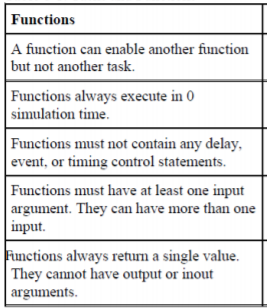
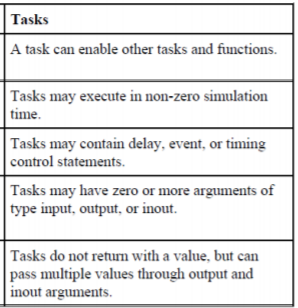
**CH.8**

펑션은 조합적, 딜레이 없음 , 1개의 리턴값

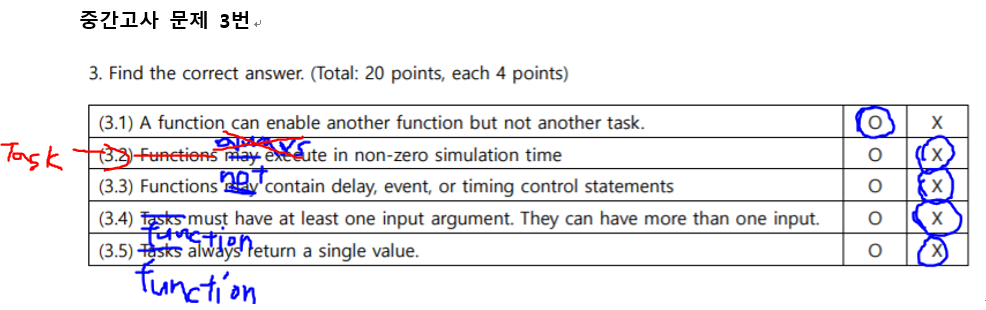
태스크는 순차적, 딜레이 사용가능, 여러 리턴값 가능 , 태스크는 또다른 태스크 불러오기 가능

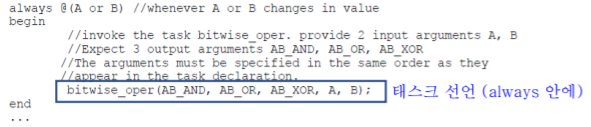
펑션 , 태스크 둘다 모듈 안에 정의되어야 함 (local)

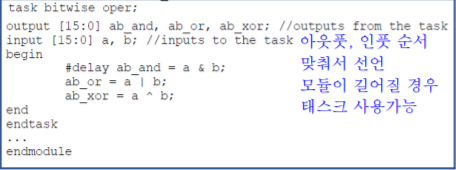
**태스크**

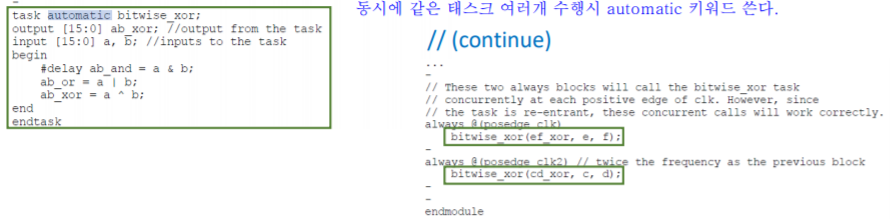


**중간고사 문제 3번**

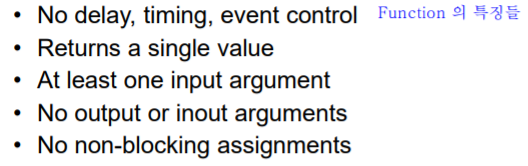


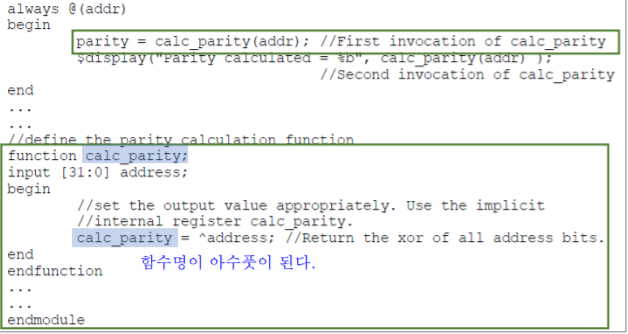


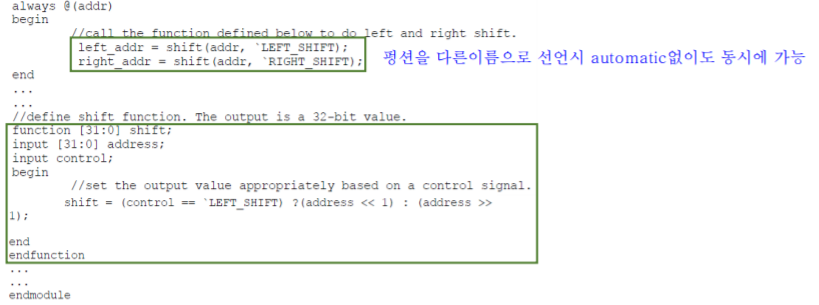


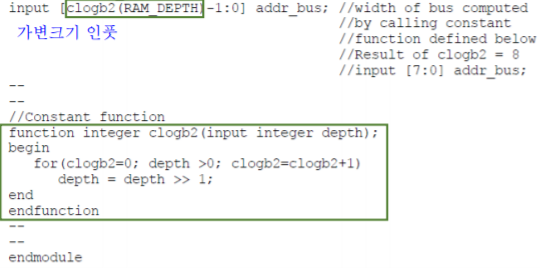


**펑션**

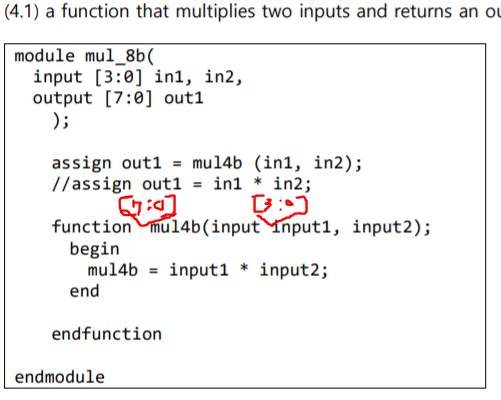
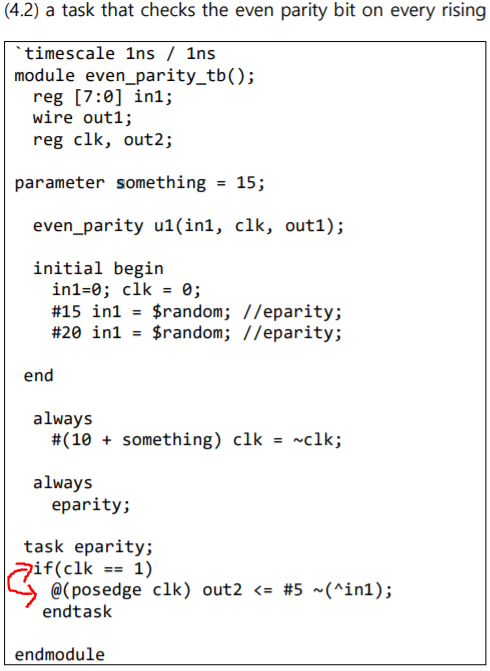




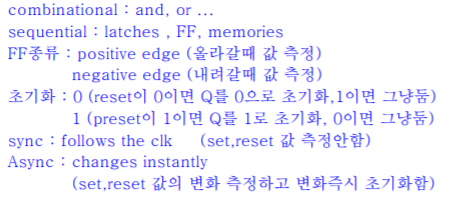


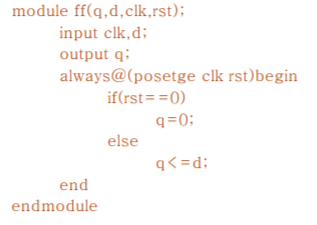


**중간고사 문제 4번**

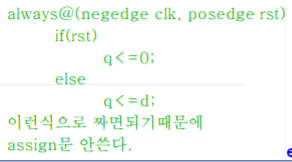


**플립플롭 (종류별로 짜는게 중간고사 5번문제)**

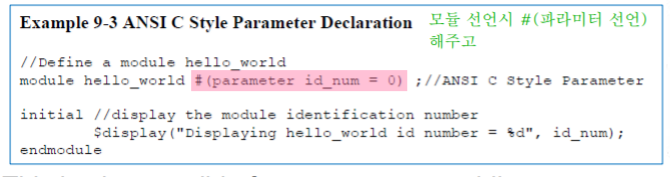


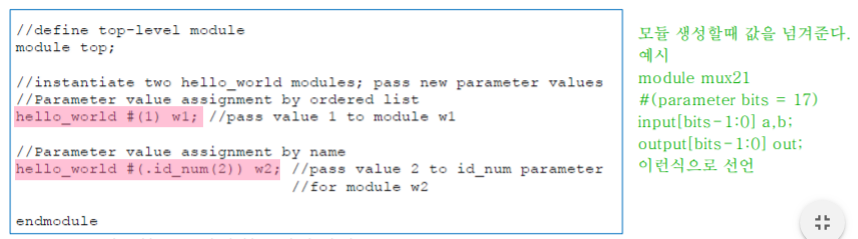


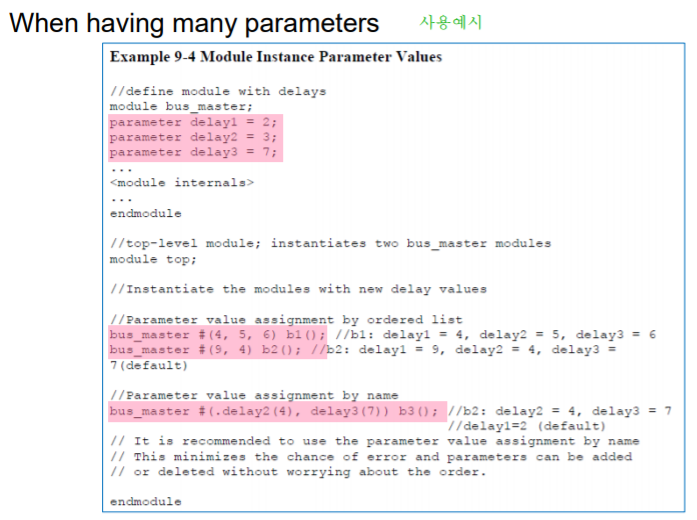
**CH.9**

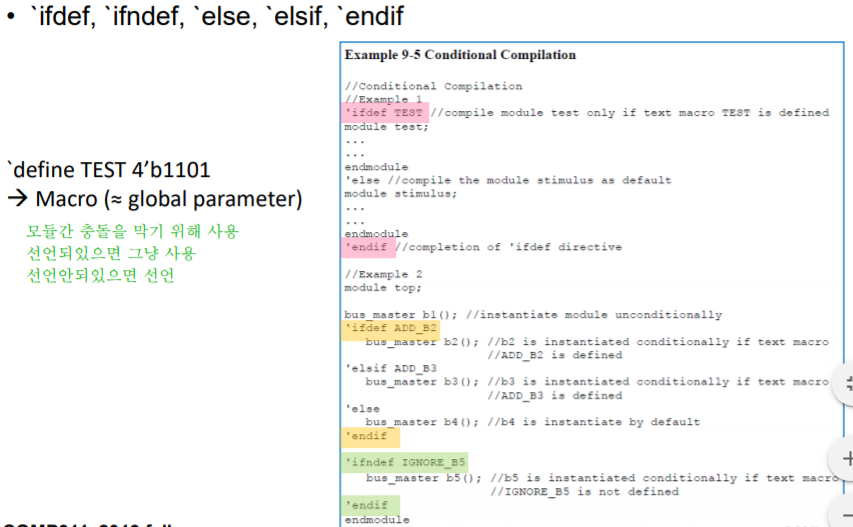


오버라이딩 : 강제로 값을 넣어줄 때 사용

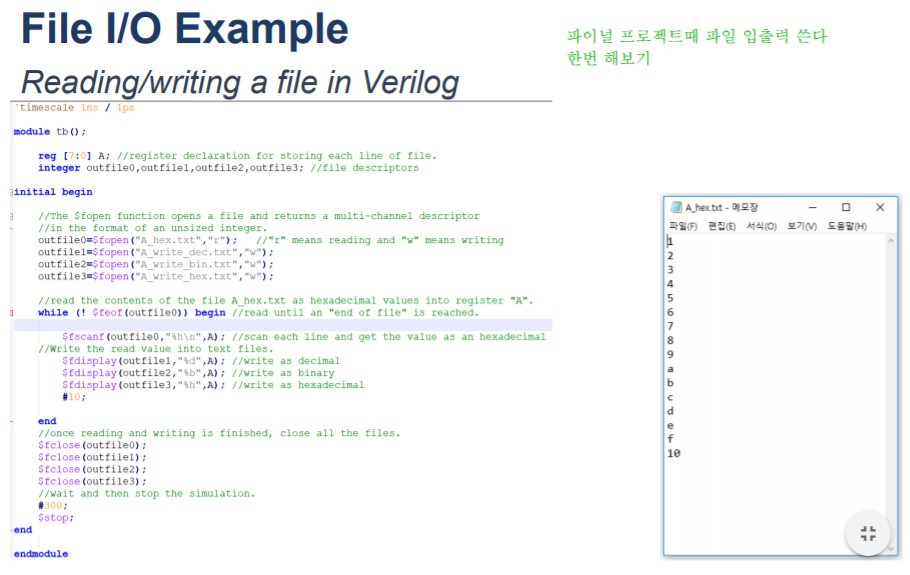




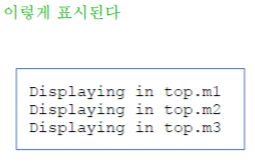




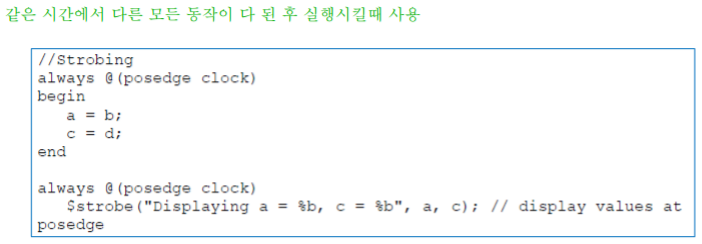
파일입출력 (9장에서 이밑에부턴 별로안중요)

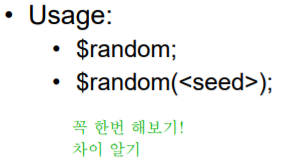


Hierarchy 는 %m으로 표시할 수 있다.



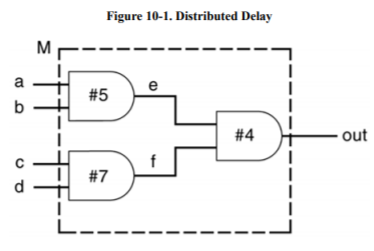
Strobing

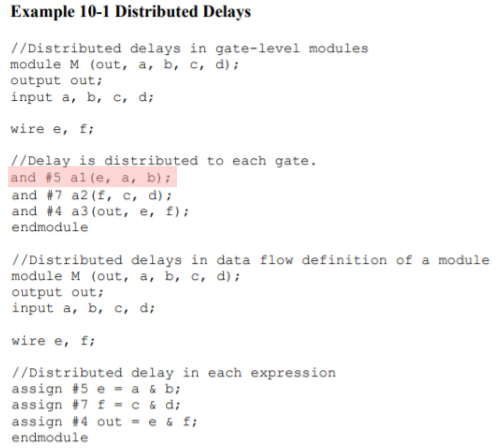


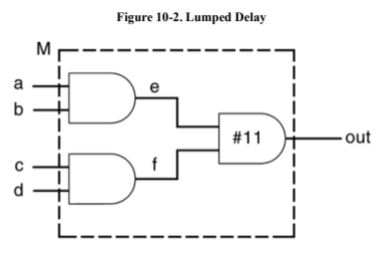


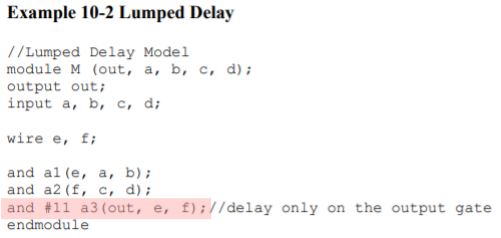
**CH.10**

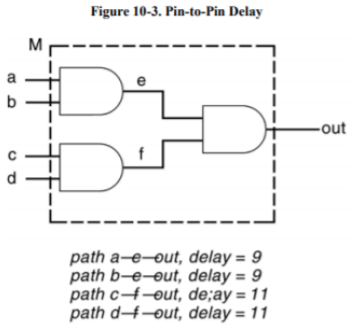
**딜레이 3가지 종류**

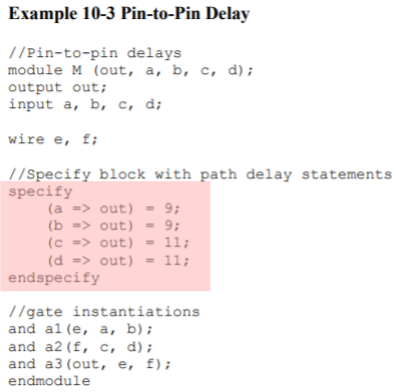


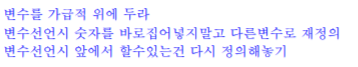


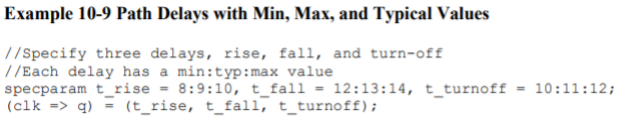






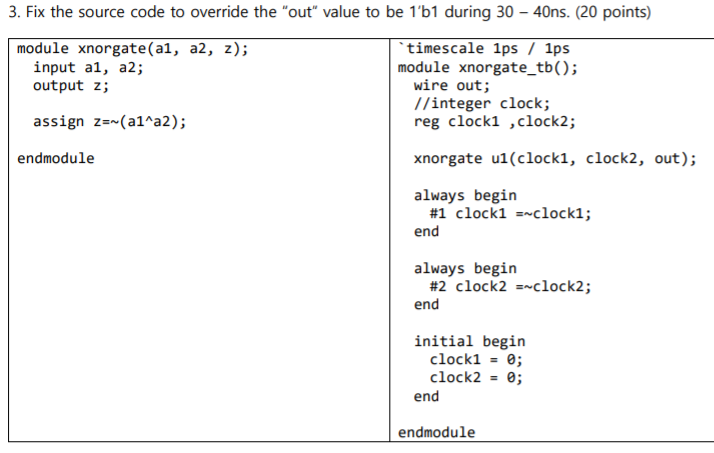


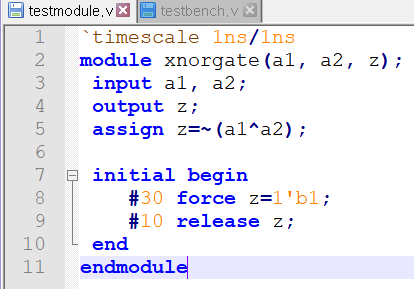




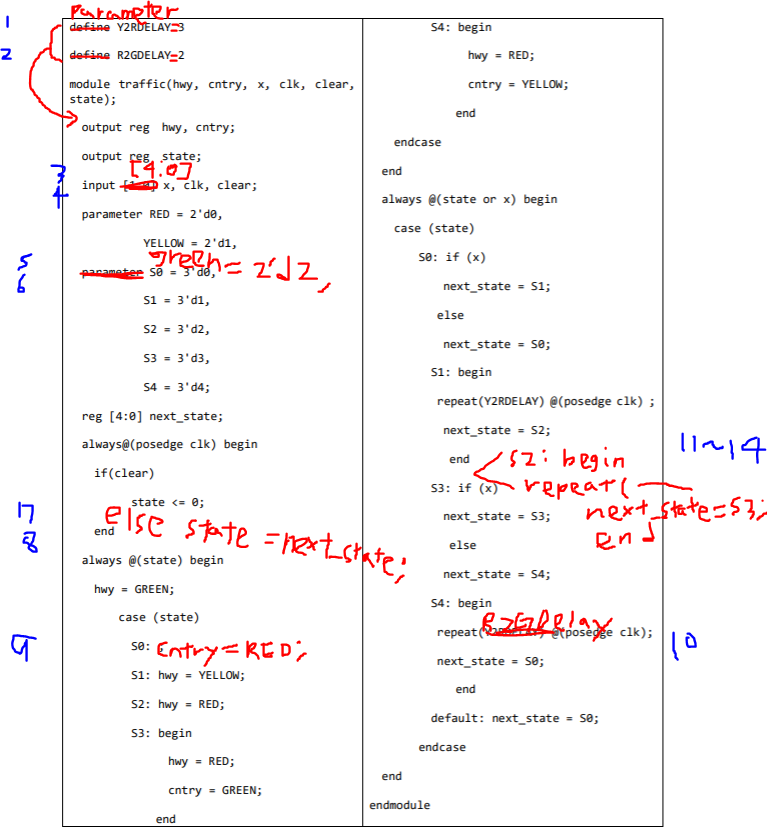
**퀴즈 2 문제**

**3번 (force , release 활용하는 문제)**

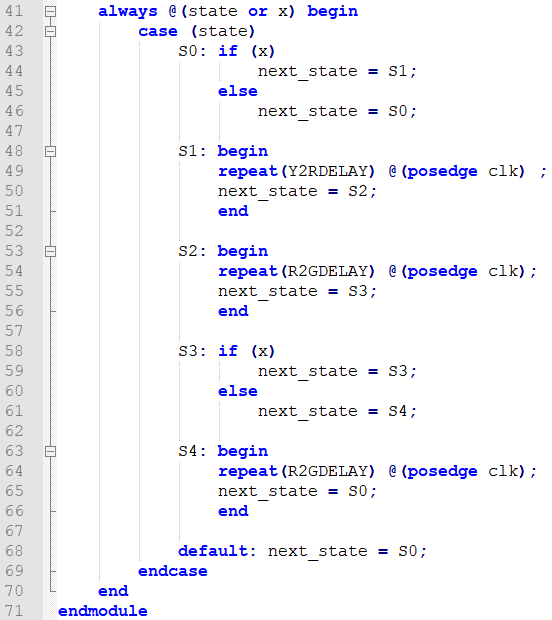
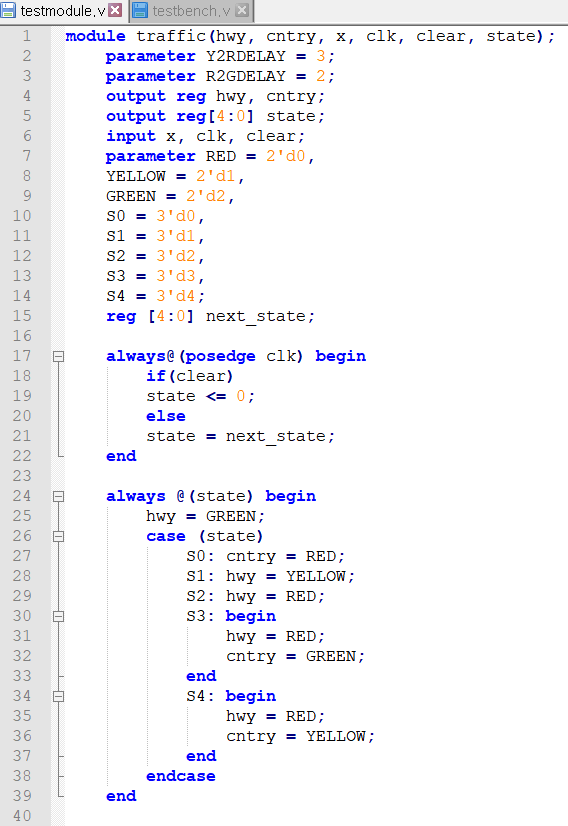




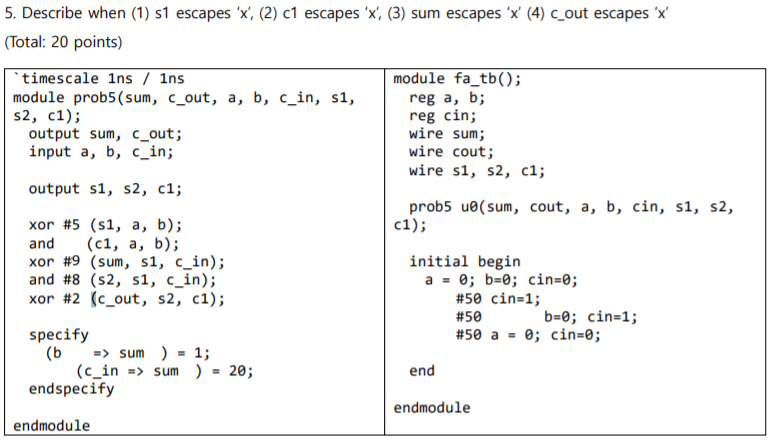
**4번**

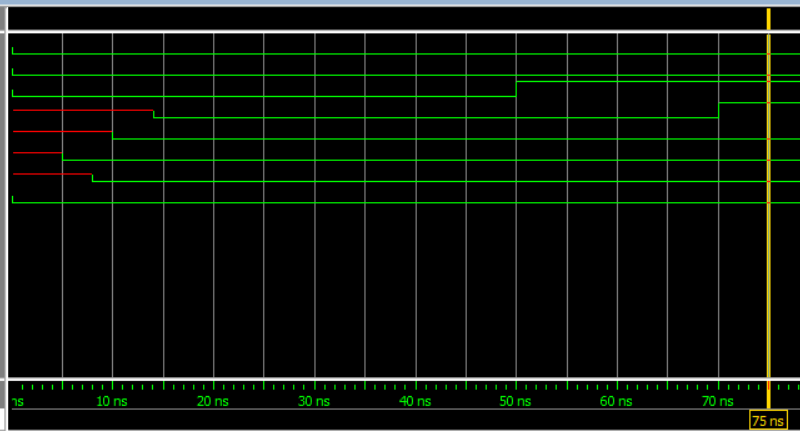


1. **베릴로그에는 디파인 안쓴다!! -> parameter A = 1; 이런식으로 상수 선언**
2. **인풋 아웃풋 크기 확인**
3. **변수 잘 보고 빠진거 없나 확인**
4. **If else나 case에서 빼먹은 경우 없나 확인**
5. **변수자리에 알맞는 변수가 있는지 확인 (다른변수 넣어놓을수도 있음)**
6. **나머지는 코드설명 잘읽고 코드를 잘보면서 찾아야한다.**
7. **여긴 없는데 혹시 코드안에 #2 이런식으로 시간 지정해주면 타임스케일 꼭 써야함**



**5번**





**S1 = 5ns**

**S2 = 8ns (and 게이트라서 입력하나라도 0이면 0이 바로나옴)**

**C1 = 0 ns**

**Sum = 14ns**

**Cout = 10ns (8+2)**

**6번 synthesis를 고려할 때 루프문에서 어떤 for와 while문은 조합적인데 왜 어떤건 안될까?**

**루프문에서 특정 조건에 의해 멈추는 루프문은 루프가 언제 끝날지 정확히 알수없기 때문에 조합적이지 않다. 하지만 루프의 횟수가 정해져 있는 경우에는 조합적이 된다.**

**이를 해결하려면 (루프문을 조합적으로 하려면)**

**루프문에서 반복 횟수를 확실히 하기 <- 이게 루프문에서 제일 중요 이거만 적어도 될거같음**

**그밖에 조합적으로 코딩하려면 지켜야할것들**

**X와 Z를 모두 고려하기**

**괄호를 잘 사용하기**

**같은변수에 여러 assign문은 사용하지 말기**

**If else나 case문 명시적으로 사용하기(모든 경우 고려)**