DIVA-EMIN: Efficient Dependability for Post-Silicon Materials

Oliver Kaltstein † Shinya Takamaeda-Yamazaki † Jun Yao † Yasuhiko Nakashima †

Abstract

Microprocessors for new materials (e.g. IGZO thin film) are required to have a small amount of transistors, so that they can fit onto the material. At the same time due to the sensitivity of film-material more redundancy is necessary to avoid early system-failure. In this paper, we propose DIVA-EMIN, a novel method to provide a high dependability to keep the circuit smaller than in a straightforward Dual Modular Redundancy (DMR) implementation. In order to reduce the circuit overheads, infrequently used parts of the processor have been isolated from the additional system for the dependability. DIVA-EMIN reduces the size of a DMR-module to half of the surface while degrading dependability by only 10%.

1 Introduction

Instead of silicon, new material devices using amorphous oxide semiconductors, such as indium-gallium-zinc oxide (IGZO)[1], will become very cheap if mass-produced. This characteristic makes it possible to create an ultra low-power and cheap embedded computer system.

Unfortunately, the maximum number of available transistors is limited to a few 1000 transistors on the new materials. If such material is very cheap, it may be much less protected against scratches, because the modern dependable technique might not be adopted. Additionally chemical effects will make it become unreliable even faster than the silicon-based devices.

We already proposed the small footprint processor architecture for emerging materials, named EMIN[2]. EMIN is designed for directly executing 32-bit ARM programs by using emulation techniques on an 8-bit architecture. By employing the emulation approach, complex logics that consume very large circuits are realized as software and stored in memory. However, EMIN does not provide any features to detect an error at runtime.

Commonly, in order to provide dependability, DMR (Dual Modular Redundancy) is used to detect an error that appeared on a system. Such straightforward DMR that appends an additional copy of the protected component requires a huge number of transistors.

In this paper, we propose DIVA-EMIN, a novel technique to provide high dependability for EMIN using emerging cheap but unreliable materials. Instead of DMR, we choose DIVA for the error detection with small circuit footprints. In contrast to the original DIVA, our approach is to reduce the memory footprint for emulator instructions. We found that only some portions of the emulator are heavily utilized in DMR. Portions that are not used frequently are kept non redundant to reduce the amount of used transistors.

†Nara Institute of Science and Technology

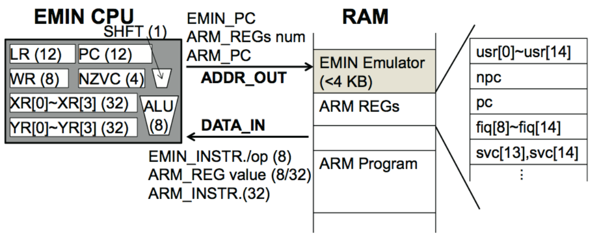

Figure 1: EMIN

2 Architectural Techniques for Tiny Processors on Post-Silicon Materials

2.1 EMIN: Emulation Oriented Small Processor

EMIN is an optimized processor architecture executing the ARM emulator to execute ARM binaries. Figure 1 shows the architecture of EMIN. EMIN is an 8-bit processor with 8 data-lines to memory, however, it has optimizations to handle 32-bit data and addresses, so up to 32 address-lines can be implemented. There are further optimizations for running emulator-software that emulates the ARMv4T processor. Even though it is an 8-bit processor it has specific mechanisms to easily load 32 bit instructions in 4 steps from memory into the processor-registers. There are optimizations for addressing as well by using the WR-Register to cache the working address anytime while loading the Y-Registers. The instruction-set balances the performance with the circuit-area. Instructions that are not used often in ARM programs, such as shifting, are emulated in a more compact but slower way by iterations.

An advantage of having an ARM emulator is that ARM programs can be run directly without disassembling. Source-code is often unavailable, however, ARM is a popular system. EMIN can be used as a replacement for existing systems with ARM processors. In this way micro-controllers can be migrated to much cheaper thin-film materials than printed circuit boards. Low-cost controllers are a great advantage in sensor-networks to help monitor the environment. ARM is the de-facto standard for mobile devices[3]. A tiny processor-system will allow better reliability during use and a higher yield during production.

This approach avoids 32-bit hardware and therefore saves space - space is an issue on new materials where space for transistors or cabling is very limited. However, the complexity of the CPU (i.e., the circuit area) and the size of the emulator (i.e., performance) are in a trade-off relationship. Another reason for using an emulator is that it is easier to protect the logic of memory for example by ECC than to protect the integrity of a processor.

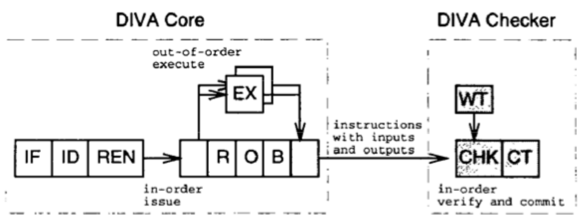

Figure 2: DIVA

2.2 DIVA

Dynamic Verification DIVA[4)] was originally designed as a dependability-solution for submicron-system to overcome low signal-quality and radiation to avoid logic-failure inside a silicon-chip. If integrations are very high, dependability decreases in silicon-systems as well as in film-computers. Even though the reasons for the reliability decrease are different, the approach to correct errors is the same one. The left side: "DIVA Core" shown in Figure 2 is the complete processor with high integration, and possibly smaller transistors. The DIVA-checker on the right is a reduced core to do only the most frequent calculations. Whenever the DIVA-checker agrees with the full core then the results will be committed. If a more specific instruction from the full core is not implemented in the DIVA-checker, then it cannot be compared and will be committed without verification. The result from the main-core will be copied to the checker so that further instructions that depend on the missing steps can be continued processing. DIVA needs only small space for the checker-unit, because it is a stripped down processor where many instructions have been reduced.

This concept is also useful for film-based computers, because space on thin-film is limited. Because dual modular redundancy (DMR) would consume too much space it couldn't be implemented. Using DIVA in combination with EMIN can save a lot of space, and still leads to a reliable overall-performance, close to the reliability of dual modular redundancy. However, since EMIN is not a pipeline-architecture out-of-order execution is not an issue and the implementation of the DIVA-EMIN system is actually simpler.

3 DIVA-EMIN

3.1 System Architecture

The important characteristic of EMIN is that a memory component storing the emulator program requires larger hardware resources than the EMIN processor: the memory for the emulator consumes 20 times more transistors than the EMIN processor[3)]. This research focuses on reducing the size of the memory while keeping the dependability almost as high as in a conventional dual modular redundancy (DMR) system.

We propose DIVA-EMIN to make the best use of materials that are limited in size and that degrade faster than silicon. Our study involves a combination of three technologies: new materials, EMIN and DIVA. Figure 3 shows the DIVA-EMIN system. Both sides have a complete EMIN processor. The processor is already about

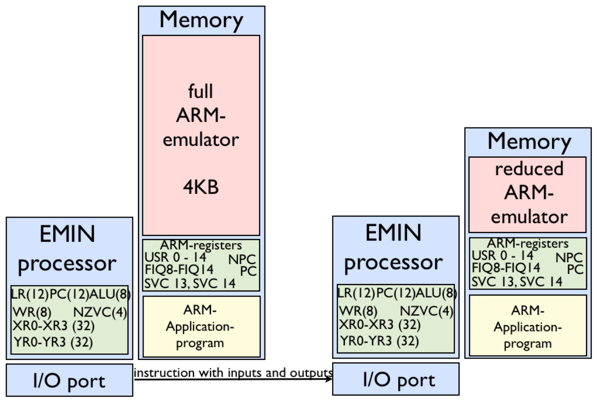

Figure 3: DIVA-EMIN

20 times smaller than memory. The left side shows the full ARM emulator, which is 4KB big. On the right side it has the exact same system, but a smaller memory is used to save memory space. The reduced ARM emulator on the right functions as a DIVA checker.

In our experiments, we are using an arbitrary number for the size of the checker-unit. This number symbolizes the remaining space available on film-material that we want to make best use of. For example 70%, so it may be not enough to fit a complete processor with the emulator as dual modular redundancy.

DIVA-EMIN avoids slowing down the system, because the full core is usually slower than the DIVA checker. However, some dependability will be sacrificed while always keeping the execution-speed of ARM binaries to the maximum. This is accomplished by sacrificing some dependability. The general idea is to keep the system redundant more than 90% of the time while using only half of the area. Whether this can be accomplished or not depends on the software and the data that is being executed on such an optimized system. Our DIVA-EMIN study is about finding the best possible optimization for reducing the size of the DIVA checker.

Both systems need to be connected to ensure that the right side can do the checking and committing of the calculations that are performed on the left side of the system in Figure 3. While there are various ways to establish such a communication-link, discussing the pros and cons of the interconnection is not included in this paper. The connection between the systems is necessary to send every result from the left processor to the right processor. Results computed in the DIVA-checker can be compared with the results of the full-core, or the results of the full core can be used to replace missing results from the checker-unit.

3.2 Adaptive Instruction Selection

In EMIN, an ARM program is executed by the software emulator of ARM architecture running on the native 8-bit processor that is realized as actual hardware. The ARM emulator has a decoder for converting an ARM instruction into a native instruction sequence. Figure 4 shows the memory size breakdown of each function region of the ARM emulator decoder. The emulator consists of a 3-level hierarchy of function regions to de-

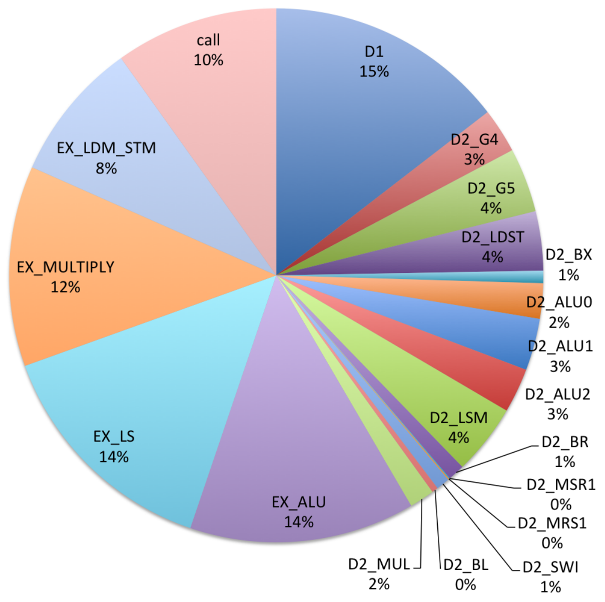

Figure 4: Breakdown of Occupied Memory Region for Decoding ARM Instructions

code an ARM instruction into a native 8-bit instruction sequence: each function level is represented by a prefix named *D1*, *D2* and *EX*, respectively. Every instruction begins at D1_START. After exiting D1, one of the D2-regions will be entered depending on which instruction is being executed. Whenever D2_MUL is entered, then EX_MULTIPLY is entered with the same frequency.

There are some variations in size of a function region. In DIVA-EMIN, removing a decode function consuming large memory can effectively reduce the consumed transistors for instruction memory of the DIVA checker. However, just removing them might decrease the dependability, if those parts are frequently executed. When the DIVA checker has no corresponding part of the original decoder, it just receives and uses the computation result from the corresponding instructions and cannot validate results by those instructions.

In order to reduce the memory consumption for the emulation program with dependability, infrequently executed instructions are the first candidate to be removed from the DIVA checker. To determine which parts are removed, we analyzed the actual frequency of use for each function region of the emulation decoder by using a software simulator of EMIN. Figure 5 shows the frequency of use for each function region. The numbers on top show how many times the multiply instruction was used. We used Stanford Integer Benchmarks[5)] as the benchmark.

The result shows that the frequency of usage strongly depends on the behavior of each program. Note that we don't distinguish which part of EX_MULTIPLY was utilized, since the chosen granularity was coarse-grained. Therefore EX_MULTIPLY and D2_MUL appear in the same length as a percentage in this graph. The result shows that every program makes some significant use of the function regions of the ALU operation. A benchmark Intmm makes heavy use of the function region of

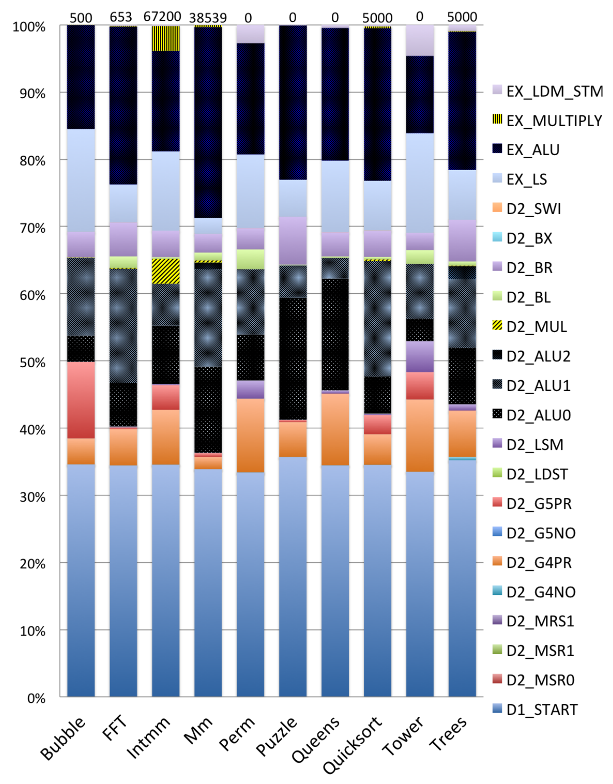

Figure 5: Breakdown of Frequency of Use of Decoder Function.

multiplication operation, while Tower, Queens, Puzzle and Perm make no use of it. Bubble and FFT make some use of the function regions of the multiplication operation.

These hints help us to make decisions which function regions we should implement as single-module (only in the full EMIN) and which function regions we implement as dual-modular redundancy in the full-EMIN and the DIVA-checker. It's revealed that removed instructions should be selected and customized based on the actual frequency of use for each application, in order to efficiently reduce the occupied memory capacity with acceptable dependability.

To determine which parts of function regions are removed from the DIVA checker, we now define a value of priority that indicates the importance of each function region using its occupied memory size and frequency of use, as following. A function region with a low priority is potentially removed from the DIVA checker.

$$Priority = \frac{Freq[times]}{OccupiedSize[Byte]} \qquad (1)$$

4 Evaluation

We evaluate DIVA-EMIN in hardware size and dependability. Dependability is defined as the cumulative number of covered instructions from the DIVA checker. If no instruction is removed from the DIVA checker, the dependability is perfect in this evaluation. To analyze the relationship between memory size reduction and dependability, we removed some function regions for decoding of ARM instructions from the DIVA checker based on its priority, and then we calculated the number of covered instructions by the DIVA checker. The number of covered instructions of the DIVA checker is calculated from the instruction mix of each benchmark measured by using the software simulator of EMIN and the correspondence relationship between an ARM instruction and used function regions. As well as the previous preliminary evaluation, we used Stanford integer benchmark as the benchmark.

Figure 6 shows the relationship between dependability and occupied memory size of the DIVA checker for various programs. Overall, the DIVA-EMIN system can achieve substantial reliability with reducing memory consumption by the instruction decoder in DIVA checker by preferentially selecting function regions with low priority (high memory consumption or low frequency of usage). In the benchmark Puzzle, the system can achieve 95% reliability of the original DMR, while the ARM emulator in the DIVA checker can be optimized in memory usage to 50% of the original.

The selection of removed function regions for decoding ARM instructions is optimized for each ARM program, because each program has different behavior. However, another application might be executed on the optimized emulator for the other. Dependability of a mismatch combination will be lower than an execution on the appropriate emulator. We evaluate the dependability in case the selection is unfortunately optimized for the other application.

Figure 7 shows the relationship between dependability and the occupied memory size both in the appropriate situation and mismatch situation. If reduction is done arbitrarily there can be up to 90% difference in dependability. For example, FFT can function with 60% of the function regions and still be 99% dependable, if reduction was done systematically, because 99% of the time is spent in function regions that exist on both the main processor and the DIVA checker. On the other hand, if the improper function regions of the DIVA checker are removed then the dependability can be as bad as 5% with the same size of the DIVA checker. Therefore an optimized reduction-procedure can help making better use than intuitively deciding while manually selecting the function regions.

5 Related Work

There are various prior researches and techniques on system-level dependability improvement. The major approach employing dual-modular redundancy is Lock-Step[6, 7]. In Lock-Step, two same components are tightly connected and execute an identical application. While the granularity of error detection depends on the system, computation results generated from these two components are validated by an additional comparator.

In multicore processor era, dependability aware architectures have been proposed. Slipstream processor[8] is a mechanism for improving dependability and performance based on multicore processor redundancy. In the slipstream processor, one of multiple cores executes

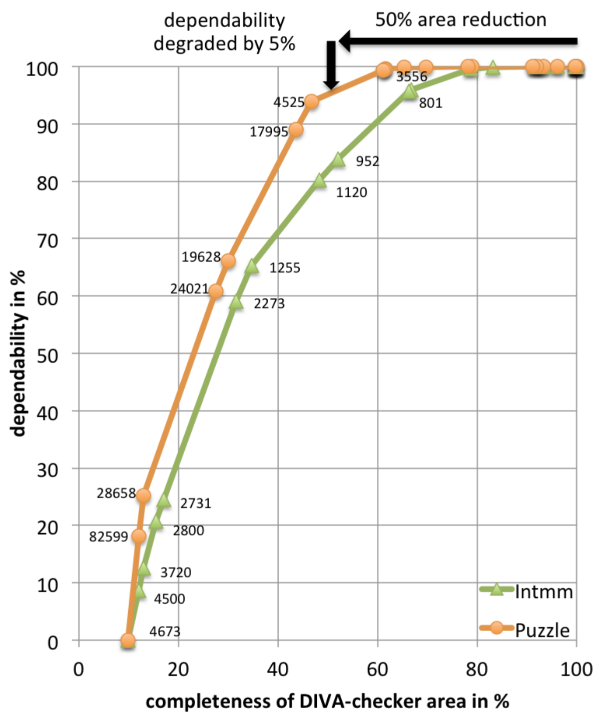

Figure 6: Relationship between Dependability and Hardware Size

a proper instruction sequence, and simultaneously another core executes a short sequence with the same behavior as the proper sequence. This mechanism requires a very low latency communication mechanism for core-core synchronization. Configurable isolation[9] is a low-cost approach providing loose Lock-Step capability for modern multicore processors. Configurable isolation enables to make DMR sets or TMR sets on multicore processors by employing separation of the on-chip interconnections into multiple parts for avoiding unnecessary error propagations. SmartCore[10] is a flexible approach for DMR execution on many-core processors with on-chip network supports. In SmartCore, a DMR pair of processor cores on a single many-core processor system executes an identical thread. Eventually the cores generate data transfers, such as cache miss and DMA, then the on-chip router waits and compares the corresponding contents of the transfer to validate that the execution was correct.

The ideas of these prior researches are useful, but the available resources and situations are very different. In processors on post-silicon materials, available transistors are limited. Therefore, architects have to make a reliable system with minimal transistor usage both in logic and memory.

6 Conclusion

In this paper, we proposed DIVA-EMIN, a lightweight approach for improving dependability of microprocessors on post-silicon materials. To reduce the transistor consumption, we employ EMIN, an emulation-oriented tiny processor that can execute a 32-bit ARM programs by emulation. Our proposed technique provides dependability improvements with small area overhead. Our ap-

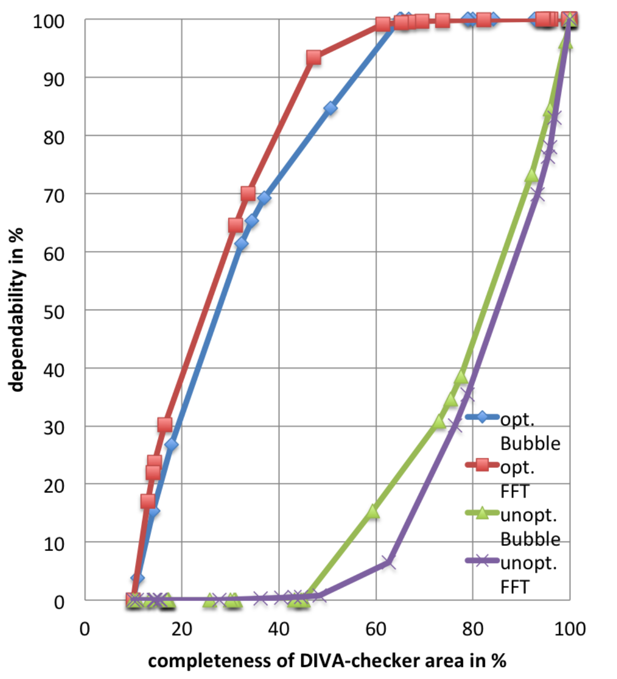

Figure 7: Relationship between Dependability and Hardware Size (Unoptimized vs. Optimized)

proach enables to reduce the memory footprint, consuming most of available transistors in EMIN processor, by adaptive instruction selection based on the analysis of program behavior. The evaluation result shows that the hardware size of an additional unit for DMR execution can be reduced to half of the original with low dependability decreases from the perfect DMR.

Acknowledgement

This work is supported in part by JSPS KAKENHI KIBAN-A (24240005) and HOUGA (24650020).

References

1) Kenji Nomura, Hiromichi Ohta, Akihiro Takagi, Toshio Kamiya, Masahiro Hirano, and Hideo Hosono. Room-temperature fabrication of transparent flexible thin-film transistors using amorphous oxide semiconductors. *Nature*, 432(7016):488–492, 2004.

2) Yasuhiko Nakashima. A Study of Emulator Oriented Small CPU for Realizing Film Computers. *IEICE Technical Report. Computer Systems*, 112(173):19–24, 2012.

3) Yuko Hara-Azumi, Masaya Kunimoto, and Yasuhiko Nakashima. Emulator-oriented tiny processors for unreliable post-silicon devices: A case study. In *Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific*, pages 85–90, Jan 2014.

4) T.M. Austin. Diva: a reliable substrate for deep submicron microarchitecture design. In *Microarchitecture, 1999. MICRO-32. Proceedings. 32nd Annual International Symposium on*, pages 196–207, 1999.

5) John Hennessy and Peter Nye. Stanford integer benchmarks. *Personal communication*, 1988.

6) PowerPC 750GX Lockstep Facility. IBM Application Note, 2008.

7) Spartan-6 FPGA Dual-Lockstep MicroBlaze Processor with Isolation Design Flow. *Xilinx Application Note*, 584, 2012.

8) Karthik Sundaramoorthy, Zach Purser, and Eric Rotenberg. Slipstream processors: Improving both Performance and Fault Tolerance. *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, 35(11):257–268, 2000.

9) Nidhi Aggarwal, Parthasarathy Ranganathan, Norman P. Jouppi, and James E. Smith. Configurable Isolation: Building High Availability Systems with Commodity Multi-core Processors. In *ISCA '07: Proceedings of the 34th annual International Symposium on Computer Architecture*, pages 470–481, 2007.

10) Shinya Takamaeda, Shimpei Sato, Takefumi Miyoshi, and Kise Kise. SmartCore System for Dependable Many-Core Processor with Multifunction Routers. In *International Conference on Networking and Computing (ICNC2010)*, pages 133 –139, nov. 2010.