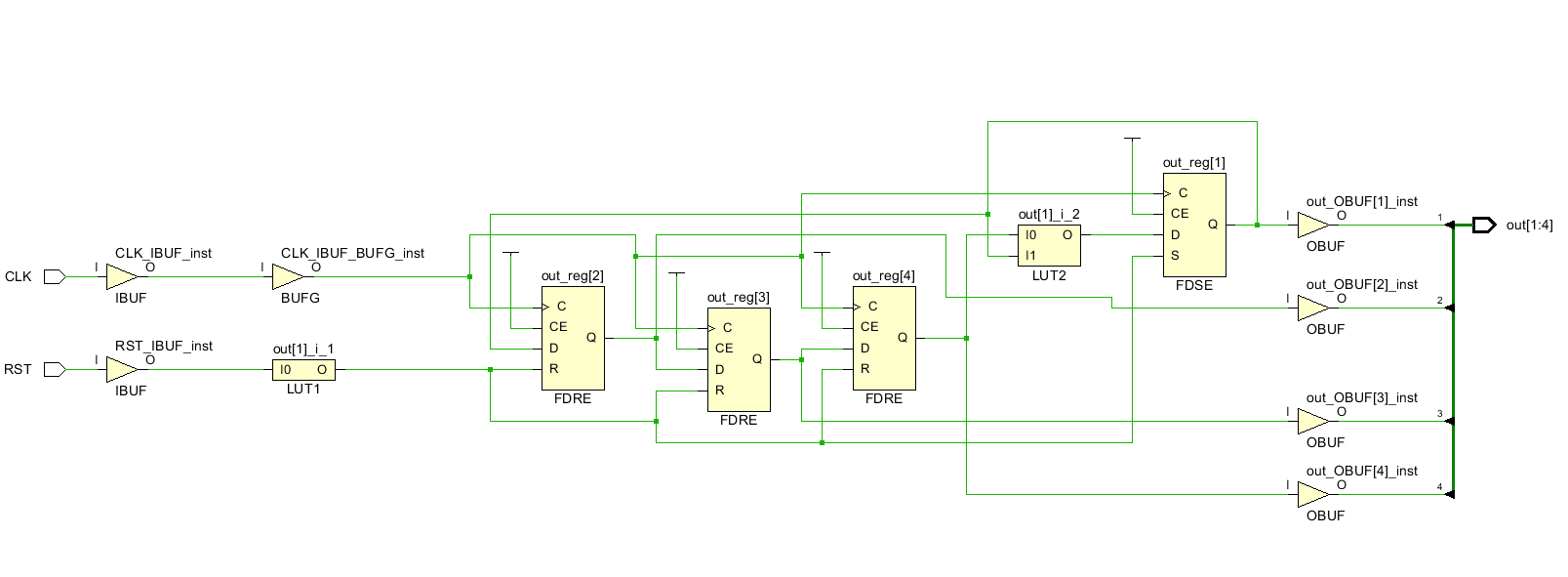
**Lab1.**

# **Homework1. INVERTER1**

**Verilog Code / 주석**



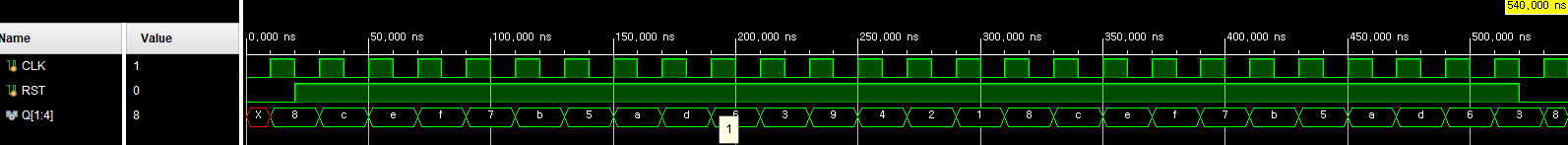
**Synthesis Report**

****

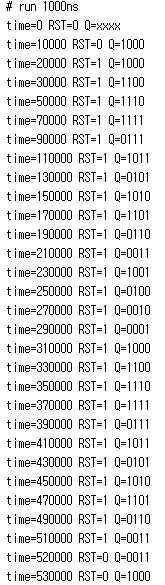
**Test Bench Code**

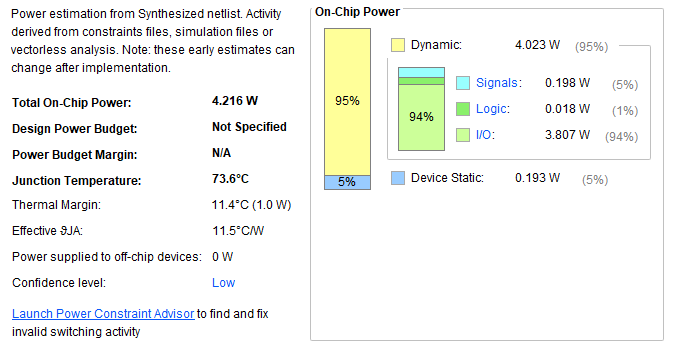


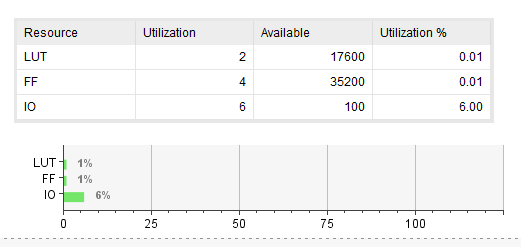
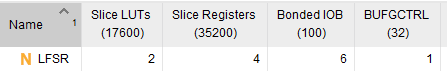
**Simulation Result**

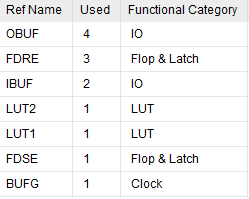


$monitor









|  |
| --- |
| **Discussion** |
| . - Verilog Coding을 시작하기 전 작성한 Block Diagram    - 작성한 Verilog Module에 대한 설명  LFSR 모듈은 입력으로 CLK, RST가 되도록 설정하였고 출력으로 4bit의 out이 나오도록 설계하였다.  이 모듈은 랜덤한 패턴의 출력을 발생시킬 수 있도록 하는 모듈이다.  Xor을 이용하여 Q1과 Q4를 비교하고 이 값을 한 비트씩 채우면서 랜덤한 값이 나오게 되는 원리이다.  - 작성한 Code가 어떤 동작을 하는지? 왜 그런 동작을 하는지?  그림과 같이 출력의 비트 순서를 매칭하기 위하여 [1:4]out으로 출력을 설정하였다.  세부적으로 살펴보면 먼저always문을 사용하여 CLK가 상승에지일 경우 출력 값에 변화가 발생하도록 설계하였다.  만약 RST가 0이면 값이 4’b1000으로 리셋 되도록 하였고 1일 경우 오른쪽으로 1비트씩 시프트 한다.  나머지 왼쪽 한 비트는 Q1과 Q4를 xor하여 넣어준다.  위 과정을 통하여 랜덤하게 테스트 패턴을 생성할 수 있도록 하였다.  - 작성한 Test Bench Code가 어떤 동작을 하는지?  testbench에서는 위 작성한 코드의 모듈을 불러오고 서로 연결시켜주는 동작을 하였고 이를 바탕으로 임의의 값에 정상적으로 동작하는지 확인하였다.  이 중 Q와 out의 이름을 다르게 설정하여 .out(Q)로 Q로 출력이 나오도록 연결해주었다.  test값은 20ns 이후 RST를 1로 바꾸고 500ns이후 RST를 0으로 설정하였다.  그 후 20ns이후 벤치를 종료하도록 설정하였다.  - Simulation 파형이 왜 그렇게 나온것인지?  먼저 입력이 들어오기 전인 20ns 이전에는 RST가 0이므로 1000으로 초기화된다.  이후 RST가 1이 되어 값이 shift되고 xor에 의해 값이 하나 추가되면서 값이 변경되었다.  500ns이 후에는 RST를 0으로 설정하여 다시 출력이 1000으로 초기화 되었고 시뮬레이션이 종료되었다.  - 작성한 Code가 잘 동작하지 않는다면 어디서 문제가 발생한 것인지? 문제가 발생하지 않았다.  - 오류를 해결했다면, 어떤 부분에서 해결을 했고, 왜 오류가 발생하였는지? 문제가 발생하지 않았다. |

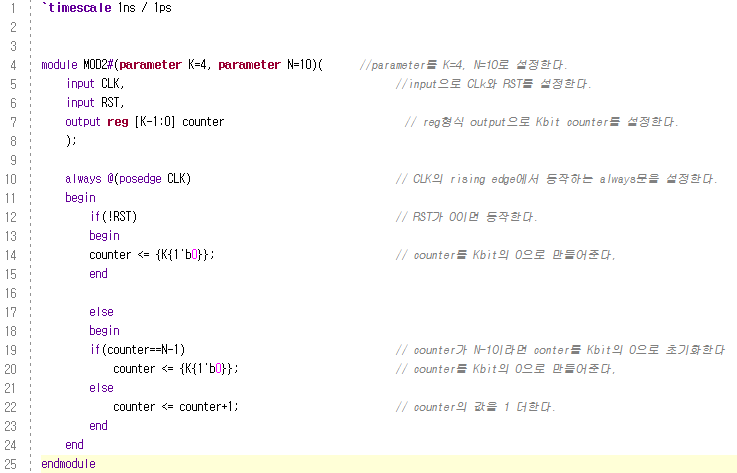
# **Homework2. INVERTER2**

**Verilog Code / 주석**

K=3, N=8

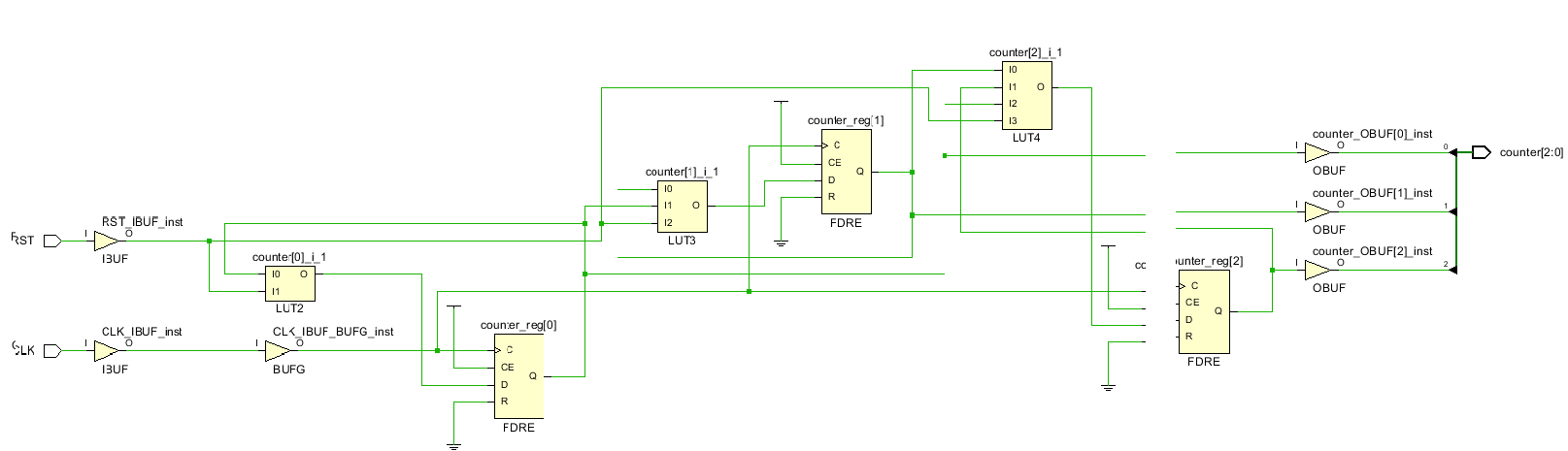


K=4, N=10

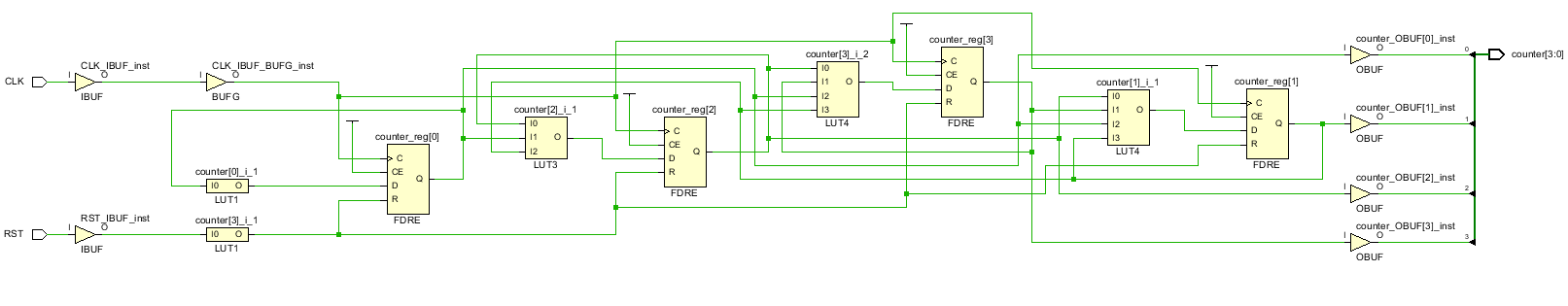


**Synthesis Report**

K=3, N=8

****

K=4, N=10

****

**Test Bench Code**

K=3, N=8

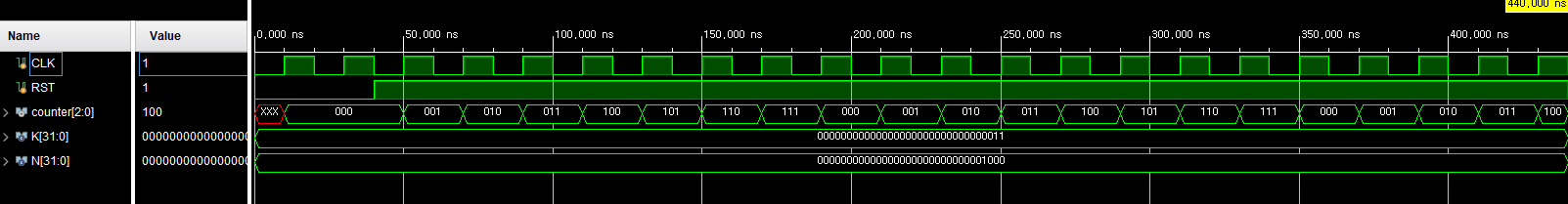


K=4, N=10

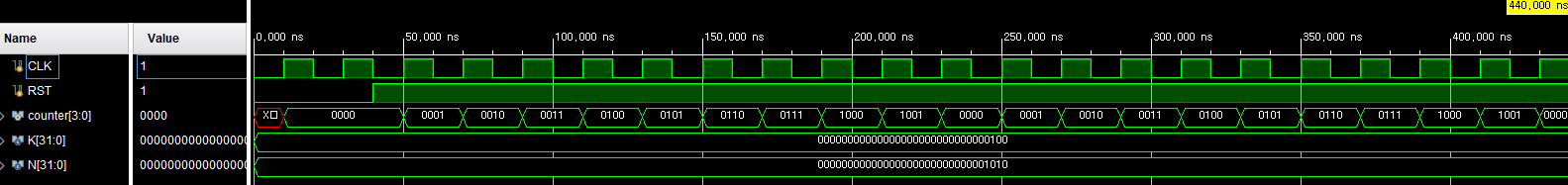


**Simulation Result**

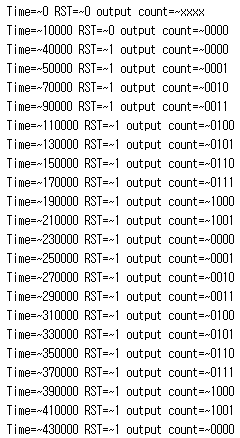
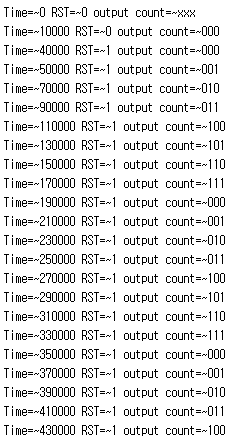
K=3, N=8



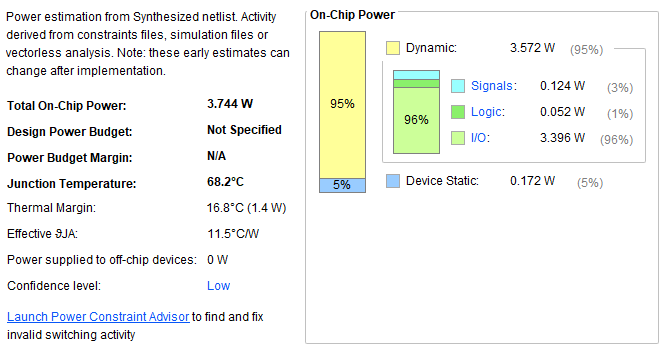
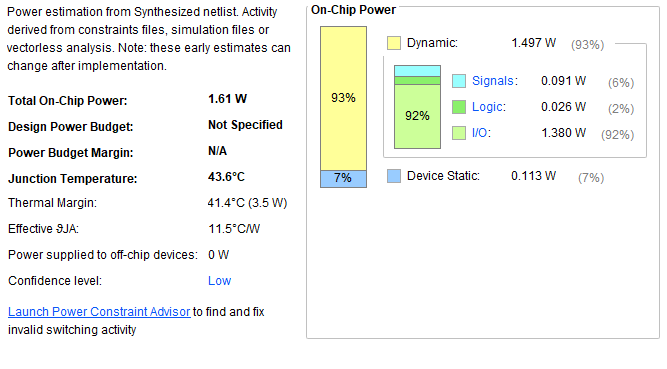
K=4, N=10



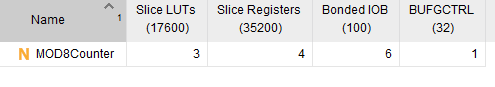
K=3, N=8 K=4, N=10

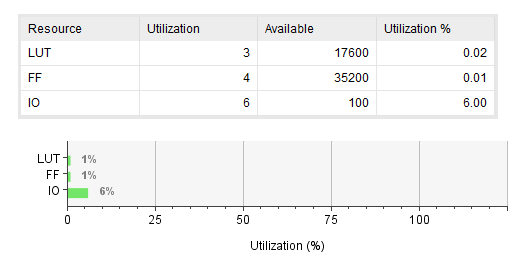
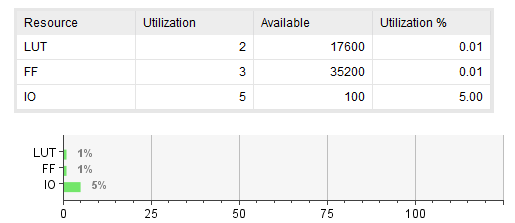


K=3, N=8 K=4, N=10

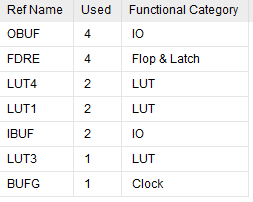
****

**텍스트이(가) 표시된 사진

자동 생성된 설명**

****

**테이블이(가) 표시된 사진

자동 생성된 설명**

|  |
| --- |
| **Discussion** |
| . - Verilog Coding을 시작하기 전 작성한 Block Diagram  s  - 작성한 Verilog Module에 대한 설명  MOD 모듈은 RST신호가 들어오기 전까지 0으로 유지되다가 0이 되면 카운터로 동작하는 모듈이다.  K개의 비트에서 0~N-1까지 N번의 카운터 이후 마지막 state가 되면 0으로 초기화되어 처음부터 카운팅하는 모듈이다.  - 작성한 Code가 어떤 동작을 하는지? 왜 그런 동작을 하는지?  먼저 always 문에 rising edge가 감지되면 모듈이 동작한다.  RST가 0이라면 카운터가 동작하지 않는 것 이므로 0으로 초기화 하였다.  RST가 1이고 State가 마지막이라면 모든 동작을 마친 것 이므로 다시 0으로 돌아가도록 설정하였다.  만약 이 경우에 해당되지 않는다면 1씩 증가하는 카운터로서 동작하도록 하였다.  - 작성한 Test Bench Code가 어떤 동작을 하는지?  testbench에서는 RST가 40ns이후 1이 되면 어떻게 동작하는지 나타낼 수 있도록 하였다.  CLK의 주기는 20ns로 설정하였고 400ns 이후에 시뮬레이션이 끝나도록 설정하였다.  - Simulation 파형이 왜 그렇게 나온것인지?  40ns이후 RST가 1이 되고 CLK의 상승에지에 감지되어 카운터가 증가하는 모습을 확인할 수 있다.  만약 마지막 State에 도달하면 다시 0으로 돌아가도록 설계하였으므로 원래대로 돌아가는 모습을 확인할 수 있었다.  K=3, N=8일 때는 0~7의 범위에서 카운팅이 진행되는 것을 확인할 수 있었고 K=4, N=10일 때에는 0~9의 범위에서 카운팅이 진행되는 것을 확인할 수 있었다.  - 작성한 Code가 잘 동작하지 않는다면 어디서 문제가 발생한 것인지? 문제없이 동작하였다.  - 오류를 해결했다면, 어떤 부분에서 해결을 했고, 왜 오류가 발생하였는지? 문제없이 동작하였다. |