



Application Specific Computing (ASC) at Institute for Computer Engineering (ZITI) Heidelberg

Andreas Kugel

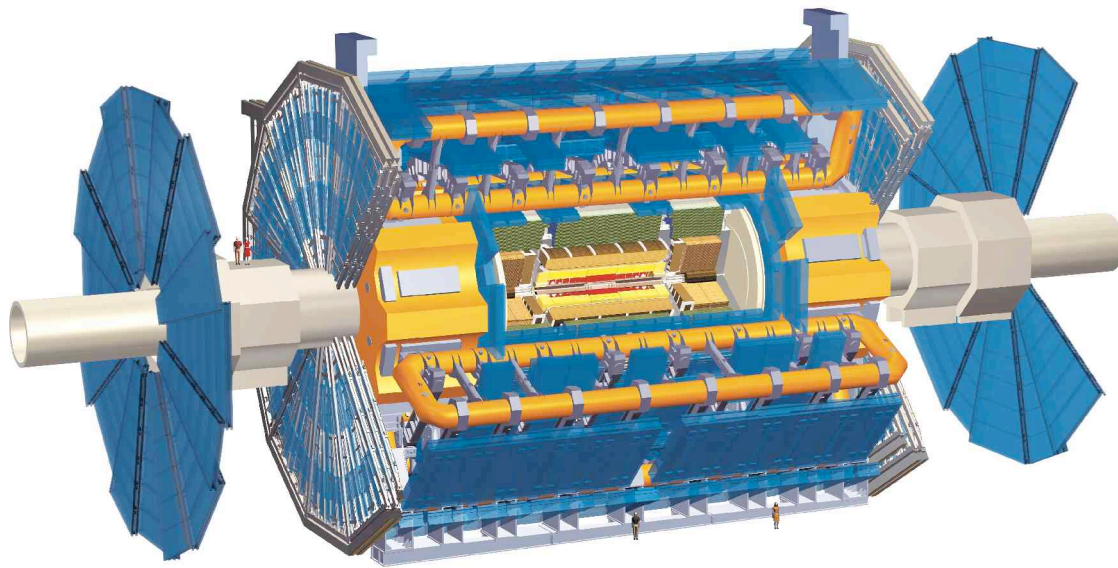
Contents

- Intro ASC @ ZITI
- FPGA Co-Processors
- GRACE Project / SPH
 - FPGA
 - GPU
- Arguments

ASC Departement @ ZITI

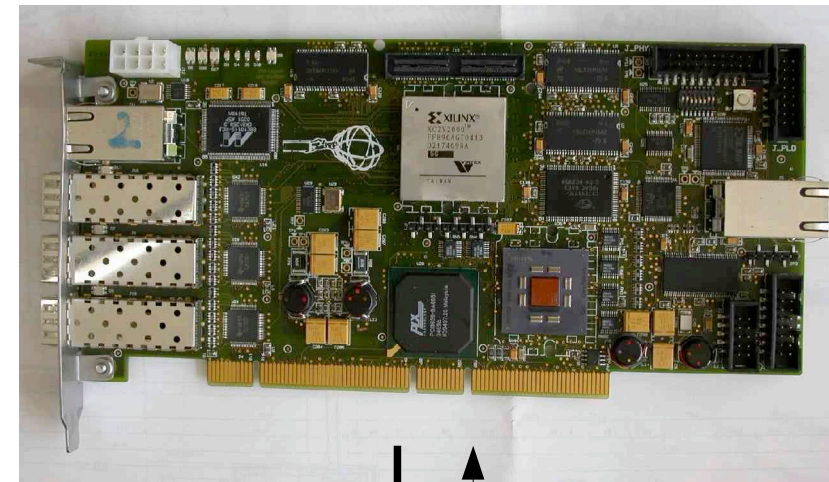
- Head: Prof. Dr. Reinhard Männer
 - Physicist => Computer Engineering
- Main Areas
 - Trigger and Data Acquisition in (high energy) Physics
 - **ATLAS** at LHC, CBM, XFEL
 - Accelerated Scientific Computing (G. Marcus)
 - **Simulations**, Biocomputing (Haralick Feature Extraction)
 - Virtual Reality in Medicine
 - Modelling, real time tracking and visualisation
 - Surgery training equipment
- General purpose (CPU, GPU) + custom (ASIC, FPGA) processors

ATLAS DAQ



100kHz
Event rate

ROBIN FPGA Card



Data

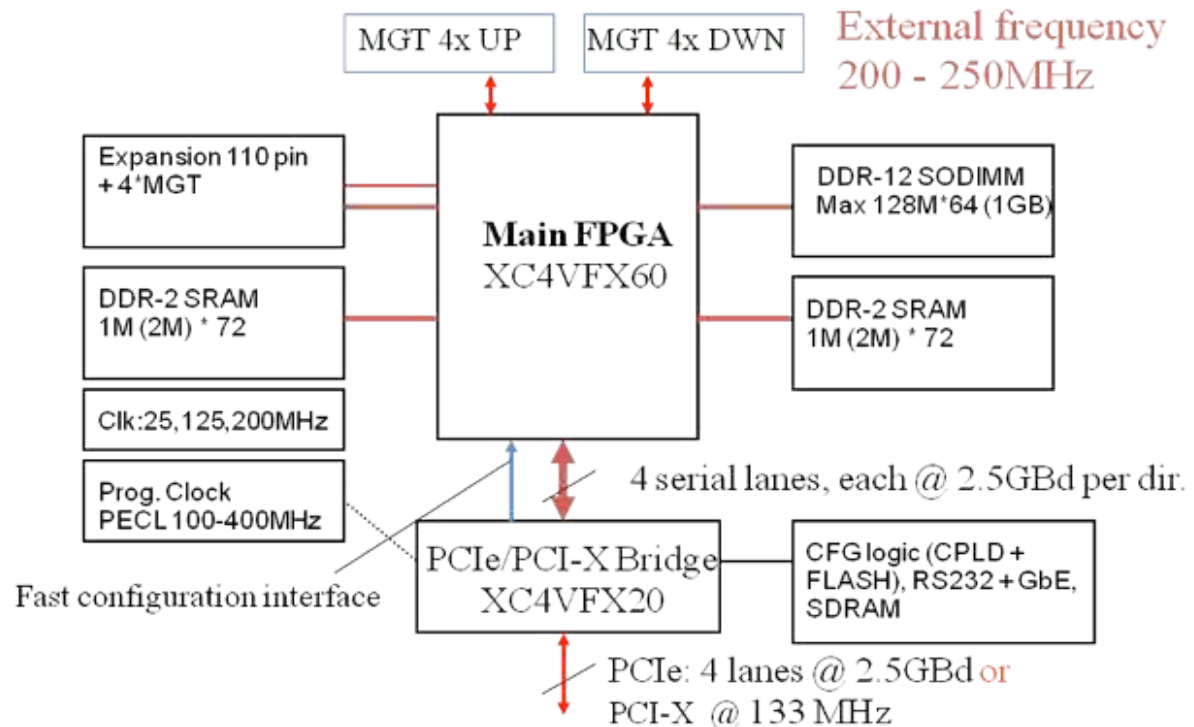
3 .. 20kHz
Requests

- 1600 x 2Gb/s from detector
- 600 ROBINS in 160 PCs
 - Intelligent buffering
- Online processing on CPU farms O(1k) nodes

FPGA Co-Processors

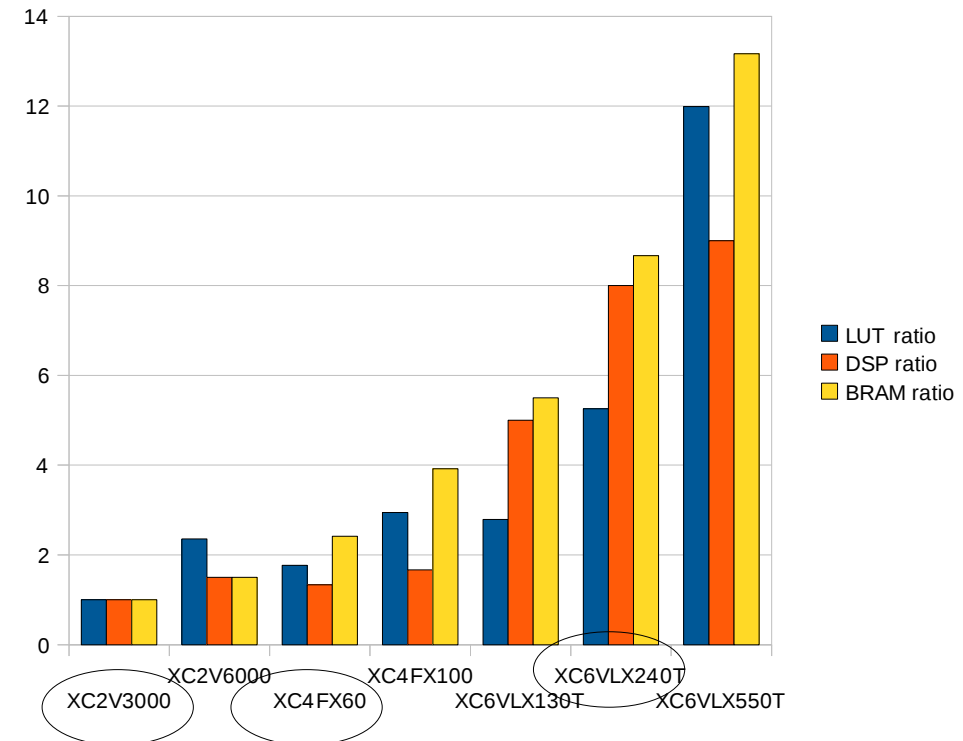
- Co-Pro Blocks
 - Main FPGA
 - Host IF
 - Local DRAM
 - Local SRAM
 - I/O expansion
 - Clk, Ctl, Cfg
- ASC activities
 - Boards: VME, cPCI, PCI, PCIe (HTX at CA dept.)
 - Algorithms: stand-alone + hybrid
 - Tools: compilers, libraries, frameworks

MPRACE-2 Block diagram



Co-processors cont'

- MPRACE-1 (equiv. ROBIN)
 - 3M gates Virtex2, 2001
 - PCI, 256MB/s
 - still in use ...
- MPRACE-2
 - 6M gates Virtex-4, 2007
 - PCIe-4X, 1GB/s
- MPRACE-3
 - 24M gates Virtex-6
 - 2010 (planned)
 - PCIe2.0-8x, 4GB/s

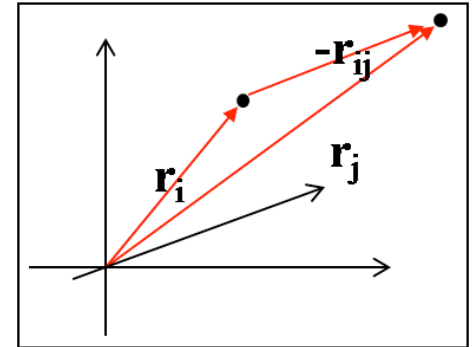


Available resources wrt MPRACE-1
LUTs: 30k, DSP/BRAMs: 100

SPH: 1 Flop = 200 LUTs, 0.5 DSP
Speed 60 .. 250MHz

Simulations – The GRACE Project

- Collaboration with R. Spurzem/ARI HD since 1998
- Goal: Hybrid system for N-Body + SPH
 - CPU: Tree building, Integration, $O(N)$



- GRAPE: Gravitational force, $O(N^2)$

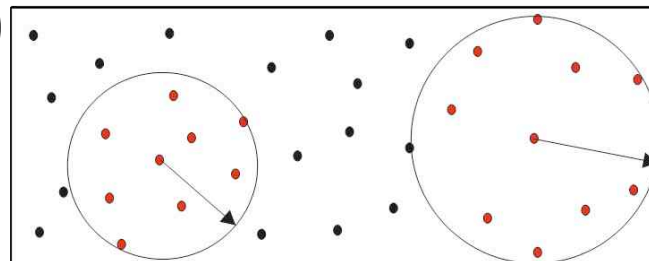
- Now on GPU

$$\vec{F}_i^{grav} = -Gm_i \sum_{i \neq j} \frac{m_j (\vec{r}_i - \vec{r}_j)}{\left(\epsilon^2 + |\vec{r}_i - \vec{r}_j|^2\right)^{\frac{3}{2}}}$$

- Once gravitation is done, SPH is most time consuming

- FPGA: SPH, $O(N_n * N)$

- Now on GPU
- But also FPGA



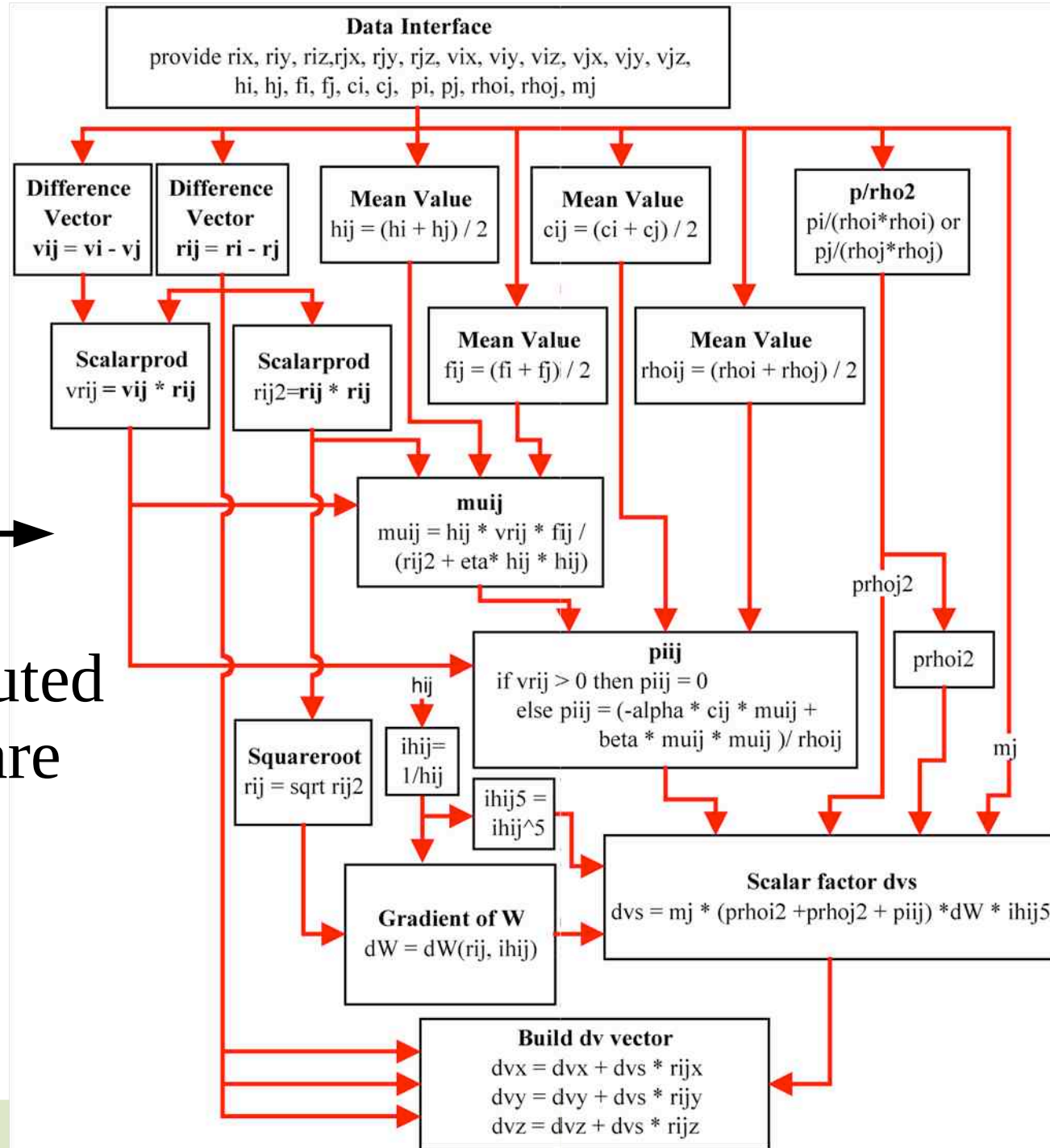
$$\vec{F}_i^{hydr} = \sum_{\text{Neighbors } j} \vec{F}_{ij}^{hydr}$$

SPH on FPGA

- 60 Float operations
- Limited precision (16 bit significant)

$$\frac{d\vec{v}_i}{dt} = -\frac{1}{\rho_i} \nabla P_i + \vec{a}_i^{visc}$$

- Algorithm not executed but cast into hardware
- Systolic process
- Very efficient IF problem matches



FPGA Programming

- HDL based programming: time consuming expert task
- MATLAB style: DSP-like apps, fixed point
- Custom: building-blocks + pipeline generator (PDL)

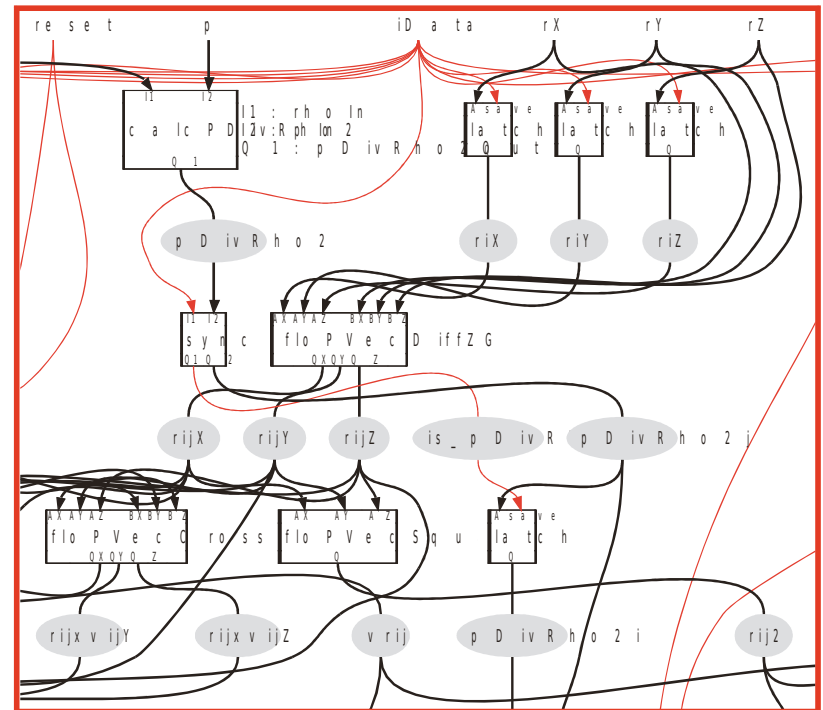
```

entity distance;
clock clk;
# parameters
floPValDef fpDef(signifLength=>24,
  expLength=>8,useSign=>1,uselsZero=>0);
# inputs
signal (suppress_v);
floPVal (v,a,t)(fpDef);

# calculate
t2 = <floPSquare> t;
s1 = v <floPMult> t;
ss1 = gated(s1,suppress_v);
half_a = <floPDiv2> a;
s2 = half_a <floPMult> t2;
s = ss1 <floPAdd> s2;

```

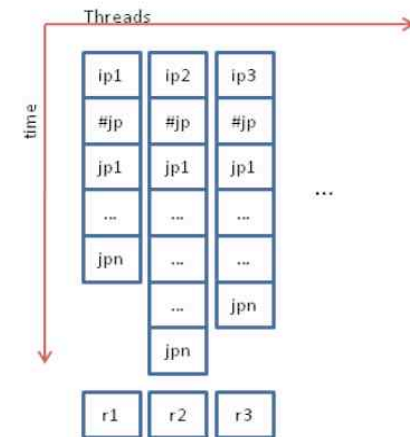
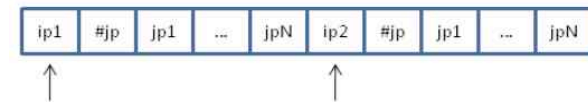
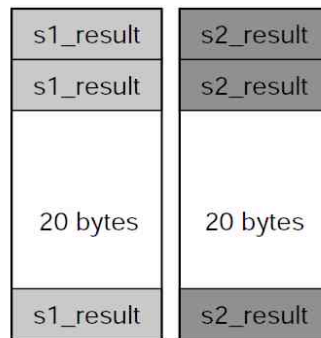
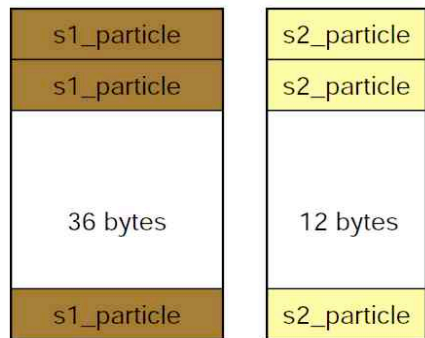
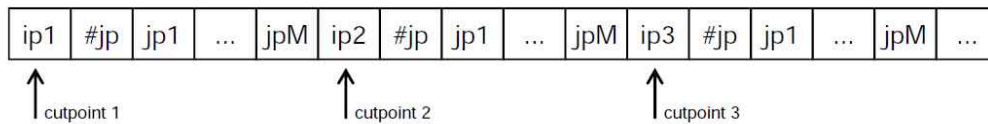
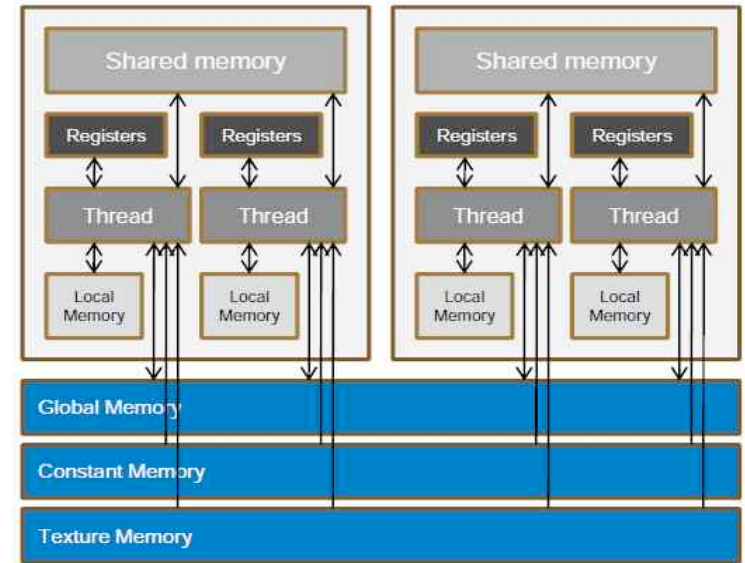
High level description



HDL + visual representation

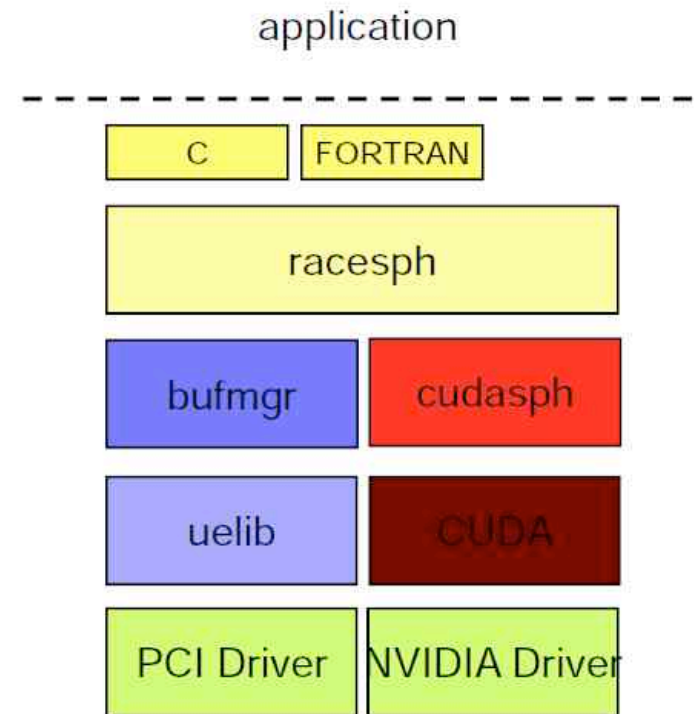
SPH on GPU

- $N(\text{pipelines}) \sim 500 \dots 1k = x(\text{processors}) * y(\text{threads})$
- Local, shared and global memory
- Map neighbour lists on threads: must fit in shared memory

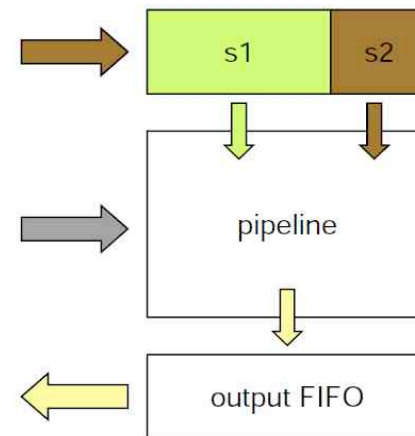


Common GPU/FPGA Framework

- C, C++, Fortran interfaces
- Complete abstraction of SPH capabilities: racesph library
- Intelligent buffer manager, incl. re-formatting for FPGA
- Device specific libraries and drivers



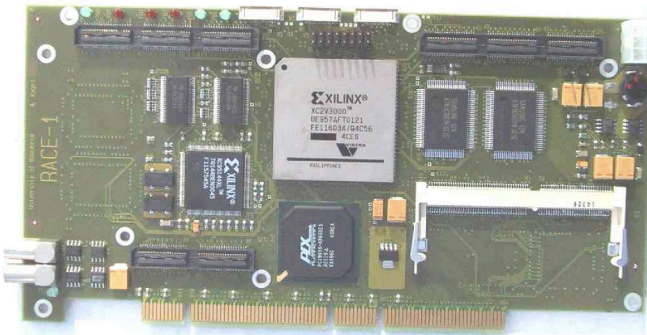
- Load particles (s1)
- Send NLs
- Read s1 results
- Load particles (s2)
- Send NLs
- Read s2 results



To-date Systems

- Titan cluster at ARI (VW project): 32 nodes/64 cores, up to 32 FPGA coprocessors (Virtex-2/-4)
- Kolob cluster at ZITI (Frontier project): 42 nodes/324 cores, 40 GPUs (Tesla C870)
- Results

	MPRACE-1	Tesla C870	MPRACE-2	MPRACE-3
Speedup	10	20	40 (estimated)	80 (est. more)



MPRACE-1: Virtex-2



GPU

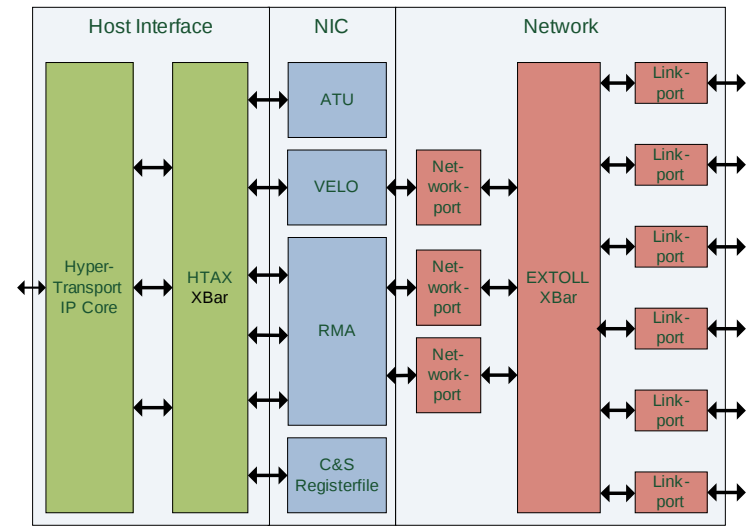
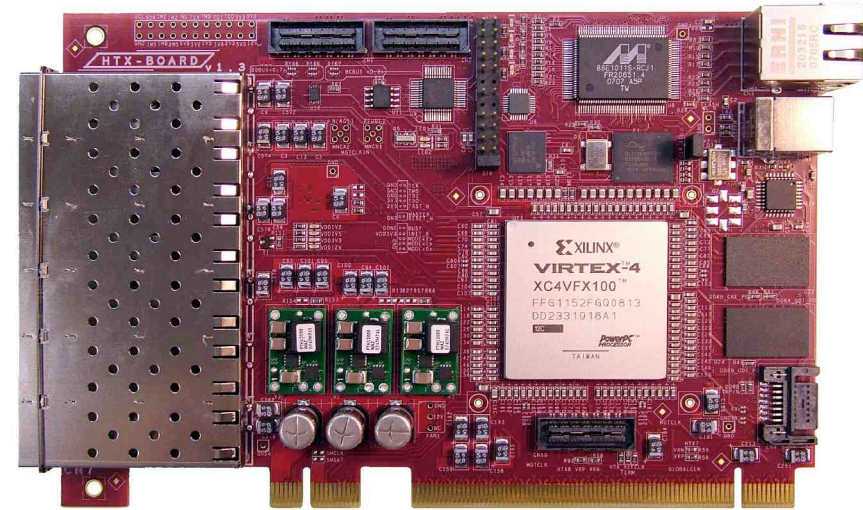


MPRACE-2: Virtex-4

Future: GPU/FPGA accelerated clusters at CAS/NAOC

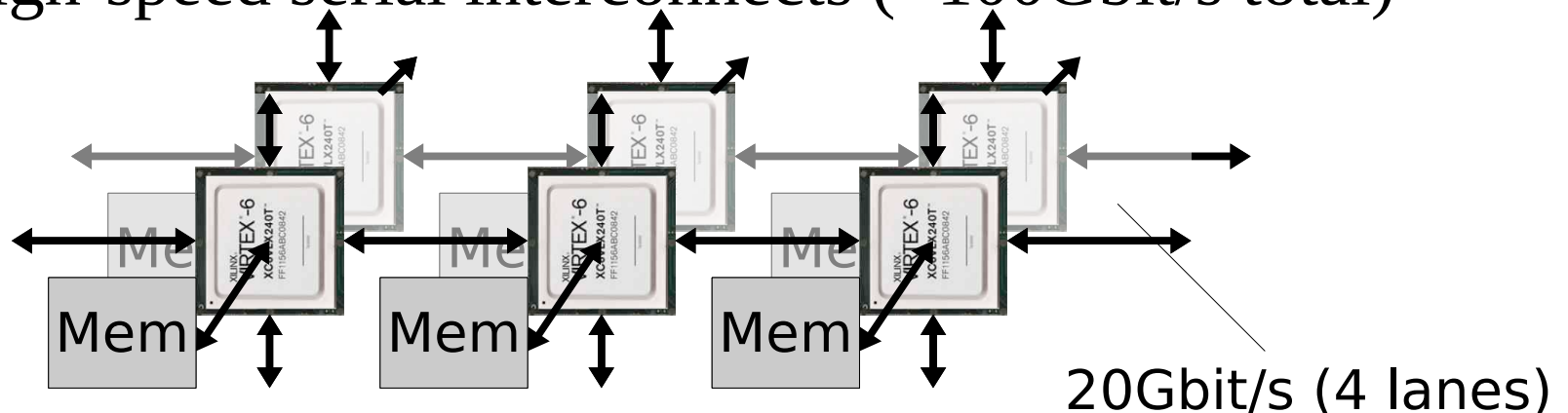
Communication

- Computer-Architecture group at ZITI (Prof. Dr. Brüning)
- EXTOLL Prototype:
 - VELO: low-latency ($<1\mu\text{s}$) communication engine
 - Link port
 - X-bar
 - Network ports
 - HTX X-bar
 - HTX core
- 15% of XC4FX60 with RMA



Why FPGAs?

- In co-processor style apps GPU competition very strong but:
 - FPGA performance “low” but sustained = peak
 - GPU sometimes only few % of peak
- Energy: FPGA – 20W, GPU – 150W
- Embedded applications
 - Density (no space to put the PCs)
 - I/O: high-speed serial interconnects (>100Gbit/s total)



Why not? (... but ...)

- Low performance compared to GPU
 - Depends on application (see previous slide)
- Complex programming compared to GPU
 - Toolboxes (improvable) for pipeline designs at hand
 - Various “C”-based approaches in academia/industry
 - OpenCL interfaces coming (well, maybe not so soon)
- Cost
 - Factor 5 more expensive than GPU. No excuse (well, there are just no games running on XILINX chips)
- Custom
 - True, but non-issue for some of our applications