|  |  |  |  |
| --- | --- | --- | --- |
| 디지털시스템실험 결과보고서 | | | |
| 이름 : 정성훈 | | | 학번 : 2019171009 |
| **실험제목** | Simple Computer 설계 및 구현 | | |
| **실험목표** | 1. Simple Computer의 Datapath를 설계한다. 2. Functional Unit의 기능을 이해하고 구현한다. 3. Register File의 기능을 이해하고 구현한다. | | |
| **실험결과** | | | |
| 1. Logical Circuit  1. 구현하고자 하는 논리회로  -4가지 논리 연산(AND, OR, XOR, NOT)을 수행하는 모듈을 만들고자 했다.  2. Verilog Code    Selector bit에 따라서 들어온 두 DATA의 AND, OR, XOR, NOT 연산을 수행하도록 하였다.  2. Register File  1. 구현하고자 하는 논리회로    위와 같이 Data와 Address를 받아서 read와 write가 가능한 register file을 만들고자 하였다.  2. Verilog Code      Daddr를 받아서 2to4 decoder를 이용해 Write신호가 들어왔을때 Daddr가 가르키는 register에 Data가 write되게 하였다.  3. Function Unit  1. 구현하고자 하는 논리회로    위의 Function Unit과 같이 DATA와 FS를 받아 해당하는 연산을 수행하고 연산결과를 출력하는 모듈을 만들고자 하였다.  2. Verilog Code    Logic Circuit과 ArithMetic Circuit에 data와 opcode를 인가해서 각각 data를 받은다음에, 2 to 1 MUX를 통해 최종적으로 out할 데이터를 선택했다.  4. Simple Computer  1. 구현하고자 하는 논리회로    지난 주차의 Control Unit과 이번 주차의 Datapath를 합쳐서 Simple Computer를 구현하고자 했다. CLK가 흐르면서 Control Unit 내의 Program Counter가 변하면서 Instruction이 정해진 순서대로 실행된다. Decode된 Instruction이 Control Word의 형태로 Datapath로 들어가 연산이 수행된 후 결과가 Data Memory에 저장된다.  2. Verilog Code    위 Block Diagram처럼 Control Unit, Datapath, Data memory 모듈에 wire를 연결했다.    위는 Testbench 코드다. 출력을 확인할 선을 연결한 뒤 clock 신호만 주었다.  3. 시뮬레이션 파형    위는 시뮬레이션 파형이다. | | | |
| **토의** | | 저번 주차의 Control Unit에 이어 이번 주차에 Datapath를 설계해 Simple Computer를 완성했다. Program counter값이 변하며 Instruction이 자동으로 실행되는 걸 보니 신기했다. 엄청나게 짧고 간단한 연산들을 실행하는데도 상당히 복잡한데, 실제 컴퓨터는 정말 많이 복잡하겠다고 생각했다. | |