Desempenho

Arquitetura de Organização de Computadores

Profº Erick de A. Barboza

Previously...

UNIVERSIDADE FEDERAL DE ALAGOAS

Código C

```
if (i == h)
  i  = g + h;
else
  i = g - h;
```

g em $s1, h em $s2, i em $s3

Código Assembly MIPS

```
bne $s3, $s2, Else
add $s3, $s1, $s2
j Exit
Else: sub $s3,$s1,$s2
Exit:...
```

Perguntas que Devem ser Respondidas ao Final do Curso

- Como um programa escrito em uma linguagem de alto nível é entendido e executado pelo HW?
- Qual é a interface entre SW e HW e como o SW instrui o HW a executar o que foi planejado?
- **O que determina o desempenho de um programa** e como ele pode ser melhorado?
- Que técnicas um projetista de HW pode utilizar para melhorar o desempenho?

Analisando o Desempenho

UNIVERSIDADE FEDERA DE ALAGOAS

- Desempenho é um importante fator de qualidade de um sistema
- Análise difícil
  - Tamanho e complexidade de softwares atuais
  - Técnicas sofisticadas utilizadas no projeto do hardware para aumento de desempenho
- Diferentes métricas podem ser utilizadas dependendo do tipo de aplicação

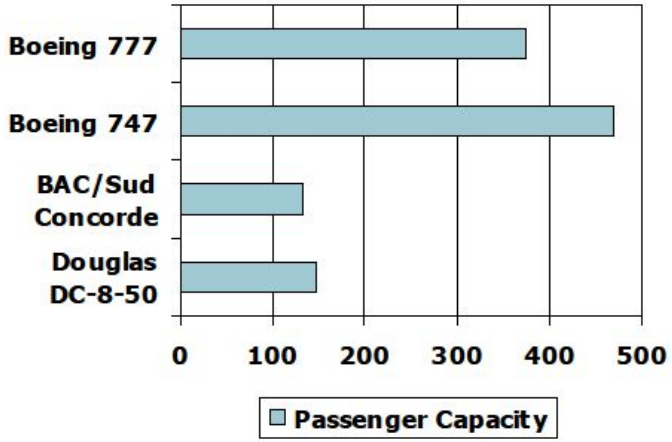
Fatores Que Influenciam Desempenho

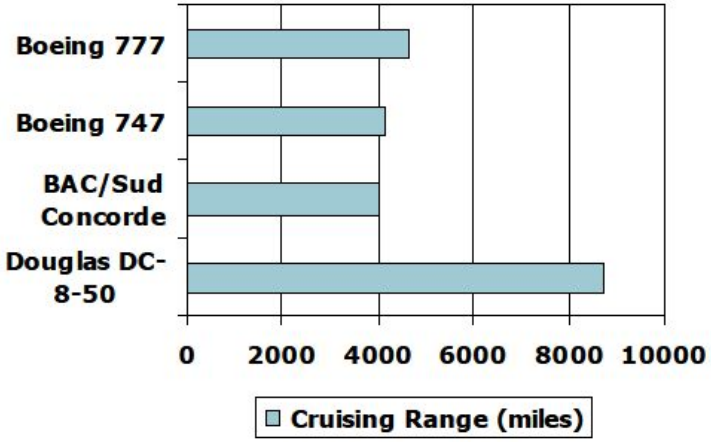
- Algoritmo
  - Determina número de operações executadas
- Linguagem de Programação, compilador, arquitetura
  - Determina número de instruções de máquina executadas por operação
- Processador e estrutura de memória
  - Determina velocidade em que instruções são executadas
- E/S (incluindo Sistema Operacional)
  - Determina velocidade em que operações de E/S são executadas

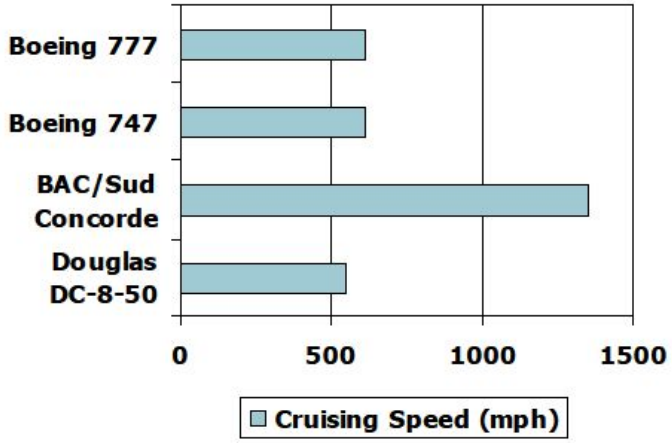
Definindo Desempenho

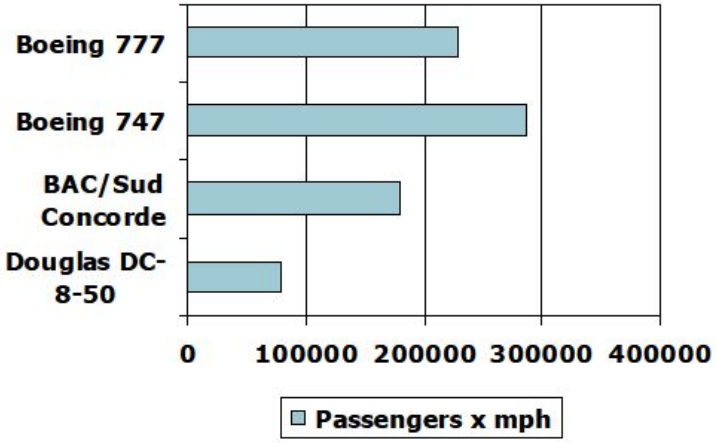
UNIVERSIDADE FEDERA DE ALAGOAS

Qual avião tem o melhor desempenho?









Diferentes Métricas

- Tempo de Execução ou Resposta
  - Tempo que leva para completar uma tarefa
  - Importante para o usuário comum de um sistema
- Throughput
  - Trabalho total feito por unidade de tempo
    - Exemplos: tarefas ou transações por segundo ou hora
  - Importante para análise de máquinas servidoras
  - Frequentemente afetado pelo tempo de execução

Analisaremos fatores que afetam tempo de execução

Diferentes Métricas

As seguintes mudanças em um computador aumentam o *throughput* , diminuem o tempo de resposta, ou ambos?

1. Trocar o processador por uma versão mais rápida?
2. Adicionar processadores adicionais em um sistema que utiliza múltiplos processadores para tarefas distintas -- por exemplo, processar imagens.

Desempenho relativo

- Desempenho = 1/Tempo de Execução
- “X é *n* vezes mais rápido que Y”

Desempenho_x / Desempenho_y
= $\text{Tempo de Execução}_y$ /$\text{Tempo de Execução}_x$ = *n*

- Exemplo: tempo para rodar um programa
  - 10s em A, 15s em B
  - $\text{Tempo de Execução}_B$ / $\text{Tempo de Execução}_A$ = 15s / 10s = 1.5
  - Então A é 1.5 vezes mais rápido que B

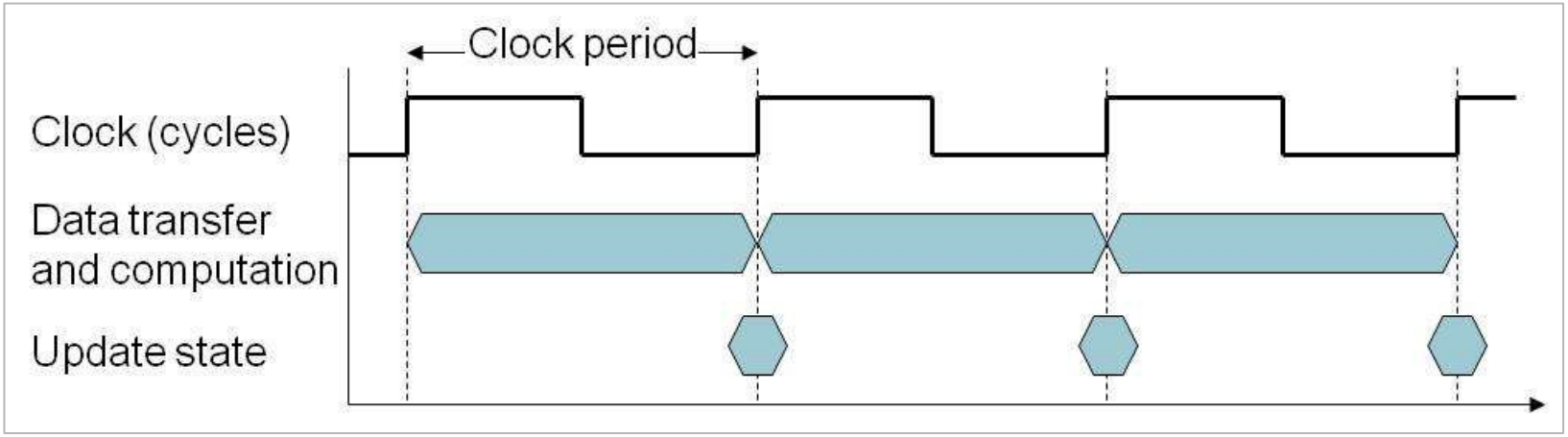
Medindo Tempo de Execução

- Tempo Gasto (*Elapsed time*)
  - Tempo total de resposta, incluindo todos os aspectos
    - Processamento, E/S, overhead de sistema operacional, tempo esperando algum evento
  - Determina desempenho do sistema

- Tempo de CPU (*CPU Time*)
  - Tempo de processamento de um programa
  - Não leva em conta tempo de E/S, execução de outros programas, etc

Clock

- Operações de uma CPU são governadas pelo clock



- Período do clock: duração de um ciclo de clock
  - Ex: 250ps = 0,25ns = $0,25 \times 10^{-9}$s
- Frequência do Clock : ciclos por segundo
  - Ex: 4,0GHz = 4000MHz = $4,0 \times 10^{9}$Hz

Tempo de CPU

Tempo de CPU = Ciclos de Clock (utilizados pela CPU) x Período de Clock

$$= \frac{\text{Ciclos de Clock}}{\text{Frequência}}$$

- Desempenho pode ser melhorado
  - Reduzindo número de ciclos de clock
  - Aumentando frequência do clock
- Projetista de Hardware deve frequentemente fazer o *trade off* entre frequência e número de ciclos

Tempo de CPU: Exemplo

- Computador A
  - Frequência de clock: 2GHz, Tempo de CPU: 10s
- Projetando Computador B
  - Objetivo: Tempo de CPU = 6s
  - Maior frequência do clock possível, mas isto causa um aumento no número de ciclos em 1,2x
- **Qual deve ser então a frequência do clock para atingir o tempo de CPU necessário?**

$$\text{FrequênciaB} = \frac{\text{Ciclos de Clock}_B}{\text{Tempo de CPU}_B} = \frac{1{,}2 \times \text{Ciclos de Clock}_A}{6s}$$

$$\text{Ciclos de ClockA} = \text{Tempo de CPU}_A \times \text{Frequência}_A$$
$$= 10s \times 2GHz = 20 \times 10^9$$

$$\text{FrequênciaB} = \frac{1{,}2 \times 20 \times 10^9}{6s} = \frac{24 \times 10^9}{6s} = 4GHz$$

Número de Instruções e CPI

Ciclos de Clock = Nº de Instruções x Ciclos por Instrução (CPI)

Tempo de CPU = Nº de Instruções x CPI x Período de Clock

$$= \frac{\text{Nº de Instruções x CPI}}{\text{Frequência}}$$

- Nº de Instruções de um programa
  - Determinado pelo algoritmo, ISA e compilador
- Quantidade média de ciclos por instrução
  - Determinado pelo hardware da CPU
  - Se diferentes instruções têm diferentes CPIs
    - Depende do que essas instruções fazem

CPI: Exemplo

- Computador A: Tempo do ciclo = 250ps, CPI = 2,0
- Computador B: Tempo do ciclo = 500ps, CPI = 1,2
- Mesma ISA
- **Qual é o mais rápido e quão mais rápido?**

Tempo de CPU_A = Nº de Instruções x CPI_A x Tempo do $Ciclo_A$
= N x 2,0 x 250ps = N x 500ps

A é mais rápido

Tempo de CPU_B = Nº de Instruções x CPI_B x Tempo do $Ciclo_B$
= N x 1,2 x 500ps = N x 600ps

$$\frac{\text{Tempo de CPU}_B}{\text{Tempo de CPU}_A} = \frac{N \times 600ps}{N \times 500ps} = 1,2$$

20% mais rápido

Mais detalhes sobre CPI

- Se diferentes classes de instruções levam diferentes números de ciclos

$$Ciclos\ de\ Clock = \sum_{i=1}^{n} (CPI_i \times N_{Instruções_i})$$

- Média ponderada da CPI

$$CPI = \frac{Ciclos\ de\ Clock}{N_{Instruções}} = \sum_{i=1}^{n} \left(CPI_i \times \frac{N_{Intruções_i}}{N_{Intruções}} \right)$$

Mais um exemplo de CPI

- Uma mesma sequência de código foi compilado de 2 formas diferentes usando instruções das classes A, B, C

Classe	A	B	C
CPI da classe	1	2	3
Nº de Instruções na sequência 1	2	1	2
Nº de Instruções na sequência 2	4	1	1

Nº de Instruções na sequência 1 = 5

Ciclos de Clock
= 2x1 + 1x2 + 2x3
= 10
Média CPI = 10/5 = 2,0

Nº de Instruções na sequência 2 = 6

Ciclos de Clock
= 4x1 + 1x2 + 1x3
= 9
Média CPI = 9/6 = 1,5

Desempenho: Resumindo

$$\text{Tempo de CPU} = \frac{\text{Instruções}}{\text{Programa}} \times \frac{\text{Ciclos de Clock}}{\text{Instrução}} \times \frac{\text{Segundos}}{\text{Ciclo de Clock}}$$

- Desempenho depende de:
  - Algoritmo: afeta nº de instruções, possivelmente CPI
  - Linguagem: afeta nº de instruções, CPI
  - Compilador: afeta nº de instruções, CPI
  - ISA: afeta nº de instruções, CPI, período do clock

Número de Instruções como Métrica de Desempenho

- **MIPS: Million Instructions Per Second**
- Algumas vezes usada como métrica de desempenho
  - Máquinas rápidas → valor de MIPS alto
- MIPS especifica a taxa de execução das instruções

$$MIPS = \frac{\text{Nr. Instruções}}{\text{Tempo Execução} \times 10^6} = \frac{\text{Frequencia}}{\text{CPI} \times 10^6}$$

- Podemos relacionar tempo de execução a MIPS

$$\text{Tempo Execução} = \frac{\text{Nr. Inst.}}{\text{MIPS} \times 10^6} = \frac{\text{Nr. Inst.} \times \text{CPI}}{\text{Frequencia}}$$

Problemas em usar MIPS

- Não leva em conta a diferença de instruções
  - Não é possível usar o MIPS dos computadores para comparar classes de operações diferentes, porque a contagem de instruções será diferente
- MIPS varia entre os programas no mesmo computador
  - Um computador pode não ter uma classificação MIPS única para todos os programas
- MIPS pode variar inversamente com o desempenho
  - Uma classificação elevada do MIPS nem sempre significa melhor desempenho

MIPS Exemplo

- Dois compiladores diferentes estão sendo testados no mesmo programa para uma máquina de 4 GHz com três classes diferentes de instruções: Classe A, Classe B e Classe C, que exigem 1, 2 e 3 ciclos, respectivamente.

Classe	A	B	C
CPI da classe	1	2	3
Nº de Instruções compilador 1	5×10^9	1×10^9	1×10^9
Nº de Instruções compilador2	1×10^{10}	1×10^9	1×10^9

- Qual compilador produz código com o melhor tempo de execução?
- Qual compilador produz código com o MIPS superior?

Solução do MIPS exemplo

Primeiro, encontramos os ciclos de CPU para ambos os compiladores

$$Ciclos_{compilador1} = (5\times1 + 1\times2 + 1\times3)\times10^9 = 10\times10^9$$
$$Ciclos_{compilador2} = (10\times1 + 1\times2 + 1\times3)\times10^9 = 15\times10^9$$

Depois, calculamos o tempo de execução para ambos os compiladores

$$Tempo\ Execução_{compilador1} = 10\times10^9 \text{ cycles} / 4\times10^9 \text{ Hz} = 2.5 \text{ sec}$$
$$Tempo\ Execução_{compilador2} = 15\times10^9 \text{ cycles} / 4\times10^9 \text{ Hz} = 3.75 \text{ sec}$$

Compilador 1 gera o programa mais rápido

Agora computamos a taxa MIPS para ambos compiladores

$$MIPS = Nr.Instruções / (Tempo\ Execução \times 10^6)$$

$$MIPS\ (compilador\ 1) = (5+1+1) \times 10^9 / (2.5 \times 10^6) = 2800$$
$$MIPS\ (compilador\ 2) = (10+1+1) \times 10^9 / (3.75 \times 10^6) = 3200$$

Compilador 2 gera o programa com MIPS mais elevado

Como podem se comparados diferentes computadores?

- Benchmarks
  - Conjunto de aplicações padrões que são utilizados para avaliar o desempenho
  - Vários domínios:
    - Científicos
    - Banco de dados
    - Processamento de sinais
    - Rede
- SPEC (System Performance Evaluation Cooperative)
  - Iniciativa criada pela indústria para criar benchmarks para avaliar seus computadores

Benchmarks da SPEC

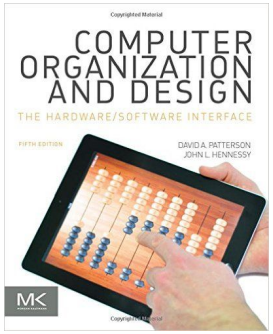
- Benchmarks usando inteiros
  - Escritos em C e C++
- Benchmarks usando ponto flutuante
  - Escritos em C e Fortran

Exemplo:
Intel Core i7 920 2,66 GHz

Description	Name	Instruction Count x 10^9	CPI	Clock cycle time (seconds x 10^{-9})	Execution Time (seconds)	Reference Time (seconds)	SPECratio
Interpreted string processing	perl	2252	0.60	0.376	508	9770	19.2
Block-sorting compression	bzip2	2390	0.70	0.376	629	9650	15.4
GNU C compiler	gcc	794	1.20	0.376	358	8050	22.5
Combinatorial optimization	mcf	221	2.66	0.376	221	9120	41.2
Go game (AI)	go	1274	1.10	0.376	527	10490	19.9
Search gene sequence	hmmer	2616	0.60	0.376	590	9330	15.8
Chess game (AI)	sjeng	1948	0.80	0.376	586	12100	20.7
Quantum computer simulation	libquantum	659	0.44	0.376	109	20720	190.0
Video compression	h264avc	3793	0.50	0.376	713	22130	31.0
Discrete event simulation library	omnetpp	367	2.10	0.376	290	6250	21.5
Games/path finding	astar	1250	1.00	0.376	470	7020	14.9
XML parsing	xalancbmk	1045	0.70	0.376	275	6900	25.1
Geometric mean	–	–	–	–	–	–	25.7

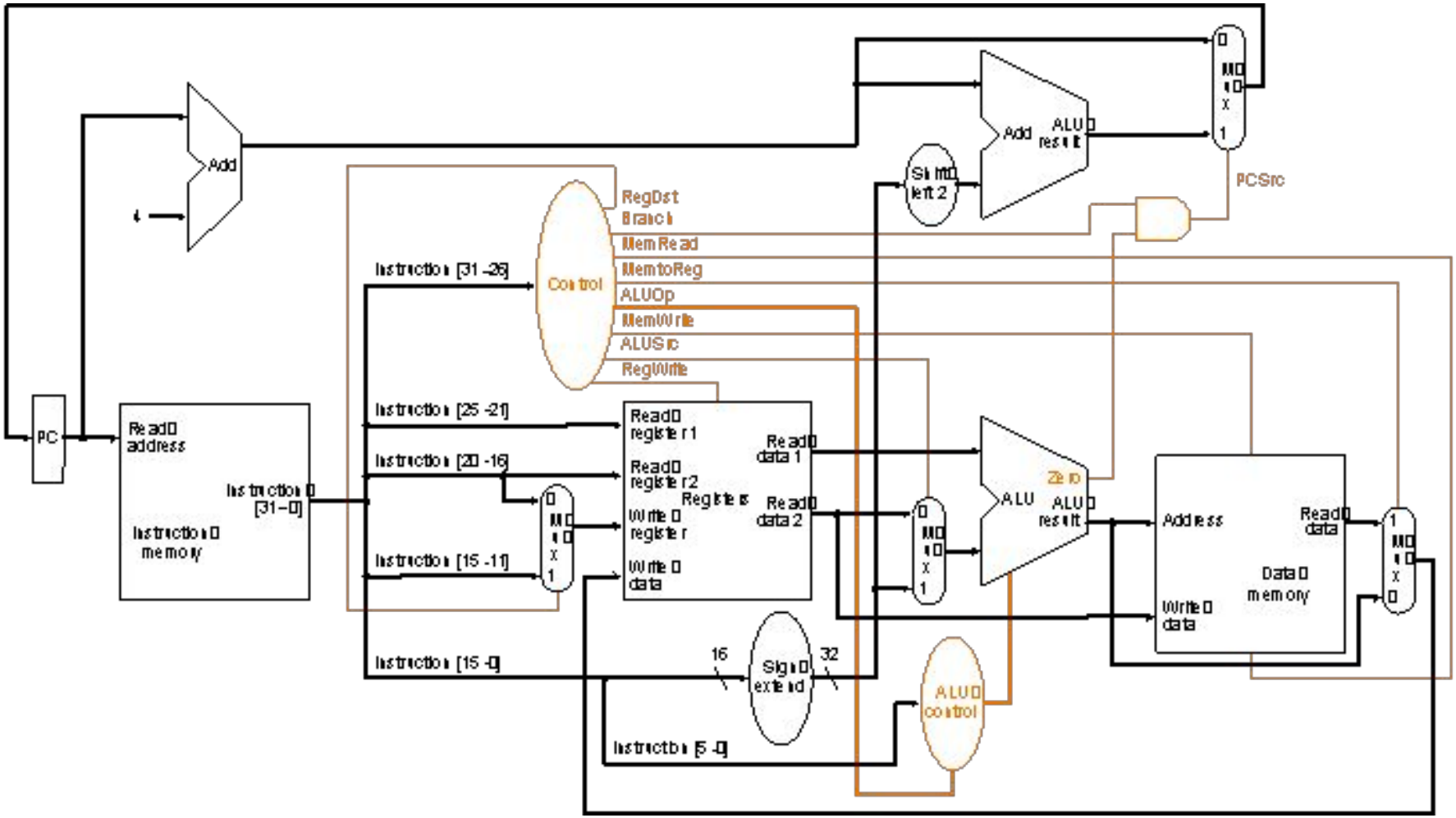
Atividades extra classe

- Leitura
  - Seção 1.6 do Patterson



- Exercícios indicados
  - Lista de exercício 2
    - Exercícios 5 e 6
  - Livro
    - Exercícios 1.5, 1.6 e 1.7

To be continued...



Dúvidas?