
PLA에 기초한 디지털논리스위칭함수 구성

박춘명*

The Construction of the Digital Logic Switching Functions using PLA

Chun-Myoung Park*

요 약

본 논문에서는 PLA를 사용하여 디지털논리스위칭함수를 효과적으로 구성하는 방법을 제안하였다. 제안한 방법은 먼저 포스트 대수를 기반으로 MIN 대수연산과 MAX 대수연산을 제안하였고, 이를 구현하기 위해 T-gate에 대해 논의하였다. 그리고 PLA의 기본 회로인 MIN 배열, MAX 배열과 리터럴에 대해 논의 하였다. PLA를 사용하여 디지털논리스위칭함수를 설계하기 위해 변수분할, 모듈러 구조, 리터럴 생성기, 복호기와 인버터를 제안하였다. 제안한 방법은 좀 더 콤팩트하고 확장성이 용이하다.

ABSTRACT

This paper presents a method of constructing the digital logic switching functions using PLA. First of all, we propose a MIN and MAX algebra arithmetic operation based on the Post algebra. And we discuss the T-gate which is used for realization of the MIN and MAX algebra arithmetic operation. Next, we discuss the MIN array and MAX array which are basic circuit of the PLA, also we discuss the literal property. For the purpose of the design for the digital logic switching functions using PLA, we propose the variable partition, modular structure design, literal generator, decoder and inverter. The proposed method is the more compactable and extensibility.

키워드

PLA, digital logic switching functions, post algebra, variable partition, modular structure

I. 서 론

최근에 디지털논리스위칭함수[1-3]를 PLA, ROM 또는 RAM 등과 같은 규칙적이고 확장성이 용이한 구조를 갖는 소자를 사용하여 구성하는 연구가 활발히 진행 [4-5]중이며 이중 PLA 소자는 모듈러와 용이성의 특징에 바탕을 둔 대표적인 집적된 디지털논리스위칭함수용 소자[6-7]이다. 본 논문에서는 리터럴 생성기를 제안하였으며, 이를 바탕으로 복호기를 구현하여 복잡한 입력단 부분을 간략화함으로써 출력단 부분의 복잡성을

줄였다. 또한, 스위칭함수를 부분스위칭함수로 분할하는 알고리즘을 제안하였으며 이를 바탕으로 부분함수를 간략화하는 방법을 도출하였다. 그 결과 P치[8-9]논리PLA의 전체적인 크기를 감소시킬 수가 있었다. 본 논문의 서술과정은 다음과 같다. 2장에서는 본 논문을 전개하는데 필요한 Post 대수연산에 기초한 MIN과 MAX 연산에 대한 논의를 하였으며 또한 이를 실현하기 위한 P치논리소자로서의 T-gate에 대해 서술하였다. 그리고 3장에서는 2장에서 논의한 MIN과 MAX 연산을 기초로 MIN 배열과 MAX 배열에 대해 논의하였으며 리터럴의

성질에 대해 서술하였다. 4장에서는 변수분할, 모듈러 구조 설계, 리터럴 생성기, 복호기 구성 및 인버터에 대해 논의하였다. 그리고 5장에서는 예를 들어 그 결과를 비교 및 검토하였다. 마지막으로 6장의 결론에서는 본 논문에서 제안한 P치논리PLA에 기초한 디지털논리스위칭함수구현의 특징을 요약하였으며 향후 연구과제에 대해 서술하였다.

II. MIN과 MAX 대수 연산과 T-gate

2-1. MIN과 MAX Post 대수 연산

MIN(·)과 MAX(∨)에 대한 Post 대수 연산을 논의하기 위해 P치논리스위칭함수에 대한 MIN과 MAX의 연산표는 다음 표1과 표2로 정의된다.

표 1. MIN Post 대수 연산표
Table 1. The MIN Post Algebraic Operation table

·	0	1	2	P-2	P-1
0	0	0	0	0	0
1	0	1	1	1	1
2	0	1	2	2	2
⋮				⋮		
P-2	0	1	2	P-2	P-2
P-1	0	1	2	P-2	P-1

표 2. MAX Post 대수 연산표
Table 2. The MAX Post Algebraic Operation table

∨	0	1	2	P-2	P-1
0	0	1	2	P-2	P-1
1	1	1	2	P-2	P-1
2	2	2	2	P-2	P-1
⋮				⋮		
P-2	P-2	P-2	P-2	P-2	P-1
P-1	P-1	P-1	P-2	P-1	P-1

2-2. T-gate

앞에서 논의한 P치 MIN과 MAX Post 대수 연산을 실현하기 위한 논리소자로서 본 논문에서는 T-gate를 사용하며, 이에 대한 블록선도는 다음 그림1과 같고 동작특성은 다음 식(1)로 정의된다.

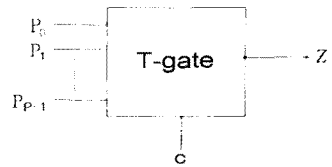


그림 1. T-gate의 블록선도
Fig. 1. The Block diagram of T-gate

$$T(P_0, P_1, P_2, \dots, P_{p-1}, C) = Z, \text{ if } C = K \tag{1}$$

여기서 $Z, C, K \in GF(P) = \{0, 1, 2, \dots, P-1\}$

III. MIN 배열과 MAX 배열을 갖는 PLA

MIN과 MAX Post 대수와 이를 수행하는 T-gate를 사용하여 m개 입력과 n개의 출력을 갖는 P치논리PLA를 구성하면 그림2와 같고 이의 동작특성은 다음 식(2)로 정의된다.

$$F = 0.g_0 \cdot 1.g_1 \cdot 2.g_2 \cdot \dots \cdot (P-2).g_{p-2} \cdot (P-1).g_{p-1} \tag{2}$$

여기서 ·는 MIN 연산자이고 ∨는 MAX 연산자이다.

위 식(2)에서 일반적으로 g_0 항과 최대값인 (P-1)은 생략할 수 있으므로 위 식(2)는 다음 식(3)으로 표현할 수 있다.

$$F = 1.g_1 \cdot 2.g_2 \cdot \dots \cdot (P-2).g_{p-2} \cdot g_{p-1} \tag{3}$$

또한, 위 식(2)에서 각각의 부분함수 $gK(K=1, 2, \dots, P-1)$ 은 다음 식(4)로 정의된다.

$$gK = \bigvee X_1^{S1} \cdot X_1^{S2} \cdot \dots \cdot X_m^{Sm} \tag{4}$$

여기서 $S_K \in \{0, 1, \dots, P-1\} (1 \leq K \leq m)$

한편, 리터럴(literal) X^S 는 다음 식(5)와 같이 정의된다.

$$X^S = \begin{cases} 0 & \text{if } X \notin S \\ P-1 & \text{if } X \in S \end{cases} \tag{5}$$

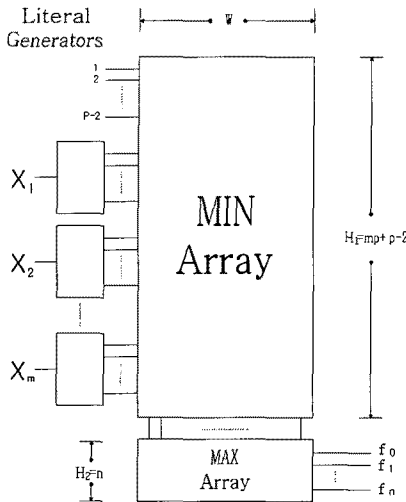


그림 2. MIN 배열과 MAX 배열을 갖는 PLA의 블록선도

Fig. 2. The block diagram of the PLA with MIN array and MAX array.

IV. PLA를 이용한 디지털논리스위칭함수 구성

본 논문에서 제안한 P치논리PLA를 토대로 다치논리 스위칭함수를 실현하기 위해 먼저 변수분할알고리즘을 제안하고 이를 바탕으로 다치논리 스위칭함수를 부분함수로 분할하였다. 그리고 다치논리PLA를 Building block으로 하여 최종 다치논리디지털시스템을 구성하였다. 따라서, 다치논리스위칭함수에 대한 다치논리디지털시스템 설계는 크게 변수분할과 모듈구조설계로 나눌 수 있다.

4-1. 변수분할

본 논문에서 제안한 변수분할알고리즘은 다음과 같다.

[변수분할알고리즘]

- (STEP1) 임의의 주어진 P치논리스위칭함수의 변수 중에서 적절한 2개의 변수를 선택하여 부분함수로 나눈다.
- (STEP2) STEP1에서의 부분함수로부터 구한 gP-1의 함수를 다시 gK(K=1, 2, ..., P-3, P-2)의 부분함수로 분할한다. 이 경우 gR(K+1 ≤ R ≤ P-1)은 무관항(don't care)으로 처리된다.

(STEP3) 입력과 출력이 동일한 gk항을 찾아 공통 출력으로 한다.

(STEP4) 2변수의 부분함수로 PLA를 구성한다.

(STEP5) 각각의 부분함수로 구성된 PLA를 종속접속(cascade connection)하여 최종 디지털논리스위칭함수를 구현한다.

4-2. 리터럴 생성기

리터럴 생성기를 논의하기 위해 예를 들어 3치논리에 대한 리터럴 생성기(literal generator)를 블록선도로 나타내면 그림3과 같다.

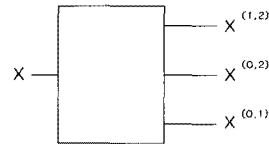


그림 3. 3치논리 리터럴 생성기의 블록선도

Fig. 3. The block diagram of the 3-valued logic literal generator.

4-3. 복호기

복호기의 구성에 대해 논의하기 위해 예를 들어 3치논리에 대한 복호기(Decoder)를 구성하면 다 그림4와 같다.

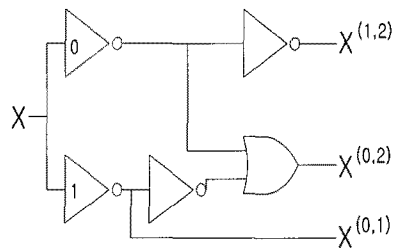


그림 4. 3치논리 복호기

Fig. 4. The 3-valued logic Decoder.

4-4. 인버터

인버터를 논의하기 위해 예를 들어 3치논리에 대한 각종 트래시홀드(threshold) 값을 갖는 인버터(Invertor)의 심볼과 그의 입출력에 대한 동작 특성을 나타내면 다음 표3과 같다.

표 3. 여러 가지 트래시홀드 값을 갖는 인버터.
Table 3. The Inverter with several threshold value.

입 력	출 력		
0	2	2	2
1	0	2	*
2	0	0	0

여기서 *는 don't care 항을 의미함.

V. 적용 예 및 비교·검토

본 장에서는 4장에서 제안한 PLA를 사용하여 임의의 디지털논리스위칭함수를 회로실현하여 디지털논리시스템을 구성하는 과정을 3치논리PLA를 사용하여 3치는 리디지털시스템을 구성하는 예를 들어 설명한다.

예) 다음 표4와 같은 5변수 3치논리에 대한 스위칭함수의 진리치표로부터 2장부터 4장까지의 내용을 적용하여 3치논리디지털논리시스템을 구성하는 과정은 다음과 같다.

표 4. 5변수 3치논리 스위칭함수 $F(X_1, X_2, X_3, X_4, X_5)$ 의 진리치표

Table 4. The truth table of the 5-variable 3-valued logic switching function $F(X_1, X_2, X_3, X_4, X_5)$

$X_2 X_1$	$X_3, 012\ 012\ 012\ 012\ 012\ 012\ 012\ 012\ 012\ 012$ $X_4, 000\ 111\ 222\ 000\ 111\ 222\ 000\ 111\ 222$ $X_5, 000\ 000\ 000\ 111\ 111\ 111\ 222\ 222\ 222$
00	100 001 110 011 110 201 100 001 110
01	122 221 012 000 000 200 122 221 012
02	200 002 120 022 220 002 200 002 120
10	200 002 120 022 220 002 200 002 120
11	100 001 110 011 110 201 100 001 110
12	122 221 012 000 000 200 122 221 012
20	122 221 012 000 000 200 122 221 012
21	100 001 110 011 110 201 100 001 112
22	200 002 120 022 220 002 200 002 120

이제 위 표4의 내용을 각각 2변수 $[X_1, X_2]$ 로 분할하면 다음 표5와 같다.

표 5. 5변수 3치논리 스위칭함수 $F(X_1, X_2, X_3, X_4, X_5)$ 를 변수 X_1 과 X_2 의 변수로 분할한 부분함수 P_A, P_B, P_C, P_D, P_E 의 진리치표.

Table 5. The truth table for the partial switching function P_A, P_B, P_C, P_D, P_E that are partitioned by variable X_1 and X_2 of 5-variable 3-valued switching function $F(X_1, X_2, X_3, X_4, X_5)$.

$X_1 \backslash X_2$	0	1	2
0	1	2	1
1	1	1	1
2	2	1	2

$P_A[X_1, X_2]$

$X_1 \backslash X_2$	0	1	2
0	0	0	2
1	2	0	0
2	0	2	0

$P_B[X_1, X_2]$

$X_1 \backslash X_2$	0	1	0
0	1	1	0
1	0	1	1
2	1	0	1

$P_C[X_1, X_2]$

$X_1 \backslash X_2$	1	2	0
0	1	2	0
1	0	1	1
2	2	0	2

$P_D[X_1, X_2]$

$X_1 \backslash X_2$	0	1	2
0	0	0	2
1	2	0	0
2	0	2	0

$P_E[X_1, X_2]$

또한, 위 표4의 내용을 각각 2변수 $[X_3, X_4]$ 로 분할하면 다음 표6과 같다.

표 6. 5변수 3치논리 스위칭함수 $F(X_1, X_2, X_3, X_4, X_5)$ 를 변수 X_3 와 X_4 의 변수로 분할한 부분함수 P_G, P_H 의 진리치표.

Table 6. The truth table for the partial switching function P_G, P_H that are partitioned by variable X_3 and X_4 of 5-variable 3-valued switching function $F(X_1, X_2, X_3, X_4, X_5)$.

$X_3 \backslash X_4$	0	1	2
0	$P_A\ P_B\ P_C$		
1	$P_B\ P_B\ P_A$		
2	$P_B\ P_A\ P_B$		

$P_G[X_3, X_4]$

$X_3 \backslash X_4$	0	1	2
0	0	$P_D\ P_E$	
1	$P_D\ P_D\ 0$		
2	$P_D\ 0\ P_D$		

$P_H[X_3, X_4]$

또한, 위 표4의 내용을 1변수 [X5]로 분할하면 다음 표 7과 같다.

표 7. 5변수 3치논리 스위칭함수 F(X₁, X₂, X₃, X₄, X₅)를 변수 X₅로 분할한 부분함수 P_K의 진리치표.
Table 7. The truth table for the partial switching function P_K, that are partitioned by variable X₅ of 5-variable 3-valued switching function F(X₁, X₂, X₃, X₄, X₅).

X ₅		
0	1	2
P _G	P _H	P _G

P_K[X₅]

또한, 표5의 부분함수 중 P_A는 다음 표8과 같이 P_{K_A}의 부분함수로 분해가 가능하다.

표 8. 부분함수 P_A의 부분함수 P_{K_A}
Table 8. The partial function P_{K_A} of the partial function P_A.

	X ₂	0	1	2		X ₂	0	1	2
X ₁	\	0	1	2	X ₁	\	0	1	2
0		1	*	1	0		0	2	0
1		1	1	1	1		0	0	0
2		*	1	1	2		2	0	2

1g_{1A}

g_{2A}

또한, 표5의 부분함수 중 P_D는 다음 표9와 같이 P_{K_D}의 부분함수로 분해가 가능하다.

표 9. 부분함수 P_D의 부분함수 P_{K_D}
Table 9. The partial function P_{K_D} of the partial function P_D.

	X ₂	0	1	2		X ₂	0	1	2
X ₁	\	0	1	2	X ₁	\	0	1	2
0		1	*	0	0		0	2	0
1		0	1	1	1		0	0	0
2		*	0	*	2		2	0	2

1g_{1D}

g_{2D}

다음에 표5의 2변수 [X₁, X₂]의 부분함수들을 식(4)와 (5)에 의해 스위칭함수식으로 나타내면 다음 식들과 같다.

$$P_A = 1.g_{1A} \vee g_{2A} = 1 \vee X_1^{(0)}X_2^{(11)} \vee X_1^{(2)}X_2^{(0,2)} \quad (6)$$

$$P_B = X_1^{(0)}X_2^{(2)} \vee X_1^{(1)}X_2^{(0)} \vee X_1^{(2)}X_2^{(1)} \quad (7)$$

$$P_C = 1.X_1^{(0,2)}X_2^{(2)} \vee 1.X_1^{(0,1)}X_2^{(11)} \vee 1.X_1^{(1,2)}X_2^{(2)} \quad (8)$$

$$P_D = 1.g_{1D} \vee g_{2D} = 1.X_1^{(0)}X_2^{(0,11)} \vee 1.X_1^{(11)}X_2^{(1,21)} \vee X_1^{(0)}X_2^{(11)} \vee X_1^{(2)}X_2^{(0,2)} \quad (9)$$

$$P_E = X_1^{(0,1)}X_2^{(0,2)} \vee X_1^{(1,2)}X_2^{(0,2)} \quad (10)$$

여기서 위 식(6)부터 (10)에서 공통 리터럴함수는 식(6)과 (9)의 X₁⁽⁰⁾X₂⁽¹¹⁾과 X₁⁽²⁾X₂^(0,2)이다.

또한 표6으로부터 부분함수 P_G와 P_H에 대한 리터럴함수는 다음 식(11) 및 (12)와 같다.

$$P_G = P_A.X_3^{(0)}X_4^{(0)} \vee P_A.X_3^{(11)}X_4^{(2)} \vee P_A.X_3^{(2)}X_4^{(11)} \vee P_B.X_3^{(0)}X_4^{(11)} \vee P_B.X_3^{(11)}X_4^{(0,1)} \vee P_B.X_3^{(2)}X_4^{(0,2)} \vee P_C.X_3^{(0)}X_4^{(2)} \quad (11)$$

$$P_H = P_D.X_3^{(0)}X_4^{(11)} \vee P_D.X_3^{(2)}X_4^{(0,11)} \vee P_D.X_3^{(2)}X_4^{(0,2)} \vee P_E.X_3^{(11)}X_4^{(2)} \quad (12)$$

위 식(6)부터 (10)으로부터 2변수 [X₁, X₂]의 부분 PLA와 식(11)과 (12)로부터 2변수 [X₃, X₄]의 부분 PLA를 중속접속하고 부호기(encoder)로 T-gate를 사용하고, T-gate의 제어입력으로 표7의 1변수 스위칭함수 P_K(X₅)를 사용하여 최종의 PLA를 사용하여 디지털논리스위칭함수에 대한 디지털논리시스템을 구성하면 다음 그림5와 같다.

본 논문에서 제안한 PLA를 사용하여 디지털논리스위칭함수를 구현하는 방법과 기존의 논문과의 결과를 비교 및 검토한 내용을 다음 표10에 나타내었다.

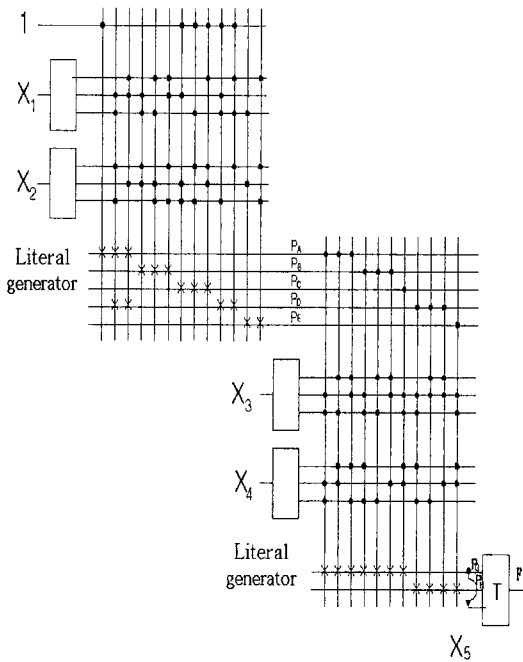


그림 5. 3치PLA를 사용하여 디지털논리스위칭함수 $F(X_1, X_2, X_3, X_4, X_5)$ 의 회로실현
 Fig. 5. The circuit realization of the digital logic switching function $F(X_1, X_2, X_3, X_4, X_5)$ using 3-valued PLA

표 10. 비교표.
 Table 10. The comparison table.

Comparison items	K.Y.Fang's paper	This paper
The number of sub-PLA column for 2-variable (X_1, X_2)	16	13
The number of sub-PLA column for 2-variable (X_3, X_4)	15	11
The number of T-gate	13	1
The number of literal generator	0	4
The complexity of input	complexity	simple

VI. 결 론

일반적으로 최소면적의 PLA를 구성하기 위해서 임의의 디지털논리스위칭함수를 최소의 부분함수로 나누어야 하며, 각각의 부분함수에 대하여 최소의 리터럴 함수로 표시하여야 한다. VLSI를 설계할 때 설계의 복잡성을 해결하고 테스트를 용이하게 하기 위하여 PLA와 같은 규칙적인 구조를 갖는 소자를 사용하는 것이 바람직하다. 제안한 PLA를 사용한 디지털논리스위칭함수 구현 설계방법은 기존의 방법에 비해 좀 더 크기를 줄일 수 있으며 정규성과 확장성을 갖는 장점이 있다. 향후 연구 과제로는 입력단의 리터럴 생성기와 출력단의 좀 더 효율적인 구성의 연구가 요구된다.

참고문헌

- [1] M.Davio, J.Deschamps and A.Thayse, *Discrete and Switching Functions*, McGraw-Hill Inc. 1978.
- [2] D. Green, *Modern Logic Design*, Addison wesely, 1986.
- [3] M.D.Ercegovac, and T.Lang, *Digital Systems and hardware/Firmware Algorithms*, John Wiley & Sons, 1985.
- [4] D.Lee, A.A. Gaffar, O.Mencer, and W.Luk, "Optimization Hardware Function Evaluation," *IEEE Trans. Comput.*, vol.54, No.12, pp.1520-1531, Dec., 2005.
- [5] S.Mitra, N.R.Saxena, and E.J.McCluskey, "Efficient Design Diversity Estimation for Combinational Circuits," *IEEE Trans. Comput.*, vol.53, No.11, pp.1483-1492, Nov., 2004.
- [6] T.Sasao, "Input variable assignment and output phase optimization of PLA's," *IEEE Trans. Comput.*, vol. C-33, pp. 879-894, Oct. 1984.
- [7] H.L.Kuo and K.Y.Fang, "The multiple-valued programmable logic array and application in modular design," in *Proc. ISMVL-85*, pp.10-18. May 1985.
- [8] M.Kameyama, "Toward The Age of Beyond-Binary Electronics and Systems", *IEEE Proc. 20th ISMVL*, pp.162-166, May, 1990.

- [9] M.Nakajima and M.Kameyama, "Design of Highly Parallel Linear Digital System for ULSI Processors", IEICE Trans, Vol.E76-C, No.7, pp.1119-1125, July, 1993.

저자소개

박춘명(Chun-Myoung Park)

제10권 12호 참조