

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>

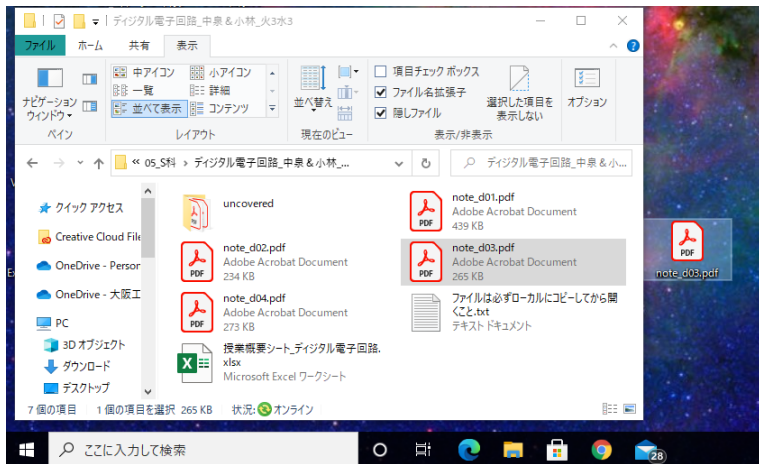


- **Slack**を補助的な連絡チャネルとする。必須ではないので使いたくなくれば使わなくてもいい。授業に関連したちょっとした（重要でない）追加説明をする。気楽な質問手段としても活用されたい。登録は大学の e-mail アドレスで行うこと。

<https://oitkobayashi.slack.com>

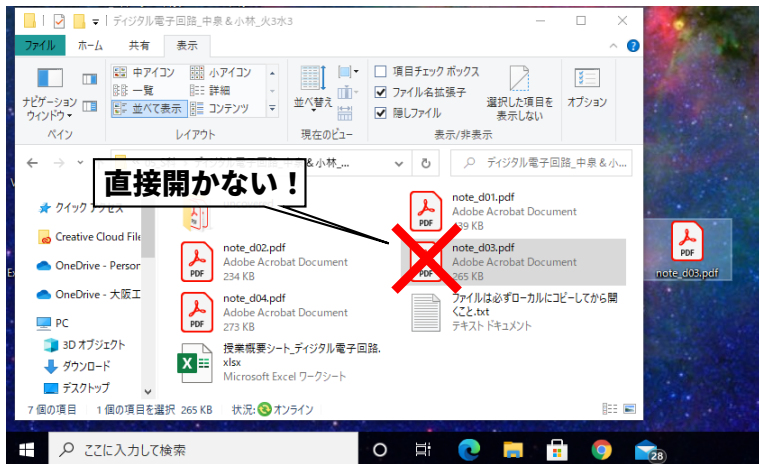
COMMON フォルダの注意事項 (全授業共通)

根源的に悪いのは Windows の仕様なのですが、ご協力ください。



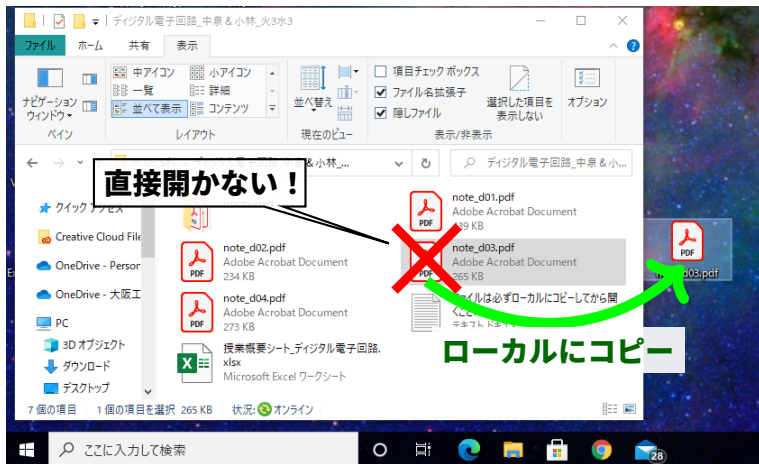
COMMON フォルダの注意事項 (全授業共通)

根源的に悪いのは Windows の仕様なのですが、ご協力ください。



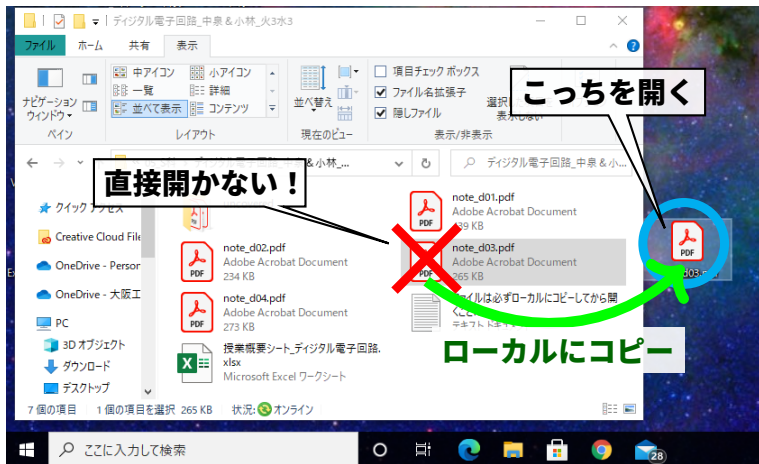
COMMON フォルダの注意事項 (全授業共通)

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

『FF』



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

7 of 14

a L^AT_EX + Beamer slideshow

これまでのあらすじ

- ① n 進数の考え方
- ② ブール代数
- ③ 論理の表現と簡単化
- ④ 組み合わせ回路における実用上のテクニック
- ⑤ 順序回路の定義・簡単化

これまでのあらすじ

- ① n 進数の考え方
- ② ブール代数
- ③ 論理の表現と簡単化
- ④ 組み合わせ回路における実用上のテクニック
- ⑤ 順序回路の定義・簡単化

今回・次回とその次で【順序回路の実現】

順序回路を作る

δ も ω も s と x の関数である点に注目

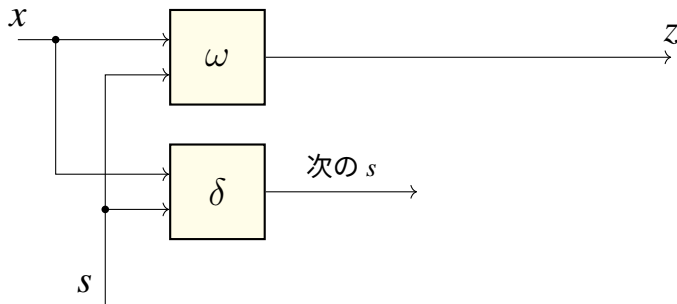
		$\delta(s, x)$ 次の状態		$\omega(s, x)$ 出力	
状態 s \ 入力 x		0	1	0	1
	00	00	01	0	0
	01	01	10	0	0
	10	10	00	0	1

この『状態遷移表』が、順序回路の動作を全て表している
ので、これを実現すればよい。つまり…

順序回路 $M = (S, X, Z, \delta, \omega)$ の実現

状態遷移関数 $\delta(s, x)$ と出力関数 $\omega(s, x)$ を実現

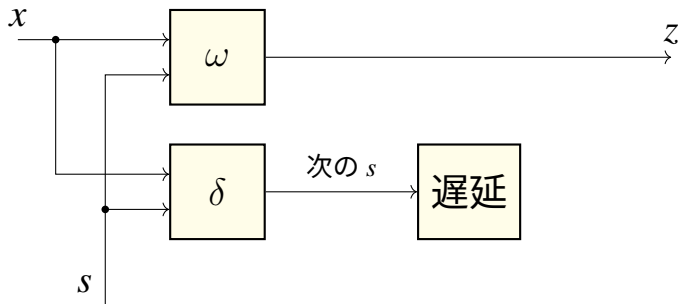
- ω は「入力と状態」から「出力」を決める単純な関数。ふつうの組み合わせ回路として作れば十分。
- δ は「入力と状態」から「次の状態」を決める関数。



順序回路 $M = (S, X, Z, \delta, \omega)$ の実現

状態遷移関数 $\delta(s, x)$ と **出力関数** $\omega(s, x)$ を実現 \Rightarrow 『遅延』が要る!

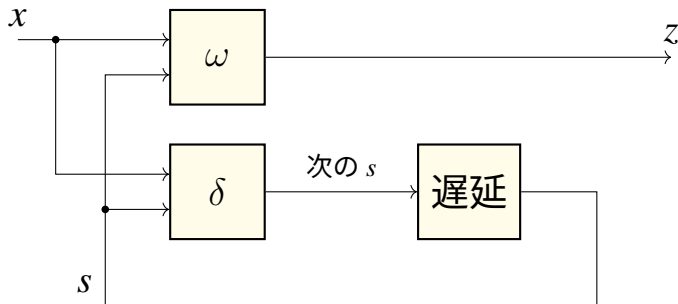
- ω は「入力と状態」から「出力」を決める単純な関数。ふつうの組み合わせ回路として作れば十分。
- δ は「入力と**状態**」から「**次の状態**」を決める関数。



順序回路 $M = (S, X, Z, \delta, \omega)$ の実現

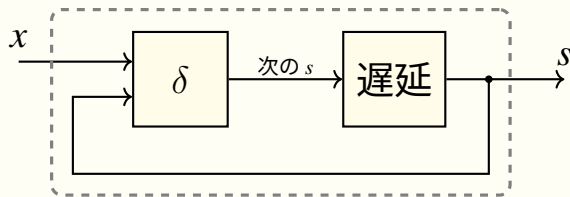
状態遷移関数 $\delta(s, x)$ と **出力関数** $\omega(s, x)$ を実現 \Rightarrow 『遅延』が要る!

- ω は「入力と状態」から「出力」を決める単純な関数。ふつうの組み合わせ回路として作れば十分。
- δ は「入力と**状態**」から「**次の状態**」を決める関数。



遅延つき状態遷移関数

欲しいものはこんなもの



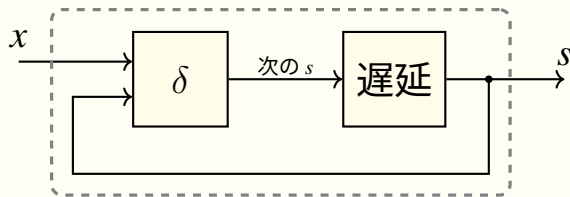
- 現在の状態 s を保持していて、
- x を入力すると、
- 『次』のステップに『次の状態』になる回路

実際に**図のような構成**である必要はない。外から見て、 x と s が**図のような動作**であれば良い。

を使うのが便利!

遅延つき状態遷移関数

欲しいものはこんなもの



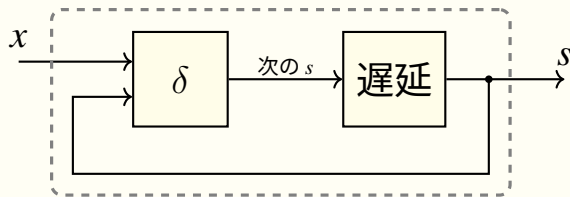
- 現在の状態 s を保持していて、
- x を入力すると、
- 『次』のステップに『次の状態』になる回路

実際に**図のような構成**である必要はない。外から見て、 x と s が**図のような動作**であれば良い。

FF()を使うのが便利!

遅延つき状態遷移関数

欲しいものはこんなもの



- 現在の状態 s を保持していて、
- x を入力すると、
- 『次』のステップに『次の状態』になる回路

実際に**図のような構成**である必要はない。外から見て、 x と s が**図のような動作**であれば良い。

FF(flip-flop)を使うのが便利!

今週・来週はその次への布石・伏兵です。

今週・来週はその次への布石・伏兵です。

- これから FF の勉強をする。

今週・来週はその次への布石・伏兵です。

- これから FF の勉強をする。
- 一見話がぶっ飛んでいるように感じると思う。

今週・来週はその次への布石・伏兵です。

- これから FF の勉強をする。
- 一見話がぶっ飛んでいるように感じると思う。
- しかも比較的面白くない。

今週・来週はその次への布石・伏兵です。

- これから FF の勉強をする。
- 一見話がぶっ飛んでいるように感じると思う。
- しかも比較的面白くない。
- しかし、すべては『遅延つき状態遷移関数』の実現のため、ひいては『順序回路の実現』のため、である。

今週・来週はその次への布石・伏兵です。

- これから FF の勉強をする。
- 一見話がぶっ飛んでいるように感じると思う。
- しかも比較的面白くない。
- しかし、すべては『遅延つき状態遷移関数』の実現のため、ひいては『順序回路の実現』のため、である。

全員、しっかりついてきて欲しい。

FF の発想

NOT を閉じてみた。



- **論理的に矛盾。**
- ちなみに実際に作ると “0” でも “1” でもない電圧値に落ち着く。

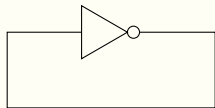
NOT を矛盾ないように閉じてみた。



- 矛盾せず、**安定。**
- Q は “0” か “1” の _____ を保持している！（一種のメモリ）
- ただし、 Q に値を設定はできない。

FF の発想

NOT を閉じてみた。



- **論理的に矛盾。**
- ちなみに実際に作ると “0” でも “1” でもない電圧値に落ち着く。

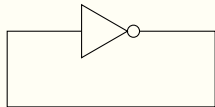
NOT を矛盾ないように閉じてみた。



- 矛盾せず、**安定。**
- Q は “0” か “1” の _____ を保持している！（一種のメモリ）
- ただし、 Q に値を設定はできない。

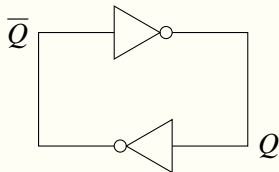
FF の発想

NOT を閉じてみた。



- **論理的に矛盾。**
- ちなみに実際に作ると“0”でも“1”でもない電圧値に落ち着く。

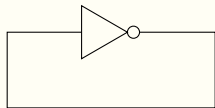
NOT を矛盾ないように閉じてみた。



- 矛盾せず、**安定。**
- Q は“0”か“1”の____を保持している！（一種のメモリ）
- ただし、 Q に値を設定はできない。

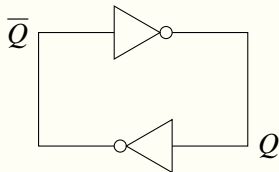
FF の発想

NOT を閉じてみた。



- **論理的に矛盾。**
- ちなみに実際に作ると“0”でも“1”でもない電圧値に落ち着く。

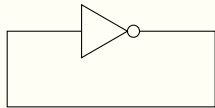
NOT を矛盾しないように閉じてみた。



- 矛盾せず、**安定。**
- Q は“0”か“1”の**状態**を保持している！（一種のメモリ）
- ただし、 Q に値を設定はできない。

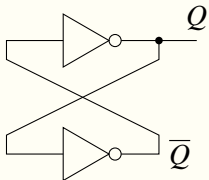
FF の発想

NOT を閉じてみた。



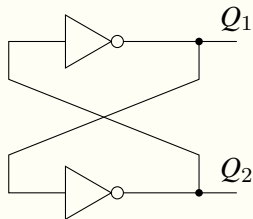
- **論理的に矛盾。**
- ちなみに実際に作ると“0”でも“1”でもない電圧値に落ち着く。

NOT を矛盾しないように閉じてみた。



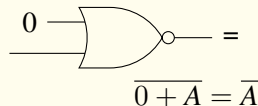
- 矛盾せず、**安定。**
- Q は“0”か“1”の**状態**を保持している！（一種のメモリ）
- ただし、 Q に値を設定はできない。

NOR を矛盾しないように閉じてみた



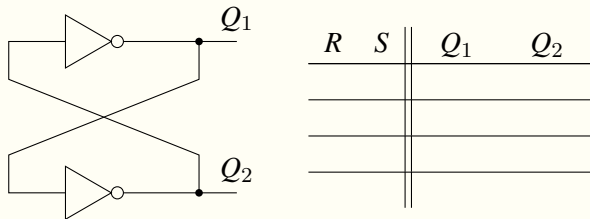
R	S	Q ₁	Q ₂

cf.

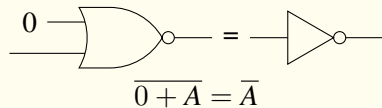


- $R = S = 0$ のとき、前ページの NOT による状態保持回路といっしょ
- $R = 1, S = 0$ のとき、 Q_1 を _____ する。
- $R = 0, S = 1$ のとき、 Q_1 を _____ する。
- $R = S = 1$ のとき、 $Q_1 = Q_2 =$ _____ となる。→ その次に $R = S = 0$ とすると論理的に

NOR を矛盾しないように閉じてみた

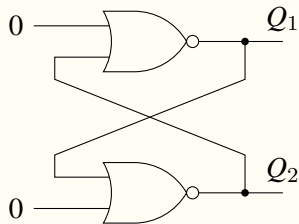


cf.



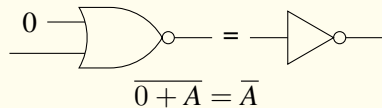
- $R = S = 0$ のとき、前ページの NOT による状態保持回路といっしょ
- $R = 1, S = 0$ のとき、 Q_1 を _____ する。
- $R = 0, S = 1$ のとき、 Q_1 を _____ する。
- $R = S = 1$ のとき、 $Q_1 = Q_2 =$ _____ となる。→ その次に $R = S = 0$ とすると論理的に

NOR を矛盾しないように閉じてみた



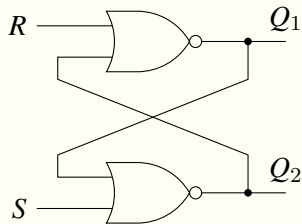
R	S	Q_1	Q_2
0	0		

cf.



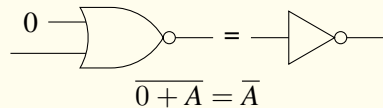
- $R = S = 0$ のとき、前ページの NOT による状態保持回路といっしょ
- $R = 1, S = 0$ のとき、 Q_1 を _____ する。
- $R = 0, S = 1$ のとき、 Q_1 を _____ する。
- $R = S = 1$ のとき、 $Q_1 = Q_2 =$ _____ となる。→ その次に $R = S = 0$ とすると論理的に

NOR を矛盾しないように閉じてみた



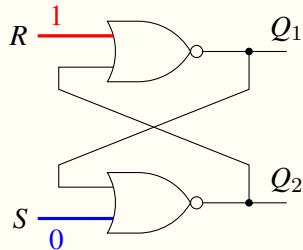
R	S	Q_1	Q_2
0	0	保持	保持

cf.



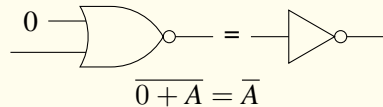
- $R = S = 0$ のとき、前ページの NOT による状態保持回路といっしょ
- $R = 1, S = 0$ のとき、 Q_1 を _____ する。
- $R = 0, S = 1$ のとき、 Q_1 を _____ する。
- $R = S = 1$ のとき、 $Q_1 = Q_2 =$ _____ となる。→ その次に $R = S = 0$ とすると論理的に

NOR を矛盾しないように閉じてみた



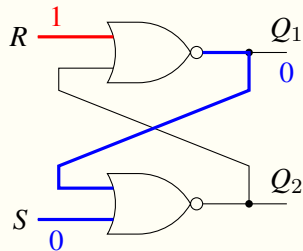
R	S	Q_1	Q_2
0	0	保持	保持
1	0		
	1		
1	1		

cf.



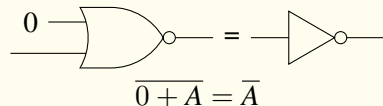
- $R = S = 0$ のとき、前ページの NOT による状態保持回路といっしょ
- $R = 1, S = 0$ のとき、 Q_1 を _____ する。
- $R = 0, S = 1$ のとき、 Q_1 を _____ する。
- $R = S = 1$ のとき、 $Q_1 = Q_2 =$ _____ となる。→ その次に $R = S = 0$ とすると論理的に

NOR を矛盾しないように閉じてみた



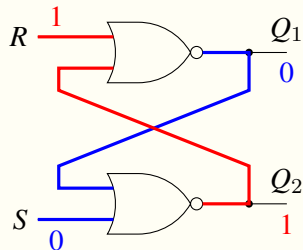
R	S	Q_1	Q_2
0	0	保持	保持
1	0		
	1		
1	1		

cf.



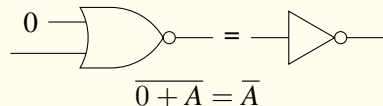
- $R = S = 0$ のとき、前ページの NOT による状態保持回路といっしょ
- $R = 1, S = 0$ のとき、 Q_1 を _____ する。
- $R = 0, S = 1$ のとき、 Q_1 を _____ する。
- $R = S = 1$ のとき、 $Q_1 = Q_2 =$ _____ となる。→ その次に $R = S = 0$ とすると論理的に

NOR を矛盾しないように閉じてみた



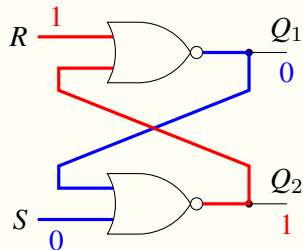
R	S	Q_1	Q_2
0	0	保持	保持
1	0		
	1		
	1		

cf.



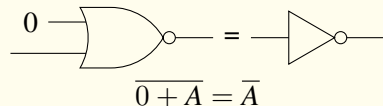
- $R = S = 0$ のとき、前ページの NOT による状態保持回路といっしょ
- $R = 1, S = 0$ のとき、 Q_1 を _____ する。
- $R = 0, S = 1$ のとき、 Q_1 を _____ する。
- $R = S = 1$ のとき、 $Q_1 = Q_2 =$ _____ となる。→ その次に $R = S = 0$ とすると論理的に

NOR を矛盾しないように閉じてみた



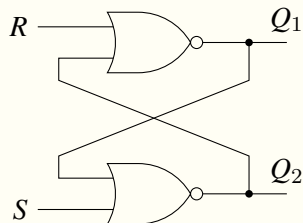
R	S	Q_1	Q_2
0	0	保持	保持
1	0	0	1

cf.



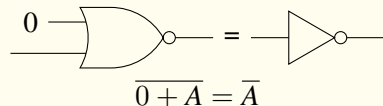
- $R = S = 0$ のとき、前ページの NOT による状態保持回路といっしょ
- $R = 1, S = 0$ のとき、 Q_1 を リセット する。
- $R = 0, S = 1$ のとき、 Q_1 を _____ する。
- $R = S = 1$ のとき、 $Q_1 = Q_2 =$ となる。→ その次に $R = S = 0$ とすると論理的に

NOR を矛盾しないように閉じてみた



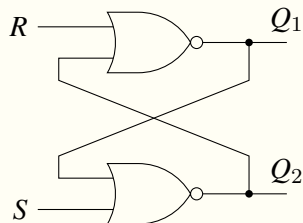
R	S	Q_1	Q_2
0	0	保持	保持
0	1	1	0
1	0	0	1

cf.



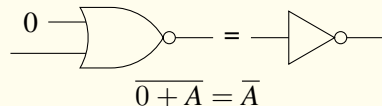
- $R = S = 0$ のとき、前ページの NOT による状態保持回路といっしょ
- $R = 1, S = 0$ のとき、 Q_1 を リセット する。
- $R = 0, S = 1$ のとき、 Q_1 を セット する。
- $R = S = 1$ のとき、 $Q_1 = Q_2 =$ となる。→ その次に $R = S = 0$ とすると論理的に

NOR を矛盾しないように閉じてみた



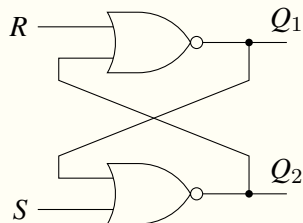
R	S	Q_1	Q_2
0	0	保持	保持
0	1	1	0
1	0	0	1
1	1	0	0

cf.



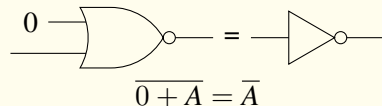
- $R = S = 0$ のとき、前ページの NOT による状態保持回路といっしょ
- $R = 1, S = 0$ のとき、 Q_1 を リセット する。
- $R = 0, S = 1$ のとき、 Q_1 を セット する。
- $R = S = 1$ のとき、 $Q_1 = Q_2 = 0$ となる。→ その次に $R = S = 0$ とすると論理的に

NOR を矛盾しないように閉じてみた



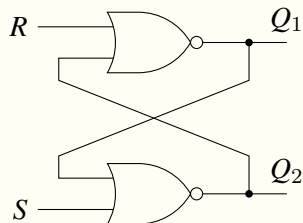
R	S	Q_1	Q_2
0	0	保持	保持
0	1	1	0
1	0	0	1
1	1	0	0

cf.



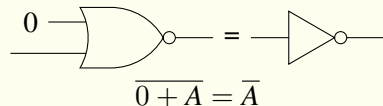
- $R = S = 0$ のとき、前ページの NOT による状態保持回路といっしょ
- $R = 1, S = 0$ のとき、 Q_1 を リセット する。
- $R = 0, S = 1$ のとき、 Q_1 を セット する。
- $R = S = 1$ のとき、 $Q_1 = Q_2 = 0$ となる。→ その次に $R = S = 0$ とすると論理的に矛盾 →

NOR を矛盾しないように閉じてみた



R	S	Q_1	Q_2
0	0	保持	保持
0	1	1	0
1	0	0	1
1	1	← 禁止入力	

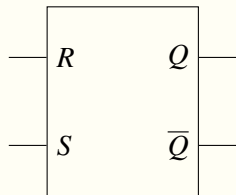
cf.



- $R = S = 0$ のとき、前ページの NOT による状態保持回路といっしょ
- $R = 1, S = 0$ のとき、 Q_1 を リセット する。
- $R = 0, S = 1$ のとき、 Q_1 を セット する。
- $R = S = 1$ のとき、 $Q_1 = Q_2 = 0$ となる。→ その次に $R = S = 0$ とすると論理的に矛盾 → **使用禁止**

RS-FF の表し方

回路記号と特性表



特性表		
R	S	Q'
0	0	Q
0	1	1
1	0	0
1	1	禁止

『次の Q 』を Q' と表している (Q^{k+1} のような表記もある。)

特性表を見ていると

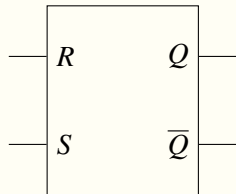
を求めたくなる。ついでに簡単化も。

特性方程式 (characteristic equation)

$$Q' =$$

RS-FF の表し方

回路記号と特性表



特性表		
R	S	Q'
0	0	Q
0	1	1
1	0	0
1	1	禁止

『次の Q 』を Q' と表している (Q^{k+1} のような表記もある。)

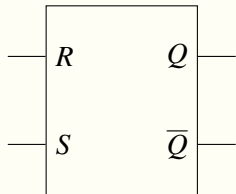
特性表を見ていると $Q'(R, S, Q)$ を求めたくなる。ついでに簡単化も。

特性方程式 (characteristic equation)

$$Q' =$$

RS-FF の表し方

回路記号と特性表



特性表		
R	S	Q'
0	0	Q
0	1	1
1	0	0
1	1	禁止

『次の Q 』を Q' と表している (Q^{k+1} のような表記もある。)

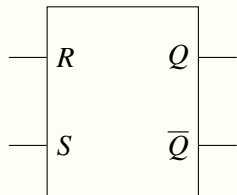
特性表を見ていると $Q'(R, S, Q)$ を求めたくなる。ついでに簡単化も。

特性方程式 (characteristic equation)

$$Q' = S + \bar{R} \cdot Q \text{ もしくは}$$

RS-FF の表し方

回路記号と特性表



特性表		
R	S	Q'
0	0	Q
0	1	1
1	0	0
1	1	禁止

『次の Q 』を Q' と表している (Q^{k+1} のような表記もある。)

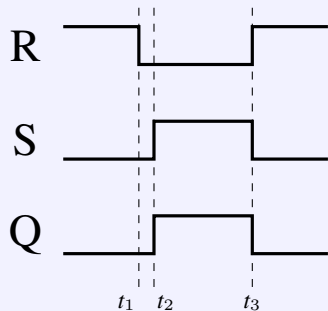
特性表を見ていると $Q'(R, S, Q)$ を求めたくなる。ついでに簡単化も。

特性方程式 (characteristic equation)

$$Q' = S + \bar{R} \cdot Q \text{ もしくは } \bar{R} \cdot (S + Q)$$

まちがい (?) 探し

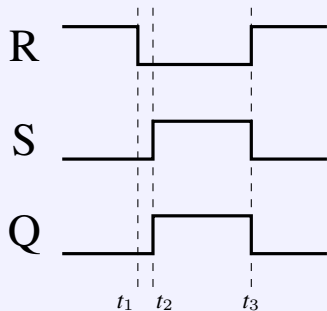
問:



左図は RS-FF の動作の時間変化を表した図（ ）である。問題点を答えよ。

まちがい (?) 探し

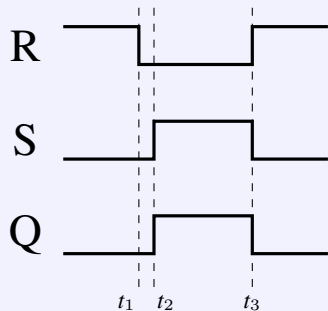
問:



左図は RS-FF の動作の時間変化を表した図（**タイミングチャート**）である。問題点を答えよ。

まちがい (?) 探し

問:

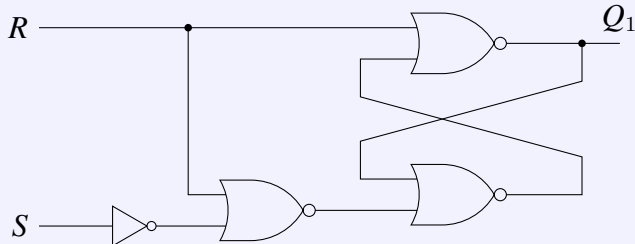


左図は RS-FF の動作の時間変化を表した図（**タイミングチャート**）である。問題点を答えよ。

➡ t_3 の瞬間が不安

RS-FF 改?

問: この《RS-FF 的な何か》は何か? その動作をひと言で表現し、特性方程式を求めよ。

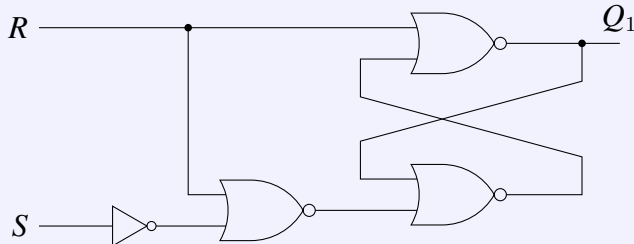


● $Q' =$

練習問題

RS-FF 改?

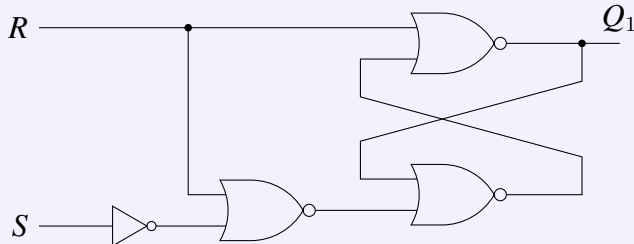
問: この《RS-FF 的な何か》は何か? その動作をひと言で表現し、特性方程式を求めよ。



- 禁止入力を回避 (リセット扱いに変更) する RS-FF
- $Q' =$

RS-FF 改?

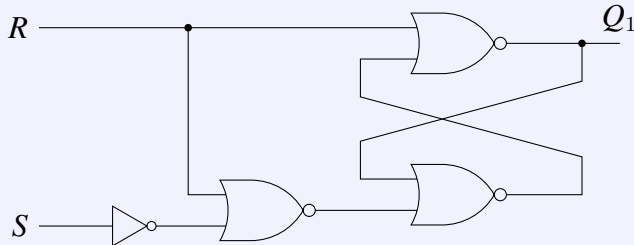
問: この《RS-FF 的な何か》は何か? その動作をひと言で表現し、特性方程式を求めよ。



- 禁止入力を回避 (リセット扱いに変更) する RS-FF
- $Q' = \bar{R} \cdot Q + \bar{R} \cdot S =$

RS-FF 改?

問: この《RS-FF 的な何か》は何か? その動作をひと言で表現し、特性方程式を求めよ。

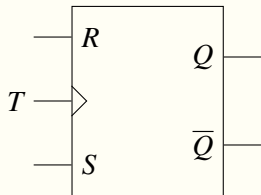


- 禁止入力を回避 (リセット扱いに変更) する RS-FF
- $Q' = \bar{R} \cdot Q + \bar{R} \cdot S = \bar{R} \cdot (S + Q)$

同期式回路とクロック

動作タイミングを同期信号 (**クロック**) に合わせたい!

同期の取り方、2つの流儀



(例) 同期式 RS-FF

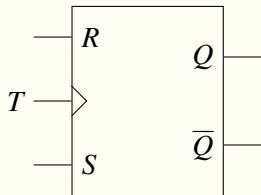
- **レベルトリガ**
クロックが _____ だけ入力有効になり、出力が変化する。 **そうでないときは保持。**
- **エッジトリガ**
クロックが _____ だけ入力有効になり、出力が変化する。 **そうでないときは保持。**
- クロック入力は“>”記号で書くことが多い。

種類	H level	L level	positive edge	negative edge
記号				

同期式回路とクロック

動作タイミングを同期信号 (**クロック**) に合わせたい!

同期の取り方、2つの流儀



(例) 同期式 RS-FF

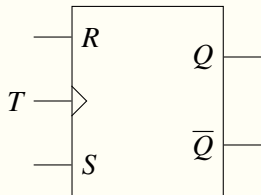
- **レベルトリガ**
クロックが **H(or L) レベルの間** だけ入力の有効になり、出力が変化する。**そうでないときは保持。**
- **エッジトリガ**
クロックが _____ だけ入力の有効になり、出力が変化する。**そうでないときは保持。**
- クロック入力は“>”記号で書くことが多い。

種類	H level	L level	positive edge	negative edge
記号				

同期式回路とクロック

動作タイミングを同期信号 (**クロック**) に合わせたい!

同期の取り方、2つの流儀



(例) 同期式 RS-FF

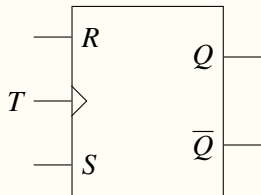
- **レベルトリガ**
クロックが **H(or L) レベルの間** だけ入力の有効になり、出力が変化する。**そうでないときは保持。**
- **エッジトリガ**
クロックが **立ち上 (or 下) がった瞬間** だけ入力の有効になり、出力が変化する。**そうでないときは保持。**
- クロック入力は“>”記号で書くことが多い。

種類	H level	L level	positive edge	negative edge
記号				

同期式回路とクロック

動作タイミングを同期信号 (**クロック**) に合わせたい!

同期の取り方、2つの流儀



(例) 同期式 RS-FF

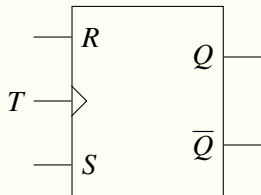
- **レベルトリガ**
クロックが **H(or L) レベルの間** だけ入力の有効になり、出力が変化する。**そうでないときは保持。**
- **エッジトリガ**
クロックが **立ち上 (or 下) がった瞬間** だけ入力の有効になり、出力が変化する。**そうでないときは保持。**
- クロック入力は“>”記号で書くことが多い。

種類	H level	L level	positive edge	negative edge
記号				

同期式回路とクロック

動作タイミングを同期信号 (**クロック**) に合わせたい!

同期の取り方、2つの流儀

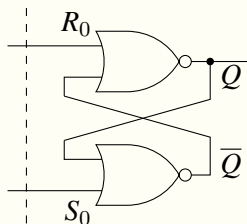


(例) 同期式 RS-FF

- **レベルトリガ**
クロックが **H(or L) レベルの間** だけ入力の有効になり、出力が変化する。**そうでないときは保持。**
- **エッジトリガ**
クロックが **立ち上 (or 下) がった瞬間** だけ入力の有効になり、出力が変化する。**そうでないときは保持。**
- クロック入力は“>”記号で書くことが多い。

種類	H level	L level	positive edge	negative edge
記号				

同期式 RS-FF(レベルトリガ) の内部構造



『状態の変化のタイミングを**同期信号 (T)**で制御できる RS-FF』

- 前段・後段に分けて考える。
- 後段は単なる非同期 RS-FF。
- 前段で $T=0$ のとき、後段は $R_0 = S_0 = 0$ つまり『 』状態になる。

問: T も入力と考え、同期式 RS-FF の特性方程式を求めよ。

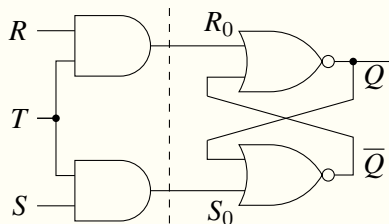
R	S	T	Q'	
0	0		Q	保持
0	1		1	セット
1	0		0	リセット
1	1		—	禁止

$$Q' =$$

もしくは

$$Q' =$$

同期式 RS-FF(レベルトリガ)の内部構造



『状態の変化のタイミングを**同期信号 (T)**で制御できる RS-FF』

- 前段・後段に分けて考える。
- 後段は単なる非同期 RS-FF。
- 前段で $T=0$ のとき、後段は $R_0 = S_0 = 0$ つまり『 』状態になる。

問: T も入力と考え、同期式 RS-FF の特性方程式を求めよ。

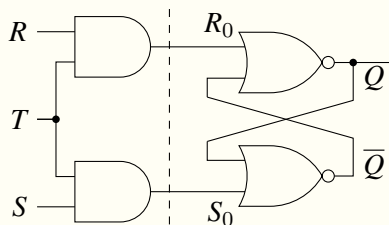
R	S	T	Q'	
0	0		Q	保持
0	1		1	セット
1	0		0	リセット
1	1		—	禁止

$$Q' =$$

もしくは

$$Q' =$$

同期式 RS-FF(レベルトリガ) の内部構造



『状態の変化のタイミングを**同期信号 (T)**で制御できる RS-FF』

- 前段・後段に分けて考える。
- 後段は単なる非同期 RS-FF。
- 前段で $T=0$ のとき、後段は $R_0 = S_0 = 0$ つまり『**保持**』状態になる。

問: T も入力と考え、同期式 RS-FF の特性方程式を求めよ。

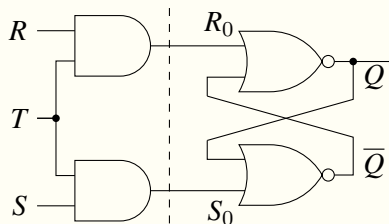
R	S	T	Q'	
0	0		Q	保持
0	1		1	セット
1	0		0	リセット
1	1		—	禁止

$$Q' =$$

もしくは

$$Q' =$$

同期式 RS-FF(レベルトリガ) の内部構造



『状態の変化のタイミングを**同期信号 (T)**で制御できる RS-FF』

- 前段・後段に分けて考える。
- 後段は単なる非同期 RS-FF。
- 前段で $T=0$ のとき、後段は $R_0 = S_0 = 0$ つまり『**保持**』状態になる。

問: T も入力と考え、同期式 RS-FF の特性方程式を求めよ。

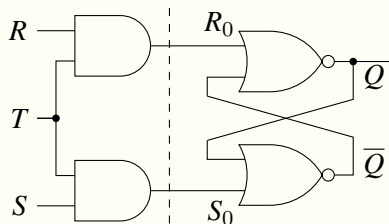
R	S	T	Q'	
0	0		Q	保持
0	1		1	セット
1	0		0	リセット
1	1		—	禁止

$$Q' = (S \cdot T) + (\bar{R} \cdot Q) + (\bar{T} \cdot Q)$$

もしくは

$$Q' =$$

同期式 RS-FF(レベルトリガ)の内部構造



『状態の変化のタイミングを**同期信号 (T)**で制御できる RS-FF』

- 前段・後段に分けて考える。
- 後段は単なる非同期 RS-FF。
- 前段で $T=0$ のとき、後段は $R_0 = S_0 = 0$ つまり『**保持**』状態になる。

問: T も入力と考え、同期式 RS-FF の特性方程式を求めよ。

R	S	T	Q'	
0	0		Q	保持
0	1		1	セット
1	0		0	リセット
1	1		—	禁止

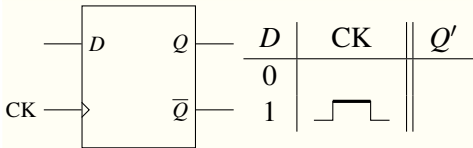
$$Q' = (S \cdot T) + (\bar{R} \cdot Q) + (\bar{T} \cdot Q)$$

もしくは

$$Q' = (\bar{R} + \bar{Q}) \cdot (S + Q) \cdot (T + Q)$$

同期式 D-FF(レベルトリガ)

D は Delay の D入力をそのまま (クロックに同期させて) 出力



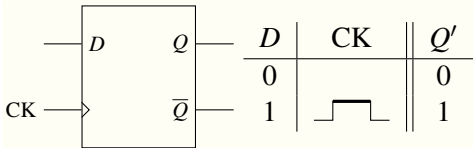
- 実際はこのようなレベルトリガのものは D-FF とは言わずに $\text{DFF}_{\text{level}}$ と言うことが多い。

問: D, CK を入力と考え、D-FF の特性方程式を求めよ

$$Q' =$$

同期式 D-FF(レベルトリガ)

D は Delay の D入力をそのまま (クロックに同期させて) 出力



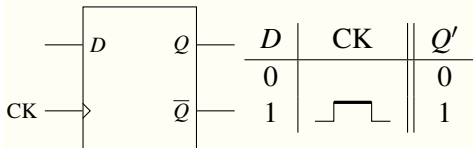
- 実際はこのようなレベルトリガのものは D-FF とは言わずに **ラッチ** と言うことが多い。

問: D, CK を入力と考え、D-FF の特性方程式を求めよ

$$Q' =$$

同期式 D-FF(レベルトリガ)

D は Delay の D入力をそのまま (クロックに同期させて) 出力



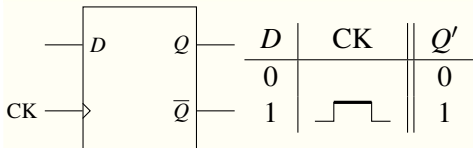
- 実際はこのようなレベルトリガのものは D-FF とは言わずに **D latch** と言うことが多い。

問: D , CK を入力と考え、D-FF の特性方程式を求めよ

$$Q' =$$
$$=$$

同期式 D-FF(レベルトリガ)

D は Delay の D入力をそのまま (クロックに同期させて) 出力



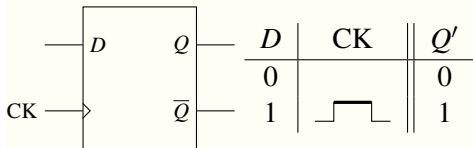
- 実際はこのようなレベルトリガのものは D-FF とは言わずに **D latch** と言うことが多い。

問: D , CK を入力と考え、D-FF の特性方程式を求めよ

$$Q' = D \cdot CK + \overline{CK} \cdot Q$$
$$=$$

同期式 D-FF(レベルトリガ)

D は Delay の D入力をそのまま (クロックに同期させて) 出力



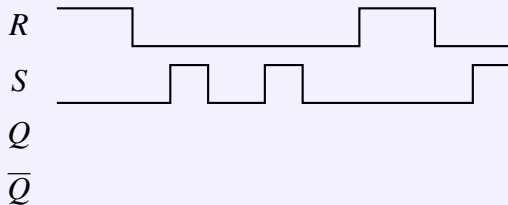
- 実際はこのようなレベルトリガのものは D-FF とは言わずに **D latch** と言うことが多い。

問: D , CK を入力と考え、D-FF の特性方程式を求めよ

$$\begin{aligned} Q' &= D \cdot CK + \overline{CK} \cdot Q \\ &= (D + \overline{CK}) \cdot (CK + Q) \end{aligned}$$

問

- ① RS-FF について、次のタイミングチャートを完成させよ。(Q, \bar{Q} について答えよという意味だがレポートとして提出する際は全部書くこと。)



- ② NAND で RS-FF を作れ。
- ③ 前問の「NAND 版 RS-FF」を改良して、「禁止入力を避ける RS-FF 的なもの」を作れ。

出席確認レポート課題 (次の月曜の 12 時締め切り)

前ページの問題 1 (タイミングチャート) を解け。2, 3 も余力があったらやること。

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>

