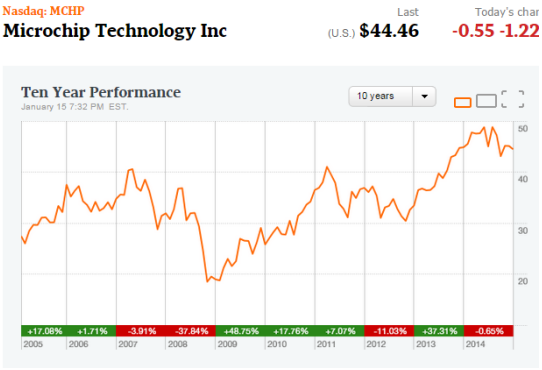
Choice of Processor

The platform used in a real-time embedded system (RTES) can be a processor[1], FPGA, ASIC, or a combination thereof. Often (not always), a processor is chosen. This document assumes a single processor is to be selected for a given application.

There are many criteria to consider when choosing a processor for an RTES. Some criteria are technical, some are non-technical. The following pages list and discuss a number of them.

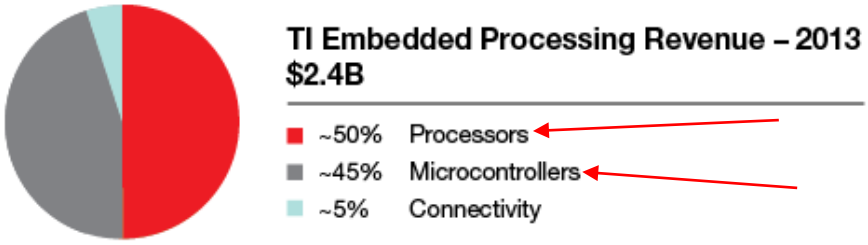
Non-Technical Criteria

1. Vendor (manufacturer) stability: - suggest choose a processor from a company that is going to be around for years to come, not some small start-up
    a. stock performance



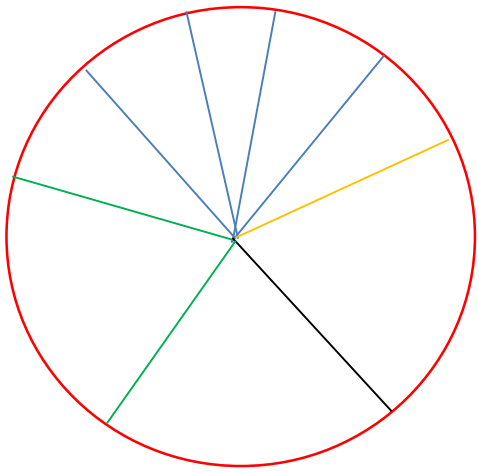
[1] Herein, the general term "processor" refers to any of the following: microprocessor, microcontroller, or digital signal processor.

b. market share pie chart

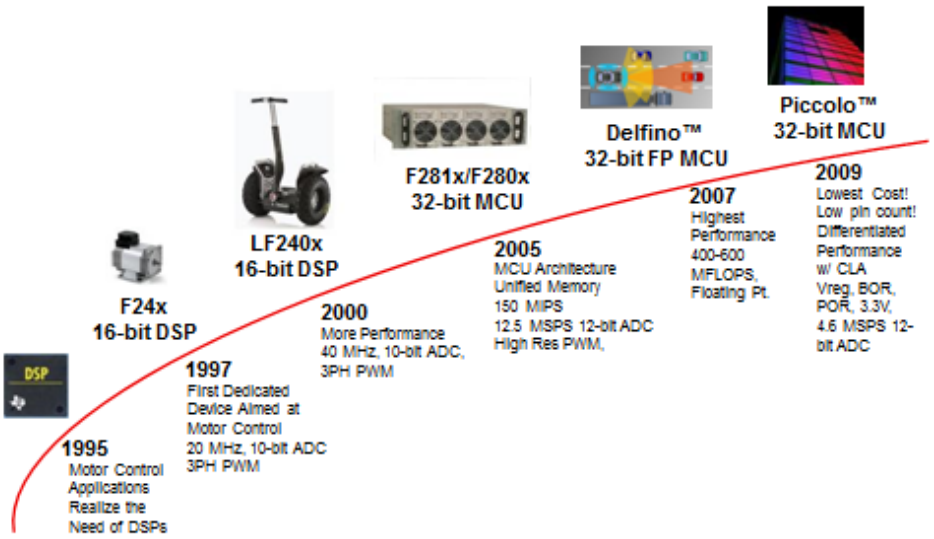


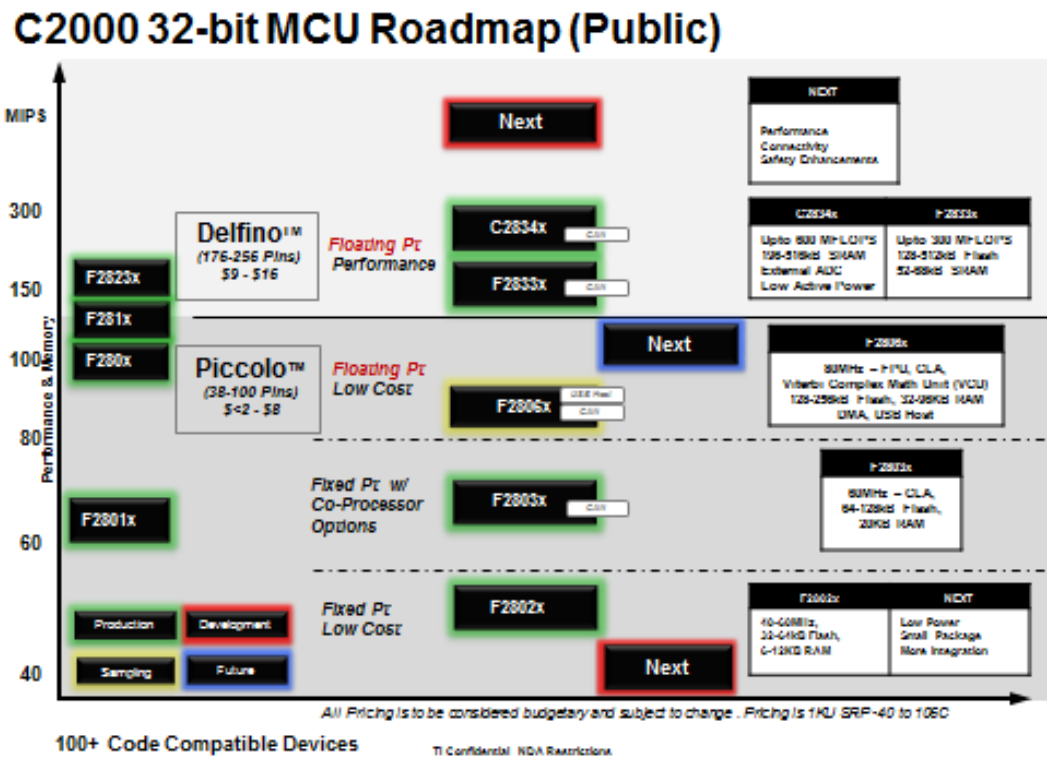
An actual pie chart showing market share of each company tends to be "confidential" information, but some information can be obtained by purchasing documents (on the order of $1,000's).

e.g. share of µC market



2. Product roadmap





3. Time-to-market issues:

    a. Maybe you are inheriting the processor, or your company simply has used it for years, then maybe you need to accept it as is (maybe upgrade).

    b. What have *you* used before, e.g. a particular vendor, a particular processor chip

Can you think of any other non-technical factors that might be criteria for choosing a processor?

Technical Criteria

Ubiquitous engineering trade-offs:

power vs **memory** vs **speed** vs **cost**

1. Speed - e.g. time to do a multiply-accumulate

   ➔ try to guestimate how fast the fastest process will be, e.g. sampling rate, and estimate what needs to be done within each sampling period - this can give you a handle on how fast the processor should be

2. Memory

   ➔ try to guestimate how much memory will be needed so you can choose a processor that has enough on-chip memory to save board area (for external memory) and design time and $

   ➔ there is program (code) size and data size to consider. Does your application collect a lot of data? e.g. lots of samples, past history, etc

3. power consumption

   a. if you need a rack of boards with many processors, low power might be attractive
   b. if you require battery operation, low power will likely be needed

4. fixed-point versus floating-point

   faster, cheaper, less power consumption higher dynamic range

5. I/O requirements

   a. number digital I/O lines
      - e.g. drive LEDs, read switches, etc

   b. number of A-D channels

   c. accuracy of A-D

   d. speed of A-D (sampling rate achievable)

   e. number of D-A channels

   f. accuracy of D-A

   g. speed of D-A (sampling rate achievable)

   h. serial communications

      - UART
      - SPI
      - I^2C
      - etc

   i. other
      - PWM
      - comparator

6. cost

    a. if small volume sold at high price, then price is less of an issue
        - e.g. substation electronics

    b. if large volume, then price is more of an issue
        - e.g. cordless phone

7. reliability

    a. fewer pins on chip → less solder joints!

    b. low power consumption means less heat generation
        - cooler operation implies higher reliability

    c. environment: operating temperature range (indoor versus outdoor)

8. possible future expansion of requirements, in the future may need…

    a. more speed
    b. more memory
    c. more ports
    d. etc

    …so consider them upfront when you first choose a processor!

9. development tools

    a. Integrated Development Environment (IDE) – bug-free, easy-to-use
    b. extensive 3rd party support
        - development boards
        - add-on boards
        - other

Can you think of any other technical factors that might be criteria for choosing a processor?