PSI3441 – Arquitetura de Sistemas Embarcados

- Instruções de Desvio

Escola Politécnica da Universidade de São Paulo

Prof. Gustavo Rehder – grehder@lme.usp.br

2020

Branch (B) – Desvio incondicional

- B <Rótulo>
  - Executa um desvio do fluxo sequencial das instruções
  - Rótulo aponta para um endereço a partir do qual o programa continuará, isto é, o endereço do rótulo é copiado no Program Counter (PC)
  - O endereço do desvio deve estar próximo ($\pm$2 kb)

Branch (B) – Desvio incondicional

Exemplo:

```
        MOV R0, #0      ; coloca 0 em R0
MAIN:                   ; Rótulo MAIN
        SUB SP, SP,#4   ; Decrementa Stack Pointer
        STR R0, [SP]    ; Armazena R0 no Stack
        B FUNC          ; Desvio para função FUNC. PC →Endereço de FUNC
        MOV R1, #3      ; coloca 3 em R1

FUNC:                   ; Rótulo FUNC
        ADD R0, R0, #1  ; Soma 1 ao valor de R0 e armazena em R0
        B MAIN          ; Desvio para MAIN
```

O que acontece com esse código? Qual o valor de R1?

Stack Overflow !!
R1?

Branch and Link (BL) – Desvio incondicional com link de retorno

- BL <Rótulo>
  - Executa um desvio do fluxo sequencial das instruções
  - Rótulo aponta para um endereço a partir do qual o programa continuará, isto é, o endereço do rótulo é copiado no Program Counter (PC)
  - Antes do desvio, o endereço do PC é copiado para o Link Register (LR)
  - O endereço do desvio deve estar próximo (±16 Mb)

Branch and Exchange (Bx) – Desvio incondicional

- Bx Rn
  - Executa um desvio do fluxo sequencial das instruções
  - O registrador Rn aponta para um endereço a partir do qual o programa continuará, isto é, o endereço do Rn é copiado no Program Counter (PC)

Exemplo: BL e Bx

Exemplo:

```
        MOV R0, #0      ; coloca 0 em R0
MAIN:                   ; Rótulo MAIN
        SUB SP, SP,#4   ; Decrementa Stack Pointer
        STR R0, [SP]    ; Armazena R0 no Stack
        BL FUNC         ; LR =PC e  PC →Endereço de FUNC
        MOV R1, #3      ; coloca 3 em R1

FUNC:                   ; Rótulo FUNC
        ADD R0, R0, #1  ; Soma 1 ao valor de R0 e armazena em R0
        Bx LR           ; Desvio para MAIN
```

Erro – Falta de instrução!!

R1=3

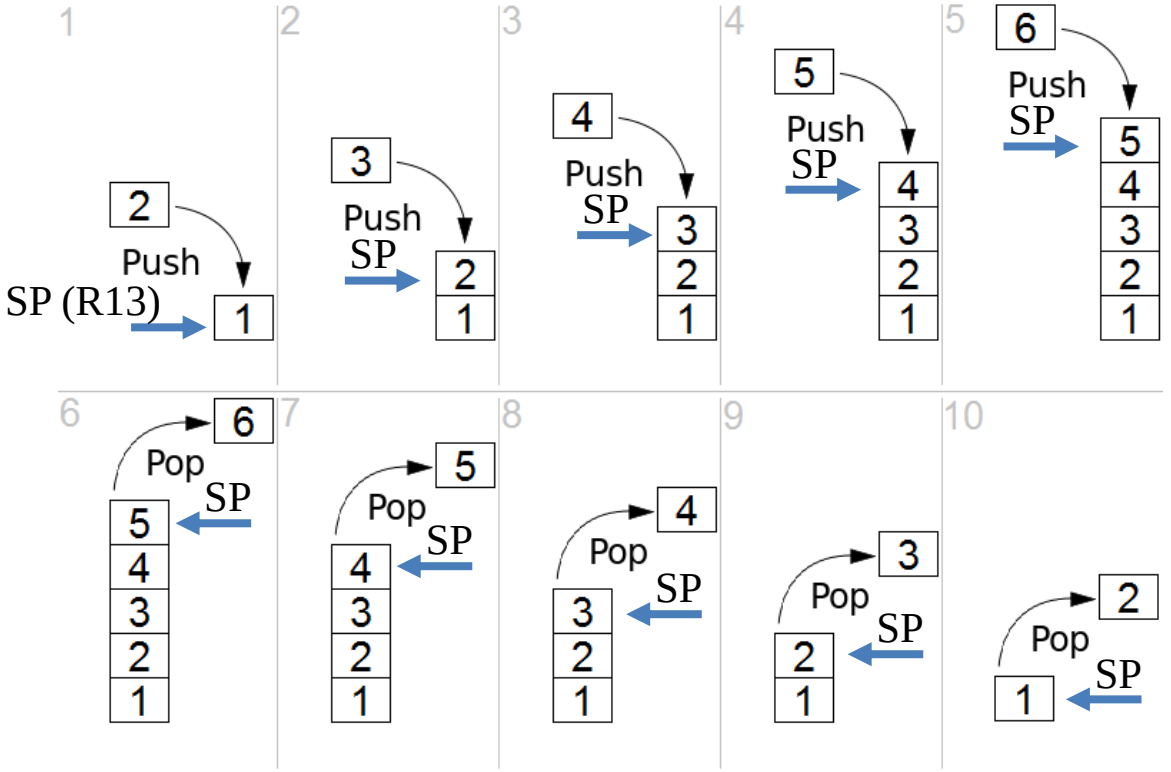
O que acontece com esse código? Qual o valor de R1?

Exemplo: BL e Bx

Exemplo:

```
        MOV R0, #0      ; coloca 0 em R0
MAIN:                   ; Rótulo MAIN
        SUB SP, SP,#4   ; Decrementa Stack Pointer
        STR R0, [SP]    ; Armazena R0 no Stack
        BL FUNC         ; LR =PC e  PC →Endereço de FUNC
        MOV R1, #3      ; coloca 3 em R1
LOOP:                   ; Rótulo LOOP
        B LOOP          ; Loop infinito
FUNC:                   ; Rótulo FUNC
        ADD R0, R0, #1  ; Soma 1 ao valor de R0 e armazena em R0
        Bx RL           ; Desvio para MAIN
```

Instruções Push e Pop



- push {R1}

```
STR R1,[R13]        ;armazena R1 no stack,
SUB R13,R13,#4      ;e decrementa Stack Pointer (R13) em 4 bytes
```

- pop {R1}

```
ADD R13,R13,#4      ; incrementa SP
LDR R1, [R13]       ;e carrega o valor do topo do stack para R1
```

Exercício

- Comente cada linha do código e mostre como mudam os registradores e o stack ao executar este programa. Otimize o programa usando PUSH e POP.

```
        LDR R13,=0xFF8
        LDR R0,=0x125
        LDR R1,=0x144
        MOV R2,#0x56
        BL FUNC
        ADD R3,R0,R1
        ADD R3,R3,R2
LOOP:
        B LOOP

FUNC:
        STR R0,[R13]
        SUB R13,R13,#4
        STR R1,[R13]
        SUB R13,R13,#4
        STR R2,[R13]
        SUB R13,R13,#4
        MOV R0,#0
        MOV R1,#0
        MOV R2,#0
        ADD R13,R13,#4
        LDR R2,[R13]
        ADD R13,R13,#4
        LDR R1,[R13]
        ADD R13,R13,#4
        LDR R0,[R13]
        BX LR
```

Auxílio

- R13 → Stack Pointer (SP)
- R14 → Link Register (LR)
- BL → Program Counter (PC) = LR antes do desvio
- Bx Rn → Copia o valor de Rn para PC
- push {R1} equivalente a:

  STR R1,[R13]

  SUB R13,R13,#4
- pop {R1} equivalente a:

  ADD R13,R13,#4

  LDR R1, [R13]

Push e Pop realmente otimizam o código?

- Quando ciclo de máquina cada instrução usa?

- http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.set.cortexm/index.html