



FPGA的DSP应用

高遗

www.farsight.com.cn

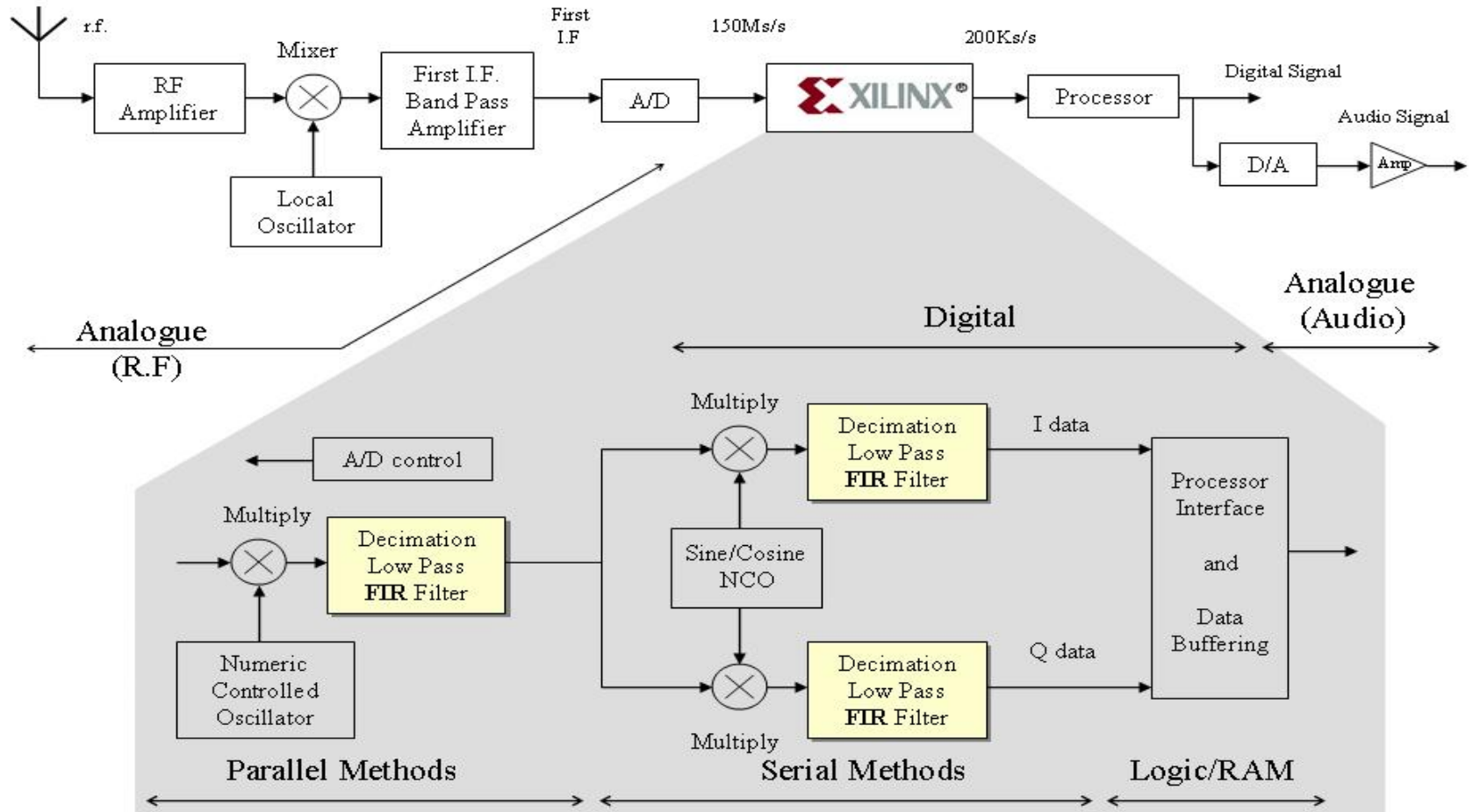
内容安排

- } **DSP**的应用
- } 用 **FPGA** 实现**DSP**的优势
- } 基本的**FPGA**资源
- } 内嵌**DSP48**的介绍
- } 基于**System Generator**的**DSP**设计

DSP的应用

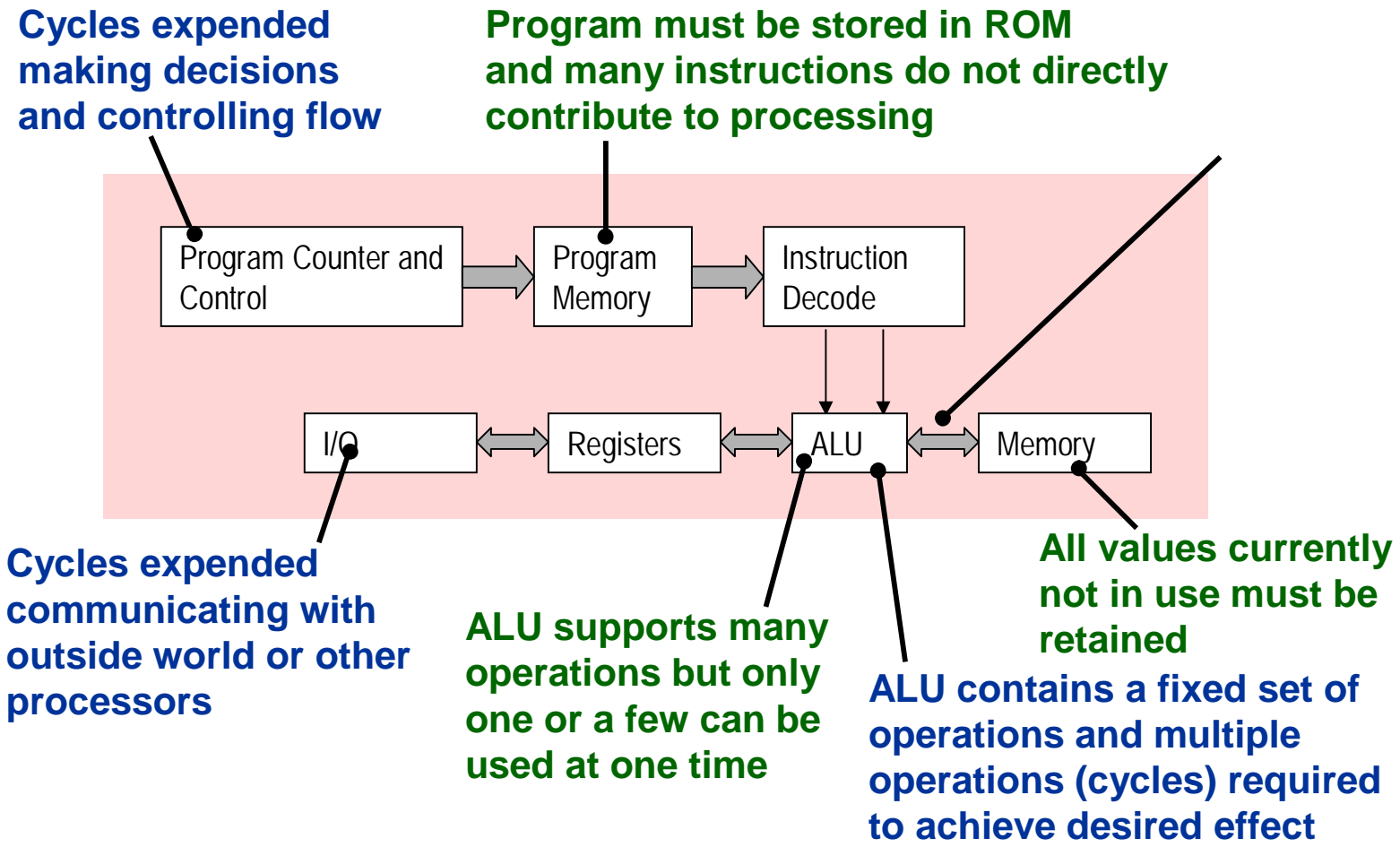
- } 无线通信领域,如软件无线电(SDR)
- } 视频图像处理领域,如高清数字电视(HDTV)
- } 军事和航空航天领域,如雷达声纳

无线接收机

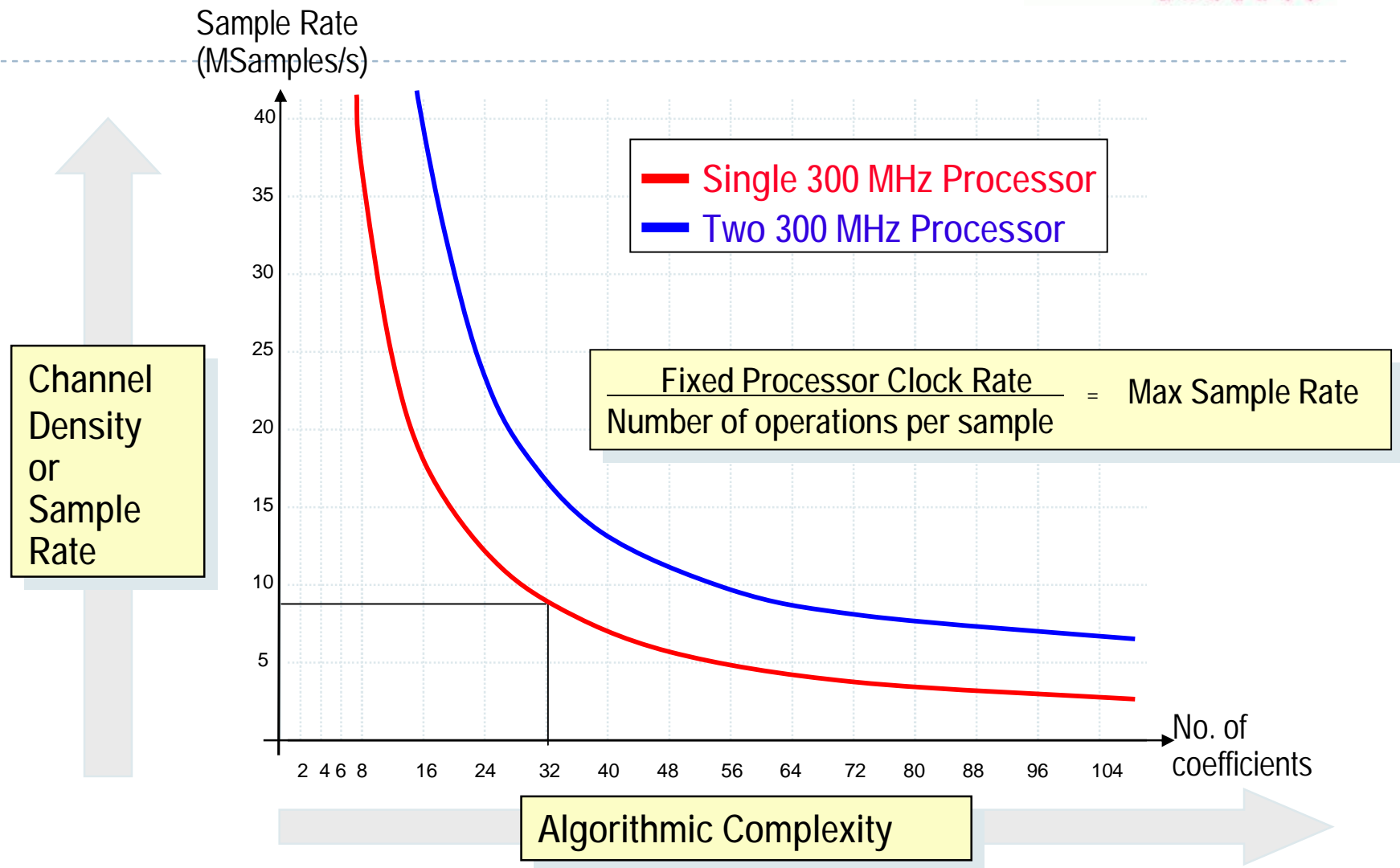


DSP处理器

算法要用固定的结构来实现

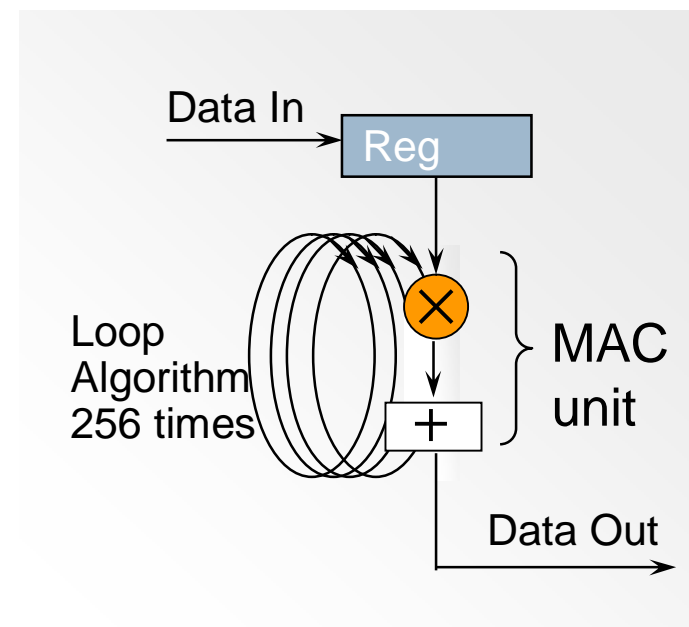


串行系统限制了系统性能



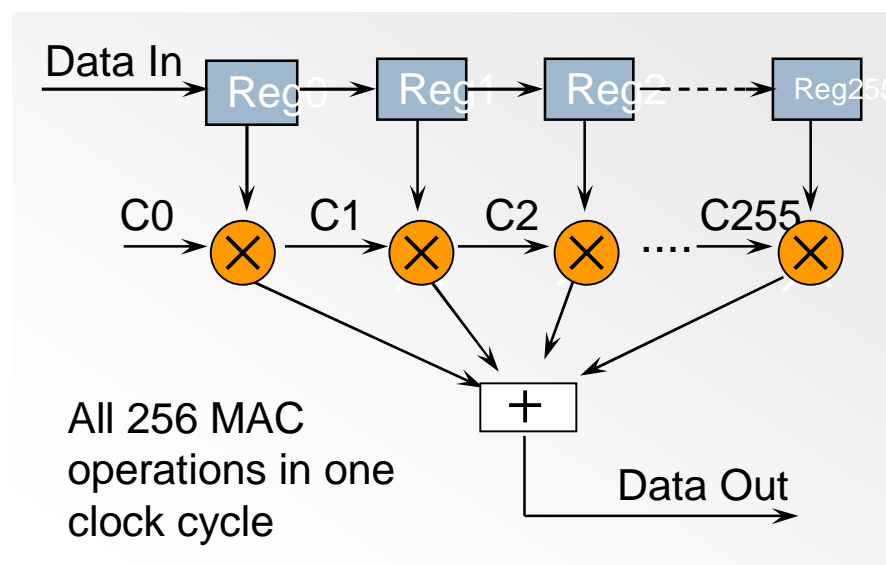
单引擎乘加器

- √ 串行处理限制了数据的吞吐量
 - ∅ 时分复用的 MAC 单元
 - ∅ 高时钟频率使系统设计比较困难
- √ 256 抽头的FIR滤波器
 - ∅ 256 乘和累加(MAC)操作每采样数据
 - ∅ 每256个时钟输出一个结果



多引擎乘加器

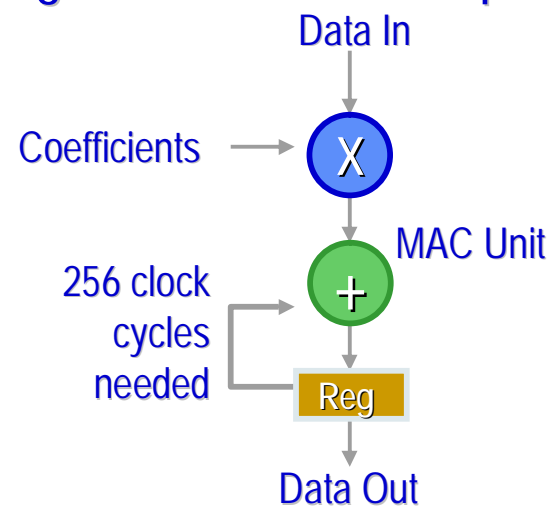
- ✓ 并行处理使数据的吞吐量最大
 - ∅ 支持任何层次的并行
 - ∅ 最佳的性能/成本的折中
- ✓ 256抽头的FIR滤波器
 - ∅ 256个乘加器 (MAC) 每采样数据
 - ∅ 每个时钟输出一个结果
- ✓ 灵活的结构
 - ∅ 分布的DSP资源 (LUT, registers, multipliers, & memory)



FPGA能做大运算量的处理

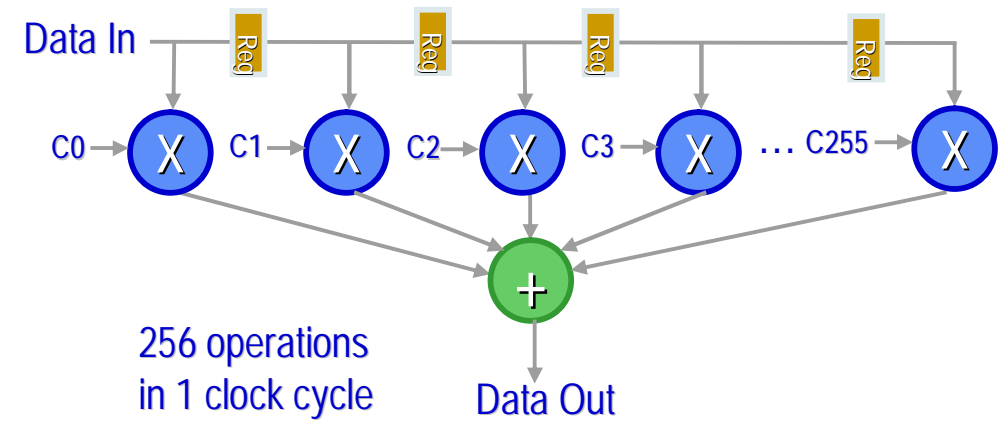
利用并行结构来加速FIR滤波器

Programmable DSP - Sequential



$$\frac{1 \text{ GHz}}{256 \text{ clock cycles}} = 4 \text{ MSPS}$$

FPGA - Fully Parallel Implementation

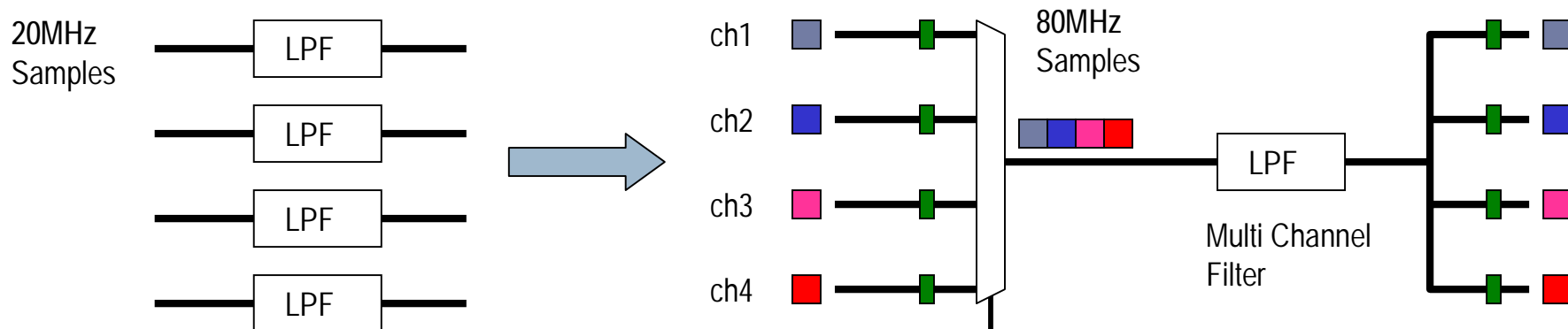


$$\frac{500 \text{ MHz}}{1 \text{ clock cycle}} = 500 \text{ MSPS}$$

FPGA是多通道DSP设计的理想器件



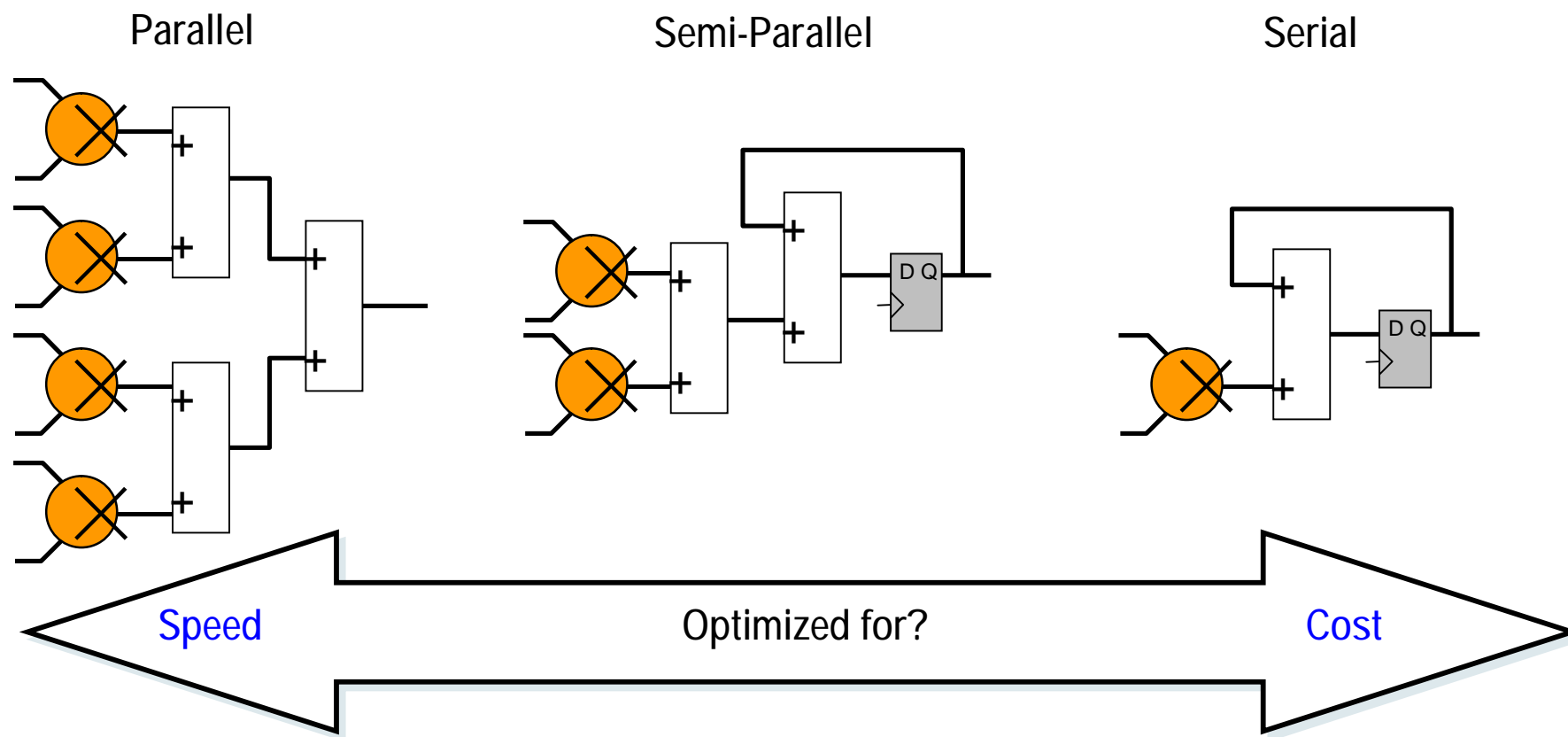
能实现多通道并行处理或使用时分复用



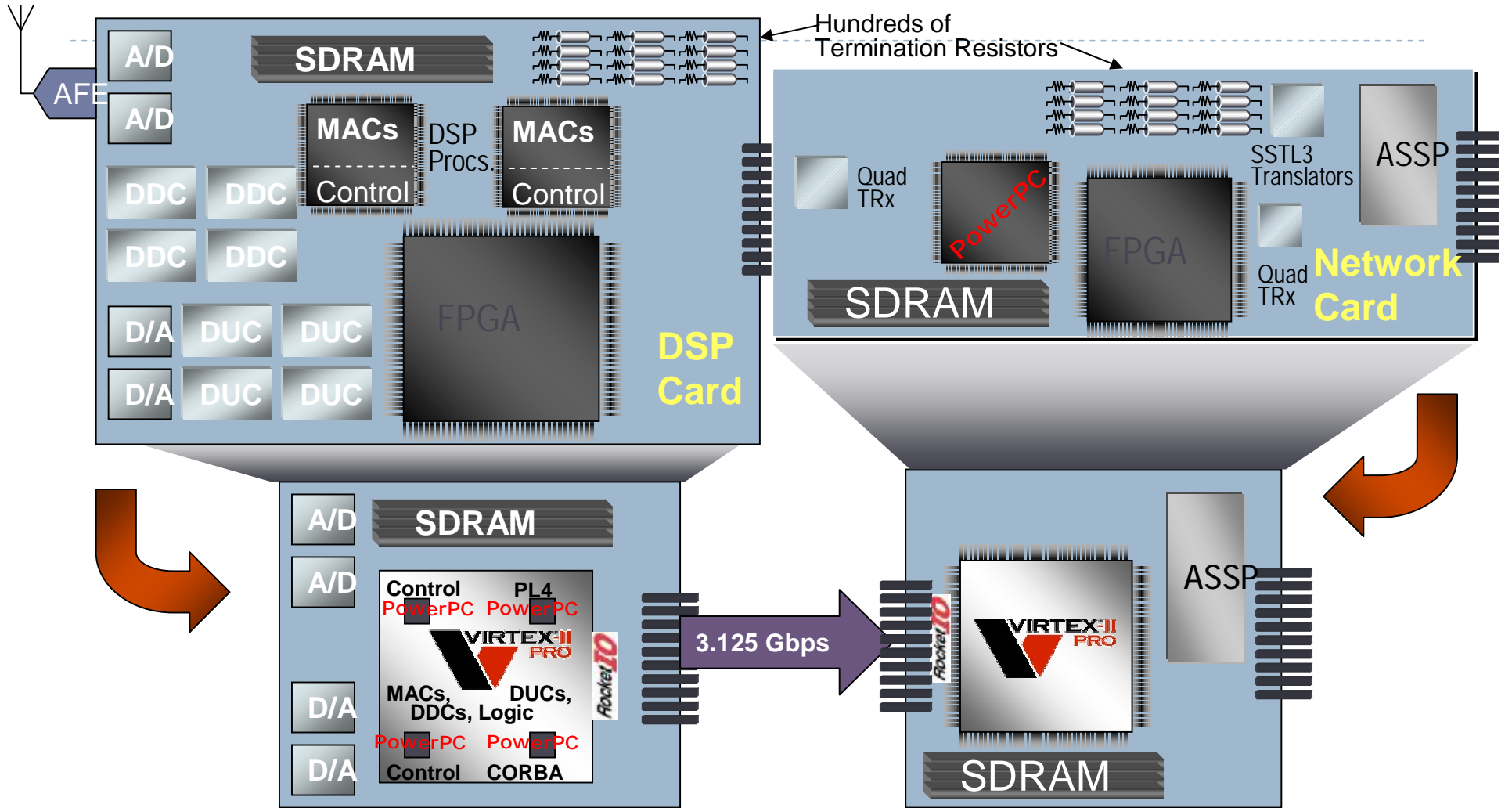
- } 很多低采样率的通道能在FPGA中使用较高的频率，利用时分复用(TDM)的方式进行处理。

可定制结构来满足需求

FPGAs 可在成本和性能之间进行折中



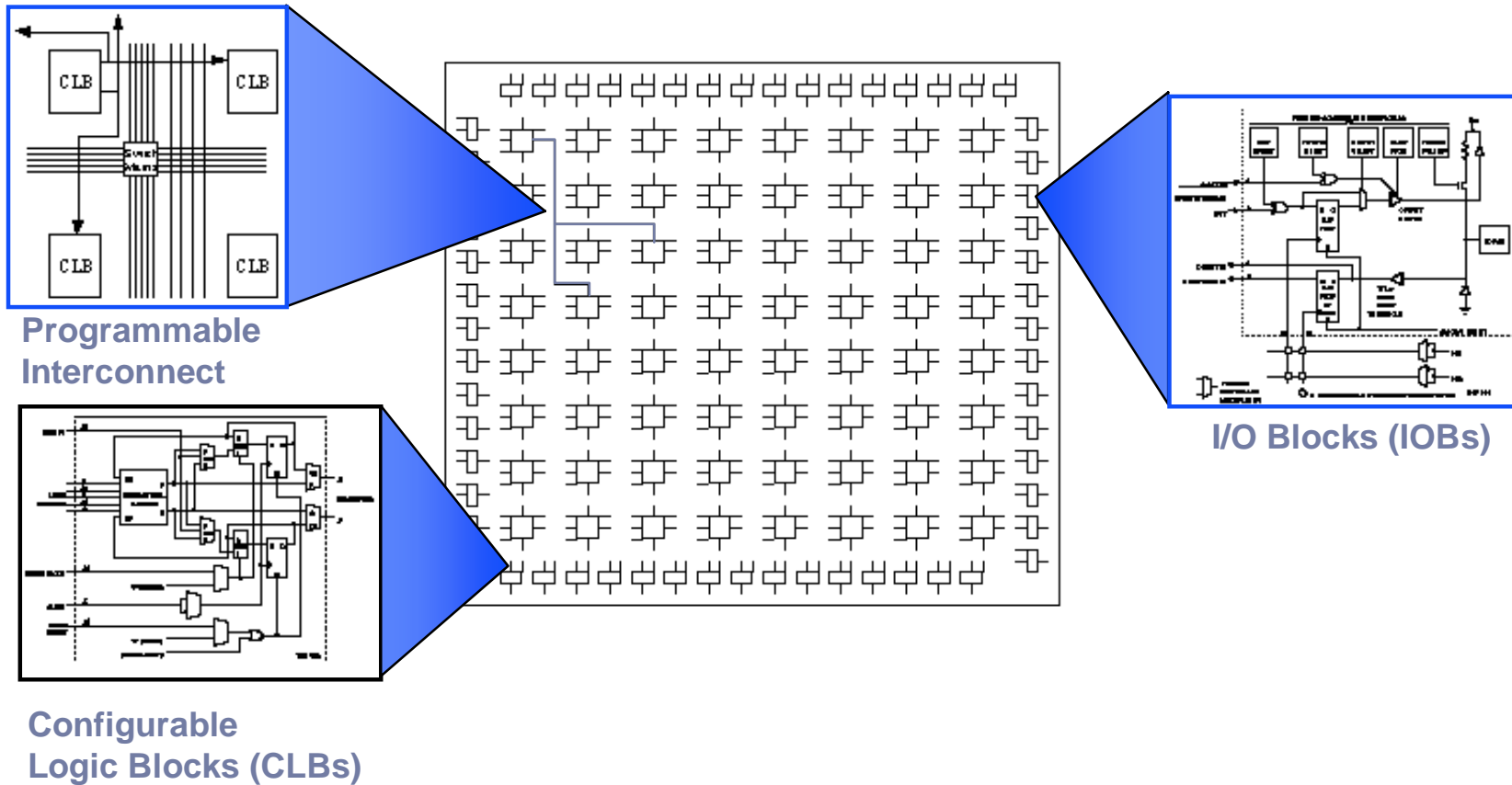
可通过集成来降低成本



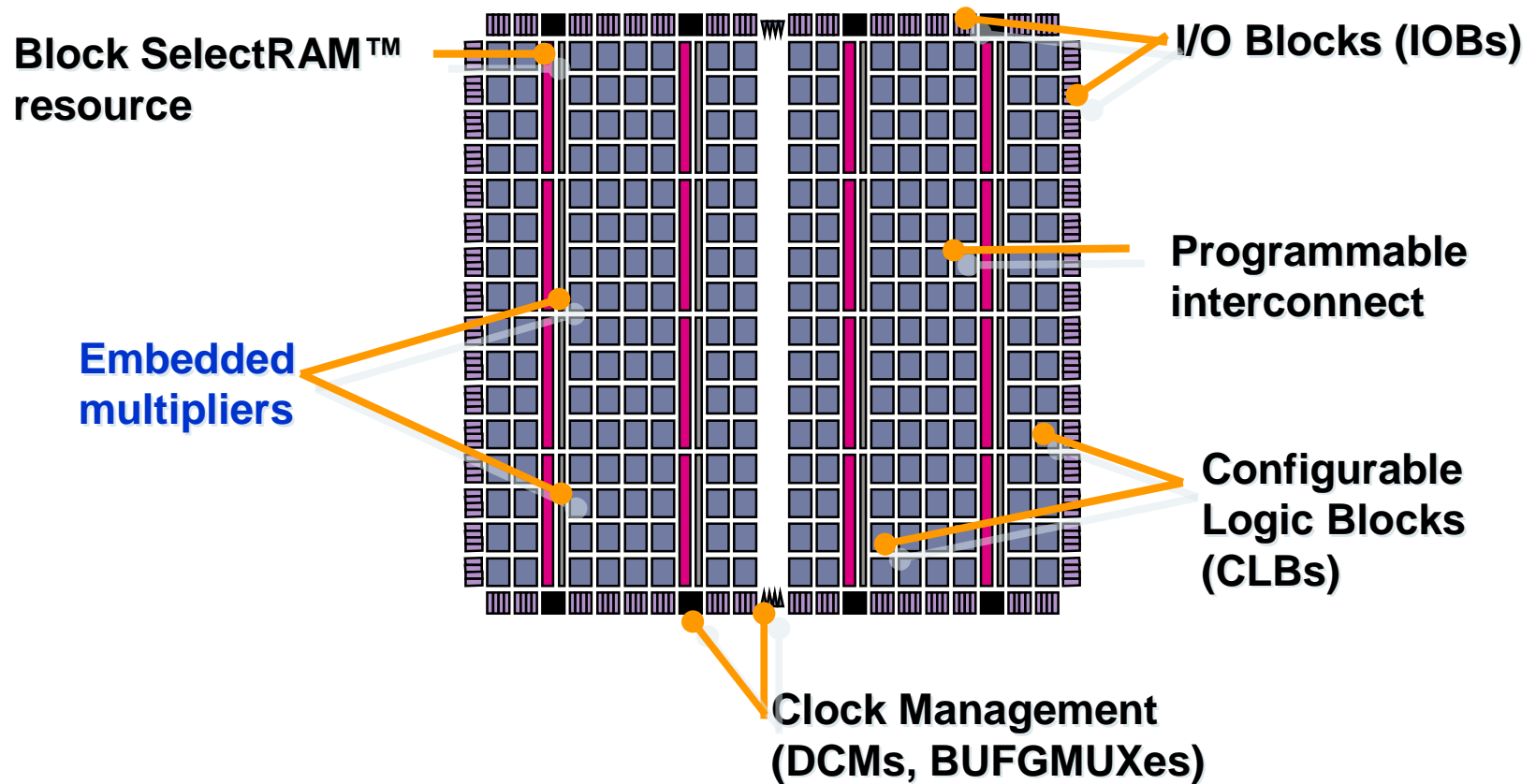
FPGA简介

- } 所有的 Xilinx FPGAs 包含了相同的基本资源
 - } Slices 组成可编程逻辑块 (CLBs)
 - } 包含组合逻辑和寄存器资源
 - } 输入与输出 (IOBs)
 - } FPGA与外围电路的接口
 - } 可编程的内部连接
 - } 别的资源
 - } 存储器
 - } 乘法器
 - } 全局时钟缓冲器
 - } 边界扫描逻辑

FPGA的基本结构

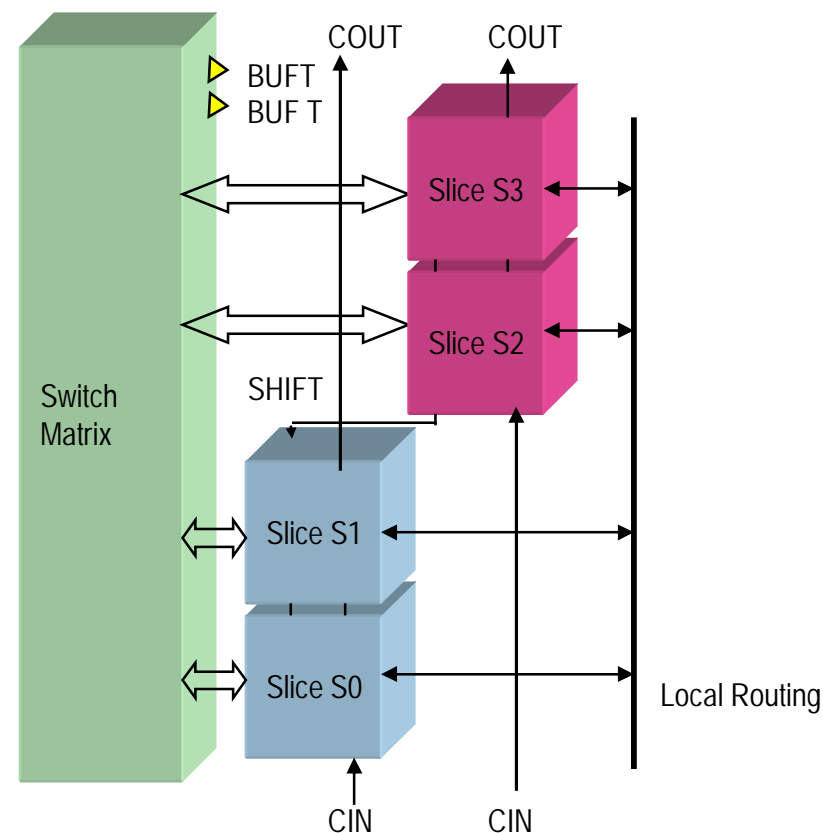


Virtex-II的结构



CLBs 和 Slices

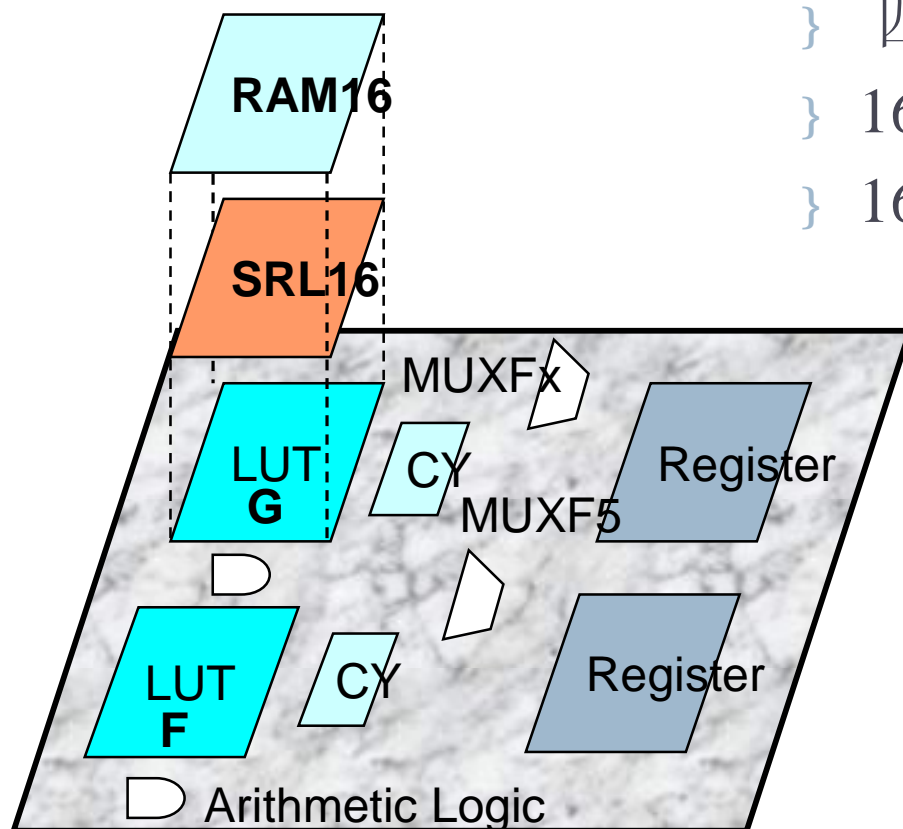
- } 每个 CLB 包含4个slices
 - } 本地的布线提供在同一个 CLB内的slices之间的反馈, 并且提供到临近的 CLB 布线
 - } 一个开关矩阵提供访问通用资源的能力



Slice 资源

} 每个slice包含两个:

- } 四输入的查找表
- } 16-bit 分布 SelectRAM
- } 16-bit 移位寄存器



每个寄存器:

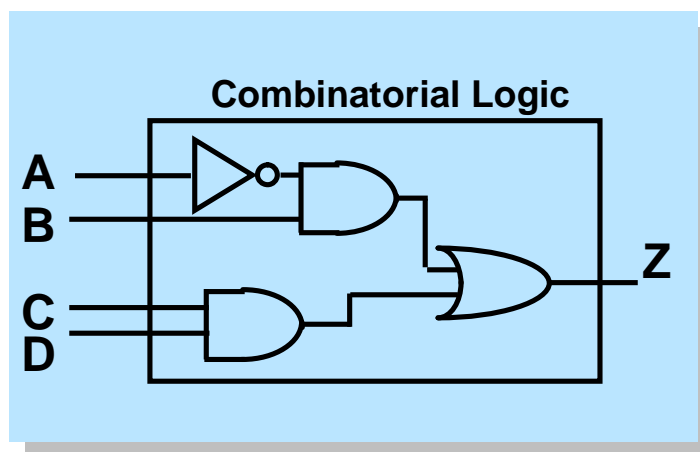
- D 触发器
- 锁存器

专用逻辑:

- 开关 (Muxes)
- 算数逻辑
 - MULT_AND
 - 进位链

查找表

- } 组合逻辑存放在查找表 (LUTs)
- } 也叫做函数发生器 (FGs)
- } 容量局限于输入而不是复杂度
- } 通过 LUT 的延时是固定的



A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
	.	.	.	
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

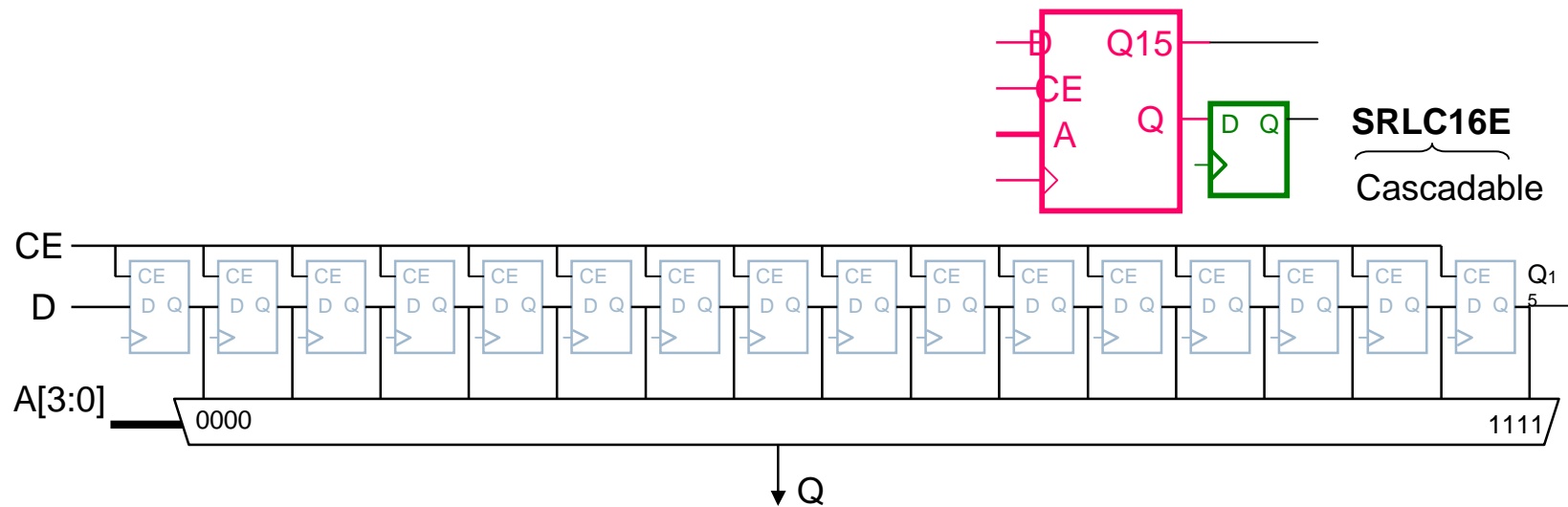
SRL16E模块

16 SRAM 单元组成一个移位寄存器

- 使用CE和clock, 往第一个寄存器写数据就实现了右移一位
- 因为是可预测的操作, 不需要写地址

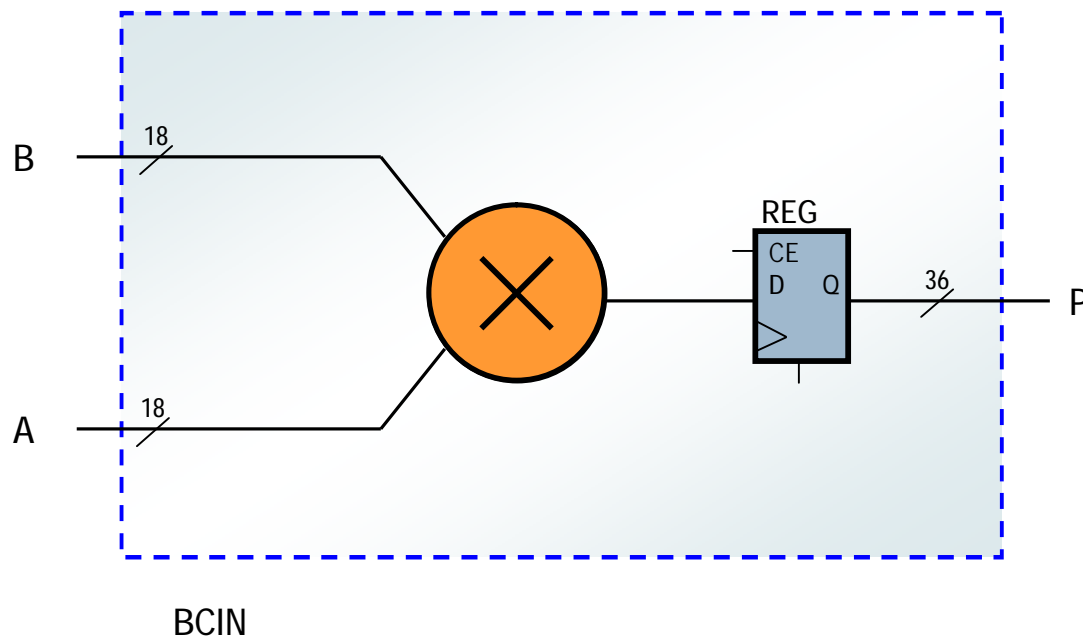
SRL16E 很适合实现一些DSP功能

- 非常有效的实现采样数据的延时
- 高速移位和扫描采样值



内嵌的高性能 DSP

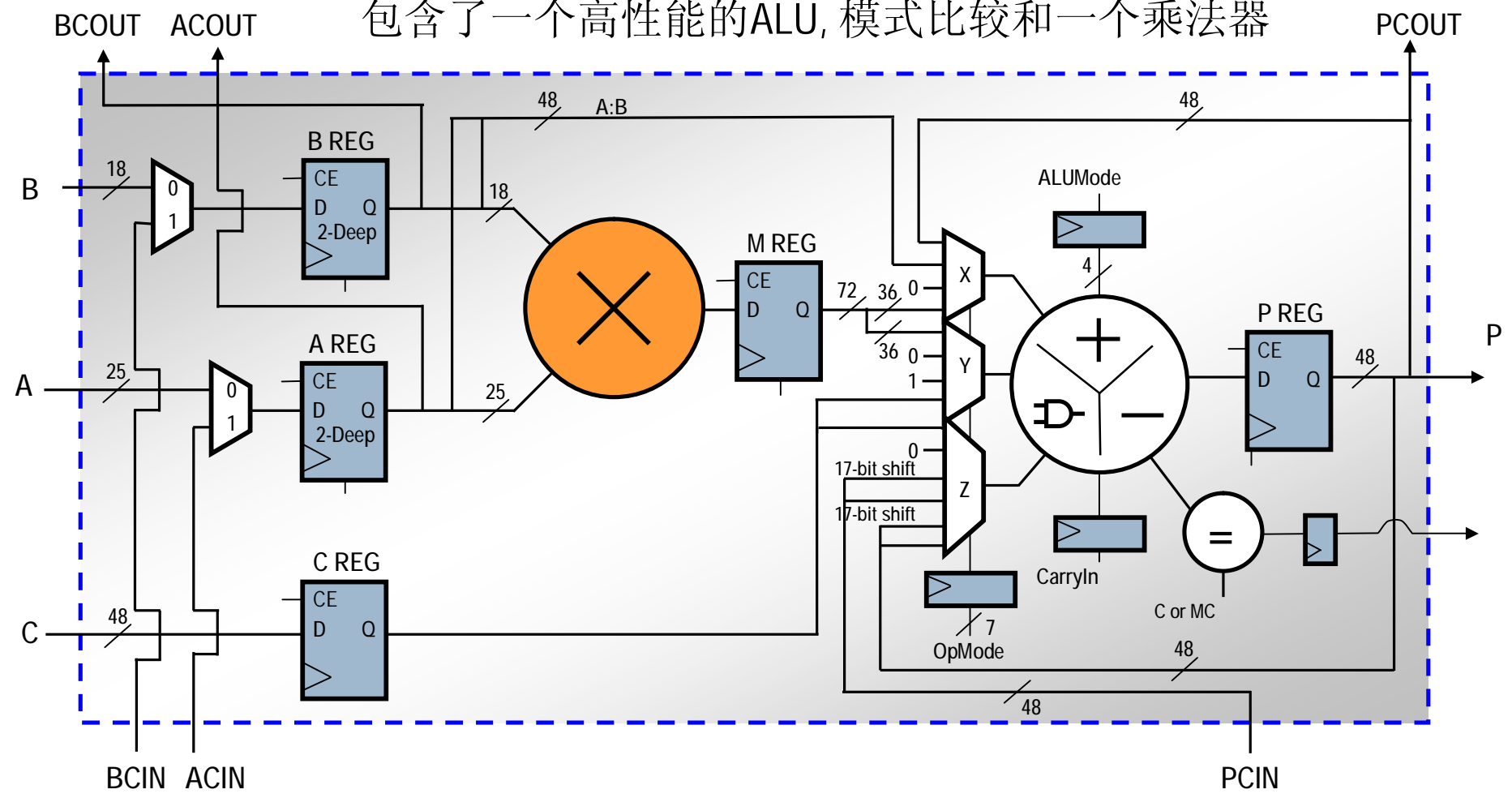
Virtex-II 使用了18x18 乘法器



位于块RAM和CLB 阵列之间，这样能产生高性能的乘-累加操作
乘法器速度的大幅提高，使FPGA的DSP处理能力和性能大幅提高

DSP48E 模块

包含了一个高性能的ALU, 模式比较和一个乘法器



在最低速的器件中也能达到450 MHz 的速度

DSP48E的功能

动态可编程的DSP操作模式

OpMode	Z			Y		X		Output
	6	5	4	3	2	1	0	
Zero	0	0	0	0	0	0	0	+/- Cin
Hold P	0	0	0	0	0	1	0	+/- (P + Cin)
A:B Select	0	0	0	0	0	1	1	+/- (A:B + Cin)
Multiply	0	0	0	0	1	0	1	+/- (A * B + Cin)
C Select	0	0	0	1	1	0	0	+/- (C + Cin)
Feedback Add	0	0	0	1	1	1	0	+/- (C + P + Cin)
36-Bit Adder	0	0	0	1	1	1	1	+/- (A:B + C + Cin)
P Cascade Select	0	0	1	0	0	0	0	PCIN +/- Cin
P Cascade Feedback Add	0	0	1	0	0	1	0	PCIN +/- (P + Cin)
P Cascade Add	0	0	1	0	0	1	1	PCIN +/- (A:B + Cin)
P Cascade Multiply Add	0	0	1	0	1	0	1	PCIN +/- (A * B + Cin)
P Cascade Add	0	0	1	1	1	0	0	PCIN +/- (C + Cin)
P Cascade Feedback Add Ad	0	0	1	1	1	1	0	PCIN +/- (C + P + Cin)
P Cascade Add Add	0	0	1	1	1	1	1	PCIN +/- (A:B + C + Cin)
Hold P	0	1	0	0	0	0	0	P +/- Cin
Double Feedback Add	0	1	0	0	1	0	0	P +/- (P + Cin)
Feedback Add	0	1	0	0	1	1	0	P +/- (A:B + Cin)
Multiply-Accumulate	0	1	0	0	1	0	1	P +/- (A * B + Cin)
Feedback Add	0	1	0	1	1	0	0	P +/- (C + Cin)
Double Feedback Add	0	1	0	1	1	1	0	P +/- (C + P + Cin)
Feedback Add Add	0	1	0	1	1	1	1	P +/- (A:B + C + Cin)
C Select	0	1	1	0	0	0	0	C +/- Cin
Feedback Add	0	1	1	0	0	1	0	C +/- (P + Cin)
36-Bit Adder	0	1	1	0	0	1	1	C +/- (A:B + Cin)
Multiply-Add	0	1	1	0	1	0	1	C +/- (A * B + Cin)
Double	0	1	1	1	1	0	0	C +/- (C + Cin)
Double Add Feedback Add	0	1	1	1	1	1	0	C +/- (C + P + Cin)
Double Add	0	1	1	1	1	1	1	C +/- (A:B + C + Cin)

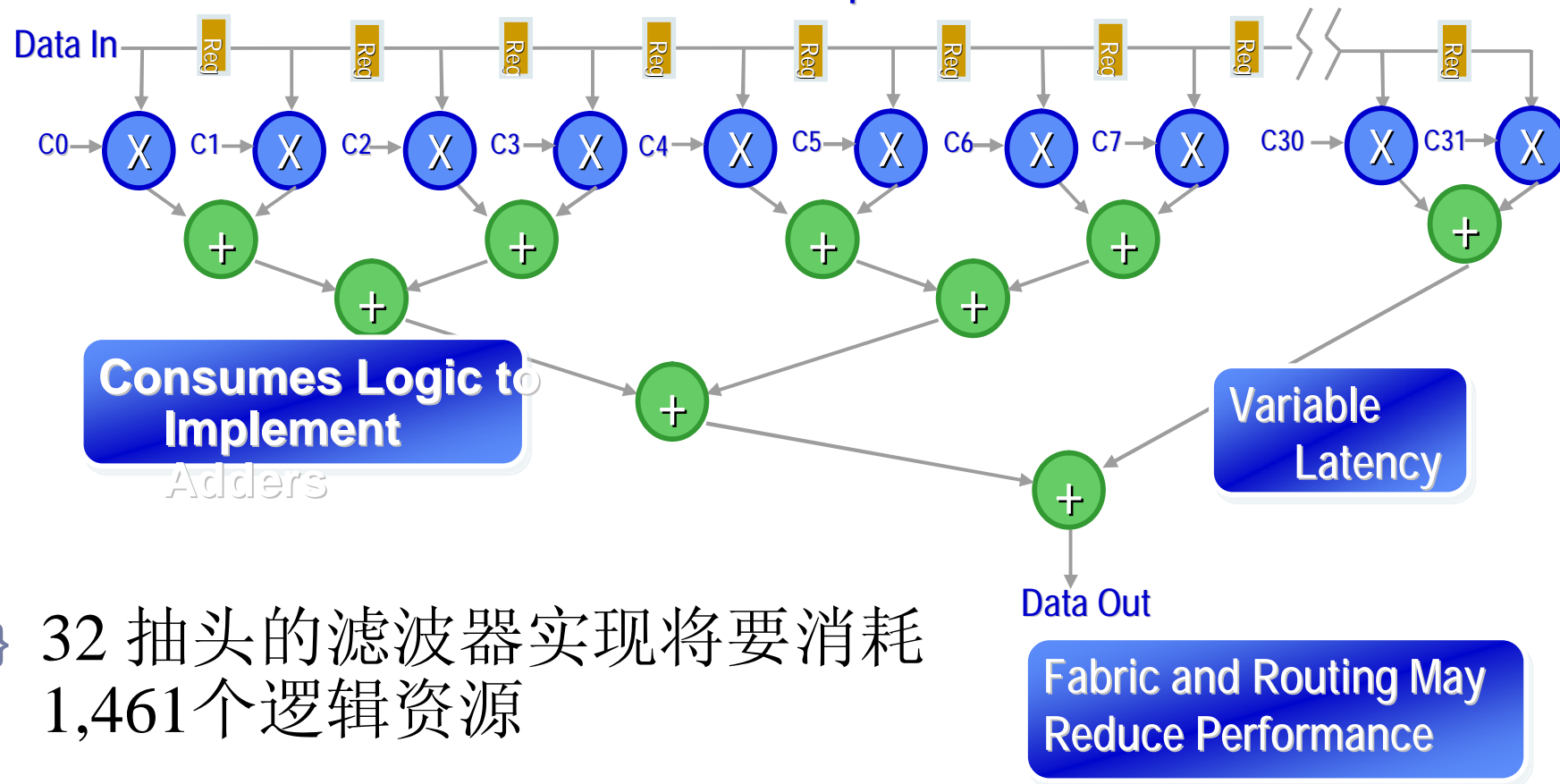
- ✓ 允许DSP的时分复用
- ✓ 超过40种的模式
- ✓ 每一个模块都能独立控制
- ✓ 在一个时钟周期内改变操作
- ✓ 控制功能可由逻辑，存储器或处理器来产生



XtremeDSP块的优势

如果不使用XtremeDSP, 并行的加法树要消耗逻辑资源

Parallel Adder Tree Implementation

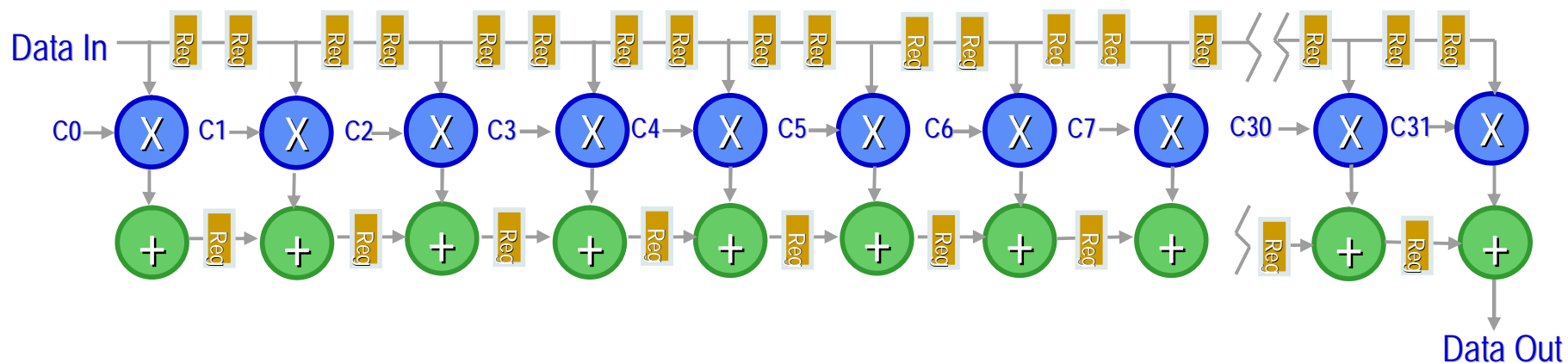


} 32 抽头的滤波器实现将要消耗 1,461 个逻辑资源

XtremeDSP块的优势

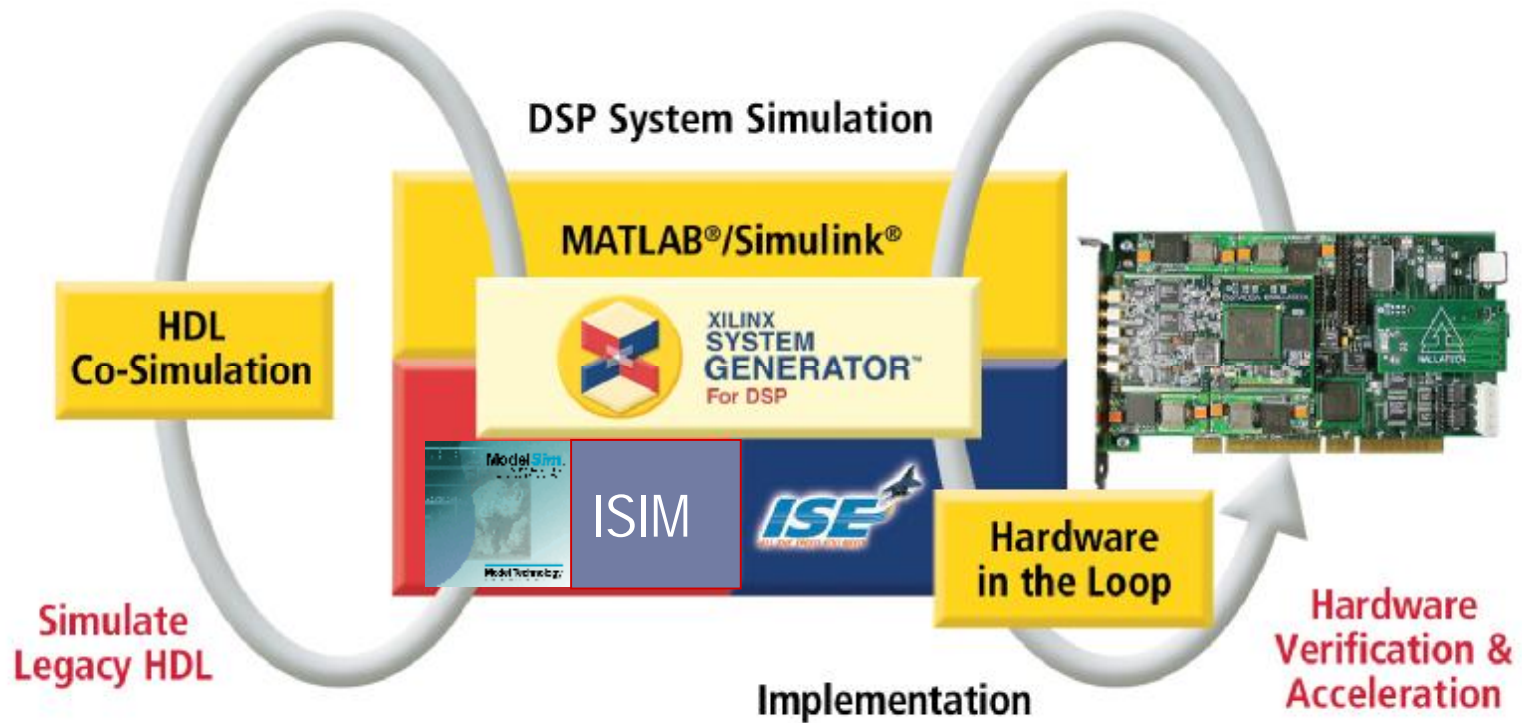
使用XtremeDSP块, 并行的加法树将不消耗逻辑资源

Parallel Adder Cascade Implementation



32 抽头的滤波器完全用XtremeDSP块来实现

System Generator的 DSP设计平台



System Generator概述

- √ FPGA的工业标准的系统级的设计环境(IDE)
 - ∅ 集成的设计流程从Simulink™软件到BIT文件
 - ∅ 整合了现存的技术
 - ü MATLAB , Simulink
 - ü HDL的综合
 - ü IP核的库
 - ü FPGA实现工具
- √ Simulink的算数, 逻辑操作及DSP函数 (Xilinx blockset)库
 - ∅ 满足FPGA实现所需的精度和时序要求
- √ 算法提取
 - ∅ 任意精度的定点,包括量化和溢出
 - ∅ 能进行双精度浮点和定点的仿真

System Generator概述

- } 能生成VHDL和Verilog的代码
 - } 硬件的展开和映射
 - } 可综合的VHDL和Verilog代码，并保留模块的层次结构
 - } VHDL/Verilog的混合语言支持
 - } 自动启动CORE Generator™ 软件来使用IP核
 - } ISE的项目生成可简化设计流程
 - } HDL的测试文件和测试向量的生成
 - } 约束文件 (XCF), 仿真DO文件的生成
 - } HDL的协同仿真
- } 通过硬件协同仿真能极大的提高验证的速度（可通过并口电缆，USB下载线及网口线与硬件相连）

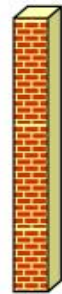
基于图形模块的设计环境



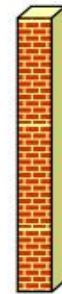
基于图形模型设计的优势



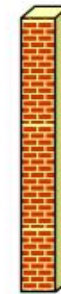
Requirements and Specifications



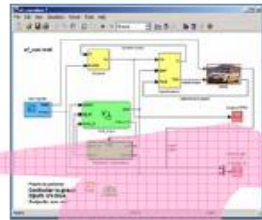
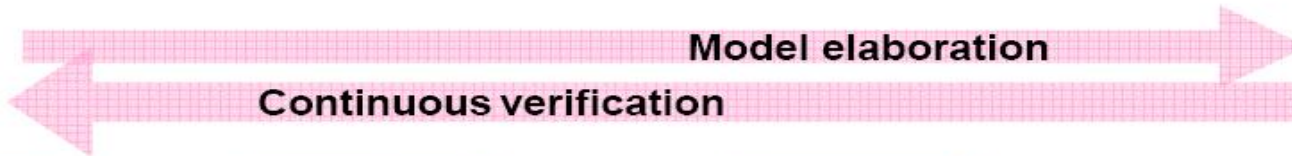
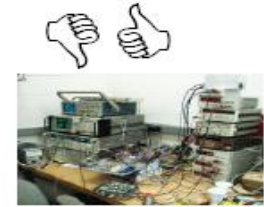
Design



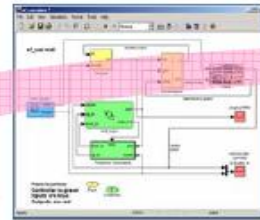
Implementation



Test and Verification



Executable Models



Simulation



Automatic Code Generation

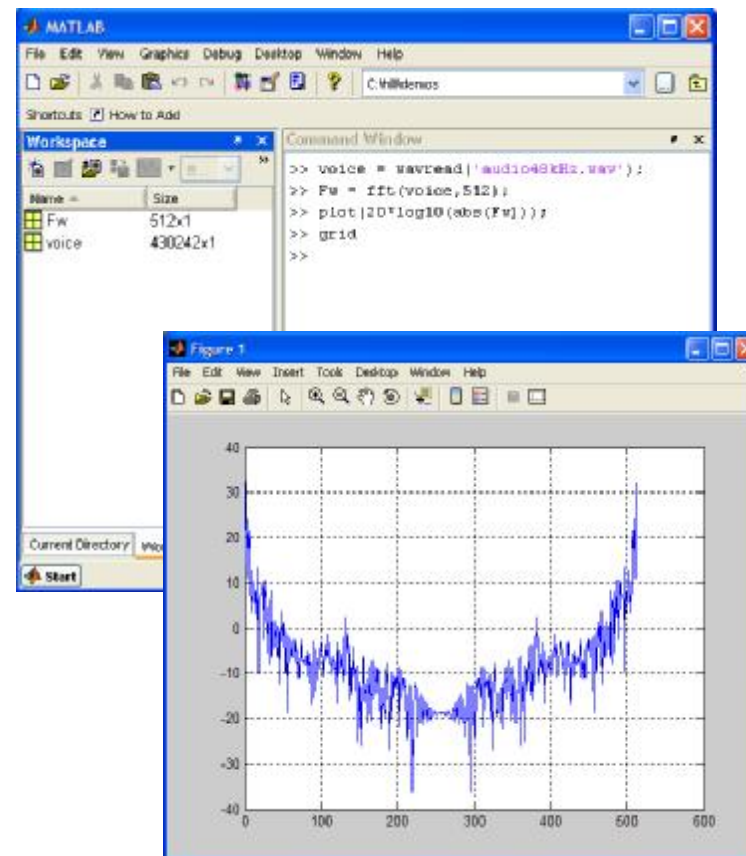


Test with Design

MATLAB



- } MathWorks MATLAB 提供了一个系统级别的快速的算法解决方案
- } 大量的数学公式库，DSP库和通讯库等等
- } 直观：各种图形化的方式来显示整个系统或设计的输出

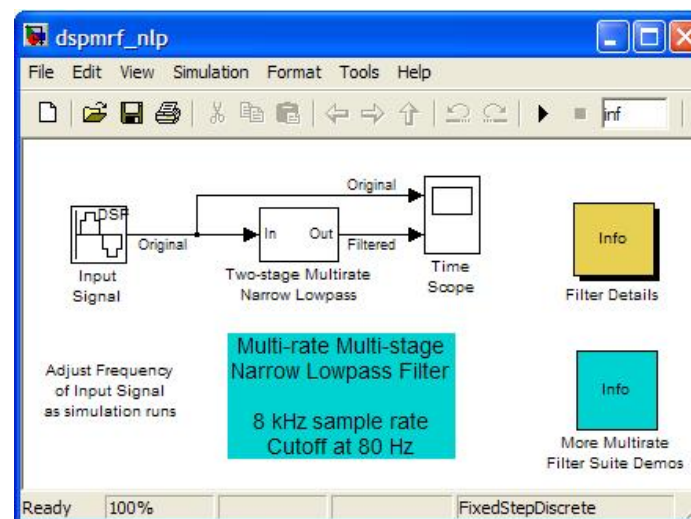


Simulink

直观的数据流工具

} MathWorks Simulink 提供了图形化的设计环境来设计可执行的动态系统

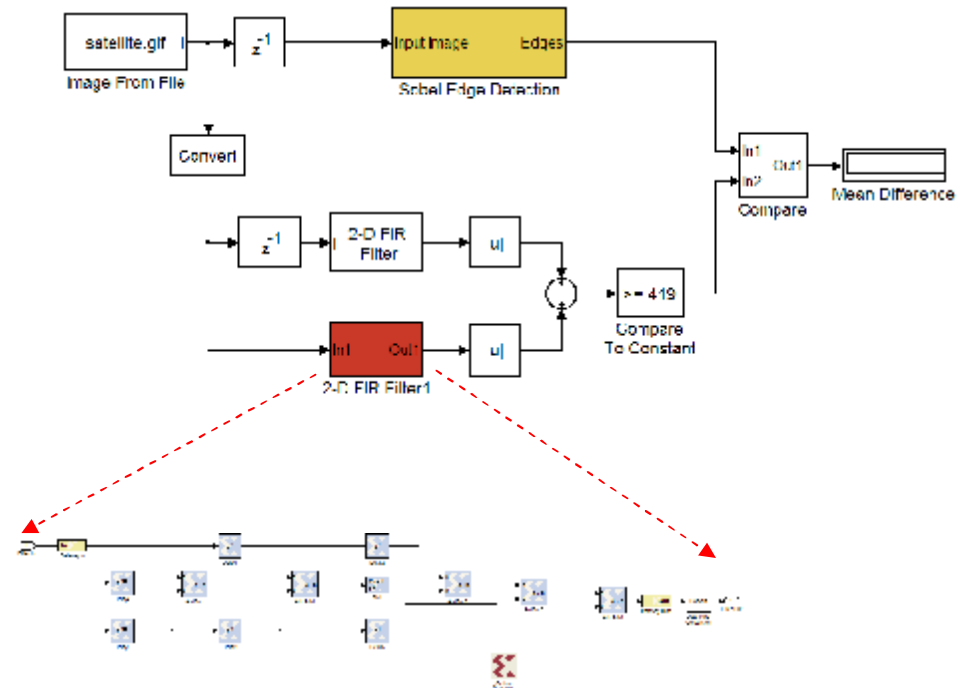
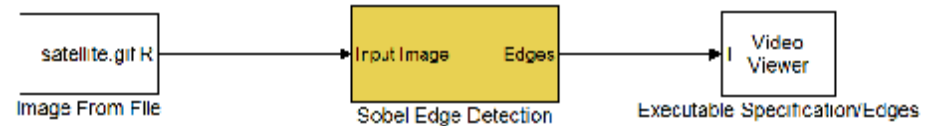
- } 完全集成了MATLAB的引擎
- } 图形化的编辑器
- } 事件驱动的仿真器
- } 模块之间是并行的
- } 大量的参数可变的函数库
 - } Simulink模块集：数学，显示和数据源
 - } DSP模块集：滤波器等
 - } 通讯模块集：调制等



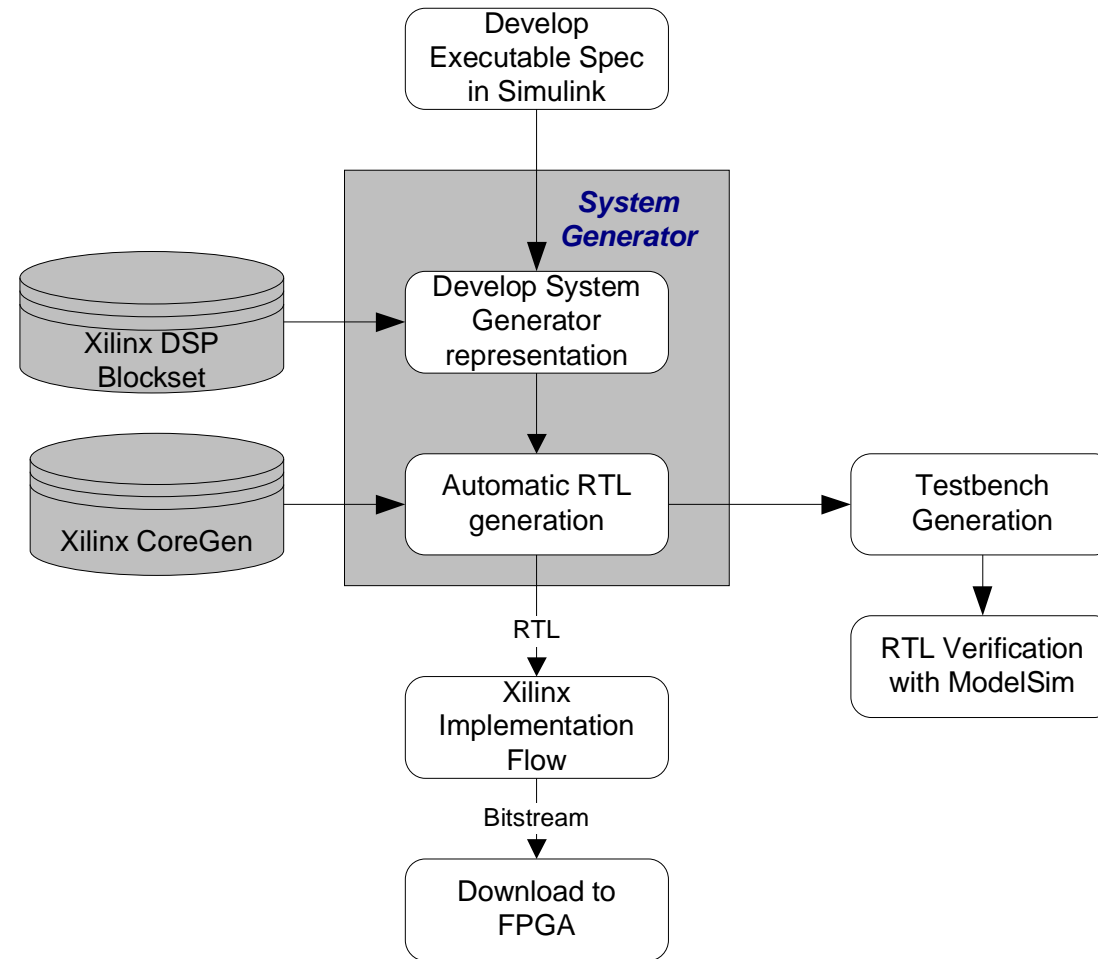
SysGen的设计流程



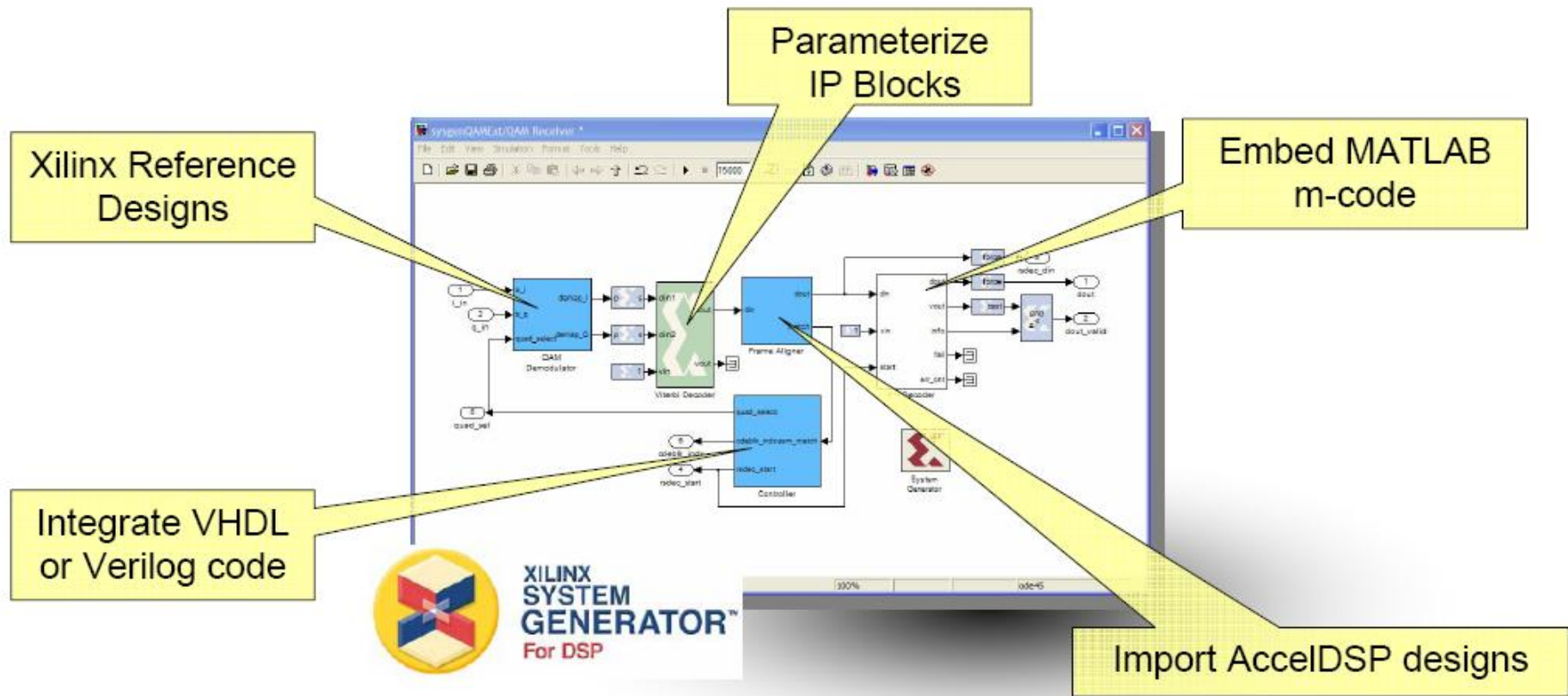
- 使用Simulink来设计可执行的设计规范（SPEC）
- 使用System generator来细化硬件算法
 - 验证硬件算法和可执行的设计规范（SPEC）的差别



SysGen的设计流程

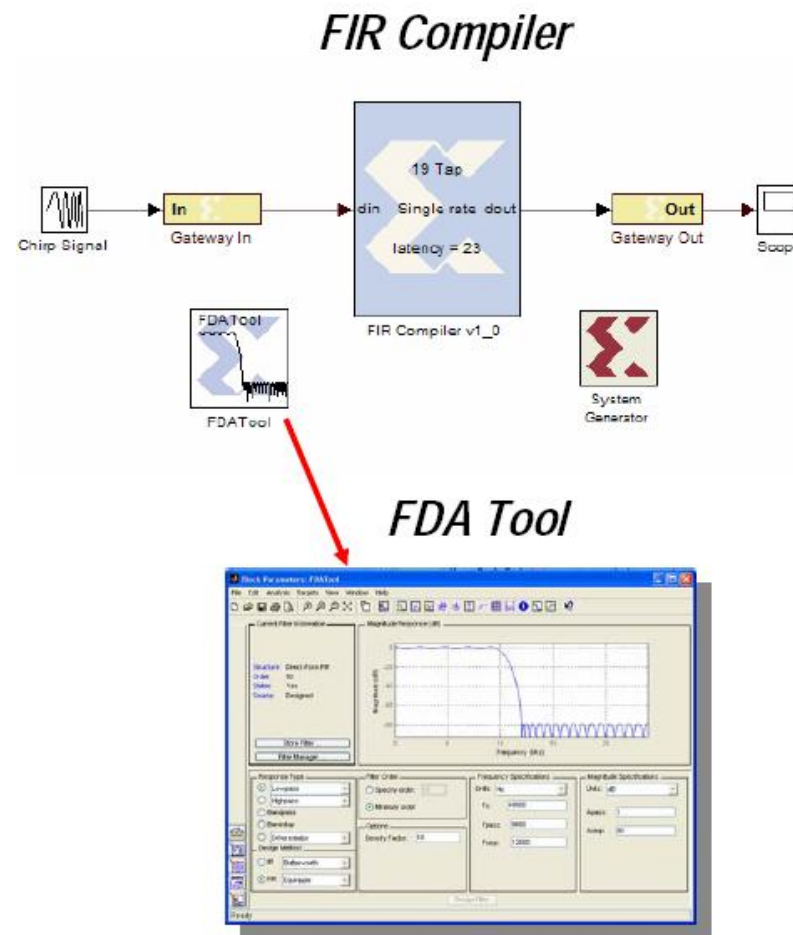


SysGen的集成



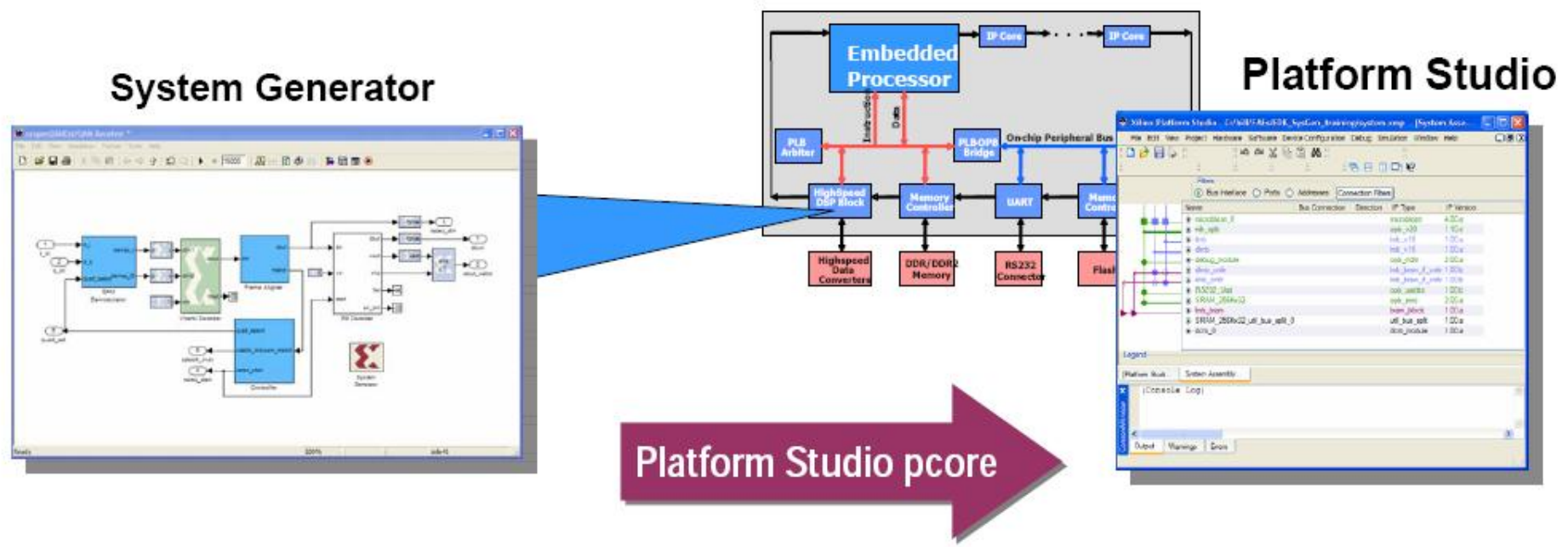
FIR滤波器的生成

- √ FIR编译器能快速生成性能优化的FIR滤波器
 - ∅ 自动实现DSP48的模块以便在V5的芯片中达到550MHz的运算性能
 - ∅ 支持多采样率，过采样，多信道和系数优化等
- √ MathWorks的FDA工具提供了图形化的设计界面和系数生成



嵌入式处理器设计

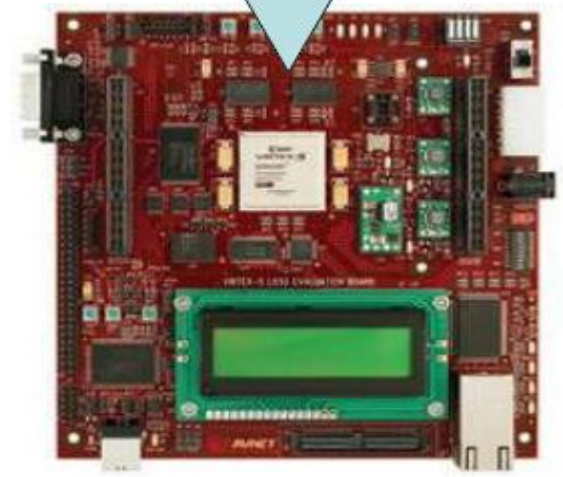
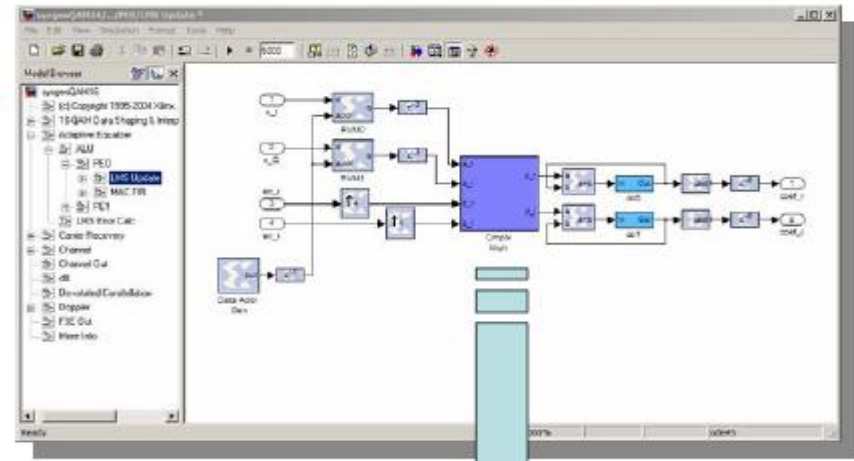
- √ DSP系统能被快速地集成到嵌入式处理器设计中
 - ∅ 集成到Xilinx Platform Studio
 - ∅ 通过一个共享的存储器来简化接口



Platform Studio pcore

硬件协同仿真

- √ 自动化的硬件协同仿真
 Ø 1000倍以上的仿真性能的提高
- √ 能在超过20块的验证板上运行



Q&A



谢谢!

