

## Major requirements considered in the design

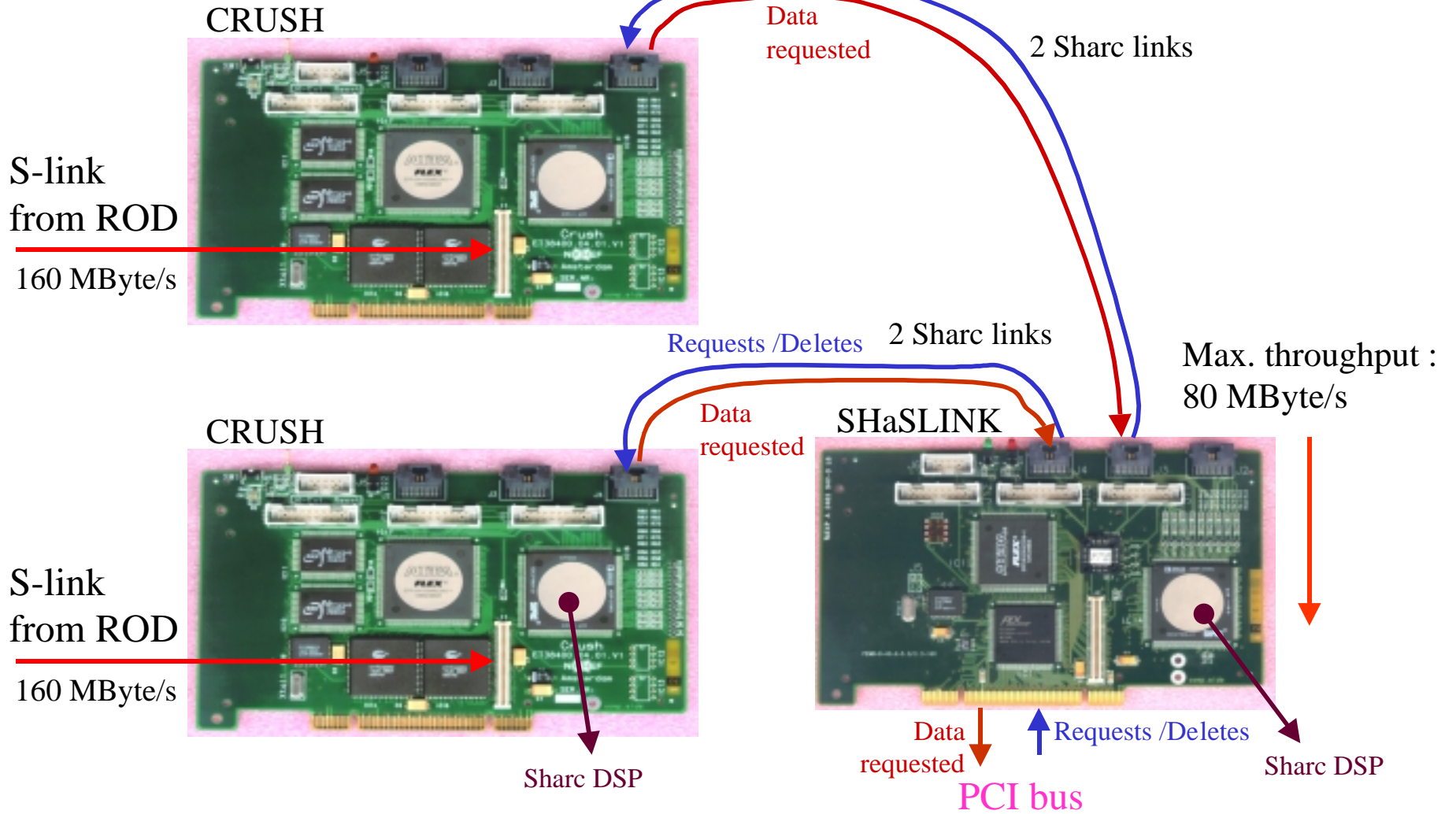
- Performant
- Compact / low power dissipation
- "Lego block" approach -> point-to-point links inside ROB Complex
- Programmability
- Optimization between functionality implemented in FPGA and functionality implemented in software
- Low cost

### Implementation choices made :

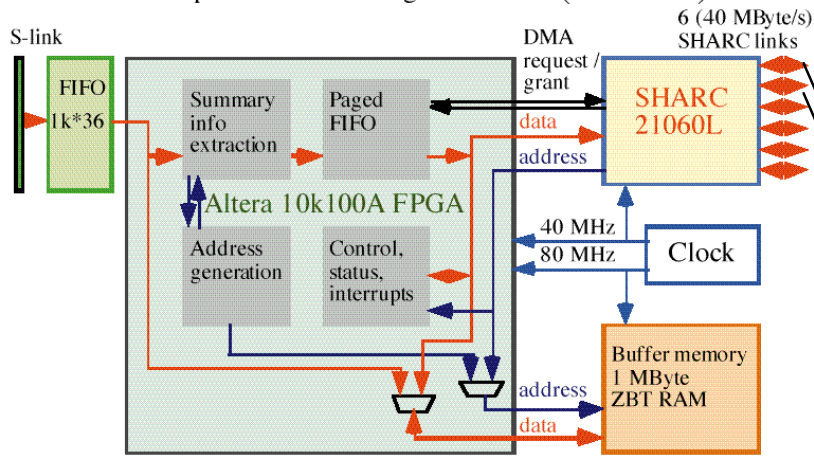
- SHARC DSPs (compact, efficient data mover, point-to-point links with automatic synchronization between sender and receiver, booting via link 4 or from internal memory (ShaSlink : accessible from PCI) )
- ZBT RAM @ 80 MHz
- Altera 10K100 FPGAs
- Software : polling loop, programming language : "C", use DMA controllers for data transport

# High level description

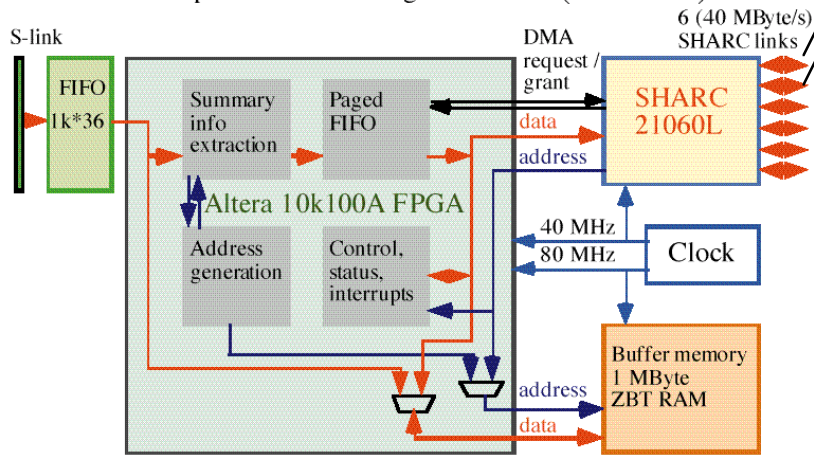
1 link : 40 MByte/s



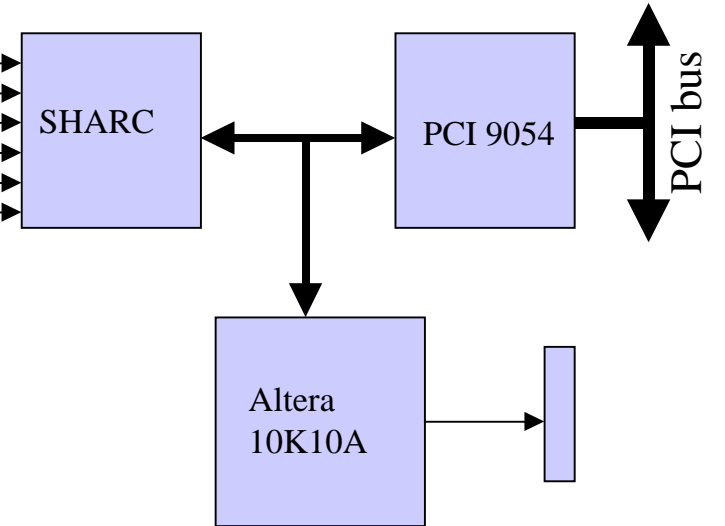
Compact ROBIN Using a SHARC (CRUSH)



Compact ROBIN Using a SHARC (CRUSH)



ShaSLINK



S-link  
@ 160 MB/s

## High level description

ROBIn :

Functionality FPGA :

1. transport S-link data into buffer memory
2. capture summary information for each event fragment (to some extent programmable which information) + start address of that fragment in the buffer memory
3. provide summary info to SHARC via "paged FIFO"
4. gives SHARC read/write access to buffer memory (ZBT RAM)

Functionality SHARC :

1. maintain administration of where event fragments are stored in buffer memory
2. buffer memory fragmentation management
3. servicing of data requests, pipelined action. Event data is transported in two stages, first from buffer memory into internal SHARC memory, then output via link.
4. servicing of delete requests, pipelined action
5. ....

"ROBOut" -> ShaSLink, functionality SHARC :

- Event fragment building using fragments received via links from ROBIns
- Receiving of event data requests and distribution to ROBIns concerned
- Receiving of event deletes and distribution to ROBIns concerned
- Transport of event data into memory PC (by DMA controller SHARC)

All actions are pipelined.

NB1 The automatic synchronization of communication across links is very helpful

NB2 For testing work data sent normally to the SHARC of the CRUSH by the FPGA can also be input via a SHARC link. This has proven to be very useful

## Major design issues encountered

- Getting the FPGA to handle the 80 MHz ZBT RAM

## Minor issues :

- Optimization options of the "C" compiler for the SHARC not completely fail-safe, "volatile" declarations critical
- Link cables need to be short
- Programmable LEDs helped a lot in initial debugging

## Major requirements achieved

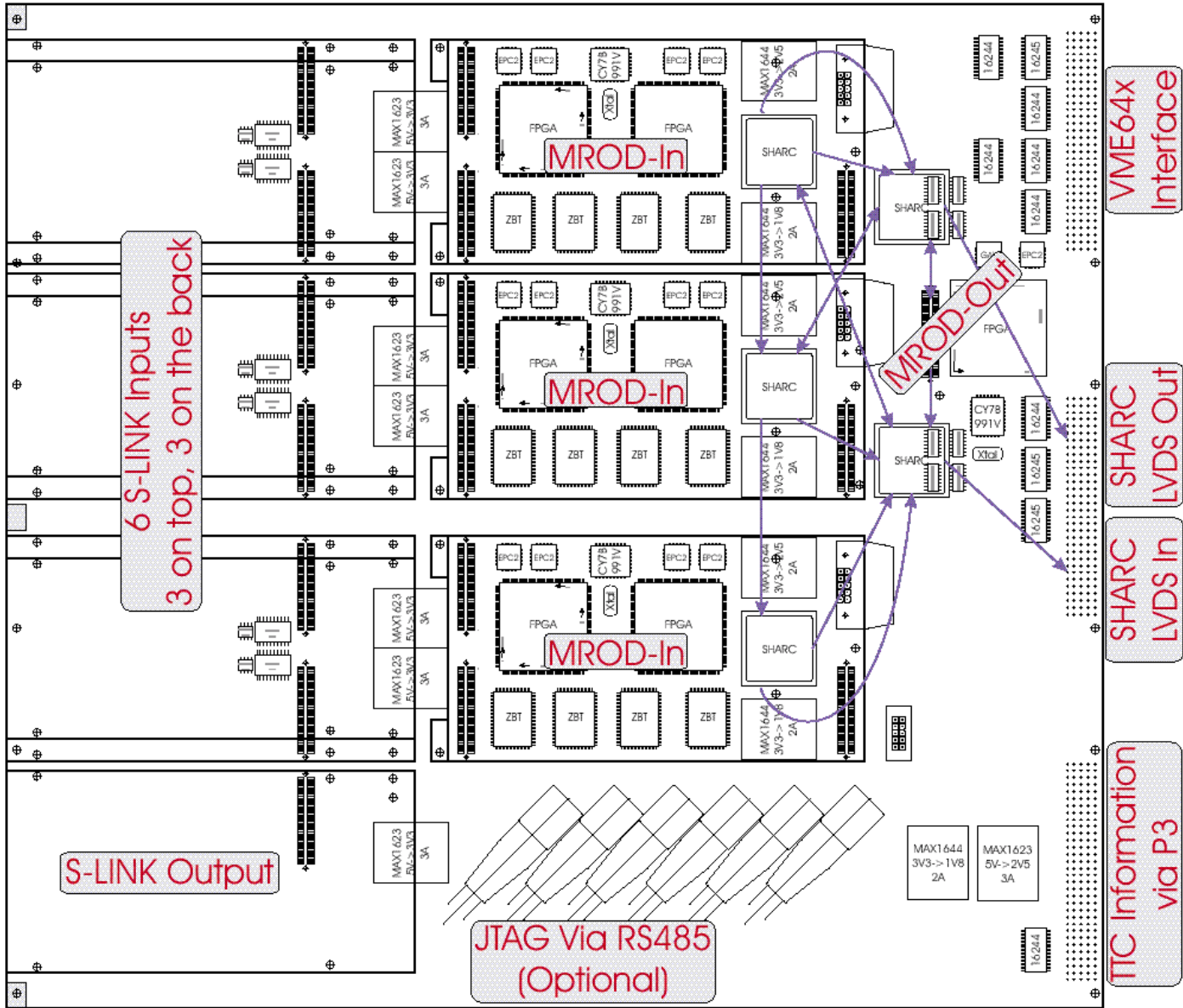
- ROBIN and RPBComplex : demonstration of ability to handle 160 Mbyte/s input data with request rates and delete rates better than "standard" requirements
- Output part : ability of event fragment building and request, delete fan-out at rates better than required
- Detailed understanding of performance, "paper model" (formulas) for computing maximum rate given fragment size and request and delete rates, "computer model" for dynamic behaviour
- Low-power dissipation (2 - 3 W for ROBIN)
- Re-use of hardware for MROD studies

"What I would and would not do if I had my time again"

- Maximum bandwidth SHARC <-> ZBT RAM = 40 MByte/s  
Pipelined transfers as used in the MCRUSH could improve this.  
(but 40 MByte/s is enough for ROBIn function) (ZBT RAM supports 160 Mbyte/s input from S-link and 160 MByte/s output)
- Present FPGA allows only for copying of 2 last words into summary information, 3 words is required, 4 words would be better (note : the information from the missing word can be looked up directly by the SHARC in the buffer memory without searching, so this is only an inconvenience, resulting in a somewhat less efficient operation)

## What we do now

- Development of the MROD using the same philosophy, and which can be turned in a ROB system by reprogramming FPGAs
  - 9 U VME board, VME64x slave interface, TTC connection, S-link out + SHARC link connections (80 Mbyte/s per SHARC link)
  - 6 160 Mbyte/s S-links in (-> front-panel connections -> 2 VME slots)
  - 1 160 Mbyte/s S-link out
  - 1 80 MHz SHARC-II per two input S-links, 1 200k gate Altera FPGA per input
  - 1 Mbyte of ZBT RAM, upgradeable to 4 Mbyte, per input
  - 2 80 MHz SHARC-IIs for building fragments, total building rate > 100 kHz
- Status : design done, ordering parts for 5 boards / 20 input channels (not all VME boards completely populated), PCB lay-out started.
- Planning : hardware available in July, testing completed in autumn.
- As ordering parts (apart from PCBs) almost done, good idea of price (~8kSf for fully populated board with 1 MByte buffer memory per input and without S-links (NB ODIN 160 MByte/s Slink source or destination now Sfr 1360) )



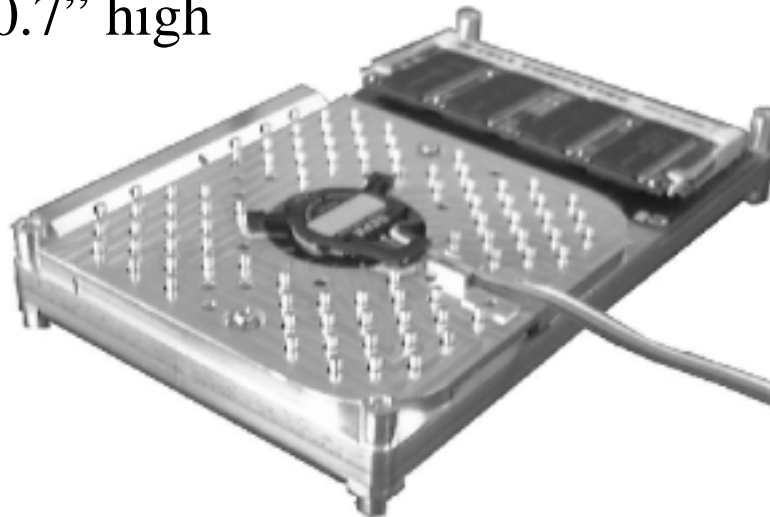
This could be of interest with Linux as OS with GigEthernet i/f



**PLUG-N-RUN™**

**HIGH PERFORMANCE MOBILE PENTIUM® AND PENTIUM II PLUG-N-RUN™ MICRO MOTHERBOARDS**

0.7" high

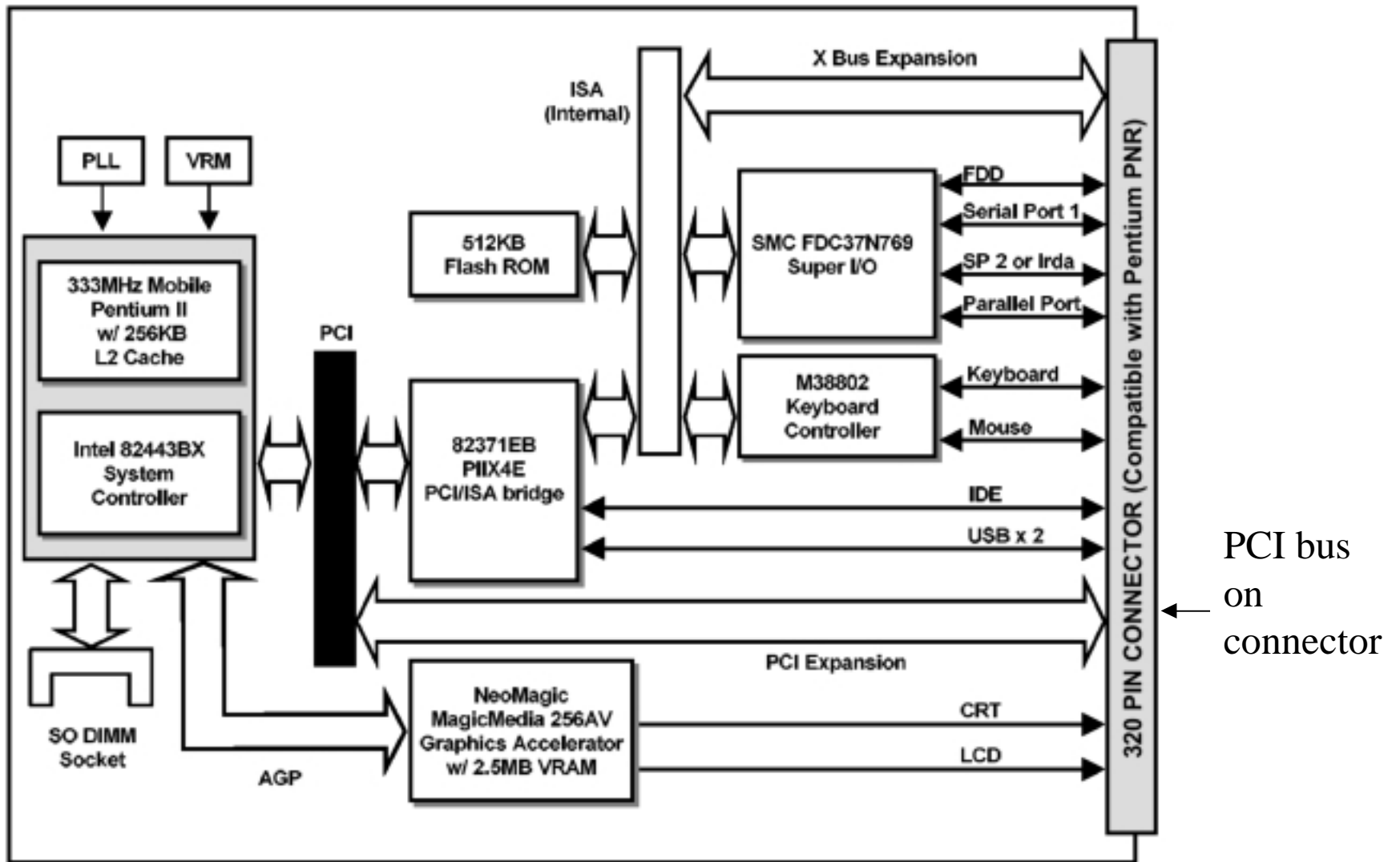


Also 500 MHz

**333MHz Mobile Pentium® II Plug-N-Run™**

**PLUG-N-RUN FEATURES**

- Complete PC Motherboard in a 3" x 5" package
- Intel Mobile Pentium® II 333MHz and Mobile Pentium® 233MHz Processors
- 256KB L2 Cache
- PCI Bus and 8-Bit X-Bus Expansion Bus
- Two USB Ports
- Two Serial, One IrDA, and One Parallel Port
- One FDD and One IDE HDD Interface
- PS/2 Keyboard/Mouse Port
- CRT and LCD Ports
- NeoMagic® Graphics Chips with integrated VRAM
- SO DIMM Socket for up to 256MB SDRAM or EDO
- Runs Windows®95/98 and Windows NT® 4.0
- PhoenixPICO™ BIOS and "Thermal Throttling"
- Rugged Aluminum Frame with Fan Heat Sink, Passive Heat Sink, and Heat Pipe Thermal Solutions



## Summary

- No real problems found in hardware and software developed, expectations fulfilled
- Development of MROD system based on system developed (MROD system consists of 1200 input links : ~ 75 % of what is required for ROB system)
- Hardware developed now for MRODs provides a good basis for further development of ROB hardware (identical MROD and ROB hardware, with differently configured FPGAs would be possible). Resources available for further development. Lego block approach allows for easy changing number of ROBs per NIC (or per 2 NICs)