

# 一种 1750A 指令集仿真软核设计与验证

李士刚, 祝周荣

(海装驻上海地区第六军事代表室, 上海 201109)

**摘要:** MIL-STD-1750A 指令集是星载弹载计算机常用指令集之一, 为实现该类指令集 CPU+FPGA 的通用性验证, 实现安全性、强度、单粒子翻转等异常测试, 满足测试覆盖率要求, 保证星载弹载计算机系统可靠性, 提出了一种 CPU+FPGA 的仿真模型搭建方法, 利用如中断和故障处理机制的实现、浮点运算单元设计方式、异常注入机制设计以及图形控制界面等关键技术, 实现了一种精简 1750A 仿真软核; 实验证明, 利用该仿真软核设计的 CPU+FPGA 的仿真模型平台, 可极大提高 1750 系列 CPU 相关接口的 FPGA 产品的验证效率和可靠性, 也为后续星载弹载软件的测试提供了一套故障注入方便、故障定位清晰的测试平台。

**关键词:** CPU; FPU; FPGA; 仿真软核; 激励; 故障注入

## Design and Verification of a 1750A Instruction set Simulation Soft Core

Li Shigang, Zhu Zhourong

(Shanghai Aerospace Electronic Technology Institute, Shanghai 201109, China)

**Abstract:** Mil-std-1750A instruction set is one of the common instruction sets of satellite-borne missile computer. In order to realize the universality verification of CPU+FPGA of this kind of instruction set, realize the abnormal testing of safety, strength and single particle flip, meet the requirements of test coverage and ensure the reliability of satellite-borne and missile-borne computer system, a simulation model construction method of CPU+FPGA is proposed, and a simplified 1750A simulation soft core is realized by using key technologies such as the implementation of interrupt and fault processing mechanism, floating point operation unit design, anomaly injection mechanism design and graphical control interface. Experiments show that the simulation model platform of CPU+FPGA designed by using the simulation soft core can greatly improve the verification efficiency and reliability of FPGA products related to 1750 series CPU interfaces, and also provide a test platform for convenient fault injection and clear fault location for subsequent tests of satellite-borne and missile-borne software.

**Keywords:** CPU (central processing units); FPU (float point unit); FPGA (field programmable gata array); the simulation soft core; testbench; fault injection

## 0 引言

国产 P1750A 的研制成功并大量投入使用, 规避了 1750 芯片完全依赖进口的风险, 使得 MIL-STD-1750A 指令集已经成为我国星载弹载计算机常用指令集之一, 它提供多种数据类型, 包括 16 位、32 位整数和 32 位、48 位浮点数, 以及直接寻址、间接寻址、立即寻址、短变址、长变址寻址等十三种寻址方式, 指令类型有 130 种。

通常星载弹载计算机的结构是 CPU+FPGA 结合的方式来控制各个存储模块和 IO 模块<sup>[1]</sup>, 如看门狗电路、存储器控制功能 (包括 EDAC)、总线接口芯片控制电路等。CPU 软件和 FPGA 产品都有程序的概念, 都要进行专门的验证和测试, 现有的测试方式通常是将 CPU 软件和 FPGA 产品分开测试, CPU 软件在硬件平台上进行测试, 而 FPGA 产品只能通过仿真器对 CPU 时序的读写 IO 时序和读写 SRAM 时序进行单一模拟, 但是实际 CPU 时序会根据各种指令集产生不同时序和不同时序组合, 因此现有测试方法

仅能满足简单测试用例的实施, 缺少通用性, 时序产生的随意性和主观性较强, 不能真正反映实际 CPU 时序的运行情况, 从而导致 FPGA 内部设计隐患无法发现。因此需要有一种测试方法, 既能实现实际意义上 CPU 时序的组合, 又能进行某些安全性、强度、单粒子翻转等异常测试激励故障注入, 从而实现软硬件联合仿真, 并完全满足测试覆盖率要求。

本文提出一种 CPU+FPGA 的仿真模型搭建方法, 设计了一种精简 1750 仿真软核<sup>[2]</sup>, 实现了基于 1750A 指令集架构微处理器仿真软核和 FPGA 软硬件联合仿真验证方法。该精简 1750 仿真软核能在执行星载弹载 CPU 程序的过程中, 恰当产生该微处理器各引脚输出信号、并根据该微处理器各引脚输入信号 (如中断等) 激发程序正确执行相应的响应处理程序, 实现真正意义上的软硬件联合仿真。

## 1 通用仿真模型搭建框架

该通用仿真模型参照硬件实际情况进行构建, 同时还

收稿日期: 2022-02-23; 修回日期: 2022-03-25。

作者简介: 李士刚 (1981-), 男, 上海人, 硕士, 工程师, 主要从事武器系统火力控制方向的研究。

祝周荣 (1977-), 女, 上海人, 硕士, 研究员, 主要从事 FPGA 设计与验证方向的研究。

引用格式: 李士刚, 祝周荣. 一种 1750A 指令集仿真软核设计与验证[J]. 计算机测量与控制, 2022, 30(5): 262-267.

应建立 PROM 的仿真验证模型和 CAN 总线控制器的仿真验证模型, 就像搭建一台单机一样, 将 FPGA 和 CPU 仿真软核的各个信号按照硬件接口图或者原理图的连接方式进行连接, 实现数字化的仿真验证平台, 如图 1 所示。模拟整个硬件环境, 主要验证整个软硬件系统包括 FPGA 产品和 CPU 软件的实现功能, 相当于<sup>[3]</sup>对装有正式 CPU 软件的单机进行测试, 从而保证整个系统功能的正确性。验证人员需要将 CPU 软件转换为二进制执行代码, 载入到 PROM 中运行。

测试内容不仅包括通常的应用功能, 如接口模块的控制和应用层协议帧的验证。更重要的是还可以进行空间寄存器单粒子翻转、空间状态机单粒子翻转、中断信号异常等在硬件上无法模拟的故障安全性验证等测试。此外, 由于仿真模型可以记录下内总线的所有操作记录, 因此可以对操作记录进行分析, 并将该操作记录和 PROM 中的指令地址中的指令数据进行比对, 从而可以得到目标码覆盖率结果。

利用该通用仿真模型进行软硬件协同仿真验证不仅可以进行 RTL 级仿真验证 (前仿), 还可以进行动态时序仿真验证 (后仿), 将不同温度和电压工况下的 FPGA 的电路延时放入仿真中进行执行运算, 与真实系统更为接近, 对各接口的时序测试更为精准。

如图 1 所示, 基于 1750A 指令集软核的 FPGA 第三方验证通用激励模型包括起主动控制作用的 MIL-STD-1750A 指令集架构微处理器软核、提供系统时钟和复位电路等仿真输入模块、用于仲裁总线控制的 CAN 总线控制器仿真模块、系统正常信号监控仿真测试模块、存储数据的 SRAM 存储仿真模块和用于抗单粒子翻转 EDAC 仿真测试模块、中断滤波仿真测试模块、用于调试的 RS422 串口协议仿真测试模块和用于存放二进制码星载程序的 PROM 存储仿真。显而易见, 精简 1750 仿真软核在该仿真激励模型中起了至关重要的作用, 其功能包括: 能够依据 MIL-STD-1750A 指令集架构规范完成指令功能, 能够执行 P1750A 典型应用系统的工作流程, 能够进行多种典型的故障注入和相应调试, 能够进行内部寄存器和外部存储器的任意单粒子翻转场景设置, 能够通过用户界面程序进行的配置和调试操作。

## 2 仿真软核内部体系结构

该仿真软核参照 P1750A 芯片<sup>[4]</sup>, 使用 32 位内总线结构, 由 3 部分总线构成, 如图 2 所示, 连接一个 24 bitx24 bit 的乘法器 (Multiplier)、一个 32 位的运算器 (ALU)、一个 32 位移位网络 (shift network)、一个多端口寄存器文件控制器 (Registerfile)、一个状态记录模块 (Flags) 和一个地址计算单元 (Address generator), 此外, 还内置了一个时序生成模块 (Sequencer)、一个外围 IO 设备控制模块 (IO control) 和一个存放微指令代码的 ROM (Microcode ROM)。其中, Sequencer 内部主要是一个状态机, 由<sup>[5]</sup>系

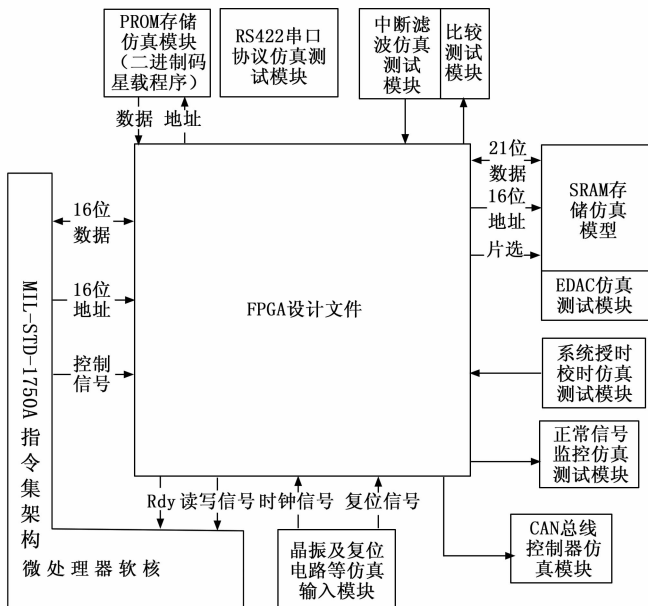


图 1 基于 1750A 指令集软核的 FPGA 第三方验证通用激励模型

统晶振控制时钟产生处理器时序和控制信号。Microcode ROM 是一个可存放 1 408 个 64 bit 字的小 ROM, 可用于存储初始程序、中断响应、总线响应、指令预取和自测试程序等。

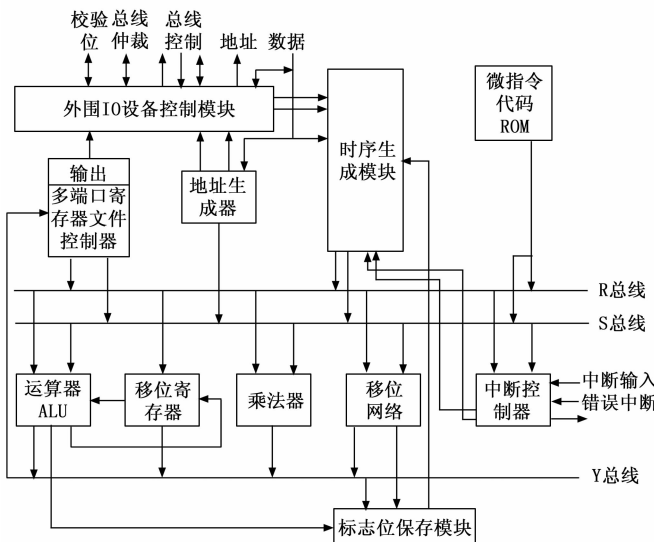


图 2 精简 1750A 仿真软核的内部体系结构

## 3 关键技术及设计方法

### 3.1 中断处理设计

<sup>[6]</sup>中断处理的完整流程如图 3 所示, 其过程概括起来可分为锁存、检测、跳转和返回四大步骤。以下将逐一阐述各个步骤的具体流程。

#### 1) 锁存:

在不同的时间节点将各个信号在锁存至中断悬挂寄存器或故障寄存器, 具体实现如表 1 所述。

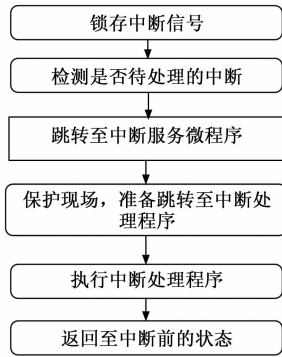


图 3 中断处理过程

表 1 锁存实现

中断悬挂寄存器	电平触发的中断位在每个时钟上升沿锁存, 边沿触发的中断立刻锁存, 其中: INTXXN: 电平或边沿触发; IOIXN: 仅电平触发。
故障寄存器	每个机器周期结束时 (AS 下降沿) 锁存故障信号, 而 SYSFN 和 FLT7N 则可以立刻 (信号有效时间需超过最低要求; 非边沿触发) 锁存。

2) 检测:

检测中断的时间节点主要为每条指令执行结束后、下一条指令开始执行前。此时检测优先级判断逻辑给出的中断请求信号, 若为有效则自动调转至中断处理服务微程序。由于 1750A 标准要求 move 等长指令在各个子周期期间允许被打断, 所以对此条指令须单独处理。不仅要在指令过程中检测中断的发生, 还要在跳转时将返回地址修改至当前 move 指令的地址。

3) 跳转:

跳转进行了表 2 两种情况处理方式设计。

表 2 跳转方式

微程序跳转地址生成	CPU 检测到中断后, 会停止指令的执行, 跳转至中断处理服务程序。其程序的入口地址由硬件给出, 直接跳转。
中断处理服务程序	中断处理服务程序由微指令构成, 位于控存 ROM 中。这些微指令用于保存 CPU 当前状态, 加载中断处理程序所需的状态, 并装载流水线。

4) 返回:

中断处理程序通常由 LST 指令返回。该指令会自动装载跳转前的处理器配置和状态, 将 IC 指向跳转前的地址, 并自动装载指令流水线。根据 1750A 标准的要求, LST 执行失败会导致中断不能返回, 进而继续执行下一条指令。CPU 的默认行为与此相符, 不需要特殊处理。

3.2 浮点运算单元

1750A 指令集共定义了两种格式的浮点数: 普通浮点数 (32 位双字) 和扩展精度浮点数 (48 位三字), 以及 6 种浮点数的操作类型, 包括加减乘除以及和定点数的相互转换<sup>[7]</sup>。浮点运算单元依据 FPU 指令集中规定的运算方法

对浮点数进行计算和处理, 图 4 描述了 FPU 运算逻辑的工作原理<sup>[8]</sup>, 送入 FPU 的操作数首先经过预规格化后根据不同的操作类型传送给对应的运算单元, 由于加减法和乘除法对操作数的规格化要求不一致, 故采用了独立的规格化模块。运算结果经规格化和舍入后输出。浮点数的比较运算与定点数不同, 无需经过实际的减操作, 而直接由逻辑引脚给出比较结果。若操作数不合法或不符合指定的运算规则<sup>[8]</sup>, FPU 模块会产生异常。这些异常大多由异常模块给出, 而除数为零的异常由除法运算器给出。

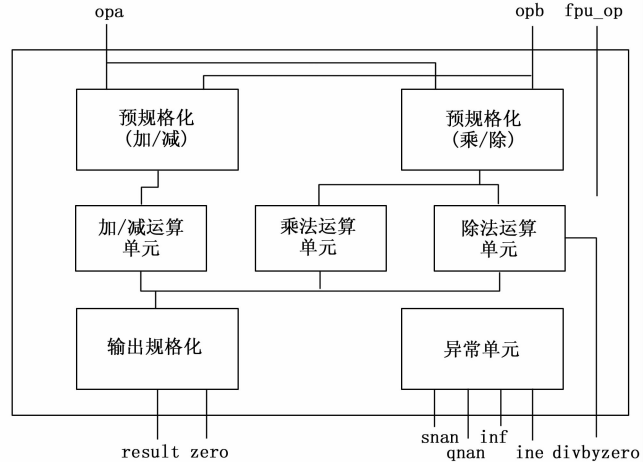


图 4 浮点运算单元模块工作原理示意图

图 4 所述的浮点运算单元模块包括加减法、乘法和除法运算功能<sup>[9]</sup>, 且各自独立, 不用的运算单元能够关闭通道, 从而提高仿真验证效率。

3.3 故障注入机制设计

通过调研当前各型号的常见错误故障, 实现如表 3 述类型的故障注入, 进一步完善软核的功能。

表 3 故障注入实现类型

总线超时故障注入	此类故障用于模拟主存或 IO 设备在总线周期中不送出有效的 RDYN 信号而导致的总线超时故障。
存储器保护错误注入	此类故障用于模拟处理器对受保护的内存数据页进行写操作, 或是对受保护的内存指令页进行读操作所产生的错误。
外部奇偶校验错误注入	此类故障用于模拟外部的奇偶校验错误的发生。可以支持的触发条件为: 指定的指令码、指定的指令周期。
数据奇偶校验错误注入	此类故障用于模拟由于存储器数据出错导致的奇偶校验失败的情况。可以在指定主存单元中注入不满足奇偶校验的数据。此类故障与外部奇偶校验错误注入 <sup>[9]</sup> 不同点在于, 并不是通过 PEN 引脚注入奇偶校验错误, 而是通过在数据总线上传送不满足奇偶校验的数据而触发该错误。
外部地址错误注入	此类故障用于模拟处理器外部地址错误, 可以支持的触发条件为: 指定的指令码、指定的指令周期。满足触发条件时, EXADEN 引脚会置为有效状态。
用户定义错误注入	此类故障用于模拟用户自定义错误 FLT7N 的注入, 可以支持的触发条件为: 指定的指令码、指定的指令周期。满足触发条件时, 相应的引脚会置为有效状态。

续表

系统错误注入	此类故障用于模拟系统错误 SYSFN 的注入, 可以支持的触发条件为: 指定的指令码、指定的指令周期。满足触发条件时, 相应的引脚会置为有效状态。
非法指令注入	此类故障用于模拟非法指令注入。此功能通过修改内存代码页的内容实现。
特权指令错误注入	此类故障用于模拟产生特权指令错误。在处理器状态 PS 非零情况下, 执行 XIO、VIO 等特权指令时, 会引起错误寄存器第 10 位置 1。此功能通过在用户代码中插入特权指令来完成。
未实现的地址状态错误注入	此类故障用于模拟未实现的地址状态错误的产生, 该错误的注入仅支持具有特权指令执行权限的代码段中。
Trigger-GO 超时注入	此类故障用于触发 Trigger-GO 的超时动作, 可以支持的触发条件为: 指定的时钟周期、指定的指令码。满足触发条件时, 相应计数器模块将产生超时信号。

在进行故障注入的时刻点, 先读取要注入故障的寄存器和信号线当前结果, 然后通过写 force 强制语句强制寄存器和信号线为相反的结果, 然后过 1 微秒后, 释放强制<sup>[10]</sup>, 从而实现故障注入。

### 3.4 图形控制界面

为方便系统和用户之间进行交互和信息交换<sup>[11]</sup>, 以图形的方式提供一套软核控制机制, 提供软核初始状态设置、故障注入、内存管理、调试断点设置、单步执行等功能, 主要分为 3 个部分: 顶层配置、内存管理和故障注入, 如表 4 所述。

表 4 图形控制界面设计部分

顶层配置界面	该部分用于配置软核运行环境的顶层。页面正中较大的区域为引线编辑区, 每个拖入该区域的元件都将实例化成一个模块。用户在编辑区中定义引脚连线。
内存管理界面	该部分实现将二进制文件载入仿真环境主存储器的功能。
故障注入界面	该部分对需要注入的故障进行设置, 并根据设置值产生故障注入配置文件。故障注入以指令为单位, 界面给出当前主存程序页的反汇编结果。用户通过反汇编代码列表指定故障注入位置, 并选择需要的故障类型进行注入。

## 4 仿真实验和结果分析

### 4.1 仿真模型构建

项目组挑选了某卫星型号项目计算机 CPU 板 FPGA 和遥测采集 FPGA 进行仿真验证<sup>[12]</sup>, 目前主流的仿真验证方法一般分为 2 步: 1) 先是通过仿真器对 CPU 时序的读写 IO 时序和读写 SRAM 时序进行单一模拟先对计算机 CPU 板 FPGA 进行仿真验证, 无法运行真正的 CPU 程序, 而且故障注入方式单一, 不能真正反映实际 CPU 时序的运行情况, 不能体现实际外部 RAM 的运行情况, 因此无法发现计算机 CPU 板 FPGA 内部设计较深层次隐患; 2) 观察计算机 CPU 板 FPGA 输出时序, 由于 CPU 时序的读写 IO 时序

和读写 SRAM 时序本身就是模拟的, 具有不确定性, 因此对遥测采集 FPGA 的输入激励就存在错误的可能性, 从而造成遥测采集 FPGA 测试的不确定。此外, 这种主流仿真方法只能进行 FPGA 产品的语句、分支和状态机覆盖率<sup>[13]</sup>, 对更为关注条件覆盖率和甚至是翻转覆盖率的测试无法达到测试要求。

利用该 1750 仿真软核搭建仿真激励模型将计算机 CPU 板 FPGA 和遥测采集 FPGA 联合起来能够很好的解决上述 2 个问题, 既能实现实际意义上 CPU 时序组合的精准输出, 还能联合构建 2 个 FPGA 同时进行仿真, 从而实现系统级软硬件联合仿真<sup>[14]</sup>, 此外还能进行 RAM 运行情况测试、单粒子翻转等异常测试激励故障注入, 实现正确的计算机 CPU 板 FPGA 和遥测采集 FPGA 仿真波形, 最终满足各种代码测试覆盖率要求。激励模型顶层连线情况如图 5 所示。

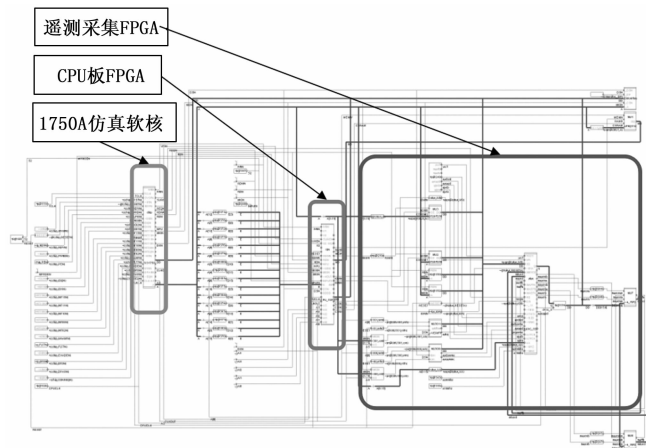


图 5 1750A、CPU 板 FPGA 和遥测采集 FPGA 连线

如图 5 所示, 该系统联合仿真模型主要由 1750A 仿真软核、CPU 板 FPGA 和遥测采集 FPGA 三大模块组成。其中 1750A 仿真软核运行星载程序产生总线时序给 CPU 板 FPGA。CPU 板 FPGA 作为 1750A 仿真软核和遥测采集 FPGA 的桥梁, 进行总线时序转换, 此外, 还完成图 1 中通用激励模型的功能。遥测采集 FPGA 实现该系统实际和外部硬件接口的遥控遥测指令控制。

激励模型构建完成后, 执行了一段 RAM 翻转检查的程序。该段程序通过 my\_printf() 函数调用 XIO 命令向 0x0100 端口写 ASCII 码, prints() 函数用于构造错误信息字符串, 程序主函数调用 my\_printf() 输出一个内容为“RAM TEST”的字符串, 标志着程序开始执行, 然后将 FIELD\_BEGIN (0xD000) 和 FIELD\_END (0xD05C) 之间的一段 RAM 区域初始化为某个数值 (0xF5A9), 再将该区域内的值读出并与期望值比较, 若发现不同, 则调用 my\_printf() 打印类似“<<Err 0x \*\*\*\* >>”字样的出错信息, 接着, 程序将该段被测区域的数值读后再写回, 以便利用 FPGA 的数据校验功能修正可能发生的单粒子翻转问题。最后, 该程序尝试向 0x8030 这个只读区域写入内容并检查是否写入。



### 4.5 覆盖率统计

图 13 为仿真验证执行覆盖率总表<sup>[17]</sup>, 大部分功能模块和接口的验证覆盖率都达到了 80% 及以上, 总覆盖率超过 85%。其中 Biu 功能模块的覆盖率较低, 原因是代码中有一部分采用了宏定义, 而 VCS 仿真无法正确识别这些宏定义, 因此在覆盖率总结中将其作为未覆盖到的部分。

SCORE	LINE	COND	BRANCH	
86.12	86.73	85.43	86.21	bench
70.76	78.95	66.67	66.67	BusArbiter
100.00	100.00		100.00	IOMonitor
86.03	86.66	85.36	86.08	TJ31750
86.03	86.66	85.36	86.08	chip31750
85.79	86.66	84.94	85.78	core
71.60	83.82	63.11	67.88	biu
100.00	100.00			entry_memory
90.54	90.18	93.06	88.37	engine
98.53	100.00	95.60	100.00	ALU
98.52	100.00	96.64	98.91	FPU
100.00		100.00	100.00	ICOR
100.00	100.00	100.00	100.00	LBusR0
100.00	100.00	100.00	100.00	LBusR1
100.00	100.00	100.00	100.00	LBusR2
100.00	100.00	100.00	100.00	LBusS2
100.00	100.00	100.00	100.00	LBusVVO12
100.00	100.00	100.00	100.00	LOPF_BR
100.00	100.00	100.00	100.00	LOPF_C
100.00	100.00	100.00	100.00	LOPF_CCODE
100.00	100.00	100.00	100.00	LOPF_CexPEG
100.00	100.00	100.00	100.00	LOPF_CREG
100.00	100.00	100.00	100.00	LOPF_D16
100.00	100.00	100.00	100.00	LOPF_DSPL
100.00	100.00	100.00	100.00	LOPF_ERA
100.00	100.00	100.00	100.00	LOPF_RA
100.00	100.00	100.00	100.00	LOPF_RB
100.00	100.00	100.00	100.00	LOPF_RX
100.00	100.00	100.00	100.00	LOPF_SIMM
100.00	100.00	100.00	100.00	LOPF_SpecRB
97.11	96.47	100.00	94.87	VIOEntry
98.84	100.00		97.67	XIOEntry
75.49	77.02	80.95	68.51	ifu
100.00	100.00	100.00	100.00	rfile
86.00	86.00			uop_memory
100.00		100.00	100.00	pad
100.00	100.00	100.00	100.00	io
100.00	100.00	100.00	100.00	mem

图 13 仿真验证执行覆盖率统计图

### 4.6 效果比对

从测试结果可以看出, 通过使用该 1750A 软核, 可以明显提高 FPGA 代码的条件覆盖率和翻转覆盖率<sup>[18]</sup>。在该测试中, 执行了 FIFO 读写, 编/解码等测试, FPGA 的翻转覆盖率由 41% 提高至 87%。在测试效率提升方面, 原为达到 FPGA 代码 91% 的测试覆盖率需人工操作 4 个工作日, 利用该软核后, 可在七个小时内完成, 测试效率大大提高。

### 5 结束语

本文提出了一种 CPU + FPGA 的仿真模型搭建方法<sup>[19]</sup>。首先从通用仿真模型搭建框架入手, 介绍了精简 1750 仿真软核的作用和地位, 接着在建立了精简 1750 仿真软核内部体系结构的基础上, 对中断处理机制的实现、浮点运算单元设计方式、故障注入机制设计以及图形控制界面的实现等关键技术进行了阐述, 最后搭建了较为完整的 1750 系列 CPU + FPGA 的仿真模型平台并在项目中应用。

实验证明, 利用该仿真模型平台, 可极大提高 1750 系列 CPU 相关接口的 FPGA 产品的验证效率和可靠性<sup>[20]</sup>。

### 参考文献:

- [1] 张剑贤, 吴文华, 刘锦辉, 等. 基于 FPGA 的 8 位 SoC 微体系结构设计 [J]. 计算机教育, 2020 (3): 31-35.
- [2] 谢 辉. 一种基于 FPGA 的高精度浮点运算处理器系统科技广场 [J]. 2008 (7): 183-185.
- [3] 杨 毅, 郭 立, 史鸿声. 可编程顶点处理器 FPGA 验证平台的设计与实现 [J]. 中国科学技术大学学报, 2009 (11): 1130-1135.
- [4] 吴君钦; 李艳丽; 一种 SOC 软硬件协同验证方法的设计 [J]. 江西理工大学学报, 2011, 32 (3): 58-65.
- [5] 丰玉田, 付宇卓, 赵 峰. 大规模 SoC 设计中的高效 FPGA 验证技术的研究与实现 [J]. 电子技术应用, 2006 (2): 111-114.
- [6] Tom Fitzpatrick; Verification HORIZONS [Z]. Mentor 公司内部手册, 2008.
- [7] P1750A 处理器数据通路设计 [D]. 西安: 西北工业大学, 2001.
- [8] 刘 凌. P1750A 浮点执行部件的设计与实现 [D]. 西安: 西北工业大学, 2001.
- [9] 张玲玲, 李克俭, 蔡启仲. 基于 FPGA 单指令浮点乘法自主控制器设计 [J]. 计算机测量与控制, 2014, 22 (10): 3323-3326.
- [10] 刘国斌, 祝周荣, 宁 静, 等. 基于 VCS 的固存坏块仿真系统设计与应用 [J]. 计算机测量与控制, 2018, 26 (4): 264-267.
- [11] 陈 锐, 门永平, 杨文强, 等. 基于 UVM 的 FPGA 软硬件联合仿真验证技术研究 [J]. 空间电子技术, 2017 (2): 38-42.
- [12] 杨松芳, 张 维, 张 勇. 针对 SoC 的软硬件联合仿真方法 [J]. 无线电工程, 2013, 43 (2): 61-63.
- [13] 路 辉, 董昕玉. 基于 CPU + FPGA 的临近空间信道模拟器 [J]. 计算机测量与控制, 2021, 29 (9): 161-169.
- [14] 张晓宇, 孙 俏. 分布式电池管理系统 CAN 总线通信仿真试验研究 [J]. 电子测量技术, 2019, 42 (9): 55-58.
- [15] 孙文莉, 王海涛, 蔡 磊. 基于 FPGA 与 DSP 的发动机参数采集系统设计 [J]. 计算机测量与控制, 2019, 27 (6): 186-189.
- [16] 祝周荣, 关俊强, 李前进, 等. SV DPI 技术在 FPGA 仿真的应用探讨 [J]. 计算机测量与控制, 2018, 26 (6): 264-267.
- [17] 周永明, 刘云秋, 许进亮, 等. 基于 DSP 和 FPGA 的信息化发射平台多协议网关设计 [J]. 计算机测量与控制, 2019, 27 (3): 137-140.
- [18] 谢 峥, 王 腾, 雍姗姗, 等. 一种基于 UVM 面向 RISC CPU 的可重用功能验证平台 [J]. 北京大学学报: 自然科学版, 2014, 50 (2): 221-227.
- [19] 何铭俊, 陆文高, 曾 鸿, 等. 一种卫星任务解译闭环仿真验证系统的设计与实现 [J]. 计算机测量与控制, 2019, 27 (1): 271-274.
- [20] 周 珊, 杨雅雯, 王金波. 航天高可靠 FPGA 测试技术研究 [J]. 计算机技术与发展, 2017, 27 (3): 1-2.