ELE617E
Lectures

Prof. Dr. Müştak E. Yalçın

Istanbul Technical University

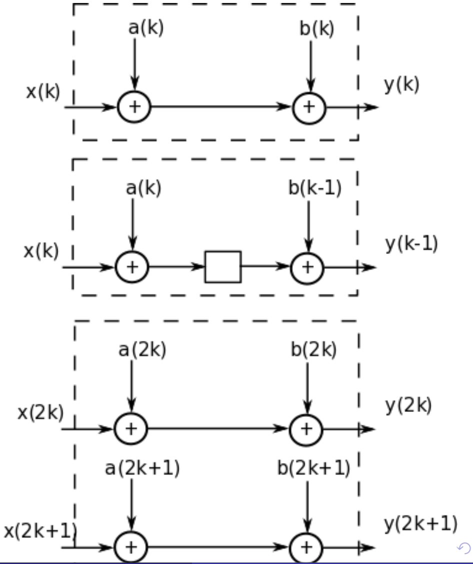
mustak.yalcin@itu.edu.tr

Pipelining and Parallel Processing

A datapath: $T_{\text{critical}} = 2T_A$ and $T_s = T_{clk} \geq 2T_A$

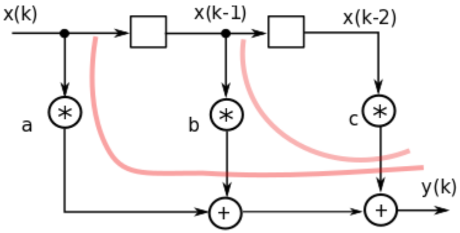
Pipelining processing:
Introduce latches along the datapath.
$T_{\text{critical}} = T_A$ and $T_s = T_{clk} \geq T_A$
latancey !

Parallel processing:
Duplicate the hardware
2 output at the same time!
$T_s = T_{clk}/2$ and $T_{clk} \geq 2T_A$



Pipelining

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

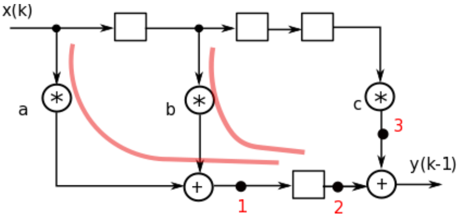


$T_{\text{critical}} = T_M + 2T_A$ then $T_s \geq T_{\text{critical}}$

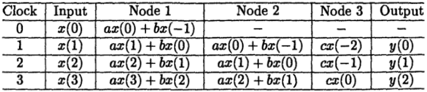
The effective critical path can be reduced by using pipeling.

HOW: introducing pipelining latches along the datapath.

Pipelining



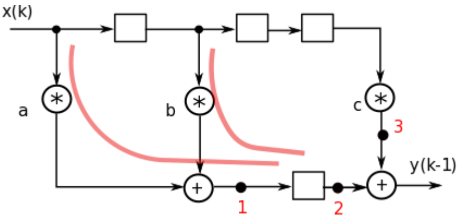
$$T_{\text{critical}} = T_M + T_A$$



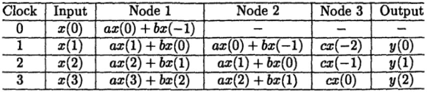
Clock	Input	Node 1	Node 2	Node 3	Output
0	$x(0)$	$ax(0)+bx(-1)$	–	–	–
1	$x(1)$	$ax(1)+bx(0)$	$ax(0)+bx(-1)$	$cx(-2)$	$y(0)$
2	$x(2)$	$ax(2)+bx(1)$	$ax(1)+bx(0)$	$cx(-1)$	$y(1)$
3	$x(3)$	$ax(3)+bx(2)$	$ax(2)+bx(1)$	$cx(0)$	$y(2)$

Pipelining reduces the critical path but it leads to a penalty:

Pipelining



$$T_{\text{critical}} = T_M + T_A$$



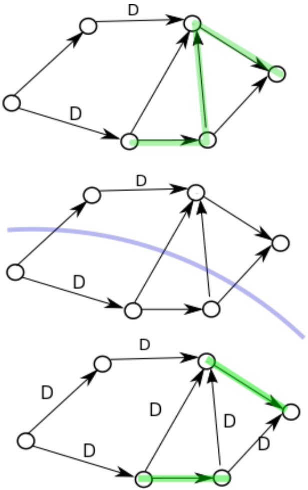
Clock	Input	Node 1	Node 2	Node 3	Output
0	$x(0)$	$ax(0)+bx(-1)$	$-$	$-$	$-$
1	$x(1)$	$ax(1)+bx(0)$	$ax(0)+bx(-1)$	$cx(-2)$	$y(0)$
2	$x(2)$	$ax(2)+bx(1)$	$ax(1)+bx(0)$	$cx(-1)$	$y(1)$
3	$x(3)$	$ax(3)+bx(2)$	$ax(2)+bx(1)$	$cx(0)$	$y(2)$

Pipelining reduces the critical path but it leads to a penalty: increase in latency! + latches

Pipelining

- Drawbacks:
    - Increase latency
    - Increase number of delay elements (registers/latches) in the critical path
- Clock period limitation: critical path may be between
    - An input and a latch
    - A latch and an output
    - 2 Latches
    - An input and an output
- Pipelining latches can only be placed 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: A cutset is called a feed-forward cutset if the data move in the forward direction on all the edges of the cutset.

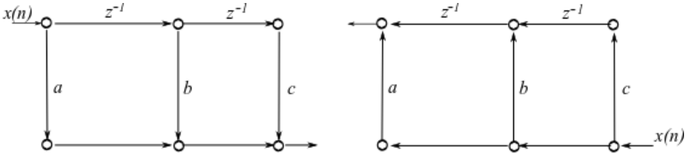
Pipelining

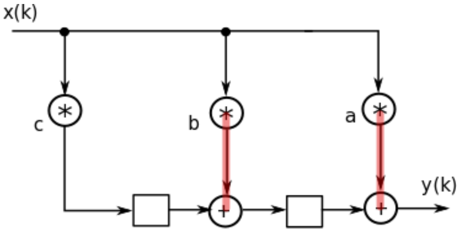


$T_{\text{critical}} = 4u.t.$ to $T_{\text{critical}} = 2u.t.$ In the 2-level pipelined system, the number of delay elements in any path from the input to the output is increased by 1.

Transposition Theorem

Reversing the direction of all edges in a given SFG and interchanging the input and output ports preserve the functionality of the system.

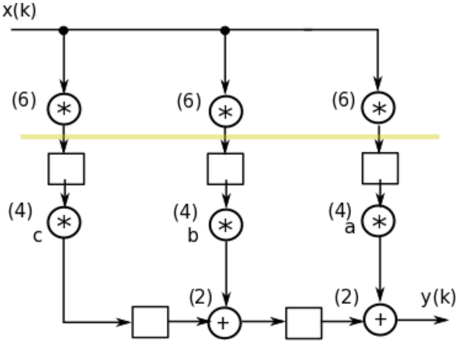




$$T_{\text{critical}} = T_A + T_M$$

Fine-Gain Pipelining

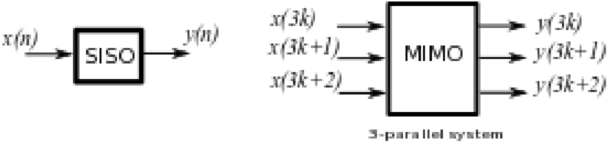
Break the MULTIPLIER into 2 smaller units with processing time of 6 and 4 units.



Parallel Processing

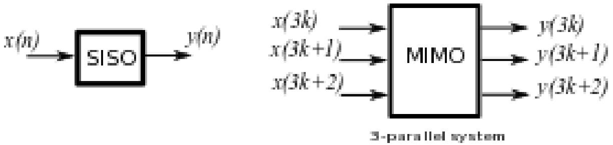
If a computation can be pipelined, it can be also be processed in parallel. To obtain a parallel processing structure, SISO system must be converted into a MIMO system.

Level of parallel processing ($L = 3$):3 input per clock cyc.



Parallel Processing = Block processing systems: number of inputs procedded in a clock is referred to as the block size.
Placing a latch at any line produces an effecrtive delay of L clock cyc. (block delay or L-slow). Delaying $x(3k)$ by 1 clock would results in $x(3(k-1))$!

Parallel Processing



Critical path of Parallel Processing system has remained unchanged!

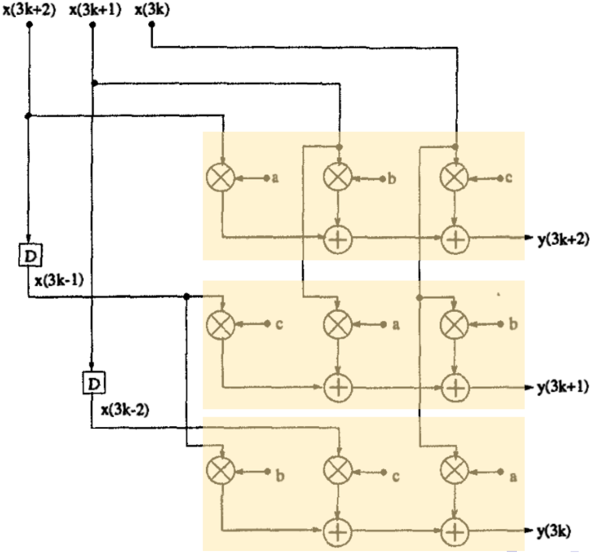
$$T_{clk} \geq T_{\text{critical}} = T_M + 2T_A$$

After 3-parallel system: 3 samples processes in 1 clock cyc.

$$T_{itr} = T_s = \frac{1}{L} T_{clk}$$

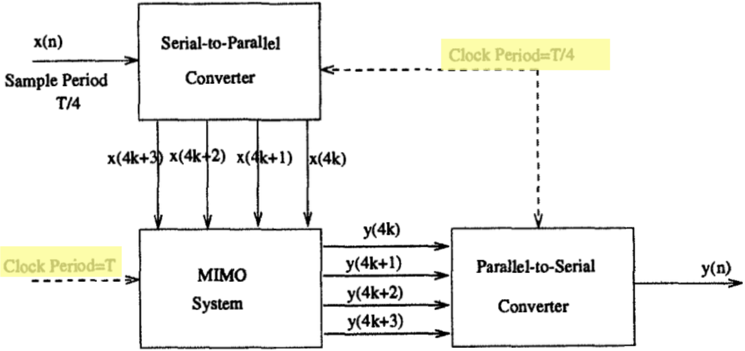
Parallel Processing

Paralle Processing Architecture for 3-tap filter:



Parallel Processing

Complete Paralle Processing System



$$T_s \neq T_{clk}$$

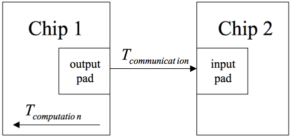
See: serial-to-parallel (Fig 3.12) and parallel-to-serial converters (Fig. 3.13)

Parallel Processing

Why use paralle processing when we can use pipelining equally well ? Why to duplicate the HW ?

Input/Output bottlenecks !

- 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.

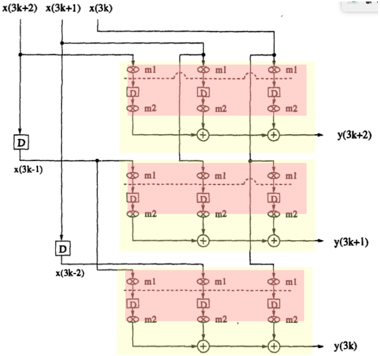


Parallel & Pipelining Processing

By combining parallel processing (block size: L) and pipelining (pipelining stage: M), the sample period can be reduce to:

$$T_{itr} = T_s = \frac{1}{ML} T_{clk}$$

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


x(3k+2)
x(3k+1)
x(3k)
m1
D
m2
D
x(3k-1)
D
x(3k-2)
y(3k+2)
y(3k+1)
y(3k)