10주차 예비보고서

전공: 컴퓨터공학과 학년: 3학년 학번: 20211547 이름: 신지원

**1.**

4-Bit Adder 4-bit Ripple Carry Adder 라고 불리며 올림수를 포함하여 연산하기 때문에 이진수 연산이 가능하다. 따라서 4-Bit 이진수의 덧셈을 연산하도록 만들어진 가산기다.

도표, 라인, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

위의 다이어그램처럼 연산에서 발생한 올림 수 (Carry) 가 발생하는 것을 볼 수 있다. 가장 오른쪽에 보이는 Cin 은 전 계산에서 올림수가 들어오지 않았기 때문에 0이 입력될 것이다.

위를 이용하여4-bit 의 덧셈, 뺄셈이 모두 가능한 4-Bit 병렬 가감산기를 만들 수 있다. 4-bit Ripple Carry Adder/Subtractor라고 불리며 기존의 가산기 회로에 XOR 연산과 마이너스 bit 값을 추가하여 구현할 수 있다. 덧셈의 경우에는 마이너스 비트가 값을 가지지 않기 때문에 그대로 빠져나오고 뺄셈의 경우에는 마이너스 비트에 값이 더해져 연산을 수행하게 된다.

도표, 스케치, 평면도, 기술 도면이(가) 표시된 사진

자동 생성된 설명

**2.**

Look ahead carry 는 자리올림수 예측 가산기라고 불리며 1번에서 알아본 Ripple Carry Adder 보다 더 속도가 빠르다. Ripple Carry Adder 는 Carry가 들어올 때까지 기다렸다가 연산을 수행하는 반면, Look ahead carry 는 carry 를 한 번에 계산하기 때문에 지연시간이 짧다.

텍스트, 도표, 라인, 폰트이(가) 표시된 사진

자동 생성된 설명

CLA 는 위처럼 P,G 를 look-ahead unit 에 전달하여 carry 를 한 번에 예측하는 방식을 취한다. 따라서 Ripple Carry Adder 와 달리 Carry 대신 P(Propagate)와 G(Generate)를 출력한다. 이를 확장시켜 16bit CLA 도 만들 수 있다.

텍스트, 폰트, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

폰트, 화이트, 텍스트, 타이포그래피이(가) 표시된 사진

자동 생성된 설명

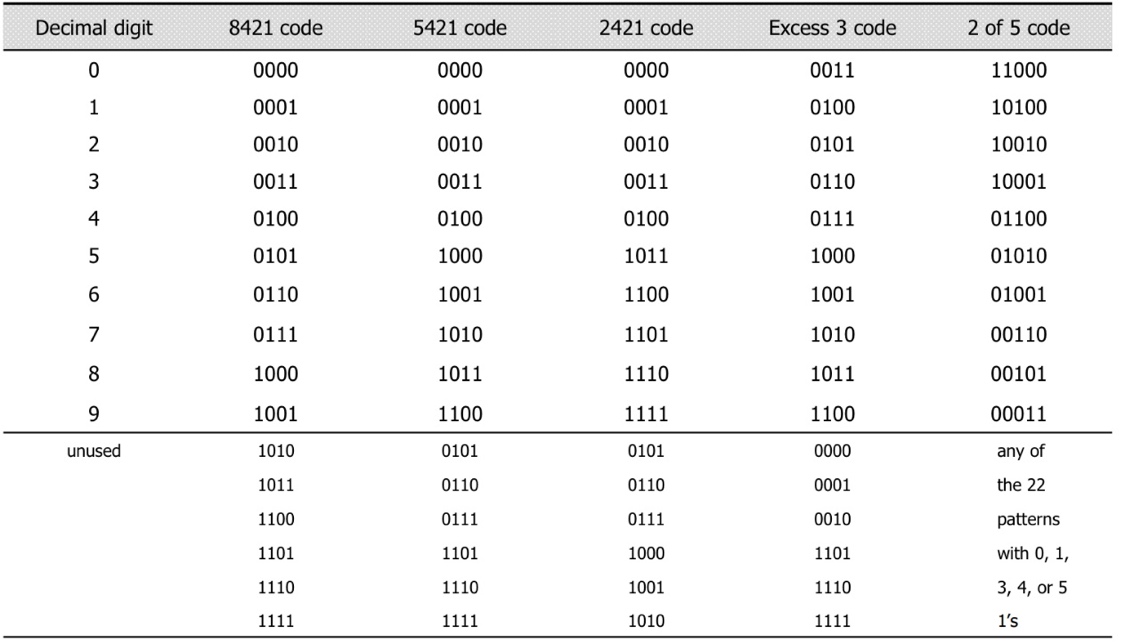
Carry 를 계산하는 방식은 위의 식처럼 계산하며 이렇게 얻어진 G와 P는 주 함수에서 S와 C를 얻는 수식으로 활용된다.

**3.**

덧셈과 뺄셈이 모두 가능한 가감산기를 구현하기 위해선 덧셈과 뺄셈의 연산에 따른 결과에 따라 다른 신호가 나타남을 생각해보아야 한다. 덧셈에서는 올림수가 발생할 것이고, 뺄셈에서는 내림수가 발생할 것이다. 2’s complement 가감산기에서 올림수는 carry bit, 내림수는 minus bit에 해당된다, S bit은 덧셈에 대해서는 0을 갖고, 뺄셈에 대해 1을 갖는다. 가감기에서는 carry를 입력할 때 S bit 를 입력하고, 각 자리의 입력과 S bit을 XOR 의 연산값을 Adder의 두번째 input으로 입력한다. S bit이 0이면 덧셈이며 B값은 그대로지만, S bit이 1이면 뺄셈이며 B값에 보수를 취하여 전달해야 한다.

**4.**

Binary-coded decimal 를 의미하는 BCD는 10진수의 덧셈을 위한 회로다. 우리는 생활 속에서 10진수를 사용하는 만큼 BCD는 가장 많이 쓰이는 회로 중 하나다. 10진수를 위한 회로기 때문에 BCD 가산기, 감산기는 기존 이진 덧셈, 뺄셈 회로에 추가적으로 회로를 구성해야 한다. 아래는 BCD의 십진화를 나타낸 표이다.



BCD 연산을 수행할 때, 두 이진 코드를 더한 값이 1001보다 크면 올림이 발생한다. 이때 6을 더하고 1을 carry 한다. 아래는 간단하게 나타낸 로직이다.

도표, 텍스트, 평면도, 기술 도면이(가) 표시된 사진

자동 생성된 설명

**5.**

arithmetic and logical unit 의 약자인 ALU 는 산술 논리 장치로 불린다. 덧셈, 뺄셈 같은 두 숫자의 산술 연산과 논리연산을 계산하는 회로다. 아주 작은 회로도 산술논리장치를 지니는데, 이처럼 대부분의 회로에서 산술논리장치가 포함된 것을 볼 수 있다. 간단한 연산에서 복잡한 연산까지 어떠한 연산도 모두 설계 가능하다. 연산이 복잡해질수록, 더 복잡한 산술논리장치를 설계해야 하고 그에 따라 실용성이 떨어진다. 실제 연산을 담당하는 하드웨어 모듈인만큼 다양한 기능을 수행할 수 있다. 아래는 주된 기능을 몇 가지다.

1. 논리조작 : 논리조작에는 연산 장치에서 기억장치로 옮기는 store, 외부에서 내부 연산 장치로 정보를 옮기는 load, 명령의 실행 순서를 변경하는 branch, 정보들을 비교하는 compare 등이 있다.

2. 자리이동 : 소수점의 위치를 바꾸거나 소수점이하의 값을 없애버리는 기능이다.

3. 편집 : 존 형식 십진수를 팩 형식 십진수로 변환하거나 혹은 그 반대를 수행하는 기능이다.

4. 변환 : 이진수를 십진수로 변환하거나 혹은 그 반대를 수행하는 기능이다.

**6.**

보수기와 레지스터, 오버플로 검출기의 의의에 대해 알아보고자 한다.

보수기는 뺄셈을 수행할 때 빼는 수를 보수로 바꾸어 덧셈 연산을 처리하도록 하는 장치다. 따라서 이처럼 빼는 수를 보수로 바꿔는 회로가 보수기다,

레지스터는 연산에서 사용되는 데이터와 연산에 대한 결과를 기억하기 위해 저장하는 공간을 의미한다.

오버플로 검출기는 말그대로 오버플로를 검출하는 회로다. 모든 결과를 레지스터에 저장할 수 없을 때, 오버플로가 발생하는데 이때 에러를 측청하여 사용자에게 알려주는 회로다.