

SR ラッチによる非同期回路の設計†

井上 訓行^{††} 奥川 峻史^{††}

SR ラッチを基本構成要素とする非同期式順序回路の設計法^{1)~3)} について述べる。この方法は状態変数1つに SR ラッチ1個を割り当て、順序回路の励起表に SR ラッチの励起表を適用して SR の制御表⁴⁾ を作り、各ラッチの S, R 入力を求めて順序回路を構成する方法である。本稿では SR ラッチを解析することによって、 $SR=0$ のとき S, R 入力の同時変化に対してもラッチは安定に動作し、 S, R に NAND ラッチでは静的0ハザード、NOR ラッチでは静的1ハザードがあっても出力に影響しないことを明確にする。SR の制御表は順序回路の励起表に SR ラッチの励起表を適用して得られる。この表から S, R を求めると $SR=0$ は満たされているので、 S, R への入力を2段回路(1段は SR ラッチのゲートを利用するので実質1段回路)で構成すればハザードを考慮する必要がないことを示す。すなわち状態割当ての後は、SR ラッチによる方法で非同期回路を設計すると、SR フリップフロップを使って同期回路を設計するのが同じであることが示される。最後に普通の方法と SR ラッチによる方法を比較するため特殊な JK フリップフロップ^{5),6)} を2つ設計する。

1. ま え が き

順序回路は遷移表が与えられたとき、各状態に状態変数を割り当て、状態変数の論理式を導くことによって設計される。非同期式順序回路ではレースがないように状態を割り当て、ハザードがないよう(ハザードフリー)に論理式を求めなければならない。本稿ではハザードの問題を考察する。

SR ラッチは単独で用いられることは少ないが、順序回路の基本構成要素として広く利用されている。SR ラッチによる方法は順序回路の状態変数1つに SR ラッチ1個を割り当て、各ラッチの S, R 入力を求めて非同期回路を設計する方法である^{1)~3)}。この方法では、ハザードフリーに論理式を導く必要はない。ラッチの入力にハザードがある場合の動作について文献1)にふれられているが、明確にされているとはいえない。

本稿ではまず、SR ラッチを解析し $SR=0$ のもとでは S, R 入力が同時に変化しても安定に動作し、 S, R 入力にハザード(NAND ラッチでは静的0ハザード、NOR ラッチでは静的1ハザード)があっても出力に影響しないことを明確にする。この結果 S, R 入力を図6¹⁾、図7に示す2段回路(1段はラッチのゲートを利用するので実質1段回路)で構成すれば、NAND ラッチでは静的1ハザードが、NOR ラッチでは静的0ハザードがなく、いずれにも動的ハザード

はない。したがってハザードの問題を生じないことがわかる。すなわち、SR ラッチによる方法では、SR フリップフロップを使って同期回路を設計すると、状態割当てを除いて、同じ方法で非同期回路が設計できることを示す。

最後に特殊な JK フリップフロップ^{5),6)} をこの方法で設計し、普通の方法と比較・検討する。

2. SR ラッチ

非同期回路の基本要素となる SR ラッチについて述べる。図1(a), (b)に示すように SR ラッチは NAND または NOR 回路2個で構成され、それぞれ NAND ラッチ、NOR ラッチと呼ばれている。このラッチの特性表と励起表を図1(c), (d)に示す。図1(d)は同期回路で使用される SR フリップフロップの励起表とまったく同じである。

2.1 SR ラッチの論理的解析

SR ラッチを設計に利用するため、まず論理的に解析する。図1(a), (b)は図2(a), (b)のように書き換えることができ、そのときの遷移表は図2(c), (d)となる。 Y は NAND ラッチでは、

$$Y = S + \bar{R}y \quad (1)$$

NOR ラッチでは

$$Y = \bar{R}S + \bar{R}y \quad (2)$$

となる。

$SR=1$ のとき次の問題を生ずる。

(1) SR が $11 \rightarrow 00$ と変化したとき、次の状態が確定しない(図2(c), (d))。基本モードで考えると、回路の入力は1つしか変化しないが、内部で S, R が

† Synthesis of Asynchronous Sequential Circuits Using SR Latches by NORIYUKI INOUE and SHUNJI OKUGAWA (Kyoto Sangyo University).

†† 京都産業大学

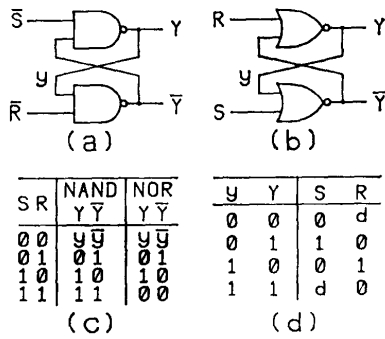


図1 SR ラッチ (a) NAND ラッチ, (b) NOR ラッチ, (c) 特性表, (d) 励起表
Fig. 1 SR latch (cross coupled circuit). (a) NAND latch, (b) NOR latch, (c) Characteristic table, (d) Excitation table

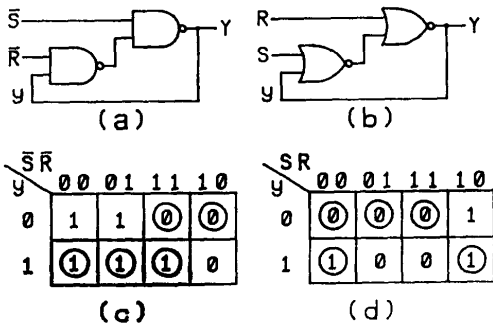


図2 SR ラッチ (a) NAND ラッチ, (b) NOR ラッチ, (c) NAND ラッチの遷移表, (d) NOR ラッチの遷移表
Fig. 2 SR latch showing feedback. (a) NAND latch, (b) NOR latch, (c) Transition table of NAND latch, (d) Transition table of NOR latch

同時に変化する可能性はある。(SR が 00→11, 01→10, 10→01 と変化しても問題はない。)

(2) 図1(a), (b)において SR=1 のとき, 安定状態ではあるが, 出力は NAND ラッチでは両方 1, NOR ラッチでは両方 0 となり, 出力の相補関係は保たれない²⁾ (図1(c)).

(3) (1), (2)式からわかるように NAND ラッチと NOR ラッチの論理式が異なる。

以上の理由からふつう SR=0 を満たすように使用される。このとき(1), (2)式は

$$Y = S + \bar{R}y, \quad SR = 0 \quad (3)$$

となる。

しかし, SR ラッチそのものは SR=1 でも動作するので, 上記(1), (2), (3)を考慮した上であれば, S, R とともに 1 になるときに使用できる。フリップフ

ロップではこのような使い方がされているものも多い⁴⁾。

2.2 遅れを考慮した場合の解析

図1(a)のラッチにゲートの遅れを考慮し, ゲートと遅延に分離すると図3(a)のように書くことができ, 遷移表は図3(b)になる⁵⁾。ここでは簡単のため遅れは純遅延のみを考える。

2.2.1 入力が同時に変化する場合

① x_1x_2 が 01→10 (10→01) のとき $\Delta_1 + \Delta_2$ 後に出力は安定状態に落ち着く。

② x_1x_2 が 11→00 のとき $\max(\Delta_1, \Delta_2)$ 後に出力 11 に落ち着く。

③ x_1x_2 が 00→11 のとき, 図4に示すように周期 $\Delta_1 + \Delta_2$ で発振状態になり, 落ち着き先は決まらない。

このような状態はメタステーブルと言われ, デジタルシステムでは避けられない。アービタなどではメタステーブルが重要な問題となるので, 慣性遅延のある場合も含めて研究されている^{7), 8)}。

安定に動作するためには x_1 が変化したあと Δ_1 以上, x_2 が変化したあと Δ_2 以上遅れて他方が変化しなければならぬ。以上の解析からゲートの遅れを考慮

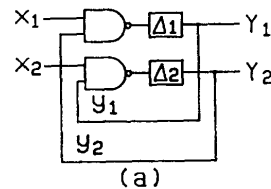


図3 遅れを考慮した NAND ラッチ (a) 回路図, (b) (a)の遷移表
Fig. 3 SR latch with delay. (a) Circuit, (b) Transition table

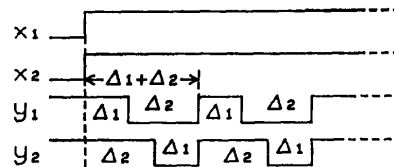


図4 x_1x_2 が 00→11 と変化したときのタイミング図
Fig. 4 Timing chart when x_1 and x_2 change 00 to 11.

しても $x_1x_2=00$ を除いて (図 1 (a) では $SR=0$ なら) 安定に動作することがわかる. NOR ラッチについても同様に解析できる.

2.2.2 入力に幅の狭いパルスが入る場合

非同期回路ではハザードにより幅の狭いパルス (グリッチ) が発生する可能性がある. パルス幅 Δ のグリッチが S, R 入力に入った場合を考える.

(1) $x_1x_2y_1y_2=1101$ のとき, x_1 に静的 1 ハザードによるグリッチが入ると, 図 5 (a) に示すように $\Delta > \Delta_1 + \Delta_2$ なら SR ラッチの内容が反転し, 安定状態になる. $\Delta < \Delta_1 + \Delta_2$ なら図 5 (b) に示すように Y_1, Y_2 は周期 $\Delta_1 + \Delta_2$ で発振し, メタステーブルになる.

x_2 に静的 1 ハザードによるグリッチが入っても $y_1=0$ であるから Y_2 には抜けない.

$x_1x_2y_1y_2=1110$ のときも同様である.

(2) $x_1x_2y_1y_2=1001$ のとき, x_1 に静的 1 ハザードによるグリッチが入ると Δ_1 後に Y_1 に出る. x_2 に静的 0 ハザードによるグリッチが入っても y_1 が 0 であるから出力には出ない.

$x_1x_2y_1y_2=0110$ の場合も同様である.

(3) $x_1x_2y_1y_2=0011$ において $x_1(x_2)$ に静的 0 ハザードによるグリッチが入ると, $\Delta_1(\Delta_2)$ 後に $Y_1(Y_2)$ に抜ける.

以上の解析結果をまとめると表 1 (a) になり, 次の定理を得る.

【定理 1】 NAND ラッチでは入力の静的 1 ハザードは出力に影響するが, 静的 0 ハザードは $x_1x_2=00$ の場合を除き出力に出ない.

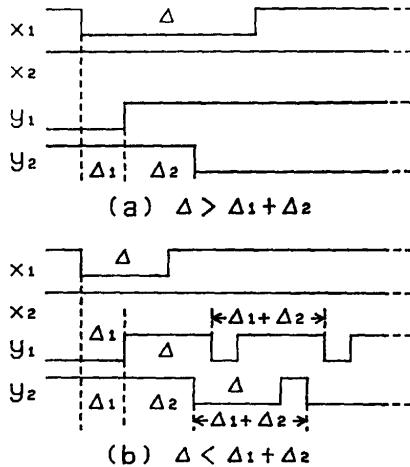


図 5 x_1 に静的 1 ハザードによるグリッチがあるときのタイミング図 ($x_2=1$, パルス幅 Δ)

Fig. 5 Timing chart with glitch by static 1 hazard in x_1 ($x_2=1$, pulse width Δ).

表 1 SR ラッチの出力とハザード
Table 1 Output and hazard of SR latch.
(a) NAND ラッチ
NAND latch.

x_1 x_2 (S R)	y_1 y_2 (Y Y)	静的 1 ハザード	静的 0 ハザード
		x_1 x_2 (S R)	x_1 x_2 (S R)
1 1	0 1	× ○	- -
1 1	1 0	○ ×	- -
1 0	0 1	× -	- ○
0 1	1 0	- ×	○ -
0 0	1 1	- -	× ×

(b) NOR ラッチ
NOR latch.

x_1 x_2 (S R)	y_1 y_2 (Y Y)	静的 0 ハザード	静的 1 ハザード
		x_1 x_2 (S R)	x_1 x_2 (S R)
0 0	0 1	× ○	- -
0 0	1 0	○ ×	- -
0 1	0 1	× -	- ○
1 0	1 0	- ×	○ -
1 1	0 0	- -	× ×

注) - : ハザードのない組合せ
× : ハザードが出力に影響する
○ : ハザードが出力に影響しない

(証明) 表 1 (a) より明らかである.

この定理は NAND ラッチを $SR=0$ の条件で使用するとき, \bar{S}, \bar{R} の静的 0 ハザードを考慮しなくてもよいことを示す.

NOR ラッチについて同様の解析を行うと表 1 (b) となり, NAND ラッチと同様に次の定理を得る.

【定理 2】 NOR ラッチでは入力の静的 0 ハザードは出力に影響するが, 静的 1 ハザードは $x_1x_2=11$ の場合を除き出力に出ない.

(証明) 表 1 (b) より明らかである.

定理 1 の静的 0 ハザードと定理 2 の静的 1 ハザードは, 関数ハザードによるグリッチであってもよいことに注意すべきである.

3. SR ラッチによる設計

順序回路の設計はふつう励起表から各状態変数と出力変数の論理式を求める. 非同期回路のとき論理式はハザードフリーに求めなければならない. 以下この方法を普通の設計法と呼ぶ. 次に SR ラッチを基本要素とする非同期回路の設計法を述べる. 順序回路の各状

態に SR ラッチ 1 つを割り当てる. この SR ラッチの入力は順序回路の励起表に図 1 (d) の SR ラッチの励起表を適用して得られる. すなわち, 現在の状態 y_i と次の状態 Y_i の関係を図 1 (d) を使って S_i と R_i で表すことができる. こうして得られた S, R の入力を表す表を SR の制御表⁹⁾と呼ぶ (表 2 参照). SR の制御表からラッチの S, R 入力は

$$S = f_1 + f_2 + \dots + f_n$$

$$R = g_1 + g_2 + \dots + g_m \quad (4)$$

の形に表すことができ, f_i ($1 \leq i \leq n$), g_j ($1 \leq j \leq m$) は入力変数と状態変数から成る積項である. こうして求めた S, R をラッチの入力とする方法を以下 SR ラッチによる設計と呼ぶ.

3.1 NAND ラッチによる設計

基本モードで考えると AND-OR (NAND 2 段) 回路には静的 1 ハザードはあっても, 静的 0 ハザードはない. したがって AND-OR-NOT 回路には静的 0 ハザードはあっても, 静的 1 ハザードはない. (4) 式をそのまま NAND 2 段回路で構成すると図 6 (a) となる. この S, R には静的 0 ハザードはなく, したがって \bar{S}, \bar{R} には静的 1 ハザードはない. ふう図 6 (a) の中間の NAND 2 段を束線化 (bun-

dling) によって取り除き図 6 (b)¹⁰⁾ のように設計される.

(補題 1) 図 6 の SR ラッチの入力には静的 1 ハザードはない.

(証明略)

(補題 2) SR の制御表から S, R を求めるとき, $SR=0$ は満たされているか, 考慮する必要はない.

(証明) SR ラッチの励起表 (図 1 (d)) は y から Y へのどの変化に対しても, 必ず一方は 0 である. もとの順序回路に don't care の状態がなければ, SR の制御表で S, R のいずれか一方は必ず 0 であり, S と R が最小項を共有することはない. もとの順序回路に don't care があると, それに対応する SR の制御表の最小項は共に don't care になる. この最小項が S と R に共有され, $SR=0$ が満たされない可能性がある. しかし, don't care はもともと考慮する必要のない状態であるから, その状態になったときのハザードも考慮する必要はない. (証明終)

[定理 3] SR ラッチによる設計法で NAND ラッチを使って設計するとき, S, R をハザードフリーに設計する必要はない.

(証明) 補題 1 により \bar{S}, \bar{R} には静的 1 ハザードはなく, 補題 2 により $SR=0$ を考慮する必要はない. したがって, 定理 1 により \bar{S}, \bar{R} に静的 0 ハザードがあっても出力に影響しない. (証明終)

3.2 NOR ラッチによる設計

NAND ラッチと同様に (4) 式を AND-OR 2 段回路で構成し, NOR ラッチの S, R 入力とすると図 7 (a) になる. この S, R には静的 0 ハザードはない. また SR の制御表を用いて設計するとき, NAND ラッチの場合と同様に $SR=0$ は満たされるか, 考慮する必要はない.

[定理 4] SR ラッチによる設計法で NOR ラッチを使って順序回路を設計するとき, S, R をハザードフリーに設計する必要はない.

(証明) S, R には静的 0 ハザードはなく, 前節の補題 2 より $SR=0$ は考慮する必要はない. よって定理 2 より S, R に静的 1 ハザードがあっても NOR ラッチの出力に影響しない. (証明終)

図 7 (a) の回路はふう途中の OR 回路を除き図 7 (b) または NOR 回路のみを使用して図 7 (c) のように構成される.

3.3 SR ラッチによる設計の特徴

まず, NAND ラッチで構成された回路と NOR

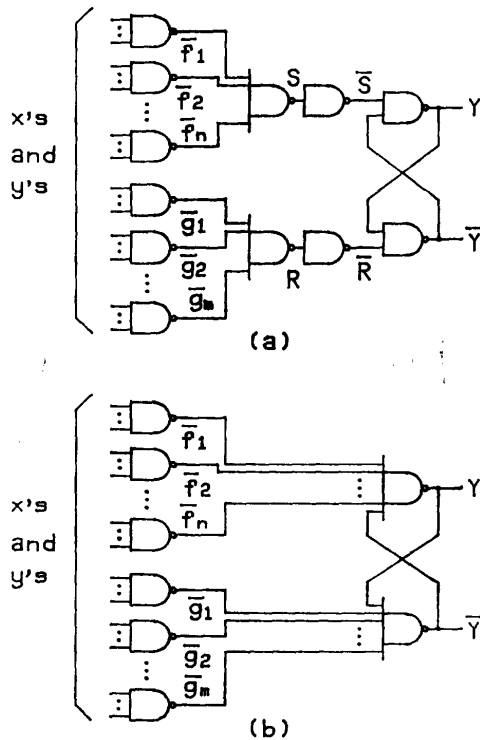


図 6 NAND ラッチによる構成
Fig. 6 Implementation by NAND latch.

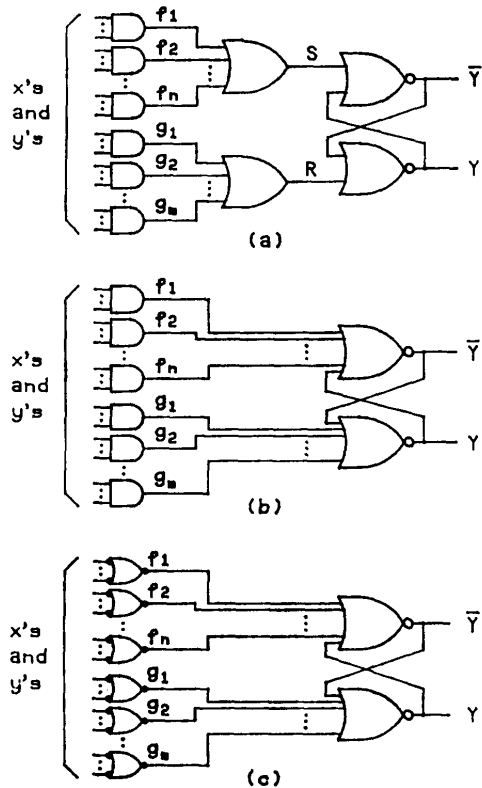


図 7 NOR ラッチによる構成
Fig. 7 Implementation by NOR latch.

ラッチで構成された回路の間に次の定理が成り立つ。

【定理 5】 NAND ラッチで構成された回路の入力の NAND を AND に置き換え、出力の極性を逆にすれば、NOR ラッチで構成できる。逆も同様である。

(証明) (4)式が NAND ラッチでは図 6 (b)に、NOR ラッチでは図 7 (b)になることから、入力を置き換えることは明らかである。また、SR の制御表を使って設計されていれば、 $SR=0$ を考慮せず、常に出力の相補関係が保たれているので、出力の逆の極性は必ずとれる。 (証明終)

次に状態割当てについて考察する。一般に u 状態に r ビット (r は $\log_2 u$ 以上の最小の整数) を割り当てる独立な割当て方は $2^r P_u / (r! 2^r)$ 通りある。この中には変数の 0 と 1 を入れ替えた割当ては独立とは考えていない。たとえば、4 状態に 2 変数を割り当てるなら $4! / (2! 2!) = 3$ 通りの独立な割当て方がある。同期回路では割当て方は任意であるが、非同期回路ではレースがないようにしなければならない。どの割当てから最も簡単な回路が導かれるかは一般論はない。

状態変数 y_i の 0 と 1 を逆にすると、普通的设计法では Y_i の補元関数を求め y_i を \bar{y}_i としなければ

ならない。こうすることによって回路の簡単さは異なる。しかし、SR ラッチによる設計法では単に y_i を \bar{y}_i として、 S_i と R_i を入れ替えるだけであるから簡単さは変わらない。

普通的设计法で Y をハザードフリーに求めた式を (1) 式の形に変形して設計することもできる。(1) 式の形に変形し、NAND ラッチの S, R 入力を求めると、NAND ラッチの出力として Y を得ることができる (4.1 節参照)。このとき $SR=0$ が成り立っていれば、①NAND ラッチのもう一方の出力は \bar{Y} となり、②(3)式より NOR ラッチでも構成でき出力は Y と \bar{Y} になる。しかし、一般に $SR=0$ は成り立つとは限らない。また、普通的设计法による構成と SR ラッチによる設計とではどちらが簡単になるかは一般にはわからない。

最後に回路の時間遅れを考える。入力変数の補元は得られるものとし、ゲートの遅れはすべて同じであると仮定する。普通的设计では Y を 2 段回路で構成でき、 Y の補元をとるためインバータが必要である (特殊な場合として Y の補元を要しないこともある)。したがって、ふつう遅れは 3 ゲート分である。SR ラッチによる設計法で構成したときは、ラッチの反転に 2 ゲート分必要であり、 S, R 入力は 1 段で構成できる。したがってどちらの設計法によっても最大時間遅れが 3 ゲート分になるように設計できる。

3.4 設計例

SR ラッチによる設計によって簡単な回路が得られる例を示す。表 2 (a) の遷移表で示される順序回路を設計する。前節で述べたように状態変数 y_1, y_2 の割当て方は 3 通りあるが、レースがないようにすると 1 通りになる (y_1, y_2 の 0, 1 を入れ替えた割当てはある)。この割当てを表 2 (a) の左端欄に示す。表 2 (b) から

$$\begin{aligned}
 Y_1 &= \bar{x}_2 y_1 \bar{y}_2 + x_1 \bar{x}_2 y_1 + x_1 x_2 y_2 + \bar{x}_1 x_2 y_1 \\
 &\quad + \bar{x}_1 y_1 \bar{y}_2 + x_1 y_1 y_2 + x_2 y_1 y_2 \\
 Y_2 &= \bar{x}_1 \bar{x}_2 y_2 + x_1 \bar{x}_2 \bar{y}_1 + x_1 x_2 y_2 + x_2 y_1 \bar{y}_2 \\
 &\quad + \bar{x}_1 y_1 y_2 + x_1 \bar{y}_1 y_2 + \bar{x}_2 \bar{y}_1 y_2 \quad (5)
 \end{aligned}$$

となる。(5) 式の $\bar{x}_1 y_1 y_2, x_1 \bar{y}_1 y_2, \bar{x}_2 \bar{y}_1 y_2, \bar{x}_1 y_1 \bar{y}_2, x_1 y_1 y_2, x_2 y_1 y_2$ はハザードフリーにするための項であり、 $x_1 x_2 y_2, x_2 y_1 y_2$ は Y_1 と Y_2 に共用できる。この設計では \bar{y}_1, \bar{y}_2 のためにインバータが必要となり、回路の遅れは 3 ゲート分になる。ゲート数は 16、入力数は 52 である。

つぎに表 2 (b) に SR ラッチの励起表を用いて

表2 設計例
Table 2 Example.

(a) 遷移表 Transition table.					(b) 励起表 Excitation table.					
y_2y_1		x_2x_1			y_2y_1		x_2x_1			
	a	00	01	11	10		00	01	11	10
00	a	Ⓐ	d	Ⓐ	Ⓐ	00	00	10	00	00
01	b	Ⓑ	Ⓑ	a	Ⓑ	01	01	01	00	01
11	c	d	b	Ⓒ	Ⓒ	11	10	01	11	11
10	d	Ⓓ	Ⓓ	c	a	10	10	10	11	00

(c) S_1R_1 の制御表 Control table of S_1R_1 .					(d) S_1R_1 の制御表 Control table of S_1R_1 .					
y_2y_1		x_2x_1			y_2y_1		x_2x_1			
		00	01	11	10		00	01	11	10
00		0d	10	0d	0d	00	0d	0d	0d	0d
01		0d	0d	0d	0d	01	d0	d0	01	d0
11		d0	01	d0	d0	11	01	d0	d0	d0
10		d0	d0	d0	01	10	0d	0d	10	0d

S_2R_2 と S_1R_1 の制御表を作ると表2(c), (d)になる。これより

$$\begin{aligned}
 S_1 &= x_1x_2y_2 \\
 R_1 &= \bar{x}_1\bar{x}_2y_2 + x_1x_2\bar{y}_2 \\
 S_2 &= x_1\bar{x}_2\bar{y}_1 \\
 R_2 &= x_1\bar{x}_2y_1 + \bar{x}_1x_2\bar{y}_1 \quad (6)
 \end{aligned}$$

となり、ゲート数は10、入力数は28である。この回路図を図8に示す。

4. フリップフロップの設計

同期回路に広く使われているフリップフロップ(以下FFと略す)は非同期回路理論に基づいて設計できるが⁴⁾、ここでは特殊なFFを2種類設計し、普通の方法とSRラッチによる設計を比較検討する。

4.1 エッジセンシング・マスタスレーブ型 JKFF

エッジセンシング・マスタスレーブ型 JKFF はクロックの立上りで J, K 入力を取り込み、クロックの

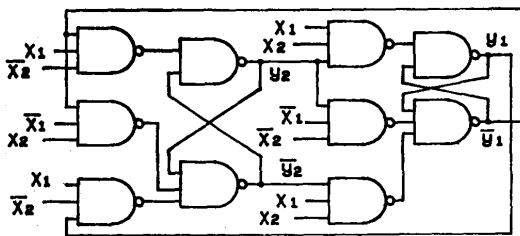


図8 設計例
Fig. 8 Example.

表3 エッジセンシング・マスタスレーブ型 JK フリップフロップ

Table 3 Edge-sensing master-slave JK flip-flop.

(a) 遷移表 Transition table ¹⁾ .										
	CJK					CJK				z_2z_1
	000	001	011	010	100	101	111	110		z_2z_1
a	Ⓐ	Ⓐ	Ⓐ	Ⓐ	e	e	b	b		00
e	a	a	a	a	Ⓔ	Ⓔ	Ⓔ	Ⓔ		00
b	c	c	c	c	Ⓑ	Ⓑ	Ⓑ	Ⓑ		01
c	Ⓒ	Ⓒ	Ⓒ	Ⓒ	f	d	d	f		11
f	c	c	c	c	Ⓕ	Ⓕ	Ⓕ	Ⓕ		11
d	a	a	a	a	Ⓖ	Ⓖ	Ⓖ	Ⓖ		10

(b) 励起表 Excitation table.									
y_2y_1	CJK					CJK			
	000	001	011	010	100	101	111	110	
a	000	000	000	000	100	100	001	001	
b	001	011	011	011	001	001	001	001	
c	011	011	011	011	111	010	010	111	
d	010	000	000	000	010	010	010	010	
e	100	000	000	000	100	100	100	100	
	101	ddd	ddd	ddd	ddd	ddd	ddd	ddd	
f	111	011	011	011	111	111	111	111	
	110	ddd	ddd	ddd	ddd	ddd	ddd	ddd	

(c) S_1R_1 の制御表 Control table of S_1R_1 .									
y_2y_1	CJK					CJK			
	000	001	011	010	100	101	111	110	
000	0d	0d	0d	0d	0d	0d	10	10	
001	d0	d0	d0	d0	d0	d0	d0	d0	
011	d0	d0	d0	d0	d0	01	01	d0	
010	0d	0d	0d	0d	0d	0d	0d	0d	
100	0d	0d	0d	0d	0d	0d	0d	0d	
101	dd	dd	dd	dd	dd	dd	dd	dd	
111	d0	d0	d0	d0	d0	d0	d0	d0	
110	dd	dd	dd	dd	dd	dd	dd	dd	

立下りで出力が変化する¹⁾。このFFの遷移表(表3(a))は文献1)の図15.9に示されているが、導き方を簡単に示す。FFの出力は z_2 であり、クロック C が0のとき安定状態 a (出力0) または c (出力1) にある。状態 a で C が 0→1 となると、J=1 ならば b に移る。b は次にクロックが0になると出力が変化して状態 c に移る準備状態である。J が0ならば状態 e に移り、次にクロックが0になると a に戻る。出力が変化しない入力による状態遷移である。状態 b, e と

も $C=1$ の間に J, K が変化しても状態は変わらない。すなわち C が $0 \rightarrow 1$ になるときの J の値 (K は無関係) によって状態が b か e に遷移する。普通のマスタースレーブ型 JKFF では $CJ=1$ で状態 a から b に移るが、 C, J のどちらが先に変化しても b に移るため状態 e はない (文献4 参照)。 $C=0$ で状態 c にあるときも J を K に変えて同様に導くことができる。

表3 (a) の6状態に状態変数 $y_1 y_2 y_3$ をレースがないように割り当てる (表3 (b) の左端欄)。この割当ては変数の順序と $0, 1$ の入替えを除いてユニークである。表3 (b) から Y_1, Y_2, Y_3 と出力 z_1, z_2 をハザードフリーに求めると、

$$\begin{aligned} Y_1 &= y_1 y_3 + y_1 \bar{y}_2 + y_1 \bar{C} + y_1 \bar{K} + C J \bar{y}_2 \bar{y}_3 \\ Y_2 &= \bar{C} y_1 + C y_2 + y_1 y_2 \\ Y_3 &= C y_3 + C \bar{J} \bar{y}_1 \bar{y}_2 + C \bar{K} y_1 y_2 \\ z_1 &= y_1 \\ z_2 &= y_2 \end{aligned} \quad (7)$$

となる。 y_1, y_2, y_3 のためのインバータを含め16ゲート、入力総数40である。

次に表3 (b) から $S_1 R_1$ の制御表を作ると表3 (c) になる ($S_2, R_2; S_3, R_3$ も同様)。これから各 S, R を求めると、

$$\begin{aligned} S_1 &= C J \bar{y}_2 \bar{y}_3 & R_1 &= C K y_2 \bar{y}_3 \\ S_2 &= \bar{C} y_1 & R_2 &= \bar{C} \bar{y}_1 \\ S_3 &= C \bar{J} \bar{y}_1 \bar{y}_2 + C \bar{K} y_1 y_2 & R_3 &= \bar{C} \end{aligned} \quad (8)$$

となり、12ゲート、入力総数33である (図9)。したがって、この設計ではSRラッチによる設計が有効である。図9の出力側のSRラッチ2段から成る回路 (y_1, y_2) は広く使われているマスタースレーブ型 JKFF と同じであり、入力側にクロックのエッジで取り込んだ入力を保持するSRラッチ (y_3) が付いていることがわかる。

(7)式を(1)式の形に変形すると

$$\begin{aligned} Y_1 &= C J \bar{y}_2 \bar{y}_3 + (\bar{C} + \bar{K} + \bar{y}_2 + y_3) y_1 \\ Y_2 &= \bar{C} y_1 + (C + y_1) y_2 \\ Y_3 &= C \bar{J} \bar{y}_1 \bar{y}_2 + C \bar{K} y_1 y_2 + C y_3 \end{aligned} \quad (9)$$

となり、これより S, R を求めると(8)式と同じにな

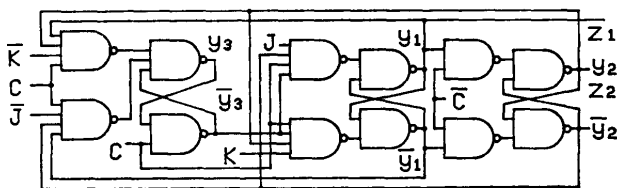


図9 エッジセンシング・マスタースレーブ型 JK フリップフロップ
Fig. 9 Edge-sensing master-slave JK flip-flop.

表4 ダブルエッジトリガ型 JK フリップフロップ
Table 4 Double-edge-triggered JK flip-flop.

(a) 遷移表^{a)}
Transition table^{a)}

	CJK							
	000	001	011	010	110	111	101	100
1	①,0	①,0	2,0	2,0	4,0	4,0	①,0	①,0
2	1,0	1,0	②,0	②,0	3,1	②,1	②,1	3,1
3	③,1	4,1	4,1	③,1	③,1	2,1	2,1	③,1
4	3,1	④,1	④,1	3,1	④,0	④,0	1,0	1,0

(b) 励起表 ($Y_1 Y_2 Y_3$)
Excitation table ($Y_1 Y_2 Y_3$).

	$y_1 y_2$	CJK							
		000	001	011	010	110	111	101	100
1	00	000	000	010	010	100	100	000	000
2	01	000	000	010	010	111	011	011	111
3	11	111	101	101	111	111	011	011	111
4	10	111	101	101	111	100	100	000	000

(c) $S_1 R_1$ の制御表
Control table of $S_1 R_1$.

	$y_1 y_2$	CJK							
		000	001	011	010	110	110	101	100
00	00	0d	0d	0d	0d	10	10	0d	0d
01	01	0d	0d	0d	0d	10	0d	0d	10
11	11	d0	d0	d0	d0	d0	01	01	d0
10	10	d0	d0	d0	d0	d0	d0	01	01

る。また、 $S_1 R_1 = S_2 R_2 = S_3 R_3 = 0$ は満たされている。この設計では Y_1, Y_2, Y_3 をハザードフリーに求め、式の変形によりSRラッチの入力を求めたのと、SRラッチによる設計法でSRラッチの入力を求めたのと、結果は同じになる。

4.2 ダブルエッジトリガ型 JKFF

ダブルエッジトリガ型 JKFF は、クロック C の立上りと立下りの両方で J, K 入力にตอบสนองする JKFF である⁶⁾。この FF の8状態の遷移表と4状態に簡単化された遷移表が文献6)の表7と表8 (a), (b) に示されている。ここでは簡単化された遷移表 (文献6)の表8 (b)) から設計する。状態割当ても文献6)に従い、遷移表と励起表を表4 (a), (b) に示す。表4 (a) は簡単化された遷移表であるから直接導くことは困難であるが簡単に説明する。状態1 (3) は出力が $0 (1)$ で、入力 $J (K)$ が 0 なら安定した状態である。状態1で J が 1 になると次のクロックの変化で出力が変わるため、 $C=0$ なら状態2へ、 $C=1$ なら状態4へ移る。状態2に $\bar{C}J=1$ の間

とどまり, C が $0 \rightarrow 1$ となると $K=1$ なら出力が反転して 1 になり状態 2 にとどまる. $K=0$ なら $J=1$ によって出力が 1 となって状態 3 に移る. これはクロックの立上りに応答する部分である. なお, C が 1 になる前に J が 0 になると状態も 1 に戻る. 状態 4 は上と同様にクロックの立下りに応答する部分である. 状態 3 は出力 1 の状態からの遷移で状態 1 の場合と同じように考えることができる.

表 4 (b) より Y_1, Y_2, Q は

$$\begin{aligned} Y_1 &= \bar{C}y_1 + J\bar{y}_2(C+y_1) \\ &\quad + \bar{K}y_2[C+y_1] + J\bar{K}(C+y_1) \\ Y_2 &= Cy_2 + J\bar{y}_1(\bar{C}+y_2) + \bar{K}y_1[\bar{C}+y_2] + J\bar{K}y_2 \\ Q &= [\bar{C}y_1] + [Cy_2] + [\bar{K}y_2(C+y_1)] \quad (10) \end{aligned}$$

となる⁶⁾. (10)式を NAND 3段で構成した回路図が文献 6) の図 4 に示されている. \bar{y}_1, \bar{y}_2 のためのインバータを含め 15 ゲート, 39 入力であり, 遅れは 4 ゲート分である.

表 4 (b) から S_1R_1 の制御表を作ると表 4 (c) となり (S_2, R_2 も同様), これより $S_1, R_1; S_2, R_2, Q$ は

$$\begin{aligned} S_1 &= CJ\bar{y}_2 + C\bar{K}y_2 \\ R_1 &= C\bar{J}\bar{y}_2 + CKy_2 \\ S_2 &= \bar{C}\bar{J}\bar{y}_1 + \bar{C}\bar{K}y_1 \\ R_2 &= \bar{C}\bar{J}y_1 + \bar{C}\bar{K}y_1 \\ Q &= \bar{C}y_1 + Cy_2 + y_1y_2 \\ &= \bar{C}\bar{K}y_1 + \bar{C}\bar{K}y_1 + C\bar{K}y_2 + CKy_2 + y_1y_2 \quad (11) \end{aligned}$$

となり, ゲート数は 14, 入力総数は 43 である. この回路 (図 10) は出力 Q をハザードフリーにするための項 y_1y_2 を作る NAND ゲートを除いて, 対称回路となっており, 文献 6) の回路より実用上優れている.

5. あとがき

SR ラッチによる非同期式順序回路の設計法を述べた. SR ラッチの S, R 入力にハザードによるグリッチがある場合を検討し, 静的ハザードの極性によって出力に影響する場合としない場合を明確にした. 出力に影響を及ぼさない静的ハザードしか論理的に発生しない 2 段回路で S, R 入力を構成すると, 非同期回路を同期回路と同様の方法で設計できることを示した.

SR ラッチによる方法と普通の設計法を比較検討

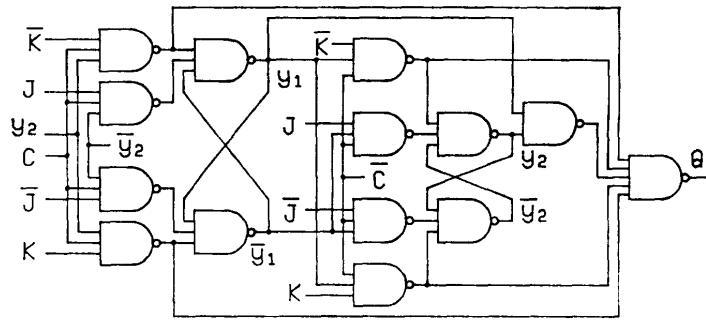


図 10 ダブルエッジトリガ型 JK フリップフロップ
Fig. 10 Double-edge-triggered JK flip-flop.

し, 具体例として JKFF を 2 つ設計した. これら 2 つの方法のどちらで設計する方が簡単な回路が得られるかについては一般論はないが, ハザードフリーにするための冗長項が多い場合には, SR ラッチによる方法の方が簡単な回路が得られる. なお, 本質的ハザードは避けられない.

参 考 文 献

- 1) Hill, F. J. and Peterson, G. R.: *Introduction to Switching Theory and Logical Design*, 3rd ed., John Wiley & Sons, New York (1981).
- 2) Mano, M. M.: *Digital Design*, Prentice-Hall Inc., Englewood Cliffs, New Jersey (1984).
- 3) Muroga, S. (室賀, 笹尾訳): 論理設計とスイッチング理論, 共立出版, 東京 (1981).
- 4) 井上訓行, 奥川峻史: 非同期回路理論によるラッチとフリップフロップの設計, 信学技報, CPSY 86-58 (1987).
- 5) 奥川峻史: LSI による論理設計, 共立出版, 東京 (1987).
- 6) Unger, S. H.: Double-Edge-Triggered Flip-flops, *IEEE Trans. Comput.*, Vol. C-30, No. 6, pp. 447-451 (1981).
- 7) Calvo, J., Acha, J. I. and Valencia, M.: Asynchronous Modular Arbiter, *IEEE Trans. Comput.*, Vol. C-35, No. 1, pp. 67-70 (1986).
- 8) 岡本卓爾, 阿部山友邦: NAND ラッチを対象としたメタステーブル動作の評価, 信学論 (D), Vol. J 68-D, No. 6, pp. 1210-1217 (1985).

(昭和 62 年 5 月 8 日受付)

(昭和 62 年 9 月 9 日採録)

**井上 剛行 (正会員)**

昭和 18 年生. 昭和 41 年京都大学工学部数理工学科卒業. 日本電気(株)を経て昭和 44 年より京都産業大学に勤務. 現在同助教授. 論理設計, マイクロコンピュータの応用などに興味を持つ. 「コンピュータの論理設計」(共訳, 共立出版) など. 電子情報通信学会会員.

**奥川 峻史 (正会員)**

昭和 9 年京都生. 昭和 34 年京都大学工学部電子工学科卒業. 京都産業大学理学部教授. 工学博士. 論理設計, 並列計算機アーキテクチャに興味をもつ. 「並列計算機」(共立出版, 共訳), 「ビットスライス LSI とその応用」(共立出版, 訳補), 「LSI による論理設計」(共立出版) など. 電子情報通信学会, IEEE 各会員.