

EEL7030 - Microprocessadores

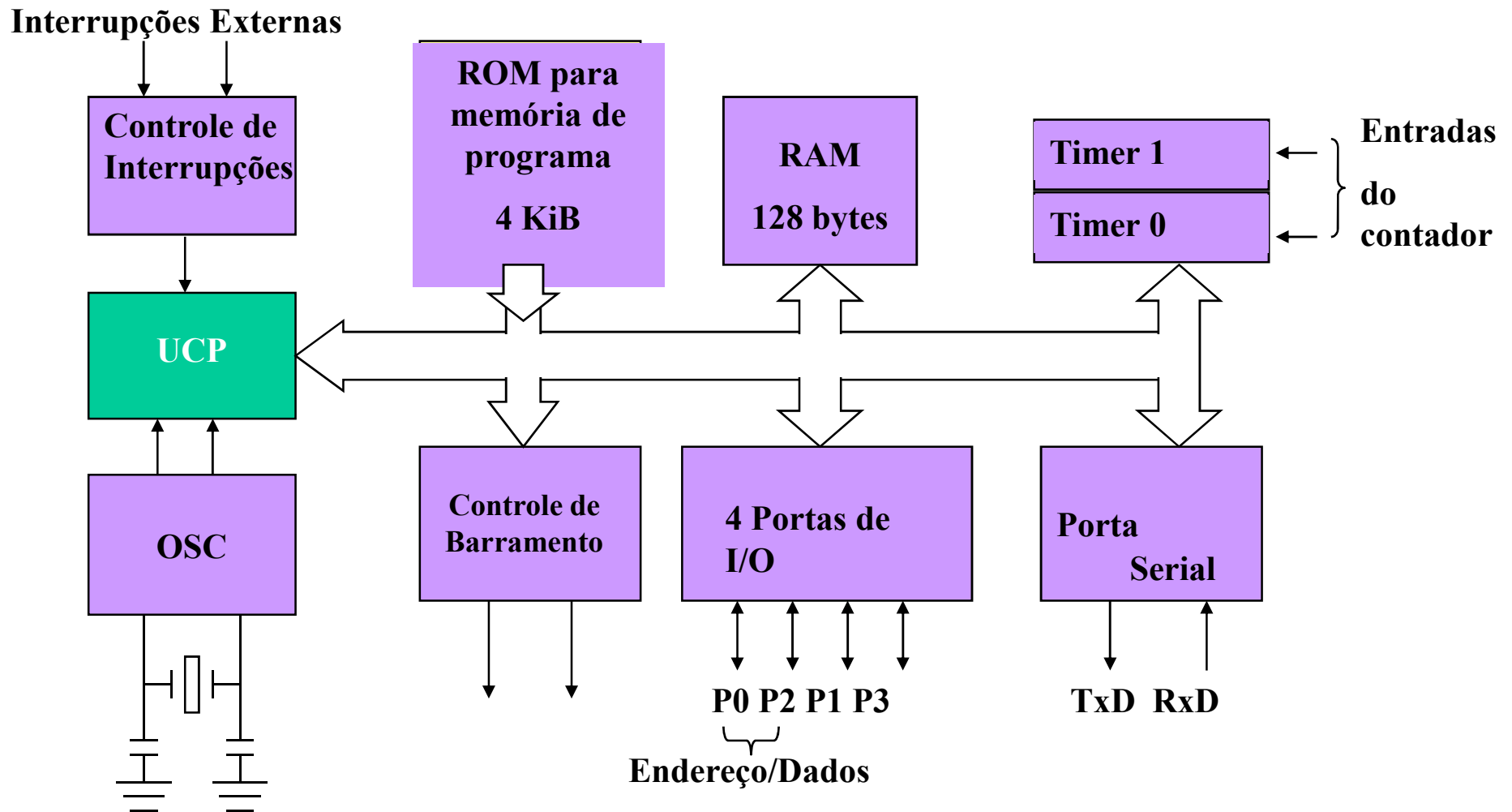


LCS

Laboratório de
Comunicações
e Sistemas
Embarcados

Prof. Raimes Moraes
EEL -UFSC

8051

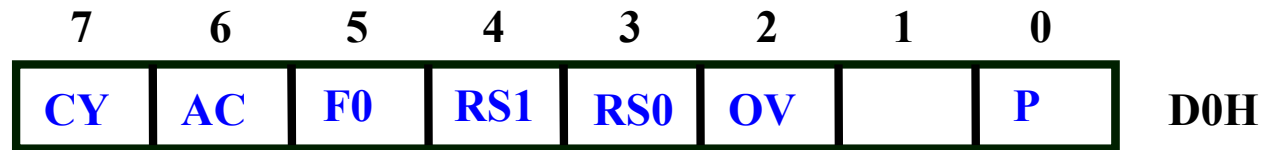


Organização da RAM Interna

7F	7E	7D	7C	7B	7A	79	78	2FH
...								...
0F	0E	0D	0C	0B	0A	09	08	21H
07	06	05	04	03	02	01	00	20H
R0 - R7 Banco 3								1FH 18H
R0 - R7 Banco 2								17H 10H
R0 - R7 Banco 1								0FH 08H
R0 - R7 Banco 0								07H 00H

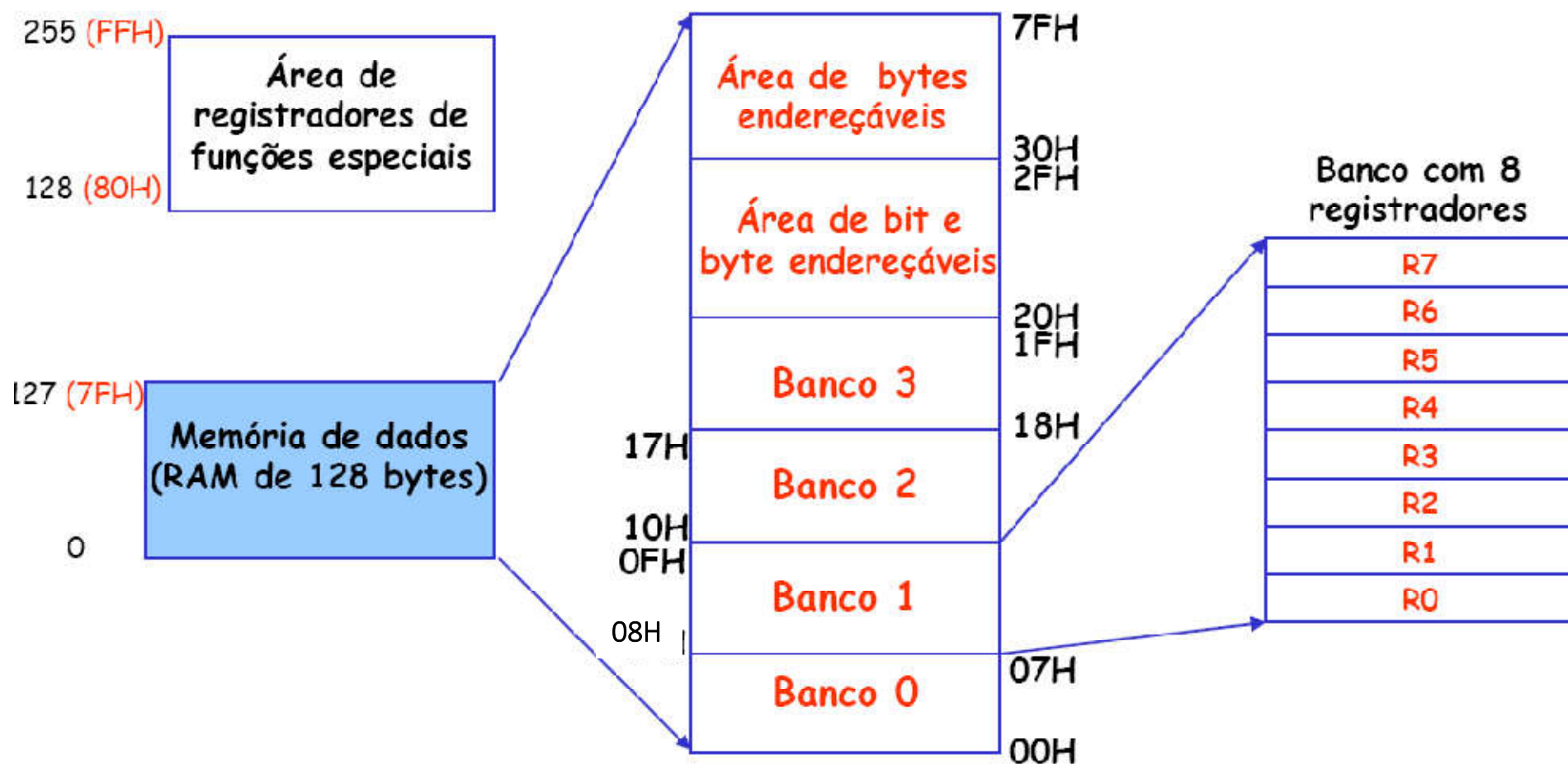
Registrador de Flags

PSW - *Program Status Word - Bit Addressable*



Nome	Localização	Descrição
CY	PSW.7	Carry flag
AC	PSW.6	Auxiliary carry flag
F0	PSW.5	Definido pelo usuário
RS1	PSW.4	Bit 1 do seletor de Register Bank
RS0	PSW.3	Bit 0 do seletor de Register Bank
OV	PSW.2	Overflow flag
	PSW.1	Definido pelo usuário
P	PSW.0	Flag de paridade. 1 = ímpar.

Organização da RAM Interna



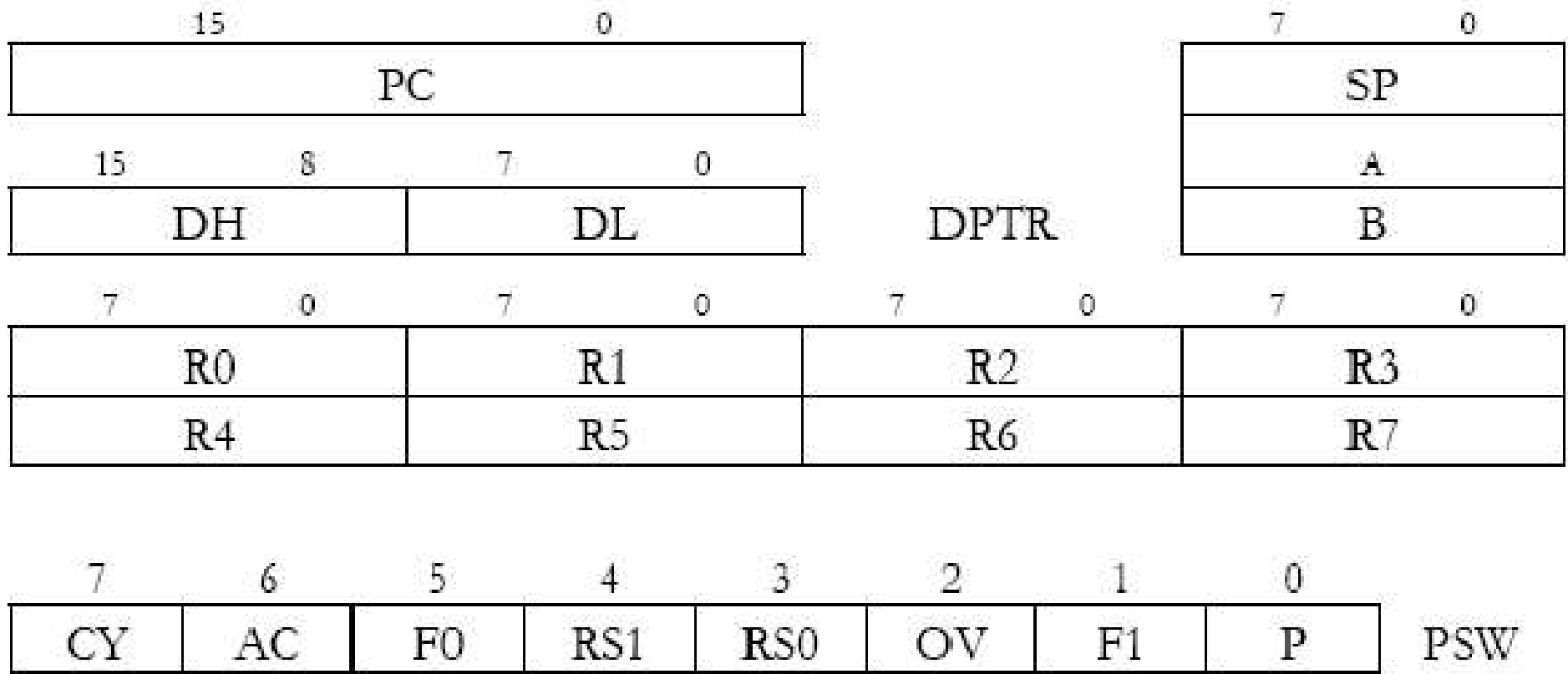
**RS0 e RS1 do
registrador PSW**

	RS0=0	RS0=1
RS1=0	Banco 0	Banco 1
RS1=1	Banco 2	Banco 3

Pilha

- Região da memória RAM utilizada pelo programador e processador;
- Pelo programador: armazenar dados temporariamente;
- Pelo processador: armazenar endereços e *flags* quando de alteração do fluxo de execução do programa (subrotina e interrupção);

Registadores do 8051



Pilha

- Em microprocessadores, programador deve informar o processador sobre área de memória RAM disponível para a pilha, inicializando o registrador *Stack Pointer* [SP - ponteiro de pilha]*.
- No 8051, ao ser resetado, o SP é inicializado com o valor 07H

***OBS: Quando o processador executa sistema operacional, este gerencia a pilha.**

Pilha

[Exemplo de utilização da pilha pelo programador]

Supondo: $[A] = CAH$; $[SP] = 07H$

END.	Mnemônico
010E	PUSH ACC
0200	MOV A,#3
0202	ADD A,R0
0203	MOV R0,A

PUSH ACC

PILHA	END.	DADO
SP	07H (valor inicial)	
SP+1	08H (valor final)	CAH

Eventos durante execução

$[SP] \leftarrow [SP] + 1$

$\{[SP]\} \leftarrow [A]$

$[] \Rightarrow$ conteúdo; $\{\} \Rightarrow$ endereço de memória apontado por

Pilha

[Exemplo de utilização da pilha pelo programador]

Supondo: [SP]=08H

END.	Mnemônico
010E	PUSH ACC
0200	MOV A,#3
0202	ADD A,R0
0203	MOV R0,A
0204	POP ACC

PILHA	END.	DADO
SP-1	07H (valor final)	
SP	08H (valor inicial)	CAH

POP ACC

Eventos durante execução

[A] ← {[SP]}

[SP] ← [SP] - 1

[] => conteúdo; {}=> endereço de memória apontado por

Pilha

OBS: **UTILIZAR ENDEREÇOS DOS REGISTRADORES de R0 a R7**

Supondo: $[SP]=1FH$ $[R0] = 3BH;$
(do banco 1)

END.	Mnemônico
010E	PUSH 08H
0200	...
0202	...
0203	...
0204	POP 08H

(R0 do banco 1)

PUSH 08H

PILHA	END.	DADO
SP	1FH (valor inicial)	
SP+1	20H (valor final)	3BH

Eventos durante execução

$[SP] \leftarrow [SP] + 1$

$\{[SP]\} \leftarrow [08H]$

$[] \Rightarrow$ conteúdo; $\{\} \Rightarrow$ endereço de memória apontado por

Pilha

OBS: **UTILIZAR ENDEREÇOS DOS REGISTRADORES de R0 a R7**

Supondo: **[SP]=20H**

END.	Mnemônico
010E	PUSH 08H
0200	...
0202	...
0203	...
0204	POP 08H

(R0 do banco 1)

(R0 do banco 1)

[R0] = 3BH;
(do banco 1)

POP 08H

PILHA	END.	DADO
SP-1	1FH (valor final)	
SP	20H (valor inicial)	3BH

Eventos durante execução

[08] ← {[SP]}

[SP] ← [SP] - 1

[] => conteúdo; {}=> endereço de memória apontado por

Exemplo de uso de bancos de registradores

(OBS: Modesta compressão do código da subrotina com uso de banco)

; Programa Principal

```
CLR    RS0
CLR    RS1
MOV    SP,#0FH
```

.....

```
MOV    R0,#5
MOV    R1,#0
```

```
RPSUB: MOV    A,P2
        PUSH  ACC
        CALL  SUBRT
        POP   ACC
        ADD   A,R1
        MOV   R1,A
        DJNZ  R0, RPSUB
```

.....

; V1 - SEM MUDAR O BANCO DE REGISTRADORES SUBRT:

```
                PUSH  00H
                PUSH  01H
                MOV    R0,#8
                MOV    R1,#0
Loop:          RLC    A
                JNC    Salta
                INC    R1
Salta :        DJNZ  R0,Loop
                MOV    P1,R1
                POP    01H
                POP    00H
                RET
```

; V2 - MUDANDO O BANCO DE REGISTRADORES SUBRT:

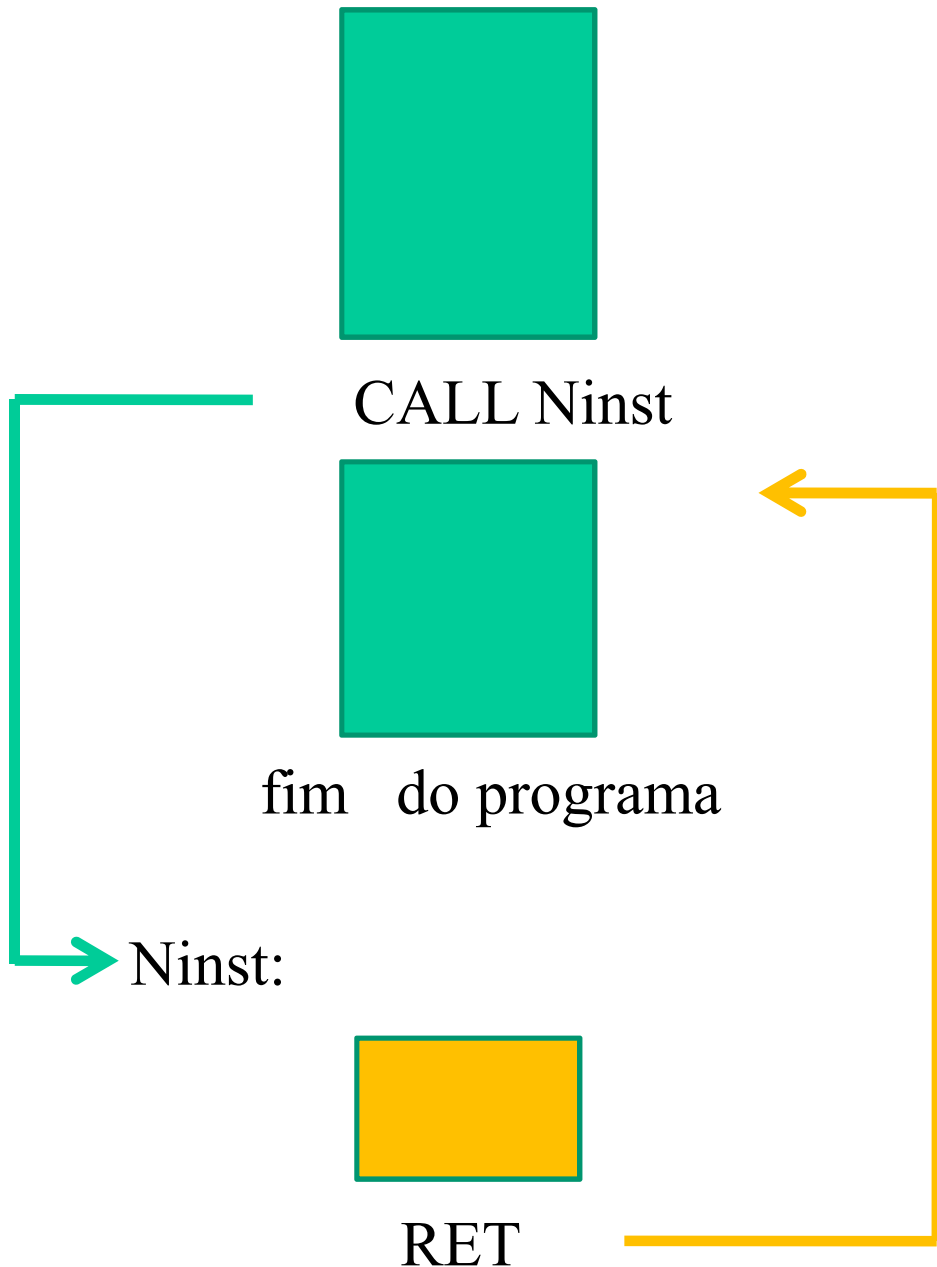
```
                SETB   RS0
                MOV    R0,#8
                MOV    R1,#0
Loop:          RLC    A
                JNC    Salta
                INC    R1
Salta :        DJNZ  R0,Loop
                MOV    P1,R1
                CLR    RS0
                RET
```

SUBROTINA

- Conjunto de instruções para o qual o fluxo de execução do programa é desviado pela instrução:

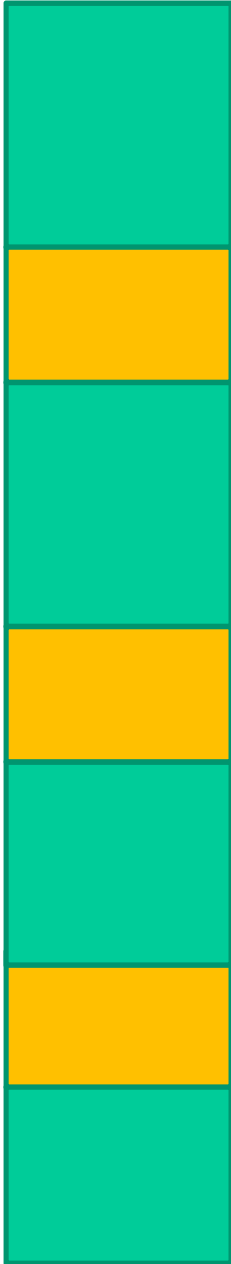
CALL [endereço]

- A instrução **RET** faz com que o microprocessador retorne à executar instrução que se segue à chamada da subrotina.

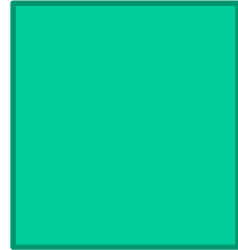


[Aguarde!!! Voltaremos a falar da pilha em breve]

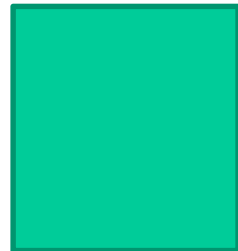
Programa
Sem Subrotina



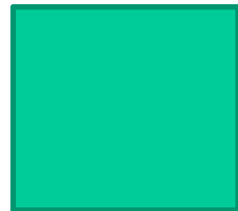
Programa
Com Subrotina



CALL Ninst



CALL Ninst



CALL Ninst



SUBROTINA

Conjunto de N
instruções repetidas

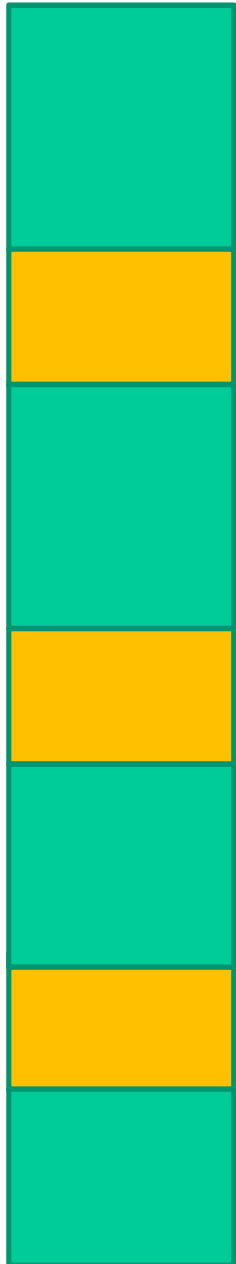
Ninst:



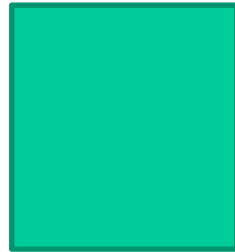
RET

SUBROTINA

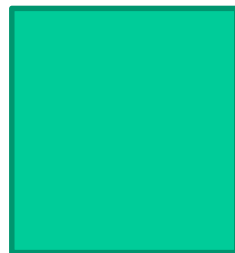
Programa
Sem Subrotina



Programa
Com Subrotina



CALL Ninst



CALL Ninst



CALL Ninst



- **Vantagens:**

- **Menor volume de código;**
- **Código mais inteligível;**

- **Desvantagem:**

- **Menor velocidade de execução**

Ninst:



RET

Mnemônicos para SUBROTINA (2 cycles)

LCALL: Especifica endereço de 16 bits. A instrução possui 3 bytes (opcode + 16 bits de endereço). Endereço de destino em qualquer lugar da memória (64 kiB).

ACALL: Especifica endereço de 11 bits. A instrução possui 2 bytes (opcode + 11 bits de endereço). Endereço de destino distante em até 2KiB (2^{11}).

SUBROTINA

1 8051 lê código de 3 bytes de LCALL e atualiza o valor do PC (de 000AH para 000DH)

2 Salva atual PC (000DH) na pilha.

....

Eventos durante execução de LCALL

$[SP] \leftarrow [SP] + 1$

$\{[SP]\} \leftarrow [PC_{LSB}]$

$[SP] \leftarrow [SP] + 1$

$\{[SP]\} \leftarrow [PC_{MSB}]$

$[] \Rightarrow$ conteúdo; $\{\} \Rightarrow$ endereço de memória apontado por

END.	Mnemônico	CÓDIGO
000A	LCALL 2028H	12 20 28
000D	MOV A,B	E5 F0

PILHA	END.	DADO
SP	07H (valor inicial)	
SP+1	08H	0DH [PC LSB]
SP+1	09H (valor final)	00H [PC MSB]

[Exemplo de utilização da pilha pelo processador]

SUBROTINA

....

3 Sobrescreve PC com endereço da subrotina; ([PC] = 2028H)

4 Executa subrotina;

5 Retorna à instrução que se segue ao LCALL no programa principal (instrução RET).

END.	Mnemônico	CÓDIGO [Hex]
000A	LCALL 2028H	12 20 28
000D	MOV A,B	E5 F0

END.	Mnemônico	CÓDIGO [Hex]
2028	INC B	05 F0
202A	RET	22

SUBROTINA

....

5 Retorna à instrução que se segue ao LCALL no programa principal (instrução RET).

Eventos durante execução de RET

$[PC_{MSB}] \leftarrow \{[SP]\}$

$[SP] \leftarrow [SP] - 1$

$[PC_{LSB}] \leftarrow \{[SP]\}$

$[SP] \leftarrow [SP] - 1$

$[] \Rightarrow$ conteúdo; $\{\} \Rightarrow$ endereço de memória apontado por

END.	Mnemônico	CÓDIGO
000A	LCALL 2028H	12 20 28
000D	MOV A,B	E5 F0

PILHA	END.	DADO
SP -1	07H (valor final)	
SP-1	08H	0DH [PC LSB]
SP	09H (valor inicial)	00H [PC MSB]

[Exemplo de utilização da pilha pelo processador]

Subrotina modifica registrador cujo conteúdo se quer preservar?

PUSH (endereço direto)



$[SP] \leftarrow [SP] + 1$
 $\{[SP]\} \leftarrow [\text{endereço direto}]$

END.	Mnemônico
010E	PUSH B
0200	LCALL 34C2H
0203	POP B

Sendo:

$[B] = 32H$

$[SP] = 2FH;$

Pilha

[Exemplo de utilização da pilha pelo programador e processador]

Sendo: $[B] = 32h$; $[SP] = 2Fh$

END.	Mnemônico
010E	PUSH B
0200	LCALL 34C2H
0203	...

PUSH B

Eventos durante a execução

$[SP] \leftarrow [SP] + 1$

$\{[SP]\} \leftarrow [B]$

$[PC] \leftarrow [PC] + 3$ (leitura do código de LCALL 34C2H)

$[SP] \leftarrow [SP] + 1$

$\{[SP]\} \leftarrow [PC]_{LSB}$

$[SP] \leftarrow [SP] + 1$

$\{[SP]\} \leftarrow [PC]_{MSB}$

$[PC] \leftarrow 34C2$

L
C
A
L
L

34
C2

PILHA	END.	DADO
SP	2FH (valor inicial)	
SP+1	30H	32H
SP+1	31H	03H
SP+1	32H (valor final)	02H

$[] \Rightarrow$ conteúdo; $\{\} \Rightarrow$ endereço de memória apontado por

Pilha

[Exemplo de utilização da pilha pelo programador e processador]

END.	Mnemônico
0203	POP B
....
34C2
....
343F	RET

POP B

R
E
T

PILHA	END.	DADO
SP	2FH (valor final)	
SP-1	30H	32H
SP-1	31H	03H
SP-1	32H (valor inicial)	02H

Eventos durante execução de RET e POP B

$[PC_{MSB}] \leftarrow \{[SP]\}$

$[SP] \leftarrow [SP] - 1$

$[PC_{LSB}] \leftarrow \{[SP]\}$

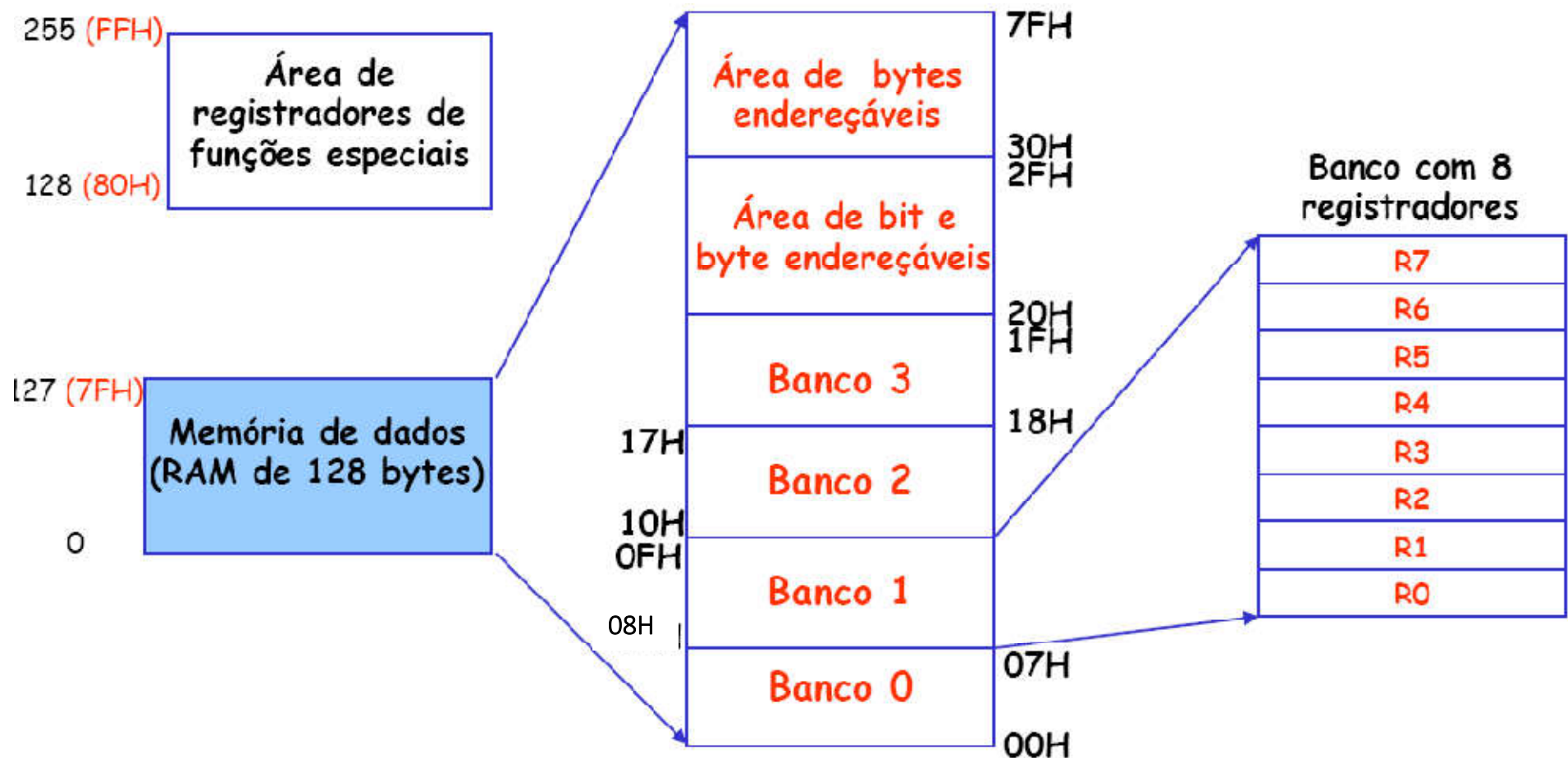
$[SP] \leftarrow [SP] - 1$

$[B] \leftarrow \{[SP]\}$

$[SP] \leftarrow [SP] - 1$

Pilha

Ao se utilizar a mesma, ter em mente a localização do banco 1 de registradores.



Exemplo de uso de bancos de registradores e necessidade de alteração do local da pilha

; Programa Principal

```
CLR    RS0
CLR    RS1
MOV    SP,#0FH
```

.....

```
MOV    R0,#5
MOV    R1,#0
```

```
RPSUB: MOV    A,P2
        PUSH  ACC
        CALL  SUBRT
        POP   ACC
        ADD   A,R1
        MOV   R1,A
        DJNZ  R0, RPSUB
```

.....

; V2 - MUDANDO O BANCO DE REGISTRADORES SUBRT:

```
                SETB    RS0
                MOV     R0,#8
                MOV     R1,#0
Loop:           RLC     A
                JNC     Salta
                INC     R1
Salta :         DJNZ    R0,Loop
                MOV     P1,R1
                CLR     RS0
                RET
```