

ANEXO 1 - Conjunto de Instruções da Máquina *Wombat2*

É possível ver uma representação gráfica das instruções dentro do CPUSim. Para tanto, vá em *Modify -> Machine Instructions...* A ordem dos operandos de cada instrução (de cima para baixo) é definida a seguir.

Algumas instruções possuem bits não utilizados, os quais devem ser preenchidos com o valor zero, caso contrário, o programa não funcionará corretamente. A posição desses campos (se houverem) também é definida em cada instrução.

Código da operação (5 bits)	00 - exit
Significado	Encerra o programa, através da escrita da flag <i>halt-bit</i> .
Operandos	<ul style="list-style-type: none">• Bits não utilizados (valor zero) – 11 bits
Ação	N/A
Exemplo	<ul style="list-style-type: none">• exit• label: exit

Código da operação (5 bits)	01 - loadi
Significado	Carrega um dado da memória para um registrador. (OBS.: Caso o endereço especificado seja o de E/S, ou seja, 254, será solicitado ao usuário que digite um valor.)
Operandos	<ul style="list-style-type: none">• Registrador – 3 bits• Endereço de Memória – 8 bits
Ação	Registrador << Memória[endereço]
Exemplo	<ul style="list-style-type: none">• loadi R0 150• loadi R0 var ; 'var' alocada na memória com a pseudo-instrução <i>.data</i>• label: loadi R0 var

Código da operação (5 bits)	02 – storei
Significado	Escreve o valor de um registrador em um endereço de memória. (OBS.: Caso o endereço especificado seja o de E/S, ou seja, 254, o valor contido no Registrador 1 será impresso na tela.)
Operandos	<ul style="list-style-type: none">• Registrador – 3 bits• Endereço de Memória – 8 bits
Ação	Registrador >> Memória[endereço]

Exemplo	<ul style="list-style-type: none"> • storei R0 150 • storei R0 var ; 'var' alocada na memória com a pseudo-instrução <i>.data</i> • label: storei R0 var
----------------	---

Código da operação (5 bits)	03 - add
Significado	Faz a soma dos valores de dois registradores e escreve o resultado no primeiro.
Operandos	<ul style="list-style-type: none"> • Registrador 1 – 3 bits • Registrador 2 – 3 bits • Bits não utilizados (valor zero) – 5 bits
Ação	Registrador 1 << Registrador 1 + Registrador 2
Exemplo	<ul style="list-style-type: none"> • add R0 R1 • label: add R0 R1

Código da operação (5 bits)	04 - subtract
Significado	Faz a subtração dos valores de dois registradores e escreve o resultado no primeiro.
Operandos	<ul style="list-style-type: none"> • Registrador 1 – 3 bits • Registrador 2 – 3 bits • Bits não utilizados (valor zero) – 5 bits
Ação	Registrador 1 << Registrador 1 - Registrador 2
Exemplo	<ul style="list-style-type: none"> • subtract R0 R1 • label: subtract R0 R1

Código da operação (5 bits)	05 - multiply
Significado	<ul style="list-style-type: none"> • Faz a multiplicação dos valores de dois registradores e escreve o resultado no primeiro.
Operandos	<ul style="list-style-type: none"> • Registrador 1 – 3 bits • Registrador 2 – 3 bits • Bits não utilizados (valor zero) – 5 bits
Ação	Registrador 1 << Registrador 1 * Registrador 2
Exemplo	<ul style="list-style-type: none"> • multiply R0 R1 • label: multiply R0 R1

Código da operação (5 bits)	06 - divide
Significado	Faz a divisão dos valores de dois registradores e escreve o resultado no primeiro.
Operandos	<ul style="list-style-type: none"> • Registrador 1 – 3 bits • Registrador 2 – 3 bits • Bits não utilizados (valor zero) – 5 bits
Ação	Registrador 1 << Registrador 1 / Registrador 2
Exemplo	<ul style="list-style-type: none"> • divide R0 R1 • label: divide R0 R1

Código da operação (5 bits)	07 - jump
Significado	Pula para a instrução contida no endereço de memória especificado.
Operandos	<ul style="list-style-type: none"> • Bits não utilizados (valor zero) – 3 bits • Endereço de Memória – 8 bits
Ação	pc << endereço
Exemplo	<ul style="list-style-type: none"> • jump 50 • jump label • label: jump 50

Código da operação (5 bits)	08 - jmpz
Significado	Pula para a instrução contida no endereço de memória especificado, caso o valor contido no registrador seja igual a zero.
Operandos	<ul style="list-style-type: none"> • Registrador – 3 bits • Endereço de Memória – 8 bits
Ação	Se (Registrador = 0) pc << endereço
Exemplo	<ul style="list-style-type: none"> • jmpz R0 50 • jmpz R0 label • label: jmpz R0 50

Código da operação (5 bits)	09 - jmpn
Significado	Pula para a instrução contida no endereço de memória especificado, caso o valor contido no registrador seja menor que zero.
Operandos	<ul style="list-style-type: none"> • Registrador – 3 bits • Endereço de Memória – 8 bits
Ação	Se (Registrador < 0) pc << endereço
Exemplo	<ul style="list-style-type: none"> • jmpn R0 50 • jmpn R0 label • label: jmpn R0 50

Código da operação (5 bits)	10 - move
Significado	Copia o conteúdo do Registrador 2 para o Registrador 1.
Operandos	<ul style="list-style-type: none"> • Registrador 1 – 3 bits • Registrador 2 – 3 bits • Bits não utilizados (valor zero) – 5 bits
Ação	Registrador 1 << Registrador 2
Exemplo	<ul style="list-style-type: none"> • move R0 R1 • label: move R0 R1

Código da operação (5 bits)	11 - load
Significado	Carrega o endereço de memória contido no Registrador 2 para o Registrador 1. (OBS.: Caso o endereço especificado seja o de E/S, ou seja, 254, será solicitado ao usuário que digite um valor.)
Operandos	<ul style="list-style-type: none"> • Registrador 1 – 3 bits • Registrador 2 – 3 bits • Bits não utilizados (valor zero) – 5 bits
Ação	Registrador 1 << Memória[Registrador 2]
Exemplo	<ul style="list-style-type: none"> • load R0 R1 • label: load R0 R1

Código da operação (5 bits)	12 - store
Significado	Armazena o valor contido no Registrador 1 no endereço de memória contido no Registrador 2. (OBS.: Caso o endereço especificado seja o de E/S, ou seja, 254, o valor contido no Registrador 1 será impresso na tela.)
Operandos	<ul style="list-style-type: none"> • Registrador 1 – 3 bits • Registrador 2 – 3 bits • Bits não utilizados (valor zero) – 5 bits
Ação	Registrador 1 >> Memória[Registrador 2]
Exemplo	<ul style="list-style-type: none"> • store R0 R1 • label: store R0 R1

Código da operação (5 bits)	13 - loadc
Significado	Carrega uma constante de 8 bits (com sinal) em um registrador. A constante é armazenada nos 8 bits menos significativos do registrador.
Operandos	<ul style="list-style-type: none"> • Registrador – 3 bits • Constante – 8 bits
Ação	Registrador 1[8-15] << Constante
Exemplo	<ul style="list-style-type: none"> • loadc R0 150 • label: loadc R0 150

Código da operação (5 bits)	14 - clear
Significado	Zera o valor contido em um registrador.
Operandos	<ul style="list-style-type: none"> • Registrador – 3 bits • Bits não utilizados (valor zero) – 8 bits
Ação	Registrador << 0
Exemplo	<ul style="list-style-type: none"> • clear R0 • label: clear R0

Código da operação (5 bits)	15 - moveSp
Significado	Move o ponteiro de pilha (registrador <i>sp</i>) para cima (Constante negativa) ou para baixo (Constante positiva). Constante em bytes.
Operandos	<ul style="list-style-type: none"> • Bits não utilizados (valor zero) – 3 bits • Constante (com sinal) – 8 bits
Ação	$sp \ll sp + \text{constante}$
Exemplo	<ul style="list-style-type: none"> • moveSp 2 • label: moveSp 2 • moveSp -2 • label: moveSp -2

Código da operação (5 bits)	16 - slt
Significado	Escreve 1 no Registrador 1, caso o valor contido em Registrador 2 seja menor que o valor contido em Registrador 3, e escreve zero caso contrário.
Operandos	<ul style="list-style-type: none"> • Registrador 1 – 3 bits • Registrador 2 – 3 bits • Registrador 3 – 3 bits • Bits não utilizados (valor zero) – 2 bits
Ação	Se (Registrador 2 < Registrador 3) Registrador 1 \ll 1 Senão Registrador 1 \ll 0
Exemplo	<ul style="list-style-type: none"> • slt R0 R1 R2 • label: slt R0 R1 R2

Código da operação (5 bits)	17 - call
Significado	Chama o procedimento que está contido no endereço de memória especificado.
Operandos	<ul style="list-style-type: none"> • Bits não utilizados (valor zero) – 3 bits • Endereço de memória – 8 bits
Ação	$ra \ll pc$ (instrução seguinte à chamada) $pc \ll \text{Memória}[\text{endereço}]$
Exemplo	<ul style="list-style-type: none"> • call proc ;Procedimento identificado pelo label 'proc' • label: call proc

Código da operação (5 bits)	18 - loadSp
Significado	Carrega para o registrador o valor contido no endereço de memória deslocado a partir do ponteiro de pilha (<i>stack pointer</i>).
Operandos	<ul style="list-style-type: none"> • Registrador de destino – 3 bits • Deslocamento em relação ao registrador <i>sp</i> – 8 bits
Ação	Registrador \ll Memória[<i>sp</i> + deslocamento]
Exemplo	<ul style="list-style-type: none"> • loadSp R0 2 • label: loadSp R0 2

Código da operação (5 bits)	19 - storeSp
Significado	Carrega um valor contido em um registrador para o endereço de memória deslocado a partir do ponteiro de pilha (<i>stack pointer</i>).
Operandos	<ul style="list-style-type: none"> • Registrador de origem – 3 bits • Deslocamento em relação ao registrador <i>sp</i> – 8 bits
Ação	Registrador \gg Memória[<i>sp</i> + deslocamento]
Exemplo	<ul style="list-style-type: none"> • storeSp R0 2 • label: storeSp R0 2

Código da operação (5 bits)	20 - ret
Significado	Encerra um procedimento e retorna para o último endereço salvo, o qual deve estar contido no registrador <i>ra</i> .
Operandos	<ul style="list-style-type: none"> • Bits não utilizados (valor zero) – 11 bits
Ação	$pc \ll ra$
Exemplo	<ul style="list-style-type: none"> • ret

Código da operação (5 bits)	21 - loadRa
Significado	Carrega no Registrador <i>ra</i> o valor contido no endereço de memória deslocado a partir do ponteiro de pilha (<i>stack pointer</i>).
Operandos	<ul style="list-style-type: none"> • Bits não utilizados (valor zero) – 3 bits • Deslocamento em relação ao registrador <i>sp</i> – 8 bits
Ação	$ra \ll$ Memória[<i>sp</i> + deslocamento]
Exemplo	<ul style="list-style-type: none"> • loadRa 2

Código da operação (5 bits)	22 - storeRa
Significado	Carrega o valor contido no registrador <i>ra</i> para o endereço de memória deslocado a partir do ponteiro de pilha (<i>stack pointer</i>).
Operandos	<ul style="list-style-type: none"> • Bits não utilizados (valor zero) – 3 bits • Deslocamento em relação ao registrador <i>sp</i> – 8 bits
Ação	<i>ra</i> >> Memória[<i>sp</i> + deslocamento]
Exemplo	<ul style="list-style-type: none"> • storeRa 2

Código da operação (5 bits)	23 - addi
Significado	Adiciona ao valor de um registrador uma constante (positiva ou negativa).
Operandos	<ul style="list-style-type: none"> • Registrador de destino – 3 bits • Constante (com sinal) – 8 bits
Ação	Registrador << Registrador + constante
Exemplo	<ul style="list-style-type: none"> • addi R0 2 • addi R0 -2 • label: addi 2

Código da operação (5 bits)	24 - sgt
Significado	Escreve 1 no Registrador 1, caso o valor contido em Registrador 2 seja maior que o valor contido em Registrador 3, e escreve zero caso contrário.
Operandos	<ul style="list-style-type: none"> • Registrador 1 – 3 bits • Registrador 2 – 3 bits • Registrador 3 – 3 bits • Bits não utilizados (valor zero) – 2 bits
Ação	Se (Registrador 2 > Registrador 3) Registrador 1 << 1 Senão Registrador 1 << 0
Exemplo	<ul style="list-style-type: none"> • sgt R0 R1 R2 • label: sgt R0 R1 R2

Código da operação (5 bits)	25 - seq
Significado	Escreve 1 no Registrador 1, caso o valor contido em Registrador 2 seja igual ao valor contido em Registrador 3, e escreve zero caso contrário.
Operandos	<ul style="list-style-type: none"> • Registrador 1 – 3 bits • Registrador 2 – 3 bits • Registrador 3 – 3 bits • Bits não utilizados (valor zero) – 2 bits
Ação	Se (Registrador 2 = Registrador 3) Registrador 1 << 1 Senão Registrador 1 << 0
Exemplo	<ul style="list-style-type: none"> • seq R0 R1 R2 • label: seq R0 R1 R2

Código da operação (5 bits)	26 - jmp
Significado	Pula para a instrução contida no endereço de memória especificado, caso o valor contido no registrador seja maior que zero.
Operandos	<ul style="list-style-type: none"> • Registrador – 3 bits • Endereço de Memória – 8 bits
Ação	Se (Registrador > 0) pc << endereço
Exemplo	<ul style="list-style-type: none"> • jmp R0 50 • jmp R0 label • label: jmp R0 50