

주제/단락	내용
학습목표	ALU 내부 구성 이해, 논리 연산 원리 이해, 정수 및 부동소수점 수의 표현과 산술 연산을 이해한다.
학습내용 개요	ALU 구성 요소, 정수 표현, 논리 연산, 시프트 연산, 정수 산술 연산, 부동소수점 표현, 부동소수점 산술 연산을 학습한다.
ALU 정의	CPU 내부에서 산술 연산과 논리 연산을 수행하는 핵심 하드웨어 모듈이다.
ALU 구성요소: 산술부	덧셈, 뺄셈, 곱셈, 나눗셈 등 산술 연산을 수행한다.
ALU 구성요소: 논리부	AND, OR, XOR, NOT 등 논리 연산을 수행한다.
ALU 구성요소: 시프트 레지스터	비트를 좌우로 이동시키는 기능을 제공한다.
ALU 구성요소: 보수기	2진 데이터를 2의 보수로 변환하여 음수화를 수행한다.
ALU 구성요소: 상태 레지스터	연산 결과 상태를 나타내는 플래그들을 저장한다.
2진수 체계	0과 1, 부호 및 소수점으로 수를 표현하며 예로 -13.625는 -1101.101로 나타낸다.
부호 없는 정수 예	8비트에서 57=00111001, 0=00000000, 1=00000001, 128=10000000, 255=11111111로 표현된다.
부호 없는 정수 변환식	n비트 2진수의 값 A는 각 비트 가중치의 합으로 계산하며 예로 110 ₂ 는 6이다.
부호 없는 정수 범위	n비트로 표현 가능한 범위는 0부터 2 ⁿ -1까지이다.
진법 변환 표	0~15에 대한 2진수, 8진수, 16진수 대응을 통해 서로 변환한다.
2진 소수 변환법	소수점 왼쪽이 MSB인 2진 소수는 음의 지수 가중치의 합으로 계산하며 예로 1101.101은 13.625이다.
음수 표현 방법	최상위 비트를 부호 비트로 사용하며 부호화-크기, 1의 보수, 2의 보수 방식이 있다.
부호화-크기 표현 정의	MSB는 부호, 나머지 비트는 크기를 나타내며 예로 +9는 0 0001001, -9는 1 0001001이다.
부호화-크기 10진 변환	부호에 따라 크기 부분을 가중치 합으로 계산해 부호를 적용한다.
부호화-크기 장점	음수 표현 방식 중 구조가 가장 단순하다.
부호화-크기 단점	덧셈·뺄셈 시 부호와 크기를 분리 처리해야 하고 +0과 -0이 공존하여 표현 가능한 수가 2 ⁿ -1개로 줄어든다.
보수 표현 개념	음수를 보수로 표현하는 방식으로 1의 보수와 2의 보수가 있다.

주제/단락	내용
1의 보수 정의	모든 비트를 반전하여 구한다.
2의 보수 정의	모든 비트를 반전한 뒤 1을 더해 구한다.
보수 표현 예시	+9=00001001, -9의 1의 보수=11110110, 2의 보수=11110111; +35=00100011, -35의 1의 보수=11011100, 2의 보수=11011101이다.
1의 보수 범위	8비트에서 $-(2^7-1)$ 부터 $+(2^7-1)$ 까지이며 일반화하면 $-(2^{n-1}-1)$ 부터 $+(2^{n-1}-1)$ 까지이다.
2의 보수 범위	8비트에서 -2^7 부터 2^7-1 까지이며 일반화하면 -2^{n-1} 부터 $2^{n-1}-1$ 까지이다.
2의 보수 양수 변환	MSB가 0이면 가중치 합으로 10진수로 변환한다.
2의 보수 음수 변환	MSB가 1이면 -2^{n-1} 에 나머지 가중치 합을 더하거나 비트 반전 후 1을 더해 크기를 구해 음수로 표시한다.
2의 보수 변환 예	10101110은 $-128+(32+8+4+2)=-82$ 이며 반전해 01010010을 해석한 뒤 음수 부호를 붙여도 -82가 된다.
비트 확장 개념	더 긴 레지스터 저장이나 연산을 위해 비트 수를 늘리는 방법이다.
부호화-크기 비트 확장	부호 비트를 새 MSB로 옮기고 나머지는 0으로 채운다.
부호화-크기 확장 예	+21: 8비트 00010101→16비트 0000000000010101, -21: 8비트 10010101→16비트 1000000000010101이다.
2의 보수 비트 확장	상위 비트를 부호 비트와 동일하게 채우는 부호 비트 확장을 사용한다.
2의 보수 확장 예	+21: 00010101→0000000000010101, -21: 11101011→1111111111101011이다.
기본 논리 연산	AND, OR, XOR, NOT의 진리표에 따라 동작한다.
멀티플렉서 정의	여러 입력 중 선택선 값에 따라 하나를 출력으로 전달하는 조합논리회로이다.
디멀티플렉서 정의	하나의 입력을 2^n 개의 가능한 출력 중 하나로 분배하는 회로이다.
논리 하드웨어 구성	입력 비트가 모든 게이트를 통과하며 4×1 MUX는 선택선에 따라 네 연산 중 하나를 출력으로 보낸다.
N비트 논리 연산장치	기본 논리 모듈을 병렬 접속해 다비트 데이터에 대한 연산을 구현한다.
부분 비트 연산	데이터 단어의 일부 비트만 선택해 값 변경에 활용될 수 있다.
AND 연산 예	두 데이터의 대응 비트를 AND하여 결과를 생성한다.
OR 연산 예	두 데이터의 대응 비트를 OR하여 결과를 생성한다.
XOR 연산 예	두 데이터의 대응 비트를 XOR하여 결과를 생성한다.
NOT 연산 예	데이터의 모든 비트를 반전시킨다.
응용 논리연산 개념	다른 데이터와의 연산을 통해 원래 데이터의 특정 비트를 세트, 반전, 클리어, 삽입, 비교한다.

주제/단락	내용
선택적-세트 연산	B의 해당 위치에 1을 세트하고 A OR B로 A의 일부 비트를 1로 만든다.
선택적-세트 예	A=10010010에서 하위 4비트를 B=00001111과 OR하여 A=10011111로 만든다.
선택적-보수 연산	B의 1인 위치에 대해 A XOR B로 A의 해당 비트를 반전시킨다.
선택적-보수 예	A=10010101, B=00001111이면 결과는 A=10011010이 된다.
마스크 연산 정의	B의 0 위치에 대응하는 A의 비트를 0으로 만들기 위해 A AND B를 사용한다.
마스크 연산 예	A=11010101, B=00011111이면 결과는 A=00010101이다.
삽입 연산 절차	먼저 마스크로 대상 위치를 0으로 클리어한 뒤 새 비트 패턴과 OR하여 대체한다.
삽입 연산 예	A=10010101에서 상위 4비트를 1110으로 바꾸기 위해 AND 후 OR하여 최종 A=11100101을 얻는다.
비교 연산 정의	A와 B를 XOR해 같은 비트는 0, 다른 비트는 1로 세트하고 모두 같으면 Z 플래그를 1로 세트한다.
비교 연산 예	A=11010101, B=10010110이면 결과 A=01000011이며 모든 비트가 0이면 Z가 1이 된다.