# Interactive simulation of complex digital systems: a case study

S. Cavaliere

Department of Physics, University of Napoli “Federico II”, Italy and
INFN Sezione di Napoli, Italy

e-mail sergio.cavaliere@na.infn.it

# Outline

- introduction to the (well known) problem
- different approaches to digital simulation
    - advantages
    - problems
- requirements in “our” case
- the proposed approach
- a simple case study
- a sketch of the Babar architecture at a gross definition level
- future work
- conclusion

# Introduction

- Early stage of a digital design requires the evaluation of different scenarios
- These scenarios are usefull in order to evaluate the performances in terms of figures regarding the overall operation (eg. speeds, miss rates, dead times, dimension of transit memories – queues etc. etc.)
- Very important is the
  - ability to freely choose system parameters and the
  - possibility to modify them iteratively, adjusting to the best values
  - Ability to obtain numerical characterization and performance figures of the overall system

# Different approaches to digital simulation advantages vs disadvantages (1)

As a premise all of the following integrate each to the other in the design process

- Verilog/VHDL
  - It is (obviously) the master tool for digital design, verification and synthesis
  - Perfect with the only misadvantage of requiring gate and transition level definition
  - Also the required level of definition makes it impractical in early stages of the design
  - Easy syntactical correctness granted

# Different approaches to digital simulation advantages vs disadvantages (2)

- Petri nets
  - Rather abstract
  - The required level of abstraction drives far from the future hardware implementation
  - More suitable for the evaluation of a realized architecture than for the design of a special purpose architecture from the scratch
- System C
  - C++ software for the design and evaluation of digital systems - both software and hardware
  - Requires large software efforts
  - Lacks of a graphical approach and interactiveness

# A graphical tool: SIMULINK and MATLAB

- Simulink is a tool for the simulation of both discrete and continuous systems, electronic, mechanical and others.
- “Simulink® is an environment for multidomain simulation and Model-Based Design for dynamic and embedded systems.”
- “It provides an interactive graphical environment and a customizable set of block libraries that let you design, simulate, implement, and test a variety of time-varying systems, including communications, controls, signal processing, video processing, and image processing.”

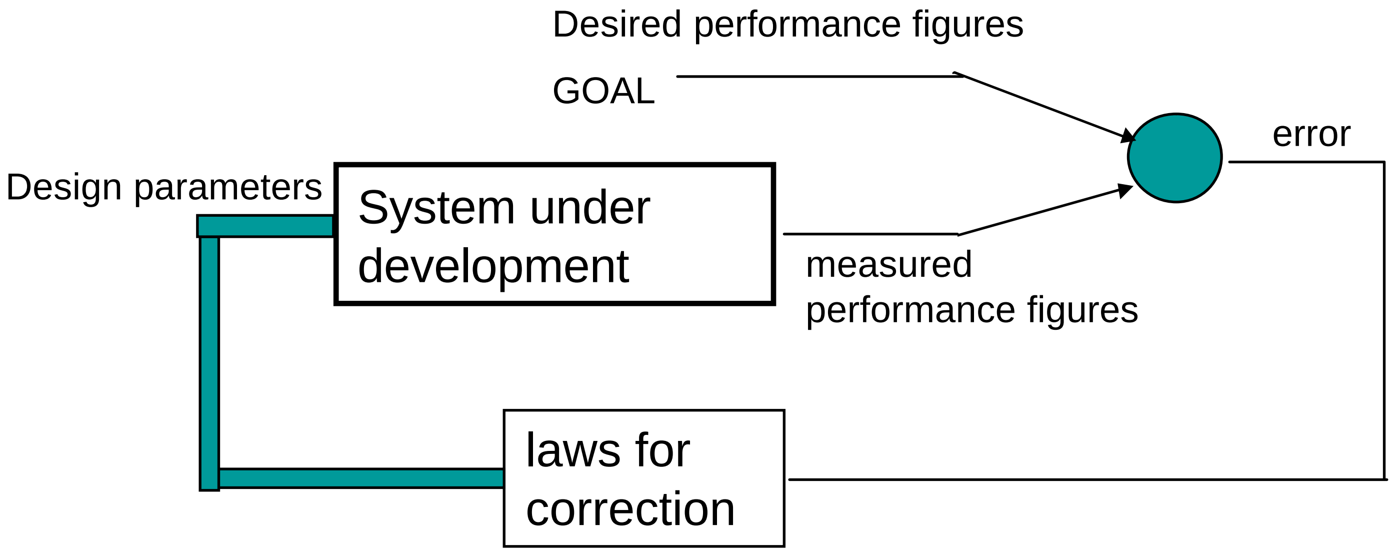
- Simulink anyway is mostly suitable for the solution of pde partial differential equations and finite difference equation: what we did was to adopt it to a entirely digital domain that of digital electronic system, a sort of expansion as we  are going to show in the following

# A graphical tool: SIMULINK and MATLAB

- What helps in the direction of our interactive approach is that:
- Simulink provides a library for discrete systems, first of all the one sample delay and then logic functions and simular
- Simulink also provides Matlab and C language as tools to develop specialized blocks and functions for specific purposes
- Important feature is that the simulation may be developed and modified ***interactively***: sliders switches and similar are provided in order to modify at will parameters of the simulation while it is running
- Often long simulation are needed since the steady state conditions are reached in long times, so it is not enough simulating just short time intervals (think of queues for example)
- Very often we are not able to give deterministic operating conditions but we know statistical laws for the operation of complex interconnected systems
- You may easily simulate faults and analyze consequences on the traffic

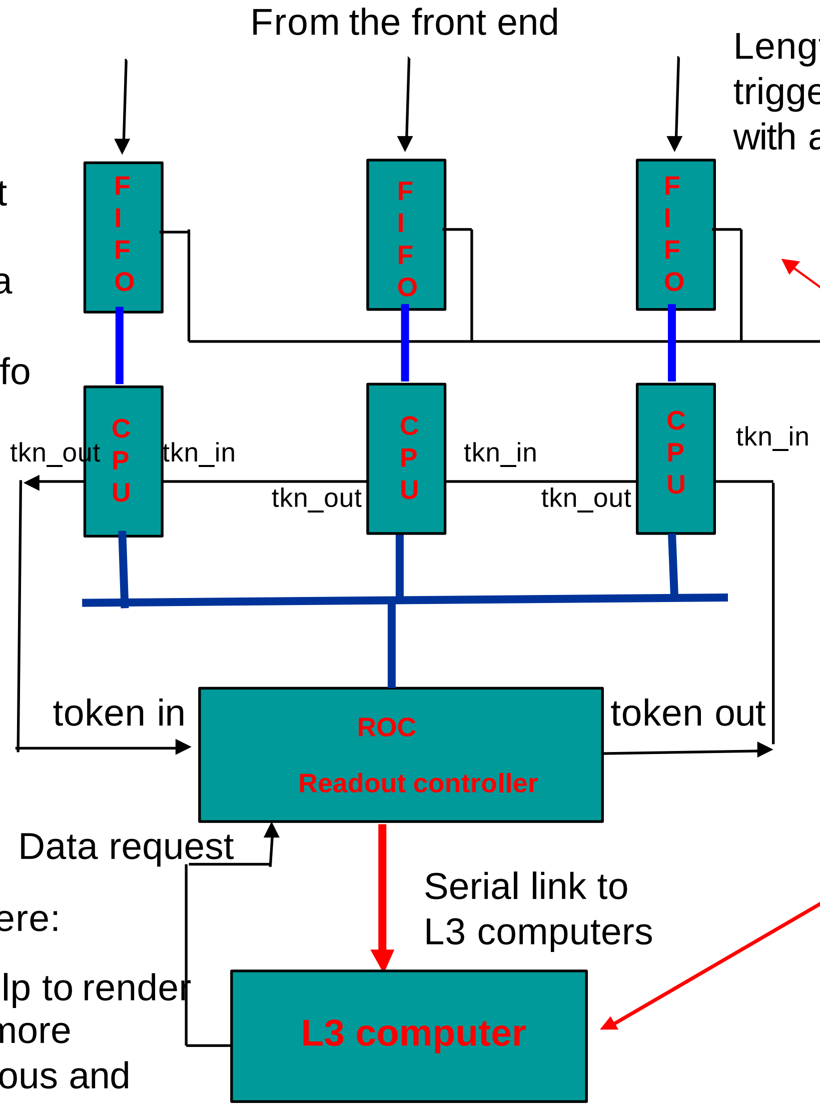
# A graphical approach to design

- **Las but not least, even if not easily practiced, numerical results from the simulation might be fed back to the system under development in a closed loop with the purpose of automatic *optimization*: some measure of the error against the desired performance might modify, with a proper law, the parameters of the system in order to drive to optimal performances**



- **The iterative nature of design tend to be automated if relationship of different and various parameters each to the other are easily controlled**

# A sketch from the BABAR DAQ architecture



Readout module with data fifo and trigger fifo

Length of data block for each trigger has random length, with a specified law

exp pdf

Fifo at the upper level are filled at the the trigger rate

Same fifos are emptied by means of reads requested by the CPU housed in the ROCa single L3 computer

No FIFO here:

It would help to render the traffic more homogeneous and improve the serial link occupancy

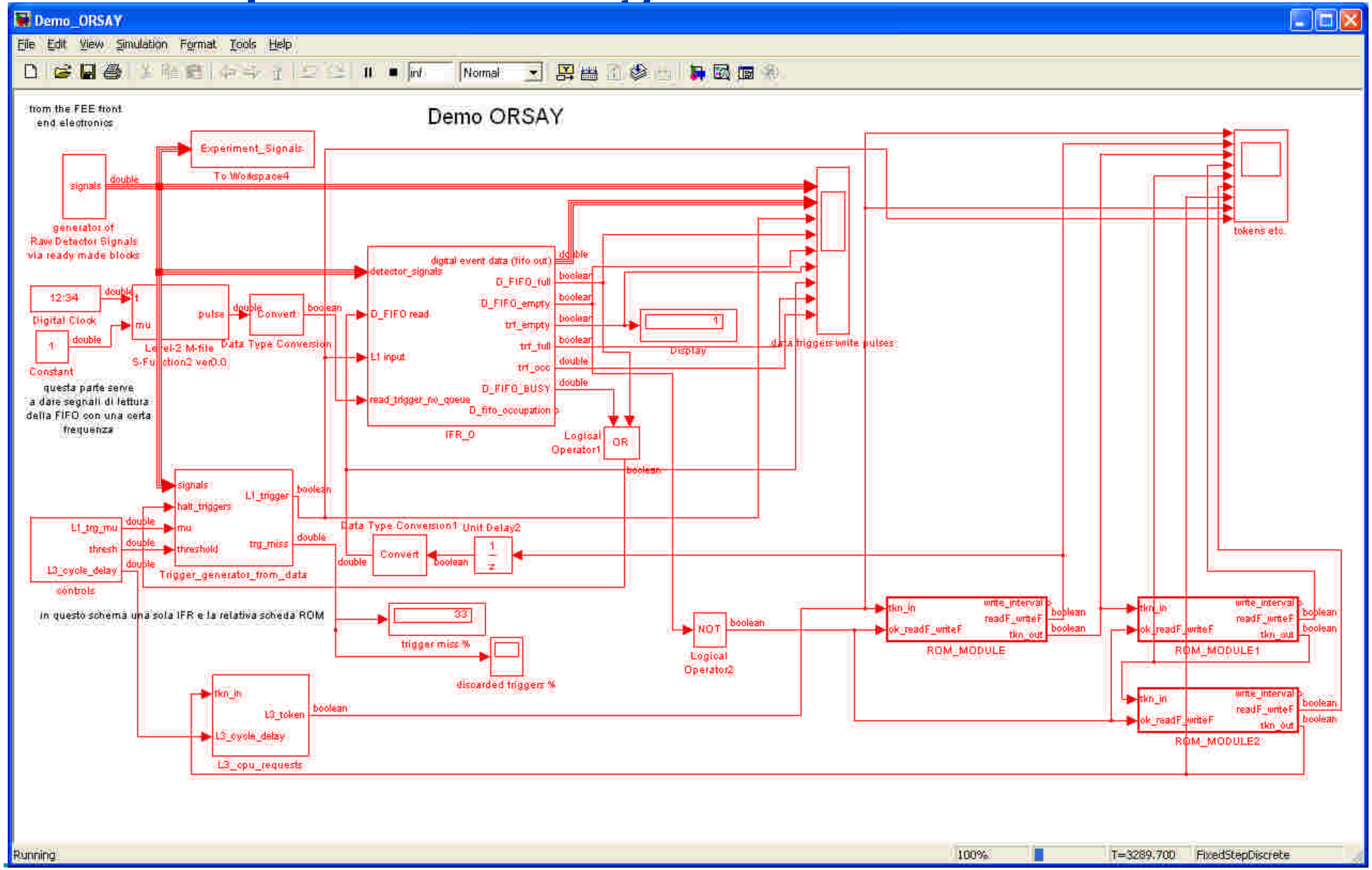
**We can control interactively**

**•the parameters of the trigger generation**

**•The average length of the data chunk for each trigger**

**•The speed of the various CPU**

# The top level design



Demo ORSAY

from the FEE front end electronics

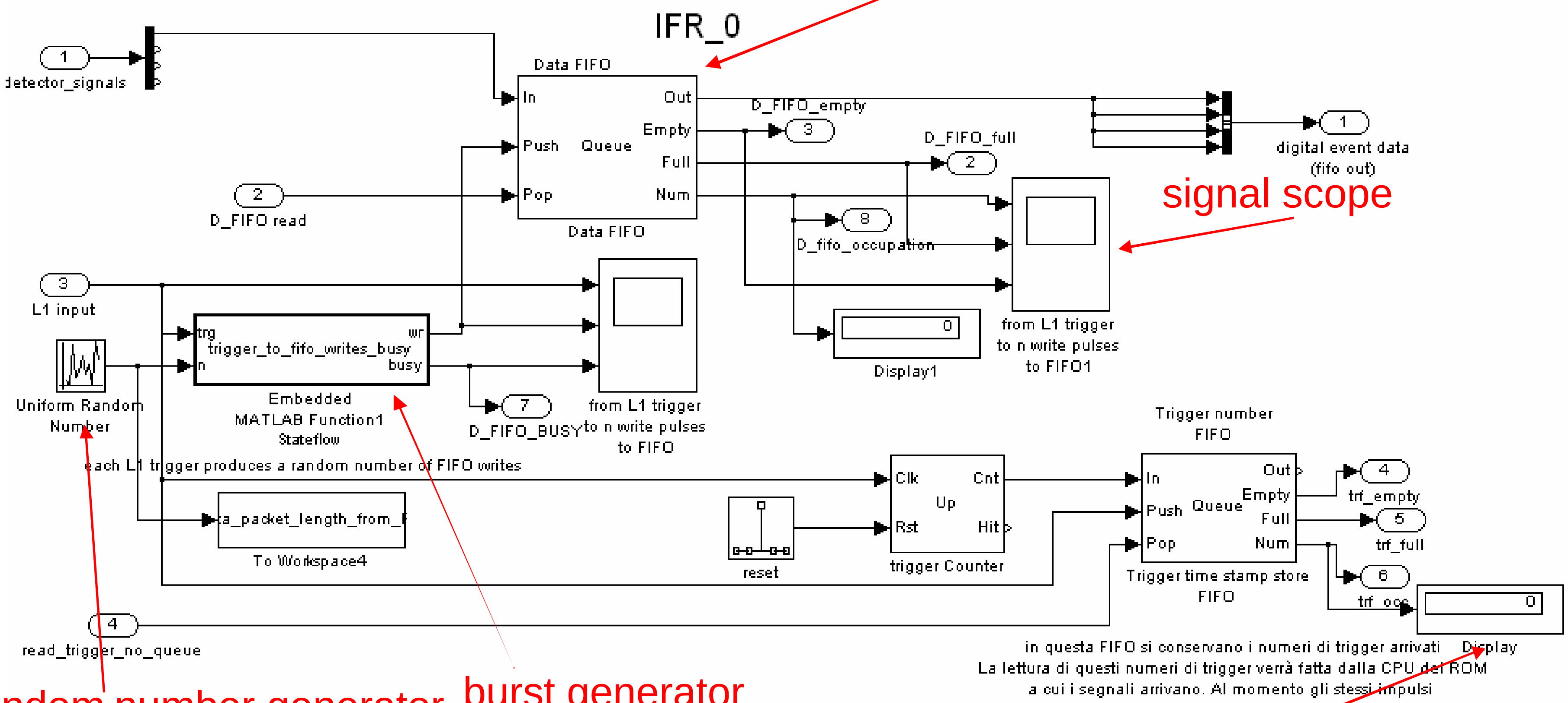
questa parte serve a dare segnali di lettura della FIFO con una certa frequenza

In questo schema una sola IFR e la relativa scheda ROM

Running | 100% | T=3289.700 | FixedStepDiscrete

# Detailing some blocks: the fifo and related logic

The data FIFO queue



each L1 trigger produces a random number of FIFO writes

in questa FIFO si conservano i numeri di trigger arrivati
La lettura di questi numeri di trigger verrà fatta dalla CPU del ROM
a cui i segnali arrivano. Al momento gli stessi impulsi

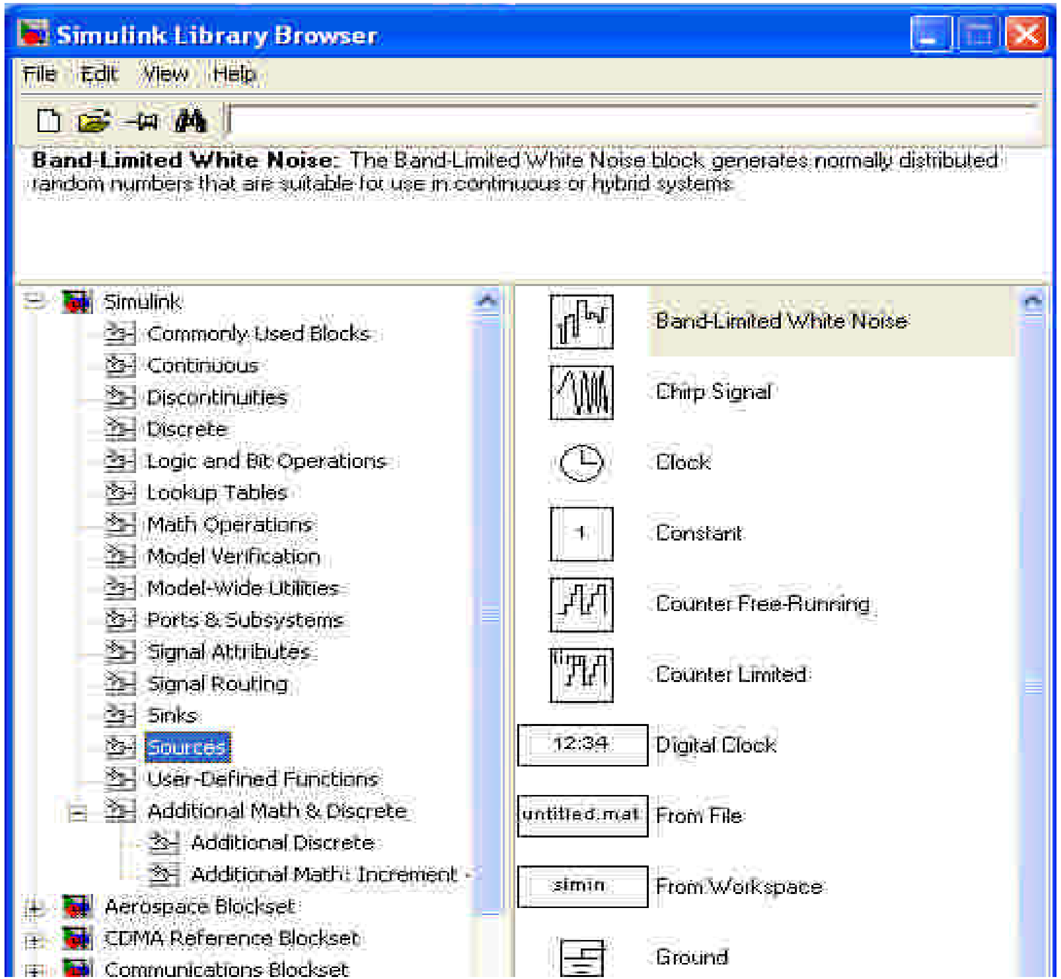
signal scope

random number generator

burst generator

numerical display

# Simulink blocks: the library and purposely built blocks in Matlab/C

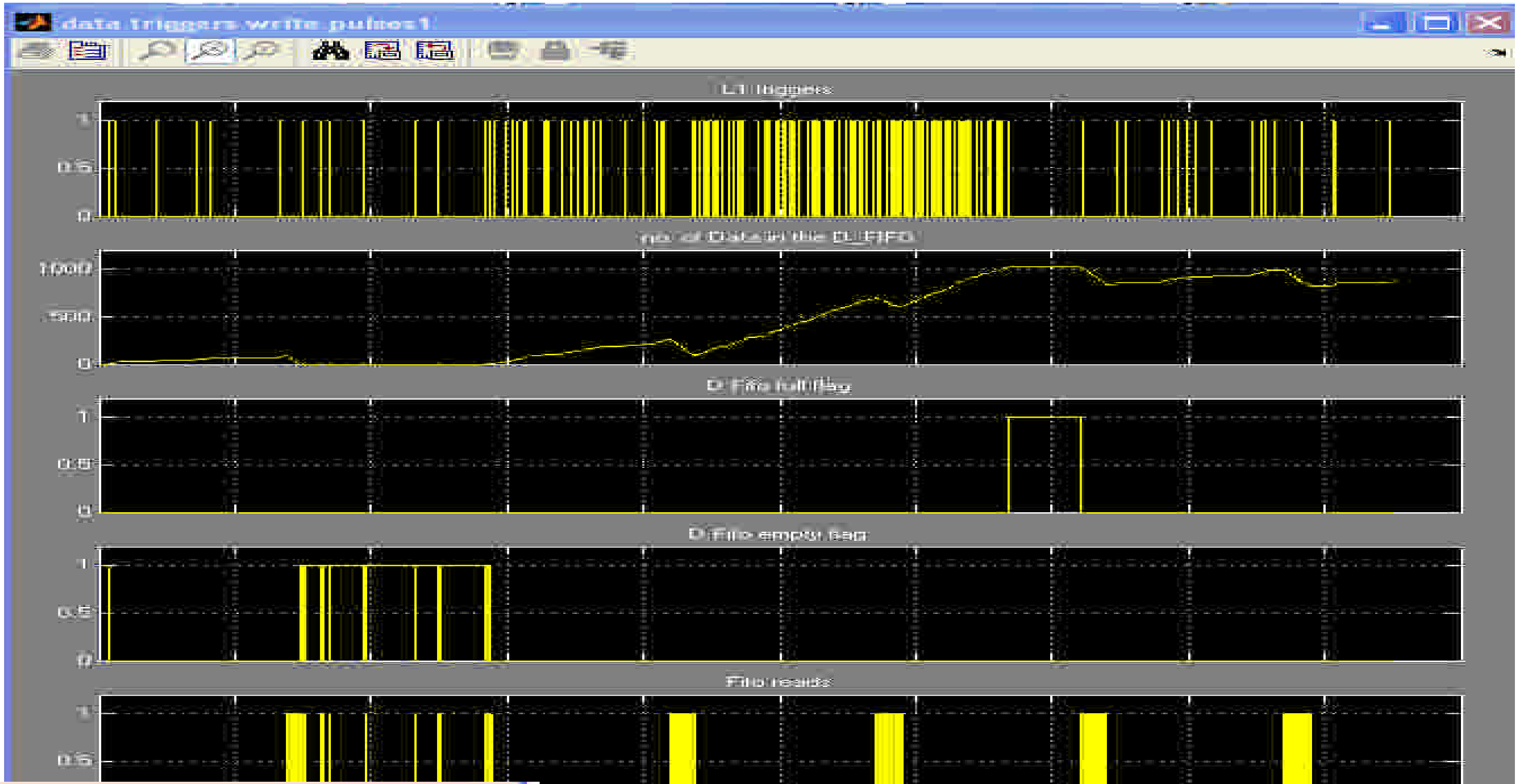


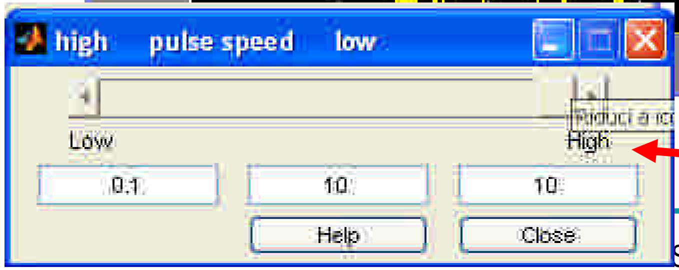
**Band-Limited White Noise:** The Band-Limited White Noise block generates normally distributed random numbers that are suitable for use in continuous or hybrid systems.

# Realtime performance of the simulation

- simulation of each time tick takes ms (depending on the complexity and your CPU)
- this allows interactive modification of design parameters such as
  - pdf of the trigger generation
  - Fifo length
  - CPU speeds
  - Packet length of front end data
  - others

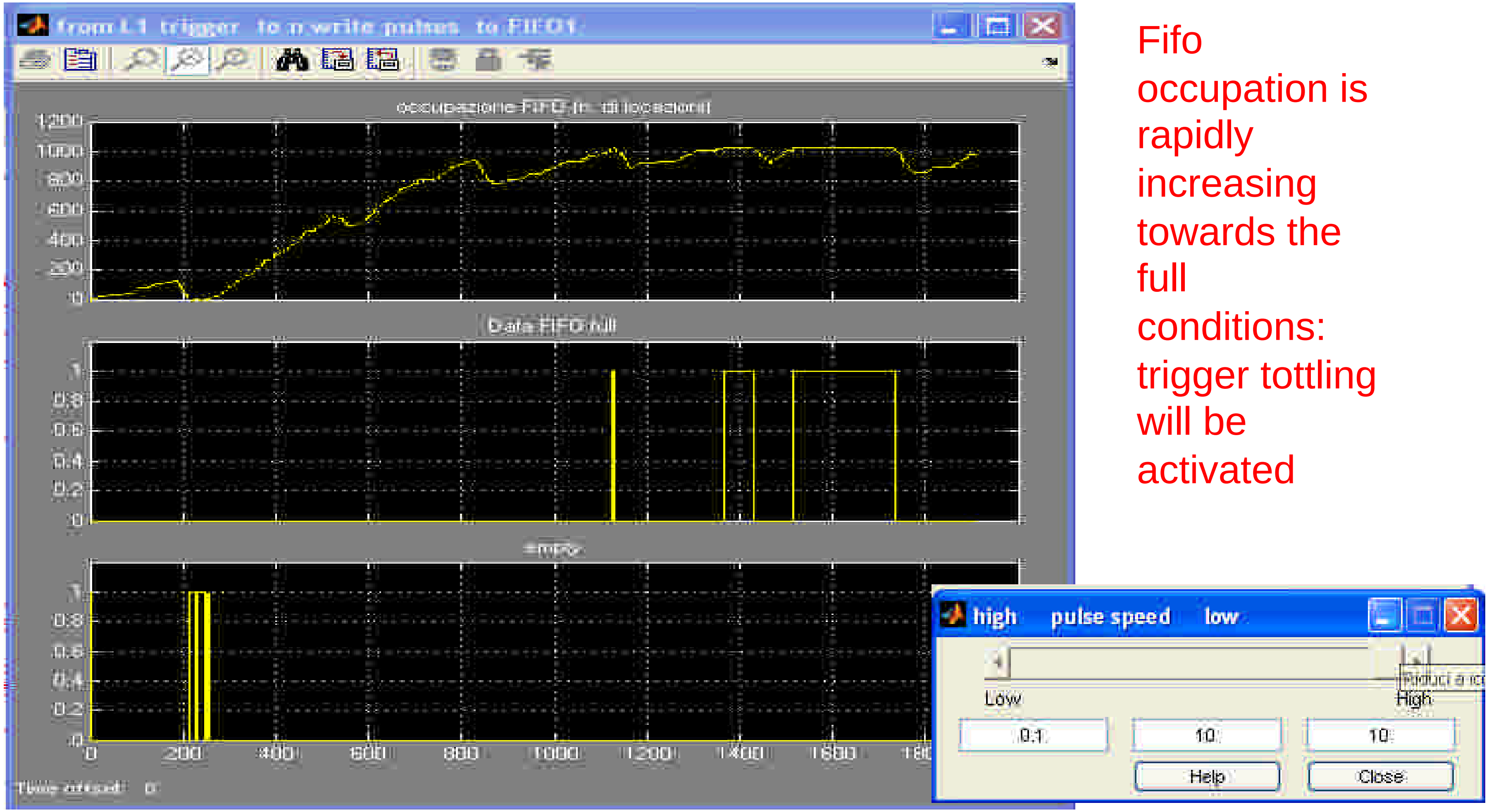
# Around the data fifo





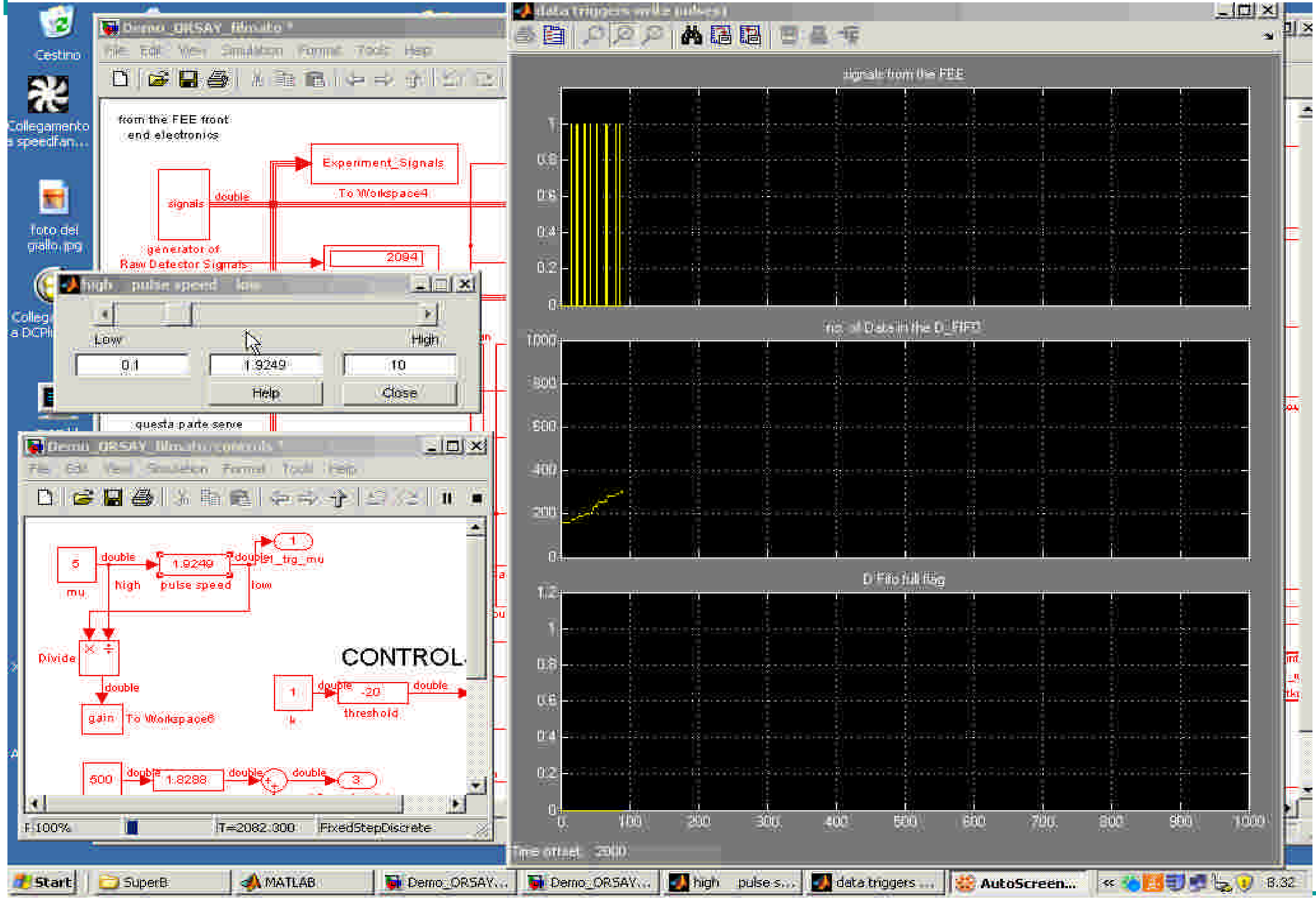
The slider controls the average value of the exponential distribution of the triggers

# Around the data fifo: occupancy, FULL and EMPTY flags

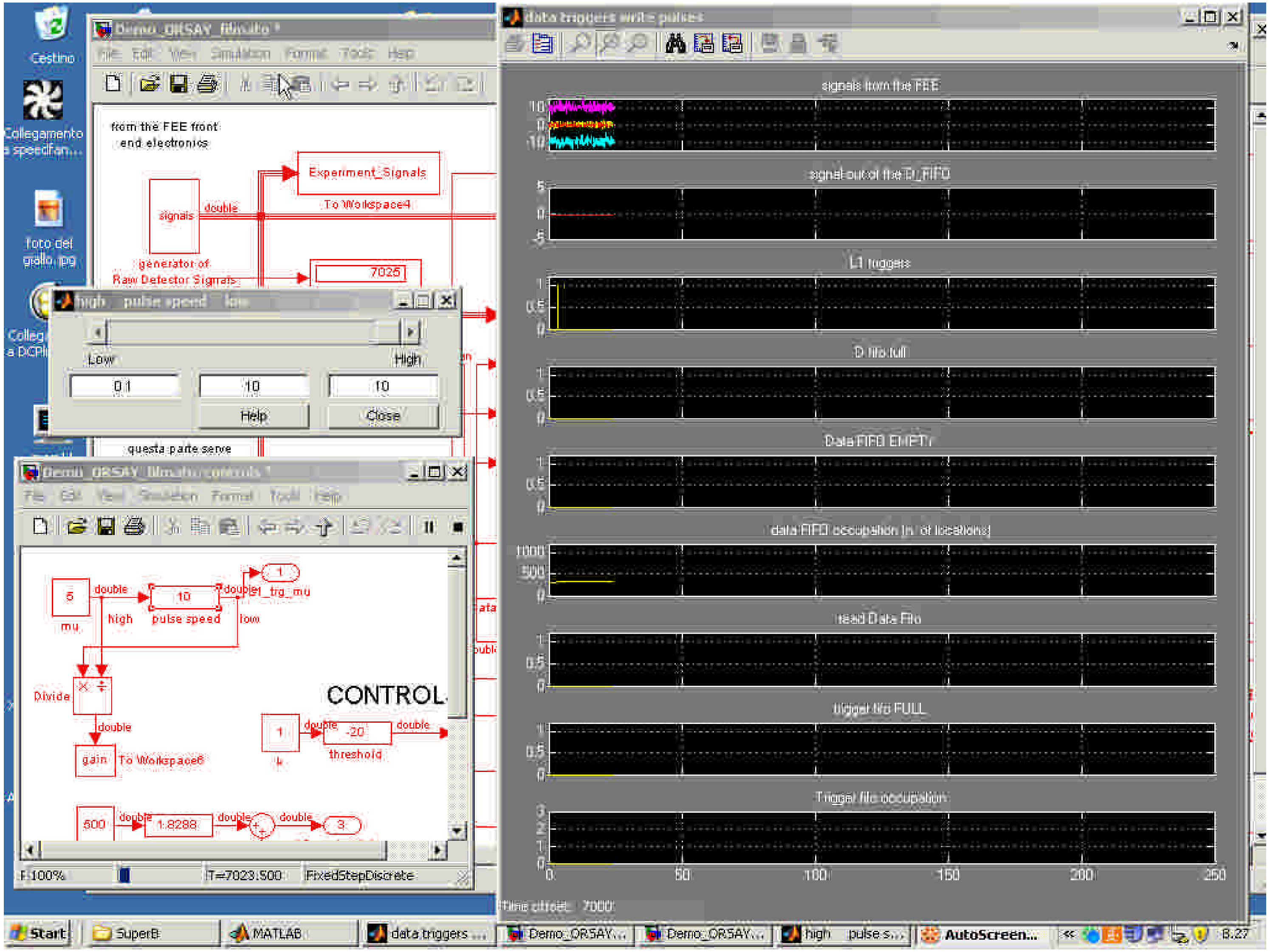


Fifo occupation is rapidly increasing towards the full conditions: trigger tottling will be activated

# The data fifo: occupancy, FULL and EMPTY flags (avi)

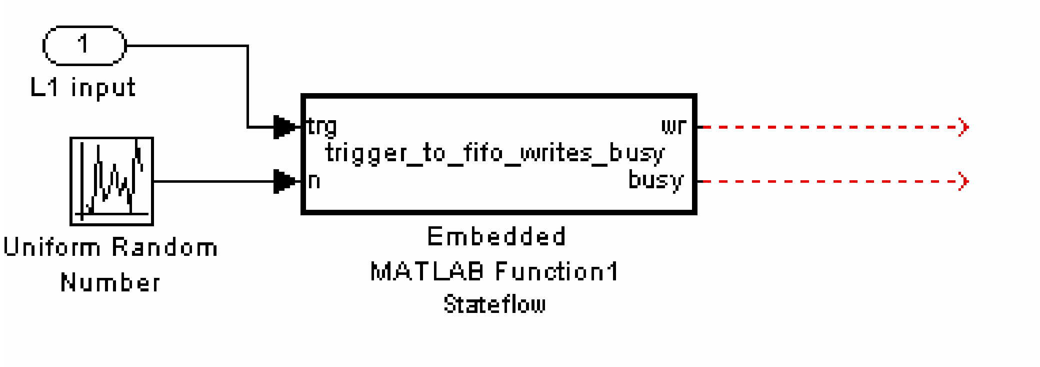


# the data fifo: all related signals (avi)



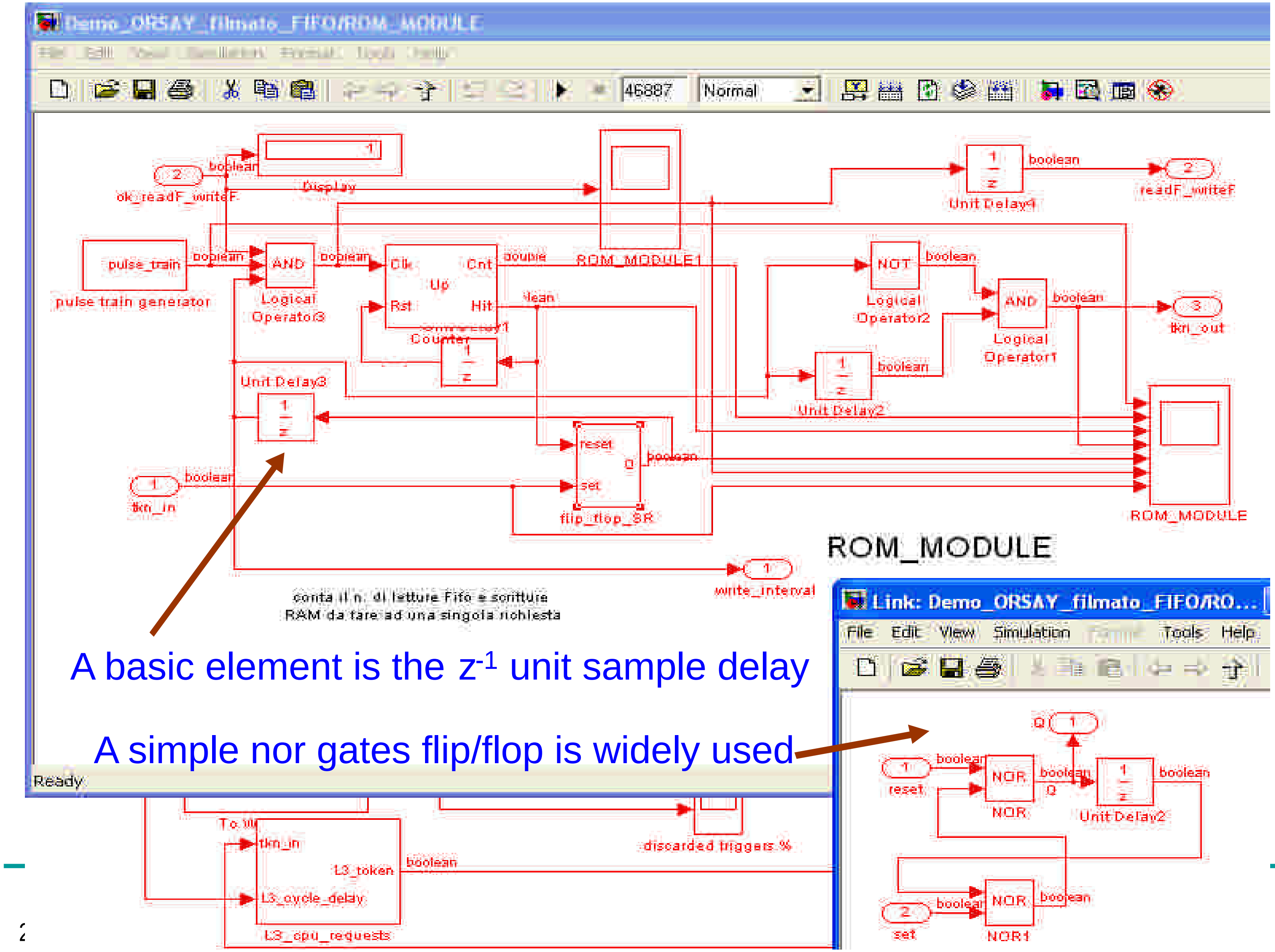
# A block built by means of a piece of software: the burst generator

```
function [wr,busy] = trigger_to_fifo_writes_busy(trg,n)
% BURST GENERATOR
%
persistent stato
values=zeros(1,200);values(2:2:end)=1;
if isempty(stato)
    stato=1;
    busy=1;
end
if trg==0 & stato==1
    stato_inc=0;
end
if trg==1 | stato>1
    stato_inc=1;busy=1;
else
    stato_inc=0;busy=0;
end
stato=stato+stato_inc;
if stato>=2*n+2
    stato=1;
end
wr=values(stato);
```



The burst generator block is driven by a random number generator which chooses at random the number of pulses to be produced according to a given statistical distribution

# Another block: the token passing logic

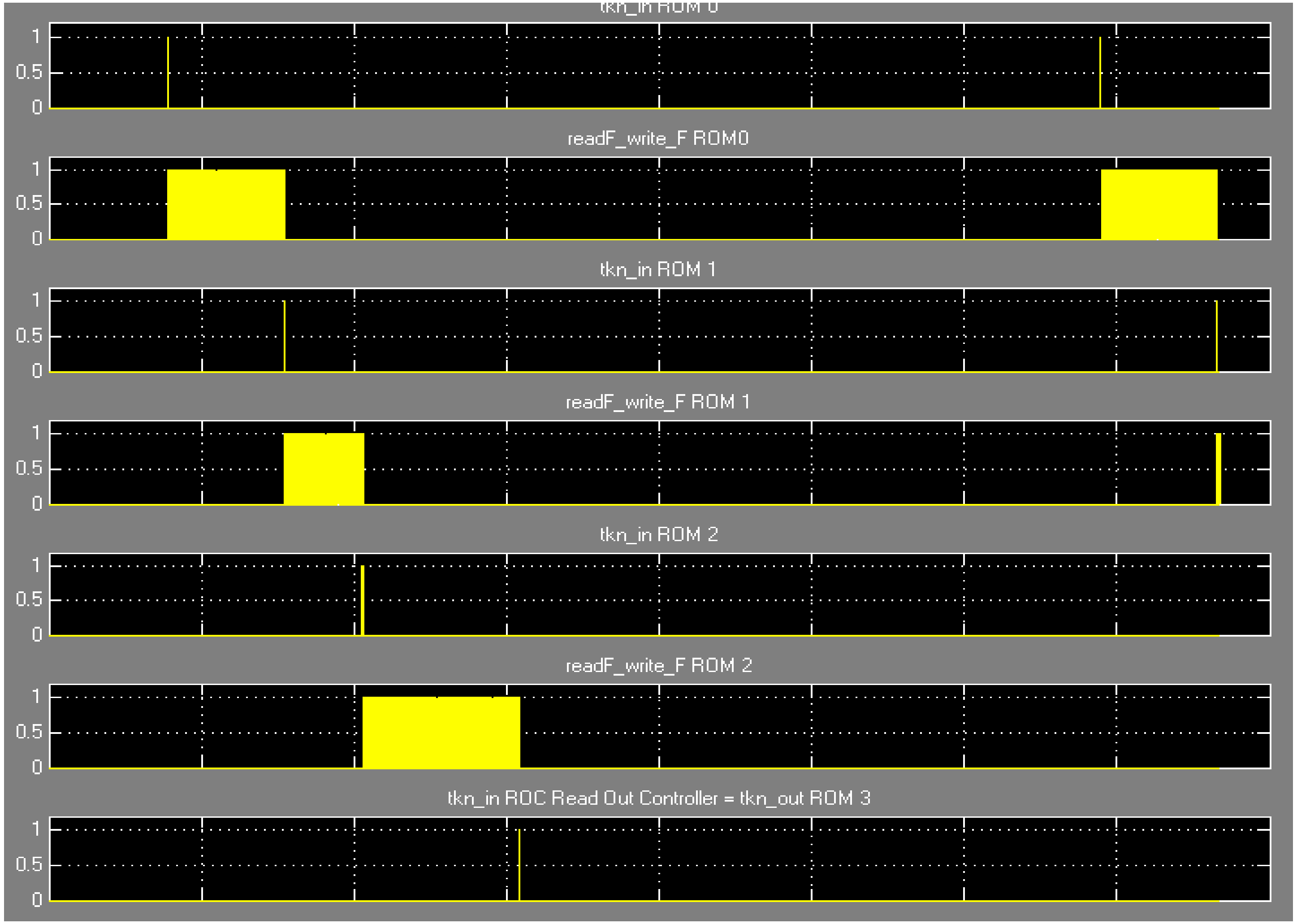


conta il n. di letture Fifo e scritture RAM da fare ad una singola richiesta

A basic element is the $z^{-1}$ unit sample delay

A simple nor gates flip/flop is widely used

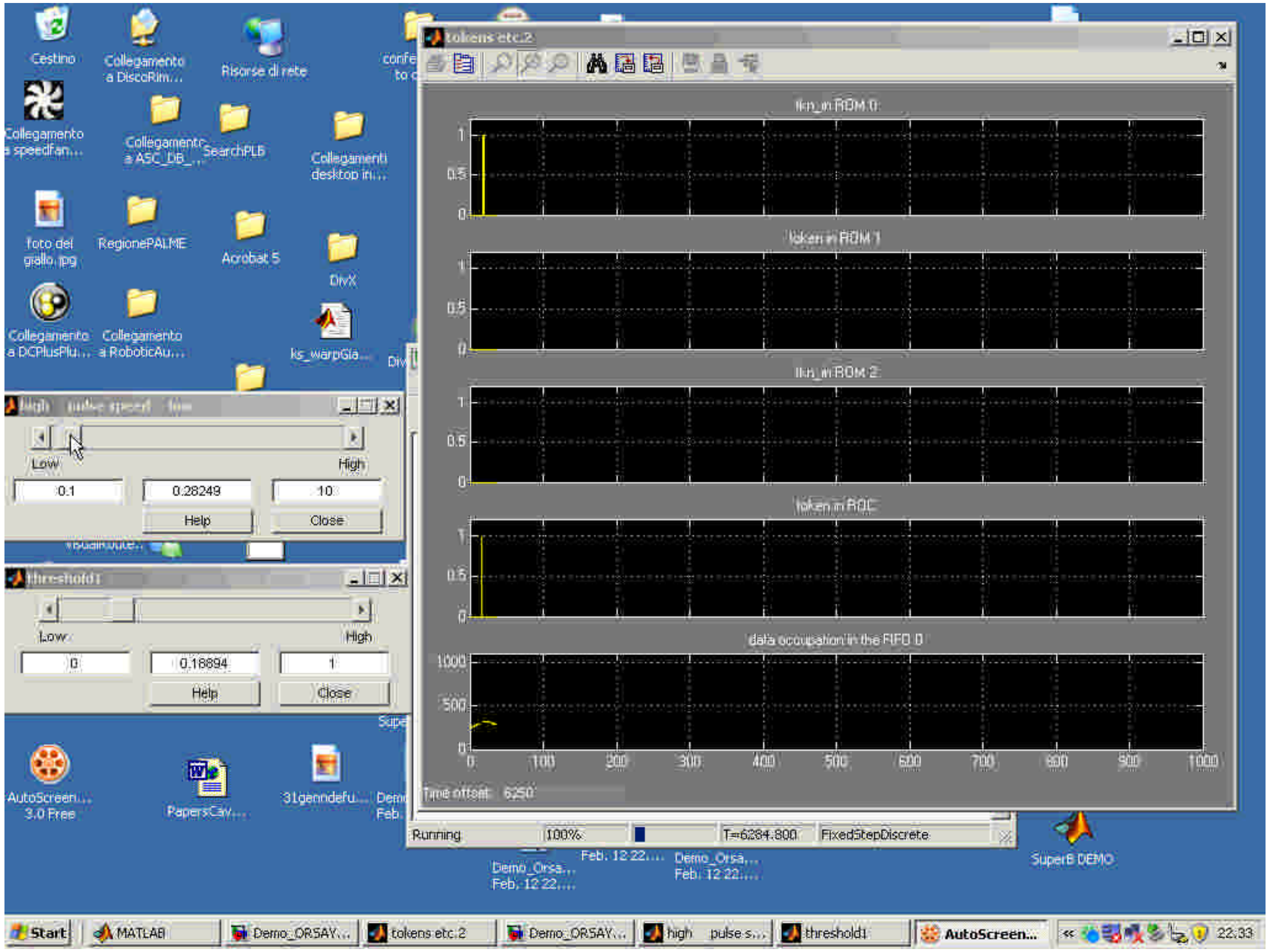
# Simulating the token passing logic



The L3 cpu are periodically involved by the token passing mechanism:

Each block in the ring receives the tkn_in signal and provides the tkn_out to the next in the chain; time required for each transaction is different for each block

# Simulating the token passing logic (avi)



high pulse speed low

Low 0.1 — 0.28249 — High 10

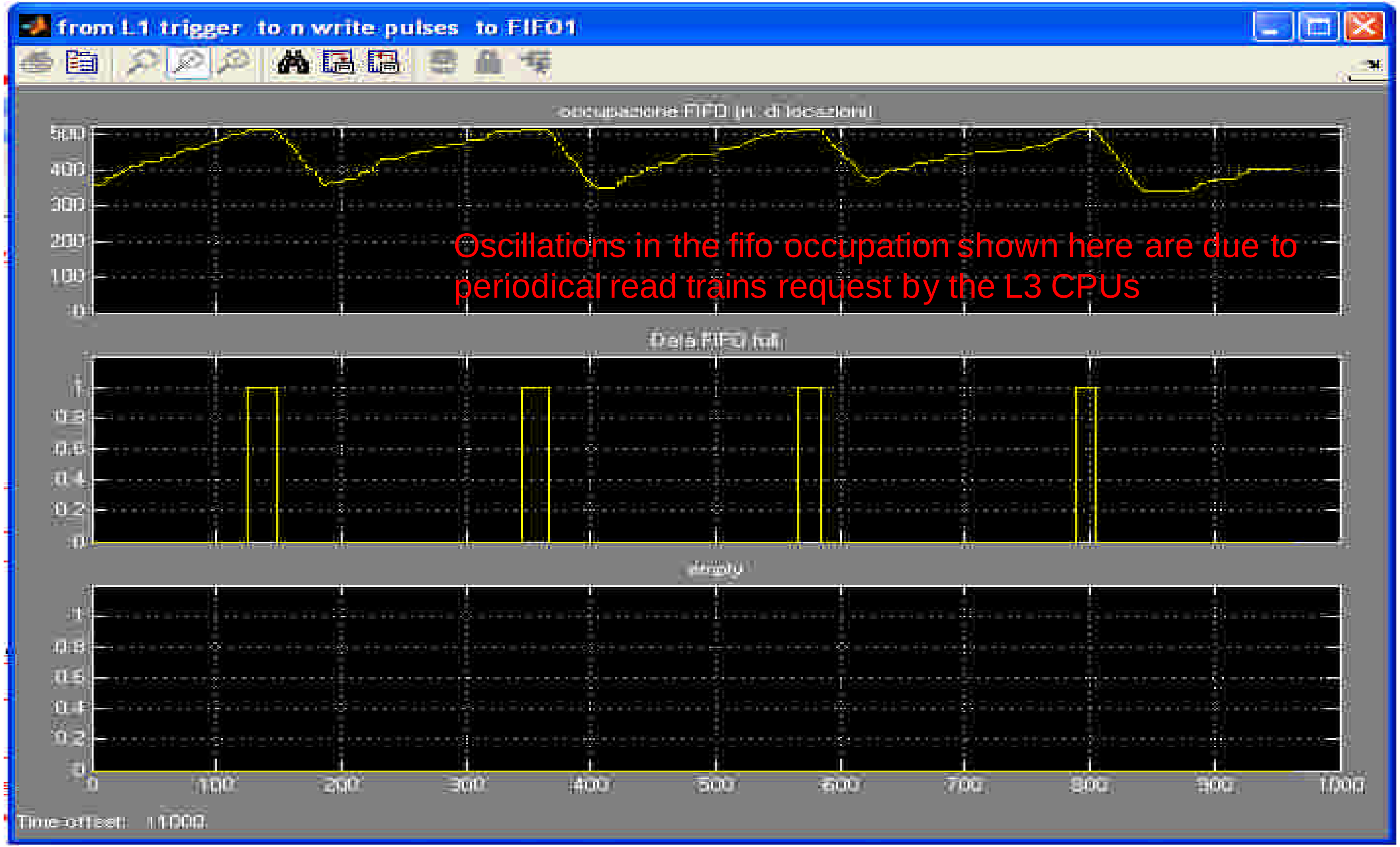
threshold1

Low 0 — 0.18894 — High 1

Running 100% T=6284.800 FixedStepDiscrete

Time offset: 6250

# Data fifo during the simulation



Oscillations in the fifo occupation shown here are due to periodical read trains request by the L3 CPUs

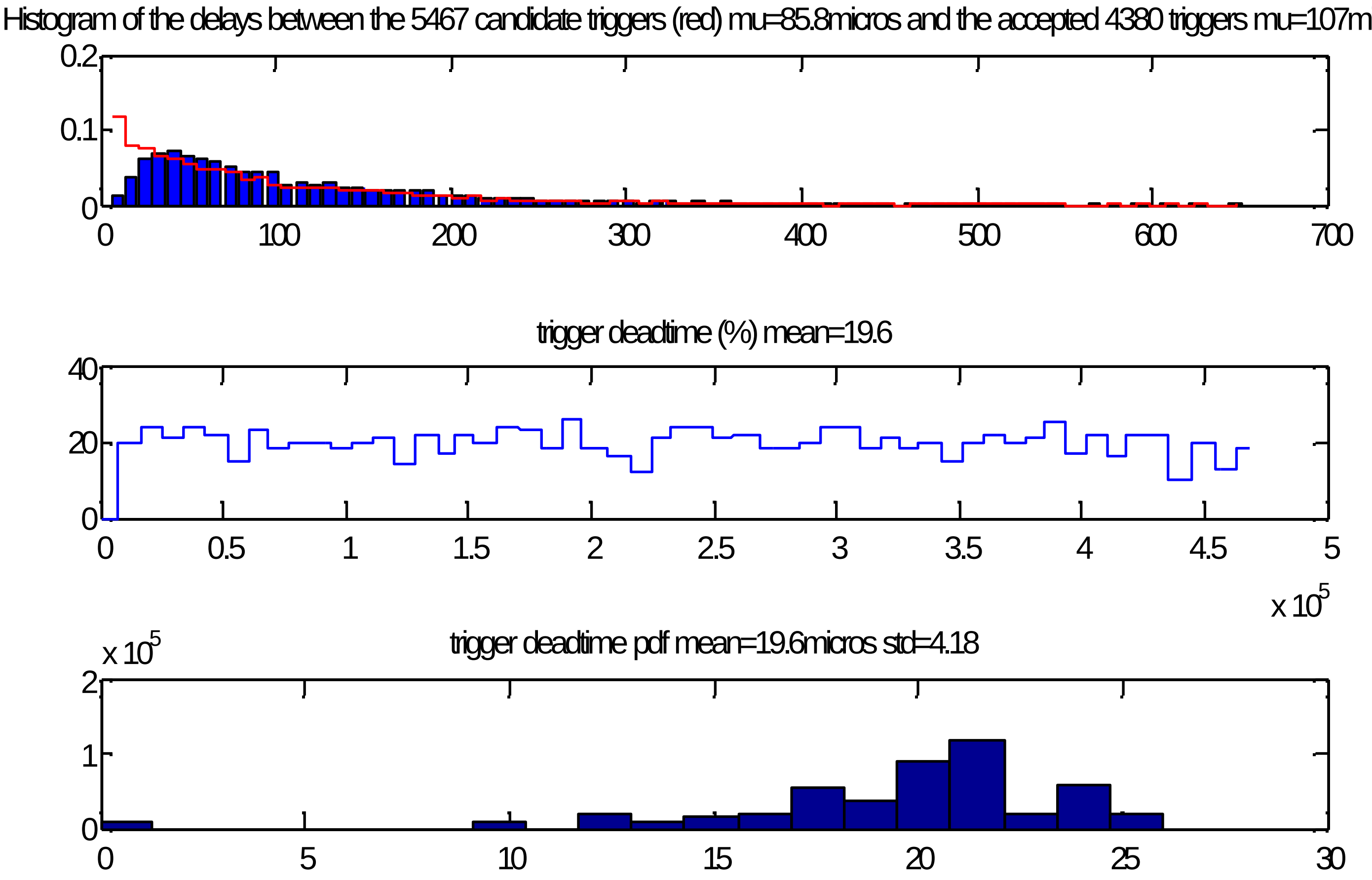
# Some figures from the simulations

- simulation time= 1.6e+3s equivalent to 3.37ms per sample
- Fifo length is 256
  - Miss percentage = 33.3%
- Fifo length is 512
  - Miss percentage = 31.6% mean value= 23.9 sigma= 6.73
- Fifo length is 1024
  - Miss percentage = 30.7% mean value= 23 sigma= 6.53

- Faster L3 cpu:
- Fifo length is 1024
  - Miss percentage = 23.5%

# Off-line data analysis in Matlab

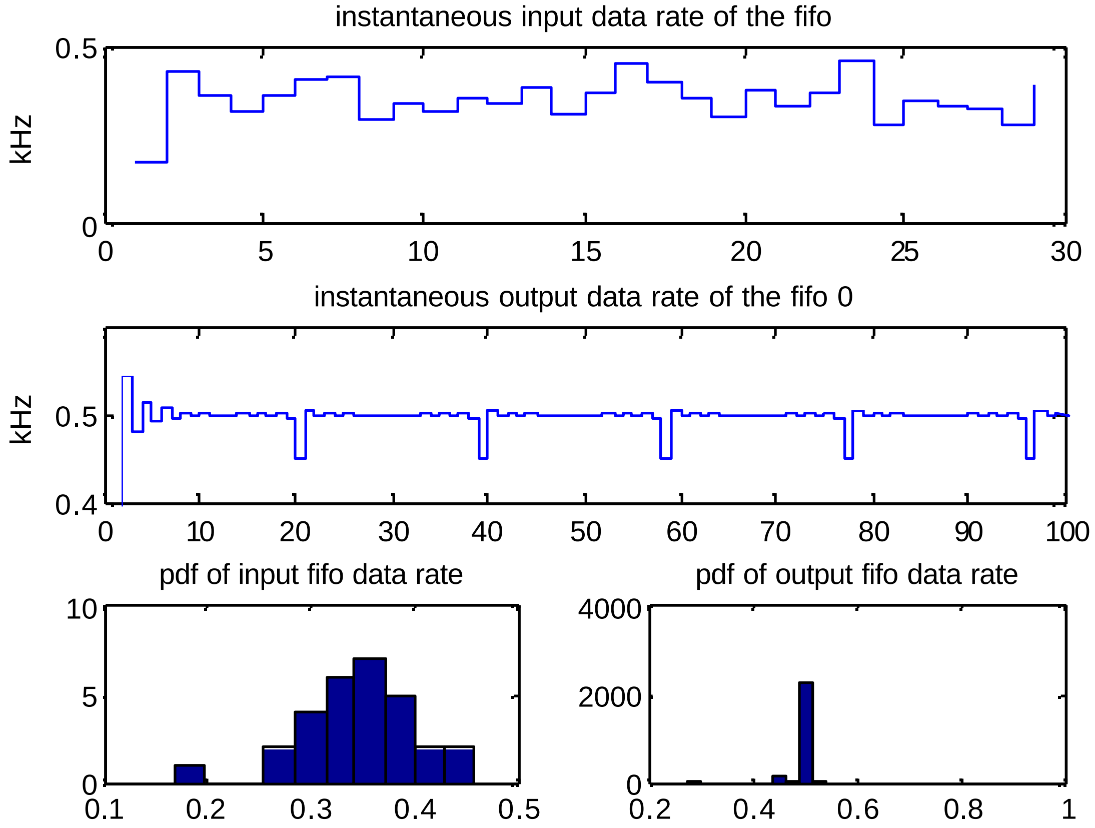
During the simulation you may log some data, so that you will be able to analyse what happened, also from a statistical viewpoint



Here we are obtaining the histograms of the trigger request issued and of the trigger accepted by the logic.

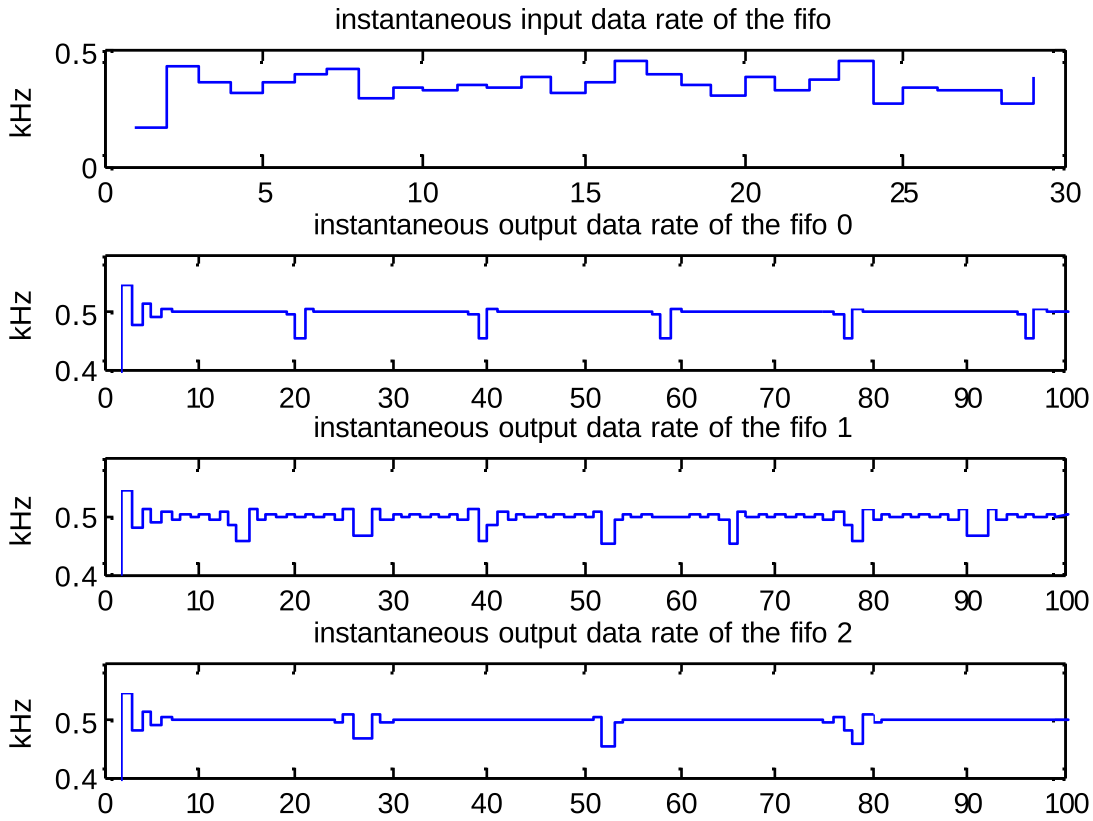
Some are in fact rejected because of trottling

# Off-line data analysis in Matlab: input and output data rate for a data fifo



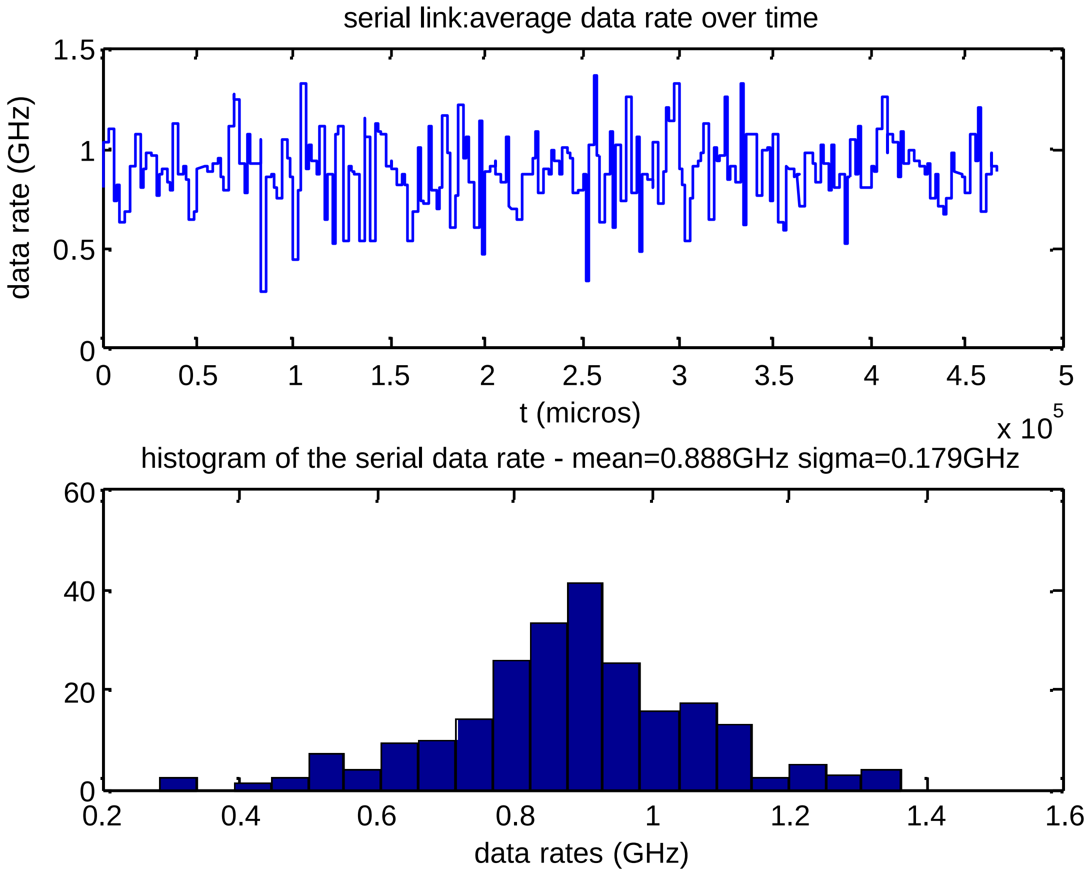
Input data rate is random, while the output data rate shows the periodic feature due to the periodic nature of the token ring mechanims.

# Off-line data analysis in Matlab: input and output data rate for the data fifos



Output data rates for different blocks in the ring are mostly periodic but obviously out of phase each to the others

# Off-line data analysis in Matlab: the serial link



We may analyse both the statistics of the traffic on the serial link including both average value and its standard deviation but also its time domain evolution

# Conclusion and developments

Features of the proposed approach are:

- very high level behavioural simulation of the architecture under development
- Enable the system architect to explore easily different what-if scenarios
- Very long time frames simulation (days?)
- Obtain figures reagarding the performance of the hardware in terms of average values but also their distribution in time and their statistical distribution

What may be done in the next:

- obtain from the physics simulations all data relevant to better model the hardware and check the performance (eg. Trigger rates, its statistical distribution, dead times and so on)
- Set up a detailed simulation of the DAQ chain as it is being developed and will be developed in the next
- Actually we are going to use the tool in order to carry on intensive performance analysis of the serial links under development in Naples as used in a complex acquisition chain
- THANK YOU