期式 2^n 進カウンタ (8進の例)

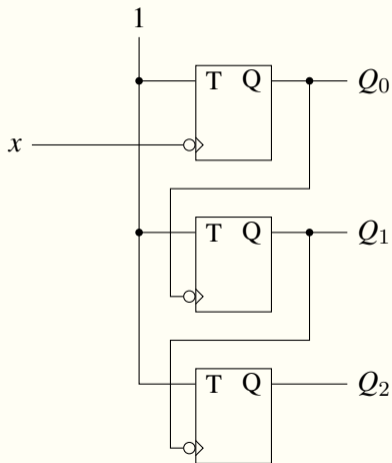


- **T-FF を数珠つなぎ**すれば良い。とても単純・お手軽。
- Q_0, Q_1, \dots の変化のタイミングが _____ ので注意。
ちなみに同期式の場合は _____。
- クロック信号を同期に使わない、いわば裏技的なテクニック。

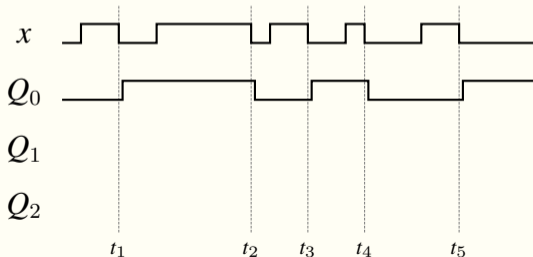


話は変わって、非同期式カウンタ

非同期式 2^n 進カウンタ (8進の例)

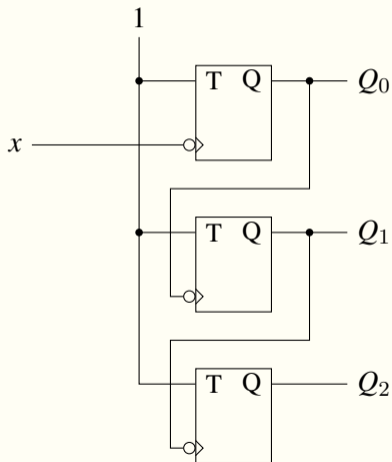


- **T-FF を数珠つなぎ**すれば良い。とても単純・お手軽。
- Q_0, Q_1, \dots の変化のタイミングが _____ ので注意。
ちなみに同期式の場合は _____。
- クロック信号を同期に使わない、いわば裏技的なテクニック。

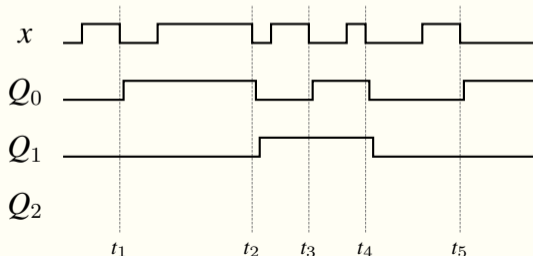


話は変わって、非同期式カウンタ

非同期式 2^n 進カウンタ (8進の例)

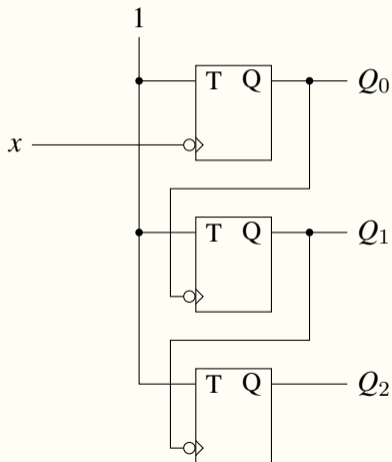


- **T-FF を数珠つなぎ**すれば良い。とても単純・お手軽。
- Q_0, Q_1, \dots の変化のタイミングが _____ ので注意。
ちなみに同期式の場合は _____。
- クロック信号を同期に使わない、いわば裏技的なテクニック。

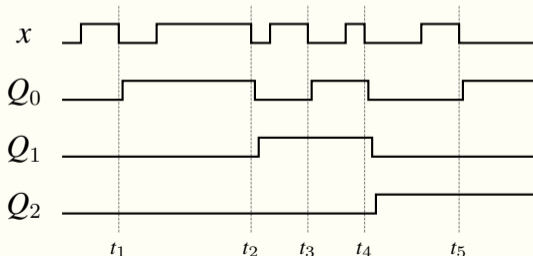


話は変わって、非同期式カウンタ

非同期式 2^n 進カウンタ (8進の例)

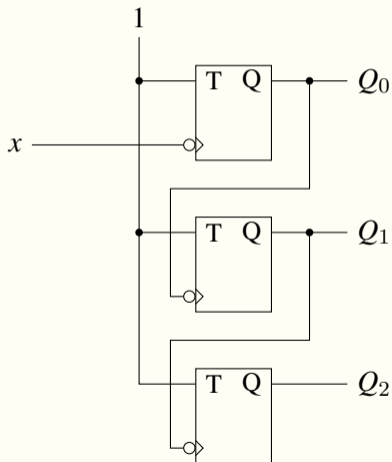


- **T-FF を数珠つなぎ**すれば良い。とても単純・お手軽。
- Q_0, Q_1, \dots の変化のタイミングが _____ ので注意。
ちなみに同期式の場合は _____。
- クロック信号を同期に使わない、いわば裏技的なテクニック。

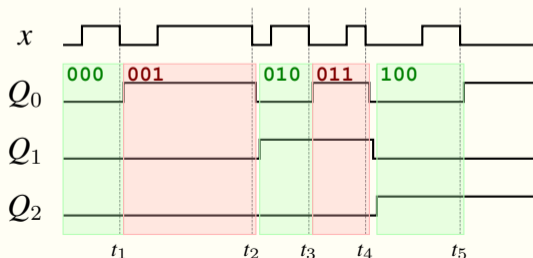


話は変わって、非同期式カウンタ

非同期式 2^n 進カウンタ (8進の例)

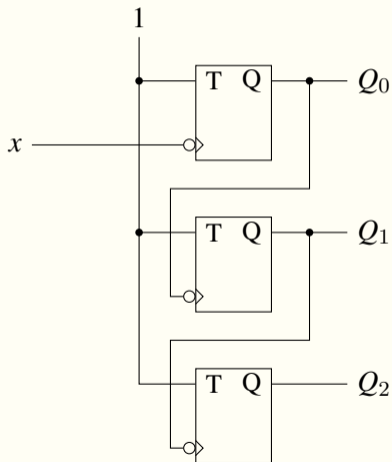


- **T-FF を数珠つなぎ**すれば良い。とても単純・お手軽。
- Q_0, Q_1, \dots の変化のタイミングが _____ ので注意。
ちなみに同期式の場合は _____。
- クロック信号を同期に使わない、いわば裏技的なテクニック。

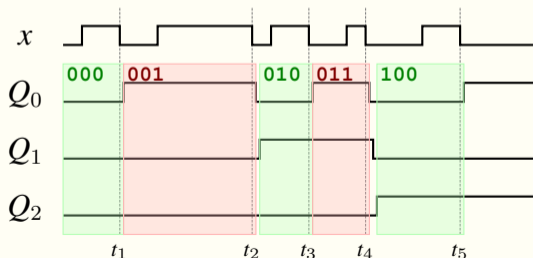


話は変わって、非同期式カウンタ

非同期式 2^n 進カウンタ (8進の例)

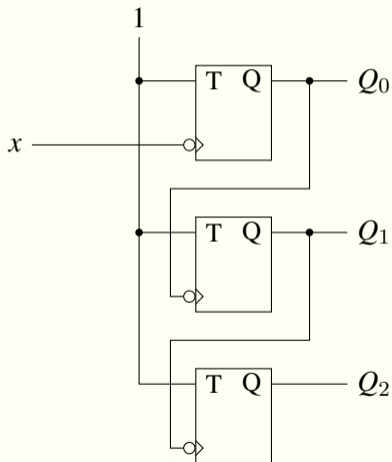


- **T-FF を数珠つなぎ**すれば良い。とても単純・お手軽。
- Q_0, Q_1, \dots の変化のタイミングが **ずれる** ので注意。
ちなみに同期式の場合は _____。
- クロック信号を同期に使わない、いわば裏技的なテクニック。

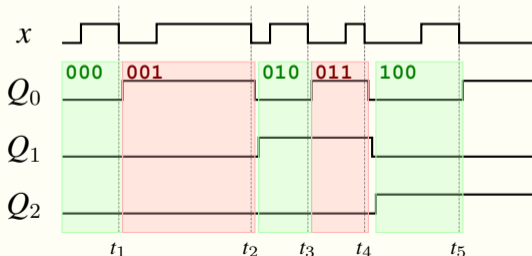


話は変わって、非同期式カウンタ

非同期式 2^n 進カウンタ (8進の例)



- **T-FF を数珠つなぎ**すれば良い。とても単純・お手軽。
- Q₀, Q₁, ... の変化のタイミングが **ずれる** ので注意。
ちなみに同期式の場合は ずれない。
- クロック信号を同期に使わない、いわば裏技的なテクニック。

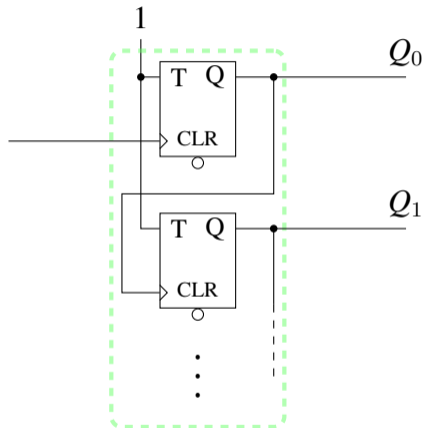


非同期式 N 進カウンタ

T-FF で作る単純な非同期式カウンタを少しだけアップグレード

リセット (クリア) 入力を使う方法

- まずは前ページのやり方で非同期式 N 進カウンタを作る。
- ここで実は市販の FF にはクリア入力 ($Q = 0$ にする機能) がある。
- そこで例えば 10 進カウンタを作りたいければ、 $(Q_3, Q_2, Q_1, Q_0) = (1, 0, 1, 0)$ になったらクリア入力を有効にするように組み合わせ論理を作る。単純。
- 【練習】 負論理のクリア入力のある T-FF を使ってこのやり方で 3 進カウンタを設計せよ。

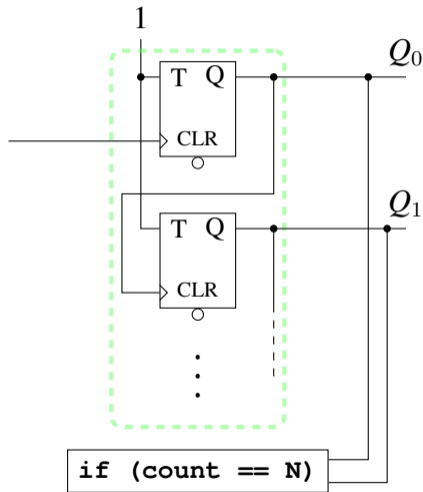


非同期式 N 進カウンタ

T-FF で作る単純な非同期式カウンタを少しだけアップグレード

リセット (クリア) 入力を使う方法

- まずは前ページのやり方で非同期式 N 進カウンタを作る。
- ここで実は市販の FF にはクリア入力 ($Q = 0$ にする機能) がある。
- そこで例えば 10 進カウンタを作りたいければ、 $(Q_3, Q_2, Q_1, Q_0) = (1, 0, 1, 0)$ になったらクリア入力を有効にするように組み合わせ論理を作る。単純。
- 【練習】 負論理のクリア入力のある T-FF を使ってこのやり方で 3 進カウンタを設計せよ。

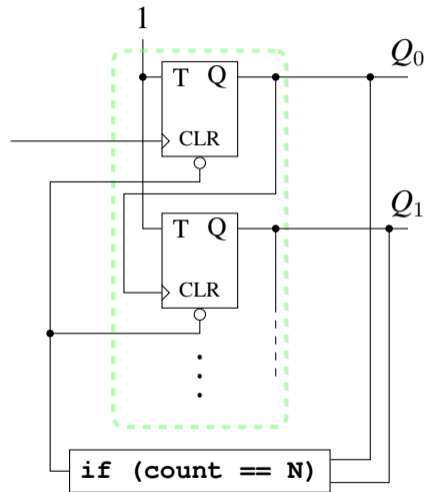


非同期式 N 進カウンタ

T-FF で作る単純な非同期式カウンタを少しだけアップグレード

リセット (クリア) 入力を使う方法

- まずは前ページのやり方で非同期式 N 進カウンタを作る。
- ここで実は市販の FF にはクリア入力 ($Q = 0$ にする機能) がある。
- そこで例えば 10 進カウンタを作りたいければ、 $(Q_3, Q_2, Q_1, Q_0) = (1, 0, 1, 0)$ になったらクリア入力を有効にするように組み合わせ論理を作る。単純。
- 【練習】 負論理のクリア入力のある T-FF を使ってこのやり方で 3 進カウンタを設計せよ。

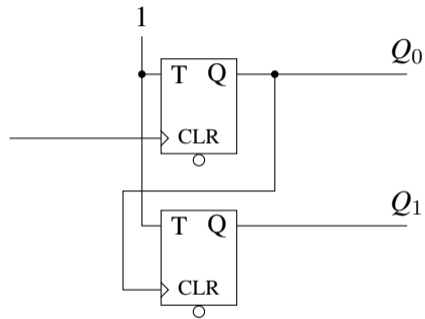


練習問題解答 (非同期式 3 進カウンタ)

こんなに簡単！

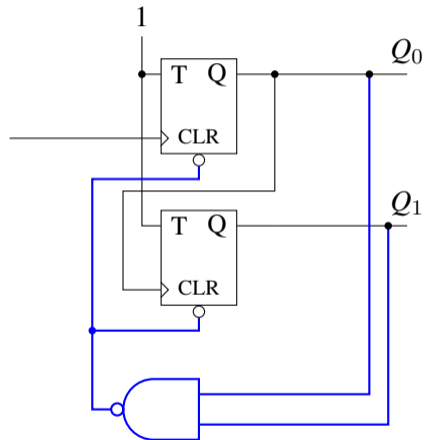
練習問題解答 (非同期式 3 進カウンタ)

こんなに簡単！



練習問題解答 (非同期式 3 進カウンタ)

こんなに簡単！



出席確認レポート課題 (次の月曜の 12 時締め切り)

問: 例 1 と同じ『リセットつき同期式 3 進 UP カウンタ』を T-FF で実装せよ。少なくとも状態遷移表から回路図までを示すこと。(T-FF を使うが非同期式ではないので間違わないように。)

提出は下記 URL の Google Forms。歪んでいない、開いた時に横倒しになっていない、コントラストが読むに耐えうる PDF で提出すること。

<https://forms.gle/9ruwtfJg5LQgQNpU7>

