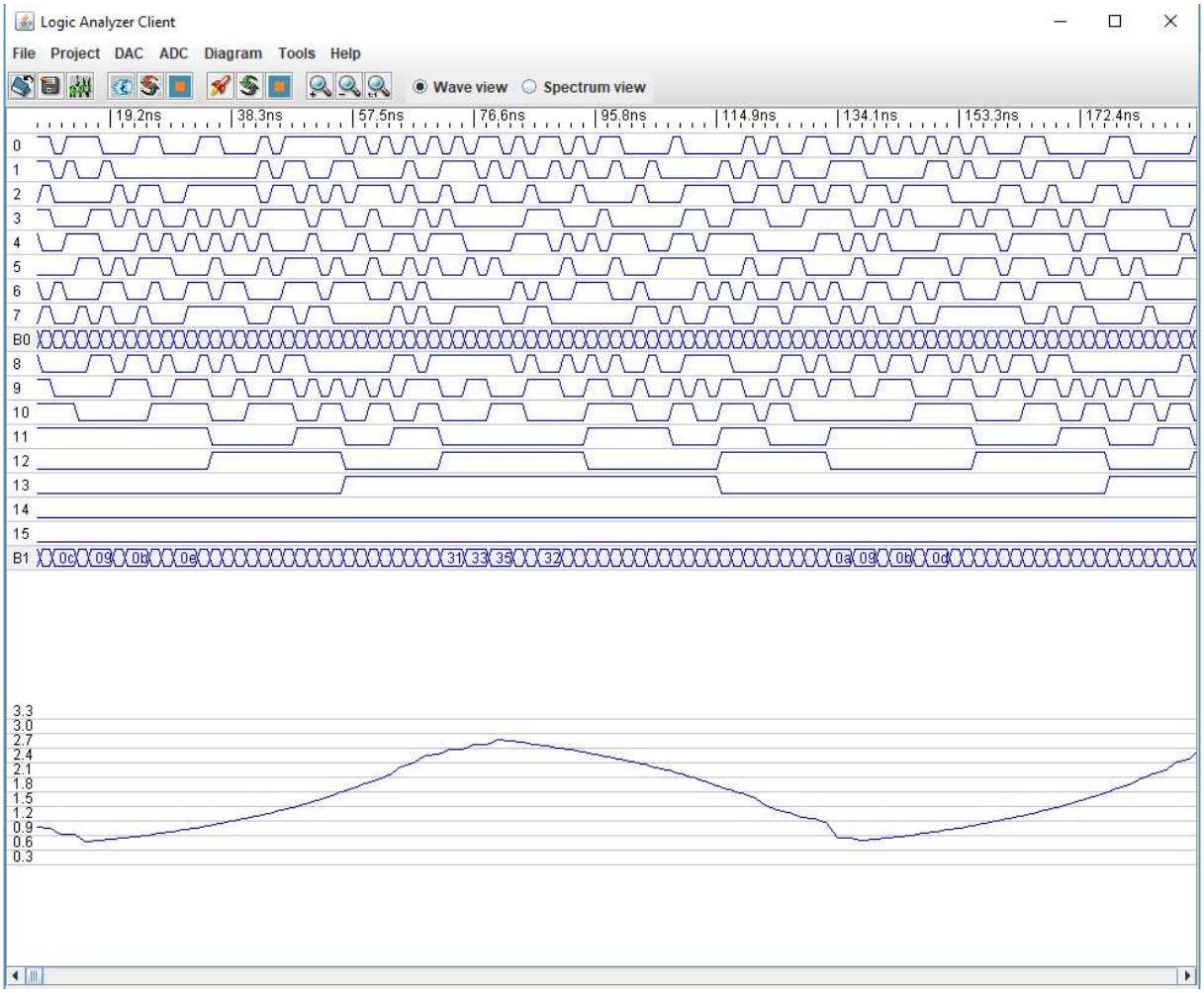
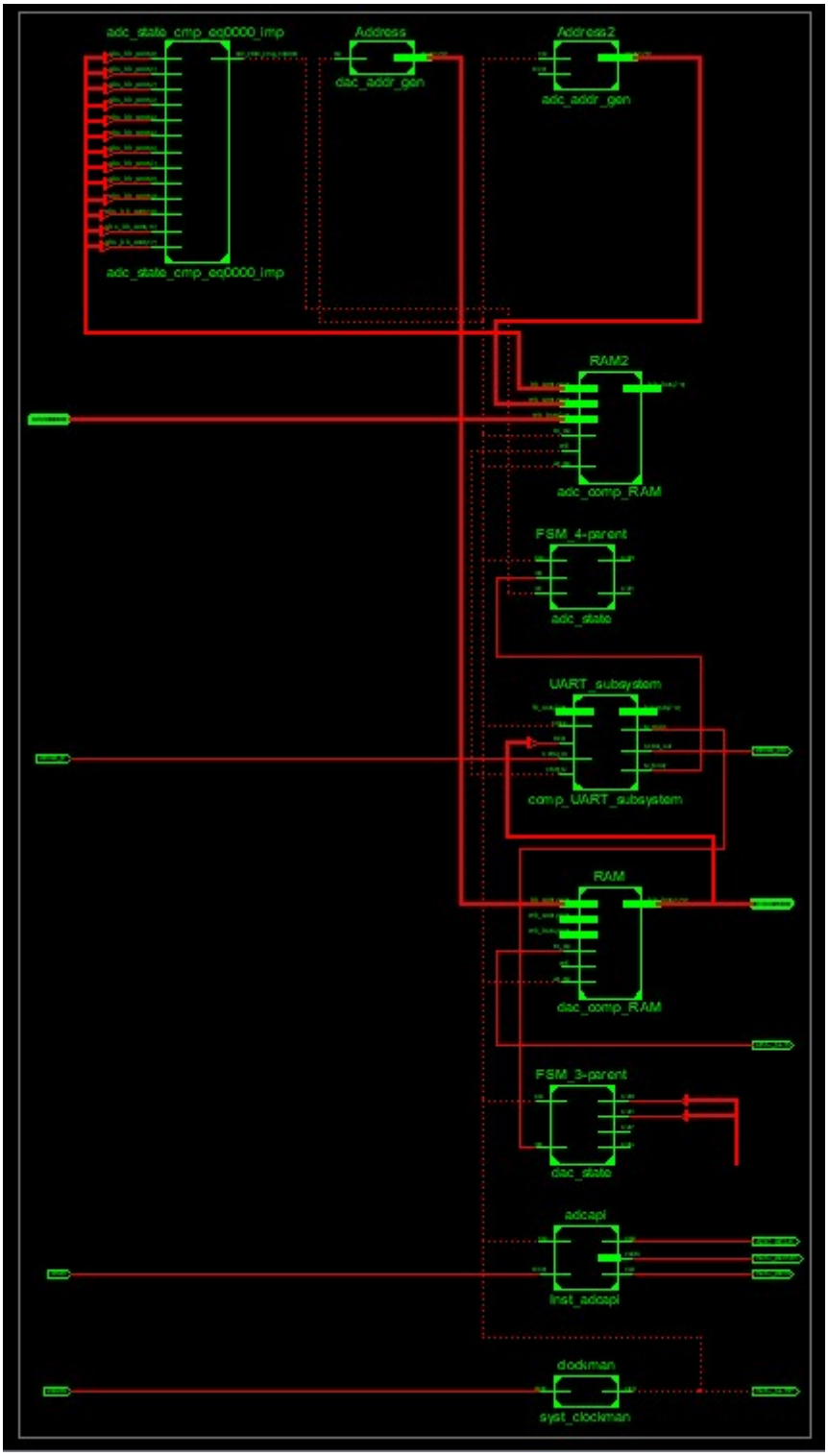
Project SumpX15 – ADC/DAC

The goal of this project is to store data captured from ADC, to replay over DAC. In first phase of the project, each sample is transmit to the DAC from PC client, then in second phase of the project, all data are received by the ADC automatically replayed on command, these data are stored in the memory bloc. The DAC buffer size is 8192 samples of 13 bits and the ADC buffer size is 8192 samples of 8 bits. In order to do a test, the loop-back configuration from output to input can be used here. All sent out data will be recorded back.

The FPGA used in this project is a Xilinx Spartan 3S500E has following characteristics, Logic cells : 10476 , Distributed RAM bits : 73K, Block RAM bits : 360K, Dedicated Multipliers 20, Digital Clock DCM : 4. The DAC operates at 125 MHz. The max. freq. of the ADC can be pushed up to 80 Mhz according to manufacturer specification. By default, all working in same frequency of 50 Mhz.



System Top Level :



This is 2-levels system design

UART

Allows to communicate with the FPGA Core using UART interface. The module (de-)serializes data and handles all data transfer between PC Client and the FPGA.

FPGA Core

The core contains modules written in VHDL and provides a simple interface to those components. The ADC_FSM manages the ADC. The DAC_FSM manages the DAC.

Controller (FSM) x 2

1. Controls the capturing & read back from ADC.

2. Controls the transmission & command DAC.

If no other operation has been activated, the ADC controller samples data into the memory. This allows to capture data. In same time, the DAC controller manage incoming data from UART. The FPGA will stop ADC when ADC buffer is full, then wait for UART signal to send data back to PC. The goal is by using UART to receive/send data from/into host to replay stored signal sequence according command from PC Client.

The ADC data are read back from BRAM memory, then should be sent to the UART. Received data from UART is going be stored to BRAM memory too before sending to the DAC.

The ADC and DAC both have internal BRAM for each. 2 Address counters and control signals have been implemented to manage the buffers. These are dual port BRAM instead of single port, because data should be stored and read in same time, giving no interruption of wave output generation.

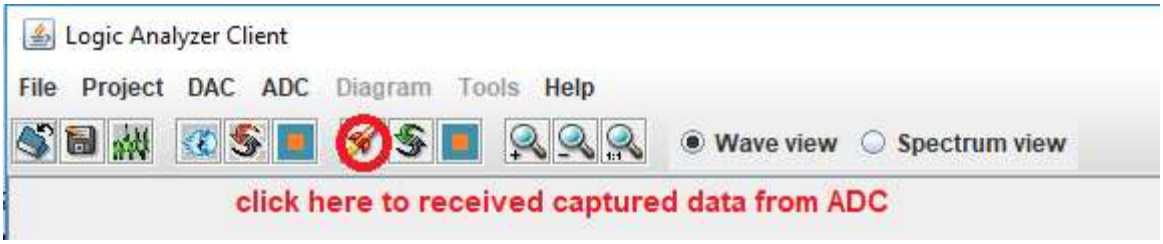
There are 2 signals, tx_ready and rx_done inside the FSM.

Address Generator x 2

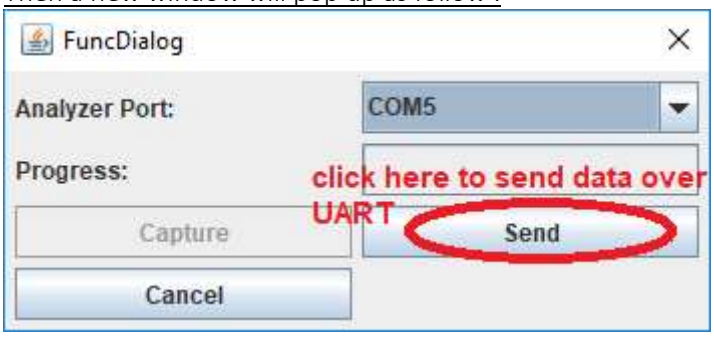
An auto incremental address is generated to access to each BRAM memory in order to get/store the samples that works at same frequency of the ADC/DAC output.

Logic Analyzer Client

The Java client should run on most platforms for which JRE should be deployed with the RXTX package. This means it should be compatible with Linux and Windows. The PC Client has to control the FPGA in order to manage the ADC and the DAC. The client allows to configure the FPGA device, read and display captured data and to perform file operations on captured data.



Then a new window will pop up as follow :



The Main Window will show whole graphic based on captured data :