디지털 논리회로2 프로젝트 제안서

Factorial computation system

학 과: 컴퓨터정보공학부

담당교수: 이혁준 교수님

실습분반: 수요일 0, 1, 2

학 번: 2022202075

성 명: 우나륜

1. Title & Object

A. Title

Factorial computation system

B. Object

본 프로젝트는 top 모듈 내에 총 3가지의 중요한 module인 bus, memory(slave 0), factorial core (slave 1)로 이루어져 있으며 master에서 입력되는 값에 따라 특정 메모리 주소에 값을 저장하거나 불러오는 기능 또는 입력한 operand의 factorial을 구하는 기능을 수행한다.

2. Component concept

A. Factorial core

Factorial core는 입력된 operand 값에 대해 factorial 연산을 수행하는 module이다. Factorial을 연산하는 알고리즘은 Booth Multiplication Algorithm을 사용하여 연산을 수행한다.

Factorial core module의 input인 s_sel=0이면 bus에서 입력받은 메모리의 주소가 factorial core의 memory map region인 0x7000 ~ 0x703F에 해당되지 않은 것이므로 아무런 동작을 수행하지 않는다. 만약 s_sel=1이면 Factorial core 내부에 미리 정의한 offset의 값을 사용하여 해당 offset에 맞는 기능을 수행한다. 만약, 정의한 offset 이외의 값이 입력되면 해당 입력은 무시한다. 여기서 offset은 기준 주소 (base address)와 목표 주소 (target address)의 차이를 지칭할 때 쓰는 단어로, factorial core의 기준 주소는 0x7000이며, 타겟 주소는 bus에서 입력받은 s_addr이다. Offset의 값과 s_wr을 사용하여 register에 s_din을 작성하거나 읽어는다. 만약, s_wr = 0이면 register를 read하며, s_wr = 1이면 register에 write하는 기능을 수행한다.

Offset 0x00은 factorial 연산 시작에 대한 값이다. opstart[0] = 1이면, operand register에 저장된 값을 사용하여 연산을 시작한다. 만약 연산 완료 후, opclear 이전에 opstart[0] = 1이면 해당 값을 무시한다. Offset 0x01은 모든 register의 값을 초기화하는 기능으로, opclear[0] = 1이면 초기화를 수행한다. Factorial 연산을 시작할 때, opdone[1] = 1을 쓰고, 연산이 끝나면 opdone[0] = 1을 쓰고 대기한다. 만약 intrEN[0]이 1이면 연산이 끝났을 때 interrupt가 발생한다. 연산 결과의 값은 opclear[0] = 0인 동안 유지된다.

B. Bus

Bus는 1개의 master interface와 2개의 slave interface들 간의 data를 전송할 수 있도

록 연결해주는 요소이다. Master interface로부터 m_req을 받아 그 값에 따라 bus에 대한 소유권을 할당할 지 결정한다. 만약 m_req = 0이면 그에 대한 응답으로 m_grant = 0이 된다. 반대로, m_req = 1이면 그에 대한 응답으로 m_grant = 1이 되며 m_req = 1인 동안 m_grant의 값은 0이 되지 않는다.

두 개의 slave interface는 각각 Memory와 Factorial core를 가리킨다. Slave 0 (Memory)는 0x0000 ~ 0x00FF 사이의 memory address map region을 가지고, Slave 1 (Factorial core)는 0x7000 ~ 0x703F 사이의 값을 가진다. 만약 m_addr의 값이 위의 범위 중 하나도 해당되지 않으면 어떠한 s0_sel = 0, s1_sel = 0으로 바꾸어 slave device도 선택하지 않는다.

C. Memory (RAM)

RAM (Random access memory)은 임의의 address 값에 대해 data를 read 또는 write하는 memory이다. 메모리의 address는 8 bits이고 wen의 값에 따라 해당 주소에 저장되어 있는 데이터를 read 또는 write한다. 만약 cen = 1, wen = 1이면, s_addr이 가리키는 주소에 입력받은 s_din의 값을 write하고 출력값인 s_dout = 0을 출력한다. 만약 cen = 1, wen = 0이면, s_addr이 가리키는 주소에 저장된 데이터의 값을 s_dout에 write한다. cen = 0이면 s_dout의 값은 0이 되고 write는 수행하지 않는다.

D. Top

Top module은 Bus, Memory, Factorial core module을 instance화하여 연결한 module이다. Top module의 input port를 통해 Bus의 master port에 접근하며, communication의 결과값을 얻을 수 있다. 또한, Top module은 memory mapped I/O 방식을 사용하기 때문에, testbench에서 주소를 통해 top module의 salve device에 접근하여 원하는 동작을 수행할 수 있다. Factorial core의 interrupt는 top module에 직접적으로 연결되어 있어 interrupt 상태를 testbench에 전달할 수 있다.

3. Schedule

	10주차	11주차	12주차	13주차	14주차
제안서					
코드 작성					
코드 검증					
결과 보고서					

Table 1. Project Schedule

본 Factorial computation system 프로젝트는 2023년 11월 07일 (10주차)부터 2023년 12월 06일 (14주차) 동안 진행된다. 프로젝트가 공고된 10주차부터 11주차 동안은 프로젝트 제안서에 대해 작성하며, 12주차부터 13주차 동안은 코드를 작성할 예정이다. 그리고 13주차동안 프로젝트를 제대로 검증할 수 있는 testbench를 설계하며 14주차에 검증을 완료하고 결과보고서를 작성할 것이다 .

4. State transition diagram

A. Factorial core

Inputs	s_sel, s_wr, s_din
States	INIT, WRITE, READ
Outputs	s_dout

Table 2. Factorial core Inputs, States, Outputs

State	Description	Condition
INIT	s1_sel = 0일 때 상태이다.	s1_sel = 0
WRITE	s1_sel = 1, s_wr = 1일 때, s_din의 값을 module 내에 선언된 operand 변수에 저장한다.	s_wr = 1 && s1_sel = 1
READ	s1_sel = 1, s_wr = 0일 때, s_dout에 result의 값을 저장한다.	s_wr = 0 && s1_sel = 1

Table 3. State of Factorial core module Description

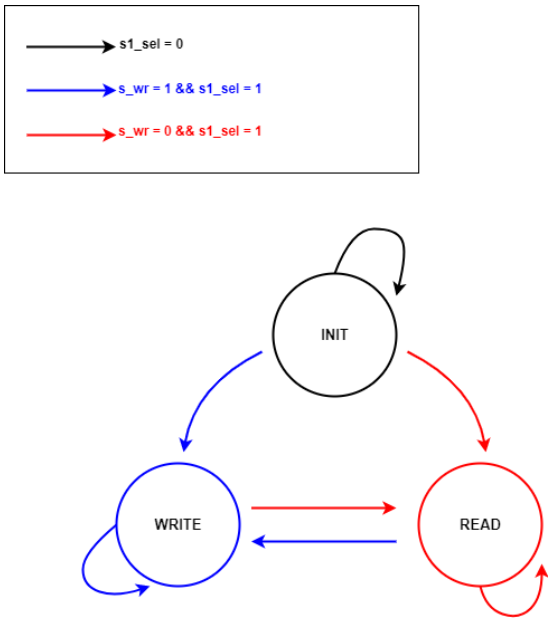

Figure 1. Bus module State Transition Diagram

B. Bus

Inputs	reset_n, m_req, m_addr, m_wr
States	INIT, MEM_RD, MEM_WR, FAC, RGT, NON
Outputs	m_grant, s0_sel, s1_sel, s_wr

Table 4. Bus Inputs, States, Outputs

State	Description	Condition
INIT	Bus module의 초기상태로 s0_sel = 0, s1_sel = 0로 초기화된다.	s0_sel = 0, s1_sel = 0
NON	m_req = 1이고, m_addr이 Slave0과 Slave1의 memory map region을 넘어가면 어떤 slave device가 선택되지 않는다.	m_req = 1 && s0_sel = 0 && s1_sel = 0
RJT	m_req = 0이면, master에 대한 BUS의 소유권을 거부 (reject) 한다.	m_req = 0
MEM_WR	m_req = 0, m_wr = 1이고, m_addr의 memory map regiondl 0x0000 ~ 0x00FF 사이에 포함되면, s_wr = 1로 설정하고 Slave0 device (Memory)를 선택한다.	m_req = 1 && s0_sel = 0 && s1_sel = 0 && m_wr = 1
MEM_RD	m_req = 0, m_wr = 0이고, m_addr의 memory map regiondl 0x7000 ~ 0x703F 사이에 포함되면, s_wr = 0으로 설정하고 Slave0 device (Memory)를 선택한다.	m_req = 1 && s0_sel = 0 && s1_sel = 0 && m_wr = 0
FAC	m_req = 0이고, m_addr의 memory map region에 포함되면, Slave1 device (Factorial core)를 선택한다.	m_req = 1 && s0_sel = 0 && s1_sel = 0

Table 5. State of Bus module Description

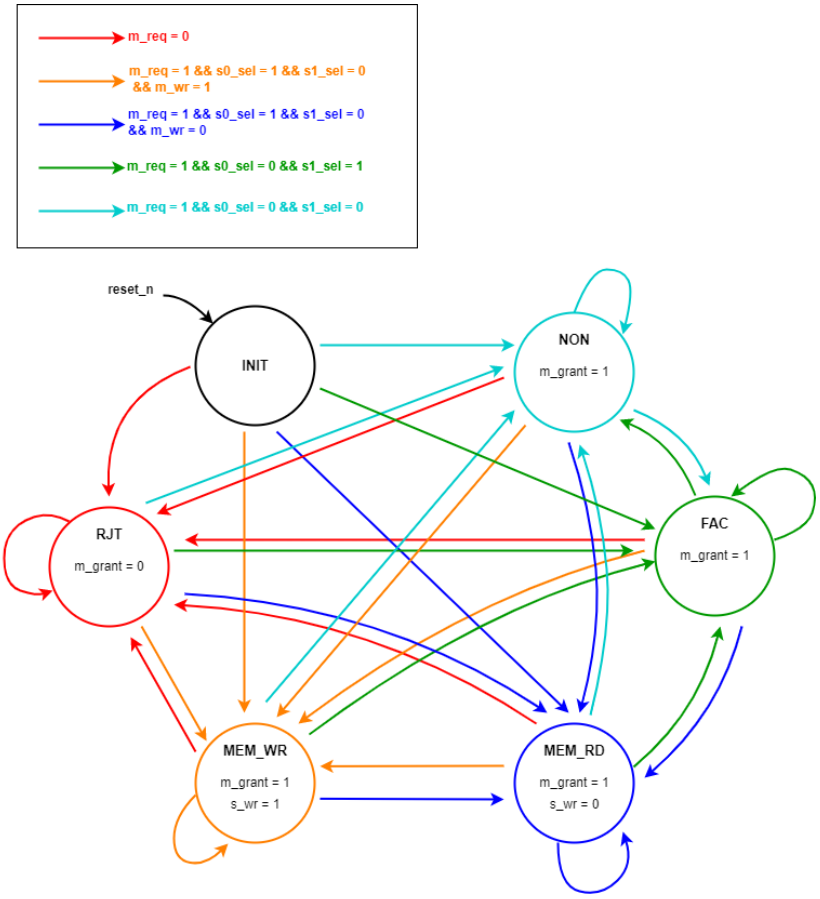

Figure 2. Bus module State Transition Diagram

C. Memory (RAM)

Inputs	cen, wen, s_addr, s_din
States	INIT, READ, WRITE
Outputs	s_dout

Table 6. Memory Inputs, States, Outputs

State	Description	Condition
INIT	Memory module의 초기 상태이며 s_dout = 64'b0으로 초기화하고, cen = 0일 때 상태이다.	cen = 0
READ	s_addr의 주소에 저장된 데이터 값을 읽어 s_dout에 write한다.	cen = 1 && wen = 0
WRITE	s_addr의 주소가 가리키는 memory에 s_din을 write한다.	cen = 1 && wen = 1

Table 7. State of Memory module Description

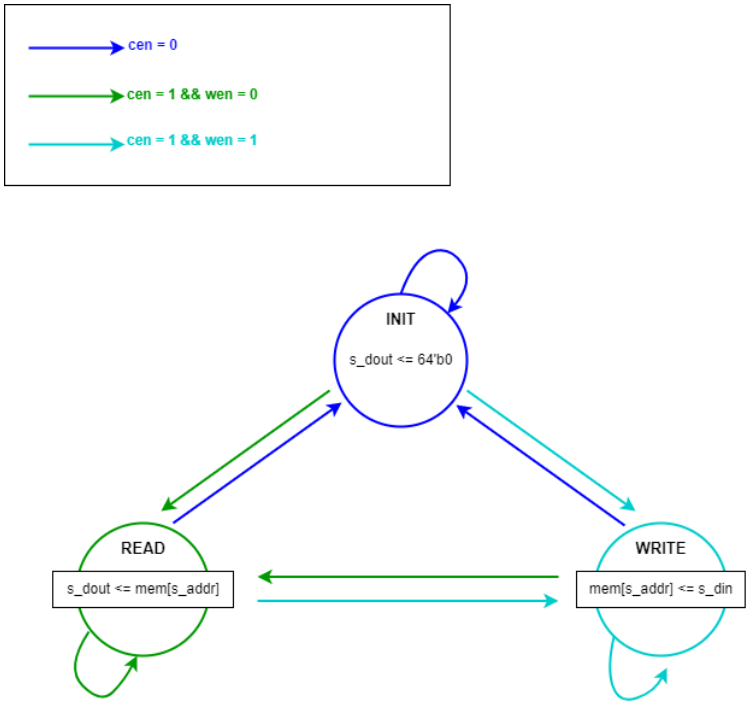

Figure 3. Memory module State Transition Diagram

5. Module instance design

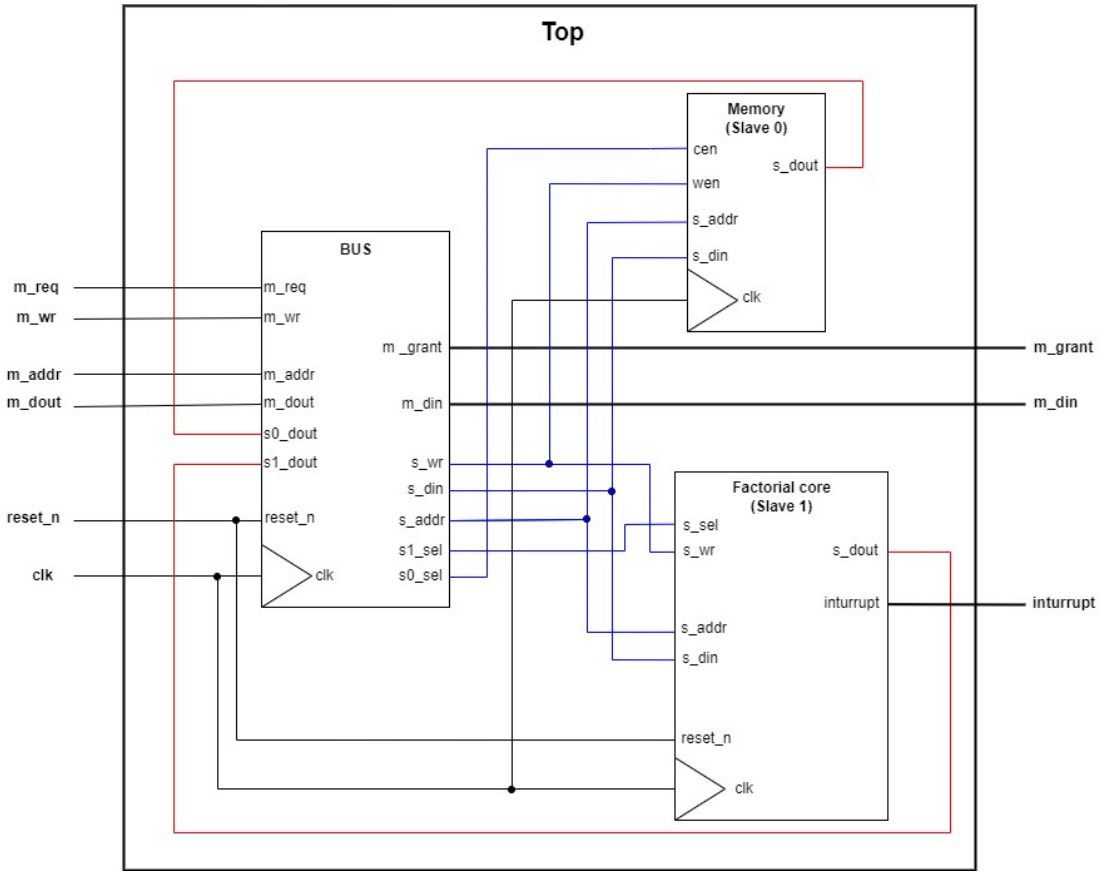

Figure 4. Top module instance design

위의 Figure 4는 Top module 내에 선언할 bus, memory, factorial core module을 instance화하여 나타낸 것이다. Top module에서 bus instance의 input port로 data를 전송하고 다른 component 간의 communication 결과를 받는다.

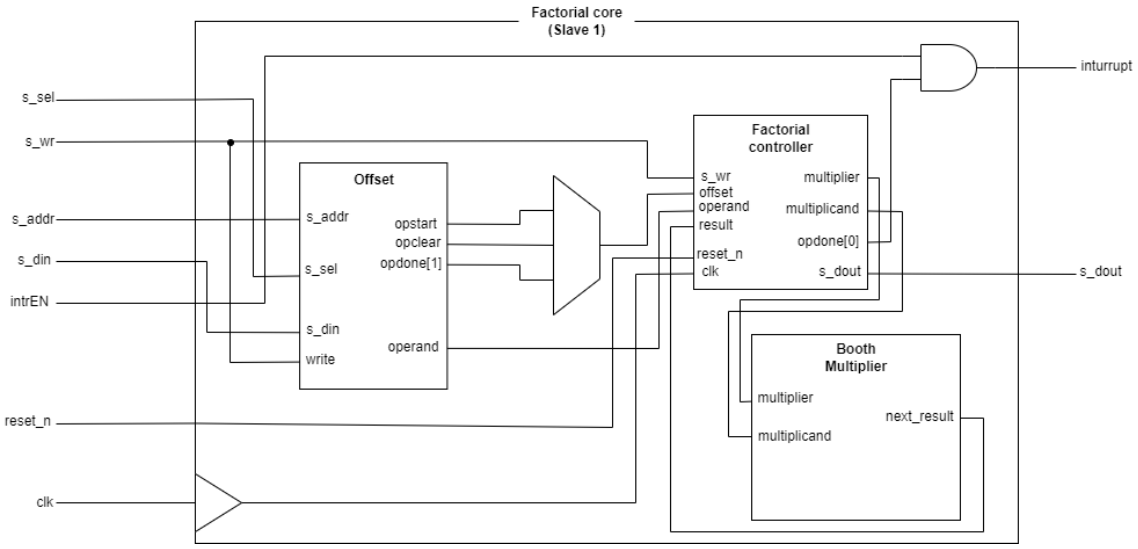

Figure 5. Factorial core module instance design

위의 Figure 5는 Factorial core module내에 선언된 instance를 나타낸다. 위의 design은 설계과정에서 변경될 수 있다.

6. Design verification strategy

본 프로젝트에서 설계한 Top module을 검증하기 전에, 모듈 내에 선언된 각 component들이 정상적으로 동작하는 지 검증하기 위한 testbench가 요구된다. 따라서 Bus, Factorial core, memory에 대해 각각 testbench를 작성하여 factorial 계산, memory에 write된 데이터를 read하는 기능 등을 검증할 것이다. 특히, factorial core module 내부에 선언된 instance들을 모두 검증하여 component의 기능에 오류가 없는 지 확인할 것이다.

7. Expected problems

1. Flip-flop의 부재로 인한 state transition이 제대로 수행되지 않거나 clock의 rising edge에서 값이 변화하지 않는 결과가 발생할 수 있다.
2. 본 프로젝트에서 Top module의 testbench는 testvetor를 사용하여 검증할 예정이다. 이전 과제에서 testvector를 사용하여 검증할 때, hexadecimal로 불러오는 과정에서 값이 변형되는 문제가 발생하였기 때문에, 이번에도 그러한 문제가 발생한다면 binary로 값을 입력받아 검증을 수행할 예정이다.