

本テキストや授業のビデオなどの電子ファイルを他人に転送したり、ネットへアップロードすることなどを禁止します。



論理設計

東野担当 2 回目

授業スライド

10月14日

基礎工学部情報科学科 東野輝夫





授業計画

2年生後期の実験と
関係が深いので
順番を入れ替えて
先に授業を実施します

- 授業計画：東野担当の授業計画は下記の通りです。
 1. ドントケアを含む論理関数の簡単化（6章）
 2. フリップフロップとレジスタ（10章）
 3. 同期式順序回路（Mealy型, Moore型順序回路）（11章）
 4. カルノー図を用いた論理関数の簡単化（1章から5章の復習）
 5. 組合せ論理回路設計、よく用いられる組み合わせ回路（7章, 8章）
 6. 加減算器とALU、カウンタ（9章, 13章）
 7. 演習
 8. 中間試験（1章～11章, 13章）
 9. 順序回路の簡単化（12章）
 10. ICを用いた順序回路の実現（15章）
 11. 演習
 12. CPUの設計（付録）
 13. CPUの設計, 演習
 14. 乗算器と除算器（14章）
 15. 期末試験（12章, 14章, 15章, 付録）



質問について

- メールで随時問い合わせや質問にお答えしますので、何かあれば、higashino@ist.osaka-u.ac.jp までメールで質問して下さい。
- また、時間を決めてZoomなどを用いて質問にお答えすることも可能ですので、まずはメールで疑問点や問い合わせ事項などを連絡して下さい。





お願い

本テキストや授業のビデオなどの電子ファイルを他人に転送したり、ネットへアップロードすることなどを禁止します。

著作権保護

- この授業のテキスト（教科書）や授業スライド、授業ビデオの著作権保護に努めて下さい。
- この授業のビデオやスナップショットを録画したり，それらを他の人に転送したり，インターネット上で公開したりすることを禁止します。
- この授業で利用するスライドにはオーム社の教科書の図などが含まれているので，著作権保護の観点から，この授業スライドの公開につながる行為は謹んでください。
- 来年度は CLE を使ったメディア授業でなく，対面の授業ができることを期待していますが，今年度の演習課題の解答が事前に公開されたりすると，来年度の授業で同じ演習課題が使えなくなり，授業テキストの大幅な修正が必要になるため，協力をお願いします。



1 回目の授業の レポート課題解答

基礎工学部情報科学科 東野輝夫





1 回目の授業のレポート課題

71頁の演習問題 1

レポート課題

(課題 1)

71頁の演習問題 1 : 次の論理関数 f のカルノー図を作成せよ

$$f = (x_3 \vee x_1 \vee x_0) \cdot (\bar{x}_3 \vee \bar{x}_2 \vee \bar{x}_1\bar{x}_0) \cdot (x_2 \vee x_0)$$

ただし, $g = \bar{x}_3x_1 \vee x_3\bar{x}_2x_0$ を 1 にする入力は禁止されている.

- 最終的な解答は教科書の末尾に書かれていますが, このレポート課題では答を得るまでの途中の計算の過程を明記してください. 関数 f や g のカルノー図がどう得られ, 最終的な関数 f のカルノー図がどう得られるかを明記ください.
- 提出先: 紙に書いた解答をスマホで写真を取るか, スキャナーなどで読み取り, pdf や jpeg, gif などの形式で電子化して CLE にアップして下さい.
- 締切: 10月13日(火) 23:59 (次の授業の前日迄).
- 2 回目の授業は 10月14日(水) の15:10 から実施します. 授業の音は当日の12:30以降に視聴できるようになります.



教科書の解答について

- 最終的な解答は教科書の末尾に書かれていると記載していますが、第1版第3刷以降は合っていると思いますが、第2刷までの教科書は解答が間違っている可能性があります。下記が最終的な解答になりますので、参考にしてください。

$x_3x_2 \backslash x_1x_0$		00	01	11	10
		00	01	11	10
00		1	X	X	
01		1	X	X	
11	1				
10		X	X		

f 但し g を 1 にする入力は禁止



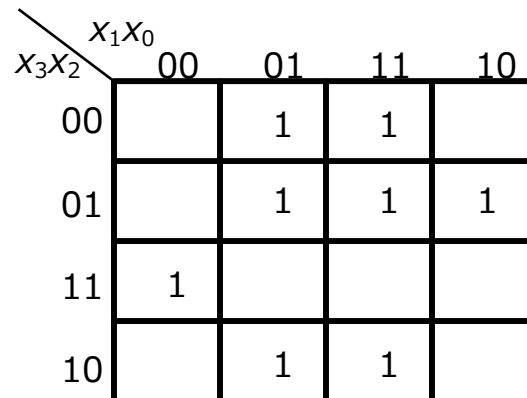
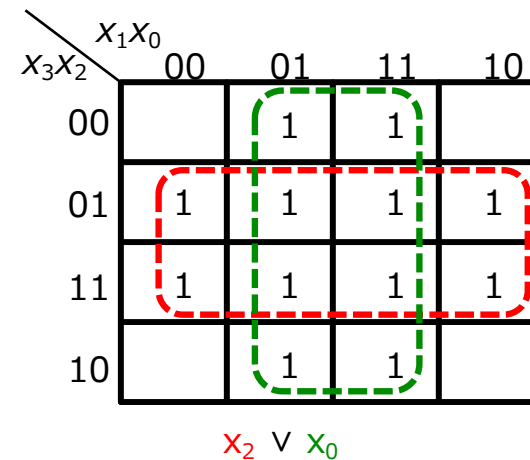
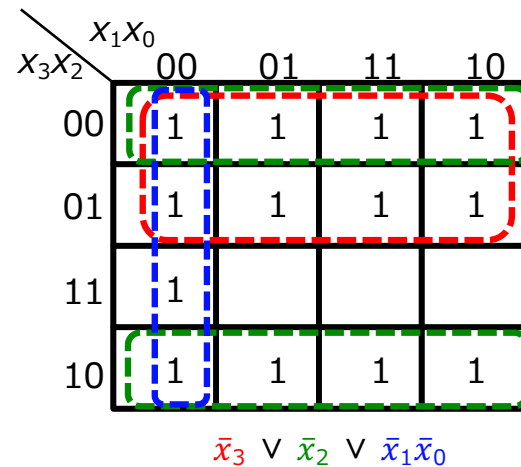
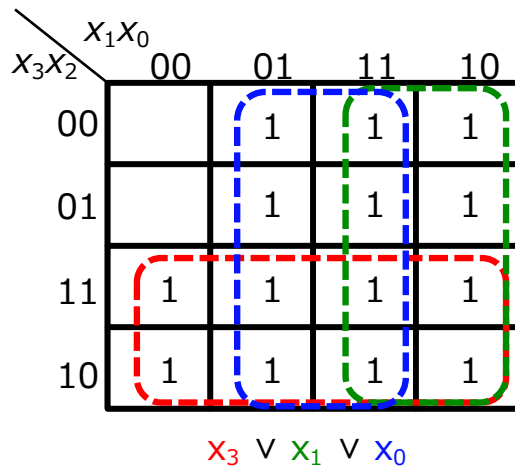
演習問題 1 解答

- 次の論理関数 f のカルノー図を作成せよ

$$f = (x_3 \vee x_1 \vee x_0) \cdot (\bar{x}_3 \vee \bar{x}_2 \vee \bar{x}_1 \bar{x}_0) \cdot (x_2 \vee x_0)$$

ただし, $g = \bar{x}_3 x_1 \vee x_3 \bar{x}_2 x_0$ を 1 にする入力は禁止されている.

(解答)



f



演習問題 1

- 次の論理関数 f のカルノー図を作成せよ

$$f = (x_3 \vee x_1 \vee x_0) \cdot (\bar{x}_3 \vee \bar{x}_2 \vee \bar{x}_1 \bar{x}_0) \cdot (x_2 \vee x_0)$$

ただし, $g = \bar{x}_3 x_1 \vee x_3 \bar{x}_2 x_0$ を 1 にする入力は禁止されている.

(解答)

$x_3 x_2 \backslash x_1 x_0$	00	01	11	10
00		1	1	
01		1	1	1
11	1			
10		1	1	

もとの f

$x_3 x_2 \backslash x_1 x_0$	00	01	11	10
00			1	1
01			1	1
11				
10		1	1	

g

$x_3 x_2 \backslash x_1 x_0$	00	01	11	10
00		1	X	X
01		1	X	X
11	1			
10		X	X	

f 但し g を 1 にする入力は禁止





第10章 フリップフロップと レジスタ



第 10 章 フリップフロップとレジスタ

- この章のねらい

10 章 フリップフロップとレジスタ

本章では、コンピュータで使われている記憶要素について説明する。まず、1 ビットの情報を記憶するための基本要素であるフリップフロップについて説明する。次に、複数ビットの情報をまとめて記憶するためのレジスタ、および複数のレジスタをまとめて扱うためのレジスタファイルについて説明する。最後にコンピュータの構成部品を接続するバスについて説明する。





フリップフロップの動作原理

- フリップフロップ (flip flop)
 - 1 ビットの情報を記憶するための基本的な論理回路
 - フリップフロップの動作原理は, 組合せ論理回路に帰還ループ (feed back loop) を加えて, 二つの安定状態を作り出すことにある.
- 図10・1(a) : スイッチ s_0 を閉じ, スイッチ s_1 を開くと, インバータの出力 u_1 の値は \bar{x} となり, 2段目のインバータの出力 y は x と等しくなる.
- 図10・1(b) : スイッチ s_0 を開き, スイッチ s_1 を閉じると, 入力 u_0 の値は y の値と等しくなり, 回路内部の信号の値はこれ以上変化せず出力も変化しない.
- 帰還ループ (feed back loop) を加えると, 0 または 1 の 1 ビットの情報を記憶できる.

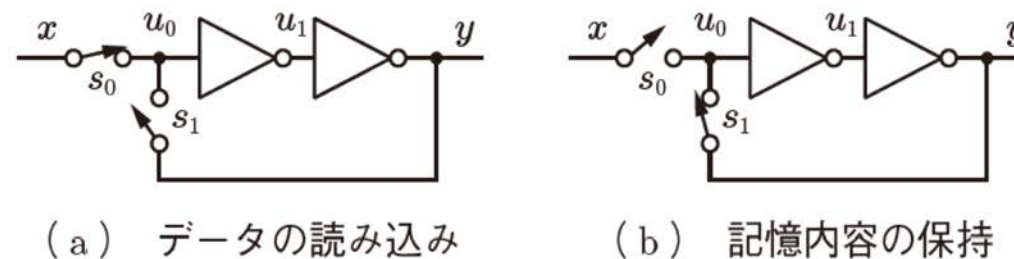


図 10・1

2 個のインバータをリング状に接続した回路





SRラッチ

- SRラッチ (SR latch) は、最も簡単な 1 ビットの記憶回路である
 - SR ラッチの S はセット (set) , R はリセット (reset) を表している. SRラッチの論理記号を 図10・2 に示す
- SR ラッチの動作を 表10・1 に示す
 - . 同表で Q および \bar{Q} は, SR ラッチの現在の出力を, また Q^+ および \bar{Q}^+ は状態遷移後の出力を表している
- 表10・1 の最後の行の出力の欄に書かれている記号 X は, 出力値が不定であることを表す
 - そのため, SR ラッチへの入力 $(S,R) = (1,1)$ は禁止入力

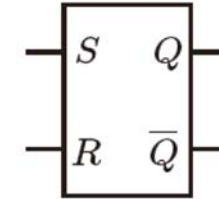


図 10・2 SR ラッチの論理記号

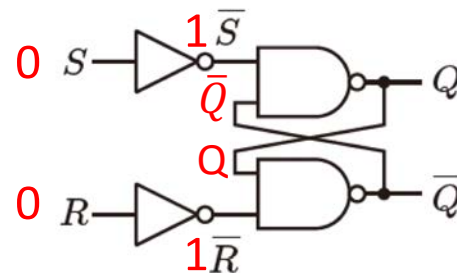
表 10・1 SR ラッチの動作表

入力		出力	
S	R	Q^+	\bar{Q}^+
0	0	Q	\bar{Q}
0	1	0	1
1	0	1	0
1	1	X	X

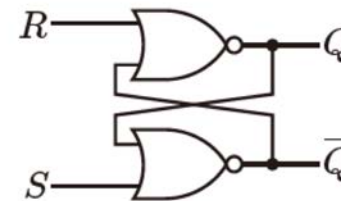


SRラッチの実装

- SRラッチ (SR latch) のNAND形での実装
 - NOT ゲートの遅延時間を T [s] とし, 2入力NAND ゲートおよび 2入力NOR ゲートの遅延時間とともに $1.2 T$ [s] と仮定する.
 - 入力が時刻 0 で $(S,R) = (0,0)$ に変化した場合, 上側の 2入力 NAND ゲートの入力は, 時刻 T に 1 および \bar{Q} となるので, 時刻 $2.2T$ の出力は $Q^+ = Q$ となる. また, 下側の 2入力NAND ゲートの入力は, 時刻 T で 1 および Q となるので, 時刻 $2.2T$ での出力は $\bar{Q}^+ = \bar{Q}$ となり, この状態で安定する.



(a) NAND 形

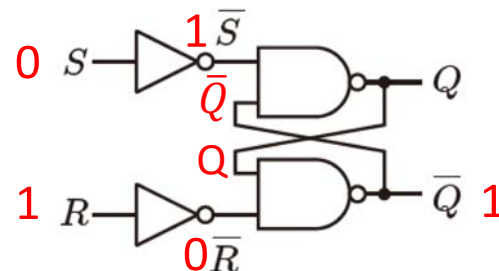


(b) NOR 形

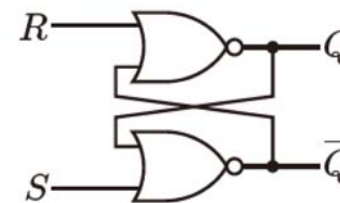


SRラッチの実装

- SRラッチ (SR latch) のNAND形での実装
 - 入力が時刻 0 で $(S, R) = (0, 1)$ に変化した場合, 上側の 2入力 NAND ゲートの入力は, 時刻 T に 1 および \bar{Q} となるので, 時刻 $2.2T$ での出力は $Q^+ = Q$ となる. しかし, 下側の 2入力 NAND ゲートの入力は, 時刻 T に 0 および Q となるので, 時刻 $2.2T$ での出力は $\bar{Q}^+ = 1$ となり,



(a) NAND 形

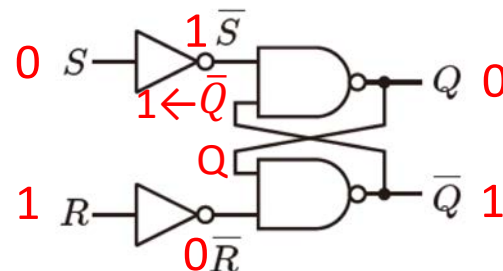


(b) NOR 形

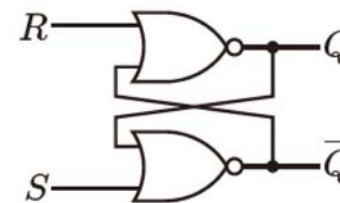


SRラッチの実装

- SRラッチ (SR latch) のNAND形での実装
 - 入力が時刻 0 で $(S, R) = (0, 1)$ に変化した場合, 上側の 2入力 NAND ゲートの入力は, 時刻 T に 1 および \bar{Q} となるので, 時刻 $2.2T$ での出力は $Q^+ = Q$ となる. しかし, 下側の 2入力 NAND ゲートの入力は, 時刻 T に 0 および Q となるので, 時刻 $2.2T$ での出力は $\bar{Q}^+ = 1$ となり, 時刻 $2.2T$ では上側の 2入力 NAND ゲートの入力がともに 1 となり, 時刻 $3.4T$ での出力は $Q^+ = 0$ となる. この場合, 時刻 $3.4T$ での出力は $(Q, \bar{Q}) = (0, 1)$ となり, その後, この値の組で安定する.



(a) NAND 形

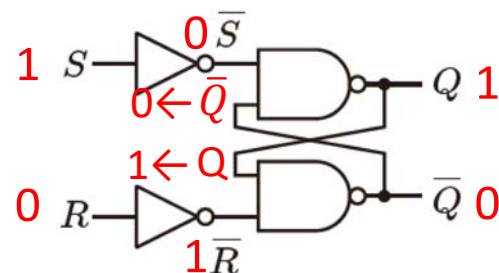


(b) NOR 形

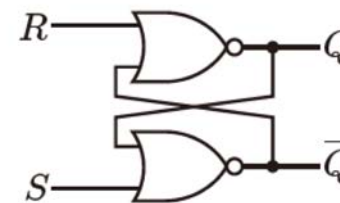


SRラッチの実装

- SRラッチ (SR latch) のNAND形での実装
 - 入力が時刻 0 で $(S, R) = (1, 0)$ に変化した場合, 上側の 2入力 NAND ゲートの入力は, 時刻 T に 0 および \bar{Q} となるので, 時刻 2.2T での出力は $Q^+ = 1$ となる. しかし, 下側の 2入力 NAND ゲートの入力は, 時刻 T に 1 および Q となるので, 時刻 2.2T での出力は $\bar{Q}^+ = \bar{Q}$ となる. 時刻 2.2T で Q の値が 1 に変わるので, 時刻 3.4T での出力は $\bar{Q}^+ = 0$ となり, その後, この値の組で安定する.



(a) NAND 形

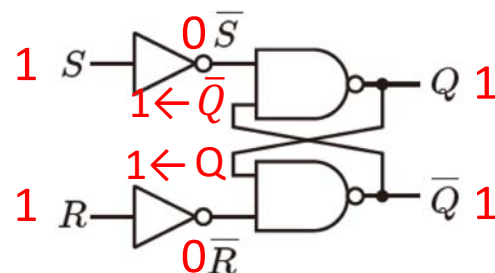


(b) NOR 形

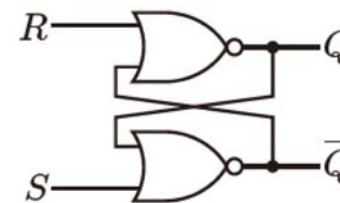


SRラッチの実装

- SRラッチ (SR latch) のNAND形での実装
 - 入力が時刻 0 で $(S,R) = (1,1)$ に変化した場合には, 時刻 T に 2 つの 2入力NAND ゲートの入力的一方がともに 0 となるので, 時刻 $2.2T$ での 2入力NAND ゲートの出力はともに 1 となる. 時刻 $2.2T$ で, 2つの 2入力NAND ゲートの入力 Q, \bar{Q} はともに 1 になるが, 出力はこれ以上変化しない. したがって, 時刻 $2.2T$ 以降の出力は $(Q, \bar{Q}) = (1,1)$ となる.



(a) NAND 形

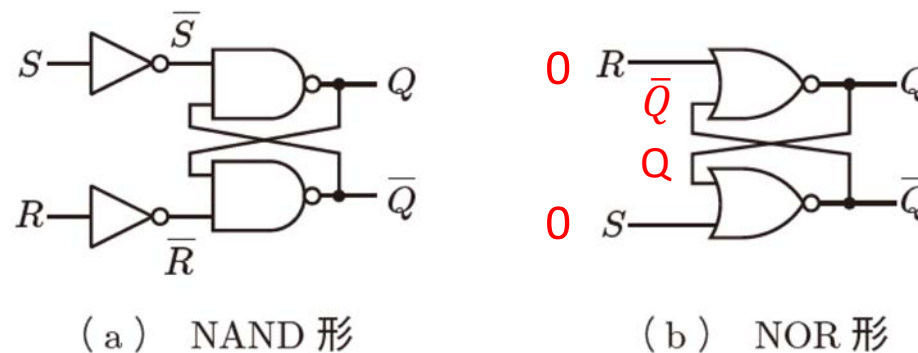


(b) NOR 形



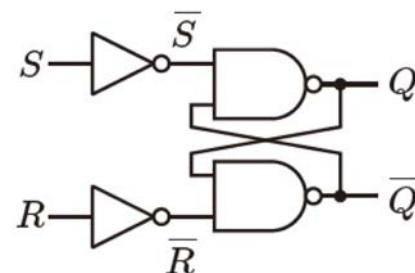
SRラッチの実装

- SRラッチ (SR latch) のNOR形での実装
 - 入力が時刻 0 で $(S,R) = (0,0)$ に変化した場合, 2つの 2 入力 NOR ゲートの出力は, 時刻 $1.2T$ に Q および \bar{Q} となる (NAND 形と入力 S と R の位置が逆になっている点に注意すること) .
また, 時刻 $1.2T$ 以降の出力は Q および \bar{Q} 共に変化せず, この状態で安定する.

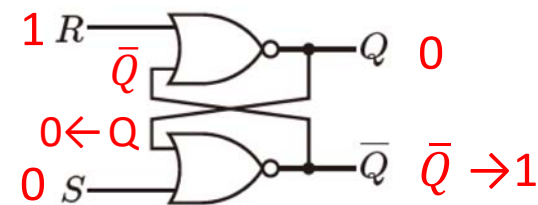


SRラッチの実装

- SRラッチ (SR latch) のNOR形での実装
 - 入力が時刻 0 で $(S, R) = (0, 1)$ に変化した場合, 2つの 2入力 NOR ゲートの出力は, 時刻 $1.2T$ に上から順に 0 および \bar{Q} となる (NAND形と入力 S, R の位置が逆になっている点に注意) .
 - しかし, 下側の 2入力NOR ゲートの入力は, 時刻 $1.2T$ に 0 となるので, 時刻 $2.4T$ での出力は $\bar{Q}^+ = 1$ となる. この場合, 時刻 $2.4T$ での出力は $(Q, \bar{Q}) = (0, 1)$ となり, その後, この値の組で安定する.



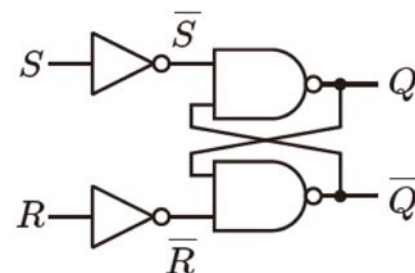
(a) NAND 形



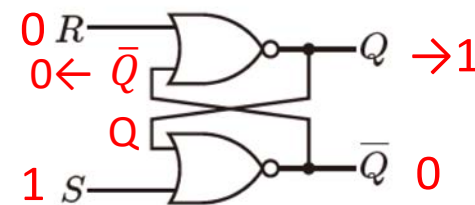
(b) NOR 形

SRラッチの実装

- SRラッチ (SR latch) のNOR形での実装
 - 入力が時刻 0 で $(S, R) = (1, 0)$ に変化した場合, 2つの 2入力 NOR ゲートの出力は, 時刻 $1.2T$ に上から順に Q および 0 となる (NAND形と入力 S, R の位置が逆になっている点に注意) .
 - しかし, 上側の 2入力NOR ゲートの入力は, 時刻 $1.2T$ に 0 となるので, 時刻 $2.4T$ での出力は $Q^+ = 1$ となる. この場合, 時刻 $2.4T$ での出力は $(Q, \bar{Q}) = (1, 0)$ となり, その後, この値の組で安定する.



(a) NAND 形



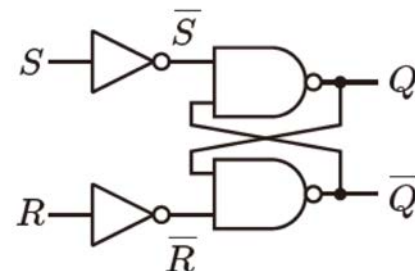
(b) NOR 形



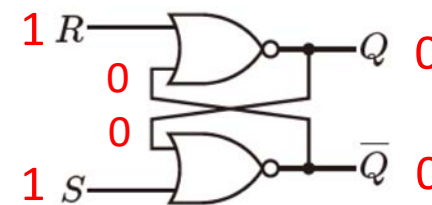


SRラッチの実装

- SRラッチ (SR latch) のNOR形での実装
 - 入力が時刻 0 で $(S,R) = (1,1)$ に変化した場合, 2つの 2入力 NOR ゲートの出力は, 時刻 $1.2T$ に上から順に 0 および 0 となる (NAND形と入力 S, R の位置が逆になっている点に注意) .
 - 上側の 2入力NOR ゲートの出力は時刻 $1.2T$ に 0 となるが, 時刻 $2.4T$ での出力は $Q^+ = 0$ のままとなり, その後 $(Q,\bar{Q}) = (0,0)$ で安定する.
 - NAND形の実装とNOR形の実装で $(S,R) = (1,1)$ の出力が異なるので, **$(S,R) = (1,1)$ は禁止入力**となっている



(a) NAND 形



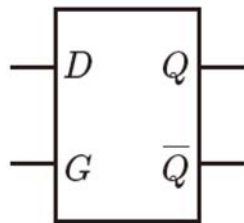
(b) NOR 形



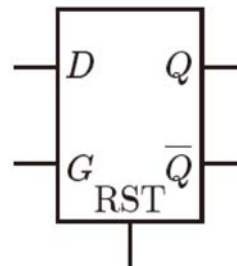


Dラッチ

- SRラッチは構造が簡単でハードウェア量も少ないという利点はあるが、前節で述べた「禁止入力」が存在するという欠点もあり、取扱いに注意が必要
- そこで、SRラッチの入力側に論理回路を追加して取扱いを容易にした回路が D ラッチ (D latch) である
- Dラッチの論理記号を 図10・4 に示す．また Dラッチの動作を表10・2 に示す



(a) D ラッチの論理記号

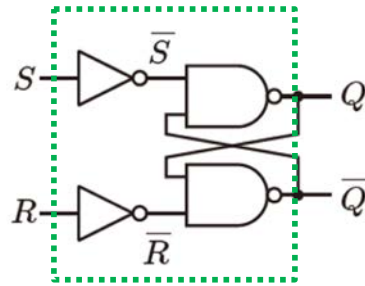


(b) リセット機能付き
D ラッチの論理記号

表 10・2 D ラッチの動作表

入力		出力	
G	D	Q^+	\overline{Q}^+
0	0	Q	\overline{Q}
0	1	Q	\overline{Q}
1	0	0	1
1	1	1	0

図 10・4 D ラッチの論理記号

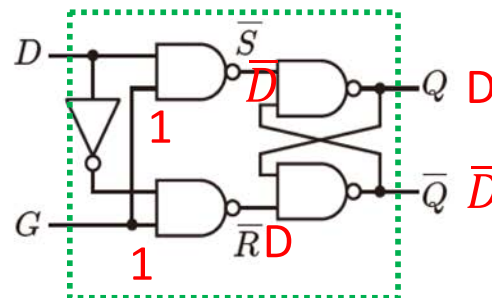


Dラッチ

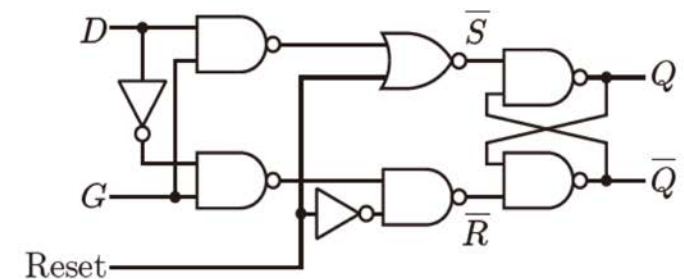
- Dラッチの論理回路を 図10・5 に示す. 同図(a) で, Dラッチの入力信号のうち, G は制御信号であり, D がデータである.
- Dラッチの後段は 2 個のNAND の帰還をもつ縦続接続になっており, これはSRラッチの後段と同じ構成になっている. この後段の部分回路の入力信号を \bar{S} および \bar{R} とすると, $G = 1$ のときには, $\bar{S} = \bar{D}$, $\bar{R} = D$ ($S = D, R = \bar{D}$) となるので出力信号 Q には D の値がそのまま出力される.
- G の値が 0 になると, $\bar{S} = 1$, $\bar{R} = 1$ ($S = 0, R = 0$) となるので, SRラッチの入力が (0, 0) の場合と同様にフリップフロップの内部状態がそのまま保持される.

表 10・2 D ラッチの動作表

入力		出力	
G	D	Q^+	\bar{Q}^+
0	0	Q	\bar{Q}
0	1	Q	\bar{Q}
1	0	0	1
1	1	1	0



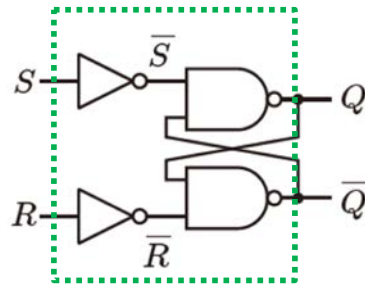
(a) リセット機能をもたない D ラッチ



(b) リセット機能をもつ D ラッチ

図 10・5 D ラッチの論理回路図



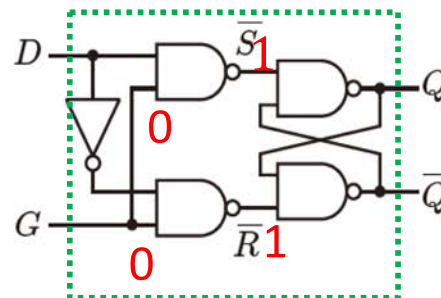


Dラッチ

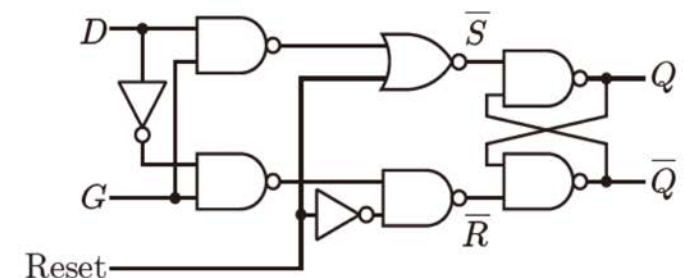
- Dラッチの論理回路を 図10・5 に示す. 同図(a) で, Dラッチの入力信号のうち, G は制御信号であり, D がデータである.
- Dラッチの後段は 2 個のNAND の帰還をもつ縦続接続になっており, これはSRラッチの後段と同じ構成になっている. この後段の部分回路の入力信号を \bar{S} および \bar{R} とすると, $G = 1$ のときには, $\bar{S} = \bar{D}$, $\bar{R} = D$ ($S = D$, $R = \bar{D}$) となるので出力信号 Q には D の値がそのまま出力される.
- G の値が 0 になると, $\bar{S} = 1$, $\bar{R} = 1$ ($S = 0$, $R = 0$) となるので, SRラッチの入力が (0, 0) の場合と同様にフリップフロップの内部状態がそのまま保持される.

表 10・2 D ラッチの動作表

入力		出力	
G	D	Q^+	\bar{Q}^+
0	0	Q	\bar{Q}
0	1	Q	\bar{Q}
1	0	0	1
1	1	1	0



(a) リセット機能をもたない D ラッチ



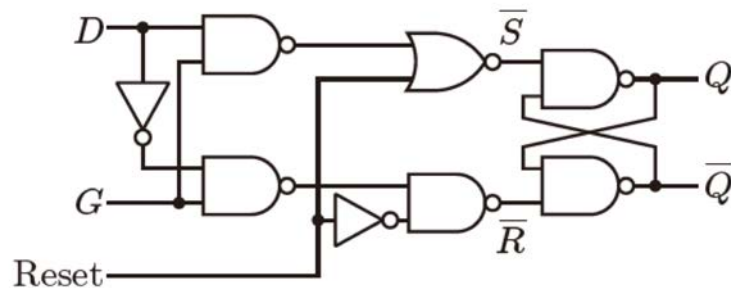
(b) リセット機能をもつ D ラッチ

図 10・5 D ラッチの論理回路図



Dラッチ

- Dラッチでは SR ラッチの禁止入力に対応する $\bar{S} = 0, \bar{R} = 0$ という状態は取らず, 禁止入力が存在しないので, SR ラッチよりも取扱いが容易である.
- 図10・5 (b) はリセット信号をもつ Dラッチである. この回路では, リセット信号 (RST) が 1 になるとD ラッチの内部状態を強制的にリセットして 0 が出力される. リセット信号をもつ Dラッチの動作を表10・3 に示す.



(b) リセット機能をもつ
D ラッチ

図 10・5 D ラッチの論理回路図

表 10・3 リセット信号をもつ D ラッチの動作表

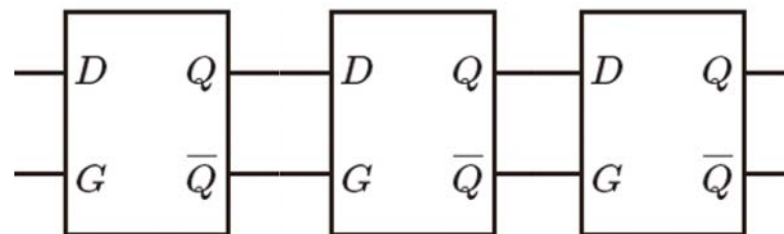
入力			出力	
RST	G	D	Q^+	\bar{Q}^+
1	—	—	0	1
0	0	0	Q	\bar{Q}
0	0	1	Q	\bar{Q}
0	1	0	0	1
0	1	1	1	0





Dラッチの接続の問題点

- Dラッチの接続の問題点
 - Dラッチでは、制御信号G の値によって出力の制御を行っていた。
 - Dラッチで $G = 1$ の場合、入力 D の値が変化すると、入力の変化がそのまま出力信号 Q に出力されてしまうという性質がある。
 - そのため、複数の Dラッチが下図のように直列に接続され、同じタイミングの信号 G を用いて制御すると、初段の Dラッチの入力信号の変化が最終段の Dラッチの出力にまで伝搬してしまう。
 - したがって、このままでは演算がクロックに同期してパイプライン的に実行される同期回路 (synchronous circuit) が正しく動作しない。





Dフリップフロップ

- 表10・4 で \uparrow はクロックの立上りエッジ (rising edge) を, \downarrow はクロックの立下りエッジ (falling edge) をそれぞれ表している.
- クロックの立上り (立下り) とは, クロック信号の値が 0 から 1 (1 から 0) に変化する瞬間を表している.
- Dフリップフロップは, クロック信号が立上った瞬間の入力の値を取り込んで出力する.
- また出力値はクロックの立下りエッジおよびクロックの値が 0 または 1 で安定している場合は影響を受けず, 次のクロックの立上りまで変化しない.

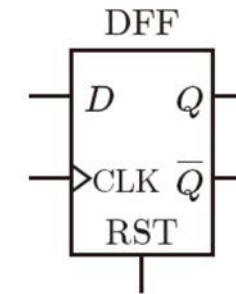
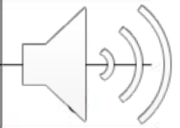


図 10・6 D フリップフロップの論理記号

表 10・4 D フリップフロップの動作表

入力			出力	
RST	CLK	D	Q^+	\overline{Q}^+
1	—	—	0	1
0	0	—	Q	\overline{Q}
0	1	—	Q	\overline{Q}
0	\downarrow	—	Q	\overline{Q}
0	\uparrow	0	0	
0	\uparrow	1	1	





Dフリップフロップの実現法

- Dフリップフロップは, 図10・7 のように 2 個の Dラッチを組み合わせることによって実現できる.
- 同図で前段の Dラッチはマスタラッチ (master latch), 後段の Dラッチはスレーブラッチ (slave latch) と呼ばれ, それぞれクロックの値が 0 の場合および 1 の場合にデータの取込みが行われる. この回路は, マスター・スレーブ形 (master-slave type) フリップフロップと呼ばれる.

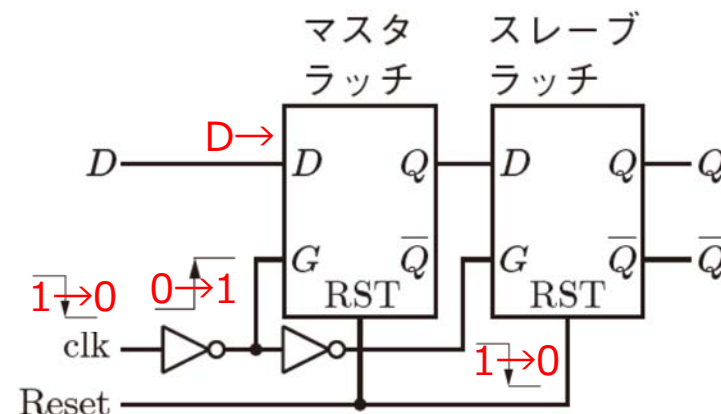


図 10・7

D フリップフロップの実現方法





Dフリップフロップの実現法

- Dフリップフロップは, 図10・7 のように 2 個の Dラッチを組み合わせることによって実現できる.
- 同図で前段の Dラッチはマスタラッチ (master latch), 後段の Dラッチはスレーブラッチ (slave latch) と呼ばれ, それぞれクロックの値が 0 の場合および 1 の場合にデータの取込みが行われる. この回路は, マスター・スレーブ形 (master-slave type) フリップフロップと呼ばれる.

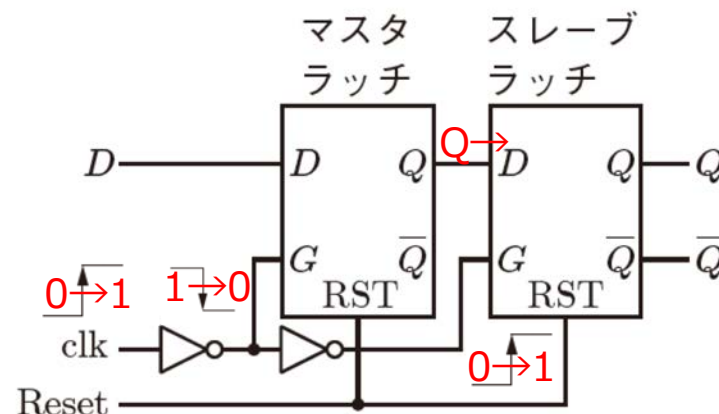


図 10・7

D フリップフロップの実現方法



レジスタ

- レジスタ (register) は、複数のビットから構成される情報をまとめて記憶するための論理回路である。レジスタの論理記号を 図10・8 に示す。n ビットのレジスタは、n 個の D フリップフロップと **セクタ (マルチプレクサ)** を 図10・9 のように接続することによって実現できる (同図には $n = 4$ の場合を示す)。この回路では、制御信号 Load の値が 1 の場合に入力 $D = (D_3, D_2, D_1, D_0)$ がレジスタ中の 4 個の D フリップフロップの中に記録される。また、Load の値が 0 の場合には現在のレジスタの値がそのまま保持される。レジスタの値は常に出力信号 $Q = (Q_3, Q_2, Q_1, Q_0)$ に出力されている。

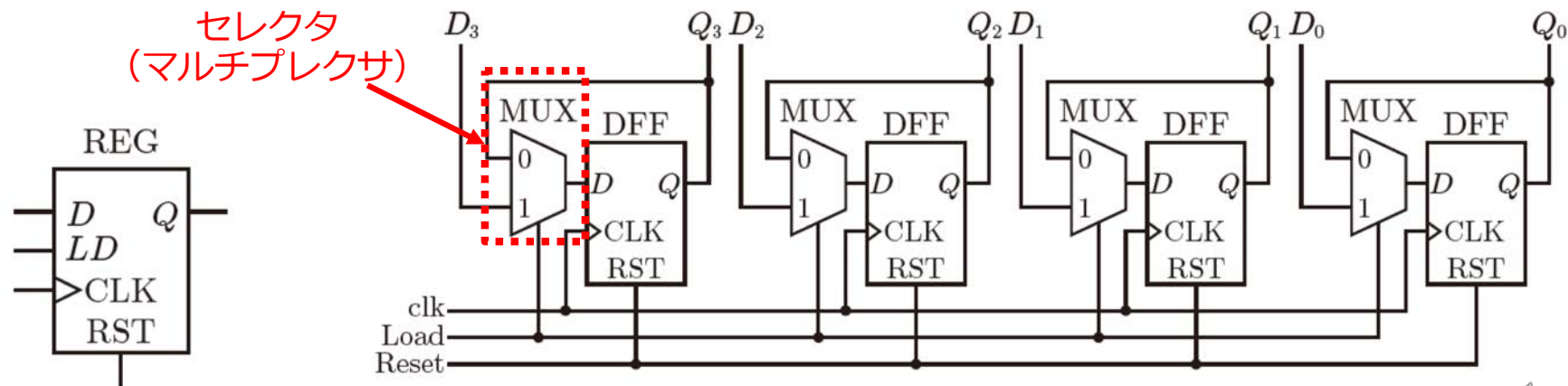


図 10・8 レジスタの論理記号

図 10・9 レジスタの実装方法



セクタ (マルチプレクサ)

- **マルチプレクサ (multiplexer)** は, n ビットの制御入力 s_{n-1}, \dots, s_0 の値によって, 2^n 個のデータの入力 i_{2^n-1}, \dots, i_0 のいずれかを出力 o に出力する回路である.
- マルチプレクサは **セクタ (selector)** と呼ばれる.
- $n = 2$ とし, $(s_1, s_0) = (0, 0), (0, 1), (1, 0), (1, 1)$ のときにそれぞれ i_0, i_1, i_2, i_3 を o に出力するマルチプレクサの出力 o の積和形論理式は (図8・3)

$$o = i_0 \bar{s}_1 \bar{s}_0 \vee i_1 \bar{s}_1 s_0 \vee i_2 s_1 \bar{s}_0 \vee i_3 s_1 s_0$$

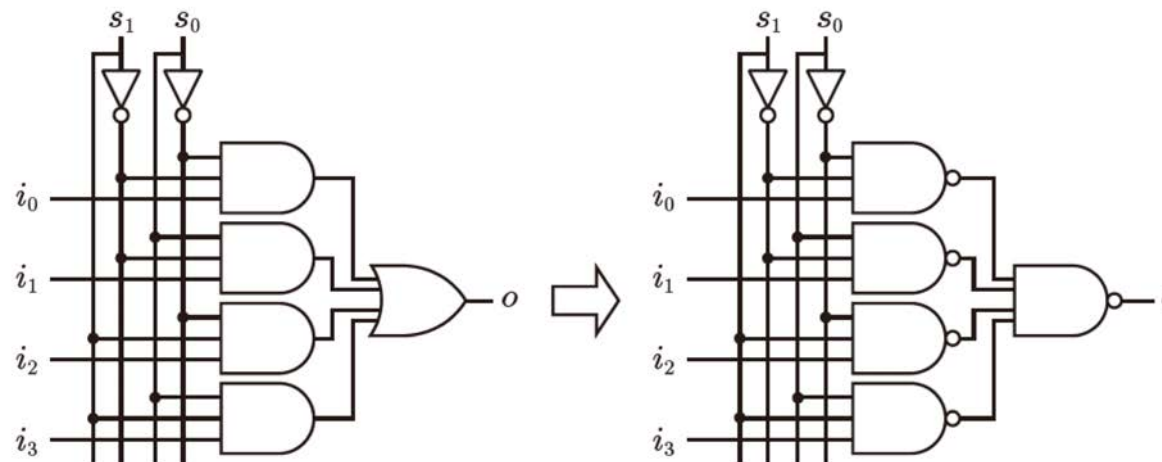


図 8・3 4 入力マルチプレクサの回路図



レジスタ

- レジスタ (register) は、複数のビットから構成される情報をまとめて記憶するための論理回路である。レジスタの論理記号を 図10・8 に示す。n ビットのレジスタは、n 個の D フリップフロップと**セクタ (マルチプレクサ)**を 図10・9 のように接続することによって実現できる (同図には $n = 4$ の場合を示す)。この回路では、制御信号 Load の値が 1 の場合に入力 $D = (D_3, D_2, D_1, D_0)$ がレジスタ中の 4 個の D フリップフロップの中に記録される。また、Load の値が 0 の場合には現在のレジスタの値がそのまま保持される。レジスタの値は常に出力信号 $Q = (Q_3, Q_2, Q_1, Q_0)$ に出力されている。

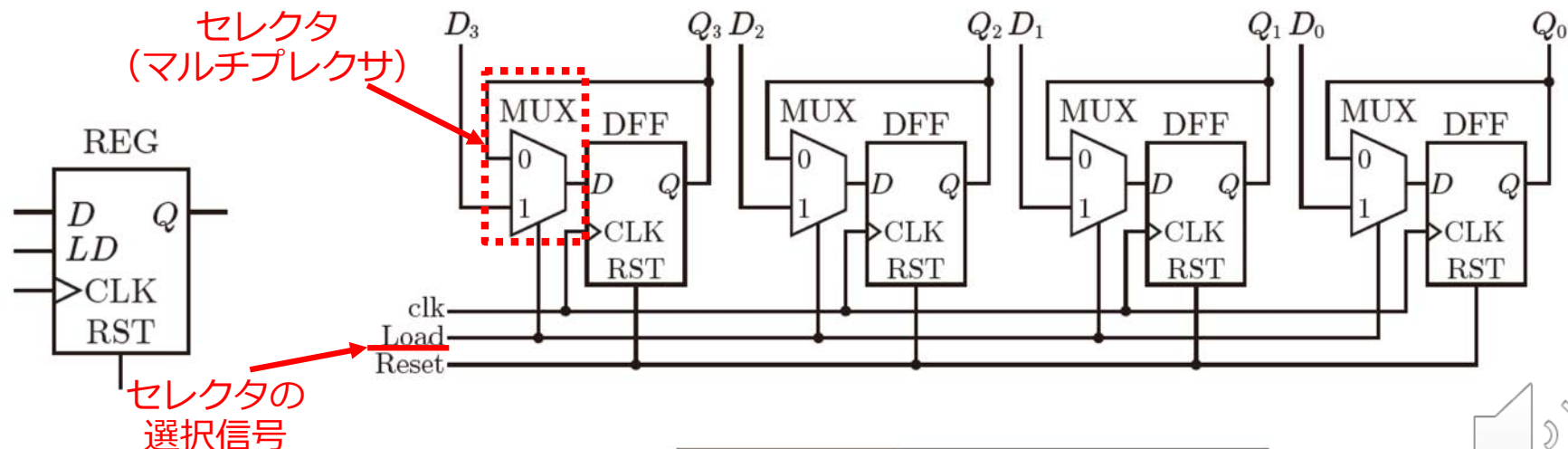


図 10・8 レジスタの論理記号

図 10・9 レジスタの実装方法



バス



- コンピュータの内部には、プロセッサ、レジスタ、メモリ、周辺回路などのさまざまなモジュールが存在する。あるモジュールから他のモジュールにデータを転送するには、モジュールの間にデータ転送路を設ける必要がある。
- データ転送はプロセッサとメモリもしくは周辺回路の間で行われる。このような場合には、複数の周辺回路からプロセッサへのデータ転送要求があった場合に、その中から一つだけを選択して転送する仕組みが必要となる。このようなデータ転送路を**共有バス (common bus)** と呼ぶ。

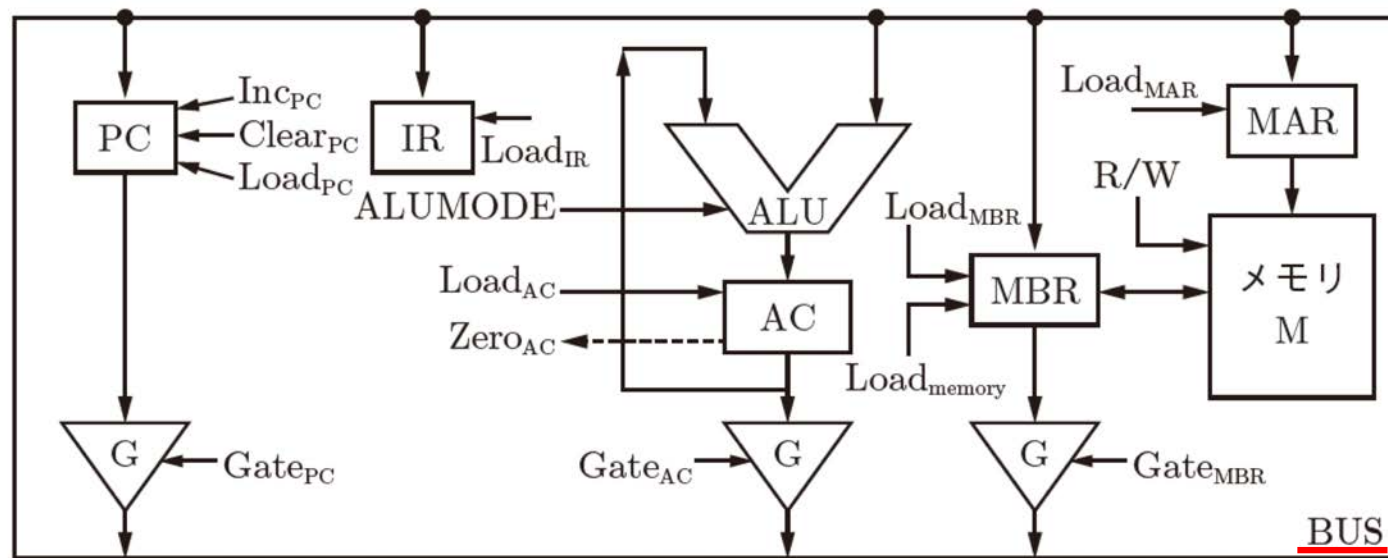


図 A・5

BUS を用いた CPU の回路図



バス

- マルチプレクサを用いたバスの実装方法
 - マルチプレクサを用いたバスの実装方法を 図10・12 に示す. この方法では, 1本のバスに対して一つのバスアービタと一つの n 入力マルチプレクサを用いることによりデータ転送路が実装できる.

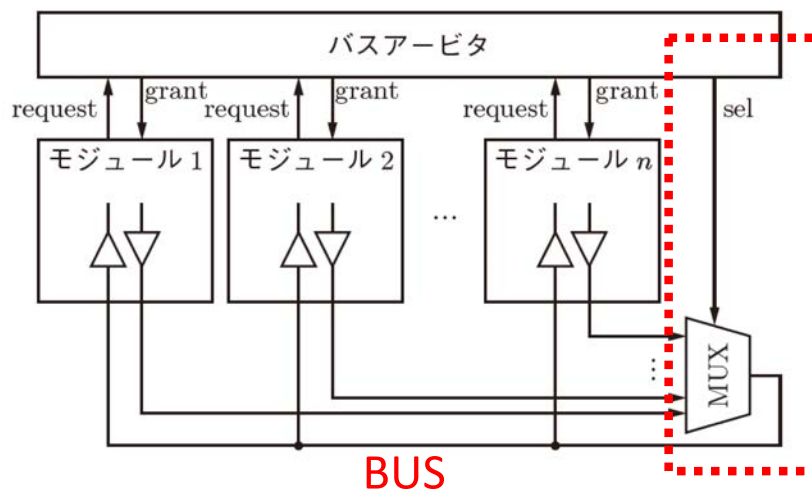


図 10・12 マルチプレクサを用いたバスの実装

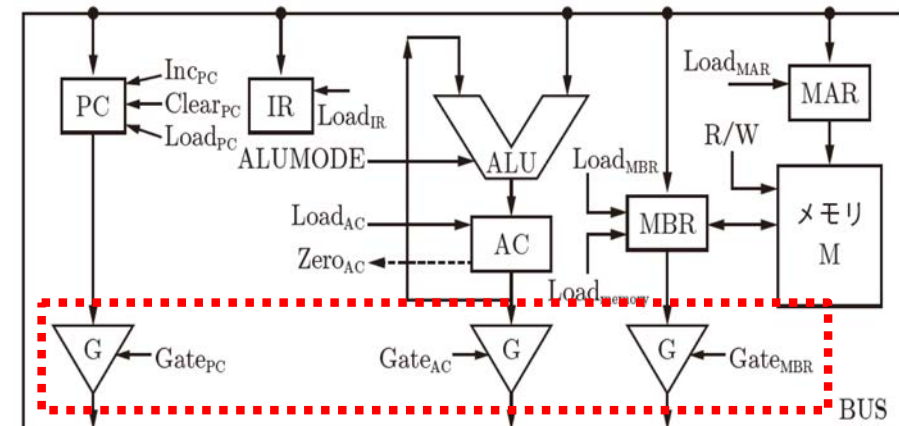


図 A・5 BUS を用いた CPU の回路図



- 1** T フリップフロップの論理記号を図 10・16 に示す. このフリップフロップは, 表 10・6 に示すように, クロックの立ち上がりで状態が反転する.

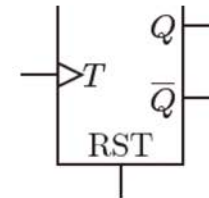


図 10・16 TFF の論理記号

表 10・6 T フリップフロップの動作表

入力		出力	
RST	CLK	Q^+	\overline{Q}^+
1	—	0	1
0		Q	\overline{Q}
0		\overline{Q}	Q

D フリップフロップを用いて T フリップフロップを実現せよ.



- 2 JK フリップフロップの論理記号を図 10・17 に示す。このフリップフロップは、リセット、クロック以外に二つの入力信号 J および K をもち、表 10・7 に示すような動作を行う。

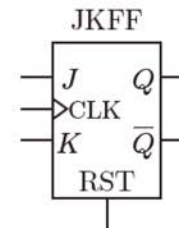


図 10・17 JKFF の論理記号

D フリップフロップを用いて JK フリップフロップを実現せよ。

- 3 JK フリップフロップを用いて T フリップフロップを実現せよ。

表 10・7 JK フリップフロップの動作表

入力				出力	
RST	CLK	J	K	Q^+	\overline{Q}^+
1	—	—	—	0	1
0		—	—	Q	\overline{Q}
0		0	0	Q	\overline{Q}
0		0	1	0	1
0		1	0	1	0
0		1	\overline{Q}	Q	



- 4 イネーブル (enable) 機能付き D フリップフロップの論理記号を図 10・18 に示す。このフリップフロップは、表 10・8 に示すように、入力信号 EN の値が 1 のときにのみフリップフロップの内部状態が更新される。

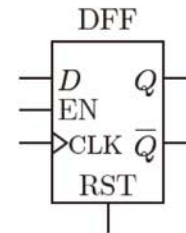


図 10・18 イネーブル機能付き DFF の論理記号

表 10・8 イネーブル機能付き D フリップフロップの動作表

入力				出力	
RST	CLK	EN	D	Q^+	\overline{Q}^+
1	—	—	—	0	1
0		—	—	Q	\overline{Q}
0		0	0	Q	\overline{Q}
0		0	1	Q	\overline{Q}
0		1	0	0	1
0		1	1	1	0

イネーブル機能をもたない D フリップフロップに論理ゲートを追加してイネーブル機能付き D フリップフロップを実現せよ。



演習問題 1

- Tフリップフロップの論理記号を 図10・16 に示す. このフリップフロップは, 表10・6 に示すように, クロックの立ち上がりで状態が反転する. Dフリップフロップを用いて Tフリップフロップを実現せよ.

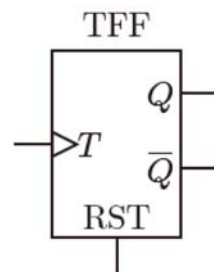


図 10・16 TFF の論理記号

表 10・6 T フリップフロップの動作表

入力		出力	
RST	CLK	Q^+	\overline{Q}^+
1	—	0	1
0		Q	\overline{Q}
0		\overline{Q}	Q





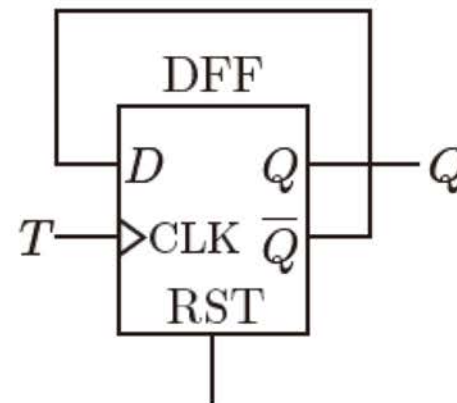
考慮時間

- 5-10分間程度で問題を解いてみてください。その間、ビデオを止めてください。
- この頁は30秒程度で次の頁に移行します。



演習問題 1 解答

- Dフリップフロップを用いた T フリップフロップの実現方法を解図 10・1 に示す.



解図 10・1

D フリップフロップを用いた T フリップフロップの実現方法





演習問題 2

- JKフリップフロップの論理記号を図10・17 に示す．このフリップフロップは，リセット，クロック以外に2つの入力信号 J および K をもち，表10・7 に示すような動作を行う．Dフリップフロップを用いて JK フリップフロップを実現せよ．

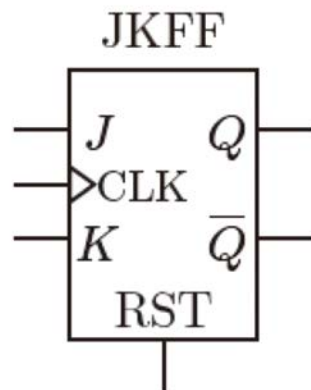


図 10・17

JKFF の論理記号

表 10・7

JK フリップフロップの動作表

入力				出力	
RST	CLK	J	K	Q^+	\overline{Q}^+
1	—	—	—	0	1
0		—	—	Q	\overline{Q}
0		0	0	Q	\overline{Q}
0		0	1	0	1
0		1	0	1	0
0		1	\overline{Q}	Q	





考慮時間

- 5-10分間程度で問題を解いてみてください。その間、ビデオを止めてください。
- この頁は30秒程度で次の頁に移行します。

演習問題 2 解答

- Dフリップフロップを用いた JK フリップフロップの実現方法を 解図10・2 に示す.

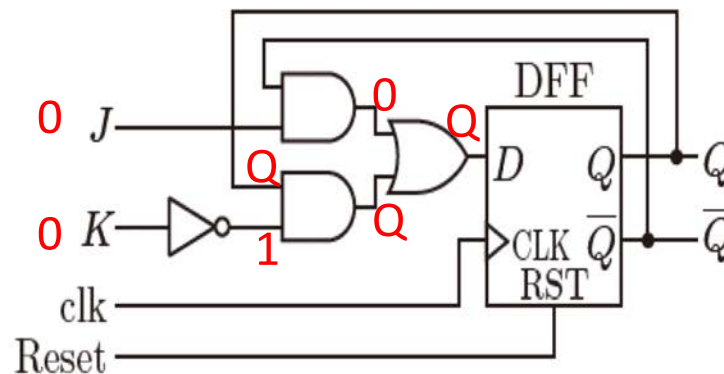







表 10・7 JK フリップフロップの動作表

入力				出力	
RST	CLK	J	K	Q^+	\overline{Q}^+
1	—	—	—	0	1
0		—	—	Q	\overline{Q}
0		0	0	Q	\overline{Q}
0		0	1	0	1
0		1	0	1	0
0		1	\overline{Q}	Q	

解図 10・2

D フリップフロップを用いた JK フリップフロップの実現方法



演習問題 2 解答

- Dフリップフロップを用いた JK フリップフロップの実現方法を 解図10・2 に示す.

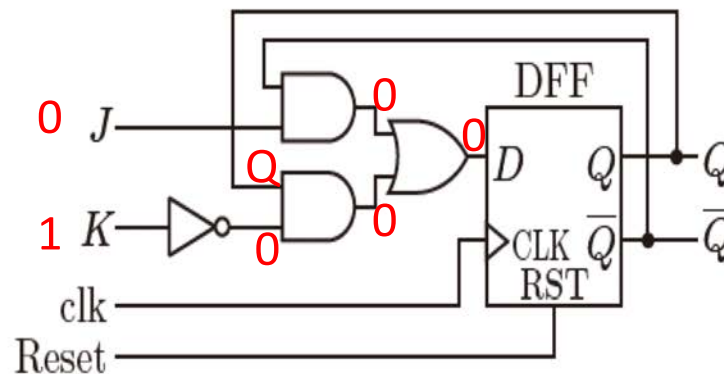







表 10・7 JK フリップフロップの動作表

入力				出力	
RST	CLK	J	K	Q^+	\overline{Q}^+
1	—	—	—	0	1
0		—	—	Q	\overline{Q}
0		0	0	Q	\overline{Q}
0		0	1	0	1
0		1	0	1	0
0		1	\overline{Q}	Q	

解図 10・2

D フリップフロップを用いた JK フリップフロップの実現方法



演習問題 2 解答

- Dフリップフロップを用いた JK フリップフロップの実現方法を 解図10・2 に示す.

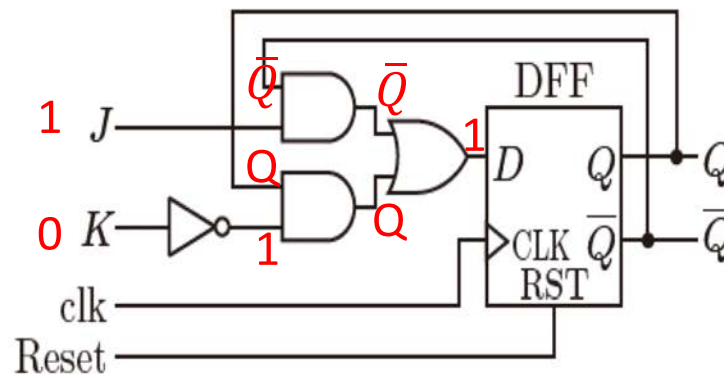







表 10・7 JK フリップフロップの動作表

入力				出力	
RST	CLK	J	K	Q^+	\overline{Q}^+
1	—	—	—	0	1
0		—	—	Q	\overline{Q}
0		0	0	Q	\overline{Q}
0		0	1	0	1
0		1	0	1	0
0		1	\overline{Q}	Q	

解図 10・2

D フリップフロップを用いた JK フリップフロップの実現方法



演習問題 2 解答

- Dフリップフロップを用いた JK フリップフロップの実現方法を 解図10・2 に示す.

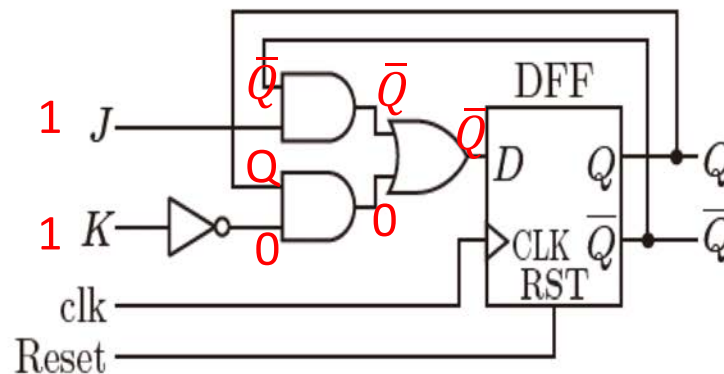







表 10・7 JK フリップフロップの動作表

入力				出力	
RST	CLK	J	K	Q^+	\overline{Q}^+
1	—	—	—	0	1
0		—	—	Q	\overline{Q}
0		0	0	Q	\overline{Q}
0		0	1	0	1
0		1	0	1	0
0		1	\overline{Q}	Q	

解図 10・2

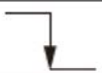

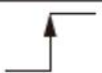
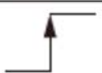

D フリップフロップを用いた JK フリップフロップの実現方法



演習問題 3

- JK フリップフロップを用いて T フリップフロップを実現せよ。

表 10・7 JK フリップフロップの動作表

入力				出力	
RST	CLK	J	K	Q^+	\overline{Q}^+
1	—	—	—	0	1
0		—	—	Q	\overline{Q}
0		0	0	Q	\overline{Q}
0		0	1	0	1
0		1	0	1	0
0		1	\overline{Q}	Q	

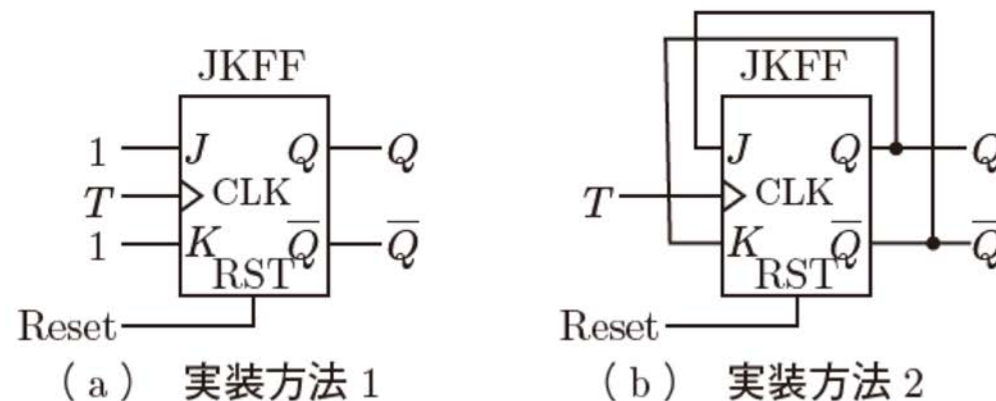


考慮時間

- 5-10分間程度で問題を解いてみてください。その間、ビデオを止めてください。
- この頁は30秒程度で次の頁に移行します。

演習問題 3 解答

- JKフリップフロップを用いた Tフリップフロップの実現方法を解図 10・3 に示す.



解図 10・3

JK フリップフロップを用いた T フリップフロップの実現方法



演習問題 4

- イネーブル (enable) 機能付き Dフリップフロップの論理記号を図 10・18 に示す. このフリップフロップは, 表10・8 に示すように, 入力信号 EN の値が 1 のときにのみフリップフロップの内部状態が更新される. イネーブル機能をもたない Dフリップフロップに論理ゲートを追加してイネーブル機能付き Dフリップフロップを実現せよ.

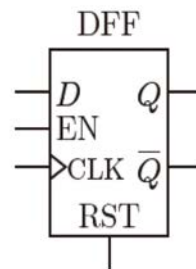


図 10・18 イネーブル機能付き DFF の論理記号

表 10・8 イネーブル機能付き D フリップフロップの動作表

入力				出力	
RST	CLK	EN	D	Q^+	\overline{Q}^+
1	—	—	—	0	1
0		—	—	Q	\overline{Q}
0		0	0	Q	\overline{Q}
0		0	1	Q	\overline{Q}
0		1	0	0	1
0		1	1	1	0





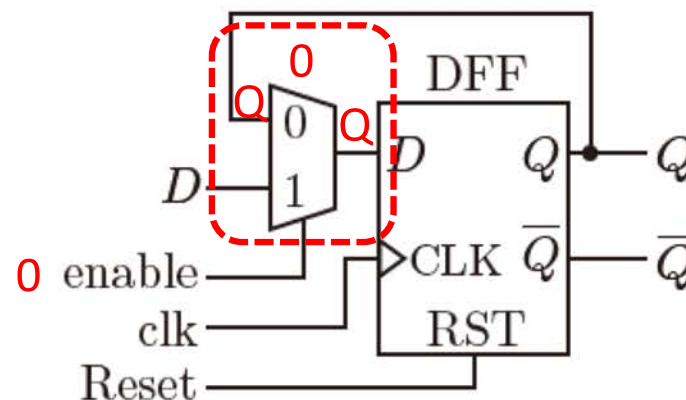
考慮時間

- 5-10分間程度で問題を解いてみてください。その間、ビデオを止めてください。
- この頁は30秒程度で次の頁に移行します。



演習問題 4 解答

- イネーブル機能付き Dフリップフロップの実現方法を解図10・4 に示す.



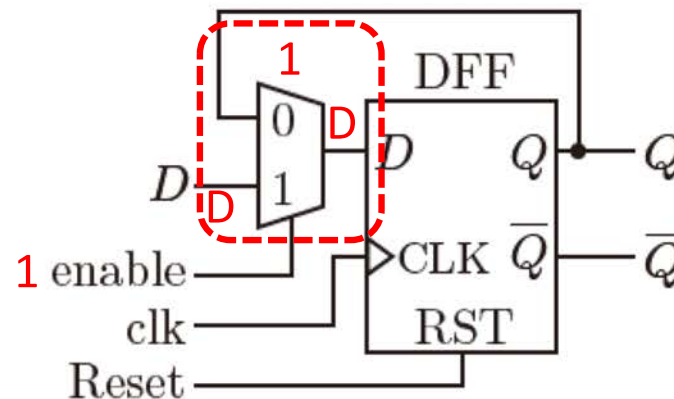
解図 10・4

イネーブル機能付き D フリップフロップの実現方法



演習問題 4 解答

- イネーブル機能付き Dフリップフロップの実現方法を解図10・4 に示す.



解図 10・4

イネーブル機能付き D フリップフロップの実現方法





2 回目の授業終了

基礎工学部情報科学科 東野輝夫





授業終了

皆さん

今日はレポート課題はありません

