

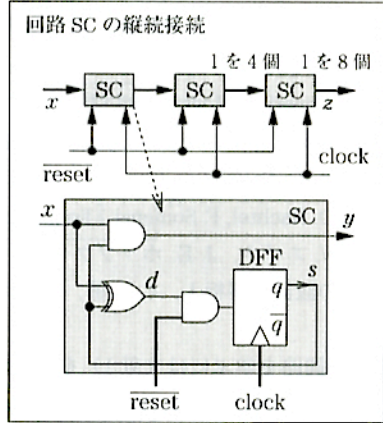
### 5.14 縦続接続による設計\*

#### 8 カウンタ

図 x に 1 が 8 回入力されるたびに、z に 1 を出力する

直接状態方程式を解く

現状態	次状態				出力 z	状態割当て		
	入力 x	0	1	0		1	$q_1$	$q_2$
$Q_0$	$Q_0$	$Q_1$	0	0	$Q_0$	0	0	0
$Q_1$	$Q_1$	$Q_2$	0	0	$Q_1$	0	0	1
$Q_2$	$Q_2$	$Q_3$	0	0	$Q_2$	0	1	0
$Q_3$	$Q_3$	$Q_4$	0	0	$Q_3$	0	1	1
$Q_4$	$Q_4$	$Q_5$	0	0	$Q_4$	1	0	0
$Q_5$	$Q_5$	$Q_6$	0	0	$Q_5$	1	0	1
$Q_6$	$Q_6$	$Q_7$	0	0	$Q_6$	1	1	0
$Q_7$	$Q_7$	$Q_0$	0	1	$Q_7$	1	1	1



この章では、ここまで、順序回路を設計する次のような手法を紹介した。

1°: 状態を用いて状態遷移表を作成し、2°: 状態数を削減した後、3°: 状態割当てにより、状態を符号化し、4°: 論理値に関する状態遷移表および出力表を生成する。5°: これから状態遷移関数および出力関数を求め、6°: 状態変数を記憶しておく D フリップフロップへの入力方程式から ~~入力~~ 回路を、7°: 出力方程式から出力回路を作成する。 状態遷移

例えば、入力 x に 1 が 8 回入力されるたびに、出力 z から 1 を出力する 8 カウンタの場合には、表に示す 8 個の状態に対する状態遷移表と状態割当てを与え、D フリップフロップを用いて状態変数  $q_1, q_2, q_3$  を記憶すれば、後は、これまでに述べた設計手順で順序回路を実現できる。

これ以外の方法として、複数の順序回路を直列に接続（縦続接続）することにより、所望の順序回路を実現することもできる。例えば、8 カウンタは、図に示すように、1 が 2 回入力されるたびに 1 を出力する 2 カウンタ SC を 3 個直列に接続することにより実現できる。ここで、2 カウンタ SC は、この章の前半(5.6 節)で示した 2 カウンタ回路に、reset 信号が 1 ( $\overline{\text{reset}}=0$ ) のときに D フリップフロップを初期化するための AND ゲートを付加したものである。このような縦続接続により、既存の回路を用いて容易に 8 カウンタを実現できることがわかる。ただし、この方法では、入力 x から出力 z まで、AND ゲートが 3 段連なっており、そのため、x が変化してから z が変化するまでに時間を要することがある。これに関しては 6 章で述べる。

複数の順序回路を並列接続して所望の順序回路を実現することもできる。すなわち、全体の入力を各順序回路に入力し、各順序回路からの出力を用いて全体の出力を生成したり（ムーア型）、各順序回路の出力と全体の入力をを用いて全体の出力を生成したり（ミーリ型）することもできる。