# User's Guide 使用 Xilinx FPGA 启用 AFE79xx SPI 的指南

# TEXAS INSTRUMENTS

#### Thyagaraju Naidu and Dhruvil Solanki

摘要

本教程介绍了使用 Xilinx Vivado 和 Vitis 开发环境为德州仪器 (TI) AFE79xx EVM 启用串行外设接口 (SPI) 和非时序关键型通用输出 (GPO) 以及配套的 LMK 系列时钟芯片的过程,因此可以更轻松地将 AFE79xx 器件集成到系统设计中。本指南将以 Xilinx ZCU102 为例展开介绍。

内宓

1 <b>1</b> . M.	
1 引言	3
2 先决条件	3
3 典型裸机设计流程	4
4 背景	
5 添加 Microblaze 和 SPI IP 以便在 Vitis 中用千嵌入式开发	
6 在 Vitis 中创建新平台	
7 在 Vitis 中新建应用工程	15
8 构建应用工程	18
9为AFE79xx FVM 生成 SPI 日志	10
91 生成 I MK SPI 日志	
9.2 生成 AFF SPI 日志	20
0.2 生成用TECTTECTTECTTECTTECTTECTTECTTECTTECTTEC	20
5.5 剂 51 T L 芯衣 庆分 Vius 的相式	
10 AT E75XXEVIII 伙 廖以	
11 1 初始化 CDIO	
11.1 初知化 GFIO	
11.2 以且刀凹	
II.3 针对相应的位反直为局电干以低电干	
12 <u> </u>	
13 设置硬件和为硬件加电	
14 为 VADJ_FMC 设置 ZCU102 电路权接口	
15 调试应用工程并设置 Vitis 甲行终端	27
16 执行应用	
17 修订历史记录	

## 插图清单

图 3-1. 裸机设计流程	4
图 4-1. 采用 Microblaze 和 AXI 外设的典型块设计	5
图 5-1. 创建块设计	6
图 5-2. 命名块设计	6
图 5-3. 向块设计中添加 IP	6
图 5-4. 向块设计中添加 Microblaze	7
图 5-5. 为 Microblaze 运行块自动化	7
图 5-6. 为 Microblaze 选择 CLKIN	7
图 5-7. 为 Microblaze 复位连接	8
图 5-8. 向块设计中添加 IP	8
图 5-9. 将 "AXI QUAD SPI" IP 添加到块设计中	9
图 5-10. 针对 "AXI LITE" 运行连接自动化	9
图 5-11. "ext spi clk"在"AXI QUAD SPI"中显示"No Connection"	10

图 5-12. "ext_spi_clk"已连接到"s_axi_aclk"	10
图 5-13. 选择 AXI QUAD SPI 中的 SPI 从站数量	11
图 5-14. 在 "AXI QUAD SPI"中突出显示用于外部连接的端口	11
图 5-15. 验证块设计	12
图 6-1. 新建平台项目	12
图 6-2. 平台项目的命名	
图 6-3. 硬件规格	13
图 6-4. 选择平台	14
图 6-5. 构建新工程	14
图 7-1. 新建应用工程	15
图 7-2. 新应用工程	15
图 7-3. 选择应用工程	16
图 7-4. 新应用工程名称	
图 7-5. 应用工程名称	17
图 7-6. 选择模板	17
图 8-1. 构建工程	
图 14-1. ZCU102 板用户接口	25
图 14-2. 设置 VADJ	25
图 14-3. VADJ_FMC 电压	
图 15-1. 调试应用工程	27
图 15-2. Vitis 串行终端	
图 16-1. 执行应用	

# 表格清单

表 2-1. 先决条件
表 11-1. IP I/O 的位映射

# 商标

所有商标均为其各自所有者的财产。

# 1 引言

本用户指南详细介绍了在使用 Xilinx FPGA 的 AFE79xx 系统中启用 SPI 和 GPO 的完整硬件和软件流程。本例中 的硬件是指基于 Xilinx Microblaze 处理器的块设计以及 AXI SPI、AXI GPIO 和其他所需的外设。

具体分步目标如下:

- 在 Vivado 工程中使用 SPI IP 实例化块设计。
- 将块设计所需的信号映射到 FPGA IO。
- 导入硬件设计并构建新的 Vitis 应用工程以进行软件开发。
- 编译、链接和下载 C 程序至处理器以及 FPGA 的位文件。

## 2 先决条件

为了有效使用本文档,请确保满足以下先决条件:

- Xilinx Vitis IDE v2020.1.0 (或更高版本)
- Xilinx Vivado v2020.1.0(或更高版本)
- Xilinx FPGA 板和 TI AFE EVM
- FPGA 位文件下载/调试编程器
- 用于调试终端的 USB-UART 电缆
- TI 提供的 C-API

#### 表 2-1. 先决条件

TI AFE	AFE79xx
采样配置	2T-2R-1FB
通道数	2 个 RX 通道(1RX、1FB)和 2 个 TX 通道(速率为 5Gbps)
AFE EVM	AFE79xx EVM
FPGA 板	Xilinx ZCU102 EVM



Vitis

Environment

# 3 典型裸机设计流程 Vivado Environment



图 3-1. 裸机设计流程

# 4 背景

此示例使用了软核 Microblaze,因为 Microblaze 可以在大多数 Xilinx FPGA 系列中实例化。SPI、UART 和 GPIO AXI 块在频率相对较低的 AXI 时钟上运行。如图 4-1 所示,AXI 外设由 Microblaze 块通过智能互连进行控制。



图 4-1. 采用 Microblaze 和 AXI 外设的典型块设计

Microblaze 的 HP 端口驱动 AXI 外设块设计。100MHz 差分时钟源需要为整个 IP 提供时钟。此示例使用 100MHz 差分时钟源,因为该时钟在大多数 FPGA EVM 中通常作为"用户时钟"提供。所有其他时钟频率通过时钟向导在 内部生成。根据系统中所需的独立 SPI 总线数量,可将更多 AXI SPI IP 添加到块设计中。



# 5 添加 Microblaze 和 SPI IP 以便在 Vitis 中用于嵌入式开发

- 1. 打开现有的 Vivado 工程或创建新工程。
- 2. 在左窗格中的"IP Integrator"下,点击"Create Block Design"。



图 5-1. 创建块设计

3. 为块设计命名并点击"OK"。

🝌 Create Block Design		×
Please specify name	of block design.	4
<u>D</u> esign name:	design_1	⊗
Directory:	Section 2012 Contemporal Action Project	~
Specify source set:	🖻 Design Sources	~
?	ОК	Cancel

图 5-2. 命名块设计

4. 在新添加的块设计中,点击"+"以添加 IP。

This design is empty. Press the 🕂 button to add IP.

## 图 5-3. 向块设计中添加 IP

5. 搜索 Microblaze 并将"Microblaze"添加到块设计。



#### ENTER to select, ESC to cancel, Ctrl+Q for IP details

图 5-4. 向块设计中添加 Microblaze

6. 点击"Run Block automation",然后点击"OK"。

Diagr	ram	×	Addres	s Edi	tor	×									
Đ,	Q,	23		O	Q,	*	$rac{}{=}$	+	D <sub>5</sub>	s	ľ	*	C	ণ্	Default View
* c	)esign	ier A	ssistand	e ava	ilable	. Ru	n Blo	ick Al	itom	ation					

	microblaze_0	
+ INTERRUPT + DEBUG Clk Reset	MicroBlaze.#	DLMB +
	MicroBlaze	

图 5-5. 为 Microblaze 运行块自动化

- 7. Vivado 会自动为 Microblaze 添加多个 IP 配套块。
- 8. 点击"Run Connection automation"。
- 9. 在连接自动化弹出窗口中,选择"CLK\_IN1\_D",将其映射到"user\_si570\_sysclk",然后点击"OK"。

omatically make connections in your design right.	r by checking the boxes of the interfaces to connect. Select an interface on the left to display its configuration options on
Q X ♦	Description Connect Board Part Interface to IP interface.
✓ ■ ♥ dk_wiz_1 Ø ⊕ CLK_IN1_D >> reset	Interface: /clk_witz_1/ICLK_IN1_D Options
_ ≫ ext_reset_in	Select Board Part Interface user_si570_syscik (User Programmable differential clock ) 💙

#### 图 5-6. 为 Microblaze 选择 CLKIN



- 10. 再次点击"Run Connection automation"。
- **11.** 在连接自动化弹出窗口中,选择"reset"、"ext\_reset\_in",将其映射到"reset (FPGA\_reset)",然后点击"OK"。

omatically make connections in your design right.	n by checking the boxes of the interfaces to connect. Select an interface on the left to display its configuration optic	ons on 🗼
Q   ¥   ≑	Description	
<ul> <li>✓ All Automation (2 out of 2 selected)</li> <li>✓ ♥ ♥ clk_wiz_1</li> <li>✓ ♥ ♥ rst_clk_wiz_1_100M</li> <li>Ø ♥ ♥ rst_clk_wiz_1_100M</li> </ul>	Connect Board Part Interface to IP Interface. Interface: /rsl_clk_wiz_1_100M/ext_reset_in Options	
	Select Board Part Interface reset (FPGA Reset) v	

图 5-7. 为 Microblaze 复位连接

12. 右键点击框设计并添加"AXI Quad SPI",如图 5-8 和图 5-9 所示。



图 5-8. 向块设计中添加 IP



# Address Editor × K S ⊕ Q ≍ ⊕ + ∞ M M C 9 = Default View ×



#### 图 5-9. 将 "AXI QUAD SPI" IP 添加到块设计中

13. 点击"Run Connection automation"。

an connection Automation		
omatically make connections in your desigr right.	by checking the boxes of the interfaces to connect. Select an interface on the left to display its configuration options on	À
Q ≚ ≑	Description	
✓ ■ All Automation (1 out of 2 selected) ✓ ■ ₱ axi_quad_spi_0 ✓ ● AXI_LITE ● ■ \$	Make IP Interface External: /axi_quad_spi_0/SPI_0 Options	

图 5-10. 针对 "AXI\_LITE" 运行连接自动化



14. 选择 "AXI\_LITE" 并点击 "OK" 。







15. 将"s\_axi\_aclk"连接到"ext\_spi\_clk"。



图 5-12. "ext\_spi\_clk"已连接到 "s\_axi\_aclk"



16. 双击 "AXI Quad SPI" ->选择"No. of slaves", 然后点击"OK"。

Enable XIP Mode     Enable Performance Mode	í.				
SPI Options					
Mode	Standard	~			
Transaction Width	8	~			
Frequency Ratio 16	~ X 1			ø	(1128)
No. of Slaves	1	~			
Enable Dual Quad Mode	1	î			
Enable Master Mode	3				
Enable FIFO	4				
FIFO Depth	6		16		~
Flash (Memory) access through	7	ive			
Enable Async Clock Mode (Au	8				
Note:	10	~			

图 5-13. 选择 AXI QUAD SPI 中的 SPI 从站数量

- 17. 从"Quad SPI IP"中,按以下方式映射信号:
  - a. "io0\_o" -> SPI\_SDO
  - b. "lo1\_i" <- SPI\_SDI
  - c. "sck\_o" -> SPI\_SCL
  - d. ss\_o[1:0] -> SPI\_SEN0、SPI\_SEN1
- 18. 系统将根据在步骤 16 中选择的从站数量计算 "ss\_o" 位宽



图 5-14. 在 "AXI QUAD SPI" 中突出显示用于外部连接的端口



19. 验证设计以确保未出现如图 5-15 所示的错误。



图 5-15. 验证块设计

20. 要添加 GPIO,请从目录中添加"AXI GPIO",并重复上述类似步骤。

## 6 在 Vitis 中创建新平台

.

要在 Vitis 中创建新平台,请执行以下步骤:

1. 打开 File 菜单,转到 New,然后点击 Platform Project (请参阅图 6-1)。

File	Edit	Search	Run	Xilinx	Project	Wi	ndow Help		-
	New Onen Fil	_		Alt+Sh	ift+N >		Application Project		* • 0 • 🛷 • 🏷
	Open File	e				1	Hw Kernel Project		
	Close			C	trl+W	E.	Platform Project		
	Close All		C	trl+Sh	ift+W	-	Other	Ctrl. N	
	Save			0	Ctrl+S	-	Other	Cui+iv	]
F.	Save As								
ß	Save All		9	Ctrl+Sł	nift+S	ι.			
	Move								
2	Rename.				F2	ι.			
8	Refresh				F5				
	Import								
	Export					L			
	Propertie	s		Alt+	Enter				

#### 图 6-1. 新建平台项目



#### 2. 输入所需的平台名称。以名称 ZCU102ps 为例 (请参阅图 6-2)。

## Vew Platform Project Create new platform project Enter a name for your platform project This wizard will guide you through creation of a platform project from the output of Vivado [Xilinx Shell Archive (XSA)] or from an exi you to specify options for the kernels, BSPs, as well as settings required for creating new applications. Platforms are currently support Platform project name: ZCU102ps A platform provides hardware informati Platform System A system project contains one or more a Project Project A domain provides runtime for applicat · A workspace can contain unlimited plat Processor Domain Арр XSA A new platform project can be created from one of the two inputs: From hardware specification (XSA) Create a new platform project from a hardware specification file. You can specify the OS and processor to start with. The platfor platform project editor. From existing platform Load the platform definition from an existing platform. You can choose any platform from the platform repository as a base for 图 6-2. 平台项目的命名 3. 命名新平台后,将出现一个菜单(请参阅图 6-3)。选择 XSA (Xilinx 支持存档)文件。

ease selec	t a platform to create the project		
Create a	new platform from hardware (XSA)	Select a platform from repository	
Hardware	e Specification		
	Provide your XSA file or use a pre-bu	ilt board description	
XSA File:	vck190 zc702 zc706 zcu102 zed		Browse
Software Specify th	Specification he details for the initial domain to be a	dded to the platform. More domains can be after the platform is created by double	clicking the platform.spr file
Processo	n	~	

图 6-3. 硬件规格



#### 4. 从 FPGA 文件夹中浏览并选择与本文档共享的 .XSA 文件 (请参阅图 6-4)。

		,	σv	© Search ZCU102_B	EL
ime	Date modified	Туре	Size		
ZCU102_BEL.cache	4/22/2022 8:19 AM	File folder			
ZCU102_BEL.hbs	4/22/2022 8:19 AM	File folder			
ZCU102_BEL.hw	4/22/2022 8:19 AM	File folder			
ZCU102_BEL.ip_user_files	4/22/2022 8:19 AM	File folder			
ZCU102_BEL.runs	4/22/2022 8:20 AM	File folder			
ZCU102_BEL.sim	4/21/2022 1:00 PM	File folder			
ZCU102 BEL.srcs	4/22/2022 8:20 AM	File folder			
TI_204c_IP_ref.xsa	4/21/2022 1:26 PM	XSA File	1	1,563 KB	
	me ZCU102_BEL.cache ZCU102_BEL.hbs ZCU102_BEL.hus ZCU102_BEL.uns ZCU102_BEL.uns ZCU102_BEL.uns ZCU102_BEL.uns ZCU102_BEL.uns	Date modified           ZCU102_BEL.cache         4/22/2022 &19 AM           ZCU102_BEL.bbs         4/22/2022 &19 AM           ZCU102_BEL.bw         4/22/2022 &19 AM           ZCU102_BEL.juser_files         4/22/2022 &19 AM           ZCU102_BEL.juser_files         4/22/2022 &19 AM           ZCU102_BEL.sim         4/22/2022 &20 AM           ZCU102_BEL.arcs         4/22/2022 &20 AM           TL204c_IP_refxxxx         4/22/2022 &120 PM	me Date modified Type ZCU102_BEL.csche 4/22/2022 8:19 AM File folder ZCU102_BEL.hbs 4/22/2022 8:19 AM File folder ZCU102_BEL.hw 4/22/2022 8:19 AM File folder ZCU102_BEL.guer_files 4/22/2022 8:10 AM File folder ZCU102_BEL.sim 4/22/2022 8:02 AM File folder ZCU102_BEL.sim 4/21/2022 8:02 AM File folder ZCU102_BEL.wes 4/22/2022 8:02 AM File folder	me Date modified Type Size ZCU102_BEL.cache 4/22/2022 8:19 AM File folder ZCU102_BEL.hbs 4/22/2022 8:19 AM File folder ZCU102_BEL.hv 4/22/2022 8:19 AM File folder ZCU102_BEL.hv 4/22/2022 8:19 AM File folder ZCU102_BEL.get.files 4/22/2022 8:09 AM File folder ZCU102_BEL.sim 4/21/2022 8:00 PM File folder ZCU102_BEL.wc 4/22/2022 8:00 AM File folder	

图 6-4. 选择平台

5. 右键点击新平台工程以打开下拉菜单。点击 Build Project 开始构建(请参阅图 6-5)。这可能需要一些时间才 能构建完成。

Vorkspace1 - Vit	is IDE	Run Xilinx Project Window Heln	
	8 -	<b>≪</b> • ! D	ā 🗷 🖸 💋
<ul> <li>Explorer X</li> <li>ZC706ps</li> <li>ZC706PS_SP</li> <li>ZCU102_1</li> <li>ZCU102ps (r</li> <li>bitstrean</li> <li>bitstrean</li> <li>bitstrean</li> <li>bitstrean</li> <li>bitstrean</li> <li>control bitstrean</li> <licontrol bitstrean<="" li=""> <li>contr</li></licontrol></ul>	I_syst	Refresh	> Ctrl+V
> 🗁 zynqmp	<	Build Project	
✓ platform >  ZCUSPI_syst	◄	Clean Project Copy referenced files into project Update Hardware Specification Team Run As Debug As	> > >

图 6-5. 构建新工程

## 7 在 Vitis 中新建应用工程

构建完成后,在 Vitis 中新建应用工程。若要新建工程,请执行以下步骤:

1. 右键点击 Platform 工程,转到 New, 然后点击 Application Project (请参阅图 7-1)。



图 7-1. 新建应用工程

2. 出现 New Application Project 窗口时(请参阅图 7-2),点击 Next。

te a New Application Pr	oject				
This wizard will guide you 1. Choose a <b>platform</b> or 2. Put application project 3. Prepare the application 4. Choose a template for	through the 4 steps of creatin create a <b>platform project</b> fror in <b>a system project</b> , associate n runtime – <b>domain</b> application to quick start deve	g new application projects. n Vivado exported XSA e it with a processor dopment			
	Platform Project	System Project			
Processor	Domain XSA	Арр			
A platform provides har     A system project contai     A domain provides runt     A workspace can contai	dware information and softwa ns one or more applications th ime for applications, such as o n unlimited platforms and unl	re environment settings, at run at the same time, perating system or BSP, imited system projects.			
	🗌 Skij	p welcome page next time. (Can b	e reached with Back button)		

#### 图 7-2. 新应用工程

### 3. 选择新创建的平台 ZCU102ps, 然后点击 Next (请参阅图 7-3)。

form from repository	Create a new pl	atform from hardware (XSA)				
ß.					🕂 Add 🔅 Manage	
[custom] 1 [custom]	Board zc706 zcu102	Flow Embedded SW Dev Embedded SW Dev	Vendor Pat xilinx C:\\ xilinx C:\\	h Jsers\imhbdlab\workspace Jsers\imhbdlab\workspace	1\ZC706ps\export\ZC706ps 1\ZCU102_1\export\ZCU102	\ZC7 2_1\Z
		Acceleration Resources		Domain Details		
ZCU102ps	^	The selected platform does not	have application	Domains		^
xczu9eg-ffvb1156-2-e	=	acceleration capabilities		Domain name	Details	
zynquplus				standaione_domain	CPO: psu_conexa55_00	
os						
	ZCU102ps       xczu9eg-ffvb1156-2-e       zynquplus	Custom     Custom       Custom     Custom       Icustom     zc706       Icustom     zcv102       scutor     zcv102       ZCU102ps     xcu9eg-ffvb1156-2-e       zynquplus     ss	Icustom     Icustom       Icustom     zcv06       Embedded SW Dev       Icustom     zcv102       Embedded SW Dev       Icustom       zcu102       Embedded SW Dev       Icustom       Icustom       Zcu102       Embedded SW Dev       Icustom       Icustom       Zcu102       Embedded SW Dev       Icustom       Icu	Icustom]     ZCU102       ZCU102ps       ZCU102ps	Iterate a new partonin noin national exception       Iterate     Iterate       Iterate	Image: Contraction reported by the contraction reported by the contraction report of the contractin report of the contraction report of the con

图 7-3. 选择应用工程

4. 键入新的应用名称。以 ZCU102ps\_SPI 为例。

,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	y xen project properties		
lication project name ( ZCU102ps_SPI stem Project			
Create a new system project for the app	lication or select an existing one from the work System project details	psace 👔	
Create new	System project name: ZCL	102ps_SPI_system	
	Target processor	Application project	
	Processor	Associated applications	
	ps7_cortexa9_0	ZCU102ps_SPI	
	Show all processors in the h	ardware specification 🗌 👔	

图 7-4. 新应用工程名称



5. 选择 standalone on microblaze\_0, 然后点击 Next (请参阅图 7-5)。

Vew Application Project				-	Х
Domain Select a domain for your project or create a new domain					••
Select the domain that the application would link to or cre Note: New domain created by this wizard will have all the	nate a new domain requirements of the application temp	plate selected in the next step			
select a domain	Domain details				 
Create new	Name:	standalone_domain			
	Display Name:	standalone on microblaze_0			
	Operating System:	standalone	~		
	Processor	microblaze_0			

图 7-5. 应用工程名称

6. 从模板列表中选择 Hello World, 然后点击 Finish (请参阅图 7-6)。

mplates		
elect a template to create your project.		
vailable Templates:		
nd: /2_	Hello World	
SW development templates	Let's say 'Hello World' in C.	
Dhrystone		
Empty Application		
Empty Application (C++)		
Hello World		
IWIP Echo Server		
IwiP ICP Perf Client		
INP ICP Perf Server		
IwiP UDP Perf Client		
Memory Tests		
Open AMP estes		
OpenAMP matrix multiplication Demo		
OpenAMP RPC Demo		
Perinheral Tests		
RSA Authentication Ann		
Zvng DRAM tests		
Zyng FSBL		
Lyndroot		
	.14	
8		

图 7-6. 选择模板

7. 一个全新的 C 工程会显示在顶部,从这里可以开始实际的应用开发。



## 8 构建应用工程

要构建应用工程,请按照以下步骤操作:

- 1. 右键点击应用名称,然后选择 Build Project (请参阅图 8-1)。
  - ✓ workspace1 ZCU102ps\_SPI/src/main.c Vitis IDE File Edit Search Run Xilinx Project Window Help 🔁 • 🔄 🕸 • 🔦 • 🕪 🗉 🔳 🐼 🗢 📭 🔯 🖾 😓 🔊 🖄 🖻 🔩 🚵 🗢 🗖 💁 Explorer 🖾 .c main.c 🖾 > 🔄 ZC706ps 27 \* Except as contained in this notice, the name of th \* in advertising or otherwise to promote the sale, u 28 > T ZC706PS\_SPI\_system \* this Software without prior written authorization 29 > G ZCU102\_1 30 \* > 🔄 ZCU102ps 31 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 32 #include <stdio.h> ZCU102ps\_SPL1> and alone on nsu\_contex. "xparameters.h" /\* EDK generated paramet 33 #include New > Binaries tiafe\_bringup.h> > 🔊 Includes Ctrl+V Paste > 🗁 Debug platform.h" 💥 Delete > 🗁 Include xil\_printf.h" 8 Refresh math.h" V 🕞 src > .c main. > 🖻 platfo 🖾 Export as Archive tiAfe79\_interface.h" > 🖸 platfo 🛛 Build Project > h platfo **Clean Project** > .c tiAfe7 latform(); intf("SPI Writing for AFE Registers!"); S Iscript Team > p\_platform(); > 🗭 \_ide 🔀 ZCU102p Run As > set(); //Reset of PS SPI > 🗁 Debug Debug As .//. > //AEC COT Coto ZCU102ps\_S > 👕 ZCUSPI\_system Properties Problems 📗 Vitis Log 🥡 Guidance TCF Debug Virtual Terminal - Cortex-A53 #3 Г

图 8-1. 构建工程

2. 确保工程构建过程不会出现任何错误。

## 9为 AFE79xx EVM 生成 SPI 日志

SPI 日志生成分为三个不同部分:

- 1. 使用 AFE79xx GUI 生成 LMK SPI 日志
- 2. 使用 AFE79xx GUI 生成 AFE SPI 日志
- 3. 将生成的 SPI 日志转换为 Vitis 中使用的格式

#### 9.1 生成 LMK SPI 日志

在 AFE79xx GUI 中,可添加以下行以启用 LMK SPI 写入的日志记录。这些应添加在 AFE.LMK.lmkConfig() 行或 AFE.deviceBringup() 之前。

```
lmklogDumpInst=mLogDump.logDump(ASTERIX_DIR+DEVICES_DIR+r"\Afe79xxPg1_LMK.txt")
lmklogDumpInst.logFormat=0x1
lmk.logClassInst = lmklogDumpInst
lmk.rawWriteLogEn=1
```

在 AFE79xxGUI 中运行该脚本后,将在以下文件夹中生成一个名为 "Afe79xxPg1\_LMK.txt" 的文件,其中包含 LMK 的寄存器写入: "C:\Users\<UserName>\Documents\Texas Instruments\Afe79xxLatte\lib"



## 9.2 生成 AFE SPI 日志

在 AFE79xx 启动脚本中,可以更新以下参数,以便启用格式1和格式5两种格式的 AFE SPI 写入日志记录。

• logDumpInst.logFormat 更改为 "0x21"

在 AFE79xxGUI 中运行该脚本后,将在以下文件夹中生成一个名为 "Afe79xxPg1Format5'.txt"的文件,其中包含 AFE 的寄存器写入: "C:\Users\<UserName>\Documents\Texas Instruments\Afe79xxLatte\lib"。

## 9.3 将 SPI 日志转换为 Vitis 的格式

在 AFE79xx Secure 文件夹中,您可以找到名为 "SPI\_Convert.py"的脚本,该脚本用于将生成的 SPI 日志转换 为正确的格式。该脚本应在 AFE79xx GUI 中打开,并在运行脚本后,会生成两个名为 "AFEspiwrites.c"和 "LMKspiwrites.c"的文件。

可以将这些 spiwrite 文件中的所有命令复制并粘贴到 Vitis 工程中的 tiafe\_bringup.h 文件中。

1. "AFEspiwrites.c" 文件的内容必须复制到 bringupafe() 函数中。下图中显示了包含相应命令的函数。

```
52 #include <stdio.h>
53 #include <stdint.h>
54 #include "xil_printf.h"
55 #include <tiAfe79 interface.h>
56
57 void bringupafe()
58 {
       int rdVal=0;
59
60
       int pollIter=0;
61
       int pollVal=0;
62
63 }
64
```

- 2. "LMKspiwrites.c" 文件的内容必须复制到 lmk\_config() 函数中。下图中显示了包含相应命令的函数。
  - 66 void lmk\_config() 67 { 68 69 70 }

在 Vitis 中更新 tiafe\_bringup.h 文件后,应该按照第8节"构建应用工程"中概述的步骤再次编译应用工程。

## 10 AFE79xxEVM 板修改

要将 AFE SPI 连接到 FPGA,必须进行以下修改:

- 1. 移除 R266、R267、R268 和 R274 (位于 EVM 底部)。
- 2. 为 R88、R87、R86 和 R83 安装 0 Ω 电阻器 (位于 EVM 顶部)。

R266 R267	49.9 49.9	SPIACLK SPIASDIO	DNI R88 0 DNI R87 0	SPIACLK_FMC	SH18 SH18
R268 R269	49.9 49.9	SPIASEN SPIB2CLK	DNI R86 0	SPIASEN_FMC	SH18 SH9
R270	49.9	SPIB2SDIO		->>> SPIB2SDIO	SH9
R271	49.9	SPIB2SEN		->>> SPIB2SEN	SH9
R272 R273 R274 R275	49.9 49.9 49.9 49.9	SPIBCLK SPIBSDIO SPIASDO SPIB2SDO	DNI R85 0 DNI R84 0 DNI R83 0	SPIBCLK_FMC SPIBSDIO_FMC SPIASDO_FMC	SH18 SH18 SH18
R276	49.9	SPIBSDO	DNI R82 0		SH9 SH18

LMK SPI 未连接到 FMC 连接器,因此为了使 FPGA 控制 LMK SPI,必须对电路板进行以下修改。

- 1. 将 LMK\_SCK 连接至 SPIACLK (从 R88 至 R248 接线)。
- 2. 将 LMK\_SDIO 连接到 SPIASDIO (从 R87 至 R249 接线)。
- 3. 将 LMK\_CS 连接至 FMC 连接器的引脚 D26。
  - a. 要执行此操作,可以在 R242 和连接到 FMC 的 R9 的开放式焊盘之间连接一根导线。



# 11 配置 AXI GPIO

AXI GPIO 配置的序列应为:

- 1. 初始化 GPIO 模块
- 2. 设置方向
- 3. 为相应的位设置高电平或低电平

初始化 AFE 和 LMK 器件之前必须完成上述序列。

## 11.1 初始化 GPIO

初始化 AXI GPIO 的 C 语法分为两个步骤:

- 1. XGpio GPO:初始化指向 GPIO 配置寄存器的指针 (GPO)。
- 2. XGpio\_Initialize(&GPOs, XPAR\_AXI\_GPIO\_0\_DEVICE\_ID): 请参阅 *Xparameters.h* 以查找正确的 AXI GPIO 器件 ID。

## 11.2 设置方向

XGpio\_SetDataDirection(&GPOs, 1, 0);

第一个实参 & GPO 指向在上一命令中初始化的 GPIO 实例。

- 第二个实参1指示 GPIO组。在此示例中, 仅使用第一个组。
- 第三个实参0指示为输出设置了所有 GPIO 位。

### 11.3 针对相应的位设置为高电平或低电平

XGpio\_DiscreteWrite(&GPOs, 1, regval);

第一个实参 & GPO 指向在上一命令中初始化的 GPIO 实例。

第二个实参1指示 GPIO组。在此示例中, 仅使用第一个组。

#### 表 11-1. IP I/O 的位映射

位描述	位位置
JESD RSTn	4
JESD TXRST	3
RSTn	2
RXTDD	1
TXTDD	0

例如:

XGpio\_DiscreteWrite(&GPOs, 1, 0x14);

此命令将 JESD RSTn 和 RSTn 设置为 1,将所有其他位设置为 0



## 12 配置 AXI SPI

在标准模式下使用 TI AFE SPI IP 中的 AXI SPI 实例。

外设 select 0 和 select 1 分别用作 AFE 和 LMK 时钟器件的芯片选择

SCL 频率在 TI IP 内硬编码为 10MHz

在 Vitis 中初始化和使用 SPI 的关键命令如下所述:

1. XSpi\_Config \*ConfigPtr;

初始化指针 (ConfigPtr)。

2. ConfigPtr = XSpi\_LookupConfig(XPAR\_AXI\_QUAD\_SPI\_0\_DEVICE\_ID);

参考 "Xparameters.h" 找到正确的 AXI QUAD SPI DEVICE ID。

3. XSpi\_CfgInitialize(&Spidev, ConfigPtr, ConfigPtr->BaseAddress);

初始化 SPI (Spidev) 的新实例。

4. XSpi\_SetOptions(&Spidev, XSP\_MASTER\_OPTION);

将 Spidev 实例设置为控制器模式。

- 5. XSpi\_Start(&Spidev);
- 6. XSpi\_SetSlaveSelect(&Spidev, 1);

选择外设:AFE。

7. XSpi\_SetSlaveSelect(&Spidev, 2);

选择外设:LMK。

8. XSpi\_Transfer(&Spidev, WrBufdev, RdBufdev, 3) ;

第二个实参 WrBufdev 是一个包含 3 个字节 (要在 SPI 上传输的 24 位数据)的数组。

第三个实参 RdBufdev 是一个包含 3 个字节的数组,最后一个字节有 SPI 读取值。

第四个实参是要发送/接收的字节数...在本例中为3。

D23 是 24 位数据的 MSB 位,因为 D23 指示它是读操作还是写操作:

- 如果 D23 设为 1,则为读操作, RdBufdev[2] 存储读回地址内容
- 如果 D23 设为 0,则为写操作,RdBufdev[2] 无关紧要

#### 13 设置硬件和为硬件加电

若要设置硬件并为其上电,请遵循以下步骤:

- 1. 将 AFE EVM 连接到 ZCU102 板上的 J5 (HPC0) FMC。
- 2. 通过时钟源将 1.5GHz 信号连接到 J14 处的 RFROM EVM。
- 3. 将 J2 (JTAG) 和 J83 (UART) USB 连接器从 ZCU102 FPGA 板连接到计算机。
- 4. 在 J52 处连接 ZCU102 的 12V Xilinx EVM 适配器。
- 5. 完成上述所有连接后,为设置上电。请注意,本示例中的 AFE EVM 完全由 ZCU102 FMC 接口供电。

#### 14 为 VADJ\_FMC 设置 ZCU102 电路板接口

若要为 VADJ\_FMC 设置 ZCU102 电路板接口,请遵循以下步骤:

- 1. 执行 ZCU102-Board User Interface 软件 (可从 Xilinx.com 下载)。
- 选择适当的 COM 端口以启用 ZCU102 的板载 MSP430 与 PC 之间的通信。为 FPGA 的 FMC 组打开 1.8V 的 FMC\_AUX 电源需要该软件(请参阅图 14-1)。

1	Clocks Voltages Power	FMC	GTR MUX	EEPROM Data	GPIO Commands	System Monitor	About
	Set Read Set Boot Frequ	ency	Restore Devi	ice Defaults			
	Read Si570 User Frequency	Freq	uency: 0.000 M	MHz			
	Read Si570 MGT Frequency	Freq	uency:				
	Read Si5328 Frequency	Freq	uency:				
			Calculation		×		
			Select the po	at that system cont	roller is on:		
			COM58		~		
			Advanced				
			Auvanceu	<u> </u>			
					OK		

图 14-1. ZCU102 板用户接口

3. 选中 Set VADJ to 1.8V 复选框 (请参阅图 14-2)。





4. 通过读取 VADJ\_FMC 电压来确认相同。电压值必须为 1.80V (请参阅图 14-3)。



图 14-3. VADJ\_FMC 电压

## 15 调试应用工程并设置 Vitis 串行终端

调试应用工程并设置 Vitis 串行终端。请按照下列步骤操作:

1. 右键点击工程名称,然后从下拉菜单中转到 Debug As。点击 Launch on Hardware (Single Application Bug) 以运行调试(请参阅图 15-1)。



图 15-1. 调试应用工程



2. 以波特率 115200 连接 Vitis 串行终端(请参阅图 15-2)(这可用于查看 SPI 写入或读取状态)。

<pre>56 delaySec(0.001); 57 lmk.config(); 58 delaySec(0.001); 59 bringupafe(); 60 xil_printf("Done!"); &lt; </pre>	, <b>v</b>	•
📴 Console 🖷 Progress 🕼 Vitis Serial Terminal 🔯 🕼 Executables 📗 Vitis Log 🖹 Problems 🙀 Debugger Console		🔶 k L = 🗆 🔳
Click on + button to add a port to the terminal.		Connect to serial port.
		A NO Sea

#### 图 15-2. Vitis 串行终端

## 16 执行应用

要执行应用运行,请完成以下步骤:

1. 点击如图 16-1 所示的右箭头按钮。



图 16-1. 执行应用

2. 请注意, UART 终端上输出了 SPI 日志:



# 17 修订历史记录

注:以前版本的页码可能与当前版本的页码不同

Changes from Revision * (November 2022) to Revision A (May 2024)	
• 更改了 <i>摘要</i> 部分	1
• 通篇更新了表格、图和交叉参考的编号格式	3
• 删除了 AFE SPI IP 容器引脚分配、TI AFE SPI IP 容器 和使用 TI AFE SPI IP 创建块设计 部分	3
• 更新了节1	3
• 从节 2 中删除了 TI 提供的 AFE SPI IP	3
• 更新了节 3	4
• 更新了节 4	5
<ul> <li>添加了节 5</li> </ul>	6
• 删除了 AFE SPI IP 容器引脚分配、TI AFE SPI IP 容器 和使用 TI AFE SPI IP 创建块设计 部分	12
<ul> <li>添加了节9</li> </ul>	19
<ul> <li>添加了节 10</li> </ul>	<mark>21</mark>
• 更新了节 11.2	
• 更新了节 11.3	
• 删除了 创建在 SD 卡上运行的引导映像一节	

#### 重要声明和免责声明

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 产品的应用。严禁对这些资源进行其他复制或展示。 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 Copyright © 2024,德州仪器 (TI) 公司