

【1】

(i) =====

問題の指示に従って、データ入力 I_3, I_2, I_1, I_0 を4ビットのシフトレジスタに入れ、 I_3, I_2, I_1, I_0 の中の1の個数を2ビットのアップカウンタで覚えることにすると、データパスの入力として、 I_3, I_2, I_1, I_0 の他に、シフトレジスタ用の制御信号 S_1, S_0 およびアップカウンタ用の制御信号 $reset, enable$ が必要となる。さらに、シフトレジスタの入力 I_L の値を入力する必要があるが、この I_L は常に $I_L = 0$ であるので、ここでは制御回路からの入力としないでおく。

データパスの出力としては、1の個数を示す $carry, c_1, c_0$ の他に、シフトレジスタの最下位ビットの値 q_0 およびシフトレジスタ内の全ビット $q_0 \sim q_3$ が0になったとき1になる出力 $zero$ が必要となる。

従って、これらを纏めると下記となる。

データパスの入力	
I_3, I_2, I_1, I_0	: 4ビットのデータ入力
S_1, S_0	: シフトレジスタ用の制御信号
$reset, enable$: アップカウンタ用の制御信号
データパスの出力	
$carry, c_1, c_0$: 1の個数を示す3ビットのデータ出力
q_0	: シフトレジスタの最下位ビット(LSB)
$zero$: シフトレジスタ内の全ビット $q_0 \sim q_3$ が0になったとき1, さもなくば0

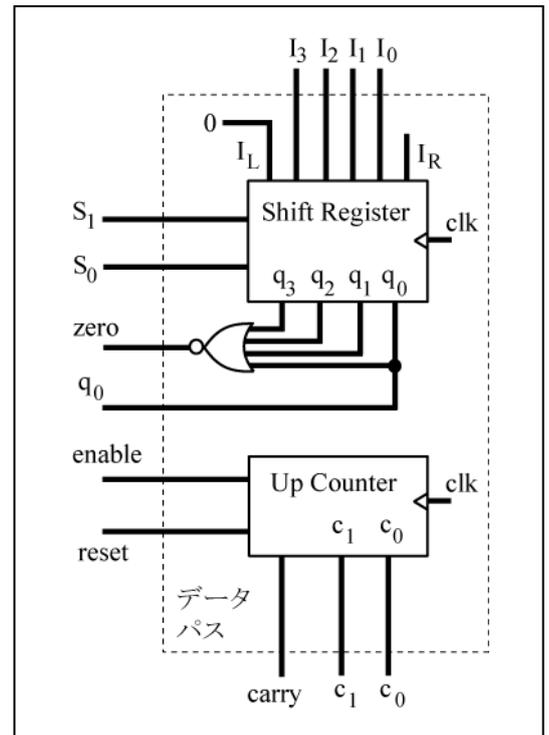
(ii) =====

シフトレジスタ内の全ビット $q_0 \sim q_3$ が 0 のとき 1, さもなくば 0 となる出力 zero を生成するには, $q_3+q_2+q_1+q_0$ の否定をとればよいから, 右図のような回路ができる.

この回路は, $q_3 \sim q_0$ の中の 1 の個数が 0 であるか否かを 1 クロックで判定できることを示している. 1 の個数が 0 であるか否かを 1 クロックで判定できるのであれば, 1 の個数が k ($k = 1, 2, 3, 4$) であるか否かも 1 クロックで判定でき, carry, c_1 , および c_0 がそれぞれ 1 になる場合を I_3, I_2, I_1, I_0 の論理式で表すこともできるはずである. すなわち, 4 ビットの入力 I_3, I_2, I_1, I_0 の中の 1 の個数を, 3 ビットの出力 carry, c_1, c_0 として出す組合せ回路を作成することができる.

このように, デジタル回路を作成する際, 組合せ回路で実現する(1 クロックで実行する)か, あるいは順序回路を用いて実現する(複数クロックで実行する)かという自由度があり, 回路規模, 動作速度などの評価指標を勘案して, 最適な実現方法を見出す必要がある. ここでは, 問題に指定されたように, 順序回路を用いて, I_3, I_2, I_1, I_0 の中の 1 の個数を数えることにする.

なお, このデータパス回路は, レジスタと演算器(ここでは, NOR ゲート)からできているので, データパス内での転送は無いが, レジスタ転送レベル (RT レベル) の回路であるということもできる.



(iii) =====

制御回路の入出力は, start および done の他に, データパスと制御回路の間でやりとりしなければならない信号があり, それらを纏めると下記ようになる.

制御回路の入力		
start	: 処理の開始を指示する信号	
ステータス信号(データパスから制御回路への信号)		
q ₀	: シフトレジスタの LSB	
zero	: q ₃ +q ₂ +q ₁ +q ₀ = 0 であれば, zero = 1	
	: q ₃ +q ₂ +q ₁ +q ₀ = 1 であれば, zero = 0	
制御回路の出力		
done	: 処理の終了を知らせる信号	
データパスへの制御信号(動作の詳細は下記の表参照)		
S ₁ , S ₀	: シフトレジスタ用	
reset, enable	: アップカウンタ用	

制御入力		動作
S ₁	S ₀	
0	0	変化なし
0	1	入力取り込み
1	0	左 1 ビットシフト
1	1	右 1 ビットシフト

制御入力		動作
reset	enable	
0	0	変化なし
0	1	カウントアップ
1	*	値を 0 にする

制御回路は, start = 1 になったならば, まず, シフトレジスタにデータ入力 I₃, I₂, I₁, I₀ を取り込み, アップカウンタを reset する. その後, q₀ = 1 ならばアップカウンタをカウントアップするという操作を, zero = 1 になるまで繰り返す. 従って, シフトレジスタおよびアップカウンタの制御信号の意味を考えれば, 次のような動作をすればよいことが分かる.

- 0°: start = 1 になったならば, 1° 以降の操作を実行する. start = 0 である間は, この操作 0° を繰り返す. どちらの場合も出力 done は 0 とする.
- 1°: シフトレジスタにデータ入力 I₃, I₂, I₁, I₀ を取り込み, アップカウンタを reset する.
- 2°: zero = 0 の間, 以下の操作を繰り返す.
 - 2-1°: q₀ = 1 であれば, アップカウンタをカウントアップする.
 - 2-2°: I_L = 0 として, シフトレジスタを右に 1 ビットシフトする.
- 3°: done を 1 とし, carry, c₁, c₀ を出力して, 0° に戻る.

(iv) =====

ここでは、ミラー型順序回路にするので、出力 (done, reset, enable, S_1 , S_0) を、入力 (start, q_0 , zero) を用いて決定できる。そうすると、上記の $0^\circ \sim 3^\circ$ の操作を実行するには、次のような4つの状態があればよいことが分かる。

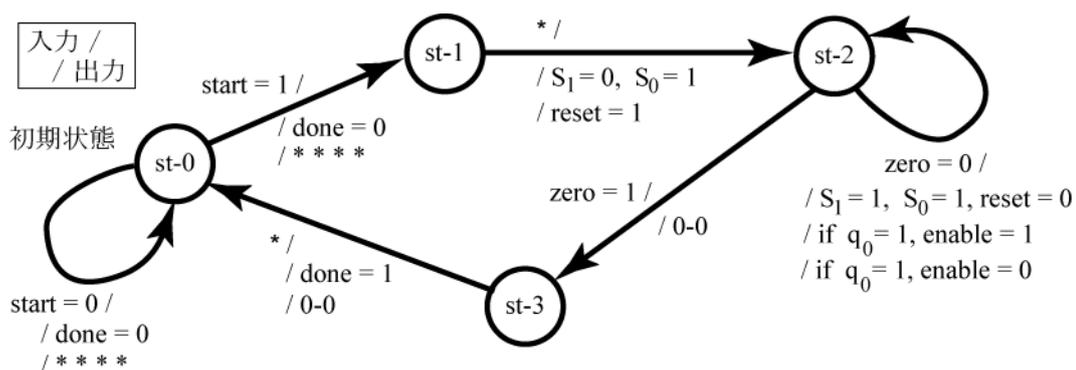
状態 st-0: 入力に関係なく done = 0 を出力し、他の出力は 0 でも 1 でもよい (don't care である).
 start = 1 であれば、次は状態 st-1 に行く。
 start = 0 であれば、次は状態 st-0 に戻る。

状態 st-1: 入力に関係なく次の出力をし、状態 st-2 に行く。
 シフトレジスタにデータを入力するため、 $S_1 = 0$, $S_0 = 1$ を出力する。アップカウンタを reset するため、reset = 1 を出力する。enable は don't care で、done は 0 である。

状態 st-2:
 zero = 0 であれば、 $q_3+q_2+q_1+q_0 \neq 0$ であるので、シフトレジスタを右に1ビットシフトするため、 $S_1 = 1$, $S_0 = 1$ を出力する。また、 q_0 の値に従ってアップカウンタをカウントアップするため、reset = 0 を出力し、 $q_0 = 1$ であれば、Enable = 1 を、 $q_0 = 0$ であれば、enable = 0 を出力する。done は 0 である。次は状態 st-2 に戻る。
 zero = 1 であれば、 $q_3+q_2+q_1+q_0 = 0$ であるので、もう数えるべき1はない。従って、シフトレジスタもアップカウンタも変化しないよう、 $S_1 = 0$, $S_0 = 0$, reset = 0, enable = 0 を出力する。done も 0 である。次は状態 st-3 に行く。

状態 st-3: 入力に関係なく次の出力をし、状態 st-0 に戻る。
 入力の中の1の個数を調べ終わったので、done = 1 を出力し、他の出力は全て0にする。

このような動作をする順序回路の状態遷移図は下図のようになる。ここで、*/ は入力に依存しないことを、/0-0 は全ての出力を0にすることを、/**** は done 以外の全ての出力が don't care であることを示し、書かれていない出力はその前の時刻の値を取るものとする。



(v) =====

4つの状態があるので、2つの状態変数 x_1, x_2 を導入し、下記の表の左端にあるような状態割当を行うと、状態遷移表は下の表ようになる。ただし、この表において、入力値は start, zero, q_0 の順である。

現状態:		次状態: x_1', x_2'											
x_1, x_2		start, zero, q_0											
		start = 0						start = 1					
		0, 0, 0	0, 0, 1	0, 1, 1	0, 1, 0	1, 1, 0	1, 1, 1	1, 0, 1	1, 0, 0				
st-0	0, 0	0, 0	0, 0	0, 0	0, 0	st-1	0, 1	st-1	0, 1	st-1	0, 1	st-1	0, 1
st-1	0, 1	*, *	*, *	*, *	*, *	st-2	1, 0	st-2	1, 0	st-2	1, 0	st-2	1, 0
st-2	1, 0	*, *	*, *	*, *	*, *	st-3	1, 1	st-3	1, 1	st-2	1, 0	st-2	1, 0
st-3	1, 1	*, *	*, *	*, *	*, *	st-0	0, 0	st-3	0, 0	st-3	0, 0	st-3	0, 0

出力表は下記のようになる。ここでは、入力 start に関する場合分けを省いているが、表は start = 1 の場合である。start = 0 の場合、現状態が st-0 では、出力は下の表と同じであり、st-0 以外では、don't care となる。

現状態:		出力: done, S_1, S_0 , reset, enable																			
x_1, x_2		入力	zero=0, $q_0=0$				zero=0, $q_0=1$				zero=1, $q_0=1$				zero=1, $q_0=0$						
			st-0	0, 0	0	*	*	*	*	0	*	*	*	*	0	*	*	*	*	0	*
st-1	0, 1	0	0	1	1	*	0	0	1	1	*	0	0	1	1	*	0	0	1	1	*
st-2	1, 0	0	1	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0
st-3	1, 1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0

(vi) =====

x_1' のカルノー図は下記のようなになる.

x_1'						start				
		zero								
		q_0				q_0				
		0, 0, 0	0, 0, 1	0, 1, 1	0, 1, 0	1, 1, 0	1, 1, 1	1, 0, 1	1, 0, 0	
x_1	x_2	0, 0								
		0, 1	*	*	*	*	1	1	1	1
	1, 1	*	*	*	*					
		1, 0	*	*	*	*	1	1	1	1

従って, 次式を得る.

$$x_1' = \bar{x}_1 \cdot x_2 + x_1 \cdot \bar{x}_2$$

x_2' のカルノー図は下記のようなになる.

x_2'						start			
		zero							
		q_0				q_0			
		0, 0, 0	0, 0, 1	0, 1, 1	0, 1, 0	1, 1, 0	1, 1, 1	1, 0, 1	1, 0, 0
x_1	x_2	0, 0				1	1	1	1
		0, 1	*	*	*	*			
	1, 1	*	*	*	*				
		1, 0	*	*	*	*	1	1	

従って, 次式を得る.

$$x_2' = \bar{x}_1 \cdot \bar{x}_2 \cdot \text{start} + x_1 \cdot \bar{x}_2 \cdot \text{zero}$$

出力 done は, 上の出力表から分かるように, 状態 st-3, すなわち $(x_1, x_2) = (1, 1)$ の場合に 1 にすればよいから, 次式で書けることが分かる.

$$\text{done} = x_1 \cdot x_2$$

S_1 および S_0 は、下図のようなカルノー図となる。

S_1		zero			
		q_0			
x_1	x_2	0, 0	0, 1	1, 1	1, 0
		0, 0	*	*	*
	0, 1				
	1, 1				
	1, 0	1	1		

S_0		zero			
		q_0			
	0, 0	0, 1	1, 1	1, 0	
	0, 0	*	*	*	
	0, 1	1	1	1	
	1, 1				
	1, 0	1	1		

従って、次式を得る。

$$S_1 = \overline{x_2} \cdot \overline{zero}$$

$$S_0 = \overline{x_1} + \overline{x_2} \cdot \overline{zero}$$

reset および enable は、下図のようなカルノー図となる。

reset		zero			
		q_0			
x_1	x_2	0, 0	0, 1	1, 1	1, 0
		0, 0	*	*	*
	0, 1	1	1	1	
	1, 1				
	1, 0				

enable		zero			
		q_0			
	0, 0	0, 1	1, 1	1, 0	
	0, 0	*	*	*	
	0, 1	*	*	*	
	1, 1				
	1, 0		1		

従って、次式を得る。

$$reset = \overline{x_1}$$

$$enable = \overline{x_2} \cdot q_0 \cdot \overline{zero}$$

(vii) =====

以上の式において共通項の共有化を図ると、次式を得る.

$$\begin{aligned}
 x_1' &= \bar{x}_1 \cdot x_2 + x_1 \cdot \bar{x}_2 = \bar{x}_1 \cdot x_2 + C1, & C1 &= x_1 \cdot \bar{x}_2 \\
 x_2' &= \bar{x}_1 \cdot \bar{x}_2 \cdot \text{start} + x_1 \cdot \bar{x}_2 \cdot \text{zero} = \bar{x}_1 \cdot \bar{x}_2 \cdot \text{start} + C1 \cdot \text{zero} \\
 S_1 &= \bar{x}_2 \cdot \overline{\text{zero}}, & S_0 &= \bar{x}_1 + \bar{x}_2 \cdot \overline{\text{zero}} = \bar{x}_1 + S_1 \\
 \text{reset} &= \bar{x}_1, & \text{enable} &= \bar{x}_2 \cdot q_0 \cdot \overline{\text{zero}} = q_0 \cdot S_1 \\
 \text{done} &= x_1 \cdot x_2
 \end{aligned}$$

これらの式より、下記のような制御回路が合成できる。ここで、制御回路の出力は、状態変数の値 x_1, x_2 だけでなく、入力(ステータス信号)の値にも依存しているから、この制御回路はミラー型の順序回路であることが分かる。

