

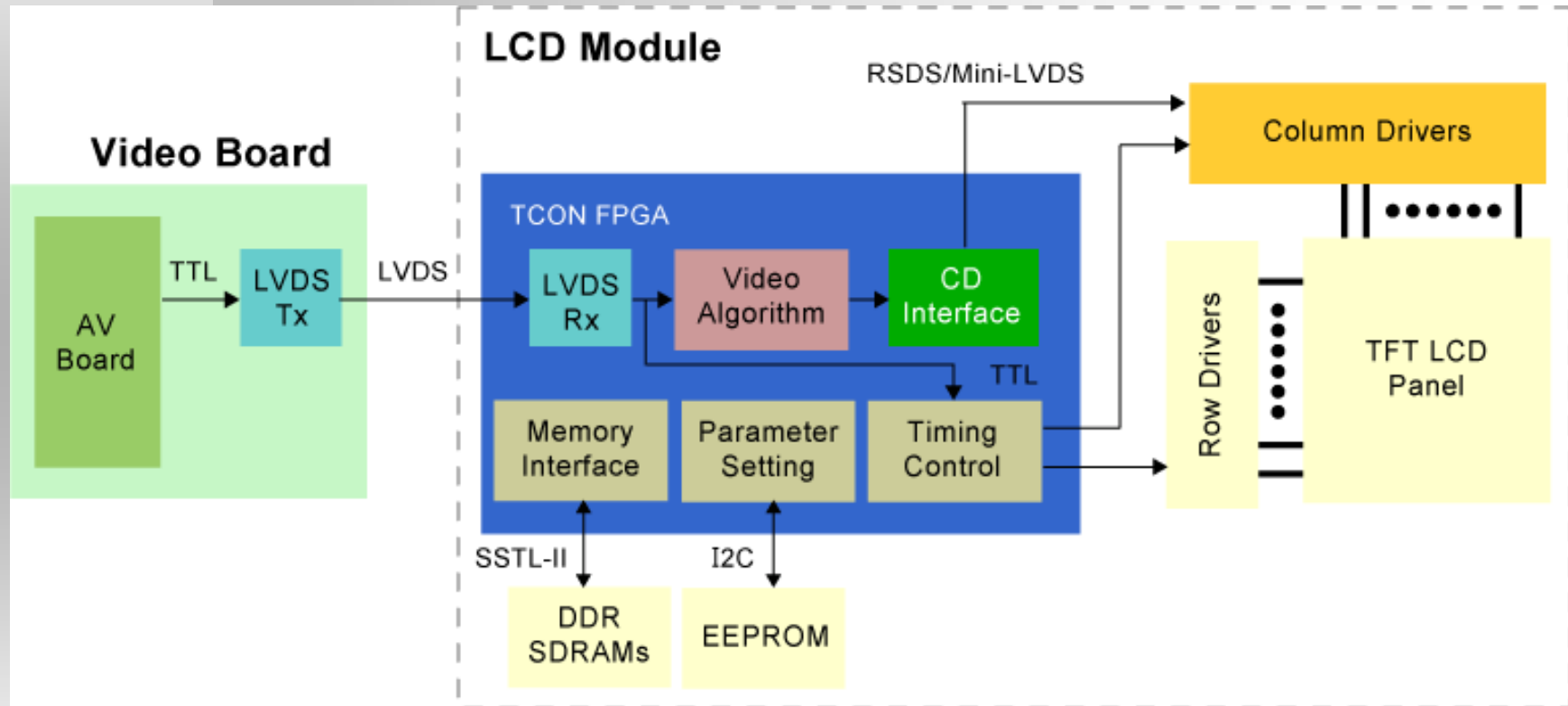


FPGA 在视频处理领域的应用

www.farsight.com.cn

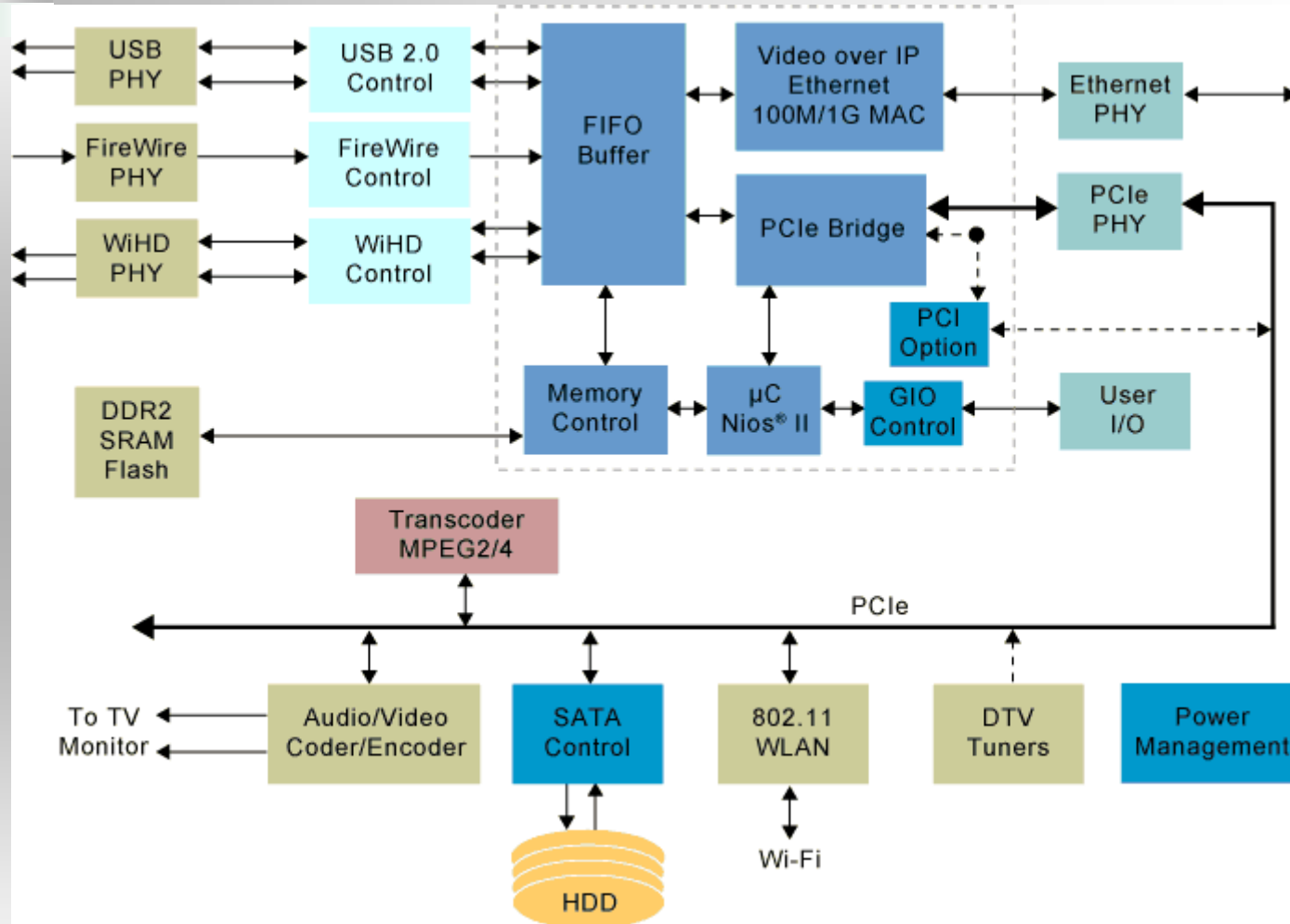
www.FPGADev.com

- √ 介绍视频处理领域FPGA的主要应用场合
- √ 视频处理领域常用的IP模块
- √ FPGA + DSP的系统设计方法

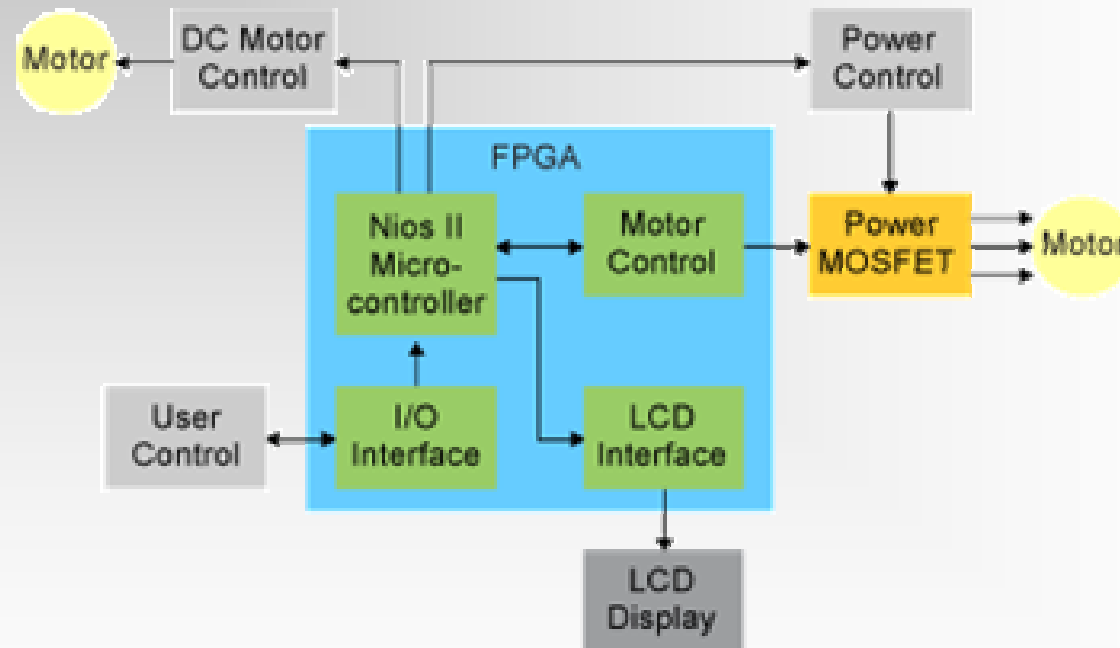


- √ 平面显示器(LCD、等离子面板、等离子小截面显示、硅液晶等)和视频投影仪从各种信息源中接收、解码并显示数字视频流。
- √ LCD HDTV的核心是其图像处理和时序控制模块(如图1所示)。图像处理模块一般包括扫描速率转换器、帧速率转换器、色彩解码器、移动探测、scalar和去隔行等功能。
- √ HDTV LCD显示器的色彩响应时间取决于色彩内容,要比传统的显示器慢。这对图像处理算法而言是一种挑战,需要进行更多的处理来消除相关的显示瑕疵。FPGA在设计上的灵活性有很大优势,您可以在器件中重新设计算法,而不用重新编程。
- √ 数据处理后,视频板通过400Mbps的LVDS总线或者低摆幅差分信号(RSDS)总线与LCD行列驱动进行接口。

针对消费娱乐的家庭网络连接



- 现代化的家电(参见图1)支持以下功能：LCD触摸屏等人机接口(HMI)、用于监控和诊断的网络连接、电机控制、传感器以及功耗管理等。



常用模块一：RAM/ROM/CAM

片上存储器由内嵌于FPGA内部的逻辑资源（分布RAM和块RAM）搭建而成。搭建出来的这些片上存储器的调用方法与分立器件基本一致，但却为开发者省去了大量的PCB布线资源。在小容量的存储器设计中有着非常广泛的应用，同时这些片上存储器还常常用作数据缓冲及时钟域转换等工作。

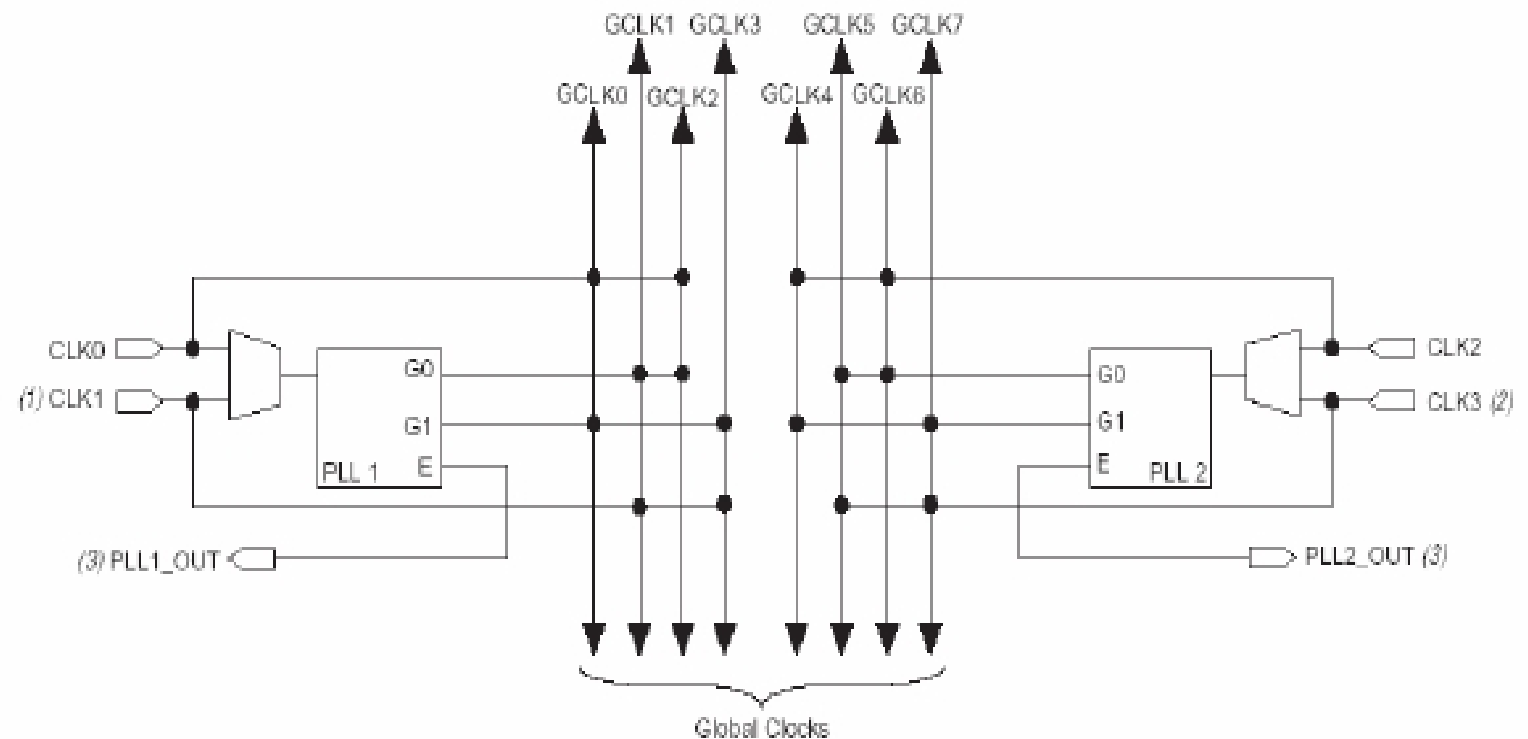
常用模块二：锁相环

锁相环在数字系统中的应用非常广泛，在Altera的FPGA产品中集成的锁相环是PLL模块，在Xilinx的FPGA产品中集成的锁相环是DLL模块。两者采用的技术不同，但是实现的基本功能大抵一致，但也各有特点。

Altera的PLL结构

- ✓ Enhanced PLL
- ✓ Fast PLL
- ✓ PLL

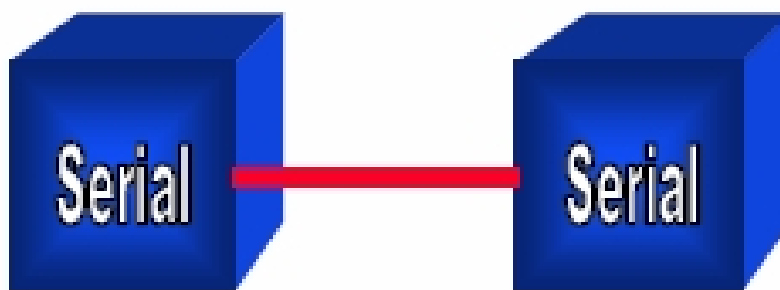
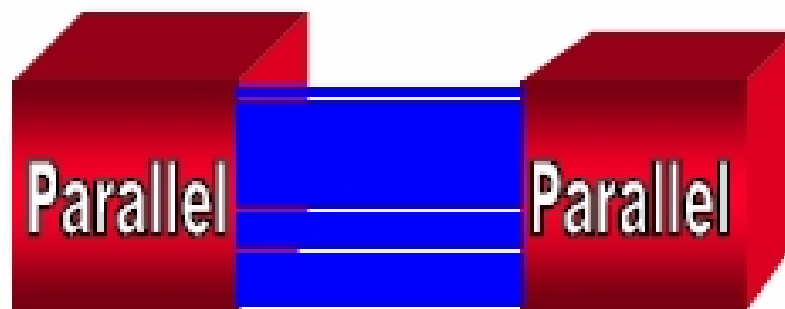
Cyclone系列的PLL结构



常用模块三：高速串行收发器

- 并行数据传输：
 - 多根线占用板面积；
 - 每根线相互之间有干扰；
 - 每根线需要自己的匹配电路；

- 串行数据传输：
 - 更少的走线占用的板面积减少；
 - 信号线的干扰降到最小；
 - 相对于并行传输只用了一小部分的匹配电路；
 - 没有相位差(skew)的问题；



基于DSP芯片的信号处理系统

- ✓ DSP芯片是基于软件可编程的，因此不可避免的问题就是功耗大，处理能力受到主频的限制等不足。而且，DSP芯片的数据带宽受到总线接口位宽限制，不能应付高速数据流的处理要求。
- ✓ 通用DSP的接口扩展能力有限，不能直接和ADC或者DAC芯片互联。DSP需要外部的逻辑器件来配合才能实现数据采集或者输出，例如CPLD或者FPGA。
- ✓ 目前大部分的DSP都没有内存管理单元（MMU），还不能支持复杂的操作系统，如Linux或者WinCE等。DSP的内存管理和保护能力比较弱，目前还不支持虚拟内存。

FPGA + DSP 系统的优势

- ✓ (1) 支持更高的计算处理能力
- ✓ (2) 灵活的系统重构方案
- ✓ (3) 开发难度和系统成本的折衷

FPGA + DSP系统的适用领域

- ✓ 基于FPGA + DSP的非常适合既有复杂算法处理又需要大量的重复计算应用，例如软件无线电，高清视频图像处理，信息安全等领域。
- ✓ FPGA作为DSP的协处理器，给DSP提供单元加速功能。例如，FPGA可以实现一个浮点协处理单元（FPU），给高性能的定点DSP提供浮点计算能力，适合于数值分析和高精度信号处理领域。
- ✓ 基于FPGA + DSP的系统配置方式比较灵活。FPGA和DSP可以单独配置，也可以相互配置。因此，这种方案可以用于智能的动态重构计算平台。

FPGA + DSP 系统的设计流程

设计可以分为三种主要任务：

✓ 功能划分

需要把整个功能分解为DSP实现和FPGA实现两个部分，同时细化FPGA和DSP上实现的模块定义

✓ 接口设计

设计DSP和FPGA的接口时序和数据格式

✓ 调度安排

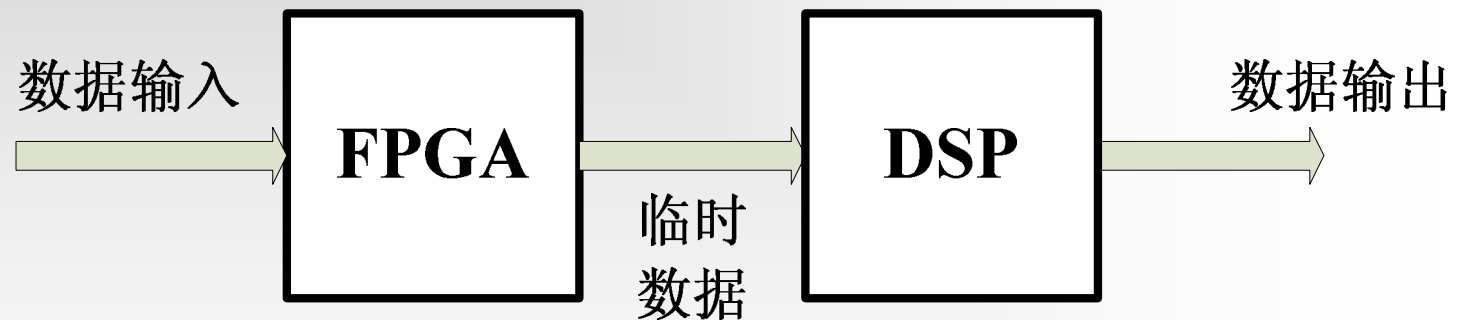
合理设计好DSP的每个模块的执行时间，特别是多个模块共享一个硬件资源的时候。

功能划分的原则和依据

- ✓ 功能划分的基本原则就是把数据密集，但算法简单，重复性高的功能放在FPGA上执行，而把算法复杂，重复性低的功能分配给DSP实现。这样可以重复发挥两种芯片的特长。
- ✓ 功能划分的主要依据是系统的指标需求和功能定义，即系统在进行功能划分之后，不能影响系统的整体性能和指标。

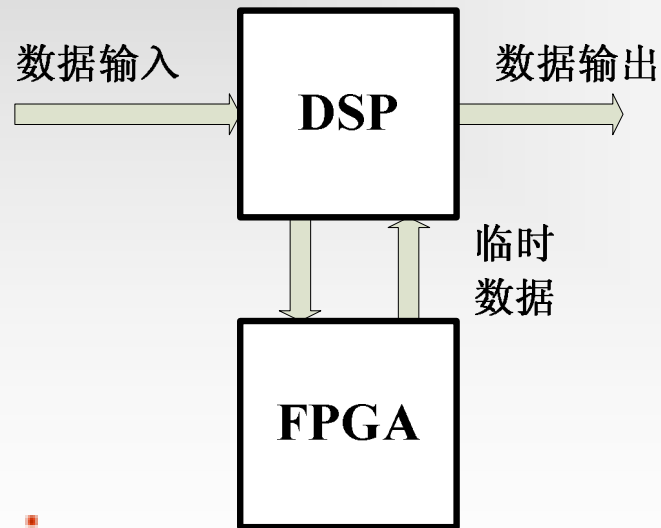
✓ 按照数据流向来划分

这种划分方法一般应用于数据采集领域，具体来说，就是FPGA负责采集数据，并把数据送给DSP芯片，而DSP用来完成对采集到的数据进行处理。

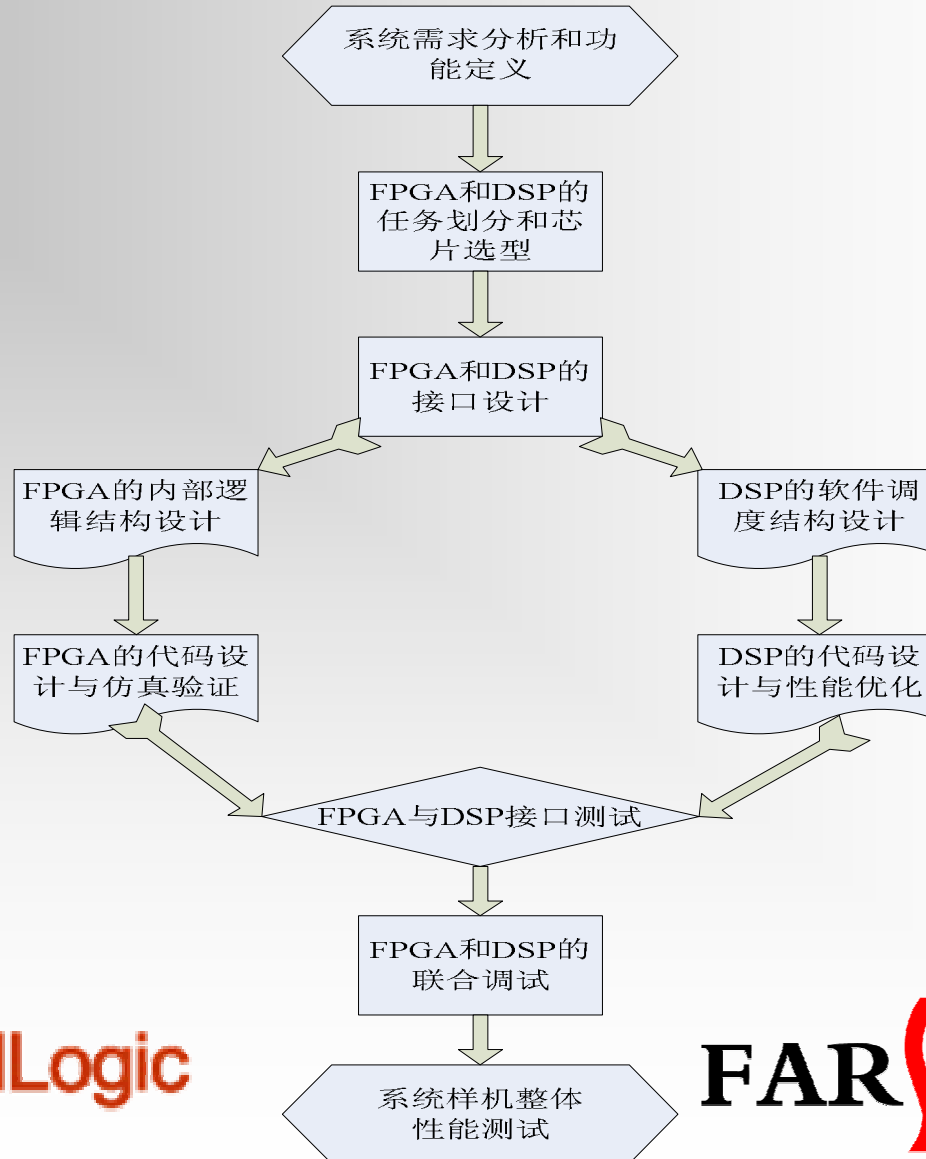


✓ 按照算法流程来划分

如果FPGA作为DSP的协处理器，那么一般采用按算法流程来划分的方法。DSP负责整个算法的调度和数据流的控制，把需要处理的数据送给FPGA，并从FPGA取回处理后的结果。



FPGA + DSP的系统设计流程



FPGA与DSP的通信接口设计

- ✓ 存储型接口，如EMIFA，EMIFB等；
- ✓ 主机型接口，如HPI等；
- ✓ 总线型接口，如PCI，EMAC等；
- ✓ 串口型接口，如McBSP，McASP，SPI，UART等。
- ✓ 特殊类接口，如VLYNQ，VPORT等。

以上的接口中，比较常用的接口是EMIF和HPI。其中总线接口需要协议支持，开发难度较大，串行接口开发简单，但是速率较慢。VPORT等特殊接口一般是在特定的场合下应用，不具备通用性，而且需要修改DSP驱动，开发周期较长。

√ EMIF接口

Feature	C62x/C67x EMIF			C64x EMIF [‡]		
	C6201/C6701	Other C620x/C670x	C621x/C671x [†]	EMIFA		EMIFB
				C6416/15/14/12, DM642	C6411, DM640/641	C6416/15/14
Bus width	32-bit	32-bit	32-bit [†]	64-bit	32-bit	16-bit
Number of memory spaces	4	4	4	4	4	4
Addressable space (Mbytes)	52	52	512 [†]	1024	512	256
Synchronous clocking	CPU clock and/or 1/2× CPU clock	1/2× CPU clock	Independent ECLKIN	Independent ECLKIN, 1/4× CPU clock or 1/6× CPU clock	Independent ECLKIN, 1/4× CPU clock or 1/6× CPU clock	Independent ECLKIN, 1/4× CPU clock or 1/6× CPU clock
Width support	32 bit; 8-/16-bit ROM	32 bit; 8-/16-bit ROM	8-/16-bit, 32-bit [†]	8-, 16-, 32-, or 64-bit	8-, 16-, or 32-bit	8-bit or 16-bit
Supported memory type at CE1	Asynchronous memory	Asynchronous memory	All types	All types	All types	All types
Control signals	Separate	Muxed synchronous signals	Muxed all control signals	Muxed all control signals	Muxed all control signals	Muxed all control signals

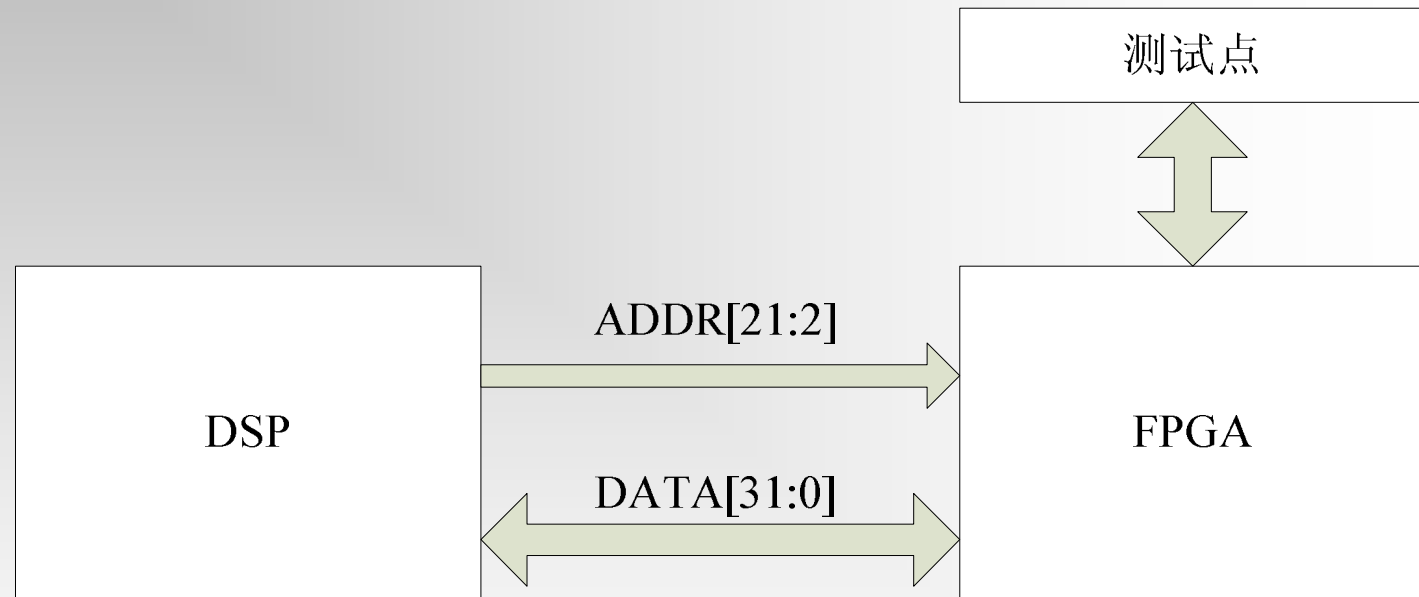
FPGA+DSP的协同处理平台的 电路设计注意事项

√ FPGA和DSP的隔离调试技术

对于FPGA芯片，如果没有进行配置，那么所有的管脚都处于高阻状态。由于高阻态没有驱动能力，所以不会对DSP产生直接的影响。但是在某些情况下，会产生间接的影响。

例如，很多DSP会在系统复位的时候读取某些地址信号来确定启动后的工作模式。如果这些信号连接到了FPGA的管脚上，那么上电后这些信号就会处于不确定的状态。如果处于错误的电平状态，就会对DSP产生很大的影响。

✓ FPGA测试点的设计



✓ 借助FPGA的内部逻辑分析仪来辅助调试

Altera公司的Signal TAPII和Xilinx公司的Chip Scope等。这些工具可以用来测试FPGA运行状态下信号的变化情况，特别是总线的运行状态。在FPGA + DSP的系统中，DSP的很多信号需要连接到FPGA的管脚。利用内部逻辑分析仪功能，就可以通过FPGA来观察这些信号的时序和状态，提高了调试的可视化程度。

华清远见

让我们一起讨论！



 RedLogic

FAR  SIGHT



www.farsight.com.cn

www.FPGADev.com

谢谢！