Seminar 1

Contents

- Definition of microcontroller (mC)
- Definition of Digital Signal Processor (DSP)
- Criteria for performance comparison of mCs e DSPs
- Performance measurements

Microcontrollers (mCs)

A microcontroller is a processor specifically designed and optimized to perform control, timing, supervising tasks on various target devices. It is characterized by the availability of relatively large amounts of "on chip" memory (ROM, EEPROM, Flash ...) and of several peripheral units, for different functions (I/O, A/D conversion, timer, counters, PWM, …). It is normally characterized by reduced complexity and low cost.

Microcontrollers (mCs)

Peripheral units in mCs:

- A/D converters (number of bit, conversion speed, linearity vary a lot among different devices)
- Timer and counters
- PWM modulators
- External memories (ROM, EEPROM, FLASH)
- Communicaion ports (serial, I2C, field bus e.g. CAN)

Microcontrollers (mCs)

The use of mCs is very common for the implementation of:

- portable measurement instruments;
- PC periferals;
- fax/photocopiers;
- home appliances;
- cell phones;
- industrial applications, in particular in the automotive and electrical drives fields.

Digital Signal Processors (DSPs)

DSPs are microprocessors specifically designed and optimized to efficiently perform real time signal processing tasks. They are characterized by high computational power and relatively low cost (if compared to general purpose processors). Particular care is taken in minimizing the power consumption (e.g. in embedded portable applications).

Digital Signal Processors (DSPs)

Several different DSP families are available on the market. They all exhibit some common features:

- availability of a built-in multiplier circuit (MAC instruction);
- capability to operate multiple memory accesses in a single clock cycle;
- specific addressing modes for circular registers and stacks;
- sophisticated program flow control instructions;
- availability of DMA circuitry (top level).

Digital Signal Processors (DSPs)

The major application areas for DSPs are related to:

- coding/decoding of speech, hi-fi audio segnals, video signal processing;
- compression/decompression of data;
- encryption/decryption of data;
- mixing of audio and video signals;
- sound synthesis.

DSPs vs mCs

Traditionally, mCs were used in the implementation of control functions, thanks to the wide range of peripheral units available on-chip. The computational power was limited (CPUs had 8 bits or less, no hardware multiplier).

DSPs were used, instead, almost only for signal-processing applications, where the key parameter is computational power.

Currently, the differences in the application fields of mCs and DSPs are a lot fuzzier.

DSPs vs mCs

More recent DSPs include peripheral units traditionally typical of mCs. On the other hand, mCs present, at least in top range models, hardware organizations and computational powers closer and closer to those typical of DSPs. Costs and performance may be very close and, for particular applications, the choice of the device may be quite difficult.

We definitely need criteria to compare different devices.

DSPs vs mCs

The fundamental parameters for the comparison are, of course, cost and performance.

To minimize the cost parameter, for given specifications, it is normally required to take into account not only the device cost, but also the estimated development time, the so called time to market.

DSPs vs mCs

The cost of device is largely dependent on the expected production volume.

Time and resources required by the development of the application are a function of several factors, like:

- availability of high quality and high reliability development tools;
- effective technical support from the device manufacturer.

DSPs vs mCs

The application specifications determine the performance level required for the selected microprocessor in terms of:

- required peripheral units and their basic parameters (e.g. A/D converter with 8, 10 or 12 bits);
- operating conditions (e.g. maximum allowable power consumption, temperature range);
- required computational power (real time control, signal processing …).

Performance measurement

The perfomance level of any processor can be measured only in terms of time required to excute a given program.

In the case of mCs or DSPs this is the same time the processor effectively spends on the program instructions (computation time), unless an operating system coordinating several tasks in time sharing is running on the device.

Estimation of computation time

The computation time of a program is a key parameter in real time applications (both in control and signal processing). This can be estimated based on three parameters:

- processor clock period;
- number of clock cycles required by the instructions in the program;
- number of di instructions required by the program.

Estimation of computation time

The clock period and the number of clock cycles required by the various program instructions can be read on the processor datasheet/user manual.

The number of instructions required by a given algorithm is a function of the processor architecture.

By architecture we mean the set of resources that are available to the programmer for the implementation of the algorithm (instruction set).

Estimation of computation time

Any given architecture can be implemented in different ways at the hardware level.

We therefore make a distinction between processor organization and architecture: the former is the particular hardware implementation of the latter.

The architecture has a direct effect on the number of instructions required by a given program. The organization determines the clock period and the number of clock cycles required by any instruction.

Estimation of computation time

The computation time of a program can be estimated by using the following formula:

$$T_{cal} = T_{clk} \cdot \sum_{i=1}^{N_{cl}} N_i \cdot NC_i \qquad (1)$$

where T_{clk} is the processor clock period, N_i is the number of class i instructions in the program, NC_i is the average number of clock cycles required by class i *instructions*, N_{cl} is the number of considered instruction classes.

Stimation of computation time

Relation (1) assumes that the program is not interrupted by other processes and neglects the delays due to memory accesses.

To increase the speed of a processor, we therefore need to:

- reduce the clock cycle (T_{clk});
- reduce the number of cycles required by the more commonly used instructions (NC);
- reduce the number of instructions required by a given algorithm.

Speed limits!

Reducing the clock cycle duration always implies the increase of power consumption for the processor.

This can be limited by reducing also the supply voltage.

Which tells us why there is a strong need for lower and lower power supply voltages (<1V) in computer applications.

The limitations are basically technological (we need new processes/materials).

Speed limits!

The reduction of the number of clock cycles required by an instruction calls for a more sophisticated hardware organization of the processor, e.g. wired control instead of micro-programmed control, higher degree of parallelism (achievable in several different ways: VLIW, SIMD, etc.) or the use of pipelines.

This trend leads to complex processors, with high cost. The limitation in this case is basically "economical".

Speed limits!

The number of instructions required by a given algorithm is a function of the processor architecture, i.e. of its instruction set, as seen by the programmer/compiler.

The reduction of this parameter leads to complex instruction set computers (CISC), instead of reduced (and simple) instruction set computers (RISC). This again affects the processor organization and its cost. That´s why RISC processors are a lot more used than CISC processors.

Maximizing performance

The processor performance is a function of both its architecture and of its organization, at the hardware level.

The maximization of performance calls for a co-ordinated design of hardware and software.

The problem is further complicated by the action of several design constraints such as:

- cost;
- electric power consumption.

Performance measurements

A typical way to measure a general purpose processor performance is to use benchmarking, i.e. the execution of suitably designed test programs. This method helps to evaluate the overall processor performance, including memory management.

Recently the same strategy is being applied also to DSPs. The considered test programs are typical signal processing algorithms (FFT, FIR, IIR filters etc.).

Performance measurements

The usual benchmarking method for general purpose processors uses complete applications (e.g. SPEC method).

This approach cannot be used with DSPs, because the results would be strongly dependent on the quality of the adopted compiler.

Besides, any code optimization becomes very difficult and comparison of different devices almost impossible. Kernel benchmarking is the adopted solution.

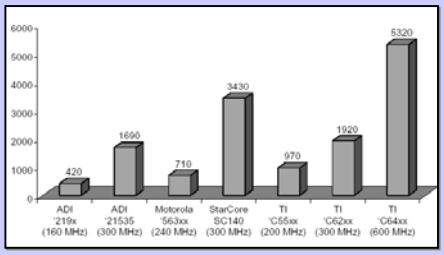
DSP Benchmarking

Benchmarking programs must have, at least, the following basic features:

1) relevance with respect to the typical DSP applications;
2) clear and precise definition (e.g. what type of DFT algorithm is considered);
3) simplicity;
4) optimization: they must be easy to optimize for any given DSP architecture.

DSP Benchmarking

Results of benchmarking tests are available on the market as reports edited by several specialized companies (e.g. BDTi). They are extremely expensive (>2000 Euros).



Performance indexes

The following list include the more commonly encountered perfomance indexes:

- MACS: MAC operations per second
- MIPS: millions of instructions per second
- MOPS: millions of operations per second
- FLOPS: millions of floating point operations per second

All the indexes give little information on the actual processor speed. They do not allow any comparison between different devices.

Performance indexes

The MACS index shows the maximum number of multiplies (with sum on the accumulator) a CPU is able to operate in a second (peak value).

However, in any DSP program, a lot of different operations are used (e.g. sums, memory read/write ...), whose impact on speed is not described by the index.

The MACS index does not give any serious measure of the actual processor speed.

Performance indexes

The MIPS index shows the maximum number of instructions a CPU is able to execute in a second (peak value).

But, the number of instructions required by any algorithm depends on the CPU architecture (a single instruction can perform different amounts of operations in different architectures) and also on the compiler quality.

It is, at most, only possible to compare devices sharing the same basic architecture.

Performance indexes

The MOPS index shows the maximum number of operations a CPU is able to execute in a second (peak value).

Its definition is ambiguous itself, because it does not clearly define what exactly is an operation, or at least what is the reference set of operations, if any.

The meaning of the index is therefore not very clear and this should not be considered for comparison purposes.

Performance indexes

The FLOPS index shows the maximum number of floating point operations a CPU is able to execute in a second (peak value).

The validity of this index is similar to that of the MOPS index, with the further limitation that it can be applied only to floating point architectures.

None of the presented indexes takes into account other key issues for CPU performance measurement, like, for example, memory management and organization.

Example: MIPS vs Computation Time

Instruction classes	Average number of clock cycles (NC)
A	1
B	2
C	3

A processor has 3 different instruction classes. Each class requires a different number of clock cycles. Any given program will use a certain amount of instructions of each class (compiler dependent).

Example: MIPS vs Computation Time

Compiler / Programmer	Number of instructions per class (hundreds)		
	A	B	C
1	5	1	1
2	10	1	1

Two different compilers/programmers produce two different programs for the same algorithm, using a different number of instructions and a different distribution among the three classes.

Example: MIPS vs Computation Time

Applying (1) we can now evaluate the computation time of the two programs. We find:

$$T_{cal1} = T_{clk} \cdot 100 \cdot (5 \cdot 1 + 1 \cdot 2 + 1 \cdot 3) = 1000 \cdot T_{clk}$$

$$T_{cal2} = T_{clk} \cdot 100 \cdot (10 \cdot 1 + 1 \cdot 2 + 1 \cdot 3) = 1500 \cdot T_{clk}$$

The second program has a computation time longer than the first by 50%. It is then much slower the the first.

Example: MIPS vs Computation Time

If we compute the MIPS index for the two programs we find:

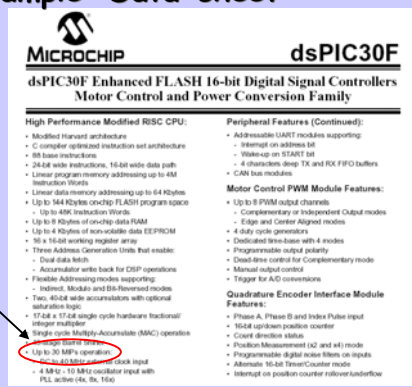
$$MIPS_1 = 10^{-6} \cdot 100 \cdot (5+1+1)/(1000 \cdot T_{clk}) = 0.7 \cdot F_{clk} \cdot 10^{-6}$$

$$MIPS_2 = 10^{-6} \cdot 100 \cdot (10+1+1)/(1500 \cdot T_{clk}) = 0.8 \cdot F_{clk} \cdot 10^{-6}$$

The second program has a higher MIPS rate compared to the first. According to this index the second program should be faster!

Example: Data-sheet

The manufacturer claims a 30 MIPS (F_{clk} = 40 MHz) operating speed.


MICROCHIP dsPIC30F

dsPIC30F Enhanced FLASH 16-bit Digital Signal Controllers
Motor Control and Power Conversion Family

High Performance Modified RISC CPU:
- Modified Harvard architecture
- C compiler optimized instruction set architecture
- 84 base instructions
- 24-bit wide instructions, 16-bit wide data path
- Linear program memory addressing up to 4M Instruction Words
- Linear data memory addressing up to 64 Kbytes
- Up to 144 Kbytes on-chip FLASH program space
  - Up to 48K Instruction Words
- Up to 8 Kbytes of on-chip data RAM
- Up to 4 Kbytes of non-volatile data EEPROM
- 16 x 16-bit working register array
- Three Address Generation Units that enable:
  - Dual data fetch
  - Accumulator write back for DSP operations
- Flexible Addressing modes supporting:
  - Indirect, Modulo and Bit-Reversed modes
- Two, 40-bit wide accumulators with optional saturation logic
- 17-bit x 17-bit single cycle hardware fractional/integer multiplier
- Single cycle Multiply-Accumulate (MAC) operation
- Up to 30 MIPS operation:
  - DC - 40 MHz external clock input
  - 4 MHz - 10 MHz oscillator input with PLL active (4x, 8x, 16x)

Peripheral Features (Continued):
- Addressable UART modules supporting:
  - Interrupt on address bit
  - Wake-up on START bit
  - 4 characters deep TX and RX FIFO buffers
- CAN bus modules

Motor Control PWM Module Features:
- Up to 8 PWM output channels
  - Complementary or Independent Output modes
  - Edge and Center Aligned modes
- 4 duty cycle generators
- Dedicated time-base with 4 modes
- Programmable output polarity
- Dead-time control for Complementary mode
- Manual output control
- Trigger for A/D conversions

Quadrature Encoder Interface Module Features:
- Phase A, Phase B and Index Pulse input
- 16-bit up/down position counter
- Count direction status
- Position Measurement (x2 and x4) mode
- Programmable digital noise filters on inputs
- Alternate 16-bit Timer/Counter mode
- Interrupt on position counter rollover/underflow

Example: Data-sheet

The manufacturer claims a maximum speed of 8000 MIPS! But here F_{clk} = 1000 MHz!

TMS320C6414T, TMS32
FIXED-POINT DIGITA

- Highest-Performance Fixed-Point Digital Signal Processors (DSPs)
  - 1.67-, 1.39-, 1-ns Instruction Cycle Time
  - 600-, 720-MHz, 1-GHz Clock Rate
  - Eight 32-Bit Instructions/Cycle
  - Twenty-Eight Operations/Cycle
  - 4800, 5760, 8000 MIPS
  - Fully Software-Compatible With C62x™
  - C6414/15/16 Devices Pin-Compatible
- VelociTI.2™ Extensions to VelociTI™ Advanced Very-Long-Instruction-Word (VLIW) TMS320C64x™ DSP Core
  - Eight Highly Independent Functional Units With VelociTI.2™ Extensions:
    - Six ALUs (32-/40-Bit), Each Supports Single 32-Bit, Dual 16-Bit, or Quad 8-Bit Arithmetic per Clock Cycle
    - Two Multipliers Support Four 16 x 16-Bit Multiplies (32-Bit Results) per Clock Cycle or Eight 8 x 8-Bit Multiplies
- Two External Mer
  - One 64-Bit (EM
  - Glueless Interf. Memories (SRA Synchronous M SBSRAM, ZBT
  - 1280M-Byte To Memory Space
- Enhanced Direct- Controller (64 Ind
- Host-Port Interfac
  - User-Configura
- 32-Bit/33-MHz, Interface Conforr [C6415T/C6416T]
  - Three PCI Bus Prefetchable Non-Prefetcl
  - Four-Wire Seria
  - PCI Interrupt R Program Contr

Example: Data-sheet

The manufacturer claims a maximum speed of 40 MIPS (F_{clk} = 80 MHz).

MOTOROLA

Technical Data

56F801 16-bit Hybrid Controller

- Up to 30 MIPS operation at 60MHz core frequency
- Up to 40 MIPS operation at 80MHz core frequency
- DSP and MCU functionality in a unified, C-efficient architecture
- MCU-friendly instruction set supports both DSP and controller functions: MAC, bit manipulation unit, 14 addressing modes
- Hardware DO and REP loops
- 6-channel PWM Module
- 8K × 16
- 1K × 16
- 2K × 16
- 1K × 16
- 2K × 16
- Serial P
- General
- JTAG/C
- On-chip