

在ASIC模块中实现数字示波器

Mircea Moldovan, Dan Nicula, Traian Tulbure
eASIC Corporation

介绍

随着深亚微米型标准单元设计所需的研发费用已达到数百万美元级别，一些设计人员不可避免地开始寻求可替代的解决办法来减少花费，有的甚至以牺牲成本或性能为代价来解决。

在这种情况下，结构化ASIC就脱颖而出。结构化ASIC兼具了标准单元ASIC和高密度FPGA的优点，缓解了两者之间性能和成本难两全的矛盾。结构化ASIC可以比标准ASIC更快速投放市场和降低研发经费，同时较FPGA又有更高的性能和更低的单元花费。

结构化ASIC包括了可编程ASIC，可编程ASIC采用可定制VIA而非金属层定制。在可编程ASIC阵列中，所有金属层都是标准和预布线的，四层是用于有效的布线，只有via层由客户根据设计直接实现的。

下面的例子就叙述了用eASIC公司的可编程ASIC实现数字示波器，称为eScope。它包括一个双通道的数字采样示波器和一个单USB动力模块中的任意波形发生器。

eScope应用在130纳米的可编程ASIC上。芯片包括数字逻辑（内插数字触发逻辑的采样缓冲存储器，波形输出缓冲存储器，数据序列器和USB的I/O界面逻辑），外部逻辑电路界面以及USB收发器逻辑。单板上80MHZ的示波器驱动片上PLL的时钟发生器来给每个数字输入和输出通道以及USB的I/O通道产生单独时钟区域。

Escope通过USB与PC相连。PC GUI的作用是用来

查看和处理已获的数据。

下一节将介绍eScope的应用。

I.总揽

eScope是一种基于PC的数字采样示波器。基于PC的示波器比桌面示波器更省钱。它可以使用PC现有更大的彩色显示器，高速处理器和大型磁盘存储，而不是不得不另外购买一台示波器。

数字采样示波器使用等效时间采样的方法来捕捉和显示信号采样。采样示波器测量信号的速度比实时示波器快一个数量级。基于此，这些示波器是捕捉和定义计算机，数据通讯和电信信号的理想工具。



图.1—eScope系统

图一所示，这个eScope系统只需要eScope板和主计算机。

II. eScope架构

eScope设计使用九个子模块和OCP来内部连接一个通用共享存储器，让不同的数据访问端口。这些子模块是：

- **clkGen:** eScope时钟/初始发生器；
- **adcInput:**从ADC中同步数据采样；

- **dacOutput**: FIFO同步波形数据到DAC；
 - **trigGen**: 含有OCP初始器端口的触发生成器；
 - **waveGen**:含有OCP初始器端口的波形发生器逻辑；
 - **hostIf**:含有OCP初始器端口的USB模块主机接口；
 - **sampleMem**:用来存储数据采样和波形发生器的模块存储器；
- ocpMerge**: 3个OCP发生器端口和1个OCP目标端口的3:1组合OCP合并；
- **ocp2mem**: OCP到存储器接口转换器,包含一个OCP目标端口。

子模块互联如图2所示：

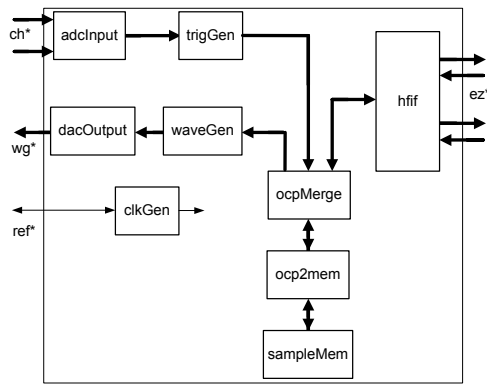


图2—eScope内部架构

III. 探测途径

探测途径使用三个模块：*adcInput*, *clkGen* 和 *trigGen*。

在芯片中，时钟发生器(*clkGen*)为每个时钟域生成时钟和清零信号。eScope在*scClk4x*执行探测，在*wgClk*生成波形，两者的速度都是210MHz。其它的片上时钟包括*ezClk* (24 MHz) 和*hfClk* (48 MHz)。

*adcInput*模块接受来自于ADC的12比特数据采样，这些ADC把数据采样插值到96比特带宽字。这些字被发送到*trigGen*模块。时钟基准也在这个模块中使用。

来自于ADC (*ch1Data_i* and *ch2Data_i*)的12比特数据通常以并发数据流的形式传输到存储器中。当eScope针对双采样方式进行配置后，信道中的采样时钟被180度偏移，来自于两个信号的数据被交织发送到存储器。基于触发和16比特 *hfFreqDiv_i*参数的采样向前传输。这个参数指定了多少数据采样能够在来自于ADC的数据流中保持。时钟基准的使用基于下列公式 (1)：

$$No_sample * T_sampled * (hfFreqDiv_i + 1) = No_divisions * Time/division \quad (1)$$

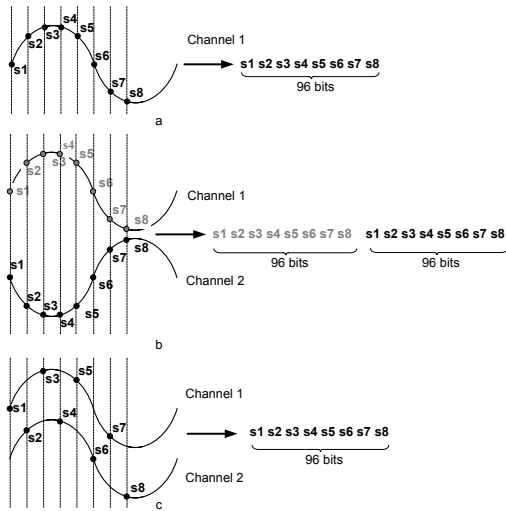


图3—数据封包为96比特字：a--信道模式，b—双信道模式，c---双采样模式

trigGen模型使用所有触发逻辑，这些触发器逻辑被探测参数控制。使用OCP内部互连信道写96比特宽字。

触发器逻辑按照图四所示的序列运行。1比特信号 $hfStartAcq_i$ 初始化触发器逻辑，该信号由 $hfif$ 一个时钟周期生成。1比特信号 $hfEndWindow_o$ 返回到 $hfif$ ，指示探测结束。在一个探测周期内， $aqPend$ 信号是活动的。在缓冲触发周期之前，在 $hfTriggerPos_i * 16$ 存储器地址 * 8个采样写入存储器之后， $trigPend$ 信号变得活跃。根据来自于 $hfif$ 的8比特参数 $hfTriggerPos_i$ ，触发器事件在存储器区域指定专门的信道。只有在 $trigPend$ 信号是活跃的时候才会产生触发器事件的搜索。当触发器事件作为 $doTrigger$ 信号变得活跃的时候才会产生这种情况。

CPU ($hfOffsetAddr_o$) 读取第一个数据地址， $hfif$ 模块计算和存储触发器事件8个采样的偏移，该偏移来自于96比特宽的字。在每个触发器事件发生后，跟着一个缓冲触发器周期，该缓冲触发器指定 $postTrigger$ 信号。当 $doTrigger$ 产生时，信号

变得活跃，当 $hfEndWindow_o$ 产生时，信号变得不活跃。所有这些信号用来控制不同操作的信号，这些操作产生于触发器逻辑模块。

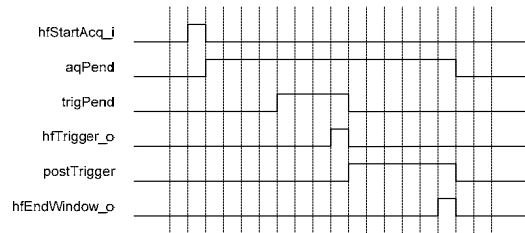
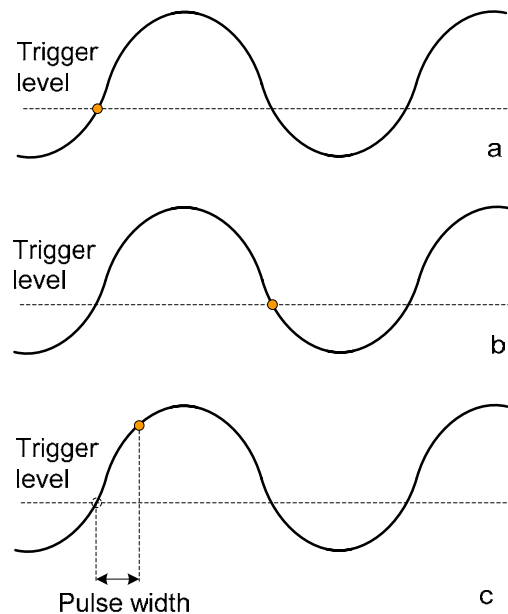


图4---触发器控制逻辑

当指定从 $hfif$ 模块中以2比特的 $hfTriggerType_i$ 输入时， $eScope$ 的检测可以用四种不同的触发器方式工作：

- 上升沿触发器；
- 下降沿触发器；
- 上升脉冲触发器；
- 下降脉冲触发器；



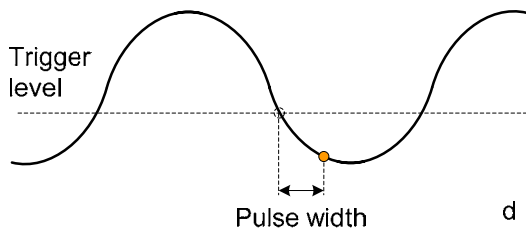


图5---触发方式：a---上升沿，b---下降沿，c----上升脉冲，d----下降脉冲

完整的采样可以打包为96比特字，并用OCP协议写入存储器。每个信道有独立的地址计数器。当一个写请求发生时，就会根据检测模式选择合适的计数器。整个存储器按照下面的方法进行管

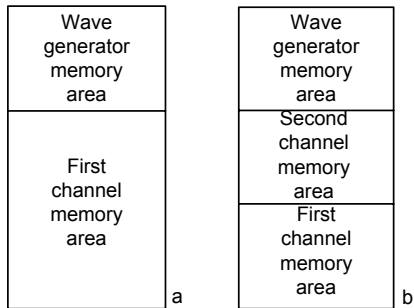


图6---存储器管理：a---信号和交织采样模式b---双信道检测模式

IV.波形发生器方式

检测和波形发生器共用主要的片上存储器。用户定义的采样可以上传到波形发生器存储器中。一旦操作完成，波形发生器输出用户选择的频率信号。

波形发生器输出方法有两种模式：*waveGen* 和 *dacOutput*。

waveGen 的角色是生成OCP读命令到存储器，传输接收到的96比特数据字到*dacOutput*模块。使用OCP协议同步到*scClk4x_i* 时钟进行传输。

dacOutput 从*waveGen* 模块接收数据，传送这些数据到DAC输出。*dacOutput* 模块使用异步FIFO存储器从*scClk4x_i* 时钟到*wgClk_i* 同步数据。这些FIFO是32位深的96比特FIFO。使用双口RAM存储器来创建FIFO的存储器，这些RAM存储器存在于整个eASIC可编程的ASIC逻辑模块中。

在连续的*wgClk_i* 时钟周期中，每96比特数据字通过 $(8 \times hfSampleWidth_i)$ 12比特接口送到DAC。每个采样在*wgClk_i* 的*hfSampleWidth_i*周期保持在DAC的输入中。针对波形发生器的输出，这种架构使用简单的分频器。分频器参数(*hfSampleWidth_i*)写入eScope控制寄存器。

嵌入式FSM序列器生成*waveGen* 存储器请求，当FIFO存储器有空间的时候就生成这些请求。

V.功能方案

通过EZ-USB驱动程序GUI来访问eScope：

- **write** 通过指定片上地址总线寄存器地址或者通过指定控制信号，写程序配置 *hfif* 寄存器。

- **read** 通过指定片上地址总线寄存器地址或者通过指定控制信号，读程序从 *hfif* 寄存器获得数据。
- **burst write** 通过指定数据总线的采样或者通过指定控制信号，触发写程序在波形发生器存储器区域存储采样。
- **burst read** 通过指定控制信号，触发读程序从 eScope 探测存储器区域获得采样。*hfif* 模块管理该地址。

使用这四个程序，可以对该应用环境进行编程，以支持探测和波形发生器的操作。功能方案如下所示：

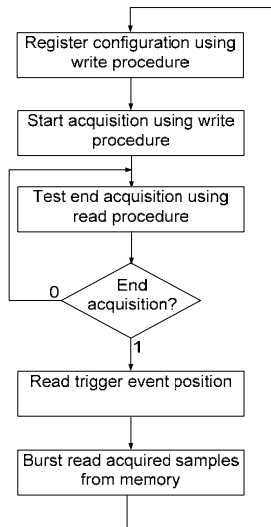


图7—探测方法功能方案

波形发生器途径探测方案：

- 使用写程序建立存储器配置；
- 触发写波形发生器采样到存储器；
- 使用写程序开始波形发生器工作；
- 运行。

VI. 仿真环境

该仿真bench包含eScope实例，该实例含有逻辑(

在 *eScopeLogic* 中用具体例证说明)，时钟信号发生器(在 *eScopeClk* 中)和 pad instances (存在于 *eScopePads*)，该bench还包含ADC和DAC实例，叫做 *ez_usb_dummy* 的EZ-USB接口和两个额外的生成参考时钟(*clk_gen*)和参考清零(*reset_gen*)信号的模块。还包含 *ezusb.h* 文件，该文件用来描述EZ-USB接口的功能性。这个文件可以对含有不同功能方案的设计进行编程。

用来仿真的ADC模块含有一个存储器，用来存储每个读取采样的信道信息。从预置文件中加载这些存储器。每个采样被乘以增益参数，以此来放大被送到eScope输入端的信号。

DAC模块从波形发生器获得数据，生成一个文件用来进行分析。

为了访问eScope，*ez_usb_dummy*模块应用了如先前描述的四个任务(程序)：写，读，触发写和触发读。

因为eScope的操作是高度可配置的，所以它不可能去仿真所有的操作情况。可以通过预置每个存储器来生成操作情况，有四种可能的值：低极限值，高极限值和两个中间值。使用随机值功能发生器生成一个15比特的数字。针对特殊的仿真，来自于这个数字的比特组与寄存器的预置值相关联。

在仿真中，使用Sine，Cosine和复杂的波形作为测试信号。

仿真分为三种：

1. 确定使用 Verilog 仿真器的触发事件是否正确测试。

2. 检查获得的数据是否与 ADC 生成的原始数据匹配的测试。该采样信号通过 USB 接口传输到一个虚拟主机，使用 MATLAB 脚本来检查是否背离原始数据。
3. 检查波形发生器输出是否匹配上传到波形发生器存储器里的原始数据，该测试也使用 MATLAB 脚本。

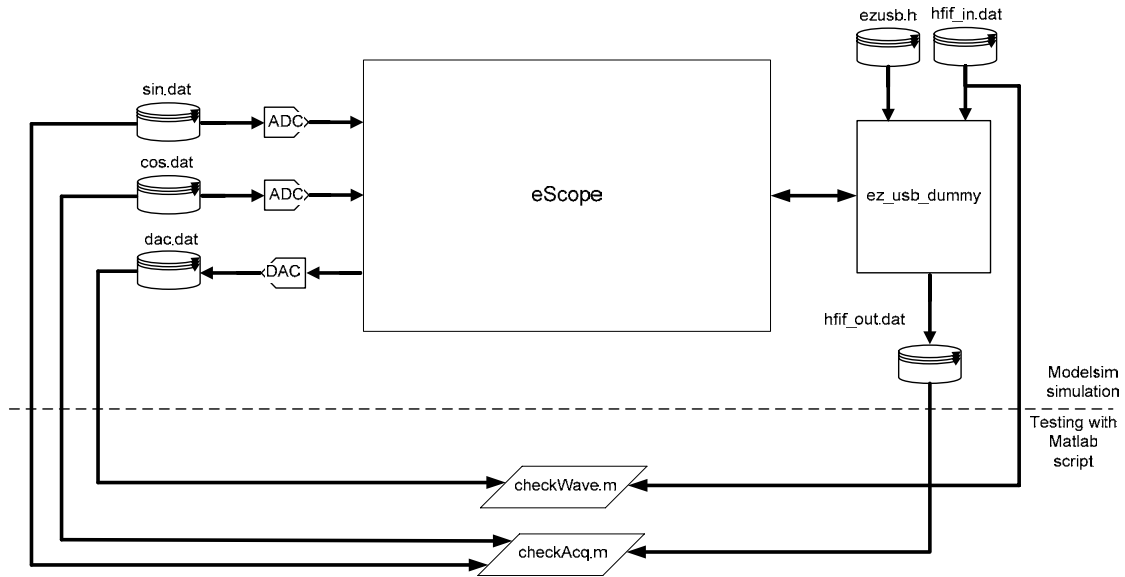


图8—eScope仿真环境

VII. eASIC可编程ASIC模块的应用

对于该应用，eTool设计流程如图9所示，它基于标准单元设计的ASIC设计工具。这个工具流程尽可能的支持现有的ASIC设计方法学和基础设施。

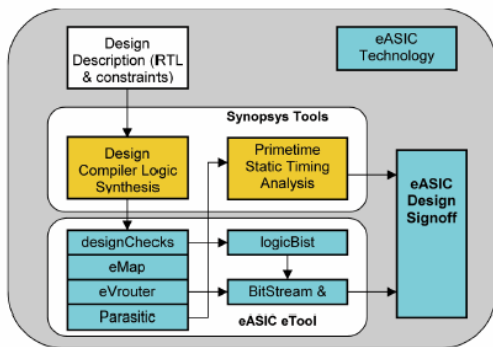


图9---eTool设计流程

使用Synopsys Design Compiler综合工具来执行逻辑综合，同时使用eASIC自由格式综合库，该库

包含全部两个、三个、四个和五个可以组合使用eCell中的LUT配置实现的输入函数，以及反转缓冲模型和大量的DesignWare组件，这些组件已经在eCell逻辑中进行优化应用。

Design Compiler生成网表，该网表通过了eASIC开发“技术映射和打包”工具(称为eMap)传递，该工具尽可能使用速度较快的nand和mux函数替换速度较慢的、基于LUT的函数，然后尽可能将多个函数打包成一个eCell。在布局布线之前，eMap创建的网表采用标准的线负载建模，与Synopsys Primitime静态时序分析配合使用。

图10使用Synopsys流程，eASIC架构的ASIC结构上eScope的应用

Clocks	Synthesis (ns)	mapping (ns)	Place & Route (ns)
--------	----------------	--------------	--------------------

system	3.95	4.72	7.53
wave generator	3	4.43	5.42
host interface	26.37	28.97	35.62

Synthesis (eCells)	Mapping (eCells)	Place & Route (eCells)
5945	3923	3923

图10---eScope时序和面积结果

后端设计流程使用eASIC的eMap和eRouter工具来实现最后的布局布线。使用Primetime针对后端（signoff）静态时序分析来执行寄生提取来注释最后的网表。同时，创建logicBIST（嵌入在BIST）测试参数，并将其嵌入到最后的位流和过孔膜（Via Mask）数据库。使用eDK工具来实现位流汇编和配置过孔膜（Via Mask）生成步骤。

VIII.图形用户接口

eScope GUI应用屏幕截图如图11所示

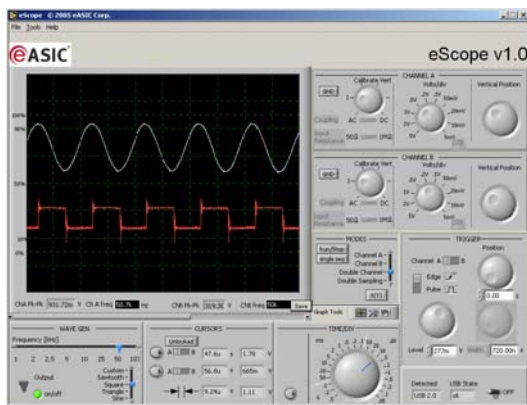


图11---ESCOPE GUI应用

图形对象包括控制和现实。控制分为布尔控制（按钮），切换按钮，水平开关和数字控制：倍

eScopeSW用来控制eScope硬件的GUI应用。它使用LABVIEW开发环境和C库作为LabVIEW“VI”进行开发。USB接口作为物理媒介，用来实现eScope硬件和PC运行的eScope软件之间的数据传输。

GUI和eScope设备之间的通信使用两层结构：

- EZ-USB驱动（仅针对Win32平台）；
- 能够访问低级驱动函数的C高级访问函数。

EZ-USB负责软件和硬件模块之间的数据传输。它提供用在C库中的低级函数。

有两种C函数：

- 使用应用在EZ-USB驱动特性的低级函数（读/写采样/寄存器值，连接状态）；
- 驱动的高级函数。当一个操作访问硬件模块结束时，LABVIEW应用请求这些函数。

值的水平和垂直滑动。

显示分为数字，文本显示，LED和图像波形显示。

和传统的桌面示波器一样，前面板的控制和显示也按照功能分组。

IX. 结论

eScope是作为参考设计进行开发的，针对高速逻辑设计，把它用在eASIC设计工具流程教程中，显示了eASIC创新的可编程ASIC模块技术先进性。此外，使用来自于eASIC公司的可编程ASIC技术，eScope证明了开发高速低成本产品的可能性。

参考

- [1] eScope design specification, eASIC Corp., www.easic.com
- [2] “OCP-based memory access arbitration for a digital sampling oscilloscope”, by Traian Tulbure and Dan Nicula, 2005
<http://www.pldesignline.com/showArticle.jhtml?articleID=174903173>
- [3] www.ocpip.org
- [4] www.oscilloscopeguide.com/pc-oscilloscope.htm
- [5] Oscilloscopes, Fifth Edition by Ian Hickman – Reed Educational and Professional Publishing Ltd 2001. ISBN 0 7506 47574
- [6] Tektronix TDS5000B Series Digital Phosphor Oscilloscopes Quick Start Manual 071-1355-02 by Tektronix Inc.

Chinese Words Account: 4,139
Translation By SSIPEX
2006-9-4