APÊNDICE A2: CÓDIGO DA IMPLEMENTAÇÃO DA PARTIÇÃO DE S/W DO EMULADOR

Neste apêndice encontram-se o código fonte da partição de s/w do emulador do GLiTCH, juntamente com o texto de alguns utilitários.

1 Lista de Ficheiros e Resumo do seu Conteúdo

1.1 Utilitários

runme.bat	Batch para configurar as ligações entre *transputers* na placa B008 à qual se liga o EDgAR.
heloskip.hl1	Descrição de ligações entre *transputers*, a estabelecer por s/w, na placa B008.
b008.hl1	Descrição das ligações entre *transputers*, definidas por h/w, na placa B008.
extract.prg	Exemplo dum ficheiro com microcódigo aceite pelo emulador.
run-btl.bat	*Batch* que se pode usar para por a correr no *transputer* do EdgAR a componente de s/w do emulador[1].
pe-hw.bat	*Batch* que permite gerar o ficheiro executável do emulador.
pe-sw.bat	*Batch* que permite gerar um ficheiro executável, ao qual corresponde um simulador do GLiTCH equivalente ao emulador descrito neste relatório.

1.2 Ficheiros *Header*

edgar-hw.h	*Header* com funções e variáveis associadas com a plataforma EDgAR.
data.h	*Header* que define as variáveis, funções e contantes usadas no emulador. Só é utilizado em **pe.c** .
datadef.h	*Header* equivalente a **data.h**, com a excepção de apenas declarar as variáveis. É utilizado em todos os ficheiros "*.c" excepto **pe.c** .

[1] No resto da introdução do apêndice A2, passa a chamar-se "emulador" à "componente de s/w do emulador".

1.3 Ficheiros com Código Fonte do Emulador

arithrom.c	Microinstruções que executam as operações aritméticas.
chip.c	Funções usadas nas microinstruções que operam com as TAGs (fisrt, alt, ...).
fetch.c	Função que lê o microcódigo do ficheiro "*.prg".
lca-util.c	Funções associadas com a utilização de FGAs do EDgAR.
logicrom.c	Microinstruções que executam as operações lógicas.
magnirom.c	Código fonte das microinstruções que executam as operações de comparação de magnitude.
matchrom.c	Microinstruções que executam as operações de *match*.
pbl.c	Operações do PBL (excepto ***write***).
pblwrit2.c	Parte da operação ***write*** do PBL.
pblwrite.c	Parte da operação ***write*** do PBL.
pe.c	Função **main()** e uma função de inicialização.
pe_rom.c	Microinstruções que executam as operações: de deslocamento de TAGs, de manipulação de TAGs de vários *chips*, de leitura da CAM, de escrita na CAM e variadas.
sgnbtrom.c	Microinstruções que executam as operações de cálculo do bit de sinal.
teste.c	Ficheiro que só é utilizado se for definido o símbolo DEBUG em data.h. Contém funções para efectuar a depuração do emulador e de interface com o utilizador, que basicamente mostram o conteúdo das unidades com memória implementadas em s/w (CAM, PBLR, TAGs e registos CARRY e MS).
vsr-sw.c	Microinstruções que executam as operações sobre o VSR (no caso do VSR ser simulado).
vsr.c	Microinstruções que executam as operações sobre o VSR (no caso do VSR ser emulado).

2 Lista das Funções Incluídas nos Ficheiros

Junta-se a linha de ocorrência nos ficheiros fonte e a declaração de cada função.

arithrom.c Pág. 39

12: void arith_this (void)
163: void arith_above (void)
194: void arith_below (void)
224: void arith_image (void)
252: void swap_this (void)
296: void swap_above (void)
333: void swap_below (void)
370: void swap_image (void)
405: void arithholdms (void)
416: void swap_holdms (void)

chip.c Pág.47

18: void TagAbove(void)
53: void TagBelow(void)
87: void Chip_ALT_IN(void)
162: void Chip_ENABLE(void)
180: void Chip_FIRST_IN(void)
254: void Chip_ALT_LATCH (int X)
276: void Chip_ENABLE_LATCH (int X)
301: void Chip_FIRST_LATCH (int X)

fetch.c Pág.53

35: void fetch (WORD *inst)

lca-util.c Pág.55

28: void ClearFfsResetReprogram (void)
61: void programarLca(unsigned long *addLca, char *file)

logicrom.c Pág.57

11: void logic_this (void)
82: void logic_above (void)
104: void logic_below (void)
126: void logic_image (void)
149: void xor_this (void)
206: void xor_above (void)
227: void xor_below (void)
248: void xor_image (void)

magnirom.c Pág.62

11: void cpdata_this (void)
70: void cpdata_abov (void)
88: void cpdata_belo (void)
106: void cpdata_imag (void)
125: void cpsign_this (void)
184: void cpsign_abov (void)
203: void cpsign_belo (void)
222: void cpsign_imag (void)

matchrom.c Pág.67

11: void match (void)
37: void tag_match (void)
56: void cy_match (void)
73: void and_match (void)

pbl.c Pág.69

31: void pbl_nop (void)
39: void pbl_match (void)
118: void pbl_read2 (void)
153: void pbl_read3 (void)
177: void pbl_mergeR (void)
189: void pbl_mergeL (void)
201: void pbl (n)

pblwrit2.c Pág.73

32: int parte1_pbl_write (void)

pblwrite.c Pág.83

31: void pbl_write (void)
42: void parte2_pbl_write (void)

pe.c Pág.98

28: int main(int argc, char *argv[])
169: void inicializacoes (WORD *ninst, char *file)
216: void erro (char *s)

pe_rom.c Pág.102

44: void nop (void)
64: void shift_up (void)
121: void shift_down (void)
166: void init_alt_ch (void)
175: void init_alt_b1 (void)
184: void init_alt_b0 (void)
197: void alt(void)
202: void tag_alt(void)
207: void carry_alt(void)
218: void util_tags(int instrucao)
301: void first(void)
403: void init_first0 (void)
414: void init_first1 (void)
426: void clear_first (void)
450: void read_this (void)
550: void read_above (void)
557: void read_below (void)
566: void write_this (void)
573: void write_above (void)
581: void write_below (void)
591: void tag_to_cy (void)
605: void tag_to_subs (void)
612: void tag_subset (void)

623: void cy_subset (void)
642: void clear_n_sub (void)
656: void refresh (void)
663: void extract_tag (void)
699: void restore_tag (void)
762: void init_rom (void)

sgnbtrom.c Pág.117

11: void overf_this (void)
86: void overf_above (void)
113: void overf_below (void)
140: void overf_image (void)
165: void overfholdms (void)

teste.c Pág.120

20: int MostraCam()
210: void GravaChip()
371: void MostraPblr ()
387: int MenuDebug()
431: int teste(void)
285: void LoadChip(fpstr)

vsr-sw.c Pág.128

42: void TagImage(void)
65: void write_image(void)
91: void rotate_imag(void)
118: void extract_imag(void)
151: void shiftVsr(void)
197: void inicializarVsr(void)

vsr.c Pág.132

25: void TagImage(void)
62: void write_image (void)
104: void rotate_imag(void)
144: void shiftVsr(void)
180: void extract_imag(void)
211: void inicializarVsr(void)

3 Conteúdo dos Ficheiros