EE312\_P1\_20110343\_김진엽\_v1

1. 데이터 구조

레지스터를 담는 structure krpreg와 메모리를 담는 structure krpmem으로 구성.

krpreg는 스펙에 주어진 레지스터들과 Delay slot pc를 담는 delay\_slot으로 되어있다.

krpmem은 program memory와 data memory가 배열로 각각 4kB씩 할당되어있다.

1. Procedure
   1. 초기화
   2. 사용자 입력 및 instruction 실행 반복
   3. dump
   4. 종료
2. 초기화

Argument를 받고, Register와 memory instance를 만들고, binary파일을 받아서 program memory에 넣는다.

1. 사용자 입력

사용자 입력은 s, r, b, d, q를 받고 나머지는 재입력을 받는다.

s: Instruction을 한번 실행한다

r: Break point 또는 instruction 끝까지 계속하여 실행한다

b: Break point 설정

d: Register 출력

q: 프로그램 종료

1. Instruction 실행

fetch와 execute로 나뉜다.

Fetch: PC에 있는 IR을 받는다, 이후 PC에 4를 더하거나 delay\_slot이 설정된 경우 그 PC를 받는다.

Execute: IR을 decode하여 Opcode를 얻는다. Opcode에 따라 IR를 parsing하고 그 동작을 수행한다. Branch에서 pc에 어떤 값을 넣는 경우 바로 넣지않고 delay\_slot에 값을 넣은 후 execute가 끝난 후 적용되도록 했다.

1. Dump

프로그램이 종료된 경우 data memory를 little endian으로 dump한다.

1. Test bench

input\_hex를 수정하여 test bench를 만들고

예상결과를 적었다.

“EE312\_P1\_20110343\_김진엽\_첨부파일” 참고

너무 길어서 여기에는 적지 않았습니다.

1. 결과

예상한 결과와 같게 출력되는 것을 볼 수 있었다.

register의 경우 실행 중에 출력되는 값을 통해 비교하였고,

memory는 프로그램이 끝난 후 dump파일을 열어 확인하였다.