

# TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

## PROGRAMAÇÃO DE COMPUTADORES I

### Prática 02: Organização e Arquitetura de Computadores / Lógica Digital (Parte I)

#### Lógica Digital – Nível 0



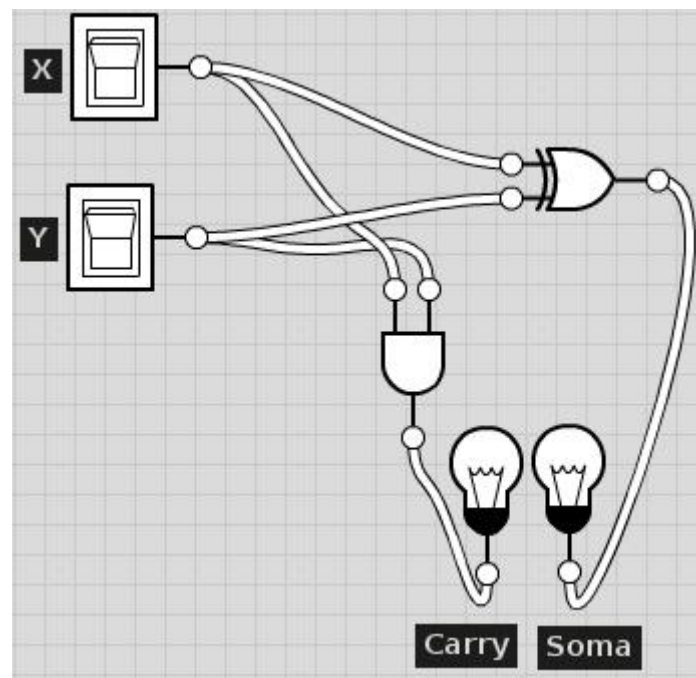
- A atividade prática aqui proposta deve ser efetuada através da utilização aplicativo de simulação de circuitos digitais disponível no link:  
<http://logic.ly/demo/>

#### Circuitos Combinacionais

Os circuitos Integrados (CI) como portas lógicas digitais podem ser combinados para fornecer circuitos mais complexos e úteis. Esses circuitos lógicos podem ser categorizados como lógica combinacional ou lógica sequencial.

##### 1. Circuito Semi-Somador

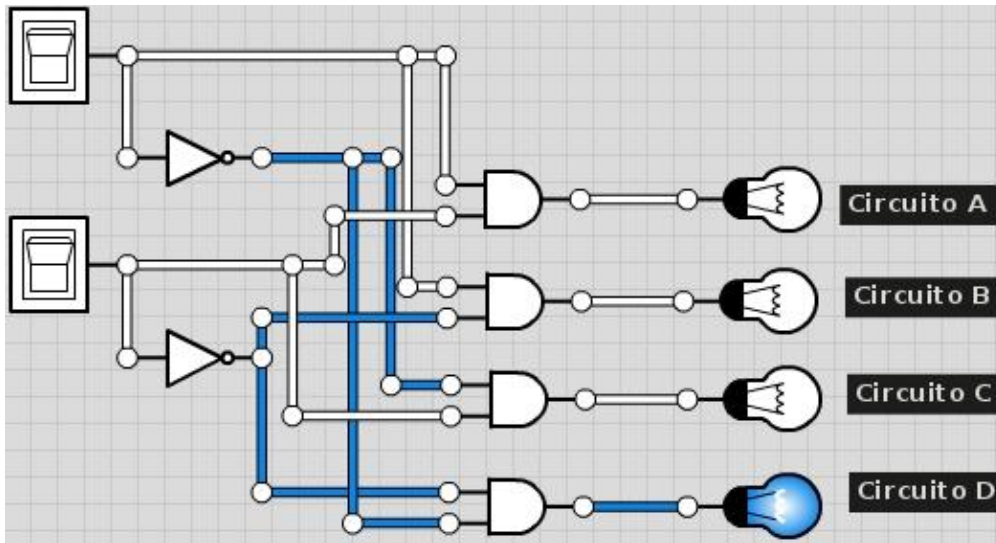
Um circuito semi-somador permite efetuar a soma (parcial) de dois números binários, utilizando para tal as portas lógicas AND e XOR.



##### 2. Decodificador

Um computador deve ser capaz de determinar qual chip (circuito eletrônico) deve ser ativado de acordo com as suas necessidades. Sendo assim,

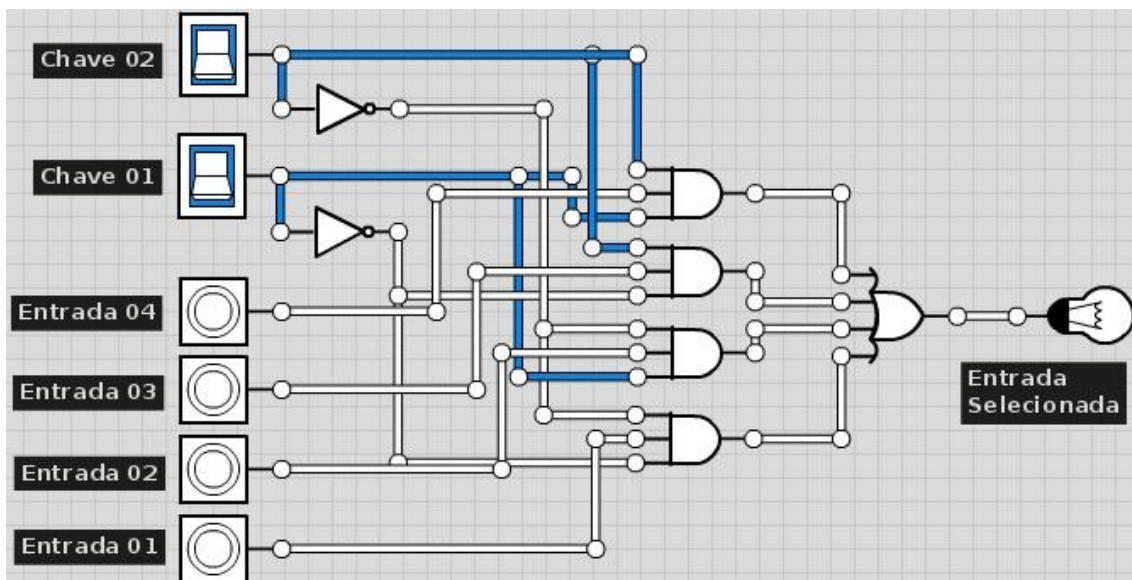
o decodificador é utilizado para determinar, de acordo com os dados (binários) colocados nas suas portas de entrada (x, y), qual dos circuitos (A, B, C, D) ligados a uma de suas saídas será ativado.



### 3. Multiplexador Digital

O multiplexador é um circuito muito comum dentro da eletrônica digital, ele tem como possibilitar a seleção de uma informação binária dentre várias linhas colocadas na entrada de um dado circuito eletrônico.

A seleção de qual entrada será lida para um dado momento é controlada por um conjunto de chaves (bits) de seleção, ou linhas de controle.



### 4. Relógio Digital – Exemplo Pronto

Efetue o download do circuito já criado, através do acesso ao link: <http://www.gileduardo.com.br/ifpr/ii/downloads/contador.dwm>, a seguir abra-o no software digital works e verifique seu funcionamento.

# TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

## PROGRAMAÇÃO DE COMPUTADORES I

### Prática 02: Arquitetura Von-Neumann / Memórias e Tipos de Memórias (Parte II)

#### Máquina/Instruções – SAVON (Níveis 1 e 2)



- A atividade prática aqui proposta deve ser efetuada através da utilização do software SAVON (Windows), apresentado em laboratório e disponível para download no link:

<http://www.gileduardo.com.br/ifpr/oac/downloads/savon.zip>

#### Informações sobre a ISA (Conjunto de Instruções da Arquitetura)

O conjunto de Instruções da Arquitetura aqui proposta é composto por:

- **SET**: coloca um valor digitado dentro de um registrador. Ex. **SET 2, Ra**, move o valor **2** para dentro de **Ra**. Na simulação observe que quando esta instrução é executada o valor do registrador **Ra** no canto superior esquerdo recebe o valor **2**.
- **MOV**: move valores entre registradores. Ex. **MOV Ra, Rb**, move o valor de **Ra** para dentro de **Rb**, digamos que **Ra** tinha o valor **1**, então após a execução desta instrução **Rb** também terá o valor **1**.
- **ADD**: Soma os valores armazenados em dois registradores especificados, colocando o resultado dentro do registrador **Ac**. Ex. **ADD Ra, Rb**, soma os valores de **Ra** e **Rb** colocando o resultado em **Ac**, considerando que **Ra=2** e **Rb=1**, então **Ac=3**.
- **SUB**: Subtrai os valores armazenados em dois registradores especificados colocando o resultado dentro do registrador **Ac**. Ex. **SUB Ra, Rb**, subtrai os valores de **Ra** e **Rb** colocando o resultado em **Ac**, considerando que **Ra=2** e **Rb=1**, então **Ac=1**.
- **MUL**: Multiplica os valores armazenados em dois registradores especificados colocando o resultado dentro do registrador **Ac**. Ex. **MUL Ra, Rb**, multiplica os valores de **Ra** e **Rb** colocando o resultado em **Ac**, considerando que **Ra=2** e **Rb=1**, então **Ac=2**.

- **DIV**: Divide os valores armazenados em dois registradores especificados colocando o resultado dentro do registrador **Ac**. Ex. **DIV Ra, Rb**, divide o valor de **Ra** pelo valor de **Rb** colocando o resultado em **Ac**, considerando que **Ra=4** e **Rb=2**, então **Ac=2**.

A Arquitetura proposta apresenta também, 4 registradores, sendo que 3 deles podem ser acessados diretamente, modificando os valores armazenados neles: **Ra**, **Rb** e **Rc**. O registrador **Ac** funciona apenas como repositório onde o resultado das operações matemáticas são colocados. Seu valor pode ser acessado (lido) apenas através da instrução **MOV**, que permite movê-lo para um dos outros 3 registradores quando este valor precisa ser utilizado.

O simulador disponibilizado possui um tamanho de memória limitado, contendo apenas 10 posições. Portanto podemos carregar um programa com no máximo 10 instruções.

### Simulações – Atividades

1. Efetue uma simulação que possua 3 instruções, ao seu final os registradores deverão apresentar os seguintes valores Ra=1, Rb=2 e Rc=3.
2. Efetue uma simulação que some os valores 2 e 6. Utilize para isso 4 instruções. Ao final da simulação os registradores deverão apresentar os valores Ra=1, Rb=6, Rc=2 e Ac=8.
3. Efetue uma simulação que divida o resultado da operação 2 + 3 + 4 (efetuar essa soma operações com instruções ADD) pelo valor 6. Para isso utilize 9 instruções sendo 3 instruções SET, 2 instruções ADD, 3 instruções MOV e 1 instrução DIV. Ao final da simulação os registradores devem conter os seguintes valores Ra=9, Rb=1, Rc=6 e Ac=1.
4. Efetue uma simulação que possua apenas 4 instruções, sendo uma instrução DIV, uma instrução MOV e duas instruções SET. Ao final da simulação os registradores apresentarão os seguintes valores Ra=2, Rb=3, Rc=7 e Ac=2.
5. Efetue uma simulação que multiplique os números 2 e 4, utilizando para isso exatamente 7 instruções, essas instruções podem ser apenas do tipo ADD, SET e MOV. Ao final da simulação os registradores devem conter os seguintes valores Ra=4, Rb=2, Rc=6 e Ac=8.

### Memória – Código-fonte em C



- A atividade prática aqui proposta deve ser efetuada através da utilização do código-fonte em linguagem C (linux), disponível para download no link: <http://www.gileduardo.com.br/ifpr/ii/downloads/memoria.c>