

Trabalho 3 – Sistemas Digitais – 2016/1

# Projeto Multiplicação de Matrizes

Do fluxograma ASM ao VHDL  
PC-PO e implementação na placa

# Multiplicação de matrizes

Matrizes A e B 4x4 cada com dados de 4 bits

$$\begin{bmatrix} \boxed{\begin{matrix} A00 & A01 & A02 & A03 \\ A10 & A11 & A12 & A13 \\ A20 & A21 & A22 & A23 \\ A30 & A31 & A32 & A33 \end{matrix}} & \times & \begin{bmatrix} \boxed{\begin{matrix} B00 & B01 & B02 & B03 \\ B10 & B11 & B12 & B13 \\ B20 & B21 & B22 & B23 \\ B30 & B31 & B32 & B33 \end{matrix}} \end{bmatrix} = \begin{bmatrix} \boxed{A00 * B00 + A01*B10 + \dots A03*B30\dots} \\ \dots \\ \boxed{A10 * B00 + A11*B10 + \dots A33*B33\dots} \end{bmatrix}$$

# Exemplo do algoritmo MxM

- ```
programa multiplica_matrizes;  
  matriz mat1, mat2, mat3;  
  inteiro linha, coluna, i, acumula;  
  "leia mat1";  
  "leia mat2";  
  "verifique se mat1 é compatível com mat2";  
  para linha de 1 até "numero de linhas de mat1" faça  
    para coluna de 1 até "numero de colunas de mat2" faça  
      acumula=0;  
      para i de 1 até "numero de colunas de mat1" faça  
        acumula=acumula+mat1[linha][i]*mat2[i][coluna];  
      fimpara;  
      mat3[linha][coluna]=acumula;  
    fimpara;  
  fimpara;  
  imprima mat3;  
fim programa;
```

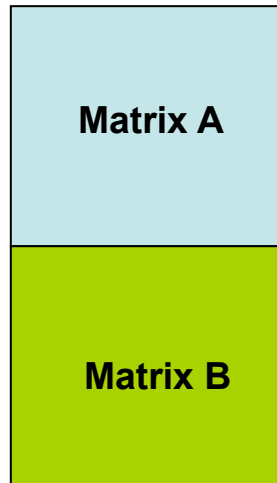
# Dados de entrada

- Cada matriz tem 16 dados de 4 bits que podem ser organizados em:
  - 16 endereços de memória com 1 dado de 4-bits em cada end.ou seguindo outra organização de memória, como por exemplo armazenando toda uma linha e coluna da matriz em um unico endereço.

Memoria pode ser single ou dual port para possibilitar a leitura da memoria A e memoria B em paralelo e/ou pode-se usar mais de uma memoria...

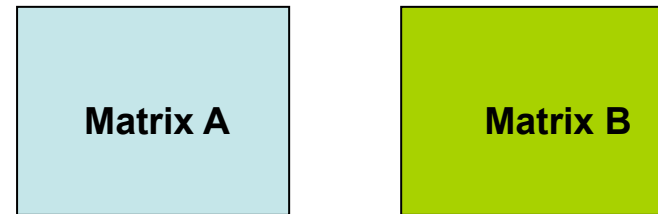
# Organização dos dados de entrada

1 BRAM



Dual port

2 BRAM



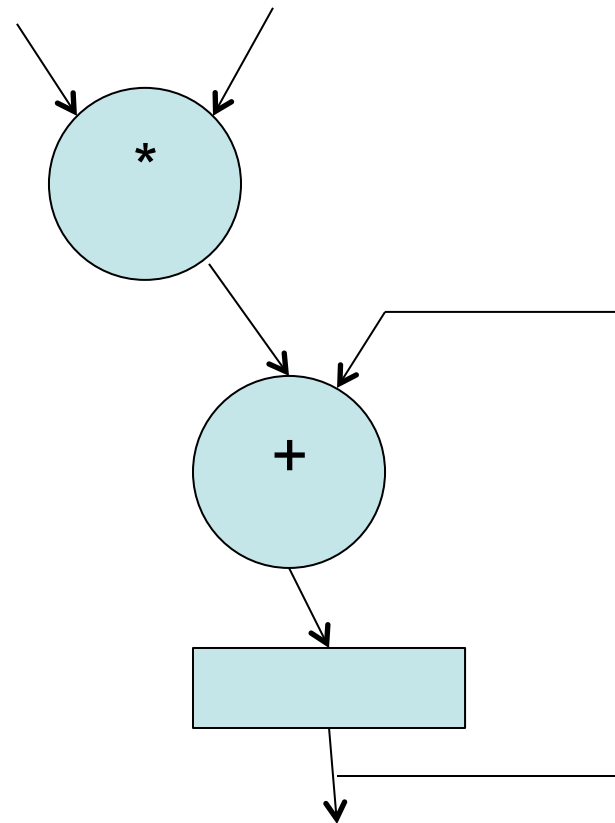
Single port

Ou outra organização que preferirem.

# Projeto PC-PO

Processamento do Datapath seguindo  
uma abordagem mais sequencial

$$\begin{aligned} &A_{00} * B_{00} \\ &+ \\ &A_{01} * B_{10} \\ &+ \\ &A_{02} * B_{20} \\ &+ \\ &A_{03} * B_{30} \\ &+ \\ &A_{04} * B_{40} \\ &+ \\ &A_{05} * B_{50} \\ &+ \\ &A_{06} * B_{60} \\ &+ \\ &A_{07} * B_{70} \\ &+ \\ &A_{08} * B_{80} \\ &+ \\ &A_{09} * B_{90} \end{aligned}$$



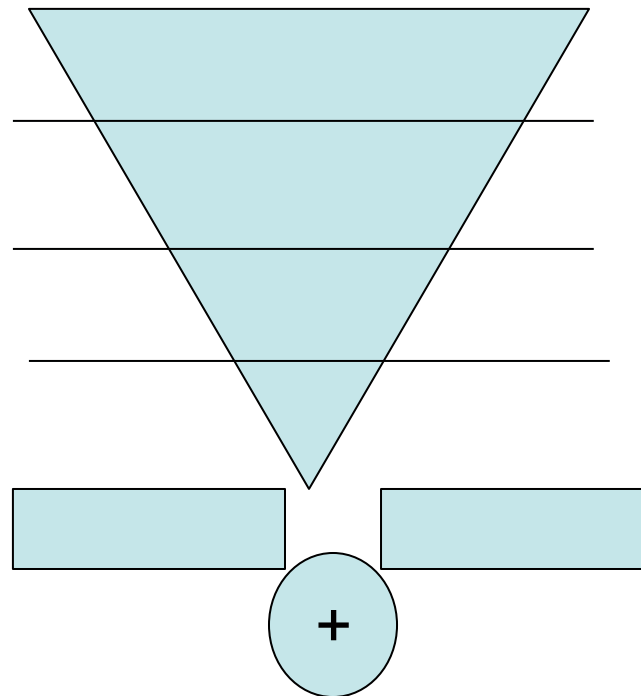
1 multiplicadores  
1 somadores

# Projeto PC-PO

Processamento do Datapath seguindo uma abordagem mais paralela e com possibilidade de inserir pipeline.

$$\begin{aligned} &A_{00} * B_{00} + A_{01} * B_{10} + A_{02} * B_{20} + A_{03} * B_{30} + A_{04} * B_{40} \\ &+ \\ &A_{05} * B_{50} + A_{06} * B_{60} + A_{07} * B_{70} + A_{08} * B_{80} + A_{09} * B_{90} \end{aligned}$$

pipeline



5 multiplicadores  
4/5 somadores

# Metricas a serem apresentadas

- Area
- Desempenho em Frequencia
- Desempenho em tempo de resposta



# Apresentação

Deve conter:

- 1) Fluxograma ASM do algoritmo usado
- 2) Desenho da arquitetura PO-PC
- 3) Simulações sem e com atraso
- 4) Metricas
- 5) Video que mostra a implementação na placa com uso de interfaces a escolher.

# Pontuação

- Apresentação e qualidade da mesma (2 pontos)
  - Fluxograma ASM (1 ponto)
  - VHDL funcionando na simulação sem atraso (4 pontos)
  - Simulação com atraso funcionando (1 ponto)
  - Implementação na placa (2 pontos)
- 
- Grupo com menor area +1 ponto
  - Grupo com menor tempo de execução +1 ponto