

10주차 예비보고서

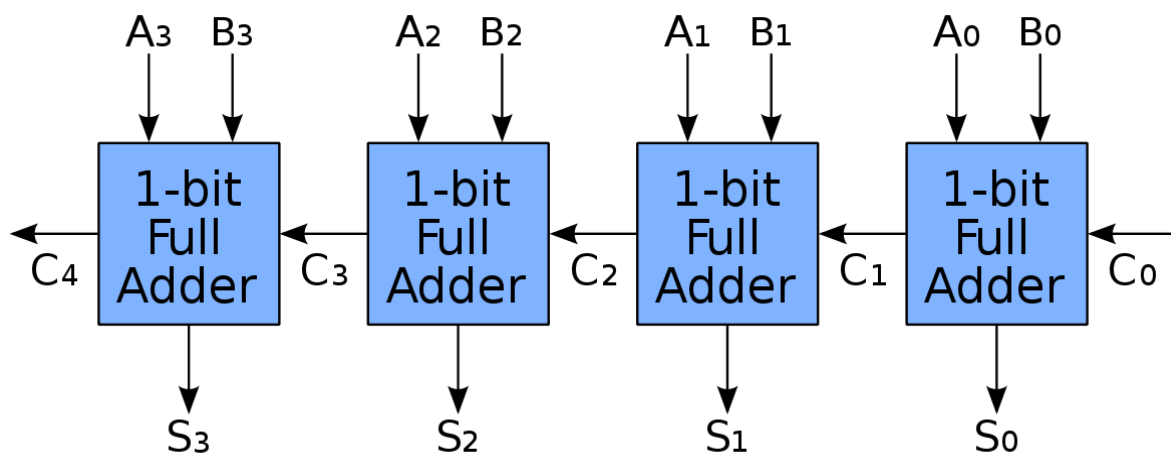
전공: 컴퓨터공학

학년: 2학년

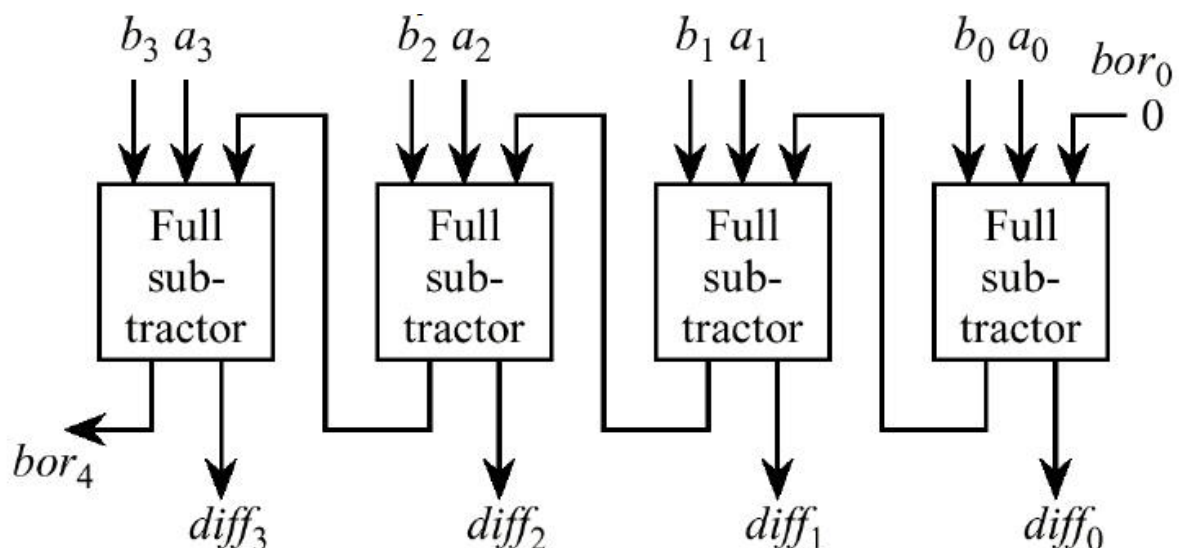
학번: 20191629

이름: 이주현

1. 4-bit adder 및 subtractor 이진 병렬 연산 기능에 대하여 조사하시오.



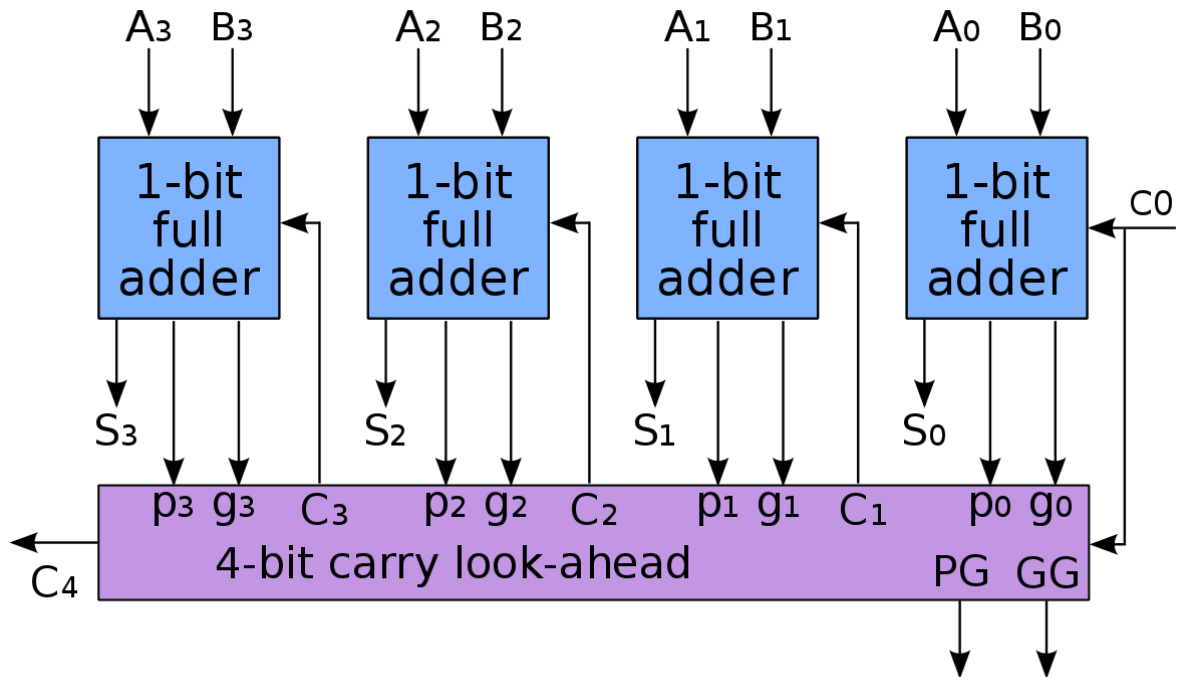
위와 같이 이전 소자의 받아올림 핀을 다음 소자의 받아올림 입력에 연결하여 간단하게 4-bit adder를 구현할 수 있다.



반대로 위와 같이 이전 소자의 받아내림 핀을 다음 소자의 받아내림 핀과 연결하여 4-bit subtractor를 구현할 수도 있다.

2. Lookahead carry에 대하여 조사하시오.

기본적으로 1번에서 설명한 것과 같은 가감산기는 여러 소자를 연결한 형태이기 때문에 전파 지연 시간이 크다는 단점이 있다. 이 전파 지연 시간은 받아올림 또는 받아내림 연산에 의해 영향을 가장 많이 받게 되는데, 이를 수정하기 위해 받아올림 또는 받아내림을 미리 계산하는 회로가 바로 carry lookahead 회로이다.



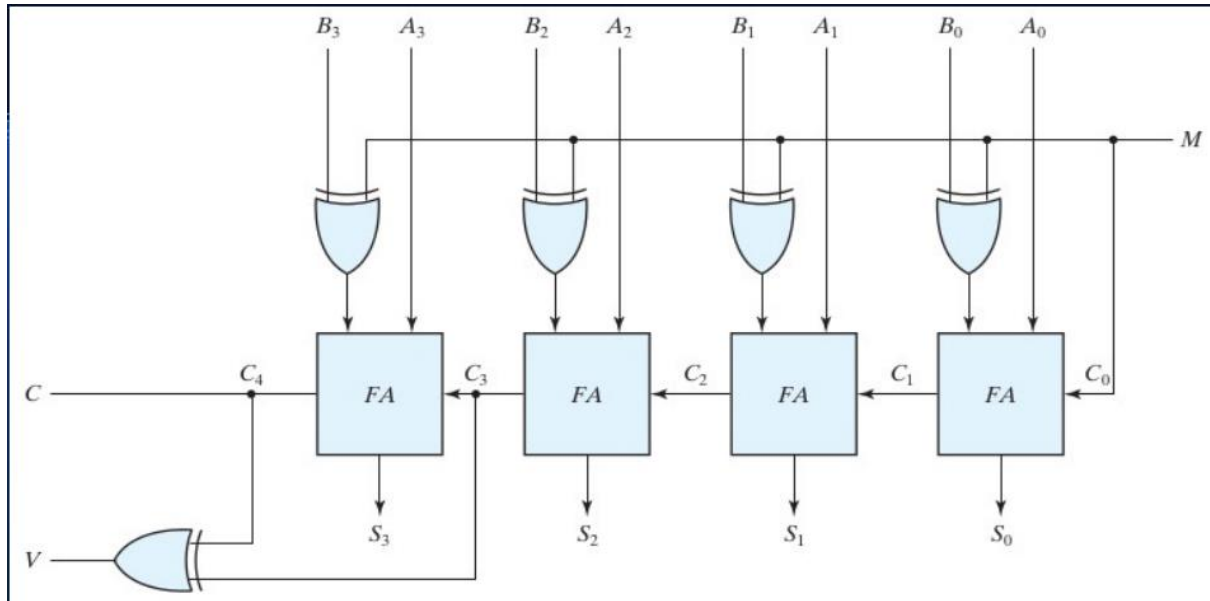
이렇게 미리 계산한 받아올림 비트를 사용하면 위와 같이 불필요한 전파 지연이 생기는 것을 막을 수 있어 더욱 빠르게 수학 연산이 가능해진다. 이 회로는 기본적으로 다음 식을 통하여 구성된다.

$$c_{n+1} = x_n y_n + c_n (x_n \oplus y_n)$$

이 재귀 식을 각 받아올림 비트에 대하여 전개하면 모든 비트에 대하여 받아올림을 순차적으로 계산할 필요가 없어지기 때문에 훨씬 빠르게 덧셈을 계산할 수 있다.

3. XOR을 활용한 2's complement 가감산에 대하여 조사하시오.

보통 adder와 subtractor는 수를 더하거나 수를 빼는 것밖에 할 수 없고, 서로가 서로의 기능을 수행할 수 없다. 그런데, 2의 보수법을 이용하면 가산기에 뺄셈 연산을 할 수 있도록 구현할 수 있다. 이를 두고 XOR 가감산기라고 부르는데, 다음과 같은 회로로 구성되어 있다.



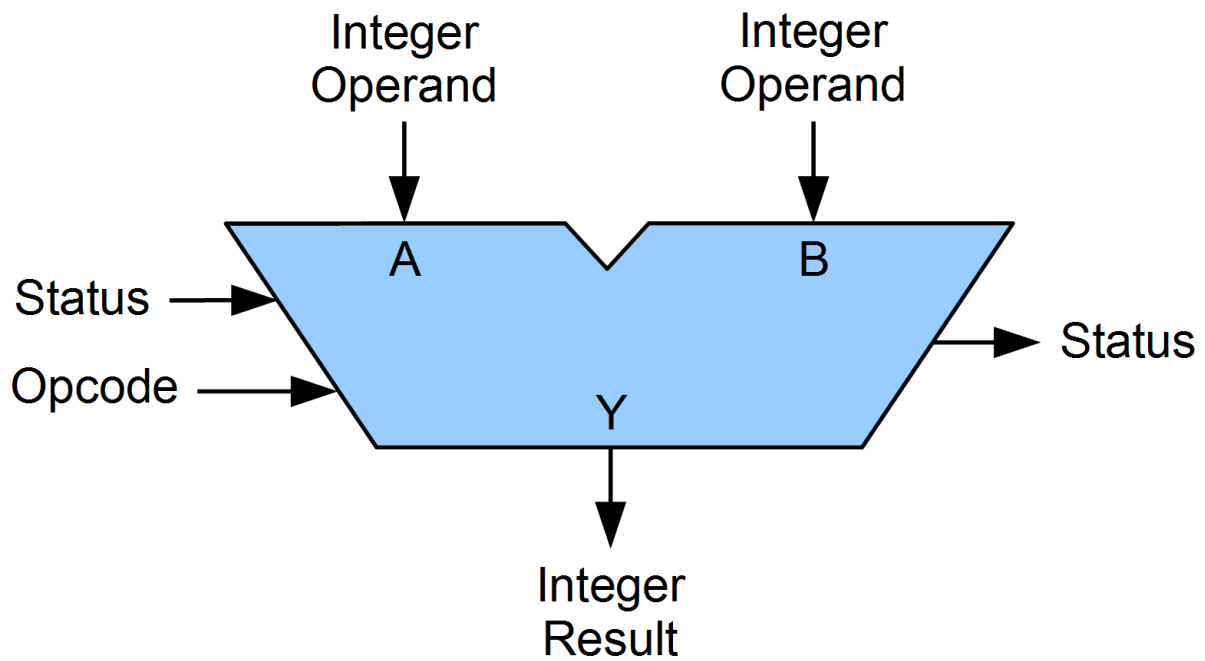
여기에서 M 회로는 덧셈인지 뺄셈인지 결정하는 입력이다. M 입력이 1이라면, B의 신호는 XOR 게이트에 의해 2의 보수로 바뀌게 되는데, 2의 보수와 일반적인 이진수를 더하는 것은 뺄셈을 하는 것과 같은 효과를 가진다. 만약 M 입력이 0이라면 B의 신호가 그대로 전달되므로, 일반적인 가산기의 효과를 가진다고 할 수 있다.

4. BCD 연산에 대하여 조사하시오.

BCD는 binary-coded decimal의 약자로, 10진수의 각 자릿수를 4비트 2진수로 나타낸 것을 의미한다. 즉, 1의 자리, 10의 자리, 100의 자리와 같은 10진수 자리수에 각각 4비트를 할당하여 2진수로 표현한다는 것이다. BCD를 연산하는 것은 기본적인 이진수 연산과 같지만, 한 가지 주의할 점은 4비트 이진수가 9를 넘는 수를 표현하게 된다면 다음 자릿수로 추가적으로 받아올림을 해 주어야 한다는 점이 특징이다.

5. ALU의 기능에 대하여 조사하시오.

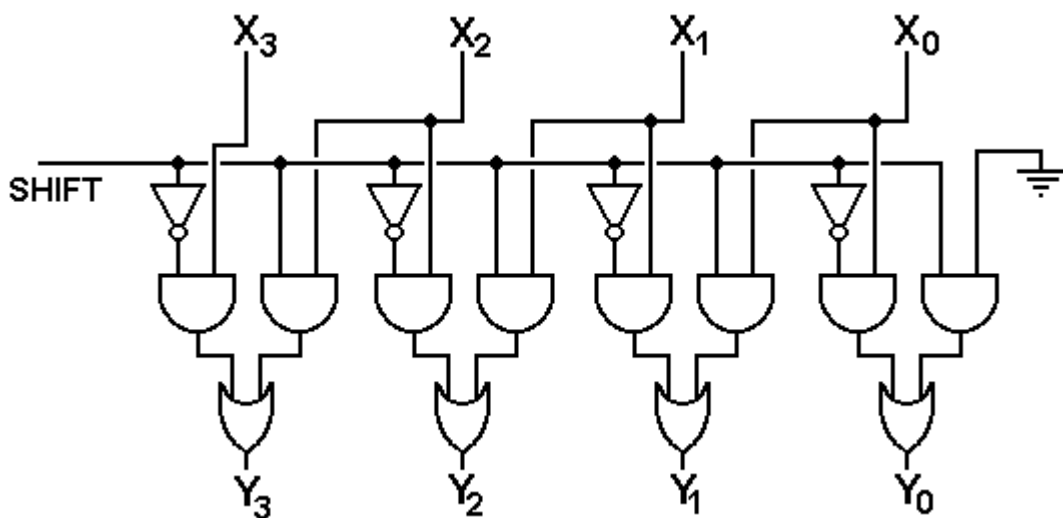
ALU는 arithmetic logic unit의 약자로, 산술 연산을 주로 행하는 CPU 내의 서브시스템이다. ALU는 연산할 데이터를 CPU의 레지스터 또는 주기억장치로부터 입력받고, 수행할 연산의 종류를 입력받아 해당 연산을 수행하는 논리 회로라고 볼 수 있다.

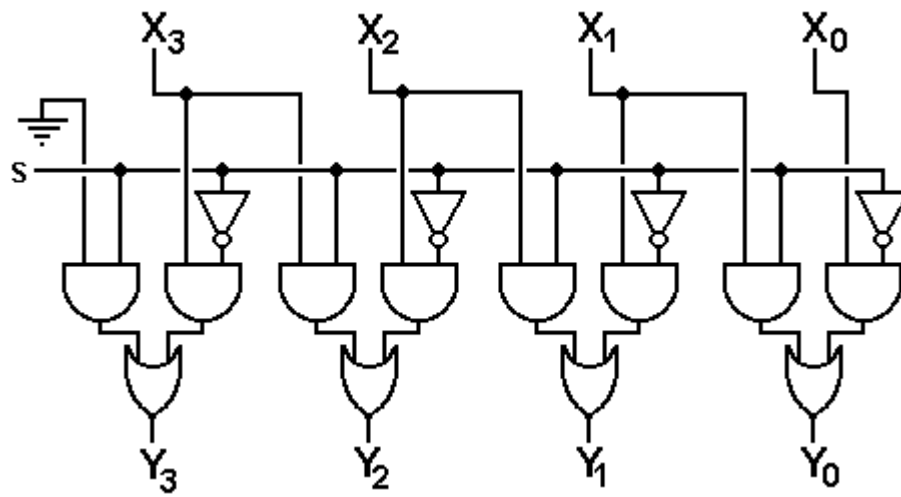


여기서 opcode는 operation code의 준말로, 어떤 연산을 할지 결정하는 입력이다. 이를 바탕으로 올바른 연산을 한 뒤, status 출력을 설정하고 결과값을 레지스터에 저장하는 논리 회로라고 볼 수 있다. Opcode의 내용은 ALU의 구현체마다 다르다.

6. 기타 이론

ALU가 주로 맡는 연산은 산술 연산과 논리 연산 이외에도 비트시프트 연산이 있다. 비트시프트 연산은 비트의 위치를 “옮기는” 것으로, 논리 비트시프트와 순환 비트시프트 두 가지 종류가 존재한다. x86 어셈블리에서 해당 연산의 opcode는 shl과 shr이다. 아래는 각각 왼쪽 시프트 연산 회로와 오른쪽 시프트 연산 회로이다.





논리적 시프트 연산을 수행하면, 시프트로 인해 밀려난 비트는 사라지고, 빈 비트 공간은 0으로 채워지게 된다.