

Trabajo Práctico de Laboratorio Nro. 2

NOWIK, Ariel 58309
MASPERO, Martina 57120
MESTANZA, Joaquin 58288
REGUEIRA, Marcelo 58300
October 17, 2018

Task 3

Given a truth table, it was requested to see what happens when the truth table is implemented with the smallest quantity of logic gates. To perform this task, ICs that have NAND logic gates only will be used. This is because in most of ICs there are many logic gates so it's a waste of logic gates if we use for example just one of them. Note that this can also be implemented with NOR only ICs.

Low cost approach

		ab			
		00	01	11	10
c	0	0	1	0	0
	1	1	1	0	1

Figure 1: Karnaugh Map of the given truth table

Let Z be the output label, its reduced minterms expression is:

$$Z = (\bar{A}.B) + (C.\bar{B}) \quad (1)$$

Also, its reduced maxterms expression is:

$$Z = (\bar{A} + \bar{B}).(C + B) \quad (2)$$

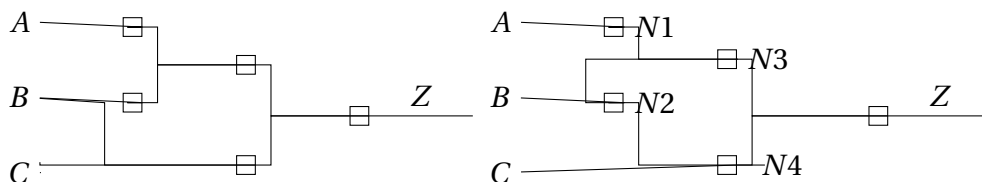


Figure 2: Implementation with NOR gates and NAND gates respectively

Note that the quantity of logic gates in both cases are the same.

Hazards

Sometimes propagation delays cause unexpected and unwanted transitions in the output. Depending on what we see in the output is how this issue is called. If there is a transition from 1 to 0 and the output was supposed to stay at 1 it is said that the circuit has a static 1-hazard. If the output must go from 0 to 1 (or 1 to 0) and before establishing at 1 (or 0), it changes its value it is said that the circuit has a dynamic hazard.

Hazards can always be found at the Karnaugh map. An input that follows multiple paths to an output can create a glitch (due to the hazards in the circuit) if, for example, one path has an

inverter (in this case implemented with a NAND logic gate) and one does not. This issue is called "asymmetric path delay" and it can be seen in Figure 1 with the input B.

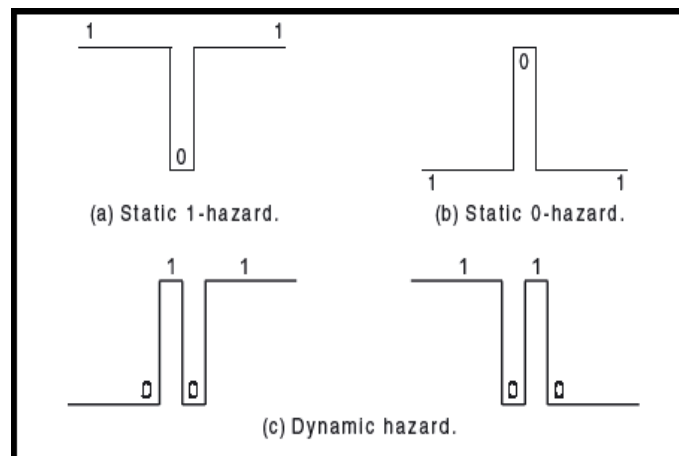


Figure 3: Different types of hazards

Static 1-Hazard example

Take the NAND implementation into account, assume all the propagation times equal and then observe transition of the states A,B,C from 011 to 001 respectively.

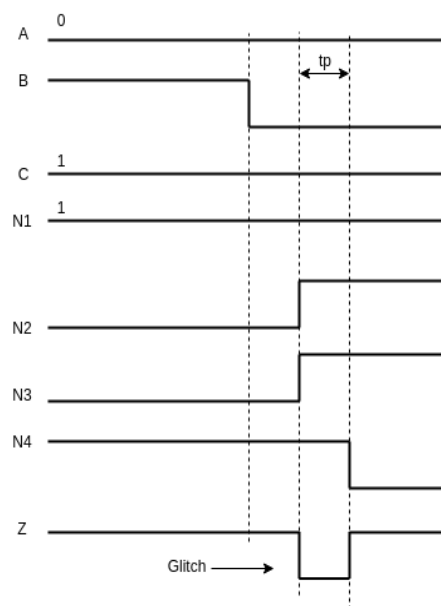


Figure 4: Propagation time analysis

Note that t_p in Figure 4 denotes the propagation time.

It can be seen that there should be a glitch at the output. The different combination of delays that produce a glitch may or may not be likely to occur in the implementation of the circuit. In some instances it is very unlikely that such delays would occur.

Measures obtained from NAND implementation

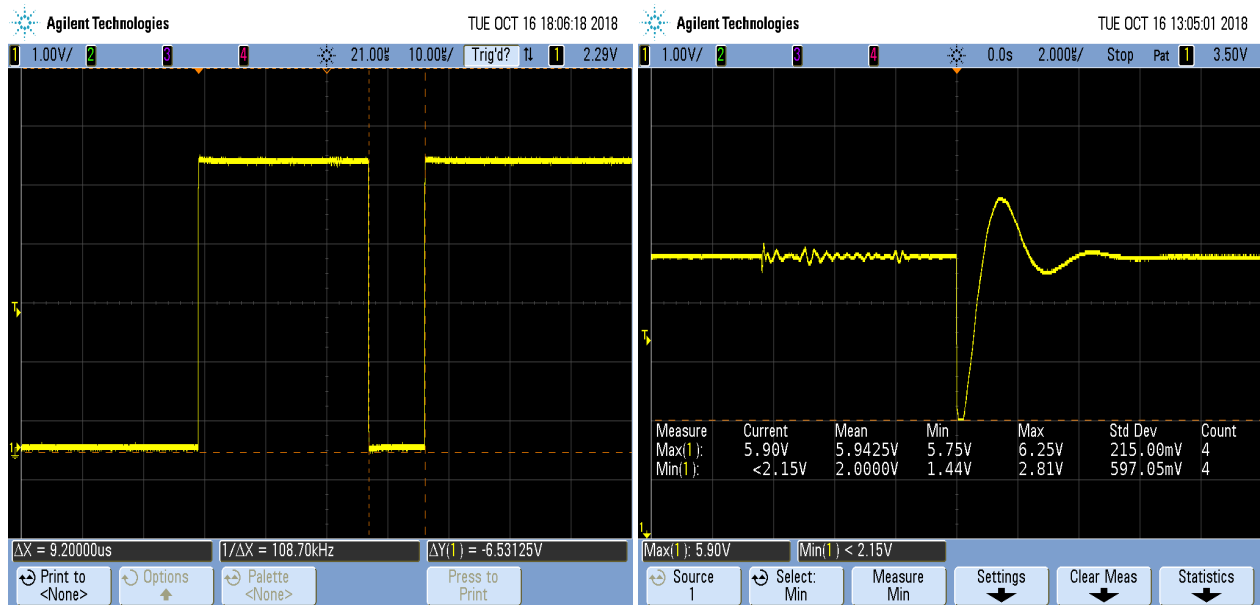


Figure 5: Dynamic hazard and static 1-hazard respectively

Conclusion

In order to remove glitches it is needed to add redundant groups in the Karnaugh Map so that disjoint groups overlap.

		ab			
		00	01	11	10
c	0	0	1	0	0
	1	1	1	0	1

Figure 6: Karnaugh with redundant logic

In this case the output is given by:

$$Z = (\bar{A} + \bar{B}) \cdot (C + B) + (\bar{A} \cdot C)$$

Note that the new term term of the expression is not dependent of B so regardless of the variation of B there won't be glitches in the output.

Task 4

In this section, there are measured the propagation delay, rise time and fall time of a 74HC02 logic gate (CMOS tecnology) in several configurations: without load, and with a circuit load as shown below.

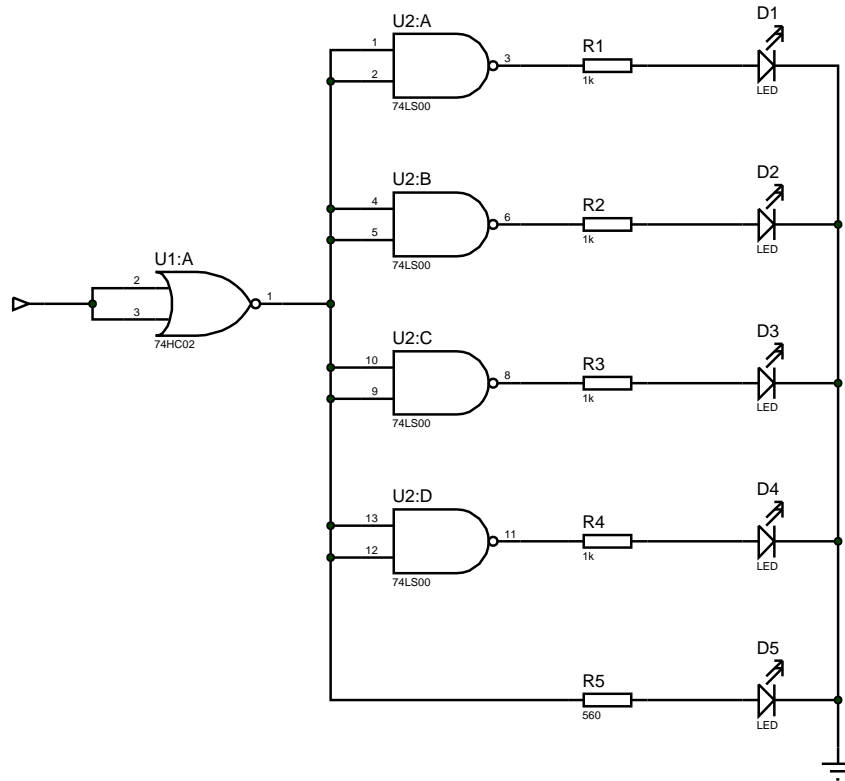


Figure 7: Circuit load schematic - Made in Proteus 7.8

In the following table are the measured times for the diferent configurations.

CASE	tpd_{L-H}	tpd_{H-L}	t_r	t_f
Without LOAD	15ns	5ns	42ns	45ns
Circuit LOAD	17ns	5ns	46ns	45ns
Circuit LOAD (100KHz)	18ns	4ns	45ns	52ns
Circuit LOAD (100KHz with Capacitors)	9ns	4ns	50ns	52ns

Table 1: Measured times

In the first two cases, it was observed that the measured times had no appreciable differences when the gate is charged with the external circuit. In the case with the circuit, the IC raises its temperature a little. This is mainly due to the direct charge of the R5 and L5 componentes, requesting the IC a current that is near to the maximum value of $4mA$ per output.

Also the output voltage level decreases in the case with the circuit charge, compared to the case without load, as shown in the next figures.

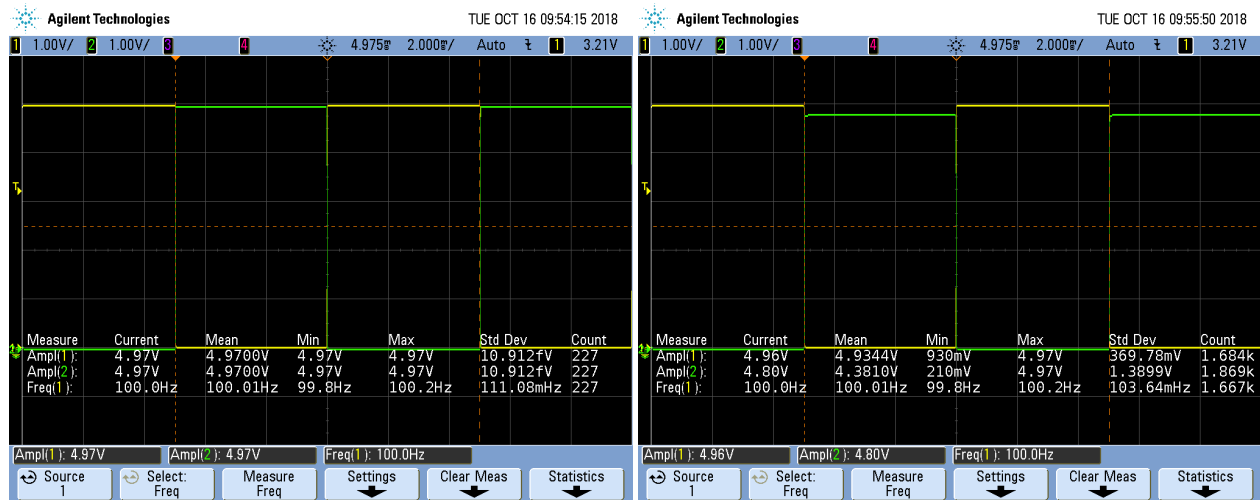


Figure 8: Voltage levels: CH1 - Gate input, CH2 - Gate output. On the left without load, and on the right with the circuit load.

In the same circuit, increasing the frequency up to 100KHz, a decrease was also observed in the output voltage level, with a little oscilation in the transition from low to high. This is because the external circuit requires more current for turning the LED on, which causes a voltage overshoot for a short period of time, until the circuit becomes stable. To absorb this overshoot, a decoupling capacitor is placed near the supply pins of the IC, with a value of 100nF. This value is suggested by the datasheet of Texas Instruments. Below is shown the output change from low to high with the circuit at 100KHz, without and with the capacitor.

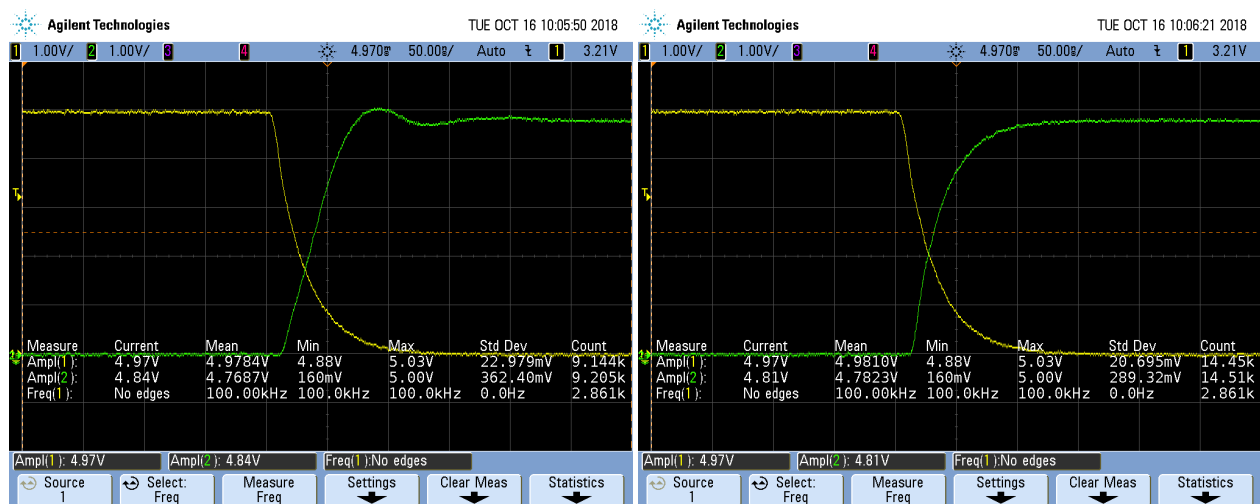


Figure 9: Transition L-H CH1 - Gate input, CH2 - Gate output. On the left without capacitor, and on the right with it.

Another fact observed is the high noise level (about 1V) at the supply voltage. This was also fixed with the decoupling capacitors, as shown below.

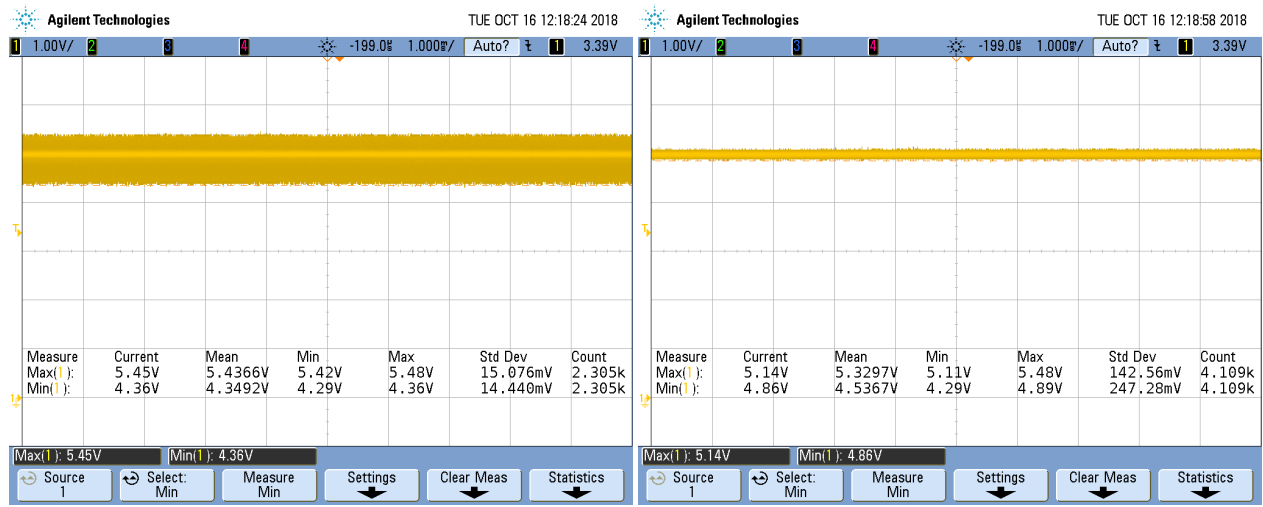


Figure 10: Supply voltage: CH1 - VCC. On the left without capacitor, and on the right with it.

Task 6

In this section, a D Flip-Flop and a SR Latch are implemented, based on logic gates.

SR Latch

For the SR Latch, it is implemented using NOR gates of 74HC02 integrated circuit. The resulting schematic is shown below.

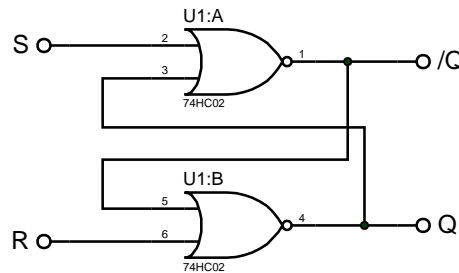


Figure 11: SR Latch circuit - Made in Proteus 7.8

PARAMETROS IMPORTANTES: PROPAGACION DE SET O RESET A LAS SALIDAS. MEDIR. COMPARAR CON UNO COMERCIAL CUALQUIERA

PARAMETER	FROM	TO	VALUE
t_{pd}	Circuit	S	Q
	74HC279	S	Q
t_{pd}	Circuit	R	Q
	74HC279	R	Q

Table 2: Measured values

D Flip-Flop

For the D flip-flop, it is implemented using the SR Latch designed before, adding the remaining parts as shown below.

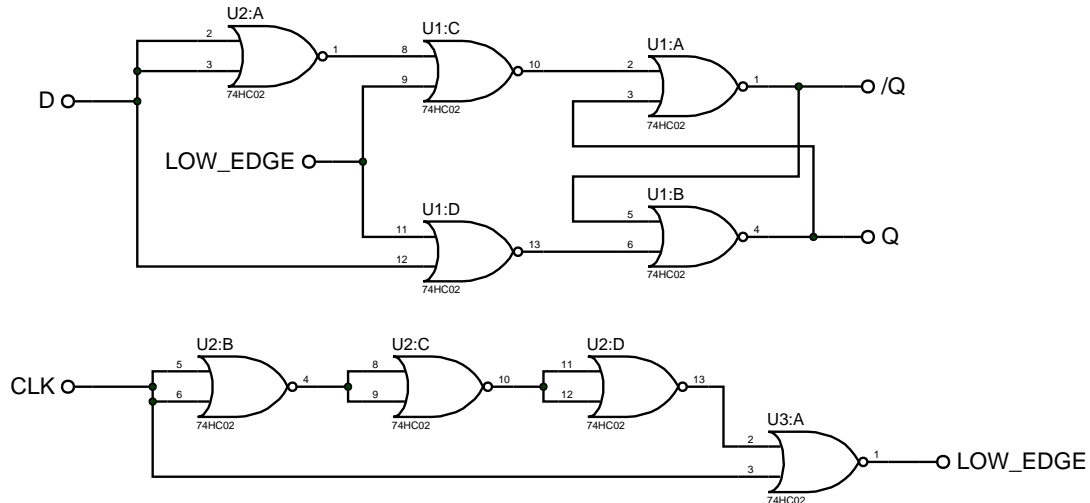


Figure 12: D Flip-Flop circuit - Made in Proteus 7.8

PARAMETROS IMPORTANTES: PROPAGACION DEL D A LA SALIDA LUEGO DEL CLK, TANTO PARA 0 COMO PARA 1. MEDIR. COMPARAR CON UNO COMERCIAL CUALQUIERA. The designed circuit will be compared with the 74HC74 D Flip-Flop, using the datasheet of Texas Instruments.

PARAMETER	FROM	TO	VALUE
t_{pd}	Circuit	CLK	Q or /Q
	74HC74	CLK	Q or /Q
t_t	Circuit		Q or /Q
	74HC74		Q or /Q

Table 3: Measured values

Task 7

In this task we're asked to implement one synchronous, and one asynchronous 3-bit counter, and contrast them.

Async. counter

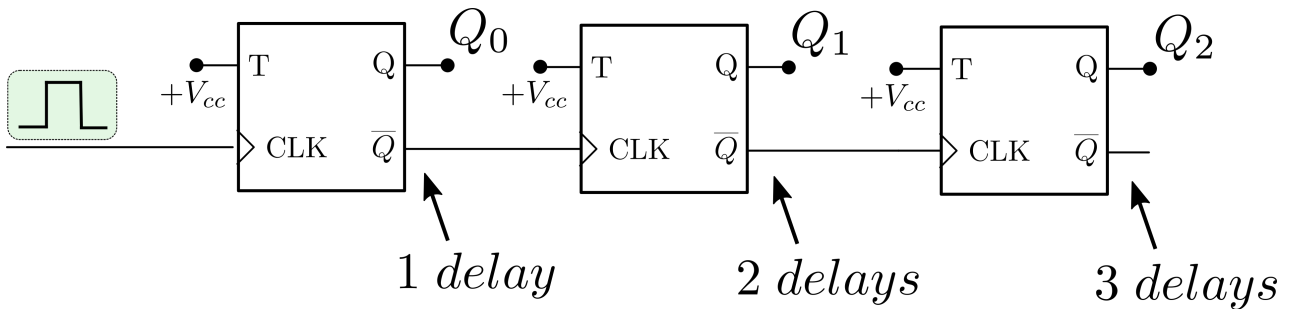


Figure 13: Async. counter - delays

This counter work with a casquade triggering flip flops clocks done by the not Q output.

The idea is that, everytime the left flip flop is clocked two times, not Q goes up one time, thus, right flip flop is clocked one time (the half) Propagating this logic to the general case, n-th flip flop has a period 2^n thus we made a binary counter.

The theoretical problem of this system is the delay propagation. As there is a delay between clock rising edge and Q update, the total delay will be multiplied by the number of flip flops.

Sync. counter

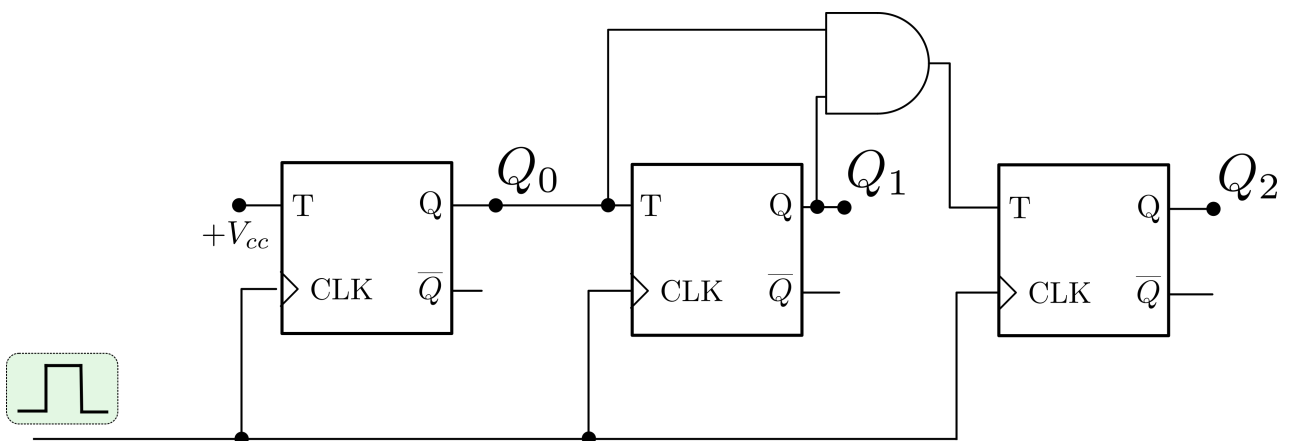


Figure 14: Sync. counter - delays

This counter is made to fix the problem of the previous counter. In this logic, all flip flops clocks are connected to the same trigger, thus, all of them are updated in the same instant, not, one after the other. We control if each flip flop switches by controlling the T input. Flip flop 2 needs flip flop 1 to be in 1 to switch (01->10). Flip flop 3 needs both previous two flip flops in 1 to switch. (011->100)

Measurements

We built the counters for both cases. As expected, in low frequencies differences weren't noticed. But, when we increased frequency to 1Mhz at clock input we were able to see a difference

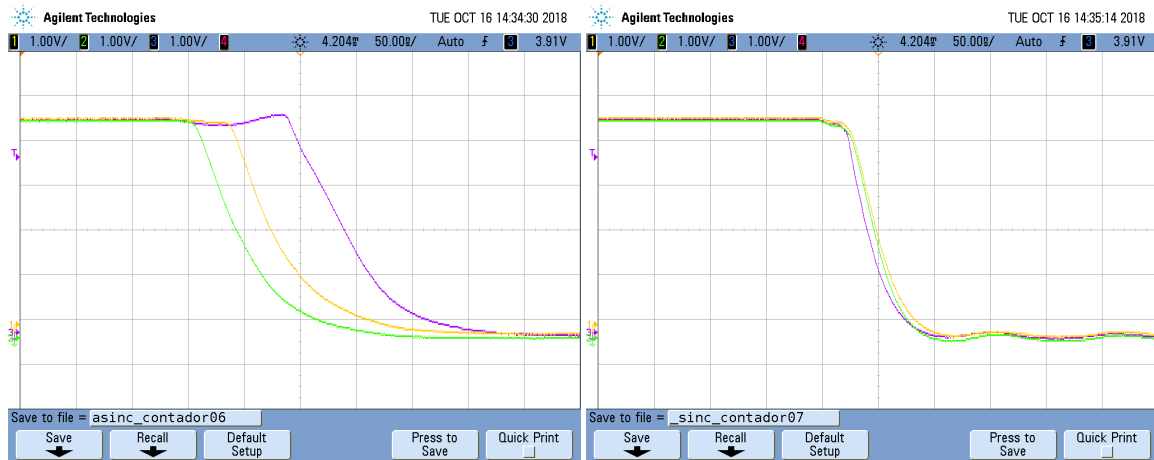


Figure 15: Comparison - 1Mhz clock - Q_0, Q_1, Q_2 signals - async, sync counters, left to right

From figure we can see that in async circuit bits are updated one after the other, while in sync circuit all of them are updated at the same time. Also we should note that updates are not instant, as it was assumed in previous analysis. There is continuous change, that creates an evident limitation to circuit. This limitation made both counters unusable with frequencies slightly higher than 1Mhz (1Mhz was the higher frequency with proper functioning found in both cases), cause the flip flops were switched before the output signal reached its corresponding logic level. So, with 3 bits in practice there is no difference between both circuits, as this limitation impacts both circuits in, apparently, the same way.

Task 8

In this section, a distance measurement system is implemented using discrete logic and the ultrasonic sensor HC - SR04. With it, distances between 1.7cm and 4.25m can be measured. The design is shown in the following block diagram.

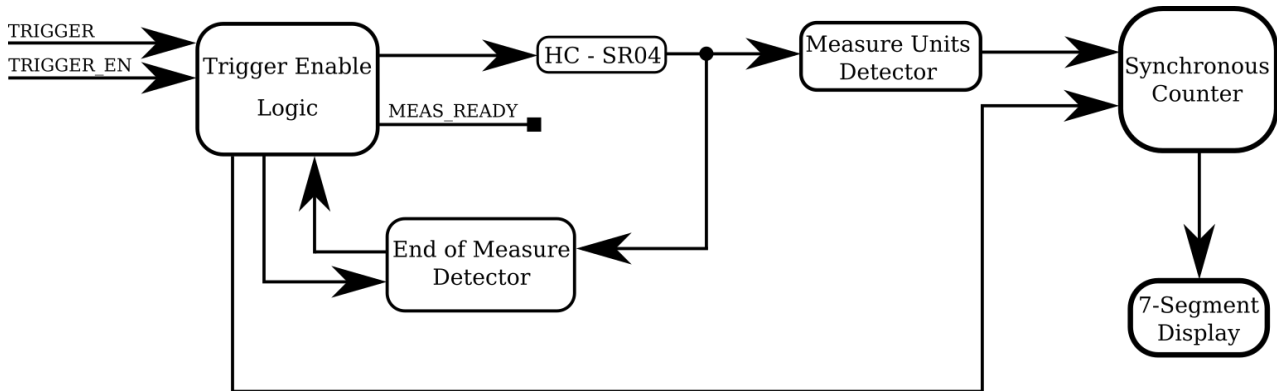


Figure 16: Distance measurement system - Block diagram

Trigger Enable Logic

For this part, a T flip-flop is used to define two states: (1) Measure enabled and (2) Measuring.

In state (1), with TRIGGER_EN on HIGH level, a measure can be made by sending a positive edge to the TRIGGER input, then the flip-flop changes to state (2), preventing a new retrigger while measuring if a new positive edge is detected in TRIGGER input. This two states and the MEASURE_READY bit are referenced from the \bar{Q} output of the flip-flop. When the measure ends, an edge is sent to the CLR pin of the flip-flop, for returning to state (1), allowing to make a new measure. The previous performance is shown in the following time diagram.

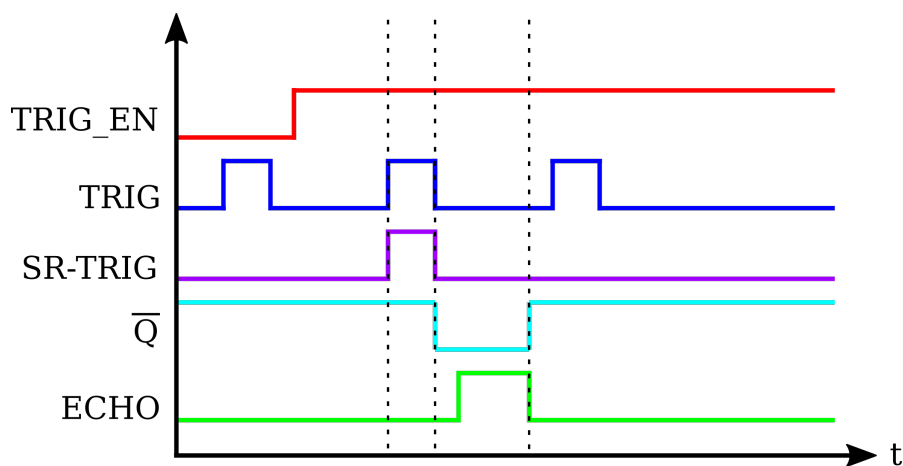


Figure 17: Trigger Enable Logic - Time diagram

The ultrasonic sensor needs a pulse of a period $T > 10\mu\text{Seg}$, so the edge of TRIGGER input is sent to a monostable circuit (implemented with a LM555 timer) to ensure that the condition is met. The final schematic of this section is shown below.

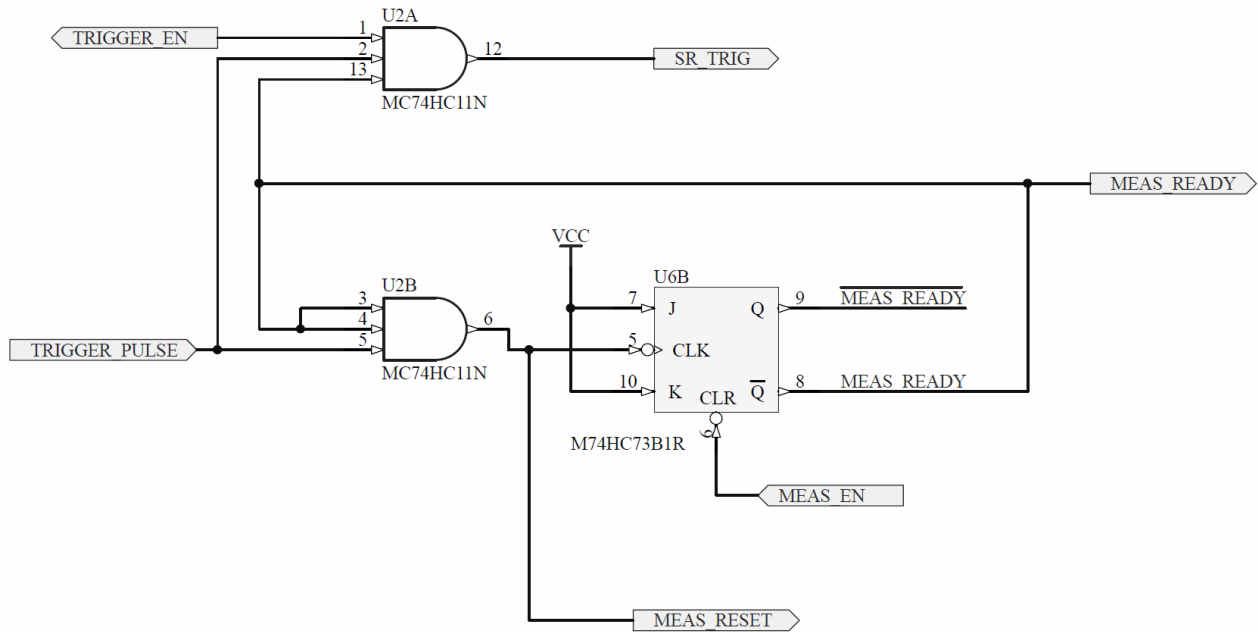


Figure 18: Trigger Enable Logic - Schematic circuit made in Altium 15

HC - SR04 Sensor

To make the measures in the mentioned interval, the HC - SR04 after receiving the TRIG input pulse, it answers at the ECHO out with a pulse of a period $T > 100\mu\text{Seg}$ (1.7cm), up to $25000\mu\text{Seg}$ (4.25m). It will be fractioned in units of $100\mu\text{Seg}$ (to be counted).

To know the measure value, the obtained number N is processed in the following calculation:

$$Distance[m] = 170 \cdot 100 \cdot 10^{-6} \cdot N$$

By fractioning the ECHO pulse in discrete units, the resulting resolution is equivalent to 1.7cm, which is the smallest unit that can be counted ($100\mu\text{Seg}$).

Measure units detector

For fractioning the ECHO response into units of $100\mu\text{Seg}$, it is connected to an AND gate with a clock source (CLK) whose period is $100\mu\text{Seg}$.

In this way, while the ECHO out stays at HIGH level, the signal at the AND gate output is equal to the clock source. While not measuring, the output stays at LOW level. The previous performance is shown in the time diagram below.

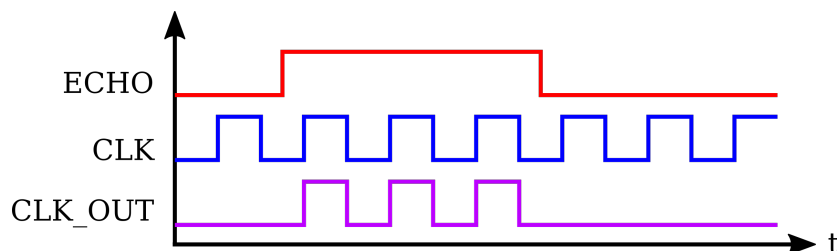


Figure 19: Measure units detector - Time diagram

The CLK signal was implemented with a LM555 timer in astable configuration. The schematic for this section is shown below.

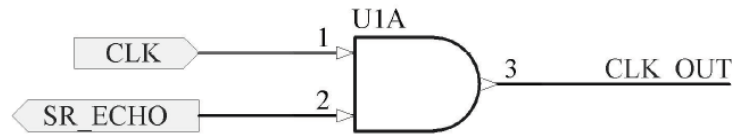


Figure 20: Measure units detector - Schematic circuit made in Altium 15

Synchronous counter

For counting the time units, an integrated synchronous counter is implemented (CD4040), whose CLK input is connected to the out signal of the measure units detector. It detects the positive edges of the CLK signal, in this case having one per $100\mu\text{Seg}$. From the 12-bit output, only 8-bit are used, because for measuring the maximum distance, only 250 units are needed ($25000\mu\text{Seg}$ over $100\mu\text{Seg}$ results in 250 units), and if the binary out is converted to decimal, the maximum number is $2^8 = 256$ (which covers the 250 maximum).

The integrated circuit implementation is shown below.

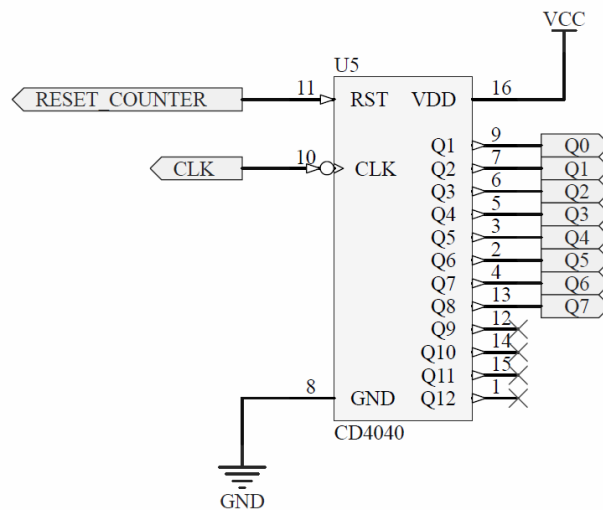


Figure 21: Synchronous counter - Schematic circuit made in Altium 15

End of measure detector

When the measure ends, the negative edge produced by the ECHO is used in a discrete edge detector to reset the flip-flop in the trigger enable logic, so a new measure can be started at any time. The circuit diagram of the implementation is shown below.

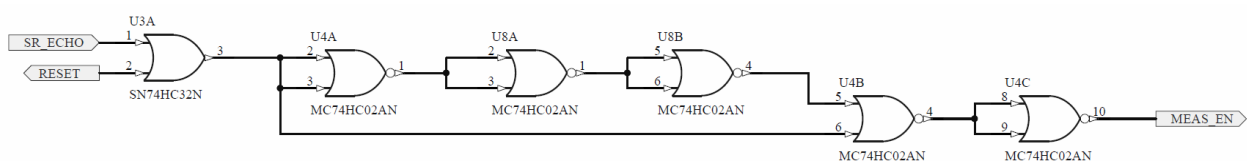


Figure 22: End of measure detector - Schematic circuit made in Altium 15

The binary output resulting from the measure stays on until a new positive edge from the trigger is received. Then, the pulse used to change the flip-flop state in a new measure is also used to reset the previous binary output from the synchronous counter.

Manual reset

If when connecting the power supply to the circuit, the counter is not in zero, or the flip-flop starts switched into the state (2), a manual switch for reset is added with a pull-down resistor, as shown below.

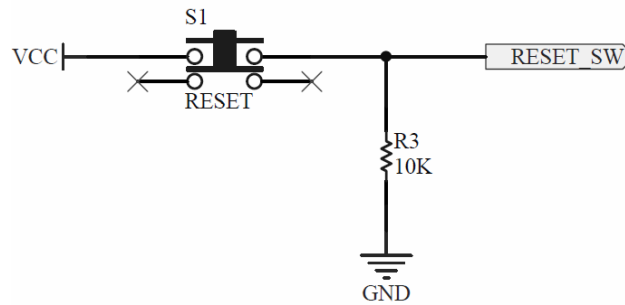


Figure 23: Manual reset with pull-down resistor - Made in Altium 15

Additional settings

MEASURE_READY indicator

To know if the flip-flop starts in the correct state when connecting the power supply or if the last measure has finished, a bicolor LED was used through two transistors, using the Q and \bar{Q} outputs of the flip-flop. When a measure ends (or when at the power supply connection the flip-flop starts at the correct state), the LED turns green as an OK indication. When measuring, the LED turns red until it ends. If when connecting the power supply (or in any moment) the flip-flop starts in the wrong state, the LED will remain in red. This can be fixed by pressing the reset switch. The schematic circuit is shown below.

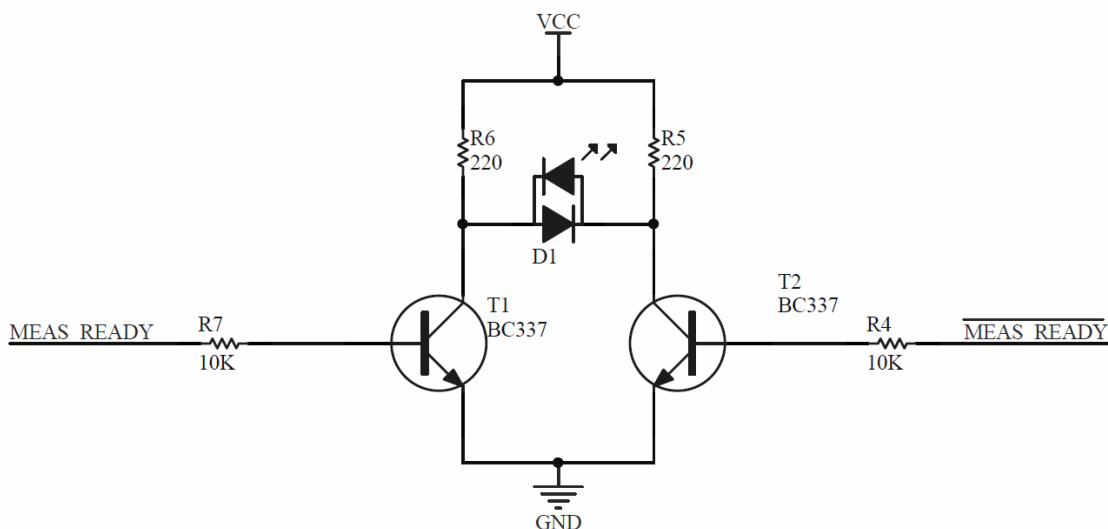


Figure 24: MEASURE_END indicator - Transistor logic made in Altium 15

The calculation of the resistors can be read from the Annex.

Measure counted units indicator

To obtain the amount of time units measured more easily, the CLK out of the measure units detector is used in decade counters implemented with CD4033, because their out pins are decoded for use in 7-segment displays. Since the maximum measure has 3 digits, there are 3 counters and displays implemented (their reset pin is connected to the reset signal of the end of measure detector, and to the reset switch through an OR gate). The implemented circuit schematic is shown below. It has been done in a separate PCB.

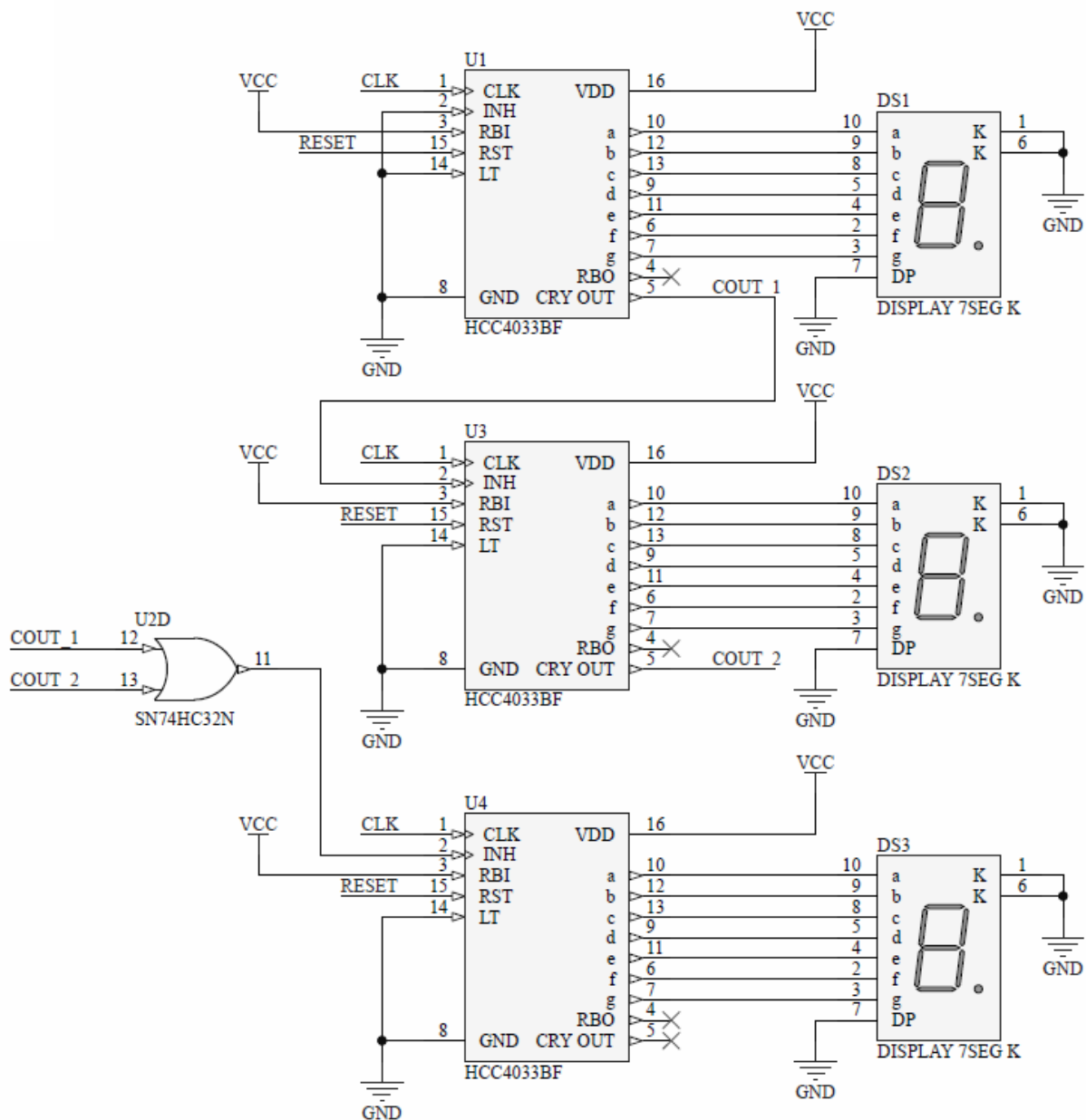


Figure 25: Measure display - Schematic circuit made in Altium 15

Appendix

MEASURE_READY indicator - Resistors calculation

For the circuit design, the transistors are used in saturation-cutt off mode. Considering a $15mA$ current for the LED, it will be the IC_{SAT} . Going through the out loop we have:

$$V_{CC} - IC_{SAT}R_C - V_{LED} - V_{CE_{SAT}} = 0$$

Considering as $V_{CE_{SAT}} = 0.2V$ and a $V_{LED} = 1.8V$, the value of R_C (collector resistor) can be obtained as follows:

$$\frac{V_{CC} - V_{LED} - V_{CE_{SAT}}}{IC_{SAT}} = R_C = 200\Omega$$

Normalizing the value, it remains as $R_C = 220\Omega$. Recalculating the new IC_{SAT} :

$$\frac{V_{CC} - V_{LED} - V_{CE_{SAT}}}{R_C} = IC_{SAT} = 13.6mA$$

Using the BC337 transistors, according to ON Semiconductor datasheet, the minimum HFE is 100, which will be used to calculate the minimum IB_{SAT} (because its the worst case, to guarantee the saturation):

$$\frac{IC_{SAT}}{HFE_{MIN}} = IB_{SAT-MIN} = 136\mu A$$

With that value, the base resistors can be calculated, and then normalized down to get an IB_{SAT} over the previous limit. Going through the entry loop:

$$V_{Q-OUT} - V_{BE_{ON}} - IB_{SAT}R_B = 0$$

Considering that $V_{BE_{ON}} = 0.7V$ and the output voltage of the flip-flop as $5V$ on HIGH level:

$$\frac{V_{Q-OUT} - V_{BE_{ON}}}{IB_{SAT}} = RB_{MAX} = 31.6K\Omega$$

Normalizing the value, it results in $R_B = 10K\Omega$. Checking the resulting IB_{SAT} :

$$\frac{V_{Q-OUT} - V_{BE_{ON}}}{R_B} = IB_{SAT} = 436\mu A$$

According to maximun ratings of Texas Instruments datasheet, the Q and \bar{Q} outs can manage a current up to $25mA$, so the resulting IB_{SAT} is in range.