

**An Embedded Microprocessor Radiation Hardened by
Microarchitecture and Circuits**

| | |
|------------------|---|
| Journal: | <i>Transactions on Computers</i> |
| Manuscript ID: | TC-2014-06-0440.R1 |
| Manuscript Type: | Regular |
| Keywords: | B.1.3.c Redundant design < B.1.3 Control Structure Reliability, Testing, and Fault-Tolerance < B.1 Control Structures and Microprogramming < B Hardware, B.2.3 Reliability, Testing, and Fault-Tolerance < B.2 Arithmetic and Logic Structures < B Hardware, B.3.1.c SRAM < B.3.1 Semiconductor Memories < B.3 Memory Structures < B Hardware, B.3.2.b Cache memories < B.3.2 Design Styles < B.3 Memory Structures < B Hardware, B.3.4.b Error-checking < B.3.4 Reliability, Testing, and Fault-Tolerance < B.3 Memory Structures < B Hardware, B.6.2.c Redundant design < B.6.2 Reliability and Testing < B.6 Logic Design < B Hardware, B.6.3.a Automatic synthesis < B.6.3 Design Aids < B.6 Logic Design < B Hardware, B.7.1.f Microprocessors and microcomputers < B.7.1 Types and Design Styles < B.7 Integrated Circuits < B Hardware, C.0.c Instruction set design < C.0 General < C Computer Systems Organization |
| | |

An Embedded Microprocessor Radiation Hardened by Microarchitecture and Circuits

Lawrence T. Clark, *Sr. Member, IEEE*, Dan W. Patterson, Chandarasekaran Ramamurthy and Keith E. Holbert, *Sr. Member, IEEE*

Abstract—A radiation hardened by design embedded microprocessor is presented. The design uses multiple approaches to minimize the performance reduction from hardening, while simultaneously limiting the power increase. The speculative portions of the pipeline are protected by microarchitecture approaches, i.e., the speculative pipeline is dual redundant, whereby instructions that have errors in one copy cause a pipeline restart—only matching results commit to architectural state. The register file is dual redundant with mechanisms for correction using one copy whose parity is correct. The data cache memory is write-through, allowing protection with parity. The remaining architectural state is protected via hardened circuits. These are implemented with self-correcting triple mode redundant (TMR) flip-flops and TMR logic. The design, implemented here on a 90-nm bulk CMOS process, achieves unprecedented single event effects hardness and 400+ MHz operating frequency at less than 500 mW power consumption. The main constituent circuit hardening approaches have been fabricated and tested separately. Broad beam testing of the constituent circuits has resulted in no uncorrectable soft errors below 100 MeV-cm²/mg LET_{EFF}. We describe the CAD flows used to ensure node separation to achieve high immunity to multiple node charge collection and discuss the relative costs of the chosen hardening techniques.

Index Terms—Radiation hardening by design (RHBD); microprocessor; cache memory; total ionizing dose; single event transients, soft error mitigation, single-event effects.

1 INTRODUCTION

RADIATION hardened microprocessors intended for aerospace applications [1] have always lagged, but more recently have fallen considerably behind commercially available (unhardened) designs as measured by both performance and power dissipation. The RAD750, implemented on a 250-nm radiation hardened process operated at 133 MHz [2]. A more recent version, implemented on a 150-nm hardened process achieves 200 MHz [3]. Hardening by process has used SOI for reduced charge collection [4] and passive elements for single event upset (SEU) immunity [5][6]. The passives are extremely effective since they provide hardening without adding charge collection nodes to the circuit. The processor performance lag is in part due to the difficulty in keeping such rad-hard processes up to date, as they support relatively low volume devices [7].

Radiation hardening by design (RHBD) [7][8] allows the use of less expensive, more up to date process technologies, but puts the onus of achieving hardness solely on the designers. The promise of RHBD is primarily much better power and performance by migrating to smaller technology nodes. However, recent publications describing RHBD processors demonstrated relatively low performance. For example, The SPARC AT697 introduced in 2003 has an operating frequency of 66 MHz, uses triple modular redundancy (TMR) for logic soft error protection, and error detection and correction (EDAC) for

memory soft error protection [9][10]. More recent RHBD processors have reached 125 MHz [11].

In contrast, unhardened embedded microprocessors achieve dramatically better performance on similar processes. For instance, the StrongARM design operates at 160 MHz in a 350-nm process and the 180-nm XScale microprocessor operates at 600 to 900 MHz [11][12]. The 90-nm versions of the XScale microprocessors achieved 1.2 GHz with the cache performance being even higher [13][14]. More modern designs, such as those in 32-nm cell phone system on chip (SOC) devices, are multi-core, out-of-order execution microprocessors, running at over 1.5 GHz [15]. As portable devices have become common, power dissipation has become the overriding concern in microprocessor design. This is equally important for aerospace applications, where thermal issues predominate.

1.1 Radiation Effects

Single event effects (SEEs) are manifest due to an ionizing radiation particle passing through a semiconductor device, which generates electron-hole pairs that may subsequently be collected by the circuit nodes (N and P type source/drains) by both drift and diffusion as shown in Fig. 1(a). The former creates a fast transient and the latter can cause a long transient duration, as charge lingers in the substrate before collection. If the collection is at a combinational circuit node, a temporary voltage glitch, i.e., a single event transient (SET), may be produced, depending on the logic state and diffusion type as shown in Fig. 1(b). The SET can propagate through logic and if captured by a sequential circuit, can upset the IC architectural state. Collection at a sequential circuit storage node can

• L.T. Clark, C. Ramamurthy, and K.E. Holbert are with Arizona State University, Tempe, AZ 85287. E-mail: lawrence.clark@asu.edu, cramamur@asu.edu, keith.holbert@asu.edu.
• D.W. Patterson's E-mail: dan_w_patterson@yahoo.com.

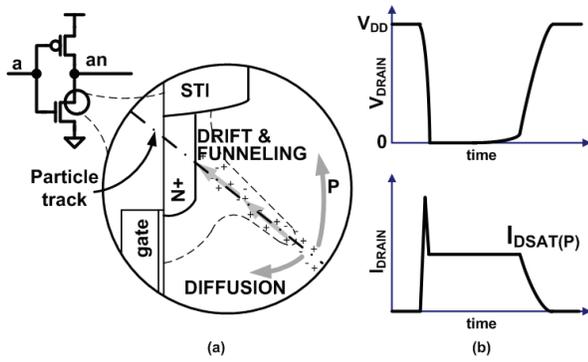


Fig. 1. (a) Basic drift and diffusion charge collection mechanisms. (b) SET duration is controlled by the driving circuits' ability to remove the deposited charge.

upset the logic state directly, referred to as a single event upset (SEU).

Recently, RHBD efforts have changed from a sole focus on SEU to approaches that mitigate SETs as well, since the susceptibility to SET at high operating frequencies becomes comparable [16]. What makes overcoming SETs so difficult is that their duration has been measured experimentally to be between 200 ps and 3 ns [17][18][19]. The latter can consume an entire clock cycle of even a moderately fast design. The transient duration is a function of the deposited charge (this a function of the linear energy transfer (LET) and of the circuit drive [20] as shown in Fig. 1(b)). Here, the SET lasts until the PMOS transistor removes nearly all of the deposited charge.

Accelerated beam testing has high flux rates, potentially causing many errors per second. In contrast, actual applications, even in high radiation space environments generally only have a few per year. The conundrum in hardening is that even low probability events can have catastrophic impact in real systems, and can manifest or be limiting in accelerated testing.

1.2 Multiple Node Charge Collection

Soft error susceptibility increases as circuit areas diminish the critical charge required for an upset while the charge deposited by an ion is unchanged, since it is based on the physics of charge deposition and collection [21][22]. A single energetic particle can upset multiple storage nodes in SRAMs, i.e., a multiple bit upset (MBU), and in flip-flops. This can be due to directly deposited or diffused charge, collected at multiple junctions. The likelihood of such multiple node charge collection (MNCC) has increased dramatically with fabrication process scaling, since circuit nodes become more closely spaced.

Some MNCC results on 90-nm SRAMs we measured comprise Fig. 2. The results were obtained after subjecting the SRAM test chip to ion broad beams at different LETs and orientations. The horizontal (along the N-well) and vertical (across the N-well) simultaneous MBU extents are shown. The SRAM uses horizontal wells. N-well breaks help to collect electrons in the P type substrate, which can attenuate the MNCC extents. The N-well depth attenuates the collected hole charge as a charge track ends at the well boundary. The MNCC extents exceed 7 μm at high LET.

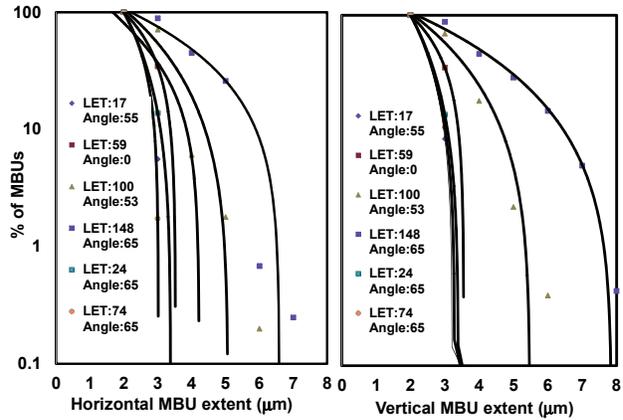


Fig. 2. Measured MNCC spatial extent of bit upsets in a 90-nm SRAM. LETs are effective, i.e., adjusted for the angle of the impinging ions. Horizontal extent is with the ion beam oriented in the well direction, while vertical has the ion beam oriented across the N-wells. Fitted lines are only an aid to the eye.

MNCC can obviously thwart EDAC by causing MBUs so SRAM designers have enforced bit interleaving to avoid it for decades. However, it can also thwart RHBD by upsetting multiple nodes in a design hardened to single node failures. With a high likelihood of MNCC, SEU hardness requires placing storage nodes far enough apart so that simultaneous upset of such critical nodes is unlikely—a single strike at a specific solid angle (directly passing through the nodes) can always disturb multiple nodes.

1.3 Radiation Hardening by Design

RHBD techniques generally utilize some kind of circuit or delay redundancy, the former exemplified by SRAM and latches using dual interlocked storage cells (DICE) [23]. In such designs, SETs on flip-flop (FF) clock and reset signals are particularly problematic [24] since the DICE structure only protects against SEU. Consequently, extreme hardness requires that FFs are also hardened to SETs on inputs and control signals. To protect against SETs, temporal filtering circuits have been introduced [25]. Delay circuits in the latch itself are more straightforward [26][27] but are large. Temporal filtering requires filters that have delay longer than the expected SET duration t_{SET} and increase the FF setup time by twice the filter delay, i.e., $t_{SETUP} > 2 t_{SET}$ [28].

Many RHBD schemes assume that only one circuit node is upset at any given time. Triple mode redundancy (TMR) on advanced nodes has demonstrated the need to provide spatial separation of redundant circuits to avoid their coincident upset, i.e., domain crossing errors [29]. MNCC due to one impinging particle generating charge has also made hardened FF design more difficult, requiring careful physical design as well as clever circuits [27][28][30].

1.4 Soft Errors and Architectural State

Soft errors affecting the processor speculative state can disrupt the architectural state (AS) when instructions “commit” their results to registers or system memory. AS is that logic state required for correct execution [31][32]. Since transients can potentially last through an entire

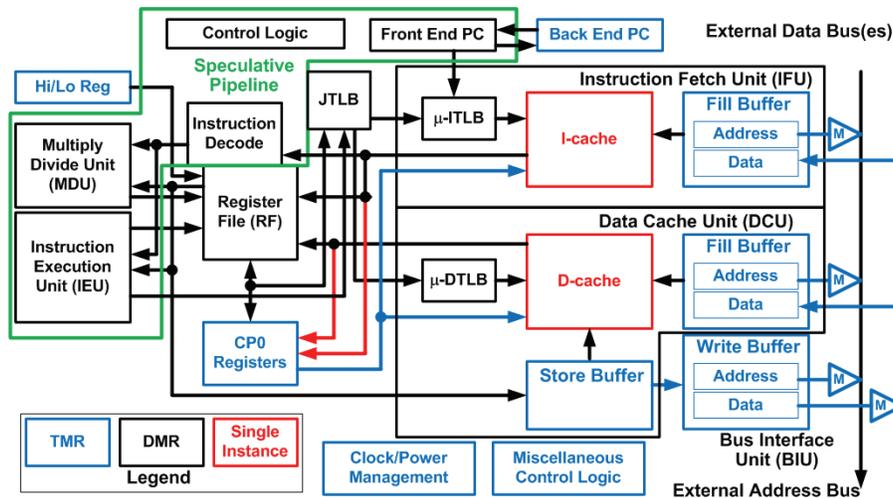


Fig. 3. HERMES high level block diagram showing protection by DMR and TMR as black and blue, respectively. TMR is used to store machine state that is critical to the machine operation, e.g., CP0 registers; HI/LO registers; fill and store buffers; Program Counter (PC) in certain pipeline stages.

clock cycle, the basis of the design here, which we believe is applicable to any high performance VLSI hardening, is avoiding the commission of upsets to AS rather than avoiding them altogether.

1.5 Paper Contribution and Organization

The HERMES (High-performance Embedded Radiation-hardened Microprocessor Enabling Spacecraft) microprocessor core design presented here achieves SEE hardness by combining classical fault tolerance techniques at the microarchitectural level with hardened circuits and careful physical design. The former primarily uses upset detection while instructions are in the speculative state, and instruction restart. The core is a MIPS 4Kc clone with instructions added to allow a software approach to soft error recovery. Software response allows flexible error recovery and complete error logging. Specialized circuits provide the error detection.

Hardened physical design prevents MNCC from circumventing the hardening—special CAD flows enforce critical node separation. The design approaches emphasize logic synthesis and automated place and route (APR) for maximum portability and applicability to other designs. None of the radiation hardening techniques used increases the latch and flip-flop setup times, so they allow full commercial IC performance levels. Additionally, the hardening approaches allow use of dynamic circuits, which are common in high performance commercial processors, particularly memories. Finally, none of the methods impede clock gating as the approaches in [24] do, so they can also achieve low power operation. The hardening techniques used have been experimentally verified through proton and heavy ion broad beam testing for hardness at the block level, which is reviewed.

Section 1 has provided background and motivation for this work. Section 2 describes the microarchitectural and circuit hardening approaches in detail. Section 3 describes the CAD methodologies and overall design results. Section 4 summarizes experimental broad beam testing results and suggests directions for future work.

2 HARDENING APPROACHES

The approaches vary by block function (see Fig. 3). The register file (RF) and speculative pipeline state are protected by dual modular redundancy (DMR). Speculative DMR state differences are checked at their commission to AS or when transitioning to a TMR domain, most often at stores to the cache and store buffer or to the RF. The DMR RF is protected by interleaving and parity. Added instructions allow software-based repair. The RF includes a third read port, which facilitates state repair and allows background scrubbing. Other (non-RF) AS, e.g., program counter, write buffers, coprocessor registers and core I/O, is protected by self-correcting TMR flip-flops and controlling logic. The 16 kB caches are single redundant. The cache SRAM bits are interleaved with byte level parity. A write-through policy allows data cache invalidation upon error detection. We assume primary memory or next level cache is protected by EDAC, which is essential in a hardened system. DMR hit logic, word-line encoding and control signal error check circuitry allow SET induced cache timing and control error mitigation.

2.1. Architecture and Microarchitecture

In keeping with the MIPS approach of using software rather than hardware where possible, we use a software visible approach to soft error mitigation. A detected SET or SEU triggers an exception. The exception handler repairs the processor state via instructions added to the base MIPS instruction set. If the error is in the speculative pipeline, the instruction is restarted before its commission to AS. For memory errors, the response depends upon the memory. The RF is repairable under software control; caches, including the TLBs and micro-TLBs, are invalidated. This software error management allows seamless error logging and flexible response to different detected soft errors.

The MIPS 4Kc has a five-stage pipeline. HERMES uses the same basic pipeline for normal instruction operation.

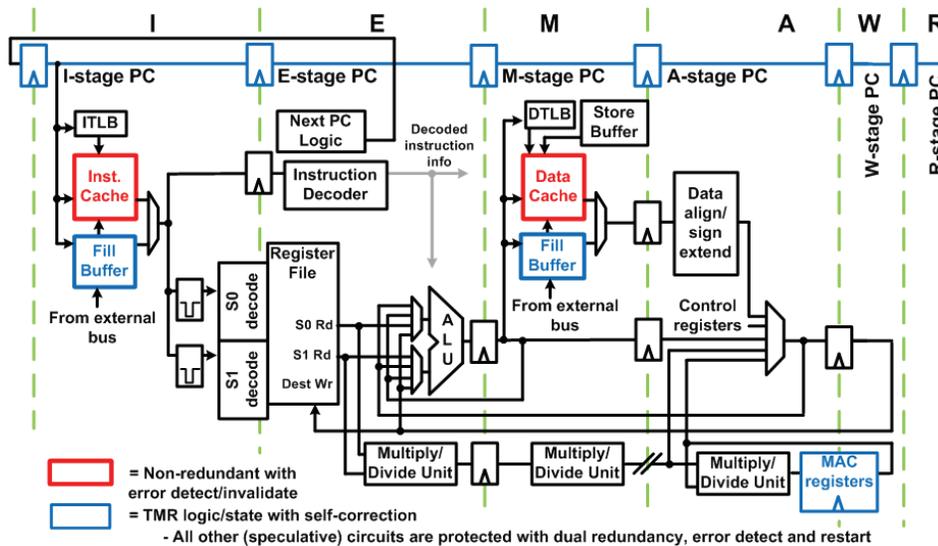


Fig. 4. HERMES pipeline showing protection by DMR and TMR as black and blue, respectively. The 6th (R) pipeline stage facilitates DMR speculative pipeline restart when the instruction is in the branch delay slot.

However, an added 6th pipeline stage supports error recovery when an error is detected in an instruction in the branch delay slot (Fig. 4). The back-end PC is TMR, since the PC information is critical to restarting instructions after detecting a soft-error. Most of the pipeline is DMR, as evident in Fig. 4.

2.1.1 Error Detection Overview

In the speculative pipeline, the RF, caches, and TLBs, HERMES relies exclusively on error detection to avoid propagating incorrect AS. Consequently, complete detection coverage is imperative. The RF flags mismatching store data or write back addresses. For instance, in the case of an SET or SEU within the speculative pipeline, or data misread from the RF, whether due to an SEU in the RF or an SET in the dynamic read circuits, the mismatching DMR data are detected where effects are manifested at an AS write operation. Both RF write and cache addresses are checked at the word lines (WLs), detecting mis-assertions due to SETs as well as DMR errors. The RF data comparators detect all DMR mismatches as it writes. Moreover, since SETs can occur at the writing clock edge, SETs as the write occurs are still caught. The DMR RF write WLs are directly compared, and both A and B copies must match to write a location. The cache WLs are generated by one DMR copy, re-encoded, and then checked against the other DMR copy. Similarly, all cache timing and logic errors are detected. Cache tag and data array parity is checked on every cache operation. Speculative state DMR to AS TMR crossovers check the DMR data matches in what is usually a two-clock operation. The circuit details are described in Section 2.2. An error is flagged on any DMR/TMR crossover copy mismatch.

2.1.2 Soft Error Exception Handler

On a soft error (SE) exception due to a detected error, in-flight instructions are flushed—any state in the speculative DMR pipeline may be corrupted, and the source of the error is unknown as the exception is taken. At this

point, the dedicated SE exception handler is invoked, and further interrupts are disabled. In most cases the program counter (PC) of the last retired instruction is saved as the restart address for execution resumption. If the instruction is in the branch delay slot, the restart PC corresponds to the previous branch instruction (using the added sixth pipeline stage).

At minimum, the SE exception handler first restores the RF state to that before the last instruction retired; repairs RF SEUs; invalidates caches and TLBs; and returns the program flow to the last retired instruction before the SE detection. This “scorched Earth” handler policy is very fast, limiting the possibility of nested SE exceptions, even in accelerated beam testing. The minimal handler, providing just restart and register file SEU repair, requires 86 instructions including NOPs for exposed hazards, since the processor supports single-cycle TLB and cache invalidates. Full error logging, i.e., stepping through the cache to find latent SEUs, requires 115,328 instructions, but is optional. Since the handler is in un-cached kernel space, the actual time depends on the system clock and bus latencies. At a 100 MHz bus speed and no wait states, the mandatory handler code requires less than 1 μ s.

This section describes the added instructions, presented in their context of a complete SE exception handler with full error logging. The SE exception vectors to the same entry address as a reset, soft reset, or non-maskable interrupt (NMI). The handler executes in unmapped, un-cached memory avoiding access to potentially corrupted processor resident data in the TLBs or caches. For non-SE exceptions, the type is set by the CP0 status register. By MIPS software convention, a general purpose register is guaranteed available [33]. However, since a soft error may occur within a Reset or NMI handler, this may not be the case. Thus, the standard R0 register behavior [34] is modified in our design—for instructions executed within the SE exception handler the R0 register is read/write. This base MIPS behavior extension provides the handler a temporary working register. The entry point for Re-

set/NMI/SE exceptions is:

```
LUI    $0, 0x0001
BGTZ  $0, Offset_To_SEE_Exception_Handler
# branch to SE exception handler when condition is
# true; otherwise, fall through to reset handler
NOP
```

For non-SE exceptions, R0 returns zero and the code falls through to the Reset/NMI entry code. For SE exceptions, the value returned is non-zero and results in a branch to the SE exception code. Once the SE exception handler has repaired the corrupted state, it restores the registers it used and executes a return from exception (ERET). Caches and TLBs reload normally. Recovery operations are completely software controlled—data/error logging is optional, and can be altered based on the error type.

2.1.3 Added Instructions

Instructions are added to the HERMES architecture for rapid error handling and AS repair. Since the last instruction to retire may have corrupted state, e.g., its write to the RF may have non-matching DMR data, its state is backed out if the RF was written on that clock cycle. Moreover, the destination register may also have been a source to the instruction, which will be re-executed with the original data. Thus, in the A-stage the RF value to be replaced in the W-stage is read out via its 3rd read port, to prevent a resource conflict with the other two read ports (not shown in Fig. 4). Added RF instructions facilitate repair of its state. To further accelerate error handling, single cycle TLB and cache invalidation instructions have been added. Other added instructions allow RF testability and cache reads and writes for data examination and error validation, as well as SE detection logic testing.

A number of instructions allow access to the DMR arrays individually, bypassing any correction mechanisms. For example, the RF testability write instruction (RFTW) allows single instance writes. This facilitates testing of the repair and parity error detection circuitry, by allowing mismatching writes to the arrays. Additionally, for error reporting, it is necessary to read the RF copies and parity bits independently. The added RDRFPAR instruction in the following example makes subsequent reads of the RF parity only. Also shown is the equivalent RDRFDAT, to allow reading the data only. The read instance instruction (RDINST_x) sets which of the DMR RF or TLB arrays is to be read. There is a hazard on RDRFPAR, but we cannot use NOPs here since R0 will be overwritten (the MIPS NOP is a SLL R0) and R0 contains the base address to dump to (recall R0 does not return zero inside the SE exception). Consequently, SYNC instructions are used instead of NOPS in these cases. To summarize some of the added instructions and their usage, we pick up with the error handler code after the RF A copy has been read out for diagnostic purposes:

```
RDRFPAR      # switch to reading RF parity
RDINSTB     # switch to reading 'B' instance
SYNC
# Byte parity per register
SB    $1, 124($0) # store R1
SB    $2, 125($0) # store R2
....
SB    $31, 154($0) # store R31
```

```
RDRFDAT      # switch to reading RF data
SYNC; SYNC
SW    $1, 156($0) # store R1
SW    $2, 160($0) # store R2
....
SW    $31, 276($0) # store R31
....
```

After potentially corrupted RF data have been (optionally) saved to memory for analysis, the code repairs the RF registers 1-31. The backup register file (BURF) instruction restores the register file state to that before the instruction that triggered an error. The repair general purpose register (RGPR) instruction is used to repair RF SEUs one register at a time, where five bit groups with parity errors are overwritten by the redundant group with correct parity. Only the A copy parity is checked. If there is an error in both arrays, a silent data corruption may occur. This is unlikely, as the RF arrays are automatically scrubbed, using the third read port opportunistically whenever an instruction is retired that does not write the register file. The scrubbing mechanism continuously reads sequential register entries on any clock that does not store the RF in the next cycle. If a parity error is detected, the SE exception is triggered. Three NOPs are required for the hazard on BURF (the handler no longer needs the R0 value at this point). The repair of the RF follows:

```
BURF      # 3 NOPs required for hazard on BURF
NOP; NOP; NOP
RGPR $1
NOP;
RGPR $2
NOP
....
RGPR $31
NOP
```

After the RF repair, cache and TLB values can be dumped. This is useful for determining cross-section and latent SEUs in the arrays.

2.1.4 Added Coprocessor Registers

The MIPS coprocessor register extensions include error masking for SEE detected error discrimination—specific errors can be disabled. All cache array errors are logged, including control SET and SEU locations. DMR to TMR crossovers in instruction fetch, load/store, multiply/divide, and instruction execution units are uniquely identified. Finally, DMR RF word line, write-back data mismatches, and data read parity errors are flagged. Added CP0 registers include the SEE EPC, which stores the PC to return to after a SEE exception. Other added registers provide BURF with a pointer to the last written RF entry and the data for RF restoration to its pre-error state as well as registers for enhanced error visibility. The CP0 error log registers are dumped as follows:

```
MFC0  $5, $9 # sel = 110, R5 <- Error Log 1
SW    $5, 0($1)
ADDI  $1, $1, 0x0004 # increment R1 to next word
MFC0  $5, $9 # sel = 111, R5 <- Error Log 2
SW    $5, 0($1)
```

At this point the base address for the next dump of processor state to memory is updated to prepare for the next SE exception. Then we save the CP0 ErrCtl register in R1

and then clear it. Clearing the WST bit in this register ensures that the added cache global invalidate instructions are properly decoded. After the cache invalidations, the ErrCtl register is restored. Since this register is only used for testing, this step may not really be necessary, but it is possible some code was using it when the soft error was detected.

```

MFC0    $1, $26           # sel = 000, R1 <- ErrCtl
LUI     $0, 0x0000       # R0 <- 0
MTC0    $0, $26
NOP
NOP
CACHE  ICACHE_GLOB_INV, 0($0)
CACHE  DCACHE_GLOB_INV, 0($0)
TLBINV # global invalidate of TLBs
MTC0    $1, $26           # sel = 000, ErrCtl <- R1

```

Finally the handler restores the registers it used and executes a return from exception (ERET). Caches and TLBs reload normally as misses occur.

2.1.5 Limiting Exception Duration

There is a possibility of a second soft error during the exception handler. A nested exception would overwrite the return address, which is stored in the SEE EPC. The added cache and TLB invalidate instructions were added to significantly shorten the handler latency. However, full error reporting can cause the exception to have thousands of clock cycles, depending on the bus multiplier. Moreover, in hard real-time systems, full error reporting may affect latency unacceptably. A simple solution to this problem is to set a software exception for relatively lengthy reporting operations, which is called after the SE exception return. If it is interrupted, the data loss is evident from the truncated error reporting data set, which will not affect normal program operation.

2.1.6 Special Cases

Restarted load and store instructions are specially handled by the hardware—writes to I/O devices may have side effects and thus cannot be re-issued to the bus. Incoming bus data from load instructions are also TMR, so that the data can be used without re-issuing the operation to the external bus, as such operations may also have system level side effects.

The next PC logic (which includes the ALU adder) is dual redundant, minimizing the hardware overhead, with a transition to a TMR PC occurring at the front-end of the pipeline to provide a non-corrupted PC at the back end of the pipeline, which provides the restart address when the pipeline is flushed. A separate PC pipeline is maintained (in the IEU) for multiply divide unit (MDU) instructions (M- thru W-stages), since the W-stage PC for an MDU instruction is required for some SE exception return cases.

The MDU pipeline runs concurrently with the integer pipeline and its depth (particularly for divides) is instruction dependent, so the necessary logic to allow it to complete despite an SE exception in the DMR pipeline is included. This is critical as the RF may no longer contain the divide instruction inputs when the MDU pipeline is restarted. This restart information is TMR.

The data cache is written simultaneously with the

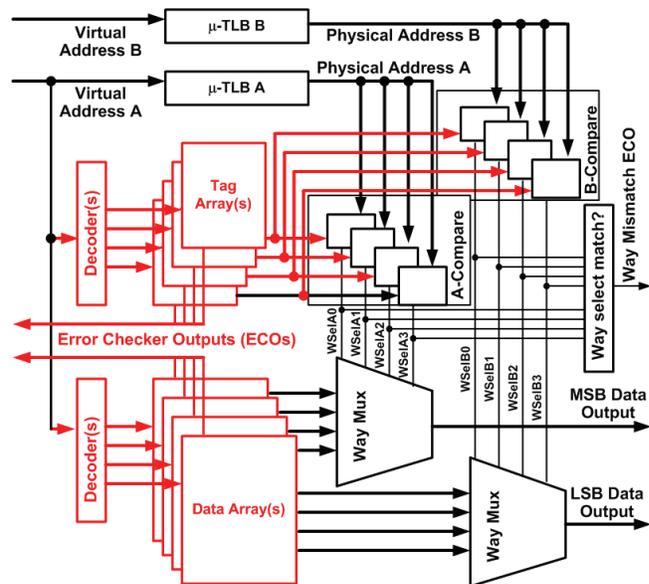


Fig. 5. Cache organization showing DMR comparators and way multiplexers. Single instance cache blocks are in red. The error checkers provide a re-encoded copy of the WL addresses to check against the B copy, as well as SET detection. The A copy tag compare circuits drive the MSB way multiplexers and the B copy tag comparators drive the LSB way multiplexers.

store buffer when no array conflict arises. Since the hit/miss state is initially unknown the tag is looked up in the M-stage. On a hit, the data array is written at the first subsequent cycle that the cache is not executing a load. All other writes to TMR AS require two clock cycles.

When writing to the CP0 registers, a dual-to-triple redundant crossover occurs in the A-stage of the pipeline, and the actual register update is in the W-stage. This allows the prevention of errors that originate on the DMR side of the crossover logic from making it into the CP0 registers. Once updated in the W-stage, the TMR self-correction mechanism ensures the integrity of these registers. The crossover circuits are described in Section 2.2.4.

2.2 Circuit Design

In this section we describe the specific circuit techniques used. The caches use SET error detection, primarily based on catching transients. Error checkers to determine DMR mismatches are also shown. We provide details of the self-correcting TMR FFs that are the basis of the TMR hard macros last.

2.2.1 Cache Circuits

Identical cache arrays are used for the data and instruction caches. For the instruction cache, byte enables are tied together. The caches are implemented as 16 arrays for each of the tag and data SRAMs, with each array comprised of two sub-arrays. Each sub-array receives a gated clock. For power savings, each array resides in precharge when its clock is not active. Six clocks are active (four for the tag and two for the data) for a lookup. Only the active bank is activated for a write, but a fill can write to four sub-banks at once, allowing single cycle line fills.

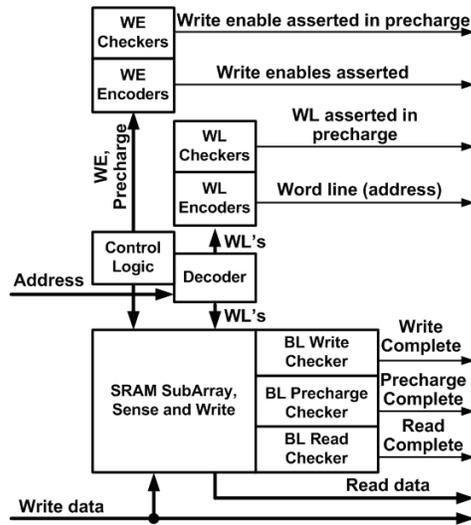


Fig. 6. Cache SRAM array checking circuits to detect mis-assertion and timing errors in the periphery and control circuits.

SRAM cells are TID and SEL hardened by using annular NMOS transistors and full guard rings. The 16 kB virtually indexed and physically tagged caches are not redundant. The instruction and write-through data caches are SEU protected by bit interleaving and parity. When a parity error is detected in either cache, the entire cache can be invalidated.

2.2.2.1. Memory Circuit Peripheral Errors

McDonald *et al.* observed decades ago that SETs in SRAM decoders and control logic could cause upsets that are non-random and thus not amenable to mitigation using EDAC [35]. These types of errors have been observed in SRAMs hardened by process as well, where resistor-hardened SRAM cells exhibited no static errors, but did exhibit dynamic (operating) errors at relatively low LET [36]. These errors can cause the wrong word-line (WL) to be asserted, causing a silent data corruption (SDC) error as the parity or EDAC bits read out may match the data [37]. Local WL mis-assertions in a SRAM with hierarchical WLs have also been shown to cause incorrect writes [38]. Other SET induced errors can include writes to multiple cells or writes from read out cells to others, in the case where a WL glitches on after the bit lines (BLs) are fully driven but before BL precharging occurs. Experimental measurements have shown that simply increasing the array size so as to have large SET immune drivers is largely impractical [39].

2.2.1.2. Peripheral and DMR Error Protection

Tag compares are DMR as shown in Fig. 5. This redundancy is leveraged by using one for each half of the data cache, to limit the circuit loading of the cache way selects. This limits the total circuit size, where one copy would have to be nearly twice the size to drive the full data array way multiplexer loading [40]. TLBs are DMR, and reload automatically after an invalidation operation, as the caches do. A TLB error will affect one copy only, which is detected at the caches or store/write buffers—no local checking is required. The caches are extensively protected from periphery errors by checking circuits (Fig. 6). The

TABLE 1
Cache checking circuits and transient errors detected.

| SET Caused Assertion or Timing Error | Detection |
|--|--|
| WL mis-assertion during precharge | WL precharge checker |
| WL trailing edge pull-in; WL encoder produces the correct address. | BL read and write checkers |
| WL trailing edge push-out | WL precharge checker |
| Precharge mis-assertion, trailing edge push-out or pull-in | BL read and write checkers (if sufficient to cancel read or write) |
| Precharge leading edge push-out | BL precharge checker |
| Precharge trailing edge pull-in | BL precharge checker |
| Write enable (WE) mis-assertion or leading edge pull-in | WE checker |
| WE leading edge push-out | BL write checker (determines that write completed) |
| WE trailing edge pull-in | BL write checker |
| WE trailing edge push-out | WE in precharge phase checker |
| Write enable (WE) mismatch | Dual redundant copy check |
| Write way (WW) mismatch | Dual redundant copy check |
| Tag hit/way select mismatch | Dual redundant copy check |
| Address mismatch | Dual redundant copy check |
| Write data mismatch | Dual redundant copy check |
| Bank clock mis-assertion | A and B copies are compared |

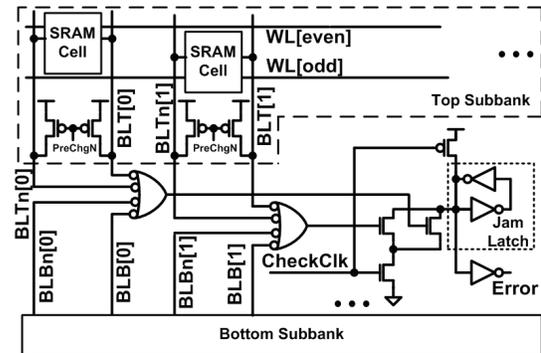


Fig. 7. BL precharge checking circuit. The CheckClk signal is asserted to provide a timing window, during which the BLs should all be high. Any low BL triggers an error.

checkers are sensitive to incorrect cache inputs as well as timing errors.

Table 1 shows the key checking circuits, their types and functions. As mentioned, they can detect both transient faults in the cache circuits, many of which are dynamic, as well as faults caused by non-matching addresses or write data from the speculative pipeline.

SETs are transient, so they can dissipate before a sampling clock edge, while still disrupting memory operation. Consequently, the memory checking circuits are “one’s catching,” with set-reset operation. The basic dynamic error checker (EC) circuit is based on a classical domino gate, which with a full jam latch or PMOS keeper (which comprises a half-latch) captures incorrect transients. An example is shown in Fig. 7. Here, an incorrect BL precharge will leave a BL low. This is sensed by the 4-input NAND gate whose output, qualified with the check window will assert an error. The check window is a chopped clock, which provides time for the precharge to occur prior to the check.

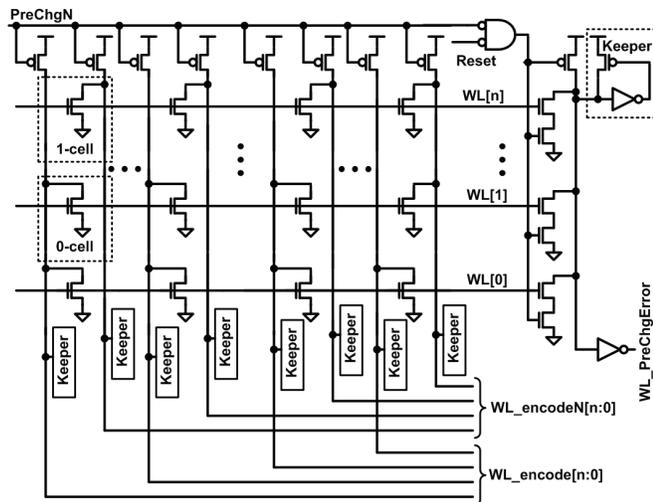
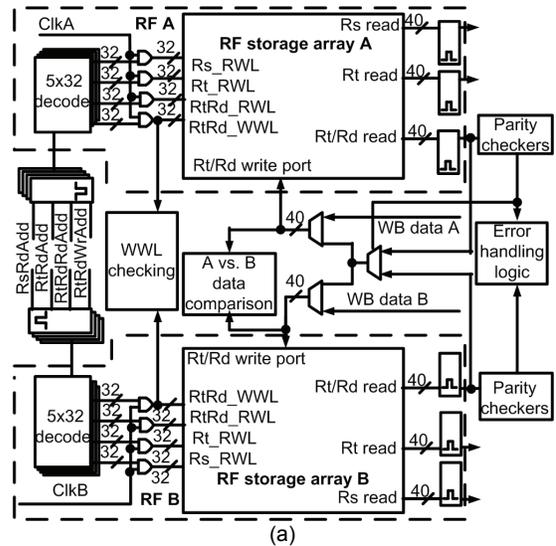


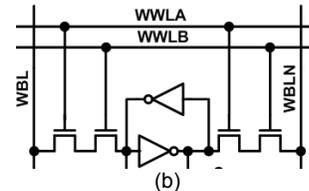
Fig. 8. Cache WL encoder (left). The positive and negative versions physically overlap (i.e., either a one cell or zero cell occupies the array space), providing high density and coverage of multiple WL assertions. The WL precharge checker is shown on the right and its ECO reports a WL assertion during precharge.

Some checking circuits focus on effect, some focus on insuring a signal is not asserted during a timing window. The former is used where possible, since the mis-assertion or timing error may have been benign, i.e., a short precharge due to a trailing edge pull-in is unimportant if the BLs have fully precharged to V_{DD} . The same approach is used for writes and reads—reads are checked with a replica read column and a replica write column ensures that a write completed. It is not possible to determine if a WL assertion during precharge had an adverse effect. If the BLs are not precharged, this can result in some cells being written. If the BLs are nearly precharged, this is basically a read case and should not upset the cell, although it may preclude a full precharge for the next read. Consequently, there are checkers to determine if WLs or WEs are asserted during the precharge phase. Byte and way enable trailing edge push-out can result in contention with the BL precharge drivers.

Addresses into the caches are DMR but only one copy is used to decode WLs. They are protected by re-encoding at the bank level. The encoding circuit, shown in Fig. 8 re-encodes the activated WL as determined by decoding address copy A. This generated address is compared with the intended address copy B at the next pipeline stage. In the encoders, both positive and negative logic ensure that that a multiple WL assertion is not masked. For instance, the dynamic encoder will mask WL 16 (binary 01000) with WL 31 (binary 11111). However, this is caught by the complementary WL encoder. As evident in Fig. 8, no additional area is required to support both polarities, since either the pull down on positive or negative output will occupy each location. Thus, the circuit can detect multiple WL assertions as an error. The cache arrays do not use column multiplexers [40]. This avoids back writing a cell after a read, as well as incorrect column selects due to a driver SET or other address error.



(a)



(b)

Fig. 9. (a) Register file block diagram showing WWL checking, DMR write data A and B comparison and data paths for SEU repair. The third read port allows data that will be replaced on the next write to be read out and saved in case it needs to be put back in the event of a pipeline restart. (b) RF cell requiring coincident WWLs to write. The readout is conventional domino (not shown).

2.2.2 Register File

Register files are difficult to protect with EDAC since the read path is critical—an extra pipeline stage would be required [41] or timing is severely impacted as in [42]. This has led to interesting approaches to implementing parity, e.g., a multi-cycle calculation in [43]. The DMR register file has two complete copies as shown in Fig. 9(a). The physical implementation interleaves the A and B copies [44]. The RF cell uses domino BL readout, with the top and bottom 16 BLs multiplexed by a NAND set-dominant latch. The RF cell write is controlled by two access devices, one controlled by the A DMR copy and one controlled by the B copy of the write WLs (WWLs) diagrammed in Fig. 9(b). This prevents a SET induced write from corrupting both copies. On write operations, the two WWLs are checked to be identical using domino checking circuits similar to those described for the caches above. A mis-assertion triggers an SE exception.

The Rt/Rd read port is added to the two (Rs and Rt) read ports required for source operands as shown in Fig. 9(a). This port reads the register to be written one clock cycle before a W-stage RF write, thus providing the data overwritten for use when the write is backed out. On cycles where there will not be a RF write in the next clock, this port is used to read the registers in sequence, implementing the scrub operation mentioned in Section 2.1.3. Consequently, parity need only be checked at this port, where the parity bits can be used to control whether data

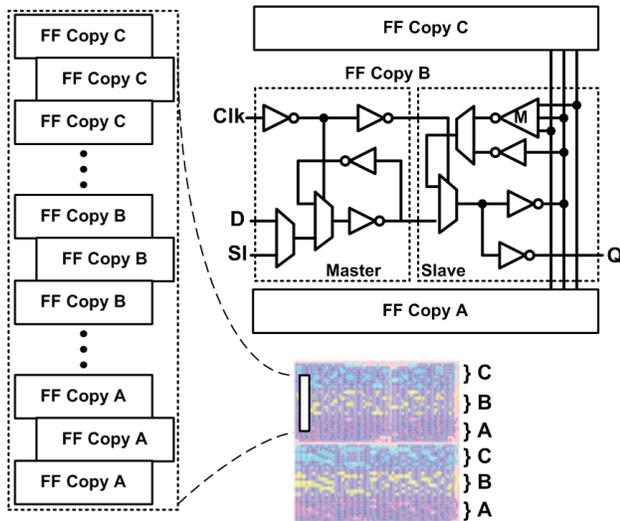


Fig. 10. Self-correcting TMR flip-flop circuits (upper right) and an example of automated layout maintaining separation between domains. Only the slave latch is corrected as this has no delay impact and minimizes the number of connecting wires between copies. The alternate feedback path without the majority gate allows separate domain testing.

are transferred from the A copy to the B copy, or vice-versa on RF repair operations. The scheme here is very similar to that implemented in a contemporary effort [45]. The parity granularity is 5 bits of data per parity bit—this maximized the spacing between the data bits in one group. Repair data are multiplexed into the write path.

2.2.3 TMR Self-correcting Flip-flops

The FFs are built in TMR groups, with three FFs with the slave latch feedback majority voted (Fig. 10 upper right). This allows an incorrect copy, whether brought into one of the D inputs or due to an SEU to be corrected in the low phase of the clock, when the slave feedback path is active [46]. The slave feedback is not in a critical timing path, so there is no impact on performance aside from the increased routing in TMR circuits. The clocks are also TMR, so a clock SET or failure to assert a clock has no impact—the incorrect copy will be overwritten in the slave feedback phase. The TMR circuits are laid out in nine standard cell height groups (A, B, and C) as shown (Fig. 10 lower right). The TMR groups can be tested separately since defects in one pipeline would disable the correction. This is provided by the added multiplexer in the slave feedback. The TMR APR flows automatically choose the correct cells and routes for the TMR voting in the slave latch, based on the FF placement as described below.

2.2.4 DMR to TMR Crossover Logic

The DMR to TMR crossover logic comprises Fig. 11. The crossovers are part of the TMR logic, implemented as hard TMR macros. This is facilitated by aspects allowing their construction by synthesis/APR, albeit at some area penalty. Referring to Fig. 11, there are three TMR groups. Two groups at the left, duplicate the inputs from the DMR A and B copies. These are XORed to determine a mismatch, which will cancel the AS commit and flag an

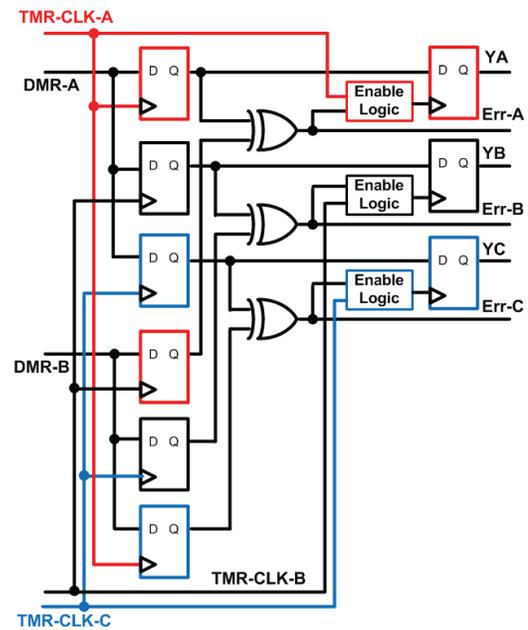


Fig. 11. DMR to TMR crossover circuit. The enable logic blocks the commit to the final FFs if there is a mismatch between the copies. The rotated A, B, C clocks on the DMR-B copy allow detection of a clock mis-assertion. A, B, and C TMR domains are color coded.

error. The enable logic controls a commit to TMR AS. An SET or SEU in the commit clock cycle will upset only one copy, allowing the transfer to AS, where it is corrected by the last stage of self-correcting TMR FFs, outputting YA, YB and YC. The clocks are purposely not connected to their respective DMR inputs, as this allows a clock error to be detected as well.

3 PHYSICAL IMPLEMENTATION

Due to the need to maintain separation of redundant circuits so that two are not easily upset by a single impinging ion, circuit placement is a key challenge. The HERMES core includes TMR hard macro blocks, as well as hard macros for the RF, caches and clock spine.

3.1 Top Level Synthesis

The top level flow begins with a conventional (unhardened) synthesis run using Cadence RTL Compiler (RC). This run provides essential timing parameters for the TMR hard macros, which are synthesized separately. Synopsys design constraint (.sdc) files are generated for each hard macro. Details of the TMR synthesis and APR are described subsequently. The hard macros, including the caches, are then brought into the design described by liberty (.lib) timing files and (.lef) abstracts. RC physical is used to re-synthesize the top level with these hard blocks.

3.2 RHBD Physical Layout

There are 25 hard macros, comprised of the Cache, Clock-spine, RF, and 22 TMR blocks. All core DMR and TMR circuits are synthesized and laid out using specially developed synthesis and APR flows. The DMR spacing is enforced using fences, and the two regions have a minimum spacing of 32 μm . The distance to logic in the TMR

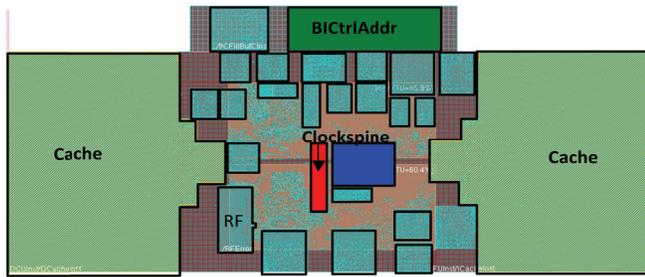


Fig. 12. HERMES core layout showing RF, clockspine (red), IEArch (blue) and the TMR hard macros, as well as data (left) and instruction cache (right).

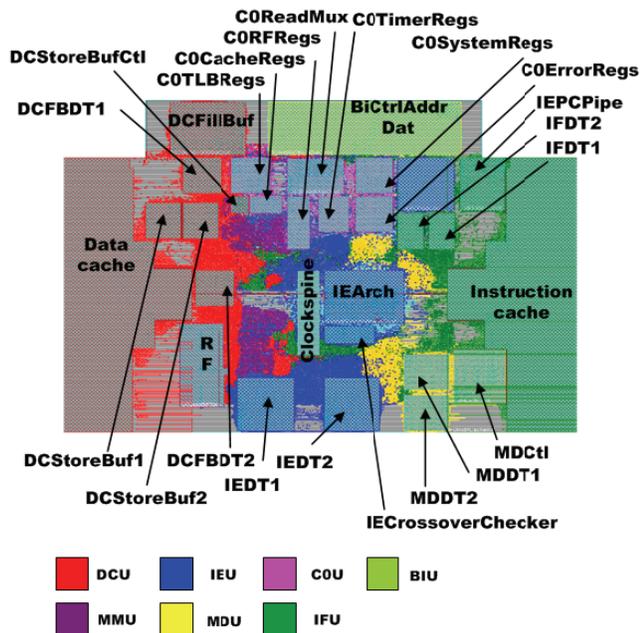


Fig. 13. Top level core layout amoeba view showing the distribution of blocks by function. Caches are cut off at the right and left.

embedded blocks is at least $8\ \mu\text{m}$. Distance between TMR like cells is over $32\ \mu\text{m}$.

3.2.1 TMR Hard Macro Generation

The fine grained self-correcting TMR FF layout ensures greater than $32\ \mu\text{m}$ critical node spacing (Fig. 10)—the APR flow guarantees similar gate cone separation to mitigate MNCC SETs [46]. Only one TMR domain error was detected in multiple days of broad beam heavy ion testing, which we attributed to a fabrication defect in one of the TMR registers. The TMR blocks now incorporate full scan with the self-correcting feedback disabled—note the added multiplexer in the slave latch feedback in Fig. 10. Consequently, testing can screen out such defect-induced softness in the core.

The TMR macros are each resynthesized using the appropriate timing constraints as single, rather than TMR instances. The TMR FFs are uncoupled for this stage (they essentially appear as single-redundant versions). Placement is run in Cadence Encounter for the single A instance of the logic. The floor plan is divided into equal stripes, each 8 cells tall for the A, B, and C copies—cells are only placed in the regions reserved for the A copy. An

TABLE 2
Hard macros and their functions.

| Macro | Function |
|--------------------|--|
| BICtrlAddrDat | Bus interface unit (TMR) |
| C0*Regs. | C0 register blocks (TMR) |
| C0ReadMux | C0 interface to the DMR |
| DCFBDT1, 2 | Data cache (DC) fill buffer |
| DCFillBuf | DC fill buffer (TMR) |
| DCStoreBufCtl | DC store buffer control logic (TMR) |
| DCStoreBuf1, 2 | DC store buffer (TMR) |
| IEArch | Architectural state storage and control for instruction execution (IE) (TMR) |
| IECrossoverChecker | DMR to TMR cross over checking circuits for IE (TMR) |
| IEDT1, 2 | IE dual to triple redundant crossovers (TMR) |
| IEPCPipe | IE program counter TMR copies for instruction restart (TMR) |
| MDCtl | Multiply-divide (MD) control (TMR) |
| MDDT1, 2 | MD dual to triple redundant crossovers (TMR) |
| IFDT1, 2 | Instruction fetch (IF) dual to triple crossovers (TMR) |
| RF | Dual redundant register file and checking logic |
| Clockspine | Clock spine and top level gating |
| Cache | 16 kB 4 way set associative caches |

extra intervening cell row separates each stripe to avoid domain crossing errors at the borders. Custom programs then triplicate the logic and merge the FFs and transparent latches into the appropriate self-correcting TMR cells. These sequential circuit elements now span 19 rows, but do not block logic underneath. Pin placement is important, since many of the TMR blocks implement logic with a great deal of communication with other blocks, e.g., DMR to TMR crossover logic. Another program moved all pins to the top hard block layer (which was generally M5) and ensured separation to ease the top level routing congestion.

The custom programs then replicate the A copy logic and placement to the B and C copies and regions. Depending on the FF or latch cell placement, an appropriate physical cell is used that has the tracks interconnecting the A, B, and C feedback voting wires in the correct placement. In this manner eight FFs can occupy the same horizontal position without their voting wires interfering with one another [46]. Clock tree synthesis is run, providing one or more of each top level A, B, and C clock signals to the macro, depending on the number of sequential elements. Clock trees are pre-routed. Internal skew minimization is the primary target. We endeavored to match the latency of all the TMR blocks, but due to large loading differences this was not possible. Routing then finishes the block. One limitation with this scheme is that post-route optimization is limited. This is mitigated somewhat by over-constraining the block timing.

3.2.2 Custom Hard Macros

The HERMES hard macros and functions are listed in Table 2. The RF and cache arrays are full custom. One hard macro supplies both caches. The data cache is a superset of the instruction cache, having finer write granularity, so the block supports that and controls are tied

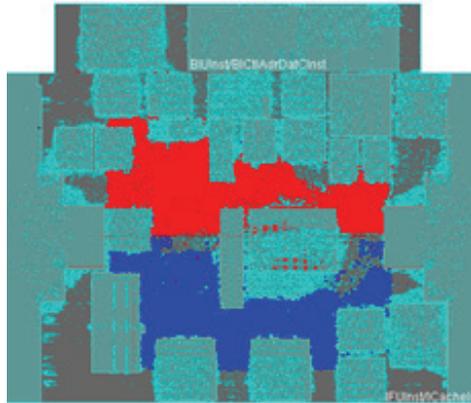


Fig. 14. Core layout showing large separation between the A (red) and B (blue) DMR domains.

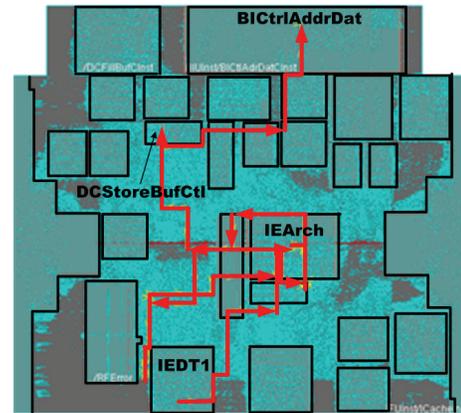


Fig. 15. Core critical timing path (traced in red). The path traverses the IP core, passing through IEArch twice.

together to give instruction cache functionality. The cache layouts used structured datapath placements to ensure data buffering and multiplexer, as well as clock placement provided adequate separation. The cache physical design uses a structured placement that maintains bit spacing throughout the input and output buffering. This avoids a MNCC induced MBU in data bits protected by the same parity bits. Byte parity is used in the data arrays, providing separation of approximately $32\ \mu\text{m}$. Tag array bit separation is smaller, at $21\ \mu\text{m}$.

The RF interleaves bits in groups of five bits, with a critical node separation of $67\ \mu\text{m}$. The error correction block is attached, with bit interleaving maintained throughout by using structured placement.

The clock spine is semi-custom, using hand placement of standard cells. The top five inverter stages use 38 distributed drivers [47]. In this manner, an SET can cause a phase shift of less than 2.7%. At lower stages, each clock driver is triplicated. The top level of clock gating for the unit level, e.g., data or instruction cache, RF or MDU is performed within the clock spine. The clock enable latches are checked against a DMR copy. An error is signaled if any clock in mis-asserted.

3.2.3 Top Level Floor Plan and Placement

Placing hard macros is historically difficult with limited tool support for automatic optimization. We evaluated nearly 100 floor plans, settling first on having the caches on each side and then on the final HERMES hard macro placements (see Fig. 12). Routing congestion estimates were used to drive this process.

The clock spine, which is the central clock unit, is placed near the absolute center of the chip to limit clock route lengths and balanced clock delay and skew. It is shown in red in Fig. 12. The most routing intensive TMR block IEArch, which holds instruction execution architectural state, is also centrally placed (blue in Fig. 12) to ensure adequate routing resources. The Bus-interface unit (the block labelled BICtrlAddr) is placed on the periphery, green at the top of Fig. 12. The caches are placed at the two sides of the chip since their interfaces then reside along the edges with the rest of the core. The overall block placement after APR is shown in Fig. 13.

3.2.4 DMR Speculative Datapath Placement

The DMR datapaths only interact at the commit to AS, which occurs in the RF or in TMR hard macros. Consequently, maintaining physical separation of these circuits is straightforward. Separate ‘fences’ are created and assigned to the DMR A and B logic copies, divided vertically near the core center. The placer maintains the separation between them achieving very large logic cone separation as shown in Fig. 14.

3.3 Clock Trees

Clock gating is the most important feature for low power, but has been shown to be difficult to implement in hardened designs, as each buffer or gater level provides a target for SET generation or an SEU in the gater latches [24]. The methodology used in HERMES comprehends this by using a hardened spine as the root of the clocks and redundant clocks to all other redundant blocks. Domains are not mixed, i.e., clocks to TMR A, B, C or DMR A or B domains are never mixed. Consequently, a SE propagating a clock (or blocking one) will impact only one domain, allowing correction or for the error to be caught. Clocks to the caches, which are only sent when a bank is required to operate, and thus double as control signals, are checked against their enables at the cache periphery. These clock errors can trigger an SE exception like any other. This approach also allows second-level clock gating to be inserted at any point in the tree with no impact on hardness.

The clock spine is a clock mesh compressed into a smaller area and provides multiple copies of the clock with increased drive strengths for different logic blocks. Within the clock spine upper clock distribution network, multiple drivers allow for negligible SET induced skew and different clocks have greater than $30\ \mu\text{m}$ separation between drivers to mitigate domain crossing errors [48]. There is less than 10 ps of systematic skew between the clock spine clock outputs, which are tuned to the loads post clock tree synthesis (CTS) and route. After the clock spine, the core clock trees use conventional CTS.

As mentioned, each of the TMR blocks has at least one internal TMR clock tree, so as to allow the global differ-

ences between TMR blocks to be de-skewed at the core level CTS. All the TMR blocks are implemented with less than ± 10 ps of skew and a maximum latency of 220 ps, which corresponds to 8 inversion stages of buffering and gating. The DMR clock trees are then synthesized. The top level DMR clock tree is constructed to match the latencies of the TMR macros. The global systematic skew was iteratively minimized. Finally, top level hierarchical CTS then balances clock delays to the FFs and latches of all TMR and DMR regions. A ± 75 ps global skew was achieved when measuring to all FFs and latches.

4 RESULTS

4.1 Performance

In this section we evaluate the results and analyze the compromises between hardening, timing, and power.

4.1.1 Timing

The HERMES core database with fully extracted RC parasitics at $V_{DD} = 1.2$ V has 0 ns timing slack at 450 MHz as measured by Primitime. The maximum path depth is 55 inversions, which is shared by numerous critical timing paths. The worst critical timing path traverses four TMR macros in all, spanning the entire core vertically and passes through the IEArch TMR block twice (see Fig. 15). While further macro placement optimization may help, it is likely that a larger number of smaller hard macros would be required—our future efforts are focused on a “soft” methodology that does not require the hard TMR macros to achieve the same critical node separation. The hard macro approach also increases routing congestion over a pure soft-macro approach.

4.1.2 Area

The HERMES core occupies 11 mm². TID hardening accounts for a large portion of the size, increasing the SRAM cell area by approximately 4x and essentially doubling the standard cell height (the TID hardened cell height is 16 M2 tracks). All cells support the low standby power or standard version of the process. The instruction and data cache arrays occupy 44% of the total core area. Less than 18% of the cache array area is error checking and associated logic. In each cache, the tag arrays and comparators occupy 0.56 mm² while the data arrays occupy 1.875 mm². DMR logic is 21% of the total. 11% of the cells are buffers. The clock trees comprise 1.3% of the cell area. Crossover logic comprises 13.9% of the overall cell area, indicating that our decision to triplicate the first stage rather than building a custom cell increased the core area by 4% over a hardened full custom crossover.

4.1.3 Power

Primitime simulated average power dissipation is 465 mW running Dhrystone 2.1 at 1.2 V, 450 MHz using the fully extracted netlist for the core and extrapolating measured data (from the cache test chips) for the caches. The core power is 364 mW, with the caches comprising the remainder. Global wire and the top level clock tree capacitance account for 22% of the core power dissipation.

It is difficult to fairly compare this design’s power dis-

sipation to that of a non-hardened version without implementing the latter. The TMR has low activity factor, but obviously triples those circuits power. Cache power increase is modest as the only dual redundant power is comparators. Checkers have a small impact and caches are completely clock gated as mentioned. The DMR data path dissipates twice the power of an unhardened non-redundant implementation and should dominate the power dissipation, bounding the power cost to more than 2× but we believe well below 3× that of an unhardened version with the same cell library.

4.2 Radiation Hardening

Multiple test chips were broad beam tested to confirm the approaches used in HERMES provided the requisite hardness.

4.2.1 Experimentally Measured TID and SEL Hardness

A cell library test chip [48] and the other test chips confirmed negligible I_{DD} leakage increase (a few %) up to 2 Mrad(Si) [47][46]. The caches have similar TID hardness, using annular NMOS transistors [40]. The cost of TID hardening is substantial, essentially doubling the standard cell areas. This area impact is due to N-type guard rings between NMOS diffusions and the N-wells, as well as metal 1 instead of polysilicon connections over the guard rings.

No single-event latchup (SEL) was observed in testing until 200 MeV-cm²/mg effective LET. Above 200 MeV-cm²/mg the devices did occasionally require power-down to reset, although no increased supply current was observed.

4.2.2 SEE Testing Results of the Approaches

Two test chips were designed and fabricated to determine the constituent circuit SEE hardness. Both used a programmable built-in test engine implemented entirely using the self-correcting TMR logic. As mentioned, only one error in the TMR self-correcting logic was detected in multiple days of testing. Since it occurred at very low effective LET, we suspected but could not confirm, a manufacturing defect nullified the voting in those latches.

The RF and DMR ALU combination was tested with heavy ions and protons. The latter results are shown in Fig. 16(a) where fluence was $5(10^{11})$ particles/cm² for most tests, at fluxes ranging from $2.3(10^8)$ to $8.9(10^8)$ protons/cm²/s. The RF was successfully repaired after each error detection, i.e., in no case did both DMR parity groups have an error. The accumulated SEUs residing in the RF are shown, as well as the relative write back (WB) vs. WL errors detected. Note that a WL error is not necessarily an SET in the RF, but more likely an error propagated in one copy of the DMR address control logic. The higher cross-section may thus merely reflect the greater circuitry dedicated to 32-bit data as opposed to RF addresses. There is a subtle voltage dependency, which is stronger at low energy.

In heavy ion broad beam testing with LET_{EFF} from 1.4 to 219.8 MeV-cm²/mg at fluences from 5×10^5 to 2×10^7 particles/cm², all RF and cache upsets were corrected or

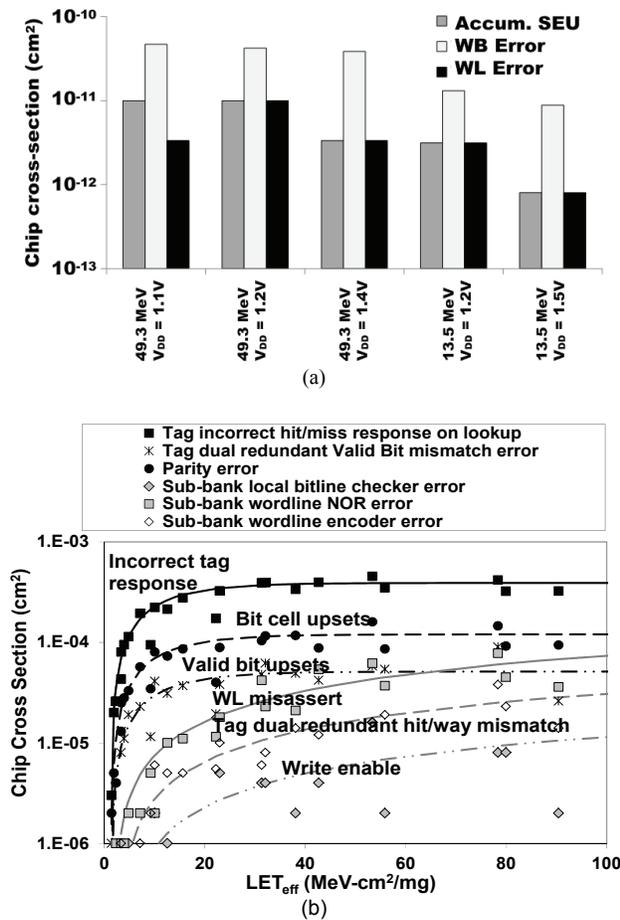


Fig. 16. RF and DMR ALU proton test results (a) at different energies and voltages. Cache tag heavy ion cross sections (b).

invalidated, respectively, using the aforementioned schemes. Selected cache tag cross-section data are shown in Fig. 16(b) where the cross-section of periphery errors is approximately just one order of magnitude below the array SEU errors. This validates the importance of protecting memories against SET, particularly in these relatively small arrays.

5 CONCLUSION

We have described a complete embedded processor design that provides excellent radiation effects hardness. The design presented here reaches 450 MHz at less than 1 mW/MHz power dissipation by hardening with circuits and microarchitecture. The hardening approaches efficacy has been experimentally proven at the block level. However, the design approaches, particularly for physical design, require further refinement.

Unhardened commercial MIPS 4Kc processors exceed 600 MHz on similar processes indicating that the hardening approaches do impact the design, although we believe limited resources and design iterations may account for most of this—as mentioned, none of the soft-error hardening approaches directly increases the critical timing paths. All of the hard macros have better than 1 GHz timing. The cache timings were compromised post-test chip testing to 800 MHz, to allow greater automation in their

assembly. Overall processor power dissipation approaches 1mW/MHz running Dhrystone 2.1.

TID hardening increased the SRAM cell area by 4× and the standard cells by about 2× over an equivalent unhardened design. The TID hardening adversely affects the processor size, routing lengths, and timing but larger cells take strain off of pin and routing density. The hard TMR macro methodology created unnecessary critical timing paths due to inefficient routing. Work is ongoing to improve on the hardened circuit placement and routing to further approach unhardened core power and performance.

6 ACKNOWLEDGMENT

The authors gratefully acknowledge the contributions of past ASU students and funding by AFRL/VSSE in Albuquerque, NM. NASA/JPL contributed to proton testing.

7 REFERENCES

- [1] K. Label, et al., "Single event effect proton and heavy ion test results for candidate spacecraft electronics," Proc. Radiation Effects Data Workshop, 1994, pp. 64-71.
- [2] E. Normand, "Single-event effects in avionics," IEEE Trans. Nuc. Sci., vol. 43, no. 2, pp. 461-474, 1996.
- [3] D. Rea, et al., "PowerPC RAD750-A microprocessor for now and the future," Proc. IEEE Aerospace Conf., 2005, pp. 1-5.
- [4] N. Haddad, et al., "Second generation (200MHz) RAD750 microprocessor radiation evaluation," Proc. RADECS, 2011, pp. 877-880.
- [5] T. Hoang, et al., "A radiation hardened 16-Mb SRAM for space applications," Proc. IEEE Aerospace Conf., pp. 1-6, 2006.
- [6] H. Weaver, C. Axness, J. McBrayer, J. Browning, J. Fu, A. Ochoa, and R. Koga, "An SEU tolerant memory cell derived from fundamental studies of SEU mechanisms in SRAM," IEEE Trans. Nuc. Sci., vol. 34, no. 6, pp. 1281–1286, Dec. 1987.
- [7] R. Lacoce, J. Osborne, R. Koga, and D. Mayer, "Application of hardness-by-design methodology to radiation-tolerant ASIC technologies," IEEE Trans. Nuc. Sci., vol. 47, no. 6, pp. 2334-2341, Dec. 2000.
- [8] G. Anelli, et al., "Radiation tolerant VLSI circuits in standard deep sub-micron CMOS technologies for the LHC experiments: practical design aspects," IEEE Trans. Nuc. Sci., vol. 46, no. 6, pp. 1690–1696, Dec. 1999.
- [9] C. Hafer, et al., "LEON 3FT processor radiation effects data," Proc. Radiation Effects Data Workshop, 2009, pp. 148–151.
- [10] S. Guertin, C. Hafer, S. Griffith, "Investigation of low cross section events in the RHBD/FT UT699 Leon 3FT," Proc. Radiation Effects Data Workshop, 2011, pp. 1-8.
- [11] F. Stuesser, J. Gaisler, R. Ginosar, T. Liran, "Radiation characterization of a dual core LEON3-FT processor," Proc. RADECS, 2011, pp. 938–944.
- [12] J. Montanaro, et al., "A 160 MHz 32 b 0.5 W CMOS RISC microprocessor," IEEE J. Solid-state Circuits, Vol. 31, No. 11, 1996, pp. 1703–1714.
- [13] L. Clark, et al., "An embedded 32-b microprocessor core for low-power and high-performance applications," IEEE J. Solid-state Circuits, vol. 36, no. 11, Nov. 2001, pp. 1599-1608.
- [14] F. Ricci, et al., "A 1.5 GHz 90-nm embedded microprocessor core," VLSI Cir. Symp. Tech. Dig., pp. 12-15, June 2005.
- [15] S. Yang, et al., "A 32nm high-k metal gate application processor with GHz multi-core CPU," ISSCC Tech. Dig., 2012, pp. 214-215.
- [16] M. Suhasish, T. Karnik, N. Seifert, and M. Zhang, "Logic soft errors in sub-65nm technologies design and CAD challenges," Proc. DAC, pp. 2-4, 2005.

- 1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
- [17] M. Gadlage, et al., "Single event transient pulsewidths in digital microcircuits," *IEEE Trans. Nuc. Sci.*, 51, pp. 3285-3290, Dec. 2004.
- [18] J. Benedetto, et al., "Heavy ion induced digital single-event transients in deep submicron processes," *IEEE Trans. Nuc. Sci.*, vol. 51, no. 6, pp. 3480-3485, Dec. 2004.
- [19] B. Narasimham, et al., "Characterization of digital single event transient pulse-widths in 130-nm and 90-nm CMOS technologies," *IEEE Trans. Nuc. Sci.*, vol. 54, no. 6, Dec. 2007.
- [20] D. Kobayashi, T. Makino, and K. Hirose, "Analytical expression for temporal width characterization of radiation-induced pulse noises in SOI CMOS logic gates," *Proc. IRPS*, pp. 165-169, 2009.
- [21] R. Baumann, "The impact of technology scaling on soft error rate performance and limits to the efficacy of error correction," *Proc. IEDM*, pp. 329-332, 2002.
- [22] N. Seifert, et al., "Radiation-induced soft error rates of advanced CMOS bulk devices," *Proc. Int. Phys. Reliab. Symp.*, March 2006, pp. 217-225.
- [23] T. Calin, M. Nicolaidis, and R. Velazco, "Upset hardened memory design for submicron CMOS technology," *IEEE Trans. Nuc. Sci.*, vol. 43, no. 6, Dec. 1996, pp. 2874-2878.
- [24] K. Warren, et al., "Heavy ion testing and single event upset rate prediction considerations for a DICE flip-flop," *IEEE Trans. Nuc. Sci.*, vol. 56, no. 6, pp. 3130-3137, Dec. 2009.
- [25] R. Naseer and J. Draper, "DF-DICE: A scalable solution for soft error tolerant circuit design," *Proc. ISCAS*, pp. 3890-3893, 2006.
- [26] D. Mavis and P. Eaton, "Soft error rate mitigation techniques for modern microcircuits," *Proc. IEEE IRPS*, pp. 216-225, 2002.
- [27] B. Matush, et al., "Area-efficient temporally hardened by design flip-flop circuits," *IEEE Trans. Nuc. Sci.*, vol. 57, no. 6, pp. 3588-3595, Dec. 2010.
- [28] J. Knudsen and L. Clark, "An area and power efficient radiation hardened by design flip-flop," *IEEE Trans. Nuc. Sci.*, vol. 53, no. 6, pp. 3392-3399, Dec. 2006.
- [29] H. Quinn, et al., "A review of Xilinx FPGA architectural reliability concerns from Virtex to Virtex-5," *Proc. RADECS*, 2007.
- [30] O. Amusan, A. Sternberg, A. Witulski, B. Bhuvu, J. Black, M. Baze, and L. Massengill, "Single event upsets in a 130-nm hardened latch design due to charge sharing," *Proc. IRPS*, 2007, pp. 306-311.
- [31] A. Biswas, et al., "Computing architectural vulnerability factors for address-based structures," *Proc. ISCA*, 2005.
- [32] S. Mukherjee, J. Emer, and S. Reinhardt, "The soft error problem: an architectural perspective," *Proc. HPCA*, pp. 243-247, 2005.
- [33] R. Britton, *MIPS Assembly Language Programming*, Pearson Prentice Hall, Upper Saddle River, NJ, 2004.
- [34] *MIPS32 Architecture for Programmers*, Vol. 1, 2001.
- [35] P. McDonald, W. Stapor, A. Campbell, and L. Massengill, "Non-random single event upset trends," *IEEE Trans. Nuc. Sci.*, vol. 36, no. 6, Dec. 1989, pp. 2324-2329.
- [36] L. Jacunski, et al., "SEU immunity: the effects of scaling on the peripheral circuits of SRAMs," *IEEE Trans. Nuc. Sci.*, vol. 41, no. 6, Dec. 1989, pp. 2324-2329.
- [37] L. Clark, "Radiation Hardened by Design SRAM Strategies for TID and SEE Mitigation," in "Radiation Effects in Semiconductor Devices and Circuits," CRC Press, 2010.
- [38] D. Mavis, et al., "Multiple bit upsets and error mitigation in ultra-deep submicron SRAMs," *IEEE Trans. Nuc. Sci.*, vol. 55, no. 6, pp. 3288-3294, Dec. 2008.
- [39] K. Mohr and L. Clark, "Experimental characterization and application of circuit architecture level single event transient mitigation," *IRPS Proc.*, pp. 312 - 317, April 2007.
- [40] X. Yao, D. Patterson, K. Holbert and L. Clark, "A 90 nm bulk CMOS radiation hardened by design cache memory," *IEEE Trans. Nuc. Science*, Vol. 57, No. 4, pp. 2089-2097, Aug. 2010.
- [41] R. Naseer, R. Bhatt, and J. Draper, "Analysis of soft error mitigation techniques for register files in IBM Cu-08 90nm technology," *Proc. IEEE Int. Midwest Symp. Circ. and Sys.*, pp. 515-519, 2006.
- [42] Gaisler, J., "A portable and fault-tolerant microprocessor based on the SPARC v8 architecture," *Proc. Dependable Systems and Networks*, 2002, pp. 409 - 415.
- [43] E. Fetzner, D. Dahle, C. Little, and K. Safford, "The parity protected, multithreaded register files on the 90-nm Itanium microprocessor," *IEEE J. Solid-State Circuits*, vol. 41, no. 1, pp. 246-255, Jan. 2006.
- [44] L. Clark, D. Patterson, N. Hindman, K. Holbert, and S. Guertin, "A dual mode redundant approach for microprocessor soft error hardness," *IEEE Trans. Nuc. Science*, Vol. 58, No. 6, pp. 3018-3025, 2011.
- [45] G. Ditlow, et al., "A 4R2W register file for a 2.3GHz wire-speed POWER™ processor with double-pumped write operation," *ISSCC Tech. Dig.*, 2011, pp. 256 - 258.
- [46] N. Hindman, L. Clark, D. Patterson, and K. Holbert, "Fully automated, testable design of fine-grained triple mode redundant logic," *IEEE Trans. Nuc. Science*, Vol. 58, No. 6, pp. 3046-3052, 2011.
- [47] S. Chellappa, L. Clark and K. Holbert, "A 90-nm radiation hardened clock spine," *IEEE Trans. Nuc. Sci.*, Vol. 59, No. 4, pp. 1020-1026, 2012.
- [48] L. Clark, D. Pettit, K. Holbert, and N. Hindman, "Validation of and delay variation in total ionizing dose hardened standard cell libraries," *Proc. ISCAS*, pp. 2051-2054, 2011.

Lawrence T. Clark has a B.S. in Computer Science (1984), M.S. in Electrical Engineering (1987) and PhD. in Electrical Engineering (1992). Prior to 1992 he worked at Intel as a test engineer and at VLSI technology performing PC chipset design. In 1992 he joined Intel where he contributed to the Pentium, Itanium and XScale microprocessor designs (where he was a Principal Engineer and Circuit Design Manager), compact modeling and CMOS imager projects. He received an Intel Achievement award for the XScale effort. In 2004 he joined Arizona State University where he is now a Professor. From 2009-2004 he was also with SuVolta (on leave and partial leave from ASU) where he was Chief Architect. Prof. Clark has been awarded 100 patents and has published over 110 refereed technical papers and six book chapters. He has been Associate Editor of *IEEE Trans. Circ. and Sys. II*, Guest Editor of *TCAS-I* and twice for *IEEE Journal of Solid-state Circuits*. He is a Sr. Member of the IEEE and Member of the IEEE Computer Society and ACM.

Dan W. Patterson has a B.S. in Electrical Engineering (1983) and has over 25 years of experience in the field of digital VLSI design with the majority of that time involved in microarchitecture and logic design, validation, and silicon debug (mostly of high performance commercial microprocessors). Since 2007 he has been working at Arizona State University designing radiation hardened by design chips, including the architecture enhancements, and microarchitecture and logic design of the entire HERMES microprocessor, as well as most of its logic validation.

Chandarasekaran Ramamurthy has a B.S. in electrical and electronics (2007), M.S in Electrical Engineering (2013) and is currently an electrical engineering PhD candidate at Arizona State University. He worked at Open-Silicon Research, Bangalore India from 2008 to 2011 where he contributed to IP integration, qualification and physical design implementation. His PhD research efforts focus on circuit and chip level implementation techniques for radiation hardening. In 2014 he interned with the IP hardening methodology team at Apple.

Keith E. Holbert earned B.S. (1984), M.S. (1986) and Ph.D. (1989) in nuclear engineering. From 1983 until 1988, he was an engineer at AMS Corp. Since 1989, he has been a faculty member at Arizona State University where he is Director of the nuclear power generation program. He is a registered professional (nuclear) engineer and has published over 150 refereed journal and conference papers, two textbooks, and holds one patent. He is a member of the American Nuclear Society and Sr. Member of IEEE.