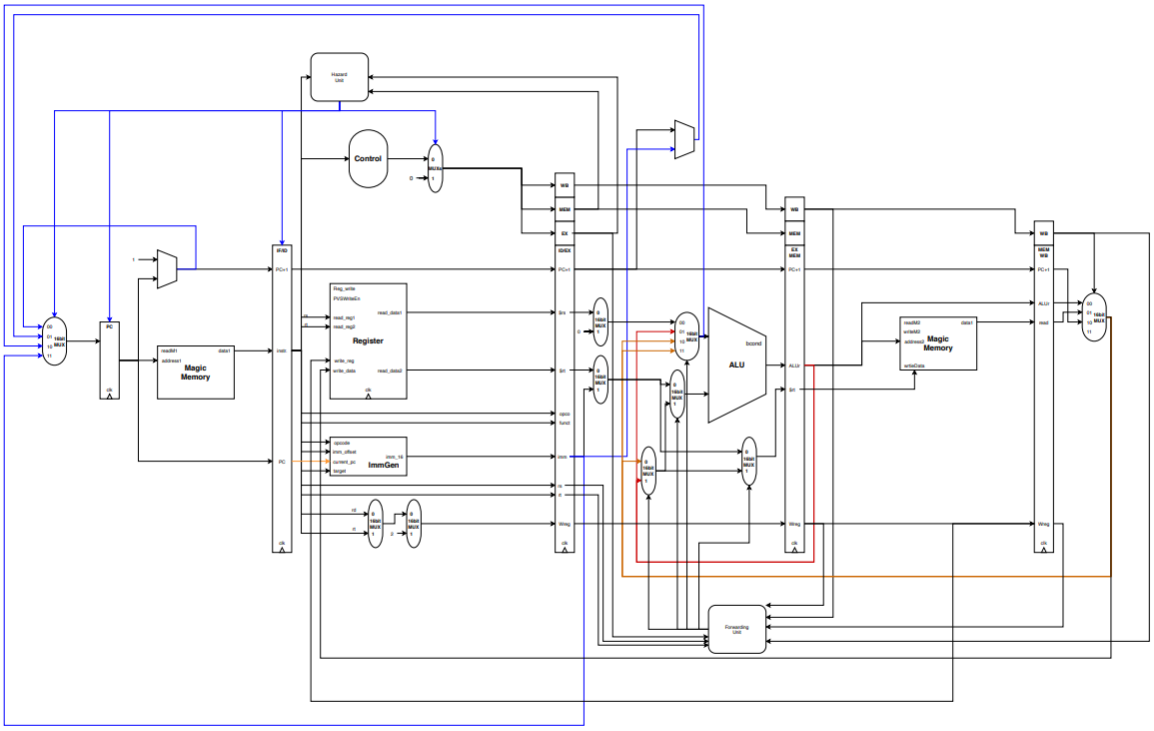
LAB 7. DMA

201800038 박형규, 20180480 성창환

1. Introduction

이번 Lab에서는 이전 Lab에서 구현한 Cached CPU에 External Device가 있을 때 memory의 접근을 처리하지 않고 DMA Controller를 이용하여 Cache에 의한 CPU 연산이 정상적으로 작동할 수 있게 할 것이다.

2. Design



기본적인 CPU의 Design은 이전 Lab에서 구현했던 Cached CPU와 동일하다. 이 구조에서 External Device와 DMA Controller에 따른 처리를 추가하여 구현하였다. 또한 External Device와 DMA Controller도 각각 구현하여 작동되도록 하였다. 작동 방식을 살펴보면 External Device에서 MEMWrite을 하려고 할 때 ddma_begin_interrupt를 발생시키면 DMA Controller에서 BR을 발생시키고, CPU에서 BG를 발생시킨다. 그리고 External Device에서 memory에 write할 address를 CPU에서 주게 되는데, 이에 따라 DMA Controller에서 memory 접근에 대한 처리가 이루어진다. External Device에 대한 memory 처리가 끝나면 DMA에서 BR signal을 없애고 CPU에서도 BG를 내리며 DMAController에서 Interrupt를 발생시키며 프로세스가 끝나게

된다.

3. Implementation

먼저 DMA Controller는 DMAC.v에 구현되었다. length신호가 CPU에서 들어온다면 length가 1이 되며 그 경우에 Bus Request 신호를 1로 올려주고 state를 0으로 설정해주며, address2Wr에 address를 할당해준다. 그리고 CPU에서 Bus Granted 신호가 도착하게 된다면 state가 12 미만일때 use_bus를 1로 계속 설정해주고, state를 1개씩 올려주며 12clock동안 작동되게 한다. 그리고 address2Wr을 1씩 올려주면서 메모리에서 clock이 지날때마다 메모리 address를 하나씩 올려주며 저장되는 공간을 지정해준다. 그리고 state가 12이면 cycle이 12cycle이 지난 것이므로 모든 설정을 초기화해준다.

다음으로 hazard_forward.v에서 HazardDetectionUnit 모듈에서 stall에 !access_mem을 추가해주어 memory에 access할 수 없을 때 stall이 되도록 하였다.

마지막으로 cpu.v.에서 cpu 모듈에서 처음에는 memory에 access 가능하기 때문에 access_mem을 1로 설정해주고, ddma_interrupted를 0으로 설정해주고, length를 초기화해주기 위한 temp_d를 0으로 초기화해주며 length도 0으로 초기화해준다. 그리고 dma에서 interrupt가 올 때 temp_d가 0이라면, 즉 dma가 실행중이 아니라면, 그리고 메모리에 접근중이 아니라면 dma가 프로세스를 수행할 수 있도록 설정해준다.

마지막으로 address2에는 access_mem이 0이라면 dma에게 16'hc7을 할당해주어 이 메모리 주소부터 접근할 수 있게 해주고, access_mem이 1이라면 cache에서 나온 address를 할당해준다.

4. Discussion

이번 과제에서는 12word를 external device가 요청한다고 생각하여 CPU에서 오래 기다릴 필요가 없었다. 하지만 현실 상황에서는 12word보다 더 많은 메모리를 요청할 수 있을 수도 있기 때문에 더 많은 시간을 CPU에서 기다릴 수 있다. 따라서 이후에 word를 12word가 아니라 작은 단위별로 나누어 처리하고, CPU의 작업을 처리하는 구현을 하면 좀 더 효율적인 구현이 될 수 있을 것이다.

5. Conclusion

이번 랩을 통해서 External Device와 연동하여 DMA Controller와 함께 CPU를 작동시키는 것을 구현하였다. 기존에 의도하였던 대로 External device의 요청에 따른 DMA Controller의 정

상적인 작동을 구현해낼 수 있었다.

정상적인 작동은 아래와 같은 이미지를 통해 확인 할 수 있었다. 우리가 의도한 memory address에 정상적으로 의도한 값이 들어가 있는 것을 확인 할 수 있다.