컴퓨터 공학 기초 실험 2 – Lab 8

Synchronous FIFO

# Synchronous FIFO

* Description
* 이전 실습에 진행한 register file을 포함한다.
* Status flag로 full과 empty를 제공한다.
* Invalid한 read 또는 write 요청에 대해 FIFO 상태를 변화시키지 않는다.
* 4개의 handshake signal(**wr\_ack**, **wr\_err**, **rd\_ack**, **rd\_err**)을 write와 read 요청에 대하여 feedback으로 제공한다.
* Count vector는 FIFO안의 현재 data 수를 제공한다.
* Structural specification
* Next state logic은 외부로 wr\_en과 rd\_en을 받고 내부에서 현재 state와 data\_count를 받아 다음 state를 출력한다.
* Calculate address logic은 state, data\_count, head, tail을 받아 다음 state에 대한 head, tail값(register file의 주소값)과 현재 data의 개수를 계산한다.
* Output logic은 현재 state와 data\_count를 받아 full, empty 및 handshake signal을 출력한다.
* Register file은 사용자로부터 받은 입력을 저장하거나 알맞은 register의 값을 출력한다.
* Design specification
* 아래의 내용은 조교가 구현한 코드의 module configuration과 I/O configuration으로 Top 모듈의 이름과 I/O를 제외하고 동일할 필요는 없음
* Module configuration (예시)

|  |  |  |
| --- | --- | --- |
| 구분 | 이름 | 설명 |
| Top module | **fifo** | **FIFO의 top module** |
| Sub module | fifo\_ns | Next state module(top module에서 instance) |
| Sub module | fifo\_cal | Calculate address module(top module에서 instance) |
| Sub module | fifo\_out | Output logic module(top module에서 instance) |
| Sub module | Register\_file | Register file module(top module에서 instance) |

그 외 필요한 모듈에 대해선 자유롭게 추가한다.

* I/O configuration (예시)
* Top module인 fifo의 I/O만 반드시 동일하게 구현

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module 이름 | 구분 | 이름 | 비트 수 | 설명 |
| fifo | input | clk | 1-bit | Clock |
| reset\_n | 1-bit | Active low에 동작하는 reset 신호로 값이 인가되면 register의 값을 0으로 초기화 |
| rd\_en | 1-bit | Read enable |
| wr\_en | 1-bit | Write enable |
| d\_in | 32-bits | Data in |
| output | 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 |

# Report & Submission

* 레포트는 공지사항에 올린 보고서 양식에 맞추어 작성하고, 다음의 사항에 대하여서도 추가적으로 작성한다.
* 제출할 프로젝트: fifo
* FIFO를 구현하고 검증한다.
* 원리(배경지식)에 stack과 queue에 대해 조사한다.
* 강의 당일 후 1주까지(delay 2 days 20% 감점)
* 실습 미수강은 디지털 논리2 조교 공지에 따름
* Source code 압축 시 db, incremental\_db, simulation ~.bak 파일 및 폴더는 삭제 (미수행시 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 |  | | | | |