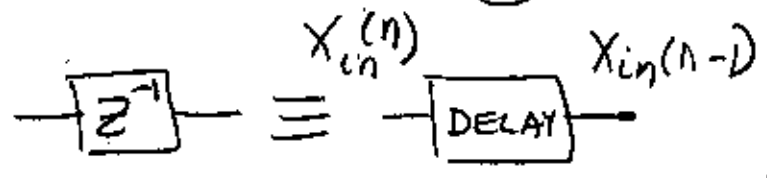
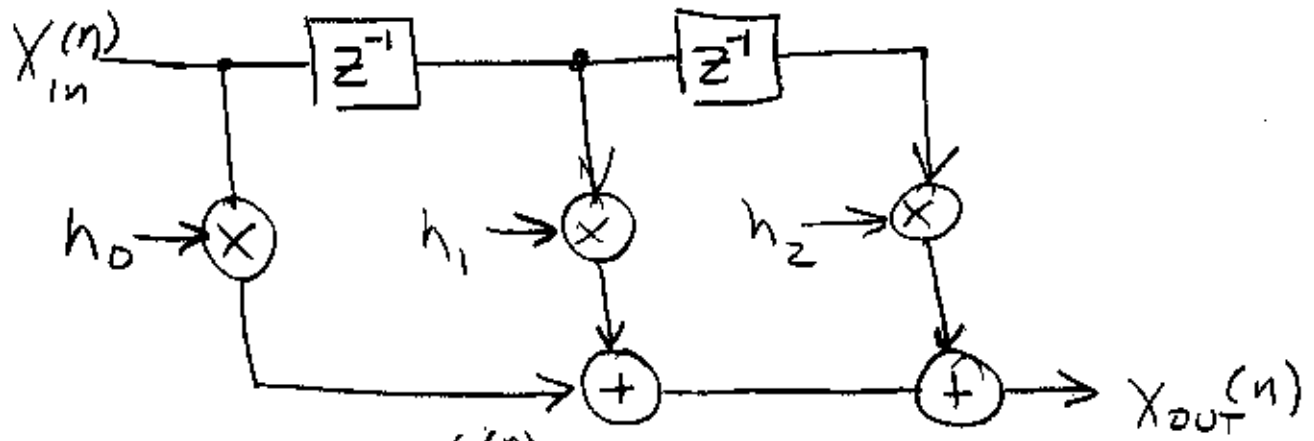


SIGNAL FLOW GRAPH MANIPULATION

LEC 7

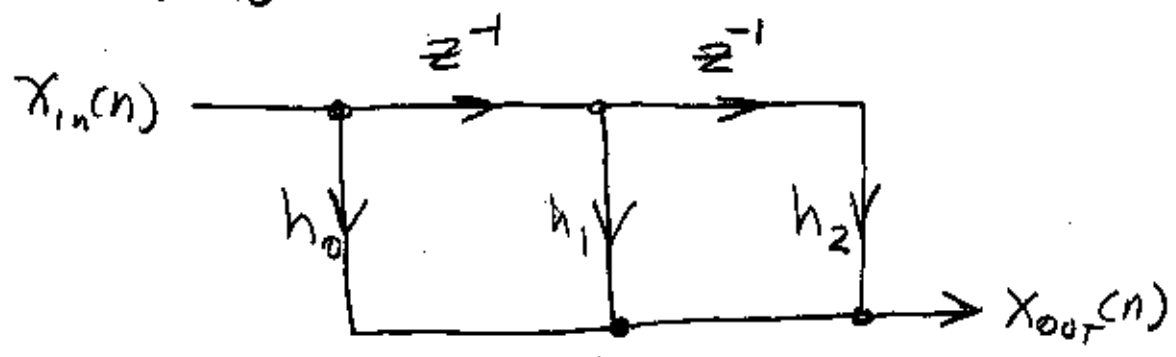
READING - PARHI BOOK ON WEB SITE MANY PAPERS

LETS START WITH OUR FIR FILTER



A DELAY IS BY ONE SAMPLE PERIOD

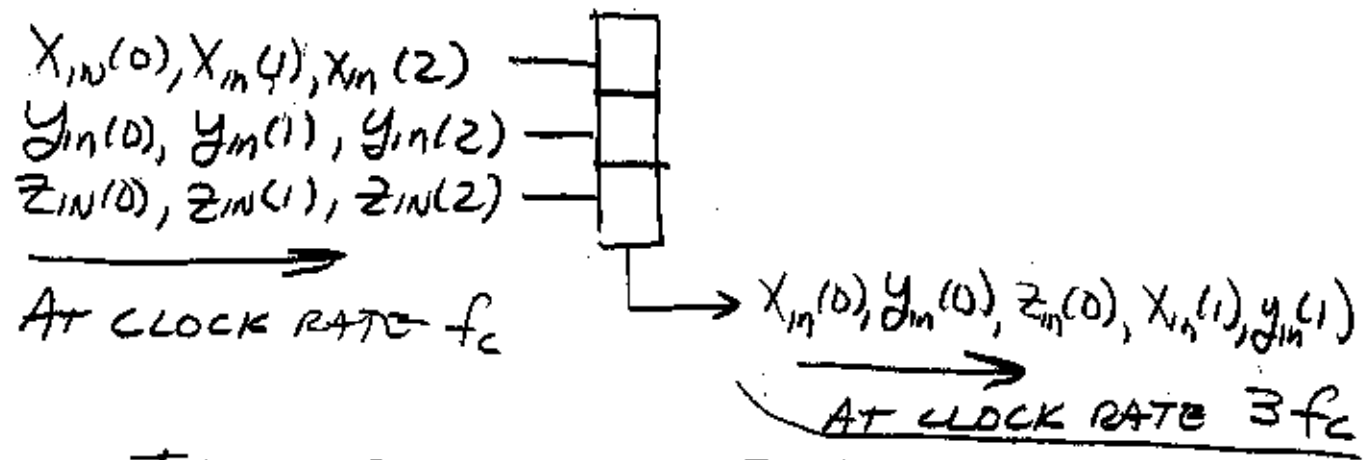
A MORE ABSTRACT AND EFFICIENT NOTATION



ASSUME AN ADD WHEN NODES COME TOGETHER

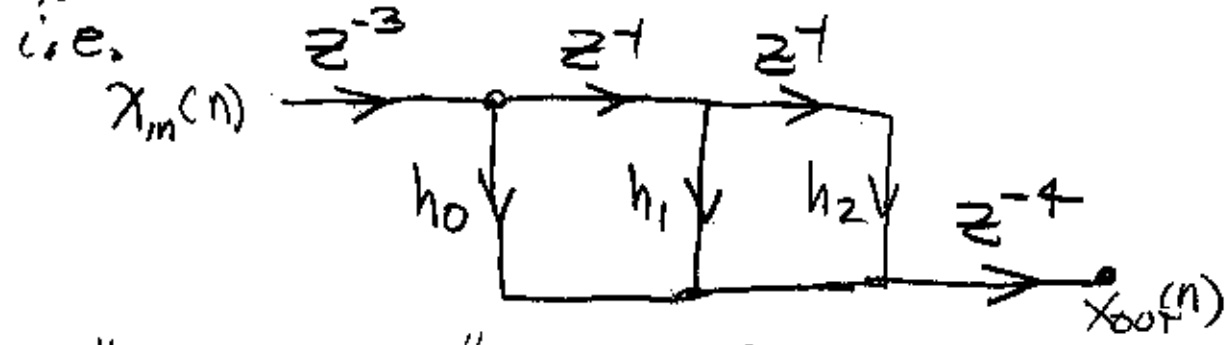
2) SAMPLE INTERLEAVING (CONT.)

MULTIPLE DATA STREAMS CAN BE MULTIPLEXED INTO ONE SFG



THIS GOES INTO SFG ON PREVIOUS PAGE

3) CHANGING THE LATENCY "USUALLY" (DELAY FROM INPUT TO OUTPUT) IS NOT CONSIDERED IMPORTANT

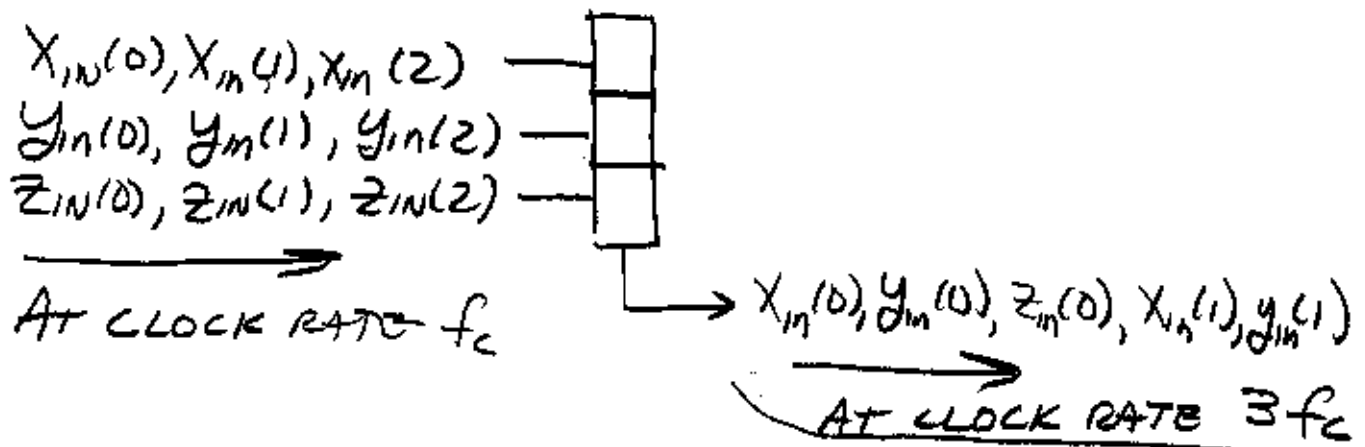


IS "EQUIVALENT" TO OUR ORIGINAL FILTER.

THE QUESTION IS "HOW TO OPTIMALLY USE ALL THESE EXTRA DELAYS TO REDUCE THE CRITICAL PATH?"

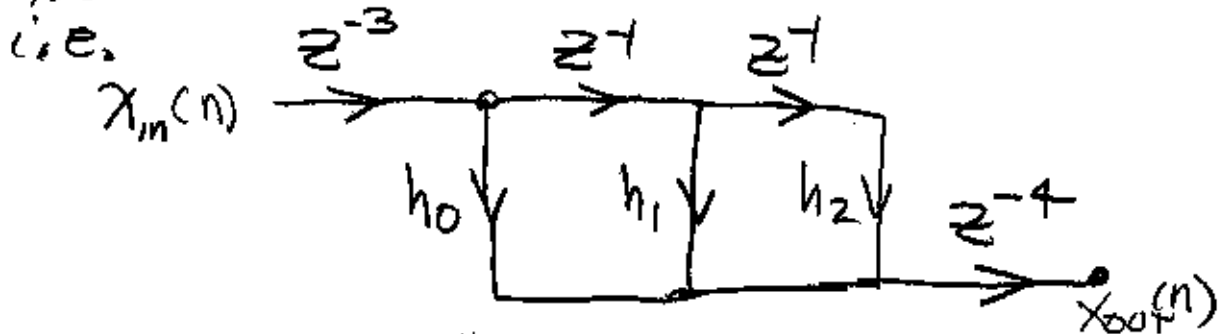
2) SAMPLE INTERLEAVING (CONT.)

MULTIPLE DATA STREAMS CAN BE MULTIPLEXED INTO ONE SFG



THIS GOES INTO SFG ON PREVIOUS PAGE

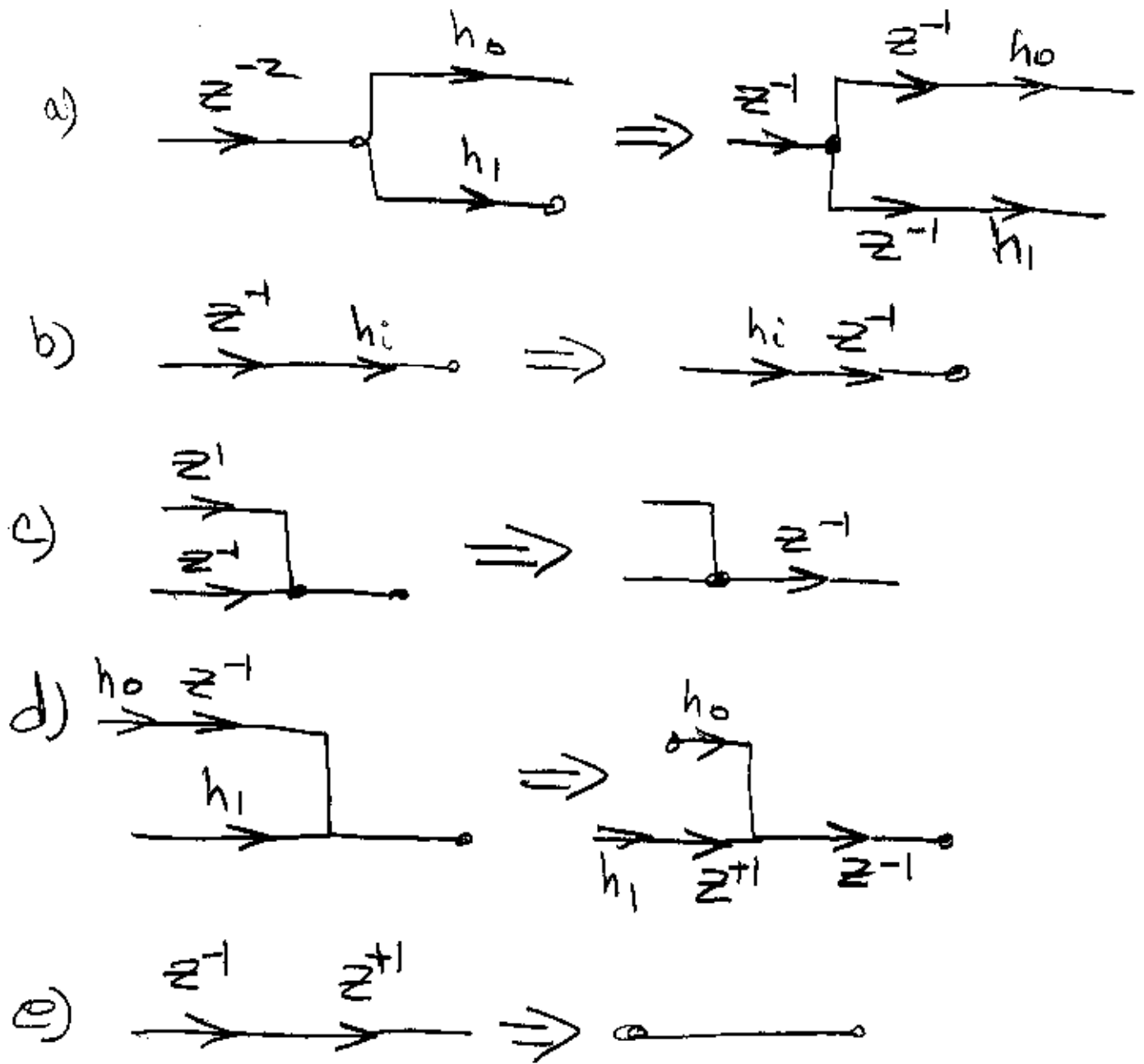
3) CHANGING THE LATENCY "USUALLY" (DELAY FROM INPUT TO OUTPUT) IS NOT CONSIDERED IMPORTANT



IS "EQUIVALENT" TO OUR ORIGINAL FILTER.

THE QUESTION IS "HOW TO OPTIMALLY USE ALL THESE EXTRA DELAYS TO REDUCE THE CRITICAL PATH?"

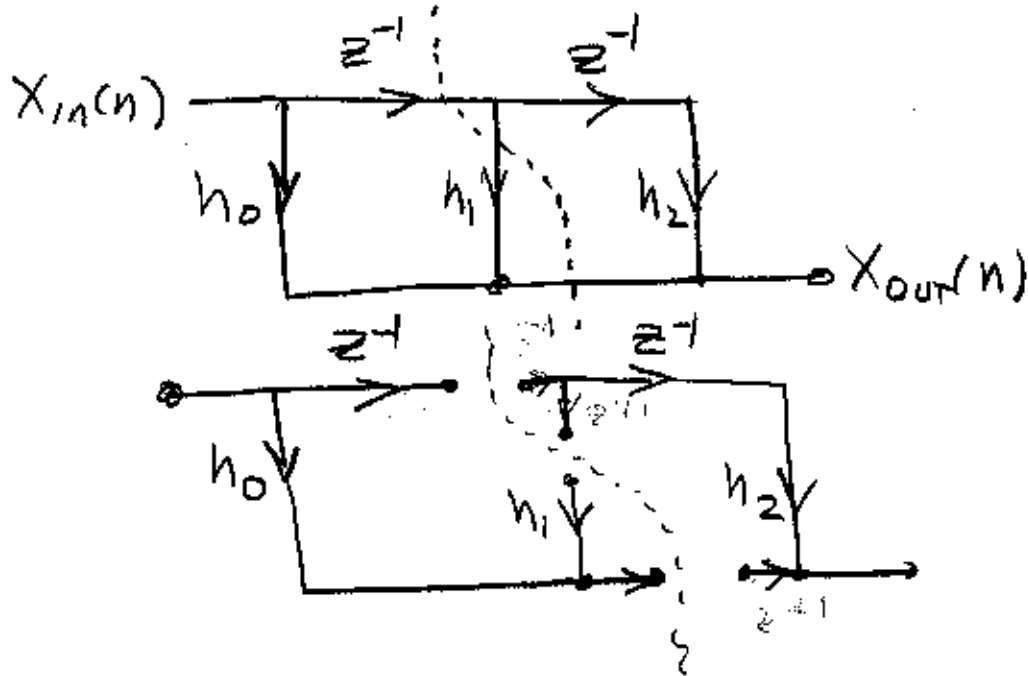
4) NODAL DELAY TRANSFER



SINCE z^{+1} ARE NON-CAUSAL THEY MUST BE REMOVED FROM THE SFG BY COMBINING WITH z^{-1} 'S TO BE IMPLEMENTABLE

DEFINITION OF A CUTSET:

SEPARATES THE SFG INTO TWO DISJOINT GRAPHS



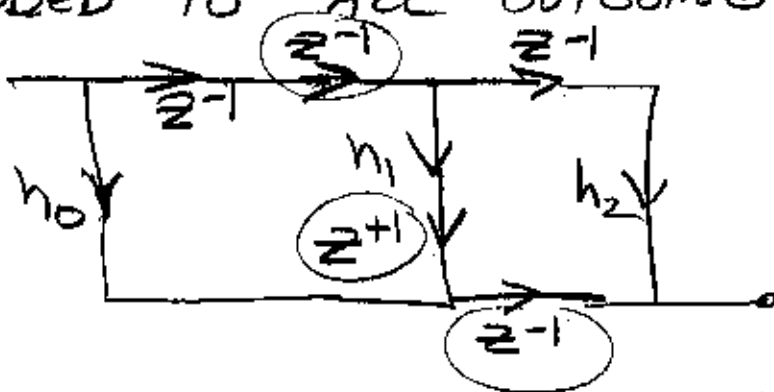
5) CUTSET

RETIMING

USES A GENERALIZATION OF 4 ABOVE

DELAY CAN BE ADDED TO ALL INCOMING EDGES TO A CUTSET IF ADVANCES ARE ADDED TO ALL OUTGOING EDGES

AND VICE-VERSA



5) LCUTSET RETIMING ⑥

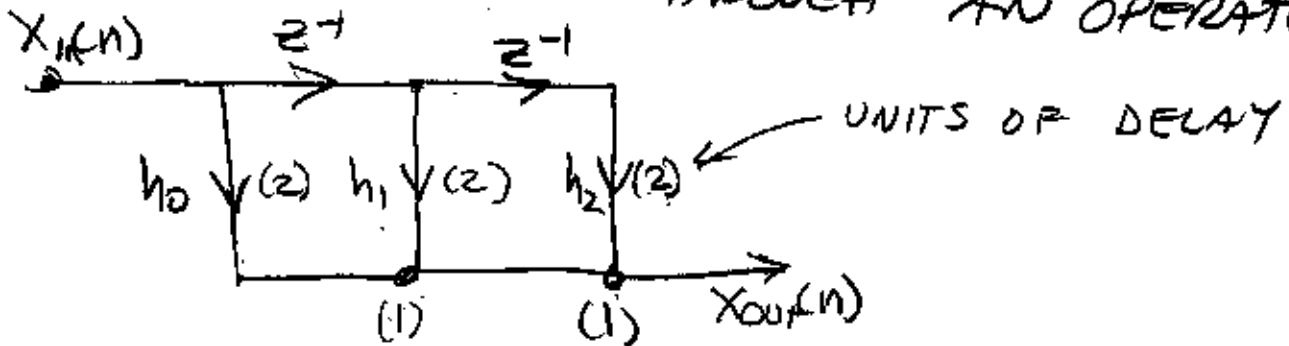
OPTIMIZES THE PLACEMENT OF DELAYS TO MINIMIZE THE CRITICAL PATH.

(OTHER FORMS OF RETIMING TO DECREASE THE NUMBER OF REGISTERS, REDUCE POWER, REDUCE THE CLOCK RATE, ETC.)

6) LEISERSON-SAXE RETIMING

DEVELOPED AN ALGORITHM TO OPTIMALLY PLACE THE DELAYS TO MINIMIZE THE CRITICAL PATH.

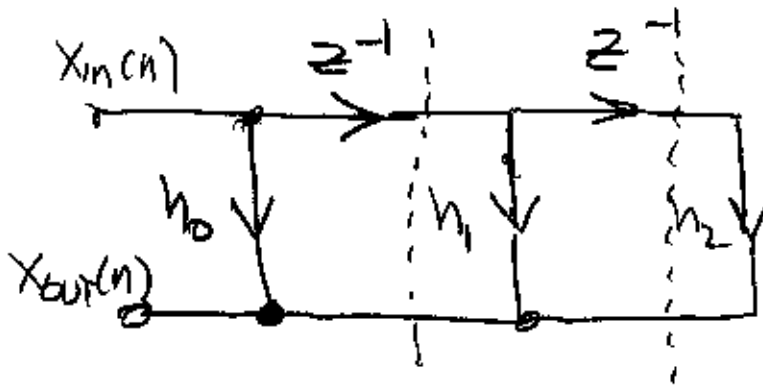
WE NEED TO ADD THE TIME IT TAKES TO GO THROUGH AN OPERATOR



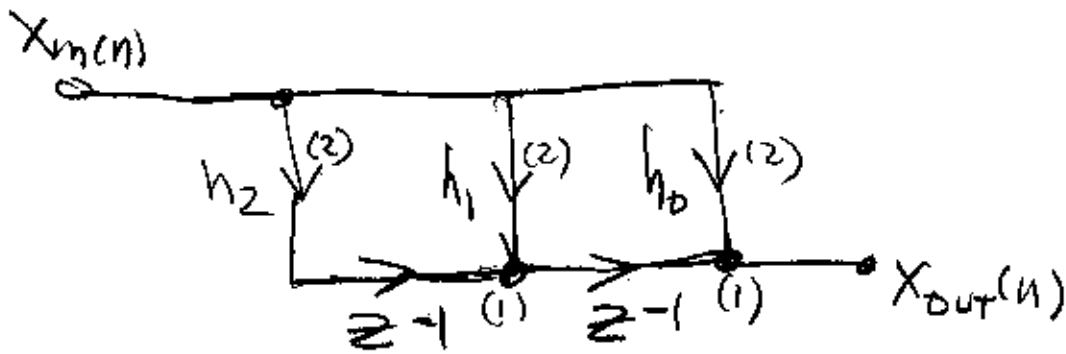
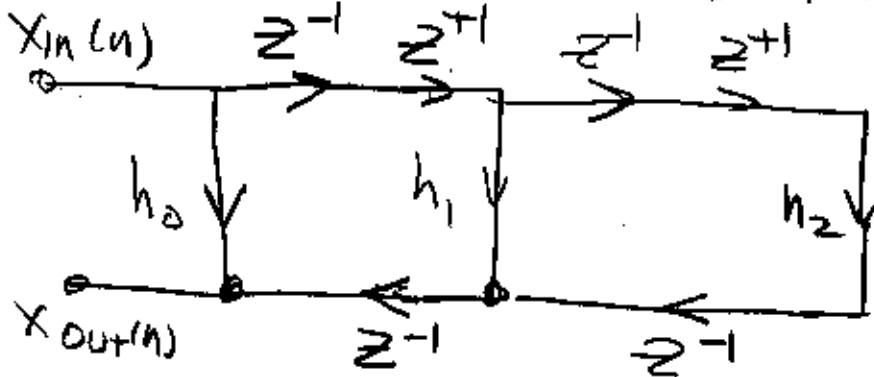
CRITICAL PATH IS "4" - CLOCK RATE IS THEN $1/4$ (IN SOME UNITS)

(7)

SINCE ADDS CAN BE PERFORMED IN ANY ORDER WE CAN REDRAW OUR FILTER



USING THESE TWO CUTSETS ADD AN ADVANCE TO THE TOP LINE



CRITICAL PATH IS NOW 3
MUCH MORE IMPROVEMENT FOR A
LARGER FILTER.

(8)

THERE ARE A NUMBER OF BOUNDS IN ADDITION TO CRITICAL PATH

MINIMUM CLOCK PERIOD - LONGEST ZERO DELAY PATH

$$T_{MIN} = \sum_i (T_{OP})_i$$

T_{IB} :

1) ITERATION BOUND ON CLOCK PERIOD

$D_L \equiv$ TOTAL OPERATOR DELAY IN EACH SIMPLE LOOP (NO REPEATING EDGES)

$N_L \equiv$ # OF DELAYS

$$T_{IB} = \max_{\text{LOOPS}} \left\{ \frac{D_L}{N_L} \right\}$$

(NOTE: RETIMING DOES NOT IMPROVE ITERATION BOUND)

2) PROCESSOR (OPERATOR) BOUND
ASSUME ALL DELAYS ARE OPTIMALLY PLACED

$$P_{BOUND} = \frac{\sum_i (T_{OP})_i}{\max \{ T_{IB}, T_{MIN} \}} \Rightarrow \text{NO. OF PROCESSORS NEEDED}$$

TOTAL TIME TO DO ALL COMPUTATION



Multiplierless FIR Filter Implementation

- For dedicated filtering applications canonic signed-digit (CSD) coefficients result in substantial hardware reductions
- Radix-2 canonic signed-digit code representation:

$$x = \sum_{k=1}^L s_k 2^{-p_k}$$

where

$s_k \in \{-1, 0, 1\}$ and $p_k \in \{0, 1, \dots, M\}$

L = number of nonzero (ternary) digits in code

$M+1$ = total number of digits in code



Properties of CSD Codes

- A given number has several signed-digit (SD) representations

- Canonic representation has minimum number of nonzero digits (also not necessarily unique)

Example of 4-digit radix-2 SD representation:

$$3 = 0011 = 010\bar{1} = 01\bar{1}1 = 1\bar{1}0\bar{1} = 1\bar{1}\bar{1}1$$

where $\bar{1}$ denotes -1

- Added flexibility of negative digits allows most numbers to be represented with much fewer nonzero digits

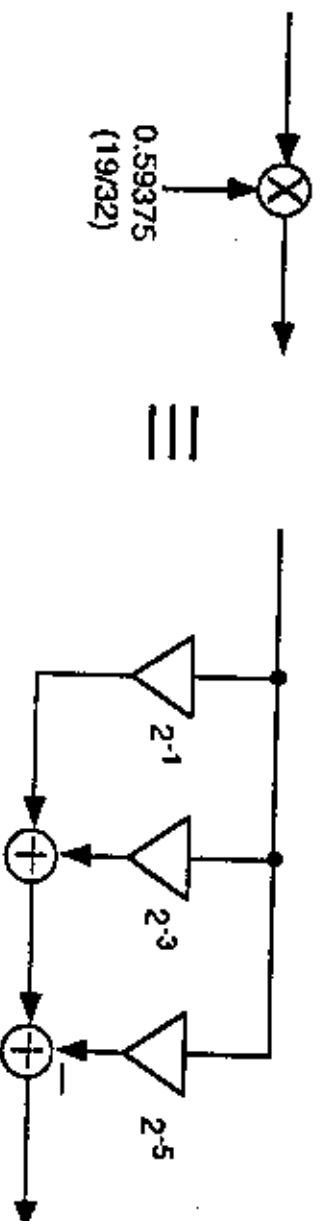
$$127 = 0111111\bar{1} = 1000000\bar{1}$$

(6 adders vs. 1 adder hardware complexity)



Multiplierless FIR Filter Implementation

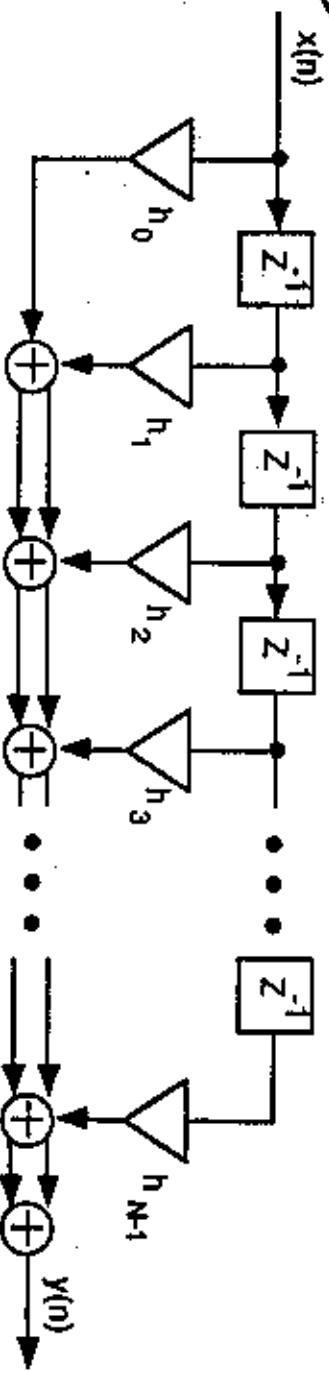
- Power-of-two multiplications are obtained for free by simply shifting data busses
- Example of CSD multiplier coefficient implementation



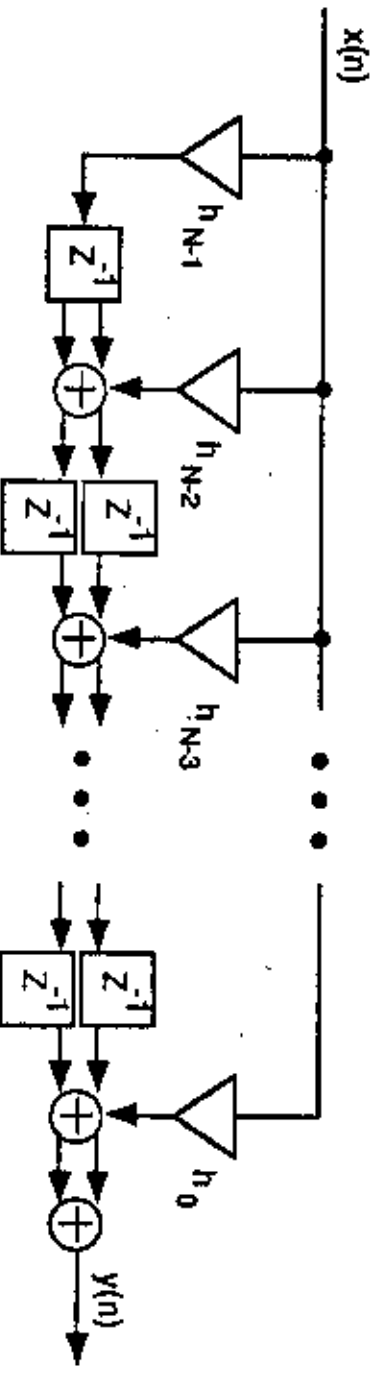
- Result is a low-complexity multiplierless FIR filter with little performance degradation from the ideal filter response



FIR Filter Architecture



Direct Form Architecture

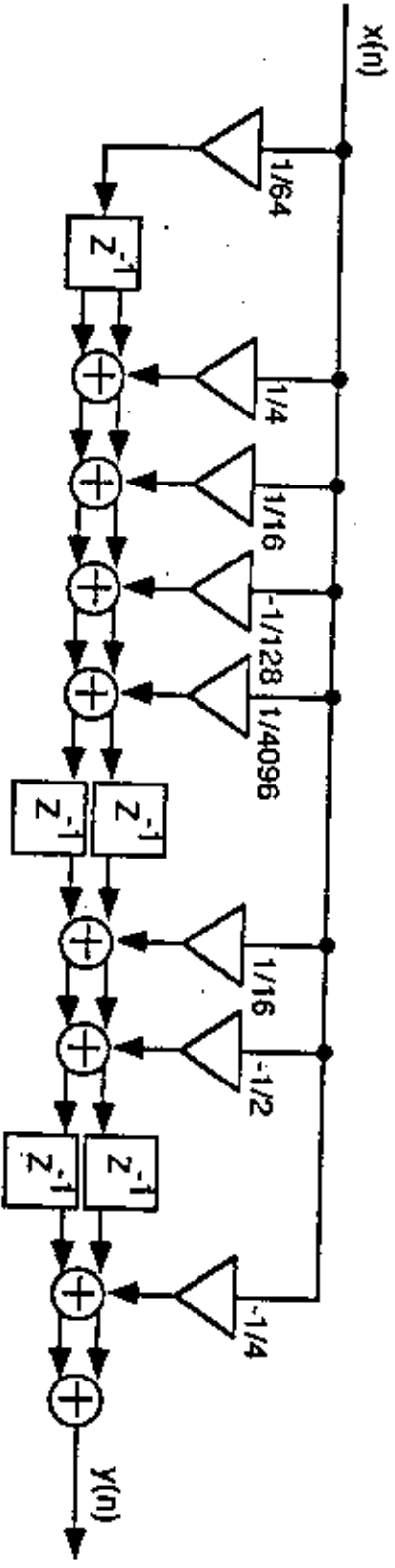


Transpose Direct Form Architecture

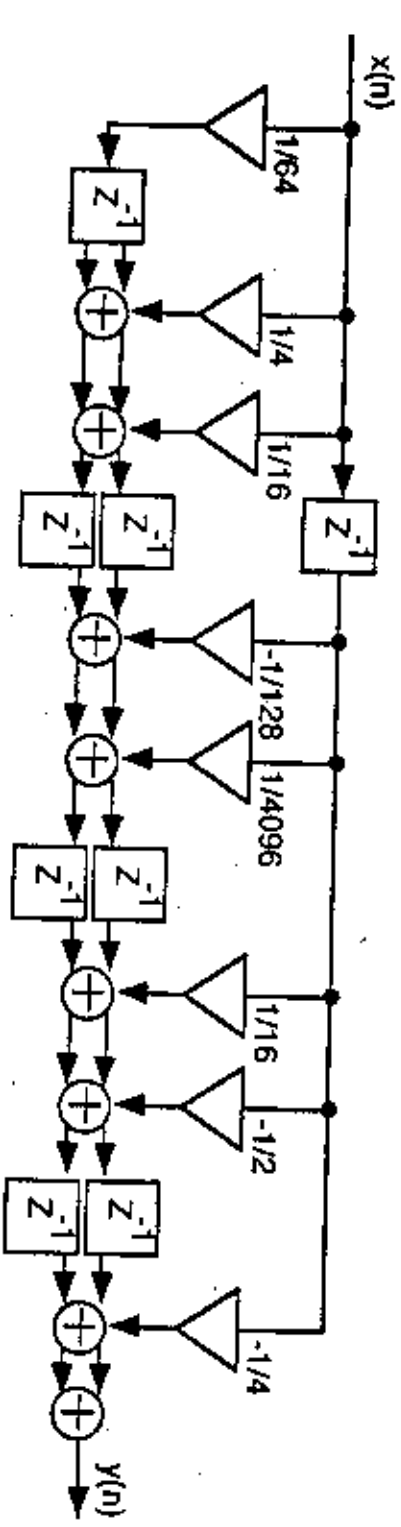
- ==> Transpose Direct Form has pipelining "built-in"
- ==> Critical path contains less adder delays



FIR Filter Implementations



Without Pipelining



With Pipelining