Date: **01-08/04/2026** **CS29204**: Switching Circuits Laboratory

Assignment 9

Design of complete systems

Group 36
Dipam Sen (24CS10059)
Mitali Laddha (24CS10111)

Part (a)

Aim

To design a circuit to generate successive 4-bit Fibonacci numbers.

Apparatus

- Breadboard
- IC 74LS173 × 2 (4-bit register)
- IC 74LS157 × 2 (Quad 2x1 MUX)
- IC 74LS83 × 1 (4-bit adder)
- IC 74LS73 × 1 (Dual JK flip-flop)
- IC 74LS04 × 1 (Hex inverter)
- IC 74LS08 × 1 (Quad 2-input AND gate)
- IC 74LS32 × 1 (Quad 2-input OR gate)
- Wires, wire cutter, tweezers

Design

First, we design a high level flow diagram of the circuit.

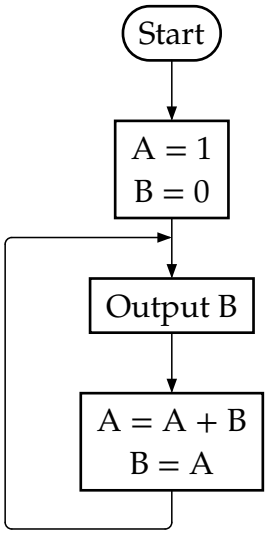


Figure 1: Flow diagram of the circuit

As described above, the circuit will output successive Fibonacci numbers, starting with 0, 1, 1, 2, 3, 5, 8, 13, and then will overflow.

We can design the corresponding data path circuit. We will use two 4-bit registers to store A and B, and a 4-bit adder to compute $A + B$. We will also need some multiplexers to select the inputs to the registers, and some control signals to control the flow of data.

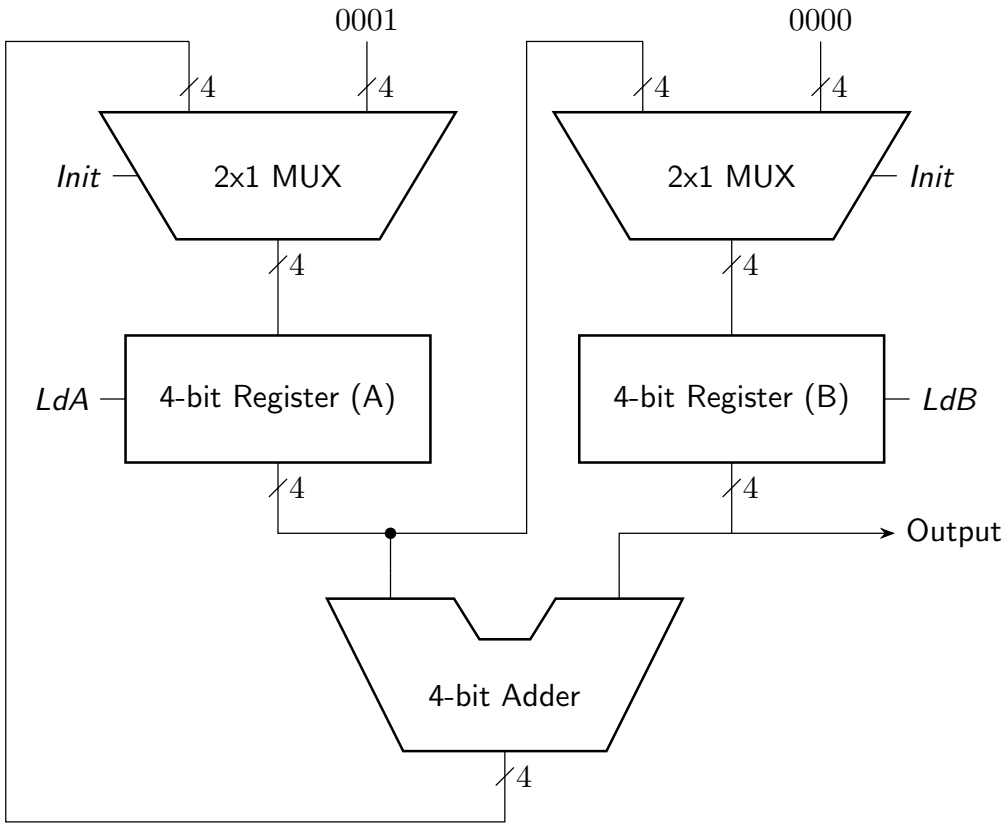


Figure 2: Data path for the circuit

Our data path takes in three control signals, namely *Init*, *LdA* and *LdB*. The *Init* signal is used to initialize the registers. The *LdA* and *LdB* signals are used to load the values of A and B into the registers, respectively. When *LdA* is applied (while *Init* is false), the output of the adder will be loaded into A. When *LdB* is applied (while *Init* is false), the value of A will be loaded into B.

We need a corresponding sequential circuit (the control path) which will generate the required control signals to control the flow of data in the data path. From our flowchart, we can design the control path as follows.

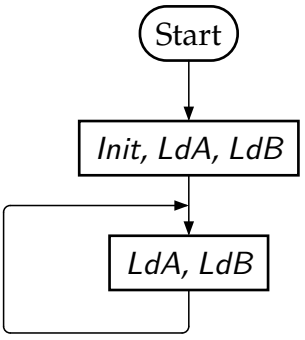


Figure 3: Control Path for the circuit

We can implement the control path as a finite state machine. It has a single input Start, and it generates 3 outputs: *Init*, *LdA* and *LdB*. The state transition diagram of the control path is shown below.

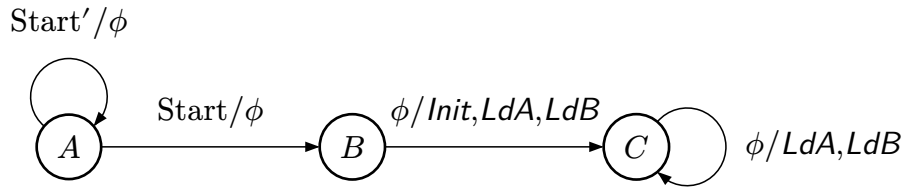


Figure 4: State transition diagram of the control path

Correspondingly, we can draw the state table for the FSM.

PS	NS, Z	
	Start	Start′
A	$A, 000$	$B, 000$
B	$C, 111$	$C, 111$
C	$C, 011$	$C, 011$

Table 1: State table for FSM. Here, $Z =$ (*Init*, *LdA*, *LdB*)

Next, we need to assign values for the different states. Since we have 3 states, we need at least 2 bits to represent the state. We can use the following state assignment:

$$A = 00$$
$$B = 01$$
$$C = 10$$

Thus, we can rewrite the state table using the above state assignment (We also let the invalid state 11 to transition to the 00 state in the next clock cycle):

PS		NS		*Init*		*LdA* = *LdB*	
y_1	y_2	$S=0$	$S=1$	$S=0$	$S=1$	$S=0$	$S=1$
0	0	00	01	0	0	0	0
0	1	10	10	1	1	1	1
1	0	10	10	0	0	1	1
1	1	00	00	0	0	0	0

Table 2: State table for FSM after state assignment

Now, we need to choose the type of flip flop to use. In this case, we are given J-K flip-flops. So we can write down the excitation table for a J-K flip-flop:

	J	K
$0 \to 0$	0	X
$0 \to 1$	1	X
$1 \to 0$	X	1
$1 \to 1$	X	0

Table 3: Excitation table for J-K flip-flop

We need to find the combinatorial circuit which will produce the values of the flip-flop inputs and the circuit outputs.

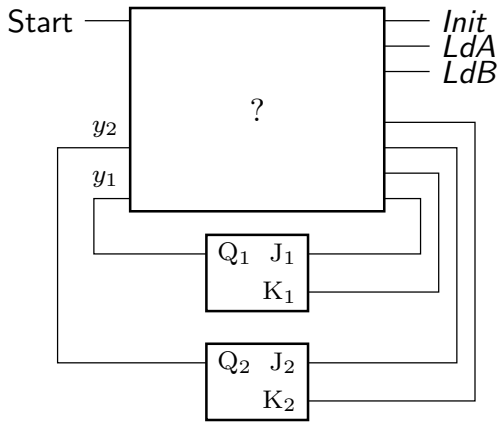


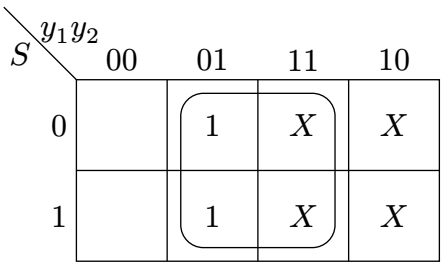
Figure 5: Schematic diagram for the FSM

Using the excitations for the J-K flip-flops, we can write the final excitation and output table for the FSM.

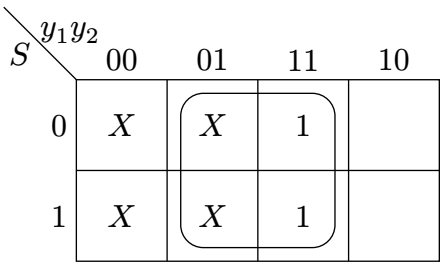
		$S=0$				$S=1$				*Init*		*LdA* = *LdB*	
y_1	y_2	J_1	K_1	J_2	K_2	J_1	K_1	J_2	K_2	$S=0$	$S=1$	$S=0$	$S=1$
0	0	0	X	0	X	0	X	1	X	0	0	0	0
0	1	1	X	X	1	1	X	X	1	1	1	1	1
1	0	X	0	0	X	X	0	0	X	0	0	1	1
1	1	X	1	X	1	X	1	X	1	0	0	0	0

Table 4: Final excitation and output table for FSM

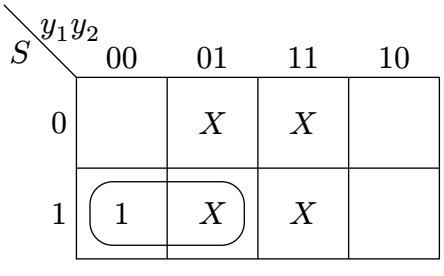
To obtain the combinational circuit which generates the J/K inputs and the control signals, we can draw Karnaugh maps for the outputs and obtain their minimal switching expressions.



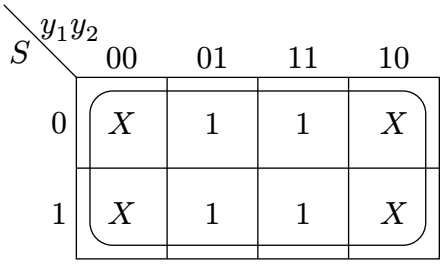
$J_1 = y_2$



$K_1 = y_2$



$J_2 = y_1'S$



$K_2 = 1$

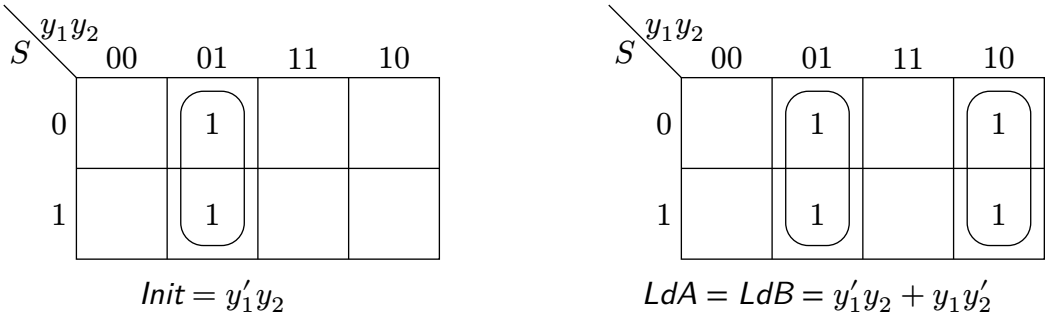


Figure 6: Karnaugh maps for all the output signals of the FSM

We can implement the combinational circuit for the FSM using AND, OR and NOT gates.

Finally, we connect the control signals generated by the FSM to the Data path as required, and the output of the data path to the output lines.

Circuit Diagrams

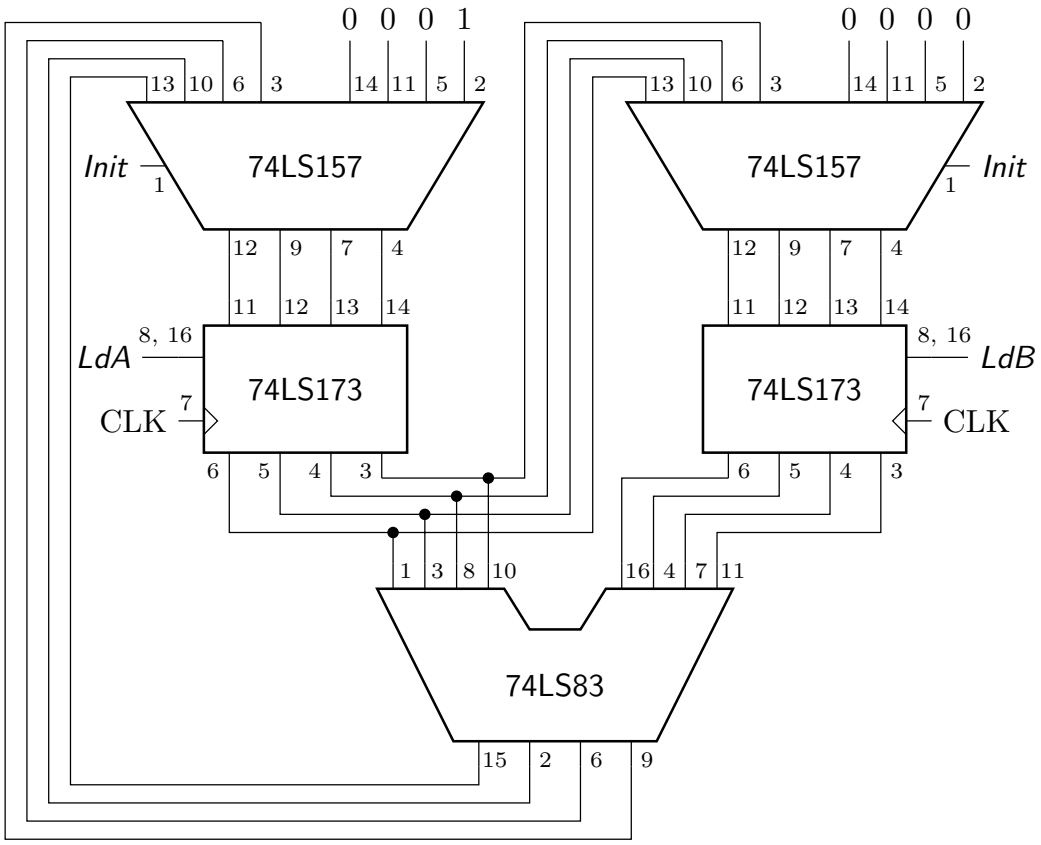


Figure 7: Circuit diagram for the data path of the circuit

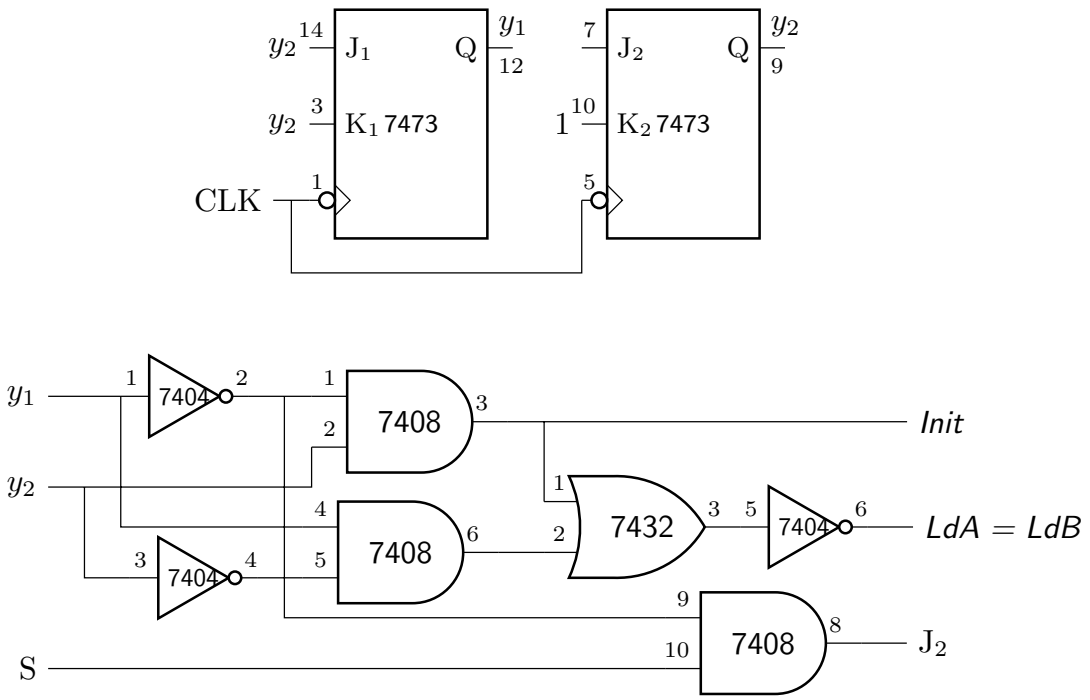


Figure 8: Circuit diagram for the control path of the circuit

Observations

When initialised to the start state, and when the Start signal is triggered, the circuit outputs consecutive Fibonacci numbers on each clock pulse: 0, 1, 1, 2, 3, 5, 8, 13, after which it overflows.

Part (b)

Aim

To design a circuit that can display four different digital waveforms on a single oscilloscope channel.

Apparatus

- Breadboard
- Digital Storage Oscilloscope
- 74LS153 × 1 (Dual 4x1 MUX)
- 74LS93 × 2 (4-bit Counter)
- LM741 × 2 (Operational Amplifier)
- LM555 × 1 (Timer IC)
- Resistors, Capacitors
- Wires, wire cutter, tweezers

Design

Let us call the four input waveforms as *A*, *B*, *C* and *D*. We need to select one of these waveforms to be displayed on the oscilloscope channel (at a time). Let us say, we have two select signals S_1 and S_0 which selects the waveform to be displayed. Then, we can use a 4x1 multiplexer to select the input waveforms.

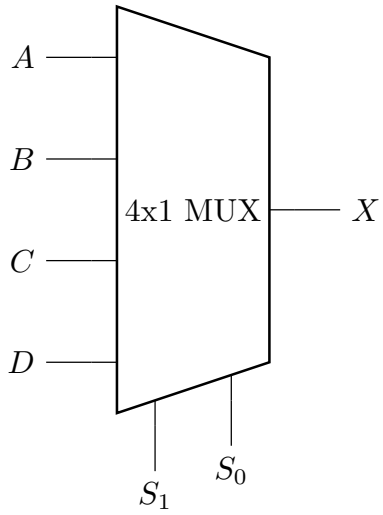


Figure 9: Multiplexer for selecting the input waveforms

Since we need to cycle through the four waveforms repeatedly, we can use a (2-bit) counter output to drive the multiplexer's select lines. So the select signals S_1 and S_0 come from the counter output.

We also need to generate a DC offset for the oscilloscope channel, which is different for each waveform. We can use a D/A converter to convert the select signal $S = (S_1 S_0)$ to an analog signal which will be proportional to S. We can use a resistive ladder digital-to-analog converter (DAC) to generate the offset.

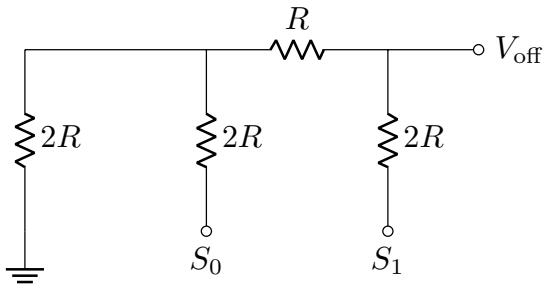


Figure 10: Resistive ladder 2-bit D/A converter. Here, $V_{\text{off}} = (S_1/2 + S_0/4)$.

Assume a digital signal is represented by 0 volts for a logic 0 and V volts for a logic 1. Then, the offsets generated by the DAC will be:

$S = (S_1 S_0)$	Offset
00	0
01	$\frac{V}{4}$
10	$\frac{V}{2}$
11	$\frac{3V}{4}$

Table 5: Offsets generated for the different states

Thus, the step size of the DAC is $V/4$.

Now, we need to add the selected waveform to the DC offset thus generated. To add the analog signals, we can use a operational amplifier (op-amp) as a summing amplifier.

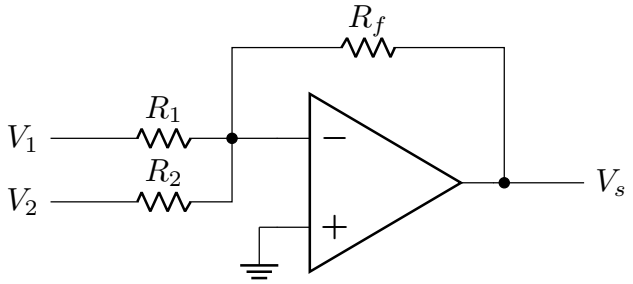


Figure 11: Summing amplifier; $V_s = -\left(\frac{R_f}{R_1}V_1 + \frac{R_f}{R_2}V_2\right)$.

Note that the X waveform (selected by the MUX out of A, B, C and D) is a digital signal which varies between 0 and V volts. We want to show the output of A, B, C and D on the oscilloscope channel in different bands of the total voltage range of 0 to V volts. In accordance with the offsets generated, we can use the following voltage ranges for the different waveforms:

	Voltage Range
A	0 to $\frac{V}{4}$
B	$\frac{V}{4}$ to $\frac{V}{2}$
C	$\frac{V}{2}$ to $\frac{3V}{4}$
D	$\frac{3V}{4}$ to V

Table 6: Voltage ranges for the different waveforms

So, instead of directly adding the X waveform to the DC offset, we can scale down the X waveform to be between 0 and under $V/4$ volts, to realise the above voltage ranges.

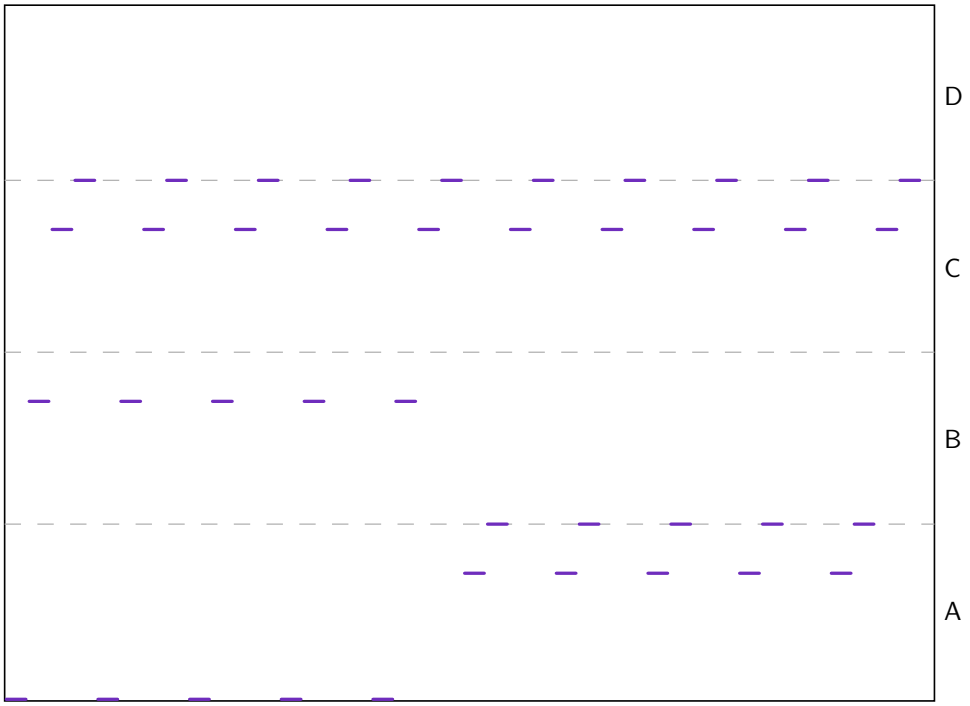


Figure 12: Sample output on the oscilloscope. Patterns for $ABCD = 0110$ and 1010 are shown.

We can choose $R_1 = R_f = 10\,\text{k}\Omega$, and $R_2 = 56\,\text{k}\Omega$, which will give the scaling factors of $R_f/R_1 = 1$ for the offset and $R_f/R_2 = 1/5.6$ for the X waveform, so the X waveform will be scaled to be between 0 and $\lesssim V/4$ volts.

Since the output is inverted, we can add an inverter (op-amp) to the output of the summing amplifier to invert it back to the original voltage range.

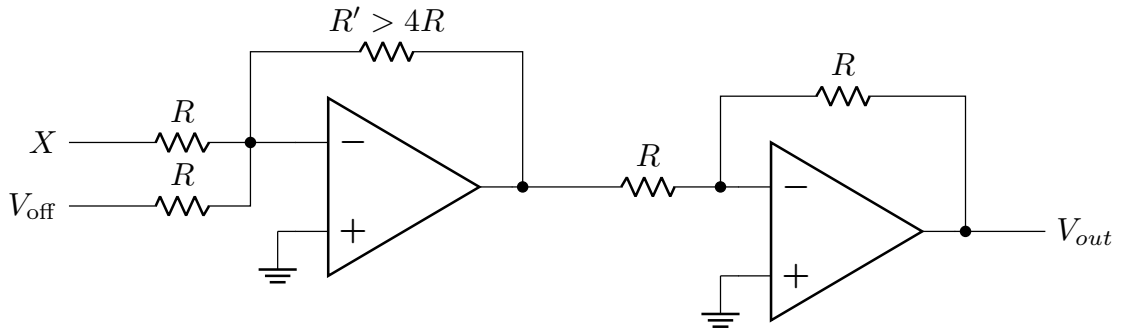


Figure 13: Circuit for adding the input waveform and the offset

To test the circuit, we will use a 4-bit counter whose outputs will be used as the four input waveforms, A, B, C and D. This counter will be driven by a slow clock signal, so that the counter output stays constant for multiple pulses of the fast clock signal, so that we can see the persistent output on the oscilloscope.

Circuit Diagram

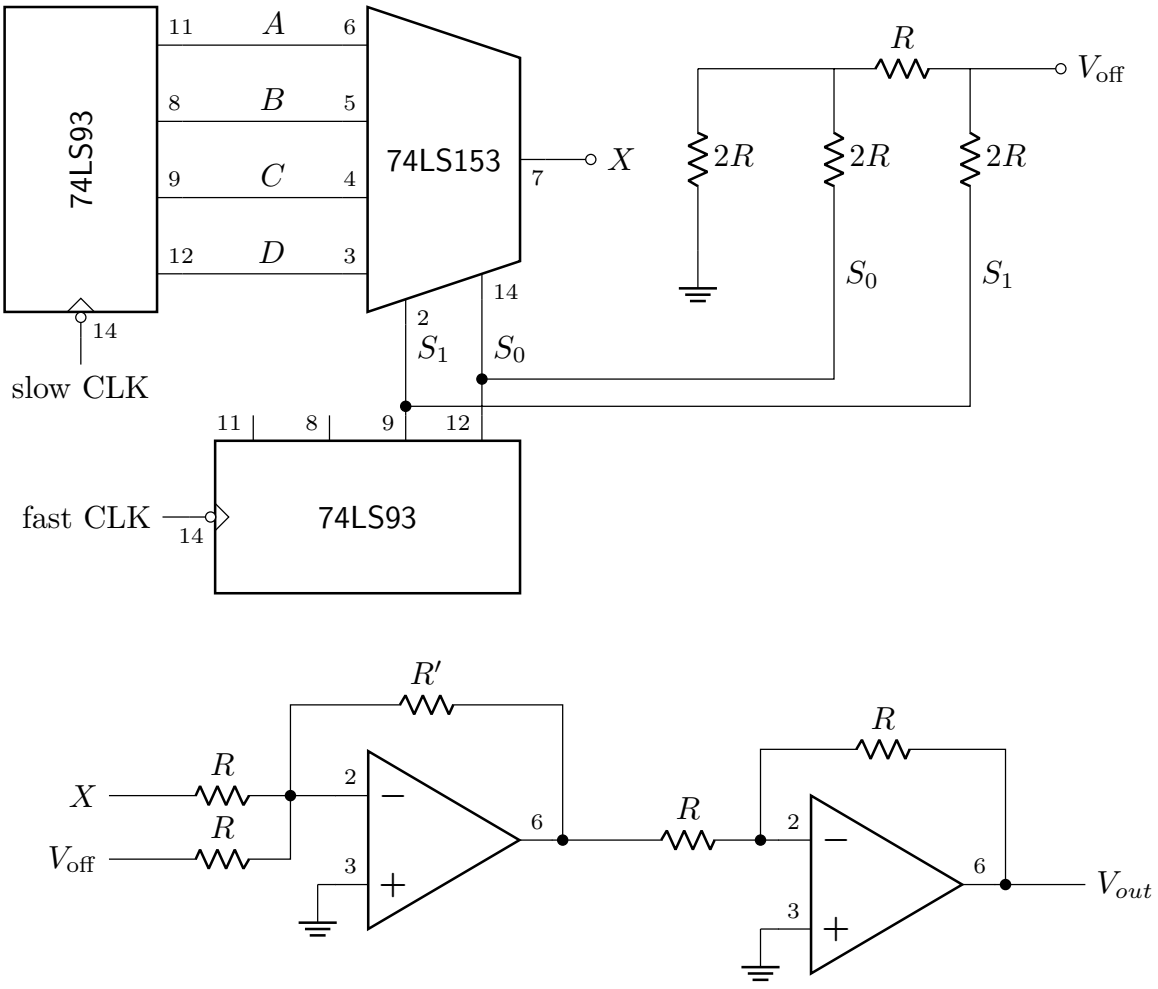


Figure 14: Final circuit diagram

Here, we are generating the slow clock using LM555 IC. The fast clock uses the on-board clock. Resistors used are $R = 10\,\text{k}\Omega$, $R' = 56\,\text{k}\Omega$.

Observations

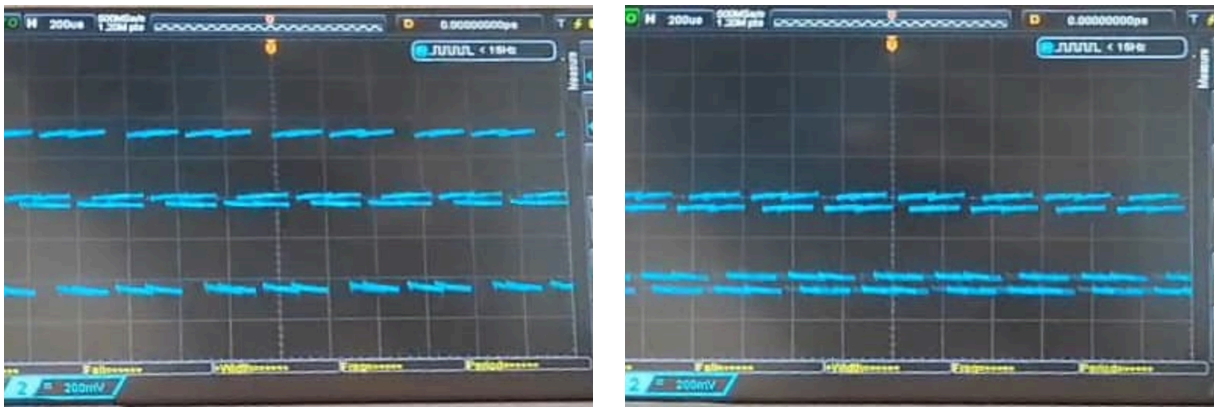


Figure 15: Output signals on the oscilloscope

Here, we can observe the output signals which resembles the expected output (Figure 12). In some cases it is hard to distinguish between two consecutive signals due to their bands being close together. This can be solved by using a higher resistance R' so that the resolution between consecutive signals is higher.