디지털 논리회로2 프로젝트 제안서

Multiplier machine

학 과: 컴퓨터공학과

담당교수: 이혁준 교수님

실습분반: 목요일 0, 1, 2

학 번: 2013722095

성 명: 최재은

1. Title & Object
   1. Title

Multiplier machine

* 1. Object

지금까지 실험에서 구현해본 FIFO, ram, bus등과 새롭게 구현할 multiplier, Direct memory access controller(DMAC)를 활용하여 32bit입력 값 multiplicand, multiplier를 입력 받고 그

곱을 연산하여 메모리에 저장하는 동작을 수행하는 machine을 구현하여 본다.

1. Component concept
   1. FIFO

- First In First Out의 약자로 먼저 입력된 데이터가 먼저 출력되는 Queue의 형태를

갖는다. 외부에서 write/read enable을 입력 받고 현재 state와 갖고 있는 데이터의 개수를 기반으로 다음 state를 계산하며, 이를 통해 head와 tail(register file의 address0)의 값과 현재 갖고 있는 data의 개수를 계산한다.

* 1. Bus

- 여러 component들 간에 data를 전송할 수 있도록 연결해주는 component로 slave마다 할당된 메모리 주소 범위를 갖고 있다. Master의 요청에 따라서 arbiter가 우선권을 할당해 주며, 우선권을 먼저 갖고 있는 master가 지속적으로 request를 갖는다면, 그 우선권은 request가 없어질 때까지 유지된다.

* 1. Multiplier

- 피승수(multiplicand)와 승수(multiplier)를 곱하여 결과값을 도출하는 Hardware로

multiplication을 하는 데에는 binary / booth, 두 가지 방법을 배웠으나, 연산 속도를

향상 시키기 위해 multiplier의 1비트를 최소화 시키는 booth multiplication을 사용한

다. 이에는 몇 가지 규칙 이 있는데 이는 다음과 같다.

|  |  |  |  |
| --- | --- | --- | --- |
| Xi | Xi-1 | Operation | Description |
| 0 | 0 | Shift only | String of zeros |
| 0 | 1 | Add and shift | End of a string of ones |
| 1 | 0 | Subtract and shift | Beginning of a string of ones |
| 1 | 1 | Shift only | String of ones |

* 1. DMAC

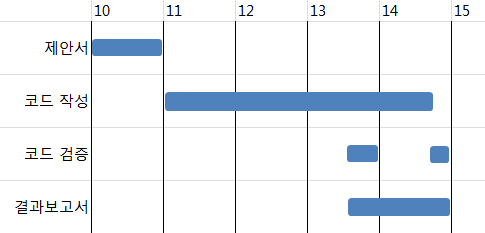
- testbench로부터 받은 start signal을 받아, memory에 저장되어있는 multiplicand와 multiplier를 설정된 opmode에 따라서 MULTIPLIER module로 보내며 이때는 DMAC 모듈이 master의 역할을 수행한다. MULTIPLIER module에서 연산을 마치고 Register file에 저장해둔 결과값을 memory에 전송하는 역할을 수행한다.

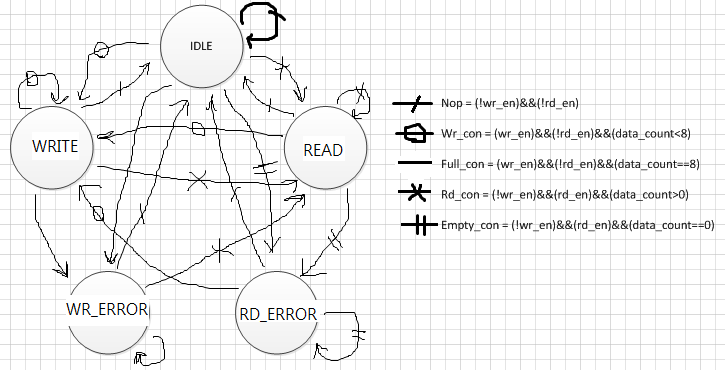
* 1. RAM

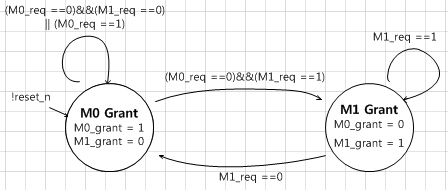
- address에 기반하여 data를 저장하는 hardware로서 32bit data를 최대 32개까지

저장할 수 있도록 구현되어있다. 이 프로그램에서는 testbench로부터 받아온 multiplier와 multiplicand를 저장하고 MULTIPLIER module에서 출력된 연산 결과 값을 저장하는 역할을 수행하며 총 3개가 필요하다.

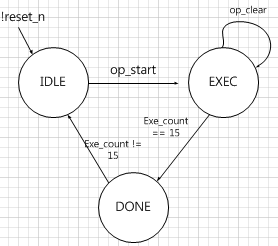
1. Schedule



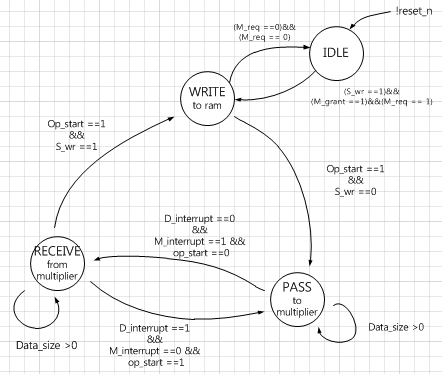
1. State transition diagram
   1. FIFO
   2. Bus



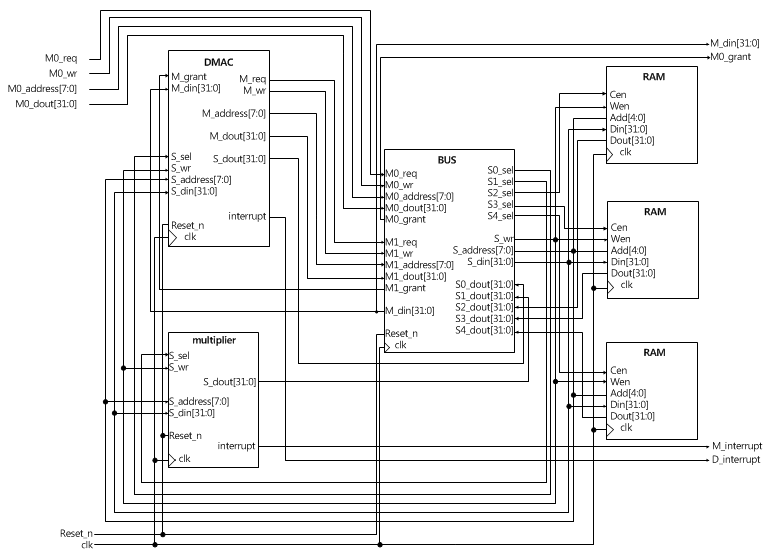
* 1. Multiplier



* 1. DMAC



1. Module instance design



1. Design verification strategy

먼저 구현하였던 BUS, RAM module을 연결해서 입력이 잘 저장이 되는지, 연산 후에

들어올 값이 다른 memory에 잘 저장이 되는지를 먼저 확인할 것입니다. 이후 MULTIPLIER를 구현하고 나서 입출력 인자들간의 연결을 해주고, 버스가 메모리에서 multiplicand와 multiplier를 잘 읽어오는지, 읽어온 값이 잘 연산이 되는지, 연산한 값을 bus가 다른 세번째 memory에 잘 저장하는지를 확인하고 나서 DMAC의 기능이 잘 되는지를 확인하고, bus와 연결해주어 전체적으로 잘 동작하는가를 확인해 볼 예정입니다.