

芯动力——硬件加速设计方法

第二章：高质量VerilogHDL描述方法(3)

邸志雄@西南交通大学

zxdi@home.swjtu.edu.cn

RTL设计指导原则

RTL 级设计的评判标准

时序性能

所占面积

可测试性

可重用性

功耗

时钟域的分配

复位信号设计

是否与所用 EDA 工具匹配

FPGA

CPLD

- 是否能发挥这些 PLD 的结构特点等

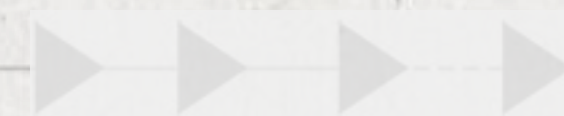
- 根据这些目标的组合和优先级设置，可以派生出很多不同的设计原则。

指导原则：

面积与速度互换

乒乓操作

流水线设计



面积与速度互换

面积

FPGA

CPLD

ASIC

- 一个设计所消耗的目标器件的**硬件资源数量**或者**ASIC芯片的面积**。

- 可以用所消耗的**触发器**和**查找表**数量来衡量;

CFF

CLUT

- 可以用设计的系统门衡量。

速度

- 设计在芯片上稳定运行时所能够达到的**最高频率**，这个频率由设计的时序状况决定

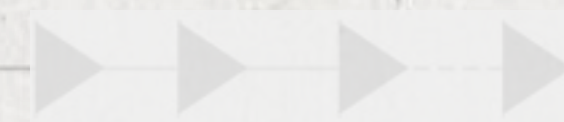
时钟周期

PAD to PAD Time

Clock Setup Time

Clock Hold Time

Clock-to-Output Delay



面积

速度



RTL 设计的始终

衡量设计质量的终极标准

面积

- 设计面积最小

速度

- 运行频率最高

对立统一的矛盾体

科学的设计目标

面积和速度的平衡思想

面积

- 在满足设计时序要求(包含对设计最高频率的要求)的前提下, 占用最小的芯片面积;

速度

- 在所规定的面积下, 使设计的时序余量更大, 频率更高。

要求

- 应该认识到它们是和产品的质量、成本直接相关的。

如果设计的时序余量比较大，运行的频率比较高

- 设计的健壮性更强
- 整个系统的质量更有保证

设计所消耗的面积更小

- 在单位芯片上实现的功能模块更多
- 需要的芯片数量更少
- 整个系统的成本也随之大幅度削减

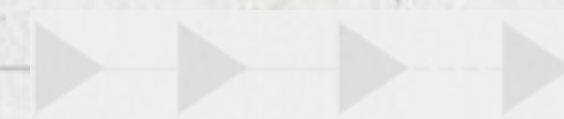
面积

速度



满足时序

工作频率



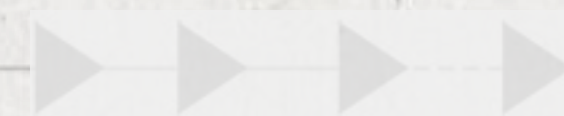
面积与速度互换

理论

- 一个设计如果时序余量较大，所能跑的频率远远高于设计要求
 - 能通过功能模块复用减少整个设计所消耗的芯片面积
 - 用速度的优势对换面积的节约
- 一个设计的时序要求很高，普通方法达不到设计频率
 - 可以通过将数据流串并转换
 - 并行复制多个操作模块
 - 对整个设计采取"乒乓操作"和"串并转换"的思想进行处理
 - 在芯片输出模块处再对数据进行"并串转换"

宏观

- 整个芯片满足了处理速度的要求，这相当于用面积复制换取速度的提高



面积和速度互换

操作技巧



模块复用

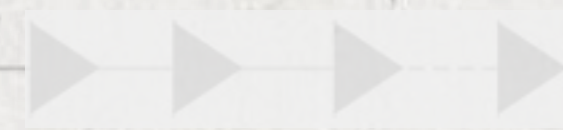
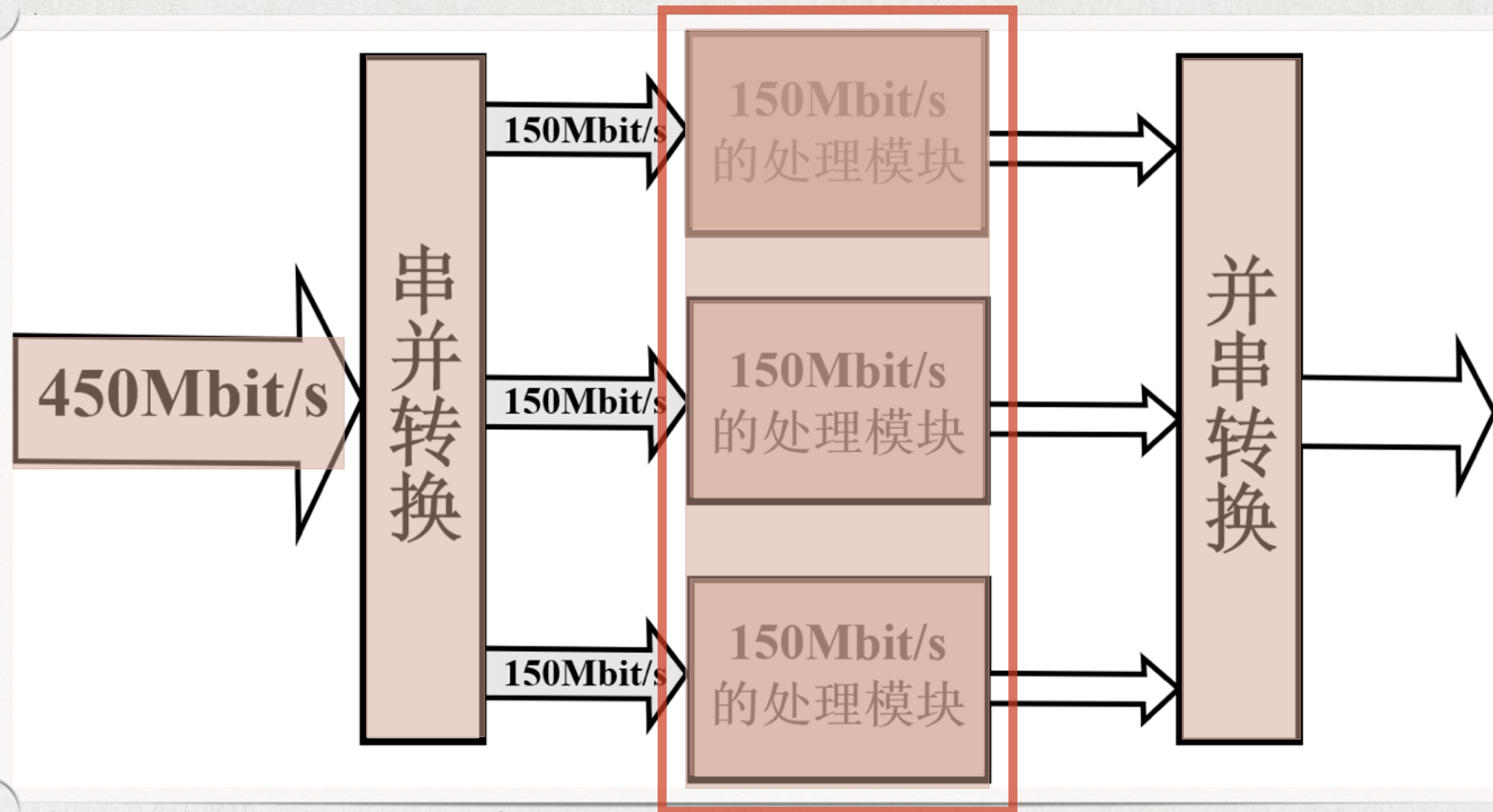
乒乓操作

串并转换

例：如何进行“面积换速度”的操作？

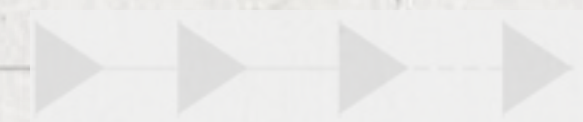
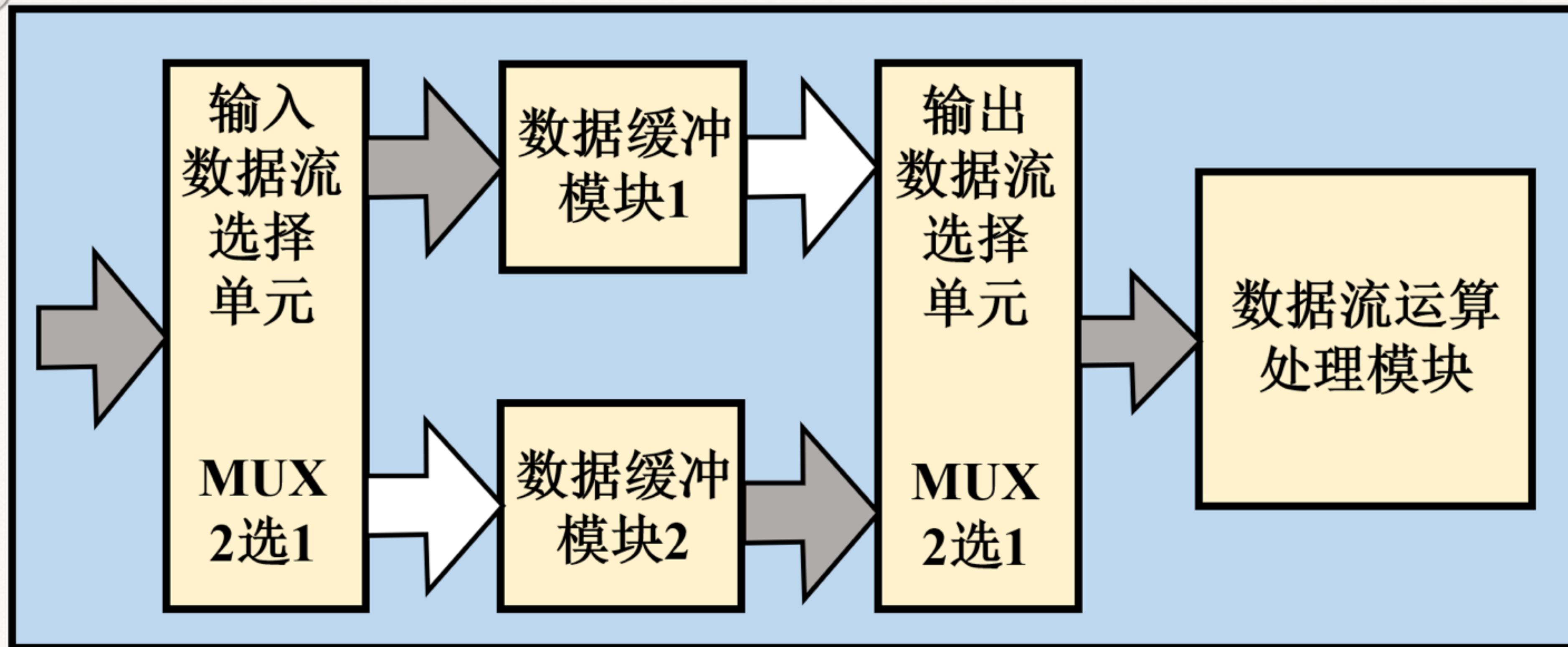
- 本例是一个路由器设计实例。假设输入数据流的速率是450Mbit/s的，而在FPGA上设计的数据处理模块的处理速度最大为150Mbit/s，由于处理模块的数据吞吐量满足不了要求，因此直接在FPGA上实现是一个不可能完成的任务。在这种情况下，就应该利用“面积换速度”的思想，至少复制3个处理模块。首先将输入数据进行串并转换，然后利用这3个模块并行处理分配的数据，最后将处理结果“并串转换”以满足数据速率的要求。在处理模块的两端，其数据速率是450Mbit/s，而在FPGA的内部，每个子模块处理的数据速率是150Mbit/s。其实整个数据吞吐量的保障是依赖于3个子模块的并行处理来完成的，也就是说通过占用更多的芯片面积来实现高速处理。设计示意图如所示。

例：如何进行 **面积换速度** 的操作？



乒乓操作

- “乒乓操作”是一个常常应用于数据流控制的处理技巧



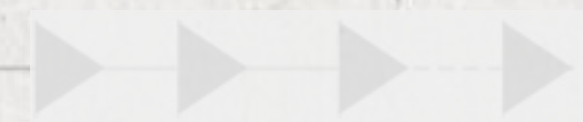
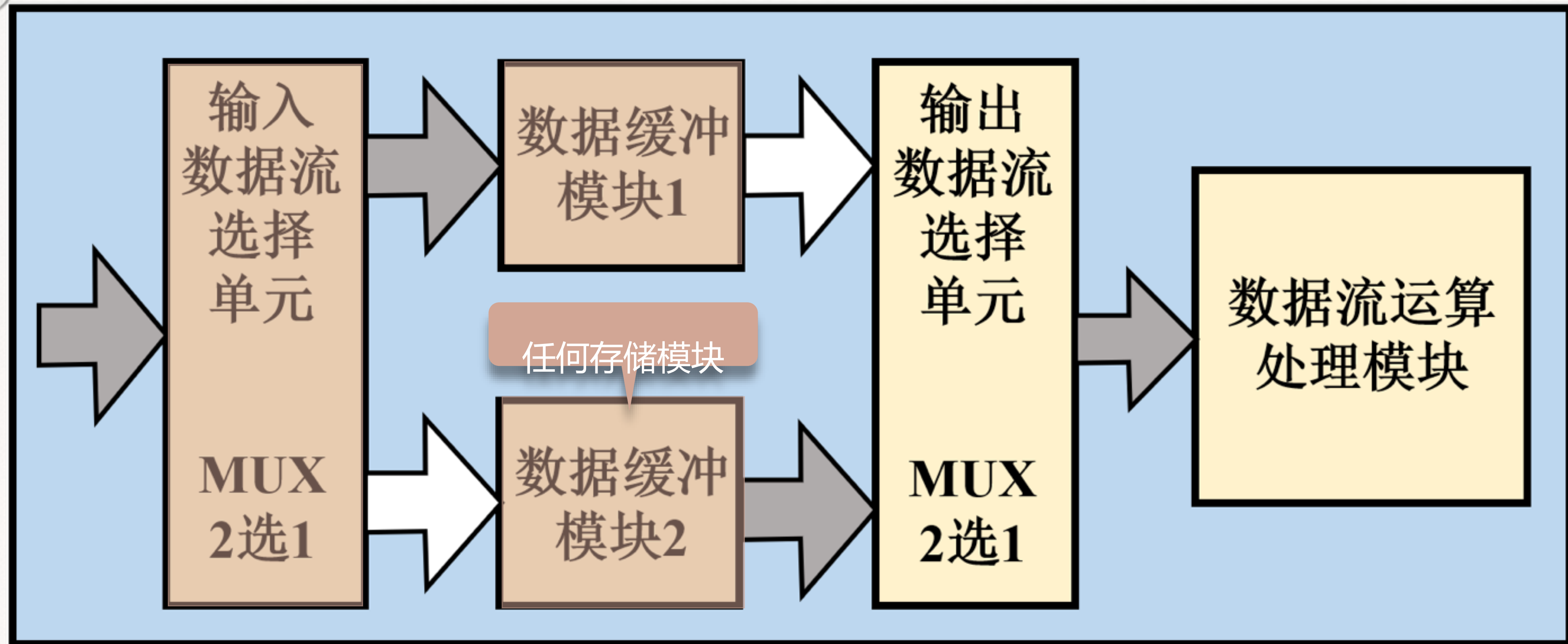
乒乓操作的处理流程

常用的存储单元

DPRAM
双口RAM

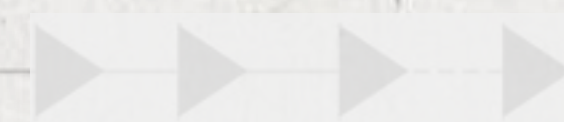
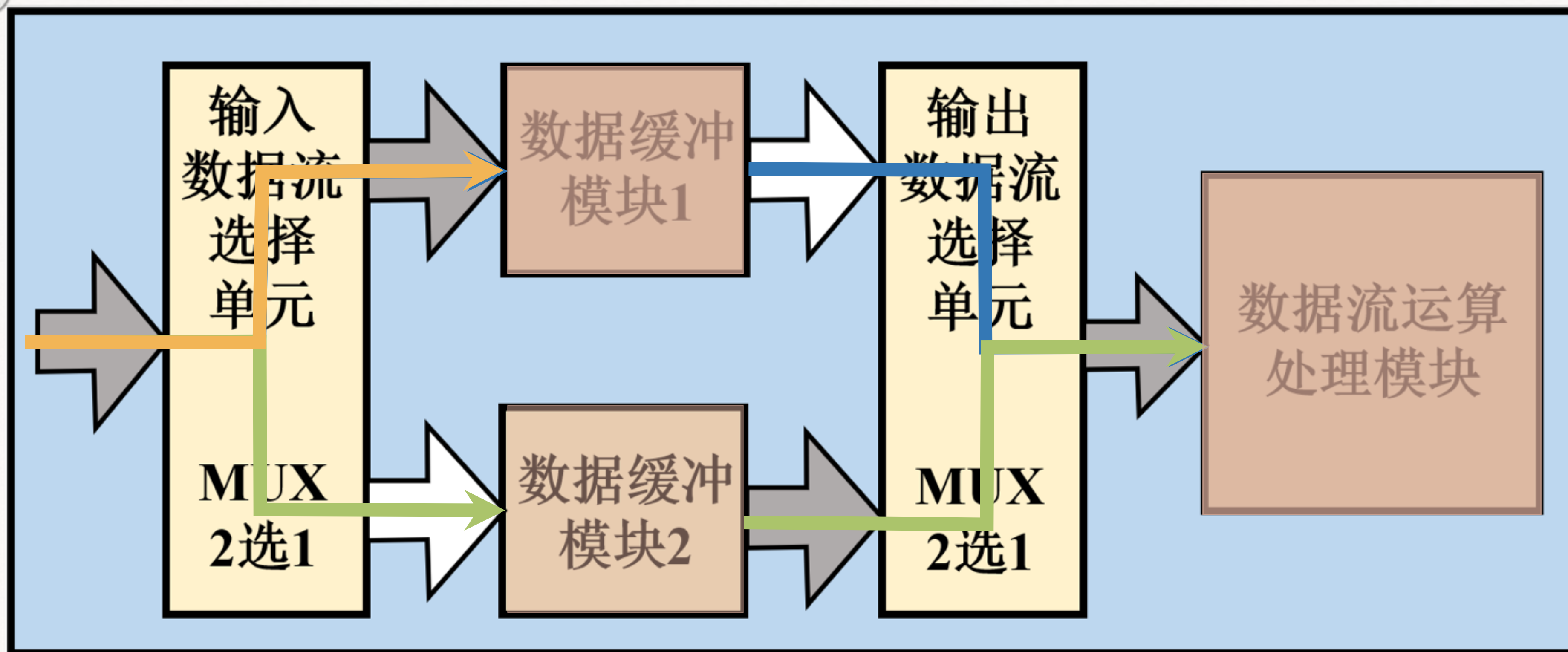
SPRAM
单口RAM

FIFO



乒乓操作的处理流程

第3个缓冲周期



乒乓操作

最大特点

输入数据选择单元

输出数据选择单元

- 按节拍、相互配合的切换

- 经过缓冲的数据流

数据流运算处理模块

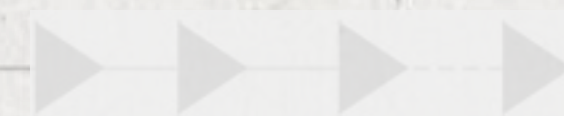
- 进行运算与处理

乒乓操作模块

- 站在这个模块的两端看数据，输入数据流和输出数据流都是连续不断的，没有任何停顿，因此非常适合对数据流进行流水线式处理。

流水线式算法

- 完成数据的无缝缓冲与处理



第二个优点

节约缓冲区空间

WCDMA基带应用

1个帧

15个时隙

缓冲需要将1整帧的数据速率一个时隙内处理

缓冲区长度

1整帧

办法是将这

3.84Mbps

来,

10ms

1个时

38400位

乒乓操作

- 两个能缓冲1个时隙数据

- 单口RAM即可

RAM

- 写数据

RAM

- 读数据

处理单元

5120

$$38400 = 3.84 \times 10^6 \times 10 \times 10^{-3}$$

$$\frac{38400}{15} = 2560$$

用低速模块处理高速数据流的效果

WCDMA设计中

输入数据流



解扩

解扰

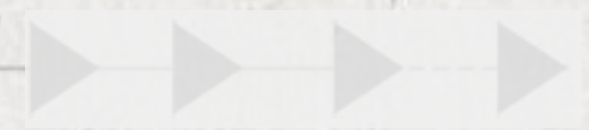
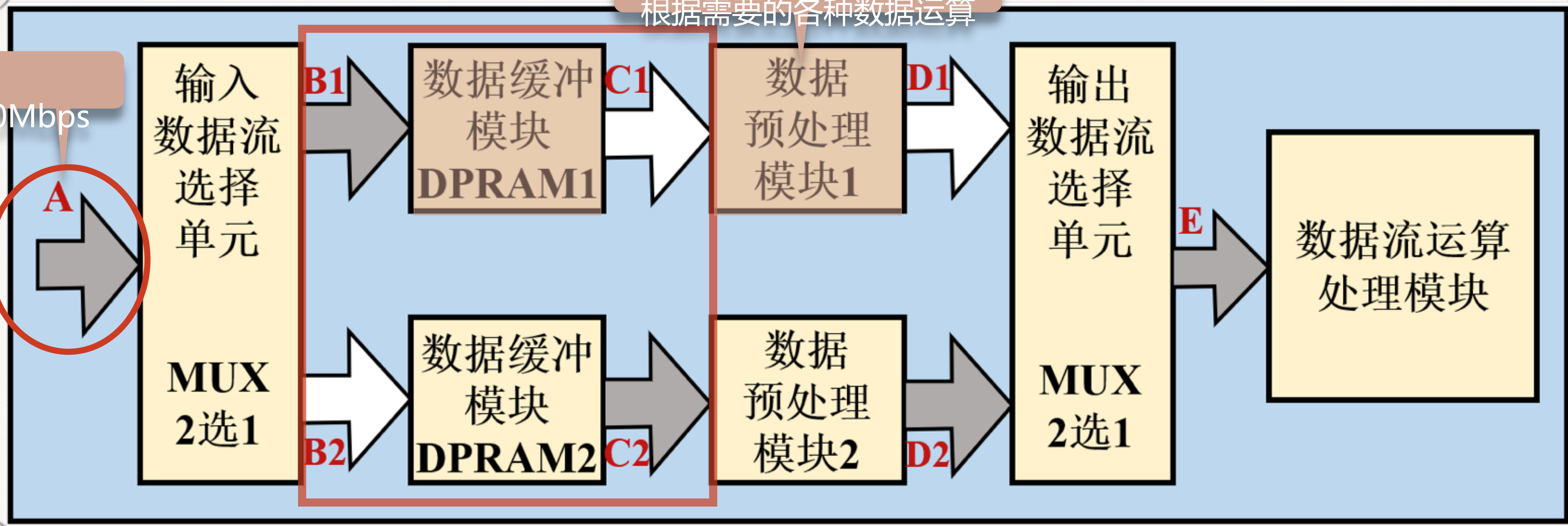
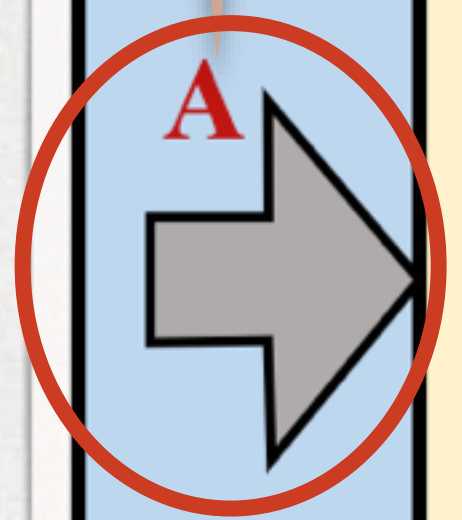
去旋转

乒乓操作的缓冲周期

10ms

根据需要的各种数据运算

100Mbps



第1个缓冲周期10ms内

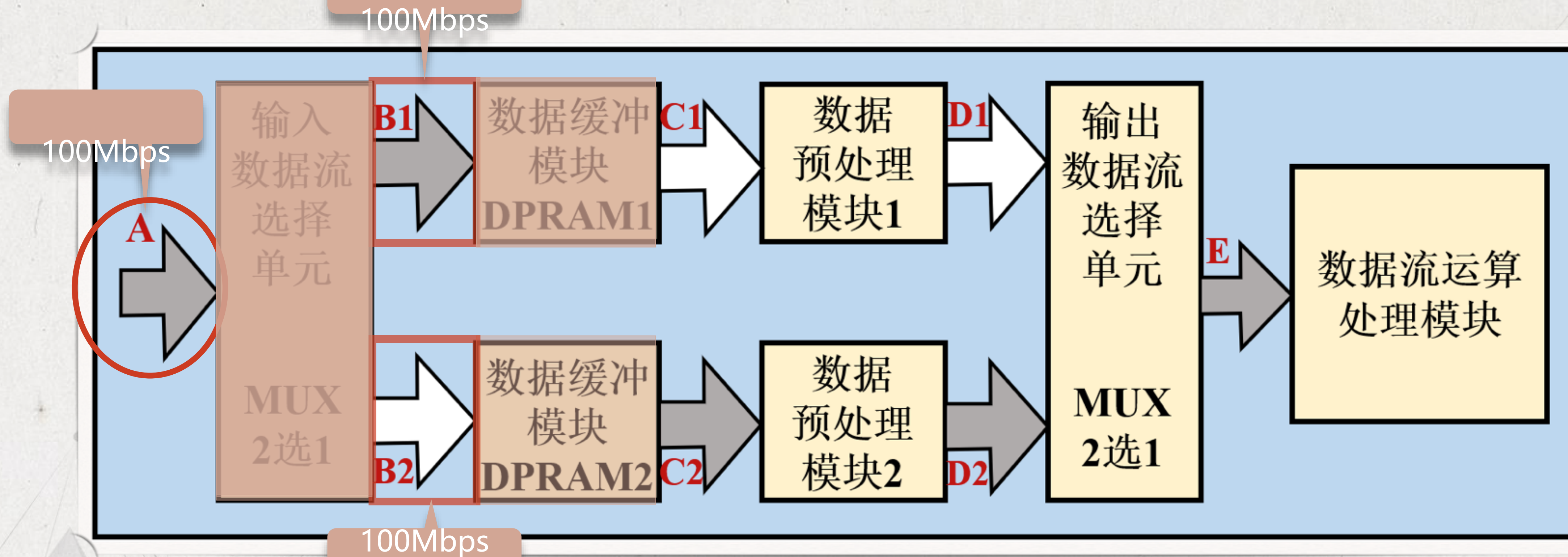
- DPRAM1要在10ms内写入1Mb数据

第2个10ms

- DPRAM2在第2个10ms被写入1Mb数据

第3个10ms

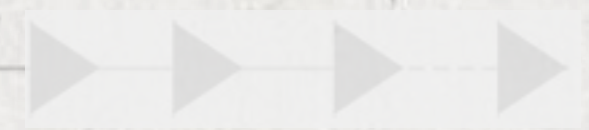
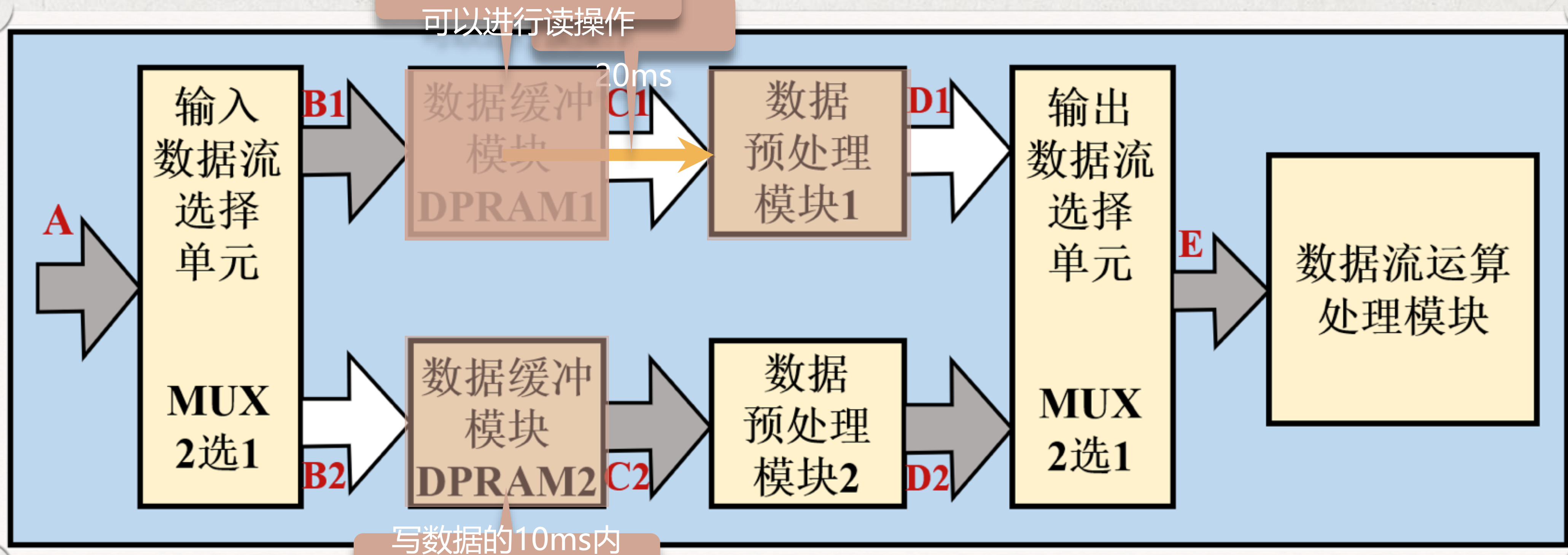
- DPRAM1被写入1Mb数据



DPRAM1就可以一边向500K以后的地址写数据，一边从地址0读数
到达10ms时，DPRAM1刚好写完了1Mb数据，并且读了500K数据，这个缓冲时间内
DPRAM1读了5ms；

第1个缓冲周期

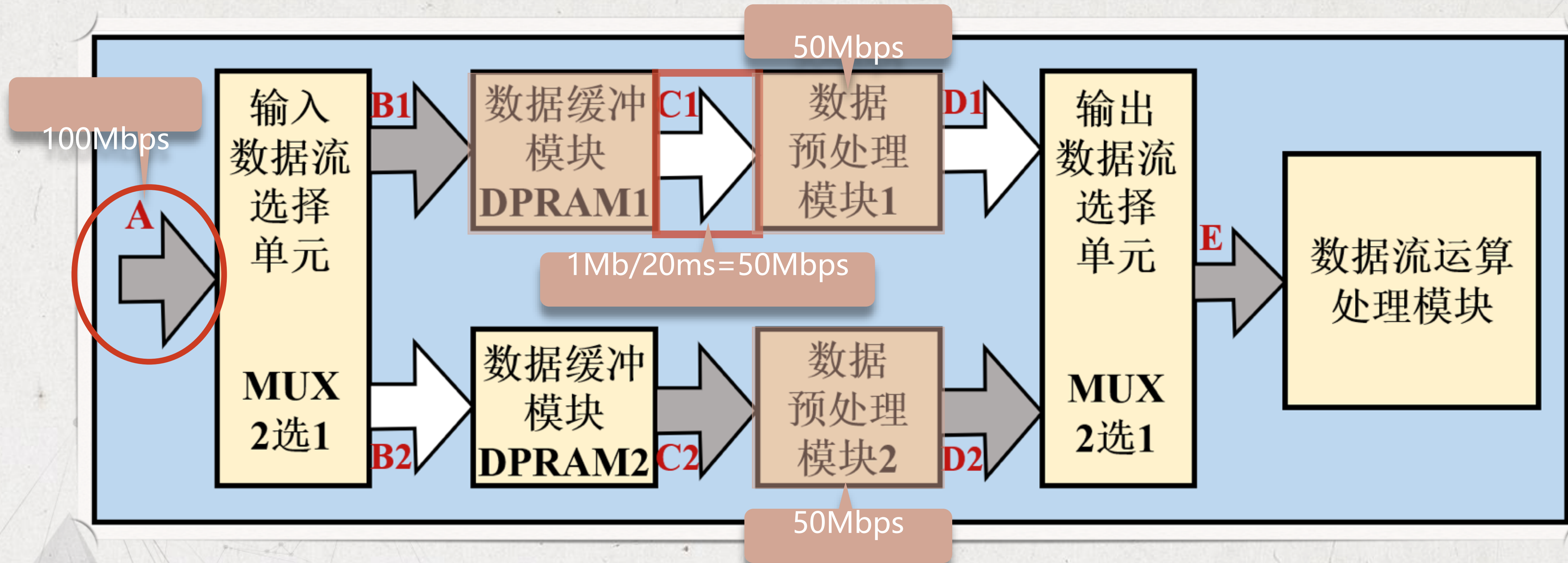
• 第5ms起(绝对时间为5ms时刻)



第3个缓冲周期

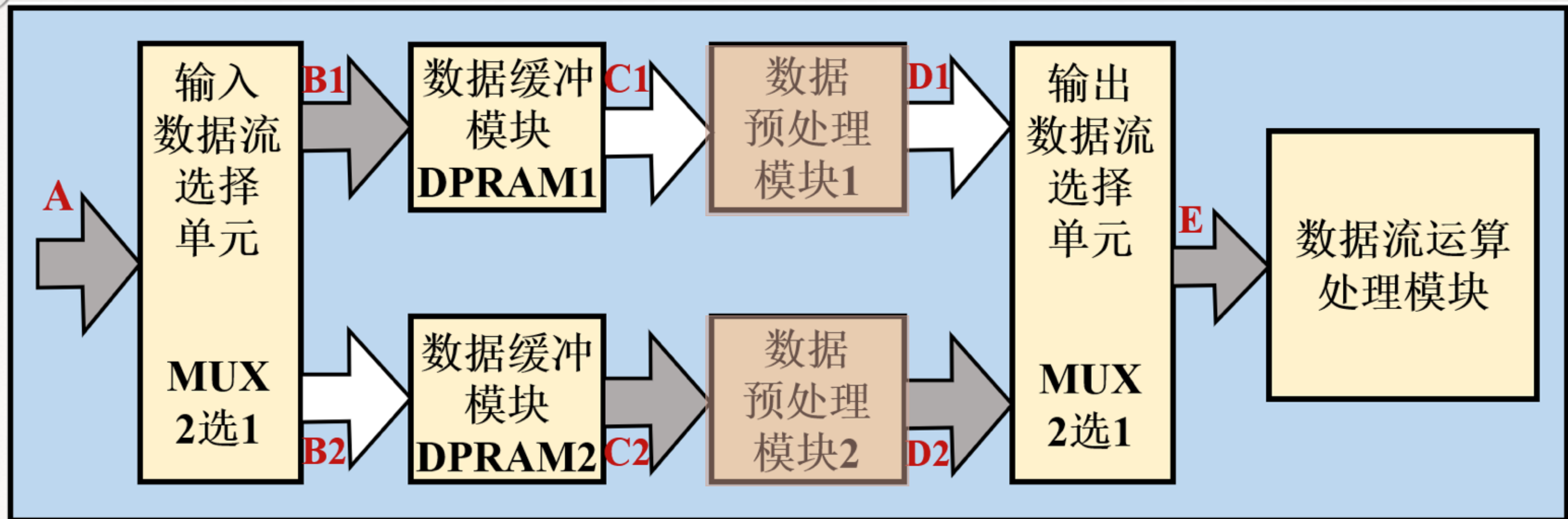
- 第5ms起(绝对时间为35ms时刻)

同理可以一边向500K以后的地址写数据一边从地址0读数，又读取了5个ms，所以截止DPRAM1第一个周期存入的数据被完全覆盖以前，DPRAM1最多可以读取20ms时间，而所需读取的数据为1Mb



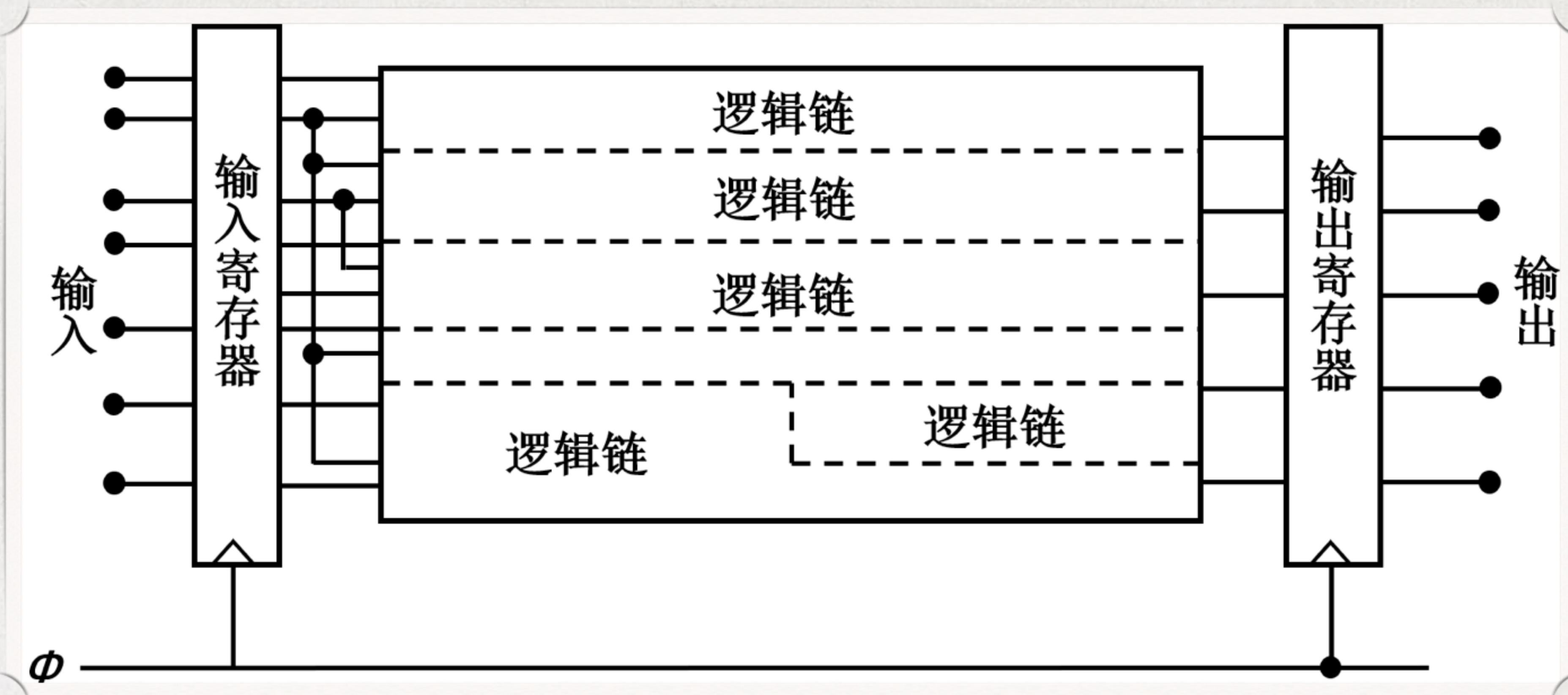
通过乒乓操作实现低速模块处理高速数据的实质

- 通过DPRAM这种缓存单元实现了数据流的串并转换，并行用“数据预处理模块1”和“数据预处理模块2”处理分流的数据，是面积与速度互换原则的体现！

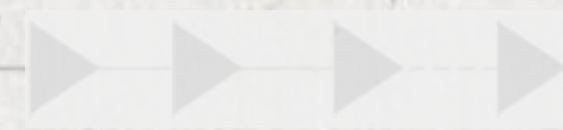


流水线

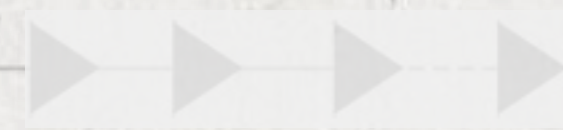
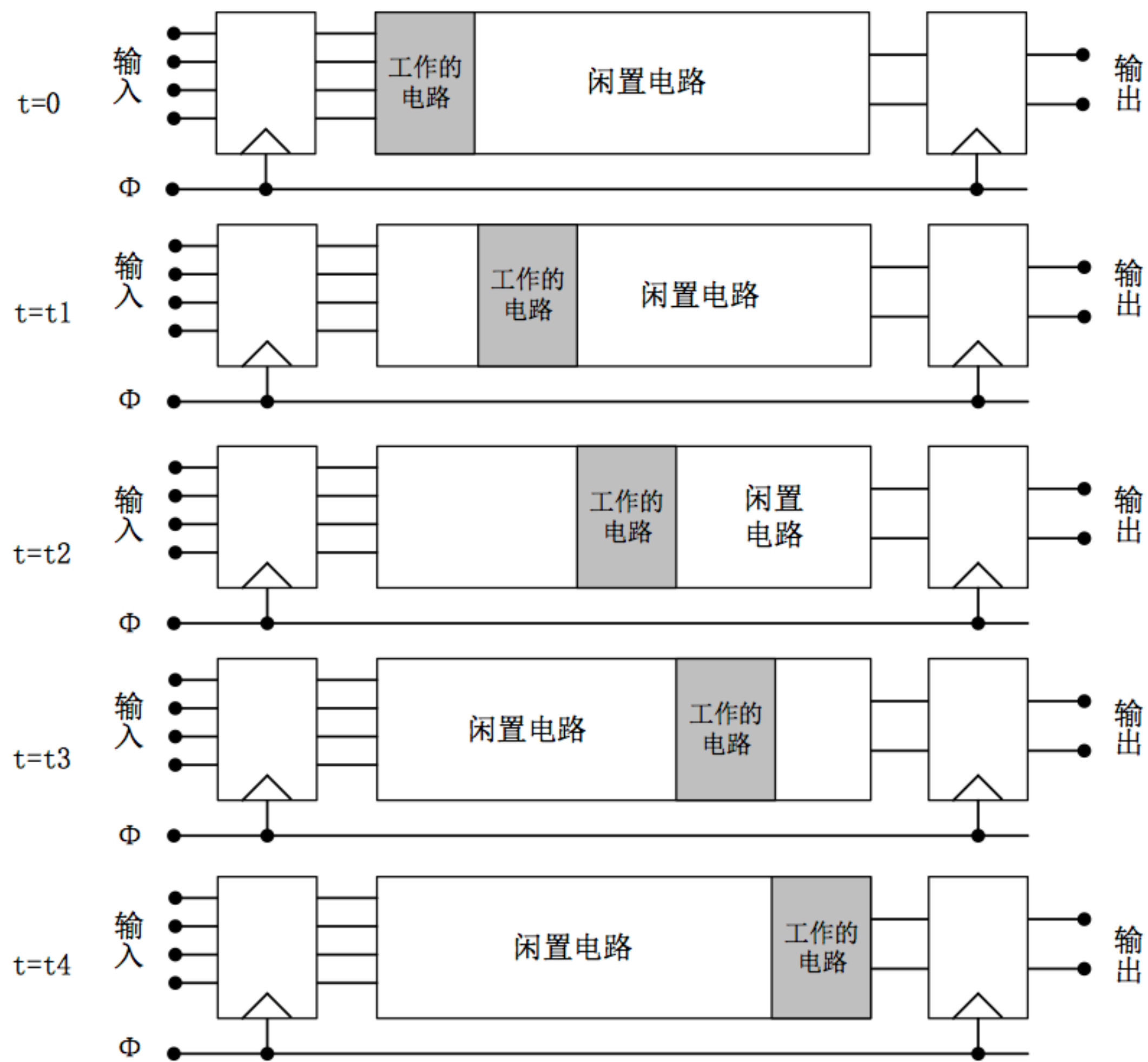
组合逻辑链的并行闲置



最长的组合逻辑链 → 最大的 t_d 值 → 系统时钟频率上限 → 系统速度上限

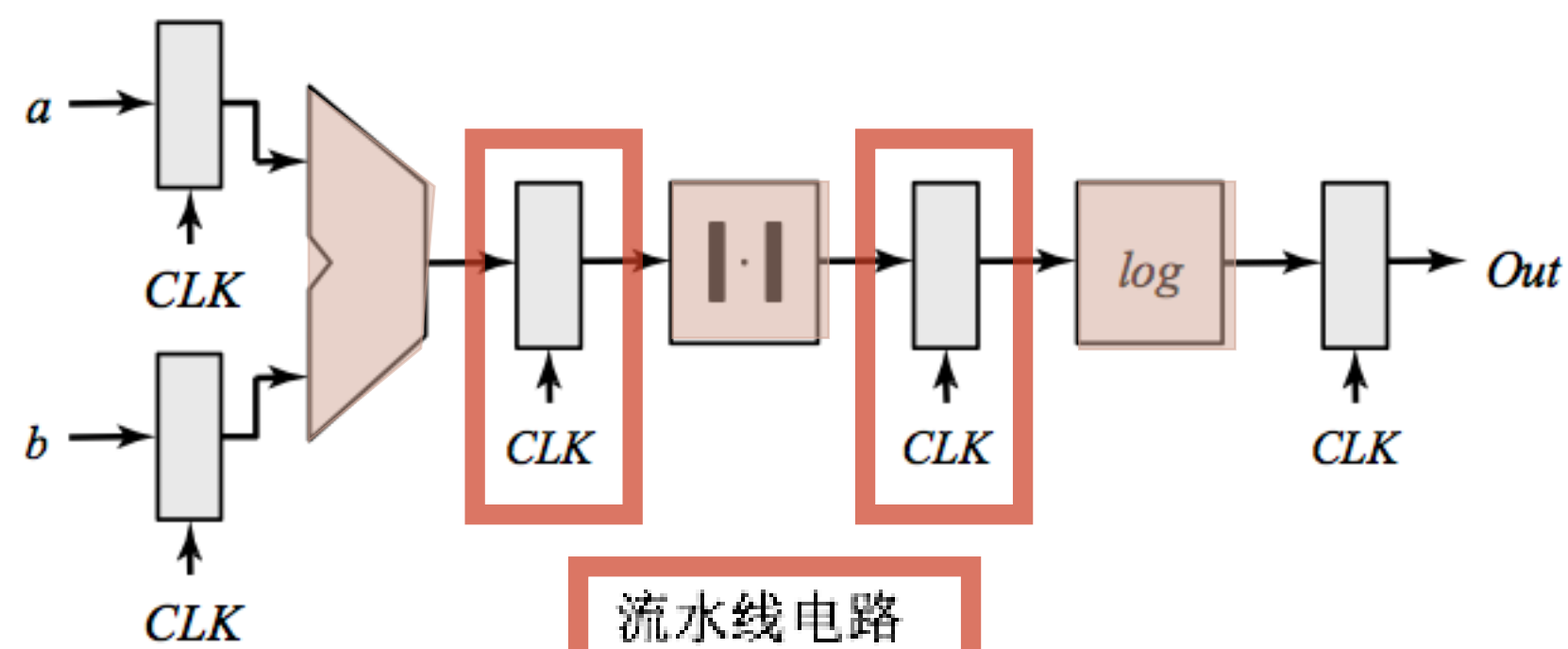
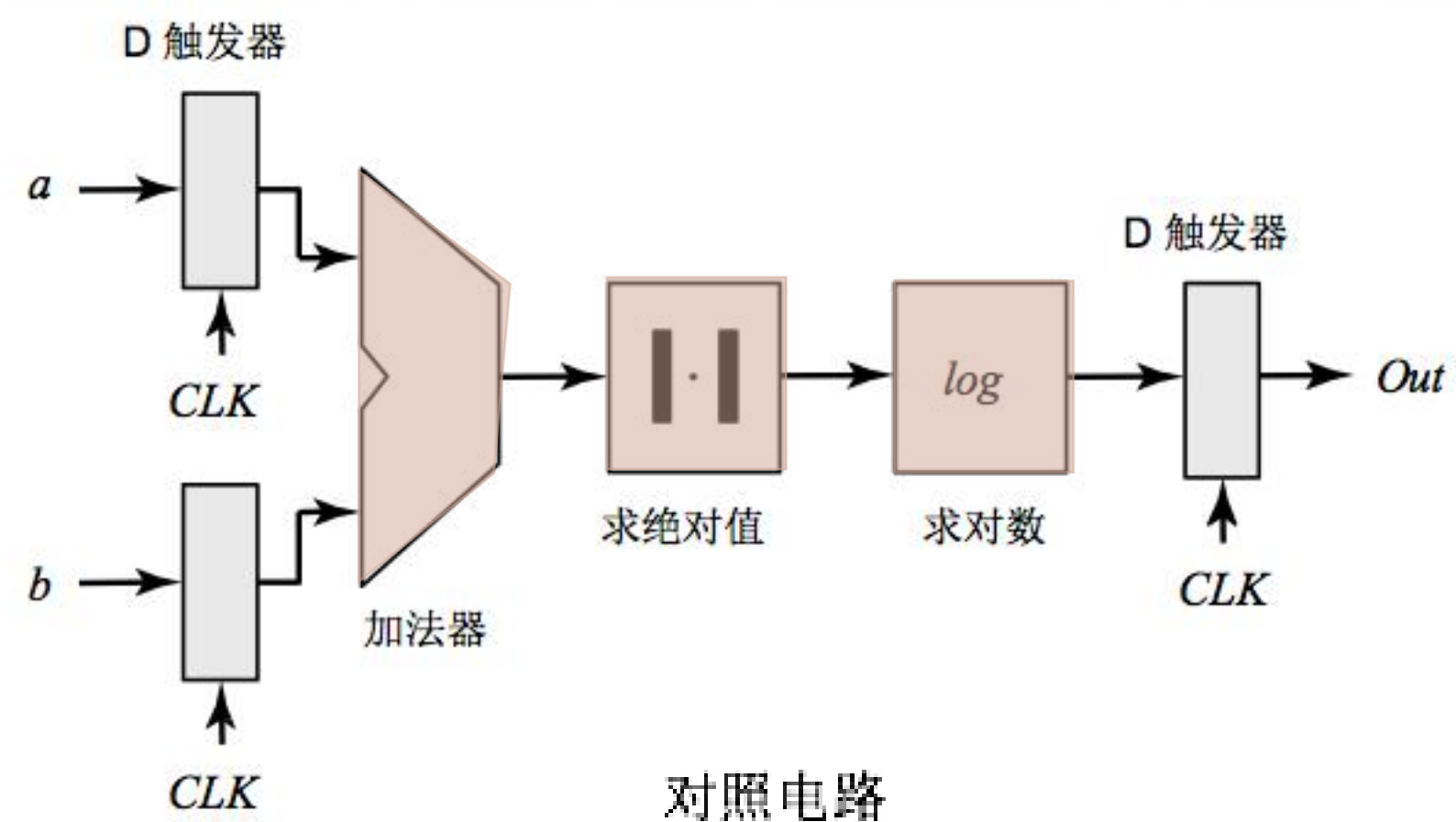


组合逻辑链的串行闲置



流水线

实例：计算 $\log(|a+b|)$



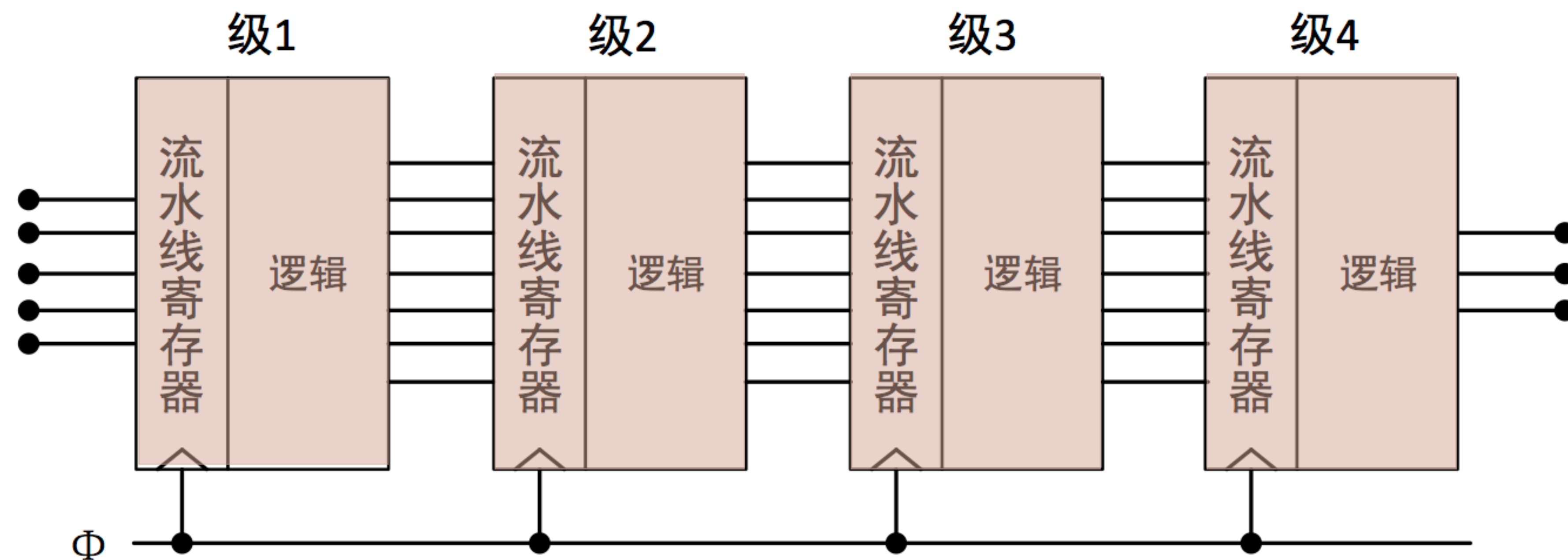
$$T_{\min} = t_{c-p} + t_{pd,add} + t_{pd,abs} + t_{pd,log} + t_{su} \\ = t_{pd,add} + t_{pd,abs} + t_{pd,log}$$

$$T_{\min}' = t_{c-p} + \max(t_{pd,add}, t_{pd,abs}, t_{pd,log}) + t_{su} \\ = \max(t_{pd,add}, t_{pd,abs}, t_{pd,log})$$

$$= \frac{1}{3}T \text{ (若 } t_{pd,add} = t_{pd,abs} = t_{pd,log} = \frac{1}{3}T \text{ min)}$$

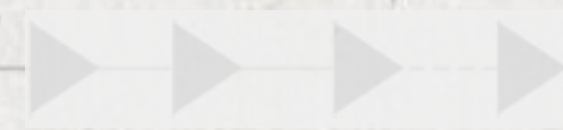
流水线

四级流水线的结构 (1)



按延迟来划分的

正沿触发



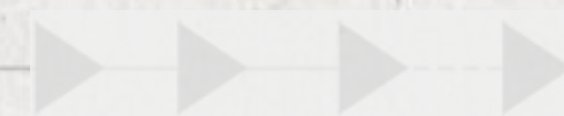
流水线

四级流水线比较

	非流水线	流水线
逻辑链延迟时间	T (整个逻辑链)	T_{pipe} (逻辑链的最长单元)
系统最高时钟频率	$f=1/T$	$f_{\text{pipe}}=1/T_{\text{pipe}}$
一组 N 个顺序输入数据的运算时间	NT	$4T_{\text{pipe}}+(N-1)T_{\text{pipe}}$ $=(N+3)T_{\text{pipe}}$

注: $T_{\text{pipe}} < T$ (因流水线单元一定比非流水线单元短)

$4T_{\text{pipe}} > T$ (因流水线需插入寄存器, 寄存器有延迟)



流水线

特点

- 通过插入寄存器，将长的串行逻辑链分成较小的部分
- 当系统运算是串行的时候，利用时钟控制，使运算依照顺序接续进行
- 在任何给定时刻，大部分电路都在工作

好处

- 每一部分延时较小→可使用更快的时钟
- 大部分电路同时进行运算→可提高数据通过量

流水线参数设计

系统时钟取决于最慢的流水线级的延时

- 流水线时钟周期 $T_{\text{pipe}} = \max\{T_1, T_2, \dots, T_m\}$
- 第 i 级的时钟周期 $T_i > t_{\text{ff}} + t_{\text{su}} + t_{\text{d},i} + t_{\text{s},i+1}$

流水线分割点及级数的确定要考虑的因素

- 单元延迟时间及时钟频率的大小决定了数据通过速率
- 过多的级数不一定能产生最快的结果
- 太多寄存器的插入会导致芯片面积增加，布线困难，时钟偏差增加

