## **Usando o simulador MIPS**

O objetivo desta aula prática será a utilização do simulador MipsIt para executar programas escritos em linguagem de máquina do MIPS.

### 1 - Criando um projeto

Cada programa a ser executado deverá estar inserido dentro do contexto de um projeto. Para criar um projeto execute o programa MipsIt.exe.



Agora seu projeto acaba de ser criado. Devemos passar para a próxima etapa.

### 2 – Editando um programa em linguagem de máquina:

Agora devemos criar um arquivo assembly para que possamos observar o funcionamento do programa. Para tanto vá ao menu File e selecione a opção New. Neste momento a seguinte caixa de texto surgirá na tela.

New File Project   Text Sasembler C	I	Escolha a opção Assembler entre as três apresentadas. Na opção File name indique o nome que seu arquivo deve receber (neste
	C:\exercicio\	tutorial ele irá se chamar exemplo).

Use o programa abaixo para seguir neste tutorial. Copie-o e salve o arquivo e o projeto.

.word 5	5
.word 1	0
.word 5	5
.text	
.globl s	start
.ent sta	rt
lw	\$8, a
lw	\$9, b
lw	\$10, c
add	\$11, \$9, \$8
sub	\$11, \$11, \$10
SW	\$11,a
	word 5. word 1. word 5. text .globl s. .ent sta lw lw lw lw sub sw

.end start

Os construtores iniciando com ponto (.) são pseudo instruções ou diretivas que tem como função dar algumas indicações ao montador sem que gere código executável.

O construtor de montagem .*data* reserva área de memória para as variáveis *a*, *b* e *c*, que são do tipo inteiro de 32 bits (*.word*). O construtor *.text* delimita início de código e os construtores *.globl start* e *.ent start* declaram *start* como endereço simbólico de início de programa. A partir do endereço simbólico *start* tem-se um programa em linguagem de máquina. O construtor *.end start* delimita o fim do programa cujo endereço inicial simbólico foi declarado como *start*. Nossa próxima etapa é compilar o programa acima.

#### 3- Compilando um programa em assembly.

Para compilar o programa escolha a opção compile no menu *build*. Após corrigir os erros de sintaxe, o código executável poderá ser gerado através da opção *build* no menu *build*. Os códigos objeto e executável serão armazenados no diretório de projeto.

Mipslt - [exemplo.s]           File         Edit         View         Project	Build Window Help		
exemplo.s	Compile exemplo.s Ctrl+F7 Build exercicio F7 Rebuild All Clean Update All Dependencies	← → ۞ ☑ ☆ uso do MipsIt	Use as opções do menu Build para compilar o programa e gerar o executável
<pre>x Compiling i exemplo.s Linking Post build Done</pre>		Quando a aparecerem esquerdo é próximo pass	mensagens ao lado no canto inferior hora de passar a 50.

Atenção: se o se o programa acima não compilar pode ser útil observar o tópico configurações que se encontra logo abaixo neste tutorial.

### 4- Carregando código executável para o simulador

Antes de carregar seu programa para o simulador o programa Mips.exe deverá ser executado. O carregamento para o simulador será feito através do programa MipsIt como está indicado na figura abaixo. Após o carregamento a janela do simulador é aberta automaticamente.

Mipslt - [exemplo.s]         File       Edit       View       Project         Image: Ima	Build Window Help Compile exemplo.s Ctrl+F7 Build exercicio F7 Rebuild All Clean Update All Dependencies	← → Ə 🗹 lso do MipsIt	Use o menu Build e na opção Upload indique To Simulator. Neste momento a janela do simulador deve abrir.
	Upload 🕨	To Hardware C	trl+F5
	C: .WOLG 3	To Simulator	F5
	.text .globl start	File to Hardware.	

Devemos, neste momento, passar para a manipulação do simulador.

#### 5- Usando o simulador.

Uma vez que o código foi carregado no simulador, a janela de simulação é aberta e o usuário pode verificar os registradores (clicando no desenho da CPU) ou a memória (clicando no desenho RAM). Outros componentes como memória cache e

dispositivos de E/S poderão ser visualizados, mas estão fora do escopo desta etapa inicial. Clique apenas na CPU e na RAM.

O quadro de registradores mostra o conteúdo de todos os registradores em notação hexadecimal e para cada registrador tem-se o número do mesmo e o nome simbólico. Além dos registradores de propósito geral é possível visualizar alguns registradores específicos como o PC.

A visualização da memória inclui quatro colunas: endereço, conteúdo em hexadecimal, rótulos e instrução de máquina simbólica. Com isto pode-se reconhecer as instruções digitadas no programa exemplo. Veja as figuras abaixo para obter detalhes.

dit View Cpu Help					Memory				
o h⊘l r⊊l ⊾l m					Address	Content	Label		
8 Nº 0 1 10					8001FFC8	00 00 00	00	NOP	
					8001FFCC	00 00 00	00	NOP	
					8001FFD0	00 00 00	00	NOP	
					8001FFD4	00 00 00	00	NOP	
					8001FFD8	00 00 00	00	NOP	
					8001FFDC	00 00 00	00	NOP	
					8001FFE0	00 00 00	00	NOP	
					B001FFE4	00 00 00	00	NOP	
					80015528	00 00 00	00	NOP	
Pagistars					BOOIFFEC	00 00 00	00	NOP	
r0/zero=000000000	r1/at =00000000	x2/x0 =0		/w1 =00000000	80015554	00 00 00	00	NOP	
r4/a0 =00000000	r5/a1 =00000000	x6/a2 =0	00000000 23	/a3 =00000000	80017778	00 00 00	00	NOP	
r8/t0 =00000000	r9/t1 =00000000	r10/t2 =0	00000000 ±1	1/t3 =00000000	8001FFFC	00 00 00	00	NOP	
r12/t4 =00000000	r13/t5 =00000000	r14/t6 =0	00000000 r1	5/t7 =00000000	80020000	3C 08 80	02 startf)	LUI	\$08. 0x8002
r16/s0 =00000000	r17/s1 =00000000	r18/s2 =0	00000000 r1	9/#3 =00000000	80020004	8D 08 00	30	LW	\$08, 0x30(\$08)
r20/s4 =00000000	r21/s5 =00000000	r22/s6 =0	00000000 r2	3/s7 =00000000	80020008	3C 09 80	02	LUI	\$09, 0x8002
r24/t8 =00000000	r25/t9 =00000000	r26/k0 =0	00000000 r2	7/k1 =00000000	8002000C	8D 29 00	34	LW	\$09, 0x34(\$09)
r28/gp =00000000	r29/sp =800bc000	r30/fp =0	00000000 r3	1/ra =bfc00088	80020010	3C 0A 80	02	LUI	\$10, 0x8002
					80020014	8D 4A 00	38	LW	\$10, 0x38(\$10)
pc =80020000	mdhi =00000000	mdlo =0	0000000 co	nf =00000000	80020018	01 28 58	20	ADD	\$11, \$09, \$08
bad va =00000000	status =00400000	cause =0	00000000 ep	c =00000000	8002001C	01 6A 58	22	SUB	\$11, \$11, \$10
		1			80020020	3C 01 80	02	LUI	\$01, 0x8002
	— I – ('a	chd			80020024	AC 2B 00	30	SW	\$11, 0x30(\$01)
		UIIQ			80020028	00 00 00	00	NOP	
					8002002C	00 00 00	00	NOP	
					80020030	00 00 00	05 a:	???	
			ם ו	71 M I I	Address mode: Vi	rtual	View mode:	Assembler	
			I K	AM II	CONSOL	ei			
			-		0011001	<b>~</b>			
		esque	erda no	demos vi	sualizar o	s regio	stradore	s	
		esque	Po po		Junizar 0	5 1051			
			nnõem	a CPU	a à dir	eita v	emos	a 🚺	
	q		npoem		c a un	una v	cinos	u	



### 6- Executando um programa passo a passo

Chegou a hora de executar o programa que foi apresentado. Para tanto siga as figuras abaixo.

ile Edit View	Cpu Hel	p			
2 2 2	Step				
	Run Stop Reset ( Reset	Clock			
				aha	
Registers				aha	
Registers	=0000000	0 r1/at	<u>م</u> =00000000	2/v0	=
Registers r0/zeros r4/a0	=0000000	0 r1/at 0 r5/a1	<u>م</u> =00000000 =00000000	2/v0 r6/a2	
Registers r0/zeros r4/a0 r8/t0 r12/r4	=0000000	0 r1/at 0 r5/a1 0 r9/t1	D C ∽ =00000000 =00000000 =00000000	2/v0 r6/a2 r10/t2	

Após ter ido para o rótulo de inicio como indicado na seção anterior, escolha a opção *Step* dentro do menu *CPU* do simulador Mips.exe ou clique no botão com uma seta apontando para quadrado azul.



Após todos esses passos chegou à hora de executarmos o programa. Faça-o e veja o que acontece...

					8001FFDC 8001FFE0	00 00 00 00	NOP		
					8001FFE8	00 00 00 00	NOP		
	In Car	aha			8001FFEC	00 00 00 00	NOP		
Registers				×	8001FFF0	00 00 00 00	NOP		
r0/zero=00000000	r1/at =00000000	r2/v0 =00000000	r3/v1 =000000	00	8001FFF4	00 00 00 00	NOP		
r4/a0 =00000000	r5/a1 =00000000	r6/a2 =00000000	r7/a3 =000000	00	8001FFF8	00 00 00 00	NOP		
r8/t0 =00000005	r9/t1 =0000000a	r10/t2 =80020000	r11/t3 =000000	00	8001FFFC	00 00 00 00	NOP		
r12/t4 =00000000	r13/t5 =00000000	r14/t6 =00000000	r15/t7 =000000	00 🕴	80020000	3C 08 80 02	start() LUI	\$08, 0x8002	
r16/s0 =00000000	r17/s1 =00000000	r18/s2 =00000000	r19/s3 =000000	00	80020004	8D 08 00 30	LW	\$08, 0x30(\$0	08)
r20/s4 =00000000	r21/s5 =00000000	r22/s6 =00000000	r23/s7 =000000	00	80020008	3C 09 80 02	LUI	\$09, 0x8002	
r24/t8 =00000000	r25/t9 =00000000	r26/k0 =00000000	r27/k1 =000000	00	8002000C	8D 29 00 34	LW	\$09, 0x34(\$0	09)
r28/gp =00000000	r29/sp =800bc000	r30/fp =00000000	r31/ra =bfc000	88	80020010	3C 0A 80 02	LUI	\$10, 0x8002	
					80020014	8D 4A 00 38	LW	\$10, 0x38(\$1	10)
pc =80020014	mdhi =00000000	mdlo =00000000	conf =000000	00	80020018	01 28 58 20	ADD	\$11, \$09, \$0	08
bad va =00000000	status =00400000	cause =00000000	epc =000000	00	8002001C	01 6A 58 22	SUB	\$11, \$11, \$1	10
		1	1		80020020	3C 01 80 02	LUI	\$01, 0x8002	
		and			80020024	AC 28 00 30	SW	\$11, 0x30(\$0	91)
	I Val		_	-	80020028	00 00 00 00	NOP		
					8002002C	00 00 00 00	NOP		
				1	80020020	00 00 00 05	a. 222		
_			RAM	Ao regis	observ	var a es são	figura ao carregado	lado no os com o sequiência	otamos que os valores
				send	o segu	idas na	memória	sequencia	i de mstruço

### 7- Usando nome simbólico dos registradores

Para poder se usar o nome simbólico dos registradores em vez de seus números inclua o texto: include <Iregdef.h> no início do programa. Edite o arquivo exemplo.s incluindo esta linha e use os nomes dos registradores \$8= t0, \$9= t1 e \$10 = t2. Salve este arquivo, recompile e simule novamente. Mude os valores das variáveis, compile e simule novamente...

# 8 - Configurações

Agora vamos ver algumas configurações básicas que podem ser o diferencial no momento de compilar o programa. Em caso de alguma dúvida consulte os monitores.

Para começar, selecione o menu File do MipsIt e escolha a opção Options. Neste momento irá aparecer a janela abaixo. É nela que as alterações devem ser feitas.

ptions		
General Directories	1	
COM port: COM2		
COM speed: 38400		
Changes to COM setting take effect	gs need a restart of the application to	
V Upload symbols	Console size: 80 × 30	Em Compiler exeutable indique o endereço do programa xgcc.exe. Este executável deve estar na pasta bin.
Compiler exeutable:		
U:\Mipsit\bin\xgcc.exe	Browse	
	OK Cancelar	
Seneral Directories Sor: Show directories for: Executable files Directories: C:\exercicio\Dbjects\	▼ <sup>™</sup> × ↑ ↓	General Directories Show directories for: Include files Directories:
	OK Cancelar	OK Cancelar
	-	
	Use a opção <i>show direc</i> onde estão arquivos aux Siga a seguinte configur Executable files aponta p Include files aponta para Library files aponta para	<i>tories for</i> para selecionar os diretórios iliares do MipsIt. ação: para a pasta Objects do MipsIt. a o diretório include do MipsIt. a o diretório lib do MipsIt.

Após esta breve instrução ao simulador, resolva os exercícios da lista referente a essa aula e simule os programas propostos.