

## MROD-0 : MROD design study and demonstrator

Jos Vermeulen, NIKHEF, September 12, 1999

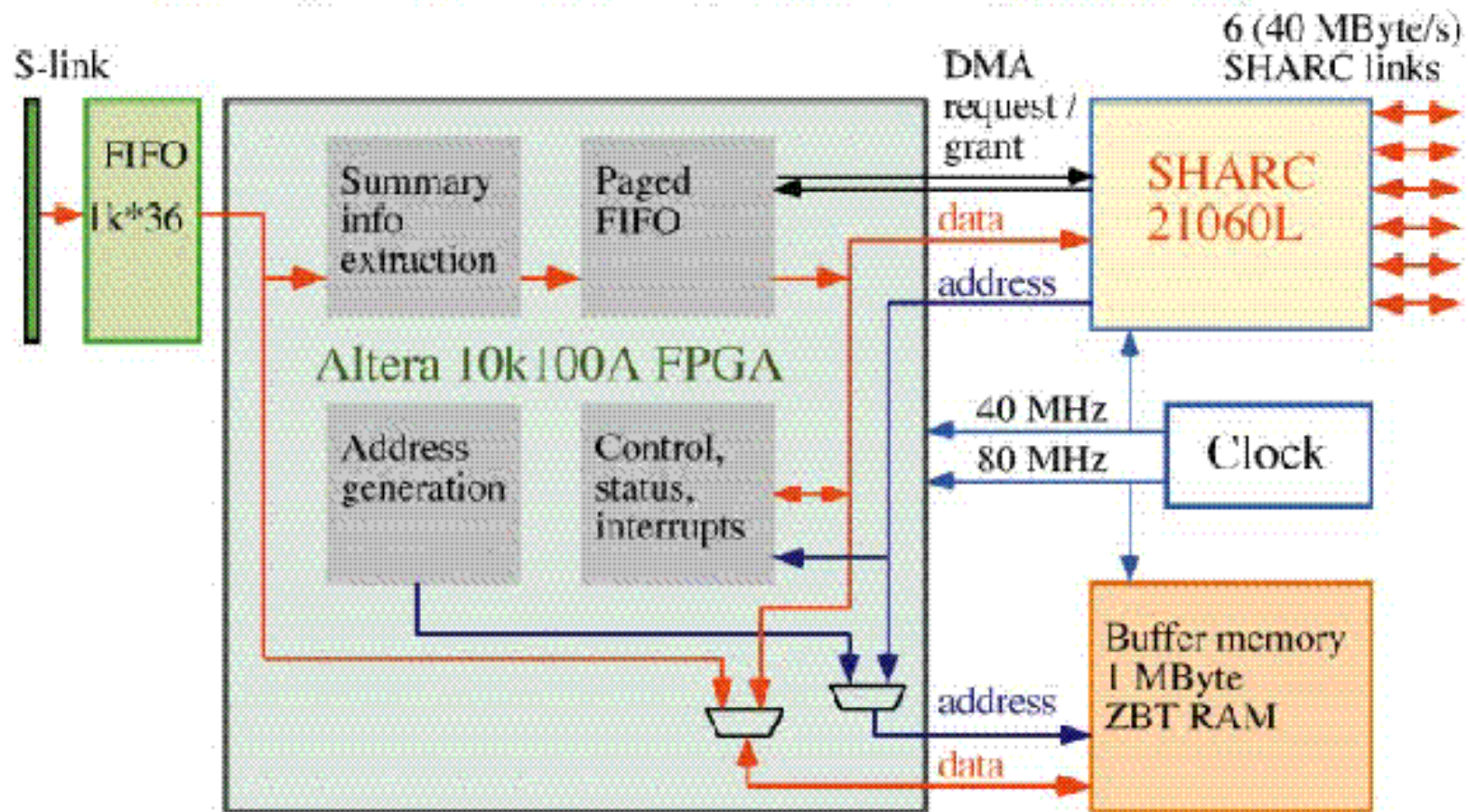
Reminder :

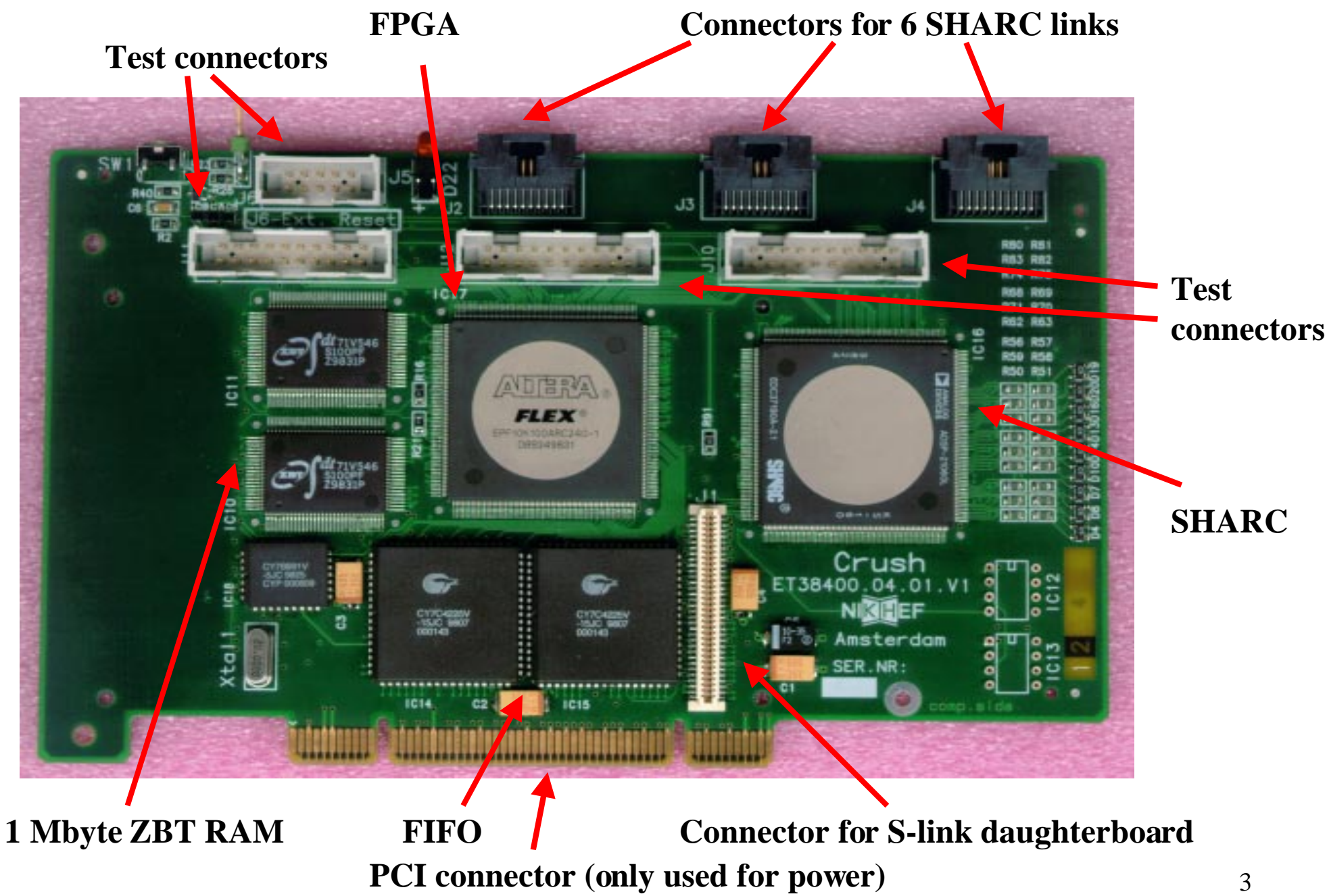
M-CRUSH : **existing** CRUSH (ROBIn) hardware, but with **reprogrammed FPGA**.

Goal : learn about unforeseen difficulties and achieve knowledge required for finding a good compromise between minimization of cost and functionality / flexibility.

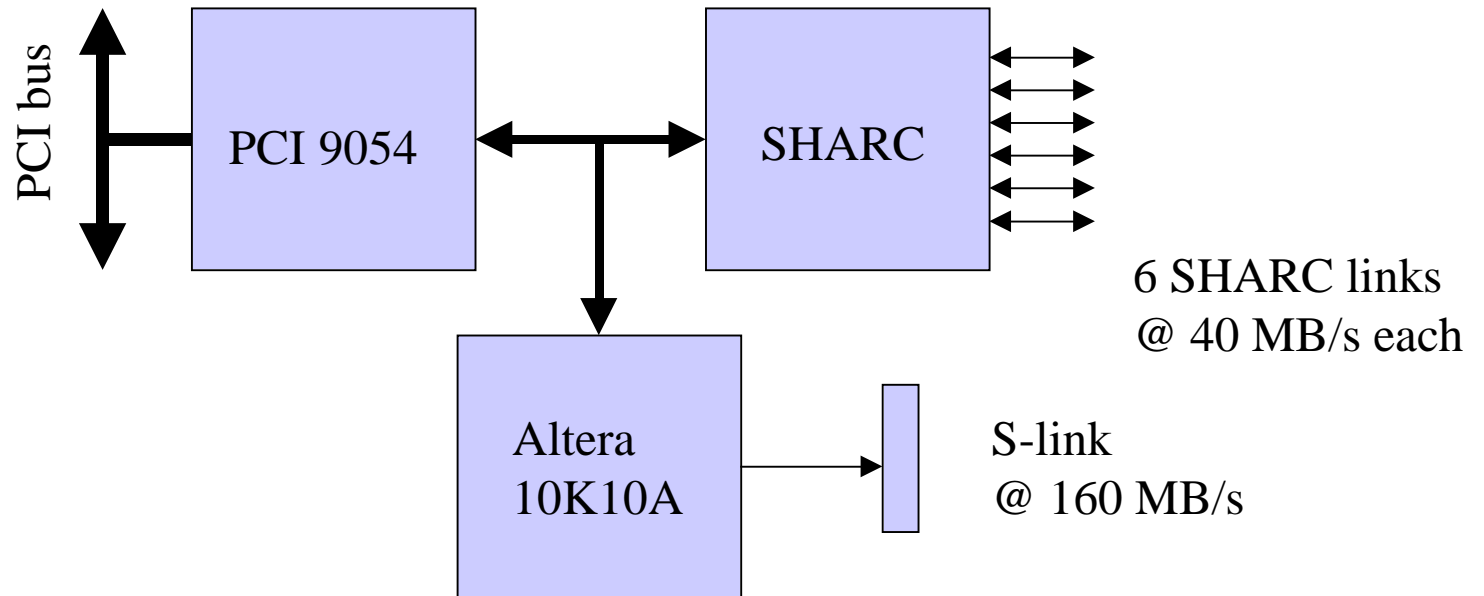
*Use ShaSLINK for output to S-link AND as programmable data generator for S-link input of M-CRUSH*

## Compact ROB Using a SHARC (CRUSH)





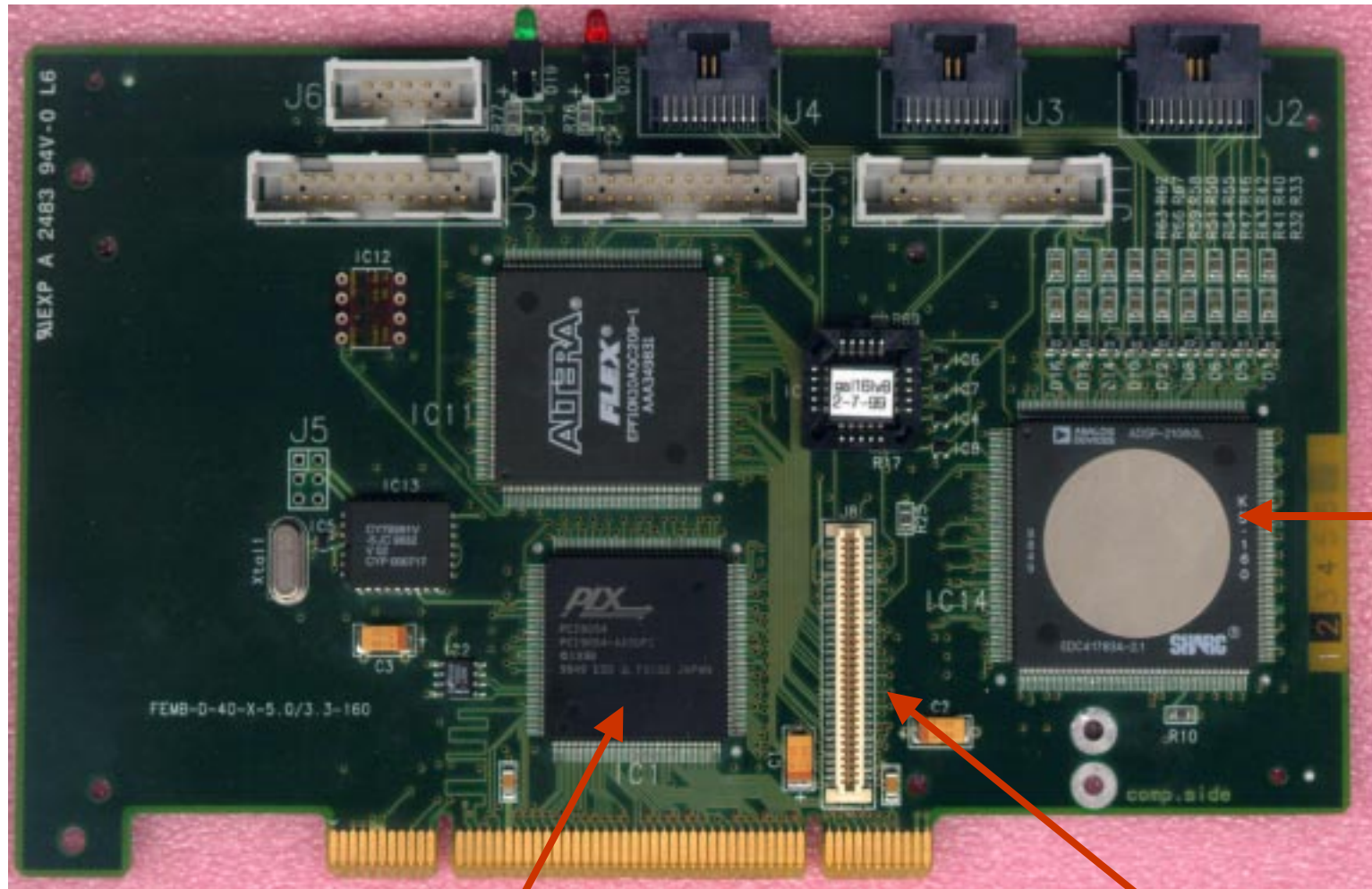
## SHaSLINK



5 boards produced and tested, all fully functional, work on support software (library for SHARC + server program + driver for PC) in progress

# ShaSlink

Connectors for 6 SHARC links



SHARC

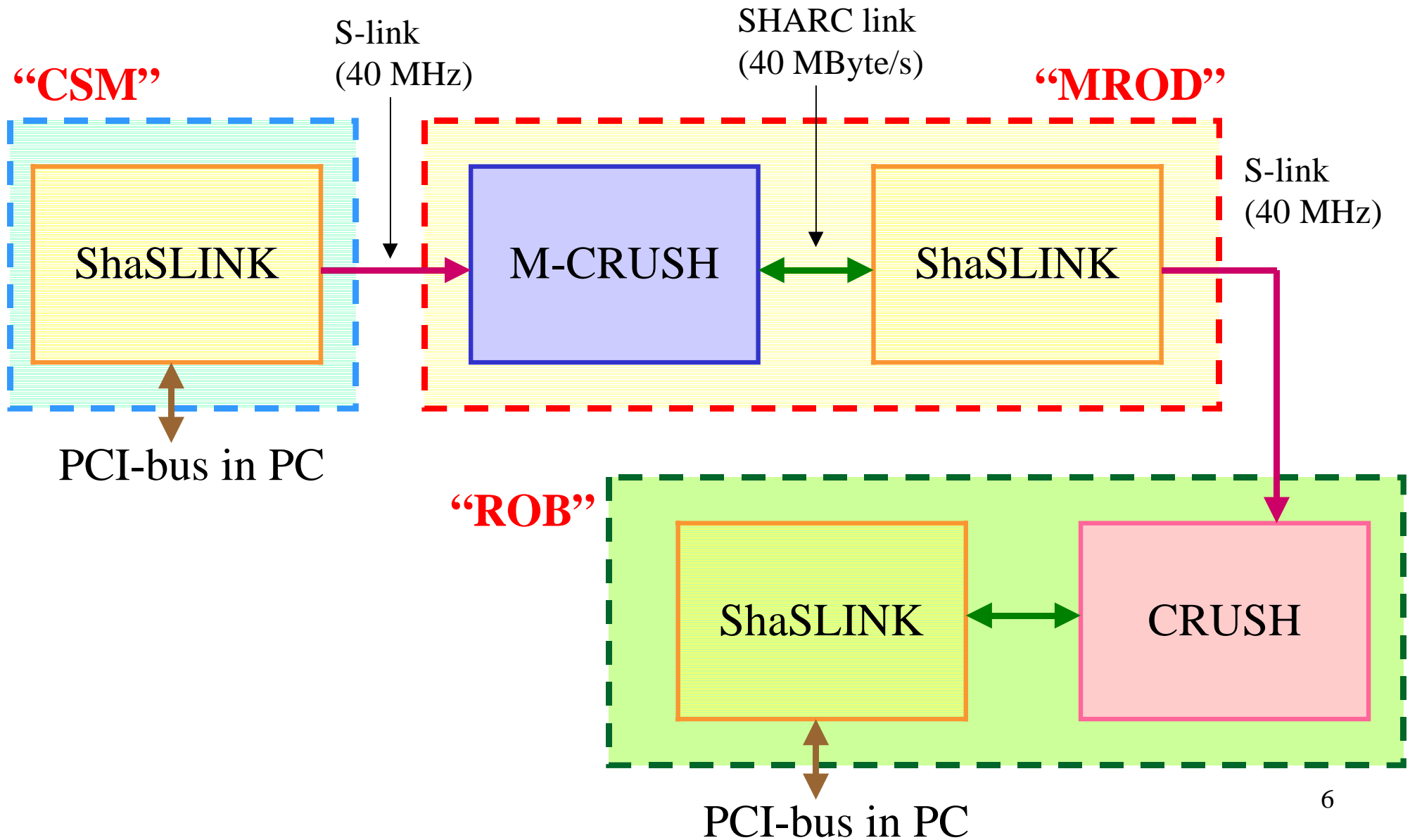
*Measured!*

PCI interface (PLX 9054), supports full PCI bus speed transfers from SHARC memory

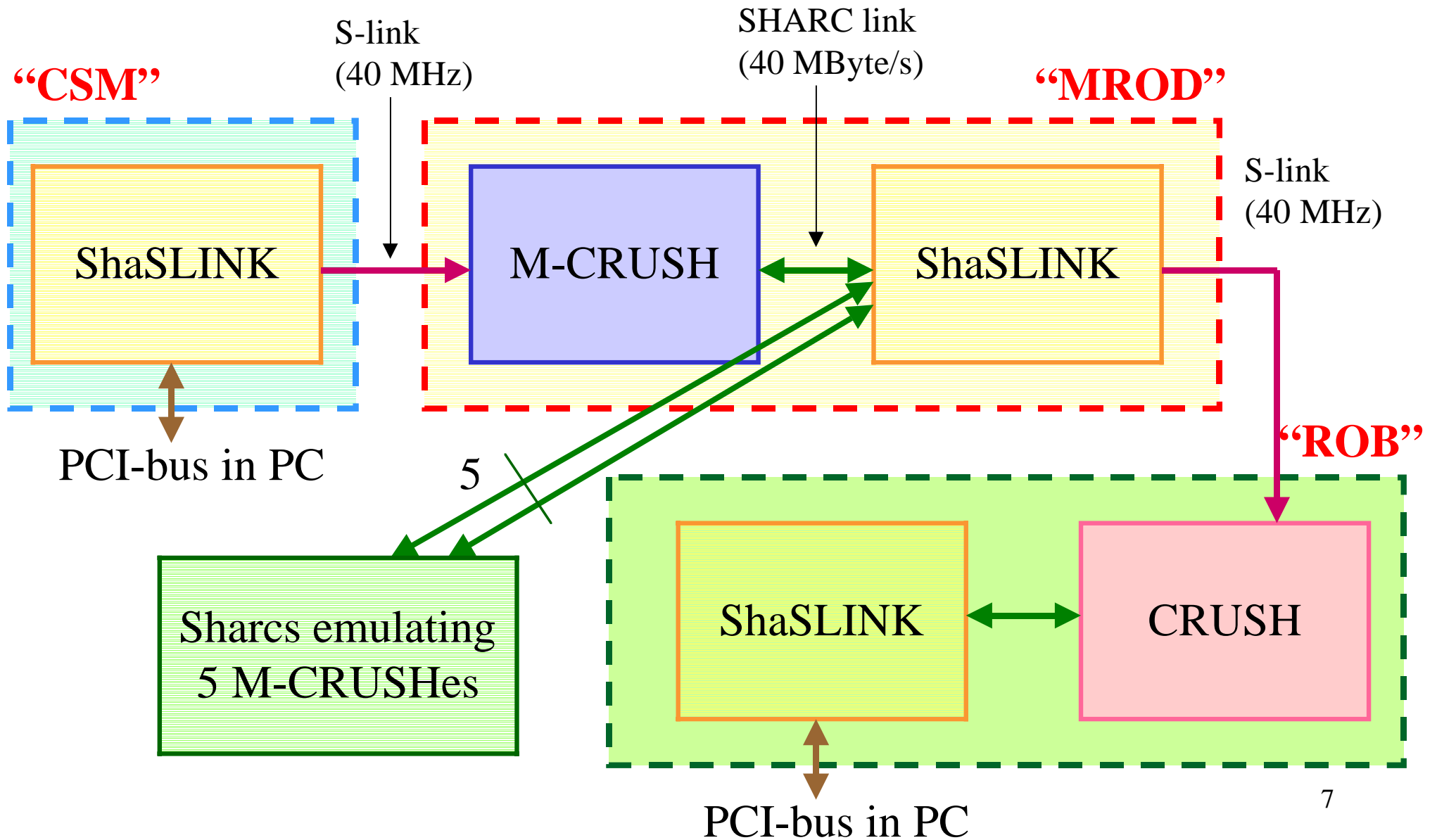
S-link connector for 160 MByte/s output S-link

*Measured!*

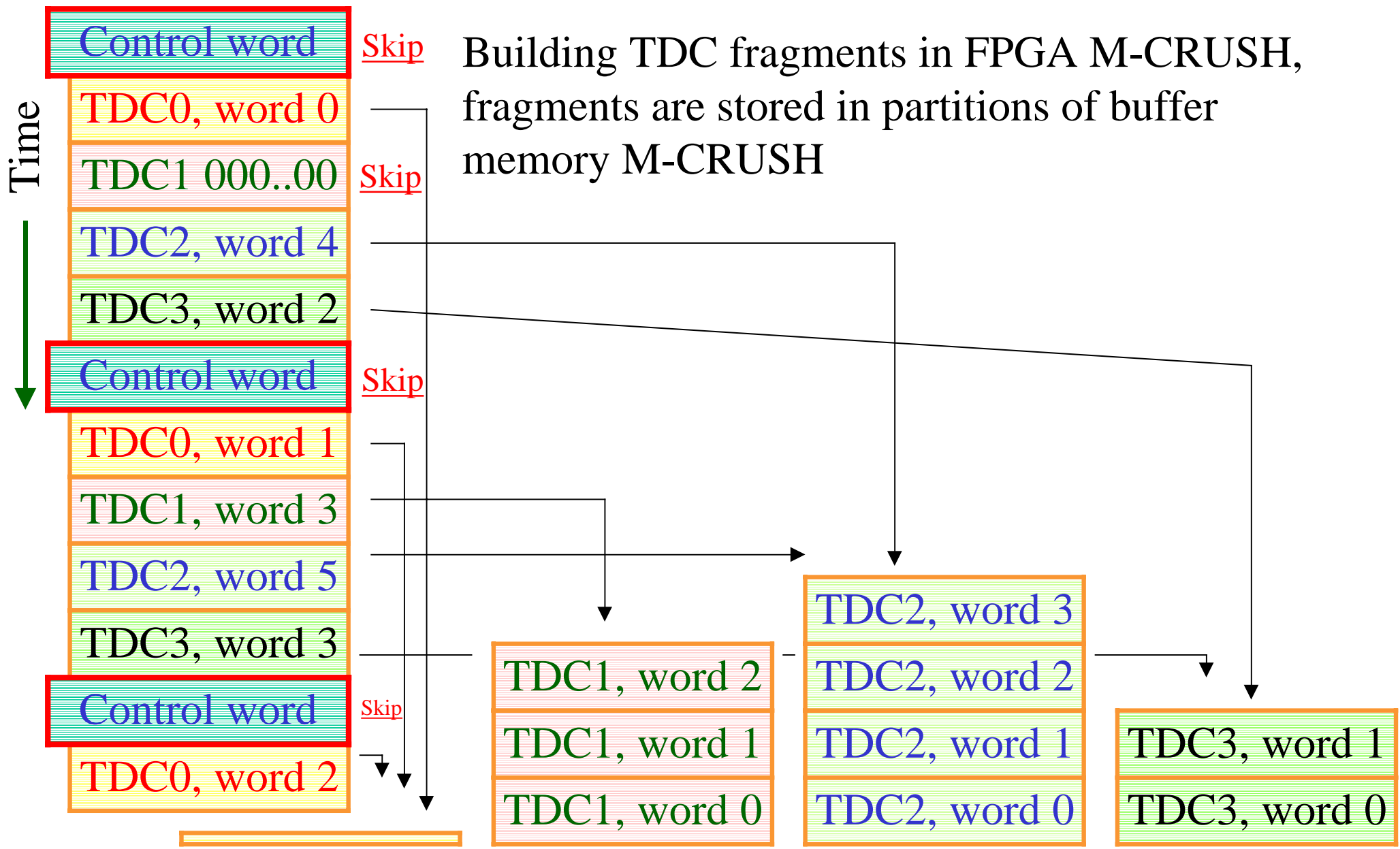
# Test set-up MROD-0



# Test set-up MROD-0, 6 input link test



Building TDC fragments in FPGA M-CRUSH, fragments are stored in partitions of buffer memory M-CRUSH



Position in sequence determines TDC id, can check against TDC id in data

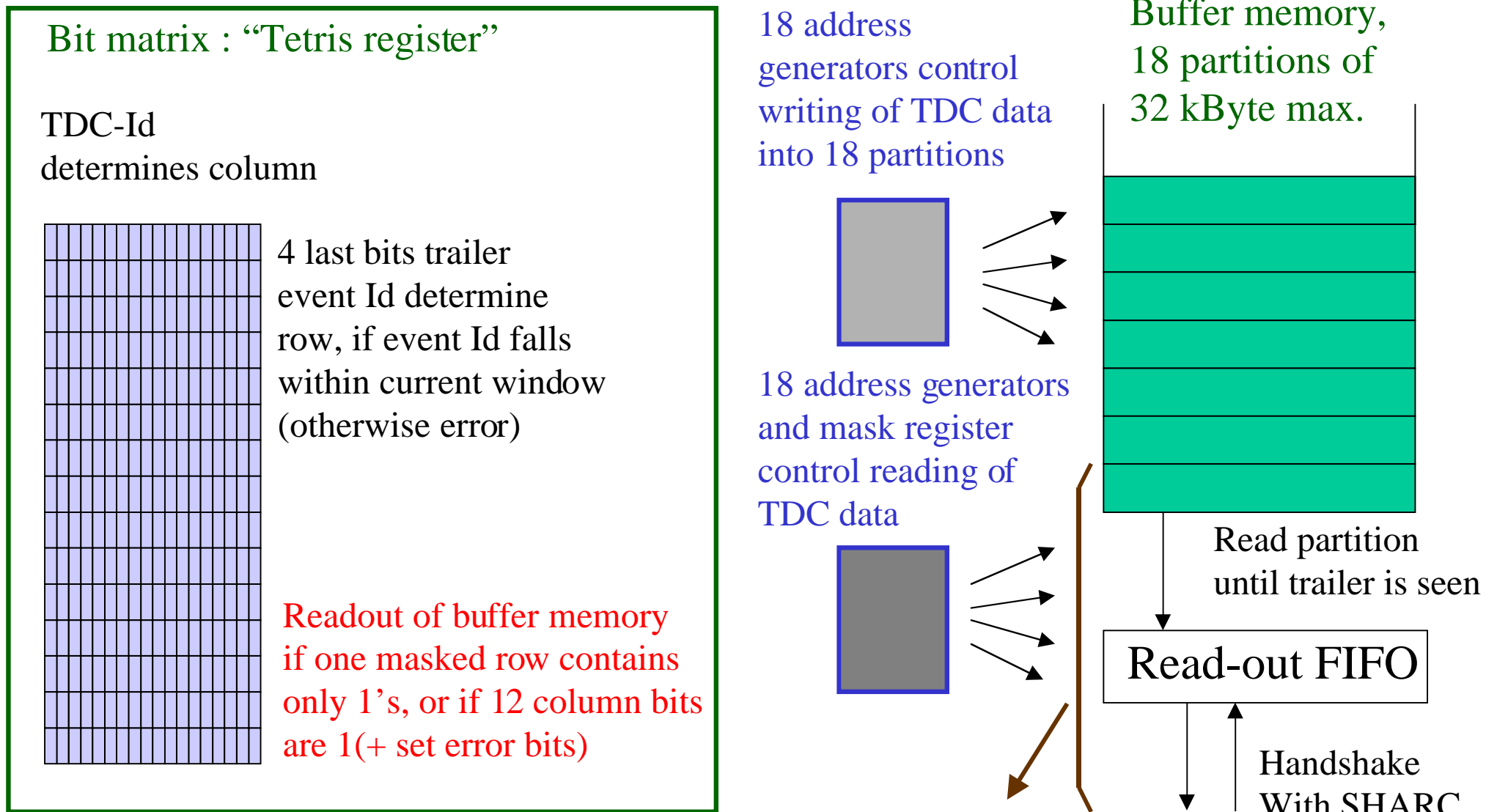
NB : Control words could be used to send TDC parity bits

## M-CRUSH readout organisation (1)

- All incoming data are stored in the buffer memory, trailer id's control setting of bits in "Tetris register"
- Normal operation : readout of data out of buffer memory only when all trailers have been seen, in that case all data is available in the buffer memory
- Errors that can disturb operation :
  1. corruption of trailer words,
  2.  $> 12$  difference in event Id's : unlikely (simulation study)

Both cases are handled by incomplete readout. Data not readout for a certain TDC may be read out later together with data from later event fragments (signalled with error bits), when read-out for a later event id is signalled by Tetris register. Reading out of data of each individual TDC can be shut off under control of the SHARC. *The SHARC can also stop the complete read-out and then has random-access to the buffer memory for error checking.*

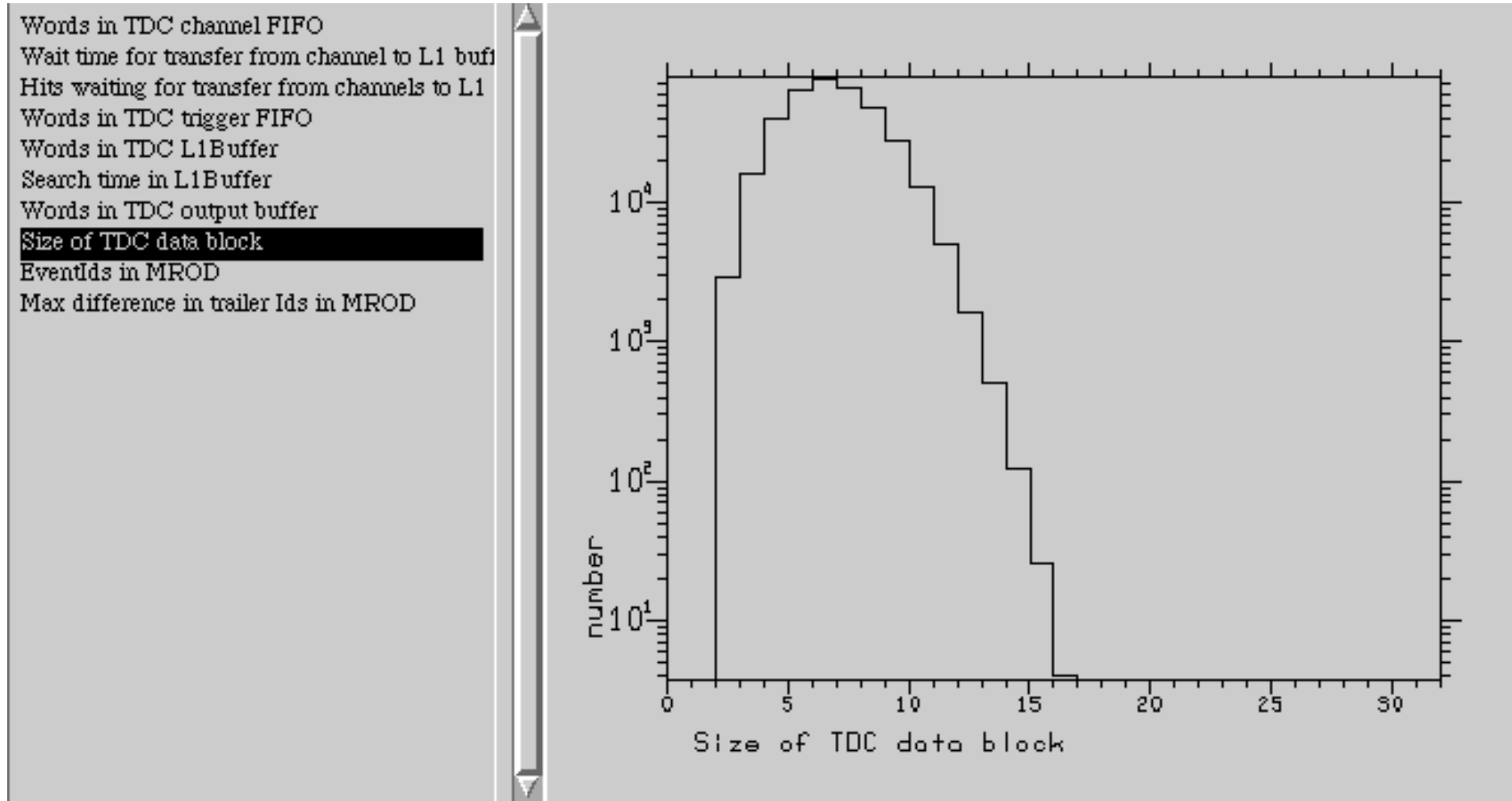
## M-CRUSH readout organisation (2)



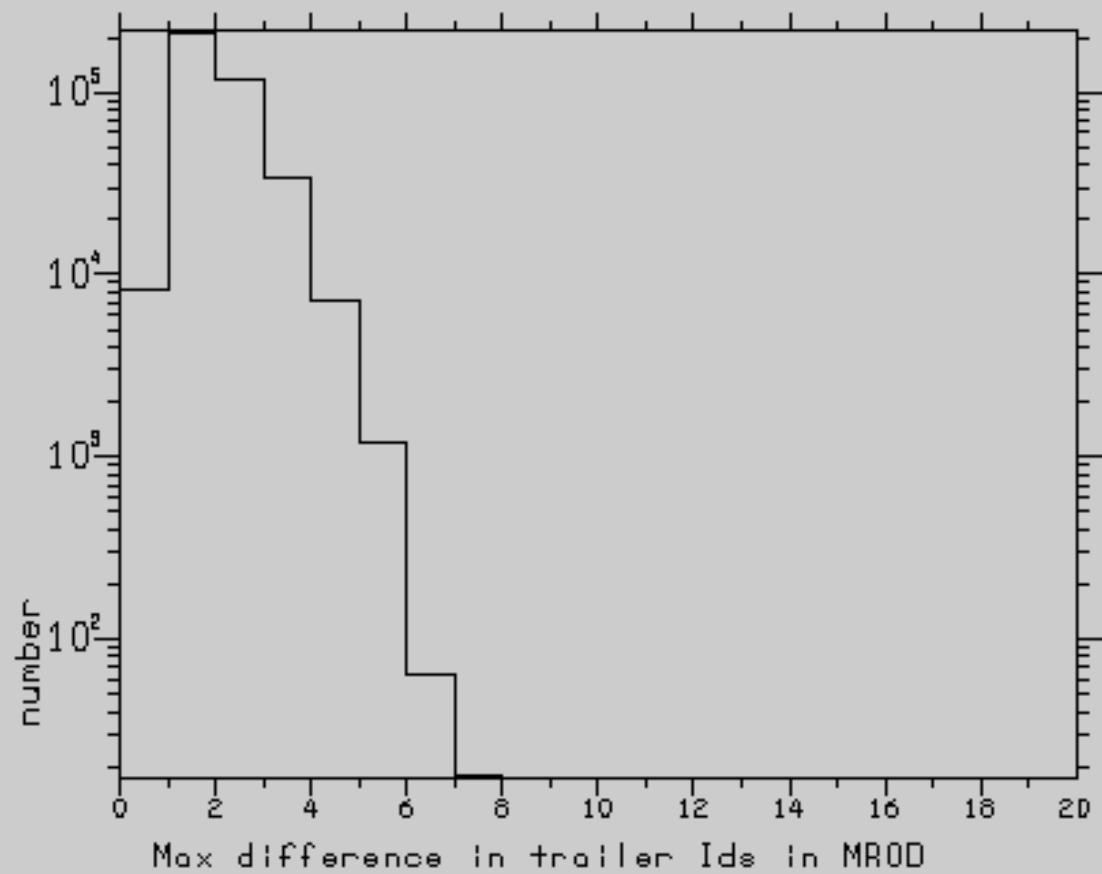
“Tetris register” + other circuitry have been simulated on FPGA simulator

*This part has been implemented and tested in the CRUSH, transfer speed : 80 MByte/s*

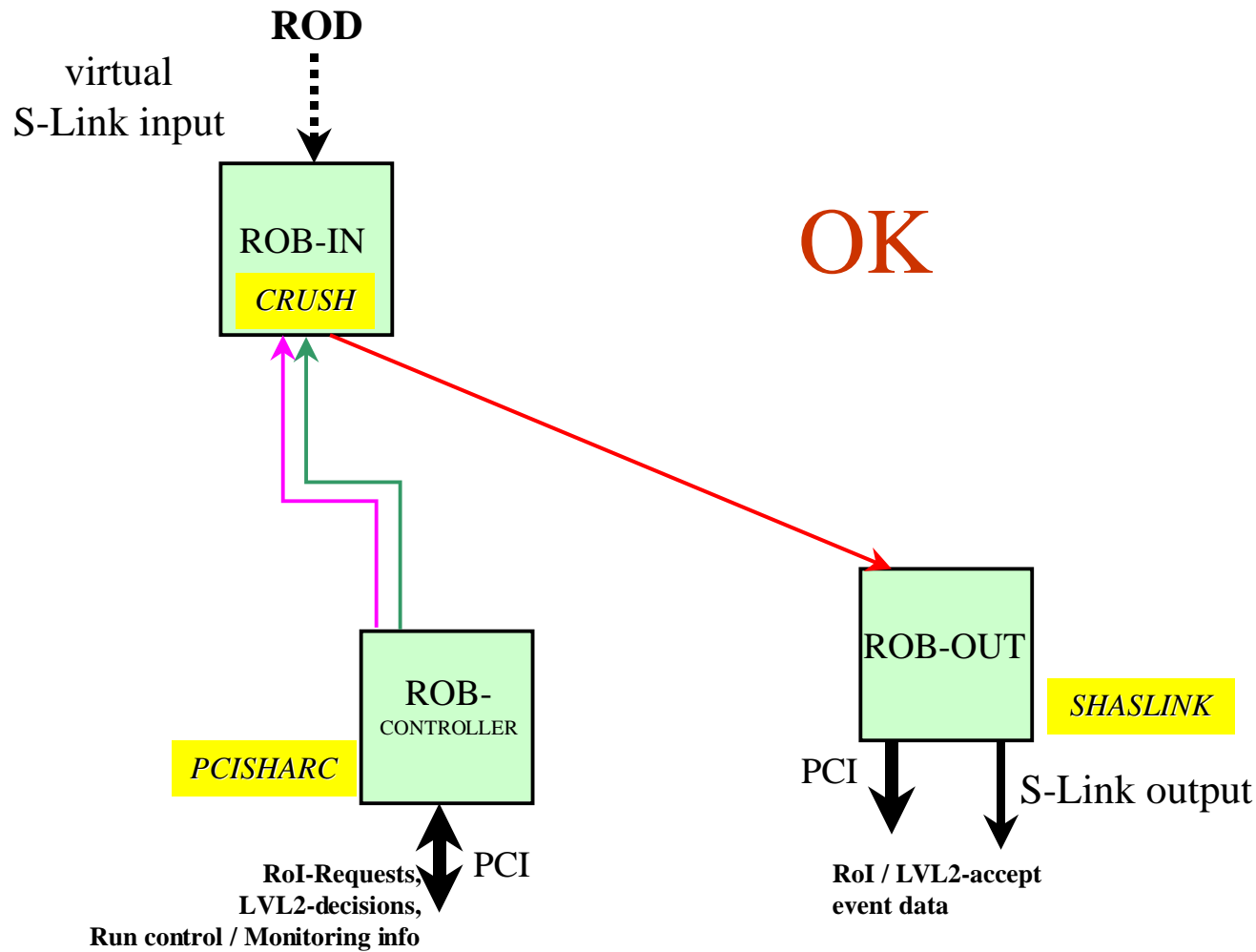
# First simulation results



Words in TDC channel FIFO  
Wait time for transfer from channel to L1 buff  
Hits waiting for transfer from channels to L1  
Words in TDC trigger FIFO  
Words in TDC L1Buffer  
Search time in L1Buffer  
Words in TDC output buffer  
Size of TDC data block  
EventIds in MROD  
**Max difference in trailer Ids in MROD**



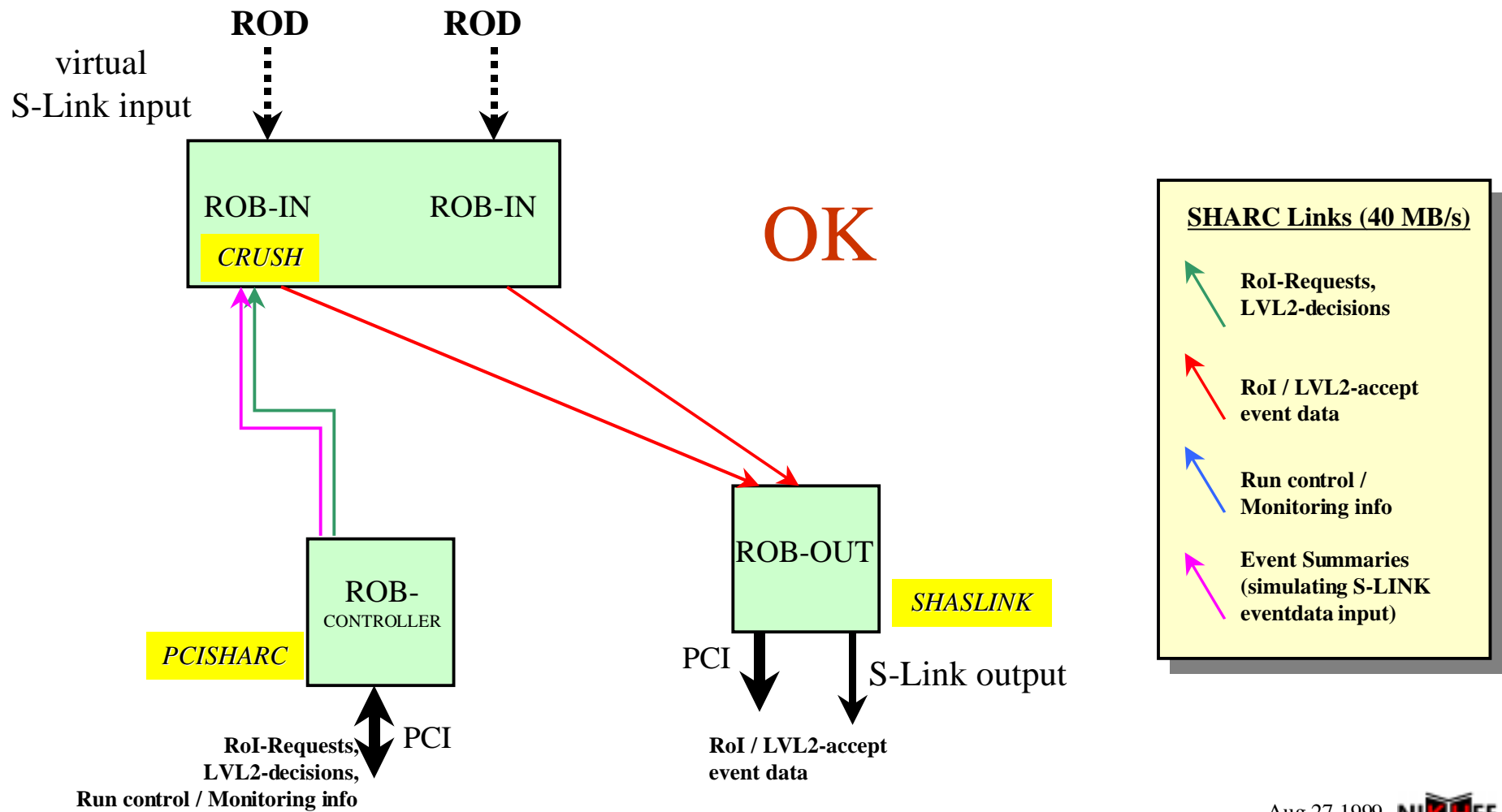
# ROB-Complex test 1: single ROBIN



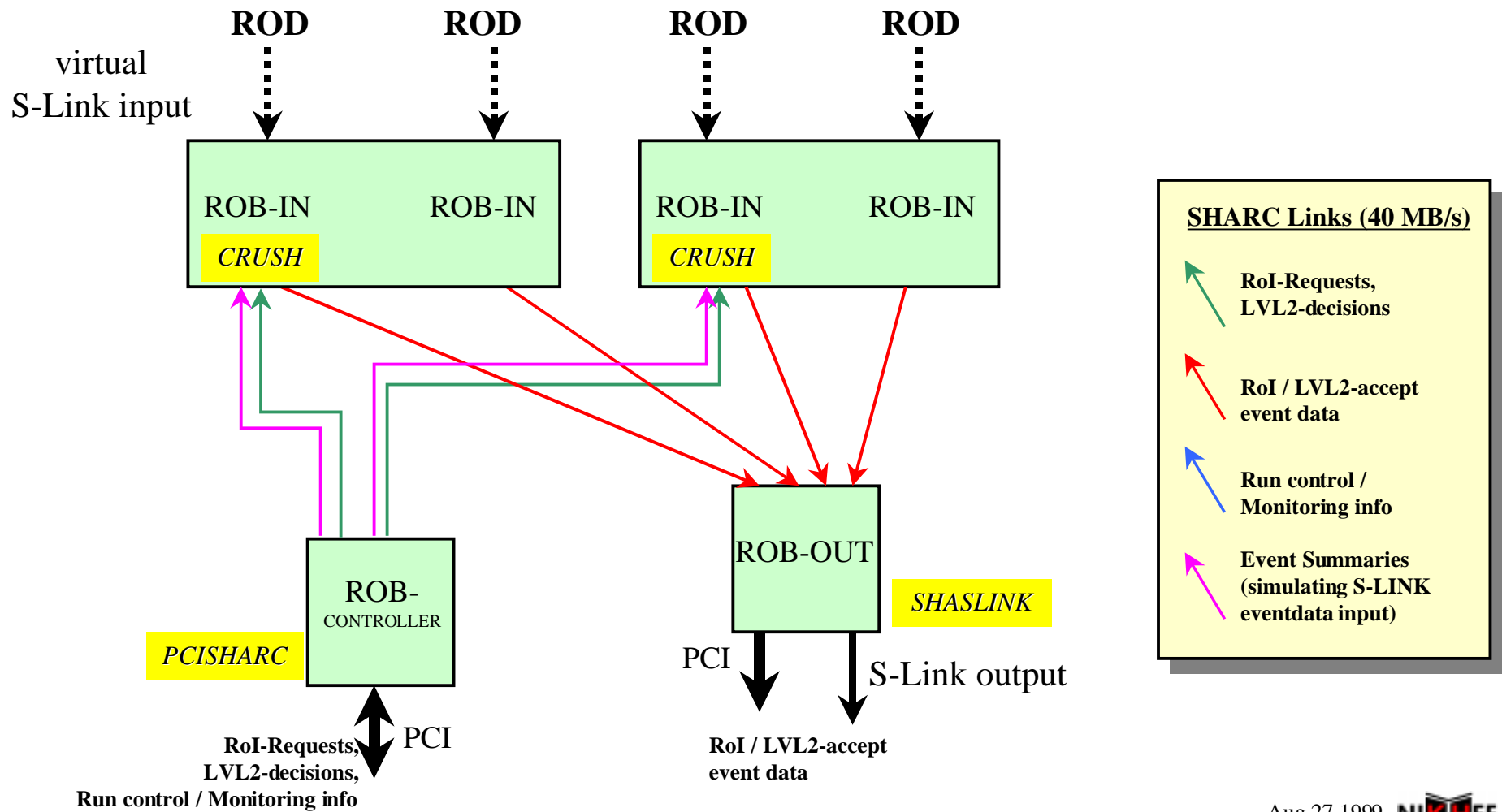
**SHARC Links (40 MB/s)**

- RoI-Requests, LVL2-decisions
- RoI / LVL2-accept event data
- Run control / Monitoring info
- Event Summaries (simulating S-LINK eventdata input)

# ROB-Complex test 2: one CRUSH simulating 2 ROBINS



# ROB-Complex test 3: two CRUSHes simulating 4 ROBINS



In progress : 10-9-99 : OK

## Status MROD-0, 11 September 1999

**SHASLINK operational**, Windows NT support in development,  
MROD software : simplification of ROB-Out software which is under test

**Error-tolerant scheme for fragment building** in M-CRUSH found,  
simulations look fine

**Test of output part FPGA M-CRUSH was succesful**

**Implementation of full design in CRUSH FPGA under way**, may have  
to compromise somewhat on speed (i.e. max. S-link input speed  
< 160 MByte/s, but still higher than the required 100 MByte/s)

**High-level simulation in C++ of TDCs - CSM - MROD system :**  
first version available, checking against available results of VHDL  
simulation of TDCs