

LICENCIATURA EM ENGENHARIA INFORMÁTICA E DE COMPUTADORES



Introdução à Arquitetura de Computadores

Tutorial do P3

2013 / 2014

INSTITUTO SUPERIOR TÉCNICO

1. Conteúdo

1.	Conteúdo.....	1
2.	Introdução.....	2
3.	O Processador P3.....	2
	Conjunto de Instruções do P3.....	2
	Aritméticas	2
	Lógicas	2
	Deslocamento	2
	Controlo.....	3
	Transferência	3
	Genéricas	3
	Diretivas do Assemblador do P3	3
	Registos do P3	3
	Modos de endereçamento do P3	4
	Constantes no P3	4
4.	O Ambiente de Simulação do P3.....	5
	A Janela Principal.....	5
	A Janela de Texto	6
	Janela da Placa de IO.....	7
5.	Utilização do Assemblador.....	8
6.	Utilização do Simulador	9
7.	Ensaio de Utilização	11
	Começar uma sessão.....	11
	Introdução ao P3.....	11
	Introdução à linguagem de programação Assembly.....	13
8.	Utilização dos periféricos do P3.....	16
	Janela de texto	16
	Botões de pressão	17
	Interruptores.....	17
	Leds	17
	Display de 7 segmentos	17
	Display de cristal líquido ou LCD.....	18
	Máscara de interrupções	18
	Temporizador.....	18
9.	Exemplos de Código do P3	18
10.	Referências	19

2. Introdução

Este guia destina-se a dar uma introdução ao P3 e seu ambiente de desenvolvimento e simulação, mas não dispensa a leitura do manual do P3 [1].

3. O Processador P3

Conjunto de Instruções do P3

Pseudo	Aritméticas	Lógicas	Deslocamento	Controlo	Transferência	Genéricas
ORIG	NEG	COM	SHR	BR	MOV	NOP
EQU	INC	AND	SHL	BR.cond	MVBH	ENI
WORD	DEC	OR	SHRA	JMP	MVBL	DSI
STR	ADD	XOR	SHLA	JMP.cond	XCH	STC
TAB	ADDC	TEST	ROR	CALL	PUSH	CLC
	SUB		ROL	CALL.cond	POP	CMC
	SUBB		RORC	RET		
	CMP		ROLC	RETN		
	MUL			RTI		
	DIV			INT		

Aritméticas

- NEG – Determina o simétrico do operando
- INC – Incrementa o operando
- DEC – Decrementa o operando
- ADD – Adiciona os operandos
- ADDC – Adiciona com transporte
- SUB – Subtrai os operandos
- SUBB – Subtrai os operandos com transporte
- CMP – Compara os operandos
- MUL – Multiplica os operandos
- DIV – Divide os operandos

Lógicas

- COM – Complementa bit a bit o operando
- AND – Faz o “e” bit a bit dos operandos
- OR – Faz o “ou” bit a bit dos operandos
- XOR – Faz o “ou exclusivo” bit a bit dos operandos
- TEST – Faz o “e” bit a bit dos operandos mas não guarda o resultado

Deslocamento

- SHR – Deslocamento para a direita.
- SHL – Deslocamento para a esquerda.
- SHRA – Deslocamento aritmético para a direita.
- SHLA – Deslocamento aritmético para a esquerda.
- ROR – Rotação para a direita.
- ROL – Rotação para a esquerda.
- RORC – Rotação para a direita com transporte.

ROLC – Rotação para a esquerda com transporte.

Controlo

BR	– Salto relativo
BR.cond	– Salto relativo condicional
JMP	– Salto absoluto
JMP.cond	– Salto absoluto condicional
CALL	– Chamada a sub-rotina
CALL.cond	– Chamada a sub-rotina condicional
RET	– Retorno de sub-rotina
RETN	– Retorno de sub-rotina de remoção de argumentos da pilha
RTI	– Retorno de interrupção
INT	– Chamada a interrupção

Cond pode ser: O , NO, N, NN, C, NC, Z, NZ, I , NI, P, NP

Transferência

MOV	– Cópia de um operando para outro operando
MVBH	– Cópia da parte alta de um operando para a parte alta de outro operando
MVBL	– Cópia da parte baixa de um operando para a parte baixa de outro operando.
XCH	– Troca os valores dos operandos
PUSH	– Coloca o valor do operando na pilha
POP	– Remove um valor da pilha e copia para o operando

Genéricas

NOP	– Nenhuma operação
ENI	– Ativar as interrupções
DSI	– Desativar as interrupções
STC	– Ativa o bit de transporte
CLC	– Inativa o bit de transporte
CMC	– Complementa o bit de transporte

Diretivas do Assembler do P3

ORIG	– Indica a posição de memória onde o assembler deve colocar o código que se segue
EQU	– Atribui um valor constante a uma cadeia de caracteres do assembler
WORD	– Reserva um endereço de memória, e inicializa-o com um valor
STR	– Coloca uma cadeia de caracteres na memória
TAB	– Reserva o número de posições de memória indicados pelo operando

Registos do P3

R0-R7	– registos de uso genérico. O registo R0 não pode ser alterado e tem sempre o valor 0.
PC	– Registo contador de programa (program counter)
SP	– Registo apontador para o topo da pilha (stack pointer)

RE – registo de estado

Modos de endereçamento do P3

op – operando

Rx – registo Rx

W – constante de valor W

M[y] – referência à posição de memória com endereço y

PC – registo contador de programa (program counter)

SP – registo do apontador para o topo da pilha (stack pointer)

Constantes no P3

binário – 111111111111111b .

octal – 177777o .

decimal – 65456

hexadecimal – F45A6h .

alfanumérico – 'g'

4. O Ambiente de Simulação do P3

A Janela Principal

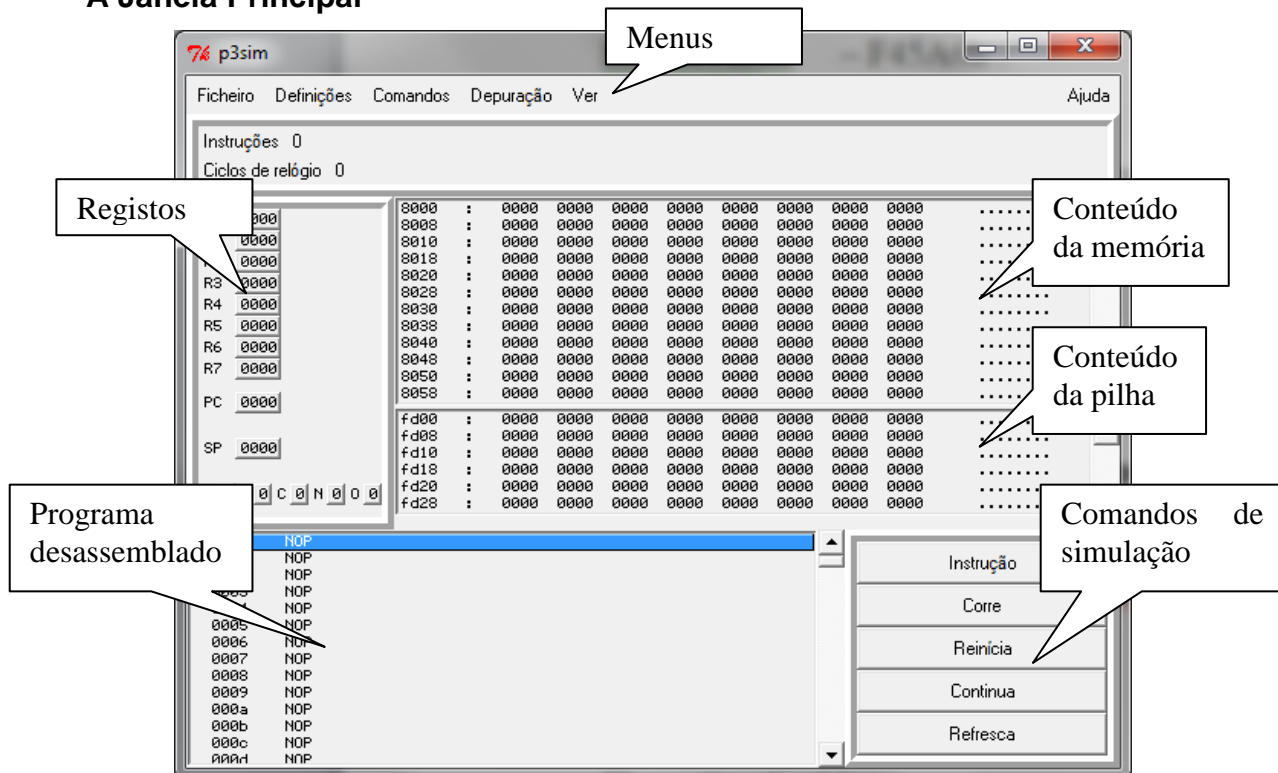
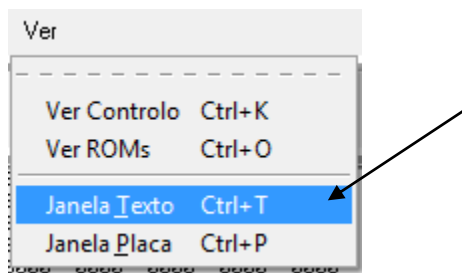


Ilustração 1 – Janela Principal do P3

A janela principal é composta por 6 zonas:

- **Menus:** Que permitem lidar com ficheiros, Definir zonas de memória, dar comandos, depurar código e ver as janelas do P3.
- **Zona com número de instruções e ciclos de relógio.**
- **Zona com valores de registos e bits de estado.**
- **Zona com o conteúdo da memória.**
- **Zona com o código desassemblado do programa.**
- **Zona com comandos de simulação.**

A Janela de Texto



Faça Ver→Janela de Texto. Deverá aparecer a seguinte janela:

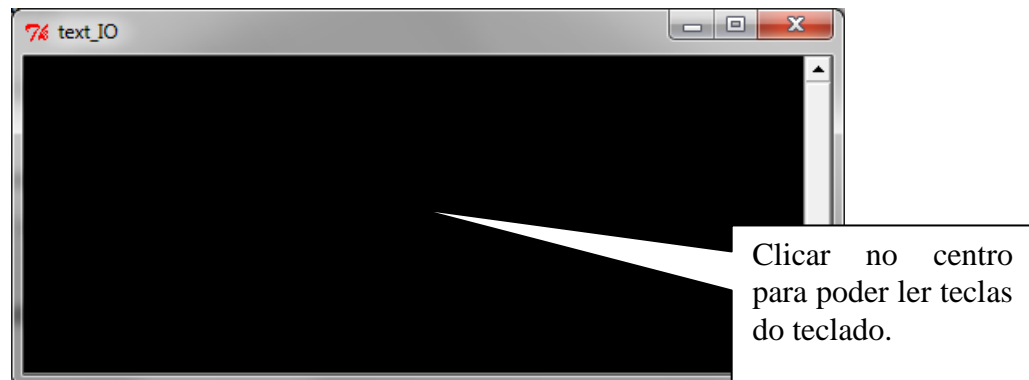
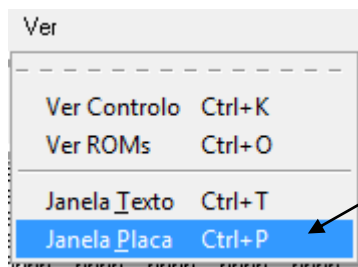


Ilustração 2 – Janela de Texto do P3

Esta é uma janela de 25 por 80 caracteres onde é possível escrever texto e ler teclas do teclado. Para poder ler teclas deve selecionar esta janela clicando no centro da janela.

Janela da Placa de IO



Fazendo Ver→Janela da Placa deverá surgir a seguinte janela.

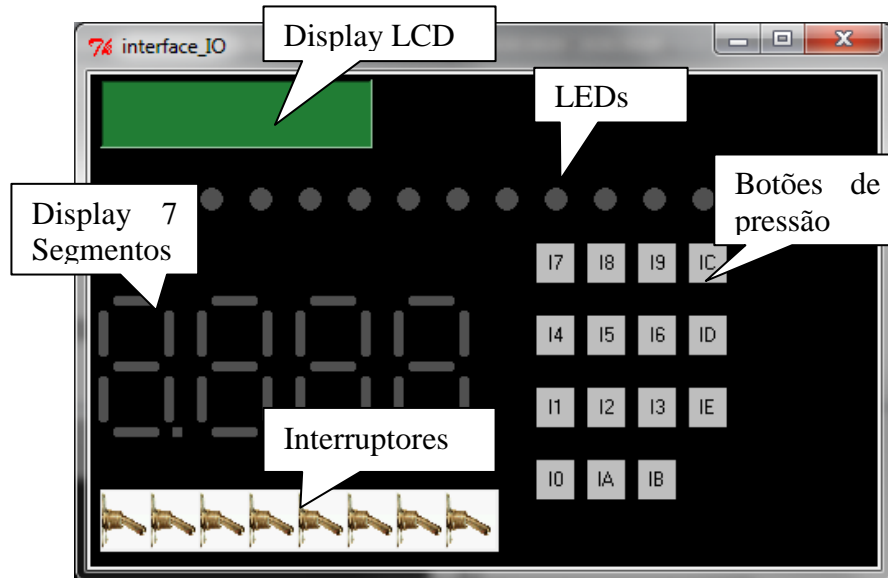


Ilustração 3 – Janela da Placa do P3

A janela é composta pelos seguintes elementos:

- Display LCD
- LEDs
- Display de 7 Segmentos
- Botões de pressão
- Interruptores

5. Utilização do Assemblador

- Compilar o ficheiro assembler usando o comando:

```
p3as.sh nome_do_ficheiro.as
```

gera dois ficheiros, nome_do_ficheiro.lis e nome_do_ficheiro.exe

- Carregar para o simulador, usando o comando:

```
p3sim.sh nome_do_ficheiro.exe
```

Ou correndo o simulador e carregando nome_do_ficheiro.exe com file→Open.

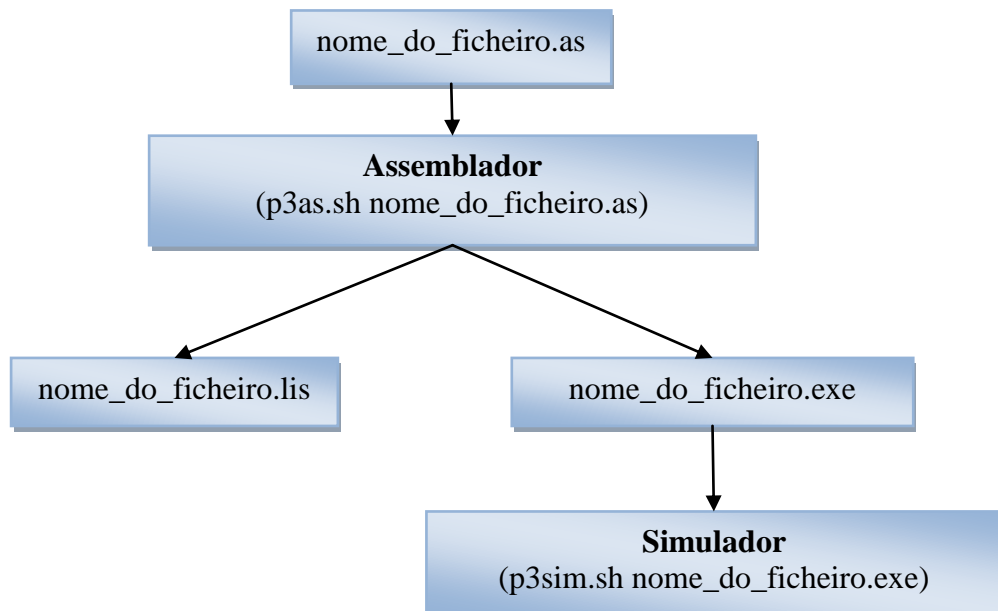
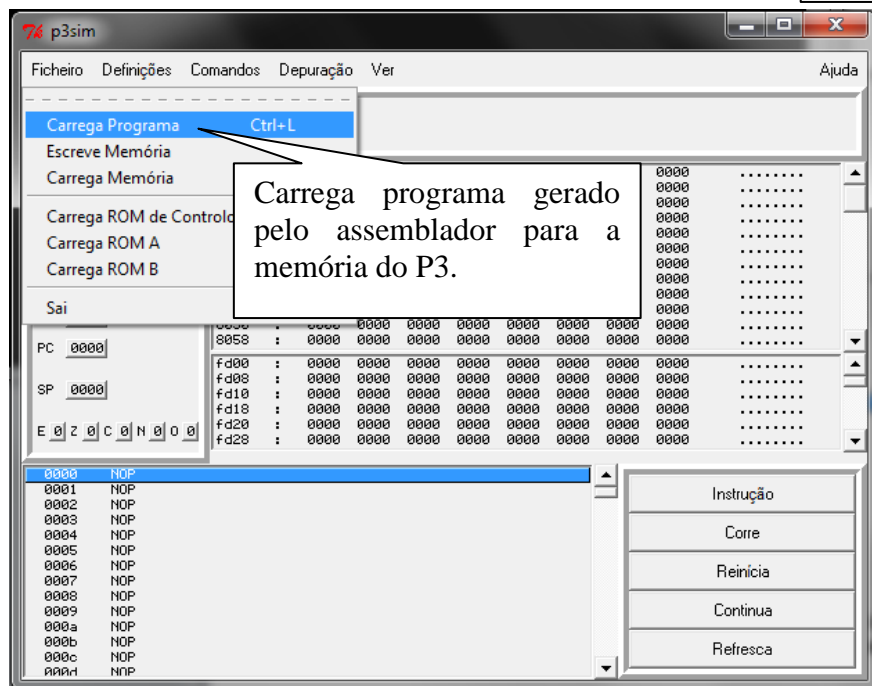
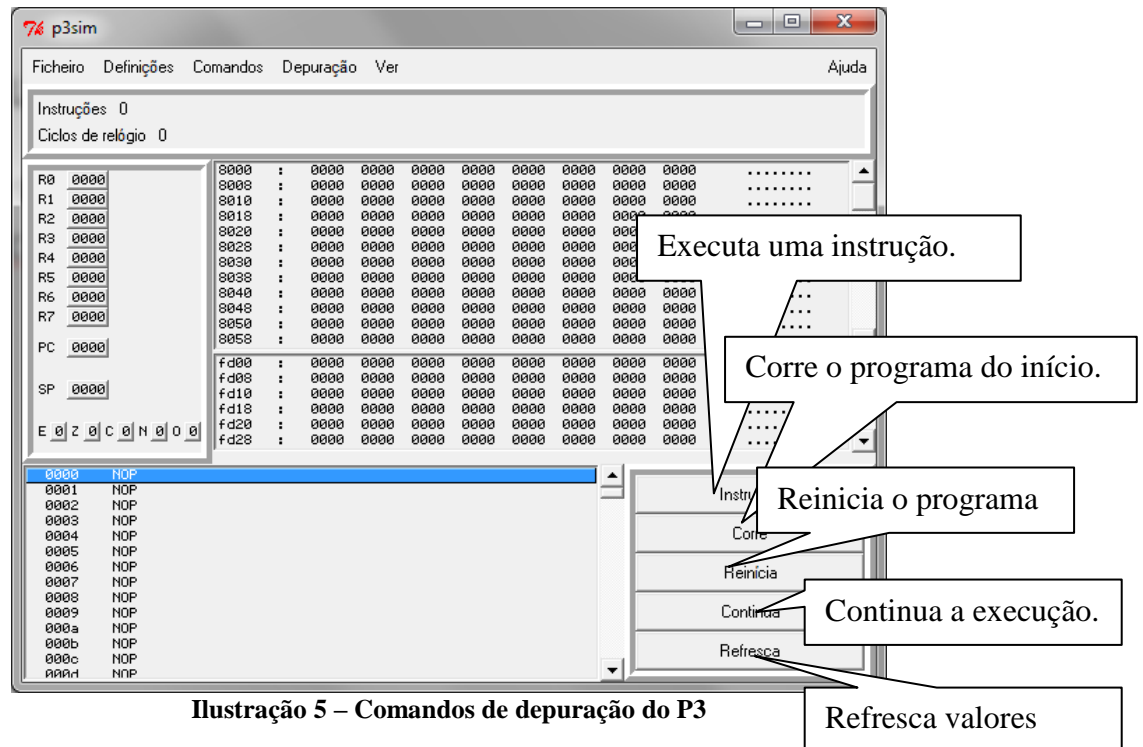


Ilustração 4 – Compilação de um ficheiro no P3

6. Utilização do Simulador

As seguintes figuras ilustram a utilização do simulador.



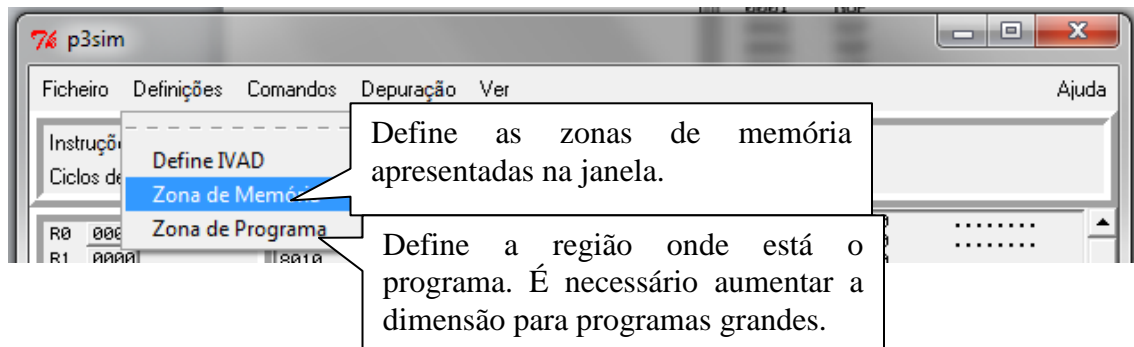


Ilustração 7 – Definições do P3

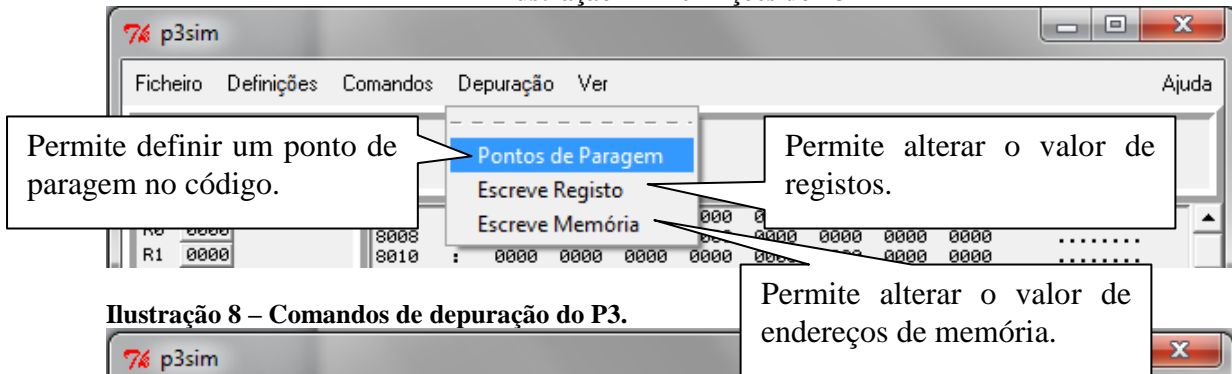


Ilustração 8 – Comandos de depuração do P3.



Ilustração 9 – Menu Ver do Simulador do P3.

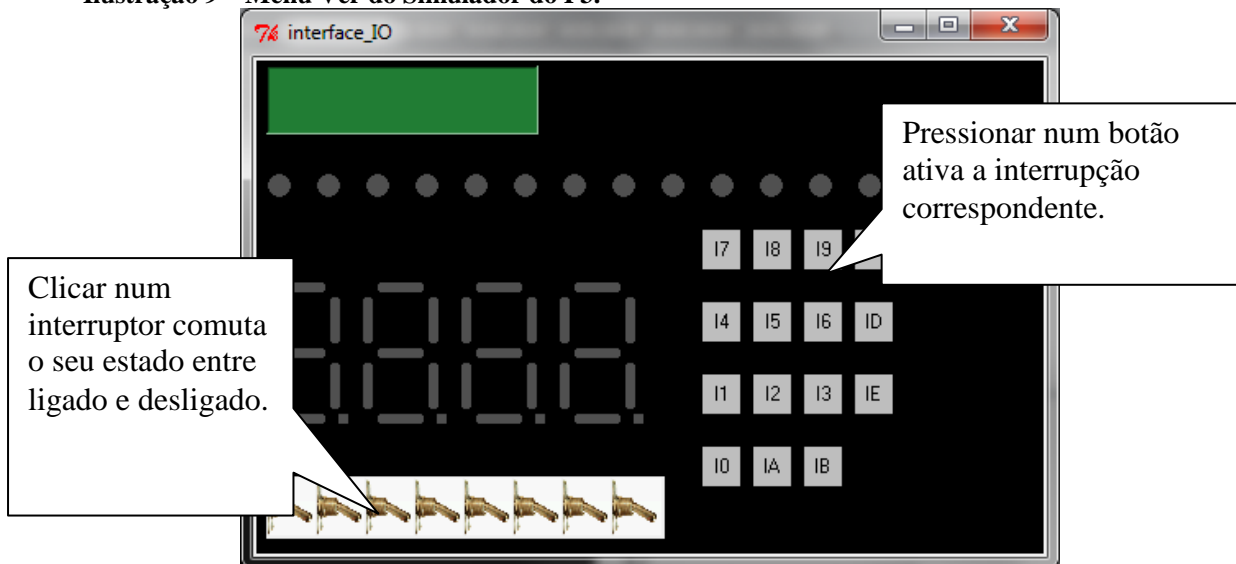


Ilustração 10 – Interação com a janela da placa.

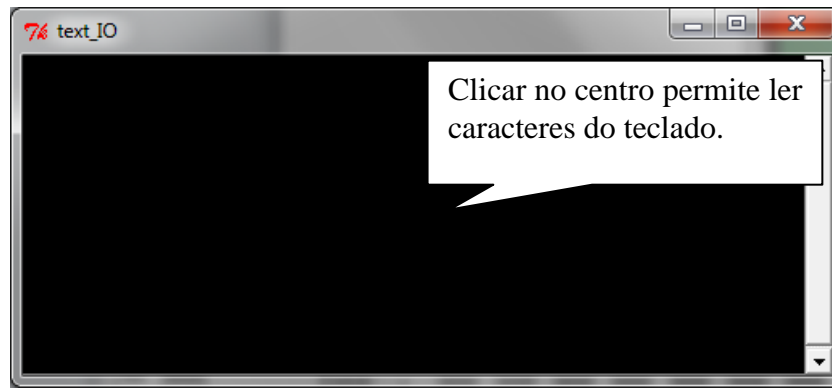
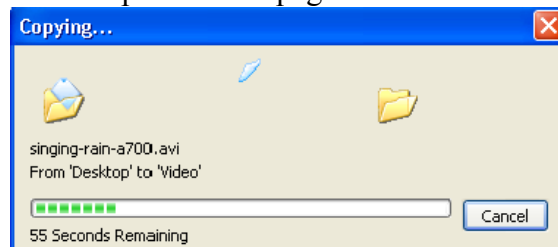


Ilustração 11 – Interação com a janela de texto.

7. Ensaio de Utilização

Começar uma sessão

Inicie a máquina virtual. Copie, para a sua máquina, os ficheiros “tutorial_1.as” e “tutorial_2.as” que estão disponíveis na página da cadeira.



Introdução ao P3

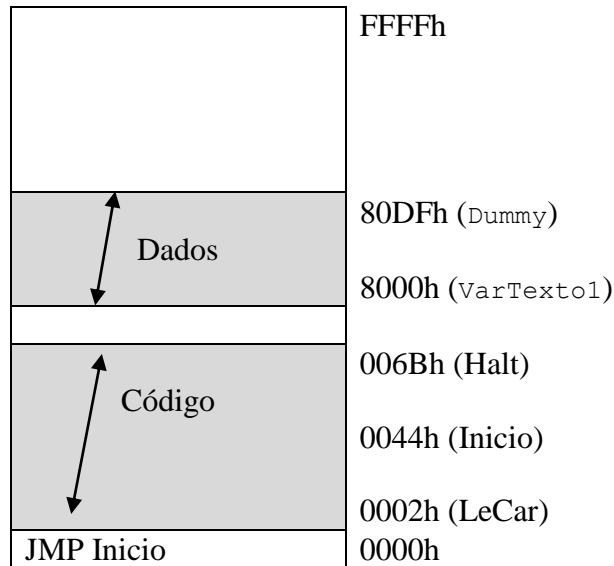
Utilizando o ficheiro “tutorial_1.as” execute as seguintes funções:

Proceda à assemblagem do programa, gerando assim o ficheiro executável (tutorial_1.exe) e o ficheiro de referências (tutorial_1.lis). Para isso proceda como se indica em seguida:

```
> p3as.sh tutorial_1.as <enter>    obtendo:
p3as, Version 1.3, last modified Mar 20 2006
Assembling completed with success, object file: tutorial_1.exe
References file: tutorial_1.lis
```

Examine o ficheiro “tutorial_1.lis”. Anote o valor dos endereços correspondentes às etiquetas VarTextol, Dummy, LeCar, Inicio e Halt. Verifique que o mapa de memória do programa é o que se encontra em baixo.

Mapa de memória:



Utilize o simulador p3sim para testar e executar o programa. Para isso, evoque o simulador da seguinte forma:

```
> p3sim.sh <enter>
```

Seguidamente, efetue o carregamento do programa executável tutorial_1.exe, através da execução da opção **Carrega Programa**, existente no menu **Ficheiro**. Verifique a alteração do conteúdo da janela de código. Analise o código desassemblado (que encontra na janela de código) e compare-o com o do programa fonte (ficheiro “tutorial_1.as”).

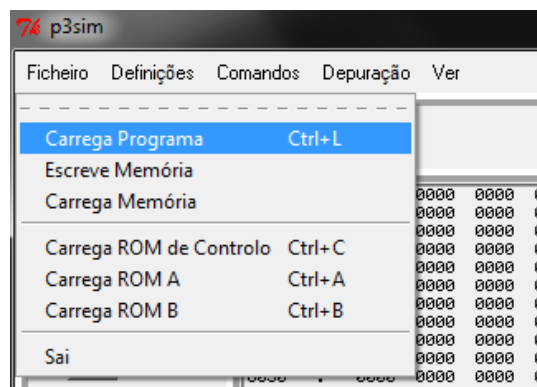


Ilustração 12 – Carrega programa.

Antes de iniciar a execução do programa, abra a janela de Entradas/Saídas. Para tal, seleccione a opção **Janela Texto**, existente no menu **Ver**. Seguidamente, execute o programa, seleccionando o botão **Corre**.

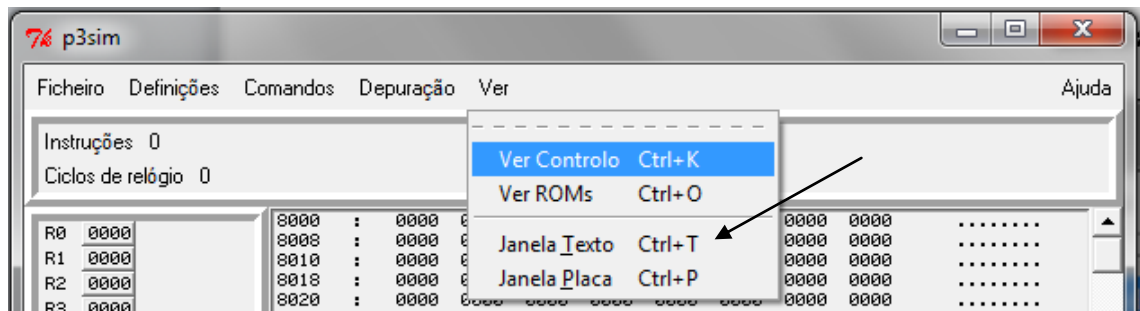


Ilustração 13 – Ver Janela de Texto

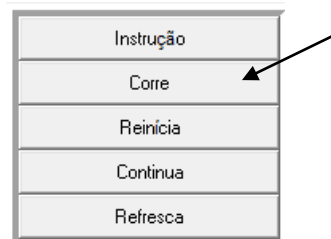


Ilustração 14 – Botão Corre.

Introdução à linguagem de programação Assembly

Compile o ficheiro tutorial_2.as e carregue-o para simulador. Deixe o ficheiro correr durante uns segundos e pressione em Parar.

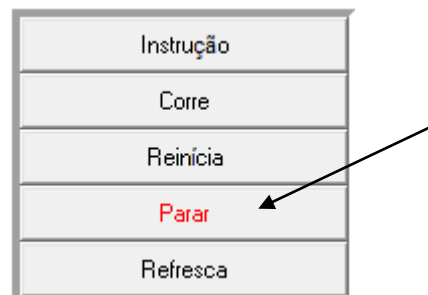


Ilustração 15 – Botão Parar.

Verifique os valores dos registos R0 a R7, nomeadamente:

R0: 0000h – R0 tem sempre o valor 0

R1: 000Fh – Foi carregado com o valor de ConstUmByte que é 0Fh.

R2: FFFFh – Foi carregado com o valor 65535 em decimal que é FFFFh em hexadecimal.

R3: FFFFh – Foi carregado com o valor de R2 e por isso tem o mesmo valor que este.

R4: 0064h – Foi carregado com o valor da posição de memória 0Fh+8001h=8010h em que temos 64h (carater 'd').

R5: 000Fh – Foi carregado com o valor da posição de memória 8001h correspondente à variável VarOutroByte que foi inicializada com o valor 0Fh.

R6: AE20h – Foi carregado com o valor da posição de memória zero, pois este é o valor inicial de SP. Na posição de memória 0000h está a primeira palavra da instrução “MOV R0, LetraA”, que é codificada com AE20h.

R7: 0000h – Foi carregado com o valor da posição de memória 000Ch (valor de PC) + 0Fh (ConstUmByte) = 0001Bh a qual se encontra na zona de programa não tendo sido inicializada e tendo por isso o valor zero.

Observe o valor que se encontra na posição 8003h da memória.

M[8003h]: 0041h - Corresponde ao valor ASCII da letra ‘A’ o primeiro carater da sequência de caracteres de 'Arquitectura de Computadores'.

Através da utilização do comando *Escreve Registo*, do menu *Depuração*, inicialize os registos R1 a R7 com o valor FFFFh. Utilizando o comando *Escreve Memória*, do mesmo menu, inicialize as posições de memória correspondentes às variáveis VarUmByte e VarOutroByte, também com o valor FFFFh.

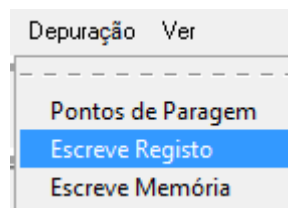


Ilustração 16 – Escreve Registo e Escreve memória.

Repita a execução do programa utilizando o comando *Reinicia* seguido de *Corre*. Compare a informação dos registos com os valores anteriores.

Deverá obter:

R0: 0000h – Igual.

R1: 000Fh – Igual.

R2: FFFFh – Igual.

R3: FFFFh – Igual.

R4: 0064h – Igual.

R5: FFFFh – É carregado com o valor da posição de memória 8001h que desta vez é inicializada com o valor FFFFh.

R6: AE20h – Igual.

R7: 0000h – Igual.

Acrescente no final do programa referido, antes da instrução Halt: BR Halt, as seguintes instruções:

a) MOV M[VarOutroByte], R4

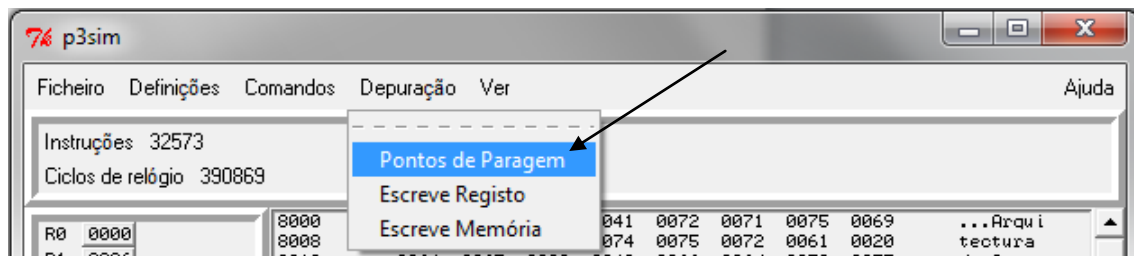
- b) MOV R1, VarUmaWord
MOV M[R1], R6
- c) MOV R1, M[VarOutroByte]
MOV M[VarUmByte], R1

Compile novamente o programa e carregue para o simulador. Coloque um ponto de paragem em a). Para colocar o ponto de paragem faça:

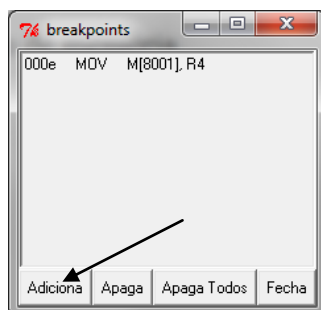
Selecione a instruções no endereço 000Eh, que corresponde á instrução a), na janela com o programa desassemblado.

0007	MOV	R4, M[R1-7fff]
0009	MOV	R5, M[8001]
000b	MOV	R6, M[SP]
000c	MOV	R7, M[PC+ff]
000e	MOV	M[8001], R4
0010	MOV	R1, 8002

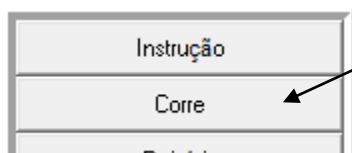
Selecione pontos Depuração→Pontos de paragem do Menu,



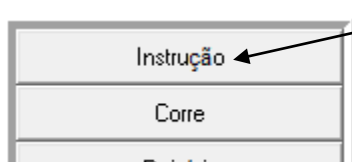
Na caixa de diálogo que surge no ecrã seleciona Adiciona.



Mande correr o programa até ao ponto de paragem.



Corra cada uma das instruções passo a passo utilizando o botão Instrução,



Verifique o efeito de cada uma das instruções:

- Utiliza o modo de endereçamento direto para copiar o conteúdo do registo R4 para a posição de memória *VarOutroByte*.
- Utiliza o modo de endereçamento indireto por registo para copiar o conteúdo do registo R6, para a posição de memória *VarUmaWord*. Começa por colocar a posição de memória no registo R1.
- Utiliza duas instruções de endereçamento direto para copiar o conteúdo da posição de memória *VarOutroByte* para a posição de memória *VarUmByte*. Utiliza o registo R1 como registo auxiliar.

8. Utilização dos periféricos do P3

Janela de texto



Ilustração 17 – Janela de Texto do P3

MOV R1, M[FFFFh] : R1 receber caracteres teclados na janela de texto

MOV M[FFFEh], R1 : escrever o carater dado por R1 na janela de texto

MOV R1, M[FFFDh] : R1 não é nulo se houve alguma tecla premida na janela de texto

MOV M[FFFCh], R1 : O valor de R1 dita a posição do cursor na janela de texto

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Linha								Coluna							

Ilustração 18 – Posicionamento do Cursor na janela de texto

O porto de controlo deve ser inicializado com uma escrita de FFFFh.

Botões de pressão

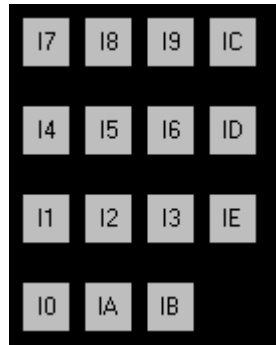


Ilustração 19 – Botões de pressão do P3

Conjunto de 15 interruptores de pressão. A ativação de cada um destes botões gera uma interrupção com o correspondente vetor de interrupção. Os endereços das rotinas de interrupção estão na tabela de vetores de interrupção no endereço M[FE00h+vector].

Interruptores



Ilustração 20 – Interruptores do P3

MOV R1, M[FFF9h] : Cada um dos 8 bits menos significativos de R1 contém o estado do interruptor correspondente.

Leds



Ilustração 21 – LEDs do P3

MOV M[FFF8h], R1 : Cada bit de R1 indica quais dos 16 LEDs estão ligados.

Display de 7 segmentos



Ilustração 22 - Display de 7 segmentos do P3

MOV M[FFF3h],R1; MOV M[FFF2h],R1; MOV M[FFF1h],R1; MOV M[FFF0h],R1;

Cada uma destas operações controla um display de 7 LEDs, com um valor de 0 a F.

Display de cristal líquido ou LCD



Ilustração 23 - Display de cristal líquido ou LCD do P3

Display de texto com 16 colunas e duas linhas.

MOV M[FFF5h], R1 : Escreve o carater em R1 no display.

MOV M[FFF4h], R1 : R1 controla diferentes funções no display.

Tabela 1 – Funções dos bit do porto de controlo do display de cristal liquido.

Bit	Acção
15	liga ou desliga o <i>display</i> LCD
5	limpa o <i>display</i> LCD
4	posiciona na linha 0 ou 1 o cursor
3 a 0	posiciona o cursor na coluna especi_cada

Máscara de interrupções

MOV M[FFFAh], R1 : Permite seleccionar individualmente quais dos 16 primeiros vetores de interrupção (de 0 a 15) estão habilitados.

Temporizador

Dispositivo que fornece a geração de uma interrupção ao fim de um intervalo de tempo real, especificado pelo utilizador.

MOV M[FFF7h], R1 : R1=1 arranca o temporizador, R1=0 para o temporizador.

MOV M[FFF6h], R1 : R1 contém o número de intervalos de 100ms ao fim dos quais o temporizador gerará uma interrupção.

9. Exemplos de Código do P3

Junto com este tutorial, estão os seguintes ficheiros de exemplo de código do P3:

- Demo1.as
Demonstração da utilização dos periféricos e interrupções no P3.
 - (1) Escrita de mensagem numa posição pre-definida da janela de texto.
 - (2) Ciclo de escrita no display de 7 segmentos
 - (3) Pausa/Recomeço da escrita por ação do interruptor da direita
 - (4) Mudança de display por ativação da interrupção IO
 - (5) Alteração do estado dos LEDs em função do número de interrup.
 - (6) Escrita no LCD por ativação da interrupção I1
- BubbleSort.as
Implementação do Algoritmo de ordenação Bubble Sort para Assembly do P3.
 - (1) Inicialização de Dados em Memória
 - (2) Ordenação de dados por ordem crescente

(3) Ordenação de dados por ordem decrescente

- display7seg
- display7segint.as
- display7segintled.as
- display7segintledlcd.as
- IOwINT.as
- janelatexto.as

Demonstração da utilização da janela de texto

10. Referências

[1] G.Arroz, J.C.Monteiro, A.Oliveira, “Manual do Simulador do P3”, IST, 2003