Hochschule
Ravensburg-Weingarten
Elektrotechnik und Informatik

ASIC "Door Lock"

Digital Circuit Design WS 2018/19

Hochschule Ravensburg - Weingarten

Kostadin Lazarov
29410

Siyi Dai
29245

January 20, 2019

Contents

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
20/11/2018	Siyi Dai	The comparator part is done.
14/11/2018	Siyi Dai	Finish the parity check.
06/11/2018	Siyi Dai	Basecode of clock divider.
30/10/2018	Siyi Dai	Implement the functional requirement.
27/10/2018	/	Release of the document. No changes up to now.

All Changes

Date	Name	Commit
30/10/2018	Siyi Dai	Implement the functional requirement.
06/11/2018	Siyi Dai	Basecode of clock divider.
14/11/2018	Siyi Dai	Finish the parity check.
20/11/2018	Siyi Dai	The comparator part is done.

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. Implementations

OSI Layer Model

The OSI model (Open Systems Interconnection model), developed by the ISO ((International Organization for Standardization), is a conceptual model that defines a framework for communications with seven abstraction layers.

The aims of the OSI layer model is to separate different parts of communications sub-systems to help with the debugging process, and to move structures from one sub-system to another.

Description of OSI layers

Layer	Functional Description
Application (7)	Refers to interfaces between network and application software. Also includes authentication services.
Presentation (6)	Defines the format and organization of data. Includes encryption.
Session (5)	Establishes and maintains end-to-end bidirectional flows between endpoints. Includes managing transaction flows.
Transport (4)	Provides a variety of services between two host computers, including connection establishment and termination, flow control, error recovery, and segmentation of large data blocks into smaller parts for transmission.
Network (3)	Refers to logical addressing, routing, and path determination.
Data link (2)	Formats data into frames appropriate for transmission onto some physical medium. Defines rules for when the medium can be used. Defines means by which to recognize transmission errors.
Physical (1)	Defines the electrical, optical, cabling, connectors, and procedural details required for transmitting bits, represented as some form of energy passing over a physical medium.

In this architecture, each layer serves the layer above it and, in turn, is served by the layer below it and control is passed from one layer to the next. If there are two parties to a communication session, data generated by each starts at the top layer, undergoing any required configuration and processing through the layers, and is finally delivered to the physical layer for transmission across the physical medium.

In our case, most interfaces that are integrated on microcontrollers only cover the data link layer, with the physical layer being implemented externally.

Implementations of OSI layers

Data link layer

With reference to the OSI model, the UART in our project implements the data link layer (layer 2).

In this layer, the data packets of ASCII characters consist of one start bit, 8 data bits, an even parity bit and one stop bits are transmitted with 9600 baud rate.

The UART data link layer offer the even parity bit in terms of error checking.

Physical layer

The full physical layer (layer 1) of the OSI in our case is implemented in the Spartan-3E FPGA Starter Kit using VHDL codes.

FPGA contains an array of programmable logic blocks, and a hierarchy of reconfigurable interconnects that allow the blocks to be wired together. In most blocks, it includes the memory block. Therefore, the rest tasks in our project are processed in this layer.

The physical layer features include pulse shaper and serial communication.

4. Specifications

In the sections below, there is a block diagram 4.1 of the design and a description of the block diagram with its modules 4.2.

Block Diagram

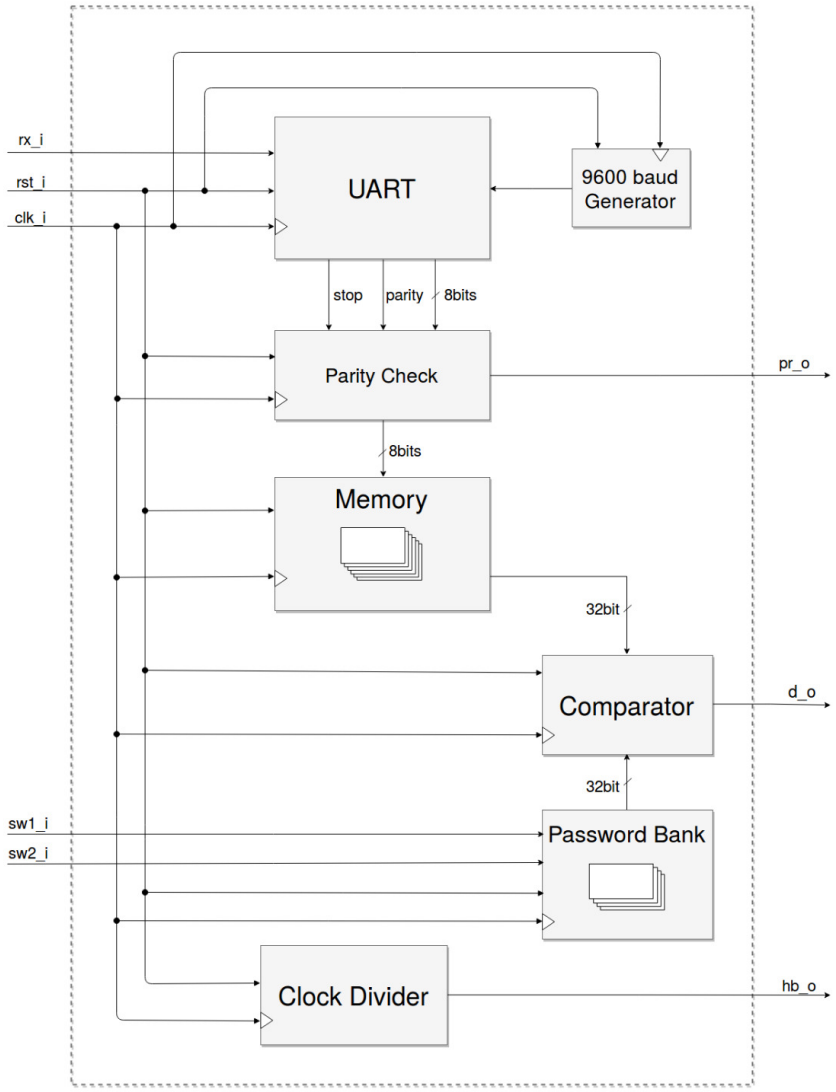

Diagram Description

In the diagram presented above in section 4.1, 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 µs 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.

5. 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 wall as the board pin which is used.

Top-level

Inputs

Name	Bit(s)	Pin	Description
rx_i	1	R7	This input brings the data which is sent from the micro-controller to the ASIC.
clk_i	1	C9	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	N17	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	L14	This is the input taken from switch number 1. The switch is located on the board where the design is realized.
sw2_i	1	H18	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
pas_o	1	F12	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.
l_o	1	E11	This output will set an LED on/off depending whether the parity is correct. The LED is located on the board.
hb_o	1	E12	This output implements a heart-beat signal on an LED.

UART-Receiver

Inputs

Name	Bit(s)	Pin	Description
rx_i	1	R7	This input brings the data which is sent from the micro-controller to the ASIC.
clk_i	1	C9	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	N17	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.
bd_i	1	/	This is the input taken from baud generator with shifted pulse for correct sampling.

Outputs

Name	Bit(s)	Pin	Description
l_o	1	E11	This output will set an LED on/off depending whether the parity is correct. The LED is located on the board.
dat_o	8	/	This is the output which contains the ASCII binary data.
ena_o	1	/	This output will send to enable baud generator.
mem_o	1	/	This output informs register when one password is received.
par_o	1	/	This output indicates the result of parity-check of the received package.
cn_o	1	/	This output announces the status of data input to password comparator.
flg_o	1	/	
s_o	1	/	
ss_o	2	/	

Baud Generator

Inputs

Name	Bit(s)	Pin	Description
clk_i	1	C9	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	N17	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.
ena_i	1	/	This is the input taken from UART-Receiver which activates baud generator.

Outputs

Name	Bit(s)	Pin	Description
bd_o	1	/	This is the onput which gives UART-Receiver shifted pulse for correct sampling.
d_o	1	/	

Register

Inputs

Name	Bit(s)	Pin	Description
clk_i	1	C9	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	N17	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.
par_i	1	/	This input comes from UART-Receiver with the result of parity-check.
mem_i	1	/	This input contains information of whether one password is received from UART-Receiver.
dat_i	8	/	This is the input which contains the ASCII binary data from UART-Receiver.
m_i	1	/	This input taken from password comparator requests password.
fs_i	1	/	

Outputs

Name	Bit(s)	Pin	Description
d_o	8	/	This is the output which contains the ASCII binary data.
g_o	1	/	

Password Comparator

Inputs

Name	Bit(s)	Pin	Description
clk_i	1	C9	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	N17	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.
dat_i	8	/	This is the input which contains the ASCII binary data sent from Register.
tno_i	1	/	This input gets the status of data input from UART-Receiver.
s1_i	1	/	This is the input taken from switch number 1.
s2_i	1	/	This is the input taken from switch number 2.
g_i	1	/	

Outputs

Name	Bit(s)	Pin	Description
m_o	1	/	This output requests password after getting the switch value.
p_o	1	/	This is the output which indicates whether the password is correct or not.

Heart-beat

Inputs

Name	Bit(s)	Pin	Description
clk_i	1	C9	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	N17	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.

Outputs

Name	Bit(s)	Pin	Description
hb_o	1	E12	This output implements a heart-beat signal on an LED.

Bibliography

[1] Wikipedia - UART,
https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter

[2] Wikipedia - Functional requirements,
https://en.wikipedia.org/wiki/Functional_requirement

[3] Wikibooks - Serial programming,
https://upload.wikimedia.org/wikipedia/commons/1/1f/Serial_Programming.pdf

[4] ICND1 topic - OSI layers, http://ptgmedia.pearsoncmg.com/imprint_downloads/informit/learninglabs/9780134213736/ch30.html

[5] Wikipedia - OSI model, https://en.wikipedia.org/wiki/OSI_model

[6] Prof. Dr. -Ing. Andreas Sigelkow - Lecture Plan WS 2018/19