디지털 논리회로1

32 – Bit Ripple Carry Adder in verilog

제출일자: 2018년 06월 15일 (금)

학 과: 컴퓨터정보공학부

담당교수: 이준환 교수님

학 번: 2017202087

성 명: 홍 세 정

1. Report

◼ Problem statement (5%) (문제 설명)

input A, B, Cin 각32개와 output S, Cout 각 2개를 이용하여 32-bit ripple carry adder 를 만든다. 1 bit adder을 먼저 만들어서 Cout에서 Cin으로 넣어주어 32 bit adder을 verilog를 이용하여 만드는 것이다. 하드웨어 기술 언어인 verilog로 32-bit RCA를 설계하고 검증, 구현 등을 사용할 수 있다.

◼ Descriptions for inputs and outputs (5%) (입력 출력 설명)

1개의 full adder에는 A, B, Cin 총 3개의 input과 S, Cout 총 2개의 output이 사용된다. n자리의 2개의 2진수를 An, Bn, 이라 하고, n-1 자리에서의 자리올림수로 올라 온 수를 Cout이라 한다. 

32개의 Full adder에 각 A, B가 들어가고 S가 나오게 된다. 처음 full adder에 Cin이 들어가고 나온 출력을 c를 다음 full adder의 Cin으로 넣어주어 최종적으로 나오는 32번째 Full adder의 출력은 Cout이 된다.

◼ Gate-level design of a full adder (5%) (full adder 설계)



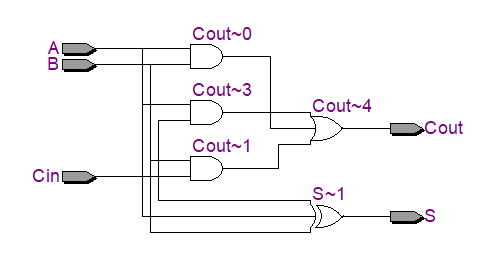
1 bit adder는 다음과 같은 회로도로 나타낼 수 있다.

여기서 output 논리식은

Sn = An ⊕ Bn ⊕ Cn-1 ,

Cn = (An\*Bn) + (An\*Cin) + (Bn\*Cin) 이다.

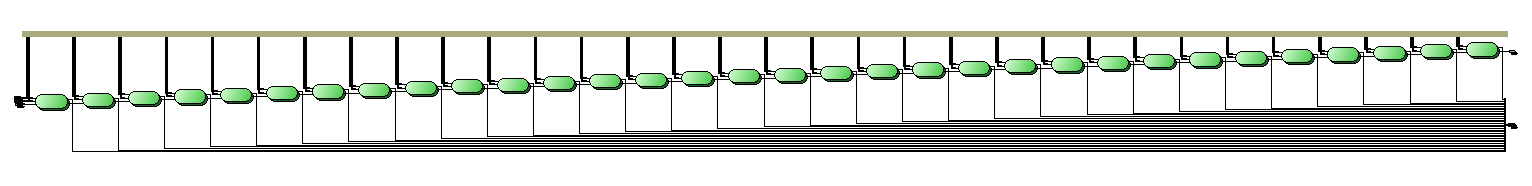
다음과 같이 1bit full adder의 진리표를 만들 수 있다.



◼ Block diagram of your RCA (5%) (RCA 표)



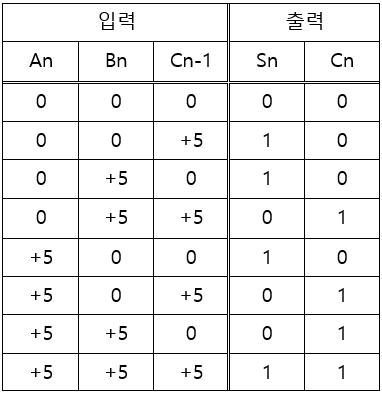
1bit Full adder을 32개 연결한 형태로 32 bit Full adder을 구성하여서 다음과 같은 Block diagram을 나타낼 수 있다.



◼ Descriptions of the operation of your RCA with few examples (15%) (예제 설명)

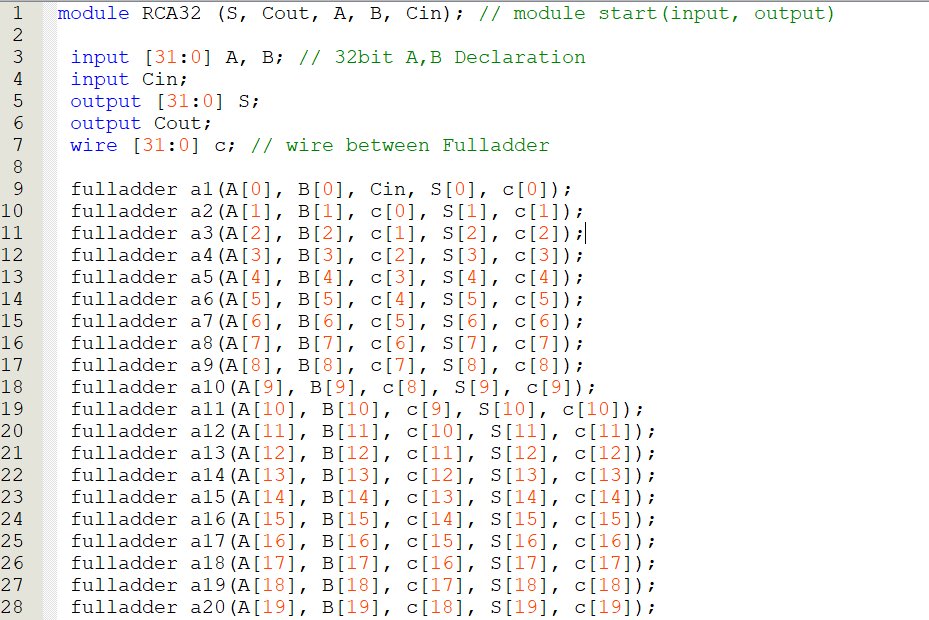
* Why do you need Cin? (Cin 필요한 이유)

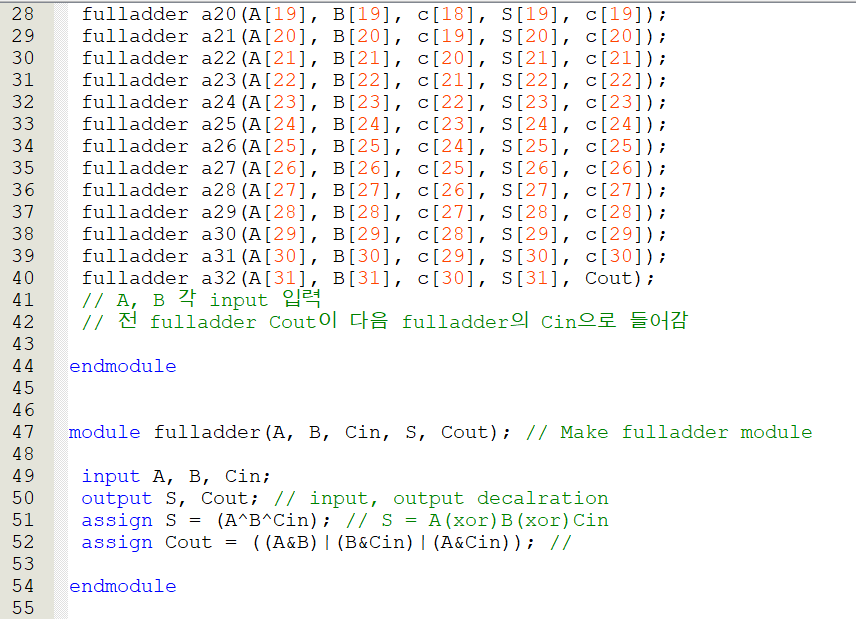
Fulladder에서 input이 총 3개 A, B, Cin이다. 32RCA는 전 fulladder Cout에서 나와 다음 Cin으로 들어간다. 여기서 Cin이 없어지면 fulladder에 들어가는 Cin이 없어지기 때문에 각 1bit fulladder 32개를 연결해주는 wire가 없다. Cin이 없어지면 1bit 32개가 따로 떨어져있는 형태이기 때문에 32bit가 되지 않는다. 그러므로 RCA를 구현할 때 무조건 Cin이 있어야한다.

* What is the value of Cin? (Cin의 값)

Cin의 값에 따라 S의 값과 Cout값이 나오게 된다.

◼ Code analysis (line-by-line code description is needed) (15%) (코드 분석)





1. Module RCA32 start 32bit ripple carry adder 설계하는 module

2~7. Input, output 선언 fulladder사이의 연결을 wire라 선언.

9~40. 32개의 fulladder input과 output 입력 다음 adder에 Cin이 그 전의 Cout 입력.

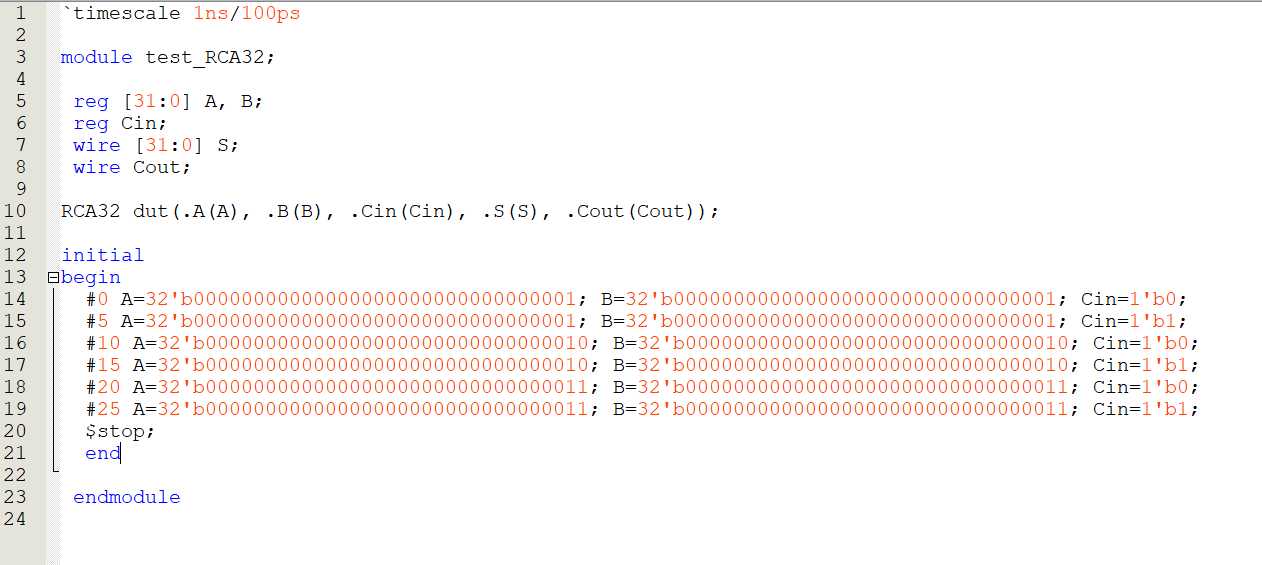
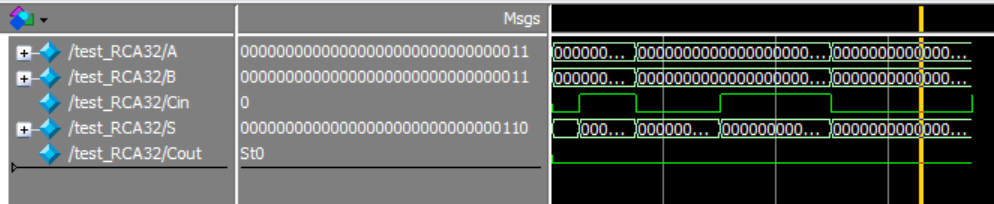
44. RCA32 module을 끝냄.

47. fulladder module 가산기 생성하는 module start

49~5. input A,B,Cin output S, Cout 선언

51~52. S와 Cout을 선언

54. fulladder module finish

◼ Verification strategy & corresponding examples with explanation (50%) (검증 전략 및 설명과 해당 예제)

#0 A=1, B=1 , Cin=1’b0; #5 A=1, B=1 , Cin=1’b1; // A와 B를 고정으로 두고 Cin만 바꿔서 Testbench가 작동하는지 확인할 수 있다.

#10 A=1, B=1 , Cin=1’b0; #15 A=1, B=1 , Cin=1’b1;

#20 A=1, B=1 , Cin=1’b0; #25 A=1, B=1 , Cin=1’b1; // 1줄과 마찬가지로 #20일 때 #25일 때 바뀌게 된다.