260195280 - Clinton Yuen

Xxxxxxxxx – Keren He

Xxxxxxxxx – Chelsea Ma

Prof. Joseph Vybihal

COMP 273 – Classic CPU Project Report

The structure of our circuit is as follows:

-To the top left is our RAM, holding data to be read into registers. For example, when our control unit performs a LOAD instruction.

-To the right of the RAM is our ROM, holding our "program" that executes the multiplication task. Each line of ROM holds an OP code, which is fed into our Control Unit. The way the circuit knows to read the next line of ROM and to get the next OP code is through the program counter, which gets fed into the MAR.

-Below the RAM is our 3 digit display, **prints from MBR. (***)**

-Below the 3 digit display is our keypad. In order to convert decimal to binary for our bus, we implemented the double dabble algorithm.

-Below the ROM is the Control Unit. Through the use of the control unit and a generous use of multiplexers, we were able to control the data flow in the bus and feeding it into the required components, depending on the state of the control unit.

-Located near the bottom middle is the ALU. The ALU is capable of a variety of operations, including Addition and Subtracting. Unique to our project is the amount of different registers our ALU is able to pull values from. This was necessary, along with 2 different OP codes for both addition and subtraction (4 in total), to execute the program required for the project. The reason being that to multiply (for example, 5x3), we needed to add 5 to 5 and save that value to a register. Since the second addition requires us adding 10 + 5, we need another addition to add the previous sum (located in a third SUM register) and the register where 5 is located. For similar reasons, we needed 2 OP codes for subtraction.

-Another problem our group ran into is that a few operations required more ticks than others. In an attempt to amend the situation, we created clock delays such that they would only tick once 2 real clock ticks.

Due to pc problem when start better load ram first from the 00000000 address and use cu manually.

OP Codes:

0000 - Load

0001 - Store

0010 - Add1

0011 - Subtract1

0100 - BEQ

0101 - BNE

0110 - Print

0111 - Input

1000 - Stop

1001 - Fetch

1010 – Add2

1011 – Subtract2

1100 – MULT