|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | |  |
|  | | |
|  | | |
| 方圆ok1 | | | |
|  | **FACS接口手册** | | 附件1-16K |
|  | |
| **文档版本** | **01** |
| **发布日期** | **2017-07-30** |
| **华为技术有限公司** | |

|  |
| --- |
| 版权所有 © 华为技术有限公司2017。 保留一切权利。  非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。  商标声明  附件3-版权声明页图和其他华为商标均为华为技术有限公司的商标。  本文档提及的其他所有商标或注册商标，由各自的所有人拥有。  注意  您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。  由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。 |

|  |  |
| --- | --- |
| 华为技术有限公司 | |
| 地址： | 深圳市龙岗区坂田华为总部办公楼 邮编：518129 |
| 网址： | <http://e.huawei.com> |
|  |  |
|  |  |

目 录

[目 录 iii](#_Toc534809604)

[1 FACS 接口手册简介 4](#_Toc534809605)

[1.1 概要 4](#_Toc534809606)

[1.2 术语介绍 4](#_Toc534809607)

[1.3 规格说明 5](#_Toc534809608)

[1.3.1 PCIe接口规格 5](#_Toc534809609)

[1.3.2 DDR接口规格 5](#_Toc534809610)

[1.3.3 队列规格 5](#_Toc534809611)

[1.3.4 BAR空间规格 5](#_Toc534809612)

[1.4 硬件说明 5](#_Toc534809613)

[1.4.1 硬件资源介绍 5](#_Toc534809614)

[2 FACS通用型(SDAccel)接口说明 6](#_Toc534809615)

[2.1 FACS SDAccel概要说明 6](#_Toc534809616)

[2.2 术语介绍 6](#_Toc534809617)

[2.3 FACS SDAccel方案说明 7](#_Toc534809618)

[2.4 FACS SDAccel开发说明 9](#_Toc534809619)

[2.4.1 Host Code开发 9](#_Toc534809620)

[2.4.2 Kernel Code开发 9](#_Toc534809621)

[2.5 Example设计说明 9](#_Toc534809622)

# FACS 接口手册简介

## 概要

FACS接口手册为华为fpga云服务器用户在实现加速设计的时候提供统一的设计说明，包括用户可以使用的硬件资源、接口说明、语法和使用示例。

## 术语介绍

FACS：FPGA Accelerated Cloud Server，现场可编程门阵列加速云服务。

HDK：Hardware Develop Kit，硬件开发套件。包括编码环境、仿真平台、自动化编译工具、代码加密和调试工具等。

SDK：Software Develop Kit，软件开发套件。包括应用示例、硬件抽象接口、加速器抽象接口、加速器驱动和runtime、版本管理工具等。

SHELL：静态逻辑，由华为对外提供。包括PCIe、DDR4等外部接口。

UL：User Logic，用户逻辑，由开发者提供。

PF：Phsical Function。

VF：Virtual Function。

DMA：Direct Memory Access，直接存储器访问。

HPI：High Performance Interface，高性能接口。

BD：Buffer Description，缓存描述符。

PR：Partial Reconfigration，部分重加载技术。

AE：Acceleration Engine，加速引擎。

Bar：BAR Base Address Register，基地址寄存器。

AXI-4 ：ARM Advanced eXtensible Interface。

AXI-4 Stream：ARM Advanced eXtensible Stream Interface。

AXI-4 Lite ：ARM Advanced eXtensible Lite Interface。

M-指：AXI总线Master端。

S-指：AXI总线Slave端。

PMD：Poll Mode Drivers，轮询模式驱动。

## 规格说明

### PCIe接口规格

单个FPGA支持一个PCIe X16接口，最大支持110Gbps带宽。

### DDR接口规格

单个FPGA支持4个DDR4通道，单通道DDR4容量为16GB，共64GB。

### 队列规格

* 1个PF。
* 1个VF。
* 每个VF支持8个队列。

### BAR空间规格

UL的VF Bar空间使用规格：

* BAR2为64-bit BAR，大小为16MiByte。
* BAR4为64-bit BAR，大小为512MiByte。

## 硬件说明

### 硬件资源介绍



用户可以使用的资源如下：

* 1个PCIe Gen3 x16接口。
* 4个DDR4 DIMM接口。
* 9个100Gbps的ETH300G接口。

DDR区域划分：

* 1个DDR控制器放置在静态逻辑部分。
* 3个DDR控制器放置在用户逻辑部分。
* 支持用户最多使用4个DDR控制器。

# FACS通用型(SDAccel)接口说明

## FACS SDAccel概要说明

FACS SDAccel是以Xilinx SDAccel 2017.4.op版本为原型，支持用户在虚机上进行基于OpenCL的开发和应用，且在开发及应用方式上兼容Xilinx SDAccel 2017.4.op版本。本章节将简要介绍FACS的SDAccel方案，以及针对用户如何开发Host、Kenel做简单说明。

## 术语介绍

HAL：Hardware Abstraction Layer，硬件抽象层

XDMA：Xilinx Direct Memory Access，Xilinx直接存储器访问

Mgmt：Management

Drv：Driver，驱动

PF：Physical Function

OpenCL：Open Computing Language，开放运算语言

## FACS SDAccel方案说明

支持将SDAccel方案部署在虚机内，用户在虚机运行OpenCL Host/kernel code的开发、调试及应用。基于用户安全使用的考虑，FPGA的管理由Host来完成，因此用户在虚机内发起的部分Kernel的管理操作实际是在Host上完成，例如：xclbin加载。



FACS SDAccel方案框图

FACS SDAccel方案说明：

1. OpenCL Runtime：Xilinx OpenCL Runtime，对用户呈现OpenCL API
2. HAL：实现OpenCL Runtime和Kernel Driver的适配及Global Memory的地址管理
3. XDMA Drv：Xilinx DMA内核驱动
4. Fpga tool：提供基本的fpga管理工具，例如：fpga kernel加载命令，加载状态查询
5. Host Mgmt Drv：运行在Host的管理驱动，实现fpga kernel的加载
6. User PF：用户面PF接口，直通到虚机，为用户提供FPGA访问通道
7. Mgmt. PF：管理面PF接口，为主机访问FPGA的通道
8. Based Resgion：FPGA静态逻辑区域
9. Expanded Region：FPGA动态逻辑区域

由于User PF和Mgmt PF分别绑定到用户虚机和服务器主机，因此用户在虚机上直接执行管理相关的命令会受到限制：首先，FACS SDAccel在虚机上提供其管理工具，fpga\_tool，用户在虚机时，需要使用fpga\_tool工具进行管理相关的操作；其次，相较于在xilinx xbsak工具而言，fpga\_tool工具支持的命令有限，暂时还未支持所有的xbsak命令。因此HAL层中一些API用户将无法在虚机使用，FACS SDAccel目前支持的HAL的API如下：

|  |  |  |
| --- | --- | --- |
| **序号** | **API名称** | **备注** |
| 1 | xclOpen( ) | - |
| 2 | xclClose( ) | - |
| 3 | xclGetDeviceInfo2( ) | 支持，但有修改 |
| 4 | xclGetAXIErrorStatus( ) | - |
| 5 | xclLoadXclBin( ) | 支持，但有修改 |
| 6 | xclAllocDeviceBuffer( ) | - |
| 7 | xclAllocDeviceBuffer2( ) | - |
| 8 | xclFreeDeviceBuffer( ) | - |
| 9 | xclCopyBufferHost2Device( ) | - |
| 10 | xclCopyBufferDevice2Host( ) | - |
| 11 | xclWrite( ) | - |
| 12 | xclRead( ) | - |
| 13 | xclReClock2( ) | 支持，但有修改 |
| 14 | xclProbe( ) | - |
| 15 | xclLockDevice( ) | - |
| 16 | xclWriteHostEvent( ) | - |
| 17 | xclGetDeviceTimestamp( ) | - |
| 18 | xclGetDeviceClockFreqMHz( ) | - |
| 19 | xclGetReadMaxBandwidthMBps( ) | - |
| 20 | xclGetWriteMaxBandwidthMBps( ) | - |
| 21 | xclSetOclRegionProfilingNumberSlots( ) | - |
| 22 | xclPerfMonClockTraining( ) | - |
| 23 | xclPerfMonStartCounters( ) | - |
| 24 | xclPerfMonStopCounters( ) | - |
| 25 | xclPerfMonReadCounters( ) | - |
| 26 | xclPerfMonStartTrace( ) | - |
| 27 | xclPerfMonStopTrace( ) | - |
| 28 | xclPerfMonGetTraceCount( ) | - |
| 29 | xclPerfMonReadTrace( ) | - |

## FACS SDAccel开发说明

### Host Code开发

完整支持Xilinx SDAccel 2017.4.op所支持的OpenCL API，具体的支持API见xilinx UG1027。用户可以基于xilinx SDx 2017.4.op版本支持的Host code开发。

UG1023链接如下。<https://www.xilinx.com/support/documentation/sw_manuals/xilinx2017_4/ug1023-sdaccel-user-guide.pdf>

### Kernel Code开发

完整支持Xilinx SDAccel 2017.4.op OpenCL C的语法，具体的支持见UG1023、UG1207。用户基于xilinx SDx 2017.4.op版本支持的语法进行Kernel Code的开发。

UG1207链接如下。<https://www.xilinx.com/support/documentation/sw_manuals/xilinx2017_4/ug1207-sdaccel-optimization-guide.pdf>

## Example设计说明

FACS SDAccel提供了三个example，分别从三种不同的Kernel开发方式（即，RTL开发kernel，HLS开发kernel和OpenCL C开发kernel）上进行开发调试过程的说明。每个example分为两个部分：

1. kernel的开发及仿真在/huaweicloud-fpga/fp1/hardware/sdaccel\_design/examples目录下，并提供了对应的执行脚本。
2. Host code的开发及上板调试放在/huaweicloud-fpga/fp1/software/app/sdaccel\_app目录下，并提供了对应的执行脚本。

* example vadd\_cl

该example示例了以OpenCL C进行开发kernel的流程，其kernel源码来自于Xilinx SDx安装工具的<SDx\_install\_area>SDx/2017.4.op/examples/vadd。同时该example提供了kernel编译，仿真的脚本，如果用户选择OpenCL C的开发方式，用户需要替换掉kernel源代码，然后执行example提供的脚本进行编译及仿真调试或者上板调试。

* example mmult\_hls

该example是示例了以HLS进行开发kernel的流程，其kernel源码来自于Xilinx SDx安装工具的<SDx\_install\_area>SDx/2017.4.op/ samples/mmult。同时该example提供了kernel编译，仿真的脚本，如果用户选择HLS的开发方式，用户需要替换掉kernel源代码，然后执行example提供的脚本进行编译及仿真调试或者上板调试。

* example vadd\_rtl

该example示例了以RTL进行开发kernel的流程，其kernel源码来自于Xilinx github开源例子，链接为：<https://github.com/Xilinx/SDAccel_Examples/tree/master/getting_started/rtl_kernel>，该example提供了kernel编译，仿真的脚本，当用户选择RTL的开发方式，用户需要RTL kernel源代码及xml说明文件，然后执行example提供的脚本进行编译及仿真调试或者上板调试。

关于SDAccel的开发调试过程，用户也可以参见xilinx tutorial文档，链接如下。

https://www.xilinx.com/support/documentation/sw\_manuals/xilinx2017\_4/ug1207-sdaccel-optimization-guide.pdf