面向车载功能安全的低开销超标量双核锁步处理器架构设计

张承译[1]，王明羽[1]，虞志益[1]，李兆麟[2]

（1. 中山大学 微电子科学与技术学院，广州 510275；
2. 清华大学 计算机科学与技术系，北京 100084）

摘 要：在车载功能安全领域，双核锁步架构是一种被广泛应用于解决处理器故障的冗余架构。为支持细粒度故障处理的超标量处理器提出一种新颖的双核锁步架构，通过以分支跳转指令的形式执行程序回滚，该架构能在故障发生的同一时钟周期内检测和纠正故障，且不需要额外的专用硬件模块来满足细粒度回滚的需求。还提出一种虚拟写回机制，该机制将特定数据传送到只读寄存器以防止故障衍生，使处理器无需在程序执行期间持续保存现场，从而显著节省了面积开销。试验结果表明，该架构对注入处理器的故障实现了较彻底的故障覆盖，对处理器原型的性能影响很小，与先前双核锁步相关的工作相比，时间和面积开销更小。

关键词：双核锁步；处理器；故障处理；程序回滚

中图分类号：TP332 **文献标志码：**A **DOI：**10.3969/j.issn.2095–1469.2024.02.15

Design of a Low-Overhead Superscalar Dual-Core Lockstep Processor Architecture for Automotive Functional Safety

ZHANG Chengyi[1]，WANG Mingyu[1]，YU Zhiyi[1]，LI Zhaolin[2]

（1. School of Microelectronics Science and Technology，SUN Yat-sen University，Guangzhou 510275，China；
2. Department of Computer Science and Technology，Tsinghua University，Beijing 100084，China）

Abstract: In the field of automotive functional safety, the dual-core lockstep (DCLS) architecture is a redundancy architecture widely used for addressing processor faults. This paper proposes a novel dual-core lockstep architecture for superscalar processors that supports fine-grained fault handling. By executing program rollback in the form of a branch instruction, the proposed architecture can detect and correct faults within the same clock cycle they occur, without the need for additional hardware support. Furthermore, the virtual writeback (VW) mechanism is also presented, which feeds specific data to read-only registers to prevent fault propagation. This allows the processor to avoid continuous context saving during program execution, which reduces area overhead significantly. The experimental results show that this architecture

收稿日期：2023–05–05 **改稿日期：**2023–06–05

基金项目：国家重点研发计划项目（2020YFB1600202）；广东省重点领域研发计划项目（2021B0101410004）；广东省自然科学基金项目（2022A1515011708）

参考文献引用格式：

张承译，王明羽，虞志益，等. 面向车载功能安全的低开销超标量双核锁步处理器架构设计[J]. 汽车工程学报，2024，14（2）：313–320.

ZHANG Chengyi，WANG Mingyu，YU Zhiyi，et al. Design of a Low-Overhead Superscalar Dual-Core Lockstep Processor Architecture for Automotive Functional Safety[J].Chinese Journal of Automotive Engineering，2024，14（2）：313–320.（in Chinese）

achieves more thorough fault coverage with minimal impact on the processor performance, while exhibiting reduced latency and area overhead compared with the DCLS-related previous work.

Keywords: dual-core lockstep; processor; fault handling; rollback

功能安全是系统检测、诊断和纠正故障的能力，该能力对于面向自动驾驶应用的车载系统至关重要[1-3]。锁步是一种用于实现功能安全的高效容错机制[4-6]，作为一种流行的锁步架构，双核锁步（Dual-Core Lockstep，DCLS）是实现其他锁步架构的重要基础[7-9]。在DCLS架构中，两个相同的处理器核执行同一段应用程序，执行结果会输入若干个检查器模块进行验证，处理器将根据验证结果执行故障处理流程[8, 10]。

经典的DCLS处理器锁步过程包括两个步骤：故障检测和纠正。回滚是一种常见的故障纠正方法，该方法通过回到处理器的历史状态使系统从故障中恢复。进一步地，回滚机制可以通过3种方式实现：系统复位、指令级回滚和流水线级回滚。系统复位实现简单，但时间开销最大，在双核锁步架构中通常作为辅助性的回滚方法[11]。为了避免复位后从起始位置开始执行程序所带来的时间开销，指令级回滚为处理器执行的程序设置若干检查点，并在发生故障时将系统恢复到上一个检查点[12-15]。然而，指令级回滚需要中断流水线并至少等到该故障指令执行结束才能开始故障检测和纠正的过程，这依然会导致不小的时间开销[14]。例如，文献［15］中提出的DCLS处理器在100 MHz时的故障处理延迟高达微秒级。

流水线级回滚能在一条指令执行完所有流水线步骤之前进入回滚的过程，从而实现更加细粒度的故障处理[16-18]。为了实现流水线级回滚，系统需要在每个时钟周期保存寄存器现场，这需要投入更多的硬件资源并导致更大的面积开销。例如，文献［16］中提出的采用流水线级回滚进行故障处理的DCLS架构需要额外两个时钟周期来保存和重新加载寄存器现场，与处理器原型相比，面积开销增加了300%。文献［17］中的设计进行了面积开销的优化，然而这一开销在该设计中仍高达297%。此外，目前引入流水线级回滚机制的处理器原型基本为单发射处理器，不能满足高性能应用的需求。例如，文献［18］中的设计采用的处理器原型是RISC-V单发射五级流水线处理器。

近年来，车载处理器对性能和可靠性的要求不断提高，为了平衡高性能和功能安全，本文提出了一种用于超标量处理器的低开销DCLS架构设计。具体来说，该设计的主要贡献如下。

（1）流水线级回滚被创新性地引入DCLS超标量处理器，实现高性能处理器细粒度的故障检测和纠正。

（2）程序回滚被视为伪分支跳转指令，帮助实现流水线级回滚并加速故障检测和纠正的过程。

（3）提出了虚拟写回（Virtual Writeback，VW）机制防止超标量处理器中特有的故障衍生，且DCLS处理器无需在执行过程中不断保存现场，从而显著节省了面积开销。

1 超标量DCLS处理器架构

本文提出的超标量DCLS处理器整体架构如图1所示。该DCLS架构基于一个基本的超标量处理器原型[19]搭建，该处理器是一个具有六级流水线的超标量双发射RISC-V处理器。六级流水线包括：预测下一条程序指针（PC级）；取指令（fetch）；在两个通道中并行解码（issue）；执行计算并为内存和寄存器访问准备数据（ALU）；访问内存（MEM）；将数据写回寄存器（WB）。前两级流水线阶段属于前端，后四级流水线阶段属于后端。本文提出的DCLS架构引入的是处理器核级[20]的DCLS机制，内存模块则采用ECC编码校验。

检查器配置在流水线的最后4个阶段中的每1个阶段。在发生故障时，检查器产生相应的程序指针“fault PC”和故障标志信号“fault”。这些信号通过故障信息总线（Fault Information Bus，FIB）

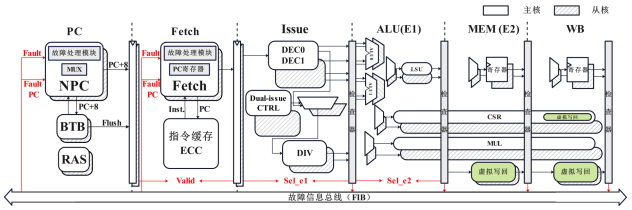

图 1　超标量 DCLS 处理器架构

发送到故障纠正模块，使处理器恢复正常运行。检查器输出的“fault PC”传输到位于 PC 和 fetch 阶段的故障处理模块中进行进一步诊断，从而确认故障的种类以及对应的流水线通道和阶段，PC 和 fetch 阶段会据此设置新的指令执行起点。同时，NPC 模块和分支目标缓冲区（Branch Target Buffer，BTB）会帮助刷新流水线以防止故障传播。MEM 和 WB 阶段引入 VW 机制，通过将故障数传送到只读寄存器，防止故障衍生。此外，这种机制不需要保存寄存器现场，这大大减少了面积开销。通过上述方式，所提出的设计能支持超标量处理器的快速和细粒度回滚。

2　微架构电路设计

2.1　故障检测电路设计

如图 2 左侧所示，来自各个流水线阶段的故障相关信号“fault”经由 FIB 向优先级仲裁器发送具有不同优先级的故障信息。比如，如果在某个时刻同时有两个故障信号传入仲裁器，一个来自译码（issue）阶段，一个来自访存（MEM）阶段，那么仲裁器会先处理来自访存阶段的故障信号；而如果一个故障信号来自双发射处理器中的通道 0，另一个来自通道 1，那么仲裁器会优先处理来自通道 0 的故障信号。上述处理能确保系统回滚至更早执行过的发生故障的程序，从而帮助彻底解决系统中存在的所有故障。“fault”也被传递到故障计数器，如果故障的持续时间超过可容忍的阈值，该信号将触发“rst”信号以重置系统。

检查器的目标是使用尽量短的组合逻辑关键路径快速识别故障。如图 2 右侧所示，主从流水线寄存器的待检测信号被直接送入一组异或门进行按位比较，异或门的输出为“fault”信号。该信号被用作标志信号来控制前端 PC 分支跳转和数据向下一个流水线阶段的传输。

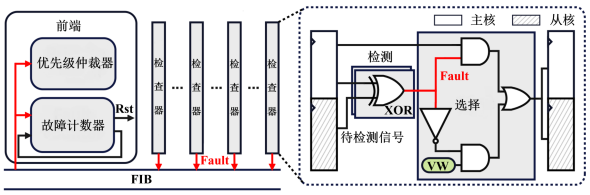

图 2　检查器电路和故障相关信号的数据通路

2.2　前端故障纠正电路设计

基于流水线级回滚的纠正电路设计分为前端和后端两部分。流水线级回滚的优势是无需等待处理器到达 WB 阶段再进行故障纠正，从而加快故障纠正速度，提高性能。如图 3 所示，前端故障纠正电路将 FIB 下发的故障 PC 传送给位于 NPC 和 fetch 模块的故障处理模块，使系统跳转到故障指令并重新开始执行该指令。前端纠正电路还冲刷流水线，保证错误指令不会执行到后端。该设计将回滚与分支跳转和分支预测相结合，将回滚视为一种“伪分支跳转指令”，以满足流水线级回滚的要求。但是，与分支（J 型）指令不同，故障是随机的、不可预测的，因此，回滚必须通过强制性的分支跳转和刷新来实现。实现上述设计需要分别在 PC 和 fetch 阶段修改 NPC 和 fetch 模块。

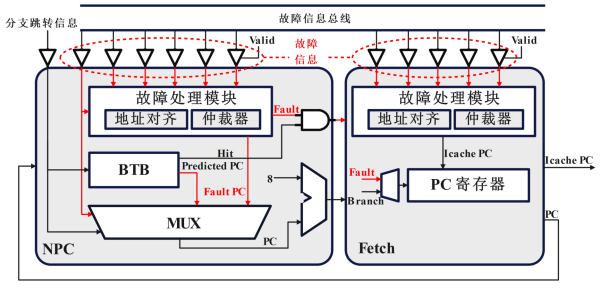

图 3　流水线级回滚在前端的实现

NPC 模块，位于 PC 阶段，负责生成下一个 PC 并进行分支预测。如图 3 所示，NPC 模块包含一个 BTB，BTB 输出两个信号：分支预测“hit”标志和预测的“predicted PC”。冲刷流水线的信号以如下两种方式产生：（1）BTB 输出缺失标志，直接冲刷流水线；（2）由故障标志“fault”和 BTB 输出的

“hit”通过与运算产生，而不是以影响分支预测历史的方式在 BTB 中产生。同时，电路还需要为“fault”信号提供比“branch”和“hit”更高的优先级来执行强制的分支跳转和对流水线的刷新。

fetch 阶段的 fetch 模块负责将当前 PC 值输出到指令缓存中。因此，FIB 下发的“fault”信号和故障 PC 将直接发送给 fetch 模块的故障处理子模块。故障处理子模块中的仲裁器将对故障 PC 进行进一步处理，来自检查器的故障 PC 具有比分支跳转 PC 更高的优先级，靠近处理器后端的故障 PC 具有比靠近前端的故障 PC 更高的优先级，来源于通道 0 的故障 PC 具有比通道 1 的故障 PC 更高的优先级。由于超标量处理器的特性，故障 PC 会被自动处理成字节对齐的 PC。因此，处理器在每次遇到故障时，都可以快速更新前两个阶段的状态。

2.3 后端故障纠正电路设计

后端故障纠正电路需要确保故障不会传播到寄存器、存储器和外设。不同阶段的故障处理方式不同。为了不显著增加组合逻辑电路中关键路径的长度，处理器前端并没有配置检查器，在处理器流水线前端出现的故障将不会在对应的阶段被处理。在前端发生的故障将会经由流水线寄存器进入 issue 级流水线，并输入 issue 级流水线的检查器中。因此，处理器前端发生的故障将会在 issue 级被处理，等效于处理 issue 级发生的故障。

图 4 是 issue 阶段的故障纠正过程。“pipe0_pc”和“pipe1_pc”是这对并行指令对应的 PC 值，也可以看作是指令本身，主核和从核分别同时向检查器输出“data_m”和“data_s”。在 T0 时钟周期，issue 阶段执行的指令为 pc，此时“data_m”与“data_s”不一致，检查器判定出现系统故障。在同一个周期内，故障标志信号“fault”置 1，PC 和 fetch 阶段分别被强制分支跳转到故障 pc+8 和 pc。需要注意的是，为了便于理解，图 4 简化了处理器流水线的工作方式，实际上并非每组指令都可以并行运行，且在 PC 和 fetch 阶段执行的指令在握手信号为 0 时会进入等待队列而不是停留在该阶段流水线中。

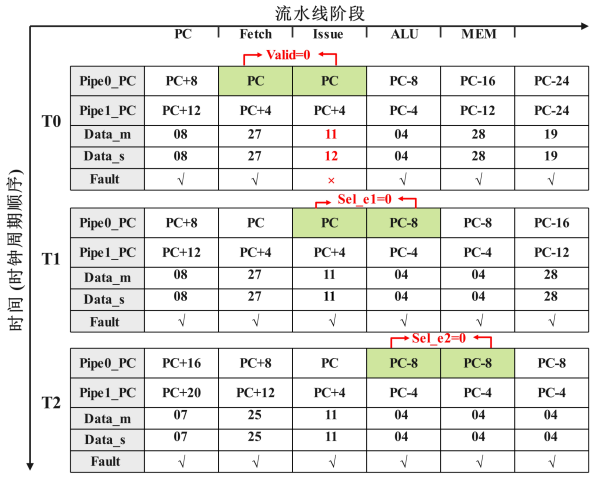

图 4 处理器后端故障纠正流程

如图 4 所示，为了避免故障传播，握手信号“valid”、“sel_e1”和“sel_e2”依次作用于流水线中相邻的两级，以控制信号的传输。在发生故障的 T0 周期，系统将握手信号“valid”置 0，分支跳转到故障 *pc*。在 T1 周期，握手信号生效，issue 阶段重复执行故障 pc。假设故障已成功纠正（data_m = data_s），“sel_e1”作用于 issue 和 ALU 阶段，使其执行与前一个周期相同的指令。在 T2 周期，“sel_e2”信号作用于 ALU 和 MEM 阶段，并产生上述效果。同时，纠正后的数据从 issue 阶段传输至 ALU 阶段，处理器恢复到正确的状态。

2.4 后端虚拟写回机制

在 ALU、MEM 或 WB 阶段纠正故障的过程更复杂。虽然对于访存指令，“sel_e1”和“sel_e2”信号能在不污染内存的情况下帮助处理器将访存数据和地址锁存在 MMU（内存管理单元）模块中，但握手信号并不能阻止数据被写回到寄存器，这可能会导致数据冲突。此外，这类数据冲突是引入流水线级回滚的超标量 DCLS 处理器特有的。

如图 5 左侧实例所示，在 T0 周期，ALU 阶段发生故障，由于存储字（store word，sw）跳转链

接（jump and link，jal）指令是并行执行的，所以会向 ra 寄存器写入 0x88+4 覆盖原来存储在的 ra 寄存器的数据。同时，由于处理器执行流水线级别的回滚，issue 阶段正在执行 sw 指令，该指令读取的 ra 寄存器数据正是 jal 指令写回寄存器的数据，所以最后写入内存的数据是错误的 0x88+4，这是超标量 DCLS 处理器特有的由数据冲突衍生出的故障。

传统的解决方式是消耗存储资源保存处理器的寄存器现场，而本文引入的虚拟写回（VW）机制将数据输入只读寄存器以防止错误衍生，并消除了保存现场的需要。该机制基于 RISC-V ISA 的规定，即寄存器文件中地址为 0 的寄存器为只读寄存器，且始终存 0。如图 5 右侧所示，在出现故障时，VW 机制直接将处理器两个通道的地址和数据设置为 0。当执行到 WB 阶段时，处理器将 0 写回到地址为 0 的寄存器中，所以再次执行指令时寄存器现场并不会发生改变。通过上述设计，issue 阶段将能读取未被覆盖的寄存器值，该方法等效于图 5 中标红的 3 条指令。

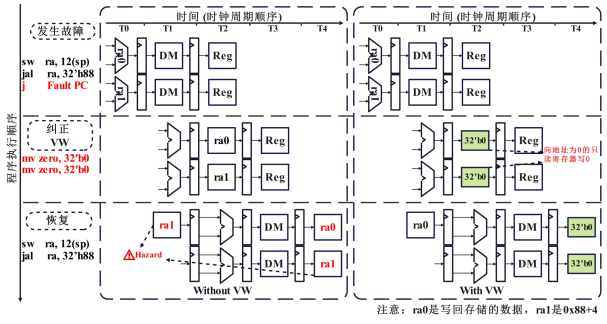

图 5　ALU 阶段数据冲突实例和 VW 机制

MEM 和 WB 阶段的故障纠正也使用 VW 机制来防止故障衍生。不同之处在于，处理器需要对应地扩展 VW 机制的引入范围，从发生故障的流水线阶段到 ALU 阶段，将两个通道的寄存器地址和数据置为 0。基于 VW 机制，处理器在纠错时不需要中断流水线或保存寄存器现场。

故障处理也并非千篇一律都采用流水线级别的回滚，不采用流水线级别的回滚来进行故障处理有两种情形。第 1 类情形是某些特殊的信号出现故障，此时双核锁步架构也无法确认究竟该回滚至哪一条 PC 值，例如 PC 信号本身或某些关键控制信号出现错误，此时 DCLS 处理器直接执行系统复位，程序回滚至第一条指令重新执行。第 2 类情形则是信号连续出现故障的时间过长，当故障在回滚处理后依旧连续发生的周期数达到事先配置的阈值，则视为系统出现难以处理的故障，即可视为永久性故障，此时也需要进行系统复位。

第 1 类故障情形的处理实现起来较简单，只需令跳转的指令为程序的起始指令，并根据故障发生的流水线的位置来确定优先级。

第 2 类故障情形的处理需要搭建一个全局的故障计数器模块，其输入为每一级流水线的故障标志信号，输出为一个拥有最高优先级的故障标志信号，该故障标志信号将作为系统复位信号促使处理器复位。全局故障计数器的工作方式是：对各类故障持续的周期数进行计数，只要达到 3 个周期，就发射一个故障标志信号。

3　试验分析

本文采用 RISC-V ISA 测试平台验证所提出的处理器是否能正常运行，并基于 Xilinx Vivado 验证用于故障检测和纠正的锁步处理器的功能操作。

为了测试流水线级回滚在超标量处理器中的可行性，本文对处理器系统进行了随机性的单粒子故障注入，其中包括向流水线数据寄存器、地址寄存器以及控制信号寄存器中注入故障。研究采用 Coremark 程序作为测试例程，并采用随机的故障注入方式来模拟不同的故障情况。具体而言，注入故障的时机和注入的故障类型均是随机的，故障持续的时间也是随机的。测试过程中，着重观察双核锁步处理器是否能检测和处理每一次注入的故障，并查看程序是否能正常运行到“exit”代码段，即测试通过的标志。

本文分别采用软件注入和硬件注入的方式进行了系统测试，验证系统功能方案的可行性。软件注

入方面，采用脚本随机生成大量的“force”和“release”语句，对处理器内部的信号进行较彻底的覆盖性检测，通过行为级仿真观察波形，验证功能的正确性，并进一步得到故障处理的时间开销。硬件注入方面，如图 6 所示，在每一级流水线均配置故障注入模块。故障注入模块主要通过一组异或门实现对特定信号特定比特的翻转，同时通过一组选择器根据输入的使能信号判定是否需要注入故障。本文基于 Zedboard FPGA 开发平台进行电路综合与实现，采用 UART 接口下载例程，将 PC 相关信号与注入故障的信号接入逻辑分析仪（Integrated Logic Analyzer，ILA），通过观察 ILA 的输出波形进行功能验证。

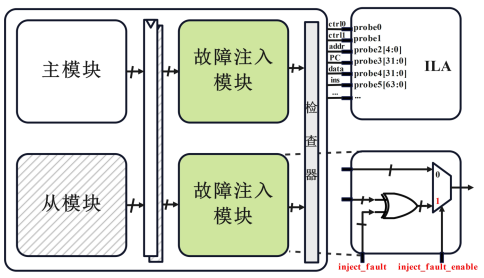

图 6 硬件层面注入故障

表 1 展示了每种信号类型注入和成功处理的故障数量。n_1/n_2 表示注入了 n_2 个故障并且能处理其中 n_1 个故障。统计分析表明，所提出的处理器能检测并纠正所有注入的故障，并最终能运行到“exit”代码段。

使用逻辑分析仪对从每级流水线阶段的故障中恢复所需的时间开销进行分析，收集的统计数据如图 7 所示，由于不需要使用保存现场的方法防止故障衍生，本文提出的架构可以显著节省故障纠正过程中保存现场的时间开销，该架构进行故障处理的时间开销低于其他引入流水线级回滚的设计。

如表 2 时序分析所示，引入本文提出的 DCLS 机制后，最差负时序裕量（Worst Negative Slack，WNS）降低了 0.259 ns，最差保持时序裕量（Worst Hold Slack，WHS）升高了 0.082 ns。该结果表明，将故障检测模块配置于后四级流水线确实不可避免

表 1 功能测试结果

流水线阶段	系统异常		数据或地址故障	
	控制信号	PC	流水线寄存器	
			通用	CSR
PC	26/26	32/32	1 440/1 440	
Fetch	26/26	32/32	1 440/1 440	
Issue 通道 0	4/4	32/32	2 880/2 880	1 440/1 440
Issue 通道 1	4/4	32/32	2 880/2 880	
ALU 通道 0	26/26	32/32	1 440/1 440	1 440/1 440
ALU 通道 1	26/26	32/32	1 440/1 440	
MEM 通道 0	12/12	32/32	1 440/1 440	1 440/1 440
MEM 通道 1	12/12	32/32	1 440/1 440	
WB 通道 0	8/8	32/32	1 440/1 440	1 440/1 440
WB 通道 1	8/8	32/32	1 440/1 440	

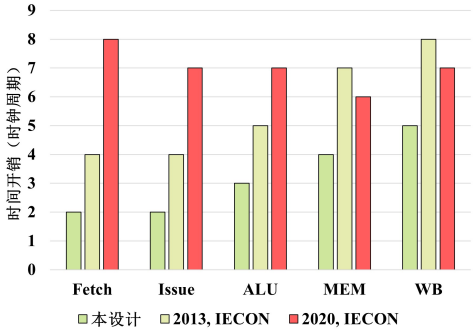

图 7 故障恢复时间开销对比

地会对电路时序造成影响。然而，该影响不会导致时序恶化，系统中不存在时序违例的情况。因此，在不向处理器注入故障的情况下，本处理器架构能在几乎不牺牲处理器原型性能的情况下引入 DCLS 机制。本处理器架构与处理器原型跑分相同（Coremark 得分为 4.1 CoreMark/MHz，Dhrystone 得分为 1.9 DMIPS/MHz）。

本文使用 Zedboard FPGA 开发平台对处理器原型和提出的双核锁步处理器分别进行电路综合。表 2 展示了本处理器架构与其他设计所提出的 DCLS 处理器的面积开销对比，本文所提出的 DCLS 处理器架构的面积开销显著减少。由于不需要使用保存现场的方法防止故障衍生，该架构可以显著节省用于保存现场的寄存器资源的面积开销。如表 3 所示，本设计的 LUT 和寄存器的面积开销分别为原型的

219% 和 227%，远低于其他使用流水线级回滚的设计。其中，虽然文献［16］中使用的处理器原型为较简单的 8 bit 处理器，本文提出的方案搭建锁步功能块所需额外的寄存器资源开销仍低于该设计。此外，该方案的面积开销甚至优于某些使用指令级回滚的处理器。

表 2　时序分析数据

参数类别	最差裕度/ns		总违例时长/ns	
处理器类别	DCLS 处理器	处理器原型	DCLS 处理器	处理器原型
建立时间	13.777	14.036	0	0
保持时间	0.142	0.060	0	0
脉冲宽度	9.600	9.600	0	0

表 3　面积开销测试结果对比

参数	本文提出的方案		2013，IECON		2012，TC	2017，LACAS	
回滚机制	流水线级		流水线级		流水线级	指令级	
架构	超标量		标量		标量	超标量	
Logic	LUTs	寄存器	LUTs	寄存器	N/A	LUTs	寄存器
处理器原型	1 132	636	71	219	N/A	1 138	1 405
DCLS	2 477	1 444	196	706	N/A	3 130	3 425
锁步电路	213	172	54	268	N/A	854	615
DCLS/原型	219%	227%	276%	322%	297%	275%	243%

4　结论

本文提出了一种面向车载功能安全的低开销超标量 DCLS 架构。在所提出的架构中，流水线级回滚被引入到超标量处理器中，支持在高性能处理器中以细粒度的方式检测和纠正故障。回滚以分支跳转指令的形式实现，加速了故障检测和纠正过程。通过 VW 机制，该架构能在不保存寄存器现场和中断流水线的情况下完成故障恢复过程。试验结果表明，该架构在处理故障方面表现良好，并且时间和面积开销显著降低。其中，故障处理时间开销相较于其他基于流水线级回滚的设计至少节省两个时钟周期，LUT 和寄存器的面积开销分别为原型的 219% 和 227%，远低于其他使用流水线级回滚的设计。

参考文献（References）

［1］ Arm. Arm Safety Ready IP for Functional Safety 2022［Z/OL］.［2023-05-05］.https://developer.arm.com/documentation/102852/0200/? lang=en.

［2］ NORMAND E. Single Event Upset at Ground Level［J］. IEEE Transactions on Nuclear Science, 1996, 43(6): 2742-2750.

［3］ LAPRIE J C. Dependable Computing and Fault-Tolerance: Concepts and Terminology［C］//Twenty-Fifth International Symposium on Fault-Tolerant Computing, 1995, "Highlights from Twenty-Five Years", June 27-30, 1995, Pasadena, CA, USA.

［4］ AVIZIENIS A, LAPRIE J C, RANDELL B. Fundamental Concepts of Computer System Dependability［C］//Workshop on Robot Dependability: Technological Challenge of Dependable Robots in Human Environments. Citeseer, 2001: 1-16.

［5］ BAUMANN R C. Radiation-Induced Soft Errors in Advanced Semiconductor Technologies［J］. IEEE Transactions on Device & Materials Reliability, 2005, 5(3): 305-316.

［6］ GAO Zhiwei, CECATI C, DING S X. A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part I: Fault Diagnosis with Model-Based and Signal-Based Approaches［J］. IEEE Transactions on Industrial Electronics, 2015, 62(6): 3757-3767.

［7］ PEÑA-FERNÁNDEZ M, SERRANO-CASES A, LINDOSO A, et al. Dual-Core Lockstep Enhanced with Redundant Multithread Support and Control-Flow Error Detection［J］. Microelectronics Reliability, 2019, 100/101: 113447.1-113447.5.

［8］ DUBROVA E. Fault-Tolerant Design［M］. Germany: Springer Publishing Company, 2013.

[9] SIDDIQUI K S, BAIG M A. FRAM Based TMR (Triple Modular Redundancy) for Fault Tolerance Implementation [C]//2011 International Conference on Information and Communication Technologies, July 23-24, 2011, Karachi, Pakistan. Piscqataway NJ: IEEE, c2011: 1-5.

[10] NELSON V P. Fault-Tolerant Computing: Fundamental Concepts[J]. Computer, 1990, 23(7): 19-25.

[11] DA CRUZ M I. A Loosely-Coupled Arm and RISC-V Locksteping Technology[D]. Braga: University of Minho, 2020.

[12] ABATE F, STERPONE L, VIOLANTE M. A New Mitigation Approach for Soft Errors in Embedded Processors [J]. IEEE Transactions on Nuclear Science, 2008, 55(4): 2063-2069.

[13] HERNANDEZ C, ABELLA J. Timely Error Detection for Effective Recovery in Light-Lockstep Automotive Systems [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2015, 34(11): 1718-1729.

[14] DÖRFLINGER A, K KLEINBECK B, ALBERS M, et al. A Framework for Fault Tolerance in RISC-V [C]//2022 IEEE International Conferrence on Dependable, Autonomic and Secure Computing, International Conferrence on Pervasive Intelligence and Computing, International Conferrence on Cloud and Big Data Computing, International Conferrence on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Sept. 12-15, 2022, Falerna, Italy. Piscataway NJ: IEEE, c2022: 1-8.

[15] DE OLIVEIRA Á B, TAMBARA L A, KASTENSMIDT F L. Applying Lockstep in Dual-Core ARM Cortex-A9 to Mitigate Radiation-Induced Soft Errors [C]//2017 IEEE 8th Latin American Symposium on Circuits & Systems (LASCAS), Feb. 20-23, 2017, Bariloche, Argentina. Piscataway NJ: IEEE, c2017: 1-4.

[16] GOMEZ-CORNEJO J, ZULOAGA A, KRETZSCHMAR U, et al. Fast Context Reloading Lockstep Approach for SEUs Mitigation in a FPGA Soft Core Processor [C]// IECON 2013—39th Annual Conference of the IEEE Industrial Electronics Society, Nov. 10-13, 2013 Vienna, Austria. Piscataway NJ: IEEE, c2013: 2261-2266.

[17] PHAM H M, PILLEMENT S, PIESTRAK S J. Low-Overhead Fault-Tolerance Technique for a Dynamically Reconfigurable Softcore Processor[J]. IEEE Transactions on Computers, 2012, 62(6): 1179-1192.

[18] SIM M T, ZHUANG Y. A Dual Lockstep Processor System-on-a-Chip for Fast Error Recovery in Safety-Critical Applications [C]//IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, Oct. 18-21, 2020, Singapore. Piscataway NJ: IEEE, c2020: 2231-2238.

[19] Ultraembedded. "Ultraembedded/BIRISCV: 32-bit superscalar RISC-V CPU" 2021[Z/OL]. [2023-05-05]. https://github.com/ultraembedded/biriscv.

[20] OZER E, VENU B, ITURBE X, et al. Error Correlation Prediction in Lockstep Processors for Safety-Critical Systems [C]//2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), Oct. 20-24, 2018, Fukuoka, Japan. Piscataway NJ: IEEE, c2018: 737-748.

作者简介

张承译（2000-），男，广东深圳人，学士，主要研究方向为集成电路设计和计算机体系结构。
Tel：0756-3668562
E-mail：zhangss6@mail2.sysu.edu.cn

通信作者

王明羽（1989-），男，四川成都人，博士，副教授，主要研究方向为高性能处理器、智能芯片和嵌入式系统。
Tel：0756-3668562
E-mail：wangmingyu@mail.sysu.edu.cn