

컴퓨터 공학 기초 실험 2

Assignment 8. Synchronous FIFO

1. Synchronous FIFO

➤ Description

- ✓ 이전 실습에 진행한 register file 을 포함한다.
- ✓ Status flag 로 full 과 empty 를 제공한다.
- ✓ Invalid 한 read 또는 write 요청에 대해 FIFO 상태를 변화시키지 않는다.
- ✓ 4 개의 handshake signal(**wr_ack**, **wr_err**, **rd_ack**, **rd_err**)을 write 와 read 요청에 대하여 feedback 으로 제공한다.
- ✓ Count vector 는 FIFO 안의 현재 data 수를 제공한다.

➤ Structural specification

- ✓ Next state logic 은 외부로 wr_en 과 rd_en 을 받고 내부에서 현재 state 와 data_count 를 받아 다음 state 를 출력한다.
- ✓ Calculate address logic 은 state, data_count, head, tail 을 받아 다음 state 에 대한 head, tail 값(register file 의 주소값)과 현재 data 의 개수를 계산한다.
- ✓ Output logic 은 현재 state 와 data_count 를 받아 full, empty 및 handshake signal 을 출력한다.
- ✓ Register file 은 사용자로부터 받은 입력을 저장하거나 알맞은 register 의 값을 출력한다.

➤ Design specification

- ✓ 아래의 내용은 조교가 구현한 코드의 module configuration 과 I/O configuration 으로 Top 모듈의 이름과 I/O 를 제외하고 동일할 필요는 없음
- ✓ Module configuration (예시)

구분	이름	설명
Top module	fifo	FIFO 의 top module
Sub module	fifo_ns	Next state module(top module 에서 instance)
Sub module	fifo_cal	Calculate address module(top module 에서 instance)
Sub module	fifo_out	Output logic module(top module 에서 instance)
Sub module	Register_file	Register file module(top module 에서 instance)

그 외 필요한 모듈에 대해선 자유롭게 추가한다.

➤ I/O configuration (예시)

- ✓ Top module 인 fifo 의 I/O 만 반드시 동일하게 구현

Module 이름	구분	이름	비트 수	설명
fifo	input	clk	1-bit	Clock
		reset_n	1-bit	Active low 에 동작하는 reset 신호로 값이 인가되면 register 의 값을 0 으로 초기화
		rd_en	1-bit	Read enable

	output	wr_en	1-bit	Write enable
		d_in	32-bits	Data in
		d_out	32-bits	Data out
		full	1-bit	Data full signal
		empty	1-bit	Data empty signal
		wr_ack	1-bit	Write acknowledge
		wr_err	1-bit	Write error
		rd_ack	1-bit	Read acknowledge
		rd_err	1-bit	Read error
		data_count	4-bits	Data count vector
fifo_ns	input	wr_en	1-bit	Write enable
		rd_en	1-bit	Read enable
		state	3-bits	Current state
		data_count	4-bits	Data count vector
	output	next_state	3-bits	Next state
fifo_cal_addr	input	state	3-bits	Current state
		head	3-bits	Current head pointer
		tail	3-bits	Current tail pointer
		data_count	4-bits	Current data count vector
	output	we	1-bit	Register file write enable
		re	1-bit	Register file read enable
		next_head	3-bits	Next head pointer
		next_tail	3-bits	Next tail pointer
		next_data_count	4-bits	Next data count vector
fifo_out	input	state	3-bits	Current state
		data_count	4-bits	Current data count vector
	output	full	1-bit	Data full signal
		empty	1-bit	Data empty signal
		wr_ack	1-bit	Write acknowledge
		wr_err	1-bit	Write error
		rd_ack	1-bit	Read acknowledge
		rd_err	1-bit	Read error

2. Report

- 레포트는 공지사항에 올린 보고서 양식에 맞추어 작성하고, 다음의 사항에 대하여서도 추가적으로 작성한다.
- 제출할 프로젝트: fifo
- FIFO 를 구현하고 검증한다.
- 새로 구현한 submodule 을 포함한 모든 module 을 검증한다.
- ✓ 검증을 위한 Testbench 입력을 선택한 이유와 waveform 을 설명한다.
- 채점기준

세부사항		점수	최상	상	중	하	최하
소스코드	Source code 가 잘 작성되었는가? (Structural design 으로 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					

3. Submission

- 제출기한
 - 자세한 제출기한은 KLAS 와 일정을 참고
- 과제 업로드
 - ✓ Source code 와 report 를 같이 ZIP 파일로 압축하여 KLAS(종합정보서비스) 과제 제출에 해당 과제 upload
 - ✓ 업로드 파일명은 (요일#)_(학번)_Assignment_#.zip
 - 요일번호
 - 실습 미수강은 0
 - 월요일 0, 1, 2 교시 1
 - 화요일 0, 1, 2 교시 2
 - 수요일 0, 1, 2 교시 3
 - Ex) 월요일 반 수강, 2019110609, Assignment 1 제출 시
1_2019110609_Assignment_01.zip 으로 제출
 - ✓ Report 명은 (요일#)_(학번)_Assignment_#.pdf
 - 요일 번호는 위의 업로드 파일명과 동일하게 진행
 - ✓ Ex) 수요일 반 수강, 2019110609, Assignment 1 제출 시
1_2019110609_Assignment_01.pdf 으로 제출
 - ✓ Report 는 PDF 로 변환해 제출 (미수행시 감점)
- Source code 압축 시 db, incremental_db, simulation 폴더는 삭제 (미수행시 감점)
- Source code 압축 시 ~.bak 파일 삭제 (미수행시 감점)
- 제출할 프로젝트
 - ✓ FIFO