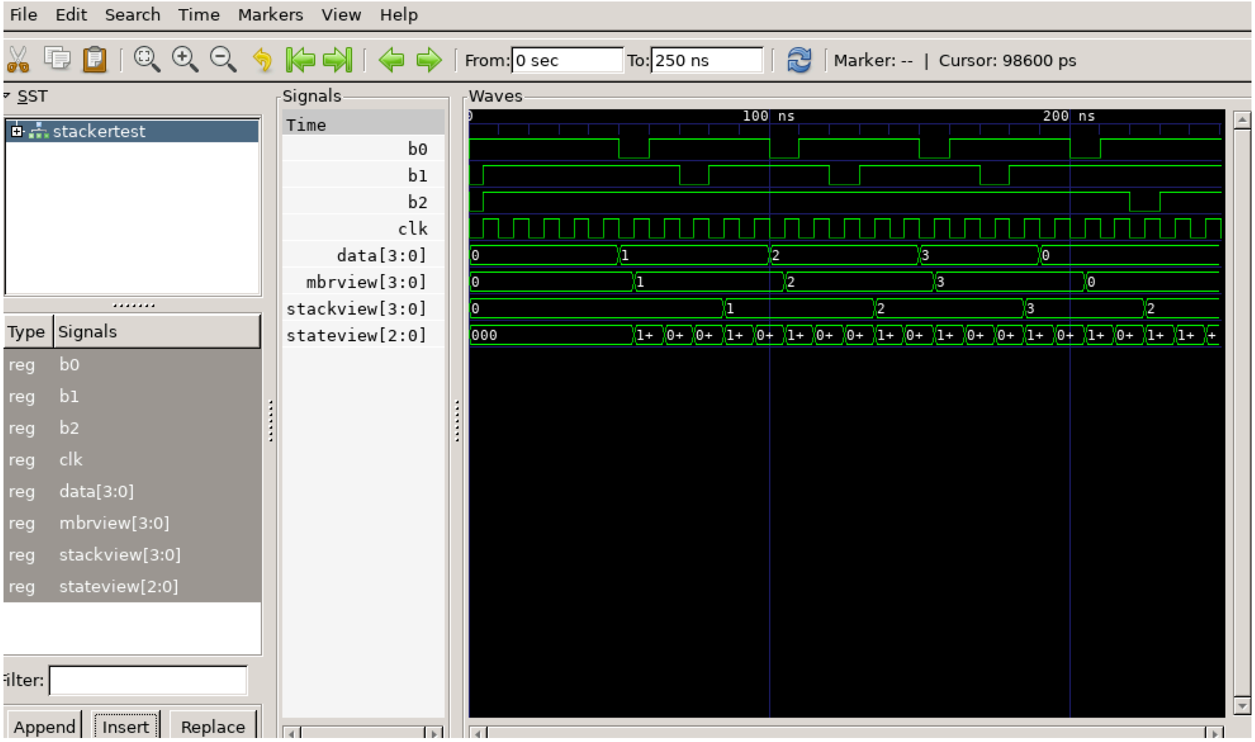
In this project, the purpose was to implement a stack-based four-function calculator using VHDL. This calculator uses a stack to store operands and execute arithmetic operations such as addition, subtraction, multiplication, and division. The stack allows for operations to be performed on the top two operands, with the result being placed back in the memory buffer register (MBR) and displayed on a seven-segment display.

In the lab, I was able to get the stacker to work which is shown through the Gtkwave output below.



From: 0 sec To: 250 ns

Marker: -- | Cursor: 98600 ps

Type	Signals
reg	b0
reg	b1
reg	b2
reg	clk
reg	data[3:0]
reg	mbrview[3:0]
reg	stackview[3:0]
reg	stateview[2:0]

The design of the stack-based calculator is made up of different components working together. One is a 16x8-bit RAM used to implement the stack for storing operands. Another is a state machine which controls the sequence of operations, including capturing values, pushing values to the stack, and executing arithmetic operations. Another are control inputs, where the calculator has three buttons (Capture, Enter, Action) and two operation switches (for selecting the arithmetic operation) that drive the state machine and perform calculations. Last is a seven segment display. There are two seven segment displays used to show the result stored in the MBR. The state machine allows the user to capture values from the data switches, push them onto the stack, and perform operations by selecting the corresponding operation switches. The result is then displayed on the seven-segment displays.

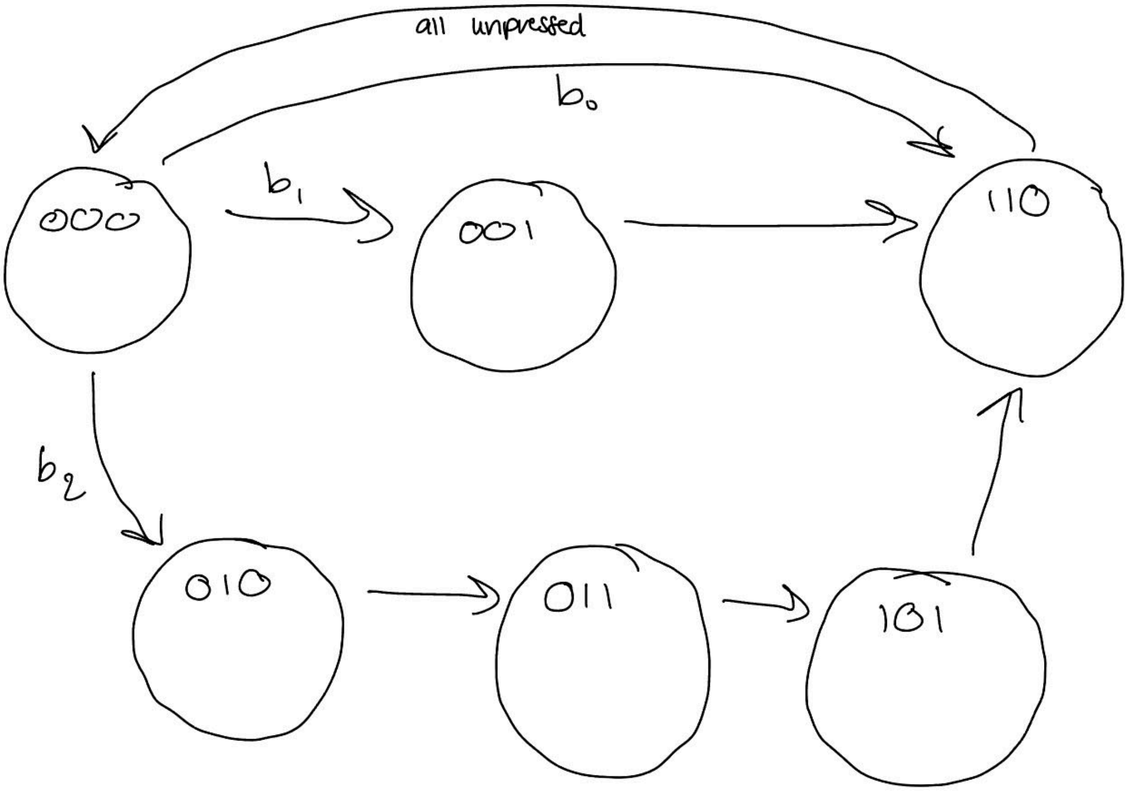


Image 1: State machine diagram

For testing, these are the tests I did:

Capture and enter test: The values were entered using the data switches and captured into the MBR. After each value was entered, it was pushed onto the stack using the buttons.

Arithmetic operations test: The calculator was tested for all four operations (addition, subtraction, multiplication, and division). The appropriate operation was selected using the operation switches, and the action button was pressed to perform the calculation.

Reset test: The reset state was tested by pressing two buttons together, which reset the calculator and cleared the stack.

Testing on board: The calculator was tested on a physical board, with the buttons and switches controlled manually to verify that all operations performed correctly.

I was unable to include pictures or videos of my testing because I didn't realize that it was necessary when I tested it and by the time I realized, I didn't have an opportunity to go back and take pictures or videos.

Acknowledgements: I got help from many online resources, as well as TAs, and other students in my lab.