

Slide 0

10장. 컴퓨터 산술 연산

Slide 1

덧셈과 뺄셈

- 부호 절대값 데이터를 이용한 덧셈과 뺄셈
 - 두수의 덧셈 뺄셈시 부호와 연산에 따라 8가지 경우가 생김 (표 10-1)

Operation	Add Magnitudes	Subtract Magnitudes		
		When $A > B$	When $A < B$	When $A = B$
$(+A) + (+B)$	$ A + B $	$ A - B $	$-(B - A)$	$+(A - B)$
$(+A) + (-B)$	$ A - B $	$-(A - B)$	$+(B - A)$	$+(A - B)$
$(-A) + (+B)$	$ A + B $	$ A - B $	$-(B - A)$	$+(A - B)$
$(-A) + (-B)$	$ A + B $	$ A - B $	$-(B - A)$	$+(A - B)$
$(+A) - (+B)$	$+ A - B $	$+ A - B $	$-(B - A)$	$+(A - B)$
$(+A) - (-B)$	$+ A + B $	$+ A + B $	$-(B - A)$	$+(A - B)$
$(-A) - (+B)$	$+ A + B $	$+ A + B $	$+(B - A)$	$+(A - B)$
$(-A) - (-B)$	$- A - B $	$- A - B $	$+(B - A)$	$+(A - B)$

- 덧셈 (뺄셈) 알고리즘
 - A, B 의 부호가 같으면 (다르면) 두수의 크기를 더해서 A 의 부호를 결과 부호로
 - A, B 의 부호가 다르면 (같으면) 크기를 비교 큰수에서 작은수를 뺀다음
 - $A > B$ 이면 결과부호로 A 의 부호를 취함
 - $A < B$ 이면 A 부호의 보수를 결과 부호로 취함

덧셈과 뺄셈 (계속)

- * 두수가 같으면 A - B 를 하고 결과 부호를 양으로 함

- 하드웨어 구성

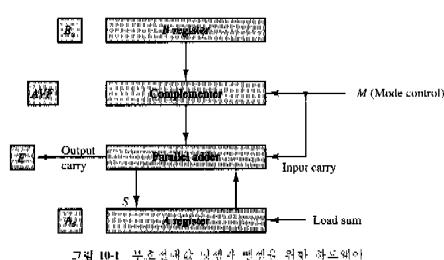
Slide 2

- * 필요한 하드웨어
 1. 병렬 가산기
 2. 두개의 병렬 감산기 → 크기 비교 및 감산에 사용
 3. exclusive-OR 게이트
- * 뺄셈시 2의 보수를 이용하면 병렬 가산기와 보수기만 필요
- * 부호 절대값 덧셈과 뺄셈을 위한 하드웨어 (그림 10-1)

덧셈과 뺄셈 (계속)

프로그램 구성을 위하여, 병렬 가산기는 그림 4-7에 있는 전자산기과 구성을 둘다.
M 신호는 가산기의 입력 케이블로 돌아온다. 따라서 M=0이면 가산기에서 A
B의 합역과 입력 케이블 0이 들어가므로 기산기의 출력은 A + B가 된다. M

Slide 3



- 덧셈시 $M = 0$ 임으로 $A + B$
- 뺄셈시 $M = 1$ 임으로 $A + B' + 1 = A - B$
- 하드웨어 알고리즘 (그림 10-2)

덧셈과 뺄셈 (계속)

Slide 4

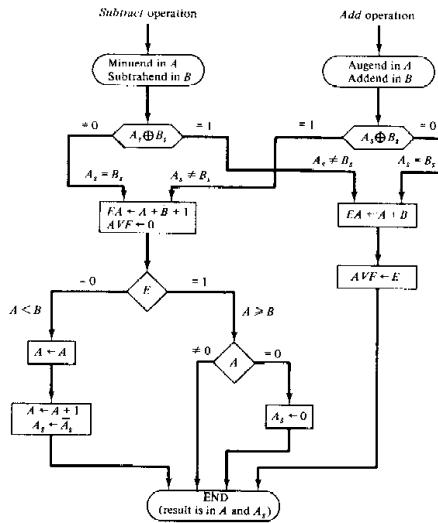


그림 10-2 덧셈과 뺄셈 연산을 위한 프로토

덧셈과 뺄셈 (계속)

Slide 5

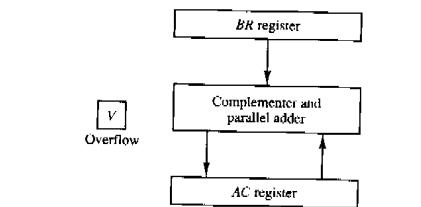


그림 10-3 부호가 있는 2의 보수 데이터를 이용한 덧셈과 뺄셈

- 부호가 있는 2의 보수 데이터를 이용한 덧셈과 뺄셈
 - 부호 2의 보수로 표현된 수의 덧셈과 뺄셈을 위한 하드웨어 (그림 10-3)

- overflow : msb \oplus msb - 1 의 결과가 1일 때 발생

- 알고리즘 (그림 10-4)

덧셈과 뺄셈 (계속)

Slide 6

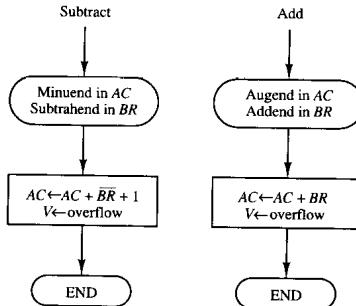


그림 10-4 부호가 있는 2의 보수로 표현된 숫자의 덧셈과 뺄셈 알고리즘

→ 매우 간단 → 대부분의 컴퓨터에서 사용

곱셈 알고리즘

- 부호 절대값 형식의 고정 소숫점 이진수의 곱셈

- 과정

Slide 7

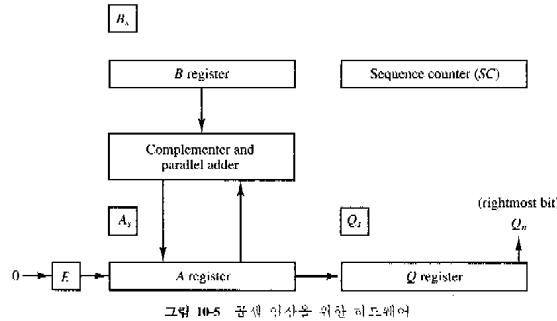
23	10111	Multiplicand
19	10011	Multiplier
--	-----	
	10111	
	10111	
	00000	
	00000	
	10111	+

437	110110101	Product

곱셈 알고리즘 (계속)

- 하드웨어 구성 (그림 10-5)

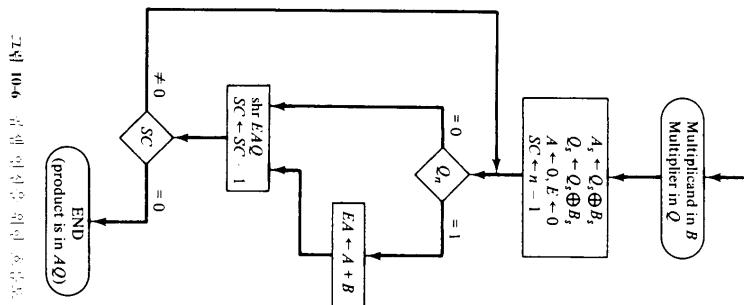
Slide 8



- 하드웨어 알고리즘 (그림 10-6)

곱셈 알고리즘 (계속)

Slide 9



- 이진 승산기에 대한 숫자 예 (표 10-2)

곱셈 알고리즘 (계속)

Slide 10

표 10-2 이진 승산기에 대한 숫자 예

Multiplicand $B = 10111$	E	A	Q	SC
Multiplier in Q	0	00000	10011	101
$Q_n = 1$; add B		<u>10111</u>		
First partial product	0	10111		
Shift right EAQ	0	01011	11001	100
$Q_n = 1$; add B		<u>10111</u>		
Second partial product	1	00010		
Shift right EAQ	0	10001	01100	011
$Q_n = 0$; shift right EAQ	0	01000	10110	010
$Q_n = 0$; shift right EAQ	0	00100	01011	001
$Q_n = 1$; add B		<u>10111</u>		
Fifth partial product	0	11011		
Shift right EAQ	0	01101	10101	000
Final product in $AQ = 0110110101$				

- Booth의 곱셈 알고리즘

- 방법 설명 예

- * $001110 (+ 14) = 2^4 - 2^1 = 16 - 2 = 14$ 이다

- * $M \times 14 = M \times 2^4 - M \times 2^1$ 이된다

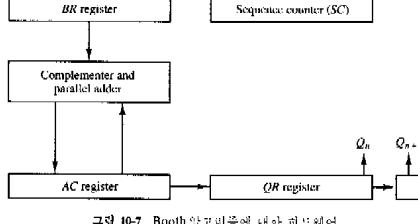
- * 피승수 M 은 오른쪽으로 한비트씩 이동하여 AC 와 연산됨으로 부분곱 계산됨

곱셈 알고리즘 (계속)

Slide 11

- * 이때 연산은 다음의 규칙을 따름
 - . 오른쪽 비트부터 따져서 0 → 1 이면 피승수를 빼고
 - . 1 → 0 이 되면 부분곱에 피승수를 더한다

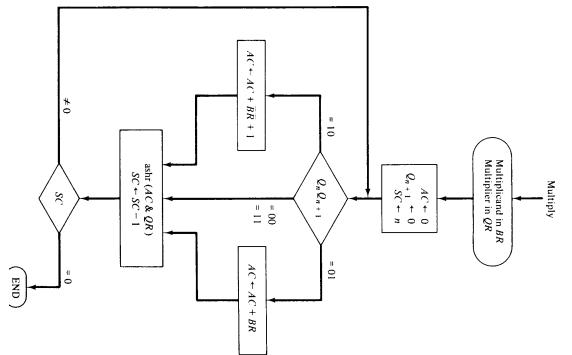
- 하드웨어 구성 (그림 10-7)



- 알고리즘 (그림 10-8)

곱셈 알고리즘 (계속)

Slide 12



- 곱셈 예 (표 10-3)

곱셈 알고리즘 (계속)

Slide 13

표 10-3 Booth 알고리즘을 이용한 곱셈 예					
$Q_n Q_{n+1}$	$BR = 10111$ $BR + 1 = 01001$	AC	QR	Q_{n+1}	SC
1 0	Initial Subtract BR	00000 01001 01001	10011	0	101
1 1	ashr	00100	11001	1	100
0 1	ashr Add BR	00010 10111 11001	01100	1	011
0 0	ashr	11100	10110	0	010
1 0	ashr Subtract BR	11110 01001 00111	01011	0	001
	ashr	00011	10101	1	000

배열 승산기

- 두 이진수의 곱셈을 조합회로를 이용 하나의 마이크로 연산으로 수행
- 빠른 방법이나 많은 수의 게이트를 필요로 함
- 두비트 이진수의 곱셈 조합회로 예 (그림 10-9)

Slide 14

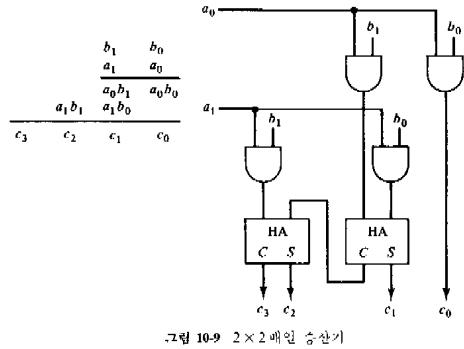


그림 10-9 2 × 2 배열 승산기

배열 승산기 (계속)

- 4×3 배열 승산기 (그림 10-10)

Slide 15

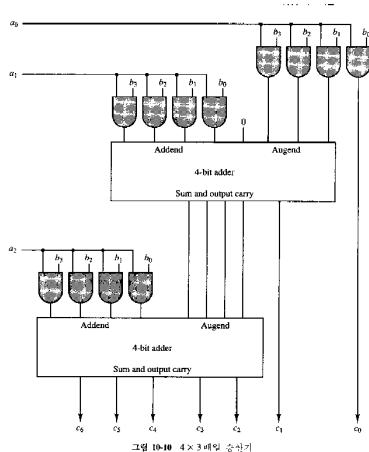


그림 10-10 4 × 3 배열 승산기

나눗셈 알고리즘

- 이진 나눗셈의 예 (그림 10-11)

Divisor: $B = 10001$	$\begin{array}{r} 11010 \\ \times 10001 \\ \hline 011100000 \end{array}$	Quotient = Q Dividend = A 5 bits of $A < B$, quotient has 5 bits 6 bits of $A \geq B$ Shift right B and subtract; enter 1 in Q -10001 ----- -010110 ----- -10001 ----- -001010 ----- -010100 ----- -10001 ----- -000110 ----- -00110 ----- Final remainder
-------------------------	--	--

그림 10-11 이진 나눗셈의 예

Slide 16

- 부호 절대값 데이터에 대한 하드웨어 구성
 - 하드웨어 구성은 그림 10-5의 곱셈기와 동일
 - EAQ에 피▣수를 넣고 B에 접수를 넣는다
 - A - B 를 하여
 - * E 가 1이면 A > B 이므로 원쪽으로 shift 하며 Q_n 에 1을 삽입

나눗셈 알고리즘 (계속)

Slide 17

E 가 0이면 A < B 이므로 B를 다시 더하고 원쪽으로 shift 하며 Q_n 에 0을 삽입

- 접수가 5비트 피▣수가 10비트 이므로 5번 수행후 끝
- A에는 나머지 Q에는 뭇이 있다
- 나눗셈 계산 예 (그림 10-12)

나눗셈 알고리즘 (계속)

Slide 18

Divisor $B = 10001$,		$\bar{B} + 1 = 01111$		
	<u>E</u>	<u>A</u>	<u>Q</u>	<u>SC</u>
Dividend:		01110	00000	
shl $E \bar{A} Q$	0	11100	00000	5
add $\bar{B} + 1$		01111		
$E = 1$	1	01011		
Set $Q_n = 1$	1	01011	00001	4
shl $E \bar{A} Q$	0	10110	00010	
Add $\bar{B} + 1$		01111		
$E = 1$	1	00101		
Set $Q_n = 1$	1	00101	00011	3
shl $E \bar{A} Q$	0	01010	00110	
Add $\bar{B} + 1$		01111		
$E = 0$; leave $Q_n = 0$	0	11001	00110	
Add \bar{B}		10001		
Restore remainder	1	01010		2
shl $E \bar{A} Q$	0	10100	01100	
Add $\bar{B} + 1$		01111		
$E = 1$	1	00011		
Set $Q_n = 1$	1	00011	01101	1
shl $E \bar{A} Q$	0	00110	11010	
Add $\bar{B} + 1$		01111		
$E = 0$; leave $Q_n = 0$	0	10101	11010	
Add \bar{B}		10001		
Restore remainder	1	00110	11010	0
Neglect E		00110		
Remainder in A :				
Quotient in Q :				11010

그림 10-12 나눗셈 알고리즘의 계산 과정을 보여주는 예

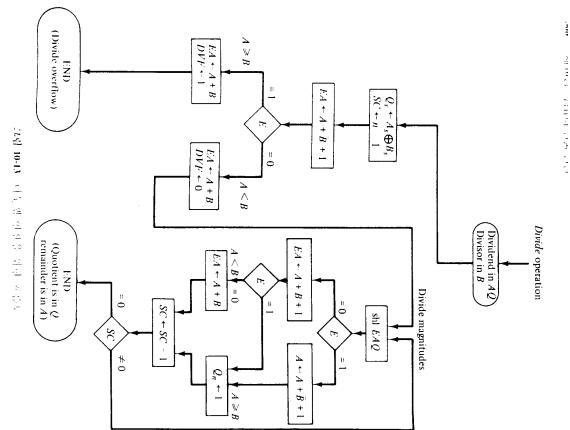
나눗셈 알고리즘 (계속)

Slide 19

- 나눗셈 오버플로우
 - 둑이 5비트 보다 클때 나눗셈 오버플로우 발생
 - 피겟수의 상위 5비트가 젯수보다 크거나 같을때 발생 (참조 그림 10-11)
 - overflow 가 발생하면 DVF (divide overflow) 플립플롭이 set 됨
 - 0 으로 나누는것도 overflow 발생
 - 보통 인터럽트가 걸리고 인터럽트 서비스루틴에서 해결하거나 프로그래머가 해결
 - 부동 소수점에서는 해결이 됨
- 하드웨어 알고리즘
 - 하드웨어 알고리즘 흐름도 (그림 10-13)

나눗셈 알고리즘 (계속)

Slide 20



- 다른 알고리즘
 - * 그림 10-13 의 알고리즘은 restoring 방법이라함 $\rightarrow A < B$ 일때 다시 더함

나눗셈 알고리즘 (계속)

Slide 21

- * 다른 방법으로서 비교방법과 nonrestoring 방법이 있음
 - * 비교 방법 : A 와 B 를 비교하여 $A > B$ 이면 빼고 $A < B$ 이면 원쪽 shift
 - * nonrestoring 방법 : $A < B$ 일때 B를 다시더하지 않고 원쪽 shift 후 B를 더함
 - restoring 방법 : $2(A - B + B) - B = 2A - B$
 - nonrestoring 방법 : $2(A - B) + B = 2A - B$

부동 소수점 산술 연산

- 부동 소수점수의 형식 : $m \times r^e$ (m 은가수 e 는 지수 r 은 밑수)

- 537.25 의 저장 $\rightarrow m = 53725, e = 3 \rightarrow 0.53725 \times 10^3$ 으로 해석

- 정규화 된 표현 가수의 최좌단 비트가 0이 아닌 숫자가 오게하는것
→ 가능한 많은 유효숫자를 표현할수 있음

- 48비트 워드에서

Slide 22

- 고정 소수점 표현 $\rightarrow + - 10^{14}$ 표현

- 부동 소수점 표현 (36비트 가수 16비트 지수로 표현) $\rightarrow + - (1 - 2^{-35}) \times 2^{2047}$

- 두수의 연산

- 두수의 지수를 일치시킴 (지수가 작을 수의 가수를 오른쪽으로 shift 하며 지수를 옮겨줌)

- overflow 해결

- overflow 시 가수를 shift 하며 지수를 조정하여 쉽게 해결

부동 소수점 산술 연산 (계속)

- 바이어스된 지수 표현 (biased exponent)

- 모든 지수를 양수로 표현

- 예) 지수가 -50 에서 +49를 갖을수 있다면

* +50을 지수에 더해서 연산, 연산후 조정

* 장점 : 지수비교 및 가수 위치조정시 편리

Slide 23

- 레지스터 구성

- 구성 (그림 10-14)

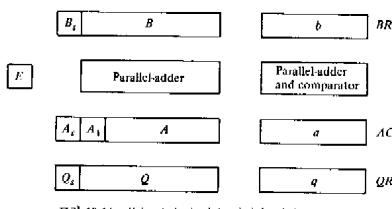


그림 10-14 부동 소수점 산술 연산을 위한 레지스터

부동 소수점 산술 연산 (계속)

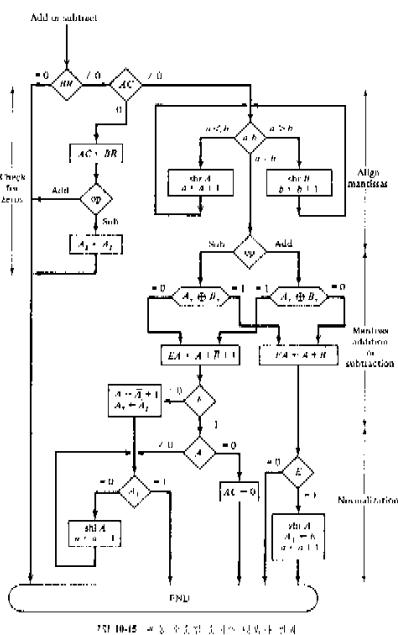
- 대문자 레지스터 →가수, 소문자 레지스터 →지수
 - 바이어스된 지수표현 가정

Slide 24

- 두수는 각각 AC 와 BR 에저장, 결과는 AC 에 저장됨
 - 연산 순서
 1. 0인지 여부를 조사
 2. 가수의 위치 조정
 3. 가수의 덧셈 또는 뺄셈
 4. 결과를 정규화
 - 흐름도 (그림 10-15)

부동 소수점 산술 연산 (계속)

Slide 25



부동 소수점 산술 연산 (계속)

- 곱셈

- 가수를 곱하고 지수를 더함

Slide 26

- 연산 순서
 1. 0인지 여부 조사
 2. 지수를 더함
 3. 가수를 곱함
 4. 결과를 정규화
- 흐름도 (그림 10-16)

부동 소수점 산술 연산 (계속)

Slide 27

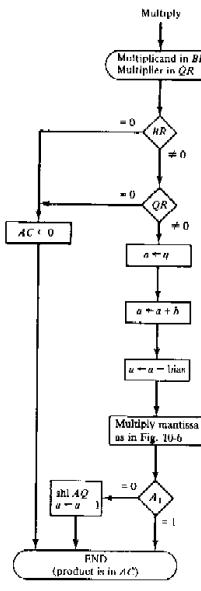
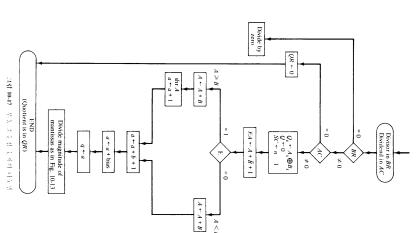


그림 10-16 부동 소수점 숫자의 곱셈

부동 소수점 산술 연산 (계속)

- 나눗셈
 - 지수의 뺄셈과 가수의 나눗셈으로 계산
 - 연산 순서
 1. 0인지 여부 조사
 2. 레지스터 초기화 및 부호 결정
 3. 피젯수의 위치 조정
 4. 지수의 뺄셈
 5. 가수의 나눗셈
 - 흐름도 (그림 10-17)

Slide 28



Slide 29

십진 산술 장치

- BCD 가산기

- 두 개의 십진수를 더하는 가산기: 9개의 입력 5개의 출력이 필요
- 논리설계방법으로는 너무 복잡

- 두 개의 4bit 2진가산기를 이용

- * 두 개의 한자리 십진수 A,B 를 더함

---십진수--- ---이진수---

Slide 30

$$\begin{array}{r}
 & \text{Ci} & & \text{Ci} \\
 & \text{A} & & \text{A}_4 \text{ A}_3 \text{ A}_2 \text{ A}_1 \\
 + \text{ B} & & + \text{ B}_4 \text{ B}_3 \text{ B}_2 \text{ B}_1 \\
 \hline
 & \text{-----} & & \text{-----} \\
 & \text{Ci+1 S} & & \text{Ci+1 S}_4 \text{ S}_3 \text{ S}_2 \text{ S}_1
 \end{array}$$

- * 두 개의 십진수를 더했을 때 최대크기 : $9 + 9 + 1$ (carry) = 19

- * 설계방침

- 두 개의 십진수를 더한 결과가 9보다 작으면 그대로 출력, carry 없음
- 두 개의 십진수를 더한 결과가 9보다 크면 10진수로 바꿈 : 6 (0110) 을 더함, carry 있음

십진 산술 장치 (계속)

Slide 31

- 결국 carry 가 있으면 0110 을 더함, 없으면 그대로 출력
- 진리표 : 표 5-1

십진 산술 장치 (계속)

Slide 32

2진수로 표현된 합					BCD 합				10 진수	
K	Z ₈	Z ₄	Z ₂	Z ₁	C	S ₈	S ₄	S ₂	S ₁	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

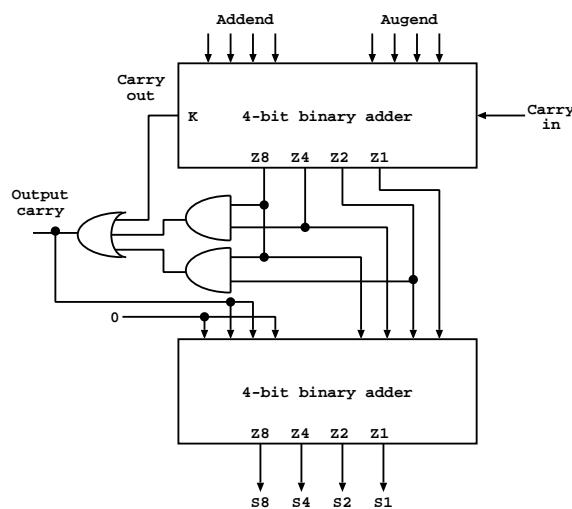
십진 산술 장치 (계속)

Slide 33

- 진리표에서 carry 가 발생하는 조건을 구함

$$\rightarrow C_{i+1} = K + Z_8Z_4 + Z_8Z_2$$

- 구현 그림 : 그림 10-18



십진 산술 장치 (계속)

- BCD 감산

Slide 34

- BCD 감산기를 이용하거나 BCD 가산기를 이용
- 9나 10의 보수를 이용 BCD 가산을 하는것이 효율적
- BCD 가, 감산기의 회로 (그림 10-19)

십진 산술 장치 (계속)

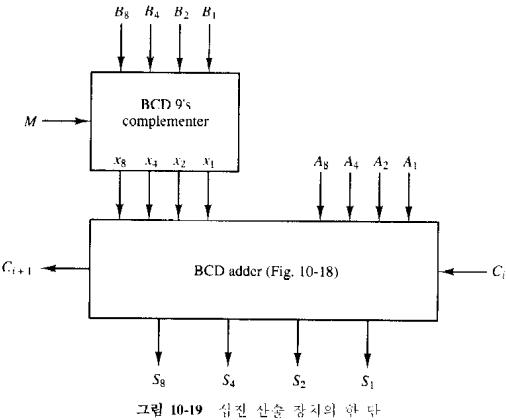
Slide 35

그림 10-19 십진 산술 장치와 하단

- 9의 보수를 만드는 조합회로

십진 산술 장치 (계속)

Slide 36

$$\begin{aligned}x_1 &= B_1 M' + B'_1 M \\x_2 &= B_2 \\x_4 &= B_4 M' + (B'_4 B_2 + B_4 B'_2) M \\x_8 &= B_8 M' + B'_8 B'_4 B'_2 M\end{aligned}$$

십진 산술 연산

- 십진 산술 마이크로 연산 기호 (표 10-5)

Slide 37

10.7 십진 산술 연산 → 321

표 10-5 십진 산술 마이크로 연산 기호

Symbolic Designation	Description
$A \leftarrow A + B$	Add decimal numbers and transfer sum into A
\bar{B}	9's complement of B
$A \leftarrow A + \bar{B} + 1$	Content of A plus 10's complement of B into A
$Q_t \leftarrow Q_t + 1$	Increment BCD number in Q_t
dshr A	Decimal shift-right register A
dshl A	Decimal shift-left register A

본의상, 십진 산술 연산에서도 이진 산술 연산과 같은 기호를 쓰나 약간만 해석을 달리한다. 즉 표 10-5에서 레지스터 문자 기호 위의 bar(−)는 레지스터에 저장된 수의 9의 보수를 말하며, 9의 보수에 1을 더한 경우 10의 보수를.

- shift 연산
 - 한개의 BCD 단위로 shift
 - 예) 레지스터 A에 7860이 저장되어 있다

십진 산술 연산 (계속)

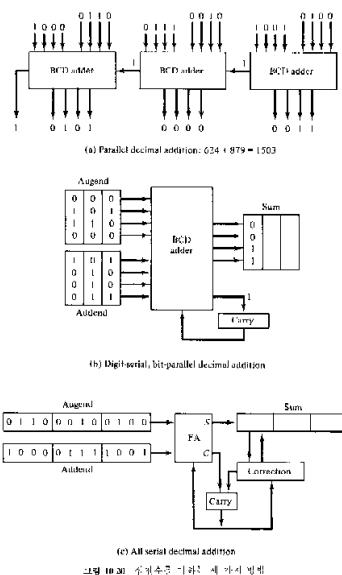
Slide 38

* dshr A 의 결과는 0786 이된다

- 덧셈과 뺄셈
 - 십진수의 덧셈의 세가지 구현 방법 (그림 10-20)

십진 산술 연산 (계속)

Slide 39



십진 산술 연산 (계속)

- 곱셈

- 십진 산술 연산을 위한 레지스터 구성도 (그림 10-21)

한 뒤 그 값을 부분곱에 더한다. 이 과정은 승수의 숫자의 값만큼 반복하여 외승수를 부분곱에 더함으로써 이루어진다.

그림 10-21에 십진 곱셈을 위한 레지스터 구성도가 나타나 있다. A , B , Q 의 세 레지스터가 있으며 각각 A_s , B_s , Q_s 의 부호를 위한 플립플롭이 있다. 레지스터 A 와 B 는 A_s , B_s 로 표시되는 네 개의 비트를 더 갖고 있는데, 이는 레지스터 Q 에 있는 4비트 부호를 더 갖고 있는 것과 같다.

Slide 40

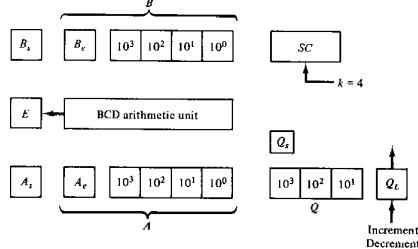


그림 10-21 십진 산술 곱셈과 나눗셈을 위한 레지스터들

십진 산술 연산 (계속)

- 흐름도 (그림 10-22)

Slide 41

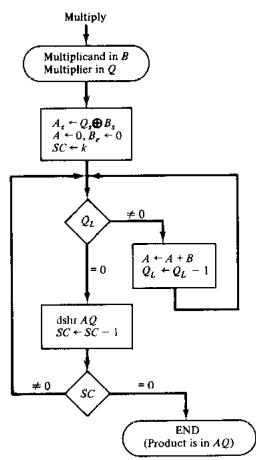


그림 10-22 십진 곱셈을 위한 흐름도

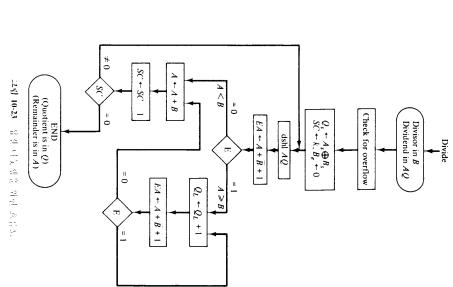
※※※ 자리수를 하나 더 확장시켜준다. RCD 가산 장치는 다음 자리의 수자들

십진 산술 연산 (계속)

- 나눗셈

- 흐름도 (그림 10-23)

Slide 42



Homework

- 9장 연습문제

Slide 43

- 10장 연습문제