A Sea-of-Gates-Based, 10 MIPS 16-Bit RISC Processor Testbed for Failsafe Applications

Michael Jurczyk and Thomas Schwederski
Institute for Microelectronics Stuttgart
Allmandring 30a, 7000 Stuttgart 80
Federal Republic of Germany

Abstract

A Sea-of-Gates-based 16-bit RISC processor testbed with a maximum performance of 10 MIPS at a 20 MHz clock rate is described. Starting from a small core requiring only 5.000 gates, features can be added in a flexible manner to obtain various system architectures suited for fail-safe applications. The core has a Load-Store Harvard architecture with 24-bit instructions, a 16-bit data path, and a 2-stage pipeline. The data path contains sixteen 16-bit general purpose registers and a high-speed 16-bit Carry-Select adder. The core version has been fabricated on a 1.2 mm GATE FOREST master. An experimental version with control flow checking, boundary scan capability with integrated pad-test and 100% stuck-fault coverage is in fabrication. Software support includes high-level and RT-level simulators, assembler and PASCAL-compiler.

1. Introduction

More and more safety critical tasks are currently associated with electronic systems, as examplified by the ABS system in cars, fly-by-wire aircraft and driverless trains. For systems with a safe state (such as the power-off of an ABS system), fail-safe systems must be employed that continually check for errors and switch to the safe state if an error is detected. These systems must be self-checking to detect errors during operation [NiN89,NiC88], and they require complete off-line tests to avoid accumulation of undetected errors [Nic88]. Standard methods for on-line checks are hardware duplication and coding. In a duplicated system, all operations are performed by two processors, and their results are compared continuously as in the Sequoia system [Ber88] or the VAXft 3000 [Sie90] (Figure 1a). In a system with coding, inputs and outputs to the system are coded; all operations are performed on the codes as well as on the data, and the correctness of the operation can be derived via the codes [NiN89,NiC88] (Figure 1b). Examples for such codes are parity, Hamming code, Berger code or dual-rail code [SiS85]. While coding generally has a

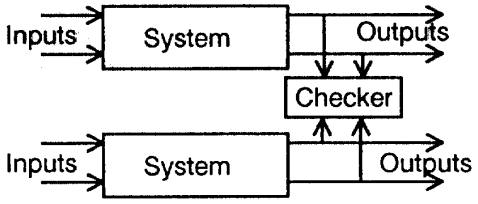

Figure 1 (a) Duplicated system architecture

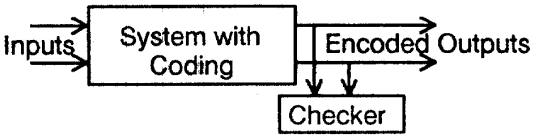

Figure 1 (b) Coded system architecture

lower hardware overhead, it is difficult to implement, especially for ALUs where both arithmetic and logical operations are performed [Tra84,LoR92].
Currently, commercial microprocessors are frequently used in safety-critical applications. Because complete test sets with a guaranteed test quality are generally unavailable for such processors, a high risk of accumulation of undetected faults exists. Also, no provision for fail-safe comparisons such as totally self-checking checkers [HuM84,KuR90] exist in these processors so that on-line tests are problematic. Therefore, specialized fail-safe systems must be used in the future. Advances in fail-safe circuit theory have led to one such fail-safe microcontroller [ChA90]. However, the main drawbacks of the techniques used in this and similar designs are high area overhead due to the fail-safe mechanisms and very long design time because the full custom design style is employed exclusively. To address this problem, fail-safe semicustom integrated circuits are investigated at Grenoble, Milano and IMS within an ESPRIT working group; at the IMS, emphasis is placed on fail-safe Sea-of-Gates structures. The RISC testbed described in this paper is part of these activities. Fail-

1066-1409/93 $03.00 © 1993 IEEE

safe embedded control applications can be supported in a flexible manner by ASIC-based microprocessors that support fail-safe operation. For such a CPU, fail-safe properties should be combined with low complexity and efficient design, with simple additions of specialized interface circuits. In this paper, a RISC processor testbed for such embedded control applications is described; it is used for studies of fail-safe CPU architecture and circuit design, testing, and software support.

In the next section, the microprocessor core is described. Section 3 discusses enhancements for off-line test, while Section 4 examines on-line fault detection via control flow checking. Implementation issues of a core version and an enhanced version are presented in Section 5.

2. Microprocessor core

The processor employs a load-store Harvard architecture [HeP90] with 24-bit wide instructions and a 16-bit wide data path. It can address 64K words of instructions and 64K words of data via 16-bit wide ROM and RAM address busses. The register file consists of 16 dual-port read, single port write registers with register R_0 hardwired to 0. The processor data path architecture is shown in Figure 2. The ALU is used for both data and address manipulations.

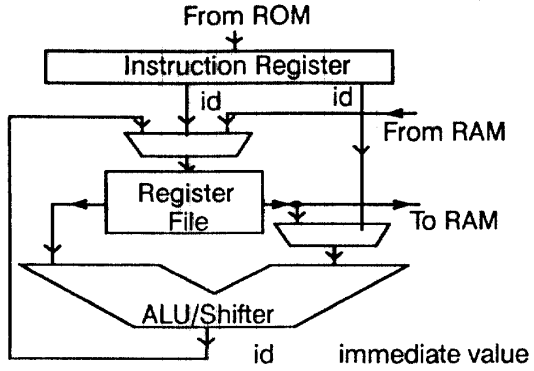

Figure 2 Processor data path architecture

A two-stage pipeline was implemented; subcycles are added during RAM read and write operations to synchronize the processor with memory. This structure leads to a smaller hardware overhead in the control flow checking implementation which is implemented in the enhanced processor (see Section 4). One interrupt level without interrupt nesting is provided to avoid interrupt priority problems and stack overflows; this conforms with the German standards on safety-relevant electronics [DIN801].

To obtain a small and simple processor control path, an instruction set with a low instruction count and high regularity was utilized. The set is divided into four groups: 3-register ALU operations, load/store instructions, conditional branches and program status word changes, and subroutine and stack handling. The ALU operations encompass integer addition and subtraction, six logic functions and one-bit shift and rotate operations. The overall instruction set contains 70 different instruction types and supports four different data addressing modes (immediate, absolute, indirect and indexed).

A tow-down design strategy was employed, starting from a behavioral processor model which was then transformed into a register-transfer-level model. A netlist was derived and automatically placed and routed on the IMS Gate Forest [BeH88,KeB89]. To both validate the processor function and perform a high quality test, a self-test program was developed which consists of 340 instructions with 130 memory accesses, resulting in an overall test time of approx. 37 µs at 20 MHz clock rate. The test requires observation of the outputs which can be achieved via boundary scan (see Section 3). Test vectors for the CPU were derived by executing the test program via the RT-model and verifying the fault coverage by using a deterministic fault simulator; the resulting program achieves 100% stuck-at fault coverage at the gate level. A high-level and a register transfer level simulator, an assembler and a PASCAL-compiler including extensions for hardware error detection during program execution (see Section 4) were developed simultaneously to the processor design.

3. Processor enhancement for off-line tests

To avoid fault accumulations, the processor has to be tested periodically. As discussed in the previous section, execution of a test program detects all stuck-at faults which requires that the processor outputs be observed. This can be accomplished with boundary scan which supports efficient board and system test as well. Therefore, the IEEE 1149.1 compatible SUNBAR boundary scan architecture with built-in pad test [BuB92,ScB91] was included in the enhanced processor version. To improve processor testability, tests for other fault models including shorts and stuck-open faults [LeK93] via I_{DDQ}-Testing, and enhanced scanpath testing are considered.

4. Processor enhancement for on-line tests

To avoid some of the hardware overhead required by a fully duplicated system, control flow monitoring

[ScS87,WiS87,MaM88,DeS91,Wil91] of the instruction ROM is supported, resulting in the architecture of Figure 3. The continuous signature monitoring technique (CSM) [WiS88] is implemented to detect corrupted instructions, incorrect instruction fetches as well as the execution of an incorrect instruction sequence.

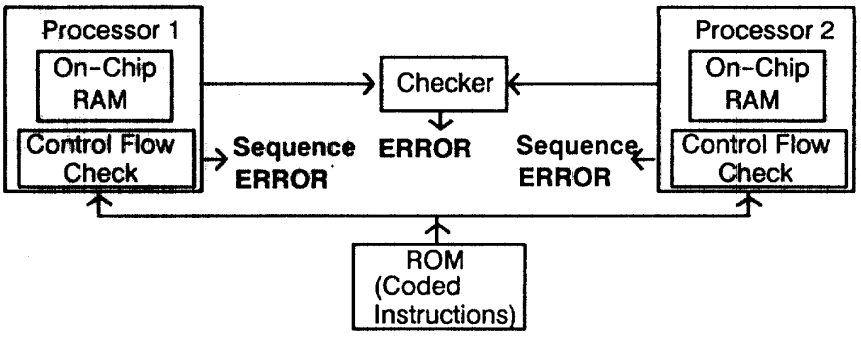

Figure 3 Fail-safe processor system model with control flow check

In the CSM technique, the horizontal signatures of each instruction is formed by generating the signature from the beginning of a program block up to each instruction within the block and across block boundaries as shown in Figure 4. The h-bit wide signatures

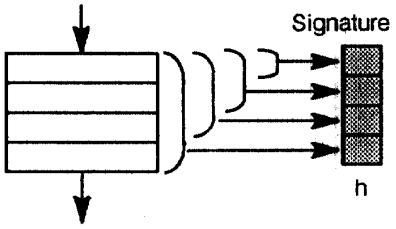

Figure 4 Horizontal signaturing

are associated with the instructions in the ROM. During program execution the processor calculates the horizontal signature and compares it with the stored signature to detect sequence errors. Errors during

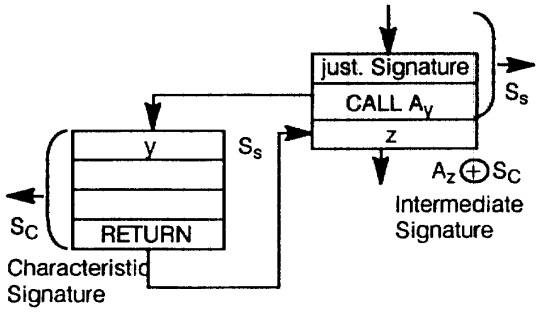

Figure 5 CSM subroutine handling

the execution of conditional branches, subroutine calls and returns can be detected by adding justifying signatures into the program [WiS87] and by using the return address during the subroutine execution [WiS88] as shown in Figure 5.

Signatures are compared during each instruction execution in the CSM technique so that the average latency L of the CSM approach with h bit signatures can be approximated by $L = [\, 2^{-h} / (1 - 2^{-h}) \,]^2$ [WiS88]. An error occurring at the beginning of an instruction block with the length i is detected with the probability $1 - 2^{-ih}$. In the extended processor, 8-bit wide signatures are used which results in an average latency of $1.5 * 10^{-5}$ instruction executions and an error detection probability of 99.6% during the error occurrence and a probability of 99.998% during the fetch of the following instruction. Because 32-bit ROM is used instead of two 24-bit ROMS for the duplicated system, only 66% of the hardware required for a duplicated ROM solution is needed for the CSM technique.

5. Processor implementation issues

To obtain short design cycles in conjunction with fast production time and low cost, the processor core was designed in the 1.2 µm GATE FOREST sea-of-gates environment [BeH88,KeB89,KeS90]; the GATE FOREST is characterized by fast turn around Direct-Write-Electron-Beam personalization [BeC89]. The core chip uses only 5.000 gate equivalents, has a chip area of 49 mm^2, 85% master utilization and a performance of 10 MIPS at a clock rate of 20 MHz. A photo of the core chip is shown in Figure 3 while the gate equivalents of the major core components are listed in Table 1. The processor design was validated in a test board. An enhanced version was implemented with boundary scan and the CSM control flow checking method with 8-bit signatures (see Section 4). Figure 4 shows the processor chip plot; gate equivalents of the processor parts are listed in Table 2. The enhanced processor design is pin-limited and has

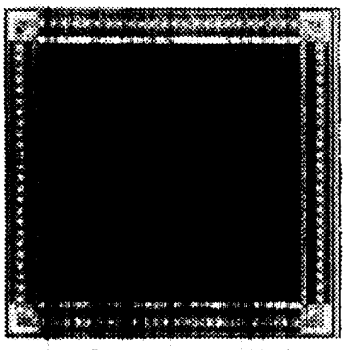

Figure 3 Core-processor chip photo

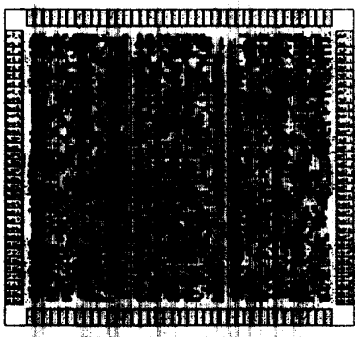

Figure 4 Chip plot of the enhanced processor version

Core Parts	Gate-Equiv.	Transistors
Data Path	3.900	15.600
Control Path	1.100	4.400
Overall Design	5.000	20.000

Table 1 Gate-equivalents of the different core parts

Design Parts	Gate-Equiv.	Transistors
Processor Core	5.000	20.000
Sequence Check	1.250	5.000
Boundary Scan	3.750	15.000
Overall Design	10.000	40.000

Table 2 Gate-equivalents of the enhanced processor version

a chip area of 110 mm^2 at 60% master utilization; it uses 10.000 gates and has a peak performance of 10 MIPS. The hardware overhead of the control flow check implementation is 25% compared to the core hardware which is acceptable because in the fail-safe processor system shown in Section 4, the ROM duplication is substituted by instruction coding which leads to a hardware reduction of 34%. In the processor design, the boundary scan part is implemented as soft macro in the core logic and consists of 15.000 transistors. This overhead corresponds to the high count of 98 I/O pins, including 16 bidirectional pins. Currently, the boundary scan part is implemented in full-custom as part of the master which will lead to a significant overhead reduction [BuB92].

7. Summary

A 16-bit 10 MIPS Sea-of-Gates RISC processor testbed was described. A control flow checking method ensures the detection of errors in the program flow with an error detection probability of 99.998% after two instruction fetches and an average error detection latency of $1.5*10^{-5}$ instruction fetches. The SUNBAR boundary scan architecture with built-in pad test enhances processor and system testability. The processor design can be tested for single stuck-at faults with 100% fault coverage by executing a test program on the processor and by observing the processor output signals via boundary scan.
High-level and register transfer level simulators, an assembler and a PASCAL-compiler for the processor are available.

References

[BeC89] M. Beunder, G. Chen, B. Hoefflinger, J. Kernhoff, W. Haas, M. Schau, T. Schwederski, and R. Springer' "Advanced semicustom design and fabrication at the Institute for Microelectronics Stuttgart," *UGIM '89*, June 1989.

[BeH88] M. Beunder, B. Hoefflinger, and J. Kernhof, "New directions in semi-custom arrays," *IEEE Journ. Solid-State Circuits*, Vol. 23, No. 3, June 1988, pp. 728-735.

[Ber88] P. A. Bernstein, "Sequoia: A fault-tolerant tightly coupled multiprocessor for transaction processing," *IEEE Computer*, Vol. 21, No. 2, February 1988, pp. 37-45.

[BuB92] T. Buechner, E. Bernath, R. Gurkasch, T. Schwederski, and H. Werkmann, "SUNBAR - A universal boundary scan architecture for a sea-of-gates semi-custom environment." *to be published in ESSCIRC 92 Proceedings*, 1992.

[ChA90] G. Chaumontet, V. C. Alves, M. Nicolaidis, A. Guyot, and B. Courtois, "A fail-safe microcontroller for railway signalling," *ESSCIRC 90*, September 1990

[DeS91] X. Delord and G. Saucier, "Formalizing signature analysis for control flow checking of pipelined RISC microprocessors ," *Proc. International Test Conference*, 1991, pp. 936-945.

[DIN801] DIN V VDE 0801/01.90, "Grundsätze für Rechner in Systemen mit Sicherheitsaufgaben," 1990

[HeP90] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann Publishers, INC., San Mateo, CA 94403, 1990.

[HuM84] J. L. A. Hughes, E. J. McCluskey, and D. J. Lu, "Design of totally self-checking comparators with an arbitrary number of inputs," *IEEE Transactions on Computers*, Vol. C-33, No. 6, June 1984, pp. 546-550.

[KeB89] J. Kernhof, M. Beunder, B. Hoefflinger, and W. Hass, "High-Speed CMOS adder and multiplier modules for digital signal processing in a semicustom environment," *IEEE Journ. Solid-State Circuits*, Vol. 24, No. 3, June 1989, pp. 570-575.

[KeS90] J. Kernhoff, M. Selzer, M. A. Beunder, B. Hoefflinger, B. Laquai, and I. Schindler, "Mixed static and domino logic on the CMOS Gate Forest," *IEEE Journal of Solid-State Circuits*, Vol. 25, No. 2, April 1990, pp. 396-402.

[KuR90] S. Kundu and S. M. Reddy, "Embedded totally self-checking checkers: A practical design," *IEEE Design & Test*, Vol. 7, No. 4, August 1990, pp. 5-12.

[LeK93] J. Leenstra, D. Keck, M. Koch and T. Schwederski, "On scan path design for stuck-open and delay fault detection," *submitted to the 1993 European Test Conference*, 1993.

[LoR92] J.-C. Lo, S. Thanawastien, T. R. N. Rao and M. Nicolaidis, "An SFS Berger check prediction ALU and its application to self-checking processor designs," *IEEE Transactions on Computer-Aided Design*, Vol. 11, No. 4, April 1992, pp. 525-540.

[MaM88] A. Mahmood and E. J. McCluskey, "Concurrent error detection using watchdog processors - A survey," *IEEE Transactions on Computers*, Vol. 37, February 1988, pp. 160-174.

[NiC88] M. Nicolaidis and B. Courtois, "Strongly code disjoint checkers," *IEEE Transactions on Computers*, Vol. 37, No. 6, June 1988, pp. 751-756.

[NiN89] M. Nicolaidis, S. Noraz, and B. Courtois, "A generalized theory of fail-safe systems," *FTCS-19*, 1989, pp. 398-406.

[Nic88] M. Nicolaidis, "A unified built-in-self-test Scheme: UBIST," *FTCS-18*, 1988, pp. 157-163.

[ScB91] T. Schwederski, T. Buechner, J. Leenstra, G. Roos, and L. Spaanenburg, "Built-In pad test with boundary scan," *Proc. European Test Conference*, 1991, pp. 385-392.

[ScS87] M. Schuette and J. P. Shen, "Processor control flow monitoring using signatured instruction streams," *IEEE Transactions on Computers*, Vol. C-36, No. 3, March 1987, pp. 264-276.

[SiS82] D. P. Siewiorek and R. S. Swarz, *The Theory and Practice of Reliable System Design*, Digital Press, Bedford, Mass., 1982.

[Sie90] D. P. Siewiorek, "Fault tolerance in commercial computers," *IEEE Computer*, Vol. 23, July 1990, pp. 26-37.

[Tra84] W. M. Trautwein, "Concurrent error detection/correction of logical operations" in *Fehlertolerierende Rechensysteme/Fault-Tolerant Computing Systems*, Springer-Verlag, Berlin, W.-Germany, 1984, pp. 189-194.

[WiS87] K. D. Wilken and J. P. Shen, "Embedded signature monitoring analysis and technique," *Proc. International Test Conference*, 1987, pp. 324-333.

[WiS88] K. D. Wilken and J. P. Shen, "Continuous signature monitoring: Efficient concurrent-detection of processor control errors," *Proc. International Test Conference*, 1988, pp. 914-925.

[Wil91] K. D. Wilken, "Optimal signature placement for processor-error detection using signature monitoring," *FTCS-21*, 1991 , pp. 326-333.