

Zásobník

♣ Zásobník

- slouží k **automatickému** uložení návratové adresy při zavolání podprogramu nebo přerušení.
- Může být využit k dočasnému uložení proměnných.
- U ARM existují tři zásobníky:
 - o **Jednoúrovňový realizovaný** registrem **LR** (R14) – rychlý přístup k návratové adrese. Vyzvednutí může být z LR nebo přímo do PC. Instrukci CALL nahrazují skoky s uložením do LR (BL,BLX).
 - o **Hlavní zásobník**, na který ukazuje ukazatel **MSP** (defaultní SP pro operační systém a přerušení)
 - o **Uživatelský zásobník** - ukazatel **PSP** (používaný v aplikačním programu).

Při zřetězeném volání podprogramů LR $\Rightarrow$ do MSP, PSP.

OBSLUHA A POUŽITÍ ZÁSObNÍKU V JAZYCE SYMBOLICKÝCH ADRES

Mějme program volající podprogram DELAY.

```
      . . . . .
      BL DELAY      ; Volání podprogramu DELAY

      . . . . .
      ; *****
      ; * Jméno funkce      : DELAY - Softwarové zpoždění procesoru
      ; * Vstup            : R0 = počet opakování cyklu zpoždění
      ; * Ničí registry    : R3, R0
      ; *****
      DELAY          ; Navěští začátku podprogramu
                     PUSH {LR}      ; Návratová adresa do zásobníku
WAIT1  LDR R3,=20000 ; Konstanta pro prodlevu do R3
WAIT   SUBS R3,R3,#1 ; R3=R3-1,nastavení registru PSR
        BNE WAIT      ; Skok při nenulovosti R3
        SUBS R0,R0,#1
        BNE WAIT1
;Návrat z podprogramu
        POP {LR}      ; Obnovení hodnoty LR ze zásobníku
        BX LR         ; Návratová adresa do PC
;        POP {PC}      ; Náhrada předchozích dvou řádku
      ; *****
      ;
      END            ; Konec programu, další text se
                     ; nepřekládá
```

ZÁKLADNÍ ČÁSTI CENTRÁLNÍ PROCESOROVÉ JEDNOTKY

Proč potřebujeme zásobník?

```
LDR R2, =GPIOC_ODR      ; Adresy brany C ODR do R2
0x08000108  4A0A          LDR  r2,[pc,#40]  ; @0x08000134
LDR R7, =0xFFEE          ; R7 zapisovaná hodnota
0x0800010A  F64F77EE      MOVW r7,#0xFFEE
STR R7, [R2]              ; Zapis do brany C
0x0800010E  6017          STR  r7,[r2,#0x00]
LDR R0, =15               ; Načtení počtu opakování
0x08000110  F04F000F      MOV  r0,#0x0F
BL DELAY                  ; Volání DELAY
0x08000114  F000F803      BL.W 0x0800011E
0x08000118  F64F77EF      MOVW r7,#0xFFEF
0x0800011C  6017          STR  r7,[r2,#0x00]
```

DELAY

```
0x0800011E  B500          PUSH {lr}
0x08000120  B404          PUSH {r2}
0x08000122  4B05          LDR  r3,[pc,#20]  ; @0x08000138
. . .
0x0800012C  BC04          POP  {r2}
0x0800012E  F85DEB04      LDR  lr,[sp],#0x04
0x08000132  4770          BX  lr
```

PC=08000118

LR=08000118
Nebo
Zásobník

PC=0800011E

V podprogramu musí
být stejný počet
instrukcí PUSH a POP

PŘÍKLAD NA FUNKCI ZÁSOBNÍKU PŘI VOLÁNÍ PODPROGRAMU

```

0x08000108  4A0A  LDR  r2, [pc, #40] ; @0x08000134
0x0800010A  F64F77EE MOVW r7, #0xFFEE
0x0800010E  6017  STR  r7, [r2, #0x00]
0x08000110  F04F000F MOV  r0, #0x0F
0x08000114  F000F803 BL.W 0x0800011E
0x08000118  F64F77EF MOVW r7, #0xFFEF
0x0800011C  6017  STR  r7, [r2, #0x00]
DELAY
0x0800011E  B500  PUSH {lr}
0x08000120  B404  PUSH {r2}
WAIT
0x08000122  4B05  LDR  r3, [pc, #20] ;
0x08000124  1E5B  SUBS r3, r3, #1
0x08000126  D1FD  BNE  0x08000124
0x08000128  1E40  SUBS r0, r0, #1
0x0800012A  D1FA  BNE  0x08000122
0x0800012C  BC04  POP  {r2}
0x0800012E  F85DEB04 LDR  lr, [sp], #0x04
0x08000132  4770  BX   lr
0x08000134  100C  DCW  0x100C
0x08000136  4001  DCW  0x4001
0x08000138  9250  DCW  0x9250
0x0800013A  0004  DCW  0x0004
    
```

Zásobník		
SP	Adresa	Obsah
→	0x20000200	Obsazená
	0x200001FF	Nepoužitý
	0x200001FE	Nepoužitý
	0x200001FD	Nepoužitý
	0x200001FC	Nepoužitý
	0x200001FB	Nepoužitý
	0x200001FA	Nepoužitý
	0x200001F9	Nepoužitý
	0x200001F8	Nepoužitý
	0x200001F7	Nepoužitý
	0x200001F6	Nepoužitý

PŘÍKLAD NA FUNKCI ZÁSOBNÍKU PŘI VOLÁNÍ PODPROGRAMU

```

0x08000108  4A0A      LDR    r2, [pc, #40] ; @0x08000134
0x0800010A  F64F77EE  MOVW   r7, #0xFFEE
0x0800010E  6017      STR    r7, [r2, #0x00]
0x08000110  F04F000F  MOV    r0, #0x0F
0x08000114  F000F803  BL.W   0x0800011E
0x08000118  F64F77EF  MOVW   r7, #0xFFEF
0x0800011C  6017      STR    r7, [r2, #0x00]
DELAY
0x0800011E  B500      PUSH   {lr}
0x08000120  B404      PUSH   {r2}
WAIT
0x08000122  4B05      LDR    r3, [pc, #20] ;
0x08000124  1E5B      SUBS   r3, r3, #1
0x08000126  D1FD      BNE    0x08000124
0x08000128  1E40      SUBS   r0, r0, #1
0x0800012A  D1FA      BNE    0x08000122
0x0800012C  BC04      POP    {r2}
0x0800012E  F85DEB04  LDR    lr, [sp], #0x04
0x08000132  4770      BX     lr
0x08000134  100C      DCW    0x100C
0x08000136  4001      DCW    0x4001
0x08000138  9250      DCW    0x9250
0x0800013A  0004      DCW    0x0004
    
```

Zásobník		
SP	Adresa	Obsah
→	0x20000200	Obsazená
	0x200001FF	Nepoužitý
	0x200001FE	Nepoužitý
	0x200001FD	Nepoužitý
	0x200001FC	Nepoužitý
	0x200001FB	Nepoužitý
	0x200001FA	Nepoužitý
	0x200001F9	Nepoužitý
	0x200001F8	Nepoužitý
	0x200001F7	Nepoužitý
	0x200001F6	Nepoužitý

PŘÍKLAD NA FUNKCI ZÁSOBNÍKU PŘI VOLÁNÍ PODPROGRAMU

```

0x08000108  4A0A      LDR    r2, [pc, #40] ; @0x08000134
0x0800010A  F64F77EE  MOVW   r7, #0xFFEE
0x0800010E  6017      STR    r7, [r2, #0x00]
0x08000110  F04F000F  MOV    r0, #0x0F
0x08000114  F000F803  BL.W   0x0800011E
0x08000118  F64F77EF  MOVW   r7, #0xFFEF
0x0800011C  6017      STR    r7, [r2, #0x00]
DELAY
0x0800011E  B500      PUSH   {lr}
0x08000120  B404      PUSH   {r2}
WAIT
0x08000122  4B05      LDR    r3, [pc, #20] ;
0x08000124  1E5B      SUBS   r3, r3, #1
0x08000126  D1FD      BNE    0x08000124
0x08000128  1E40      SUBS   r0, r0, #1
0x0800012A  D1FA      BNE    0x08000122
0x0800012C  BC04      POP    {r2}
0x0800012E  F85DEB04  LDR    lr, [sp], #0x04
0x08000132  4770      BX     lr
0x08000134  100C      DCW    0x100C
0x08000136  4001      DCW    0x4001
0x08000138  9250      DCW    0x9250
0x0800013A  0004      DCW    0x0004
    
```

Zásobník		
SP	Adresa	Obsah
→	0x20000200	Obsazená
	0x200001FF	Nepoužitý
	0x200001FE	Nepoužitý
	0x200001FD	Nepoužitý
	0x200001FC	Nepoužitý
	0x200001FB	Nepoužitý
	0x200001FA	Nepoužitý
	0x200001F9	Nepoužitý
	0x200001F8	Nepoužitý
	0x200001F7	Nepoužitý
	0x200001F6	Nepoužitý

PŘÍKLAD NA FUNKCI ZÁSOBNÍKU PŘI VOLÁNÍ PODPROGRAMU

```

0x08000108  4A0A      LDR    r2, [pc, #40] ; @0x08000134
0x0800010A  F64F77EE  MOVW   r7, #0xFFEE
0x0800010E  6017      STR    r7, [r2, #0x00]
0x08000110  F04F000F  MOV    r0, #0x0F
0x08000114  F000F803  BL.W   0x0800011E
0x08000118  F64F77EF  MOVW   r7, #0xFFEF
0x0800011C  6017      STR    r7, [r2, #0x00]
DELAY
0x0800011E  B500      PUSH  {lr}
0x08000120  B404      PUSH  {r2}
WAIT
0x08000122  4B05      LDR    r3, [pc, #20] ;
0x08000124  1E5B      SUBS   r3, r3, #1
0x08000126  D1FD      BNE    0x08000124
0x08000128  1E40      SUBS   r0, r0, #1
0x0800012A  D1FA      BNE    0x08000122
0x0800012C  BC04      POP    {r2}
0x0800012E  F85DEB04  LDR    lr, [sp], #0x04
0x08000132  4770      BX     lr
0x08000134  100C      DCW    0x100C
0x08000136  4001      DCW    0x4001
0x08000138  9250      DCW    0x9250
0x0800013A  0004      DCW    0x0004
    
```

Zásobník		
SP	Adresa	Obsah
→	0x20000200	Obsazená
	0x200001FF	Nepoužitý
	0x200001FE	Nepoužitý
	0x200001FD	Nepoužitý
	0x200001FC	Nepoužitý
	0x200001FB	Nepoužitý
	0x200001FA	Nepoužitý
	0x200001F9	Nepoužitý
	0x200001F8	Nepoužitý
	0x200001F7	Nepoužitý
	0x200001F6	Nepoužitý

PŘÍKLAD NA FUNKCI ZÁSOBNÍKU PŘI VOLÁNÍ PODPROGRAMU

```

0x08000108  4A0A      LDR    r2, [pc, #40] ; @0x08000134
0x0800010A  F64F77EE  MOVW   r7, #0xFFEE
0x0800010E  6017      STR    r7, [r2, #0x00]
0x08000110  F04F000F  MOV    r0, #0x0F
0x08000114  F000F803  BL.W   0x0800011E
0x08000118  F64F77EF  MOVW   r7, #0xFFEF
0x0800011C  6017      STR    r7, [r2, #0x00]
DELAY
0x0800011E  B500      PUSH  {lr}
0x08000120  B404      PUSH  {r2}
WAIT
0x08000122  4B05      LDR    r3, [pc, #20] ;
0x08000124  1E5B      SUBS   r3, r3, #1
0x08000126  D1FD      BNE    0x08000124
0x08000128  1E40      SUBS   r0, r0, #1
0x0800012A  D1FA      BNE    0x08000122
0x0800012C  BC04      POP    {r2}
0x0800012E  F85DEB04  LDR    lr, [sp], #0x04
0x08000132  4770      BX     lr
0x08000134  100C      DCW    0x100C
0x08000136  4001      DCW    0x4001
0x08000138  9250      DCW    0x9250
0x0800013A  0004      DCW    0x0004
    
```

Zásobník		
SP	Adresa	Obsah
→	0x20000200	Obsazená
	0x200001FF	Nepoužitý
	0x200001FE	Nepoužitý
	0x200001FD	Nepoužitý
	0x200001FC	Nepoužitý
	0x200001FB	Nepoužitý
	0x200001FA	Nepoužitý
	0x200001F9	Nepoužitý
	0x200001F8	Nepoužitý
	0x200001F7	Nepoužitý
	0x200001F6	Nepoužitý

PŘÍKLAD NA FUNKCI ZÁSOBNÍKU PŘI VOLÁNÍ PODPROGRAMU

```

0x08000108  4A0A      LDR    r2, [pc, #40] ; @0x08000134
0x0800010A  F64F77EE  MOVW   r7, #0xFFEE
0x0800010E  6017      STR    r7, [r2, #0x00]
0x08000110  F04F000F  MOV    r0, #0x0F
0x08000114  F000F803  BL.W   0x0800011E
0x08000118  F64F77EF  MOVW   r7, #0xFFEF
0x0800011C  6017      STR    r7, [r2, #0x00]
DELAY
0x0800011E  B500      PUSH  {lr}
0x08000120  B404      PUSH  {r2}
WAIT
0x08000122  4B05      LDR    r3, [pc, #20] ;
0x08000124  1E5B      SUBS   r3, r3, #1
0x08000126  D1FD      BNE    0x08000124
0x08000128  1E40      SUBS   r0, r0, #1
0x0800012A  D1FA      BNE    0x08000122
0x0800012C  BC04      POP    {r2}
0x0800012E  F85DEB04  LDR    lr, [sp], #0x04
0x08000132  4770      BX     lr
0x08000134  100C      DCW    0x100C
0x08000136  4001      DCW    0x4001
0x08000138  9250      DCW    0x9250
0x0800013A  0004      DCW    0x0004
    
```

Zásobník		
SP	Adresa	Obsah
→	0x20000200	Obsazená
	0x200001FF	Nepoužitý
	0x200001FE	Nepoužitý
	0x200001FD	Nepoužitý
	0x200001FC	Nepoužitý
	0x200001FB	Nepoužitý
	0x200001FA	Nepoužitý
	0x200001F9	Nepoužitý
	0x200001F8	Nepoužitý
	0x200001F7	Nepoužitý
	0x200001F6	Nepoužitý

PŘÍKLAD NA FUNKCI ZÁSOBNÍKU PŘI VOLÁNÍ PODPROGRAMU

```

0x08000108  4A0A      LDR    r2, [pc, #40] ; @0x08000134
0x0800010A  F64F77EE  MOVW   r7, #0xFFEE
0x0800010E  6017      STR    r7, [r2, #0x00]
0x08000110  F04F000F  MOV    r0, #0x0F
0x08000114  F000F803  BL.W   0x0800011E
0x08000118  F64F77EF  MOVW   r7, #0xFFEF
0x0800011C  6017      STR    r7, [r2, #0x00]
DELAY
0x0800011E  B500      PUSH  {lr}
0x08000120  B404      PUSH  {r2}
WAIT
0x08000122  4B05      LDR    r3, [pc, #20] ;
0x08000124  1E5B      SUBS   r3, r3, #1
0x08000126  D1FD      BNE    0x08000124
0x08000128  1E40      SUBS   r0, r0, #1
0x0800012A  D1FA      BNE    0x08000122
0x0800012C  BC04      POP    {r2}
0x0800012E  F85DEB04  LDR    lr, [sp], #0x04
0x08000132  4770      BX     lr
0x08000134  100C      DCW    0x100C
0x08000136  4001      DCW    0x4001
0x08000138  9250      DCW    0x9250
0x0800013A  0004      DCW    0x0004
    
```

Zásobník		
SP	Adresa	Obsah
	0x20000200	Obsazená
	0x200001FF	0x08
	0x200001FE	0x00
	0x200001FD	0x01
→	0x200001FC	0x19 (0x18)
	0x200001FB	Nepoužitý
	0x200001FA	Nepoužitý
	0x200001F9	Nepoužitý
	0x200001F8	Nepoužitý
	0x200001F7	Nepoužitý
	0x200001F6	Nepoužitý

PŘÍKLAD NA FUNKCI ZÁSOBNÍKU PŘI VOLÁNÍ PODPROGRAMU

```

0x08000108  4A0A      LDR    r2, [pc, #40] ; @0x08000134
0x0800010A  F64F77EE  MOVW   r7, #0xFFEE
0x0800010E  6017      STR    r7, [r2, #0x00]
0x08000110  F04F000F  MOV    r0, #0x0F
0x08000114  F000F803  BL.W   0x0800011E
0x08000118  F64F77EF  MOVW   r7, #0xFFEF
0x0800011C  6017      STR    r7, [r2, #0x00]
DELAY
0x0800011E  B500      PUSH  {lr}
0x08000120  B404      PUSH  {r2}
WAIT
0x08000122  4B05      LDR    r3, [pc, #20] ;
0x08000124  1E5B      SUBS   r3, r3, #1
0x08000126  D1FD      BNE    0x08000124
0x08000128  1E40      SUBS   r0, r0, #1
0x0800012A  D1FA      BNE    0x08000122
0x0800012C  BC04      POP    {r2}
0x0800012E  F85DEB04  LDR    lr, [sp], #0x04
0x08000132  4770      BX     lr
0x08000134  100C      DCW    0x100C
0x08000136  4001      DCW    0x4001
0x08000138  9250      DCW    0x9250
0x0800013A  0004      DCW    0x0004
    
```

Zásobník		
SP	Adresa	Obsah
	0x20000200	Obsazená
	0x200001FF	0x08
	0x200001FE	0x00
	0x200001FD	0x01
	0x200001FC	0x19
	0x200001FB	0x40
	0x200001FA	0x01
	0x200001F9	0x10
→	0x200001F8	0x0C
	0x200001F7	Nepoužitý
	0x200001F6	Nepoužitý

PŘÍKLAD NA FUNKCI ZÁSOBNÍKU PŘI VOLÁNÍ PODPROGRAMU

```

0x08000108  4A0A      LDR    r2, [pc, #40] ; @0x08000134
0x0800010A  F64F77EE  MOVW   r7, #0xFFEE
0x0800010E  6017      STR    r7, [r2, #0x00]
0x08000110  F04F000F  MOV    r0, #0x0F
0x08000114  F000F803  BL.W   0x0800011E
0x08000118  F64F77EF  MOVW   r7, #0xFFEF
0x0800011C  6017      STR    r7, [r2, #0x00]
DELAY
0x0800011E  B500      PUSH  {lr}
0x08000120  B404      PUSH  {r2}
WAIT
0x08000122  4B05      LDR    r3, [pc, #20] ;
0x08000124  1E5B      SUBS   r3, r3, #1
0x08000126  D1FD      BNE    0x08000124
0x08000128  1E40      SUBS   r0, r0, #1
0x0800012A  D1FA      BNE    0x08000122
0x0800012C  BC04      POP    {r2}
0x0800012E  F85DEB04  LDR    lr, [sp], #0x04
0x08000132  4770      BX     lr
0x08000134  100C      DCW    0x100C
0x08000136  4001      DCW    0x4001
0x08000138  9250      DCW    0x9250
0x0800013A  0004      DCW    0x0004
    
```

Zásobník		
SP	Adresa	Obsah
	0x20000200	Obsazená
	0x200001FF	0x08
	0x200001FE	0x00
	0x200001FD	0x01
	0x200001FC	0x19
	0x200001FB	0x40
	0x200001FA	0x01
	0x200001F9	0x10
→	0x200001F8	0x0C
	0x200001F7	Nepoužitý
	0x200001F6	Nepoužitý

PŘÍKLAD NA FUNKCI ZÁSOBNÍKU PŘI VOLÁNÍ PODPROGRAMU

```

0x08000108  4A0A      LDR    r2, [pc, #40] ; @0x08000134
0x0800010A  F64F77EE  MOVW   r7, #0xFFEE
0x0800010E  6017      STR    r7, [r2, #0x00]
0x08000110  F04F000F  MOV    r0, #0x0F
0x08000114  F000F803  BL.W   0x0800011E
0x08000118  F64F77EF  MOVW   r7, #0xFFEF
0x0800011C  6017      STR    r7, [r2, #0x00]
DELAY
0x0800011E  B500      PUSH  {lr}
0x08000120  B404      PUSH  {r2}
WAIT
0x08000122  4B05      LDR    r3, [pc, #20] ;
0x08000124  1E5B      SUBS   r3, r3, #1
0x08000126  D1FD      BNE    0x08000124
0x08000128  1E40      SUBS   r0, r0, #1
0x0800012A  D1FA      BNE    0x08000122
0x0800012C  BC04      POP   {r2}
0x0800012E  F85DEB04  LDR    lr, [sp], #0x04
0x08000132  4770      BX     lr
0x08000134  100C      DCW    0x100C
0x08000136  4001      DCW    0x4001
0x08000138  9250      DCW    0x9250
0x0800013A  0004      DCW    0x0004
    
```

Zásobník		
SP	Adresa	Obsah
	0x20000200	Obsazená
	0x200001FF	0x08
	0x200001FE	0x00
	0x200001FD	0x01
	0x200001FC	0x19
	0x200001FB	0x40
	0x200001FA	0x01
	0x200001F9	0x10
→	0x200001F8	0x0C
	0x200001F7	Nepoužitý
	0x200001F6	Nepoužitý

PŘÍKLAD NA FUNKCI ZÁSOBNÍKU PŘI VOLÁNÍ PODPROGRAMU

```

0x08000108  4A0A      LDR    r2, [pc, #40] ; @0x08000134
0x0800010A  F64F77EE  MOVW   r7, #0xFFEE
0x0800010E  6017      STR    r7, [r2, #0x00]
0x08000110  F04F000F  MOV    r0, #0x0F
0x08000114  F000F803  BL.W   0x0800011E
0x08000118  F64F77EF  MOVW   r7, #0xFFEF
0x0800011C  6017      STR    r7, [r2, #0x00]
DELAY
0x0800011E  B500      PUSH   {lr}
0x08000120  B404      PUSH   {r2}
WAIT
0x08000122  4B05      LDR    r3, [pc, #20] ;
0x08000124  1E5B      SUBS   r3, r3, #1
0x08000126  D1FD      BNE    0x08000124
0x08000128  1E40      SUBS   r0, r0, #1
0x0800012A  D1FA      BNE    0x08000122
0x0800012C  BC04      POP    {r2}
0x0800012E  F85DEB04  LDR    lr, [sp], #0x04
0x08000132  4770      BX     lr
0x08000134  100C      DCW    0x100C
0x08000136  4001      DCW    0x4001
0x08000138  9250      DCW    0x9250
0x0800013A  0004      DCW    0x0004
    
```

Zásobník		
SP	Adresa	Obsah
	0x20000200	Obsazená
	0x200001FF	0x08
	0x200001FE	0x00
	0x200001FD	0x01
	0x200001FC	0x19
	0x200001FB	0x40
	0x200001FA	0x01
	0x200001F9	0x10
→	0x200001F8	0x0C
	0x200001F7	Nepoužitý
	0x200001F6	Nepoužitý

PŘÍKLAD NA FUNKCI ZÁSOBNÍKU PŘI VOLÁNÍ PODPROGRAMU

```

0x08000108  4A0A      LDR    r2, [pc, #40] ; @0x08000134
0x0800010A  F64F77EE  MOVW   r7, #0xFFEE
0x0800010E  6017      STR    r7, [r2, #0x00]
0x08000110  F04F000F  MOV    r0, #0x0F
0x08000114  F000F803  BL.W   0x0800011E
0x08000118  F64F77EF  MOVW   r7, #0xFFEF
0x0800011C  6017      STR    r7, [r2, #0x00]
DELAY
0x0800011E  B500      PUSH   {lr}
0x08000120  B404      PUSH   {r2}
WAIT
0x08000122  4B05      LDR    r3, [pc, #20] ;
0x08000124  1E5B      SUBS   r3, r3, #1
0x08000126  D1FD      BNE    0x08000124
0x08000128  1E40      SUBS   r0, r0, #1
0x0800012A  D1FA      BNE    0x08000122
0x0800012C  BC04      POP    {r2}
0x0800012E  F85DEB04  LDR    lr, [sp], #0x04
0x08000132  4770      BX     lr
0x08000134  100C      DCW    0x100C
0x08000136  4001      DCW    0x4001
0x08000138  9250      DCW    0x9250
0x0800013A  0004      DCW    0x0004
    
```

Zásobník		
SP	Adresa	Obsah
	0x20000200	Obsazená
	0x200001FF	0x08
	0x200001FE	0x00
	0x200001FD	0x01
	0x200001FC	0x19
	0x200001FB	0x40
	0x200001FA	0x01
	0x200001F9	0x10
→	0x200001F8	0x0C
	0x200001F7	Nepoužitý
	0x200001F6	Nepoužitý

PŘÍKLAD NA FUNKCI ZÁSOBNÍKU PŘI VOLÁNÍ PODPROGRAMU

```

0x08000108  4A0A      LDR    r2, [pc, #40] ; @0x08000134
0x0800010A  F64F77EE  MOVW   r7, #0xFFEE
0x0800010E  6017      STR    r7, [r2, #0x00]
0x08000110  F04F000F  MOV     r0, #0x0F
0x08000114  F000F803  BL.W    0x0800011E
0x08000118  F64F77EF  MOVW   r7, #0xFFEF
0x0800011C  6017      STR    r7, [r2, #0x00]
DELAY
0x0800011E  B500      PUSH   {lr}
0x08000120  B404      PUSH   {r2}
WAIT
0x08000122  4B05      LDR    r3, [pc, #20] ;
0x08000124  1E5B      SUBS   r3, r3, #1
0x08000126  D1FD      BNE     0x08000124
0x08000128  1E40      SUBS   r0, r0, #1
0x0800012A  D1FA      BNE     0x08000122
0x0800012C  BC04      POP    {r2}
0x0800012E  F85DEB04  LDR    lr, [sp], #0x04
0x08000132  4770      BX     lr
0x08000134  100C      DCW    0x100C
0x08000136  4001      DCW    0x4001
0x08000138  9250      DCW    0x9250
0x0800013A  0004      DCW    0x0004
    
```

Zásobník		
SP	Adresa	Obsah
	0x20000200	Obsazená
	0x200001FF	0x08
	0x200001FE	0x00
	0x200001FD	0x01
	0x200001FC	0x19
	0x200001FB	0x40
	0x200001FA	0x01
	0x200001F9	0x10
→	0x200001F8	0x0C
	0x200001F7	Nepoužitý
	0x200001F6	Nepoužitý

PŘÍKLAD NA FUNKCI ZÁSOBNÍKU PŘI VOLÁNÍ PODPROGRAMU

```

0x08000108  4A0A      LDR    r2, [pc, #40] ; @0x08000134
0x0800010A  F64F77EE  MOVW   r7, #0xFFEE
0x0800010E  6017      STR    r7, [r2, #0x00]
0x08000110  F04F000F  MOV    r0, #0x0F
0x08000114  F000F803  BL.W   0x0800011E
0x08000118  F64F77EF  MOVW   r7, #0xFFEF
0x0800011C  6017      STR    r7, [r2, #0x00]
DELAY
0x0800011E  B500      PUSH   {lr}
0x08000120  B404      PUSH   {r2}
WAIT
0x08000122  4B05      LDR    r3, [pc, #20] ;
0x08000124  1E5B      SUBS   r3, r3, #1
0x08000126  D1FD      BNE    0x08000124
0x08000128  1E40      SUBS   r0, r0, #1
0x0800012A  D1FA      BNE    0x08000122
0x0800012C  BC04      POP    {r2}
0x0800012E  F85DEB04  LDR    lr, [sp], #0x04
0x08000132  4770      BX     lr
0x08000134  100C      DCW    0x100C
0x08000136  4001      DCW    0x4001
0x08000138  9250      DCW    0x9250
0x0800013A  0004      DCW    0x0004
    
```

Zásobník		
SP	Adresa	Obsah
	0x20000200	Obsazená
	0x200001FF	0x08
	0x200001FE	0x00
	0x200001FD	0x01
→	0x200001FC	0x19
	0x200001FB	0x40
	0x200001FA	0x01
	0x200001F9	0x10
	0x200001F8	0x0C
	0x200001F7	Nepoužitý
	0x200001F6	Nepoužitý

PŘÍKLAD NA FUNKCI ZÁSOBNÍKU PŘI VOLÁNÍ PODPROGRAMU

```

0x08000108  4A0A      LDR    r2, [pc, #40] ; @0x08000134
0x0800010A  F64F77EE  MOVW   r7, #0xFFEE
0x0800010E  6017      STR    r7, [r2, #0x00]
0x08000110  F04F000F  MOV    r0, #0x0F
0x08000114  F000F803  BL.W   0x0800011E
0x08000118  F64F77EF  MOVW   r7, #0xFFEF
0x0800011C  6017      STR    r7, [r2, #0x00]
DELAY
0x0800011E  B500      PUSH  {lr}
0x08000120  B404      PUSH  {r2}
WAIT
0x08000122  4B05      LDR    r3, [pc, #20] ;
0x08000124  1E5B      SUBS   r3, r3, #1
0x08000126  D1FD      BNE    0x08000124
0x08000128  1E40      SUBS   r0, r0, #1
0x0800012A  D1FA      BNE    0x08000122
0x0800012C  BC04      POP    {r2}
0x0800012E  F85DEB04  LDR    lr, [sp], #0x04
0x08000132  4770      BX     lr
0x08000134  100C      DCW    0x100C
0x08000136  4001      DCW    0x4001
0x08000138  9250      DCW    0x9250
0x0800013A  0004      DCW    0x0004
    
```

Zásobník		
SP	Adresa	Obsah
→	0x20000200	Obsazená
	0x200001FF	0x08
	0x200001FE	0x00
	0x200001FD	0x01
	0x200001FC	0x19
	0x200001FB	0x40
	0x200001FA	0x01
	0x200001F9	0x10
	0x200001F8	0x0C
	0x200001F7	Nepoužitý
	0x200001F6	Nepoužitý

PŘÍKLAD NA FUNKCI ZÁSOBNÍKU PŘI VOLÁNÍ PODPROGRAMU

```

0x08000108  4A0A      LDR    r2, [pc, #40] ; @0x08000134
0x0800010A  F64F77EE  MOVW   r7, #0xFFEE
0x0800010E  6017      STR    r7, [r2, #0x00]
0x08000110  F04F000F  MOV    r0, #0x0F
0x08000114  F000F803  BL.W   0x0800011E
0x08000118  F64F77EF  MOVW   r7, #0xFFEF
0x0800011C  6017      STR    r7, [r2, #0x00]
DELAY
0x0800011E  B500      PUSH   {lr}
0x08000120  B404      PUSH   {r2}
WAIT
0x08000122  4B05      LDR    r3, [pc, #20] ;
0x08000124  1E5B      SUBS   r3, r3, #1
0x08000126  D1FD      BNE    0x08000124
0x08000128  1E40      SUBS   r0, r0, #1
0x0800012A  D1FA      BNE    0x08000122
0x0800012C  BC04      POP    {r2}
0x0800012E  F85DEB04  LDR    lr, [sp], #0x04
0x08000132  4770      BX     lr
0x08000134  100C      DCW    0x100C
0x08000136  4001      DCW    0x4001
0x08000138  9250      DCW    0x9250
0x0800013A  0004      DCW    0x0004
    
```

Zásobník		
SP	Adresa	Obsah
→	0x20000200	Obsazená
	0x200001FF	0x08
	0x200001FE	0x00
	0x200001FD	0x01
	0x200001FC	0x19
	0x200001FB	0x40
	0x200001FA	0x01
	0x200001F9	0x10
	0x200001F8	0x0C
	0x200001F7	Nepoužitý
	0x200001F6	Nepoužitý

PŘÍKLAD NA FUNKCI ZÁSOBNÍKU PŘI VOLÁNÍ PODPROGRAMU

```

0x08000108  4A0A      LDR    r2, [pc, #40] ; @0x08000134
0x0800010A  F64F77EE  MOVW   r7, #0xFFEE
0x0800010E  6017      STR    r7, [r2, #0x00]
0x08000110  F04F000F  MOV    r0, #0x0F
0x08000114  F000F803  BL.W   0x0800011E
0x08000118  F64F77EF  MOVW   r7, #0xFFEF
0x0800011C  6017      STR    r7, [r2, #0x00]

```

DELAY

```

0x0800011E  B500      PUSH   {lr}
0x08000120  B404      PUSH   {r2}

```

WAIT

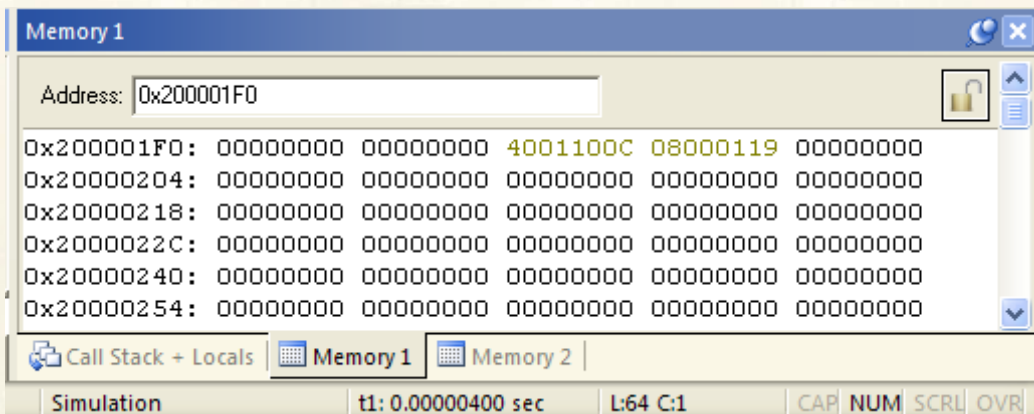
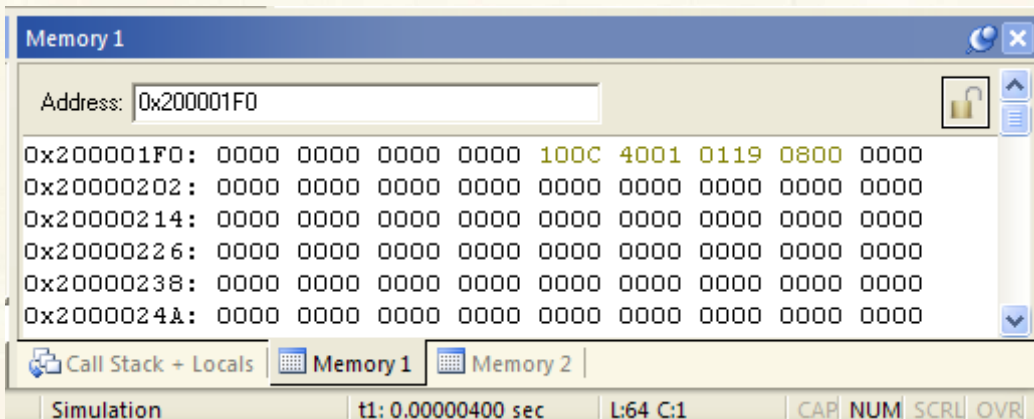
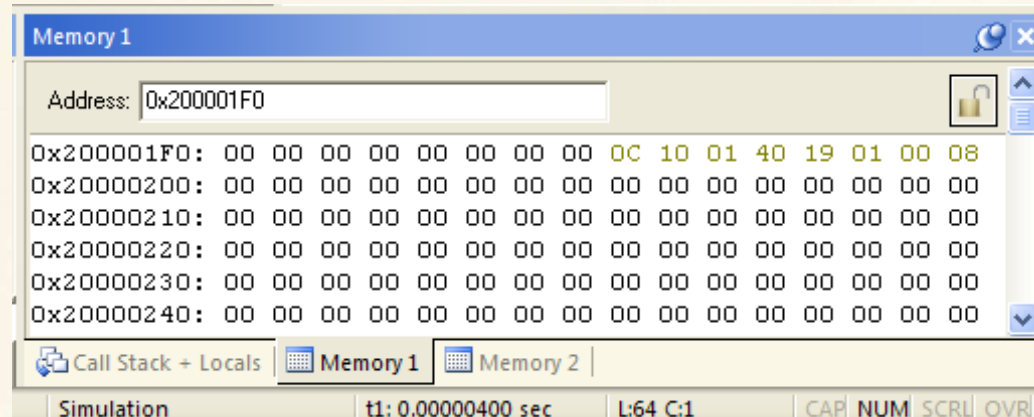
```

0x08000122  4B05      LDR    r3, [pc, #20] ;
0x08000124  1E5B      SUBS   r3, r3, #1
0x08000126  D1FD      BNE    0x08000124
0x08000128  1E40      SUBS   r0, r0, #1
0x0800012A  D1FA      BNE    0x08000122
0x0800012C  BC04      POP    {r2}
0x0800012E  F85DEB04  LDR    lr, [sp], #0x04
0x08000132  4770      BX     lr
0x08000134  100C      DCW    0x100C
0x08000136  4001      DCW    0x4001
0x08000138  9250      DCW    0x9250
0x0800013A  0004      DCW    0x0004

```

Zásobník		
SP	Adresa	Obsah
→	0x20000200	Obsazená
	0x200001FF	0x08
	0x200001FE	0x00
	0x200001FD	0x01
	0x200001FC	0x19
	0x200001FB	0x40
	0x200001FA	0x01
	0x200001F9	0x10
	0x200001F8	0x0C
	0x200001F7	Nepoužitý
	0x200001F6	Nepoužitý

ZÁKLADNÍ ČÁSTI CENTRÁLNÍ PROCESOROVÉ JEDNOTKY



0x19 místo 0x18 - U procesorů se sadou Thumb a Thumb-2 jsou adresy na instrukce odlišeny od ARM bitem **b0=1**. Pokud budeme sledovat v prostředí KEIL obsah paměti přidělené zásobníku při krokování předcházejícího příkladu, musíme dát pozor v jakém formátu se zobrazuje datová paměť tj. po bytech (Char), slovech (Short) nebo (Int). Obsahy pak vypadají následovně (Little-Endian) a mohou vést k představě o jiném umístění bytů v datové paměti.

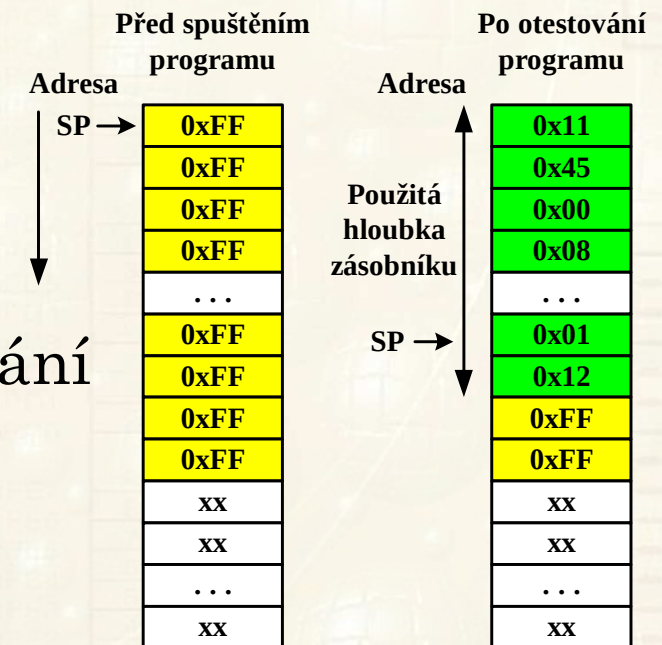
ZÁSOBNÍK PŘI VÝVOJI PROGRAMŮ

Praktické zjištění **hloubky zásobníku**, zrealizujeme v následujících krocích:

- ❖ Paměťový prostor vymezený pro zásobník vyplníme nějakou hodnotou (např. FFh).
- ❖ Spustíme program včetně všech přerušení, volání, atd.
- ❖ Po určité době zjistíme kolik hodnot zůstalo v paměti nedotčených viz obrázek.
- ❖ Situaci opakujeme i pro jiné počáteční hodnoty, aby náhodou nedošlo ke shodě s částí návratové adresy.

Umístění proměnných a zásobníku?

Prostor pro lokální proměnné (nevolání podprogramu s lokálními proměnnými uloženého ve zdrojovém kódu. Funkce `calloc()`, `malloc()`, `realloc()`, `free()`).



Formát instrukcí a dat

Operační kód instrukce – obsahuje bity určující instrukci.

Počet bitů určuje maximální možný počet instrukcí, který může i nemusí být využit. Zbývající bity instrukce určují:

Registry účastníci se instrukce , **Data** nebo **Adresu** (podprogramu, skoku nebo paměťového místa pro čtení/zápis)

Větší počet bitů instrukce → umožňuje realizaci **jednoslovných instrukcí**. Rozsah **dat** může být menší roven nebo širší, než počet bitů datové sběrnice.

Se znalostí formátu instrukcí se setkáváme zřídka – obvykle při hledání:

- ❖ Chyby v překladači
- ❖ Chyby v paměťovém prostoru (najdeme i jinak)
- ❖ Nutnosti vytvořit zpětný překladač (realizace **disassembleru**) – (z paměti, z object file) pozor na tabulky.

Uložení dat v datové paměti

- **Jedinečné** – při tvorbě programu v JSA (assembleru)
- **Big endian** – adresa přiřazená symbolickému názvu proměnné ukazuje na nejvyšší byte hodnoty proměnné. Další byty jsou na vyšších adresách.
- **Little endian** - adresa přiřazená symbolickému názvu proměnné ukazuje na nejnižší byte hodnoty proměnné. Další byty jsou na vyšších adresách.

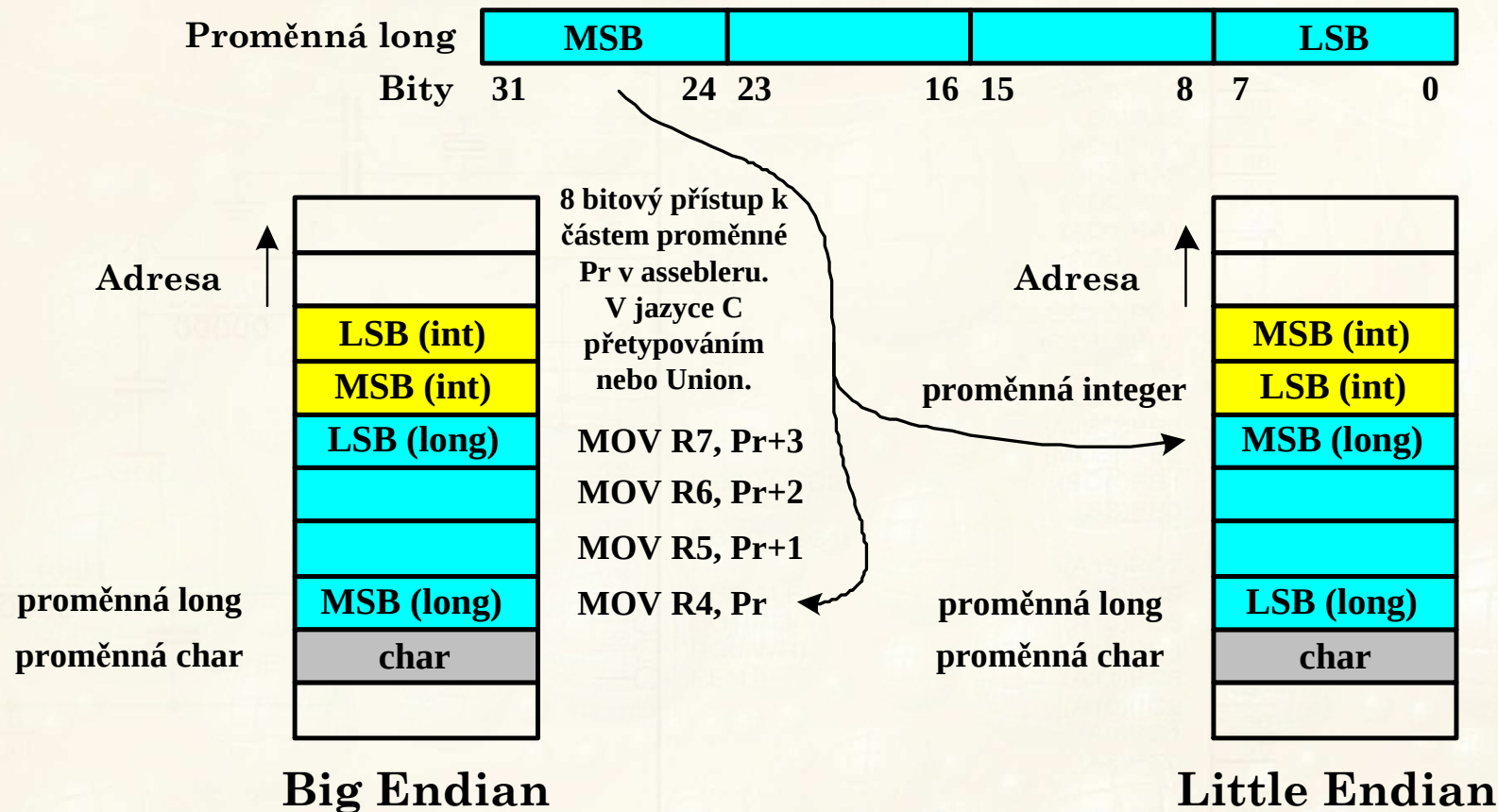
Znalost formátu → **důležitá u smíšených programů** tj. vložený assembler do jazyka C.

Práce s kratšími operandy může souviset

- S podporou instrukcí **SIMD**
- Se strukturami **pole bitů, union**, atd.

Neodůvodněná práce s kratšími operandy může vést k neefektivnímu strojovému kódu.

ULOŽENÍ DAT V DATOVÉ PAMĚTI



Ukázka nevhodné práce s krátkými operandy ARM

```
unsigned int a,b;
a=a+b;
```

```
signed short a,b;
a=a+b;
```

Může být přeložen

```
ADD r0,r0,r1
```

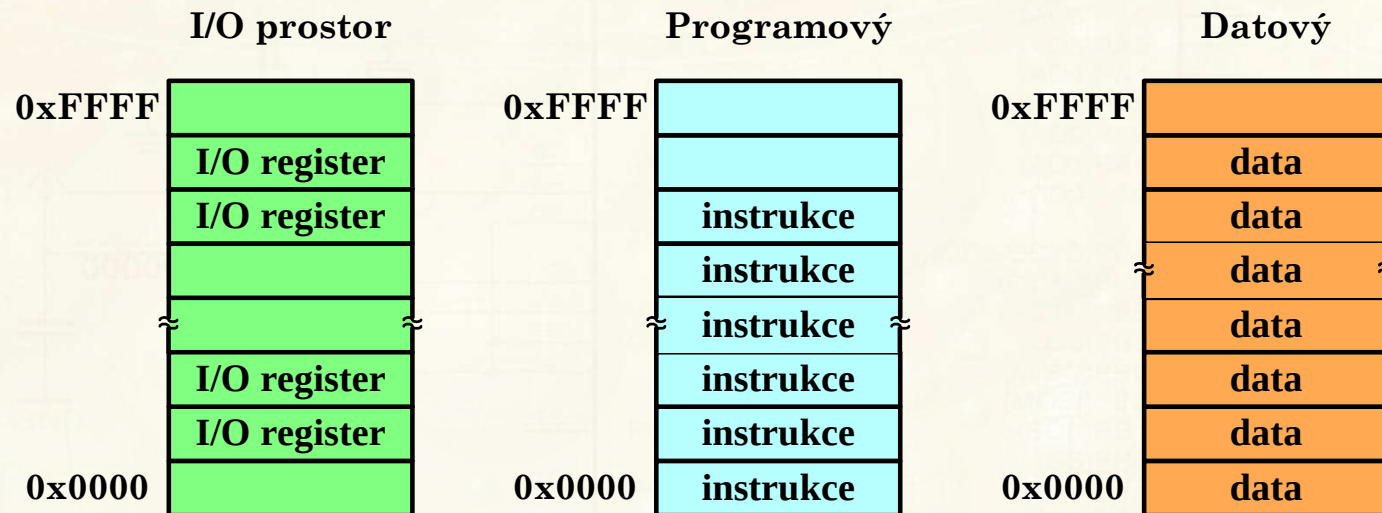
```
ADD r2,r0,r1 ;součet
```

```
LSL r2,r2,#16 ;logická rotace o 16b vlevo
```

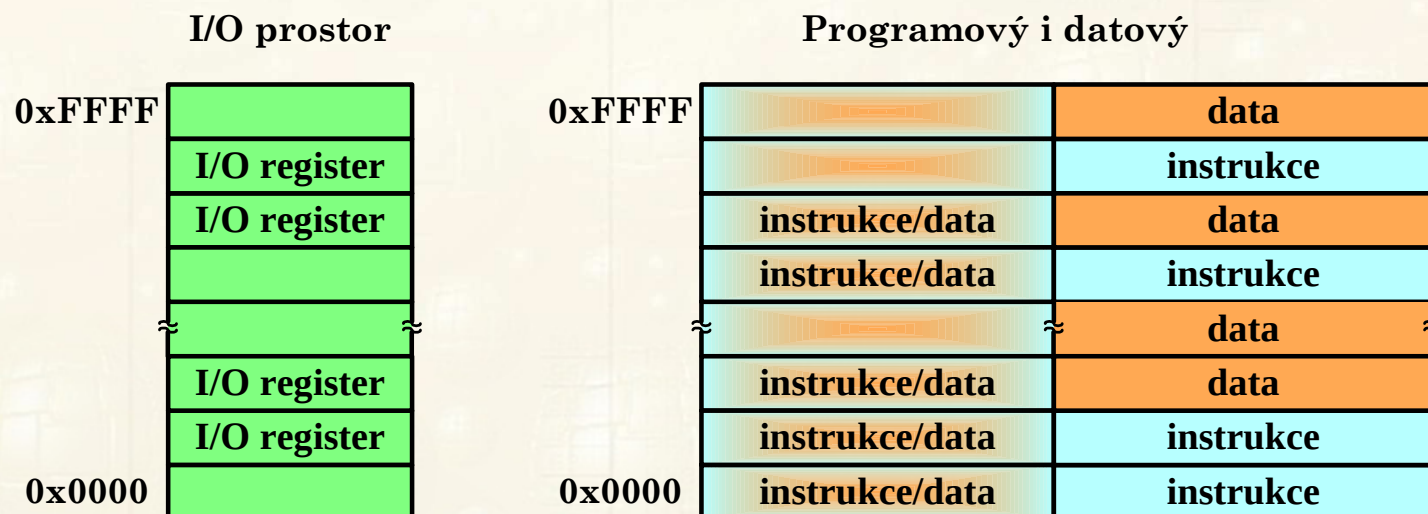
```
ASR r0,r2,#16 ;aritmetická rotace vpravo
```

ADRESOVÉ PROSTORY PROCESORU

- ❖ Programový, datový a I/O prostor – zcela samostatné prostory

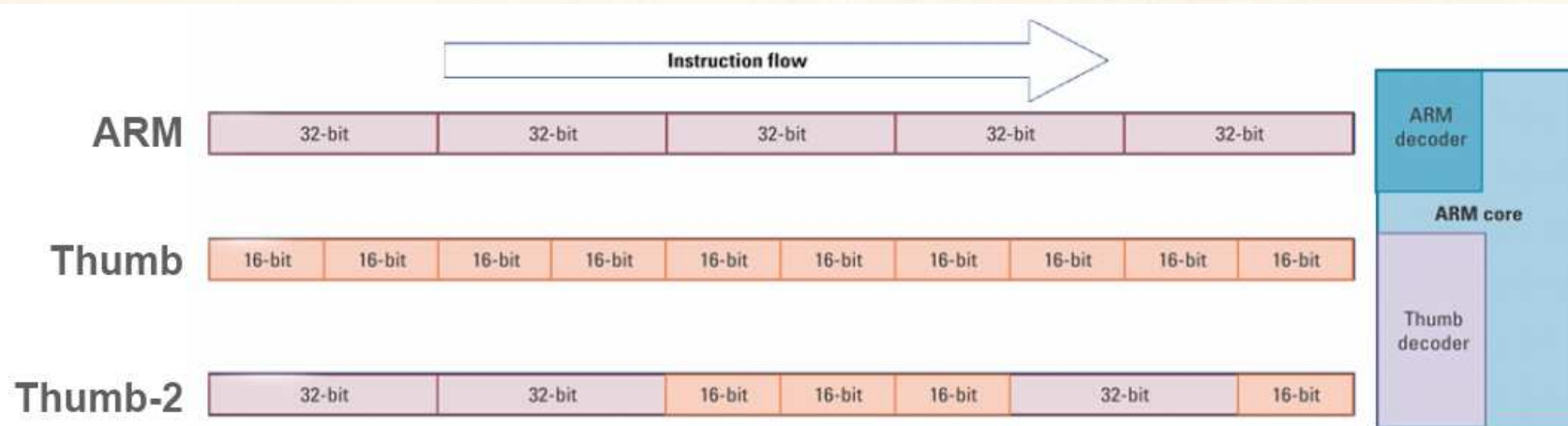
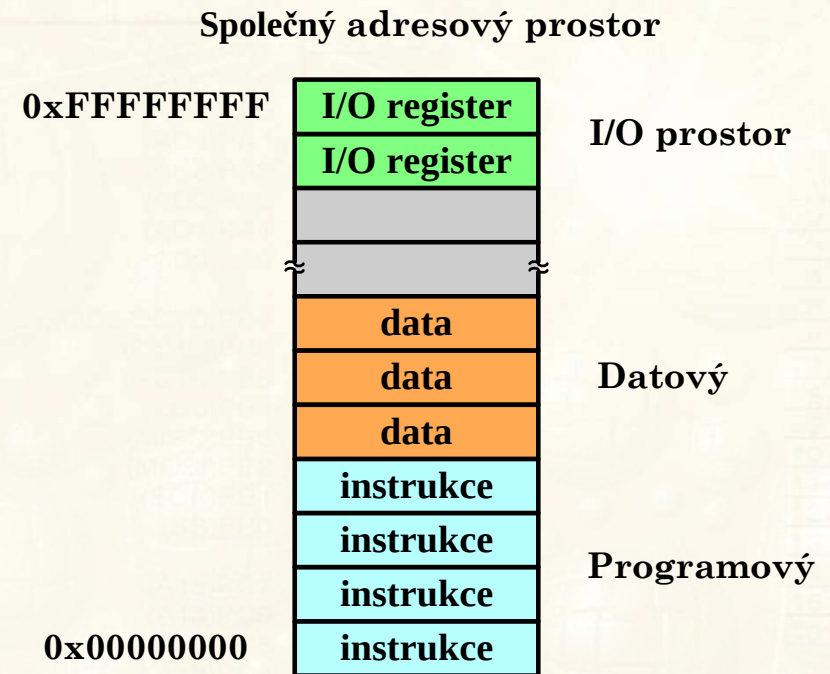


- ❖ Programový a datový prostor (operační paměť) je společný (Von Neumanovská architektura např. x86)



ADRESOVÉ PROSTORY PROCESORU S 32-BITOVOU ADRESOU

- ❖ Programový, datový a případně i vstupně-výstupní prostor leží v oddělených částech jednoho adresového prostoru. Např. ARM, 32bit. DSP
- ❖ Thumb a Thumb-2 efektivnější využití paměti programu, (též zrychlení čtení z FLASH, čtení 32 bitů – dvě instrukce najednou. Adresy instrukcí jsou proti ARM odlišeny bitem **b0=1**.



ČTENÍ PROGRAMU Z PAMĚŤOVÉHO PROSTORU

Čtení ovlivňuje nejenom počet bitů instrukce, ale také:

- Šířka programové (instrukční) sběrnice
- Schopnost procesoru číst z jednoho nebo více paměťových míst najednou

Procesor	Obsah instrukce		
8-bitový	Operační kód (se vším potřebným)		
	Operační kód		Data, Relativní adresa
	Operační kód	Část Adresy, 16 bit.dat	Část Adresy, 16 bit.dat
16-bitový	Operační kód (se vším), případně za cenu menšího adresového prostoru		
	Operační kód		16 –bitová adresa, data
32-bitový	Operační kód (se vším potřebným)		
	Dva 16-bitové operační kódy		
?	Operační kód + 32 bitová konstanta nebo adresa		

Jak načíst 32 bitovou adresu/data? Z připravené tabulky čteme instrukci s relativním posunem obvykle vůči programovému čítači (PC). Např. součet aktuální adresy PC a hodnoty 160.

0x08000130 4828 LDR r0, [pc, #160] ;@0x0800 01D4

ULOŽENÍ PROGRAMU V PAMĚŤOVÉM PROSTORU

Adresa v tabulce je vypočtena následovně:

Instrukce je uložena na adrese 0x0800 0130 a jsou čteny 4 byty, další čtená adresa (se zarovnáním na hranici 4 byte) je

$0x0800\ 0130 + 4 = 0x0800\ 0134$, posun 160 = 0xA0 (hexadecimálně)

$$\mathbf{0x0800\ 0134 + 0xA0 = 0x0800\ 01D4}$$

V protokolu se tedy objeví **@0x0800 01D4** a znak @ signalizuje, že adresa byla získána relativně.

Nevyužití všech bitů v 32 bitových instrukcích ARM $\Rightarrow$ neefektivní využití paměťového prostoru.

Vznik 16-bitových instrukcí označovaných **Thumb** $\Rightarrow$ lepší využití prostoru a zrychlení čtení.

Sada Thumb doplněna o některé 32 bitové instrukce $\Rightarrow$ sada **Thumb-2** (strojový kód není shodný ARM).

Směs 16 a 32 bitových instrukcí u JSA vede k nutnosti zarovnávat adresy na modulo 4 (align).

Stejná situace je u DSP při kruhovém adresování.