Computer Architectures and Algorithms: From Filtering to Deep Learning

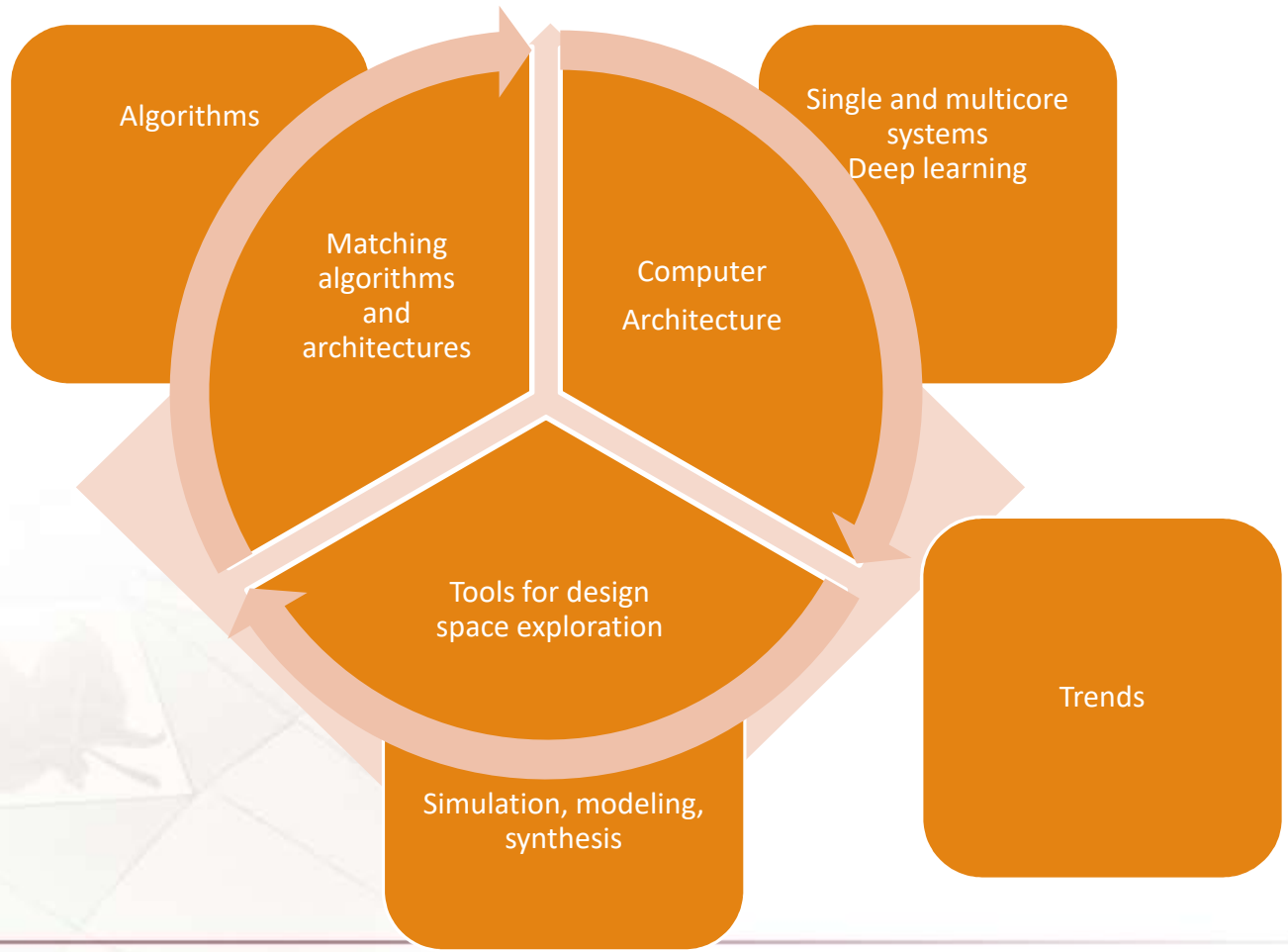
Miodrag Bolic

Professor
School of Electrical Engineering and Computer Science
University of Ottawa
mbolic@uottawa.ca

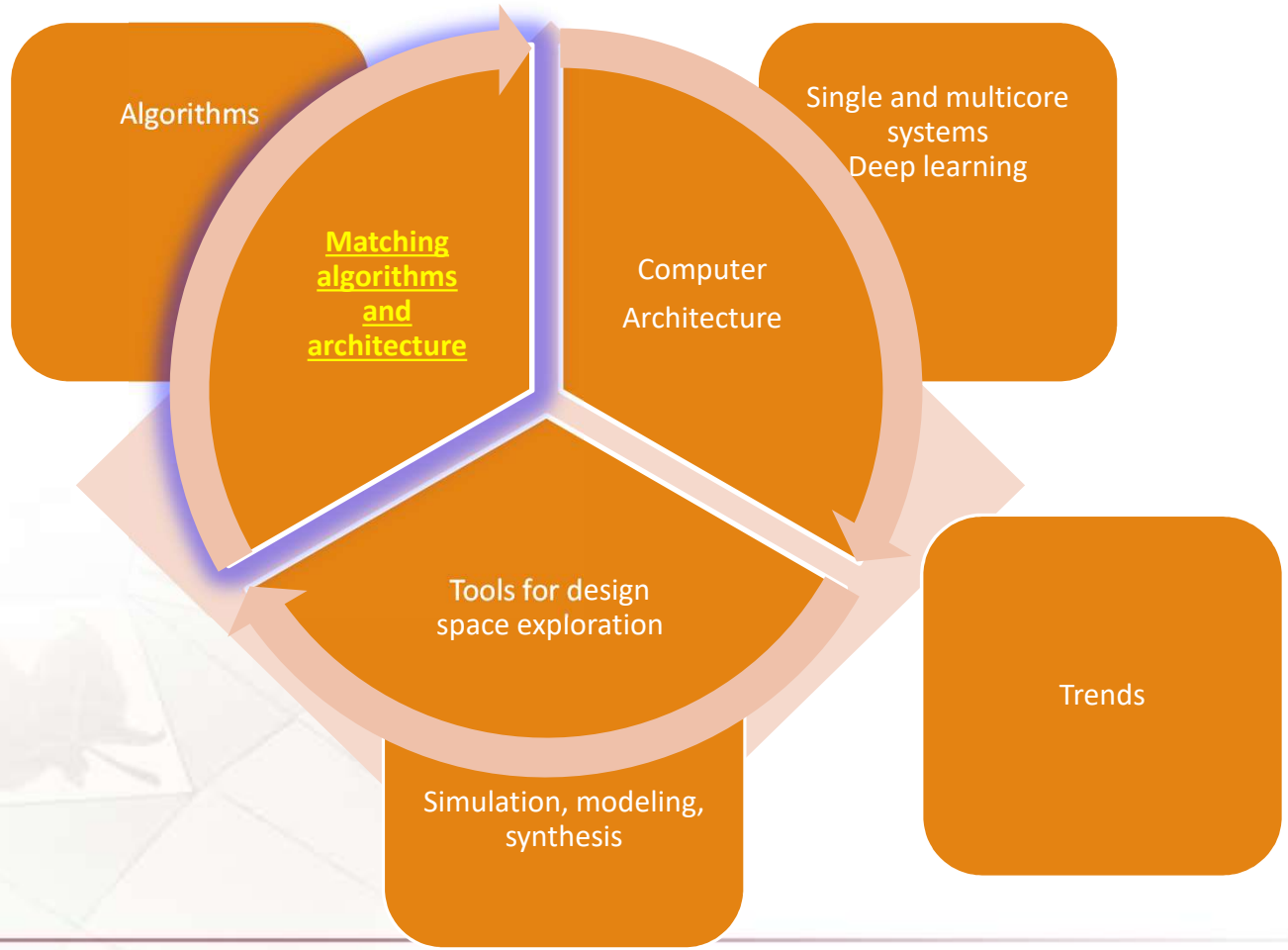
My teaching and research

- Teaching
  - Current courses
    - Computer system design
    - Digital system design: VHDL
    - Parallel computer architecture: Multicore, GPUs
    - Biomedical instrumentation
  - New courses in 2019/2020
    - Data analysis and machine learning with uncertainty
    - Decision making and AI under uncertainty
- Research
  - Hardware/software accelerators
    - Custom instruction identification, coprocessor hardware generation, Java-accelerators, acceleration of circuit analysis software on GPUs
  - Biomedical instrumentation, DSP and machine learning

Outline

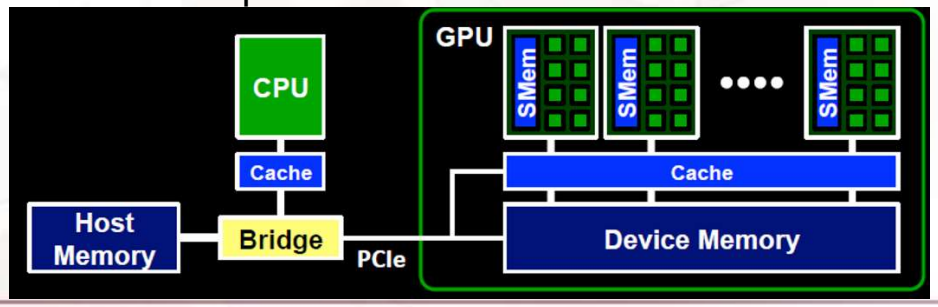

Algorithms
Matching algorithms and architectures
Computer Architecture
Single and multicore systems
Deep learning
Tools for design space exploration
Simulation, modeling, synthesis
Trends

Outline


Algorithms
Matching algorithms and architecture
Computer Architecture
Single and multicore systems
Deep learning
Tools for design space exploration
Simulation, modeling, synthesis
Trends

Design approaches for DSP algorithm implementation

1. **Matching hardware to algorithm**
   - Hardware architecture must match the characteristics of the algorithm.
   - Example
     - CPU excels at executing few sequential threads
     - GPU excels at executing many parallel threads

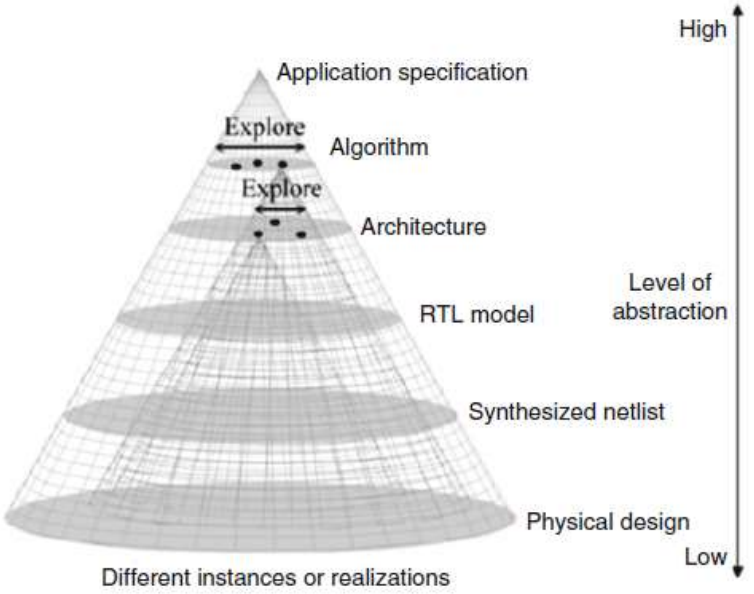


Hardware Software Co-design

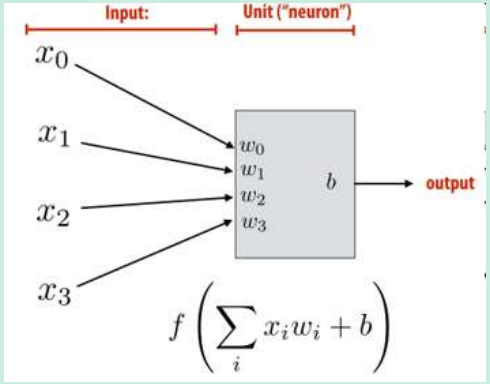
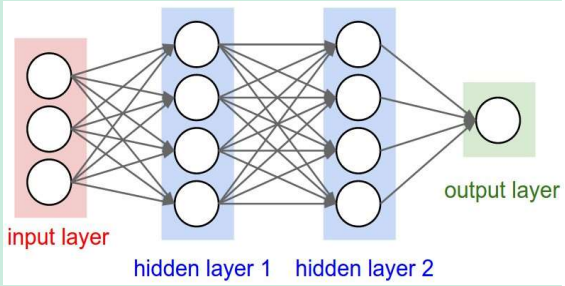
2. **Formulate algorithm to match hardware**
   - Algorithm must be formulated so that they can best exploit the potential of architecture.
   - Example:
     - PDSP architectures is fixed.
     - One must formulate the algorithm properly to achieve best performance.
     - Eg. To minimize number of operations.

Reducing power and improving performance at all design levels

- Algorithmic level
  - Reducing # of operations
  - Unnecessary reads and writes
- Compiler level
- Architecture level
  - Parallel
  - Heterogeneous
  - Local memories
  - Clustered FU
  - Reduced bit-width
- Circuit level
- Silicon level



Representative algorithms: Multiply-ACcumulate (MAC)

Algorithm	Operation	Result
Filter	$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{r=0}^{M} b_r x[n-r]$	Denoised signal
Image processing convolution	Kernel: $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ * Image	Sharpened image
Neural network		Classification, Prediction 

Addressing increased processing requirements

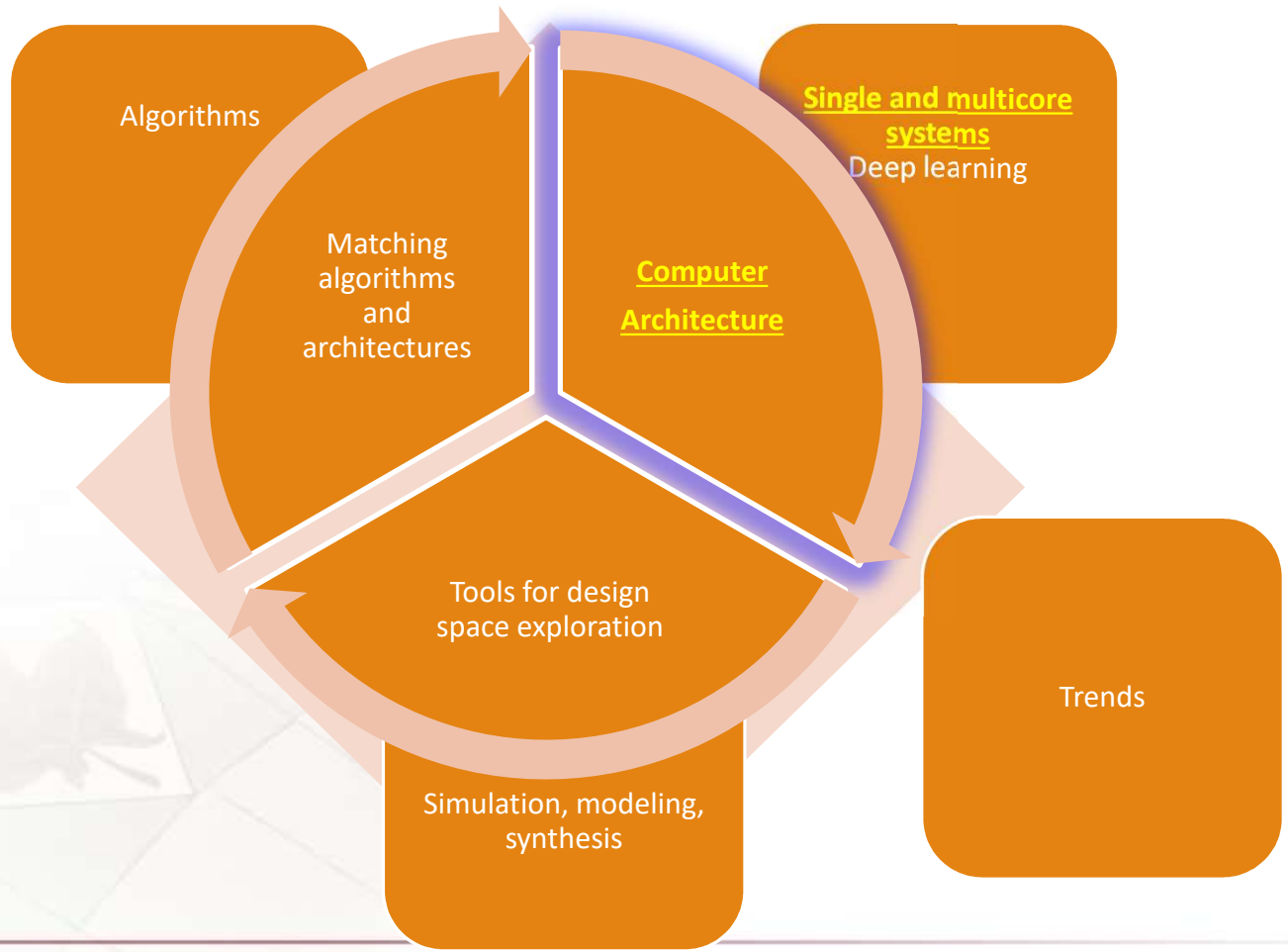
- Algorithms
  - Filters
  - FFT
  - Image processing
  - Neural networks

  Complexity increases

- This is addressed by
  - Parallelism
  - Customization
  - Heterogeneity

	Operations per cycle
CPU	a few
CPU (vector extension)	tens
GPU	tens of thousands
TPU	hundreds of thousands, up to 128K

Outline

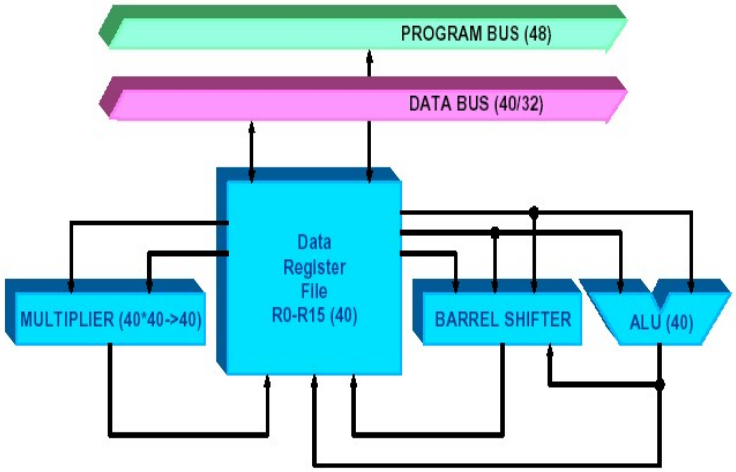

Algorithms
Matching algorithms and architectures
Computer Architecture
Single and multicore systems
Deep learning
Tools for design space exploration
Simulation, modeling, synthesis
Trends

Single processors

Architecture	Addressing the issue
Programmable DSPs	Support for MAC
SIMD and multimedia	Multiple identical operations Image processing
Vector architectures	Block/array processing
VLIW	Multiple functional units, Instruction level parallelism, Multiple MACs
Multithreading	Not enough instruction level parallelism –> thread level parallelism
Processor with customizable instruction sets	Support for heterogeneous acceleration inside the processor

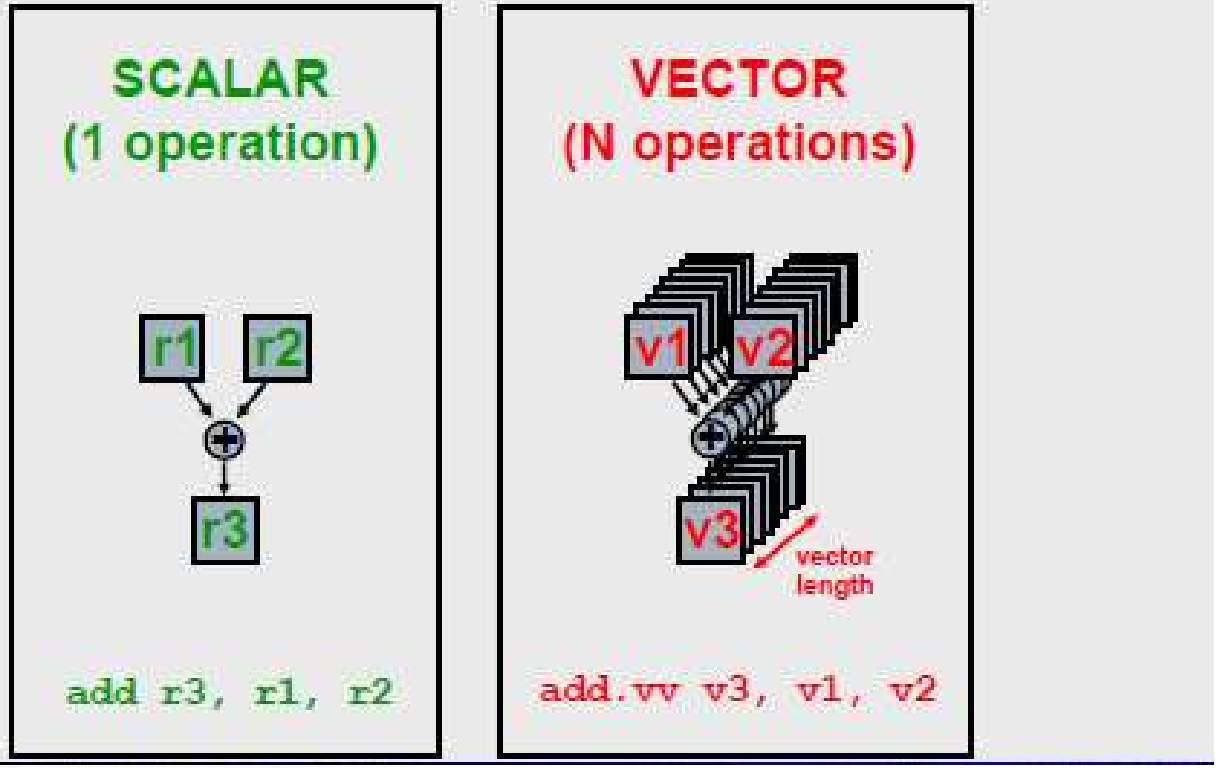
Programmable Digital Signal Processors (PDSPs)

- Micro-processors designed for signal processing applications.
- Special hardware support for:
  - Multiply-and-Accumulate (MAC) ops
  - Saturation arithmetic ops
  - Zero-overhead loop ops
  - Dedicated data I/O ports
  - Complex address calculation and memory access



Vector processing

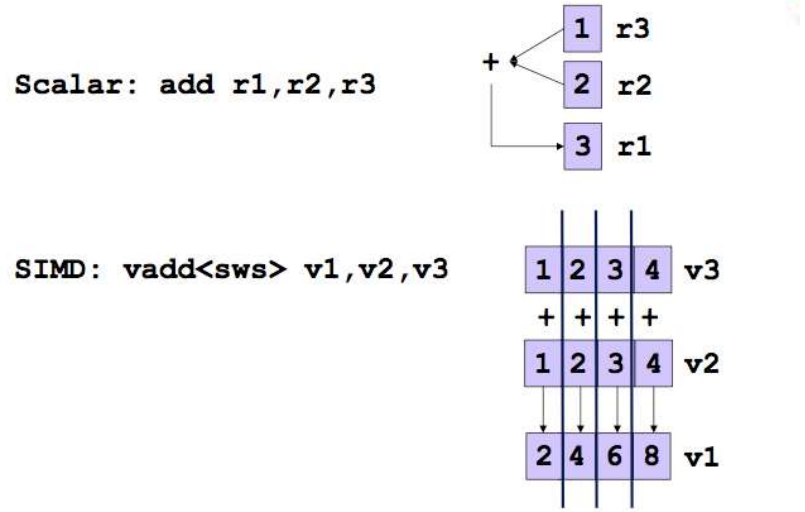
- Vector processors have high-level operations that work on linear arrays of numbers: "vectors"



SIMD and multimedia

- Benefit:
  - Multiple ALU ops → One SIMD op
  - Multiple load/stores→ One wide mem op
  - Image processing

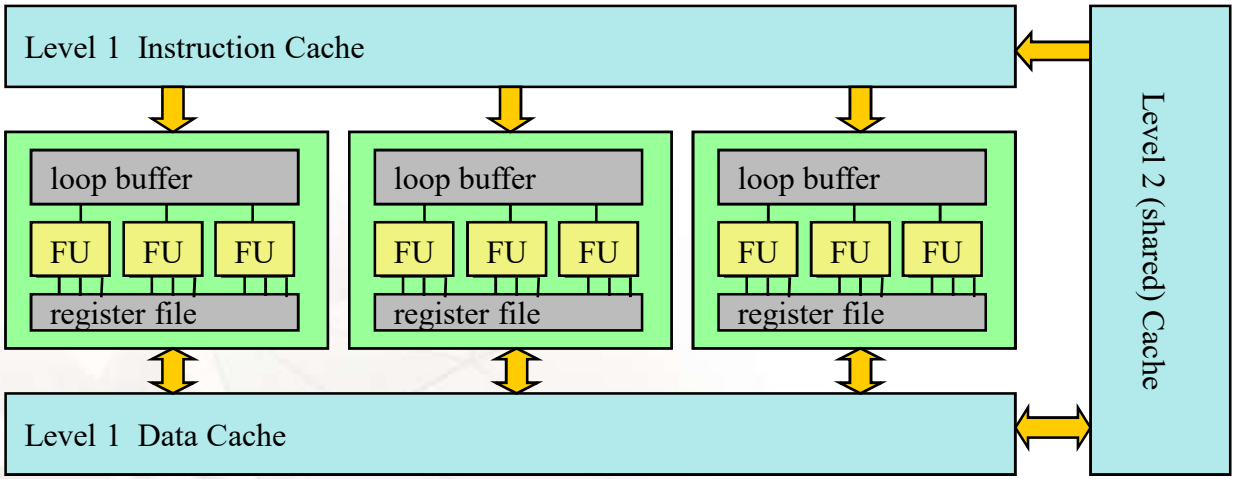
- What are the overheads:
  - Alignment overhead
  - Control flow
    - Control flow may require executing all paths




```
R       R       XR       1.08327
G   =   G   +   XG   *   1.89234
B       B       XB       1.29835
```

Exploring instruction level and loop parallelism

- Very long instruction word VLIW
  - clustered VLIW architecture with multiple register files and many functional units (e.g. TI C64 series)



- Superscalar

Processors with customizable instruction sets

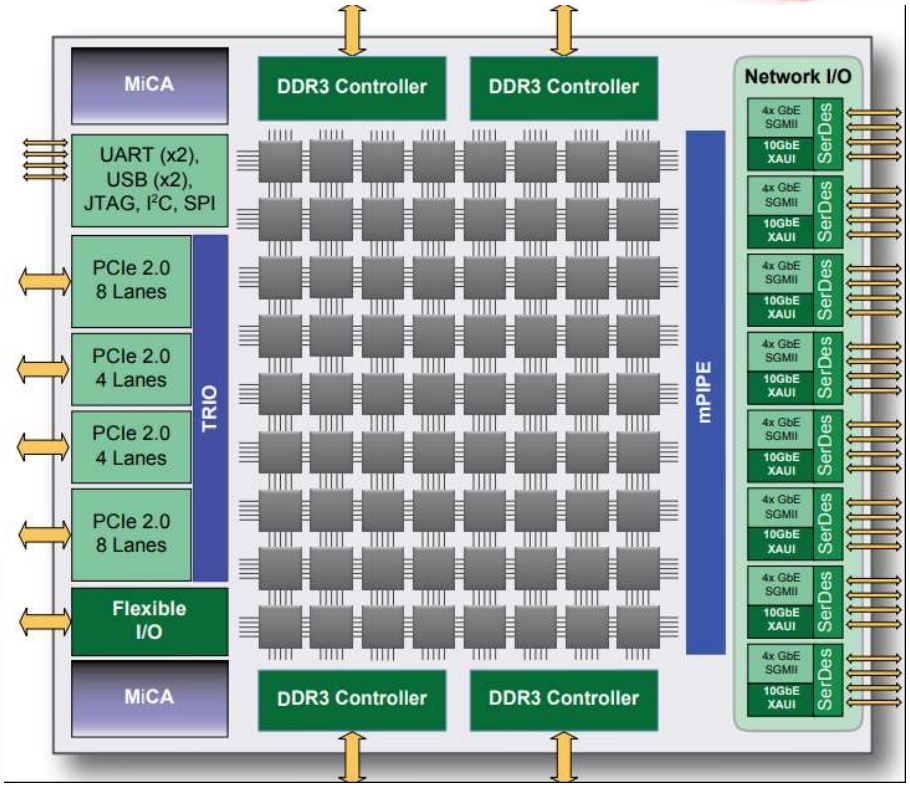
- Tensillica, NIOS II from Altera/Intel
- RISC-V
  - RISC-V is a project that began in 2010 at the University of California, Berkeley.
  - Is an open-source instruction set architecture (ISA) based on established reduced instruction set computing (RISC) principles.
  - Features
    - Small core instructions sets
    - Space for new opcodes
    - Single, double and quadruple floating point
    - Atomic memory operations
  - A complete software stack for both embedded and general-purpose computing
  - Extensions: multiplication and division, atomic read and write, floating point, vector, bit-level, compressed instructions, privileged.

Going parallel

Architecture	Features
Homogeneous	Parallelism
• Multicore	Up to 16 cores complex cores
• Manycore	> 16 cores simpler cores
• GPU	Hundreds or thousands of cores
Heterogeneous	Specialization
• Application specific accelerators	DSP+RISC+ASIC+FPGA
• Multicore+ FPGA	Cloud applications

Manycore

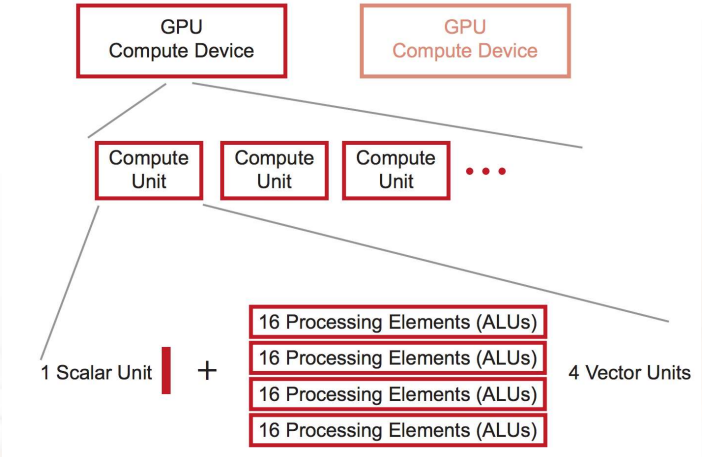
- Epiphany V, Tilera (now Mellanox), Kalray, and KnuEdge
  - map computations across a large number of homogeneous cores,
  - the dynamic distribution of memory and communication between the processors eventually limits the scalability.

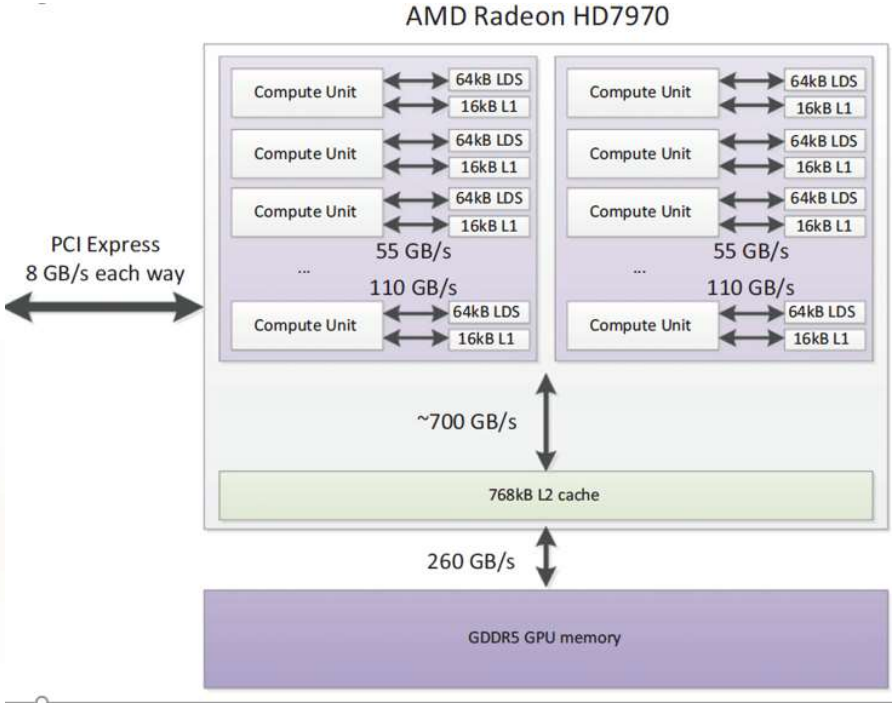


TILE-Gx8072 Processor Block Diagram

Graphical processing unites: GPU

- Programmed in Cuda or OpenCl
- Multiple GPU programmed in OpenAcc
- From embedded systems to cloud



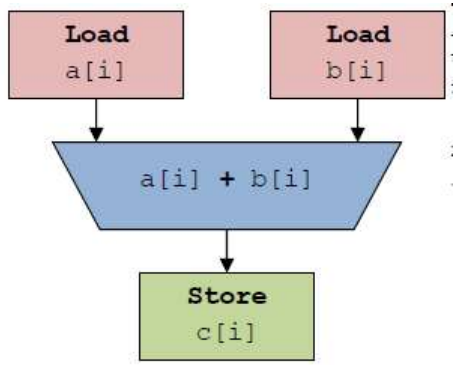


FPGA accelerators

- Field programmable gate array
  - Custom hardware accelerator
  - Low quantity system deployment
- High Level Synthesis (HLS) frameworks, like Intel FPGA SDK for OpenCL and Xilinx SDAccel
- On an FPGA, each OpenCL kernel is compiled to a custom circuit.
  - Kernel vectorization
  - Kernel replication

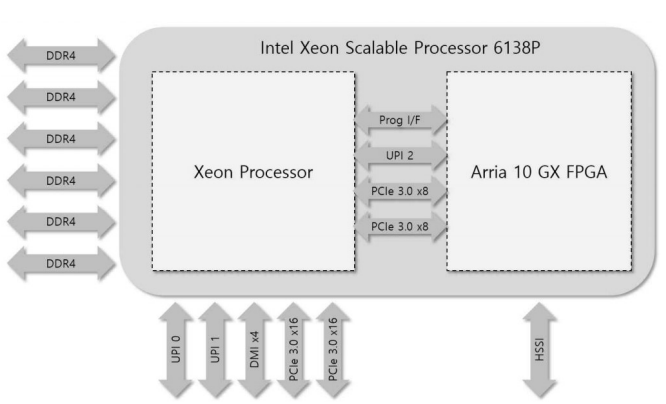
```
kernel void Add(global float *a,
                global float *b,
                global float *c)
{
  /* Collect unique work-item ID */
  int i = get_global_id(0);

  c[i] = a[i] + b[i];
}
```

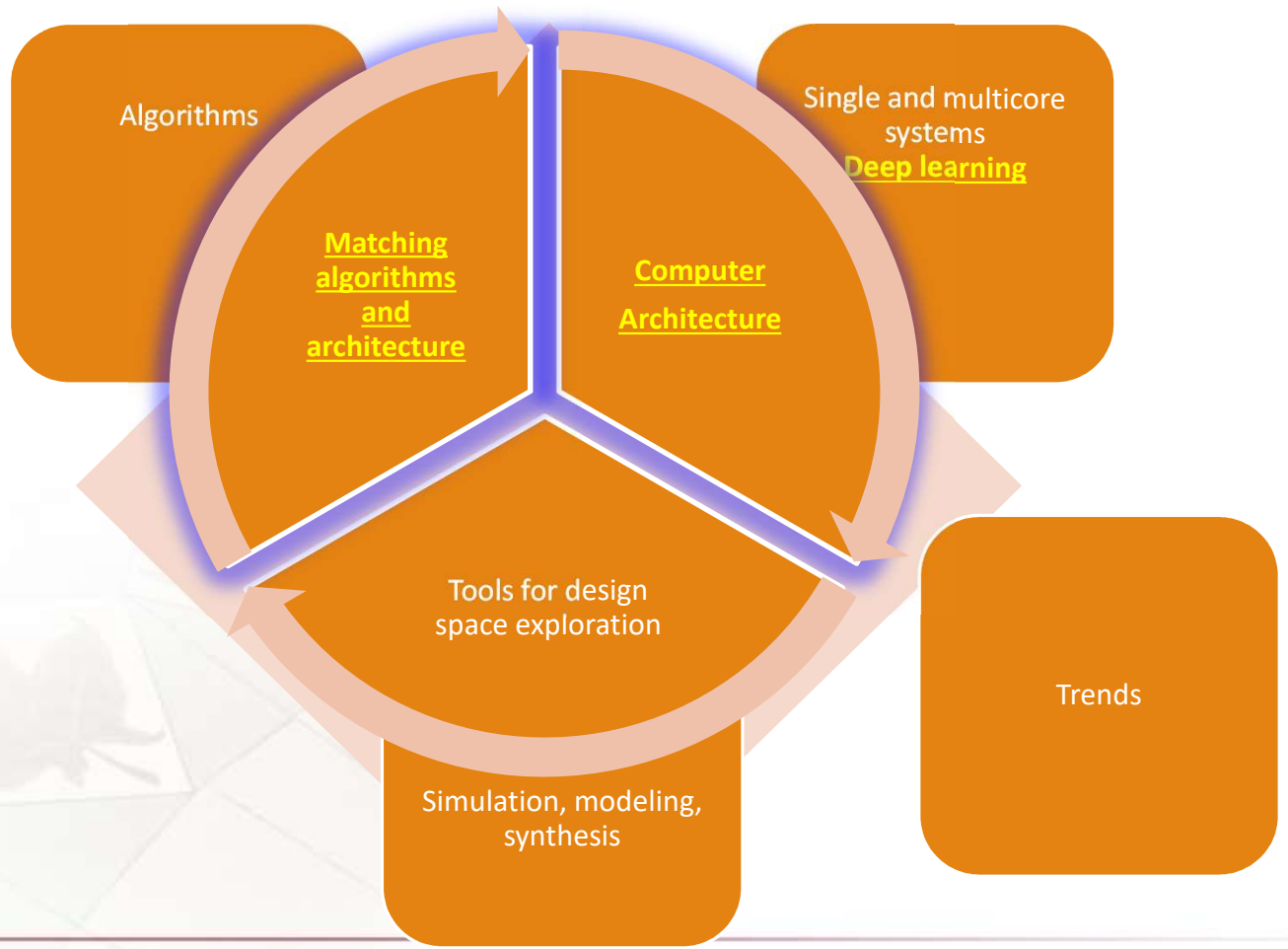


Heterogeneous CPU-FPGA Platforms in a chip

- Data-center requirements: high performance + power efficiency
  - provide application specific hardware accelerators
  - data center workloads are quite diverse and tend to change
- Main application in data-centers (Amazon, IBM, Microsoft – PCIe connection)
- Coherent, shared memory: the Intel Xeon+FPGA
  - Estimated price $6,500 for 20 core Xeon processor and Arria (1.3 TFLOPs)
  - accelerator function units (AFUs)
  - FPGA interface unit (FIU)
  - last-level cache (LLC)

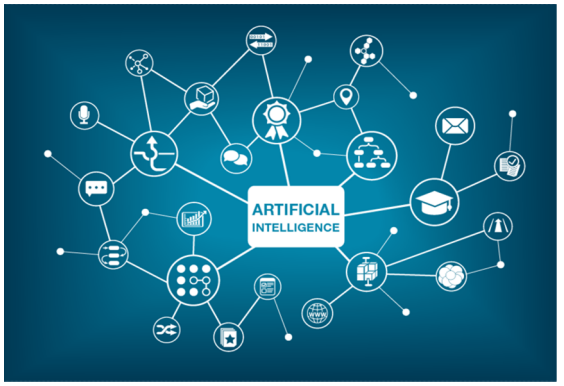


Outline


Algorithms
Single and multicore systems
Deep learning
Matching algorithms and architecture
Computer Architecture
Tools for design space exploration
Simulation, modeling, synthesis
Trends

From signal processing to deep learning

- Machine learning and AI are everywhere
  - Medical Diagnoses
  - Driving Cars
  - Robotics
  - Wireless
- Traditional machine learning
  - Feature engineering
  - Neural networks are shallow
- Deep learning
  - Automatic feature extraction
  - Very wide and deep architectures



Drivers and challenges

- Drivers
  - Large amounts of data
  - Capable hardware for training, cloud and edge
  - Algorithm advancements
  - Open source tools
  - Sharing code and data by researchers
- Challenges
  - Size of the network and speed
  - Energy efficiency
  - Mainly supervised
  - Hype

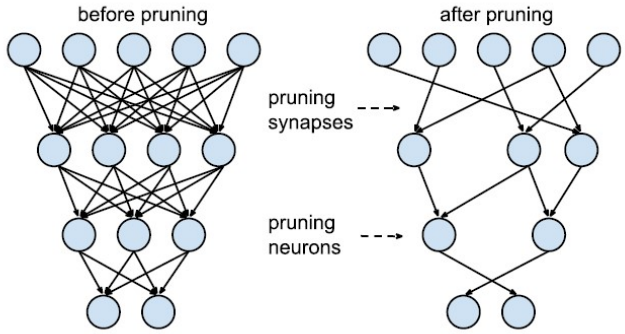
How complex are deep learning networks

- Multiple convolution layers + fully connected network

Year	Model	Layers	Parameter	FLOPs	ImageNet Top-5 error
2012	AlexNet	5+3	60M	725M	16.4%
2013	Clarifai	5+3	60M	—	11.7%
2014	MSRA	5+3	200M	—	8.06%
2014	VGG-19	16+3	143M	19.6B	7.32%
2014	GoogLeNet	22	6.8M	1.566B	6.67%

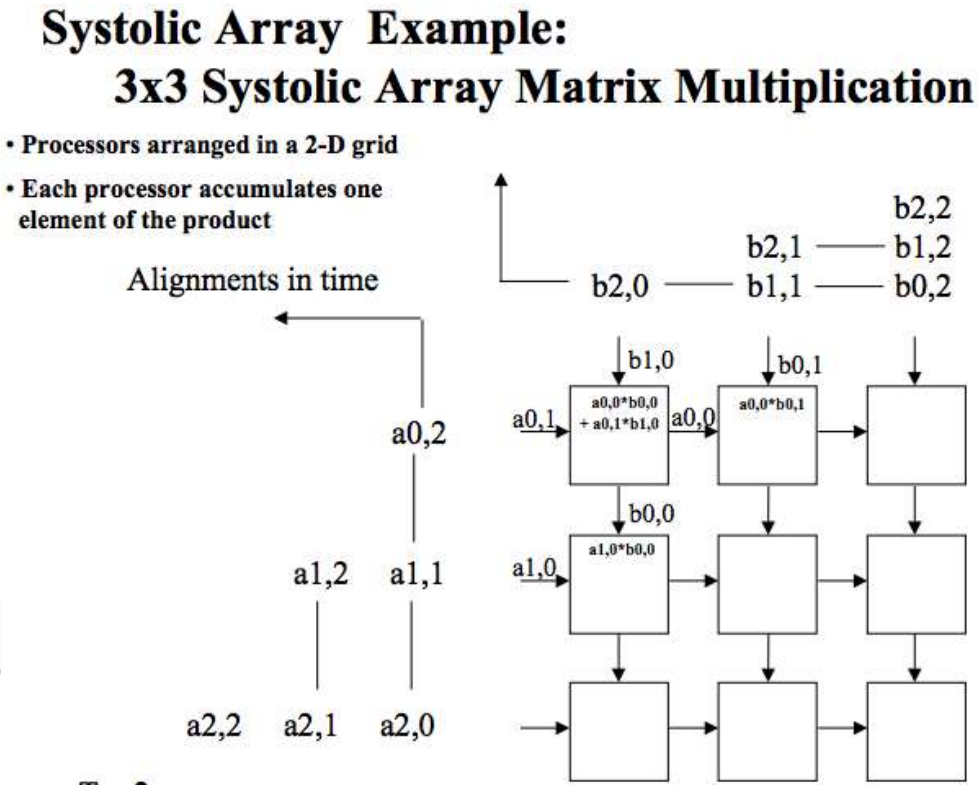
Formulate algorithm to match hardware

- Pruning
- Weight sharing and quantization
  - AlexNet
    - Size 240MB->6.9MB 35x
    - Accuracy 80.27% -> 80.30%
- Reducing complexity of matrix multiplication
  - FFT
  - Other efficient algorithms
- Local communication – reducing number of DRAM accesses



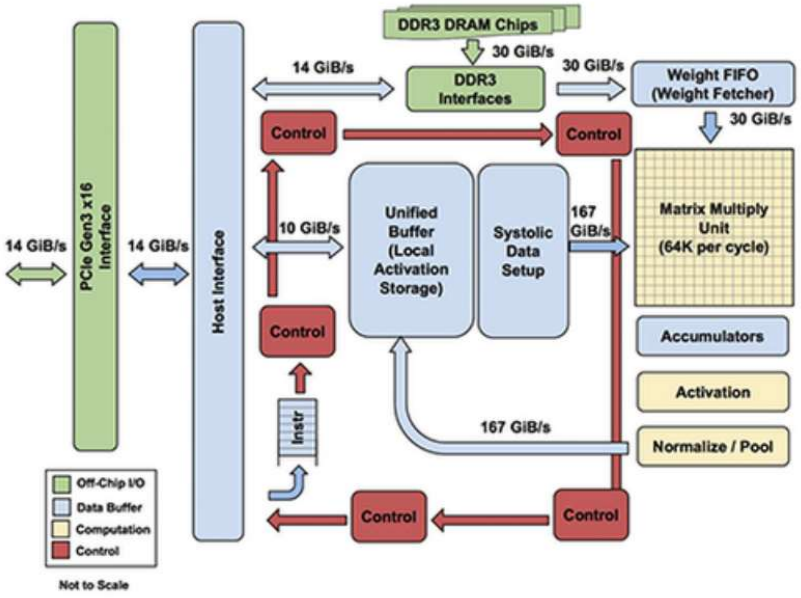
Matching hardware to algorithms

- Parallel processing mainly for matrix multiplication
  - Parallel MACs
  - Systolic arrays
    - Reuse previous results to reduce communication, storing in the register files
    - Increasing on-chip memory
    - Allow for static scheduling
- Support for integer, mixed precision or floating point operations with smaller number of bits



Google Tensor Processing Unit

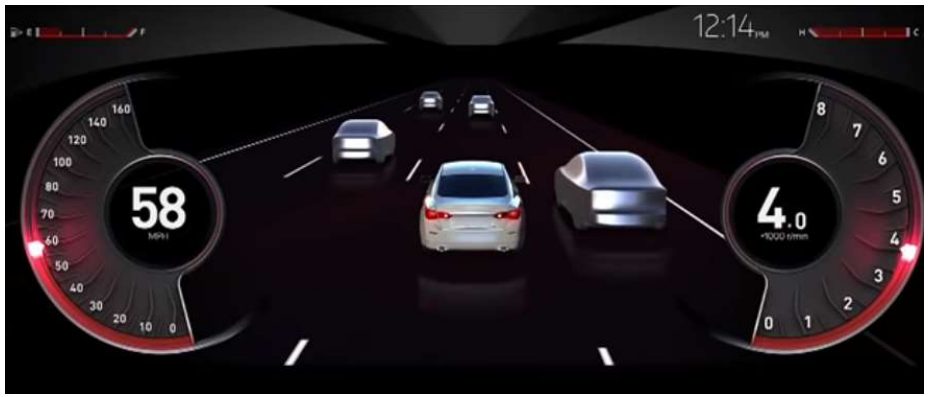
- Run machine learning workloads
  - CISC style
  - Vector Processing
- Matrix Multiplier Unit
  - 65536 8bits multiply and add unit for matrix operation
- Unified Buffer:
  - 24MB of SRAM work as register
- Activation Unit:
  - Hardwired activation functions



Source Google

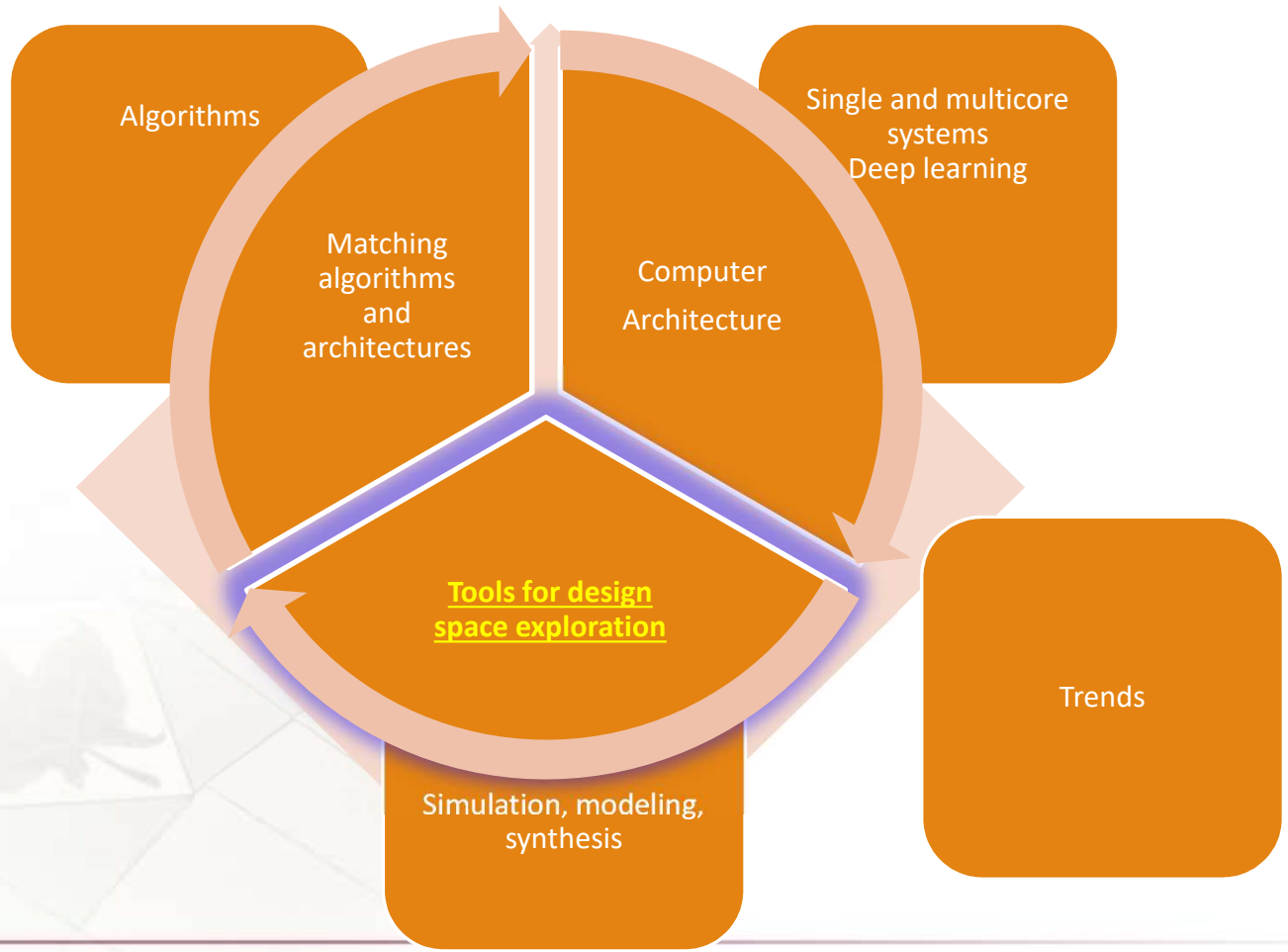
Self-Driving Car, Nvidia Drive PX

- Sensing, localization, planning and action taking
- Multiple radars, lidars, cameras
- Multiple neural networks forming occupancy grid



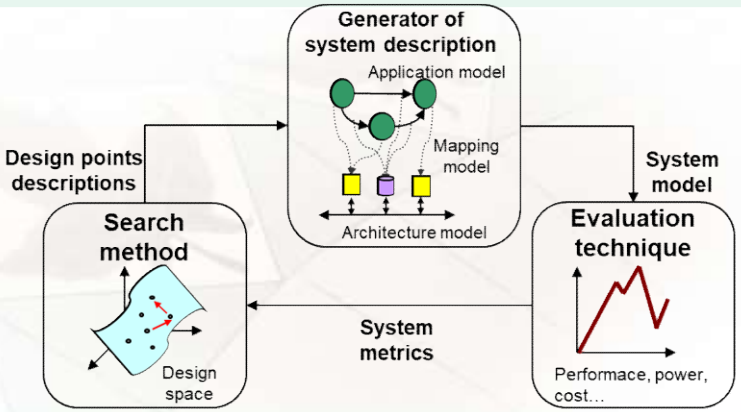
- Xavier SoC
  - 30 trillion operations per second
  - 8-core CPU,
  - 512-core Volta GPU,
  - deep learning accelerator,
  - computer vision accelerators and new 8K HDR video processors
- Pegasus
  - two Xavier Systems-on-a-Chip for developing Society of Automotive Engineering (SAE) Level 2/3 autonomous driving applications
  - $15,000

Outline


Algorithms
Matching algorithms and architectures
Single and multicore systems
Deep learning
Computer Architecture
Tools for design space exploration
Trends
Simulation, modeling, synthesis

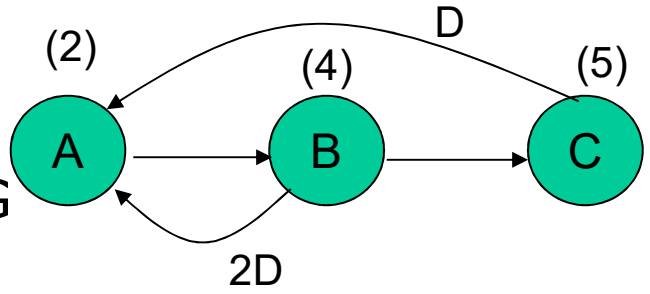
Tools

Tool type	Example
Complete simulation and implementation toolchain	Simulink
High-level synthesis of algorithms	FPGA tools, SystemC
Design space exploration	Cycle accurate: Multi2Sim, System level: Mirabilis VisualSim
Automated parallelization	Silexica SLX



DSP algorithm optimizations

- Algorithm Representations as dataflow graphs (DFG)
- Parallelism and Pipelining
- Retiming
  - transforms a dataflow graph by redistributing delays in the DFG



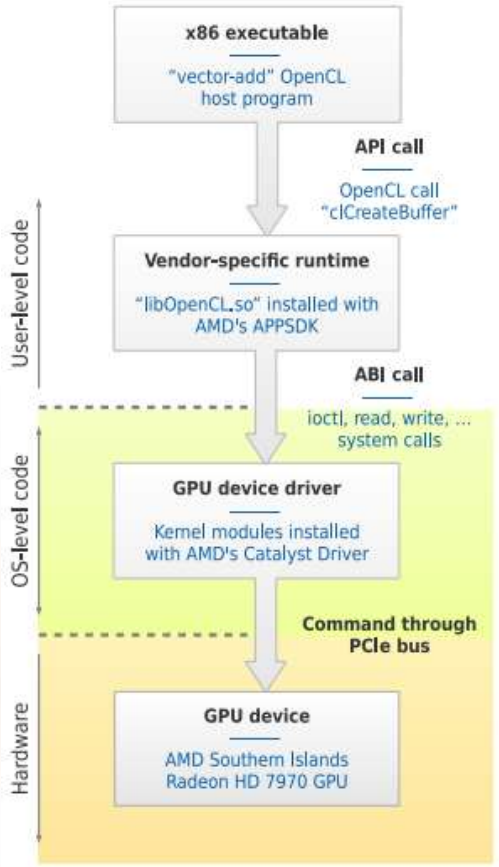
- Unfolding
  - replicates a loop body several times to increase the instruction level parallelism in the unfolded loop
- Folding
- Bitwidth Analysis and Optimization
- Memory Space Allocation
- Loop optimization and software pipelining

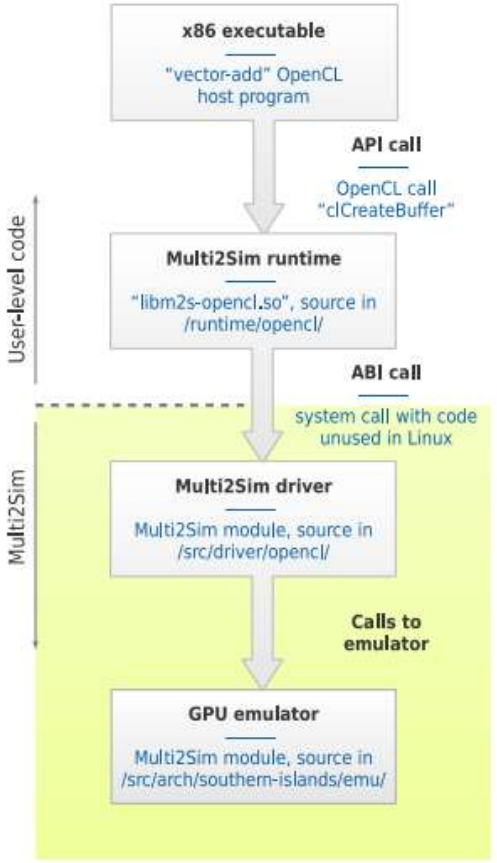
FPGA High Level Synthesis tools

- Domain specific languages
  - Bluespec System Verilog BSV
    - Written in C++
    - Open-source
    - Synthesizable Verilog RTL or SystemC executable output
    - Atomic transactions for simple expression of concurrency
    - Parameterization: features, dimensions, types, functions, behaviors and micro-architectures
    - High-level specification of parallel, nested sequential FSMs
    - Significant developments regarding Risc-V: RISC-V Verification Factory, RISC-V Piccolo Core, RISC-V Acceleration Factory
- Generic C/C++
  - Cintesizer SystemC
- Simulators
  - Simulink

Computer architecture explorations: Multi2Sim

- Open source
- Support for CPU architectures: x86, ARM, and MIPS
- Support for GPU architectures: AMD Southern Islands, NVIDIA Kepler
- OpenCl, Cuda, C/C++





uOttawa

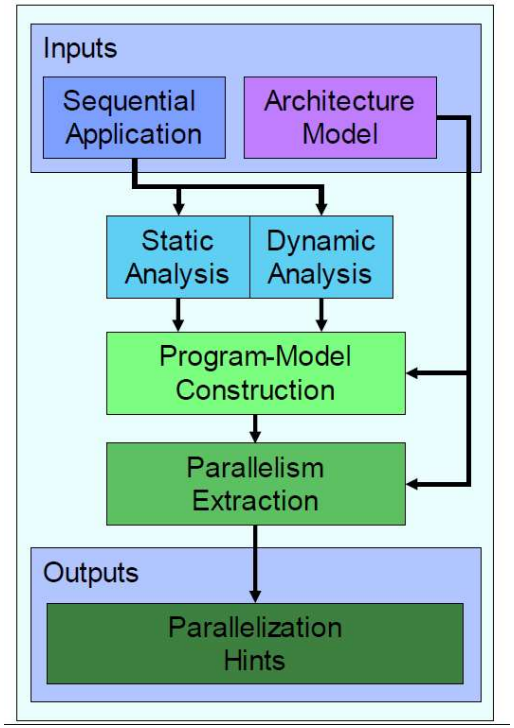
System architecture exploration: Mirabilis VisualSim

- Simulation for system architecture exploration
- Library for stochastic, cycle-accurate and custom modeling
- Modeling different processors, FPGAs and GPUs:
  - ARM, RISC-V, Nvidia-Drive-PX, Configurable GPU, DSP, mP and mC, PowerPC, X86- Intel and AMD, DSP- TI and ADI, MIPS, Tensilica, TPU
- Types of modeling
  - Flow control and behavior modeling
  - Signal Algorithmic Modeling
  - Control and Mixed Signal Modeling
  - Architecture Modeling
- Can be used for performance trade-offs using metrics
  - Hardware: Component selection, sizing and topology evaluation
  - Software: scheduling, resource allocation
  - Network: bandwidth utilization, application response time, buffer requirements
  - CPUs: pipeline, memory hierarchy

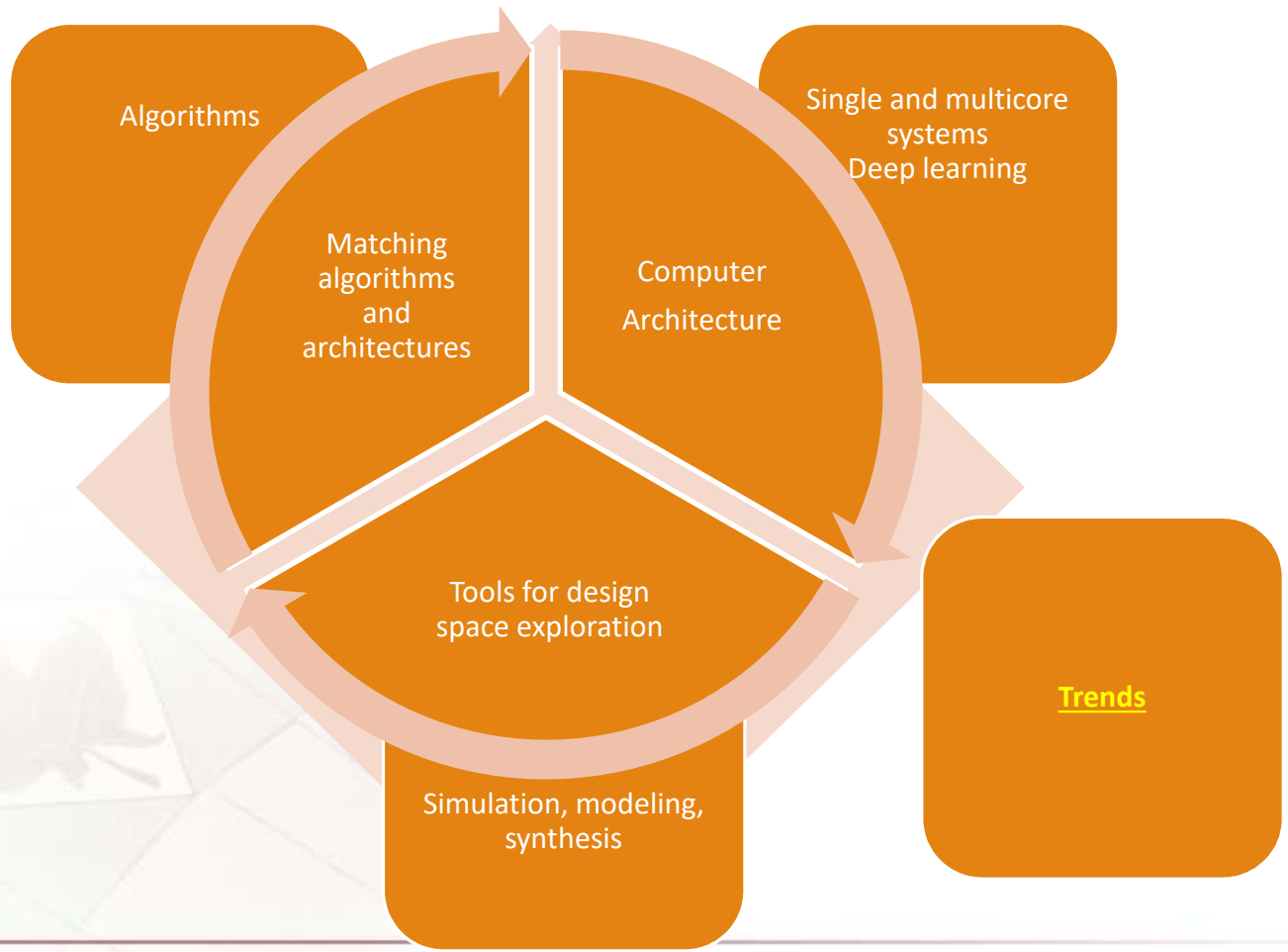
Tools for parallelization

- Programming models:
  - Parallel: OpenMP, OpenCl, MPI
  - Dataflow: Kahn, Synchronous
- Platform description:
  - Processors and accelerators
  - Memory subsystem and network
  - Timing: annotation, table, measurement, analytical or estimation
- Software parallelization
  - Intermediate representation
  - Statement, basic block, function
  - Flow and dependence analysis
- Software distribution
  - Accelerator offloading
  - Mapping code blocks to processing elements (static or dynamic)
  - Scheduling code blocks

- Academic tools: Shapes
- Industrial tools: Silexica SLX



Outline


Algorithms
Matching algorithms and architectures
Computer Architecture
Single and multicore systems
Deep learning
Tools for design space exploration
Simulation, modeling, synthesis
Trends

Trends in algorithm implementation

- Advances in computer architectures are accelerating emerging applications: personalized health care, analytics, 5G, smart cars
- DSP
  - Algorithm selection
    - Automatically select the best algorithm from the library of algorithms
  - Stochastic/probabilistic approaches – Monte Carlo based
- Deep learning
  - Designing tools while having in mind efficient hardware implementation
  - Going beyond supervised learning: reinforcement learning
  - Dynamic network architectures

Trends in Computer Architectures

- Architectures – from sensors to cloud
  - Edge processing: processing data where it is collected because the energy required to communicate data outweighs that of computation.
    - Leverage intermittent power (e.g., from harvested energy)
    - New communication modalities
    - New storage technologies (e.g., NVRAM).
    - Security and reliability
    - Approximate computing techniques -> energy savings
    - Portable edge devices: robots, google glasses
  - New architectures for DSP
    - Coarse grained reconfigurable architectures
  - Cloud
    - Need for tools that can partition between the edge and the cloud while responding dynamically to changes in the reliability and energy efficiency of the cloud uplink.
- Energy
  - For >1000 level parallelism, communication energy> computation energy
  - Parallelism, specialization
- New technologies
  - Memory
  - 3D chips
  - …

Our group at UOttawa and trends

- Data-centric personalized healthcare
  - Edge computing
- Accelerating propabilistic algorithms on FPGAs (particle filters)
  - Modifications of the algorithms to make computation and communication pattern deterministic
- Accelerating circuit simulators on GPUs
  - LU matrix factorization
  - >GB matrices that need to be factorized
  - Algorithms are modified so that the communication problem is reduced
- Decision making (AI) and machine learning with uncertainty

Questions