NEWTC FPGA 강좌 4. 계층구조 설계하기 (Byte Adder)

(주) 뉴티씨 (N EWTC)
http://www.newtc.co.kr

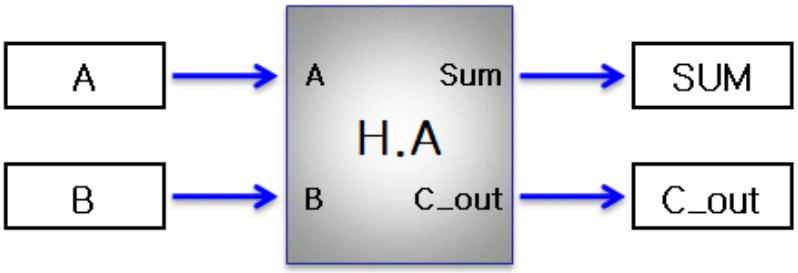
1 계층 구조 설계

이번 강좌에서는 바이트 덧셈기를 계층 구조를 이용하여 설계해 보겠습니다. Half Adder를 설계하고 이 응용하여 Full Adder 및 Byte Adder를 설계하는 것입니다.

2 계층구조 설계하기 (덧셈기)

2.1 Half Adder 설계

Half Adder란 2비트 덧셈기를 말합니다. 아래 그림과 같이 A, B 입력이 들어가고 C_out (Carry out) 과 Sum 출력이 나옵니다.



이것을 진리표로 나타내면 아래 그림과 같습니다.

Input		Output	
A	B	C_out	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Sum은 A와 B중 하나가 1일 경우 1이 되며 C_out은 A와 B 둘다 1일 경우 1이 됩니다. 따라서 위 진리표를 논리 연산으로 표현하면 아래와 같습니다.

Sum	=	A ^ B	(XOR)
C_out	=	A & B	(AND)

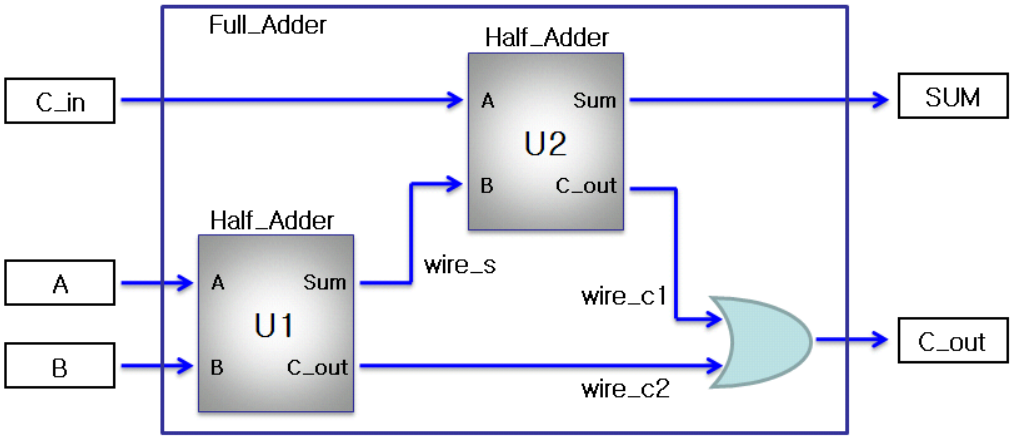
이것을 Verilog HDL을 이용하여 설계하고 시뮬레이션 합니다.

AND, OR, XOR 게이트는 기본적으로 제공됩니다. 여러가지 방법으로 설계할 수 있지만 XOR와 AND 게이트를 이용하여 설계합니다.

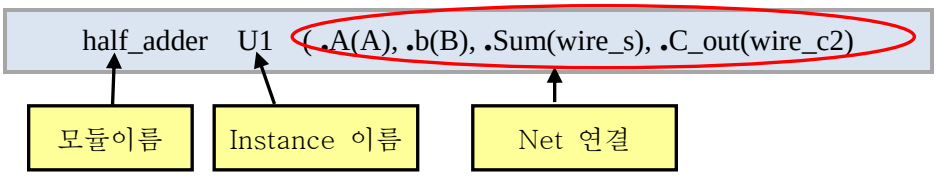
로직 게이트 사용법
xor (Q, A, B) // Q : 출력, A/B : 입력
and (Q, A, B) // Q : 출력, A/B : 입력

2.2 Full Adder 설계

Full Adder란 3비트 덧셈기를 말합니다. 2.1 장에서 설계한 Half Adder를 이용하여 Full Adder를 설계 합니다. Full Adder 는 Half Adder 2개와 OR 게이트 1개로 구성할 수 있습니다. 내부에 wire 가 3개 있는데 이름은 아래와 같이 정하도록 합니다.



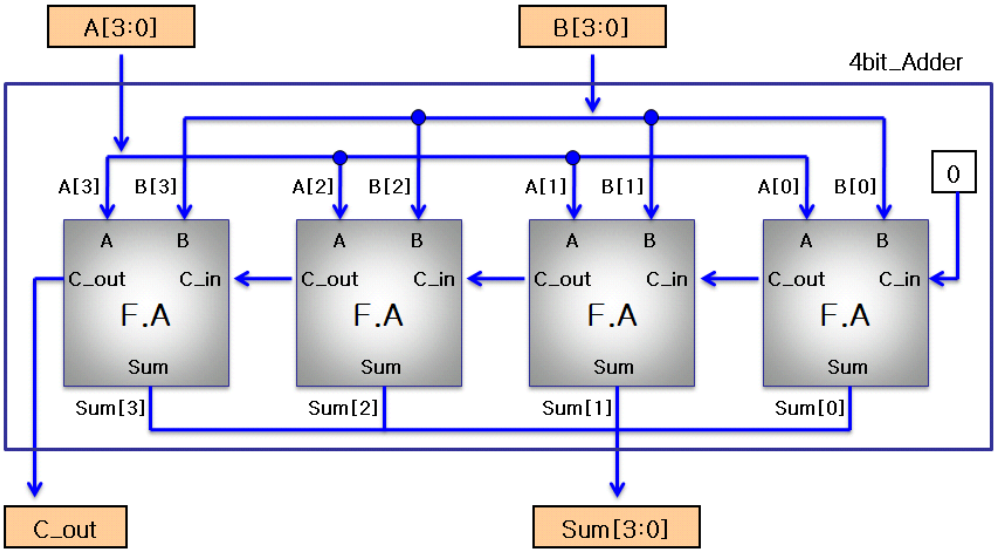
Half Adder를 사용하는 방법은 아래와 같습니다. Instance 이름은 모듈을 생성을 때 실제 생성된 모듈의 이름입니다. 여기서는 2개의 half_adder 를 생성했습니다. 모듈의 실제 구성은 half_adder 이지만 생성된 이름은 U1, U2와 같이 Instance 이름이 됩니다. 아래 예제에서는 이름을 이용하여 포트를 연결한 것입니다. 각 포트 연결은 "**.**포트명(Wire명)" 형식으로 합니다. "**.**"으로 시작하고 포트명과 Wire명이 나오게 됩니다. 괄호 안에 값이 상위 블록에서 사용하는 Wire명입니다.



포트를 연결하는 또 다른 방법으로는 포트의 위치를 이용한 방법이 있습니다. half_adder를 설계할 때 AND, OR, XOR를 이용한 방법과 같습니다.

2.3 Byte Adder 설계

아래 그림은 Full Adder 를 이용한 4bit Adder 블록도 입니다. 이와 같은 구조로 byte_adder를 설계하고 시뮬레이션 합니다.



2.4 시간 지연 시뮬레이션

시간 지연을 시뮬레이션 하기 위해서는 시뮬레이션 시간 단위를 지정해야 합니다. 시간 단위는 소스 파일 상단에 아래와 같이 timescale이라는 구문을 이용하여 기술합니다.

1ns로 단위 시간 지정

```
`timescale        1ns/10ps
```

앞에 기술한 1ns는 시뮬레이션 시간단위(1스텝시간)이고 슬러시 뒤의 10ps는 시뮬레이터에서 사용하는 정밀도입니다. 위와 같이 기술한 경우 #10은 10ns시간 지연의 의미하게 됩니다.(` 표시는 작은 따옴표가 아니라 숫자 1의 왼쪽에 있습니다.)

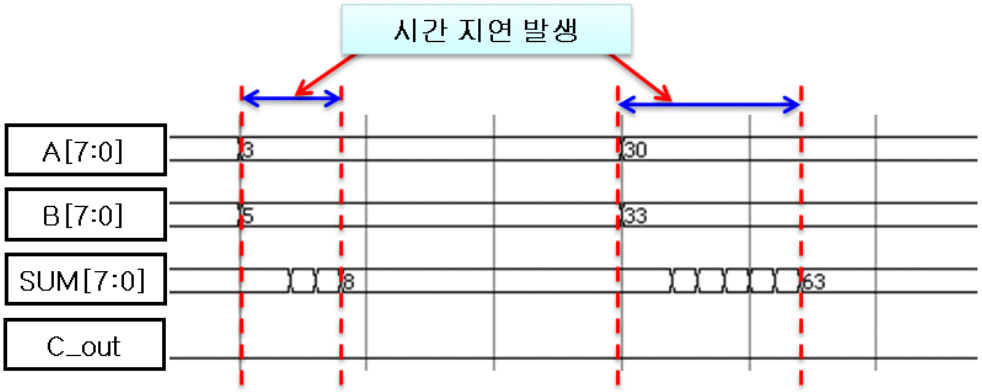
지금까지는 설계한 블록들을 시뮬레이션 할 때 입출력의 시간 지연을 고려하지 않았습니다. 설계된 byte_adder 에 타이밍 파라미터를 추가하여 시간 지연이 얼마나 발생하는지 시뮬레이션 해보겠습니다. 설계된 블록이 구현되는 환경에 따라 파라미터 값은 달라지겠지만 Net 지연은 발생하지 않으며 Gate 지연은 아래와 같이 대략적인 지연 값을 가정하여 인가하도록 합니다.

Logic Gate	Delay Time
AND	1ns
OR	1ns
XOR	2ns
INV	1ns

Gate 지연 값 인가는 아래와 같이 할 수 있습니다. #뒤에 숫자가 시간 지연 값을 나타냅니다.

```
xor     #2   (SUM, A, B);
and     #1   (C_out, A, B);
or      #1   (C_out, wire_c1, wire_c2);
```

시간 지연을 추가하여 시뮬레이션 하면 아래와 같이 출력이 나오는데 시간 지연이 발생한 것을 볼 수 있습니다. 이 시간 지연은 시뮬레이션 상에서만 나타나며 실제 구현 시 시간 지연 구문은 무시되게 됩니다. 시간 지연은 구현 환경에 따라 다르게 나타나게 됩니다.



실습 과제

1. Half Adder, Full Adder, Byte Adder를 계층구조로 설계하고 각각 시뮬레이션 합니다. (2-1, 2-2, 2-3 장 참조)

2. 게이트 시간 지연을 추가하여 시뮬레이션 결과를 확인합니다. (2-4장 참조)