VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kansliets noteringar Kod	Dnr
2011-40480-88494-55	

2011
Project Research Grant

Area of science
Natural and Engineering Sciences

Announced grants
Project research grant NT 13 April 2011

Total amount for which applied (kSEK)

2012	2013	2014	2015	2016
2855	2976	1902	2005	1923

APPLICANT

Name(Last name, First name)
McKee, Sally A.

Date of birth
630611-2126

Gender
Female

Email address
mckee@chalmers.se

Academic title
Associate professor

Position
Chalmers Computer Science and Engineering

Phone
+46762623832

Doctoral degree awarded (yyyy-mm-dd)
1995-05-21

WORKING ADDRESS

University/corresponding, Department, Section/Unit, Address, etc.
Chalmers tekniska högskola
Department of Computer Science and Engineering
Computer Engineering
Rännvägen 6, EDIT 4124
41296 Göteborg, Sweden

ADMINISTERING ORGANISATION

Administering Organisation
Chalmers tekniska högskola

DESCRIPTIVE DATA

Project title, Swedish (max 200 char)
FlexSoft: En mjukvaruplattform som möjliggör samverkande mjuk- och hårdvaruutveckling för exponerade arkitekturer

Project title, English (max 200 char)
FlexSoft: Software Infrastructure to Support Hardware/Software Codesign of Exposed Architectures

Abstract (max 1500 char)

Instruction Set Architectures (ISAs) traditionally specify the hardware/software interface. In most modern architectures, implementations of this interface rely on microcoding (to break ISA instructions into smaller micro-operations, or uops), pipelining (to allow simultaneous, overlapped execution of uops), and central register files (to temporarily store operands and results). In contrast, "Exposed Architectures" dramatically change the hardware/software contract, giving the compiler control of all computational resources. Doing so creates much potential for streamlining the hardware, which leads to more performance- and energy-efficient processors. Harnessing this potential places a heavier burden on the compiler: the exponential number of possible uop schedules creates an intractable solution space. This research creates efficient software tools for such Exposed Architectures. Our solution builds on a compiler infrastructure that generates compact uop schedules by employing a SAT solver to efficiently evaluate potential solutions. The compiled applications influence the design-time hardware configuration (we need not provide means to execute all possible uop sequences, but can instead simplify the hardware to implement only those uop combinations found in the target applications). The hardware configuration, in turn, dictates required software functionality, creating a "design-evaluate-refine" cycle that embodies the fundamentals of hardware-software codesign.

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2011-40480-88494-55

Name of Applicant
McKee, Sally A.

Date of birth
630611-2126

Abstract language
English
Keywords

Research areas
Computer Science
Review panel
NT-S
Classification codes (SCB) in order of priority
20206, 20207,
Aspects

Continuation grant
Application concerns: New grant
Registration Number:
Application is also submitted to
similar to: **identical to:**

ANIMAL STUDIES

Animal studies
No animal experiments

OTHER CO-WORKER

Name(Last name, First name)
Larsson-Edefors, Per
University/corresponding, Department, Section/Unit, Addressetc.
Chalmers tekniska högskola
Department of Computer Science and Engineering
Date of birth
670820-5536
Gender
Male
Academic title
Professor
Doctoral degree awarded (yyyy-mm-dd)
1995-03-24

Name(Last name, First name)
,
University/corresponding, Department, Section/Unit, Addressetc.
Date of birth
Gender
Academic title
Doctoral degree awarded (yyyy-mm-dd)

Name(Last name, First name)
,
University/corresponding, Department, Section/Unit, Addressetc.
Date of birth
Gender
Academic title
Doctoral degree awarded (yyyy-mm-dd)

Name(Last name, First name)
,
University/corresponding, Department, Section/Unit, Addressetc.
Date of birth
Gender
Academic title
Doctoral degree awarded (yyyy-mm-dd)

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2011-40480-88494-55

Name of Applicant
McKee, Sally A.

Date of birth
630611-2126

ENCLOSED APPENDICES

A, B, B, C, C, S

APPLIED FUNDING: THIS APPLICATION

Funding period (planned start and end date)
2012-01-01 -- 2016-12-31

Staff/ salaries (kSEK)

Main applicant	**% of full time in the project**	**2012**	**2013**	**2014**	**2015**	**2016**
Sally A. McKee	20	271	280	289	299	309
Other staff						
Per Larsson-Edefors	5		84	87	90	
Postdoc	100	1002	1036			
Doktorand 1	80	581	600	620	642	663
Doktorand 2	80	581	600	620	642	663
Total, salaries (kSEK):		2435	2600	1616	1673	1635

	2012	**2013**	**2014**	**2015**	**2016**
Resor	140	140	140	140	140
Datorutrustning	60			40	
Lokaler	187	200	124	129	126
IT-kostnader	33	36	22	23	22
Total, other costs (kSEK):	420	376	286	332	288

Total amount for which applied (kSEK)

2012	2013	2014	2015	2016
2855	2976	1902	2005	1923

ALL FUNDING

Other VR-projects (granted and applied) by the applicant and co-workers, if applic. (kSEK)

Proj.no.(M) or reg.nr.	**Funded 2011**	**Funded 2012**	**Applied 2012**
2009-4566	1857	1857	

Project title
CHAMPP: CHalmers Adaptable Multicore Processing Project

Applicant
PI Per Stenström

Funds received by the applicant from other funding sources, incl ALF-grant (kSEK)

Funding source	**Total**	**Proj.period**	**Applied 2012**
EC 7th Framework Programme 249059	2300	2010-2012	

Project title
Embedded Reconfigurable Architectures

Applicant
Sally A. McKee (leader: Stephan Wong, TUDelft)

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2011-40480-88494-55

Name of Applicant
McKee, Sally A.

Date of birth
630611-2126

Funding source	Total	Proj.period	Applied 2012
HiPEAC NoE Compilation Cluster	109	2010-2011	

Project title
Hardware Support for Javascript Collaboration Grant

Applicant
PI Christian Probst (DTU)

Funding source	Total	Proj.period	Applied 2012
Ericsson AB	880	2011-2012	

Project title
Smarter Resource Management via Portable, Scalable, On-Line Power Estimation

Applicant
PI Per Strenström

Funding source	Total	Proj.period	Applied 2012
Ericsson AB	250	2011	

Project title
Power-Aware Scheduling: Infrastructure and Algorithmns

Applicant
PI Sally A. McKee

POPULAR SCIENCE DESCRIPTION

Popularscience heading and description (max 4500 char)

rubrik: En ny typ av datorsystem för applikationsspecifik bearbetning

För att ett datorprogram ska fungera måste det vara anpassat till datorns hårdvara. Programmeraren kan i de flesta fall bortse från detta faktum: hon bygger applikationer med hjälp av programmeringsspråk på en högre abstraktionsnivå, och ett översättarprogram (en kompilator) genererar den anpassade följd av instruktioner som faktiskt exekveras. Allt sedan IBM 360 på 1960-talet har datorers hårdvara beskrivits i termer av sin instruktionsuppsättning, eller ISA; ett och samma anpassade program kan exekveras på två helt olika datorer så länge deras ISAer är desamma.

I en typisk modern processor översätts varje ISA-instruktion till ett "mini-program" av mikrooperationer på en lägre abstraktionsnivå. Dessa mikrooperationer utförs sedan av hårdvaran enligt en löpande-band-princip -- s.k. pipelining -- där flera mikrooperationer samtidigt är under behandling och där påföljande mikrooperationer kan påbörjas när de nödvändiga resurserna inte längre behövs för de föregående.

ISAn erbjuder ett väldefinierat "snitt" mellan mjuk- och hårdvara, till förmån för båda. Den förra behöver inte beskriva mikrooperationerna i detalj, och ett och samma program kan exekveras på processorer med vitt skilda pris/prestanda-förhållanden. Den senare drar fördel av en regelbunden struktur som är enklare att konstruera och verifiera.

Fördelarna till trots utgör en fast ISA en begränsande faktor för prestandan. I detta projekt löser vi upp den fasta gränsen och ger mjukvaran detaljkontroll över mikrooperationerna. Härigenom öppnas möjligheter att rationalisera bort hårdvara som inte används, vilket leder till effektivare och energisnålare processorer. Kraven på kompilatorn blir å andra sidan mycket högre; särskilda metoder behövs för att hitta den bästa bland en mycket stor mängd möjliga sekvenser av mikrooperationer.

I detta projekt kommer vi att bygga på vår nyutvecklade kompilatorprototyp som uppfyller dessa krav. Härmed kan programmen få påverka processorkonstruktionen, vilken i sin tur kan motivera ändringar i mjukvaran. Resultatet blir en cykel av stegvis förfining som omfattar samkonstruktion av hård- och mjukvara för bästa prestanda på systemnivå.

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2011-40480-88494-55

Name of applicant
McKee, Sally A.

Date of birth
630611-2126

Title of research programme
FlexSoft: Software Infrastructure to Support Hardware/Software Codesign of Exposed Architectures

Appendix A

Research programme

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se

FlexSoft: Software Infrastructure to Support Hardware/Software Codesign of Exposed Architectures

1 Purpose and Aims

Computer hardware and software interact through clearly specified interfaces. In traditional computer systems, this interface is the Instruction Set Architecture (ISA), and the hardware that implements this interface can take many forms. Most modern architectures translate program instructions into micro-operations (uops) that are then executed concurrently within multiple pipeline stages [1]. Such an organization 1) allows many uops to progress in parallel within the pipeline, and 2) tends to reduce both software and hardware complexity. On the software side, the compiler need not control low-level operation of the hardware (i.e., the execution of the uops), and can instead focus on generating efficient schedules of ISA instructions. On the hardware side, the architecture replicates regular hardware constructs to implement the different pipeline stages, which simplifies both the design and the verification of its correctness.

In spite of the advantages, this rigid hardware/software contract limits how applications get translated and executed. We thus target a hardware organization that changes the hardware/software interface, exposing fine-grained control of all computational resources to the compiler. Doing so creates much potential for streamlining the hardware, which leads to more efficient processors that consume less energy. The cost is that harnessing this potential places a much heavier burden on the software: the sheer number of possible schedules of micro-operations creates an intractible solution space.

This project addresses the design of efficient compilers and tools to adapt software applications to run on such Exposed Architectures. The core of our solution is a compiler infrastructure that generates compact, efficient schedules of micro-operations. It does so by employing a SAT solver ("Satisfiability Solver") to evaluate exponential solution spaces defined by logic formulas representing schedule constraints that must be satisfied. This infrastructure is sufficiently powerful and flexible to form the basis of an end-to-end toolchain that directly impacts our ability to fully exploit Exposed Architectures. Since the compiled applications influence the required functionality of the architecture (we need not provide the means to execute all possible sequences of uops, but can instead streamline the hardware to implement only those used by the target applications), stronger compiler tools enable more efficient hardware design. This, in turn, dictates the required functionality of the software, creating a"design, evaluate, refine" cycle that embodies the fundamentals of hardware-software codesign.

2 Survey of the Field

There are rich bodies of related work in architecture, compilers, and efficient heuristics to address problems with intractible exact solutions. For instance, Very Long Instruction Word (VLIW) [2] architectures execute operations in parallel based on a fixed, compiler-defined schedule. Based on principles from horizontal microcoding [3], which concatenates

bitfields that individually and directly control different parts of the microprocessor, VLIW concatenates multiple opcodes (plus their operand information) into longer "instruction words". Just as pipeline parallelism relies on avoiding expensive branch instruction resolution, exploiting instruction-level parallelism (ILP) in VLIW architectures requires sufficiently long collections of "straight-line" VLIW instructions. Scheduling these "traces" of multiple basic blocks that are likely to be executed in sequence speculatively avoids intervening control flow instructions that thwart parallelism [4, 5]. Compensation code undoes side effects and correctly updates the PC when control flow jumps off a trace.

Just like the Exposed Architectures that we target, VLIW architectures expose many more fine-grained computational resources to the compiler. VLIW designs offered increased computational power with less hardware complexity, but the original designs were limitated by the available (admittedly aggressive) compiler technology [6]. Dramatic improvements in the strength of compiler analyses warrant revisiting the potential of architectures that rely on the compiler to manage fine-grained harware resources.

Instruction reordering and register allocation are two interdependent scheduling phases with conflicting goals. During instruction reordering, parallelism is exploited at the cost of increasing register pressure and spilling. In contrast, register allocators attempt to decrease the number of spills at the cost of instruction-level parallelism. Norris and Pollock summarize many approaches to choosing an optimal scheduling phase sequence [7]. They have also proposed several strategies for phase communication and for making the instruction scheduler and register allocator mutually sensitive. Most recent work adheres to this scheme of separate but iteratively communicating phases, but with minor variations in proposed communication strategies [8, 9]. Johnson and Mycroft instead perform CFG serialization incrementally [10]. These approaches are not directly applicable to an Exposed Architecture for which the compiler controls bith instances of executions units and forwarding paths that values take.

The computational complexity of combining the instruction reordering and register allocation is intractible (NP-hard) [11]. Employing heuristic but inexact phase communication strategies improves performance, but often makes it hard to guarantee a specific scheduling outcome. Even though SAT solvers still employ heuristics to find solution quickly [12, 13], they benefit from problem definitions that impose scheduling constraints that preserve optimization criteria. Indirectly, these constraints increase solution speed [14]. Improving performance of SAT solvers is a topic of ongoing research, including proposals to parallelize solvers for multicore and multinode platforms [15, 16], GPUs [17], and FPGAs [18]. The separation of the scheduling problem definition from the scheduling engine is the key property that enables new, even faster solvers.

3 Project Description

Impressive advances in compiler technology over the past decades make it both relevant and timely to revisit computer system designs that give software more and finer control over hardware resources. Much like VLIW machines, statically scheduled, Exposed Architectures placing all control signals under compiler management offer much potential for highly parallel, energy-efficient performance. In particular, design-time configurable datapaths can increase computational efficiency for targeted applications, but such finely tuned (application-specific) microarchitectures require configurable compilers to generate code for the architectural variants targeting different application domains. The exposed datapath is controlled by wide instruction words formed by concatenating uops, which

in this case are represented in Register Transfer Notation (RTN). Instead of relying on a hardware instruction decoder, this organization forces the compiler to bear the burden of assigning uops to functional units. In contrast to traditional, pipelined architectures whose compiler back-ends perform instruction selection and register allocation in separate, consecutive phases, Exposed Architectures require compilers to concurrently compute the mutually dependent assignments of computational units and value locations.

We first expand on our research philosophy, then we present more specifics of our target architecture to motivate our approach to software tools. Our research divides the scheduling problem into subproblems by expressing them as logical constraints to be simultaneously addressed by a SAT solver. To demonstrate the validity of this approach, we build a compiler construction library that allows generic scheduling constraints and target-specific resource constraints to be expressed declaratively. It does so in a manner independent from both the set of available execution units and from the instruction decode logic. We use this library to implement a prototype compiler for an instance of an Exposed Architecture based on the FlexCore [19] processor defined in the FlexSoC project [20].

3.1 Design-Time Configurable, Exposed Architectures

Configurable architectures can trade generality for increased performance and lower power for domain-specific applications [24]. The type and number of functional units and application-specific accelerators can be selected at design time, and the datapath between these units can then be specified to support the specific communication behaviors of applications in the target domain. Architectures that expose the control signals for all datapath units and the interconnect addresses require wide control words (as in VLIW [5]). Customizing both computational units and their communication medium can improve computational efficiency in terms of performance and power, but allowing such flexibility complicates the compiler, which must satisfy many more concurrent resource constraints and instruction dependences than typical, phase-based code generators. Nonetheless, replacing dynamic instruction scheduling performed by fixed-function logic with static instruction scheduling performed by a compiler enables dramatically more compact schedules.

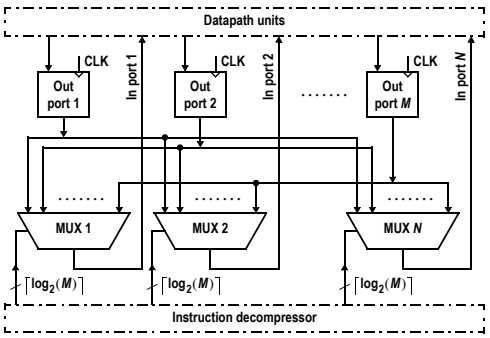

Figure 1: In its most complex configuration the datapath interconnect acts as a crossbar switch, supporting communication from each of M output ports to each of N input ports.

The architectural template of design-time configurable (application-specific), Exposed Architecture contain datapath units that communicate through an interconnect that logically forms a crossbar switch in its most complex instantiation. Figure 1 depicts the output of a particular datapath unit (i.e., an output port) connected to a register. The output of the register is routed to several input multiplexers, each driving the input of

a datapath unit (i.e., an input port). Assuming there are M output ports and N input ports, the interconnect template can, at most, support $N \cdot M$ communication paths. To avoid area, delay, and power dissipation overheads from extra wiring and multiplexers, paths unused in the target applications can be omittd.

Our platform lacks a conventional instruction set architecture (ISA) and has no fixed set of assembly instructions. Operations at the machine level are instead expressed as register transfer notations (RTN) specifying operations to be performed on output port registers of the datapath units. The output port from which a value is read represents the address of the interconnect multiplexer, and the operation represents the control signals (i.e., the op-code) to a specific datapath unit. A special RTN code reads a value from the register file without affecting the output ports of any datapath units. Decoded control words are simply concatenations of the RTN operations of all datapath units for a given clock cycle. (An orthogonal path of research addresses compact representation of these control words along with the design of efficient decoders [25].)

3.2 Compilation Approach

This section highlights differences in functionality requirements between conventional compilers and the infrastructure we will leverage in this research. To further elucidate the novelty of our approach, we provide a brief tutorial of the underlying technologies.

Traditional compilers rely schedule for fixed microarchitectures with given interconnects and functional units. Out-of-order architectures commonly replicate existing units to increase ILP, avoiding ISA changes and allowing new architectures to leverage existing compilers. In contrast, in our Exposed Architecture the data forwarding among datapath units is explicitly compiler-controlled. Output ports of datapath units include (local) registers to store the "last computed value". The compiler controls interconnect addressing, and it arranges to forward values appropriately or to store them in the register file (to compensate for producer/consumer timing mismatches). Furthermore, compilers for Von Neumann architectures rely on large register files as central operand source/repositorioes (and CISC architectures allow memory operands, of course). Hardware forwarding between pipeline stages helps overcome limitations of this fixed-pipeline/central register file execution model. Our architectures lack centralized register resources and remove hardwired pipelines, which prevent the compiler from performing sequential instruction scheduling and register allocation: in our systems, these operations must be performed concurrently and globally.

We build our prototype on top of LLVM [21], which implements complete compiler functionality (from language front-ends, to multiple optimization passes, instruction schedulers, and target-specific code generators). The highly modular LLVM infrastructure uses a single intermediate representation for every compilation step, making front-ends, back-ends, and optimization passes independent from each other — they can be developed in separate source trees and even written in different languages. Many languages can already be compiled to LLVM bytecode, and many optimization techniques are implemented as LLVM passes. Although there is much LLVM support for RISC or CISC back-ends (e.g., instruction selectors and schedulers, register allocators, and peephole optimizers), these cannot be readily reused for our hardware targets. We thus develop our own compiler back-end to compile LLVM bytecode to the RTN uops executed on our target architecture. This translation requires two major steps: lowering the LLVM instructions to the RTN micro-operations (uops) supported by a given architectural instance, and

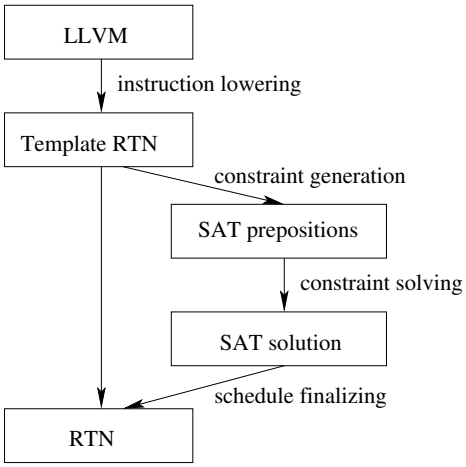

Figure 2: Compilation Flow

allocating resources to assign uops to particular execution units and to allocate registers [when necessary]). Figure 2 illustrates the flow chart of activities.

The first step generates RTN in its template form, which includes references to types of execution units (but not to specific units, themselves) and specifiers for which dataport to use on a given unit type. The static single assignment form of LLVM bytecode is preserved in this transformation, and lowering can be accomplished by instantiating instruction templates defined in terms of variant-specific micro-operations. The second step consists of resource allocation — specifying units at which to map uops and the dataports from which to read operands. Because every unit has its own register to store its result, the choice of dataports for a uop depends on the unit used to compute a given operand. Resource allocation can be expressed as a constraint satisfaction problem, and recent progress in SAT solver implementation makes such tools attractive building blocks for powerful schedulers, due to both the simplicity of formulating and refining problems and to the impressive performance [12]. The scheduling problem can then be generalized by expressing it as a set of constraint templates and then using a template engine to produce constraints; a stand-alone SAT solver finds the solution for a given set of resources. Since our declarative library is written in Haskell, and since both LLVM and RTN codes are represented with a hierarchy of abstract data types, the translator can be implemented by straightforward, recursive pattern matching. While traversing the code tree, the translator calls a target-specific, instruction-lowering function for each leaf. The type system performs dispatch according to the fully specified result types.

The parts of the compiler that are specific to a given architectural instance must then: 1)define abstract datatypes that represent new execution units, and 2) define lowering functions that transform LLVM instructions to RTN code. Symbolic value names are preserved, and spilling between output port registers, the register file, and memory is not defined until later. Spill code is inserted after evaluating constraint solutions.

For a given target architecture, an execution unit is characterized not only by its intended functionality, but also by types of inputs and outputs and by latency. A configuration file contains a list of triples, where every triple notes a type of execution unit, number of units of this type available, and the unit type's delay. For example, for a Viterbi accelerator, a configuration file entry might look like `(VITERBI, 1, 3)`, which specifies that "this architecture variant has one Viterbi accelerator, and its delay is three cycles". The scheduler relies on this information to avoid hazards when allocating resources.

The template RTN representation is transformed into final RTN by substituting full names of execution units and data ports. When the choice of an execution unit instance comes to choosing an instance number, the choice of data ports to read operands from is not straightforward. Because of spilling, an operand might come not only directly from another execution unit, but also from a buffer unit, from the register file, or from the load/store unit (essentially, from memory). Thus, the constraint set can be viewed as two individual sets— one to define the instruction placement in space (units) and time (cycles) and another to define forwarding of values between units and cycles.

Entries of the schedule can be represented with a tuple of parameters and constraints imposed on these parameters define valid schedules. Indexed variables to which the SAT solver assigns the value "true" denote valid entries. Statements can then be rewritten as propositional logic expressions [26]. The SAT solver's problem is thus to satisfy a conjunction of constraints for every instruction, pair of instructions, and execution unit. Additional constraints can be specified to discard schedules deemed suboptimal for given criteria; these require no changes to the solution interpreter, since they only refine existing solutions (as opposed to creating new classes of solutions). Likewise, extending the infrastructure to perform optimizations like trace scheduling is straightforward — the interpreter already includes a generic procedure for code rearrangement.

SAT solver interaction issues are abstracted away by means of a standard library that provides primitives for logical relations, i.e., assignments of truth values to tuples. Elements of relations correspond to indexed boolean variables that constitute boolean propositions representing stated constraints. Constraints can be imposed on relations to build clauses from variables included in the relation. After finding a solution is found, the relation data structure representing the derived schedule can be directly translated to final RTN code by a solution interpreter. Interpretation of instruction schedules represents one-to-one template substitution — every uop receives the number of a unit instance on which it is scheduled. Spill code generation takes place when the value schedule is interpreted — dataport names are supplied instead of value names for uop generation, and additional uops are inserted to ensure that values are stored in memory after being computed (and they are read back as required by other uops). Different spill strategies can optimize for either access latency or total number of transfers: our prototype uses a simple strategy that allocates faster memory to values that are soon reused.

3.3 Experimental Agenda

Our present infrastructure already generates correct schedules, and the compactness of these schedules is promising. More importantly, we have established the foundation of a comprehensive toolset that enables much more thorough codesign of application-specific, design-time configurable architectures together with the software to exploit them. Our ability to express constraints on schedules enables more software optimizations such as trace scheduling or pruning schedules to reduce the number of functional units required or the number of network connections between them. These optimizations will inform hardware simplifications that further improve the effectiveness of systems based on Exposed Architectures. Our LLVM-based prototype readily supports the study of many back-end compiler optimizations.

To improve upon our codesign infrastructure, we will refine the FlexCore hardware specification language, creating semi-automated tools to assist the hardware designer in creating efficient instruction decoders as well as in simplifying the overall hardware design.

Genetic algorithms represent one promising approach to exploring these large, complex design spaces. To extend our infrastructure further into the software domain, we will incorporate the Rose [22] source-to-souce infrastructure to enable better optimization of applications written in abstraction-rich languages like C++. Together, these front-end and back-end tools enable powerful static and dynamic optimization of target software and facilitate the design of correspondingly efficient hardware. We will investigate specific architectural support for dynamic code optimization. In its simplest form, this accelerator will implement timely "overlays" to enable small decoders to support larger sets of wide RTN instructions (the compiler must translate distinct program phases to use the current decoder patterns). More advanced accelerator designs will implement support for specific dynamic code optimizations (e.g., specialization based on runtime parameters, code restructuring for temporal locality, and dynamic trace construction).

We will empirically evaluate our tools and the designs they produce via multiple modeling approaches. Our bases for comparison will be both optimized, ILP-based pipeline architectures and established VLIW baselines [23] (note that due to the revolutionary nature of our systems, no exact apples-to-apples comparisons exist). High-level simulation verifies hardware functionality and the correctness of the hardware/software interface. More detailed, cycle-accurate simulation elucidates interactions among the software and the microarchitecture, highlighting bottlenecks to be addressed in future design cycles. Finally, FPGA hardware implementation allows even more detailed investigations of software and hardware design choices. Evaluation criteria include efficiency and accuracy of the tools we create, along with performance, energy, and hardware area analyses of the systems we create. Note that the intent of the proposed research is mainly to provide software infrastructure to support better hardware/software codesign. Producing VLSI implementations of specific architectural designs lies outside the scope of this project.

Year 1	Staff project. Student 1 improves SAT-based schedular, investigates trace scheduling, begins Rose compiler work.
	Student 2 refines FlexCore design framework, begins implementation of iterative tool for semi-automated design space exploration.
	Team publishes initial scheduling results and design of new toolset, releases alpha version of tools.
Year 2	Postdoc drives industry collaborations based on compiler results to date. Team disseminates current results and lastest version of tools. PI Larsson-Edefors
	leads empirical studies leveraging codesign infrastructure. Students identify specific sub-problems that interest them, begin planning individual dissertation research.
	Student 1 begins studying optimizations enabled by integrated Rose front-end/LLVM back-end, begins implementation of dynamic optimization system.
	Student 2 continues codesign toolset implementation, helps implement dynamic optimization.
Year 3	Students continue improving tools, continue refining and pursuing individual research agendas. Team begins study of accelerator design to support
	dynamic optimization, publishes results of integrated compiler optimizations, updates tool release, and continues technology transfer activities.
Year 4	Students continue improving tools, continue refining and pursuing individual research agendas.
	Team studies combined static and dynamic trace compilation, code restructuring, and memory optimization;
	continues dissemination and technology transfer activities.
Year 5	Team wraps up ongoing research activities, disseminates final project results and tools, continues technology transfer with industry.
	Students write and defend dissertations. PIs plan directions for future work.

Table 1: Project Management Timeline

4 Significance

This research program breathes new life into the question of the division of functionality between software and hardware. We address the design of streamlined, low-power, application-specific hardware together with the software that runs on it. Large segments of the Swedish economy rely on the production of affordable, efficient such systems (e.g., for consumer electronics, automotive control, and radio-astronomy and space applications, to name a few). Furthermore, many of the world's most powerful supercomputers are based on low-power, embedded processors [27]. Our proposed work has high potential to influence future exascale platforms, either in software or hardware. Building on Sweden's traditional computational strengths is essential for maintaining prominence in the emerging European supercomputing community.

5 Preliminary Results

Preliminary experiments compare performance of four scheduler design strategies. We compile three applications from the EEMBC embedded benchmark suite [28, 29] for our example FlexCore (exposed) architecture for a MIPS-based baseline, a FlexCore basic-block based sequential scheduler [30], the default LLVM scheduler, and our SAT-based scheduler. Table 2 compares sizes of the benchmark assembly code produced by the less powerful schedulers to that produced by our prototype. All EEMBC benchmarks include largely sequntial initialization and output phases (which make many function calls to output results); the main computation routine generally consists of nested loops (or inner loops) with few function calls. Table 2 thus lists numbers of instructions in the entire applications and in their main computational kernels.

Note that the schedulers accept different input representations — the first two operate on MIPS assembly code (a traditional and common RISC ISA), and LLVM and our infrastructure operate on LLVM bytecode. The LLVM SSA (static single assignment) form (which never reuses value names — each value is written exactly once) is lexically longer than equivalent MIPS-based representations.

	autocor		fft		viterbi	
Scheduler	all	main kernel	all	main kernel	all	main kernel
MIPS-based	346	38	691	337	617	324
sequentially phased	402	42	827	392	641	307
LLVM default	465	70	698	569	690	617
SAT-based	204	35	295	266	401	338

Table 2: Number of Generated Instructions

Our SAT-based prototype produces RTN assembly that is about twice as short as that of the other schedulers for these benchmark kernels (determining effects of schedule length on performance is part of the proposed work). Our scheduler output is 20-50% smaller for the `autocor` and `fft`transform benchmarks, which perform no function calls within their kernels. In contrast, our prototype generates slightly longer output for `viterbi`: the Viterbi kernel algorithm depends on a recursive function call, and recursion imposes high pressure on the call stack in the current version of our infrastructure. Addressing this limitation is one of the first steps in the work proposed here. In spite of its limitations, our initial prototype establishes the viability of our research approach.

We have demonstrated a prototype, SAT-based scheduler that reduces the required compiler-development effort for new instances of configurable architectures. It does so by separating architecture-specific resource constraints from generic constraints. Furthermore, it enables separating the scheduling algorithm from the scheduling engine (in this case, the SAT solver), reducing the code base that developers must maintain. We show that the modularized design of the scheduler establishes clear boundaries between loosely related properties of the schedule, which allows better hardware resource utilization. This approach embodies a powerful "divide-and-conquer" philosophy for attacking problems in the hardware/software codesign of application-specific, embedded systems. Our instruction scheduling results will generalize to the design of compiler scheduling strategies for broad classes of architectures, and our tools will inform hardware innovations.

6 Part of Project Requested

Chalmers requires 100% research support: no project activities are supported by teaching.

7 Budget Justification

The success of this research plan relies on building solid software tools, the design, implementation, debugging, and deployment/dissemination of which is notoriously labor intensive. We thus request two doctoral students, plus one postdoc to jumpstart the implementation efforts and to drive dissemination. Combining McKee's expertise in tools, compilers, software optimization, and architectures with Larsson-Edefors's expertise in hardware architecture design and circuit innovation strengthens the impact of the proposed work. Larsson-Edefors's experience with respect to the background FlexSoC project is essential to the adaptation and refinement of our software tools. Travel funds will allow team members to attend one international and one national meeting per year to disseminate results.

References

[1] J. Hennessy and D. Patterson, *Computer Organization and Design: The Hardware / Software Interface*, 2nd ed. Morgan Kaufmann Publishers, Inc., 1998.

[2] J. Fisher, "Very long instruction word architectures and the eli512," in *Proc. 10th IEEE/ACM International Symposium on Computer Architecture*, June 1983, pp. 140–150.

[3] ——, "The optimization of horizontal microcode within and beyond basic blocks; an application of processor scheduling with resources," Courant Institute of Mathematical Sciences, New York University, Tech. Rep. COO-3077-161, Oct. 1979.

[4] ——, "Trace scheduling: A technique for global microcode compaction," *IEEE Transactions on Computers*, vol. C-30, no. 7, pp. 478–490, July 1981.

[5] J. Fisher, J. Ellis, J. Ruttenberg, and A. Nicolau, "Parallel processing: A smart compiler and a dumb machine," in *Proc. ACM SIGPLAN Symposium on Compiler Construction*, June 2984, pp. 37–47.

[6] J. Ellis, "Bulldog: a compiler for VLIW architectures," Ph.D. dissertation, Yale University, Jan. 1985.

[7] C. Norris and L. Pollock, "Experiences with cooperating register allocation and instruction scheduling," *International Journal of Parallel Programming*, vol. 26, no. 3, pp. 241–284, 1998.

[8] I. Cutcutache and W.-F. Wong, "Fast, frequency-based, integrated register allocation and instruction scheduling," *Software: Practice and Experience*, vol. 38, no. 11, pp. 1105–1126, Sept. 2008.

[9] D. Koes, "Register allocation aware instruction selection," Carnegie Mellon University School of Computer Science, Tech. Rep. CMU-CS-09-169, Oct. 2009.

[10] N. Johnson and A. Mycroft, "Combined code motion and register allocation using the value state dependence graph," in *Proc. 12th International Conference on Compiler Construction*, Apr. 2003, pp. 1–16.

[11] R. Motwani, K. Palem, V. Sarkar, and S. Reyen, "Combining register allocation and instruction scheduling," Courant Institute, Tech. Rep. TR 698, July 1995.

[12] N. Een and N. Sörensson, "An extensible SAT-solver," in *Proc. International Conference on Theory and Applications of Satisfiability Testing*, ser. Lecture Notes in Computer Science, May 2003, no. 2919, pp. 333–336.

[13] M. Moskewicz, C. Madigan, Y. Zhao, L. Zhang, and S. Malik, "Chaff: Engineering

an efficient SAT solver," in *Proc. 38th ACM/IEEE Design Automation Conference*, June 2001, pp. 530–535.

[14] J. Crawford and A. Baker, "Experimental results on the application of satisfiability algorithms to scheduling problems," in *Proc. Conference on Artificial Intelligence (AAAI)*, July 1994, pp. 1092–1097.

[15] M. Lewis, T. Schubert, and B. Becker, "Multithreaded SAT solving," in *Proc. 12th Asia and South Pacific Design Automation Conference*, Jan. 2007, pp. 926–931.

[16] Y. Hamadi and L. Sais, "Manysat: a parallel sat solver," *Satisfiability, Boolean Modeling and Computation*, vol. 6, no. 12, pp. 245–262, June 2009.

[17] C. Thompson, S. Hahn, and M. Oskin, "Using modern graphics architectures for general-purpose computing: A framework and analysis," in *Proc. IEEE/ACM 35th International Symposium on Microarchitecture*, Nov. 2002, pp. 306–317.

[18] A. Dandalis and V. Prasanna, "Run-time performance optimization of an FPGA-based deduction engine for SAT solvers," *ACM Transactions on Automation of Electronic Systems*, vol. 7, no. 4, pp. 547–562, Oct. 2002.

[19] M. Thuresson, M. Själander, M. Björk, L. Svensson, P. Larsson-Edefors, and P. Stenstrom, "FlexCore: Utilizing exposed datapath control for efficient computing," *Signal Processing Systems*, vol. 57, no. 1, pp. 5–19, 2009.

[20] J. Hughes, K. Jeppson, P. Larsson-Edefors, M. Sheeran, P. Stenstrom, and L. Svensson, "FlexSoC: Combining flexibility and efficiency in SoC designs," in *Proc. IEEE NorChip Conference*, Nov. 2003, pp. 52–55.

[21] C. Lattner and V. Adve, "LLVM: A compilation framework for lifelong program analysis & transformation," in *Proc. 2nd IEEE/ACM International Symposium on Code Generation and Optimization*, Mar. 2004, pp. 75–86.

[22] D. Quinlan, "ROSE: Compiler support for object-oriented frameworks," in *Proc. Conference on Parallel Compilers*, ser. Parallel Processing Letters, vol. 10. Springer Verlag, Jan. 2000.

[23] J. Fisher, P. Faraboschi, and C. Young, *Embedded Computing: a VLIW Approach to Architecture, Compilers and Tools*. Morgan Kaufmann/Elsevier, 2005.

[24] R. Gonzalez, "Xtensa: a configurable and extensible processor," *IEEE Micro*, vol. 20, no. 2, pp. 60–70, Mar. 2000.

[25] M. Thuresson, M. Sjlander, and P. Stenstrom, "A flexible code compression scheme using partitioned look-up tables," in *Proc. High Performance Embedded Architectures and Compilers*, Jan. 2009, pp. 95–109.

[26] S. Memik and F. Fallah, "Accelerated SAT-based scheduling of control/data flow graphs," in *Proc. IEEE International Conference on Computer Design*, Sept. 2002, pp. 395–400.

[27] G. Almasi, C. Archer, J. Casaos, J. Gunnels, C.Erway, P. Heidelberger, X. Martorell, J. Moreira, K. Pinnow, J. Ratterman, B. Steinmacher-Burow, W. Gropp, and B. Toonen, "Design and implementation of message-passing services for the blue gene/l supercomputer," *IBM Journal of Research and Development*, vol. 49, no. 2/3, pp. 393–406, 2005.

[28] "Embedded Microprocessor Benchmark Consortium," http://www.eembc.org.

[29] J. Poovey, T. Conte, M. Levy, and S. Gal-On, "A benchmark characterization of the eembc benchmark suite," *IEEE Micro*, vol. 29, no. 5, pp. 18–29, Sept./Oct. 2009.

[30] T. Schilling, M. Själander, and P. Larsson-Edefors, "Scheduling for an embedded architecture with a flexible datapath," in *Proc. IEEE Symposium on VLSI*, May 2009, pp. 151–156.

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2011-40480-88494-55

Name of applicant
McKee, Sally A.

Date of birth
630611-2126

Title of research programme
FlexSoft: Software Infrastructure to Support Hardware/Software Codesign of Exposed Architectures

Appendix B

Curriculum vitae

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se

1) Education

B.A. Computer Science, Yale University, May 1985
M.S.E. Computer Science, Princeton University, January 1990

2) Doctoral Work

Ph.D. Computer Science, University of Virginia, May 1995

3) Post-Doctoral Work

Post-Doctoral Research Associate, Computer Science, University of Virginia, 1995-1996

5) Current Position

Associate Professor, Computer Science and Engineering, Chalmers University of Technology, 2008-present

6) Previous Positions

Cornell University, Ithaca, NY
- 7/02-11/08 Assistant Professor, School of Electrical and Computer Eng.
- 9/02-11/08 Member, Computer Science Graduate Field of Study

University of Utah, Salt Lake City, UT
- 9/00-7/01 Research Assistant Professor (joint), Dept.of Electrical Eng.
- 8/98-6/02 Research Assistant Professor, School of Computing

Oregon Graduate Institute of Science and Technology, Portland, OR
- 7/96-5/98 Adjunct Assistant Professor, Dept. of Computer Science and Eng.

Intel Corp., Hillsboro, OR
- 7/96-5/98 Computer Architect, Microcomputer Research Lab

8) Supervised Doctoral Degrees

1. Peter K. Szwed, Ph.D. 2011, Cornell University, dissertation: *Transparent and Efficient Coherence for Aliased Physical Memory* (1st position: Senior Engineer at IBM).
2. Vincent K. Weaver, Ph.D. 2010, Cornell University, dissertation: *Using Dynamic Binary Instrumentation to Create Fast, Validated, Multi-Core Memory Simulation* (1st position: post-doctoral fellow, High Performance Computing, Univ. of Tennessee).
3. Major B. Bhadauria, Ph.D. 2009, Cornell University, dissertation: *Thread Scheduling for Chip Multiprocessors*. *Canadian NSERC Graduate Fellow* (1st position: Senior Performance Hardware Engineer with EMC, Boston).
4. Karan Singh, Ph.D. 2009, Cornell University, dissertation: *Prediction Strategies for Power-Aware Computing on Multicore Processors* (1st position: Technical Advisor for Ropes and Gray L.L.P., Boston).
5. Brian S. White, Ph.D. 2008, Cornell University, dissertation: *Improving Computational Intensity of Irregular Mesh-Based Scientific Applications*. *Dept. of Energy Krell Institute High-Performance Computer Science Fellow* (1st position: post-doctoral fellow, Computational Bio-Physics, Cornell Univ.).
6. Dee A.B. Weikle, Ph.D. 2001, University of Virginia, dissertation: *A New Approach to Cache Analysis*. co-advised with Bill Wulf, 1997-2001 (1st position: Assistant Professor, Eastern Mennonite Univ.).

8) Additional Information

Selected Grants, Gifts, and Awards

1. EC 7th Framework Programme 249059, Embedded Reconfigurable Architectures, 256,332€, main PI S. Wong, co-PIs G. Gaydadjiev (Delft), S. Kaxiras (Uppsala), R. Giorgi, S. Bartolini (Siena), S.A. McKee (Chalmers), M. Cintra (Edinburgh), P. Gai (Evidence), G. Desoli (ST Micro), A. Zaks (IBM Israel), L. Carro (FURGS), 2010.
2. Ericsson AB, "Smarter Resource Management via Portable, Scalable, On-Line Power Estimation", 880 KSEK, PI S.A. McKee, 2009.
3. Vetenskapsrådet (VR), Chalmers Adaptive Multicore Processing Project (CHAMPP), 11,142 KSEK, Co-PIs P. Stenström, P. Lars-Edefors, S.A. McKee, L. Svensson, 2009.
4. DOE Lawrence Livermore National Laboratory Center for Applied Scientific Computing subcontract B571234, PI D. Quinlan; "Leveraging OpenAnalysis for Alias Analysis within ROSE", $40,000, PI S.A. McKee, 2007.
5. NSF CCF Award 0702616, "Towards Designing Complex Systems: Exponential Design/Configuration/Parameter Space Exploration Tools That Are Efficient, Accurate, and Easily Usable", $300,000, PI S.A. McKee, 2007.
6. NSF CNS Award 0509406, "Collaborative SMA: Dynamic Program Phase Adaptation and Hardware Reconfiguration in Multiprocessor Systems", $350,000, PI J.F. Martínez, Co-PI S.A. McKee, 2005.
7. DOE Lawrence Livermore National Laboratory Center for Applied Scientific Computing award, ASC, PI D. Dossa; "BlueGene/L: Studies in Scalability and Reconfigurability: Memory Performance", $65,000, Subcontract PI S.A. McKee, 2004.
8. NSF ST-HEC Award 0444413, "Scalable, Interoperable Tools to Support Autonomic Optimization of High-End Applications", $750,000, Main PI S.A. McKee, co-PIs A. Malony (Univ. of Oregon) and G.S. Tyson (Univ.f of Florida), 2004.
9. NSF ITR/NGS Medium Award 0325536, "Toward Autonomous Computing: System-Wide Hardware/Software Monitoring and Adaptation", $830,000, Main PI S.A. McKee, co-PI H.S. Lee (Georgia Institute of Technology)2003.
10. DOE LLNL CASC award LLNL LDRD 01-ERD-043,"Overcoming the Memory Wall for SMP-Based Systems", $238,000, PI B.R. de Supinski, co-PI S.A. McKee, 2001.

Selected External Service

ACM SIGMICRO Board, 2005-present.

International Federation for Information Processing (IFIP) Working Group 10.3, 2009-present.

Editorial Board, Springer *International Journal on Parallel Processing (IJPP)*, 2005-present.

Steering Committee, ACM Int'l. Conf. on Computing Frontiers (CF), May 2005-present.

Steering Committee, ACM Int'l. Conf. on Supercomputing (ICS), June 2011-June 2013.

IEEE/ACM/IFIP Int'l. Conf. on Parallel Architectures and Compilation Techniques (PACT), 2002-2004; 2009-2011.

Program co-Chair, ACM Int'l. Conf. on Supercomputing (ICS) (w/ B.R. de Supinski), June 2011.

General co-Chair, ACM/IEEE/IFIP International Conference on Parallel Architectures and Compilation Techniques (PACT) (w/ M. Schulz), September 2009.

Short CV of Per Larsson-Edefors (April 11, 2011)

- **University Degree:** M.Sc. 1991 in Engineering Physics and Electrical Engineering, Linköping University, Sweden
- **Doctoral Degree:** Ph.D. 1995 in Electronic Devices, Linköping University. Thesis title "High-Speed CMOS Design – Bit-Serial Arithmetic Applications and Technology Mapping of Combinational Boolean Equations". Supervisor: Professor Christer Svensson.
- **Postdoctoral Work:** National Microelectronics Research Centre, Ireland, 1996–1997. Scholarship from the Swedish Research Council for Engineering Sciences (TFR).
- **Qualification as Associate Professor:** 1998
- **Current Position:** Chair of Computer Engineering, Head of VLSI Research Group, Chalmers University of Technology, since July 2001. Research constitutes 50% of this position.
- **Previous Positions:**
  - Professor in VLSI Design, Linköping Univ., 2000
  - Visiting Professor at Microprocessor Research Labs (CRL), Intel Corp., USA, 2000
  - Acting head of Division of Electronic Devices, Linköping Univ., 1999–2000
  - Docent, Linköping Univ., 1998–1999
  - Associate Professor (lektor), Linköping Univ., 1997–1998
- **Parental Leave:** 50% Sept. 2004–Dec. 2004, 100% June 1998–Aug. 1998, 100% Aug. 1995–Oct. 1995
- **Graduate Students Supervised to Doctorate:**
  - Magnus Själander, "Efficient and Flexible Embedded Systems and Datapath Components", June 2008
  - Daniel Andersson, "Interconnect Delay and Integrity Issues in On-Chip Circuits", June 2008 (co-supervised with Lars "J" Svensson)
  - Minh Q. Do, "Accurate Leakage-Conscious Architecture-Level Power Estimation for SRAM-based Memory Structures", June 2007
  - Mindaugas Drazdziulis, "Static Power Reduction and Estimation in CMOS Circuits Considering Emerging Leakage Mechanisms", 2006
  - Daniel Eckerbert, "Power Estimation and Multi-Phase Clock Generation for the Deep Submicron Era", 2003
  - Henrik Eriksson, "Efficient Implementation and Analysis of CMOS Arithmetic Circuits", 2003
  - Atila Alvandpour, "Power Estimation and Low Power CMOS Techniques", 1999 (co-supervised with Christer Svensson)
- **Current Graduate Students:**
  - Tung Hoang, LicEng, MSc; Erik Ryman, MSc; Kasyab Subramaniyan, MSc; Alen Bardizbanyan, MSc

- **Other:**
  - Technical program committee duties: Symp. on VLSI Circuits [2002-2006], IEEE Int. Conf. on Computer Design (ICCD) [2003-2005], European Solid-State Circuits Conf. (ESSCIRC) [2003-], Int. Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS) [1998-2006], Designing Correct Circuits (DCC) Workshop [2006], IEEE Conf. on Ph.D. Research in Microelectronics and Electronics (PRIME) [2011]. Session chairman at numerous occasions for Symp. on VLSI Circuits and ESSCIRC.
  - Reviewer for international journals: IEEE J. of Solid-State Circuits, IEEE Trans. on Computers, IEEE Trans. on Circuits and Systems (part I and II), IEEE Trans. on VLSI Systems, IEEE Trans. on Electron Devices, IEE/IET Electronics Letters, IEE/IET Proceedings–Circuits, Devices and Systems, IEE/IET Proceedings–Computers and Digital Techniques, J. of Systems Architecture (Elsevier), J. of Signal Processing Systems (Springer), Integration-the VLSI Journal, Transactions on HiPEAC.
  - Reviewer for international conferences: Symp. on VLSI Circuits, IEEE Int. Conf. on Computer Design (ICCD), European Solid-State Circuits Conf. (ESSCIRC), IEEE Int. Symp. on Circuits and Systems (ISCAS), IEEE Int. Conf. on Electronics, Circuits and Systems (ICECS), Int. Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS), Designing Correct Circuits (DCC) Workshop, IEEE Int. Symp. on High-Performance Computer Architecture (HPCA), IEEE/ACM Int. Symp. on Computer Architecture (ISCA).
  - Invited reviewer for funding applications: the Knowledge Foundation (KK-stiftelsen), the IT faculty of Umeå University, the VR–Swedish Research Council (both EURYI and postdocs), the electrical engineering panel of Academy of Finland (chairman of Academy of Finland's EE panel in 2011).
  - Peer-review evaluation: Member of international peer-review panel on nation-wide evaluation (Technopolis) of research and funding system in Czech Republic (2011).
  - Graduate student evaluation: Eight times on PhD defence evaluation committee, four times as chairman, and opponent on several defences.
  - Commissions of trust: On the appointment board of the Computer Science and Engineering, Chalmers, since 2002. Elected member of the Chalmers faculty senate (2002-2004). On the Chalmers honorary doctorate committee (2002-2007). On the Chalmers Signals and Systems department council since 2009. Director (and responsible for the development) of the new Master's program in Integrated Electronic System Design at Chalmers (2005-2008). On the working group that defined the nation-wide SoCWare Master's program (1999-2000).
  - Awards: Chalmers nominee for Tage Erlander Prize 2003. Invited Visiting Professor at Intel Corp. 2000. Recipient of TFR postdoctor scholarship 1996.

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2011-40480-88494-55

Name of applicant
McKee, Sally A.

Date of birth
630611-2126

Title of research programme
FlexSoft: Software Infrastructure to Support Hardware/Software Codesign of Exposed Architectures

Appendix C

List of publications

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se

(citations from Google Scholar; untrackable contributions marked with "*"; most important publications with respect to this proposal prefaced by "*")

Refereed Journals

1. N. Xu, S.A. McKee, L. Nozick, R. Ufomata, "Augmenting Priority Rule Heuristics with Justification and Rollout to Solve the Resource-Constrained Project Scheduling Problem", *Elsevier Computers and Operations Research*, 35(10):3284-3297, October 2008. number of citations: 13
2. K. Singh, E. Ipek, S.A. McKee, B.R. de Supinski, M. Schulz, R. Caruana, "Predicting Parallel Application Performance via Machine Learning Approaches", *Wiley Concurrency and Computation: Practice and Experience*, 19(17):2219-2235, May 2008. number of citations: 14
3. E. Ipek, S.A. McKee, K. Singh, R. Caruana, B.R. de Supinski, M. Schulz, "Efficient Architectural Design Space Exploration via Predictive Modeling", *ACM Transactions on Architecture and Code Optimization*, 4(4), Article 1, January 2008. number of citations: 26
4. M. Bhadauria, S.A. McKee, K. Singh, G.S. Tyson, "Data Cache Techniques to Save Power and Deliver High Performance in Embedded Systems", *Transactions on High Performance Embedded Architectures and Compilers*, 2(1):62-81, 2007. number of citations: 14
5. J. Marathe, F. Mueller, T. Mohan, S.A. McKee, B.R. de Supinski, A. Yoo, "METRIC: Memory Tracing via Dynamic Binary Rewriting to Identify Cache Inefficiencies", *ACM Transactions on Programming Languages and Systems,* 29(2), Article 12, April 2007. number of citations: 50
6. M.J. Geiger, S.A. McKee, G.S. Tyson, "Specializing Cache Structures for High Performance and Energy Conservation in Embedded Systems", *Transactions on High Performance Embedded Architectures and Compilers*, 1(1):50-90, January 2007. number of citations: 21
7. A. Bunker, G. Gopalakrishnan, S.A. McKee, "Formal Hardware Specification Languages for Protocol Compliance Verification", *ACM Transactions on Design Automation of Electronic Systems*, 9(1):1-32, January 2004. number of citations: 49
8. B. Chandramouli, W.C. Hsieh, J.B. Carter, S.A. McKee, "A Cost Model for Integrated Restructuring Optimizations", *Journal of Instruction Level Parallelism*, August 2003. number of citations: 1
9. V.S. Pingali, S.A. McKee, W.C. Hsieh, J.B. Carter, "Restructuring Computations for Temporal Data Cache Locality", Springer *International Journal of Parallel Programming*, 31(4):305-338, August 2003. number of citations: 33

Refereed Conferences

10. S. Wong, A. Brandon, F. Anjam, R. Seedorf, R. Giorgi, Z. Yu, N. Puzovic, S.A. McKee, M. Själander, G. Keramidas, L. Carro, "Early Results from ERA – Embedded Reconfigurable Architectures", Portugal, July 2011, to appear.
11. J. Wiedendorfer, T. Küstner, S.A. McKee, "Performance Optimization by Dynamic Code Transformation", Proc. ACM International Conference on Computing Frontiers (CF'11), Ischia, IT, May 2011 (short paper and poster), to appear.
12. J. Lidman, S.A. McKee, "Increasing Opportunities for Automatic Parallelization", ACM Conference on Code Generation and Optimization, Chamoninx, FR, April 2011 (abstract and poster). number of citations: 0
13. P. Larsen, R. Ladelsky, J. Lidman, S.A. McKee, S. Karlsson, A. Zaks, "Interactive Compilation and Code Modification to Automatically Parallelize Loops", ACM Conference on Code Generation and Optimization, Chamoninx, FR, April 2011 (abstract and poster). number of citations: 0
14. N. Puzovic, S.A. McKee, R. Eres, A. Zaks, P. Gai, S. Wong, R. Giorgi, "A Multi-Pronged Approach to Benchmark Characterization", Proc. IEEE International Conference on Cluster Computing (CLUSTER), Crete, GR, September 2010 (short paper and poster). number of citations: 0
15. R. Gioiosa, S.A. McKee, M. Valero, "Designing an OS for HPC: Scheduling", Proc. IEEE International Conference on Cluster Computing (CLUSTER), Crete, GR, September 2010 pp. 78-87 (31% acceptance). number of citations: 0
16. K. Singh, M. Curtis-Maury, S.A. McKee, F. Blagojevic, D.S. Nikolopoulos, B.R. de Supinski, M. Schulz, "Comparing Scalability Prediction Strategies on an SMP or CMPs", Proc. Euro-Par, Ischia, IT, September 2010, pp. 143-155 (35% acceptance). number of citations: 4
17. B. Goel, S.A. McKee, R. Gioiosa, K. Singh, M. Bhadauria, M. Cesati, "Portable per-Core Power Estimation for Intelligent Resource Management", Proc. IEEE 1st International Green Computing Conference (IGCC), Chicago, IL, August 2010. number of citations: 3
18. M. Bhadauria, S.A. McKee, "An Approach to Resource-Aware co-Scheduling for CMPs", Proc. ACM International Conference on Supercomputing (ICS), Tsukuba, JP, June 2010, pp. 189-199 (18% acceptance). number of citations:8
19. M. Zahran, S.A. McKee, "Global Management of Cache Hierarchies", Proc. ACM International Conference on Computing Frontiers (CF'10), Bertinoro, IT, May 2010, pp. 131-140 (27% acceptance). number of citations: 4

*20. V.M. Weaver, S.A. McKee, "Code Density Concerns for New Architectures", Proc. 27th IEEE International Conference on Computer Design (ICCD), Lake Tahoe, CA, October 2009, pp. 459-464 (34% acceptance). number of citations: 15

21. M. Bhadauria, V.M. Weaver, S.A. McKee, "Understanding PARSEC Performance on Contemporary CMPs", Proc. IEEE International Symposium on Workload Characterization (IISWC), Austin, TX, October 2009, pp. 98-107 (41% acceptance). number of citations: 6

22. K. Singh, M. Bhadauria, S.A. McKee, “Prediction-Based Power Estimation and Scheduling for CMPs”, Proc. ACM International Conference on Supercomputing (ICS), Manhattan, NY, June 2009, pp. 501-502 (extended abstract and poster). number of citations: 0
23. M. Bhadauria, V. Weaver, S.A. McKee, “PARSEC: Hardware Profiling for CMP Design of Emerging Workloads”, Proc. ACM International Conference on Supercomputing (ICS), Manhattan, NY, June 2009, pp. 509-510 (extended abstract and poster). number of citations:0
24. Md. Mafijul Islam, P. Stenström, S.A. McKee, “Cancellation of Loads that Return Zero Using Zero Valued Caches”, Proc. ACM International Conference on Supercomputing (ICS), Manhattan, NY, June 2009, pp. 493-494 (extended abstract and poster). number of citations: 0
25. P.E. West, Y. Peress, G.S. Tyson, S.A. McKee, “Core Monitors: Monitoring Performance in Multicore Processors”, Proc. ACM International Conference on Computing Frontiers (CF), Ischia, IT, May 2009, pp. 31-40 (23% acceptance). number of citations: 12
26. G. Bronevetsky, K. Pingali, D. Marques, R. Rugina, S.A. McKee, “Compiler-Enhanced Incremental Checkpointing for OpenMP Applications”, Proc. International Parallel and Distributed Processing Symposium (IPDPS), Rome, IT, May 2009, pp.1-12 (23% acceptance). number of citations: 8
27. J. Li, X. Ma, K. Singh, M. Schulz, B.R. de Supinski, S.A. McKee, “Machine Learning Based Online Performance Prediction for Runtime Parallelization and Task Scheduling”, Proc. IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Boston, MA, April 2009, pp. 89-100 (28% acceptance). number of citations: 5
28. V.M. Weaver, S.A. McKee. “Optimizing for Size: Exploring the Limits of Code Density”, Proc. Architectural Support for Programming Languages and Operating Systems (ASPLOS), Washington DC, March 2009 (poster). number of citations: 0
29. G. Venkatasubramanian, D. Wolinsky, R.J.O. Figueiredo, P.O. Boykin, J.A.B. Fortes, T. Li, J.-K. Peir, L.K. John, D. Kaeli, D. Lilja, S.A. McKee, G. Memik, A. Roy, B. Burnett, G.S. Tyson, “A Community Distributed Infrastructure for Computer Architecture Research and Education”, Proc. Architectural Support for Programming Languages and Operating Systems
(ASPLOS), Washington DC, March 2009 (poster). number of citations: 7
30. M. Bhadauria, V.M. Weaver, S.A. McKee, “Accommodating Diversity in CMPs with Heterogeneous Frequencies”, Proc. 4th EC International Conference on High Performance Embedded Architectures and Compilers (HiPEAC), Paphos, CY, January 2009, pp. 248-262 (28% acceptance). number of citations: 0
31. M.A. Watkins, S. McKee, L. Schaelicke, “Revisiting Cache Block Superloading: A Phase-Adaptive Approach to Increasing Cache Performance”. Proc. 4th EC International Conference on High Performance Embedded Architectures and Compilers (HiPEAC), Paphos, CY, January 2009, pp. 339-354 (28% acceptance). number of citations: 11

32. R.J.O. Figueiredo, P.O. Boykin, J.A.B. Fortes, T. Li, J.-K. Peir, D. Wolinsky, L.K. John, D.R. Kaeli, D.J. Lilja, S.A. McKee, G. Memik, A. Roy, G.S. Tyson, "Archer: A Community Distributed Computing Infrastructure for Computer Architecture Research and Education", 4th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), November 2008, pp. 70-84. number of citations: 7
33. V.M. Weaver, S.A. McKee, "Can Hardware Performance Counters be Trusted?", Proc. IEEE International Symposium on Workload Characterization (IISWC), Seattle, WA, September 2008, pp. 141-150 (35% acceptance). number of citations: 9
34. P.A. Castillo Valdivieso, J.J. Merelo Guervós, M. Moretó, F.J. Cazorla, M. Valero, A.M. Mora, J.L. Jiménez Laredo, S.A. McKee, "Evolutionary System for Prediction and Optimization of Hardware Architecture Performance", Proc. IEEE Congress on Evolutionary Computation (CEC), Hong Kong, CN, May 2008, pp. 1941-1948. number of citations: 1

*35. B.S. White, S.A. McKee, D.J. Quinlan, "A Projection-Based Optimization Framework for Abstractions with Application to the Unstructured Mesh Domain", Proc. ACM International Conference on Supercomputing (ICS), Kos, GR, June 2008, pp.104-113. number of citations: 5

36. M. Bhadauria, S.A. McKee, "Optimizing Thread Throughput for Multithreaded Workloads on Memory Constrained CMPs", Proc. ACM Computing Frontiers (CF), Ischia, IT, May 2008. cited by: 16
37. G. Bronevetsky, D. Marques, K. Pingali, S.A. McKee, R. Rugina, "Compiler-Enhanced Incremental Checkpointing for OpenMP Applications", Proc. 13th ACM Symposium on Principles and Practices of Parallel Programming (PPoPP), Salt Lake City, UT, February 2008, pp. 275-276 (extended abstract and poster). number of citations: 8
38. V.M. Weaver, S.A. McKee, "Using Dynamic Binary Instrumentation to Generate Multi-Platform SimPoints: Methodology and Accuracy", Proc. 3rd EC International Conference on High Performance Embedded Architectures and Compilers (HiPEAC), Gothenburg, SE, January 2008, pp. 305-319 (34% acceptance). number of citations: 6
39. M. Watkins, S.A. McKee, L. Schaelicke, "A Phase Adaptive Approach to Increasing Cache Performance", Proc. 16th IEEE/ACM International Conference on Parallel Architectures and Compilation Techniques (PACT), Brasov, RO, September 2007 (extended abstract and poster). number of citations: 0
40. C. Dolen, C. Haymes, K. Inoue, D. Kuchta, S. Lekuch, J.E. Moreira, E. Shenfield, X. Shen, C. Trammell, M. Tsao, S.A. McKee, "Chameleon Shared Memory Project", Proc. 8th Linux Cluster Institute International Conference on High-Performance Clustered Computing (LCI), Lake Tahoe, CA, May 2007 (best poster). number of citations: 0
41. B.C. Lee, D.M. Brooks, B.R. de Supinski, M. Schulz, K. Singh, S.A. McKee, "Methods of Inference and Learning for Performance Modeling of Parallel Applications", Proc. ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP), San Jose, CA, March 2007, pp. 240-258 (33% acceptance). number of citations: 42

42. M. Bhadauria, S.A. McKee, K. Singh, G.S. Tyson, “Leveraging High Performance Data Cache Techniques to Save Power in Embedded Systems”, Proc. 2nd EC International Conference on High Performance Embedded Architectures and Compilers (HiPEAC), Ghent, BE, February 2007, pp. 23-37 (29% acceptance). number of citations: 15
43. N.B. Sam, S.A. McKee, P. Kudva, “Rethinking Processor Design: Parameter Correlations”, Proc. 13th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Nice, FR, December 2006, pp. 156-159 (62% acceptance). number of citations: 0
44. E. Ipek, S.A. McKee, B.R. de Supinski, M. Schulz, R. Caruana, “Efficiently Exploring Architectural Design Spaces via Predictive Modeling”, Proc. 12th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), San Jose, CA, October 2006, pp. 195-206 (24% acceptance). number of citations: 116
45. M. Bhadauria, S.A. McKee, K. Singh, G.S. Tyson, “A Precisely Tunable Drowsy Cache Management Mechanism”, Proc. IBM P=ac2 (P= Power/Performance, a = architecture, and c2 = $\mathrm{circuits} \times \mathrm{compilers}$) Conference, October, 2006 (33% acceptance). number of citations: 3
46. M.J. Geiger, S.A. McKee, G.S. Tyson, “Beyond Region Caching: Specializing Cache Structures for High Performance and Energy Conservation”, Proc. EC International Conference on High Performance Embedded Architectures and Compilers (HiPEAC), Barcelona, ES, November 2005, pp. 102-115 (20% acceptance). number of citations: 11
47. E. Ipek, B.R. de Supinski, M. Schulz, S.A. McKee, “An Approach to Performance Prediction for Parallel Applications”, Proc. Euro-Par, Lisbon, PT, August 2005, pp. 196-205 (30% acceptance). number of citations: 43

*48. B.S. White, S.A. McKee, B.R. de Supinski, B.J. Miller, D.J. Quinlan, M. Schulz, “Improving the Computational Intensity of Unstructured Grid Applications”, Proc. ACM International Conference on Supercomputing (ICS), Boston, MA, June 2005, pp. 341-350 (28% acceptance). number of citations: 13

49. M.J. Geiger, S.A. McKee, G.S. Tyson, “Drowsy Region-Based Caches: Minimizing Both Dynamic and Static Power Dissipation”, Proc. ACM International Conference on Computing Frontiers (CF), Ischia, IT, May 2005, pp. 378-384 (acceptance 41%). number of citations: 23

*50. M. Schulz, B.S. White, S.A. McKee, H.S. Lee, J. Jeitner, “Owl: Next-Generation System Monitoring”, Proc. ACM International Conference on Computing Frontiers (CF), Ischia, IT, May 2005, pp. 116-124 (acceptance 41%). number of citations: 33

*51. S.A. McKee, “Reflections on the Memory Wall”, Proc. ACM International Conference on Computing Frontiers (CF), Ischia, IT, April 2004, pp. 162-167 (invited paper). number of citations: 86

52. T. Mohan, B.R. de Supinski, S.A. McKee, F. Mueller, A. Yoo, M. Schulz, “Identifying and Exploiting Spatial Regularity in Data Memory References”, Proc. ACM/IEEE Supercomputing: International Conference on High Performance Computing, Networking, Storage, and Analysis (SC), Phoenix, AZ, November 2003, p. 49 (one of six

candidates for best paper, 29% acceptance). number of citations: 21

53. S.A. McKee, D.M. Kubarek, “Real World Engineering: a Course for Masters Students Headed for Industry”, Proc. ASEE/IEEE Frontiers in Education Conference (FIE), Boulder, CO, November 2003, Session F1E, pp. 16-21. number of citations: 1
54. T. Mu, J. Tao, M. Schulz, S.A. McKee, “Interactive Locality Optimization on NUMA Architectures”, Proc. ACM Symposium on Software Visualization (SoftVis), San Diego, CA, June 2003, pp. 133-142 (31% acceptance). number of citations: 20
55. M. Schulz, S.A. McKee, “A Framework for Portable Shared-Memory Programming”, Proc. IEEE/ACM International Parallel and Distributed Processing Symposium (IPDPS), Nice, France, April 2003, pp. 54-62 (29% acceptance). number of citations: 0

Book Chapter

1. S.A. McKee, R.W. Wisniewski, “The Memory Wall”, *Encyclopedia of Parallel Computing*, D. Padua, Ed., Springer Verlag, 2011, to appear.

Selected Refereed Workshops

1. K. Singh, M. Bhadauria, S.A. McKee, “Real Time Power Sstimation and Thread Scheduling via Performance Counters”, Proc. Workshop on Design, Architecture, and Simulation of Chip Multi-Processors, Como, IT, November 2008 (also published as ACM SIGARCH Computer Architecture News, 37(2):46-55, May 2009). number of citations: 21
2. M. Andersson, L. Svensson, M. Sjalander, S. A. McKee, E. Catovic, P. Ingelhag, “Yield Optimization Using Redundant Cores”, Third Swedish Workshop on Multi-Core Computing, October 2010. number of citations: 0

Popular Science Presentations

1. R. Figueiredo, co-PIs J.-K. Peir, J.A.B. Fortes, T. Li, P.O. Boykin, G.S. Tyson, L.K. John, D. Kaeli, D. Lilja, G. Memik, S.A. McKee, “Archer: Zero-Configuration Virtual Appliances for Architecture Simulation”, Tutorial in conjunction with IEEE International Symposium on Workload Characterization (ISSWC’09), Austin, TX, October 2009. number of citations: *
2. D. Brooks, B.R. de Supinski, B.C. Lee, S.A. McKee, M. Schulz, K. Singh, “Methods of Learning and Inference for Large Design and Parameter Spaces”, Tutorial in conjunction with 13th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS XIII), Seattle, WA, March 2008. number of citations: *
3. S.A. McKee, K.Singh, D. Brooks, B.C. Lee, B.R. de Supinski, M. Schulz, “Inference and Learning for Large Scale Micro-Architectural Analysis”, Tutorial in conjunction with the ACM/IEEE International Symposium on Computer Architecture (ISCA), Federated Computer Research Conferences, San Diego, CA, June 2007. number of citations: *

Five Most Cited Publications

1. Wm.A. Wulf, S.A. McKee, “Hitting the Memory Wall: Implications of the Obvious”, *Computer Architecture News*, 23(1):20-24, March 1995. number of citations: 642
2. E. Ipek, S.A. McKee, B.R. de Supinski, M. Schulz, R. Caruana, “Efficiently Exploring Architectural Design Spaces via Predictive Modeling”, Proc. 12th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), San Jose, CA, October 2006, pp. 195-206 (24% acceptance). number of citations: 116
3. L. Zhang, Z. Fang, M. Parker, B.K. Mathew, L. Schaelicke, J.B. Carter, W.C. Hsieh, S.A. McKee, “The Impulse Memory Controller”, *IEEE Transactions on Computers*, 50(11):1117-1132, November 2001. number of citations: 98
4. S.A. McKee, Wm. A. Wulf, “Access Ordering and Memory-Conscious Cache Utilization”, Proc. IEEE Symposium on High Performance Computer Architecture (HPCA), Raleigh, NC, January 1995, pp. 253-262 (19% acceptance). number of citations: 100
5. S.A. McKee, “Reflections on the Memory Wall”, Proc. ACM International Conference on Computing Frontiers (CF), Ischia, IT, April 2004, pp. 162-167 (invited paper). number of citations: 86

Publication List (2003–) of Per Larsson-Edefors (April 13, 2011)

Citations according to Google Scholar.

Reviewed Journal Papers:

1. T. T. Hoang, M. Själander, and P. Larsson-Edefors, "High-Speed, Energy-Efficient 2-Cycle Multiply-Accumulate (MAC) Architecture and Its Application to a Double-Throughput MAC Unit", IEEE Transactions on Circuits and Systems, I: Regular papers, Invited for SOCC special issue, vol. 57, no. 12, pp. 3073-3081, Dec. 2010.

2. P. Larsson-Edefors, "Teaching Bachelors Electronic Circuits with Electronic Systems Design in Mind", International Journal of Electrical Engineering Education, vol. 47, no. 3, pp. 263-276, July 2010.

3. (*) M. Thuresson, M. Själander, M. Björk, L. Svensson, P. Larsson-Edefors, and P. Stenstrom, "FlexCore: Utilizing Exposed Datapath Control for Efficient Computing", Journal of Signal Processing Systems, vol. 57, no. 1, pp. 519, Oct. 2009. Number of citations: 24.

4. M. Själander and P. Larsson-Edefors, "Multiplication Acceleration through Twin Precision", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 17, no. 9, pp. 1233-46, Sept. 2009. Number of citations: 12.

5. D. Andersson, L. "J" Svensson, and P. Larsson-Edefors, "Time-Domain Interconnect Characterization Flow for Appropriate Model Segmentation", IET Computers and Digital Techniques, vol. 2, no. 4, pp. 265-274, July 2008.

6. H. Eriksson, P. Larsson-Edefors, and D. Eckerbert, "Toward Architecture-Based Test-Vector Generation for Timing Verification of Fast Parallel Multipliers", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 14, no. 4, pp. 370-9, April 2006. Number of citations: 12.

Reviewed Conference Papers[1]:

7. (*) B. Hidaji, S. Alipour, J. Lidman, K. P. Subramaniyan, and P. Larsson-Edefors, "Application-Specific Energy Optimization of General-Purpose Datapath Interconnect", to be presented/published at IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Chennai, India, July 4-6 2011.

[1]Conference articles in this field of computer science and engineering are peer-reviewed full articles typically ranging 4-6 pages, and are the normal form of refereed publication. The top conferences in each subfield typically have the highest impact factor within that field. All articles listed in this category are selected for publication after a rigorous peer-review process.

8. (*) A. Bardizbanyan, M. Själander, and P. Larsson-Edefors, "Reconfigurable Instruction Decoding for a Wide-Control-Word Processor", to be presented/published at Reconfigurable Architectures Workshop (RAW), IEEE International Symposium on Parallel and Distributed Processing (IPDPS), Anchorage, Alaska, USA, May 16-17 2011.

9. M. Olsson, J. Pihl, D. Andersson, and P. Larsson-Edefors, "Extracting Vectors from Application Traces for Power Integrity Analysis", to be presented/published at IEEE Workshop on Signal Propagation on Interconnects (SPI), Naples, Italy, May 9-12 2011.

10. M. W. Azhar, T. T. Hoang and P. Larsson-Edefors, "Cyclic Redundancy Checking (CRC) Accelerator for the FlexCore Processor", Proceedings of Euromicro Conference on Digital System Design (DSD), pp. 675-680, Lille, France, Sept. 1-3 2010.

11. (*) T. T. Hoang, U. Jälmbrant, E. der Hagopian, K. P. Subramaniyan, M. Själander, and P. Larsson-Edefors, "Design Space Exploration for an Embedded Processor with Flexible Datapath Interconnect", Proceedings of IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP), pp. 55-62, Rennes, France, July 7-9 2010.

12. A. Bardizbanyan, K. P. Subramaniyan, and P. Larsson-Edefors, "Generation and Exploration of Layouts for Area-Efficient Barrel Shifters", Proceedings of IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 454-455, Lixouri Kefalonia, Greece, July 5-7 2010.

13. L. "J" Svensson, J. Pihl, D. A. Andersson, and P. Larsson-Edefors, "On-chip Power Supply Noise and Its Implications on Timing", Proceedings of the 20th ACM Great Lakes Symposium on VLSI (GLSVLSI), pp. 389-392, Providence, Rhode Island, USA, May 16-18 2010. Number of citations: 1.

14. E. Ryman, P. Larsson-Edefors, A. Emrich, L. "J" Svensson, and S. Andersson, "A Single-Chip 64 Input Low Power High Speed Cross Correlator for Space Application", Proceedings of European Space Agency Microwave Technology and Techniques Workshop, Noordwijk, the Netherlands, May 10-12 2010.

15. K. Jeppson, L. Peterson, L. Svensson, and P. Larsson-Edefors, "Implementing Constructive Alignment in a CDIO-oriented Master's Program in Integrated Electronic System Design", Proceedings of European Workshop on Microelectronics Education (EWME), pp. 135-140, Darmstadt, Germany, May 10-12 2010.

16. E. Ryman, P. Larsson-Edefors, K. Subramaniyan, M. Islam, T. Hoang, and M. Själander, "FlexTools: Design Space Exploration Tool Chain from C to Physical Implementation", CDNLive! EMEA 2010, Munich, Germany, May 4-6 2010.

17. E. Ryman, A. Emrich, J. Embretsen, J. Riesbeck, S. Andersson, P. Larsson-Edefors, and L. Svensson, "Digital Cross-Correlators: Two Approaches", Proceedings of Gigahertz Symposium, pp. 39, Lund, Sweden, March 9-10 2010.

18. P. Kimfors, N. Broman, A. Haraldsson, K. P. Subramaniyan, M. Själander, H. Eriksson, and P. Larsson-Edefors, "Custom Layout Strategy for Rectangle-Shaped Log-Depth Multiplier Reduction Tree", Proceedings of IEEE International Conference on Electronics, Circuits and Systems, pp. 77-80, Tunisia, Dec. 13-16, 2009.

19. K. P. Subramaniyan, E. Axelsson, M. Sheeran, and P. Larsson-Edefors, "Layout Exploration of Geometrically Accurate Arithmetic Circuits", Proceedings of IEEE International Conference on Electronics, Circuits and Systems, pp. 795-798, Tunisia, Dec. 13-16, 2009. Number of citations: 1.

20. T. T. Hoang, M. Själander, and P. Larsson-Edefors, "High-Speed, Energy-Efficient 2-Cycle Multiply-Accumulate Architecture", Proceedings of IEEE System on Chip Conference (SOCC), pp. 119-122, Belfast, UK, Sept. 9-11, 2009. Number of citations: 2.

21. L. "J" Svensson, J. Pihl, D. A. Andersson, B. Nilsson, and P. Larsson-Edefors, "Towards Supply-Grid-Based Derating of Timing Margins", Proceedings of IEEE Workshop on Signal Propagation on Interconnects (SPI), Strasbourg, France, May 12-15 2009. Number of citations: 1.

22. D. A. Andersson, B. Nilsson, J. Pihl, L. "J" Svensson, and P. Larsson-Edefors, "Supply Voltage Drop Study Considering On-Chip Self Inductance of a 32-bit Processor's Power Grid", Proceedings of IEEE Workshop on Signal Propagation on Interconnects (SPI), Strasbourg, France, May 12-15 2009. Number of citations: 2.

23. T. Schilling, M. Själander, and P. Larsson-Edefors, "Scheduling for an Embedded Architecture with a Flexible Datapath", Proceedings of IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 151-6, Tampa, USA, May 13-15 2009. Number of citations: 3.

24. T. T. Hoang, M. Själander, and P. Larsson-Edefors, "Double Throughput Multiply-Accumulate Unit for FlexCore Processor Enhancements", Reconfigurable Architectures Workshop (RAW), Proceedings of IEEE International Symposium on Parallel and Distributed Processing (IPDPS), Rome, May 23-29 2009. Number of citations: 3.

25. M. Själander and P. Larsson-Edefors, "High-Speed and Low-Power Multipliers Using the Baugh-Wooley Algorithm and HPM Reduction Tree", Proceedings of IEEE International Conference on Electronics, Circuits and Systems, pp. 33-36, Malta, Aug 31-Sept 3 2008. Number of citations: 4.

26. K. Jeppson, L. Peterson, L. Svensson, L. Bengtsson, and P. Larsson-Edefors, "A New Master's Program in Integrated Electronic System Design", European Workshop on Microelectronics Education (EWME), Budapest, May 28-30 2008.

27. D. A. Andersson, L. J. Svensson, and P. Larsson-Edefors, "Noise-Aware On-Chip Power Grid Considerations Using a Statistical Approach", Proceedings of International Symposium on Quality Electronic Design (ISQED), pp. 663-9, San Jose, CA, USA, March 17-19 2008. Number of citations: 2.

28. D. A. Andersson, S. Kristiansson, L. J. Svensson, P. Larsson-Edefors, and K. O. Jeppson, "Noise Interaction Between Power Distribution Grids and Substrate", Proceedings of International Symposium on Quality Electronic Design (ISQED), pp. 84-9, San Jose, CA, USA, March 17-19 2008.

29. M. Q. Do, P. Larsson-Edefors, and M. Drazdziulis, "High-Accuracy Architecture-Level Power Estimation for Partitioned SRAM Arrays in a 65-nm CMOS BPTM Process", Invited paper, Proceedings of Euromicro Conference on Digital System Design, pp. 249-256, Lübeck, Germany, Aug. 27-31 2007.

30. M. Thuresson, M. Själander, M. Björk, L. "J" Svensson, P. Larsson-Edefors, and P. Stenstrom, "FlexCore: Utilizing Exposed Datapath Control for Efficient Computing", Proceedings of International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (IC-SAMOS), pp. 18-25, Samos, Greece, July 16-19 2007. Number of citations: 24.

31. D. A. Andersson, L. "J" Svensson, and P. Larsson-Edefors, "Toward a Systematic Sensitivity Analysis of On-Chip Power Grids Using Factor Analysis", Proceedings of IEEE Workshop on Signal Propagation on Interconnects (SPI), pp. 155-158, Genova, Italy, May 13-16, 2007. Number of citations: 2.

32. M. Drazdziulis, P. Larsson-Edefors, and L. "J" Svensson, "Overdrive Power-Gating Techniques for Total Power Minimization", Proceedings of IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 125-30, Porto Alegre, Brazil, May 9-11 2007.

33. M. Själander, P. Larsson-Edefors, and M. Björk, "A Flexible Datapath Interconnect for Embedded Applications", Proceedings of IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 15-20, Porto Alegre, Brazil, May 9-11 2007. Number of citations: 8.

34. M. Q. Do, M. Drazdziulis, P. Larsson-Edefors, and L. Bengtsson, "Leakage-Conscious Architecture-Level Power Estimation for Partitioned and Power-Gated SRAM Arrays", Proceedings of International Symposium on Quality Electronic Design (ISQED), pp. 185-91, San Jose, CA, USA, March 26-28 2007. Number of citations: 8.

35. M. Björk, M. Själander, L. Svensson, M. Thuresson, J. Hughes, K. Jeppson, J. Karlsson, P. Larsson-Edefors, M. Sheeran, and P. Stenstrom, "Exposed Datapath for Efficient Computing", HiPEAC Workshop on Reconfigurable Computing, Ghent, Belgium, Jan. 28-30 2007. Number of citations: 2.

36. D. Andersson, L. "J" Svensson, and P. Larsson-Edefors, "Interconnect Characterization Flow for Minimal-Segment Model Selection", Proceedings of the Norchip Conference, pp. 53-8, Linköping, Sweden, Nov. 20-21 2006. Number of citations: 2.

37. H. Eriksson, P. Larsson-Edefors, M. Sheeran, M. Själander, D. Johansson and M. Schölin, "Multiplier Reduction Tree with Logarithmic Logic Depth and Regular Connectivity", Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), pp. 5-8, May 21-24 2006. Number of citations: 14.

38. (*) M. Q. Do, M. Drazdziulis, P. Larsson-Edefors, and L. Bengtsson, "Parameterizable Architecture-Level SRAM Power Model Using Circuit-Simulation Backend for Leakage Calibration", Proceedings of International Symposium on Quality Electronic Design (ISQED), pp. 557-63, San Jose, CA, USA, March 27-29 2006. Number of citations: 22.

39. M. Själander, M. Drazdziulis, P. Larsson-Edefors, and H. Eriksson, "A Low-Leakage Twin-Precision Multiplier Using Reconfigurable Power Gating", Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1654-7, Kobe, Japan, May 23-26 2005. Number of citations: 3.

40. D. A. Andersson, L. "J" Svensson, and P. Larsson-Edefors, "Accounting for the Skin Effect during Repeater Insertion", Proceedings of ACM Great Lakes Symposium on VLSI (GLSVLSI), pp. 32-7, Chicago, IL, USA, April 17-19 2005. Number of citations: 3.

41. M. Själander, H. Eriksson, and P. Larsson-Edefors, "An Efficient Twin-Precision Multiplier", Proceedings of IEEE International Conference on Computer Design (ICCD), pp. 30-3, San Jose, CA, USA, Oct. 11-13 2004. Number of citations: 14.

42. M. Drazdziulis, P. Larsson-Edefors, D. Eckerbert, and H. Eriksson, "A Power Cut-Off Technique for Gate Leakage Suppression", Proceedings of European Solid-State Circuits Conference (ESSCIRC), pp. 171-4, Leuven, Belgium, Sept. 20-24 2004. Number of citations: 7.

43. M. Q. Do, P. Larsson-Edefors, and L. Bengtsson, "Table-Based Total Power Consumption Estimation of Memory Arrays for Architects", Proceedings of 14th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS), pp. 869-78, Isle of Santorini, Greece, Sept. 15-17 2004. Number of citations: 3.

44. D. Andersson, L. "J" Svensson, and P. Larsson-Edefors, "On Skin Effect in On-Chip Interconnects", Proceedings of 14th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS), pp. 463-70, Isle of Santorini, Greece, Sept. 15-17 2004. Number of citations: 3.

45. M. Drazdziulis and P. Larsson-Edefors, "Evaluation of Power Cut-Off Techniques in the Presence of Gate Leakage", Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), pp. II 745-8, Vancouver, Canada, May 23-26 2004. Number of citations: 2.

46. H. Eriksson and P. Larsson-Edefors, "Glitch-Conscious Low-Power Design of Arithmetic Circuits", Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), pp. II 281-4, Vancouver, Canada, May 23-26 2004. Number of citations: 7.

47. H. Eriksson and P. Larsson-Edefors, "Dynamic Pass-Transistor Dot Operators for Efficient Parallel-Prefix Adders", Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), pp. II 461-4, Vancouver, Canada, May 23-26 2004. Number of citations: 1.

48. J. Hughes, K. Jeppson, P. Larsson-Edefors, M. Sheeran, P. Stenström, L. "J" Svensson, "FlexSoC: Combining Flexibility and Efficiency in SoC Designs", Proceedings of the Norchip Conference, Riga, Latvia, Nov. 10-11 2003. Number of citations: 9.

49. D. Ciuplys and P. Larsson-Edefors, "On Maximum Current Estimation in CMOS Digital Circuits", Proceedings of the 17th International Conference on VLSI Design, pp. 658-61, Mumbai, India, Jan. 5-9 2004. Number of citations: 1.

50. D. Eckerbert, L. "J" Svensson, and P. Larsson-Edefors, "A Mixed-Mode Delay-Locked Loop Architecture", Proceedings of the IEEE International Conference on Computer Design (ICCD), pp. 261-3, San Jose, California, USA, Oct. 13-15 2003. Number of citations: 7.

51. M. Drazdziulis and P. Larsson-Edefors, "A Gate Leakage Reduction Strategy for Future CMOS Circuits", Proceedings of European Solid-State Circuits Conference (ESSCIRC), pp. 317-20, Lisbon, Portugal, Sept. 16-18 2003. Number of citations: 14.

52. M. Q. Do, L. Bengtsson, and P. Larsson-Edefors, "Models for Power Consumption Estimation in the DSP-PP Simulator", GSPx & ISPC 2003 (The International Signal Processing Conference), Dallas, Texas, USA, March 31-April 3 2003.

53. P. Larsson-Edefors, D. Eckerbert, H. Eriksson, and L. "J" Svensson, "Dual Threshold Voltage Circuits in the Presence of Resistive Interconnects", Proceedings of IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 225-30, Tampa, Florida, USA, Feb. 20-21 2003. Number of citations: 3.

54. M. Q. Do, L. Bengtsson, and P. Larsson-Edefors, "DSP-PP: A Simulator/Estimator of Power Consumption and Performance for Parallel Architectures", Proceedings of Parallel and Distributed Computing and Networks Symposium, pp. 767-72, Innsbruck, Austria, Feb. 10-13 2003. Number of citations: 2.

55. H. Eriksson, T. Henriksson, P. Larsson-Edefors, and C. Svensson, "Full-Custom vs. Standard-Cell Design Flow - An Adder Case Study", Proceedings of Asia South-Pacific Design Automation Conference (ASPDAC), pp. 507-10, Kitakyushu, Japan, Jan. 21-24 2003. Number of citations: 11.

56. D. Eckerbert and P. Larsson-Edefors, "A Deep Submicron Power Estimation Methodology Adaptable to Variations Between Power Characterization and Estimation", Proceedings of Asia South-Pacific Design Automation Conference (ASPDAC), pp. 716-9, Kitakyushu, Japan, Jan. 21-24 2003. Number of citations: 7.

Patents:

57. A. Alvandpour, P. Larsson-Edefors, R. Krishnamurthy, and K. Soumyanath, "Fast Dual-Rail Dynamic Logic Style", U.S. Patent #6,838,910, Intel Corporation, Jan. 4 2005.

58. A. Alvandpour, P. Larsson-Edefors, R. Krishnamurthy, and K. Soumyanath, "Flash [II] - Domino: A Fast Dual-Rail Dynamic Logic Style", U.S. Patent #6,717,441, Intel Corporation, April 6 2004.

Software developed:

FlexCore design framework: `http://www.flexsoc.org/downloads.shtml`.

Previous projects have generated several pieces of software and algorithms that were publicly available, for example, PRIMUS was a technology mapping tool for high-speed CMOS (as it was defined in 1996), with a source code repository of 22,300 C code lines.

Highest-Cited Publications Overall:

"FlexCore: Utilizing Exposed Datapath Control for Efficient Computing", M. Thuresson, M. Själander, M. Björk, L. Svensson, P. Larsson-Edefors, and P. Stenstrom, Journal of Signal Processing Systems, vol. 57, no. 1, pp. 519, Oct. 2009. Number of citations: 24.

"Parameterizable Architecture-Level SRAM Power Model Using Circuit-Simulation Backend for Leakage Calibration", M. Q. Do, M. Drazdziulis, P. Larsson-Edefors, and L. Bengtsson, Proceedings of International Symposium on Quality Electronic Design (ISQED), pp. 557-63, San Jose, CA, USA, March 27-29 2006. Number of citations: 22.

"Separation and Extraction of Short-Circuit Power Consumption in Digital CMOS VLSI Circuits", A. Alvandpour, P. Larsson-Edefors and C. Svensson, Proceedings of International Symposium on Low-Power Electronics and Design (ISLPED), pp. 245-9, USA, Aug. 1998. Number of citations: 20.

"A Leakage-Tolerant Multi-Phase Keeper for Wide Domino Circuits", A. Alvandpour, P. Larsson-Edefors and C. Svensson, Proceedings of IEEE International Conference on Electronics, Circuits and Systems (ICECS), vol. I, pp. 209-12, Greece, Sept. 1999. Number of citations: 18.

"A Gate Leakage Reduction Strategy for Future CMOS Circuits", M. Drazdziulis and P. Larsson-Edefors, Proceedings of European Solid-State Circuits Conference (ESSCIRC), pp. 317-20, Portugal, Sept. 2003. Number of citations: 14

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2011-40480-88494-55

Dnr
2011 -

Name of applicant
McKee, Sally A.

Date of birth
630611-2126

Reg date
2011-04-13 14:33:39

Project title
FlexSoft: Software Infrastructure to Support Hardware/Software Codesign of Exposed Architectures

NT

Project research grant NT 13 April 2011
Project Research Grant
Computer Science
Chalmers tekniska högskola

NT-S

Appendix S

Signatures

2012	2013	2014	2015	2016	
2855	2976	1902	2005	1923	2012-01-01 -- 2016-12-31

Applicant | **Date**

Head of department at host University | **Clarifi cation of signature** | **Telephone**

A signature on the application is required not only from the applicant but also from the authorised representative of the host university/institution or equivalent (normally the head of the department or establishment where the research is to be conducted). The signature confirms that the department can accommodate the proposed research, position or equipment; that the costing in the application is approved for the department's part, that any proposed experimentation on human or animal subjects has been reported, and that the applicant has reported any secondary occupations and commercial ties (s)he may have, and nothing inconsistent with good research practice has thereby emerged. The applicant must have discussed these conditions with the representative of the host university/institution

International Postdoctoral Fellowships are administered by the Swedish Research Council. The only signature required on the application is that of the applicant.

Vetenskapsrådets noteringar
Kod

2011-40480-88494-55

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se