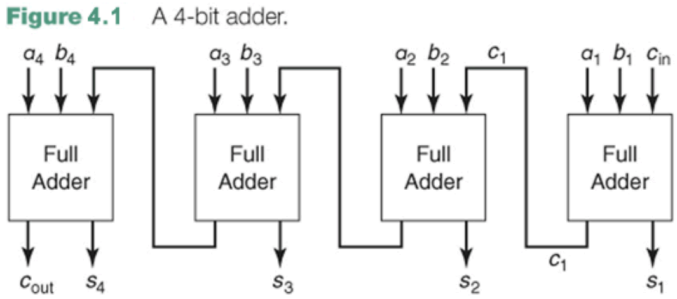
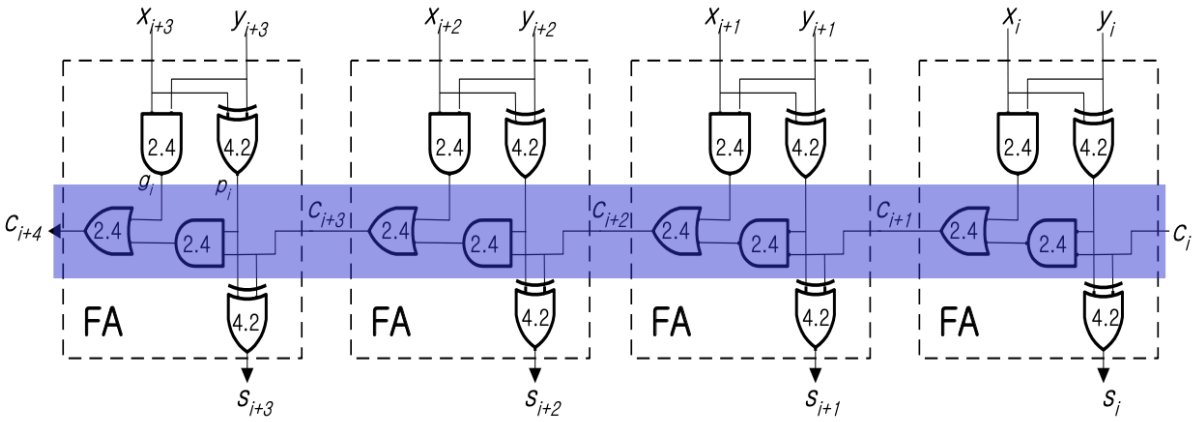
**컴퓨터공학실험 2 10 주차 예비보고서**

**전공: 컴퓨터공학과 학년: 2 학번: 20191559 이름: 강상원**

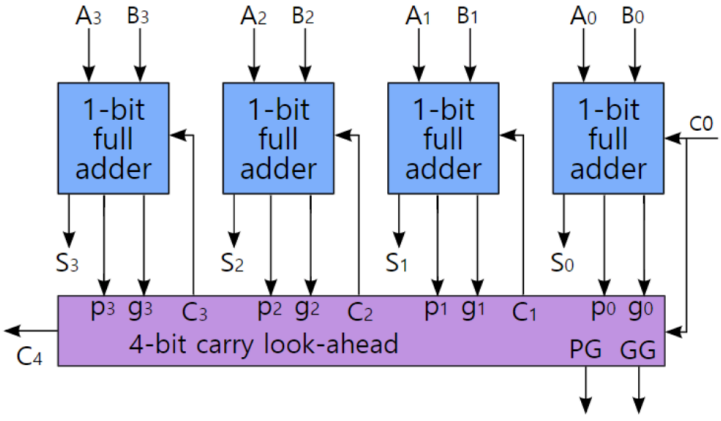
## 1. 4-Bit Adder 및 Substractor 이진 병렬 연산 기능에 대하여 조사하시오.



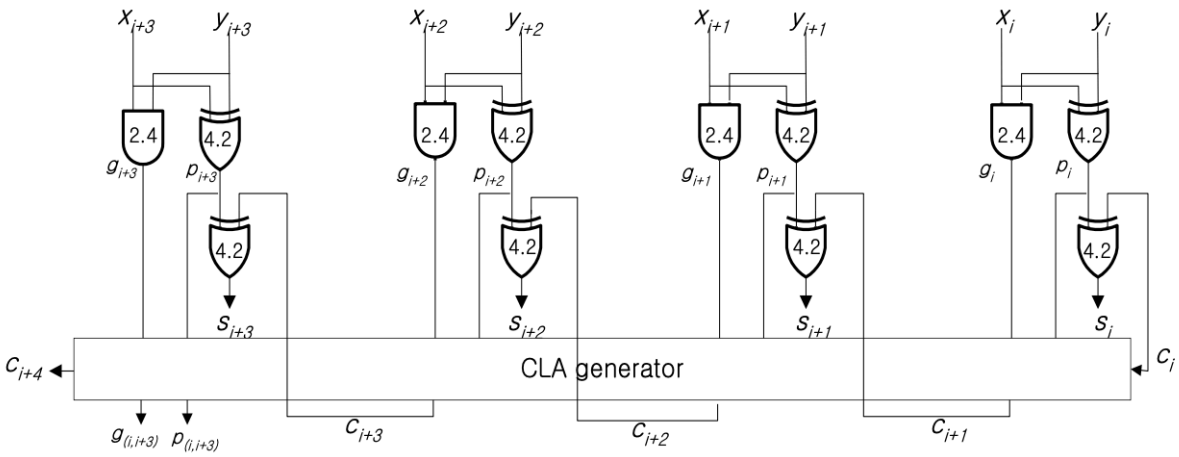

4개의 Full Adder를 병렬로 연결하여 위와 같이 나타낼 수 있다. (Ripple Carry Adder) 각 자리의 수를 더하고, (carry도 포함), 합을 반환하며 다음 전가산기에 새로운 carry를 넘겨주는 구조이다. 그러나 이와 같이 설계하면 각 단계에서 carry를 받기 위한 gate delay가 발생하므로 비효율적인 측면이 있다. (n-비트 가산기의 출력이 안정화되는 시간은 $(2n+4)\Delta$ 이다.) 이를 개선한 것이 아래와 같은 carry-look-ahead adder이다. 이에 대해서는 2번 항목에서 자세히 설명하겠다.

감산기 또한 가산기와 동일한 구조로 만들 수 있다. (Full Subtractor 이용)



**2. Look ahead carry 대하여 조사하시오.**

Look ahead carry는 앞서 1번 항목에서 말한, carry값 계산 결과를 기다리게 됨으로써 발생하는 delay 문제를 해결하는 방법이다. Carry를 미리 계산하는 회로를 만들어 (Carry Lookahead Logic) 처리한다. 자세한 구동 방법은 앞선 주차(6주차)의 보고서에 정리되어 있으나 하단에 한번 더 언급하도록 한다.




Carry Generate $g_i = x_i y_i$. Carry Propagate $p_i = x_i \oplus y_i$라 부른다.

이를 이용해 정리하면

$c_{i+1} = g_i + p_i c_i$

$c_{i+2} = g_{i+1} + p_{i+1} g_i + p_{i+1} p_i c_i$

$c_{i+3} = g_{i+2} + p_{i+2} g_{i+1} + p_{i+2} p_{i+1} g_i + p_{i+2} p_{i+1} p_i c_i$

$c_{i+4} = g_{i+3} + p_{i+3} g_{i+2} + p_{i+3} p_{i+2} g_{i+1} + p_{i+3} p_{i+2} p_{i+1} g_i + p_{i+3} p_{i+2} p_{i+1} p_i c_i$

와 같이 표현된다. 간략화된 버전은 아래와 같다.

$c_{i+1} = g_i + p_i c_i$

$c_{i+2} = g_{i+1} + p_{i+1} c_{i+1}$

$c_{i+3} = g_{i+2} + p_{i+2} c_{i+2}$

$c_{i+4} = g_{i+3} + p_{i+3} c_{i+3}$

∴ 직접 입력 bit 와 carry $c_i$로부터 계산 수행이 가능하다.

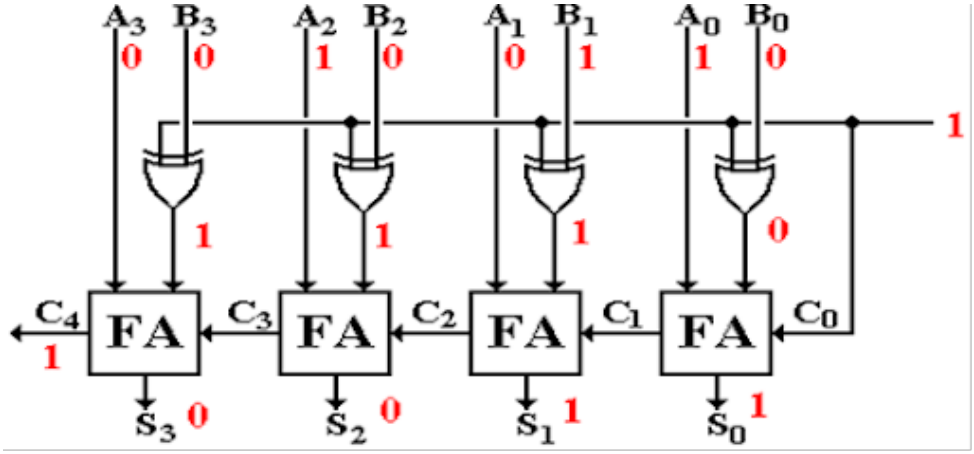
➔ 다음 carry를 예측하여, 그 이전 carry가 계산되어 있는 상태가 아닐지라도 두 입력값 A와 B만으로도 carry를 구할 수 있게 된다.

이러한 방식으로 Carry Look-Ahead Logic 이 구성된다.

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

이미 만들어진 가산기 회로를 이용하여 감산기 기능까지 작동하는 가감산기를 구현할 수 있다. Control 신호의 입력에 따라, 가산기 동작 / 감산기 동작이 결정되는데, 일반적으로 0이 입력되면 가산기 동작, 1이 입력되면 감산기 동작을 수행한다. 아래 회로도를 통해 작동 원리를 이해할 수 있는데, control 신호가 1이라면 자동적으로 B의 값을 2의 보수화 함을 볼 수 있다.

어떤 수의 2의 보수를 취한 것을 더한 것은 그 수를 뺀 것과 같다는 것은 이미 알려진(배운) 사실이다.




**4. BCD 연산에 대하여 조사하시오.**

BCD(Binary-coded decimal)는 10진수 자리끼리 2진수로 나타낸 것이다. 4 bit로 0 ~ 9의 수를 표현하고 다음 자리수도 마찬가지로 표현한다. 기본 이진법 수를 BCD로 변환하기 위해서는 원래 수를 계속해서 10으로 나누며 그 나머지를 얻어야 한다.

일반 이진법으로 4 + 9를 나타내면 1101이 되지만, BCD로 나타내면 0001 0011이다.

➔ 각 자리의 계산 결과 값이 10 이상이면 6 (0110)을 더해 그 위 자리에서 처리해 주어야 한다.

다음은 일반적인 8-4-2-1 BCD 의 표현 방법이다.

| BCD | | Decimal |
|---|---|---|
| 0000 | = | 0 |
| 0001 | = | 1 |
| 0010 | = | 2 |
| 0011 | = | 3 |
| 0100 | = | 4 |
| 0101 | = | 5 |
| 0110 | = | 6 |
| 0111 | = | 7 |
| 1000 | = | 8 |
| 1001 | = | 9 |