Chapter 3: Pipelining and Parallel Processing

Keshab K. Parhi

Outline

- Introduction
- Pipelining of FIR Digital Filters
- Parallel Processing
- Pipelining and Parallel Processing for Low Power
  - Pipelining for Lower Power
  - Parallel Processing for Lower Power
  - Combining Pipelining and Parallel Processing for Lower Power

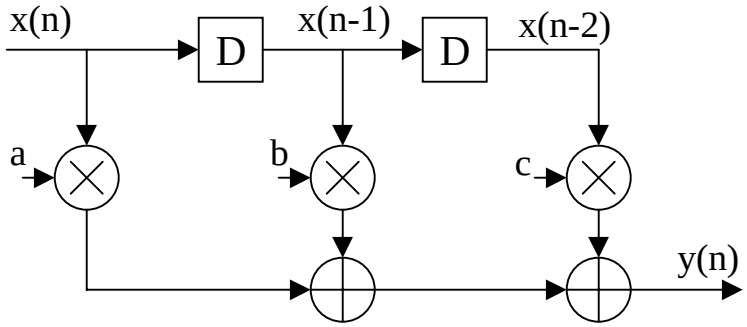
Introduction

- Pipelining
  - Comes from the idea of a water pipe: continue sending water without waiting the water in the pipe to be out

  - leads to a reduction in the critical path
  - Either increases the clock speed (or sampling speed) or reduces the power consumption at same speed in a DSP system
- Parallel Processing
  - Multiple outputs are computed in parallel in a clock period
  - The effective sampling speed is increased by the level of parallelism
  - Can also be used to reduce the power consumption

Introduction (cont'd)

- **Example 1**: Consider a 3-tap FIR filter: y(n)=ax(n)+bx(n-1)+cx(n-2)



$$T_M: \quad multiplica\,tion - time$$

$$T_A: \quad Addition - time$$

  - The critical path (or the minimum time required for processing a new sample) is limited by 1 multiply and 2 add times. Thus, the "sample period" (or the "sample frequency") is given by:

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

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

Introduction (cont'd)

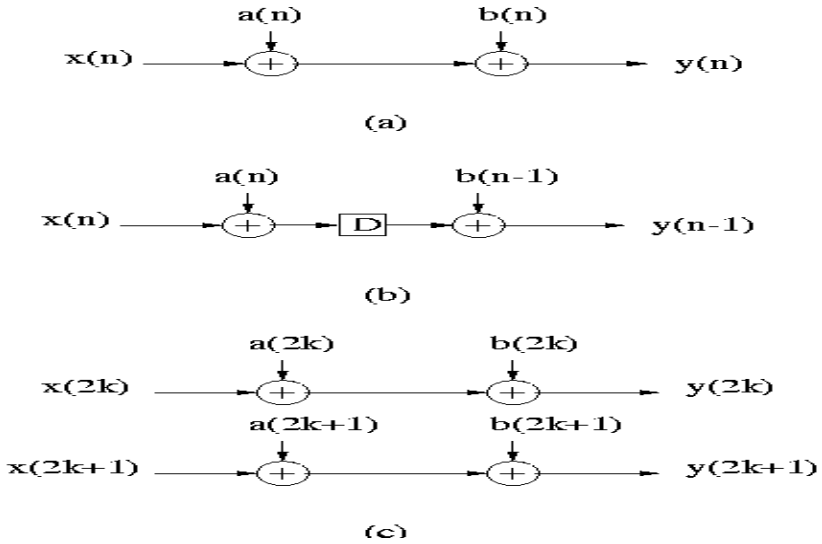
  - If some real-time application requires a faster input rate (sample rate), then this ***direct-form structure*** cannot be used! In this case, the critical path can be reduced by either ***pipelining*** or ***parallel processing***.

- Pipelining: reduce the effective critical path by introducing pipelining latches along the critical data path
- Parallel Processing: increases the sampling rate by replicating hardware so that several inputs can be processed in parallel and several outputs can be produced at the same time
- Examples of Pipelining and Parallel Processing
  - See the figures on the next page

Introduction (cont'd)

Example 2

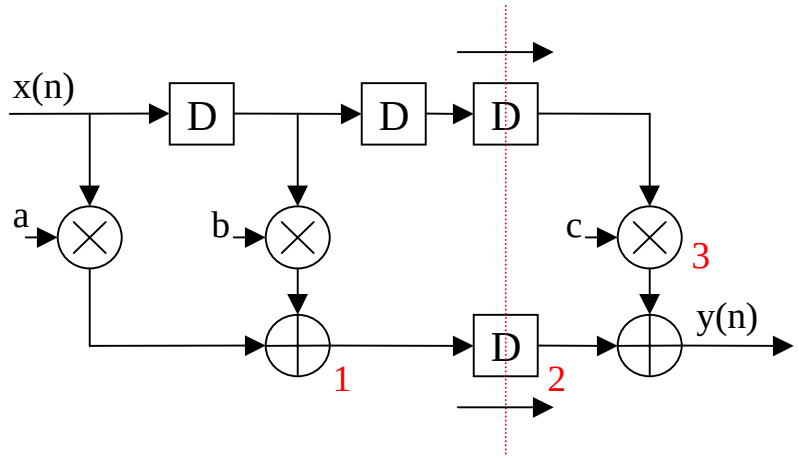
Figure (a): A data path
Figure (b): The 2-level pipelined structure of (a)
Figure (c): The 2-level parallel processing structure of (a)



Pipelining of FIR Digital Filters

- The pipelined implementation: By introducing 2 additional latches in Example 1, the critical path is reduced from T_M+2T_A to T_M+T_A.The schedule of events for this pipelined system is shown in the following table. You can see that, at any time, 2 consecutive outputs are computed in an interleaved manner.

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)



Pipelining of FIR Digital Filters (cont'd)

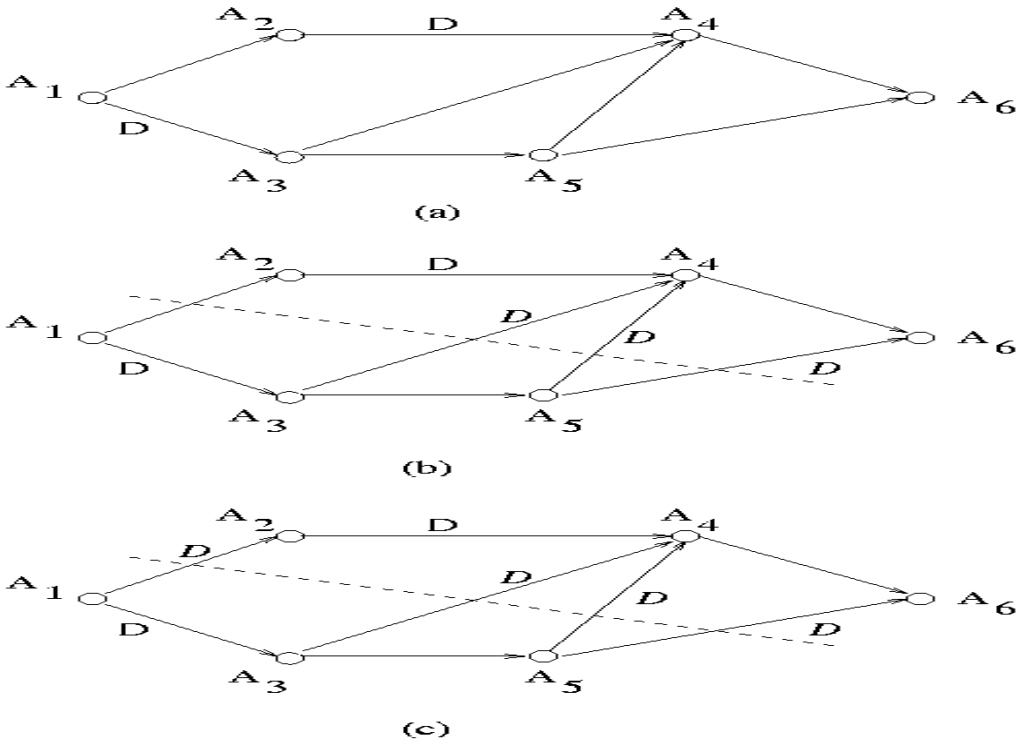
- In a pipelined system:
  - In an M-level pipelined system, the number of delay elements in any path from input to output is (M-1) greater than that in the same path in the original sequential circuit
  - Pipelining reduces the critical path, but leads to a penalty in terms of an increased latency
    - *Latency: the difference in the availability of the first output data in the pipelined system and the sequential system*
  - Two main drawbacks: increase in the number of latches and in system latency
- Important Notes:
  - The speed of a DSP architecture (or the clock period) is limited by the longest path between any 2 latches, or between an input and a latch, or between a latch and an output, or between the input and the output

Pipelining of FIR Digital Filters (cont'd)

  - This longest path or the "critical path" can be reduced by suitably placing the pipelining latches in the DSP architecture
  - The pipelining latches can only be placed across any ***feed-forward cutset*** of the graph
- Two important definitions
  - **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:** a cutset is called a feed-forward cutset if the data move in the forward direction on all the edge of the cutset
  - **Example 3:** (P.66, Example 3.2.1, see the figures on the next page)
    - (1) The critical path is $A_3 \rightarrow A_5 \rightarrow A_4 \rightarrow A_6$, its computation time: 4 u.t.
    - (2) Figure (b) is not a valid pipelining because it's not a feed-forward cutset
    - (3) Figure (c) shows 2-stage pipelining, a valid feed-forward cutset. Its critical path is 2 u.t.

Pipelining of FIR Digital Filters (cont'd)

Signal-flow graph representation of Cutset



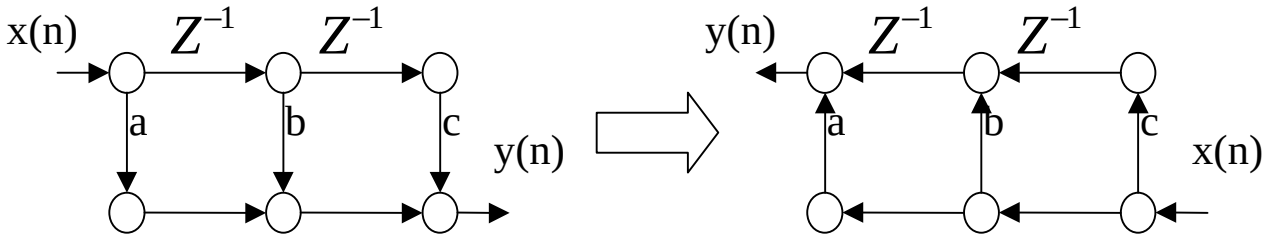
- Original SFG
- A cutset
- A feed-forward cutset

Assume:

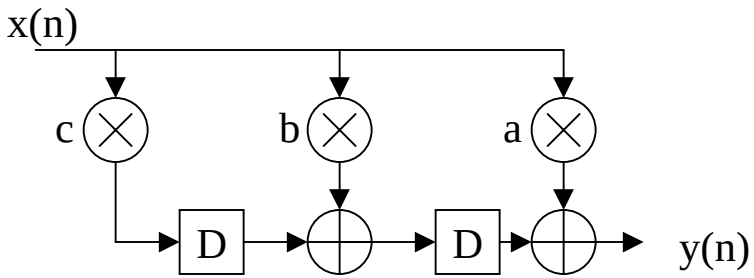
The computation time for each node is assumed to be 1 unit of time

Pipelining of FIR Digital Filters (cont'd)

- Transposed SFG and Data-broadcast structure of FIR filters
  - Transposed SFG of FIR filters



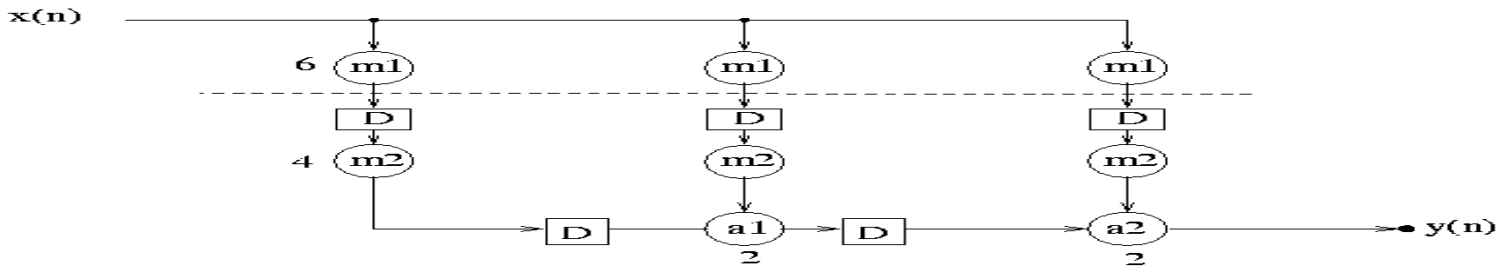
  - Data broadcast structure of FIR filters



Pipelining of FIR Digital Filters (cont'd)

- Fine-Grain Pipelining
  - Let T_M=10 units and T_A=2 units. If the multiplier is broken into 2 smaller units with processing times of 6 units and 4 units, respectively (by placing the latches on the horizontal cutset across the multiplier), then the desired clock period can be achieved as $(T_M+T_A)/2$
  - A fine-grain pipelined version of the 3-tap data-broadcast FIR filter is shown below.

Figure: fine-grain pipelining of FIR filter



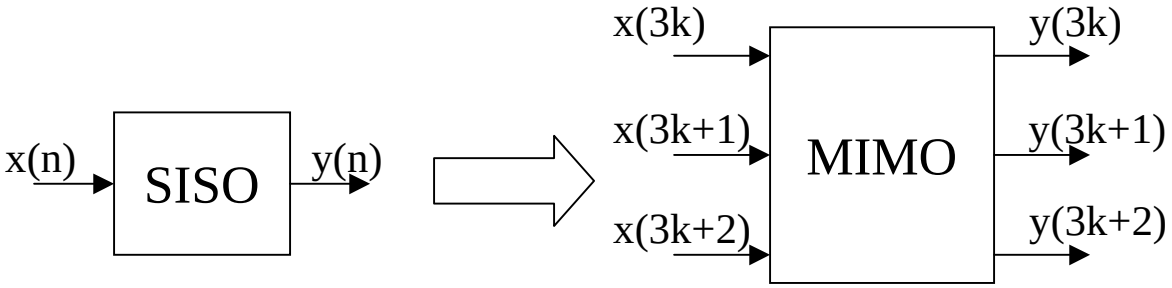
Parallel Processing

- Parallel processing and pipelining techniques are duals each other: if a computation can be pipelined, it can also be processed in parallel. Both of them exploit concurrency available in the computation in different ways.
- How to design a Parallel FIR system?
  - Consider a single-input single-output (SISO) FIR filter:
    - $y(n)=a*x(n)+b*x(n-1)+c*x(n-2)$
  - Convert the SISO system into an MIMO (multiple-input multiple-output) system in order to obtain a parallel processing structure
    - For example, to get a parallel system with 3 inputs per clock cycle (i.e., level of parallel processing L=3)

$$y(3k)=a*x(3k)+b*x(3k-1)+c*x(3k-2)$$
$$y(3k+1)=a*x(3k+1)+b*x(3k)+c*x(3k-1)$$
$$y(3k+2)=a*x(3k+2)+b*x(3k+1)+c*x(3k)$$

Parallel Processing (cont'd)

- Parallel processing system is also called ***block processing***, and the number of inputs processed in a clock cycle is referred to as the ***block size***



Sequential System 3-Parallel System

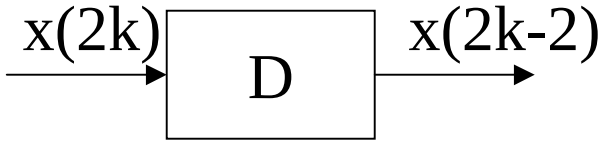
- In this parallel processing system, at the k-th clock cycle, 3 inputs x(3k), x(3k+1) and x(3K+2) are processed and 3 samples y(3k), y(3k+1) and y(3k+2) are generated at the output

- Note 1: In the MIMO structure, placing a latch at any line produces an effective delay of L clock cycles at the sample rate (L: the block size). So, each delay element is referred to as a ***block delay*** (also referred to as L-slow)

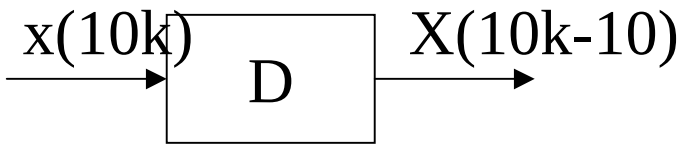
Parallel Processing (cont'd)

– For example:

When block size is 2, 1 delay element = 2 sampling delays

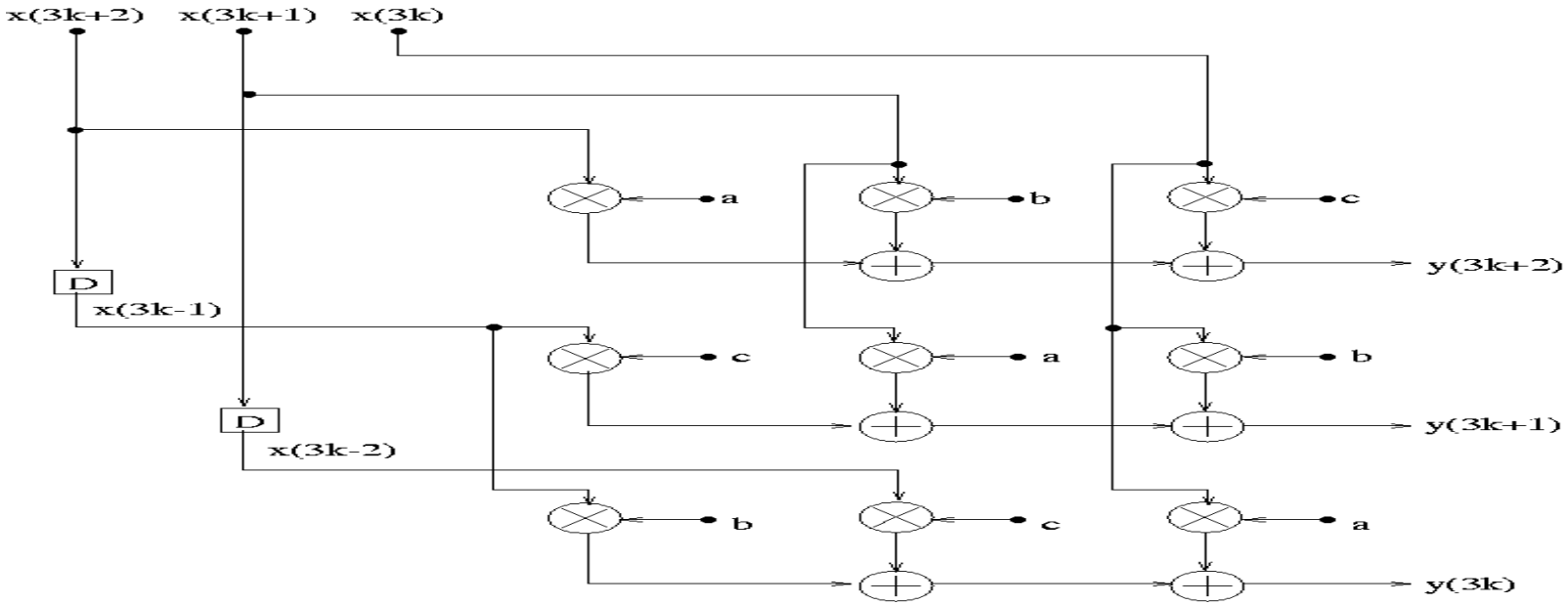


When block size is 10, 1 delay element = 10 sampling delays



Parallel Processing (cont'd)

Figure: Parallel processing architecture for a 3-tap FIR filter with block size 3



Parallel Processing (cont'd)

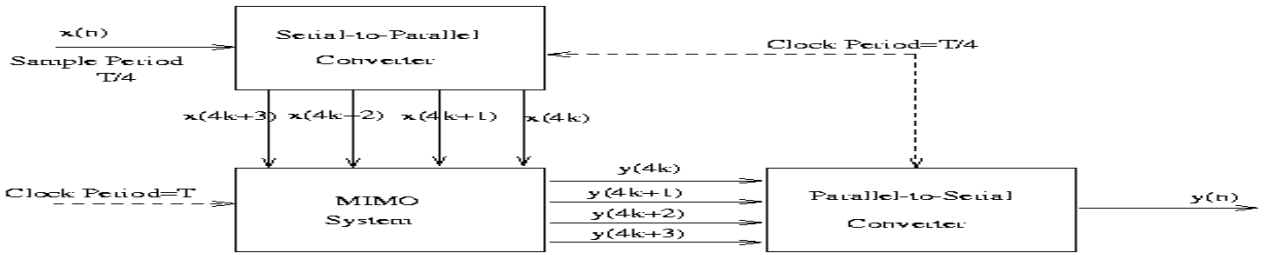
- Note 2: The critical path of the block (or parallel) processing system remains unchanged. But since 3 samples are processed in 1 (not 3) clock cycle, the iteration (or sample) period is given by the following equations:

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

$$T_{iteration} = T_{sample} = \frac{T_{clock}}{L} \geq \frac{T_M + 2T_A}{3}$$

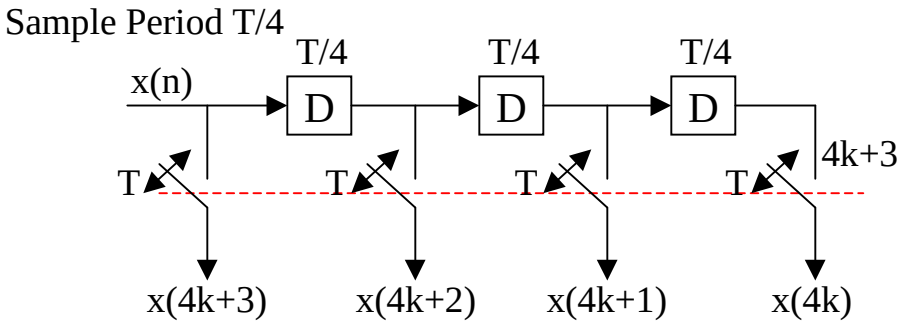
  - So, it is important to understand that in a parallel system $T_{sample} \neq T_{clock}$, whereas in a pipelined system $T_{sample} = T_{clock}$

- **Example:** A complete parallel processing system with block size 4 (including serial-to-parallel and parallel-to-serial converters) (also see P.72, Fig. 3.11)

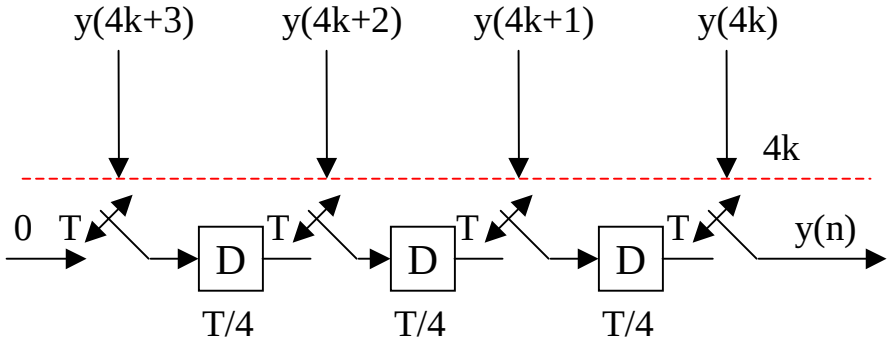


Parallel Processing (cont'd)

- A serial-to-parallel converter

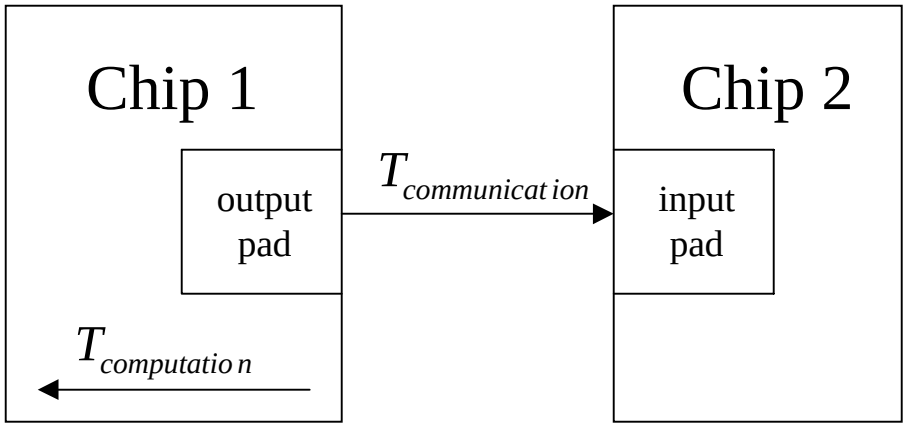


- A parallel-to-serial converter



Parallel Processing (cont'd)

- Why use parallel processing when pipelining can be used equally well?
  - Consider the following chip set, when the critical path is less than the I/O bound (output-pad delay plus input-pad delay and the wire delay between the two chips), we say this system is ***communication bounded***
  - So, we know that pipelining can be used only to the extent such that the critical path computation time is limited by the communication (or I/O) bound. Once this is reached, pipelining can no longer increase the speed



Parallel Processing (cont'd)

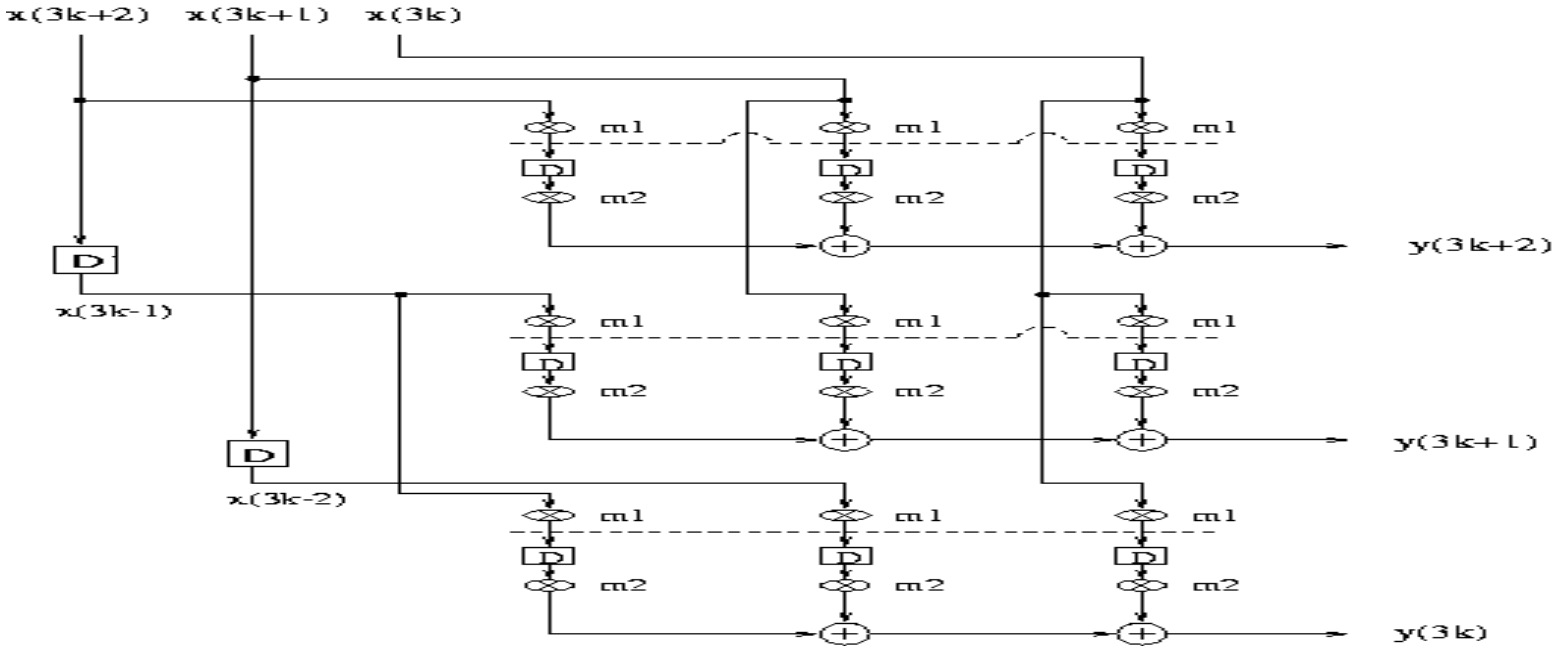
- So, in such cases, pipelining can be combined with parallel processing to further increase the speed of the DSP system
- By combining parallel processing (block size: L) and pipelining (pipelining stage: M), the sample period can be reduce to:

$$T_{iteration} = T_{sample} = \frac{T_{clock}}{L \cdot M}$$

- Example: (p.73, Fig.3.15) Pipelining plus parallel processing Example (see the next page)
- Parallel processing can also be used for reduction of power consumption while using slow clocks

Parallel Processing (cont'd)

Example:Combined fine-grain pipelining and parallel processing for 3-tap FIR filter



Pipelining and Parallel Processing for Low Power

- Two main advantages of using pipelining and parallel processing:
  - ***Higher speed*** and ***Lower power consumption***
- When sample speed does not need to be increased, these techniques can be used for lowering the power consumption
- Two important formulas:
  - Computing the propagation delay T_{pd} of CMOS circuit

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

Ccharge: the capacitance to be charged or discharged in a single clock cycle

  - Computing the power consumption in CMOS circuit

$$P_{CMOS} = C_{total} \cdot V_0^2 \cdot f$$

Ctotal: the total capacitance of the CMOS circuit

Pipelining and Parallel Processing for Low Power (cont'd)

- **Pipelining for Lower Power**
  - The power consumption in the original sequential FIR filter

$$P_{seq} = C_{total} \cdot V_0^{\,2} \cdot f, \quad f = 1/T_{seq}$$

T_{seq}: the clock period of the original sequential FIR filter

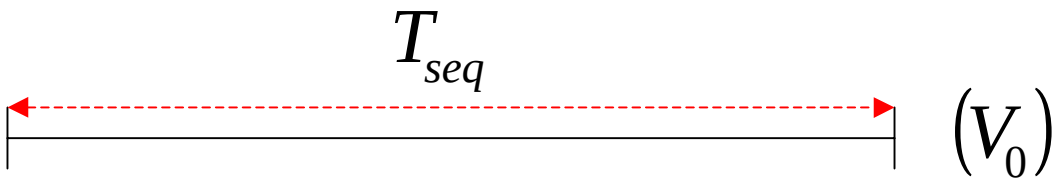
  - For a M-level pipelined system, its critical path is reduced to 1/M of its original length, and the capacitance to be charged/discharged in a single clock cycle is also reduced to 1/M of its original capacitance
  - If the same clock speed (clock frequency f) is maintained, only a fraction (1/M) of the original capacitance is charged/discharged in the same amount of time. This implies that the supply voltage can be reduced to βV_o ($0<\beta<1$). Hence, the power consumption of the pipelined filter is:

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

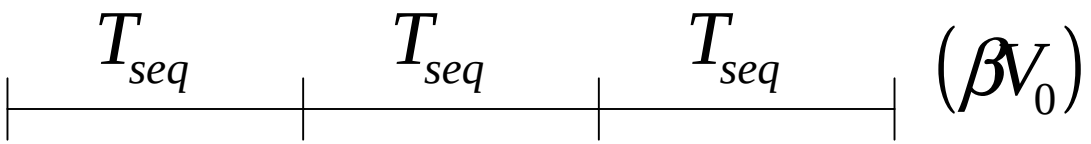
Pipelining and Parallel Processing for Low Power (cont'd)

- The power consumption of the pipelined system, compared with the original system, is reduced by a factor of β^2
- How to determine the power consumption reduction factor β?
  - Using the relationship between the propagation delay of the original filter and the pipelined filter

Sequential (critical path):



Pipelined: (critical path when M=3)



Pipelining and Parallel Processing for Low Power (cont'd)

- The propagation delays of the original sequential filter and the pipelined FIR filter are:

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

- Since the same clock speed is maintained in both filters, we get the equation to solve β:

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

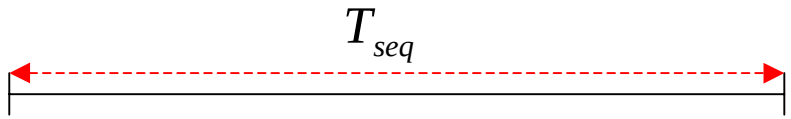
- Example: Please read textbook for Example 3.4.1 (pp.75)

Pipelining and Parallel Processing for Low Power (cont'd)

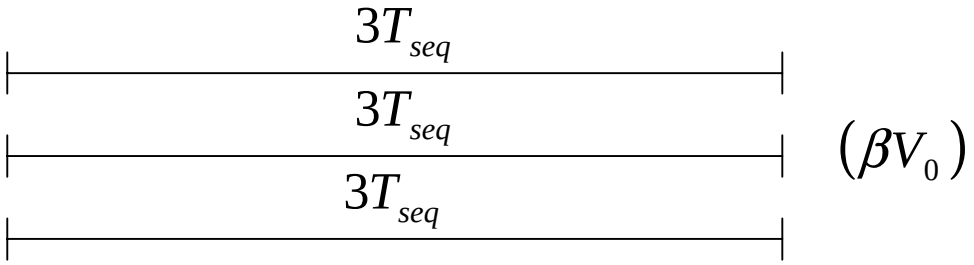
- **Parallel Processing for Low Power**
  - In an L-parallel system, the charging capacitance does not change, but the total capacitance is increased by L times
  - In order to maintain the same sample rate, the clock period of the L-parallel circuit is increased to LT_{seq} (where T_{seq} is the propagation delay of the original sequential circuit).
  - This means that the charging capacitance is charged/discharged L times longer (i.e., LT_{seq}). In other words, the supply voltage can be reduced to βV_o since there is more time to charge the same capacitance
  - How to get the power consumption reduction factor β?
    - The propagation delay consideration can again be used to compute β (Please see the next page)

Pipelining and Parallel Processing for Low Power (cont'd)

Sequential(critical path):



Parallel: (critical path when L=3)



- The propagation delay of the original system is still same, but the propagation delay of the L-parallel system is given by

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

Pipelining and Parallel Processing for Low Power (cont'd)

- Hence, we get the following equation to compute β:

$$L(\beta V_0 - V_t)^2 = \beta (V_0 - V_t)^2$$

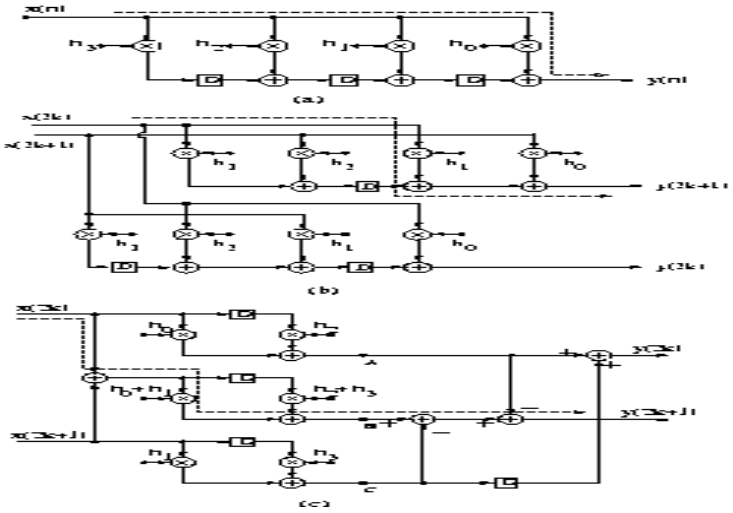
- Once β is computed, the power consumption of the L-parallel system can be calculated as

$$P_{para} = (LC_{total})(\beta V_0)^2 \frac{f}{L} = \beta^2 \cdot P_{seq}$$

– **Examples:** Please see the examples (Example 3.4.2 and Example 3.4.3) in the textbook (pp.77 and pp.80)

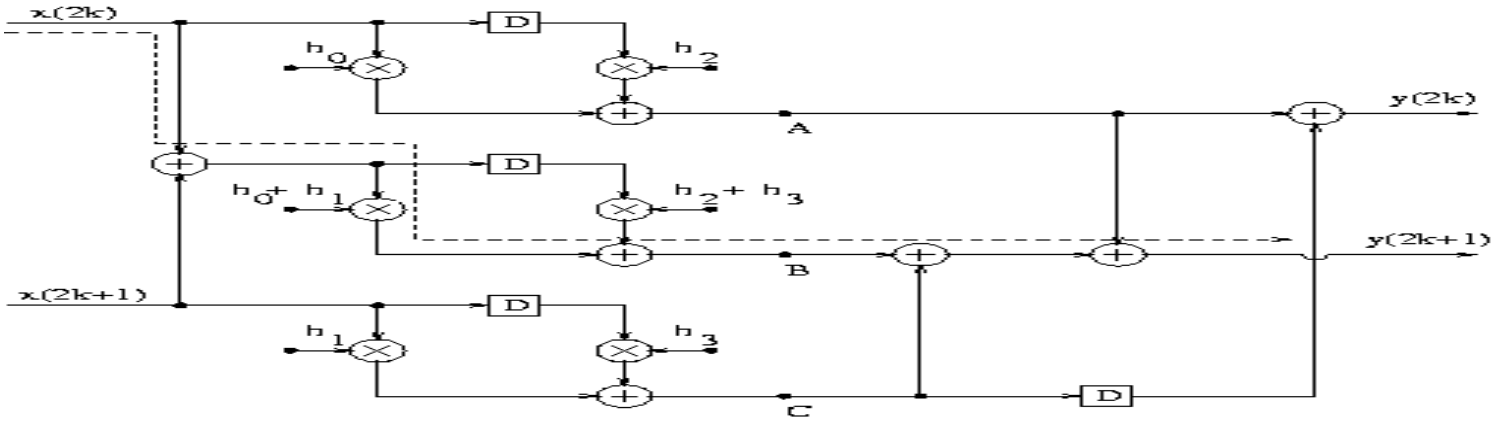
Figures for Example 3.4.2

- A 4-tap FIR filter
- A 2-parallel filter
- An area-efficient 2-parallel filter



Figures for Example 3.4.3 (pp.80)

An area-efficient 2-parallel filter and its critical path



Pipelining and Parallel Processing for Low Power (cont'd)

- **Combining Pipelining and Parallel Processing for Lower Power**
  - Pipelining and parallel processing can be combined for lower power consumption: pipelining reduces the capacitance to be charged/discharged in 1 clock period, while parallel processing increases the clock period for charging/discharging the original capacitance

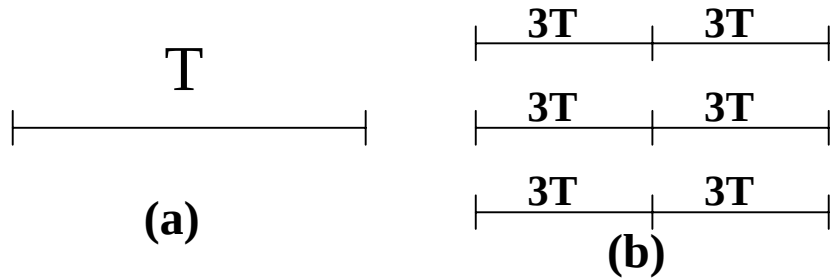


Figure: (a) Charge/discharge of entire capacitance in clock period T (b) Charge/discharge of capacitance in clock period 3T using a 3-parallel 2-pipelined FIR filter

Pipelining and Parallel Processing for Low Power (cont'd)

– The propagation delay of the L-parallel M-pipelined filter is obtained as:

$$LT_{pd} = \frac{\left(C_{ch\,\mathrm{arg}\,e} / M\right) \cdot \beta V_0}{k(\beta V_0 - V_t)^2} = \frac{L \cdot C_{ch\,\mathrm{arg}\,e} \cdot V_0}{k(V_0 - V_t)^2}$$

– Finally, we can obtain the following equation to compute β

$$LM \cdot (\beta V_0 - V_t)^2 = \beta (V_0 - V_t)^2$$

– Example: please see the example in the textbook (pp.82)