

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

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)

		M Reg. 1011			
X	1 0 1 1 1 1 0 1	Carry Bit	A Reg. 0000	Q Reg. 1101	
	1 0 1 1		0000 1011 1011 0101	1101 1101 1101 1110	초기 값 더할 값 더한 값 Logical Shift Right
	0 0 0 0		0101 0000 0101 0010	1110 1110 1110 1111	초기 값 더할 값 더한 값 Logical Shift Right
	1 0 1 1		0010 1011 1101 0110	1111 1111 1111 1111	초기 값 더할 값 더한 값 Logical Shift Right
	1 0 1 1		0110 1011 0001 1000	1111 1111 1111 1111	초기 값 더할 값 더한 값 Logical Shift Right
	1 0 0 0 1 1 1 1	1	1000	1111	

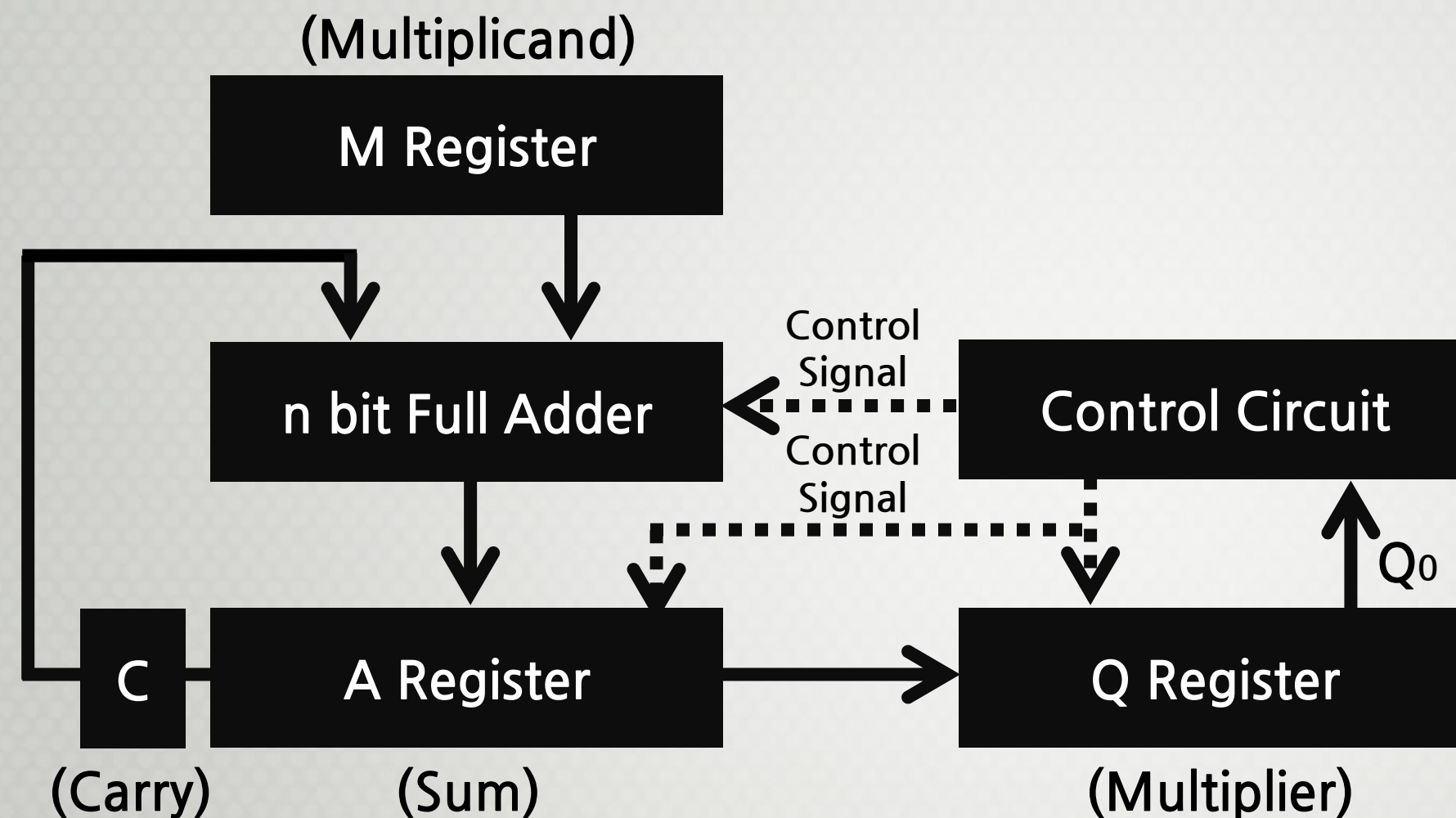
Integer Arithmetic: Unsigned Multiplication

$n \text{ bits} \times n \text{ bits} = 2n \text{ Bits}$

11 (1011, Multiplicand, 피승수, 4bits, M Register)

\times 13 (1101, Multiplier, 승수, 4bits, Q Register)

= 143 (10001111, 8bits, AQ Register, ACC & LOGICAL SHIFT)



Integer Arithmetic: Signed Multiplication(Booth's Algorithm)

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

$$\begin{array}{r} 0010 = 2 = 2^2 - 2^1 \\ + 0001 = 1 = 2^1 - 2^0 \\ \hline 0011 = 3 = 2^2 - 2^0 \end{array}$$

이를 기계적으로 수행하는
방법 0011(0) → 현재 자릿수
값과 다음 자릿수 값

00이나 11이면	: 0
01이면	: +
10이면	: -

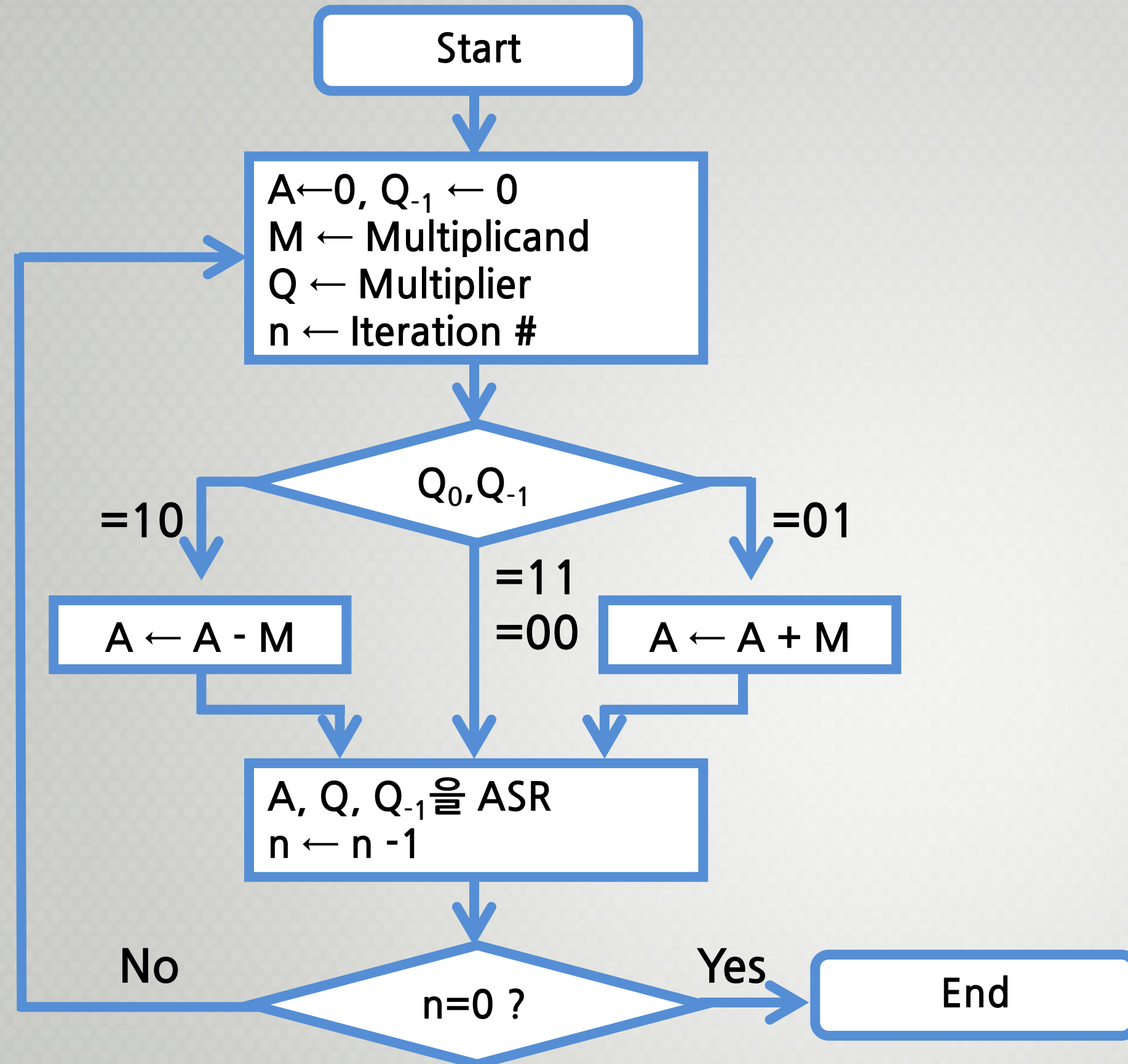
-7 (1001, Multiplicand, 피승수, 4bits, M Register)
× 3 (0011, Multiplier, 승수, 4bits, Q Register)
= -21 (11101011, 8bits, AQ Register, ADD/SUB & ARITH SHIFT)

1001	1001
× 0011	× (0100 - 0001)

Integer Arithmetic: Signed Multiplication(Booth's Algorithm)

		Carry Bit	A Reg.	Q Reg.	M Reg. 1001
X	1001 0011		0000	0011	
		0	0000 0111 0111 0011	0011 ← 0011 1001	0 초기 값 더할 값 0 더한 값 1 Arith. Shift Right
0000	0111		0011 0000 0011 0001	1001 ← 1100 1100	1 초기 값 더할 값 1 더한 값 1 Arith. Shift Right
		-	0001 1001 1010 1101	1100 ← 1100 0110	1 초기 값 더할 값 1 더한 값 0 Arith. Shift Right
1110	0100		1101 0000 1101 1110	0110 ← 0110 1011	0 초기 값 0 더할 값 더한 값 0 Arith. Shift Right
0000	0000		1110	1011	
1110	1011				

Integer Arithmetic: Signed Multiplication(Booth's Algorithm)



Integer Arithmetic: Signed Multiplication(Booth's Algorithm)

