

ARQ I - ATIVIDADE PRÁTICA

PROJETO PARA AMPLIAÇÃO DE ARQUITETURA DE CONJUNTO DE INSTRUÇÕES (ISA)

Os arquivos solicitados nas atividades relacionadas abaixo deverão ser entregues no formato texto (TXT, **NÃO USAR** .doc ou .docx), devidamente identificado com nome e matrícula.

O arquivo com a definição de máquina (.cpu) deverá ser entregue junto com os programas.

O arquivo resumo (HTML), com as definições de máquina, poderá acompanhar as definições, mas não devem ser substituídas.

Os arquivos de programas em formato texto (TXT) com testes das instruções.

Exemplo de Projeto de Arquitetura de Conjunto de Instruções (ISA) com CPUSim

1.) Acrescentar microinstrução

Modify

|_ **Microinstruction**

|___ New

Type of Microinstruction			Logical	
name	type	source1	source2	destination
<i>acc&mdr->main</i>	<i>AND</i>	<i>acc</i>	<i>mdr</i>	<i>acc</i>

OK

2.) Acrescentar instrução de máquina

Modify

|_ **Machine instruction**

New

Editing machine instructions					
name	opcode	format	Implementation		Existing micros
<i>AND</i>	<i>D</i>	<i>op addr</i>	<i>ir(4-15)->mar</i>	<<insert<<	TransferRtoR _ <i>ir(4-15)->mar</i>
			<i>Main(mar)->mdr</i>	<<insert<<	Memory _ <i>Main(mar)->mdr</i>
			<i>acc&mdr->main</i>	<<insert<<	Logical _ <i>acc&mdr->main</i>
			End	<<insert<<	 _ End

3.) Testar a nova instrução de máquina

; This program tests logical operation AND

```
Main:  load  zero    ; acc<-0
        AND  one     ; acc<-acc&mdr
        write          ; output<-acc
        load  one     ; acc<-1
        AND  one     ; acc<-acc&mdr
        write          ; output<acc
        stop          ; stop
```

```
zero:   .data 2 0      ; 2-byte location for constant zero
one:     .data 2 1      ; 2-byte location for constant one
res:     .data 2 0      ; 2-byte location where result is stored
```

INSTRUÇÕES:

- 1.) Instalar o simulador CPUSim3.9.0.X.zip constante no pacote de programas da disciplina.
- 2.) Carregar e abrir o exemplo Wombat1.a na pasta SampleAssignments (usar Cpusim.bat).
DICAS: Fazer uma cópia de segurança (*backup*) dos arquivos originais, antes de alterá-los.
Se usar Linux, copiar o arquivo Cpusim.bat para Cpusim.sh,
editar o arquivo para trocar ponto-e-vírgula(';') por dois pontos(':') no *classpath*, e
conceder permissão para execução (*chmod +x Cpusim.sh*).
Consultar o arquivo Wombat1.html para um resumo das definições já efetuadas e,
ao final, gerar arquivo equivalente Wombat_new.html com as atualizações abaixo.
- 3.) Modificar o conjunto de instruções da linguagem de máquina para
 - a.) acrescentar uma operação lógica - NOT (opcode C (1100) / format: op unused)
editar, primeiro, a microinstrução equivalente (Logical->NOT)
 - b.) acrescentar uma operação lógica - NAND (opcode D (1110) / format: op addr)
editar a instrução equivalente (usando as microinstruções AND e NOT)

DICA: Os resultados serão aritméticos e estarão em complemento de 2,
deverão ser convertidos para os valores lógicos esperados (usar máscara).
- 4.) Desenvolver programas para
 - a.) testar a instrução NOT (mostrar caso com **true** e com **false**)
 - b.) testar a instrução NOR (tabela-verdade, por exemplo)
 - c.) montar e mostrar a tabela-verdade da instrução XOR usando NOR
 - d.) calcular e mostrar os oito primeiros números da série de Fibonacci
 - e.) calcular e mostrar o quadrado de um valor inteiro N usando a soma de ímpares.

EXTRA

Desenvolver opcionalmente programas para

5.) ordenar um arranjo em ordem decrescente pelo método da Bolha;

6.) procurar por valor em arranjo usando pesquisa sequencial rápida.

DICA: Supor que o número procurado está repetido depois da última posição válida (limite).

O objetivo é encontrar outra ocorrência do mesmo valor antes dessa posição.

Quando terminar, não deixar qualquer valor válido na posição usada como limite.