

---

# ASIC "Door Lock"

---

DIGITAL CIRCUIT DESIGN WS 2018/19  
HOCHSCHULE RAVENSBURG - WEINGARTEN

Kostadin Lazarov  
29410

Siyi Dai  
29245

October 30, 2018

# Contents

|          |                                |          |
|----------|--------------------------------|----------|
| <b>1</b> | <b>Document Versions</b>       | <b>3</b> |
| 1.1      | About The Document . . . . .   | 3        |
| 1.2      | Recent Changes . . . . .       | 3        |
| 1.3      | All Changes . . . . .          | 3        |
| <b>2</b> | <b>Functional Requirements</b> | <b>4</b> |
| <b>3</b> | <b>Specifications</b>          | <b>5</b> |
| 3.1      | Block Diagram . . . . .        | 5        |
| 3.2      | Diagram Description . . . . .  | 6        |
| 3.2.1    | UART - Receiver . . . . .      | 6        |
| 3.2.2    | 9600 Baud Generator . . . . .  | 6        |
| 3.2.3    | Parity Check . . . . .         | 6        |
| 3.2.4    | Memory . . . . .               | 6        |
| 3.2.5    | Password Bank . . . . .        | 7        |
| 3.2.6    | Comparator . . . . .           | 7        |
| 3.2.7    | Clock Divider . . . . .        | 7        |
| <b>4</b> | <b>I/O Specifications</b>      | <b>8</b> |
| 4.1      | Inputs . . . . .               | 8        |
| 4.2      | Outputs . . . . .              | 8        |

# 1. Document Versions

## About The Document

The purpose of this document is to elaborate on the design of ASIC "Door Lock" as part of the project for the course Digital Circuit Design. The work which was done during the design of this project is presented here. The reader can become more familiar with the design's functional requirements, specification, definition of I/O etc.

All the necessary changes which will be made during the design period will be stated in the tables below 1.2 and 1.3.

## Recent Changes

The most recent changes that were made in the document are listed below. All the other later changes can be find in All changes 1.3.

| Date       | Name     | Commit   |
|------------|----------|--|
| 27/10/2018 | /        | Release of the document. No changes up to now. |
| 30/10/2018 | Siyi Dai | Implement the functional requirement.          |

## All Changes

| Date       | Name     | Commit                                |
|------------|----------|---------------------------------------|
| 30/10/2018 | Siyi Dai | Implement the functional requirement. |

## 2. Functional Requirements

The functional requirement given below is taken into consideration and adjusted respectively for the design.

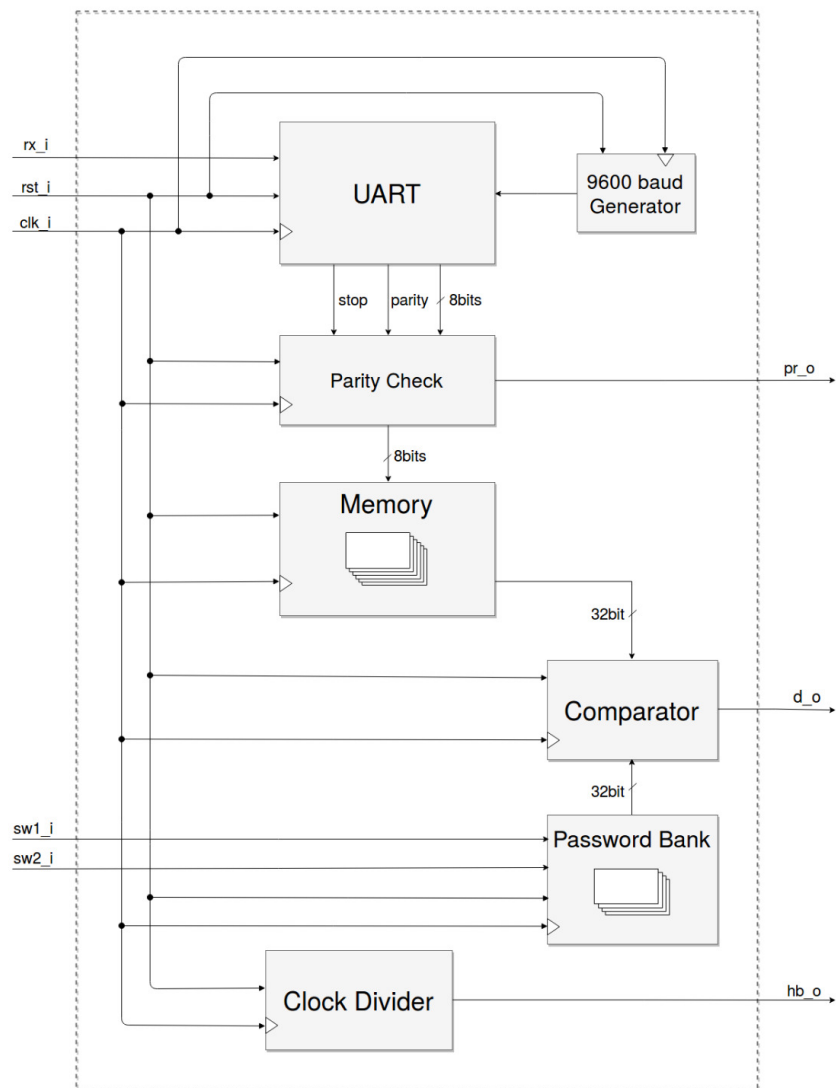
|              |                        |
|--------------|------------------------|
| Project Name | ASIC "Door Lock"       |
| Course       | Digital Circuit Design |
| Student Name | Kostadin Lazarov       |
| Student ID   | 29410                  |

| ID  | Requirement   |
|-----|---|
| R01 | As an input, the device has to receive data stream of 4 ASCII characters from a micro-controller. |
| R02 | The device should be able to receive data at 9600 baud rate.                                      |
| R03 | The device needs to implement/support the receiving part of UART.                                 |
| R04 | The device should be able to receive a package of 8 bits data, a parity bit and a stop bit.       |
| R05 | The device should be able to check for correct parity.  |
| R06 | The device should signalize correct parity by an LED.   |
| R07 | The device needs to have pre-stored characters.   |
| R08 | Each string should consist of four characters (letters), line feed and carriage return.           |
| R09 | The device needs to have four sets of different characters to be chosen.                          |
| R10 | The characters set should be selectable by switches.  |
| R11 | The received characters should be compared to the pre-stored characters.                          |
| R12 | When the received characters are matched (correct), the door opens.                               |
| R13 | The door opens only if the chosen set of characters is matched.                                   |
| R14 | The device should light up an LED when the door opens.  |
| R15 | The device should have a heartbeat LED to indicate the system is running.                         |

## 3. Specifications

In the sections below, there is a block diagram 3.1 of the design and a description of the block diagram with its modules 3.2.

### Block Diagram



## Diagram Description

In the diagram presented above in section 3.1, there are five inputs and three outputs. All of the inputs are single bit inputs as well as the outputs.

Two of the inputs are taken from switches on the board and their purpose is to select a pre-stored password. The reset input is taken from a push-button from the development board, as well as the clock signal which is taken from the board (50 MHz). The last input comes from a programmed micro-controller and brings the transmitted data into the ASIC.

The outputs are realized on LEDs from the development board. There is an output which will signalize correct parity and another to signalize when the door opens. There is also an output which implements a heart-beat signal to show that the system is running.

The ASIC should be able to input the data which is sent from the micro-controller. The UART receiving principal is implemented regarding this matter at 9600 baud rate line. After the data package is received, the device should check for correct parity. If the condition for correct parity is met, then the data bits from the package are stored in "memory". Finally, after the transmission of all characters is finished, the device should compare the transmitted characters with the switch-selected password (pre-stored characters). If there is a match, an LED should light up.

A brief description of each block/module inside the design is given below.

### UART - Receiver

This module gets an input from the micro-controller and the baud generator, as well as a clock and reset input. It receives a single bit data from the micro-controller at 9600 baud rate. However the sampling will not be done exactly on the rising or the falling edge of the transmitted bit, but roughly in the middle. The purpose of this is to improve the accuracy and reduce errors. After the stop bit, the module sends the received package of 8 bits data, parity bit and stop bit in the Parity Check module.

### 9600 Baud Generator

This is the module which generates pulses for maintaining the 9600 baud rate transmission or in other words every 104.2  $\mu$ s a pulse is generated and send to the UART module. It works on the principle of dividing the clock of 50 MHz which is taken as an input.

### Parity Check

At this module the parity of the received package is checked. This is done because we want to check for possible errors in the transmission. For the purpose of this project, an even parity will be used, i.e. the number of logic-high bits in the data package (including the parity) should be even. If the parity is correct the 8 data bits are send to the memory , else and error occurred and the received package is dismissed.

### Memory

This module needs to store the characters which are received from the micro-controller. The memory stores characters that only passed the parity check. After the transmission, the characters are packed in a 32 bit vector and send to the Comparator module. The stored data is removed when a system reset is done.

## **Password Bank**

At this module, the device keeps the pre-stored characters/passwords. The characters encoding is ASCII. Each password can be selected by two switches which are the inputs of this module. The selected password is packed in a 32 bit vector and send to the Comparator module.

## **Comparator**

This is the module which receives packages of 32 bits from the Memory and from the Password Bank. Its duty is to check/compare both of those packages. If they have the same content, the Comparator sends an output which signalizes that the password is correct, i.e. the door opens. If the password is not correct the LED will not light up.

## **Clock Divider**

This module is used to create a pulse at a required frequency. It divides the system clock up to the particular frequency which should be generated. The pulse will be send as an output to an LED to implement a heart-beat signal on the board. The purpose of this heart-beat signal is to show that the system is running.

## 4. I/O Specifications

All the I/Os which are used in the design are declared and specified here. It is also given a short description for each I/O as well as the board pin which is used.

### Inputs

| Name  | Bit(s) | Pin | Description   |
|-------|--------|-----|---|
| rx_i  | 1      | /   | This input brings the data which is sent from the micro-controller to the ASIC.   |
| clk_i | 1      | /   | This is the clock that is imported from the board which is used for realizing the design. The imported clock from the board is 50 MHz.                        |
| rst_i | 1      | /   | This is the reset of the system. The purpose is to provide a reset state to all the working components. This signal is taken from a push-button on the board. |
| sw1_i | 1      | /   | This is the input taken from switch number 1. The switch is located on the board where the design is realized.  |
| sw2_i | 1      | /   | This is the input taken from switch number 2. The switch is located on the board where the design is realized.  |

### Outputs

| Name | Bit(s) | Pin | Description   |
|------|--------|-----|---|
| d_o  | 1      | /   | This is the output which will turn on/off an LED depending whether the door is open or closed. The LED is located on the board. |
| pr_o | 1      | /   | This output will set an LED on/off depending whether the parity is correct. The LED is located on the board.                    |
| hb_o | 1      | /   | This output implements a heart-beat signal on an LED.   |



# Bibliography

- [1] Wikipedia - UART,  
[https://en.wikipedia.org/wiki/Universal\\_asynchronous\\_receiver-transmitter](https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter)
- [2] Wikipedia - Functional requirements,  
[https://en.wikipedia.org/wiki/Functional\\_requirement](https://en.wikipedia.org/wiki/Functional_requirement)
- [3] Prof. Dr. -Ing. Andreas Sigelkow - Lecture Plan WS 2018/19