

# *Why Systolic Architecture ?*

VLSI Signal Processing

台灣大學電機系

吳安宇



## Motivation & Introduction

- We need a high-performance , special-purpose computer system to meet specific application.
- I/O and computation imbalance is a notable problem.
- The concept of Systolic architecture can map high-level computation into hardware structures.
- Systolic system works like an automobile assembly line.
- Systolic system is easy to implement because of its regularity and easy to reconfigure.
- Systolic architecture can result in cost-effective , high-performance special-purpose systems for a wide range of problems.



## Key architectural issues in designing special-purpose systems

- **Simple and regular design**

Simple, regular design yields cost-effective special systems.

- **Concurrency and communication**

Design algorithm to support high concurrency and meantime to employ only simple.

- **Balancing computation with I/O**

A special-purpose system should be match a variety of I/O bandwidth.



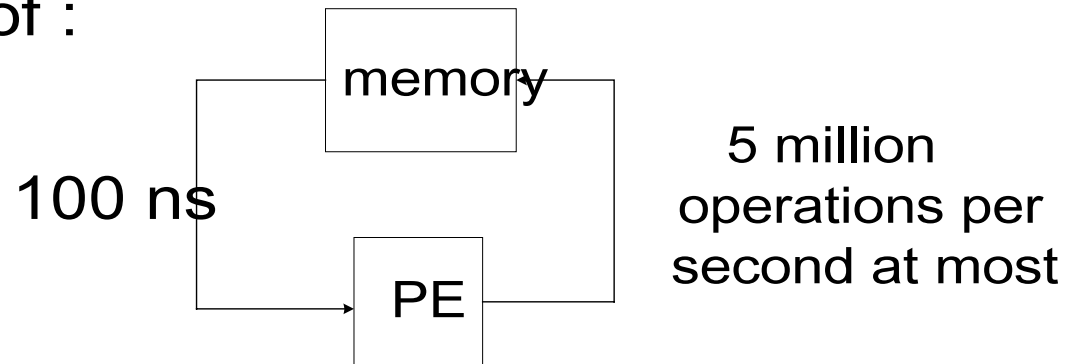
# Basic principle of systolic architecture

- Systolic system consists of a set interconnected cells , each capable of performing some simple operation.
- Systolic approach can speed up a compute-bound computation in a relatively simple and inexpensive manner.
- A systolic array in particular , is illustrated in next page. (we achieve higher computation throughput without increasing memory bandwidth)

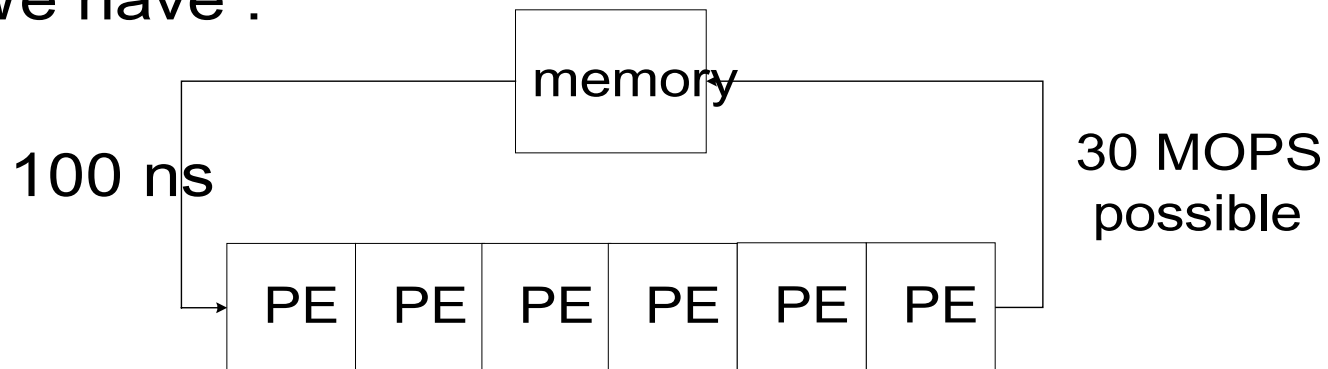


# Basic principle of a systolic system

Instead of :



We have :



The systolic array



## A family of systolic designs for convolution computation

- Given the sequence of weight

$$\{w_1, w_2, \dots, w_k\}$$

- And the input sequence

$$\{x_1, x_2, \dots, x_k\},$$

- Compute the result sequence

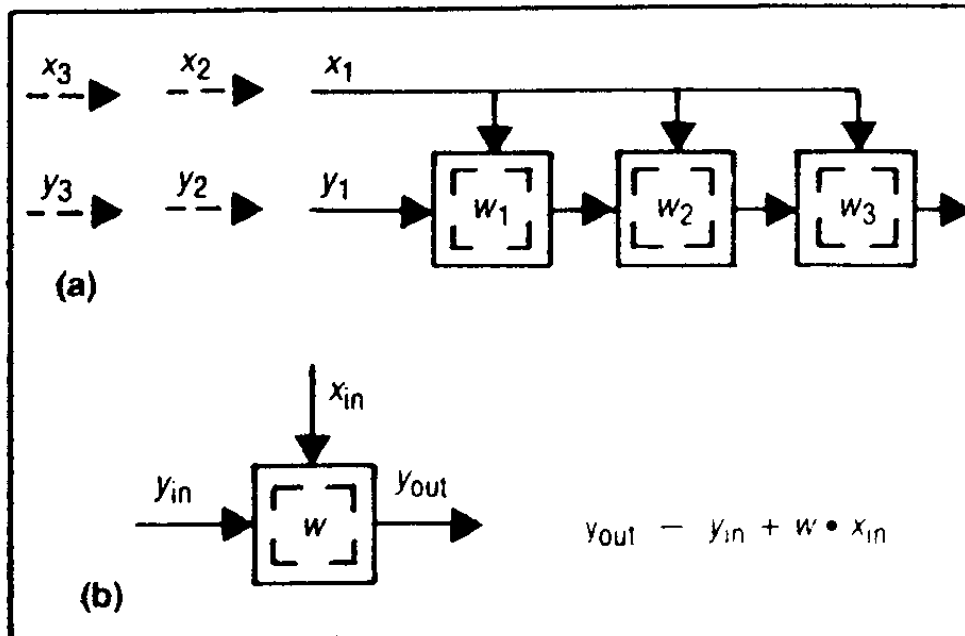
$$\{y_1, y_2, \dots, y_{n+1-k}\}$$

- Defined by

$$y_i = w_1 x_i + w_2 x_{i+1} + \dots + w_k x_{i+k-1}$$



# Design B1

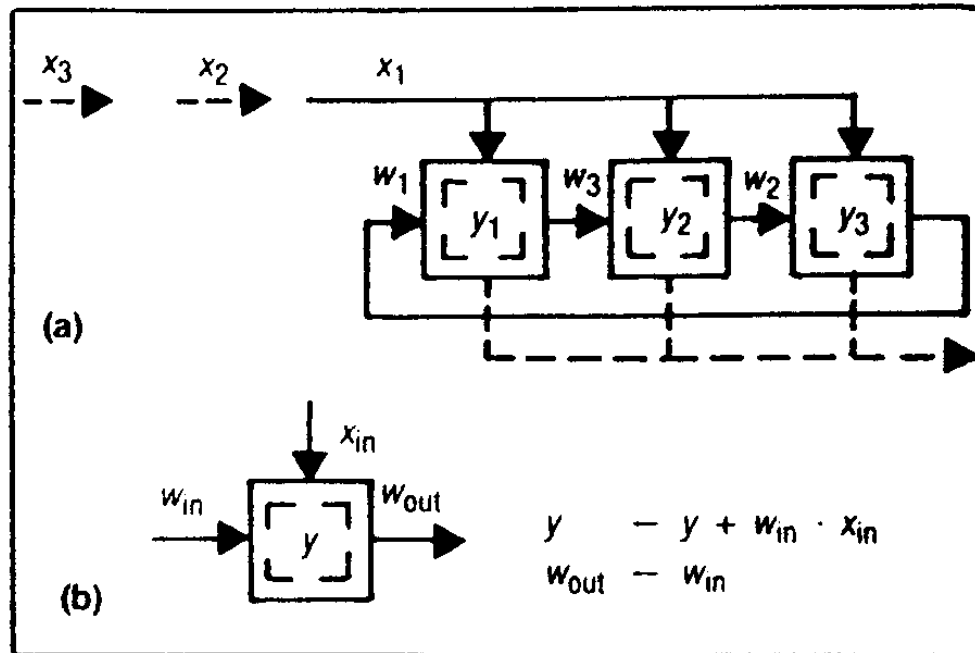


- Previously propose for circuits to implement a pattern matching processor and for circuit to implement polynomial multiplication.

- Broadcast input , move results , weights stay
- (Semi-systolic convolution arrays with global data communication



# Design B2



- The path for moving  $y_i$ 's is wider than  $w_i$ 's because of  $y_i$ 's carry more bits than  $w_i$ 's in numerical accuracy.

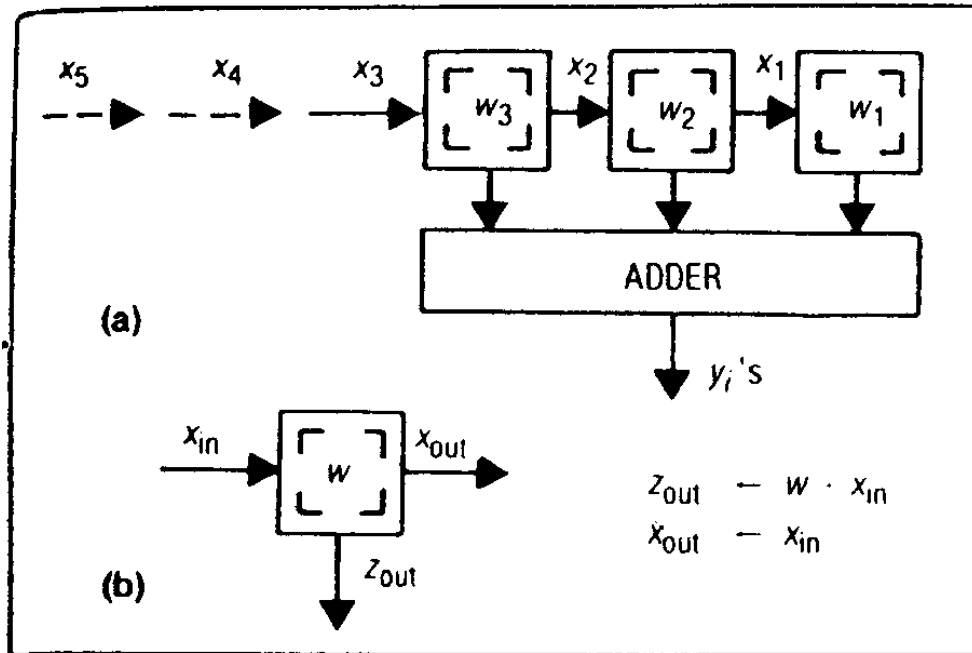
- The use of multiplier-accumulators may also help increase precision of the result, since extra bit can be kept in these accumulators with modest cost.

Broadcast input, move weights, results stay  
 [(Semi-) systolic convolution arrays with  
 global data communication]





# Design F



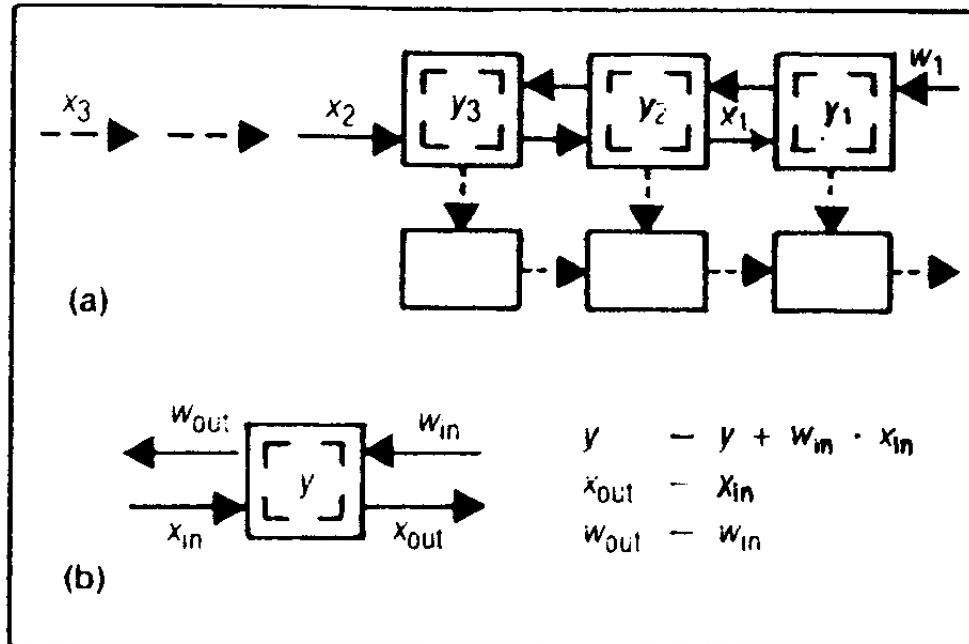
- When number of cell is large , the adder can be implemented as a pipelined adder tree to avoid large delay.

- Design of this type using unbounded fan-in.

- Fan-in results, move inputs, weights stay
- Semi-systolic convolution arrays with global data communication



# Design R1



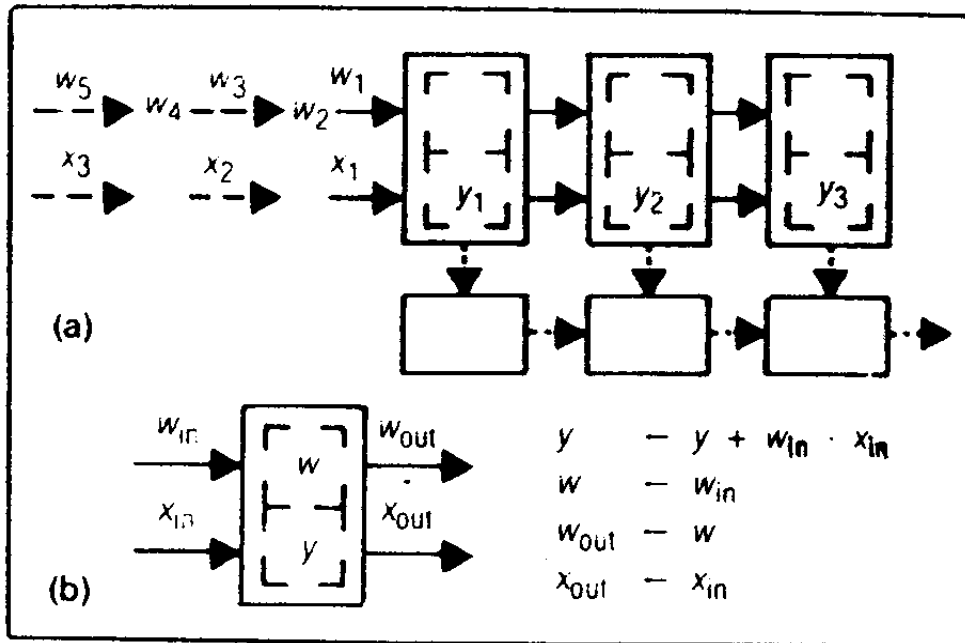
- Design R1 has the advantage that it does not require a bus, or any other global network, for collecting output from cells.

- The basic ideal of this design has been used to implement a pattern matching chip.

- Results stay, inputs and weights move in opposite directions
- Pure-systolic convolution arrays with global data communication



# Design R2



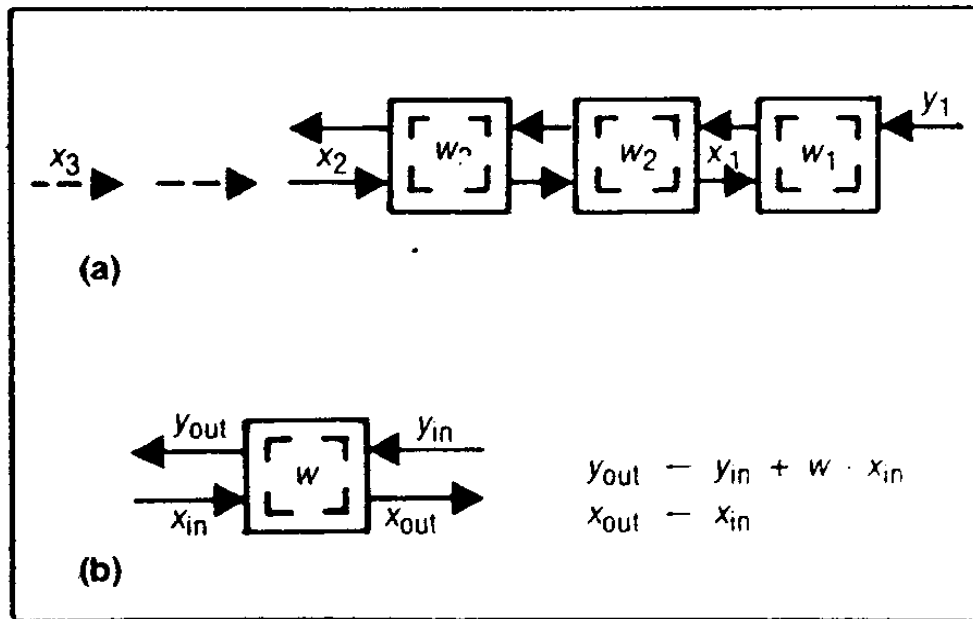
- Multiplier-accumulator can be used effectively and so can tag bit method to signal the output of each cell.

- Compared with R1, all cells work all the time when additional register in each cell to hold a  $w$  value.

- Results stay, inputs and weights move in the same direction but at different speeds
- Pure-systolic convolution arrays with global data communication



# Design W1



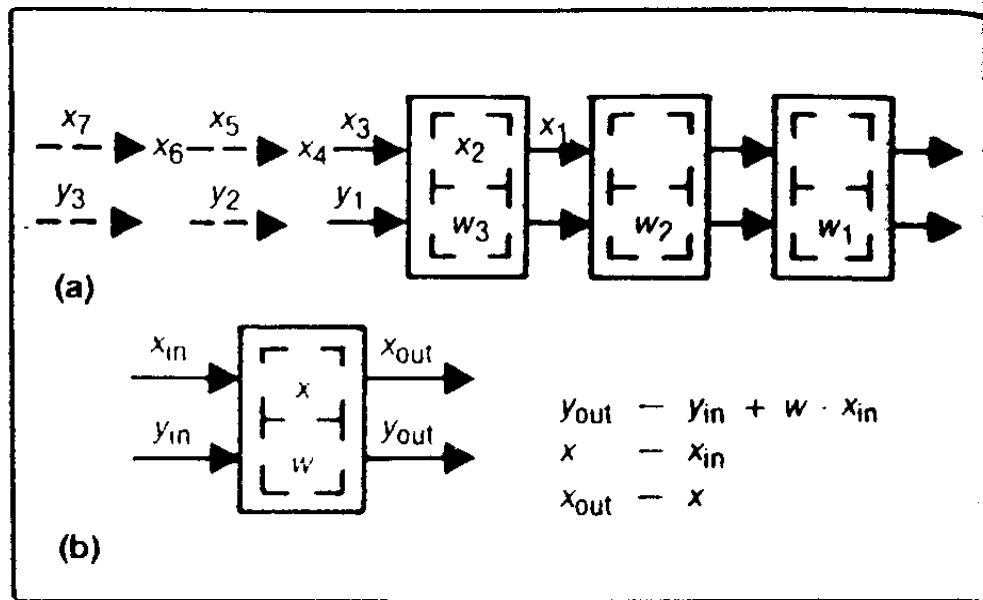
- This design is fundamental in the sense that it can be naturally extended to perform recursive filtering.

- This design suffers the same drawback as R1, only approximately 1/2 cells work at any given time unless two independent computations are interleaved in the same array.

- Weights stay, inputs and results move in opposite direction
- Pure-systolic convolution arrays with global data communication



# Design W2



- This design lose one advantage of W1 , the constant response time.
- This design has been extended to implement 2-D convolution , where high throughputs rather than fast response are of concern.

- Weights stay, inputs and results move in the same direction but at different speeds
- Pure-systolic convolution arrays with global data communication

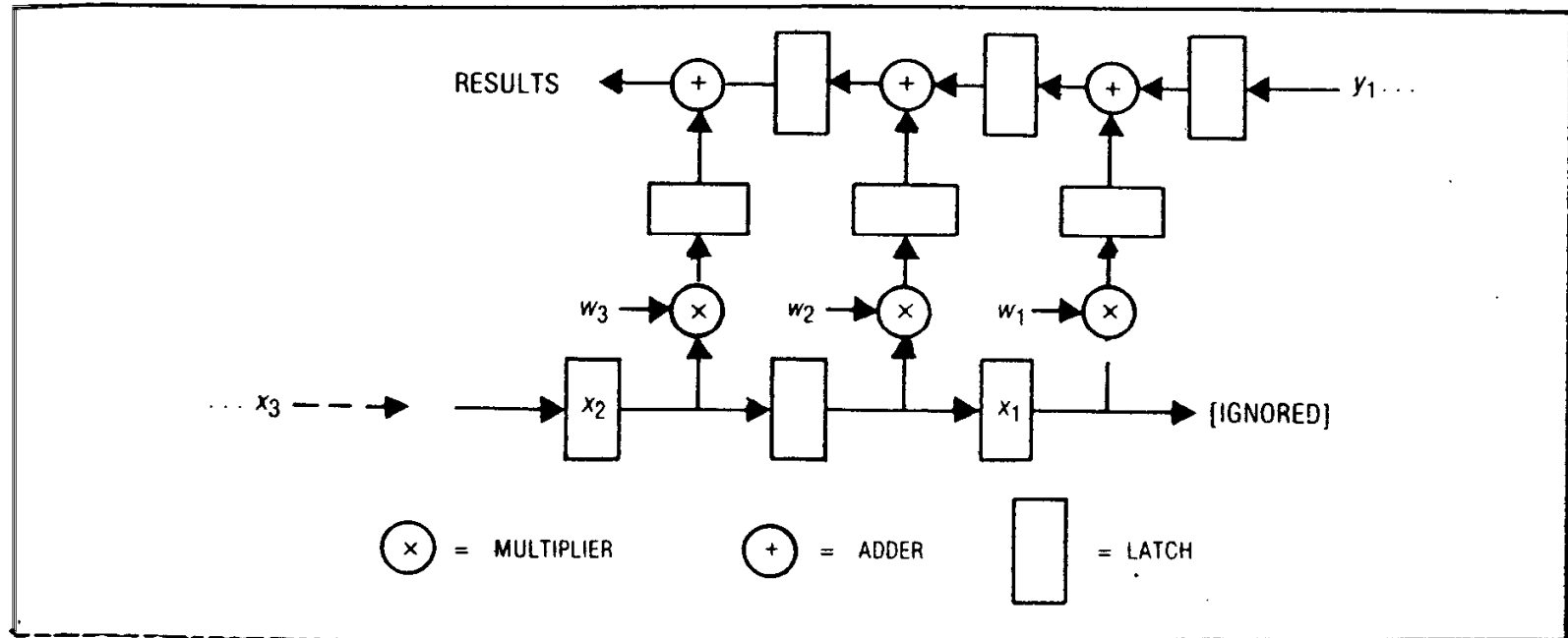


# Remarks

- Above designs are all possible systolic designs for the convolution problem.
- Using a systolic control path , weight can be selected on-the-fly to implement interpolation or adaptive filtering.
- We need to understand precisely the strengths and drawbacks of each design so that an appropriate design can be selected for a given environment.
- For improving throughput, it may be worthwhile to implement multiplier and adder separately to allow overlapping of their execution. (Such as next page show)
- When chip pin is considered , pure-systolic requires four; semi-systolic requires three I/O ports.



# Overlapping the executions of multiply-and-add in design W1





# Criteria and advantages

- The design makes multiple use of each input data item

Because of this property , systolic systems can achieve high throughputs with modest I/O bandwidths for outside communication.

- The design uses extensive concurrency

Concurrency can be obtained by pipelining the stages involved in the computation of each single result , by multiprocessing many results in parallel, or by both.





# Criteria and advantages

- There are only a few types of simple cells

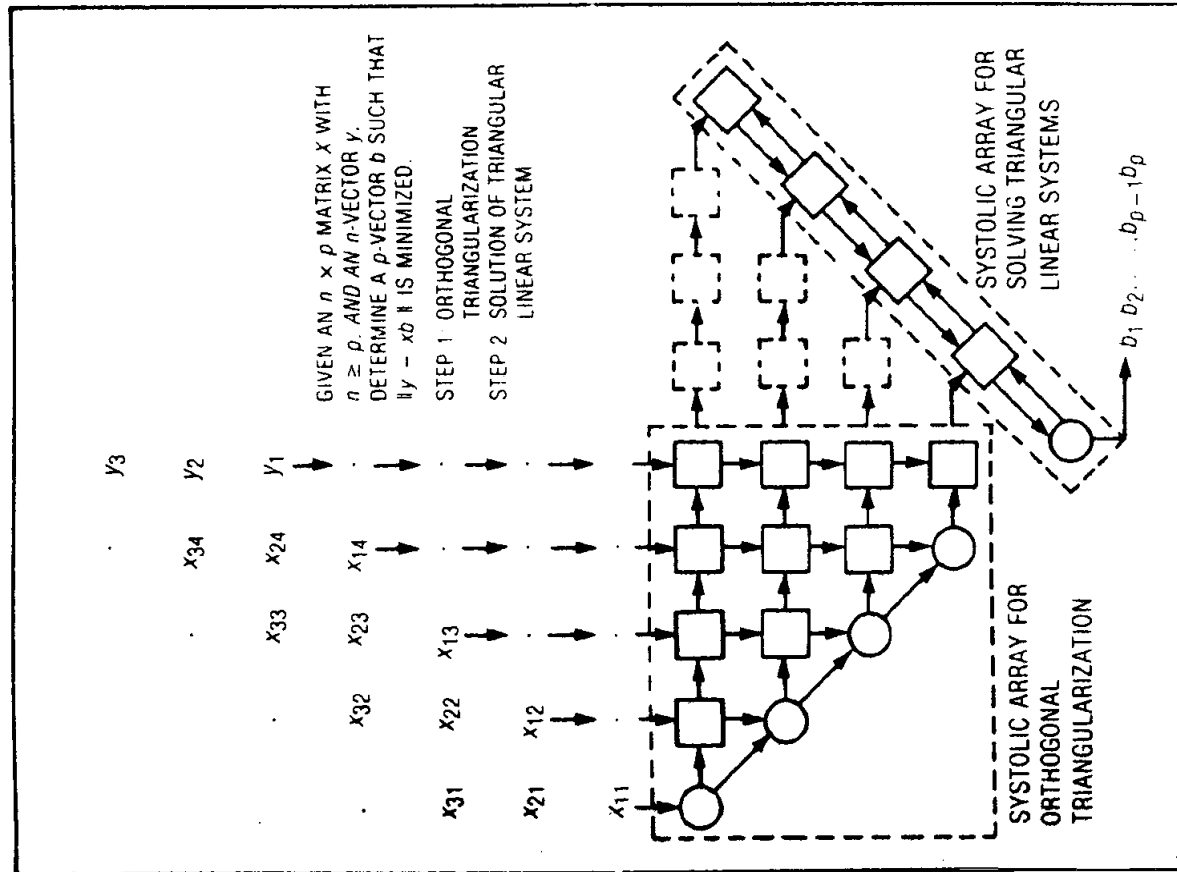
To achieve performance goals, a systolic system is likely to use a large number of cells which must be simple and of only a few types to curtail design and implementation cost.

- Data and control flow are simple and regular

Pure systolic system totally avoid long-distance or irregular wires for data communication.



# On-the-fly least-squares solutions using one and two dimensional systolic array, with $p=4$ .





# Applications of Systolic Array

## - Signal and image processing:

- FTR , IIR filtering , and 1-D convolution.
- 2-D convolution and correlation.
- Discrete Furier transform
- Interpolation
- 1-D and 2-D median filtering
- Geometric warping



# Applications of Systolic Array

## - **Matrix arithmetic:**

- Matrix-vector multiplication
- Matrix-matrix multiplication
- Matrix triangularization  
(solution of linear systems , matrix inversion)
- QR decomposition  
(eigenvalue , least-square computation)
- Solution of triangular linear systems



# Applications of Systolic Array

- **Non-numeric applications:**
  - Data structure
  - Graph algorithm
  - Language recognition
  - Dynamic programming
  - Encoder (polynomial division)
  - Relational data-base operations