



Pipelining and Parallel Processing

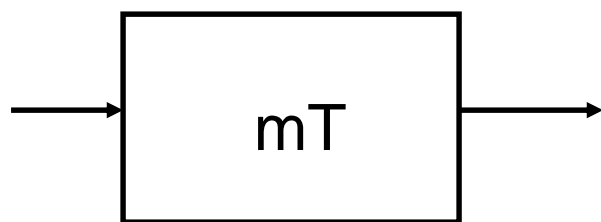
Shao-Yi Chien



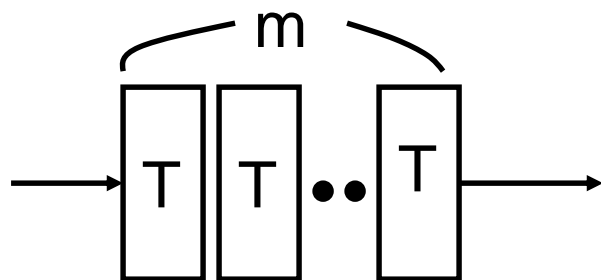
Introduction

- Pipelining and parallel are the most important design techniques in VLSI DSP systems
- Make use of the inherent parallel property of DSP algorithms
 - Pipelining: different function units working in parallel
 - Parallel: duplicated function units working in parallel

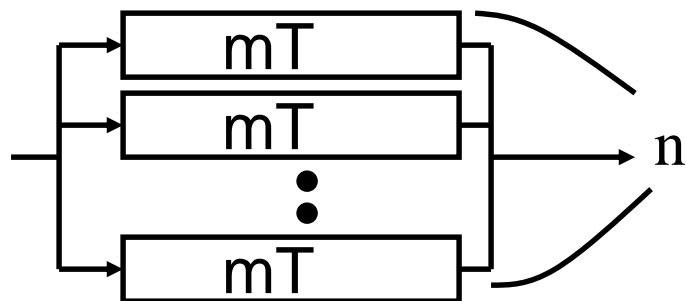
An Example: Car Production Line



Cost: 1
Throughput: $1/mT$
Latency: mT



Cost: >1
Throughput: $1/T$
Latency: $>mT$



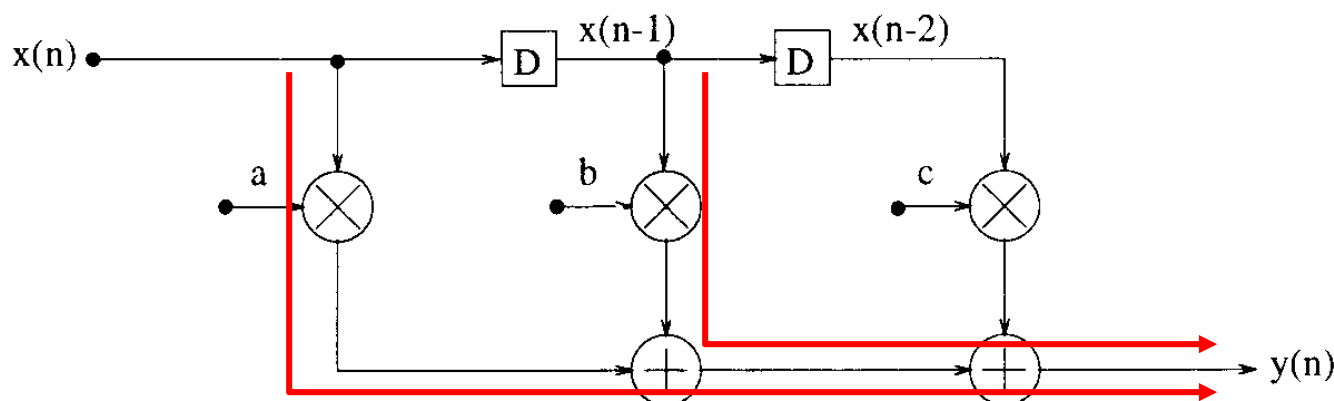
Cost: $>n$
Throughput: $(1/mT)*n$
Latency: $>mT$

Throughput: how many cars produced in one hour
Latency: how long will it take to produce one car

Pipelining of Digital Filters (1/3)

■ FIR filter

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

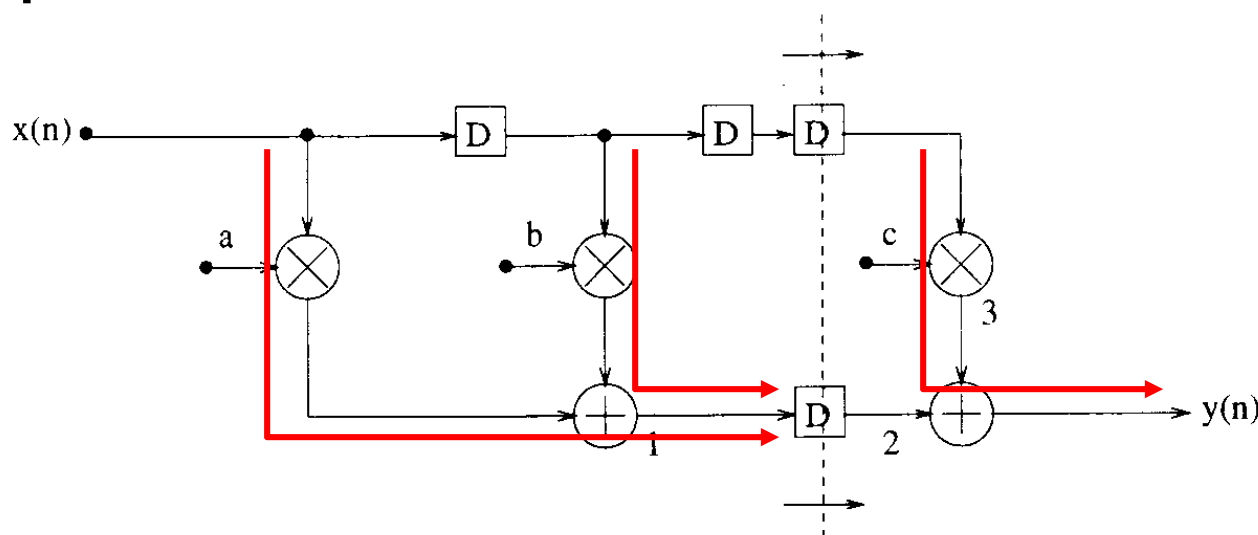


□ Critical path: $T_M + 2T_A$

$$T_{sample} \geq T_M + 2T_A \quad f_{sample} \leq \frac{1}{T_M + 2T_A}$$

Pipelining of Digital Filters (2/3)

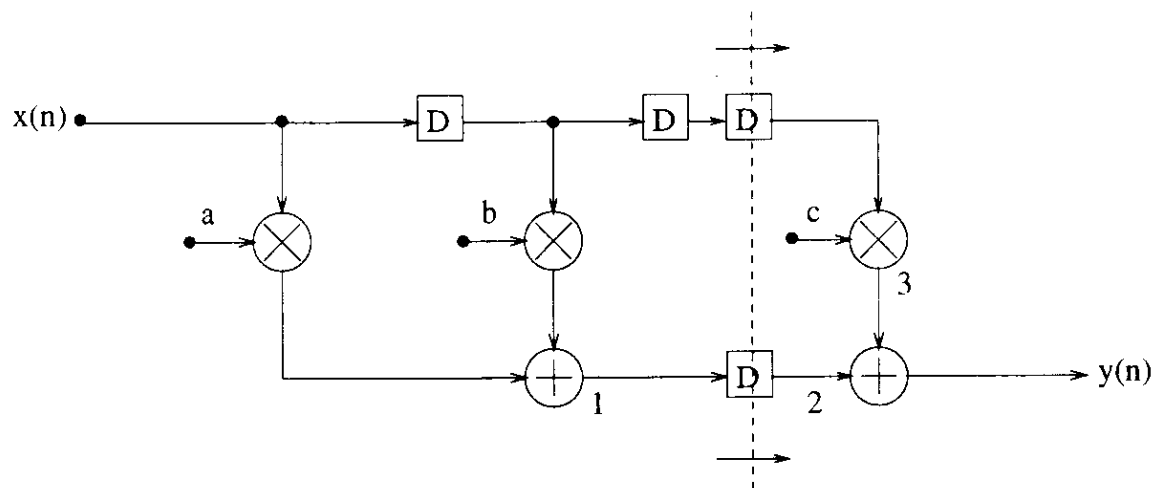
■ Pipelined FIR filter



$$T_{sample} \geq T_M + T_A \quad f_{sample} \leq \frac{1}{T_M + T_A}$$

Pipelining of Digital Filters (3/3)

■ Schedule



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

Pipeline

- Can reduce the critical path to increase the working frequency and sample rate

- $T_M + 2T_A \rightarrow T_M + T_A$

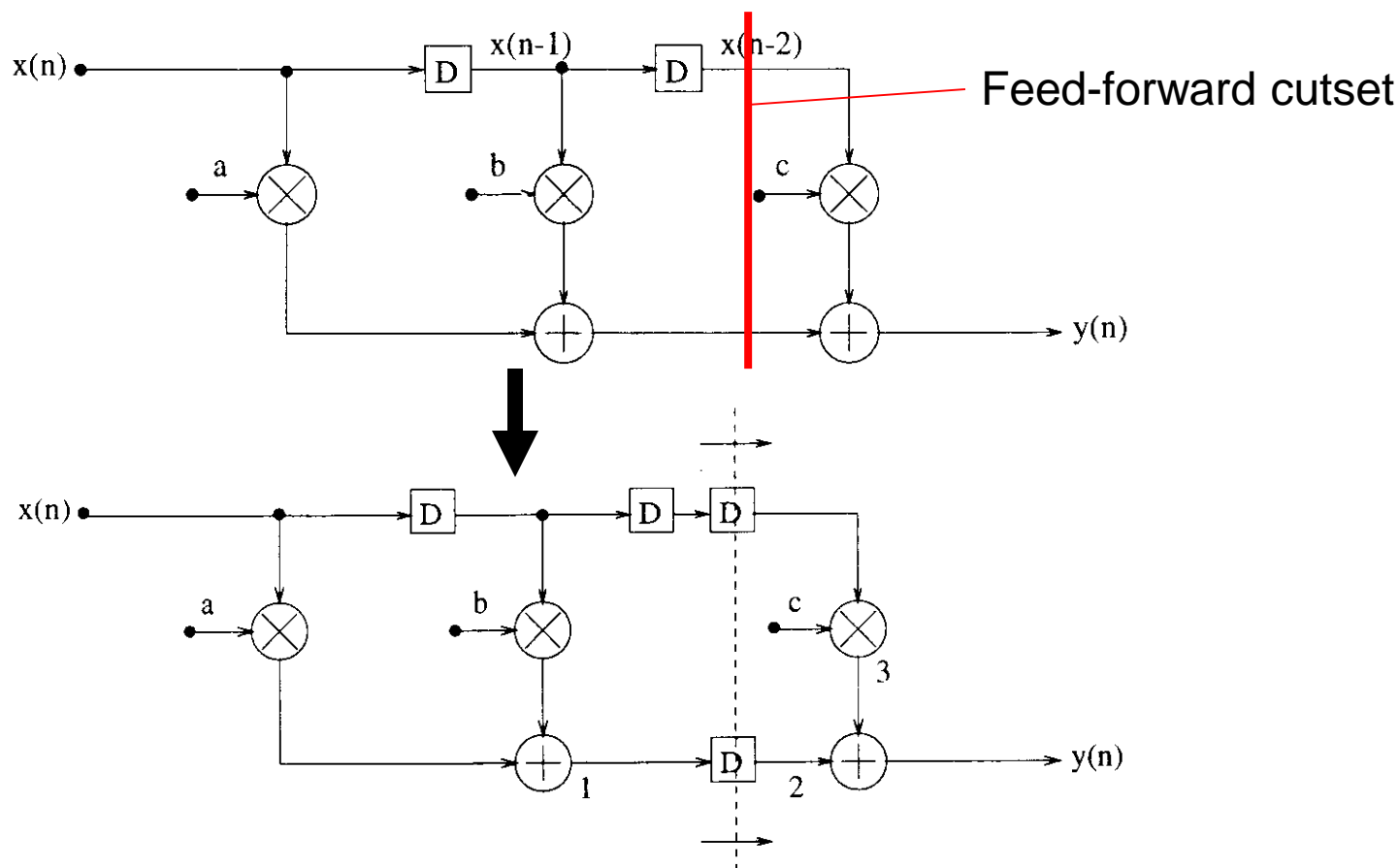
Drawbacks of Pipelining

- Increasing latency (in cycle)
 - For M-level pipelined system, the number of delay elements in any path from input to output is (M-1) greater than the origin one
- Increase the number of latches (registers)

How to Do Pipelining?

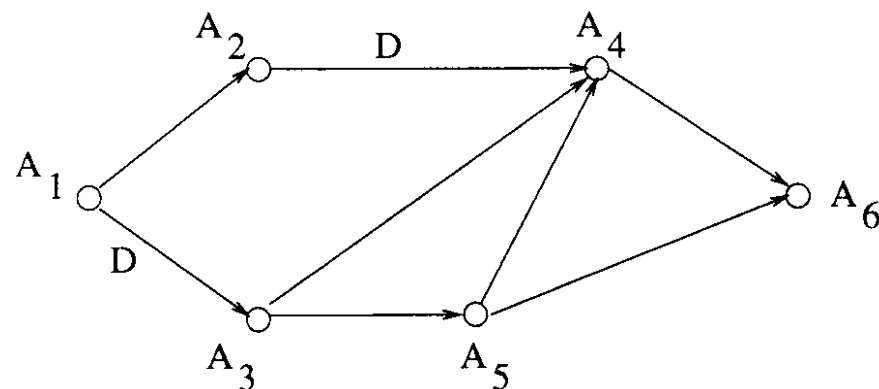
- Put pipelining latches across any feed-forward cutset of the graph
- Cutset
 - A cutset is a set of edges of a graph such that if these edges are removed from the graph, the graph becomes disjoint
- Feed-forward cutset
 - The data move in the forward direction on all the edges of the cutset

How to Do Pipelining?

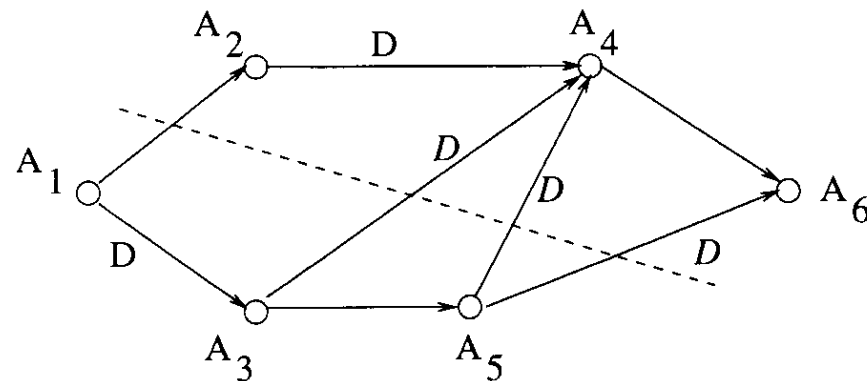


Example

- In the SFG in Fig. (a), all the computation time for each node is 1 u.t.
- (a) Calculate the critical path
- (b) The critical path is reduced to 2 u.t. by inserting 3 extra delay elements. Is it a valid pipelining?

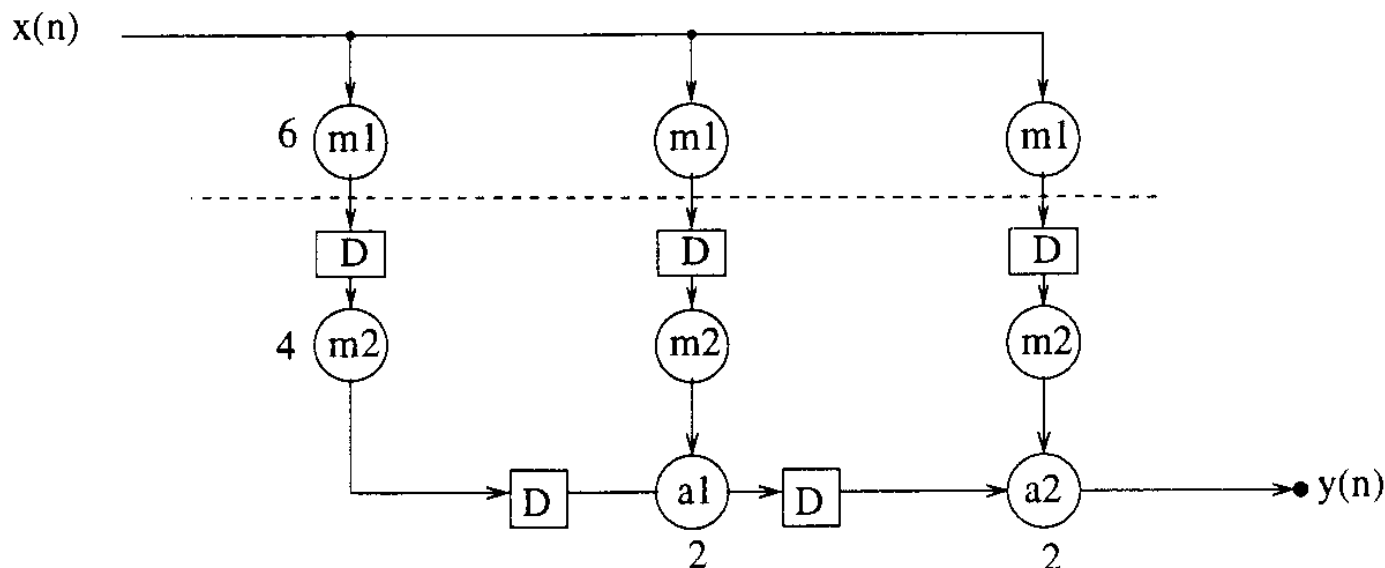


(a)



(b)

Fine-Grain Pipelining



■ Critical path ($T_M=10$, $T_A=2$)

- $T_M+2T_A=14$
- $T_M+T_A=12$
- $T_{M1}=6$ or $T_{M2}+T_A=6$

Notes for Pipelining (1/2)

- Pipelining is a very simple design technique which can maintain the input output data configuration and sampling frequency
- $T_{\text{clk}} = T_{\text{sample}}$
- Supported in many EDA tools
- Still has some limitations
 - Pipeline bubbles
 - Has some problems for recursive system
 - Introduces large hardware cost for 2-D or 3-D data
 - Communication bound



Notes for Pipelining (2/2)

■ Effective pipelining

- Put pipelining registers on the critical path

- Balance pipelining

- $10 \rightarrow (2+8)$: critical path=8

- $10 \rightarrow (5+5)$: critical path=5

Parallel of Digital Filters (1/5)

- Single-input single-output (SISO) system

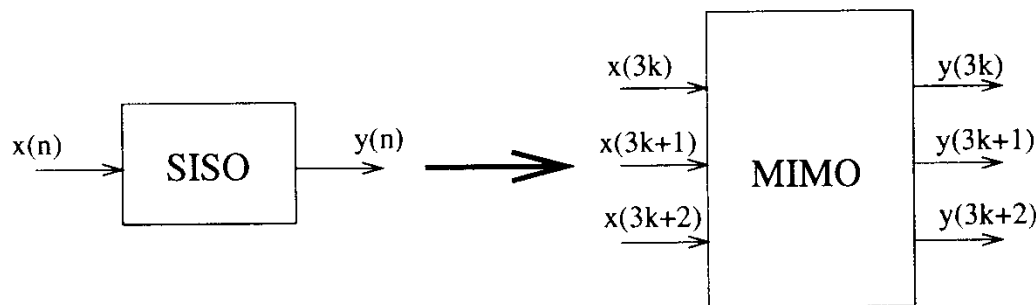
$$y(n) = ax(n) + bx(n - 1) + cx(n - 2).$$

- Multiple-input multiple-output (MIMO) system

$$y(3k) = ax(3k) + bx(3k - 1) + cx(3k - 2)$$

$$y(3k + 1) = ax(3k + 1) + bx(3k) + cx(3k - 1) \quad \text{3-Parallel System!}$$

$$y(3k + 2) = ax(3k + 2) + bx(3k + 1) + cx(3k).$$



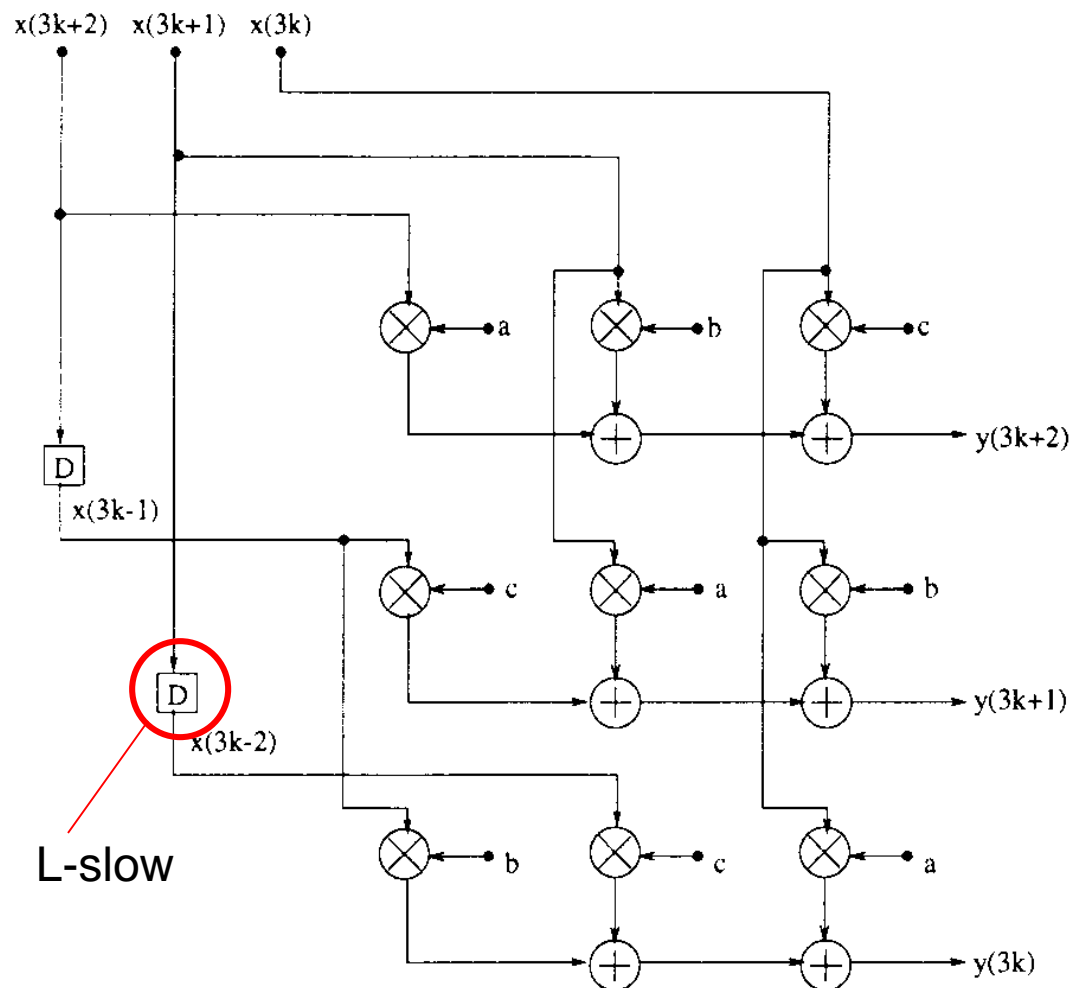
Sequential System

3-Parallel System

Parallel of Digital Filters (2/5)

- Parallel processing, block processing
- Block size (L): the number of data to be processed at the same time
- Block delay (L -slow)
 - A latch is equivalent to L clock cycles at the sample rate

Parallel of Digital Filters (3/5)



$$T_{clk} \geq T_M + 2T_A$$

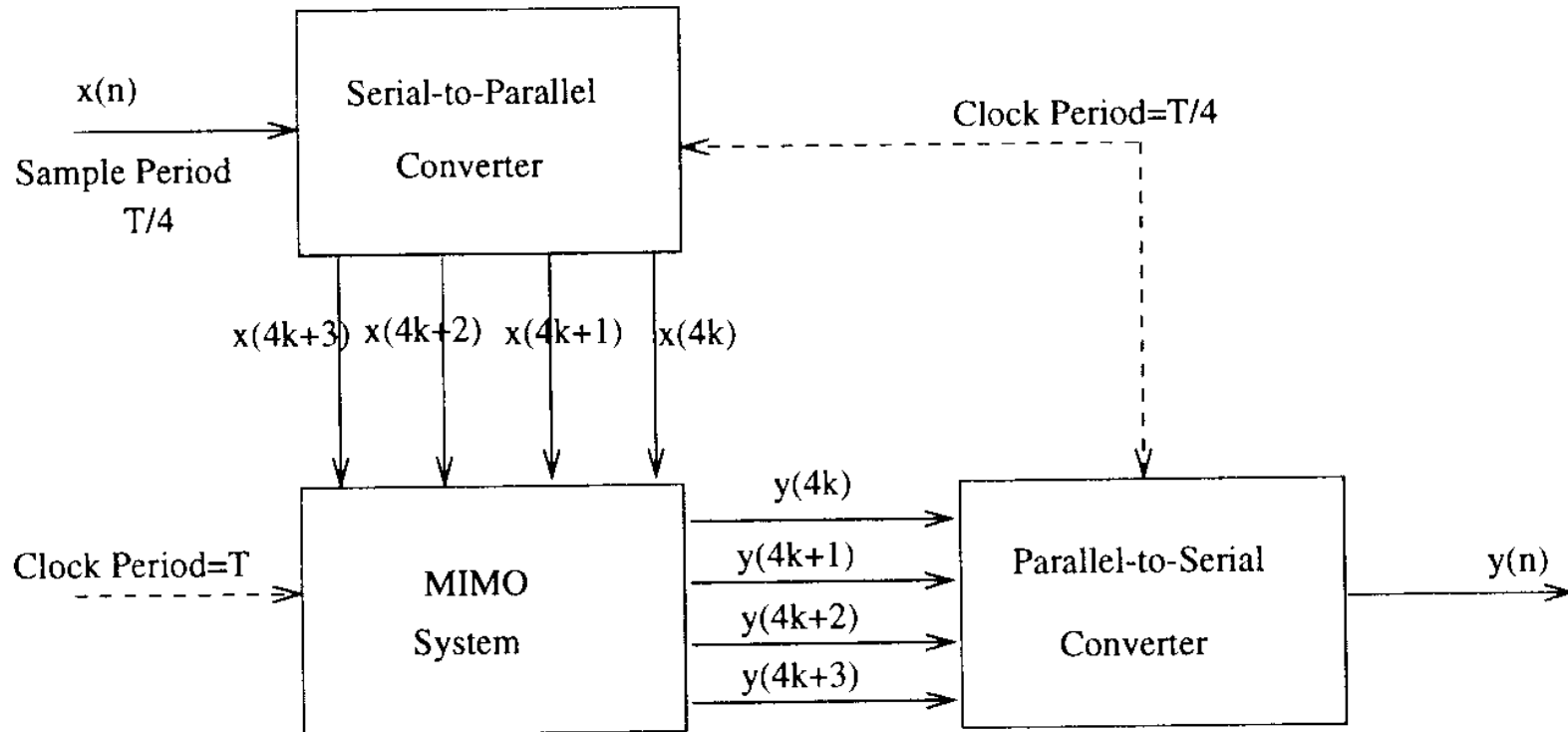
The critical path is the same

$$T_{iter} = T_{sample} = \frac{1}{L} T_{clk} \geq \frac{1}{3} (T_M + 2T_A)$$

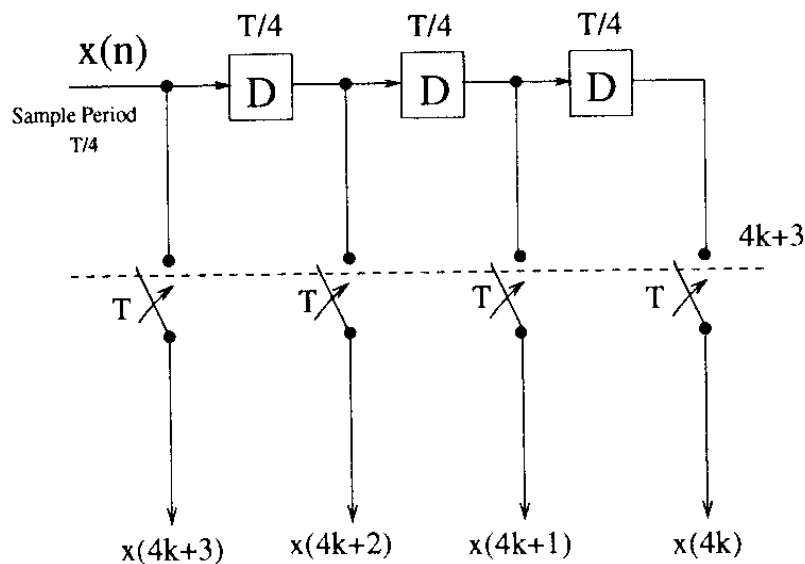
T_{clk} is not equal to T_{sample}

Parallel of Digital Filters (4/5)

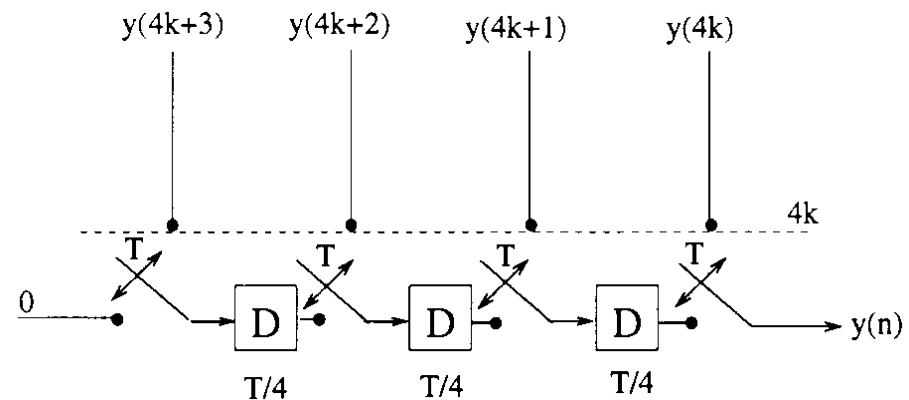
■ Whole system



Parallel of Digital Filters (5/5)



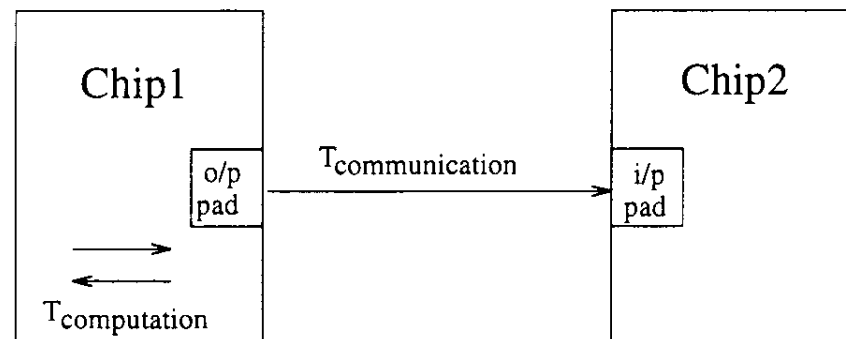
Serial-to-parallel converter



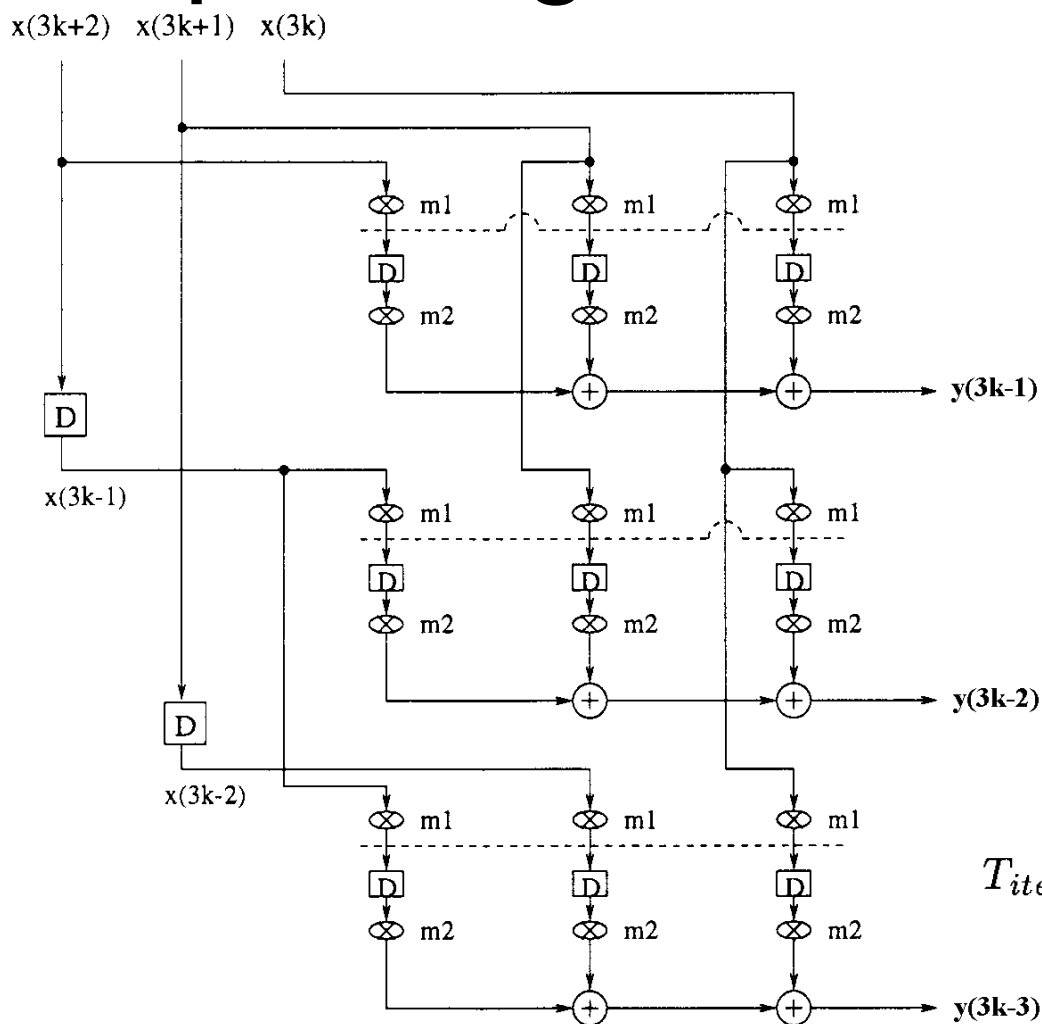
Parallel-to-serial converter

Parallel Processing v.s. Pipelining

- Parallel processing is superior than pipelining processing for the I/O bottleneck (communication bounded)
 - Pipelining only can increase the clock rate
 - System clock rate = sample rate for pipelining system
 - Use parallel processing can further lower the required working frequency



Pipelining-Parallel Architecture



$$T_{iter} = T_{sample} = \frac{1}{LM} T_{clk} = \frac{1}{6} (T_M + 2T_A)$$



Notes for Parallel Processing

- The input/output data access scheme should be carefully designed, it will cost a lot sometimes
- $T_{\text{clk}} > T_{\text{sample}}$, $f_{\text{clk}} < f_{\text{sample}}$
- Large hardware cost
- Combined with pipelining processing

Low Power Issues

- Pipelining and parallel processing are also beneficial for low power design

- Propagation delay
$$T_{pd} = \frac{C_{charge} V_0}{k(V_0 - V_t)^2}$$

- Power consumption

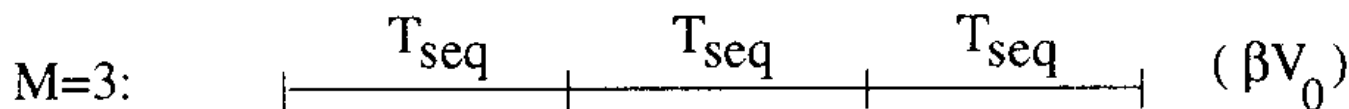
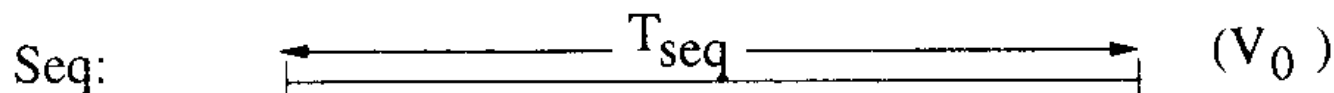
$$P = C_{total} V_0^2 f \quad f = \frac{1}{T_{seq}}$$

- Assume the sampling frequency is the same

Pipelining for Low Power (1/2)

■ For M-level pipelining

- Critical path is reduced to $1/M$
- The capacitance is also reduced to C_{charge}/M
- The supply voltage can be reduced to βV_0 , and the propagation delay remains unchanged



Pipelining for Low Power (2/2)

- Power consumption:

$$P_{pip} = C_{total} \beta^2 V_0^2 f = \beta^2 P_{seq}$$

- How about the parameter β ?

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

||

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

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

Solve this equation to get β

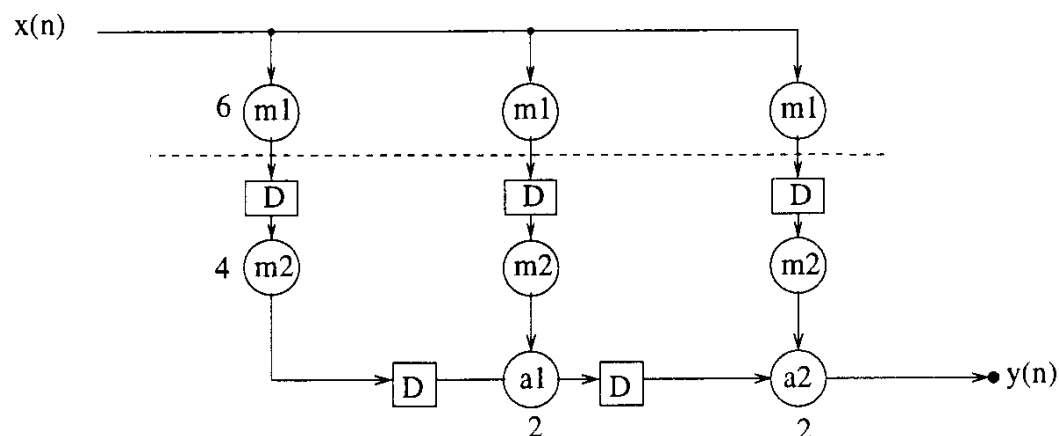
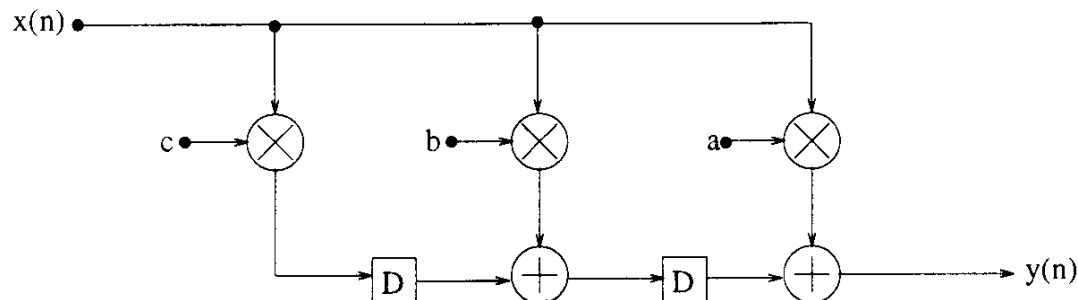
Example

Parameters

- ☐ $T_M = 10$ u.t.
- ☐ $T_A = 2$ u.t.
- ☐ $T_{m1} = 6$ u.t.
- ☐ $T_{m2} = 4$ u.t.
- ☐ $C_M = 5C_A$
- ☐ $V_t = 0.6V$
- ☐ Normal $V_{CC} = 5V$

■ (a) New supply voltage?

■ (b) Power saving percentage?



Answer

(a)

Origin system: $C_{charge} = C_M + C_A = 6C_A$

Pipelined system: $C_{charge} = C_{m1} = C_{m2} + C_A = 3C_A$

$$\longrightarrow 50\beta^2 - 31.36\beta + 0.72 = 0$$

$$\beta = 0.6033, \text{ or } \beta = 0.0239$$

Invalid value, less than threshold voltage

$$V_{pip} = \beta V_0 = 3.0165 \text{ V.}$$

(b)

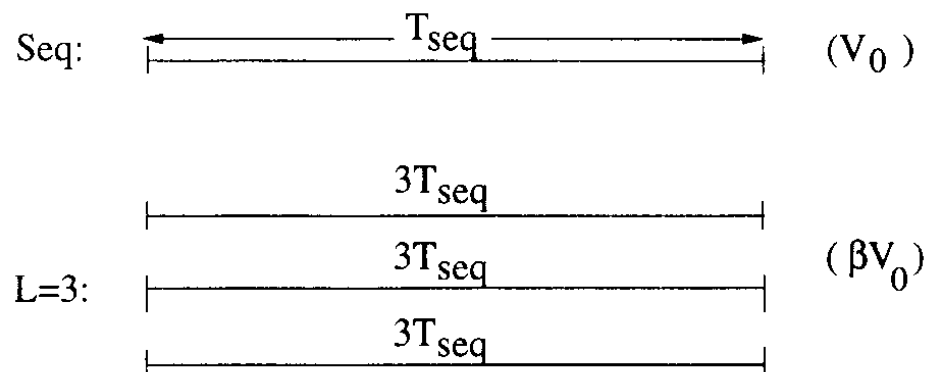
$$Ratio = \beta^2 = 36.4\%.$$



Parallel Processing for Low Power (1/2)

■ For L-parallel system

- Clock period: $T_{seq} \rightarrow LT_{seq}$
- C_{charge} remains unchanged
- $C_{total} \rightarrow LC_{total}$
- Have more time to charge the capacitance, the supply voltage can be lower βV_0



Parallel Processing for Low Power (2/2)

■ Power consumption

$$\begin{aligned}P_{par} &= (LC_{total})(\beta V_0)^2 \frac{f}{L} \\&= \beta^2 C_{total} V_0^2 f \\&= \beta^2 P_{seq}\end{aligned}$$

■ To derive the parameter β

$$\begin{aligned}T_{seq} &= \frac{C_{charge} V_0}{k(V_0 - V_t)^2} \\&\downarrow \\LT_{seq} &= \frac{C_{charge} \beta V_0}{k(\beta V_0 - V_t)^2} \longrightarrow L(\beta V_0 - V_t)^2 = \beta(V_0 - V_t)^2\end{aligned}$$

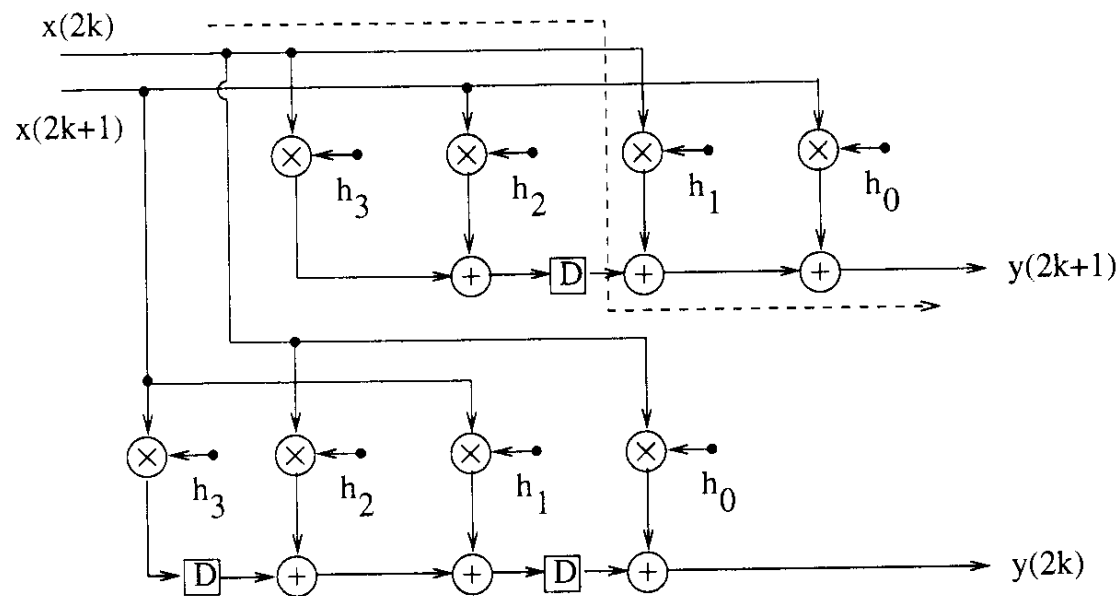
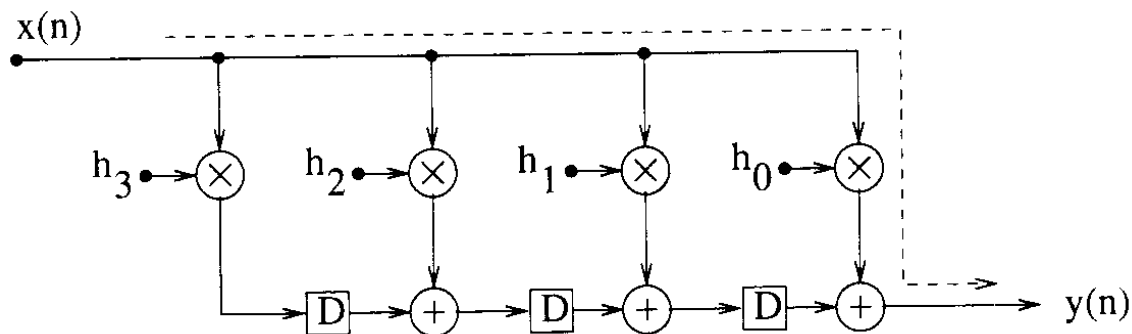
Example

Parameters

- ☐ $T_M = 8$ u.t.
- ☐ $T_A = 1$ u.t.
- ☐ $T_{\text{sample}} = 9$ u.t.
- ☐ $C_M = 8C_A$
- ☐ $V_t = 0.45V$
- ☐ Normal $V_{cc} = 3.3V$

■ (a) New supply voltage?

■ (b) Power saving percentage?



Answer

(a)

Origin: $C_{charge} = C_M + C_A = 9C_A$

2-parallel system: $C_{charge} = C_M + 2C_A = 10C_A$

$$T_{seq} = \frac{9C_A V_0}{k(V_0 - V_t)^2},$$

$$5\beta(V_0 - V_t)^2 = 9(\beta V_0 - V_t)^2$$

$$T_{par} = \frac{10C_A \beta V_0}{k(\beta V_0 - V_t)^2}.$$

$$\beta = 0.6589, \text{ or } \beta = 0.0282.$$

Invalid value, less than threshold voltage

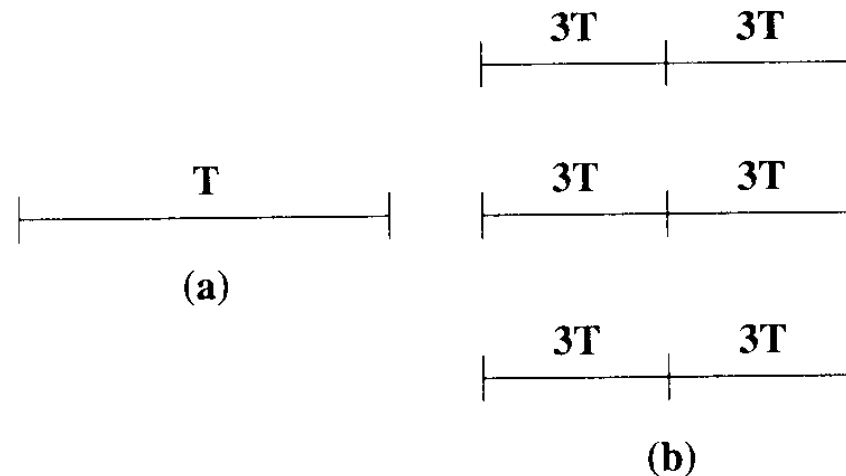
$$T_{par} = 2T_{sample} = 2T_{seq}$$

(b)

$$Ratio = \beta^2 = 43.41\%.$$



Pipelining-Parallel for Low Power



$$LT_{pd} = \frac{(C_{charge}/M)\beta V_o}{k(\beta V_o - V_t)^2} = \frac{LC_{charge}V_o}{k(V_o - V_t)^2}.$$

$$ML(\beta V_o - V_t)^2 = \beta(V_o - V_t)^2$$

Conclusion

■ Pipelining

- ☐ Reduce the critical path
- ☐ Increase the clock speed
- ☐ Reduce power consumption at the same speed

■ Parallel

- ☐ Increase effective sampling rate
- ☐ Reduce power consumption at the same speed