
Cache Debugger设计手册

Cache小组

2015年1月11日

目录

1 引言

1.1 开发信息

系统名称：硬件调试工具Cache Debugger

开发者：计23 李天润

计23 胡津铭

计23 孙皓

1.2 内容简介

本文档介绍了硬件调试工具的功能设计以及软硬件部分实现细节，方便读者更深入了解本工具的运行方式，以及根据自己的需要对工具进行修改、开发。

本文档假设读者已经阅读过调试工具的使用手册，并且了解硬件描述语言的基本应用以及元件例化的方法。

2 设计原理

硬件逻辑分时序逻辑与组合逻辑，其中组合逻辑仅与输入有关，时序逻辑受时钟控制。

在系内课程与FPGA有关的实验中，并没有涉及与延时相关的逻辑，比如振荡电路，因此FPGA芯片内部的硬件延时在设计硬件逻辑时一般不会被考虑，仅影响时钟频率的提高。基于以上考虑，可以认为大多数实验中，硬件逻辑的执行是由时钟驱动的，因此控制了时钟也就控制了逻辑的执行。而且，由于设计逻辑时不会将延时作为正面的逻辑功能的一部分，时钟沿无论延迟多久到达都不会产生负面作用，因此暂停时钟相当于将电路暂停，之后再次给入时钟信号即可无损地继续执行程序。

本工具正是基于以上结论，结合调试的断点信息，产生被调试模块的时钟信号，使得被调试模块既可正常运行，又可以在给定的状态下暂停，查看调试信息。

通过逻辑操作，本工具可以为被调试模块提供与调试工具的输入时钟信号相同频率的时钟信号，使得被调试模块在调试环境下有着与实际单独运行时几乎相同的表现。

3 硬件部分

3.1 整体设计

硬件部分，需要在软件客户端的控制下，产生适当的时钟信号，控制程序运行指定的时钟数或者运行到指定的断点状态。

运行结束之后，应当将硬件上的信号值发送到客户端，以供调试人员查看信号值，分析芯片运行状况。

调试工具的硬件部分工作流程如下：

上述功能由两个模块实现，一个模块用于与客户端进行通信，另一个模块用于控制被调试模块的时钟信号以及调试工具的状态跳转。

3.2 模块设计

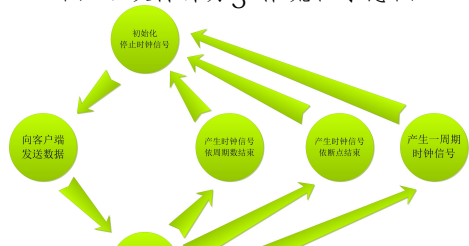
3.2.1 控制模块

此模块位于bp_debug.vhd
端口说明

端口名	端口方向	端口类型
		端口描述
clk	in	std_logic 时钟输入。
clk_o	out	std_logic 由此模块产生的时钟信号，可直接作为被测试模块的时钟。
e	in	std_logic 使能信号。
datas	in	std_logic_vector (4095 downto 0) 从被测试模块发来的信号值数据。
monitor	in	std_logic_vector (63 downto 0) 从被测试模块发来的监控信号数据，用于断点的监控。
serialport_txd	out	std_logic 串口写端口。
serialport_rxd	in	std_logic 串口读端口。

内部实现
控制模块的状态机大致与调试工具硬件部分的工作流程相对应。
需要查看的信号值部分，从被调试模块获取信号值（默认长度为4096bit），使用组合逻辑的方法根据通信模块给出的字节编号选取对应的字节提供给通信模块。
调试部分，目前仅支持在上升沿之前断点，实现方式为输出时钟由两个中间值取异或得到。一个中间值的改变由输入时钟下降沿触发，在输出时钟为高电平时对其取反。另一个中间值的改变由输入时钟上升沿触发，在程序运行阶段，且仍需继续运行时，如果输出时钟为低电平，则将其取反。
被调试模块是否应当继续运行，根据指令的不同而有不同的判断方式。
对于断点调试而言，当检测信号与段电信号值相同时应当进行停止运行，否

图 1: 硬件部分工作流程示意图



则应当继续运行。

对于继续调试而言，应当无条件给出一个周期的时钟信号，然后按照断点调试判断。

对于单步调试而言，设置一个倒计时计数器，每次时钟上升沿减一且被调试模块运行，减到0时被调试模块停止运行。

3.2.2 通信模块

此模块位于com_debug.vhd

端口说明

端口名	端口方向	端口类型
	端口描述	
clk	in	std_logic 串口发送部分开始信号。
slc	out	std_logic_vector(15 downto 0) 串口发送的字节编号，用于从上层模块获取对应字节。
data	in	std_logic_vector (7 downto 0) 串口发送的字节值，使用编号slc从上层模块获取。
serialport_txd	out	std_logic 串口写端口。
serialport_rxd	in	std_logic 串口读端口。
data_coming	out	std_logic 标志位，表示客户端指令到来。
coming_data	out	std_logic_vector (135 downto 0) 来自客户端的指令信息。高64位为客户端第二个参数；中64位为客户端第一个参数；低8位表示指令类型。
data_read	in	std_logic 标志位，表示这一指令已经被读取，可以消除指令到来信号。

内部实现

通信模块包括接受与发送两部分，分别工作，相互不影响，通过软件端、控制模块保证发送与接受的顺序。

发送部分用于运行结束时，向客户端发送需要的内部信号值。收到控制端的发送开始信号后，从0号字节开始循环发送。这一部分使用slc端口给出字节编号，然后由data读入字节信息，通过串口发送。

接收部分从软件端接受指令，收到17个字节后，通过标志位通知控制模块数据到来，同时提供数据。收到控制模块读走数据的信号后，接收部分将数据到来标志位清零，同时继续等待新数据到来。

4 复杂应用

此部分涉及较多细节，请在理解本文档前述部分之后再进行阅读。

4.1 断掉调试

本调试工具的断点调试可以按照64位均匹配的方式执行，也可以根据需要进行一定的统配匹配。

对于包含通配符的断点信息，来自客户端的指令的第二个参数不再是0xFFFFFFFF，而在需要统配的位上为0。断点的匹配是将断点值与监控值作异或，如果结果为0x00000000，则表示到达断点。考虑通配符后，异或的结果会再与统配值进行与操作，将被设为统配的位置0，从而使其不再参与断点匹配。

4.2 观察信号变长

默认文件中，观察信号长度为4096bit，使用中可以将CPU顶层模块的所有信号以及通用寄存器、CPO寄存器的所有信号发到客户端，而且发送这一长度的观察信号消耗的时间在通过终端查看的情况下可以忽略。

使用中，考虑到不同的使用者可能有不同的需要，比如自动调试中希望每次发送更少的信号值或者需要发送更多的信号，在此给出改变观察信号长度的方法。

需要注意的是，数据发送以字节为单位，因此需要保证观察信号的长度为8的倍数。

改变观察信号长度后，需要将被调试模块中的输出数据长度改变，调试模块的输入数据长度改变。

接下来，需要将数据发送部分的字节编号范围改为需要的数值，此处需修改com_debug.vhd中的常量slc_max。

最后，需要在软件端进行相应修改，使其一次性接受的信号数量匹配。