

EE303 Lab1 Report

20180155 김준범

20160791 권용빈

I. Introduction

Lab1은 verilog를 사용하여 16가지의 연산을 수행할 수 있는 간단한 ALU를 구현하는 과제이다. Lab1을 통해 verilog의 기본적인 문법, 효율적인 Module 디자인 등의 사용법을 익히고 배우는 것이 가장 중요한 task였다고 생각한다.

II. Design

ALU라는 하나의 Module만으로 ALU를 구현하였다. 4bit짜리의 input의 관한 case문을 통하여 각 4bit 이진수에 상응하는 총 16가지의 연산을 수행하게 했다.

III. Implementation

ALU Module은 16bit input A,B와 4bit OPcode로 16bit output C와 1bit output Cout을 만들었다. 내재적으로 16bit register X와 1bit register Xout을 만들었고 코드의 마지막 부분에서 X를 C에, Xout을 Cout에 assign 해주었다.

OP code는 switch case 문을 이용해 해당하는 연산에 대응하였다.

Cout에 대해서는 ADD와 SUB 부분에서 조건문을 사용하여 Overflow의 유무를 판별한 뒤 값을 대입하였다.

IV. Evaluation

선택 명령 프롬프트

```
TEST      Ars-1 :
PASSED
TEST      Ars-2 :
PASSED
TEST      Ars-3 :
PASSED
TEST      Ars-4 :
PASSED
TEST      Rr-1  :
PASSED
TEST      Rr-2  :
PASSED
TEST      Lls-1 :
PASSED
TEST      Lls-2 :
PASSED
TEST      Als-1 :
PASSED
TEST      Als-2 :
PASSED
TEST      RI-1  :
PASSED
TEST      RI-2  :
PASSED
Passed = 50, Failed = 0
```

위 사진에서 볼 수 있듯이 50개의 test case를 모두 통과한 것을 볼 수 있다. 따라서 ALU

구현은 성공적이었다고 생각한다. 다만 test case만을 통과하기 위해 코딩을 한 듯한 부분들이 있어 그런 점에 있어서는 아쉬움이 남는다. 다음에는 더 정밀한 module을 디자인하기를 희망한다.

V. Discussion

Lab1 assignment는 별다른 문제 없이 순조롭게 진행되었다. Verilog 코드의 compile은 Icarus 프로그램을 사용하였다.

VI. Conclusion

이번 과제에서는 아주 기본적인 ALU를 디자인해보며 verilog 언어에 대한 이해를 조금 높였던 것 같다. 하지만 Register을 사용해 코드를 작성했기 때문에 wire만을 이용한 코드를 작성해보지 못한 것이 아쉬웠고 ADD와 SUB을 제외한 다른 연산에 대한 overflow를 고려해보지 않아서 그 부분도 기회가 되면 해보고 싶다.