

ELTD13z
Laboratório
Microcontroladores/Microprocessadores

Prática_03a2

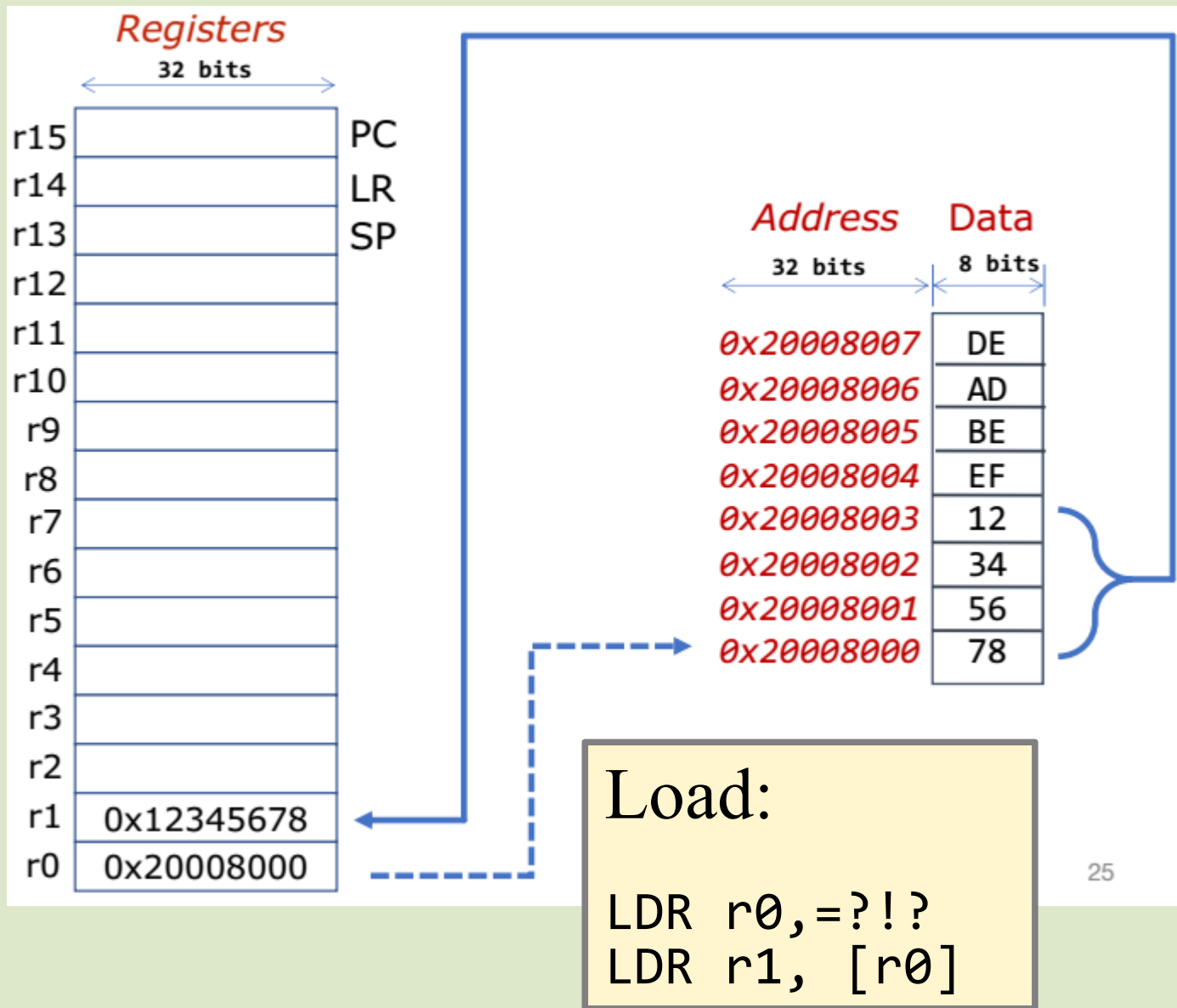
Prof. Enio R. Ribeiro

Universidade Federal de Itajubá - UNIFEI





2. Endereçamento indexado





3. Branch - endereçamento relativo

```
;---Extr_1 - uVision
;--- end. Relativo (desvio
incondicional)
    export __main
;=====
;---diretiva area - dados (ram)
    area    dados_1, data, readonly
ctx1    dcw 0x7cde
;---diretiva area - programa (flash)
    area    m_prog, code, readonly
__main
pk0 ldr r0,=ctx1
    ldrb    r1,[r0]
    b       pk2
pk1 nop
    nop
    b       pk0
pk2 ldrh    r2,[r0]
    add r3,r1,r2
    b       pk1
end
```

0x0800024C	4804
@0x08000260	
0x0800024E	7801
0x08000250	E002
0x08000252	BF00
0x08000254	BF00
0x08000256	E7F9
0x08000258	8802
0x0800025A	EB010302
0x0800025E	E7F8
0x08000260	0264
0x08000262	0800
0x08000264	7CDE

__main:	LDR	r0,[pc,#16] ;
	LDRB	r1,[r0,#0x00]
	B	0x08000258
	NOP	
	NOP	
	B	0x0800024C __main
	LDRH	r2,[r0,#0x00]
	ADD	r3,r1,r2
	B	0x08000252
	DCW	0x0264
	DCW	0x0800
	DCW	0x7CDE



3. Branch - endereçamento relativo

```

;---Extr_1 - uVision
;--- end. Relativo (desvio
incondicional)
    export __main
;=====
;---diretiva area - dados (ram)
    area    dados_1, data, readonly
ctx1      dcw 0x7cde
;---diretiva area - programa (flash)
    area    m_prog, code, readonly
__main
    mov r4,#&aa
    nop
pk0 ldr r0,=ctx1
    ldrb    r1,[r0]
    b      pk2
pk1 nop
    nop
    b      pk0
pk2 ldrh    r2,[r0]
    add r3,r1,r2
    b      pk1
end

```

		main:	
0x0800024C	F04F04AA	MOV	r4,#0xAA
0x08000250	BF00	NOP	
0x08000252	4805	LDR	r0,[pc,#20] ;
@0x08000268			
0x08000254	7801	LDRB	r1,[r0,#0x00]
0x08000256	E002	B	0x0800025E
0x08000258	BF00	NOP	
0x0800025A	BF00	NOP	
0x0800025C	E7F9	B	0x08000252
0x0800025E	8802	LDRH	r2,[r0,#0x00]
0x08000260	EB010302	ADD	r3,r1,r2
0x08000264	E7F8	B	0x08000258
0x08000266	0000	DCW	0x0000
0x08000268	026C	DCW	0x026C
0x0800026A	0800	DCW	0x0800
0x0800026C	7CDE	DCW	0x7CDE



4. Exercícios

DICA importante: leia com muita atenção e até o final cada exercício.

1a) Seja um conjunto de 8 bytes devidamente colocados em memória flash. Escreva um programa cíclico para copiar, byte por byte, isto é, cada valor de 1 byte da memória flash para a memória sram. **Condições:** use SOMENTE as instruções e os modos de endereçamento estudados. Use as diretivas necessárias. Assemblar e testar o programa; analisar o arquivo *.MAP e área de memória.

2a) Seja um conjunto de 8 bytes devidamente colocados em memória flash. Escreva um programa cíclico para dobrar o valor de cada byte e salvá-lo, adequadamente, em memória sram. **Condições:** use SOMENTE as instruções e os modos de endereçamento estudados. Os dados são binários não sinalizados. Use as diretivas necessárias. Assemblar e testar o programa; analisar o arquivo *.MAP e área de memória.

3a) Sejam dois vetores de dados de entrada, alocados em memória sram. Cada vetor possui 4 elementos. Um vetor possui fatores de 1 byte. O outro vetor possui fatores de 2 bytes. Escreva um programa cíclico para somar os elementos de cada conjunto. Em outras palavras: somar o primeiro elemento de um vetor ao primeiro elemento do segundo vetor, e assim por diante. Salve cada soma em um outro vetor de saída, o qual deve estar em memória sram. **Condições:** use SOMENTE as instruções e os modos de endereçamento estudados. Use as diretivas necessárias. Assemblar e testar o programa; analisar o arquivo *.MAP e área de memória.

4a) Sejam dois vetores de dados de entrada, alocados em memória sram. Cada vetor de entrada possui 4 elementos. O vetor vt1 possui elementos de 2 bytes (binário sinalizado). O vetor vt2 possui elementos de 1 byte (binário sinalizado). Faça um programa cíclico de subtração entre os elementos de cada vetor. Em outras palavras: subtrair o primeiro elemento de vt2 do primeiro elemento de vt1 ($vt1_i - vt2_i$) e assim por diante. Salve os resultado em um outro vetor de saída, o qual deve estar em memória sram. **Condições:** use SOMENTE as instruções e os modos de endereçamento estudados. Use as diretivas necessárias. Assemblar e testar o programa; analisar o arquivo *.MAP e área de memória.

5a) Sejam dois vetores de dados de entrada, alocados em memória sram. Cada vetor de entrada possui 4 elementos. O vetor vtx possui fatores de 2 bytes. O vetor vty possui fatores de 2 bytes (binário sinalizado). Faça um programa cíclico para subtrair vty de vtx, elemento a elemento. Em outras palavras: (o primeiro elemento de vtx) – (o primeiro elemento de vty) e assim por diante. Salvar cada resultado da subtração em um outro conjunto de valores de saída, o quais devem estar em memória sram. **Condições:** use SOMENTE as instruções e os modos de endereçamento estudados. Use as diretivas necessárias. Assemblar e testar o programa; analisar o arquivo *.MAP e área de memória.

SRAM	0.5 GB	0x3FFFFFFF
		0x20000000
FLASH	0.5 GB	0x1FFFFFFF
		0x00000000