

## 4 J-7

## 同期式順序回路の機能情報抽出について

大村昌彦 安浦寛人 田丸啓吉

京都大学工学部

## 1 はじめに

論理回路の設計においては、設計された回路が仕様を満足するかどうかを確かめる設計検証の問題が重要である。設計検証の有効な手法の一つとして、設計された論理回路レベルの記述から機能レベルの記述を抽出し、仕様と比較するという手法が、提案されている [1]。我々も、組合せ回路の回路記述から論理演算や算術演算等を抽出し、人間の目で見えてわかりやすい機能表で記述する手法について研究を行なってきた [2]。本稿では、その手法を同期順序回路に適用した結果について報告する。

## 2 準備

本稿では図1のような構造の同期式順序回路を対象とする。この回路は、組合せ回路部(C.C.)と記憶部(幾つかのD-FF)から構成されており、各D-FFには、同一のクロック入力に加えらる。この回路の入力変数は $X_1, \dots, X_i$ であり、出力変数は $Y_1, \dots, Y_j$ である。各FFにおいては、 $Q_1, \dots, Q_k$ の値が保持されている。各D-FFの入力 $P_1, \dots, P_k$ は、クロック入力を与えられた時に、それぞれ $Q_1, \dots, Q_k$ に代入されるので、 $Q_1, \dots, Q_k$ を現状態変数、 $P_1, \dots, P_k$ を次状態変数と呼ぶ。出力変数及び次状態変数は、入力変数と現状態変数に依存する。これらの状態変数をベクトルと見なしたものを状態と呼び、本稿では $S$ で表す。 $k$ 個の状態変数によって、 $2^k$ 個の状態を表すことが出来る。

さて、われわれの提案している機能情報抽出の手法においては、入力変数をコントロール系のものでデータ系のもので区別し、コントロール系入力の値によって回路の機能を分割して表示している。これと同様の概念を、状態変数に導入する。設計者は、回路のデータ系の部分と制御系の部分の区別を意識して設計を行なっているが、この時出力や次状態を制御する働きをしていると設計者が意図した状態変数をコントロール系状態変数、またレジスタの出力などのようにその値そのものが何らかのデータであるとみなした状態変数をデータ系状態変数と呼ぶことにする。機能情報抽出を行なう際には、コントロール系状態変数とデータ系状態変数の区別は、付加情報としてあらかじめ与えられているものと仮定する。

## 3 同期式順序回路の機能情報抽出の手法

図1に示した回路の動作は、次のように記述できる。

- (1) 組合せ回路部において、入力と現状態から、出力と次状態が決まる。
- (2) クロック入力に加えられた時に、(1)で定められた次状態が一気に決まる。

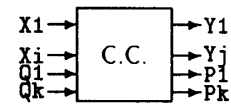
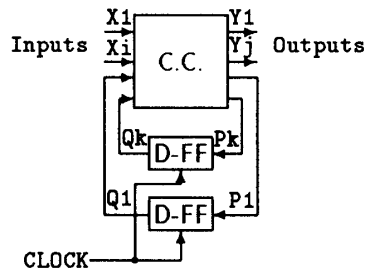


図1: 同期式順序回路 図2: 図1の組合せ回路部

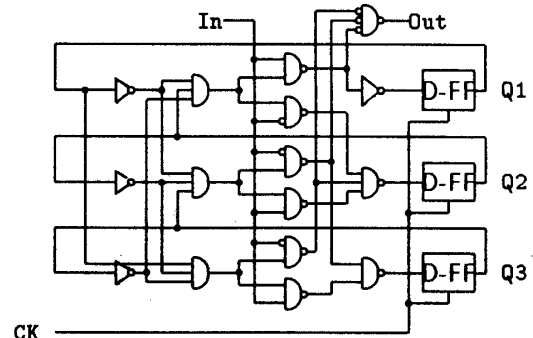
機能情報抽出の立場から考えると、(2)の動作よりも、(1)の動作が重要であり、この動作のみに注目することにする。そのために、図1においてクロック入力とD-FFを取り除いた図2のような回路を考える。

これは、入力( $X_1, \dots, X_i, Q_1, \dots, Q_k$ )、出力( $Y_1, \dots, Y_j, P_1, \dots, P_k$ )なる組合せ回路であるので、従来の手法を用いて機能情報抽出を行なうことが出来ると考えられる。ここで問題となるのは、入力 $Q_1, \dots, Q_k$ をどのように取り扱うのか、また出力 $P_1, \dots, P_k$ に現れた結果をどのように解釈するのか、ということである。これらのことは、場合によって異なるので、以下に実例を挙げながら説明する。

## 4 状態変数がコントロール系の場合

図3(a)に示した回路について、3.で説明した手法に基づき、機能情報抽出を行なってみる。

3つのD-FFの出力(現状態変数)を $Q_1, Q_2, Q_3$ 、入力(次状態変数)を $P_1, P_2, P_3$ とし、クロック入力及びD-FFを取り除くと、4入力( $Q_1, Q_2, Q_3, In$ )、4出力( $Out, P_1, P_2, P_3$ )の組合せ回路が得られる。ここで、3つの状態変数は全て、コントロール系の働きをしているものとする。また、入力 $In$ もコントロール系入力とする。この回路に従来の手法を適用すれば、同図(b)の



(a) 回路図

図3: 状態変数がコントロール系の場合の例

$Q_1 Q_2 Q_3 In$	$P_1 P_2 P_3 Out$	(d-1)	$Q_1 Q_2 Q_3 In$	$P_1 P_2 P_3 Out$
0 0 0 0	0 0 0 0		0 0 1 0	0 0 1 1
0 0 0 1	0 0 0 1		0 0 1 1	0 1 0 0
0 0 1 0	0 0 1 1			
0 0 1 1	0 1 0 0	(d-2)	$Q_1 Q_2 Q_3 In$	$P_1 P_2 P_3 Out$
0 1 0 0	0 1 0 1		0 1 0 0	0 1 0 1
0 1 0 1	1 0 0 0		0 1 0 1	1 0 0 0
0 1 1 $In$	0 0 0 1			
1 0 0 0	0 1 0 0	(d-3)	$Q_1 Q_2 Q_3 In$	$P_1 P_2 P_3 Out$
1 0 0 1	0 0 1 1		1 0 0 0	0 1 0 0
1 0 1 $In$	0 0 0 1		1 0 0 1	0 0 1 1
1 1 0 $In$	0 0 0 1	(d-4)	$Q_1 Q_2 Q_3 In$	$P_1 P_2 P_3 Out$
1 1 1 $In$	0 0 0 1		0 0 1 0	0 0 1 1

(b) 機能表 (1)

現	$In = 0$		$In = 1$	
	次	Out	次	Out
$S_0$	$S_0$	0	$S_0$	1
$S_1$	$S_1$	1	$S_2$	0
$S_2$	$S_2$	1	$S_4$	0
$S_3$	$S_0$	0	$S_0$	0
$S_4$	$S_2$	0	$S_1$	1
$S_5$	$S_0$	1	$S_0$	1
$S_6$	$S_0$	1	$S_0$	1
$S_7$	$S_0$	1	$S_0$	1

(c) 状態遷移表 (1)

$Q_1 Q_2 Q_3 In$	$P_1 P_2 P_3 Out$
0 0 1 0	0 0 1 1
0 0 1 1	0 1 0 0
0 1 0 0	0 1 0 1
0 1 0 1	1 0 0 0
1 0 0 0	0 1 0 0
1 0 0 1	0 0 1 1
1 0 1 $In$	0 0 0 1
1 1 0 $In$	0 0 0 1
1 1 1 $In$	0 0 0 1

(d) 機能表 (2)

現	$In=0$		$In=1$	
	次	Out	次	Out
$S_1$	$S_1$	1	$S_2$	0
$S_2$	$S_2$	1	$S_4$	0
$S_4$	$S_2$	0	$S_1$	1

(e) 状態遷移表 (2)

図 3 (続き)

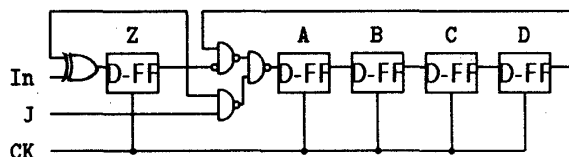
ような機能表が得られる。この機能表の1行目は、入力 ( $Q_1, Q_2, Q_3, In$ ) が (0,0,0,0) の時、出力 ( $P_1, P_2, P_3, Out$ ) が (0,0,0,0) になる、ということを表しているが、これは見方を変えれば、状態  $Q_1 Q_2 Q_3$  が 000 の時に入力  $In = 0$  ならば、次状態  $P_1 P_2 P_3$  が 000 になり、出力  $Out = 0$  になる、と読み取ることも出来る。したがって、この機能表は状態遷移表である、とみなすこともできる。もし、次のような状態割当

$S_0 : 000 \quad S_1 : 001 \quad S_2 : 010 \quad S_3 : 011$   
 $S_4 : 100 \quad S_5 : 101 \quad S_6 : 110 \quad S_7 : 111$

が与えられていれば、同図 (c) の状態遷移表で表すことができる。

この手法の問題点としては、実際に存在しない状態に対する動作まで記述してしまう、ということが挙げられる。例えば、この回路の初期状態が  $S_1$  であると指定されていたとすると、実際には状態  $S_1, S_2, S_4$  しか遷移し得ないのである。例えば、8つの状態変数で256個の状態が表せるが、1-hot コードで状態割当が行なわれているとすれば、実際には8個の状態しか使用しない。このような回路に対して、256状態の動作を記述するのは、非常に無駄である。この問題点の解決策として、初期状態から実際に遷移し得る状態のみについて機能表を作成し、それらを連結して状態遷移表を作成する、という手法が考えられる (図 3(d))。

まず、( $Q_1 Q_2 Q_3$ ) を初期状態 (001) とおいて、機能表を作成する (図 (d-1))。この表より、次状態は (001) 又は (010) であるから、今度は ( $Q_1 Q_2 Q_3$ ) を (010) とお



(a) 回路図

$Q_z$	$In$	$Q_a$	$Q_b$	$Q_c$	$Q_d$	$J$	$P_z$	$P_a$	$P_b$	$P_c$	$P_d$
0	0	$Q_a$	$Q_b$	$Q_c$	$Q_d$	$J$	0	$Q_d$	$Q_a$	$Q_b$	$Q_c$
0	1	$Q_a$	$Q_b$	$Q_c$	$Q_d$	$J$	1	$Q_d$	$Q_a$	$Q_b$	$Q_c$
1	0	$Q_a$	$Q_b$	$Q_c$	$Q_d$	$J$	1	$J$	$Q_a$	$Q_b$	$Q_c$
1	1	$Q_a$	$Q_b$	$Q_c$	$Q_d$	$J$	0	$J$	$Q_a$	$Q_b$	$Q_c$

(b) 機能表

図 4: データ系の状態変数を含む場合の例

て、機能表を作成する (図 (d-2))。このようにして、遷移し得る全ての状態において機能表を作成する (図 (d-1) ~ (d-3))。最後にこれらの機能表を連結して、一つの機能表にする (図 (d-4))。これより、図 (e) のような状態遷移表が得られる。

### 5 状態変数がデータ系のものを含む場合

図 4(a) の回路についても、前節と同様に機能情報抽出を行なってみる。

5つの D-FF A, B, C, D, Z の出力を  $Q_a, Q_b, Q_c, Q_d, Q_z$ 、入力を  $P_a, P_b, P_c, P_d, P_z$  とし、D-FF とクロック入力を取り除く。この回路においては、状態変数  $Q_a, Q_b, Q_c, Q_d$  はデータ系で、 $Q_z$  のみがコントロール系である。入力変数  $J$  をコントロール系入力、 $In$  をデータ系入力とみなして機能表を作成すると、図 (b) のようになった。 $Q_z = 0$  の時を状態  $S_0$ 、 $Q_z = 1$  の時を状態  $S_1$  とすると、これは状態  $S_0$  においては巡回シフトが、 $S_1$  においては非巡回シフトが行なわれ最下位ビットに入力  $J$  の値が代入される、という動作を表しており、入力  $In$  の値によって状態を変化させることが出来る。この機能表は、状態の遷移を表しているというよりはむしろ、レジスタ間のデータの移動を表していると考えられ、レジスタ転送レベルの記述であると言える。

### 6 おわりに

本稿では、順序回路の機能情報抽出の手法に関して、対象を図 1 のような構造で実現できる回路に限定して、考察を行なった。今後の課題としては、抽出された情報を如何にして自動的に解釈するか、また D-FF でなく JK-FF などを用いて構成される回路、あるいは非同期的順序回路をどのように取り扱うか、などが挙げられる。

### 参考文献

- [1] J.C.Madre, et al., "Proving Circuit Correctness using Formal Comparison Between Expected and Extracted Behaviour", 26th DAC.
- [2] M.Ohmura, et al., "Extraction of Functional Information from Combinational Circuits", ICCAD90.