Sirius项目Veloce验证平台介绍

1. 前言

在现代集成电路设计中，集成电路工艺的突飞猛进和IP复用技术越来越成熟，使得高度集成化、系统化、大规模化已经成为集成电路一个发展趋势。在这种趋势下，芯片的功能越来越复杂，研发周期也越来越长。往往在现实中，我们遇到的问题不是如何保证芯片设计的顺利实现，而是如何在最短的时间之内验证芯片功能的正确性，并及时开发出相应的系统软件，因此如何提高大规模，超大规模芯片系统的验证效率，缩短软件开发的周期已经成为大规模芯片设计正在关注和急需解决的问题。

传统的验证技术中，主要采用两种手段：

一种是基于EDA工具(Simulator)的仿真验证。这种验证方式是基于软件平台的，优点是使用方便，便于问题定位和调试。但是当被验证的设计的规模变得很大，到达上百万门甚至上千万门的时候，仿真速度将会急剧下降，已经不再满足芯片开发时间上的需求。

另外一种是基于FPGA(Emulator)的仿真验证。这种验证方式是基于硬件的，优点是速度快，可以进行长时间测试向量的测试，还可以测试芯片的一些性能参数。但是FPGA仿真需要事先定制相应的FPGA板，并且由于FPGA的规模有限，对于超大规模SOC系统，FPGA已经无法胜任。

在传统的芯片系统开发流程中，软件总是在芯片流片回来后才完成开发或者测试的，这将极大的延迟芯片投放市场的时间，影响产品的竞争力。

Veloce 硬件加速仿真平台综合了传统的EDA仿真和FPGA仿真的优点，可以实现像EDA工具一样的debug同时也具备较高的仿真速度。其大容量和通用性成功的解决了千万甚至上亿门级的多核系统的验证问题，同时Veloce 硬件加速ICE模式可以和JTAG调试工具联合使用，实现软硬件联合仿真和并行开发。

1. Veloce加速器平台简介

Veloce硬件加速平台是Mentor Graphics公司开发的一个业界先进的高速，通用化，大容量，可视化，便于debug，多用户，多模式的仿真验证平台。

通用化大容量：Sirius项目所用Quattro系列Emulator最高可支持2亿门的设计，不需要定制，可适用于任何架构的数字芯片系统。能很好解决超大规模多核芯片系统的验证和软硬件协同仿真。

高速性:仿真速度可达到1M到1.5M，可在短时间内执行大量用例，更快的发现逻辑的错误，适合随机测试和用例回归。

快速化编译:可支持VHDL、Verilog，以及混合语言的输入，高度可靠和自动的编译器，大约1500万门每小时。

信号100%可视:设计中的所有信号100%可见。

强大的debug能力：Veloce硬件加速平台具有友好的图形界面，能够像EDA工具一样对所有信号在任何时间点进行debug。提供多种类EDA工具一样的debug方式，包括

● 断言(Assertion)

● 源文件设置断点(Break Pointing On RTL Source Line)

● 检测点保存和恢复(Check Point Save-n-Restore)

● 查看寄存器级电路(Path Browser)

● 波形比较(Waveform Comparison)

● 信号trigger(Trigger On RTL Signals)

● 观察波形(Waveform Viewer)

● 查看电路原理图(Schematic Browser)

● 单步调试(Step Debug)

多用户:可实现多用户同时进行验证仿真

多模式:提供三种模式，分别为Co-Simulation(Hdllink)、Co-Modeling(TBX)、Emulation(ICE)

在Hdllink模式下，Testbench在Workstation端的软件仿真器运行，可综合部分的Testbench及DUT(Design Under Test)，综合后加载到硬件加速器中，Workstation和Veloce之间通过Co-model卡进行通信。Hdllink模式的加速倍数在5到15倍之间。该模式支持VCS，NC-Verilog，Questasim仿真器。传统的EDA仿真环境基本不用做修改即可切换到Hdllink模式进行加速仿真。

TBX模式是基于事物级或者Transaction的交互。Hdllink中时序部分的Testbench(比如monitor，driver)，用可综合的SV语言(XRTL)改写后，和Design一起加载到了Veloce中， 而非时序部分的Testbench仍然放在Workstation中。Workstation和veloce中通过Co-model卡进行通信。TBX模式中，非时序部分的Testbench和时序部分的Testbench之间通过TBX interface通信。如果非时序部分是SV，则TBX接口是TLM Interface和XTL LIB；如果是C/C++/SC则通过transaction Interface(DPI接口)通信。TBX模式的加速倍数在几十倍到几百倍之间。该模式支持Questasim仿真器。

ICE模式和前两者最大的不同是没有Testbench。测试向量来自和Veloce相连的外设。ICE模式下Workstation将测试向量通过IOBOX发送到Veloce。ICE模式的加速倍数在几百到上千倍之间。

Veloce的三种应用模式可以应用于无线、多媒体以及嵌入式等众多领域。

1. Veloce基本流程

Veloce的三种模式虽然有所不同，但是基本流程大致一样，分别为库单元映射、编译、仿真及调试四个步骤。

* 1. 库映射

建立物理库，并把逻辑库映射到物理库，类似于FPGA。

* 1. 编译

编译分为6个步骤：

Analyzer：分析输入源文件（包括SV、Verilog、VHDL）的类型以及部分语法分析。

Rtlc：对源文件进行编译，将源文件转化为Verilog识别的门级网表，并生成相应report文件和用于debug的库。作为Rtlc的输入文件可以是SV、Verilog、或者混合语言，也可以是Verilog门级网表。

Velsyn：将Rtlc生成的门级网表分割后，生成布局布线所需要的库文件，并进行相关的时序分析和生成dump波形所需要的库文件。Velsyn开始前，需要设定时序、端口映射，相关Memory初始化设置，以及其他用于调试的report文件的设置。

Velcc：用Velsyn生成的库文件完成对各个AVB板上各个FPGA芯片的布局布线。

Velgs：进行最后的时序分析，产生Veloce仿真所需要的各个事件和资源调度的时序信息。

Ssrcc：针对SVB板上的每个FPGA dump波形做分析。

* 1. 仿真

仿真步骤大致如下，前5步是必须的步骤，后面的为可选：

Connect：选择要连接的加速器，建立目标设备和加速器之间的连接。

Configure：将编译好的设计代码加载到Veloce的FPGA中。

Targerpower ignore：将Veloce的所有的IO设置成高族态。

Enableio：将Veloce和目标设备及IOBOX的连接使能。

Run：运行仿真。

Download memory：将二进制程序加载到Memory中。

Upload memory：读取memory的值。

Upload waveform：Dump波形。

Download trigger：加载trigger文件。

其他仿真手段，可参考Veloce UG。

* 1. 调试

如果仿真中出现错误，就要进行Debug。

1. Veloce在Sirius项目中的应用

Sirius是一款具有四个ARM7，四个CEVA DSP处理器，片内高速总线，ISP图像处理器，baseband基带模块，Video Codec视频编解码模块以及各种高速、慢速外设接口的超大规模SOC芯片。

由于芯片规模巨大，传统的EDA仿真需要很长的时间完成整个芯片的验证。而现有的FPGA无法容纳一次性如此大规模的芯片系统，只能进行部分模块的联调。所以为了保质保量并且快速的完成整个芯片的验证，公司决定选择Mentor公司的Veloce2验证平台进行芯片的验证。Veloce2不仅拥有2亿门的容量和各种通用化的解决方案，而且具有比传统EDA工具快近1000倍的验证速度，符合我们的要求。

* 1. Veloce验证平台的使用

Workstation中装有Veloce调试软件。Veloce调试软件对设计源文件进行编译，综合，将综合的网表加载到Veloce中，以及download、upload memory、debug等。

* 1. 验证仿真流程

第一：将Design中Memory替换成能被Veloce综合的Memory。

第二：裁减掉Design中不可综合的部分，比如PLL，把所有的clock通过clockfile方式加载进去。去除顶层处Jtag的其他IO，外加全芯片WRAP。

第三：将RTL级别的设计源文家进行编译、综合、布局布线，生成可加载到Veloce中的FPGA芯片中的库文件。这一步综合阶段要指明Jtag仿真器的管脚和IOBOX上IO的对应连接关系（ICE模式时使用）。这一步可以在veloce.config文件中加很多选项来使这个设计更加完善，方便后续debug。

第四：编译HVL部分，这部分就是Mentor提供的虚拟外设的解决方案，主要跑在Workstation端，通过Comodel与Veloce进行信息交换。

第五：将生成的文件加载到Veloce中。

第六：通过run.tcl使Veloce开始运行。运行中有很多debug手段，比如upload/download memory、抓波形、下trigger文件等等。

第七：比对运行结果。

* 1. Sirius build 介绍

在考虑到Sirius芯片的用户实际应用场景以及Emulator AVB板子及license的合理利用率，我们将Sirius的验证分为4个build。

Build1是最小系统验证环境，主要用来给每个IP owner 作为原型加入各自的IP进行单独验证以及为软件部分开发系统driver和系统软件做准备。Build1包括了4个A7 cores、M7和M7 periphreal、DDR和TZC、NOC和CCI、DMA、Security Subsystem、USB、GMAC、Type-C、Coresight、SPI Debug等。

对于Build1，我们需要在Testbench中准备好相关Veloce Model与我们的DUT对接进行验证，包括外部存储介质DDR4/DDR3/LPDDR3，支持HBM/QBM/FBM（DDR BUS width with 16/32/64 bit）以及inline ECC；外部SPI Flash/EMMC Device/SD Card；外部USB host/device，支持2.0/3.0协议； OTP model；1000M GMAC model；PCEI model。

Build2是A7加CEVA，主要是为CEVA小组开发相关软件算法以及cache coherency 和MMU 的测试做准备。Build2一个A7、2个CEVA、DDR+TZC、NOC\_main+NOC\_CEVA+CCI、SMMU\_CEVA、DMAC。具体结构如下图：



对于Build2，我们可以在build1的基础上去掉不需要的外设，然后把CEVA连接到DUT。

Build3是一个图像处理的应用case，用来验证图像数据从外部通过高速接口接收进来，经过ISP处理，再发送出去等功能。主要包括A7、DDR+TZC、NOC\_main+NOC\_vision+CCI、SMMU\_Video、DMAC、ISP、Display Engine、MIPI RX or HDMI、Video\_interface、MIPI TX or DP。模块结构如下图：



Build3除了最小系统那些外设外（没测到的可以不用加），需要接HDMI，DP，I2S，MIPI这些Veloce multimedia的transactor。

Build4是在build3的基础上将两个Sirius芯片对接，一个Sirius将处理好的图像用vcodec编码，通过baseband发送给另一个Sirius，然后解码。模块结构如下图：



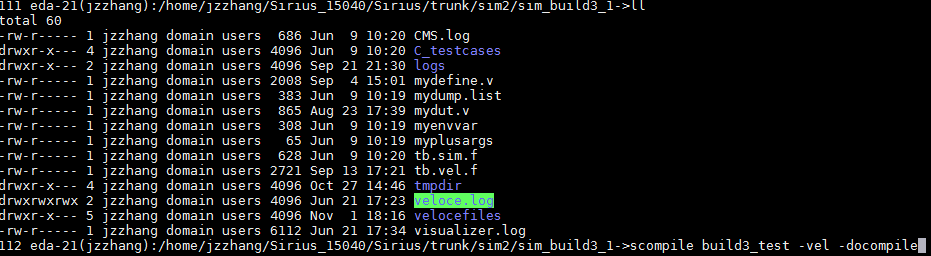
Build4主要加入了baseband的模块，然后在testbench上又例化了一个Sirius。

1. Veloce在Sirius项目使用中的一些tips

编译：

我们以Sirius\_15040版本的build3\_1为例，Source $ROOT/sim2下common/simsetp环境变量。然后cd到相应的build目录进行如下操作编译综合veloce环境。

* 1. scompile test\_case –vel –docompile



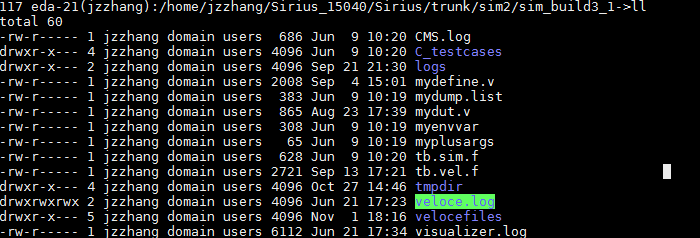
* 1. cd to tmpdir/test\_case\_VEL
  2. make build



* 1. make sim



a步骤是对hdl部分的编译，c步骤是对hvl部分的编译，主要是一些dpi和transactor的编译，d步骤是veloce的runtime。这里除了我们的DUT之外，其他veloce的filelist主要存放在$ROOT/sim2/sim\_build3\_1/tb.vel.f里，veloce编译时的相关选项在$ROOT/sim2/sim\_build3\_1/velocefile目录里，其他相关的define等变量也在同一级目录下。综合完成等到的database在tmpdir/\_db\_VEL下。

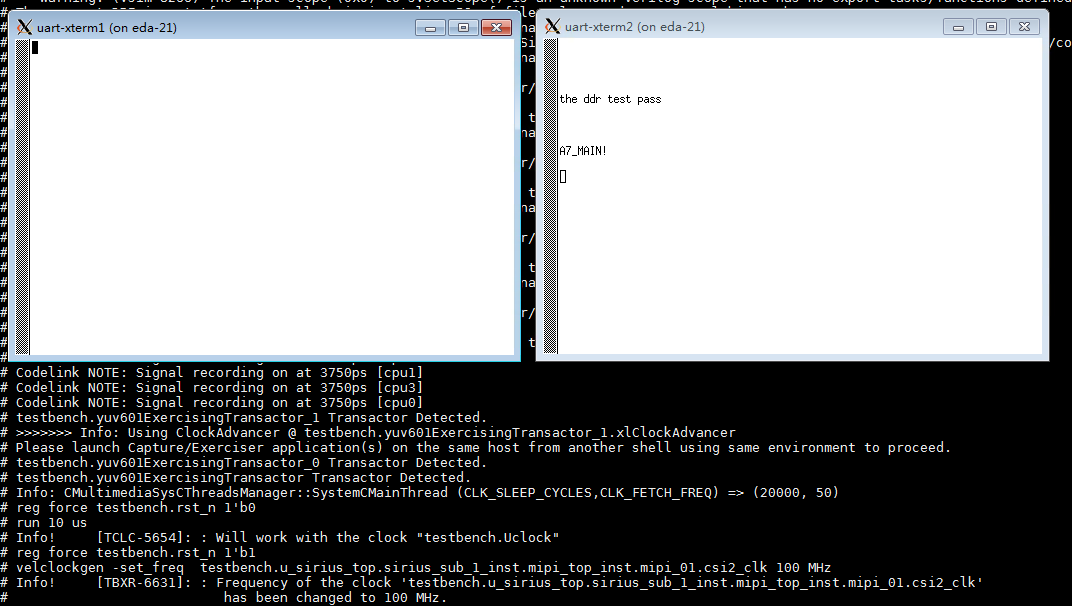


Runtime：

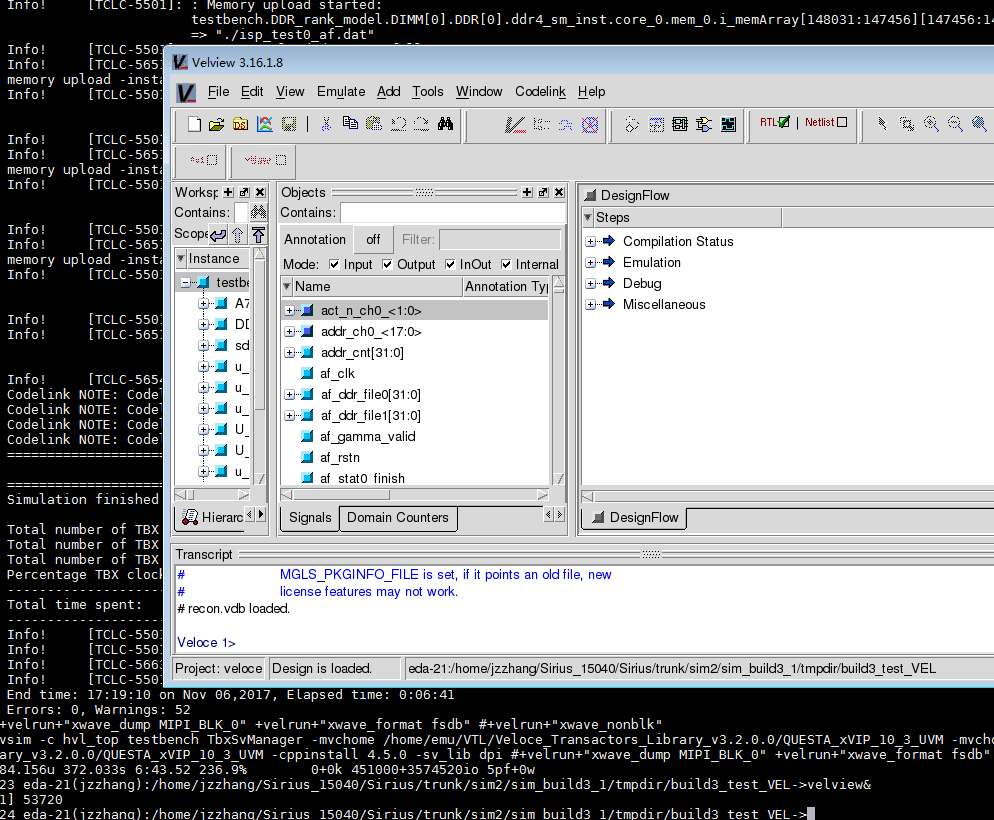
编写好run.tcl，然后make sim（如上图），连接veloce进行仿真。Runtime的时候有很多实用的命令可以帮助我们仿真，如memory upload/download，trigger download，reg force value，xwave on等，具体可以参考手册。

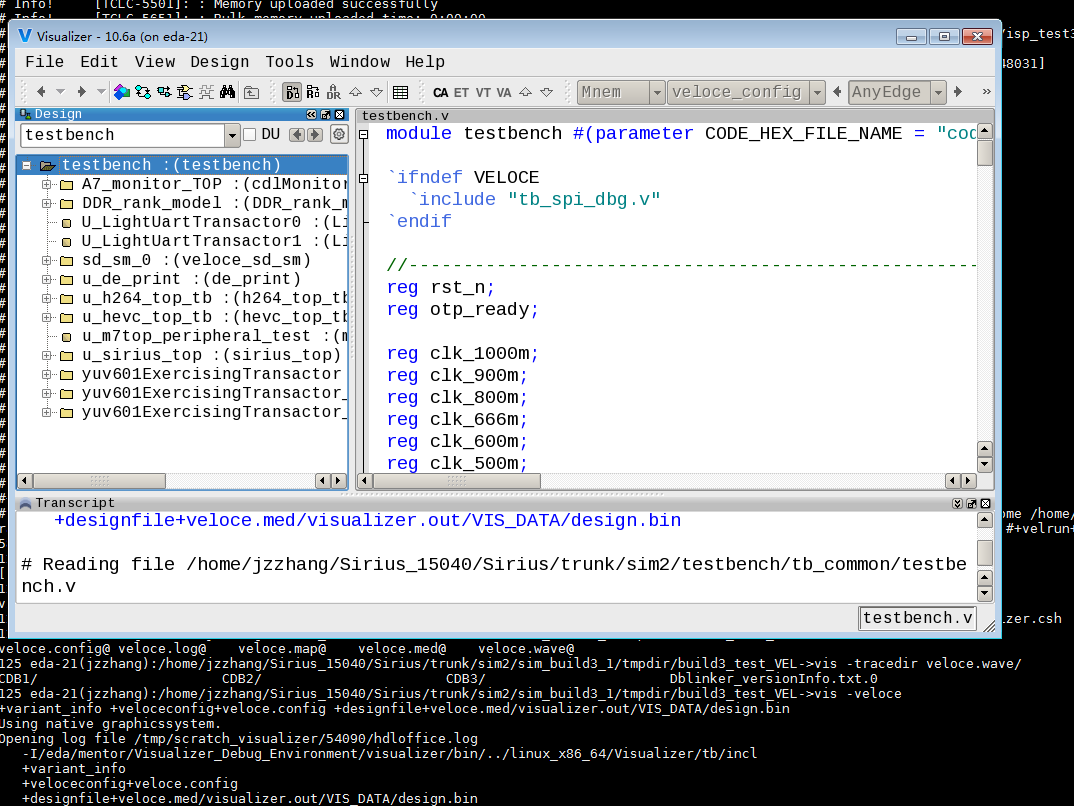
查看UART信息：

在veloce环境中，我们加入了两个UART model与我们的DUT对接，这样我们可以在我们的firmware中加入打印信息帮助我们debug。



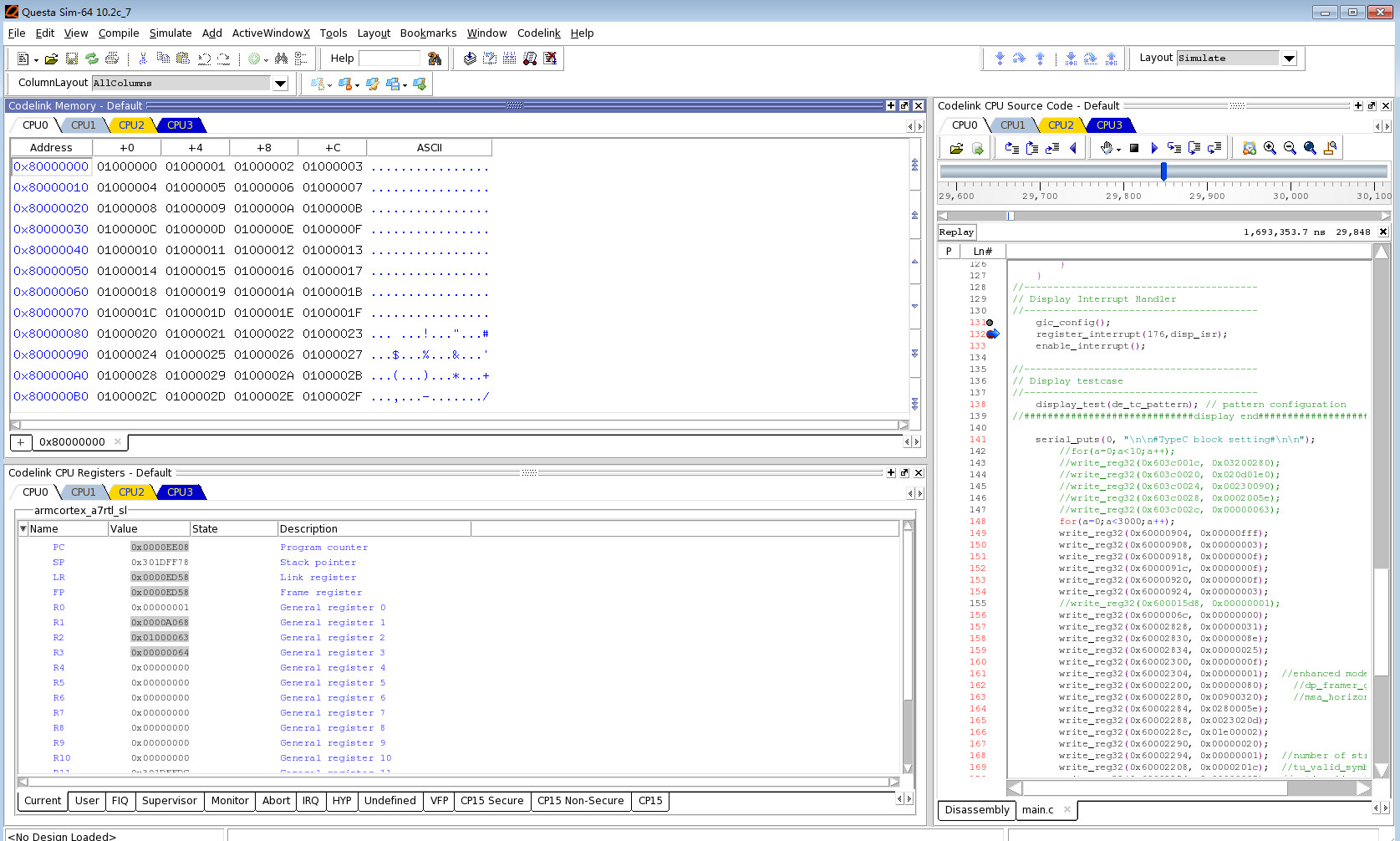
打开波形：

有两种方式打开波形，一个是velview，还有一个是visualizer。



Codelink：

Veloce提供了一个可以monitor A7行为的codelink model。这个是一个很方便软件debug的功能，可以监控软件的所有行为已经系统中内存以及寄存器的值。支持回放，断点，单步等多种debug方式。

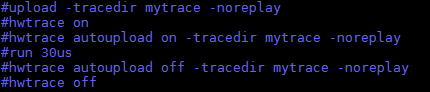


Mutilmedia transactor：

最新版本的软件将analyzer和capture合成为一个叫做sia的软件，这个集成了收数据和分析功能。然后我们可以再打开exerciser发送数据。我们需要将相应的transactor综合进我们的设计里。比如我们在调试HDMI接口的时候，我们就可以将RGBexerciser和RGBcapture编译进去，这样我们可以用exerciser按HDMI的协议发送RGB data给DUT的HDMI接口，然后通过我们的DUT接收处理，再把处理完的data发送给capture。这样我们可以在我们的PC上很直观并且实时的看到一帧一帧的图像信息，相当方便我们debug。

全波形：

当发现结果与预期不符，而且需要debug的信号没有加入xwave中，这时我们可以通过抓全部信号的波形进行debug。可以在run.tcl里加入如下命令进行操作。



建议一般抓小于100us的波形，因为这个会抓取整个设计的所有信号，所以数据量很大，而且抓取的时候Veloce只做数据传输，会进行得比较慢，所以需要验证人员能比较精确的定位需要debug的时间点。建议是可以根据设计的代码去找相应的reg翻转点设置trigger点，用trigger的方式去抓对应的波形。

1. 总结

在大规模SOC系统功能验证和软硬件联合仿真中，Veloce验证平台可以大大缩短验证周期,同时也实现了软硬件的并行开发，为项目的研发争取了宝贵的时间。另外Veloce还具有非常友好强大的图形界面以及多样化的debug功能，大容量，通用化。Veloce硬件加速验证平台结合了EDA工具的强大的debug功能和FPGA原型验证的速度，在Sirius项目SOC系统功能验证和软硬件联合仿真中发挥了巨大的作用。