

基于LabVIEW FPGA 构建高速流盘应用

NIDays
WORLDWIDE GRAPHICAL SYSTEM DESIGN
CONFERENCE
全球图形化系统设计盛会 · 中国站

图形有边
系统无界



高速流盘应用

- 高速内联数字信号处理
 - 射频和通信
 - 航空航天和国防
 - 信号情报（Signal intelligence）
 - 医学成像

被动雷达应用 视频

概述

- 流盘的概念
- LabVIEW FPGA 流盘应用产品
- 案例研究

吞吐量

在一定时间内处理或传输的数据或条目的数量

$\frac{\text{数据}}{\text{时间}}$

- 字节/秒
- 位/秒



$\frac{\text{条目}}{\text{时间}}$

- 样本/秒
- 像素/秒
- 帧/秒



$\frac{\text{操作}}{\text{时间}}$

- FLOP
- FFT/秒

带宽

一个通道或处理单元可获取的最大吞吐量。

影响实际带宽的因素

- 需要根据情况采取特定的使用或者访问模式，以取得最高的带宽利用率
- 在应用层面上，并不是所有的带宽都能可用

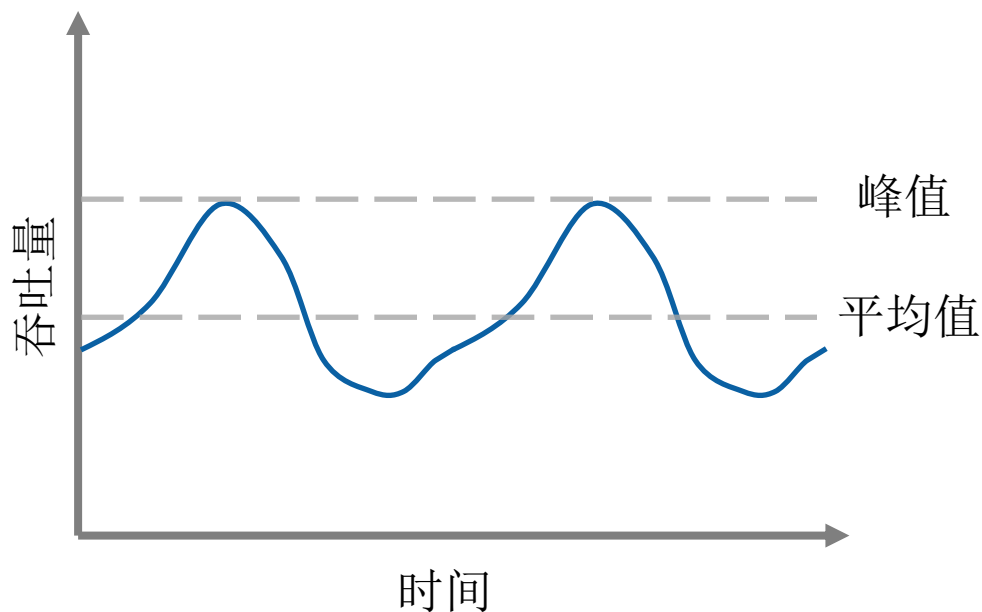
平均吞吐量vs.瞬时吞吐量

- 平均

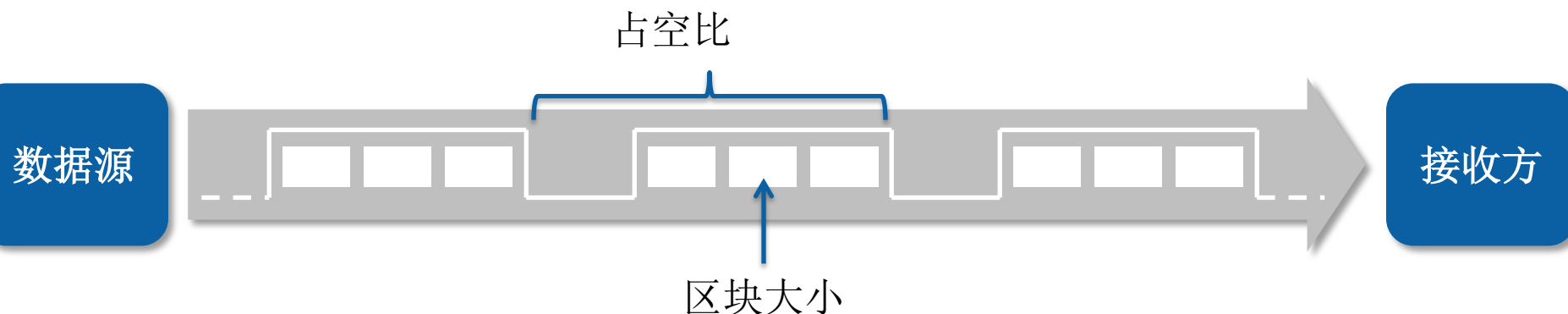
- 可持续性
- 经历任意的一段时间

- 瞬时

- 突发性数据
- 峰值吞吐量

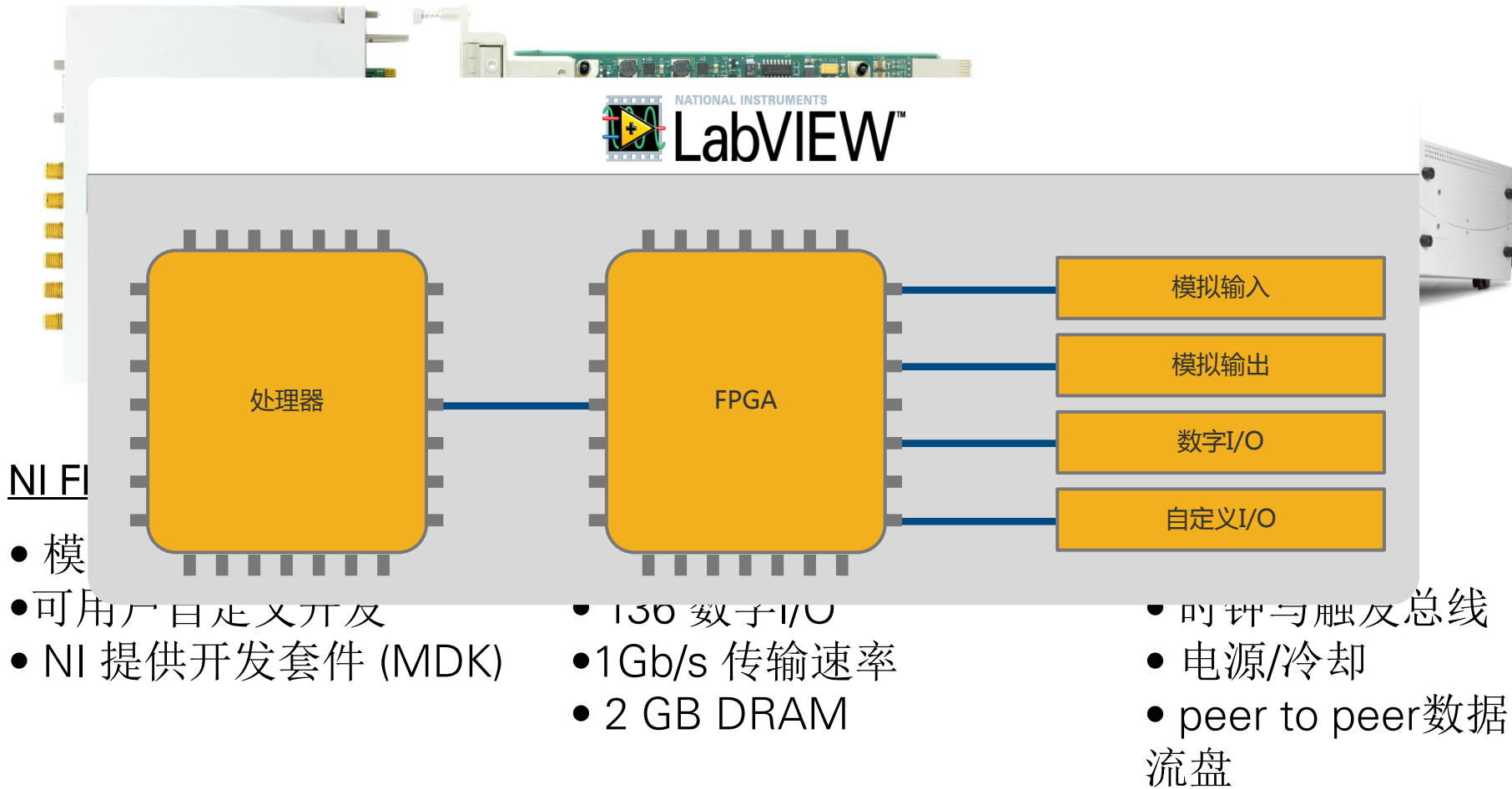


平均吞吐量vs.瞬时吞吐量

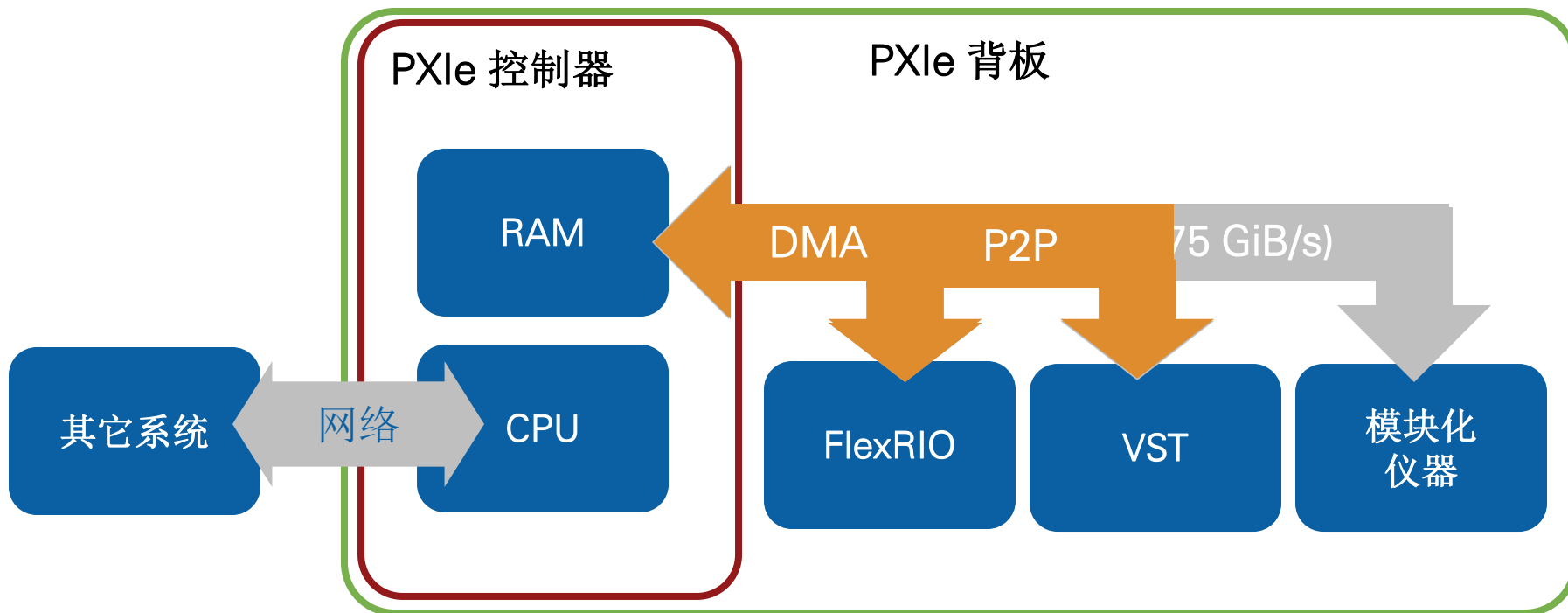


- 占空比(开/关)
 - 传输可能是周期性的
 - $\text{平均吞吐量} = \text{瞬时吞吐量} \times \text{占空比}$
- 区块大小
 - 数据源提供区块大小整数倍的数据
- 接收方可以
 - 处理瞬时吞吐量。
 - 或者：使用缓存，并处理平均吞吐量

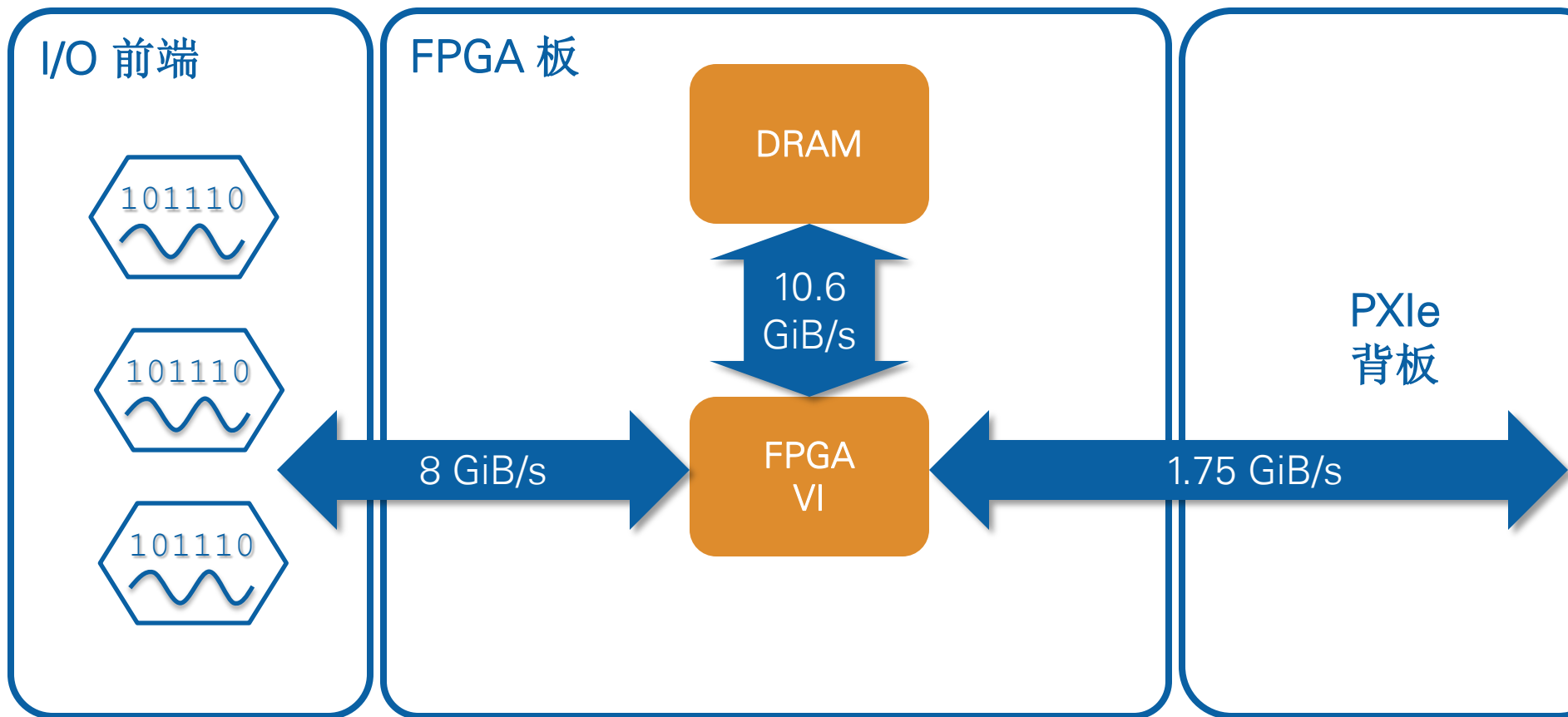
NI RIO架构 与 NI FlexRIO



系统数据流盘的机制

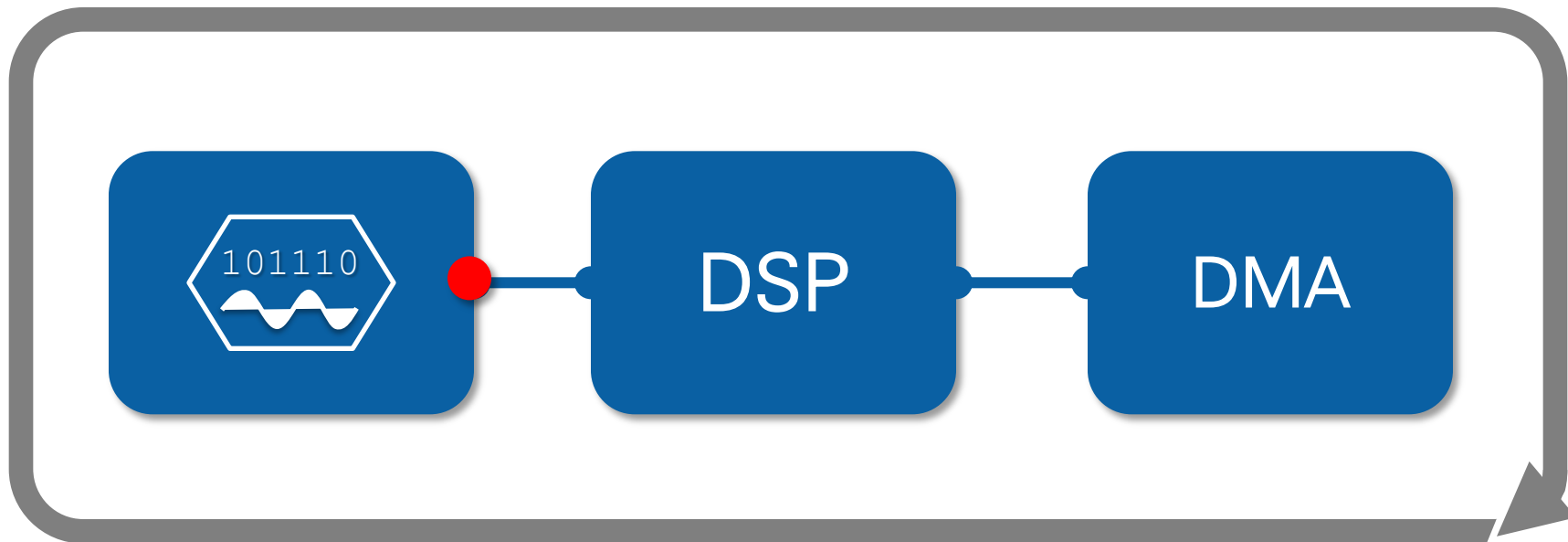


FlexRIO 和 MI 流盘机制

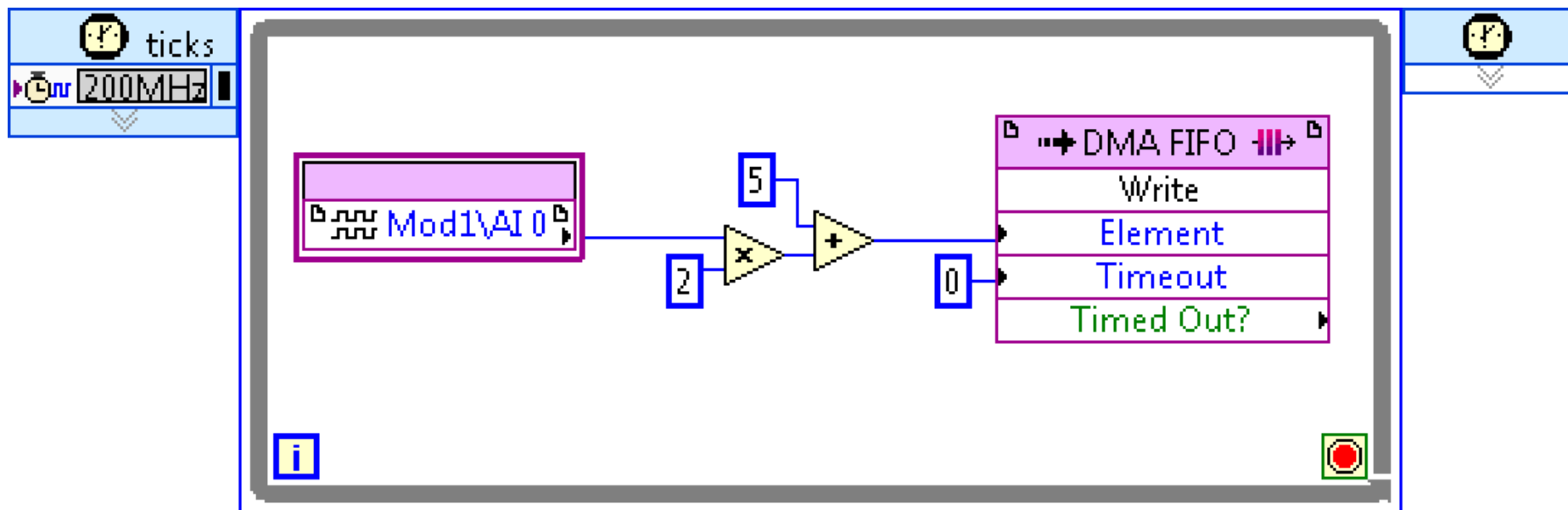


FPGA VI 的数据传输机制

受限的资源 → 逐点处理
小型的、固定大小的数组 → 并行处理

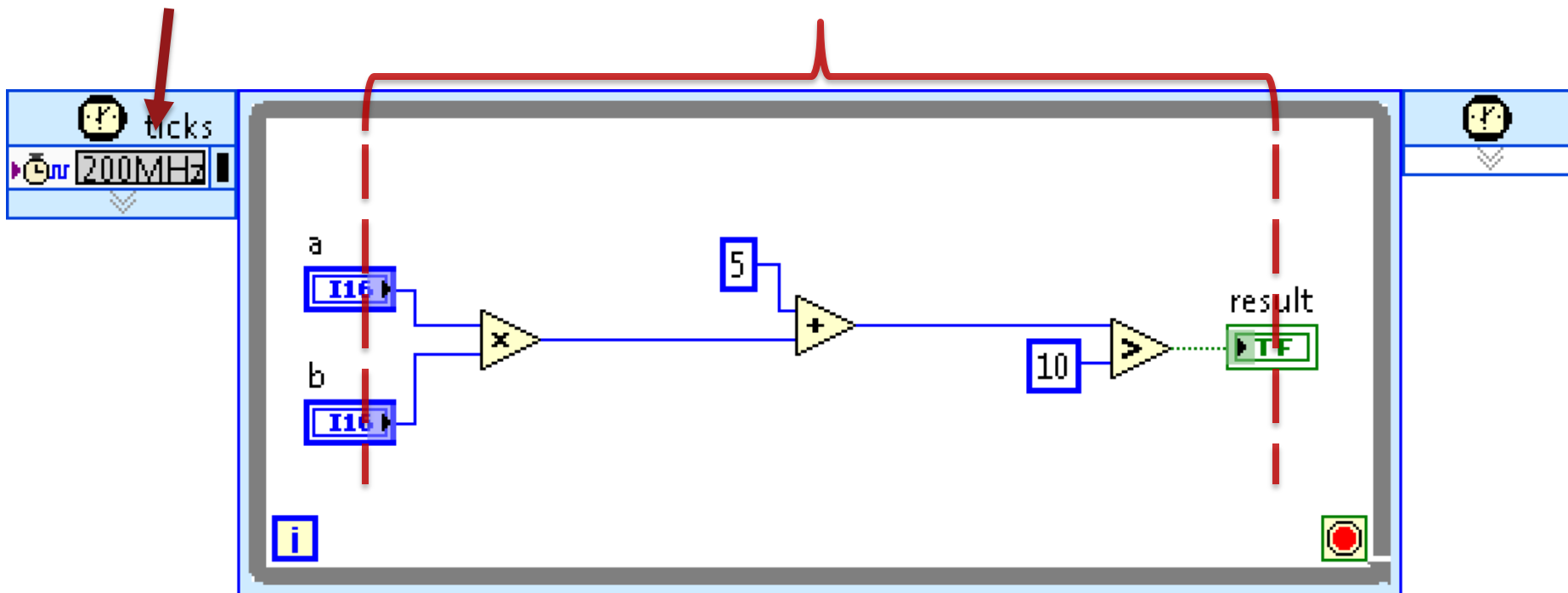


单周期定时循环

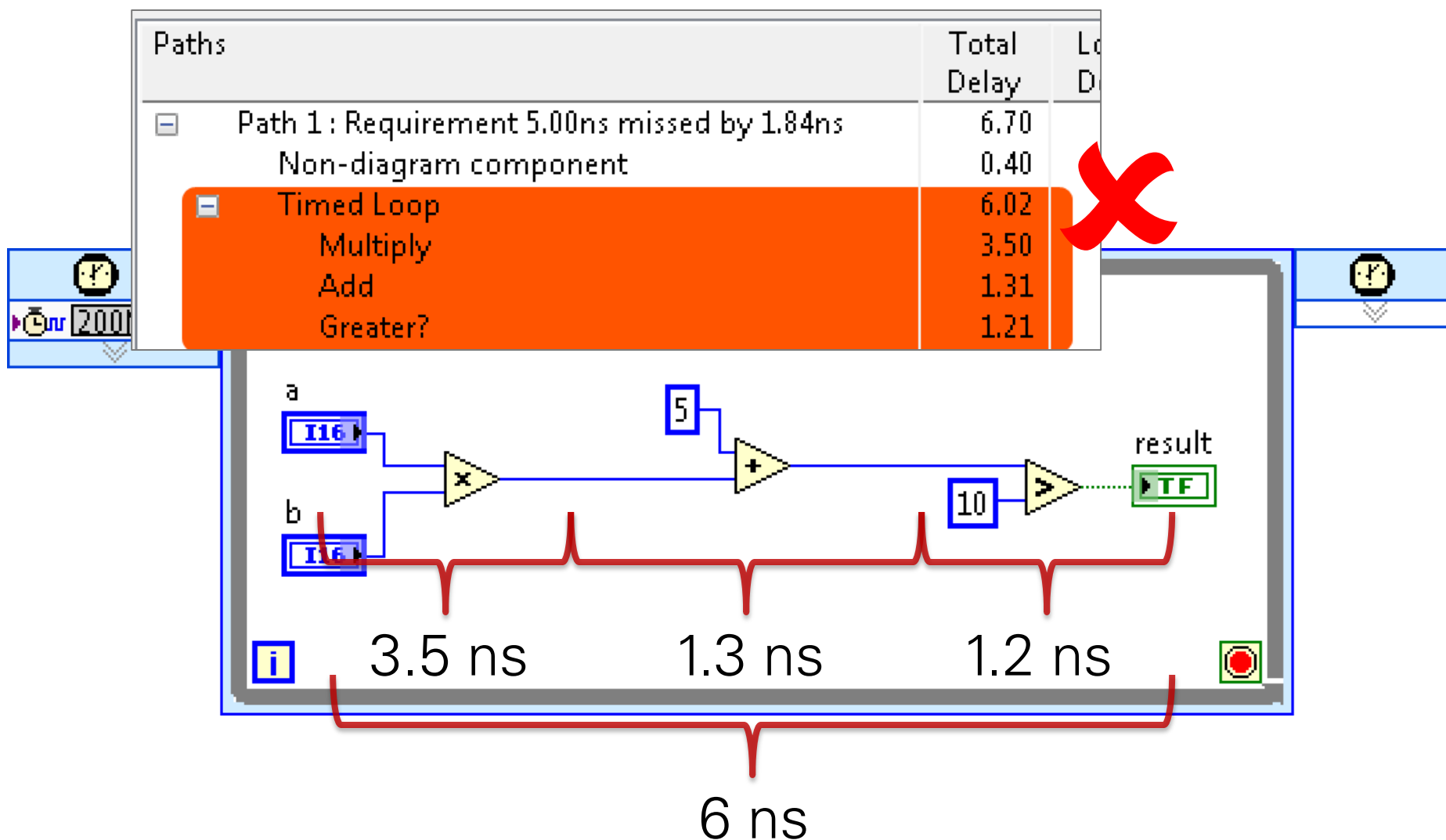


循环延时的限制

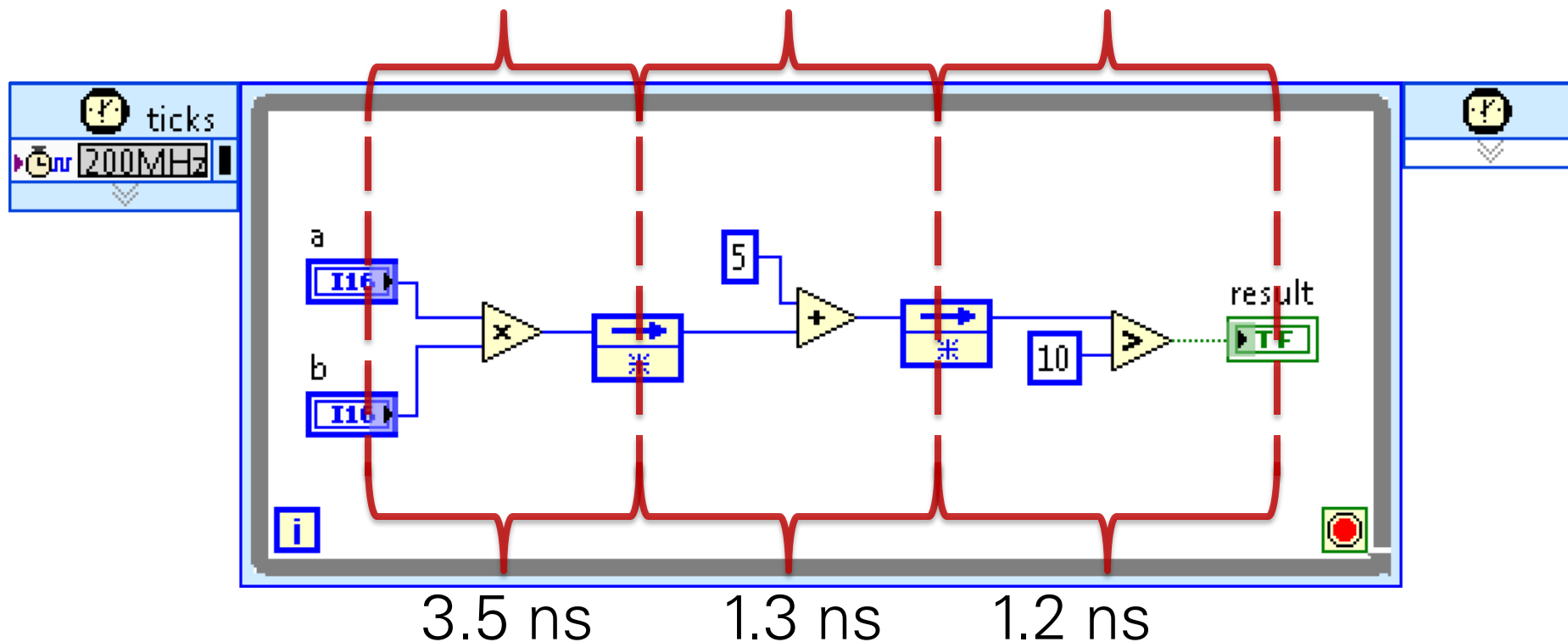
200 MHz \rightarrow < 5 ns



时序错误

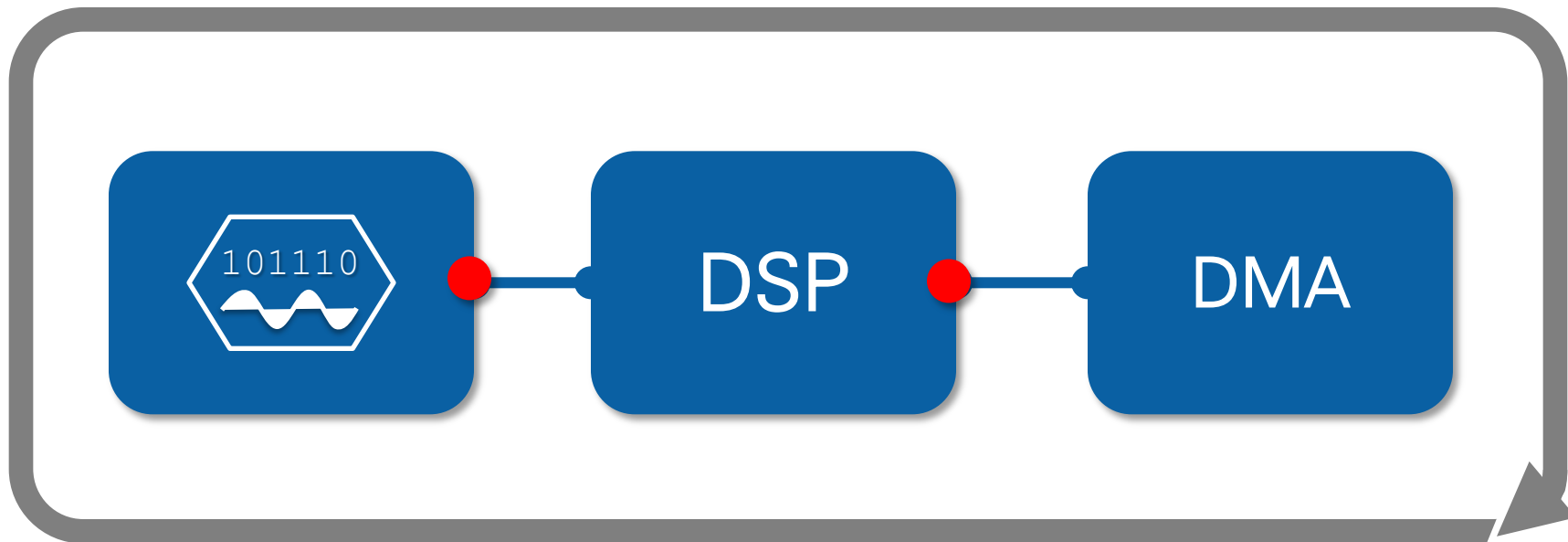


通过流水线获得更高的时钟速度

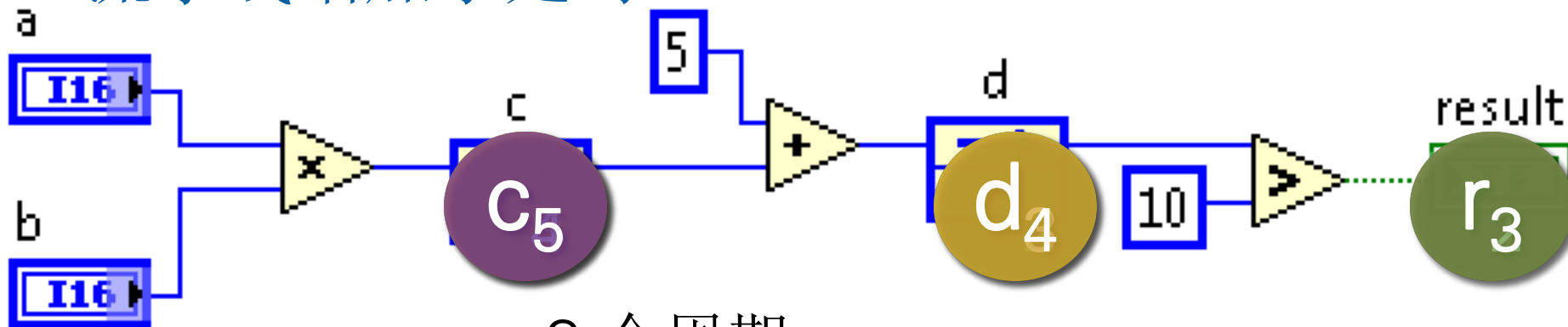


FPGA VI – 流水线型数据传输机制

流水线型的数据传输机制设计可以让操作同步进行



流水线增加了延时



3 个周期

c	$c_1=a_1*b_1$	$c_2=a_2*b_2$	$c_3=a_3*b_3$	$c_4=a_4*b_4$	$c_5=a_5*b_5$
d	无效	$d_1=5+c_1$	$d_2=5+c_2$	$d_3=5+c_3$	$d_4=5+c_4$
Result	无效	无效	$r_1=d_1>10$	$r_2=d_2>10$	$r_3=d_3>10$

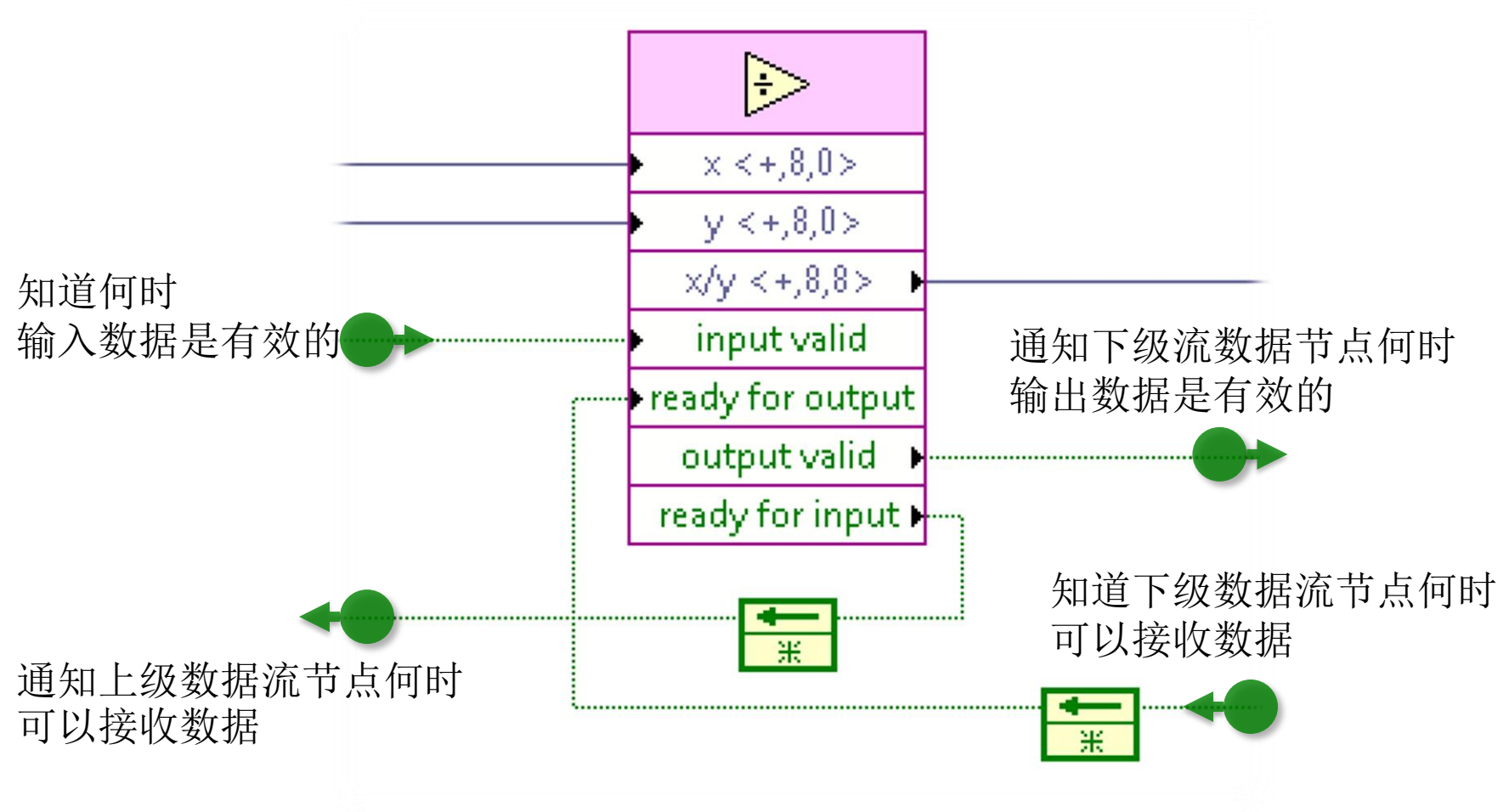
多周期IP和单周期定时循环

初始化流水线时会产生无效数据

同时，

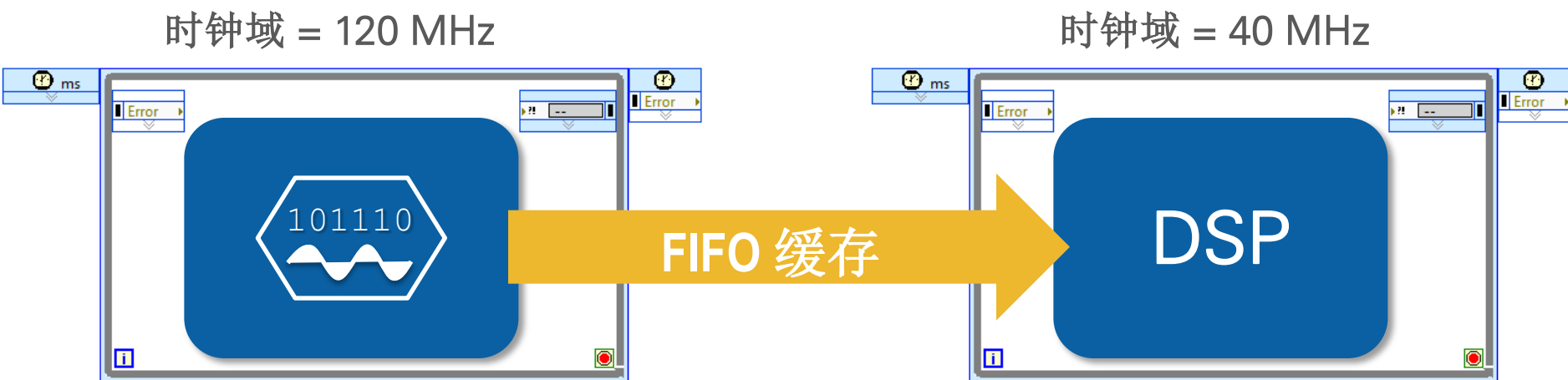
- IP可能需要多个周期完成这项工作
- 但是每个周期必须返回些什么...
- 需要在程序框图中识别数据的有效性

4线握手



多循环速率设计

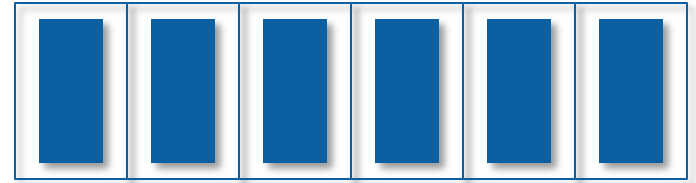
- 不是所有的设备都可以在同样的速率下运行
- 不是所有的设备都需要在同样的速率下运行
 - 更低的时钟速率可以让编译更简单，并提高了资源的利用率



对于存储的考虑

- 空间需求

- 元素的数目
- 元素数据的宽度



- 访问需求

- 顺序: 顺序访问vs.随机访问
- 延时: 单周期访问vs.多周期访问
- 元素数量: 单个元素vs.多个元素

选择存储/通信方式

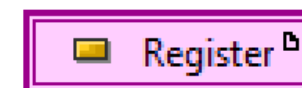
单个元素

- 使用输入控件或者显示控件 用于与主机进行数据的发送/接收
- 本地或全局变量 用于在FPGA内存储或传输数据
- 寄存器或握手 用于在循环之间传递的节点

多个元素

- FIFO 用于在FPGA的不同位置内连续的传递数据
- FIFO 用于通过DMA向主机传递数据
- Memory 用于存储FPGA随机访问的数据
- DRAM 用于NI FlexRIO内的高速存储，容量高达2GB

Control

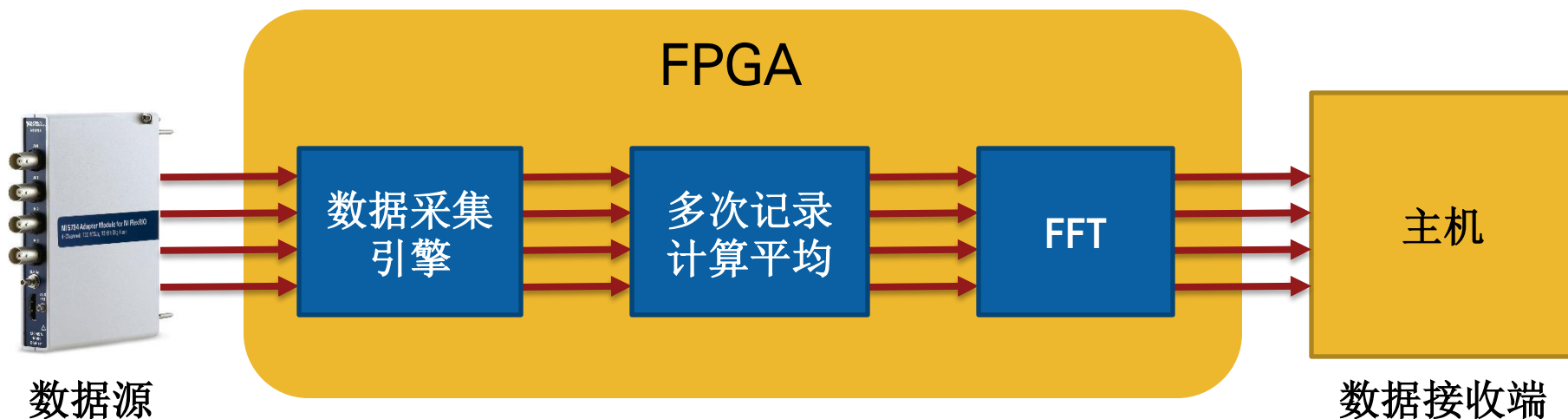


案例分析

使用我们已经学习过的概念

案例研究: 概述

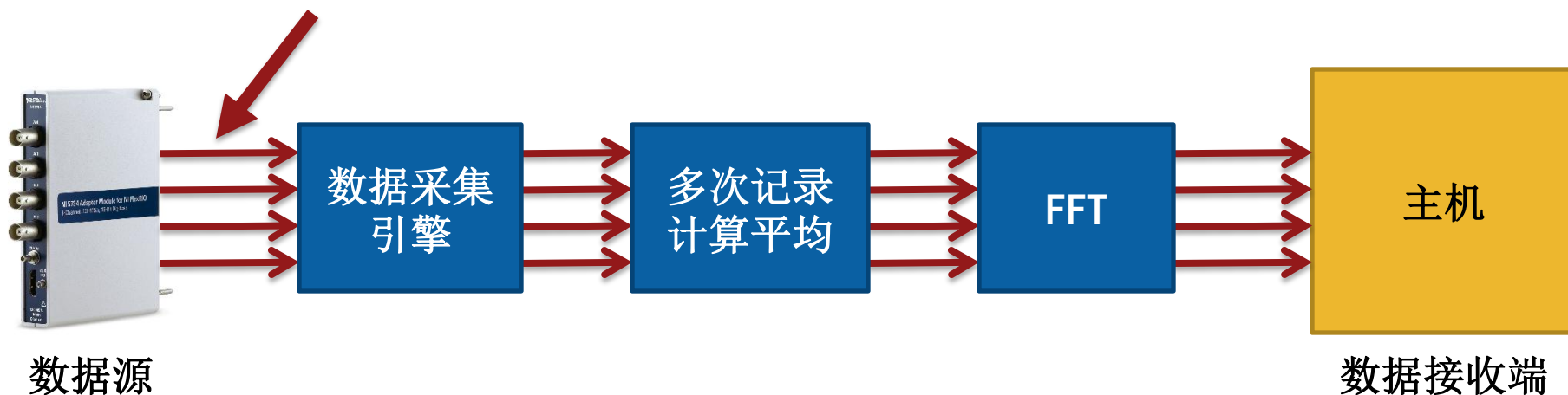
- 目标: 搭建能够满足下面要求的架构:
 1. 从NI 5734(4通道, 每个记录中包含1024个样本)中获取样本
 2. 为多次记录计算平均值(每次计算处理256次记录)
 3. 为经过平均的波形计算FFT
 4. 将FFT的结果发送给主机或显示设备



案例分析: 分析吞吐量

适配器模块为FPGA生成连续的数据流

- 占空比: 100%
- 瞬时吞吐量: $16\text{位} \times 4\text{通道} \times 120\text{MHz} = 960\text{MB/s}$
- 平均吞吐量: $960\text{MB/s} \times 100\% = 960\text{MB/s}$

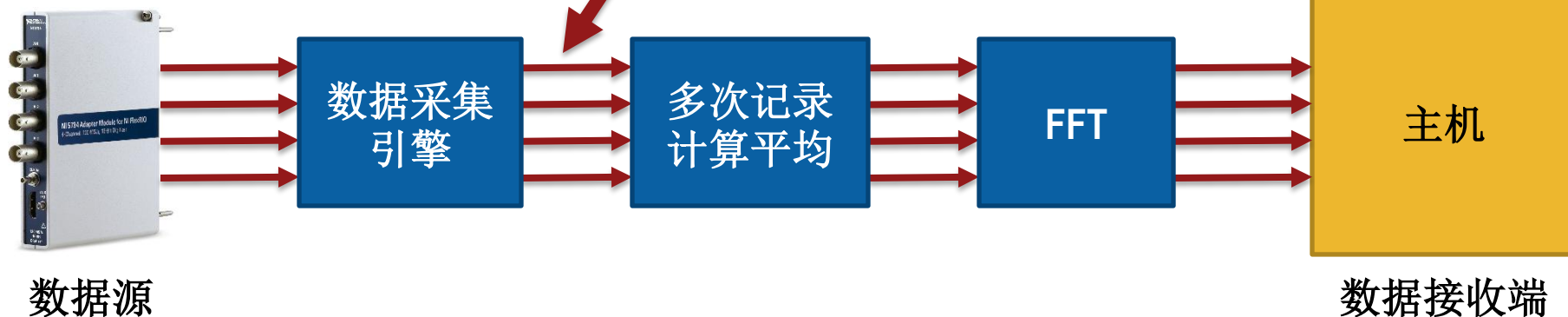


案例分析: 分析吞吐量

数据采集引擎之外的吞吐量会根据波形变化

获取: 波形的上升沿触发.

- 区块大小: 1024个样本
- 占空比: 根据波形变化; $\leq 100\%$
- 瞬时吞吐量: $16\text{位} \times 4\text{通道} \times 120\text{MHz} = 960\text{MB/s}$
- 平均吞吐量: $960\text{MB/s} \times \text{占空比} \leq 960\text{MB/s}$



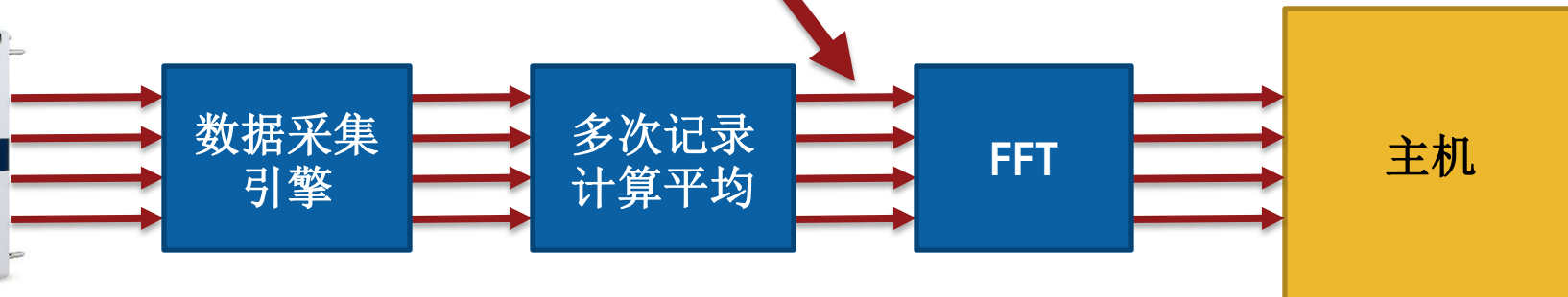
案例分析: 分析吞吐量

多次记录计算平均单元之后的吞吐量取决于上级流数据的吞吐量

- 区块大小: 1024个样本
- 占空比: $\leq 100\% \div 256$ (对每个记录进行平均计算) = 0.39%
- 瞬时吞吐量: $16\text{位} \times 4\text{通道} \times 120\text{MHz} = 960\text{MB/s}$
- 平均吞吐量: $960\text{MB/s} \times 0.39\% = 3.75\text{ MB/s}$



数据源

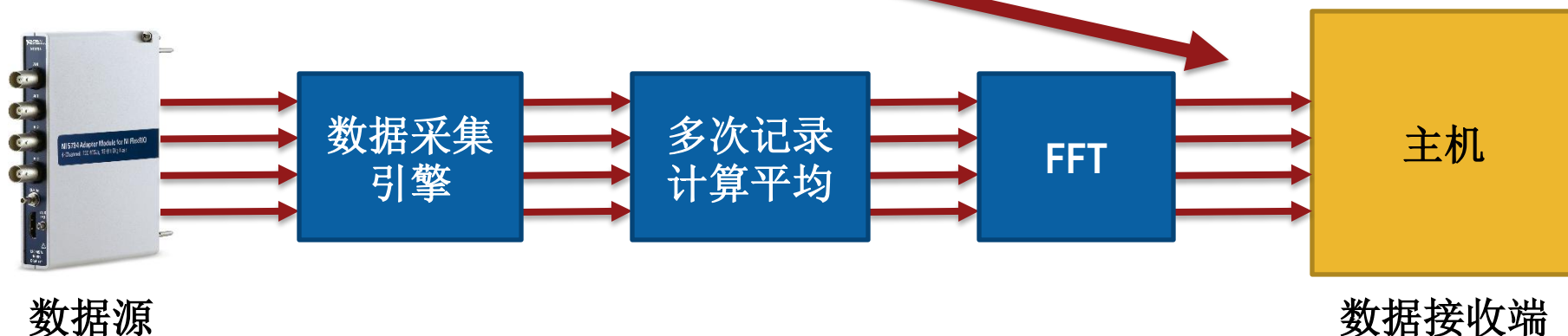


数据接收端

案例分析: 分析吞吐量

FFT操作将数据带宽增加到23位，
并引入了几个周期的延时

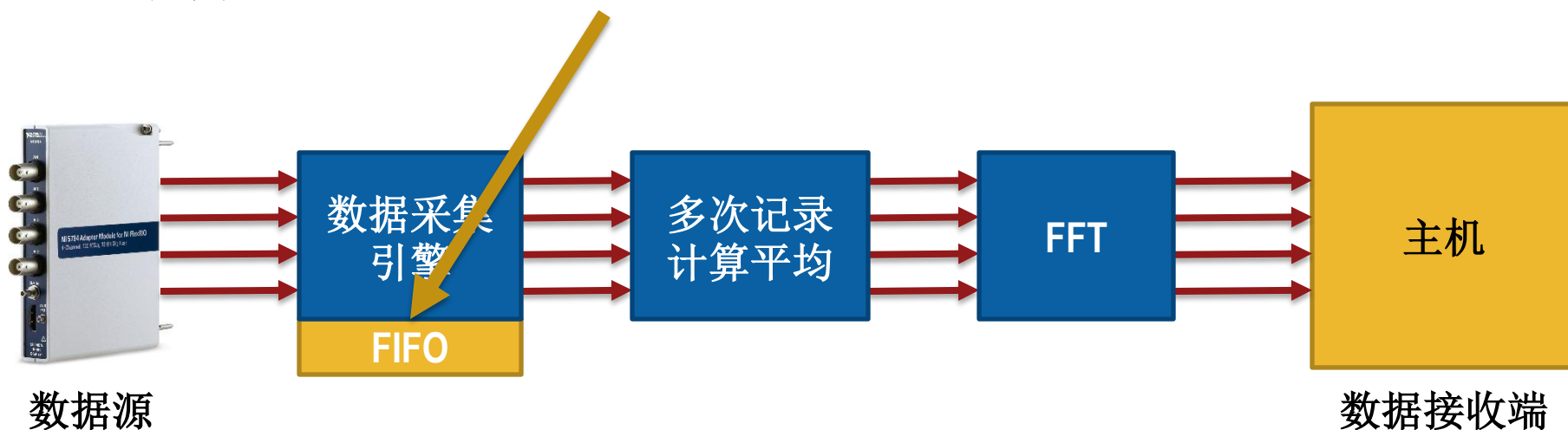
- 区块大小: 1024个样本
- 占空比: 0.39%
- 瞬时吞吐量: $23\text{位} \times 4\text{通道} \times 120\text{MHz} = 1.38\text{GB/s}$
- 平均吞吐量: $1.38\text{GB/s} \times 0.39\% = 5.38\text{MB/s}$



案例分析: 分析存储

数据采集引擎使用FIFO缓存触发前和触发后的数据

- 数据带宽: $16 \text{ 位} \times 4 \text{ 通道} = 8 \text{ 字节}$
- 元素数目: 预期的触发前采样数 (128个样本, 1KB)
- 应用: 4个并行块RAM FIFO



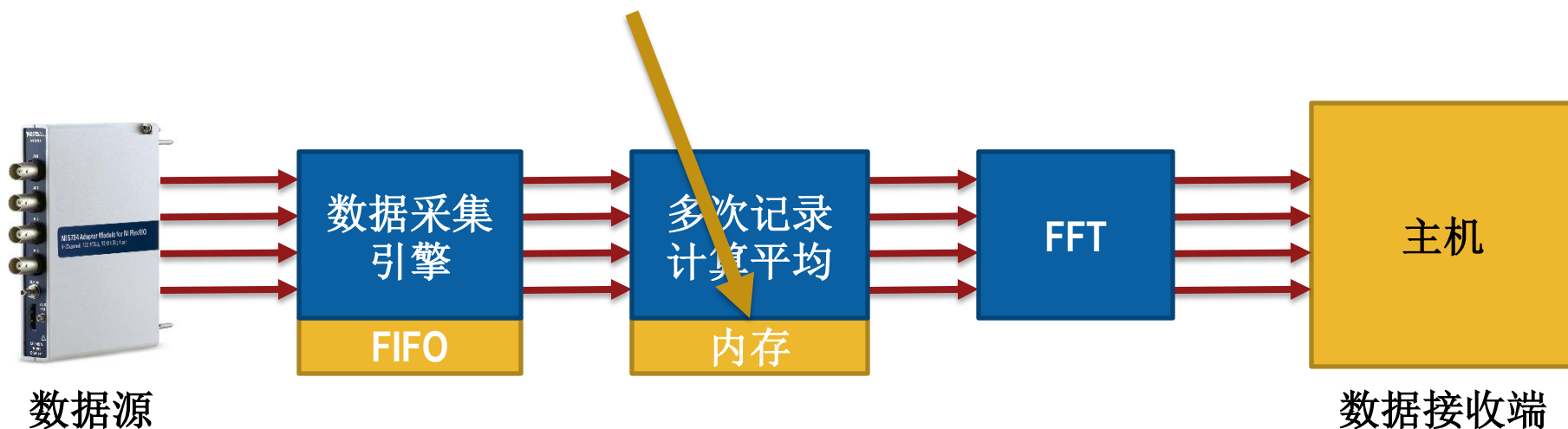
数据源

数据接收端

案例分析: 分析存储

若要对多记录进行平均计算，则需要使用内存来收集多个记录。

- 数据带宽: $32 \text{ 位} \times 4 \text{ 通道} = 16 \text{ 字节}$
- 元素数目: 记录的长度(1024个样本, 16KB)
- 应用: 4 个并行块RAM内存



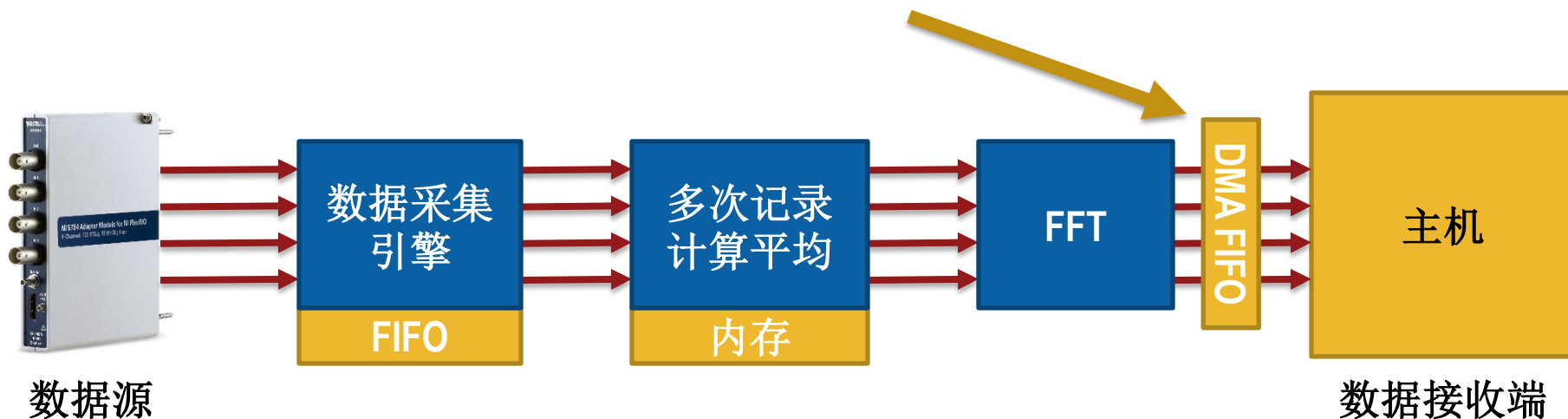
数据源

数据接收端

案例分析: 分析存储

使用DMA FIFO 来缓存被传输到主机的数据，避免总线错误引起的数据丢失。

- 数据带宽: $32 \text{ 位} \times 4 \text{ 通道} = 16 \text{ 字节}$
- 元素个数: 记录的长度(1024个样本, 16KB)
- 方法: 4 个并行块RAM FIFO



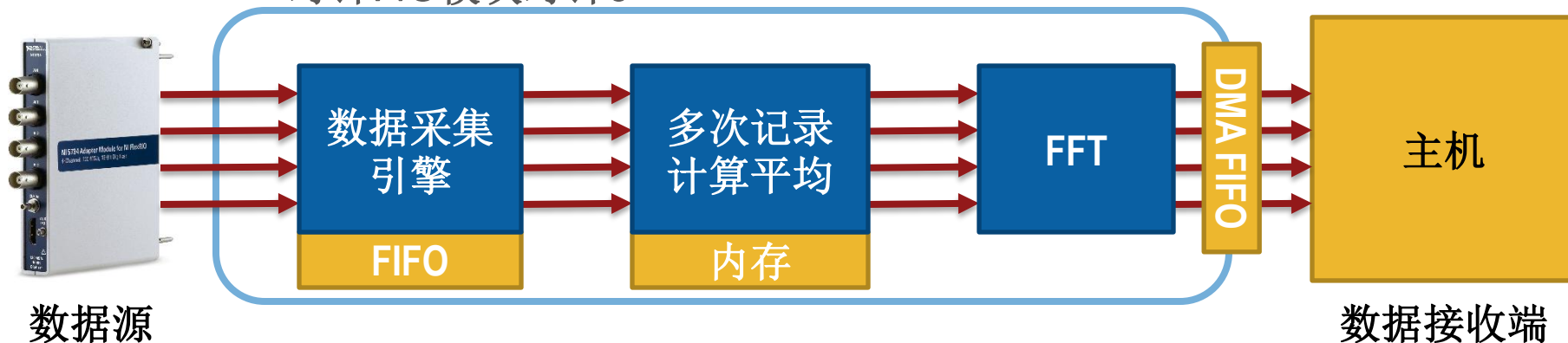
案例分析: 时钟架构

NI 5734以120MS/s的速度进行采样，并为IO模块提供120MHz时钟。

为模块的各种操作提供120MHz时钟。



时钟: IO模块时钟0

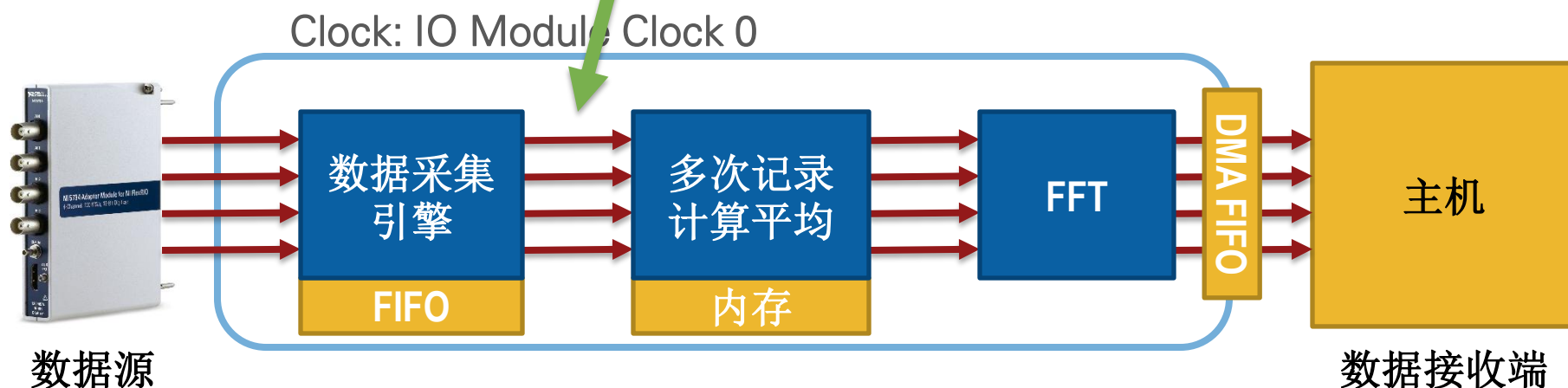


案例分析: 通信策略

数据采集引擎和多次记录计算平均单元随时准备收取新的数据。

可以使用一种2线协议作为握手协议:

- 数据有效输入
- 数据有效输出

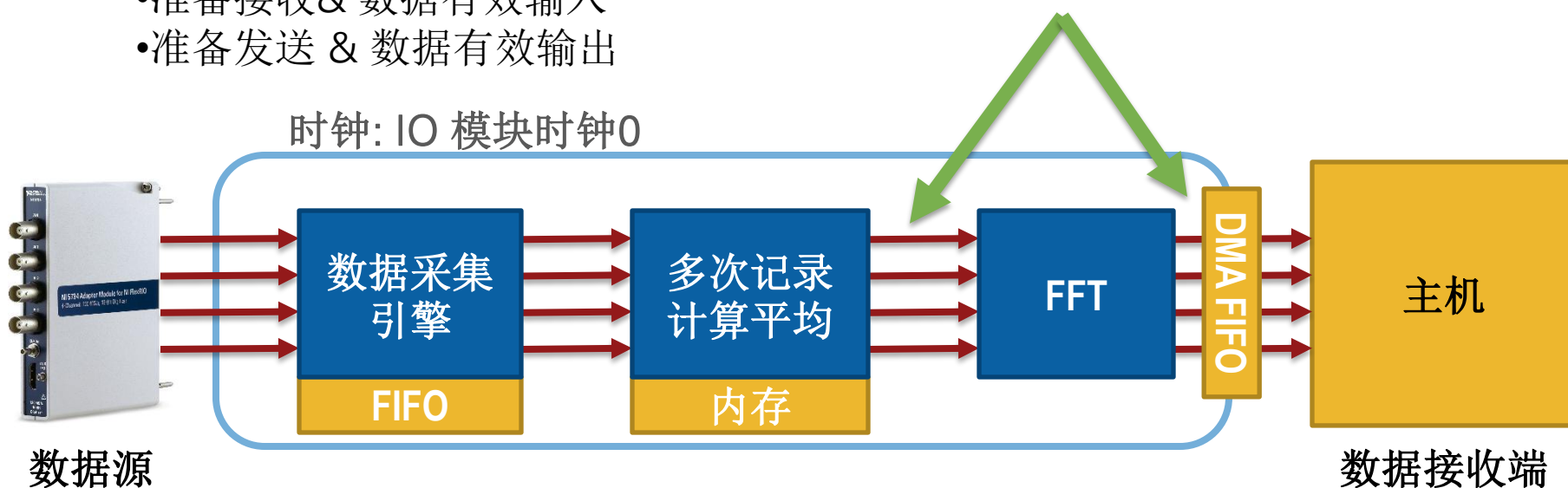


案例分析: 通信策略

FFT单元和DMA FIFO可能不需要随时准备接收数据, 这些数据可以由平均器的内存进行缓存。

使用4线协议控制多次记录计算平均单元、FFT和DMA FIFO之间的数据流

- 准备接收& 数据有效输入
- 准备发送 & 数据有效输出



案例分析: 并行应用

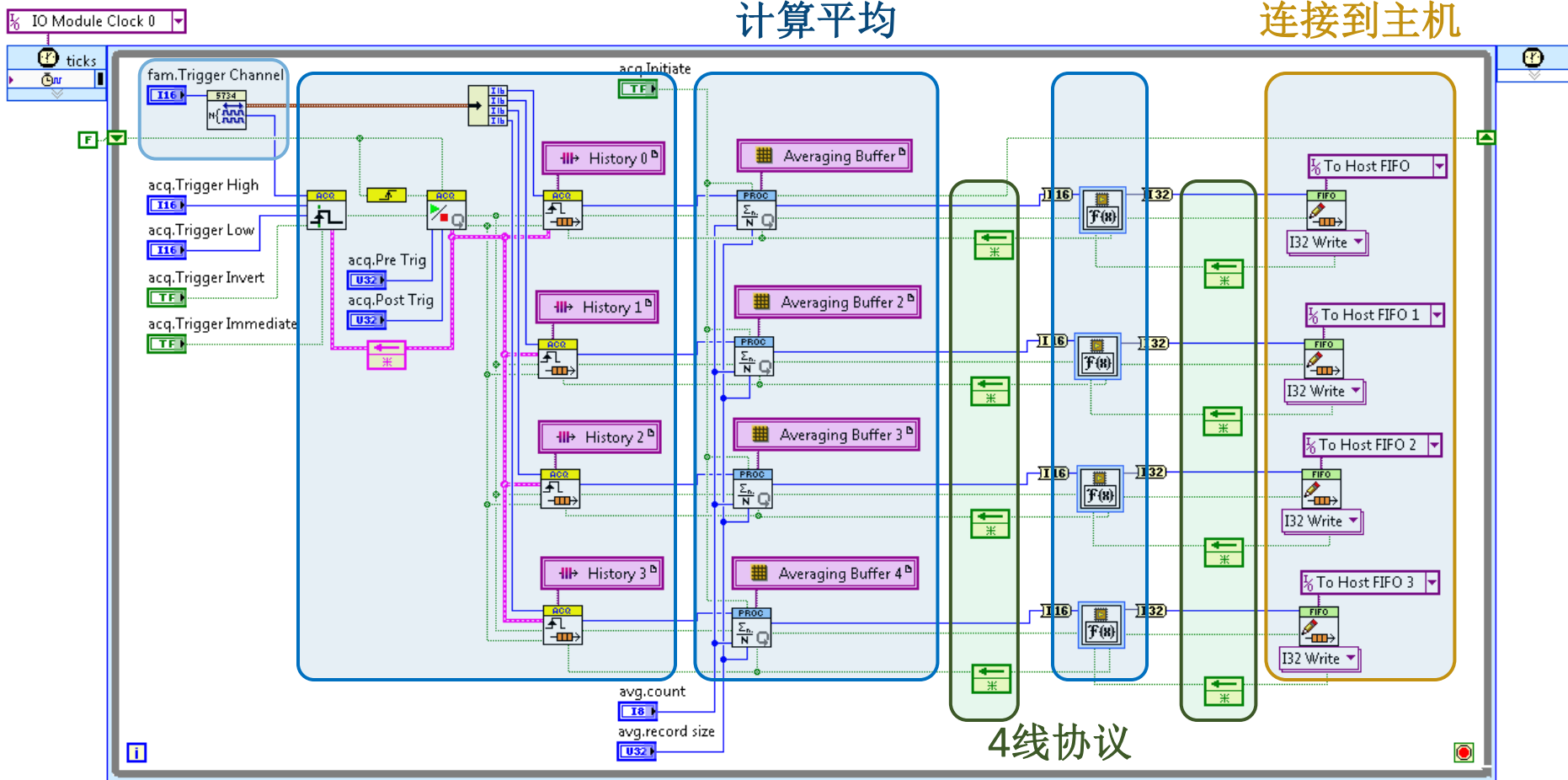
NI 5734

数据采集引擎

多次记录
计算平均

FFT

DMA FIFO
连接到主机



*备注: 此结构相关组件来自NI FlexRIO仪器开发库
ni.com/china

案例分析:

Timing Violation Analysis (fpgaTop.vi on Example 1.lvproj/FPGA Target)

Timing Information

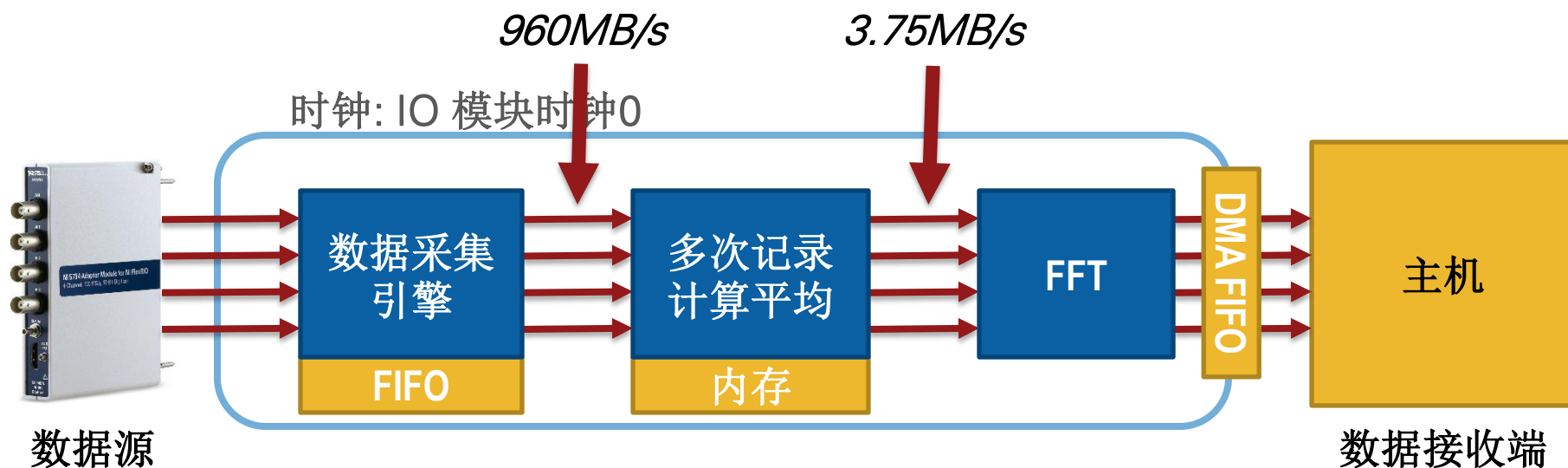
Paths	Total Delay	Logic Delay	Routing Delay	Max Fanout
Path 1 : Requirement 4.00ns missed by 0.01ns	3.73	0.48	3.25	1
Path 2 : Requirement 8.33ns missed by 4.78ns	12.73	1.43	11.30	1288
Non-diagram component	0.45	0.45	0.00	
Timed Loop	7.21	0.47	6.74	1288
Processing - Multi Record Average Loop.vi	1.04	0.19	0.85	192
FFT 2	1.29	0.09	1.19	69
FFT	4.88	0.19	4.70	1288
Non-diagram component	1.72	0.41	1.30	31
Path 3 : Requirement 8.33ns missed by 4.93ns	12.88	1.36	11.52	1288
Non-diagram component	0.47	0.47	0.00	
Timed Loop	7.34	0.38	6.97	1288
Processing - Multi Record Average Loop.vi	1.17	0.09	1.08	23
FFT 2	1.29	0.09	1.19	69
FFT	4.88	0.19	4.70	1288
Non-diagram component	1.72	0.41	1.30	31
Path 4 : Requirement 8.33ns missed by 4.86ns	12.80	1.43	11.37	1288
Non-diagram component	0.45	0.45	0.00	
Timed Loop	7.21	0.47	6.74	1288
Processing - Multi Record Average Loop.vi	1.04	0.19	0.85	192
FFT 2	1.29	0.09	1.19	69
FFT	4.88	0.19	4.70	1288
Non-diagram component	1.79	0.41	1.38	31
Path 5 : Requirement 8.33ns missed by 5.01ns	12.95	1.36	11.60	1288
Non-diagram component	0.47	0.47	0.00	
Timed Loop	7.34	0.38	6.97	1288
Processing - Multi Record Average Loop.vi	1.17	0.09	1.08	23
FFT 2	1.29	0.09	1.19	69
FFT	4.88	0.19	4.70	1288
Non-diagram component	1.79	0.41	1.38	31
Path 6 : Requirement 8.00ns missed by 0.63ns	8.42	2.22	6.21	64
Path 7 : Requirement 8.00ns missed by 0.70ns	8.49	2.22	6.27	64

Close Show Element Show Path Help

案例分析: 重新评估吞吐量

多次记录平均计算单元输出的数据是原数据的256分之一。

可以充分利用数据抽取来减少下级数据流节点串行化时所需要的元素数目。

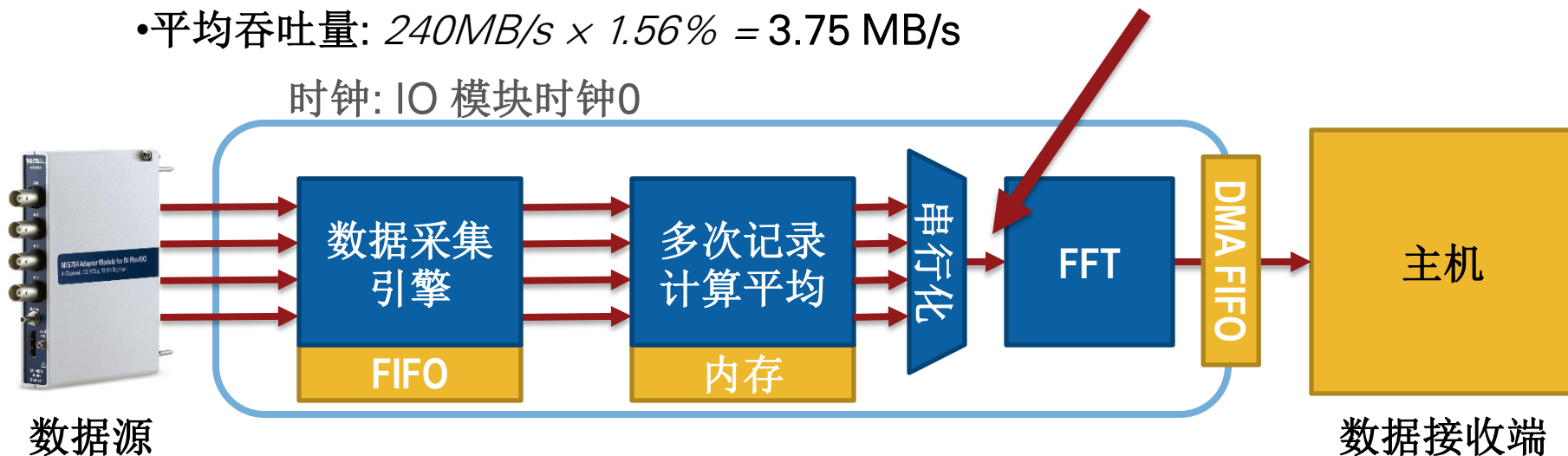


案例分析: 重新评估吞吐量

多次记录平均计算单元输出的数据是原数据的256分之一。

- 区块大小: 1024个样本
- 占空比: $0.39\% \times 4 = 1.56\%$
- 瞬时吞吐量: $960\text{MB/s} \div 4 = 240\text{MB/s}$
- 平均吞吐量: $240\text{MB/s} \times 1.56\% = 3.75\text{ MB/s}$

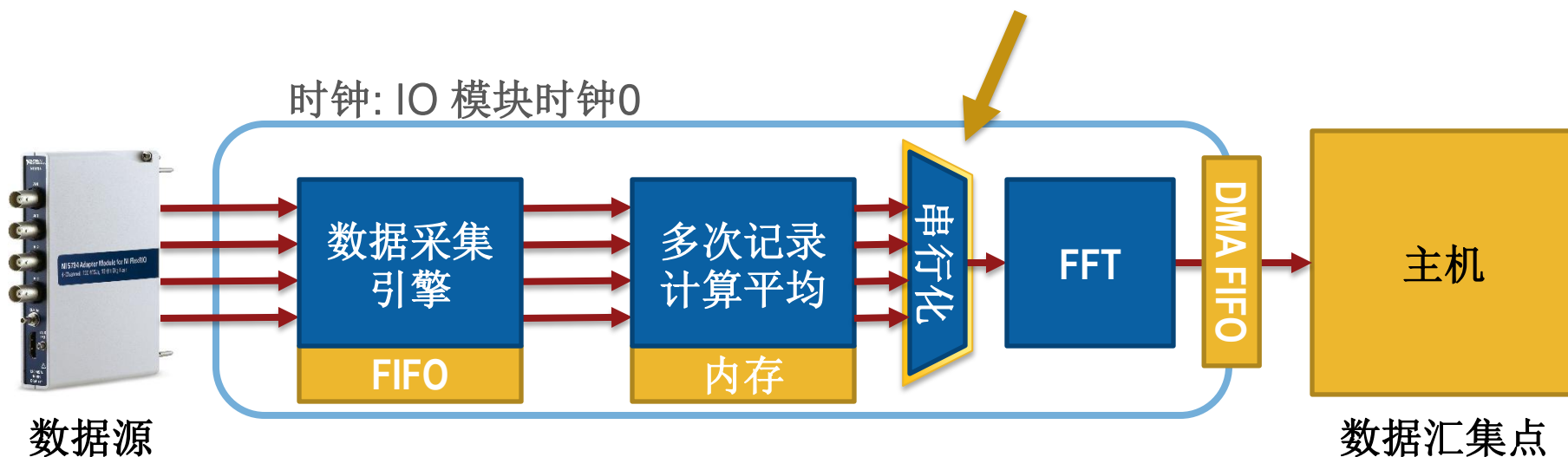
时钟: IO 模块时钟0



案例分析: 重新评估存储

新的串行化单元需要内存来缓存并行的输入数据。

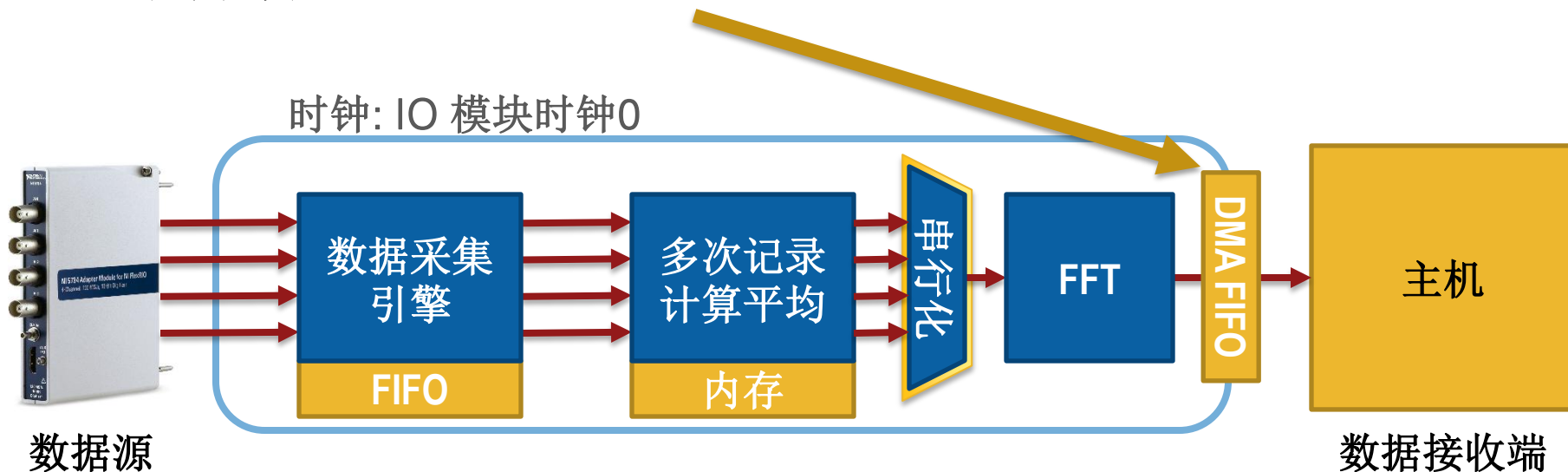
- 数据带宽: $32 \text{ 位} \times 4 \text{ 通道} = 16 \text{ 字节}$
- 元素个数: 记录的长度 (1024个样本)
- 应用: 4个并行块RAM FIFO



案例分析: 重新评估存储

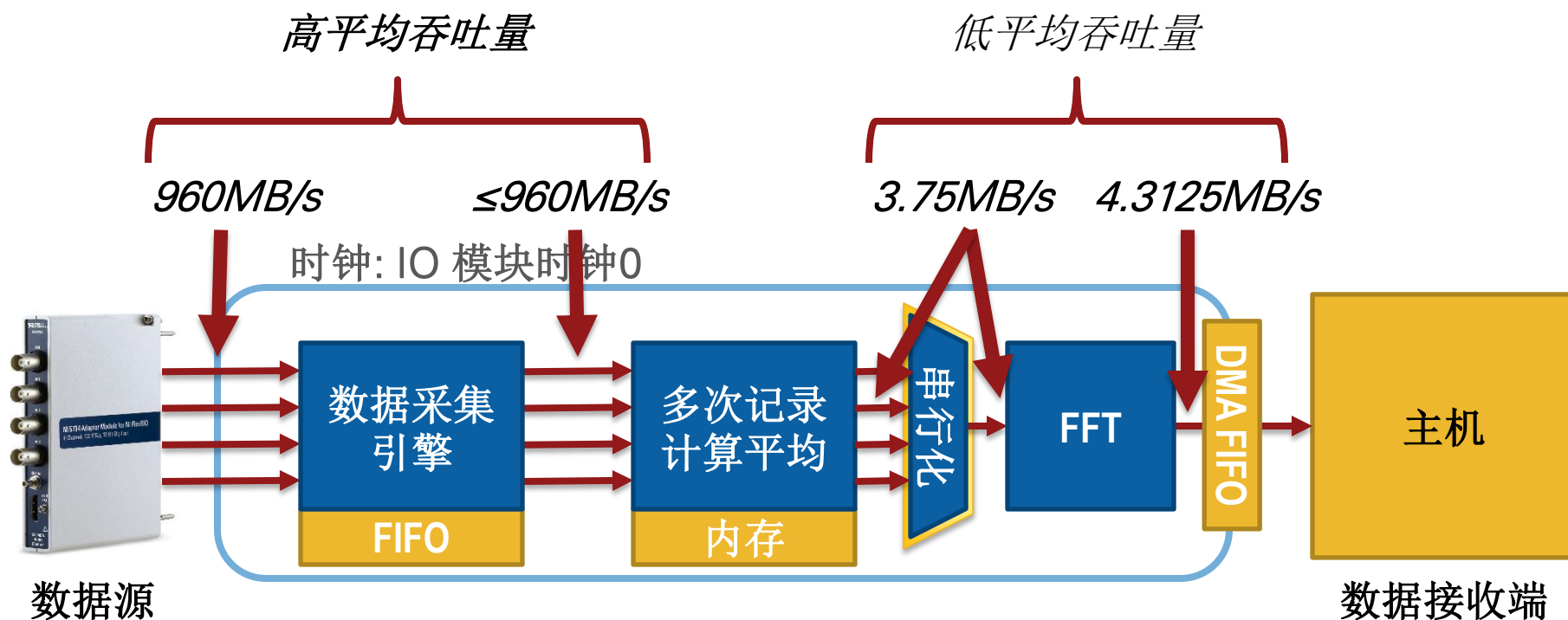
修改DMA FIFO以适应新的数据带宽和元素个数。

- 数据带宽: 23 位→ 4 字节(为FIFO取整)
- 元素个数: 4 个记录的长度(4096个样本)
- 应用: 块RAM连接主机的FIFO



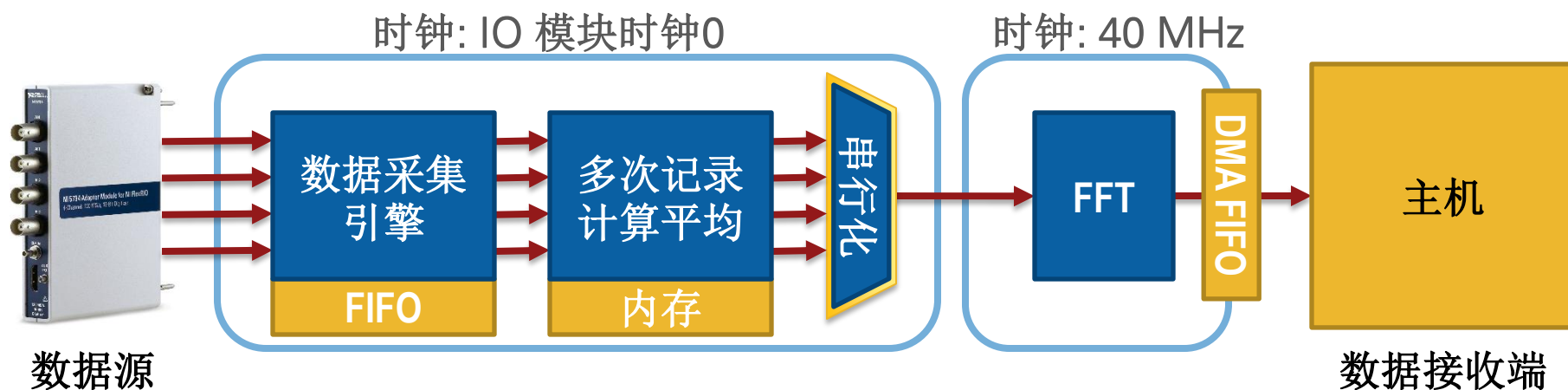
案例分析: 重新评估时钟

FFT单元的吞吐量更低，因此可以在更低的时钟频率下运行。



案例分析: 重新评估时钟

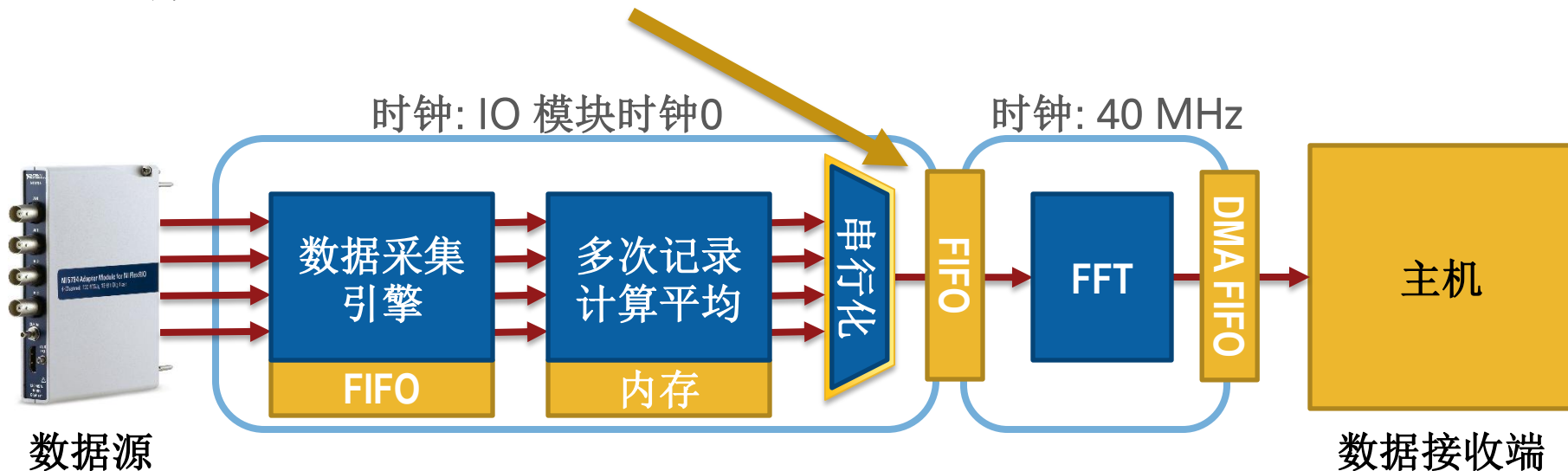
FFT单元的吞吐量更低，因此可以在更低的时钟频率下运行。



案例分析: 重新评估存储

需要使用一个块RAM FIFO，用于在时域内传递数据。

- 数据带宽: $32 \text{ 位} \times 1 \text{ 通道} = 4 \text{ 字节}$
- 元素个数: 4 次记录(4096个样本, 16KB)
- 方法: 单个块RAM定向连接FIFO



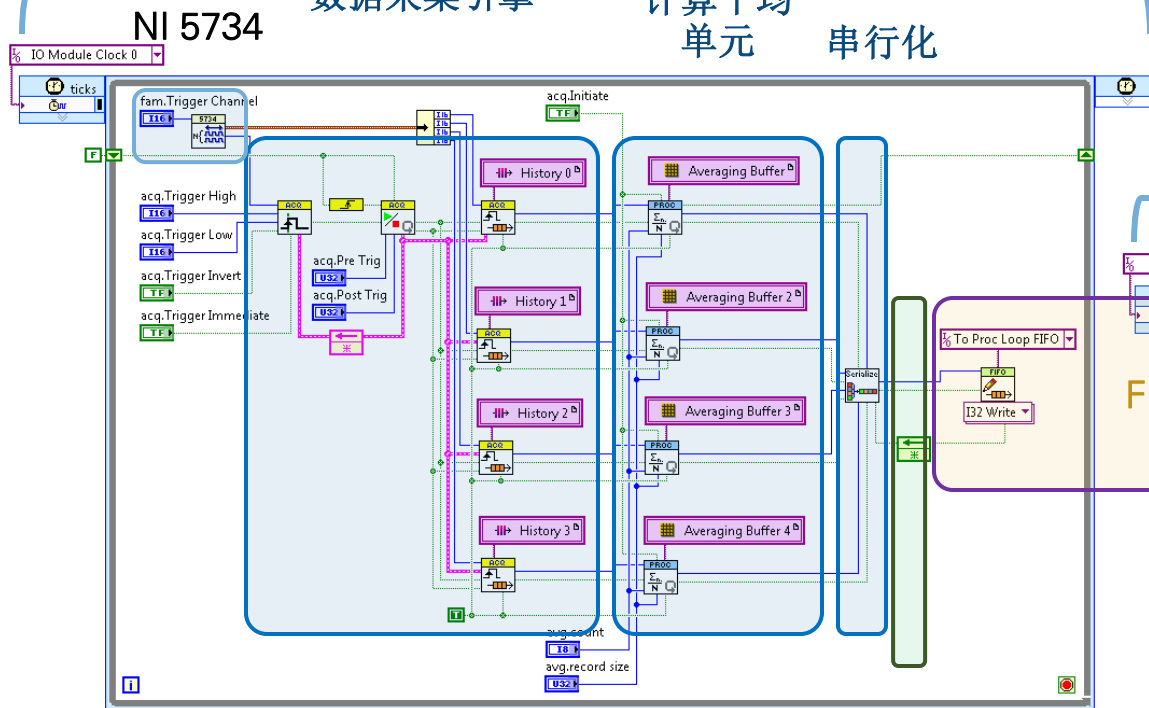
案例分析: 串行化 & 多时钟

IO 模块时钟0 (120 MHz)

数据采集引擎

多次记录
计算平均
单元

串行化

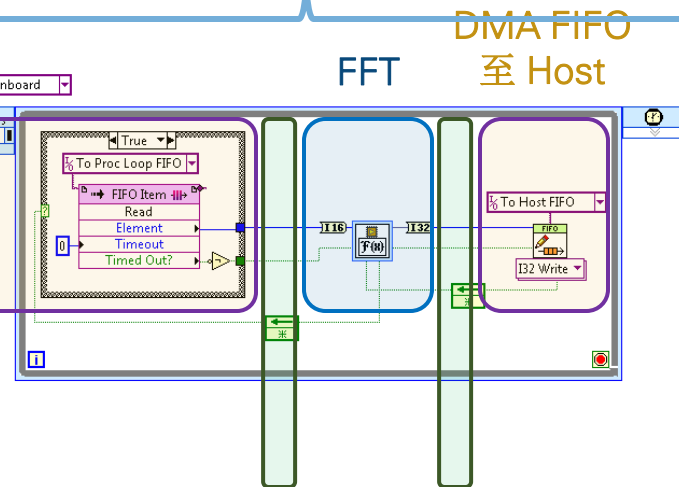


40 MHz 时钟

DMA FIFO
至 Host

FFT

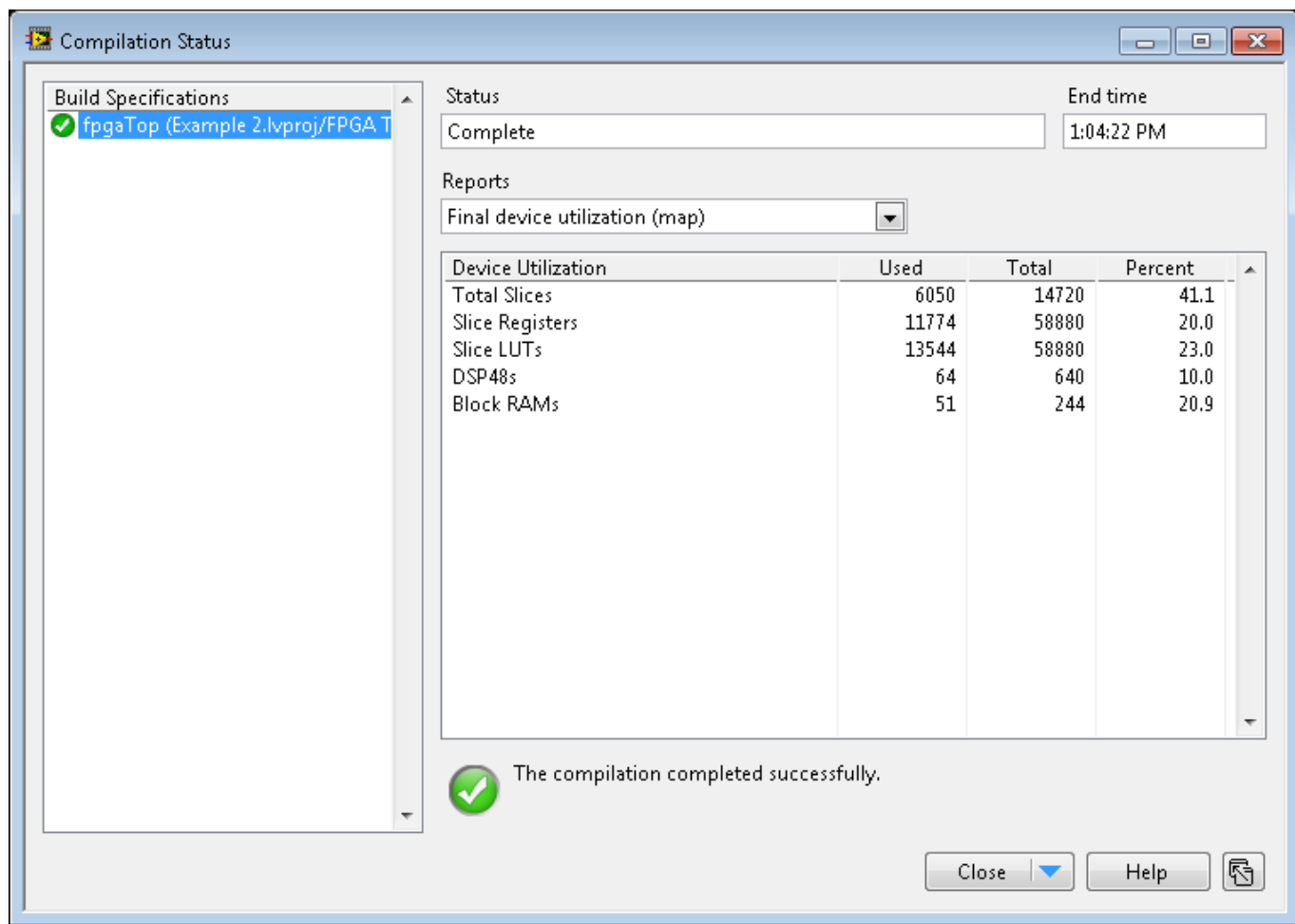
FIFO



4线通信协议

*Note:此结构相关组件来自NI FlexRIO仪器开发库
ni.com/china

案例分析: 再次尝试编译...



结论

- 数据流盘应用在数据输入时立即处理，一般是逐点式的处理
- 理解硬件的不同存储设备和带宽的兼容性
 - 每个案例都可以采用不同的数据源组合方式
- 在设计系统时 (编写代码之前!) 需要考虑:
 - 吞吐量
 - 存储
 - 时序
 - 通信协议

其它资源

- 高性能RIO开发手册: ni.com/hprioiguide
- 高吞吐量FPGA培训: ni.com/training
- NI FlexRIO设备开发库: ni.com/labs
- LabVIEW FPGA产品文档 ni.com/flexrio

更多自动化测试与测量专题

- 11:15–12:00 会议厅3E
《为自动化测试选择最佳软件工具》
- 11:15–12:00 黄河厅
《纵览新一代数据记录系统》
- 13:30–14:15 黄河厅
《使用最新技术和总线进行高速、高吞吐量测试》
- 13:30–14:15 会议厅3J
《从概念到原型-基于LabVIEW的软件无线电平台介绍》
- 14:30–15:15 国际厅
《选择PXI平台的几点考虑》
- 15:30–16:15 国际厅
《多通道数据采集与分析系统的构建要诀》
- 15:30–16:15 会议厅3J
《基于LabVIEW FPGA构建高速流盘应用》

谢谢