컴퓨터 공학 기초실험2 프로젝트 제안서

Mini-Processor block with Bus

학 과: 컴퓨터공학과

담당교수: 공진흥 교수님

실습분반: 월요일 0, 1, 2

학 번: 2019202052

성 명: 김호성

1. Title & Object
   1. Title

Mini-Processor block with Bus

* 1. Object

Bus와 간단한 processor 역할을 하는 mini-processor block MP를 이용하여 곱셈, 덧셈, 뺄셈 및 논리연산을 한다.

1. Component concept

Project에 포함된 각 module별 기본 개념을 설명할 것

* 1. Bus

1개의 master와 1개의 slave가 존재하며, 데이터의 흐름을 제어하는 기능을 수행한다.

>> arbiter의 역할

먼저 bus\_arbit에서 master를 사용할지 결정한다. m\_req가 1이면 읽기/쓰기 동작이 이루어지지만, 그렇지 않을 경우에는 읽기/쓰기는 이루어지지 않게 된다.

>> decoder의 역할

arbiter에서 읽기/쓰기에 대한 허용여부를 결정했다면, decoder에서 slave를 허용할지 결정해준다.

* 1. ALU

A, B에 숫자를 받고, 어떤 기능을 수행할지 결정해주는 op값을 넣은 후 일단 모든 기능들에 값이 들어간다. 이후 mx8에서 op값에 의해 특정 기능을 수행한 값을 도출한다. 또한 ADD와 SUB의 경우 conditional register 값이 필요하기 때문에 cal\_flags64bit 모듈에 값을 넣어준 후, C, N, Z, V값을 도출한다.

* 1. Multiplier

하위 32bit만을 multiplier와 multiplicand값으로 이용하여 곱셈을 도출해낸다. 여기서 곱셈 결과의 bit length는 64bit이다.

* 1. Register

DATA, INST, CONT로 나뉘어져 있으면 모든 REG는 64bit로 이루어져 있다. 단, INST는 하위 16비트만, CONT는 하위 1비트만 사용하여 나타낸다.

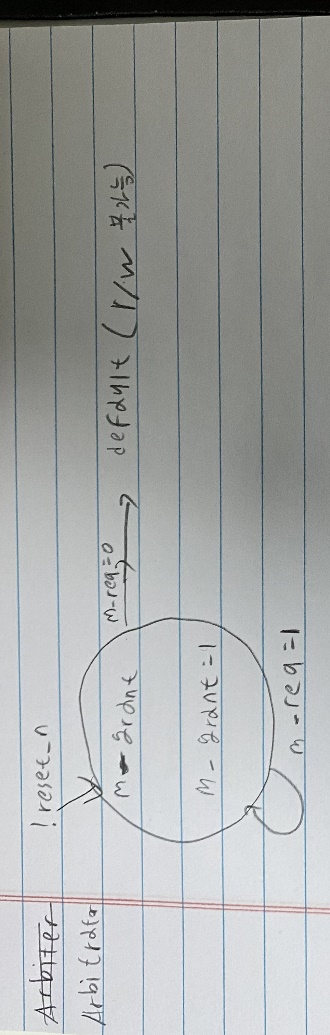
1. Schedule

|  |  |  |  |
| --- | --- | --- | --- |
|  | 11 | 12 | 13 |
| 제안서 |  |  |  |
| 코드작성 및 검증 |  |  |  |
| 결과보고서 |  |  |  |

1. State transition diagram

각 module 별 state transition diagram을 작성

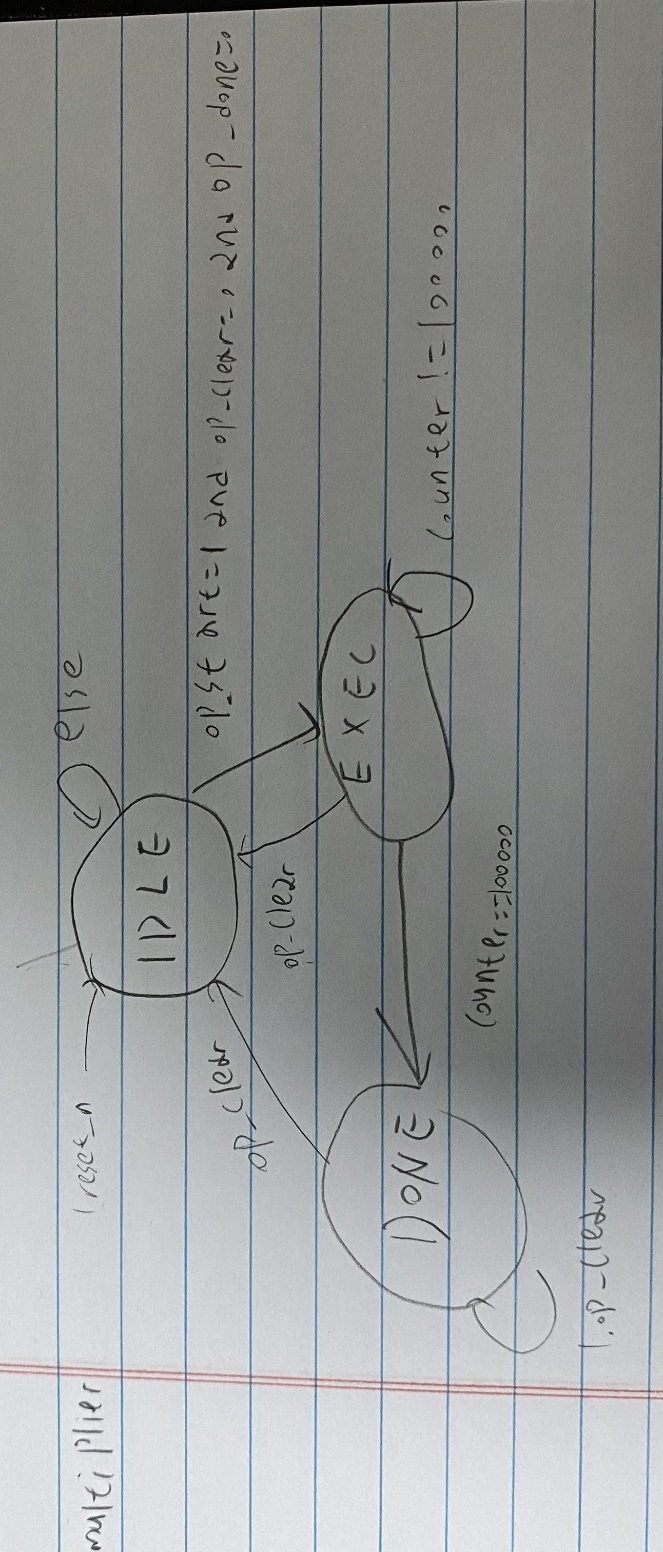
* 1. Bus



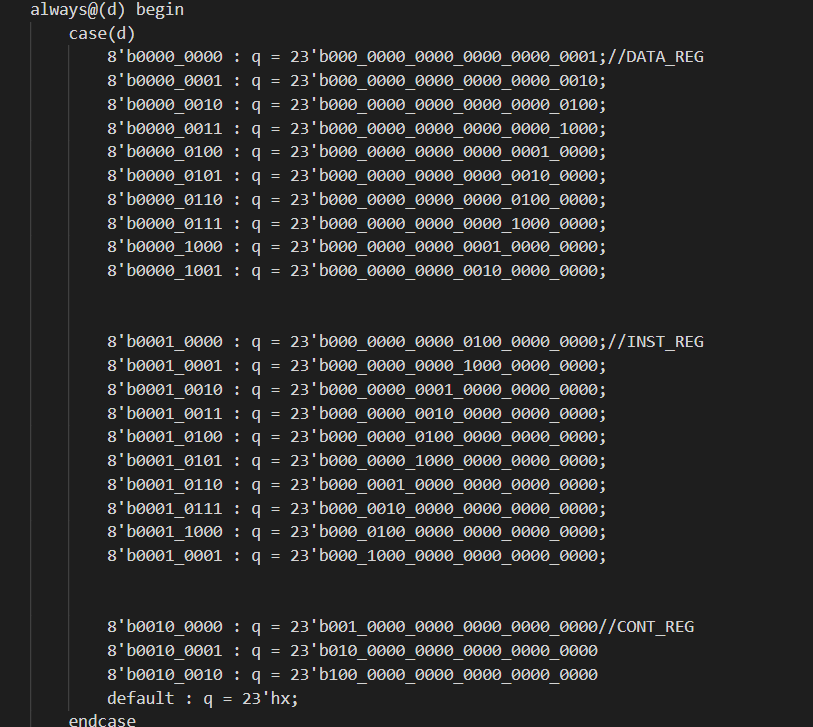
* 1. ALU

State transition diagram이 존재하지 않다.

* 1. Multiplier



* 1. Register



1. Module instance design

1. BUS

BUS의 경우 arbiter와 decoder로 나뉘어져 있으며, arbiter에서 m\_grant를 활성화 시킬지를 결정하고, decoder에서 sel을 활성화시킬지 아닐지를 결정한다.

2. RF

말 그대로 Register덩어리이며, Offset을 맞추기 위해 8to23decoder를 만들 것이다. One hot encoding으로 코드를 구현할 예정이며, 현재 DATA\_REG, INST\_REG, CONT\_REG가 연속되어 있는 숫자가 아니기 때문에 State transition diagram중 D번에서 사용한 사진처럼 코드를 구현할 계획이다.

3. ALU

ALU의 경우 지난 과제에서 했던 것과 동일하며, 64bit로 바꾸고, opcode 4비트중 LSB 3bit만 사용하기 때문에 기존 코드를 다음과 같이 변환하면 된다.

mx8\_64bits U8\_mx8\_64bits (.y(ALU\_result), .a(w\_not\_a), .b(w\_not\_b), .c(w\_and), .d(w\_or), .e(w\_xor),.f(w\_xnor), .g(w\_add), .h(w\_sub), .s2(op[3]), .s1(op[2]), .s0(op[1]));

4. MUL

MUL의 경우 지난 과제에서 했던 것과 동일하며, 32bit로 바꾸면 된다. 단 counter역시 7bit -> 6bit로 바꾸어 주어야 한다.

1. Design verification strategy

1. 각각의 모듈(ALU, MUL, RF, BUS)별로 tb를 만들어 모듈별로 정상적으로 동작하는지 확인한다.

2. tb\_top을 만들어 DATA\_REG 10개는 초기화, INST\_REG 10개는 16비트 code를 저장한다.

3. CONT\_REG에 있는 op\_start에 0x1에 testbench로 값을 바꾼 후, INST\_TRG에 있는 명령에 맞게, ALU와 MUL을 수행하여 DATA\_REG에 저장한다.