Application Note 借助 Lauterbach® 在 AM26x 器件上启用跟踪

TEXAS INSTRUMENTS

Shaunak Deshpande, Nilabh Anand, Aakash Kedia

摘要

跟踪是一项实时监测软件的技术,可帮助开发人员调试和诊断应用的问题、异常和运行时行为。跟踪也可用于性能基准标记或记录。实时跟踪是解决复杂问题的理想选择。

Lauterbach[®] 是全球公认的嵌入式系统开发工具提供商,专注于高性能调试和跟踪设计。Lauterbach 提供的 TRACE32[®] 工具套件集成了硬件和软件,能够为高速跟踪和调试、代码分析及实时跟踪提供全面的支持。工具套 件广泛用于各个行业,可优化和验证嵌入式软件。对于 AM26x 器件,Lauterbach 工具有助于无缝调试并支持详 细查看系统行为,使开发人员能够高效地解决嵌入式软件开发中的难题。通过使用 Lauterbach 的 TRACE32 工具 进行跟踪,开发人员可以详细了解 AM26x 器件上的软件执行。通过记录执行指令、存储器访问和外设交互的序 列,跟踪功能有助于识别性能问题、调试复杂场景以及验证软件正确性。对于时序和执行流程非常关键的实时嵌 入式系统,此功能至关重要。Lauterbach 的非侵入式跟踪方法与事件时间戳等功能相结合,可在保持系统完整性 的同时支持深入分析,最终提高开发效率和系统可靠性。

ARM R5F 内核以及 ARM M4 内核支持 Lauterbach 跟踪。本文档提供了为德州仪器 (TI) 高性能 AM26x 微控制器 启用 Lauterbach ETM 跟踪的分步指南。

	备注
本文档版本仅适用于 AM263x 和 AM263Px 器件。	

内容	
1 首字母缩略词列表	3
2 软件设置	3
3 硬件设置	4
3.1 AM263x 连接	4
3.2 AM263Px 连接	5
3.3 Lauterbach® 连接	6
4 构建 MCU_PLUS_SDK 示例	8
4.1 CCS 导入和构建	8
4.2 命令行构建	8
5 CMM 脚本	8
5.1 AM263x CMM 脚本	8
5.2 AM263Px CMM 脚本	11
6 刷写 SBL Null	14
6.1 使用 UniFlash 工具	14
6.2 使用命令行 Python 脚本	14
7 使用 Trace32 软件进行调试	15
8 总结	16
9参考资料	16

商标

Code Composer Studio[™] is a trademark of Texas Instruments. FreeRTOS[™] is a trademark of Amazon Web Services, Inc. Lauterbach[®] and TRACE32[®] are registered trademarks of Lauterbach GmbH. Windows[®] is a registered trademark of Microsoft Corporation.



Linux[®] is a registered trademark of Linus Torvalds. 所有商标均为其各自所有者的财产。

1 首字母缩略词列表

- 1. ETM 嵌入式跟踪宏单元
- 2. MCU 微控制器单元
- 3. PRU 可编程实时单元
- 4. SDK 软件开发套件
- 5. SBL 次级引导加载程序
- 6. OSPI 八线串行外设接口
- 7. QSPI 四线串行外设接口
- 8. GPIO 通用输入输出
- 9. IOMUX 引脚多路复用
- 10. I2C 内部集成电路
- 11. ROM 只读存储器
- 12. CPU 中央处理单元

2 软件设置

若要在 AM26x 器件上运行 MCU_PLUS_SDK 应用,需要安装以下软件和工具:

- 1. MCU_PLUS_SDK (版本 10.01 及更高版本)
 - a. AM263x: AM263x MCU_PLUS_SDK 的下载链接
 - b. AM263Px: AM263Px MCU PLUS SDK 的下载链接
- 2. Code Composer Studio[™]: Code Composer Studio (CCS) 的下载链接
- 3. Syscfg: Sysconfig 工具的下载链接
- 4. TI-ARM-CLANG 编译器:TI-ARM-CLANG 编译器的下载链接
- 5. TI UniFlash 工具(可选): TI UniFlash 工具的下载链接
- 6. Python3: Python3的下载链接
- 7. OpenSSL: OpenSSL 的下载链接

如果用户需要进一步帮助,请参阅官方文档的下载、安装和设置 SDK 和工具页面。

Lauterbach 软件 - 可从以下位置下载 Trace32 软件包:Lauterbach 支持与培训。对于 Windows[®],请将该软件 包安装在 C:\T32 中,对于 Linux[®],请将该软件包安装在默认位置。



3 硬件设置

下面列出了启用 Lauterbach® 跟踪所需的硬件:

- 1. AM26x 微控制器:
 - a. AM263x controlCARD : TMDSCNCD263。
 - b. AM263Px controlCARD : TMDSCNCD263P。
- 2. HSEC 集线站和分线板:TMDSHSECDOCK-AM263。
- 3. 适用于 HSEC 集线站分线板的标准电源。
- 4. USB Type A 转 USB micro-B 电缆,用于 JTAG、XDS110 与 AM26x 微控制器的连接。
- 5. Lauterbach 连接器适配器。
- 6. 跟踪探头。
- 7. PowerView 迹线。
- 8. JTAG 电缆。
- 9. Lauterbach 的电源。
- 10. 跟踪带状电缆。

3.1 AM263x 连接

- 1. 将 AM263x 器件对接到 HSEC 集线站分线板上。
- 2. 将电源连接到 HSEC 集线站分线板。
- 3. 将 USB Type-A 转 micro-B 从主机 PC 连接到 AM26x 微控制器。
- 4. 在 AM263Px 上,将 SW-5 开关置于断开状态以断开板载调试器。

开关	状态
SW-5	低

5. 为 HSEC 集线站分线板上电。用户此时看到 AM263x 上的 LD1、LD6、LD14、LD15 亮起。



图 3-1. AM263x PCB# PROC E2 SW-5 开关





图 3-2. 安装在 HSEC 集线站板上的 AM263x 控制卡

3.2 AM263Px 连接

AM263Px 原理图与 AM263x 不同。因此,对于 AM263Px,某些步骤可能会不同。

- 1. 将 AM263Px 器件放置在 HSEC 集线站分线板上。
- 2. 将电源连接到 HSEC 集线站分线板。
- 3. 将 USB Type-A 转 micro-B 从主机 PC 连接到 AM26x 微控制器。
- 4. 在 AM263Px 上,将 SW-1 开关置于关断状态以断开板载调试器。接下来,将 SW-14 开关置于关断状态,将 SW-15 开关置于关断状态,将 SW-16 开关置于导通状态。这是将信号路由到 Lauterbach 迹线引脚所连接的 HSEC 板所必需的。

开关	状态
SW-1	低
SW-14	低
SW-15	低
SW-16	高

5. 为 HSEC 集线站分线板上电。AM263Px 上的 LD2、LD4、LD5、LD9 亮起。









图 3-4. 安装在 HSEC 集线站板上的 AM263Px 控制卡

3.3 Lauterbach® 连接

- 1. 将 Lauterbach 适配器连接到集线站板。
- 2. 将 Lauterbach 跟踪探头连接到与连接到集线站板的适配器相同的端口(A或B)。
- 3. 确保跟踪探头和 PowerView 迹线已正确连接到适配器。
- 4. 在 PowerView 迹线与适配器之间连接 JTAG 电缆。
- 5. 将 Lauterbach 电源连接到 PowerView 迹线并为硬件上电。
- 6. 在跟踪探头与适配器之间连接跟踪带状电缆。





图 3-5. Lauterbach 设置连接



图 3-6. Lauterbach 设置连接

现在,用户可以打开 Lauterbach 电源。



4 构建 MCU_PLUS_SDK 示例

小心

调试防火墙由 hsmRtImg 打开。此图形显示在 <mcu_plus_sdk>/source/security/security_common/ drivers/hsmclient/soc/<device_name>/hsmRtImg.h 上

如果用户使用的是较早的 MCU_PLUS_SDK 版本 (在 v10.01 之前) ,请将 hsmRtImg.h 替换为更新 后的文件:

AM263x - Github

AM263Px - Github

在本应用手册中,用户使用 MCU_PLUS_SDK 示例,在 AM26x MCU 上运行这些示例并获取跟踪。如果用户希望 使用不同的应用,请跳过本节。确保已构建.debug 配置以获得一致的跟踪结果。

4.1 CCS 导入和构建

- 1. 使用此处提到的步骤在 CCS 中导入应用:使用 SDK 及 CCS 工程
- 2. 在工程视图窗口中右键点击工程,然后在调试配置中构建应用。这会生成.out 二进制文件,该文件被加载到 AM26x 器件以用于调试。

4.2 命令行构建

1. 从顶层 MCU_PLUS_SDK 文件夹中打开终端窗口并使用 GNU Make 命令构建应用。例如,

#TO CLEAN
gmake -sj -C examples/drivers/gpio/gpio_led_blink/am263px-cc/r5fss0-0_nortos/ti-arm-clang/
PROFILE=debug clean

#TO SCRUB
gmake -sj -C examples/drivers/gpio/gpio_led_blink/am263px-cc/r5fss0-0_nortos/ti-arm-clang/
PROFILE=debug scrub

#TO BUILD
gmake -sj -C examples/drivers/gpio/gpio_led_blink/am263px-cc/r5fss0-0_nortos/ti-arm-clang/
PROFILE=debug all

有关更多详细信息,请参阅使用 makefile 构建 SDK。

备注

若要对 AM263x 运行上述命令,请替换器件名称以使用 am263x 而不是 am263px。

5 CMM 脚本

CMM 是调试器使用的批处理类型脚本语言。下面的 CMM 脚本处理内核的复位和连接、配置 I2C 时钟、跟踪引脚、IO 扩展器配置、生成片外跟踪结果并将它们显示在窗口中。默认情况下,这适用于 R5F 内核 0,并可修改为 针对其他 ARM R5F 和 ARM M4 内核运行。

小心 这些脚本针对基于 NoRTOS 的应用程序进行了验证。需要执行一些额外步骤才能获取 FreeRTOS™ 任务表和 FreeRTOS 组件详细信息。请参阅操作系统感知手册 FreeRTOS。

5.1 AM263x CMM 脚本

AM263x CMM 脚本

8

; -----: : Copyright (C) 2023-2024 Texas Instruments Incorporated

CMM 脚本

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of Texas Instruments Incorporated nor the names of contributors can be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. @Title: Simple demo script for Cortex-R5F on TMDSCNCD263 @Description: Prompts the user to load application into RAM and sets up a demo debug scenario for the first Cortex-R5F of the SSO cluster. In addition, configures the off-chip trace. Prerequisités: * Plug TMDSCNCD263 onto TI debug&trace adapter TMDSHSECDOCK-AM263 * Connect Debug Cable to J13 via adapter LA-3818 * SW5=OFF to deactivate onboard debugger and enable external JTAG debugger @Keywords: ARM, Cortex-R5F @Board: TMDSCNCD263 @Chip: AM2634 @Copyright: 2023-2024 Texas Instruments Incorporated &sbl=TRUE() ; When running in QSPI Bootmode RESet SYStem.CPU AM2634-SS0 CORE.ASSIGN 1. IF &sbl==FALSE() ; In case of QSPI bootmode, SBL takes care of the below config (_____ Attach to system bus and make some preparations SYStem Mode PREPARE Unlock MSS_CTRL register Data.Set EAHB:0x50D01008 %Long 0x01234567 ; MSS_CTRL_LOCK0_KICK0 Data.Set EAHB:0x50D0100c %Long 0x0fedcba8 ; MSS_CTRL_LOCK0_KICK1 ; Eclipse ROM, use RAM in ATCM Data.Set EAHB:0x50D00080 %Long 0x00000007 ; MSS_CTRL_R5SS0_ROM_ECLIPSE : Let core run Data.Set EAHB:0x50D00024 %Long 0x00000000 ; MSS_CTRL_R5SS0_CORE0_HALT) Attach to the cores SYStem Mode Attach Break IF &sbl==TRUE() (Data.Set EAHB:0x50D01008 %Long 0x01234567 ; MSS_CTRL_LOCK0_KICK0
Data.Set EAHB:0x50D0100c %Long 0x0fedcba8 ; MSS_CTRL_LOCK0_KICK1

CMM 脚本

)



Disable the MPU and the caches that have been enabled by the firmware (SCTLR) Data.Set C15:0x1 %Long (Data.Long(C15:0x1)&(~0x1005)) _____ _____ Load demo program, opens up a file explorer. Navigate and select the application binary you wish data.LOAD.Elf * _____ Configure off-chip trace ÍF Analyzer()||CAnalyzer() ; Unlock TOP_RCM register Data.Set EAHB:0x53201008 %Long 0x01234567 ; TOP_RCM_LOCK0_KICK0 Data.Set EAHB:0x5320100C %Long 0x0fedcba8 ; TOP_RCM_LOCK0_KICK1 WAIT 1.MS IF &sbl==FALSE() ; Incase of OSPI Bootmode, SBL does the PLL clock configuration (; Config core PLL , config cone y co) ; Select trace clock source and divider Data.Set EAHB:0x53200C20 %Long 0x00000222 ; TOP_RCM_TRCCLKOUT_CLK_SRC_SEL Data.Set EAHB:0x53200C24 %Long 0x00000222 ; TOP_RCM_TRCCLKOUT_DIV_VAL Unlock IOMUX register and configure IOs Data.Set EAHB:0x53100298 %Long 0x83e70b13 ; IOMUX_IO_CFG_KICK0 Data.Set EAHB:0x5310029C %Long 0x95a4f1e0 ; IOMUX_IO_CFG_KICK1 WAIT 1.MS Data.Set EAHB:0x53100064 %Long 0x00000501 ; IOMUX_UART0_RTSN_CFG_REG Data.Set EAHB:0x53100068 %Long 0x00000501 ; IOMUX_UART0_CTSN_CFG_REG Data.Set EAHB:0x531000B0 %Long 0x000007F7 ; IOMUX_EPWM0_B_CFG_REG (GPI044 -> input + pull-high) Data.Set EAHB:0x531000BC %Long 0x000007F7 ; IOMUX_EPWM2_A_CFG_REG (GPI047 -> input + pull-high) Data.Set EAHB:0x5310029C %Long 0x95a4fle0 ; IOMUX_IO_CFG_KICK1 WAIT 1.MS Data.Set EAHB:0x53100298 %Long 0x83e70b13 ; IOMUX_IO_CFG_KICK0 Data.Set EAHB:0x531001DC %Long 0x004 ; IOMUX_PR0_PRU1_GP019_CFG_REG (TRC_CLK) Data.Set EAHB:0x531001E0 %Long 0x204 ; IOMUX_PR0_PRU1_GP018_CFG_REG (TRC_CTL) Data.Set EAHB:0x5310019C %Long 0x204 ; Data.Set EAHB:0x5310019C %Long 0x204 ; IOMUX_PR0_PRU1_GP05_CFG_REG (TRC_DATAD0) IOMUX_PR0_PRU1_GP09_CFG_REG (TRC_DATAD1) Data.Set EAHB:0x531001A4 %Long 0x204 ; IOMUX_PR0_PRU1_GP010_CFG_REG (TRC_DATAD2) Data.Set EAHB:0x531001A8 %Long 0x204 ; IOMUX_PR0_PRU1_GP08_CFG_REG (TRC_DATAD3) IF Analyzer() Data.Set EAHB:0x531001AC %Long 0x204 ; IOMUX_PR0_PRU1_GP06_CFG_REG (TRC_DATAD4) Data.Set EAHB:0x531001B0 %Long 0x204 IOMUX_PR0_PRU1_GPO4_CFG_REG (TRC_DATAD5) Data.Set EAHB:0x531001B0 %Long 0x204 ; Data.Set EAHB:0x531001B4 %Long 0x204 ; IOMUX_PR0_PRU1_GPO0_CFG_REG (TRC_DATAD6) Data.Set EAHB:0x551001B4 %Long 0x204 ; IOMUX_PR0_PRU1_GP01_CFG_REG (TRC_DATAD7) Data.Set EAHB:0x531001B8 %Long 0x204 ; IOMUX_PR0_PRU1_GP02_CFG_REG (TRC_DATAD8) Data.Set EAHB:0x531001C0 %Long 0x204 ; IOMUX_PR0_PRU1_GP03_CFG_REG (TRC_DATAD9) Data.Set EAHB:0x531001C4 %Long 0x204 IOMUX_PR0_PRU1_GP016_CFG_REG (TRC_DATAD10) IOMUX_PR0_PRU1_GP015_CFG_REG (TRC_DATAD11) Data.Set EAHB:0x531001C4 %Long 0x204 ; Data.Set EAHB:0x531001C8 %Long 0x204 ; Data.Set EAHB:0x531001CC %Long 0x204 ; Data.Set EAHB:0x531001CC %Long 0x204 ; Data.Set EAHB:0x531001D0 %Long 0x204 ; Data.Set EAHB:0x531001D4 %Long 0x204 ; IOMUX_PR0_PRU1_GP011_CFG_REG (TRC_DATAD12) IOMUX_PR0_PRU1_GP012_CFG_REG (TRC_DATAD13) IOMUX_PR0_PRU1_GP013_CFG_REG (TRC_DATAD14) Data.Set EAHB:0x531001D8 %Long 0x204 ; IOMUX_PR0_PRU1_GP014_CFG_REG (TRC_DATAD15)) ; Use I2C to control the GPIO expander (TCA6416) on the control card to route signals to the docking station Data.Set EAHB:0x52502024 %Long 0x00004620 ; I2C2_ICMDR Data.Set EAHB:0x5250200C %Long 0x00000009 ; I2C2_ICCLKL Data.Set EAHB:0x52502010 %Long 0x00000009 ; I2C2_ICCLKH Data.Set EAHB:0x5250201C %Long 0x00000020 ; I2C2_ICSAR Data.Set EAHB:0x52502020 %Long 0x00000006 ; I2C2_ICDXR;

> Data.Set EAHB:0x52502024 %Long 0x00006e20 ; I2C2_ICMDR; Data.Set EAHB:0x52502020 %Long 0x00000003 ; I2C2_ICDXR; Data.Set EAHB:0x52502024 %Long 0x00006c20 ; I2C2_ICMDR IF Analyzer() (Trace.METHOD Analyzer TPIU.PortSize 16) ELSE (Trace.METHOD CAnalyzer TPIU.PortSize 4) TPIU.PortMode Continuous Trace.TERMination ON Trace.AutoFocus) _____ Open some windows WinCLEAR Mode.H]] WinPOS 0. 0. 116. 26. List.auto WinPOS 120. 0. 100. 8. Frame.view WinPOS 120. 14. Var.Watch Var.Addwatch %SpotLight ast flags WinPOS 120. 25. Trace.List WinPOS 0. 32. ;Var.DRAW %DEFault sinewave ENDDO

5.2 AM263Px CMM 脚本

AM263Px CMM 脚本

_____ Copyright (C) 2023-2024 Texas Instruments Incorporated Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of Texas Instruments Incorporated nor the names of the contributors can be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. _____ @Title: Simple demo script for Cortex-R5F on TMDSCNCD263P @Description:





Prompts the user to load application into RAM and sets up a demo debug scenario for the first Cortex-R5F of the SSO cluster. In addition, configures the off-chip trace. Prerequisites: * Plug TMDSCNCD263P onto TI debug&trace adapter TMDSHSECDOCK-AM263 * Connect Debug Cable to J13 via adapter LA-3818 * SW1=OFF to deactivate onboard debugger and enable external JTAG debugger * SW14=OFF, SW15=OFF, SW16=ON to have signals routed from PRU1 to HSEC @Keywords: ARM, Cortex-R5F @Board: TMDSCNCD263P @Chip: AM263P4 @Copyright: 2023-2024 Texas Instruments Incorporated &sbl=TRUE() ; When running in OSPI Bootmode RESet SYStem.CPU AM263P4-SS0 CORE.ASSIGN 1. IF &sbl==FALSE() ; In case of OSPI bootmode, SBL takes care of the below config (_____ Attach to system bus and make some preparations SYStem.Mode PREPARE ; Unlock MSS_CTRL register Data.Set EAHB:0x50D01008 %Long 0x01234567 ; MSS_CTRL_LOCK0_KICK0 Data.Set EAHB:0x50D0100c %Long 0x0fedcba8 ; MSS_CTRL_LOCK0_KICK1 ; Eclipse ROM, use RAM in ATCM Data.set EAHB:0x50D00080 %Long 0x00000007 ; MSS_CTRL_R5SS0_ROM_ECLIPSE Let core run Data.Set EAHB:0x50D00024 %Long 0x00000000 ; MSS_CTRL_R5SS0_CORE0_HALT _____ _____ Attach to the cores SYStem.Mode Attach Break IF &sbl==TRUE() (Data.Set EAHB:0x50D01008 %Long 0x01234567 ; MSS_CTRL_LOCK0_KICK0 Data.Set EAHB:0x50D0100c %Long 0x0fedcba8 ; MSS_CTRL_LOCK0_KICK1) Disable the MPU and the caches that have been enabled by the firmware (SCTLR) Data.Set C15:0x1 %Long (Data.Long(C15:0x1)&(~0x1005)) _____ Load demo program, opens up a file explorer. Navigate and select the application binary you wish data.LOAD.Elf _____ Configure off-chip trace IF Analyzer()||CAnalyzer() ; Unlock TOP_RCM register Data.Set EAHB:0x53201008 %Long 0x01234567 ; TOP_RCM_LOCK0_KICK0 Data.Set EAHB:0x5320100C %Long 0x0fedcba8 ; TOP_RCM_LOCK0_KICK1 WAIT 1.MS IF &sbl==FALSE() ; Incase of OSPI Bootmode, SBL does the PLL clock configuration (; Config core PLL Data.Set EAHB:0x53200410 %Long 0x00010009 ; TOP_RCM_PLL_CORE_M2NDIV Data.Set EAHB:0x53200414 %Long 0x00000320 ; TOP_RCM_PLL_CORE_MN2DIV Data.Set EAHB:0x53200408 %Long 0x00000001 ; TOP_RCM_PLL_CORE_TENABLE Data.Set EAHB:0x53200404 %Long 0x00095001 ; TOP_RCM_PLL_CORE_CLKCTRL Data.Set EAHB:0x53200430 %Long 0x00000103 ; TOP_RCM_PLL_CORE_HSDIVIDER_CLKOUT1) ; Select trace clock source and divider Data.Set EAHB:0x53200C20 %Long 0x00000222 ; TOP_RCM_TRCCLKOUT_CLK_SRC_SEL Data.Set EAHB:0x53200C24 %Long 0x00000222 ; TOP_RCM_TRCCLKOUT_DIV_VAL

; Unlock IOMUX register and configure IOs Data.Set EAHB:0x53100298 %Long 0x83e70b13 ; IOMUX_IO_CFG_KICK0 Data.Set EAHB:0x5310029C %Long 0x95a4f1e0 ; IOMUX_IO_CFG_KICK1 WAIT 1.MS Data.Set EAHB:0x53100064 %Long 0x00000501 ; IOMUX_UART0_RTSN_CFG_REG
Data.Set EAHB:0x53100068 %Long 0x00000501 ; IOMUX_UART0_CTSN_CFG_REG Data.Set EAHB:0x531000B0 %Long 0x000007F7 ; IOMUX_EPWM0_B_CFG_REG (GPI044 -> input + pull-high) Data.Set EAHB:0x531000BC %Long 0x000007F7 ; IOMUX_EPWM2_A_CFG_REG (GPI047 -> input + pull-high) Data.Set EAHB:0x53100298 %Long 0x83e70b13 ; IOMUX_IO_CFG_KICK0
Data.Set EAHB:0x5310029C %Long 0x95a4f1e0 ; IOMUX_IO_CFG_KICK1 WAIT 1.MS Data.Set EAHB:0x531001DC %Long 0x004 ; IOMUX_PR0_PRU1_GP019_CFG_REG (TRC_CLK) Data.Set EAHB:0x531001E0 %Long 0x204 IOMUX_PR0_PRU1_GP018_CFG_REG (TRC_CTL) Data.Set EAHB:0x5310019C %Long 0x204 IOMUX_PR0_PRU1_GP05_CFG_REG (TRC_DATAD0) Data.Set EAHB:0x531001A0 %Long 0x204 ; IOMUX_PR0_PRU1_GP09_CFG_REG (TRC_DATAD1) Data.Set EAHB:0x531001A4 %Long 0x204 ; IOMUX_PR0_PRU1_GP010_CFG_REG (TRC_DATAD2) Data.Set EAHB:0x531001A8 %Long 0x204 ; IOMUX_PR0_PRU1_GP08_CFG_REG (TRC_DATAD3) IF Analyzer() (Data.Set EAHB:0x531001AC %Long 0x204 ; IOMUX_PR0_PRU1_GP06_CFG_REG (TRC_DATAD4) Data.Set EAHB:0x531001B0 %Long 0x204 ; IOMUX_PR0_PRU1_GP04_CFG_REG (TRC_DATAD5) Data.Set EAHB:0x531001B4 %Long 0x204 ; IOMUX_PR0_PRU1_GPO0_CFG_REG (TRC_DATAD6) Data.Set EAHB:0x531001B8 %Long 0x204 IOMUX_PR0_PRU1_GP01_CFG_REG (TRC_DATAD7) Data.Set EAHB:0x531001BC %Long 0x204 ; IOMUX_PR0_PRU1_GPO2_CFG_REG (TRC_DATAD8) Data.Set EAHB:0x531001C0 %Long 0x204 ; Data.Set EAHB:0x531001C4 %Long 0x204 ; IOMUX_PR0_PRU1_GP03_CFG_REG (TRC_DATAD9) IOMUX_PR0_PRU1_GP016_CFG_REG (TRC_DATAD10) Data.Set EAHB:0x531001C4 %Long 0x204 ; Data.Set EAHB:0x531001C8 %Long 0x204 ; IOMUX_PR0_PRU1_GP015_CFG_REG (TRC_DATAD11) IOMUX_PR0_PRU1_GP011_CFG_REG (TRC_DATAD12) Data.Set EAHB:0x531001D0 %Long 0x204 ; IOMUX_PR0_PRU1_GP012_CFG_REG (TRC_DATAD13) Data.Set EAHB:0x531001D4 %Long 0x204 ; Data.Set EAHB:0x531001D8 %Long 0x204 ; IOMUX_PR0_PRU1_GP013_CFG_REG (TRC_DATAD14) IOMUX_PR0_PRU1_GP014_CFG_REG (TRC_DATAD15)) ; Use I2C to control the GPIO expander (TCA6424) on the control card to route signals to the docking station Data.Set EAHB:0x52502024 %Long 0x00004620 ; I2C2_ICMDR Data.Set EAHB:0x5250200C %Long 0x00000009 ; I2C2_ICCLKL Data.Set EAHB:UX52502010 %Long 0x00000009 ; I2C2_ICCLKH Data.Set EAHB:0x5250201C %Long 0x00000022 ; I2C2_ICSAR Data.Set FAHB:0x52502020 %Long 0x00000022 ; I2C2_ICSAR Data.Set EAHB:0x52502020 %Long 0x0000000C ; I2C2_ICDXR Data.Set EAHB:0x52502024 %Long 0x00006e20 ; I2C2_ICMDR Data.Set EAHB:0x52502020 %Long 0x00000006 ; I2C2_ICDXR Data.Set EAHB:0x52502024 %Long 0x00006c20 ; I2C2_ICMDR IF Analyzer() Trace.METHOD Analyzer TPIU.PortSize 16) ELSE Trace.METHOD CAnalyzer TPIU.PortSize 4) TPIU.PortMode Continuous Trace.TERMination ON Trace.AutoFocus) _____ Open some windows **win**CLEAR Mode.H11 WinPOS 0. 0. 116. 26. List.auto WinPOS 120. 0. 100. 8. Frame.view WinPOS 120. 14. Var.Watch Var.Addwatch %SpotLight ast flags WinPOS 120. 25. Trace.List

TEXAS INSTRUMENTS www.ti.com.cn

WinPOS 0. 32. ;Var.DRAW %DEFault sinewave ENDDO

6 刷写 SBL Null

TI 建议用户将 SBL Null 刷写到 AM26x 器件中。

6.1 使用 UniFlash 工具

要使用 UniFlash 工具将 SBL Null 映像刷写到器件中,请按照以下步骤操作。

- 1. 请按照此视频中的步骤操作:TI视频。
- 2. 确保选择文件 uart_uniflash 和 sbl_null 进行刷写。

6.2 使用命令行 Python 脚本

要使用 SDK python 脚本进行刷写,请执行以下步骤。

- 1. 关断器件电源并切换到 UART 引导模式。
- 2. 从以下文件夹中: *mcu_plus_sdk/tools/boot*,在命令提示符下运行以下命令。(从器件管理器检查 UART COM 端口)。
- 3. python uart_uniflash.py -p COM<your_port_number> --cfg=sbl_prebuilt/<device>/default_sbl_null.cfg。
- 4. 确认成功刷写并执行上述脚本中的两个步骤。
- 5. 关断器件电源并切换到基于 AM26x 器件的 OSPI/QSPI 引导模式。

有关更多详细步骤,请参阅在 AM26x 器件上刷写应用。



7 使用 Trace32 软件进行调试

Lauterbach 硬件、AM26x 器件和应用已准备好进行跟踪。要对应用提取跟踪,请执行以下步骤:

- 1. 启动 TRACE32 软件。
- 2. 从 T32Start 显示屏中选择 PowerView 实例, 然后点击 Start。



图 7-1. T32 启动窗口

- 3. 将器件设置为 OSPI、QSPI 引导模式,并对器件进行下电上电(HSEC 集线站板)。
- 4. 点击 File → Run Script → 选择 .cmm 脚本 *am263px-trace.cmm*。
- 5. 首先,该文件复位 CPU、连接到系统总线并解锁 MSS_CTRL 寄存器,然后切换 ROM 并运行 R5F 内核。然后,该文件提示用户选择要调试的应用二进制文件。浏览文件资源管理器并选择文件。
- 6. 然后,该脚本会配置跟踪时钟源、IOMUX 和 I2C,以将信号路由到集线站。
- 7. 该脚本会打开窗口以显示调试跟踪。

TRACE32 PowerView for ARM 0 [Power Debug USB @]	×
File Edit View Var Break Run CPU Misc Trace Perf Cov Peripherals Window Help	
H H M ↓ 4 C F U ⊠ ? K ⊕ = = = = ⊕ ⊕ @ @ # ⊕ 1 ⊅	
[Bulistauto] Go Break	🔀 🗟 B-Frams.vinv 💿 🖸 🕄
▶ Step ▶ Over 🙏 Diverge ✔ Return 🔮 Up 🕨 Go 🔢 Break 🎇 Mode 6af t. 😳 Find:	Cl t. Up 🐨 Down 🖉 Args 🗌 Locals 🗋 Caller Task:
addr/tine source	<pre>> Concelled:Piinterive(handle = 0.7000HXX) - 000[period=bereficies() = 0.7000HXX) - 000[period=bereficies() = 0.7000HXX) - 000[period=bereficies() = 0.7000HXX] - 000[per</pre>
Ø Setup iii Groups II Config Q Goto Q Goto ∯ Find ⊕ In +Q+ Out @ Full	V I GO Watch 60 View 💥
PXXESSING .000ms -700.000ms -600.000ms -500.000ms -400.000ms -200.000ms -200.0000ms -200.0000ms -200.0000ms -200.0000ms -200.0000000000000000000000000000000000	Variable watch view
ink/Global/HwiP_restore	B-FaceList 그 브 프 프
Clock/Wei/P.irc.handler- to: Wei/P.irc.handler- to: Clock/EtmeTicLir- P_timeClock/SetmeTicLir- Comments (Clock Setmes clock - Look	
	DODDERTTON DIE DIXTODESECTO DE DANS CAMPANANAN ANNU DI
B::B::List.auto	
[ok] <address> <addressange> options</addressange></address>	previous
ST:700533F6 \\qpio_led_blink\ClockP_nortos\ClockP_isActive	stopped HLL UP

图 7-2. 跟踪结果窗口

有关 AM263Px controlCard 上的 GPIO led 闪烁示例的调试跟踪,请参阅图 8-1。



8 总结

本文档提供了为德州仪器 (TI) 高性能 AM26x 微控制器启用 Lauterbach ETM 跟踪的分步指南。通过执行上述硬件 和软件设置步骤并将 .cmm 脚本与 Trace32 软件配合使用,可在 AM26x 微控制器上启用 Lauterbach 跟踪。图 8-1 展示了 OSPI 引导模式下来自 AM263P4 的 MCU_PLUS_SDK (在 R5F 内核 0 上运行)的 GPIO LED 闪烁示例的跟踪。

TRACE32 PowerView for ARM 0 (Power Debug USB @)	- 0 X			
File Edit View Var Break Run CPU Misc Trace Perf Cov Peripherals Window Help				
Av Batrace.Chart.sYmbol /Track				
🥬 Setup 👖 Groups 😫 Config 🔒 Goto 🔒 Goto 🎒 Find 📀 In 🖓 Out 🚥 Full	L Up L Down Mars Local task: -000[Semaphore2_period(s) = 0x70040482, timeout = 1000)			
PROCESSENC .000ms -600.000ms -400.000ms -200.000ms 0. address () (other) ()	-001[k][k]ckck_s[eepTrks(t1rks = 1000] -002[c]ckck_s]eep(sec = 1) -003[onin_ted blink_main(erns = 0x0)			
_nortos\SemaphoreP_pend ink\Global\HwiP_restore	-004 hmain() -005 hypas_muto_init(asm)			
Thickoloal (Harty_of sables) - Inortos (ClockP_isActives) - (FIFOULT) a	end of frame			
lers_nortos\wipe_getIRQ	& B-Var.Watch			
ars_norcos/mar_isruise Innk/Global/WwiP_enable tos/ClockP_timerTickIsru	V 1 65 Watch 66 View			
P_timerClearOverflowInt : TimerP_clearOverflowInt :	ast = ? flags = ?			
Global Hm/P_irg_handler: lers_nortos Hm/P_ackIRQ+	v .			
Dorotos ClockP_destruct 3 protos ClockP_sleepTicks 3 pabecPP_constructBingry 3				
SemaphoreP_Params_init: os\SemaphoreP_construct:	Setup II Config O Goto ∯ Find A Chart II Profile II MIPS ♦ More I Less			
	record irun laddress cycle data [symbo] [ti.back -0000788862 D:7004D410 rd-long 00002716 _ed_blink\clobal_stack+0x3E80 0.000us			
	-0000788854 T:70057C40 ptracekP_nortos\ClockP_isActive+0x10 0.030us cbz r0,0x70057C42 trace kP_nortos\ClockP_isActive+0x10 0.030us T:70057C42 trace kP_nortos\ClockP_isActive+0x12 0.000us			
🐱 BacCTS.List	+ b 0x70057C44 r movs r0,#0x1			
Destup Destup	172 str r0.[7]			
	-0000788850 D:7004D3E4 wr-long 00000001 .ed_blink\Global_stack+0x3E54 0.000us -0000788844 T:70057C48 ytrace .kP_nortos\ClockP_isActive+0x18 0.020us			
	-0000788842 T:70057C4A ptracekP_nortos/ClockP_isActive+0x1A 0.000us			
167	-0000788838 D:7004D3E4 rd-long 00000001 .ed_blink\Global_stack+0x3E54 0.030us -0000788832 T:70057C4C ptrace .kP_nortos\ClockP_isActive+0x1C 0.020us			
-0000788928 . -0000788914	-0000788830 T:70057C4E ptracekP_nortos\ClockP_isActive+0x1E 0.000us			
169 -0000788884 • • • • 0.090us 0.030us	-0000788826 T:700530FC ptraceePnortos\SemaphoreP_pend+0x6C 0.000us			
-0000788878 · 0.030us 0.070us	180 180 180 180 180 180 180 180 180 180			
172 -0000788552 · 0.070us 0.020us	str r0,[r13] -0000788824 D:70040356 wr-long 00000001 _ed_blink/Global_stack+0x3560 0.020us Tr700023104 ptrans = P portos SamaphoreP pande0v24 0.000us			
-0000788844 ·	-0000788816 5770053106 ptraceeP_nortos\SemaphoreP_pend+0x76 0.030us			
175 -0000788842 · 0.000us 0.050us	-0000788814 D:700403F0 rd-long 00000001 _ed_blink\Global_stack+0x3E60 0.000us T:70053108 ptrace _eP_nortos\SemaphoreP_pend+0x78 0.020us			
-0000788826 180 0.294us	-0000788806 D:7004DF4 wr-long 0000001 _ed_blink/Global/_stack+0x3E64 0.000us			
B::B::SYStem.Mode Attach				
annual and the first form for for first black there for the state of t				
unangenening use una ver una ver una stem signer inder stimol rear ander rear ander inder stimol rear engeneer reu inderstation CALIFE LURE APU REAM LANDER PER PROVID 000 genous				

图 8-1. 采样跟踪输出

9 参考资料

- Lauterbach AM2634 支持, AM2634 器件的官方 Lauterbach 支持, 网页
- Lauterbach,官方 Lauterbach 错误消息指南,错误指南
- Lauterbach,操作系统感知手册 FreeRTOS, FreeRTOS 手册
- 德州仪器 (TI), AM263x 器件的 MCU_PLUS_SDK 官方文档, AM263x MCU_PLUS_SDK
- 德州仪器 (TI), AM263Px 器件的 MCU_PLUS_SDK 官方文档, AM263Px_MCU_PLUS_SDK
- 德州仪器 (TI), AM26x Academy 了解详情并开始开发, TI AM26x Academy
- 德州仪器 (TI), AM26x 视频培训集锦, AM26x 视频培训

重要通知和免责声明

TI"按原样"提供技术和可靠性数据(包括数据表)、设计资源(包括参考设计)、应用或其他设计建议、网络工具、安全信息和其他资源, 不保证没有瑕疵且不做出任何明示或暗示的担保,包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担 保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任:(1) 针对您的应用选择合适的 TI 产品,(2) 设计、验 证并测试您的应用,(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更,恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的相关应用。 严禁以其他方式对这些资源进行 复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索 赔、损害、成本、损失和债务,TI 对此概不负责。

TI 提供的产品受 TI 的销售条款或 ti.com 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址:Texas Instruments, Post Office Box 655303, Dallas, Texas 75265 版权所有 © 2025,德州仪器 (TI) 公司