디지털 논리회로2 프로젝트 제안서

Factorial computation system

학　　과: 컴퓨터정보공학부

담당교수: 이준환 교수님

실습분반: 월요일 0, 1, 2

학　　번: 2022202061

성　　명: 양서은

1. Title & Object

A. Title

Factorial computation system

B. Object

이번 프로젝트에선 Factorial 연산을 수행하는 컴퓨터 시스템을 제작한다. Bus를 통해 operand와 연산 수행 신호를 받아와 연산을 수행하고 Memory에 저장한다. operand는 32bit이고 결과값은 128bit까지 저장 가능하며 64bit씩 나눠 각각 저장한다. Booth multiplier를 사용하여 Factorial core을 제작한다. Bus의 master, slave 관계를 이해하고 값이 전달되는 것을 확인한다.

2. Component concept

A. Factorial core

Factorial의 연산 결과는 총 128bit로 result_h, result_l 2개에 64bit씩 저장된다. 곱셈은 booth multiplier를 사용하고, 기타 덧셈과 뺄셈은 128bit CLA를 사용한다. opstart[0]이 1일 때 operand 값을 받아와 연산을 시작한다. operand는 32bit로 양수값만 주어진다고 가정한다.

opdone[1:0]	
00	연산 대기
10	연산 시작
11	연산 종료

opdone에 저장된 값을 통해 연산 여부를 확인할 수 있다.

연산을 수행할 때 operand 값이 0이라면 곱셈을 수행하지 않고 바로 result_h엔 0, result_l에 1을 저장하고 opdone 값을 변경함으로써 연산을 종료한다.

operand register는 64bit이지만 최대 2^{32}-1의 값만을 저장할 수 있다. 따라서 operand[63:32]는 reserved이다. 다른 register도 상위 몇 비트는 사용되지 않아 reserved일 수 있다.

offset	type	bit width	name	description	default
0x00	write	64	opstart	LSB에 1 쓰이면 연산 시작	0
0x01	write	64	opclear	LSB에 1 쓰이면 모든 register default로 초기화	0
0x02	read	64	opdone	위의 표에서 설명함	0
0x03	write	64	intrEn	연산 종료되었을 때 interrupt 신호 발생시키기 위해 사용 (LSB=1)	0
0x04	write	64	operand	피연산자	0
0x05	read	64	result_h	연산 결과 상위 64bit 저장	0
0x06	read	64	result_l	연산 결과 하위 64bit 저장	1

B. Bus

Bus는 여러 component 사이에서 data를 전송할 수 있도록 연결해준다. master와 slave 한 쌍씩 data와 명령어를 주고받을 수 있다. address를 전달하며 base address에 따라 선택되는 slave component가 달라진다. 잘못된 주소가 접근되는 경우 slave를 선택하지 않는다. master request 신호가 들어오면 master grant 신호를 통해 사용을 허락하고 bus에 대한 소유권을 할당한다. 그 후 slave와 소통을 시작한다.

C. Memory (RAM)

임의의 주소에 대해 data를 읽고 쓰는 memory이다. address에 기반하여 data를 저장한다. wen 신호가 1이면 write, 0이면 read 동작을 수행한다.

D. Top

Factorial core, Memory, Bus를 instance하여 연결한 Top module이다. 모듈의 input으로 bus의 master port에 접근한다. 따라서 testbench의 input으로 address가 들어가기 때문에 slave component까지 접근 가능하다. 또한 interrupt 신호로 Factorial core의 동작 여부를 top module

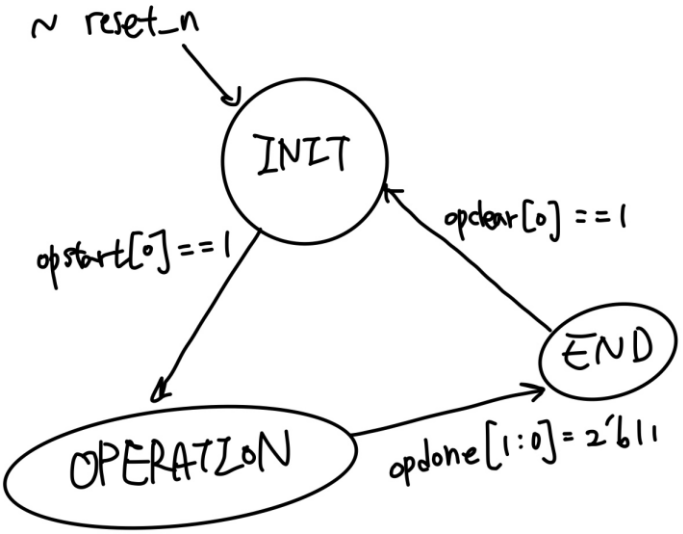
외부로 전달할 수 있다,

3. Schedule

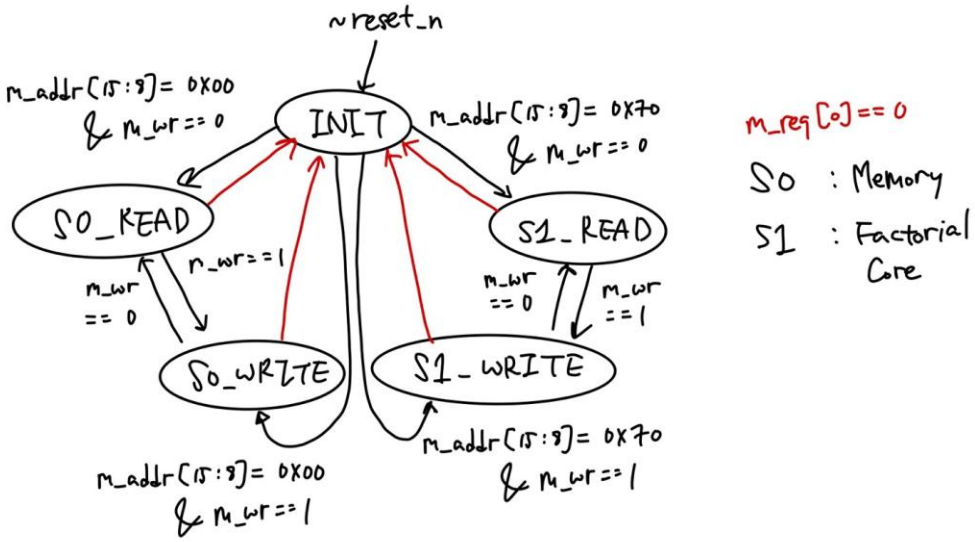
	~11/16	11/17~11/22	11/23~11/29	11/30~12/06
제안서				
코드 작성				
코드 검증				
결과보고서				

4. State transition diagram

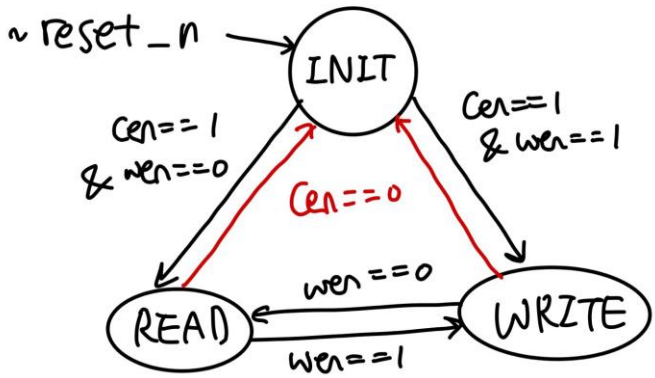
A. Factorial core



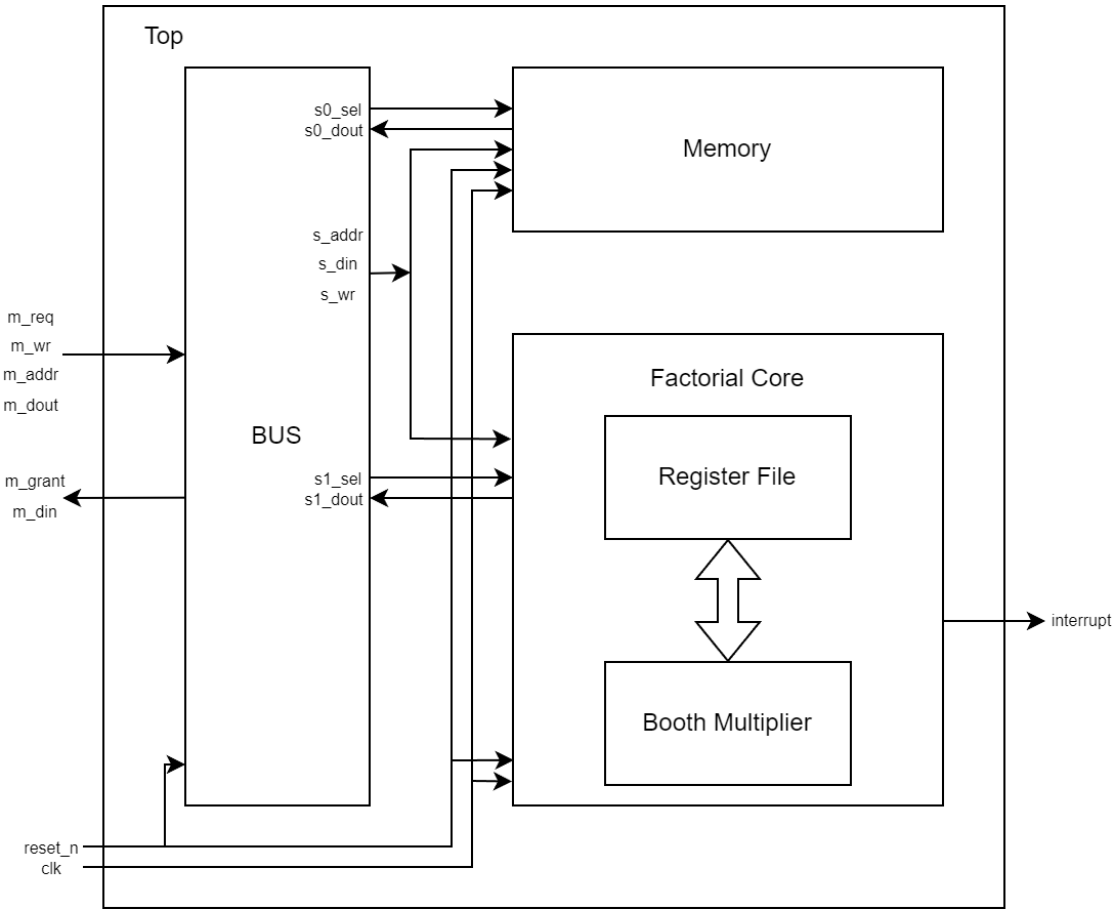
B. Bus



C. Memory



5. Module instance design



6. Design verification strategy

Factorial core, Bus, Memory 모듈을 각각 제작하고 서브 모듈에 대한 testbench를 작성하고 Top module에 instance한 후 전체 testbench를 통해 시스템이 정상적으로 동작하는지 확인한다.

A. Factorial Core

Booth Multiplier, controller, Register File에 대한 서브 모듈을 작성 후 testbench를 통해 동작을 확인한다. operand와 연산 관련 신호를 제대로 인식하고 연산에 따라 신호가 제대로 변경되는지 확인한다. result_h, result_l에 결과값이 제대로 저장되는지 확인한다.

B. Bus

입력 신호와 출력 신호가 제대로 동작하는 것이 중요하다. master의 입력에 따라 해당하는 slave에 address와 기타 입력값들이 제대로 전달되는지 확인한다.

C. Memory

cen, wen 신호에 따라 메모리에 정보가 정상적으로 저장되고 불러오는지 확인한다. input으로 들어온 address에 값이 저장되는 것을 확인한다.