

## 74181 Arithmetic Logic Units의 분석

이재석\*, 정태상\*\*

\*중앙대학교 제어계측학과, \*\*중앙대학교 전자전기공학부

## Analysis of 74181 Arithmetic Logic Units

Jae-Seok Lee\*, Tae-Sang Chung\*\*

\*Dpt. of Control and Instrumentation Eng., \*\*School of Electrical and Electronics Eng., Chung-Ang Univ.

**Abstract** - The 74181 is arithmetic logic units(ALU)/function generator. This circuit performs 16 binary arithmetic operations on two 4-bit words. And a full carry look-ahead scheme is made available in this device. The 74181 can also be utilized as a comparator. This circuit has been also designed to provide 16 possible functions of two Boolean variables without the use of external circuitry.

This paper analyzes the function of the logic and the implementation adopted in the design of 74181. The understanding of the logic characteristics of this chip enables us to improve future applications.

## 1. 서 론

74181은 산술논리회로(Arithmetic Logic Units, ALU)로서 두 개의 4-bit word를 입력 받아 XOR, Comparator, AND, NAND, OR, NOR 등 16가지의 논리연산과, 덧셈, 뺄셈, 쉬프트연산 등 16가지의 산술연산을 수행한다.

논리연산과 산술연산을 구별하기 위해 모드 선택 입력  $M$ 이 있는데,  $M$ 이 High이면 논리연산을 수행하고,  $M$ 이 Low이면 산술연산을 수행한다. 그리고 각각의 경우 연산의 종류를 선택하기 위한 4-bit의 선택 입력  $S_3, S_2, S_1, S_0$ 이 있다. 즉, 모드 선택 입력  $M$ 과 연산의 종류를 선택하는 입력  $S_3, S_2, S_1, S_0$ 에 따라 모두 32가지의 연산을 수행할 수 있다.

이 외에도 74181은 내부적으로 look-ahead carry를 생성하는데, 이는 cascade 출력으로 74182 Look-ahead Carry 생성기와 연결하여 고속으로 동작하는 다단계의 가산기나 ALU를 구성하는데 사용될 수 있다.

이 논문에서는 74181의 논리함수와 그것의 구현에 대해 상세하게 분석하여 74181이 32가지의 연산을 하나의 칩에 어떻게 효과적으로 통합하여 디자인하였는지 살펴보고 실제 덧셈, 뺄셈 등을 수행할 때 주의해야 할 점에 대해 알아볼 것이다.

## 2. 본 론

## 2.1 74181의 논리 다이어그램

74181은 active-low 출력을 내기 위해 active-low 입력을 받을 수도 있고 active-high를 출력하기 위해 입력으로 active-high를 받을 수도 있다. 각각의 경우 논리 다이어그램과 함수표(function table)가 약간 달라지지만 대동소이하다. 여기서는 active-low를 기준으로 설명하겠다.

## 2.1.1 논리 다이어그램(Active-Low의 경우)

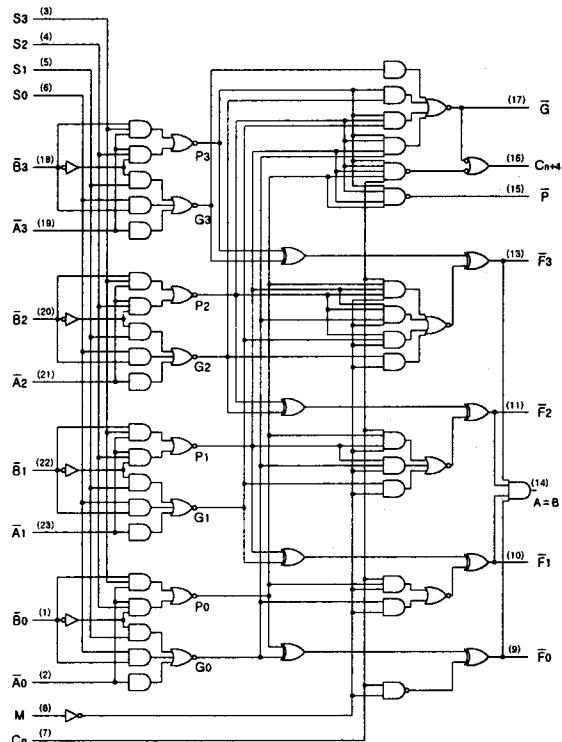


그림 1. 74181의 논리 다이어그램

그림 1은 74181의 active-low일 경우의 논리 다이어그램을 나타낸다.

## 2.2 74181의 Active-Low 논리 해석

$P_3, P_2, P_1, P_0$ 과  $G_3, G_2, G_1, G_0$ 은 다음과 같이 표현된다.

$$\begin{aligned} P_i &= \overline{S_3 \bar{A}_i \bar{B}_i + S_2 \bar{A}_i B_i} \\ &= (\overline{S_3} + A_i + B_i)(\overline{S_2} + A_i + \overline{B}_i) \end{aligned}$$

$$\begin{aligned} G_i &= \overline{\bar{A}_i + S_1 B_i + S_0 \bar{B}_i} \\ &= A_i(\overline{S_1} + \overline{B}_i)(\overline{S_0} + B_i) \end{aligned}$$

$$(i = 3, 2, 1, 0)$$

내부 carry는 다음과 같다.

$$c_0 = \overline{M} C_n$$

$$c_1 = \overline{M} P_0 C_n + \overline{M} G_0 = \overline{M} (P_0 C_n + G_0)$$

$$c_2 = \overline{M} P_1 P_0 C_n + \overline{M} P_1 G_0 + \overline{M} G_1$$

$$= \overline{M} (P_1 P_0 C_n + P_1 G_0 + G_1)$$

$$c_3 = \overline{M} P_2 P_1 P_0 C_n + \overline{M} P_2 P_1 G_0 + \overline{M} P_2 G_1 + \overline{M} G_2$$

$$= \overline{M} (P_2 P_1 P_0 C_n + P_2 P_1 G_0 + P_2 G_1 + G_2)$$

또한  $F_i, G, P, C_{n+4}$ 은 각각 다음과 같다.

$$\begin{aligned} F_i &= (P_i \oplus G_i) \oplus c_i = P_i \oplus G_i \oplus c_i \\ (i &= 3, 2, 1, 0) \end{aligned}$$

$$G = P_3 P_2 P_1 G_0 + P_3 P_2 G_1 + P_3 G_2 + G_3$$

$$P = P_3 P_2 P_1 P_0$$

$$C_{n+4} = G + P_3 P_2 P_1 P_0 C_n$$

여기서  $G, P$ 는 74182와 연결하여 다단계의 ALU나 가산기를 만드는데 사용할 수 있는 group-carry generator와 propagator 신호이다. 또한  $C_{n+4}$ 은 ripple-carry 출력으로 속도가 중요하지 않은 가산기를 구성하는 경우 또 다른 74181의  $C_n$ 에 인가되는 입력으로 사용될 수 있다.

### 2.2.1 논리연산 모드 : $M = 1$

논리연산 모드로 동작하기 위해서는  $M = 1$ 이어야 하는데, 이 경우 내부에서 발생하는 carry들이 모두 0이 된다.

$$c_3 = c_2 = c_1 = c_0 = 0$$

이렇게 되면,  $F_i$ 는 다음과 같다.

$$F_i = (P_i \oplus G_i) \oplus c_i = (P_i \oplus G_i) \oplus 0 = P_i \oplus G_i$$

이 경우,  $S_3, S_2, S_1, S_0$ 에 따라 bit단위의 16가지의 논리 연산을 수행한다.

상세 진리표는 표 1과 같다. 여기서  $S_3, S_2, S_1, S_0$ 의 값에 따라  $P_i, G_i$ 는 다음의 경우처럼 된다.

$$\begin{aligned} P_i &= \overline{S_3 A_i B_i + S_2 A_i B_i} \\ &= (\overline{S_3} + A_i + B_i)(\overline{S_2} + A_i + \overline{B_i}) \end{aligned}$$

$$\begin{cases} P_i(S_3 S_2 = 00) = 1 \\ P_i(S_3 S_2 = 01) = A_i \vee \overline{B_i} \\ P_i(S_3 S_2 = 10) = A_i \vee B_i \\ P_i(S_3 S_2 = 11) = A_i \end{cases}$$

$$\begin{aligned} G_i &= \overline{\overline{A_i} + S_1 B_i + S_0 \overline{B_i}} \\ &= A_i(\overline{S_1} + \overline{B_i})(\overline{S_0} + B_i) \end{aligned}$$

$$\begin{cases} G_i(S_1 S_0 = 00) = A_i \\ G_i(S_1 S_0 = 01) = A_i B_i \\ G_i(S_1 S_0 = 10) = A_i \overline{B_i} \\ G_i(S_1 S_0 = 11) = 0 \end{cases}$$

( $\vee$  : 논리 OR)

$S_3 S_2 S_1 S_0$	$P_i$	$G_i$	$M = 1$
			논리연산
			$G_i \oplus P_i$
0 0 0 0	논리 1	$A_i$	$\overline{A_i}$
0 0 0 1		$A_i B_i$	$\overline{A_i} \vee \overline{B_i}$
0 0 1 0		$A_i \overline{B_i}$	$\overline{A_i} \vee B_i$
0 0 1 1		논리 0	논리 1
0 1 0 0		$A_i$	$\overline{A_i} B_i$
0 1 0 1	$A_i \vee \overline{B_i}$	$A_i B_i$	$\overline{B_i}$
0 1 1 0		$A_i \overline{B_i}$	$\overline{A_i} \oplus B_i$
0 1 1 1		논리 0	$A_i \vee \overline{B_i}$
1 0 0 0		$A_i$	$\overline{A_i} B_i$
1 0 0 1		$A_i B_i$	$A_i \oplus B_i$
1 0 1 0	$A_i \vee B_i$	$A_i \overline{B_i}$	$B_i$
1 0 1 1		논리 0	$A_i \vee B_i$
1 1 0 0		$A_i$	논리 0
1 1 0 1		$A_i B_i$	$A_i \overline{B_i}$
1 1 1 0		$A_i \overline{B_i}$	$A_i B_i$
1 1 1 1		논리 0	$A_i$

표 1.  $M = 1$ 일 때의 74181의 진리표

### 2.2.2 산술연산 모드 : $M = 0$

$M = 0$ 이면 산술연산을 수행하는데, 이 경우 내부에서 carry  $c_0, c_1, c_2, c_3$ 들이 발생하여 산술연산에 참여한다.

$$\begin{aligned} c_0 &= C_n \\ c_1 &= P_0 C_n + G_0 \\ c_2 &= P_1 P_0 C_n + P_1 G_0 + G_1 \\ c_3 &= P_2 P_1 P_0 C_n + P_2 P_1 G_0 + P_2 G_1 + G_2 \end{aligned}$$

그리고  $F_i$ 는 다음처럼 된다.

$$F_i = (P_i \oplus G_i) \oplus c_i = P_i \oplus G_i \oplus c_i$$

만일  $P_3 P_2 P_1 P_0$ 와  $G_3 G_2 G_1 G_0$ 를 carry look-ahead 덱 셉방식으로 더할 때, carry propagator와 carry generator  $p_i$ 와  $g_i$ 가 다음의 관계를 만족한다면.

$$\begin{cases} p_i = P_i + G_i = P_i \\ g_i = P_i G_i = G_i \end{cases}$$

$c_0, c_1, c_2, c_3$ 는 look-ahead carry가 되고 따라서  $F_i = G_i \oplus P_i \oplus c_i$ 는  $G_i, P_i$  그리고  $c_i$ 의 3비트를 더한 sum bit가 된다. 이 때에도  $P_i, G_i$ 는  $S_3, S_2, S_1, S_0$ 의 값에 따라 논리연산 모드의 경우와 동일하게 된다. 세부 진리표는 표 2에 나타나 있다.

( $\vee$  : 논리 OR,  $+$  : 산술 덧셈)

		$M = 0$ : 산술연산	
$S_3S_2$	$S_1S_0$	$c_i$ 는 $G_3G_2G_1G_0$ 과 $P_3P_2P_1P_0$ 를 더하는데 필요한 look-ahead carry가 된다.	
Propagator	Generator	$F = P + G + C_n$	
$G_i \vee P_i = P_i$	$G_i P_i = G_i$		
0 0 0 0		$A_i$	$-1 + A + C_n$
0 0 0 1	논리 1	$A_i B_i$	$-1 + AB + C_n$
0 0 1 0		$A_i \bar{B}_i$	$-1 + A\bar{B} + C_n$
0 0 1 1		논리 0	$-1 + C_n$
0 1 0 0	$A_i \vee \bar{B}_i$	$A_i$	$(A \vee \bar{B}) + A + C_n$
0 1 0 1		$A_i B_i$	$(A \vee \bar{B}) + AB + C_n$
0 1 1 0		$A_i \bar{B}_i$	$(A \vee \bar{B}) + A\bar{B} + C_n$
0 1 1 1		논리 0	$(A \vee \bar{B}) + C_n$
1 0 0 0	$A_i \vee B_i$	$A_i$	$(A \vee B) + A + C_n$
1 0 0 1		$A_i B_i$	$(A \vee B) + AB + C_n$
1 0 1 0		$A_i \bar{B}_i$	$(A \vee B) + A\bar{B} + C_n$
1 0 1 1		논리 0	$(A \vee B) + C_n$
1 1 0 0	$A_i$	$A_i$	$2A + C_n$
1 1 0 1		$A_i B_i$	$A + AB + C_n$
1 1 1 0		$A_i \bar{B}_i$	$A + A\bar{B} + C_n$
1 1 1 1		논리 0	$A + C_n$

표 2.  $M = 0$ 일 때의 74181의 진리표

여기서 덧셈과 뺄셈을 2's-complement(2의 보수)방식으로 계산하기 위해서는 다음과 같이 한다. 먼저 뺄셈의 경우  $S_3S_2S_1S_0 = 0110$ 을 인가하면  $F = F_3F_2F_1F_0$ 는  $(A \vee \bar{B}) + A\bar{B} + C_n$ 가 되는데, 이것은 다시  $C_n = 0$ 일 때  $F = A - B - 1$ 이 되고 따라서  $C_n = 1$ 을 인가해야만  $F = A - B$ 를 구할 수 있다.

또한, 덧셈의 경우에는  $S_3S_2S_1S_0 = 1001$ 인 경우에  $F = (A \vee B) + AB + C_n$ 가 되는데, 이것은 다시  $C_n = 0$ 일 때,  $F = A + B$ 가 되어 두 수의 덧셈을 구할 수 있다.

74181은 비교기로도 동작할 수 있는데, 이는 산술연산의 뺄셈 모드에서 가능하다. 즉  $S_3S_2S_1S_0 = 0110$ 이고  $C_n = 1$ 이면  $A - B$ 가 되므로, 이 때 두 입력이 같으

면, 그 결과  $F_3F_2F_1F_0$ 은 0000이 되므로 다음과 같이 된다.

$$(A \equiv B) = \overline{F_3} \overline{F_2} \overline{F_1} \overline{F_0} = 1 \text{ (논리)}$$

이는 입력한 두 수가 같음을 나타낸다.

### 3. 결 론

이 논문에서는 74181의 설계를 분석하여 74181의 논리와 구현에 대해서 알아보았다. 74181은 32가지의 연산을 하나의 칩에서 수행할 수 있도록 간결하게 디자인되었는데, 특히 산술연산의 경우 carry look-ahead 덧셈방식을 채용하여 고속의 연산이 가능하도록 디자인되었다. 앞에서 보았듯이 32가지의 연산과 carry look-ahead 덧셈방식을 하나로 통합하기 위하여 74181은 재치 있는 방식으로 설계되었다.

74181에서 덧셈과 뺄셈을 2의 보수 방식으로 계산하기 위해서는, 전술한 것처럼  $S_3S_2S_1S_0$ 의 값뿐만 아니라  $C_n$ 의 값도 다르게 인가해야 한다.

또 다른 MSI ALU로는 74381과 74382 등이 있다. 이것들은 74181의 연산 중 핵심적인 논리 및 산술연산 8가지만을 수행할 수 있도록 구성되어졌는데, 따라서 74181의 이해가 진행된다면 74381이나 74382 등을 쉽게 분석할 수 있다. 나아가 74381이나 74382처럼 74181의 여러 가지 연산 중 자신의 목적에 맞게 필요한 연산만을 골라 수행할 수 있는 새로운 ALU를 설계할 수도 있다.

### (참 고 문 헌)

- [1] 정태상, "Supplemental Notes", pp. 121-133, 2000
- [2] John F. Wakerly, "Digital Design: principles and practices", Prentice-Hall, pp. 439-443, 2000
- [3] Texas Instruments Inc., "SN74LS181, 74LS381A, and SN74S182 Datasheets"