Universidade do Minho
Escola de Engenharia

Manchester-Based Master-Slave Bus Interface

Report prepared by Diogo Gomes Rodrigues

Summary

It is considered a Master-Slave (bus) structure in which communications take place between the structures using Manchester Code and an open-drain data.

1. Introduction

This work focuses on a Master-Slave communication structure that enables multiple sensors to interact with the master device. The communication protocol follows the Manchester encoding method and operates using two open-drain lines: a clock line and a data line. The master device is responsible for selecting which sensor transmits data and controlling the start and end of communication.

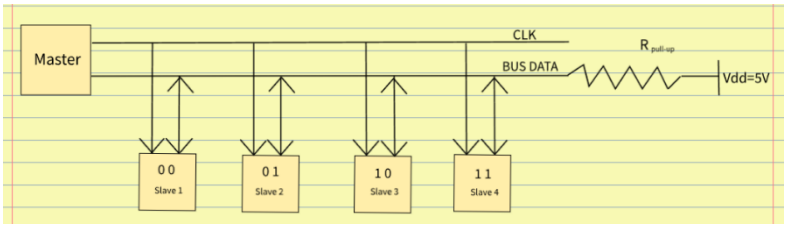
Manchester encoding (see Table 1) ensures synchronization between the bus data and the clock because it enforces a level transition at every clock cycle. This eliminates the need for additional error control bits, as errors can be detected by monitoring synchronization. This encoding is beneficial since, for example, if logic levels “0” and “1” were represented by 0V and 5V, respectively, a communication failure could cause the receiver to misinterpret a signal as a logical "0." However, with Manchester encoding, this issue is avoided.

CLK	10	>	1
Data	10		
CLK	10	>	0
Data	01		
CLK	10	>	Idle
Data	11		
CLK	10	>	Null
Data	00		

Table 1 - Manchester Encoding

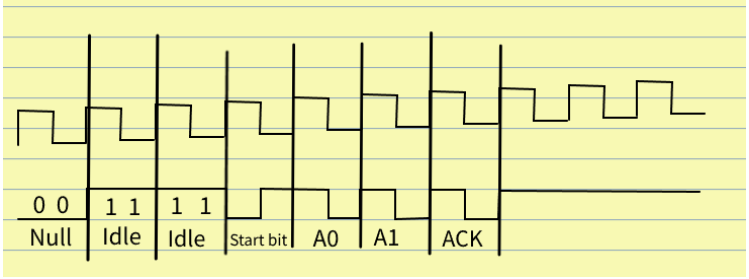
The open-drain lines include pull-up resistors, ensuring that the default logic level remains “1.” This configuration allows multiple sensors to share the same communication line, enabling both reading and writing.

All sensors continuously monitor the data line. When a sensor recognizes its assigned address from the master, it sends an acknowledgment signal and begins data transmission. The data line must be bidirectional, ensuring that only one sensor writes at a time. Meanwhile, the clock line only needs to be readable, so it does not require an open-drain configuration.



Scheme 1 – Master-Slave communication

Initially, the data line remains in an idle state ("11"). When the master initiates communication, it sends a start bit ("01" in Manchester encoding, equivalent to logic 0). The sensors then wait for the two address bits. If these bits match a sensor’s address, it transmits an acknowledge signal ("10") and proceeds with data transmission.



Scheme 2 – Manchester code with clock and bus data

In this project, the structure was implemented for a single sensor with an

address of "11," which transmits only logical "1s." However, since the address consists of two bits, the system can accommodate up to four different sensors.

The communication system was designed using S-Edit software. For simulation purposes, an input signal ("0011110110101010111") was generated. The clock signal was created using repeated "10" sequences. The rise time (RT) and fall time (FT) were set to minimal values to ensure rapid logic transitions.

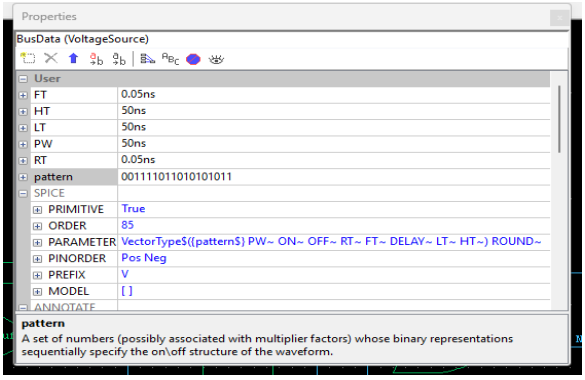


Figure 1 – Bus data signal

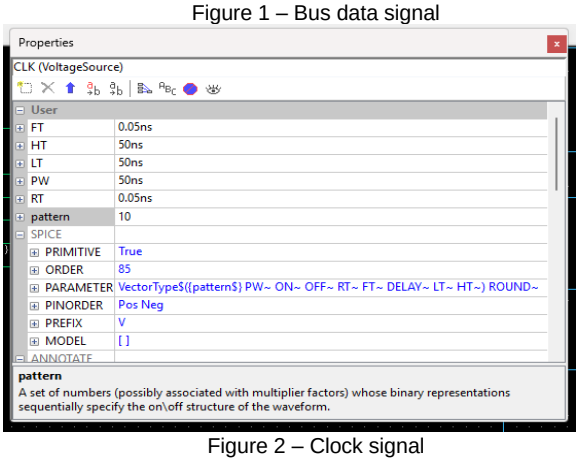


Figure 2 – Clock signal

2. Manchester decoder

A key component of this system is the Manchester decoder, responsible for interpreting incoming signals.

The bus data is fed into a p-latch, which is created by placing an inverter before the clock signal of an n-latch. Unlike flip-flops, p-latches operate throughout the clock cycle, providing continuous memory storage, which is essential for Manchester decoding.

- The p-latch allows data to pass through when the clock is at logic 1 but holds the previous value when the clock is at logic 0.
- At the p-latch output, "00" represents a null state or logic 0, while "11" represents idle or logic 1.

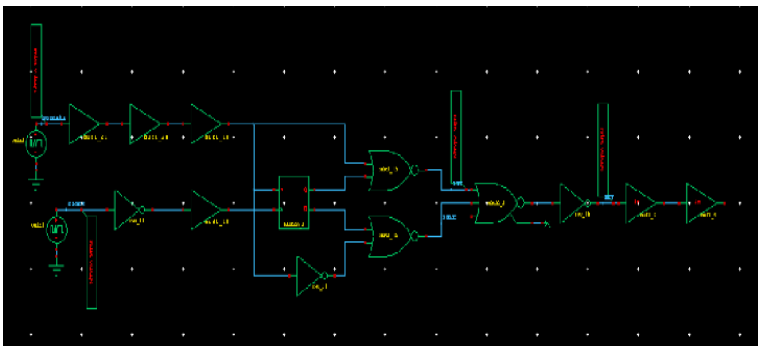


Figure 3 – Manchester decoder

In the following logic gates, the outputs Q and negated Q of the latch are compared with the input signal to distinguish between a logical value and a null or idle state. The output is "10" if the incoming signal represents a logical value or "11" otherwise.

In the first NOR gate, the Q output of the latch is compared with the bus data to distinguish between a null state and the logical value zero. Thus, the INT signal will be "11," "10," or "00" if the

input is null, logical "0," or any other case, respectively.

In the second NOR gate, the negated Q output is compared with the bus data to differentiate between an idle state and the logical value "1." As a result, the IDLE signal will be "11" if the input is idle, "10" if it represents logical "1," or "00" for any other cases.

The INT and IDLE signals are fed into an OR gate, generating the SET signal, which takes the value "10" for a logical value and "11" otherwise. This signal is used to control the flip-flops. Later, this OR gate will also receive the ERROR signal so that, in the event of an error (e.g., loss of synchronization between systems), the SET signal is activated, triggering the setting of the flip-flops.

Clk	Busdata	Q	Q negada	INT	IDLE	SET
10	**00**	00	11	11	00	11
10	**01**	00	11	10	00	10
10	**10**	11	00	00	10	10
10	**11**	11	00	00	11	11

Table 2 – P-Latch, NOR and OR response

Buffers are placed at the output of the clock, bus data, and SET signals to ensure they have enough power to be applied to other points in the system.

The IDLE, INT, and SET signals obtained at the end of the Manchester decoder are visible in the following graphs.

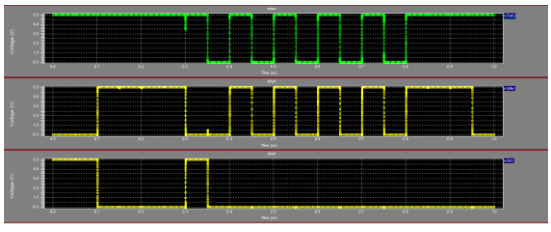


Figure 4 – IDLE, INT and SET signals

3. Serial-to-parallel converter

The circuit that enables the discrimination of the start bit and the addressing sent by the master is shown in Figure 5.

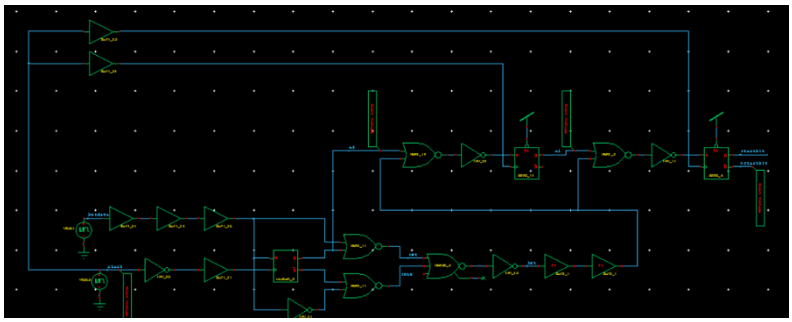


Figure 5 - Serial-to-parallel converter that allows the collection of addressing bits and the startbit

When the SET signal has the value “11,” the flip-flops are set, and their output becomes “1,” meaning the line is held at a logical “1.” When SET has the value “10,” the flip-flops output either “1” or “0” depending on the logical input value.

When the start bit “01” is sent, the latch output is “00,” and the SET value becomes “10.” As a result, the output of the OR logic gate is “10,” and the flip-flop output is “10.” This means that the line, which was previously at 1, is set to 0.

Clk	Busdata	Q	SET	*Flip-flop*
10	**00**	00	11	11
10	**01**	00	10	10
10	**10**	11	10	11
10	**11**	11	11	11

Table 3 – flip-flops exit signals

After the start bit, the addressing bits appear. Since the flip-flops introduce delays in the signals and two flip-flops are used, it is possible to observe the start bit signal and the addressing bits (A0 and A1). At the output of the second flip-flop, the start bit signal is visible, while at the output of the first flip-flop, the first addressing bit can be seen, and at the output of the p-latch, the second addressing bit is visible.

The start bit marks the beginning of data transmission, and the addressing

bits allow the identification of the sensor to which the information is intended to be sent. The sensor corresponding to the address must recognize it and send a signal confirming that the recognition has been completed.

The obtained signals can be visualized in Figure 6.

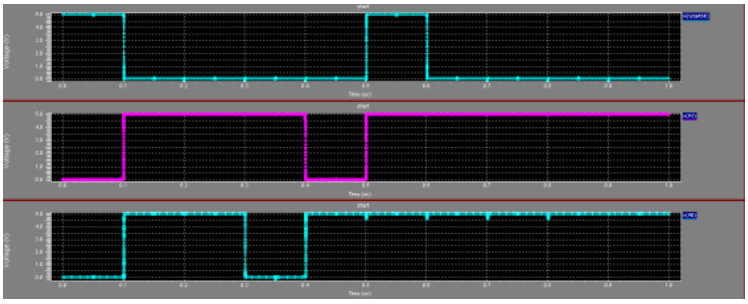


Figure 6 – Start bit, A1 and A0 signals

4. Slave brain

The circuit presented below allows the generation of the STOP signal. This signal differentiates whether the start bit is present or not. When two idle states arrive, the signal changes to 1, and when a start bit appears, it changes to 0.

A flip-flop is used to delay the signal by one period so that the STOP signal changes only after the start bit. When a new idle state appears, the signal switches back to 1.

The obtained signal can be visualized in Figure 7.

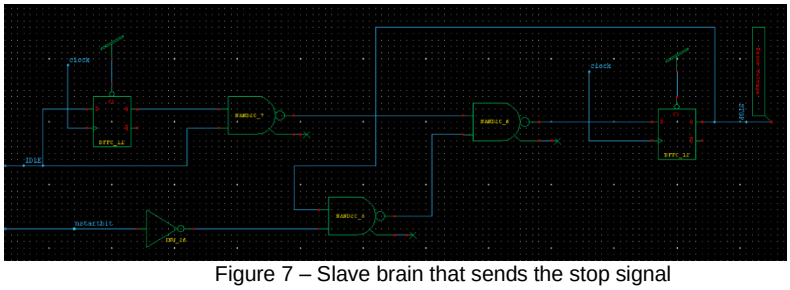


Figure 7 – Slave brain that sends the stop signal

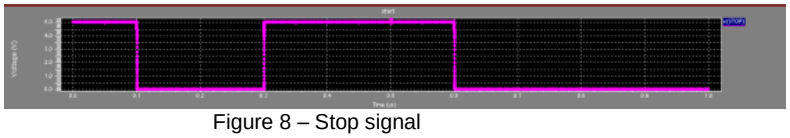


Figure 8 – Stop signal

5. Sending the ack (Aknowledge) bit

The presented circuit provides an example of a sensor with the address "11." This address is set by the buffers. The signals A0 and A1 are compared with the sensor's address, and if they match, the XOR gate outputs a logical "1."

If the address present on the bus data matches the sensor's address, and both the STOP signal and the start bit are "1," then the output of the NAND3 gate is "1," triggering the acknowledge (ACK) signal.

This signal is placed on the bus data, indicating that the sensor has recognized it was being addressed and is ready to start transmitting data.

A flip-flop with a set input is used to delay the signal so that the ACK appears after the start bit. At this moment, the set signal is "10", as it corresponds to the input of the second addressing bit and, therefore, to a logical value.

This flip-flop is implemented by placing an inverter and an OR gate at its input.

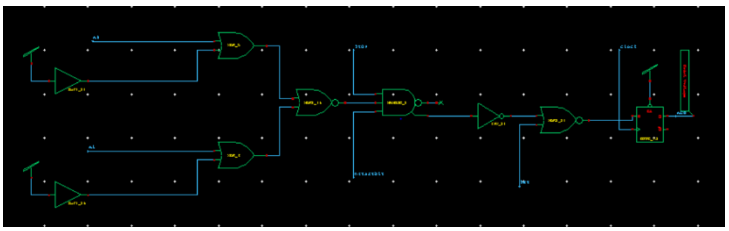


Figure 9 – Circuit that sends the ACK bit

Figure 10 – ACK signal

6. Sending the enable bit after ack bit

The ENABLE signal allows the sensor to write to the data line. When it is "1", the sensor can send data; when it is "0", the sensor can no longer place data on the data line.

Initially, both ACK and ENABLE are set to zero. When ACK becomes "1", ENABLE is activated and also set to "1". To ensure that ENABLE only appears after ACK, a flip-flop is used to delay it.

The ENABLE signal returns to "0" when the set signal becomes "11", meaning that an idle state appears on the bus data. This indicates that communication has ended, and the sensor must stop transmitting data.

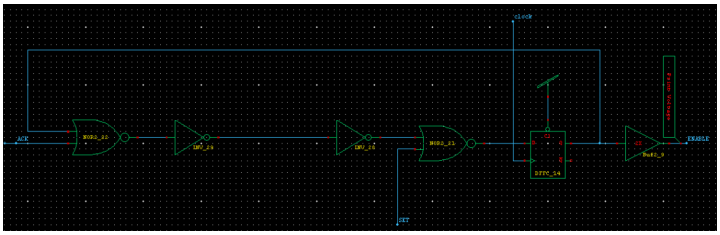


Figure 11 – Circuit that sends the enable signal

Figure 12 – Enable signal

7. Manchester encoder

For the sensor to send data to the line, it must first encode the data using Manchester coding.

The encoding is done using an XNOR gate, with the clock signal and the data from the sensor as inputs. Since the sensor is sending logical "1"s, and in an XNOR gate, the output is 1 when both inputs are equal and 0 when they are different, the comparison with the clock signal results in multiple "10" sequences, which represent "1" in Manchester code.

The encoded signal is then compared with the inverted ENABLE signal in an OR gate:

- When inverted ENABLE is "0", the OR gate output corresponds to the sensor's data.
- When inverted ENABLE is "1", the OR gate output is always "1" (the sensor stops transmitting).

Before sending data, the ACK signal must be transmitted and encoded. This is done using an OR gate with negated ACK and the clock signal as inputs:

- If ACK is "0", it enters as "1", making the OR output always "1".
- If ACK is "1", it enters as "0", making the OR output equal to the clock signal, generating "10", which represents "1" in Manchester code.

It's important to note that ACK is always sent before the data, because ENABLE only becomes "1" after ACK returns to "0".

The encoded signals are then fed into an AND gate:

- When ACK is "0" and ENABLE is "1", the AND gate allows sensor data to pass through.
- The resulting signal (Dout) resembles busdata from the moment the system becomes active.

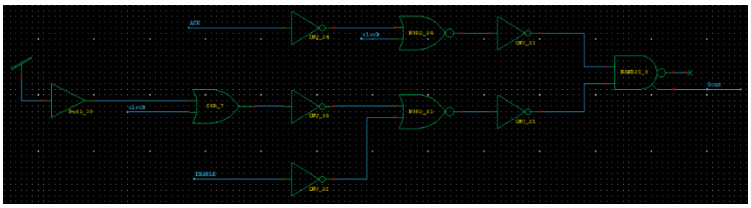


Figure 13 - Circuit that encodes the Manchester code and gives the DOUT signal

Figure 14 – DOUT signal

8. Error Detection

The error detection system is responsible for identifying synchronization issues between the clock and busdata, by comparing these two signals.

- The ACK and ENABLE signals are fed into a NOR gate, which outputs "0" since these two signals are always different.
- The Dout signal is compared with busdata using an XNOR gate. If they are equal (meaning no error), the XNOR outputs "1".
- These two outputs are then fed into another NOR gate, which outputs "0" when there is no error and "1" when Dout does not match busdata (indicating an error).

This error signal is applied to two parallel flip-flops:

- One is triggered on the rising edge of the clock.
- The other is triggered on the falling edge.
- This ensures that errors can be detected at both phases of the clock cycle.

The outputs of these flip-flops are combined in an OR gate:

- If no error is detected, the output is "0".
- If an error is detected, the output is "1".

This ERROR signal is then used in a final OR gate with Dout:

- If ERROR = "0", the sensor's data is transmitted normally.
- If ERROR = "1", the output is forced to "1", preventing corrupted data from appearing on the line.

Additionally, the ERROR signal is used to generate the SET signal:

- If an error occurs, ERROR forces SET to "11", which resets the flip-flops, setting busdata to "1" and terminating the communication.

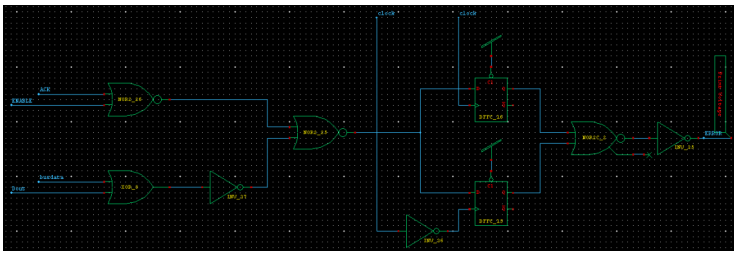


Figure 15 – Circuit that generates the error signal

Figure 16 – Error signal

9. Final circuit

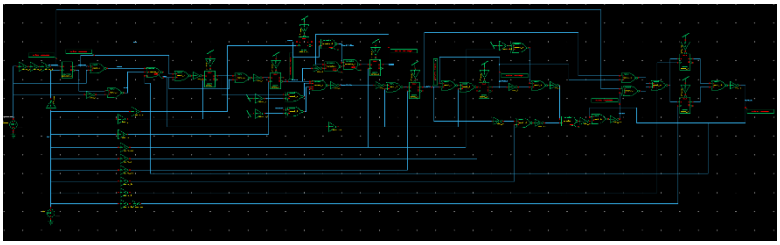


Figure 17 – Final Circuit

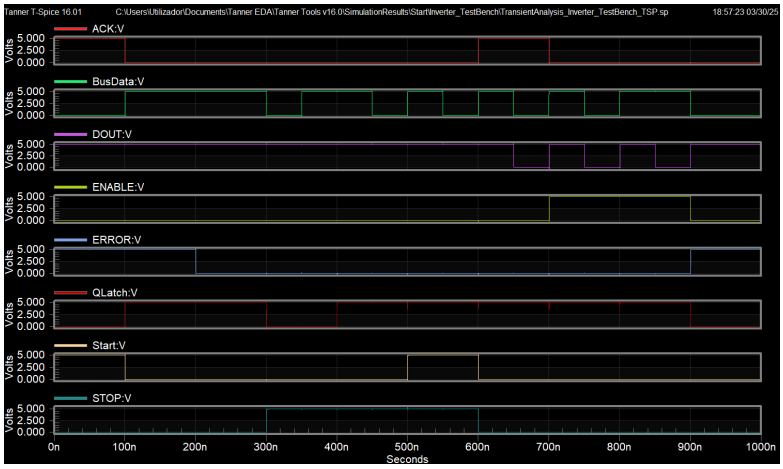
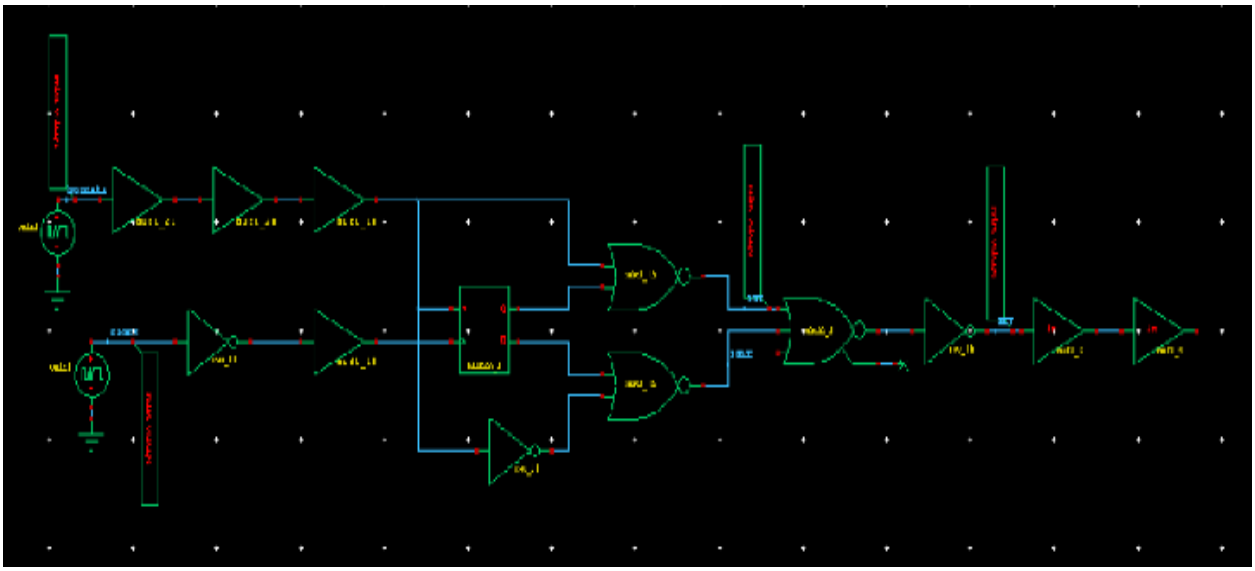
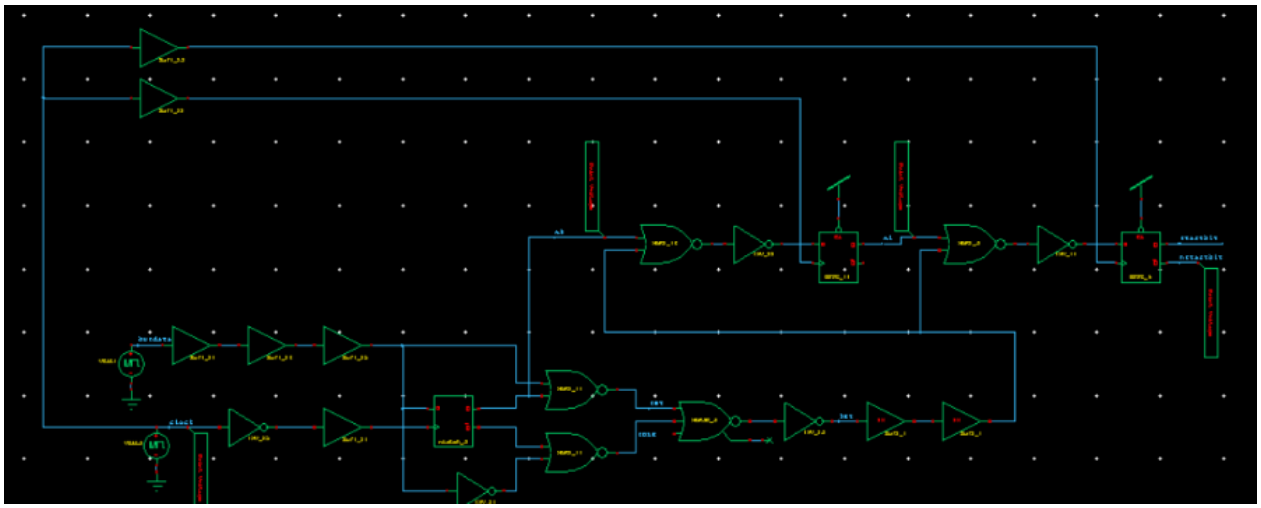


Figure 18 – All signals

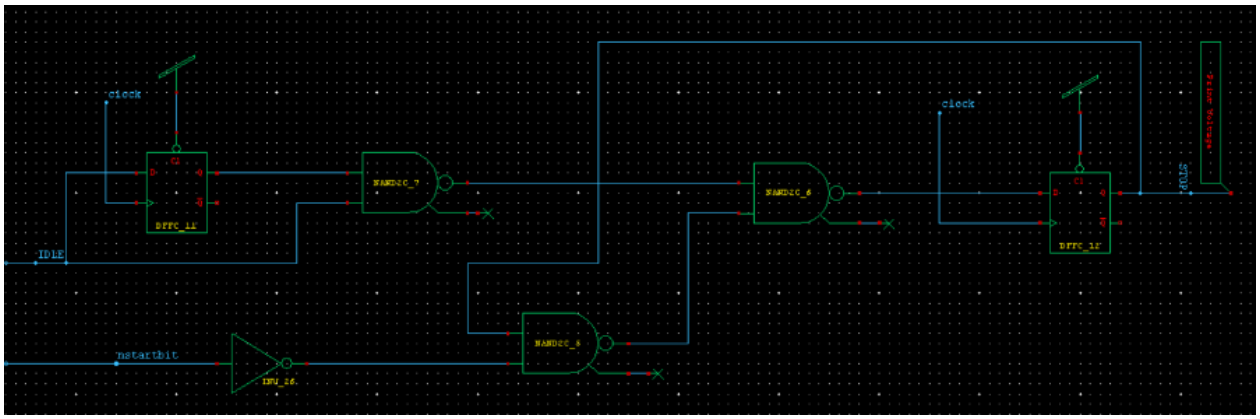
11. Annexes



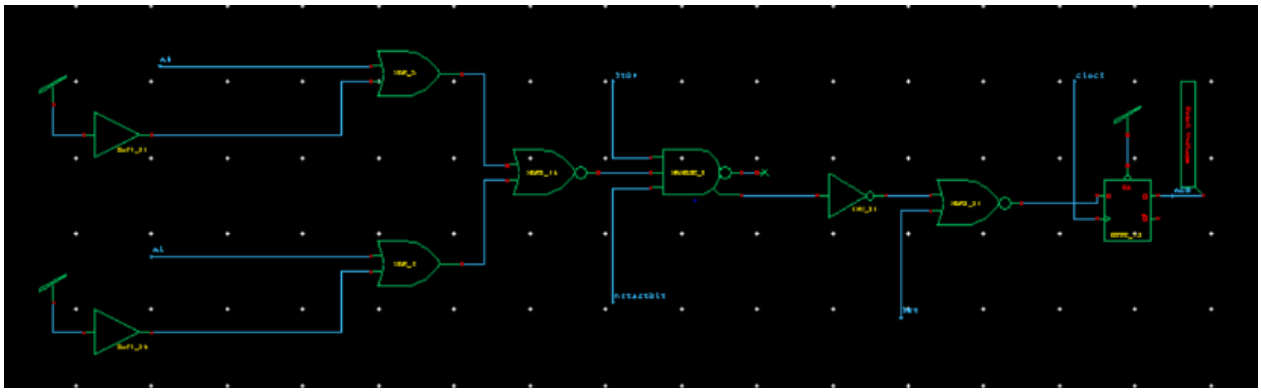
Manchester decoder



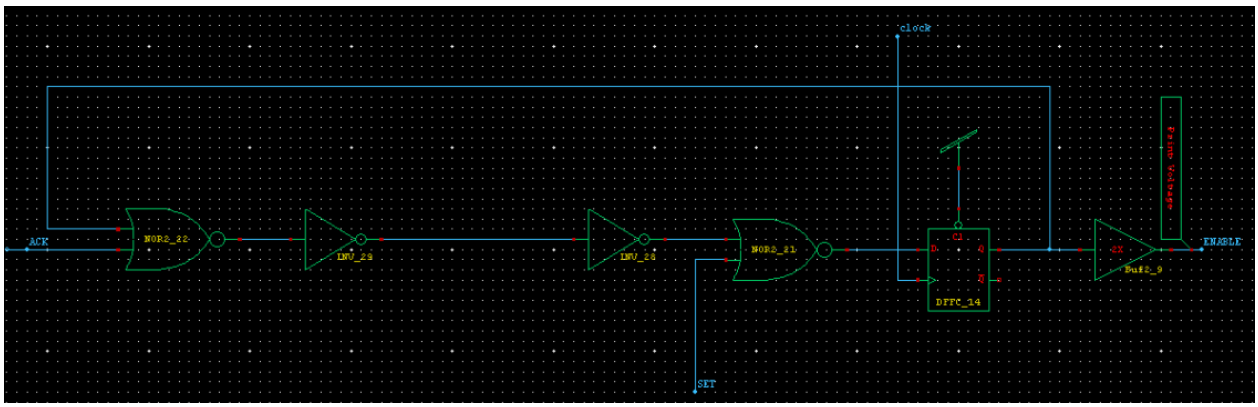
Serial-to-parallel converter



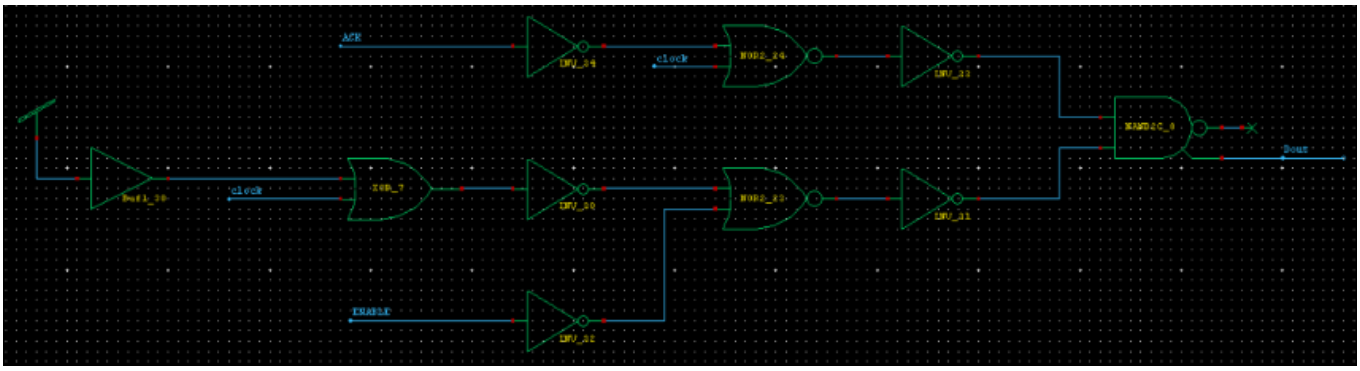
Slave brain



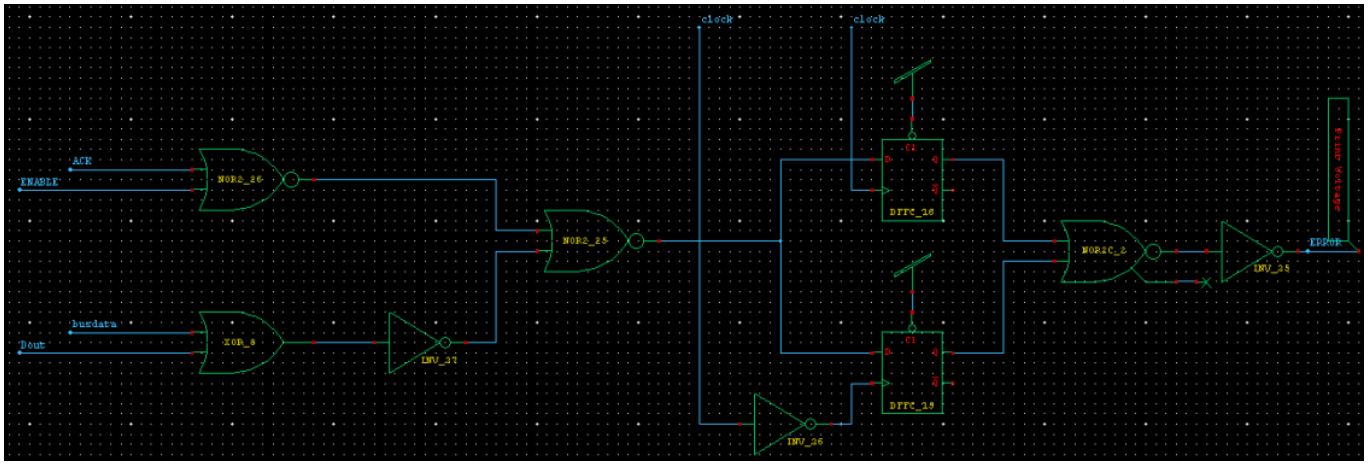
Circuit that sends the ACK bit



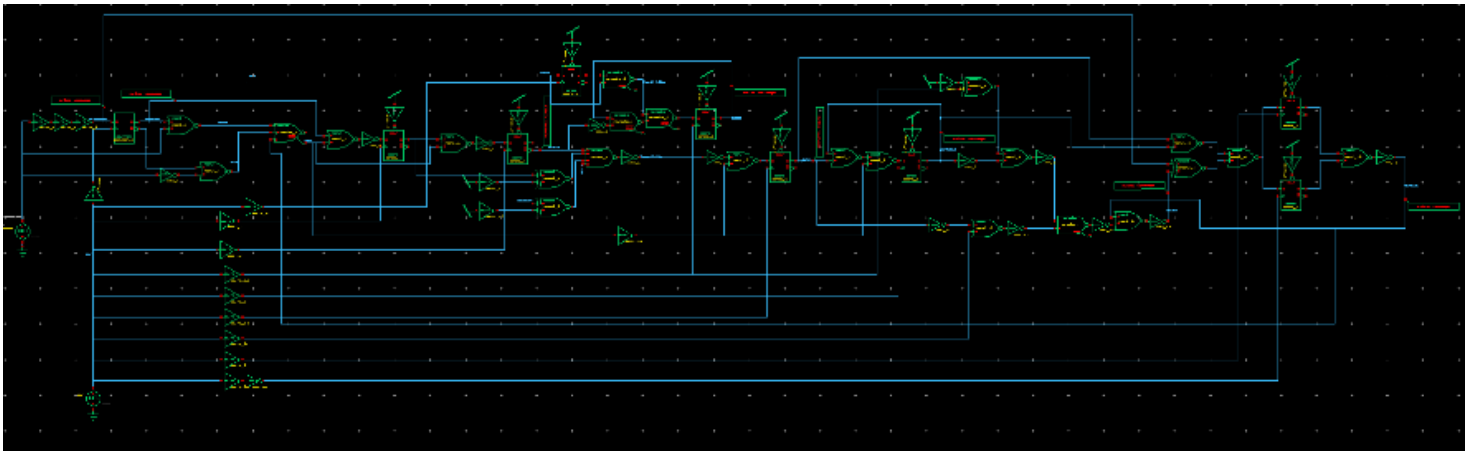
Circuit that sends the enable signal



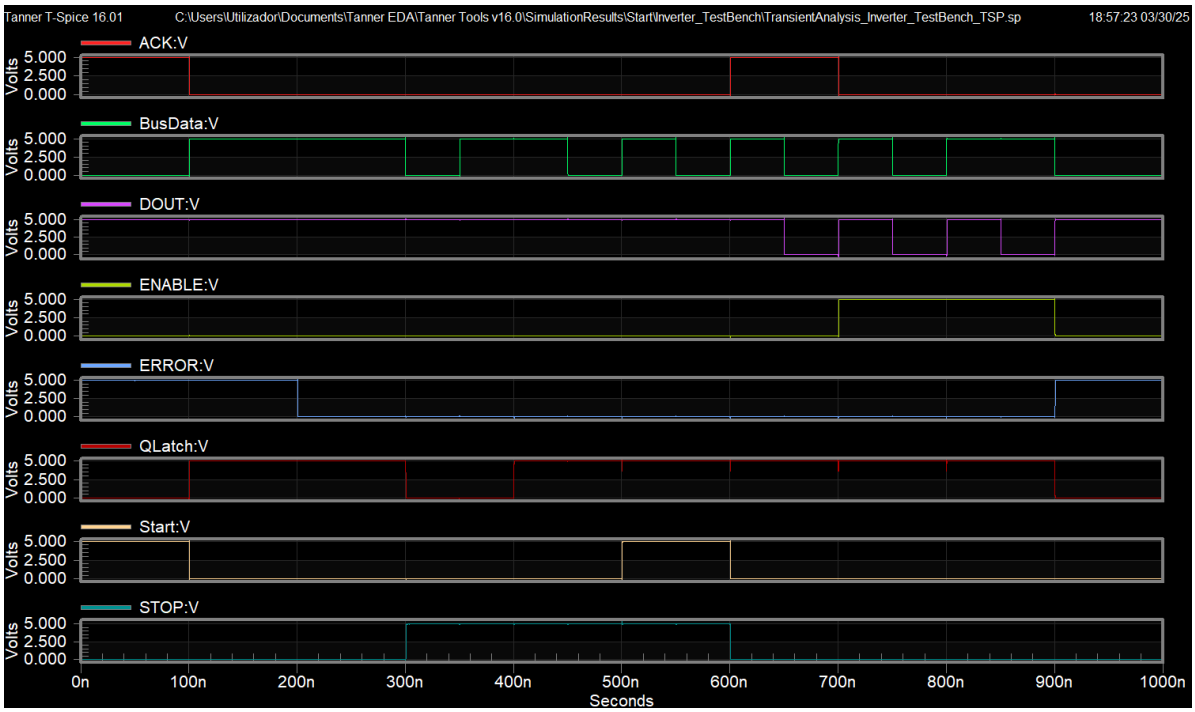
Circuit that encodes the Manchester code



Circuit that generates the error signal



Final circuit



All signals

Bibliography

[1] Work guide.

[2] Notes from Electronics Complements classes.

[3] Notes from Introduction to microtechnologies in silicon classes

[4] Previous support texts