

FMC-IMAGEON VITA 直通教程

1.0 版

FMC-IMAGEON - VITA 直通教程

FMC-IMAGEON - VITA 直通教程

修訂歷史

版本說明日期

1.0 VITA 直通教程

•Vivado 2013.3 版本

我

2014 年 3 月 20 日

維目錄

修訂歷史.....	
1 目錄.....	2
表圖的.....	3
表表的.....	4
關於本指南.....	- -
介紹.....	- 2 -
要求.....	- 4 -
軟件.....	- 4 -
硬件.....	- 4 -
ZYNQ 嵌入式設計工具.....	- 4 -
賽靈思視頻 IP	- 4 -
設置.....	- 5
-提取教程存檔您的計算機上.....	- 5 -
教程概述.....	- 6 -
可复用構件.....	- 6 -
VITA 直通概述.....	- 6 -
與處理器的嵌入式設計.....	- 8 -
FMC-IMAGEON - I2C 控制器.....	- 8 -
FMC-IMAGEON -視頻接口.....	- 8 -
AXI4-流橋.....	- 8 -
實現一個 VITA 直通.....	- 10 -
授權的視頻和圖像處理包 IP 核。	- 10 -
創建一個新的 Vivado 項目.....	- 10
-創建 IP 集成嵌入式硬件設計.....	- 11 -
添加了 I2C 控制器.....	- 13 -
添加視頻管線.....	- 16 -
構建硬件與 Vivado 設計套件.....	- 21
-創建 SDK 中的嵌入式軟件應用.....	- 24 -
設置您的 ZC702 硬件.....	- 30 -

設置您的 ZedBoard 硬件..... - 31 -

使用 EXECUTE 硬件上 VITA 直通設計 SDK - 31 -

參考..... - 33 -

FMC-IMAGEON - VITA 直通教程

已知問題和限制。	- 34 -
Hello World 模板-錯誤：衝突的類型'打印'	- 34 -
故障排除.....	- 35 -

圖 1 表

圖-在具有 HDMI 輸入/輸出 FMC 捆綁.....半導體圖像傳感器-	2 -
圖 2 - FMC-IMAGEON 硬件-連接圖.....	- 3 -
圖 3 - FMC -IMAGEON 硬件-框圖.....	- 3 -
圖 4 -VITA 直通-顏色傳感器.....	- 7 -
圖 5 -VITA 直通-單色傳感器.....	- 7 -
圖 6 -創建塊設計.....	- 12 -
圖 7 -設計師援助-添加 IP。	- 12 -
圖 8 -設計師援助-運行座自動化.....	- 12 -
圖 9 -Run 座自動化.....	- 13 -
圖 10 - AXI IIC -塊屬性.....	- 13 -
圖 11 -運行連接自動化- / fmc_imageon_iic_0 / S_AXI	- 14 -
圖 12 -運行連接自動化- / fmc_imageon_iic_0 / IIC	- 14 -
圖 13 -外部接口屬性- fmc_imageon_iic	- 15 -
圖 14 -外部接口屬性- fmc_imageon_iic_rst_n	- 15 -
圖 15 -驗證設計.....	- 16 -
圖 16 - IP 資源庫管理器.....	- 17 -
圖 17 - VITA 接收器/ HDMI 輸出子模塊.....	- 18 -
圖 18 -驗證設計.....	- 21 -
圖 19 -創建 HDL 封裝.....	- 22 -
圖 20 -讓 Vivado 管理和包裝自動更新.....	- 22 -
圖 21 - VITA 直通-資源利用.....	- 23 -
圖 22 - SDK 中的 system.xml 標籤地址地圖.....	- 24 -
圖 23 -板級支持包設置..	- 25 -
圖 24 -生成一個鏈接描述.....	- 26 -
圖 25 -從文件系統導入-對話框 1	- 27 -
圖 26 -導入放入文件夾.....	- 28 -

圖 27 -從文件系統導入-對話框 2 - 29 日 -

表的表

表 1-新的項目設置..... - 11 -

關於本指南

-本教程介紹如何從頭開始創建 Vivado IP 集成視頻設計

這手冊包含以下章節

：•第“簡介”介紹提供的概述和 FMC-的

本教程所需 IMAGEON FMC 模塊功能，以及硬件和軟件。

•章節“教程概述”提供本教程的概述。•

章“實現一個 VITA 直通”描述了所需的步驟

一個嵌入式系統，實現了 VITA 直通。•

附錄“參考”提供了有關 FMC 文件引用的列表

模塊。•

附錄“已知問題和限制”提供的已知問題和列表

限制，在本教程中使用的工具和/或 IP”

•附錄故障排除“提供的本的故障排除建議列表

教程-

介紹

安森美半導體圖像傳感器 FMC 包提供的幾個高清視頻
了啟用 FMC-賽靈思踢腳板接口。該 FMC 模塊具有板載 HDMI
輸入/輸出接口。安森美半導體 VITA-2000 彩色圖像傳感器模塊提供了
高清晰度攝像頭支持高幀速率和具有全局快門。
這 FMC 捆綁是理想的機器視覺，運動監控，開發應用
以及高端安全和監視。

圖 1 -開帶 HDMI 輸入/輸出 FMC 捆綁半導體圖像傳感器

如圖 2 和圖 3 所示，FMC 模塊連接到 FMC 運營商，並提供
以下接口

：•HDMI 輸入

•VITA HDMI 輸出

圖像傳感器•LCEDI 接口模塊

下面的框圖說明了如何在 FMC 模塊的連接



12V

FMC LPC (72 I / O , 36 差分)

圖 3 - FMC-IMAGEON 硬件-框圖

FMC-IMAGEON - VITA 直通教程

要求

本教程的軟件和硬件要求在下面的章節中描述。

軟件

構建和運行演示所需的軟件是

- 賽靈思 Vivado

2013.3• 仿真終端 (HyperTerminal 或 TeraTerm)

硬件

來運行該參考設計所需的最低限度是

- 電腦，最小 4 GB 的完成設計

1

- 以下 ZYNQ 運營商 (FMC-IMAGEON) • HDMI (

ØZC702 (包括電源和電纜

) OZedBoard (包括電源和電纜

) • 安富利 HDMI 輸入/輸出 FMC 模塊

或者 DVI-D) 顯示器，包括 HDMI 電纜

- VITA-2000 攝像頭模塊，包括 LCEDI 電纜

ZYNQ 嵌入式設計工具

本教程假設你有經驗建立 ZYNQ 嵌入式設計。更具體地說，它假定的 Vivado 工具，包括 IPI (Vivado IP 集成器) 和 SDK 的應用知識。

如果你沒有這方面的經驗，我們強烈建議您按照介紹 ZYNQ 在線培訓課程，可在 zedboard.org 等

- <http://www.zedboard.org/course/introduction-zynq>

賽靈思視頻 IP

本教程將利用幾個 Xilinx 的視頻 IP 核

- 1

參見 <http://www.xilinx.com/design-tools/vivado/memory.htm>

FMC-IMAGEON - VITA 直通教程

雖然可以完成本教程沒有諮詢的數據表，這些視頻 IP 內核，強烈建議諮詢每個視頻 IP 核心，以數據表充分理解和欣賞他們的潛力。

視頻定時控制器

<http://www.xilinx.com/products/intellectual-property/EF-DI-VID-TIMING.htm>

視頻輸入到 AXI4-流

http://www.xilinx.com/products/intellectual-property/video_in_to_axi4_stream.htm

AXI4-流至視頻輸出

http://www.xilinx.com/products/intellectual-property/axi4_stream_to_video_out.htm

AXI 視頻 DMA

http://www.xilinx.com/products/知識產權/axi_video_dma.htm 特別感興趣的是“三重幀緩衝示例”一章

對於 Xilinx 視頻 IP 內核的完整列表，請參閱以下網頁

: http://www.xilinx.com/ipcenter/video/video_core_listing.htm

安裝程序

開始之前本教程，請仔細閱讀下面的章節將介紹

如何提取教程存檔以及如何測試您的視頻設備：

提取您的計算機上的教程存檔

提取存檔教程在 C 盤根 \ 駕駛。它將包含以下目錄

C : \ FMC_IMAGEON \ 2013_3 \ avnet_fmc_imageon_cores C : \ FMC_IMAGEON \ 2013_3 \ 限制 C : \ FMC_IMAGEON \ 2013_3 \ 代碼 C : \ FMC_IMAGEON \ 2013_3 \ 腳本

教程概述

本教程將指導您創建一個視頻的設計，實現了 VITA 直通為安富利 HDMI 輸入/輸出 FMC 模塊：•IP

可重複使用的組件

本教程將利用可重用組件的視頻輸入和視頻輸出接口

內核，每個 FMC-IMAGEON 模塊上的視頻接口

ØFMC-IMAGEON- HDMI 輸入

◆FMC-IMAGEON - HDMI 輸出

ØFMC-IMAGEON - VITA 接收器

•TCL 腳本：自動生成 IP 集成子模塊

對於單色圖像傳感器

：øfmc_imageon_vita_mono.tcl：視頻輸入通道，8 位 y 格式

Øfmc_imageon_hdmio_y.tcl：視頻輸出通道，8 位 y 格式

對於彩色圖像傳感器

：øfmc_imageon_vita_color.tcl：視頻輸入通道，24 位 RGB 格式

Øfmc_imageon_hdmio_rgb。TCL：視頻輸出通道，24 位 RGB 格式

•XDC 約束：定義引腳和約束各種載體

Øzc702_fmc_imageon_vita_passthrough.xdc：為 ZC702 約束

Øzedboard_fmc_imageon_vita_passthrough.xdc：為的限制

ZedBoard•C 源代碼：例如提供初始化代碼

Øfmc_imageon_vita_passthrough.c / .h：為 VITA 初始化代碼直通

有關這些可重用的組件的更多信息，請參閱 FMC-IMAGEON - 核

IPVivado 文檔。

VITA 直通

概述用戶可以選擇實施視頻直通兩種視頻格式 6

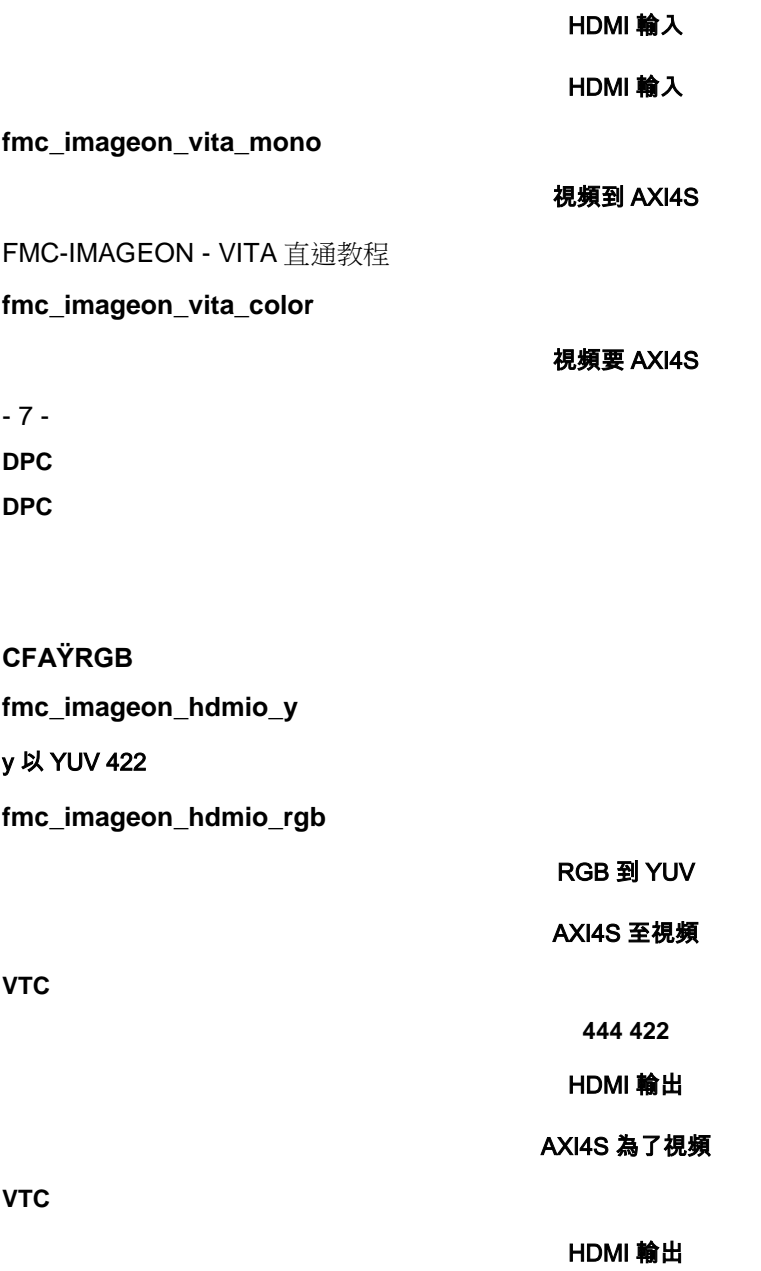
- -

一個直通可以用於彩色圖像傳感器-

圖 4 -VITA 直通顏色傳感器

的備用直通可以用於單色圖像傳感器。單色傳感器將只擁有 Y（亮度或）組件，這是強度。所述 AXI4 流協議轉換器核心可用於交錯的 UV 組件（或色度）設置為 0x80，有效地創造在 16 位顏色更少的像素的 YUV 4：2：2 視頻格式。

圖 5 -VITA 直通- 單色傳感器



- 7 -

DPC

DPC

CFA²RGB

`fmc_imageon_hdmio_y`

y 以 YUV 422

`fmc_imageon_hdmio_rgb`

RGB 到 YUV

AXI4S 至視頻

VTC

444 422

HDMI 輸出

AXI4S 為了視頻

VTC

HDMI 輸出

FMC-IMAGEON - VITA 直通教程

與處理器嵌入式設計

，設計要求的嵌入式設計，包括處理器，以及以下外設

：•外部存儲器

•串口（UART 或 USB-UART）

本教程將創建一個基於 ZYNQ 嵌入式設計。

此外，對於非 ZYNQ 為基礎的載體，一個基於的 MicroBlaze 嵌入式設計也可以被創建。然而，這不是本教程中描述，並作為練習留給用戶：•AXI

FMC-IMAGEON - I2C 控制器

一旦創建了嵌入式處理器設計，一個 I2C 控制器實現具有以下賽靈思 IP 核

I2C 控制器

Ø 在前面的框圖沒有顯示，這個核心將允許處理器

來配置 FMC-IMAGEON 的硬件外設，包括

：•ADV7611：HDMI 輸入設備

•ADV7511：HDMI 輸出設備

•CDCE913：視頻時鐘合成器

FMC-IMAGEON -視頻接口

的以下磁芯，設置有教程，將用於接口到 VITA-2000 的圖像傳感器，以及所述的 FMC-IMAGEON 模塊上的 ADV7511 設備。需要注意的是，在 FMC-IMAGEON 模塊上，該 ADV7511 器件在 16 位使用，這是非常重要的 YCbCr 4：2：2“的嵌入式同步”模式

•FMC-IMAGEON VITA 接收機

○ 此核心包含將德邏輯序列從 VITA-2000 的原始像素
圖像傳感器。

的 FMC-IMAGEON HDMI 輸出

○ 該芯包含將嵌入中的 16 位的中的同步信號邏輯

在發送給 ADV7511 HDMI 輸出裝置視頻數據的。

的 VITA 接收機和 HDMI 輸出 IP 芯使一個通用的並行視頻接口給
提供設計

AXI4 插播橋樑

下面的芯中使用的 VITA 和 HDMI 接口和 AXI4-流協議之間進行橋接-

•視頻定時控制器

的 FMC-IMAGEON - VITA 密碼通過教程

Ø 這個核心能夠

- 檢測視頻定時上的視頻輸入接口

- (重新) 生成一個視頻輸出接口 OA 單 VTC 核心視頻定時將用於

- 視頻輸入的視頻定時將被檢測由的 VTC 芯的檢測器部分

- 該局芯的發電部將被同步到檢測器，從而重新生成對輸出同一視頻定時

- 視頻輸入到 AXI4 插播

○ 該核心變換的通用並行視頻接口（即。DVI / HDMI）到 AXI4-

流視頻協議 Ø 核心包括 FIFO 使 AXI4-流視頻接口在運行

不同時鐘

- AXI4-流視頻輸出

Ø 這個核心產生的通用並行視頻接口（即從一個 AXI4-DVI / HDMI）

流視頻接口 Ø 核心包括 FIFO 使 AXI4-流視頻接口上的運行

不同時鐘

為了說明 AXI4 流接口的背壓能力，該視頻流水線將兩個單獨的時鐘域來實現。輸入和輸出接口將在 VITA 接口的視頻時鐘運行。該 AXI4 流接口在一個單獨的時鐘運行 9

-

實現一個 VITA 直通

-在本節中，一個新的 Vivado 項目將被創建，實施安富利 FMC-IMAGEON 模塊 VITA 直通設計

許可視頻和圖像處理包 IP 核

本設計採用幾個賽靈思視頻和圖像處理包的 IP 核，必須在使用前行貨。請按照以下步驟請求評估許可證

：1。轉到

： www.xilinx.com/products/intellectual-property/EF-DI-VID-IMG-IP-PACK

2。點擊位於網頁左上角的評估環節，

並按照聯機說明

3。生成的許可文件通過電子郵件發送。按照隨附的說明添加評估

了視頻和圖像處理包許可證的功能的

創建一個新 Vivado 項目

要創建一個新的項目，啟動 Vivado™ 設計和分析軟件和嵌入式處理器系統的頂層創建一個項目。

1。啟動 Vivado 2013.3 軟件。 2。選擇創建新項目打開新建項目嚮導 3。使用下表中的信息，使您的選擇在嚮導屏幕

嚮導屏幕系統屬性設置或命令來使用項目名稱項目名稱指定項目名稱，如：

教程項目的位置指定要在其中存儲項目文件的目錄

： FMC-IMAGEON- VITA 直通教程

- 10 -

C:\FMC_IMAGEON\2013_3 創建項目的子目錄離開這個檢查。項目類型指定項目的類型來

創建

使用默認選項，指定 RTL 項目

添加源不使這個屏幕上的任何變化。添加現有 IP 不要讓這個屏幕上的任何變化。添加約束，不要讓這個屏幕上的任何變化。默認部分指定選擇板

過濾器在家庭列表中，選擇 ZYNQ-7000 主板列表中選擇以下主板之一

: ZYNQ-7ZC702 評估板或

ZedBoard ZYNQ 評估和開發套件新建項目總結項目總結單

擊 Finish 之前，請查看項目總結

以創建項目 1

表 -新項目設置

當您單擊完成，新建項目嚮導將關閉，您剛剛創建的項目 Vivado 打開。

創建 IP 集成嵌入式硬件設計

本節將指導您如何創建一個基本的嵌入式硬件設計。這部分

可能會略有不同的運營商不同。

創建一個新的 Vivado IP 集成器的框圖

1. 點擊 IP 集成流程導航創建塊設計

，打開創建塊設計對話框。2.指定框圖設計名稱，然後單擊確定。3.單擊下一步：

FMC-IMAGEON - VITA 直通教程

圖 6 -創建塊設計

注意 Vivado IP 集成商提供的設計輔助

圖 7 -設計師援助-添加 IP

的 ZYNQ7 處理系統的核心加入塊設計

4。點擊的設計援助添加

塊設計或右鍵單擊 IP，然後選擇添加 IP。5.選擇 ZYNQ7 處理系統的核心，然後按 ENTER 鍵確認（或雙擊選擇）

注意 Vivado IP 集成器提供了額外的設計幫助：

圖 8 -設計師援助-運行座自動化

配置使用設計援助 ZYNQ7 處理系統的核心

：6。單擊運行自動化座在設計上援助 7.選擇/ processing_system7_0 8.確保應用董事會預設選項被選中。
9.單擊 OK

FMC-IMAGEON - VITA 直通教程

圖 9 -Run 座自動

添加 I2C 控制器

的 AXI IIC IP 核添加到設計，將用於向 FMC 模塊上配置 I2C 外設

1。在塊設計的空白部分單擊鼠標右鍵，然後選擇添加 IP。2.選擇 AXI IIC 核心，然後點擊 Enter 鍵（或雙擊選擇）

重命名 IIC 控制器

fmc_imageon_iic_0。3。選擇框圖中的 AXI IIC 核心。4.在塊屬性對話框，

修改名稱塊到 fmc_imageon_iic_0

圖 10 - AXI IIC -塊屬性

FMC-IMAGEON - VITA 直通教程

-使用設計師援助 AXI IIC 核心連接到 ZYNQ7 處理系統的 GP0 端口

。 5. 單擊運行連接自動化設計援助 6.選擇/ fmc_imageon_iic_0 / S_AXI 7.單擊 OK-

圖 11 運行連接自動化- / fmc_imageon_iic_0 / S_AXI

使用設計協助連接 AXI IIC 核心的外部 I / O

8。單擊運行連接自動化設計援助 9.選擇/ fmc_imageon_iic_0 / IIC 10. [如果出現選擇板接口下拉列表中，指定自定義] 11.單擊 OK-

圖 12 運行連接自動化- / fmc_imageon_iic_0 / IIC

FMC-IMAGEON - VITA 直通教程

重命名外部 iic_rtl 端口 fmc_imageon_iic

12。選擇在框圖 iic_rtl 外部端口。 13.在外部接口屬性對話框，修改塊的名稱 fmc_imageon_iic

圖 13 -外部接口屬性- fmc_imageon_iic

的 AXI IIC 核心的 GPO [0 : 0]連接端口，外部端口

14。選擇 GPO [0 : 0]在 AXI IIC 核心 15 端口單擊鼠標右鍵，然後選擇無需外部重命名外部 GPO [0 : 0]端口 fmc_imageon_iic_rst_n

16。選擇 GPO [0 : 0]框圖外部端口。 17.在外部接口屬性對話框，修改塊的名稱 fmc_imageon_iic_rst_n

圖 14 -外部接口屬性- fmc_imageon_iic_rst_n

在這一點上，驗證你有一個正確的設計

18。在塊設計的空白部分單擊鼠標右鍵，然後選擇驗證設計

FMC-IMAGEON - VITA 直通教程

19。如果成功的話，點擊 OK，下一個章節進行。否則，繼續之前解決設計中的任何錯誤， -

圖 15 驗證設計

添加視頻管線

視頻管線將利用安富利提供的 IP 核。為了 Vivado 設計套件能夠識別這些內核，我們需要將安富利 FMC-IMAGEON 核心的位置添加到庫路徑

- 1。在 Vivado 菜單中選擇工具=>項目設置，打開項目設置對話框。
- 2.點擊左側的 IP 圖標。
- 3.在資源庫管理器選項卡，單擊添加存儲庫按鈕。
- 4.指定“C:\FMC_IMAGEON\2013_3\avnet_fmc_imageon_cores”目錄，然後單擊選擇按鈕

下面的三個 IP 內核應檢測，並在選定的儲備庫出現在

窗口 IP：•FMC-IMAGEON

- HDMI 輸入

•FMC- IMAGEON - HDMI 輸出

•FMC-IMAGEON - VITA 接收

FMC-IMAGEON - VITA 直通教程

- 16 -

圖 16 - IP 資源庫管理器

5。單擊確定：•fmc_imageon_vita_color.tcl：

- 17 -

維塔接收器和 HDMI 輸出接口將與 TCL 腳本來創建•fmc_imageon_hdmio_rgb。

如果你有一個彩色圖像傳感器，運行以下兩個 Tcl 腳本

視頻輸入通道，24 位 RGB 格式

.tcl：視頻輸出通道，24 位 RGB 格式，

如果你有一個單色圖像感應器，運行以下兩個 Tcl 腳本

：•fmc_imageon_vita_mono.tcl：視頻輸入通道，8 位原始 Y（亮度）格式

•fmc_imageon_hdmio_y.tcl：視頻輸出路徑，高 8 位 Y（亮度）格式

的使用 TCL 控制台來運行你選擇兩個 TCL 腳本。這個教程將創建硬件設計用於彩色圖像傳感器，使得使用 24 位 RGB 視頻格式，但教程說明也可以應用於創建用於單色圖像傳感器的設計

6。在 TCL 控制台，鍵入以下命令來生成視頻輸入路徑

源 C：/FMC_IMAGEON/2013_3/scripts/fmc_imageon_vita_color.tcl 然後按{ENTER} 7.在 TCL 控制台，按 {向上}重複前一個命令

，編輯命令生成的視頻輸出路徑。來源 C：/FMC_IMAGEON/2013_3/scripts/fmc_imageon_hdmio_rgb.tcl 然後按{ENTER}

這將創建一個在您框圖兩個新的視頻子模塊。

圖 17 - VITA 接收器/ HDMI 輸出子模塊

的 AXI4-精簡版時鐘連接在 ZYNQ7 處理系統的 FCLK_CLK0 端口

1。選擇 fmc_imageon_vita_color 子模塊的 axi4lite_clk 端口 2.單擊並按住鼠標左鍵，然後拖動到 ZYNQ7 的 FCLK_CLK0 端口。 3.鬆開鼠標按鈕進行連接

FMC-IMAGEON - VITA 直通教程

FMC-IMAGEON - VITA 直通教程

4。選擇 fmc_imageon_hdmio_rgb 子模塊的 axi4lite_CLK 端口 5.單擊並按住鼠標左鍵，然後拖動到 ZYNQ7 的 FCLK_CLK0 端口。 6.鬆開鼠標按鈕，使

連接將 AXI4-精簡版復位到 proc_sys_reset 核心的 peripheral_aresetn 端口

。 7。選擇 fmc_imageon_vita_color 子模塊的 axi4lite_aresetn 端口 8.單擊並按住鼠標左鍵，然後拖動到 proc_sys_reset 核心的

peripheral_aresetn [0 : 0]端口。 9.鬆開鼠標按鈕進行連接

10。選擇 fmc_imageon_hdmio_rgb 子模塊的 axi4lite_aresetn 端口 11.單擊並按住鼠標左鍵，然後拖動到 proc_sys_reset 核心的

peripheral_aresetn [0 : 0]端口。 12.鬆開鼠標按鈕，使連接

使用設計師協助連接 VITA 核心的控制端口 (vita_ctrl) 與 VTC 核心的控制端口 (vtc_ctrl) 到 ZYNQ7 處理系統的 GP0 端口

13。單擊設計協助 14.選擇/ fmc_imageon_vita_color / vita_ctrl 15.單擊確定運行連接自動化

。 16。單擊設計協助 17.選擇/ fmc_imageon_hdmio_rgb / vtc_ctrl 18.單擊確定運行連接自動化。

使用設計援助 DPC 核心的控制端口 (dpc_ctrl) 和 CFA 核心的控制端口 (cfa_ctrl) 連接到 ZYNQ7 處理系統的 GP0 端口

19。單擊設計協助 20.選擇/ fmc_imageon_vita_color / dpc_ctrl 21.單擊確定運行連接自動化

。 22。單擊運行連接自動化設計援助 23.選擇/ fmc_imageon_vita_color / cfa_ctrl 24.單擊確定。

配置 ZYNQ7 處理系統產生的 FCLK_CLK1 端口，將被用於 AXI4-基於流的互連和核心上 150MHz 的時鐘。此外，生成其 FCLK_CLK2 端口，這將是由 VITA 接收機核心其 IDELAY 基元被用於上的 200MHz 的時鐘

25。在 ZYNQ7 處理系統的核心雙擊。 26.單擊時鐘產生塊 27 上展開 PL 面料時鐘部分 28 選擇 FCLK_CLK1 時鐘 29.指定為 150MHz 的指定的頻率。 30.選擇 FCLK_CLK2 時鐘 31 指定為 200MHz 的指定的頻率。 32.單擊 OK (確定) 19

FMC-IMAGEON - VITA 直通教程

連接 200MHz 的時鐘到 ZYNQ7 處理系統的 FCLK_CLK2 端口

33。選擇 fmc_imageon_vita_color 子模塊的 clk200 端口 34.單擊並按住鼠標左鍵，然後拖動到 ZYNQ7 的 FCLK_CLK2 端口。 35.鬆開鼠標按鈕來進行連接。

該 AXI4-流時鐘連接到 ZYNQ7 處理系統的 FCLK_CLK1 端口

36。選擇 fmc_imageon_vita_color 子模塊的 axi4s_clk 端口 37.單擊並按住鼠標左鍵，然後拖動到 ZYNQ7 的 FCLK_CLK1 端口。 38.鬆開鼠標按鈕來進行連接

。39。選擇 fmc_imageon_hdmi_rgb 子模塊的 axi4s_clk 端口 40.單擊並按住鼠標左鍵，然後拖動到 ZYNQ7 的 FCLK_CLK1 端口。 41.鬆開鼠標按鈕來進行連接。

在框圖 VITA 輸入時鐘創建一個外部端口

42。選擇 fmc_imageon_vita_color 子模塊的 vita_clk 端口 43.單擊鼠標右鍵，然後選擇使外部連接 HDMI 輸出時鐘的 VITA 輸入時鐘

。44。選擇 fmc_imageon_hdmi_rgb 子模塊的 hdmi_clk 端口 45 單擊並按住鼠標左鍵，然後拖動到 vita_clk 網。 46.鬆開鼠標按鈕進行連接

。連接 VITA 和 HDMI I / O 端口 (vita_io & hdmi_io) 到外部端口

47。選擇 fmc_imageon_vita_color 子模塊的 vita_io 端口 48.單擊鼠標右鍵，然後選擇撥打外線

49。Select the fmc_imageon_hdmi_rgb sub-module's hdmi_io port 50. Right-click, then select Make External

Connect the VITA input and HDMI output sub-modules together, implementing a video pass-through.

51. Select the fmc_imageon_vita_color sub-module's vita_axi4s_video port 52. Click and hold the left mouse button,

then drag to the fmc_imageon_hdmi_rgb sub-module's hdmi_axi4s_video 53. Release the mouse button to make the connection.

54. Select the fmc_imageon_vita_color sub-module's vita_vtiming port 55. Click and hold the left mouse button,

then drag to the fmc_imageon_hdmi_rgb sub-module's video_vtiming 56. Release the mouse button to make the connection.

At this point, validate that you have a correct design.

57. Right-click in a blank portion of the block design, then select Validate Design

58. If successful, Click OK and proceed with the next section. Otherwise, fix any errors in the design before proceeding.

Figure 18 – Validate Design

Save the block diagram

59. In the menu, select File => Save Block Design

Build the hardware with Vivado Design Suite

Create the top level HDL file.

1. In the Sources tab, select the block diagram 2. Right-click, then select Create HDL Wrapper

FMC-IMAGEON – VITA Pass-Through Tutorial

- 21 -

Figure 19 – Create HDL Wrapper

3. When asked whether to add or copy the HDL wrapper,
select Let Vivado manage wrapper and auto-update.

Figure 20 – Let Vivado manage wrapper and auto-update

Add the XDC constraints file.

FMC-IMAGEON – VITA Pass-Through Tutorial

4. In the Flow Navigator, under the Project Manager section, click Add Sources. 5. Select the Add or Create Constraints option 6. Click Next 7. In the dialog box that opens, click the Add Files ... button to add an existing XDC file 8. Select one of the following XDC files, depending on your hardware:

a. For the ZC702:

..\constraints\zc702_fmc_imageon_vita_passthrough.xdc b. For the ZedBoard:

..\constraints\zedboard_fmc_imageon_vita_passthrough.xdc Once selected, click OK 9. Click Finish.

Build the bitstream.

10. In the Flow Navigator, under the Program and Debug section, click Generate Bitstream.

A dialog box appears asking whether all the processes starting for synthesis should be done. 11. Click Yes.

The “Bitstream Generation Completed” dialog box will open, asking what to do Next.

12. Select Open Implemented Design 13. Click OK.

The resource utilization and power estimation for this design can be seen in the Project Summary. Note that results may be different depending on the video format chosen, and for different carriers.

Figure 21 – VITA Pass-Through – Resource Utilization

You have successfully created the hardware design !

FMC-IMAGEON – VITA Pass-Through Tutorial

Create the Embedded Software Application with SDK

Launch SDK from Vivado Design Suite.

1. In the Vivado menu, Select File > Export > Export Hardware for SDK.

The “Export Hardware for SDK” dialog box opens. By default, the “Include Bitstream” and “Export Hardware” check boxes are checked. 2. Check the Launch SDK check box. 3. Click OK. SDK opens.

Notice that when SDK launched, the hardware description file was automatically read in. The system.xml tab shows the address map for the entire Processing System.

Figure 22 – Address Map in SDK system.xml Tab

The avnet_fmc_imageon_cores directory contains some software drivers that we will use in this design. In order for the project to recognize the contents of this directory, the path must be added to the project repositories, as described below.

4. In the SDK menu, select Xilinx Tools => Repositories

The Preferences dialog box opens.

FMC-IMAGEON – VITA Pass-Through Tutorial

5. In the Local Repositories section, click on the New ... button. 6. Select the C:\FMC_IMAGEON\2013_3\avnet_fmc_imageon_cores directory, then click

OK 7. Click OK in the Preferences dialog box.

Create a standalone BSP (board support package).

8. In the SDK menu, select File => New => Board Support Package.

The New Board Support Package Project dialog box opens. 9. In the Project name field, type "vita_passthrough_bsp". 10. Keep the default settings, and click Finish.

The Board Support Package Settings dialog box opens. 11. In the Supported Libraries, select the fmc_iic_sw and fmc_imageon_sw libraries.

Figure 23 – Board Support Package Settings

FMC-IMAGEON – VITA Pass-Through Tutorial

- 25 -

12. Click OK.

If the Build Automatically setting is enabled, SDK will automatically build the standalone BSP.

Create a new C project.

13. In the SDK menu, select File => New => Application Project.

The Application Project dialog box opens. 14. In the Project Name field, type "vita_passthrough_app".

15. For the Board Support Package, select Use Existing, then select the BSP that was created previously. 16. Click Next.

The Templates dialog box opens. 17. Select the Hello World template. 18. Click Finish.

Configure the application's memory map to execute from external memory.

19. Right-click on the vita_passthrough_app application 20. Select Generate Linker Script

This opens the Generate a linker script dialog box.

21. Select the ps7_ddr_0 memory for each of the Code, Data, Heap and Stack sections.

Figure 24 – Generate a Linker Script

FMC-IMAGEON – VITA Pass-Through Tutorial

- 26 -

22. Click Generate.

A dialog box appears asking whether it is OK to overwrite the existing linker script file. 23. Click Yes
Import the provided example source files for the vita_passthrough

1. In the Project Explorer window, select the vita_passthrough_app application 2. Right-click, then select Import from the pop-up menu.

The Import wizard appears. 3. Expand the General section 4. Select File System, then click Next.

Figure 25 – Import from File System – Dialog 1

The next dialog of the Import wizard appears. 5. Next to the From directory field, click the Browse button 6. Specify the following directory:

C:\FMC_IMAGEON\2013_3\code\fmc_imageon_vita_passthrough 7. Click OK

8. Select the following source files:

fmc_imageon_vita_passthrough.c fmc_imageon_vita_passthrough.h sleep.c sleep.h video_resolution.c

FMC-IMAGEON – VITA Pass-Through Tutorial

video_resolution.h video_generation.c video_generation.h 9. Next to the Into directory field, click the Browse button 10. Specify the following directory : vita_passthrough_app\src, then click OK

Figure 26 – Import into Folder

FMC-IMAGEON – VITA Pass-Through Tutorial

Figure 27 – Import from File System – Dialog 2

11. Click Finish.

Modify the hello world application

12. Open the helloworld.c file and edit the source code as follows:

```
/*  
* helloworld.c: simple test application */  
#include <stdio.h> #include "platform.h"  
#include "fmc_imageon_vita_passthrough.h" fmc_imageon_vita_passthrough_t demo;  
//void print(char *str); void print( const char *str);  
int main() {  
    init_platform();
```

FMC-IMAGEON – VITA Pass-Through Tutorial

strange bug: when vtc driver is active, need to modify print declaration to match the one in xil_printf.h

FMC-IMAGEON – VITA Pass-Through Tutorial

```
print("Hello World\n\r");  
  
demo.uBaseAddr_IIC_FmcImageon = XPAR_FMC_IMAGEON_IIC_0_BASEADDR;  
demo.uBaseAddr_VITA_Receiver =  
  
    XPAR_FMC_IMAGEON_VITA_COLOR_FMC_IMAGEON_VITA_RECEIVER_0_S00_AXI_BASEADDR;  
demo.uDeviceId_VTC_HdmioGenerator =  
  
    XPAR_FMC_IMAGEON_HDMIO_RGB_V_TC_0_DEVICE_ID; demo.uBaseAddr_DPC =  
XPAR_DPC_0_BASEADDR; demo.uBaseAddr_CFA = XPAR_CFA_0_BASEADDR;  
fmc_imageon_vita_passthrough_init( &demo );  
  
cleanup_platform();  
  
    return 0; }
```

13. If the Build Automatically setting is enabled, SDK will automatically build the application. If not, right-click on the application and select Build Project to build the application.

You have successfully created the software application !

Set up your ZC702 Hardware

Setup your ZC7020-based hardware, as described below.

1. Set the ZC702 board's boot mode to cascaded JTAG using jumpers
 - a. J21,J20,J22,J25,J26 should all be set to '0' b. J27,J28 should be set to '1'
2. Connect a mini USB cable to the ZC702's USB-UART connector (J17) 3. Connect one of the following JTAG connections:
 - a. Connect platform USB pod to the ZC702's JTAG header (J2)
and set SW10 to '10' b. Connect a micro USB cable to the ZC702's on-board Digilent JTAG module
and set SW10 to '01' 4. Populate the FMC-IMAGEON board on FMC Slot #2. 5. Connect the VITA-2000 camera to the FMC module's CON200 connector 6. Connect a DVI or HDMI monitor to the FMC module's HDMI OUT connector 7. Power on the ZC702 board

8. Open a serial communication utility for the COM port assigned on your system.

Note: The standard configuration for Zynq Processing System is baud rate 115200, 8 bit, parity

Set up your ZedBoard Hardware

Setup your ZedBoard hardware, as described below.

1. Set the ZedBoard's boot mode to cascaded JTAG using jumpers a. JP7, JP8, JP9, JP10, JP11 should all be set to '0'
2. Connect a micro USB cable to the ZedBoard's USB-UART connector (J14) 3. Connect one of the following JTAG connections:
 - a. Connect platform USB pod to the ZedBoard's JTAG header (J15) b. Connect a micro USB cable to the ZedBoard's on-board Digilent JTAG connector (J17) 4. Populate the FMC-IMAGEON board on FMC Slot #1. 5. Connect the VITA-2000 camera to the FMC module's CON200 connector 6. Connect a DVI or HDMI monitor to the fMC module's HDMI OUT connector 7. Power on the ZedBoard board 8. Open a serial communication utility for the COM port assigned on your system.

Note: The standard configuration for Zynq Processing System is baud rate 115200, 8 bit, parity

Execute the VITA Pass-Through Design on Hardware using SDK

From SDK, configure the FPGA bitstream and launch the application.

1. In the SDK menu, select Xilinx Tools => Program FPGA

The "Program FPGA" dialog opens. 2. Make sure the path to the bitstream is valid

(HINT : If you moved the project, you will need to update the path to the bitstream file) 3. Click Program.

It will take approximately 10 seconds to program the bitstream to hardware 4. Right-click vita_passthrough_app

and select Run as > Run Configurations 5. Click Xilinx C/C++ ELF and click New launch configurations.

FMC-IMAGEON – VITA Pass-Through Tutorial

FMC-IMAGEON – VITA Pass-Through Tutorial

6. The new run configuration is created named vita_passthrough_app Debug.

The configurations associated with application are pre-populated in the main tab of these launch configurations. 7. Click the Device Initialization tab in the launch configurations and check the settings

here. Notice that there is a configuration path to the initialization TCL file. The path of ps7_init.tcl is mentioned here. This is file that was exported when you exported your design to SDK; it contains the initialization information for the processing system. (HINT : If you moved the project, you should delete the previous run configuration and create a new one) 8. The STDIO Connection tab is available in the launch configurations settings. You can use

this to have your STDIO connected to the console. We will not use this now because we have already launched a serial communication utility. There are more options in launch configurations but we will focus on them later. 9. Click Apply and then Run. 10. If you get a Reset Status dialog box indicating that the current launch will reset the entire

system, click OK. 11. You should see something similar to the following on your serial console:

Hello World

----- -- FMC-IMAGEON VITA Pass-Through -- -----

FMC-IMAGEON Initialization ... Video Clock Synthesizer Configuration ... ADV7511 Video Output Information

Video Output = DVI, Progressive Color Depth = 8 bits per channel HSYNC Timing = hav=1920, hfp=88, hsw=44(hsp=1), hbp=148 VSYNC Timing = vav=1080, vfp=04, vsw=05(vsp=1), vbp=036 Video Dimensions = 1920 x 1080 HDMI Output Initialization ... FMC-IMAGEON VITA Receiver Initialization ... FMC-IMAGEON VITA SPI Config for 10MHz ... FMC-IMAGEON VITA Initialization ... FMC-IMAGEON VITA Configuration for 1080P60 timing ... VITA Status =

Image Width = 1920 Image Height = 1080 Frame Rate = 60 frames/sec CRC Status = 0 Video Timing Controller (generator) Initialization ...

Video Resolution = 1080P Defect Pixel Correction (DPC) Initialization ...

DPC done Color Filter Array Interpolation (CFA) Initialization ...

CFA done

Done

Press ENTER to re-start ...

To re-start the initialization of the VITA receiver, press ENTER.

You have successfully executed the VITA pass-through on hardware !

References

All documentation supporting the ON Semiconductor Image Sensor with HDMI Input/Output FMC Bundle is available on the Avnet Design Resource Center (DRC):

<http://www.em.avnet.com/fmc-imageon-v2000c>

1. Getting Started with the HDMI Input/Output FMC Module

<http://www.em.avnet.com/fmc-imageon> → Support Files & Downloads

2. Avnet FMC-IMAGEON – Hardware User Guide

<http://www.em.avnet.com/fmc-imageon> → Support Files & Downloads

3. Getting Started with the ON Semiconductor Image Sensor with HDMI Input/output FMC Bundle

<http://www.em.avnet.com/fmc-imageon-v2000c> → Support Files & Downloads

The following reference provides links to documentation for video intellectual property (IP).

4. Video and Image Processing IP

http://www.xilinx.com/ipcenter/video/video_core_listing.htm

5. Video Timing Controller

<http://www.xilinx.com/products/intellectual-property/EF-DI-VID-TIMING.htm>

6. Video Input to AXI4-Stream

http://www.xilinx.com/products/intellectual-property/video_in_to_axi4_stream.htm

7. AXI4-Stream to Video Output

http://www.xilinx.com/products/intellectual-property/axi4_stream_to_video_out.htm

8. AXI Video DMA

http://www.xilinx.com/products/intellectual-property/axi_video_dma.htm

The following reference provides links to documentation for AXI interconnect.

9. UG761 - AXI Reference Guide

10. PG065 – AXI4-Stream Infrastructure

Known Issues and Limitations

The following issues are known to exist. When applicable, the workaround used is described.

Hello World Template – error: conflicting types for 'print'

The “Hello World” C project template has an issue that may manifest itself depending on which drivers are included in the design.

The “print(....)” declaration does not match the declaration in the xil_printf.h file and will result in the following error:

helloworld.c:29:6: error: conflicting types for 'print'

xil_printf.h:39:6: note: previous declaration of 'print' was here

The solution is to simply fix the “print(...)” declaration as shown below.

```
//void print(char *str);
```

```
void print( const char *ptr);
```

Troubleshooting

- 35 -