講 座

開閉回路의 論理設計(Ⅳ)

高 明 三*

— 차 례 —

5. 組合開閉回路의 設計 (Ⅱ)

前回까지 기술한 論理開閉函數, 開閉式의 表現形式 및 簡單化技法을 encoder 設計과정에서 어떻게 적용할 것인가를 설명한다.

5.1 encoder 設計例

지금 純2進數(8421 code)로 5421 code化된 2進數 10進數(a binary coded decimal number)로 變換하기 위한 encoder를 설계하기로 하자. 이 경우의 眞値表는 표 5.1과 같으며, 6個의 don't care項이 나타난다.

표 5.1

A	B	C	D	E	F	G	H
8	4	2	1	5	4	2	1
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0				
1	0	1	1				
1	1	0	0	don't care項			
1	1	0	1				
1	1	1	0				
1	1	1	1				

上記한 表로 부터 各出力條件式을 구하면

$$Y_E=\bar{A}B\bar{C}D+\bar{A}BC\bar{D}+\bar{A}BCD+A\bar{B}\bar{C}\bar{D}+A\bar{B}\bar{C}D$$

$$Y_F=\bar{A}B\bar{C}\bar{D}+A\bar{B}\bar{C}D$$

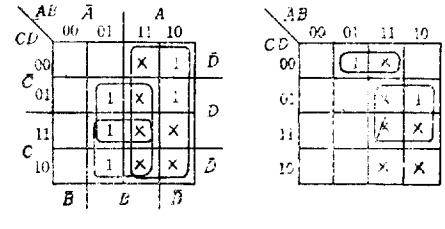
$$Y_G=\bar{A}\bar{B}C\bar{D}+\bar{A}\bar{B}CD+\bar{A}BCD+A\bar{B}\bar{C}\bar{D}$$

$$Y_H=\bar{A}\bar{B}\bar{C}D+\bar{A}\bar{B}CD+\bar{A}BC\bar{D}+A\bar{B}\bar{C}\bar{D}$$

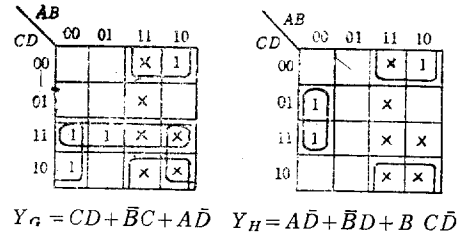
한편 Don't care 조건은

$$Y'_E,Y'_F,Y'_G,Y'_H=A\bar{B}C\bar{D}+A\bar{B}CD+AB\bar{C}\bar{D}+AB\bar{C}D+ABC\bar{D}+ABCD$$

Karnaugh 圖를 이용하여 그림 5.1과 같이 各함수를 最小化하면 다음과 같은 最小項을 얻는다.



$Y_E=A+BD+BC \quad Y_F=AD+B\bar{C}\bar{D}$



$Y_G=CD+\bar{B}C+A\bar{D} \quad Y_H=A\bar{D}+\bar{B}D+B\,C\bar{D}$

그림 5.1 Encorder 설계를 위한 Karnaugh 圖

*正會員 : 서울工大副敎授(工博)·當學會編修理事

$Y_E = A + BD + BC$

$Y_F = AD + B\bar{C}\bar{D}$

$Y_G = CD + \bar{B}C + A\bar{D}$

$Y_H = A\bar{D} + \bar{A}\bar{B}D + BC\bar{D}$

이들 네식을 직접 AND/OR 逆論理로 實現한다면 17個 기본소자를 필요로 하므로, 前回에서 유도된 Mc Clusky法을 이용하여 이들 式을 집단적으로 최소화하자.

우선 各項에 내포된 1의 갯수에 따라서 표 5.2와

표 5.2

	10 進項	함 수	2 進展開項
G_0	1	$\vee Y_H$	0 0 0 1
	2	$\vee Y_G$	0 0 1 0
	4	$\vee Y_F$	0 1 0 0
	8	$\vee Y_E Y_G Y_H$	1 0 0 0
G_1	3	$\vee Y_G Y_H$	0 0 1 1
	5	$\vee Y_E$	0 1 0 1
	6	$\vee Y_E Y_H$	0 1 1 0
	9	$\vee Y_E Y_F$	1 0 0 1
	10	$\vee Y_E Y_F Y_G Y_H$	1 0 1 0
	12	$\vee Y_E Y_F Y_G Y_H$	1 1 0 0
G^2	7	$\vee Y_E Y_G$	0 1 1 1
	11	$\vee Y_E Y_F Y_G Y_H$	1 0 1 1
	13	$\vee Y_E Y_F Y_G Y_H$	1 1 0 1
	14	$\vee Y_E Y_F Y_G Y_H$	1 1 1 0
G_3	15	$\vee Y_E Y_F Y_G Y_H$	1 1 1 1

표 5.3

10 進比較項	함 수	第1次壓縮2進項	
1, 3	Y_H	00-1	*a*
2, 3	$\vee Y_G$	001-	
2, 10	$\vee Y_G$	-010	
4, 12	Y_F	-100	*b*
8, 9	$\vee Y_E$	100-	
8, 10	$\vee Y_E Y_G Y_H$	10-0	
8, 12	$\vee Y_E Y_G Y_H$	1-00	
3, 11	$Y_G Y_H$	-011	*c*
3, 7	$\vee Y_G$	0-11	
5, 7	$\vee Y_E$	01-1	
5, 13	$\vee Y_E$	-101	
6, 7	$\vee Y_E$	011-	
6, 14	$Y_E Y_H$	-100	*d*
9, 11	$\vee Y_E Y_F$	10-1	
9, 13	$\vee Y_E Y_F$	1-01	
10, 11	$\vee Y_E Y_F Y_G Y_H$	101-	
10, 14	$\vee Y_E Y_F Y_G Y_H$	1-10	
12, 13	$\vee Y_E Y_F Y_G Y_H$	110-	
12, 14	$\vee Y_E Y_F Y_G Y_H$	11-0	
7, 15	$\vee Y_E Y_G$	-111	
11, 15	$\vee Y_E Y_F Y_G Y_H$	1-11	
13, 15	$\vee Y_E Y_F Y_G Y_H$	11-1	
14, 15	$\vee Y_E Y_F Y_G Y_H$	111-	

표 5.4

10進比較項	함 수	第2次壓縮2進項	
2, 3/10, 11	Y_G	-01-	*e*
2, 10/13, 11	Y_G	-01-	
8, 9/10, 11	$\vee Y_E$	10--	
8, 9/12, 13	$\vee Y_E$	1-0-	
8, 10/ 9, 11	$\vee Y_E$	10--	
8, 10/12, 14	$Y_E Y_G Y_H$	1--0	*f*
8, 12/ 9, 13	$\vee Y_E$	1-0-	
8, 12/10, 14	$\vee Y_E Y_G Y_H$	1--0	
3, 11/17, 15	Y_G	--11	*g*
3, 7/11, 15	Y_G	--11	
5, 7/13, 15	Y_E	-1-1	*h*
5, 13/ 7, 15	Y_E	-11-	*i*
6, 14/ 7, 15	Y_E	-11-	
9, 11/13, 15	$Y_E Y_F$	1--1	*j*
9, 13/11, 15	$Y_E Y_F$	1--1	
10, 11/14, 15	$Y_E Y_F Y_G Y_H$	1-1-	*k*
10, 14/11, 15	$Y_E Y_F Y_G Y_H$	1-1-	
12, 13/14, 15	$Y_E Y_F Y_G Y_H$	11--	*l*
11, 14/13, 15	$Y_E Y_F Y_G Y_H$	11--	

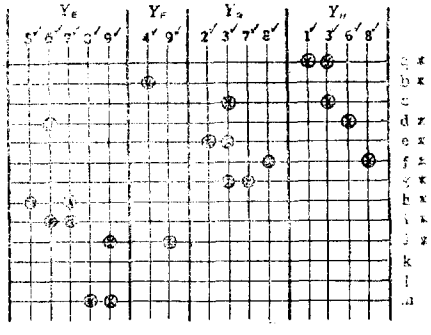
표 5.5

10 進 比 較 項	함 수	第3次壓縮2進項	
8, 9, 10, 11/12, 13, 14, 15	Y_E	1---	*m*
8, 9, 10, 11/12, 14, 13, 15	Y_E	1---	
8, 9, 12, 13/10, 14, 11, 15	Y_E	1---	
8, 9, 12, 13/10, 11, 14, 15	Y_E	1---	
8, 10, 9, 11/12, 13, 14, 15	Y_E	1---	
8, 10, 9, 11/12, 14, 13, 15	Y_E	1---	
8, 10, 12, 14/9, 11, 13, 15	Y_E	1---	
8, 10, 12, 14/9, 13, 11, 15	Y_E	1---	
8, 12, 9, 13/10, 11, 14, 15	Y_E	1---	
8, 12, 9, 13/10, 14, 11, 15	Y_E	1---	
8, 12, 10, 14/9, 11, 13, 15	Y_E	1---	
8, 12, 10, 14/9, 13, 11, 15	Y_E	1---	

같이 組別로 분류하면 G_0, G_1, G_2, G_3가 된다. $A\bar{B}\bar{C}\bar{D}$項은 1000으로 주어지므로 한개의 '1'을 내포하며 出力함수 $Y_EY_GY_H$에 나타난다. 그러므로 이 項은 G_0組에 속하게 된다. 표 5.3은 표 5.2의 각 組의 展開項을 비교하여 第1次壓縮項을 나타낸 것이다. 즉 G_0組에 속하는 項 Y_H 0001과 G_1組의 第一項인 Y_GY_H 0011의 비교로 공통 함수 Y_0가 내포되었고, 한개의 변수만 틀림을 알 수 있다. 즉 第1次壓縮項 Y_D 00−1을 얻게 된다. 이 과정에서 변수 C가 소거되었고 '_'로 나타내며 共通函數 Y_H는 표 5.2에서 '∨'인 기호로 표시한다. 이런 과정을 반복한 결과가 곧 표 5.4 및 표5.2이다. '∨'인 표시를 하지 않은 것이 主項이며 小文字로 나타낸다. 표

다음은 主項圖로 부터 最小項을 구하는 문제가 되며 표 5.6로부터 主項 a, b, d, e, f, g, h 및 j는 必須項임을 알 수 있다.

표 5.6 主項圖



出力함수 Y_E, Y_F, Y_G 및 Y_H에 관한 積合形式은 다음과 같다.

$$Y_E = d+f+h+j = BCD + A\bar{D} + BD + AD$$

$$Y_F = b+j = B\bar{C}\bar{D} + A\bar{D}$$

$$Y_G = e+f+g = \bar{B}C + A\bar{D} + C\bar{D}$$

$$Y_H = a+d+f = \bar{A}BD + BC\bar{D} + A\bar{D}$$

이들 式을 직접 AND/OR 逆論理表示로 實現한다면 16個가 소요됨을 알 수 있다. 앞에서 얻는 결과보다 한 개 더 절약할 수 있게 되었다. 일반적으로 論理回路의 최종선택은 論理시스템의 사용형식, fan-in 因子 시스템에서의 論理信號의 可用性 등 여러 因子들의 영향을 받는다.

5.2 반복설계와 부분시스템仕 方

組合論理設計에 있어서 간단한 回路인 경우에는 眞値表에 의한 回路의 서술이 편리하지만 relay 論理를 적용시키는 복잡한 回路에 있어서는 眞値表法이 오히려 문제를 더 복잡하게 만드는 경우가 많다.

지금 어떤 論理시스템을 여러개의 部分시스템으로 分解할 수 있다고 하면 이들 개개의 部分시스템에 대한 設計를 完成후 이들 部分시스템을 그림과 같이 종속 연결할 수 있게 된다. 그러므로 完全시스템보다 오히려 部分시스템의 設計와 이들의 入出力에서의 仕方(specification)을 어떻게 정하는가 하는 것이 더 큰 문제가 된다.

odd-parity check digit를 내포하는 five fit 並列 $\frac{2}{y}$ 進數에서의 오차의 발생을 찾아내는 회로 소위 parit check回路의 설계를 위에서 이야기한 부분시스템 仕方 관점에서 진행시키기로 하자. 표 5.7은 parity check

표 5.7 Parith Checy 回路 眞値表

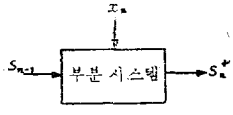
A	B	C	D	E	Y
0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	1	1
1	1	0	0	1	1
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	0	1	1
0	1	0	1	0	1
0	1	0	1	1	0
0	1	1	0	0	1
0	1	1	0	1	0
0	1	1	1	0	0
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	0	1	1
1	0	0	1	0	1
1	0	0	1	1	0
1	0	1	0	1	1
1	0	1	0	1	0
1	0	1	0	1	0
1	0	1	1	1	1
1	1	0	0	0	1
1	1	0	0	1	0
1	1	0	1	0	0
1	1	0	1	1	1
1	1	1	0	0	0
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	0

回路의 眞値表이며 Y는 다음과 같이 주어진다.

$$Y=A\bar{B}\bar{C}\bar{D}\bar{E}+\bar{A}\bar{B}\bar{C}DE+\bar{A}\bar{B}C\bar{D}E+\bar{A}\bar{B}CD\bar{E}$$
$$+\bar{A}B\bar{C}\bar{D}E+A\bar{B}C\bar{D}\bar{E}+\bar{A}BC\bar{D}\bar{E}+\bar{A}BCDE$$
$$+A\bar{B}\bar{C}\bar{D}D+A\bar{B}\bar{C}\bar{D}E+A\bar{B}\bar{C}D\bar{E}+A\bar{B}CDE$$
$$+AB\bar{C}\bar{D}\bar{E}+AB\bar{C}DE+ABC\bar{D}E+ABCD\bar{E}$$

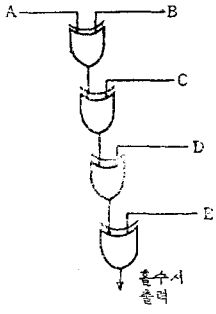
이 回路를 부분시스템으로 分解시키기 위하여 우선 필요한 상태변수 및 外部入力의 갯수를 결정하여야 한다. 每부분시스템當 單一 bit 혹은 여러개의 bits를 外部入力으로 선정할 수 있으나 여기서는 單一빗트 外部入力을 취한다. 왜냐하면 parity check 回路에서는 그 進數에서 1의 개수의 짝수 혹은 홀수의 差만이 문제이기 때문에 單一빗트 상태변수 S만으로 充分하다.

그림 5.2는 外部시스템을 나타내며, 상태변수 S의 값은 홀짝으로 나타내고, 出力상태 S^+는 S_{n-1}과 x_n의 함수로 표시한다. 그림 5.2로부터 만일 入力상태변수가 1의 갯수가 홀수($\bar{S}$)이고 外部入力이 0이면, 상태변수의 값이 변하지 않고, 그 부분시스템의 出力상태는 계속 1의 홀수 즉 $\bar{S}^+$를 나타냄을 알 수 있다. 즉



入力상태 변수 $\bar{S}$	外部入力 x_n 0	x_n 1
$\bar{S}_{n-1}(0)$	$\bar{S}_n^+(0)$	$S_n^+(1)$
$\bar{S}_{n-1}(1)$	$\bar{S}_n^+(1)$	$\bar{S}_n^+(0)$

· 5.2 Parity check回路를 위한 기본부분시스템



$$S_n^+=x_n\bar{S}_{n-1}+\bar{x}_nS_{n-1}$$

이 式은 소위 排他的 OR 論理函數 혹은 半加算器를 의미하며 論理시스템에서의 기본 요소이다.

그러므로 parity check 回路는 그림 5.3과 같은 排他的素子를 종속 결합함으로 이루어진다.

그림 5.3 Parity check 回路

6. 同期式關開回路 및 非同期式開閉回路

論理장치를 설계시 각 부분시스템의 論理回路를 非同期式으로 하고 各部간의 접속도 非同期式으로 구성하면 까다로운 clock 發生器가 不必要하게 되고 多相 clock를 이용시 이러나기 쉬운 clock의 位相差로 인한 誤동작도 염려할 필요가 없다. 또한 어떤 設計方法에 의한 非同期式開閉回路는 論理素子에서의 信號 전송 늦음이 변하여도 誤동작을 이르키지 않는다. 따라서 이러한 回路를 사용하면 電源電壓의 변동, 온도변화 및 經時變化에 대하여 매우 높은 信賴性을 갖는 論理裝置를 구성할 수 있다.

또한 非同時式開閉回路는 同期式開閉回路에는 없는 여러개의 특수한 성질을 갖는다. 이러한 장점을 充分히 살려 非同期式開閉回路를 사용하기 위하여는 이들의 특수한 성질을 알고 있을 필요가 있다. 이절에서는 非同期式開閉回路와 同期式開閉回路의 차이점, 組合回路의 해석 및 설계 非同期式順序回路의 해석 및 설계, 각종의 非同期式順序回路와 그應用을 非同期에 중점을 두어 이야기 한다.

6.1 非同期式回路 및 同期式回路

이미 잘 알려진 바와 같이, 論理回路로 쓰이는 論理素子인 AND, OR, NOT, NAND, NOR 등은 그 入力信號가 변화하여 出力信號가 변화되기까지에는 어느 시간이 소요된다. 이 신호변화의 모양을 그림 6.1(a), (b)로 나타낸다. 그러나 論理回路에서는 이들 변화나분의 中間値는 불필요하다. 그러므로 한 素子의 入力부터 出力까지의 信號傳播의 지연시간은 그 素子으로의 入力値가 入力의 문턱값(threshold value)을 넘고나서 그 素子부터의 出力値가 다음 단계의 素子의 入力의 문턱값을 넘을때까지의 時間이며 이를 그림 6.1 (c), (d)로 나타낸다.

어떤 論理함수로 나타내는 "論理的인 機能"을 論理回路로 실현하려면, 信號의 처리에 있어서 반드시 上述한 바와 같은 시간지연 현상이 수반된다. 이 지연은 回路의 誤동작에 따른 誤出力을 일으키는 원인이 된다. 이러한 문제를 해결하기 위하여는 다음과 같은 여러가지 方法이 사용된다.

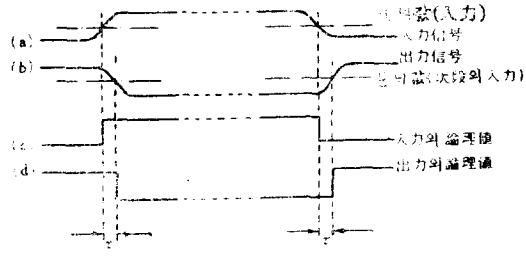


그림 6.1 NOT素子의 入出力信號 및 지연

方法 (1) 素子한개의 최대지연치 τ_M가 정해지고 信號가 回路의 入力端子부터 出力端子로 전달되기까지 통과되는 素子의 數(段數)의 最大値 L_M가 旣知인 경우 回路의 入力 및 出力信號를 周期 $T>\tau_M\times L_M$로 sampling하여 그 값을 유지한다. 예로 表 6.1의 眞値表로부터 가장 간단한 論理關係를 만들면 다음과 같다.

$$Y=\bar{B}\cdot D+\bar{A}B\bar{D}+ABC \qquad (6.1)$$

식(6.1)을 NOT, AND, OR 素子로 실현하면 그림 6.2와 같이 된다.

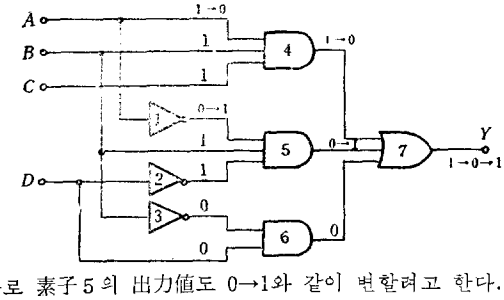
표 6.1 출력 Y인 眞値表

C	D	A / B	0 0	0 1	1 1	1 0
0	0		0	1	0	0
0	1		1	0	0	1
1	1		1	0	1	1
1	0		0	1	1	0

지금 그림 6.2의 回路에서 入力値가 $A=1$, $B=1$, $C=1$, $D=0$로 주어 졌다고 하자. 이때 素子4의 出力値가 1이기 때문에 回路 Y의 出力 Y는 値1을 취한다.

지금 入力 A의 값을 1→0로 변화시키면 素子4의 出力値는 1→0로 변하며, 素子의 出力値의 0→1의 변

그림 6.2 表 6.1을 만족하는 論理回路



화로 素子5의 出力値도 0→1와 같이 변할려고 한다. 그 결과 素子4의 지연 τ_4가 素子1의 지연과 素子5의 지연의合, $\tau_1+\tau_5$보다 적으면 素子7의 出力値는 1→0→1와 같이 변한다. 즉 이 回路는 表 6.1의 眞値表에 나타나지 않은 出力0(誤出力)을 발생한다. 여기서 그림 6.2 回路의 入力端子 및 出力端子에 그림 6.3(a)의 時間表와 같은 기능을 가지며 D-Flip Flop (D-FF)를 접속하여 그림 6.3(b)와 같은 回路를 고려한다.

그림 6.3(b)인 回路에서 (A, B, C, D)의 값이 그림 6.2인 경우와 같이 (110)→(0110)로 변하는 경우를 생각하면 A의 값이 변한 시각직후의 clock 信號의 발생으로 FF-1의 出力値는 1→0와 같이 변한다. 이때 素子7의 出力値는 계속 1이기 때문에 그다음의 clock 信號의 발생까지의 기간내에 FF-5의 出力은 1을 계속 유지한다. 이 回路에서도 素子4의 지연 τ_4가 素子5의 지연τ_5보다 적을 경우에는 素子7의 出力値는 전과 같이 1→0→1와 같이 변한다. 그러나 clock의 周期 τ와 素子 4, 5, 6, 7 중의 최대 지연 τ_M의 관계가 $\mathbf{T}>2\tau_M$인 관계를 만족하면 素子7의 誤出力0은 후속되는

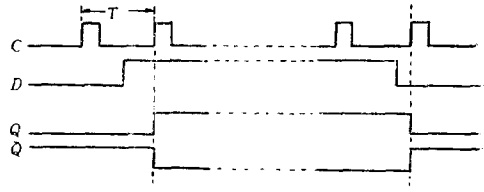


그림 6.3 (a)

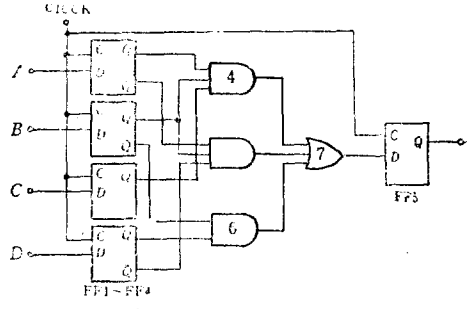


그림 6.3 (b)

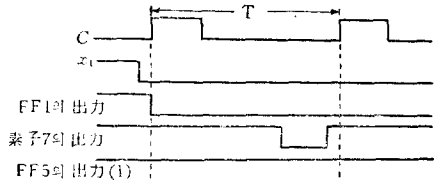


그림 6.3 (c) D-FF

clock 信號의 발생 전에 지워진다. 따라서 出力 Y는 다음 信號가 도래하여도 값 "1"이 된다. 이 모양을 圖示한 것이 그림 6.3(c)이다. 이 例에서는 D-FF를 사용하였으나 JK-FF, clock RS-FF 등도 이용된다.

方法 (2) 모든 素子의 지연시간이 동일한 경우 信號가 回路의 入力端子로 부터 出力端子까지 전달하는 동안, 통과하는 素子의 數를 모든 入力端子, 出力端子에 관하여 동일하게 하고 信號의 傳達속도의 차로 인하여 발생하는 誤出力을 없앤다.

이 方法은 그림 6.2의 回路에 적용하면 그림 6.4와 같이 된다. 그림 6.4의 回路에서 그림 6.2에서 기술한 OR素子에 대한 入力變化 동시에 일어난다. 따라서 出力 Y의 값은 1을 그대로 유지한다.

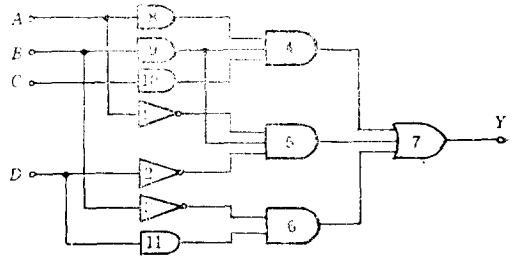


그림 6.4 동일지연 소자로 구성된 回路

이 方法은 한 종류의 素子(예를 들면 NAND, NOR)만을 사용하며 回路를 구성하는 경우에 많이 사용된다 그러나 일반적으로 素子의 지연은 어느 범위에서 不規則性을 띠게 되며 한개의 素子일지라도 上昇시와 下降에 지연이 달라지는 경우가 많기 때문에 이 방법은 信賴度가 높은 回路에서는 구성할 수 없다.

方法 (3) 素子 한개의 지연 최대치 τ_M, 최소치 τ_m, 및 回路의 段書의 最大値 L_M, 最小値 L_m가 기지인 경우 $D>\tau_M\times(L_M-1)-\tau_m\times(L_m-1)$을 만족하고 성질 (i), (ii)을 갖는 素子를 回路의 出力端子에 부가한다.

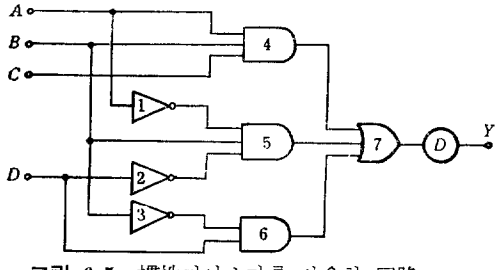


그림 6.5 慣性지연소자를 사용한 回路

(i) 素子의 入力 및 出力端子는 각각 하나씩이며 入力値가 두번 변하였을때 (0→1→0 혹은 1→0→1 그 변화의 간격이 一定時間보다 적을 때는 出力値는 不變이다.

(ii) 변화간격이 D보다 큰 경우, 入力變化는 D만큼 늦게 出力에 나타난다.

그림 6.2인 回路에 에 지연소자를 부가시킨 경우를 생각해 보자. 그림 6.5는 이 경우를 나타낸다.

그림 6.5의 回路에서도 入力이 (1110)→(0110)로 변화하는 경우에는 그림 6.2의 경우와 같이 素子 4, 5, 의 出力値가 각각 1→0, 0→1와 같이 변한다. 그러나 이 두가지 信號變化의 시간 간격은 素子 4 의 지연시간 τ_4와 素子 1, 및 5 의 지연시간의 合 $\cdot\ \tau_1+\tau_5$와의 差이다. 따라서 素子 4 의 지연시간을 τ_m라 하고, 素子 1 및 5 의 지연을 τ_M라 가정하여도 變化의 時間間隔은 $2\tau_M-\tau_m$이다. 즉 誤出力의 pulse幅은 $2\tau_M-\tau_m$보다 커지지 않는다. 그러므로 지연소자의 성질 (i)로부터 出力 Y의 값은 變하지 않는다' 실제 회로에서는 관성지연 素子로서 콘덴서가 잘 이용되고 있다.

예를 들면 그림 6.6(a)와 같은 DTL의 出力端子에 콘덴서 C를 접속하고, 出力値가 0→1와 같이 變化시에는 그림 6.6(b)와 같은 等價回路를 얻게 된다.

그러므로 그림 6.7과 같이 콘덴서가 없는 경우의 出力値는 0→1→0와 같이 變化하는 경우에도 變化간격이 어느 一定値를 초과하지 않는 이상은 出力의 論理値는 不變이다.

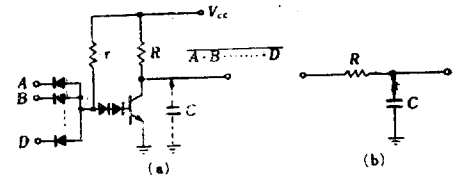


그림 6.6 콘덴서에 의한 관성지연

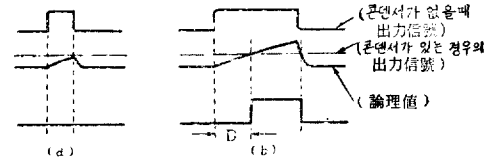


그림 6.7 그림 6.6 回路의 出力波形과 論理値

方法 (4) 論理素子間의 接續方法에 어떤 기준을 설정하여 일시적인 誤出力을 일으키지 않게 回路를 구성한다. 이 方法에서는 나중에 상세히 기술하기로 한다. 例로서 表 6.1의 機能을 이 方法으로 실현한 回路는 그림 6.8과 같다. 이 回路는 그림 6.2의 回路에 두개의 AND 素子, 8, 9를 더한 것이다. 이들 두개의 素子의 出力値는 表 6.1의 眞値表에서, 0가 되어 있는 入力의 組合에서는 반드시 0을 취한다. 그러므로 回路의 入力値가 安定되어 있을 때는 이들의 두 素子의 出力値는 回路의 出力에는 전혀 관계가 없다.

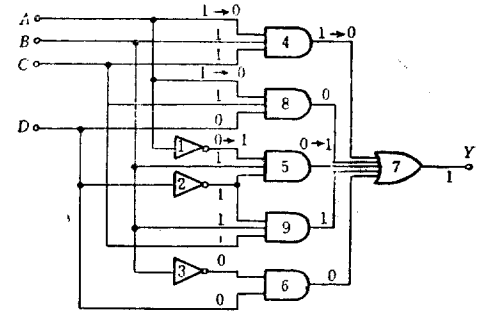


그림 6.8 方法(4)로 구성한 回路

그러나 그림 6.8의 回路의 入力을 그림 6.2인 경우와 같이 (1110)→(0110)로 변화시켰을 때, 素子 1, 2, 3, 4, 5, 6의 동작은 回路 6.2와 마찬가지지만 素子 9 의 出力値는 1을 그대로 유지한다. 그 결과 素子 4, 5의 出力値의 變化 時刻에는 차가 있어도 回路의 出力値는 1을 그대로 유지한다.

素子 8 은 回路의 入力이 (1111)→(1011)로 변하였을 때 上記한 素子 9 와 동일한 作用을 하게 된다.

回路의 入力은 한번에 하나만 변한다고 가정하면 그림 6,8의 回路는 어떤 入力變化에서도 一時的인 誤出力을 일으키지 않음을 알 수 있다.

한개의 論理機能을 回路로 실현하는 方法은 이외에도 여러가지로 생각할 수 있으나, 이들은 결국 지금까지 기술한 네가지 方法들의 組合이라고 생각할 수 있다.

위에서 기술한 方法중〔1〕은 入力信號와 出力信號를 同期시키기 위한 同期信號(clock)를 필요로 한다. 그러므로 이러한 方法으로 구성되는 回路를 同期式回路라 한다.

한편 〔3〕 및 〔4〕의 方法에서는 回路의 入出力信號를 동기시킬 필요는 없으며 信號의 대기 현상은 어느 곳에서든지 일어나지 않는다. 이러한 回路를 非同期式回路라 한다. 또한 方法〔2〕는 方法〔3〕에서 $\tau_M=\tau_m$, $L_M=L_m$로 한 특별한 경우에 해당한다.

이상의 例로부터, 同期回路式回路의 出力値는 clock 周期의 정수배의 時間間隔만큼만 변하지만, 非同期式回路인 경우에의 出力値는 變化하는 시각은 정해져 있지 않다. 이것이 곧 非同期式回路의 특징의 하나이다.

또한 方法〔4〕에서 구성된 非同期式回路는 素子의 시간지연의 크기에는 관계없이 정확한 出力을 낸다. 그러므로 지연시간의 不規則的인 分布가 큰 소자를 사용하는 경우에도 信賴性이 높은 回路를 구성할 수 있고, 이 方法으로 구성된 回路는 주위 온도의 변화, 전원변동 등으로 인한 素子 지연의 변화에 대해서 매우 안전한 동작을 한다.

7. 組合回路 및 順序回路

7.1 入力系列

論理回路에서는 回路의 出力이 현재 가해지고 있는 入力의 組合에 의해서만 결정되는 것과 과거부터 현재까지 가해진 入力의 이력에 의하여 出力이 결정되는 것의 두가지가 있다.

예를 들면 그림 7.1(a)와 같은 回路는 (A, B)가 (0, 1), 인때 그 以前에 가해진 入力에는 관계없이 出力은 1이 되고, (0,1), (1,0)인 때는 값 0을 취하게 된다 그러나 (b)圖는 (0,0)인때 出力値가 0, (1.0), (1,1)까지의 入力의 組合의 이력을 入力系列이라 한다.

앞절에서 기술된 同期式 回路에서는 入力信號는 clock 周期로 sampling되기 때문에 每 clock time마다 의의를 갖는다. 그러나 非同期回路에서는 入力信號는 變化時點에만 그 의의를 가지므로 동일한 入力信號일지라도 同期式과 非同期式에서는 각각 서로 다른 入力系列이 된다.

그림 7.2와 같은 入力信號는 同期式回路에서는 1100 10와 같은 入力系列이 되지만, 非同期式回路인때 出力値는 1을 취하나, (0,1)인때는 그 이전의 入力이 (0,0) 및 (1,0)이면 出力値는 0을 취하지만, (1,1)이면 1을 취한다.

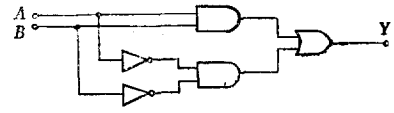


그림 7.1(a)　組合回路

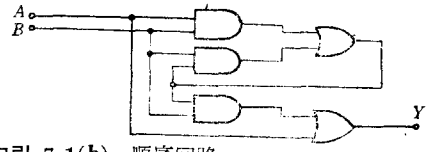


그림 7.1(b)　順序回路

그림 7.1　組合回路 및 順序回路

그림 7.1(a)와 같이 出力値가 현재의 入力組合에 의해서 결정되는 회로를 組合回路라 하며 현재의 入力의 組合만으로는 결정되지 않는 回路를 順序回路라 한다. 또한 그림(b)의 경우 즉, (0,1)→(0,1), (1,1)→(0,1)와 같이 과거로 부터 현재에서는 단순히 1010의 入力系列로 볼 수 있다.

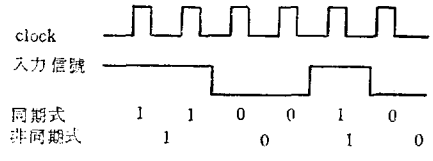


그림 7.2　同期式回路 및 非同期式回路의 入力系

7.2 列序序回路의 구성

入力 $A, B, C\cdots\cdots, N$을 갖는 論理回路의 시각 t에서 入力系列은 正의 數 $\varepsilon_1, \varepsilon_2, \cdots\cdots, \varepsilon_p, \cdots$: $(\varepsilon_1<\varepsilon_2<<\varepsilon_p<\cdots\cdots)$을 사용하여,

$$\cdots\cdots\cdots\cdots\cdots\cdots,$$
$$\cdots\cdots\cdots\cdots\cdots\cdots,$$
$$(A(t-\varepsilon_p),\ B(t-\varepsilon_p), \cdots\cdots, N(t-\varepsilon_p)),$$
$$\cdots\cdots\cdots\cdots\cdots\cdots,$$
$$(A(t-\varepsilon_2),\ B(t-\varepsilon_2),, \cdots\cdots N(t-\varepsilon_2)),$$
$$(A(t-\varepsilon_1),\ B(t-\varepsilon_1), \cdots\cdots, N(t-\varepsilon_1)),$$
$$(A(t), B(t), \cdots\cdots\cdots\cdots, N(t))$$

와 같이 표시할 수 있다.

난 $t-\varepsilon_1,\ t-\varepsilon_2, \cdots\cdots, t-\varepsilon_p, \cdots\cdots$. 은 同期式 回路인 경우에는 각각의 clock time을 의미하며, 非同期式回路에서는 入力 $A, B, C, \cdots\cdots, N$ 중에서 적어도 한개의 値가 變化한 時刻을 나타낸다.

두値의 入力의 組合$(A, B, \cdots\cdots, N)$을 vector라 생각하여 $\cdots\cdots, \underline{X}(t+\varepsilon_p), \cdots\cdots, \underline{X}(t-\varepsilon_2), \cdots\cdots \underline{X}(t-\varepsilon_1), X(t)$

와 같은 vector 형식으로 표시한다.

이와 같이 하면 順序回路의 시각 t에서의 出力 $Y(t)$는 組合回路와 마찬가지로 論理함수를 사용하여 다음과 같이 나타낼 수가 있다.

$$Y(t)=H(\underline{A}(t),\underline{X}(t-\varepsilon_1),\cdots,\underline{X}(t-\varepsilon_2),\underline{X}(t-\varepsilon_p),\cdots) \quad (7.1)$$

또한 回路의 出力端子가 l 本인 경우에는 각각의 端子의 出力에 添字를 첨가하여

$$Y_i(t)=H_i(\underline{X}(t),\underline{X}(t-\varepsilon_1),\underline{X}(t-\varepsilon_2),\cdots),\underline{X}(t-\varepsilon_p)\cdots,) \quad i=1,2,\cdots\cdots,l \quad (7.2)$$

와 같이 나타낼 수 있다.

그림 7.1(b)의 回路를 예로 취하면 入力系列의 길이는 최고 2이므로, 回路의 出力$Y(t)$는 $\underline{A}(t),A(t-\varepsilon_1)$의 組合을 변수로 하는 眞値表로 나타낼 수 있다. 단 (b)圖의 回路는 非同期式이기 때문에

$$(0,0)\rightarrow(0,0),\ (1,0)\rightarrow(1,0)$$

와 같은 系列은 존재하지 않는다. 따라서 眞値表 7.1에서는 이러한 入力에 해당하는 곳을 X인 표시로 나타낸다. X인 곳은 0 혹은 1의 어느 값도 취할 수 있다고 하면 表 7.1로부터

$$Y(t)=A(t)+B(t)\cdot C(t-\varepsilon_1) \quad (7.3)$$

그런데 실제의 回路는 현재의 回路狀態와 現在의 入力만으로 현재의 出力이 결정된다. 즉 식(7.2)의 $\underline{A}(t-\varepsilon_1)$, $\underline{B}(t-\varepsilon_2)$, …… 는 어떤 형태로서든지 回路에 기억될 것이므로 이러한 回路狀態를 $S(t)$로 나타내면 식(7.2)는

$$Y_i(t)=G_i(\underline{A}(t),S(t) \quad (7.4)$$

$(i=1,2,\cdots\cdots,l)$

또한 $S(t)$는 다음과 같이 된다.

$$S(t)=Q(\underline{A}(t-\varepsilon_1),B(t-\varepsilon_2),\cdots,N(t-\varepsilon_p),\cdots) \quad (7.5)$$

表 7.1 (b)回路出力

$A(t-\varepsilon_1)$ \ $\underline{A}(t)$	00	01	11	10
00	※	0	1	1
01	0	※	1	1
11	0	1	※	1
10	0	0	1	※

단 $S(t)$ 자신도 현재의 回路狀態를 현재의 入力에 따라서 변하며 時間 $\Delta t>0$ 경과 후에는 다음 $S(t+\Delta t)$가 될 것이다. 그러므로

$$S(t+\Delta t)=F(\underline{A}(t),S(t) \quad (7.6)$$

이 Δt는 同期式 回路에서는 clock周期T이며, 非同期式回路에서는 回路의 지연 時間이다.

이 Δt는 同期式回路에서는 clock同期T이고, 非同期式 回路에서는 回路의 지연 時間이다.

回路의 狀態 $S(t)$ 역시 실제 回路에서는 2値로 주어지는 어떤 情報로서 기억된다. 그러므로 $S(t)$ 역시 2値 vector로서 다음과 같이 나타날 수 있다.

$$S(t)=\underline{Z}(t),\ Z_2(t)=(Z_1(t),\cdots\cdots,Z_m(t)) \quad (7.9)$$

식(7.7)을 사용하면 식(7.4) (7.6)은 각각 다음과 같이 쓸 수 있다.

$$Y_i(t)=G_i(A(t),B(t),\cdots\cdots,N(t),Z_1(t),Z_2(t),\cdots Z_m(t)) \quad (7,8)$$

$$i=1,2,\cdots\cdots,l$$

$$Z_j(t+\Delta t_j)=F_j(A(t),B(t),\cdots,N(t),\cdots,Z_1(t),Z_2(t),\cdots\cdots,Z_m(t)) \quad (7.9)$$

$$j=1,2,\cdots\cdots,m$$

이들 식은 現在의 時刻 t에서의 論理函數이다. 그러므로 組合回路로 이 기능을 實現할 수 있다. 順序回路가 組合回路와 구조적으로 다른 點은 식(7.9)와 같은 回路狀態를 기억하는 기능을 갖는 部分回路가 存在한다는 점이다.

$F_j(j=1,2,\cdots,m)$의 機能을 同期式回路로 實現할 경우 또는 非同期式回路라도 6절에서의 方法[3]으로 구성하여 지연소자 D의 지연시간 D_j가 어떤 값보다 큰 경우 Δt_j는 단순히 F_j를 실현한 回路에서의 信號 전파의 지연시간이 된다. 그러므로 이 경우의 順序回路는 그림 7.3(a)와 같은 構造를 갖는 論理回路로 생각할 수가 있다.

그림 7.3(a)의 組合回路 1,2는 각각 $F_j,G_i(r=1,2,\cdots,m,\ i=1,2,\cdots,l)$의 기능을 갖는 理想的인 信號傳播의 지연이 없는 組合回路이며, $\varepsilon_j(j=1,2,\cdots,m)$는 F_j를 實現한 경우에 생기는 信號傳播의 지연을 나타내는 지연요소이다.

그러나 $F_j(j=1,2,\cdots,m)$가 앞절의 方法(3)으로 구성되어도 D_j가 어느 값보다 적은 경우, 및 方法(4)로 구성된 경우에는 반드시 그림 7.3(a)와 같이 나타내지 못한다. 그 理由를 간단히 기술하면 入力 $x_i(i=1,2,$

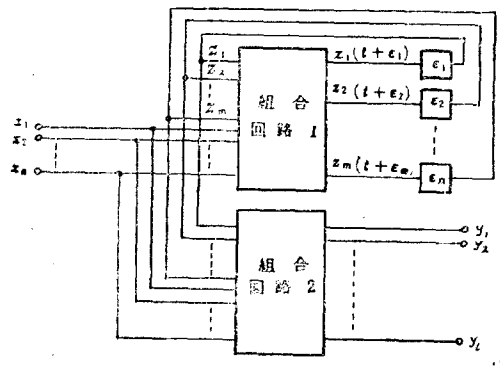


그림 7.3 (a)

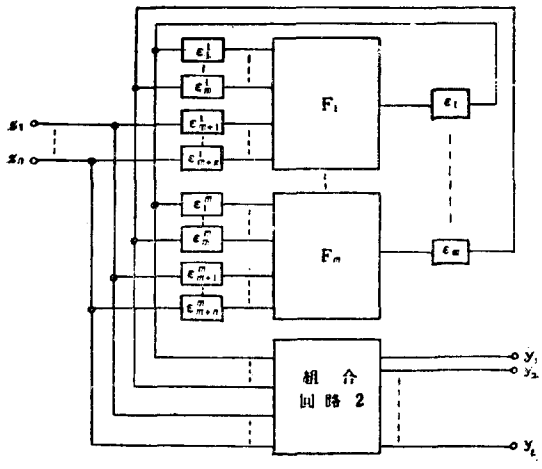


그림 7.3 (b)

그림 7.3 順序回路의 構造

…, m) 혹은 값 $Z_j(j=1,2,\cdots,m)$의 어느 하나라도 변하면 $F_1, F_2, \cdots, F_m$를 실현한 회로의 모든 것에 그 變化를 동시에 미치게 할 수는 없기 때문이다. 이 현상의 설명 및 D_j의 크기와의 관련에 관해서는 非同期式順序回路의 解析, 設計欄에서 상세히 기술하겠으나, 이들의 경우 組合回路 1은 그림 7.3(b)와 같이 각각의 함수에 대응하는 이상적인 組合回路의 入力側에도 지연 요소를 갖는 論理回路로 생각할 수 있다.

G_j의 기능을 실현한 回路에도 지연은 존재하나, 이 지연은 단순히 出力 $y_1, y_2, \cdots, y_l$이 변하는 경우에 그 변화시각을 지연시킬 뿐이므로, 順序回路의 動作으로서는 본질적인 뜻은 갖지 못한다. 단 앞절에서 기술한 바와 같이 일시적인 誤出力을 일으킬 때도 있다. 예로 그림 6.2(b)의 회로를 이용하면 이 回路는 식(7.8), (7.9)에 대응한 다음과 같은 두가지 함수로 표현할 수 있다.

$$y(t)=x_1(t)+x_2)\cdot z(t) \qquad (7.10)$$

$$z(t+\varepsilon)=x_1(t)\cdot x_2(t)+x_2(t)\cdot y(t) \qquad (7.11)$$

이 回路에서는 함수 F_j는 하나만 存在한다. 그러므로 그림 7.3(a)와 같이 회로의 구조를 나타낼 수 있다 이를 그림 7.4와 같이 나타낸다.

지금 同期式順序回路와 非同期式順序回路의 차이점을 다소 언급한다. 식(7.6)과 그림7.3(a)의 구조로 부터 확실한 바와 같이 同期式順序回路에서는 $\varepsilon_1=\varepsilon_2=\cdots=\varepsilon_m=T$이므로, 時刻 $t+\Delta$에서의 回路의 狀態 $S(t+\Delta)$는 식(7.6)에 의하여 唯一하게 결정 된다,

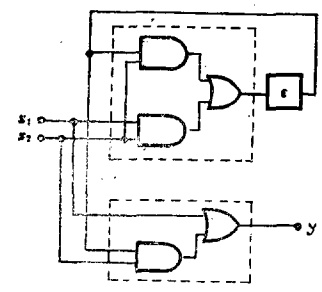


그림 7.4 두 組合回路를 분해한 그림 7.1(b)의 回路

한편 非同期式順序回路에서는 $\varepsilon_j(j=1,2,\cdots,m)$는 잡다한 여러가지 값을 갖는다, 그러므로 $m\geq 2$인 경우에는 그림 7.3(a), (b)로 부터 식(7.6)의 $S(t+\varepsilon)$은 ε의 크기에 따라서 다른 값이 된다. 이 때문에 非同期式順序回路는 同期式順序回路에서 볼 수 없는 특수한 동작을 하게 된다는 것을 알 수 있다.

〔**Quiz 4**〕

(1) 그림 6.3의 回路를 D-FF 대신에 master slave 形의 JK-FF로 구성하라. 또 이 경우의 最高 clock周波數($1/T_{min}$)도 구하라.

(2) 그림 A와 같은 回路의 出力 y를 入力 $x_1(t)$, $x_2(t)$, $x_1(t-\Delta_1)$, $x_2(t-\Delta_2)$, … 로 나타내라.

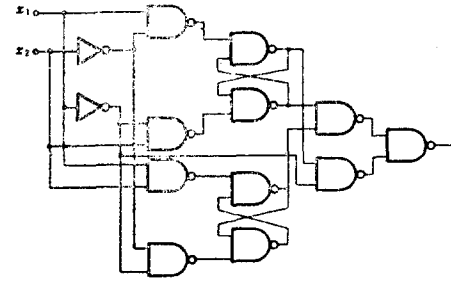


그림 A

(3) 그림 B와 같은 구조의 flip flop를 사용하여, 다음式으로 나타내는 非同期式順序回路를 구성하라.

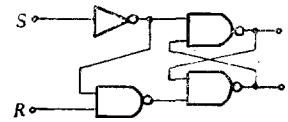


그림 B