

仿真扩展芯片简化高速汽车嵌入式处理器的集成

由于微控制器时钟速度的加快，调试和校准变得更加地困难。嵌入式仿真设备能够为汽车电子设计工程师提供高度集成的系统级芯片 (SoC) 微控制器内部的可视性。

自动代码生成工具能够通过基于图示模型之上的设计生成 C 代码。对该工具越来越多的使用可以节省软件工程投入的时间，但同时也给软件工程文化带来巨大的变化。

软件工程师将现有的模块组合在一起，建立一个系统，并对该系统作出检测。与把定制模块写入一个系统中以满足系统要求的方法相比，该方法的效率更高。对该通用的模块化软件的再利用提高了开发效率和软件质量。但同样也导致结构松散，存储器效率低下，长时间的延迟及将设计过程从硬件中抽象出来。

在不改变基本软件的情况下，这些系统的性能能够转变成不同版本的机械硬件。一般情况下，这是通过将许多校准了的变量包括在内而实现的，因此才能够在运行期间使能或禁止功能，调整增益，同时在运行时改变查找表。

软件工程师依靠仿真技术来跟踪真实系统中的程序流程，关注随时更新的数据，测量延迟，并且排除逻辑错误。但是，在为这些系统除错时，由于时钟速度过高，给多内核存储器系统级芯片的集成带来了一些挑战。

设计障碍

嵌入式非易失存储器的体积持续增大，硅片的几何形状持续缩小，使得规模大的系统级芯片微控制器能够与应用广泛的高速内部总线连接在一起，并通过这些总线将信息反馈给多个具有高速缓冲存储器的流水线型内核存储处理器及协调处理器。该体系同样能够使微控制器子系统能够被集成到更深层次的应用环境。由于存在物理连接问题(有时甚至连一根外部总线都没有)，以及时钟速度过高、电缆长度及环境温度等因素，将分析设备集成到深度嵌入式器件的外部总线上的难度很大。

在许多情况下，可通过外部总线察看的外部取数情况并不代表整个程序的流程，因为内部高速缓冲存储器及流水线能够对取数做出预测。突发模式闪存同样由于假定的连续地址增量使得所获得的数据的解码变得更加复杂。

连接仿真系统时的首要问题在于连接长度受到限制，因为系统时钟的速度过高。例如，如果使用一个频率为 150MHz 的微控制器，那么，50cm 长的连接线造成的传输时间延迟大约为 2.0ns，但是时钟周期仅为 6.67ns，所以 2.0ns 的单向延误时间是相当巨大的。由于在这些高频器件中，连接线扮演着传输线的角色，并且它们的终接阻抗是无法得到保障的，因此，这看似短暂的时间几乎妨碍了任何远离目标器件的控制功能。在这个例子中，想要忽略传输线造成的时间上的延

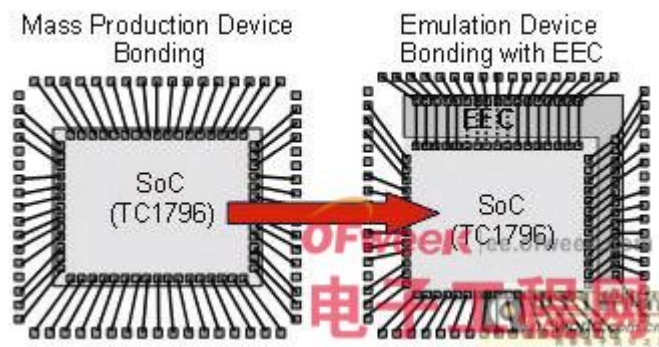
迟，线路的最大长度不能超过 16cm。所以，在线仿真器(ICE)具有与被测发动机电控单(ECU)相同的环境需求。

Ingeon 公司的 TC1796 产品即为此类系统级芯片实现的一个很好的例子。32 位的 TriCore 中央处理器(CPU)拥有代码和数据分别独立的总线,并且通过 LFI 与系统总线桥接起来,建立起一个与外设子系统相连的数据通道。此外,还有一个通过直接存储器存取(DMA)与远端外设总线相连的通道。

外设控制处理器(PCP2)也是一个 32 位的 CPU。同样的,该处理器也有独立的通常无法看见的数据和程序总线。该处理器的子系统的最高频率为 150MHz, 外设子系统最高频率为 75MHz, 因此,它有两个时钟区域。该器件被封装于一个 416 引脚的球型格栅极阵列封装中,能够提供标准的 JTAG 除错接口以支持除错功能。但是,要完全仿真这样一个微控制器,需要能够检验与外部针脚不相连的许多不同内部总线之间的数据交换。巨大的嵌入式存储器(2Mb 闪存)有广泛的内部取数路径(128 位)和局部高速缓冲存储器。因此,内置存储器的执行比外置(32 位存取)存储器快得多(见下方 TC1796 方框图)。

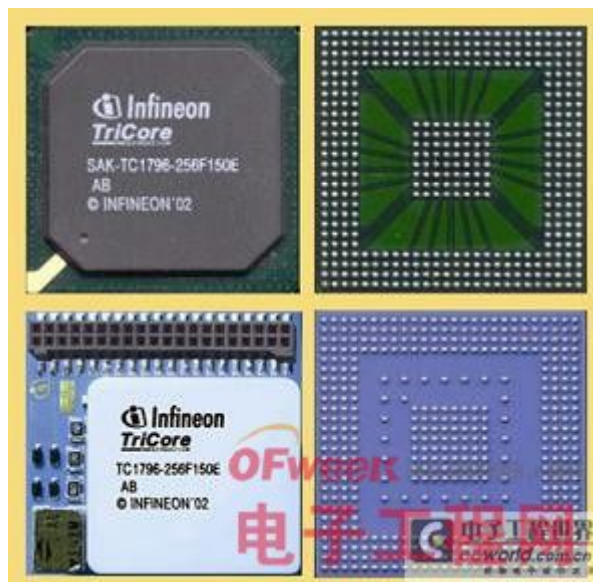


多总线及多内核存储器的复杂性意味着,只有一个外绑器件(现场可编程门阵列, FAPG)才能够达到完全除去系统错误所需的可视度水平。但是,如前面所讨论的,频率为 150MHz 的总线周期仅为 6.67ns,一个外置的外绑控制器在这段时间内要接收总线信息,并对其解码,还要决定需要触发的中断及暂停处理器,因此,给予控制器的时间是远远不够的。在这种情况下,解决问题的方法就是将仿真器安装于外绑器件中,形成所谓的仿真设备(见下方对比图:左方为大规模生产器件,右方为拥有仿真扩展芯片的仿真设备)。



仿真设备使用一个原始生产器件的宏指令，完成所有的一般性功能、外设及端口，接着在外缘添加一个 512K 的静态随机存取存储器 (SRAM)、几个总线观察模块 (BOB) 及一个拥有一些本地存储器来控制仿真器的局部 CPU (在此情况下为另一 PCP2)。若干高速串口，包括 USB、JTAG 和 Micro-Link 端口提供了外部连接口。该附加的线路，也叫 EEC (仿真扩展芯片)，在大规模生产器件发生改变时，使得仿真器很容易被重新设计，因为互连点并没有发生变化。

传统的外绑器件的另一常见问题在于封装的大小。仿真设备被特别设计成与标准的生产器件占位面积相符合，并且拥有一组额外添加的针脚来提供更多的信号。同时，还给封装直接添加了一个顶端连接，通过可移动的连接器就可以采集相同的信号，下方图象分别显示了大规模生产器件封装 (顶端) 及仿真设备封装 (底端)。



仿真设备甚至提供了比传统的 ICE 更多的性能，这些性能包括：

跟踪性能：

TriCore 程序、数据及状况跟踪；

PCP 程序、数据及线路跟踪；

所有控制多个组件的主线全方位的可视性；

缓冲器跟踪数据的优化压缩；

所有的跟踪在时间上都全面地得到校正；

中央时标单元；

事件前后跟踪缓冲(逻辑分析仪)

触发逻辑性能：

触发器能够用于触发中断、跟踪限制及开始/停止跟踪；

用于控制指令指针及数据地址的范围比较器；

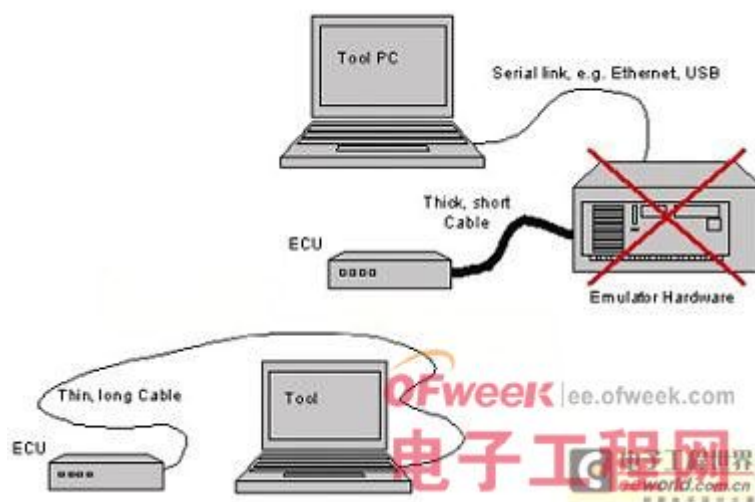
用于控制数据掩模对等/范围比较器；

附加的外部事件输入(2)及输出(4)；

用于计算事件、并在计算和时间基础上进行时间测量及事件生成的计数器；

同时地、有选择性地开始和停止所有内核存储器的中央机制；

仿真器同时也提供了特殊的重启模式，通过各种重启进行调试和校准，开机重启除外(见下图)。



仿真设备将仿真器功能嵌入目标应用的生产器件封装中，因此，外置仿真器硬件不再为人们所需。

EEC 中 512KbSRAM 是由许多能够配置的零碎存储空间组合而成的，这些零碎的存储空间能够被任意指派不同的任务。这使单一器件能够使用于不同的应用情况中。这些功能包括：

—逻辑分析仪模式：SRAM 用于(与压缩规则系统同时)跟踪任一内部总线及内核存储器的程序及数据流程。

—软件开发模式：SRAM 用于存储程序代码，避免改变时产生老化的闪存，同时允许无限的基于软件之上的断点的存在。

—校正标定：SRAM 用于暂时存储所需的校正常量。当发生存取时，校正标定硬件将取数方向由内部闪存转变为 EECSRAM。外置校正工具能够通过 USB 接口或 JTAG 接口来对 SRAM 的零碎存储空间进行快速灵活的读写。

—快速制作原型：SRAM 作为外置快速原型硬件和微控制器之间的消息缓冲器。由于延迟时间要求低于 3ms，USB 接口不适于此项任务，因此可以使用 MicroLink 端口或 JTAG 接口(延迟时间约为 2us，带宽约为 3Mbps)。

—“飞行”记录仪：SRAM 用于记录系统中使用的数据，或用于跟踪发生特定错误时系统的情况。

仿真设备是由主机 PC 在运行时配置而成的，该接口受到严格的规定，因此各种软件工具能够基于同一标准进行互操作。NEXUS 并行的、基于总线的跟踪端口在此不能直接应用，因此只能重新使用软件编程接口(API)。支持仿真概念的软件提供了一个器件存取服务器(DAS)。该接口允许多个工具例程装于一台 PC 中，以共享与嵌入式主机(USB、JPAG)相连的单一通信路径。DAS 同样允许多个处理器例程与内置的嵌入式主机相连。因此，在将来，也可以访问更大规模的系统级芯片器件。DAS 也支持外在控制协议(XPC)标准，该协议使校准和记录工具能够与物理连接媒介独立地连接起来(CAN、FlexRay、USB、JTAG)。

由于时钟速度的加快，高度集成的系统级芯片微控制器的调试和校准只会变得更加的困难。使用一个嵌入式仿真设备能够为软件工程师们提供微控制器内部工作情况的可视性，确保系统软件在现实世界任何条件下都有正确而强大的功能。