

컴퓨터 공학 기초 실험2 보고서

실험제목: Register File

실험일자: 2022년 10월 18일 (화)

제출일자: 2022년 10월 31일 (월)

학 과: 컴퓨터공학과

담당교수: 공영호 교수님

실습분반: 화요일 0, 1, 2

학 번: 2021202058

성 명: 송채영

1. 제목 및 목적

A. 제목

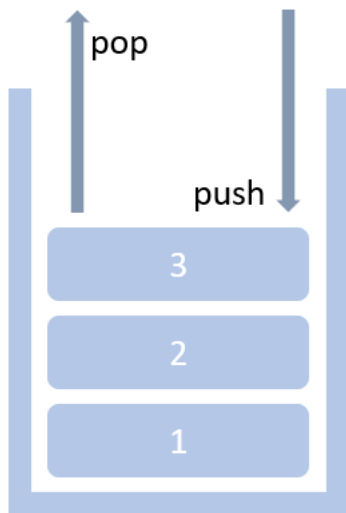
Register File

B. 목적

지난 실습 때 설계한 32bit register를 여러 개 instance하고 각 register에 주소를 할당하여 read & write를 수행하는 register file을 설계한다. Resettable enabled D flip flop을 instance 하여 32-bit register module을 생성하고 이를 다시 8번 instance해 8개의 32-bit register module을 구현해본다.

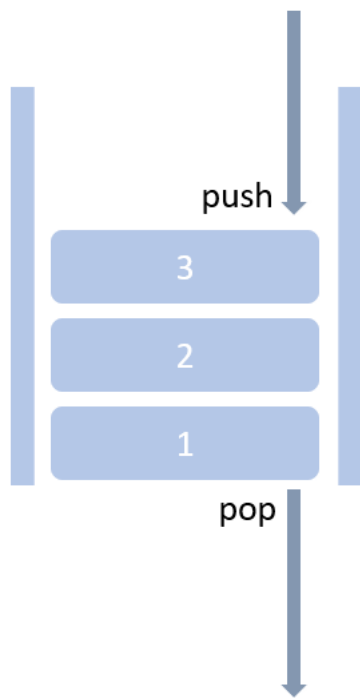
2. 원리(배경지식)

- Stack



스택(stack)은 한 쪽 끝에서만 자료를 넣거나 뺄 수 있다. 선형 구조, 즉 LIFO로 되어 있다. LIFO는 Last In First Out으로 먼저 들어간 데이터가 가장 아래에 깔리며, 데이터를 꺼낼 때는 제일 위에서부터 꺼내는 구조이다. 자료를 넣는 것을 push, 자료를 꺼내는 것을 pop이라고 하며 가장 최근에 push, 즉 가장 최근에 넣은 것이 가장 먼저 나오게 된다. d 여기서 put은 insert를, push는 delete를 의미하며 stack에서 insert와 delete는 같은 곳에서 발생한다. 이 과정이 위의 그림과 같다.

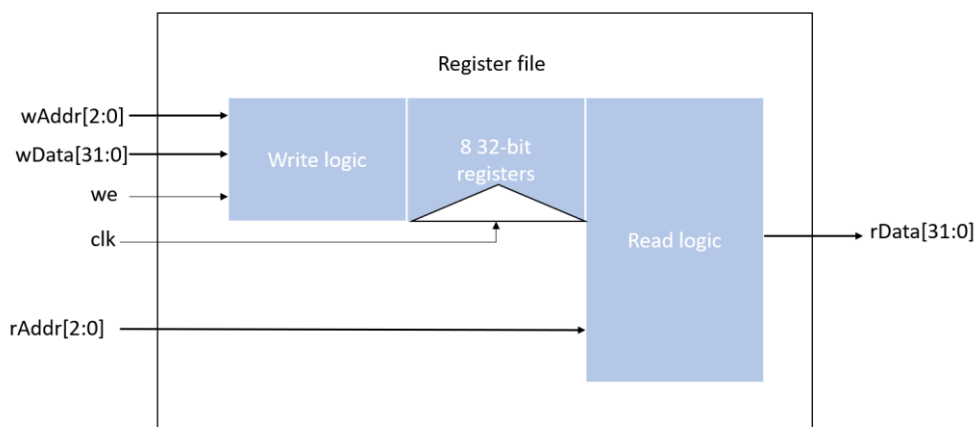
- Queue



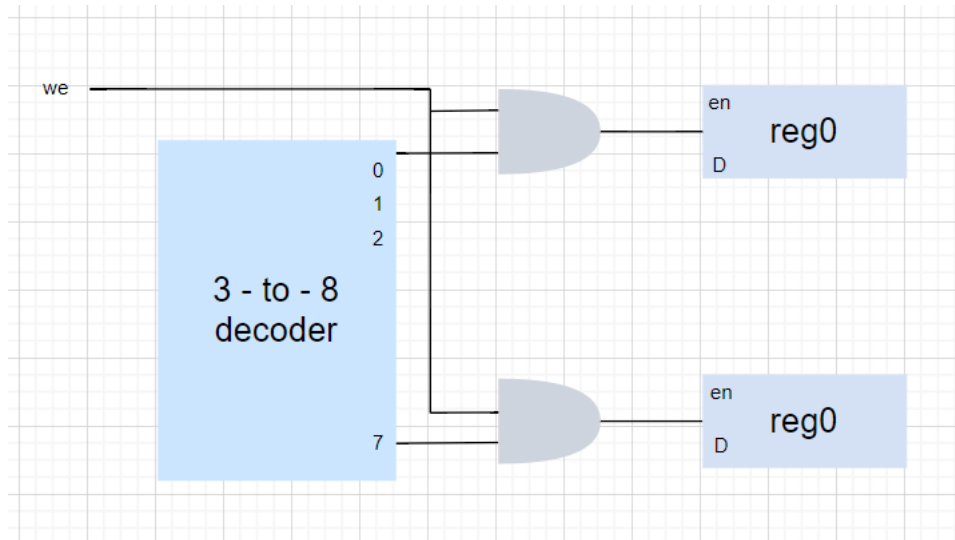
큐(Queue)는 FIFO의 구조이다. FIFO는 First In First Out으로 먼저 집어 넣은 데이터가 먼저 나오는 구조를 말한다. 나중에 집어 넣은 데이터가 먼저 나오는 스택(Stack)과 반대되는 개념이다.

큐에서 insert하는 것을 enqueue, delete하는 것을 dequeue라고 하는데 enqueue는 뒤에서, dequeue는 앞에서부터 진행한다. 이때 데이터를 insert 할 수 있는 위치를 front(head), 데이터를 delete 할 수 있는 위치를 rear(tail)이라고 한다. 또한 큐가 꽉 차서 더 이상 자료를 넣을 수 없는 경우를 overflow, 큐가 비어 있어서 자료를 꺼낼 수 없는 경우를 underflow라고 한다.

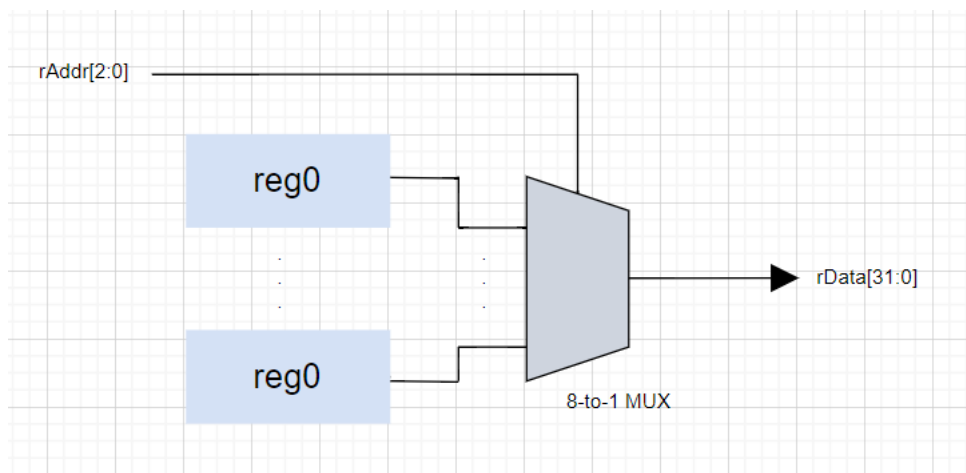
3. 설계 세부사항



Register_file module은 위의 사진처럼 32-bits register module을 8개 instance 한 것과, Write logic, Read logic으로 구성되어 있다. Write logic은 사용자로부터 wAddr를 받아 8개의 register 중 한 개를 선택하며, 만약 we 이 1이면 register에 값을 쓴다. 8개의 32bit register 중 write logic에 의해 선택되어진 register는 사용자로부터 받은 data를 저장하도록 한다. Read logic 은 사용자로부터 rAddr를 받아 8개의 register 중 한 개를 선택하여 값을 출력하도록 한다.



Decoder 는 write address를 받아 8개의 register 중 한 개를 선택하고 en signal과 decoder의 출력을 And gate로 묶어 하나의 register만 enable 시킨다.



MUX는 read address로 8개의 register output 중 하나를 선택한다.

4. 설계 검증 및 실험 결과

A. 시뮬레이션 결과

- Register file

Register file의 rtl viewer로, register 32와 write operation, read operation의 input과 output이 서로 연결되어 설계된 것을 확인할 수 있다.

Flow Summary	
<<Filter>>	
Flow Status	Successful - Mon Oct 31 06:01:31 2022
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	Register_file
Top-level Entity Name	Register_file
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	N/A
Total registers	256
Total pins	73
Total virtual pins	0
Total block memory bits	0
Total DSP Blocks	0
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0
Total DLLs	0

Register file의 flow summary이다. 총 256개의 register와 73개의 pin을 사용한 것을 알 수 있으며 성공적으로 컴파일이 완료된 것을 알 수 있다.

5. 고찰 및 결론

A. 고찰

실습 횟수가 늘어날수록 Testbench를 작성하고, verilog를 이용해 코드를 구현하는 것이 익숙해지고 있는 것 같다. Ppt에 제공된 waveform을 참고하여 testbench를 작성하니 수월하게 진행된 것 같다. 다만 wAddr과 rAddr의 값을 000, 001, 010, 011로 설정해 주었는데 radix를 16진수로 할 경우 0, 1, 2, 3으로 나왔다. 그 이유를 잘 모르겠어 코드도 다시 보고 다르게 설정해주기도 해보았지만, 이유를 알지 못해 아쉬웠다.

B. 결론

이번 실습을 통해 register file과 register에 address를 할당하여 read & write를 수행하는

write operation, read operation에 대해 처음 알게 되었으며, 읽고 쓰는 것이 가능하다는 사실이 신기했다. 레지스터 파일은 중앙 처리 장치, 즉 CPU에 있는 프로세서 레지스터의 배열인데, 이는 디지털논리회로 시간 뿐만 아니라, 어셈블리프로그래밍설계및실습과 오픈소스소프트웨어설계및실습 과목에서도 배운 내용이다. stack과 queue, register file등 다른 과목에서도 배우는 부분을 디지털 논리회로와 컴퓨터공학기초실험에서 배워 더 잘 이해가 되었다. 2학년 2학기의 과목들이 비슷한 내용을 바탕으로 진도를 나가기 때문에 도움이 많이 되는 것 같다. 앞으로의 남은 실습들도 잘 해결해 나가며 곧 있을 프로젝트도 성공적으로 마무리 해야겠다는 생각이 들었다.

6. 참고문헌

공영호 교수님/디지털논리회로2 강의자료/광운대학교 컴퓨터 공학과 2022 강의자료

공영호 교수님/컴퓨터공학기초실험2/광운대학교/2022 강의자료