1 Moore's Finite State Machine

Moore's Finite State Machine (FSM) follows a model where the next state of the machine is determined by the current inputs and its current state, while the output is determined by the current state of the machine, following the designed combinational logic.

1.1 Design

Given the specifications required of the pump controller, the functionality of the fsm was represented on Figure 1 and on Table 1, where "LnA" stands for "Last not Activated".

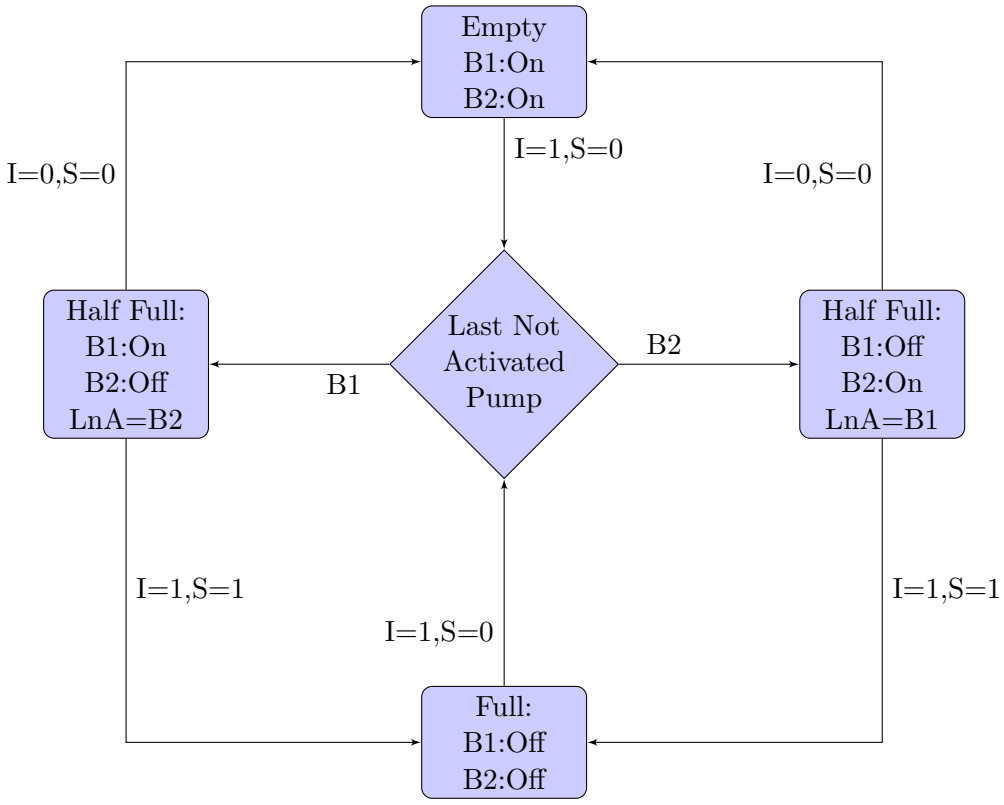

Figure 1: Flow Diagram for Moore's Machine

It can be observed that an error state was added. This was to cover every possible case of inputs. It was decided that in case a "strange" input was received (Superior sensor activated while the Inferior sensor is not), both pumps should stop working, as that seemed to be the safer decision, as this seemed to simply be a replenishment system and an empty tank would alert the user of the error.

After deducing the combinational logic behind the next state assignment and the corresponding outputs for each state, and taking into consideration the available components in the laboratory, the circuit in Figure 2 was obtained.

Current State	Next State				Output	
	S I 0 0	S I 0 1	S I 1 0	S I 1 1	B1	B2
Empty (0 0)	Empty	Half Full	Error	Full	1	1
Half Full (0 1)	Empty	Half Full	Error	Full	∼LnA	LnA
Error (1 0)	Empty	Half Full	Error	Full	0	0
Full (1 1)	Empty	Half Full	Error	Full	0	0

Table 1: State Transition Table

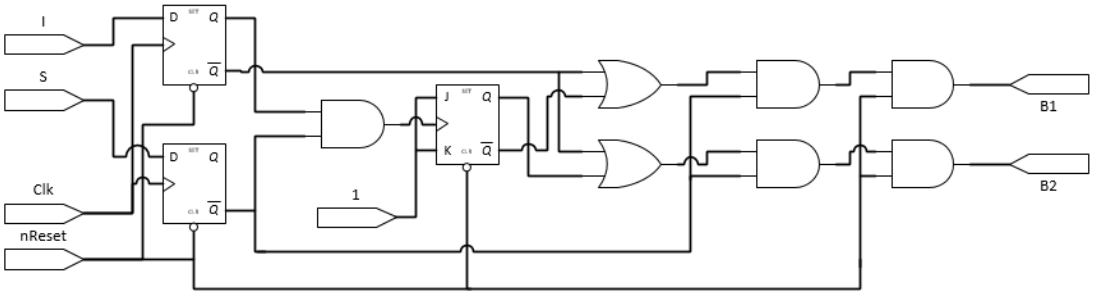

Figure 2: Resulting circuit logic diagram

1.2 Simulation

Before physically implementing the FSM, first it was simulated in Verilog and ran through a testbench, where the waveform in Figure 3 was obtained with gtkwave.

It needs to be mentioned that in the design, a method of resetting the machine was included in case it was needed. This reset case will set the LnA variable to 0 (which arms B1 for use) and deactivate both pumps until it is no longer set.

1.3 Physical Implementation

After confirming the correct behaviour of the device, it was implemented on a Printed Circuit Board with the schematic on Figure 4.

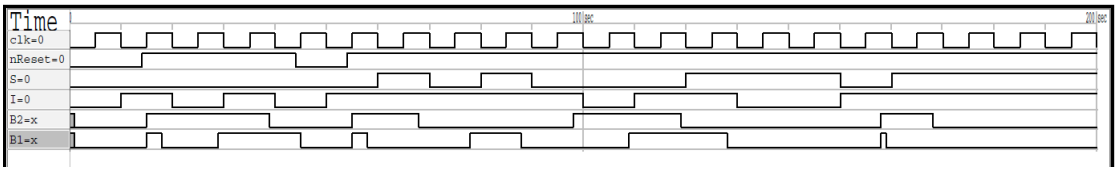

Figure 3: Simulation results from gtkwave

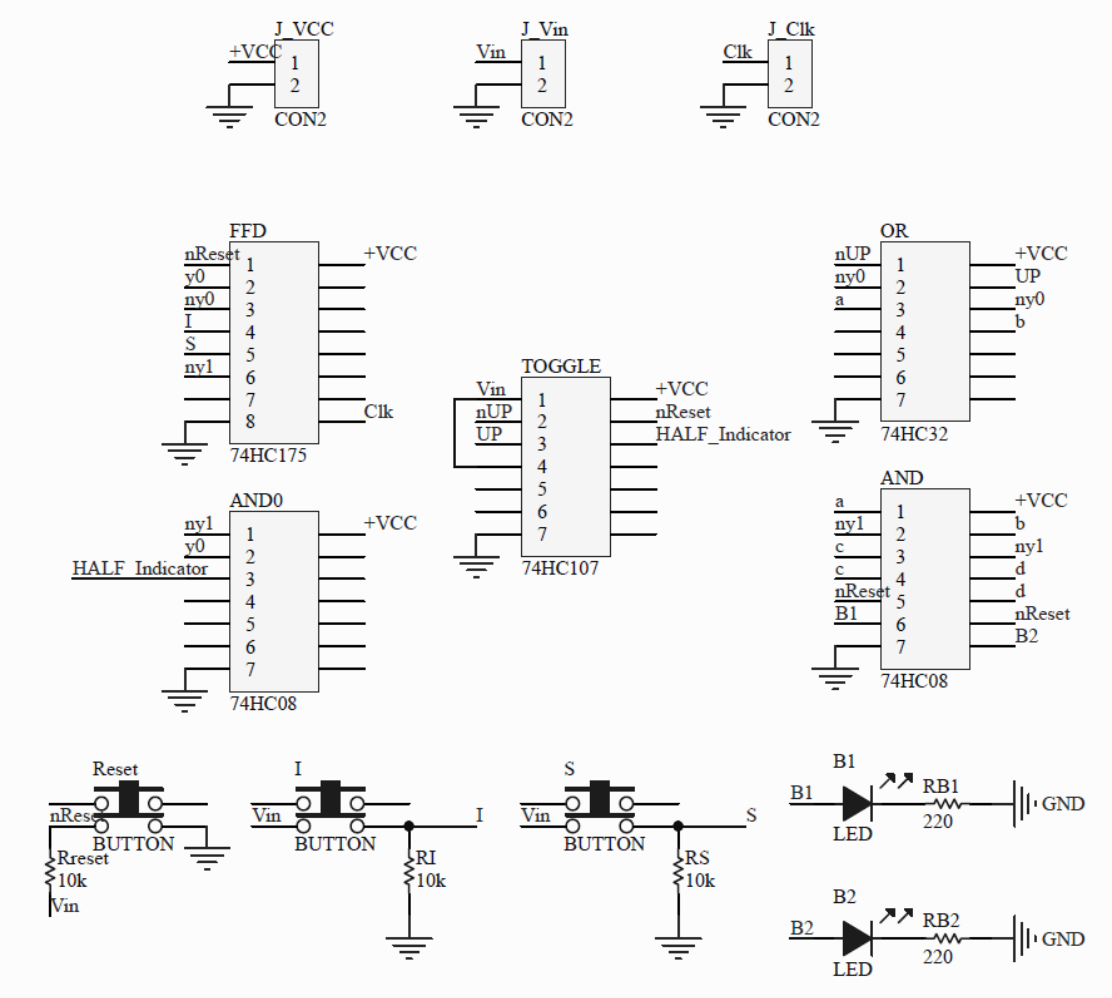

Figure 4: Device Schematic

After it was fully implemented and tested, several measurements were taken to figure out its characteristics. In Figures 6 and 7 both the functionality and the delay between the output and the reset input can be seen and were measured to be 6.6ns.

The delay between the clock's positive edge and the output signals was also measured to be around 30ns in Figure 8.

Lastly, some general testing for the correct behaviour of the machine was tested, which was the one expected and designed, as well as consistent to the one obtained in the Verilog simulation. From Figures 9 and 10 it can be seen that the maximum output voltage is around 4.3V when fed with a 5V source and a minimum of near 0V, taking into account the noise.

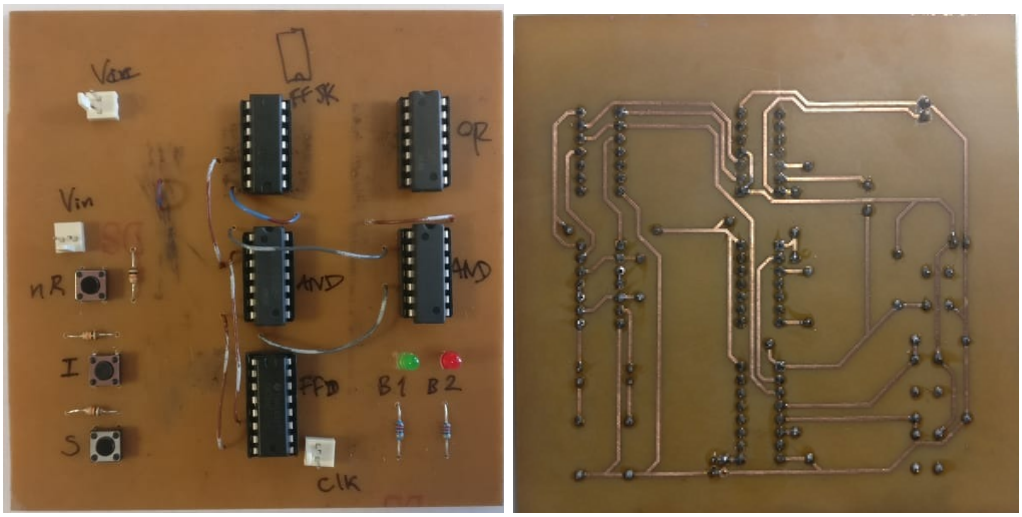

Figure 5: Final Physical device (front/back)

1.4 Conclusions

From all the measurements taken from the physical device and the simulations ran in Verilog, it can be concluded that, given the case use this device would have, it is working as expected, with low delays between input and output signals and accounting for all possible input cases, with a safety precausion of disabling all pumps from activating if there is an error in the system or it is wished to be shut down.

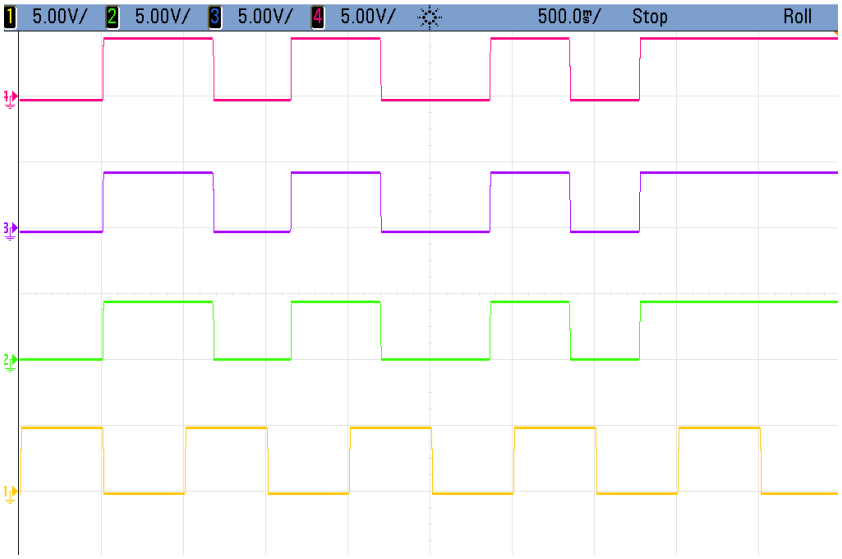

Figure 6: Reset (pink) functionality testing

Figure 7: Reset (pink) delay measurements with B1(green) and B2(blue)

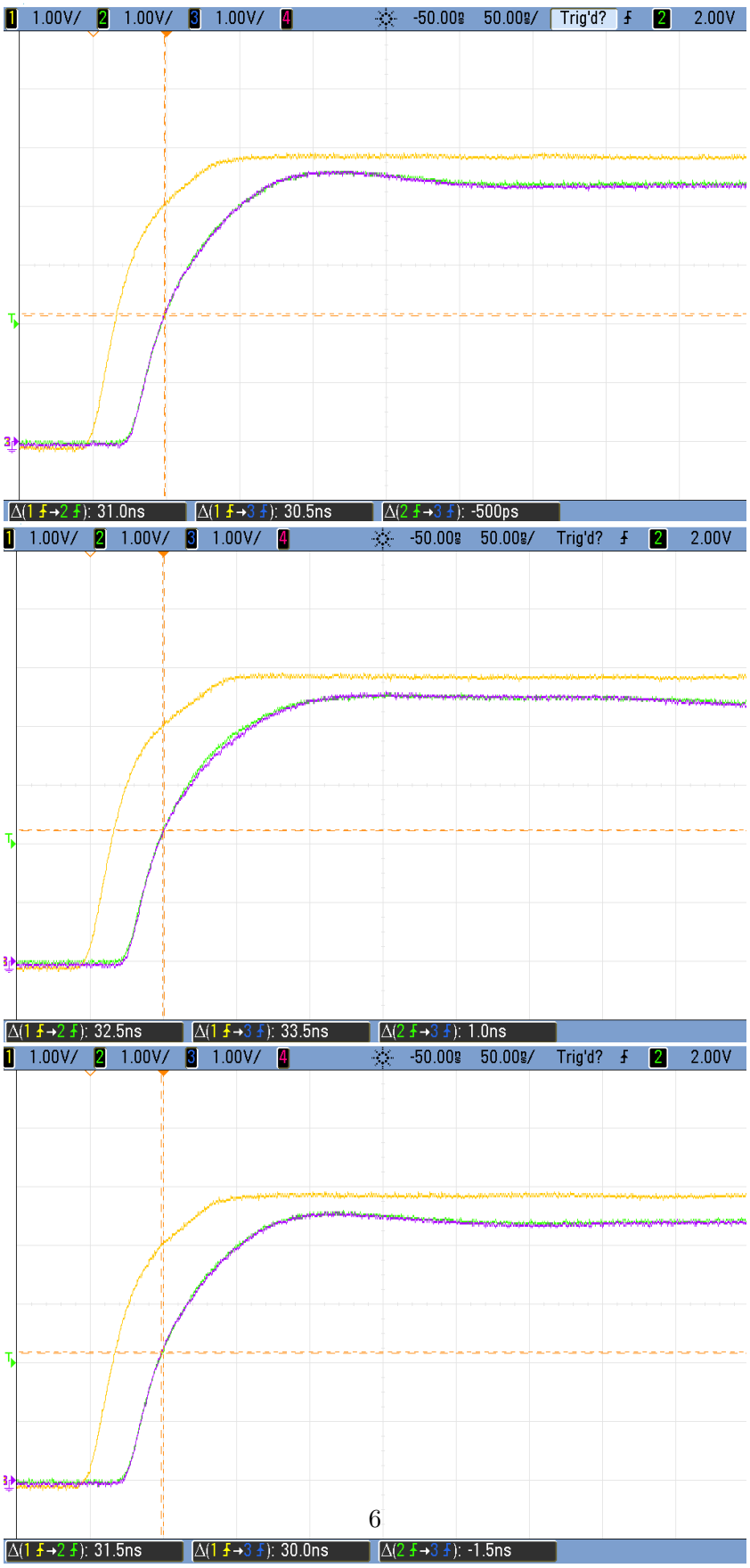

Figure 8: I/O Delay measurements at 1Hz, 1 kHz and 100 kHz respectively

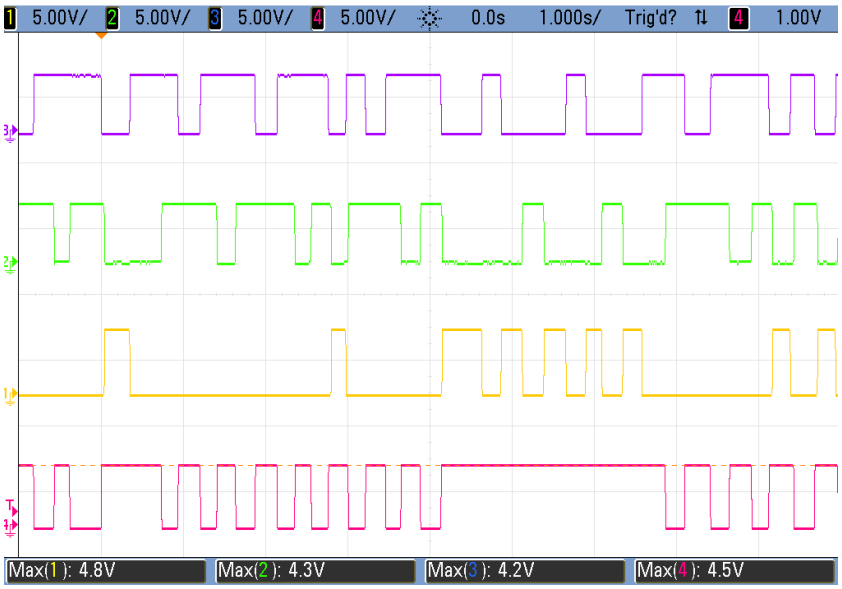

Figure 9: High I/O voltage measurements

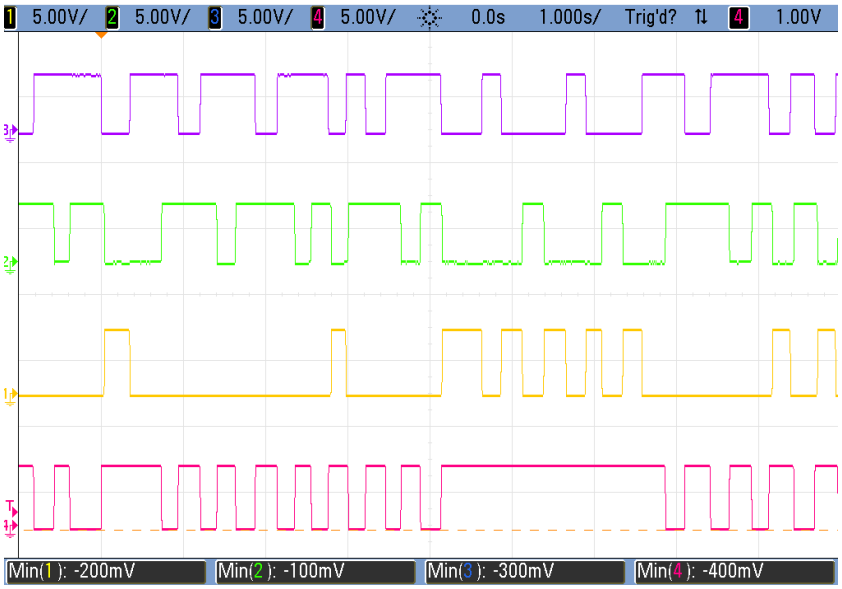

Figure 10: Low I/O voltage measurements