UART

NAME: Nir Rafman, Itay Parag & Moshe Sharabai DATE: 5/10/19

E-mail: [rafman@post.bgu.ac.il](mailto:rafman@post.bgu.ac.il) [itaypa@post.bgu.ac.il](mailto:itaypa@post.bgu.ac.il) [mossh@post.bgu.ac.il](mailto:mossh@post.bgu.ac.il)

# 1 Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | DATE | BY | BRIEF DESCRIPTION OF CHANGES |
| Rev 0 | 22-02-2019 | Nir Rafman | Initial Draft |
|  |  | Itay Parag |  |
|  |  | Moshe Sharabi |  |
|  |  |  |  |

# Protocol

The purpose of the protocol is to implement a **universal asynchronous receiver-transmitter** ( UART ).  
  
The user’s input will be sampled in the **Receiver** module, then passed to the **Control module** & the **Transmitter module** in order to echo the input at the **LCD** and the users  
screen.

**Overview:**

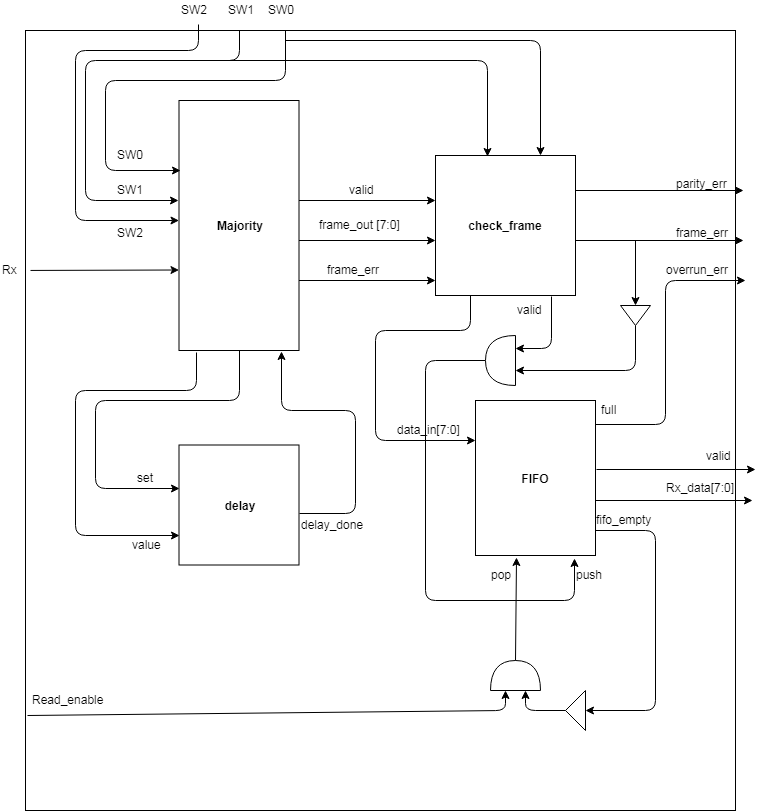


# Receiver

## Overview

The receiver module is responsible to:

* Detecting a “start bit”.
* Deciding on bit value using multiple samples in the middle of the bit transmission time.
* De-serialize to frame format.
* Check the frame.
* Store Incoming data in a FIFO queue.



## Pin in/out

|  |  |  |  |
| --- | --- | --- | --- |
| Pin name | Size in bit | In /out | Description |
| Rx | 1 | In | User input from a UART application |
| SW0 | 1 | In | Even / no Parity selection |
| SW1 | 1 | In | Variable data length (0 for 7 bit length, 1 for 8 bit length) |
| SW2 | 1 | In | Selectable baud rate (0 for 9600b/s, 1 for 115200b/s) |
| Read\_en | 1 | In | The current valid data have been read |
| Clk | 1 | In | Clk |
| rst | 1 | In | Rst |
| Frame\_err | 1 | Out | No stop bit was found. |
| Parity\_err | 1 | Out | A parity error was detected. |
| Overrun\_err | 1 | Out | A data came in and couldn’t be stores as Rx FIFO was full |
| Valid | 1 | Out | Data on d-out is valid |
| Data\_out | 8 | out | 8 bits of data |

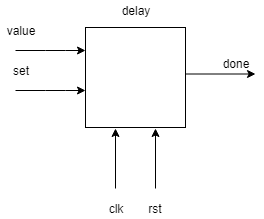
# Sub block’s:

## Delay

### Overview:

This block is designed to count and delay the sampling according to the Baud rate.  
At first the delay will be a bit and a half long, and afterwards a bit long in order   
to sample the bit in the middle.

### Block diagram / method



### Pin in/out

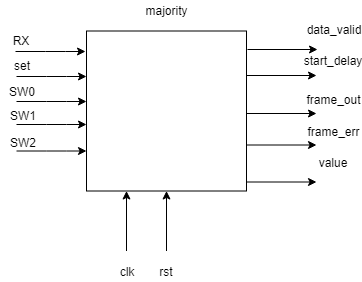
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pin name | Size in bit | In /out | Connect to | Description |
| Enable | 1 | In | Start\_bit module | Start counting |
| Clk | 1 | In | Rx | Clock |
| Rst | 1 | In | Rx | Reset |
| SW2 | 1 | In | Rx | Baud rate 9600\115200 |
| Done | 1 | out | Majority module | The counting finished |

## Majority

### Overview:

This blocks purpose is to sample the received bit in order to decide if it’s 1 or 0.  
The sampling is done with the intervals of the systems clock.  
This block is also checking if a stop bit is recognized.

### Block diagram / method



### Pin in/out

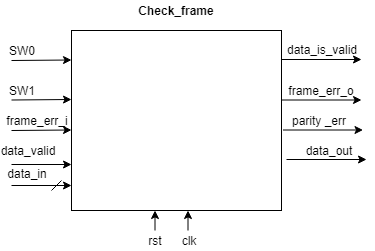
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pin name | Size in bit | In /out | Connect to | Description |
| Enable | 1 | In | Delay module | Star the sampling  process |
| Rx | 1 | In | Rx | The received bits |
| Clk | 1 | In | Rx | Clock |
| Rst | 1 | In | Rx | Reset |
| SW0 | 1 | In | Rx | Even / no Parity selection |
| SW1 |  | in | Rx | Variable data length |
| Done | 1 | out | Start bit module | Telling the start module to sample again |
| Frame\_err | 1 | out | Check\_frame module | Returns 1 if no stop bit is detected |
| Data bits | 8/9/10 | out | Check\_frame module | The actual data bits |

## Check\_frame

### Overview:

This block checks if our information is good, it checks the parity bit value, if assigned.

### Block diagram / method



### Pin in/out

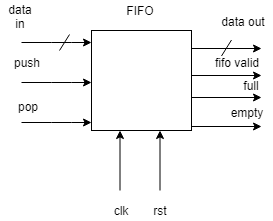
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pin name | Size in bit | In /out | Connect to | Description |
| Frame\_err | 1 | In | Majority module | Calculated before at the Majority module |
| Clk | 1 | In | Rx | Clock |
| Rst | 1 | In | Rx | Reset |
| Data | 7/8 | In | Majority module | The whole dicapsulated frame |
| Frame\_err | 1 | out | Rx & Queue FIFO | Returns 1 if no stop bit is detected |
| Parity\_err | 1 | out | Rx & Queue FIFO | Returns 1 if the parity doesn’t match |
| Data\_out | 7/8 | out | Queue FIFO | The real data frame |

## Queue FIFO

### Overview:

This block is designed to be a FIFO queue to store the 7/8 bit data that arrives.  
The block is connected directed to the controller and sends/receives data from/to him.

### Block diagram / method



### Pin in/out

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pin name | Size in bit | In /out | Connect to | Description |
| push | 1 | In | Check\_frame module | This bit tells us that we have no errors and we can push the data into the queue |
| Clk | 1 | In | Rx | Clock |
| Rst | 1 | In | Rx | Reset |
| Data\_in | 7/8 | In | Check\_frame module | The whole dicapsulated and checked frame |
| pop | 1 | in | Controller | This bit indicates that data have been read |
| full | 1 | out | Rx | This bit indicates that the queue is full |
| empty | 1 | out | Controller | This bit indicates that the queue is empty |
| Data\_out | 7/8 | out | controller | The real data frame that goes to the controller |

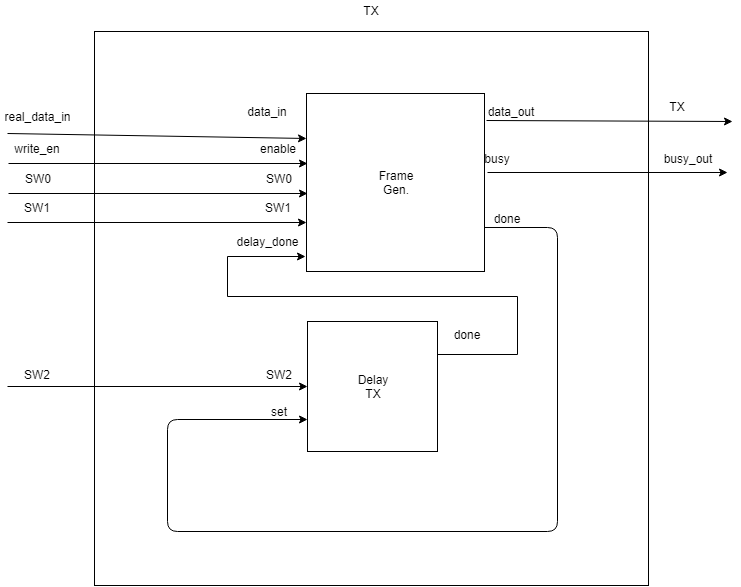
# Transmitter

## Overview

The transmitter module is responsible to:

* Read outgoing data.
* Encapsulate data into a frame.
* Serializing data at required rate.

## Block diagram



## Pin in/out

|  |  |  |  |
| --- | --- | --- | --- |
| Pin name | Size in bit | In /out | Description |
| Data\_in | 8 | In | 8 bits to transmit |
| Write\_en | 1 | In | Data on din is valid. This signal should be raised for a single clock only. |
| Clk | 1 | In | clk |
| Rst | 1 | In | Rst |
| SW0 | 1 | In | Even / no Parity selection |
| SW1 | 1 | In | Variable data length (0 for 7 bit length, 1 for 8 bit length) |
| SW2 | 1 | In | Selectable baud rate (0 for 9600b/s, 1 for 115200b/s) |
| busy | 1 | Out | Can’t except any more data for the transmission, TX FIFO is full. |
| Tx | 1 | Out | The bit to transmit to the user |

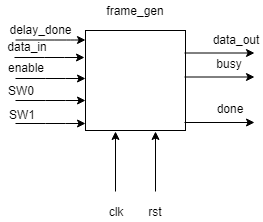
# Sub block’s:

## Frame\_gen

### Overview:

This block’s purpose is to generate the frame according to the switch properties,  
It will generate the parity bit according the incoming data and will add start and stop bit.

### Block diagram / method



### Pin in/out

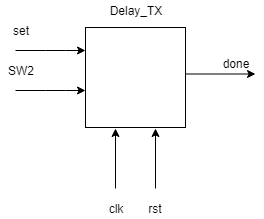
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pin name | Size in bit | In /out | Connect to | Description |
| enable | 1 | In | Tx FIFO | This bit tells us that we can start the generation process |
| Clk | 1 | In | Tx | Clock |
| Rst | 1 | In | Tx | Reset |
| Data\_in | 7/8 | In | controller | The whole dicapsulated data |
| SW0 | 1 | in | Tx | Even / no Parity selection |
| SW1 | 1 | Out | Tx | Variable data length ( 7 or 8) |
| done | 1 | Out | delay | Indicates that the generation process is finished |
| Data\_out | 7/8 | out | Sender (mux) | The data bits with all overhead |

## Delay

### Overview:

This block is designed to count and delay the transmission of each bit in our frame  
according to our pre-defined baud rate.

### Block diagram / method



### Pin in/out

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pin name | Size in bit | In /out | Connect to | Description |
| Enable | 1 | In | Frame\_gen | Start counting |
| Clk | 1 | In | Tx | Clock |
| Rst | 1 | In | Tx | Reset |
| SW2 | 1 | In | Tx | Baud rate 9600\115200 |
| Done | 1 | out | Sender (mux) | The counting finished |

# Control

## Overview

The control module is responsible to:

* Initiate the LCD.
* Receive the 7/8 bit frame.
* Check the incoming data for special characters.
* Transmit the data to the LCD and to the Tx.

## Block diagram

## Pin in/out

|  |  |  |  |
| --- | --- | --- | --- |
| Pin name | Size in bit | In /out | Description |
| Data\_in | 8 | In | 8 bits to transmit |
| valid | 1 | In | Data\_in is valid |
| Clk | 1 | In | clk |
| Rst | 1 | In | Rst |
| busy | 1 | In | Cant expect any more data for transmission |
| Frame\_err | 1 | In | No stop bit was found |
| Overrun\_err | 1 | In | Rx fifo is full |
| Parity\_err | 1 | in | A parity error was detected |
| LCD\_DB | 8 | Out | The data for the LCD |
| RS | 1 | Out | LCD command |
| RW | 1 | Out | LCD command |
| Enable | 1 | Out | LCD enable |
| LED\_1 | 1 | Out | Turns on the LED |
| LED\_2 | 1 | Out | Turns on the LED |
| LED\_3 | 1 | Out | Turns on the LED |
| D\_out | 8 | Out | The data to the Tx |
| Write\_en | 1 | Out | Data for Tx is valid |
| Read\_en | 1 | Out | It is possible to read new data from the receiver |

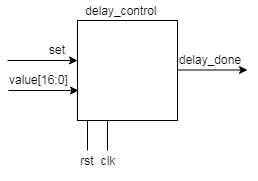
# Sub block’s:

## Delay

### Overview:

This blocks purpose is to count the intervals between each write to the LCD.  
He starts to count at reset and works until we close the program.  
Furthermore, this block signal to the enable and initialization block to start.

### Block diagram / method



### Pin in/out

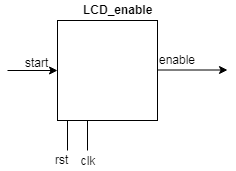
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pin name | Size in bit | In /out | Connect to | Description |
| set | 1 | In | Delay\_module | This bit tells us to start the counting again |
| Clk | 1 | In | Control | Clock |
| Rst | 1 | In | Control | Reset |
| Delay\_done | 1 | output | Mem\_init, enable and delay module | This bit indicates that the counting is over and we can write to the LCD now. |

## Enable

### Overview:

This block is designed to enable the writing process to the LCD.  
It’s outputting single bit data every time the delay module signals him to start his process.  
The enable process takes about 14 clocks.

### Block diagram / method



### Pin in/out

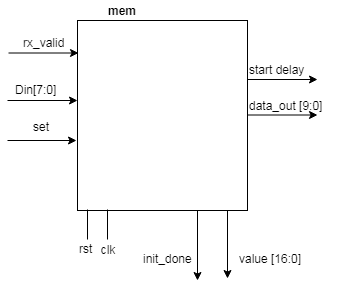
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pin name | Size in bit | In /out | Connect to | Description |
| set | 1 | In | Delay module | This bit tells us that we can start the enable process |
| Clk | 1 | In | Control | Clock |
| Rst | 1 | In | Control | Reset |
| Set\_2 | 1 | out | Checker  module | After 5 times, we would like to get acknowledgment that new data from Rx is arrived and its valid |
| Enable\_done | 1 | out | LCD | Enable the LCD writing |

## Mem

### Overview:

This block is the one that transfer the appropriate data and command to the LCD.  
At first, he transfers the initialization commands and then the received data from the user’s input.

### Block diagram / method



### Pin in/out

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pin name | Size in bit | In /out | Connect to | Description |
| Init\_done | 1 | In | Mem\_init module | This bit tells that the initialization process is over |
| Clk | 1 | In | Control | Clock |
| Rst | 1 | In | Control | Reset |
| Din | 8 | in | Checker module | The received unpacked frame after the check |
| DB\_init | 8 | in | Mem\_init | Initialization data |
| RS\_init | 1 | in | Mem\_init | Initialization command |
| RW\_init | 1 | in | Mem\_init | Initialization command |
| Mem\_done | 1 | out | Control | Indicates that the writing to the LCD is done |
| DB | 8 | out | LCD | The data |
| RS | 1 | out | LCD | LCD command |
| RW | 1 | out | LCD | LCD command |