

PSI3441 – Arquitetura de Sistemas Embarcados

- Instruções de Desvio – Loop

Escola Politécnica da Universidade de São Paulo

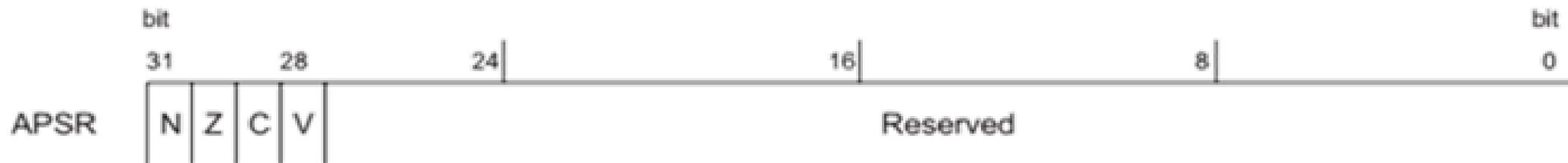
Prof. Gustavo Rehder – gprehder@usp.br





Registrador APSR - Flags

- N (Bit 31) – Flag setado → resultado Negativo
- Z (Bit 30) – Flag setado → resultado Zero ou de mesmo valor em uma comparação
- C (Bit 29) – Flag setado → “Carry” (vai 1) do resultado de soma
- V (Bit 28) – Flag setado → overflow do resultado de soma ou subtração

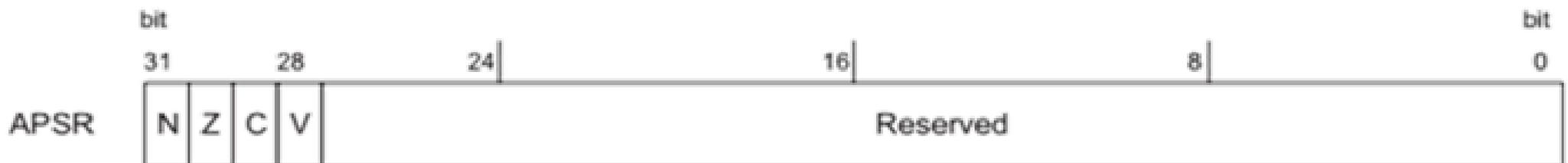




Desvio Condicional - Instruções

B{cond} <Rótulo>

- BNE <Rótulo> ; Branch if Not Equal - Desvio se $Z = 0$
- BEQ <Rótulo> ; Branch if EQual - Desvio se $Z = 1$
- BLS <Rótulo> ; Branch if Less or Same - Desvio se $Z = 1$ ou $C = 0$
- BHI <Rótulo> ; Branch if Higher- Desvio se $Z = 0$ e $C = 1$
- BCS/BHS <Rótulo> ; Branch if Carry Set/Higher or Same - Desvio se $C = 1$
- BCC/BLO <Rótulo> ; Branch if Carry Clear/LOWer- Desvio se $C = 0$





Loop *For* – BNE (Branch if Not Equal)

```
For (i=0, i<10, i++){
```

```
...
```

```
}
```

```
MOV R0, #10 ; carrega o contador
```

```
VOLTA instrução 1
```

```
instrução 2
```

```
...
```

```
SUBS R0, R0, #1 ; decrementa 1 do contador e checa se o  
; contador chegou a zero, se sim, seta o flag Z.
```

```
BNE VOLTA ; desvia ao rótulo se flag Z=0.
```

```
Z=1 continua o código
```



Desvio Condicional com Comparação

Comparação de Números

CMP Rn, Op2 ; compara os operando e muda os flags. Não tem
 ; registrador de destino, não muda os operandos.
 ; operação de subtração ($Rn - Op2$) para setar os flags

Instruction	C	Z
Rn > Op2	1	0
Rn = Op2	1	1
Rn < Op2	0	0

Instruction	Action
BCS/BHS branch if carry set/branch if higher or same	Branch if $Rn \geq Op2$
BCC/BLO branch if carry clear/branch lower	Branch if $Rn < Op2$
BEQ branch if equal	Branch if $Rn = Op2$
BNE branch if not equal	Branch if $Rn \neq Op2$
BLS branch if less or same	Branch if $Rn \leq Op2$
BHI branch if higher	Branch if $Rn > Op2$

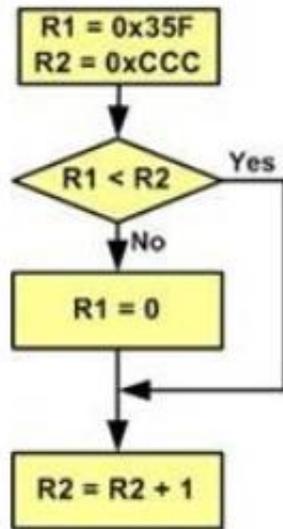


Loop *lf* – BCC (Branch if Carry Clear)

```

R1=0x35F;
R2=0xCCC;
If (R1>=R2){
R1=0;
}
R2=R2+1

```



Comparação CMP

Instruction	C	Z
Rn > Op2	1	0
Rn = Op2	1	1
Rn < Op2	0	0

```

LDR    R1,=35F
LDR    R2,=CCC
CMP    R1,R2
BCC    SE_SIM

```

; R1<R2 → C =0

; desvia para rótulo se flag C=0.

```

SE_NAO MOV    R1,#0
SE_SIM  ADD R2, R2, #1

```

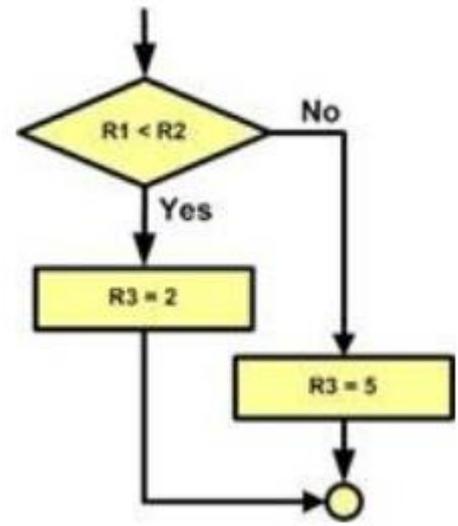


Loop *If Else* – BHS (Branch if Higher or Same) e B (Branch)

```

if(R1 < R2)
{
R3 = 2;
}
else
{
R3 = 5;
}

```



Comparação CMP

Instruction	C	Z
Rn > Op2	1	0
Rn = Op2	1	1
Rn < Op2	0	0

CMP R1,R2

BHS L1

MOV R3,#2

B FIM

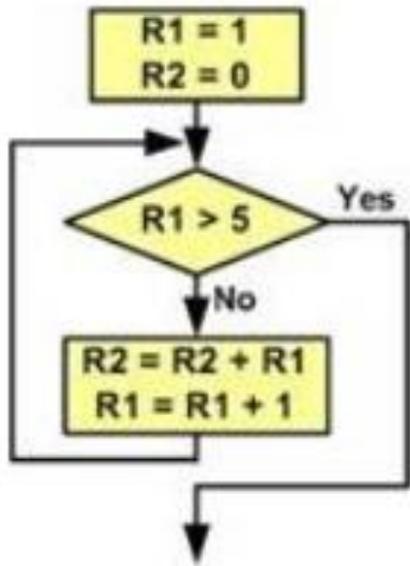
L1 MOV R3,#5

FIM

; desvio se C=1.



Loop *While* – BHI (Branch if Higher) e B (Branch)



```

int R1 = 1;
int R2 = 0;
while (R1 < 5) {
    R2 = R2 + R1;
    R1 = R1 + 1;
}
...
  
```

Comparação CMP

Instruction	C	Z
Rn > Op2	1	0
Rn = Op2	1	1
Rn < Op2	0	0

```

L1    MOV R1,#1
      MOV R2,#0
      CMP R1,#5
      BHI L2
      ADD R2,R2,R1
      ADD R1,R1,#1
      B L1
  
```

; desvia para rótulo se C=1 e Z = 0.