Matching the Architecture to the Application: Reconfigurable Computing

Andreas Koch
Embedded Systems and Applications Group (ESA)
Department of Computer Science
koch@esa.cs.tu-darmstadt.de

TECHNISCHE UNIVERSITÄT DARMSTADT

Embedded Systems & Applications

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

- Motivation
- Selected Application Successes
- Reconfigurable Device Architecture
- Application-Specific Computing Structures
- System Architecture
- Programming Issues
- Conclusion

Embedded Systems & Applications

Overview

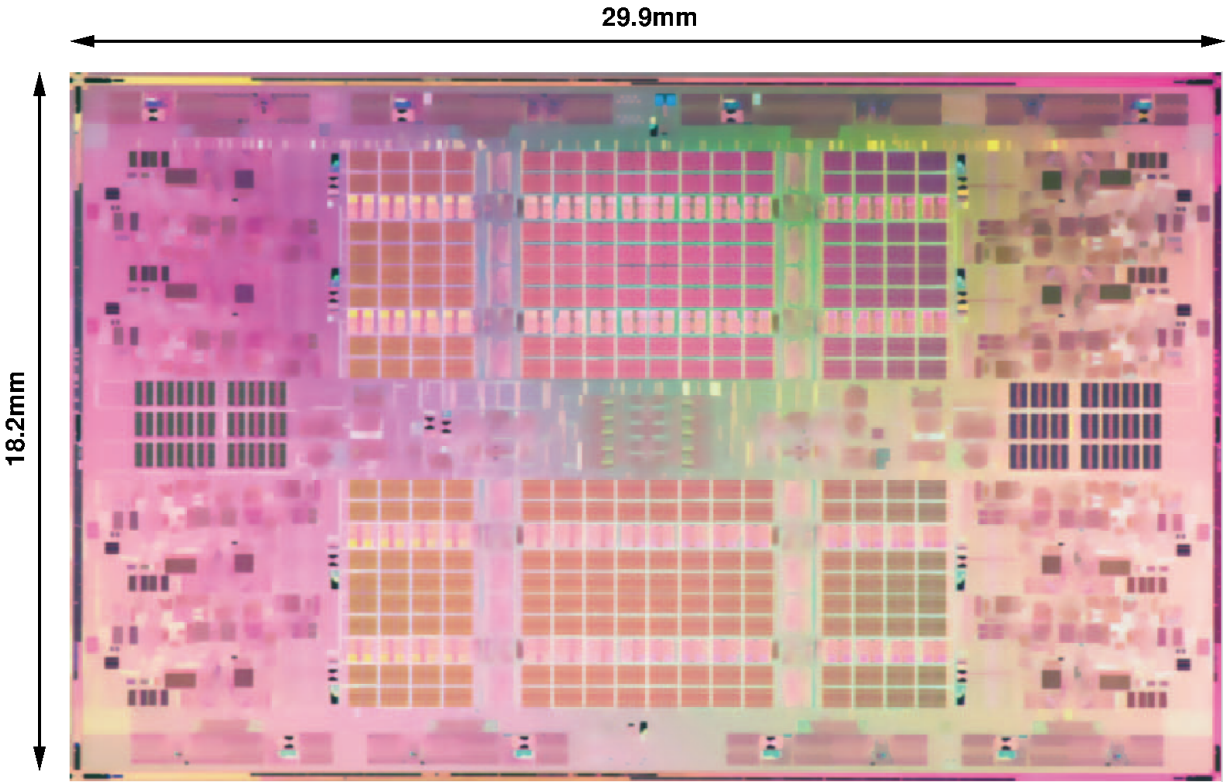
TECHNISCHE UNIVERSITÄT DARMSTADT

- **Motivation**
- Selected Application Successes
- Reconfigurable Device Architecture
- Application-Specific Computing Structures
- System Architecture
- Programming Issues
- Conclusion

Embedded Systems & Applications

Example of a Modern Microprocessor
Intel Poulson

TECHNISCHE UNIVERSITÄT DARMSTADT



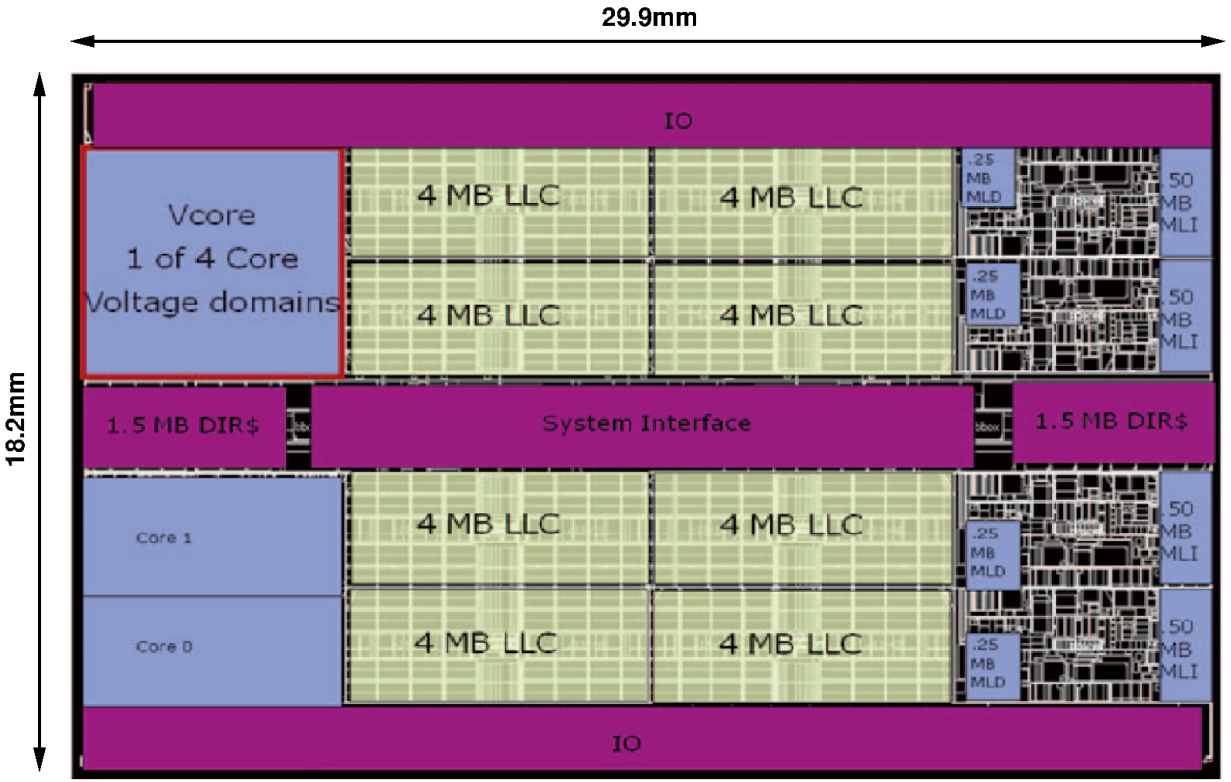
- $3.1 * 10^9$ transistors
- 544.18 mm^2

Source: Intel ISSCC 2011

Embedded Systems & Applications

Example of a Modern Microprocessor
Intel Poulson

TECHNISCHE UNIVERSITÄT DARMSTADT



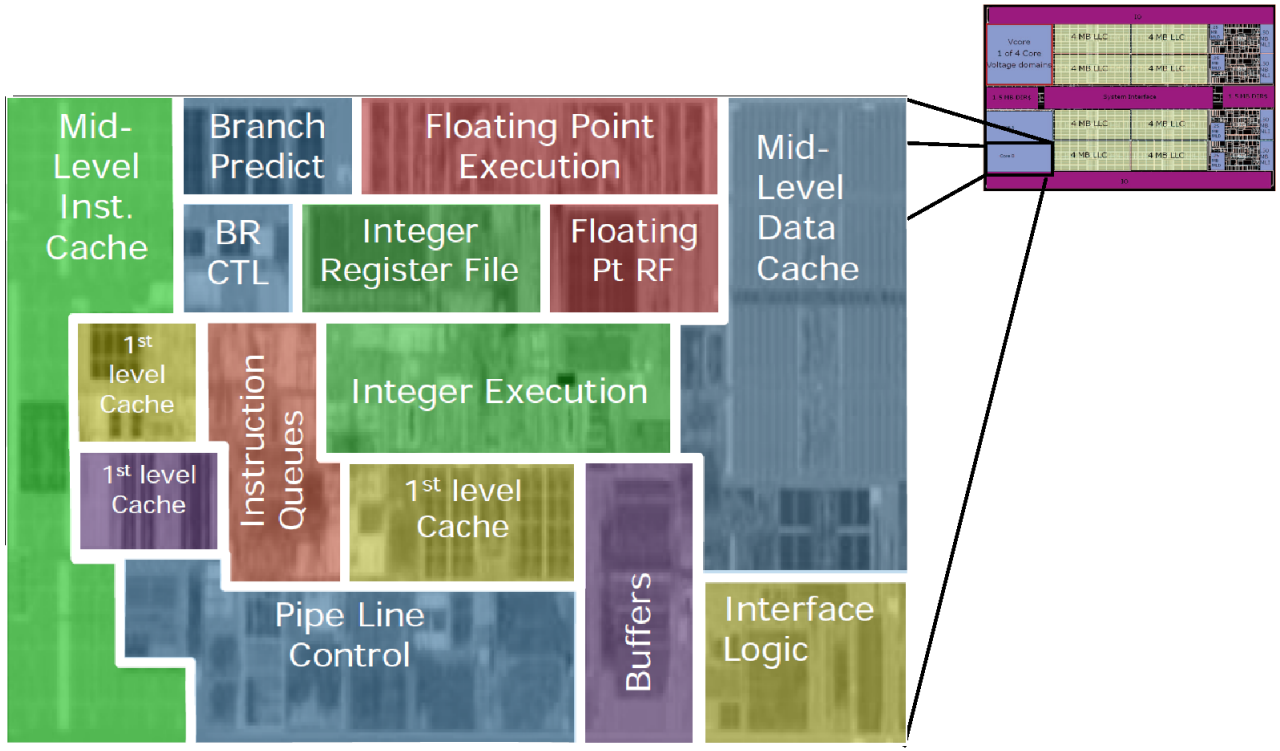
- 54 MB of on-chip memories
  - I/D-caches, coherency directories

Source: Intel ISSCC 2011

Embedded Systems & Applications

Example of a Modern Microprocessor
Intel Poulson

TECHNISCHE UNIVERSITÄT DARMSTADT



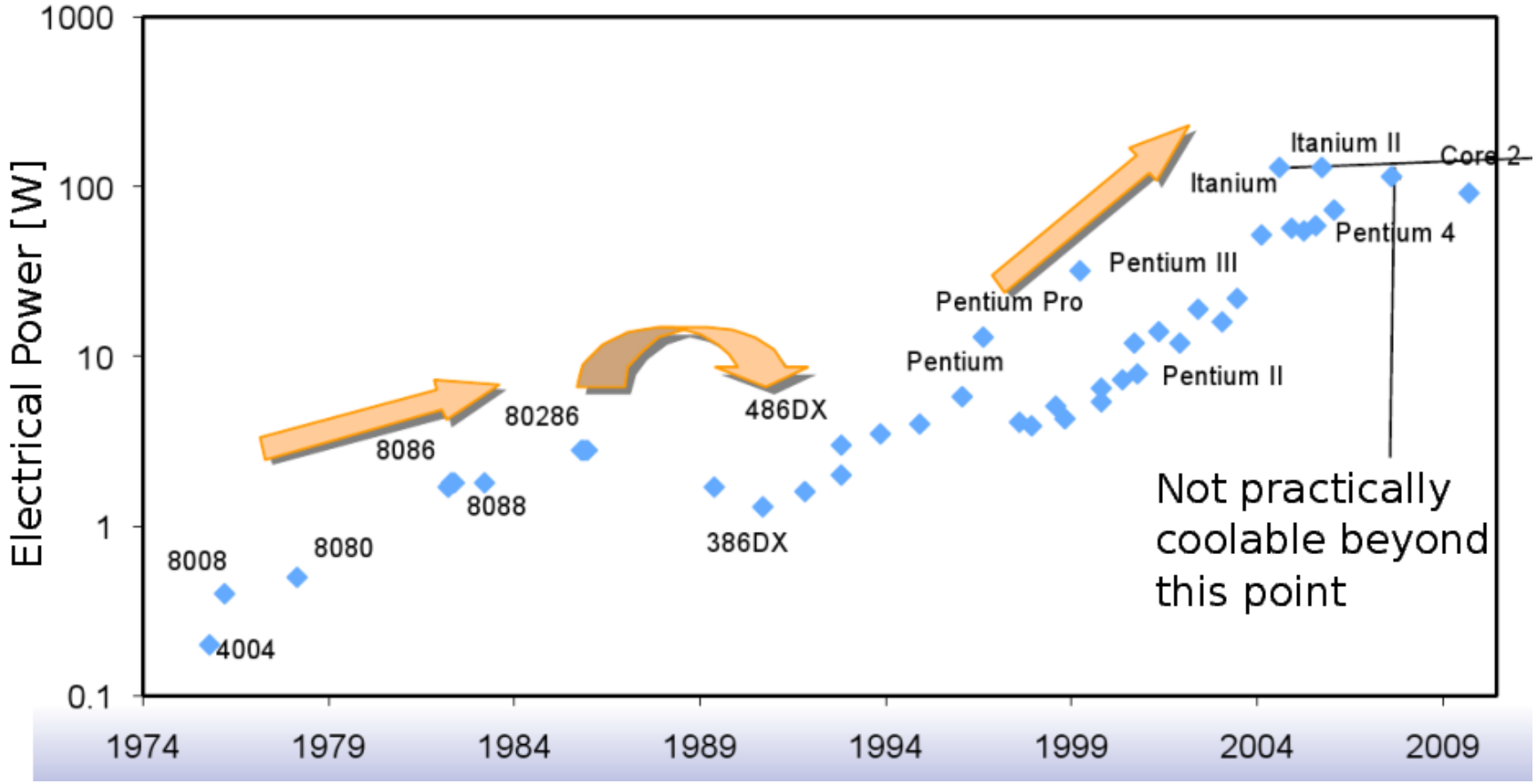
- Chip area used for actual computation?

Source: Intel ISSCC 2011

Embedded Systems & Applications

CPU Power Dissipation

TECHNISCHE UNIVERSITÄT DARMSTADT



Source: J. Rabaey, UC Berkeley

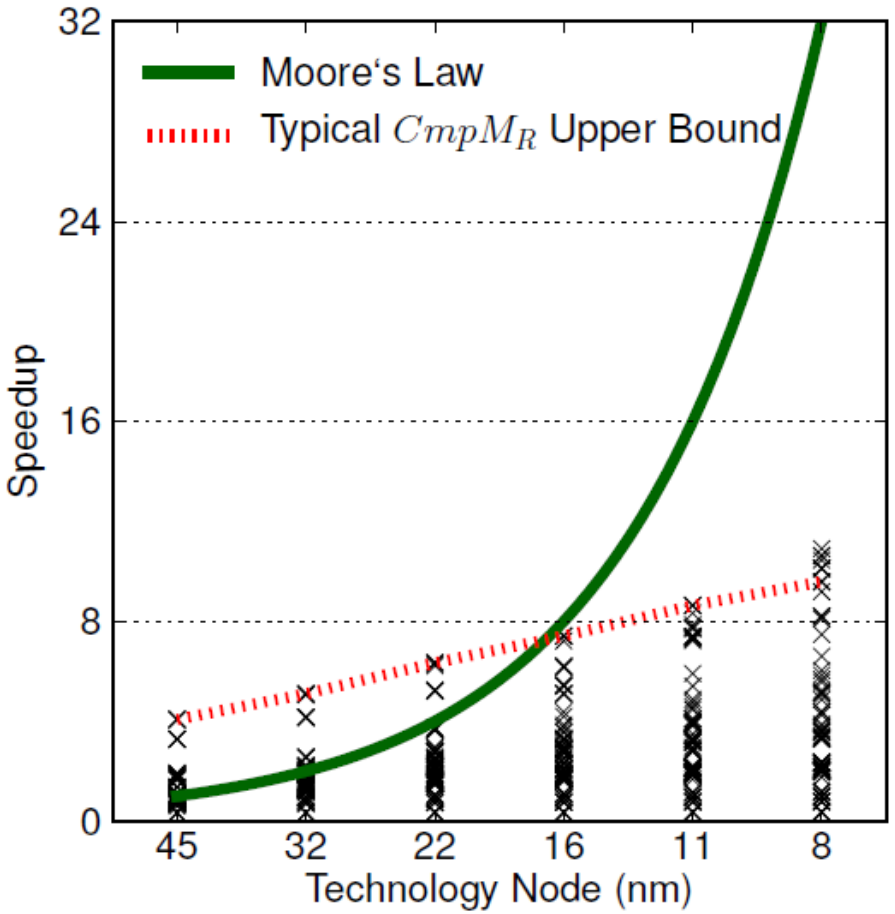
Embedded Systems & Applications

Many-Cores, the Panacea?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Highly parallel computations
  - PARSEC suite
- Executed on "best" CPU
  - Simulated
  - Exploring wide design spectrum
- Fixed area and power budgets
- More transistors
  - Moore's Law
  - Better chip fabrication processes

➔Performance does **not** scale with available transistors



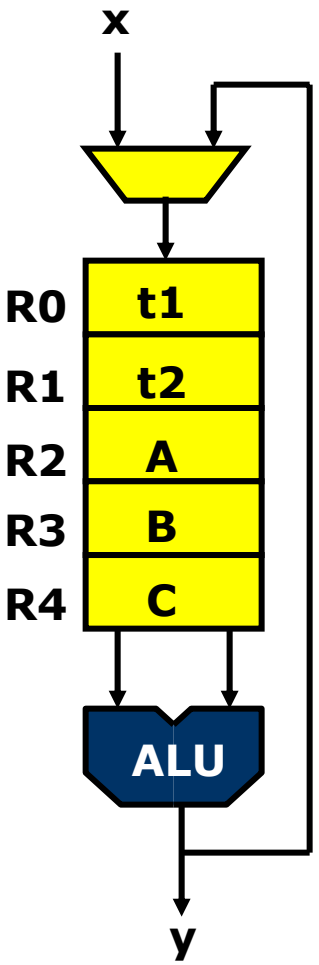
Source:H. Esmaeilzadeh et al., ISCA 2011

Embedded Systems & Applications

Conventional Execution Paradigm

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

- **Temporal** distribution of computation
- Across multiple instructions
- Reusing hardware unit(s)
- Computationally universal by changing the stored program




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

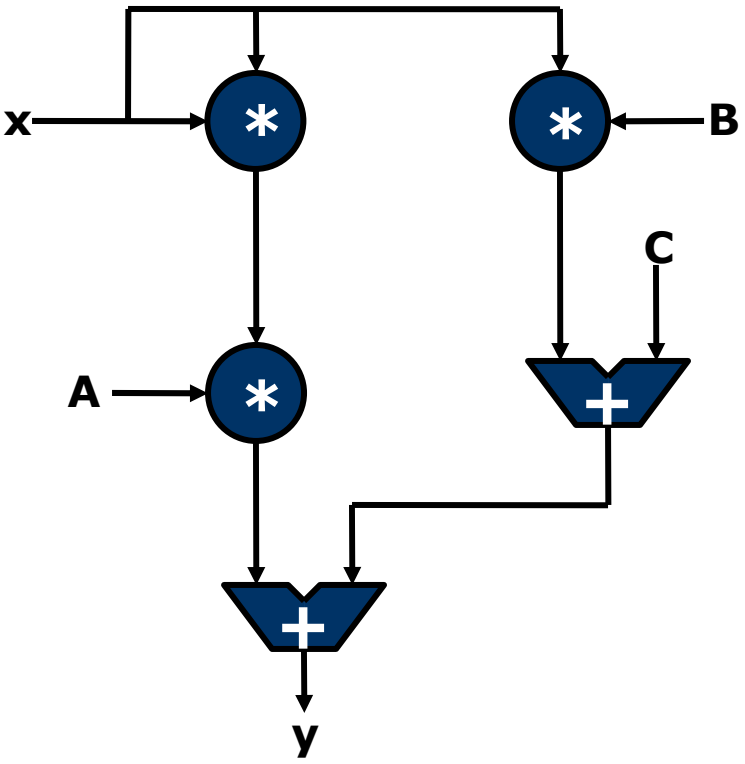
Alternative Execution Paradigm

TECHNISCHE UNIVERSITÄT DARMSTADT

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

- **Spatial** distribution of computation
- Dedicated hardware for each operation
- Match computing structure to application
- Universal by changing
  - Operators
  - Interconnections

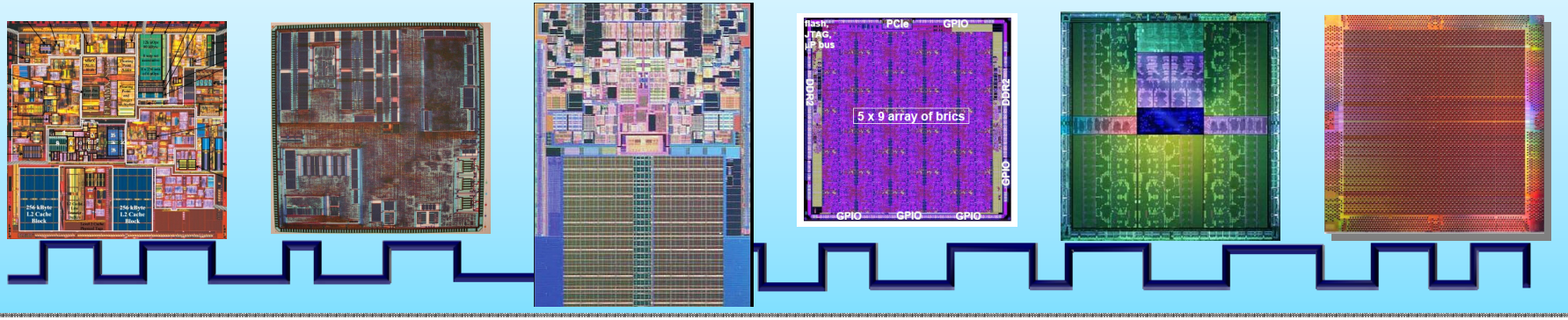
➔Enabled by **reconfigurable** circuits
  - Field Programmable Gate Arrays (FPGAs)



Embedded Systems & Applications

Widen Spectrum of Computing Architectures

TECHNISCHE UNIVERSITÄT DARMSTADT



CPUs | DSPs | Multi-Cores | Array | GPGPUs | FPGAs

Single Cores

Multi-Cores
Coarse-Grained
CPUs and DSPs

Coarse-Grained
Massively
Parallel
Processor
Arrays

Fine-Grained
Massively
Parallel
Arrays

Source: D. Singh, FPL 2012 Keynote

Embedded Systems & Applications

Overview

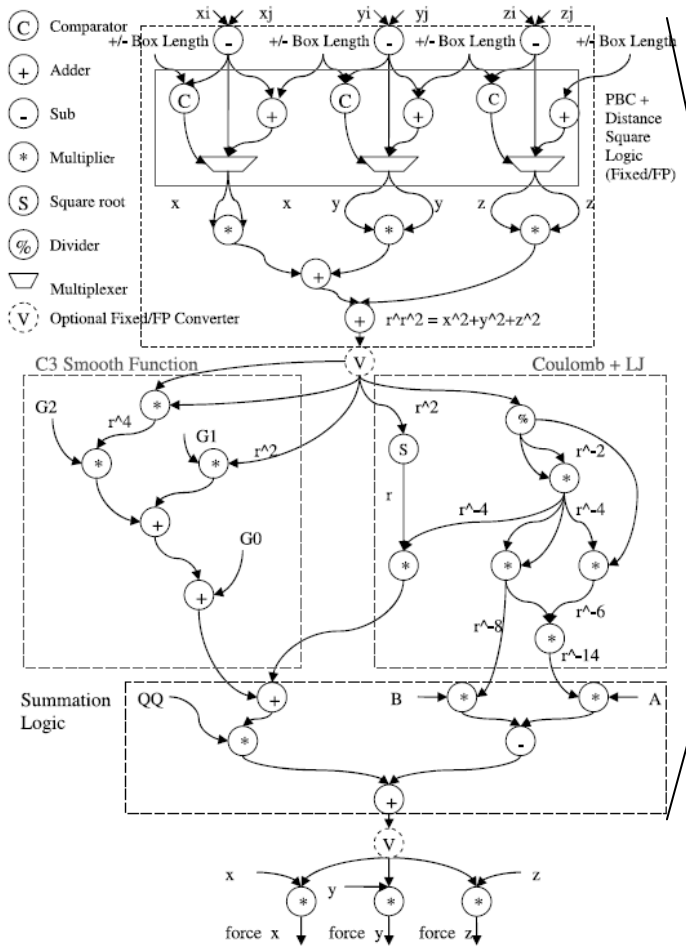
TECHNISCHE UNIVERSITÄT DARMSTADT

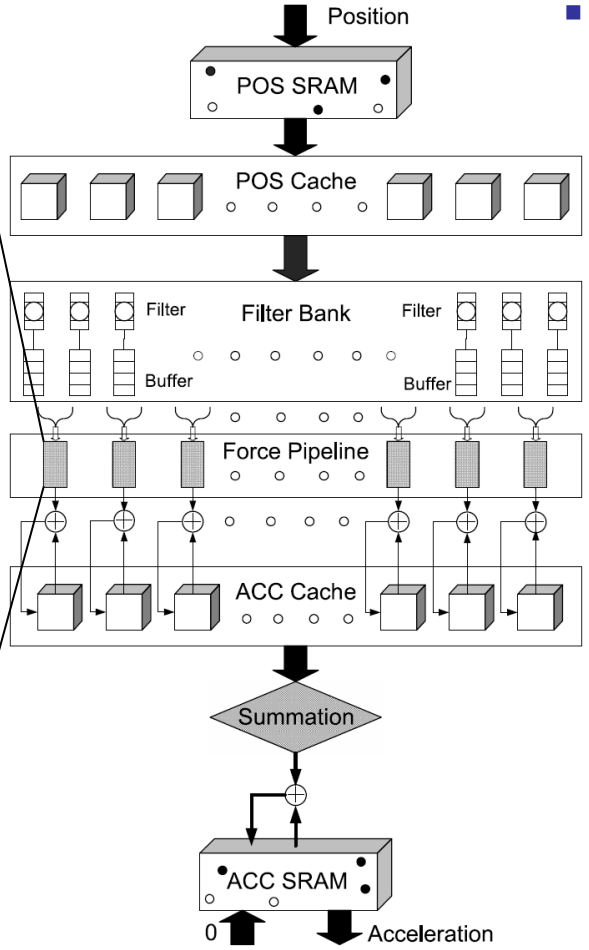
- Motivation
- **Selected Application Successes**
- Reconfigurable Device Architecture
- Application-Specific Computing Structures
- System Architecture
- Programming Issues
- Conclusion

Molecular Dynamics Simulation

Short range force computation on FPGA

TECHNISCHE UNIVERSITÄT DARMSTADT





- **20x speed-up** over NAMD on quad-core
  - ApoA1 in 22ms

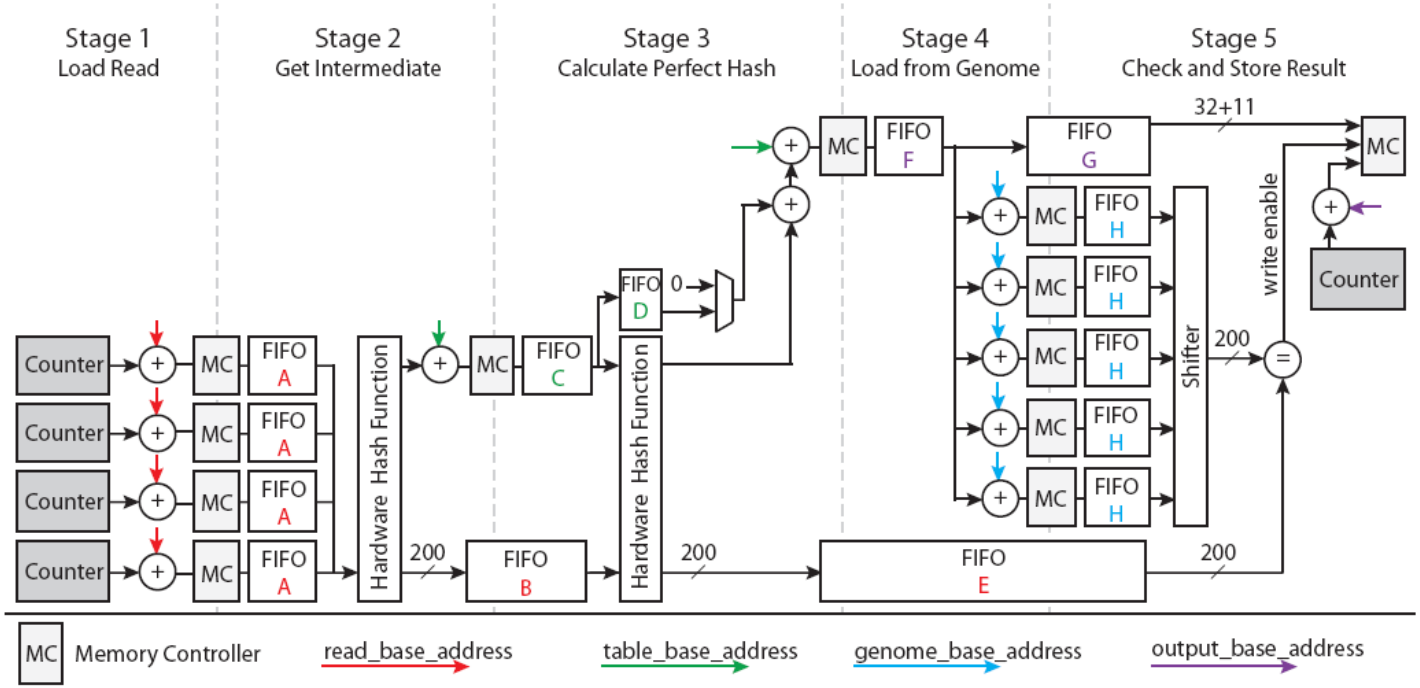
Source: M. Herbordt, ACM TRETS 11/2010

Embedded Systems & Applications

Accelerator for Short Read Alignment

Bioinformatics: Assemble subsequences in gene sequencing

TECHNISCHE UNIVERSITÄT DARMSTADT



Tool	Platform	Reads/Second
Bowtie	CPU	2,500
SOAP3	GPU	6,000,000
Shepard	FPGA	350,000,000

Source: C. Nelson, MEMOCODE 2012

Embedded Systems & Applications

Graph 500 Benchmark
Breadth-First-Search in Graphs

TECHNISCHE UNIVERSITÄT DARMSTADT

November 2012

No.	Rank	Machine	Installation Site	Number of nodes	Number of cores	Problem scale	GTEPS
1	1	DOE/NNSA /Lawrence Livermore National Laboratory Sequoia (IBM - BlueGene/Q, Power BQC 16C 1.60 GHz)	Lawrence Livermore National Laboratory	65536	1048576	40	15363
...							
43	43	Franklin (Cray - XT4)	Lawrence Berkeley National Laboratory	4000	16000	36	19.3274
44	44	fox6 (Convey - MX-100, host-210)	Convey Computer Corporation	1	16	29	14.56
45	45	Altix ICE 8400EX (SGI - Altix ICE 8400EX)	SGI	256	1024	31	13.9586
46	46	Nebulae (Sugon - TC3600 Blade System, Xeon X5650 6C 2.66GHz)	National Supercomputing Centre in Shenzhen	512	6144	33	12.119
...							

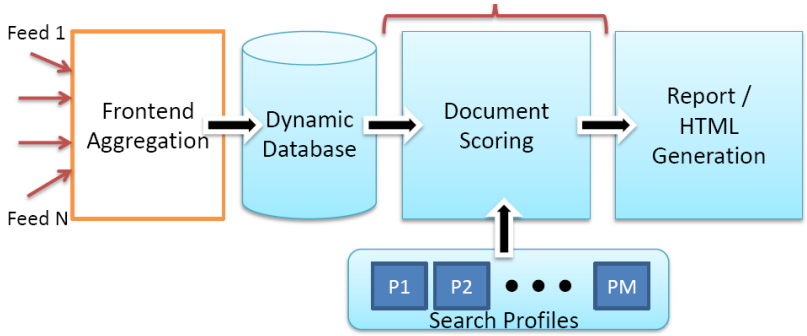
Source: www.graph500.org

Embedded Systems & Applications

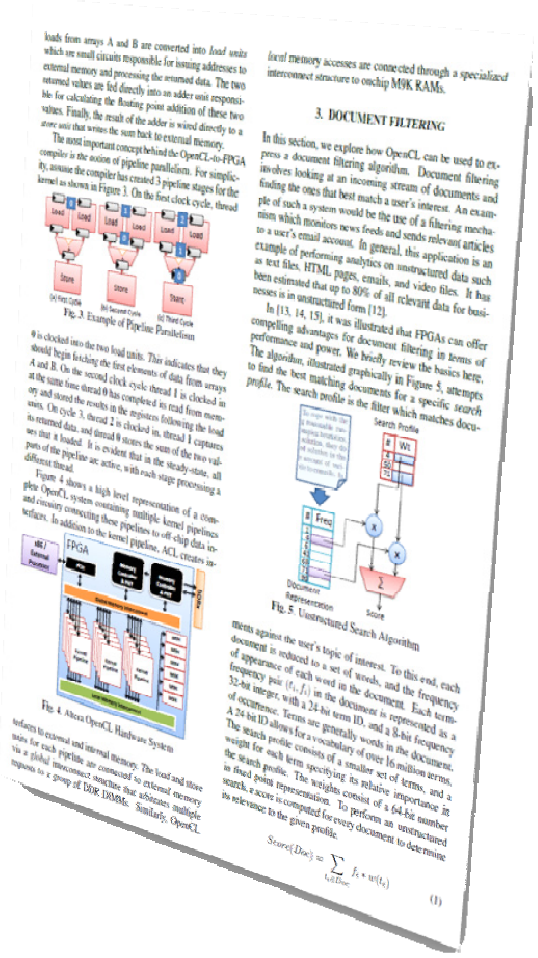
Document Filtering

TECHNISCHE UNIVERSITÄT DARMSTADT

- Unstructured data analytics
  - Based on Bloom Filter



Platform	Power (W)	Performance (MTs)	MTs/W	
W3690 Xeon Processor	130	2070	15.92	CPU
nVidia Tesla C2075	215	3240	15.07	GPU
DE4 Stratix IV-530 Accelerator	21	1755	83.57	FPGA
PCIe385 A7 Accelerator	25	3602	144.08	FPGA



Source: D. Singh, FPGA 2012 Tutorial

Embedded Systems & Applications

Color Edge Detection
Compiled from Geometric Algebra Description

TECHNISCHE UNIVERSITÄT DARMSTADT

Platform Processing 10240x10240 image	**Time** [ms]	**Power** [W]
Tesla C1060 **GPU** @ 1.3 GHz [canny detector, Ogawa 2010]	364.4	<200
i7-3930K **CPU** @ 3.2 GHz	3,236.0	<130
XC7VX690T-2 **FPGA** @ 0.4 GHz	262.1	0.6

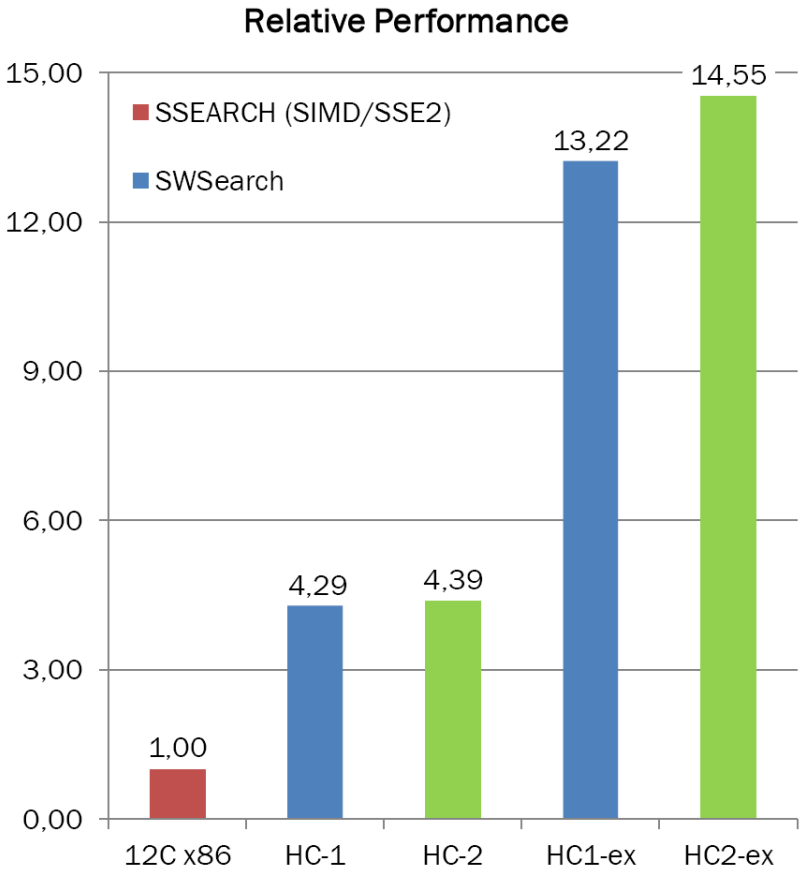
Embedded Systems & Applications

Smith-Waterman Search

Bioinformatics: Exact subsequence search

TECHNISCHE UNIVERSITÄT DARMSTADT

- **14.5x speedup** over fastest software approach on 12C x86
- SIMD approach 15-20x faster than naïve implementation
  - Can exploit larger FPGA capacities in EX-platforms



Source: Convey Computer

Embedded Systems & Applications

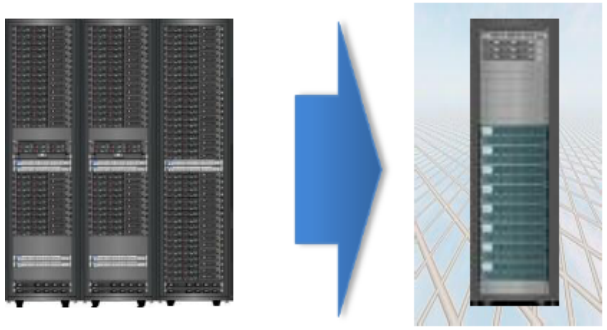
Smith-Waterman Search
Economical impact of performance per watt

TECHNISCHE UNIVERSITÄT DARMSTADT

PERF	HC-2ex 64/192 ≈ 15 X 12-Core 3.33 GHz x86		
POWER	Power Requirements[1]		
	1 racks (8 nodes) Convey	60.0	MW-h/yr
	4 racks (117 nodes) x86	354.0	MW-h/yr
	1 Year Electricity costs (@ 0.07 /kWh) [2]		
	Convey	8.3	K$/yr
	x86	49.5	K$/yr
SITE	1 Year Infrastructure costs[3]		
	Convey	15.2	K$/yr
	X86	89.9	K$/yr
TCO	3-Year TCO[4]		
	Convey	792	K$/yr
	X86	3,106	K$/yr

Reduction in floor space	75%
Reduction in datacenter watts	83%
Reduction in 3 yr TCO	75%

117 x 1U 12-core servers



8 x 4U Convey HC-2^{ex}

Source: Convey Computer

Embedded Systems & Applications

Overview

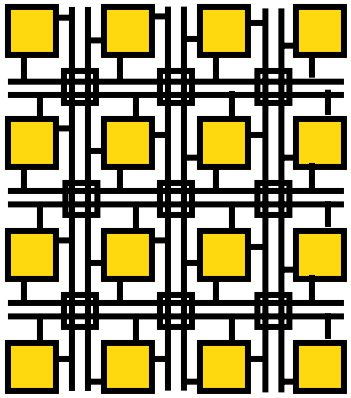
TECHNISCHE UNIVERSITÄT DARMSTADT

- Motivation
- Selected Application Successes
- **Reconfigurable Device Architecture**
- Application-Specific Computing Structures
- System Architecture
- Programming Issues
- Conclusion

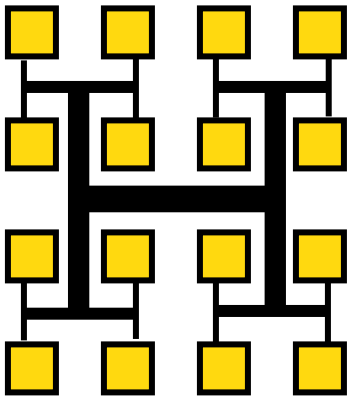
Embedded Systems & Applications

Examples for Reconfigurable Interconnect

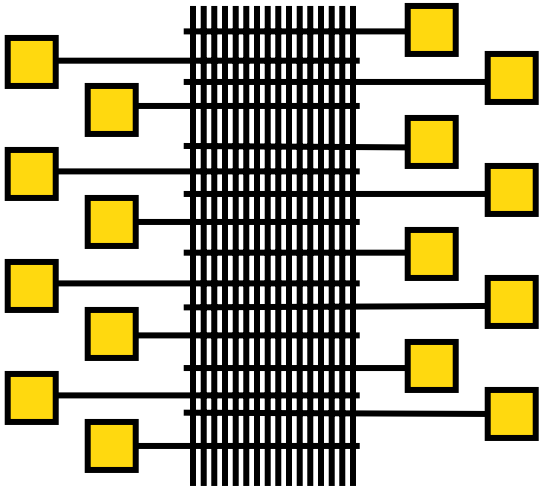
TECHNISCHE UNIVERSITÄT DARMSTADT



Symmetrical Matrix



Hierarchical Matrix



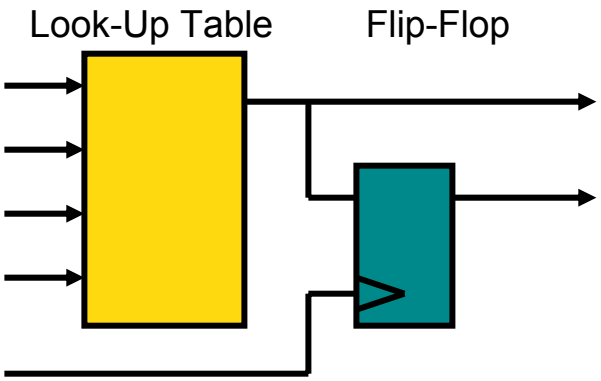
Crossbar

- Flexibly establish connections between processing elements
  - For FPGAs: Single-bit wires of multiple lengths
  - Linked using programmable switches
    - Pass transistors and multiplexers

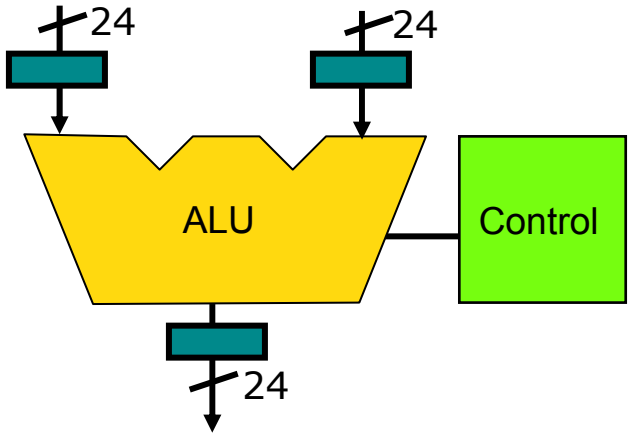
Embedded Systems & Applications

Examples for Reconfigurable Processing Elements

TECHNISCHE UNIVERSITÄT DARMSTADT



Fine grained (single bits)



Coarse grained (multi-bit words)

- Most common today: **Fine grained devices - FPGAs**
  - Look-Up Table computes boolean function of 6-7 inputs
  - Combination of computation and storage: Logic Cell / Logic Elements

Embedded Systems & Applications

Heterogeneous FPGA Fabrics

TECHNISCHE UNIVERSITÄT DARMSTADT

- Not just LUTs+FFs
- More **specialized** blocks
  - Many 18x18 … 27x27 Multipliers/MAC
  - Many small on-chip memories
  - High-speed I/O (28.5 Gb/s per pin)

Logic Fabric
LUT-6 CLB

Precise, Low Jitter Clocking
MMCMs

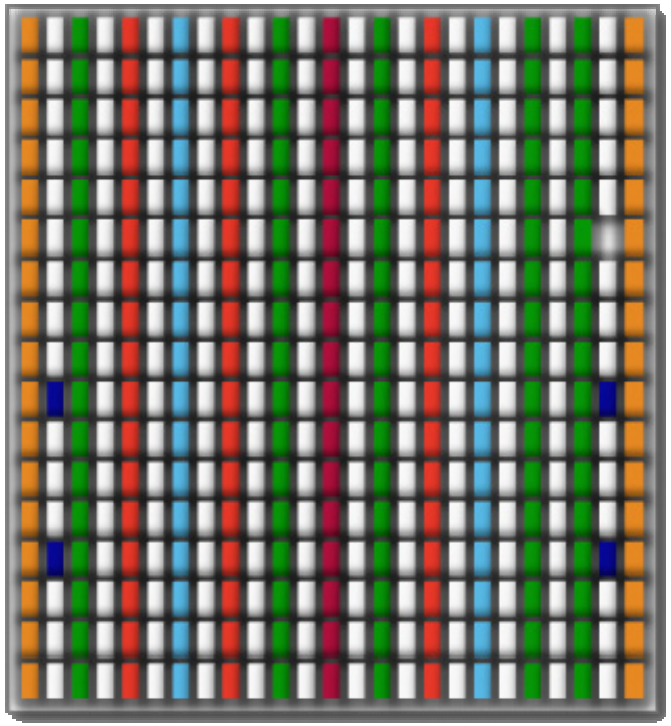
On-Chip Memory
36Kbit/18Kbit Block RAM

Enhanced Connectivity
PCIe® Interface Blocks

DSP Engines
DSP48E1 Slices

Hi-perf. Parallel I/O Connectivity
SelectIO™ Technology

Hi-performance Serial I//O Connectivity
Transceiver Technology



Source: Xilinx 7 Series Overview

Embedded Systems & Applications

Major FPGA Vendors

Current 28nm FPGA families well suited for computing

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Altera Corp.**
  - **Stratix V** series: 5SEEB
  - 952,000 Logic Elements
  - 704 multiply/MAC 18b x 18b
  - 352 multiply/MAC 27b x 27b
  - 17,960 MLAB memories (11 Mb)
    - 32x20b, 64x10
  - 2,640 M20K memories (52 Mb)
    - 16Kx1b…512x40b
  - 6 DDR3-SDRAM interfaces
    - Hardwired
  - Max. internal clock 700+ MHz
    - In practice up to 450 MHz
- **Xilinx Inc.**
  - **Virtex 7** series: XC7VX1140T
  - 1,139,200 Logic Cells
  - 3,600 DSP blocks 25b x18b
  - 283,200 SelectRAM memories (17 Mb)
    - 64x1b
  - 1,500 BlockRAM memories (67 Mb)
    - 32Kx1 … 512x72b
  - 4 PCI Express GEN3 Interfaces
    - Hardwired
  - Max. internal clock 600+ MHz
    - In practice up to 450 MHz

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

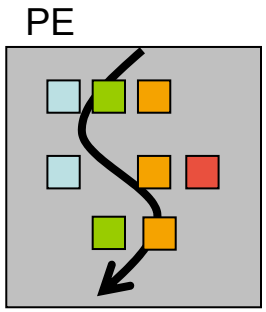
- Motivation
- Selected Application Successes
- Reconfigurable Device Architecture
- **Application-Specific Computing Structures**
- System Architecture
- Programming Issues
- Conclusion

Embedded Systems & Applications

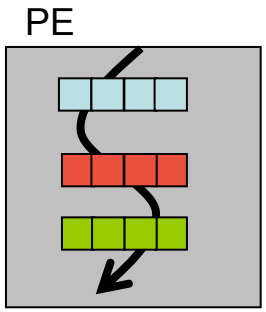
Example Models of Computation

Match to the Needs of the Application

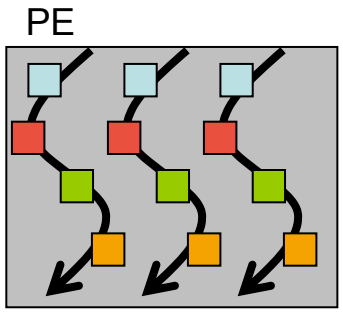
TECHNISCHE UNIVERSITÄT DARMSTADT



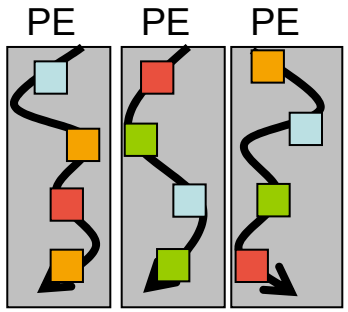
Single Threaded
ILP



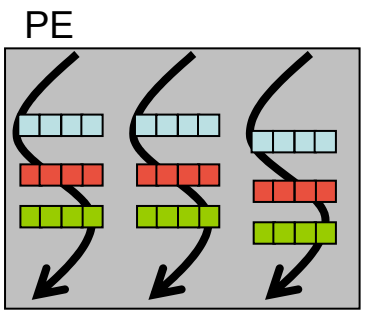
Single Threaded
Vector



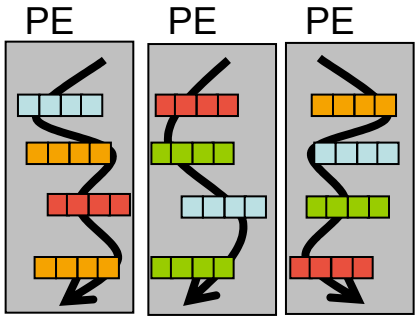
Multi-Threaded
SIMT



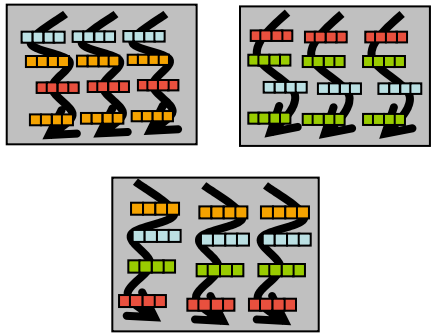
Single-Threaded
MIMD



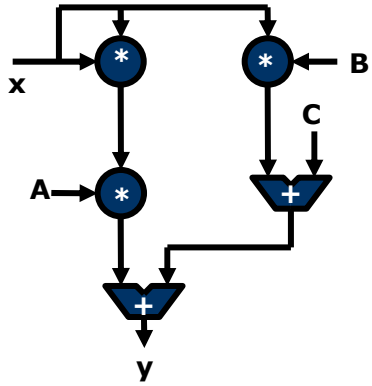
Multi-Threaded
SIMT / Vector



Single-Threaded
MIMD / Vector



Multi-Threaded
MIMD / Vector / SIMT



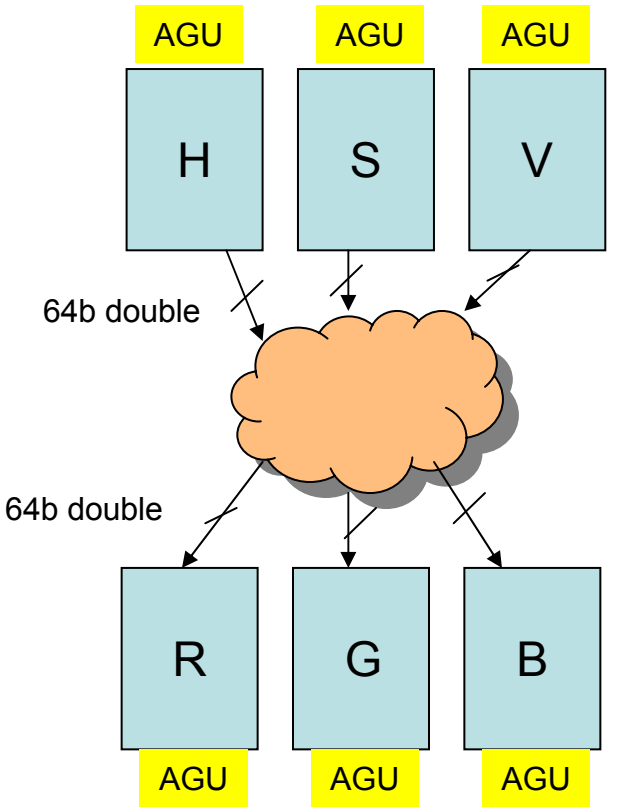
Data Flow

Embedded Systems & Applications

Memory System Variations 1

TECHNISCHE UNIVERSITÄT DARMSTADT

- Local (on-chip) memories
- Small capacity, but …
- **… fully independent accesses**
  - Single and dual-ported operation
- Example
  - Virtex 7 XC7VX1140T
  - 1,500 BlockRAMs, total ca. 8 MB
  - Aggregate bandwidth
    1,500 x 9B x 400 MHz = **5.4 TB/s**
- Common usage pattern
  - Loop tiling (to fit in 8 MB)
  - Then assign each data structure to dedicated memory



6 accesses in parallel

Embedded Systems & Applications

Memory System Variations 2

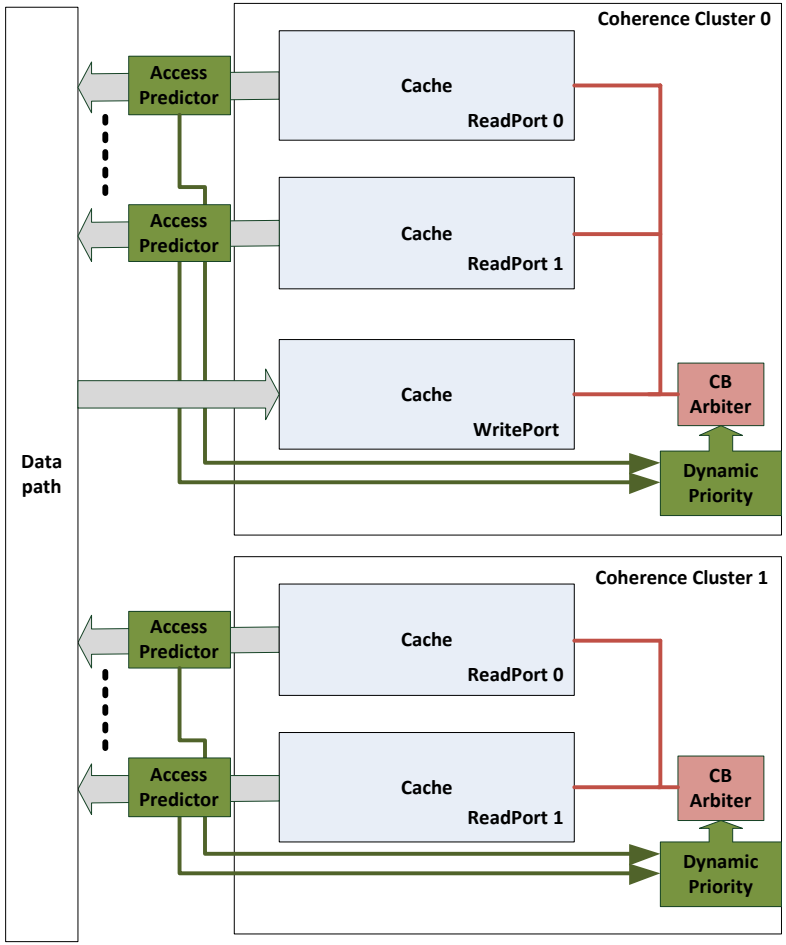
TECHNISCHE UNIVERSITÄT DARMSTADT

- Multi-PE shared memory system
- With **configurable coherency**
  - Generate dedicated coherency/snoop network for specific application
  - Avoid bottlenecks
- Example:

```
int a[], b[]; // non-overlapping

a[i]=  a[i] + a[i+1] +
       b[i] + b[i+1];
```

- No coherency traffic between a[] and b[]

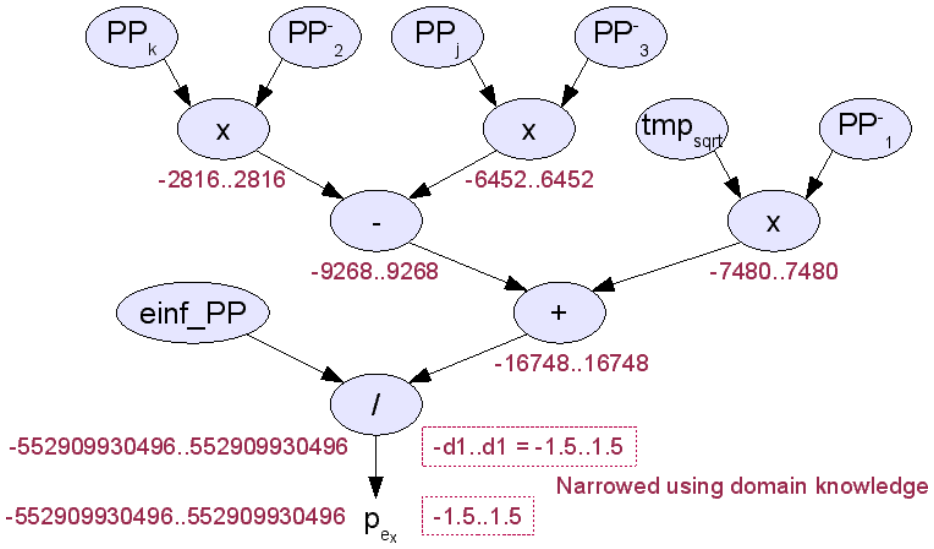


Embedded Systems & Applications

Custom Number Formats 1

TECHNISCHE UNIVERSITÄT DARMSTADT

- Binary **fixed-point** representation
- N.M
  - N bits integer part
  - M bits binary fraction
- Advantage
  - No normalization necessary
- Disadvantage
  - Limited dynamic range
- Generate hardware operators specialized for each required range and precision
  - Faster, smaller, and lower power than floating-point



Embedded Systems & Applications

Custom Number Formats 2

TECHNISCHE UNIVERSITÄT DARMSTADT

- Non-standard floating point formats
- Example: XMIR
  - Residual calculation for dense linear system solver
  - Iterative refinement
  - Arbitrary accuracy (exceeding 64b double supported by GPGPUs)

Precision		Pipeline Depth (Add/Mult)	DSP 48E	Slices Registers LUTs	# of BRAM (36Kb)	# of PEs s_2	PAR CLK	GFLOPs
Exp Size	Mantissa Size							
8	23(S)	11/8	4/2,016	1,278/595,200 ***1,748/297,600***	4/1,064	170	210 MHz	**71**
11	38	12/12	5/2,016	2,355/595,200 ***2,807/297,600***	6/1,064	106	167 MHz	**35**
11	52(D)	14/15	13/2,016	2,912/595,200 ***3,546/297,600***	8/1,064	83	125 MHz	**21**
15	63	13/22	16/2,016	3,816/595,200 ***4,517/297,600***	8/1,064	65	178 MHz	**23**

  - Other approach: Trade bandwidth over accuracy (8b float, REAL*1)

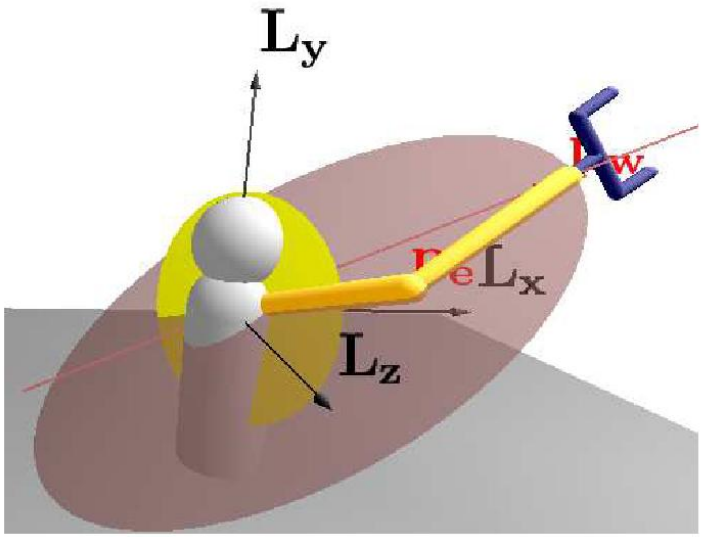
Source: J. K. Lee, U Tennessee, SAAHPC 2011

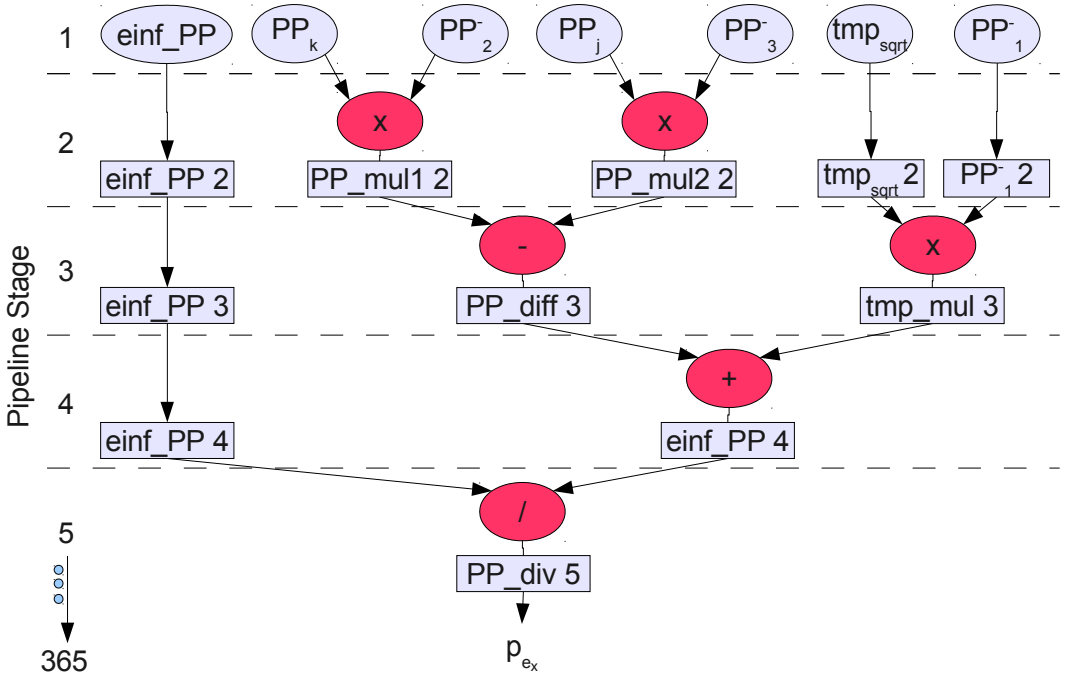
Embedded Systems & Applications

Deep Pipelining for Data Flow

Inverse Kinematics compiled from Geometric Algebra

TECHNISCHE UNIVERSITÄT DARMSTADT





- **140 parallel operators**, organized in 365 stages
- 8x throughput of 2.4 GHz quad-core Xeon, but draws only 7 W

Embedded Systems & Applications

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

- Motivation
- Selected Application Successes
- Reconfigurable Device Architecture
- Application-Specific Computing Structures
- **System Architecture**
- Programming Issues
- Conclusion

Embedded Systems & Applications

General Purpose Prototyping 1
GiDEL ProceV

TECHNISCHE UNIVERSITÄT DARMSTADT

- Entry-level systems
  - EUR 1,000 – EUR 20,000
- PCI Express Expansion Cards
  - GEN2 or GEN3
- Usually aimed at hardware development, not specifically for computing
- Operating system drivers provided
- Otherwise very limited programming support
  - Fall back to FPGA vendor tools
- Need **significant hardware expertise** to bring-up for practical use

- Altera Stratix V GS D8
- PCI Express GEN3 x8
- 2x 16GB DDR3 ECC SODIMM (19.2 GBs/)
- 2x 144Mb SRAM (6.4 GB/s, low latency)
- **1x 100G**, 2x 10G Ethernet

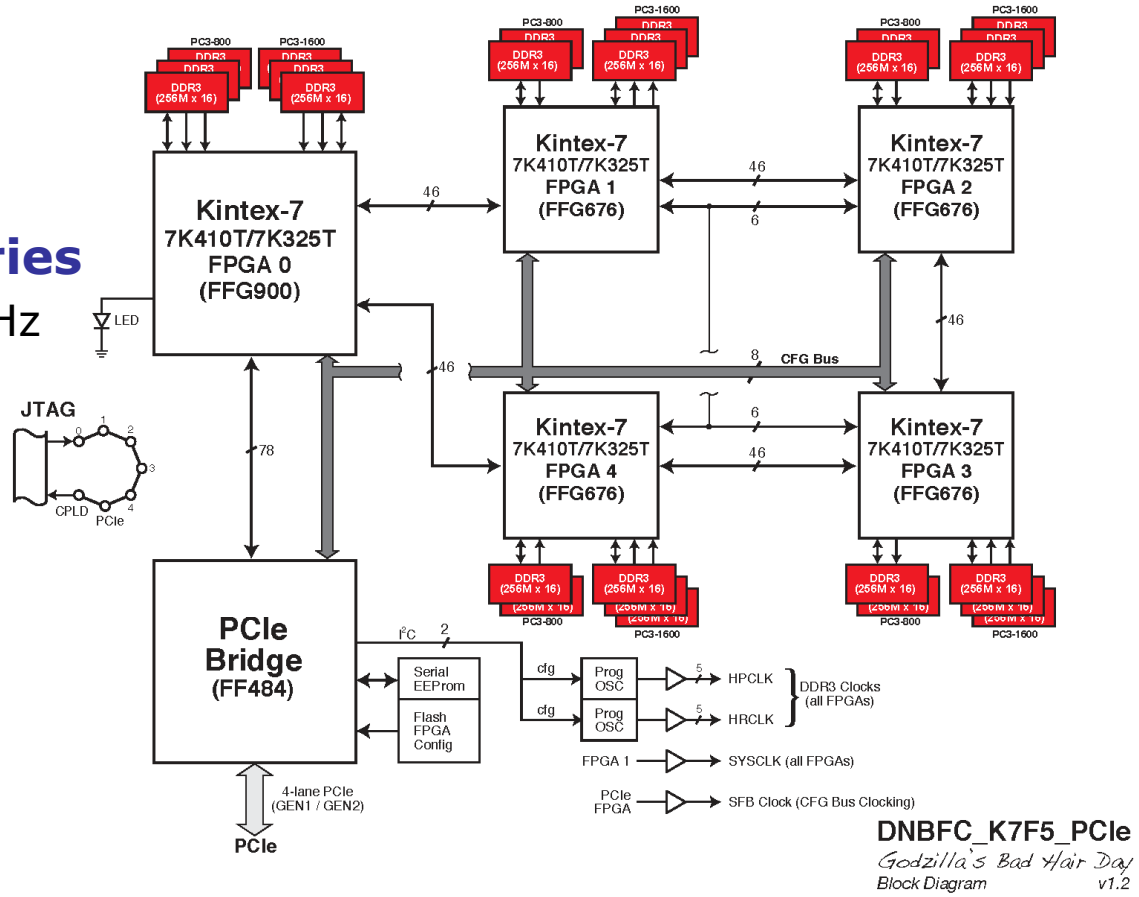
Source: GiDEL Ltd.

Embedded Systems & Applications

General Purpose Prototyping 2
DINI Group DNK7_F5PCIe

TECHNISCHE UNIVERSITÄT DARMSTADT

- Multi-FPGA approach
- 5x Kintex-7 7K410T
- PCI Express GEN2 x4
- **26 independent memories**
  - DDR3-SDRAM 1600/800 MHz
  - 65.6 GB/s total external bandwidth



Source: Dini Group, Inc.

Embedded Systems & Applications

Turnkey Computing Systems 1

TECHNISCHE UNIVERSITÄT DARMSTADT

- Specialized for reconfigurable **computing**
- Often delivered as complete systems
  - To avoid main-board compatibility issues
  - EUR 4,500 … EUR 50,000+

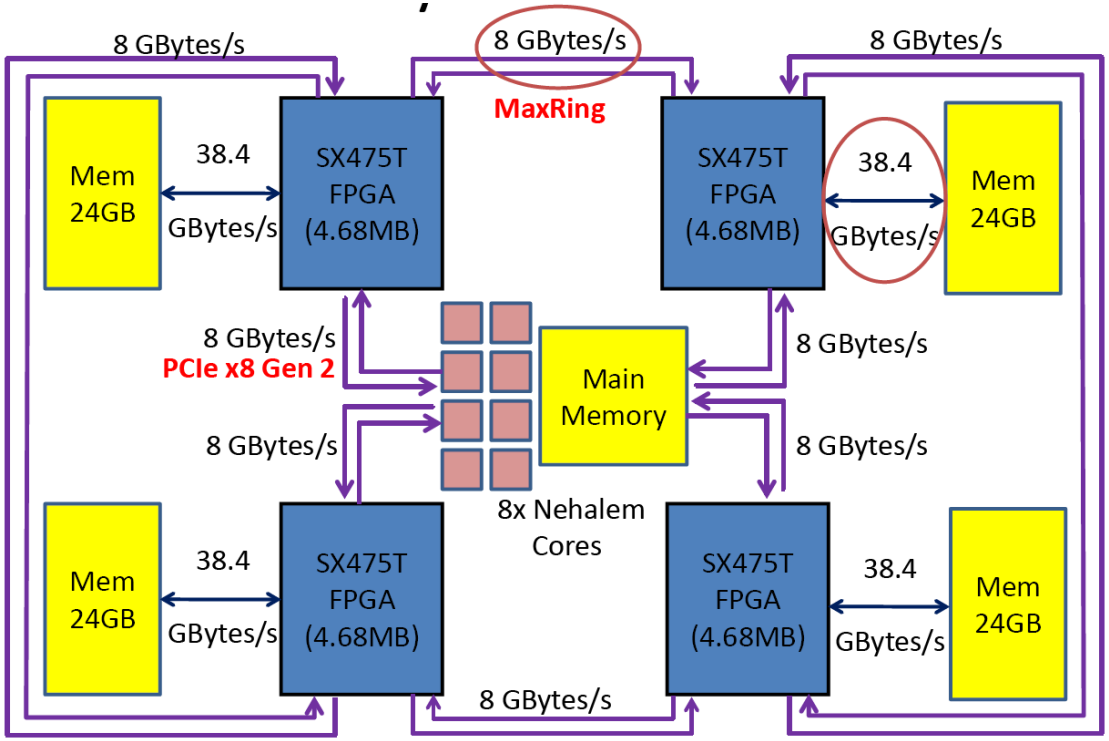
- **Comprehensive tool support**, reaching from …
  - ❶ Complete integration of FPGA vendor tools
    - Still requires hardware design expertise
    - But avoid engineering effort of bringing-up system
  - ❷ Specialized high-level compilers
    - Significantly reduced design effort
    - … but some computer architecture expertise required for optimization
  - ❸ Complete hardware accelerators („personalities“) available off-the shelf
    - Bioinformatics, MD, financial analytics, …
    - Usable from software as library calls

Embedded Systems & Applications

Turnkey Computing Systems 2
MAXELER MaxWorkstation / MPC-X

1 Data Flow Engine

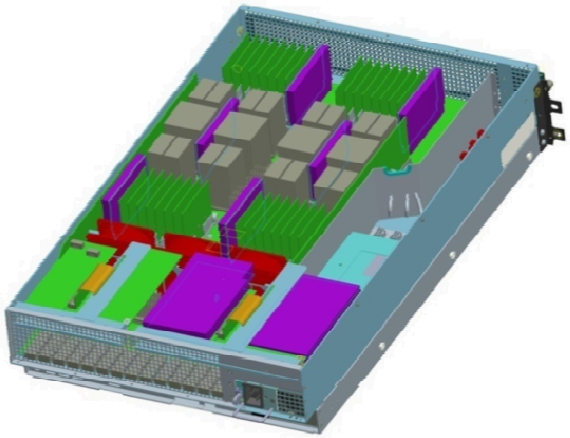
8 Data Flow Engines
Infiniband Interconnect

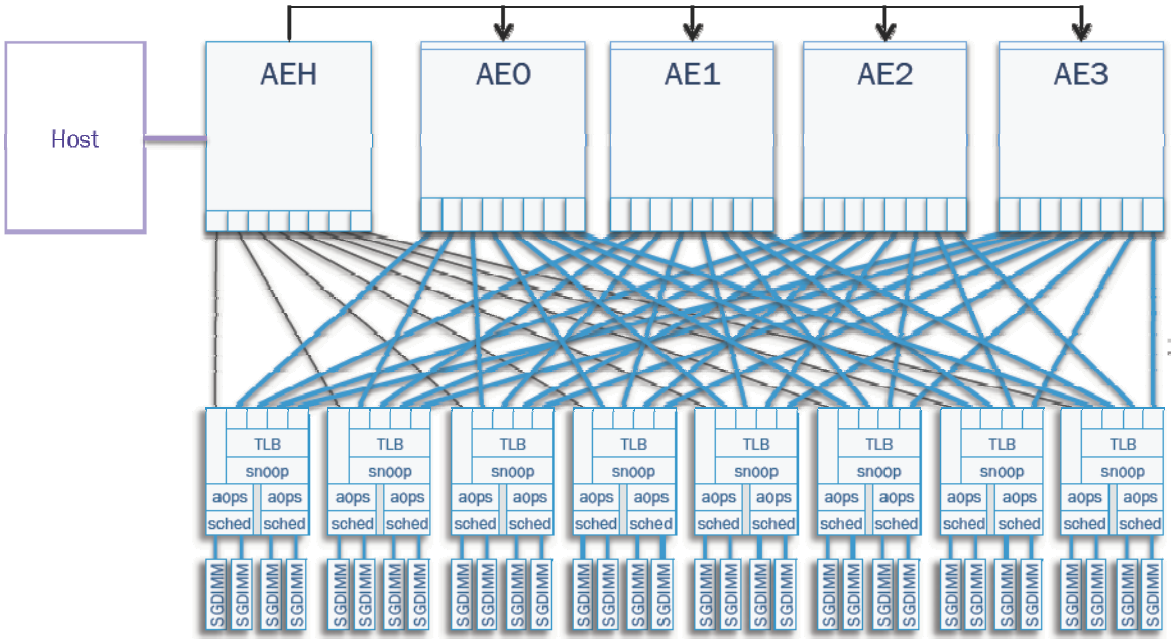


- Specialized for stream-based programming (special language)
  - **153.6 GB/s aggregate memory bandwidth (only for streaming!)**
- Scalable systems: multi-engine, multi-node

Source: O. Lindtjorn, Hot Chips 2010

Turnkey Computing Systems 3
Convey Computer MX-100





- **Up to 1 TB memory per node, 100 GB/s bandwidth**
  - Globally shared, cache coherent virtual address space between host and accelerator
- Wide spectrum of high-performance bioinformatics personalities
- User-programmable also in OpenMP as MIMD array of 64b RISC cores

Source: J. Leidel, Irregular Applications Workshop, SC 2012

Larger Reconfigurable Computers
Maxwell and Novo-G

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Maxwell (2007)**
  - University of Edinburgh, EPCC
- 32 nodes
  - Each: Single-core Xeon + 2 Virtex 4 FPGAs
  - 16 GB DDR2-SDRAM per FPGA
  - FPGAs linked into 8x8 torus
- Example: Black-Scholes option pricing
  - Speed-Up: 300x over 16 plain Xeon cores

Source: www.fphca.org, R. Baxter et al., Engineering Letters 16:3, 2008

- **Novo-G (2010)**
  - NSF Center for High-Performance Reconfigurable Computing (CHREC)
- 24 nodes
  - Each: Quad-core Xeon + 6 Stratix III FPGAs
  - 4 GB DDR2-SDRAM per FPGA
- Example: Needleman-Distance Application
  - Speed-Up 365,000x over single Opteron core

Source: www.chrec.org

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

- Motivation
- Selected Application Successes
- Reconfigurable Device Architecture
- Application-Specific Computing Structures
- System Architecture
- **Programming Issues**
- Conclusion

Embedded Systems & Applications

Obstacle to Reconfigurable Computing
Programming Support

TECHNISCHE UNIVERSITÄT DARMSTADT

- Commonly requires
  - Expertise in digital logic design an computer architecture
  - Proficiency in specialized hardware design languages (Verilog/VHDL)
  - Familiarity with specialized hardware design tools
    - Simulation, Synthesis, Place & Route, Power Analysis, …
- Skills generally **unavailable** to most application developers

- **Situation is improving**
  - Higher abstraction levels supported in languages and tools
  - However: Possible performance penalty compared to low-level designs

Embedded Systems & Applications

Register-Transfer Level Design
Hardware Description in Verilog or VHDL

TECHNISCHE UNIVERSITÄT DARMSTADT

```
// Excerpt from Daubechies-Fouveau wavelet
module wavelet_8_4_l (
        CLK,
        RESET,
        ENABLE,

        STREAM_READ,
        STREAM_WRITE,

        ROW_WIDTH
    );

    // Inputs
    input           CLK;
    input           RESET;
    input           ENABLE;
    input  [31:0]   STREAM_READ;
    input  [8:0]    ROW_WIDTH;

    // Outputs
    output [31:0] STREAM_WRITE;

    wire            CLK;
    wire            RESET;
    wire            ENABLE;
    wire   [31:0]   STREAM_READ;
    wire   [8:0]    ROW_WIDTH;
    reg    [31:0]   STREAM_WRITE;
…
```

Very close to hardware

- Bit-granularity descriptions
- Synchronize execution at clock edges
- Explicitly formulated parallelism
- Memory interface already abstracted

```
// Parallel computation in hardware datapath
assign A_NEW = A0 + XB_PLUS_B_NEW_SHIFTED;
assign XB_PLUS_B_NEW = XB + B_NEW;
assign XB_PLUS_B_NEW_SHIFTED[15] = XB_PLUS_B_NEW[15];
assign XB_PLUS_B_NEW_SHIFTED[14:0]= {XB_PLUS_B_NEW[15],
   XB_PLUS_B_NEW[15], XB_PLUS_B_NEW[15],
   XB_PLUS_B_NEW[14:3]};

assign B_NEW = (B0 << 1) - A0 - C0; …

// Pipeline controller
always @(posedge CLK or posedge RESET) begin …
   if (ENABLE) begin
     A1 <= A2;
     B1 <= B2;
     C1 <= C2;
     D1 <= D2;
     A0 <= A1;
     B0 <= B1;
     C0 <= C1;
     D0 <= D1;
     XB <= D_NEW;

     if (COUNT_COLUMN == 2)
       STREAM_WRITE <= {A_NEW_BEGINNING, C_NEW};
     else if (COUNT_COLUMN == 1)
       STREAM_WRITE <= {A_NEW, C_NEW_END};
     else
       STREAM_WRITE <= {A_NEW, C_NEW};

     if (COUNT_COLUMN == ROW_WIDTH - 1) begin
       COUNT_COLUMN <= 0;
     end else begin
       COUNT_COLUMN <= COUNT_COLUMN + 1;
     end …
```

Parallel

Embedded Systems & Applications

Powerful Types and Atomic Rules

Bluespec SystemVerilog, inspired by Haskell

TECHNISCHE UNIVERSITÄT DARMSTADT

```
interface SudokuSolver#(numeric type order);

    // set the cell value indexed by row and column to a given value
    method Action setCellValue(Index#(order) r, Index#(order) c,
  Cell#(order) v);

    // retrieve the cell value at given row and column
    method Cell#(order) getCellValue(Index#(order) r, Index#(order)
  c);

    // start solving the current grid
    method Action startSolver();
  …
module mkSolver(SudokuSolver#(order))
    provisos(Mul#(order,order,size), Mul#(size,size,num_cells),
             HasTactics#(order),
             Add#(_,TLog#(order),TLog#(TSquare#(order))),
             Bits#(Tactic#(order),sz0),
             Bits#(TacticResult#(order),sz1));

    SudokuRegGrid#(order) grid <-
  replicateM(replicateM(mkReg(unknown())));

    Tactics#(order) tactics <- mkSudokuTactics();
…
function Action apply(Tactic#(order) tactic);
    action
      let tactic_result = unknown();
      case (tactic) matches
          tagged Singleton {group: .g, idx: .i}:
            tactic_result = tactics.elim_other_singletons(g,i);
          tagged Elimination {group: .g, idx: .i}:
            tactic_result = tactics.process_of_elimination(g,i);
          tagged Pairs {group: .g, idx: .i}:
            tactic_result = tactics.repeated_2_set(g,i); ...
```

```
module mkTactics(Tactics#(order))
  provisos(Mul#(order,order,size),
           Add#(1,_,SizeOf#(Cell#(order))));

  method Cell#(order)
            elim_other_singletons(
                      Group#(order) g,
                      Index#(order) n);
    return ~singletons(maskOne(g,n))
  endmethod: elim_other_singletons
  …
```

- Fully synthesizable to hardware
- Abstract types converted to bits
- Parallelism formulated as **atomically executing** rules
- Control/arbitration logic generated automatically

Source: Bluespec Inc.

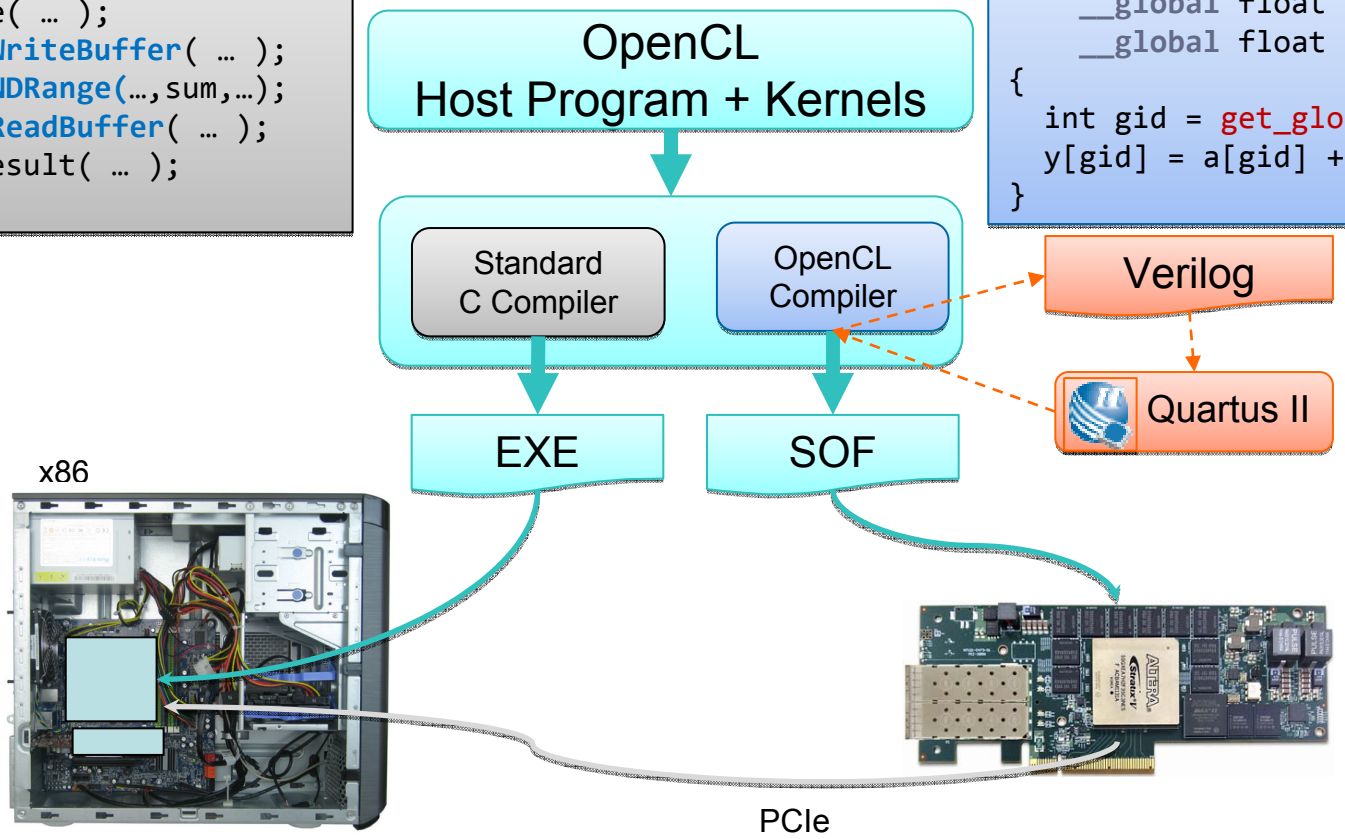
Embedded Systems & Applications

OpenCL for Altera FPGAs 1
Tool Flow

TECHNISCHE UNIVERSITÄT DARMSTADT

```
main() {
   read_data( … );
   manipulate( … );
   clEnqueueWriteBuffer( … );
   clEnqueueNDRange(…,sum,…);
   clEnqueueReadBuffer( … );
   display_result( … );
}
```

```
__kernel void
sum(__global float *a,
    __global float *b,
    __global float *y)
{
  int gid = get_global_id(0);
  y[gid] = a[gid] + b[gid];
}
```

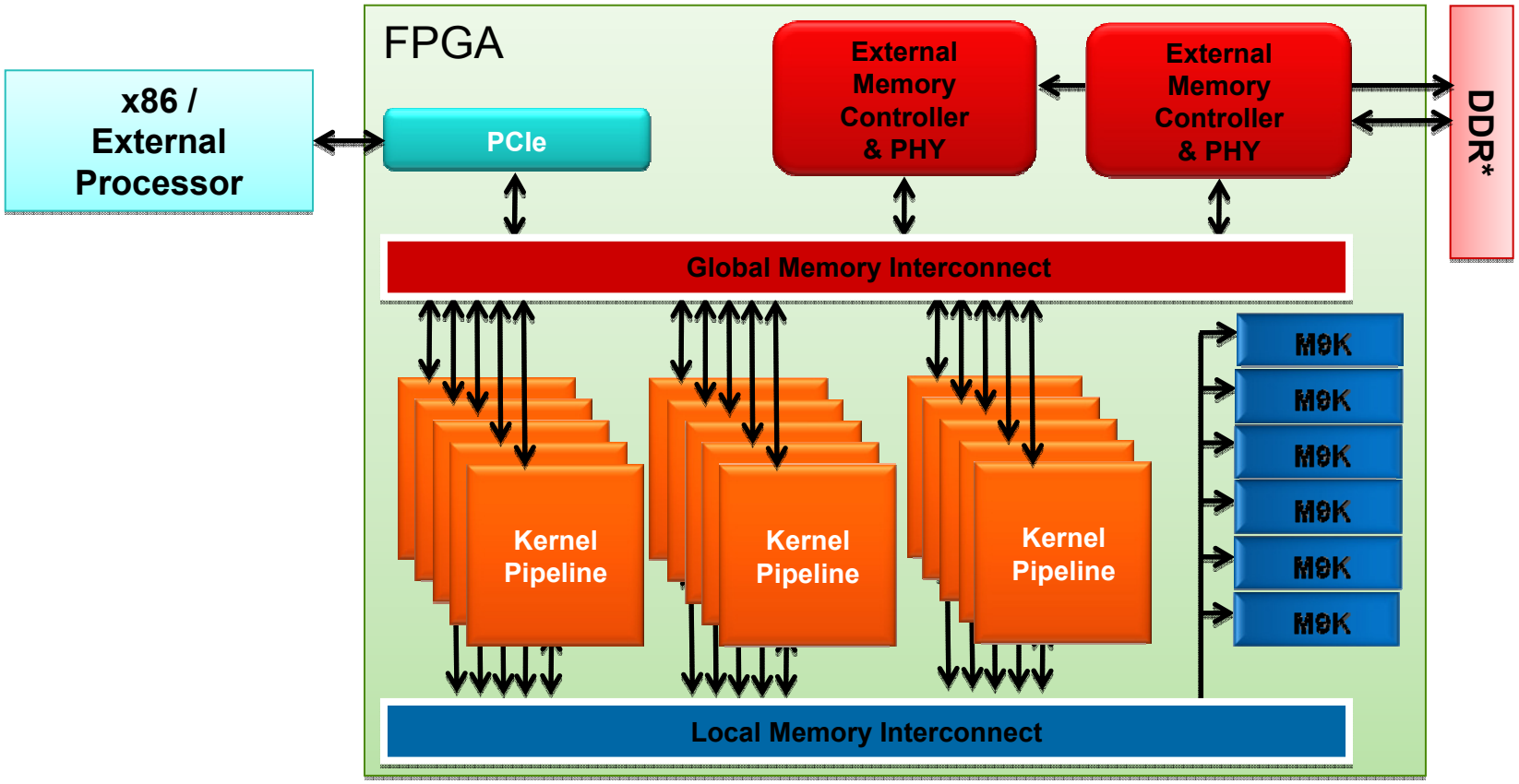


Source: D. Singh, Tutorial, FPGA 2013

Embedded Systems & Applications

OpenCL for Altera FPGAs 2
On-Chip Infrastructure

TECHNISCHE UNIVERSITÄT DARMSTADT



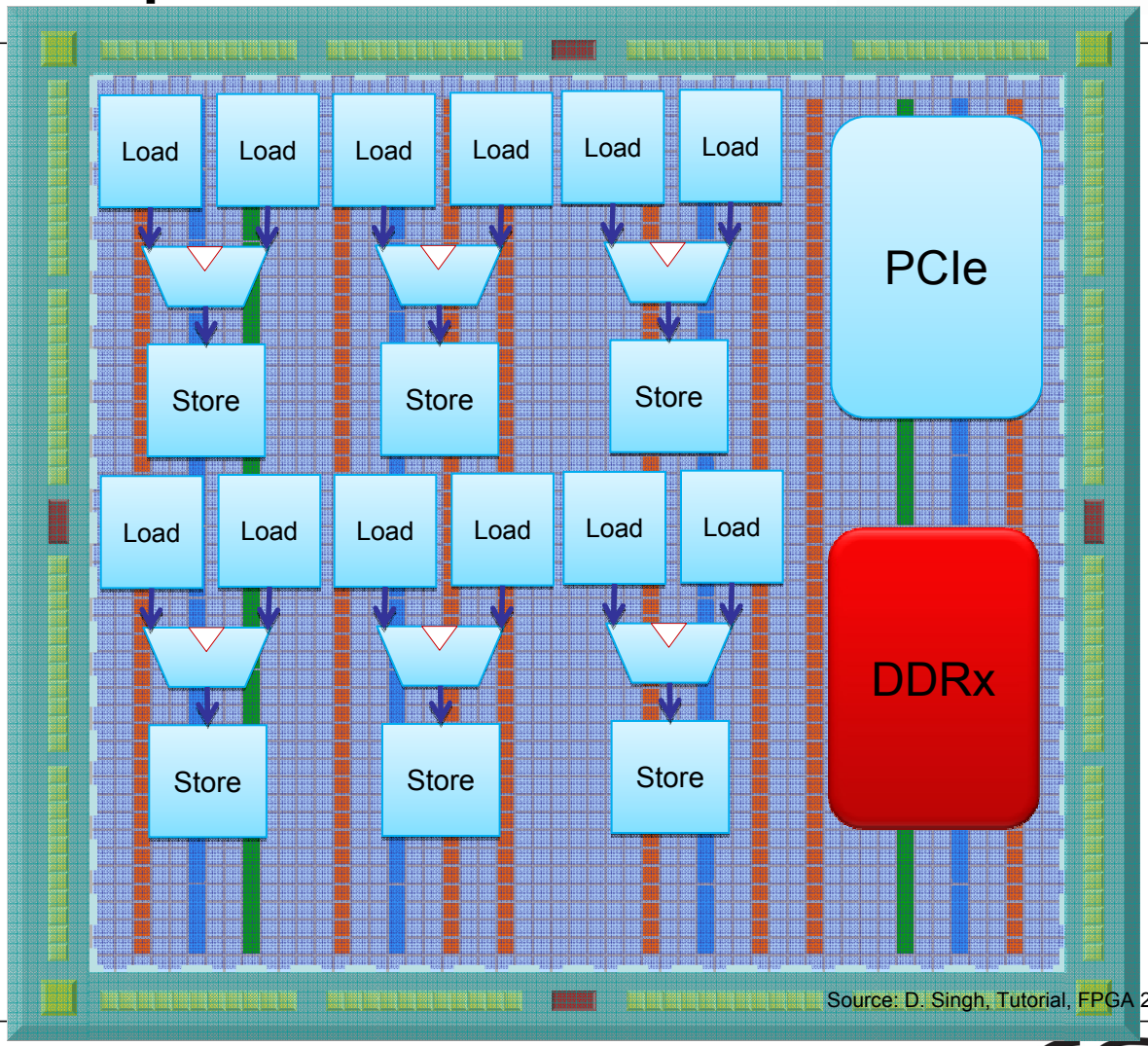
Source: D. Singh, Tutorial, FPGA 2013

Embedded Systems & Applications

OpenCL for Altera FPGAs 3
Micro architecture for compute kernels

TECHNISCHE UNIVERSITÄT DARMSTADT

```
__kernel void
sum(__global const float *a,
__global const float *b,
__global float *answer)
{
int xid = get_global_id(0);
answer[xid] = a[xid] + b[xid];
}
```



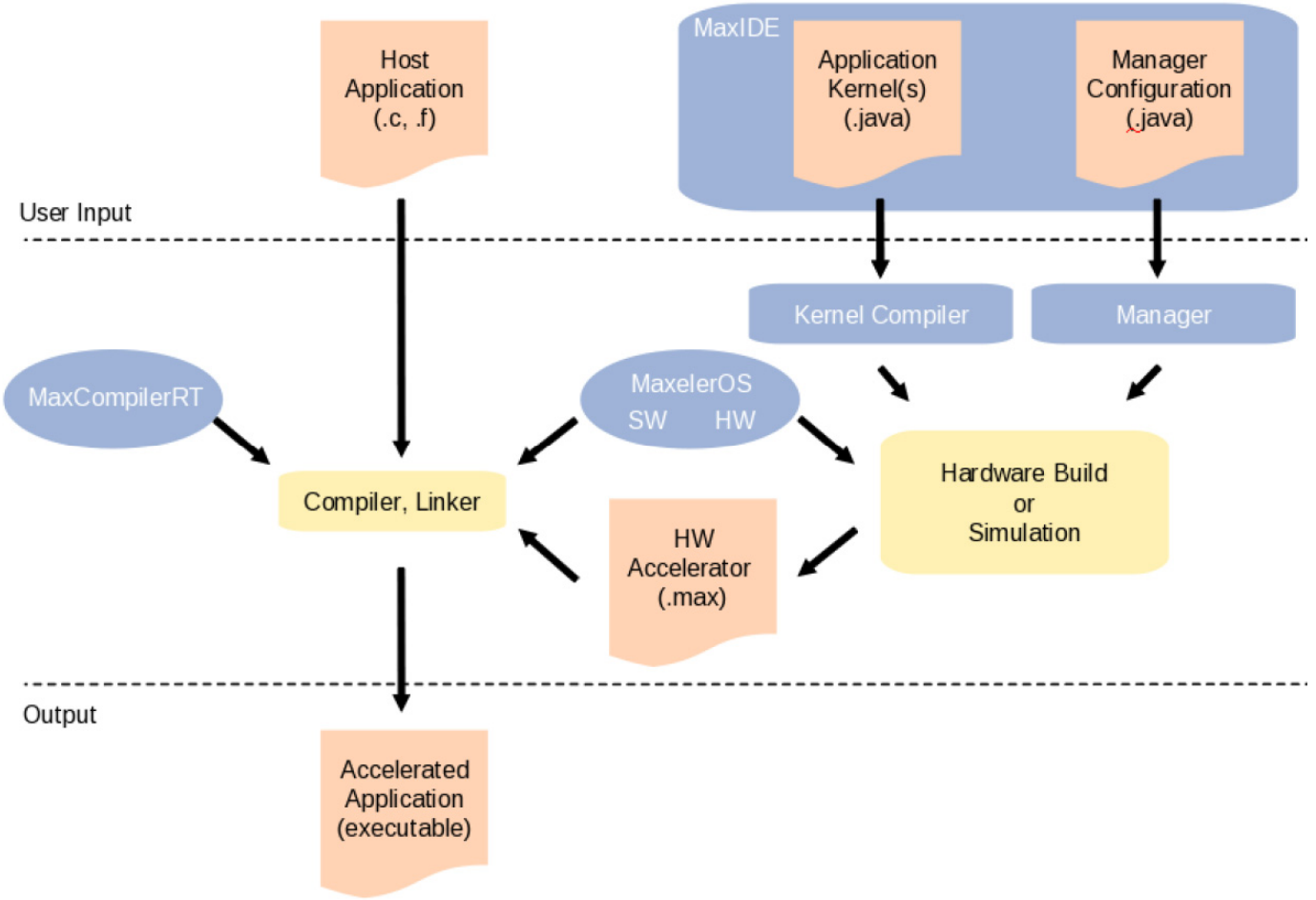
Source: D. Singh, Tutorial, FPGA 2013

- Tuning knobs, e.g.
  - Memory allocation
    - Global
    - Local
    - Constant Cache
  - Loop unrolling
  - Vectorization
  - Relaxed floating point computations

Embedded Systems & Applications

Maxeler Data Flow Programming 1
Tool Flow

TECHNISCHE UNIVERSITÄT DARMSTADT



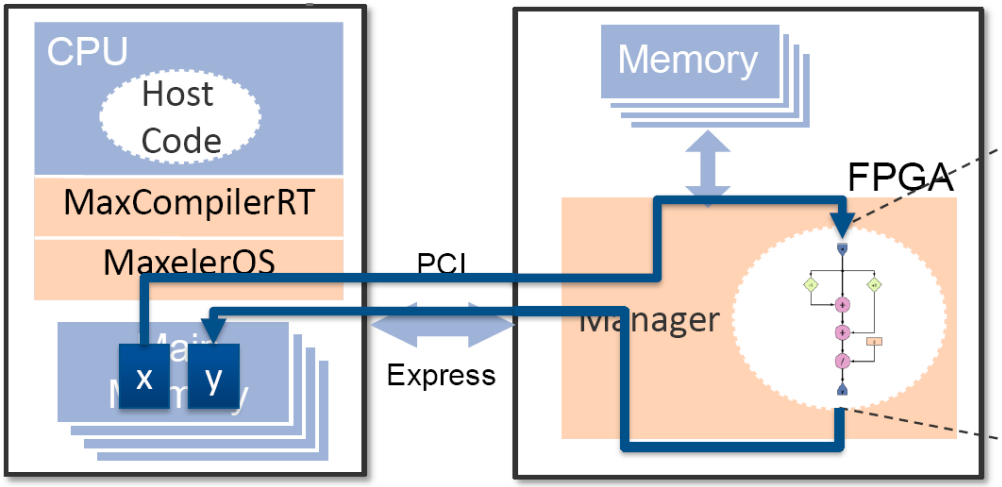
Source: O. Pell, Tools & Techniques for Accelerating HPC Applications, SEG Workshop 2011

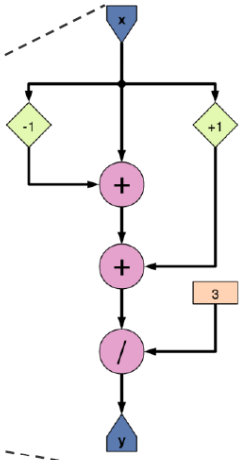
Embedded Systems & Applications

Maxeler Data Flow Programming 2

Example: Moving average $y_i = (x_{i-1} + x_i + x_{i+1})/3$

TECHNISCHE UNIVERSITÄT DARMSTADT





Host Code (.c)

```
float *x, *y;


max_run(device,
   max_input("x", x, DATA_SIZE*4),
   max_output("y", y, DATA_SIZE*4),
   max_runfor("Kernel", DATA_SIZE));
```

Manager (.java)

```
Manager m = new Manager();
Kernel k =
    new MovingAverageKernel();

m.setKernel(k);
m.setIO(
   link("x", PCIE),
   link("y", PCIE));
m.build();
```

MovingAverageKernel (.java)

```
HWVar x = io.input("x", hwFloat(8, 24));
HWVar prev = stream.offset(x, -1);
HWVar next = stream.offset(x, 1);

HWVar sum = prev+x+next;
HWVar result = sum/3;

io.output("y", result, hwFloat(8, 24));
```

Source: O. Pell, Tools & Techniques for Accelerating HPC Applications, SEG Workshop 2011

Embedded Systems & Applications

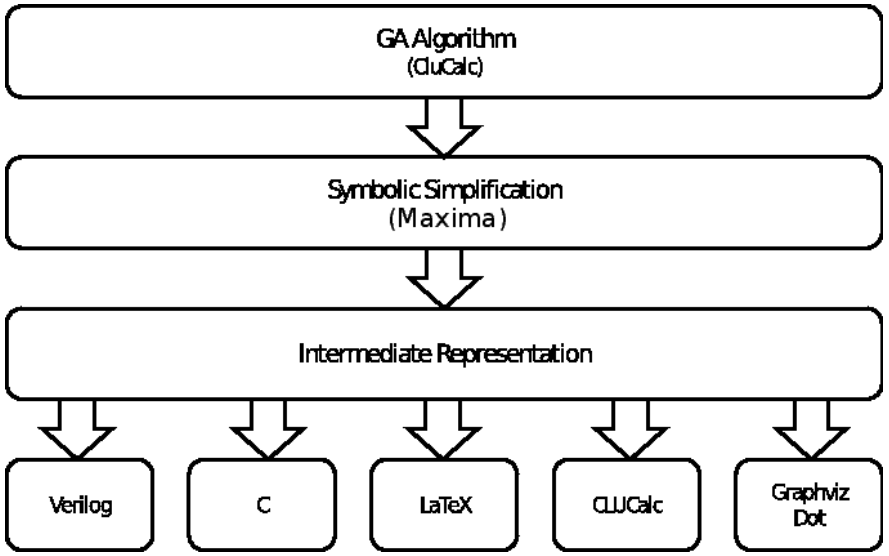
TECHNISCHE UNIVERSITÄT DARMSTADT

Compile Geometric Algebra described in CLUcalc

```
// Color Edge Detection
// {r,g,b}{1,..9} are RGB input arrays

// convert RGB into GA bivector
c1 = r1*e2^e3 + g1*e3^e1 + b1*e1^e2;
c2 = r2*e2^e3 + g2*e3^e1 + b2*e1^e2;
c3 = r3*e2^e3 + g3*e3^e1 + b3*e1^e2;
c7 = r7*e2^e3 + g7*e3^e1 + b7*e1^e2;
c8 = r8*e2^e3 + g8*e3^e1 + b8*e1^e2;
c9 = r9*e2^e3 + g9*e3^e1 + b9*e1^e2;

// rotor-based edge detection computation
s = 1/sqrt(6);
n = (e2 * e3 + e3*e1 + e1*e2)/sqrt(3);
R = s * ((sqrt(2)/2) + n * (sqrt(2)/2));
?p = ~R*(c1+c2+c3)*R +  R*(c7+c8+c9)*~R;
```



- **Gaalop** developed by D. Hildenbrand and FG ESA
- Very abstract domain-specific language (DSL)
- Can target CPU, GPU, and FPGA

Embedded Systems & Applications

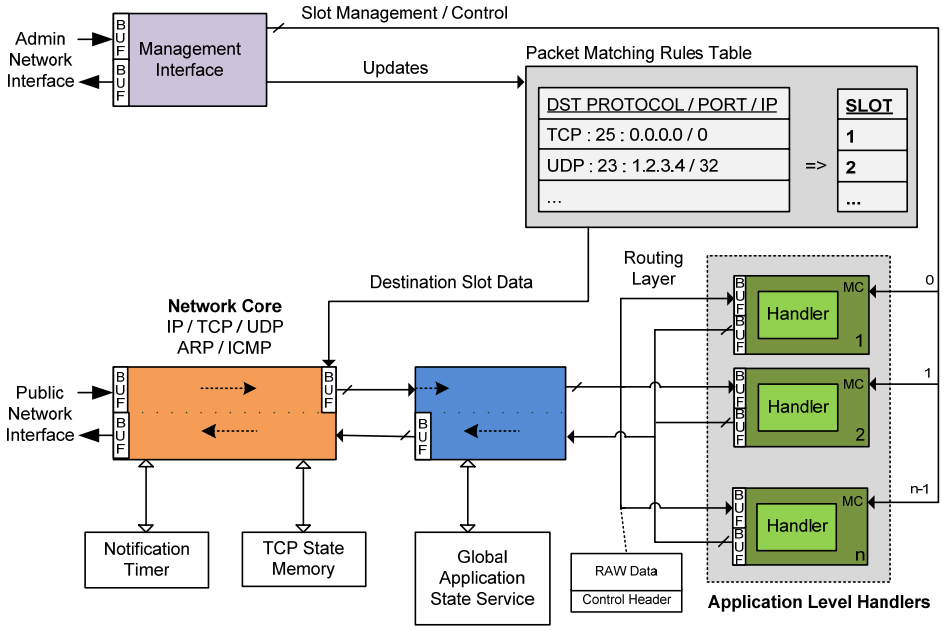
Malacoda for NetStage

DSL for specifying FPGA-accelerated honeypots

TECHNISCHE UNIVERSITÄT DARMSTADT

```
// Emulate login to a root shell
TELNET {
  // define variables
  dynamic username[14];
  // main fsm
  dialogue {
    // default and initial state for a new
        connection
    DEFAULT:
      addresponse("login:");
      $state = LOGIN;
    // next state
    LOGIN:
      // extract user name
      $username = chomp($INPKG);
      addresponse("password:");
      log("TELNET: Login attempt detected");
    ...
    SHELL:
      // emulate Unix uname command
      if ($INPKG =~ /^uname -a/) {
        // send the system identification
        addresponse("Linux myhost 2.6.35.6 ...");
        addresponse("\n");
        addresponse("[localhost]# ");
      }
    ...
  }
}
```

NetStage reconfigurable network processing platform



- Developed by S. Mühlbach (FG ESA)
- High-Throughput (20 Gb/s), low latency (ICMP Echo reply in 700ns)
- Adapts hardware to current network traffic patterns

Embedded Systems & Applications

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

- Motivation
- Selected Application Successes
- Reconfigurable Device Architecture
- Application-Specific Computing Structures
- System Architecture
- Programming Issues
- **Conclusion**

Embedded Systems & Applications

Conclusion

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Motivated** the use of reconfigurable computing (RC)
  - But also shown obstacles: mainly immature programming support
- RC is **NOT** for all applications
  - … many-core CPUs and GPUs are still fine for general-purpose use
- RC has proven successful in **specialized application** areas
  - Bioinformatics
  - Graph processing
  - Embedded computing (high performance, low power)
  - Your application domain? (talk to us ➔ koch@ieee.org)
- Currently **unsurpassed efficiency** for programmable devices
  - Performance per Watt