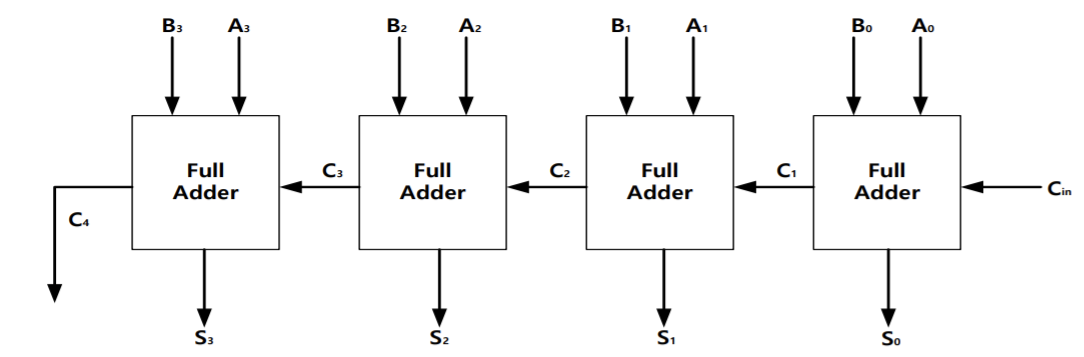
10주차 예비보고서

전공 : 컴퓨터공학과 학년 : 2학년 학번 : 20201597 이름 : 신동준

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

이전에 공부했던 1-bit full adder를 병렬로 n개 연결하여서 n-bit adder를 만들 수 있다. 4-bit adder는 4개의 full adder를 연결하여 아래와 같은 circuit의 형태를 가진다.



각각 들어오는 수를 자리 수 별로 나눠서 full adder에 넣고, 이전 full adder의 carry out을 다음 full adder에 carry in으로 넣을 수 있다. Subtractor도 1-bit full subtractor의 병렬인 동일한 형태로 구성할 수 있다.

1. **Look ahead carry 대하여 조사하시오.**

Look ahead carry는 Carry look Ahead Full Adder(이하 CLA)에서 사용되는 carry bit이다. 기존의 adder의 문제점을 극복하기 위함이다. 위의 이진 병렬 가산기의 경우 낮은 자리수의 연산이 끝나면 그 carry를 받아서 다음 연산을 진행하는 방식인데, n-bit끼리 합치려면 숫자가 커질 때 연산에 많은 시간이 필요하다. 이때 CLA는 sum연산과 carry연산을 분리해서 빠르게 계산할 수 있도록 한다.

방법은 먼저 G(generation)과 P(propagation)을 G = A and B; P = A xor B; 로 구한다.

이후 i번째에서 Si = Pi xor Ci; C(i+1) = Gi + PiCi;의 형태로(점화식 형태로) 구할 수 있다. 이렇게 하면 각각의 carry가 이미 존재하는 연산에 상관없이 처리될 수 있기에 더 빠른 연산을 할 수 있다.

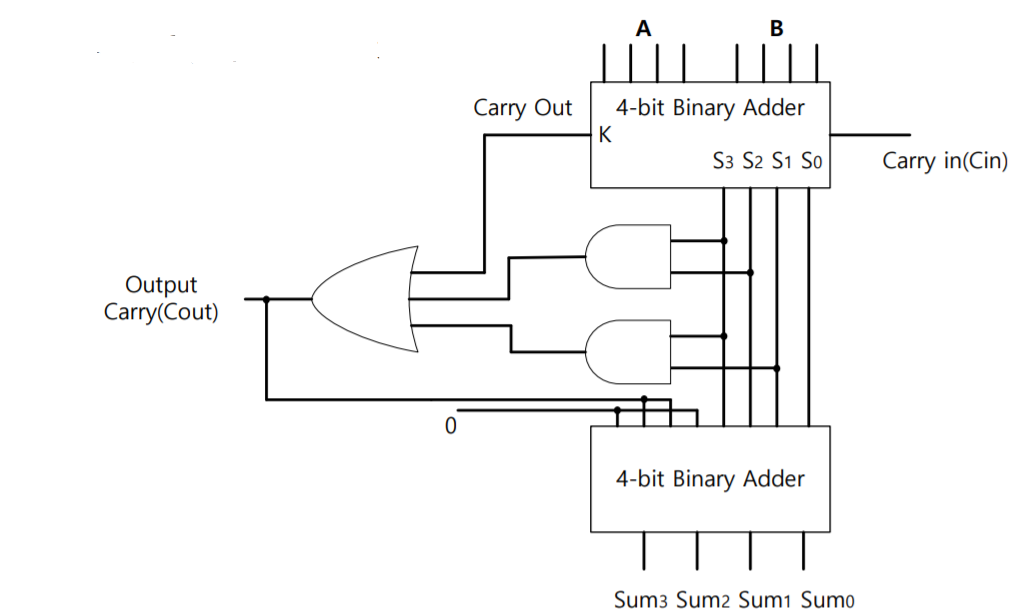
1. **XOR을 활용한 2‘s complement 가감산에 대하여 조사하시오.**

2의 보수 형태로 양음의 부호 구분이 있는 두 이진수에 대해서 덧셈, 뺄셈을 한다.

양수를 음수로, 음수를 양수로 변환할 때 XOR을 이용해서 0을 1로 1을 0으로 바꾼 뒤 1을 더해주게 된다. 입력 값이 둘다 양수 일 때 뺄셈은 음수의 덧셈과 같기 때문에 덧셈 뺄셈 여부를 K로 받고 K가 1일 때는 뺄셈으로 보고 모든 수를 XOR연산한 뒤 adder에 넣어주고, K가 0일 때는 덧셈으로 보고 그대로(위의 1번 방식으로) adder에 넣어준다.

1. **BCD 연산에 대하여 조사하시오.**

BCD연산은 10진수의 덧셈(뺄셈)을 위한 logic이다. BCD로 표현된 10진수 2개를 입력받고, 이 둘을 더해서 다시 BCD로 출력해주면 된다. 이때 0~9까지의 숫자, BCD 형태일 때 0000~1001 사이의 숫자만 이용하게 된다. (4bit기준) 4-bit Binary Sum과 0~9를 표현함에 있어서는 차이가 없지만, 10~19를 표현할 때는 10을 예로 들어 Binary Sum에서는 01010, BCD Sum에서는 10000으로 표현된다. 6이 더해진 binary 형태로 표기되는데 Carry bit의 역할이 기존에는 단순 overflow의 표현으로 자리수가 늘어나는 개념이었다면, BCD에서는 십의자리 역할을 하게 된다. 회로의 구성은 아래와 같이 4-bit binary adder로 합치고 이를 BCD형태로 변환시켜주는 과정이 합쳐져있다.



1. **ALU의 기능에 대하여 조사하시오.**

ALU란 arithmetic logic unit의 줄임말로 산술연산과 논리연산을 수행하는 디지털 회로이다. 이때 산술연산(덧셈, 뺄셈, shift)를 담당하는 부분을 AU, 논리연산(논리곱, 논리합, 배타적논리합)등을 수행하는 부분을 LU라고 한다. ALU는 연산을 위한 수를 입력받고, 어떤연산을 할지 유무를 결정하는 Selection number도 입력받는다. 입력받는 line을 Select Line이라고 하는데 n개의 Select line이 있으면 최대 2^n개의 연산을 선택해서 수행할 수 있음을 의미한다. 이는 컴퓨터 CPU의 기본 단위라고도 볼 수 있다.

1. **기타 이론.**

MSI/LSI는 집적회로의 분류 중 하나이다. 집적회로의 집적도에 따라서 SSI, MSI, LSI, VLSI로 구분하게 되는데, Small-scale, Medium-scale, Large-scale integration의 약자이다. SSI는 가장 작은 단위로 10개 이내의 게이트, 인버터로 구성되며, flip-flop 등이 해당된다.

MSI는 100개 이내의 게이트를 포함하며 우리가 배운 Adder, Register, MUX 등이 해당된다. LSI는 이들이 모여 만든 마이크로프로세서가 해당되며 수천 개 이내의 게이트를 포함한다. 여기서 더 큰 규모가 되면 CPU와 같은 VLSI(ULSI)라고 지칭한다.