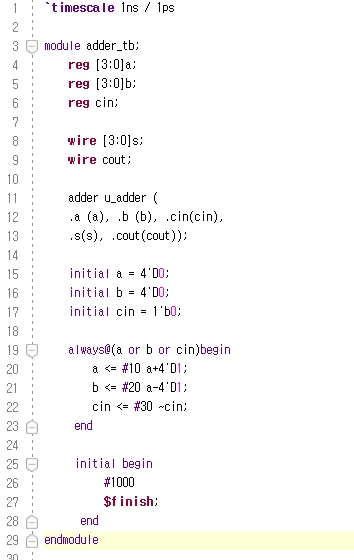
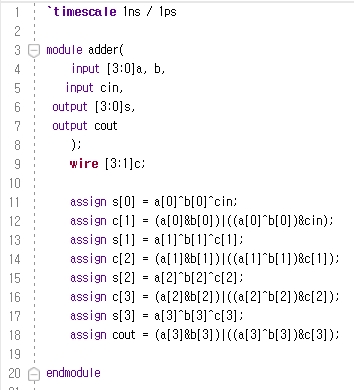
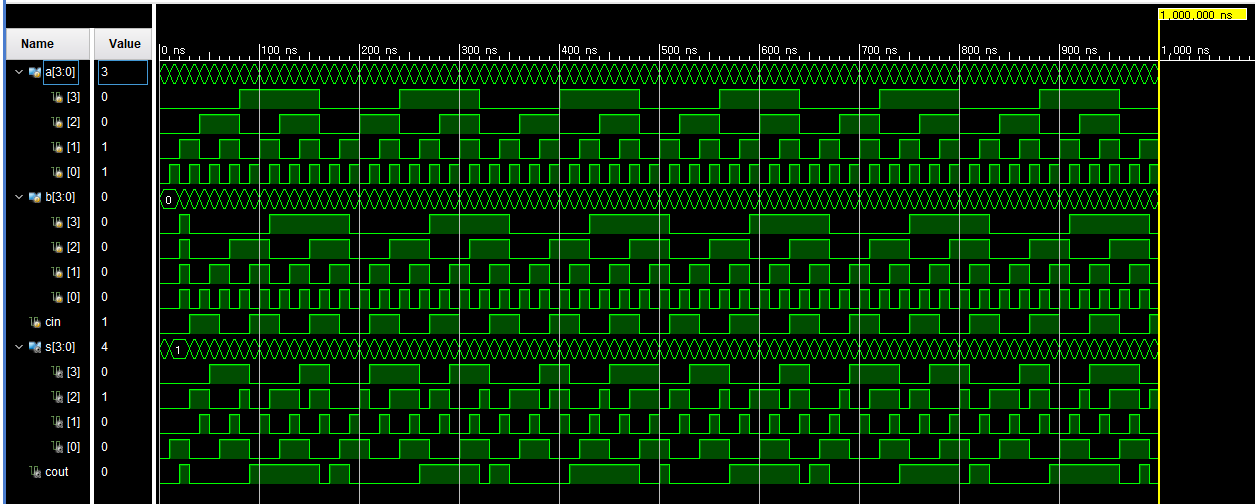
10주차 결과보고서

전공: 기계공학과 학년: 3학년 학번: 20191820 이름: 김형준

**1.**



[4bit Binary Parallel Adder의 Verilog Code]



[4bit Binary Parallel Adder의 Simulation 결과]

4bit Binary Parallel Adder(이진 병렬 가산기)는 1bit Full Adder 4개를 병렬로 연결하여 4bit의 두 이진수의 덧셈연산을 수행하는 논리 회로이다.

9 input (), 5 output () 회로로, 4비트의 두 이진수(A, B)와 입력 캐리()를 입력으로 받아 4비트의 합(S)과 캐리()를 출력한다.

입력 캐리()의 값은 4비트의 두 이진수의 덧셈 연산에서는 항상 0이 되지만, 8비트 이진 병렬 가산기를 4비트 이진 병렬 가산기 2개를 연결해 구현하는 경우, 상위 4비트의 가산기는 하위 4비트의 가산기의 출력 캐리()를 입력 캐리()로 받으므로 하위 4비트 가산기의 출력 캐리의 값이 1인 경우에는 상위 4비트 가산기의 입력 캐리의 값이 1이 될 수 있다.

4bit Binary Parallel Adder의 각 자릿수는 1bit Full Adder로 이루어져 있으므로 각 자릿수의 합과 출력 캐리는 1bit Full Adder와 논리식이 같다.

따라서 4bit Binary Parallel Adder의 논리식은 아래와 같다. ()

4bit Binary Parallel Adder는 LSB(i=0)부터 MSB(i=3)까지 순차적으로 연산이 수행되므로

먼저 이 LSB(i=0)의 Full Adder의 입력으로 들어와서 이 계산된다.

그 다음, 다음 자릿수(i=1)의 Full Adder에서 이 입력으로 들어와서 이 계산되고, 그 다음 자릿수(i=2)의 Full Adder에서 이 입력으로 들어와서 이 계산된다. 마지막으로, MSB(i=3)의 Full Adder에서 이 입력으로 들어와서 이 계산된다. 위 Verilog Code에서는 입력값 A, B와 출력값 S를 배열을 사용하여 나타내었으며, 입력값과 출력값이 아닌 캐리()은 wire타입 배열로 나타내었다.

또한, LSB부터 MSB까지 각 자릿수의 합과 캐리를 순차적으로 계산한다.

위 Simulation 결과 중에서 예시를 들면, 인 경우,

이다.

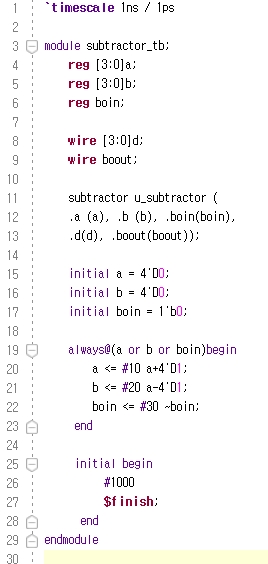
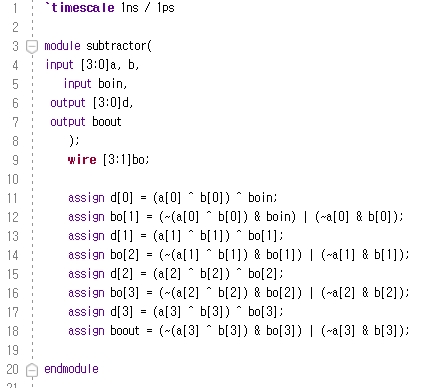
먼저 LSB에서 이 계산되고, 다음 자릿수에서 이 계산된다. 그 다음 자릿수에서는 이 계산되고, 마지막으로 MSB에서 이 계산되어 연산이 끝난다. 결과적으로 이 계산된다.

또 다른 예시를 들면, 인 경우,

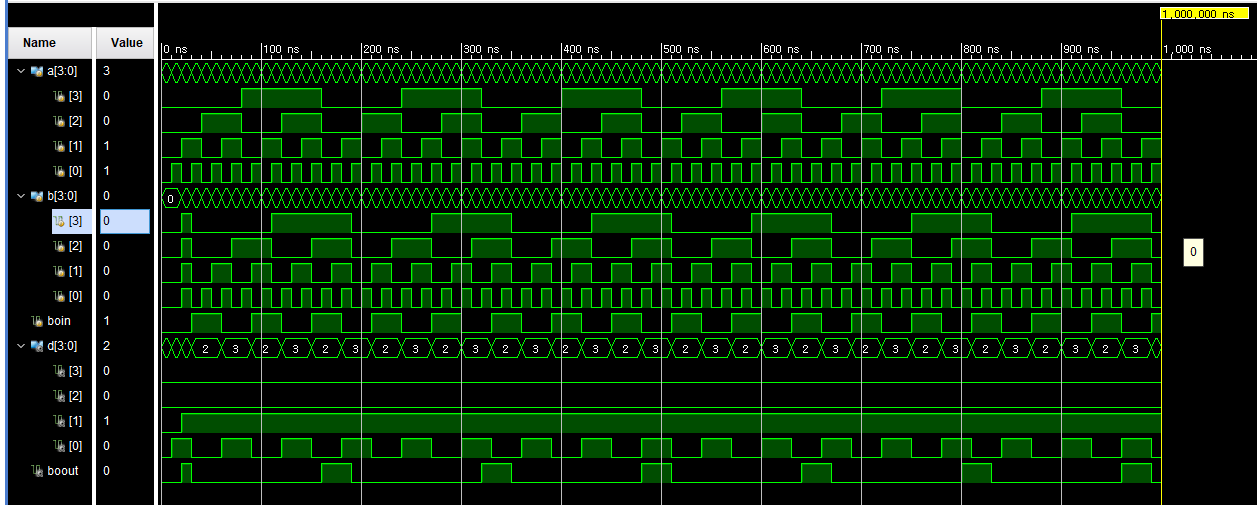
이므로 마찬가지로,

이 되어 결과적으로 이 계산된다.

**2.**



[4bit Binary Parallel Subtractor의 Verilog Code]



[4bit Binary Parallel Subtractor의 Simulation 결과]

4bit Binary Parallel Subtractor(이진 병렬 감산기)는 1bit Full Subtractor 4개를 병렬로 연결하여 4bit의 두 이진수의 뺄셈연산을 수행하는 논리 회로이다.

9 input (), 5 output () 회로로, 4비트의 두 이진수(A, B)와 입력 빌림수()를 입력으로 받아 4비트의 차(D)과 빌림수()를 출력한다.

입력 빌림수()의 값은 4비트의 두 이진수의 뺄셈 연산에서는 항상 0이 되지만, 8비트 이진 병렬 감산기를 4비트 이진 병렬 감산기 2개를 연결해 구현하는 경우, 상위 4비트의 감산기는 하위 4비트의 감산기의 출력 빌림수()를 입력 빌림수()로 받으므로 하위 4비트 감산기의 출력 빌림수의 값이 1인 경우에는 상위 4비트 감산기의 입력 빌림의 값이 1이 될 수 있다.

4bit Binary Parallel Subtractor의 각 자릿수는 1bit Full Subtractor로 이루어져 있으므로 각 자릿수의 차와 출력 빌림수는 1bit Full Subtractor와 논리식이 같다.

따라서 4bit Binary Parallel Subtractor의 논리식은 아래와 같다. ()

4bit Binary Parallel Subtractor는 LSB(i=0)부터 MSB(i=3)까지 순차적으로 연산이 수행되므로 먼저 이 LSB(i=0)의 Full Subtractor의 입력으로 들어와서 이 계산된다. 그 다음, 다음 자릿수(i=1)의 Full Subtractor에서 이 입력으로 들어와서 이 계산되고, 그 다음 자릿수(i=2)의 Full Subtractor에서 이 입력으로 들어와서 이 계산된다. 마지막으로, MSB(i=3)의 Full Subtractor에서 이 입력으로 들어와서 이 계산된다. 위 Verilog Code에서는 입력값 A, B와 출력값 D를 배열을 사용하여 나타내었으며, 입력값과 출력값이 아닌 빌림수()는 wire타입 배열로 나타내었다. 또한, LSB부터 MSB까지 각 자릿수의 차와 빌림수를 순차적으로 계산한다.

위 Simulation 결과 중에서 예시를 들면, 인 경우,

이다.

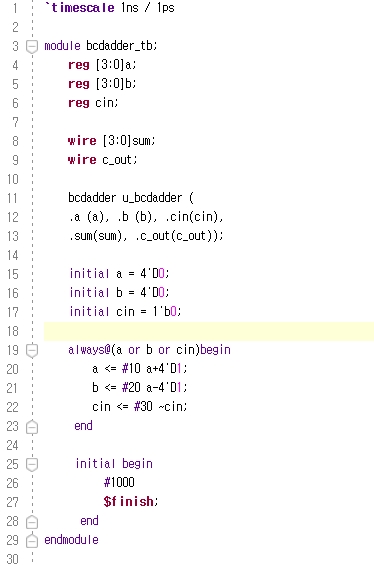
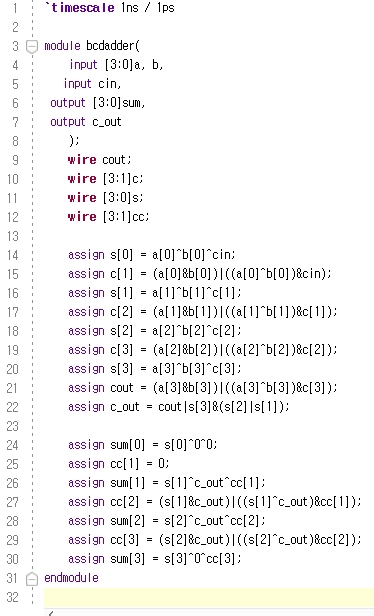
먼저 LSB에서 이 계산되고, 다음 자릿수에서 이 계산된다. 그 다음 자릿수에서는 이 계산되고, 마지막으로 MSB에서 이 계산되어 연산이 끝난다. 결과적으로 이 계산된다.

또 다른 예시를 들면, 인 경우,

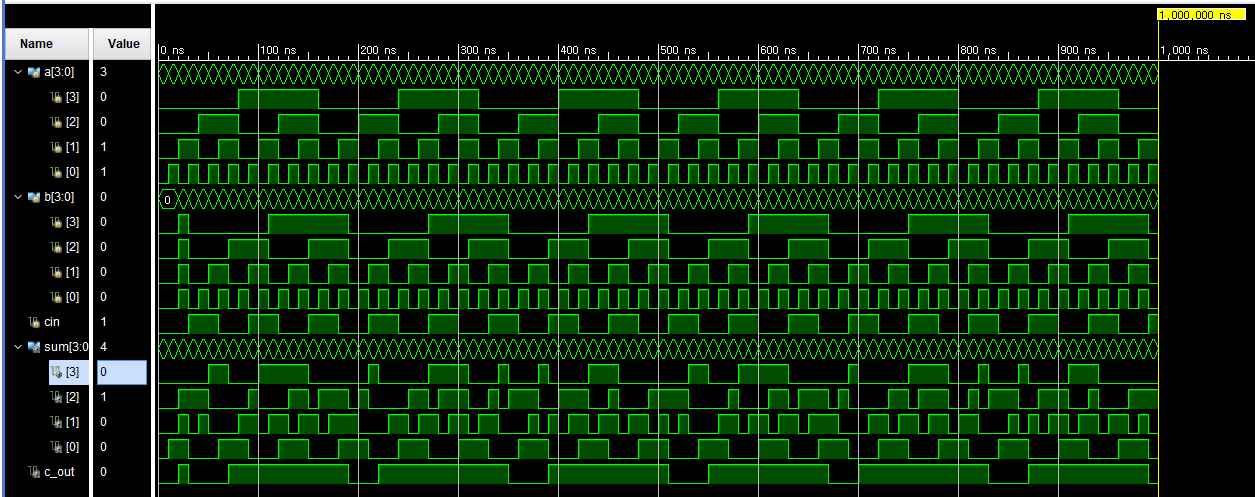
이므로 마찬가지로,

이 되어 결과적으로 이 계산된다.

**3.**

****

[4bit BCD Adder의 Verilog Code]

****

[4bit BCD Adder의 Simulation 결과]

BCD Adder는 두 개의 4bit BCD코드의 덧셈연산을 수행하는 논리 회로이다.

BCD Adder 2개의 4bit Binary Parallel Adder를 연결하여 만들 수 있다.

9 input (), 5 output () 회로로, 4비트의 두 BCD 코드(A, B)와 입력 캐리()를 입력으로 받아 BCD 코드의 합(S)과 캐리()를 출력한다.

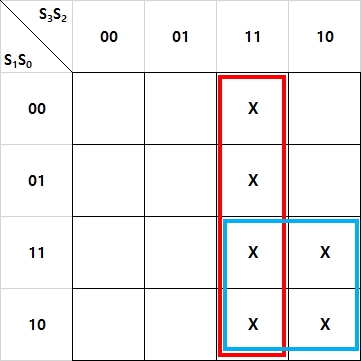
BCD Adder에 입력으로 들어오는 4비트의 두 이진수는 10()미만의 값만이 들어오는데, 이는 BCD 코드는 한 자리수의 십진수를 표현하므로 0()부터 9() 사이의 값 만을 가지기 때문이다.

입력 캐리()의 값은 4bit Binary Parallel Adder에서와 마찬가지로 두 자리수 이상의

BCD 연산을 수행할 때 하위 자리수의 BCD Adder의 출력 캐리() 값이 상위 자리수의 BCD Adder의 입력 캐리()로 들어오므로 상위 자리수의 입력 캐리의 값은 0또는 1의 값을 가질 수 있다.

BCD Adder에서는 먼저, 입력으로 들어온 두 이진수를 더한 다음, 합이 10 이상이면 출력 캐리의 값을 1로 설정하고 합에 6()을 추가로 더한다. (한자리수의 10진수로 표기 불가)

합이 10 미만이면 출력 캐리의 값을 0으로 설정하고 합에 0()을 추가로 더한다. (한자리수의 10진수로 표기 가능)



[합(S)이 10 이상인 경우의 카르노맵]

두 BCD 코드의 합이 10 이상이 되는 경우를 위와 같이 카르노맵을 그려 구하면,

이 되고, 출력 캐리의 값은 두 BCD 코드의 합이 10 이상이 되거나 4비트 가산기의 출력 캐리가 1인 경우이므로 이 된다.

위 Verilog Code에서는 입력값 A, B와 최종 출력값 sum을 배열을 사용하여 나타내었으며, 입력값과 출력값이 아닌 캐리()와 중간 결과값()는 wire타입 배열로 나타내었다. 또한, 입력으로 들어온 두 BCD코드를 먼저 더하고, 출력 캐리값을 구한 다음, 출력 캐리값이 1이면 중간 결과값(S)에 6을 더하고, 0이면 중간 결과값에 0을 더한다.

위 Simulation 결과 중에서 예시를 들면, 인 경우,

이므로

이 되어 중간 결과값으로 이 계산된다.

이므로 출력 캐리의 값이 1이므로 결과값에 6()을 더하면, (는 항상 0임)

이 되어 최종 결과값으로 이 계산된다. (는 필요 없으므로 구하지 않음)

**4.**

이번 실험을 통해 4bit Binary Parallel Adder, 4bit Binary Parallel Subtractor, BCD Adder의 구조와 작동방식에 대해서 이해할 수 있었다. 4bit Binary Parallel Adder/Subtractor는 각각 4개의 전가산기/전감산기를 이전 자릿수의 출력 캐리를 다음 자릿수의 입력 캐리로 연결하는 방식으로 병렬로 연결하여 구현할 수 있었다.

4bit Binary Parallel Subtractor의 경우, 2의 보수법을 사용하여 4개의 가산기를 사용하여 구현할 수도 있는데, 뺄셈 연산을 덧셈 연산으로 바꿔서() LSB의 입력 캐리()에 1을 입력하고, 입력값 B의 각 비트가 가산기로 들어가기 전의 위치에 NOT 게이트를 추가한다. 이 경우 결과값은 가 되므로 두 이진수의 뺄셈을 구한 것과 같은 결과를 얻게 된다.

또한, NOT 게이트 대신 XOR 게이트를 사용하고, Mode bit(M)을 추가해서 입력 캐리와 XOR 게이트의 한 입력에 연결함으로써 M의 값에 따라 덧셈과 뺄셈 연산을 모두 수행할 수 있는 4비트 병렬 가감산기도 만들 수 있다.

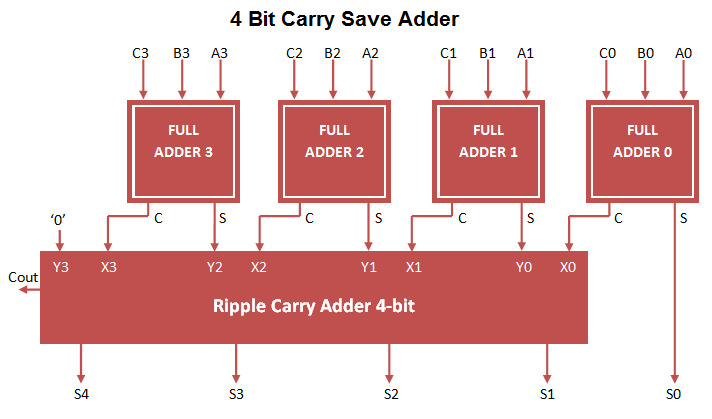
BCD Adder의 경우, 두 이진수의 합을 구한 다음. 합이 10 이상인지의 여부에 따라 출력 캐리의 값을 결정하고, 출력 캐리의 값에 따라 6 또는 0을 추가로 더하는 형식으로 구현하였다.

**5.**

- Carry Save Adder

Carry Save Adder는 이진법에서 3개 이상의 비트에 대한 덧셈을 계산하기 위한 가산기이다. 두 수의 각각의 비트는 전 가산기를 통해 합과 캐리를 구한 다음, 이 중간 결과를 Ripple Carry Adder를 사용해 최종 결과를 얻는 가산기이다.

이 가산기는 피연산자가 여러개 일 때, 병렬성을 이용하여 계산성능을 크게 향상시킬 수 있다는 장점이 있다. 일반적으로 피연산자가 여러 개인 덧셈연산을 수행하는 곱셈기를 설계할 때 사용된다.



- Carry Select Adder

Carry Select Adder는 각 자릿수의 캐리값이 0 또는 1인 경우 모두에 대해 계산을 미리 하고 이전 자릿수의 연산 결과에 따라 두 경우 중 하나의 연산 결과를 선택하는 형식의 가산기이다. 이때, 결과값의 선택에는 멀티플렉서가 사용된다.

다른 구조의 가산기보다 고속 연산을 수행할 수 있다는 장점이 있다.

