Designing a calculator FSM

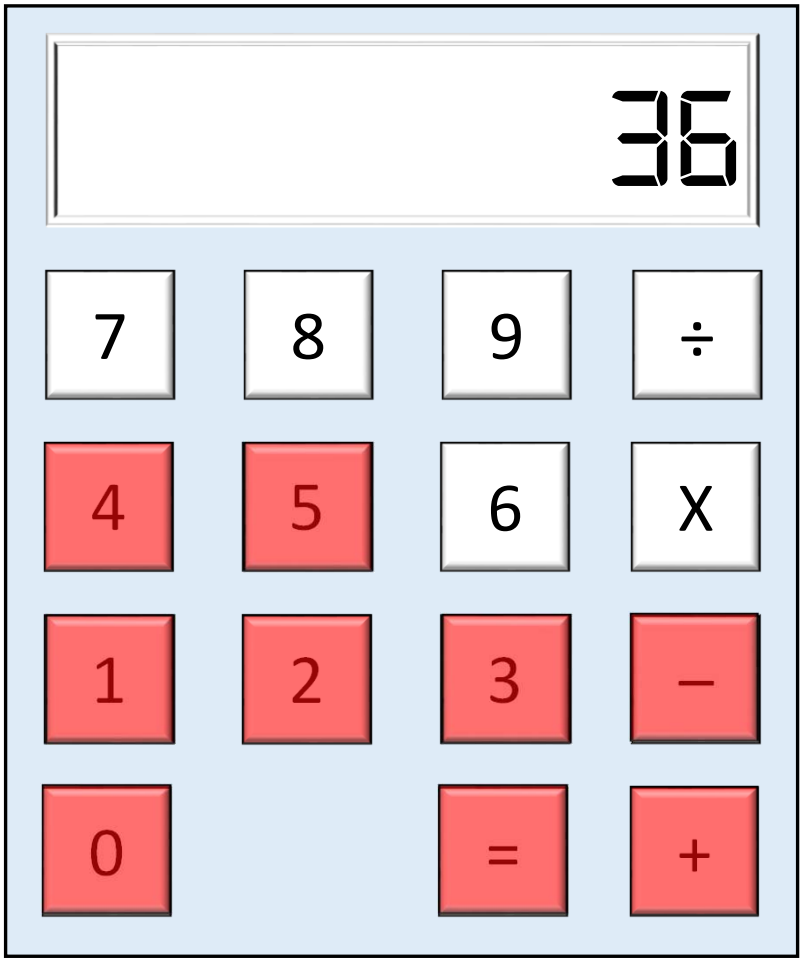
https://www.theonlinecalculator.com/

PLEASE PLEASE PLEASE

NAME YOUR STATES MEANINGFULLY

Consider a 4-function calculator

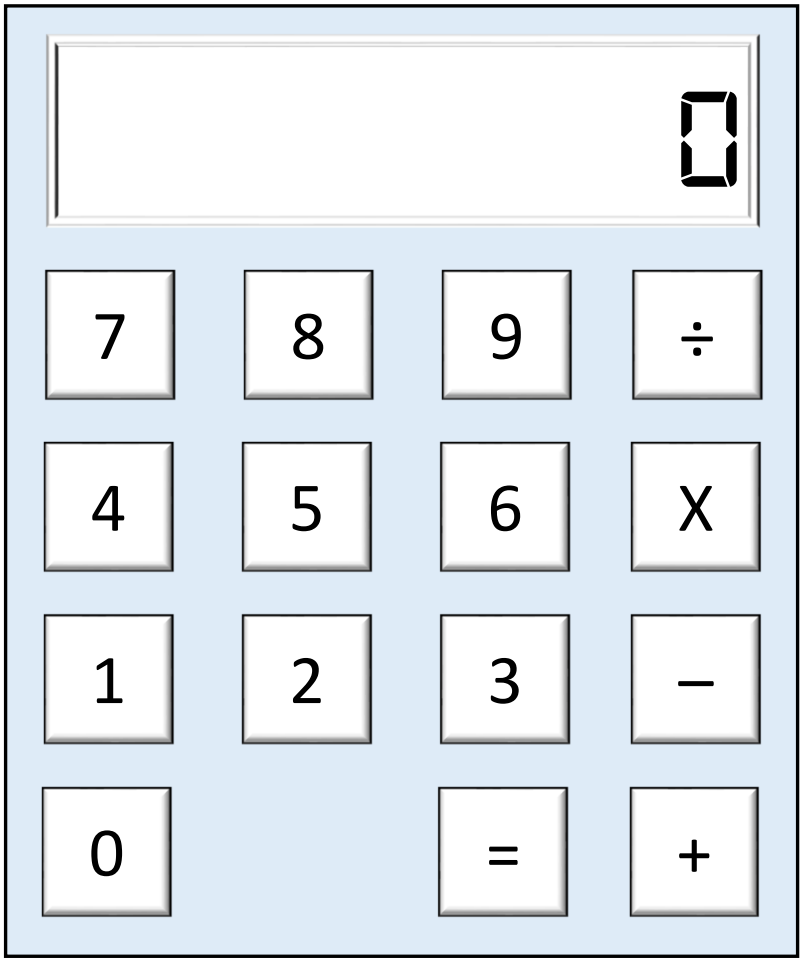
How does the calculator know what to display?



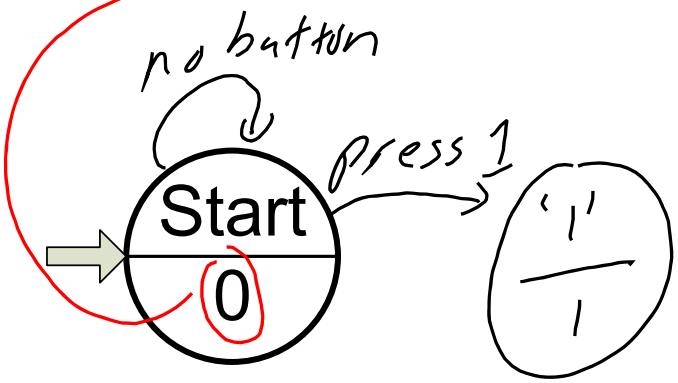
Design the FSM for parsing an input string for a calculator

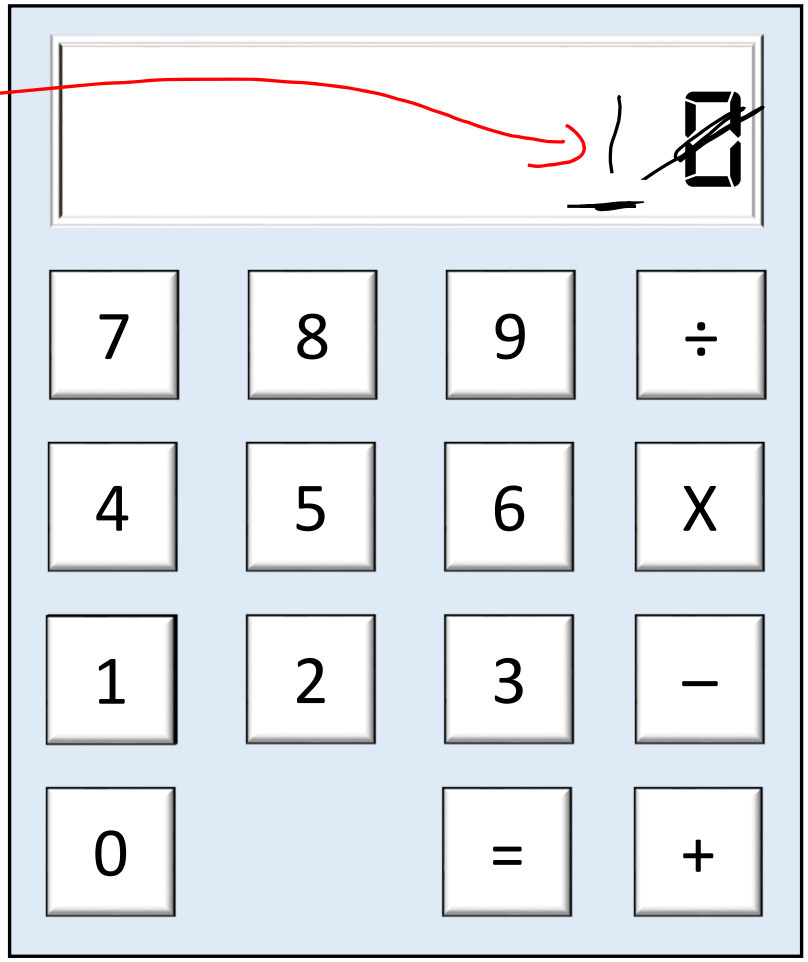
What must the finite state machine remember to execute the operation

1+2?



Remember the first operand

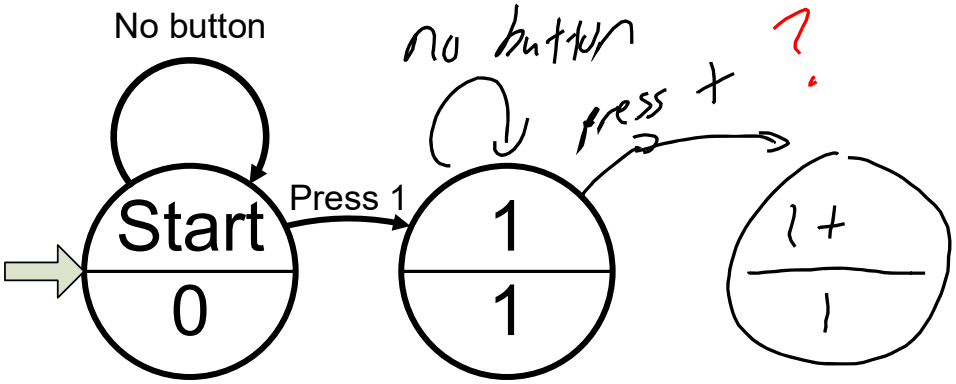


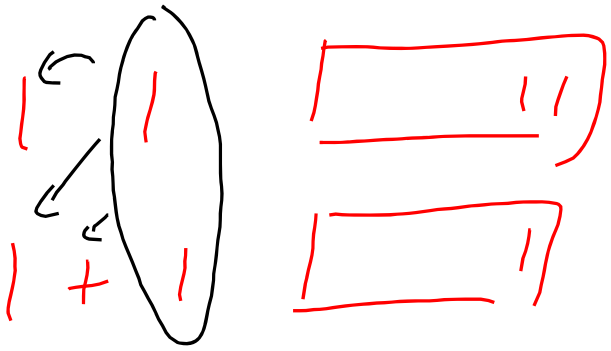


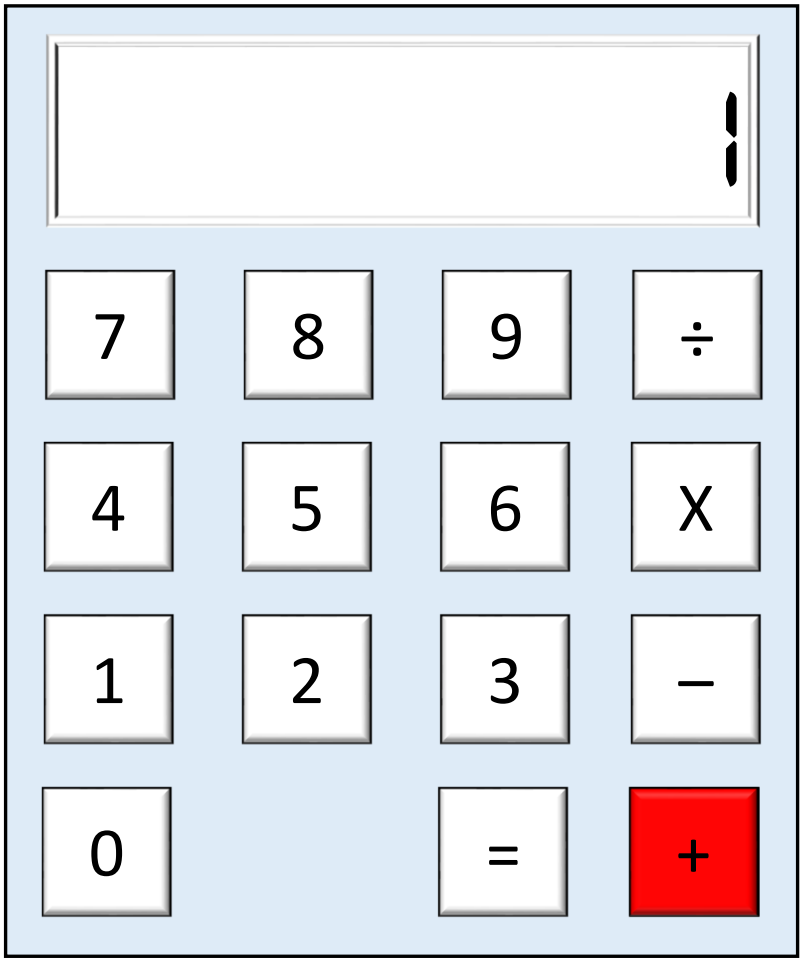
a) No new state b) new state

A) No new state B) New state

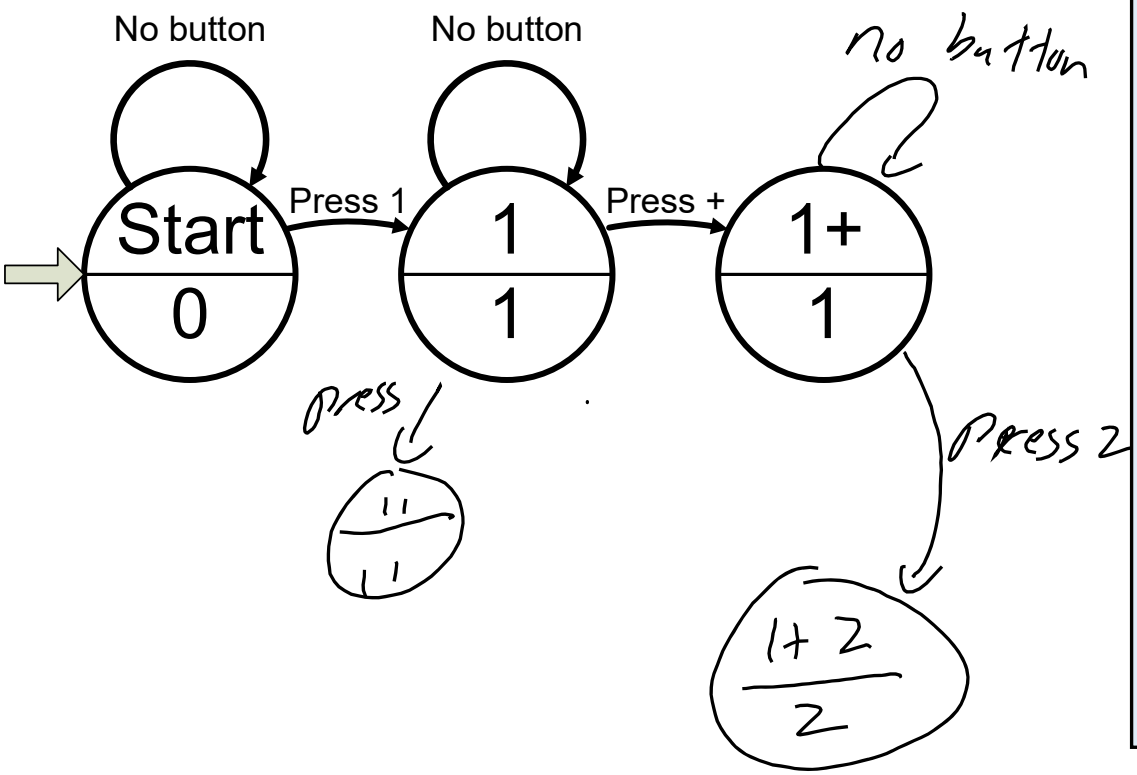
Remember the operator

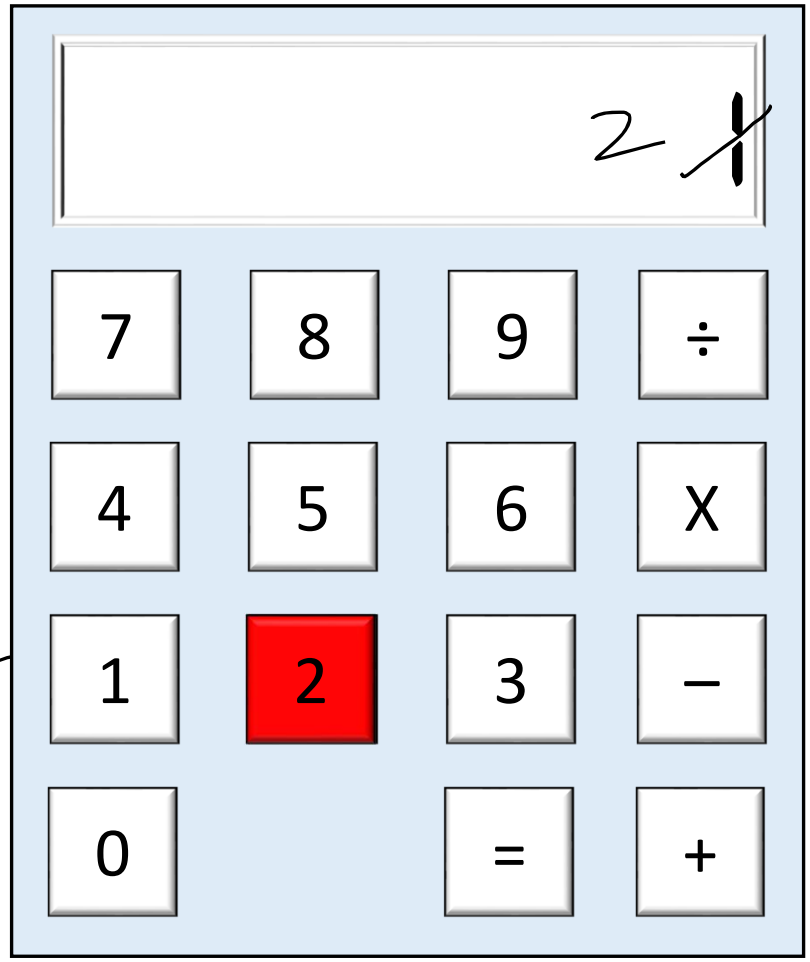




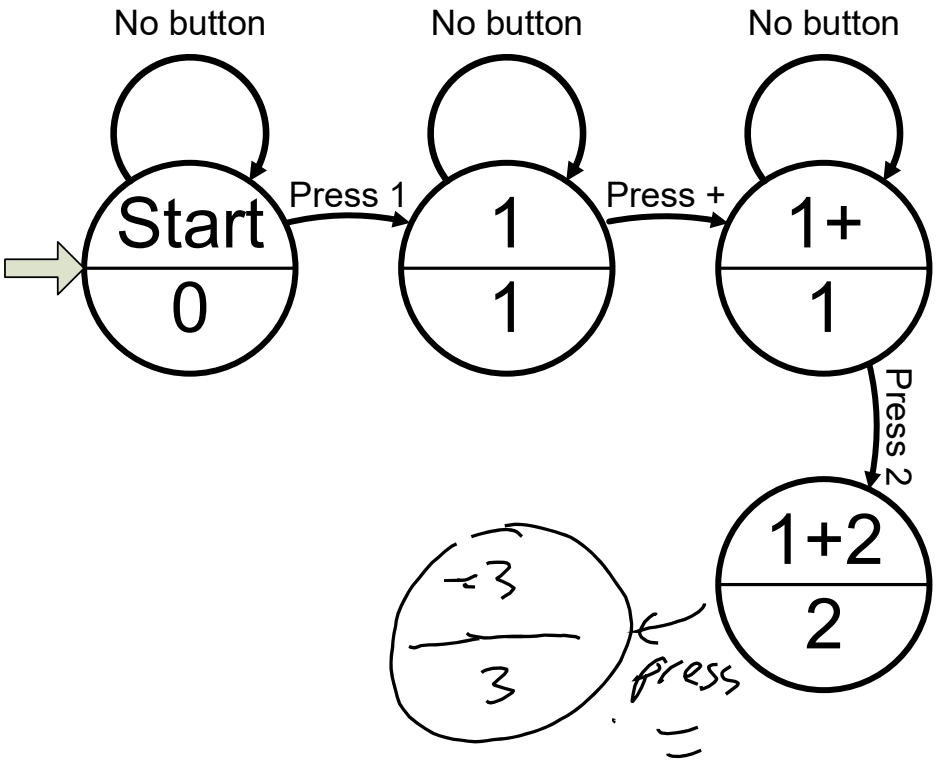


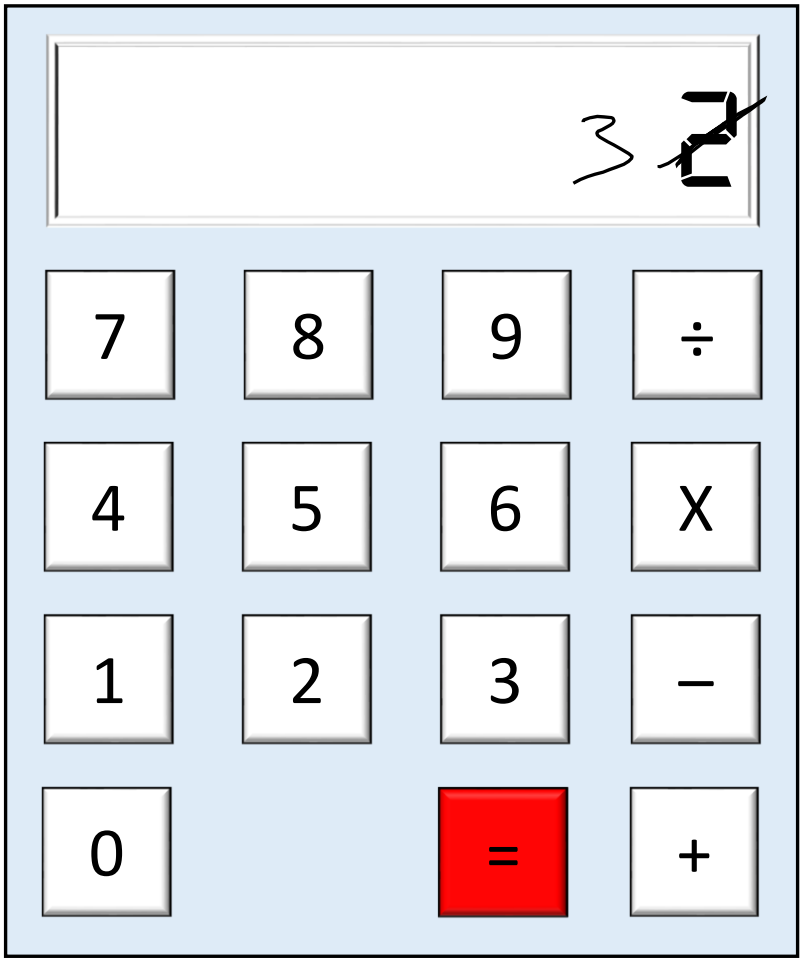
Remember the second operand



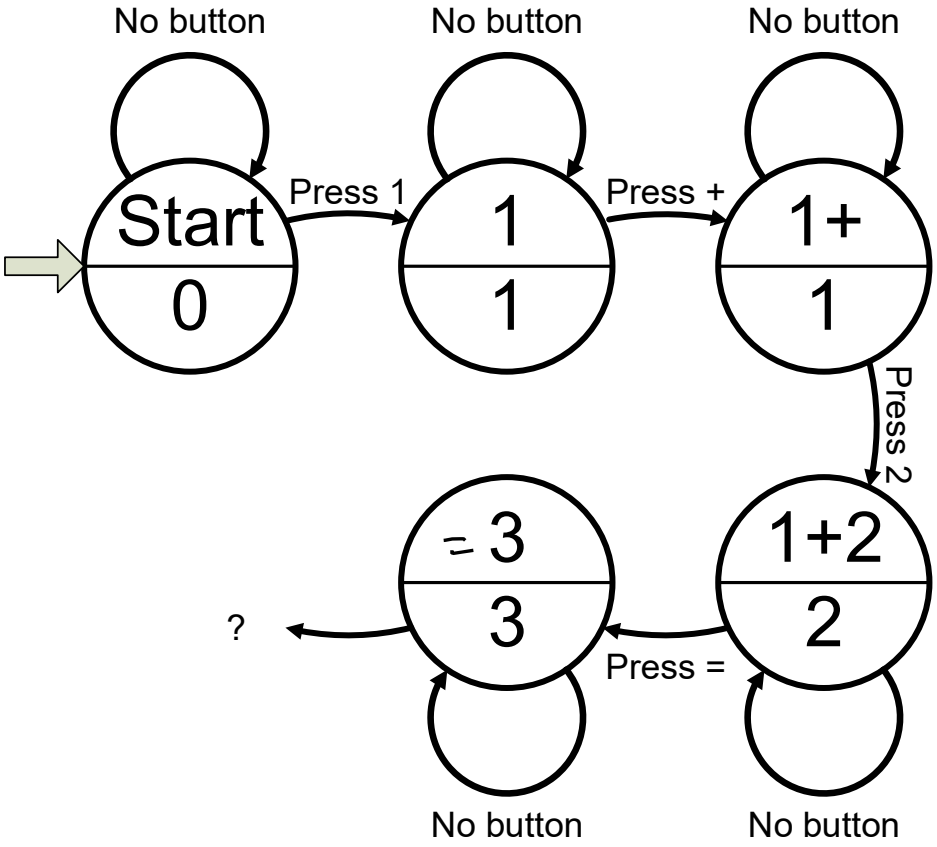


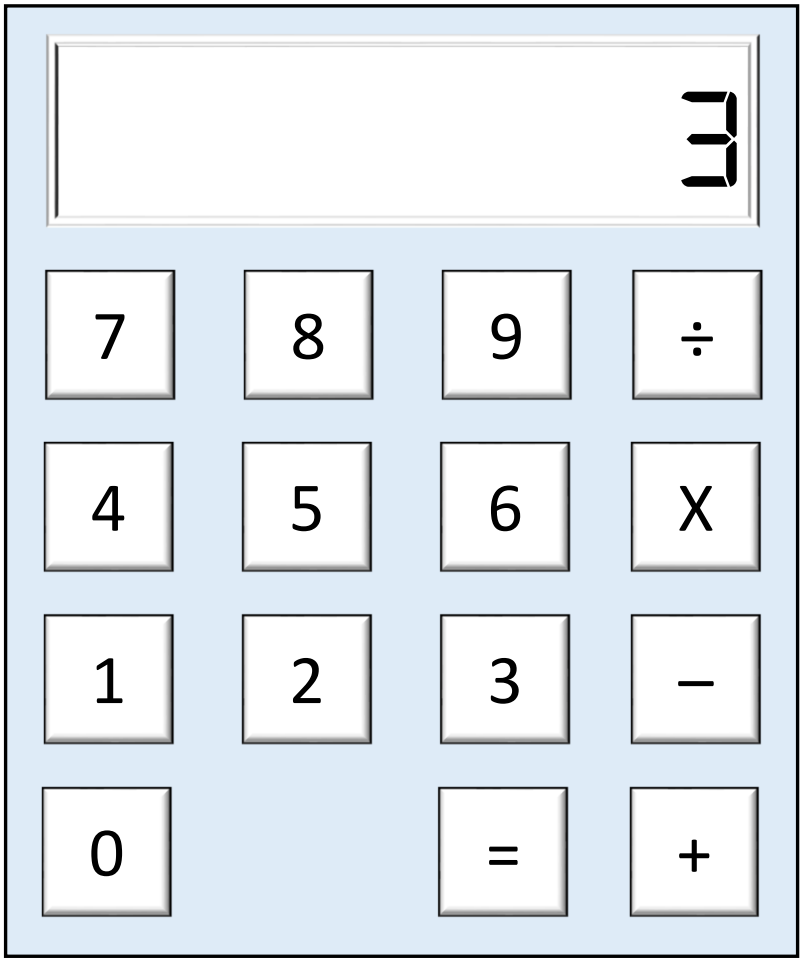
Remember the result



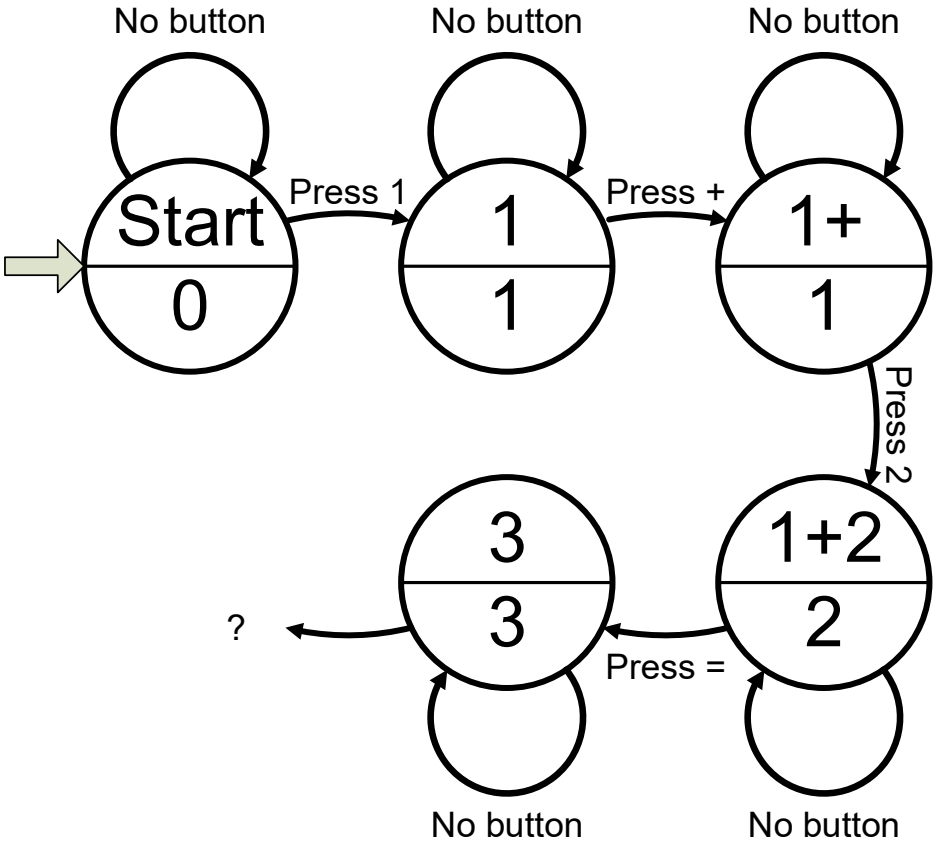


Our first attempt at a FSM for the calculator

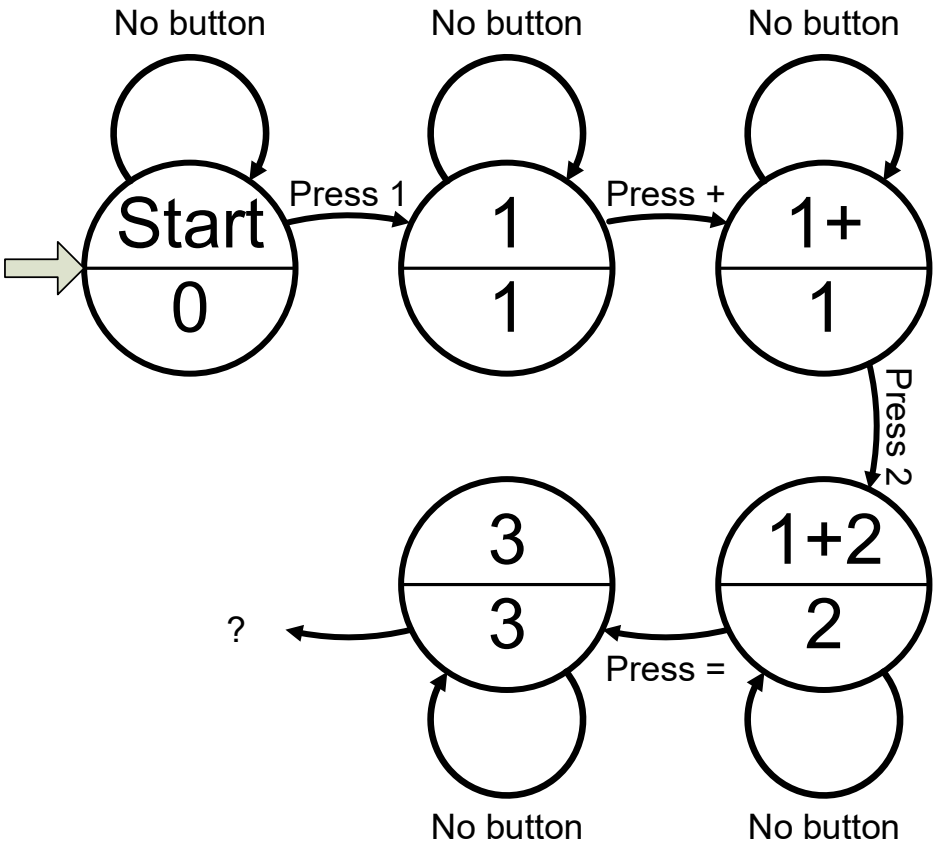




What must the FSM remember to execute the operation 1+2?



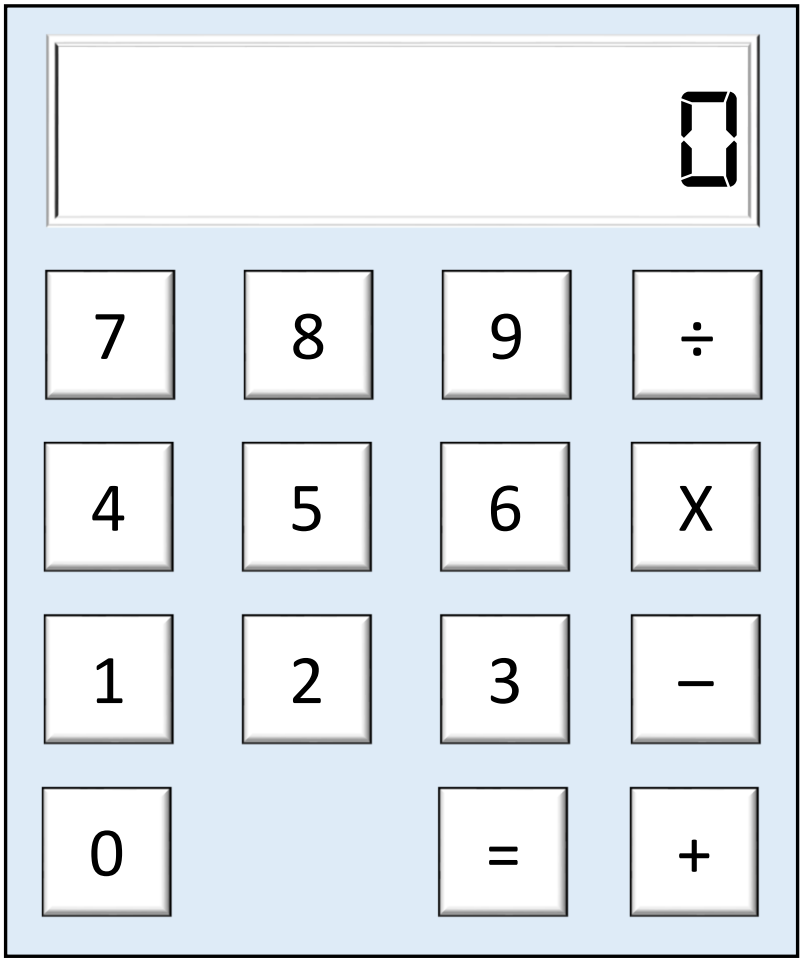
What must the FSM remember to execute the operation 1+2?



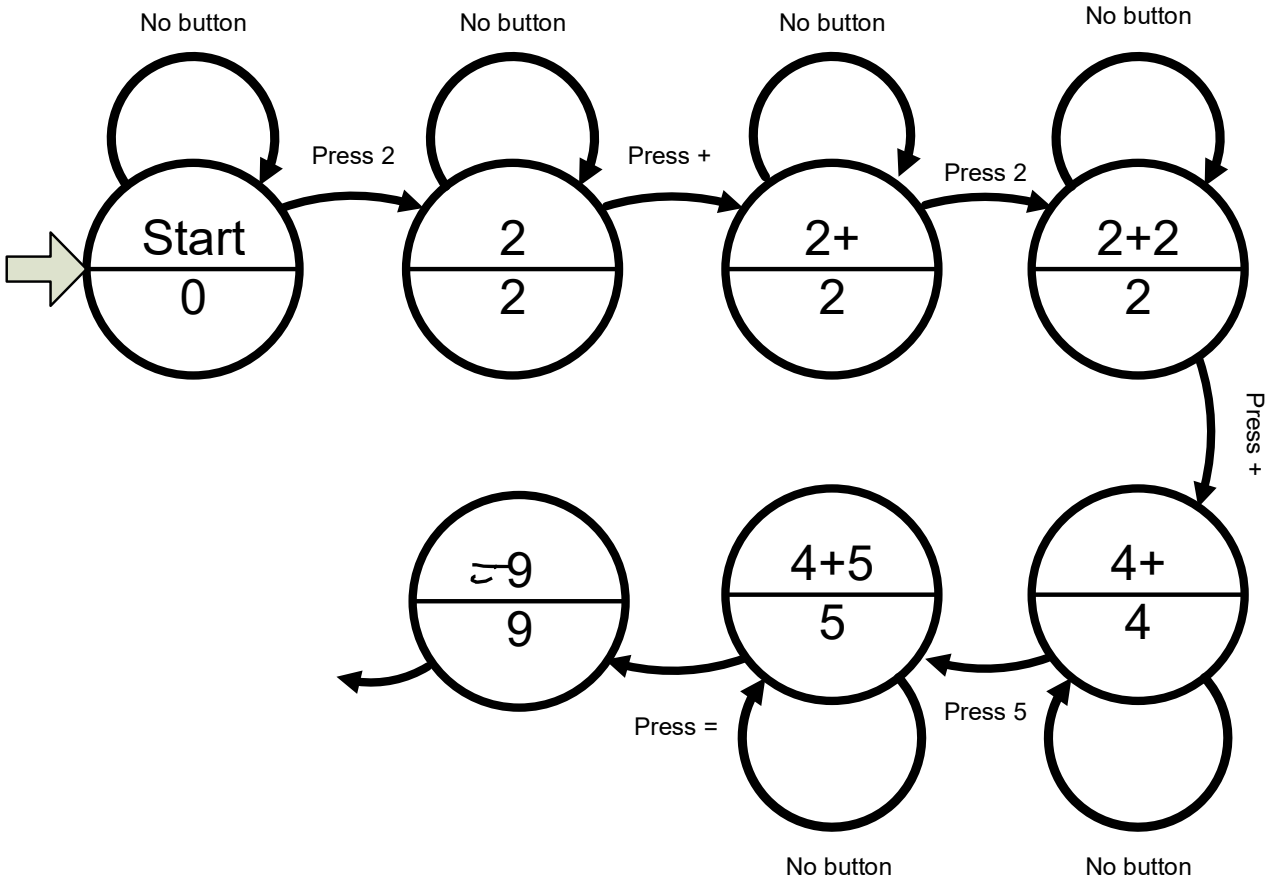
- System initialization
- The first operand
- The operator
- The second operand
- The result
- Which of them had been entered yet

Try one on your own:
Draw a state diagram for

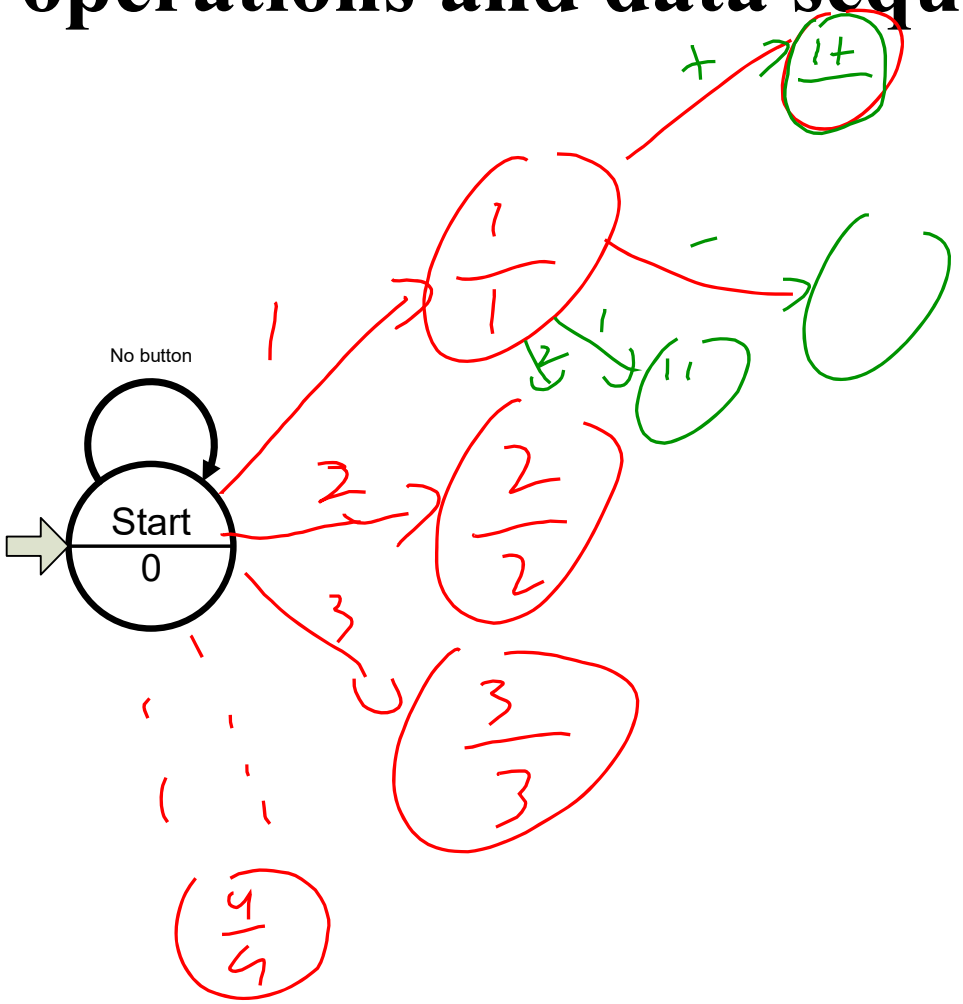
2+2+5 =



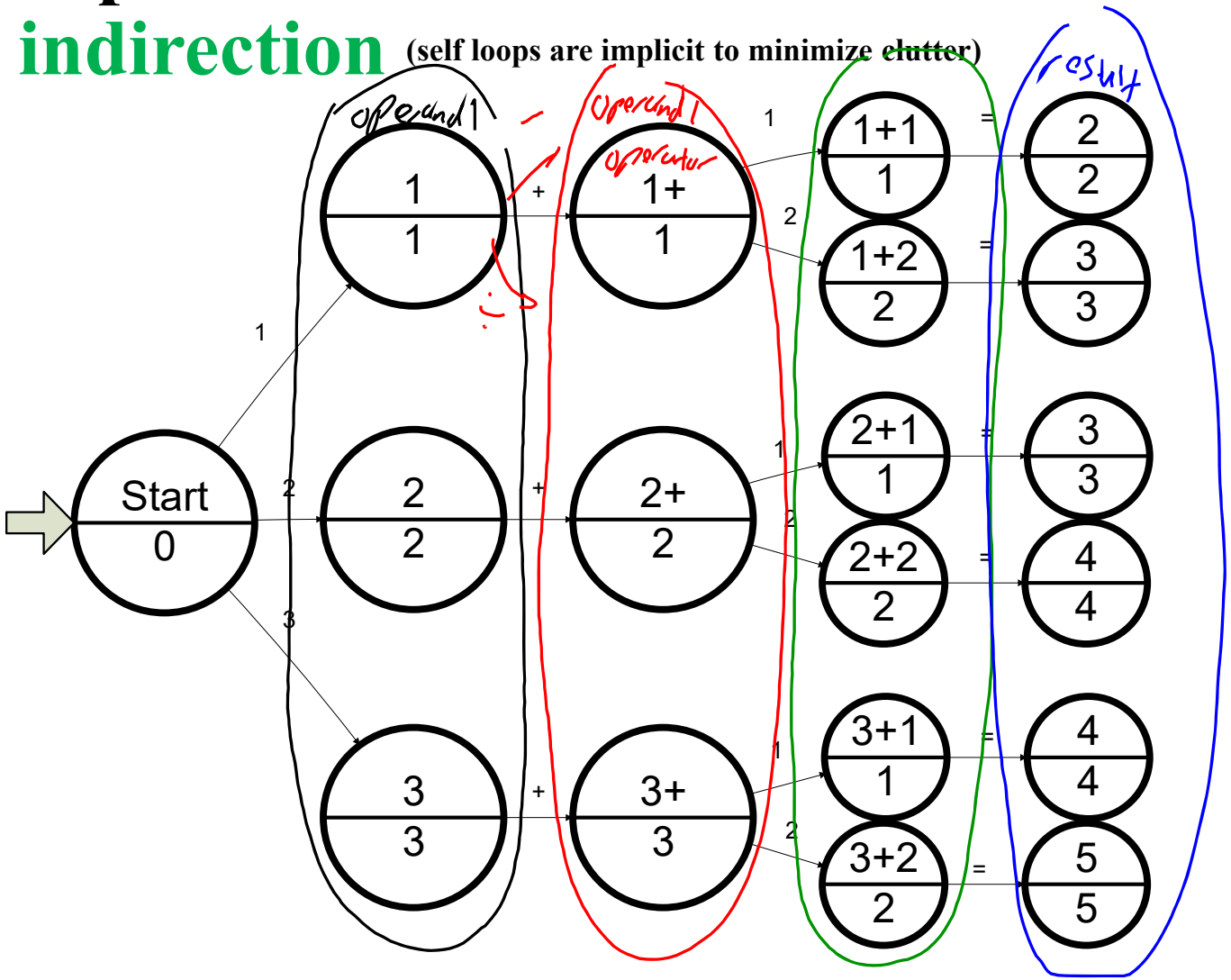
Try one on your own: Draw a state diagram for 2+2+5



Let's build a FSM that allows for several operations and data sequences



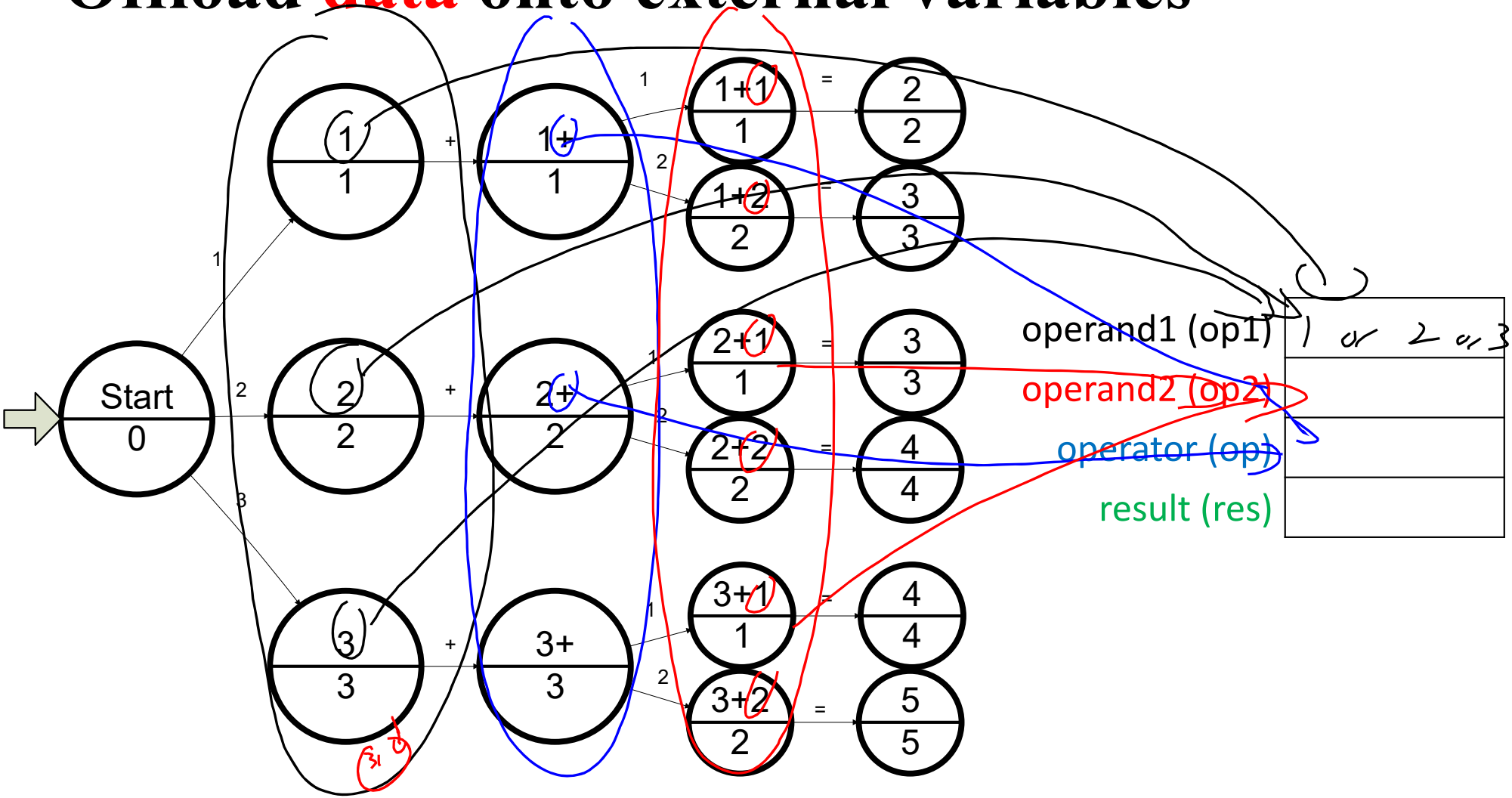
Separate the data from the control using indirection (self loops are implicit to minimize ~~clutter~~)



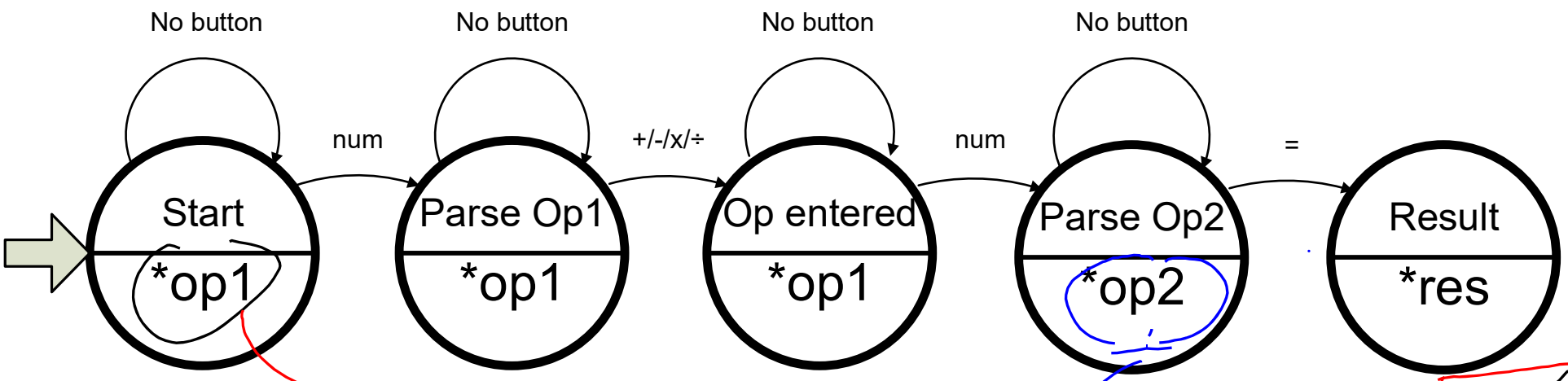
operand1, operand2, operator, and result are data

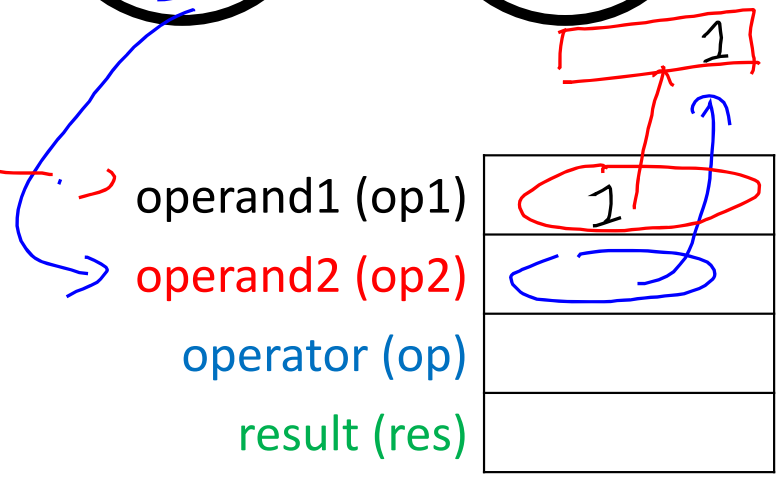
Start state and "which data has been entered" are control

Offload data onto external variables



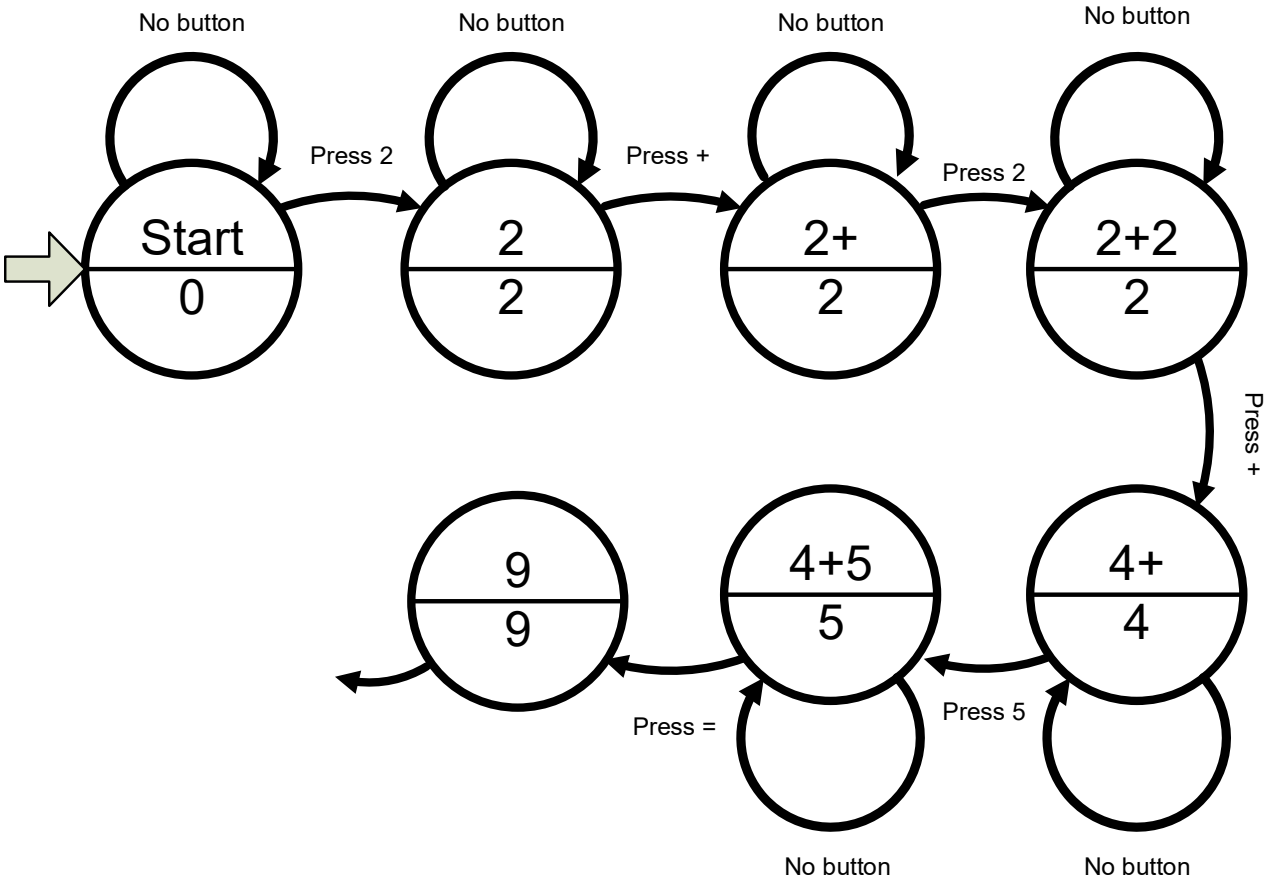
Our generalized calculator FSM so far



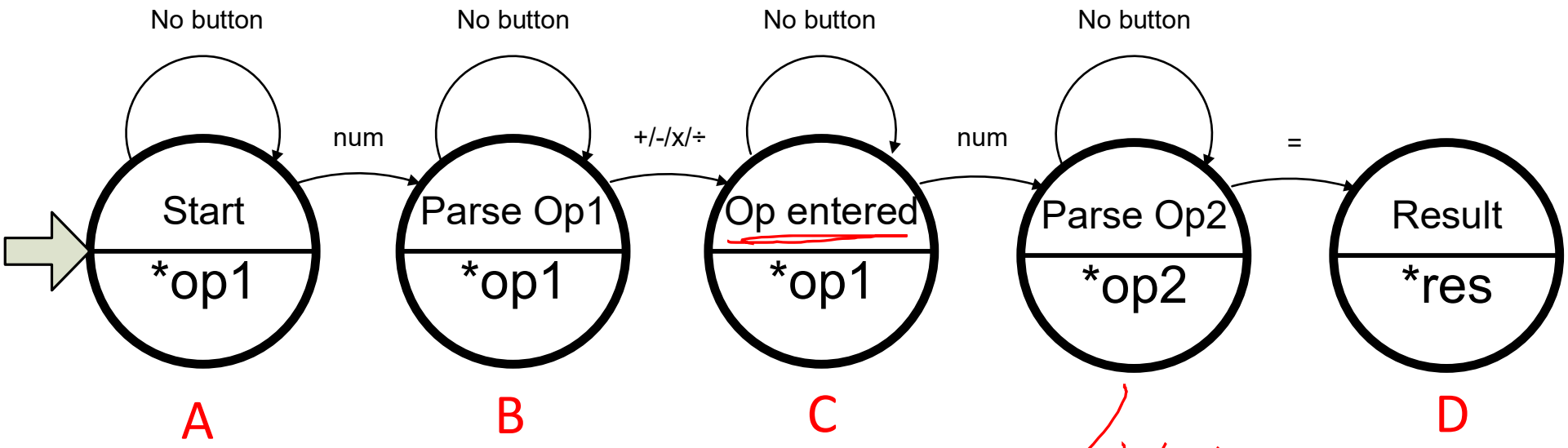


What else do we still need this state machine to do?

Note that the 2+ and 4+ states are functionally similar

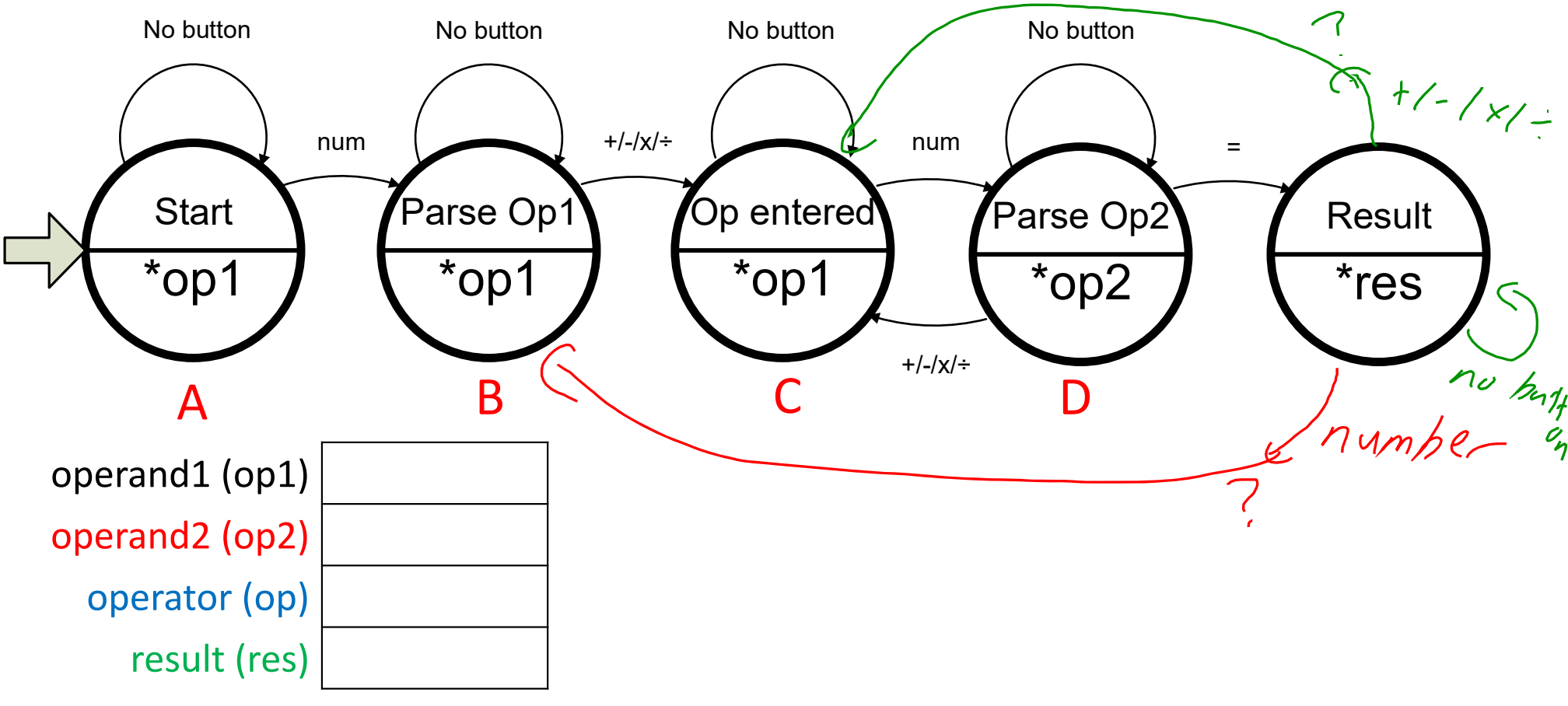


What is the next state when an operator is pressed during the *Parse Op2* state?

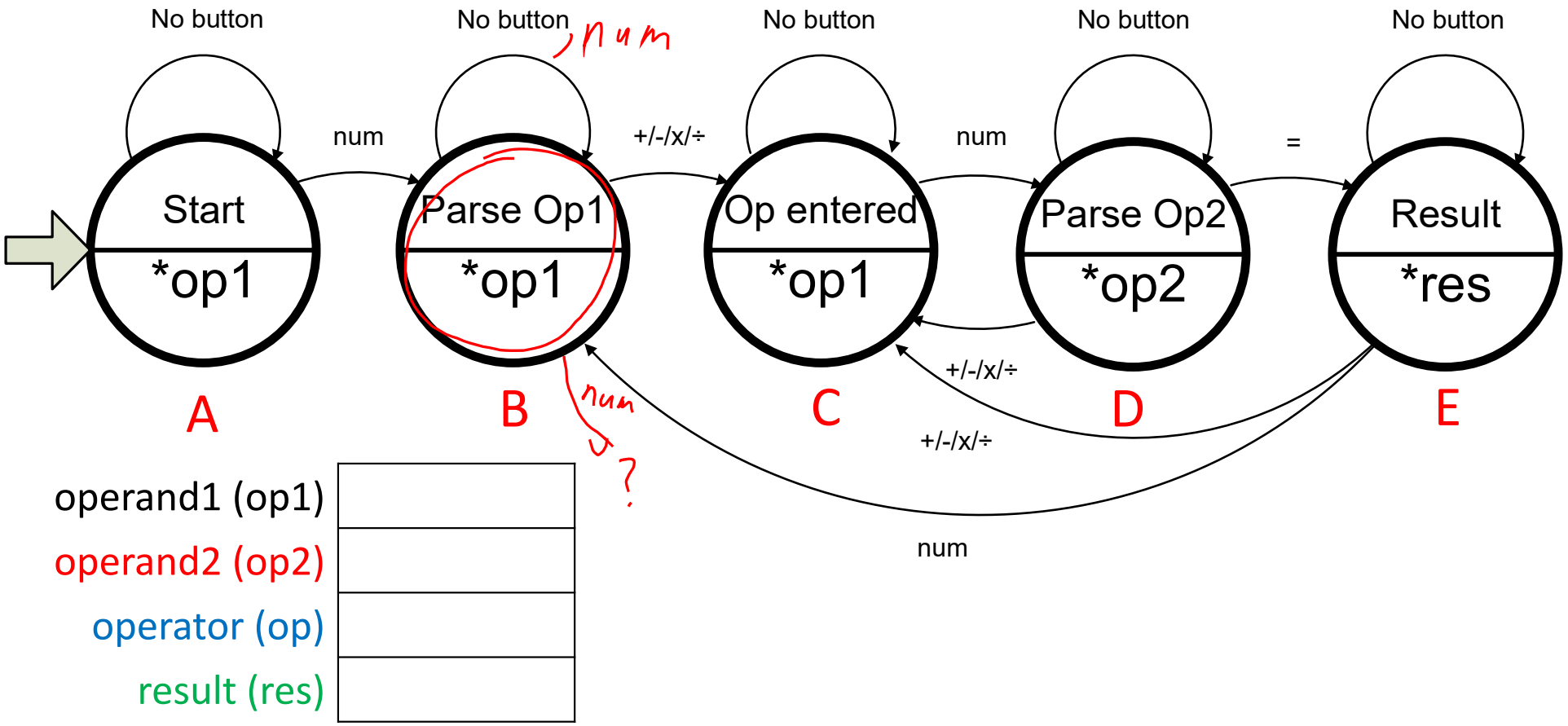


operand1 (op1)	
operand2 (op2)	
operator (op)	
result (res)	

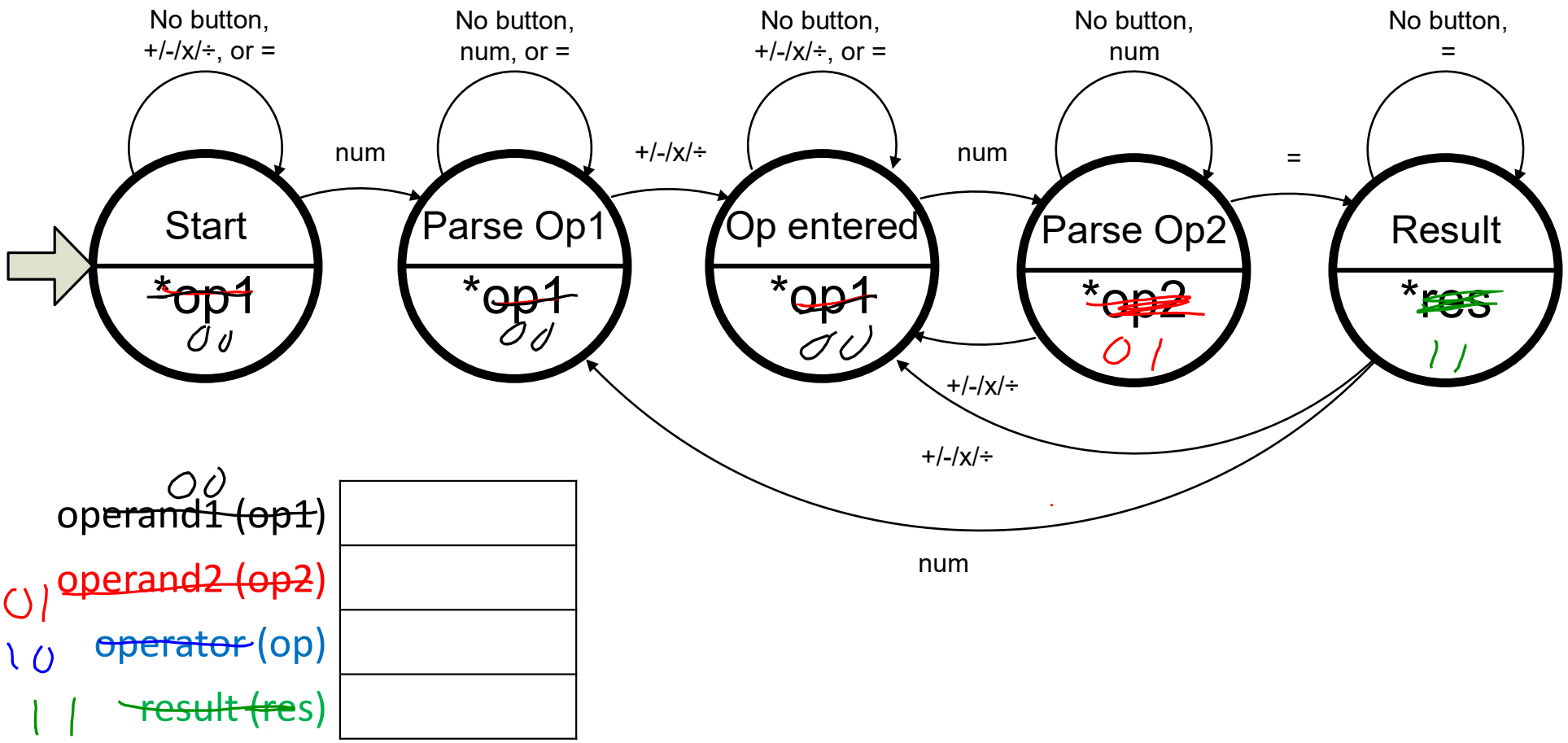
What happens after we have a result?



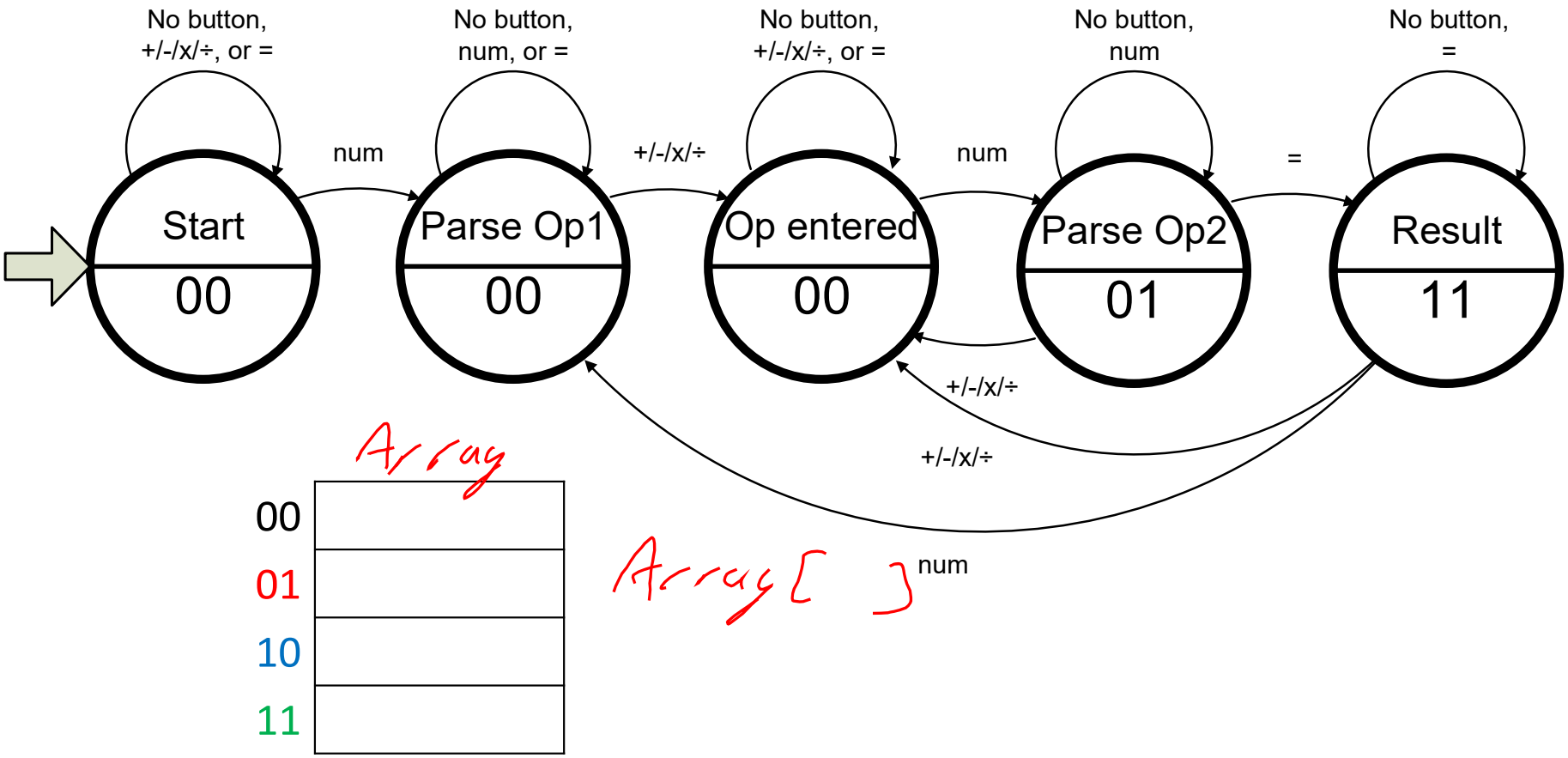
What happens if our operands have multiple digits?



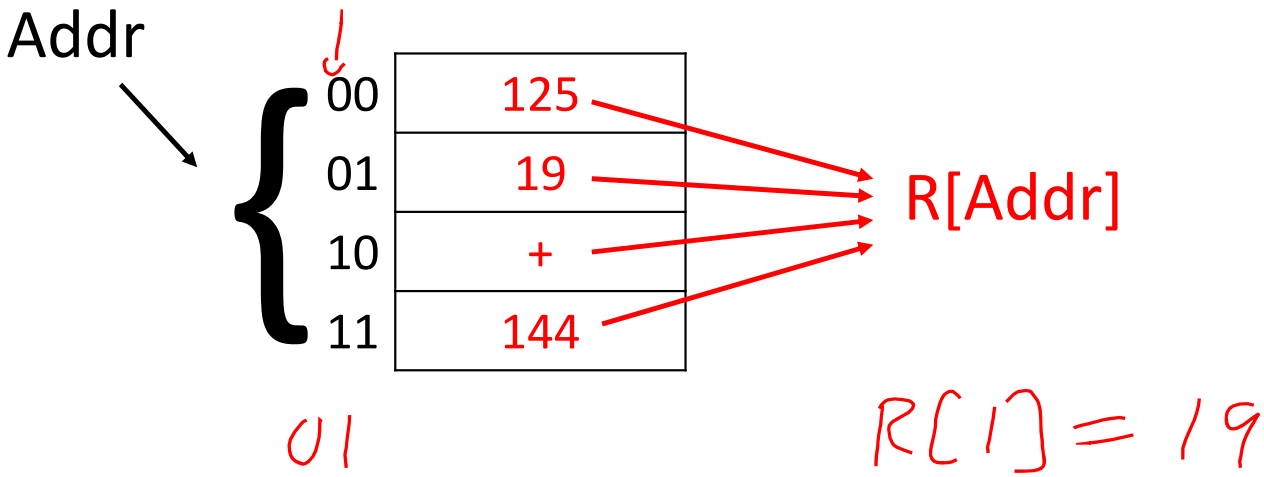
Our generalized calculator FSM



To implement indirection, treat the variables as indices in an array



Use brackets to indicate the data stored at an address in an array (a.k.a. a register file)



How many bits should be stored at each address?

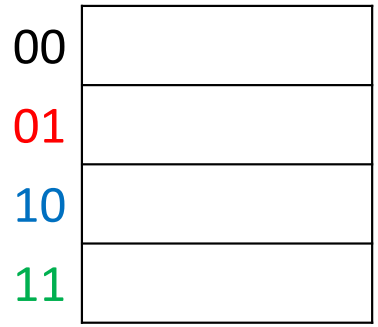
A) 8

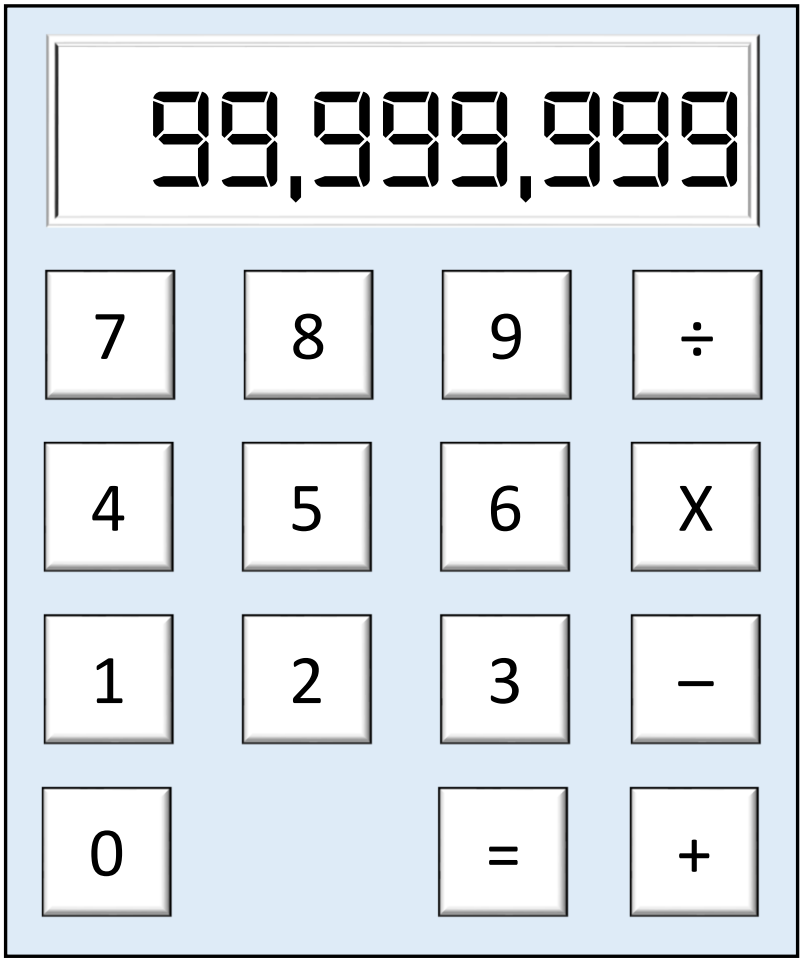
B) 27

C) 48

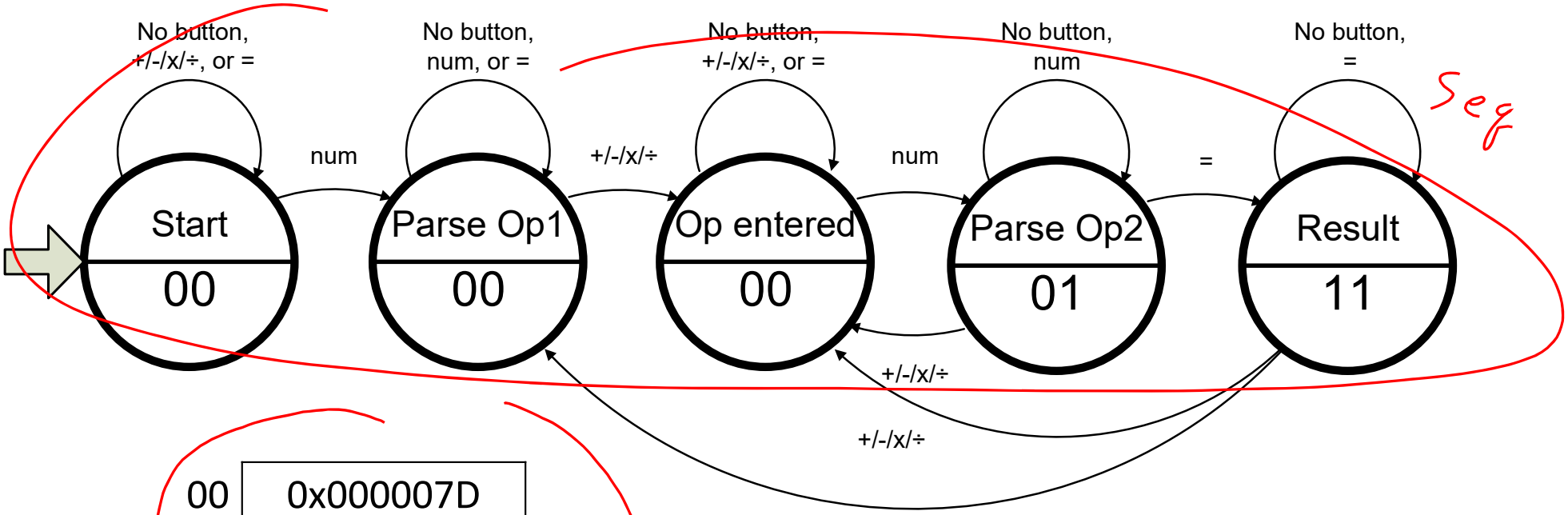
D) 108

E) 256





What components do I need in a circuit to implement this FSM?



00	0x000007D
01	0x0000013
10	0x0000000
11	0x0000090

register file

ALU

Seg

Use the FSM and system input to control the register file and ALU (the datapath)