10주차 예비보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20231609 이름: 정희선

**1.**

4-bit adder는 이진 병렬 덧셈을 수행하는 회로이다. 이 회로는 4개의 full adder가 직렬 형태로 연결되어 있는 구조이다. 4비트의 입력 A, B를 각 자리 비트마다 더해서 결과를 얻는다. 각 full adder에서 입력 비트, 이전 자리의 carry를 받아서 해당 자리의 sum과 새로운 carry를 출력한다. 최종 출력은 4 비트 덧셈 계산 결과와 최상위 carry이다. 4-bit subtractor는 이진 병렬 뺄셈을 수행하는 회로이다. 덧셈과 유사하게 4개의 full adder를 사용한다. 하지만 2’s complement를 사용하여 뺄셈 연산을 수행하는 원리이다.

**2.**

Look ahead carry는 carry를 빠른 속도로 계산하기 위하여 사용되는 방법이다. 일반적인 방식을 ripple carry 방식이라고 부르는데, 이 경우 carry가 각 자리로 전파되므로 비트 수가 많아질 때 연산 속도가 느려진다. 이러한 단점을 개선하는 방법인 look ahead carry 방법은 carry를 미리 계산하여 동시에 전달한다. 이 방식은 generate와 propagate 신호를 사용하여 각 비트의 carry를 예측한다. Generate는 carry의 발생을, propagate는 carry의 전파를 의미한다.

**3.**

2’s complement의 덧셈은 양수와 음수를 모두 동일한 덧셈 회로로 처리한다. 예를 들어A+B가 양수인 경우 그대로 더해주면 되고, 음수인 경우에도 마찬가지이다. 그 이유는 음수인 경우에도 2’s complement로 변환된 결과가 자동으로 carry를 포함한 결과를 제공하기 때문이다. 따라서 이 때 carry를 무시하고 하위 비트만 결과로 얻는다. 2’s complement의 뺄셈의 경우 뺄셈을 덧셈으로 변환하는 방식을 적용한다. 먼저 뺄셈을 덧셈으로 변환한다. 예를 들자면 A-B의 경우 A+(-B)로 바꾸어 생각한다. 다음으로 XOR을 통해 B의 각 비트를 반전한다. 이를 통해 B는 1의 보수 형태로 바뀐다. 마지막으로 B에 1을 더하면 B는 2’s complement인 -B가 된다.

**4.**

BCD 연산은 binary coded decimal의 약자로, 이진수로 표현된 십진수를 다루는 연산 방식을 말한다. 일반적인 이진수 연산과의 차이점은 각 십진수 자릿수를 4비트의 이진수로 표현한다는 것이다. 따라서 BCD 값이 십진수 범위를 넘지 않도록 추가적인 조정이 필요하다. BCD 덧셈 시, 덧셈 결과가 십진수 9(1001)을 넘어간다면 6(0110)을 더하는 과정인 BCD 보정을 통해 십진수 값을 유지한다.

**5.**

ALU는 arithmetic logic unit으로, 컴퓨터의 CPU 내에서 산술과 논리 연산을 수행하는 아주 핵심적인 구성 요소이다. ALU는 산술 연산뿐만 아니라 AND/OR/XOR 등 다양한 논리 연산을 수행할 수 있다. 즉, ALU는 CPU 내부의 데이터 처리와 결과 저장의 역할을 하며, 연산 처리의 핵심적인 요소라고 할 수 있다. ALU의 주요 기능들을 몇가지로 나누어 생각해 볼 수 있다. 첫번째로 산술 연산은 덧셈/뺄셈/곱셈/나눗셈의 기본적인 수학 연산을 수행한다. 두번째로 논리 연산은 AND/OR/XOR/NOT 등의 논리 연산으로 비트 단위의 데이터를 다룬다. 세번째로 비교 연산은 두 값의 크기를 비교하여 조건부 분기 연산에서 필요한 결과를 제공해 줄 수 있다. 마지막으로 shift 연산은 비트를 왼쪽이나 오른쪽으로 이동시키는 연산을 통해 데이터를 곱하거나 나누는 연산을 빠르게 수행하는 것을 가능하도록 해준다.

**6.**

ALU는 컴퓨터 구조와 연산 처리와 깊은 관련이 있다. 이와 연관된 개념 몇 가지를 설명할 것이다.

1. Register

CPU 내부에서 데이터와 명령을 일시적으로 저장하는 고속 메모리이다. ALU는 register에 저장된 데이터를 사용하여 연산을 수행하고, 연산 결과 역시 register에 저장된다.

1. Pipelining

파이프라인은 여러 연산을 동시에 수행하여 CPU의 처리 속도를 높이는 기법이다. 파이프 라인 구조에서, 여러 명령어를 겹쳐 실행할 수 있도록 ALU와 각종 제어 장치들이 협력한다.

1. Memory hierarchy

CPU와 ALU의 연산 성능을 극대화하기 위해 사용하는 개념이다. 메모리 계층 구조는 CPU에 가까운 순서대로 저장 장치를 배열하여 데이터로의 접근 속도를 최적화한다.