

# Instruction Set Architecture Impact on Design Space Subsetting for Configurable Systems

Mohamad Hammam Alsafrjalani

Department of Electrical and Computer Engineering  
University of Miami  
Coral Gables, FL, USA  
e-mail: Alsafrjalani@miami.edu

Ann Gordon-Ross

Department of Electrical and Computer Engineering  
University of Florida  
Gainesville, FL, USA  
e-mail: Ann@ece.ufl.edu

**Abstract**—A major challenge in design space subsetting is the time consumed in evaluating all of the design space configurations and selecting a subset of best/near-best configurations—those that adhere closely to design constraints. If an architect determines a subset for an instruction set architecture (ISA) that is historically coupled the system's domain, the evaluation process must be repeated if the architect is to consider an alternative ISA, wasting precious design time. In this paper, we compare different ISAs' best configuration subsets and investigate the possibility of selecting an ISA-independent subset that provides good domain constraint adherence regardless of the ISA.

**Keywords**—design space subsetting; configurable caches; design constraints

## I. INTRODUCTION AND MOTIVATION

Architects typically select a single ISA with good potential for design constraint adherence, and then evaluate different system designs with respect to the constraints. Different system designs may contain different combinations/configurations of architectural parameter values, such as clock frequency, voltage setting, cache memory size, associativity, and line size, pipeline depth, etc. that are implemented with configurable hardware (e.g. [2][8][9][10]). To select the *best configuration* that most closely adheres to the design constraints, architects can use various design space exploration methods [4][7][8][12][14] which provide an *ISA-specific* best configuration, or a subset of near best configurations [4][13]. However, if the architect determines that the best configuration(s) does not adhere closely enough to the design constraints and chooses to evaluate a different ISA, the entire design space must be re-explored to select the new ISA-specific best configuration.

Even though there is copious prior work on design space exploration methods, little prior work has compared a *subset of best and near-best* configurations for configurable hardware across different ISAs. If different ISAs have similar best configurations, then this redundant design space exploration effort is not necessary. Due to ISA intrinsic characteristics, it is obvious that the best configurations are likely to be ISA-dependent but if the best configurations are similar, there is potential for selecting ISA-independent *subset of best and near-best* configurations that can be later implemented on configurable hardware. Prior work [13] has shown that the design space contains several near-best

configurations that adhere nearly as well to the design constraints as the best configuration. Thus, a key challenge is for a single ISA, selecting a subset of near-best configurations that are ISA-independent, which will avoid redundant per-ISA design space exploration, and enable architects to quickly evaluate alternative ISAs.

Our work main contribution is determining whether an ISA has an impact on subset configuration selection (i.e. feasibility of selecting an ISA-independent subset), and a methodology to evaluate this impact. We note that although our work uses examples of CISC/x86 and RISC/ARM configurations, our work doesn't aim to perform direct/raw comparison of these implementations.

To evaluate the feasibility of selecting ISA-independent subsets, we develop an intra-ISA evaluation methodology to select *internal* subsets, which are ISA-specific *best* subsets (i.e., a subset that most closely adheres to the design constraints for a given ISA). Then, we perform inter-ISA subset evaluation, which evaluates this internal subset's design constraint adherence when used as an *external* subset for a different ISA. If internal and external subsets have similar design constraint adherence, then the ISA has little/no impact on the best subsets, and thus subsets can be ISA independent. Our results revealed that the ISA's impact on subset configurations varies based on the design constraint, but even with these variations external subsets adhere to design constraints nearly as well as internal subsets, with an average difference of 1.50% for performance and 4.65% for energy constraints.

## II. INTER-ISA DESIGN SPACE EXPLORATION METHODOLOGY

To evaluate as ISA's impact on subset selection, internal and external subset selection must be performed similarly across all ISAs before these subsets are compared for design constraints adherence. This section details ISA-specific subset selection and evaluation, and inter-ISA subset evaluation methodologies.

### A. ISA-specific Subset Selection Methodology

To maximize a subset's adherence to design constraints—a *high quality* subset—subsets must contain near-best configurations. The highest quality subset is the *best* subset. Since the number of near-best configurations likely varies across ISAs, ISA-specific best subset sizes are

likely to vary across different ISAs. However, in order to accurately and fairly evaluate the ISA’s impact on subset selection, and whether external and internal subsets have similar adherence to design constraints, the subset size must be fixed across all ISAs. Since our approach evaluates the ISA’s impact on subset selection rather than subset size determination, for the scope of this work, we can assume that the subset size is predetermined.

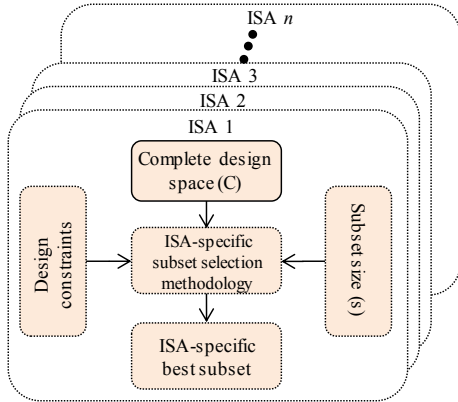


Figure 1. Intra-ISA design space exploration methodology.

Given the subset size, our intra-ISA evaluation methodology determines the best subset’s constituent configurations for each ISA. In Figure 1, for each considered ISA, our ISA-specific subset selection methodology takes as input the design constraints, the complete design space  $C$ , and the subset size  $s$ , and outputs the ISA-specific best subset.

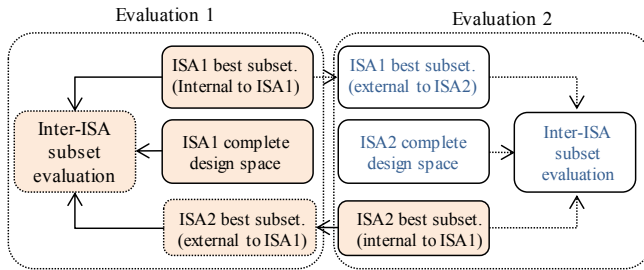


Figure 2. Example of inter-ISA evaluation.

To select the best subset’s configurations, the ISA-specific selection methodology evaluates every configuration combination  $c$  of size  $s$  out of the complete design space  $C$ . To evaluate each configuration combination  $c$ , the methodology executes the benchmarks on all configurations in  $c$ . For each benchmark, the method measures the design constraint’s value (e.g., energy consumption, execution time for performance, etc.), determines the configuration with the best design constraint adherence (e.g., lowest energy), and compares this configuration’s design constraint value with the benchmark’s best configuration in the complete design space  $C$ . The subset selection methodology averages this difference over all benchmarks, and the configuration combination  $c$  with the lowest average offers the closest

adherence to the design constraints and thus, is the ISA-specific best subset. We note that whereas our experiments exhaustively evaluated all possible subsets, to speed up subset selection, the ISA-specific subset selection methodology can leverage design space exploration heuristics such as [1][4][7][8][13].

### B. Inter-ISA Subset Evaluation Methodology

Once all ISA-specific best subsets are selected, our approach performs inter-ISA evaluation by calculating, for each ISA, the external subsets’ design constraint adherence to the considered ISA’s complete design space and internal subset. Even though our inter-ISA subset evaluation methodology can evaluate an arbitrary number of external subsets, Figure 2 illustrates our methodology using two ISAs. Since Evaluation 1 and evaluation 2 are similar, we detail Evaluation 1. Evaluation 1 evaluates ISA2’s best subset’s design constraint adherence when used as an external subset to ISA1. To perform this evaluation, our inter-ISA subset evaluation methodology evaluates the external subset as compared to ISA1’s complete design space and best subset (internal to ISA1).

To evaluate the external subset to the complete design space, our inter-ISA evaluation methodology executes all of the benchmarks in all of the configurations in ISA2’s best subset. For each benchmark, the methodology determines the configuration with the best design constraint adherence (best design constraint value) and compares this configuration’s design constraint value with the benchmark’s best configuration in ISA1’s complete design space. Similarly, to evaluate using an external subset as opposed to using an internal subset, our inter-ISA subset evaluation methodology compares for each benchmark the external and internal subsets’ configurations with the best design constraint adherence.

## III. EXPERIMENTAL SETUP

To evaluate contemporary, ubiquitous ISAs, we defined our design space, design constraints, and subset size to reflect design options for contemporary processors. To compare our approach to prior design space exploration work, we evaluated design constraint adherence for energy and performance using similar benchmarks [3][4][13].

We determined realistic design constraints by exhaustively simulating all benchmarks from the complete EEMBC Autobench benchmark suite [6]. To eliminate imprecision due to measuring tools and ensure accurate evaluation, we executed all binaries using the cycle-accurate simulator gem5 [5] to obtain performance values, and McPAT [11] to obtain energy values. Whereas selecting our ISAs as AMD, Alpha, ARMv7, ARMv8, ARM-Thumb2, x86, and MIPS provides many-to-many comparison, we selected the two common, and highly disparate, computing domain ISAs available to our simulator—x86 and ARM<sup>1</sup>.

We compiled all benchmarks with the default gcc parameters x86 and ARM. We set each benchmark’s design

<sup>1</sup> Three benchmarks did not cross-compile and were excluded from the experiment.

constraints as the lowest energy consumption and highest performance in the complete design space.

We focused on the memory hierarchy, since the memory hierarchy has a large impact on energy and performance. Since our benchmarks are small embedded system compute kernels that have small cache requirements, our design space included appropriately-sized level one (L1) cache configurations [1][3], however, our work is easily adaptable to any L1 cache configuration space, as well as level two caches and benchmarks with larger cache requirements. The L1 cache sizes ranging from 2 KB to 8 KB, associativities ranging from direct-mapped (1-way) to 4-way, and line sizes ranging from 16 B to 64 B, each in power-of-two increments; 27 total. Subset size was fixed to two and four configurations for performance and energy constraints, respectively.

We evaluated our intra-ISA design space exploration and inter-ISA subset evaluation methodologies for each design constraint separately, however our approach could be extended to jointly consider multiple design constraints to determine Pareto optimal subsets. Without loss of generality, we assumed that both L1 caches connect directly to main memory, and thus there was no dependency between the L1 instruction and data caches and these caches could be independently evaluated. Since close adherence to design constraints is achievable by varying only one microarchitectural parameter [3][4], we evaluated only the ISA’s impact on L1 cache subset selection; we set all other system parameter values (microarchitectural, die factors, fabric, etc.) to the same value for both ISAs, and these values reflect contemporary processor values: technology was 40nm and core operated at 3.1V, a 2.0 GHz clock frequency, and was out of order issue; and the reorder buffer size was fixed to 128 and 80, for x86 and ARM, respectively. Additionally, the pipeline depth was fixed to 31 and 10 for x86 and ARM, respectively.

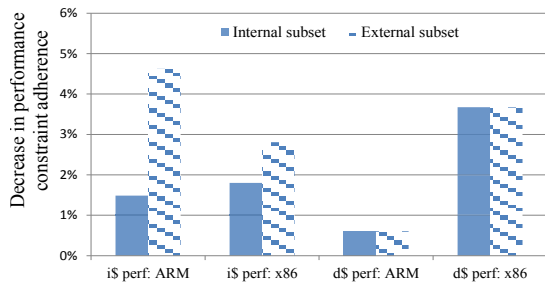


Figure 3. Inter-ISA performance constraint evaluation.

#### IV. RESULTS AND ANALYSIS

Figure 4 and Figure 3 analyze the energy and performance inter-ISA design constraint adherence for both ISAs and for the instruction cache (iS) and data cache (dS). We evaluated a subset’s energy/performance design constraint adherence using both internal and external subsets by normalizing the subset’s energy/performance to the complete design space’s design constraint adherence. The bar values represent the design constraint’s adherence, and higher values represent

less design constraint adherence, and correspond to lower quality subsets. External subsets vs. the complete design space

The results in Figure 4 and Figure 3 show that for both design constraints, using internal subsets degrades the design constraint adherence by a small amount, ranging from no degradation (dS energy for ARM) to less than 4% (dS performance for x86). In all cases, using external subsets increases this degradation, as expected, ranging from less than 1% (dS performance for ARM) to 9.5% (iS energy for ARM).

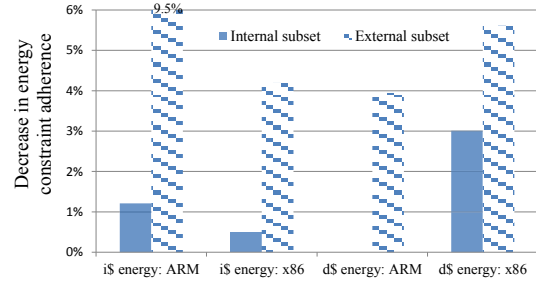


Figure 4. Inter-ISA energy constraint evaluation.

The results also show that this degradation affected energy adherence more so than performance adherence—4.65% degradation as compared to 1.04%, respectively—thus the effects of using ISA-independent subsets is design-constraint dependent. However, considering that the maximum degradation varied between 9.5% and 5% as compared to the complete design space for energy and performance constraints, respectively, and considering that these results are for an extreme example with vastly disparate ISAs, it is feasible for architects to quickly evaluate disparate ISAs using a single ISA, especially for systems with relaxed constraints [1]. This means architects can rapidly evaluate one subset for multiple ISAs (ISA impact is irrelevant) knowing that these subsets provide design constraint adherence within 5% of performance constraints. Similarly, for systems with relaxed energy constraints but a critically short time to market, architects can still evaluate system designs on different ISAs, such that energy consumption is within 10% of energy constraints. However, we also note that for systems with strict constraints (e.g., hard real time systems), ISA impact limits the feasibility of evaluating subsets for different ISAs.

TABLE I. NORMALIZED DESIGN CONSTRAINTS ADHERENCE OF EXTERNAL SUBSETS TO INTERNAL SUBSETS

		ARM	x86
Energy	iS	8.32%	3.73%
	dS	3.94%	2.61%
Performance	iS	3.00%	1.15%
	dS	0.00%	0.00%

##### A. External vs. Internal Subsets

To directly evaluate the difference in design constraint adherence for external and internal subsets, we normalized

the external subsets' design constraint adherences to the internal subsets' design constraint adherences for energy and performance constraints. For brevity, we summarize these values in TABLE I.

For energy consumption, using the data cache's external subset degrades the energy constraint adherence by 3.94% and 2.61% for ARM using x86 subsets and for x86 using ARM subsets, respectively. Similarly, using the instruction cache's external subset degraded the energy constraint adherence by 8.32% and 3.73%, for ARM using x86 subsets and for x86 using ARM subsets, respectively. For the instruction cache, the ARM external subset decreased adherence to energy constraints greater than x86 external subsets. Since x86 uses fewer instructions, it is likely that x86 subsets will be comprised of configurations with smaller parameter values as compared to the ARM subsets. Even though smaller cache parameter values result in less energy consumption [14], parameter values smaller than the benchmark's specific requirements potentially increase the miss rate, which increases the cache's idle energy. This increase is compounded by the fact that decreasing technology size ( $\leq 40\text{nm}$ ) increases the idle energy's contribution to total energy consumption [3]. However, the energy constraint adherence degradation is less apparent in the data cache since, regardless of the ISA, the data access patterns are similar, thus subsets are more ISA independent. One method to enable more detailed analysis about instruction cache energy constraint depreciation is to run experiments using our methodology for idle and dynamic energy separately.

As for performance evaluation, using the data cache's external subset performed identically to the internal subset. Since the benchmarks exhibit similar ISA-independent data locality, both ISA's best subsets contained the same configurations. However, for the instruction cache, x86 using ARM subsets degraded the performance by 3.00% and ARM using x86 subsets degraded the performance by 1.15%. This performance-loss disparity can be attributed to the code density variance between ARM (a reduced instruction set architecture (RISC) with simpler but more instructions) and x86 (a complex instruction set architecture (CISC) with more complex but fewer instructions) ISAs. Since x86 uses fewer instructions, it is likely that x86 subsets will be comprised of configurations with smaller parameter values as compared to ARM subsets. However, to alleviate this instruction complexity disparity's impact on external and internal subset design constraint adherence, one possible method is to formulate a model that considers the number of instructions and access pattern for a given benchmark (e.g., using a trace), and relationship between CISC and RISC ISAs (e.g., differentiating between machine codes) to select subsets of similar configurations on different ISAs.

Additionally, this result suggests a similarity between configurations comprising external and internal subsets. However, with a much larger design space, the similarity of these configurations is likely to lessen. Intuitively, designers must increase the subset size to increase the probability of having similar configurations. However, increasing subset sizes increases design time and efforts. To evaluate this divergence with respect to design space size and subset size,

we plan to study larger design spaces with variable subset sizes.

## V. CONCLUSION AND FUTURE WORK

In this work, we evaluated this similarity by comparing the ISA's impact on subsets containing several near-best configurations. Our results revealed that even though ISAs have an impact on subset selection, this impact is minor. Using a subset that was selected for a different ISA only degraded average performance and energy by 1.50% and 4.65%, respectively. Based on design constraint stringency, this degradation may be acceptable, thus architects can feasibly use ISA-independent subsets to quickly explore different ISA options.

Given the various implementation options (e.g., number of cores: 1, 2, 4, etc.), mapping the configurations of the ISA-independent subset is a complex task. Our future work will study the tradeoffs of implementing the ISA-independent subset to configurable processors with disparate number of cores and architect a methodology that maps the configurations based on the number of available configurable cores such that the overhead is minimized.

## REFERENCES

- [1] Adegbiya, T.; Gordon-Ross, A.; "Energy-efficient phase-based cache tuning for multimedia applications in embedded systems," IEEE Consumer Communications and Networking Conference. Jan. 2014
- [2] Albonesi, D.H., "Dynamic IPC/clock rate optimization," Proc. of Int. Sym. on Computer Architecture, Jul. 1998.
- [3] Alsafrjalani, M. H.; Gordon-Ross, A.; "Dynamic Scheduling for Reduced Energy in Configuration-Subsetted Heterogeneous Multicore Systems," Int. Conf. on Embedded and Ubiquitous Computing. 2014
- [4] Alsafrjalani, M. H.; Gordon-Ross, A.; and Viana, P. "Minimum Effort Design Space Subsetting for Configurable Caches," Int. Conf. on Embedded and Ubiquitous Computing. Aug. 2014
- [5] Binkert, N.; Beckmann, B.; Black, G.; et al. "The gem5 simulator," SIGARCH Comput. Archit. News 39, 2 (August 2011), 1-7.
- [6] EEMBC. The Embedded Microprocessor Benchmark Consortium [http://www.eembc.org/benchmark/automotive\\_sl.php](http://www.eembc.org/benchmark/automotive_sl.php), Sept. 2013
- [7] Ghosh, A.; Givargis, T.; "Cache optimization for embedded processor cores: an analytical approach," Int. Conf. on Computer Aided Design, Nov. 2003
- [8] Gordon-Ross, A., Vahid, F.; "Fast configurable-cache tuning with a unified second-level cache. In Proc. Of 2005 Int. Symp. on Low Power Electronics and Design
- [9] Ishihara, T.; Yasuura, H., "Voltage scheduling problem for dynamically variable voltage processors," Proc. on Int. Sym. on Low Power Electronics and Design, Aug. 1998.
- [10] Jejurikar, R.; Pereira, C.; Gupta, R., "Leakage aware dynamic voltage scaling for real-time embedded systems," Proc. on 41st Design Automation Conference, Jul. 2004.
- [11] Li, S.; Ahn, Jung Ho; Strong, R.D.; et al. "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," 42<sup>nd</sup> Annual IEEE/ACM Int. Symp. on Microarchitecture, Dec. 2009
- [12] Palesi, M., Givargis, T., "Multi-objective design space exploration using genetic algorithms," Hardware/ Software Codesign, Proc. of 10th Int. Symp. on, 2002.
- [13] Viana, P.; Gordon-Ross, A.; Keogh, E.; Barros, E.; Vahid, F.; "Configurable cache subsetting for fast cache tuning," 43<sup>rd</sup> ACM/IEEE Design Automation Conference, 2006.
- [14] Zhang, C., Vahid, F., "Cache configuration exploration on prototyping platforms," 14th IEEE Int. Workshop on Rapid Systems Prototyping, 2003.