DSAP

# Multiplicación matriz-matriz por bloques: algoritmo de Cannon

Consideremos la matrices *A* y *B* particionadas en *r*x*r* bloques de idéntico tamaño.

Supongamos que disponemos de una malla abierta de *r*x*r* procesadores:

$$A = \begin{bmatrix} A_{00} & A_{01} & \cdots & A_{0,r-1} \\ A_{10} & A_{11} & \cdots & A_{1,r-1} \\ \vdots & \vdots & & \vdots \\ A_{r-1,0} & A_{r-1,1} & \cdots & A_{r-1,r-1} \end{bmatrix}$$

$$B = \begin{bmatrix} B_{00} & B_{01} & \cdots & B_{0,r-1} \\ B_{10} & B_{11} & \cdots & B_{1,r-1} \\ \vdots & \vdots & & \vdots \\ B_{r-1,0} & B_{r-1,1} & \cdots & B_{r-1,r-1} \end{bmatrix}$$

$$\begin{array}{cccc} P_{00} & P_{01} & \cdots & P_{0s} \\ P_{10} & P_{11} & \cdots & P_{1s} \\ \vdots & \vdots & & \vdots \\ P_{s0} & P_{s1} & \cdots & P_{ss} \end{array}$$

*s=r-1*

# Multiplicación matriz-matriz por bloques: algoritmo de Cannon

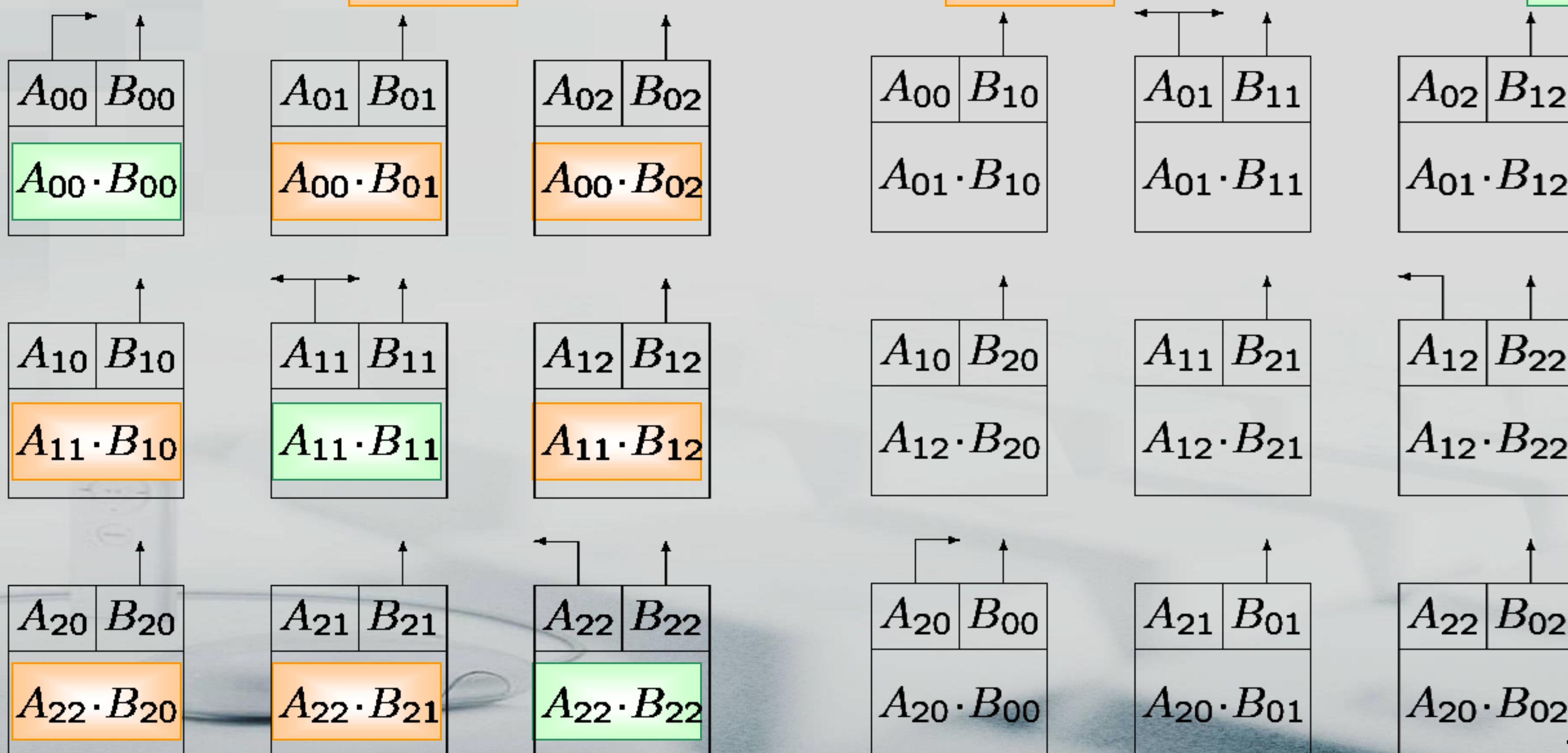
$$\begin{bmatrix} A_{00} & A_{01} & A_{02} \\ A_{10} & A_{11} & A_{12} \\ A_{20} & A_{21} & A_{22} \end{bmatrix} \times \begin{bmatrix} B_{00} & B_{01} & B_{02} \\ B_{10} & B_{11} & B_{12} \\ B_{20} & B_{21} & B_{22} \end{bmatrix} =$$

$$\begin{bmatrix} A_{00}B_{00} + A_{01}B_{10} + A_{02}B_{20} & A_{00}B_{01} + A_{01}B_{11} + A_{02}B_{21} & A_{00}B_{02} + A_{01}B_{12} + A_{02}B_{22} \\ A_{10}B_{00} + A_{11}B_{10} + A_{12}B_{20} & A_{10}B_{01} + A_{11}B_{11} + A_{12}B_{21} & A_{10}B_{02} + A_{11}B_{12} + A_{12}B_{22} \\ A_{20}B_{00} + A_{21}B_{10} + A_{22}B_{20} & A_{20}B_{01} + A_{21}B_{11} + A_{22}B_{21} & A_{20}B_{02} + A_{21}B_{12} + A_{22}B_{22} \end{bmatrix}$$

DSAP

# Multiplicación matriz-matriz por bloques: algoritmo de Cannon

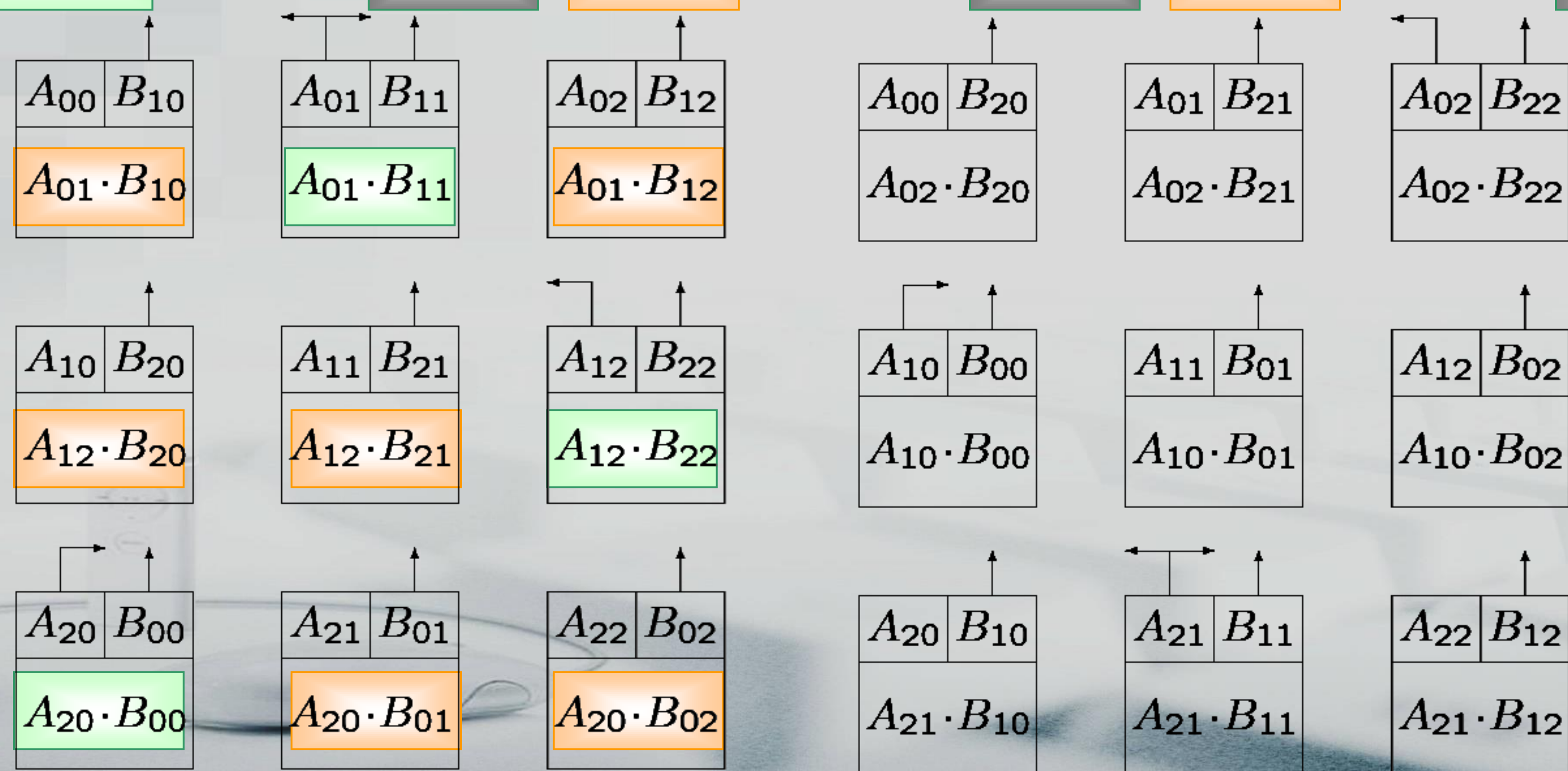
$$
\begin{bmatrix}
\boxed{A_{00}B_{00}} + A_{01}B_{10} + A_{02}B_{20} & \boxed{A_{00}B_{01}} + A_{01}B_{11} + A_{02}B_{21} & \boxed{A_{00}B_{02}} + A_{01}B_{12} + A_{02}B_{22} \\
A_{10}B_{00} + \boxed{A_{11}B_{10}} + A_{12}B_{20} & A_{10}B_{01} + \boxed{A_{11}B_{11}} + A_{12}B_{21} & A_{10}B_{02} + \boxed{A_{11}B_{12}} + A_{12}B_{22} \\
A_{20}B_{00} + A_{21}B_{10} + \boxed{A_{22}B_{20}} & A_{20}B_{01} + A_{21}B_{11} + \boxed{A_{22}B_{21}} & A_{20}B_{02} + A_{21}B_{12} + \boxed{A_{22}B_{22}}
\end{bmatrix}
$$



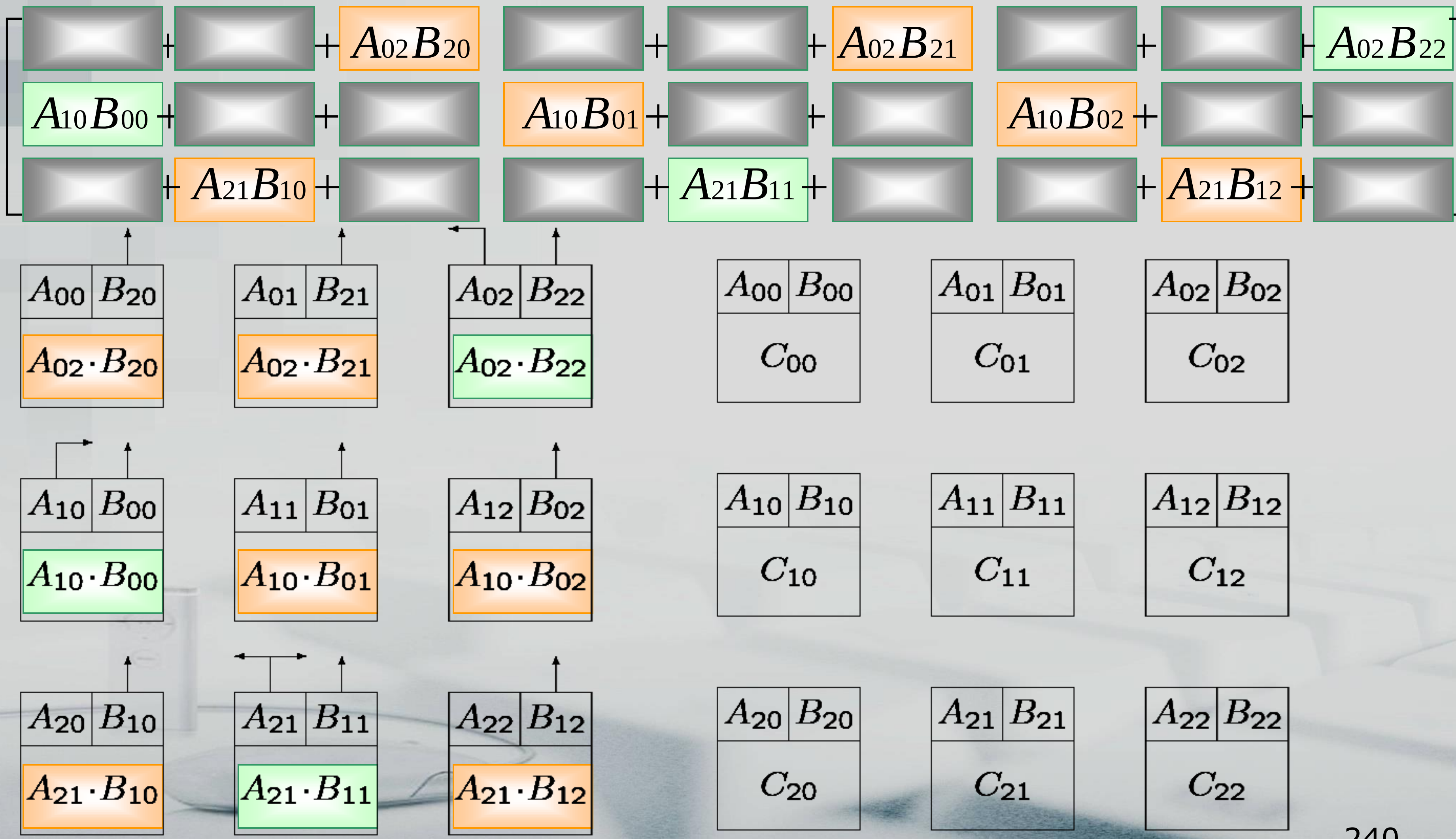
DSAP

# Multiplicación matriz-matriz por bloques: algoritmo de Cannon

$$\begin{bmatrix} \square + A_{01}B_{10} + A_{02}B_{20} & \square + A_{01}B_{11} + A_{02}B_{21} & \square + A_{01}B_{12} + A_{02}B_{22} \\ A_{10}B_{00} + \square + A_{12}B_{20} & A_{10}B_{01} + \square + A_{12}B_{21} & A_{10}B_{02} + \square + A_{12}B_{22} \\ A_{20}B_{00} + A_{21}B_{10} + \square & A_{20}B_{01} + A_{21}B_{11} + \square & A_{20}B_{02} + A_{21}B_{12} + \square \end{bmatrix}$$



# Multiplicación matriz-matriz por bloques: algoritmo de Cannon



DSAP

# Multiplicación matriz-matriz por bloques: algoritmo de Cannon

## ALGORITMO

Inicialmente $P_{ij}$ contiene $A_{ij}$ y $B_{ij}$. Cada procesador calculará el bloque correspondiente $C_{ij}$ del producto $C = AB$. Internamente estos bloques tendrán el mismo nombre, es decir

$$\mathcal{A} \leftarrow A_{ij} \quad \mathcal{B} \leftarrow B_{ij} \quad \mathcal{C} \leftarrow C_{ij}$$

Cada procesador $P_{ij}$ estará identificado por su *fila* $(= i)$ y su *columna* $(= j)$.

DSAP

# Multiplicación matriz-matriz por bloques: algoritmo de Cannon

Para $k = 0, 1, \ldots, r-1$

— Si columna=mod(fila+$k$,r):
Mandar $\mathcal{A}$ a todos los procesadores de mi fila.
$\mathcal{C} = \mathcal{C} + \mathcal{A} \star \mathcal{B}$

Si no:
Recibir $\mathcal{A}$ del procesador que envía en mi fila y almacenarlo en $\mathcal{ATMP}$.
$\mathcal{C} = \mathcal{C} + \mathcal{ATMP} \star \mathcal{B}$

— Hacer una rotación de los bloques columna de $B$, es decir, $P_{ij}$ manda su bloque $\mathcal{B}$ a $P_{i-1,j}$. El proceso $P_{0j}$ manda su bloque $\mathcal{B}$ a $P_{r-1,j}$.

Después de estas *r* iteraciones, *C* contiene el producto *AB* distribuido entre los procesadores y los bloques de *B* han sufrido una rotación entre las columnas, volviendo a estar almacenados como al inicio del algoritmo.

DSAP

# Multiplicación matriz-matriz por bloques: algoritmo de Cannon

r=3

| | | | | |
|---|---|---|---|---|
| k=0 | fila | 0 | 1 | 2 |
| | mod(fila+0,3) | 0 | 1 | 2 |
| | (fila,columna) | (0,0) | (1,1) | (2,2) |

| | | | | |
|---|---|---|---|---|
| k=1 | fila | 0 | 1 | 2 |
| | mod(fila+1,3) | 1 | 2 | 0 |
| | (fila,columna) | (0,1) | (1,2) | (2,0) |

| | | | | |
|---|---|---|---|---|
| k=2 | Fila | 0 | 1 | 2 |
| | mod(fila+2,3) | 2 | 0 | 1 |
| | (fila,columna) | (0,2) | (1,0) | (2,1) |