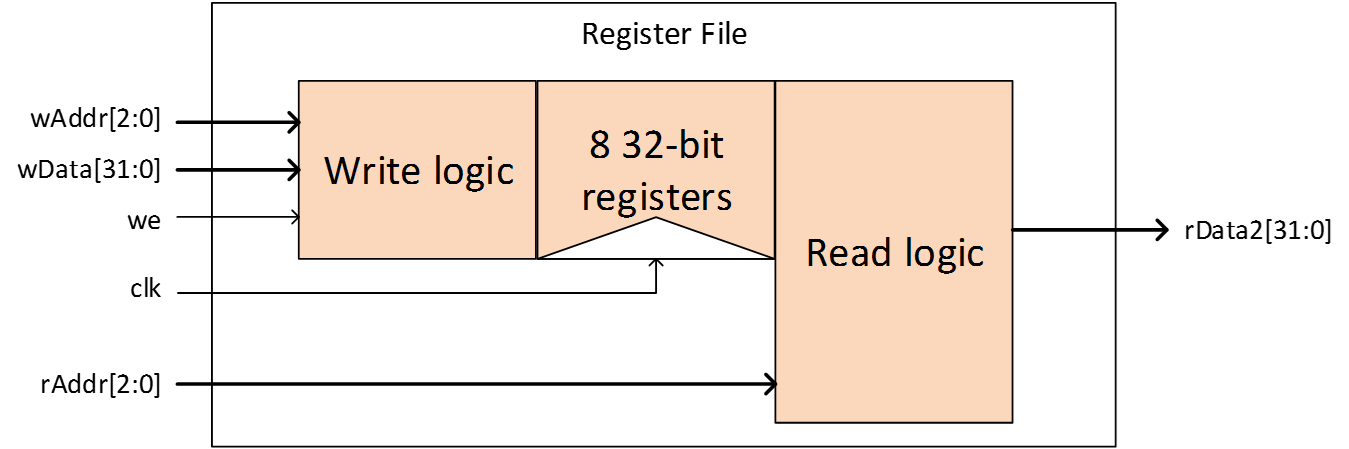
컴퓨터 공학 기초 실험 2 – Lab 7

Register File

# Register file

이번 실습에서는 지난 실습 때 설계한 32bit register를 여러 개 instance 하고 각 register에 address를 할당하여 read & write을 수행하는 register file을 설계한다.

* Description
* 32bit register가 8개 instance 된다.
* write는 write enable (we)에 의해 활성화된다.
* Write operation
* Decoder를 통해 address를 해석하여 해당 register enable
* Read operation
* MUX를 통해 8개의 register 중 한 개를 선택한다.
* Structural specification



* Write logic은 사용자로부터 wAddr를 받아 8개의 register 중 한 개를 선택한 뒤 we이 1일 때 register에 값을 쓴다
* 8개의 32bit register중 write logic에 의해 선택되어진 register는 사용자로부터 받은 data를 저장한다.
* Read logic은 사용자로부터 rAddr를 받아 8개의 register중 한 개를 선택하여 값을 출력한다.
* Design specification
* Module configuration

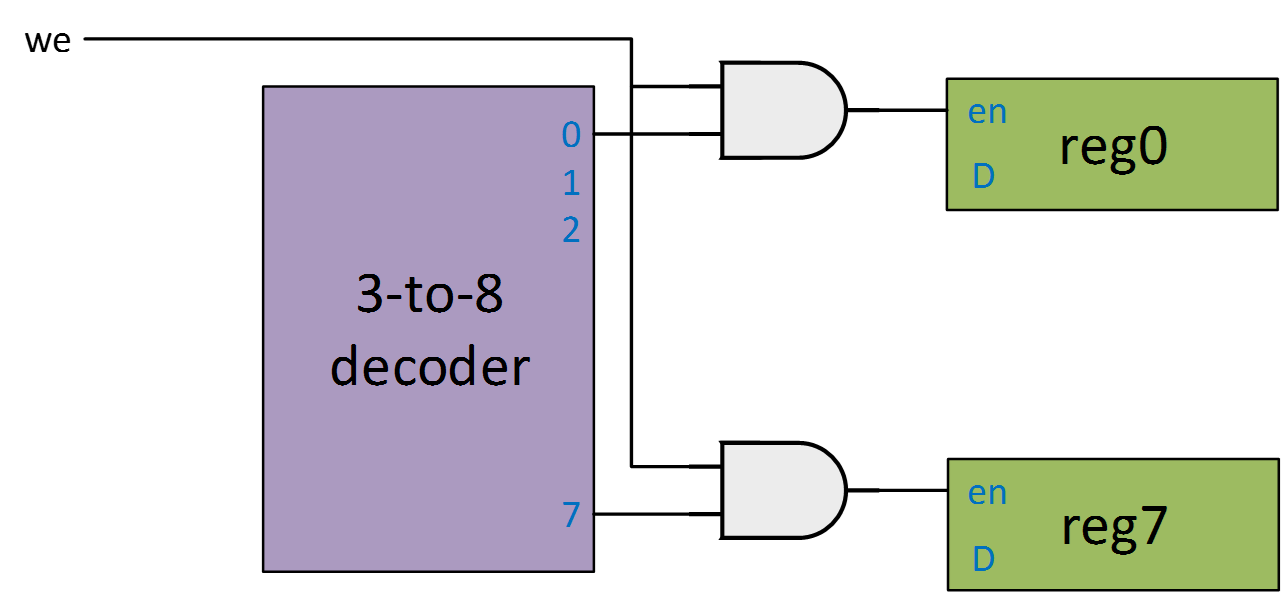
|  |  |  |
| --- | --- | --- |
| 구분 | 이름 | 설명 |
| Top module | Register\_file | Register file의 top module |
| Sub module | register32\_8 | 8개의 32bit register module(top module에서 instance) |
| Sub module | write\_operation | Write address decode module(top module에서 instance) |
| Sub module | read\_operation | Select register using read address(top module에서 instance) |

* I/O configuration

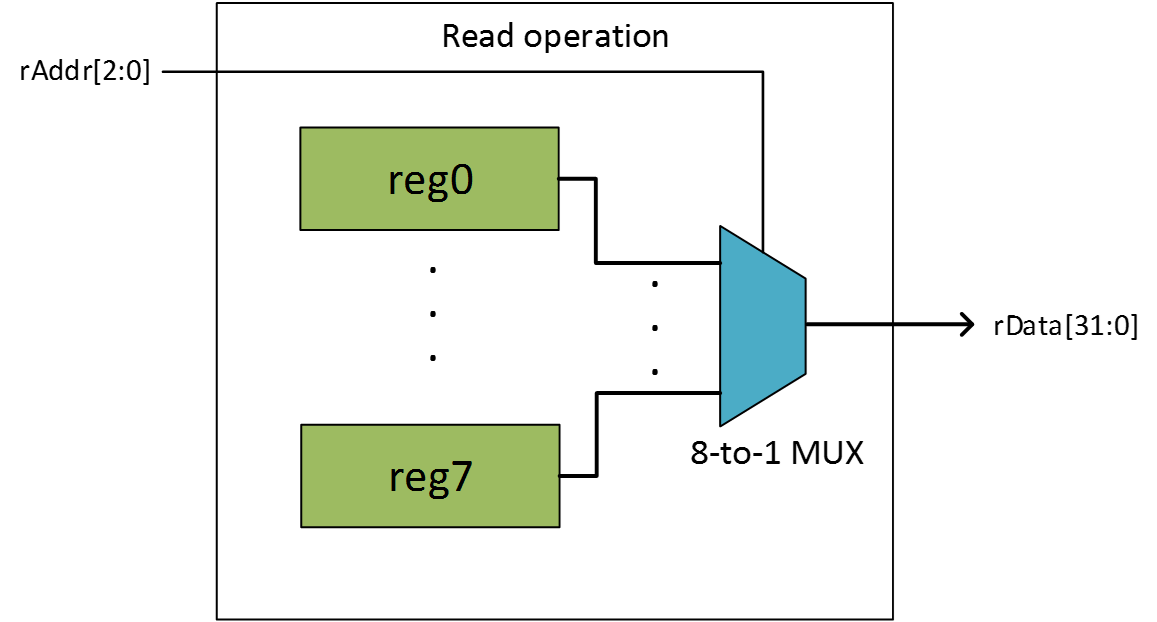
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module 이름 | 구분 | 이름 | 비트 수 | 설명 |
| Register\_  file | input | clk | 1-bit | Clock |
| reset\_n | 1-bit | Active low에 동작하는 reset 신호로 값이 인가되면 register의 값을 0으로 초기화 |
| we | 1-bit | Write enable |
| wAddr | 3-bits | Write address |
| rAddr | 3-bits | Read address |
| wData | 32-bits | Write data |
| output | rData | 32-bits | Read data |
| register32\_8 | input | clk | 1-bit | Clock |
| reset\_n | 1-bit | Active low에 동작하는 reset 신호로 값이 인가되면 register의 값을 0으로 초기화 |
| en | 1-bit | Register enable |
| d\_in | 32-bits | Data in |
| output | d\_out0~  d\_out7 | 32-bits | Register data out (8개) |
| write  \_operation | input | we | 1-bit | Write enable |
| Addr | 3-bits | Write address |
| output | to\_reg | 8-bits | Selected register enable signal |
| read  \_operation | input | from\_reg0  ~from\_reg7 | 32-bits | 8 registers’ data |
| Addr | 3-bits | Read address |
| output | Data | 32-bits | Data out |

# 실습

* Register file
* 32-bits registers  
  Resettable enabled D flip flop을 instance 하여 32-bit register module을 생성하고 이를 다시 8번 instance하여 8개의 32-bit register module 생성
* Write operation  
  Decoder는 write address를 받아 8개의 register 중 한 개를 선택하고 en signal과 decoder의 출력을 and gate로 묶어 하나의 register만 enable 시킨다.



* Read operation  
  MUX는 read address로 8개의 register output 중 하나를 선택한다.



# Report & Submission

* 레포트는 공지사항에 올린 보고서 양식에 맞추어 작성하고, 다음의 사항에 대하여서도 추가적으로 작성한다.
* 제출할 프로젝트: Register\_file
* Register file를 구현하고 검증한다.
* 원리(배경지식)에 stack과 queue에 대해 조사하시오
* 제출기한 : 1주 (KLAS 과제란 참고) / delay 2 days (20%감점)
* 실습 미수강 인원은 화요일반에 기준하여 디지털 논리 과목 담당 조교에게 보낼 것
* Source code 압축 시 db, incremental\_db, simulation 폴더는 삭제(미수행시 감점)
* 실습자료와 다르게 instance를 요구한 모듈에서 behavioral하게 구현한 경우 (gate혹은 모듈을 instance하지 않을 시) 감점
* 채점기준

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 세부사항 | | 점수 | 최상 | 상 | 중 | 하 | 최하 |
| 소스코드 | Source code가 잘 작성 되었는가?  (Structural design으로 작성되었는가?) | 10 | 10 | 8 | 5 | 3 | 0 |
| 주석을 적절히 달았는가?  (반드시 영어로 주석 작성) | 20 | 20 | 15 | 10 | 5 | 0 |
| 설계검증  (보고서) | 보고서를 성실히 작성하였는가?  (보고서 형식에 맞추어 작성) | 30 | 30 | 20 | 10 | 5 | 0 |
| 합성결과를 설명하였는가? | 10 | 10 | 8 | 5 | 3 | 0 |
| 검증을 제대로 수행하였는가?  (모든 입력 조합, waveform 설명) | 30 | 30 | 20 | 10 | 5 | 0 |
| 총점 | | 100 |  | | | | |