

7강_Integer Arithmetic

학습목표

- » ALU에서 정수형 상수의 곱셈 방식을 이해하고 부동 소수점 수의 표현 방법을 설명할 수 있다.
- » 부동 소수점 연산 방법에 대해 설명할 수 있다.

학습내용

- » Integer Arithmetic
- » BFPN Representation
- » BFPN Arithmetic
- » Quiz, PBL, 탐구주제

Unsigned Multiplication

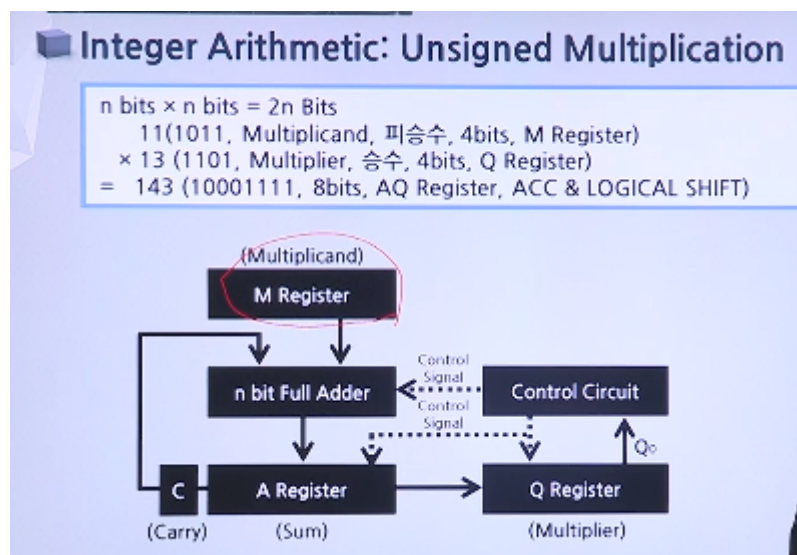
Integer Arithmetic: Unsigned Multiplication

$n \text{ bits} \times n \text{ bits} = 2n \text{ bits}$
11(1011, Multiplicand, 피승수, 4bits, M Register)
 $\times 13$ (1101, Multiplier, 승수, 4bits, Q Register)
= 143 (10001111, 8bits, AQ Register, ACC & LOGICAL SHIFT)

multiplicand 피승수, multiplier 승수

				(M Reg) 1011	
x	1011	Carry Bit	A Reg. 0000	Q Reg. 1101	
	1011		0000	1101	초기 값
			1011	1101	더할 값
			1011	1101	더한 값
			0101	1110	Logical Shift Right
	0000		0101	1110	초기 값
			0000	1110	더할 값
			0101	1110	더한 값
			0010	1111	Logical Shift Right
	1011		0010	1111	초기 값
			1011	1111	더할 값
			1101	1111	더한 값
			0110	1111	Logical Shift Right
	1011		0110	1111	초기 값
			1011	1111	더할 값
			0001	1111	더한 값
			1000	1111	Logical Shift Right
	1000 1111		1000	1111	

위 과정을 하드웨어에서



피승수는 M 레지스터에 저장됨. 상수가 계속 저장됨.

필요할 때마다 n 비트짜리 전가산기로 값을 내주기로 되어있음

A레지스터 초기에는 0000이 들어가 있음. 그 다음 Q 레지스터의 초기 값은 승수 (Multiplier) 1101 들어가있음

Q레지스터의 LSB를 Q0이라고 표현했음. 이게 1인가 0인가?

Q0을 보고 Control Signal을 줘서 n bit Full Adder에서 적절한 동작을 수행할 수 있도록 하면 된다

→ Control Circuit를 통해서 신호를 내주는데, Q가 0이라고 한다면 이 M 레지스터에 있는 값을 0000으로 셋팅 해서 내려야함

만약 Q0가 1이면, M 레지스터 값에 1011을 그대로 내려야함.

위는 부호가 없는 것

아래는 부호가 있는 경우

Integer Arithmetic: Signed Multiplication(Booth's Algorithm)

임의의 Binary #를 Sign에 관계없이 2의 제곱의 합이나 차로 표현해보자!

$$0010 = 2 = 2^2 - 2^1$$

$$+ 0001 = 1 = 2^1 - 2^0$$

$$0011 = 3 = 2^2 - 2^0$$


$2^2 + 2^0$

-7 (1001, Multiplicand, 피승수, 4bits, M Register)
 × 3 (0011, Multiplier, 승수, 4bits, Q Register)
 = -21 (11101011, 8bits, AQ Register, (ADD/SUB & ARITH SHIFT))

$1001 \times 0011 = (0100 - 0001)$


이를 기계적으로 수행하는 방법 $0011(01) \rightarrow$ 현재 자릿수 값과 다음 자릿수 값

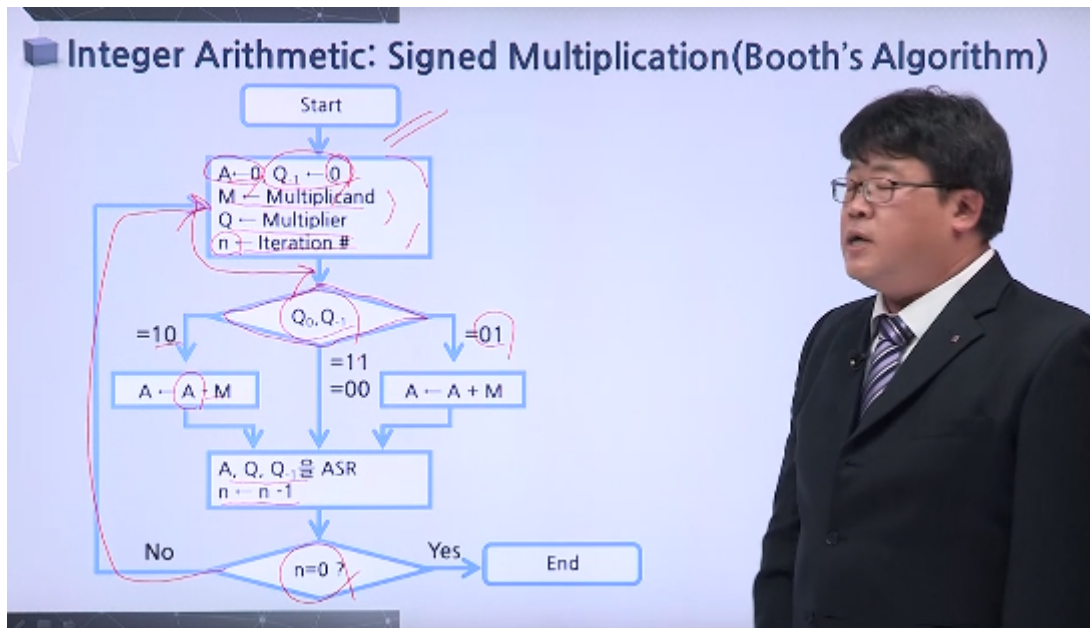
00이나 11이면 : 0
 01이면 : +
 10이면 : -



Integer Arithmetic: Signed Multiplication(Booth's Algorithm)

	x	Carry Bit	A Reg.	Q Reg.	M Reg. 1001
	1001	0	0000	0011	
	0000	0111	0000	0011	0 초기 값
			0111	0011	더할 값
			0411	0011	0 더할 값
			0011	1000	1 Arith. Shift Right
			0011	1000	1 초기 값
			0000	1100	더할 값
			0011	1100	1 더할 값
			0001	1100	1 Arith. Shift Right
			0001	1100	1 초기 값
			1001	1100	더할 값
			1010	1100	1 더할 값
			1001	0110	0 Arith. Shift Right
			1101	0110	0 초기 값
			0000	0110	0 더할 값
			1101	0110	더할 값
			1110	1011	0 Arith. Shift Right
			1110	1011	





A에는 0을 처음에 넣기, 그 다음에 Q-1에 0을 붙이기(맨 뒷부분에 붙이는 0 → M Register에는 multiplicand 피승수를 넣음, 그 다음에 Q에는 승수를 넣게 됨 → n이라는 어떤 변수에다가 iteration(반복횟수)가 몇번인지. 여기서는 4번(4비트의 4번. 8비트는 8번)

▼ 설명

1. 그런다음에 Q0와 Q-1이 두 개를 봅니다. 그게 10이냐 10이면 기존에 A값에서 빼라, 그다음에 01이면 A값의 M을 더하라, 그다음에 11이거나 00이면 아무 짓도 하지 마라. 0을 더하라는 이야기죠. 자, 더한다음에 A와 Q와 Q-1을 Arithmetic Shift Right를 하게 되고요. 그럼 자리수가 내려오죠. 그다음에 n을 하나 뺍니다. 빼면 횟수가 한번 돌아갔습니다. 그래서 n이 0이냐 확인해서 만약에 0이 맞다 그러면 4번을 다 수행하게 되니까 끝을 내면 되고요. n이 0이 아니라고 한다면 다시 이 위로 돌아가서요.
2. 작업을 반복을 해야되겠죠. 그죠? 반복할 때에는 뭐 A가 이거는 삭제가되겠죠. 빼고, 이거랑 이거는 빼고 다음, 이것도 사실은 중간에 변형이 됐기 때문에 A, Q, MQ 빼고 여기서 여기로 이렇게 연결되어야 겠네요. 이건 초기에 한번만 수행하는 것이고요 그 다음부터는 수행하지 않습니다.