

Sistemas Embebidos



Multiprocessadores

1

Sumário

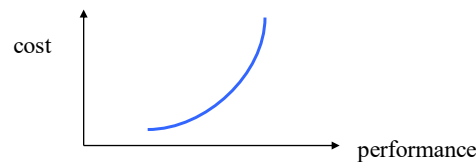


- Vantagens de multiprocessadores
- CPUs e aceleradores
- Análise da performance de multiprocessadores

2

Porquê Multiprocessadores?

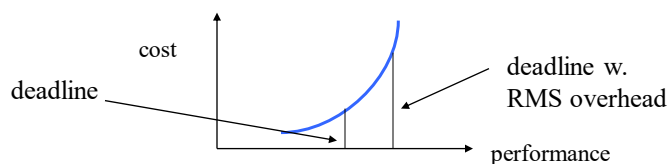
- Melhor relação custo/performance
 - Atribui-se a cada CPU um conjunto de tarefas ou usar lógica customizada (menor, mais barata).
 - O custo do CPU é uma função não-linear da performance



3

Porquê Multiprocessadores?

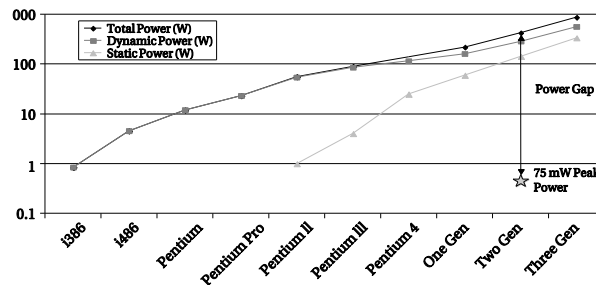
- Melhor performance tempo-real.
 - Colocar as funções de tempo-crítico em elementos de processamento menos sobrecarregados.
 - Uso de *Rate-Monotonic Scheduling (RMS)* ---ciclos de CPU extra devem ser reservados para cumprir deadlines



RMS - é um algoritmo de escalonamento usado nos sistemas de tempo real

4

Porquê Multiprocessadores?



- O uso de processadores especializados ou lógica customizada poupa energia
- Os uniprocessadores dos desktops não possuem eficiência energética suficiente para aplicações alimentadas por bateria

5

Porquê Multiprocessadores?

- Bom desempenho no processamento de I/O em tempo-real
- Podem consumir menos energia
- Melhor desempenho no *streaming* de dados (multimédia)
- A partilha da computação supera o desempenho da computação singular

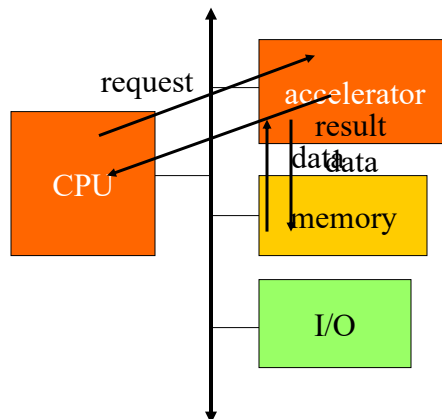
6

Sistemas aceleradores

- Uso de unidades computacionais dedicadas para algumas funções
 - Lógica *hardwired* (processamento logico, alternativo, exemplo: aceleradores)
 - CPU extra
- *Hardware/software co-design*: Conceção de arquiteturas de hardware e software combinadas. exemplo: sistema acelerado

7

Arquitetura de um sistema acelerado



Acelerador aparece no barramento do CPU como um elemento de processamento (PE)

8

Acelerador vs. co-processador

- Um co-processador executa instruções
 - Instruções despachadas pelo CPU
- Um acelerador surge como um dispositivo (*PE*) no *bus*
 - O acelerador é controlado por registos

9

Implementações de acelerador

- *IC application-specific* (*Chip* específico para a função)
- *Field-programmable gate array* (*FPGA*)
- Componente *standard*
 - Exemplo: processador gráfico (GPU)

FPGA - É um **circuito integrado** projetado para ser configurado após a fabricação (ex: chip programável).

10

Tarefas de conceção de sistemas

- Projetar uma arquitetura de multiprocessador heterogénea
 - Elementos de processamento (*PE*): *CPU*, acelerador, etc.
- Programar o sistema

11

Conceção de sistemas acelerados - etapas

- Inicialmente, determinar se o sistema realmente necessita de ser acelerado
 - Quão rápido é o acelerador na função core?
 - Qual o *overhead* provocado pelo *data transfer*?
- Projetar o acelerador
- Projetar o interface do CPU para o acelerador

12

Problemas de *caching*

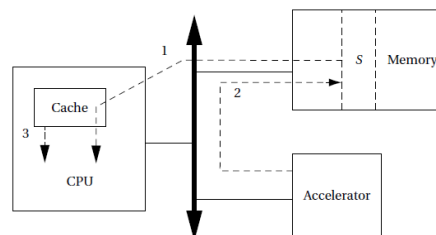
- A memória principal dispõe de um mecanismo de transferência de dados primário para o acelerador
- Os programas devem garantir que o *caching* não invalide dados da memória principal
 - O CPU pode utilizar dados inválidos através da cache, problema.

13

Sincronização

- Tal como na *cache*, a escrita (*write*) da memória principal na memória partilhada pode causar invalidação, exemplo:
 1. CPU lê a localização S
 2. Acelerador escreve na localização S -- **Não desejável**
 3. CPU lê novamente S, mas da cache

Solução possível:
Atualização da cache



14

Análise de performance de multiprocessadores

- Efeitos do paralelismo (e falta dele):
 - Processos
 - CPU e bus
 - Múltiplos processadores

15

***Speedup* do acelerador**

- Parâmetro crítico *speedup*: quanto mais rápido é o sistema com o acelerador?
- Deve ter-se em consideração:
 - Tempo de execução do acelerador
 - Tempo de transferência de dados
 - Sincronização com o *CPU* mestre

16

Tempo de execução do acelerador

- Tempo total de execução do acelerador:

$$t_{\text{accel}} = t_{\text{in}} + t_x + t_{\text{out}}$$

Data input

Accelerated
computation

Data output

17

Speedup do acelerador

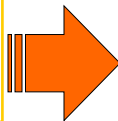
- Assumindo que um *loop* é executado n vezes.
- Comparação de sistema acelerado com sistema não acelerado:

$$S = n(t_{\text{CPU}} - t_{\text{accel}})$$

$$= n[t_{\text{CPU}} - (t_{\text{in}} + t_x + t_{\text{out}})]$$

Tempo de execução do CPU

Exemplo:
Tempo CPU = 100ns
Tempo S.Acel = 90ns
RESULTADO ?



Temos uma vantagem, que pode ser relativa! Uma vez que o custo de implementação do acelerador e consumo energético têm de ser considerados!

18

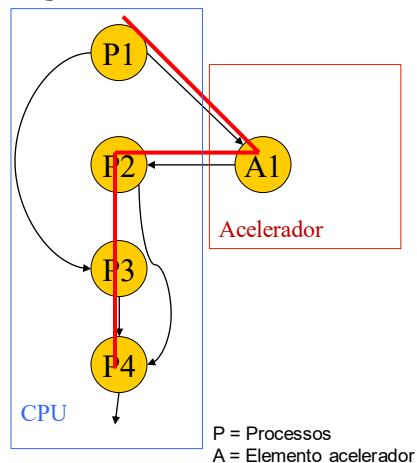
Single- vs. multi-threaded

- Um fator crítico é o paralelismo disponível:
 - *Single-threaded/blocking*: O CPU espera pelo acelerador
 - *Multi-threaded/non-blocking*: O CPU continua
- Para *multi-thread*, o CPU deve ter trabalho útil para fazer.
 - O software deve suportar também *multithreading*.

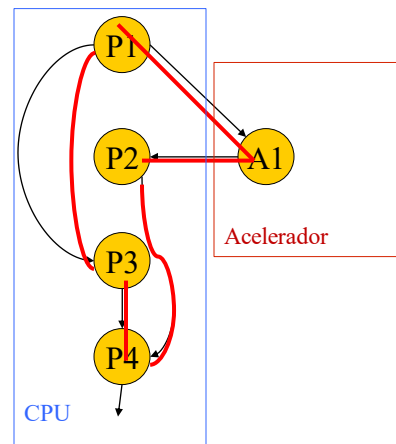
19

Tempo total de execução

□ *Single-threaded*:



□ *Multi-threaded*:



20

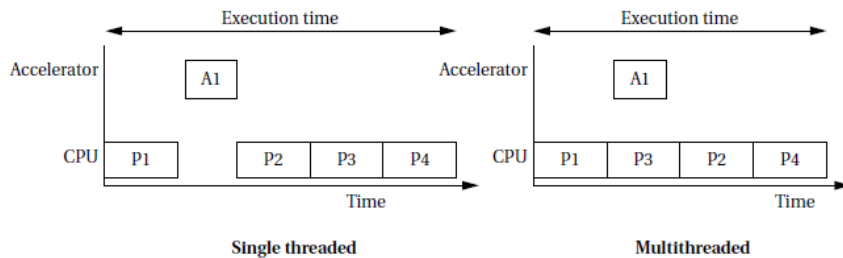
Análise do tempo de execução

□ *Single-threaded:*

- Considera o tempo de execução de todos os processos

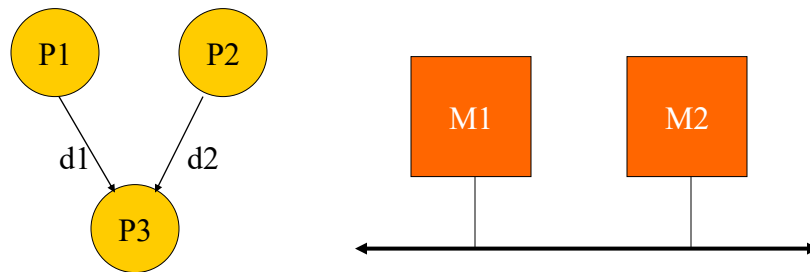
□ *Multi-threaded:*

- Considera o path de execução mais longa



21

Exemplo: Escalonamento e alocação



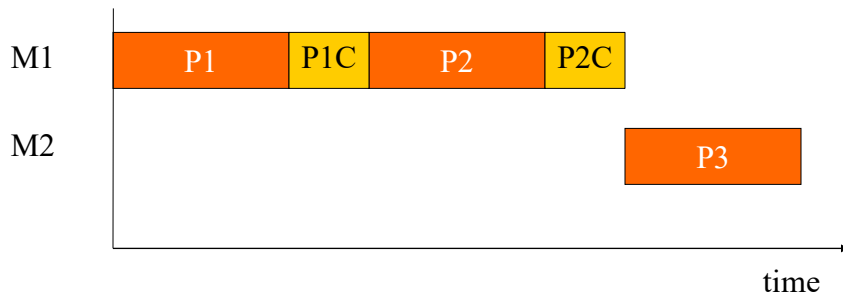
Grafo de tarefa

Plataforma de Hardware

22

Primeira abordagem

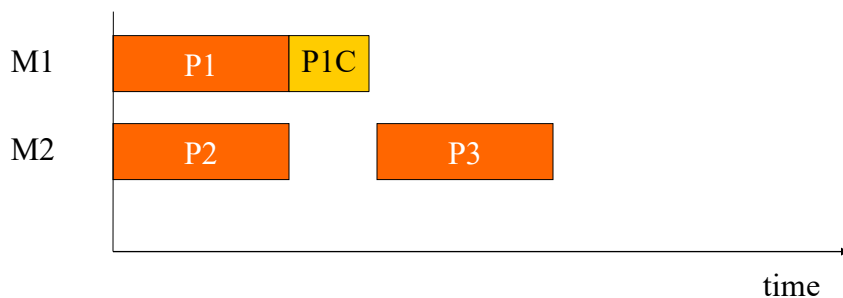
- Alocação P1, P2 -> M1; P3 -> M2.



23

Segunda abordagem

- Alocação P1 -> M1; P2, P3 -> M2:

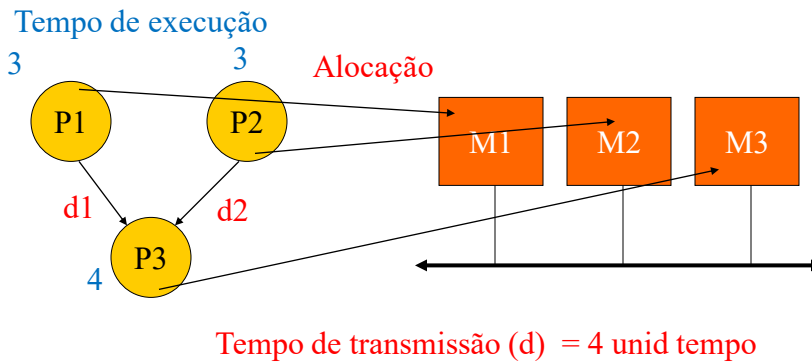


24

Exemplo: Ajuste de mensagens para reduzir o atraso

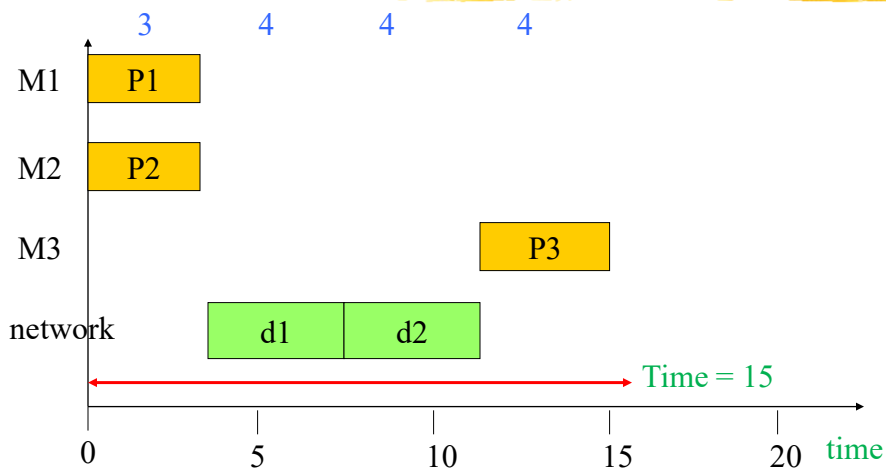
□ Grafo de tarefa:

□ Rede:



25

Escalonamento Inicial



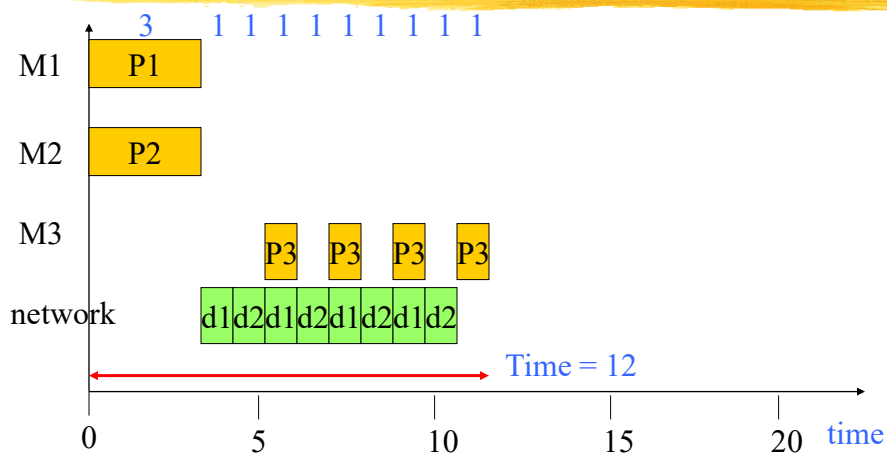
26

Novo escalonamento

- ❑ Modificar P3:
 - ❑ Ler um pacote de d1, um pacote de d2
 - ❑ Computar resultado parcial
 - ❑ Continuar para o próximo pacote

27

Novo escalonamento



28

Exercício 1

- Assumindo que o tempo de execução do CPU para uma dada função com um loop de 100 ciclos foi de 450ns e o acelerador teve um tempo de execução, leitura e escrita de 2ns, 1ns e 2ns por ciclo respectivamente.
- Verifique se a aceleração neste sistema foi vantajosa?

29

Exercício 2

- Numa dada função com um loop de 1000 ciclos, o tempo normal de execução do CPU foi de 6ns por ciclo e o sistema acelerado teve um tempo de execução de 1ns, de leitura 1ns e de escrita 3ns por ciclo.
- Analise se o sistema acelerado foi claramente mais eficiente, porquê?

30

Sistemas Embebidos



Resolução da FT6