

I completed this assignment entirely on my own except for discussions with Iramin Xu.

Ethan Wong
CSMS1A
11/23/20

CSMS1A HW8

High-level description of Circuit

When a user initially presses a number on the keypad, the input gets sent into the BCD encoder, which consequently sends the number to the display register. The display register leads into the 4×7 segment decoder and subsequently the 4×7 segment display, which allows the user to see their input in the calculator. Every time the user hits another number on the keypad (if they choose to) the number gets updated, saved in the display register, and sent into the 16×2 to 1 vector multiplexer. Once the user presses 'E' after finishing their input, the vector multiplexer will select it and push it onto the stack. A signal also gets sent to the 4 to 2 encoder, which is responsible for passing in the proper signals to the stack's control unit. These signals tell the stack what it needs to do. Whenever the stack is updated, the 2 topmost elements on the stack are sent into the ALU for computation in anticipation of an operator being selected by the user. If the user does press an operator, one of the four previously computed values is chosen (according to the operator). The value gets sent into the 16×2 to 1 vector multiplexer. This value is then chosen and pushed onto the stack. A slightly delayed signal is emitted and sent to the 4 to 2 encoder that influences how the stack behaves. Rather than sending the signal immediately, it is delayed slightly to make sure that the stack is completely accurate and reflects recent changes first. The display register is then cleared. If the user decides to hit the equals sign on the keypad, the stack is cleared (by popping). The display register then receives the last computed value (what was on top of the stack) and sends it out to be displayed.

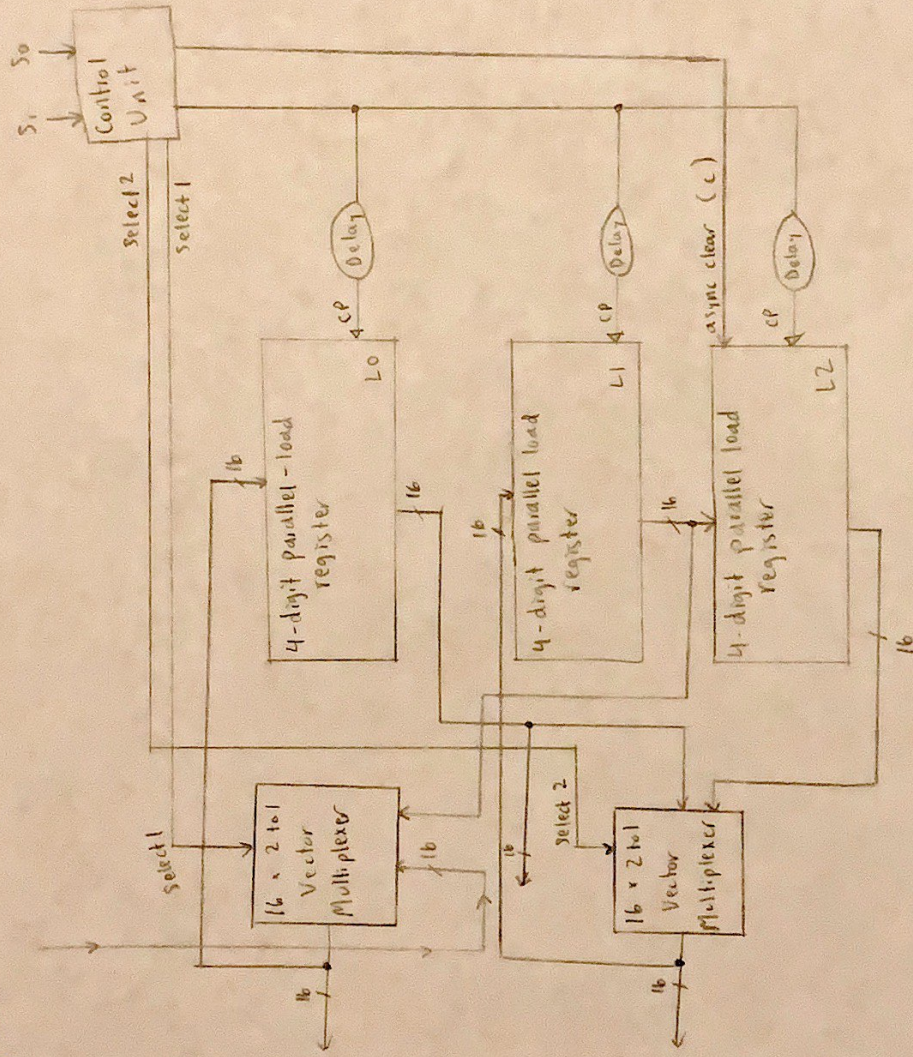
Detailed Explanation of Stack Implementation / Functionality

The stack behaves in different ways depending on what the user is doing. The different behaviors are controlled by the signals the stack receives.

- ① 00 - standby mode - either nothing is happening or the user is in the middle of inputting their number on the keypad. Nothing needs to be clocked to the stack yet.
- ② 01 - E - The user has pressed "E" on the keypad. This tells the stack to push its contents downwards to make space for the new number. The value in L1 gets pushed into L2 and the new number is pushed into L0 when CP gets clocked. The vector multiplexer on the side contains the old value from L0 or the old value from L2. Select 2 will be 0 in this case, which means the old value from L0 gets brought into L1. Basically, when "E" is pressed everything in the stack shifts down so the new number can be pushed on top of the stack.
- ③ 10 - = - The user has pressed the equals sign. The contents of L0 will be sent to the display register and get displayed on the calculator screen. Select 1 will be 1 in this case, which means the contents of L1 will be brought up into L0. Select 2 will be 1 in this case, which means the contents of L2 will be brought up into L0. L2 will be empty and therefore asynchronously cleared.
- ④ 11 - (+ - × ÷) - The user has pressed one of the 4 operators. The computed value will have been calculated and is waiting to be brought into L0. It will be brought into L0 when CP gets clocked. L2 will be asynchronously cleared because it is empty and then clocked. Select 2 is 1 in this case, which means the contents previously in L2 will be pushed into L1 and then clocked with CP.

Implementation of Stack

CP: Clock pulse
AC: Async Clear
S1: Select 1
S2: Select 2



S1	S0	Description
0	0	Nothing happening, don't need to clock CP
0	1	'E' got pressed CP
1	0	Equal sign pressed AC · CP · S1 · S2
1	1	Operator (+ - x ÷) pressed AC · CP · select 2

Delay: → → → → →

