Chapter 3
Pipelining and parallel Processing

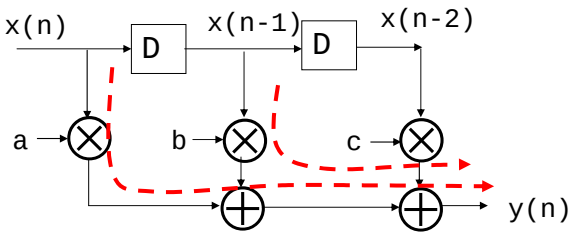
CSE4210 Winter 2012

Mokhtar Aboelaze

Pipelining -- Introduction

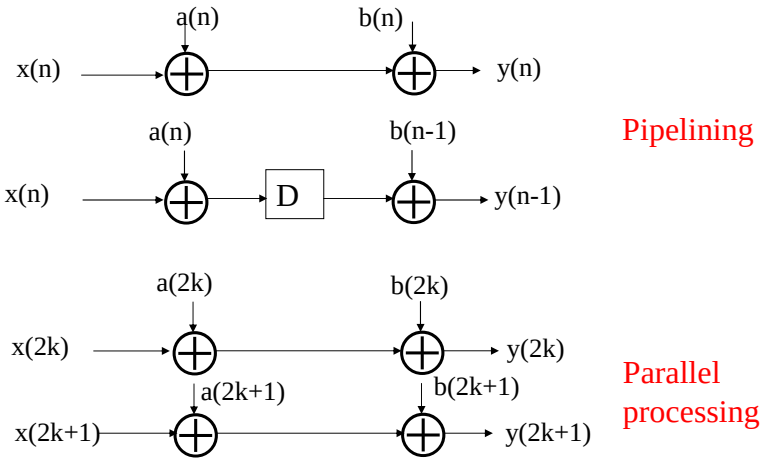
- Pipelining can be used to reduce the the critical path.
- That can lead to either increasing the clock speed, or decreasing the power consumption
- Multiprocessing can be also used to increase speed or reduce power.

Pipelining



The critical path here is two additions and one multiplication

$$T_s = 2T_{add} + T_{mul} \quad , \quad f_s = 1/(2T_{add} + T_{mul})$$



Pipelining

Parallel processing

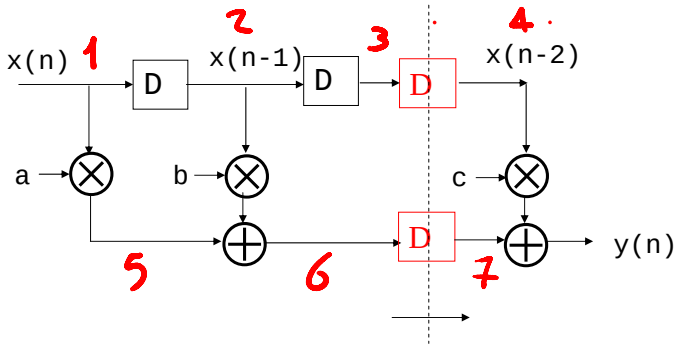
Pipelining

- Advantages
  - Could be used to reduce power and/or to increase clock rate (speed)
- Disadvantages
  - Increases number of delay elements (latches or flip-flops)
  - Increases latency

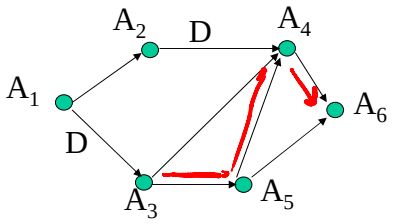
Pipelining

- **Cutset**: is a set of edges of a graph if removed, the graph becomes partitioned
- **Feed forward cutset**: a cutset where the data moves in the forward direction on all the edges in the cutset
- We can place latches on a **feed-forward cutset** without affecting the functionality of the graph.

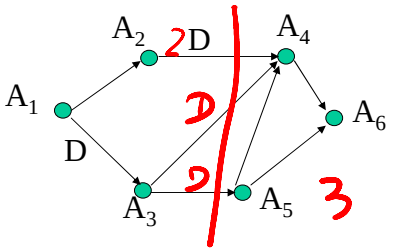
Pipelining

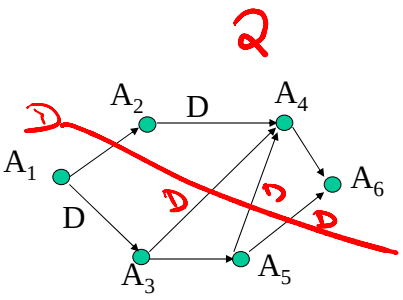


Example



Critical Path?

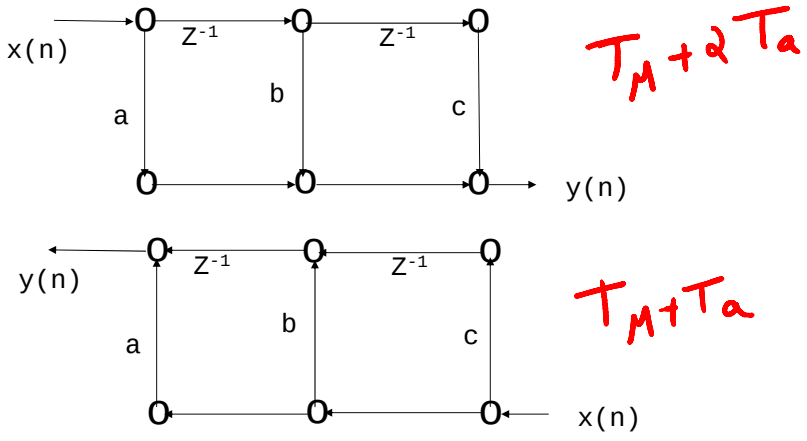




Data Broadcast Structures

- Reversing the direction of all the edges in a given SFG, and interchanging the ,input and output preserves the functionality of the system.

Data Broadcast



$T_M + 2T_a$

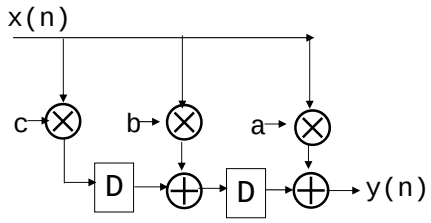
$T_M + T_a$

Fine-Grain Pipelining

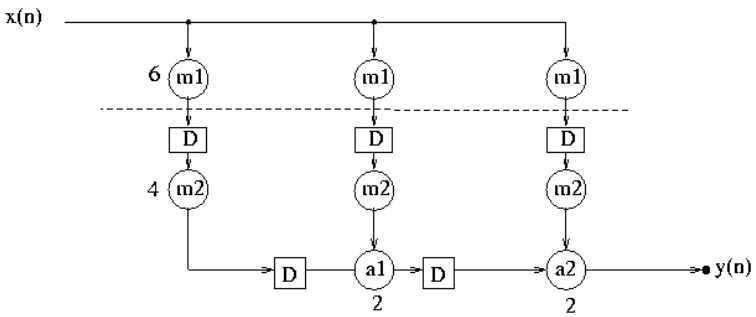
Multiplication time = 10

Addition time = 2

Critical path = ?



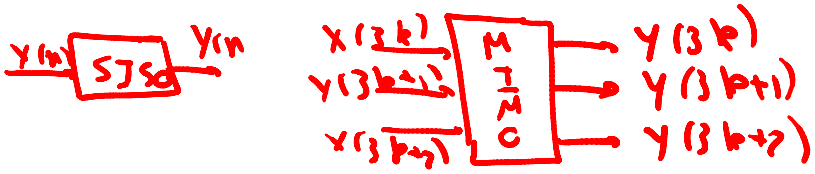
Fine grain Pipelining

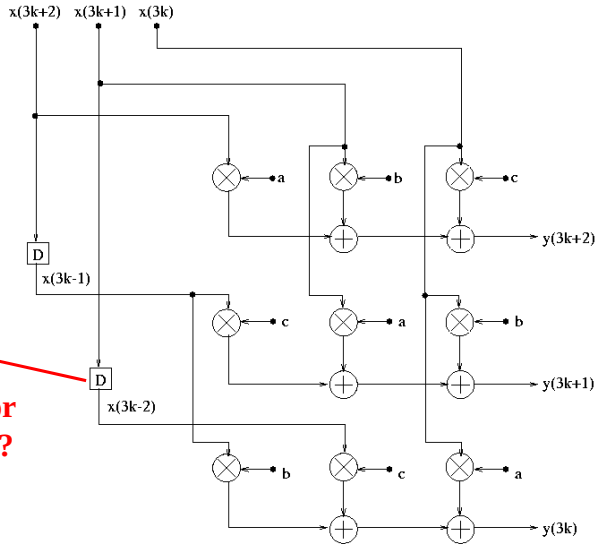


Parallel Processing

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

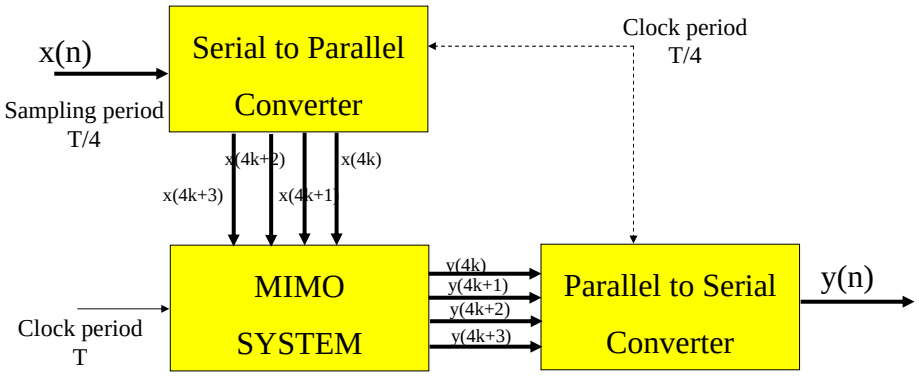
$$y(3k) = ax(3k) + bx(3k-1) + cx(3k-2)$$
$$y(3k+1) = ax(3k+1) + bx(3k) + cx(3k-1)$$
$$y(3k+2) = ax(3k+2) + bx(3k+1) + cx(3k)$$



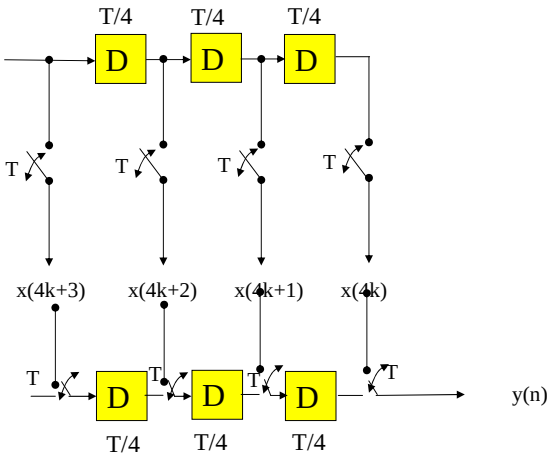


X(3k) or x(3k-2)?

Complete Parallel System



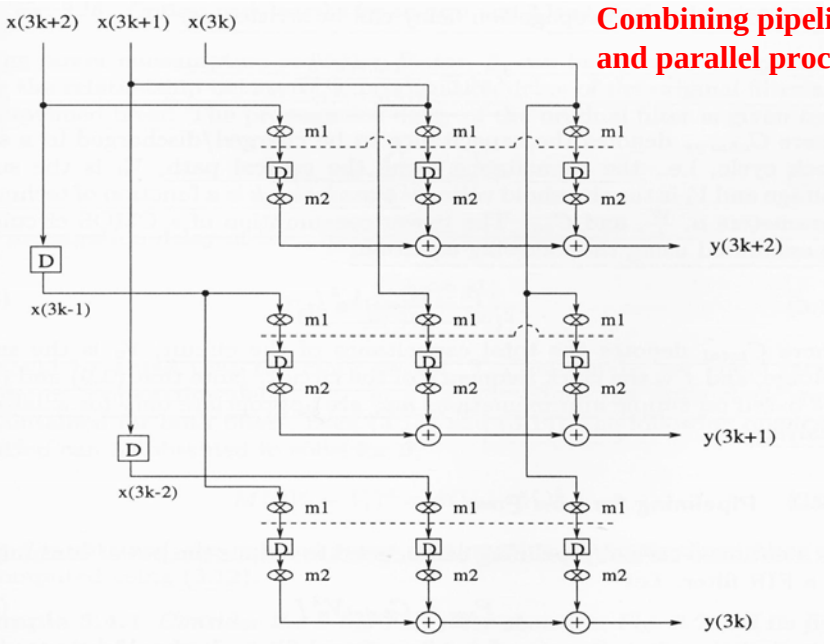
S/P and P/S Converter



Parallel Processing

- Why use parallel processing? It increases the hardware.
- There is a limit for the use of pipelining, you may not be able to pipeline a functional unit beyond a certain limie
- Also, I/O usually imposes a bound on the cycle time (communication bound)

Combining pipelining and parallel processing



Low Power

$$P = C_{total} V_o^2 f$$

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

Simple approximation for CMOS

C_{total} is the total capacitance of the circuit, Vo is the supply voltage. C_{charge} is the capacitance to be charged/discharged in a single clock cycle.

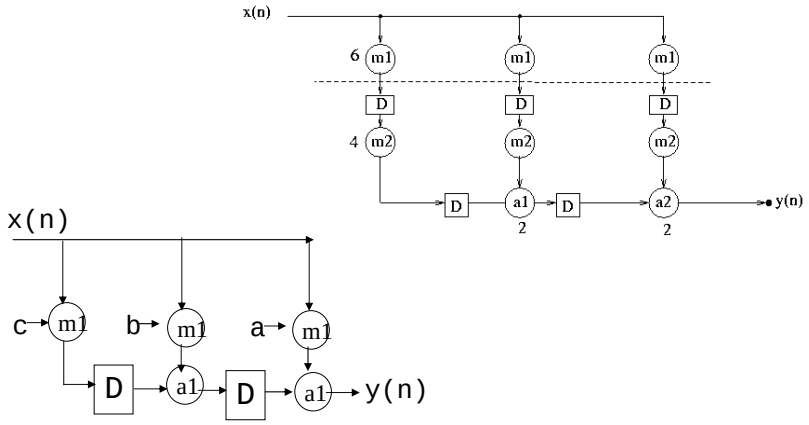
Pipelining and parallel processing could be used to minimize power or execution time.

Low Power

- What happens in case of M –pipelining?
- Critical path is reduced by $\boldsymbol{M}$, so is C_{charge}
- If we keep the same f, then we have more time to charge C_{charge}, which means we can reduce the supply voltage

$$T_{seq} = \frac{C_{\text{charge}} V_o}{k(V_o - V_t)^2} = T_{pip} = \frac{\frac{C_{\text{charge}}}{M} \beta V_o}{k(\beta V_o - V_t)^2}, P = ?$$

Low Power —Example



Example

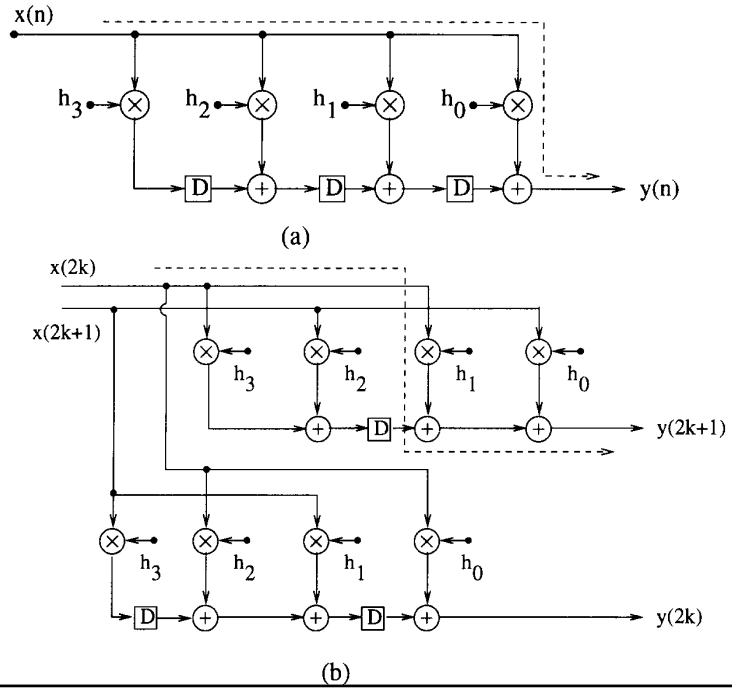
Parallel processing

- What happen for L parallel System?
- Total capacitance increased by L
- Same performance, increase T by L
- More time to charge C_{charge}, can decrease V

$$LT_{seq} = L\frac{C_{\text{charge}}V_o}{k(V_o - V_t)^2} = T_{par} = \frac{C_{charge}\ \beta V_o}{k(\beta V_o - V_t)^2}, P = ?$$

Parallel Processing Example

- Consider a 4-tap FIR filter shown in Fig. 3.18(a) and its 2-parallel version in 3.18(b). The parallel filter has exactly 2 copies of the original filter. The dashed line donates the critical path.
- Assume Also that T_M=8, T_A=1, V_t=0.45V, V_o=3.3V, C_M=8C_A
  - What is the supply voltage of the 2-parallel filter?
  - What is the power consumption of the 2-parallel filter as a percentage of the original filter?


x(n)
h3
h2
h1
h0
D
D
D
y(n)
(a)
x(2k)
x(2k+1)
h3
h2
h1
h0
D
y(2k+1)
h3
h2
h1
h0
D
D
y(2k)
(b)

Example

Different Architecture

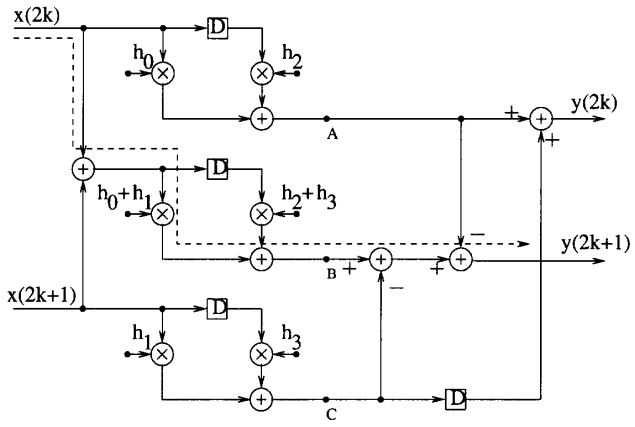


Fig. 3.18 (a) A 4-tap FIR filter. (b) A 2-parallel filter. (c) An area-efficient 2-parallel filter.