Synective Labs

Experts in Application Acceleration

Software Acceleration

FOR DUMMIES®

Magnus Peterson
Synective Labs

Synective Labs quick facts

Synective Labs

Accelerator technology is the art of using other types of devices than ordinary CPUs to (drastically) increase a computer's performance

Synective Labs

Things are no longer what they used to be …

Synective Labs

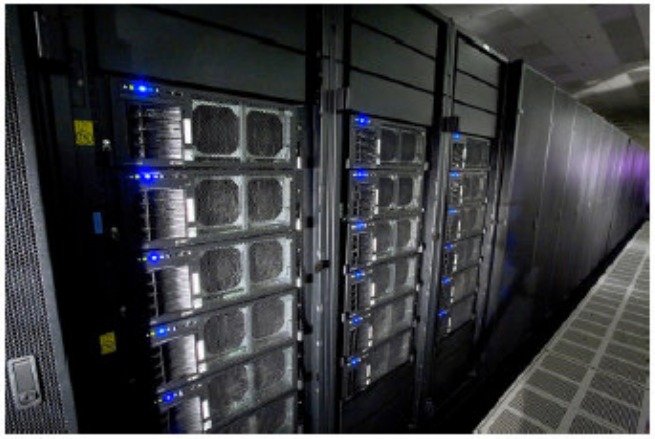
The #2 computer on the Top500 list

at Los Alamos National Laboratory

Roadrunner @ LANL: 1.1 PF/s

- 12,960 Cell chips (8+1 cores) (on IBM Model QS22 blade servers)
- 6,948 dual-core AMD Opteron (LS21 blades)
- 104 TB main memory
- Power is approximately 2.5 MWs at load
- 294 racks grouped in 18 units
- 5,500 square feet = 500 m2

= 129 600 cores!



Synective Labs

home | reviews | **news** | downloads | cnet tv | On MovieTome: Leaked images from TRANSFORMERS 2?

cnet news

Latest News | Crave | Webware | Business Tech | Green Tech | Wireles

Home › News › Underexposed

UNDEREXPOSED

a blog by Stephen Shankland

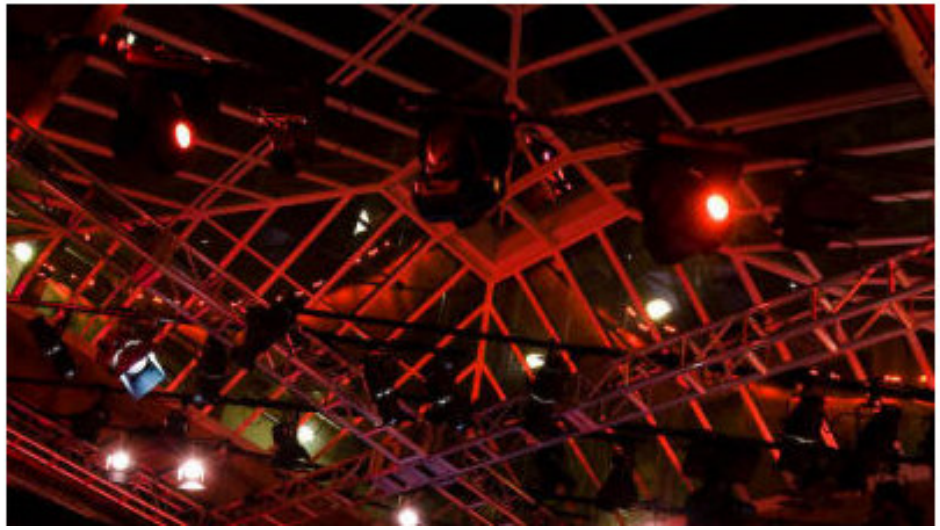
September 22, 2008 6:32 PM PDT

Adobe uses graphics chip for faster Photoshop CS4

Posted by Stephen Shankland

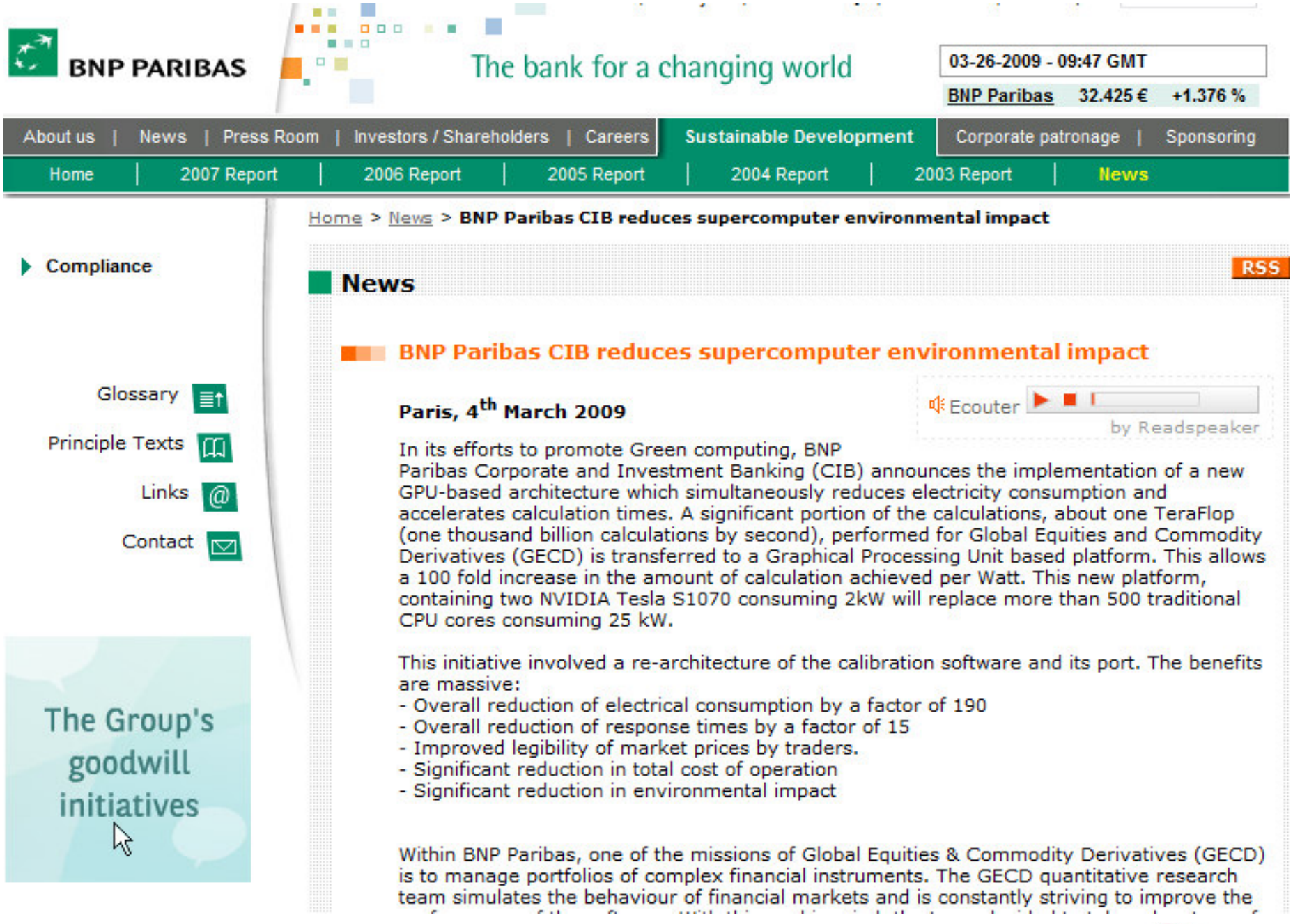
Print | E-mail | Share | 30 comments

Photoshop is a famously taxing piece of software, but beginning with the upcoming CS4 version, it'll be able to employ the muscle of your computer's graphics chip for the first time.



The new version of Adobe's flagship software product takes its first steps in using the graphics processing unit, or GPU, said John Nack, principal product manager for Adobe Photoshop. For example, the graphics chip helps Photoshop CS4 fluidly zoom in and out, rotate the canvas so

Synective Labs



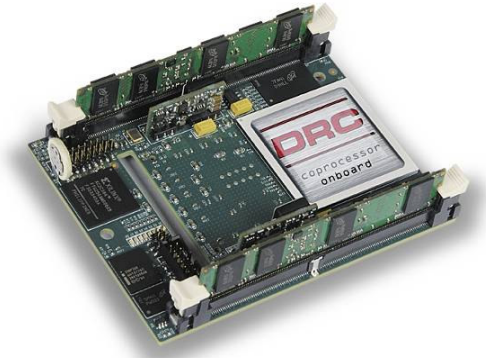
Synective Labs

... heterogeneous computing is here for real!

Synective Labs

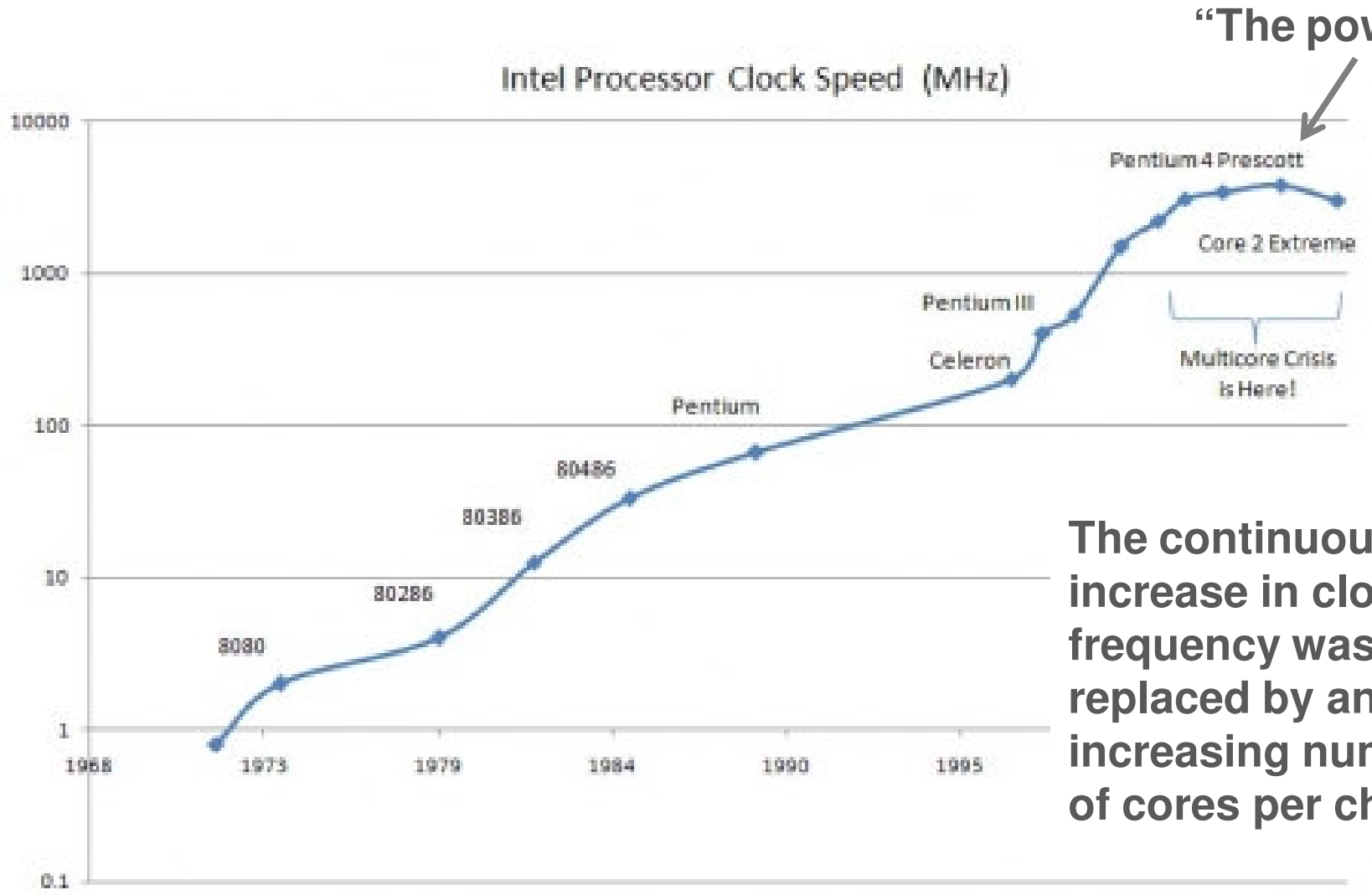
What drives the accelerator technology?

- Increased performance:
  - To turn hours or days into minutes
- Reduced power – Green computing
  - Increase the Gflop/Watt ratio
- Reduced machine size/investment
  - Do the same with less equipment – “desktop supercomputer”



Synective Labs

CPUs' development



The continuous increase in clock frequency was replaced by an increasing number of cores per chip

Synective Labs

… the good old days were gone!

- You could no longer sit back and relax – and know that your sequential code got 2x faster every 18 months
- To now utilize the CPUs' increasing performance, you *have to parallelize* your code
- This has opened up the field for accelerators – if you anyway have to rewrite your code, why not look at the alternatives?

The “big three”

- **FPGAs** – ”programmable” hardware – roughly 200 GFlops peak performance (soon 500 Gflops)
- **GPGPUs** – 800 (soon 1600) sp floating point processors – roughly 1200 GFlops (soon 2700 Gflops) peak performance
- **CELL** – 8 specialized processors – roughly 250 Gflops peak performance

- (All Gflop numbers are for single precision)

Accelerator development is driven by mass markets

- Advanced graphics for computer games drives the **GPU** development
- Signal processing in for example mobile base stations and embedded systems drives the **FPGA** technology
- Gaming is driving the **CELL** processor development

- But application acceleration is a nice spin-off!

It’s all about parallelism!

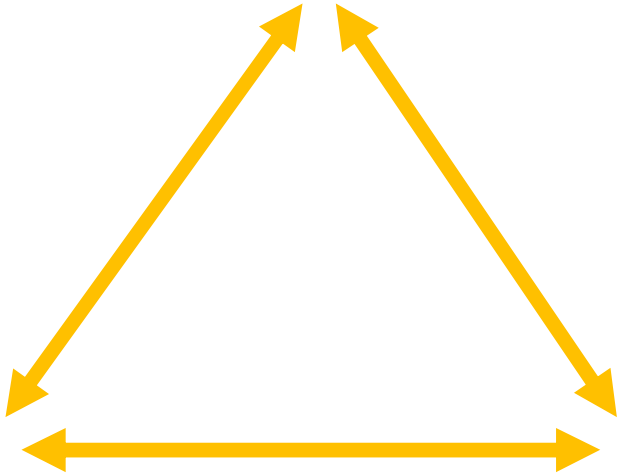
- **CPUs** – more and more cores - multicore
- **CELL** – 8 parallel floating point cores
- **GPUs** – several hundreds of parallel floating point units
- **FPGAs** – free architecture, but typically you implement a “core” that is the inner loop of your algorithm and then add as many as you can fit

- Computers no longer get faster, just wider!

Synective Labs

What to use when?

CPU – ”sequential code”



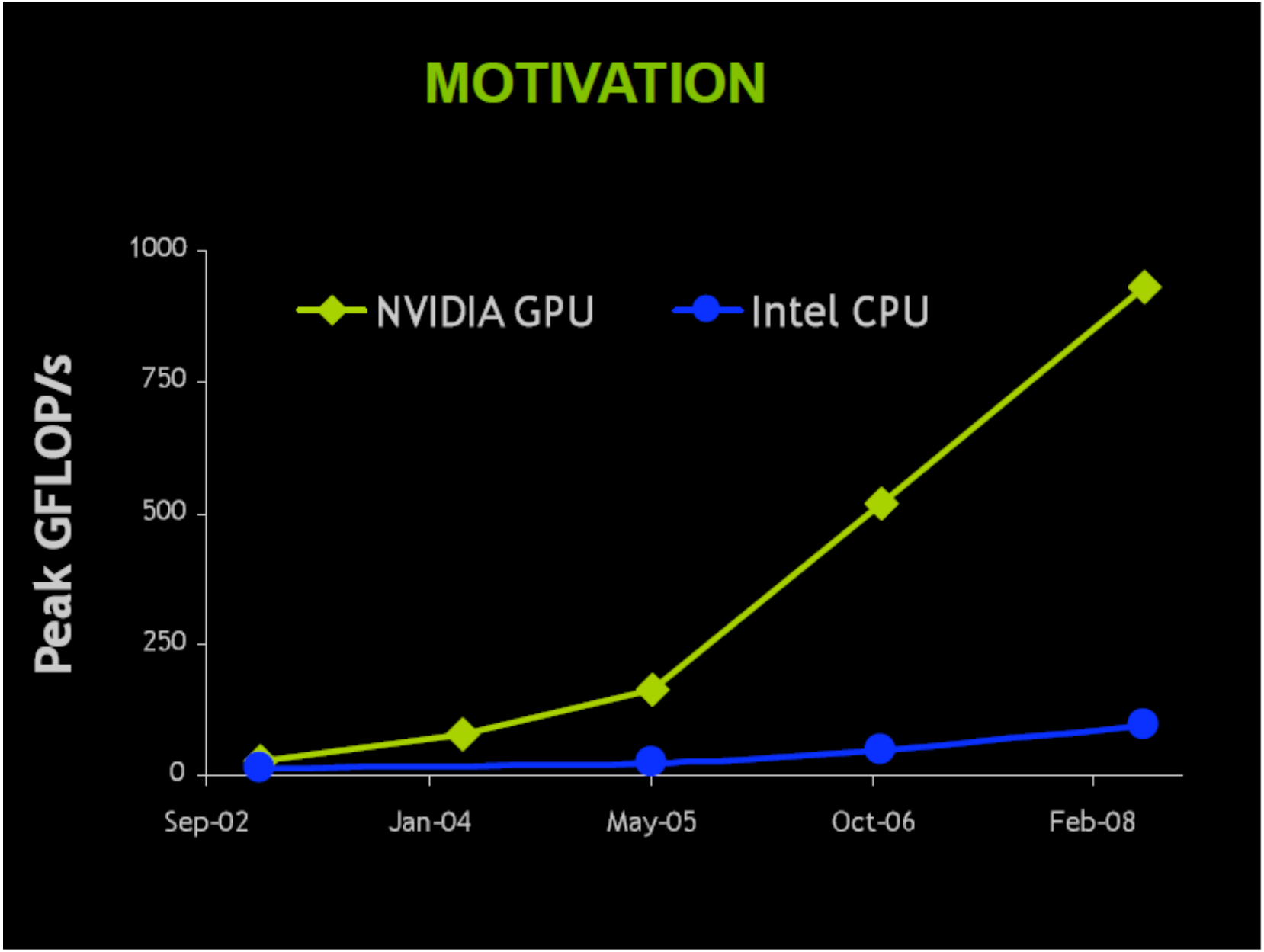
FPGAs – integer, parallel algorithms, streaming type of code – FREE architecture!

GPU, Cell – massively parallel floating point operations

Synective Labs

GPGPUs

Synective Labs



Synective Labs

GPU characteristics

- Contain large number of floating point processing elements
- Very high processing rate for data that resides in the memory on the GPGPU board – typically 100 GB/s memory bandwidth
- Bottleneck when transferring data to and from the GPU-board
- Real programs usually reach only a fraction of the peak performance
- Two large vendors; NVIDIA and ATI/AMD

AMD FireStream 9270

- 1200 GFlop/s peak Single Precision FP
- 240 Gflop/s peak double precision FP
- 800 cores
- 2 GB memory
- 6 GFlop/Watt (SP)

NVIDIA Tesla S1060

- 933 GFlop/s peak Single precision FP
- 78 GFlop/s peak Double precision FP
- 240 cores
- 4 GB memory
- 6 GFlop/Watt (SP)

Synective Labs

AMD FireStream 9270

- 1200 GFlop/s peak Single Precision FP
- 240 Gflop/s peak double precision FP
- 800 cores
- 2 GB memory
- 5 GFlop/Watt (P)

AMD Radeon HD5870

- 2720 GFlop/s peak Single precision FP
- 544 GFlop/s peak Double precision FP
- 1600 cores
- 1 GB memory
- 14 GFlop/Watt (SP)

Synective Labs

NVIDIA Tesla S1060

- 933 GFlop/s peak Single precision FP
- 78 GFlop/s peak Double precision FP
- 240 cores
- 4 GB memory
- 6 GFlop/Watt (SP)

NVIDIA Fermi, Tesla C2070

- ?? GFlop/s peak Single precision FP
- 630 GFlop/s peak Double precision FP
- 512 cores
- 6 GB memory
- 6 GFlop/Watt (SP)
- ECC

Synective Labs

What makes GPUs fast?

- Hundreds (thousands) of floating point cores, organized in a SIMD architecture
- Large number of threads (thousands) with “zero cost” thread switching hides the memory latency
- Need algorithms that can be heavily parallelized to keep all cores busy

GPGPU Tools

- **OpenGL** – very graphics oriented
- **CUDA** – NVIDIA
  - Currently the most mature development platform
- **Brook+, CAL** – AMD/ATI
- **OpenCL** – platform independant initiative
  - Now available for NVIDA and AMD GPUs plus AMD multicore CPUs
- Libraries: BLAS, FFT ...

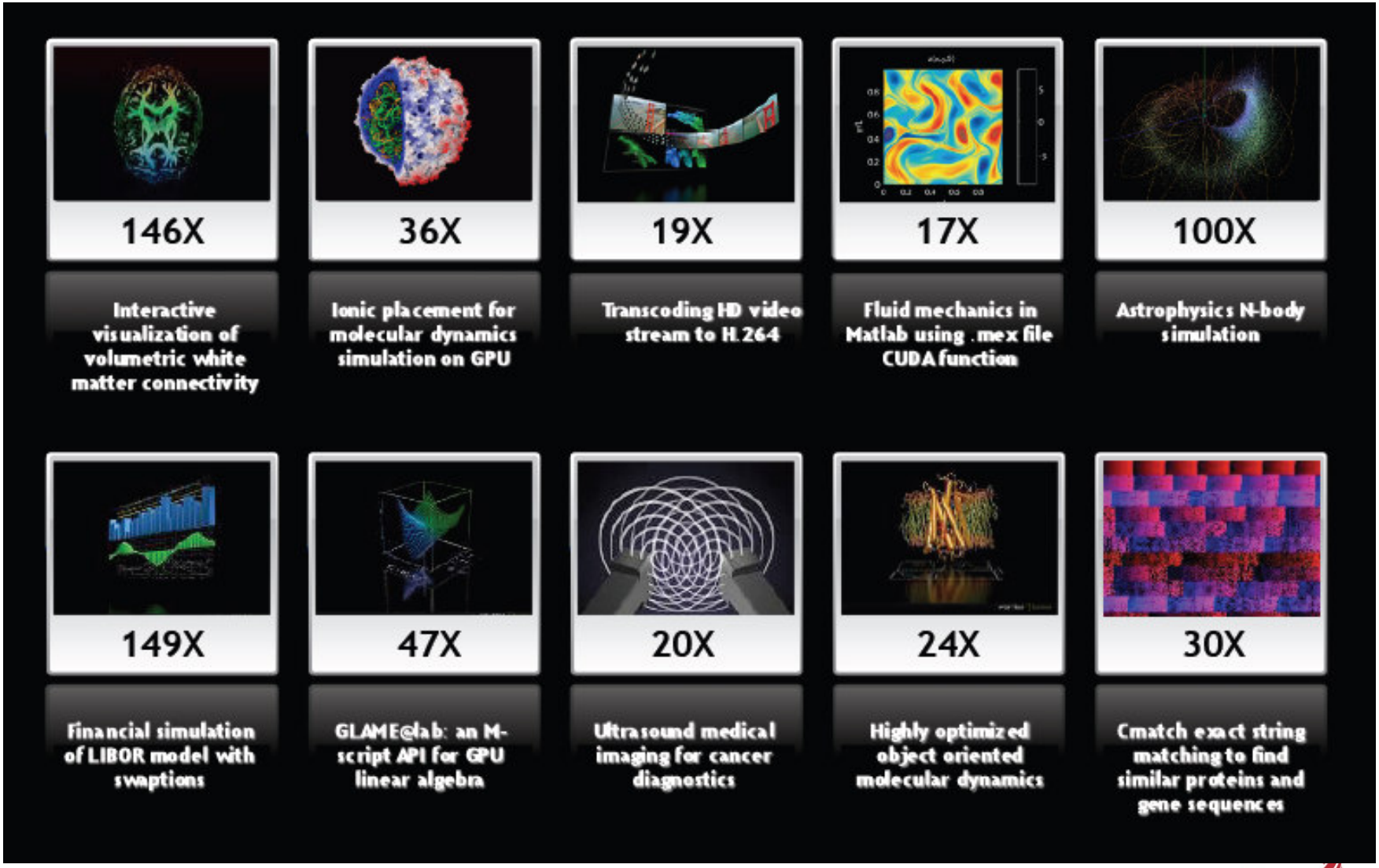
Synective Labs

Other approaches

- PGI Accelerator Fortran and C-compilers – support GPU acceleration
- Use openMP-like directives
- Supports currently NVIDIA GPUs through CUDA

The Portland Group

Some GPU Examples



Synective Labs

Customer example

- Customer makes image enhancement solution for X-ray images
- Original PC-based solution can handle 3 images/second
- The image enhancement algorithm has been fully implemented on the GPU
- Final solution: PC with GPU handles 100 images/second

Synective Labs

The IBM Cell processor

Cell characteristics PowerXCell 8i

- Has one Power Processor element (PPE) plus 8 floating point cores (Synergistics Processor Elements, SPE)
- 100+ Gflop/s peak double precision
- 200+ Gflop/s peak single precision
- Supports 16 GB local memory at 25 Gbytes/s

Cell blade system

- Available for blade systems: QS22
- 2 PowerXCell 8i processors per blade
- Up to 32 Gbyte per blade
- One chassi can carry 14 blades
  - 6.4 TFlop/s SP per chassi
  - 3.0 TFlop/s DP per chassi
- Programmed using a multi-threaded model with development tools from IBM

Cell programming

- The PPE is programmed using C, C++ or Fortran
- IBM Cell SDK to handle the SPEs
- The SPEs are also programmed using C, C++ or Fortran, but with some restrictions.
- The SDK includes good support for profiling, debugging and analysis

More on the Cell …

- Also available as PCIe extension boards
- Note: The Cell-processor in the Playstation 3 is a simpler, cheaper, slower version.
- In Roadrunner, # 2 on the Top500-list, the Cell-processors contribute to 96% of the peak performance while there is s 2:1 relation between Cells and Opterons
- IBM has announced that the development of the next generation Cell has been halted, but the technology will be offered in other forms in the future

FPGAs

FPGA's characteristics

- "programmable hardware"
- Filled with 100.000:nds of small generic building blocks that can be interconnected to form a hardware specific to the problem at hand.
- Contains also memory and multiplier blocks
- Can easily be re-programmed for another task in less than a second.
- Low power – typically 10-20W per device

Synective Labs

What makes the FPGA fast?

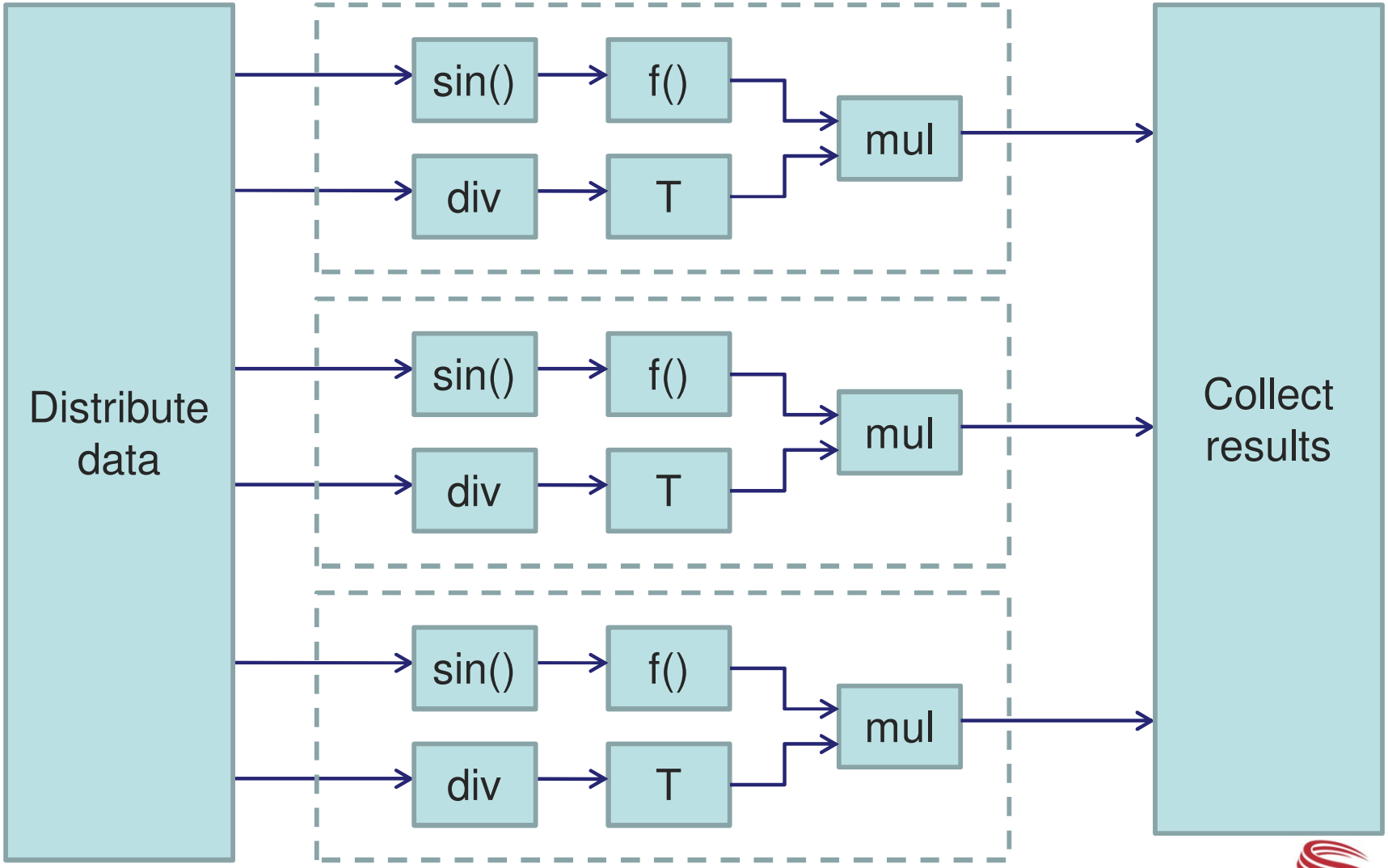
With an FPGA you build a computational architecture that is *tailored* for your algorithm instead of adapting your algorithm to a fixed architecture

Synective Labs

How to make use of an FPGA

- Make a computational pipeline of the inner loop you want to accelerate
  - The complete inner loop is run once every clock-cycle
  - Feed in new data each clock cycle -> get a new result each clock cycle – the latency doesn't matter if the amount of data is big, and it usually is …
- Parallelize – fit as many instance of that pipeline as possible into the FPGA

Pipeline and parallelize



Synective Labs

Some numbers ...

- Very high internal memory bandwidth
  - The XilinxSX240T has 516 dual-ported 36-bit wide Block RAMs, which means the design can pull and/or push (32/8 * 2 * 516) 4128 bytes every clock. At a conservative clock frequency of 250 MHz, that is 1,032 GB/s.
- High raw performance
  - The SX240T with 149,760 LUTs with 2 outputs per LUT (149,760 LUTs * 2 bit operators per LUT * 250 MHz * 1/64) is 1.17 trillion 64-bit op/s. The quad-core Opteron (4 cores * 4 ops per clk * 2500 MHz) is 40 billion 64-bit op/s. That is (1170/40) 29x more raw computing performance than the 4 core Opteron.

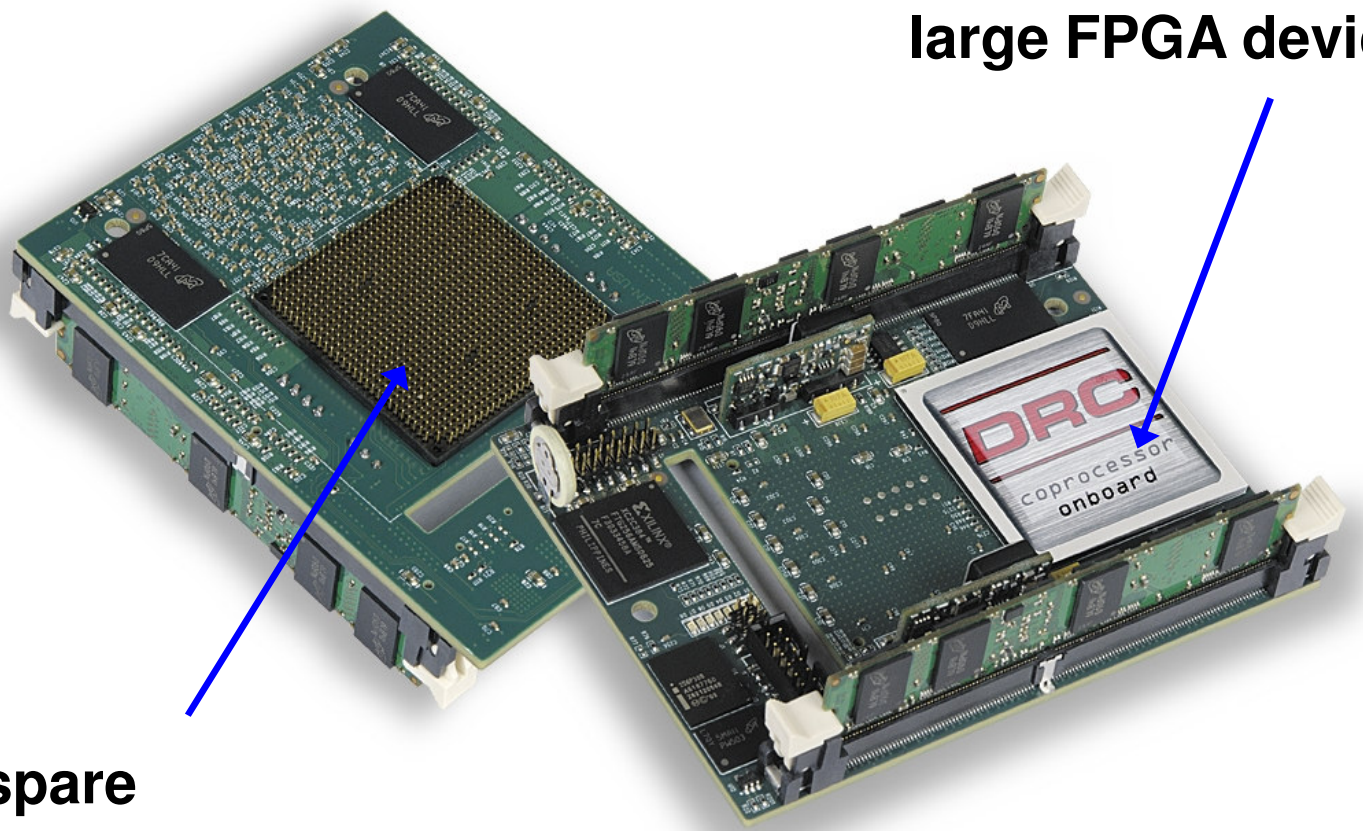
FPGA sweet spots

- Extremely good at integer, variable bit-width, variable precision
- High through-put streaming applications
- Long pipelines
- Parallel computing
- Wide, customizable high speed memory interfaces – where memory bandwidth normally is a problem
- Low power – 10-15 Gflops/watt for SP and much better ratio compared to CPUs for integers

Synective Labs

FPGA based coprocessor modules

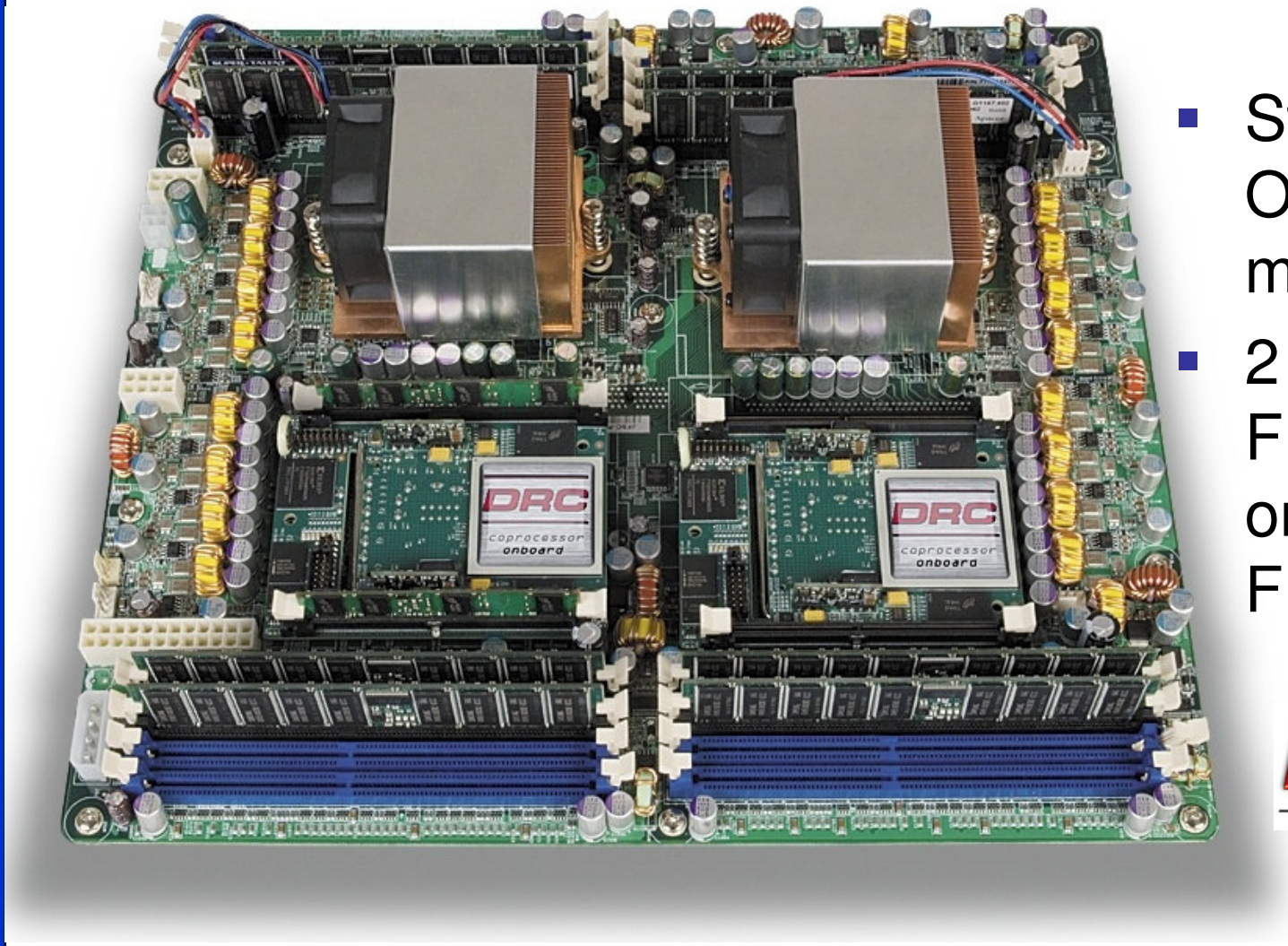
Reprogrammable large FPGA devices



Fits in spare processor sockets

Synective Labs

4 Way FPGA-module solution by DRC



- Standard 4-way Opteron motherboard
- 2 CPUs and 2 FPGA modules or 1 CPU and 3 FPGA modules

DRC Computer

Synective Labs

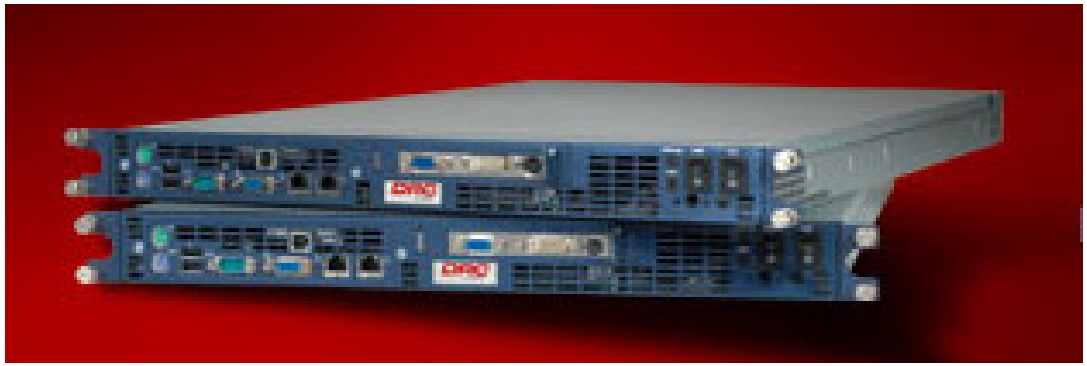
Avoiding the ”data transfer” bottleneck

- By putting the FPGA in a processor socket, it shares memory with the CPU
- The bottleneck of transferring data to and from the FPGA is eliminated – the FPGA can operate directly on the data in the motherboard’s memory
  - Passing a pointer instead of copying the data
- Very tight coupling between the CPU and the FPGA
- Uses HyperTransport (AMD) or QuickPath (Intel)

Synective Labs

FPGA accelerator alternatives

- CPU socket modules available for both AMD and Intel platforms from several vendors
- PCIe based solution with multiple FPGAs per board
- Solutions fits in 1U servers



FPGA programming tools

- Traditionally - VHDL and Verilog
  - Hardware design languages
  - Give you full control
  - Time consuming
  - Give you normally best performance

Synective Labs

HLL alternatives

- Quite a few HLL alternatives:
  - ImpulseC
  - MitrionC
  - Ylichron
  - Dime-C
  - Mobius

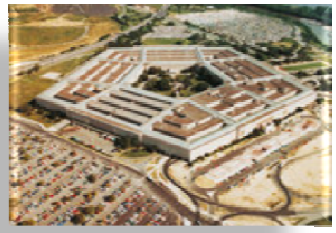
  … and many more

In most cases you pay a performance penalty by using the HLL tools, but you save a lot of development time!

Synective Labs

Examples on the DRC platform

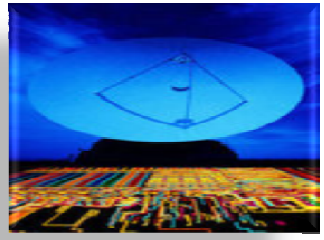
100x in ticker plants
60x Monte Carlo



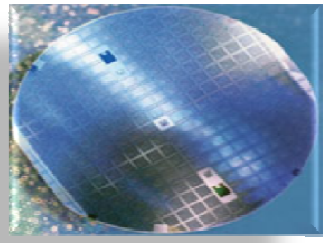
>100x in Security & Encryption

30 – 50x - RTM for seismic

>30x in Parameterized Search
>60x in Deep Sort algorithms



> 40x for convolution



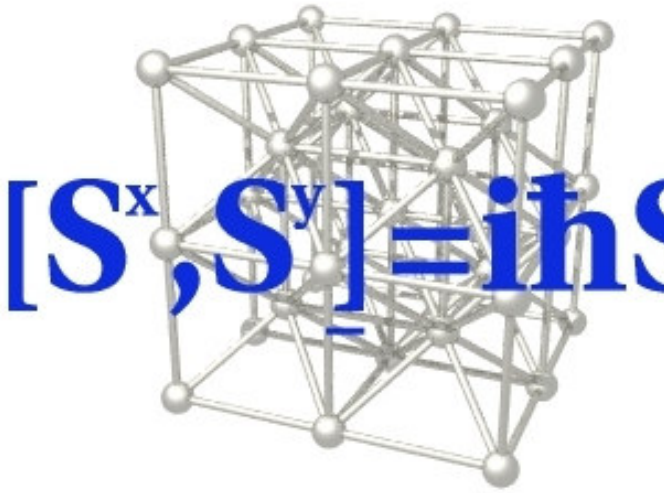
7x7 Median Filter - 3,500x

Gene Sequencing - 60x

Synective Labs

SPINPACK

What's about?



SPINPACK is a big program package to compute lowest eigenvalues and eigenstates and various expectation values (spin correlations etc) for quantum spin systems. These model systems can for example describe magnetic properties of insulators at very low temperatures (T=0) where the magnetic moments of the particles form entangled quantum states. The first SPINPACK version was based on Nishimori's TITPACK (Lanczos method), but it was early converted to C/C++ and completely rewritten. Other algorithms are implemented too. See 2x2-diagonalization for example. It is able to handle Heisenberg, t-J, and Hubbard-Systems up to 64 sites if you have enough memory, disk space and computing power. For instance we were able to get the lowest eigenstates for the Heisenberg Hamiltonian on a 40 site square lattice on our machines. The package is written mainly in C to get it running on all unix systems. C++ is only needed for complex eigenvectors and twisted boundary conditions.
The program can use all topological symmetries, S(z) symmetry and spin inversion to reduce matrix size. The results are very reliable because the package has b used since 1997 in scientific work.

Parallel processing (pthreads) can be used with version 1.9 for shared memory multiprocessor machines. Since version 2.30 MPI can also be used, but is of co slower.

News

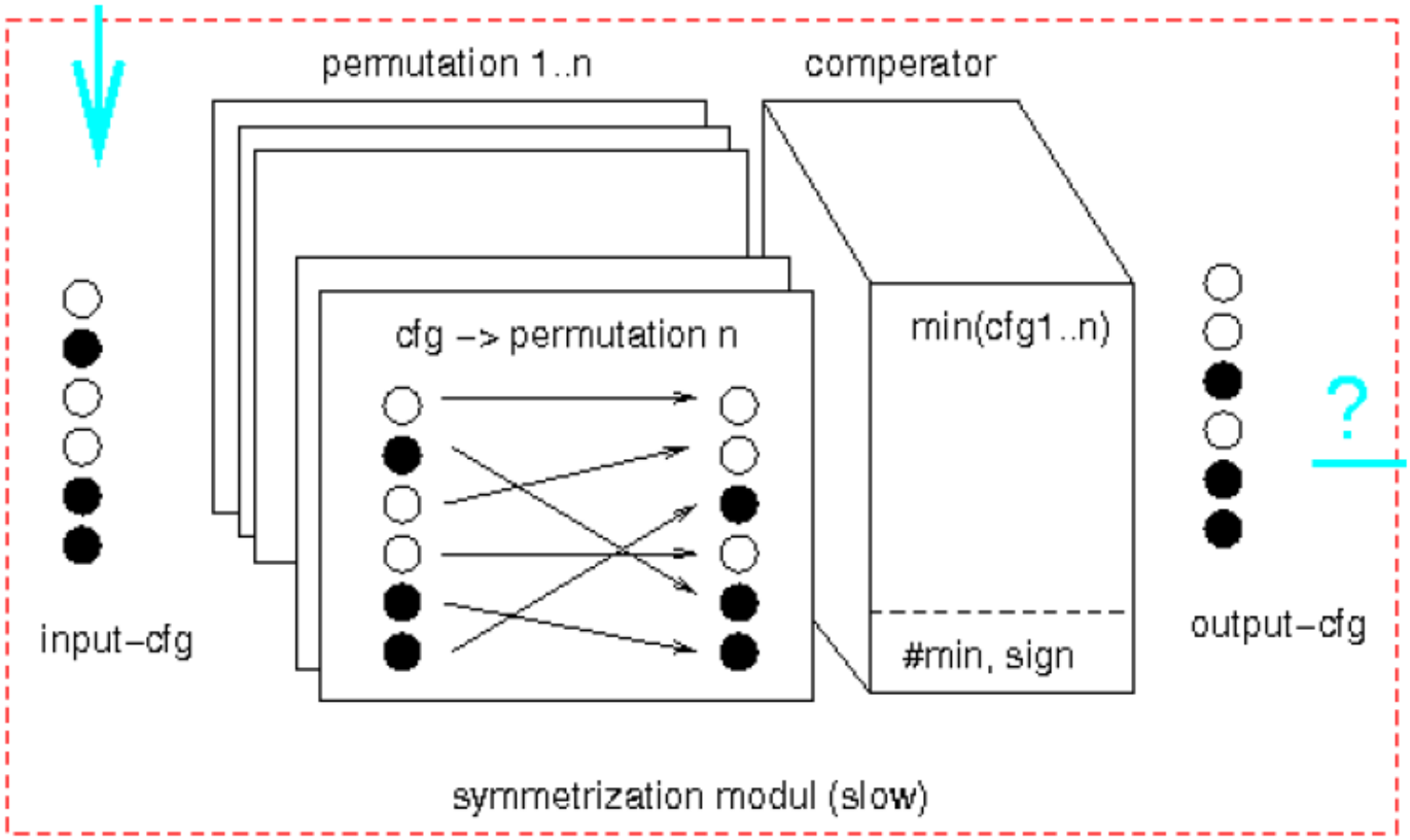
- Groundstate of the S=1/2 Heisenberg AFM on a N=42 linear chain computed (E0/Nw=-0.22180752, Hsize = 3.2e6, v2.38, Jan2009) using 900 Node SiCortex SC5832 700MHz 4GB RAM/Node (320min).
- Groundstate of the S=1/2 Heisenberg AFM on a N=42 square lattice computed (E0 = -28.43433834, Hsize = 1602437797, ((7,3),(0,6)), v2.34, Apr using 23 Nodes a 2*DualOpteron-2.2GHz 4GB RAM via 1Gb-eth (92Cores usage=80%, ca.60GB RAM, 80MB/s BW, 250h/100It).
- Program is ready for cluster (MPI and Pthread can be used at the same time, see the performance graphic) and can again use memory as storage media performance measurement (Dec07).
- Groundstate of the S=1/2 Heisenberg AFM on a N=40 square lattice computed (E0 = -27.09485025, Hsize = 430909650, v1.9.3, Jan2002).
- Groundstate of the S=1/2 J1-J2-Heisenberg AFM on a N=40 square lattice J2=0.5, zero-momentum space: E0= -19.96304839, Hsize = 430909650 memory, 185GB disk, v2.23, 60 iterations, 210h, Altix-330 IA64-1.5GHz, 2 CPUs, GCC-3.3, Jan06)
- Groundstate of the S=1/2 Heisenberg AFM on a N=39 triangular lattice computed (E0 = -21.7060606, Hsize = 589088346, v2.19, Jan2004).
- Largest complex Matrix: Hsize=1.2e9 (26GB memory, 288GB disk, v2.19 Jul2003), 90 iterations: 374h alpha-1GHz (with limited disk data rate, 4 CP til4_36)

Spinpack

- Work done together with Jörg Schulenburg at the University of Magdeburg
- His research is within atomic spin modelling
- Has developed “spinpack”
- He used ImpulseC + a DRC system, and some support from us, and implemented his algorithms in just a few days
- Result 300 times speed-up!

Synective Labs

Spinpack



The computational heavy part of this code is about generating a Hilbert space: make 160 bit-permutations and find the smallest value by comparison

Synective Labs

Spinpack

- Ideal for FPGA implementation
  - Bit-manipulations
  - To permute bits does hardly consume any logic – it is only wires!
  - To permute and test for minimum value takes roughly 6000 x86 cycles but only 1 FPGA cycle!!
  - 3 GHz x86 vs 150 MHz FPGA – speed up 300X
  - Only 25% of the FPGA is used -> potential for an additional 3x speed up!

Questions?

Thank you!

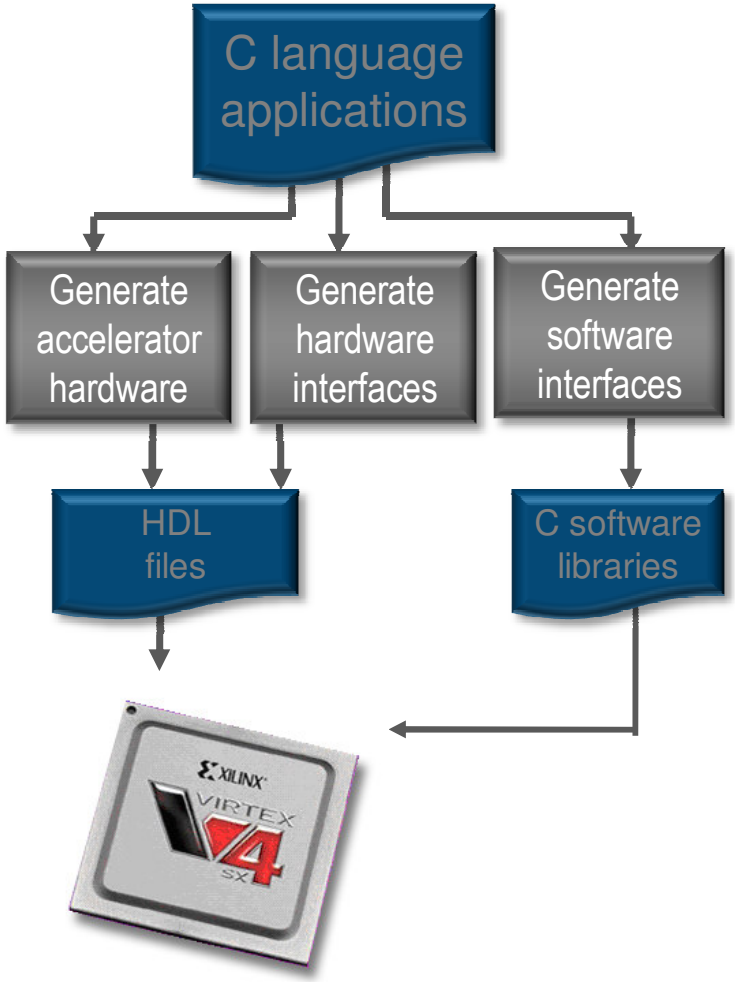
Synective Labs

Synective Labs

Experts in Application Acceleration

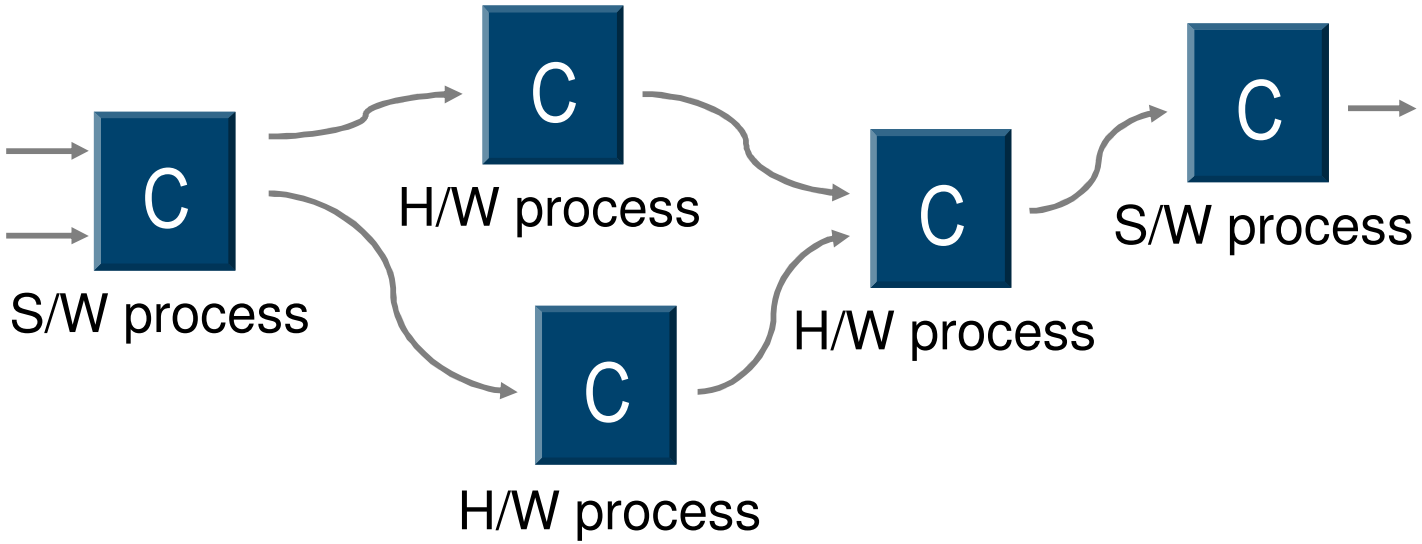
ImpulseC

- ANSI C for FPGA programming
  - Supports standard C development tools
  - Supports multi-process partitioning
  - Used with or without an embedded or host processor
- A software-to-hardware compiler
  - Optimizes C code for parallelism
  - Generates HDL, ready for FPGA synthesis
  - Generates hardware/software interfaces



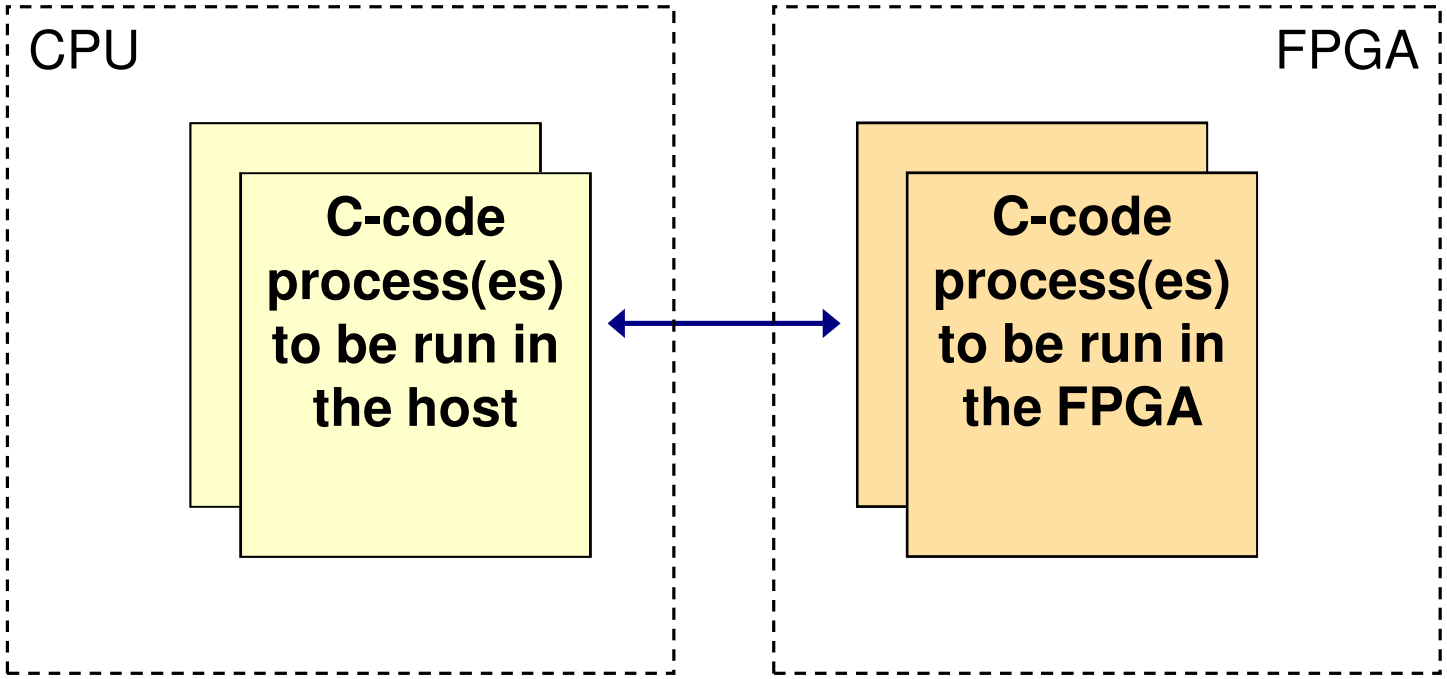
Synective Labs

Impulse C programming model



- Buffered communication channels to implement data streams
- Supports dataflow and message-based communications
- Supports parallelism at the application level and at the level of individual processes

Typical work flow



Typical work flow – debugging and testing