|  |
| --- |
| Chip Specification- Tramelblaze |
|  |
| MAY 10th  CECS 460  By: Aishwarya Ravishankar |

Table of Contents

1. Introduction ..................................................................................................................................................7

1.1 Purpose .........................................................................................................................................7

2. Applicable Documents................................................................................................................................. 7

2.1 Architecture (Tramelblaze) .......................................................................................................... 7

2.1.1 Memory.......................................................................................................................................7

2.1.2 Architecture .........................................................................................................................……8

2.1.3 I/0 Signals of PicoBlaze, TramelBlaze..................................................................................... 8

2.1.4 PicoBlaze Timing....................................................................................................................... 9

2.1.5 Instruction Set........................................................................................................................….9

2.2 ASCII Table ....................................................................................................................................10

2.3 Baud Rate Decoder ..................................................................................................................... 11

3. Requirement .................................................................................................................................................12

3.1 Hardware Interface Requirement................................................................................................ 12

3.2 Software Interface Requirement ................................................................................................. 13

4. Top Level Design ..........................................................................................................................................14

4.1 Description.....................................................................................................................................13

4.2 Top Level Block..............................................................................................................................14

4.3 Top Level Schematic..................................................................................................................... 15

4.4 I/O Signals and Connections ....................................................................................................... 15

4.4.1 Signal Name .............................................................................................................................. 15

4.4.2 Clock............................................................................................................................................15

4.4.3 Reset ...........................................................................................................................................15

4.5 I/O Description...............................................................................................................................16

4.6 Reset ..............................................................................................................................................16

4.7 Software Description ................................................................................................................... 16

5. Externally Developed Blocks. ......................................................................................................................16

5.1 TramelBlaze...................................................................................................................................16

5.1.1 Introduction ............................................................................................................................... 16

5.1.2 Block Diagram ........................................................................................................................... 17 5.1.3 TramelBlaze I/0 ........................................................................................................................ 17

5.1.4 TramelBlaze I/0 Descriptions................................................................................................... 17

6. Internal Developed Blocks .......................................................................................................................... 18

6.1 AISO.................................................................................................................................................18

6.1.1 AISO Description ....................................................................................................................... 18

6.1.2 AISO Block Diagram................................................................................................................... 18

6.1.3 AISO I/O.......................................................................................................................................18

6.2 Baud Decoder.................................................................................................................................18

6.2.1 Baud Decoder Description ....................................................................................................... 18

6.2.2 Baud Decoder Block Diagram................................................................................................... 19

6.2.3 Baud Decoder I/O....................................................................................................................... 19

6.3 PED Positive Edge Detector .......................................................................................................... 19

6.3.1 PED Positive Edge Detector Description .................................................................................. 19

6.3.2 PED Positive Edge Detector Block Diagram ............................................................................. 19

6.3.3 PED Positive Edge Detector I/O ............................................................................................... 19

Table 9: PED I/O ......................................................................................................................20

6.4 RS FLOP ..........................................................................................................................................20

6.4.1 RS FLOP Description.................................................................................................................. 20

6.4.2 RS FLOP Block Diagram ............................................................................................................ 20

6.4.3 RS FLOP I/O ................................................................................................................................20

6.5 TX Transmit Engine ....................................................................................................................... 20

6.5.1 TX Transmit Engine Description ................................................................................................ 20

6.5.2 TX Transmit Engine Diagram...................................................................................................... 21

6.5.3 TX Transmit Engine I/O .............................................................................................................. 21

6.5.4 TX Transmit Engine Internal composition................................................................................. 21

6.6 RX Receive Engine ........................................................................................................................ 24

6.6.1 RX Engine Description .............................................................................................................. 24

6.6.2 RX Engine Block Diagram .................................................................................................... 24

6.6.3 RX Engine I/O ....................................................................................................................... 24

6.6.4 RX Engine Internal Composition .......................................................................................... 25

6.6.6 RX Engine Schematics ......................................................................................................... 27

6.7 UART...........................................................................................................................................27

6.7.1 Description............................................................................................................................ 27

6.7.2 Block Diagram ...................................................................................................................... 28

6.7.3 UART I/O ............................................................................................................................... 28

6.7.4 UART Schematic ....................................................................................................................29

6.8 TSI – Technology Specific Instantiation.................................................................................... 29

6.8.1 Description.............................................................................................................................. 29

6.8.2 Block Diagram ........................................................................................................................ 30

7 Chip Level Verification .................................................................................................................................31

8. Chip Level Test……………………………………………………………………………………………………………………….........34

Table of Figures

Figure 1: PicoBlaze Detailed Block Diagram.................................................................................... 8

Figure 2: PicoBlaze I/O Timing Diagram............................................................................................ 9

Figure 3: Sample of Instruction Operations – PicoBlaze.................................................................. 10

Figure 4: Updated ASCII Table............................................................................................................ 11

Figure 5: BAUD Rate Chart................................................................................................................. 12

Figure 6: Top Level Block................................................................................................................... 14

Figure 7: Top Level Schematic.......................................................................................................... 15

Figure 8: TramelBlaze Block Diagram .............................................................................................. 17

Figure 9: TramelBlaze Detailed Block Diagram................................................................................ 17

Figure 10: AISO .................................................................................................................................. 18

Figure 11: Baud Decoder................................................................................................................... 19

Figure 12: PED.................................................................................................................................... 19

Figure 13: RS Flop ............................................................................................................................. 20

Figure 14: Transmit Engine ............................................................................................................... 21

Figure 15:TX SCHEMATIC .................................................................................................................. 23

Figure 16:TX RTL Schematic ............................................................................................................. 23

Figure 17: RX Block Diagram............................................................................................................. 24

Figure 18: RX Engine FSM ................................................................................................................ 26

Figure 19: RX Engine Schematics .................................................................................................... 27

Figure 20: UART Block Diagram........................................................................................................ 28

Figure 21: UART Schematic ............................................................................................................. 29

Figure 22: TSI.....................................................................................................................................30

Table of Tables

Table 1: PicoBlaze I/O ...................................................................................................................... 8

Table 2: Output Switches.................................................................................................................. 12

Table 3: EIGHT PEN OHEL Truth Table ............................................................................................ 13

Table 4: TOP Level I/O....................................................................................................................... 16

Table 5: TramelBlaze I/O................................................................................................................... 17

Table 6: TramelBlaze I/O Descriptions............................................................................................ 18

Table 7: AISO I/O............................................................................................................................... 18

Table 8: Address Decoder I/O.......................................................................................................... 19

Table 9: PED I/O................................................................................................................................ 20

Table 10: RS Flop I/O........................................................................................................................ 20

Table 11:TX ENGINE I/O.................................................................................................................... 21

Table 12: RX I/O................................................................................................................................. 25

Table 13: UART I/0..............................................................................................................................30

1. Introduction

Chip Specification, this document is created to document all existence information for the

full UART engine. To the core of this engine, it can break down into many smaller

functional blocks and supporting modules; Two main blocks that are the Receiving

Engine and Transmit Engine (RX | TX); they define the working purpose of the whole

UART engine. The communication can be observed through an UART based terminal on

any supported devices, with proper connection; at a configurable baud rate, the TX is

capable to the terminal total of 11 bits serially.

The data frame can be set using 4 main components: BAUD, EIGHT, PEN, OHEL.

EIGHT is to control either 7 or 8 bits of the data frame; Parity enable can be set with

PEN bit; lastly OHEL bit to determine the given bit is odd high or even low parity bit; the

baud rate can be determined with BAUD pin, which is a 4-bit wide component that can

choose a pre-defined transfer rate. All of 4 components can be collected through a

register TRANSMIT\_ENGINETL, the data frame supposes to transfer serially when the

BTU bit at logic one until the end of frame. The next data frame will follow to transmit

by indicating the high to low logic of the load.

1.1 Purpose

Chips specification delivers necessary information to future users of this UART engine

unit. Providing in-depth documentation on hardware designed functional blocks, as well

as software requirements. Unit peripherals information will also be provided including

signal I/O pins, switches and LEDs, etc. Most documented parts come with a verification

to ensure the functionality upon scenarios, these tests are found from Table of Figures.

The Rx engine’s purpose is to receive data and relay the information into the processor.

Unlike the Tx engine, the Rx engine has the ability to pass either data, or status flags

back into the processor. The Rx engine waits to see a high to low logic to determine a

start bit. Once the start bit is detected, then the Rx engine waits half a BTU (bit time unit)

to sample the incoming data at the center of the data, to avoid data corruption. The Rx

engine uses an MSB right shift register to collect the data once it’s at the collect data

state. Once the data is collected, it shifts the data to the right depending on the input

signals to correctly adjust the data bits. From there the status flags are determined, and

data is ready to be sent to the processor by asserting the Rxrdy flag.

2. Applicable Documents

2.1 Architecture (Tramelblaze)

2.1.1 Memory

TramelBlaze is a softcore microprocessor that will handle the executing part of the

whole engine. As its predecessor, the PicoBlaze, which is the base of Tramelblaze

and contain all identical functionality with fewer upgrade in memory. The

PicoBlaze was designed by Xilinx, INC, and The TramelBlaze was designed by

Professor John Tramel of Cal State Long Beach. The different is a step up in

memory space, in which Tramelblaze is a 16-bit processor and PicoBlaze is 8-bit

processor.

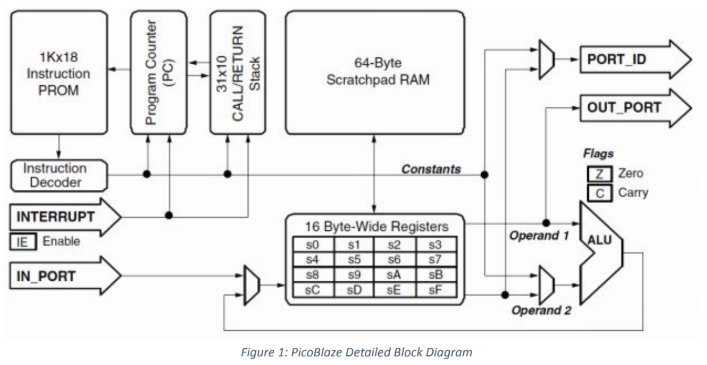
The memory of TramelBlaze can be broken down

Scratch RAM: 512\*16 / Stack RAM: 128 \* 16 / ROM: 4096 \* 16

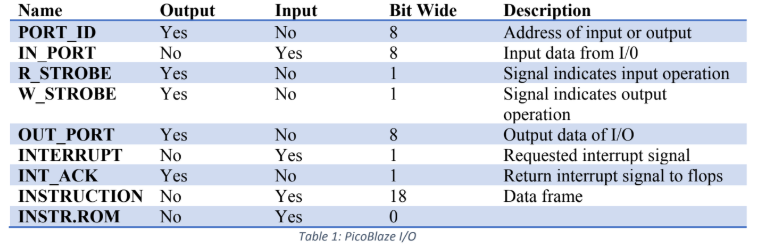
Memory spaces of PicoBlaze:

Scratch RAM: 64\*8 / Stack RAM: 31 \* 10 / ROM: 1024 \* 18

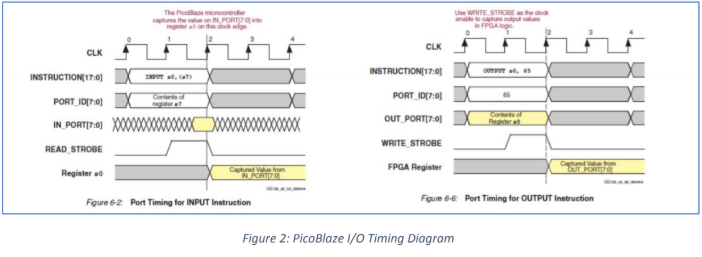
2.1.2 Architecture



2.1.3 I/0 Signals of PicoBlaze, TramelBlaze



2.1.4 PicoBlze Timing



As above, the timing diagram of PicoBlaze in which the instruction and

PORT\_ID are fetched for two clock periods. IN\_PORT reads on the negative

edge of the second period, while OUT\_PORT writes for whole 2 clock period.

Tramelblaze has a slightly different for both IN and OUT PORT, where INTR and PORT\_ID timing stay the same, but IN/OUT PORT will either read or write both on the positive edge of the second clock period.

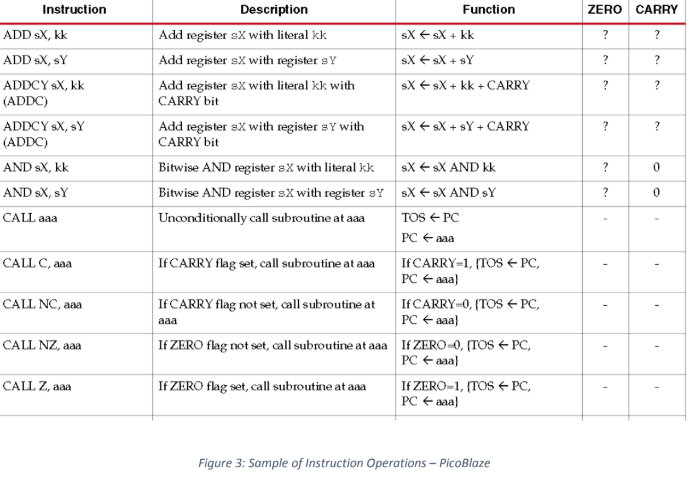
2.1.5 Instruction Set

Since Tramelblaze has PicoBlaze as the base design with several upgrades in

executing memory, they still share the same instructions set. The instruction

set is stored within instruction ROM and can be accessed as a function

(Example of PicoBlaze instructions below).



2.2 ASCII Table

The ASCII TABLE (American Standard Code for Information Interchange) is the

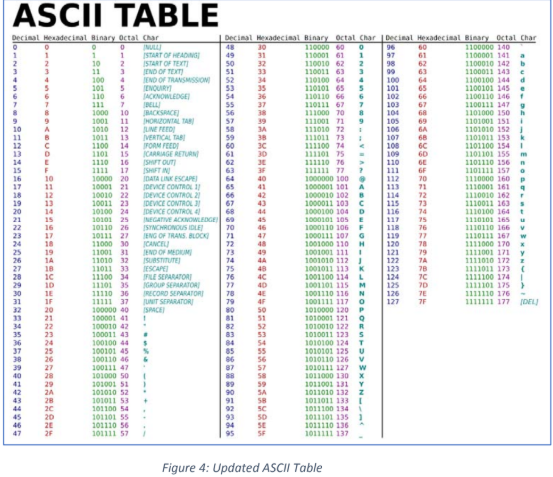
most common format for text files in computers and on the Internet. In an ASCII file,

each alphabetic, numeric, or special character is represented with a 7-

bit binary number (a string of seven 0s or 1s). 128 possible characters are defined.

We are using the ASCII TABLE to create our TBA assembly file; the alphabet letter

that be printed out on device’s terminal as a form of ASCII characters.



2.3 Baud Rate Decoder

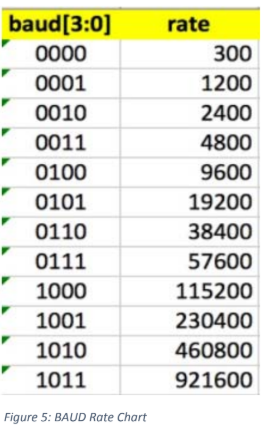
BAUD Rate is the speed of transferring bits of UART devices, it defines how bits can

be on the wire per second. Within UART engine, a BAUD rate decoder is used to

generate delays for Bit Time Unit. BAUD rate is necessary to build UART related

design, a mismatch rate between Rx and Tx can lead to data frame corruption by

sending bit as inequivalent speed.



3. Requirement

3.1 Hardware Interface Requirement

Xilinx FGPA Nexys 4 is used as main operating hardware platform. All the I/O pins

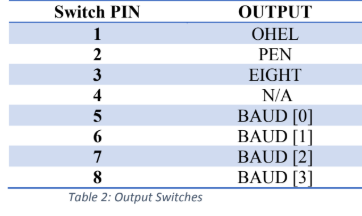
are connected to the software using a UCF type file, each pin locations are listed in

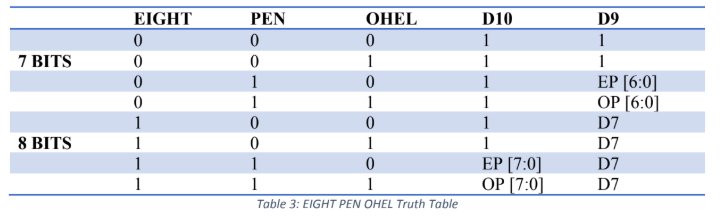
the register, by simply linking them to correct output pins of the engine. A set of leds

are used to display serial bit transfer, or as an internal counter as the engine executes

through an instruction. Also, the internal counter is displayed on the on Real Term or

HyperTerminal where it gets instructions from the assembly file.



3.2 Software Interface Requirement

According to the truth table, the 1st column Eight (MSB) bit explains that if this zero

we are going to transmit or receive 7 bits and when the Eight bit is one there is 8 bits.

The next column is PEN (parity enable), if PEN is zero, parity is disable that means

we are only sending or receiving the seven data bits. In this case, the next two bits

(D10 & D9) is always a one. It will turn in as a stop bit. But, (PEN) parity is enabled,

the data will be from bit zero to six and the last bit will be the parity if we are using

raw three or four of the encoder truth table. Then, the parity will be generated across

bit zero to six. To generate parity for all eight bits then the parity should be generated

across bit seven to zero, which raw seven and eight of the encoder truth table. To

generate an even parity in Verilog we use ^D and ~D for odd parity. It will depend on

the selection between even and odd parity it is going to be one of the two values zero

or one. The third column, OHEL (odd high even low) is the LSB will control whether

when parity is enable if the OHEL is low (0) it is even parity or if the OHEL is high

(1), it is odd parity. Thus, the above table operates under combination logic.

4. Top Level Design

4.1 Description

Top level is where everything is contained. All inputs and outputs are defined here

with correct amount of bit length. Top module includes 8 sub modules: AISO, PED,

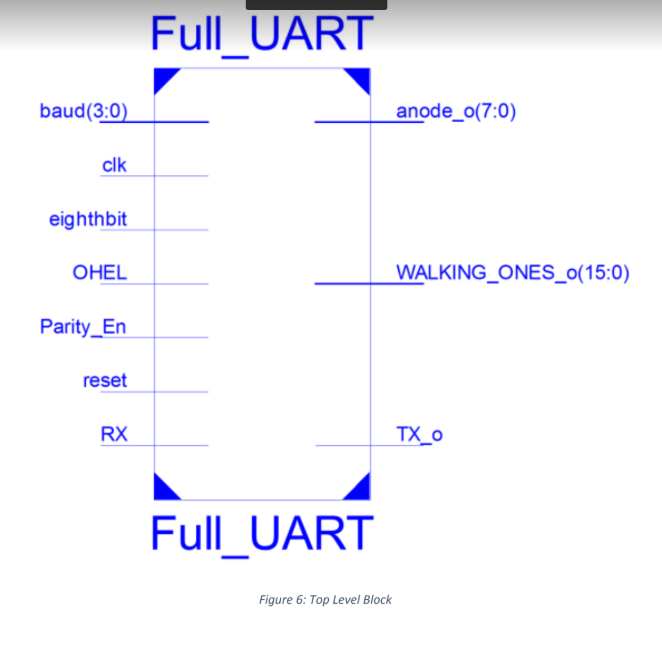
RS FLOP, RX-TX FLOP, DECODER, TRAMELBLAZE, LED, and UART

ENGINE. Addition to major functional blocks that was pre-built, there is also a TSI

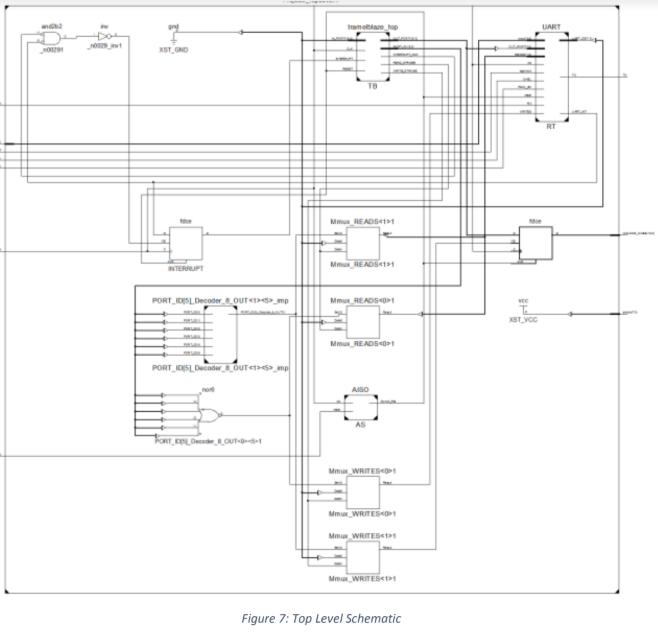
block, in which was made of the target technology library, this block help controlling

the signals that coming in and out to the device.

4.2 Top Level Block



4.3 Top Level Schematic



4.4 I/O Signals and Connections

4.4.1 Signal Name

Inputs: clk, reset, RX, eighth bit, parity\_en, OHEL, baud.

Outputs: TX\_o, walking ones\_o, anode\_o.

4.4.2 Clock

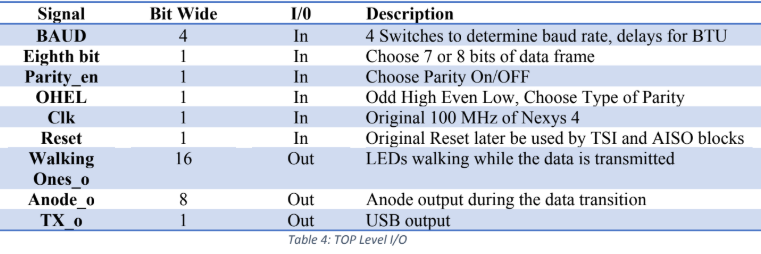
Base clock is being used.

4.4.3 Reset

Reset signal is connected to the AISO, this is to help synchronize the release

of reset, this will prevent metastability, the new reset signal will then be sent into lower modules that need to reset.

4.5 I/O Description



4.6 Reset

Reset signal will bring all the designs to zero or back to default initialize value. The

system will acknowledge the reset signal turned high regardless on any specific

edges.

4.7 Software Description

When Write [1] outputting, the onboard LED is shifted to indicate the internal

counter. The demonstration on the terminal consists of outputting a banner “C:\ for

example”, a prompt from assembly codes; By the acknowledging the RXREADY bit

(when there is an input of data frame), the engine starts the Rx operation. Once both

are received by the Receiver, the engine re transmit the frame to the device terminal

through UART USB connection. A counter is used to keep tracks the number of

character input by user and displays it to user when they input character “@”. When

user input character “\*”, the engine will display a pre-defined phrase.

5. Externally Developed Blocks.

5.1 TramelBlaze

5.1.1 Introduction

Tramelblaze is a 16-bit softcore embedded microprocessor which is to

emulate 8-bit PicoBlaze by Xilinx. Having the same instruction set with

PicoBlaze, the Tramelblaze also have a slightly different timing on read

and write strobe. The memory of Tramelblaze is also changed due to the

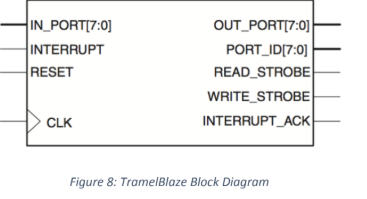
larger in size of the processor.

- Scratch RAM - 512\*16

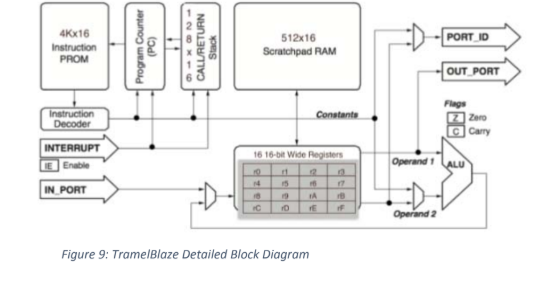
- Stack RAM - 128\*16

- TB ROM - 4096\*16

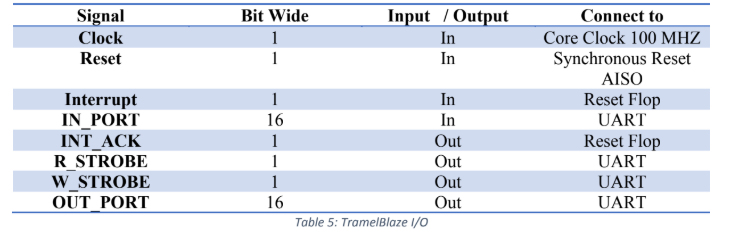
5.1.2 Block Diagram



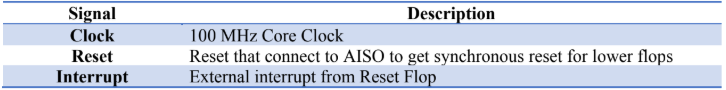
5.1.2.1 TramelBlaze I/O

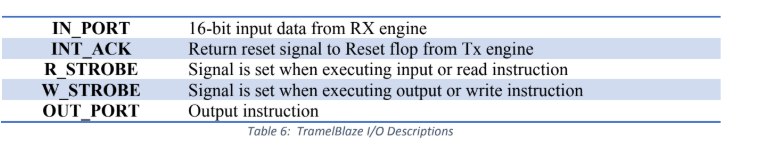


5.1.3 TramelBlaze I/O



5.1.4 TramelBlaze I/O Descriptions





6. Internal Developed Blocks

6.1 AISO

6.1.1 AISO Description

AISO – Asynchronous In, Synchronous Out. For avoiding metastability in

our design, the reset signal must be synchronized when released. The

metastability is when the output of a flop oscillates and end up in an

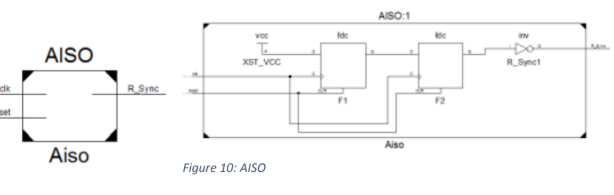
unpredictable state. The AISO takes in original reset and send them

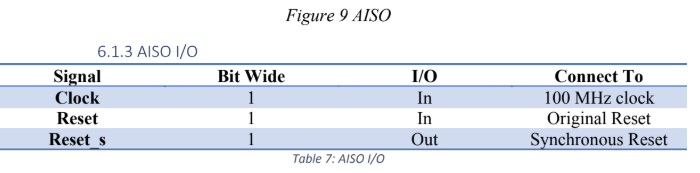
through a set of flops, the output of the second flop will be inverted and

that will be the synchronous reset. This signal will be sent out to lower

flops for future uses.

6.1.2 AISO Block Diagram





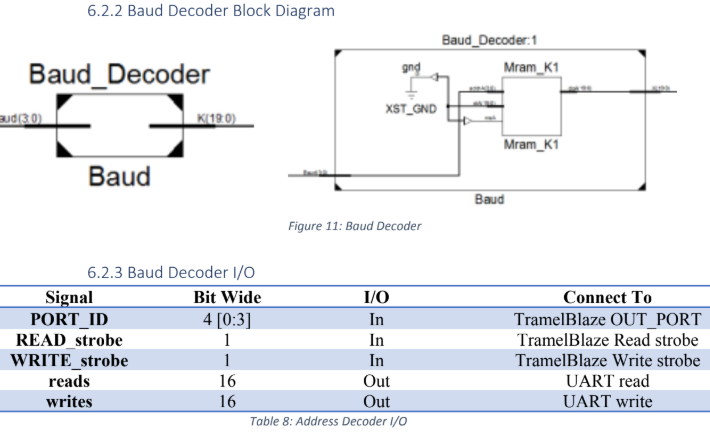
6.2 Baud Decoder

6.2.1 Baud Decoder Description

Address Decoder is a 16-bit register flop that will write the output a write

when PORT\_ID set the Write Strobe value. As well as, outputting the 16-

bit read signal when PORT\_ID set the Read Strobe value.



6.3 PED Positive Edge Detector

6.3.1 PED Positive Edge Detector Description

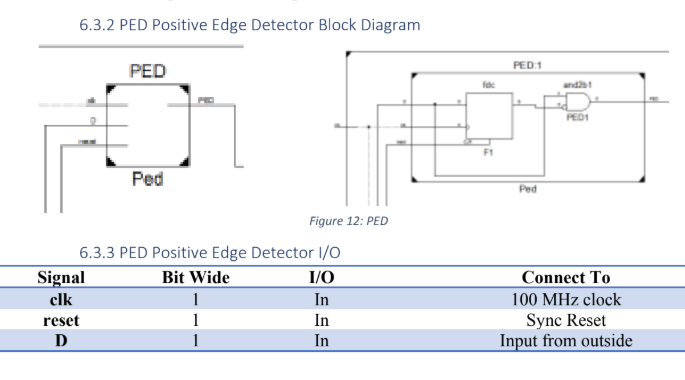
Positive Edge Detect: Detect the edge of a given signal, when the signal

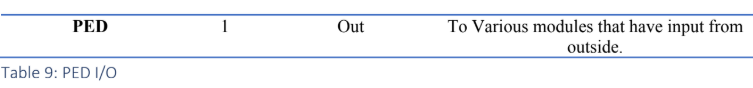
goes high, so it will output a one, for this project: it only detects from low

to high. This flop generates one clock cycle tick when the input signals

change from low to high.

6.3.2 PED Positive Edge Detector Block Diagram



6.4 RS FLOP

6.4.1 RS FLOP Description

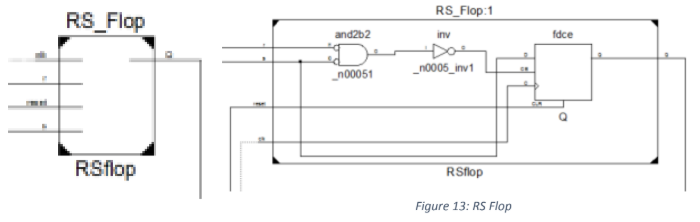
RS FLOP: Reset and Set Flop, it contains 2 reset and 1 set. One reset

comes from AISO and the other reset comes from the Interrupt

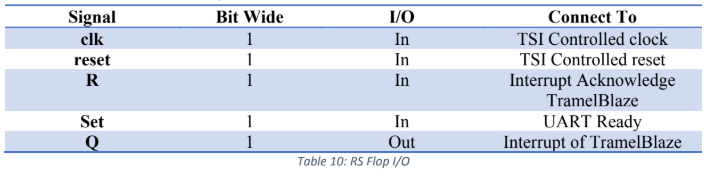
Acknowledge, the reason is to restart the SET signal that is being sent to

Interrupt Port of Tramelblaze.

6.4.2 RS FLOP Block Diagram



6.4.3 RS FLOP I/O



6.5 TX Transmit Engine

6.5.1 TX Transmit Engine Description

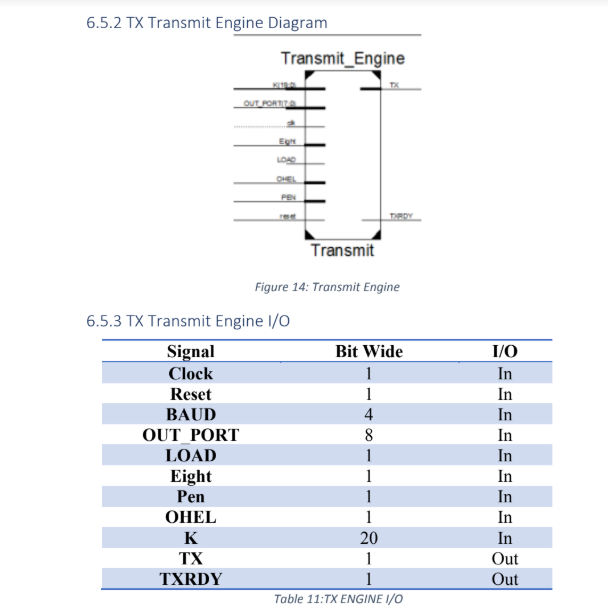
The Transmit Engine transmits data with the help of Tramelblaze. We

check parity to confirm that the data is being sent properly or not. To fully

confirm that, we also receive data checked through HyperTerminal or Real

Term at different configuration of our baud rate to ensure that the correct

data is being transmitted and there is no data loss.



6.5.4 TX Transmit Engine Internal composition

6.5.4.1 TX READY

The RS TXRDY flop informs the processor that it is ready for data

and when the processor writes the data TXRDY flop will be

inactive until the data is being transferred. While the transmit

engine transmitting the data, TXRDY become inactive. On the

other hand, the receive engine must was wait until completely

receive the data to tell the processor that the receive engine got the

data.

6.5.4.2 BTU COUNTER

BAUD block is a combinational block, which has the input of 4

switches. Outputting the constant that is the number of clocks per

bit time. Baud rate will change the bit times which determine he

time it takes for the bit to output. The baud rate determines the bit

length. Taking the bit length and dividing by the period of the

clock that gives us how many clocks there are in a bit time that

comes out decoder as a constant either 19 bits for Nexys 2 or 20

bits for Nexys four. We also need to know when the bit time

elapsed. We can choose our baud rate from the defined switches.

Below it will be bit time counter, which will output BTU – specify

the end of the bit time

DoIt data will be send to this block, if DoIt is logic zero, the

transmit will be idle, no comparing operation is working at that

moment. Leading the BTU to be low as well, if DoIt is logic one

and the count has reached the maximum, the output of BTU will be

high and transferred into the SH of shift register. BTU will shift

one to the next bit.

6.5.4.3 DONE COUNTER

Done Counter is used to notify when all the bits are sent. A Mux,

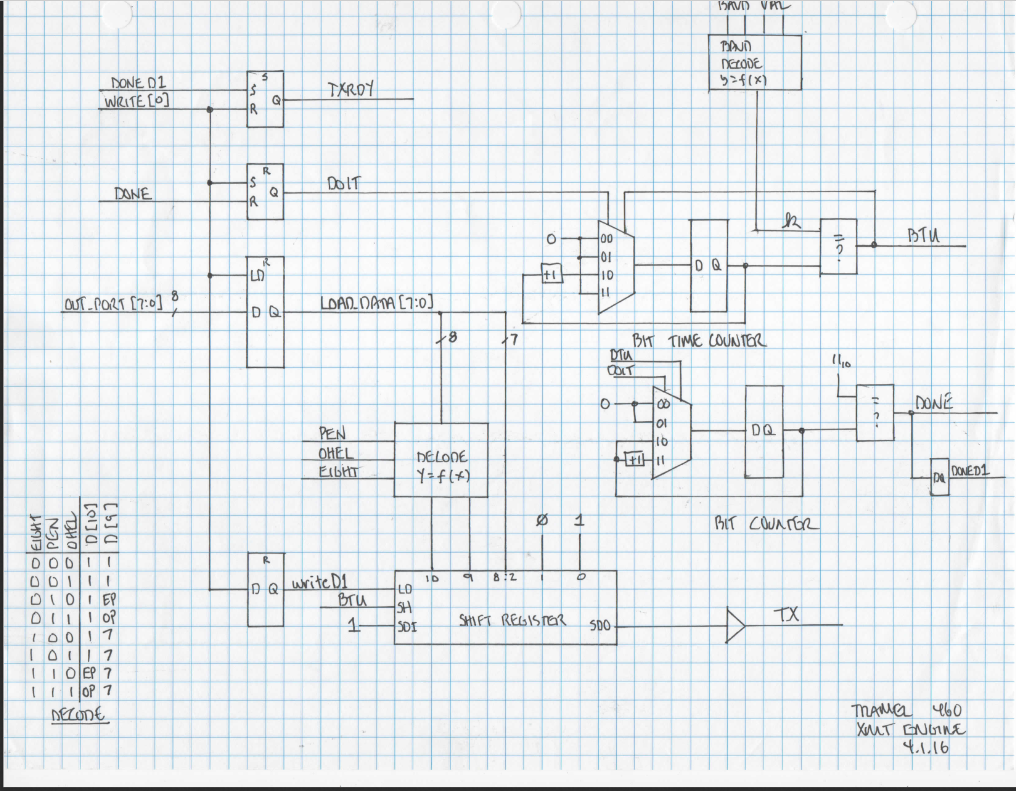
A Register and a Counter will be used to create this block, counter

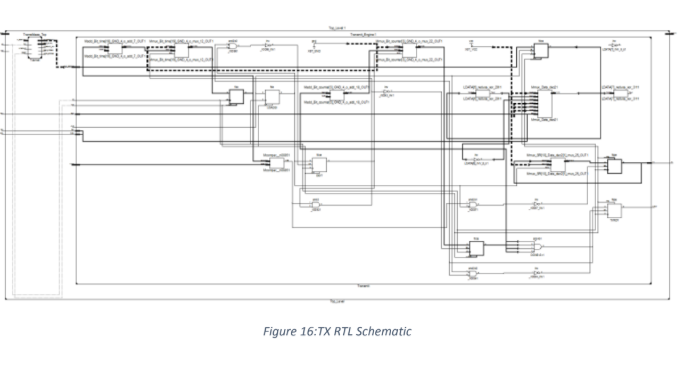
will count to decimal 11, representing 11 bits that being sent.

DOIT = 1 BTU = 0: value is kept unchanged

DOIT = 1 BTU = 1: increment the counter.

Default: stay unchanged





6.6RX Receive Engine

6.6.1 RX Engine Description

The Rx Engine is the receiving section of the UART module. It is a

serial receiver that has EIGHT, PEN, OHEL, and BAUD as the

controlling inputs. EIGHT determines whether the data received is 7 or 8

bits long. PEN is parity enable, in which it allows the receiver to

compare the generated parity with the one received, and to some extend

(not 100% efficient) determine whether there was data corruption during

the transmission/receiving. OHEL determines whether the parity

generated and compared will be even or odd logic. The baud rate

determines the delay between bits (BAUD is the bits transferred per

second). As far is how it captures data, the Rx engine has a tristate

machine in which it waits to receive the start bit of a transmission. Once

the start bit is received, the next state is to generate half a BTU so

sample the data half way through the BTU to avoid data corruption.

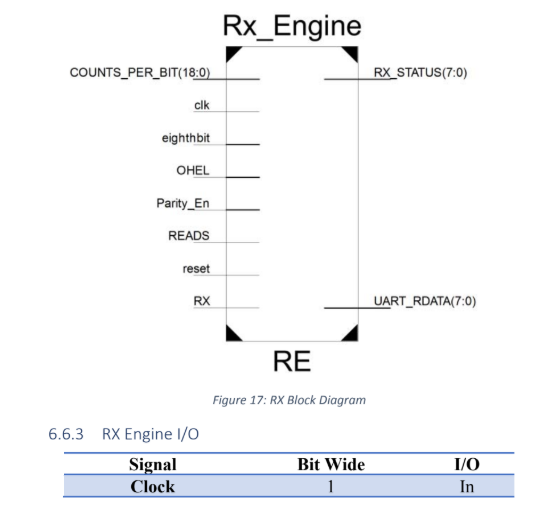
From there, once the shift register collects the data, parity bit (if enabled)

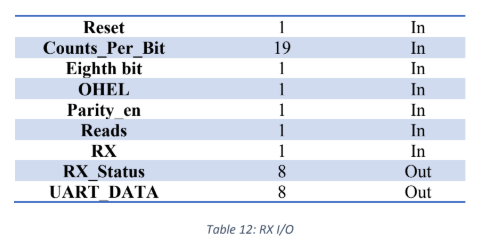
and stop bit, the data flows down to a combo register that aligns the data

to the LSB of the register. From there the status flags are generated the

Rxrdy is asserted.

6.6.2 RX Engine Block Diagram





6.6.4 RX Engine Internal Composition

6.6.4.1 Shift Register

The shift register used to capture the data in the Rx engine is an

MSB right shift register. When BTU is asserted, the shift register

will place the bit in the Rx signal in the MSB of the register and

shift the data right. Once the data is done collecting, the shift

register will then pass the data onto the Remap combo logic

register.

6.6.4.2 Remap Combo Logic

The Remap combo logic register shifts the data received to the

right depending on EIGHT, PEN, OHEL input signals.

6.6.4.3 Rxrdy logic

The Rxrdy flop is asserted once the Rx engine is done receiving a

frame of data. The Rxrdy is used to signal to the processor that

data has been collected and is ready to be stored.

6.6.4.4 PERR Logic

PERR is the parity error flag that is asserted if the generated

parity bit does not match the received parity bit. It is a way to

determine whether data was corrupted throughout the

transmitting receiving process.

6.6.4.5 FERR Logic

FERR is the framing error flag that is set if the period of a

frame (cycle of data transmission) is violated.

6.6.4.6 OVF Logic

OVF is the overflow flag that is asserted when the Rxrdy and

DONE signals are both high at the same period.

6.6.4.7 BTU Counter

BTU counter is the bit time unit counter that is used to generate

the delay between the sampling of bits. The counter is

determined by the output of the BAUD decoder (k). K is

determined by the BAUD input set by the switches at the

hardware level.

6.6.4.8 Counts per bit Counter

A register logic that counts the number of times BTU has been

asserted to keep track of how many bits have been received.

Once the DONE counter has reached the counter value

determined by the counter. the signal is asserted, to signal that

the number of desired bits has been received.

6.6.4.9 Half BAUD

Half BAUD is used (K/2) to sample the Rx signal at half the

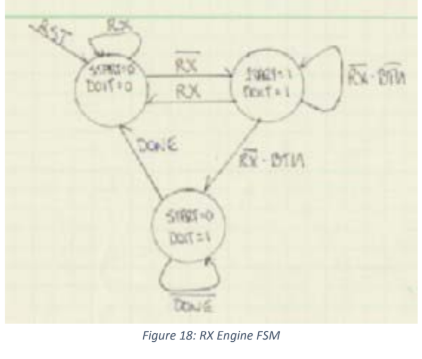
period that it is sent. It is a method to avoid data corruption by

keeping an error threshold when sampling the data. When a start

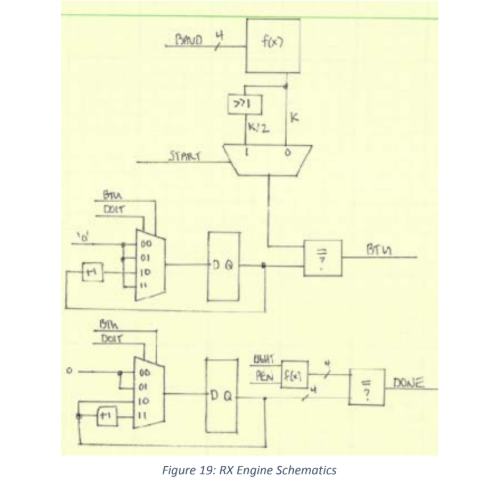
bit is received, the Rx engine waits half a BTU to begin sampling

the data half way through their bit counter.

6.6.5 RX Engine State Machine



6.6.6 RX Engine Schematics



6.7 UART

6.7.1 Description

The UART module instantiates the Tx Engine and Rx Engine, along

with OR gate with the Txrdy and Rxrdy, and providing a multiplexer for

the output of the Rx engine, which determines whether to output the data

received, or the status flags generated from the data. The Txrdy and

Rxrdy are connected to an OR gate and output to provide a single

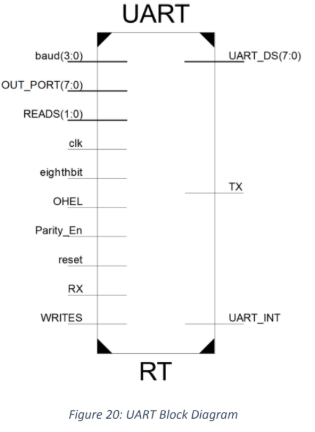
interrupt signal. The multiplexer is a selector that selects between the

data register, and the status flag register. When Reads [0] is asserted, the

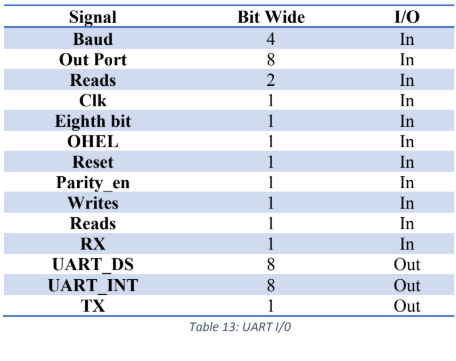
data is the output of the mux, otherwise when Reads [1] is asserted, the

status flags are passed out to the output.

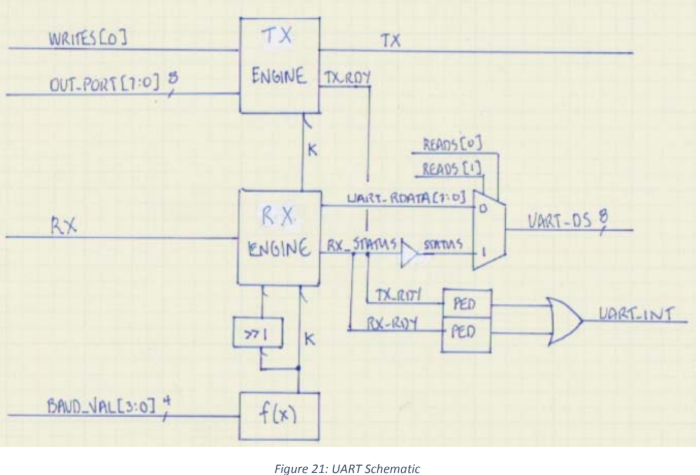
6.7.2 Block Diagram



6.7.3 UART I/O



6.7.4 UART Schematic



6.8 TSI – Technology Specific Instantiation

6.8.1 Description

The technology specific instantiation contains all the references to the

target technology library, Nexys4. Each I/O must have a targeted device to

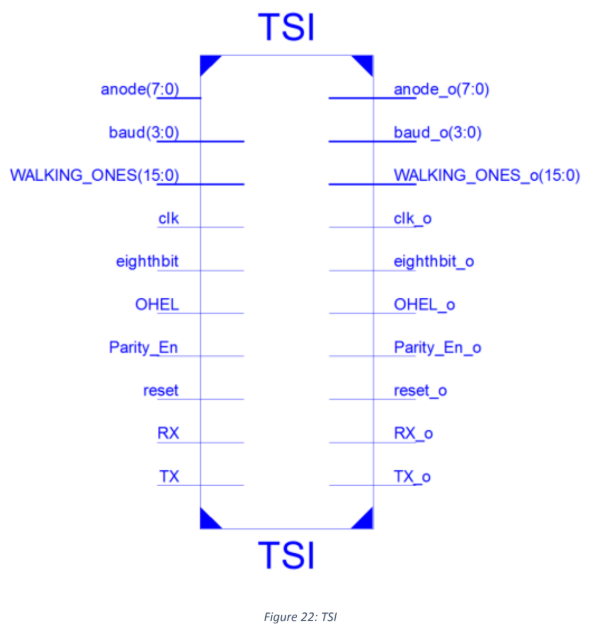
meet timing and electrical requirements that the manufacturing company

has defined. The IOBUF is a bi-directional buffer that allows core logic to

interface with chip I/O. The buffers also allow for a change in level of

signals. A change of level is needed when the I/O of the core logic is a

different voltage to that of the chip I/O.

6.8.2 Block Diagram

7 CHIP LEVEL VERFICATION

CHIP LEVEL VERIFICATION

N/A

CHIP LEVEL TEST

The undersigned acknowledge they have reviewed the UART Product Design Specification document and agree with the approach it presents. Any changes to this requirements definition will be coordinated with and approved by the undersigned or their designer’s representatives.

Signatures:

Print Name:

Title:

Role:

Date:

Appendix A: References

The following table summarizes the documents references in this document

|  |  |  |
| --- | --- | --- |
| Document Name and Version | Description | Location |
| UART\_ChipSpec 1.0 | Chip specification for the UART project assigned professor John Tramel in CECS 460 at CSULB. | N/A |

Appendix B: Key Terms

The following table provides definitions for terms relevant to this document in alphabetical order.

|  |  |
| --- | --- |
| Term | Definition |
| AISO | Asynchronous In, Synchronous Out |
| CSULB | California State University, Long Beach |
| FSM | Finite State Machine |
| I/O | Input/output |
| ISR | Interrupt Service Routine |
| LED | Light Emitting Diode |
| PED | Positive Edge Detect |
| Rx | Receive |
| TSI | Technology Specific Instantiation |
| Tx | Transmit |
| UART | Universal Asynchronous Receiver- Transmitter |

8. CHIP LEVEL TEST

8.1 Assembly Code

8.1.1 Full\_UART.tba

8.2 Modules (.V files)

8.2.1 Address decoder.v

8.2.2 AISO.v

8.2.3 Baud\_Rate.v

8.2.4 Bit\_Counter.v

8.2.5 Bit\_Time\_Counter.v

8.2.6 D\_Flop.v

8.2.7 Decode.v

8.2.8 Lab3\_Top.v

8.2.9 Lab4\_Top.v

8.2.10 Id\_reg.v

8.2.11 PED.v

8.2.12 remap.v

8.2.13 RX\_Control.v

8.2.14 RX\_DataPath.v

8.2.15 RX\_Top.v

8.2.16 Shift\_Reg10bit.v

8.2.17 Shift\_Reg11bit.v

8.2.18 SR\_Flop.v

8.2.19 TSI.v

8.2.20 TX\_Engine.v

8.2.21 UART.v