Building blocks of HPC cluster: Case for accelerators and new paradigms

Yogindra S Abhyankar
Senior Director
HPC-Technology group
C-DAC

54 TF
(Without Accelerators)

500+ TF
(With Accelerators)

Contents

- Why accelerators: Background
- What are Accelerators
- Accelerator types
  - GP-GPU
  - Xeon Phi (KNC, KNL)
  - Data Centre Accelerators
  - FPGA based accelerators
- Accelerator architecture building blocks

High Performance Computing (HPC)

- Computer Simulation +Theory & Experiments
  - Real experiments too small, large, complex, expensive, dangerous Or impossible

- Computational Sciences: *Multidisciplinary field*
  - Uses advanced computational capabilities to solve complex problems & understand, design
    - Protein folding/ computational biology
    - Climate, weather modelling
    - Astrophysics
    - Nano- Science

High Performance Computing (HPC) - 2

- HPC: Solving problems using
  - *Supercomputers +*
  - *fast networks +*
  - *large storage +*
  - *visualization*

- LINPACK: Bench mark for Top 500 supercomputers of the world
  - Summit@Oak Ridge National Laboratory (ORNL) USA 143.5 PF
  - Sierra@Lawrence Livermore National Laboratory (LLNL) USA 94.6 PF
    IBM Power9 CPUs and NVIDIA V100 GPUs.

Why use parallel Computers ?

- Only way to achieve computational goals
  - Sequential system is very slow
    - ⇒ Calculation takes days, months, years
    - √ ***Use more than one processor to get faster calculation***
  - Sequential system is very small
    - ⇒ Data does not fit in memory
    - √ ***Parallel system to accommodate more memory***

HPC Fastest Growing Sector

HPC, the massive horsepower of IT is one of the fastest growing sector in the Industry

- Rapidly growing data generated in the enterprise…
  - Every industrial sector

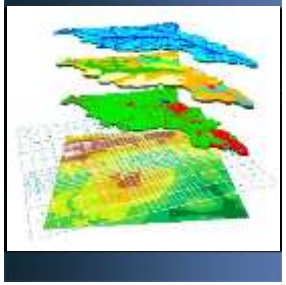
Every one wants fast results

HPC Segment Application Areas

Financial

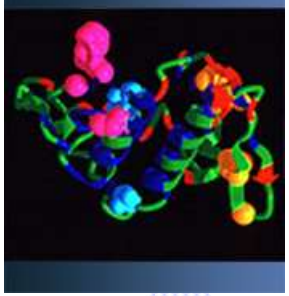
- Fin. Modeling
- Data Mining

Oil & Gas



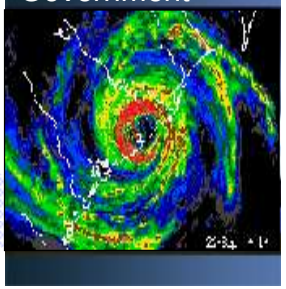
- Seismic
- Reservoir Mod

Materials & Life Sciences



- Molecular Dyn
- Med. Imaging
- DNA / iRNA
- Material Sci

Government

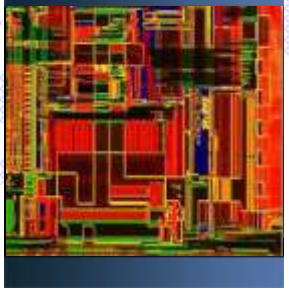


- Weather
- Crypto
- Data Mining

Film / Video

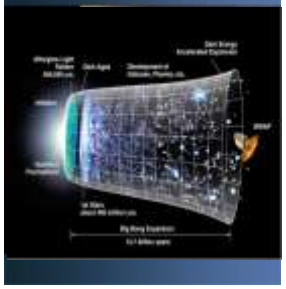
- Rendering
- Compositing

EDA



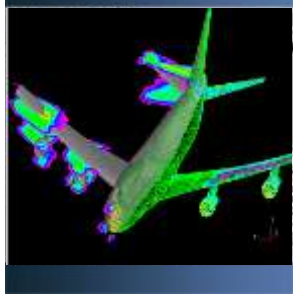
- Verification
- Layout

Scientific Research



- Cosmology
- Physics
- Math

CAE/CAD/CAM



- Structures
- Fluids
- Impact

Emerging Application Areas

..sometime not evident

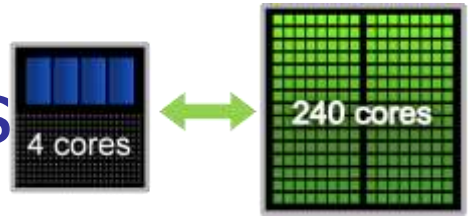
- ***Froth formation in the washing machine***
- ***Model production/Packaging of Pringles (potato chips)***

- ***Study Rotting (decay) of wood***

HPC is everywhere !!

Ref: SC

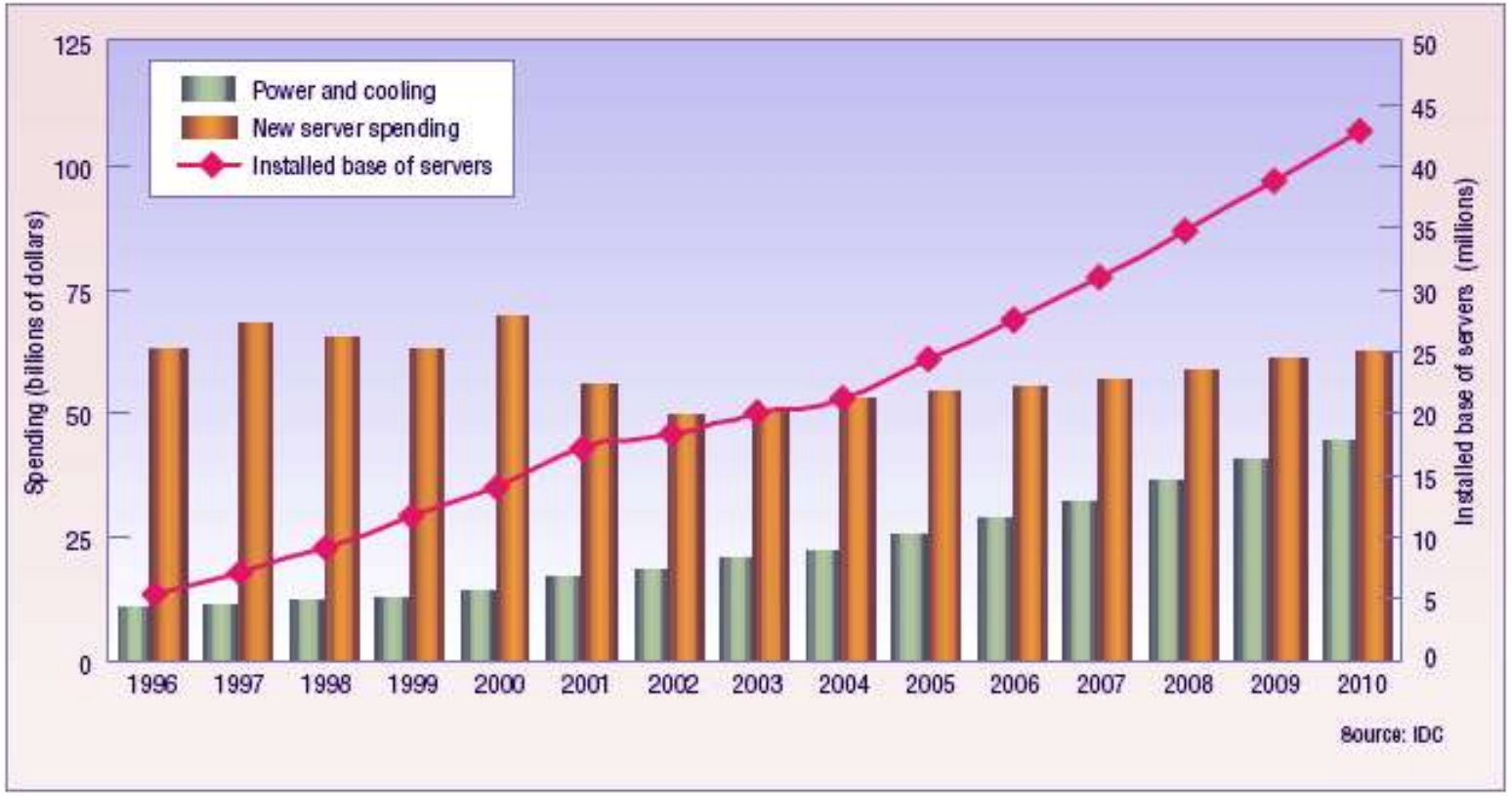
Background: Need for Accelerators



- Tackle large, complex computational problems
  - Multi-core to Many Core transition
- Large number of Computing Servers
  - Large rack Space
- High power consumption requirements

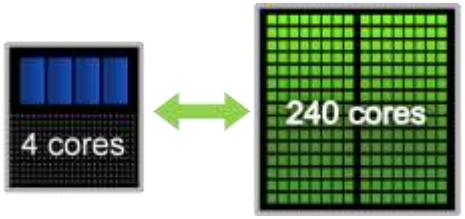
Computing cluster

Power Requirement Trends



Power and cooling costs increasing faster than equipment cost

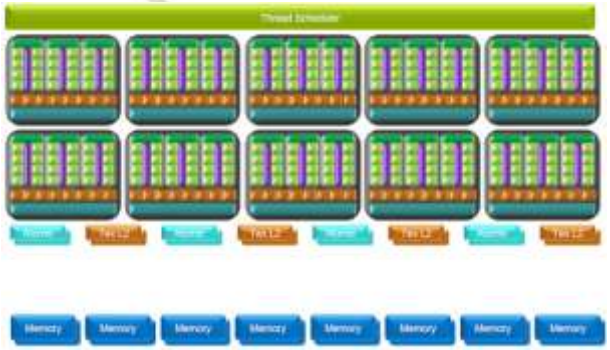
Accelerators (devices): I



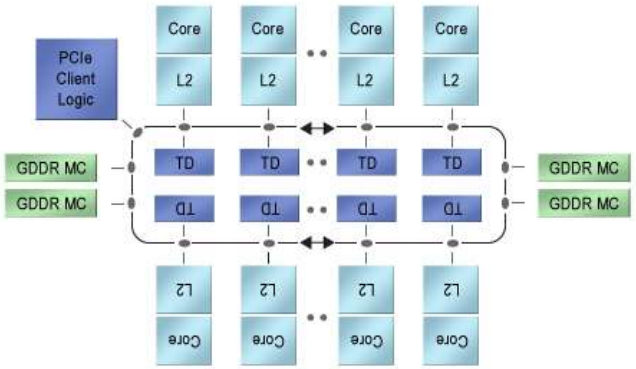
- Currently best alternatives to Increase computing power at the same time energy and space efficient
- Contain a large number of processing cores, as well as internal memory
- Most often used in conjunction with the CPUs of the node to accelerate certain 'hot spots' of a computation that requires a large amount of algebraic operations
- GPU, Xeon-phi and FPGA based

  - Large number of simple, low power cores, working together, slower frequency

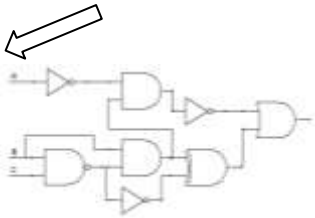
GPU

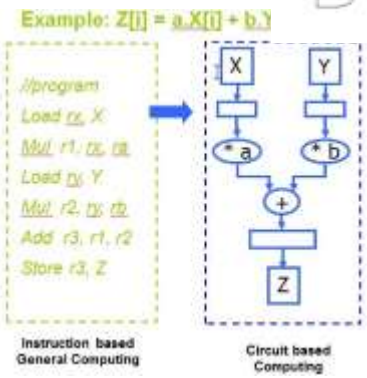


Intel

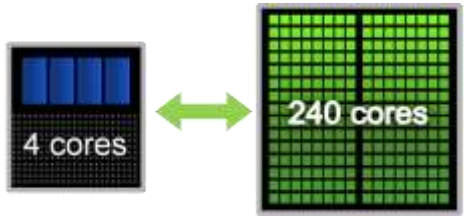


example: C-DAC card!





Accelerators (devices) : II



Historically

- Special hardware for accelerating computations :long tradition in HPC
  - Floating point units
  - SIMD/vector units

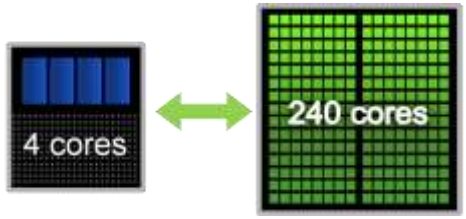
- Cell-chip based (chip jointly developed by Sony, Toshiba and IBM)
  - IBM RoadRunner Supercomputer

Hybrid system- general purpose CPU (AMD Opteron) + cell processors (PowerPC core)

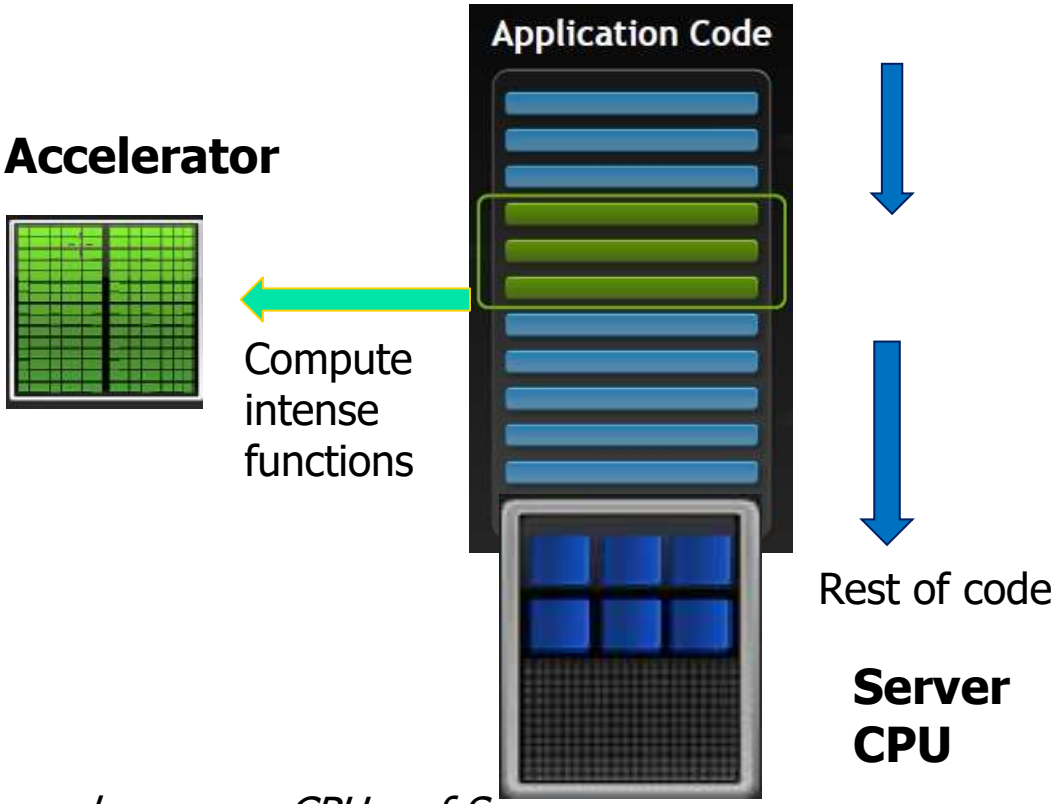
#1 system in June 2008 Top 500 list

1st Supercomputer to run PetaFlop speeds

Accelerators (devices): III



- Most often used in conjunction with the CPUs of the node to accelerate certain 'hot spots' of a computation that requires a large amount of algebraic operations
-



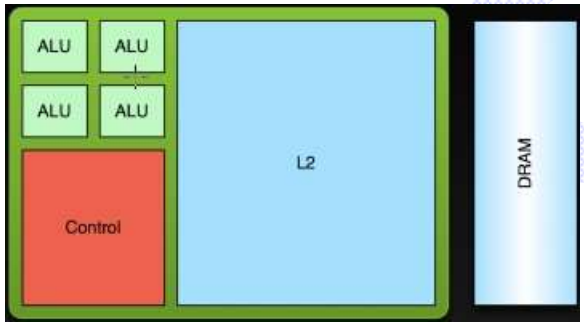
Part of the code runs on CPUs of Servers
Compute intense functions also called kernel run on accelerators

GPU Accelerators

Mainly popularized by NVIDIA & AMD for HPC ~2007

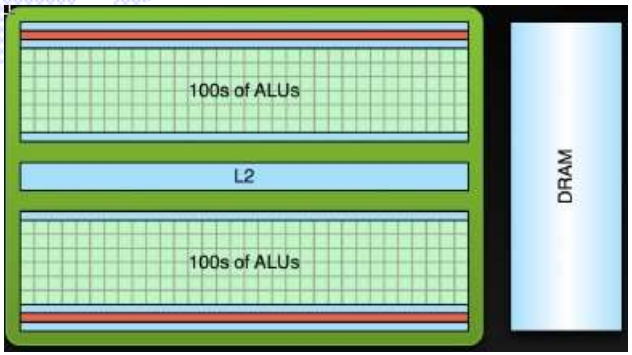
What is GPU Accelerated computing ?

- Using Graphics Processing Unit along with CPU to *accelerate applications*
  - Highly parallel many-core systems, processing large blocks of data in parallel



CPU

Optimized for low latency access



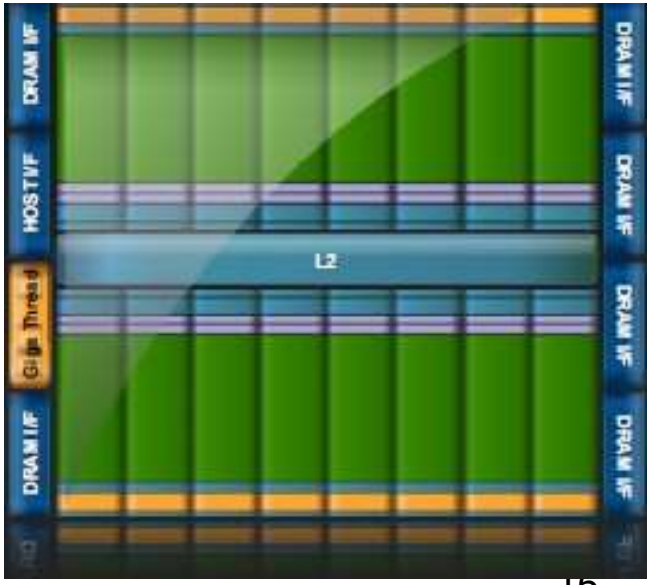
GPU

Optimized for data parallel, high throughput

GPU Architecture

Two main components

- Streaming Multiprocessors (SM)
  - perform computations
  - Control unit, execution pipeline, caches,..
- Global Memory
  - Analogous to RAM in server
  - Accessible to GPU & CPU



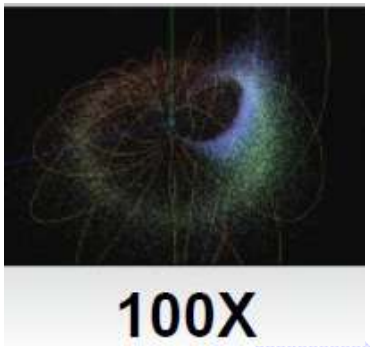
Ack: NVIDIA

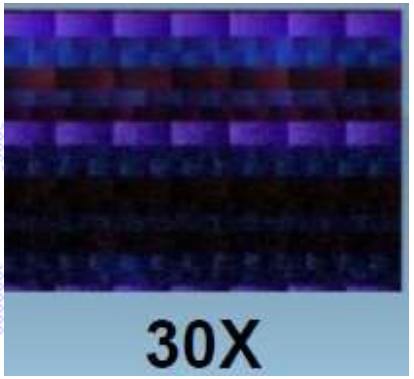
What kind of codes benefit ?

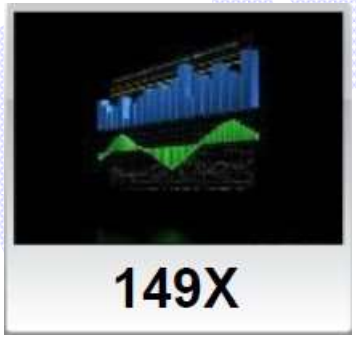
- Codes that spend lot of time in a task
  - If the task can be divided into hundreds of parallel sub-tasks
- Double precision performance for (√ accuracy)
  - Aerodynamics, Reservoir simulation, …
- Single precision performance for (√ speed)
  - Image rendering,

GPU Accelerated Applications

- Astrophysics
- Gene sequencing
- Financial analysis
- Visualization
- AI/Deep learning
- …

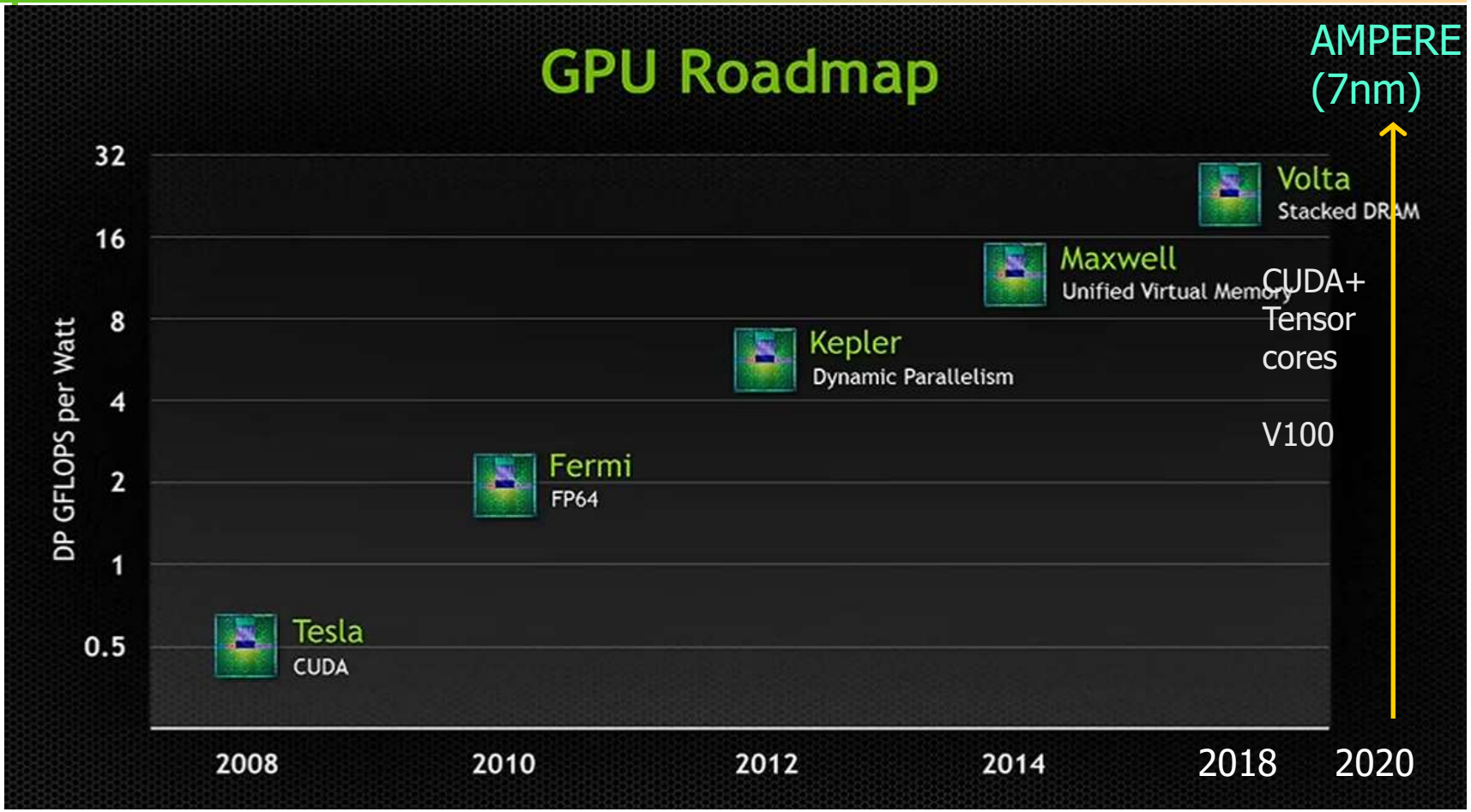






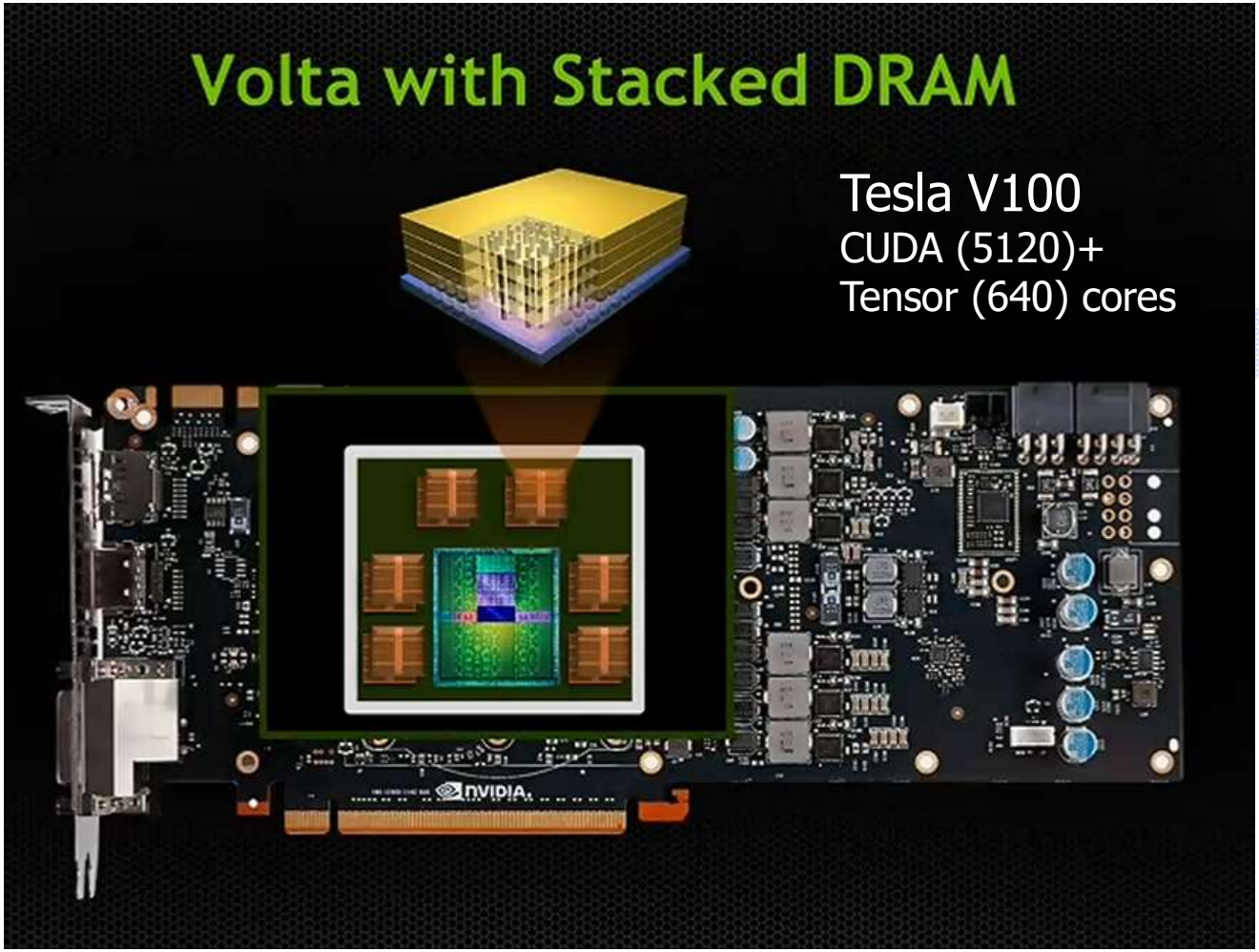
Ref: NVIDIA website

GPU roadmap (NVIDIA): 1



Ack: NVIDIA

GPU roadmap (NVIDIA): 2



Ack: NVIDIA

Accelerators – GPU (AMD) *Specifications*

- FirePro S9170 (for servers)
  - 2.62 TFLOPS (Peak) Double Precision
  - 5.24 TFLOPS (Peak) Single Precision

  Memory
  - Ultra fast, large 32 GB GDDR5 memory
    - Help to accelerate memory intensive applications, & computational complex workflows
  - Error Correcting Code (ECC) Memory

  Energy efficient
    - 275 W Power consumption (max)
  - Open CL 2.0 support

AcK: AMD http://www.amd.com/en-us/products/graphics/server/s9170

Specifications: Cooling/Power/Form Factor

- Bus Interface: PCIe x16
- Slots: Two
- Form Factor: Full height/ Full length
- Cooling: Passive heat sink

Specifications: System Requirements

- 20 CFM airflow cooling at 45° C maximum inlet temperature
- Available PCI Express® x16 (dual slot) 3.0 for optimal performance
- Power supply plus one 2x4 (8-pin) and one 2x3 (6-pin) AUX power connectors
- 2GB system memory
- Supported OS

How code/Application may use accelerator?

There are four main ways:

Applications

- Accelerator enabled libraries
- Directives /Pragma
- Explicit Programming languages
- Accelerator enabled Applications

How code/Application may use accelerator?

There are four main ways:

Applications

Accelerator enabled libraries

Only requires use of libraries (already written & available)

Ack: NVIDIA

How code/Application may use accelerator?

There are four main ways:

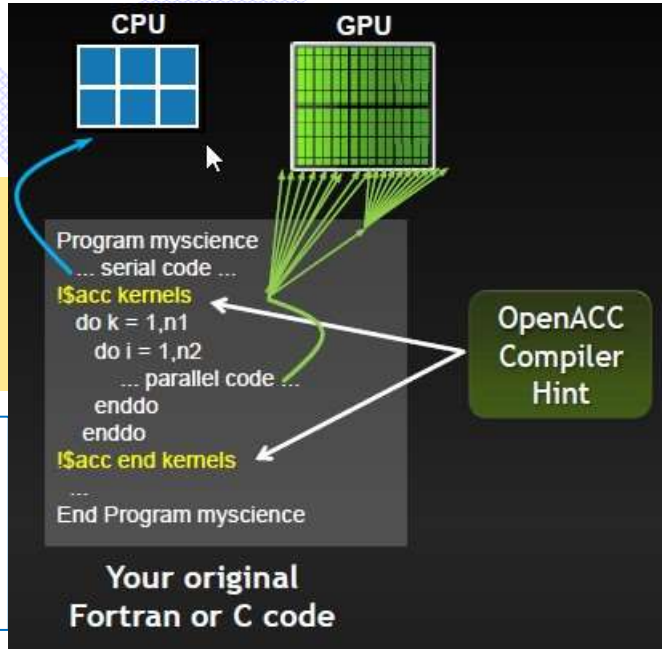
Applications

Accelerator enabled libraries

Directives /Pragma

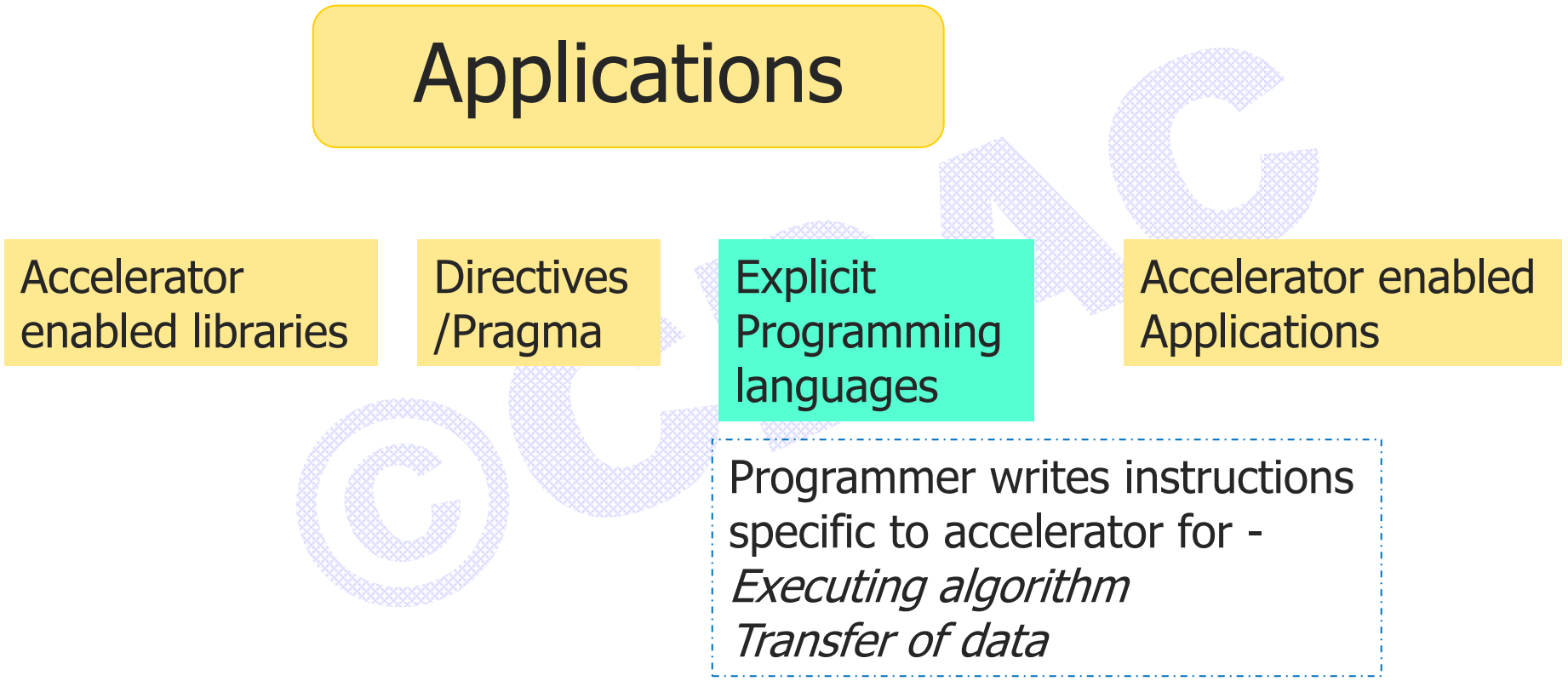
Explicit Programming languages

Accelerator code directly generated from source code by compiler (by adding hints)



How code/Application may use accelerator?

There are four main ways:



How code/Application may use accelerator?

Standard C Code

```
void saxpy_serial(int n,
                  float a,
                  float *x,
                  float *y)
{

  for (int i = 0; i < n; ++i)
    y[i] = a*x[i] + y[i];
}

// Perform SAXPY on 1M elements
saxpy_serial(4096*256, 2.0, x, y);
```

Parallel C Code

```
__global__
void saxpy_parallel(int n,
                    float a,
                    float *x,
                    float *y)
{
  int i = blockIdx.x*blockDim.x +
          threadIdx.x;
  if (i < n) y[i] = a*x[i] + y[i];
}

// Perform SAXPY on 1M elements
saxpy_parallel<<<4096,256>>>(n,2.0,x,y);
```

CUDA C

CUDA (Compute Unified Device Architecture)

Programmer writes instructions specific to accelerator for -
Executing algorithm
Transfer of data

GPGPU Programming: **Low-level**

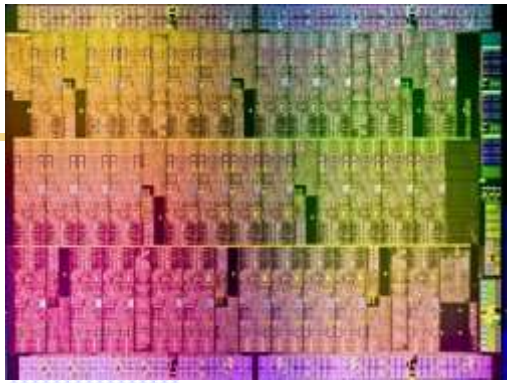
- Proprietary programming languages or extensions
  - NVIDIA: **CUDA** (C/C++ based)
  - AMD: StreamSDK or **Brooks+** (C/C++ based)
- Open Computing Language (**OpenCL**)
  - Open standard for portable, parallel programming of heterogeneous parallel computing
  - CPUs, GPUs, and other processors
- Major rewriting of the code required, not portable
- Best performance, usually only needed for Important kernels, Libraries

GPGPU Programming: **High-level**

- Compilation systems with (OpenMP-like) **directives for GPU programming**
  - User tells compiler which part of code to accelerate
  - Portland Group Fortran and C compilers
    - http://www.pgroup.com/resources/accel.htm
  - CAPS HMPP (Fortran, C)
    - http://www.caps-enterprise.com/hmpp.html
  - AMD: StreamSDK or **Brooks+** (C/C++ based)
- **OpenACC** joint-venture by:
  - NVIDIA
  - Portland Group
  - CRAY
  - CAPS

http://www.openacc-standard.org

Intel Xeon Phi – 1st generation



Intel 5110P (B1)

or

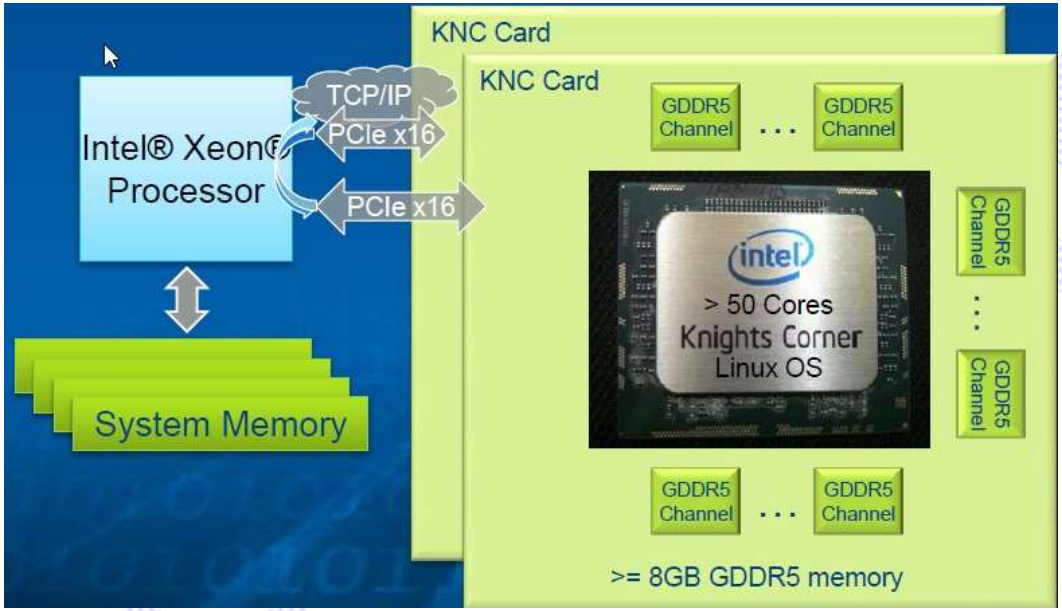
Many Integrated Core (MIC)

or

Knights Corner (KNC)

Xeon Phi Coprocessor

- Many core initiative from Intel
  - Available Cores (57/ 60/ 61,..) [based on 1st gen Pentium]

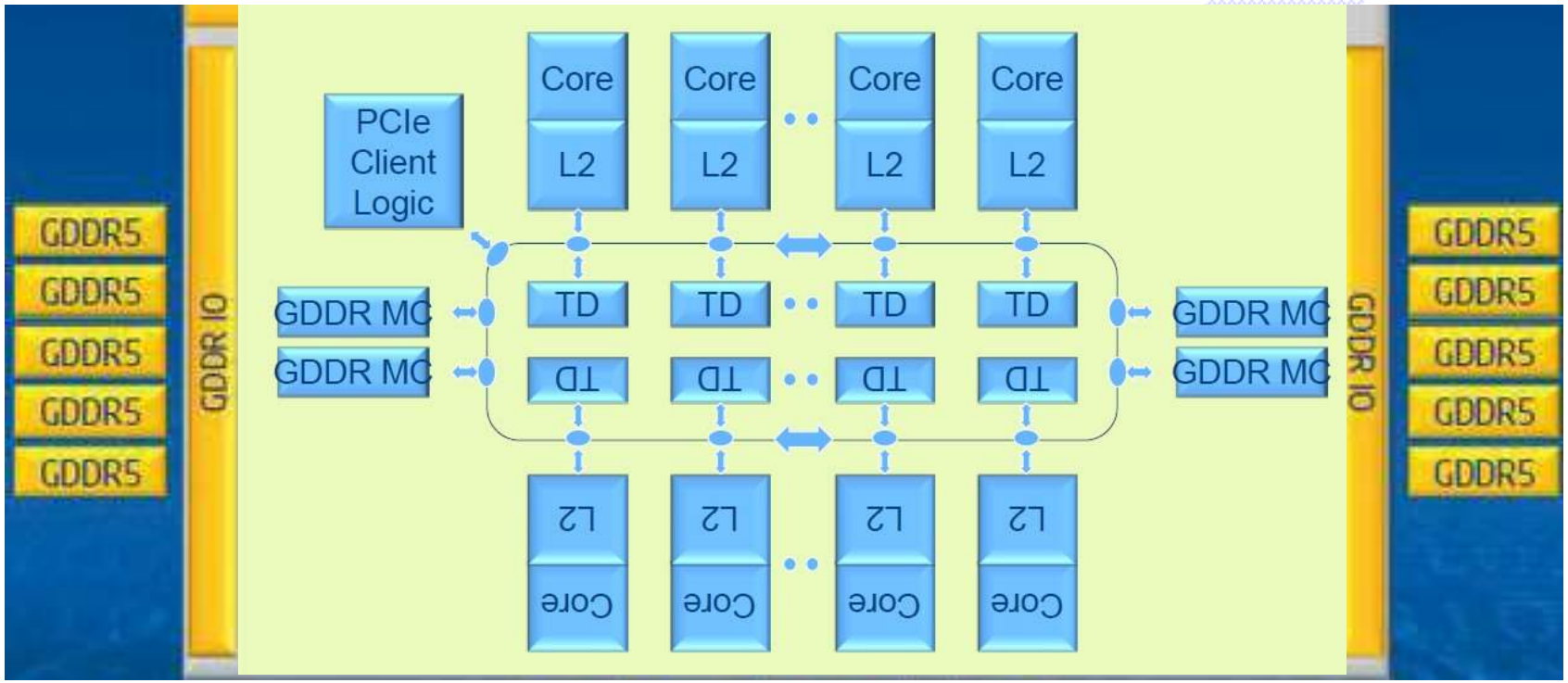


  - ~ 1 TF Peak; Clock Speed (1053/1100/1238 MHz)
  - Memory size (6 / 8/ 16 GB)
- PCI express card

Ack: James Reinders, Intel SC'12

Xeon Phi

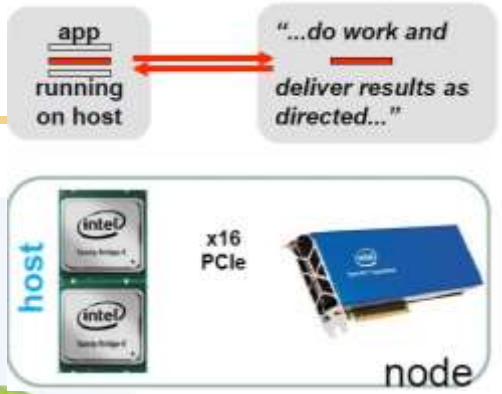
- Each core L1 cache 64KB, L2 cache 512KB
  - Cores interconnected by a ring bus

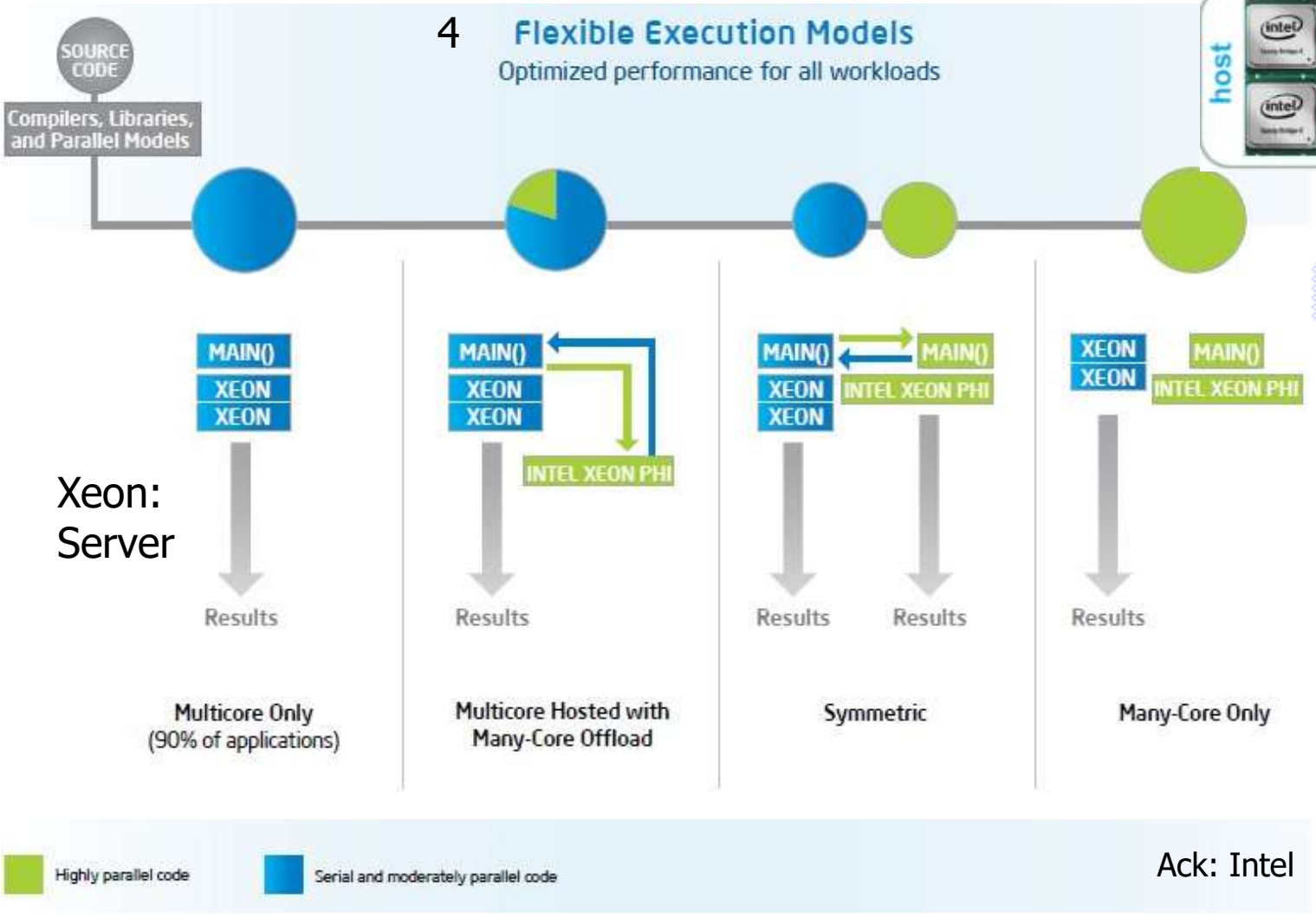


GDDR MC ➔ memory controller; GDDR5 – Graphic double data rate – type 5 synchronous memory

Ack: James Reinders, Intel SC'12

Intel Xeon Phi (Knights Corner)

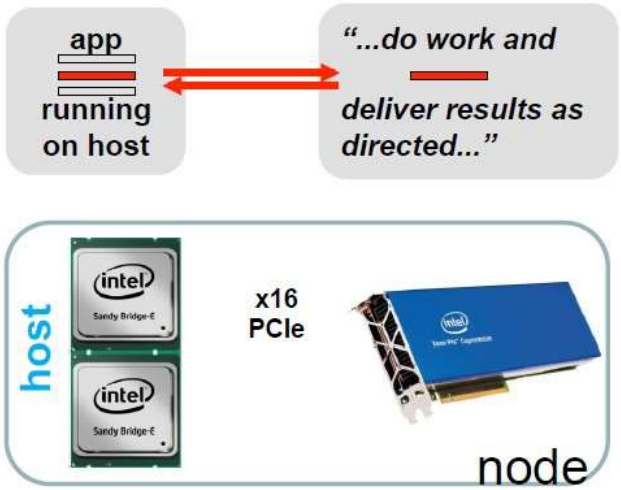




Xeon: Server

Ack: Intel

Offload directive Xeon Phi: sample snapshop



offloads

F90

```
program main

  use omp_lib

  integer :: nprocs

  !dir$ offload target(mic)
  nprocs = omp_get_num_procs()

  print*, "procs: ", nprocs

end program
```

offload directive

runs on MIC

runs on host

C/C++

```
#include <stdio.h>
#include <omp.h>

int main( void )  {

  int totalProcs;

  #pragma offload target(mic)
  totalProcs = omp_get_num_procs();

  printf( "procs: %d\n", totalProcs );
  return 0;

}
```

Ack: Intel

Xeon Phi Programming

- Based on "standard" programming models MPI, OpenMP, or MPI/OpenMP
  - On a set of MIC nodes
  - On a set of Cluster and MIC nodes
- Using offload directives
  - MPI program on Cluster nodes
  - Offloading (OpenMP) kernels to MIC nodes
- Various Intel proprietary programming models
  - Cilk Plus
  - TBB
  - OpenCL

Sample Xeon Phi Applications/performance

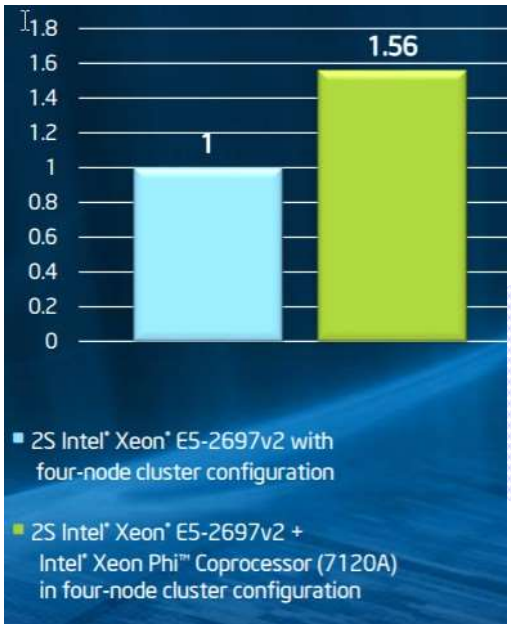
- Deep learning image classification training
- High performance ray tracing visualization
- Financial risk modelling

.....

Ref: https://newsroom.intel.com/newsroom/wp-content/uploads/sites/11/2016/06/intel-xeon-phi-product-family-performance-fact-sheet.pdf

Sample Application: WRF

- Weather Research & Forcast (WRF)



Ref: https://software.intel.com/en-us/articles/weather-research-and-forecasting-model-optimized-for-knights-landing
http://www.intel.com/performance/datacenter.

Accelerators: Many Core (Xeon-Phi)

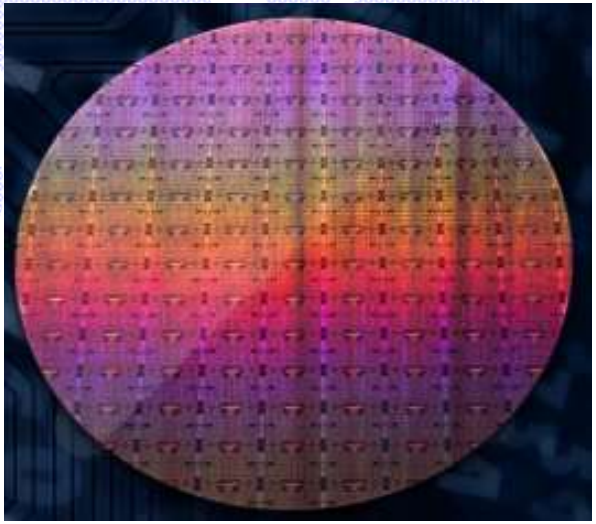
- "PARAM Yuva-II" , C-DAC : **500+ TF**
- Green500

  44th rank, Level 3 measurement

Greenest HPC System of India

Knights Landing (KNL)

KNL wafer
(typically size of a wafer is that of a dinner plate!
~ 12 inch diameter)

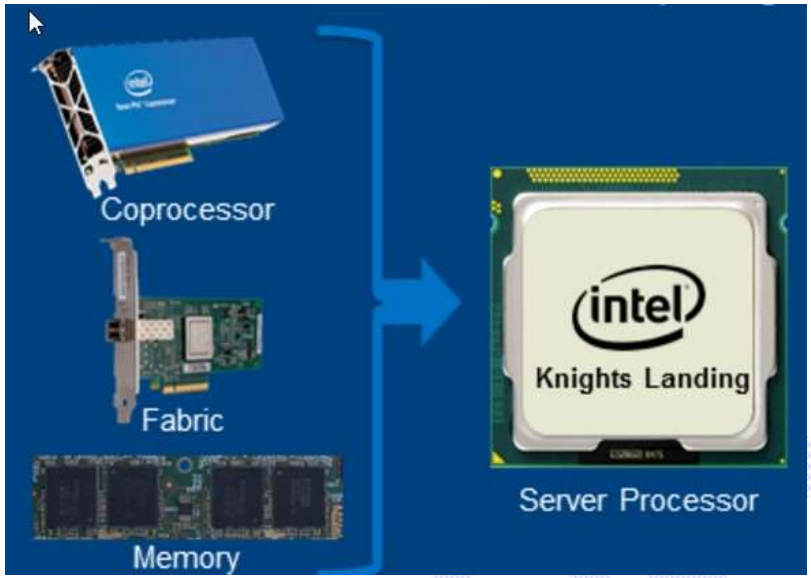


14 nm Process Technology

Motivation behind KNL ??

- Memory bandwidth, one of the bottleneck in computational Application performance
  - BLAS Level 1 & 2 (**B**asic **L**inear **A**lgebra **S**ubprograms) such as vector dot-product, matrix-vector multiplications…
  - Fast Fourier Transform (FFT),…
- Memory latency also addressed
  - Bringing near CPU
- Bootable host Chip
  - No PCIe bottleneck (limitation in the data send back and forth from the host CPU to the accelerator or coprocessor)

Intel Xeon Phi – 2nd generation Knights Landing (KNL)



3+ TeraFlops Double Preci

72 cores, 1.5 GHz

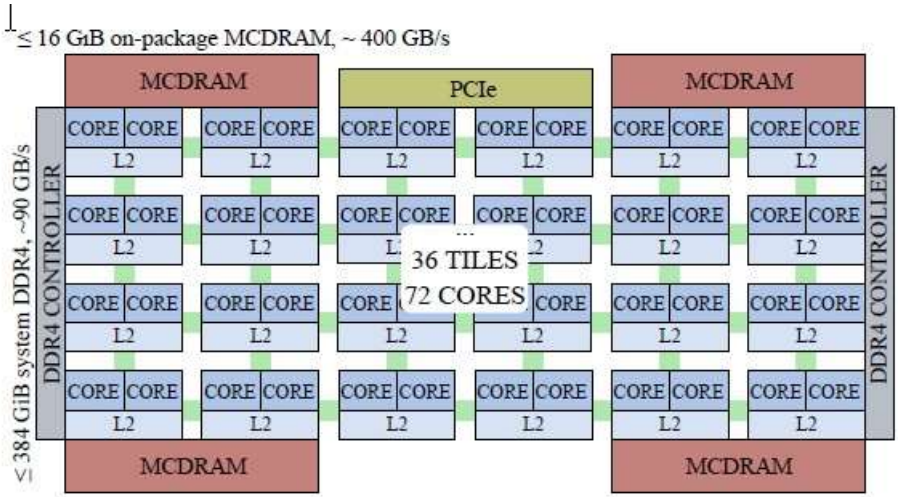
~ 245 Watt power

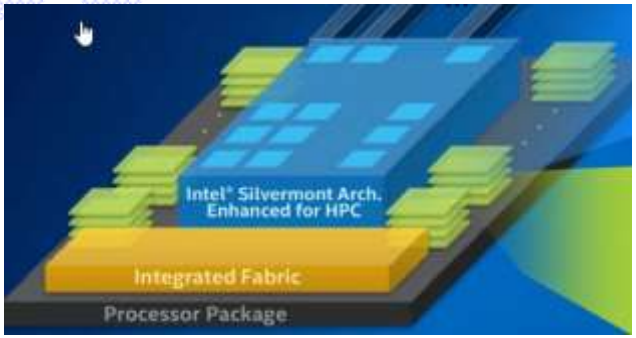
Power efficient (25%)

5x Memory bandwidth (DDR5)

Xeon Phi 2nd generation - KNL

- 72 cores (new Atom based)
- Tile structure (36)
  - 2 cores
  - L2 cache shared 2 cores
- Improved cache organization
  => complexity in chip HW
- High Bandwidth Memory (HBM)
  - MCDRAM (multi-Channel DRAM)





What applications run best on KNL ? 1

- Having high degree of parallelism & well-behaved communication with memory. Specifically,
- If memory traffic is negligible compared to the processing of arithmetic, the application is computebound
  - may run well on KNL due to its high arithmetic peak performance

What applications run best on KNL ? 2

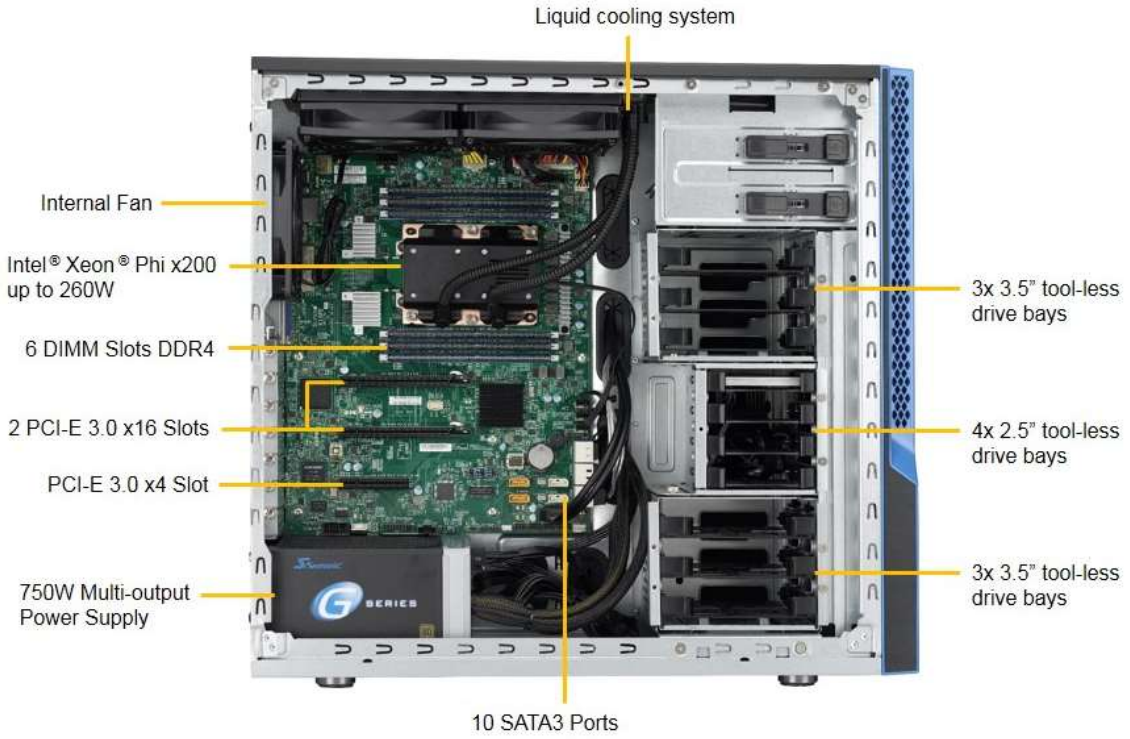
- If memory access has predictable, sequential pattern, the application is bandwidth-bound
  - run well on KNL due to its high-bandwidth memory (HBM)

What applications run best on KNL ? 3

- Application, neither compute-bound, nor bandwidth-bound because it has significant irregular memory access pattern, it belongs to the class of latency-bound applications
  - 1st generation Xeon Phi Knights Corner (KNC), performed poorly compared to Intel Xeon
  - KNL, improvements in cache organization reduce the impact of latency-bound operations

https://software.intel.com/en-us/articles/migrating-applications-from-knights-corner-to-knights-landing-self-boot-platforms

Workstation with KNL



Intel® Xeon® Phi™ x200 Processor with Up to 72 Cores and 16GB MCDRAM on package

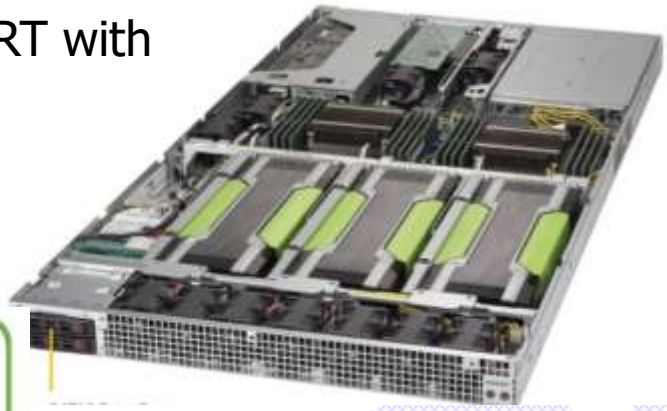
Super Workstation SYS-5038K-I with KNL

How many GPU/ Xeon Phi can be put in a server?

- PCIe slots available on the server ?
- Power supply rating ?
- Power connectors provided by the power supply
- Server must provide enough physical clearance for the cards and the power connectors

Server nodes with multiple GPU & Xeon-Phi

SuperServer 1028GQ-TRT with 4 GPU cards



Features at a Glance

- Dual 10-core Intel® Xeon® E5-2600 v2 Series Processors
- Up to 512GB of DDR3 Memory
- Supports up to 8 Intel® Xeon Phi™ coprocessors or 8 GPGPU cards utilizing proprietary PCIe switch

Features at a Glance

- Dual Intel® Xeon® Processor E5-2600 v3 Product Family
- Supports 8 GPU Graphics Cards in Dedicated PCIe x16 Gen 3 slots
- Up to 1TB of DDR4 Memory
- Up to 8TB of SATA Storage
- Maximum 96 GPUs per BladeRack*

Each node contains 16 GPU cards in one 3U enclosure cabled to a server through a PCIe Gen3 X16 Connection

Active versus Passive cooling

- Example: GPU

Passive
- No fan to take power
- No fan to get dusty
- No fan that can fail and destroy the card
- Quiet

- Active cooling gpu card are more powerful
- Better Performance than passive ones
- Runs continuously longer than the passive

Data Center Accelerators :

Optimizing Workloads

Intel Xeon Scalable Processors

for Visual Cloud

- Visual Compute Accelerator (VCA): VCA2 from **Intel**

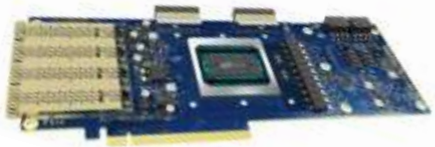
  Graphics rendering & Media transcode

  PCIe Gen3 add-in card, 235W TDP.
  3 x Intel Xeon processors E3-1585L v5
  Intel Iris Pro-graphics P580

Q2:2017

- Deep Learning/AI Accelerator: Intel Nervana platform

  with Neural Network processor



- Network optimization

  5G and other workloads

Ack: Intel

https://www.intel.com/content/www/us/en/products/docs/servers/accelerators/accelerators.html

FPGA Based HW Accelerators

Approaches for Faster Solutions

- Large clusters
- Application tuning
- Latest processors – Frequencies/Cores/Architectures..
- Change Algorithm to suite the **FIXED** hardware

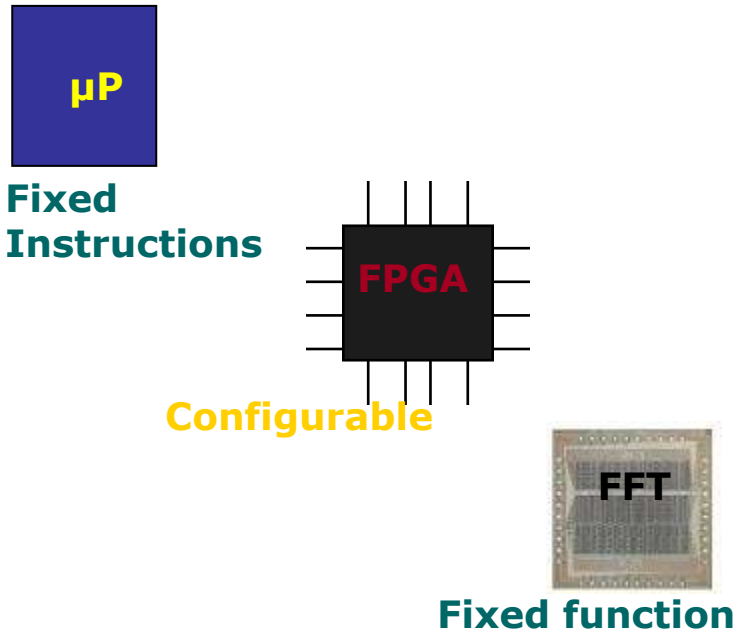
….***Reconfigurable Computing,*** *a* ***Novel*** *Approach*

for Speeding-up applications having

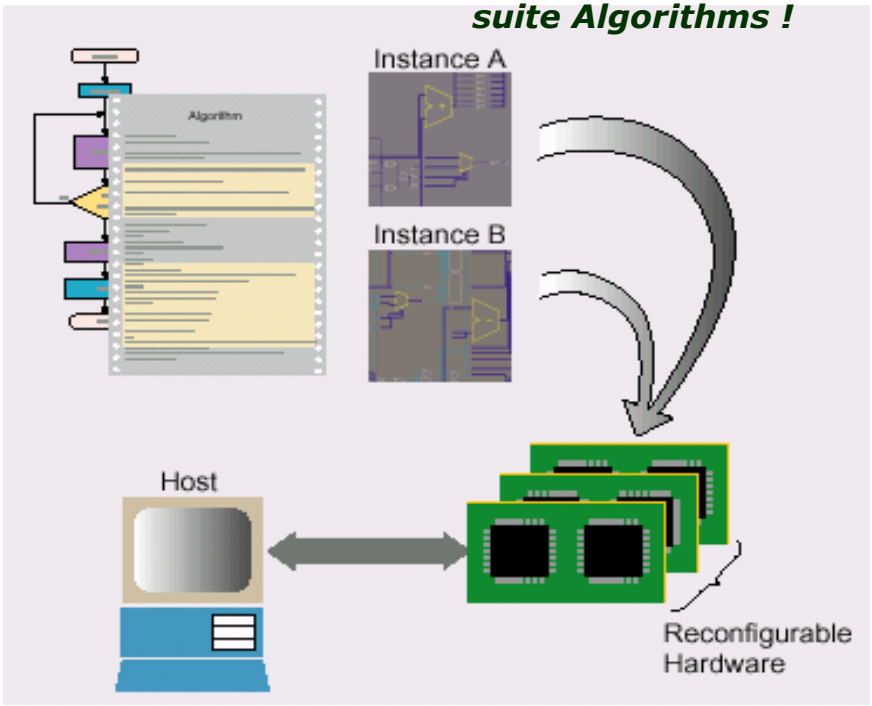
FLEXIBLE *hardware*

Reconfigurable Computing System

- ***Systems that dynamically modify their hardware for accelerating applications*** - Mainly based on **FPGAs**
  (Field Programmable Gate Array)



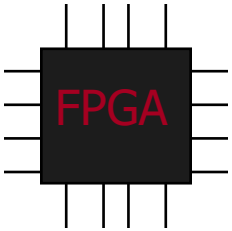
Hardware circuits to suite Algorithms !



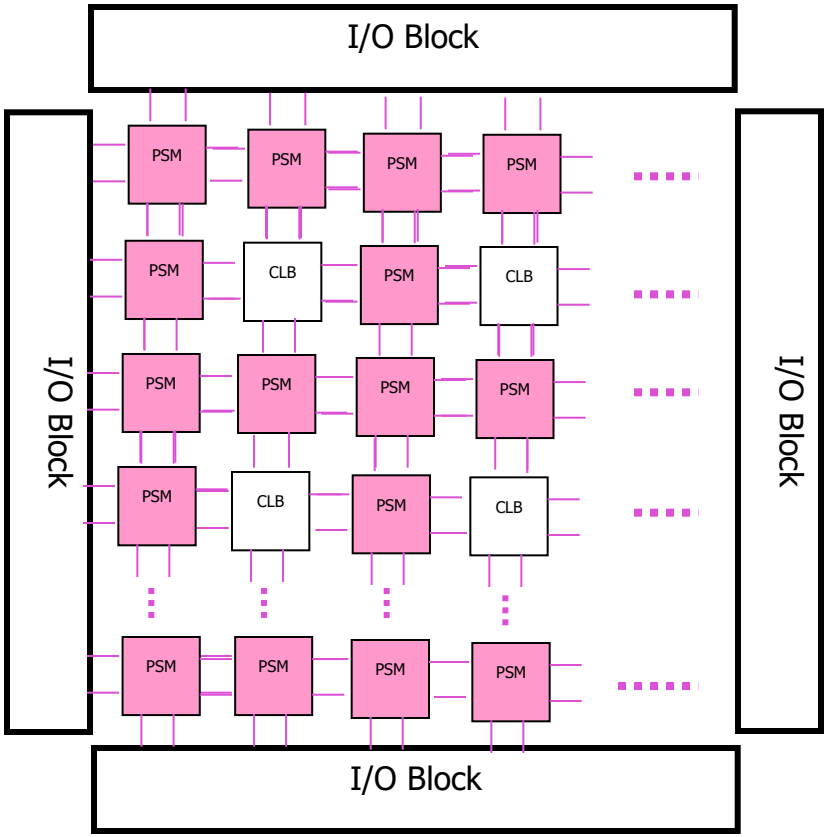
Field Programmable Gate Array (FPGA)

- ***A chip that can be configured by a user to implement different digital logic circuits***
- ***Configurable Logic Blocks and interconnects***

Invented
in 1984

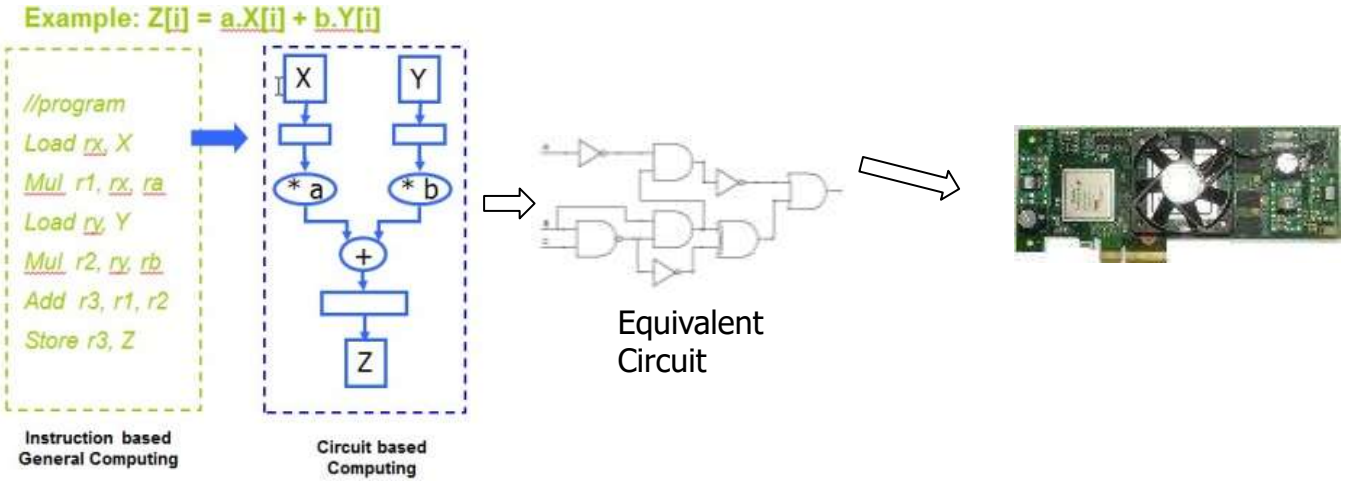


Configurable



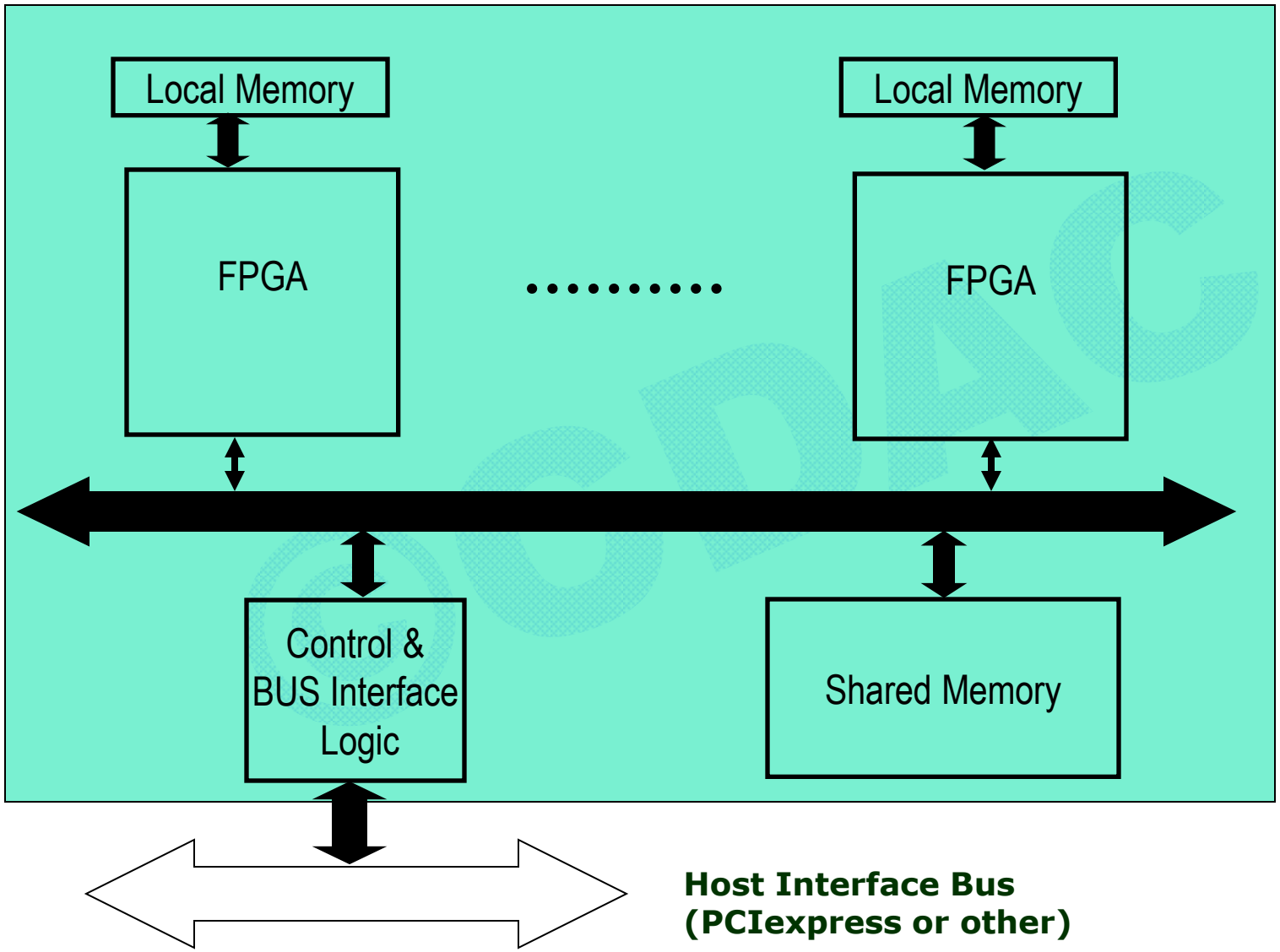
Reconfigurable Computing (RC) : **What is it ?**

- **Highly Energy Efficient**
  - Implements **Circuit** corresponding to algorithm/ compute portion of application rather than executing instructions.
  - Uses Field Programmable Gate Array (FPGA)

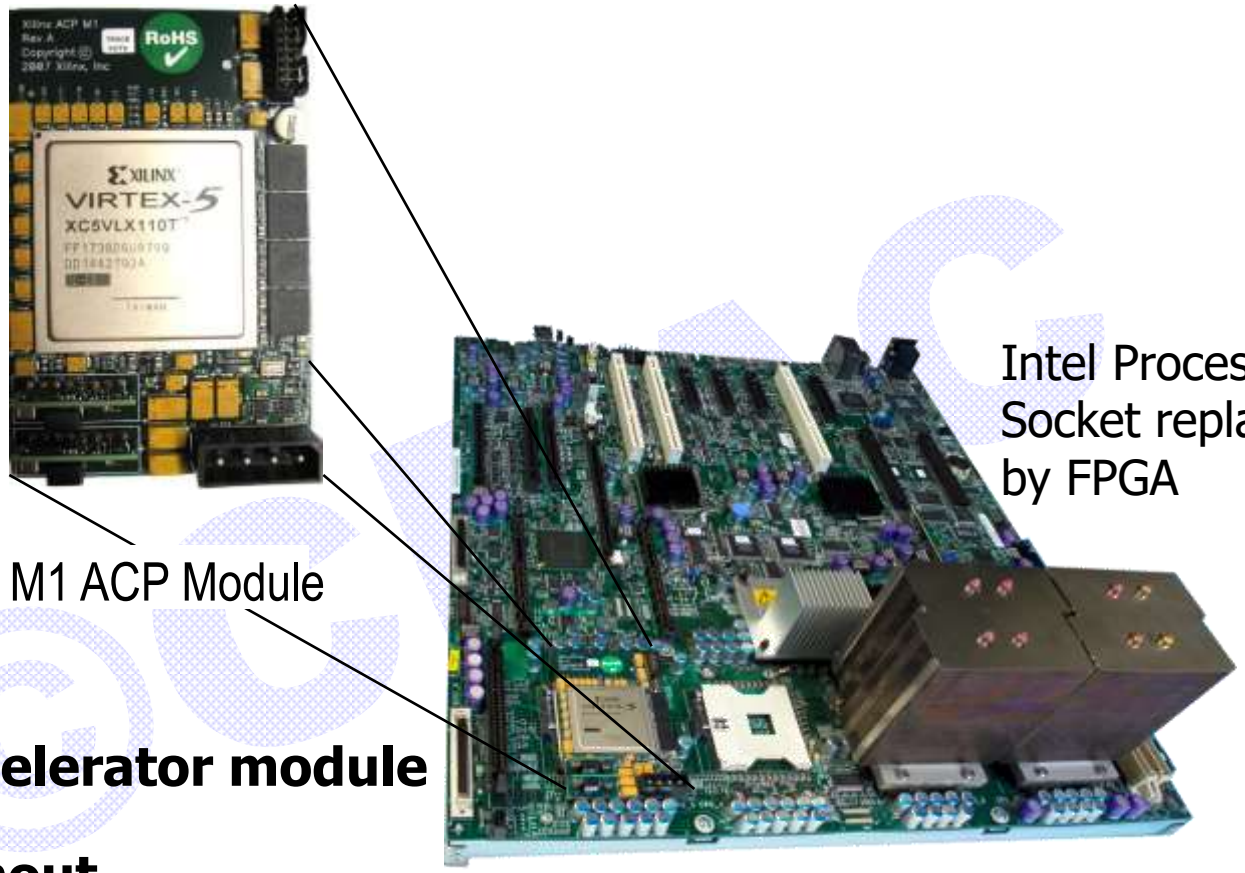


  - It doesn't work on fixed data widths/ boundaries as the processors, rather the widths are *customized* as per the *function*, saving lot of power

General RC Board Architecture



FPGA on (intel) Server Socket: Example 1



- **In socket accelerator module**
- **Xeon CPU pinout**
- **Intel FSB bus interface (soft logic) inside FPGA**

Acknowledgement: Xilinx

FPGA socket for AMD Opteron HT: Example2

AMD Processor Socket Replaced by FPGA

- **In socket FPGA based accelerator modules**
- **HyperTransport (HT) Socket interface**
- **Fits in AMD processor sockets**

Acknowledgement: DRC

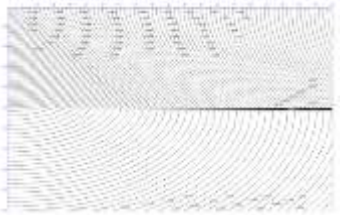
C-DAC FPGA based Reconfigurable HW Accelerators

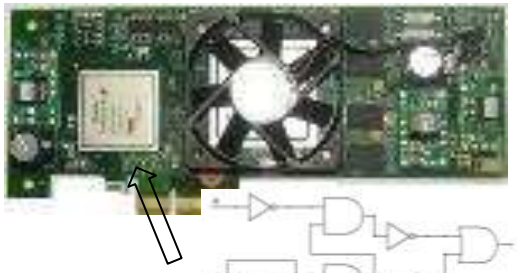
Accelerated solution provided in various areas like

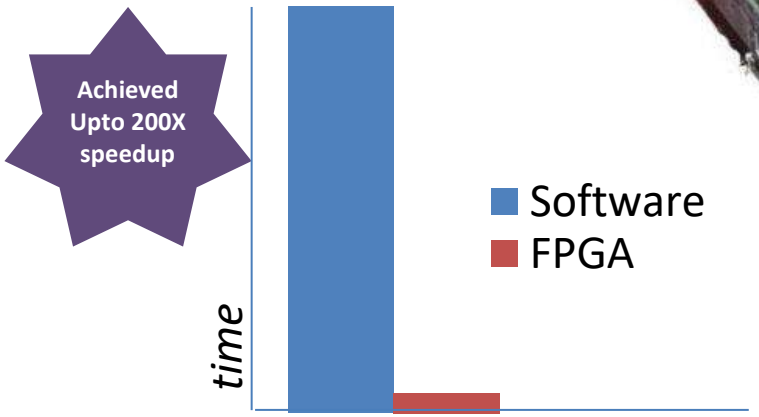
- ***Bioinformatics***
- ***Radio astronomy***
- **Fracture mechanics**
- **Scientific & engineering routines**
- **Crypto analysis**

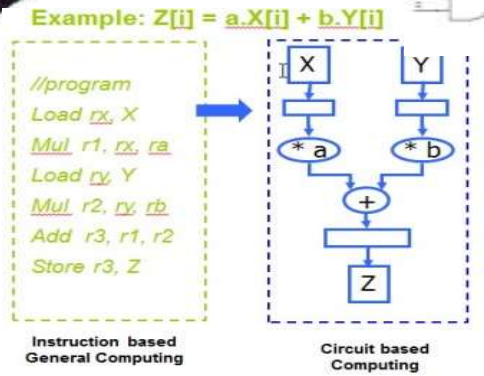
10X-200X … Acceleration





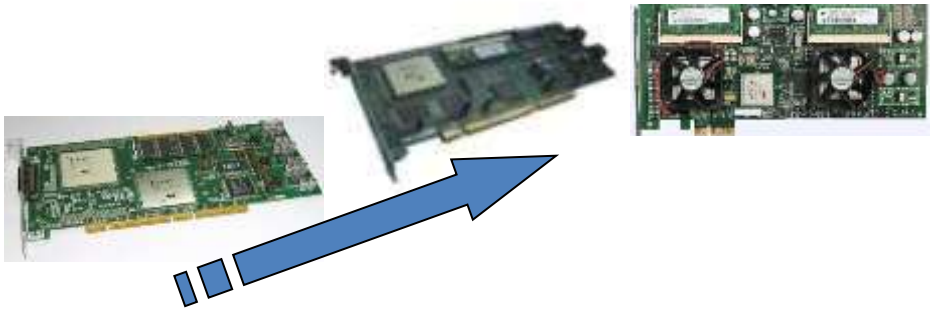




Design & Development of RC Solution Building Blocks

- **RC hardware**
  - *With state of the art FPGAs*

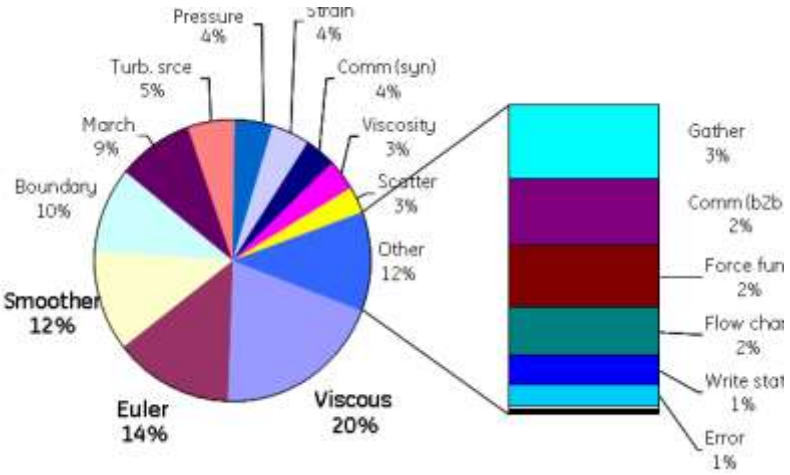


- **HW routines/libraries** (‘Avatars’)
  - *Application functions*

- **System Software** (‘VARADA’)
  - *APIs, Kernel Agent*
  - *Linux, Win 7 support*

How FPGA gives Application Acceleration

- **Selecting suitable application**
  - Application/ Kernel profiling
    - Compute intensive functions
    - Inner loops run many times
  - Analyze required precision
  - SW/HW Partitioning



- **Efficient HW logic**
  - Pipelined & Parallel
    - Hiding Computation & Communication latency
    - Many parallel blocks allows to overcome frequency limitation
  - Analyze required HW resources
  - High degree of Instruction efficiency

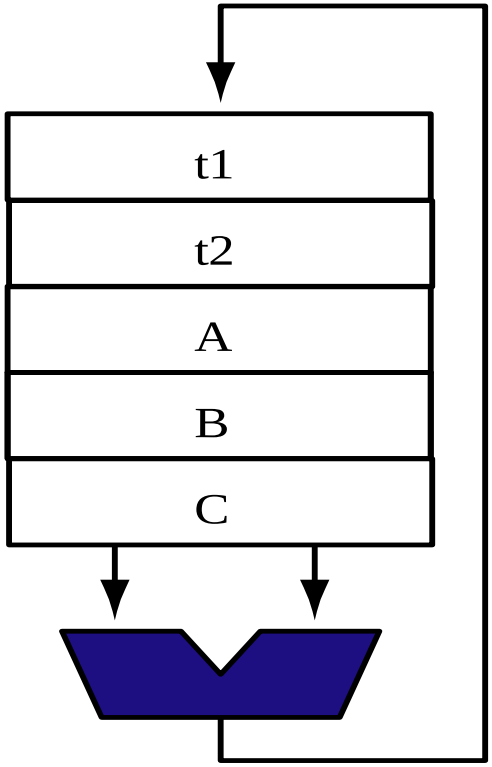
Spatial vs. Temporal Computation

Processors divide computation across time, dedicated logic divides across space

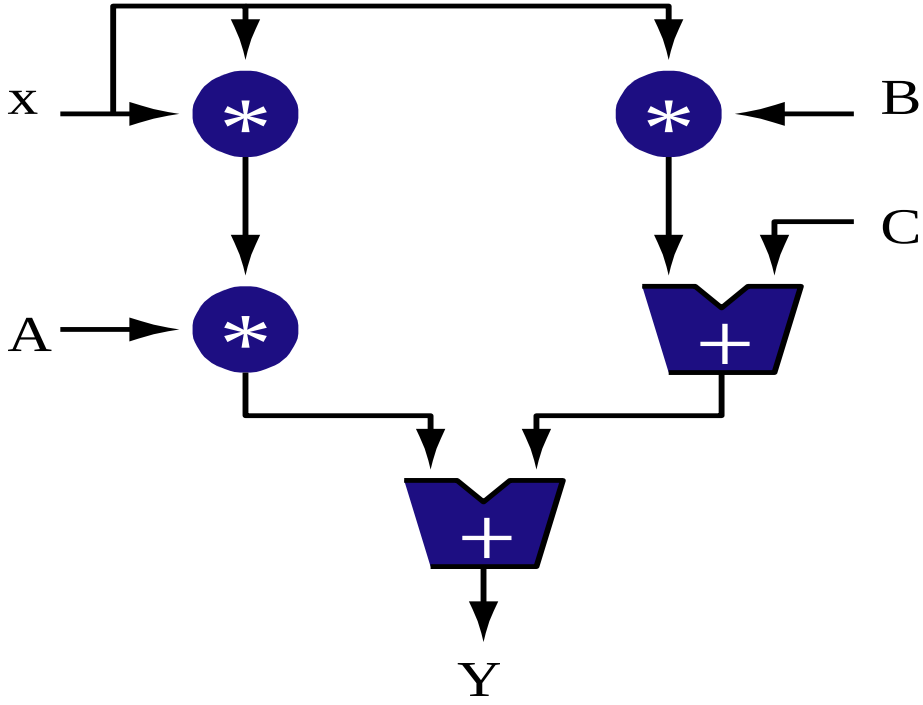
$$y = Ax^2 + Bx + C$$

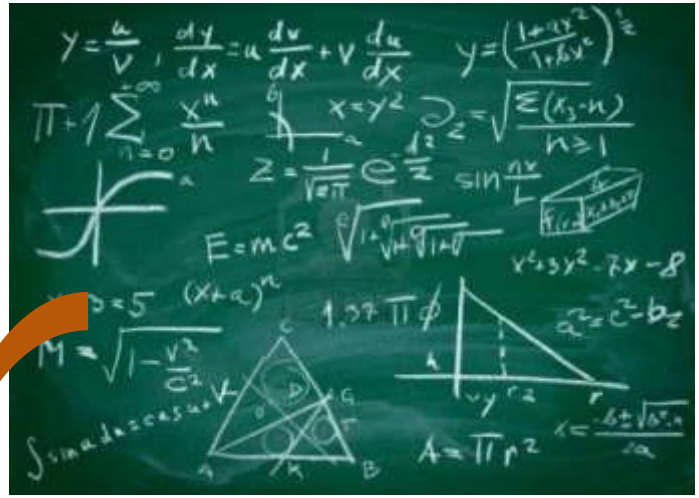
Temporal Computation

```
t1 = x
t2 = t1 * A
t2 = t2 + B
t2 = t2 * t1
y  = t2 + C
```

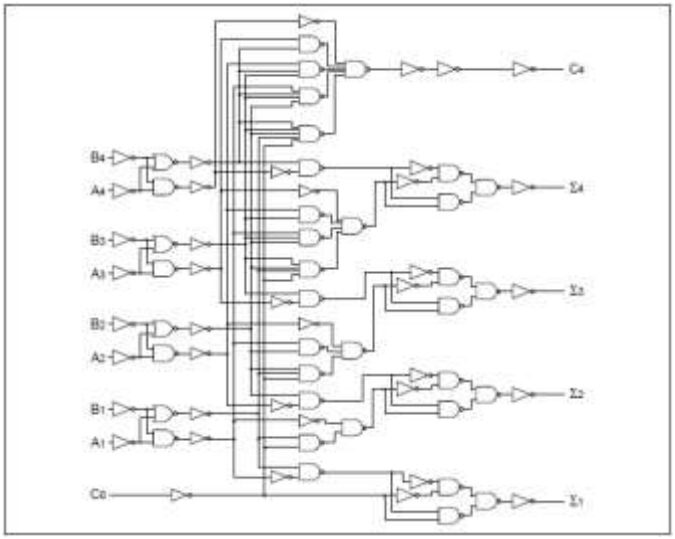


Spatial Computation



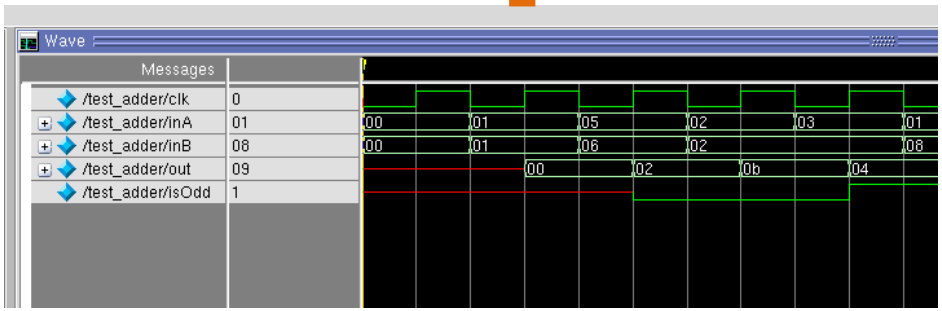


Design Flow



```
entity Test_Counter_VHDL is
    Port ( Clk_xxxHz :          in  std_logic;
           Step_Clk :           in  std_logic;
           Select_Clk :         in  std_logic;
           Clr, Count_Enable :  in  std_logic;
           Bcd0,Bcd1,Bcd2,Bcd3 : out std_logic_vector(3 downto 0));
end Test_Counter_VHDL;

architecture Behavioral of Test_Counter_VHDL is
   signal Q:   std_logic_vector( 15 downto 0);
   signal Clk: std_logic;
begin
   -- 2x1bit multiplexer: Clk_xxx or Step_Clk = [Btn0]
   Clk <= Clk_xxxHz when Select_Clk='1' else
          Step_Clk;

   process( Clk, Clr
      if Clr='1' the
         Q <= (others => '0');  -- "0000000000000000"
      elsif rising_edge( Clk) then
         if Count_Enable='1' then
            Q <= Q+1;
         end if;
      end if;
   end process;

   Bcd3 <= Q(15 downto 12);
   Bcd2 <= Q(11 downto  8);
   Bcd1 <= Q( 7 downto  4);
   Bcd0 <= Q( 3 downto  0);
end Behavioral;
```

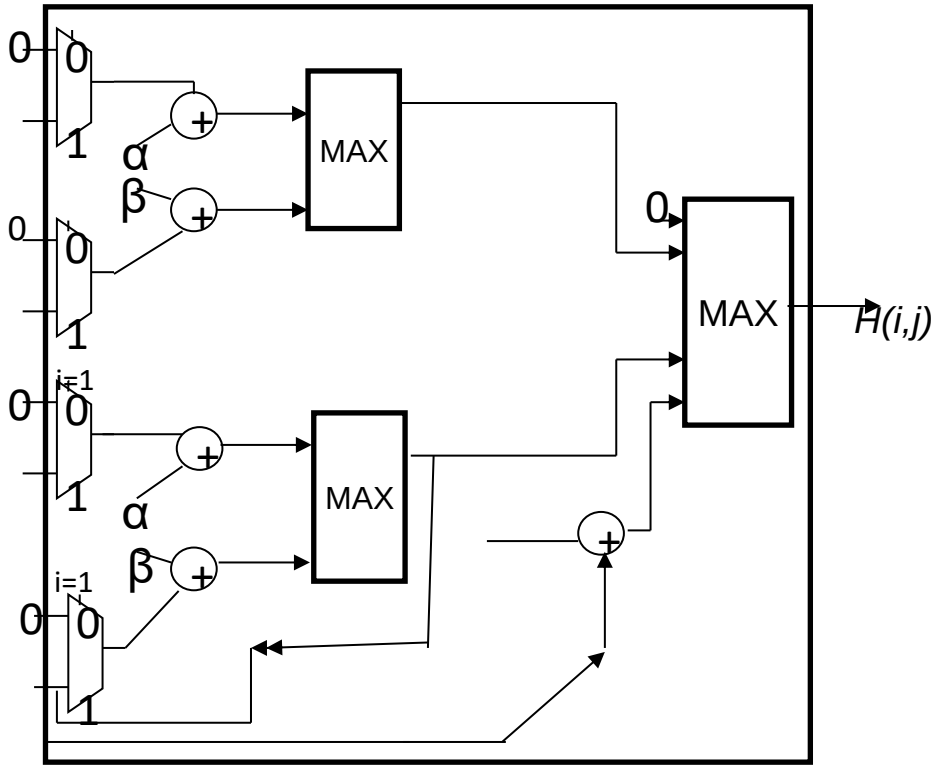
Hardware Description Language



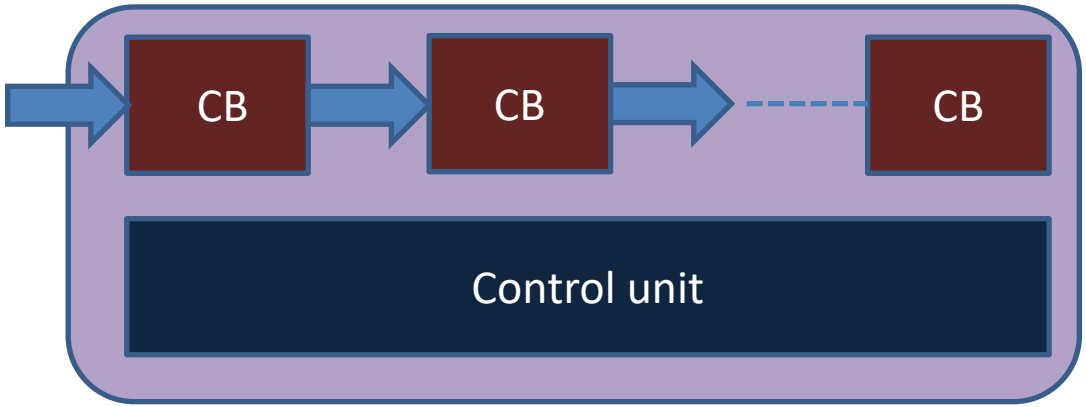
Example flow

$$H(i,j) = \max \begin{Bmatrix} 0 & \\ H(i-1,j-1) + s(a_i, b_j) & \text{Match/Mismatch} \\ \max_{k \geq 1}\{H(i-k,j) + W_k\} & \text{Deletion} \\ \max_{l \geq 1}\{H(i,j-l) + W_l\} & \text{Insertion} \end{Bmatrix},$$

Equations



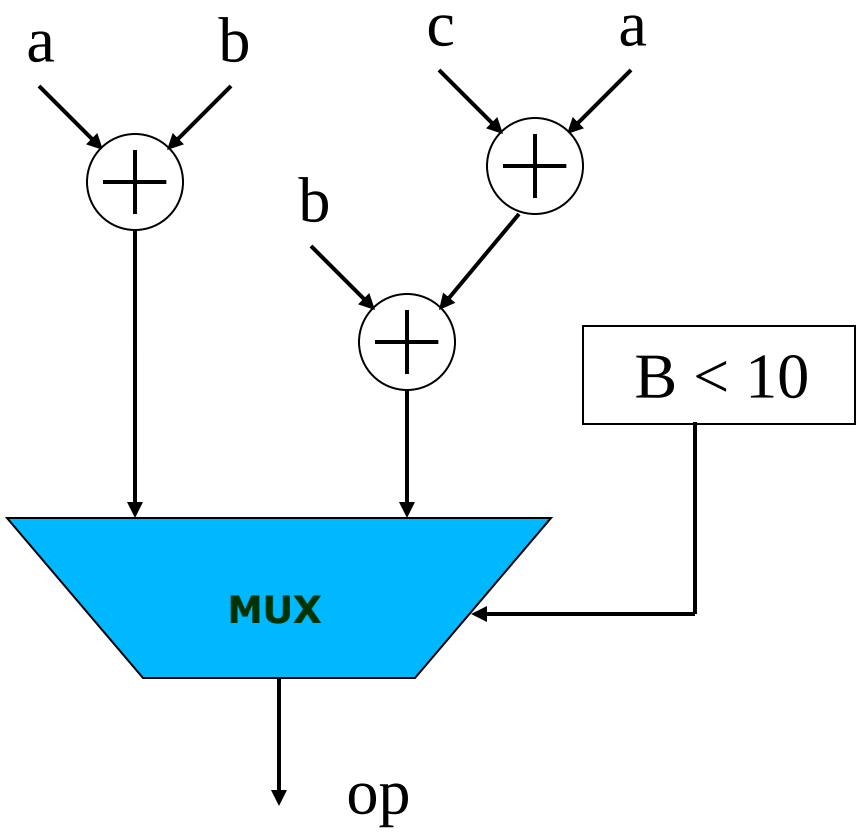
Compute Block (CB)



compute Block Array

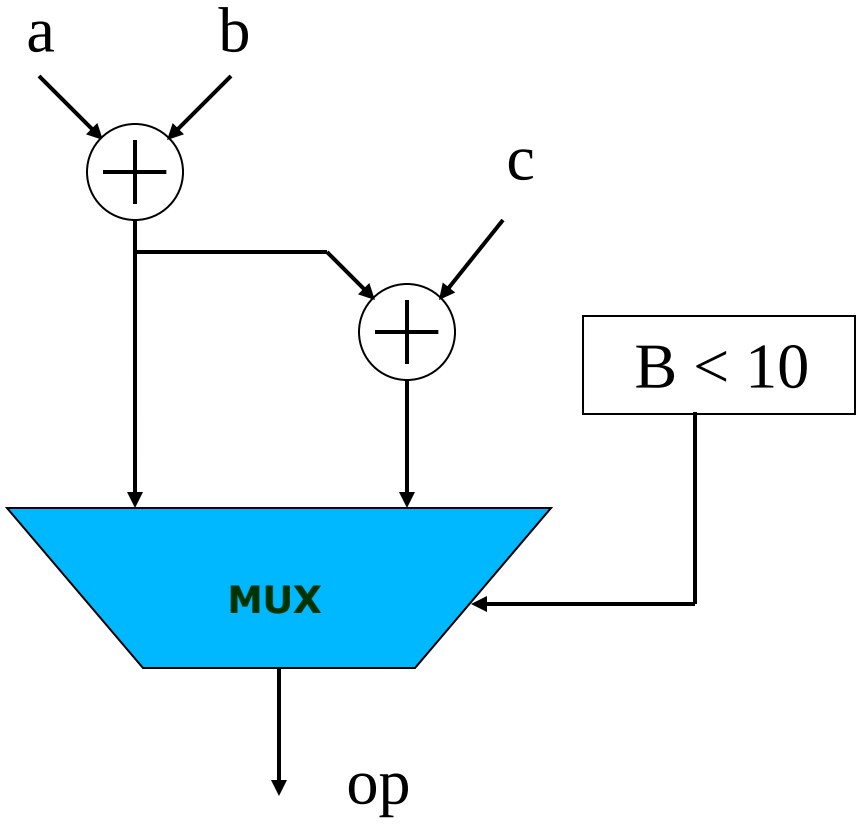
Coding Style Impact: Hardware Inference

```
entity test is
port (
   a, b, c : in std_logic_vector(5 downto 0);
         op : out std_logic_vector(5 downto 0)
         );
end entity test;

architecture test_a of test is
begin
       …
       if(b < “001010”)then
          op <= a + b;
       else
          op <= c + a + b;
       end if;
       …
end architecture test_a;
```

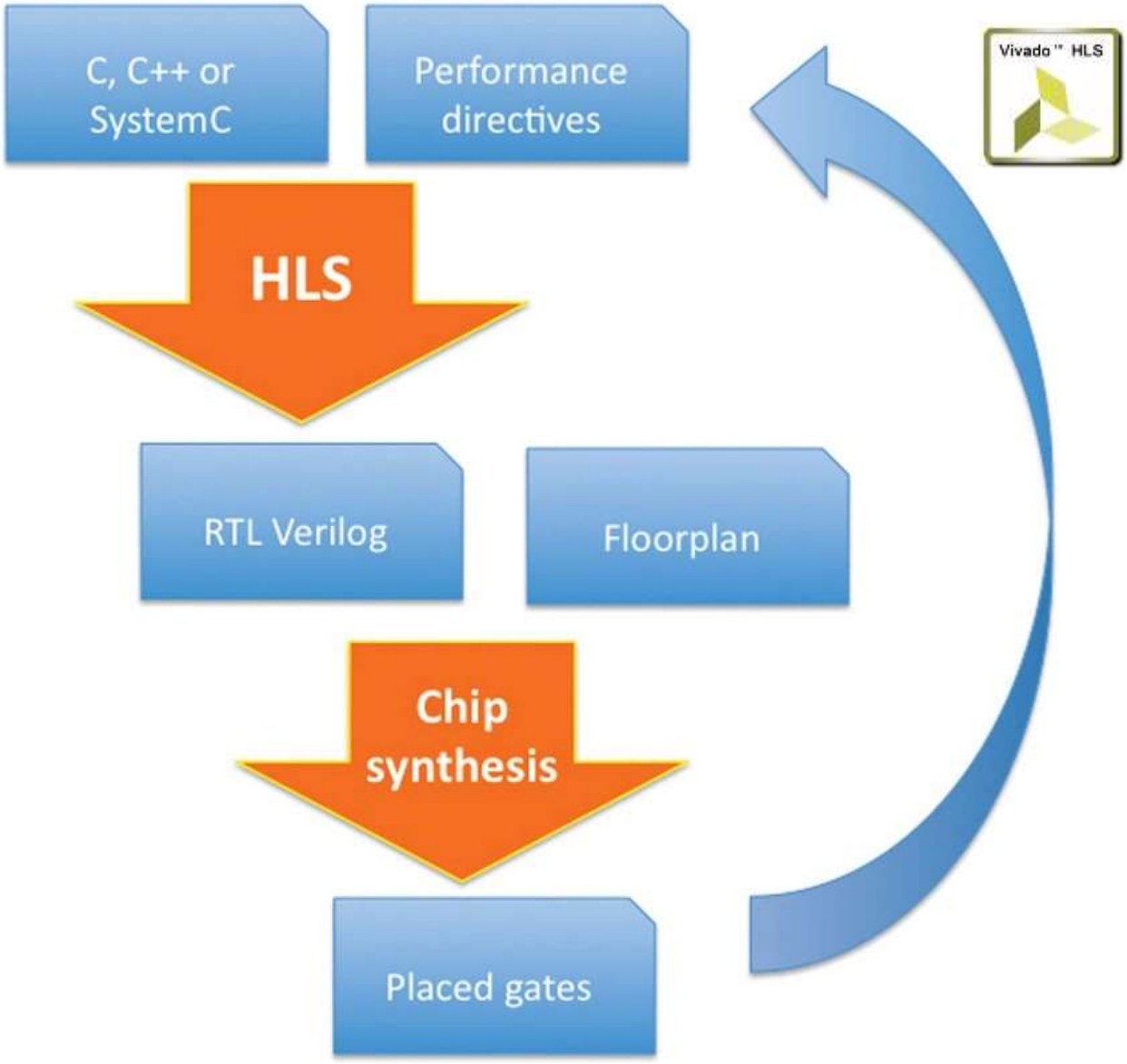


Coding Style Impact: Hardware Inference

```
entity test is
port (
   a, b, c : in std_logic_vector(5 downto 0);
         op : out std_logic_vector(5 downto 0)
         );
end entity test;

architecture test_a of test is
begin
       …
       if(b < “001010”)then
          op <= a + b;
       else
          op <= c + (a + b);
       end if;
       …
end architecture test_a;
```

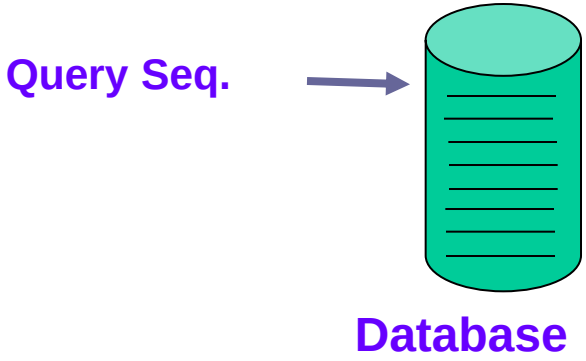


High Level Synthesis (HLS)



Application: Sequence Similarity Search

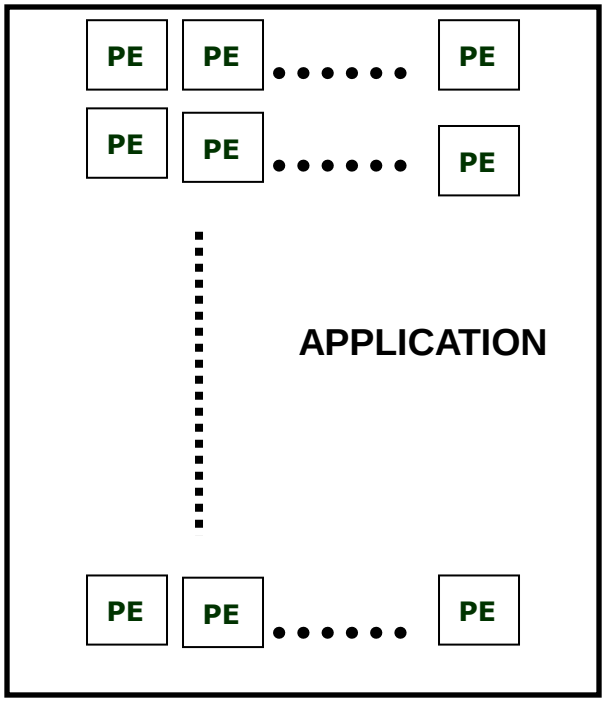
- **Comparing a query sequence against a database of known sequences -** **"a routine work for advancement in medical science".**



- **Searching queries over large databases takes hours to days and requires high-end servers or clusters.**

Sequence Search on FPGA

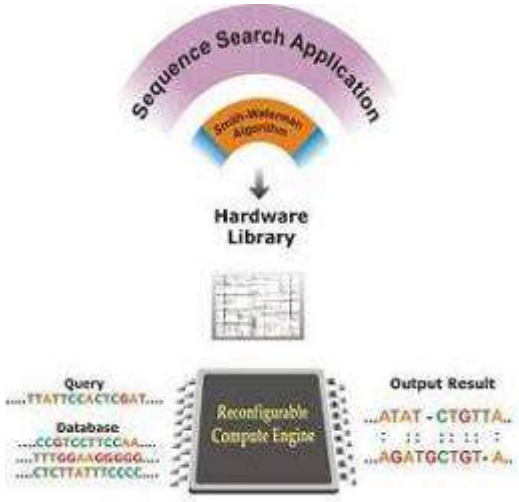
Multiple Parallel Execution units in FPGA for performance



Performance

Example: **RC for Accelerating Sequence search**

Bioinformatics Sequence search taking ***528 days*** using pure software solution was completed in just ***12 days*** by C-DAC's RC !



[what one of our customer has to say…]

"C-DAC's Reconfigurable Hardware Accelerator has helped us to achieve high scalability and reliability in our Microarray probe, Primer design business...

Thanks C-DAC for being a great partner in our endeavour to become world number one in genomic outsourcing."

…90x acceleration

C-DAC Reconfigurable Computing Accelerator in Clustered Environment

- RC further increasing the compute power of supercomputing Clusters
- Accelerating MPI based applications
- 88% saving of power

Highly space efficient 1 RC card just 25W

Advantages of FPGA Accelerator

Faster Application

Customized logic, allows to execute implemented operations in parallel

Energy Efficient

No fixed data widths/ boundaries as present in processors, rather the widths are customized as per the function, saving lot of power

Space Efficient

Compact in Size Cooling infrastructure not required due to low power consumption

PCIe based cards from Xilinx

Ack: Xilinx
ALVEO U280 Data Center accelerator card

Useful for Data Analytics, High Frequency Trading, ML Inference applications

Full height, dual slot, ¾ length (passive cooling) or full length (active cooling) form factor.

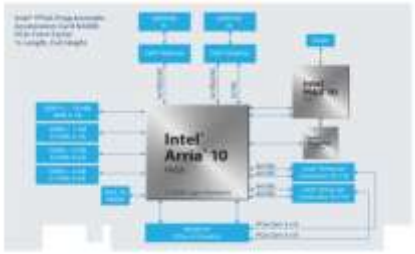
PCI Express® Gen3 x16 or Gen4 x8, 8 GB High-Bandwidth Memory (HBM2), 16 GB DDR4 @ 2400 MT/s,

Two QSFP Ethernet ports capable of 100 Gb/s each. 225W

Accelerate memory-bound, compute-intensive applications

Intel FPGA Accelerators

Intel FPGA PAC D5005
(Latest)



Intel PAC Arria 10

Intel FPGA PAC N3000
(Networking appl accleration)

Big data analytics, artificial intelligence, genomics, video transcoding, cybersecurity, and financial trading.

Ack: Intel

Intel SoC FPGAs

F-Series

- Increased DSP capabilities
- Quad-core integration option
- Upto 58 Gbps transceiver

I-Series

- Optimized for High performance Processor Interface
- PCIeGen5; 112 Gbps transceiver

M-Series

- Optimized for Compute & Memory appls
- DDR5 controller..

Ack: Intel

- **Useful for Data Center, Networking, Edge**
- **Applications with massive interface bandwidth & High performance**
- **Data-intensive applications- Massive memory + high bandwidth**

Summary

- Accelerators are an important ingredient of HPC, providing performance, energy efficiency and small rack-space.
- KNL designed to take care of many drawbacks of earlier Xeon Phi (KNC)

  Discontinued by Intel June'18
- FPGA based reconfigurable accelerators are most energy efficient compared to other accelerator technologies.

Thank You

"….paving the path for building such systems to tackle additional, unsolved important scientific problems"