**기초디지털실험**

**Week1\_Environment\_Setting**

**결과레포트**

****

**학정번호 : EEE3313-07-00**

**학번 : 2018142173**

**이름 : 이 찬**

**작성일자 : 2022-03-13**

**목차**

**1. 실험 목적**

**2. 실험 이론**

* **Test Benchmark Description for simulation**
* **비바도에서 simulation을 돌렸을 때 나오는 결과를 서술하라**
* **Module Define**
* **작성한 Module에 대한 정의**
* **Module Instantiation and Module connection**
* **모듈의 인스턴스화와 모듈 연결을 정의하라**
* **Pin Assignment and Connection to Top Module**

**->**

**3. Waveform simulation results**

* **Logic Gate Circuit Design: gate evaluation through simulation**
* **Verilog Circuit Design: addition operation evaluation through simulation**
* **Result analysis**

**4. FPGA results**

* **Logic Gate Circuit Design: show gate operation using LED**
* **Verilog Circuit Design: show 2-bit addition operation results using LED**
* **Result analysis**

**5. 토의**

**1. 실험 목적**

**- 기초적인 디지털 회로와 실험 환경을 이해한다.**

**- AND, OR, XOR 과 같은 디지털회로를 Verliog HDL을 통해 실행한다.**

**- 디지털 설계 과정을 이해한다.**

**2. 실험 이론**

**1) Test Benchmark Description for simulation**

**벤치마크란 성능을 테스트하여 이를 수치화하는 것을 말한다.**

**비바도 사용법**

**1. Module의 입출력 포트를 지정하는 design sources**

**2. constraints**

**3. simulation sources**

**절차**

**1. 모듈 생성하기 위한 design source 생성 ( logic\_gate )**

**2. waveform simulation을 하기 위해 simulation sources 생성 (testbench)**

**3. 성능 측정**

**4. 보드에 나타내기 위해서 constraints 생성 후 보드와 연결할 수 있는 코드 저장**

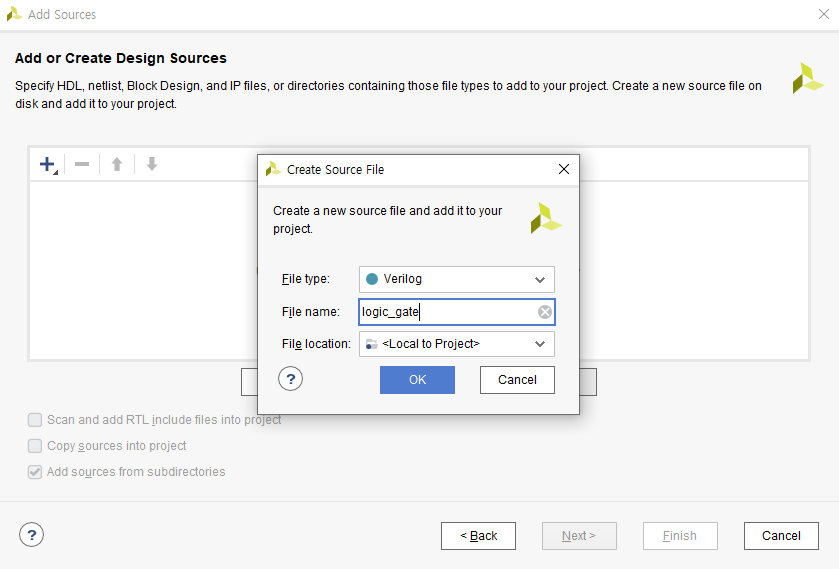
**5. 실제 보드의 핀과 연결하기 위해서 topModule파일 생성**

**6. 보드 연결 후 분석**

텍스트이(가) 표시된 사진

자동 생성된 설명

**새로운 프로젝트 생성**



**Design sources 생성 및 logic\_gate로 파일 이름 정의**

텍스트이(가) 표시된 사진

자동 생성된 설명

**Module logis\_gate : logic\_gate 라는 이름의 Module을 생성한다.**

**Input [1:0] input\_a : input 신호를 생성하기 위해서 input\_a라는 변수를 통해 2-bit signal을 지정한다.**

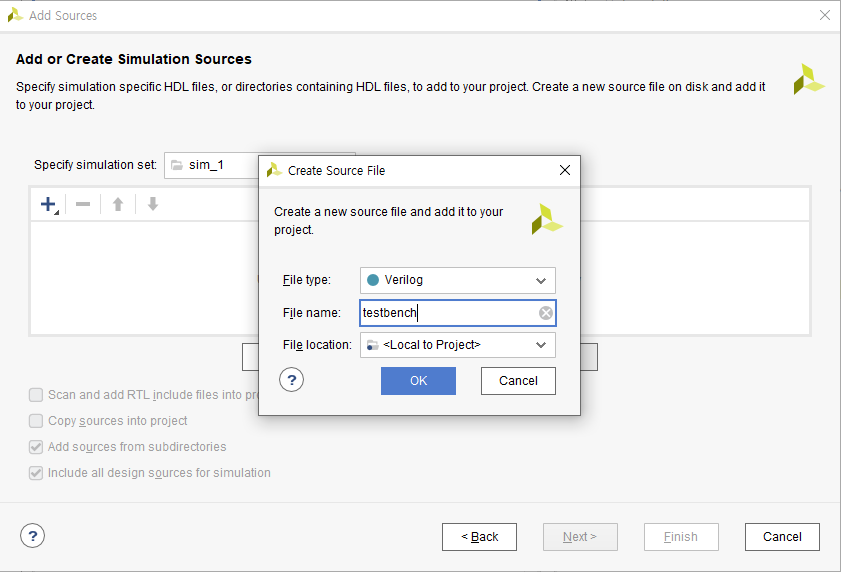
**Output [3:0] result : output 신호를 생성하기 위해서 result라는 변수를 통해 3-bit 시그널을 지정한다.**

**Assign result = input\_a + 3’b111; : module에서 선언한 input\_a 와 3비트의 111의 값을 서로 더한 결과를 result에 할당한다.**

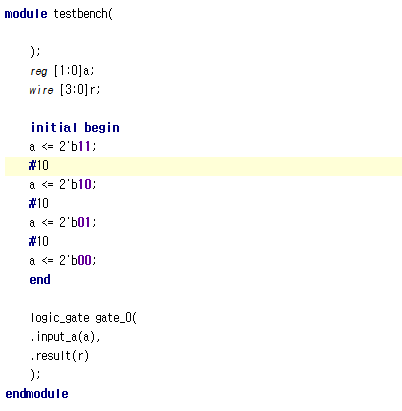
**따라서 input\_a가 가질 수 있는 경우의 수는 00,01,10,11이 되며 각 경우와 3비트의 111값을 더한 값이 result에 나타나게 될 것이다.**

**결과 표**

|  |  |  |
| --- | --- | --- |
| **Input\_a** | **3’b111** | **Result** |
| **00** | **111** | **0111** |
| **01** | **111** | **1000** |
| **10** | **111** | **1001** |
| **11** | **111** | **1010** |



**생성한 module을 simulation 하기 위해 testbench 파일 생성**



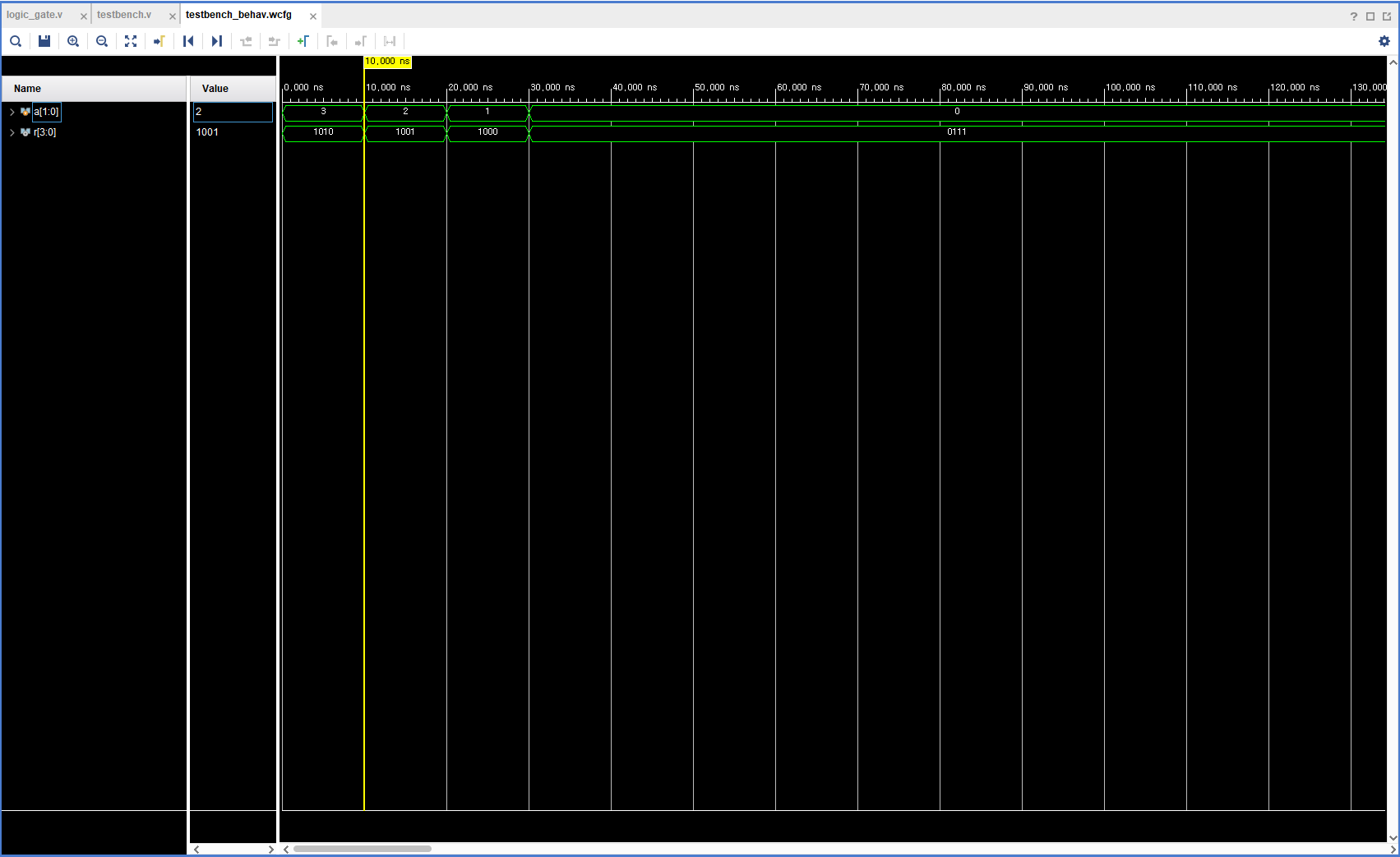
**Module testbench(); : testbench라는 이름의 모듈을 생성한다.**

**reg [1:0]a; : reg type의 변수는 값을 저장할 수 있다. 2-bit signal의 값을 가진 변수 a를 선언한다.**

**wire [3:0]r; : wire type의 변수는 값을 저장할 수 없다. 3-bit signal의 값을 가진 변수 r을 선언한다.**

**initial begin ~ end : testbench에서 simulation을 작동할 때 한 번만 작동하게 하기 위해서 initial 블록을 사용한다. Initial begin이 시작점이고 end는 종료점이다. 변수 a가 가질 수 있는 경우의 수가 총 4가지(00,01,10,11)이므로 각 경우를 시간차에 따라서 a값에 할당한다. 2-bit signal을 이용해서 a가 11부터 00까지 10초 간격으로 a값을 변경하면서 할당하는 블록을 생성하였다.**

**logic\_gate gate\_0( .input\_a(a), .result(r)) : 처음에 생성한 logic\_gate 모듈을 인스턴스화하여 모듈에서 정의된 input\_a와 result를 gate\_0이라는 인스턴스로 표현한다. 이때, testbench.v에서 선언한 a와 r을 각각 input\_a, result의 객체로 나타낸다.**



**a값이 변할 때마다 10ns 단위로 r값이 변경되는 것을 확인할 수 있다.**

* **0~10.000ns : a값이 11(3)일 때, r값이 1010**
* **10.000ns~20.000ns : a값이 10(2)일 때, r값이 1001**
* **20.000ns~30.000ns : a값이 01(1)일 때, r값이 1000**
* **30.000ns ~ : a값이 00(0)일 때, r값이 0111 로 변하는 것을 확인할 수 있다.**

**이는 처음에 제시한 결과표와 값이 동일하게 나타나는 것을 알 수 있다.**