Systolic Architecture Design

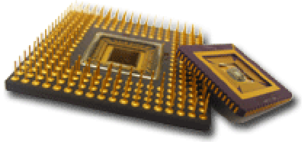
Lan-Da Van (范倫達), *Ph. D.*
Department of Computer Science
National Chiao Tung University
Taiwan, R.O.C.

Fall, 2010

ldvan@cs.nctu.edu.tw
http://www.cs.nctu.tw/~ldvan/

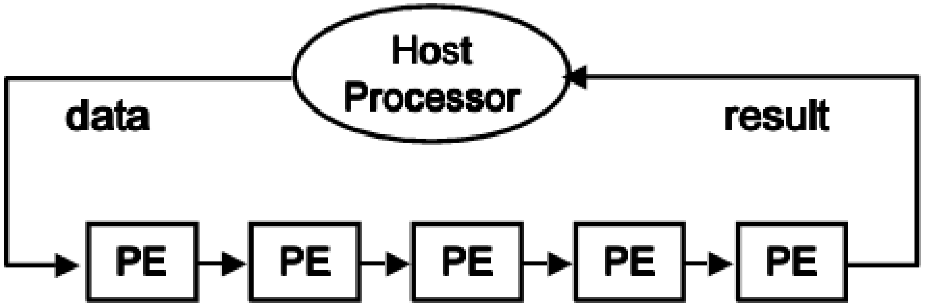
Outline

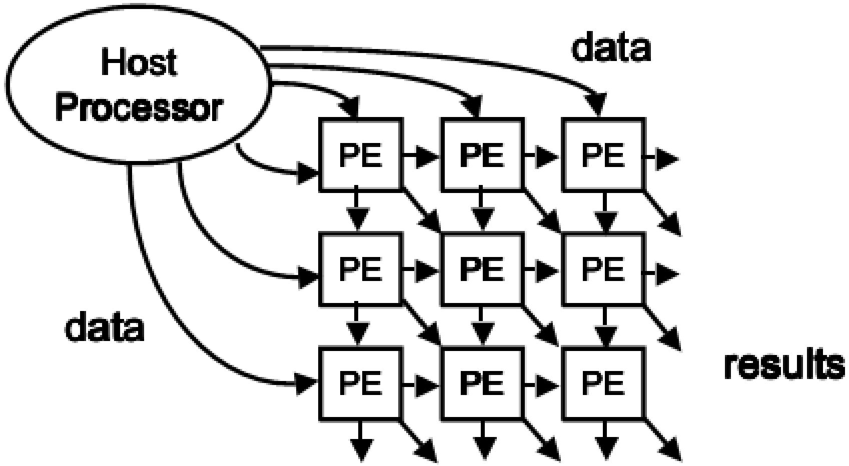
- *Introduction*
- Systolic Array Design Methodology
- FIR Systolic Arrays
- Selection of Scheduling Vector
- Conclusion

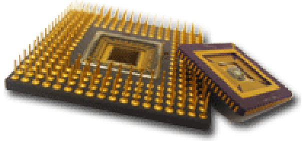


Systolic Architecture

- What is systolic architecture (also called Systolic Arrays)?
- A network of PEs that rhythmically compute and pass data through the system.
- Used as a coprocessor in combination with a host computer and the behavior is analogous to the flow of blood through the heart; thus named as systolic.

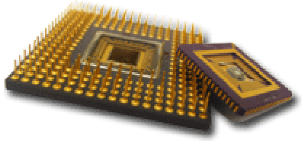






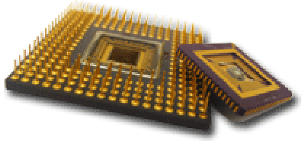
Characteristics of Systolic Arrays

- Synchronization
- Modularity
- Regularity
- Locality
- Finite Connection
- Parallel/Pipeline
- Extendibility
- Some relaxations are introduced to increase the utility of systolic arrays
  - *Neighbor interconnection (near, but not nearest)*
  - *Data broadcast operations*
  - *Different PEs, especially at the boundaries*

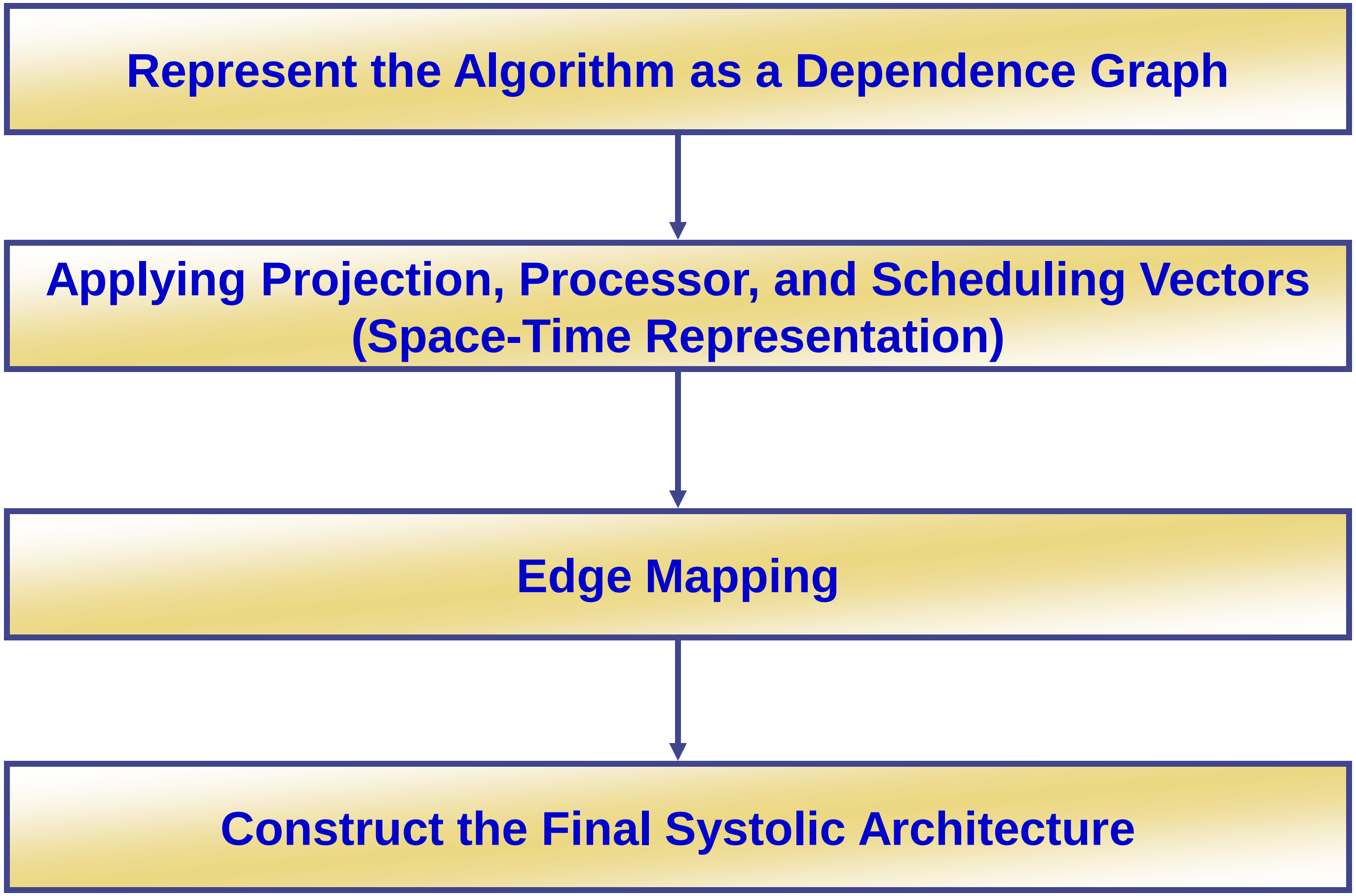


Outline

- Introduction
- *Systolic Array Design Methodology*
- FIR Systolic Arrays
- Selection of Scheduling Vector
- Conclusion



Systolic Array Design Methodology

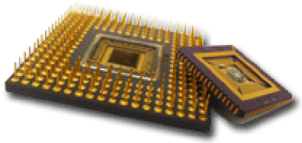


Represent the Algorithm as a Dependence Graph

Applying Projection, Processor, and Scheduling Vectors (Space-Time Representation)

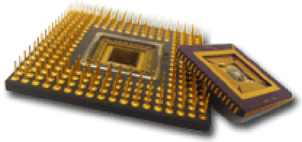
Edge Mapping

Construct the Final Systolic Architecture



Design Methodology: Basic Vectors

- Projection vector $\mathbf{d}^T = [d_1\ d_2]$
  - Determine how DG is compressed.
  - Two nodes that are displaced by **d** or multiples of **d** are executed by the same processor
- Processor space vector $\mathbf{p}^T = [p_1\ p_2]$
  - Any node with index $\mathbf{I}^T = [i, j]$ would be executed by processor $\mathbf{p}^T\mathbf{I}$.
- Schedule vector $\mathbf{s}^T = [s_1\ s_2]$
  - Any node with index $\mathbf{I}^T = [i, j]$ would be executed at **time** $\mathbf{s}^T\mathbf{I}$.
- Hardware utilization efficiency: **HUE** $= 1/|\mathbf{s}^T\mathbf{d}|$
  - This is because two tasks executed by the same processor are spaced $1/|\mathbf{s}^T\mathbf{d}|$ time units apart.
- Feasibility constrains
  - Processor space vector and the projection vector must be orthogonal to each other. **p** is orthogonal to **d**, that is, $\mathbf{p}^T\mathbf{d} = 0$
    - If A and B differ by projection vector, i.e, $\mathbf{I}_A - \mathbf{I}_B = \mathbf{d}$,
      then they must be executed by the same processor => $\mathbf{p}^T\mathbf{I}_A = \mathbf{p}^T\mathbf{I}_B$ => $\mathbf{p}^T(\mathbf{I}_A - \mathbf{I}_B) = 0$ => $\mathbf{p}^T\mathbf{d} = 0$
  - If A and B are mapped to the same processor, then they cannot be executed at the same time, i.e, $\mathbf{s}^T\mathbf{I}_A \neq \mathbf{s}^T\mathbf{I}_B$ => $\mathbf{s}^T\mathbf{d} \neq 0$
  - Edge mapping: If an edge **e** exists in DG, then an edge $\mathbf{p}^T\mathbf{e}$ exists in the systolic array with $\mathbf{s}^T\mathbf{e}$ delays.

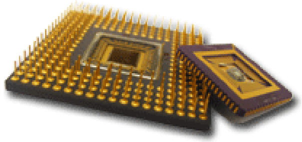


Space to Space-Time Representation

- Space-time representation
  - Interpreting one of the spatial dimensions as temporal dimension
  - j’: processor axis, t’: scheduling time instance

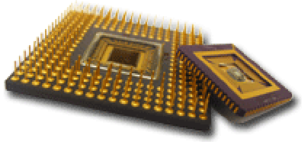
$$\begin{pmatrix} i' \\ j' \\ t' \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ & p^T & 0 \\ & s^T & 0 \end{pmatrix} \begin{pmatrix} i \\ j \\ t \end{pmatrix}$$

$$i' = t, \quad j' = \mathbf{p}^T I, \quad t' = \mathbf{s}^T I$$

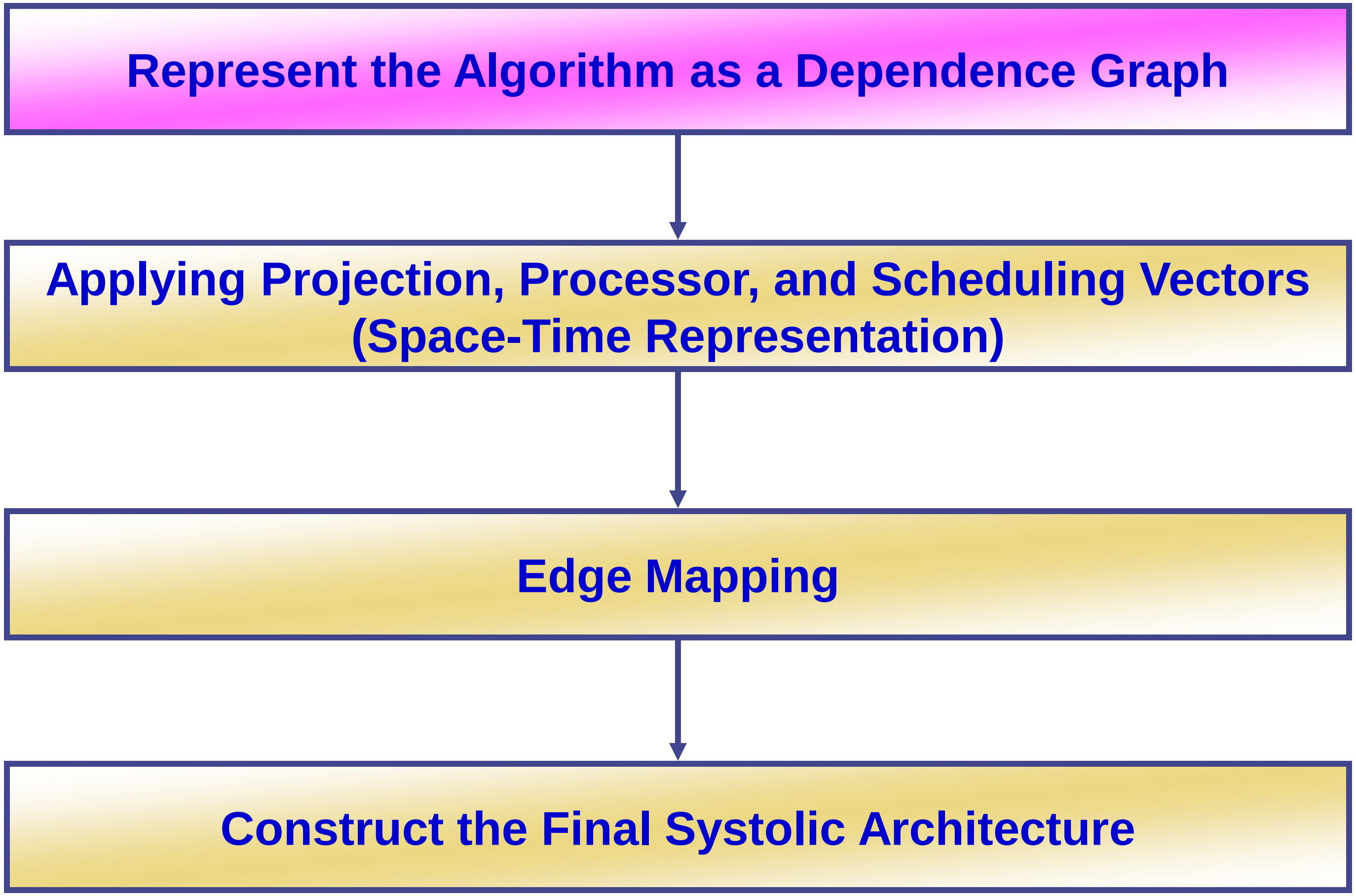


Outline

- Introduction
- Systolic Array Design Methodology
- *FIR Systolic Arrays*
- Selection of Scheduling Vector
- Matrix-Matrix Multiplication and 2D Systolic Array Design
- Systolic Design for Space Representations Containing Delays
- Conclusion



Systolic Array Design Methodology

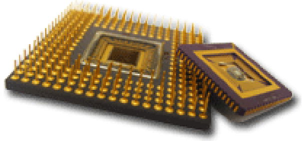


Represent the Algorithm as a Dependence Graph

Applying Projection, Processor, and Scheduling Vectors (Space-Time Representation)

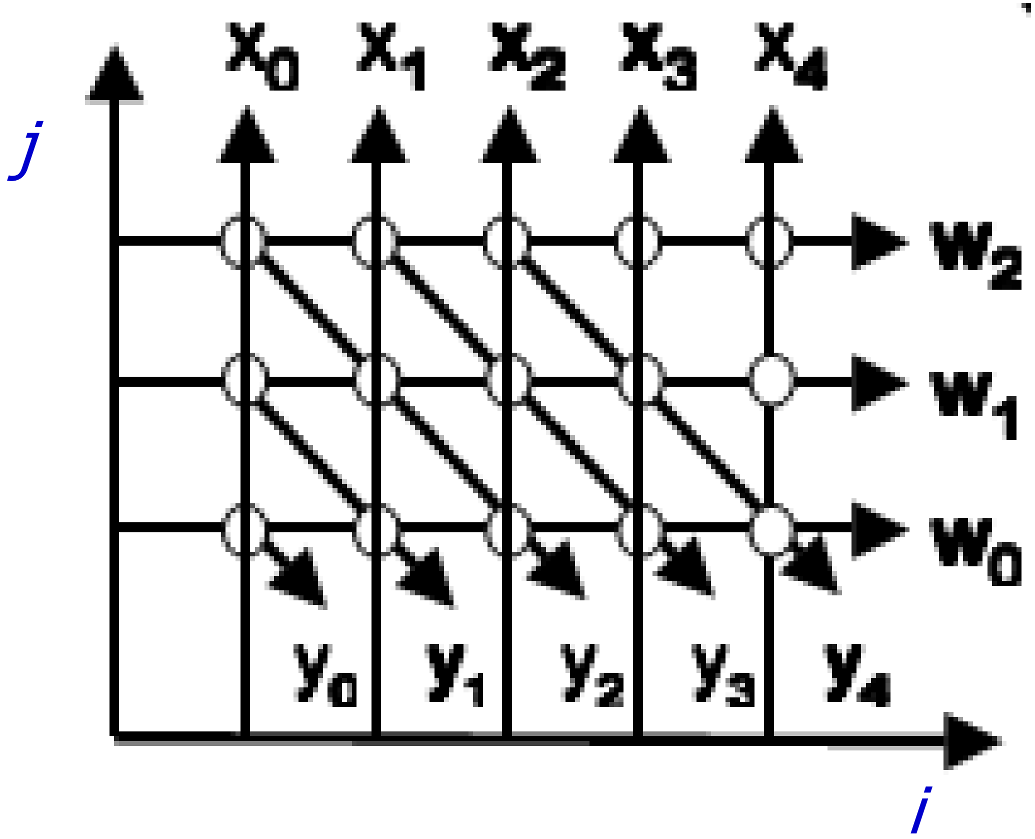
Edge Mapping

Construct the Final Systolic Architecture

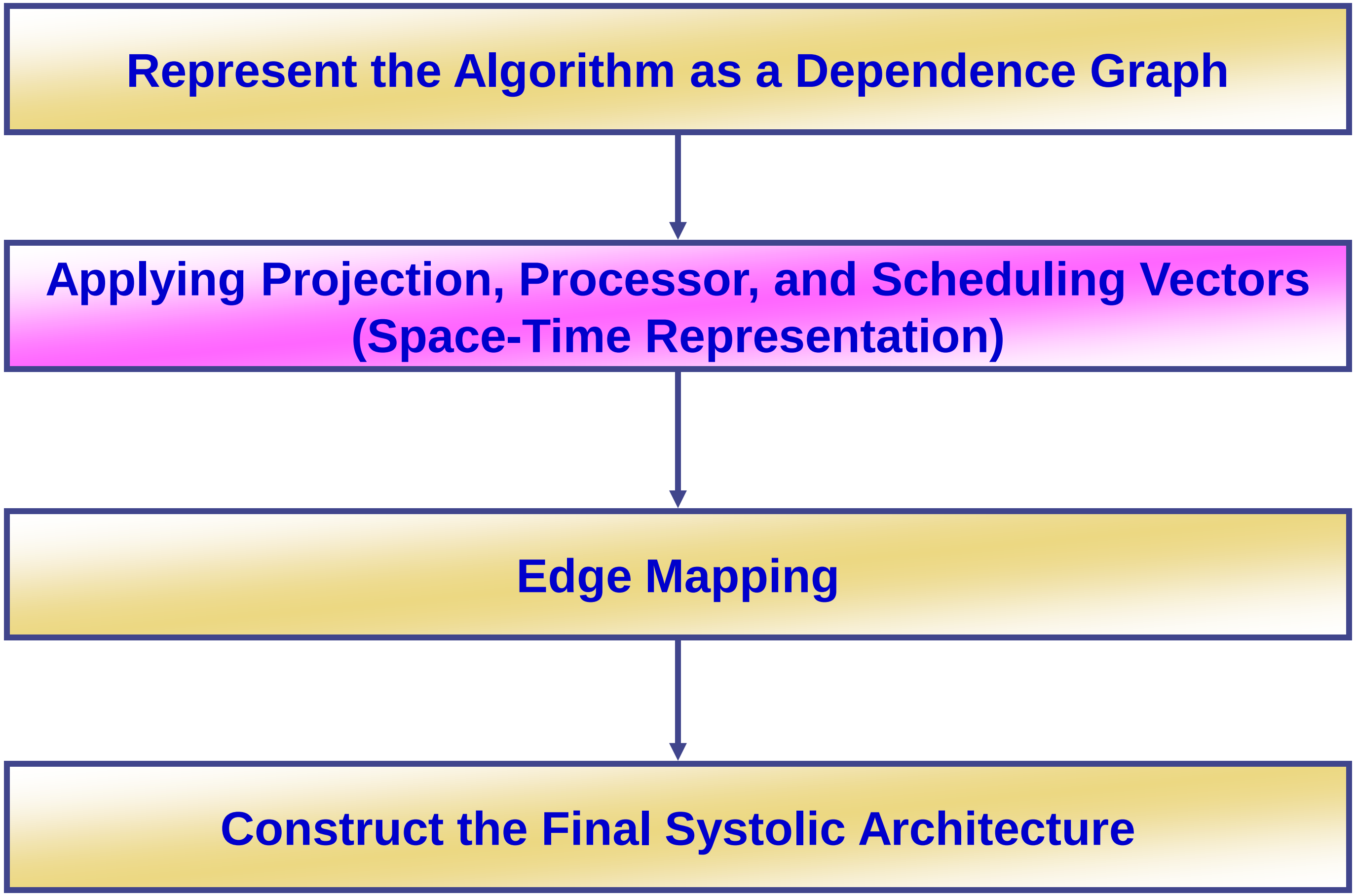


DG of FIR Filter

- Dependence Graph (DG)
  - Ex: FIR filter: $y(n) = w_0(n)x(n)+w_1x(n-1)+w_2x(n-2)$



Systolic Array Design Methodology

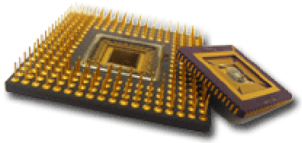


Represent the Algorithm as a Dependence Graph

Applying Projection, Processor, and Scheduling Vectors (Space-Time Representation)

Edge Mapping

Construct the Final Systolic Architecture

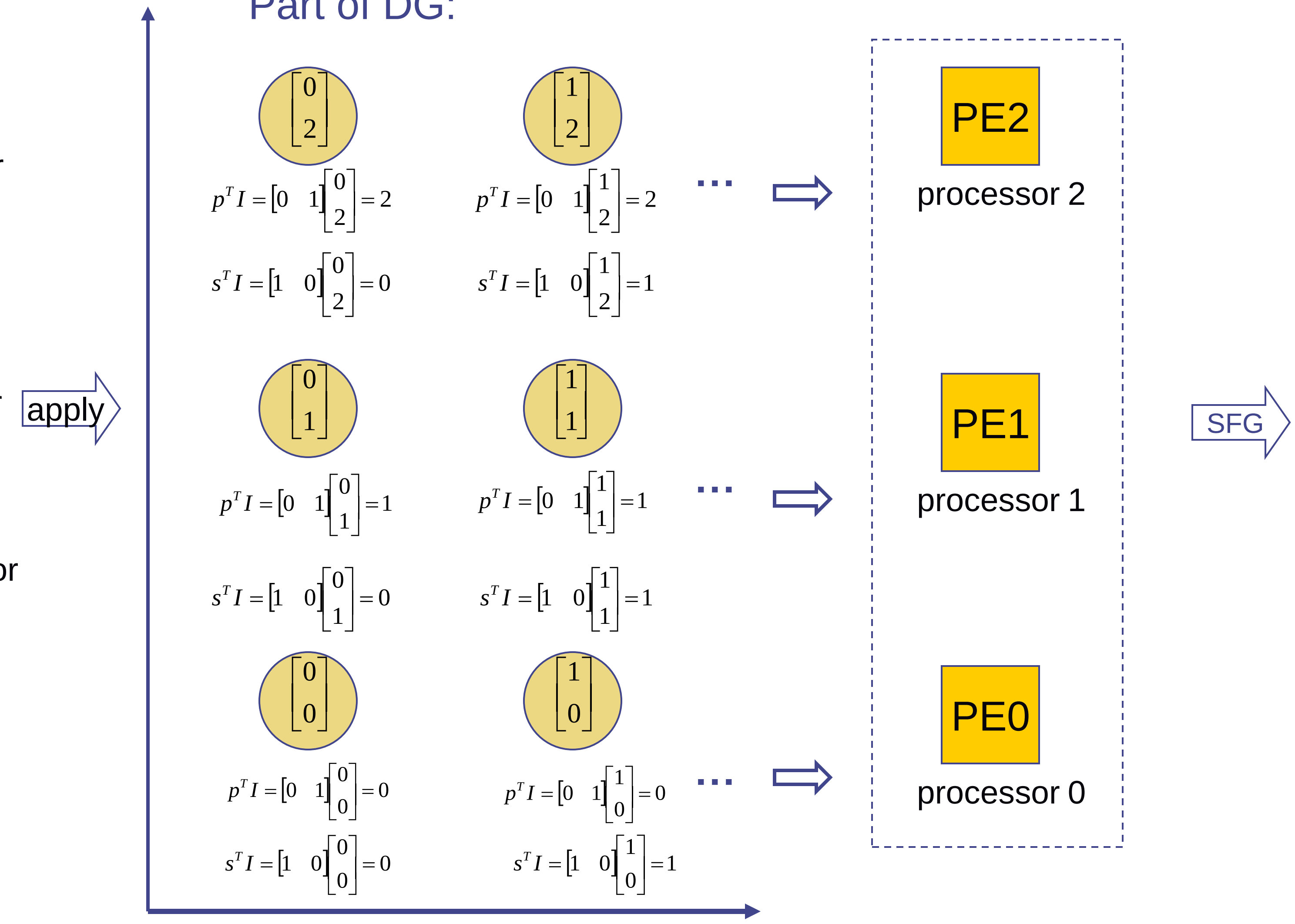


Applying Projection and Scheduling (1/2)

Processor vector
$\mathbf{p}^T = [0\ 1]$

Projection vector
$\mathbf{d}^T = [1\ 0]$

Scheduling vector
$\mathbf{s}^T = [1\ 0]$



$\begin{bmatrix}0\\2\end{bmatrix}$ $\quad p^T I = [0\ \ 1]\begin{bmatrix}0\\2\end{bmatrix} = 2 \quad s^T I = [1\ \ 0]\begin{bmatrix}0\\2\end{bmatrix} = 0$

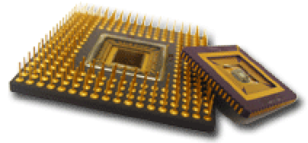
$\begin{bmatrix}1\\2\end{bmatrix}$ $\quad p^T I = [0\ \ 1]\begin{bmatrix}1\\2\end{bmatrix} = 2 \quad s^T I = [1\ \ 0]\begin{bmatrix}1\\2\end{bmatrix} = 1$ … ⇒ PE2 processor 2

$\begin{bmatrix}0\\1\end{bmatrix}$ $\quad p^T I = [0\ \ 1]\begin{bmatrix}0\\1\end{bmatrix} = 1 \quad s^T I = [1\ \ 0]\begin{bmatrix}0\\1\end{bmatrix} = 0$

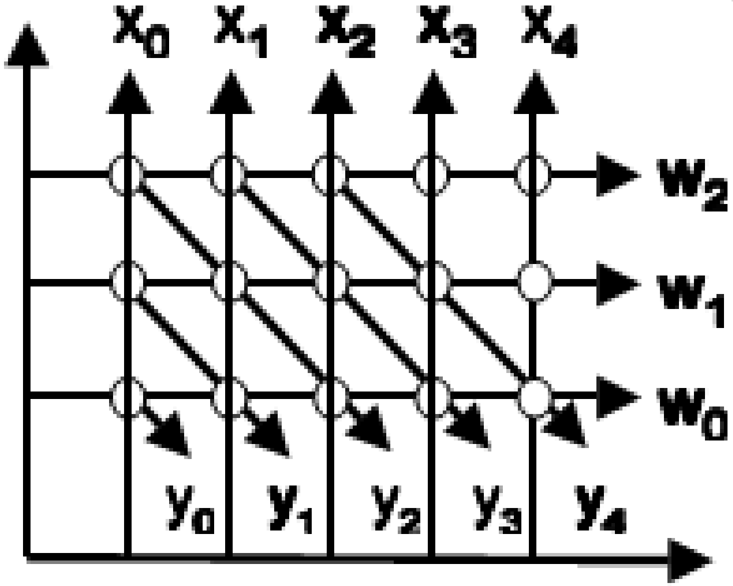
$\begin{bmatrix}1\\1\end{bmatrix}$ $\quad p^T I = [0\ \ 1]\begin{bmatrix}1\\1\end{bmatrix} = 1 \quad s^T I = [1\ \ 0]\begin{bmatrix}1\\1\end{bmatrix} = 1$ … ⇒ PE1 processor 1

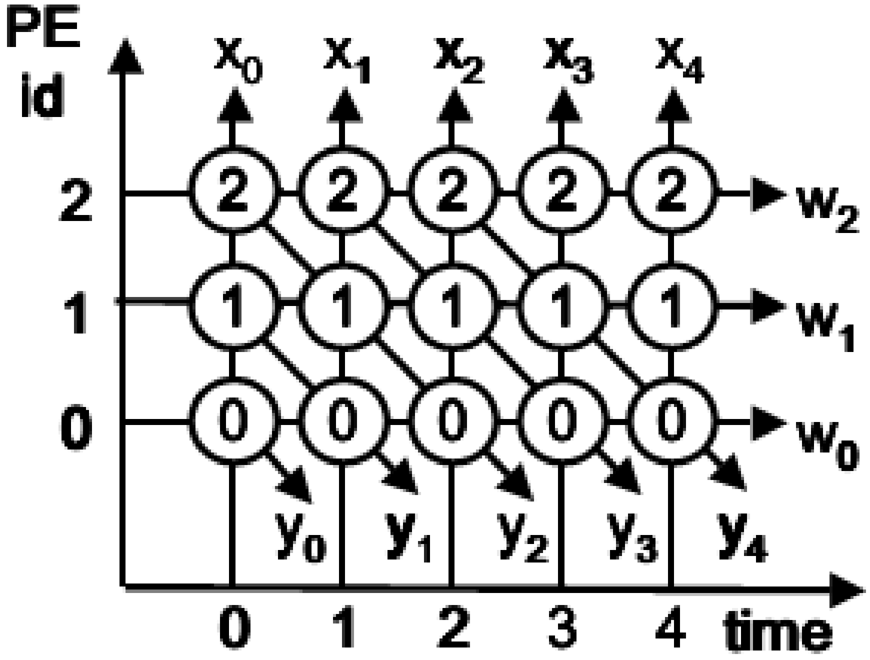
$\begin{bmatrix}0\\0\end{bmatrix}$ $\quad p^T I = [0\ \ 1]\begin{bmatrix}0\\0\end{bmatrix} = 0 \quad s^T I = [1\ \ 0]\begin{bmatrix}0\\0\end{bmatrix} = 0$

$\begin{bmatrix}1\\0\end{bmatrix}$ $\quad p^T I = [0\ \ 1]\begin{bmatrix}1\\0\end{bmatrix} = 0 \quad s^T I = [1\ \ 0]\begin{bmatrix}1\\0\end{bmatrix} = 1$ … ⇒ PE0 processor 0



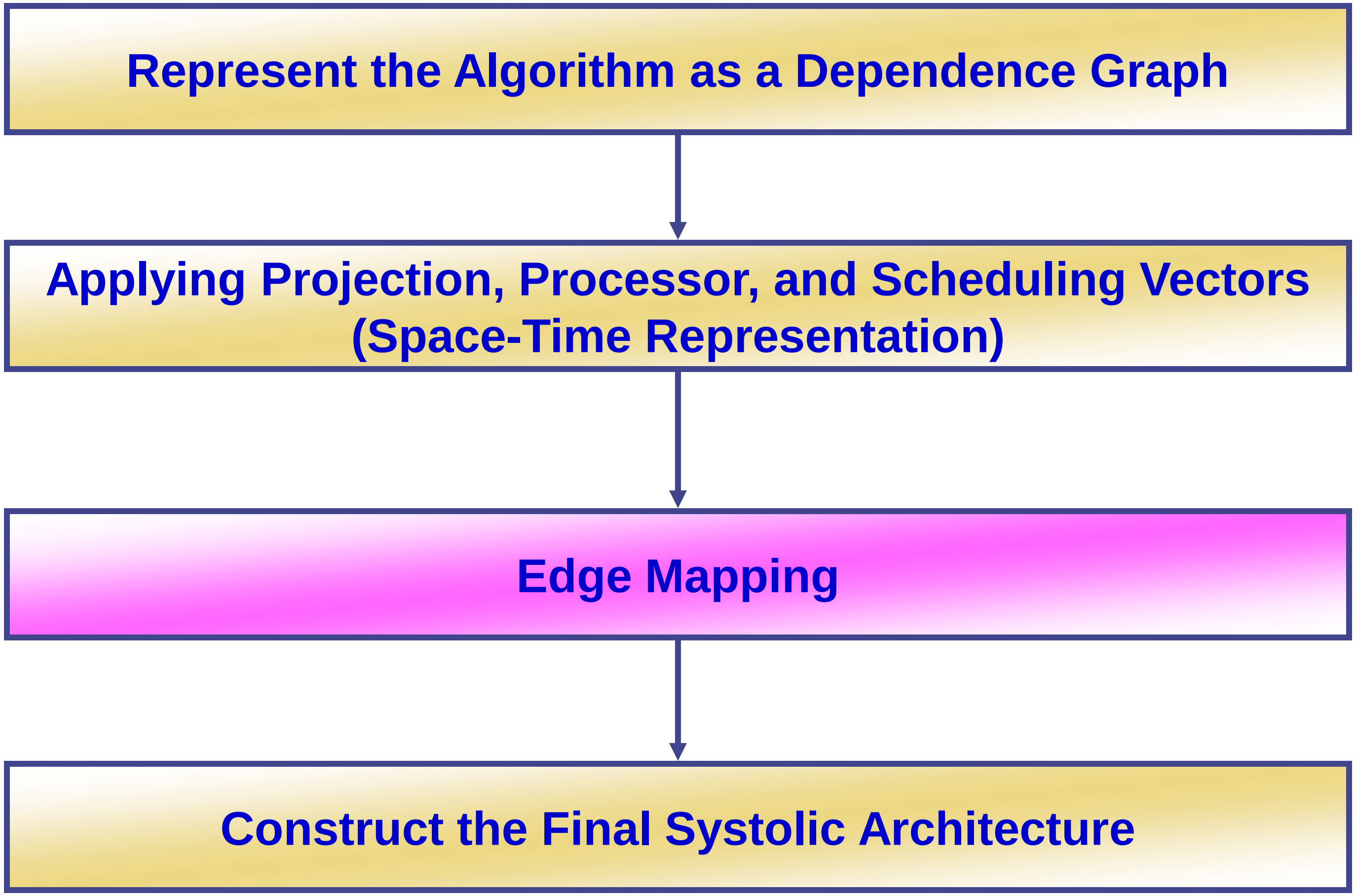
Applying Projection and Scheduling (2/2)





Dependence Graph → Applying projection and Scheduling → **Space-time representation**

Systolic Array Design Methodology

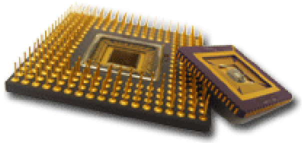


Represent the Algorithm as a Dependence Graph

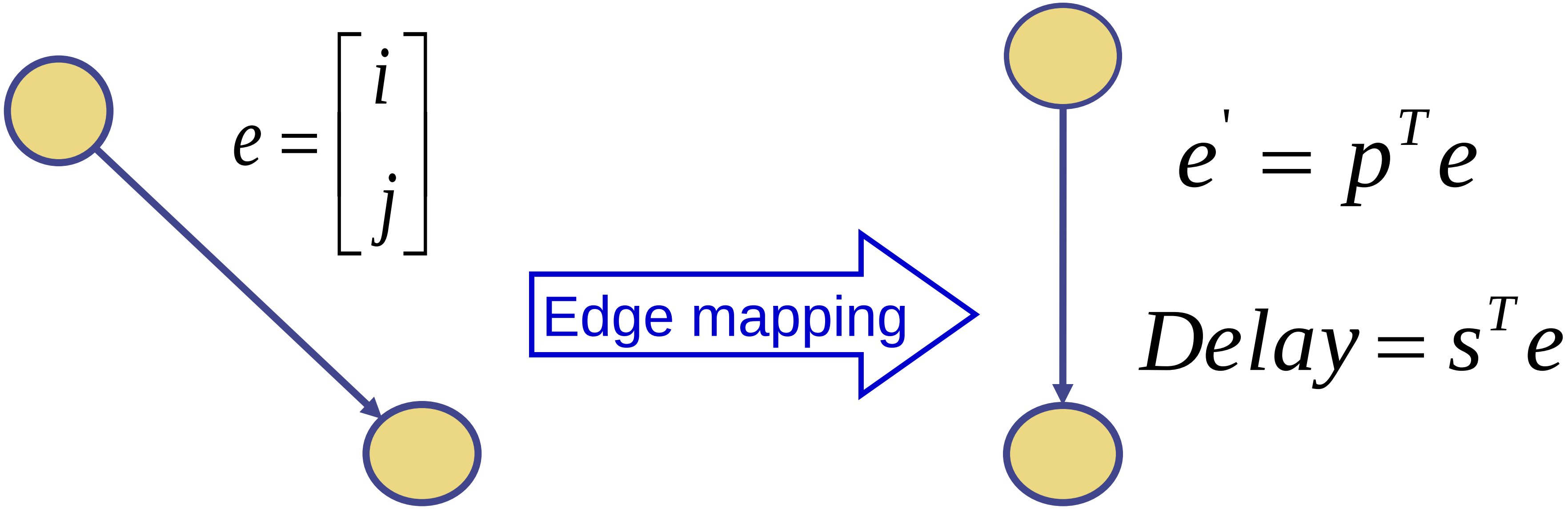
Applying Projection, Processor, and Scheduling Vectors (Space-Time Representation)

Edge Mapping

Construct the Final Systolic Architecture



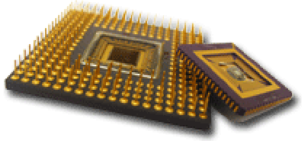
Edge Mapping



$$e = \begin{bmatrix} i \\ j \end{bmatrix}$$

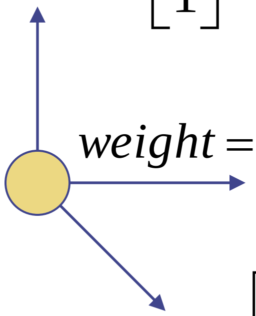
$$e' = p^T e$$

$$Delay = s^T e$$



Edge Mapping

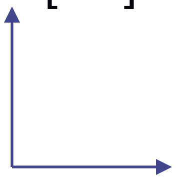
Example:



$$input = e_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

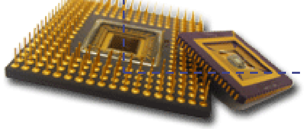
$$weight = e_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

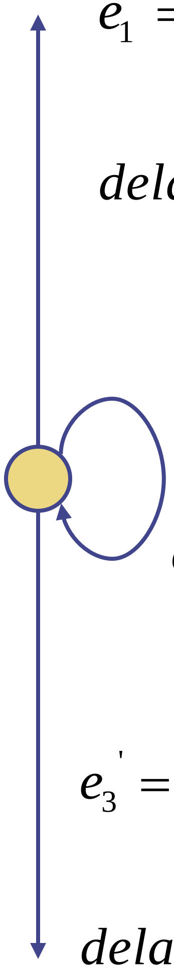
$$output = e_3 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$



p^T=[0 1]

s^T=[1 0]
d^T=[1 0]





$$e_1^{'} = p^T e_1 = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 1$$

$$delay_{e1} = s^T e_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0$$

$$e_2^{'} = p^T e_2 = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 0$$

$$delay_{e2} = s^T e_2 = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 1$$

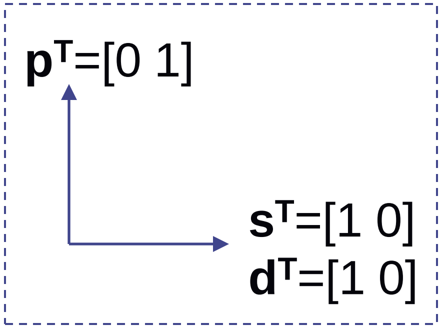
$$e_3^{'} = p^T e_3 = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = -1$$

$$delay_{e3} = s^T e_3 = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = 1$$

Edge mapping

e	$\mathbf{p}^T\mathbf{e}$	$\mathbf{s}^T\mathbf{e}$
Input $[0\ 1]^T$	1	0
Weight $[1\ 0]^T$	0	1
Output $[1\ -1]^T$	-1	1

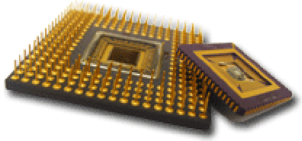
Edge mapping table



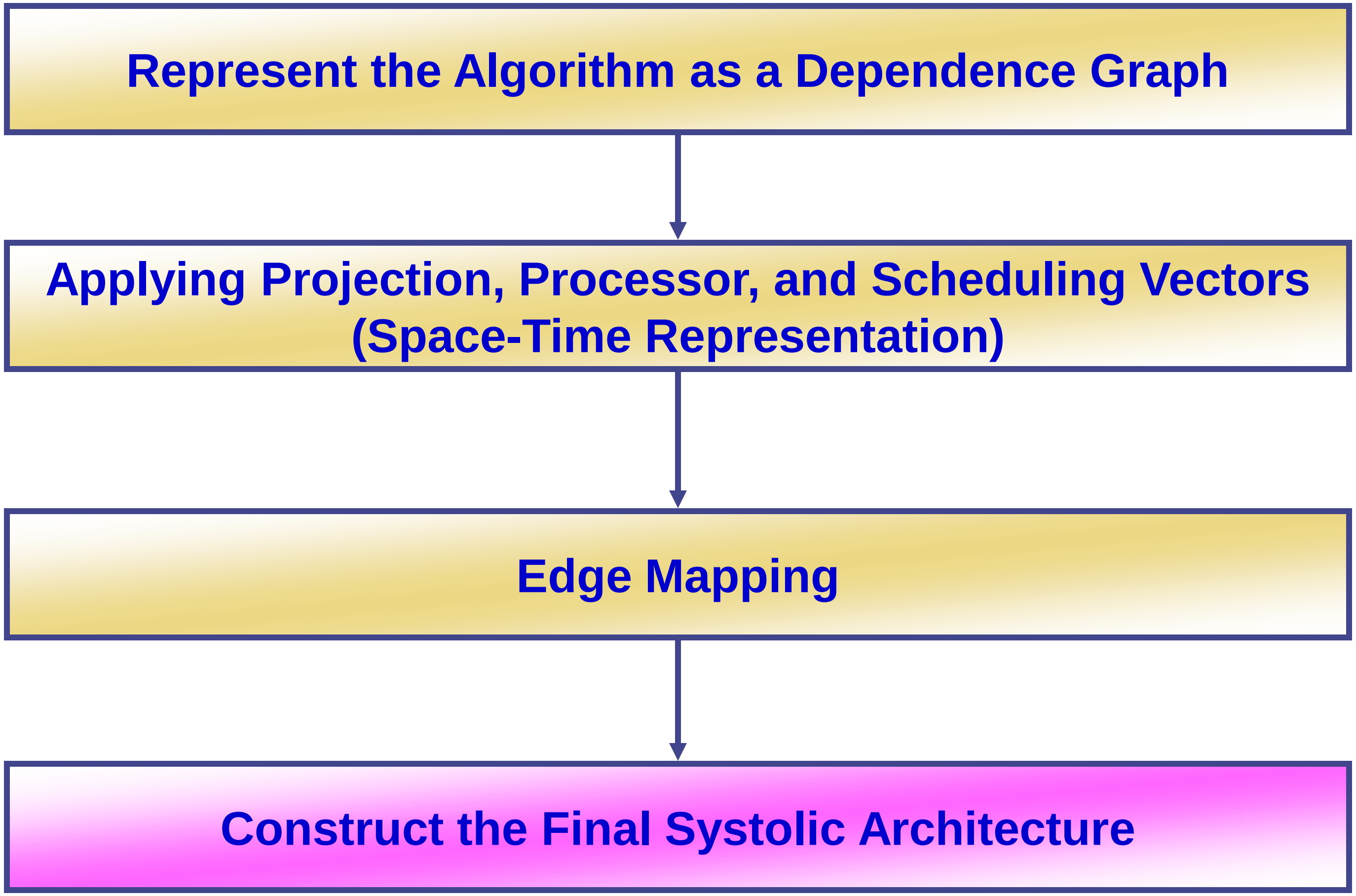
$\mathbf{p}^T=[0\ 1]$

$\mathbf{s}^T=[1\ 0]$

$\mathbf{d}^T=[1\ 0]$



Systolic Array Design Methodology

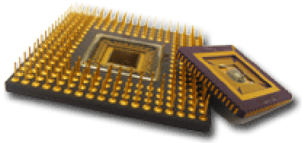


Represent the Algorithm as a Dependence Graph

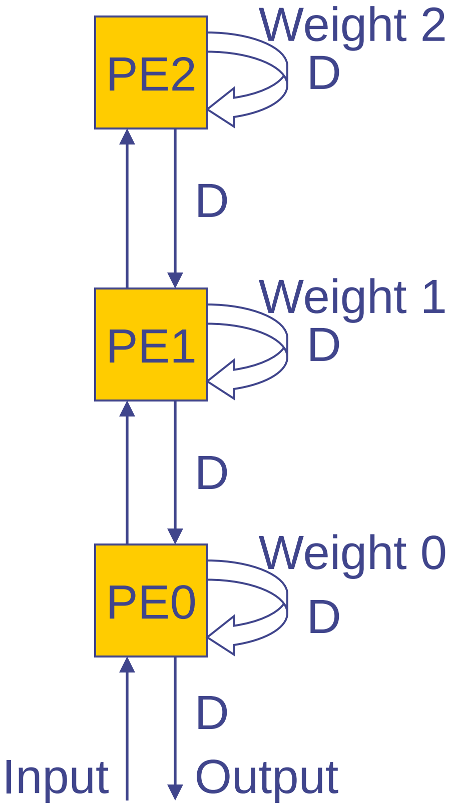
Applying Projection, Processor, and Scheduling Vectors
(Space-Time Representation)

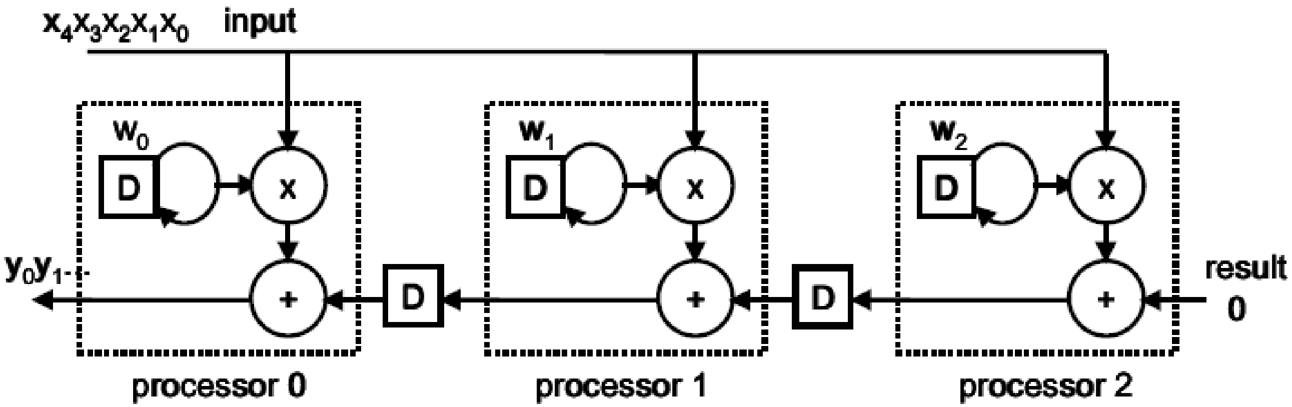
Edge Mapping

Construct the Final Systolic Architecture

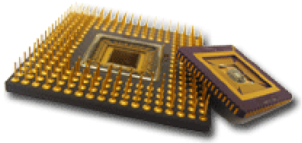


Construct the Final Systolic Architecture



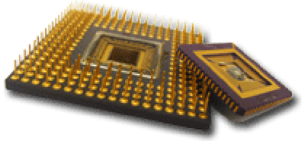


This is called B1 design



Alternative Designs

- B_1 (Broadcast inputs, Move results, Weight Stay)
- B_2 (Broadcast inputs, Move Weight, Results stay)
- F (Fan-in results, Move inputs, Weight stay)
- R_1 (Results stay, Inputs and Weight move in opposite directions)
- R_2 and Dual R_2 (Results stay, Inputs and Weights move in the same direction but at different speeds)
- W_1 (Weights stay, Inputs and Results move in opposite directions)
- W_2 and Dual W_2 (Weights stay, Inputs and Results move in same direction but at different speeds)
- Relating systolic designs using transformations

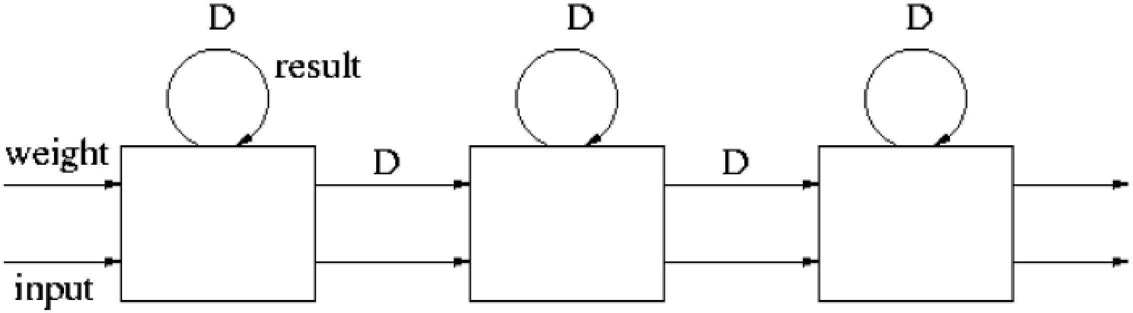


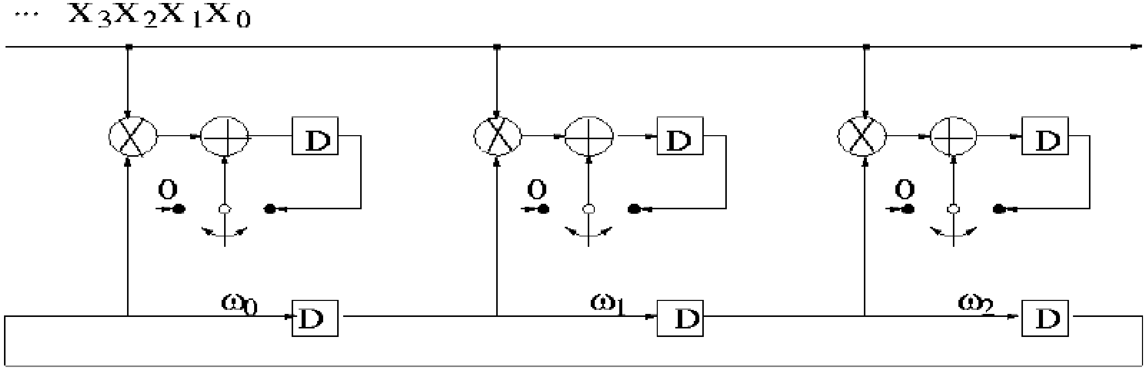
B_2 – Broadcast Inputs, Move Weight, Results Stay

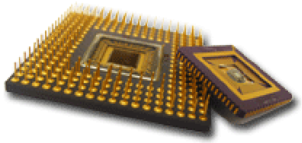
d^T=[1 -1]
p^T=[1 1]
s^T=[1 0]

e	p^Te	s^Te
wt $[1\ 0]^T$	1	1
input $[0\ 1]^T$	1	0
result $[1\ -1]^T$	0	1

$$HUE = \frac{1}{\left|s^T d\right|} = 1$$





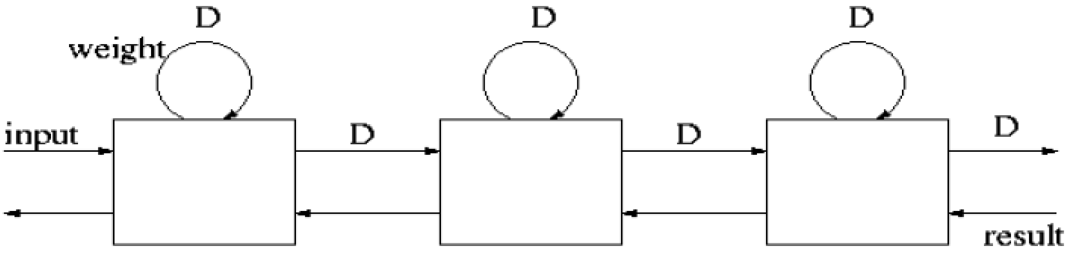


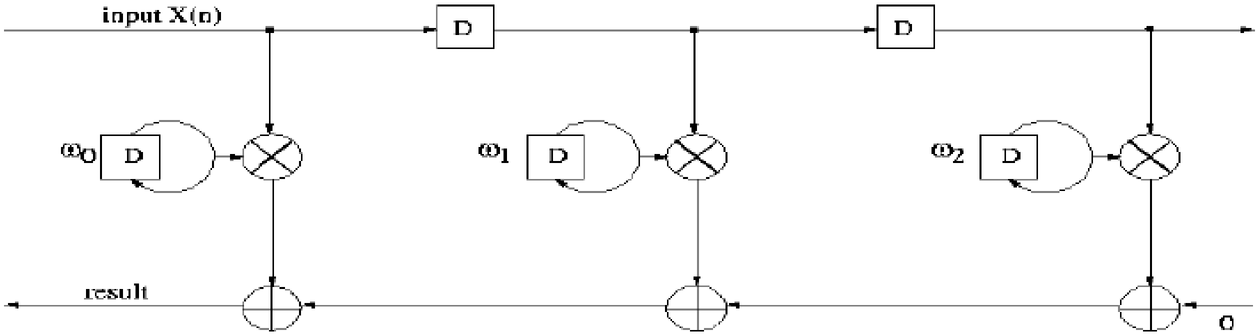
F - Fan-in Results, Move Inputs, Weight Stay

d^T=[1 0]
p^T=[0 1]
s^T=[1 1]

e	p^{T}e	s^{T}e
wt [1 0]^T	0	1
input [0 1]^T	1	1
result [1 -1]^T	-1	0

$$HUE = \frac{1}{\left|S^T d\right|} = 1$$



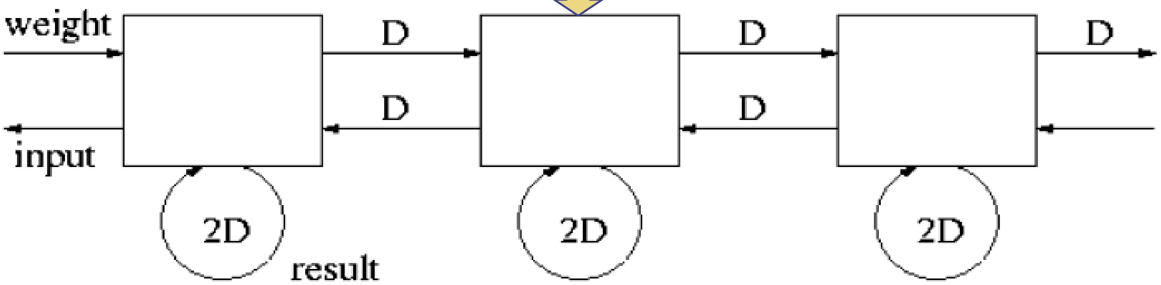


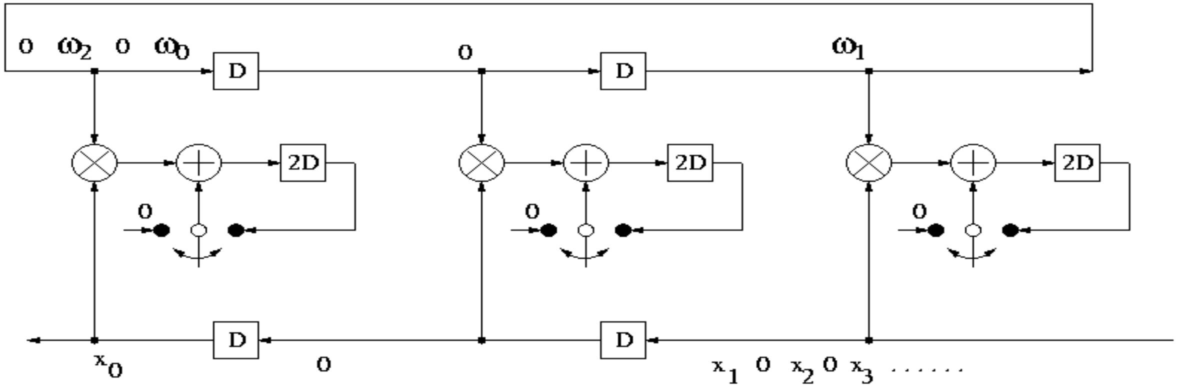
R_1 - Results Stay, Inputs and Weight Move in Opposite Directions

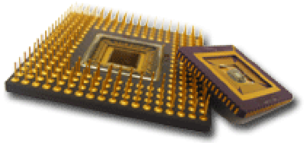
$\mathbf{d}^T$=[1 -1]
$\mathbf{p}^T$=[1 1]
$\mathbf{s}^T$=[1 -1]

e	$\mathbf{p}^T\mathbf{e}$	$\mathbf{s}^T\mathbf{e}$
wt $[1\ 0]^T$	1	1
input $[0\ 1]^T$	-1	1
result $[1\ -1]^T$	0	2

$$HUE = \frac{1}{\left|s^T d\right|} = \frac{1}{2}$$







R_2 and Dual R_2-Results Stay, Inputs and Weights Move in the Same Direction but at Different Speeds

R_2

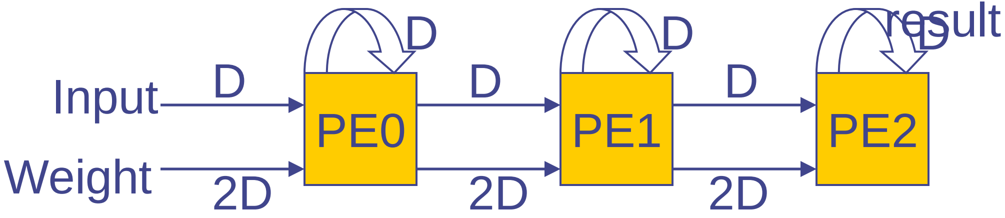
$\mathbf{d}^T=[1\ -1]$

$\mathbf{p}^T=[1\ 1]$

$\mathbf{s}^T=[2\ 1]$

e	p^Te	s^Te
wt $[1\ 0]^T$	1	2
input $[0\ 1]^T$	1	1
result $[1\ -1]^T$	0	1

$$HUE=\frac{1}{\left|s^T d\right|}=1$$



Dual R_2

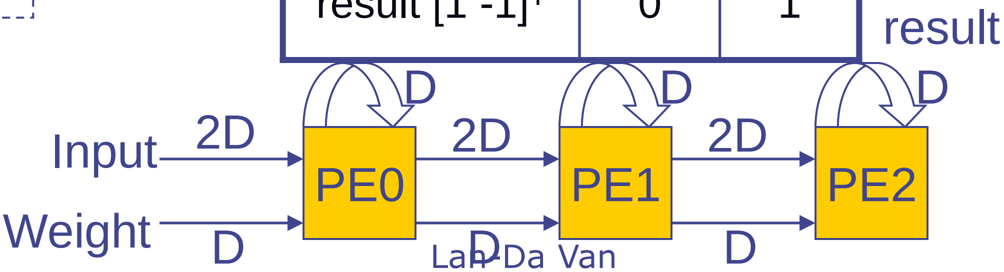
$\mathbf{d}^T=[1\ -1]$

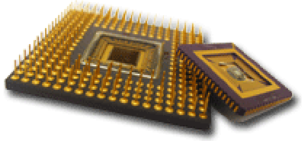
$\mathbf{p}^T=[1\ 1]$

$\mathbf{s}^T=[1\ 2]$

e	p^Te	s^Te
wt $[1\ 0]^T$	1	1
input $[0\ 1]^T$	1	2
result $[1\ -1]^T$	0	1

$$HUE=\frac{1}{\left|s^T d\right|}=1$$



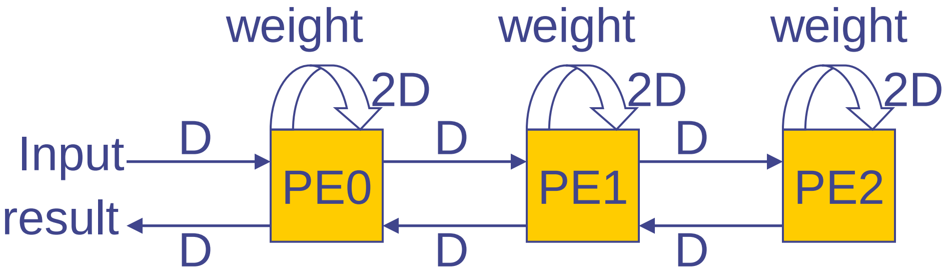


W_1 – Weights Stay, Inputs and Results Move in Opposite Directions

d^T=[1 0]
p^T=[0 1]
s^T=[2 1]

e	p^Te	s^Te
wt $[1\ 0]^T$	0	2
input $[0\ 1]^T$	1	1
result $[1\ -1]^T$	-1	1

$$HUE = \frac{1}{\left|s^T d\right|} = \frac{1}{2}$$



W_2 and Dual W_2-Weights Stay, Inputs and Results Move in Same Direction but at Different Speeds

W_2

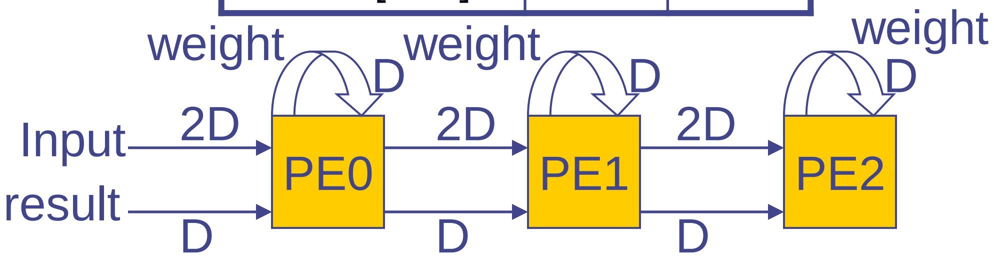
d^T=[1 0]

p^T=[0 1]

s^T=[1 2]

e	p^Te	s^Te
wt $[1\ 0]^T$	0	1
input $[0\ 1]^T$	1	2
result $[1\ -1]^T$	1	1

$$HUE = \frac{1}{\left|s^T d\right|} = 1$$



Dual W_2

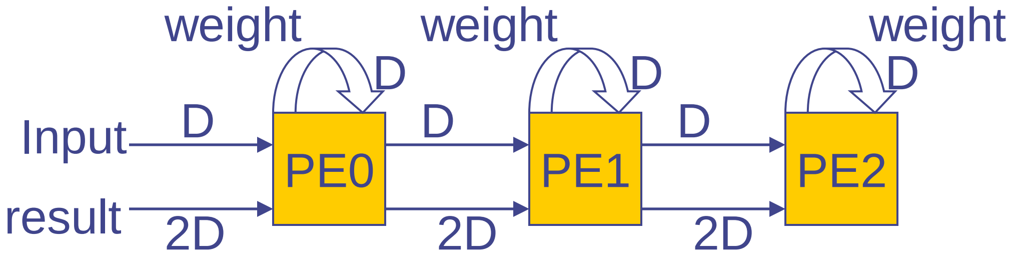
d^T=[1 0]

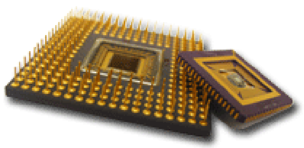
p^T=[0 1]

s^T=[1 -1]

e	p^Te	s^Te
wt $[1\ 0]^T$	0	1
input $[0\ 1]^T$	-1	1
result $[1\ -1]^T$	-1	2

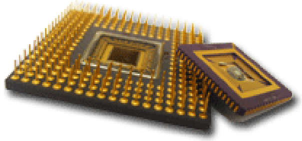
$$HUE = \frac{1}{\left|s^T d\right|} = 1$$



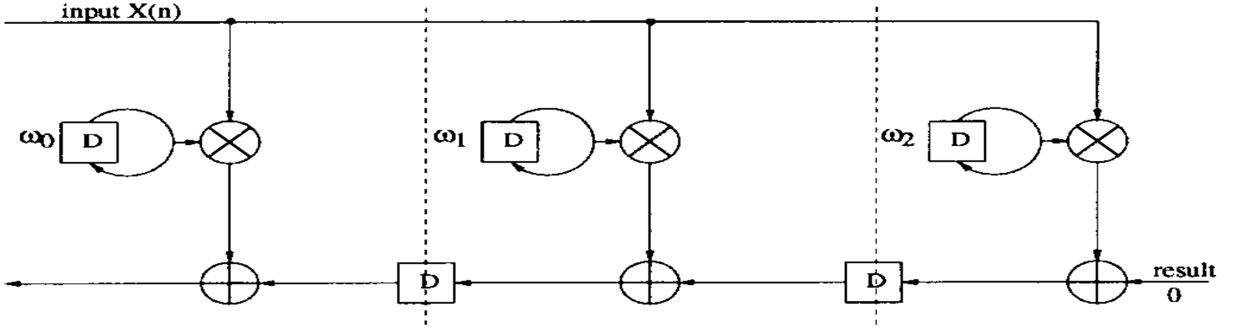


Relating Systolic Designs Using Transformations

- The same projection vector and processor space vector
- Different scheduling vectors
- Can derive each other using transformations
  - *Edge reversal* : reverse edge direction in DG when no precedence constraints
  - *Associativity* : when accumulating *(a+b)+c = a+(b+c)*
  - *Slow-down*
  - *Retiming*
  - *Pipelining*

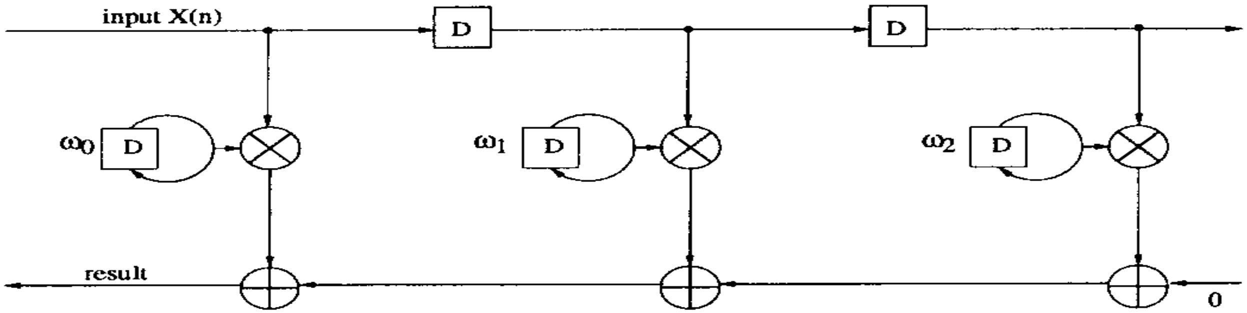


Cutset Retiming Transformation



B1

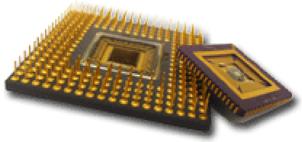
cutset retiming



F

Outline

- Introduction
- Systolic array design methodology
- FIR systolic arrays
- *Selection of scheduling vector*
- Conclusion



Scheduling Inequalities (1/3)

- Based on selected scheduling vector **s**^{T}, the projection vector **d** and the processor space vector **p**^{T} can be selected.

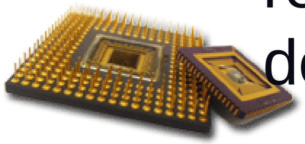
$$\mathbf{p}^T(\mathbf{I}_A - \mathbf{I}_B) = 0 \Rightarrow \mathbf{p}^T\mathbf{d} = 0$$

$$\mathbf{s}^T\mathbf{I}_A \neq \mathbf{s}^T\mathbf{I}_B \Rightarrow \mathbf{s}^T\mathbf{d} \neq 0$$

- Consider the dependence relation X -> Y,

$$X : \mathbf{I}_x = \begin{pmatrix} i_x \\ j_x \end{pmatrix} \Longrightarrow Y : \mathbf{I}_y = \begin{pmatrix} i_y \\ j_y \end{pmatrix}$$

where I_x and I_y are the indices of node X and node Y, respectively. The scheduling inequality for this dependence is defined as



Scheduling Inequalities (2/3)

$$S_y \geq S_x + T_x \qquad \text{Eq. (1)}$$

Where T_x is the time to compute node X and S_x, S_y are the scheduling times for nodes X, Y, respectively.

- Linear scheduling

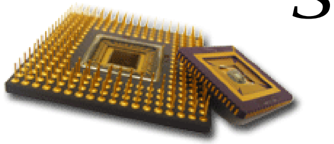
$$S_x = s^T I_x = \begin{pmatrix} s_1 & s_2 \end{pmatrix} \begin{pmatrix} i_x \\ j_x \end{pmatrix}$$

$$S_y = s^T I_y = \begin{pmatrix} s_1 & s_2 \end{pmatrix} \begin{pmatrix} i_y \\ j_y \end{pmatrix} \qquad \text{Eq. (2)}$$

- Affine scheduling

$$S_x = s^T I_x + \gamma_x = \begin{pmatrix} s_1 & s_2 \end{pmatrix} \begin{pmatrix} i_x \\ j_x \end{pmatrix} + \gamma_x$$

$$S_y = s^T I_y + \gamma_y = \begin{pmatrix} s_1 & s_2 \end{pmatrix} \begin{pmatrix} i_y \\ j_y \end{pmatrix} + \gamma_y \qquad \text{Eq. (3)}$$



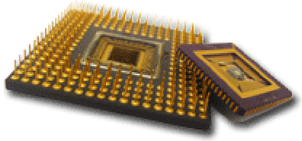
Scheduling Inequalities (3/3)

- Define the edge from node X to node Y as

$$\mathbf{e}_{x-y} = \mathbf{I}_y - \mathbf{I}_x$$

Eqs. (1) & (2) $$=> \mathbf{s}^T \mathbf{e}_{x-y} + r_y - r_x \geq T_x$$

- Hence the selection of scheduling vector consists of two steps:
  - Capture all the fundamental edges. The reduced dependence graph (RDG) is used to capture the fundamental edges and the regular iterative algorithm (RIA) description of the corresponding problem is used to construct RDGs.
  - Construct the scheduling inequalities and solve them for feasible $\mathbf{s}^T$.



Regular Iterative Algorithm (RIA)

- The regular iterative algorithm is the method for constructing the reduce dependence graph (RDG).
- The regular iterative algorithm (RIA) has two standard forms:
  - The RIA is in standard input RIA form if the index of the inputs are the same for all equations.
  - The RIA is in standard output RIA form if output indices are the same for all equations.
- FIR example:

$$W(i+1, j) = W(i, j)$$

$$X(i, j+1) = X(i, j)$$

$$Y(i+1, j-1) = Y(i, j)$$

$$+W(i+1, j-1)X(i+1, j-1)$$

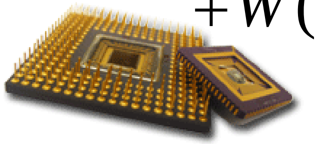
Output RIA Form

$$W(i, j) = W(i-1, j)$$

$$X(i, j) = X(i, j-1)$$

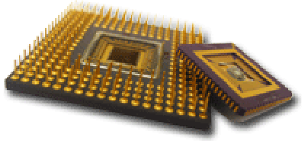
$$Y(i, j) = Y(i-1, j+1)$$

$$+W(i, j)X(i, j)$$

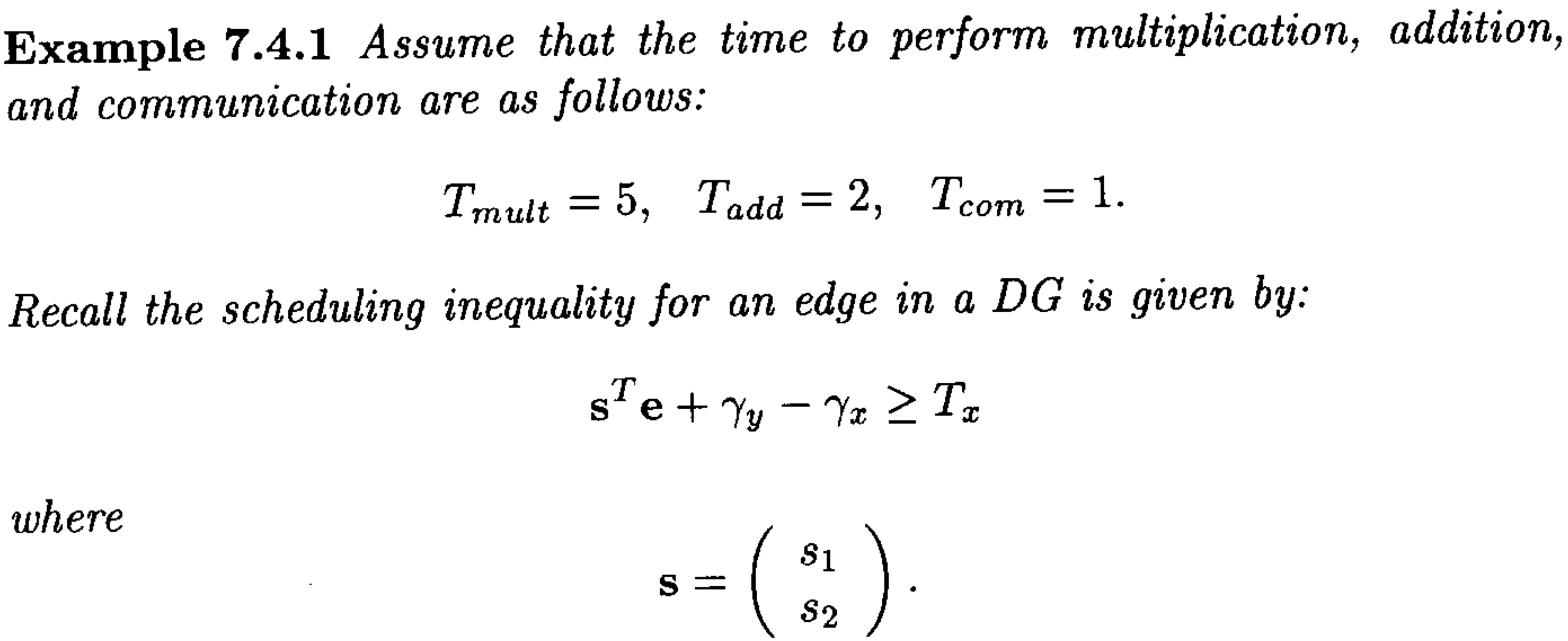


Scheduling Vector and Systolic Array Design Using RDG

- Constructing scheduling inequalities using RDG
- Determine the scheduling vector using scheduling inequalities
- Systolic mapping using the scheduling vector
- This formulation can accommodate different computation times for various operations due to its generality.



Example 7.4.1 (1/4)



Example 7.4.1 *Assume that the time to perform multiplication, addition, and communication are as follows:*

$$T_{mult} = 5, \quad T_{add} = 2, \quad T_{com} = 1.$$

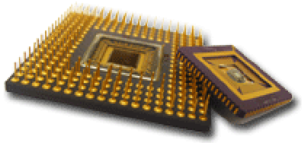
Recall the scheduling inequality for an edge in a DG is given by:

$$\mathbf{s}^T\mathbf{e} + \gamma_y - \gamma_x \geq T_x$$

where

$$\mathbf{s} = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}.$$

There are 5 edges in the above RDG.



Example 7.4.1 (2/4)

- Reduced Dependence Graph (RDG)

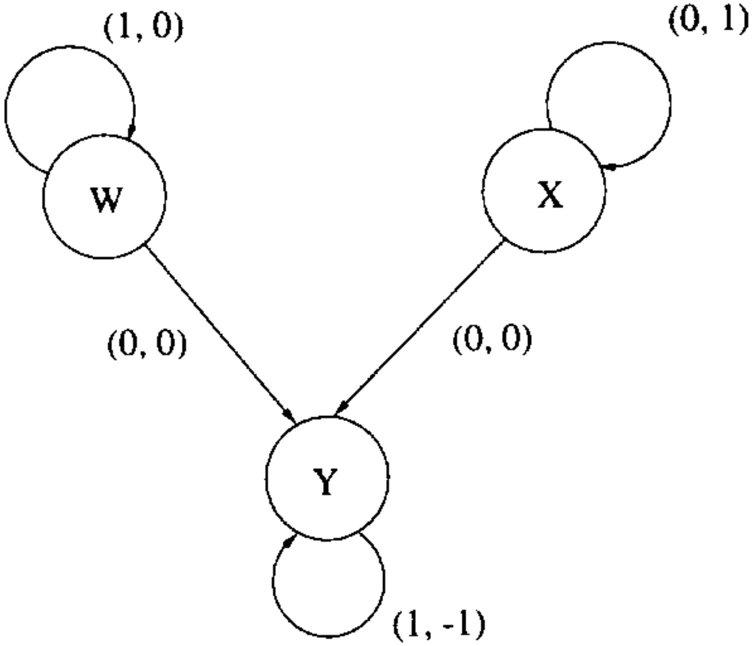
$$W \rightarrow Y : e = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \gamma_y - \gamma_x \geq 0$$

$$X \rightarrow X : e = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, s_2 + \gamma_x - \gamma_x \geq 1$$

$$W \rightarrow W : e = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, s_1 + \gamma_w - \gamma_w \geq 1$$

$$X \rightarrow Y : e = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \gamma_y - \gamma_x \geq 0$$

$$Y \rightarrow Y : e = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, s_1 - s_2 + \gamma_y - \gamma_y \geq 5 + 2 + 1$$



Example 7.4.1 (3/4)

For linear scheduling, $\gamma_x = \gamma_y = \gamma_w = 0$. *Simplifying these equations, we have*

$$s_1 \geq 1, \quad s_2 \geq 1, \quad s_1 - s_2 \geq 8.$$

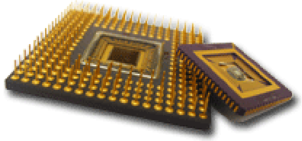
Therefore, one of the solutions is

$$s_2 = 1, s_1 = 8 + 1 = 9 \Rightarrow \mathbf{s}^T = (9, 1).$$

Now, select $\mathbf{d} = (1, -1)$ *such that* $\mathbf{s}^T\mathbf{d} \neq 0$ *and select* $\mathbf{p}^T$ *such that* $\mathbf{p}^T\mathbf{d} = 0$. *Choose* $\mathbf{p}^T = (1, 1)$. *Since*

$$\mathbf{s}^T\mathbf{d} = \begin{pmatrix} 9 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = 8,$$

$$\textit{therefore } HUE = 1/8.$$



Example 7.4.1 (4/4)

- Linear scheduling

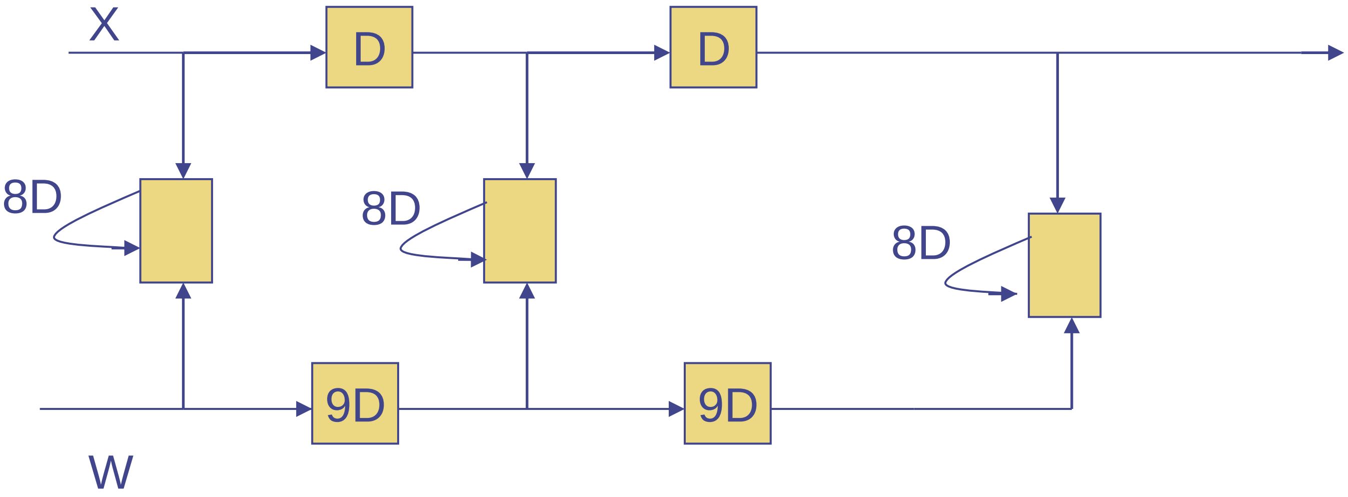
$$s_1 \geq 1, s_2 \geq 1, s_1 - s_2 \geq 8$$

$$s_2 = 1, s_1 = 9 \rightarrow s^T = \begin{pmatrix} 9 & 1 \end{pmatrix}$$

$$d = (1, -1), p^T = (1, 1)$$

e	p^Te	s^Te
wt(1,0)	1	9
i/p(0,1)	1	1
Result(1,-1)	0	8

- Systolic array architecture



Conclusion

- Systolic architecture
  - A massively parallel processing with limited I/O communication with host computer
  - Suitable for many regular interactive operations
- Design methodology
  - Map an N-dimensional DG to (N-1) dimensional space-time representation
  - Needs to determine three critical vectors
    - Projection vector
    - Processor space vector
    - Scheduling vector