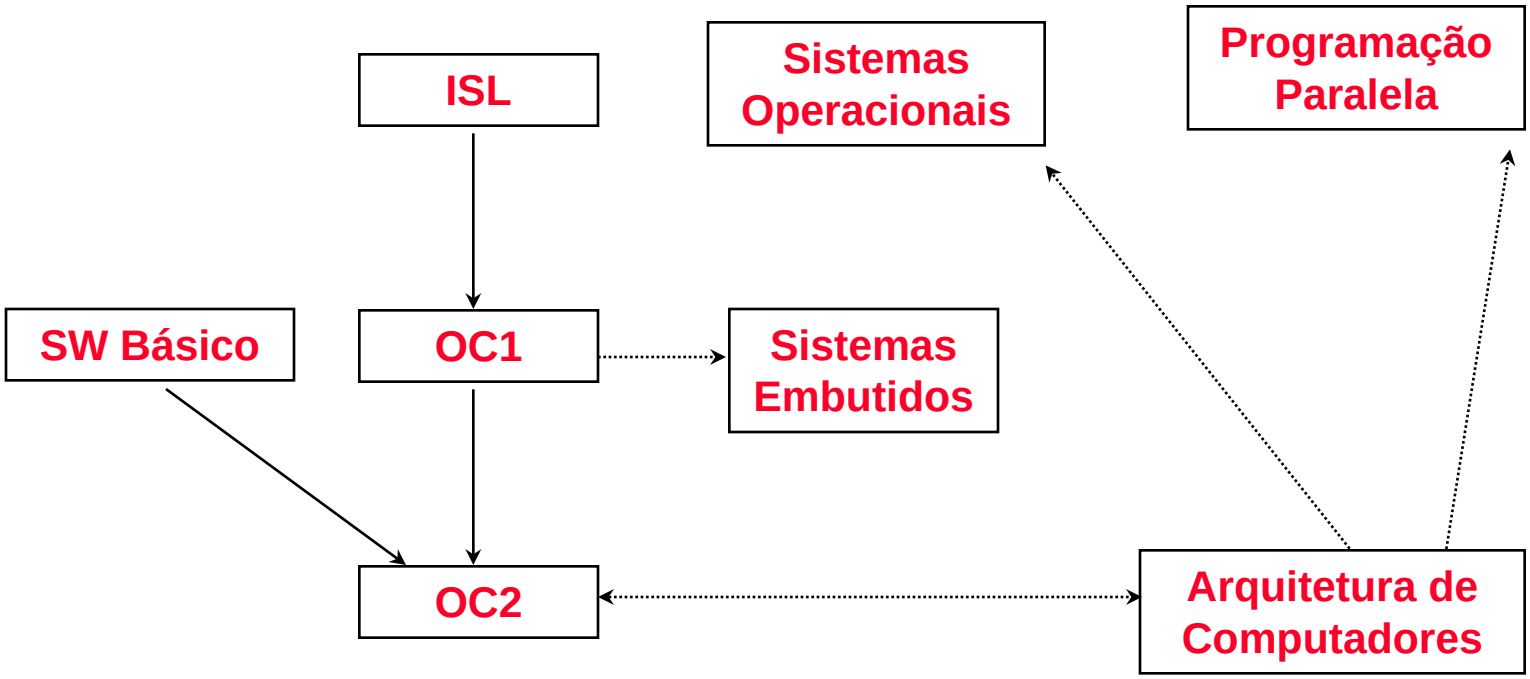
ISL

Introdução aos Sistemas Lógicos

1o Semestre de 2017

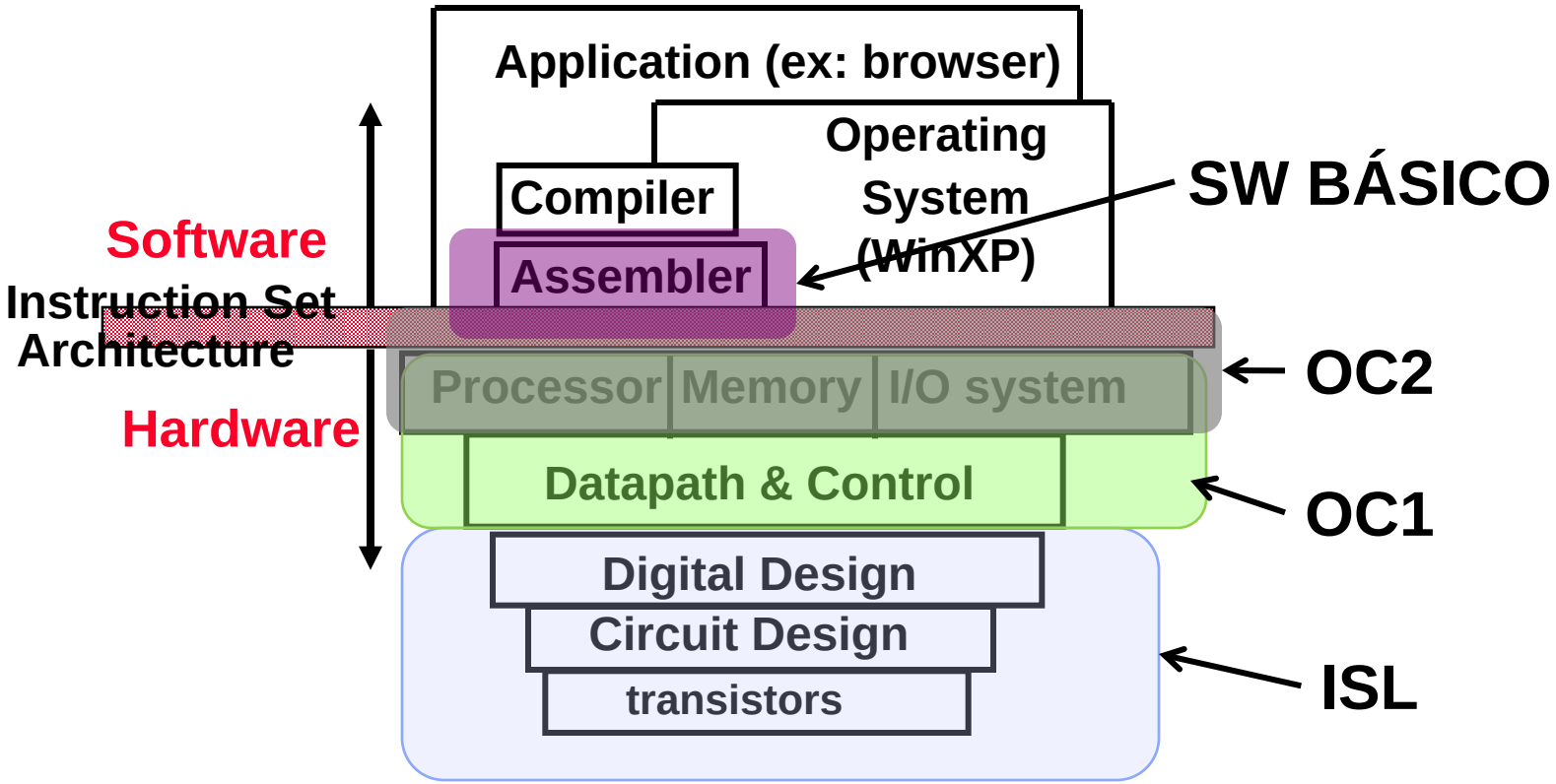
Luiz Filipe Menezes Vieira, PhD
lfvieira@dcc.ufmg.br

Cursos relacionados



Introdução

What are “Machine Structures”?



* Coordination of many

levels (layers) of abstraction

Levels of Representation

High Level Language Program (e.g., C)

Compiler

Assembly Language Program (e.g.,MIPS)

Assembler

Machine Language Program (MIPS)

Machine Interpretation

Hardware Architecture Description (Logic, Logisim, etc.)

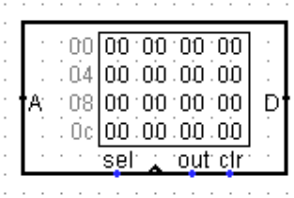
Architecture Implementation

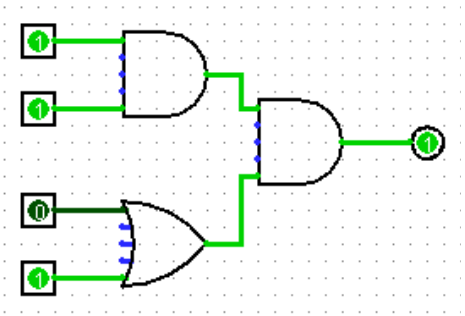
Logic Circuit Description (Logisim, etc.)

```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
lw   $t0, 0($2)
lw   $t1, 4($2)
sw   $t1, 0($2)
sw   $t0, 4($2)
```

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```





Livro Texto

- **Contemporary Logic Design**
  Randy Katz

Programa do Curso

- Introdução
- Lógica Combinatória de dois níveis
- Lógica Combinatória multi-nível
- Circuitos Lógicos Programáveis
- Lógica Seqüencial
- Máquina de Estados Finitos

Avaliação*

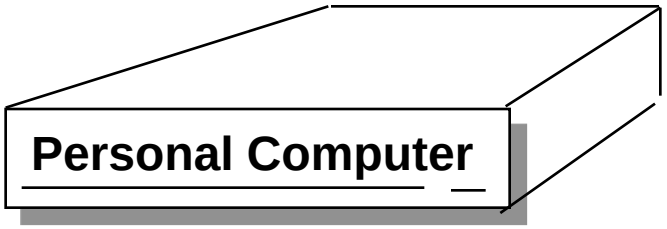
Atividade	Quant	Valor	Total
Listas de Exercícios	5	4	20
Trabalhos Práticos	4	5	20
Provas	2	30	60

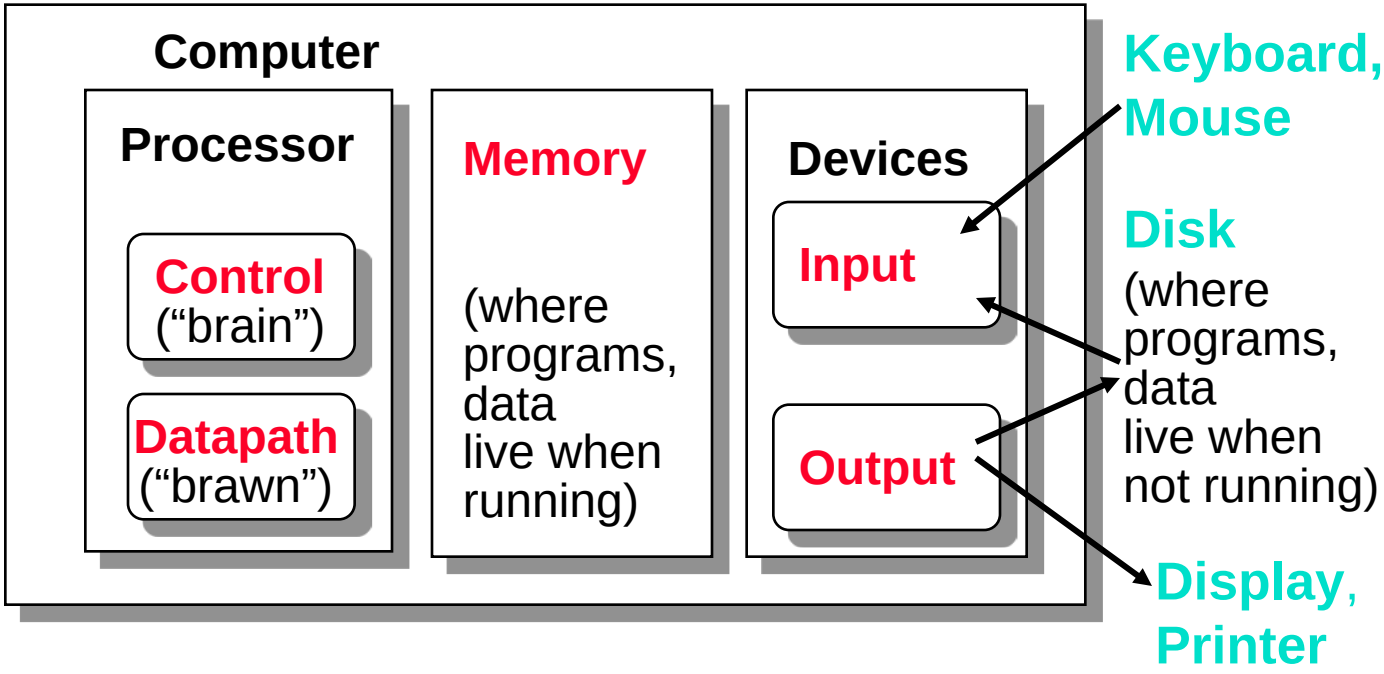
Avaliação

- **As listas de exercícios são individuais e deverão ser devolvidos no início da aula. O valor de lista ou trabalho será decrescido de 50% para cada 24 horas de atraso.**

- **Trabalhos práticos serão em grupos e deverão ser devolvidos no laboratório (sala 2023).**

Anatomy: 5 components of any Computer





Decimal Numbers: Base 10

Digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Example:

3271 =

$$(3 \times 10^3) + (2 \times 10^2) + (7 \times 10^1) + (1 \times 10^0)$$

Numbers: positional notation

- Number Base B ⇒ B symbols per digit:
  - Base 10 (Decimal): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
    Base 2 (Binary): 0, 1
- Number representation:
  - $d_{31}d_{30}$ ... d_1d_0 is a 32 digit number
  - value = $d_{31} \times B^{31} + d_{30} \times B^{30} + ... + d_1 \times B^1 + d_0 \times B^0$
- Binary: 0,1 (In binary digits called "bits")
  - 0b11010 = $1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0$
    = 16 + 8 + 2
    = 26
  - Here 5 digit binary # turns into a 2 digit decimal #
  - Can we find a base that converts to binary easily?

#s often written 0b…

Hexadecimal Numbers: Base 16

- **Hexadecimal:**
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
  - **Normal digits + 6 more from the alphabet**
  - **In C, written as 0x… (e.g., 0xFAB5)**
- **Conversion: Binary⇔Hex**
  - **1 hex digit represents 16 decimal values**
  - **4 binary digits represent 16 decimal values**

  ⇒1 hex digit replaces 4 binary digits
- **One hex digit is a "nibble". Two is a "byte"**
- **Example:**
  - **1010 1100 0011 (binary) = 0x_____ ?**

Decimal vs. Hexadecimal vs. Binary

Examples:

1010 1100 0011 (binary)
= 0xAC3

10111 (binary)
= 0001 0111 (binary)
= 0x17

0x3F9
= 11 1111 1001 (binary)

How do we convert between hex and Decimal?

00	0	0000
01	1	0001
02	2	0010
03	3	0011
04	4	0100
05	5	0101
06	6	0110
07	7	0111
08	8	1000
09	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

MEMORIZE!

Which base do we use?

- Decimal: great for humans, especially when doing arithmetic
- Hex: if human looking at long strings of binary numbers, its much easier to convert to hex and look 4 bits/symbol
  - Terrible for arithmetic on paper
- Binary: what computers use;
  you will learn how computers do +, -, *, /
  - To a computer, numbers is always binary
  - Regardless of how number is written:

    $32_{ten} == 32_{10} == 0x20 == 100000_{2} == 0b100000$
  - Use subscripts “ten”, “hex”, “two” in book, slides when might be confusing