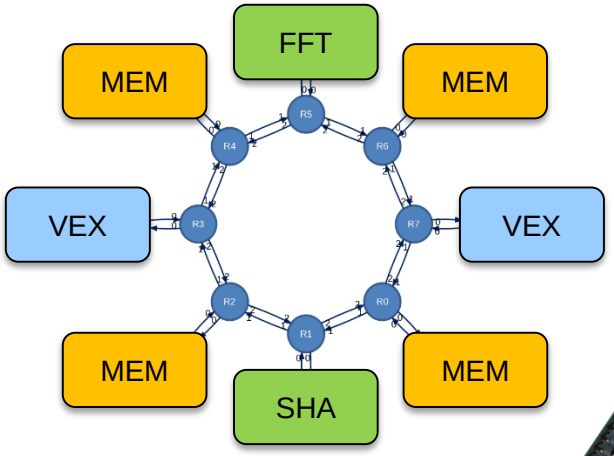
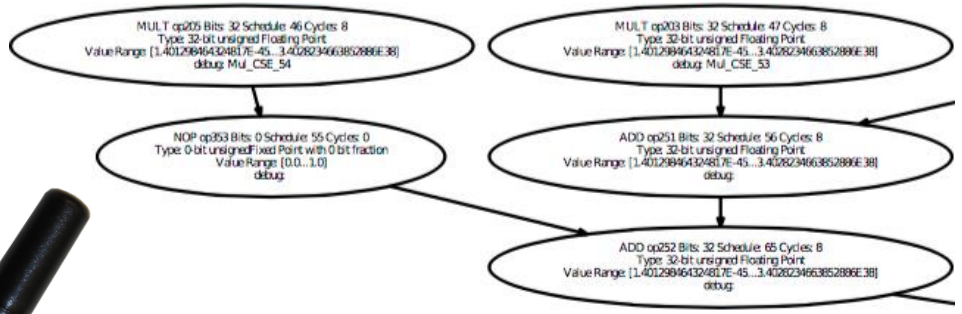
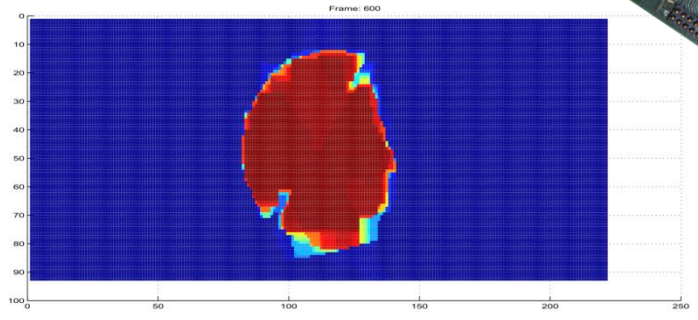
Reconfigurable Computing: Architectures, Tools and Applications

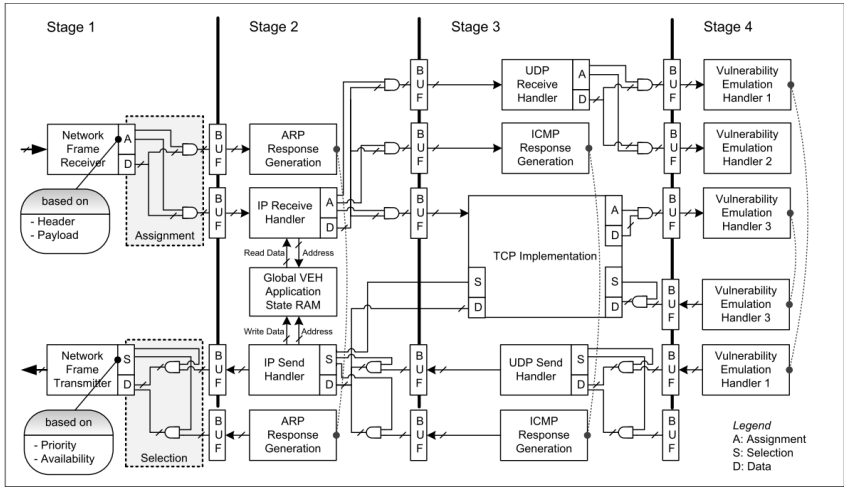
Andreas Koch
Embedded Systems and Applications Group (ESA)
koch@esa.tu-darmstadt.de

TECHNISCHE UNIVERSITÄT DARMSTADT









Embedded Systems and Applications Group

TECHNISCHE UNIVERSITÄT DARMSTADT

- Research group in CS department working in
  - Computer engineering
  - Embedded systems
  - Programming and design tools
- Currently nine researchers
  - Mostly graduate students
- Research funded by
  - State of Hesse: LOEWE Initiative
  - Federal: DFG, BMBF
  - EU
  - Direct industry contracts

Key Research Themes

TECHNISCHE UNIVERSITÄT DARMSTADT

- Design and implement application-specific computer architectures
- Provide programming tools
  - Make custom hardware accessible to non-hardware designers
- Aim for improvements in
  - Performance
  - Energy efficiency
  - Productivity
  - Security
- The following slides present some samples of recent research topics
  - For brevity, this list has to remain incomplete

Relevance of Reconfigurable Computing

TECHNISCHE UNIVERSITÄT DARMSTADT

- Reconfigurable FPGA-based computing has become a very hot topic!
  - Not just embedded systems any more, now also in the data center!
- Intel acquisition of Altera (major FPGA manufacturer) for USD 16.7B
  - Future products: FPGAs combined with Xeon processors
  - Already two+ generations of pilot systems (HARP v1 and v2)
- Microsoft deploys FPGA-accelerators at scale in all new datacenters
  - Two generations of accelerators: Catapult v1 and v2 in Bing and Azure
- Amazon offers FPGA-accelerated-computing-as-a-Service
  - Pilot deployment of “F1” instances in EC2 service
- More: IBM SuperVessel cloud, large scale use of FPGAs by Baidu, …

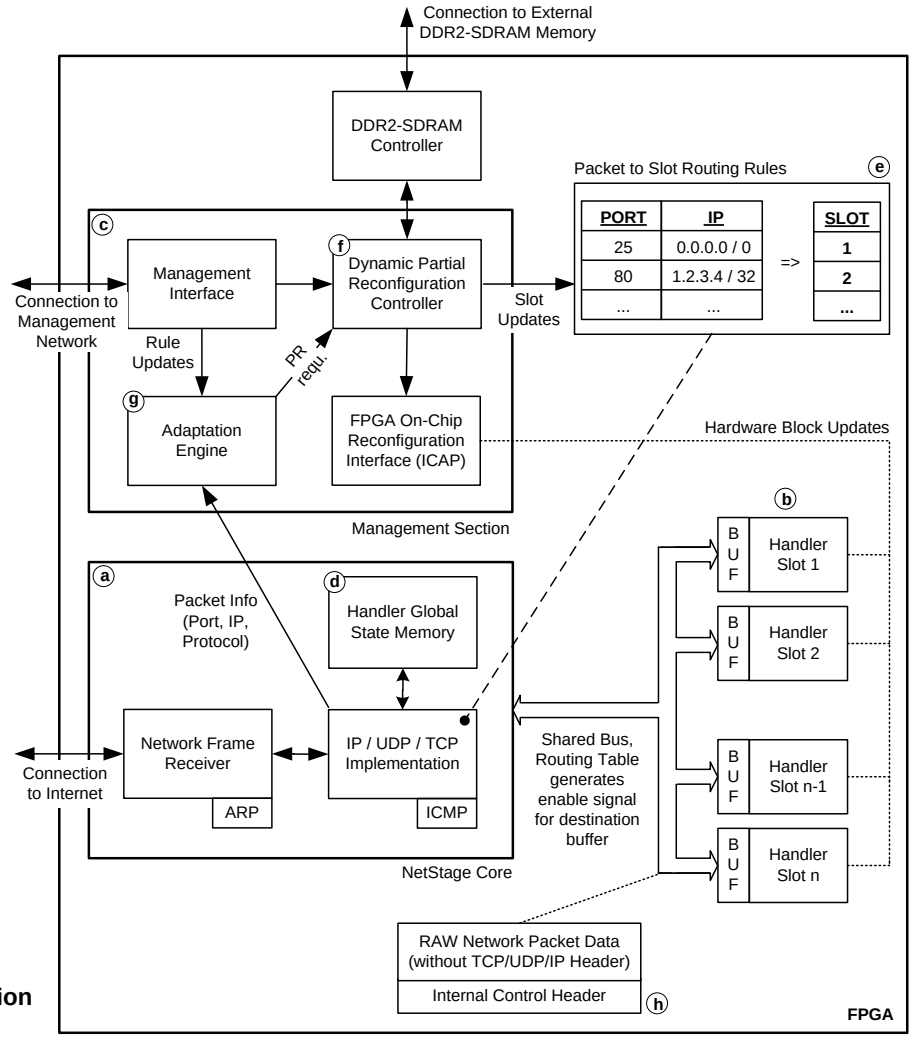
TECHNISCHE UNIVERSITÄT DARMSTADT

DOMAIN-SPECIFIC HARDWARE ARCHITECTURES

Scalable Secure Network Processing
NetStage

TECHNISCHE UNIVERSITÄT DARMSTADT

- Complete network processing stack on FPGA
  - All operations in hardware
  - No compromisable software involved
- Features
  - 10G Ethernet
  - 20 Gb/s throughput, 270ns latency
  - ICMP, ARP, UDP, TCP/IP in hardware
  - Application-layer handlers in hardware
- Dynamically adapts to network traffic
  - Up to 16,500 adaptations per second
  - Supported by on-chip
    - Partial reconfiguration scheduler
    - Per-connection memory management

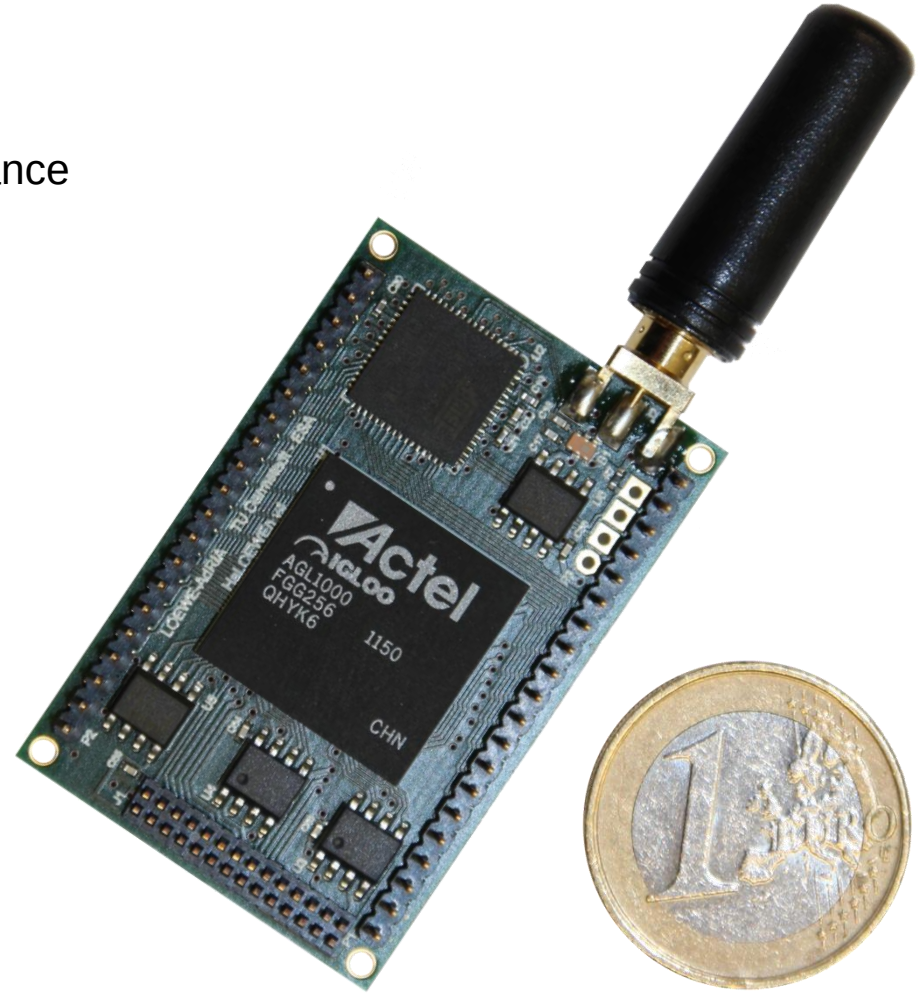


Mühlbach, S., Koch, A.
A Dynamically Reconfigured Network Platform for High-Speed Malware Collection
International Journal of Reconfigurable Computing, Hindawi Publishing, 01-2012

Hardware-Accelerated Wireless Sensor Node
HaLOEWEn

TECHNISCHE UNIVERSITÄT DARMSTADT

- Low-power heterogeneous computing
  - Supplied by energy harvesting
  - Provides significant local compute performance
- Combines
  - Low-power FPGA
  - Low-power Microcontroller
  - Aggressive power management techniques
- Allows trade-off between
  - Data transmission for central processing
  - Local pre-processing using dedicated hardware on FPGA



Andreas Engel, Andreas Koch
Heterogeneous Wireless Sensor Nodes That Target the Internet of Things
IEEE Micro Magazine, Special Issue on Internet of Things, vol. 36, no. 6, 11-2016

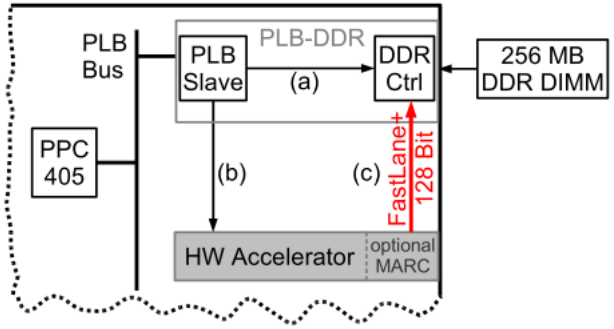
TECHNISCHE UNIVERSITÄT DARMSTADT

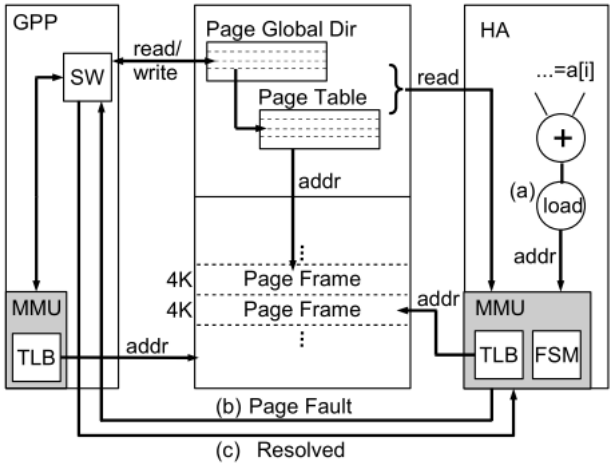
MICROARCHITECTURE TOPICS

Shared-Memory Heterogeneous Computing

TECHNISCHE UNIVERSITÄT DARMSTADT

- Adaptive Computing System (ACS) combines
  - Software-programmable CPU
  - FPGA-based processing element
- Low-latency CPU↔FPGA communication
- High-bandwidth shared memory
  - FPGA interfaces directly to central memory controller
    - Avoids bus overhead for PLB, PCI, PCIe, etc. buses
  - 89% of raw memory bandwidth available to FPGA
    - While full-scale Linux is executing on CPU
- Shared virtual address space
  - Pointers compatible between CPU and FPGA
  - Acceleration of irregular algorithms, e.g., graph traversal
  - Currently being ported to PCIe Gen3 for HSA



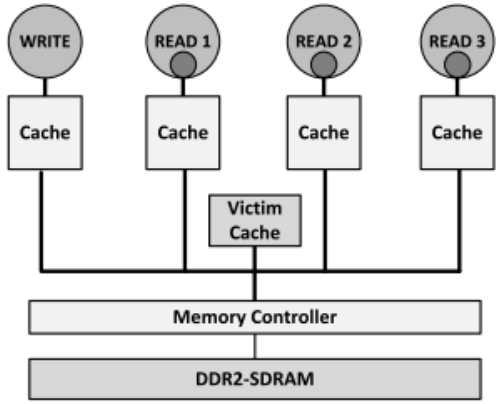


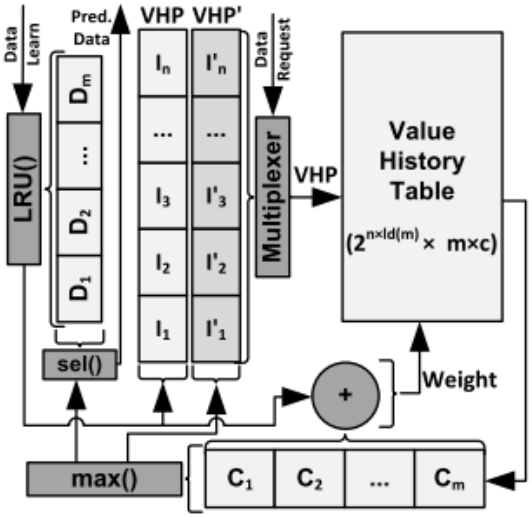
Lange, H., Koch, A.
Architectures and Execution Models for Hardware/Software Compilation and their System-Level Realization
IEEE Transactions on Computers pp. 1363-1377, IEEE Computer Society Digital Library, 10-2010

Application-specific Memory Systems

TECHNISCHE UNIVERSITÄT DARMSTADT

- Match memory system to precise needs of applications
- Configurable features include
  - Abstraction of heterogeneous physical memories
  - Number and nature of parallel access ports
  - Caching ports: architecture and organization
  - Streaming ports: pre-fetching strategies
  - Finely granular inter-port coherency
  - Support for speculative execution
    - Dynamic access prioritization for control speculation
    - Load-value prediction for data speculation
- Abstract interface allows application portability between different hardware platforms



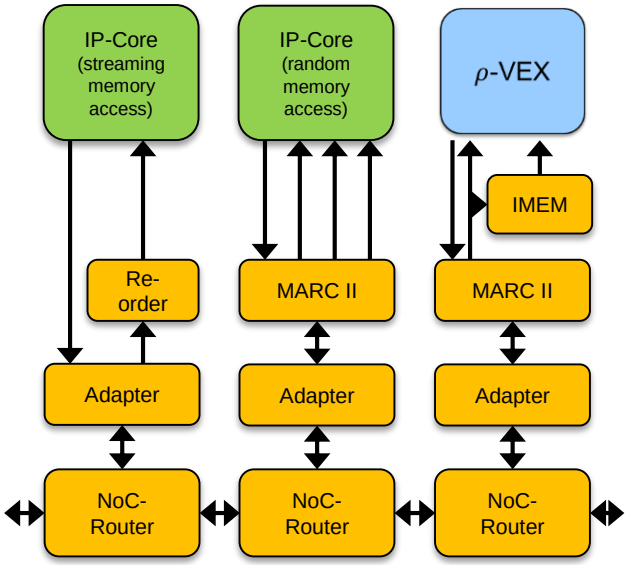


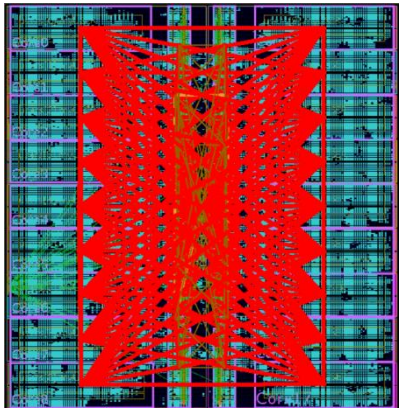
Thielmann, B., Huthmann, J., Koch, A.
Memory Latency Hiding by Load Value Speculation for Reconfigurable Computers
ACM Transactions on Reconfigurable Technology and Systems, Vol. 5, No. 3, Article 13, 10-2012

Heterogeneous Configurable Shared-Memory Many-Core Architecture for Prototyping

TECHNISCHE UNIVERSITÄT DARMSTADT

- Extensibility at multiple hierarchy levels in the architecture
  - ISA extension of individual processors
  - Integration of stand-alone coarse-grained DMA-capable accelerators
- Flow-agnostic insertion of accelerators
  - Handcrafted IP blocks or …
  - … automatically compiled from C
- Current prototype
  - Up to 18x 32b 4-way VLIW cores @ 150 MHz
    - Expected to scale to 72 cores in entire system
  - FFT and SHA blocks as accelerators
  - Communication via Network-on-Chip



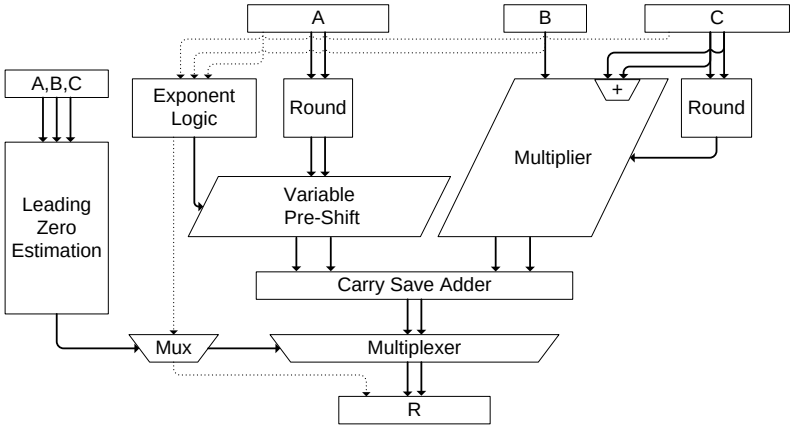


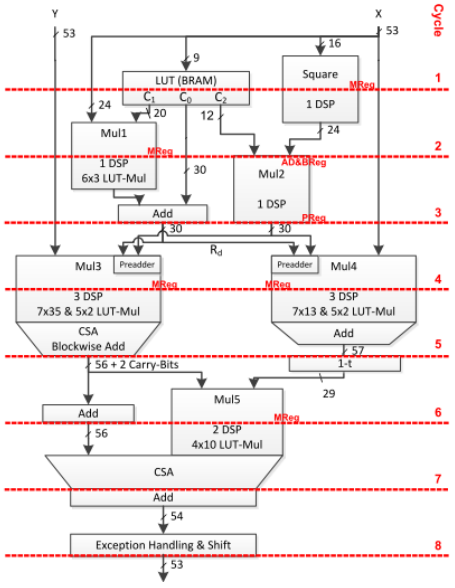
Wink, T., Koch, A.
PHAT: A Technology for Prototyping Parallel Heterogeneous Accelerators
IEEE Proc. Conference on Design & Architectures for Signal & Image Processing (DASIP), 10-2014

High-Performance Floating-Point Operators

TECHNISCHE UNIVERSITÄT DARMSTADT

- Optimize floating point operators for
  - Critical path of the application
  - Precision requirements of the application
- Fast Fused-Multiply-Add unit @ 200 MHz
  - Avoids internal FP normalization
  - 2.5x performance of closest competitor (3 cycles)
- Fast low-latency divider @ 200 MHz
  - Uses faithful rounding (1-ULP)
  - 1.4x performance of closest competitor (8 cycles)





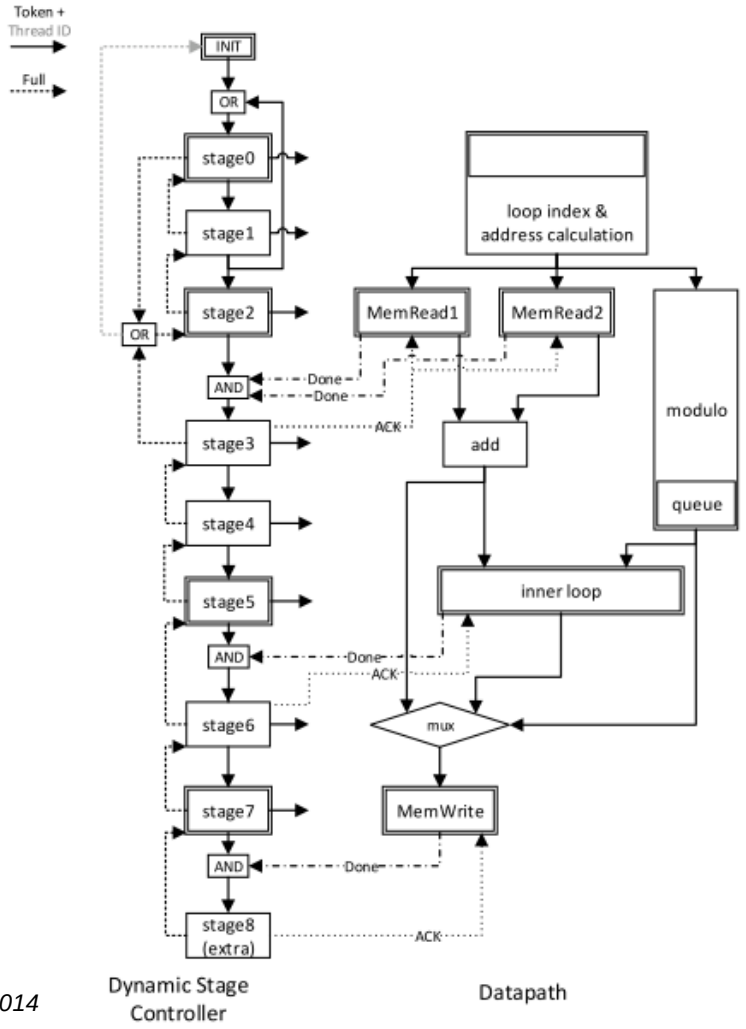
Liebig, B., Huthmann, J., Koch, A.
Architecture Exploration of High-Performance Floating-Point Fused Multiply-Add
Reconfigurable Architectures Workshop (RAW), Boston (USA), 05-2013.

Liebig, B., Koch, A.
Low-Latency Double-Precision Floating Point Division for FPGAs
IEEE Proc. Intl. Conf. on Field-Programmable Technology (FPT), Shanghai (CN)*, 12-2014*

Multi-Threaded Hardware Accelerators

TECHNISCHE UNIVERSITÄT DARMSTADT

- Selectively extend statically scheduled datapath for multithreaded execution
- Employs SMT model
  - Hides memory latencies
  - Allows thread reordering
- Threads share actual compute operators
  - Limits per-thread replication of hardware
- Example circuits with four threads
  - MIPS:3.84x throughput @ 1.53x area
  - SHA: 3.33x throughput @ 1.58x area



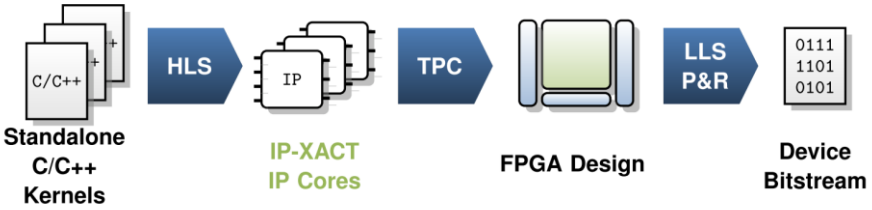
Huthmann, J., Oppermann, J., Koch, A.
Automatic High-Level Synthesis of Multi-Threaded Hardware Accelerators
IEEE Proc. Intl. Conf. on Field Programmable Logic and Applications (FPL), Munich (DE), 09-2014

Task Parallel System Composer (TaPaSCo)

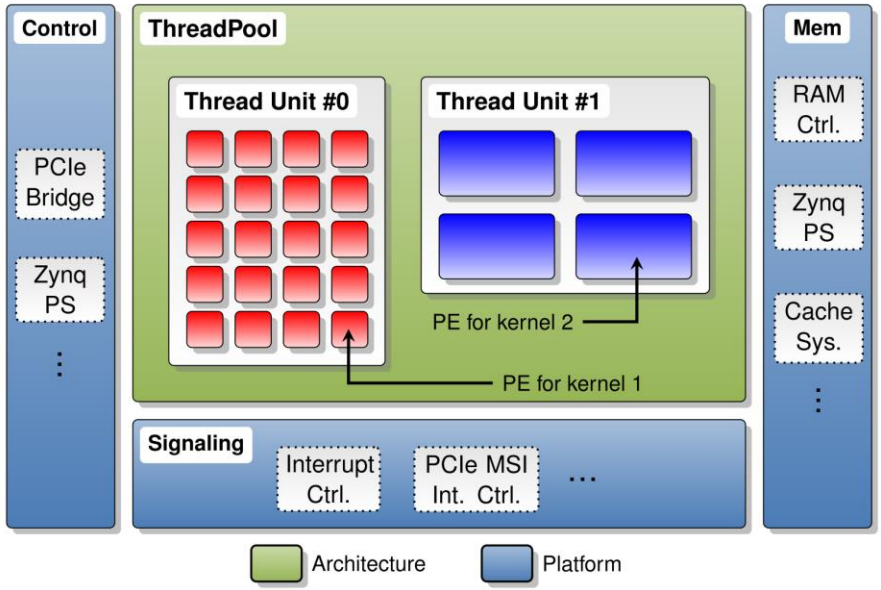
Generating Heterogeneous SoC Architectures

TECHNISCHE UNIVERSITÄT DARMSTADT

- Stitches together IP cores into SoC
- HLS-generated or custom HDL
- Customizes memory hierarchies



- Similar to Xilinx SDSoC
- But TaPaSCo supports more platforms
  - Zynq rSoC: zedboard, ZC706, …
  - PCI Express Gen3: VC709, …
  - UltraScale+: ZCU102, VCU118
- Transparent scalability
  - Automatic job distribution to PEs
  - No source code changes required



Korinth, J., de la Chevallerie, D., Koch, A.
An Open-Source Tool Flow for the Composition of Reconfigurable Hardware Threadpool Architectures
IEEE Proc. Intl. Symp. on Field-Programmable Custom Computing Machines, Vancouver (CA), 05/2015

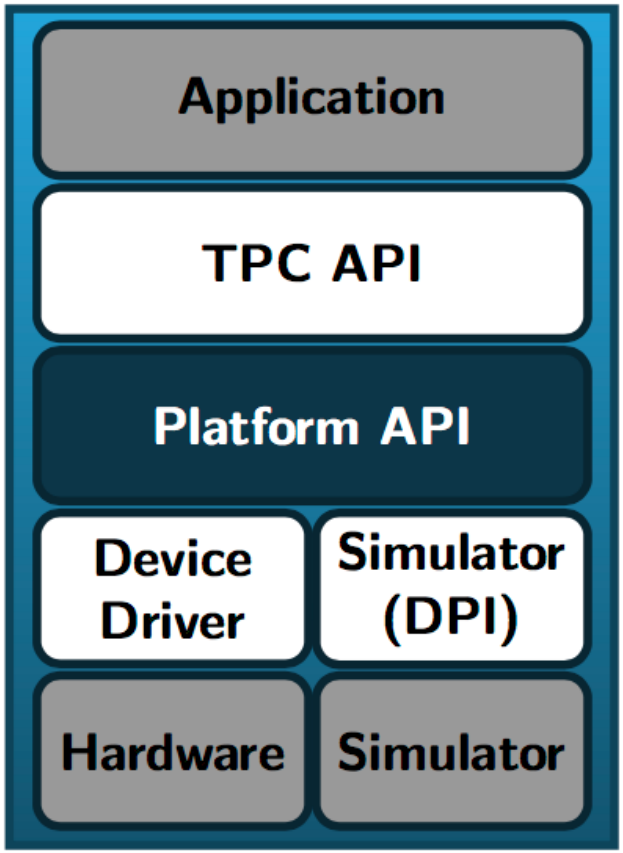
Task Parallel System Composer (TaPaSCo)

Multithreaded Software Access to Hardware Threadpools

TECHNISCHE UNIVERSITÄT DARMSTADT

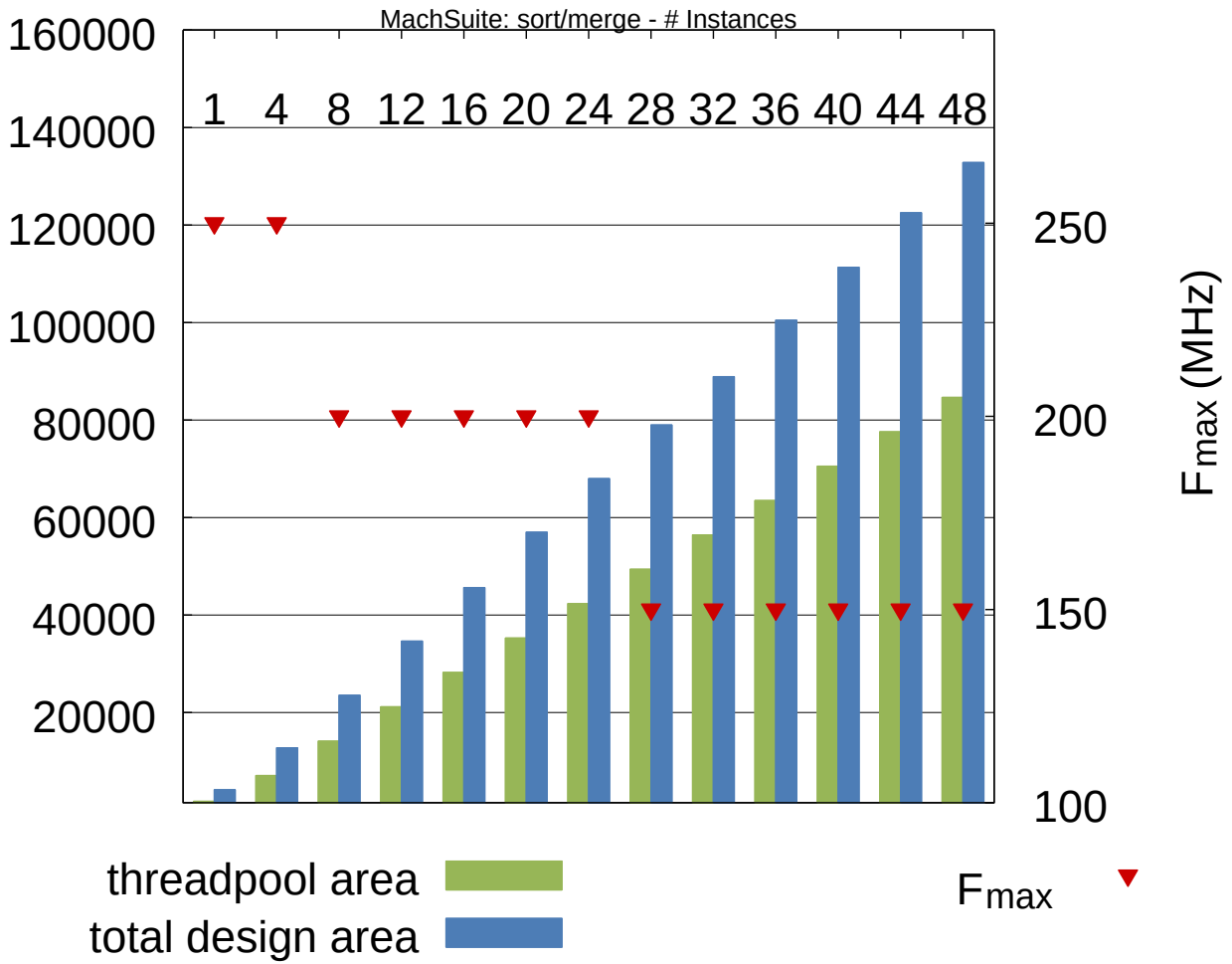
```
   /* allocate 1 KB of device-local memory */
tpc_handle_t h = tpc_device_alloc(dev, 1024);

    /* copy array 'data' to device */
tpc_device_copy_to(dev, data, h, 1024, TPC_BLOCKING_MODE);

    /* prepare a new job for function id #10 */
tpc_job_id_t j_id = tpc_device_acquire_job_id(dev, 10);

    /* set argument #0 to handle h */
tpc_device_job_set_arg(dev, j_id, 0, sizeof(h), &h);

    /* launch job */
tpc_device_job_launch(dev, j_id, TPC_BLOCKING_MODE);

    /* call blocks, so wait for completion and retrieve return value */
int r = 0;
tpc_device_job_get_return(dev, j_id, sizeof(r), &r);
printf("result␣of␣job:␣%d\n", r);

    /* release job id */
tpc_device_release_job_id(dev, j_id);
```

User/OS-level software stack



Task Parallel System Composer (TaPaSCo)
Scalability of Generated Heterogeneous SoCs on ZC706

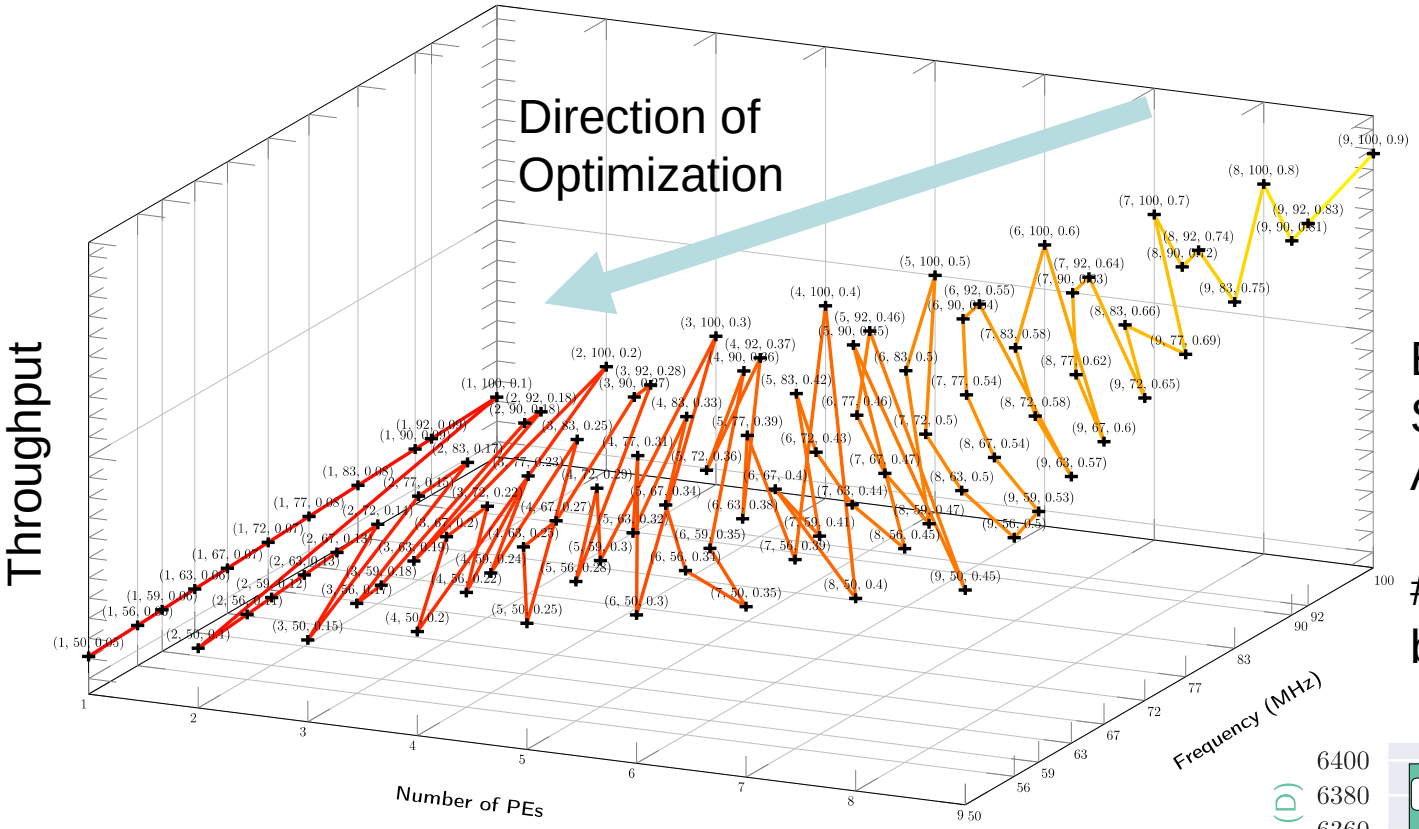
TECHNISCHE UNIVERSITÄT DARMSTADT



Task Parallel System Composer (TaPaSCo)

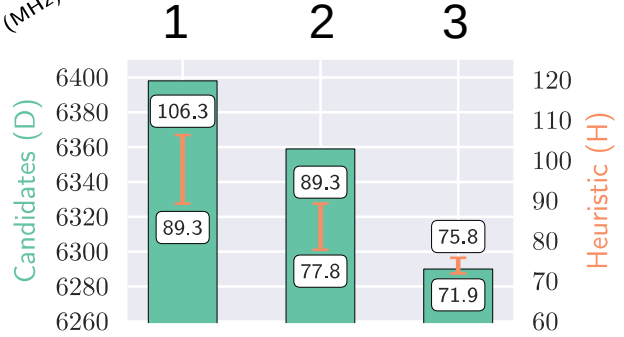
Automatic Design Space Exploration for Maximal Throughput

TECHNISCHE UNIVERSITÄT DARMSTADT



Example: Optimize Stereovision Accelerator

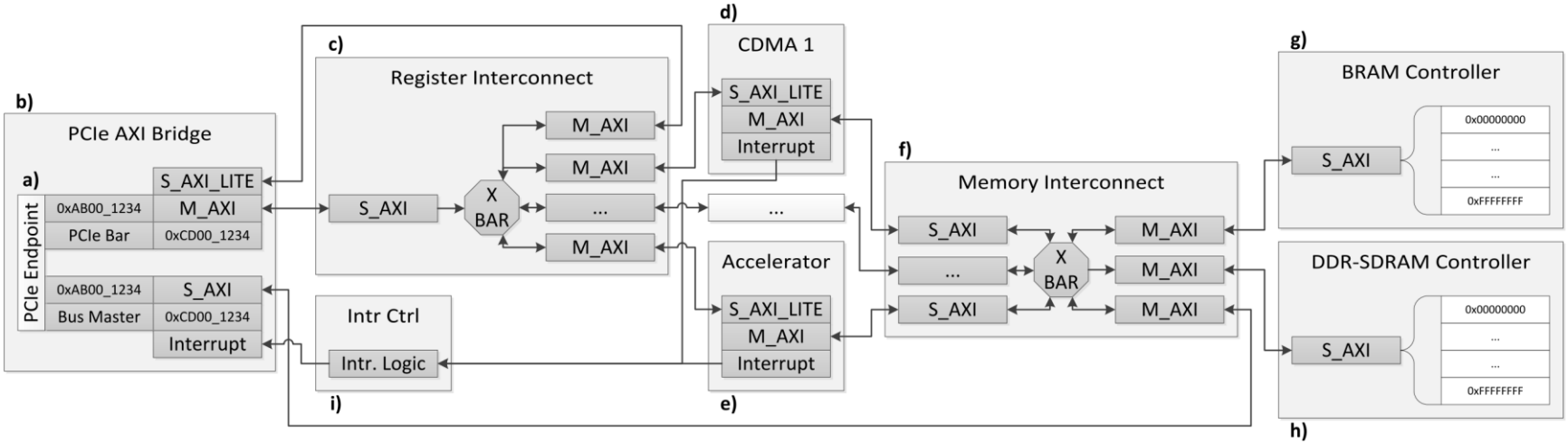
#Candidates before iteration



Open-source, available as public project at https://git.esa.informatik.tu-darmstadt.de

ffLink High-Performance PCIe Gen3 Framework

TECHNISCHE UNIVERSITÄT DARMSTADT



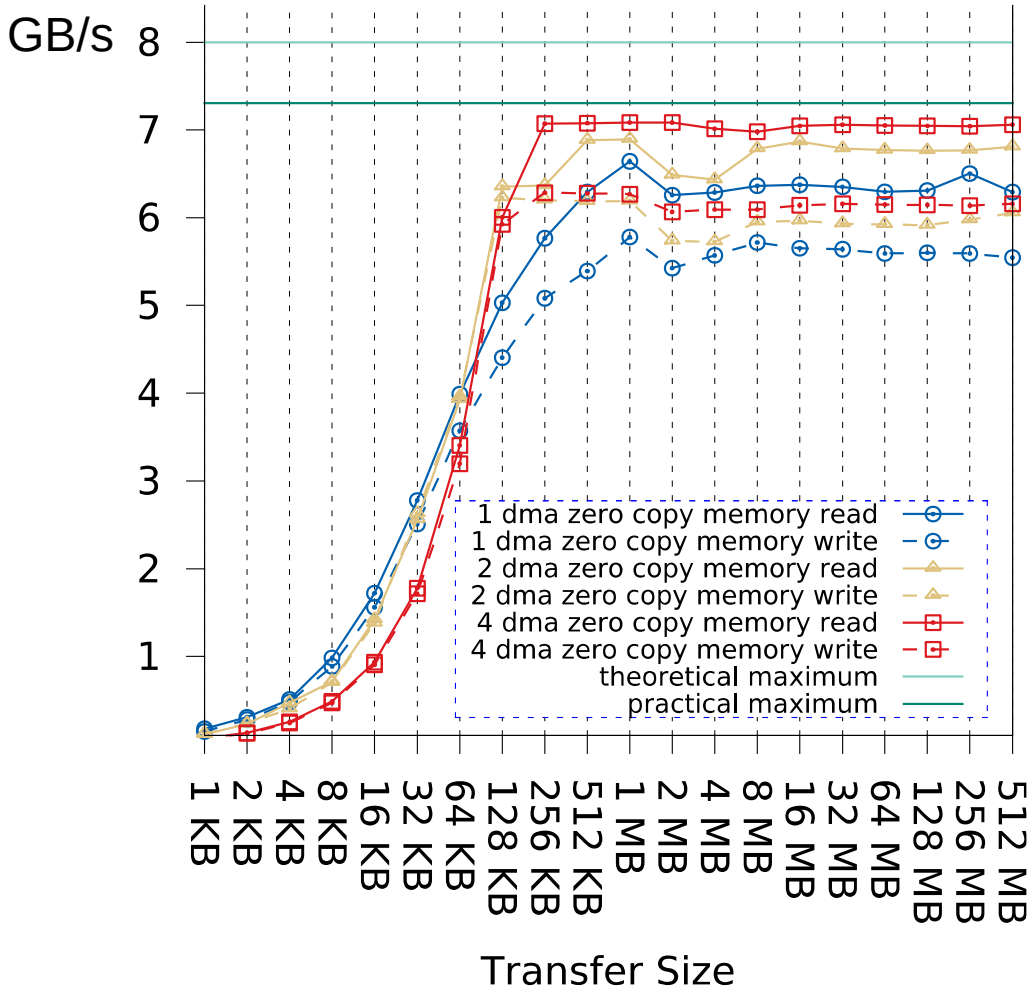
- Custom on-chip framework for high-performance accelerators
  - Supports random memory accesses, not just streaming
  - Included OS-level support: custom device driver, memory allocator etc.
- Example: Used to wrap Threadpools on the Xilinx VC709 PCIe board

de la Chevallerie, D., Korinth, J., Koch, A.
ffLink: A Lightweight High-Performance Open-Source PCI Express Gen3 Interface for Reconfigurable Accelerators
ACM Proc. Intl. Symp. on Highly Efficient Accelertors and Reconfigurable Technologies (HEART), Boston (MA, USA), 06/2015

ffLink Measured Performance on PCIe Gen3 x8

Achieved Throughput Close to Theoretical Limit

TECHNISCHE UNIVERSITÄT DARMSTADT



TECHNISCHE UNIVERSITÄT DARMSTADT

HIGH-LEVEL PROGRAMMING TOOLS

Accelerator Synthesis from Malacoda

Perl-like Domain-specific Language for Network Security

TECHNISCHE UNIVERSITÄT DARMSTADT

- NetStage attractive for honeypot
  - Attack-resilient, high-performance
- Challenge
  - Dynamic security landscape
    - Requires frequent &rapid deployment of new vulnerability emulations
  - But network security experts not familiar with hardware-description languages
- Solution: Malacoda
  - Automatically compiled into hardware vulnerability emulation handlers
  - Inserted into NetStage architecture

```
// 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]# ");
      }
    ...
  }
}
```


Sascha Mühlbach, Andreas Koch
A Reconfigurable Platform and Programming Tools for High-Level Network Applications Demonstrated as a Hardware Honeypot
IEEE Journal on Selected Areas in Communications, 32(10): 1919-1932 (2014)

Accelerator Synthesis from CVXGEN

DSL for Convex Solvers

TECHNISCHE UNIVERSITÄT DARMSTADT

- Convex optimization in widespread use
  - E.g., model-predictive controllers
- Highly compute intensive
  - Solver codes often exceed capabilities of embedded processors
  - Especially under real-time constraints
- Idea: Describe problem in CVXGEN
  - Compile CVXGEN to custom solver accelerator for each optimization problem
- Initial result for Trajectory Planning
  - **4…5x** faster than Cortex-A9 @ 800 MHz
  - Area: ~27…45% of Zynq-7045

```
dimensions
  m = 2
  n = 5
  T = 10
end

parameters
  A (n,n)  # dynamics matrix.
  B (n,m)  # transfer matrix.
  Q (n,n) psd  # state cost.
  Q_final (n,n) psd  # final state cost.
  R (m,m) psd  # input cost.
  x[0] (n)  # initial state.
  u_max nonnegative  # amplitude limit.
  S nonnegative  # slew rate limit.
end

variables
  x[t] (n), t=1..T+1  # state.
  u[t] (m), t=0..T  # input.
end

minimize
  sum[t=0..T](quad(x[t], Q) + quad(u[t], R))
              + quad(x[T+1], Q_final)
subject to
  x[t+1] == A*x[t] + B*u[t], t=0..T
  abs(u[t]) <= u_max, t=0..T
  norminf(u[t+1] - u[t]) <= S, t=0..T-1 constraint.
end
```

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{t=0}^{T} \left(x_t^T Q x_t + u_t^T R u_t \right) + x_{T+1}^T Q_{\text{final}} x_{T+1} \\ \text{subject to} \quad & x_{t+1} = A x_t + B u_t, \quad t = 0, \ldots, T \\ & |u_t| \leq u_{\max}, \quad t = 0, \ldots, T \\ & |u_{t+1} - u_t|_\infty \leq S, \quad t = 0, \ldots, T-1 \end{aligned}$$

Liebig, B., Koch, A.
High-Level Synthesis of Resource-Shared Microarchitectures from Irregular Complex C-Code , *FPT 2016*

Accelerator Synthesis from CLUscript
DSL for Geometric Algebra

TECHNISCHE UNIVERSITÄT DARMSTADT

- Dataflow with conditional expressions
  - But no branching
  - Irregular vector computations (0 … 32 elements per vector)
- Requires high-level domain-specific optimization
  - Integrates Computer Algebra System into compile flow
  - E.g., to eliminate individual constant elements from vector computations
- Performs automatic floating-to-fixed point conversion
  - Optimized representation at each individual operator
  - Analytical (forward/backward)
  - Heuristical (Monte Carlo Simulation)
- Sample Application: Inverse Kinematics
  - **8x throughput** of Intel 4-core 2.4 GHz CPU
  - But accelerator draws only 7 W of power

```
…

// Step 2
l_se_dual = e0 ^ p_e ^ einf;
l_ew_dual = p_e ^ p_w ^ einf;
l_se = *l_se_dual;
l_ew = *l_ew_dual;
CosHalf = sqrt( (1+c4)/2 );
SinHalf = -sqrt( (1-c4)/2 );'
?q_e = CosHalf + SinHalf * quat_i;

// Step 3
?p_ze = VecN3(0,0,L1);
plane_ellbow = *(e0^p_ze^p_e^einf);
middle_plane = p_ze - p_e;
middle_line = plane_ellbow ^ middle_plane;
?Inner_line = middle_line.middle_line;

// Step 4
?q_12 =  middle_line/Norm_LM;

// Step 5
plane_yz = e1;  // plane perpendicular to e1
plane_yz2 = q_12 * plane_yz * ~q_12;
SinHalf = sqrt( (1+c3)/2 );
CosHalf = sqrt( (1-c3)/2 );
quat_k = e2 ^ e1;
?q_3 = CosHalf + SinHalf*quat_k;
?q_s = q_12 * q_3;
```

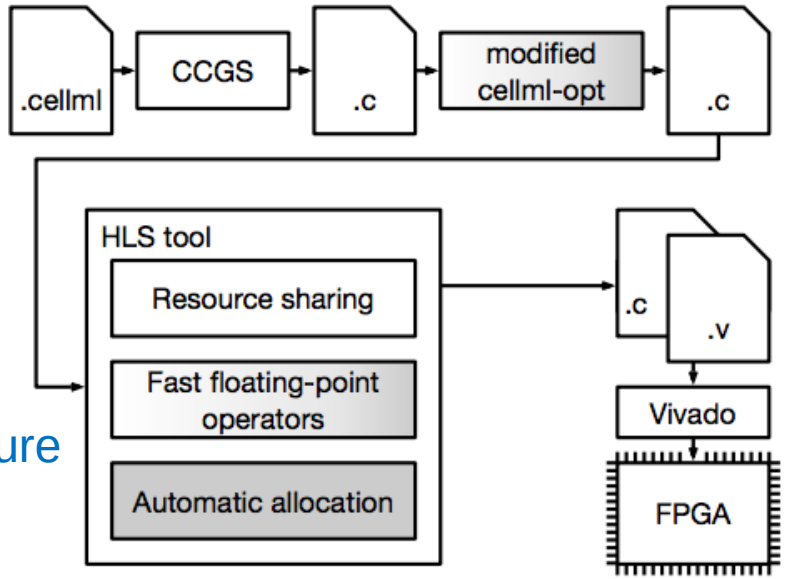
Huthmann, J., Müller, P., Stock, F., Hildenbrand, D., Koch, A.
Accelerating High-Level Engineering Computations by Automatic Compilation of Geometric Algebra to Hardware Accelerators
IEEE Proc. Intl. Conf. on Embedded Computer Systems: Architectures, MOdeling and Simulation (SAMOS), Samos (GR), 07-2010

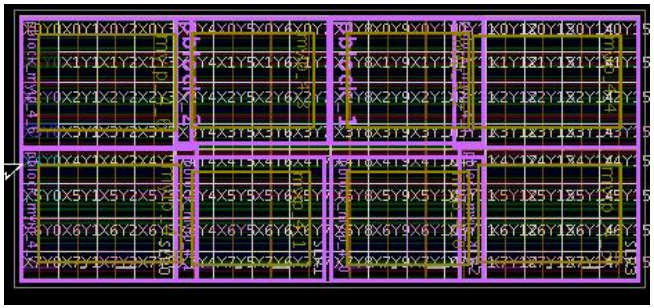
HLS of Simulation Accelerators for Cell Biology

DSL for Biomedical Models at the Cell Level

TECHNISCHE UNIVERSITÄT DARMSTADT

- Input are descriptions in CellML
  - Idiomatic C code is used as IR
- Complex computations
  - > 1,800 DP operators for large models
  - Too large for fully spatial implementation
- HLS of heavily resourced-shared µarchitecture
- Single accelerator (200 MHz on Zynq 7045)
  - **4x speed-up** vs 4.2 GHz i7 core
    - With just **4% energy** required
- Scale-up to multiple cores on UltraScale+
  - One core requires just 2.9% of XCVU13P FPGA
  - 12 cores @ 252 MHz: **80x** throughput of CPU,
    - even **10%** more than Tesla GK210 GPU, at < 1/250th of the GPU‘s latency



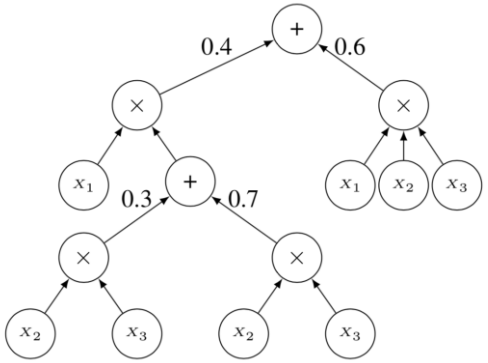


Sample scaling on VUP: 8 cores @ 282 MHz

Machine Learning Acceleration
DSL for Sum-Product Networks (SPNs)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Exact inference for probabilistic queries
- Very compute intensive
- Current implementations map poorly to GPUs
  - Inappropriate parallelization in TensorFlow
- Compiled from DSL into hardware accelerator



- Example: XC7V690T FPGA @ 200 MHz
  - Up to **6x throughput** of AMD Ryzen 1950X CPU
    - Netflix dataset, including PCIe transfer overheads
  - Actual kernel computation up to **50x faster**
    - MSNBC300 dataset

```
SumNode_0 SumNode(0.49*ProductNode_1, 0.51*ProductNode_7){
  ProductNode_1 ProductNode(PoissonNode_2, PoissonNode_3,
                            PoissonNode_4, PoissonNode_5,
                            PoissonNode_6){
                    PoissonNode_2 P(pca|λ=0.386)
                    PoissonNode_3 P(building|λ=0.204)
                    PoissonNode_4 P(training|λ=12.925)
                    PoissonNode_5 P(part|λ=1.257)
                    PoissonNode_6 P(clause|λ=0.020)
                    }
  ProductNode_7 ProductNode(PoissonNode_8, PoissonNode_9,
                            PoissonNode_10, PoissonNode_11,
                            PoissonNode_12){
                    PoissonNode_8 P(pca|λ=0.211)
                    PoissonNode_9 P(building|λ=0.296)
                    PoissonNode_10 P(training|λ=0.342)
                    PoissonNode_11 P(part|λ=1.141)
                    PoissonNode_12 P(clause|λ=0.123)
                    }
}
```

- Shared-memory system would avoid transfers
  - E.g., Xilinx UltraScale+ MPSoC, Intel HARP2

Sommer L., Oppermann J., Molinas A., Binnig C., Kersting K., Koch A.
Automatic Synthesis of FPGA-based Accelerators for the Sum-Product Network Inference Problem
Paper submitted and currently under review, 03/2018

„C“ Hardware/Software Co-Compilers

TECHNISCHE UNIVERSITÄT DARMSTADT

- Compilation from „C“ for adaptive computers
  - Interacting CPU(s) and FPGA(s)
- Aim
  - Support large relevant subset of C
  - Includes: pointers, stdlib calls, irregular control flow, variable-bounded loops, …
- Multi-year research effort

Too many publications to list here, samples include:

Huthmann, J., Liebig, B., Oppermann, J., Koch, A.
Hardware/software co-compilation with the Nymble system
in Proc. of the Intl. Workshop on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), Darmstadt, 2013

Thielmann, B., Huthmann, J., Koch, A.
Widening the Memory Bottleneck by Automatically-Compiled Application-Specific Speculation Mechanisms
in *Embedded Systems Design with FPGAs* by Athanas, P. et al., Springer, 11-2012

Gädke, H., Stock, F., Koch, A.
Memory Access Parallelization in High-Level Language Compilation for Reconfigurable Adaptive Computers
IEEE Intl. Conf. on Field Programmable Logic and Applications (FPL)
Heidelberg (D), 09-2008.

- COMRADE 1.0/2.0
  - Fully dynamic scheduling for irregular code (based on SUIF)
- Nymble
  - Efficient (mostly) static schedules for more regular code (based on LLVM)
- Nymble-PC
  - Explores data value speculation for memory latency hiding
- Nymble-RS
  - Employs resource-sharing for mapping very complex computations to hardware
- Nymble-MT
  - Automatically generates multi-threaded (OoO SMT) accelerators for memory latency hiding
- Nymble-OMP
  - Automatically generates multi-threaded (in-order SMT) accelerators from OpenMP worksharing loops

Selected Topics on Hardware Compilers

TECHNISCHE UNIVERSITÄT DARMSTADT

- Fast Optimal Modulo Scheduling by ILP

Julian Oppermann, Andreas Koch, Melanie Reuter-Oppermann, Oliver Sinnen
ILP-based Modulo Scheduling for High-level Synthesis
International Conference on Compilers, Architectures and Synthesis For Embedded Systems (CASES), ESWEEK, Pittsburgh, PA (USA), 10-2016

- Analysis of Large Code Bodies for HLS Suitability

Julian Oppermann, Lukas Sommer, Andreas Koch
SpExSim: assessing kernel suitability for C-based high-level hardware synthesis
Journal of Supercomputing, 07-2017

- HLS from OpenMP code
  - Multi-Threaded Accelerators for `par`-Loops

Lukas Sommer, Julian Oppermann, Andreas Koch
C-based Synthesis of Area-Efficient Accelerators for OpenMP Worksharing Loops
Second International Workshop on Heterogeneous High-performance Reconfigurable Computing (H2RC), SC 17, Salt Lake City, UT (USA), 11-2016

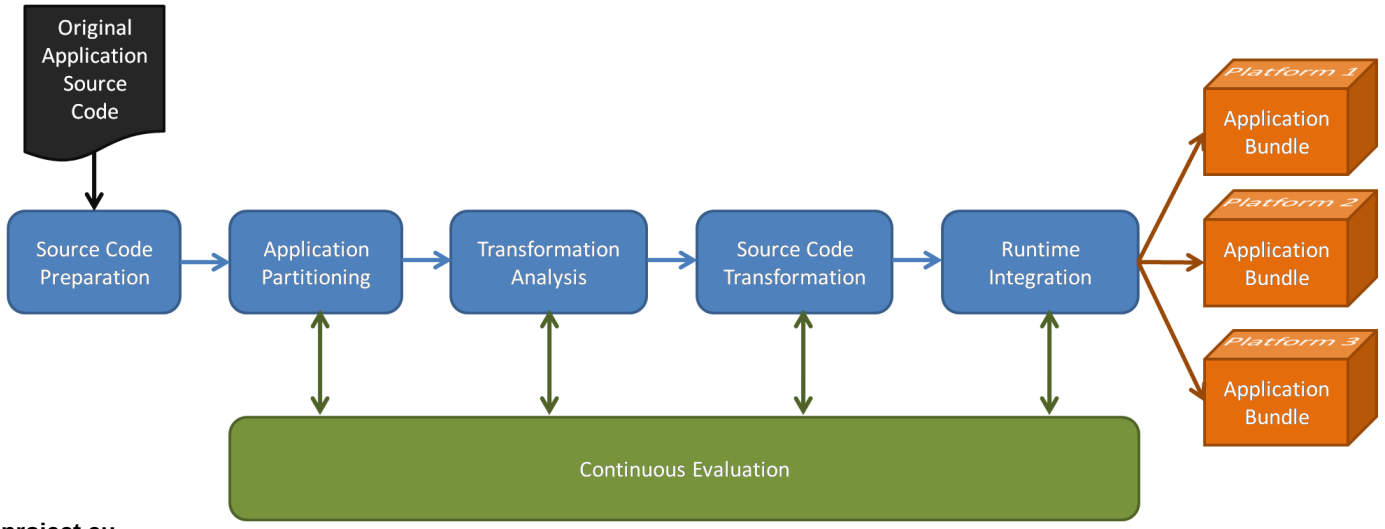
  - Support OpenMP 4.0 `target` Offload Mechanism

Lukas Sommer, Jens Korinth, Andreas Koch
OpenMP Device Offloading to FPGA Accelerators
International Conference on Application-specific Systems, Architectures and Processors (ASAP), Seattle (USA), 07-2017

REPARA Project
„C++“ for Heterogeneous Parallel Systems

TECHNISCHE UNIVERSITÄT DARMSTADT

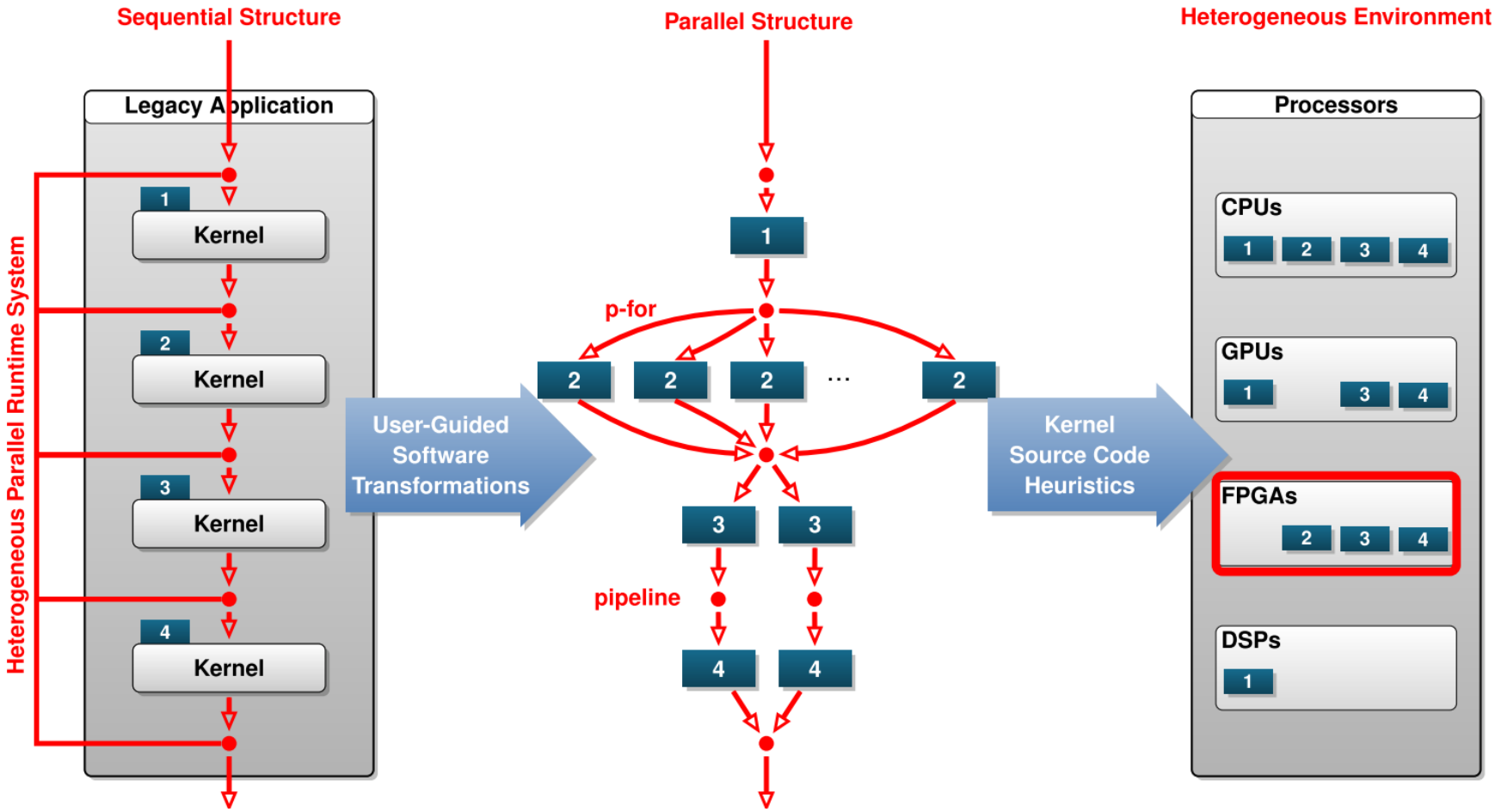
- Target architecture includes CPUs and FPGA(s), GPU(s), DSP(s)
- Use C++1x language features for convenient and efficient programming
- C++ Attributes describe, e.g.,
  - Parallelism skeletons : thread farm, pipeline, etc.
  - Data location and persistence: storage in main, on-board, on-chip memories, …
- ESA provides compilation to FPGA-based processing elements in system



http://www.repara-project.eu

REPARA Project
Programming and Execution Flow

TECHNISCHE UNIVERSITÄT DARMSTADT



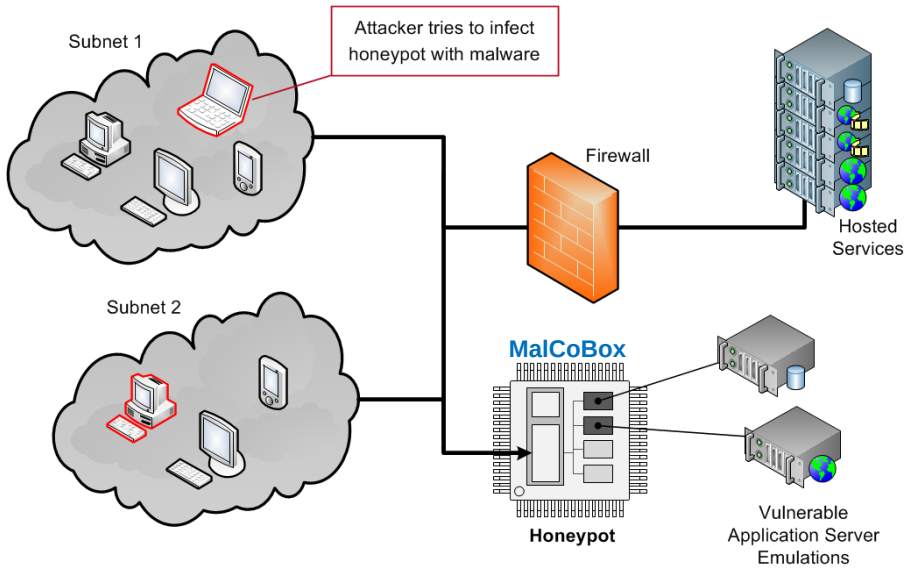
TECHNISCHE UNIVERSITÄT DARMSTADT

SAMPLE APPLICATIONS EXPERIENCE

Attack-Resilient Honeypot MalCoBox
Secure Malware Collection

TECHNISCHE UNIVERSITÄT DARMSTADT

- MalCoBox combines
  - NetStage network processor architecture
  - Malacoda language and compilers
- Directly connected to 10G Internet uplink
- Emulates typical software vulnerabilties
  - WebMail front-end
  - Telnet
  - Mail server
  - MSSQL vulnerable to Slammer
  - SMB login
  - DNS
- Flawless operation in the field

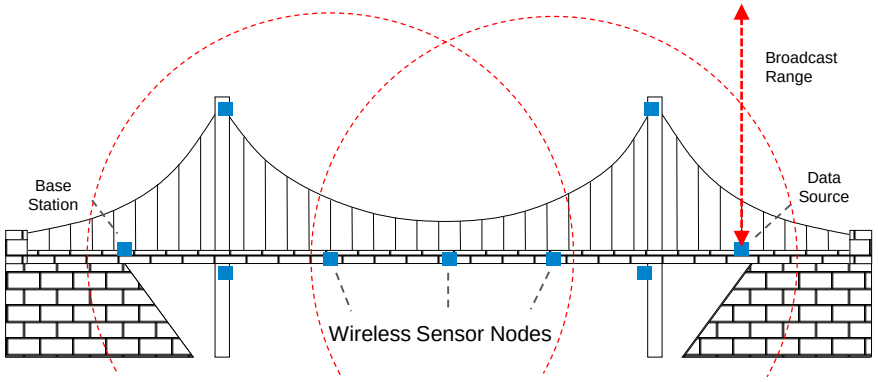


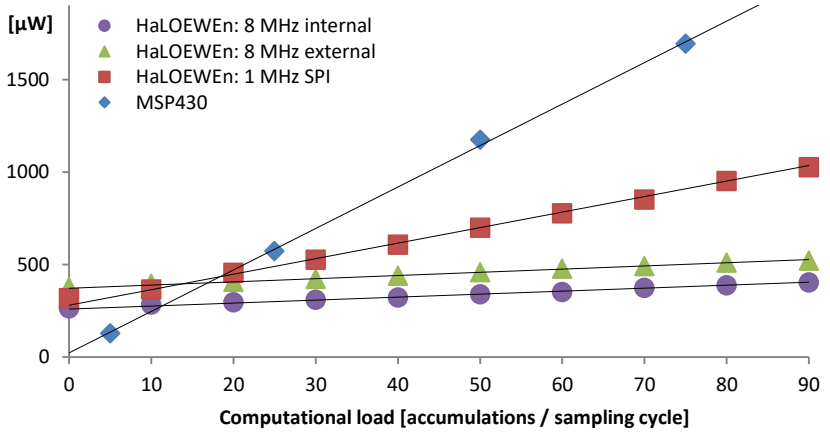
Mühlbach, S., Koch, A.
A Reconfigurable Hardware Platform for Secure and Efficient Malware Collection in Next-Generation High-Speed Networks
International Journal for Information Security Research, Vol. 1, Issue 4, Infonomics Society, 12-2011

Health Monitoring of Large Structures
Wireless Sensor Network

TECHNISCHE UNIVERSITÄT DARMSTADT

- Structural health monitoring (SHM)
  - Using random ambient vibrations as stimulus
  - No controlled excitation required in the field
    - Would be difficult for bridges and wind turbines
- Solvable by parallel distributed algorithm
  - Random Decrement Technique (RDT)
- Challenges
  - Requires significant node-local processing
    - Not achievable using conventional MSP430 MCU
  - Continuous data transmission would overtax energy supply (harvesting)
- Employ ESA HaLOEWEn sensor node
  - Excellent energy efficiency
  - RDT performed in dedicated hardware





A. Engel, A. Friedmann, M. Koch, J. Rohlfing, T. Siebel, D. Mayer, A. Koch
Hardware-Accelerated Wireless Sensor Network for Distributed Structural Health Monitoring
Proc. 2nd Intl. Conf. on System-Integrated Intelligence, Bremen, 2014

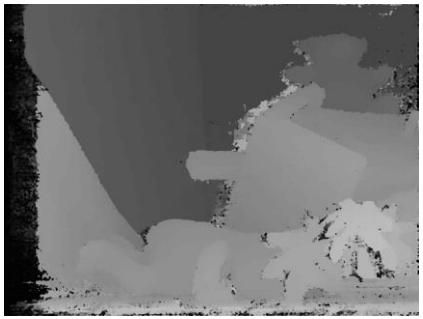
Hardware Accelerator for Stereovision

Compute Real-Time Depth Data using Two Cameras by SGM

TECHNISCHE UNIVERSITÄT DARMSTADT

Implementation	Fps for VGA	Frequency [MHz]
Original CPU [OpenCV w/ SSE]	0.1	2400
Small FPGA (Zynq 7010)	26.6	110
Medium FPGA (Zynq 7045)	197.0	205
Large FPGA (Virtex 7 690)	410.0	131

Left Camera Image Right Camera Image



Disparity Map

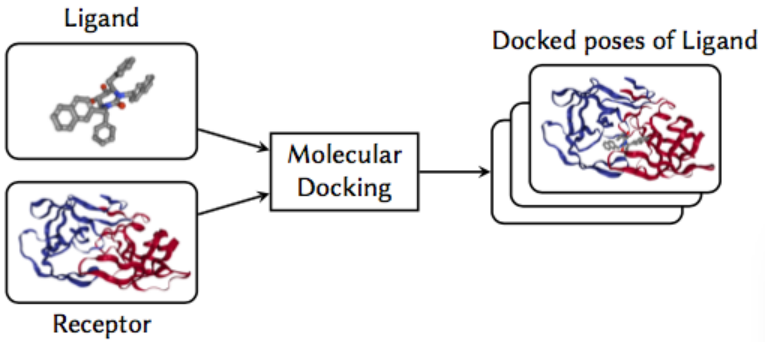
- Also supports 75fps@720p and 28fps@1080p resolutions on Virtex 7

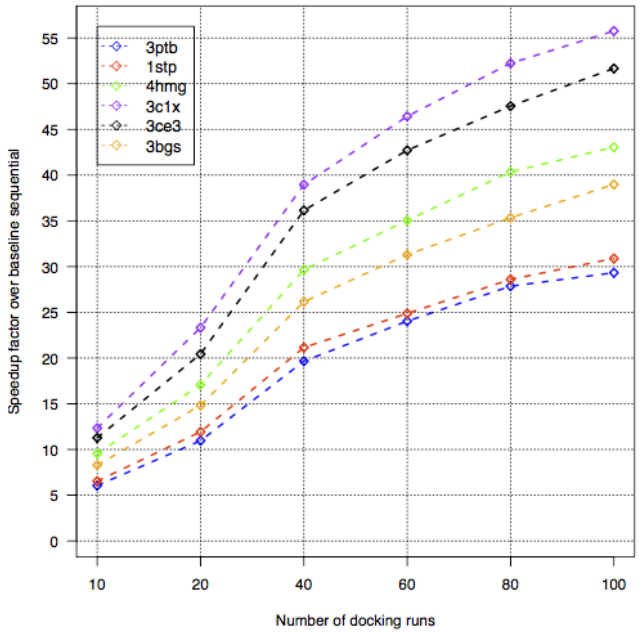
Jaco Hofmann, Jens Korinth, Andreas Koch
A Scalable High-Performance Hardware Architecture for Real-Time Stereo Vision by Semi-Global Matching
IEEE Embedded Vision Workshop at Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas (USA), 06-2016

OpenCL-accelerated Molecular Docking

TECHNISCHE UNIVERSITÄT DARMSTADT

- Based on AutoDock 4.2
- CPU/GPU OpenCL implementation
  - FPGAs are work-in-progress
- Performance vs. original sequential
  - 3.3x on quad-core CPU
  - 40.4x on GPU
- Energy efficiency vs original sequential
  - 1.4x on quad-core CPU
  - 5.4x on GPU
- Collaboration with Scripps Institute (USA)





Leonardo Solis-Vasquez, Andreas Koch
A Performance and Energy Evaluation of OpenCL-accelerated Molecular Docking
Fifth International Workshop on OpenCL (IWOCL), Toronto (Canada), 05-2017

TECHNISCHE UNIVERSITÄT DARMSTADT

FUTURE RESEARCH DIRECTIONS

Directions for Future Research

TECHNISCHE UNIVERSITÄT DARMSTADT

- Apply expertise in domain-specific computing in joint projects
- Sample domains for inter-disciplinary R&D include
  - Networking and security, Bioinformatics, Data analytics, Robotics, Automotive, High-performance (embedded) computing, Machine learning, …
- Exploit parallel heterogeneous computer systems
  - Improve power efficiency and latency
  - Specialized hardware architectures, e.g., for in-network/near-storage computing
- Improve programmability of the specialized computer architectures
  - Tools and architectural support for more advanced models of computation
    - Move up from conventional C/C++ programming to DSLs
    - Use of next-generation HDLs (Bluespec SystemVerilog, UCB Chisel)