M02: Timer, ADC & FPU

2.3. FPU

Prof. Rosa Zheng

References:

1.TM4C123GH6PM data sheet Chapter 3.

2. TM4C123 Workshop slides

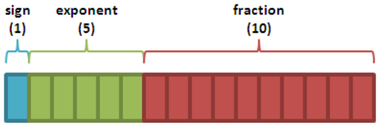
3. CMSIS DSP Library: TI Application Report spma041.pdf

4. CMSIS DSP library reference from ARM:

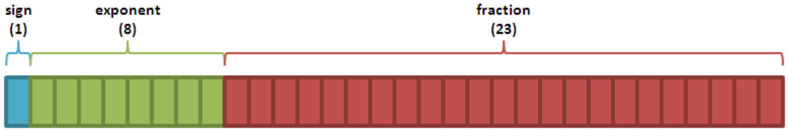
https://arm-software.github.io/CMSIS_5/DSP/html/modules.html

What is Floating-Point?

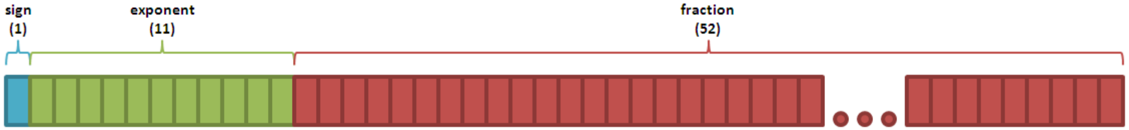
- **IEEE floating-point formats:**
  - **Half (16-bit)**



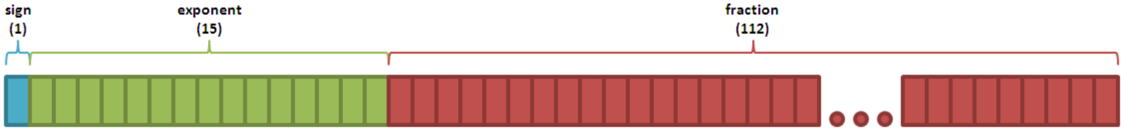
  - **Single (32-bit)**



  - **Double (64-bit)**



  - **Quadruple (128-bit)**



What is IEEE-754?

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	X
Symbol	Sign (s)	Exponent (e)								Fraction (f)																							

1 bit **8 bits** **23 bits**

$$\text{Decimal Value} = (-1)^s\,(1+f)\,2^{e-\text{bias}}$$

where: $f = \sum[(b_{-i})2^{-i}]\ \forall\, i \in (1,23)$

bias = 127 for single precision floating-point

Symbol	s	e								f																						
Example	0	1	0	0	0	0	1	1	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

sign $= (-1)^0$
$= [1]_{10}$

exponent $= [10000110]_2 = [134]_{10}$

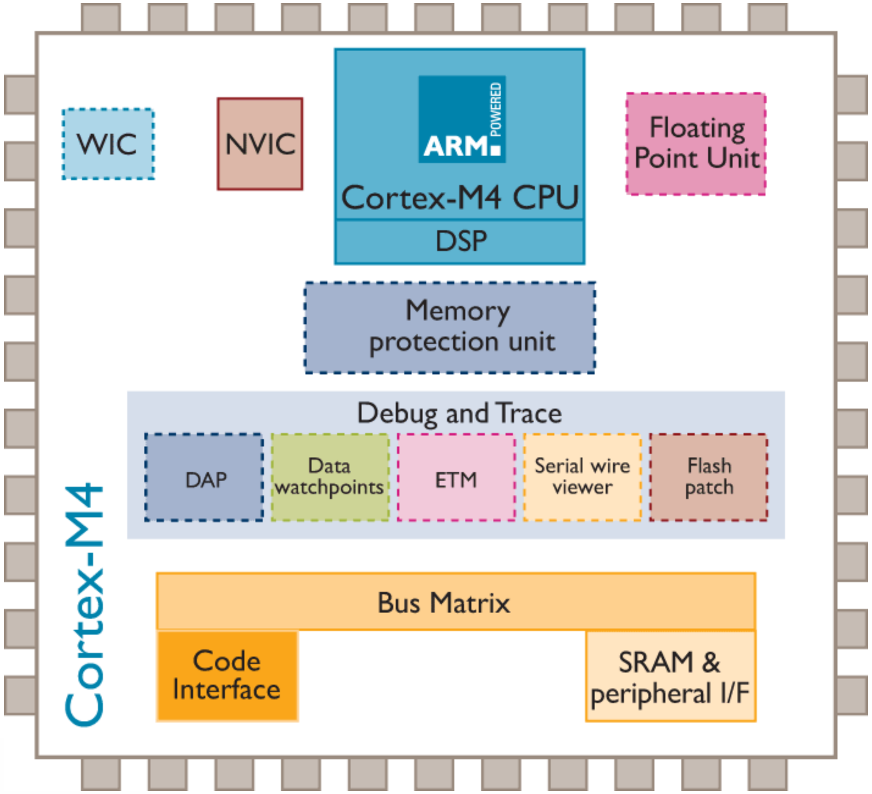
fraction $= [0.11010000100000000000000]_2 = [0.814453]_{10}$

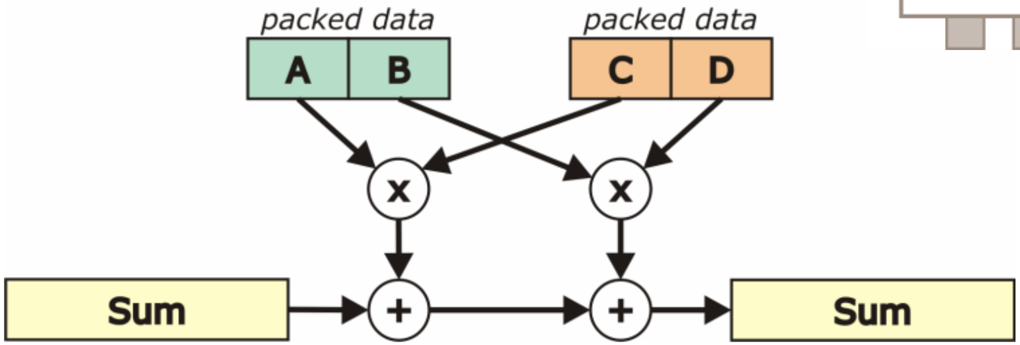
$$\begin{aligned}
\textit{Decimal Value} &= (-1)^s \text{ x } (1+f) \text{ x } 2^{e-\text{bias}} \\
&= [1]_{10} \text{ x } ([1]_{10} + [0.814453]_{10}) \text{ x } [2^{134-127}]_{10} \\
&= [1.\,814453]_{10} \text{ x } 128 \\
&= [232.249]_{10}
\end{aligned}$$

Floating-Point Unit (FPU)

The Cortex-M4F FPU fully supports single-precision:

- **Add**
- **Subtract**
- **Multiply**
- **Divide**
- **Single cycle multiply and accumulate (MAC)**
- **Square root**





- Sixteen 64-bit double-word registers, D0-D15
- Thirty-two 32-bit single-word registers, S0-S31

3 Modes of Operation

- **Full-Compliance mode** – In Full-Compliance mode, the FPU processes all operations according to the IEEE 754 standard in hardware. **No support code is required.**

- **Flush-to-Zero mode** – A result that is very small, as described in the IEEE 754 standard, where the destination precision is smaller in magnitude than the minimum normal value before rounding, is replaced with a zero.

- **Default NaN (not a number) mode** – In this mode, the result of any arithmetic data processing operation that involves an input NaN, or that generates a NaN result, returns the default NaN. (**0 / 0 = NaN**)

FPU Usage

- **The FPU is enabled automatically;**
- **Exceptions:** The FPU sets the cumulative exception status flag in the FPSCR register as required for each instruction. The FPU does not support user-mode traps.
- The processor can reduce the exception latency by using **lazy stacking***. This means that the processor reserves space on the stack for the FPU state, but does not save that state information to the stack.

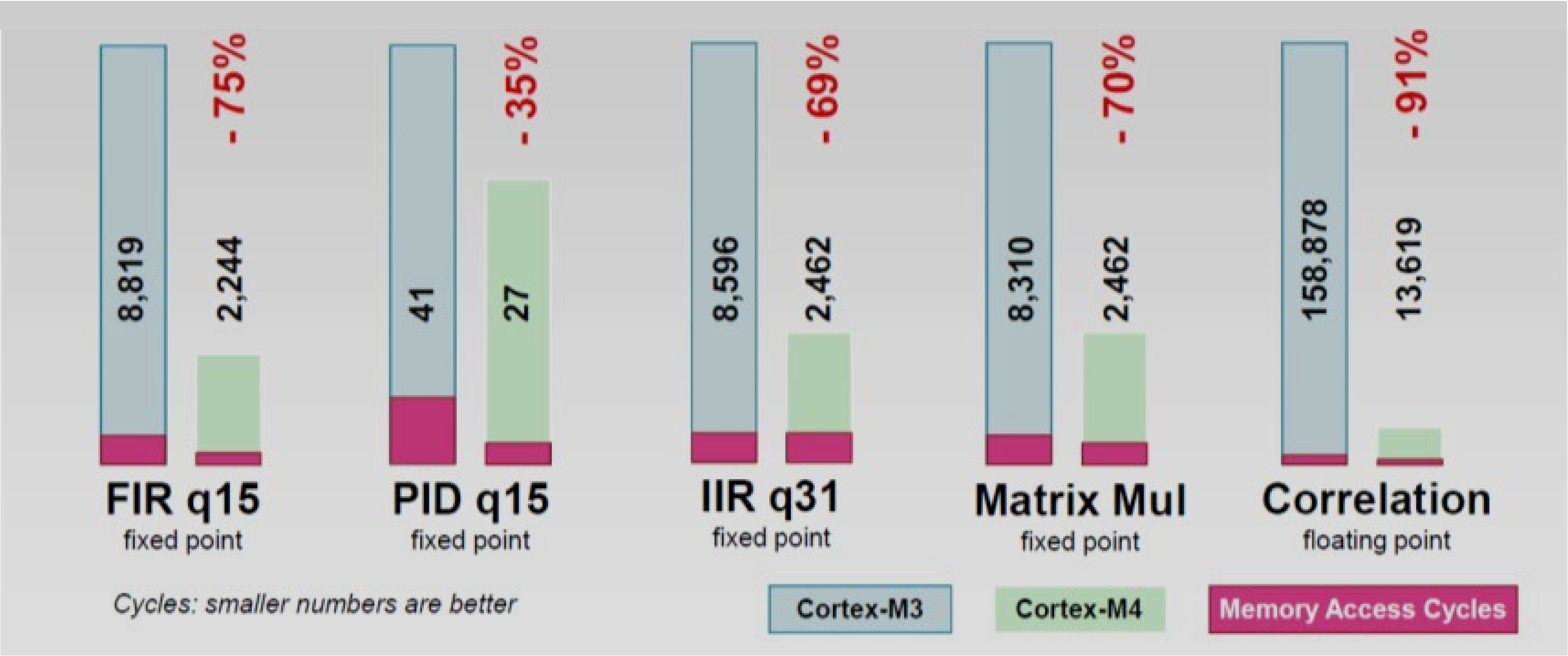
CMSIS* DSP Library Modules

- **Basic math functions:** Vector abs, add, dot, mult, shift
- **Fast math functions:** Sqrt, sin(), cos()
- **Complex math functions:** conj, dot, mult, mag, mag sq
- **Filters:** conv, IIR, FIR, NLMS, corr, interpolator, decimator
- **Matrix functions:** multiply, add, transpose, inverse, scale
- **Transforms:** complex FFT, DCT
- **Motor control functions:** PID, Clarke/Park Transform
- **Statistical functions:** min, max, power, rms, std, var
- **Support functions:** vector copy/fill, floating/integer convert
- **Interpolation functions:** linear/bilinear interpolation

Reference: https://arm-software.github.io/CMSIS_5/DSP/html/modules.html

CMSIS* DSP Library Performance

- Include a public header file arm_math.h
- Define the appropriate pre processor MACRO ARM_MATH_CM4


FIR q15
fixed point
PID q15
fixed point
IIR q31
fixed point
Matrix Mul
fixed point
Correlation
floating point
Cycles: smaller numbers are better
Cortex-M3
Cortex-M4
Memory Access Cycles