

4강_ALU Architecture과 Integer Representation

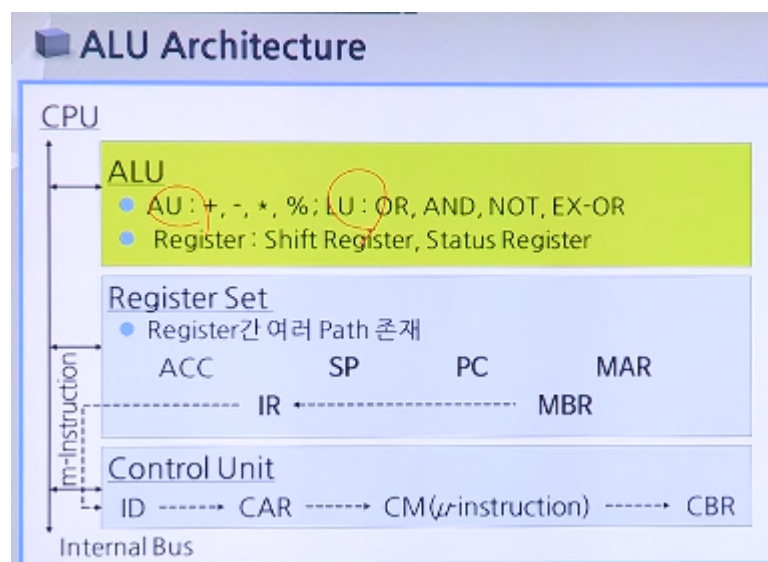
학습목표

- » ALU에 대한 구조를 이해하고 ALU에서 처리되는 정수형 상수에 대한 표현법을 이해할 수 있다.
- » 논리 연산에 대한 처리 방법을 설명할 수 있다.
- » 정수형 상수의 덧셈기 및 뺄셈기에 대해 공부함으로써 ALU에 대한 기초를 확립할 수 있다.

학습내용

- » ALU Architecture
- » Integer Representation
- » Logic Operations
- » Integer Arithmetic(+/-)
- » Quiz, PBL, 탐구주제

ALU는 산술논리 연산



Integer Representation: Unsigned

Decimal	Unsigned Binary
7	111
6	110
5	101
4	100
3	011
2	010
1	001
0	000

- n Bit 조합에서 의미있는 조합의 개수 : 2^n
- n Bit 일 때 표현 가능 범위 : $0 \sim 2^n - 1$

• Binary # $a_2a_1a_0$ 를 Decimal # d로 바꾸는 일반 식 :

$$d = \sum_{i=0}^{n-1} a_i 2^i$$

$110_2 \rightarrow 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 6_{10}$

• Bit Extension : 0을 추가 $110_2(6_{10}) \rightarrow 0110_2(6_{10})$

unsigned integer : 부호없는 정수

이진수, 십진수

음수표현

Integer Representation: Signed Magnitude

Decimal	Signed Magnitude
3	011
2	010
1	001
0	000
0	100
-1	101
-2	110
-3	111

MSB 최상위 비트 LSB 최하위 비트

- n Bit 조합에서 의미있는 조합의 개수 : $2^n - 1 \rightarrow 0$ 중복
- n Bit 일 때 표현 가능 범위 : $-2^{n-1} + 1 \sim 2^{n-1} - 1$
- Binary # $a_2 a_1 a_0$ 를 Decimal # d로 바꾸는 일반식 : $d = (-1)^{a_{i-1}} \times \sum_{i=0}^{n-2} a_i \cdot 2^i$
- Bit Extension : Sign Bit 다음에 0을 추가

$110_2 \rightarrow (-1)^1 (1 \times 2^1 + 0 \times 2^0) = -2_{10}$

$110_2 (-2_{10}) \rightarrow 1010_2 (-2_{10}), 010_2 (2_{10}) \rightarrow 0010_2 (2_{10})$

보수 얘기

Integer Representation: 1's Complement

Decimal	1's Complement
3	011
2	010
1	001
0	000
0	111
-1	110
-2	101
-3	100

- n Bit 조합에서 의미있는 조합의 개수 : $2^n - 1 \rightarrow 0$ 중복
- n Bit 일 때 표현 가능 범위 : $-2^{n-1} + 1 \sim 2^{n-1} - 1$
- 음수를 표현하는 방법 : 1의 보수를 취함

$2_{10} \rightarrow 010_2, -2_{10} \rightarrow 101_2$

- Decimal #로 바꾸는 방법

Sign Bit가 0이면 → 그대로 Decimal #로 변환(+)

Sign Bit가 1이면 → 1의 보수를 취하고 Decimal #로 변환(-)

2의 보수

Integer Representation: 2's Complement

Decimal	2's Complement
3	011
2	010
1	001
0	000
-1	111
-2	110
-3	101
-4	100

Decimal #로 바꾸는 방법

Sign Bit가 0이면 → 그대로 Decimal #로 변환(+)

Sign Bit가 1이면 → 2의 보수를 취하고 Decimal #로 변환(-)

Bit Extension : Sign Bit를 추가

$110_2(-2_{10}) \rightarrow 1110_2(-2_{10}), 010_2(2_{10}) \rightarrow 0010_2(2_{10})$

Overflow/Underflow 고찰

n Bit 조합에서 의미있는 조합의 개수 : 2^n

n Bit 일 때 표현 가능 범위 : $-2^{n-1} \sim 2^{n-1}-1$

Bit Extension : Sign Bit를 추가

$110_2(-2_{10}) \rightarrow 1110_2(-2_{10}), 010_2(2_{10}) \rightarrow 0010_2(2_{10})$