Chapter 7: Systolic Architecture Design

Keshab K. Parhi

- Systolic architectures are designed by using linear mapping techniques on regular dependence graphs (DG).
- Regular Dependence Graph : The presence of an edge in a certain direction at any node in the DG represents presence of an edge in the same direction at all nodes in the DG.
- DG corresponds to space representation → no time instance is assigned to any computation ⇒ t=0.
- Systolic architectures have a space-time representation where each node is mapped to a certain processing element(PE) and is scheduled at a particular time instance.
- Systolic design methodology maps an N-dimensional DG to a lower dimensional systolic architecture.
- Mapping of N-dimensional DG to (N-1) dimensional systolic array is considered.

- Definitions :
  - Projection vector (also called iteration vector), $d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$

    Two nodes that are displaced by d or multiples of d are executed by the same processor.
  - Processor space vector, $p^T = (p_1 \quad p_2)$

    Any node with index I^T=(i,j) would be executed by processor;

$$p^T I = (p_1 \quad p_2) \begin{pmatrix} i \\ j \end{pmatrix}$$

  - Scheduling vector, $s^T = (s_1\ s_2)$. Any node with index I would would be executed at time, $s^T I$.
  - Hardware Utilization Efficiency, HUE = $1/|S^T d|$. This is because two tasks executed by the same processor are spaced $|S^T d|$ time units apart.
  - Processor space vector and projection vector must be orthogonal to each other $\Rightarrow p^T d = 0$.

- If A and B are mapped to the same processor, then they cannot be executed at the same time, i.e., $S^T I_A \neq S^T I_B$, i.e., $S^T d \neq 0$.
- Edge mapping : If an edge e exists in the space representation or DG, then an edge $p^T e$ is introduced in the systolic array with $s^T e$ delays.
- A DG can be transformed to a space-time representation by interpreting one of the spatial dimensions as temporal dimension. For a 2-D DG, the general transformation is described by i' = t = 0, j' = $p^T I$, and t' = $s^T I$, i.e.,

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

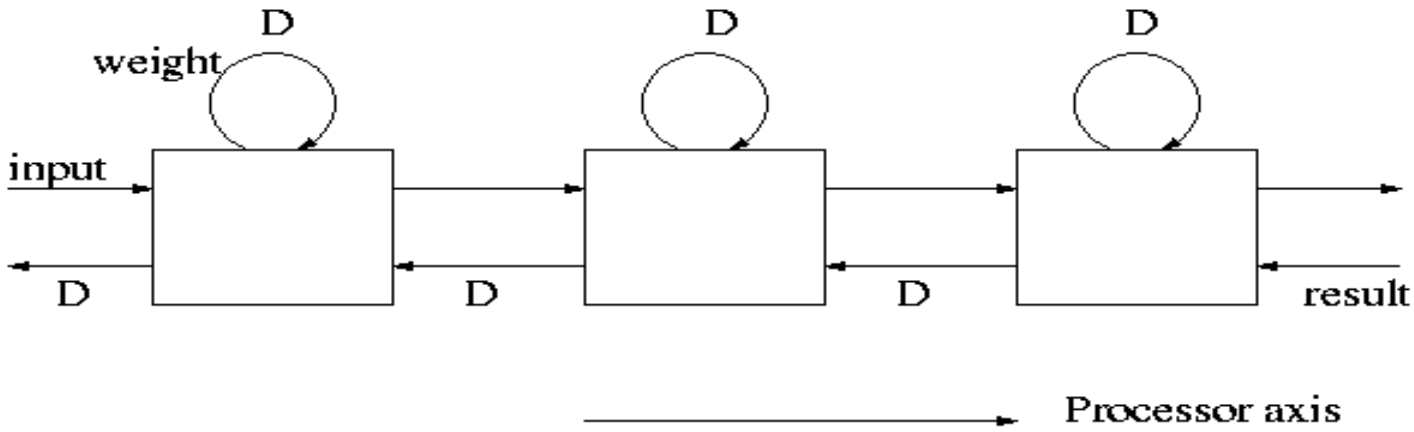
j' $\Rightarrow$ processor axis
t' $\Rightarrow$ scheduling time instance

FIR Filter Design B_1 (Broadcast Inputs, Move Results, Weights Stay)

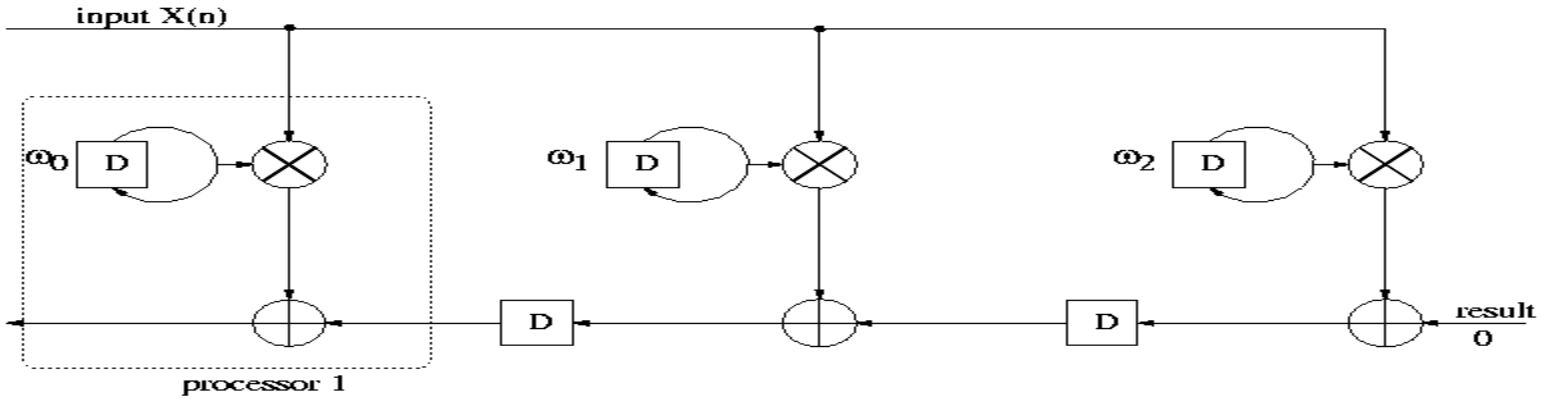
$$d^T = (1\ 0),\ p^T = (0\ 1),\ s^T = (1\ 0)$$

- Any node with index $I^T = (i\ ,\ j)$
  - is mapped to processor $p^T I = j$.
  - is executed at time $s^T I = i$.
- Since $s^T d = 1$ we have HUE = $1/|s^T d| = 1$.
- Edge mapping : The 3 fundamental edges corresponding to weight, input, and result can be mapped to corresponding edges in the systolic array as per the following table:

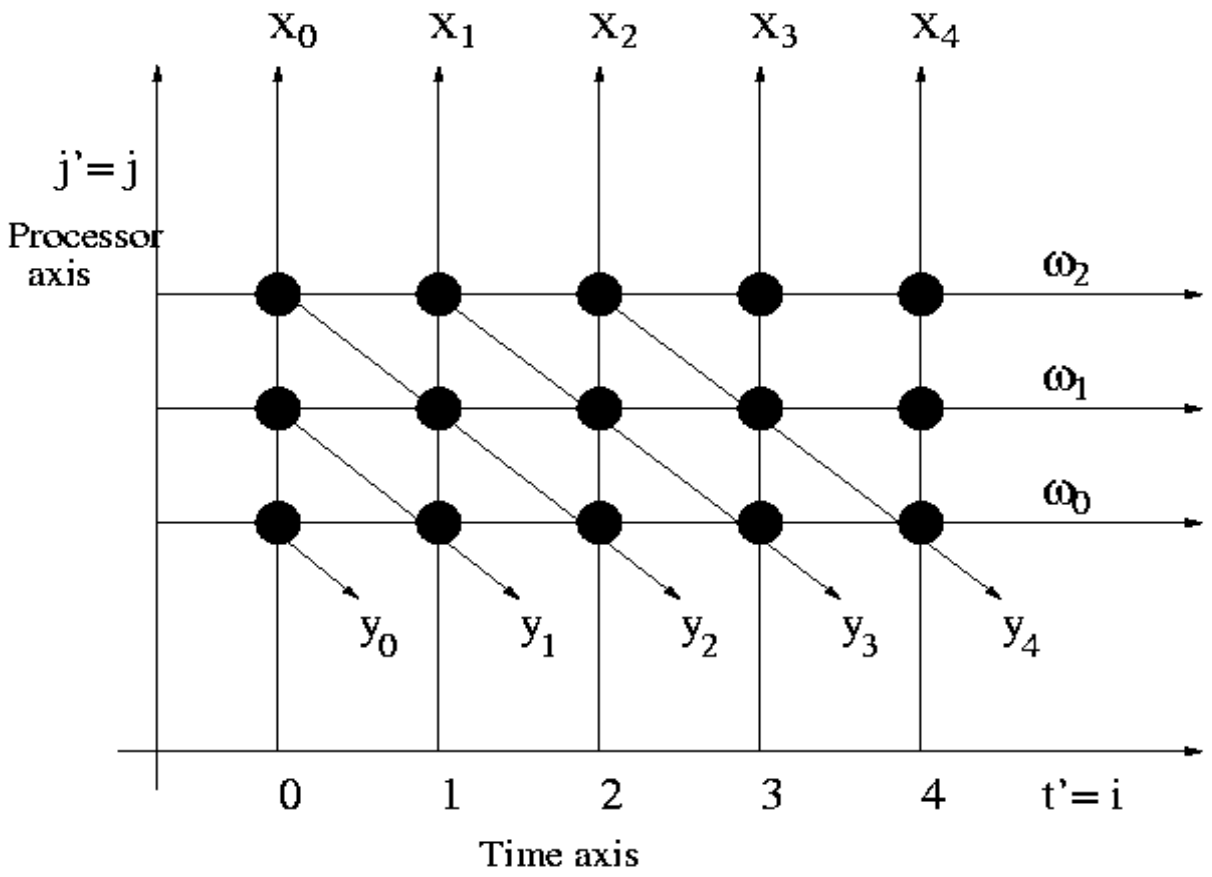
e	$p^T e$	$s^T e$
wt(1 0)	0	1
i/p(0 1)	1	0
result(1 –1)	-1	1



Block diagram of B_1 design



Low-level implementation of B_1 design



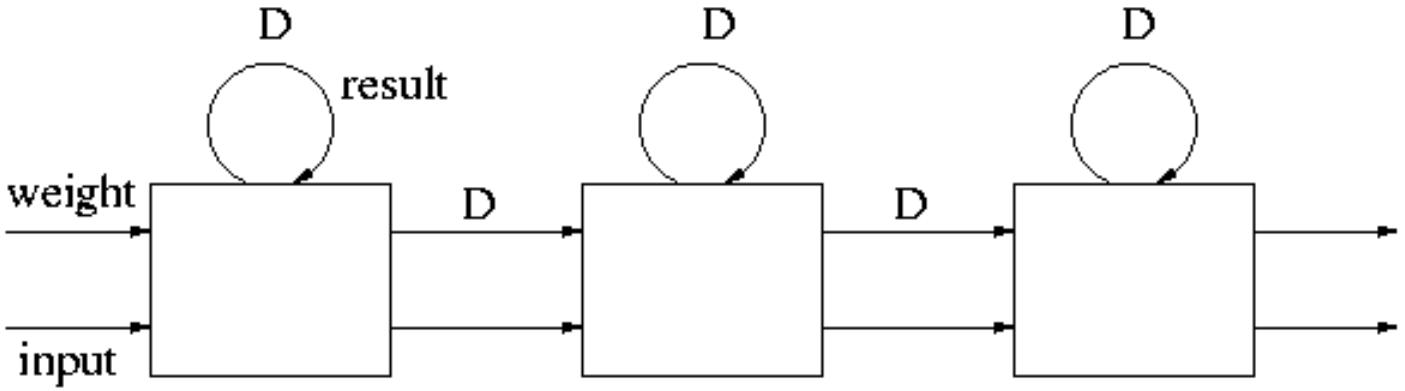
Space-time representation of B_1 design

Design B_2 (Broadcast Inputs, Move Weights, Results Stay)

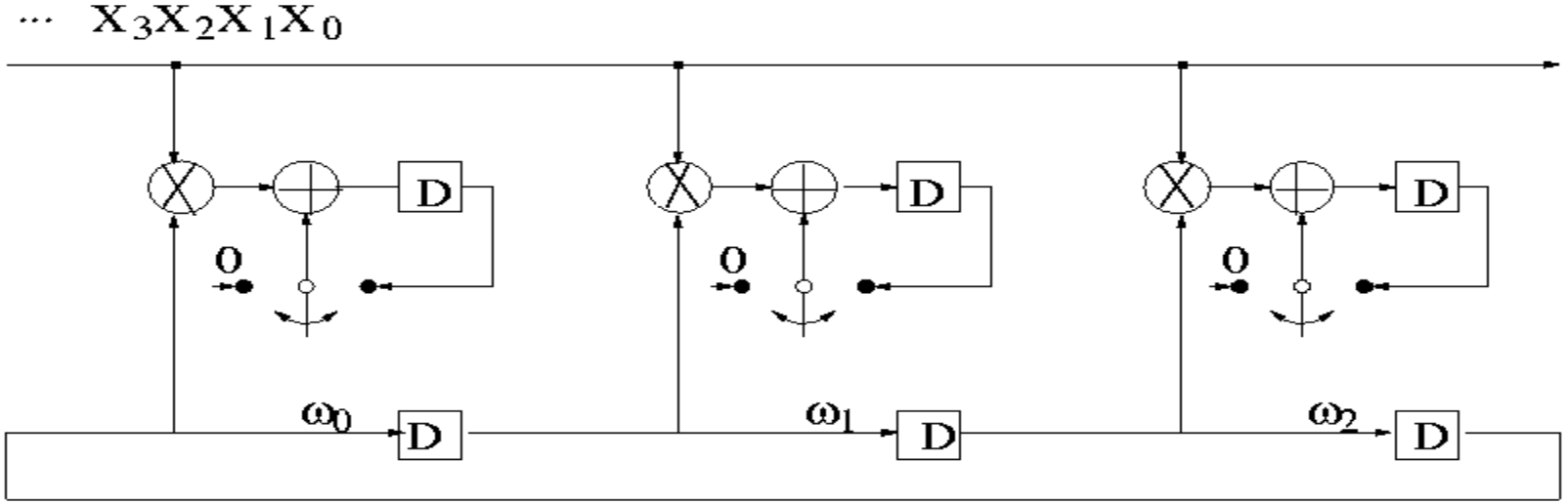
$$d^T = (1\ -1),\ p^T = (1\ 1),\ s^T = (1\ 0)$$

- Any node with index $I^T = (i\ ,\ j)$
  - is mapped to processor $p^T I = i+j$.
  - is executed at time $s^T I = i$.
- Since $s^T d = 1$ we have HUE = $1/|s^T d| = 1$.
- Edge mapping :

e	$p^T e$	$s^T e$
wt(1 0)	1	1
i/p(0 1)	1	0
result(1 –1)	0	1



Block diagram of B_2 design

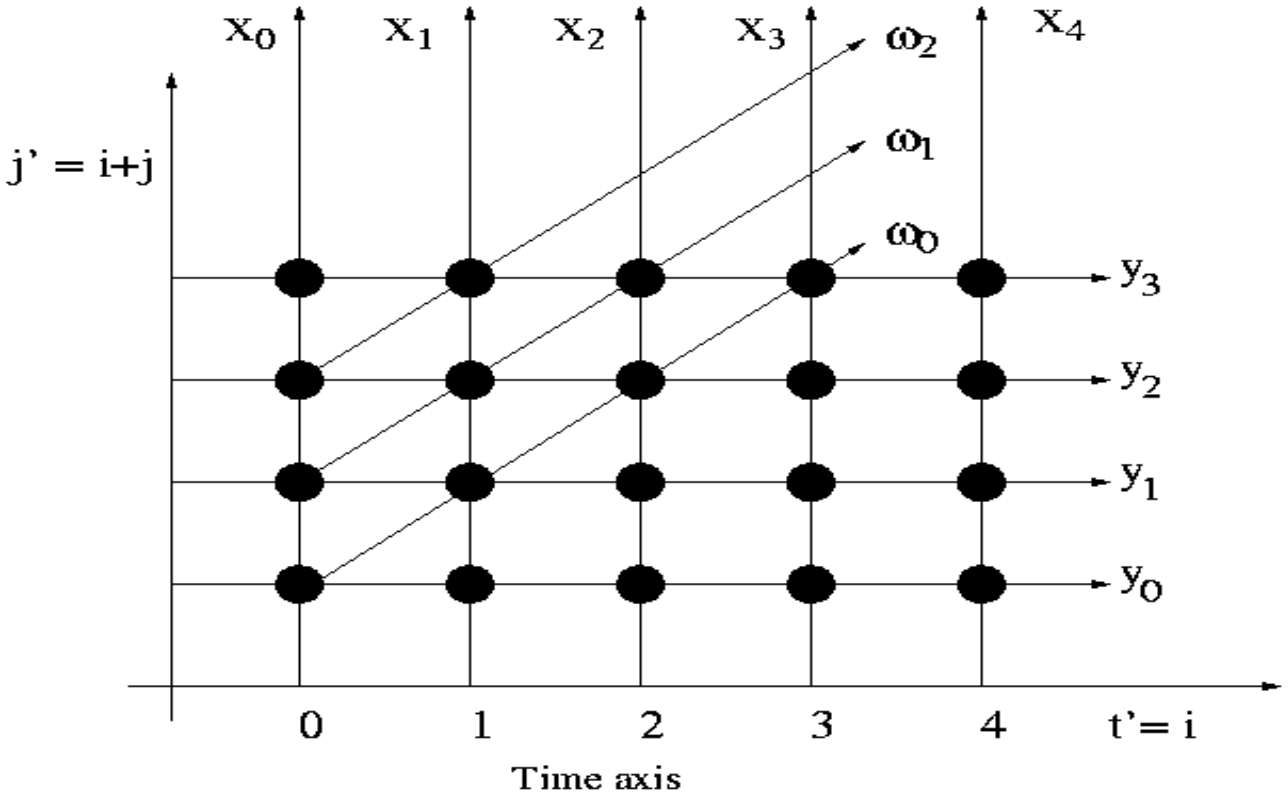


Low-level implementation of B_2 design

- Applying space time transformation we get :

$$j' = p^T(i\ j)^T = i + j$$

$$t' = s^T(i\ j)^T = i$$



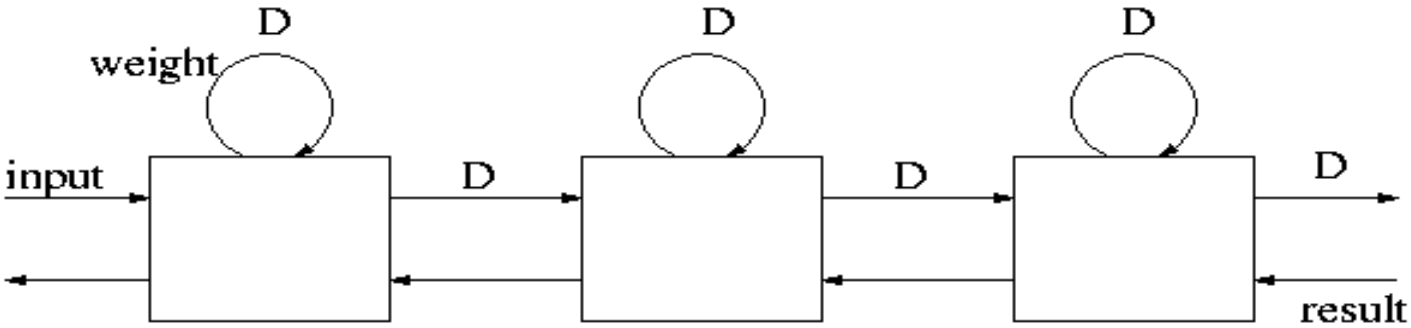
Space-time representation of B_2 design

Design F(Fan-In Results, Move Inputs, Weights Stay)

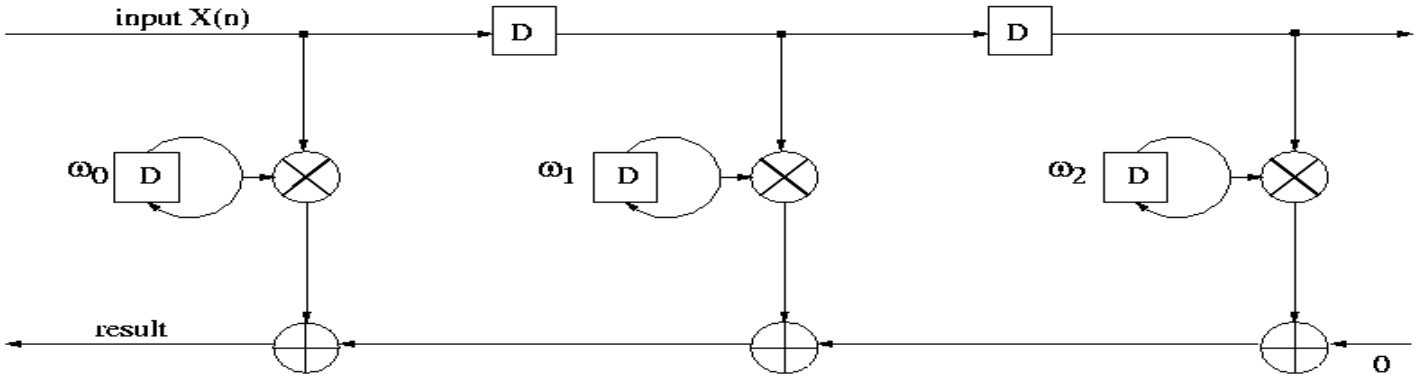
$d^T = (1\ 0)$, $p^T = (0\ 1)$, $s^T = (1\ 1)$

- Since $s^T d = 1$ we have HUE = $1/|s^T d| = 1$.
- Edge mapping :

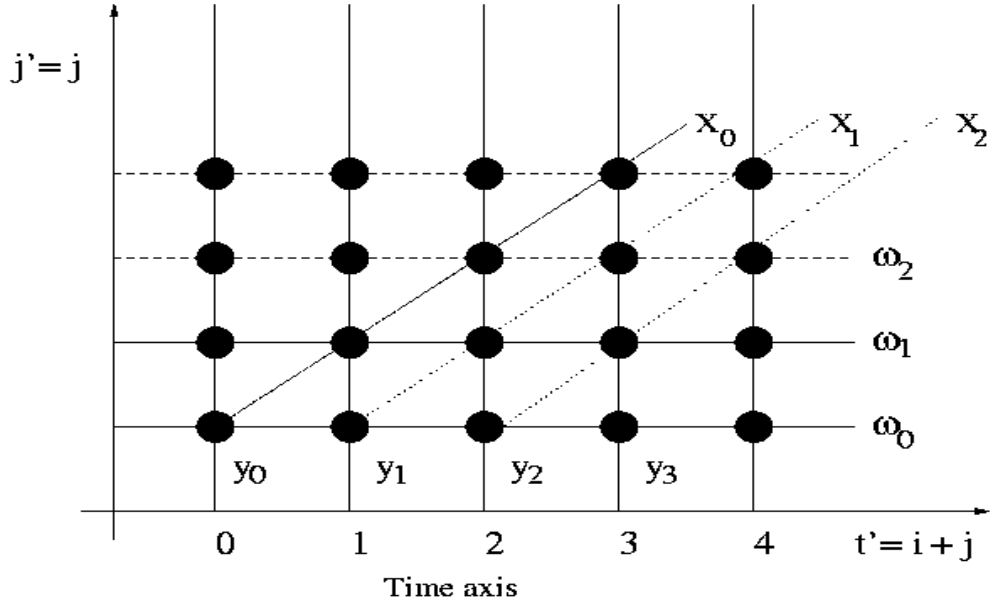
e	$p^T e$	$s^T e$
wt(1 0)	0	1
i/p(0 1)	1	1
result(1 –1)	-1	0



Block diagram of F design



Low-level implementation of F design



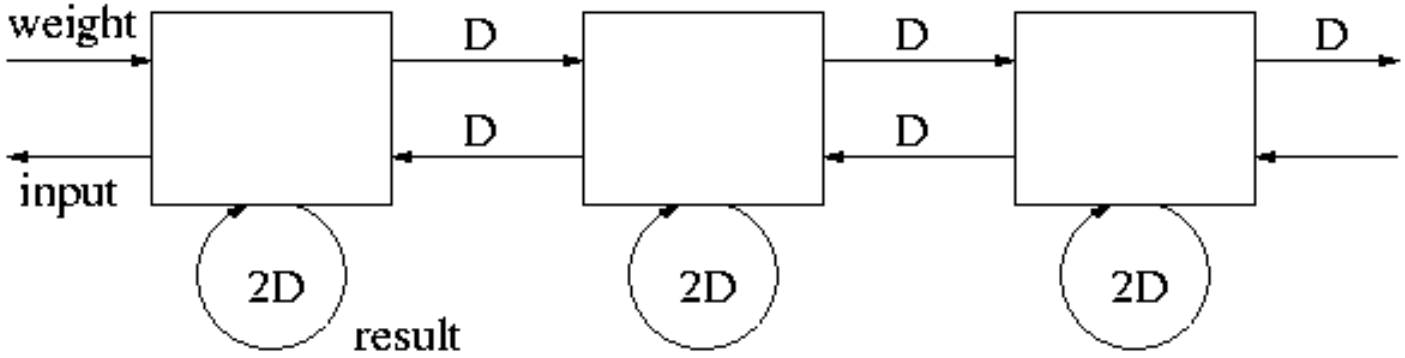
Space-time representation of F design

Design R_1 (Results Stay, Inputs and Weights Move in Opposite Direction)

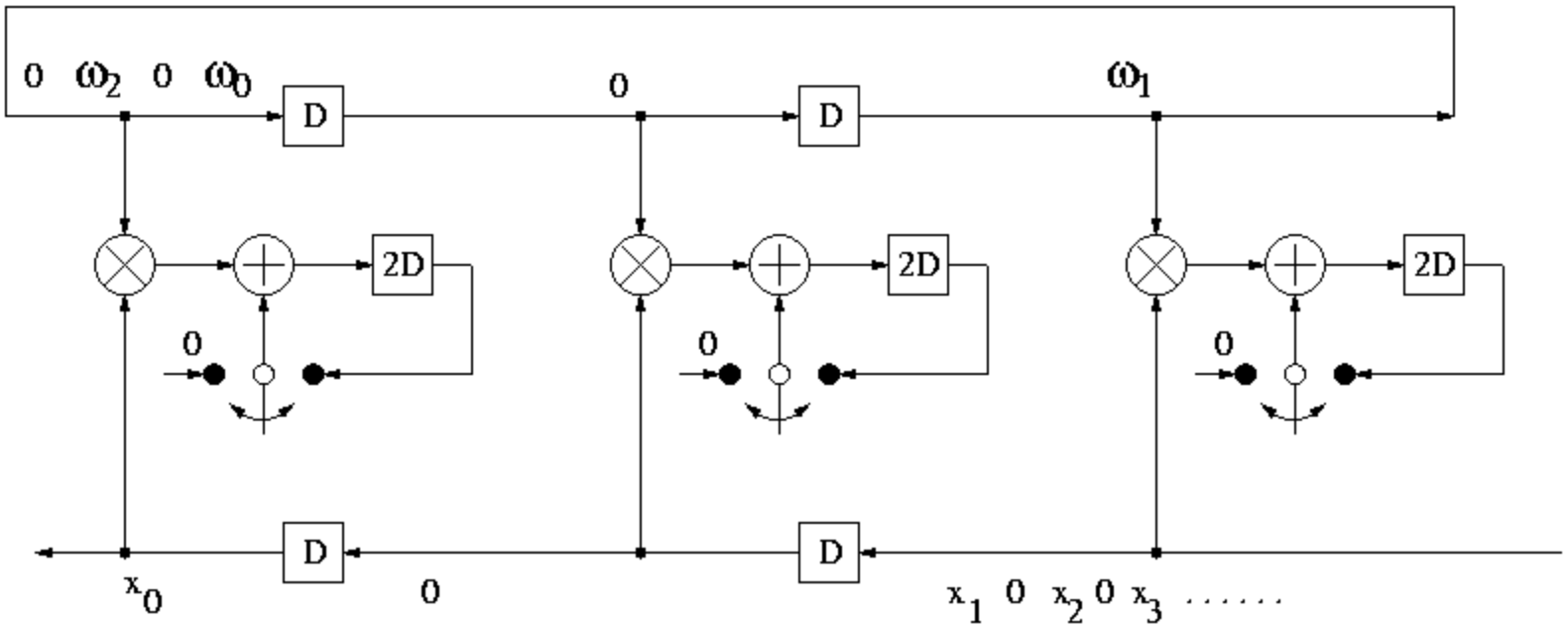
$$d^T = (1\ -1),\ p^T = (1\ 1),\ s^T = (1\ -1)$$

- Since $s^T d = 2$ we have HUE = $1/|s^T d| = ½$.
- Edge mapping :

e	$p^T e$	$s^T e$
wt(1 0)	1	1
i/p(0 -1)	-1	1
result(1 –1)	0	2



Block diagram of R_1 design



Low-level implementation of R_1 design

Note : R_1 can be obtained from B_2 by 2-slow transformation and then retiming after changing the direction of signal x.

Design R_2 and Dual R_2 (Results Stay, Inputs and Weights Move in Same Direction but at Different Speeds)

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

$R_2 : s^T = (2\ 1)$; Dual $R_2 : s^T = (1\ 2)$;

- Since $s^T d = 1$ for both of them we have HUE = $1/|s^T d| = 1$ for both.
- Edge mapping :

R_2			Dual R_2		
e	$p^T e$	$s^T e$	e	$p^T e$	$s^T e$
wt(1, 0)	1	2	wt(1, 0)	1	1
i/p(0,1)	1	1	i/p(0,1)	1	2
result(1, -1)	0	1	result(-1, 1)	0	1

Note : The result edge in design dual R_2 has been reversed to Guarantee $s^T e \geq 0$.

Design W_1 (Weights Stay, Inputs and Results Move in Opposite Directions)

$$d^T = (1\ 0),\ p^T = (0\ 1),\ s^T = (2\ 1)$$

- Since $s^T d = 2$ for both of them we have HUE = $1/|s^T d| = ½$.
- Edge mapping :

e	$p^T e$	$s^T e$
wt(1 0)	0	2
i/p(0 -1)	1	1
result(1 –1)	-1	1

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

$d^T = (1\ 0)$, $p^T = (0\ 1)$,

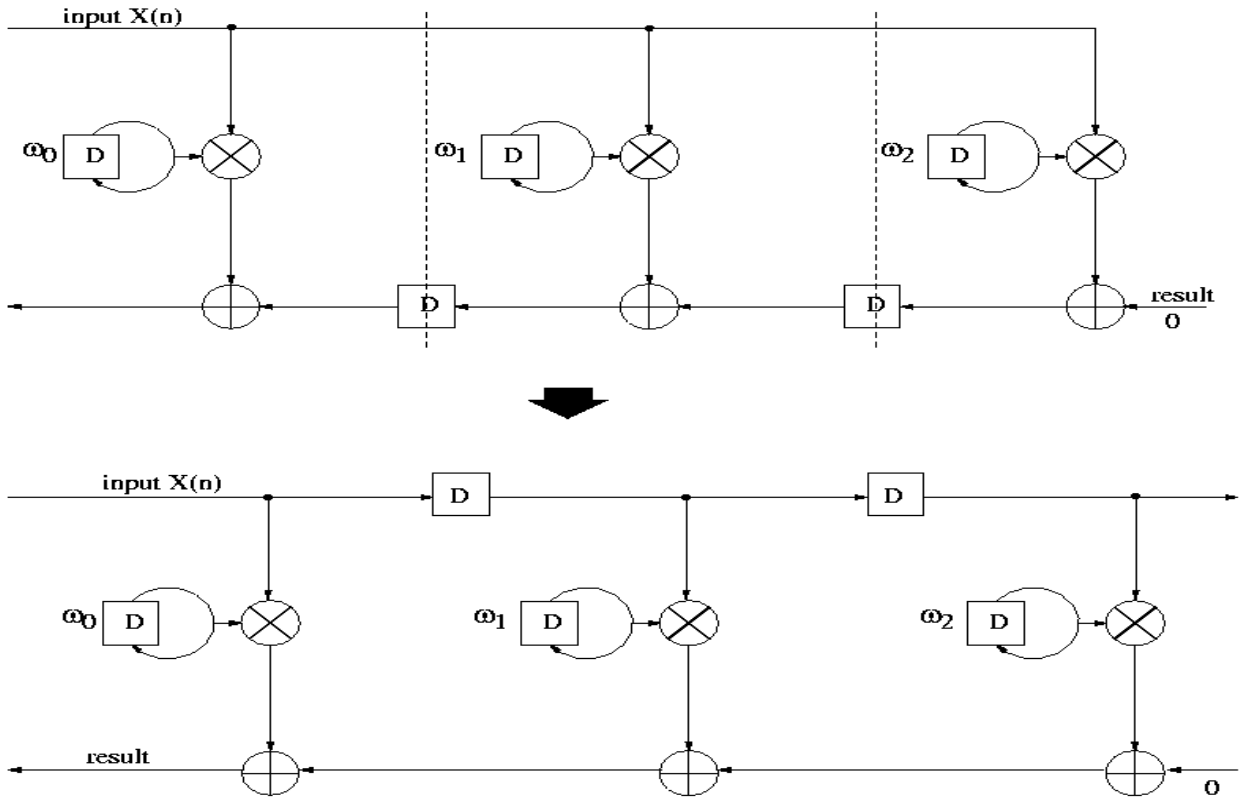
W_2 : $s^T = (1\ 2)$; Dual W_2 : $s^T = (1\ -1)$;

- Since $s^T d = 1$ for both of them we have HUE = $1/|s^T d| = 1$ for both.
- Edge mapping :

W_2			Dual W_2		
e	$p^T e$	$s^T e$	e	$p^T e$	$s^T e$
wt(1, 0)	0	1	wt(1, 0)	0	1
i/p(0,1)	1	2	i/p(0,-1)	-1	1
result(1, -1)	1	1	result(1, -1)	-1	2

- Relating Systolic Designs Using Transformations :
  - FIR systolic architectures obtained using the same projection vector and processor vector, but different scheduling vectors, can be derived from each other by using transformations like edge reversal, associativity, slow-down, retiming and pipelining.
- Example 1 : R_1 can be obtained from B_2 by slow-down, edge reversal and retiming.

- Example 2:



Derivation of design F from B_1 using cutset retiming

- Selection of s^T based on scheduling inequalities:

  For a dependence relation X → Y, where $I_x^T = (i_x, j_x)^T$ and $I_y^T = (i_y, j_y)^T$ are respectively the indices of the nodes X and Y. The scheduling inequality for this dependence is given by,

$$S_y \geq S_x + T_x$$

  where T_x is the computation time of node X. The scheduling equations can be classified into the following two types :

  - Linear scheduling, where

$$S_x = s^T I_x = (s_1\ s_2)(i_x\ \ j_x)^T$$
$$S_y = s^T I_y = (s_1\ s_2)(i_y\ \ j_y)^T$$

  - Affine Scheduling, where

$$S_x = s^T I_x + \gamma_x = (s_1\ s_2)(i_x\ \ j_x)^T + \gamma_x$$
$$S_x = s^T I_x + \gamma_y = (s_1\ s_2)(i_x\ \ j_x)^T + \gamma_y$$

  So scheduling equation for affine scheduling is as follows:

$$s^T I_x + \gamma_y \geq s^T I_x + \gamma_x + T_x$$

Each edge of a DG leads to an inequality for selection of the scheduling vectors which consists of 2 steps.

- Capture all 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 according to

$$s^T I_x + \gamma_y \geq s^T I_x + \gamma_x + T_x$$

and solve them for feasible s^T.

- RIA Description : The RIA has two forms
  - $\Rightarrow$ The RIA is in standard input RIA form if the index of the inputs are the same for all equations.
  - $\Rightarrow$ The RIA is in standard output RIA form if all the output indices are the same.
- For the FIR filtering example we have,

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

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

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

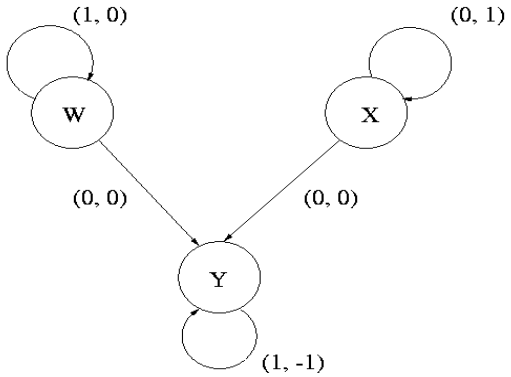
The FIR filtering problem cannot be expressed in standard input RIA form. Expressing it in standard output RIA form we get,

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

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

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

- The reduced DG for FIR filtering is shown below.



Example :

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

Applying the scheduling equations to the five edges of the above figure we get ;

W-->Y : $e = (0\ 0)^T$, $\gamma_x - \gamma_w \geq 0$

X -->X : $e = (0\ 1)^T$, $s_2 + \gamma_x - \gamma_x \geq 1$

W-->W: $e = (1\ 0)^T$, $s_1 + \gamma_w - \gamma_w \geq 1$

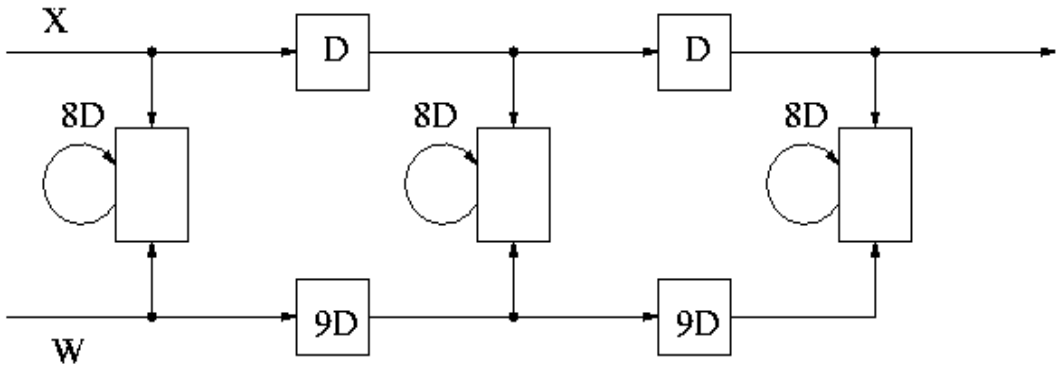
X -->Y : $e = (0\ 0)^T$, $\gamma_y - \gamma_x \geq 0$

Y --> Y: $e = (1\ -1)^T$, $s_1 - s_2 + \gamma_y - \gamma_y \geq 5 + 2 + 1$

For linear scheduling $\gamma_x = \gamma_y = \gamma_w = 0$. Solving we get, $s_1 \geq 1$, $s_2 \geq 1$ and $s_1 - s_2 \geq 8$.

- Taking s^T = (9 1), d = (1 -1) such that $s^T d \neq 0$ and p^T = (1,1) such that $p^T d = 0$ we get HUE = 1/8. The edge mapping is as follows :

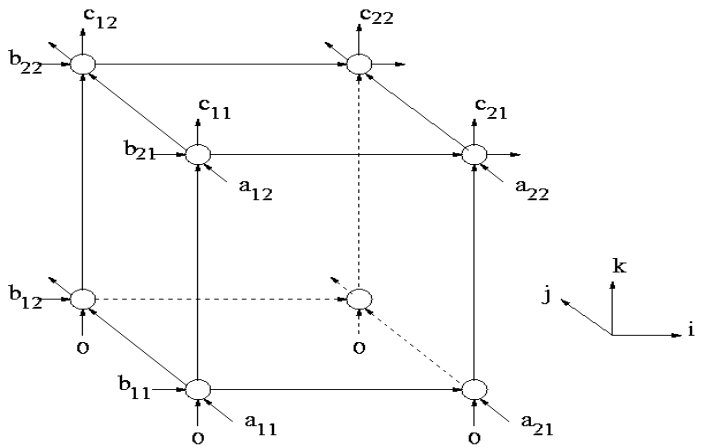
e	$p^T e$	$s^T e$
wt(1 0)	1	9
i/p(0 1)	1	1
result(1 –1)	0	8



Systolic architecture for the example

Matrix-Matrix multiplication and 2-D Systolic Array Design

$$C_{11} = a_{11}b_{11} + a_{12}\, b_{21}$$
$$C_{12} = a_{11}b_{12} + a_{12}\, b_{22}$$
$$C_{21} = a_{21}b_{11} + a_{22}\, b_{21}$$
$$C_{22} = a_{21}b_{12} + a_{22}\, b_{22}$$

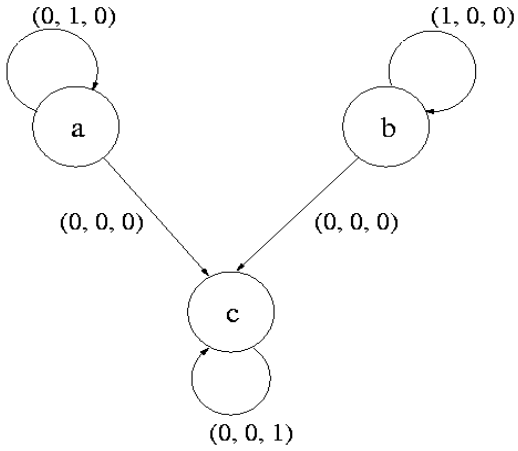


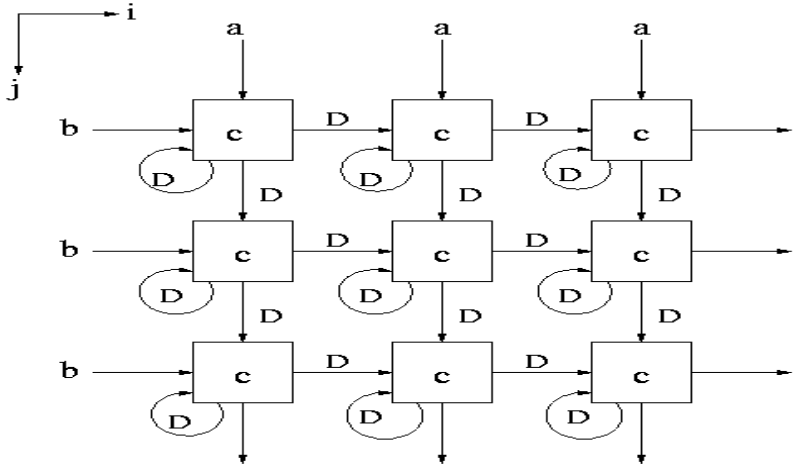
The iteration in standard output RIA form is as follows :

$$a(i,j,k) = a(i,j-1,k)$$
$$b(i,j,k) = b(i-1,j,k)$$
$$c(i,j,k) = c(i,j,k-1) + a(i,j,k)\, b(i,j,k)$$

- Applying scheduling inequality with $T_{mult-add} = 1$, and $T_{com} = 0$ we get $s_2 \geq 0$, $s_1 \geq 0$, $s_3 \geq 1$, $\gamma_c - \gamma_a \geq 0$ and $\gamma_c - \gamma_b \geq 0$. Take $\gamma_a = \gamma_b = \gamma_c = 0$ for linear scheduling.
- Solution 1 :

  $s^T = (1,1,1)$, $d^T = (0,0,1)$, $p_1 = (1,0,0)$, $p_2 = (0,1,0)$, $P^T = (p_1\ p_2)^T$

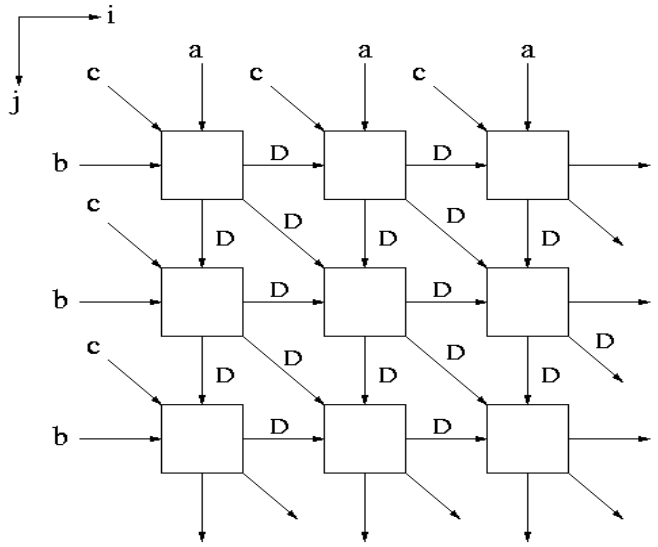




- Solution 2 :

  $s^T = (1,1,1)$, $d^T = (1,1,-1)$, $p_1 = (1,0,1)$,

  $p_2 = (0,1,1)$, $P^T = (p_1\ p_2)^T$



Sol. 1			Sol. 2		
e	$p^T e$	$s^T e$	e	$p^T e$	$s^T e$
a(0, 1, 0)	(0, 1)	1	a(0, 1, 0)	(0, 1)	1
b(1, 0, 0)	(1, 0)	1	b(1, 0, 0)	(1, 0)	1
C(0, 0, 1)	(0, 0)	1	C(0, 0, 1)	(1, 1)	1