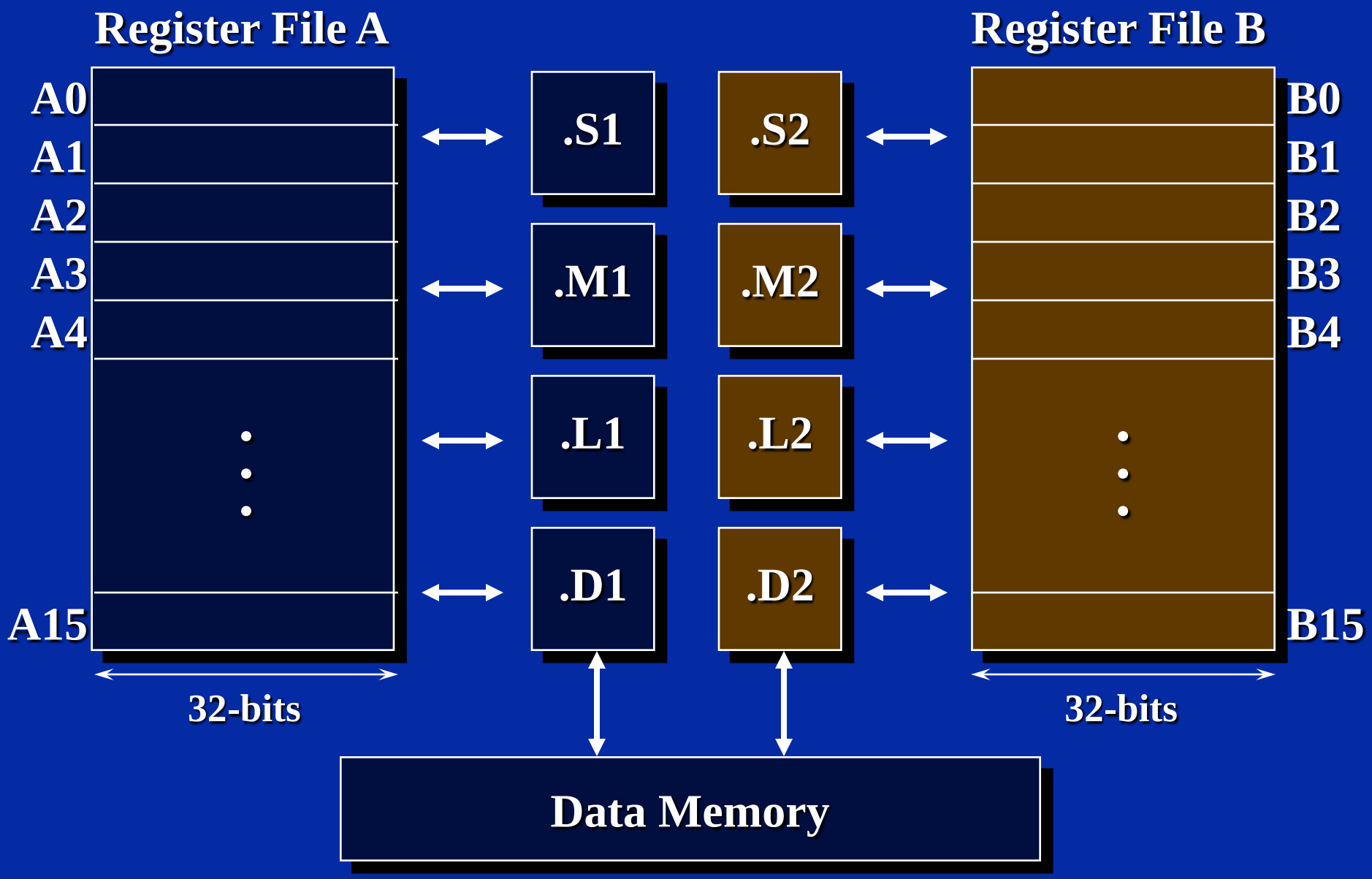
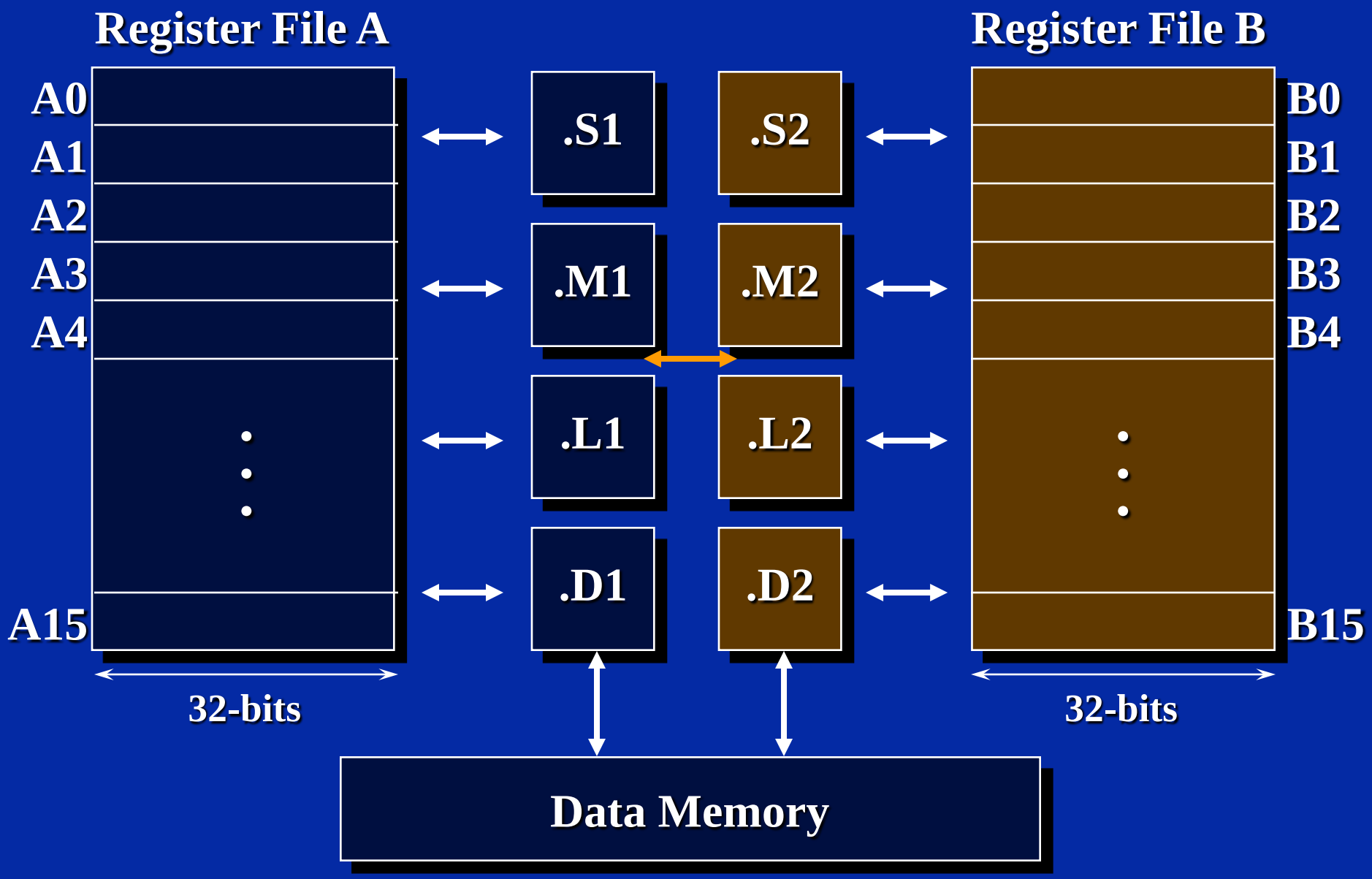
Processeurs de traitement du signal et de l'image

Chap 2: Programmation DSP TMSC6711: Unités d'échange, convolution et projet détection de contour

To increase the Processing Power, this processor has two sides (A and B or 1 and 2)



Can the two sides exchange operands in order to increase performance?



The answer is YES but there are limitations.

- To exchange operands between the two sides, some cross paths or links are required.

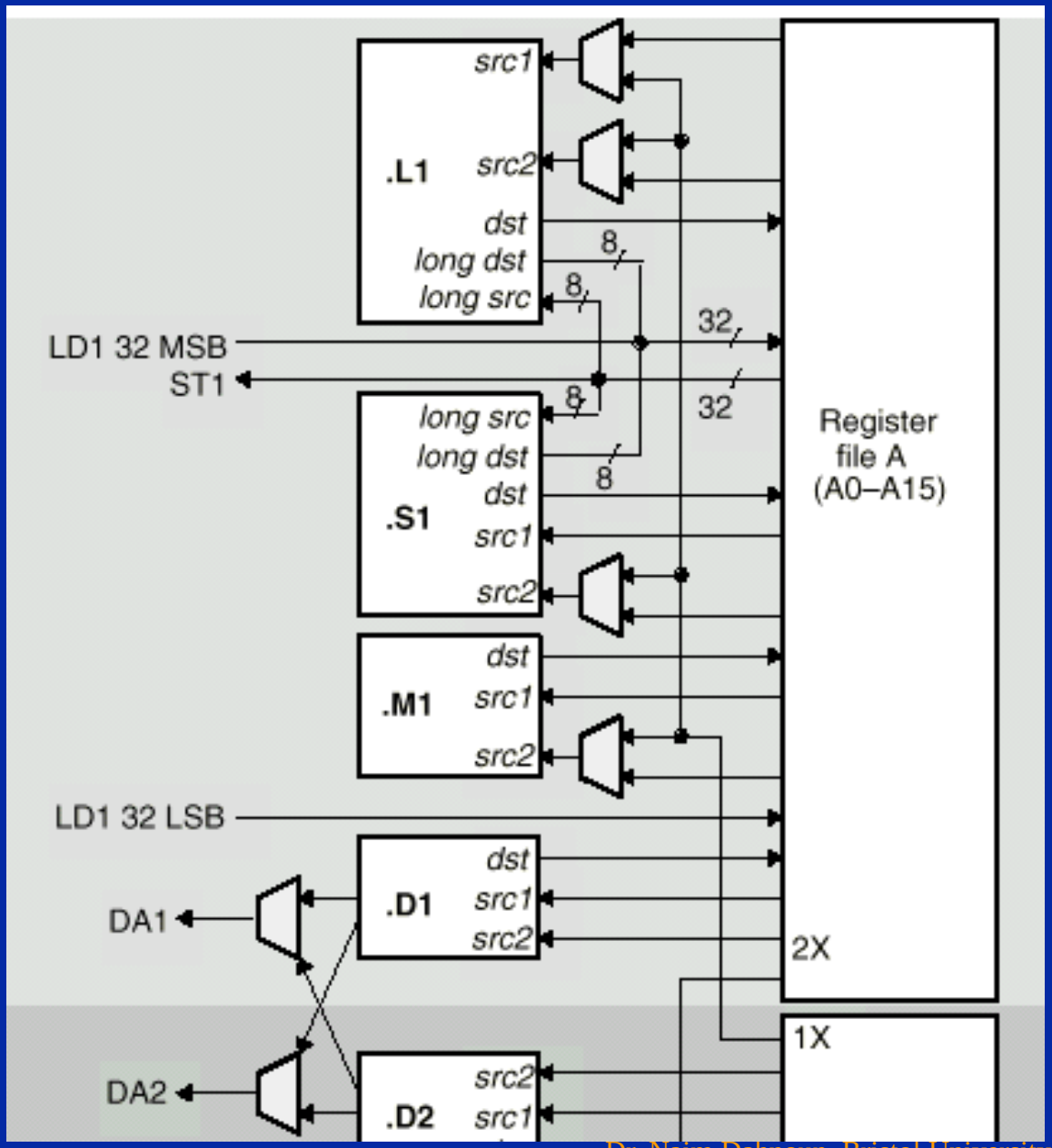
What is a cross path?

- A cross path links one side of the CPU to the other.
- There are two types of cross paths:
  - **Data** cross paths.
  - **Address** cross paths.

Data Cross Paths

- Data cross paths can also be referred to as register file cross paths.
- These cross paths allow operands from one side to be used by the other side.
- There are only two cross paths:
  - one path which conveys data from side B to side A, 1X.
  - one path which conveys data from side A to side B, 2X.

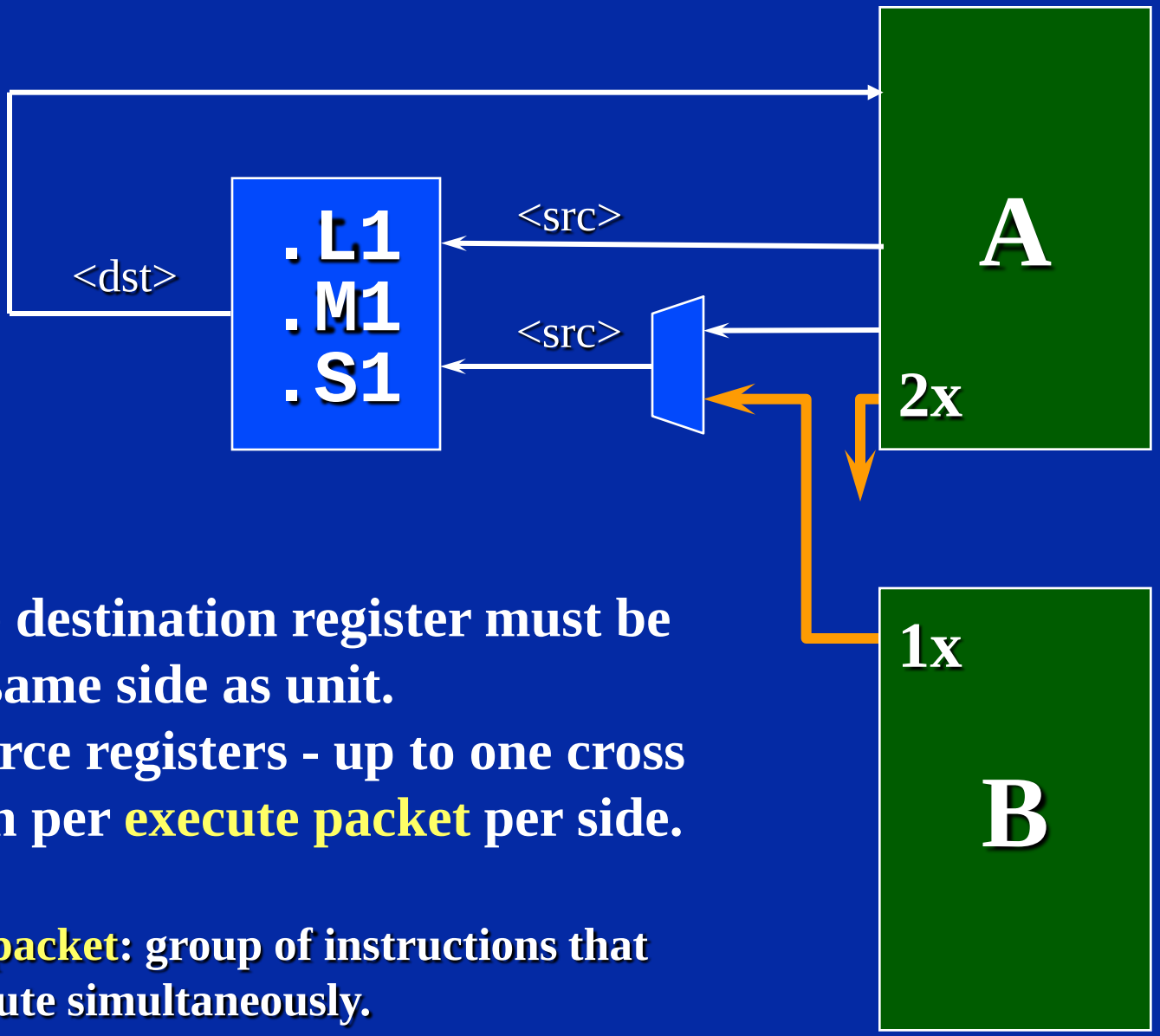
TMS320C67x Data-Path



Data Cross Paths

- Data cross paths only apply to the .L, .S and .M units.
- The data cross paths are very useful, however there are some limitations in their use.

Data Cross Path Limitations

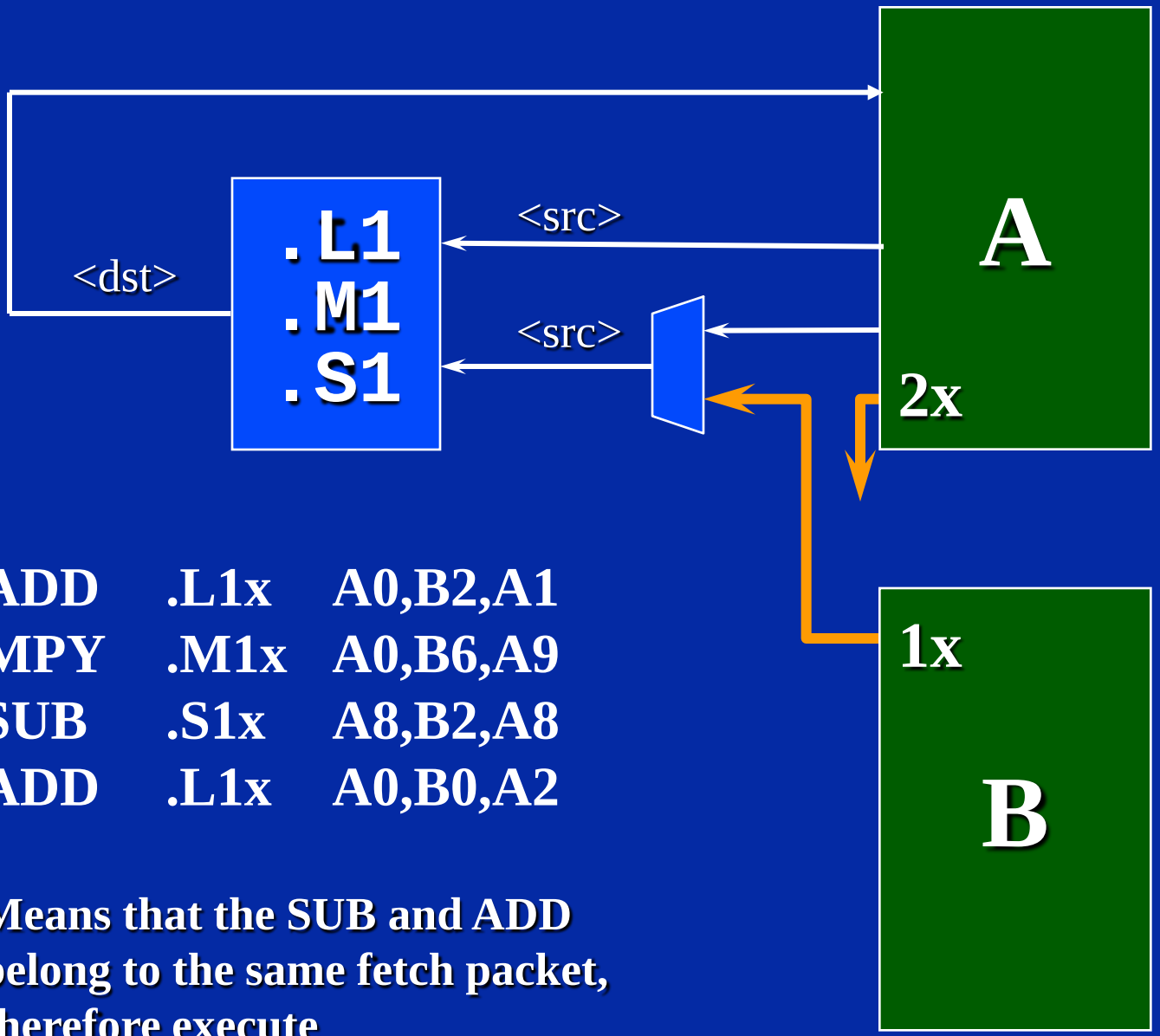


(1) The destination register must be on same side as unit.

(2) Source registers - up to one cross path per **execute packet** per side.

Execute packet: group of instructions that execute simultaneously.

Data Cross Path Limitations

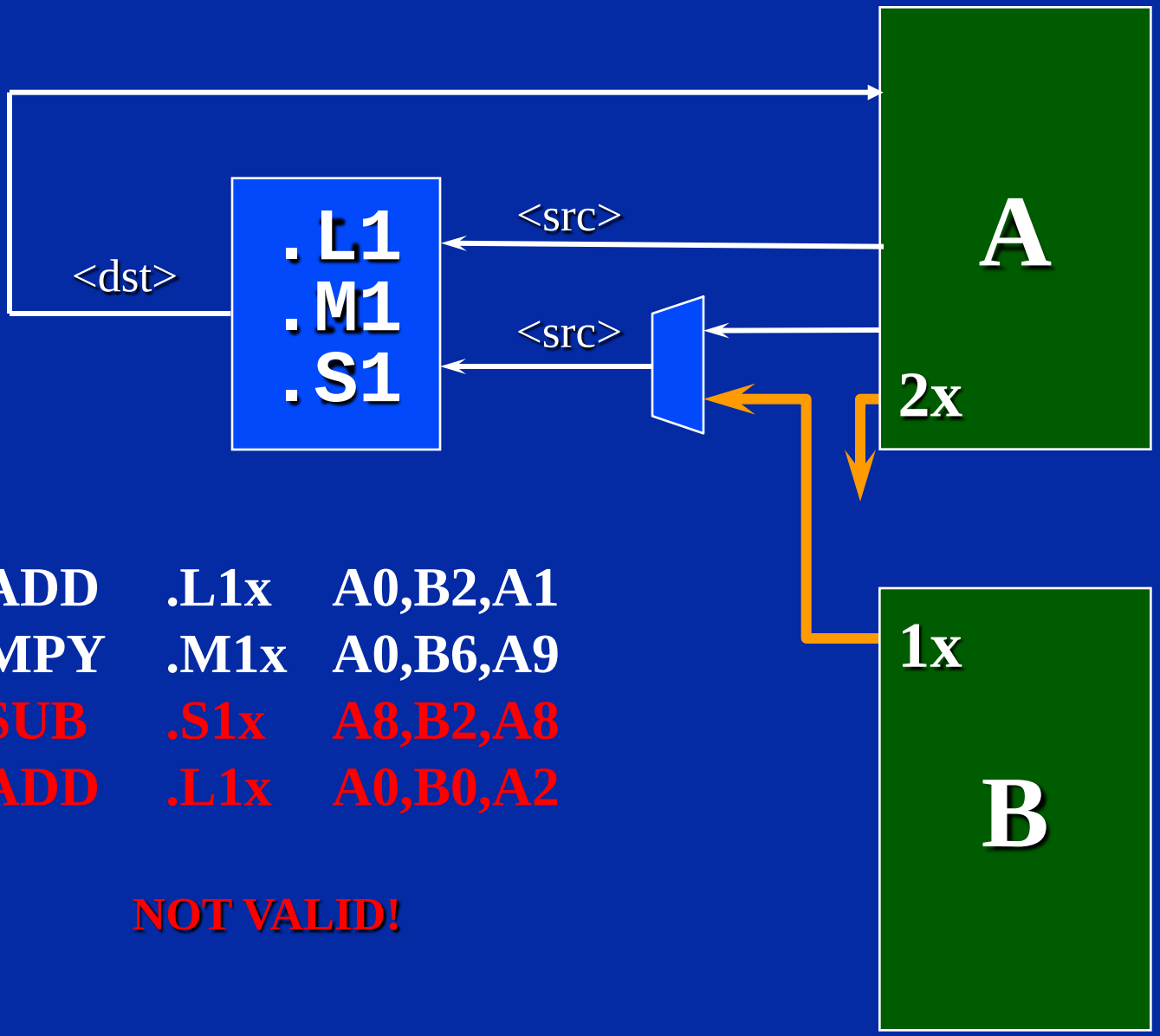


eg:

```
ADD    .L1x   A0,B2,A1
MPY    .M1x   A0,B6,A9
SUB    .S1x   A8,B2,A8
|| ADD  .L1x   A0,B0,A2
```

|| **Means that the SUB and ADD belong to the same fetch packet, therefore execute simultaneously.**

Data Cross Path Limitations

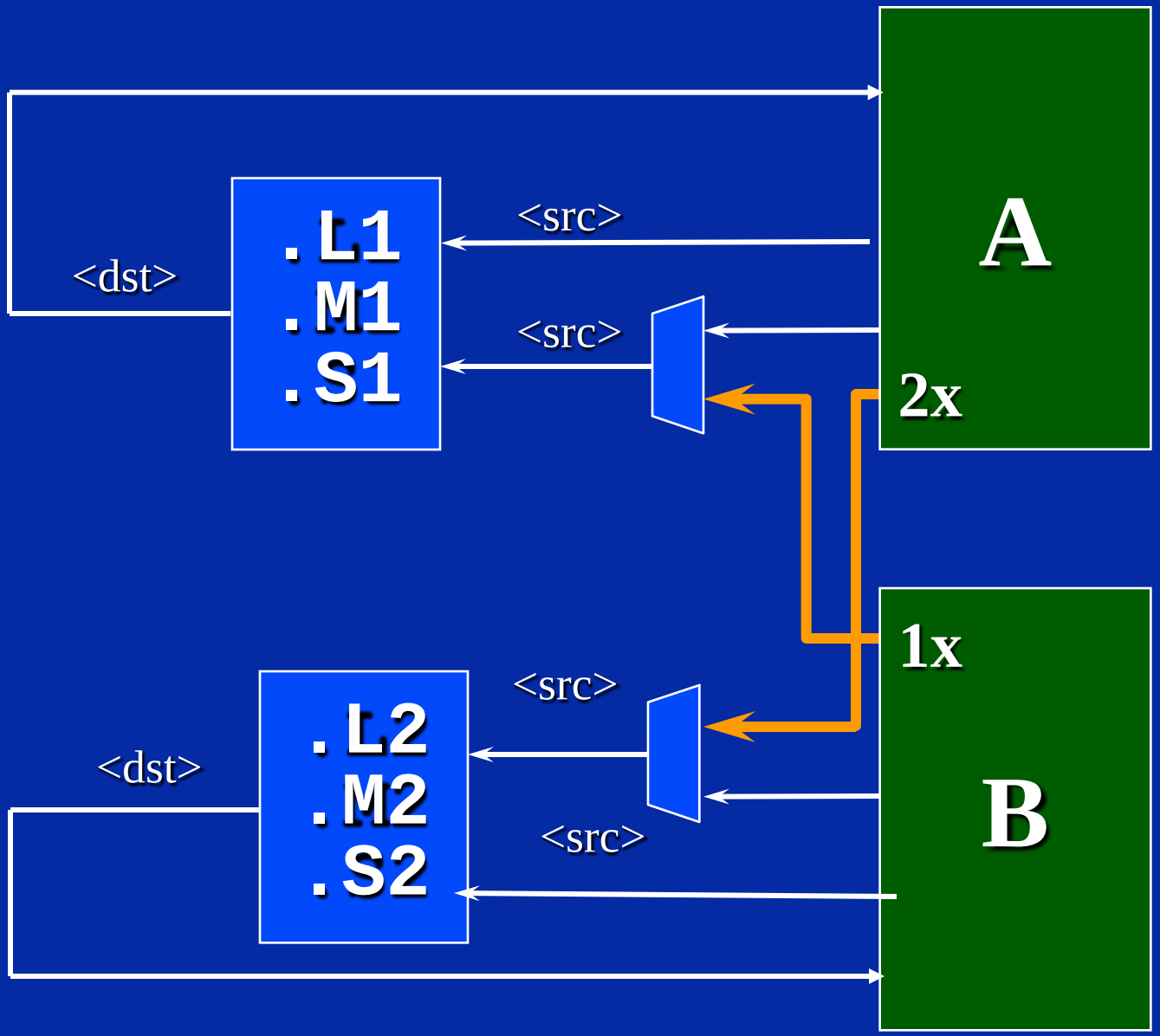


eg:

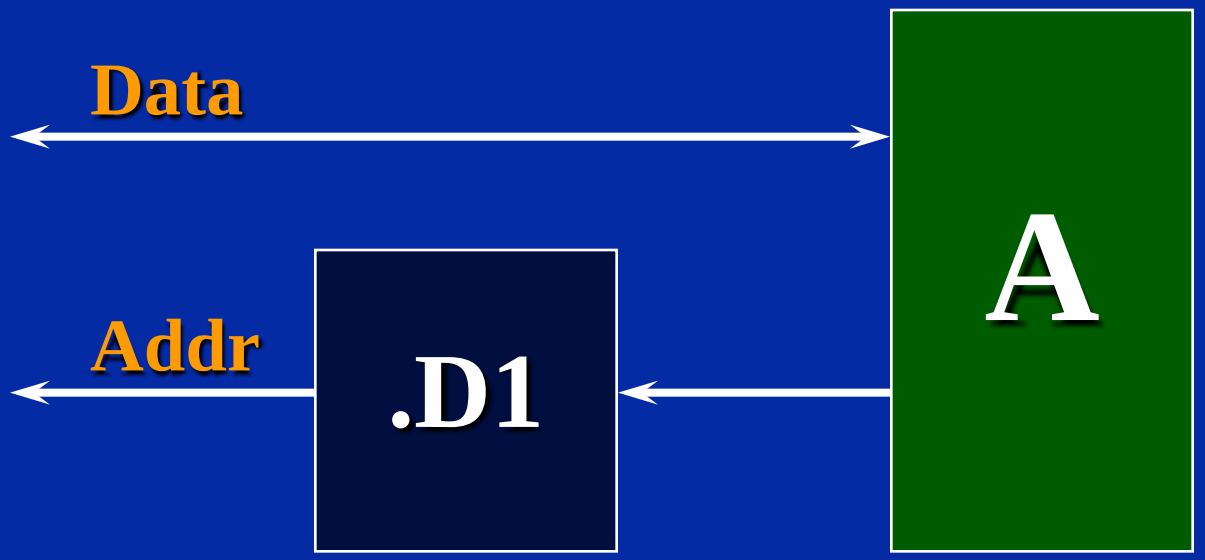
```
    ADD    .L1x   A0,B2,A1
    MPY    .M1x   A0,B6,A9
    SUB    .S1x   A8,B2,A8
||  ADD    .L1x   A0,B0,A2
```

NOT VALID!

Data Cross Paths for both sides



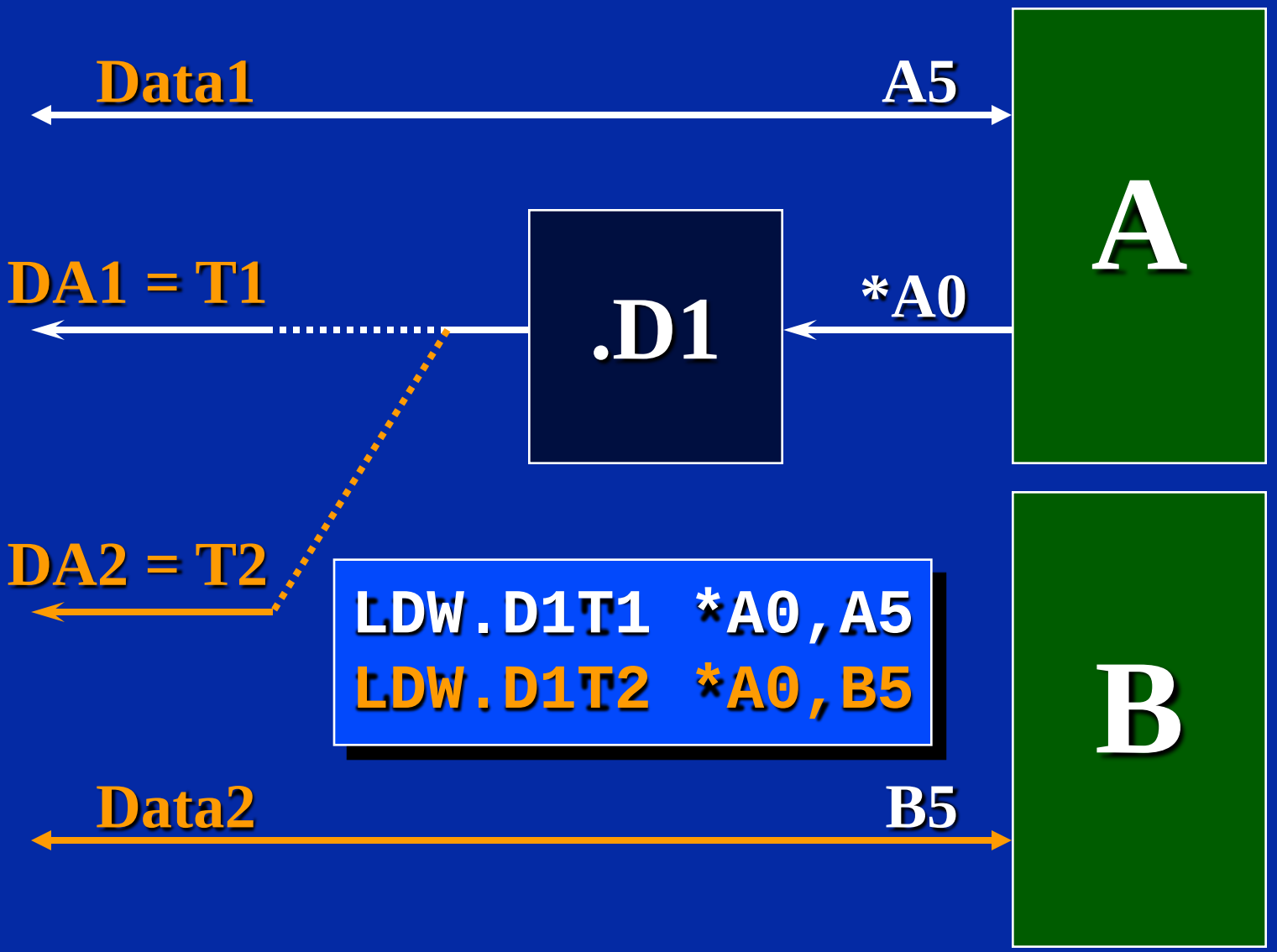
Address cross paths



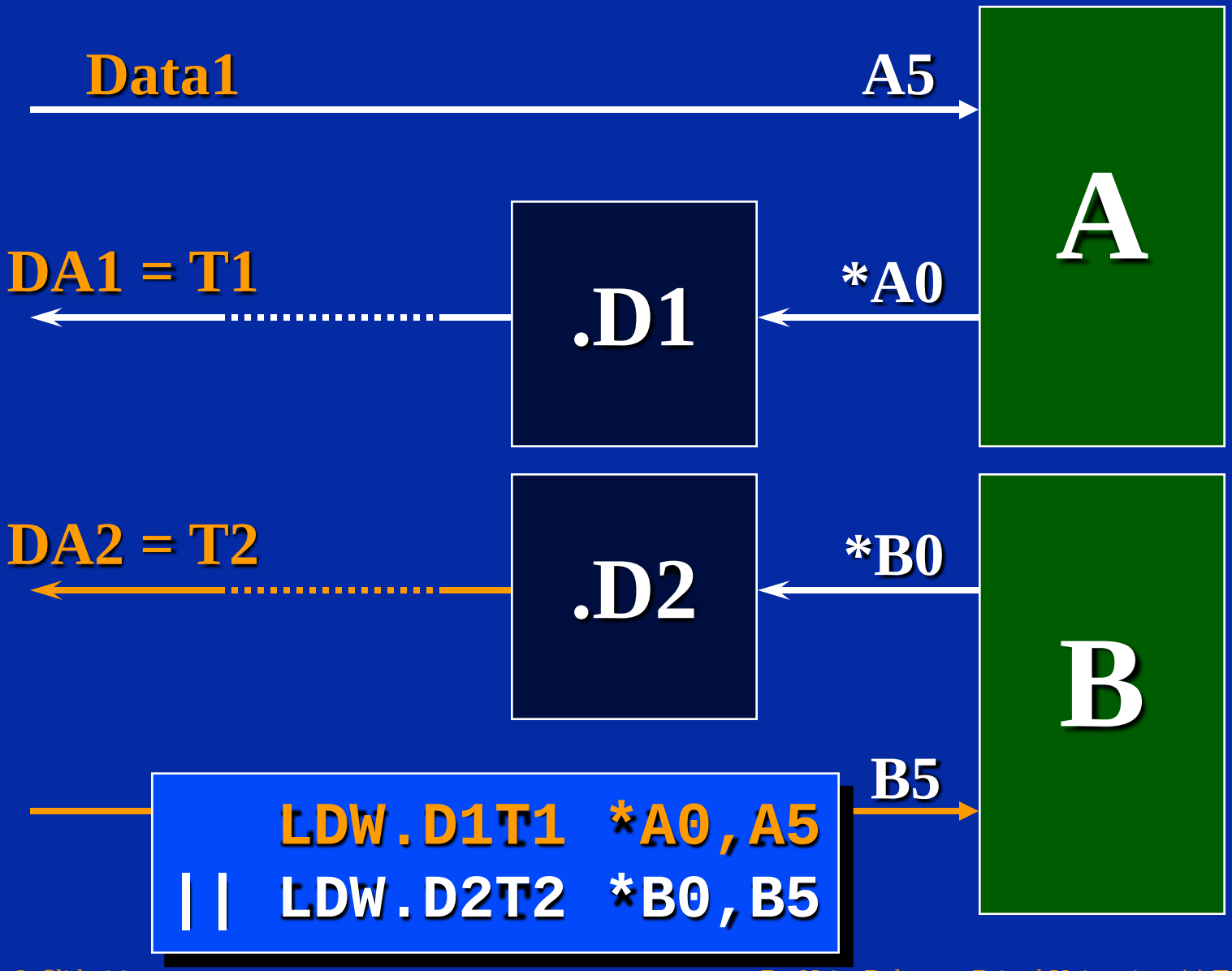
(1) The pointer must be on the same side of the unit.

```
LDW.D1T1  *A0,A5
STW.D1T1  A5,*A0
```

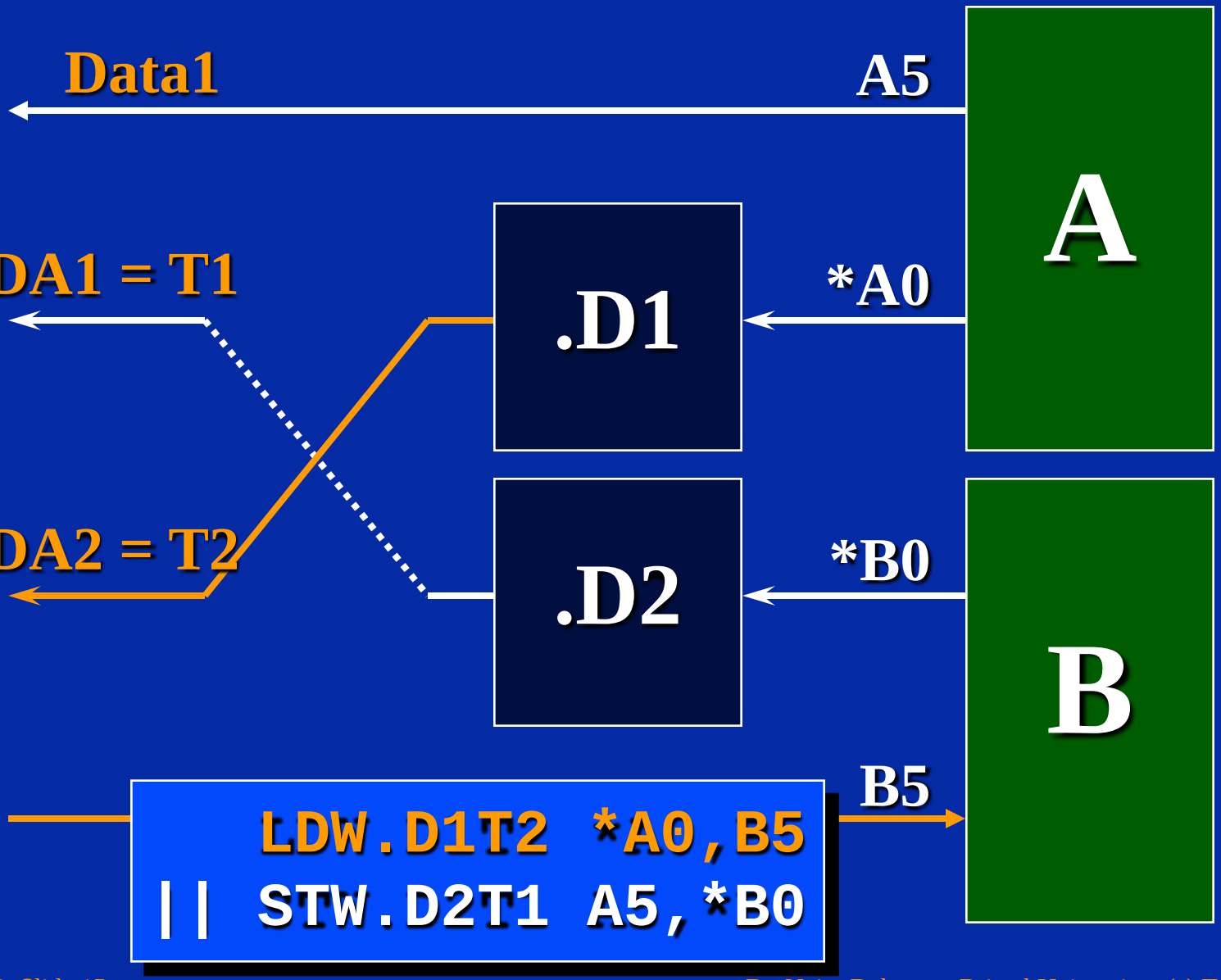
Load or store to either side



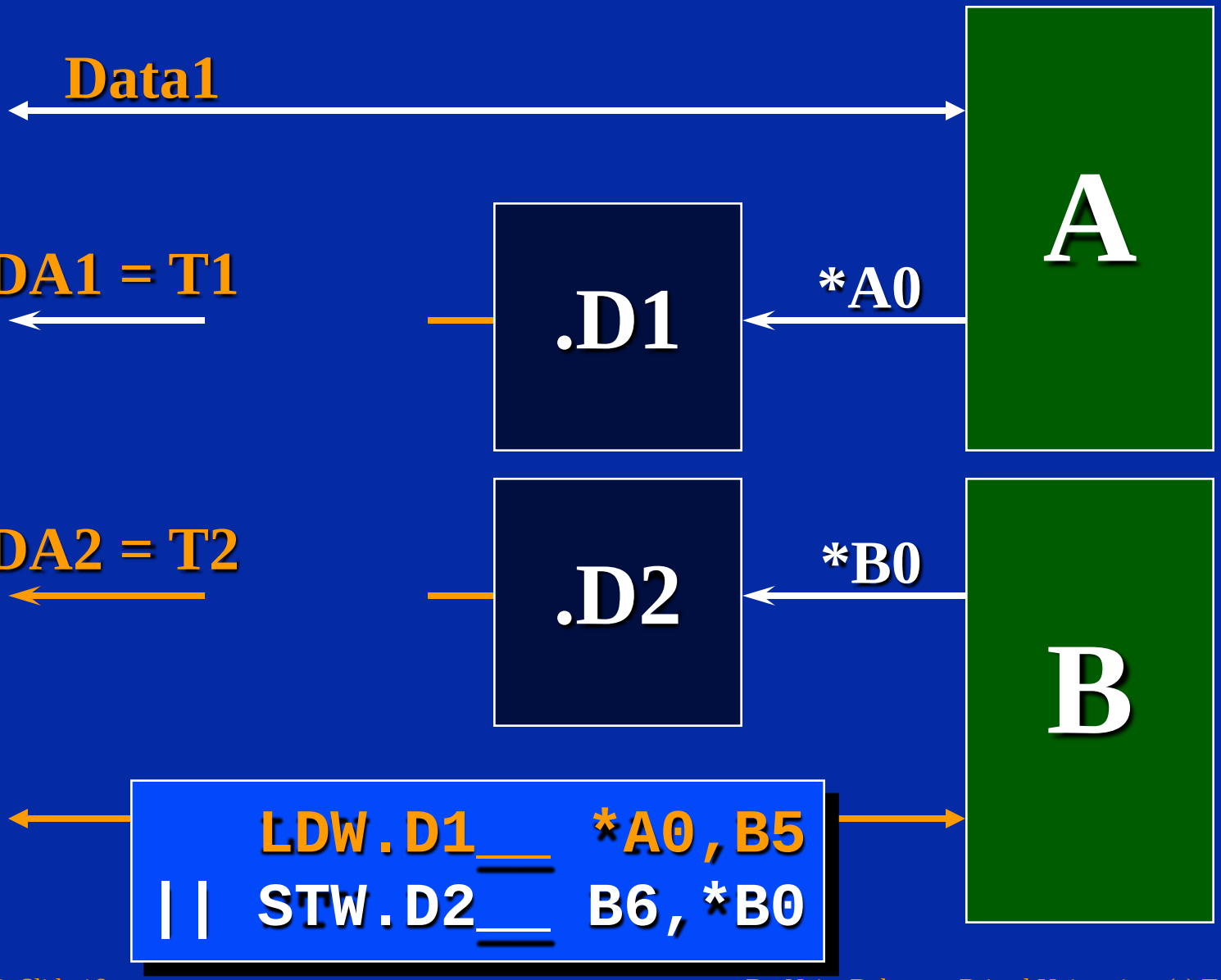
Standard Parallel Loads



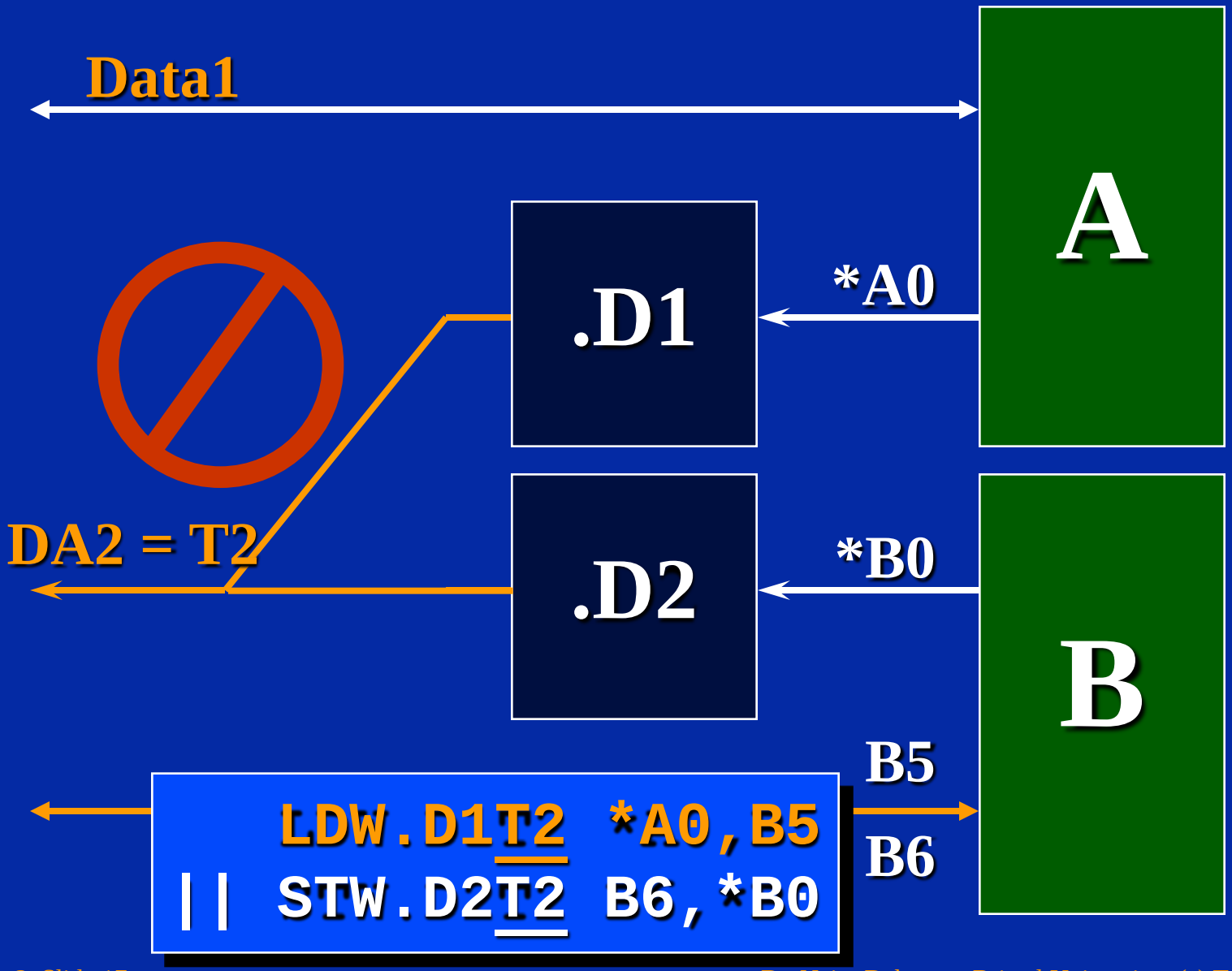
Parallel Load/Store using address cross paths



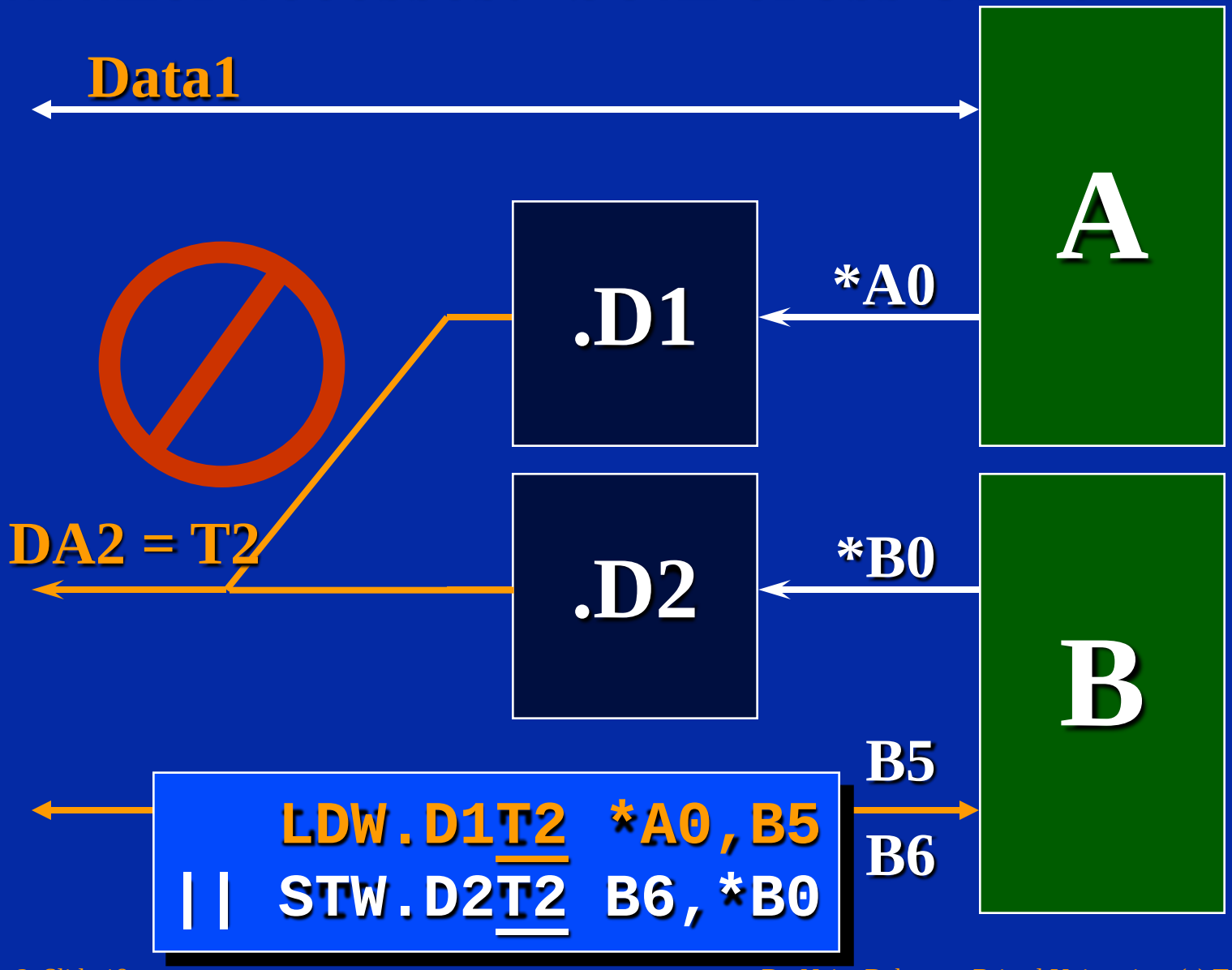
Fill the blanks ... Does this work?



Not Allowed!



Not Allowed!
Parallel accesses: both cross or neither cross

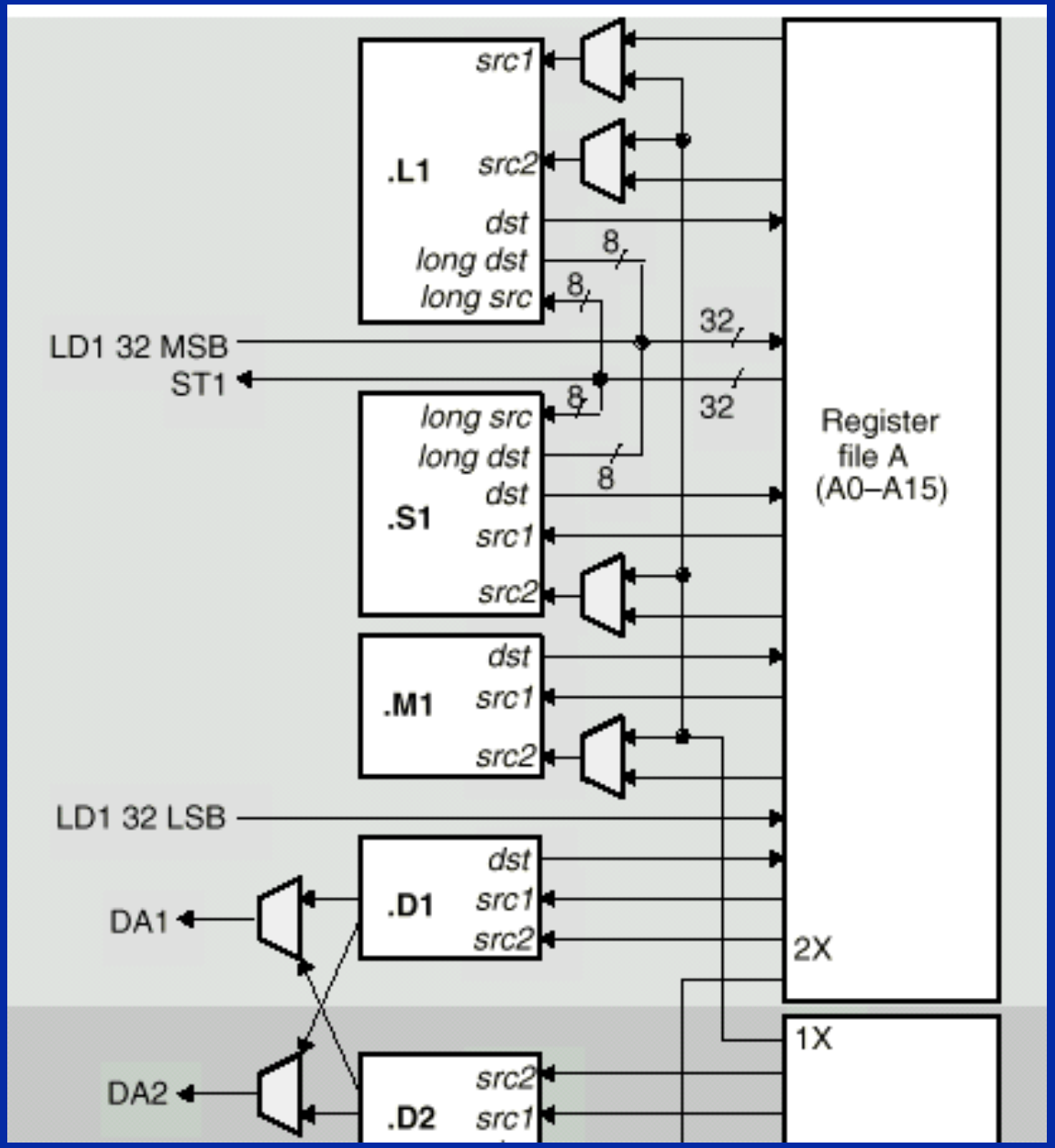


Conditions Don't Use Cross Paths

- If a conditional register comes from the opposite side, it does NOT use a data or address cross-path.
- Examples:

```
[B2]   ADD   .L1   A2,A0,A4
[A1]   LDW   .D2   *B0,B5
```

‘C67x Data-Path Summary



Cross Paths - Summary

- ✓ **Data**
  - Destination register on same side as unit.
  - Source registers - up to one cross path per execute packet per side.
  - Use “x” to indicate cross-path.
- ✓ **Address**
  - Pointer must be on same side as unit.
  - Data can be transferred to/from either side.
  - Parallel accesses: both cross or neither cross.
- ✓ **Conditionals Don’t Use Cross Paths.**

Produit de Convolution

Convolution

- x signal à N points {0..N-1}
- h signal à M points {0..M-1}
- y signal convolué à N+M-1points {0..N+M-2}
- y[n] = x[n] ** h[n]

$$Y[i] = \sum_{j=0}^{j=M-1} h[j].x[i-j]$$

- complexité: M.(N+M-1)opérations

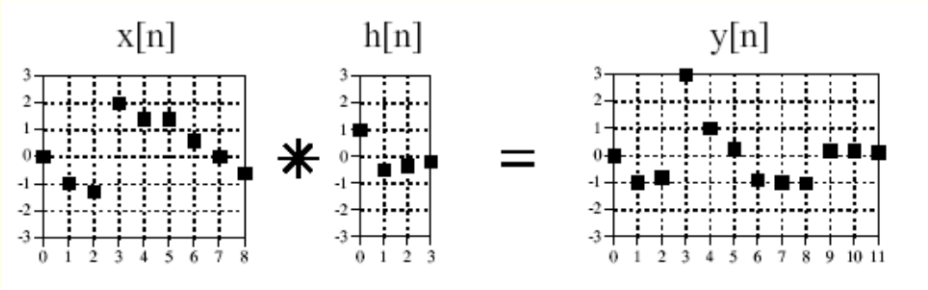


Illustration Convolution

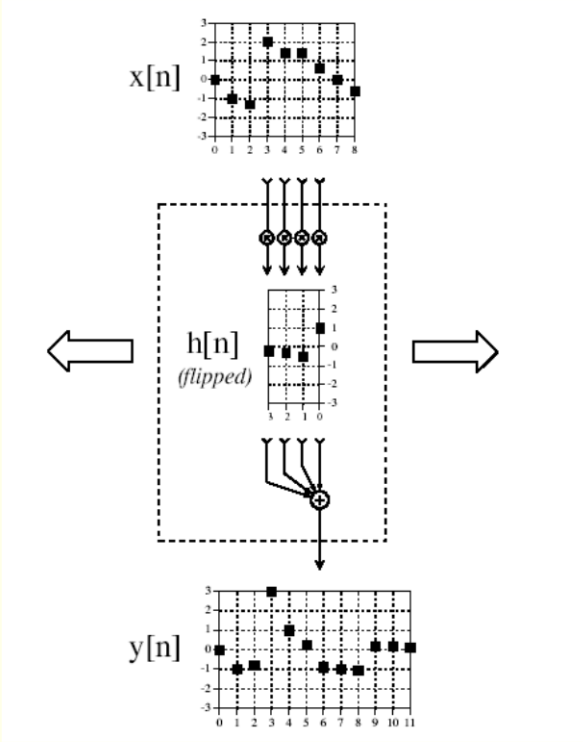


Illustration Convolution

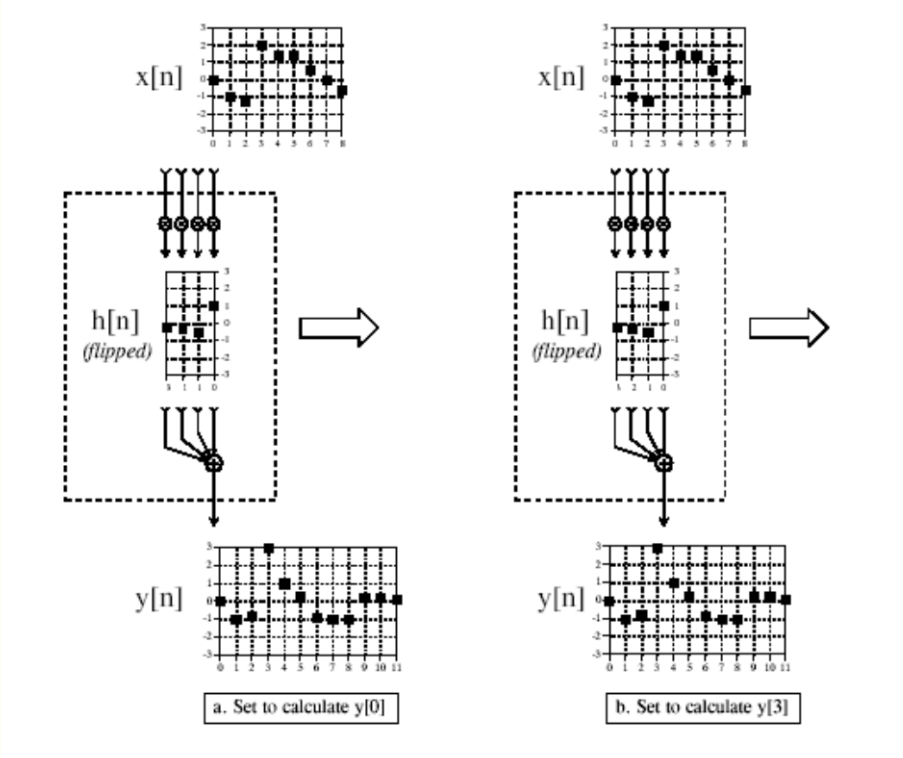
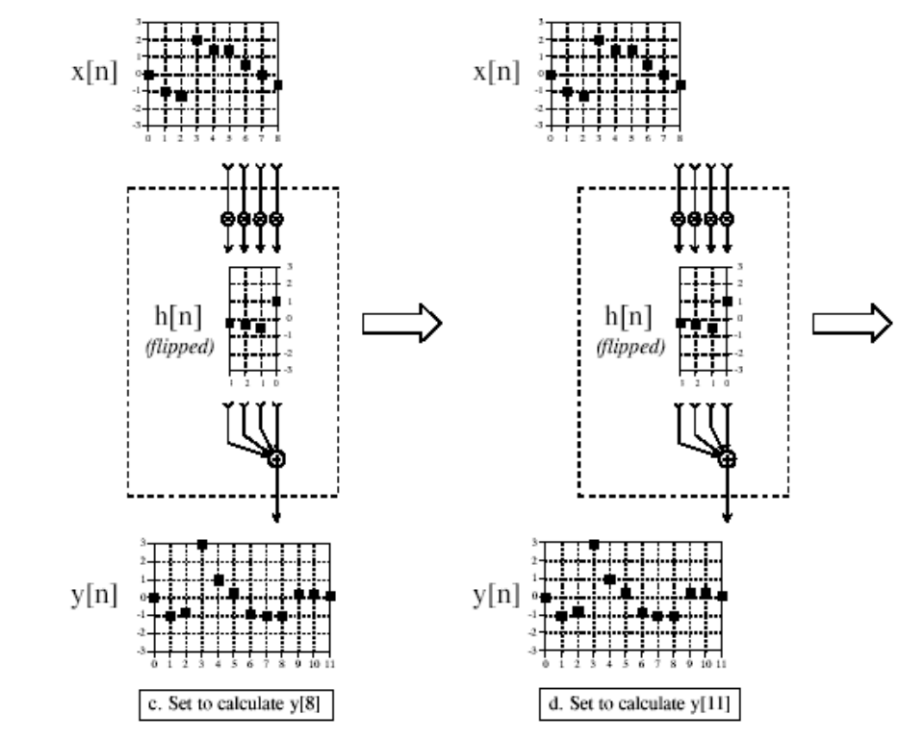


Illustration Convolution



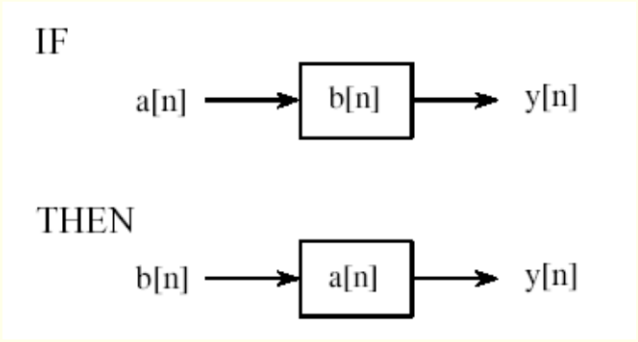
Proprietés de la Convolution

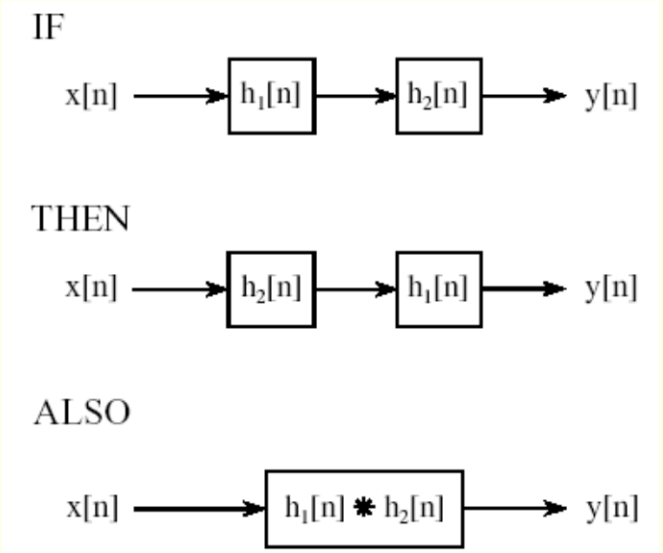
- **Commutativité**
- **a[n]**b[n] = b[n]**a[n]**
-

Associativité

(x[n]h1[n])**h2[n] = (x[n]**h2[n])**h1[n]**

= x[n](h1[n]**h2[n])**



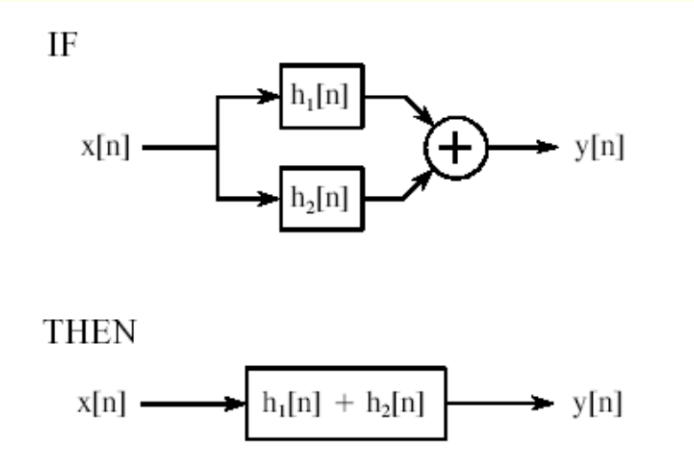


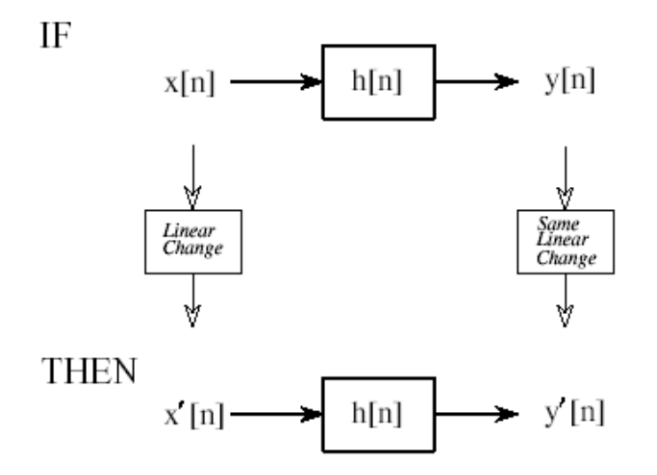
Proprietés de la Convolution

- **Distributivité**
- **(x[n]**h1[n]) + (x[n]**h2[n]) =**
- **x[n]** (h1[n] + h2[n])**

Conservation linéarité

(μx[n])h[n] = μ(x[n]**h[n])**

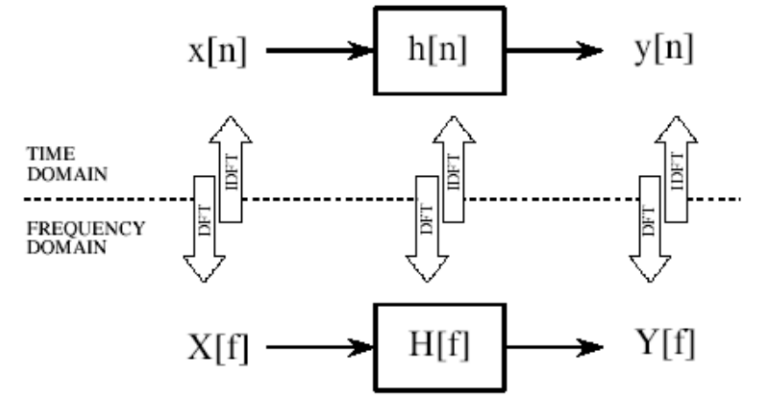




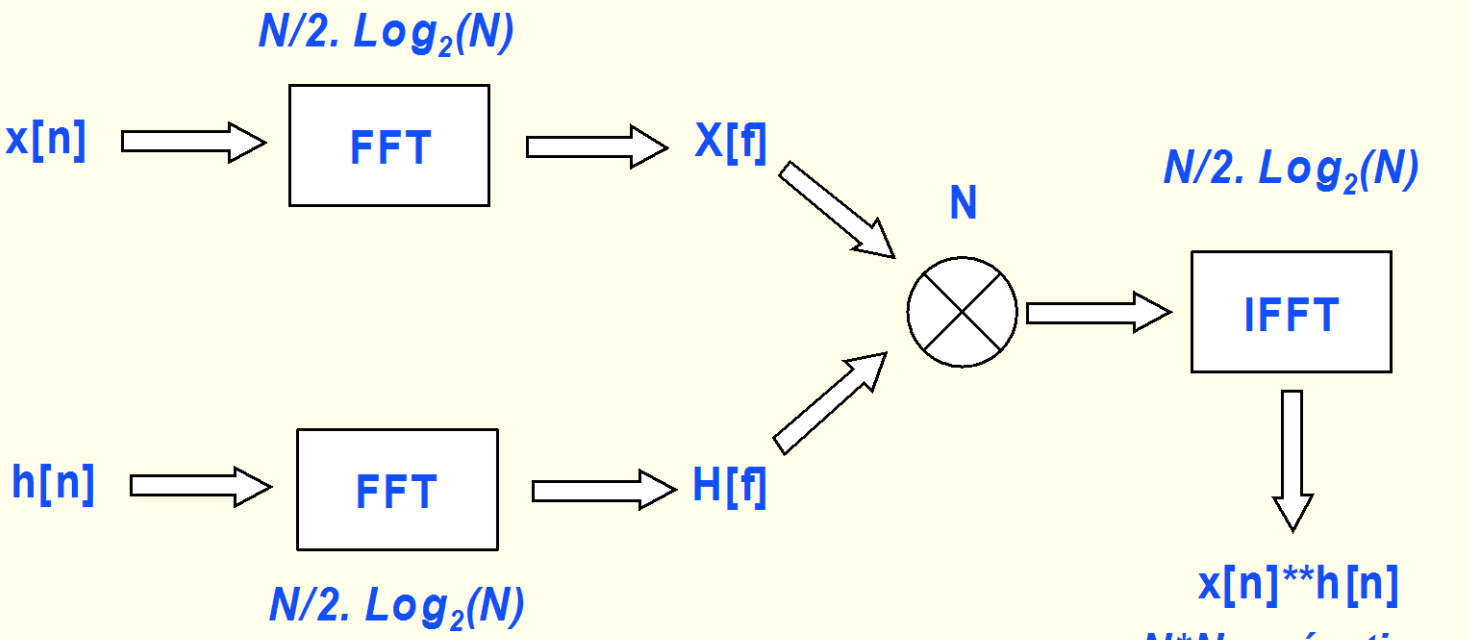
Proprietés de la Convolution

- ***Domaine Temporel*** ***Domaine Fréquentiel***

- **x[n]**h[n] ⇔ X[f]*H[f]**



Convolution par FFT



*3 * N/2. $Log_2(N)$ + N à comparer à N*N*

Projet de détection de contour

Par Convolution

-1/8	-1/8	-1/8
-1/8	1	-1/8
-1/8	-1/8	-1/8

détection de contour

0	0	0
0	1	0
0	0	-1

effet 3D

-k/8	-k/8	-k/8
-k/8	k	-k/8
-k/8	-k/8	-k/8

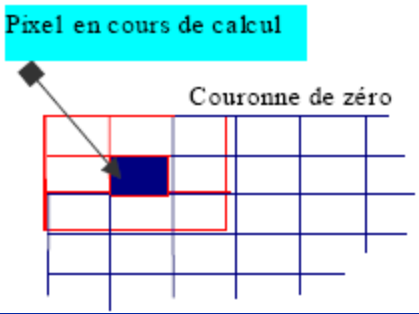
ajustement de contraste

formule mathématique:

$$Pout(i,j) = \sum_{k=0}^{3} \sum_{l=0}^{3} (H(k,l) \cdot Pin[(i-k),(j-l)])$$

Masque de convolution

-1/8	-1/8	-1/8
-1/8	1	-1/8
-1/8	-1/8	-1/8



Quelques résultats obtenues:

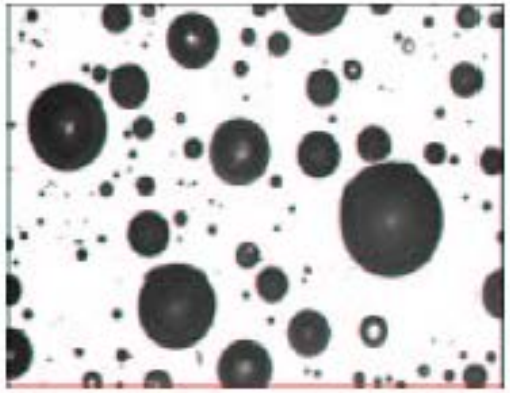
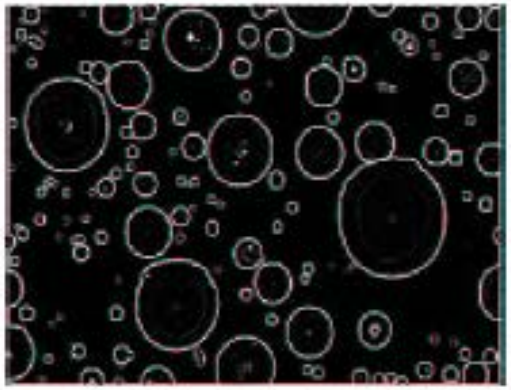
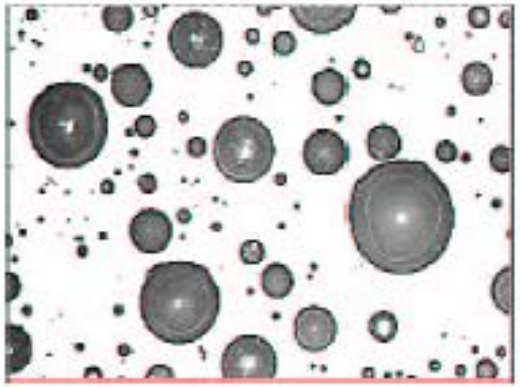


image d'origine



détection de contour



ajustement de contraste

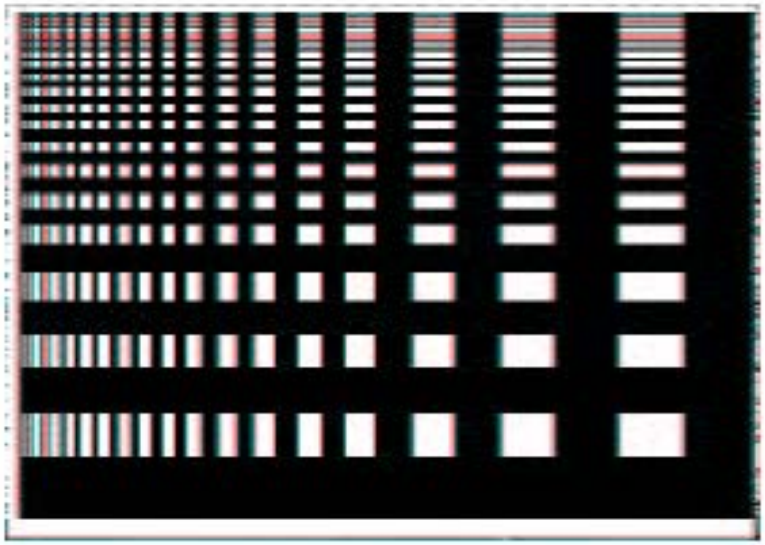
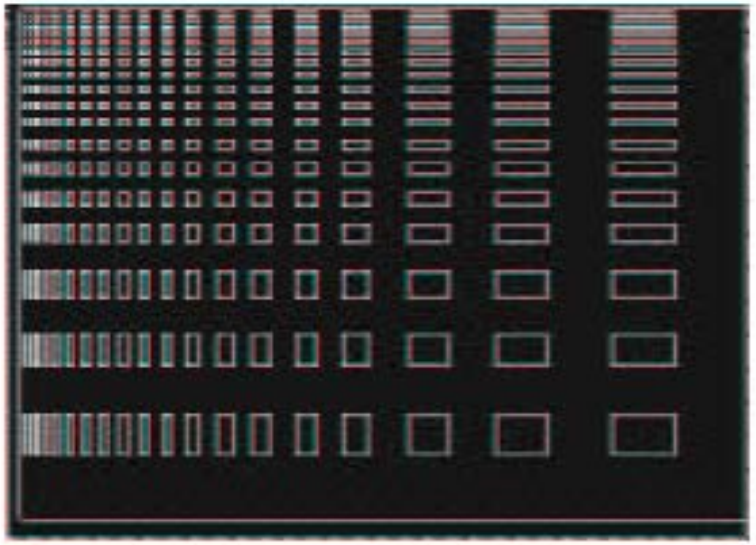


image d'origine



détection de contour

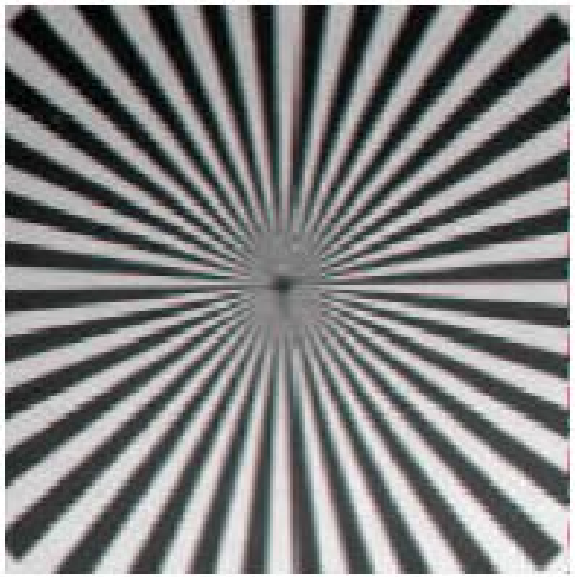
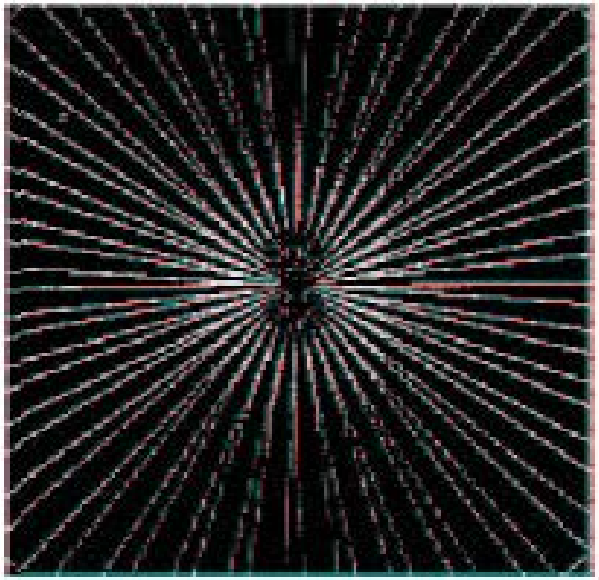


image d'origine



détéction de contour + ajustement de contraste

Objectifs du projet:

- Développer une fonction conv() avec les arguments:

  - Masque de convolution, les 9 coef

  -Les 9 pixels concernés

  -La fonction rend la valeur du produit de convolution

  - Ensuite développer un programme qui prend une image stockée

  dans un fichier, calcule le produit de convolution avec un masque

  (soit saisi au clavier soit dans un fichier) et génère un fichier

  image résultat.

  L'affichage des images se fera par les outils standards.

Image d'origine:

-Format .TIFF ou .PGMG

Format pris en charge par le CCS: .DAT

---→ Ecrire deux fonctions de conversion:

pgmg2dat() et dat2pgmg()

Pour utiliser les fonctions standards
de CCS (load et store fichier←>mémoire)

---→ Gérer soit même en C:

-le chargement Données fichier vers mémoire DSP
-la sauvegarde mémoire DSP vers fichier.

Syntax format .dat

(*MagicNumber Format StartingAddress PageNum Length*).

1.) Magic number - 1651 (fixed).

2.) Format - 1 (Hexadecimal), 2 (Integer), 3 (Long) and 4 (float).

3.) Starting Address - Starting address of the block that was saved.

4.) PageNum - Page number of the block taken from.

5.) Length - No of samples in the block.

EXEMPLE:

1651 2 0x80000000 100 100

1

5

10

4

Etc….

Format PGM (Portable Graymap file format)

Format de fichier image en niveau de gris. (.pgm)

P2 : données des pixels sont stockées en caractères (ASCII)

P5 : en binaire (RAW).

Exemple:

```
P2
# Shows the word "FEEP" (example from Netpbm main page on PGM)
24 7
15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 15 0
0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11 0 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0
0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Qui nous donne comme image :

Image PGM P2

Le fichier se compose comme suit :

- P2 est l'entête, signifie le type du fichier
- les commentaires commencent par # et vont jusqu'à la fin de la ligne
- 24 7 sont les dimensions de l'image
- 15 est la valeur maximal du gris dans l'image, cette valeur peut aller de 0 à 255
- s'en suit la valeur de chaque pixel

Le caractère d'espacement entre chaque paramètre peut être un espace, une tabulation ou un retour à la ligne (exception faite des commentaires).