Periféricos

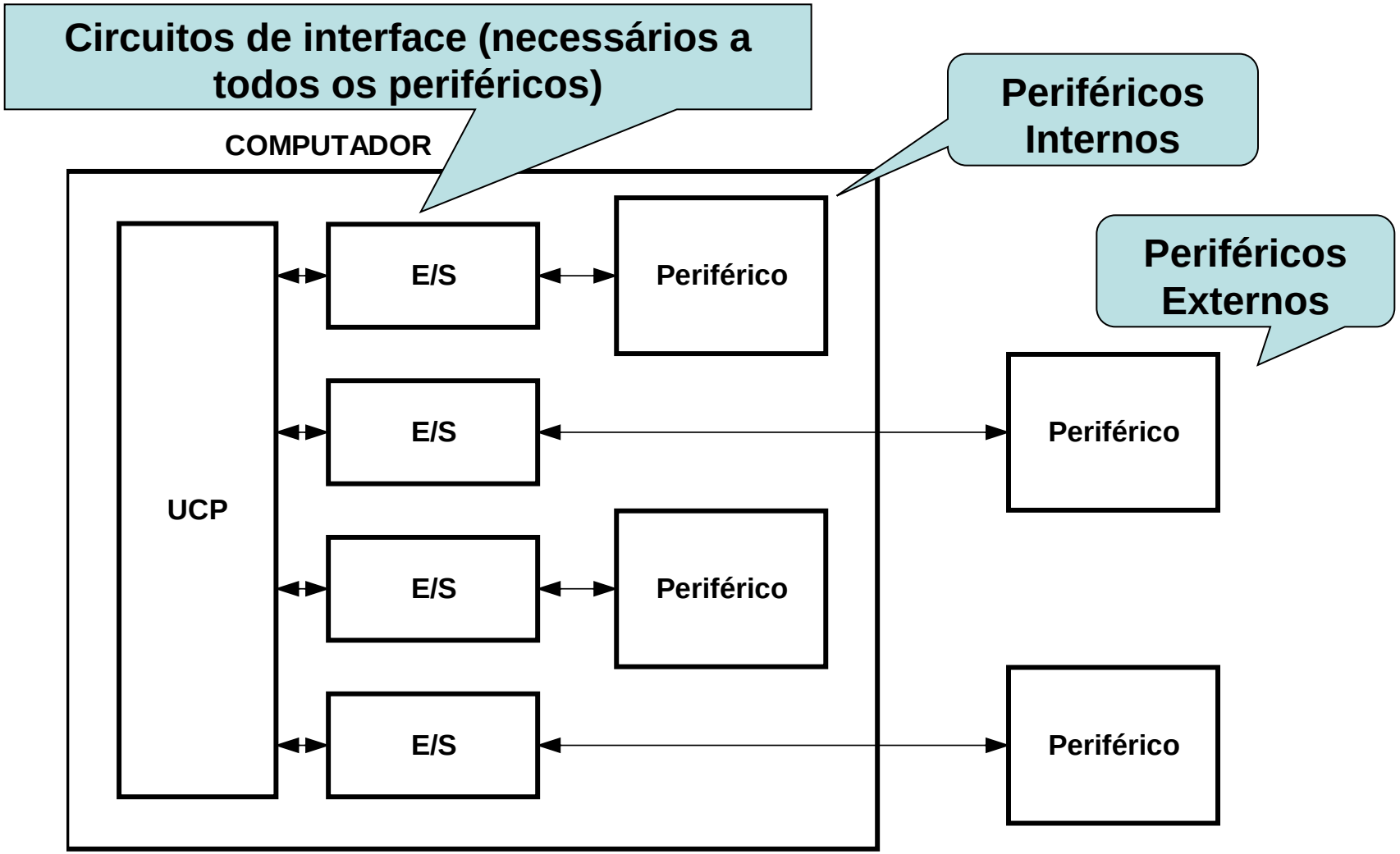
Prof. Sérgio L. Cechin

.inf
INSTITUTO
DE INFORMÁTICA
UFRGS

Entradas e Saídas

- Os computadores comunicam-se com o meio ambiente através dos periféricos
- Exemplos de periféricos **internos** ao computador
  - Controlador de interrupções
  - Controlador de barramentos de dados
  - Controlador do relógio de tempo real
- Exemplos de periféricos **externos** ao computador
  - Teclado
  - Vídeo
  - Disco
  - Rede
  - Mouse
  - Impressoras
  - etc

Entradas e Saídas



Periféricos no CESAR16i

- Teclado
  - Simulado pelo próprio teclado do PC
- Visor
  - Possui 36 posições
    - Cada posição pode apresentar um caractere
  - Os caracteres são representados em ASCII
    - 20H (espaço) até 7AH (‘z’)
- Temporizador
  - Gera pulsos com periodicidade programável entre 1 e 255 milisegundos

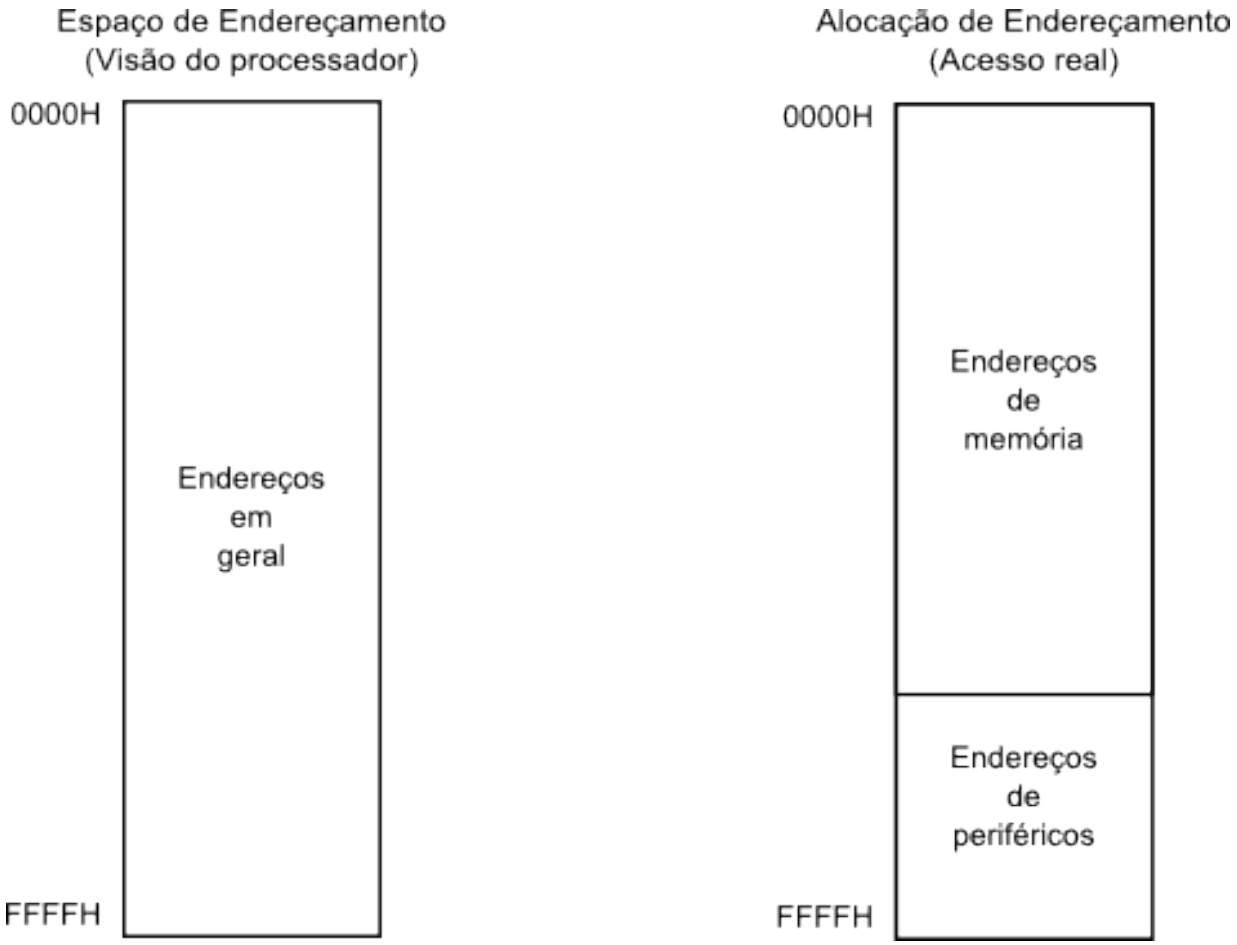
Tabela ASCII

	000	001	010	011	100	101	110	111
0000	null	dle		0	@	P	`	p
0001	soh	dc1	!	1	A	Q	a	q
0010	stx	dc2	“	2	B	R	b	r
0011	etx	dc3	#	3	C	S	c	s
0100	eot	dc4	$	4	D	T	d	t
0101	enq	nak	%	5	E	U	e	u
0110	ack	syn	&	6	F	V	f	v
0111	bell	etb	‘	7	G	W	g	w
1000	bsp	can	(	8	H	X	h	x
1001	ht	em	)	9	I	Y	i	y
1010	lf	sub	*	:	J	Z	j	z
1011	vt	esc	+	;	K	[	k	{
1100	ff	fs	,	<	L	\	l	|
1101	cr	gs	-	=	M	]	m	}
1110	so	rs	.	>	N	^	n	~
1111	si	us	/	?	O	_	o	del

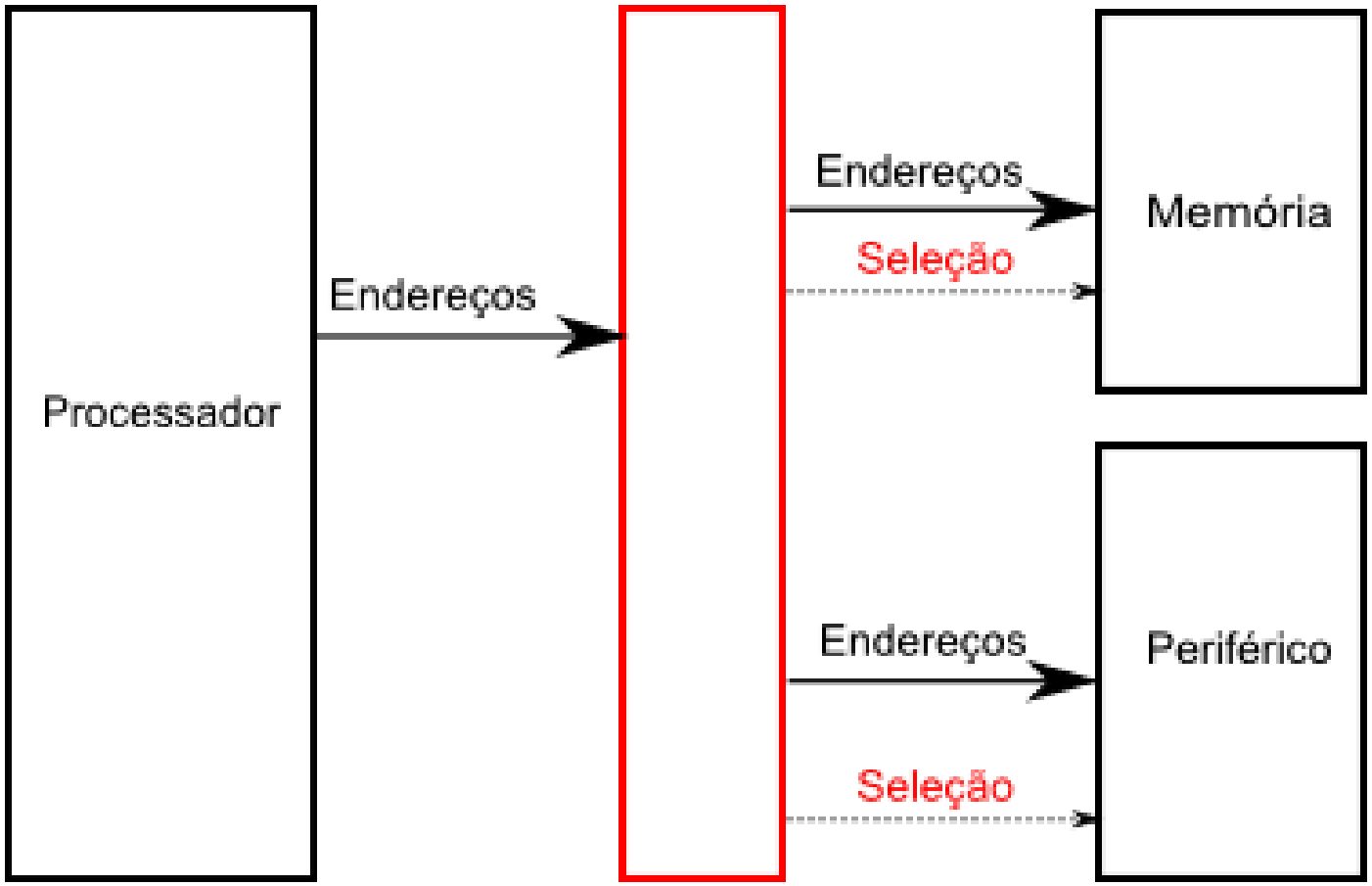
Formas de acesso à E/S

- Memory Mapped I/O
  - As E/S são mapeadas em endereços de memória
  - Alguns bytes da memória são perdidos
  - O acesso ao periférico é feito com as mesmas instruções que o acesso à memória
  - Ex: Motorola, CESAR
- I/O Ports (I/O Mapped I/O)
  - Cria-se uma espaço de endereçamento adicional
  - Os computadores geram dois sinas de seleção: memória e E/S
  - São necessárias instruções específicas
  - Ex: Intel

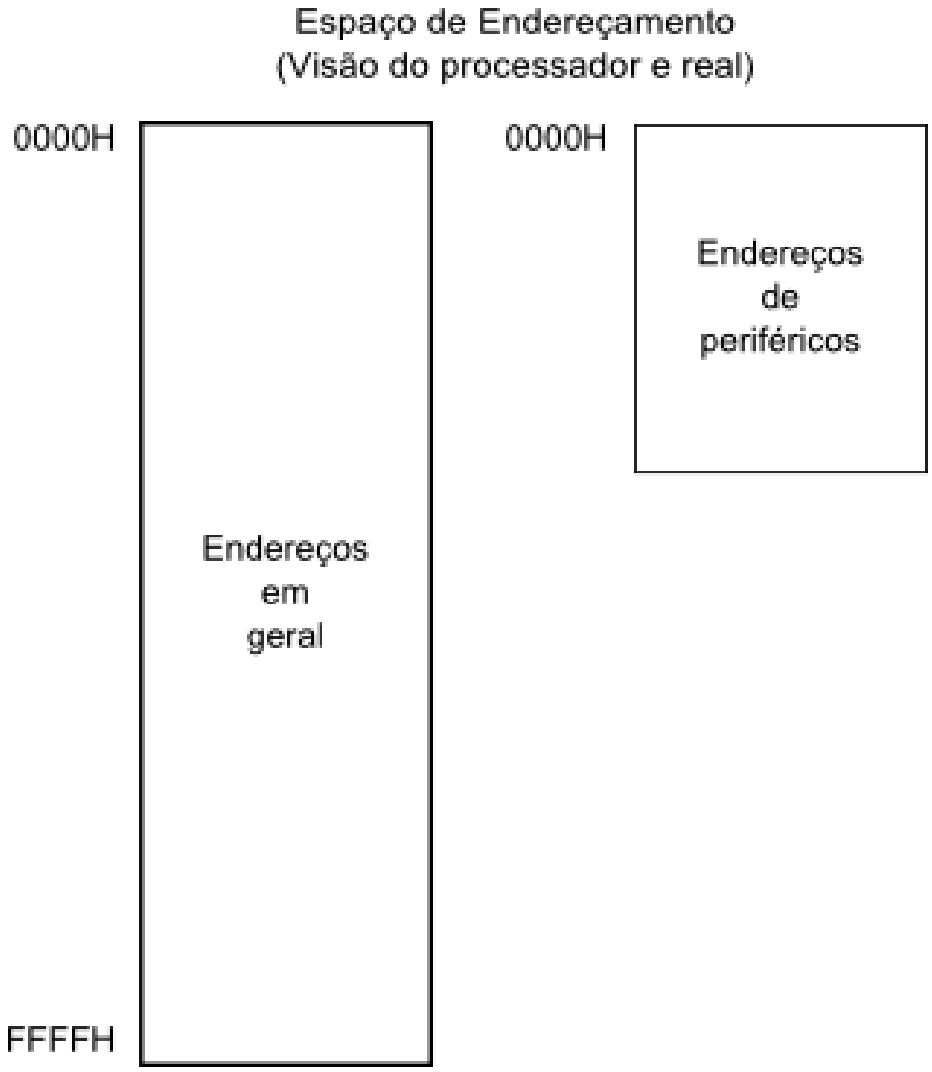
Memory Mapped I/O



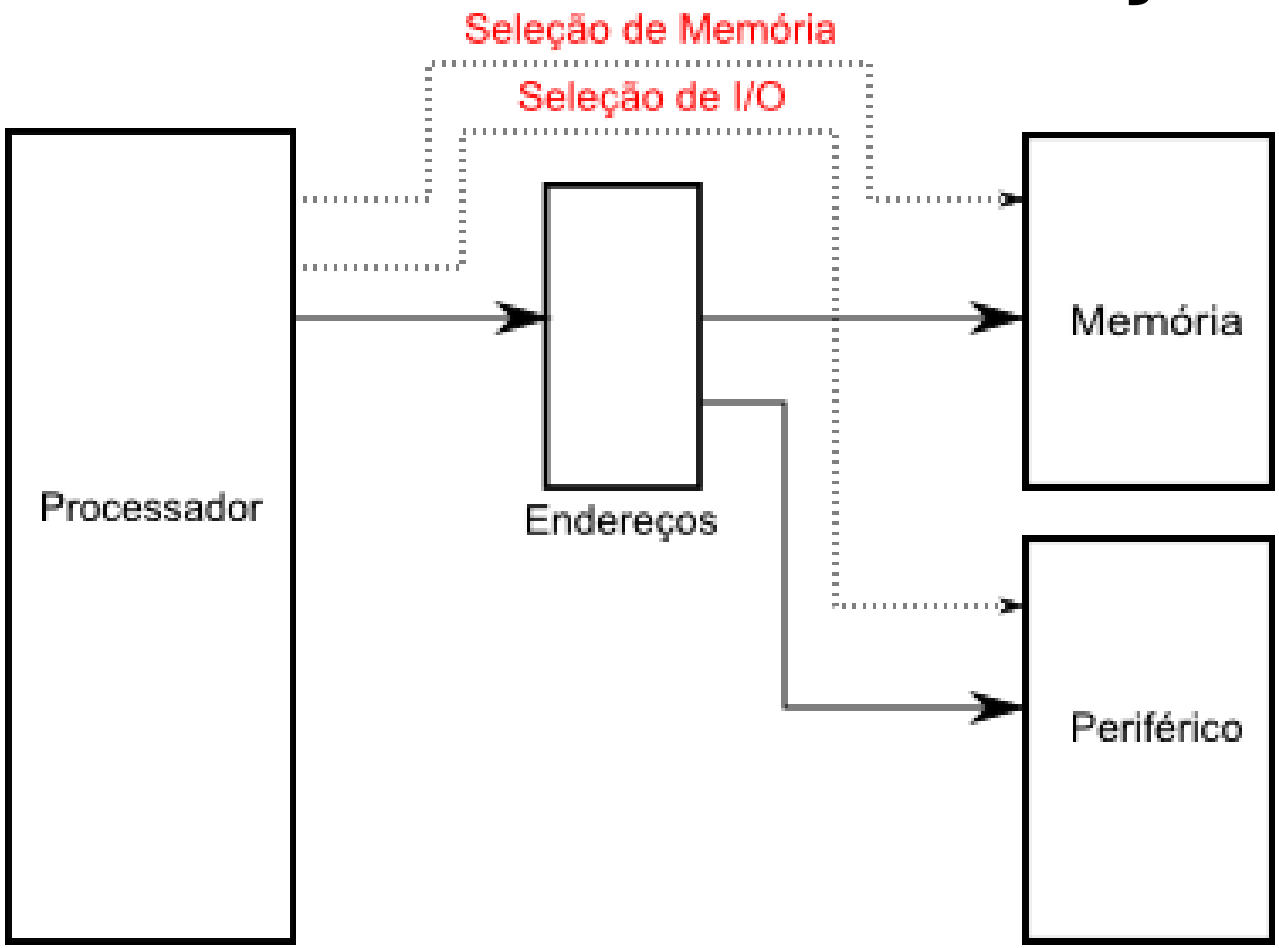
Circuito de decodificação



I/O Ports



Circuito de decodificação



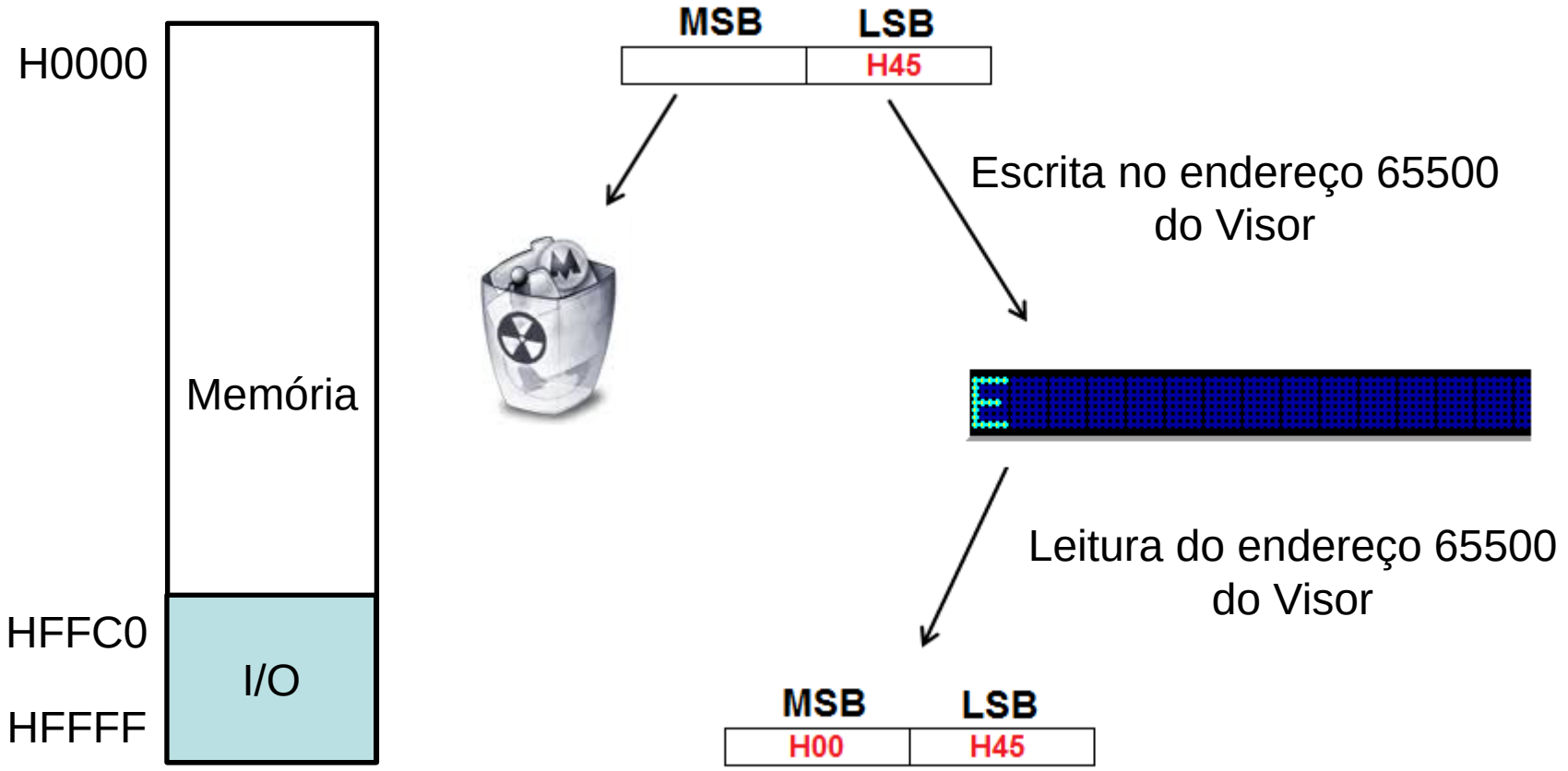
Periféricos no CESAR16i

Acesso aos periféricos

- Acesso aos endereços na região HFFC0 até HFFFF
  - São acessados byte a byte
  - Acesso diferente do acesso à memória
  - Quem faz isso é o “Circuito de Entrada”
- Escrita: MOV R0,Endereco
  - Escreve LSByte de R0 em Endereco
  - Ignora o MSByte
- Leitura: MOV Endereco,R0
  - Lê Endereco para LSByte de R0
  - Zera o MSByte do R0

Acesso aos periféricos

(região FFC0 até FFFF)



Endereços Especiais (Periféricos)

- Timer
  - Programação do timer –TIMDT: HFFD7 (65495)
  - Notificação de tempo – INTS: HFFD8 (65496)
- Teclado
  - Status do Teclado – TECST: HFFDA (65498)
  - Dado do Teclado – TECDT: HFFDB (65499)
- Visor
  - HFFDC (65500) até HFFFF (65535)

Operação do Teclado

- Se TECST=80H, então há tecla
  - O usuário digitou alguma tecla
  - Esta ficou armazenada
  - E este fato é sinalizado pelo TECST
- Se há tecla, pode-se ler de TECDT
  - Basta ler o valor de TECDT
  - Se TECST≠80H, o valor de TECDT é sem significado
- Após lê-la, deve-se “resetar” o TECST (escrever 00H)
  - Cuidado! Só “resetar” após ler TECDT, para impedir a alteração de TECDT
  - Deve-se “resetar” TECST, para liberar o armazenamento de uma nova tecla

Operação do Visor

- Escrita: escreve-se o código ASCII do caractere no endereço da posição desejada
  - A cada operação é escrito apenas o byte menos significativo da origem (registro ou memória)
  - O mais significativo é ignorado
- Leitura: lê-se o endereço da posição desejada
  - A cada operação é lido apenas o byte menos significativo
  - O mais significativo é zerado

Operação do Timer

- Escrita: escreve-se a periodicidade de notificação do timer em TIMDT
  - O valor escrito deve estar em milisegundos
  - O valor tem 1 byte, sem sinal
    - Portanto, pode-se programar para 1ms até 255ms
- A programação com valor “0” fornece um tempo de 100 milisegundos
- Para verificar se houve notificação, deve-se ler INTS
  - Se o bit “0” estiver ligado (“1”), houve notificação
    - Após detectada a notificação, deve-se desligar esse bit
- De forma semelhante aos anteriores
  - A cada operação é escrito apenas o byte menos significativo da origem (registro ou memória)
  - O mais significativo é ignorado

Exemplo 1

- Escrever um programa para ler teclado.
- Assim que entrar uma tecla, deve ser escrito “Oi” no visor
- Encerrar o programa

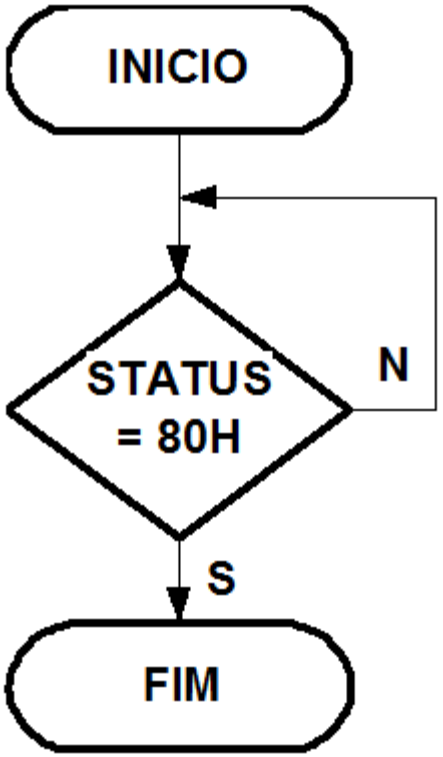
Solução em “C”

```c
#include <cesar.h>

void main(void) {
    waitkey();
    visor("Oi");
}
```

- A solução em “C” não ajudou muito!
- O motivo é que o acesso à periféricos em “C” depende de bibliotecas
  - Então, temos que criar uma biblioteca para acesso aos periféricos do CESAR

Biblioteca: waitkey()

```
void waitkey(void);
```



Visor(“Oi”)

- Não vamos fazer a biblioteca do visor (por enquanto)
- Vamos implementar o acesso direto ao visor

```
hw_visor[0] = 'O';
hw_visor[1] = 'i';
```

Exemplo1.ced

Exemplo 2

- Escrever um programa para ler teclado e colocar esta tecla no visor
- O programa deve apresentar todas as teclas digitadas, até que seja digitada a letra “F” (maiúscula)

Solução em “C”

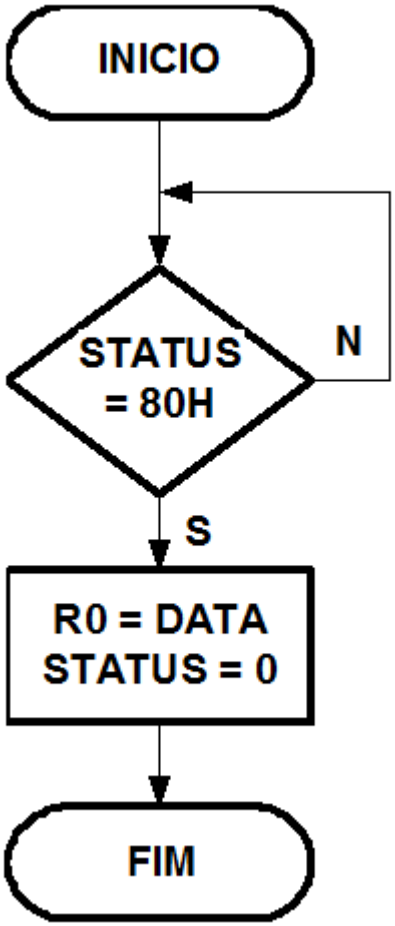
```
void main(void) {
    char c;

    while ( (c=getchar()) != 'F') {
        hw_visor[0] = c;
    }
}
```

- Estamos usando nova função (que vai estar na biblioteca)
- Continuamos acessando o visor de forma direta

Biblioteca: getchar()

```
char getchar(void);
```

- Onde retornar o valor lido?
- Sugestão: no R0
  - Na realidade, vamos definir que todas a funções devem retornar valores pelo R0



getchar()

```
char getchar(void) {
    char c;

    while ( hw_status != 0x80 );
    c = hw_data;
    hw_status = 0;

    return c;
}
```

```
;-----------------------------------
; char getchar(void)
; char c -> R0
getchar:

; while ( hw_status != 0x80 );
getchar_While:
    CMP     STATUS,#H80
    BNE     getchar_While

; c = hw_data;
    MOV     DATA,R0

; hw_status = 0;
    CLR     STATUS

; return c;
    RTS     R7
```

main

```
void main(void) {
    char c;

    while ( (c=getchar()) != 'F') {
        hw_visor[0] = c;
    }
}
```

```
; void main(void) {
; char c ---> R0

; while ( (c=getchar()) != 'F') {
InicioWhile:
    JSR     R7,getchar
    CMP     R0,#H46
    BEQ     FimWhile

; hw_visor[0] = c;
    MOV     R0,VISOR
    JMP     InicioWhile

; }
FimWhile:

; }
    HLT
```

Exemplo2.ced

Exemplo 3

- Escrever um programa para colocar um contador de 1 dígito no visor
  - Deve iniciar em “0”
  - Ao chegar a “9”, deve retornar a “0”
- O programa deve incrementar o contador em passos de 1 segundo

Solução em “C”

```
void main() {
        TIMDT = 0;
        VISOR = c = '0';
        n=0;
        while(1) {
                if (INTS & 1) {
                        INTS &= 0xFE
                        n++;
                        if (n>=10) {
                                n=0;
                                c++
                                if (c>'9') c='0';
                                VISOR=c;
                        }
                }
        }
}
```

Exemplo3.ced

Periféricos

Prof. Sérgio L. Cechin

.inf
INSTITUTO
DE INFORMÁTICA
UFRGS