**2024 Spring CSED311 Lab 2 Report**

Team ID: 67735

20220312 박준혁, 20220871 홍지우

**명예서약 (Honor Code)**

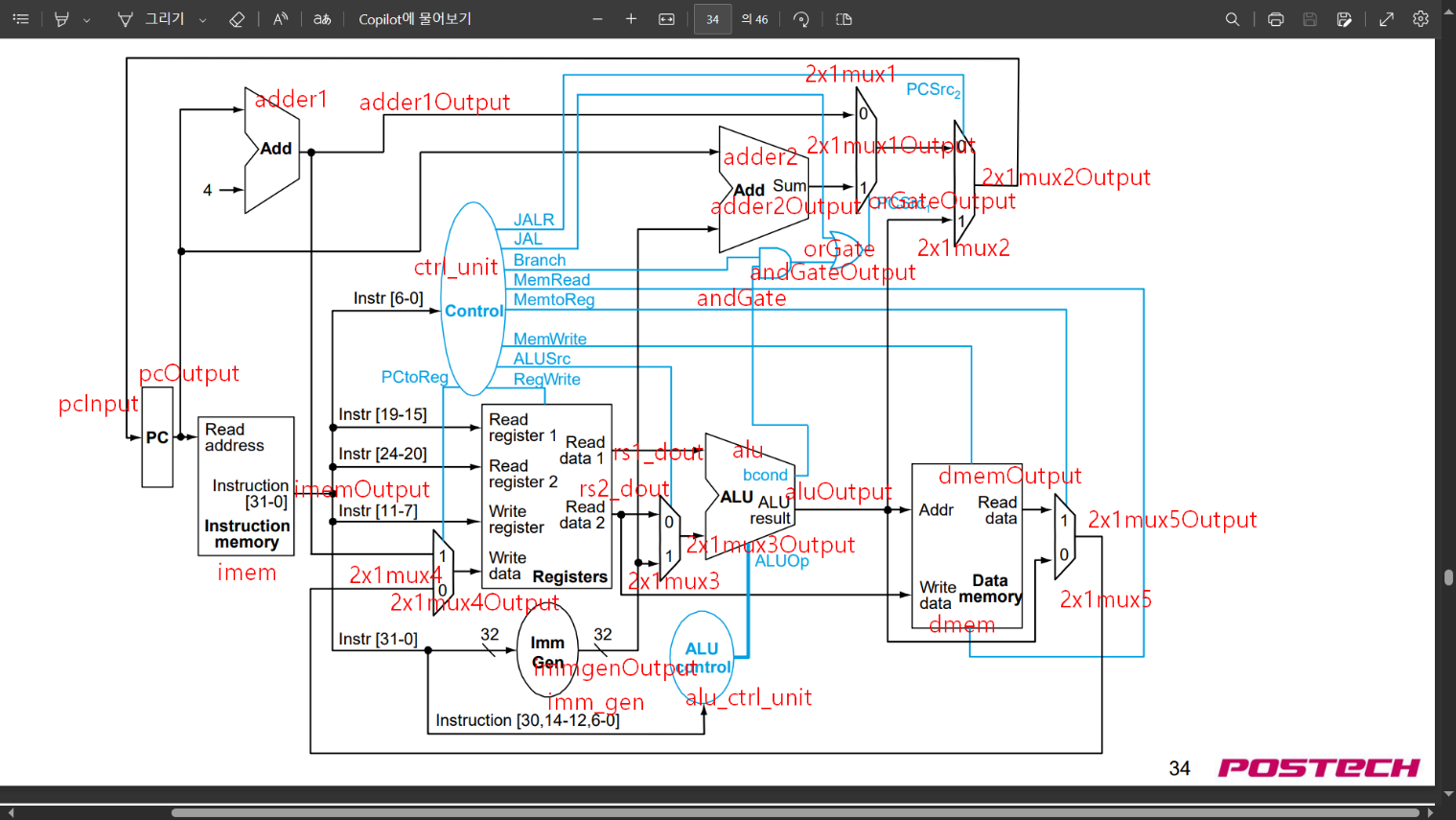
나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

I completed this programming task without the improper help of others.

1. **Introduction**

RISC-V instruction 실행 가능한 Single-Cycle CPU를 verilog로 구현하였다.  
모든 타입의 연산을 한 사이클에 실행 가능하다.

1. **Design**
   1. 전체 구조  
      기본적으로 Single-Cycle CPU 교안의 마지막에 나온 구조를 사용하였다.  
      PC, Inst memory, Register file, Data memory, imm generator, alu control ,control unit, ALU, Adder 두개와 Mux 5개, AND와 OR 게이트로 구성되어 있다.  
      코드에서 사용된 모듈과 wire 이름을 빨간 글씨로 적어 두었다.



* 1. 모듈 설계.  
     cpu.v에 정의된 모듈들은 다음과 같다. 위의 그림대로 와이어를 연결하였다.

|  |  |
| --- | --- |
|  |  |
| Program Counter 모듈 | Instruction Memory 모듈 |
|  |  |
| Register File 모듈 | Control Unit 모듈 |
|  |  |
| Immediate Generator 모듈 | ALU Control Unit 모듈 |
|  |  |
| ALU 모듈 | Data Memory 모듈 |
|  |  |
| Adder, AND, OR 모듈 | 5개의 2:1 MUX 모듈 |

* 1. Stages

**텍스트, 도표, 평면도, 개략도이(가) 표시된 사진

자동 생성된 설명**

* + 1. IF

Instruction Fetch: PC값을 기반으로 instruction을 불러오는 과정이다.  
PC와 Instruction Memory에서 담당하는 역할이다.

* + 1. ID

Instruction Decode: instruction에 정의된 값들을 해석해서 연산에 필요한 값을 생성하는 과정이다.

* + 1. EX

Excute: ALU에서 연산하는 과정이다. 본 구조에서는 ALU가 하나 존재하며 ALU Control Unit 에서 제어한다.

* + 1. MEM

Memory Access: 메모리에 접근하고 쓰는 과정이다.  
load와 Store의 경우에만 사용되며, Control Unit에서 나오는 MemWrite, MemRead 신호에 따라 동작하게 된다.

* + 1. WB

WriteBack: 연산 결과를 다음 과정으로 전달하는 과정이다.  
ALU에서 연산한 값이거나 메모리에서 읽어온 값일 수 있다.

1. **Implementation**
   1. Program Conter

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명  
Syncronous 하게 PC를 업데이트한다. 업데이트되는 값은 mux2 Output이다.  
그 값은 PC+4이거나 PC+imm (JAL) 이거나 rs1+imm (JALR) 이다.

* 1. ALU

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명  
Asynchrounous하게 주어진 연산을 한다.   
ALU Control Unit에서 나온 alu\_op 값에 따라 동작한다.  
또한 btype에 따라 branch 계산도 한다..

* 1. ALU Control Unit

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명  
입력으로는 opcode와 funct3, funct7 [5]를 받아서 alu\_op와 btype(branch)을 출력한다.  
계산 공식은 교안에 나와있는 대로 구현하였다.

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

|  |  |
| --- | --- |
| 텍스트, 스크린샷, 소프트웨어, 디자인이(가) 표시된 사진  자동 생성된 설명 | 텍스트, 스크린샷, 폰트이(가) 표시된 사진  자동 생성된 설명 |

Asynchrounous 하게 연산한다.  
산술연산은 funct3을 기반으로 결정하며 IMM이 아닌 경우에 funct7로 SUB를 구분한다  
BRANCH는 일단 두 값의 차이를 구한 다음 ALU에게 어떤 비교를 해야 하는지 btype을 통해 제어한다.

* 1. Control Unit  
     각 항목별 assert 조건은 교안의 표를 참고하였다.

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명  
Asynchrounous하게 Opcode를 받아서 이에 맞는 컨트롤 값 설정을 해준다.

* 1. Imm Generator

텍스트, 번호, 낱말맞추기 퍼즐, 영수증이(가) 표시된 사진

자동 생성된 설명

Imm의 경우에는 assignment instruction에 있는 표를 참고하였다.  
opcode에 따른 instruction의 각 위치를 imm에 할당하였다.  
sign extension을 해주었으며   
필요한 경우 (branch, JAL) 0으로 고정하는 비트도 설정했다.  
Asynchrounous하게 계산하여 전달해 준다.

|  |  |
| --- | --- |
|  |  |

* 1. Instruction Memory

Asynchrounous하게 메모리에 접근하여 instruction을 읽어오는 역할을 한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* 1. Data Memory

|  |  |
| --- | --- |
| 텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진  자동 생성된 설명 | Asynchrounous하게 메모리에서 읽어온다. clk에 맞춰 메모리에 값을 쓴다.  control unit에서 나오는  mem\_read, mem\_write에 따라 동작한다. |

* 1. Register File

|  |  |  |
| --- | --- | --- |
|  | Asynchrounous하게 레지스터를 읽는다. clk에 맞춰 레지스터에 값을 쓴다.  이 모듈 내부에서 halt 수행을 한다. ecall 신호가 발생할 시, x17==10을 검사한 뒤 is\_halted 신호를 발생시킨다.  Control Unit에서 나오는 write\_enable에 맞추어 동작한다. x0은 hardwired 0으로 고정이기 때문에, x0에 write하지 못하게 한다. |  |

* 1. Mux, Adder, Gate Module

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

구현에 필요한 모듈들이다. Asynchrounous하게 작동한다.

1. **Discussion**
   1. x0가 0으로 고정되어 있다는 것, imm에서 sign extension을 해야 한다는 것, 하위 비트를 0으로 고정하는 것을 신경써야 했다.
2. **Conclusion**
   1. 결과 비교 (다음 페이지)  
      x18부터는 사용하지 않기 때문에 사진에는 나와 있지 않다.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 번호, 평행이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 번호, 평행이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 라인, 폰트이(가) 표시된 사진

자동 생성된 설명

3개의 테스트벤치 basic\_mem, loop\_mem, non-controlflow에서  
ripes와 전부 동일한 reg value와 사이클 수를 기록하였다.

이로써 Single-Cycle CPU를 성공적으로 구현하였다.