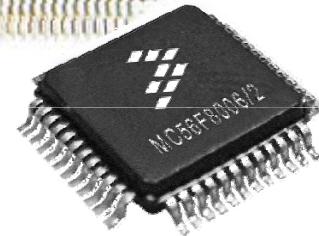




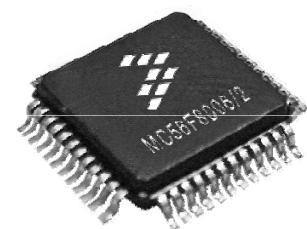
# Digitální SIGNÁLOVÉ procesory

Přehled DSP,  
základní vlastnosti, rozdělení, architektura,  
typické výpočetní jednotky, aplikace DSP



# Obsah přednášky

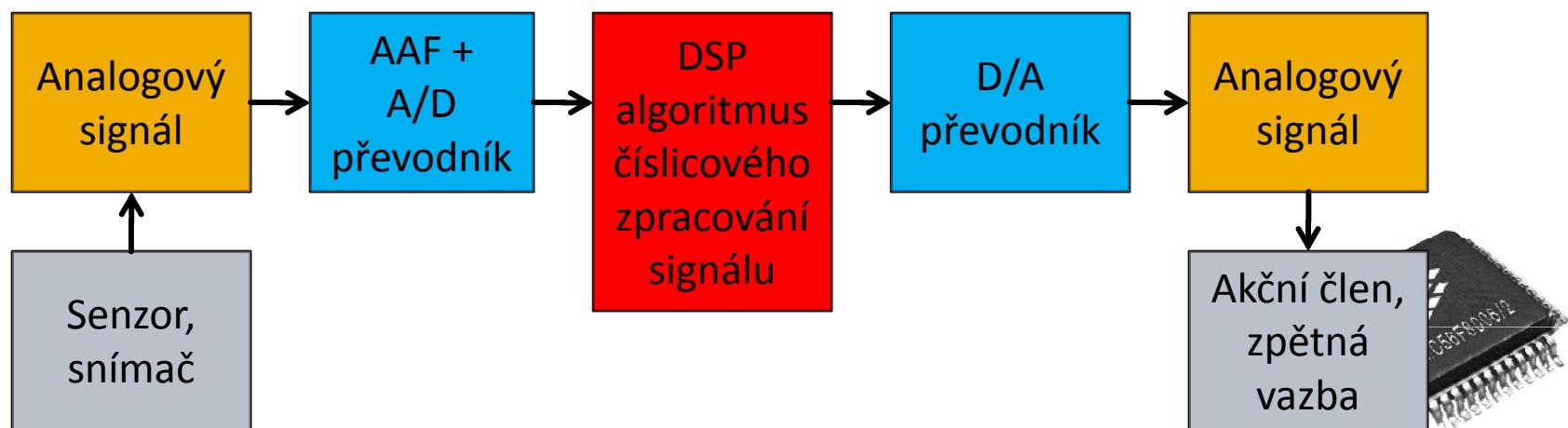
- Digitální signálový procesor (DSP) – příklady použití DSP
- Typické aplikace DSP v praxi
- Architektura DSP, hlavní rysy
- Základní rozdělení (třídění) DSP
- Zobrazení čísel na DSP – výpočty, aritmetika
- Výpočetní jednotky (ALU, MAC, SHIFT registr)
- Přední výrobci DSP - základní přiblížení výkonových řad DSP
- Jak optimalizovat výběr procesoru



# Když se řekne DSP ...

- DSP - zkratka z angl.slov **Digital Signal Processor** (digitální signálový procesor), eventuelně též velmi často užívané spojení **Digital Signal Processing** (číslicové zpracování signálu)
- DSP - procesor speciálně navržený pro číslicové zpracování digitálně reprezentovaných signálů zejména v real-time aplikacích

## Uspořádání jednoduchého číslicového systému na bázi DSP



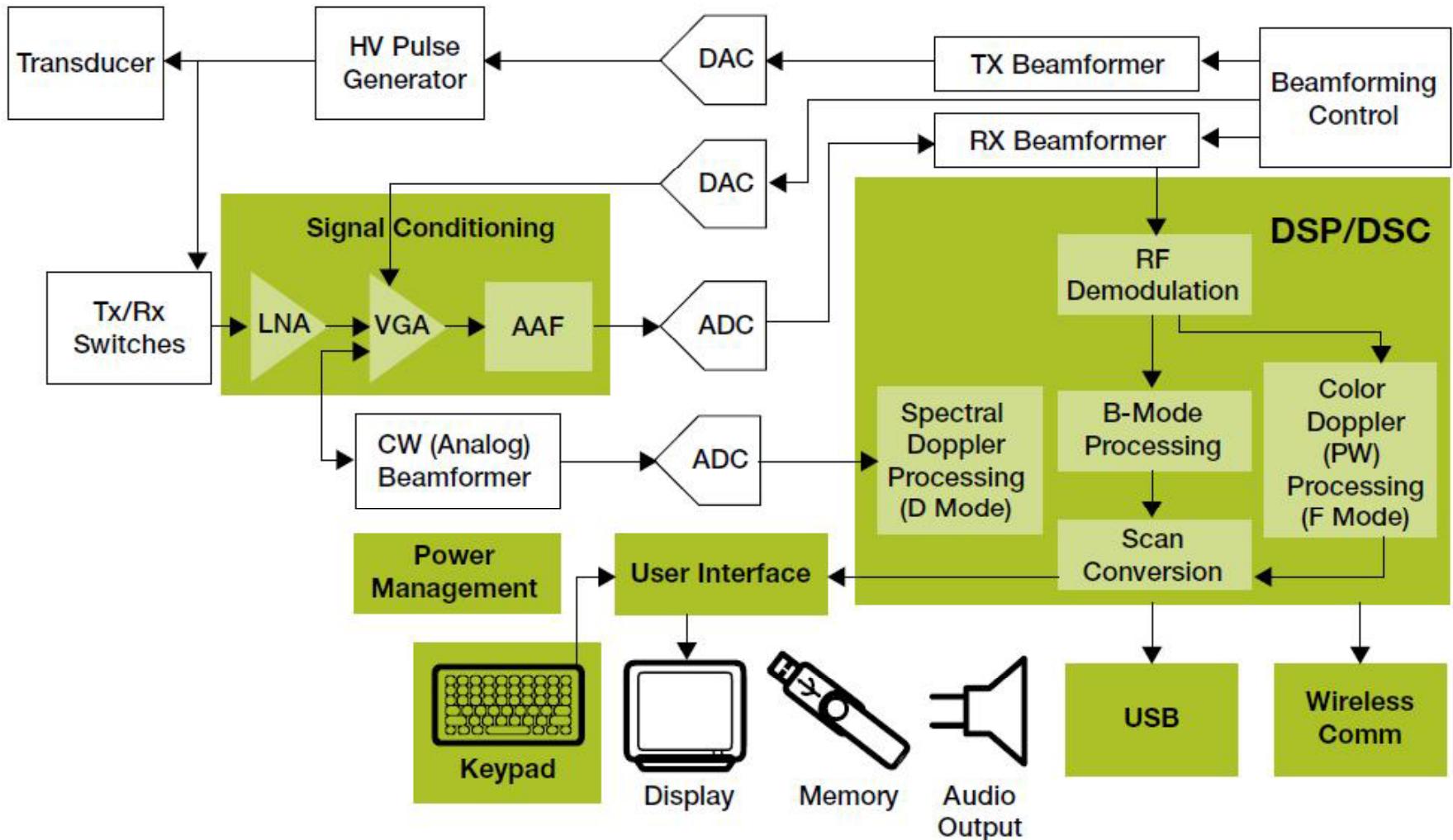
# Typické oblasti využití DSP

## Typické příklady použití DSP v praxi

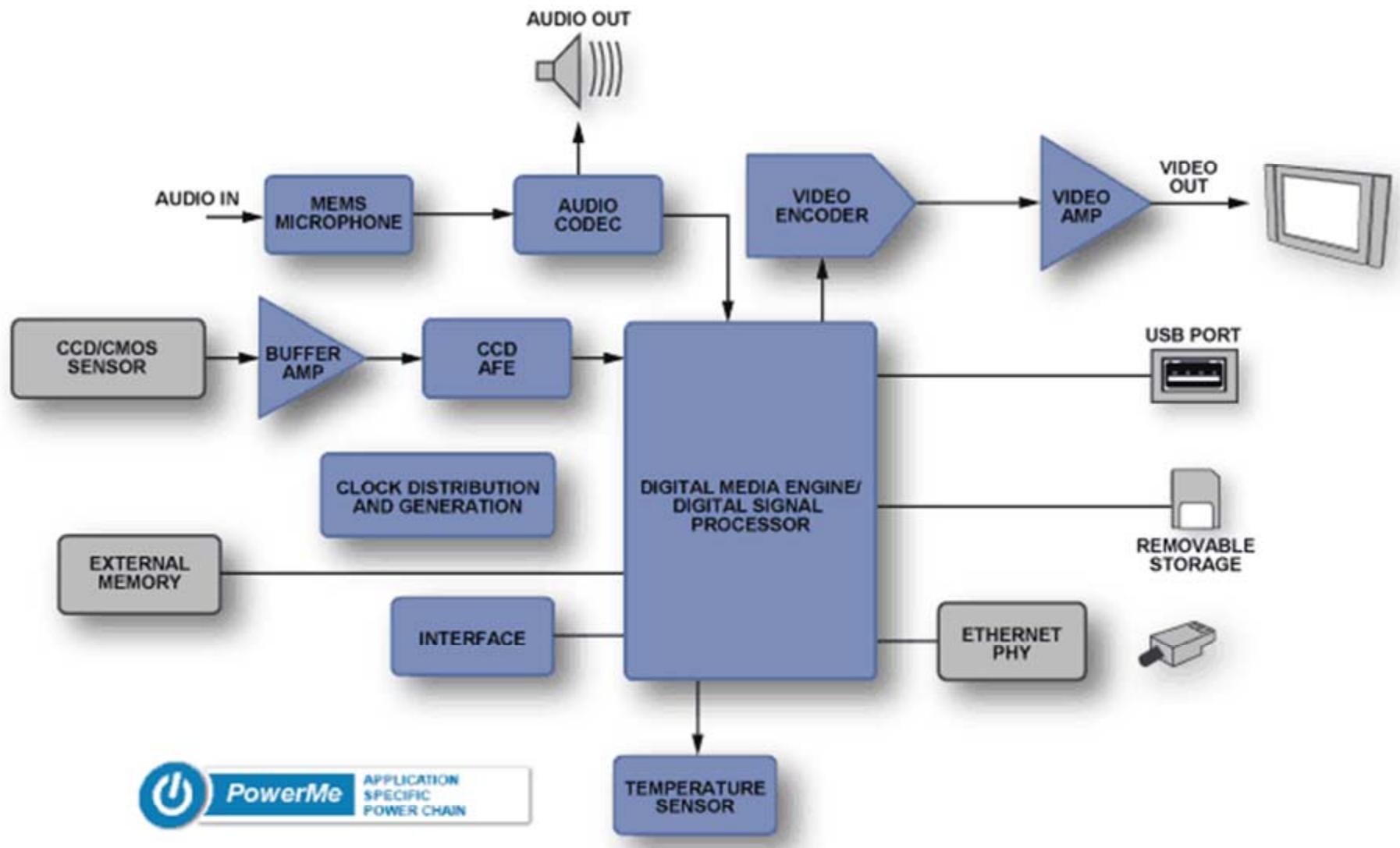
- Telekomunikace → Přenos hlasu, datová komprese, potlačení echa, filtrace, modulace, přepínání hlasových kanálů, wireless LAN
- Komerce → Zpracování obrazu a zvuku, komprese dat, speciální obrazové efekty, kamery, fotoaparáty, spotřební elektronika, mobilní telefony
- Průmysl → Monitoring a řízení procesů, nedestruktivní testování, CAD a podpůrné návrhové nástroje, aplikace v automobilovém průmyslu
- Armáda → RADAR, SONAR, zabezpečená komunikace, raketové systémy
- Vesmír → Zpracování obrazové informace např. z dalekohledů, TV, komprimace dat, analýza dat z inteligentních dálkově řízených senzorů
- Výzkum, věda → Monitoring zemětřesení & analýza a sběr dat, simulace a modelování, spektrální analýza (FFT)
- Medicína → Zobrazovací diagnostické metody (CT, MR, ultrazvuk), analýza elektrokardiogramů,



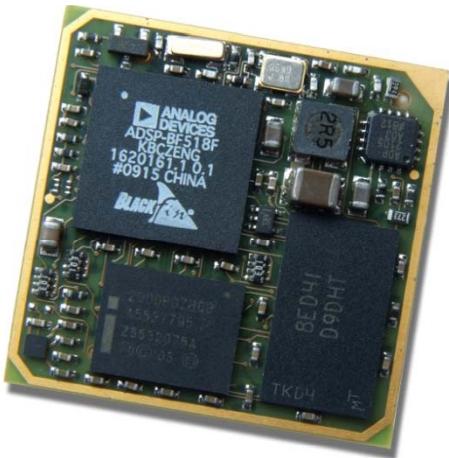
# Příklad použití DSP – ULTRAZVUK (blokové uspořádání)



# Příklad použití DSP – IP kamera (blokové uspořádání)

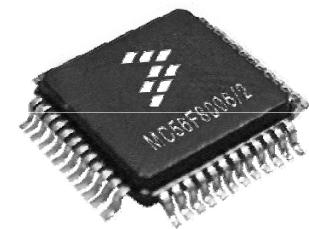


# Příklad: modul pro embedded zařízení



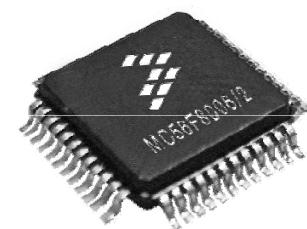
- modul pro embedded bezdrátové aplikace (**BlueTechnix**)
- použit procesor ADSP-BF518
- obsahuje napájení, 32 SDRAM a FLASH paměť
- cena okolo 230 Euro
- rozměry : 28 x 28 mm

Vývojový kit pro signálové procesory ADSP BF548



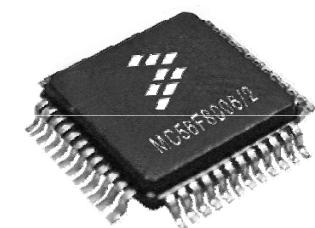
# Často používané zkratky v oblasti DSP

- ALU      **A**rithmetic / **L**ogic **U**nit
- MAC      **M**ultiplier / **A**ccumulator
- DAG      **D**ata **A**ddress **G**enerator
- MIPS      **M**illion **I**nstructions **p**er **S**econd
- MMACS      **M**illion **M**ultiply **A**ccumulate **C**ycles **p**er **S**econd
- MFLOPS      **M**illion **F**loating-**P**oint **O**perations **p**er **S**econd
- DMA      **D**irect **M**emory **A**ccess
- PCI      **P**eripheral **C**omponent **I**nterconnect
- SIMD      **S**ingle **I**nstruction **M**ultiple **D**ata
- RISC      **R**educed **I**nstruction **S**et **C**omputer



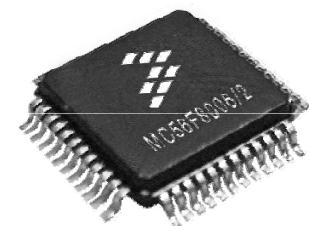
# Architektura signálových procesorů – obecné hlavní rysy

- **Harvardská architektura** - oddělená programová a datová paměť, každá paměť využívá vlastní sběrnici
- **Omezená instrukční sada (RISC jádro)**
- **Pipeline** technologie (fetch, decode, execute, access, writeback)
- **Optimalizována** pro zpracování toku dat – HW podpora:
  - ✓ kruhových bufferů (výhodné pro výpočet FIR, automatická adresace operandů)
  - ✓ bitová reverzace ( vhodné pro výpočet FFT)
- **Speciální instrukce typu SIMD** (single instruction, multiple data)
  - ✓ např.  $AX_0 = DM(I_0, M_0)$ ,  $AY_0 = PM(I_1, M_2)$ ,  $AR = AX^*AY$ ;
- **Speciální aritmeticko-logické jednotky (ALU, MAC)**
  - ✓ rychlé násobení dvou operandů s přičítáním v rámci jednoho instrukčního cyklu



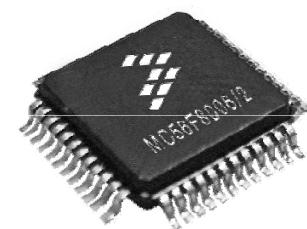
# Architektura signálových procesorů – obecné hlavní rysy

- **Bootování procesoru** – načtení programového kódu z externí paměti či periférie (externí paměti nejsou tak rychlé, aby mohly pracovat na stejné frekvenci jako DSP – opět to umožňuje vyšší výpočetní výkon)
- Obsahuje vícekanálový **řadič DMA** – lze např. využít při přenosu dat z periférie do paměti procesoru bez zatížení vlastního procesoru
- **Pro komunikaci** s okolním světem – existuje celá řada standardních komunikační rozhraní (sériový port, CAN, SPI, Ethernet, I2C, atd. - záleží na zvoleném typu DSP)
- **Výkonné DSP** nabízejí podporu např. pro **SDRAM**, nebo **HOST rozhranní** pro integraci do víceprocesorového systému
- Obsahuje-li DSP např. HW násobičky 16x16b, lze jej fiktivně rozdělit na dvě pro násobení dvou čísel s polovičním rozlišením (2x 8x8b)



# Základní rozdělení (třídění) DSP

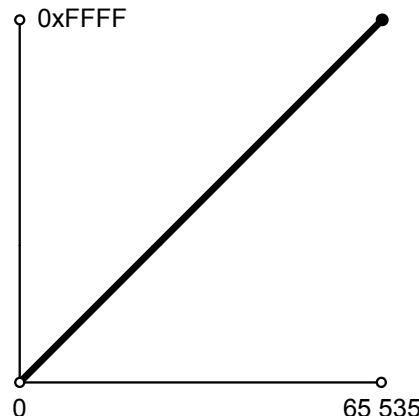
- Podle použité aritmetiky
  - ✓ aritmetika s pevnou řádovou čárkou (fixed-point)
  - ✓ aritmetika s pohyblivou řádovou čárkou (floating-point)  
norma **IEEE 754**)
- Podle šířky datové sběrnice
  - ✓ 16-bitové (typicky)
  - ✓ 24-bitové
  - ✓ 32-bitové
- Podle počtu jader procesoru
  - ✓ jednojádrové
  - ✓ vícejádrové (2,4)



# Zobrazení čísel v DSP: fixed-point formát (16-bit přesnost)

## ■ Celé číslo bez znaménka (unsigned)

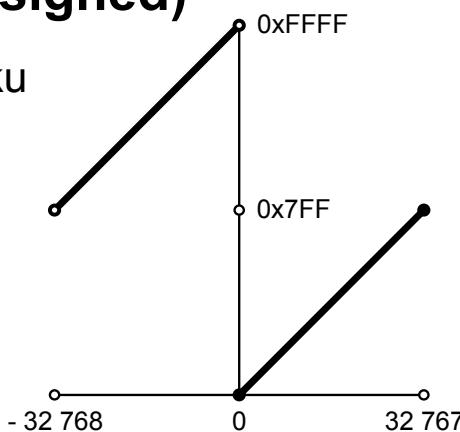
- ✓ Rozsah zobrazovaných čísel : 0 – 65535



| Číslo  | Dekadický ekvivalent |
|--------|----------------------|
| 0x0000 | 0                    |
| 0x0001 | 1                    |
| 0x7FFF | 32767                |
| 0x8000 | 32 768               |
| 0xFFFF | 65 535               |

## ■ Celé číslo se znaménkem (signed)

- ✓ Čísla vyjádřené v dvojkovém doplňku
- ✓ Poloviční rozsah než unsigned :  
 $-2^{31}$  až  $2^{31}-1$

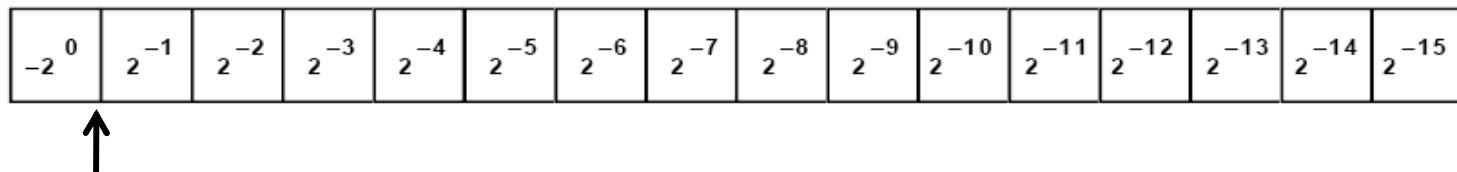


| Číslo  | Dekadický ekvivalent |
|--------|----------------------|
| 0x0000 | 0                    |
| 0x0001 | 1                    |
| 0x7FFF | 32767                |
| 0x8000 | -32768               |
| 0xFFFF | -1                   |

# Zobrazení čísel v DSP: fixed-point formát (16-bit přesnost)

## ■ 16-bit desetinné číslo se znaménkem (fractional signed 1.15)

- ✓ Doporučovaný a zároveň nejvíce používaný formát při výpočtech
- ✓ Čísla vyjádřené v dvojkovém doplňku



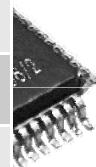
Pozice desetinné čárky

Formát 1.15 (přesnost 16 bitů)

| Číslo  | Dekadický ekvivalent |
|--------|----------------------|
| 0x0001 | 0.000 031            |
| 0x7FFF | 0.999 969            |
| 0xFFFF | -0.000 031           |
| 0x8000 | -1.000 000           |

Formát 1.31 (přesnost 32 bitů)

| Číslo       | Dekadický ekvivalent |
|-------------|----------------------|
| 0x0000 0001 | 0.000 000 000 465    |
| 0x7FFF FFFF | 0.999 999 999 534    |
| 0xFFFF FFFF | -0.000 000 000 465   |
| 0x8000 0000 | -1.000 000 000 000   |

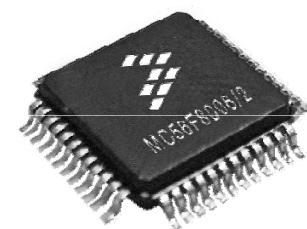
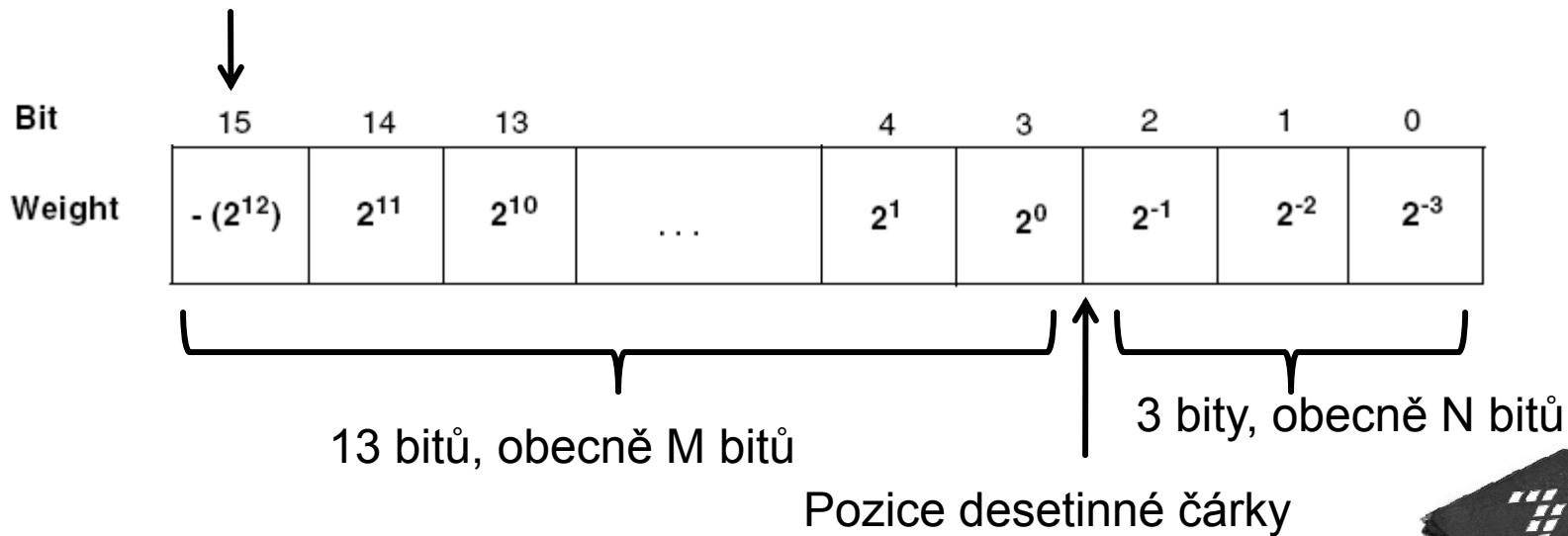


# Zobrazení čísel v DSP: obecný fixed-point (16-bit přesnost)

- Obecný formát 16-bit desetinného čísla se znaménkem (fractional signed M.N )

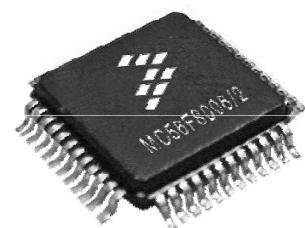
## Příklad : obecný formát fractional 13.3

Je-li MSB znamínkový bit, pak se jedná o **signed fractional**



# Desetinné formáty čísel a jejich rozsah (16-bit přesnost)

| Formát | M  | N  | Max. kladné číslo dekadicky | Max. záporné číslo dekadicky | Hodnota 1 LSB dekadicky |
|--------|----|----|-----------------------------|------------------------------|-------------------------|
| 1.15   | 1  | 15 | 0,999969482421875           | -1                           | 0,000030517578125       |
| 2.14   | 2  | 14 | 1,99993896484375            | -2                           | 0,00006103515625        |
| 3.13   | 3  | 13 | 3,9998779296875             | -4                           | 0,0001220703125         |
| 4.12   | 4  | 12 | 7,999755859375              | -8                           | 0,000244140625          |
| 5.11   | 5  | 11 | 15,99951171875              | -16                          | 0,00048828125           |
| 6.1    | 6  | 10 | 31,9990234375               | -32                          | 0,0009765625            |
| 7.9    | 7  | 9  | 63,998046875                | -64                          | 0,001953125             |
| 8.8    | 8  | 8  | 127,99609375                | -128                         | 0,00390625              |
| 9.7    | 9  | 7  | 255,9921875                 | -256                         | 0,0078125               |
| 10.6   | 10 | 6  | 511,984375                  | -512                         | 0,015625                |
| 11.5   | 11 | 5  | 1 023,96875                 | -1 024                       | 0,03125                 |
| 12.4   | 12 | 4  | 2 047,9375                  | -2 048                       | 0,0625                  |
| 13.3   | 13 | 3  | 4 095,875                   | -4 096                       | 0,125                   |
| 14.2   | 14 | 2  | 8 191,75                    | -8 192                       | 0,25                    |
| 15.1   | 15 | 1  | 16 383,5                    | -16 384                      | 0,5                     |
| 16.0   | 16 | 0  | 32 767                      | -32 768                      | 1                       |



# Zobrazení čísel v DSP: floating-point formát

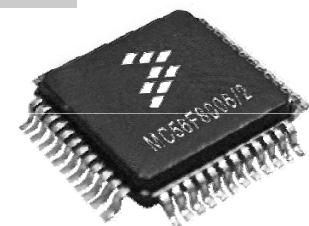
- Využívá se definic dle standardu **IEEE 754**
- Existují (stejně jako na PC platformě v C nebo C++) formáty:
  - ✓ **single precision** 32 bit (viz. tabulka),
  - ✓ **double precision** 64 bit (1S + 11E + 52M)
  - ✓ **extended** 80 bit (1S + 15E + 64M)
  - ✓ **quadruple** 128 bit. (1S + 15E + 112M)
- Normalizované číslo: nejvyšší bit mantisy nastaven na log. 1

**Příklad:** single precision

| bit    | 31 | 30 29 ... 24 23    | 22 21 ... 2 1 0   |
|--------|----|--------------------|-------------------|
| význam | S  | Exponent ( 8 bitů) | Mantisa (23 bitů) |

Vztah pro přepočet :

$$\text{hodnota}_{\text{single\_float}} = (-1)^{\text{S}} \times 2^{\text{E}-127} \times (1.\text{M})$$



# Zobrazení čísel v DSP: rozsah floating-point formátů

| Typ                 | Digity | Emin   | Emax   | Dekadické<br>digity |  |  |
|---------------------|--------|--------|--------|---------------------|--|--|
| Single precision    | 23+1*  | -126   | 127    |                     |  |  |
| Double precision    | 52+1*  | -1022  | +1023  |                     |  |  |
| Extended            | 64+1*  | -16382 | +16383 |                     |  |  |
| Quadruple precision | 112+1* | -16382 | +16383 |                     |  |  |

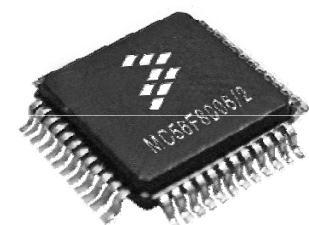
\*normalizované číslo = nejvyšší bit mantisy =1, nekóduje se do formátu

## Příklad :

32-bit. číslo : 1 **1000 0001** **00110000 00000000 00000000** = - 4,75

$$81_{\text{H}} = 129_{\text{D}} \rightarrow E=2$$

$$1.0011b * 4 = 4,75_{\text{D}}$$



# Operace s floating point čísla

## ■ Sčítání, odčítání

Příklad:  $123456.7 = 1.234567 \times 10^5$

$101.7654 = 1.017654 \times 10^2 = 0.001017654 \times 10^5$

$E=5; M=1.234567$

+  $E=5; M=0.001017\ 654$  (po operaci shift)

---

$E=5; S=1.235584\ 654$  (výsledek)

$E=5; S=1.235585$  (po zaokrouhlení a normalizaci)

## ■ Násobení, dělení

Příklad:  $E=3; M=4.734612$

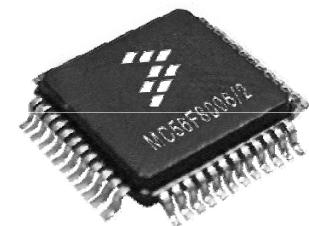
$\times E=5; M=5.417242$

---

$E=8; M=25.648538980104$  (výsledek)

$E=8; M=25.64854$  (po zaokrouhlení)

$E=9; M=2.564854$  (po normalizaci)

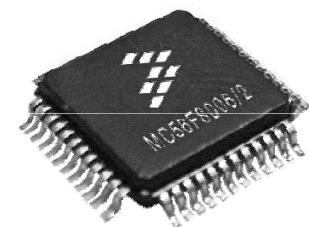


# Vícenásobné sčítání

**N-násobné sčítání M-bitových čísel vyžaduje  $M + \log_2 N$  bitů**

**Příklad:** výpočet FIR filtru 256 řádu, tj. realizace 256 násobení a 256 sčítání

- Při šířce datových vzorků 16 bitů a šířce koeficientů filtru 16 bitů – každý dílčí výsledek má 32 bitů
- Při akumulaci 256 těchto dílčích výsledků je potřeba mít 40 bitový akumulátor, jinak dojde k přetečení, tj.  $\log_2(256) = 8$



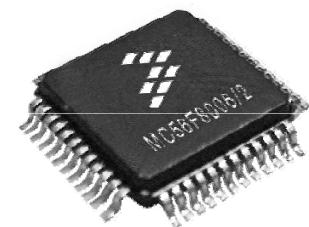
# Kvantování koeficientů

- Vypočtené koeficienty filtru přirozeně musí být kvantována na rozlišení spojené s délkou slova

$$b_{i\_kvant} = \text{round}(b_i \cdot 2^b) / 2^b$$

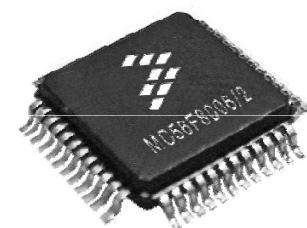
- Nechť  $b_i$  jsou koeficienty filtru, pak  $\mathbf{b}$  je počet bitů optimální pro dosažení minimální kvantovací chyby pro max. pozitivní číslo

$$b = \log_2 \left( \frac{2^{M-1} - 1}{\max|b_i|} \right)$$

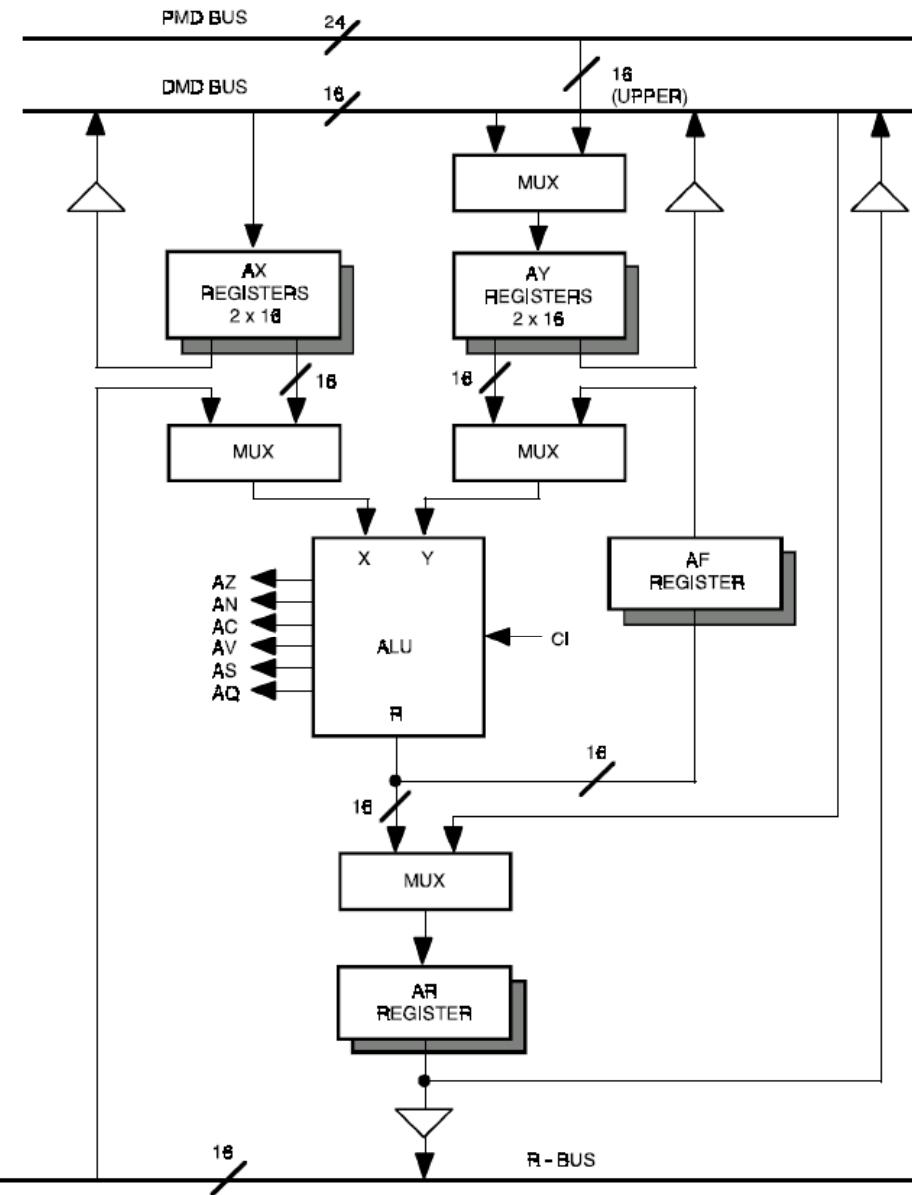


# Aritmeticko-logická jednotka ( ALU )

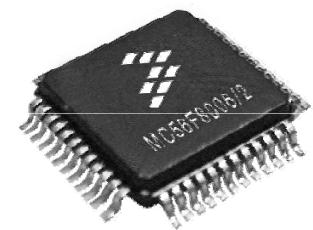
- Zajišťuje následující matematické operace:
  - ✓ Sčítání
  - ✓ Odčítání
  - ✓ Dělení
  - ✓ Negaci
  - ✓ Inkrementaci
  - ✓ Dekrementaci
  - ✓ Absolutní hodnotu
  - ✓ Logické funkce AND, OR, XOR a NOT



# Příklad: ALU v ADSP 21XX



- 16-bit. operace nad vstupy X,Y nebo AR registrem
- Registry: AX0,AY0, AX1,AY1, AR, AF
- Status příznaky:
  - AZ zero status
  - AN negative status
  - AC carry status
  - AV overflow status
  - AS X-input sign status
  - AQ quotient status



# Příklad: ADSP 21XX - standardní funkce ALU

| Funkce        | Popis                     | Funkce                 | Popis                        |
|---------------|---------------------------|------------------------|------------------------------|
| $R = X+Y$     | Sečti X a Y               | $R = Y+1$              | Inkrementuj Y                |
| $R = X+Y+C$   | Sečti X a Y a carry bit   | $R = X+1$              | Dekrementuj X                |
| $R = X-Y$     | Odečti Y od X             | $R = \text{PASS } X$   | X ulož do R                  |
| $R = X-Y+C-1$ | Odečti Y od X s výpůjčkou | $R = \text{PASS } Y$   | Y ulož do R                  |
| $R = Y-X$     | Odečti X od Y             | $R = \text{ABS } X$    | Absolutní hodnota X          |
| $R = Y-X+C$   | Odečti X od Y s výpůjčkou | $R = X \text{ AND } Y$ | Log. součin X a Y            |
| $R = -X$      | Negace X (2-kový doplněk) | $R = X \text{ OR } Y$  | Log. součet X a Y            |
| $R = -Y$      | Negace Y (2-ková doplněk) | $R = X \text{ XOR } Y$ | Log. excl. or X a Y          |
| $R = 0$       | Nuluj R                   | $R = \text{NOT } X$    | Log. negace X (1-vý doplněk) |
|               |                           | $R = \text{NOT } Y$    | Log. negace X (1-vý doplněk) |



# Jednotka MAC

- Zajišťuje efektivní vysokorychlostní násobení s možností akumulování či odečítání, je možné zapnout ošetření při přetečení (saturace) – **násobení je nejdůležitější operace** při jakémkoliv algoritmu číslicového zpracování signálu
- FRACTIONAL a INTEGER mód – jedním bitem se nastavuje chování
  - ✓ **FRACTIONAL MODE** pro formát 1.15 - před zápisem do registru výsledku (MR) - posun výsledku násobení o 1 bit doleva 2.30 -> 1.31 (LSB nastaven na log. 0 )
  - ✓ **INTEGER MODE** pro formát 16.0 – výsledek násobení se přímo zapíše do registru výsledku (MR)

General Rule

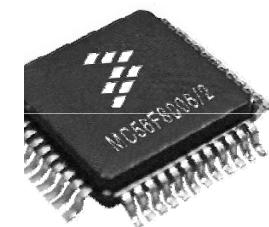
$$\begin{array}{r} M.N \\ \times P.Q \\ \hline (M + P).(N + Q) \end{array}$$

4-bit Example

$$\begin{array}{r} 1.111 \text{ (1.3 Format)} \\ \times 11.11 \text{ (2.2 Format)} \\ \hline 1111 \\ 1111 \\ 1111 \\ \hline 111.00001 \text{ (3.5 Format = (1 + 2).(2 + 3))} \end{array}$$

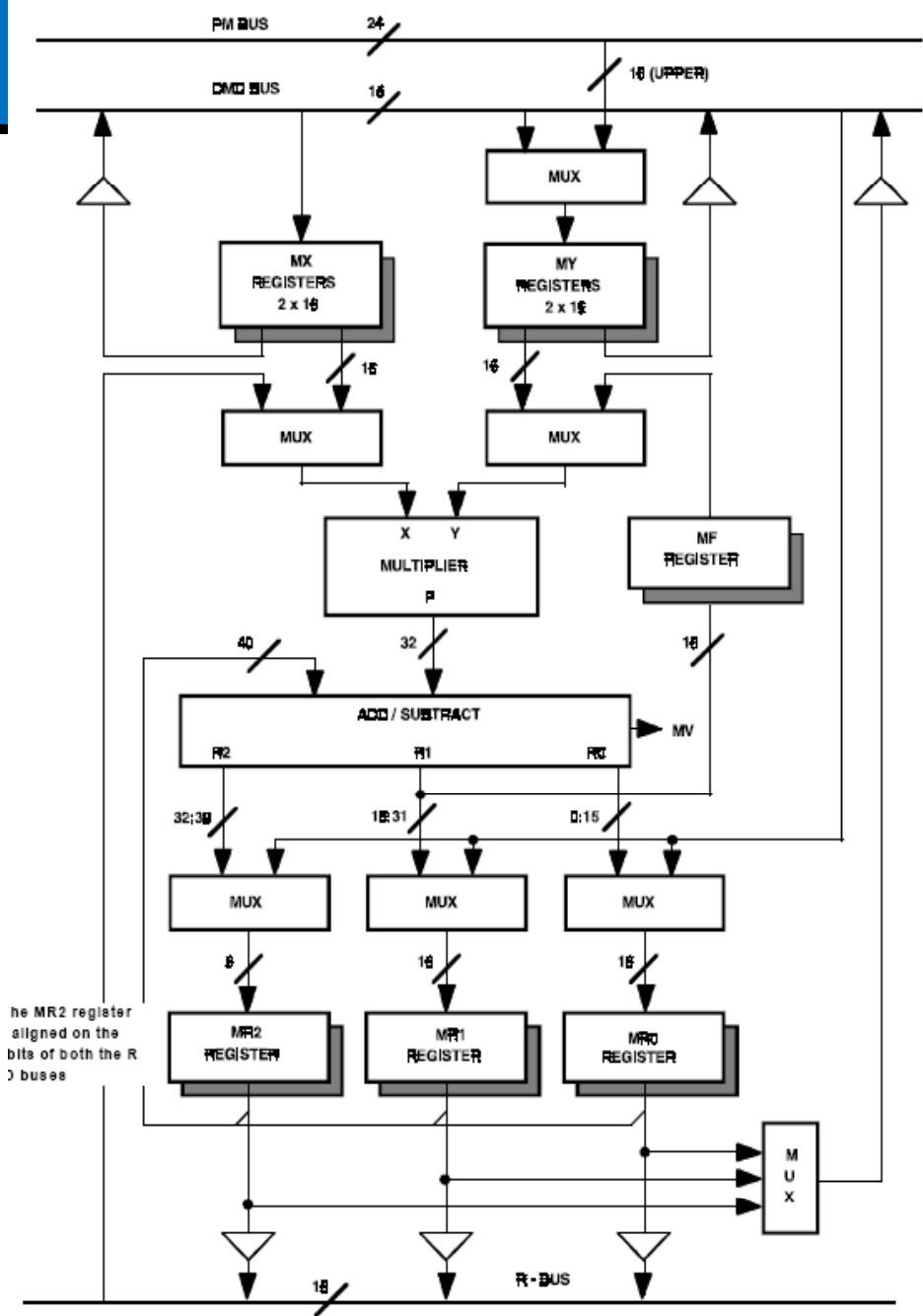
16-bit Examples

$$\begin{array}{r} 5.3 \\ \times 5.3 \\ \hline 10.6 \\ 1.15 \\ \times 1.15 \\ \hline 2.30 \end{array}$$



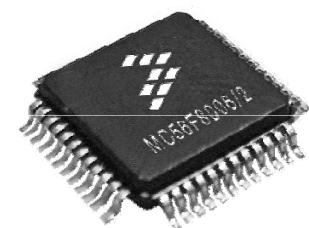
# Příklad: ALU v ADSP 21XX

- 16-bit. operace nad vstupy X,Y nebo MR registrem
- Registry: MX0,MY0, MX1,MY1, MR, MF
- Status příznaky:
  - AZ zero status
  - AN negative status
  - AC carry status
  - AV overflow status
  - AS X-input sign status
  - AQ quotient status



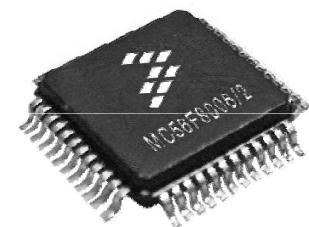
# Příklad: MAC v ADSP 21XX - standardní funkce MAC

| Funkce            | Popis                         |
|-------------------|-------------------------------|
| $MR = X * Y$      | Vynásob $X * Y$               |
| $MR = X * X$      | Vynásob $X * X$               |
| $MR = MR + X * Y$ | Vynásob $X * Y$ a přičti k MR |
| $MR = MR - X * Y$ | Vynásob $X * Y$ a odečti k MR |
| $MR = 0$          | Vynuluj MR registr            |



# Příklad: DAG v ADSP 21XX

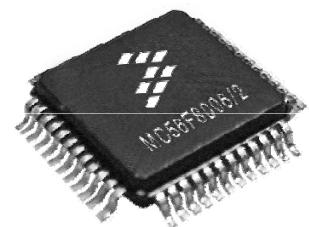
- DAG - data address generator (v jádře 2x)
- zajišťují efektivní využívání výpočetních jednotek- nepřímá adresace pamětí
- DAG1 – čtení/zápis s DM
- DAG2 – čtení/zápis s DM i PM
- každý má 4 ukazatele - I registry
- každý má 4 modifikátory - M registry
- každý má 4 L-registry pro definici délky kruh. Bufferu
- DAG1 – používá M,I, L registry s indexem 0 až 3
- DAG2 – používá M,I,L registry s indexem 4 až 7
- DAG1 – možnost bitové reverzace (výhodné pro výpočet FFT)



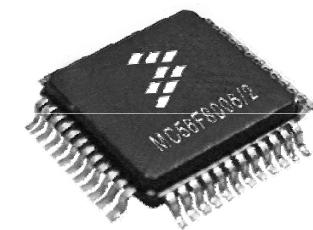
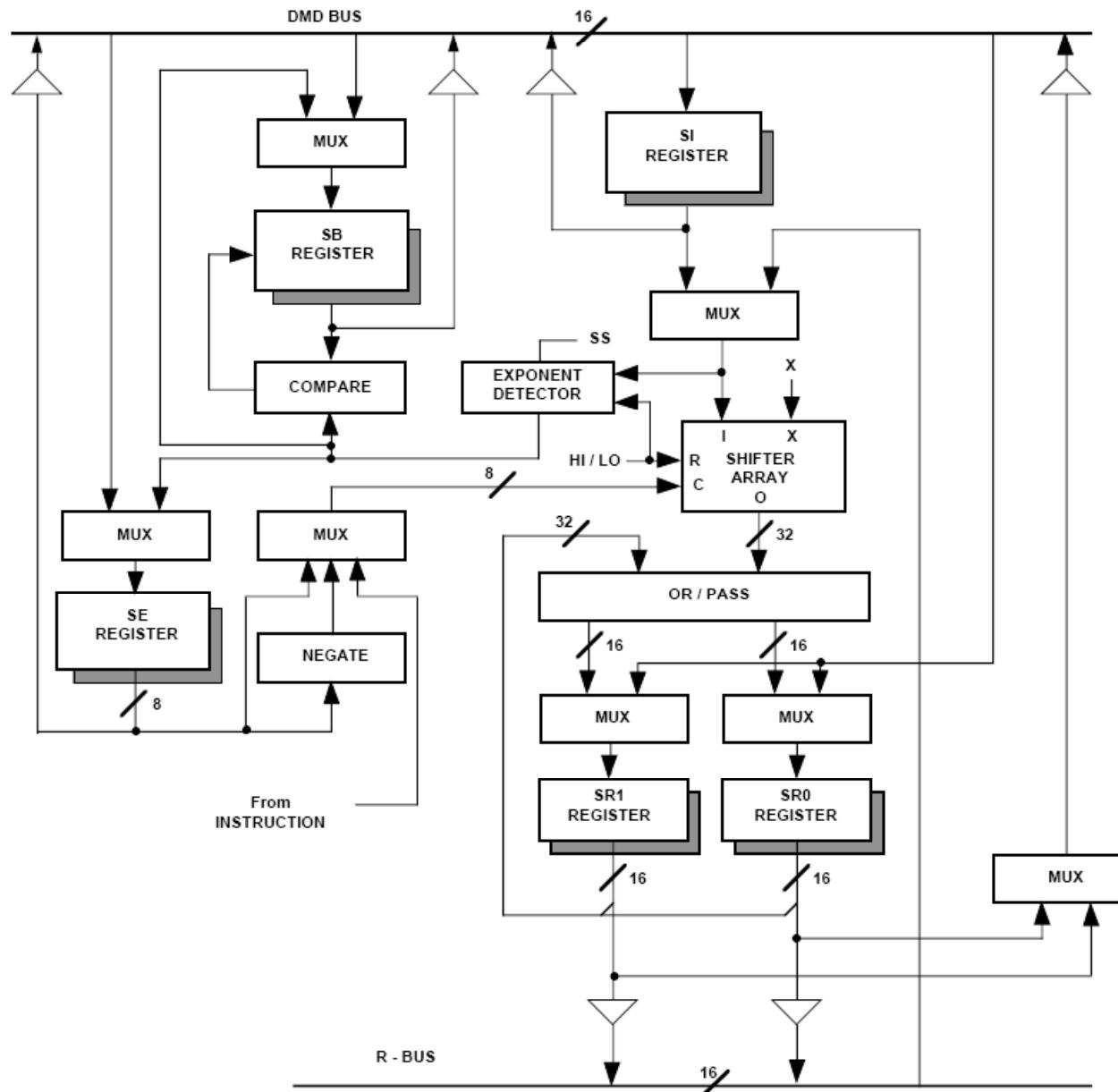
# Jednotka Barrel Shift register

Jednotka zajišťuje kompletní operace posunutí 16-bit. vstup a 32-bit. výstup:

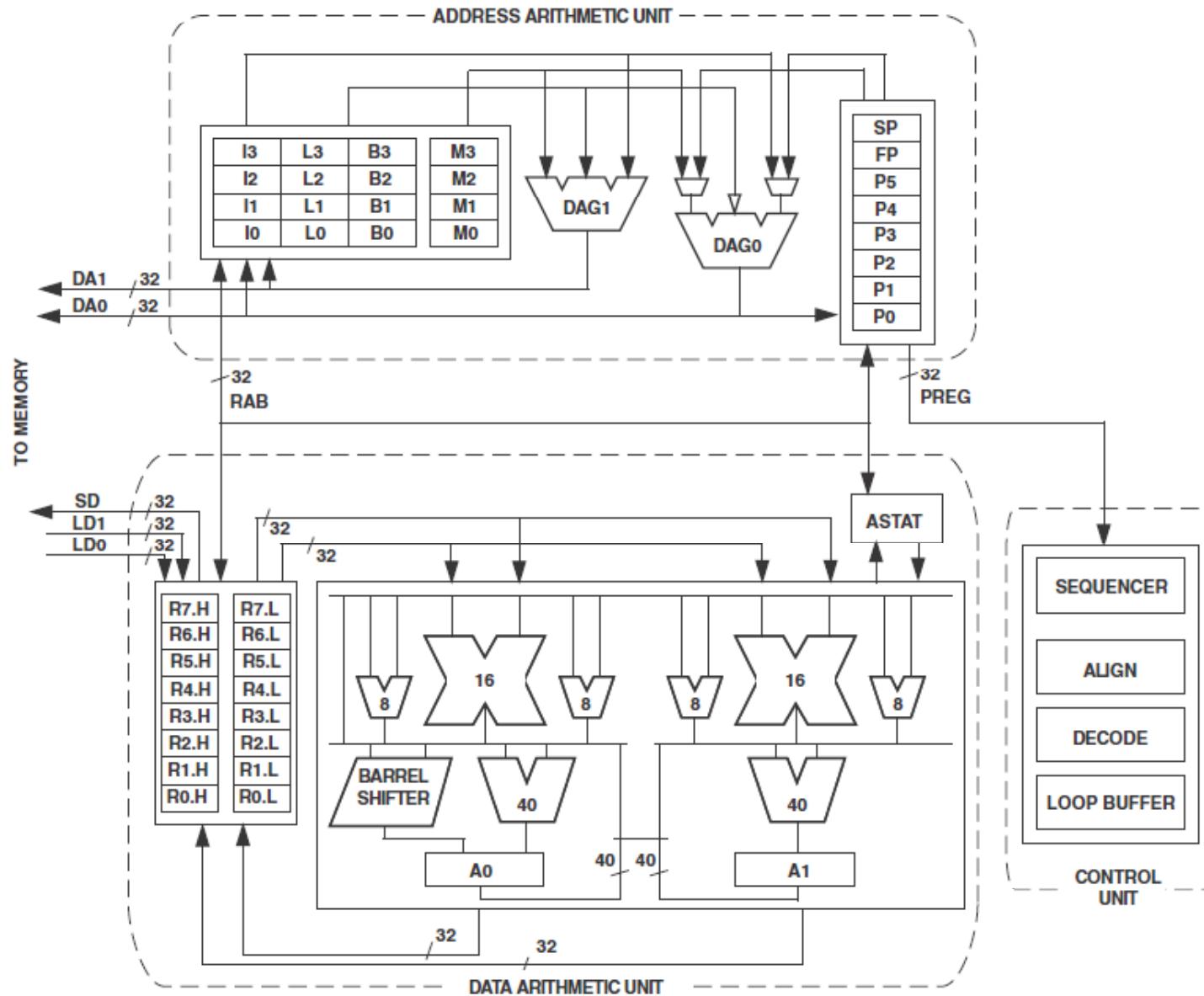
- ✓ aritmetický posun
- ✓ logický posun
- ✓ normalizaci (podpora pro floating-point operace)
- ✓ derivaci exponentu (podpora pro floating-point operace)
- ✓ derivaci exponentu vstupní bloku čísel (podpora pro floating-point operace)



# Příklad: Barrel Shift registr v ADSP 21XX



# Příklad : ALU + MAC + DAG v ADSP-5XX (Blackfin)



# Přední výrobci signálových procesorů / výkonové řady

- **Analog Devices**

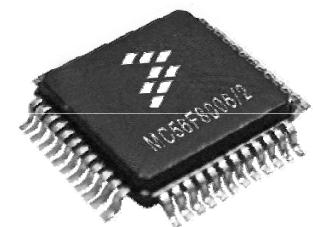
- ADSP-21xx, Blackfin, SHARC, TigerSHARC, SigmaDSP (audio)

- **Texas Instruments**

- DaVinci Digital Media Processors, C2000, C5000, C6000

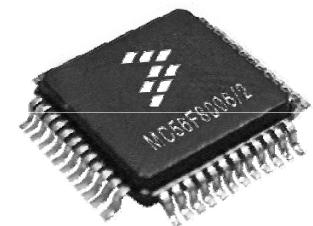
- **Freescale (dříve Motorola)**

- 16-bit. StarCore, 24-bit. Symphony DSP56xxx, 16-bit. MC56Fxx



Signálové procesory firmy

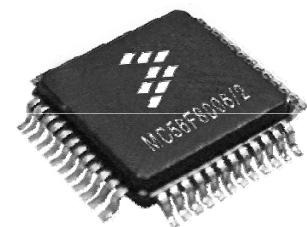
**ANALOG DEVICES**



# Přehled DSP procesorů firmy Analog Devices

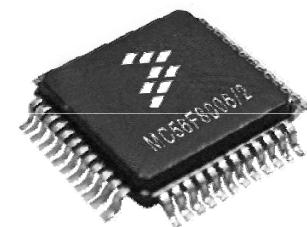
Řazeno vzestupně dle výpočetního výkonu:

- řada **ADSP21xx** (nejjednodušší, malý výkon)
- řada **Sigma DSP** (integrovaný audio-kodek, výhodné pro audio-signal processing)
- řada **Blackfin** (nepoužívanější)
- řada **Sharc**
- řada **TigerSharc** (výkonově nejsilnější, více jader na čipu)

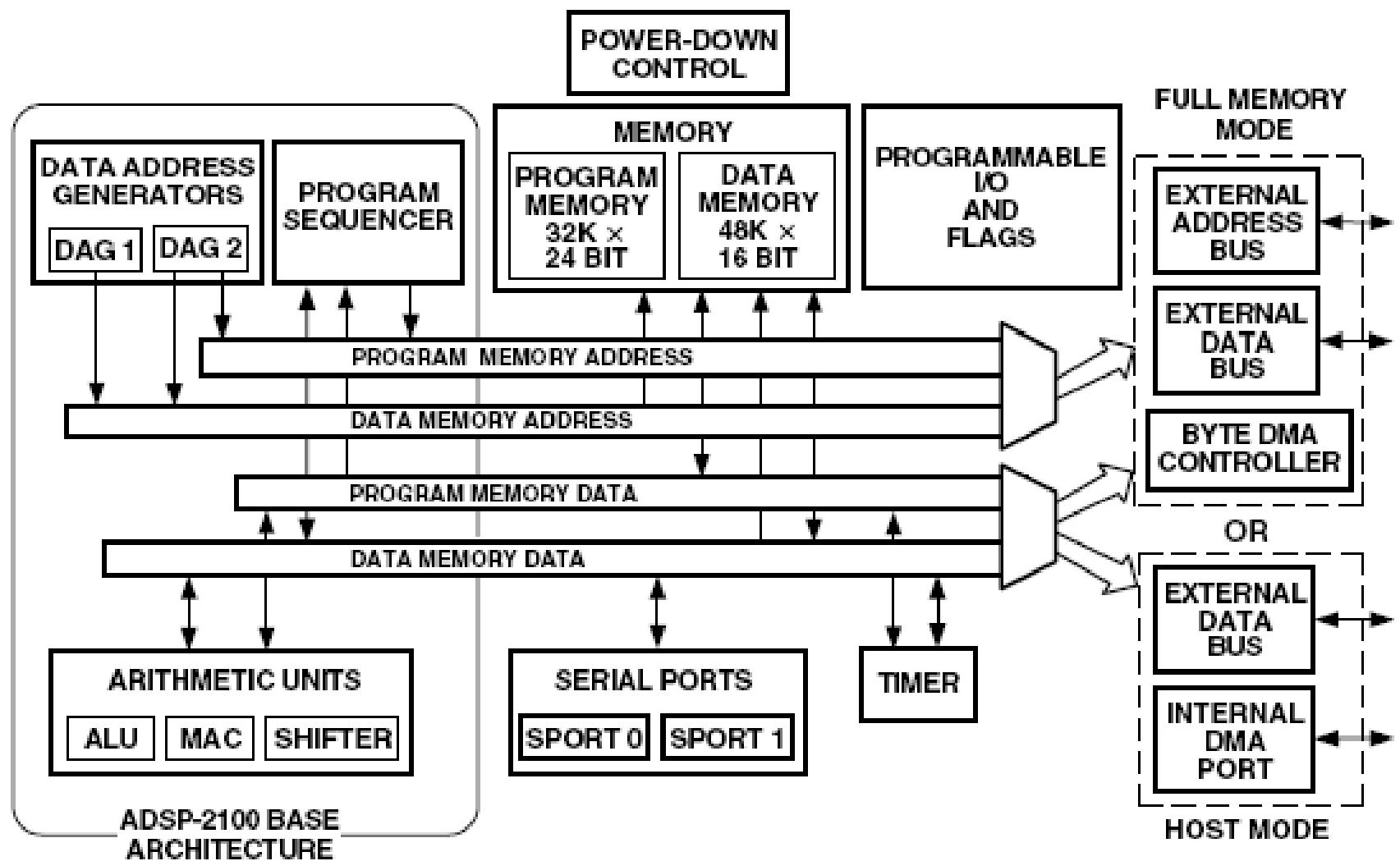


# Procesory Analog Devices řady 21xx

- výpočetní výkon 75 MIPS @ CLK 37.5MHz (13,3 ns/instrukci)
- 192kB RAM (96kB program. paměti, 96 kB datové paměti)
- šířka programové sběrnice **24 bitů**, datové **16 bitů** !
- 2x DAG
- 1x ALU,MAC , shift registr
- 16-bit. časovač
- 2x SPORT
- DMA řadič
- cena okolo \$ 30 USD

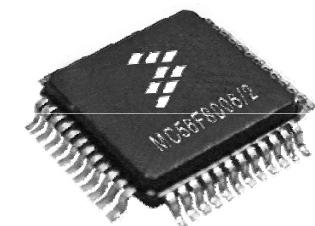


# ADSP 2189M – vnitřní architektura

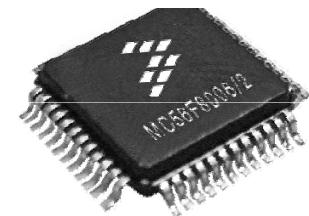
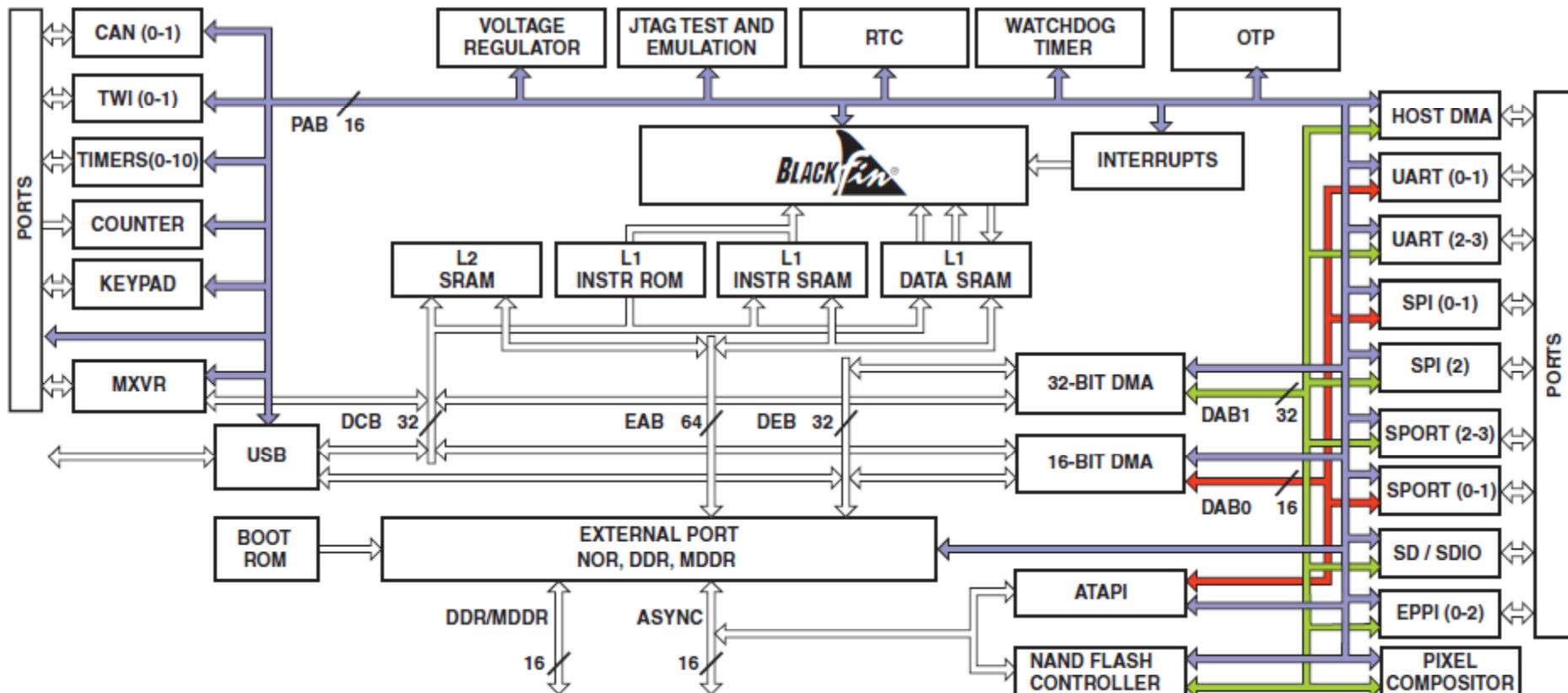


# Procesory Analog Devices - řada BLACKFIN

- 16/32-bit procesory pro vestavěné aplikace – výrazná možnost volit periférie podle potřeby aplikace
- 2x 16bit. MAC
- 2x 40 bit. ALU
- 4x 8 bit. video ALU
- 2x DAG
- Barrel shift registr
- Řadič pro připojení externích pamětí (NOR, NAND flash, SDRAM)
- Široká nabídka periferií (Ethernet MAC, UART, SPI, CAN řadiče, SD host, USB, časovače s podporou PWM, Watchdog, hodiny reálného času, JTAG, sériové porty)
- Cena okolo 20 \$ USD
- Max. CLOCK až cca 756 MHz (1500 MIPS)
- Nízký příkon – vhodný pro low power aplikace

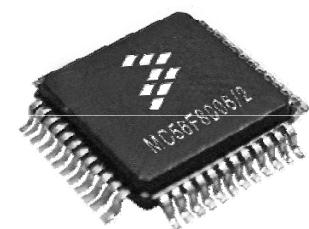


# Blackfin – typ BF548 – vnitřní architektura

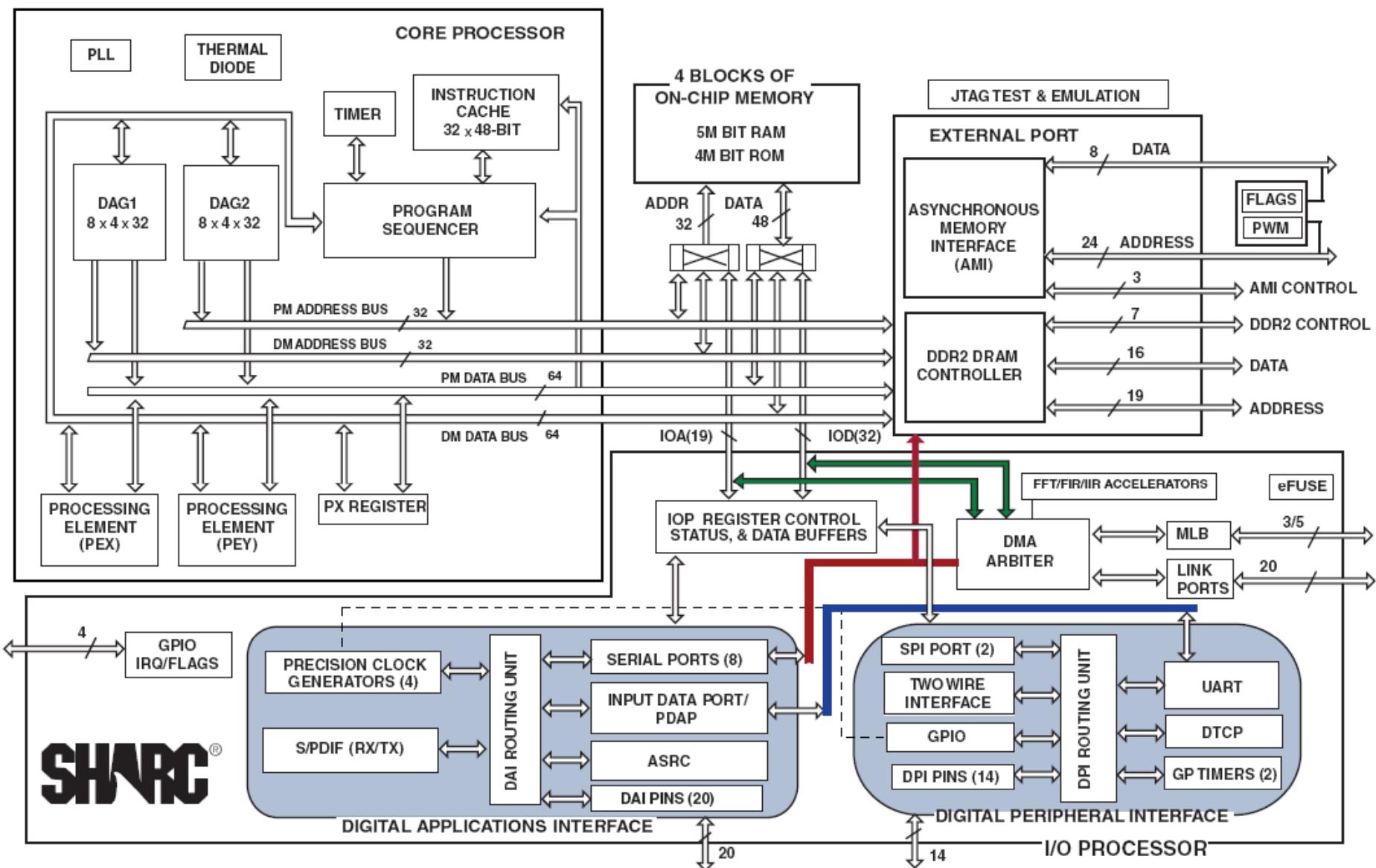


# Procesory Analog Devices řady SHARC

- 32-bit. floating-point procesor
- 32-bit. fix-point násobičky, 80-bit akumulace
- žádný pipelining pro výpočty – vše v rámci jednoho cyklu
- Podpora kruhových bufferů, 16 adresních ukazatelů
- HW podpora pro aritmetické operace dělení, odmocnina, bitové operace
- 2x procesní elementy - 32-bit. floating-point MAC, ALU, shift a data registry
- 2x DAG
- SDRAM řadič
- SRAM (max. 5Mb) na čipu
- Max. CLK 400 MHz (2,4 GFLOPS)
- Cena do \$30 USD

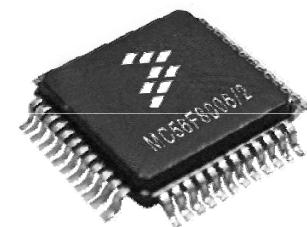


# SHARC ADSP-2146x

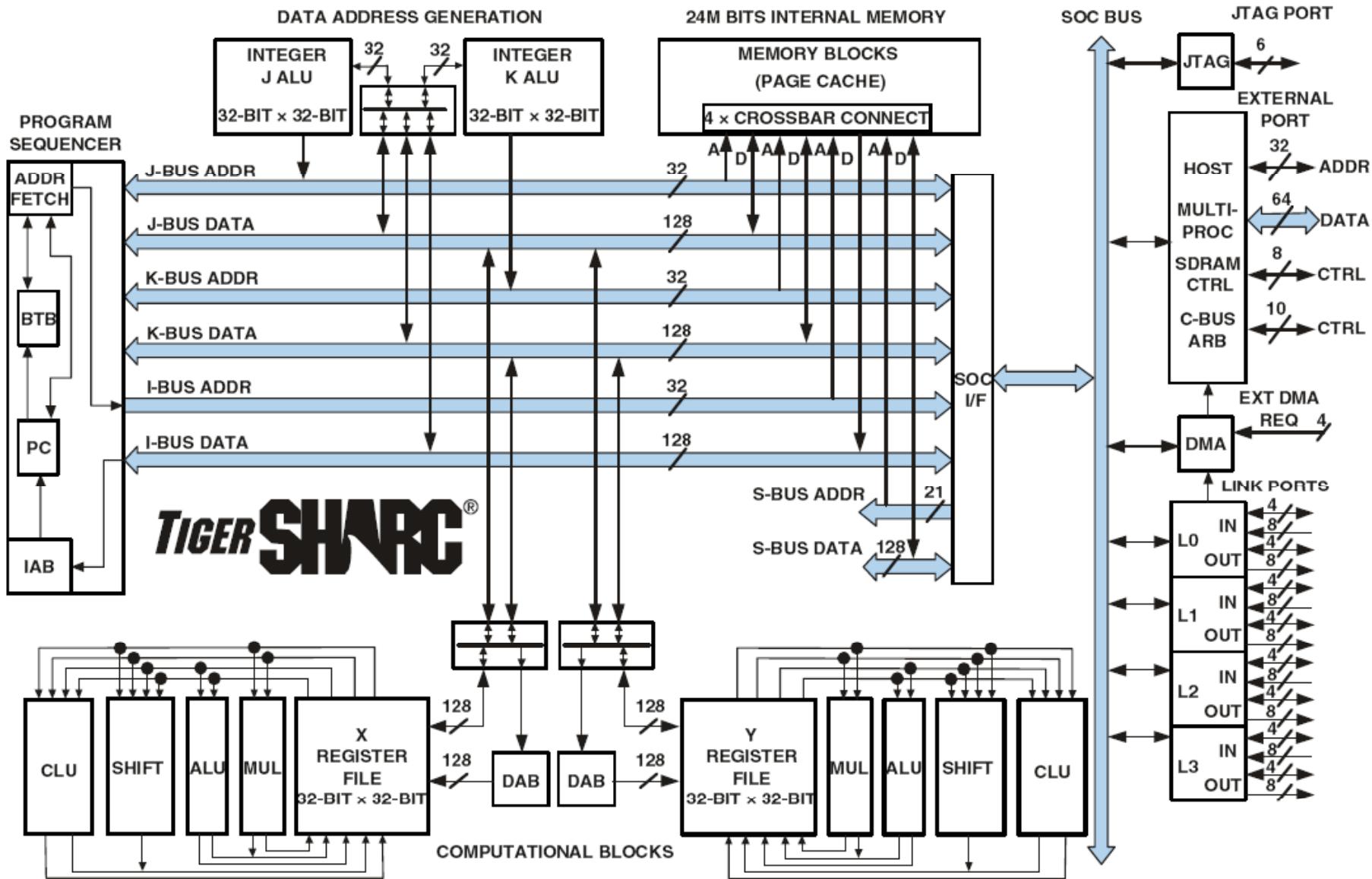


# Procesory Analog Devices řady TigerSHARC

- 2x výpočetní bloky s ALU, MAC,shift reg., CLU
- 2xDAG – dvě integer ALU
- až 4 instrukce v jednom cyklu
- 24 Mb DRAM na čipu
- Interface pro externí SDRAM,SRAM,
- 14-kanálový DMA řadič
- 4x LVDS porty (full-duplex)
- CLK max. 600 MHz (4,8 G operací typu MACs)
- cena okolo \$300 USD

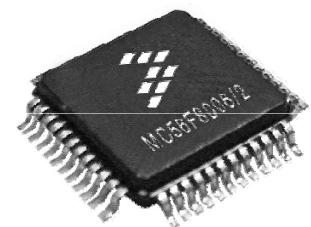


# TigerSHARC ADSP – TS201



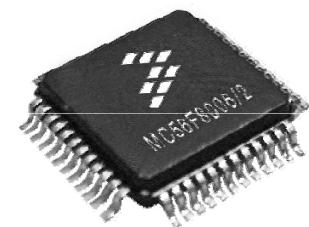
Signálové procesory firmy

**TEXAS INSTRUMENTS**

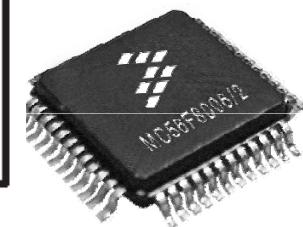
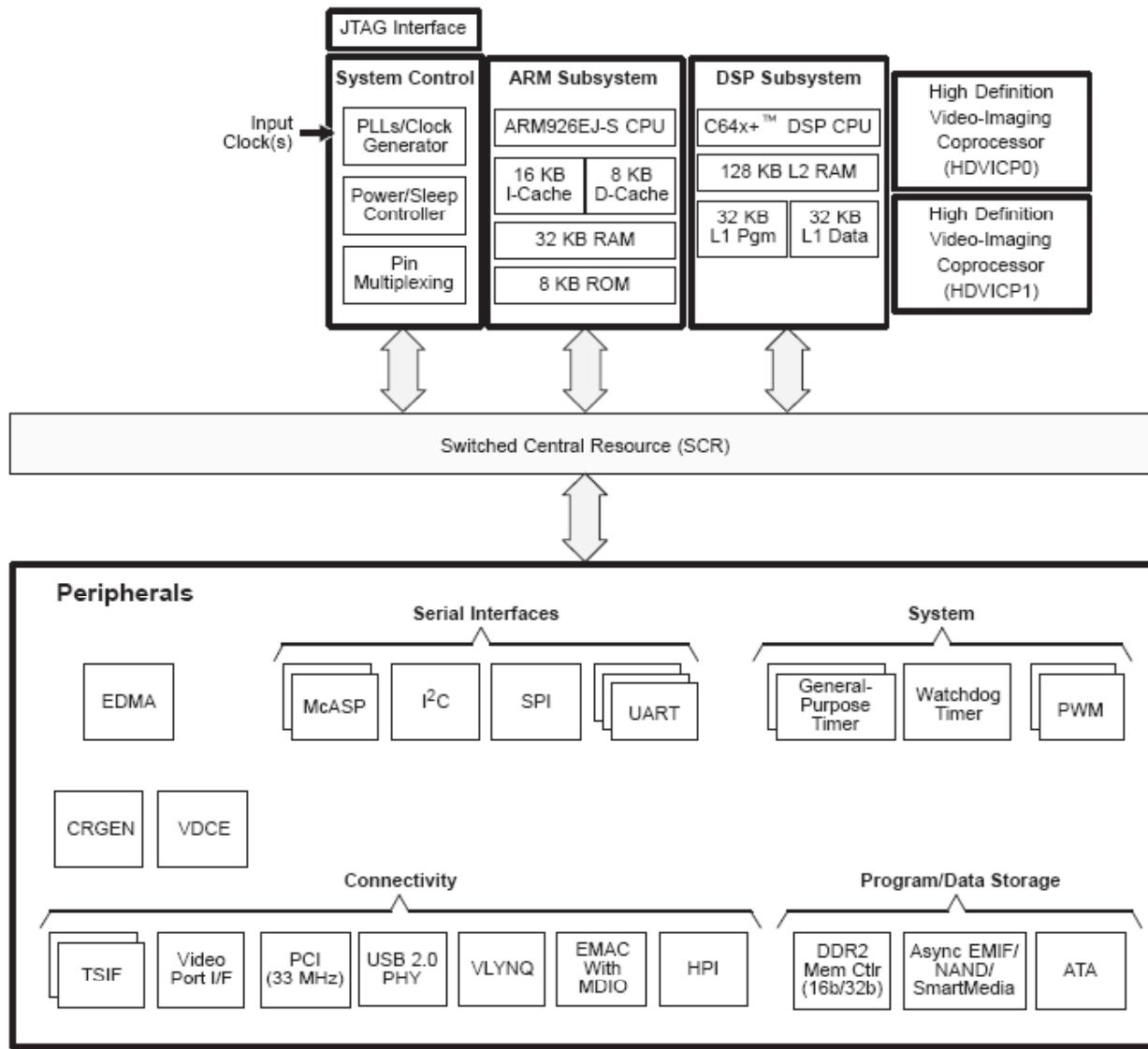


# Procesory TI řady TMS320DM64xx

- Procesory řady Digital Media Processor
- Obecně 32-bit. procesor
- Max. CLK 729 MHz
- Osm 32-bit. instrukcí / 1 cyklus
- Výpočetní výkon až 5 800 MIPS
- Jádro ARM926EJ-S, proto SW kompatibilní s ARM9
- 8 nezávislých výpočetní jednotek (6x ALU, 2x MAC)
- EMIF (řadič pro ext. SDRAM, FLASH, řadič DMA)
- Rozhraní pro FPGA, multiple audio serial port
- Široká nabídka periférií: Ethernet, USB 2.0, PCI rozhranní, 2x 64-bit. časovač, 1x 64-bit. watch-dog, 3xUART,I2C,SPI,HPI, ATA/ATAPI
- Cena okolo \$ 130 USD

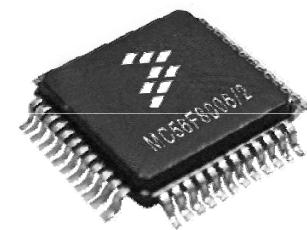


# DMS320DM6467

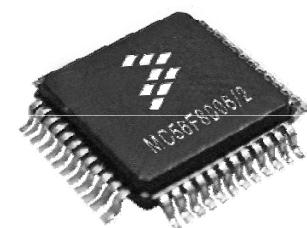
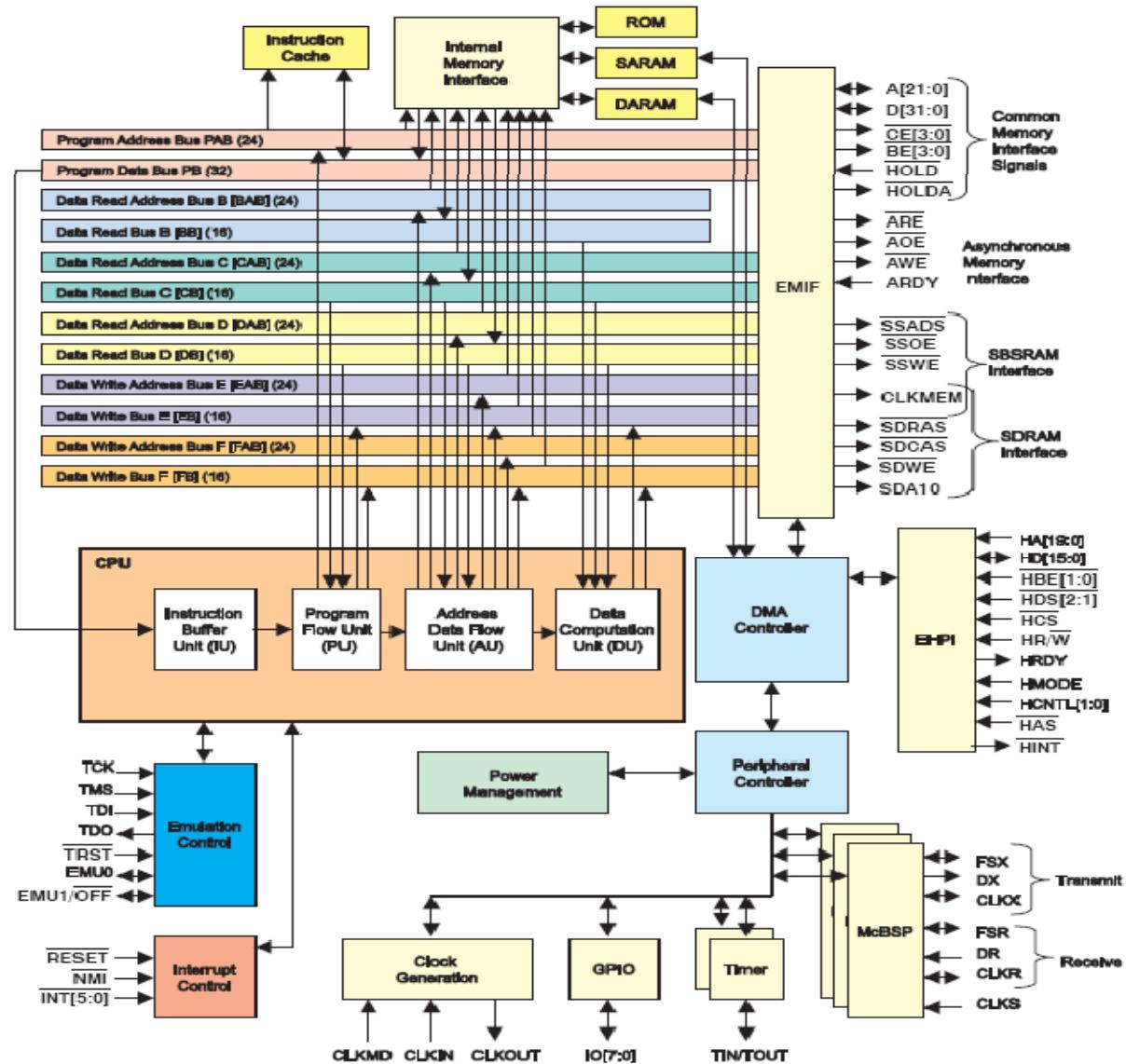


# Procesory TI řady C5000 low power

- 16-bitový fixed-point procesory
- Nízká spotřeba v standby modu (0.12mW)
- Výpočetní výkon až 600 MIPS @ 300 MHz
- 2x MAC
- 2x ALU
- RAM na čipu (cca 320 kB)
- 32-bit EMIF – external memory interface: SRAM, EPROM, SDRAM, SBSRAM)
- Cache 24 kB
- 3-násobná datová sběrnice pro čtení, 2-násobná pro zápis
- 2x 20-bit. časovače
- 3x McBSP (UART)
- 8 GPIO

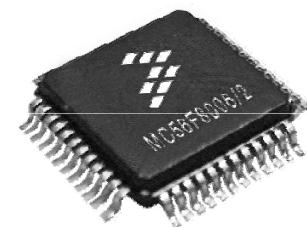


# TMS320VC5510



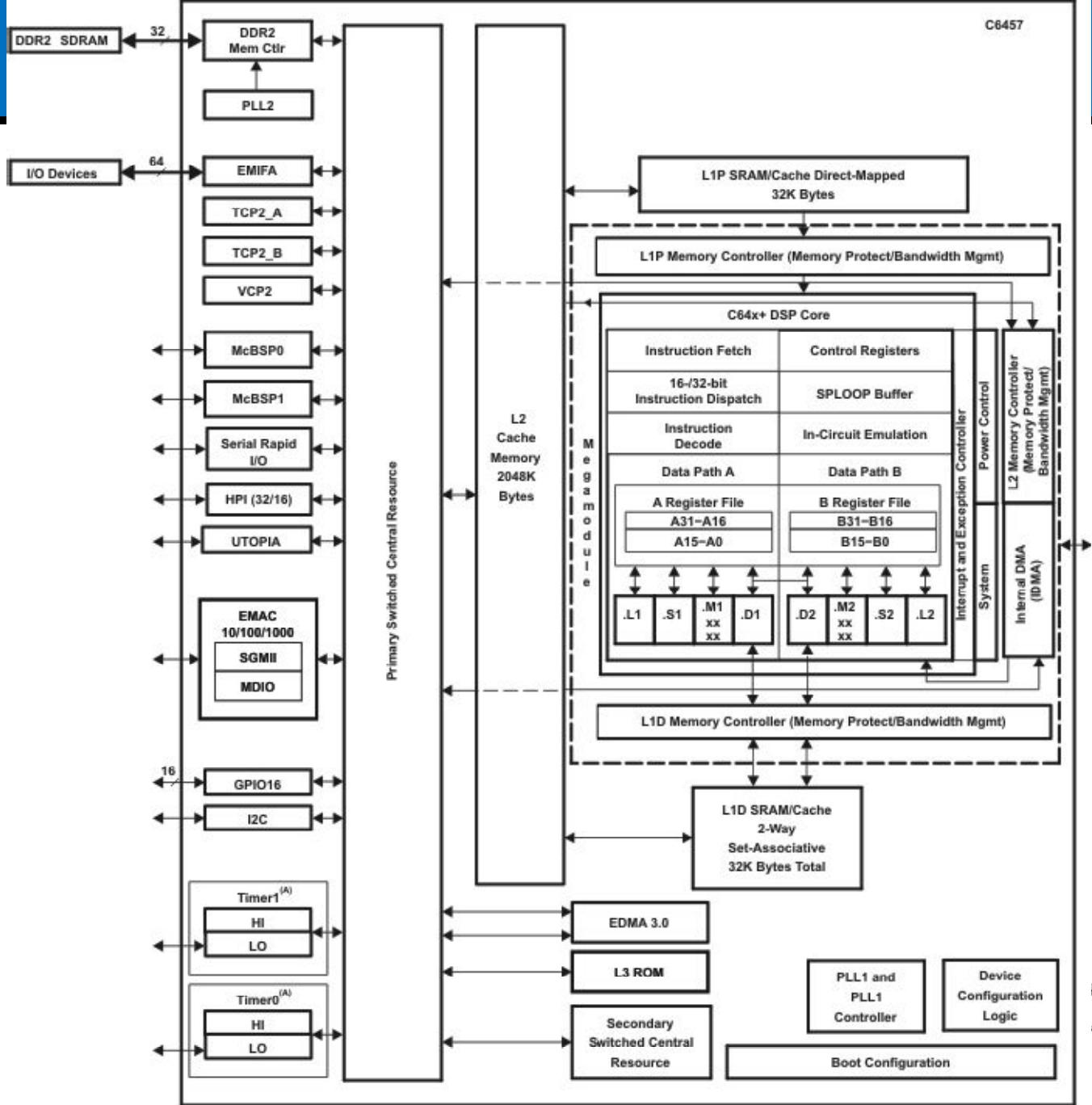
# Procesory řady TI C6000 High performance

- Nejvýkonější řada TI, výkon až 9 600 MIPS/MMAC
- DSP jádro, cache L1 (32kB), L2 (2MB)
- EDMA řadič
- DMA řadič
- PCI řadič
- Veterbi coprocessor
- Turbo coprocessor
- EMIF
- McBSP( UART)
- HPI (host processor interface)
- 32-bit X-bus
- RAPID IO – rozhranní pro DSP-DSP nebo DSP-FPGA
- 16-bit MAC, 32-bit instrukce
- Až 7 Mb RAM na čipu



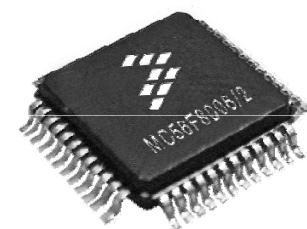
# TMS320C6457

C6457

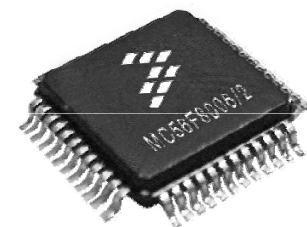
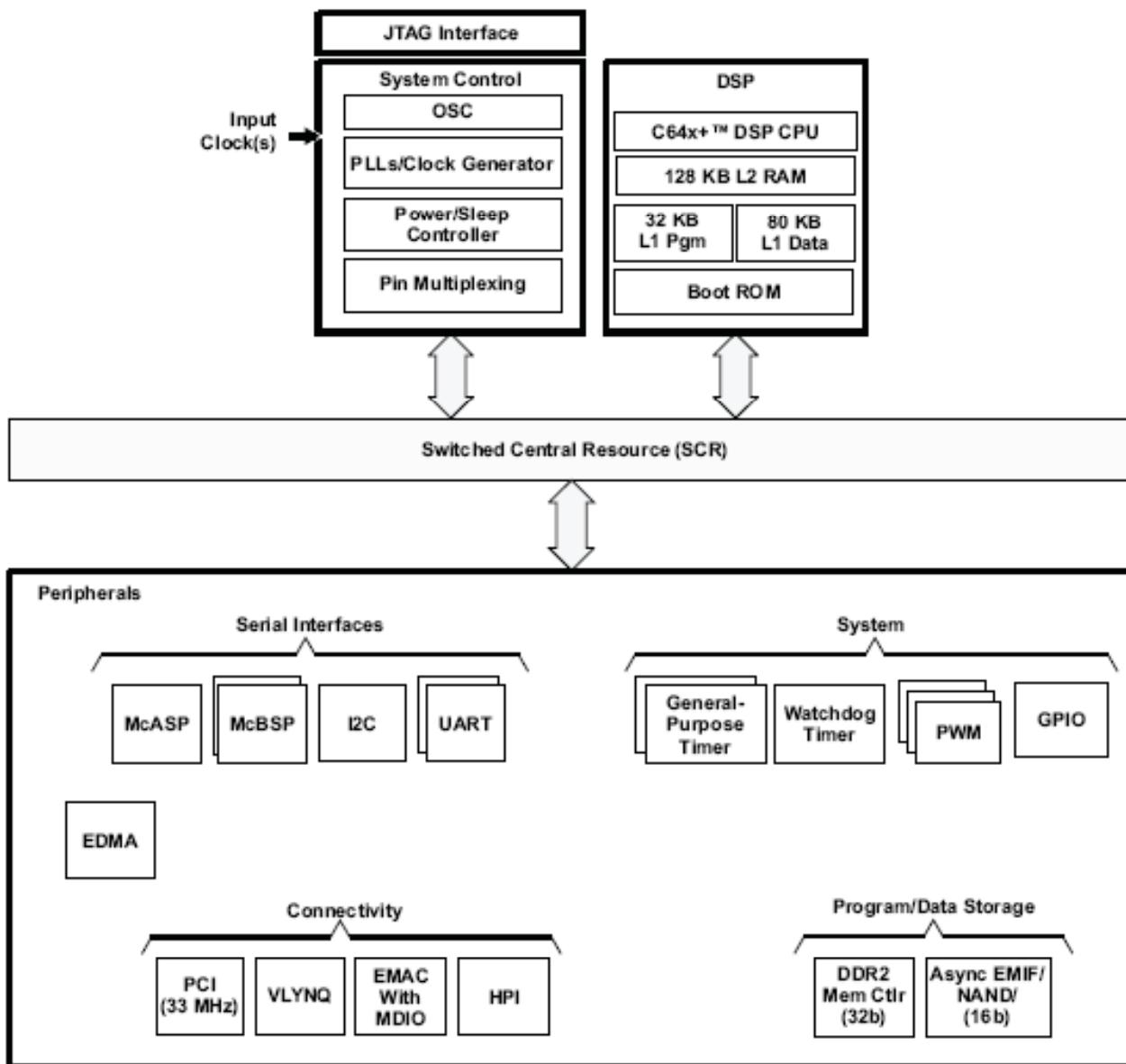


# Procesory TI řady C6000 Performance value

- Středně výkoná řada TI 32-bit. fixed-point procesorů
- Výpočetní výkon až 4800 MMACS @ 600 MHz
- 6x ALU (single 32,dual 16, quad 8-bit. aritmetice)
- 2x MAC (single 16x16,dual 8x8)
- EDMA řadič (propustnost 4,8 GB/s, DDR2 133 MHz)
- Ethernet
- Cache L1,L2
- EMIF : SDRAM, NOR FLASH, SRAM, NAND FLASH
- 2x GP časovač, 2x UART, 2x McBSP
- VLYNQ rozhranní pro připojení FPGA
- Cena cca \$ 30 USD

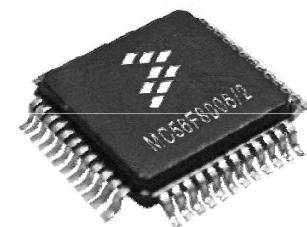


# TMS320C6424

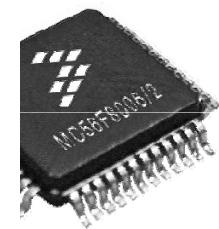
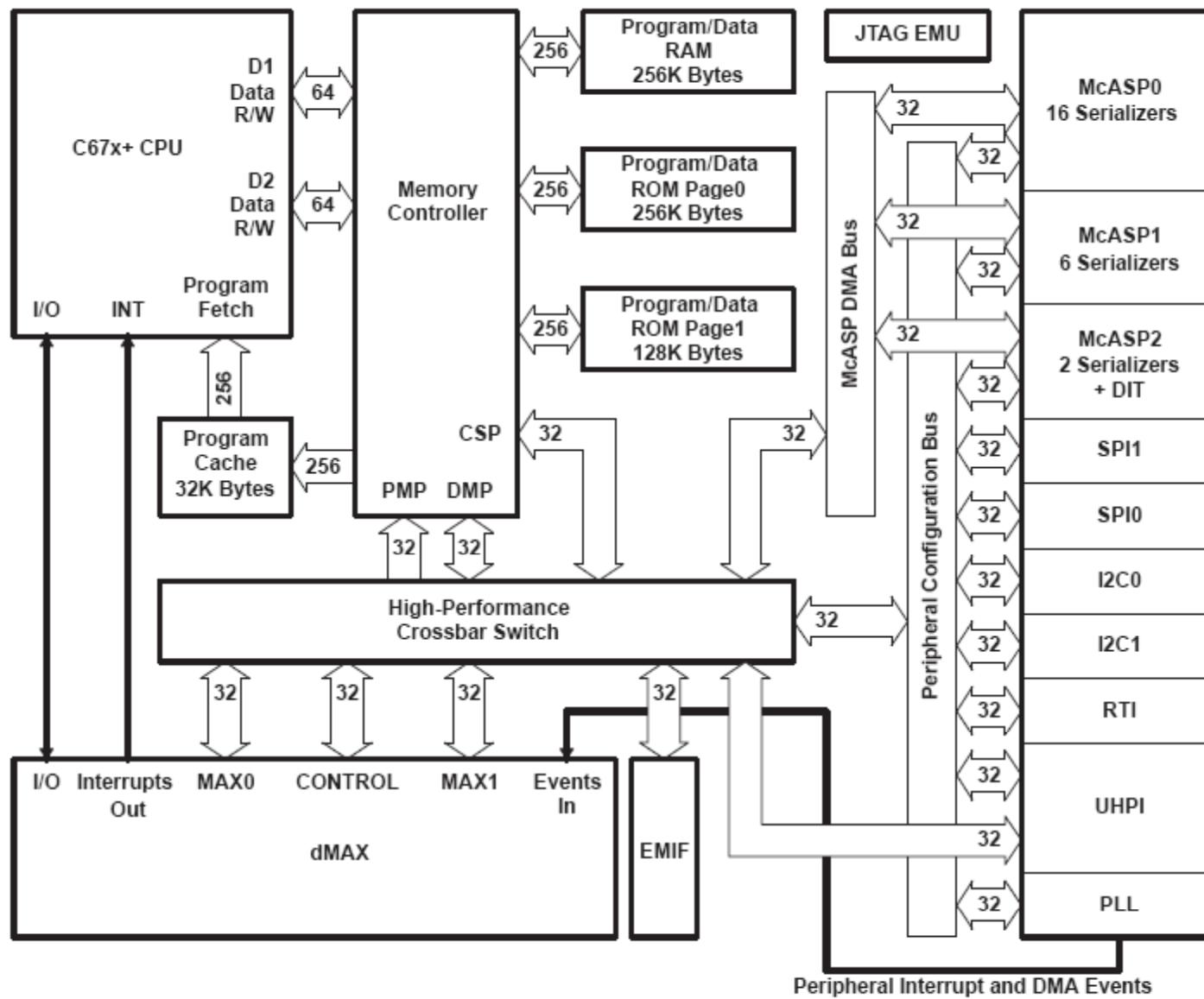


# Procesory TI řady C6000 floating-point

- 32-/64-bitový floating-point procesory
- Výkon cca 130 MFLOPS
- Max. CLK 350 MHz
- 256kB PM/DM RAM
- 384 kB PM/DM ROM
- EMIF: SDRAM, NOR FLASH, SRAM, NAND FLASH
- Crossbar switch
- 3x multichannel Audio Seriál Port
- 1x UHPI (Host processor)
- 2x SPI, I2C, hodiny reálného času, watchdog
- dMAX – dual data movement accelerator
- Cena okolo \$ 8 USD

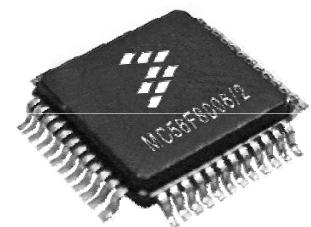


# TMS320D6420



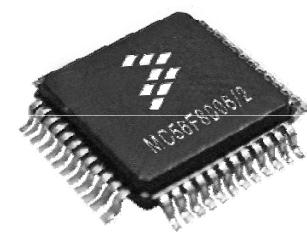
# Signálové procesory firmy

## FREESCALE

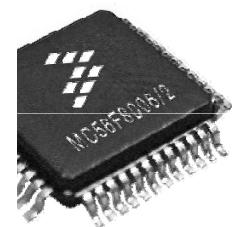
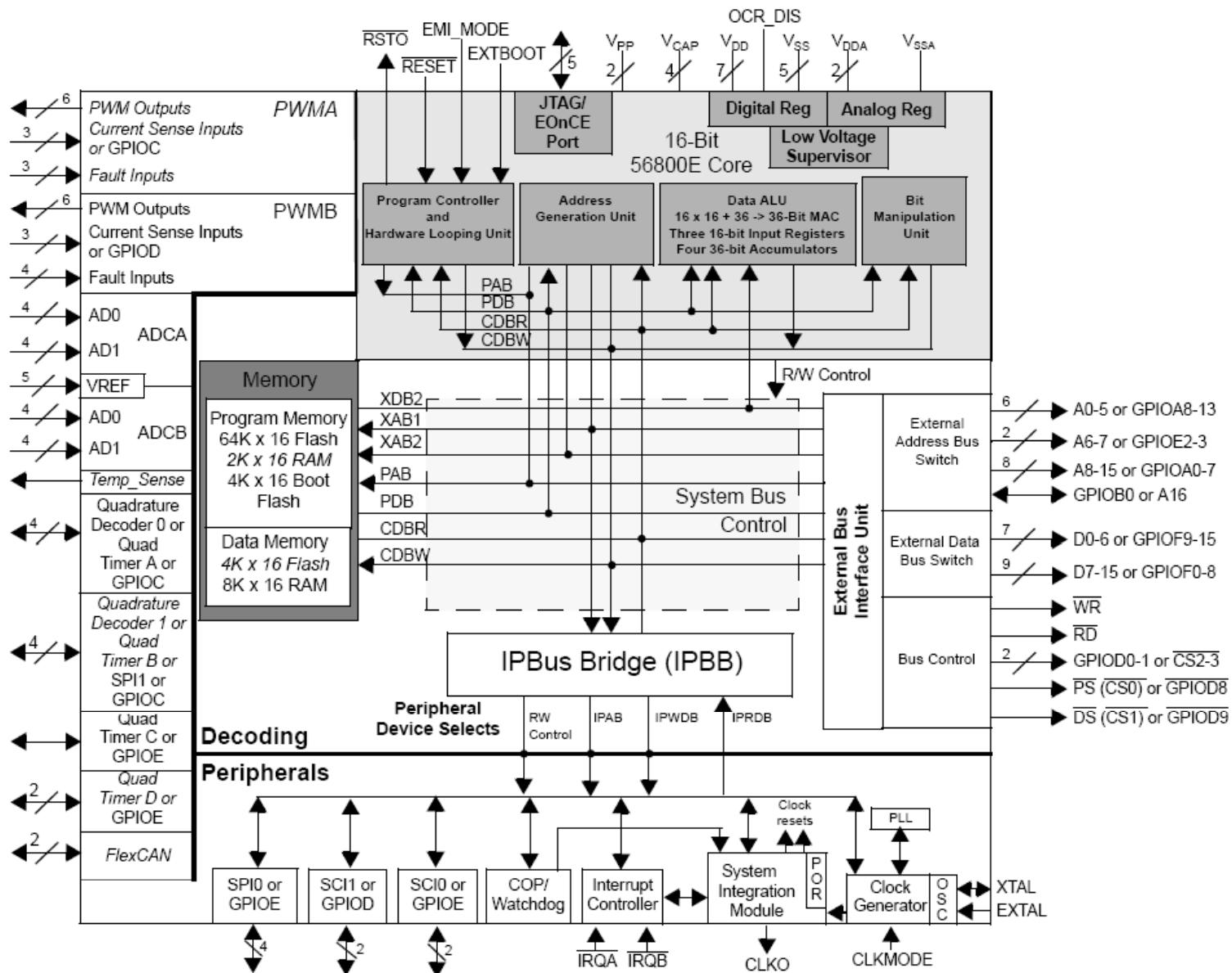


# Freescale řada 56800/E

- Nejnižší výkonová řada 16-bit. fixed point procesorů
- Výpočetní výkon cca 120 MIPS
- 128kB PM
- 4kB PM
- 6x PWM
- 2xSPI
- 2XSerial communication interface
- JTAG
- Flex CAN
- Integrovaný AD převodník
- 62 GPIO

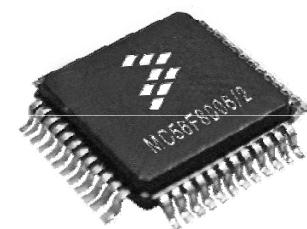


# Freescale 56F8346

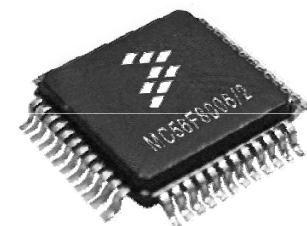
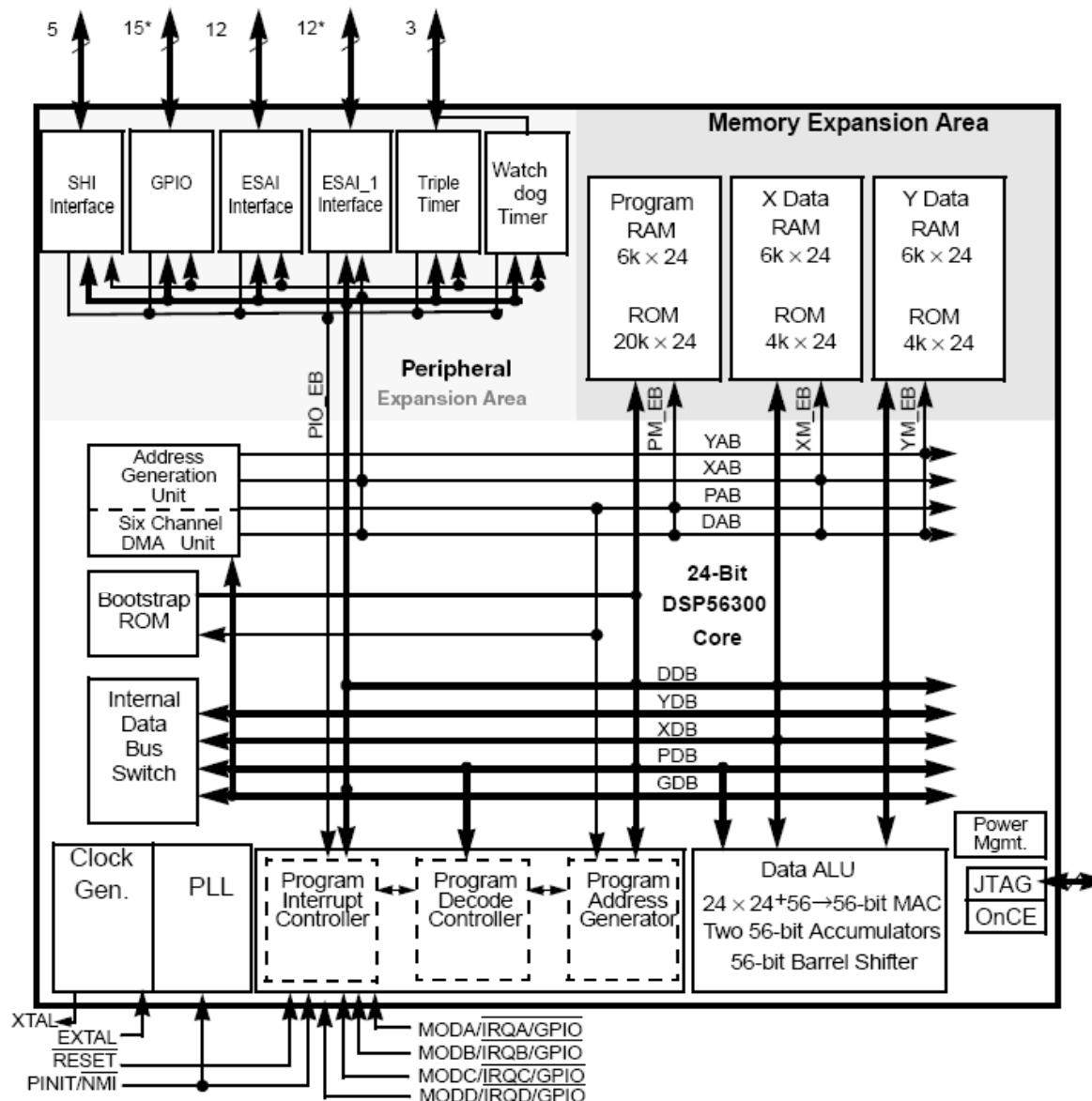


# Freescale řada Symphony

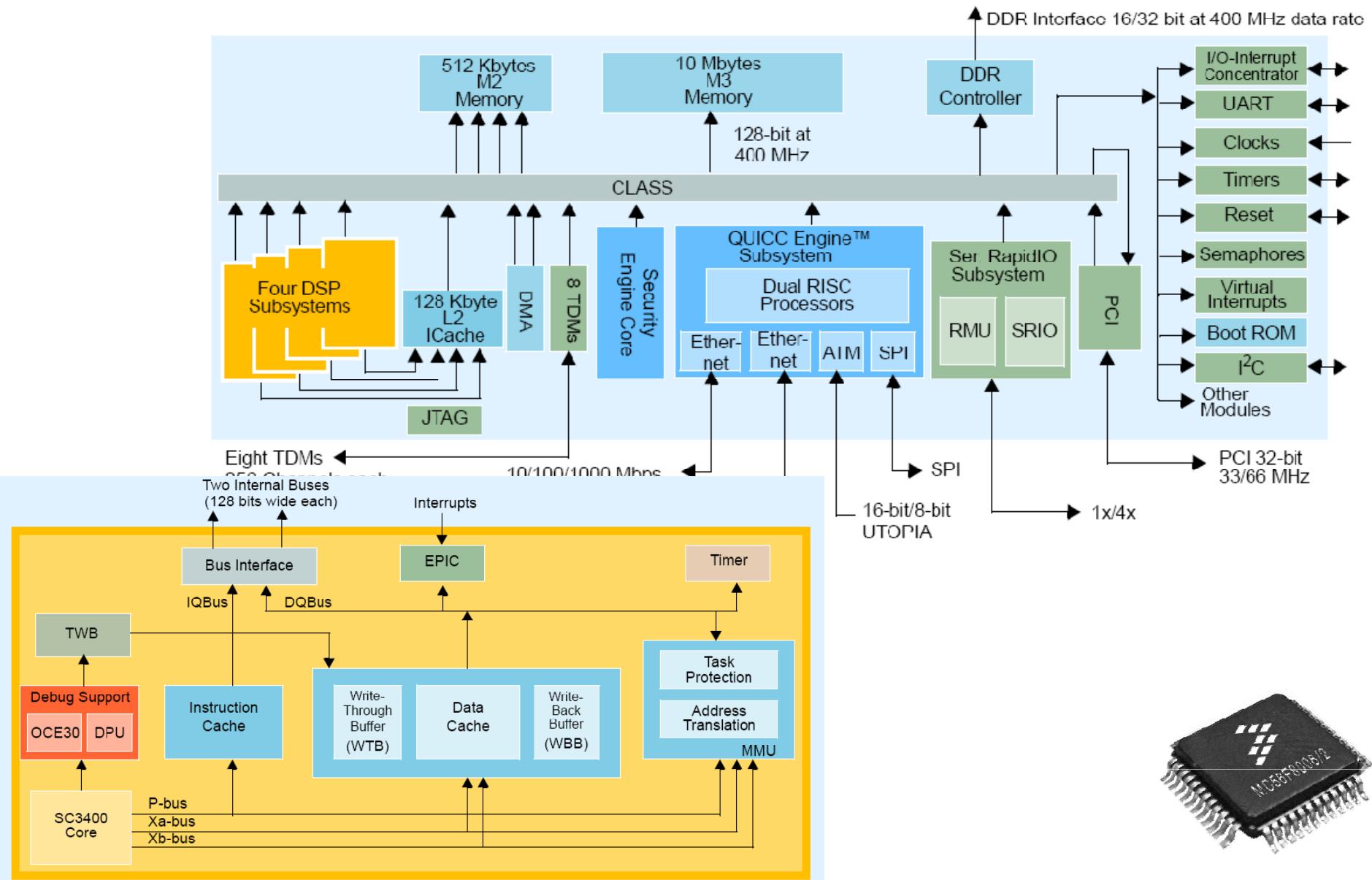
- Středně výkonná řada 24-bit. procesorů – univerzální použití
- Výkon 150 MIPS @ 150 MHz
- ALU 24x24
- 56-bit Barrel Shift registr
- 6-kanálový DMA řadič
- JTAG
- Low power design
- Volitelné velikost RAM a ROM
- Periferní moduly:
  - Enhanced Serial Audio Interface
  - Serial Host Interface
  - Triple Timer Module
  - GPIO (cca 40)



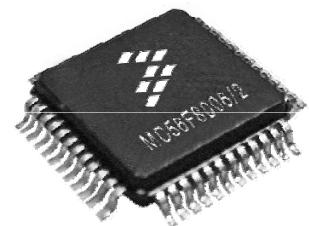
# Freescale DSP56374



# Freescale MSC8144E – quad DSP

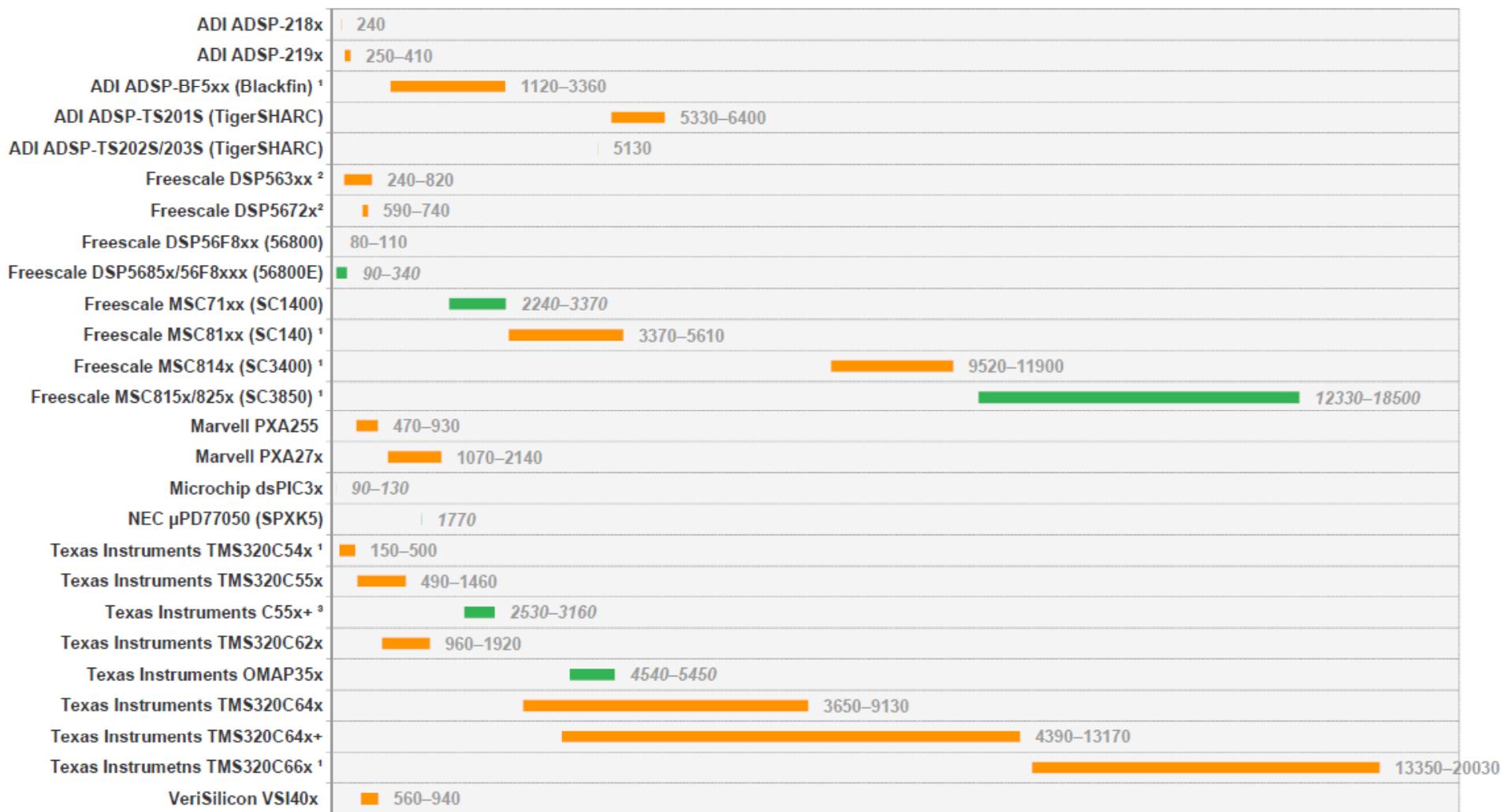


# Výkonnostní porovnání DSP



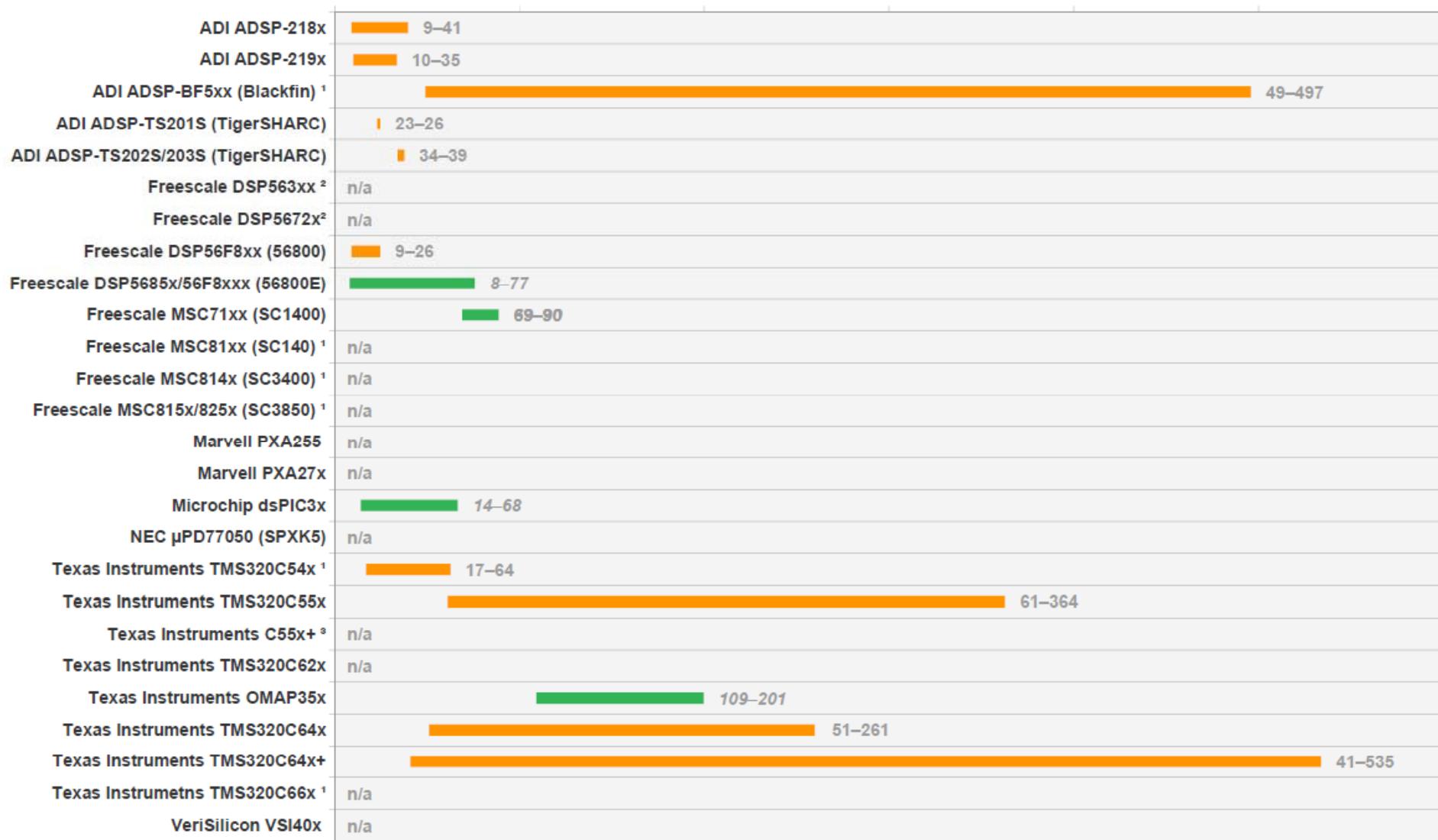
# Výkonové srovnání fixed-point procesorů

## ■ Rychlostní test DSP



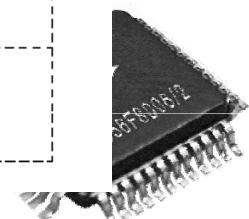
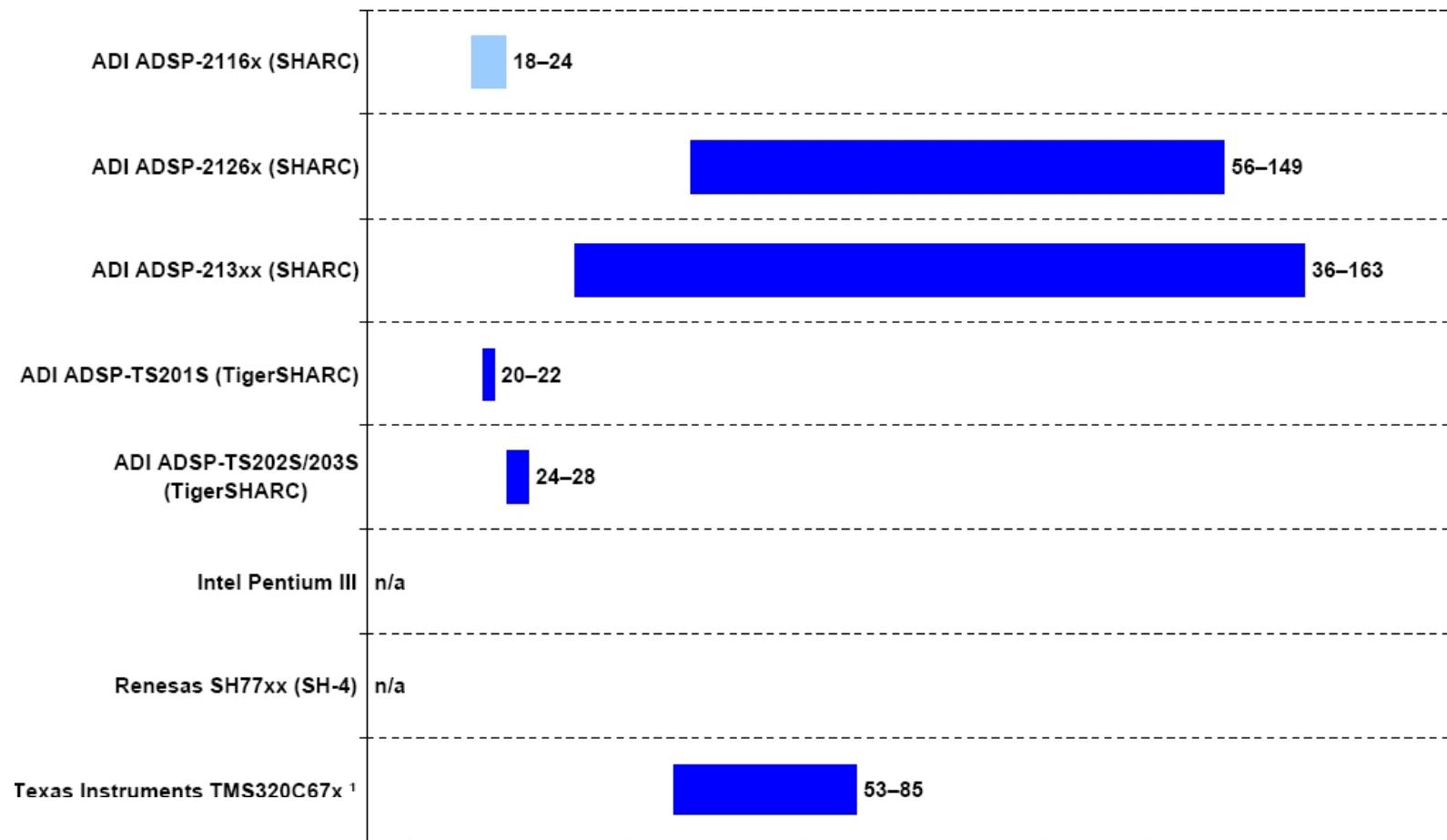
# Výkonové srovnání fixed-point DSP

## Z hlediska poměru výkon/cena



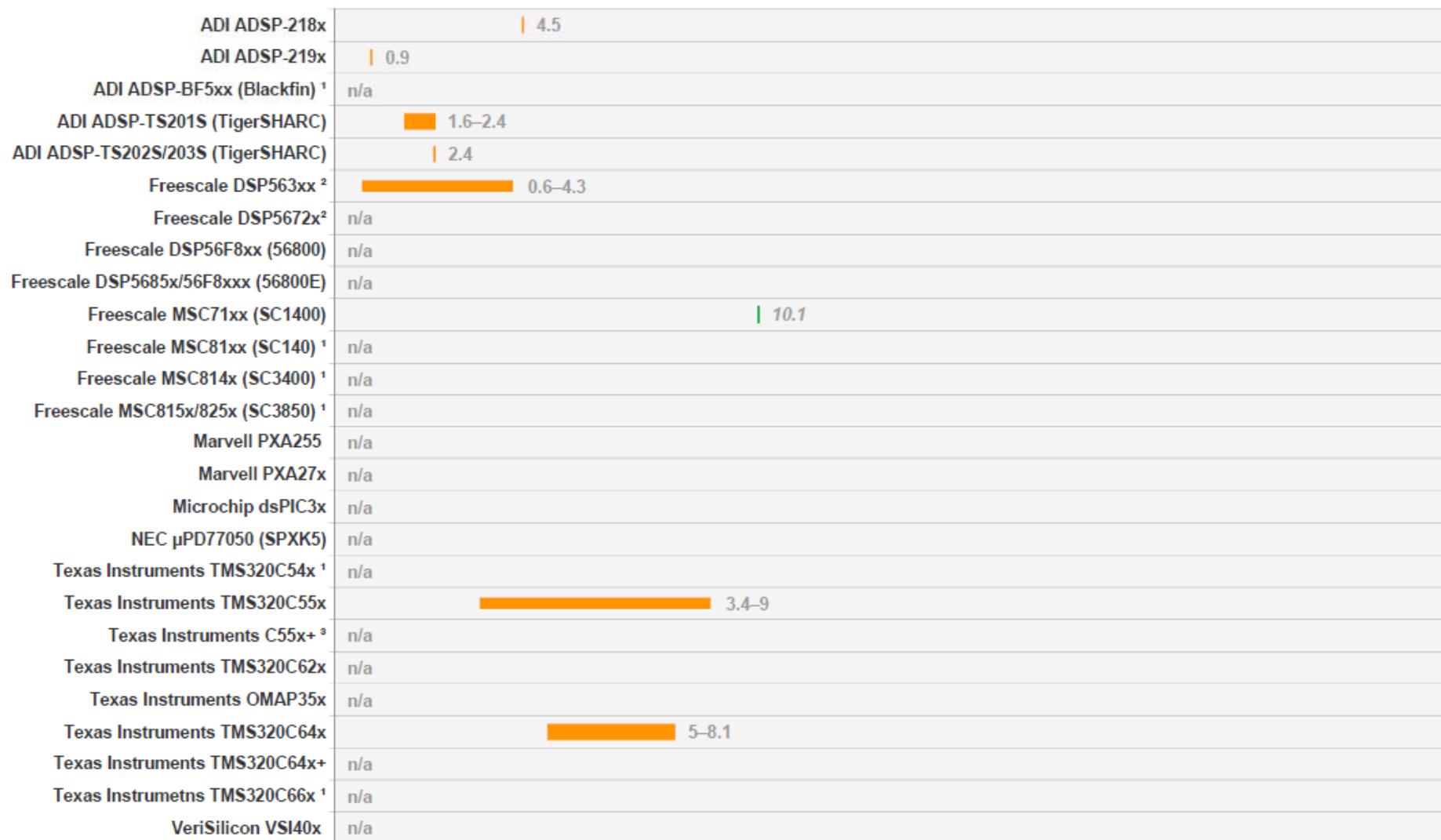
# Výkonové srovnání floating-point DSP

Z hlediska poměru výkon/cena



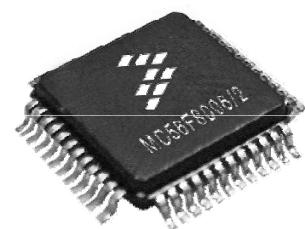
# Výkonové srovnání DSP

## Z hlediska poměru výkon/spotřeba



# Jak zvolit DSP pro danou aplikaci ?

- Volba není jednoduchá – existuje skutečně široké portfolio výrobců DSP a jednotlivých typů DSP
- Vždy záleží na konkrétní aplikaci – např. jaký je potřeba externí hardware (velikost a typ paměti, komunikační periférie, spotřeba atd.)
- Většina výrobců nabízí několik výkonových řad DSP
- Důležité faktory pro výběr konkrétního DSP:
  - ✓ Osobní zkušenosti vývojáře s daným DSP
  - ✓ Vývojové nástroje pro daný typ DSP
  - ✓ Cena celkového zařízení x tzv. time to market
  - ✓ Podpora ze strany výrobce



# Příklad asembleru pro ADSP 21XX

```
.section/pm interrupts;  
  
_reset: JUMP start; nop; nop; nop;  
        RTI; nop; nop; rti;  
        RTI; nop; nop; nop;  
/* 0x0000: Reset vector */  
/* 0x0004: IRQ2 */  
/* 0x0008: IRQL1 */  
/* 0x000C: IRQL0 */  
/* 0x0010: SPORT0 transmit */  
/* 0x0014: SPORT0 receive */  
/* 0x0018: IRQE */  
/* 0x001C: BDMA */  
/* 0x0020: SPORT1 transmit */  
/* 0x0024: SPORT1 receive */  
/* 0x0028: Timer */  
/* 0x002C: Power down */  
  
.section/pm program;  
  
start:  
    imask=0x001;  
    ena timer;  
  
again: toggle fl1;  
    cntr = 0x5ff;  
    do loop1 until ce;  
    cntr = 0xffff;  
    do loop2 until ce;  
    nop;  
loop2: nop;  
loop1: nop;  
jump again;
```

