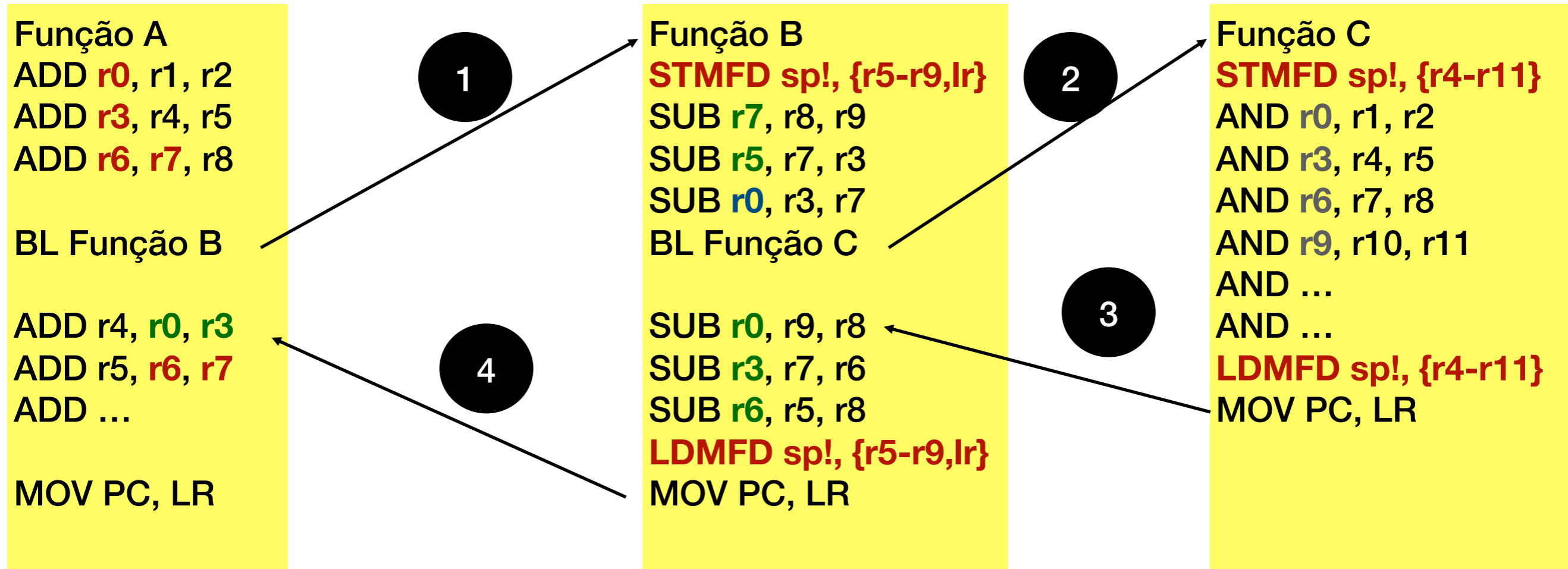


ARM

MC404

Revisando Funções e Pilha



- Você deve salvar na pilha todos os registradores entre r4 e r14 que pretende utilizar
- Se for chamar outra função e precisar preservar os registradores r0-r3, você deve salva-los também. Normalmente entre as chamadas de função

Instruções de Multiplicação

- Multiplicação convencional
 - $MUL\{\langle cond \rangle\}\{S\} rd, rm, rs \implies rd = rm * rs$
- Multiplicação e adição
 - $MLA\{\langle cond \rangle\}\{S\} rd, rm, rs, rn \implies rd = rm * rs + rn$
- rd e rm não podem ser o mesmo registrador

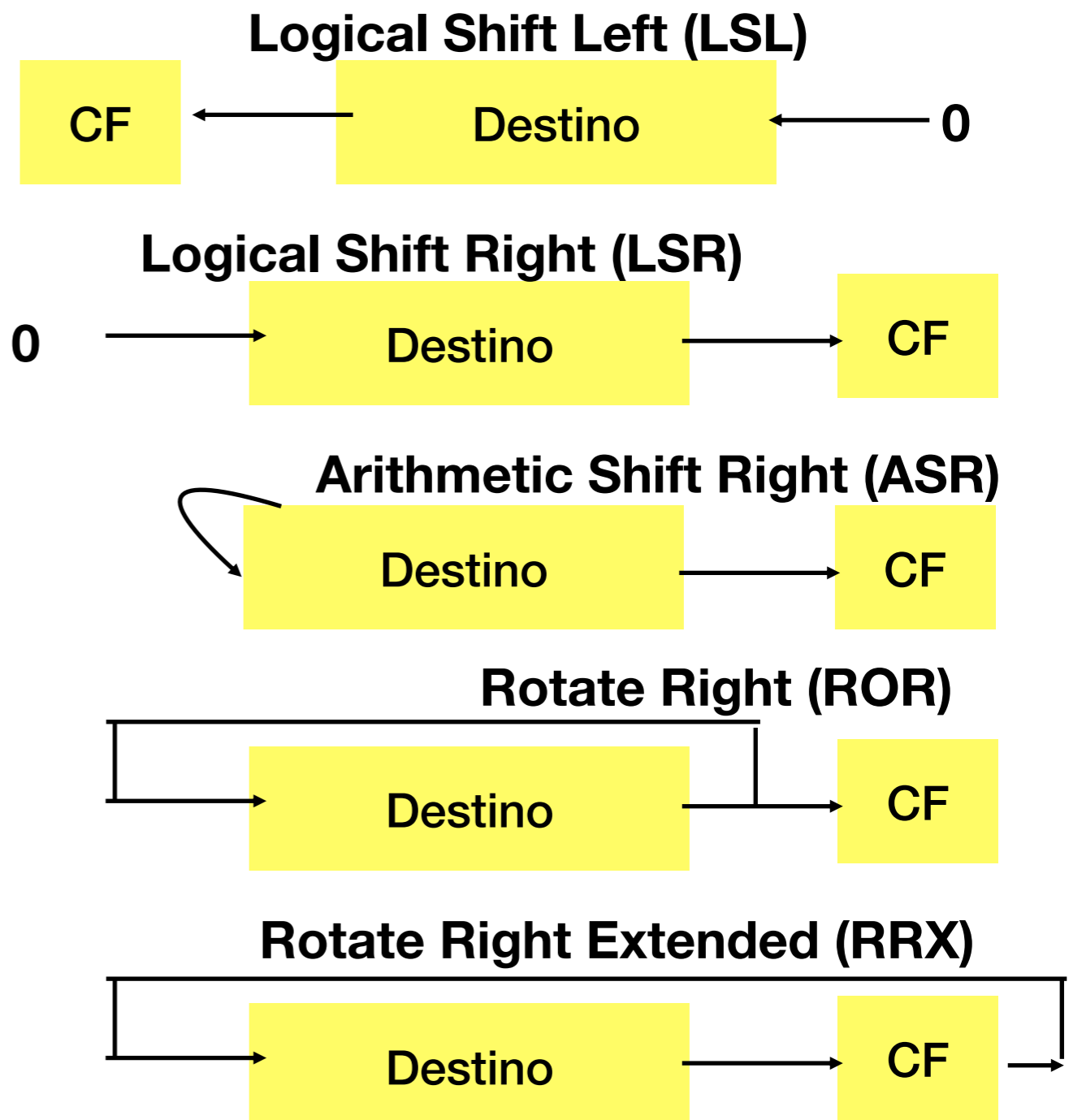
Operações de deslocamento e rotação

- As principais instruções podem receber um registrador deslocado para direita ou esquerda (operação bit a bit)

- Ex.: $17 \ll 1 = 34$

- `ADD r0, r1, r1 LSL #2`

- $r0 = 5 * r1$



Como fazer as operações abaixo sem utilizar multiplicação

- $X = 3 * Y$

- $X = 6 * Y$

- $X = 7 * Y$

- $X = 17 * Y$

- $X = 2 * Y + Y$

- $X = 4 * Y + 2 * Y$

- $X = 8 * Y - Y$

- $X = 16 * Y + Y$

Interrupções e Exceções

- Algum evento **externo** ao processador acontece e precisa de atenção de alguma rotina de código (**interrupção**)
- Algum evento **interno** ao processador acontece e precisa de atenção de alguma rotina de código (**exceção**)
- Premissas:
 - É importante salvar o contexto (não destruir a execução)
 - É importante atender o mais rapidamente possível (não sabemos quão crítica é a interrupção)
 - É importante retomar a execução rapidamente (não sabemos quão crítico é o programa sendo executado)

Modos de Operação

Modo	Origem	PSR[4:0]	Símbolo	Descrição
User	-	0x10	USR	Modo de execução normal
FIQ	FIQ	0x11	FIQ	Fast Interrupt mode
IRQ	IRQ	0x12	IRQ	Interrupt mode
Supervisor	SWI, Reset	0x13	SVC	Modo protegido para sistema operacional
Abort	Prefetch Abort, Data Abort	0x17	ABT	Modo de memória virtual ou proteção
Undefined	Instrução indefinida	0x1b	UND	Emulação por software de rotinas de hardware
System	-	0x1f	SYS	Executa rotinas privilegiadas de sistema operacional

Tipos de Operação

- FIQ é reservado para interrupções de tratamento rápido
- IRQ é usado para os demais casos
- Supervisor é para atender a chamados de sistema feitos pelo programa em execução
- Abort indica que aconteceu uma falha de memória (instrução ou dados)
- Undefined representa uma instrução não definida
- SWI é interrupção por software
 - Forma usual de realizar chamadas ao sistema operacional (abrir arquivo, encerrar programa, etc)

Prioridades das Interrupções

Evento	Prioridade	Bit I	Bit F
Reset	1	1	1
Data Abort	2	1	0
FIQ	3	1	1
IRQ	4	1	0
Pre-fetch Abort	5	1	0
SWI	6	1	-
Undefined instruction	6	1	-

Vetor de Interrupções

Interrupção/Exceção/Reset	Modo	Endereço
Reset	SVC	0x00000000
Undefined instruction	UND	0x00000004
Software interrupt (SWI)	SVC	0x00000008
Prefetch abort	ABT	0x0000000c
Data abort	ABT	0x00000010
Reserved	N/A	0x00000014
IRQ	IRQ	0x00000018
FIQ	FIQ	0x0000001c

Registadores

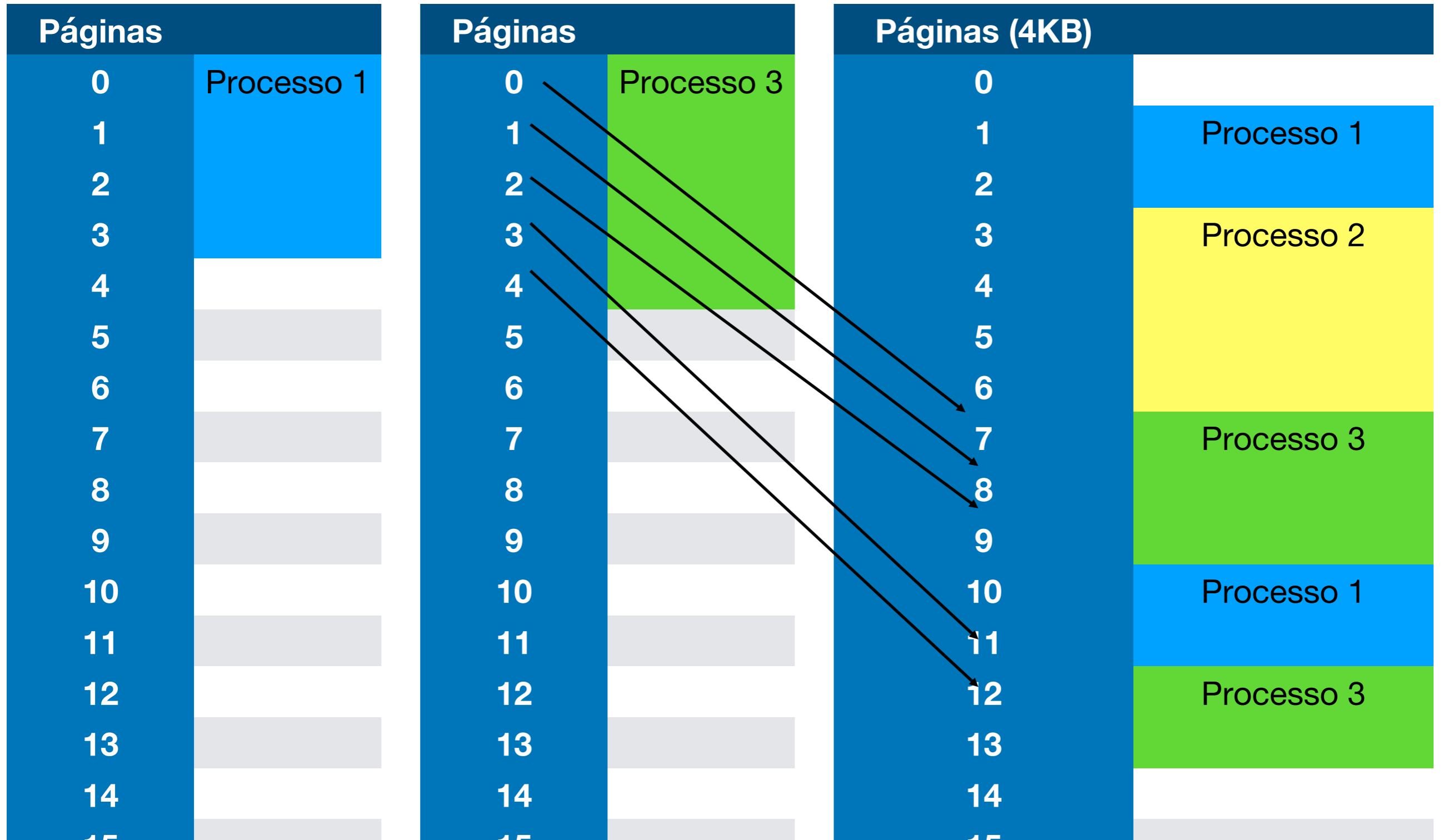
User/System	FIQ	IRQ	SVC	Undef	Abort
r0	r0	r0	r0	r0	r0
r1	r1	r1	r1	r1	r1
r2	r2	r2	r2	r2	r2
r3	r3	r3	r3	r3	r3
r4	r4	r4	r4	r4	r4
r5	r5	r5	r5	r5	r5
r6	r6	r6	r6	r6	r6
r7	r7	r7	r7	r7	r7
r8	r8_fiq	r8	r8	r8	r8
r9	r9_fiq	r9	r9	r9	r9
r10	r10_fiq	r10	r10	r10	r10
r11	r11_fiq	r11	r11	r11	r11
r12	r12_fiq	r12	r12	r12	r12
r13/SP	r13_fiq	r13_irq	r13_svc	r13_undef	r13_abort
r14/LR	r14_fiq	r14_irq	r14_svc	r14_undef	r14_abort
r15/PC	r15/PC	r15/PC	r15/PC	r15/PC	r15/PC
cpsr	-	-	-	-	-
-	spsr_fiq	spsr_irq	spsr_svc	spsr_undef	spsr_abort

Proteção de Memória

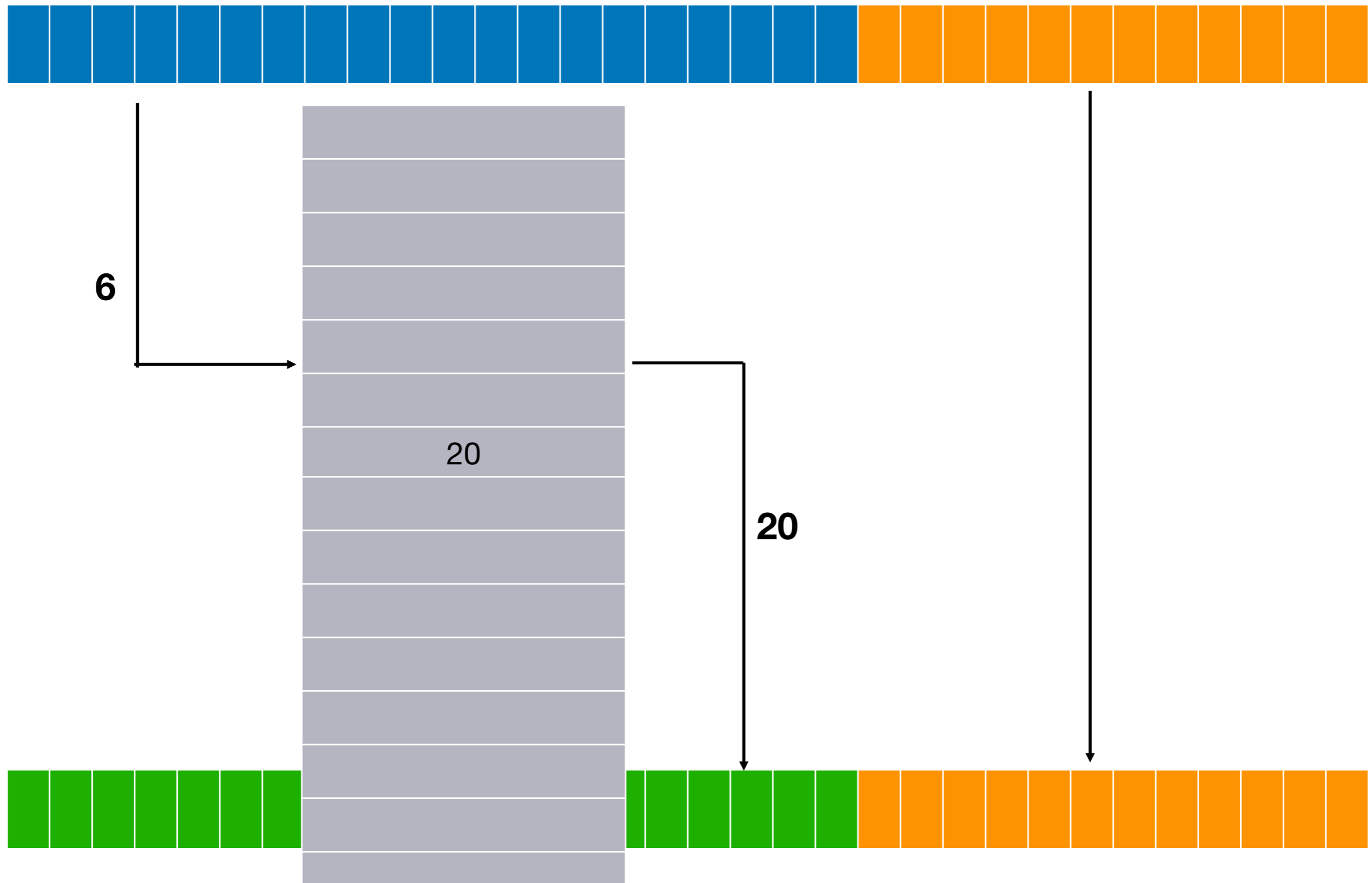
- A memória do computador é dividida em páginas gerenciadas pelo sistema operacional
- Cada página pode estar atribuída a um ou mais processos
- Bits de proteção são ajustados para controlar as permissões de acesso
- O sistema operacional mantém tabelas de página (uma por processo) para indicar quais páginas estão disponíveis

Páginas (4KB)	
0	
1	Processo 1
2	
3	Processo 2
4	
5	
6	
7	Processo 3
8	
9	
10	Processo 1
11	
12	Processo 3
13	
14	
15	

Como é feito o mapeamento de memória?



Mapeamento feito em hardware



Exercícios

1. Você não viu instrução de divisão. Implemente uma função que realize a divisão entre dois números fornecidos como parâmetros. Considere números inteiros sem sinal.
2. Implemente uma função fatorial recursivamente: $F(n) = n * F(n-1)$
3. Implemente uma função que realize a multiplicação de dois números sem utilizar a instrução de multiplicação do processador nem fazer um for com somas.
4. Dado uma matriz de 20 linhas com 10 colunas cada, contendo números inteiros, implemente uma função que localize a linha que tenha a maior soma dos elementos.

Instr	Ex.: ADD X, Y, Z
add	$X = Y + Z$
adc	$X = Y + Z + \text{carry}$
sub	$X = Y - Z$
sbc	$X = Y - Z + \text{carry} - 1$
rsb	$X = Z - Y$
rsc	$X = Z - Y + \text{carry} - 1$
cmp	$Y - Z$
cmn	$Y + Z$
tst	$Y \text{ AND } Z$
teq	$Y \text{ XOR } Z$
and	$X = Y \text{ AND } Z$
eor	$X = Y \text{ XOR } Z$
orr	$X = Y \text{ OR } Z$
bic	$X = Y \text{ AND NOT } Z$
mov	$X = Y$
mvn	$X = \text{NOT } Y$
mul	$X = Y * Z$
mla	$X = Y * Z + W$
sdiv	$X = Y / Z \text{ (c/sinal)}$
udiv	$X = Y / Z \text{ (s/sinal)}$

Cond	Descrição
EQ	igual
NE	diferente
HS / CS	\geq sem sinal
LO / CC	$<$ sem sinal
MI	negativo
PL	positivo ou zero
VS	overflow
VC	sem overflow
HI	maior sinalizado
LS	\leq sem sinal
GE	\geq
LT	$<$
GT	$>$
LE	\leq
AL	Sempre
NV	Reservado

Info	Descrição
Regist.	r0, ..., r15
r15	PC
r14	LR
r13	SP
Salto	B<cond>
Chamar função	BL<cond>
Retornar função	mov pc,lr
Ler Memória	LDR r0,[r1,#20]
Escrever Memória	STR r0,[r1,#12]
Constante	#20
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Rotate Right	ROR
Rotate Right Extended	RRX