## **State Diagram**

Our circuit consists of 8 states as instructed by the project guidelines. We will use 3 D flip flops to represent those states as they are sufficient to represent them since

At any given point in time we have access to 3 sets of valuable inputs:

The regular switch inputs:

The previous switch states (covered in separate module):

The timer variable: T (4 second timer)

We denote the states as follows:

**Boot State (000):** The state the machine always wakes up in, it is the deciding states which navigates the machine into either the regular sequencing or the locked state.

**Sequence One (001):** In this state we follow the LED activation sequence 1->2->3->4. If we draw the mapping diagram to illustrate the relationship between the switches and the LEDs, we get the following:

A black lines with dots and numbers

AI-generated content may be incorrect.

Figure - Sequence One Mapping

This figure illustrates which switch controls which LED. We will effectively use this mapping concept and diagram to explain the rest of the sequences as well.

Sequence one is reached by default through the boot state as long as the switches did not have the configuration (0100). We can reenter sequence one from the sequence detector if the last switch that was turned off was switch one. We can check this by reading the values of the previous switch states as (1000).

**Sequence Two (010):** In this state we follow the LED activation sequence 2->3->4->1. If we draw the mapping diagram to illustrate the relationship between the switches and the LEDs, we get the following:

A diagram of lines and points

AI-generated content may be incorrect.

Figure - Sequence Two Mapping

We enter sequence two from the sequence detector if the last switch that was turned off was switch two. We can check this by reading the values of the previous switch states as (0100).

**Sequence Three (011):** In this state we follow the LED activation sequence 3->4->1->2 (Ignoring the Special Trick for now). If we draw the mapping diagram to illustrate the relationship between the switches and the LEDs, we get the following:

A diagram of a cross with lines and dots

AI-generated content may be incorrect.

Figure - Sequence Three Mapping

We enter sequence three from the sequence detector if the last switch that was turned off was switch three. We can check this by reading the values of the previous switch states as (0010). We can also reenter this sequence from the special trick sequence if less than two LEDs are on.

**Sequence Four (100):** In this state we follow the LED activation sequence 3->4->1->2 (Ignoring the Special Trick for now). If we draw the mapping diagram to illustrate the relationship between the switches and the LEDs, we get the following:

A diagram of lines and dots

AI-generated content may be incorrect.

Figure - Sequence Four Mapping

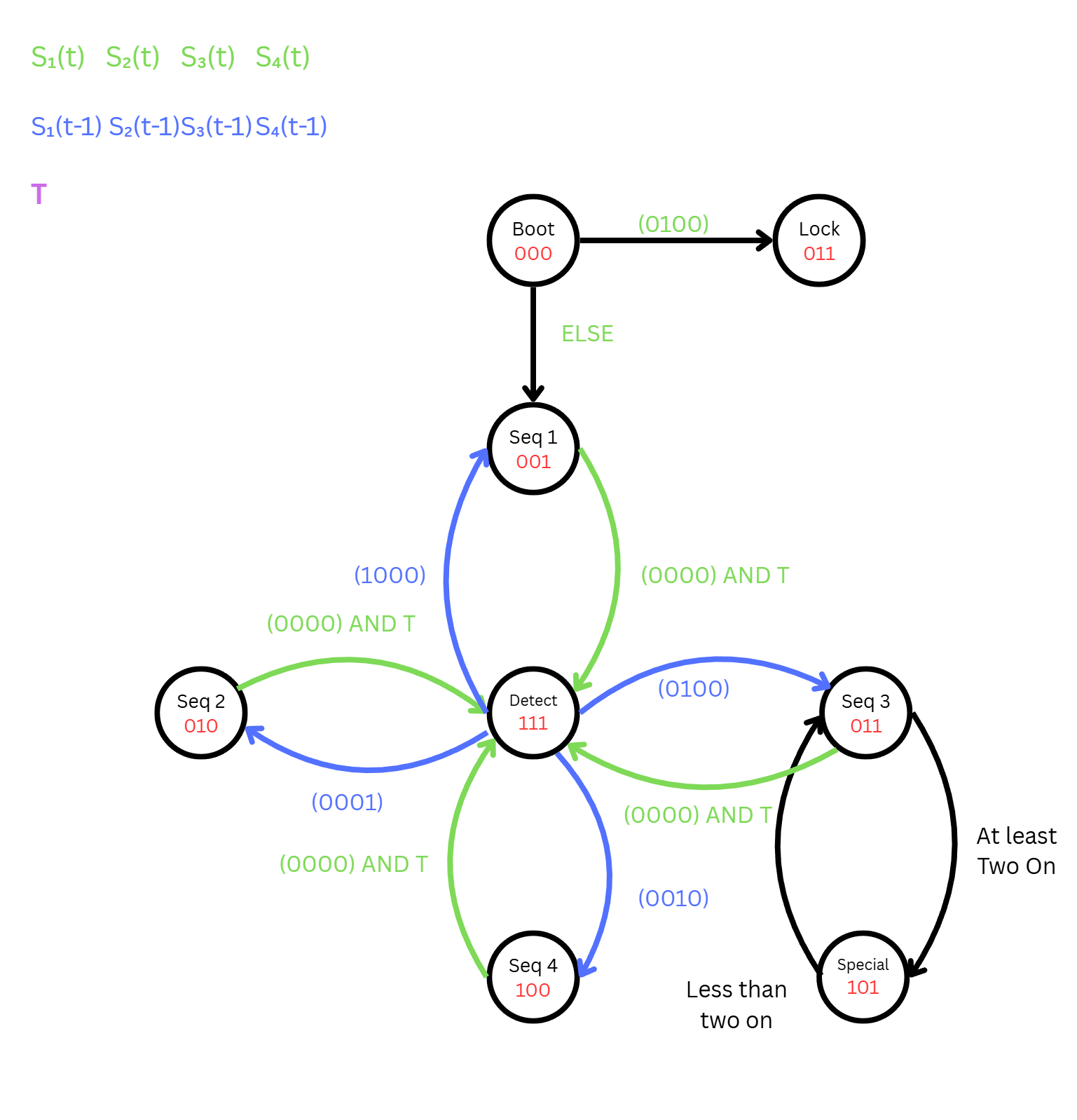
We enter sequence four from the sequence detector if the last switch that was turned off was switch four. We can check this by reading the values of the previous switch states as (0100).

**Special Trick (101):** This state is reached from sequence three on the conditions that at least two switches are enabled. This state is supposed to follow the same mapping as sequence three but with the caveat of a disabled switch. We will into this state in detail in its own section.

**Locked State (011):** This state has the exact same mapping as that of sequence one. However, this state is reached from the boot state specifically to trap the machine in it. It is effectively reached when the system boots with the switch sequence (0100). Once we are in the locked state we cannot escape it unless we reboot the machine.

**Sequence Detector (111):** This state is a decision state. It is responsible to evaluate the last switch turned off and decide the next sequence to put the machine into. It has no LED mapping, and it is not intended to control them. The aim is to be in this state for a very short and brief period of time, ideally a clock cycle. We can reach this state from any of the four-sequence states through the condition that all four switches are off (0000) and a time of 4 seconds has elapsed.

The state diagram is found below. We will not tackle the flip flop input formulas in this example but we will explain how the effective mapping will occur.

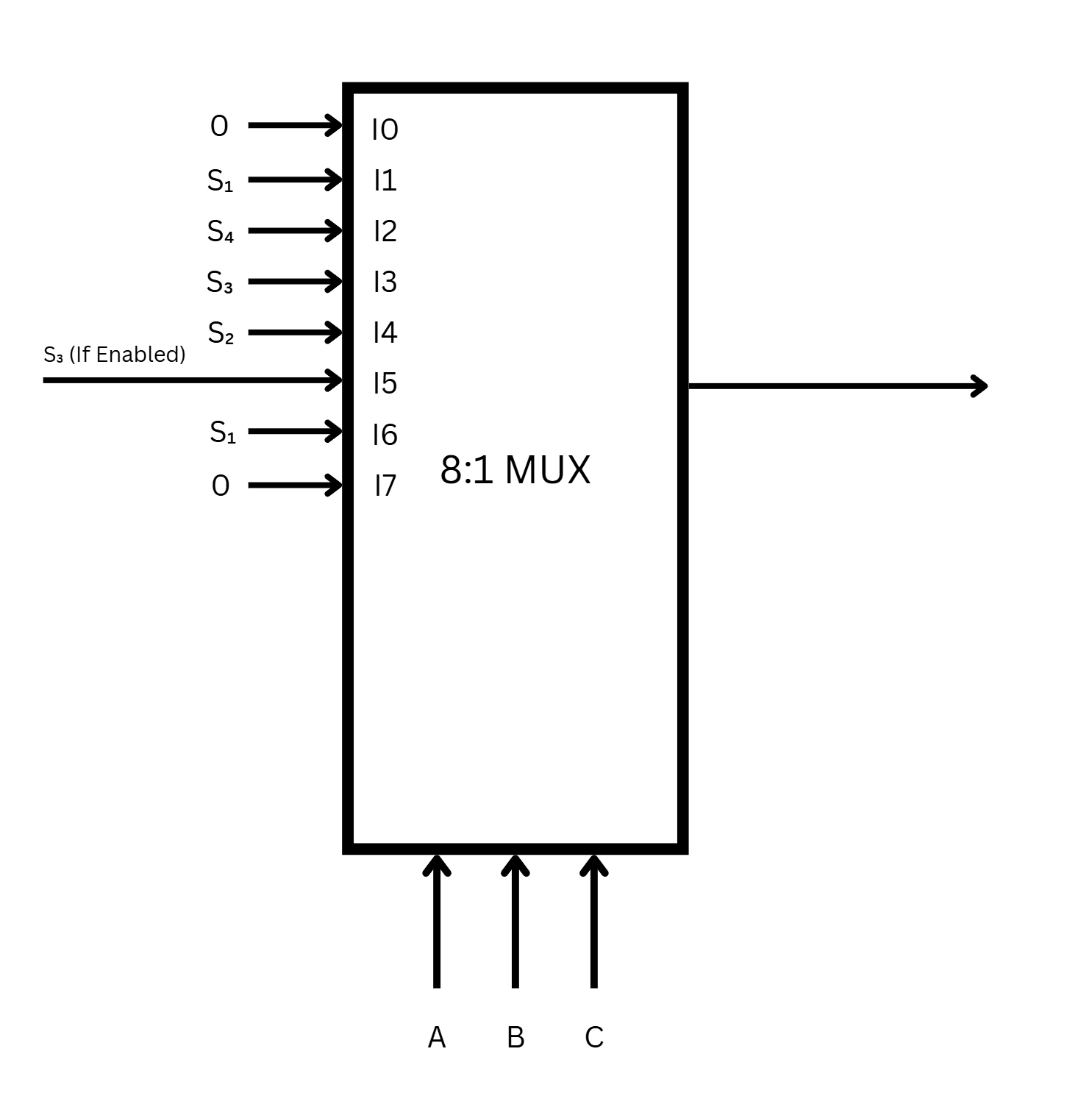


## **Output Multiplexing**

A very important issue that needs to be addressed is how we will effectively translate the changes in mapping from one state to another. Each state controls the output in a different way, and we need to figure out a way to multiplex them.

To each LED we will connect an 8:1 MUX, which will take the state bits (ABC) as select lines to determine which input to follow.

For example, here is the 8:1 MUX associated with LED 1.



here is the 8:1 MUX associated with LED 2.  
A diagram of a rectangular object with arrows pointing to the side

AI-generated content may be incorrect.

ADD TABLE

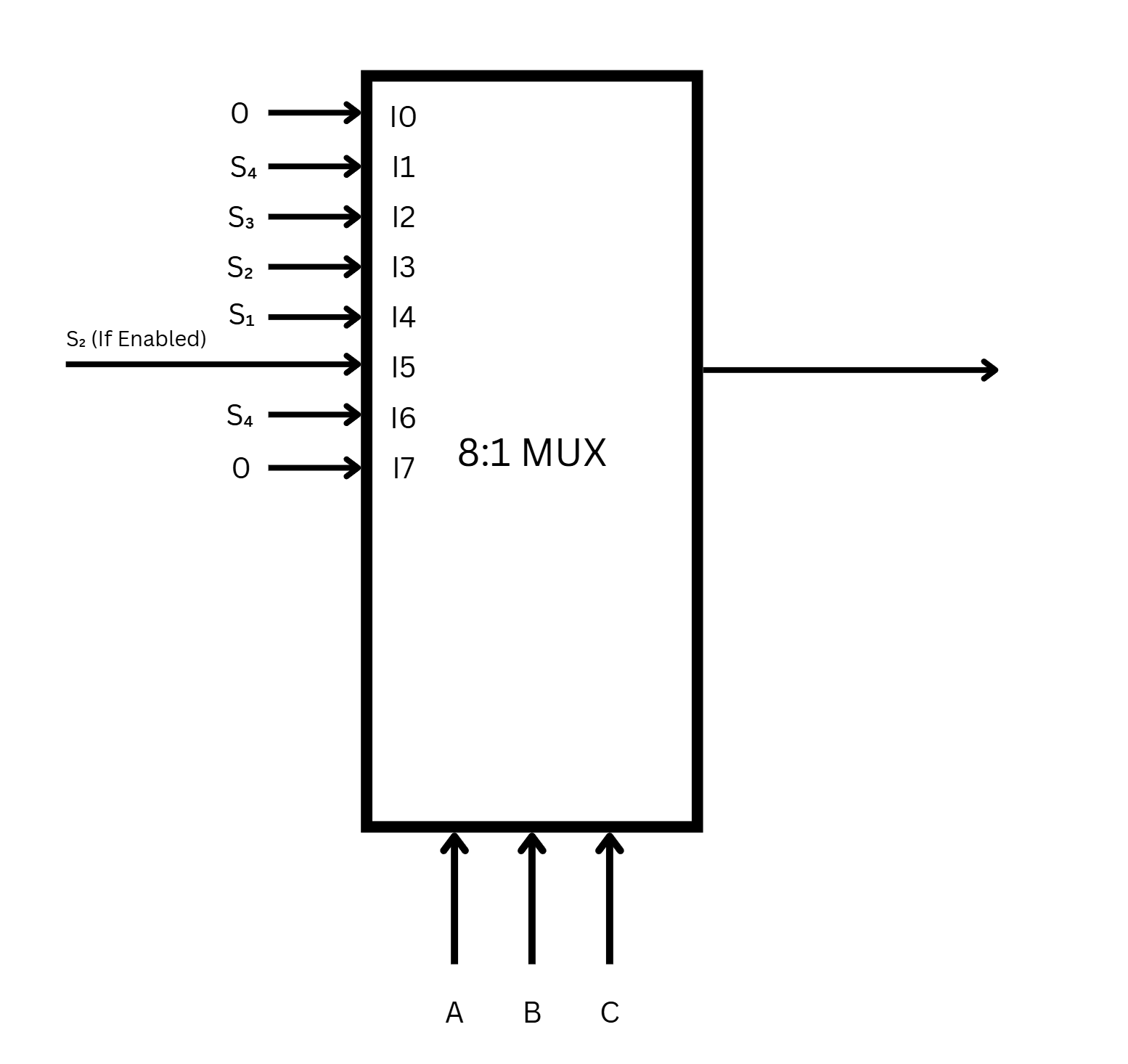
here is the 8:1 MUX associated with LED 3.

A diagram of a rectangular object with arrows

AI-generated content may be incorrect.

ADD TABLE

here is the 8:1 MUX associated with LED 4



ADD TABLE.

## **Constructing the 8:1 MUX**

Since we do not have an 8:1 MUX readily available we must construct it using 4:1 MUXs as the IC 74LS153 is the only multiplexing chip that we had available.

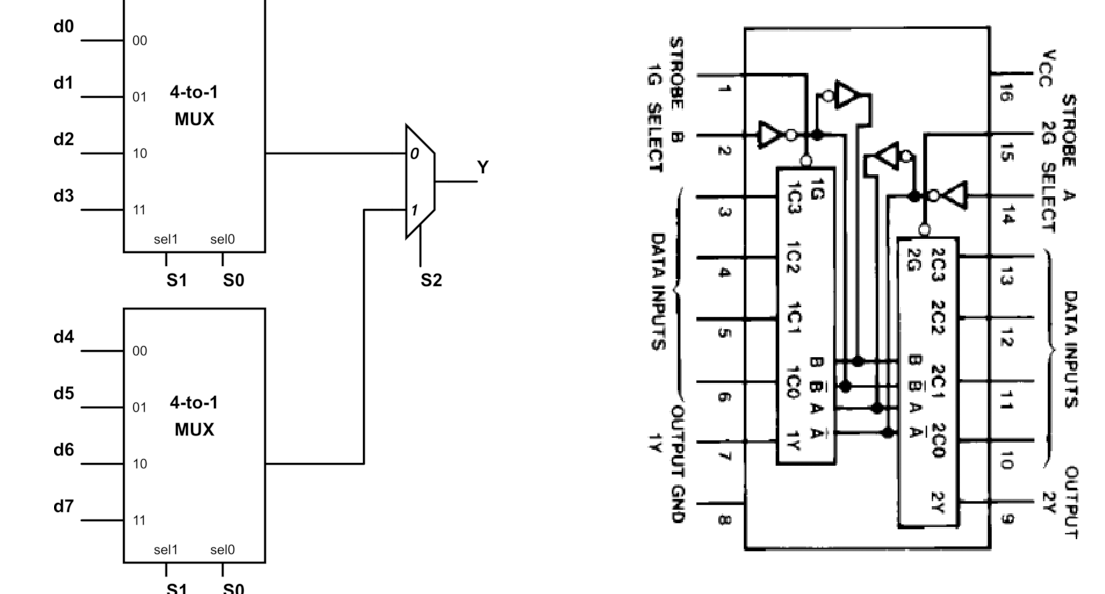


Figure - 8:1 MUX Schematic With 74LS153

Using the above schematic and looking into the datasheet of the 74LS153, we can effectively construct an 8:1 MUX using two 74LS153 chips. One chip is used to accommodate the first two 4:1 MUXs in the diagram (it helps greatly that both MUXs share the same select lines). And we can use one more chip with two adjacent inputs connected to use it as a 2:1 MUX.

Below is the circuit that is effectively the 8:1 MUX, where the first input is the right most bottom switch.

