

## **11. FPU procesorů Pentium – architektura.**

# Registry FPU

Sign	79	78	64	63	0
R7	▲	Exponent	Fraction		
R6					
R5					
R4					
R3					
R2					
R1					
R0					

15	0
Tag register	
Status register	
Control register	

47	0
FPU Instruction pointer	
FPU operand (data) pointer	

10	0
Opcode	

# Relativní adresování registrů zásobníku

			Aktuální stav	Stav po uložení	Stav po výběru
R7			ST(4)	ST(5)	ST(3)
R6			ST(3)	ST(4)	ST(2)
R5			ST(2)	ST(3)	ST(1)
R4			ST(1)	ST(2)	ST(0)
R3			ST(0)	ST(1)	ST(7)
R2			ST(7)	ST(0)	ST(6)
R1			ST(6)	ST(7)	ST(5)
R0			ST(5)	ST(6)	ST(4)
Top:			011	010	100

## Registr známek (Tag register)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R7	R6	R5	R4	R3	R2	R1	R0								

Hodnota známky udává, že v příslušném registru je:

- 00 Platné nenulové číslo
- 01 Nula
- 10 Nekonečno, denormalizované číslo, nebo neplatná hodnota (nepodporovaný formát nebo NAN)
- 11 Volno – registr je prázdný

## Stavový registr (Status register)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B	C3	TOP			C2	C1	C0	ES	SF	PE	UE	OE	ZE	DE	IE

B	Aktivní činnost - pracuje (Busy)
C3...C0	Podmínkový kód (Condition Code)
TOP	Aktuální vrch zásobníku (TOP)
ES	Stav žádostí o výjimky (Error Summary Status)
SF	Chyba zásobníku (Stack Fault: Overflow/Underflow),
PE	Porušení přesnosti (Precision Exception)
UE	Podtečení (Numeric Underflow Exception)
OE	Přetečení (Numeric Overflow Exception)
ZE	Dělení nulou (Zero Divide Exception)
DE	Denormalizovaný operand (Denormalized Operand Exception)
IE	Neplatná operace (Invalid Operation Exception) <sup>5/17</sup>

SF – chyba zásobníku (přetečení / podtečení):

Při pokusu uložit data do plného zásobníku (ukazatel zásobníku ukazuje po dekrementaci na obsazený registr) dochází k přetečení (overflow  $C1 = 1$ ) zásobníku.

Při pokusu vybrat data z prázdného zásobníku (ukazatel zásobníku ukazuje po inkrementaci na prázdný registr) dochází k podtečení (underflow,  $C1 = 0$ ) zásobníku.

K podtečení zásobníku dojde rovněž v případě, kdy zdrojovým operandem je prázdný registr.

Pozor: příznak SF je tzv. *sticky* (lepivý) a lze jej vynulovat pouze instrukcemi FINIT, FSAVE a FCLEX

# Řídící registr (Control register)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			IC	RC	PC					PM	UM	OM	ZM	DM	IM

IC                      Řízení nekonečna (Infinity Control –  
   pouze pro 8087 a 80287)

RC                      Řízení zaokrouhlování (Rounding Control)

PC                      Řízení přesnosti (Precision Control)

Maskování žádostí o výjimky (0/1 – povoleno/maskováno):

PM                      Precision Exception Mask

UM                      Underflow Exception Mask

OM                      Overflow Exception Mask

ZM                      Zero Divide Exception Mask

DM                      Denormalized operand Exception Mask

IM                      Invalid operation Exception Mask

**IC = 0/1**  $\infty/\pm\infty$  (pouze u koprocetorů 8087 a 80287)

**RC = 0/1/2/3** zaokrouhlování k: nejbližšímu\_číslo/ $-\infty$ / $+\infty$ /0

Příklad zaokrouhlování:

Necht'  $a = -1.8$ ,  $b = -1.2$ ,  $c = 1.2$ ,  $d = 1.8$

zaokrouhlení	a	b	c	d
k nejbli. číslu	-2	-1	1	2
k $-\infty$	-2	-2	1	1
k $+\infty$	-1	-1	2	2
k 0	-1	-1	1	1

**PC = 0/1/2/3** přesnost mantisy (v bitech): 23/nepoužito/52/64



Příklad: necht' výsledkem operace je normalizovaná mantisa:

1.0001 0000 1000 0011 1001 0111 0110 0011

a necht'  $PC = 0$

Protože mantisa obsahuje 32 bitů musí být zaokrouhlena na 23 bitů,  
tj. buď na

1.0001 0000 1000 0011 1001 011

nebo na

1.0001 0000 1000 0011 1001 100

podle RC a znaménka čísla

## Typy operandů:

- |                            |  |
|----------------------------|--|
| 1. Word Integer            | 16 bitů, doplňkový kód   |
| 2. Short Integer           | 32 bitů, doplňkový kód   |
| 3. Long Integer            | 64 bitů, doplňkový kód   |
| 4. Packed BCD              | 80 bitů, 18 BCD číslic v devíti slabikách, nejvyšší (desátá) slabika obsahuje pouze znaménko (nejvyšší bit), ostatní její bity jsou ignorovány |
| 5. Short/Single Real       | 32 bitů, float   |
| 6. Long/Double Real        | 64 bitů, double  |
| 7. Temporary/Extended Real | 80 bitů, formát FPU, long double   |

Typy 1, 2, 5 a 6 mají výsadní postavení a mohou být použity přímo jako paměťové operandy instrukcí. Operandy ostatních typů mohou být do/z FPU přeneseny pouze instrukcemi FLD/FILD, resp. FSTP/FISTP.

# Registr zásobníku – long double

79 78	64 63	0
S	Exponent e	f <sub>63</sub> .f <sub>62</sub> ... Mantisa f ... f <sub>0</sub>

Exponent	Mantisa	Číslo	Známka
$0 < e < 32767$	$f = 1.x$	$(-1)^S 2^{(e-16383)} 1.f_{62}f_{61} \dots f_0$	0
	$f_{63}=1 !!!$		
$e = 0$	$f = 0.0$	0	1
$e = 0$	$f = 0.x, x \neq 0$	$(-1)^S 2^{(-16382)} 0.f_{62}f_{61} \dots f_0$	2
	$f_{63}=0 !!!$ (denormalizované číslo)		
$e = 32767$	$f = 1.0$	$\infty$ (nekonečno),	2
$e = 32767$	$f = 1.x, x \neq 0$	NaN (Not a Number)	2

## Příklad převodu dekadického reálného čísla do FPU

Výchozí dekadické reálné číslo: -19.625

Absolutní hodnota tohoto čísla: 19.625

Odpovídající binární znaménko: 1

Odpovídající binární číslo v pevné řádové čárce: 10011.101

Výchozí exponent:  $(16383)_{10} = (0111111111111111)_2$

Binární číslo po normalizaci: 1.0011101

Odpovídající exponent:  $(16387)_{10} = (1000000000000011)_2$

Výsledné číslo v registru zásobníku:

$(1100\ 0000\ 0000\ 0011\ 1001\ 1101\ 0000\ 0\dots)_2 =$

$(C0\ 03\ 9D\ 00\ \dots)_{16}$

## Zpětný převod

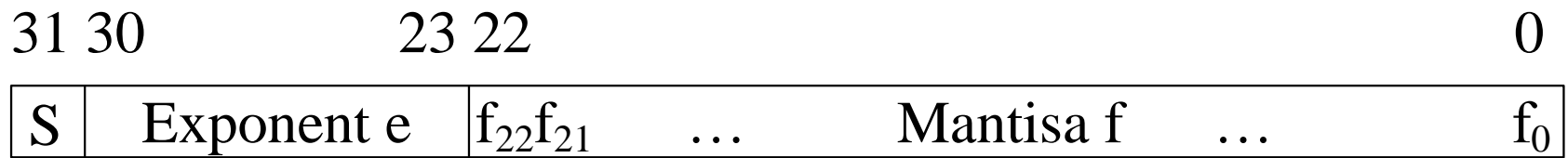
Číslo v registru zásobníku:

$$(\textcolor{blue}{1}\textcolor{green}{100}\textcolor{green}{0000}\textcolor{green}{0000}\textcolor{green}{0011}\textcolor{red}{1001}\textcolor{red}{1101}\textcolor{red}{0000}\textcolor{red}{0\dots})_2$$

Odpovídající dekadické číslo:

$$\begin{aligned} & (-1)^{\textcolor{blue}{S}} 2^{(\textcolor{green}{e}-16383)} \textcolor{red}{f}_{63} \cdot \textcolor{red}{f}_{62} \textcolor{red}{f}_{61} \dots \textcolor{red}{f}_0 = \\ & = (-1)^{\textcolor{blue}{1}} 2^{(\textcolor{green}{16387}-16383)} (\textcolor{red}{1.0011101})_2 \\ & = -1 \cdot 2^4 \cdot 1.2265625 = \\ & = -19.625 \end{aligned}$$

# float - Short/Single Real



Exponent	Mantisa	Číslo
$0 < e < 255$	$f = \text{arbitrary}$	$(-1)^S 2^{(e-127)} 1.f_{22}f_{21}\dots f_0$
$e = 0$	$f = 0$	0
$e = 0$	$f \neq 0$	$(-1)^S 2^{(-126)} 0.f_{22}f_{21}\dots f_0$
		(denormalizované číslo)
$e = 255$	$f = 0$	$\infty$ (nekonečno)
$e = 255$	$f \neq 0$	NaN (Not a Number)

## Příklad převodu reálného čísla na short real

Výchozí dekadické reálné číslo: 3586.625

Absolutní hodnota tohoto čísla: 3586.625

Odpovídající binární znaménko: 0

Odpovídající binární číslo: 111000000010.101

Výchozí exponent:  $(127)_{10} = (01111111)_2$

Binární číslo po normalizaci: 1.11000000010101

Odpovídající exponent:  $(138)_{10} = (10001010)_2$

Výsledné číslo v registru zásobníku:

$(0100\ 0101\ 0110\ 0000\ 0010\ 1010\ 0000\ 0000)_2 =$

$(45\ 60\ 2A\ 00)_{16}$

## Zpětný převod

Číslo v registru zásobníku:

$$(0100\ 0101\ 0110\ 0000\ 0010\ 1010\ 0000\ 0000)_2$$

Odpovídající dekadické číslo:

$$(-1)^S 2^{(e-127)} 1.f_{22}f_{21}\dots f_0 =$$

$$= (-1)^0 2^{(138-127)} (1.110\ 0000\ 0010\ 1010\ 0000\ 0000)_2$$

$$= 1 \cdot 2^{11} \cdot 1.75128173828125 =$$

$$= 3586.625$$



# double - Long/Double Real

63 62	52 51	0
S	Exponent e	f <sub>51</sub> f <sub>50</sub> ... Mantisa f ... f <sub>0</sub>

Exponent	Mantisa	Číslo
$0 < e < 2047$	$f = \text{arbitrary}$	$(-1)^S 2^{(e-1023)} 1.f_{51}f_{50}\dots f_0$
$e = 0$	$f = 0$	0
$e = 0$	$f \neq 0$	$(-1)^S 2^{(-1022)} 0.f_{51}f_{50}\dots f_0$ (denormalizované číslo)
$e = 2047$	$f = 0$	$\infty$ (nekonečno)
$e = 2047$	$f \neq 0$	NaN (Not a Number)