VOLTaiRE: A Fault Tolerant VLIW Design With On-line Property Checking for Reconfigurable Datapaths

ABSTRACT

Verification and reliability are emerging as bottlenecks in processor design as aggressive technology scaling and continued architectural innovations enable to perform high speed complex operations. In this context we present VOLTaiRE, a "**V**LIW processor with **O**n**L**ine **T**esting and **RE**configurability" that guarantees reliable functionality of the execution units even in the presence of manufacturing, logical and transient errors. VOLTaiRE has online property checkers that monitor the correctness of the execution units using novel property checking techniques. If any of the units are identified to be faulty, the processor ceases to use that unit and reconfigures the data path to function with the other units. VOLTaiRE trades off performance for accuracy in the reconfigured mode. We have quantified the coverage provided by the online checkers and 92% coverage for all the execution units is assured. The chip has been designed in 0.18um TSMC process using semi-custom and synthesized blocks. Simulations indicate that the overhead associated with the property checkers in terms of area and power is found to be minimal.

1. INTRODUCTION

Feature size reduction and innovative architectural changes have made it possible keep pace with Moore's law and build complex designs that operate at very high speeds. At the same time, logical errors due to insufficient verification, process variations, time dependant transistor breakdown, soft error strikes and operating conditions can cause an unanticipated breakdown of the processor. Hence verifying such a complex design and ensuring that the mean time to failure(MTTF) is at an acceptable level is a challenging task.

Functional Verification constitutes about 70% of the time and resources in a chip design. Formal, Semiformal tools[4, 5] along with random simulation and directed testing are used to functionally prove the correctness of the design. However, ensuring 100% functional coverage is difficult. Only a few selected chips go through rigorous testing after fabrication to check for manufacturing defects. Designers do not always have control on the environment in which the processor runs nor are involved in the entire process flow from conception to delivery to assure the correct functionality of the processor to the end user. Either an amount of time has to be spent to exhaustively verify all possible features of the design and miss the time-to-market window or release a chip that fails in hostile environments due to manufacturing defects that are difficult to control.

VOLTaiRE addresses the above mentioned concerns for ensuring correct operation of execution units on a VLIW processor. A VLIW processor has multiple computational resources for increased performance[3]. It is innovative in two aspects.

- It features an online property checker that continues to verify the processor even after it is released to the customer. The advantage with online checking is that a processor is verified against inputs that matters to the user as opposed to exhaustively verifying the processor against an enormous input vector space that may never be used.
- VOLTaiRE reconfigures itself to respond to failures in the design. It uses the multiplicity of VLIW architecture and uses the correctly functional units to do the computations.

This paper is organized as follows. Section 2 discusses the various approaches for online testing and the relative merits of these techniques. An architectural overview of the system in the context of property checkers is presented in Section 3. Section 4 talks about the property checkers and the reconfigurability aspect. Section 5 discusses the heuristics developed for measuring the coverage provided by the checkers. The results are detailed in Section**??**. This section also provides chip statistics and floor planning. We conclude in Section 7.

2. RELATED WORK

In the context of online testing of processors, various concurrent error detection(CED) schemes have been proposed in literaturequote mitra02. All the CED schemes incorporate a output characteristic predictor that predicts some special characteristic of the output and a checker that compares the expected behaviour with that of the actual output and generates an error when a mismatch is detected. Duplication is the simplest way of detecting errors. It has been suggested in mitra that diverse implementation of the logic is more robust than duplication as this offers protection against common-mode failures. Another solution proposed in the direction of online testing are the Berger codesquote berger 61 that can detect all unidirectional errors and Bose-lin codes quote bose 85 that can detect upto t unidirectional errors. These codes are suitable for the reliability of systems that have large occurence of one type of binary numbers. However, these codes are not very easy to be used for online testing of datapaths as they place unneccessay constraints on hardware like invertor-free logic and restriction on the fanout of the gates to ensure that all errors that occur are unidirectional.

Parity prediction schemes have been used for ensuring fault-secure datapaths nicolaidas 97. Here the multipliers and ALUs are constructed using fault-secure full-adders and half-adders that act as building blocks. The inherent drawback of the unidierctional codes and parity prediction techniques is that they place heavy restrictions on the implementation of the hardware resulting in an area overhead that compares to or in some cases greater than that of duplicationquote das98.

Further these hardware restrictions have a negative impact on the performance. Better than worst case design solutions have proposed by RazorErnst 03 and DIVA weaver 01. Razor provides correction to only timing related errors under process variation and does not address permanent defects or TDB. DIVA on the other hand uses

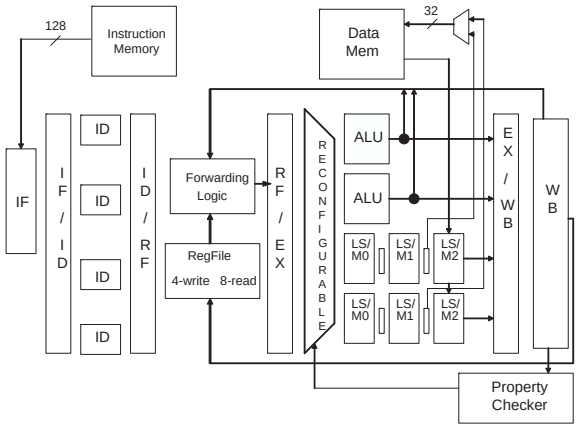

Figure 1: Block Diagram of the reconfigurable VLIW processor based on alpha architecture.

variuos redundancy techniques to ensure correctness thus consuming a lot of area and power.

VOLTaiRE is different from previous approaches as it can detect and respond to a wide range of defects at minimal area and power overhead. The on-chip property checkers not only check for functional bugs due to logical errors, soft errors, and TDB but also for manufacturing faults that have escaped the post-silicon detection. Further, VOLTaiRE does not need to replicate the entire architecture but instead, embed sufficient properties on chip to check for reasonable correctness. Also, it takes advantage of VLIW architecture to do reconfiguring rather than relying on an extra simple processor or a redundant unit.

3. ARCHITECTURE

VOLTaiRE is a 4-way, 32-bit fixed point VLIW processor based on the single-pipelined 64-bit RISC ALPHA processor as shown in Figure 1. The pipeline stages that exist in VOLTaiRE are the Instruction Fetch (IF), Instruction Decode (ID), Register File Read (RF), multi-stage Execution (EX), and Write-back (WB). Each stage is discussed in detail in the subsections 3.1 - 3.5

VOLTaiRE implements a useful subset of the full Alpha instruction set, without the special puspose instructions. Each VLIW instruction is 128-bit long consisting of 4 independent 32-bit instructions. In VLIW architectures, the compiler is responsible for generating machine code grouping independent instructions into one 128-bit bundle. VOLTaiRE has 2 identical pipelines for carrying out ALU instructions and 2 for the Load/Store or Multiply instructions. Each bundle can be filled with up to 2 ALU instructions and 2 Load/Store or Multiply instructions. In the case where independent instructions are not available to be placed in the bundle, a NOP instruction is used. The number of stages in the VLIW pipeline adjusts appropriately to the type of instructions being fetched and executed. The ALU instructions have a 5 cycle latency whereas the load/store and multiply instructions have a 7 cycle latency.

3.1 Instruction Fetch

The IF stage is responsible for fetching the 128-bit VLIW instruction from the instruction memory. It does a one-cycle fetch from the instruction memory with the current PC (program counter) and passes the instruction to the ID stage. A IF control block is used to transfer external data into the Instruction Memory.

3.2 Instruction Decode

The ID stage decodes 4 independent instructions per cycle. It has 4 decoders which are built using original instruction decoders from the Alpha architecture. The decoders issue 4 instructions simultaneously into the pipelines.

3.3 Register File

The RF stage stores the operands for the instructions. The RF read is usually done in the ID stage. However, in VOLTaiRE, this has been made into a separate pipeline stage for overcoming the longer latency associated with a 4-write 8-read RF. The 8 read ports are necessary because each instruction needs at most 2 reads and 4 instructions are executed in parallel in every cycle. The 4 write ports are needed to facilitate the 4 instructions retiring and writing their results simultaneously.

The Forwarding unit is responsible for forwarding EX results to the RF stage if there are dependent instructions across the VLIW instruction bundles. The forwarding lines form a matrix of routing because results from each of the four EX units must be checked for dependencies on each read of the next instruction in the RF stage.

3.4 Execute Stage

The EX stage consists of 2 ALU and 2 LSM (Load/Store/Multiply) units. The multiplier used in the VOLTaiRE processor is 16-bit wide as opposed to the ALU and Load/Store units which are 32-bits wide. ALU instructions take one cycle to execute while Load, Store, and Multiply (LSM) instructions take three cycles. This can be seen in Figure 1 where the LSM units are separated by two internal pipeline registers. The VOLTaiRE architecture can be viewed as four separate, single-issue pipelines. When there is a LSM instruction, the LSM pipelines will stall for two cycles to allow it to complete executing while the ALU instructions progress forward.

The reconfigurable component lies in the EX stage. It is responsible for switching the data paths between the 2 ALU and/or the 2 LSM units. Naturally, data paths cannot be interchangeable between an ALU and LSM unit.

3.5 Write Back Stage

The Write back stage writes the results of the EX stage into the register file. It checks if the instruction that passed through the EX stage was a valid instruction and an instruction that has data to be written (i.e., not a store). Also, the Write back stage provides an extra buffer zone for the property checkers to validate the results of the EX stage. Because data is not written back to the register file until the property checker validates the data, execution of instructions up to the write back stage can be seen as speculative.

4. PROPERTY CHECKERS

Properties are certain conditions that are always true during the execution of the processor. This could be relationship of output to inputs for logic blocks, timing constraints or a combination of both. These are bundled up in a structure called a property checker. Typically, a designer would incorporate these checks in the functional design stage and integrate it with the design. They can be used at various points in the design, viz.:

- At the logic block level
- Between logic boundaries
- At the system level

Property checkers are helpful in achieving high fault coverage without the need of long simulation cycles. The method of traversing the design space of a logic block helps in uncovering hard to find bugs, at the cost of occasional flagging of false negatives. This can be overcome by careful design considerations of the checkers.

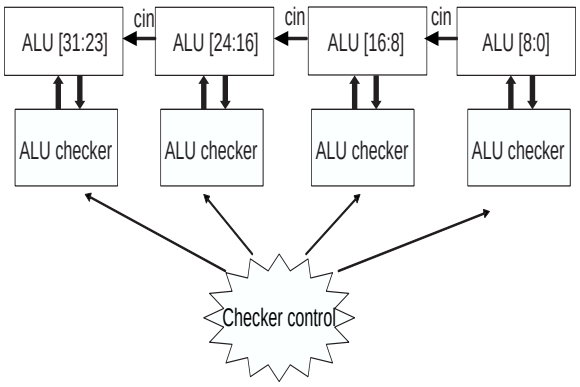

Figure 2: ALU property Checker

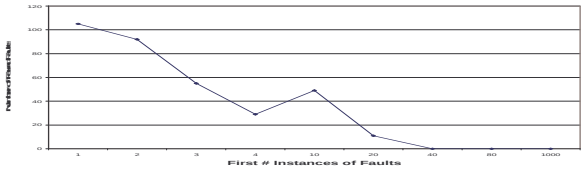

Figure 3: First catch trend of ALU property Checker.

ALU	Total	Found	Hit Rate
Add st@0	343	341	99.42%
Add st@1	335	333	99.40%
Xor st@0	57	57	100
Xor st@1	39	39	100

Table 1: ALU Property Checker Coverage.

The novel feature of VOLTaiRE is the use of property checkers for verifying the functionality of internal units. For online verification, special properties were derived to verify the results of the execution units. Typically the design is stressed out by using formal engines before tape out. If the design is very large, complex formal engines run into the BDD (binary decision diagram) blow up problems while proving properties. However, by having the properties on chip we can validate if the design is correct in the input space where the user is interested in and, due to the ability to reconfigure, rectify the design.

The property checkers are in charge of flagging faulty execution units. They check the results of the EX units based on arithmetic properties that can be used to verify them.

It is very easy to ensure the correctness of the checkers. As the checkers are trivial they can be completely functionally verified either by random simulation or formal tools[6]. The small and simple designs of the property checkers lend themselves to straight-forward design verification. The checkers are very small and are physically placed away from the execution units to reduce the probability of manufacturing defects occurring in both the structures.

4.1 ALU Checker

The ALU Property Checker verifies the results of the ALU by checking smaller bytes of the result. VOLTaiRE uses a 32-bit ALU in the EX stage. In contrast, the property checker is only a 9-bit ALU. As shown in Figure 2, in one instruction cycle, it checks the lower 9-bit result (alu_result[8:0]) of the 32-bit result. In the next cycle, it will check the next 9-bit nibble of the result (alu_result[16:8]) with an overlap of one bit. This window of verification moves up the 32-bit ladder until it reaches the top (alu_result[31:23]) and then resets to the initial lowest 9-bits, starting from the bottom again. By overlapping the result-checking by one bit or more, the carry logic of additions does not have to be implemented in the property checker making it quite small. The same ALU Property Checker is also used to validate the Load/Store address calculations which are always additions of the memory displacement to the base address. Hence, rather than having another redundant ALU unit that takes up more area and power, the ALU Property Checker is much smaller and simpler to verify.

4.2 Multiplier Property Checker

4.2.1 Boundary Checker

The Multiplier Boundary Property Checker requires additional hardware as it is more challenging to verify. The strategy used for the ALU Property Checker is not feasible for the Multiplier Property Checker. A multiply instruction has two operands and one product. The Multiplier Property Checker approximates each operand to the closest power of two. Thus, if we are multiplying by 5, the checker will multiply by 4, a lower bound, and by 8, an upper bound. Note that these multiplies are powers of two so in hardware, they are done using shifters. This technique is done on both operands and since two multiplies are actually done per operand, 4 different multiplications are done per multiply operation. The product of the multiplier is then compared to the lower and upper bound results that was computed using shifts.

4.3 Reconfigurable Data path

The status of each functional unit is stored in the Program Status Register (PSR). When any of the functional units are faulty the corresponding bit in the PSR goes high. As ALU instructions are single cycle, if one of them breaks down, we stall every other cycle to allow the other functional ALU to take on the extra burden. On the other hand, if one of LSM units breaks down, VOLTaiRE suffers an additional 3-cycle loss. The reconfigurable data path is implemented in the EX stage. To reduce the critical path delay most of the controls needed for reconfiguring are generated in parallel in the RF access stage.

The PSR will only be changed if an execution unit faults twice while the chip is on. In this way, transient faults in the checker or in processor will not wrongly set the PSR and hamper the processor performance. The first time a fault in an execution unit is flagged by the property checkers, the instruction is sent back to the IF stage and re-executed. The faulty instruction is inherently treated as a mis-predicted branch. If the execution unit fails a second time, then the PSR bits will be updated and the faulty execution unit will be flagged. At this point, the data path is reconfigured.

5. COVERAGE

Quantifying property checkers is an intricate task that is an essential element when developing on-line property checking architecture as explored in this project. First, the ALU and multiplier used in the processor were synthesized separately from the pipeline. Their netlists were than used as basis for fault injection. A Perl script was written to inject stuck-at-0 and stuck-at-1 faults into the netlists one at a time, on every wire in the netlist. Once a fault was injected into the netlist, a Verilog testbench was run using the ”buggy” netlists. The Verilog testbench ran 1000 random test inputs on the execution units. The number of valid faults (not all inputs caused errors in the results), faults caught by the property checkers, and first times the faults were caught were recorded. Tables 1 and 2 show the results of the simulations.

In the Table 1 and 2, the ”Total” column is the total number of

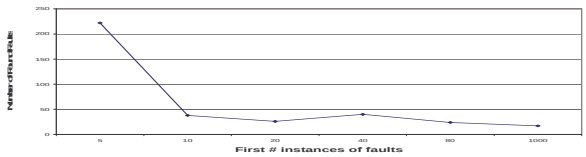

Figure 4: First catch trend of Multiplier property Checker

Multiplier	Total	Found	Hit Rate
16 bit st@0	1034	406	39.26%
16 bit st@1	1023	573	56.01%
8 bit st@0	396	367	92.68%
8 bit st@1	931	931	100%

Table 2: Multiplier Property Checker Coverage.

faults that caused an error in the result of the ALU and multiplier, respectively. The ”Found” column shows how many of those faults were actually caught by the property checkers. St@0 mean stuck-at-0 faults and St@1 mean stuck-at-1 faults. In Table 1, add and xor instructions were used. For the faults that were missed in the ALU add instructions, the simulation results showed they appeared no more than 6 times in the total 1000 input space. In Table 2, the first two tests were using all 16-bits of the multiplier operands. The last two tests constrained the operands to 8-bit vectors which increased coverage significantly since for smaller numbers, the property checker gains better accuracy on the upper and lower bounds. It is interesting to point out that simulations were also run using the Multiplier Property Checker but allowing it only to bound the lower operand of the two, not both. This resulted in a hit rate of 55.62% on the 16-bit, st@1 input test, not much lower than the 56.01% attained bounding both operands.

Figures 3 and 4 show the first times the property checkers caught the error in the results. For example, the ALU Property Checker caught 105 faults on their first appearance and the Multiplier Property Checker caught 222 faults by the 5th appearance of the faulty result.

6. VOLTAIRE CHIP STATISTICS

In Table 3, the area and power measurements of the property checkers can be compared to the rest of the pipeline stages. Notice that the property checkers form a minimal part of the pipeline. Also, the execution stage is by far the largest because of the multiple execution units. Figures 6 and 7 show the area and power percentages of each module, respectively.

The layout of VOLTaiRE is shown in Figure 5. The chip is DRC clean at the global level and LVS clean for all the blocks. The clock

Module	Area(mm^2)	Timing(ns)	Power(W)
Execute	1.2	3.10 / 3.58	1.71
Write	0.104	2.54 / 2.98	.075
RF	1.7	2.3	.200
Decode	0.066	1.82	.052
Fetch	0.057	2.23	.029
Clock	0.046*0.050		0.629u
ImemCtl	0.06*0.38		20.39m
DmemCtl	0.25*.066		3.46
Dmem	1.102*0.285		90m
Imem	0.206*2.0		360m
Core	1.317*1.0		1.75

Table 3: VOLTaiRE Statistics.

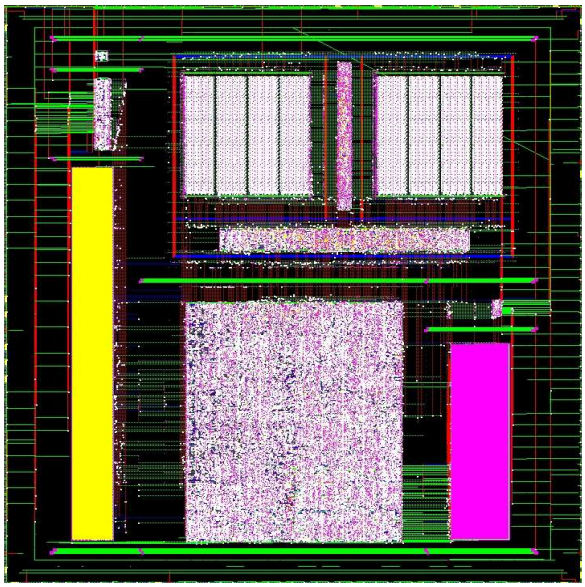

Figure 5: Final Layout of the processor.

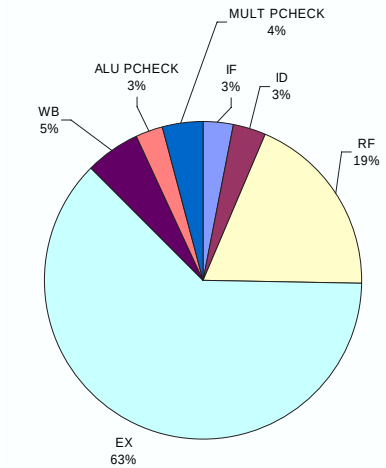

Figure 6: Area percentage breakup of VOLTaiRE processor .

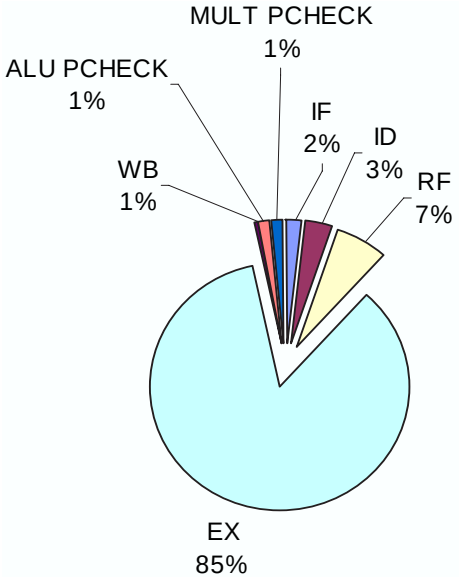

Figure 7: Power percentage breakup of VOLTaiRE processor.

tree at the global level is a first order balanced structure.

The following are the chip statistics of VOLTaiRE:

- The operating frequency is 280MHz
- The operating frequency is 280MHz
- The operating voltage is 1.8 V
- Technology is TSMC 180nm
- The total area of the chip was 2.5mm X 2.9mm.
- Number of PADS =120

7. CONCLUSIONS AND FUTURE WORK

We presented a fault tolerant VLIW processor, VOLTaiRE, that enhances the reliability and reduces verification burden by incorporating online property checkers. The additional minimal overhead and good coverage provided by these checkers prove that having synthesized checkers is an efficient way of testing. Future work could involve extending the data path ”reconfigurability” into this stage as well because of the possible multiple units that can be used. An intriguing behavior of execution units in processors are that they constantly have inputs going into them. For example, even if the multiplier is not used for one cycle, it will have some invalid inputs on it. These unused inputs possibly could increase the coverage of the property checkers if they are also used to verify the execution units. Further research can be done on different multiplier architectures and how they respond to fault injection as well. Finally, additional fault simulation could include IDDQ variations and transition/path delays.

8. REFERENCES

[1] C. Weaver and T. Austin, ”A Fault Tolerant Approach to Microprocessor Design.”, Dependable Systems and Networks (DSN), July 2001.

[2] Scott McFarling, ”Combining branch predictors,” Digital Equipment Corporation WRL Technical Note TN-36, June 1993.

[3] J.L. Hennessy, D. A. Patterson, ”Computer Architecture: A Quantitattive Approach, Third Edition.” Morgan Kaufmann, San Francisco, CA, 2003.

[4] J. Marques-Silva and K. Sakallah. GRASP – A search algorithm for propositional satisfiability. In IEEE Transactions on Computers, May 1999.

[5] K. McMillan. Applying SAT methods in unbounded symbolic model checking. In Proc. of Computer Aided Verification Conference, LNCS vol. 2404, July 2002.

[6] A. Biere, A. Cimatti, E. Clarke and Y. Zhu. Symbolic Model Checking without BDDs. In Proc. of Tools and Algorithms for the Analysis and Construction of Systems, LNCS vol. 1579, 1999