



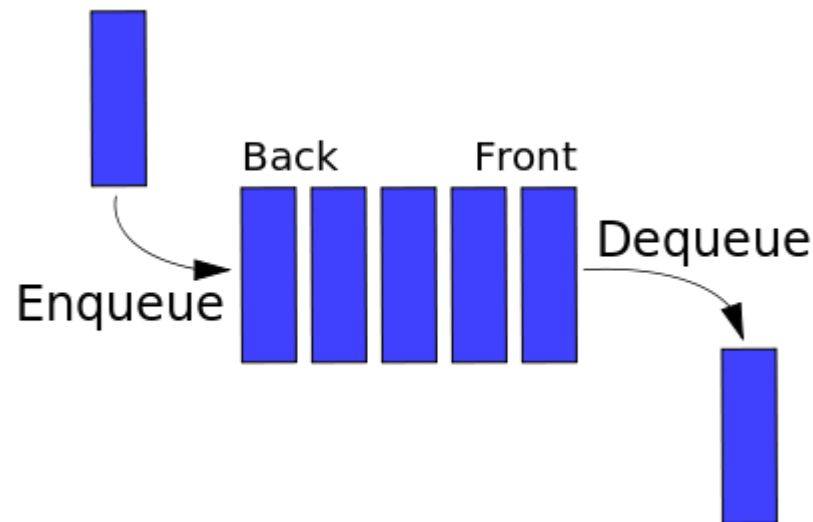
컴퓨터공학 기초 실험2

Lab #8

FIFO

First-In-First-Out (FIFO)

- FIFO has control logic that manages read & write pointer which generates status flags and provides handshake signals for interface with user logic
- Queue
 - ✓ First-in, First-out
 - ✓ Nodes are removed only from the head (dequeue)
 - ✓ Nodes are inserted only at the tail (enqueue)



Synchronous FIFO

PRACTICE I

Functional Description

- 8개의 32bit register로 이루어짐
- 각 FIFO는 status flag와 handshake signal을 출력
 - ✓ Status flag: full, empty
 - ✓ Handshake signal: wr_ack, wr_err, rd_ack, rd_err
- Counter vector는 FIFO안에 있는 현재 data 수 제공
- 다음 슬라이드는 조교가 구현한 예시이므로 동일할 필요 없음

Project Properties

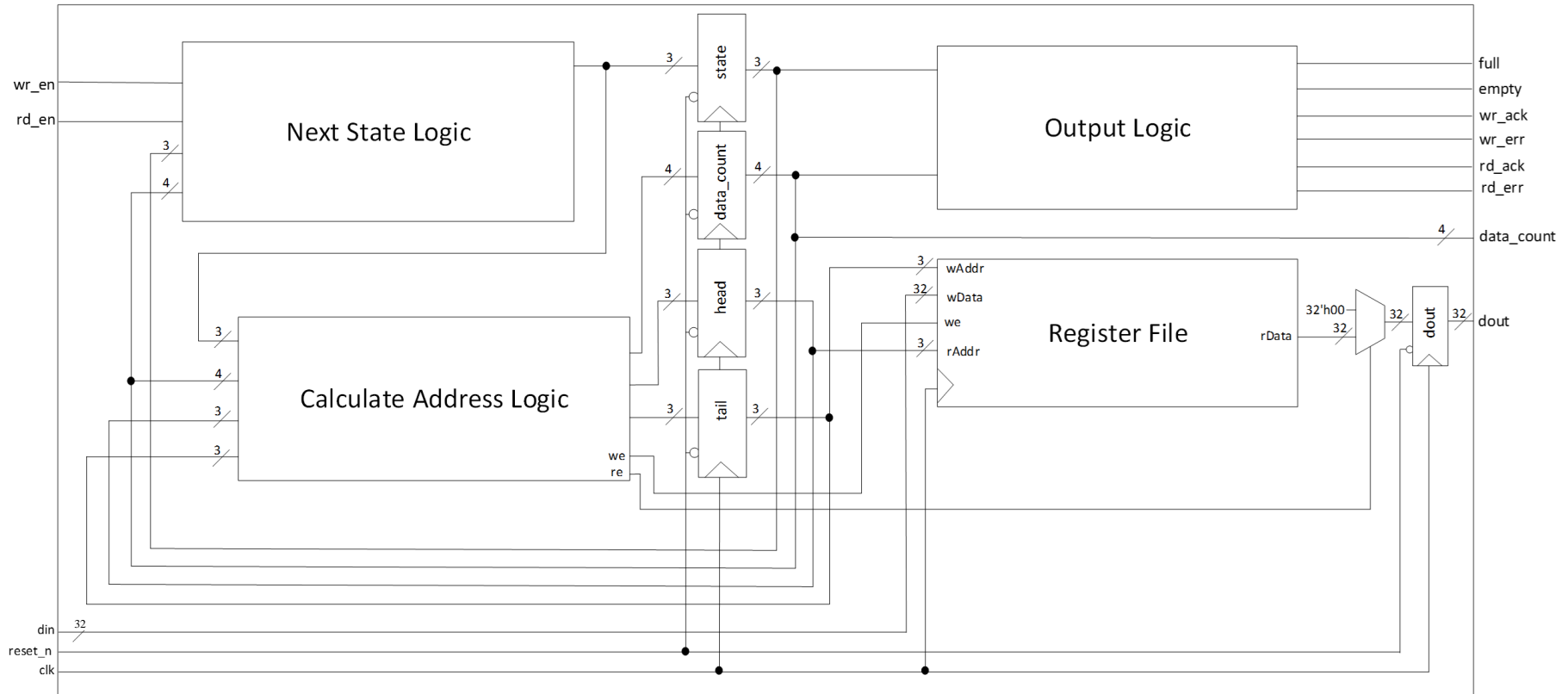
➤ New Project Wizard

- ✓ Project name : fifo
- ✓ Family & Device : Cyclone V 5CSXFC6D6F31C6 (밑에서 6번째)

➤ Verilog file

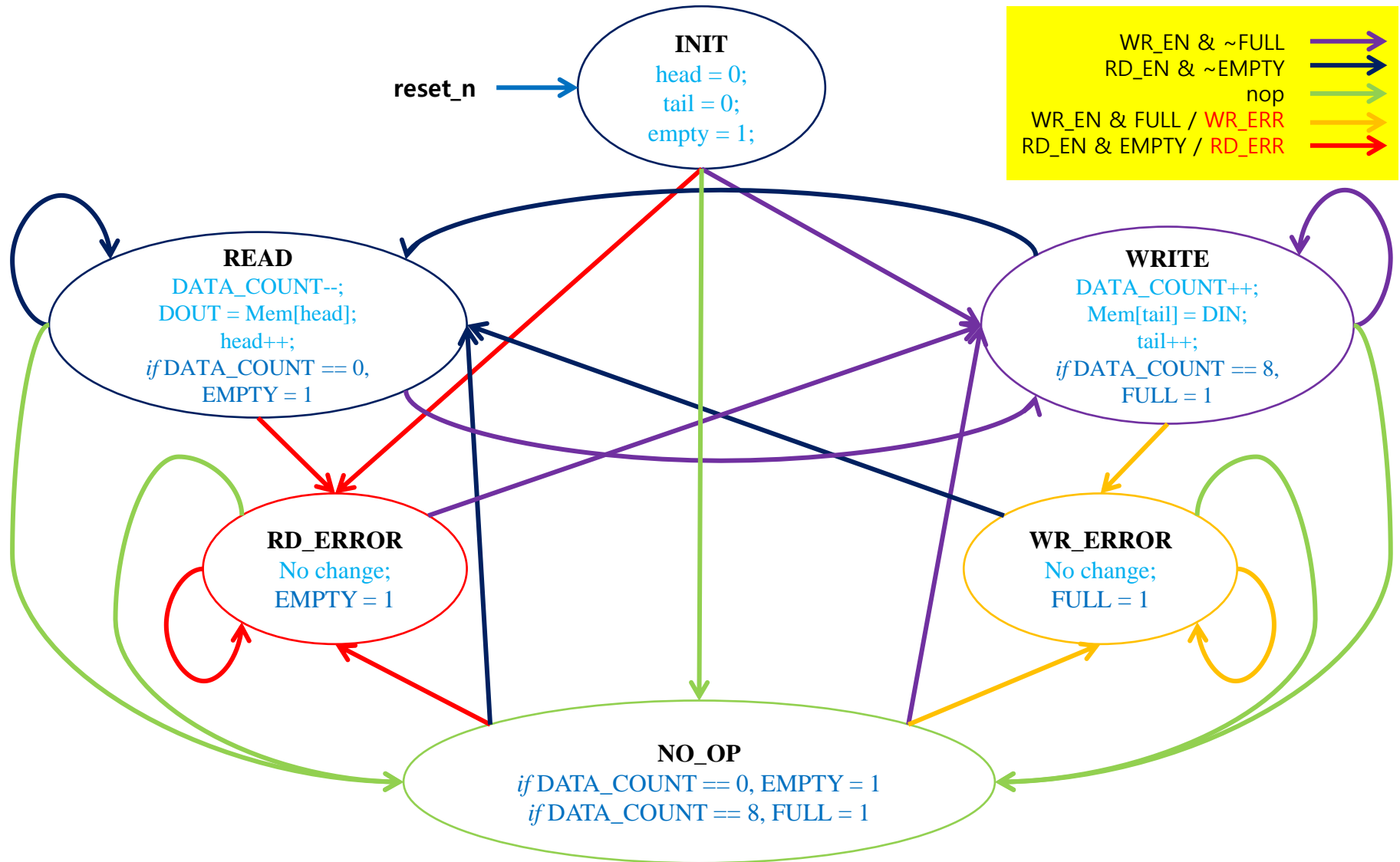
- ✓ Add files: register32_r_en.v, write_operation.v, read_operation.v, Register_file.v
- ✓ New files: fifo.v, fifo_ns.v, fifo_cal.v, fifo_out.v

Block Diagram



- Next state logic: 외부로 `wr_en`과 `rd_en`을 받고 내부에서 현재 `state`와 `data_count`를 받아 다음 `state`를 출력
- Calculate address logic: `state`, `data_count`, `head`, `tail`을 받아 다음 `state`에 대한 `head`, `tail` 값 (register file의 주소값)과 현재 `data`의 개수를 계산
- Output logic: 현재 `state`와 `data_count`를 받아 `full`, `empty` 및 handshake signal을 출력
- Register file: 사용자로부터 받은 입력을 저장하거나 알맞은 register의 값을 출력

FIFO FSM



State Operations

➤ 프로젝트 내에서 필요로 하는 State는 다음과 같음

State	Operations
INIT	head = tail = DATA_COUNT = 0;
NO_OP	No change
WRITE	mem[tail] = DIN; tail++; DATA_COUNT++; If DATA_COUNT == 8, FULL = 1;
WR_ERROR	No change; FULL = 1;
READ	DOUT = mem[head] = DIN; head++; DATA_COUNT--; If DATA_COUNT == 0, EMPTY = 1;
RD_ERROR	No change; EMPTY = 1

Output Design – Full & Empty

- Data count를 기준으로 full, empty를 판단

Data_count	Full	Empty
0	0	1
8	1	0
1 ~ 7	0	0

Assignment 8

➤ Report

- ✓ 자세한 사항은 homework & practice document 참고

➤ Submission

- ✓ Soft copy
 - 강의 당일 후 1주까지(delay 2 days 20% 감점)
 - 실습 미수강은 디지털 논리2 조교 공지에 따름

Q&A

THANK YOU