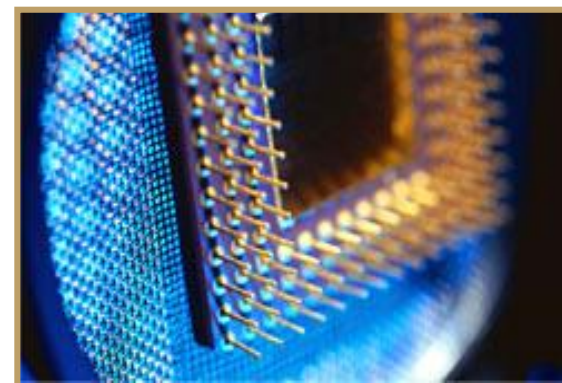
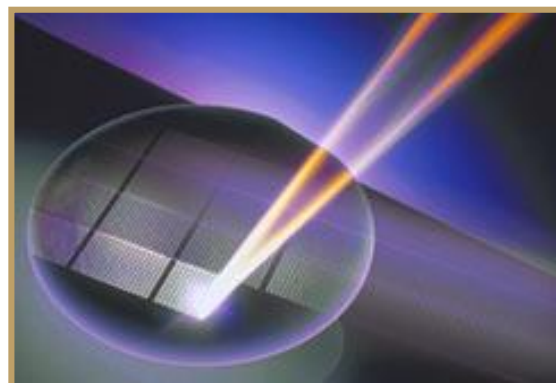
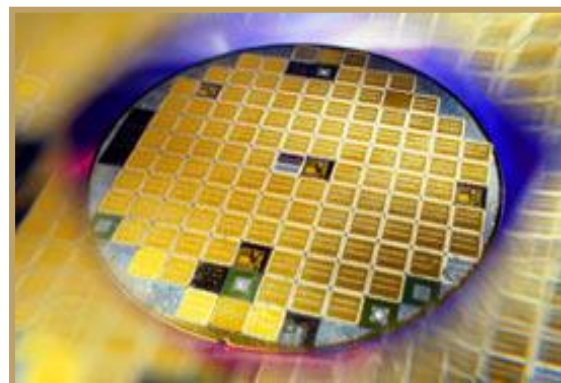




《VLSI数字通信原理与设计》课程

主讲人 贺光辉

## 第三章：流水线 and 并行处理



## 多媒体技术发展带来海量数据



模拟



数字



高清



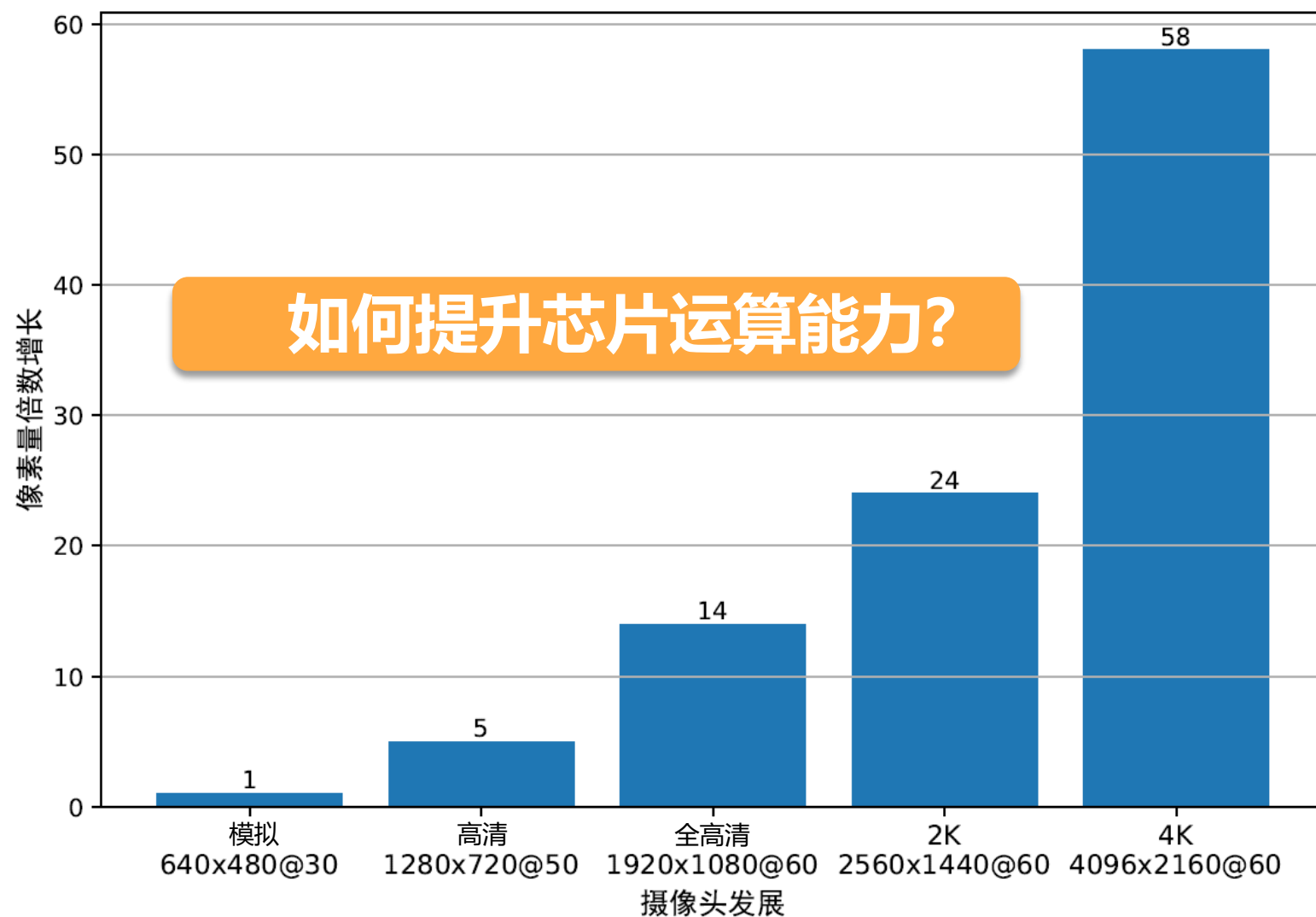
全高清



2K



4K



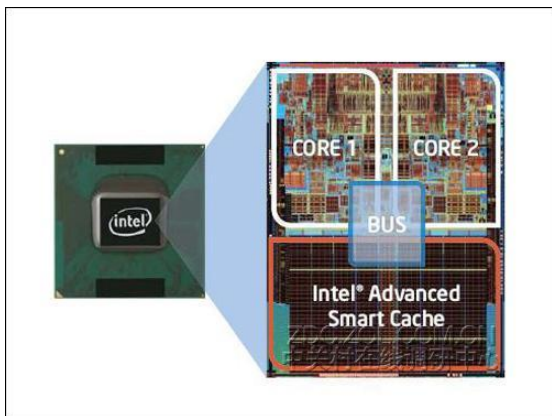
芯片运算能力  
成为关键因素



## 如何提升芯片运算能力？

- 并行系统：Paralleling

- L级并行系统在提升L倍硬件性能同时，硬件开销增加L倍。



高成本



- 流水线技术：Pipelining

- 相对增加较少硬件开销提升硬件性能



# 目录

**01** 并行处理技术

---

**02** 流水线技术

---

**03** 流水线技术降低功耗

---

**04** 并行处理技术降低功耗

---

**05** 本章总结

---





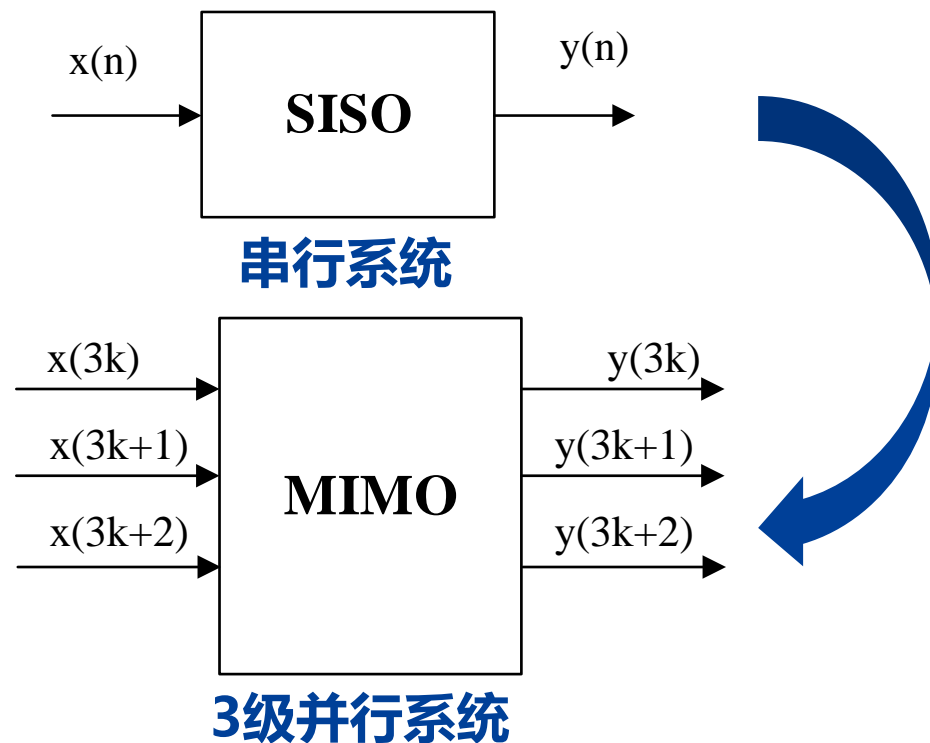
# 并行处理技术

现实生活中图像、人工智能等实际需求都要求硬件处理能力的提高。**并行处理技术**则是提高硬件处理能力常用方法之一。

**并行处理技术**：发掘了计算中的并发性，是同时性并发，复制硬件计算相同的任务。

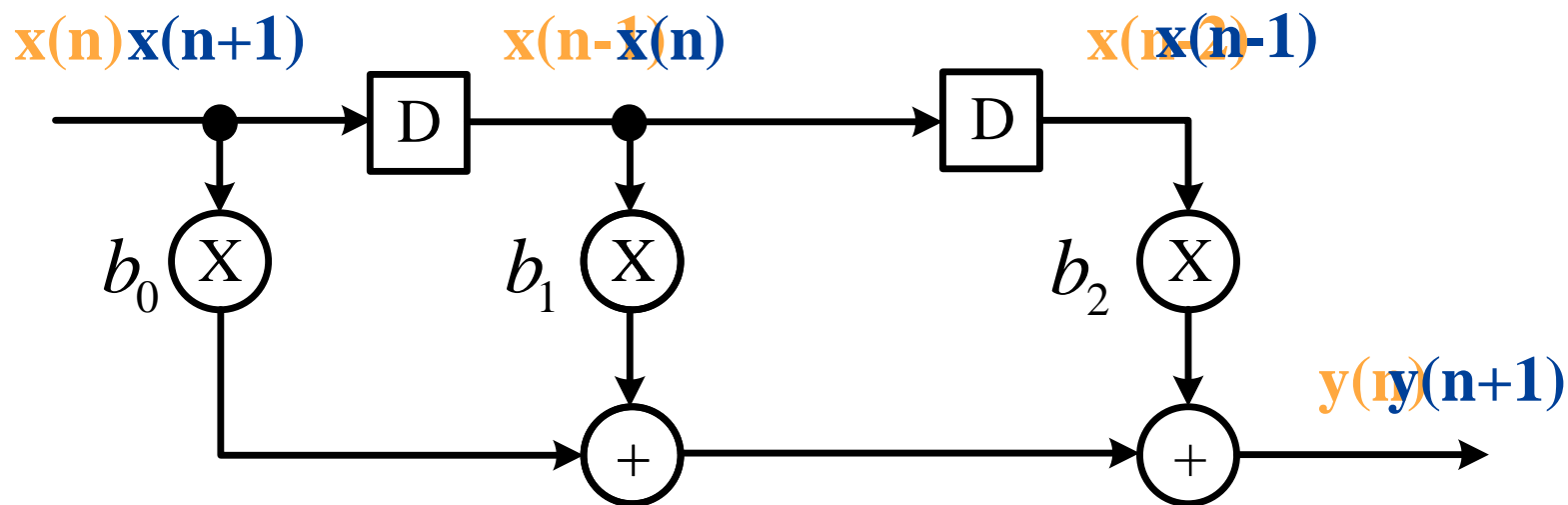
## 并行系统设计——3-Tap FIR滤波器

- 原始系统为**单输入单输出 (SISO)** 系统
- 并行处理：把**SISO**系统转换为**多输入多输出 (MIMO)** 系统，以便获得并行处理结构。例如每个时钟3个输入的系统 ( $k$ 为时钟标号,  $k=3$ )



## 未并行的三抽头滤波器


$$\begin{cases} y(n) = b_0x(n) + b_1x(n-1) + b_2x(n-2) \\ y(n+1) = b_0x(n+1) + b_1x(n) + b_2x(n-1) \end{cases}$$



## ■ 三抽头滤波器的两级并行

$$\begin{cases} y(n) = b_0x(n) + b_1x(n-1) + b_2x(n-2) \\ y(n+1) = b_0x(n+1) + b_1x(n) + b_2x(n-1) \end{cases}$$

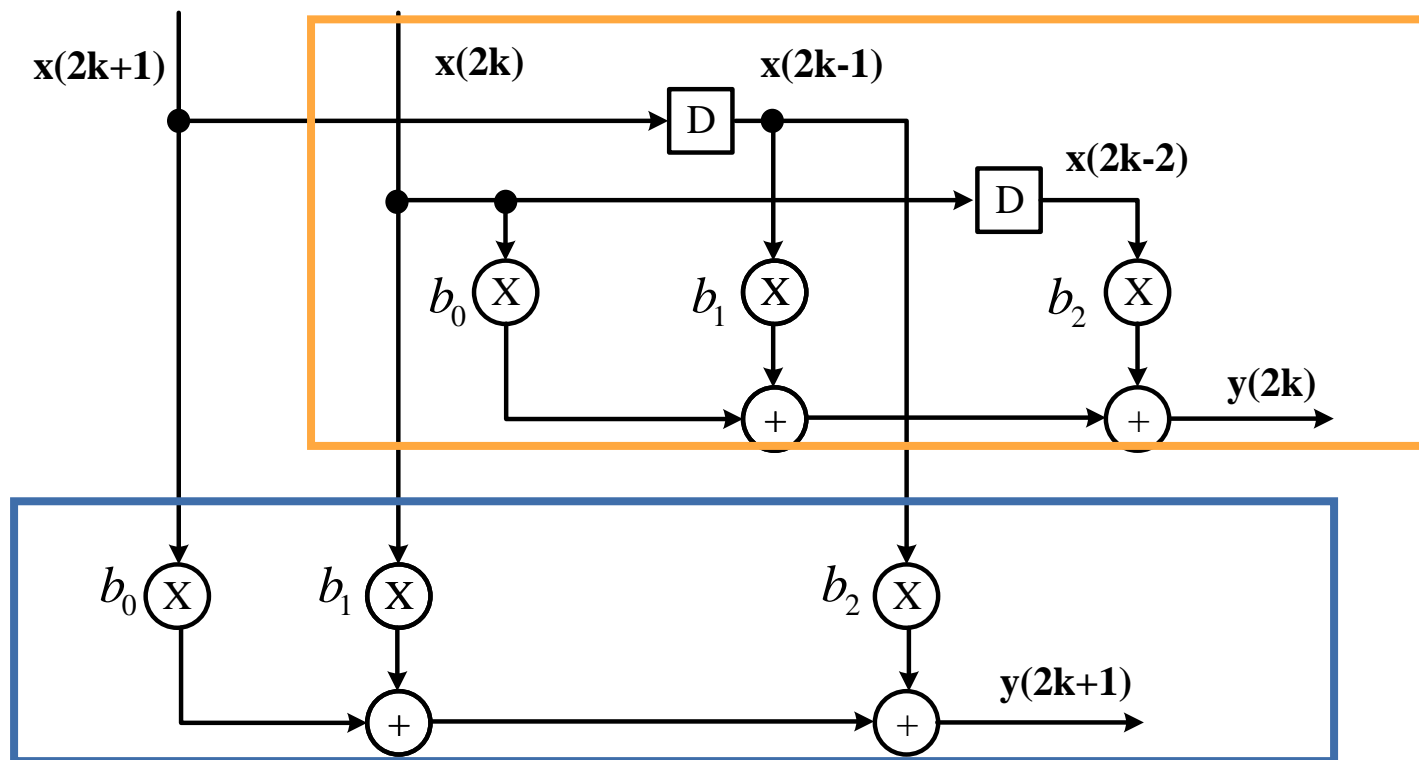
  
n changed to 2k


$$\begin{cases} y(2k) = b_0x(2k) + b_1x(2k-1) + b_2x(2k-2) \\ y(2k+1) = b_0x(2k+1) + b_1x(2k) + b_2x(2k-1) \end{cases}$$



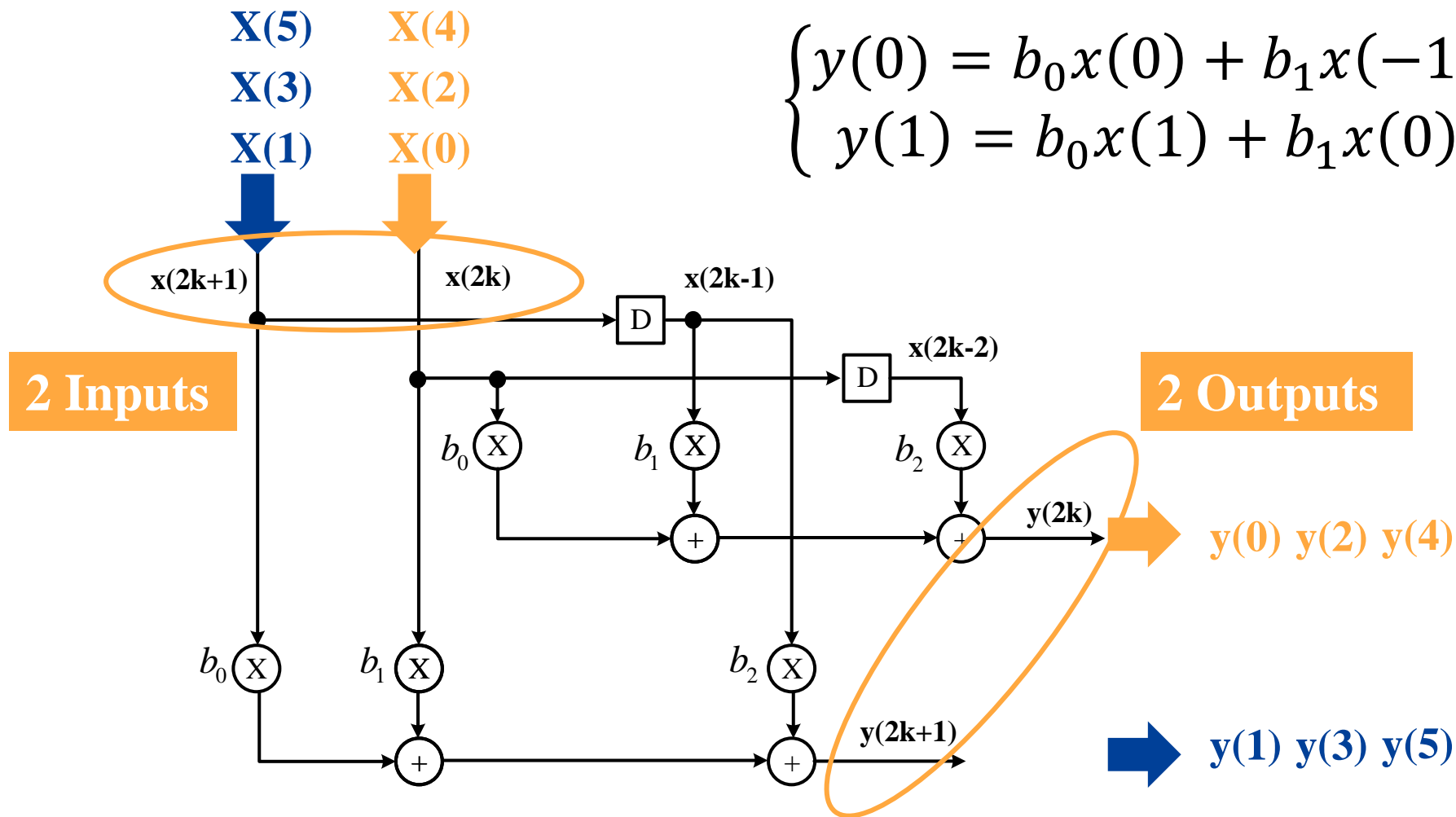
## 三抽头滤波器的两级并行

$$\begin{cases} y(2k) = b_0x(2k) + b_1x(2k-1) + b_2x(2k-2) \\ y(2k+1) = b_0x(2k+1) + b_1x(2k) + b_2x(2k-1) \end{cases}$$



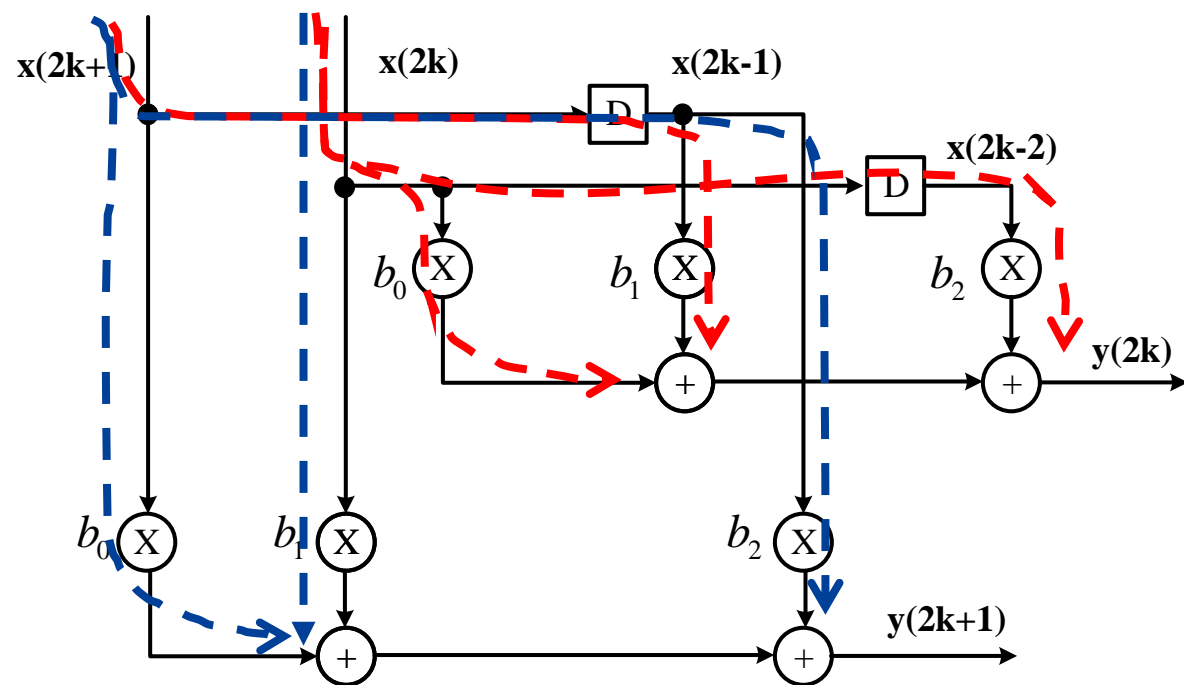
## 三抽头滤波器的两级并行

$$\begin{cases} y(0) = b_0x(0) + b_1x(-1) + b_2x(-2) \\ y(1) = b_0x(1) + b_1x(0) + b_2x(-1) \end{cases}$$



## 三抽头滤波器的两级并行

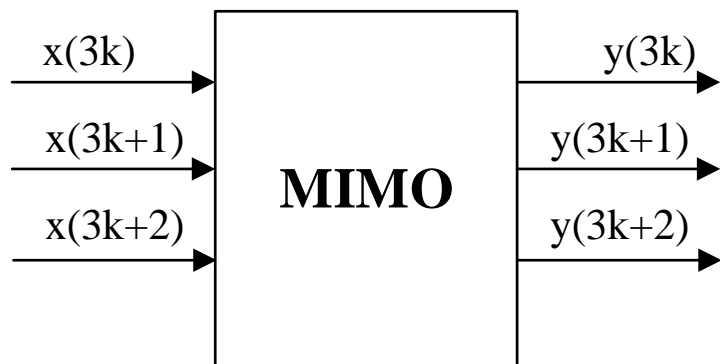
$$\begin{cases} y(0) = b_0x(0) + b_1x(-1) + b_2x(-2) \\ y(1) = b_0x(1) + b_1x(0) + b_2x(-1) \end{cases}$$



# 并行系统设计——3-Tap FIR滤波器



串行系统



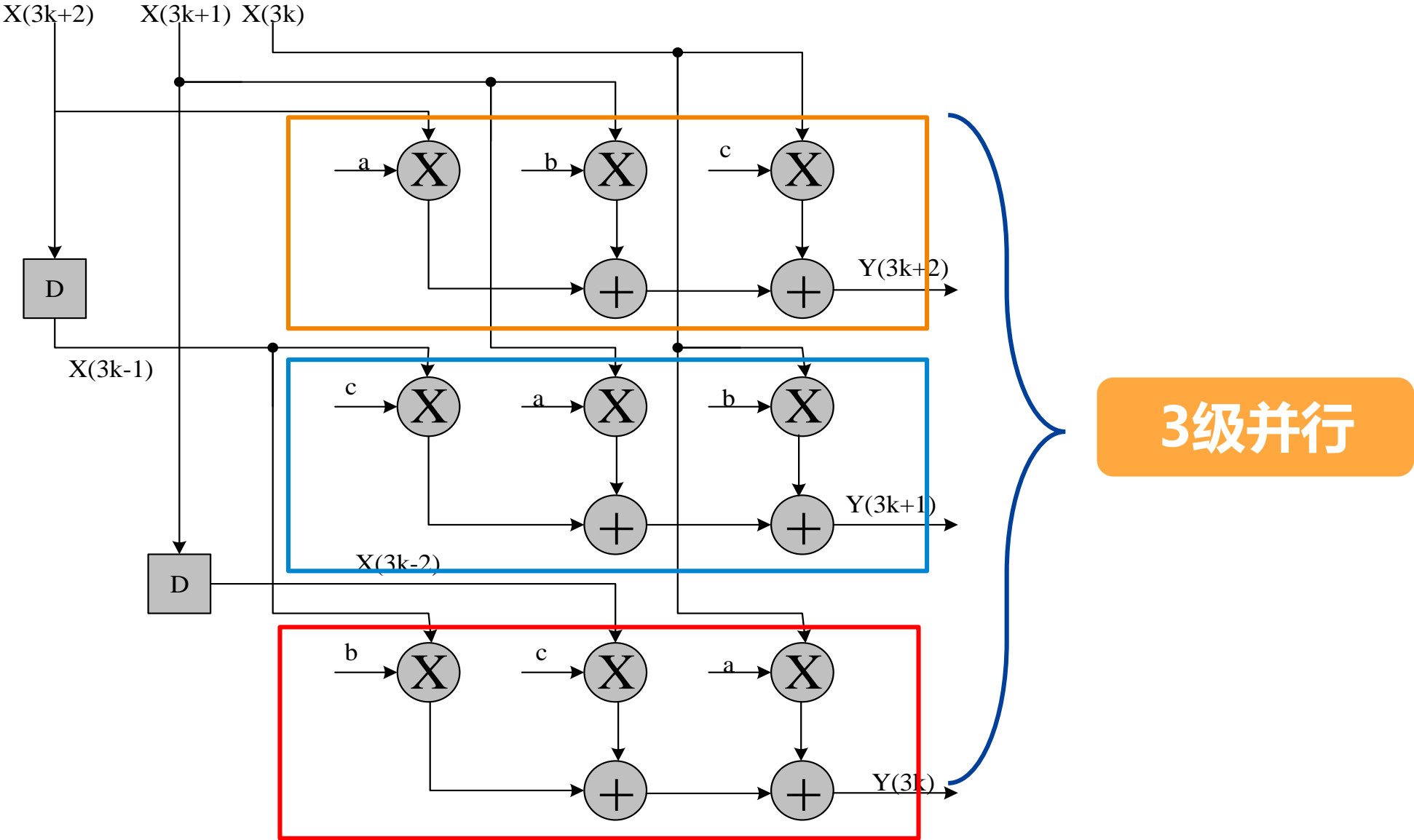
3级并行系统

$$y(n) = ax(n) + bx(n-1) + cx(n-2)$$



$$\begin{aligned} y(3k) &= ax(3k) + bx(3k-1) + cx(3k-2) \\ y(3k+1) &= ax(3k+1) + bx(3k) + cx(3k-1) \\ y(3k+2) &= ax(3k+2) + bx(3k+1) + cx(3k) \end{aligned}$$

# 3级并行系统的实现



# 并行系统的讨论

■ 并行处理系统的关键路径保持不变

■ L级并行系统的一个时钟周期处理L个样点，则迭代（或采样）周期缩小为1/L

■ L级并行系统的时钟周期： $T_{clock} \neq T_{sample}$ ， $T_{clk} = LT_{sample}$

■ 样点序列处理

- 输入样点序列分块(预处理)：通过串-并(S-P)转换器
- 输出样点序列恢复(后处理)：通过并-串(P-S)转换器
- 一个完整的并行系统包括预处理和后处理
  - 串-并(S-P)转换器
  - 并-串(P-S)转换器





# 目录

**01** 并行处理技术

---

**02** 流水线技术

---

**03** 流水线技术降低功耗

---

**04** 并行处理技术降低功耗

---

**05** 本章总结

---

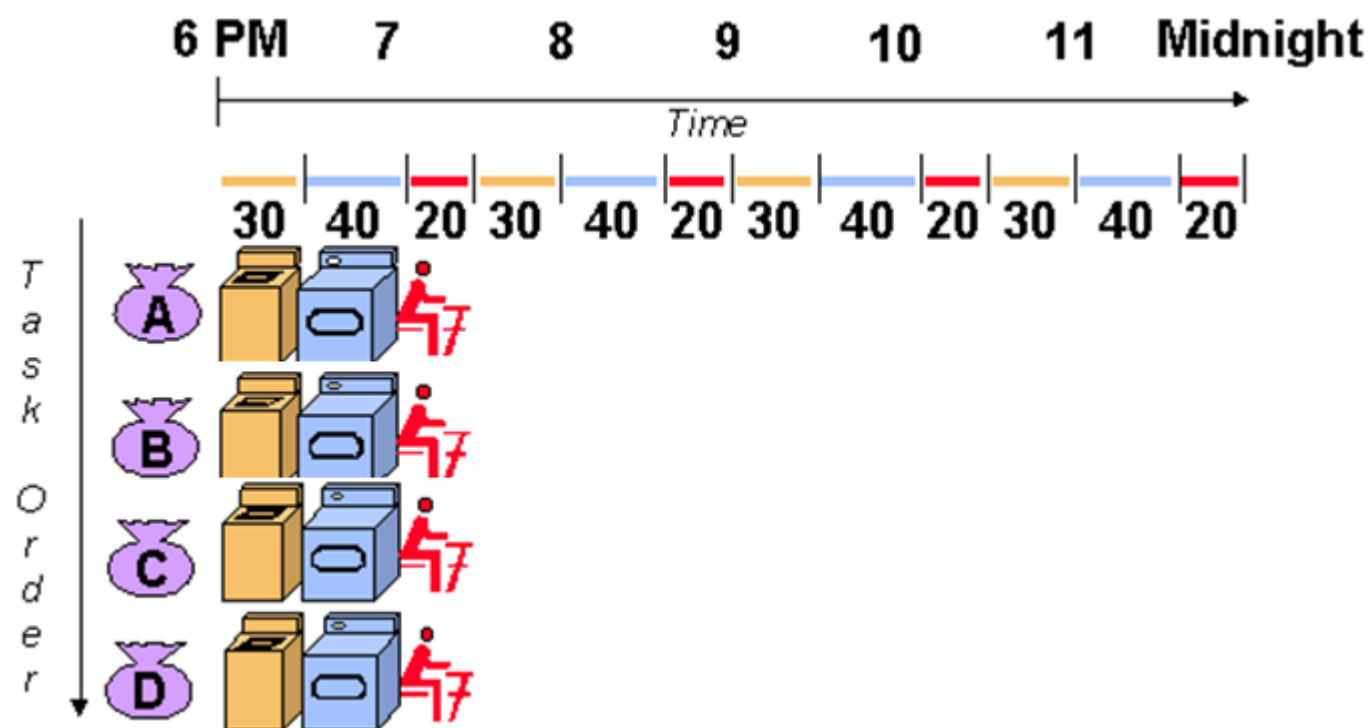


# 流水线技术的基本概念

流水线技术让不同模块能够并发工作，从而提高系统处理能力

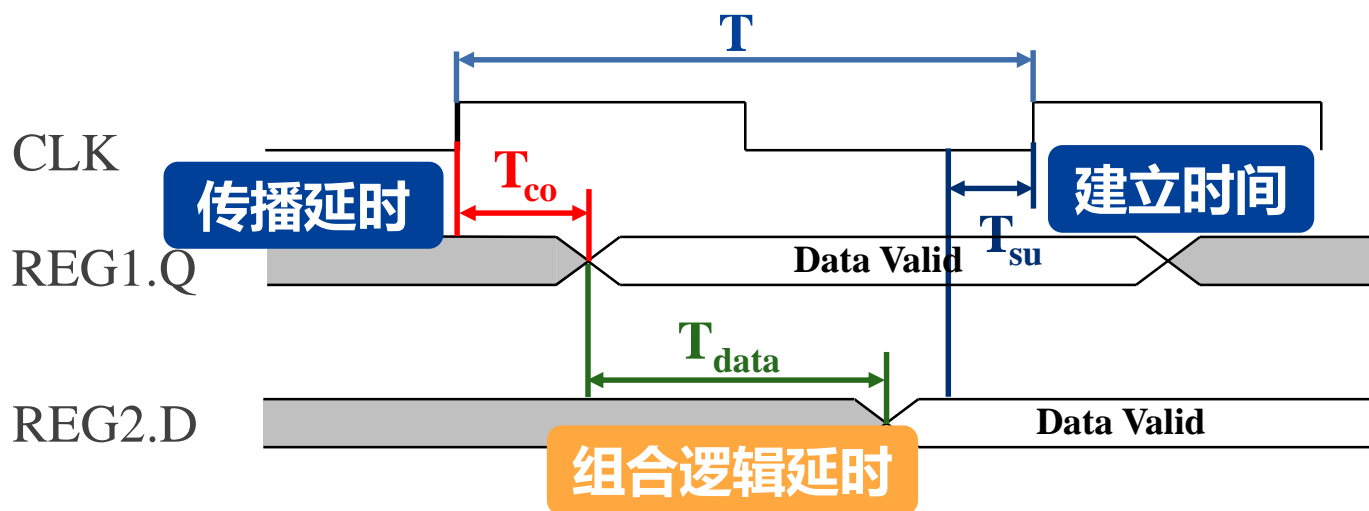
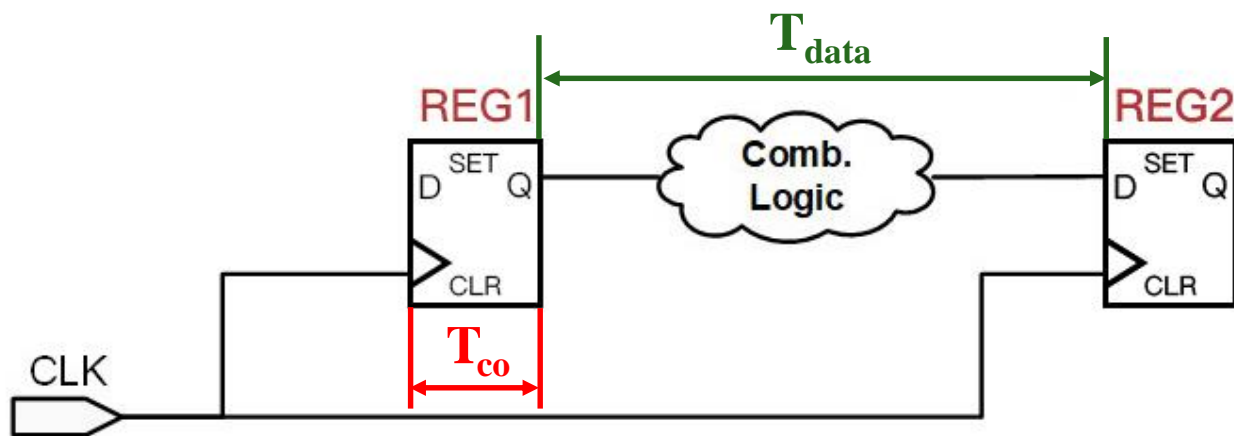
以洗衣房工作为例

采用流水线技术的洗衣房  
工作时间3.5小时



# 芯片设计中的流水线

通过在数据通路中插入寄存器，减少关键路径长度，从而提高时钟频率

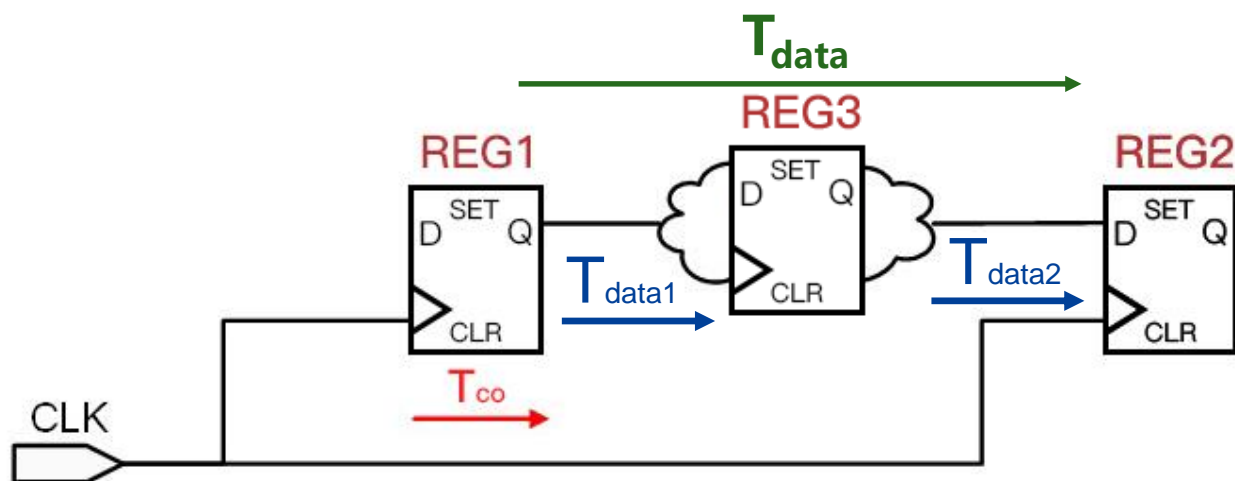


满足

$$T_{co} + T_{data} + T_{su} \leq T$$

# 芯片设计中的流水线

## 针对电路的时序要求



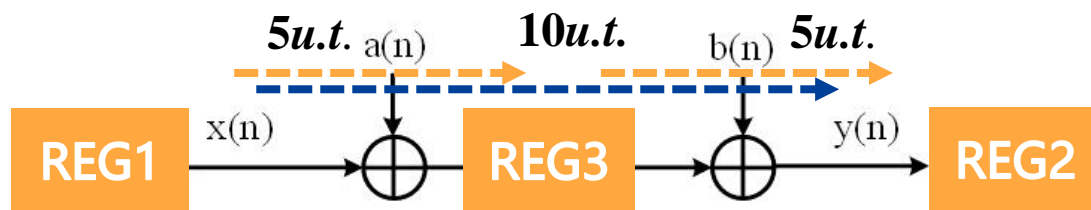
## 满足时序关系

$$T_{co} + T_{data1} + T_{su} \leq T$$

$$T_{co} + T_{data2} + T_{su} \leq T$$

$$T_{data1} + T_{data2} = T_{data}$$

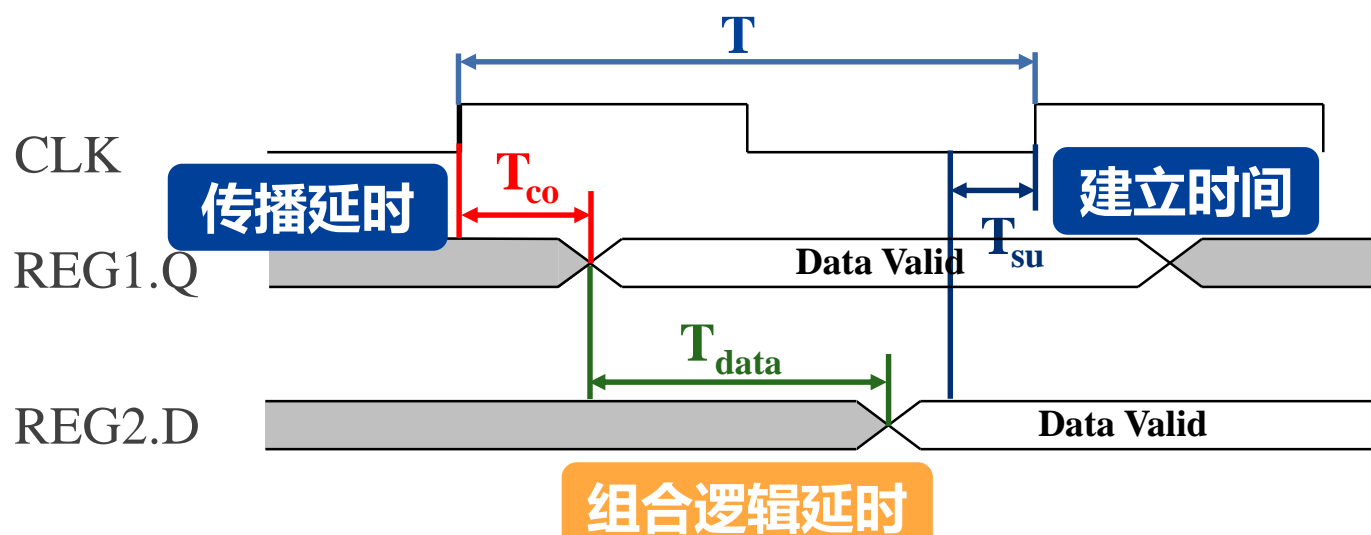
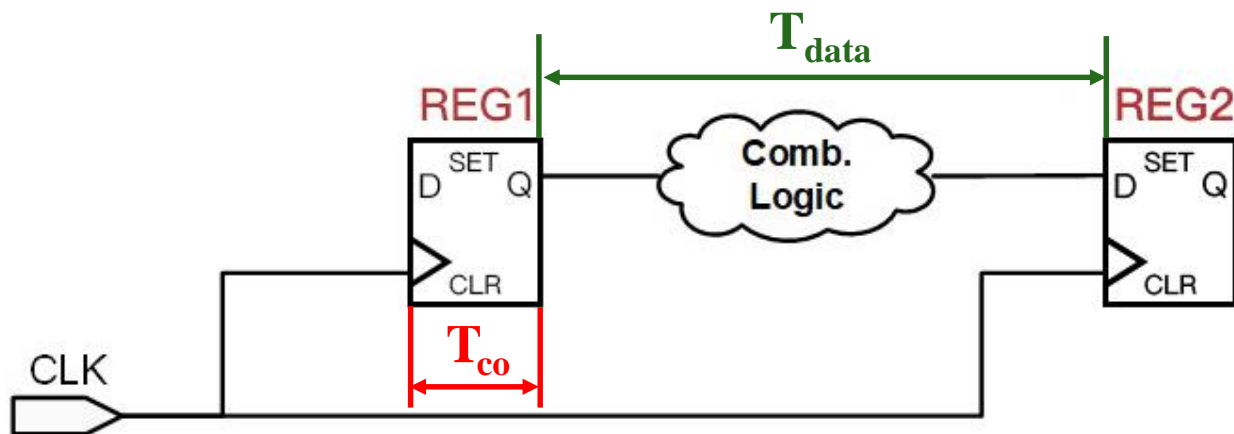
添加寄存器即将关键路径从  $10u.t.$  变为  $5u.t.$



时钟频率提升，  
增加输出延迟和寄存器

# 芯片设计中的流水线

通过在数据通路中插入寄存器，减少关键路径长度，从而提高时钟频率

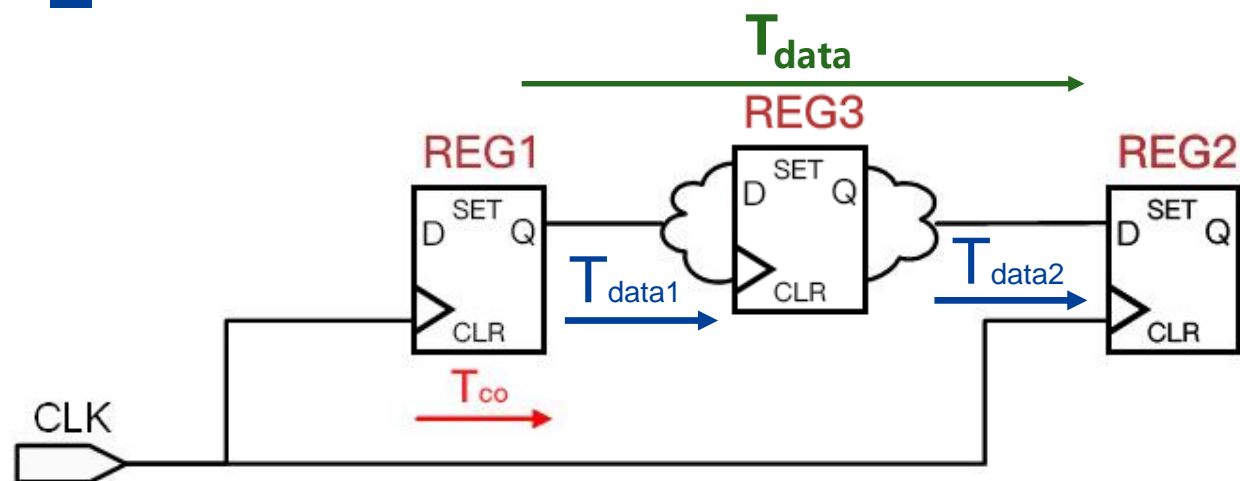


满足

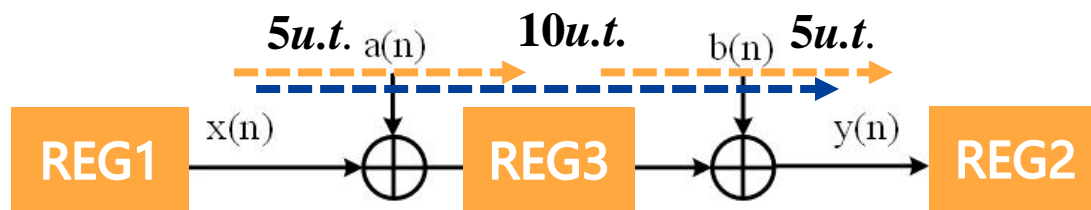
$$T_{co} + T_{data} + T_{su} \leq T$$

# 芯片设计中的流水线

■ 针对电路的时序要求，以下图中两个寄存器和一段组合逻辑为例介绍：



对于下面的简单电路，在关键路径添加寄存器即可将关键路径从10u.t.变为5u.t.。



满足时序关系：

$$T_{co} + T_{data1} + T_{su} \leq T$$

$$T_{co} + T_{data2} + T_{su} \leq T$$

$$T_{data1} + T_{data2} = T_{data}$$

时钟频率提升，  
增加输出延迟和寄存器

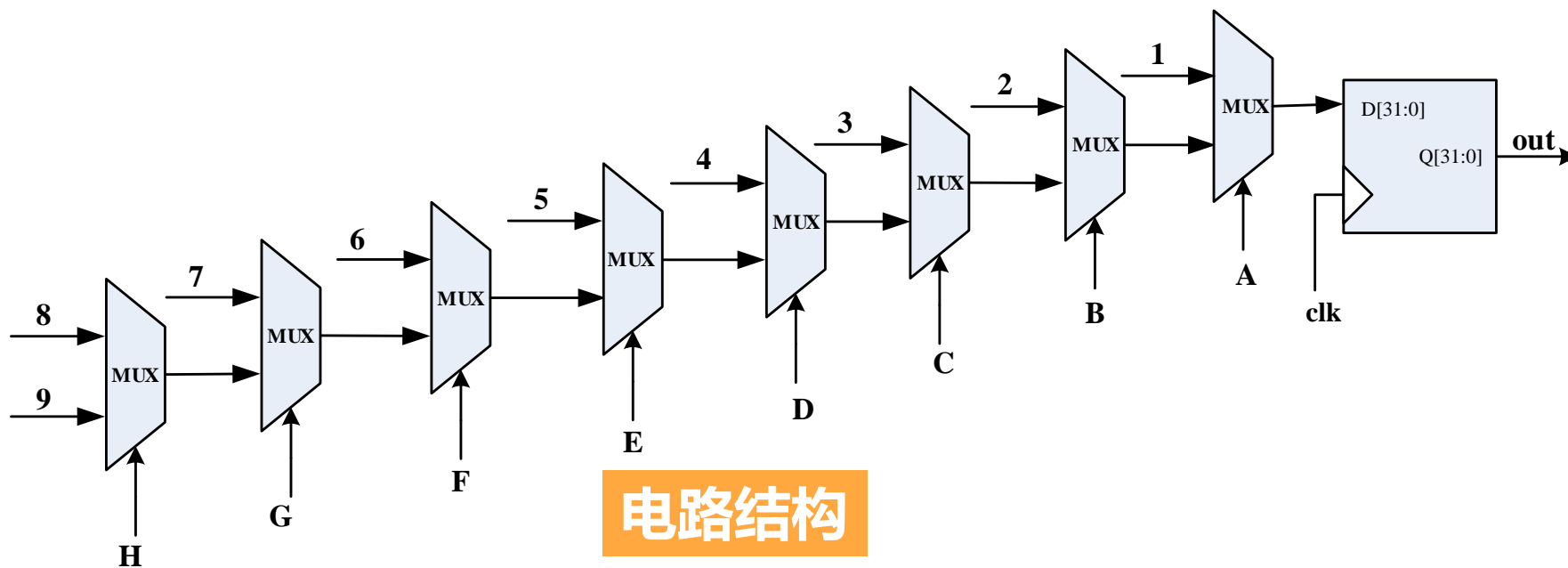


# 流水线技术的设计方法——多级MUX

## 在Xilinx V7-690T FPGA中实现多级MUX

```
always @(posedge clk)
  if(A)
    out <= 1;
  else if(B)
    out <= 2;
  else if(C)
    out <= 3;
  else if(D)
    out <= 4;
  else if(E)
    out <= 5;
  else if(F)
    out <= 6;
  else if(G)
    out <= 7;
  else if(H)
    out <= 8;
  else
    out <= 9;
```

时钟频率为  
400MHz



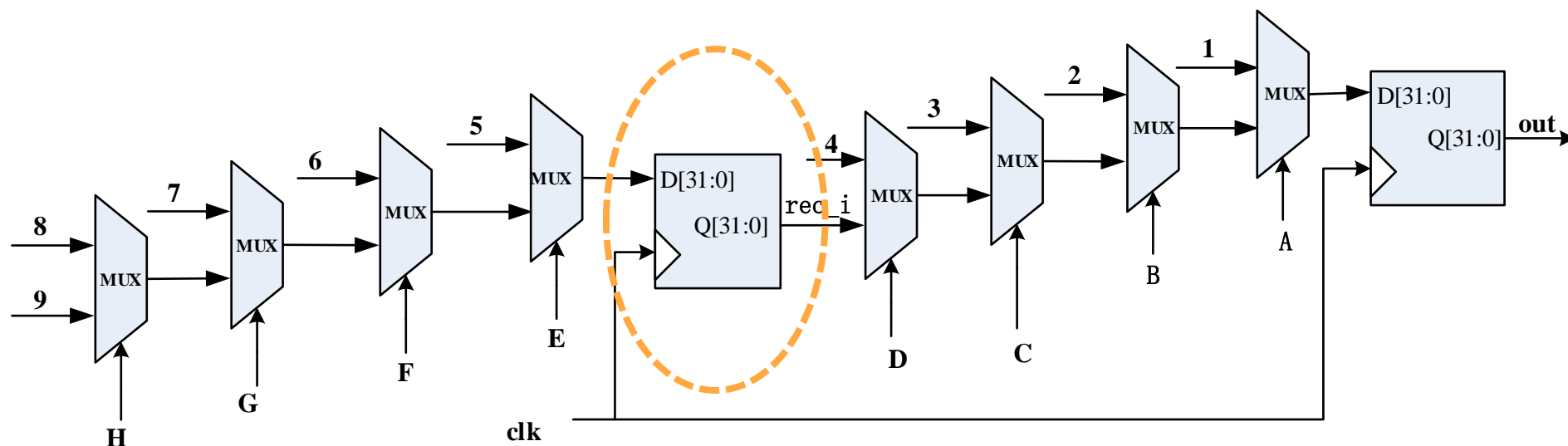
# 流水线技术的设计方法——多级MUX

## 在Xilinx V7-690T FPGA中实现多级MUX

```
always @(posedge clk)
  if(A)
    out <= 1;
  else if(B)
    out <= 2;
  else if(C)
    out <= 3;
  else if(D)
    out <= 4;
  else
    out <= reg_i;
```

```
always @(posedge clk)
  if(E)
    reg_i <= 5;
  else if(F)
    reg_i <= 6;
  else if(G)
    reg_i <= 7;
  else if(H)
    reg_i <= 8;
  else
    reg_i <= 9;
```

采用均等分割流水线

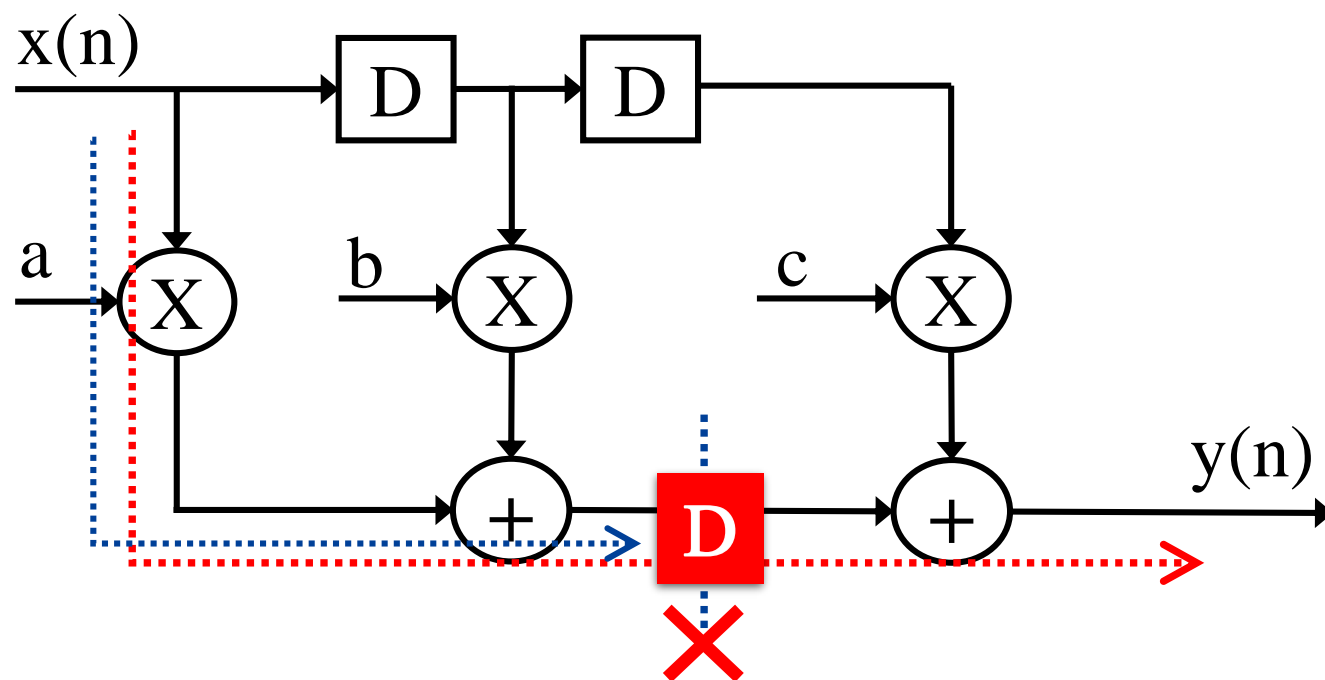


以多消耗寄存器为  
代价，时钟频率提  
升到650MHz

# 流水线技术的设计方法—— FIR滤波器

FIR滤波器的公式和直接型结构如下所示：

$$y(n) = ax(n) + bx(n-1) + cx(n-2)$$



关键路径为  $T_M + 2T_A$



关键路径为  $T_M + T_A$

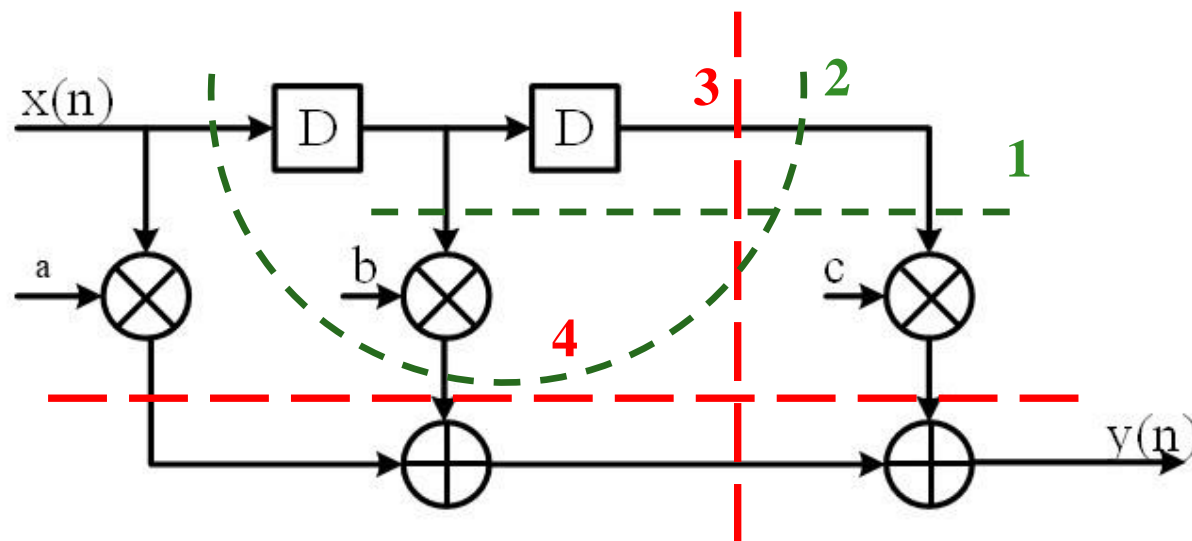
## 割集：

是图的一组边，若从图中移走这些边，则图被拆分为互不相连的两个子图或孤立节点

## 前馈割集：

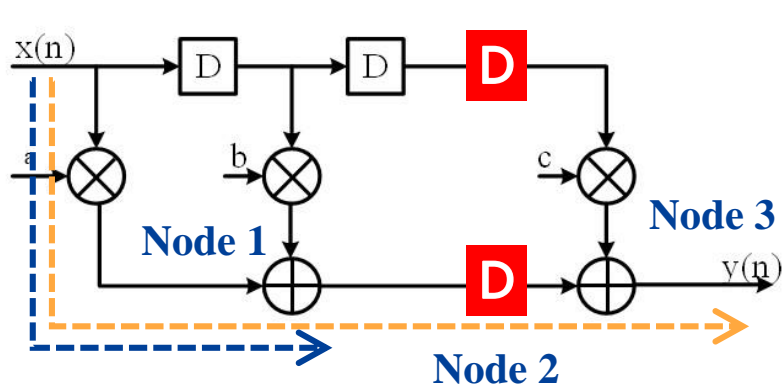
数据沿割集所有边前（同）向移动

- 绿线1不是割集
- 绿线2不是前馈割集
- 红线3、4均是前馈割集



# 流水线技术的设计方法—— FIR滤波器

## 前馈割集所有边插入寄存器，减少关键路径



关

优点

- 缩短关键路径
- 提高时钟频率

缺点

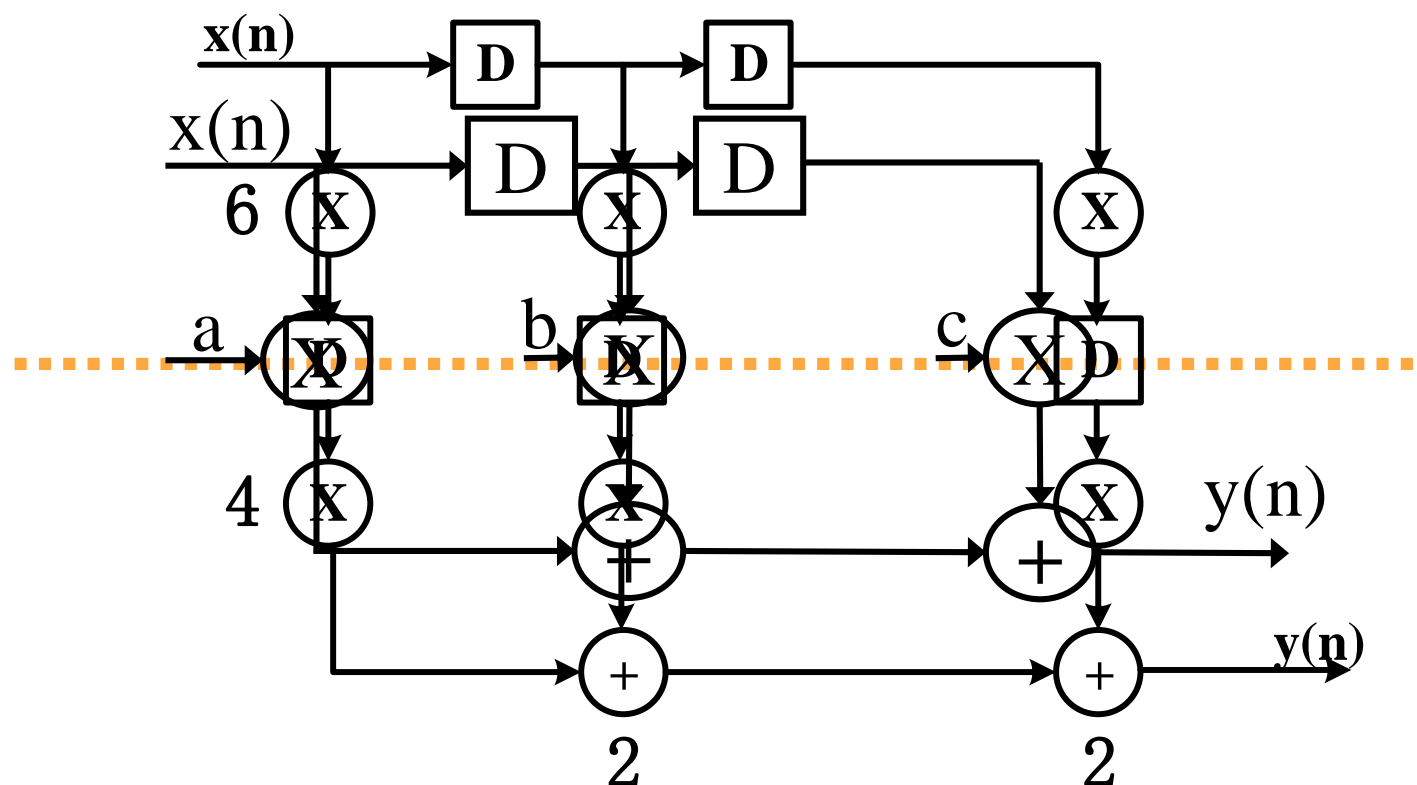
- 增加硬件开销
- 增加输出延迟

不影响算法功能： $y(n-1) = ax(n-1) + bx(n-2) + cx(n-3)$

Clock	Input	Node 1	Node 2	Node 3	Output
0	$x(0)$	$ax(0)$			
1	$x(1)$	$ax(1) + bx(0)$	$ax(0)$		
2	$x(2)$	$ax(2) + bx(1)$	$ax(1) + bx(0)$		
3	$x(3)$	$ax(3) + bx(2)$	$ax(2) + bx(1)$	$cx(0)$	$ax(2) + bx(1) + cx(0) = y(2)$

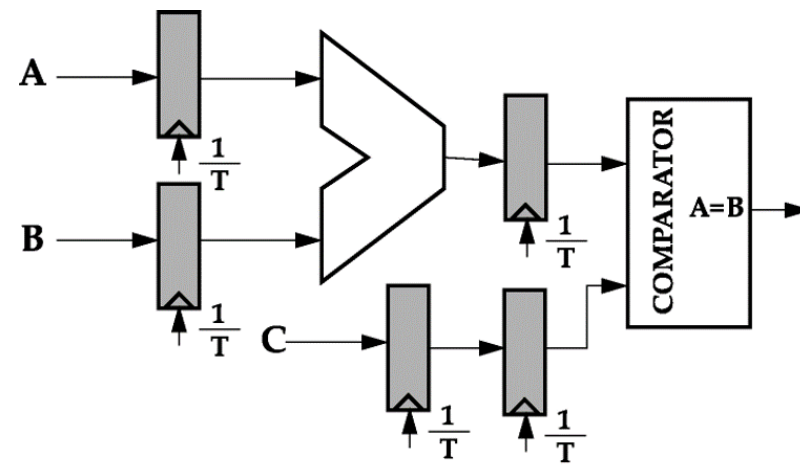
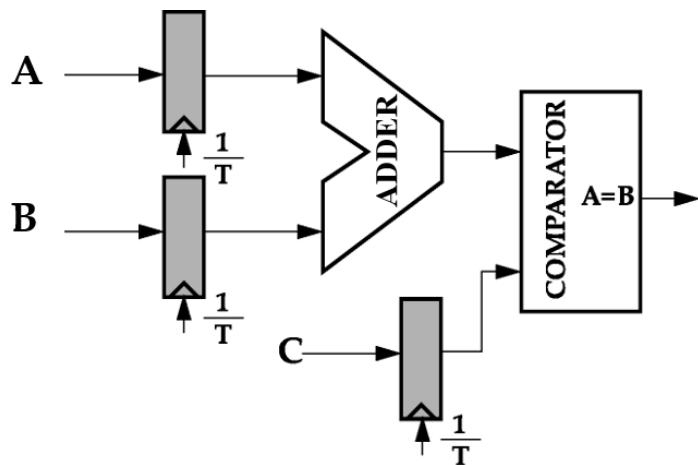
假设  $T_M = 10 \text{ u.t.}$   $T_A = 2 \text{ u.t.}$  并且预期时钟周期为  $6 \text{ u.t.}$

将乘法器分为两个更小的单元，使其处理时间分别为6和4个单位时间。

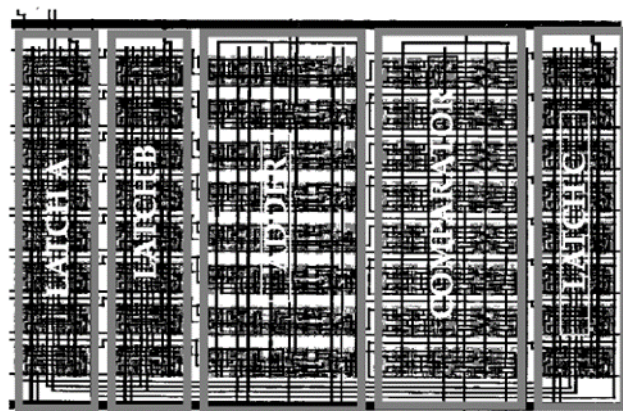




# 流水线技术的应用——加法器

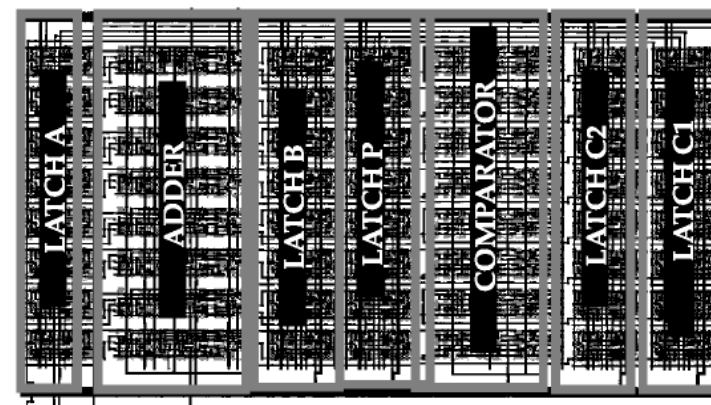


SMIC  
28nm



Area =  $636 \times 833 \text{ } \mu\text{m}^2$

面积变大

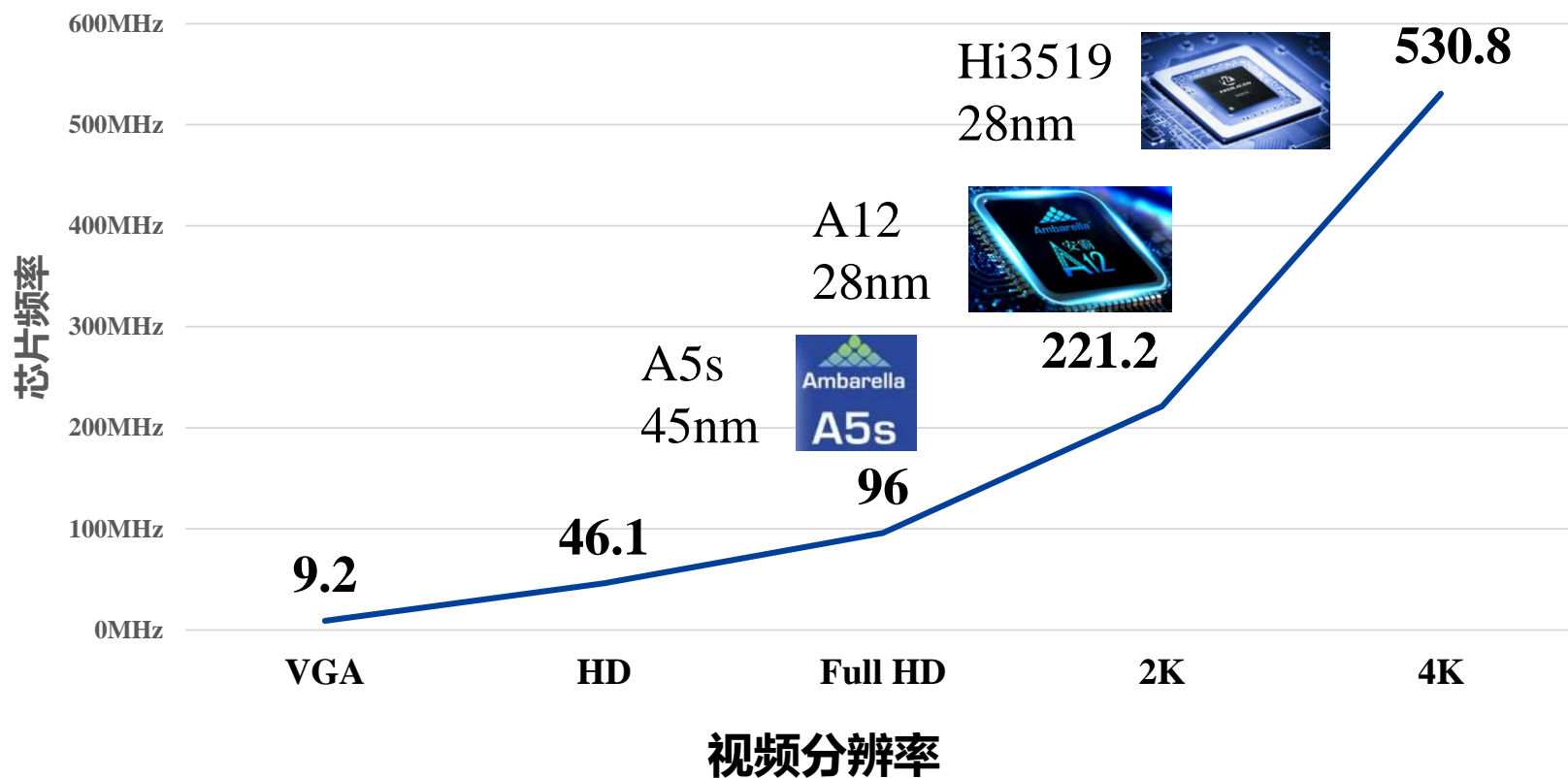


Area =  $640 \times 1081 \text{ } \mu\text{m}^2$

时钟频率 450MHz  $\longrightarrow$  550MHz

# 流水线技术的应用——视频处理芯片

## 芯片工作频率-视频分辨率关系图



芯片如何支持高分辨率？



流水线+工艺提升

## 流水线

- 关键路径的减少

- 关键路径指DFG中无延迟路径中最长的路径
- 关键路径可以通过适当插入寄存器来减少
- 流水线锁存器的插入不得改变系统功能：需采用前馈割集方法

- 在M级流水线系统中，从输入到输出任一路径的延迟数比原系统中同一路径的延迟数多(M-1)。

# 流水线技术的优缺点

■ 优点：减少时钟周期，提高系统时钟频率

■ 缺点：

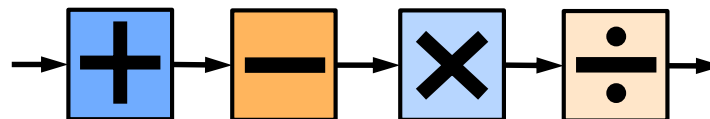
- 增加寄存器硬件开销

- 增加迟滞时间(Latency)：迟滞时间是指从第一个样点输入系统到系统输出处理完的第一个样点所需要的延迟数。

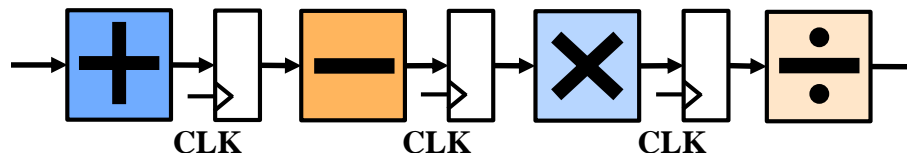
■ 适用于大量数据连续输入的情况，不适用于数据非连续输入的情况，会产生“流水气泡”。

# 流水线技术和并行技术的比较

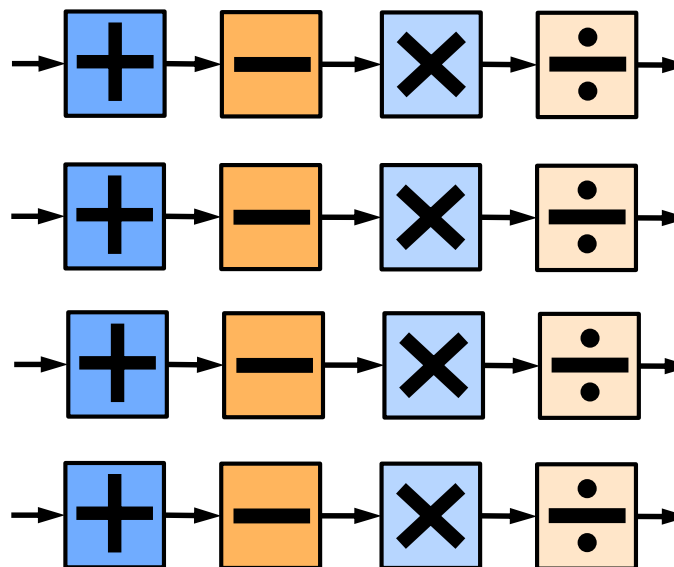
寄存器间的组合逻辑运算为：



流水线处理方式

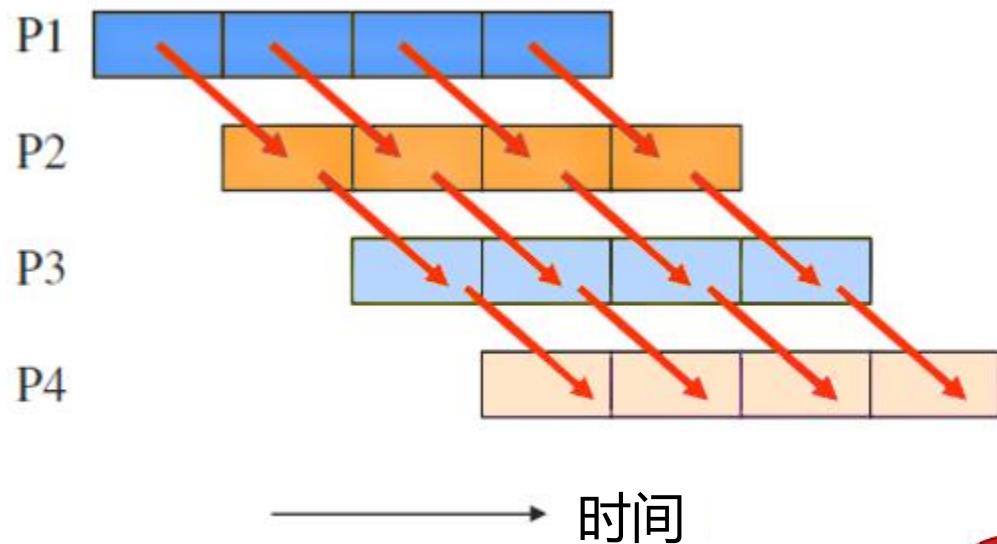


并行处理方式



# 流水线技术和并行技术的比较

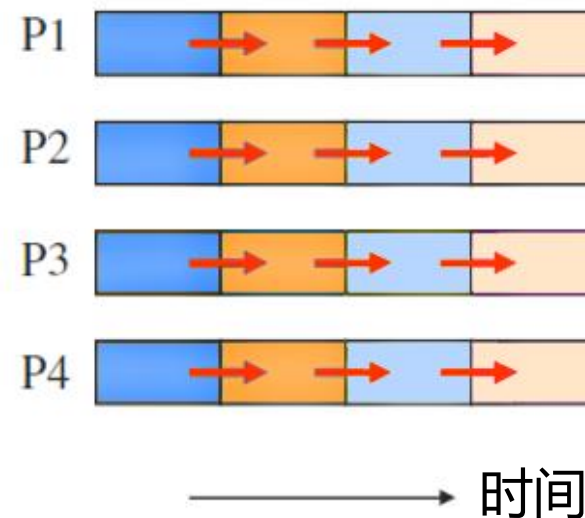
## 流水线



- 流水线交替式并发
- 各处理单元间需要通信



## 并行处理



- 并行处理同时并发
- 各处理单元无数据依赖

两种技术应用于什么场景？



# 为什么需要并行？

## 芯片间互连：

系统IO的延迟边界 $T_{io}$  = 输出延时 $T_{out}$  + 连线延时 $T_{comm}$  + 输入延时 $T_{in}$

**通信受限系统：** 关键路径 $T_c \leq$  延迟边界 $T_{io}$

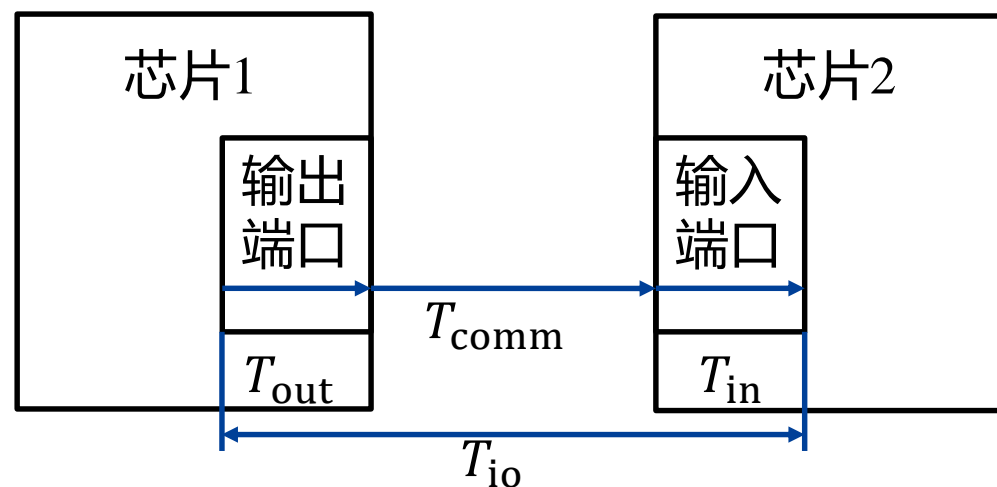
- 对通信未受限系统 ( $T_c \geq T_{io}$ )

**流水线技术**适用；更节省硬件

- 对于通信受限系统 ( $T_c \leq T_{io}$ )

**流水线技术**无法提高传输速率

**并行技术**通过增加IO数量提高传输速率





# 目录

**01** 并行处理技术

---

**02** 流水线技术

---

**03** 流水线技术降低功耗

---

**04** 并行处理技术降低功耗

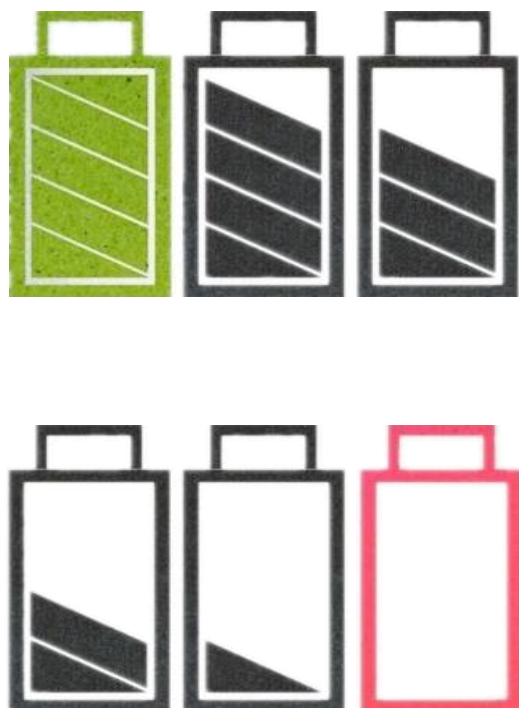
---

**05** 本章总结

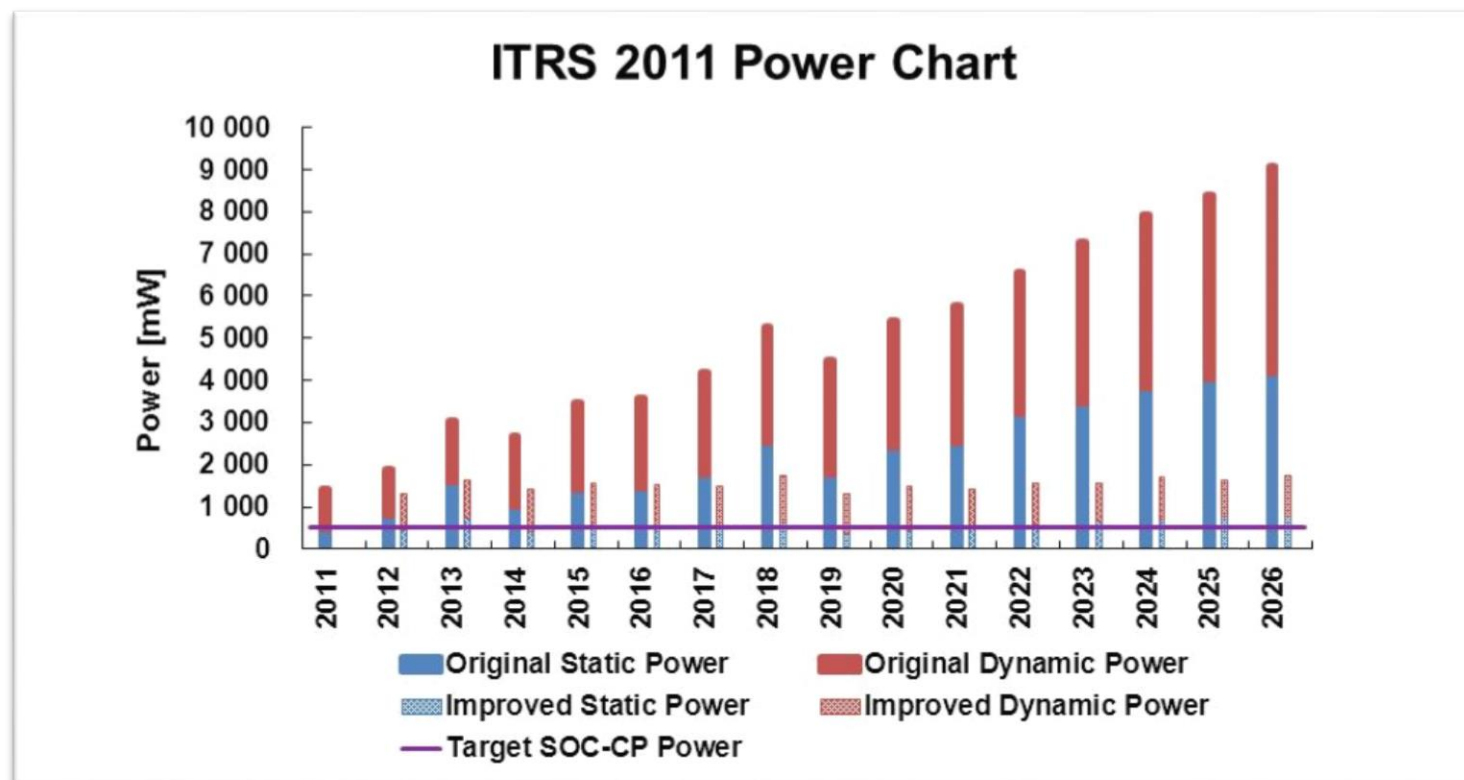
---



## 简介

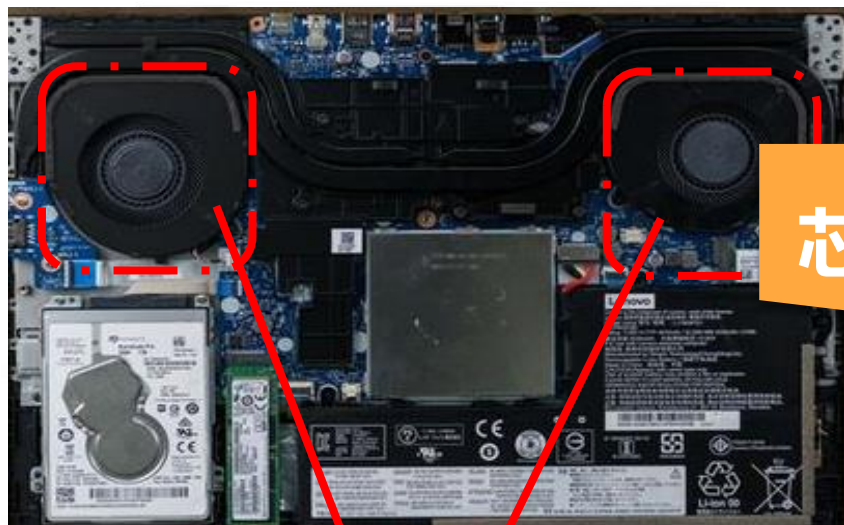


移动设备续航时间



低功耗设计对SOC移动设备功耗的影响

# 笔记本电脑的功耗

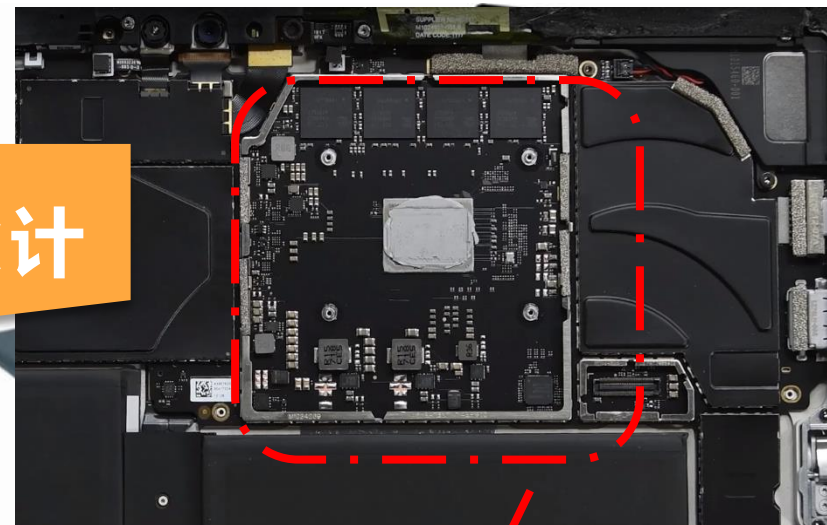


联想(Lenovo)拯救者Y7000 **i7-8750H**

**有风扇设计** 正常功耗芯片

重量 : 4kg  
电池容量: 52.5Whr  
待机时间: 5小时  
售价 : ¥ 6999

芯片低功耗设计



微软 Surface Pro **i7-7660U**

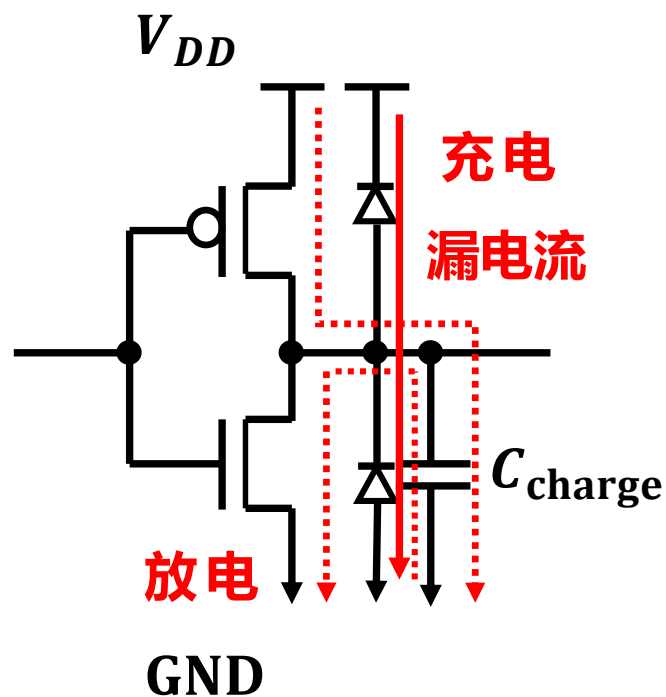
**无风扇设计** 低功耗芯片

重量 : 0.8kg  
电池容量: 45Whr  
待机时间: 13.5小时  
售价 : ¥ 11588

功耗

售价

# CMOS电路功耗组成



$$P_{total} = P_{dynamic} + P_{static}$$

总功耗      动态功耗      静态功耗

$$P_{static} = I_{leakage} \times V_{DD}$$

与工艺相关

$$P_{dynamic} = C_{total} \times V_{DD}^2 \times f$$

通过电路设计降低

一次完整的充放电, CMOS电路消耗能量为

$$\begin{aligned} E_{V_{DD}} &= \int_0^{\infty} I(t) V_{DD} dt = \int_0^{\infty} C_{charge} \frac{dV}{dt} V_{DD} dt \\ &= C_{charge} V_{DD} \int_0^{V_{DD}} dV = C_{charge} V_{DD}^2 \end{aligned}$$

# 传播延时与时钟周期

$$P_{\text{dynamic}} \downarrow = C_{\text{total}} V_{DD}^2 f \begin{cases} f \downarrow, \text{处理能力} \downarrow \\ V_{DD} \downarrow, T_{pd} \uparrow, f \downarrow, \text{处理能力} \downarrow \end{cases}$$

## 传播延时公式

充放电电容

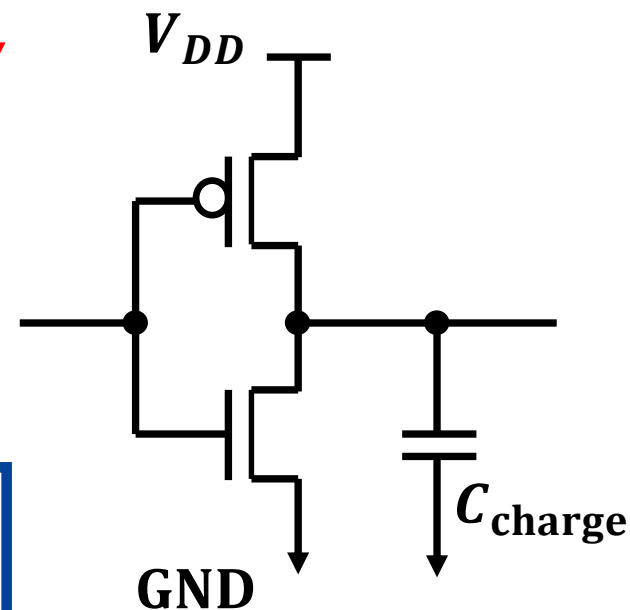
电源电压

$$T_{pd} = \frac{C_{\text{charge}} V_{DD}}{k (V_{DD} - V_t)^2}$$

$k$  为跨导因子

$V_t$  为MOS管阈值电压

$$f = \frac{1}{\max(T_{pd})}$$



# 流水线技术降低功耗——传播延时与时钟周期

$$P_{\text{dynamic}} \downarrow = C_{\text{total}} V_{DD}^2 f \begin{cases} f \downarrow, \text{处理能力} \downarrow \\ V_{DD} \downarrow, T_{pd} \uparrow, f \downarrow, \text{处理能力} \downarrow \end{cases}$$

## 传播延时公式

充放电电容

电源电压

$$T_{pd} = \frac{C_{\text{charge}} V_{DD}}{k (V_{DD} - V_t)^2}$$

$k$  为跨导因子

$V_t$  为MOS管阈值电压

$$f = \frac{1}{\max(T_{pd})}$$

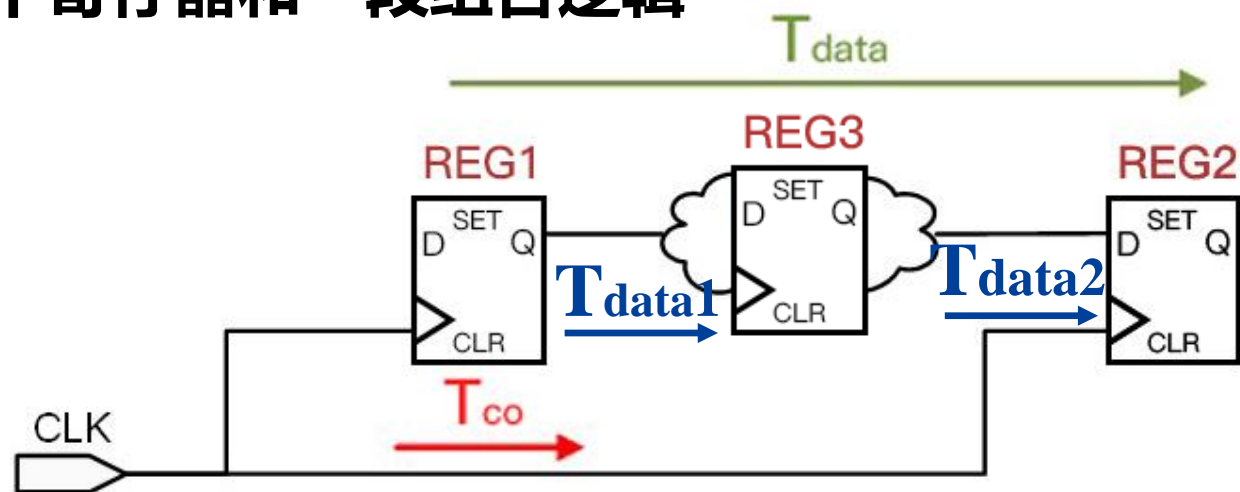
是否有方法可以不降低处理能力，而降低动态功耗？



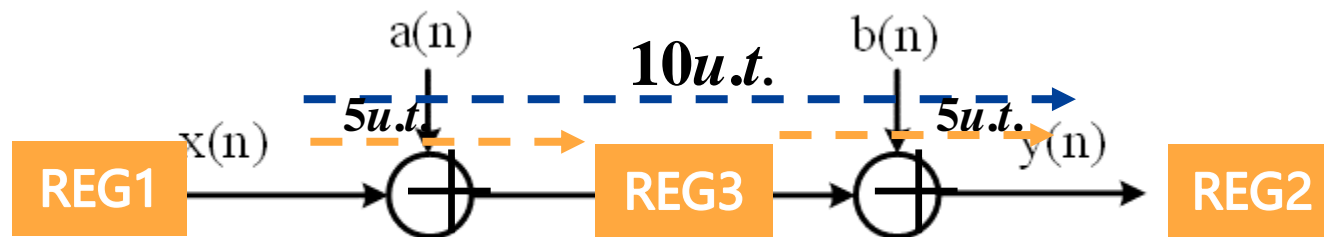
流水线技术

# 流水线技术降低功耗原理——流水线加快时钟频率

## 两个寄存器和一段组合逻辑



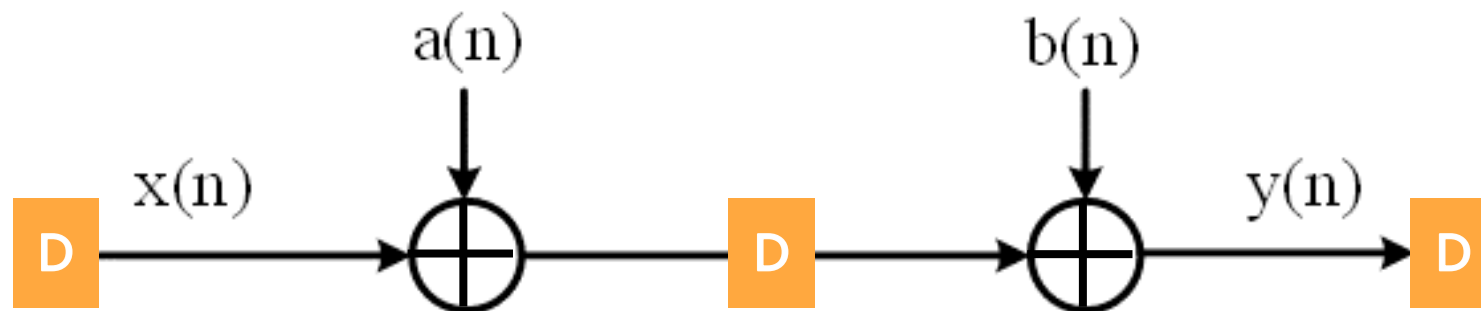
## 通过增加寄存器，缩短关键路径，提高时钟频率



频率提高



# 流水线技术降低功耗原理——电源电压降低



## 加速

- 关键路径减少 → 时钟频率提高 → 功耗增加

## 降低功耗

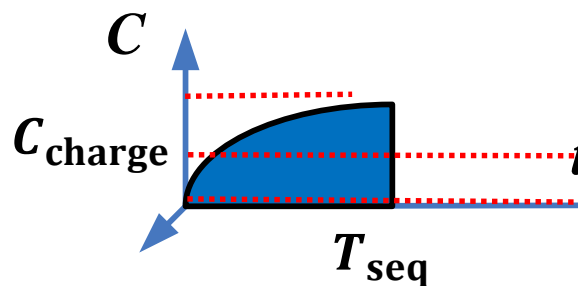
$$P \downarrow = C_{\text{total}} V_{DD}^2 \uparrow f \downarrow$$

$$T_{pd} = \frac{C_{\text{charge}} V_{DD}}{k(V_{DD} - V_t)^2}$$

- 时钟频率不变
- 电源电压  $V_{DD}$  降低，功耗降低

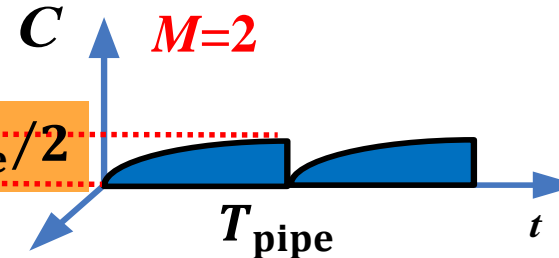
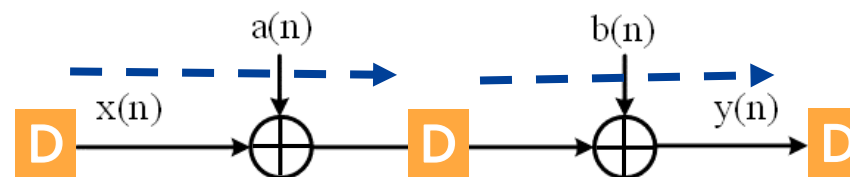
# 流水线技术降低功耗方法——求解电压和功耗

## 原始电路



$$V_{\text{seq}} = V_0$$

## 两级流水线电路



$$V_{\text{pipe}} = \beta V_0$$

$\beta$  为功耗降低因子

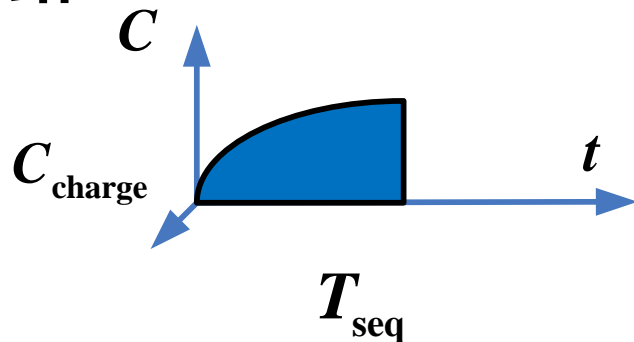
$$T_{\text{seq}} = \frac{C_{\text{charge}} V_0}{k(V_0 - V_t)^2}$$

=

$$T_{\text{pipe}} = \frac{\frac{C_{\text{charge}}}{M} \beta V_0}{k(\beta V_0 - V_t)^2}$$

# 流水线技术降低功耗方法——求解功耗降低因子

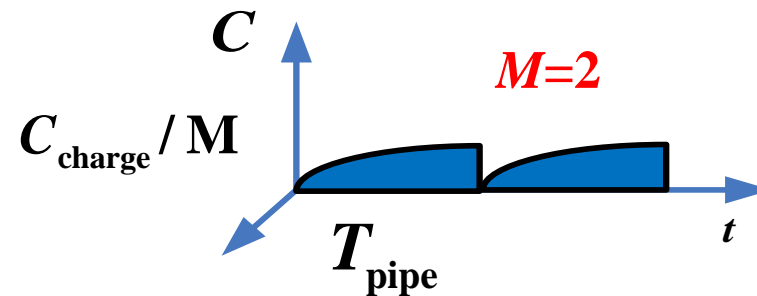
## 原始电路



时钟周期

$T_{\text{seq}}$   $V_{\text{seq}} = V_0$

## 两级流水线电路



$V_{\text{pipe}} = \beta V_0$

$T_{\text{pipe}}$   $T_{\text{pipe}}$

得到含 $\beta$ 的二次方程

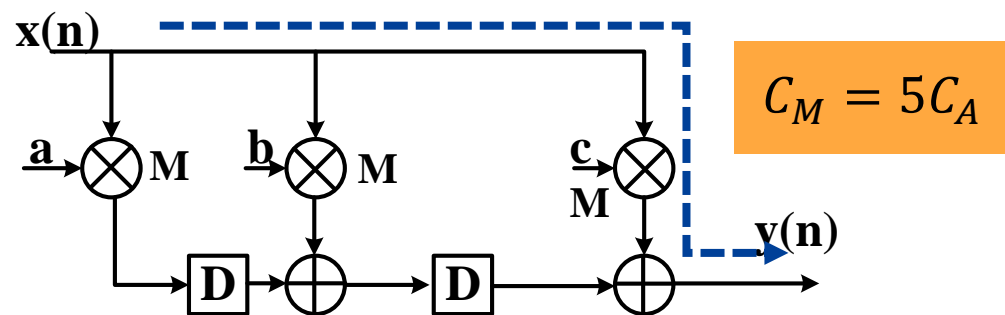
$$M(\beta V_0 - V_t)^2 = \beta(V_0 - V_t)^2 \rightarrow \beta$$

采用流水线处理后功耗为

$$P_{\text{pipe}} = C_{\text{total}} (\beta V_0)^2 f = \beta^2 P_{\text{seq}}$$

# 流水线技术降低功耗应用——FIR滤波器

## 原始滤波器

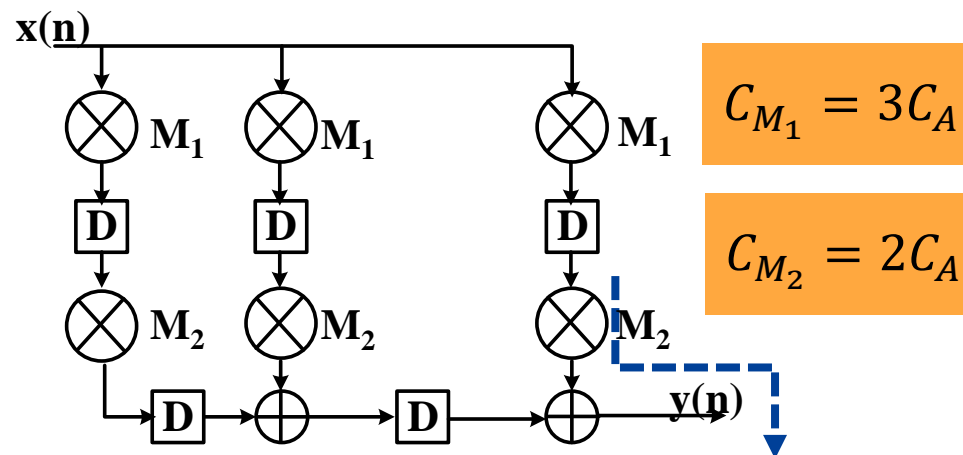


$$V_0 = 5V$$

$$V_t = 0.6V$$

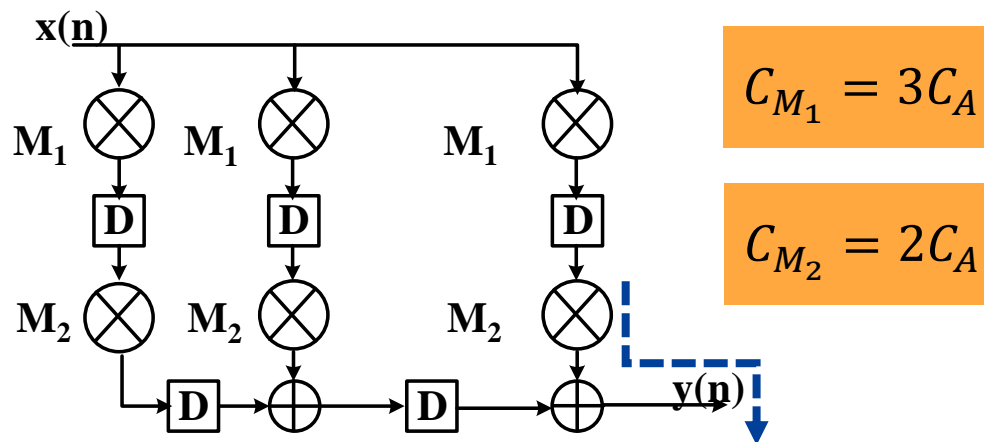
$$C_{\text{charge}} = C_A + C_M = 6C_A$$

## 两级流水滤波器



$$C_{\text{charge}'} = C_{M_1} = C_{M_2} + C_A = 3C_A$$

## 两级流水滤波器



$$C_{M_1} = 3C_A$$

$$C_{M_2} = 2C_A$$

$$C_{\text{charge}'} = C_{M_1} = C_{M_2} + C_A = 3C_A$$

$$M(\beta V_{DD} - V_t)^2 = \beta(V_{DD} - V_t)^2$$

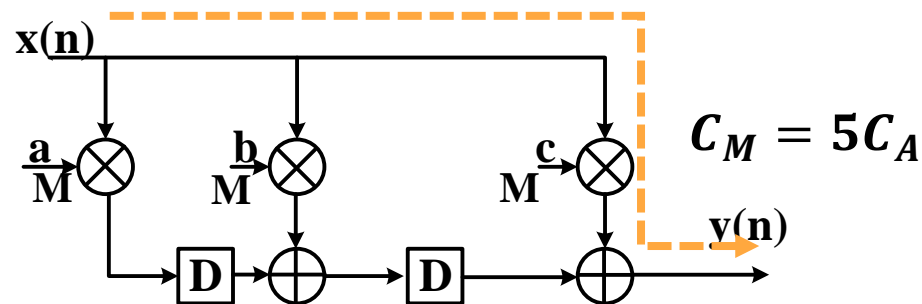
$$2(5\beta - 0.6)^2 = \beta(5 - 0.6)^2$$



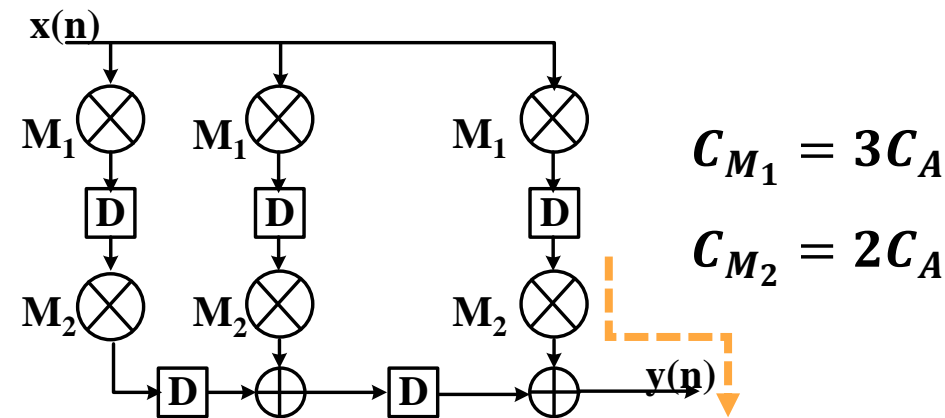
$$\beta = 0.6033 \text{ 或 } 0.0239 \text{ (舍弃)}$$

新的动态功耗为:  $P_{\text{pipe}} = \beta^2 P_{\text{seq}} = 0.364 P_{\text{seq}}$

原始滤波器



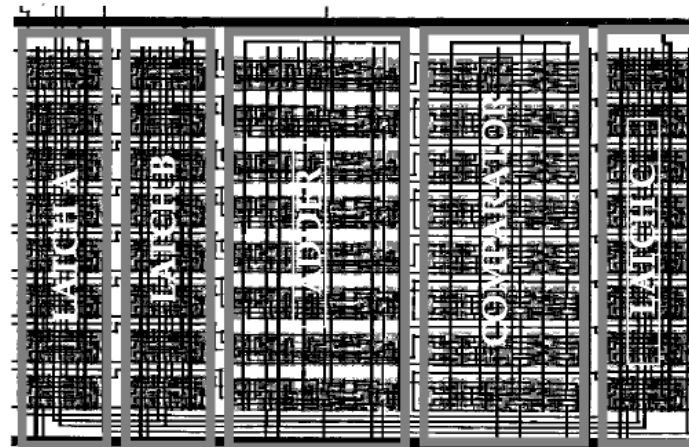
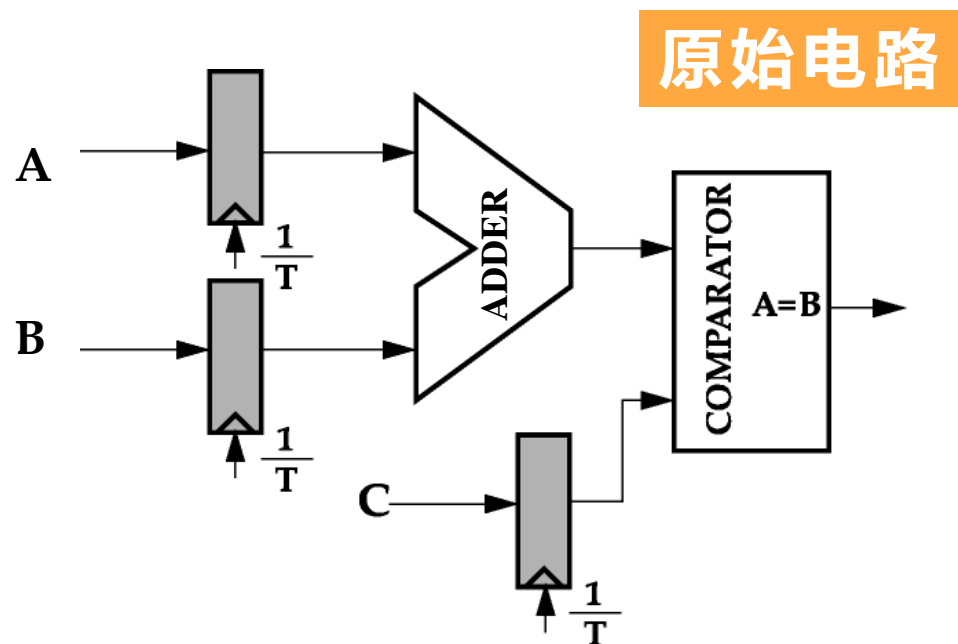
两级流水滤波器



乘法器计算时间 $10u.t.$ ，加法器计算时间 $2u.t.$

系统	原始滤波器	两级流水滤波器 (不降功耗)	两级流水滤波器 (降功耗)
时钟周期 ( $u.t.$ )	$12u.t.$	$6u.t.$	$12u.t.$
功耗	$P_{seq}$	$2P_{seq}$	$0.364P_{seq}$

# 流水线技术降低功耗应用——加法比较电路



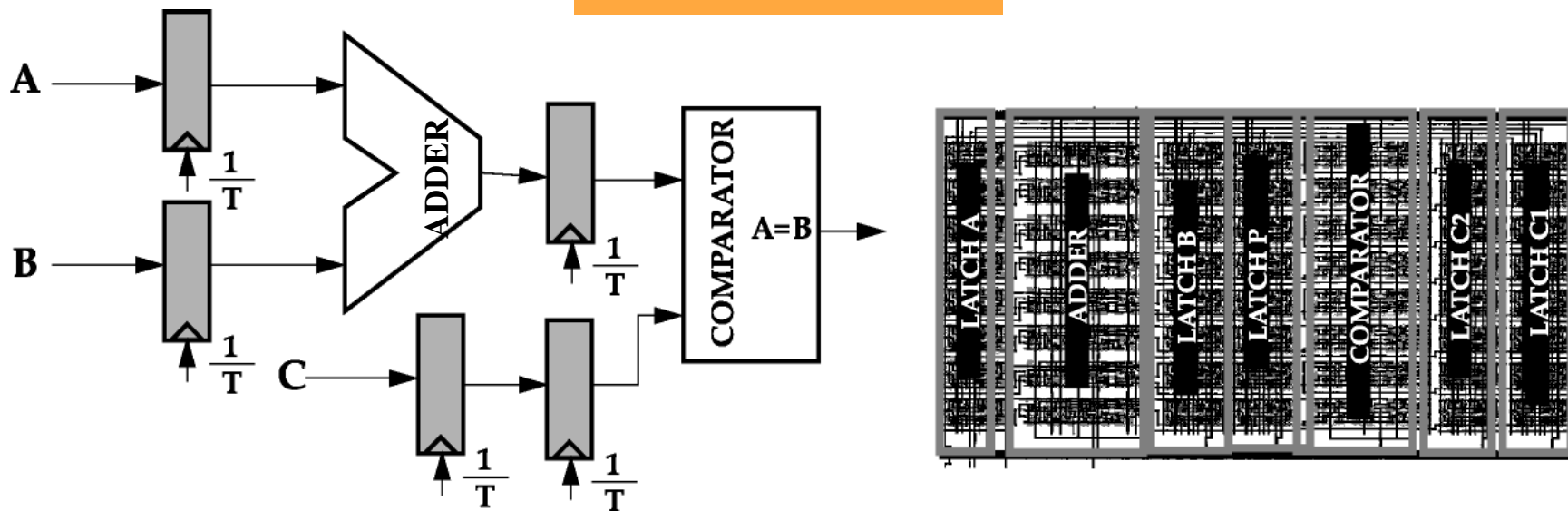
■ 关键路径  $T_{\text{adder}} + T_{\text{comparator}} = 25\text{ns} \Rightarrow f_{\text{ref}} = 40\text{MHz}$

■ 总电容  $C_{\text{ref}}$  ■ 电源电压  $V_{DD} = 5\text{V}$

■ 动态功耗  $P_{\text{ref}} = C_{\text{ref}} V_{DD}^2 f_{\text{ref}}$

# 流水线技术降低功耗应用——加法比较电路

## 两级流水电路



$$f_{\text{pipe}} = f_{\text{ref}} = 40\text{MHz}$$

$$C_{\text{pipe}} = 1.1C_{\text{ref}}$$

$$V_{\text{pipe}} = 0.59V_{DD}$$

$$P_{\text{pipe}} = C_{\text{pipe}} V_{\text{pipe}}^2 f_{\text{pipe}} = (1.1C_{\text{ref}})(0.59V_{DD})^2 f_{\text{ref}} = 0.37P_{\text{ref}}$$





# 目录

**01** 并行处理技术

---

**02** 流水线技术

---

**03** 流水线技术降低功耗

---

**04** 并行处理技术降低功耗

---

**05** 本章总结

---



# 并行处理技术降低功耗原理——传播延时与时钟周期

$$P_{\text{dynamic}} \downarrow = C_{\text{total}} V_{DD}^2 f \begin{cases} f \downarrow, \text{处理能力} \downarrow \\ V_{DD} \downarrow, T_{pd} \uparrow, f \downarrow, \text{处理能力} \downarrow \end{cases}$$

## 传播延时公式

充放电电容

电源电压

$$T_{pd} = \frac{C_{\text{charge}} V_{DD}}{k(V_{DD} - V_t)^2}$$

$k$  为跨导因子

$V_t$  为MOS管阈值电压

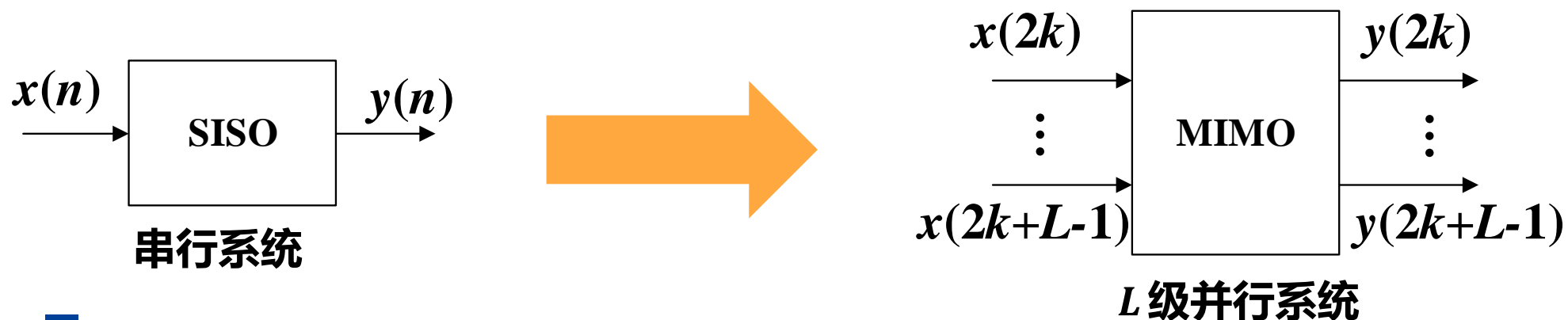
$$f = \frac{1}{\max(T_{pd})}$$

是否有方法可以不降低处理能力，而降低动态功耗？



并行处理技术

# 并行处理技术降低功耗原理——并行处理降低功耗



## 加速

- 时钟周期不变，采样速率提高，功耗增加

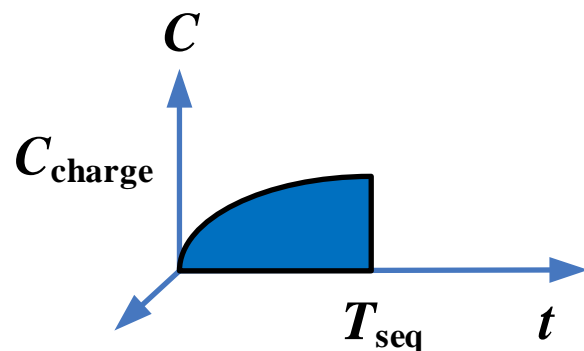
## 降低功耗

$$P \downarrow = C_{\text{total}} \uparrow V_{DD}^2 \downarrow f \downarrow \quad T_{pd} = \frac{C_{\text{charge}} V_{DD}}{k(V_{DD} - V_t)^2}$$

- 采样速率不变，时钟频率降低为  $1/L$
- 总负载电容增大  $L$  倍
- 电源电压降低，功耗降低

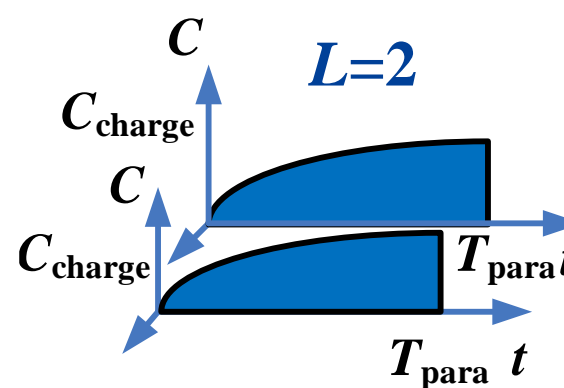
# 并行处理技术降低功耗方法——求解电压和功耗

## 原始电路



$$V_{\text{seq}} = V_{DD}$$

## 两级并行电路



$$V_{\text{para}} = \beta V_{DD}$$

$\beta$  为功耗降低因子

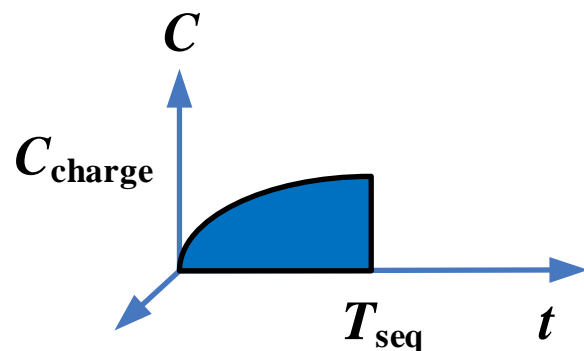
$$T_{\text{seq}} = \frac{C_{\text{charge}} V_{DD}}{k(V_{DD} - V_t)^2}$$

$$LT_{\text{seq}} = T_{\text{para}}$$

$$T_{\text{para}} = \frac{C_{\text{charge}} \beta V_{DD}}{k(\beta V_{DD} - V_t)^2}$$

# 并行处理技术降低功耗方法——降低功耗

## 原始电路

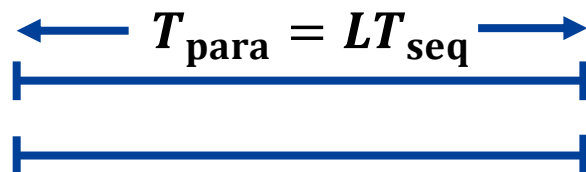
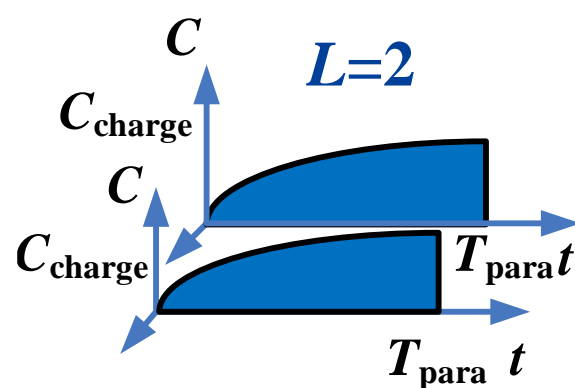


时钟周期



$$V_{\text{seq}} = V_{DD}$$

## 两级并行电路



$$V_{\text{para}} = \beta V_{DD}$$

得到含 $\beta$ 的二次方程

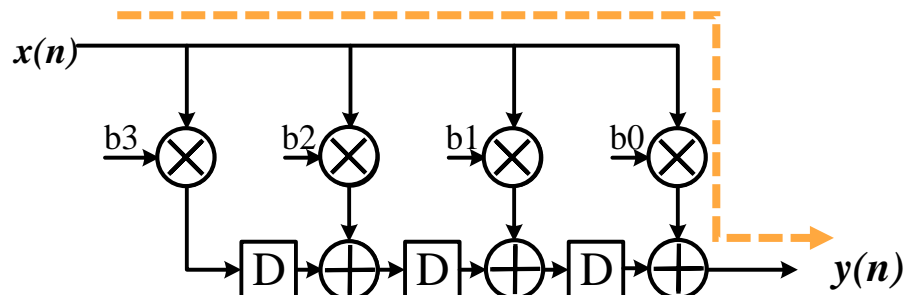
$$L(\beta V_{DD} - V_t)^2 = \beta(V_{DD} - V_t)^2 \rightarrow \beta$$

采用并行处理后功耗为

$$P_{\text{para}} = (LC_{\text{total}})(\beta V_{DD})^2 \frac{f}{L} = \beta^2 P_{\text{seq}}$$

# 并行处理技术降低功耗应用——FIR滤波器

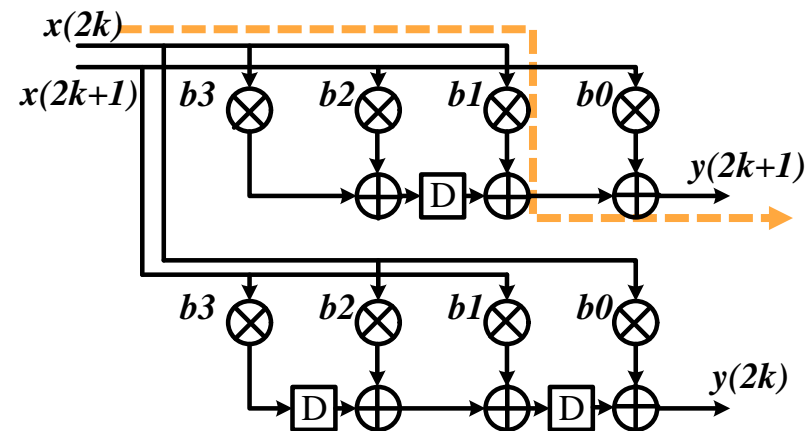
## 原始滤波器



$$V_0 = 3.3V \quad V_t = 0.45V \quad C_M = 8C_A$$

$$C_{\text{charge}} = C_M + C_A = 9C_A$$

## 两级并行滤波器



$$C_{\text{charge}}' = C_M + 2C_A = 10C_A$$

$$LC_{\text{charge}}(\beta V_{DD} - V_t)^2 = C_{\text{charge}}'\beta(V_{DD} - V_t)^2$$

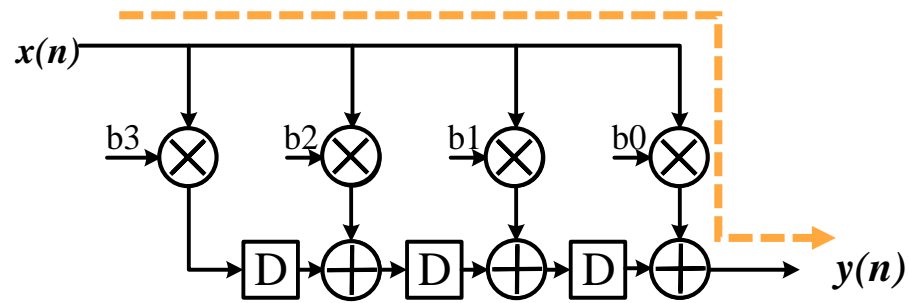
$$9(3.3\beta - 0.45)^2 = 10\beta(3.3 - 0.45)^2$$

$$\beta = 0.6589 \text{ 或 } 0.0282 \text{ (舍弃)}$$

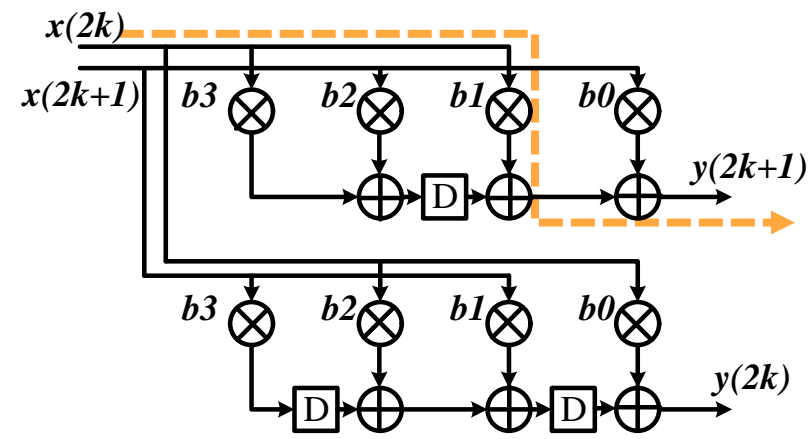
$$\text{新的功耗为: } P_{\text{para}} = \beta^2 P_{\text{seq}} = 0.434 P_{\text{seq}}$$

# 并行处理技术降低功耗应用——FIR滤波器

原始滤波器



两级并行滤波器

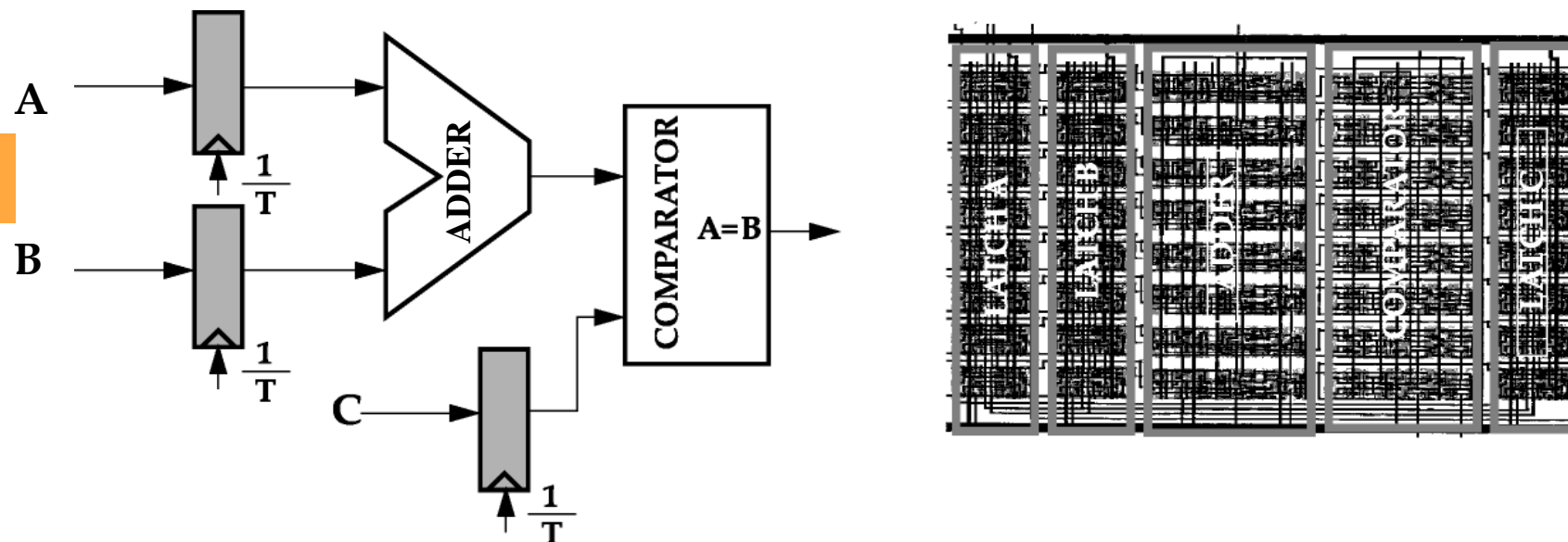


乘法器计算时间  $8u.t.$ ，加法器计算时间  $1u.t.$

系统	原始滤波器	两级 并行滤波器 (不降功耗)	两级 并行滤波器 (降功耗)
时钟周期 ( $u.t.$ )	$9u.t.$	$10u.t.$	$18u.t.$
采样周期 ( $u.t.$ )	$9u.t.$	$5u.t.$	$9u.t.$
功耗	$P_{seq}$	$1.8P_{seq}$	$0.434P_{seq}$

# 并行处理技术降低功耗应用——加法比较电路（1）

原始电路



■ 关键路径  $T_{\text{adder}} + T_{\text{comparator}} = 25\text{ns} \rightarrow f_{\text{ref}} = 40\text{MHz}$

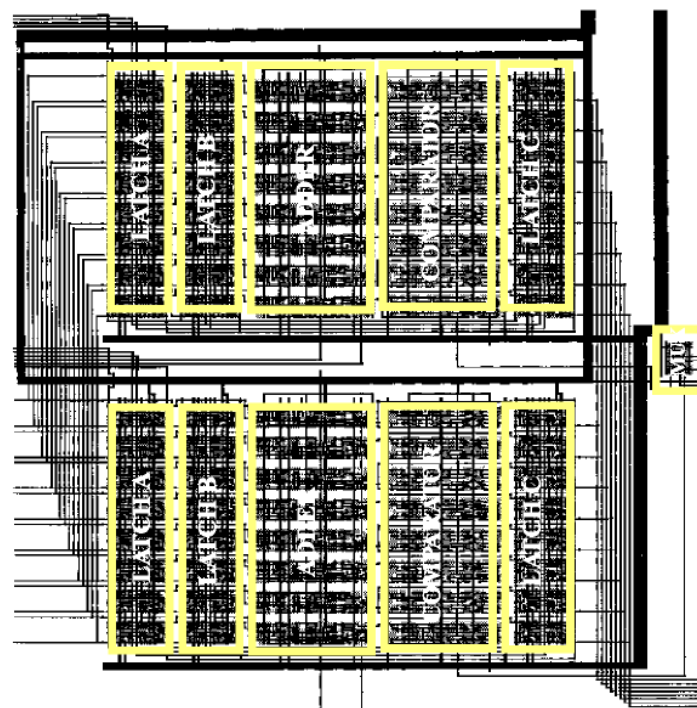
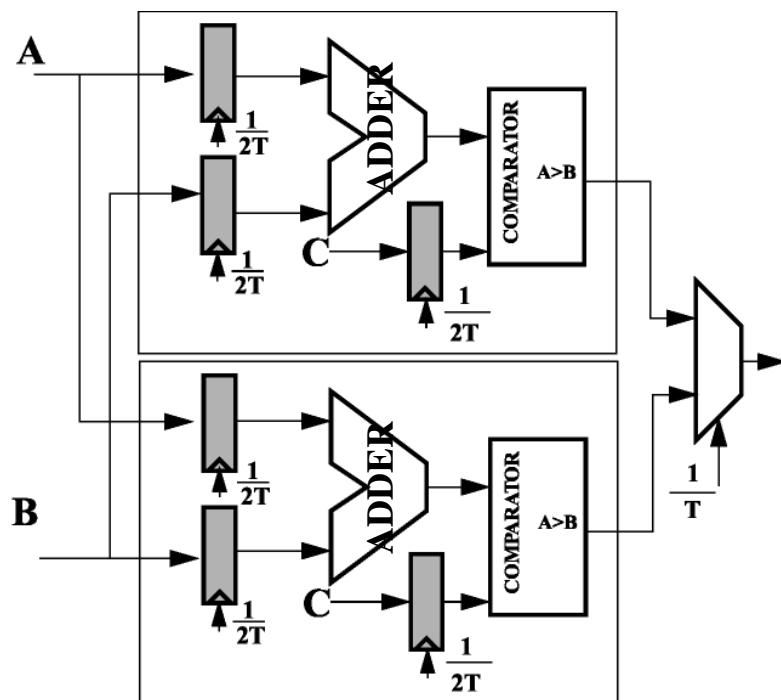
■ 总电容  $C_{\text{ref}}$  ■ 电源电压  $V_{DD} = 5\text{V}$

■ 动态功耗  $P_{\text{ref}} = C_{\text{ref}} V_{DD}^2 f_{\text{ref}}$



## 并行处理技术降低功耗应用——加法比较电路（2）

两级并  
行电路



采样速率不变，时钟频率减半  $f_{\text{para}} = f_{\text{ref}}/2 = 20\text{MHz}$

$$V_{\text{para}} = 0.59V_{DD}$$

$$C_{\text{para}} = 2.15C_{\text{ref}}$$

$$P_{\text{para}} = (2.15C_{\text{ref}})(0.59V_{DD})^2(f_{\text{ref}}/2) = 0.36P_{\text{ref}}$$

# 谢谢!

