

CESAR

E/S, subrotinas e organização

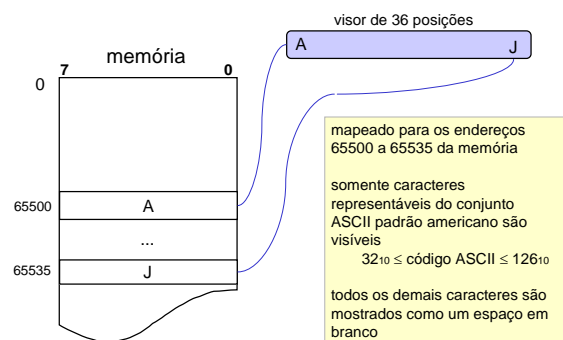
Entrada e Saída

- visor alfanumérico de 36 posições
 - permite visualizar:
 - letras (maiúsculas e minúsculas)
 - dígitos (0 a 9)
 - caracteres especiais do conjunto ASCII padrão americano (códigos ASCII 0 a 127)
- teclado
 - permite ler um caractere (ASCII) e testar se uma tecla foi pressionada

Entrada e Saída

- E/S mapeada na memória
- os últimos 38 endereços de memória (65498 a 65535) são mapeados para os dois periféricos
- transferências para esta área são sempre de 1 byte
 - somente os 8 bits menos significativos do operando são transferidos

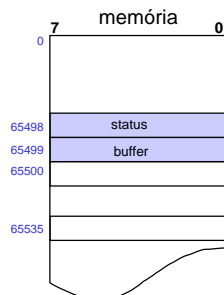
Visor alfanumérico



Teclado

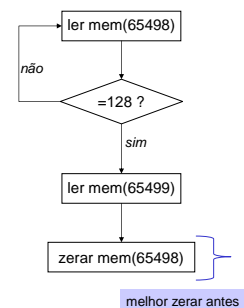


- mapeada para dois bytes da memória
- o byte 65499 contém o último caractere digitado ("buffer")
- o byte 65498 indica o estado do teclado ("status")
 - Valor 128 (80H) : foi digitado um caractere
 - Valor 0: nenhuma tecla foi pressionada

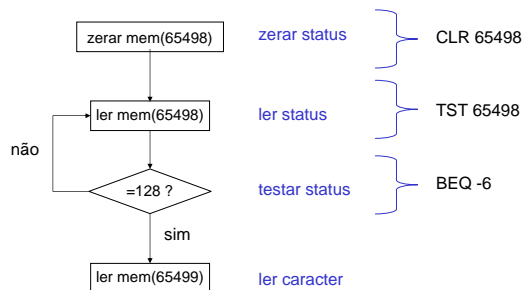


Uso do teclado

- Esperar até que o endereço 65498 (status) contenha 128
- Ler do endereço 65499 (buffer) o código ASCII digitado
- Após, o endereço 65498 deve ser zerado
 - isto indica que outro caractere pode ser recebido do teclado
- Enquanto o endereço 65498 não for zerado, todas as teclas pressionadas são ignoradas



Fluxograma

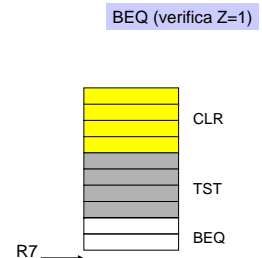


Código para teste de estado

- Rotina para testar se caractere está disponível

```

CLR 65498
TST 65498
BEQ -6 ; (250)
....
  
```

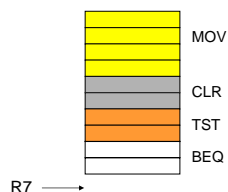


Código mais eficiente

- Alternativa mais eficiente - para digitadores profissionais ;-)

```

MOV #65498, R3
CLR (R3)
TST (R3)
BEQ -4
...
  
```



Leitura com eco

- Leitura de caractere com eco (exibição no visor)

```

MOV #65498, R3
CLR (R3)
TST (R3)
BEQ -4 ; (252)
MOV 65499, 65500
BR -14 ; (242)
  
```

BR (pula sempre)

Exemplo de E/S

- Leitura de caractere com eco e fila (exibição em posições consecutivas do visor)

```

MOV #65498, R3
MOV #65500, R1
CLR (R3)
TST (R3)
BEQ -4 ; (252)
MOV 65499, (R1)
INC R1
BR -14 ; (242)
  
```

Exemplo de E/S

- Resolvendo o problema de "estouro" do visor

```

MOV #65498, R3
MOV #65500, R1
CLR (R3)
TST (R3)
BEQ -4 ; (252)
MOV 65499, (R1)
INC R1
BEQ -18 ; (238)
BR -16 ; (240)
  
```

Subrotinas

- permite recursividade, reentrância, aninhamento e passagem de parâmetros
- usa um registrador de ligação
 - o valor do registrador de ligação é salvo na pilha
 - o registrador de ligação recebe o endereço de retorno

se o registrador de ligação for R7, então
R7 é simplesmente salvo na pilha

Subrotina: desvio e retorno

JSR r, endereço de desvio

temp ← endereço de desvio (modo 0 = NOP)
pilha ← registrador r
registrador r ← PC
PC ← temp

r – registrador de ligação

RTS r

PC ← registrador r
registrador r ← pilha

Passagem de parâmetros

- passagem de parâmetros que seguem a instrução de chamada da subrotina no programa
 - por valor
 - por nome (endereço)
- passagem pela pilha

Subrotina - caso 1

- Três parâmetros, por valor

100 JSR R5, 1000	1000 MOV (R5)+,R1
104 param1	1002 MOV (R5)+,R2
106 param2	1004 MOV (R5)+,R3
108 param3	1006
110 <próx.instr.>
	1100 RTS R5

Subrotina - caso 2

- Três parâmetros, último por nome (endereço)

100 JSR R5, 1000	1000 MOV (R5)+,R1
104 param1	1002 MOV (R5)+,R2
106 param2	1004 MOV ((R5)+),R3
108 end.param3	1006
110 <próx.instr.>
	1100 RTS R5

Subrotina - caso 3

- Dois parâmetros e um resultado

100 JSR R5, 1000	1000 MOV (R5)+,R1
104 param1	1002 MOV (R5)+,R2
106 param2	1004 MOV (R5)+,R4
108 end.resultado	1006
110 <próx.instr.>
	1096 MOV resul,(R4)
	1100 RTS R5

Subrotina - parâmetros na pilha

- Três parâmetros
 100 MOV param1, -(R6)
 102 MOV param2, -(R6)
 104 MOV param3, -(R6)
 106 JSR R7, 1000
 110 <próx.instr.>
- Como obter os parâmetros e deixar a pilha em ordem ?

Subrotina - parâmetros na pilha

- Como deixar a pilha em ordem ?
- Não mexer na pilha !
 100 MOV param1, -(R6)
 102 MOV param2, -(R6)
 104 MOV param3, -(R6)
 106 JSR R7, 1000
 110 ADD #6, R6 ; 6 bytes = 3 parâmetros
 114 <próx.instr.>

Subrotina - parâmetros na pilha

- Como obter os parâmetros ?
- Modo indexado !
 100 MOV param1, -(R6) 1000 MOV 6(R6), R1
 102 MOV param2, -(R6) 1002 MOV 4(R6), R2
 104 MOV param3, -(R6) 1004 MOV 2(R6), R3
 106 JSR R7, 1000 1006
 110 ADD #6, R6
 114 <próx.instr.> 1100 RTS R7

Subrotina - parâmetros na pilha

- Como obter os parâmetros ?
 100 MOV R6, R5 1000 MOV -2(R5), R1
 102 MOV param1, -(R6) 1002 MOV -4(R5), R2
 104 MOV param2, -(R6) 1004 MOV -6(R5), R3
 106 MOV param3, -(R6) 1006
 108 JSR R7, 1000
 112 ADD #6, R6 1100 RTS R7
 116 <próx.instr.>

Organização do Cesar

- visão geral
 - organização detalhada envolve técnicas de otimização e decisão mais sofisticadas
 - várias organizações possíveis para a mesma arquitetura
 - aqui é mostrada uma possível em alto nível de abstração

