

산술 논리 연산장치

02 산술 논리 연산 장치

❖ **산술 논리 연산 장치**(Arithmetic Logic Unit, ALU) : 산술 연산과 논리 연산

- 주로 정수 연산을 처리
- 부동 소수(Floating-point Number) 연산 : FPU(Floating-Point Unit)
- 최근에는 ALU가 부동 소수 연산까지 처리

❖ 산술 연산 : 덧셈, 뺄셈, 곱셈, 나눗셈, 증가, 감소, 보수

❖ 논리 연산 : AND, OR, NOT, XOR, 시프트(shift)

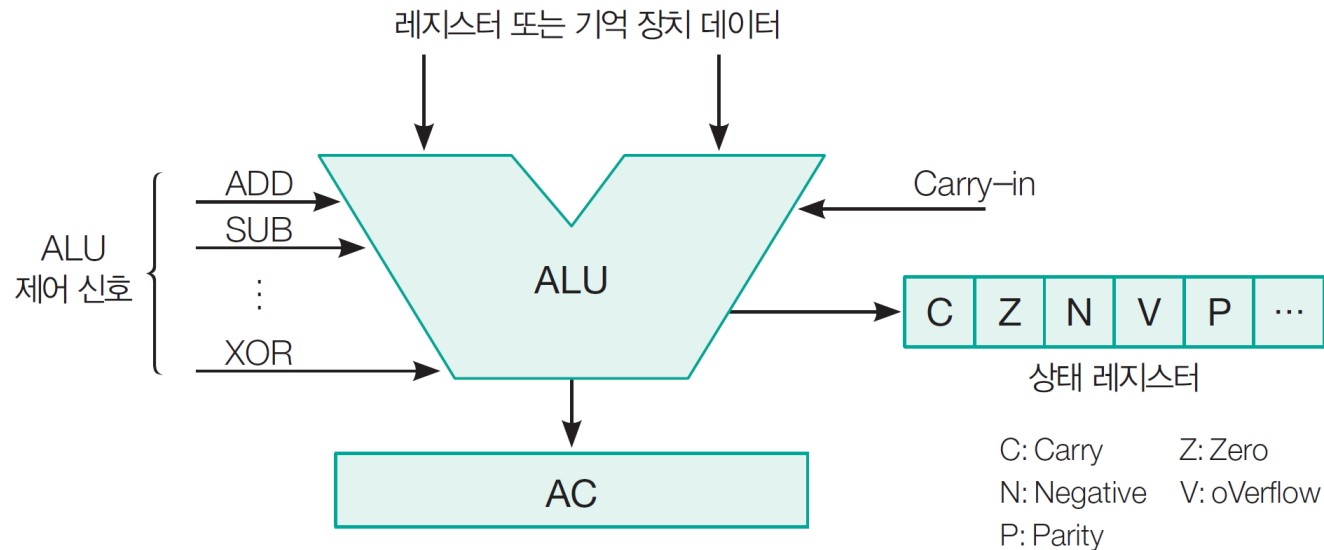


그림 4-4 ALU의 동작

02 산술 논리 연산 장치

1 산술 연산

표 4-1 산술 연산

연산	8비트 연산	
	동작	설명
ADD	$X \leftarrow A + B$	A와 B를 더한다.
SUB	$X \leftarrow A + (\sim B + 1)$	A + (B의 2의 보수)
MUL	$X \leftarrow A * B$	A와 B를 곱한다.
DIV	$X \leftarrow A / B$	A와 B를 나눈다.
INC	$X \leftarrow A + 1$	A를 1 증가시킨다.
DEC	$X \leftarrow A - 1(0xFF)$	A를 1 감소시킨다.
NEG	$X \leftarrow \sim A + 1$	A의 2의 보수다.

02 산술 논리 연산 장치

❖ Booth Algorithm(2진수 곱셈)

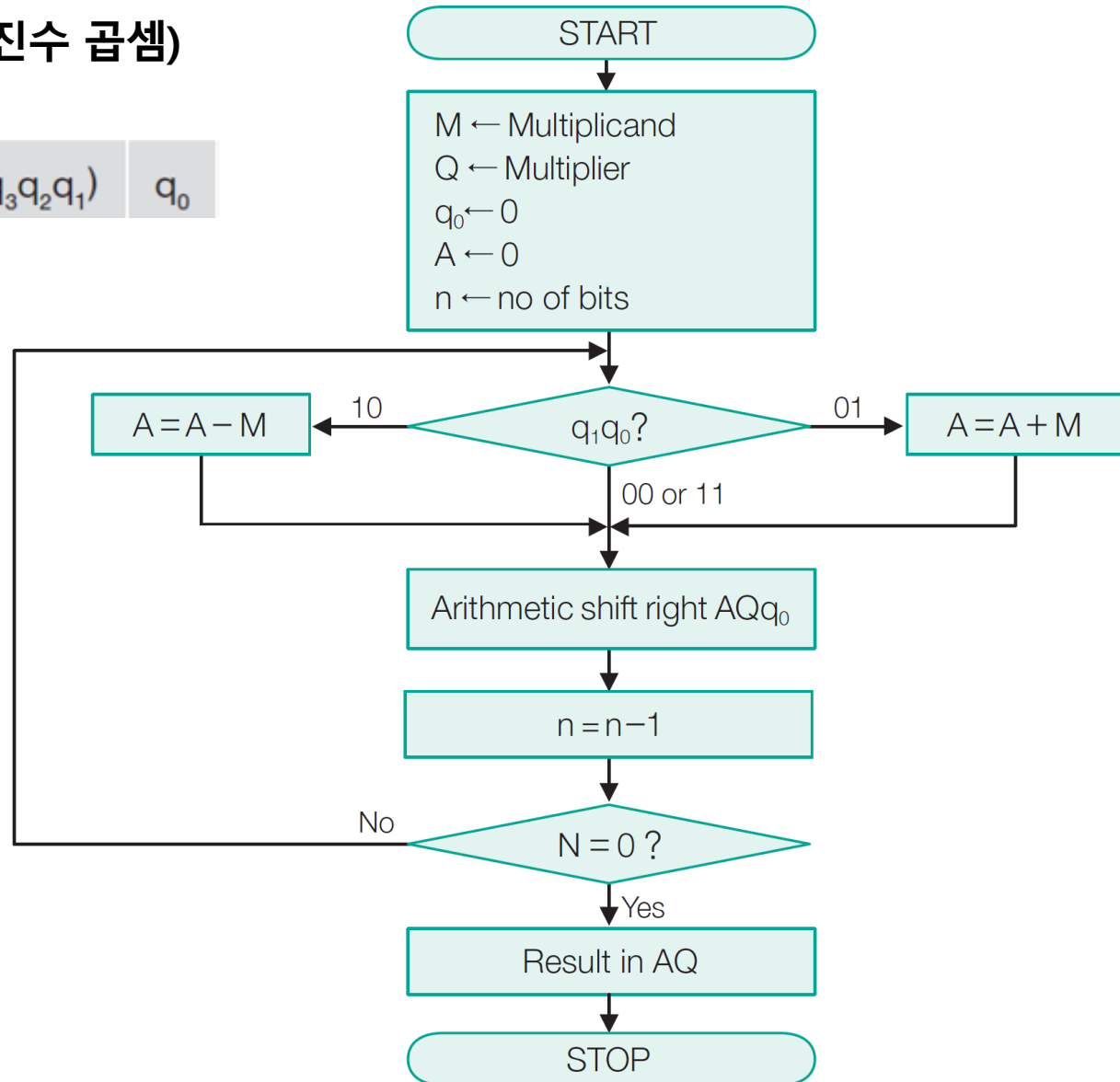
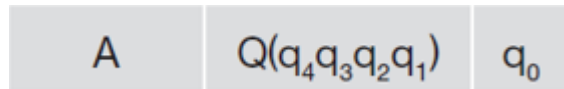


그림 4-5 부스 알고리즘 순서도

02 산술 논리 연산 장치

❖ Booth Algorithm 예 : $(-7) * (+3)$

n	A	Q($q_4q_3q_2q_1$)	q_0	설명
4	0000	0011	0	초기 상태
	0111	0011	0	q_1q_0 이 10이므로 $A=A-M=0000+0111$
3	0011	1001	1	AQ q_0 을 오른쪽 산술 시프트
2	0001	1100	1	q_1q_0 이 11이므로 연산 없이 AQ q_0 을 오른쪽 산술 시프트
	1010	1100	1	q_1q_0 이 01이므로 $A=A+M=0001+1001$
1	1101	0110	0	AQ q_0 를 ASR ASR:Arithmetic Shift Right
0	1110	1011	0	q_1q_0 가 00이므로 연산 없이 AQ q_0 을 오른쪽 산술 시프트

$(-7)*(+3)$ 에서 피승수 M 은 -7 이고 승수 Q 는 3이다.

A 와 q_0 의 초기값은 0으로 두고 계산한다. 즉, 여기서 $Q=0011_2$,
 $M=-7_{10}=1001_2$, $-M=0111_2$ 이다. 따라서 $AQ=11101011_2=-21_{10}$ 이 된다.

02 산술 논리 연산 장치

❖ Booth Algorithm 예 : $5 * (-4)$

n	A	$Q(q_4q_3q_2q_1)$	q_0	설명
4	0000	1100	0	초기 상태
3	0000	0110	0	q_1q_0 이 00이므로 연산 없이 AQq_0 을 오른쪽 산술 시프트
2	0000	0011	0	q_1q_0 이 00이므로 연산 없이 AQq_0 을 A오른쪽 산술 시프트
1	1011	0011	0	q_1q_0 이 10이므로 $A=A-M=0000+1011$
	1101	1001	1	AQq_0 을 오른쪽 산술 시프트
0	1110	1100	1	q_1q_0 이 11이므로 연산 없이 AQq_0 을 오른쪽 산술 시프트

(M) $5=0101$, $(-M) -5=1011$, $-4=1100$

$1110 \ 1100 = -00010100 = -20$

Booth Algorithm 원리 보충

23x30의 연산을 2진수로 수행 시 승수의 1의 갯수만큼 덧셈

	010111 (23)									
	011110 (30)									
					0	1	0	1	1	1
				×	0	1	1	1	1	0
				0	1	0	1	1	1	
+			0	1	0	1	1	1		
			1	0	0	0	1	0	1	0
+		0	1	0	1	1	1			
		1	0	1	0	0	0	0	1	0
+	0	1	0	1	1	1				
	1	0	1	0	1	1	0	0	1	0

Booth Algorithm 원리(앞의 $23 \times 30 = 010111 \times 011110$)

곱하는 수 $011110 = 100000 - 000010$ 으로 생각할 수 있다. 그래서
 $010111 \times (100000 - 000010) = 010111 \times 100000 - 010111 \times 000010$

010111 × 100000 – 010111 × 000010										
					0	1	0	1	1	1
				x	1	0	0	0	0	0
0	1	0	1	1	1					
					0	1	0	1	1	1
				x	0	0	0	0	1	0
				0	1	0	1	1	1	
	1	0	1	0	1	1	0	0	1	0

한 번의 뺄셈으로 해결가능

```

10111 00000
+11110 10010
-----
10101 10010

```


02 산술 논리 연산 장치

2 논리 연산과 산술 시프트 연산

표 4-2 논리 연산

연산	8비트 연산	
	동작	설명
AND	$X \leftarrow A \& B$	A와 B를 비트 단위로 AND 연산한다.
OR	$X \leftarrow A B$	A와 B를 비트 단위로 OR 연산한다.
NOT	$X \leftarrow \sim A$	A의 1의 보수를 만든다.
XOR	$X \leftarrow A \wedge B$	A와 B를 비트 단위로 XOR 연산한다.
ASL	$X \leftarrow A \ll n$	왼쪽으로 n비트 시프트(LSL과 같다.)
ASR	$X \leftarrow A \gg n, A[7] \leftarrow A[7]$	오른쪽으로 n비트 시프트(부호 비트는 그대로 유지한다.)
LSL	$X \leftarrow A \ll n$	왼쪽으로 n비트 시프트
LSR	$X \leftarrow A \gg n$	오른쪽으로 n비트 시프트
ROL	$X \leftarrow A \ll 1, A[0] \leftarrow A[7]$	왼쪽으로 1비트 회전 시프트, MSB는 LSB로 시프트
ROR	$X \leftarrow A \gg 1, A[7] \leftarrow A[0]$	오른쪽으로 1비트 회전 시프트, LSB는 MSB로 시프트
ROLC	$X \leftarrow A \ll 1, C \leftarrow A[7], A[0] \leftarrow C$	캐리도 함께 왼쪽으로 1비트 회전 시프트
RORC	$X \leftarrow A \gg 1, C \leftarrow A[0], A[7] \leftarrow C$	캐리도 함께 오른쪽으로 1비트 회전 시프트

02 산술 논리 연산 장치

❖ 논리 연산 예 1 : $A=46=00101110_{(2)}$, $B=-75=10110101_{(2)}$

A AND B	A OR B	A XOR B
00101110 46	00101110 46	00101110 46
& 10110101 -75	10110101 -75	^ 11111111 -128
00100100 36	10111111 -65	11010001 -47

02 산술 논리 연산 장치

❖ 논리 연산 예 2

A AND B		A OR B	
00101110		00001110	
& 00001111	상위 4비트 삭제	10110000	상위 4비트 값 설정
00001110		10111110	

02 산술 논리 연산 장치

❖ 시프트 연산 예

연산	왼쪽	오른쪽
산술 시프트	<p>MSB LSB</p> <p>ASL [0][0][0][1][0][1][1][0]</p> <p>[0][0][1][0][1][1][0][0] ← 0</p>	<p>MSB LSB</p> <p>ASR [1][0][0][1][0][1][1][0]</p> <p>[1][1][0][0][1][0][1][1]</p>
논리 시프트	<p>MSB LSB</p> <p>LSL [0][0][0][1][0][1][1][0]</p> <p>[0][0][1][0][1][1][0][0] ← 0</p>	<p>MSB LSB</p> <p>LSR [1][0][0][1][0][1][1][0]</p> <p>0 → [0][1][0][0][1][0][1][1]</p>
회전 시프트	<p>MSB LSB</p> <p>ROL [0][0][0][1][0][1][1][0]</p> <p>[0][0][1][0][1][1][0][0]</p>	<p>MSB LSB</p> <p>ROR [1][0][0][1][0][1][1][0]</p> <p>[0][1][0][0][1][0][1][1]</p>
캐리와 함께 회전 시프트	<p>MSB LSB C</p> <p>ROLC [0][0][0][1][0][1][1][0] 0</p> <p>[0][0][1][0][1][1][0][0] 0</p>	<p>MSB LSB C</p> <p>RORC [0][0][0][1][0][1][1][0] 1</p> <p>[1][0][0][0][1][0][1][1] 1</p>

수고하셨습니다!