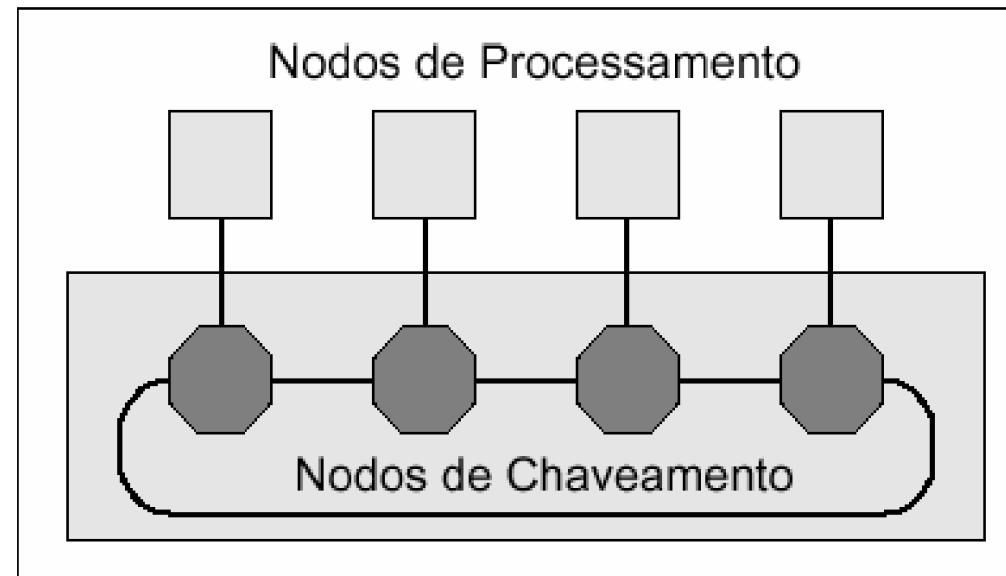


NoC

# NoC

Network-on-Chip (Redes Intra-Chip)

- Composta por nodos de processamento e nodos de chaveamento

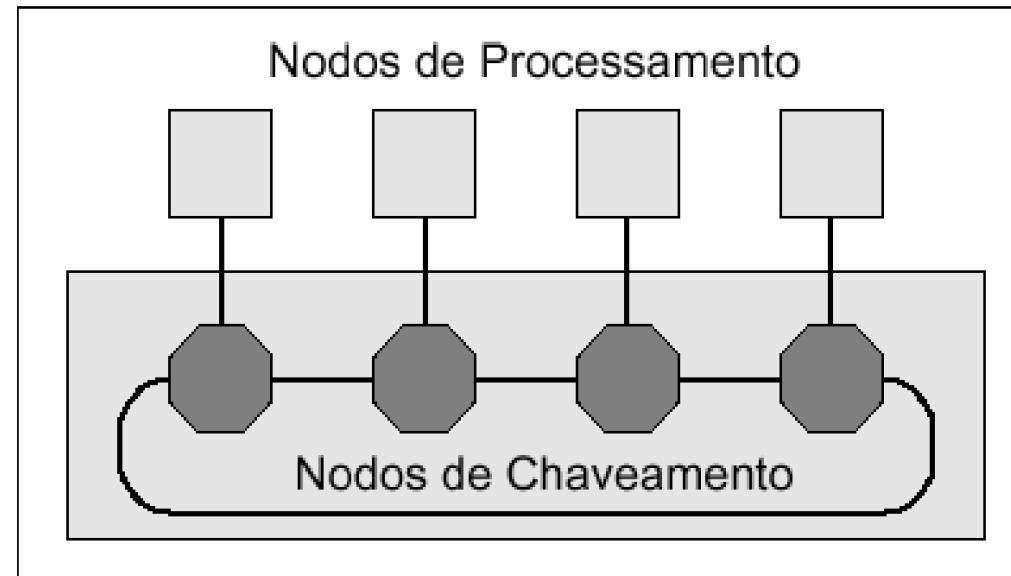


# NoC

Network-on-Chip (Redes Intra-Chip)

- Composta por nodos de processamento e nodos de chaveamento

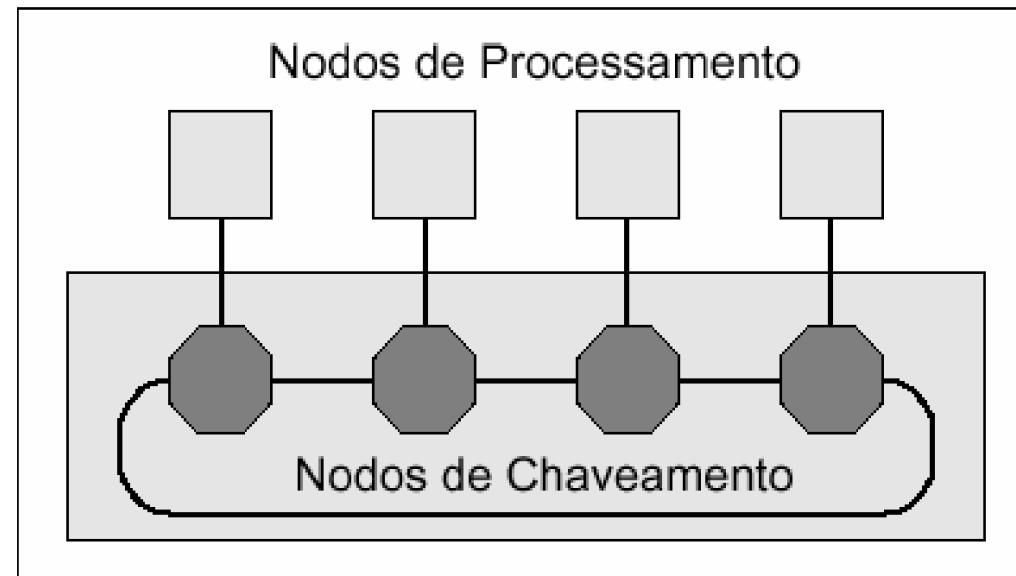
Rede em anel



# NoC

Network-on-Chip (Redes Intra-Chip)

- Composta por nodos de processamento e nodos de chaveamento



Rede em anel

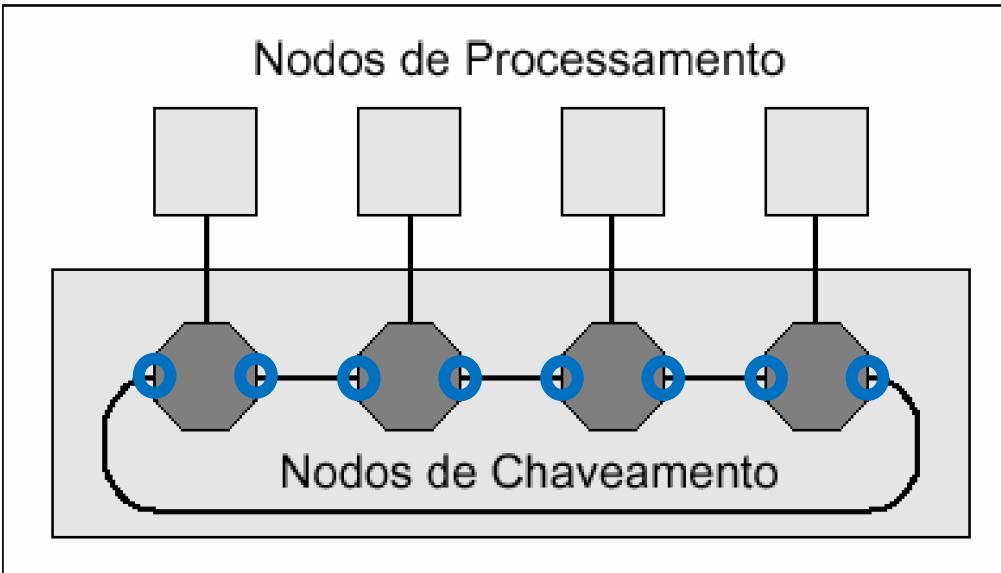
Topologia simples e  
barata

# NoC

Network-on-Chip (Redes Intra-Chip)

- Composta por nodos de processamento e nodos de chaveamento

Cada nodo de chaveamento possui ligação para outros 2 nodos de chaveamento vizinhos



Rede em anel

Topologia simples e barata

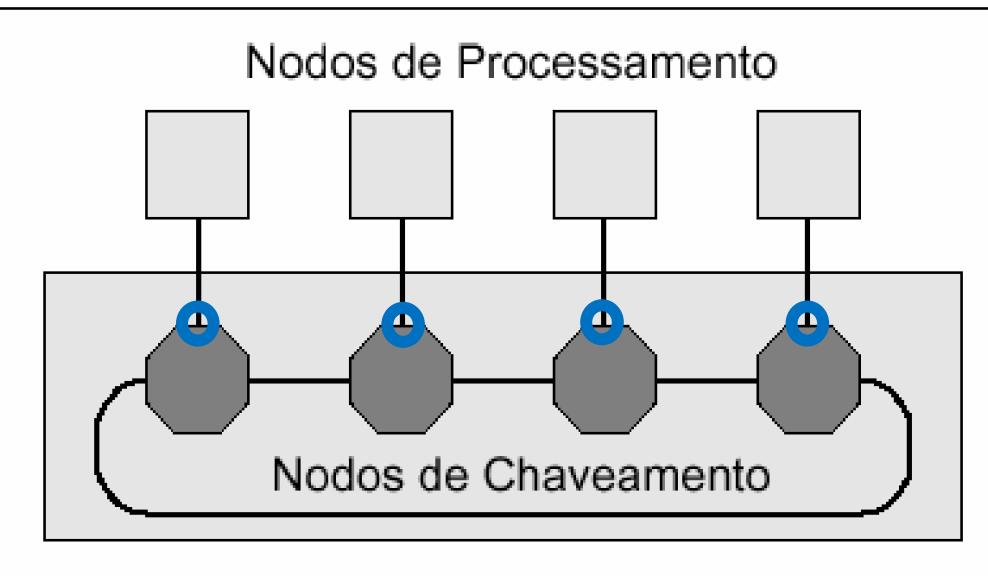
# NoC

## Network-on-Chip (Redes Intra-Chip)

- Composta por nodos de processamento e nodos de chaveamento

Cada nodo de chaveamento possui ligação para outros 2 nodos de chaveamento vizinhos

Cada nodo de chaveamento possui ligação para 1 nodo de processamento



Rede em anel

Topologia simples e barata

# NoC

## Network-on-Chip (Redes Intra-Chip)

- Composta por nodos de processamento e nodos de chaveamento
  - Os nodos de processamento são responsáveis pela execução das tarefas do sistema
  - Os nodos de chaveamento (conhecidos como roteadores) são responsáveis pela transferência dos dados entre os nodos de processamento

# NoC

## Network-on-Chip (Redes Intra-Chip)

- Os nodos de chaveamento possuem ligações físicas chamadas enlaces (links)
- Em NoCs os links são implementados com fios, utilizando uma camada de baixa resistividade (alumínio ou cobre)
- Um enlace possui um ou dois canais físicos de comunicação
  - Cada canal físico pode possuir dois ou mais canais lógicos, conhecidos como canais virtuais

# NoC

## Network-on-Chip (Redes Intra-Chip)

- Em SoCs (Sistemas Intra-Chip) a comunicação entre núcleos é implementada através do roteamento entre os módulos
- Nesta questão que topologias em NoC superam as topologias de barramento
- Em NoC as conexões são locais, com módulos de roteamento próximos
  - Isso reduz o comprimento total de roteamento
  - Aumenta o desempenho elétrico
  - Diminui chances de afetar dados com ruídos de conexões próximas
- Em barramentos, as conexões são globais, perdendo desempenho pelos seus fios longos

# NoC

---

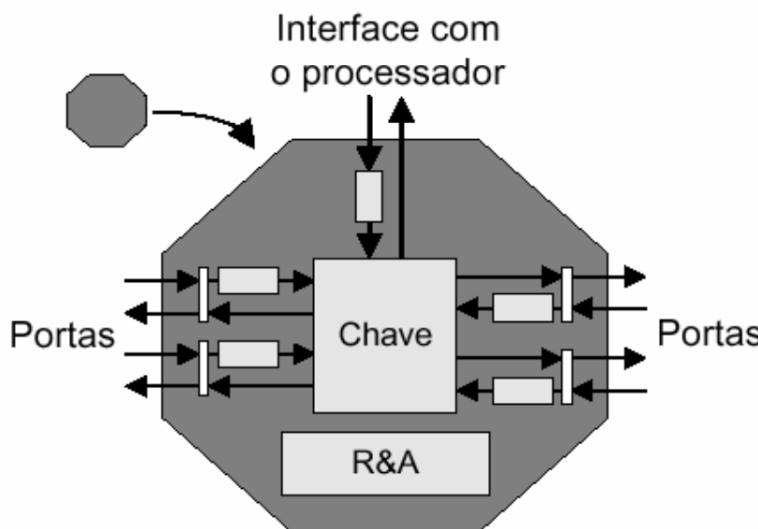
Network-on-Chip (Redes Intra-Chip)

- O nodo chaveador é o principal elemento de uma NoC

# NoC

Network-on-Chip (Redes Intra-Chip)

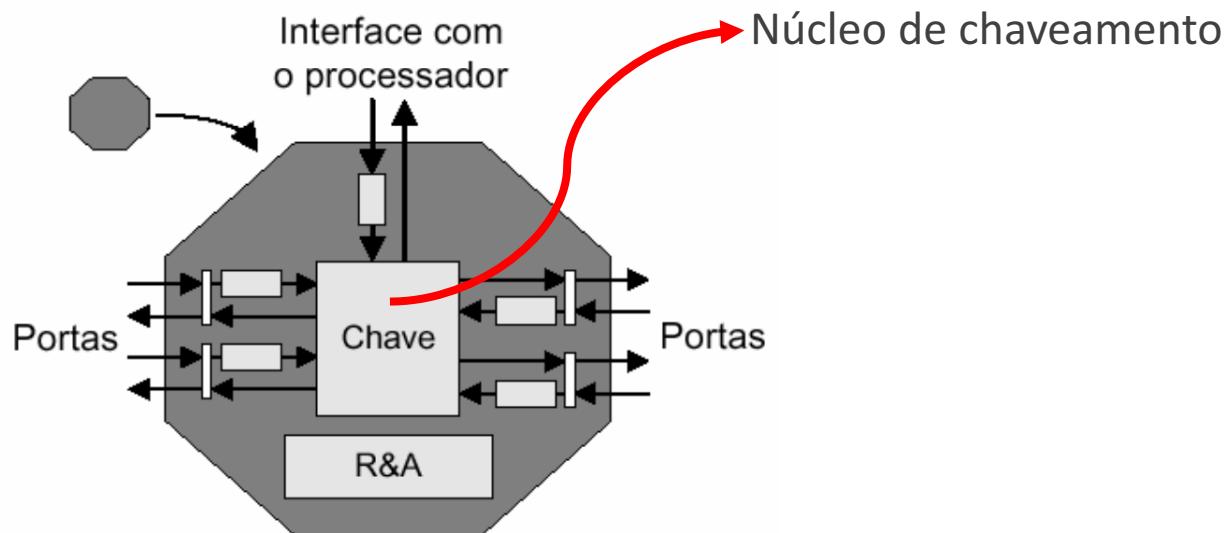
- O nodo chaveador é o principal elemento de uma NoC



# NoC

Network-on-Chip (Redes Intra-Chip)

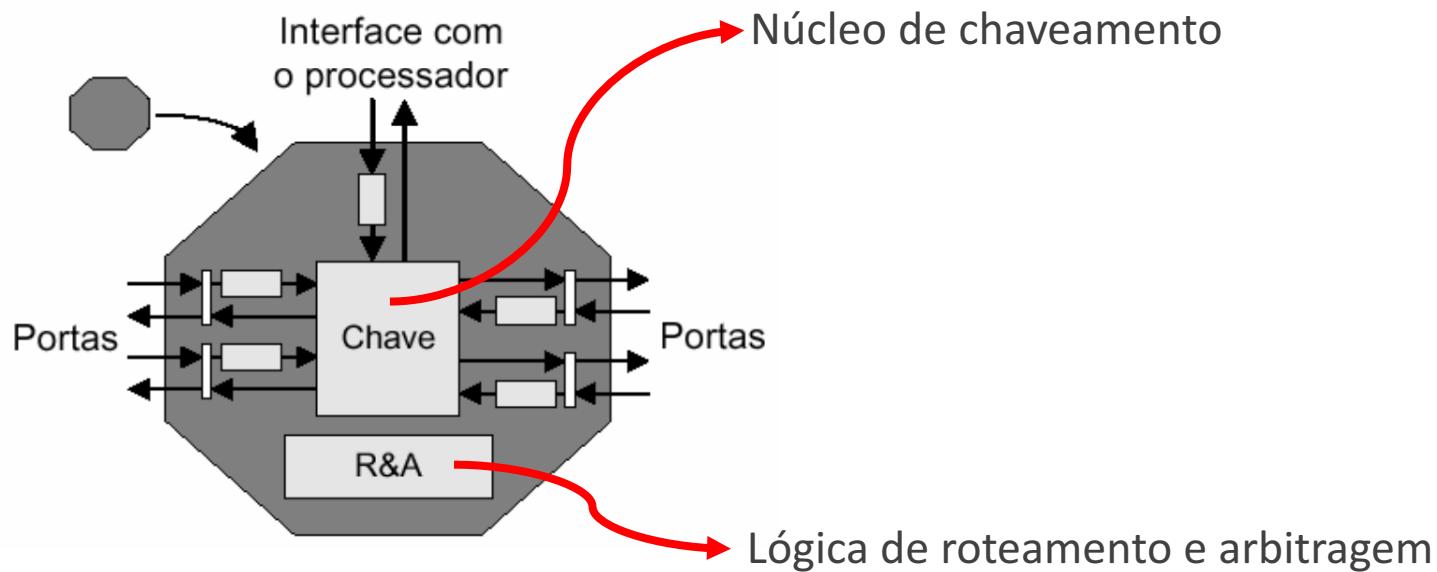
- O nodo chaveador é o principal elemento de uma NoC



# NoC

Network-on-Chip (Redes Intra-Chip)

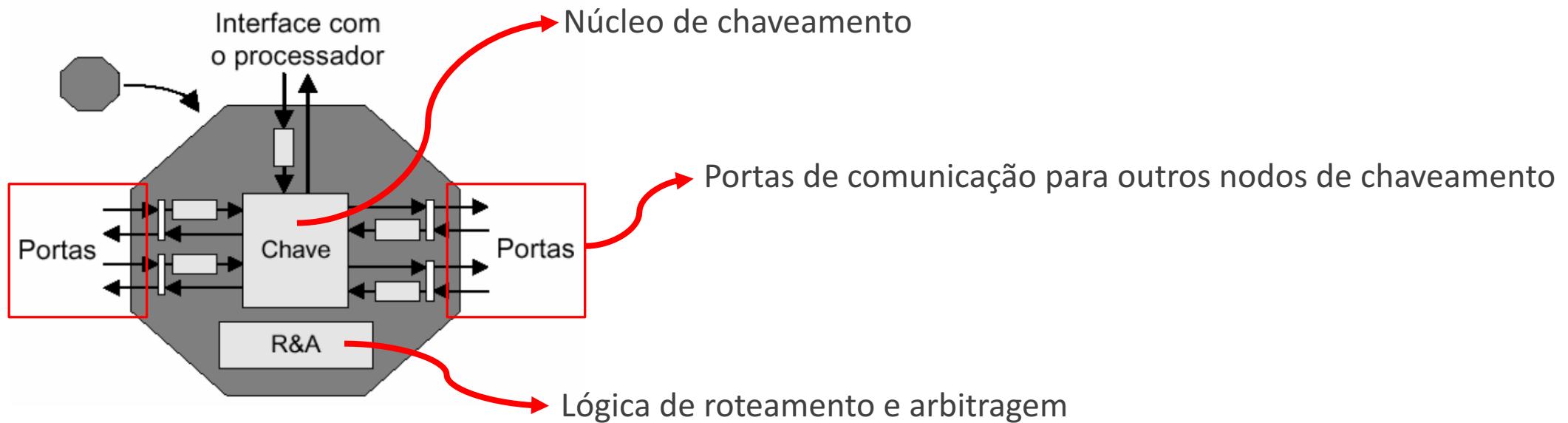
- O nodo chaveador é o principal elemento de uma NoC



# NoC

Network-on-Chip (Redes Intra-Chip)

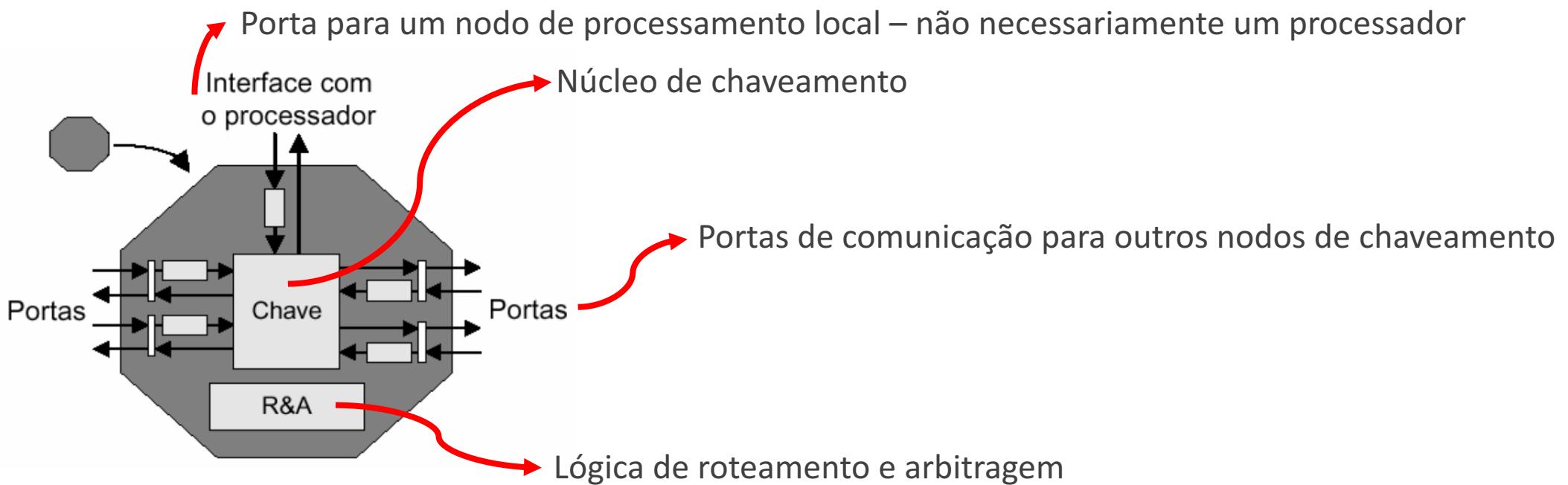
- O nodo chaveador é o principal elemento de uma NoC



# NoC

Network-on-Chip (Redes Intra-Chip)

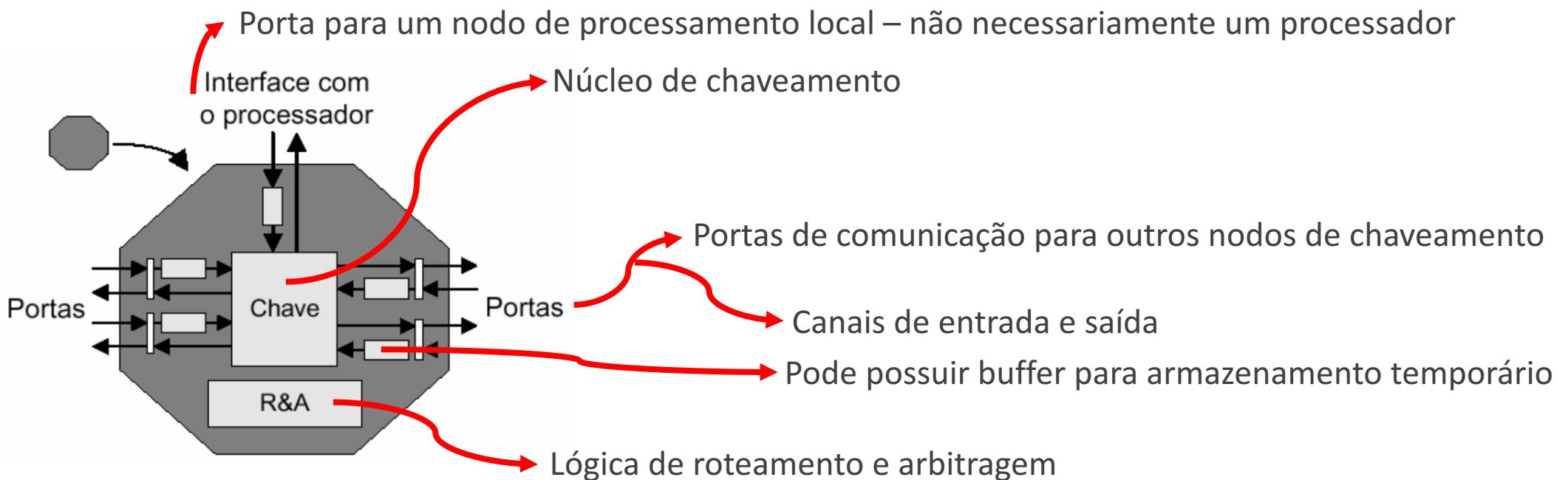
- O nodo chaveador é o principal elemento de uma NoC



# NoC

## Network-on-Chip (Redes Intra-Chip)

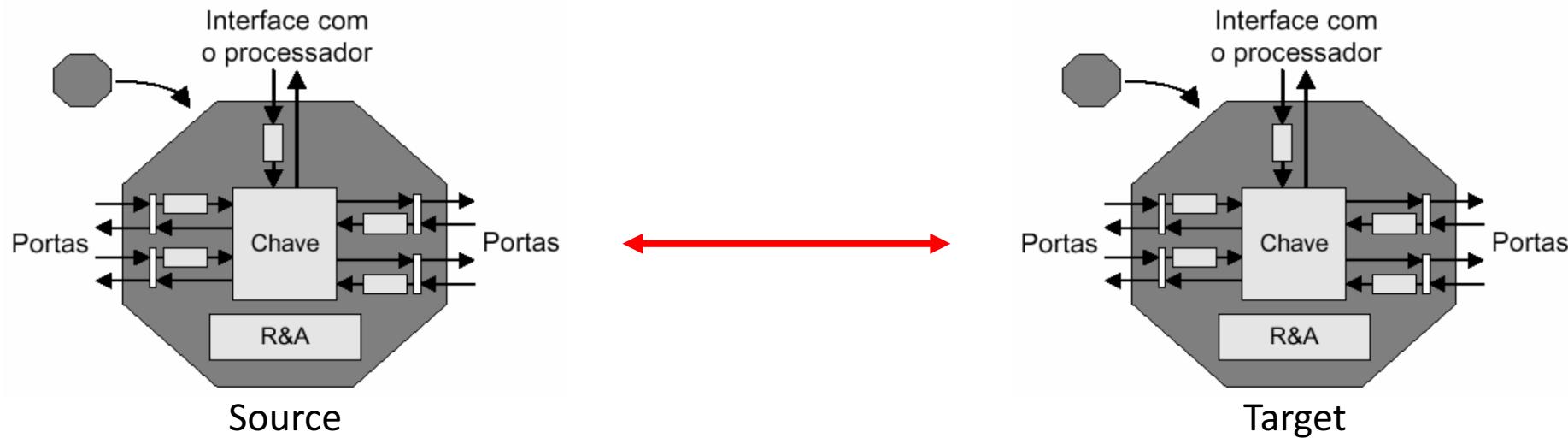
- O nodo chaveador é o principal elemento de uma NoC



# NoC

Network-on-Chip (Redes Intra-Chip)

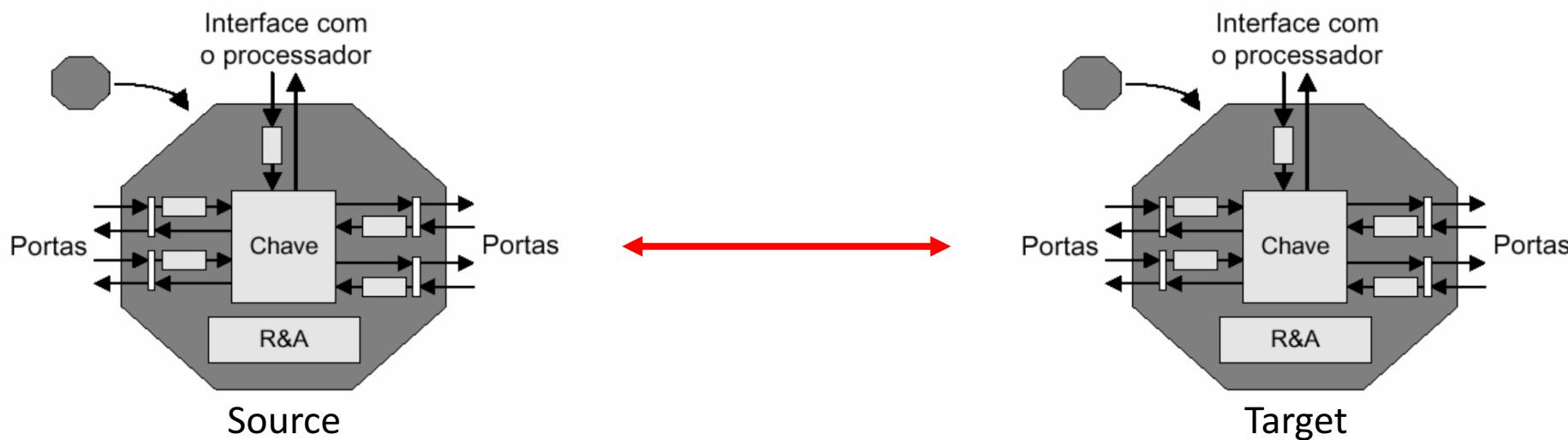
- A comunicação entre um nodo fonte (source) e um destino (target) é feita sob troca de mensagens



# NoC

## Network-on-Chip (Redes Intra-Chip)

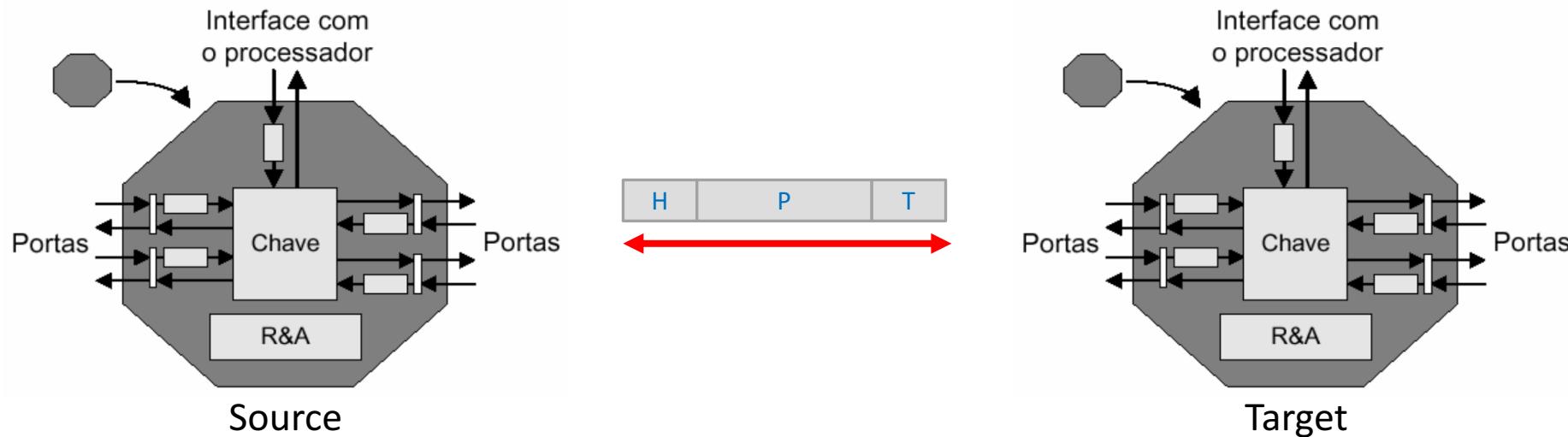
- A comunicação entre um nodo fonte (source) e um destino (target) é feita sob troca de mensagens
  - São formadas por 3 partes
    - Cabeçalho (header)
    - Corpo de dados (payload)
    - Terminador (trailer)



# NoC

## Network-on-Chip (Redes Intra-Chip)

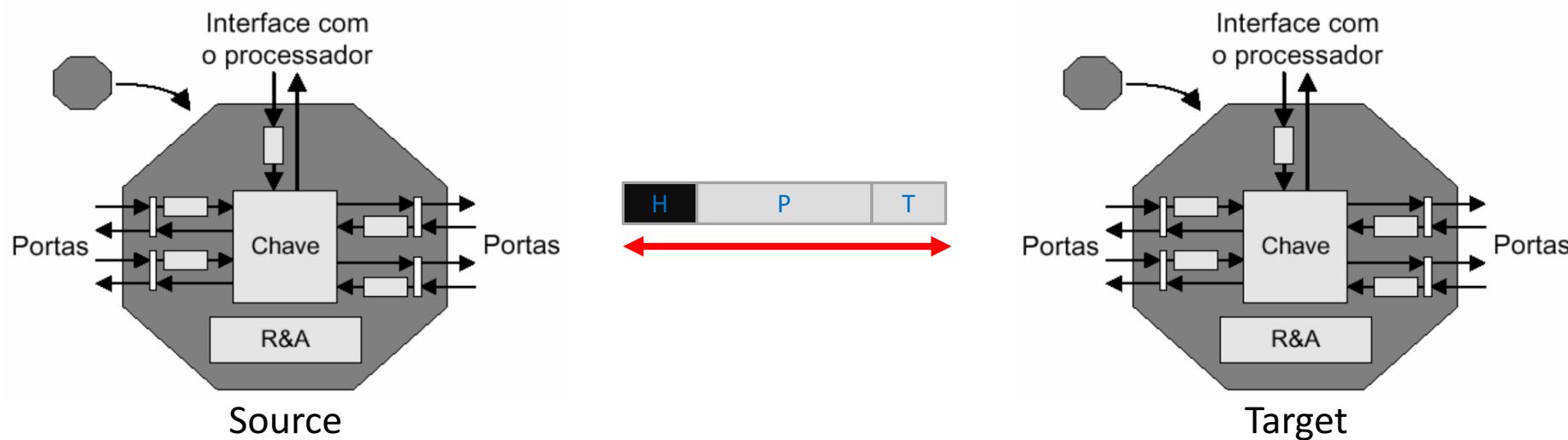
- A comunicação entre um nodo fonte (source) e um destino (target) é feita sob troca de mensagens
  - São formadas por 3 partes
    - Header e o Trailer "envelopam" o payload



# NoC

## Network-on-Chip (Redes Intra-Chip)

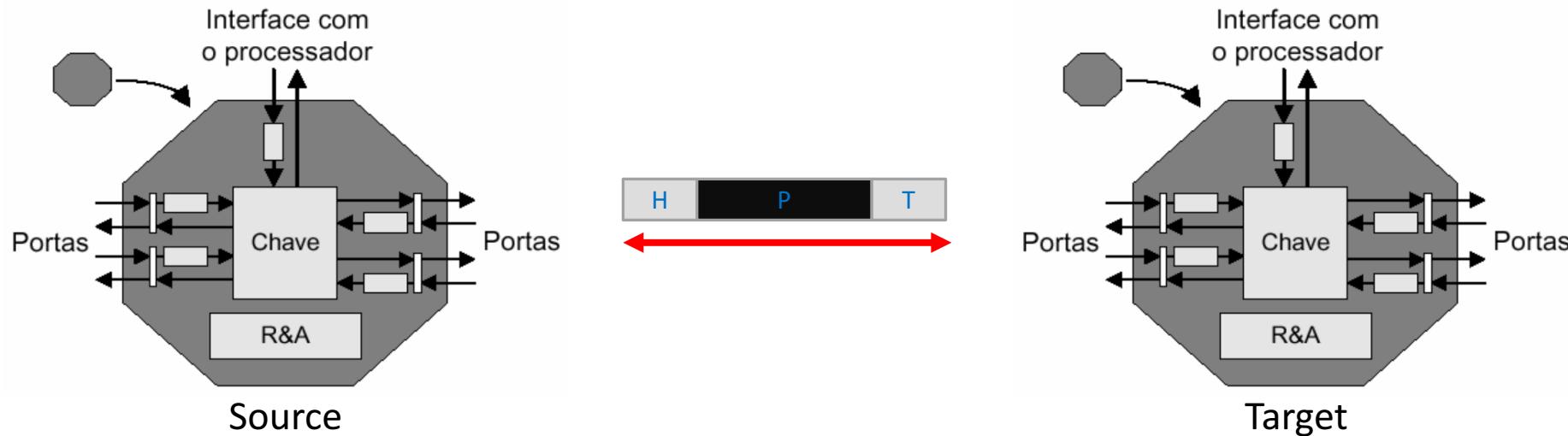
- A comunicação entre um nodo fonte (source) e um destino (target) é feita sob troca de mensagens
  - São formadas por 3 partes
    - O Header carrega informações de roteamento e de controle utilizadas pelos nodos para propagar a mensagem em direção ao target



# NoC

## Network-on-Chip (Redes Intra-Chip)

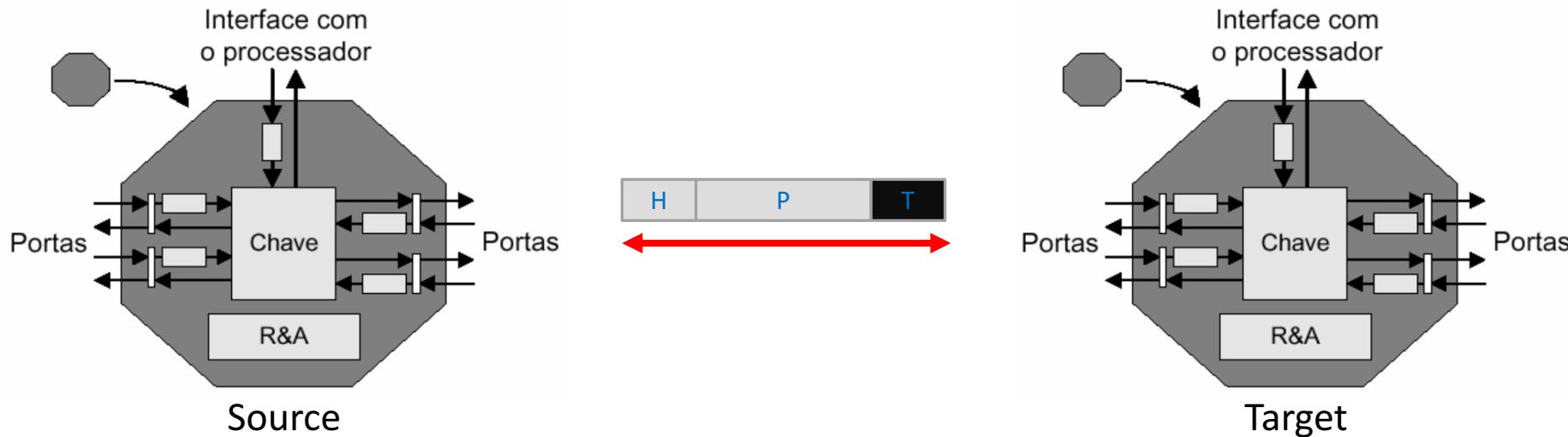
- A comunicação entre um nodo fonte (source) e um destino (target) é feita sob troca de mensagens
  - São formadas por 3 partes
    - O payload carrega a informação que deve ser consumida pelo target



# NoC

## Network-on-Chip (Redes Intra-Chip)

- A comunicação entre um nodo fonte (source) e um destino (target) é feita sob troca de mensagens
  - São formadas por 3 partes
    - O Trailer carrega informações de erros e para sinalização de final da mensagem



MPSoC

# MPSoC

---

Projetos de sistemas modernos apresentam uma tendência de integrar múltiplos núcleos de processamento

Com o aumento da complexidade das aplicações surgiu a necessidade de projetar este tipo de sistema

- MPSoC → Multiprocessor System on Chip

# MPSoC

---

A comunicação nesses sistemas multiprocessador por ser por meio de barramento simples ou NoCs

Em nível de processamento, MPSoC pode ser

- Homogêneos
  - Onde todos os núcleos são idênticos
- Heterogêneos
  - Onde pode, no mesmo sistema, conter núcleos com diferentes características

# MPSoC

---

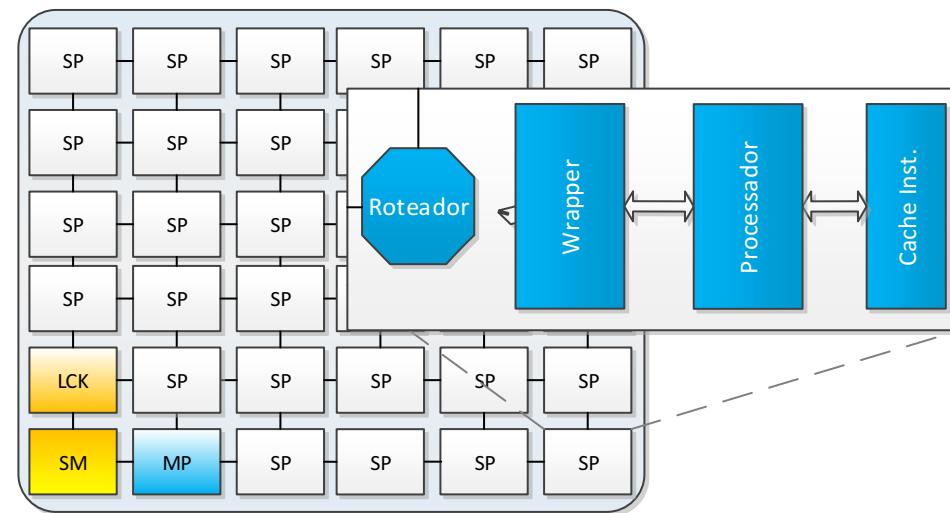
Arquitetura de memória define o modelo de programação adotado pelo MPSoC

- Aplicações baseadas em *threads* são adequadas para arquitetura de memória compartilhada
- Aplicações com a comunicação baseada em troca de mensagens são adequadas para arquitetura de memórias distribuídas

# MPSoC

Arquitetura de memória define o modelo de programação adotado pelo MPSoC

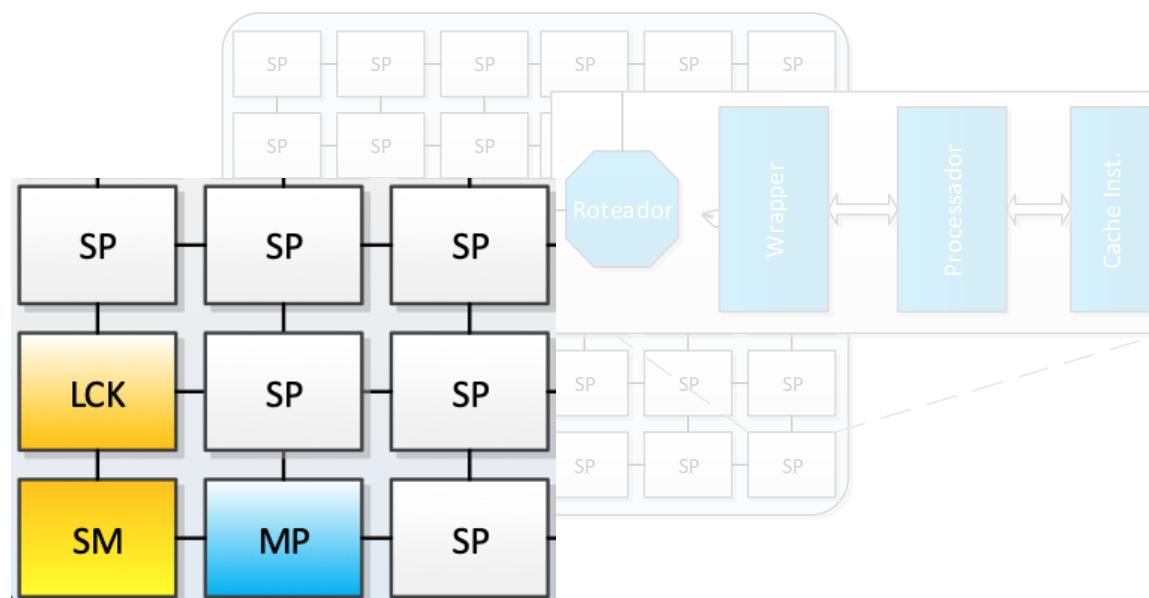
- Em sistemas de memórias compartilhadas todos os elementos de processamento buscam dados num ponto central, a memória do sistema



# MPSoC

Arquitetura de memória define o modelo de programação adotado pelo MPSoC

- Em sistemas de memórias compartilhadas todos os elementos de processamento buscam dados num ponto central, a memória do sistema

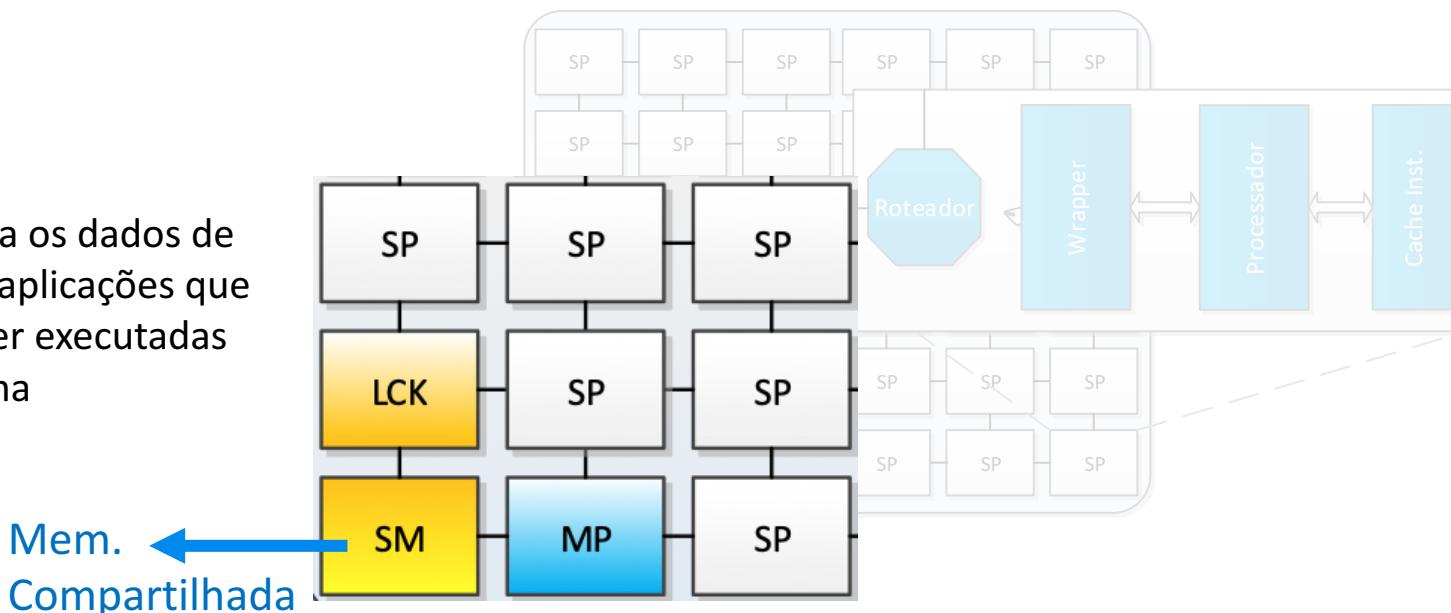


# MPSoC

Arquitetura de memória define o modelo de programação adotado pelo MPSoC

- Em sistemas de memórias compartilhadas todos os elementos de processamento buscam dados num ponto central, a memória do sistema

Armazena os dados de todas as aplicações que devem ser executadas no sistema



# MPSoC

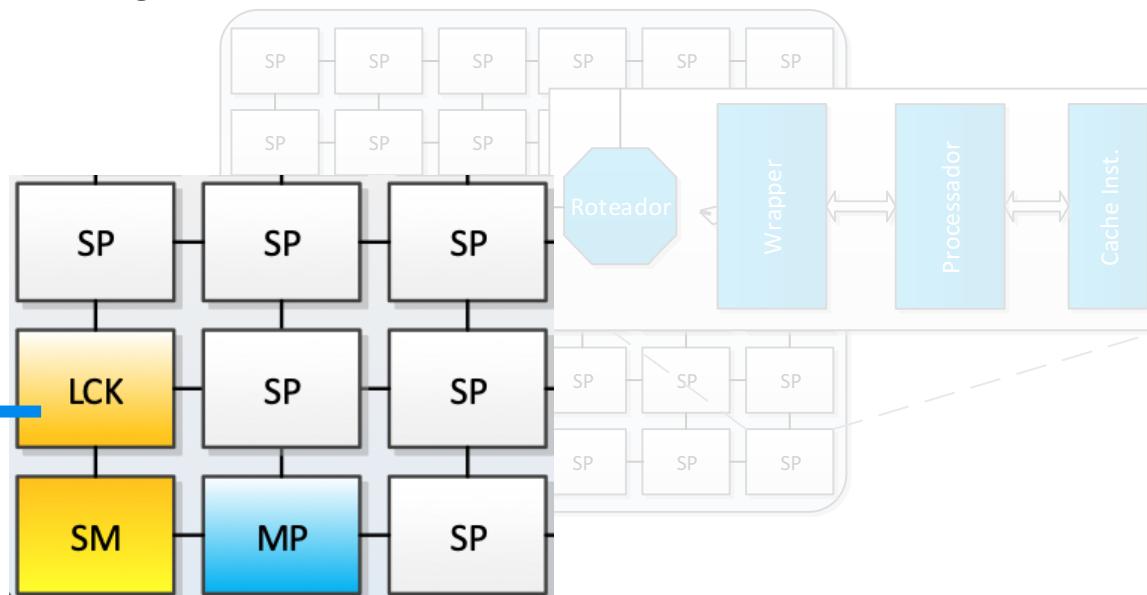
Arquitetura de memória define o modelo de programação adotado pelo MPSoC

- Em sistemas de memórias compartilhadas todos os elementos de processamento buscam dados num ponto central, a memória do sistema

Permite ou bloqueia o acesso de algum processador na memória compartilhada

Locker é o elemento de processamento responsável pela atomicidade dos dados

Locker



# MPSoC

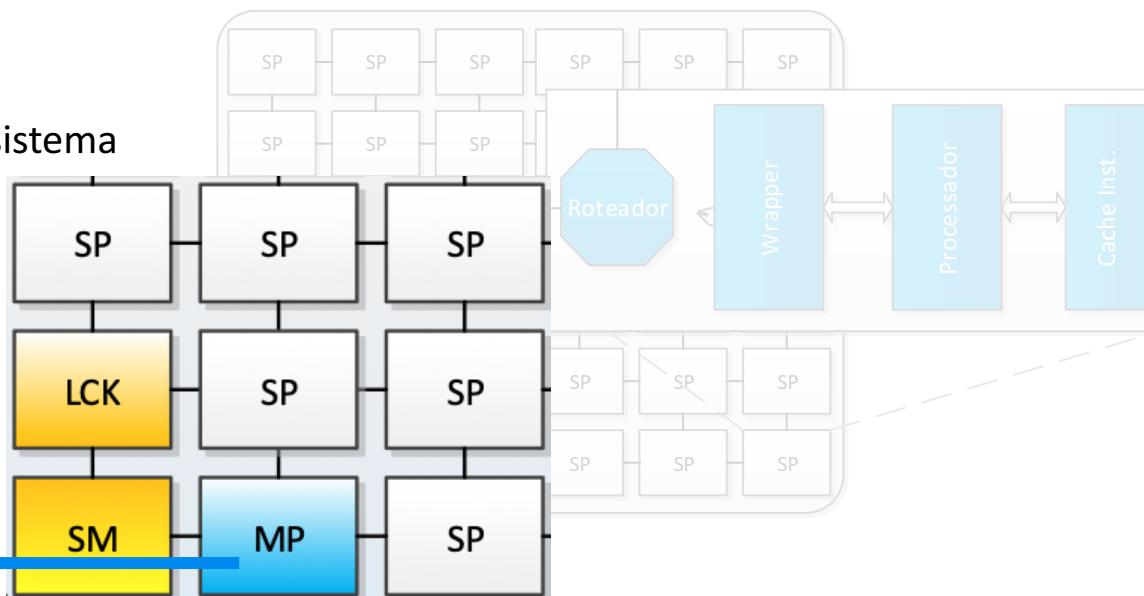
Arquitetura de memória define o modelo de programação adotado pelo MPSoC

- Em sistemas de memórias compartilhadas todos os elementos de processamento buscam dados num ponto central, a memória do sistema

Elemento de processamento  
responsável por gerenciar o sistema

Falaremos mais  
especificamente dele  
na sequência

Manager  
Processor



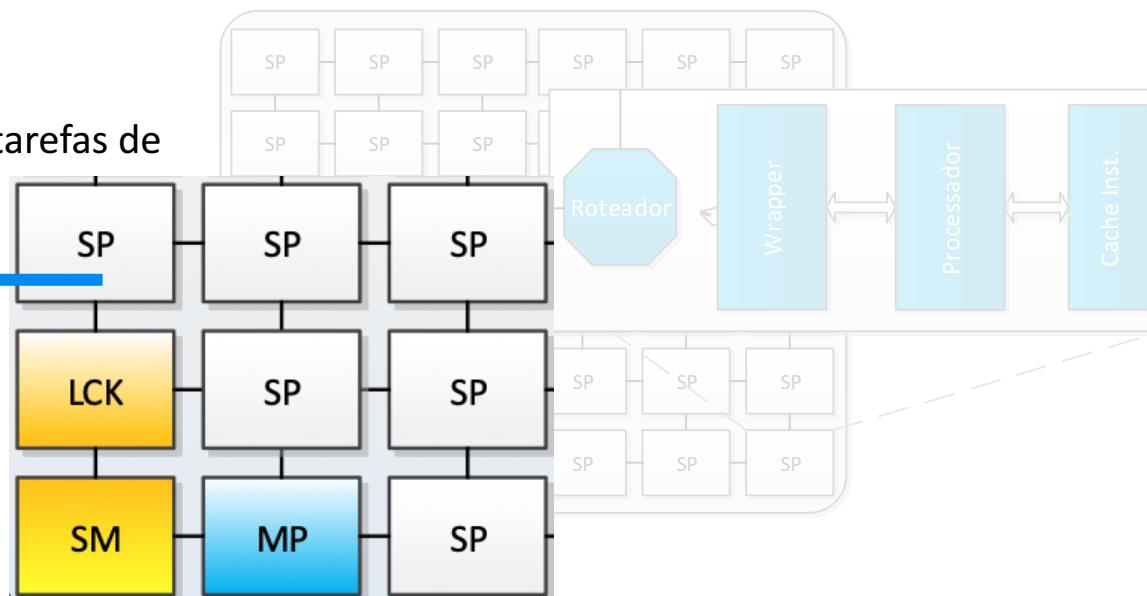
# MPSoC

Arquitetura de memória define o modelo de programação adotado pelo MPSoC

- Em sistemas de memórias compartilhadas todos os elementos de processamento buscam dados num ponto central, a memória do sistema

Elemento de processamento  
responsável por executar as tarefas de  
usuário

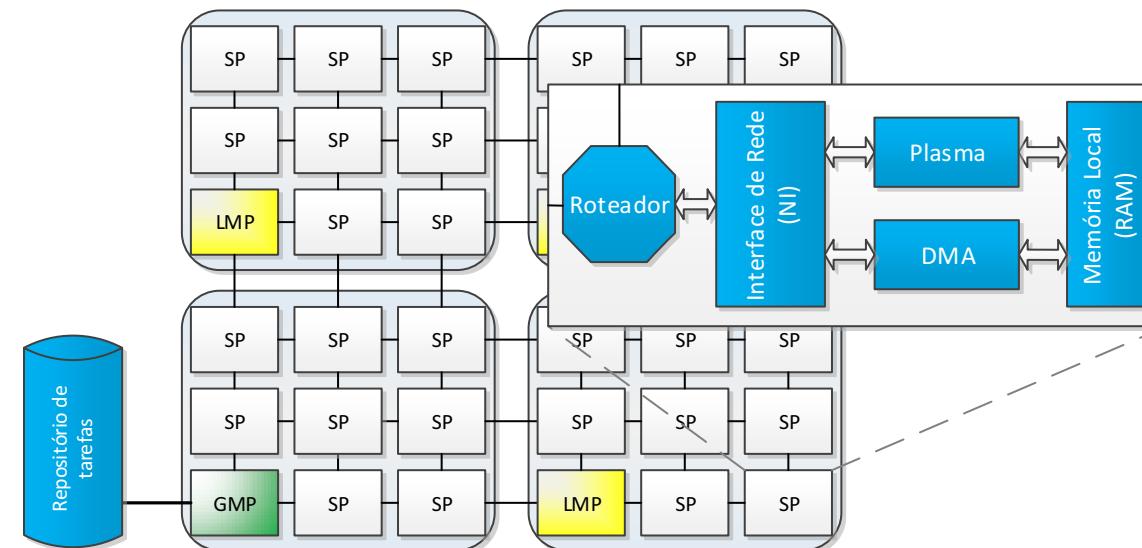
Slave  
Processor



# MPSoC

Arquitetura de memória define o modelo de programação adotado pelo MPSoC

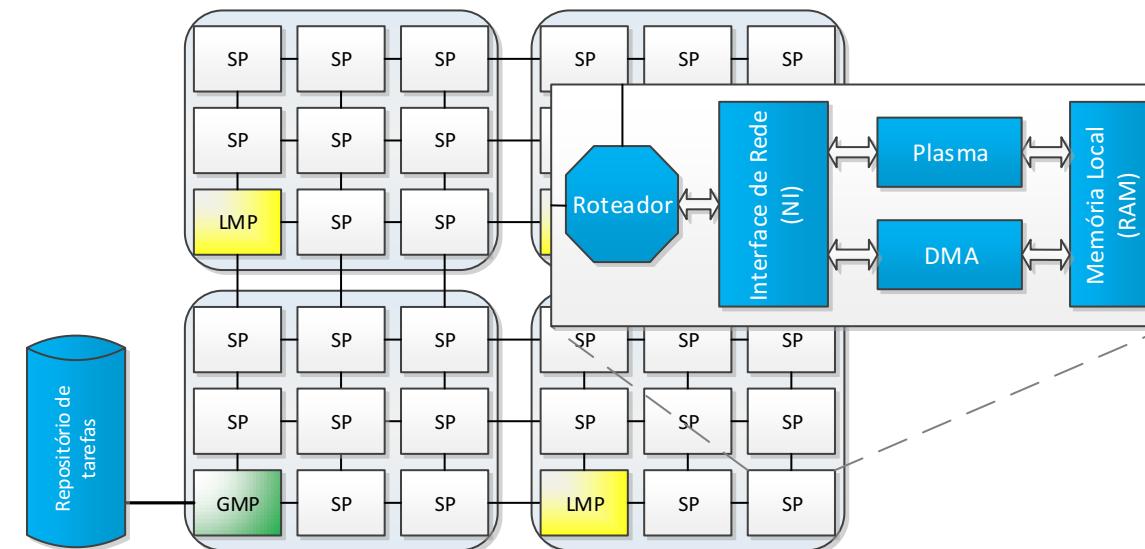
- Em sistemas de memórias distribuída cada elemento de processamento possui uma memória local
- Quando outro elemento precisar do dado de outro, troca-se mensagem (MPI)



# MPSoC

Arquitetura de memória define o modelo de programação adotado pelo MPSoC

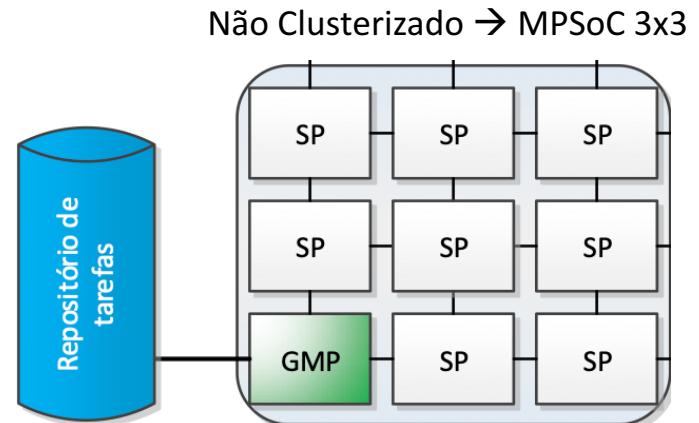
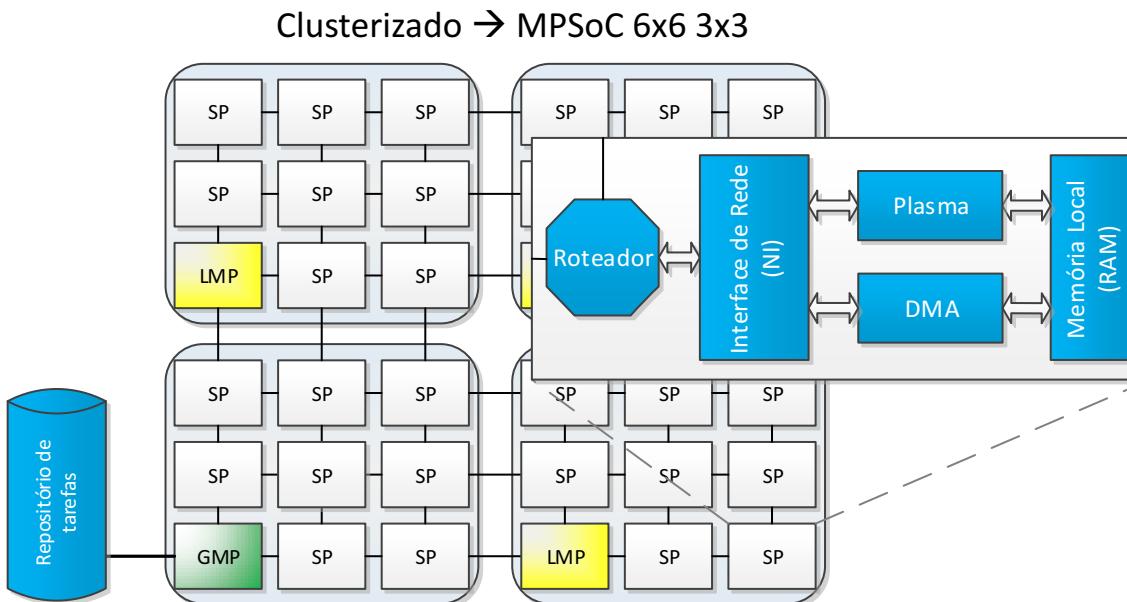
- Este tipo de sistema pode ser “clusterizado”
- Divide o gerenciamento dos processadores em pequenas regiões



# MPSoC

Arquitetura de memória define o modelo de programação adotado pelo MPSoC

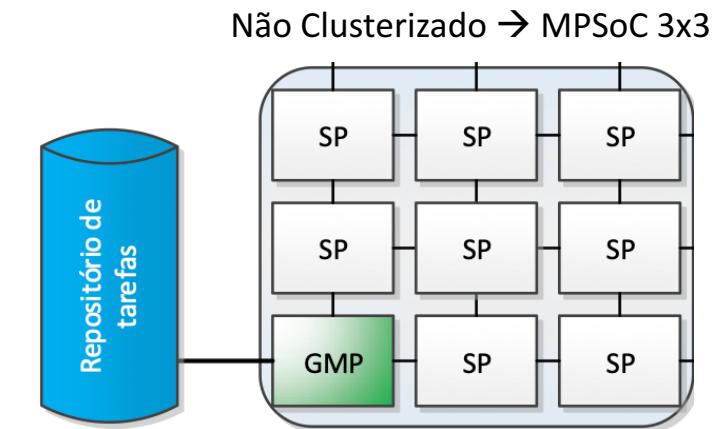
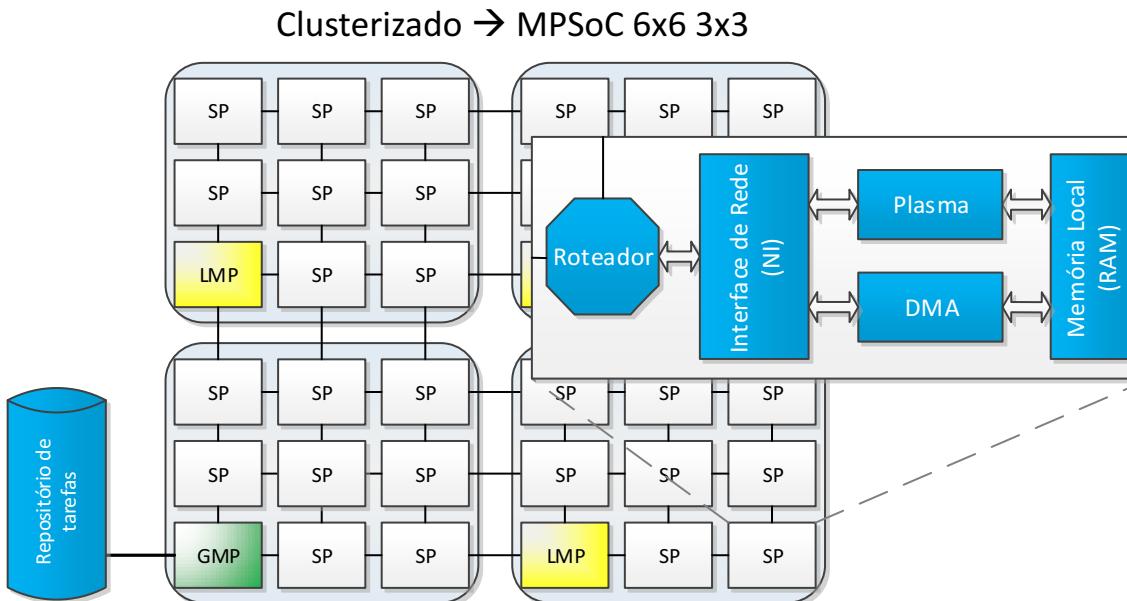
- Este tipo de sistema pode ser “clusterizado”
- Divide o gerenciamento dos processadores em pequenas regiões



# MPSoC

Arquitetura de memória define o modelo de programação adotado pelo MPSoC

- MPSoC tem o tamanho X e Y
- O cluster tem tamanho X e Y (devendo seu tamanho ser encaixado no tamanho do MPSoC)



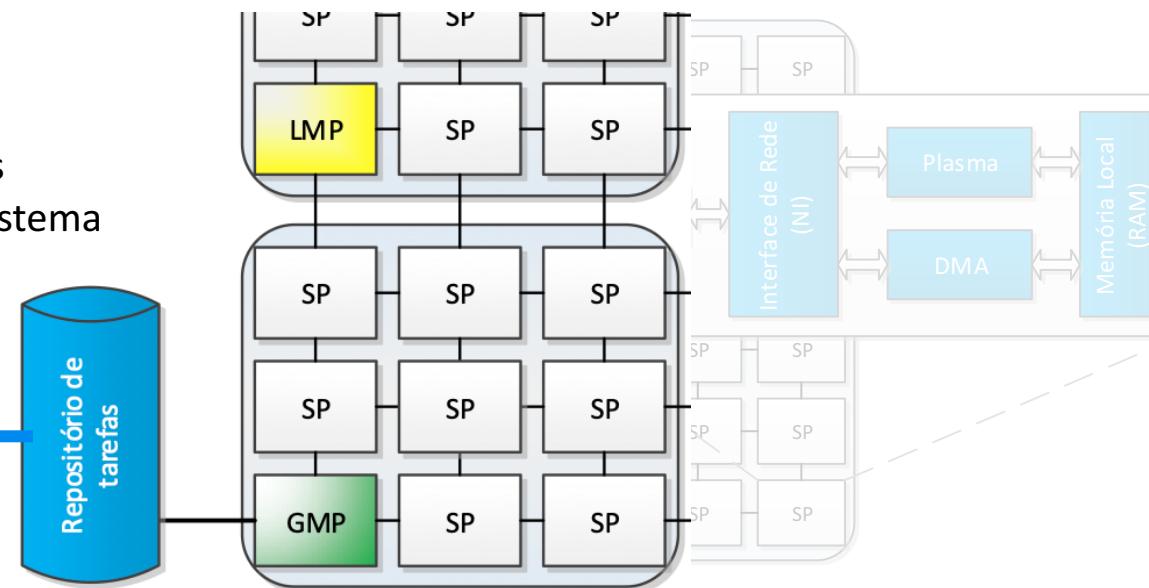
# MPSoC

Arquitetura de memória define o modelo de programação adotado pelo MPSoC

- Em sistemas de memórias distribuída cada elemento de processamento possui uma memória local
- Quando outro elemento precisar do dado de outro, troca-se mensagem (MPI)

Repositório que contém todas as tarefas a serem executadas no sistema

Repositório de tarefas

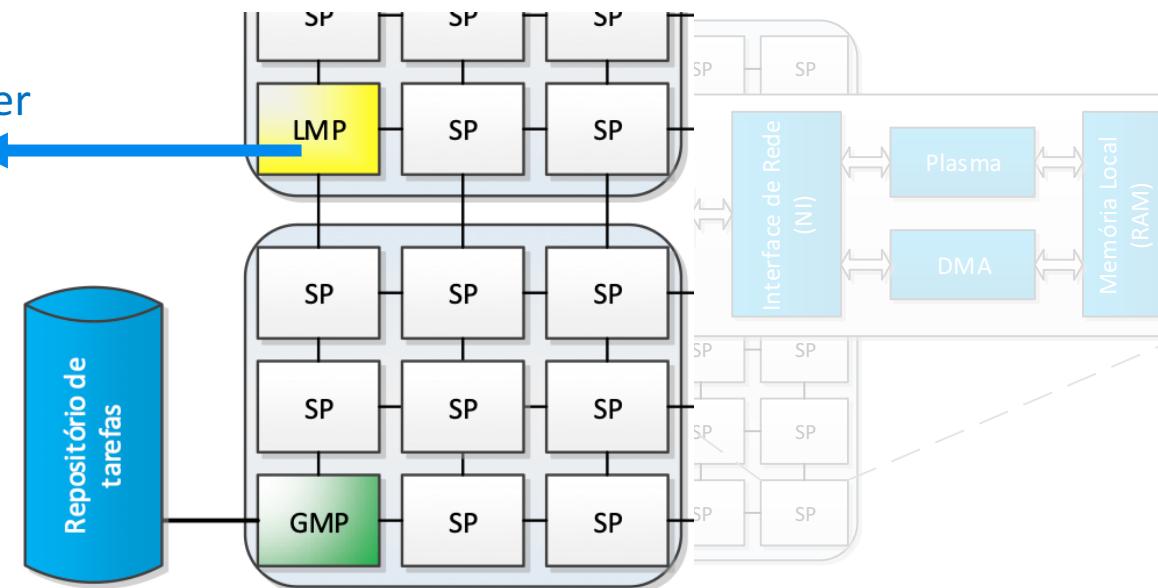


# MPSoC

Arquitetura de memória define o modelo de programação adotado pelo MPSoC

- Em sistemas de memórias distribuída cada elemento de processamento possui uma memória local
- Quando outro elemento precisar do dado de outro, troca-se mensagem (MPI)

Local Manager Processos  
Elemento de processamento responsável por distribuir as tarefas de uma aplicação entre seus SPs



# MPSoC

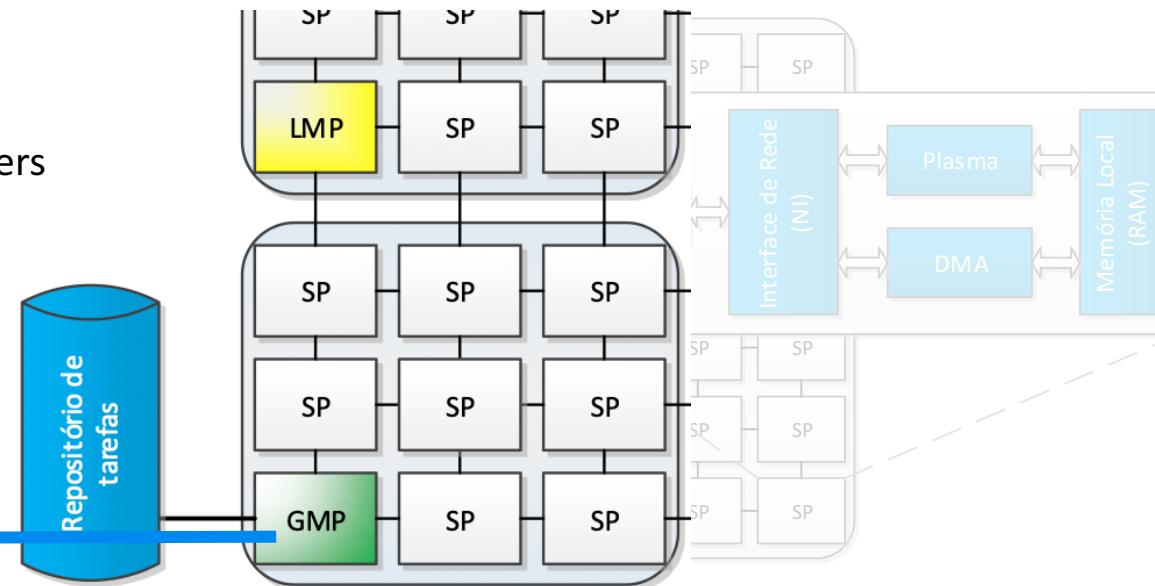
Arquitetura de memória define o modelo de programação adotado pelo MPSoC

- Em sistemas de memórias distribuída cada elemento de processamento possui uma memória local
- Quando outro elemento precisar do dado de outro, troca-se mensagem (MPI)

Elemento de processamento que distribui as aplicações entre os clusters

Possui todas as funcionalidades dos LMPs

Global Manager Processor



# MPSoC

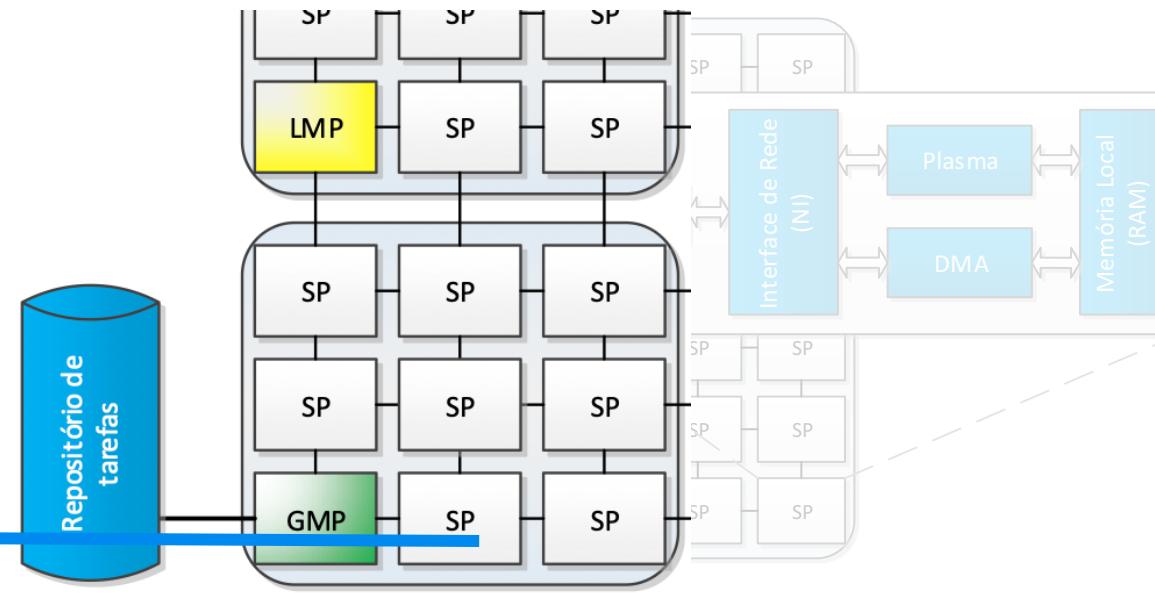
Arquitetura de memória define o modelo de programação adotado pelo MPSoC

- Em sistemas de memórias distribuída cada elemento de processamento possui uma memória local
- Quando outro elemento precisar do dado de outro, troca-se mensagem (MPI)

Elemento de processamento  
responsável pela execução das tarefas  
de usuários

Pode executar multiplas tarefas  
Possui um SO que gerencia a execução

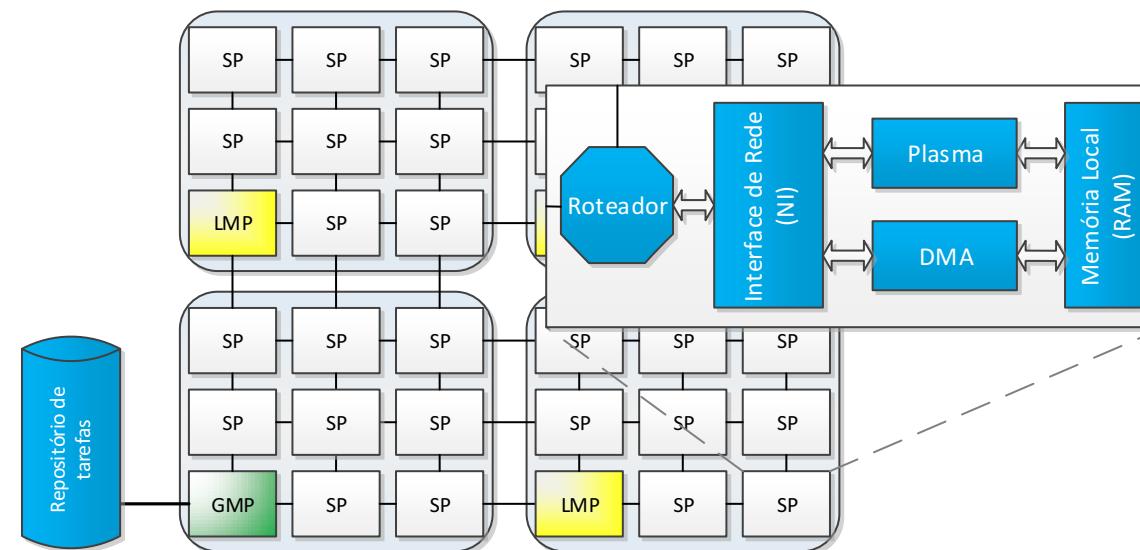
Slave  
Processor



# MPSoC

Arquitetura de memória define o modelo de programação adotado pelo MPSoC

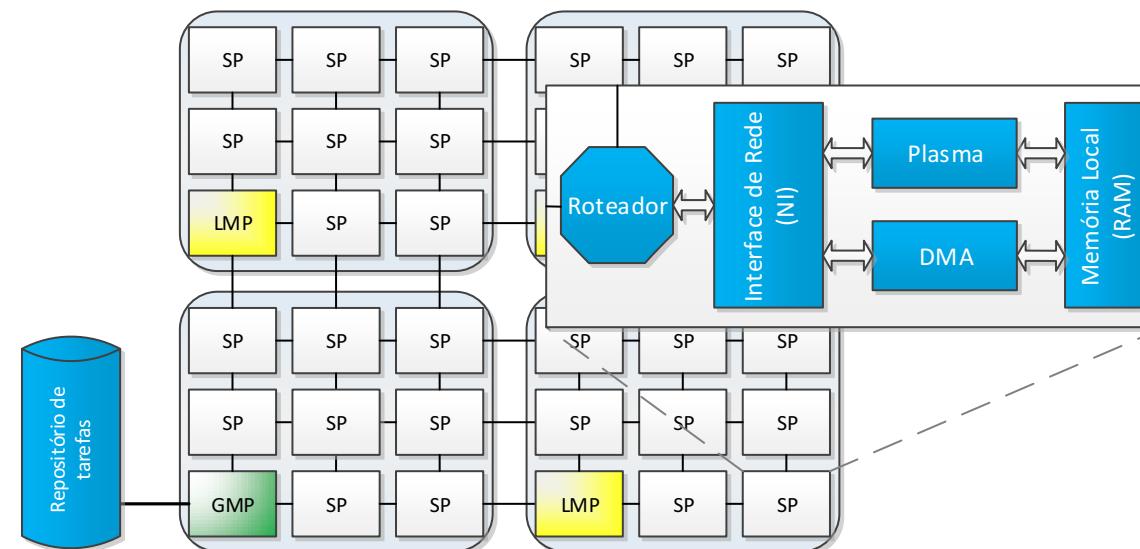
- Quando o GMP busca uma aplicação no repositório de tarefas, ele verifica quais clusters tem espaço suficiente para executar esta aplicação



# MPSoC

Arquitetura de memória define o modelo de programação adotado pelo MPSoC

- Quando o LMP recebe uma aplicação, ele executa um algoritmo de mapeamento para decidir em quais processadores cada tarefa vai executar

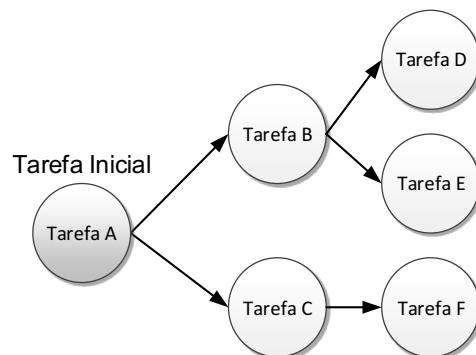


Comparativo

# MPSoC

Arquitetura de memória define o modelo de programação adotado pelo MPSoC

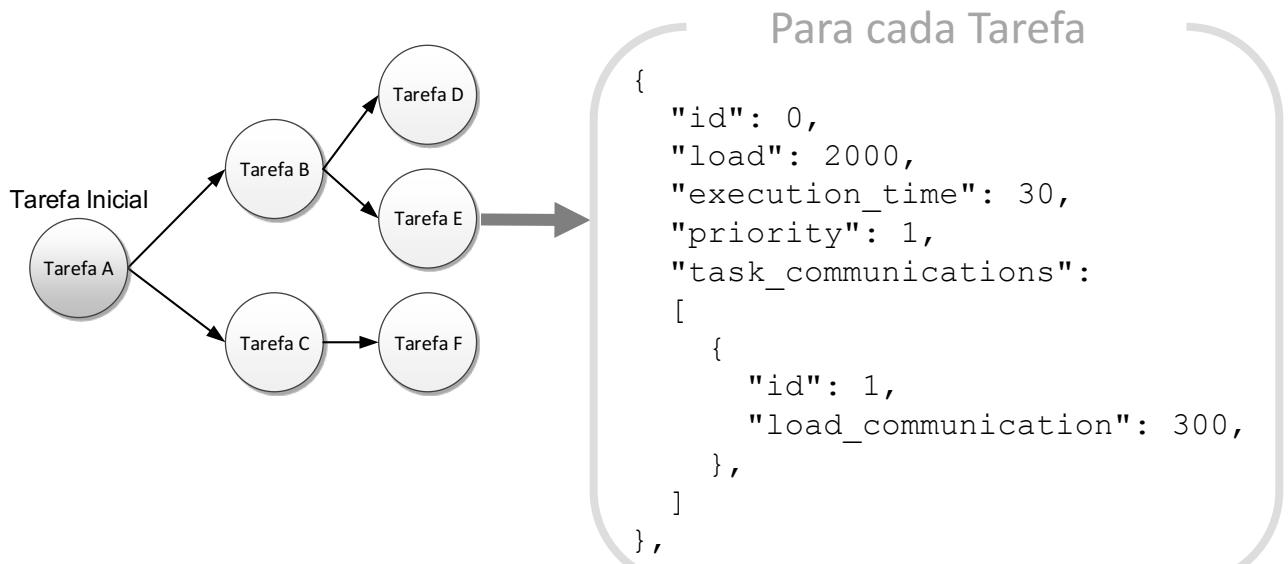
- As aplicações que executam neste tipo de sistema são modeladas como grafo de tarefas
  - Os vértices representam as tarefas
  - As arestas representam a comunicação entre as tarefas



# MPSoC

Arquitetura de memória define o modelo de programação adotado pelo MPSoC

- As aplicações que executam neste tipo de sistema são modeladas como grafo de tarefas
  - Os vértices representam as tarefas
  - As arestas representam a comunicação entre as tarefas



# Comparativo

---

Dependendo dos requisitos não funcionais que o projeto apresenta, temos que ter ciência das limitações de cada arquitetura

Para isso, vamos analisar um comparativo entre as duas arquiteturas de memória para projetos embarcados

# Comparativo

---

Executando as mesmas aplicações vamos analisar duas avaliações

1. Volume de comunicação transferido pela NoC → representa a distribuição de tráfego com balanceamento da carga de comunicação
2. Número total de instruções executada pelos processadores

# Comparativo

Executando as mesmas aplicações vamos analisar duas avaliações

1. Volume de comunicação transferido pela NoC → representa a distribuição de tráfego com balanceamento da carga de comunicação

Mem. Distribuída					
0.19	0.27	0.19	0.19	0.15	0.10
0.29	0.95	0.20	0.33	0.97	0.15
0.39	0.32	0.69	0.75	0.31	0.21
0.57	0.51	0.41	0.53	0.26	0.27
0.62	1.00	0.84	0.80	0.86	0.39
0.43	0.58	0.64	0.20	0.37	0.27

(a) Dijkstra application

Mem. Compartilhada					
0.0428	0.0268	0.0147	0.0044	0.0010	0.0003
0.0856	0.0361	0.0208	0.0064	0.0014	0.0005
0.1384	0.0491	0.0278	0.0080	0.0020	0.0008
0.2298	0.0781	0.0418	0.0121	0.0026	0.0012
0.3662	0.1181	0.0549	0.0197	0.0043	0.0018
(L) 0.6551	0.1934	0.0823	0.0216	0.0070	0.0031
(SM) 1.0000	0.3488	0.2251	0.0641	0.0179	0.0052

(a) Dijkstra application

# Comparativo

Executando as mesmas aplicações vamos analisar duas avaliações

1. Volume de comunicação transferido pela NoC → representa a distribuição de tráfego com balanceamento da carga de comunicação

Mem. Distribuída					
0.22	0.34	0.34	0.33	0.33	0.22
0.33	0.45	0.45	0.33	0.33	0.22
0.45	0.56	0.89	0.67	0.67	0.56
0.55	0.67	1.00	0.67	0.55	0.67
0.66	0.45	0.34	0.34	0.34	0.45
0.44	0.44	0.33	0.33	0.33	0.22

(b) FFT application

Mem. Compartilhada					
0.0501	0.0266	0.0195	0.0104	0.0033	0.0017
0.0893	0.0358	0.0249	0.0155	0.0047	0.0020
0.1124	0.0565	0.0324	0.0175	0.0064	0.0026
0.1991	0.0982	0.0542	0.0284	0.0082	0.0031
0.3265	0.1251	0.0811	0.0409	0.0129	0.0039
(L) 0.7112	0.2433	0.1384	0.0492	0.0162	0.0047
(SM) 1.0000	0.3771	0.2368	0.0544	0.0198	0.0061

(b) FFT application

# Comparativo

Executando as mesmas aplicações vamos analisar duas avaliações

## 2. Número total de instruções executada pelos processadores

Mem. Distribuída

<b>Número de PEs</b>	<b>Número total de instruções executadas (<math>10^6</math> instruções)</b>	
	<b>Dijkstra</b>	<b>FFT</b>
<b>4</b>	1,7	5,2
<b>16</b>	4,5	12,4
<b>36</b>	13,8	22,3
<b>64</b>	49,3	40,1
<b>100</b>	117,1	62,9
<b>144</b>	241,0	91,1
<b>225</b>	588,1	143,0

Mem. Compartilhada

<b>Número de PEs</b>	<b>Número total de instruções executadas (<math>10^6</math> instruções)</b>	
	<b>Dijkstra</b>	<b>FFT</b>
<b>4</b>	5,18	10,3
<b>8</b>	6,06	13,4
<b>10</b>	8,68	17,9
<b>12</b>	13,21	32,6
<b>18</b>	18,13	38,4
<b>42</b>	339,99	706,9

# MPSoC

---

## Entendendo o trabalho

- Ter a possibilidade de, no início do sistema, criar uma aplicação
- O sistema deve pedir:
  - Nome da aplicação → ex: whatsapp
  - Tarefas
    - A
    - B
    - C
    - D
- O sistema deverá, com base em algum algoritmo, mapear as tarefas ao longo dos nodos
- Solicitar ao usuário qual tarefa se comunica com qual