

컴퓨터 공학 기초 실험2 보고서

실험제목: FIFO

실험일자: 2023년 10월 27일 (금)

제출일자: 2023년 11월 08일 (수)

학 과: 컴퓨터공학과

담당교수: 이준환 교수님

실습분반: 금요일 0, 1, 2

학 번: 2020202037

성 명: 엄정호

1. 제목 및 목적

A. 제목

FIFO

B. 목적

베릴로그를 이용해 처음 들어간 값이 처음 출력되는 fifo로직을 베릴로그를 이용해 구현한다. 구현한 모듈을 이용해 fifo로직을 이해한다.

2. 원리(배경지식)

자료구조: 컴퓨터 구조에서 효율적인 접근 및 수정을 위해서 구성한 자료의 구성, 저장, 관리 로직을 말한다. 즉 여러 데이터의 모임, 데이터들 사이의 관계, 데이터에 적용 할 수 있는 함수나 명령을 의미한다. 자료구조에 따라 알고리즘의 속도가 달라질 수 있기 때문에 프로그램 구성에 따라 자료구조가 달라져야 한다. 자료구조는 stack, queue, 그래프, 트리, 힙 등 여러 종류가 존재한다.

프로그램 설계시 가장 우선적으로 어떤 자료구조를 이용할지 선택해야 한다. 큰 로직을 제작할 때 최종 결과물의 성능이 자료구조에 크게 의존하기 때문이다.

FIFO : 스케줄링 기법 중 하나로 먼저 들어온 데이터를 먼저 내보내는 데이터 구조이다 (QUEUE).

Queue : 선입선출(fifo)자료구조 중 하나로 스택과는 반대되는 개념이다.

프린터의 출력, 윈도우 시스템, 프로세스 관리 등의 데이터가 입력된 시간 순서대로 처리되어야 하는 일에 쓰인다.

Queue(FIFO)

INPUT	INPUT	OUTPUT	INPUT	OUTPUT
			E	E
	D	D	D	D
C	C	C	C	C
B	B	B	B	
A	A			

Stack : 큐와 다르게 가장 나중에 입력된 데이터가 가장 먼저 나가는 LIFO구조이다. 자료의 한쪽에서 삽입과 출력이 일어난다.

INPUT	INPUT	OUTPUT	INPUT	OUTPUT
			E	
	D		D	D

C	C	C	C	C
B	B	B	B	B
A	A	A	A	A

3. 설계 세부사항

초기 STATE는 INTI에서 시작하게 된다. 각 STATE마다 RE, WR활성화에 따라 값을 출력하거나 다음 상태로 이동한다.

STATE 설명

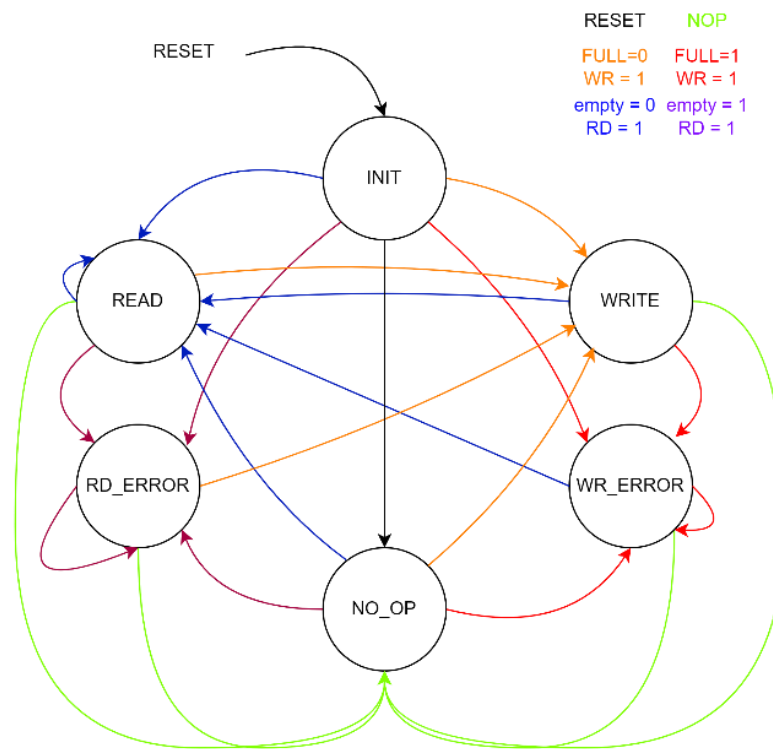
INIT	RESET이 0일 때 이동, 데이터 초기화
NO_OP	WR,RD가 둘다 0이거나 잘못된 값 입력 시 이동 동작X
WRITE	WR활성화시 이동, 입력, 카운트 증가
WR_ERROR	FULL상태에서 WR활성화시 이동, 에러 출력
READ	RD 활성화시 이동, 출력, 카운트 감소
RD_ERROR	데이터 카운트가 0일 때 RD활성화시 이동

FULL과 EMPTY의 활성화는 DATA COUNT에 의해 결정되며 아래와 같다

FULL & EMPTY

DATA COUNT	FULL	EMPTY
8	1	0
0	0	1
1 - 7	0	0

FIFO->FSM



모듈설계

Fifo_ns

현재 상태를 통해 다음 상태를 계산하는 모듈로 wr_en, rd_en과 현재 상태를 입력으로 받는다. 현재 상태에서 en값 활성화에 따라 next_state를 출력한다 출력된 next state는 바로 state로 반영되지 않고 레지스터3에 의해서 clk이 rise됐을 때 값을 반영해준다.

Fifo_cal_addr

Next_state를 기준으로 head, tail, data_count를 입력 받아 다음에 실행될 state에 따른 값을 반영 해준다.

Read : re =1, we = 0, tail의 위치를 1증가 시키고 head는 그대로 둔다 empty상태에서는 Read로 이동하지 않게 설계했기 때문에 따로 예외처리는 진행하지 않았다.

Write : re =0, we = 1, head의 위치를 1증가 시키고 tail 그대로 둔다 full 상태에서는 Write로 이동하지 않게 설계했기 때문에 따로 예외처리는 진행하지 않았다.

그외 : we: 0 re:0로 설정하고 나머지 값은 변하지 않는다.

fifo_out

현재 상태와 data_count를 기준으로 state, full, empty, wr_ack, wr_err, rd_ack, rd_err를 출력한다. Full, empty은 data_count에 의해서만 결정되고 나머지 wr, rd의 경우 현재 상태에 따른 출력을 갖게 된다.

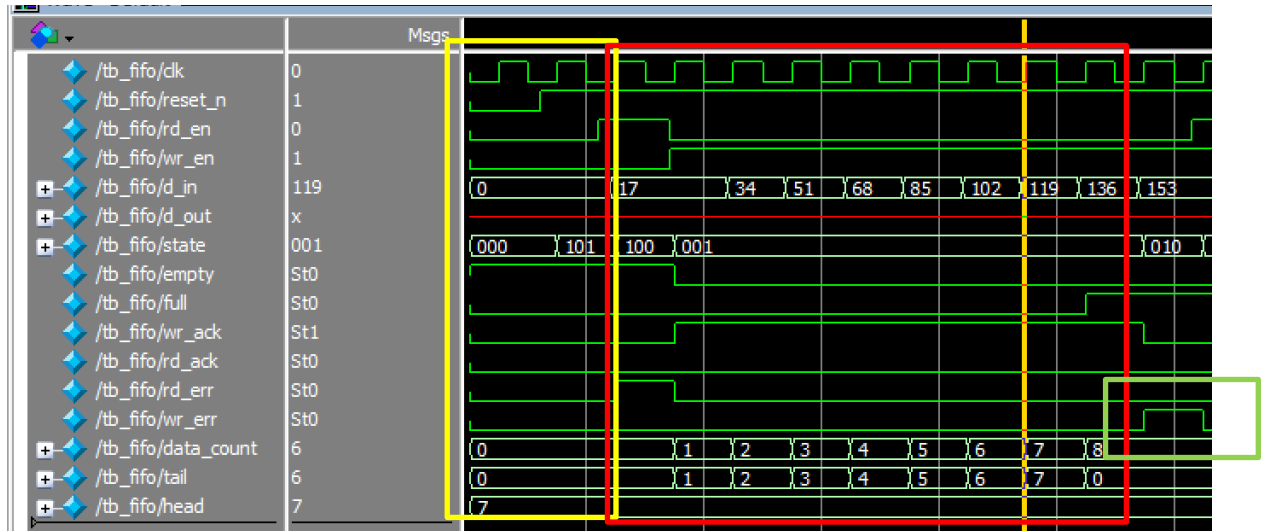
register3_r

state, tail, head, data_count의 변화를 위해 만든 모듈로 각 데이터의 다음 값을 현재 값

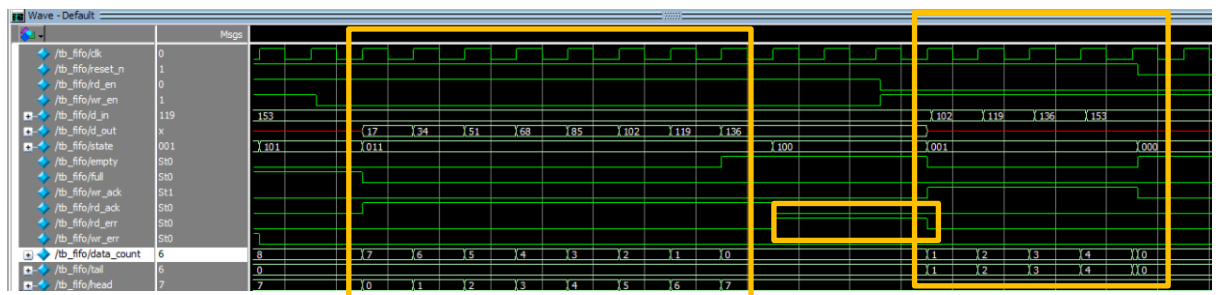
state 따라 wr_err, wr_ack, rd_err, rd_ack 중 하나의 값만 1이 되도록 설정 full과 empty의 경우 data count의 값의 변화에 의해서만 변동하도록 설정

FIFO

state변화 확인을 위해 테스트 벤치를 진행 할 때 head와 tail, state를 추가 했습니다.



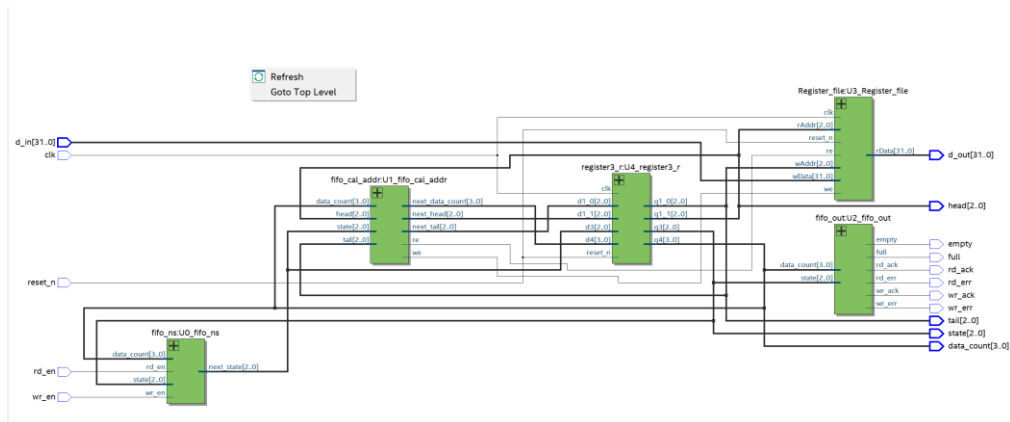
1. 리셋이 0일때 값 변화 x 헤더를 0번부터 읽기 위해 첫 헤드를 111로 초기화.
-리셋 작동 확인
2. En이 둘 다 0일 때 no_op, data_count가 0일 때 rd_en활성화 시 rd_err활성화
->State변화 확인
3. Wr_en이 활성화 시 데이터 입력, tail값 증가 head값의 경우 0으로 유지
->값 입력 확인
4. 값이 8개 입력 후 clk이 rise되면 wr_err출력
->에러 출력 검증



1. rd_en활성화시 head에서 값 생성 d_out
->값 출력 과정 확인
2. 데이터가 없을 때 rd발생시 에러 출력.
->에러 발생 확인.
3. 리셋 실행시 모든 값 초기화.
->리셋 검증

B. 합성(synthesis) 결과

FIFO



Flow Summary

<<Filter>>

Flow Status	Successful - Wed Nov 08 14:40:08 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	fifo
Top-level Entity Name	fifo
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	N/A
Total registers	269
Total pins	78

Input : 36개 : clk, reset, rd_en, wr_en, d_in-32

Output: 42개: datacount -4,full, empty, wr_ack/err, rd_ack/err, dout-32

총 78개의 pin사용

5. 고찰 및 결론

A. 고찰

Full empty설정시 둘 중 0일 때 empty만 1로 바꾸어 주니 초기 값에서 full값이 unknown이 이 나오는 상황이 발생해 0일 때 와 8일 때 둘 다 full과 empty값 모두 직접 설정해주었다.

각각의 next state로직에도 clk를 할당해서 클럭이 작동 했을 때 데이터가 변화하도록 설정했는데 그렇게 설정하게 되면 값 전달에 clk가 2번 상승해야 했다. 그래서 next state현재 state를 반영해 항상 다음 상태를 가리키도록 설정하고 register에서 clk를 통해 값을 받아 올 수 있도록 했다.

Tail과 head를 이용한 데이터 입출력시 데이터를 입력할 땐 tail 값이 0일 때부터 시작했

는데 이후 head에서 값을 가져올 땐 1부터 시작하는 것을 확인해 head값을 1111부터 리셋 되도록 설정했다 둘 다 next state로직에서 값을 받아올 수 있도록 했는데 왜 head 부분만 먼저 1을 받고 값을 출력하는지 아직 찾지 못했다.

B. 결론

베릴로그를 이용한 FIFO모듈을 구성했다. 이번 실험을 통해 현재 상태에 따른 값을 가져올 때 미리 해당 값은 구해 놓고 CLK를 이용해 값을 가져오면 된다는 것을 알게 되었고 ALWAYS문 사용시 테스트벤치에서는 조건을 지정 안해도 자동으로 돌아가지만 모듈의 경우 ALWAYS(*)이런 식으로 설정해야 한다는 것을 알게 되었다. 또한 FIFO의 동작 방식을 이해하고 구성하니 메모리 구조에 대해 더 알 수 있었다.

6. 참고문헌

자료구조/

https://ko.wikipedia.org/wiki/%EC%9E%90%EB%A3%8C_%EA%B5%AC%EC%A1%B0

이준환/디지털논리회로/광운대학교 /2023

이형근/컴퓨터공학기초실험2/광운대학교 /2023

최상호/데이터구조설계/광운대학교/2023