Diagram, schematic

Description automatically generated

**The functionality of the stack are:**

* A stack that can store 10 integer values at maximum.
* The user will enter the size of the stack according to his own choice
* Whatever the size, the same amount of digits we store.
* The value we enter in the first will become the last value of the stack(the values are PUSH int the stack, moves forward).
* The last value we enter in the stack will become the top value in stack
* When the stack is filled or nor filled completely, if the user wants to pop the values from the stack, one by one value will remove from the stack
* Whatever is on the top will remove first(means the last value we enter in the stack) and whatever is in the bottom( the first value we enter in the stack) will get removed in the last.
* If the max size is complete and we enter the another, the bottom value of the stack will discard and the second last value will be become last.

**Diagram

Description automatically generated with low confidence**

The value we enter in the first will go in the last and the last value we put in stack will be on the top of the stack.

Working:

The first step is to enter the size of the stack

Calendar

Description automatically generated

Here we will enter the size of the stack, and for this we use HEXA keyboard. Whatever button is pressed form the keyboard. The size will select.

Here we select size 4

The maximum integers we store in our stack is 10, so we use 10 registers to store 10 integers values, we use shift registers so the when one value is passed it will push the first value to next register and previous register will store next value.

Diagram, schematic

Description automatically generated

And with the registers we use 10 MUX to perform push or pop work

Diagram, schematic

Description automatically generated

The MUX we use over there is (mux 2\*4 T.s). its bi directional way, there is one selection input (s0) and it is connected with the mode switch.

A picture containing diagram

Description automatically generatedselect 0 to push and 1 to pop

Its basically a selection input if its 0 the 0 pins lines become active and on the zero pin lines the input is connected which we pressed on the input( To input the values in the stack we use another HEXA keyboard and the output is connected with the input of MUX 0 pin lines).

And the output of the mux is connected with the input of its particular register.

And the particular register output is connected with the input of second Mux(d0 pin line)

That’s how we use 10 mux and 10 registers In parallel

Diagram, schematic

Description automatically generated

To check how many registers will active (whatever the size the stack, exactly the same amount of registers will be active) for the we make a condition with decoder and the two OR gates.

The size will select and the output is attached with the decoder input

The decoder we use is 4\*10 decoder

Diagram, schematic

Description automatically generated

And every decoder output line(i) is attached with the every register(i) one by one.

Before going to register two or gates are connected.

* The first one is connected with the button.
* The second one is connected with the clock

The values we need to enter in the stack.

And the output of the input keyboard is connected with the first MUX so whatever we enter on the keyboard, the value will go to the first MUX( 0 pin line)

The button (SPTD push button) is attached.

When one time we pressed the button Chart, box and whisker chart

Description automatically generated

The first register becomes active as the register clock is connected with the decoder output and passes through two or gates.

* The first or gate contains one decoder value and one button value.
* The second or gate contains one the previous or gate value and the clock.

If the conditions are fulfilled the clock will be active and register also started storing and shifting values.

The size will be controlled from the decoder, whenever the max size will reach, the decoder value becomes one and button is also not pressed to store next value, it will become zero in the first or gate and the register connected to that particular gate will not become active so we cannot store futher values.

In that particular case we select the size as 4, the 4 registers are active and the 5 one is disabled.

To check the values in every register, we make HEXA DISPLAY attached with every output of the register.

A picture containing text, crossword puzzle

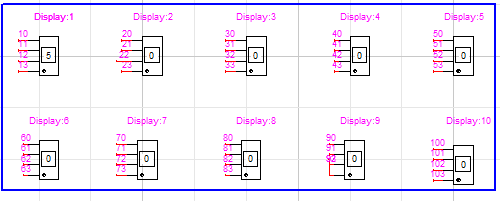
Description automatically generated

“Every (i)display is attached with the every output (i)register”

Lets check an example:

The size of a stack is 4, first we take mode to 0 (to push values) and the select the value from the input keyboard

* First we select 5 and press the button, the first register will become active and value 5 is store in the first register as we can see on the display.



**The first register store the value of 5**

* Afterwards we press another value 2, and pressed button. The first value(5) will shift to the second register and the first register contains value of 2,

A picture containing text, crossword puzzle

Description automatically generated

**The first register contains 2 value**

* Afterwards we press another value 7 and pressed button. The first value(5) will shift to the third register and the first register contains value of 7

Diagram, schematic

Description automatically generated

**The first register contains recent value we entered**

* Afterwards we press another value 6 and pressed button. The first value(5) will shift to the fourth register and the first register contains value of 6

A picture containing crossword puzzle

Description automatically generated

**The first register contains recent value**

* The size of stack is 4, 4 registers are filled if we put another value, the last value in the stack( first value we entered) will be discard as the 5 register is not active, and the new value we entered afterwards will be stored on the top of the stack.

Diagram, schematic

Description automatically generated

**The 5th value will be stored on the top**

All the four registers contains all the values four values, stack if filled

For POP, we put mode on 1, after pressing button the value will be popped out.

In popping, The values stored in the first MUX will be shifted to the (selection pin 1 lines) and the values are discard line by line.

As on the first pressed button:

Diagram, schematic

Description automatically generated

**The first value is discard**

On the second pressed button:

**Diagram, schematic

Description automatically generated**

**The second value is discard**

On the third pressed:

A picture containing room, colorful

Description automatically generated

**The third value is discard**

And on the fourth pressed:

Diagram, schematic

Description automatically generated

**The fourth value is discard**

**As we can that all the values are popped out from the stack**

**LED**

* If the stack is filled completely or partially, the LED light will glow.
* If the stack is empty, the LED light will not glow.

A picture containing text, crossword puzzle

Description automatically generated

**Here the stack is empty the light is off**

Stack is filled the light is on:

**Diagram, schematic

Description automatically generated**

* If the values are filled and we pressed the reset button the whole stack becomes empty.
* And to check if the stack is empty or not we make a circuit which is attached with the first register output

Chart, box and whisker chart

Description automatically generated

If the register contains 0 value in every bit, it will become one and its empty output is attached with the Led light with not it will become 0 and led light will show off.