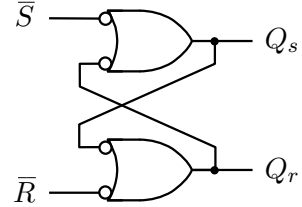


今回の内容

9.1 D-ラッチと D-フリップフロップ 9-1

9.1 D-ラッチと D-フリップフロップ

前回紹介した SR-ラッチは、 $(Q_s, Q_r) = (1, 0)$ と $(0, 1)$ の 2 つの状態を持っていて、入力 S と R のどちらか一方を 1 にすることで、この 2 つの状態を切り替えることができました。この SR-ラッチを応用することで、真理値を記憶する論理回路を実現することができます。

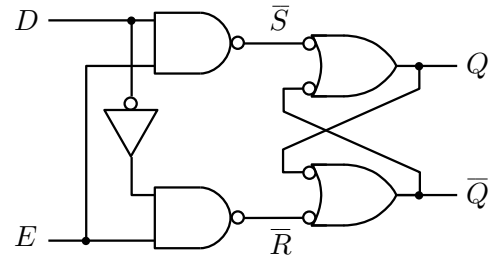


\bar{S}	\bar{R}	Q_s	Q_r
0	0	1	1
0	1	1	0
1	0	0	1
1	1	変化せず	変化せず

$\bar{S}\bar{R}$ -ラッチとその真理値表

ここでは、前回 (第 8 回) の配布資料 4 ページで紹介した右図の $\bar{S}\bar{R}$ -ラッチを元にして考えることにします。 $\bar{S}\bar{R}$ -ラッチは、通常は、 $\bar{S} = \bar{R} = 1$ を入力した状態にしておき、 Q_s および Q_r の値を保持させます。 Q_s や Q_r の値を変えたい場合は、 \bar{S} または \bar{R} のどちらか一方だけを 0 にします。 \bar{S} を 0 にしたのであれば $(Q_s, Q_r) = (1, 0)$ 、 \bar{R} を 0 にしたのであれば $(Q_s, Q_r) = (0, 1)$ となり、その後、 $\bar{S} = \bar{R} = 1$ に入力を戻しても、その状態が維持されます。

D-ラッチ この $\bar{S}\bar{R}$ -ラッチに 2 つの NAND ゲートと 1 つの NOT ゲートを追加して、右のような論理回路を考えると、



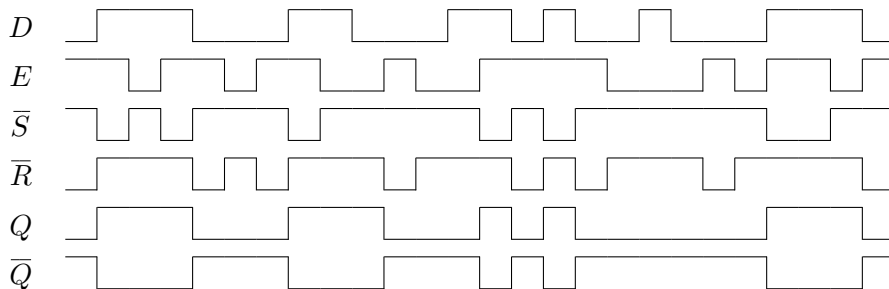
- $E = 1$ のときは、 $(Q, \bar{Q}) = (D, \bar{D})$ となり、
- $E = 0$ のときは、 (Q, \bar{Q}) の値は (E が 1 から 0 へ変化する直前の値から) 変化しない

E	D	Q	\bar{Q}
0	0	変化せず	変化せず
0	1	変化せず	変化せず
1	0	0	1
1	1	1	0

D-ラッチとその真理値表

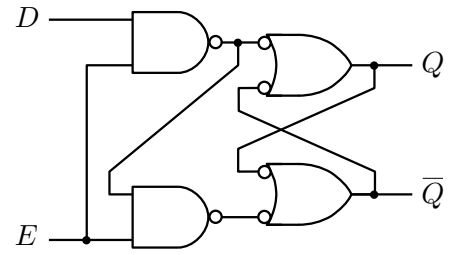
ようにできます。この論理回路を **D-ラッチ** と呼びます。

次は D-ラッチのタイミングチャートの例です。D-ラッチは、入力 E の値が 1 から 0 に変化した時点の、入力 D の値を、出力 Q の値として保持し続けることができます。



D-ラッチのタイミングチャート例

D-ラッチの別の実現方法 前ページの D-ラッチの論理回路に使用されている NOT ゲートの出力値は、入力 E との論理積を計算するだけに使われますので、実質的には、 $E = 1$ のときのみ有効となります。このため、この NOT ゲートの働きを、 \bar{S} に接続されている方の NAND ゲートにさせることで、右のように NOT ゲートを1つ節約することもできます。



D-ラッチの別の実現方法



D-ラッチによる真理値の記憶 D-ラッチは 1 bit の記憶装置として使用することができます。通常は $E = 0$ としておき、 Q の値を保持し続けます。新しい真理値を記憶したい場合は、記憶したい値を D に入力して、一瞬だけ $E = 1$ として、ただちに $E = 0$ に戻します。ただし、このとき $E = 1$ とする時間が短すぎると、 $\bar{S}\bar{R}$ -ラッチの部分の出力が安定する前に、 (\bar{S}, \bar{R}) が $(1, 1)$ に戻ってしまい、正しい値を記憶することができません。 $E = 1$ とする時間を長くすればこの問題を回避できますが、その間は入力 D の値を保ち続ける必要があります。

つまり、D-ラッチを 1 bit の記憶装置として使用するためには、

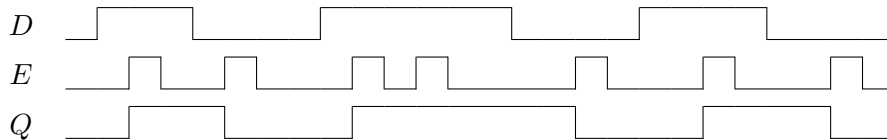
通常は $E = 0$ に保っておく

新しい D の値を記憶するためには、ある程度の時間は E を 1 に保つ

$E = 1$ となっている間は D の値は変化させない

E が 1 から 0 へ変化した時点の D の値が Q として記憶される

という使い方を守る必要があります。このような使い方をした場合の、D-ラッチのタイミングチャートは、例えば次のようなものになります。

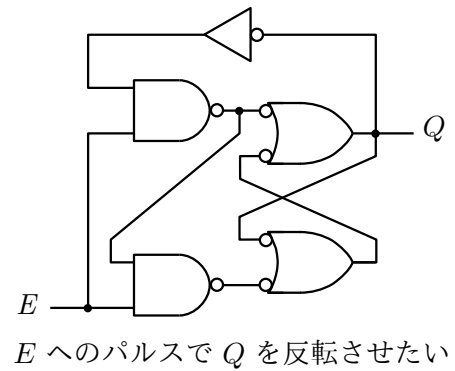


D-ラッチの問題点 上述のように、D-ラッチは1 bit の記憶装置として利用することができますが、その使用法に求められる

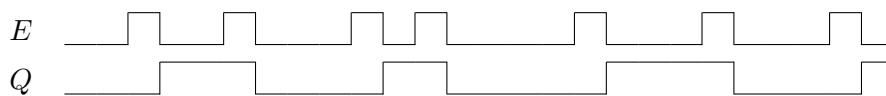
新しい値の記憶時には、一定時間 E を 1 に保つ

$E = 1$ となっている間は D の値は変化させない

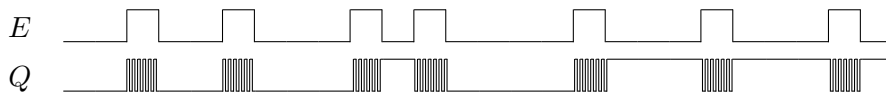
という2つの条件は、D-ラッチを使い難いものにしてしまいます。例えば、D-ラッチの出力 Q を反転して、同じD-ラッチの入力 D に接続した¹右上のような論理回路を



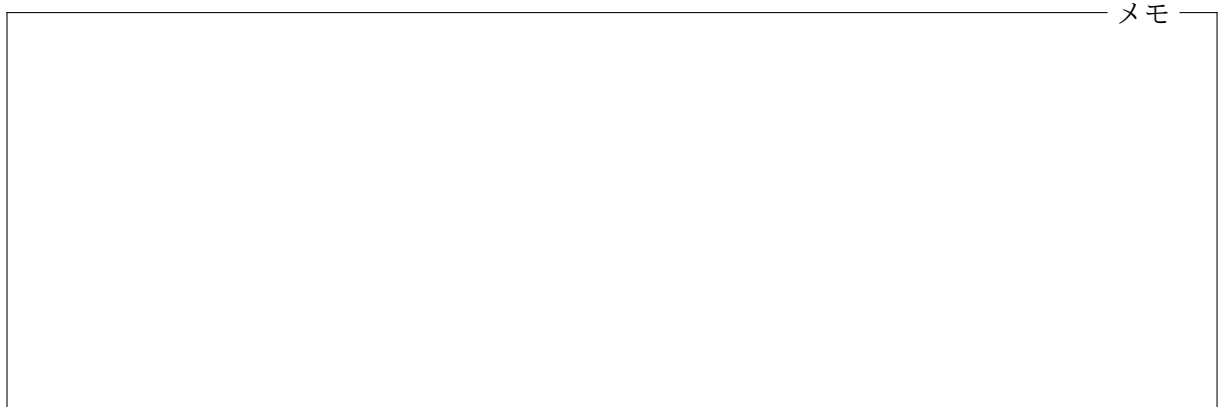
考えると、次のタイミングチャートのように、入力 E に与えるパルス (0→1→0 の変化) ごとに出力 Q の値を反転させることができそうなのですが、なかなかうまく行きません。



D-ラッチの入力 D の値を記憶するためには、 E を値を一定時間 1 にしておく必要がありますが、 $E = 1$ となっている間は、 D の値がそのまま出力 Q に伝わってしまいますので、それが NOT ゲートで反転されて、 D の値を変えてしまいます。つまり、 $E = 1$ である間、この論理回路の出力 Q は、次のタイミングチャートのように、短い周期で (0 と 1 の間を) 振動し続けることとなります²。 $E = 0$ に戻ったときの Q の値は予想できません。



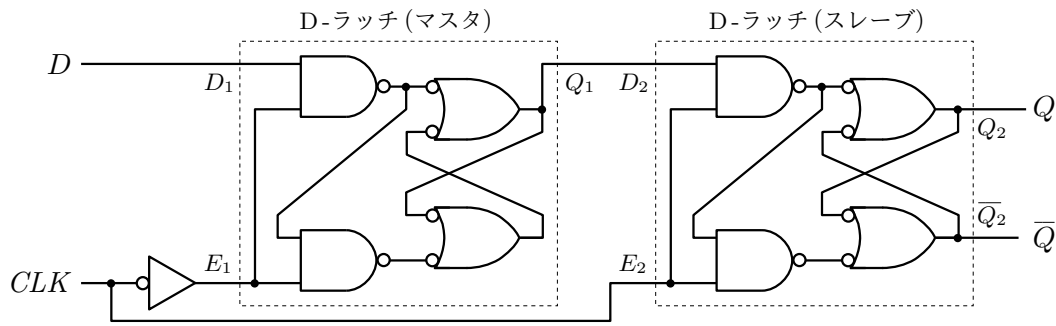
かと言って、 $E = 1$ にしておく時間を短くすると、 $\overline{S}\overline{R}$ -ラッチの部分の出力が安定する前に (\overline{S} , \overline{R}) が (1, 1) に戻ってしまい、 D の値を正しく記憶することができなくなってしまいます。



D-フリップフロップ このような D-ラッチの問題点は、2つの D-ラッチを次のように組み合わせることで解決できます。

¹ \overline{Q} を直接 D に接続しても構いません。

²実際に 74HC00 と 74HC04 を使ってこの論理回路を作成したところ、約 20MHz で発振しました。



マスタ・スレーブ構成の D-フリップフロップ

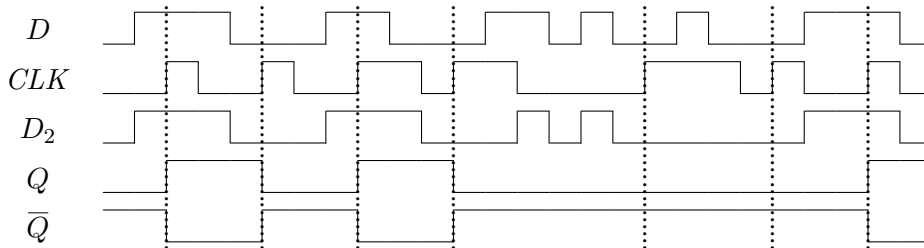
このような構成の論理回路を「マスタ・スレーブ構成の D-フリップフロップ」と呼びます。この論理回路は「マスタ (master)」と呼ばれる 1 段目の D-ラッチと「スレーブ (slave)」と呼ばれる 2 段目は D-ラッチから構成されており、1 段目の D-ラッチの出力 Q_1 を、2 段目の D-ラッチの入力 D_2 に接続し、「クロック」と呼ばれる入力 CLK を、2 段目の E_2 にはそのまま、1 段目の E_1 には反転して与えています。この論理回路 (D-フリップフロップ) 全体の入力 D が 1 段目の入力 D_1 となり、2 段目の出力 Q_2 と \overline{Q}_2 が、そのまま全体の出力 Q と \overline{Q} となります。

D-フリップフロップの動作 D-フリップフロップは、それを使う立場から見ると、次のように動作します。

入力 CLK の立ち上がり (0 から 1 への変化が起こる) 時の入力 D の値が出力 Q となる。

Q の値は、 CLK の次の立ち上がりまで変化しない。

出力 \overline{Q} の値は常に Q の否定となります。D-ラッチと違って $CLK = 1$ の時間を短くする必要はありません。D-フリップフロップのタイミングチャートは次のようなものになります。



D-フリップフロップのタイミングチャート例

マスタ・スレーブ構成の D-フリップフロップの動作原理

1 段目の D-ラッチの E_1 は \overline{CLK} 、2 段目の E_2 は CLK となっていますので、1 段目の出力 Q_1 と 2 段目の出力 Q_2 の値は、右の表のようになります。このため、 $CLK = 0$ の間、2 段目の出力 Q_2 は変化しませんが、2 段目の入力 D_2

CLK	Q_1	Q_2
0	D	変化しない
1	変化しない	Q_1

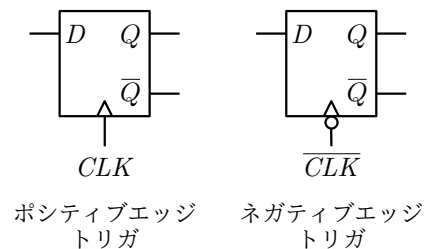
は D と等しくなっていますので、 $CLK = E_2$ の値が 0 から 1 へ変化すると、 Q_2 はその値 (CLK の立ち上がり時の D の値) に変わります。 $CLK = E_2 = 1$ の間は、1 段目の出力 Q_1 がそのまま、2 段目の出力 Q_2 になりますが、1 段目の E_1 は、この間は 0 なので、 D が変わっても Q_1 の値は (CLK の立ち上がり時の D の値から) 変化しません。つまり、 $CLK = 1$ の間も Q_2 の値は保持されます。以上が、マスタ・スレーブ構成の D-フリップフロップの動作原理です。

レベルトリガ方式とエッジトリガ方式 D-ラッチでは、入力 E の値が(一定の時間) 1 となること
 がきっかけとなって、その内部状態が変化しましたが、(マスタ・スレーブ構成の) D-フリップフ
 ロップでは、入力 CLK の値の立ち上がり (0 から 1 への変化) がきっかけとなって、その内部状態
 が変化します。一般に、順序論理回路 (内部状態を持つ論理回路) の内部状態を変化させるきっか
 けが、一定の時間の 1 (あるいは 0) の値である場合をレベルトリガ (**level trigger**) 方式と呼び、
 0 から 1 (あるいは 1 から 0) への変化である場合をエッジトリガ (**edge trigger**) 方式と呼び
 ます。

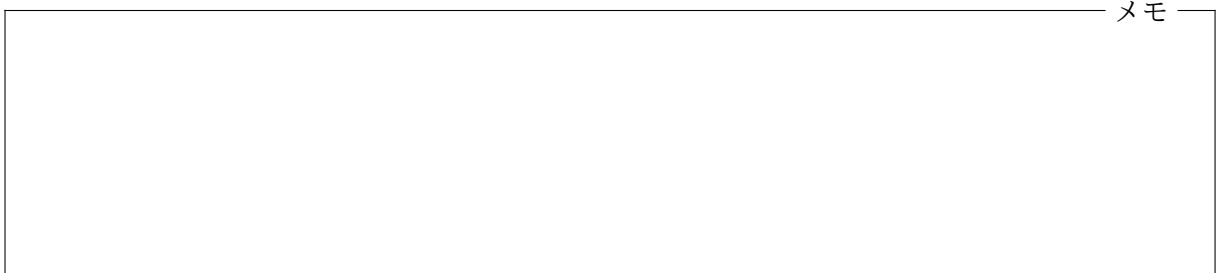
レベルトリガ方式では、一定の時間、各入力の値を固定しなければならず、論理回路の現在の状
 態に依存して、次の状態を決定しなければならないような場合には、3 ページで紹介したような問
 題が発生しがちです。このため、順序論理回路を設計する場合は、エッジトリガ方式の採用が基本
 となります。



4 ページの D-フリップフロップのように、0 から 1 へ
 の変化 (立ち上がり) が状態変化のきっかけとなるものを、
 「ポジティブエッジトリガ (positive edge trigger)」と呼び、
 1 から 0 への変化 (立ち下がり) がきっかけとなるものを
 「ネガティブエッジトリガ (negative edge trigger)」と呼び
 ます。ポジティブエッジトリガであれ、ネガティブエッジ
 トリガであれ、エッジトリガ方式で、1 bit の値を記憶でき
 る論理回路を、一般に **D-フリップフロップ (D flip-flop)**
 と呼び³、論理回路図上で右上のような記号で表します。 CLK や \overline{CLK} の入力に描かれた三角形
 は、その入力のエッジトリガであることを表しています。



D-フリップフロップの記号



³逆に、D-ラッチのことを「レベルトリガ方式の D-フリップフロップ」と呼ぶことがありますので注意して下さい。この授業では、エッジトリガ方式のみの「フリップフロップ」を扱います。

D-フリップフロップの別の実現方法 D-フリップフロップは、マスタ・スレーブ構成ではなく、次のような論理回路で実現することもできます。この回路もポジティブエッジトリガのD-フリップフロップであり、マスタ・スレーブ構成と全く同様に動作することができます。

