Management and analysis of physics datasets, Part. 1

Sixth Laboratory

Antonio Bergnoli bergnoli@pd.infn.it

30/11/2021

Laboratory Introduction

Goals

- exploit the configuration flash for the **100t** boards (refer to lab 3 slides)
- study the edge detection for synchronous signals
- Discover The Arty7 FPGA Board
- build a UART transmit Unit

Configuration flash

Add a configuration memory to project

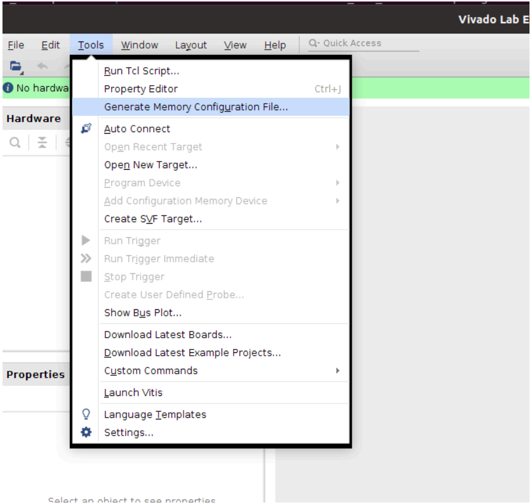
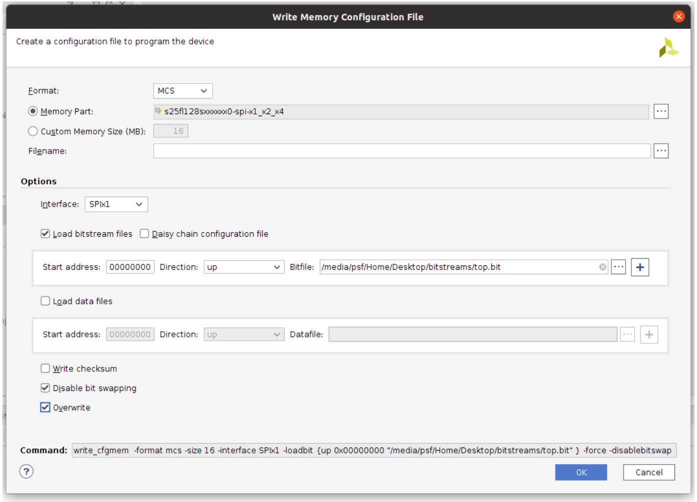


Figure 1: Add a configuration memory to project

flash chip selection



Write Memory Configuration File

Create a configuration file to program the device

Format: MCS

● Memory Part: s25fl128sxxxxxx0-spi-x1_x2_x4

○ Custom Memory Size (MB): 16

Filename:

Options

Interface: SPIx1

☑ Load bitstream files ☐ Daisy chain configuration file

Start address: 00000000 Direction: up Bitfile: /media/psf/Home/Desktop/bitstreams/top.bit

☐ Load data files

Start address: 00000000 Direction: up Datafile:

☐ Write checksum

☑ Disable bit swapping

☑ Overwrite

Command:

```
write_cfgmem  -format mcs -size 16 -interface SPIx1 -loadbit {up 0x00000000 "/media/psf/Home/Desktop/bitstreams/top.bit" } -force -disablebitswap
```

Figure 2: flash chip selection for 100t boards

Edge detection circuits

Positive Edge detection

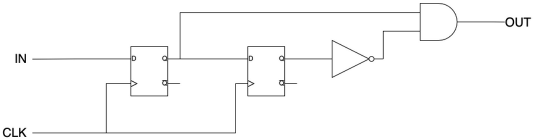


Figure 3: Positive Edge detection circuit

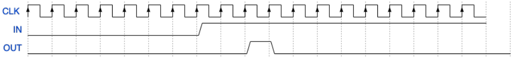


Figure 4: Positive Edge detection waves

Negative Edge detection

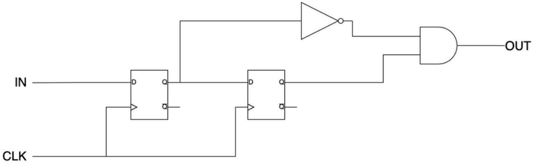


Figure 5: Negative Edge detection circuit

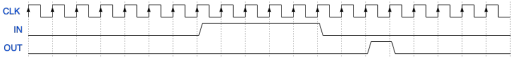


Figure 6: Negative Edge detection waves

Both Edges detecition

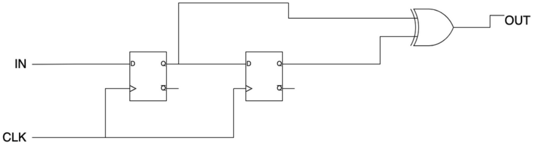


Figure 7: Both Edges detection circuit

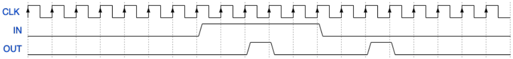


Figure 8: Both Edges detection waves

Arty 7 FPGA Board

A closer look to the Board

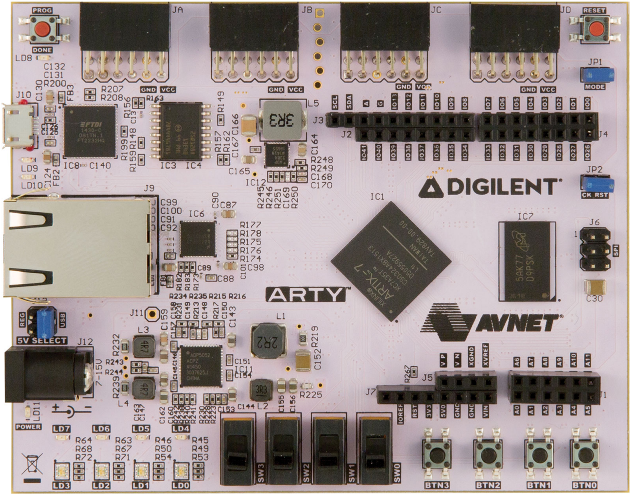
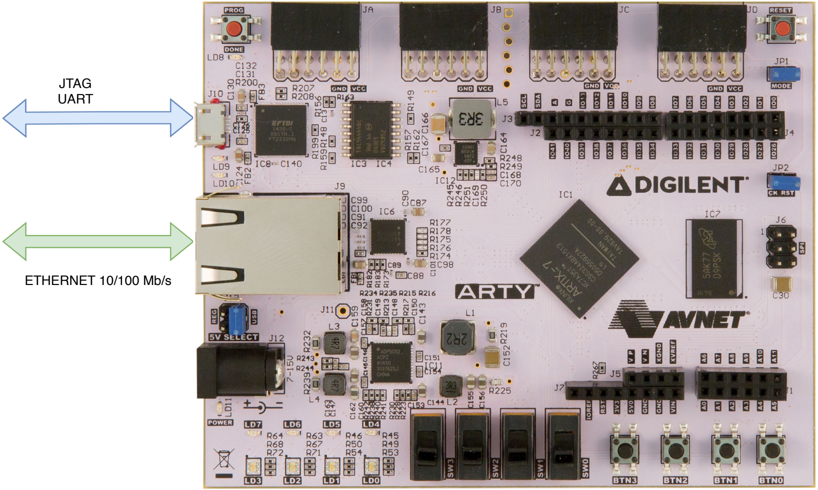


Figure 9: Arty 7 Board

Arty7: Useful interfaces

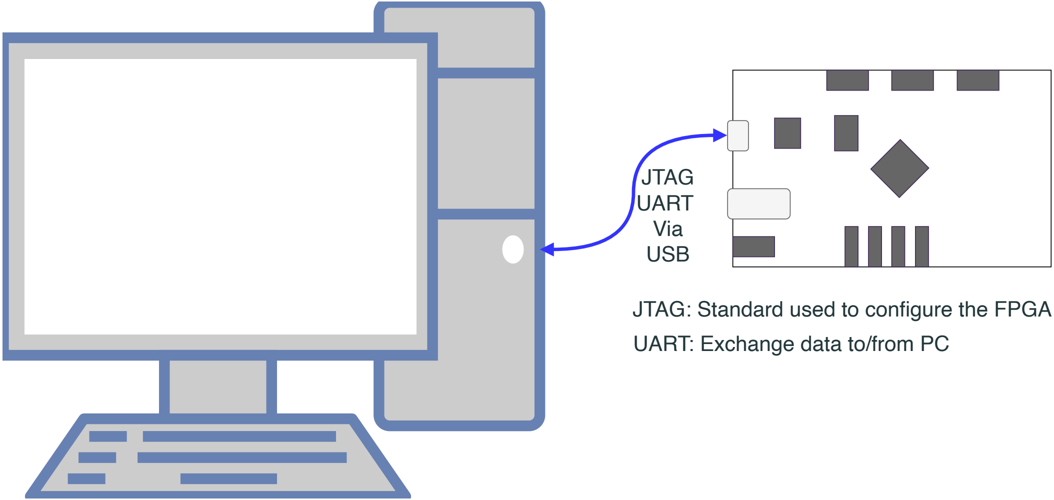


JTAG
UART

ETHERNET 10/100 Mb/s

Figure 10: Arty 7 interfaces

Arty7 connected to a PC



JTAG
UART
Via
USB

JTAG: Standard used to configure the FPGA

UART: Exchange data to/from PC

Figure 11: Arty 7 connect to a PC

Motivation

- Since we need to connect our FPGA Board to a PC, we need to enable a reasonable useful interface .
- Some interfaces are available, but we need to manage them:

Interface	Speed	tipical usage	Ease of use
Ethernet	100 Mbit/s	General purpose	very high
JTAG	< 10 Mbit/s	Config/Debug	moderately high
UART	$< < 1$ Mbit/s	text messages	**Easy**

Universal **A**synchronous **R**eceiver **T**ransmitter: The device

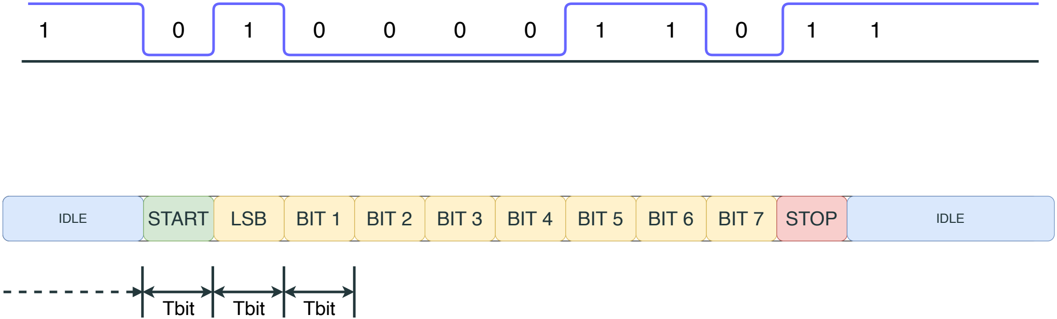
RS-232: the standard protocol

Introducing the RS-232 standard

The RS-232 Data frame: an example

- **Byte oriented frame**
- Mostly used to exchange *character stream* (ASCII code, see next slide) but also binary stream
- optional parity bit at the end of the trasmitted byte
- **start** and **stop** frame delimiter length = 1 bit (stop-bit could be 2 in some cases)

Transmit Character 'a' (ASCII CODE hexadecimal: 61, binary: 01100001) at 115200 Bit/second, no parity bit



1	0	1	0	0	0	0	1	1	0	1	1
IDLE	START	LSB	BIT 1	BIT 2	BIT 3	BIT 4	BIT 5	BIT 6	BIT 7	STOP	IDLE

Tbit Tbit Tbit

Tbit=1/115200 s = 8,68 us

The ASCII standard

```
The hexadecimal set:

    00 nul   01 soh   02 stx   03 etx   04 eot   05 enq   06 ack   07 bel
    08 bs    09 ht    0a nl    0b vt    0c np    0d cr    0e so    0f si
    10 dle   11 dc1   12 dc2   13 dc3   14 dc4   15 nak   16 syn   17 etb
    18 can   19 em    1a sub   1b esc   1c fs    1d gs    1e rs    1f us
    20 sp    21  !    22  "    23  #    24  $    25  %    26  &    27  '
    28  (    29  )    2a  *    2b  +    2c  ,    2d  -    2e  .    2f  /
    30  0    31  1    32  2    33  3    34  4    35  5    36  6    37  7
    38  8    39  9    3a  :    3b  ;    3c  <    3d  =    3e  >    3f  ?
    40  @    41  A    42  B    43  C    44  D    45  E    46  F    47  G
    48  H    49  I    4a  J    4b  K    4c  L    4d  M    4e  N    4f  O
    50  P    51  Q    52  R    53  S    54  T    55  U    56  V    57  W
    58  X    59  Y    5a  Z    5b  [    5c  \    5d  ]    5e  ^    5f  _
    60  `    61  a    62  b    63  c    64  d    65  e    66  f    67  g
    68  h    69  i    6a  j    6b  k    6c  l    6d  m    6e  n    6f  o
    70  p    71  q    72  r    73  s    74  t    75  u    76  v    77  w
    78  x    79  y    7a  z    7b  {    7c  |    7d  }    7e  ~    7f del
```

The Universal Asynchronous Receiver-Transmitter

Communication with UART

UART communication is made on a **point-to-point** connection

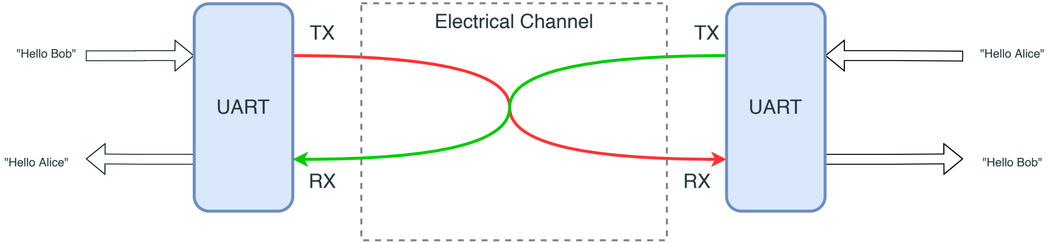


Figure 13: UART typical interconnection

The UART subsystem in the Arty7 Board

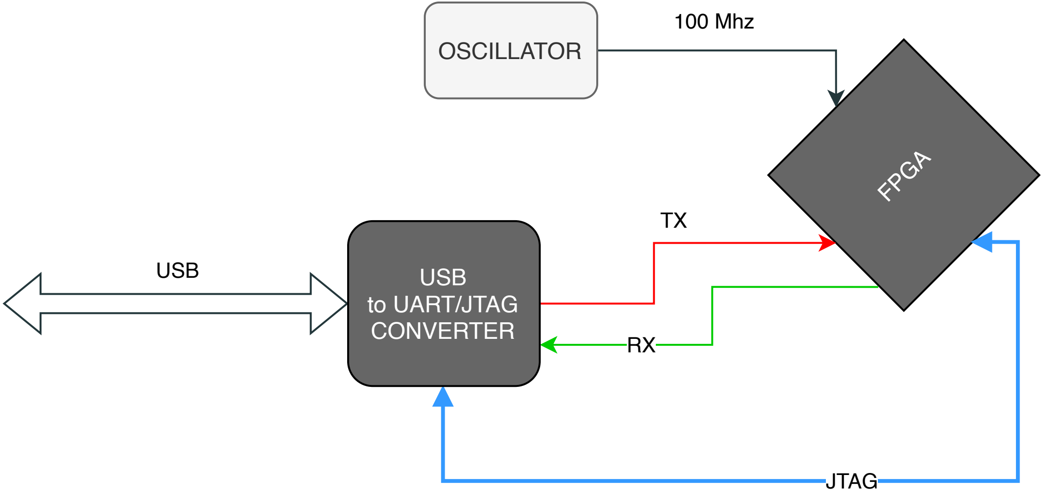


Figure 14: Arty7 Uart schematic

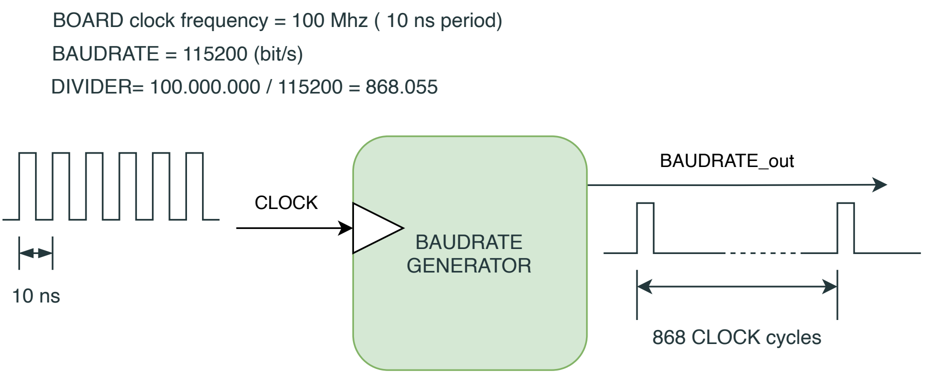
Build a UART transmitter

The Baudrate generator

- it is basically a counter
- it has only one output
- output is equal to ‘1’ for exactly one clock cycle every *bit time (Tbit)*

Motivation

We need a periodic signal (another clock) at the rate of the transmission. **Baudrate:** speed in symbol/s (1 symbol carries 1 bit)



BOARD clock frequency = 100 Mhz (10 ns period)

BAUDRATE = 115200 (bit/s)

DIVIDER= 100.000.000 / 115200 = 868.055

Figure 15: The Baudrate generator

First exercise

Write a Baudrate generator in VHDL, in the case of:

- Clock frequency= 100 Mhz
- Baudrate = 115200

The Uart Transmitter

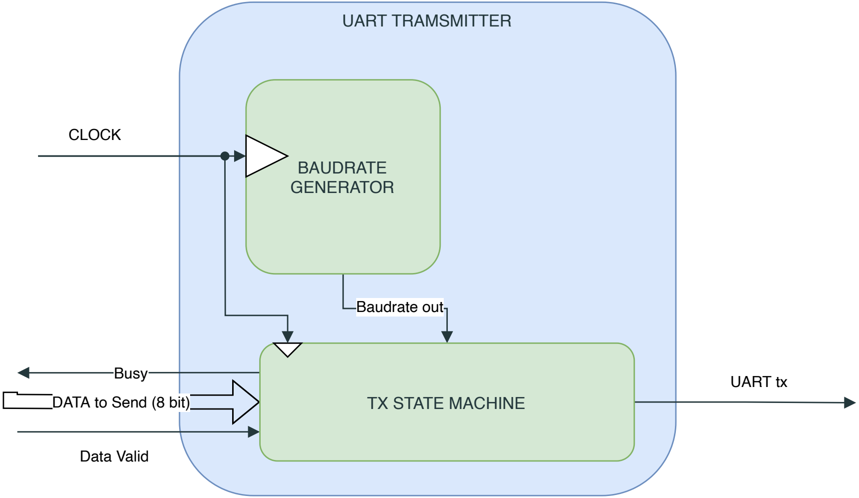


Figure 16: The Uart Transmitter

The Uart Transmitter State machine

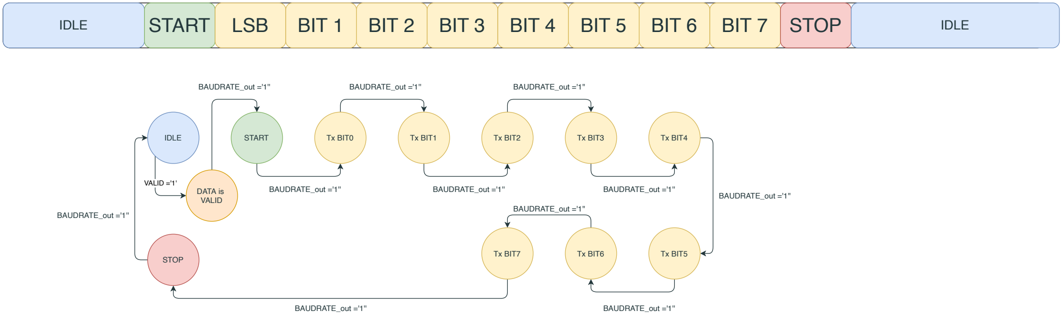


Figure 17: tx State machine idea

The BAUDRATE out signal make possible to switch from a state to another

Transmitter waveform

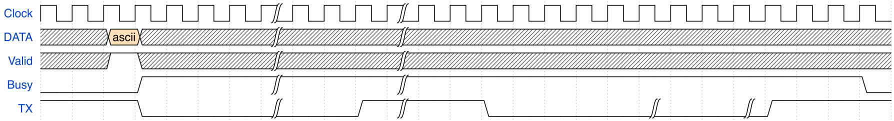


Figure 18: UART transmitter waveform

Homework

Homework

- Build a complete Uart transmitter working at data rate of 115200 baud