



ISEL

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA



PDS16 Quick Reference



SDP16 User Manual

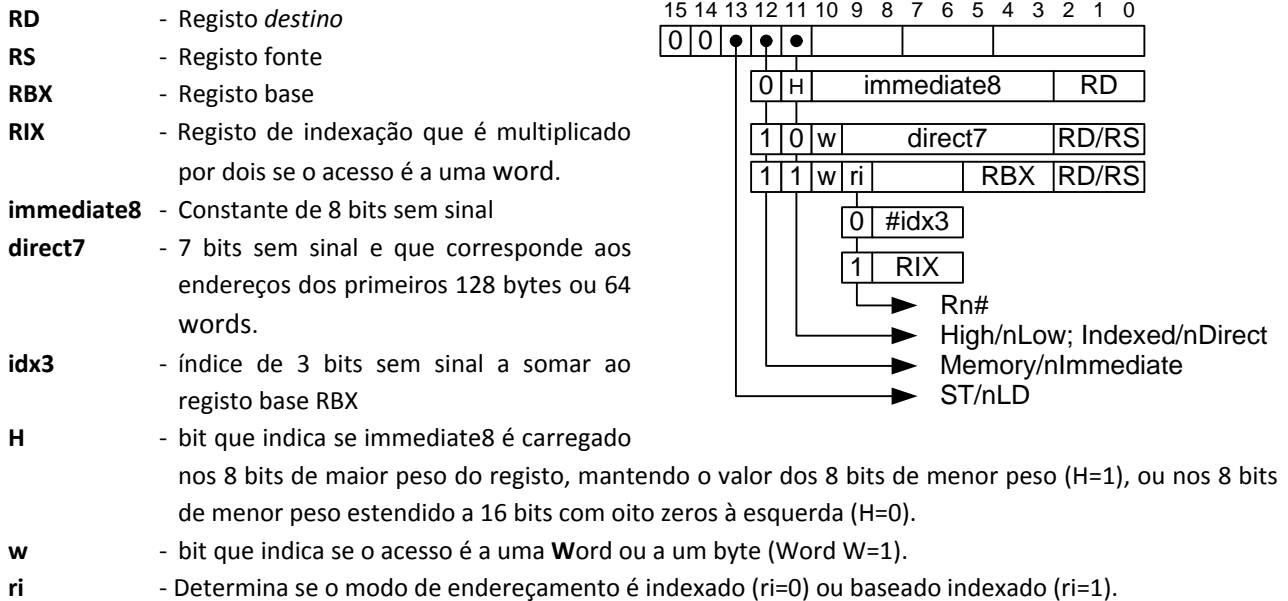
DEPARTAMENTO DE
ENGENHARIA DE
ELECTRÓNICA E
TELECOMUNICAÇÕES E DE
COMPUTADORES

Secção de
Arquitecturas e
Sistemas Operativos

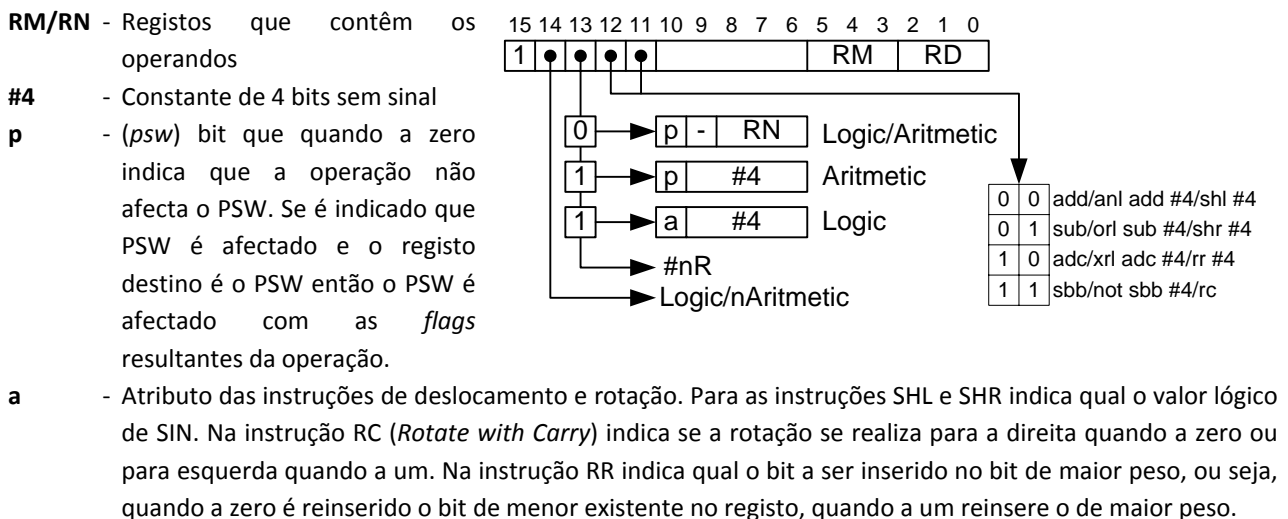
V2.0

1 PDS16

Memory Data Transfer



Data Processing



Flow Control

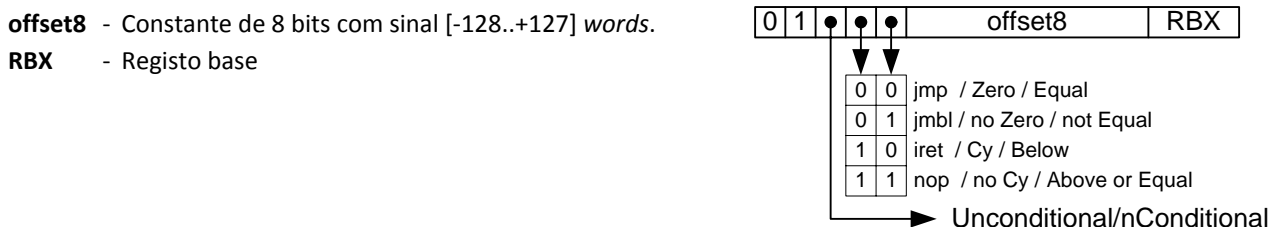
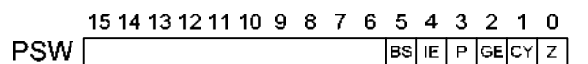


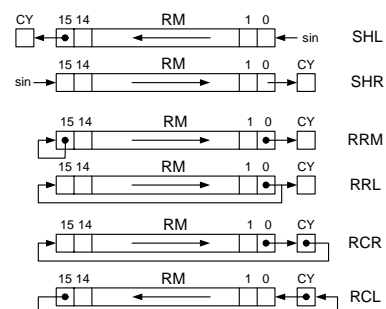
Tabela do código e sintaxe das Instruções

15	12	11	10	9	8	7	6	5	4	3	2	1	0	Mnemónica	Operandos	Função
														LOAD		
0x0	0	i7								i0	rd2		rd0	ldi	rd,#immediate8	rd= 0x00 immediate8
0x0	1	i7								i0	rd2		rd0	ldih	rd,#immediate8	rd=0ximmediate8,LSB(rd)
0x1	0	w	d6							d0	rd2		rd0	ld/b	rd,direct7	rd=[direct7]
0x1	1	w	0	ix2		ix0	rb2			rb0	rd2		rd0	ld/b	rd,[rbx,#idx3]	rd=[rbx+idx3]
0x1	1	w	1	ri2		ri0	rb2			rb0	rd2		rd0	ld/b	rd,[rbx,rix]	rd=[rbx+rix]
														STORE		
0x3	0	w	d6							d0	rs2		rs0	st/b	rs,direct7	[direct7]=rs
0x3	1	w	0	ix2		ix0	rb2			rb0	rs2		rs0	st/b	rs,[rbx,#idx3]	[rbx+idx3]=rs
0x3	1	w	1	ri2		ri0	rb2			rb0	rs2		rs0	st/b	rs,[rbx,rix]	[rbx+rix]=rs
														ARITMÉTICA		
0x8	0	p	-	rn2		rn0	rm2			rm0	rd2		rd0	add/f	rd,rm,rn	rd=rm+rn
0x9	0	p	-	rn2		rn0	rm2			rm0	rd2		rd0	adc/f	rd,rm,rn	rd=rm+rn+cy
0x8	1	p	-	rn2		rn0	rm2			rm0	rd2		rd0	sub/f	rd,rm,rn	rd=rm-rn
0x9	1	p	-	rn2		rn0	rm2			rm0	rd2		rd0	sbb/f	rd,rm,rn	rd=rm-rn-cy
0xa	0	p	c3			c0	rm2			rm0	rd2		rd0	add/f	rd,rm,#const4	rd=rm+const4
0xb	0	p	c3			c0	rm2			rm0	rd2		rd0	adc/f	rd,rm,#const4	rd=rm+const4+cy
0xa	1	p	c3			c0	rm2			rm0	rd2		rd0	sub/f	rd,rm,#const4	rd=rm-const4
0xb	1	p	c3			c0	rm2			rm0	rd2		rd0	sbb/f	rd,rm,#const4	rd=rm-const4-cy
														LÓGICA		
0xc	0	p	-	rn2		rn0	rm2			rm0	rd2		rd0	and/f	rd,rm,rn	rd=rm & rn
0xc	1	p	-	rn2		rn0	rm2			rm0	rd2		rd0	orl/f	rd,rm,rn	rd=rm rn
0xd	0	p	-	rn2		rn0	rm2			rm0	rd2		rd0	xrl/f	rd,rm,rn	rd=rm ^ rn
0xd	1	p	-	-	-	-	rm2			rm0	rd2		rd0	not/f	rd,rm	rd=~rs
0xe	0	sin	c3			c0	rm2			rm0	rd2		rd0	shl	rd,rm,#const4, sin	rd=(rm,sin) << const4
0xe	1	sin	c3			c0	rm2			rm0	rd2		rd0	shr	rd,rm,#const4, sin	rd=(rm,sin) >> const4
0xf	0	MnL	c3			c0	rm2			rm0	rd2		rd0	rr(m/l)	rd,rm,#const4	rd=(rm,m/l) >> const4
0xf	1	LnR	-	-	-	-	rm2			rm0	rd2		rd0	rc(l/r)	rd,rm	rd=(rm,cy,l/r)
														JUMP		
0x4	0	of7								of0	rb2		rb0	Jz/je	rbx,#offset8	if (Z) PC=rbx+(offset8<<1)
0x4	1	of7								of0	rb2		rb0	Jnz/jne	rbx,#offset8	if (!Z) PC=rbx+(offset8<<1)
0x5	0	of7								of0	rb2		rb0	Jc/jbl	rbx,#offset8	if (CY) PC=rbx+(offset8<<1)
0x5	1	of7								of0	rb2		rb0	Jnc/jae	rbx,#offset8	if (!CY) PC=rbx+(offset8<<1)
0x6	0	of7								of0	rb2		rb0	jmp	rbx,#offset8	PC=rbx+(offset8<<1)
0x6	1	of7								of0	rb2		rb0	jmpl	rbx,#offset8	r5=PC;PC=rbx+(offset8<<1)
0x7	0	-	-	-	-	-	-	-	-	-	-	-	-	iret		PSW=r0i; PC=r5i
0x7	1	-	-	-	-	-	-	-	-	-	-	-	-	nop		

- f** - (flags) colocado à direita da mnemónica indica que o registo PSW não é actualizado.
- sin** - (serial in) valor lógico do bit a ser inserido à esquerda ou à direita.
- MnL** - (Must/Least) indica qual o bit a ser inserido no bit de maior peso do registo destino. Quando a zero indica que é reinserido o bit de menor peso, quando a um é reinserido o bit de maior peso.
- LnR** - (Left/Rigth) bit que indica se a rotação se realiza para a direita (zero) ou para a esquerda (um).



Z – Zero
GE – Greater or Equal
IE – Interrupt Enable
CY – Carry/Borrow
P – Parity odd
BS – Bank Select



2 Assembler didáctico dasm

O programa `dasm.exe` é um assembler didáctico uni modular para o processador PDS16. O `dasm.exe` tem que ser evocado numa janela de DOS. O nome do ficheiro fonte a ser compilado não pode conter espaços e a extensão deverá ser `.asm` para melhor ser identificado. A evocação do assembler será concretizada na linha de comando sob a seguinte forma:

```
C:>dasm file.asm
```

O `dasm.exe` gera dois ficheiros com o mesmo nome que o ficheiro fonte com as extensões (`.lst` e `.hex`).

O `file.lst` é um ficheiro de texto destinado a ser impresso por conter o texto original adicionado do código de cada instrução e respectivo endereço, entre outra informação. Os erros de compilação são assinalados neste ficheiro, na linha correspondente. Para cada erro é indicado o tipo e a possível causa.

O `file.hex` é um ficheiro de texto com formato Intel HEX80 destinado a ser transmitido em série para um sistema alvo.

Para escrita do programa em linguagem *assembly* PDS16 deverá utilizar um qualquer editor de texto (por exemplo notepad ou notepad++).

Directivas

O ficheiro em linguagem *assembly* pode conter as seguintes directivas:

Directiva	Formato	Descrição
.ASCII	[label:] .ASCII "string ascii"	Define uma string ascii não terminada por zero.
.ASCIIZ	[label:] .ASCII "string ascii"	Define uma string ascii terminada por zero.
.BYTE	[label:] .BYTE expression_list	Gera uma lista de valores tipo byte (8 bits) separados por ,.
.BSS	.BSS	Secção predefinida para dados não iniciados.
.DATA	.DATA	Secção predefinida para dados iniciados.
.END	.END	Determina o fim do módulo assembler
.EQU	.EQU symbol_name, expression	Define o valor a atribuir a um símbolo de forma permanente.
.ORG	.ORG expression	Estabelece o valor inicial para o contador de endereços da secção corrente.
.SECTION	.SECTION section_name	Define uma secção de dados ou código.
.SET	.SET symbol_name, expression	Define o valor a atribuir a um símbolo de forma temporária.
.SPACE	[label:] .SPACE expression [,init_val]	Reserva espaço em unidades de byte. O espaço reservado pode ser iniciado com o valor init_val.
.TEXT	.TEXT	Secção de código predefinida.
.WORD	[label:] .WORD expression_list	Gera uma lista de valores tipo word (16 bits) separados por ,.
;	; line_comment	Comentário de uma linha
/* */	/* text_comment */	Comentário de várias linhas

Para facilitar a escrita de programas o assembler põe disponíveis as seguintes pseudo instruções:

```
mov/f  rd,rs    ;orl rd,rs,rs
inc/f   rd       ;add rd,rd,#1
dec/f   rd       ;sub rd,rd,#1
ret                     ;orl r7,r5,r5
```

Ficheiro fonte exemplo:

```
/* variaveis inicializadas */
.section directData
.org 10

var4x12:
.byte low(40*3+14/2) , high(255 + 0x101), 12, 13, 14, 32, 98, 255 ; decimal
.byte 014 ;octal
.byte 0xC ;hexadecimal
.byte -12 ; negativos -> -número
.byte 1100b ;binario

.section indirectData
varChar:
.byte 'A', 'B', 'C', 0x0a, 0x0d, 0 /* vários bytes com os códigos ascii das letras */
varWord:
.word 0xA25C ; dois bytes
varWord:
.word main, 11, rotina ; inicia array com os endereços das várias referências
varArray:
.space 5, 0xAB ;preenche 5 bytes com o valor 0xAB
.space 3 ;reserva 3 bytes e inicia com zero
varTexto:
.ascii "isto é texto que não acaba com um byte a zero"
.ascii "isto é", " texto que" , " não acaba com um byte a zero"

/* código */
.section start
.org 0
ld r7,[r7,#1] ; jmp main
jmp isr
.word main

/* utilizam-se as mesmas primitivas que em .data ou as seguintes */
.equ const1, 86
.set const2, 10
.word 0xA25C /* dois bytes */
.space 3, 0xAB

.section main
.org 0x2000
main: ldi R0,#123
ld r1,[r7,#1]
jmp l1
.word main
l1: ldi R2,#low(varArray)
ldih R3,#high(varArray)
jmp1 rotina
jmp $

rotina: st R0,[r2,#0]
ld r0,[r1,#4]
ld r3,var4x12+4
movf r0,r2
inc r2
ret

/* fim do modulo */
.end
```

3 Descrição do SDP16

O sistema didáctico SDP16, cujo diagrama de blocos é apresentado na figura 1, é constituído pelos seguintes elementos:

- CPU de 16 bits (P16)
- memória de programa e dados;
- portos paralelos de entrada e saída;
- módulo *Single Cycle*.
- mostrador dos sinais dos barramentos de controlo, de endereços e de dados;
- DMA para acesso à memória e aos portos;
- canal USB/JTAG;
- filtro e regulador da tensão de alimentação.

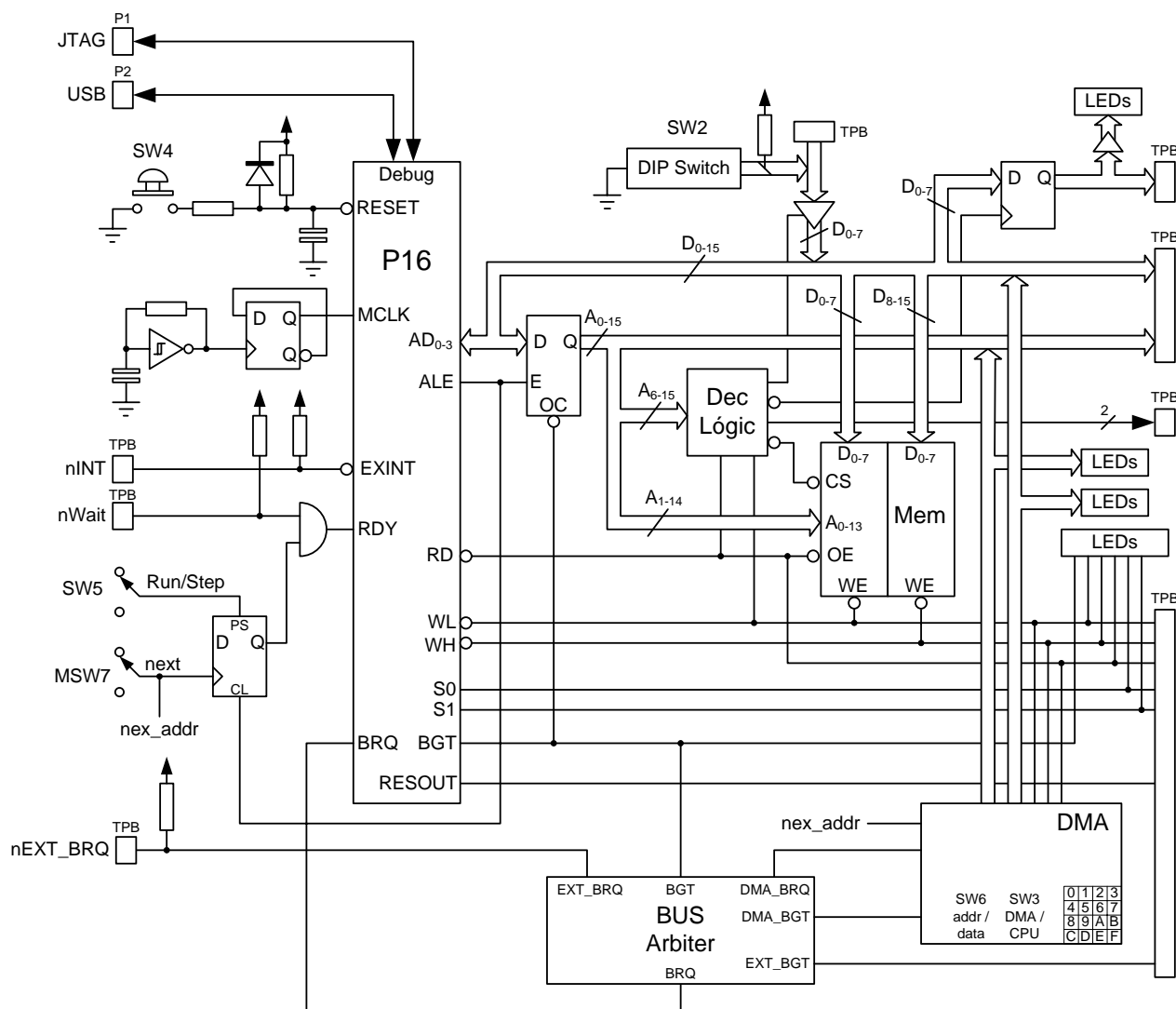


Figura 1

3.1 P16

O processador PDS16 é interligado ao SDP16 através de duas fichas, uma de 34 pinos contendo todos os sinais de dados e controlo do CPU e outra de 10 pinos para ligação à máquina de *debug* através de um canal série assíncrono ou em alternativa um canal JTAG. Na ficha de 34 pinos estão disponíveis os seguintes sinais:

Oscilador	- 500Hz com <i>duty cycle</i> de 50%, determinado por malha de condensador e resistência;
RESETIN	- Uma malha não linear (resistências, condensador e diódo) assegura o reset automático, no estabelecimento e na falha/restabelecimento da alimentação (power on e power fail), permitindo também que a acção de reset seja provocada através de um botão de pressão;
Address/Data	- Bus de 16 bits para endereços multiplexado com dados, validados pelos seguintes sinais: <ul style="list-style-type: none">• ALE (Address Latch Enable) valida endereços;• nRD (Read) valida dados de leitura de memória;• nWRL,nWRH (Write Low e High) valida dados para escrita na memória. Permite acesso a byte.
BRQ	- (Bus ReQuest) pedido para que o CPU liberte o bus;
BGT	- (Bus Grant) para indicar a quem requereu o bus a disponibilização deste;
RDY	- (Ready) quando desactivo leva a que o CPU entre em estado de espera;
EXTINT	- Entrada externa de interrupção.

3.2 Memória

Mapa de memória do sistema 64Kx8 entre 0x0000 e 0xFFFF.

Memória instalada 32K*8, de 0x0000 a 0x7FFF, constituída por dois dispositivos RAM 32k*8 (U9 e U10) que ocupam os primeiros 32K endereços do mapa de memória.

A descodificação é implementada pela PAL 22v10 (U27) com a programação em WinCupl descrita na Figura 2 e que disponibiliza para além do CS das RAMs (U9 e U10), mais três espaços de 64 bytes, criando o mapeamento presente na Figura 2. Os sinais nCS_EXT0 e nCS_EXT1 estão disponíveis no TPB (*Tie Point Block*) B13, no sentido de disponibilizar CS, aquando da implementação de trabalhos de laboratório que necessitem inserir dispositivos no mapa de memória do CPU.

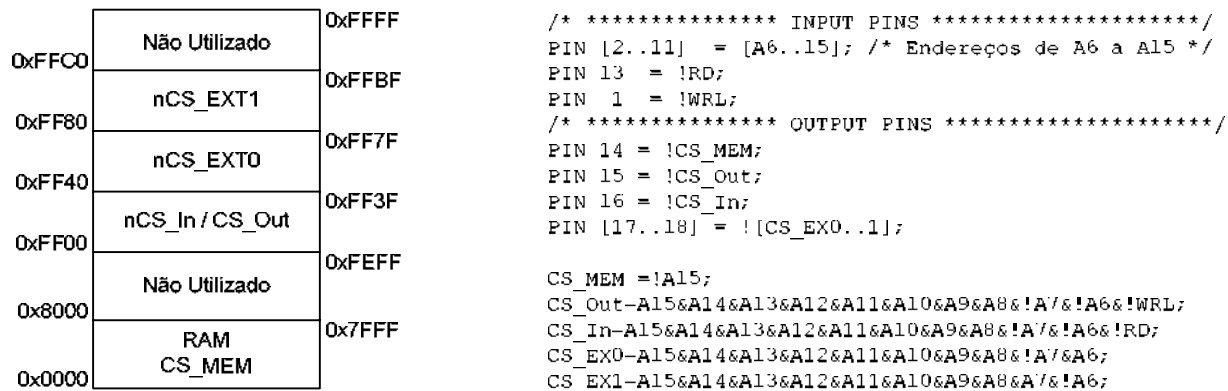


Figura 2 – Mapa de memória e descrição em CUPL do gerador de CS (PAL U12).

3.3 Porto Paralelo de entrada e de saída

O sistema inclui dois portos paralelos de oito bits – um de entrada, ao qual está associado um DIP de 8 interruptores e outro de saída, ligado a oito LEDs conforme a Figura 3 e a Figura 4. Deste modo é possível realizar alguns trabalhos de laboratório envolvendo I/O, sem necessidade de montagem externa. Os buffers U6A e U6B asseguram que os LEDs não carregam o porto de saída.

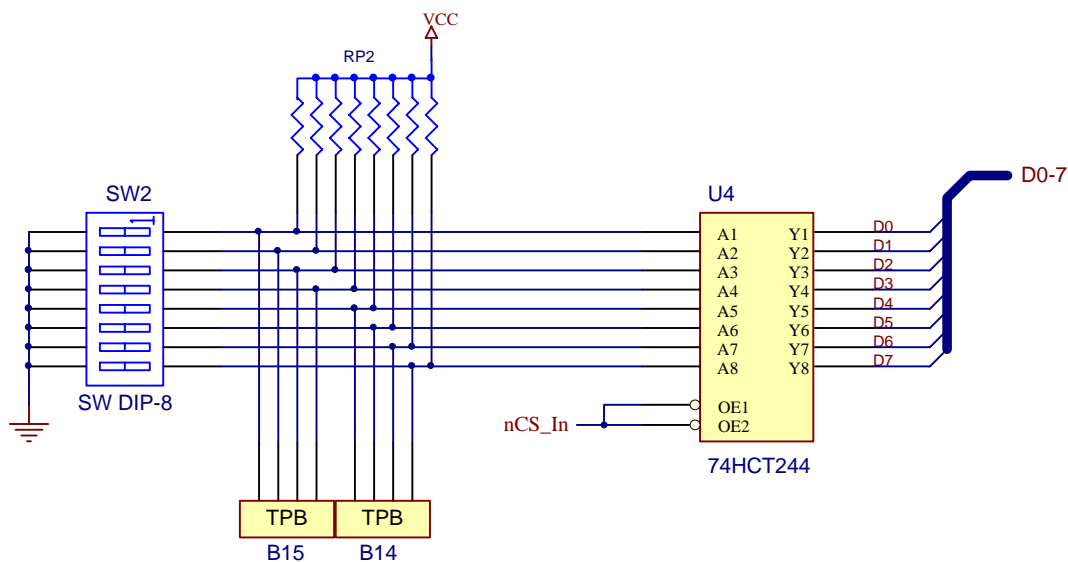


Figura 3 - Input Port

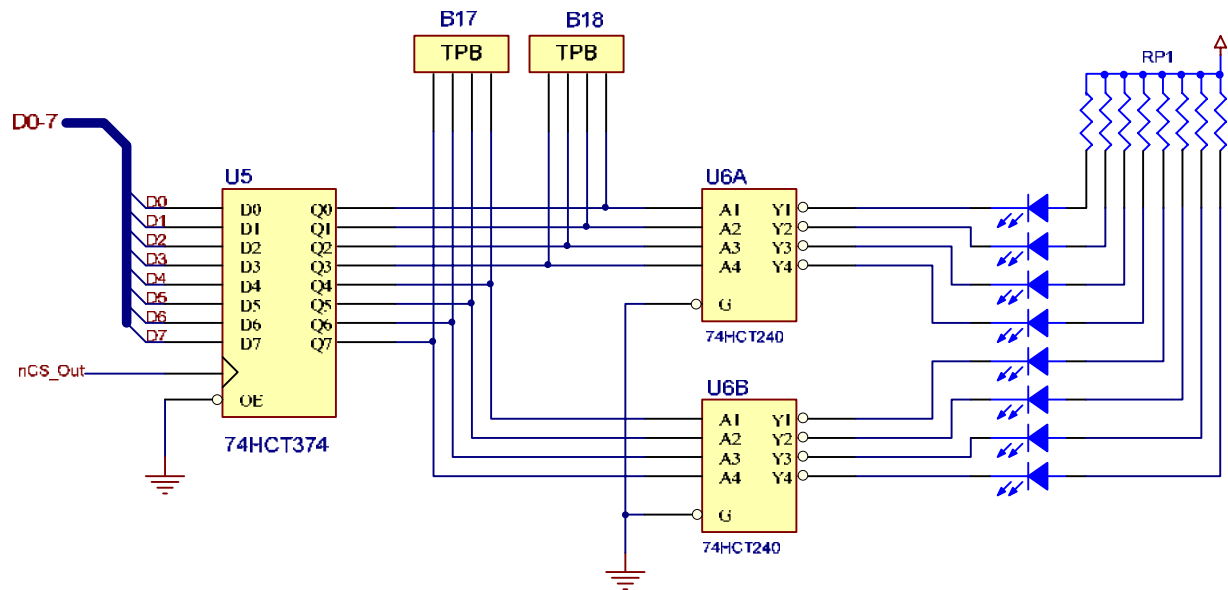


Figura 4 - Output Port

3.4 Módulo *Single Cycle*

O SDP16 dispõe de um módulo cujo esquema é mostrado na Figura 5 que, tirando partido do sinal RDY do CPU, permite observar numa fila de LEDs associados aos barramentos de dados, endereço e controlo, a informação neles presente, à medida que o CPU, na persecução das instruções de um programa, vai acedendo à memória para *fetch* de uma instrução ou execução de LOAD/STORE.

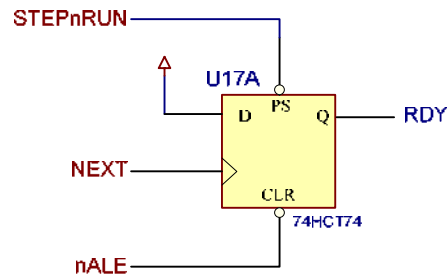


Figura 5

Esta funcionalidade permite o estudo dos ciclos do CPU e é particularmente útil no *debugging* do hardware envolvido em trabalhos de laboratório, permitindo, com o auxílio de pontas-de-prova, detectar erros de montagem ou de projecto.

3.5 Mostrador de sinais

Os sinais dos barramentos de controlo, de endereços e de dados são mostrados, através de *line drivers*, em trinta e oito LEDs, 16 de dados, 16 de endereço e 6 de controlo. O SDP16 coloca disponível em TPB (B6 a B13) todos os sinais do P16 e uma série de outros sinais para darem suporte a exercícios de laboratório.

3.6 DMA

O DMA é constituído por um teclado hexadecimal organizado em matriz espacial, um registo MBR de dezasseis bits (4 nibbles), um registo MAR de dezasseis bits (4 nibbles) e o comutador DATAnADDR (SW6) que determina se quando uma tecla é premida o respectivo código deve ser escrito no MAR ou no MBR. Nestes registos a escrita é precedida de uma acção de deslocamento do código anterior (*shift left*) a quatro bits, para que o código da última tecla premida ocupe sempre o nibble de menor peso.

O MBR é responsável por manter estável a informação no Data Bus durante a acção de escrita que é promovida pelo controlo do DMA.

Actualiza-se no MAR o endereço da posição de memória a ser acedida para escrita ou leitura. Endereços sucessivos podem ser estabelecidos por acção do comutador SW7 designado por NEXT, pois o MAR, além da acção de deslocamento já descrito, suporta acção de incremento do conteúdo nele contido.

Além dos elementos anteriormente referidos, o módulo DMA inclui um módulo denominado *BUS Arbiter* que arbitra o acesso ao bus do CPU por parte do DMA ou por um dispositivo externo. Na Figura 6 está representado o esquema bloco do módulo DMA que inclui o DMA propriamente dito e o *BUS Arbiter*.

O *BUS Arbiter*, no caso de pedidos simultâneos de acesso ao bus do CPU, dá prioridade ao DMA. Para gestão do bus, o *BUS Arbiter* dispõe dos seguintes sinais de entrada e de saída:

DMA_BRQ – Sinal de entrada proveniente do DMA e que é activado por este quando, através do comutador (SW3) DMA/CPU, o utilizador requer o acesso para consulta ou escrita na memória;

EXT_BRQ – Pedido estabelecido através do TPB (B12) por um eventual dispositivo externo;

BGT – Sinal de entrada que é activado pelo CPU para informar o BA que o bus já se encontra liberto;

DMA_BGT – Sinal de saída para informar o DMA que já pode utilizar o bus do CPU.

EXT_BGT – Sinal de saída para dar BGT ao pedido externo.

BRQ – Sinal utilizado pelo BA para informar o CPU que um dispositivo pretende aceder ao bus.

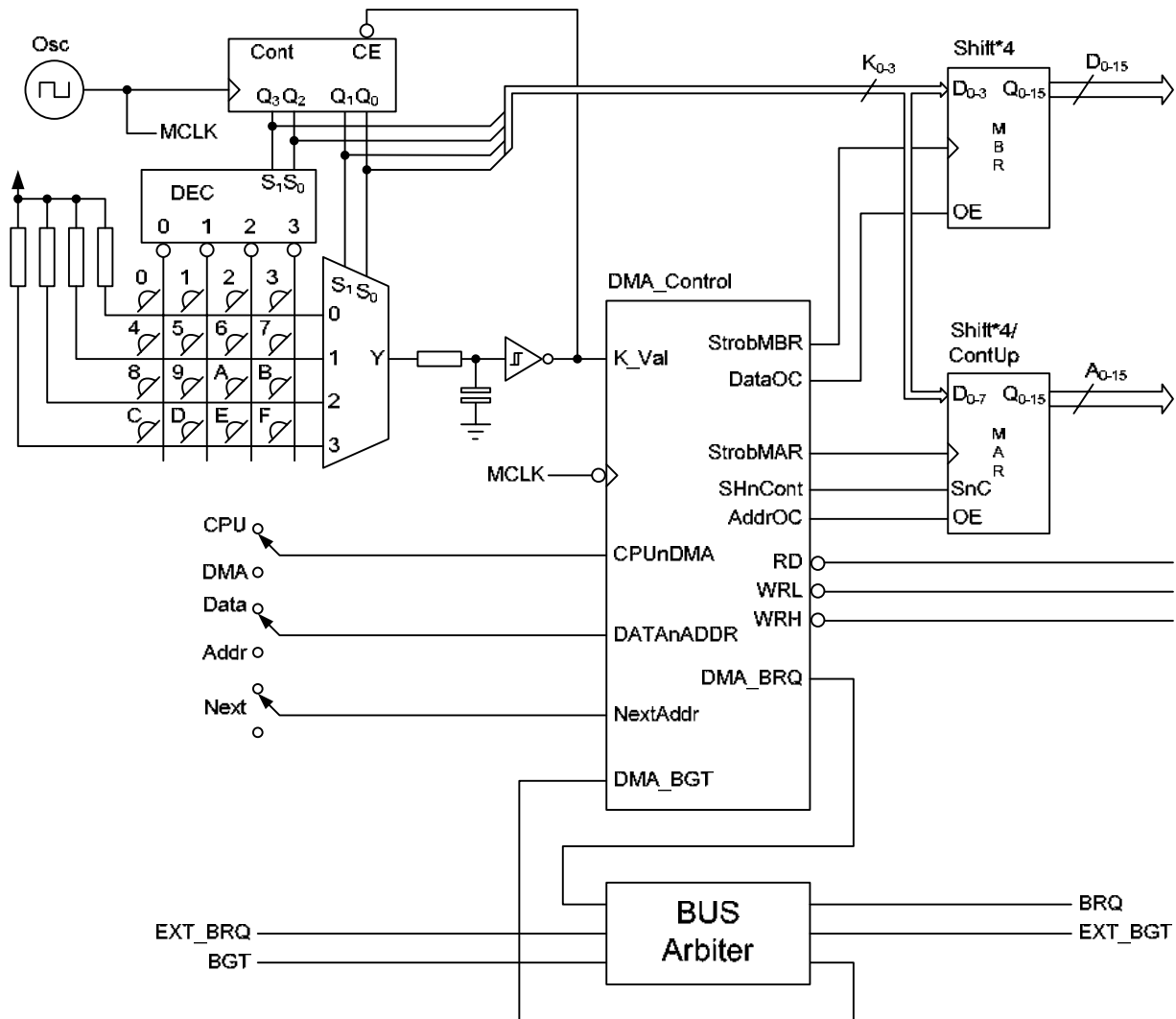


Figura 6

Os vários elementos que constituem o módulo DMA são implementados pelos LATCH U20 e U21 e pelo conjunto das PALs U21, U22, U26, U27 e U28.

3.7 Canal de teste USB / JTAG

Para teste dos programas poderá ser utilizada a plataforma computador pessoal PC. Para utilização dessa plataforma é necessária a adaptação de um canal de comunicação do PC ao protocolo que o CPU em teste suportar. O SDP16 já põe disponível um conversor USB para protocolo série assíncrono full duplex START/STOP 9600. Quanto ao JTAG, o SDP16 põe disponível um fixa JTAG standard ligada a quatro pinos do CPU.

3.8 Fonte de alimentação

Com vista a permitir a utilização de transformadores/rectificadores (*AC adapter*) de uso vulgar, o sistema assume a filtragem, a correcta polarização, a regulação da tensão e a limitação da corrente máxima.

Na entrada deve apresentar-se tensão rectificada (pelo menos “meia onda”) com valor eficaz próximo de 8V, para garantir o funcionamento do regulador linear de 5V (U2 e U3).

A escolha da série HCT (no que se refere à família TTL), leva a que a corrente máxima exigida à fonte fique abaixo de 1A (limite do regulador), reservando cerca de 150 mA para alimentação de circuitos externos, envolvidos em pequenos trabalhos de laboratório.