



STR

- Introdução
- Processador
- Memória
- Periféricos
- Considerações gerais
- Referências

Capítulo 2

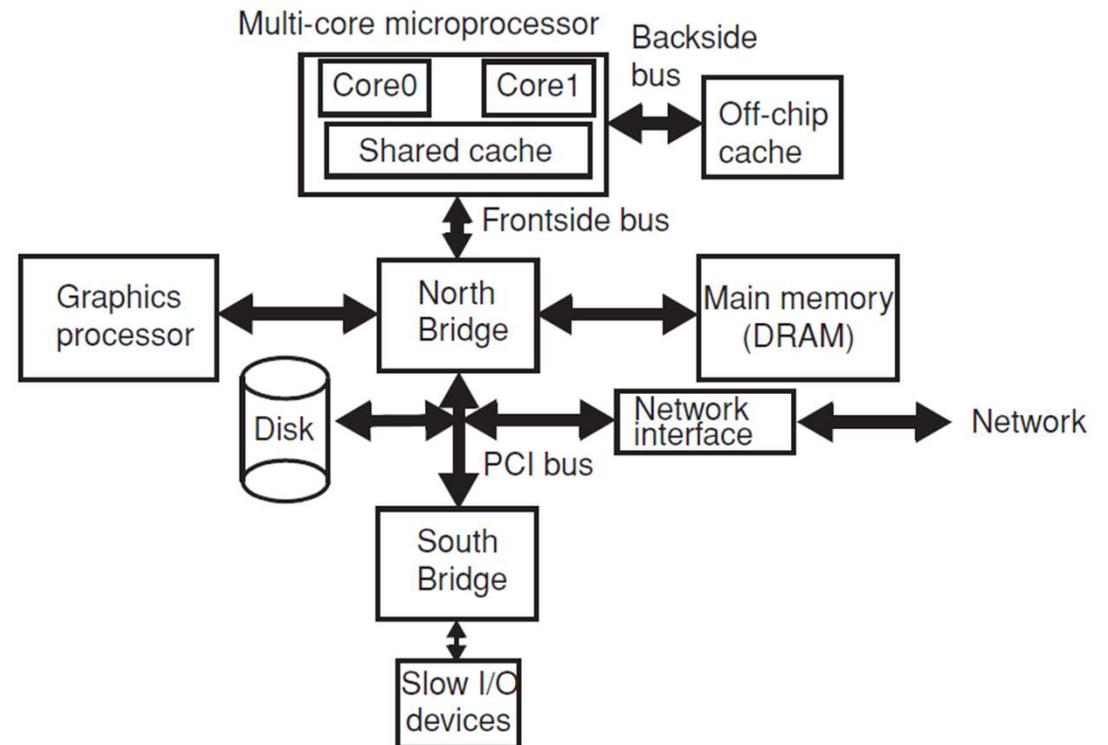
Hardware para STRs

STR

- Hardware para STR:

- Latência hardware (entrada a saída) = entendida/ gerenciável
- *“an understanding of advanced computer architectures is essential to the real-time systems engineer.”* pág 44 de [1]

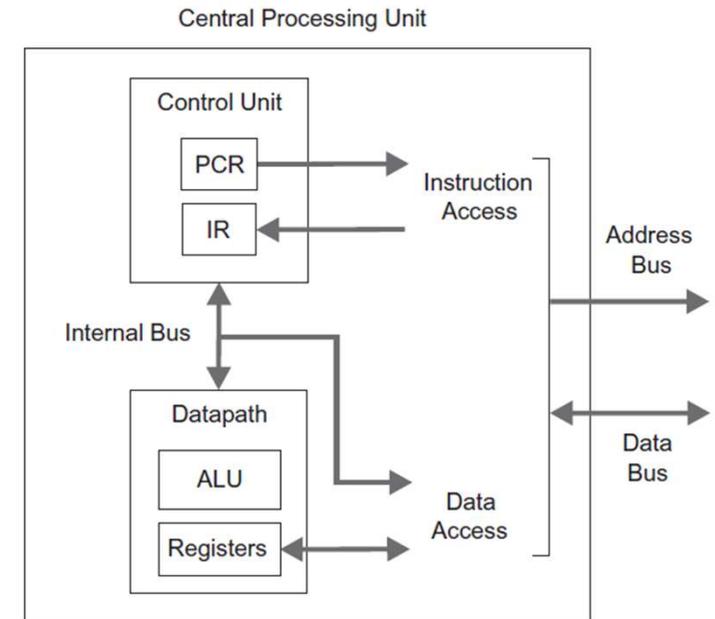
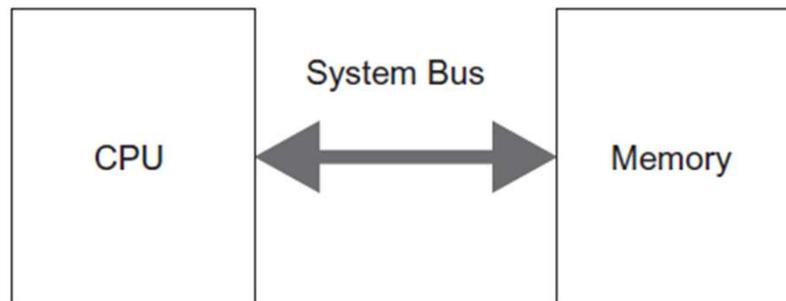
- Introdução
- Processador
- Memória
- Periféricos
- Considerações gerais
- Referências



STR

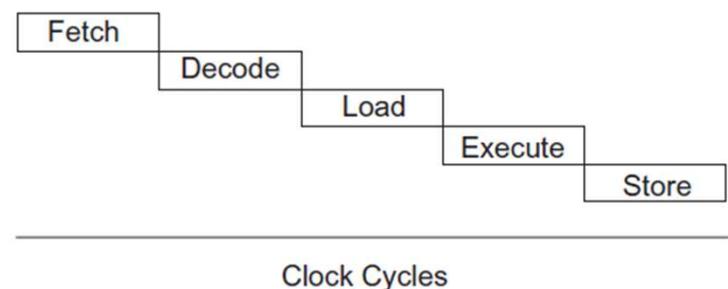
- Introdução
- Processador
- Memória
- Periféricos
- Considerações gerais
- Referências

- Arquitetura Von Neumann
 - Memória: dados e programas
 - Registradores de I/O são mapeados em memória



– Ciclo processamento

- Ciclo instrução com latências adicionais (operandos OP-CODE)

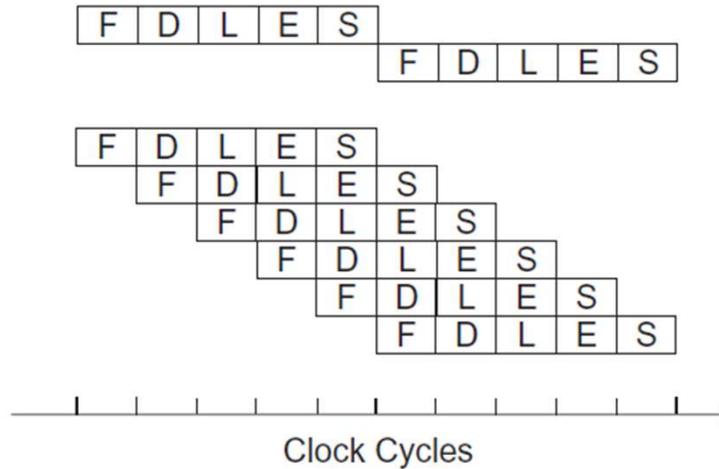




STR

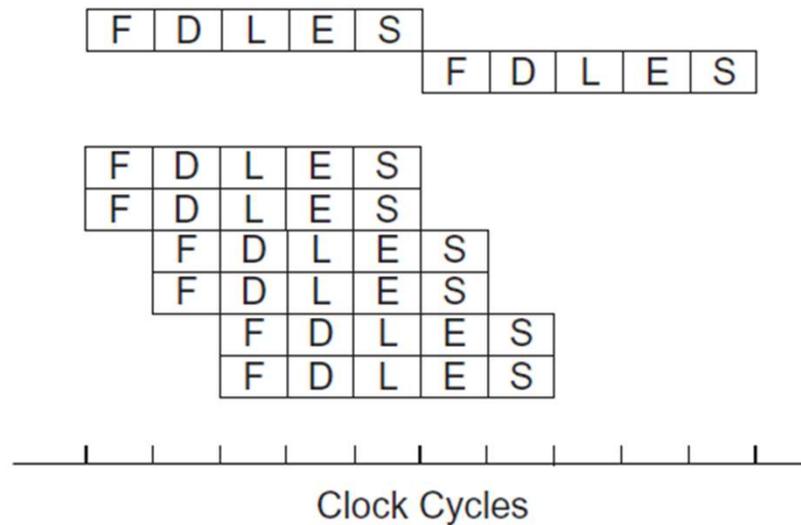
- Introdução
- Processador
- Memória
- Periféricos
- Considerações gerais
- Referências

• Pipeline de instrução



• Arquitetura superscalar

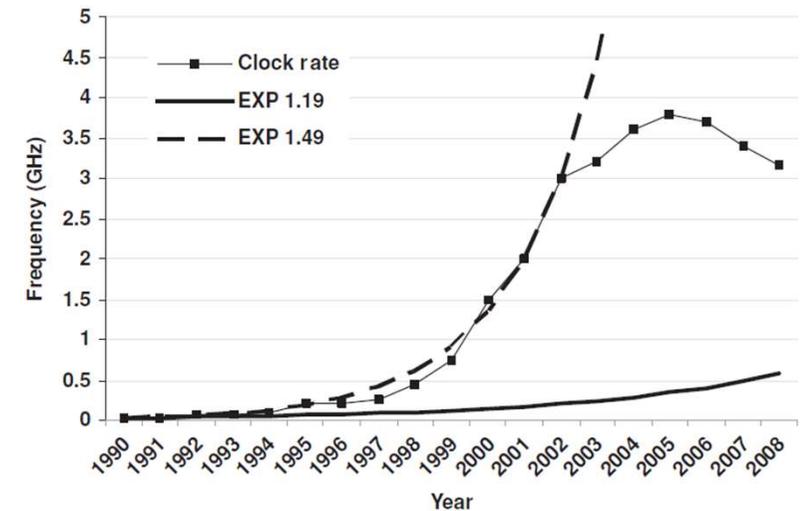
- 2 threads processador



STR

- Introdução
- Processador
- Memória
- Periféricos
- Considerações gerais
- Referências

- Tecnologia multi-core
 - Clocks >5GHz ‘inviáveis’
 - Alto aquecimento
 - Desempenho por “concorrência” dos processos



consumption P_{dyn} (measured in Watt, abbreviated as W) of a processor by

$$P_{dyn}(f) = \alpha \cdot C_L \cdot V^2 \cdot f$$

α is a switching probability,

C_L is the load capacitance,

V is the supply voltage (measured in Volt, abbreviated as V)

f is the clock frequency (measured in Hertz, abbreviated as Hz)

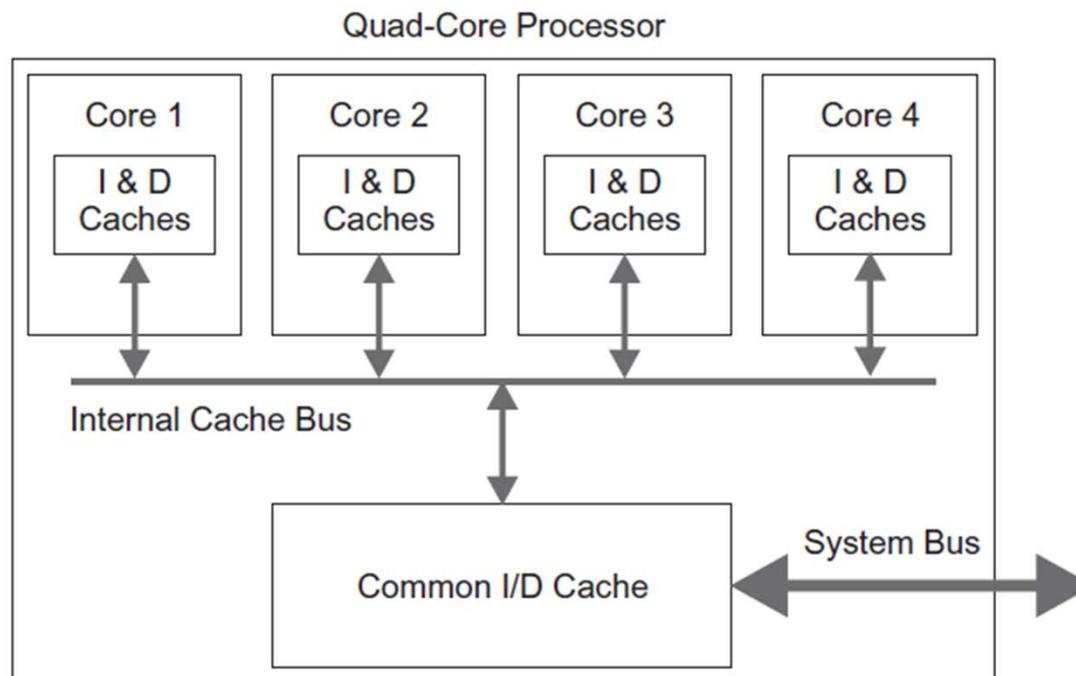


Figure 2.12. Quad-core processor architecture with individual on-chip caches and a common on-chip cache (“I” = instruction and “D” = data).

STR

- Introdução
- Processador
- Memória
- Periféricos
- Considerações gerais
- Referências

• Computador com múltiplos processadores

- P = processador;
- C = cache hierárquica;
- M = memória;
- NI = interface rede

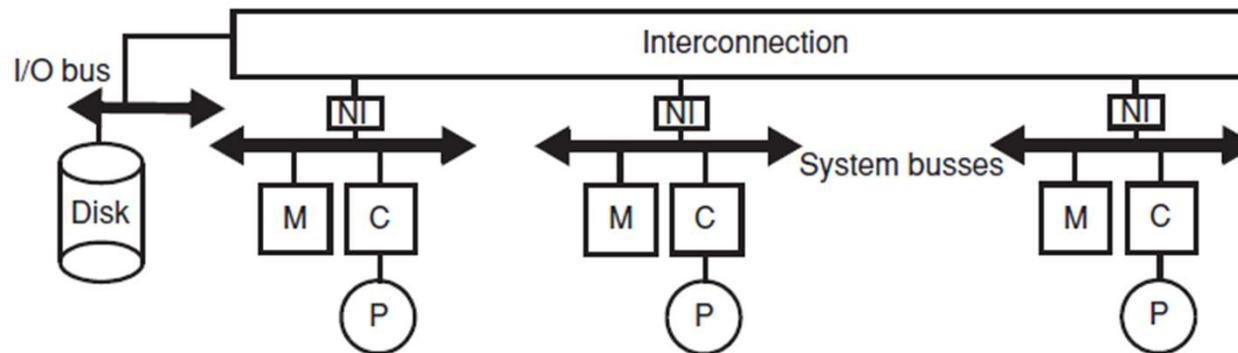
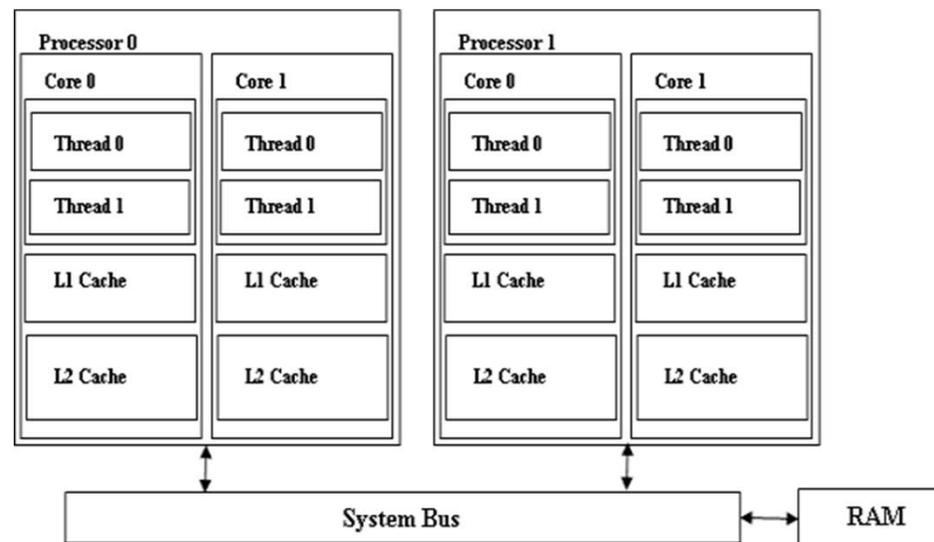


Figure 1.4. Generic multiprocessor system with distributed memory.



STR

- Introdução
- Processador
- Memória
- Periféricos
- Considerações gerais
- Referências

• Instruction-level parallelism (ILP)

- Uso de ILP é feito automaticamente pelo compilador e hardware

Alto ILP

```
for (i=0,i++,1023)
  A[i]:=B[i]+C[i];
```

(a)

```
for (i=0,i++,255)
  A[i]=B[i]+C[i];
```

(b)

Baixo ILP

```
for (i=0,i++,1023)
  S:=S+A[i];
```

```
for (i=0,i++,255)
  my_S=my_S+A[i];
  S=my_S+S;
```

Figure 1.11. ILP vs. TLP. (a) Single-threaded programs. (b) Multi-threaded programs (thread_0 of four threads).

4 threads

my_S = privada

S = compartilhada

• Thread-level parallelism (TLP)

- Dividir o Código em threads paralelas independentes (núcleos separados)
- Uso de TLP é uma tarefa de programadores (cuidado: '*parallelizing compilers*')

STR

- Introdução
- Processador
- Memória
- Periféricos
- Considerações gerais
- Referências

• Acesso a periféricos como memória

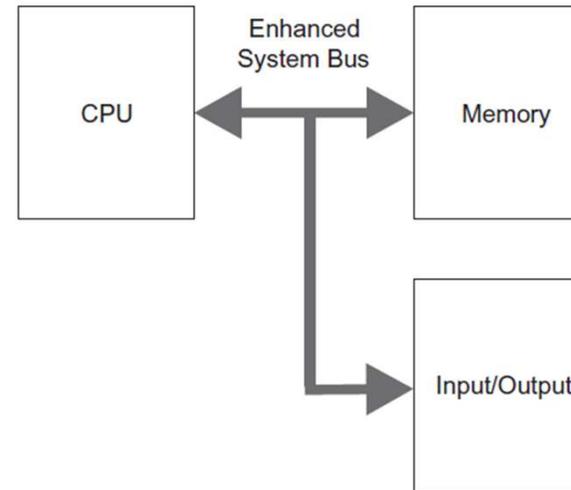


Figure 2.4. Von Neumann architecture with slightly enhanced system bus for programmed input/output.

• Interrupções de hardware:

– Algumas etapas tem tempo imprevisto

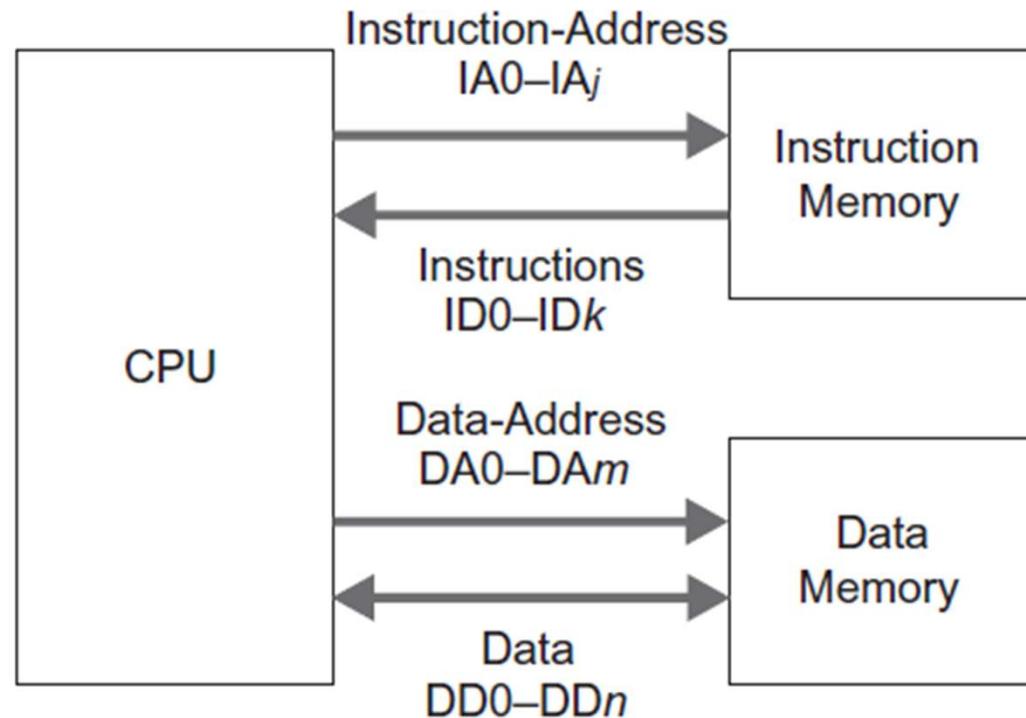
- The interrupt-request line is activated.
- The interrupt request is latched by the CPU hardware (~).
- The processing of the ongoing instruction is completed (~).
- The content of program counter register (PCR) is pushed to stack.
- The content of status register (SR) is pushed to stack.
- The PCR is loaded with the interrupt handler's address.
- The interrupt handler is executed (~).
- The original content of SR is popped from stack.
- The original content of PCR is popped from stack.



STR

- Introdução
- Processador
- Memória
- Periféricos
- Considerações gerais
- Referências

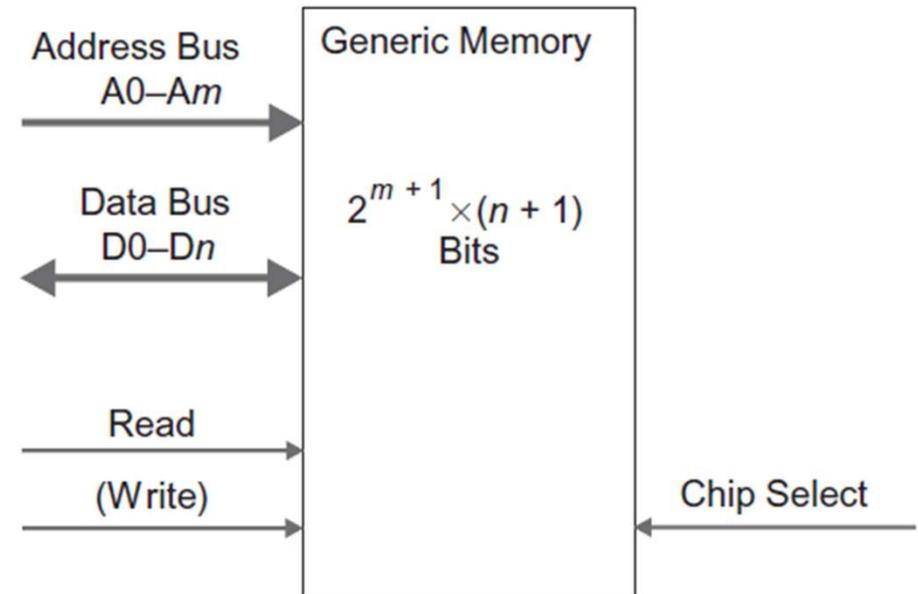
- Arquitetura Harvard
 - Pipeline para dados e instruções
 - Ciclo instrução bem conhecido (sem latências adicionais)



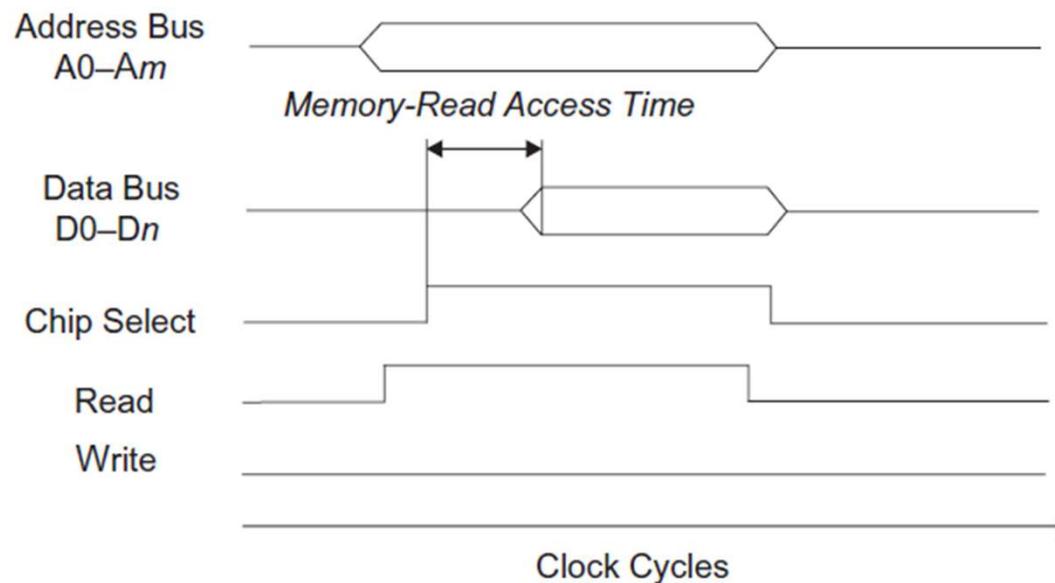
STR

- Introdução
- Processador
- Memória
- Periféricos
- Considerações gerais
- Referências

• Modelo genérico:



– Diagrama de tempo:



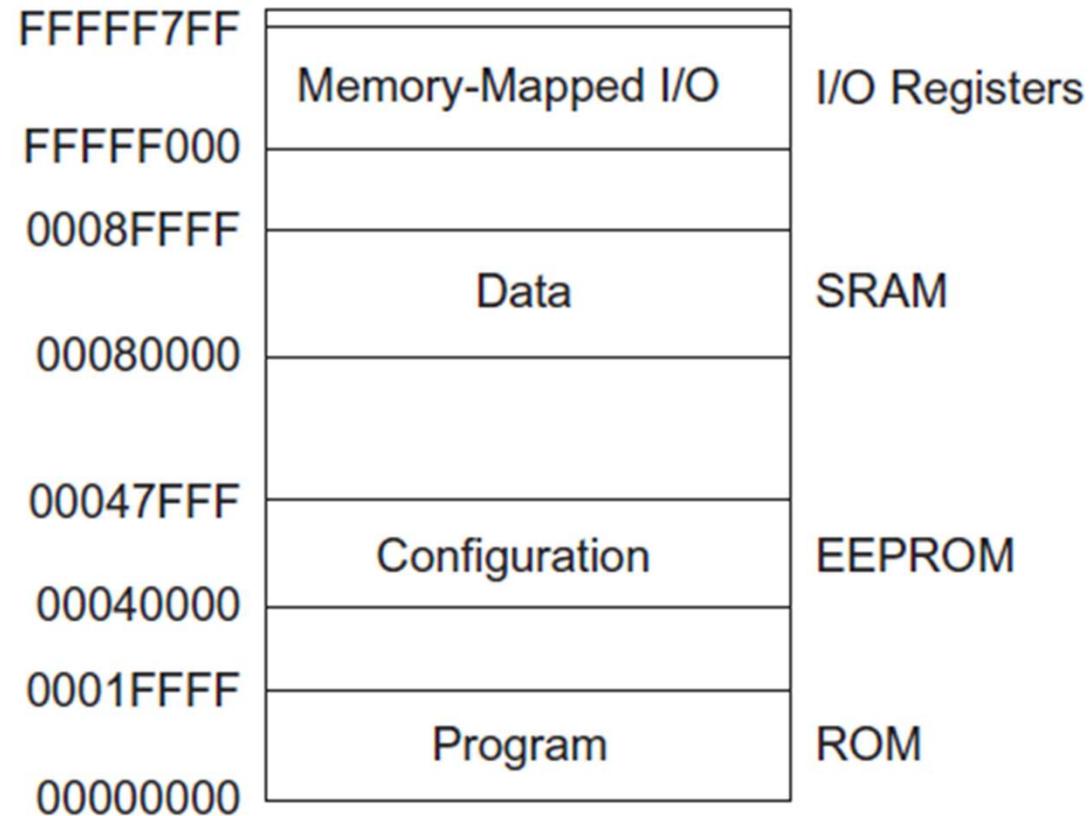


Memória

STR

- Introdução
- Processador
- Memória
- Periféricos
- Considerações gerais
- Referências

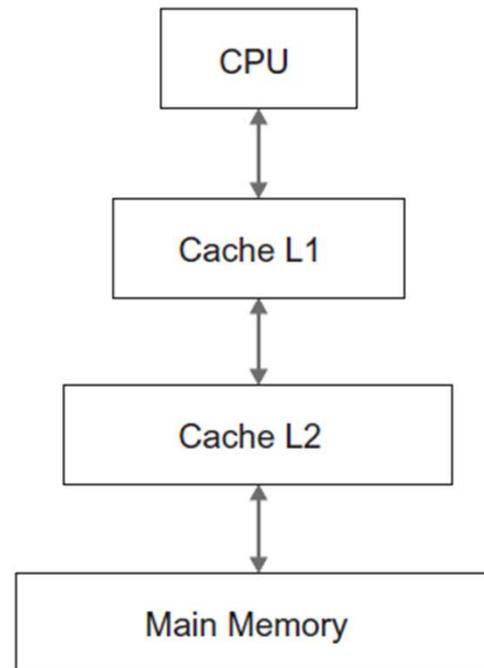
- RAM (SRAM e DRAM) x ROM
- Mapeamento



STR

- Introdução
- Processador
- Memória
- Periféricos
- Considerações gerais
- Referências

- Cache:
 - Princípio da “locality of reference”

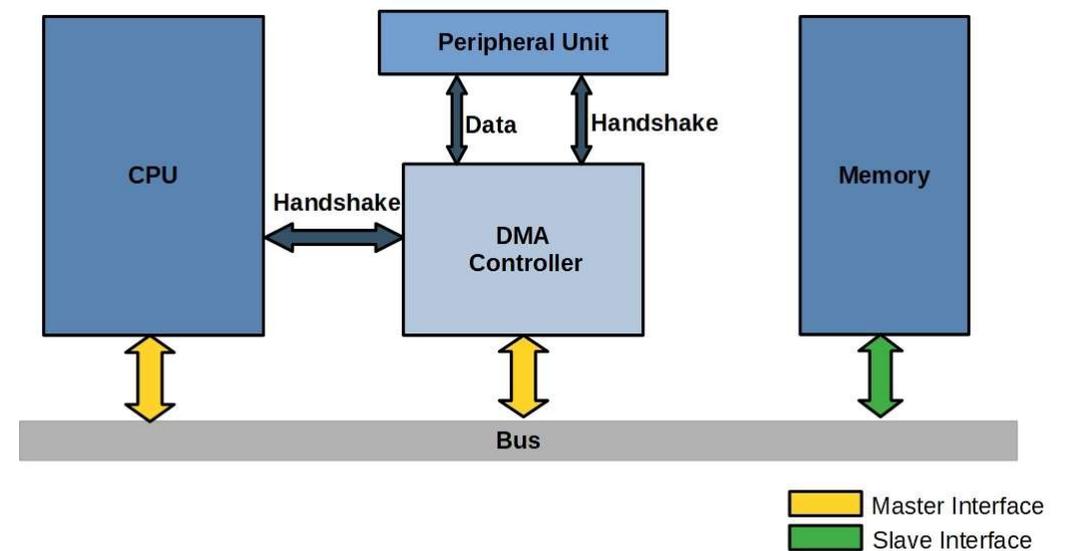
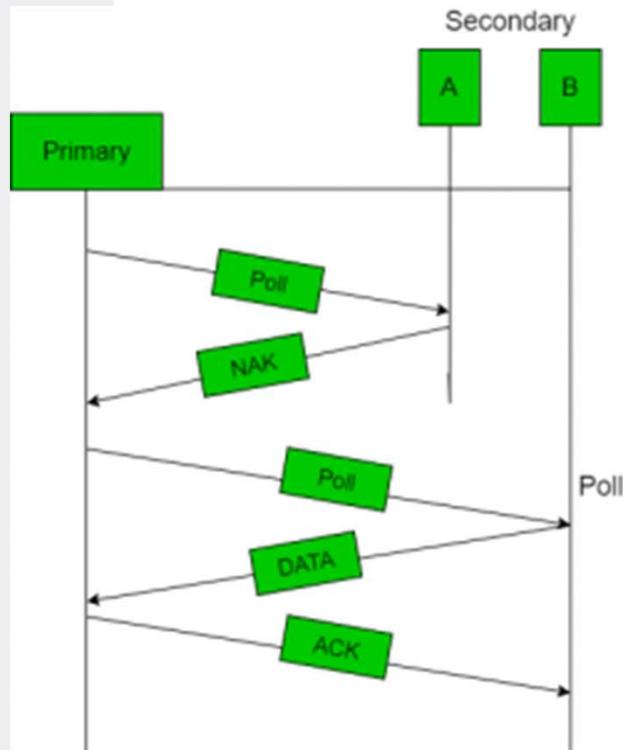
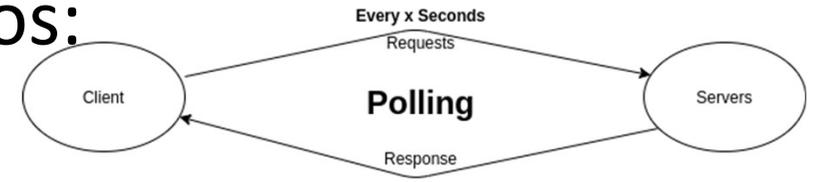


STR

- Introdução
- Processador
- Memória
- Periféricos
- Considerações gerais
- Referências

• Modos de acesso clássicos:

- Polled I/O
- Interrupt-driven I/O
- Direct memory access
- streaming



STR

- Introdução
- Processador
- Memória
- Periféricos
- Considerações gerais
- Referências

- Interface elétrica periféricos (por interrupção)
 - PIU = *peripheral interface unit*

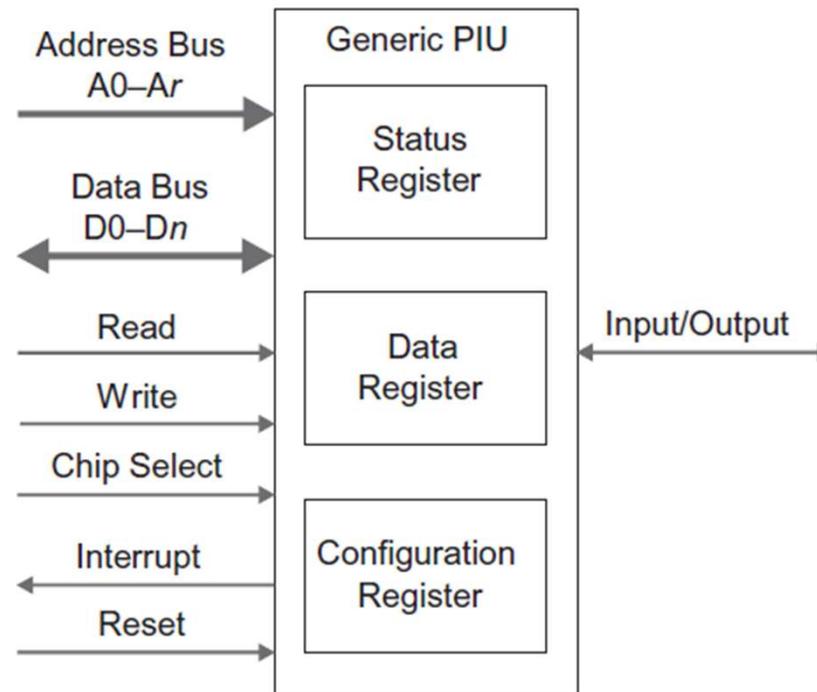
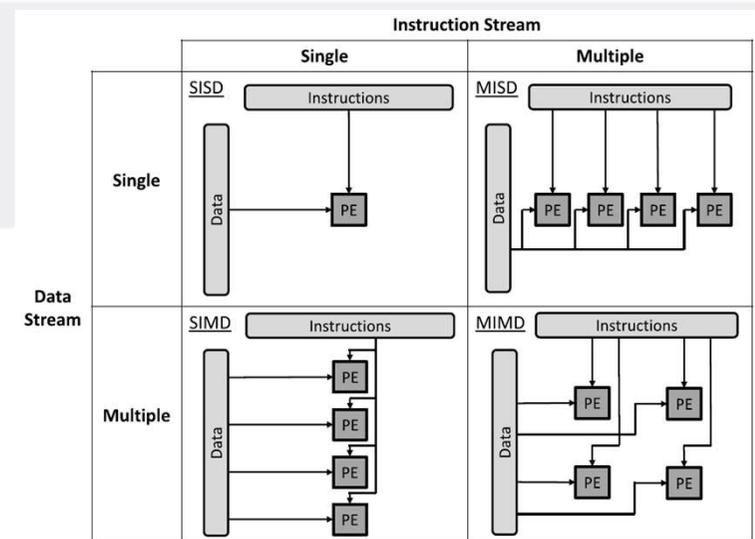


Figure 2.13. Interface lines of a generic peripheral input/output unit with three internal registers.

Taxonomia de Flynn

STR

- Introdução
- Processador
- Memória
- Periféricos
- Considerações gerais
- Referências



- Classificação de arquiteturas paralelas:
- **Single-Instruction, Single-Data (SISD):**
 - 1 elemento de processamento com acesso a 1 program e 1 simples Sistema armazenamento (von Neumann clássico)
- **Multiple-Instruction, Single-Data (MISD):**
 - múltiplos elementos processamento com memória própria mas somente 1 acesso a memória dados geral (em cada ciclo, os EP obtém o mesmo dado da memória e executa a instrução da sua memória programa própria). Não há modelos comerciais atualmente.
- **Single-Instruction, Multiple-Data (SIMD):**
 - múltiplos elementos processamento (EP) com memória privada e acesso a memória compartilhada (ou distribuída) para memória dados. Mas só 1 programa com 1 processador de controle para despachar instruções. Em cada ciclo máquina cada EP tem mesma instrução (do processador de controle) mas carrega dados diferentes.
- **Multiple-Instruction, Multiple-Data (MIMD):**
 - múltiplo EP com acesso independente a instruções e dados de uma memória compartilhada ou distribuída e em cada ciclo máquina, cada EP processa instruções e dados separados de forma assíncrona e armazena seus dados. Tecnologias multicore e sistemas de cluster são exemplos de MIMD.

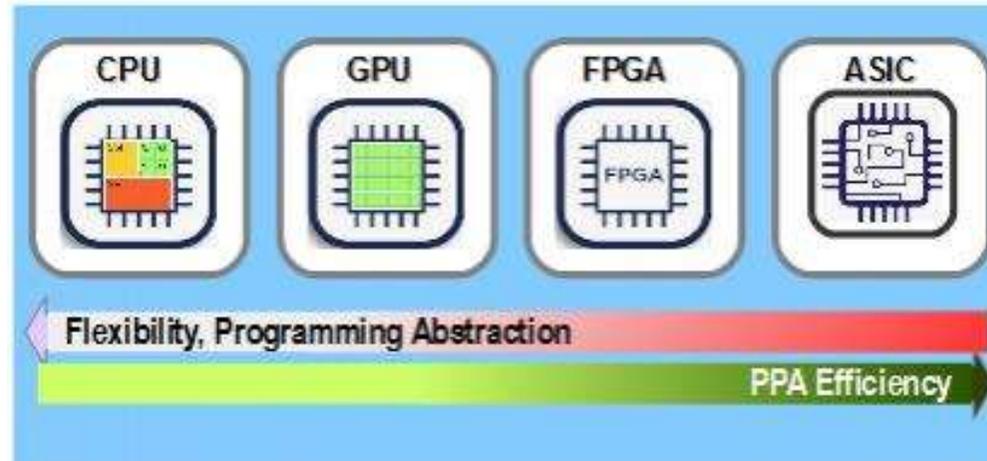


Considerações gerais

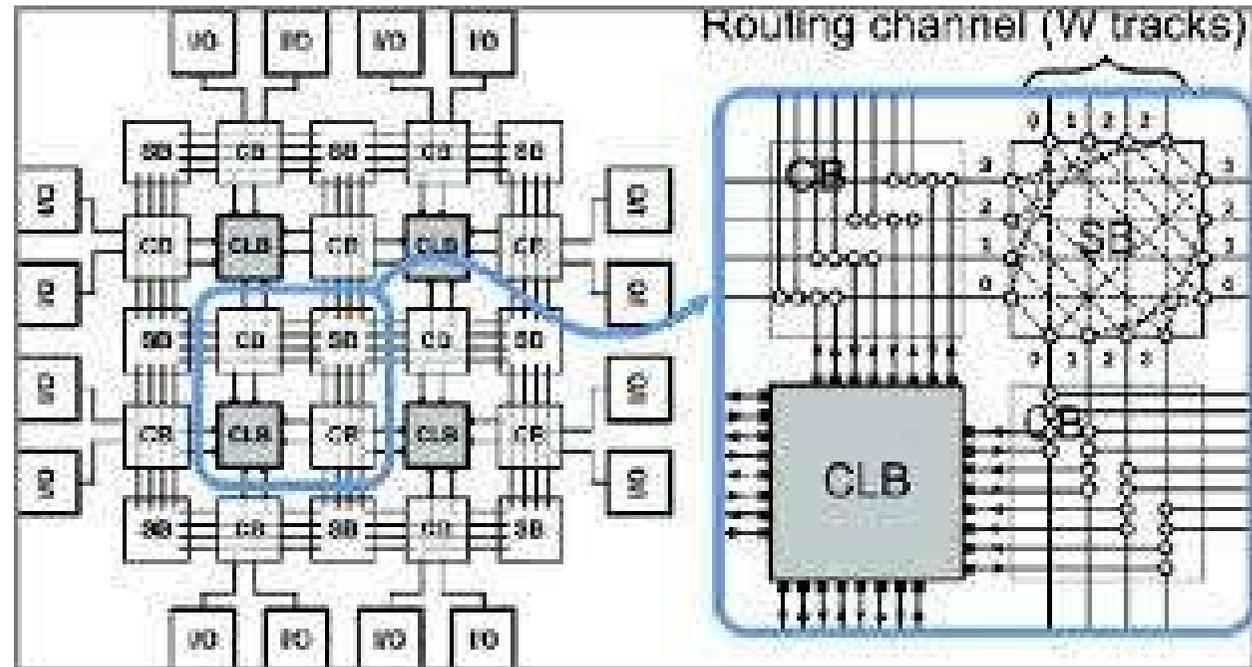
STR

- Introdução
- Processador
- Memória
- Periféricos
- Considerações gerais
- Referências

- Tecnologias de processamento:



- FPGA:



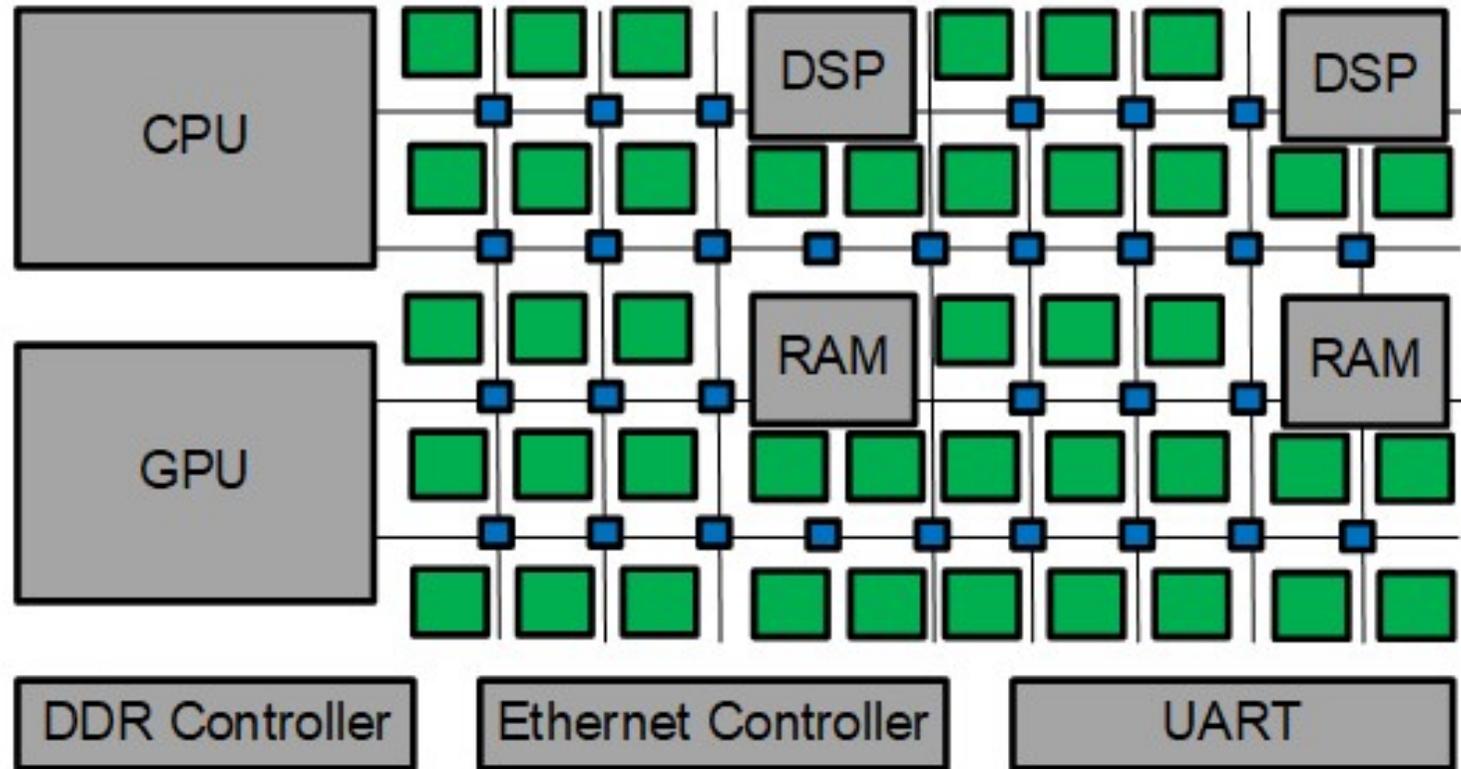


Considerações gerais

STR

- Introdução
- Processador
- Memória
- Periféricos
- Considerações gerais
- Referências

- Tendência: FPGA + SoM (*system on module*)



- Exemplo: <https://www.xilinx.com/products/som/kria/k26c-commercial.html#key-features>



Referências

STR

- Introdução
- Processador
- Memória
- Periféricos
- Considerações gerais
- Referências

- [1] Real-time systems design and analysis – tools for the practitioner
- [2] Parallel computer organization and design

