Programmable Digital Signal Processors

Real-Time Signal Processing
Spring 2010
Common features between DSP Processors

- Most DSP processors share some common basic features designed to support high-performance, repetitive, numerically intensive tasks.

- The most important feature of a DSP is the ability of perform a Multiply-Accumulation operation in a single instruction cycle.
  - MAC operation allows the execution of vector-dot product that are employed in:
    - Digital filters,
    - Correlation,
    - Fourier transforms.
MAC into Datapath

- DSP processors integrate multiply-accumulate hardware into the main datapath of the processor to allow a series of multiply-accumulate operations to proceed without the possibility of arithmetic overflow, DSP processors generally provide extra “guard” bits in the accumulator.
Common features between DSP Processors

- A second feature shared by DSP processors is the ability to complete several accesses to memory in a single instruction cycle.

- This allows the processor to fetch an instruction while simultaneously fetching operands and/or storing the result of a previous instruction to memory.
Common features between DSP Processors

- A third feature often used to speed arithmetic processing on DSP processors is one or more dedicated address generation units.

- Once the appropriate addressing registers have been configured, the address generation unit operates in the background.

- Addressing modes
  - register-indirect addressing with post-increment
  - Modulo addressing (Circular buffer)
  - Bit-reversed addressing
Summary of Common features

- MAC specific hardware introduced in the datapath
- Fast memory access by having several independent memories for instruction and data
- Addressing unit that can execute in parallel with the datapath
- Most DSP processors provide special support for efficient looping
- Most DSP processors incorporate one or more serial or parallel I/O interfaces, and specialized I/O handling mechanisms such as low-overhead interrupts and direct memory access (DMA) to allow data transfers to proceed with little or no intervention from the rest of the processor
How to choose a DSP Processor?

- The first task for the designer selecting a DSP processor is to weigh the relative importance of

  - performance,
  - cost,
  - Integration,
  - ease of development,
  - power consumption, and
  - other factors for the application at hand.
Applications for DSP Processors

- Inexpensive, high-volume embedded systems
  - Cellular phone
  - MP3 players
  - Disk drivers
  - Portable Digital Audio Players

- Processing large volumes of data with complex algorithms for specialized needs
  - Sonar and seismic exploration
  - Image processing
  - Digital Camera
Choosing the Right DSP Processor

Parameters to consider:
- Arithmetic Format
- Data Width
- Speed
- Memory Organization
- Ease of Development
- Multiprocessor Support
- Power Consumption and Management
- Cost
Arithmetic Format

### Fixed-point

- Most DSPs use *fixed point* arithmetic, where numbers are represented as integers or as fractions in a fixed range.

- On a fixed-point processor, programmers often must carefully scale signals at various stages of their programs to ensure adequate numeric precision with the limited dynamic range.

### Floating-point

- Some DSP use *floating-point arithmetic*, where values are represented by a *mantissa* and an *exponent*.

- Wider dynamic range
- More flexible
- DSP are easier to program
- More expensive and have higher power consumption
Arithmetic Format

- Most high-volume, embedded applications use fixed point processors because the priority is on low cost and, often, low power.

- Programmers and algorithm designers determine the dynamic range and precision needs of their application, either analytically or through simulation.

- For applications that have extremely demanding dynamic range and precision requirements, or where ease of development is more important than unit cost, floating-point processors have the advantage.
Data Width

- All common floating-point DSPs use a 32-bit data word.

- For fixed-point DSPs, the most common data word size is 16 bits.
  - Motorola’s DSP563xx family uses a 24-bit data word,
  - Zoran’s ZR3800x family uses a 20-bit data word.

- The size of the data word has a major impact on cost, because it strongly influences
  - the size of the chip and
  - the number of package pins required,
  - as well as the size of external memory devices connected to the DSP
Data Width

- If the bulk of an application can be handled with single-precision arithmetic, but the application needs more precision for a small section of the code,

- Selective use of double-precision arithmetic may make sense

- If most of the application requires more precision, a processor with a larger data word size is likely to be a better choice.
Speed

- A key measure of the suitability of a processor for a particular application is its execution speed.

- The most fundamental measure of speed is the processor’s instruction cycle time:
  - The amount of time required to execute the fastest instruction on the processor.

- MIPS is the reciprocal of the instruction cycle time divided by one million and multiplied by the number of instructions executed per cycle is the processor’s peak instruction execution rate in millions of instructions per second.
Speed

- However, MIPS do not account for the differences in architecture (VLIW, Multiprocessor, Uniprocessor, etc.)
- Instruction set

Therefore, MIPS is an inaccurate measurement that does not allow a fair comparison between different DSP Processors.

One solution to these problems is to decide on a basic operation (instead of an instruction) and use it as a yardstick when comparing processors.
Speed

- A more general approach is to define a set of standard benchmarks and compare their execution speeds on different DSPs.

- These benchmarks may be simple algorithm “kernel” functions (such as FIR or IIR filters), or they might be entire applications or portions of applications (such as speech coders).

- A DSP’s input clock may be the same frequency as the processor’s instruction rate, or it may be two to four times higher than the instruction rate.
Memory Organization

- Fast MAC execution requires fetching an instruction word and two data words from memory at an effective rate of once every instruction cycle.

- Multiported memories allow multiple memory accesses per instruction cycle.

- Instruction caches allow instructions to be fetched from cache instead of from memory, thus freeing a memory access to be used to fetch data.
Memory Organization

- Another concern is the size of the supported memory, both on- and off-chip

- Most fixed-point DSPs are aimed at the embedded systems market, where memory needs tend to be small.

- These processors typically have small-to-medium on-chip memories (between 4K and 64K words), and small external data buses

- Most fixed-point DSPs feature address buses of 16 bits or less, limiting the amount of easily-accessible external memory.

- Some floating-point chips provide relatively little (or no) on-chip memory, but feature large external data buses.

- As with most DSP features, the best combination of memory organization, size, and number of external buses is heavily application-dependent.
Ease of Development

- Engineers performing research or prototyping will probably require tools that make system development as simple as possible.

- On the other hand, a company developing a next-generation digital cellular telephone may be willing to suffer with poor development tools and an arduous development environment if the DSP chip selected shaves $5 off the cost of the end product.
Ease of Development

Software

• assemblers, linkers,
• simulators, debuggers,
• compilers, code libraries,
• real-time operating systems

Hardware

• Development boards
• Emulators

Higher-level tools

• block-diagram based code-generation environments
Ease of Development

- A fundamental question to ask when choosing a DSP is how the chip will be programmed.

- Most DSP are programmed in assembly language because programmers can be forced to hand-optimize assembly code to lower execution time and code size to acceptable levels.
Ease of Development

- Users of high-level language compilers often find that the compilers work better for floating-point DSPs than for fixed-point DSPs, for several reasons.

- Most high-level languages do not have native support for fractional arithmetic.

- Floating-point processors tend to feature more regular, less restrictive instruction sets than smaller, fixed-point processors, and are thus better compiler targets.

- Floating-point processors typically support larger memory spaces than fixed-point processors, and are thus better able to accommodate compiler-generated code, which tends to be larger than hand crafted assembly code.
Ease of Development

- VLIW-based DSP processors are better compiler targets than traditional DSP processors because:
  - Typically use simple, orthogonal RISC-based instruction sets
  - and have large register files.

- However, even compilers for VLIW processors tend to generate code that is inefficient in comparison to hand-optimized assembly code.
Ease of Development

- Another tools that need close attention are:
  - Debuggers
  - Hardware Emulators Tools

- Almost all manufacturers provide instruction set simulators, which can be a tremendous help in debugging programs before hardware is ready.

- If using a high-level language compiler, then
  - will it run with the simulator and/or the hardware emulator?
  - Is it a separate program from the assembly-level debugger that requires the user to learn another user interface?
Power Consumption and Management

- DSPs are increasingly being used in portable applications where power consumption is a major concern.

- Many processor vendors are reducing processor supply voltages and adding power management features.

- Power Management Features include:
  - Reduced voltage operation. Many vendors offer low-voltage (3.3-, 2.5-, or 1.8-volt) versions of their DSP processors.
  - “Sleep” or “idle” modes. Most DSPs feature modes that turn off the processor’s clock to certain sections of the processor, reducing power consumption.
Power Consumption and Management

- **Programmable clock dividers.** Some DSPs allow the processor’s clock frequency to be varied under software control to use the minimum clock speed required for a particular task.

- **Peripheral control.** Some DSPs allow the programmer to disable peripherals that are not in use.

Regardless of power management features, it is often difficult for design engineers to obtain meaningful power consumption figures for DSPs.

This is because a DSP’s power consumption may vary by as much as a factor of three depending on the instructions it executes.
Finally, when considering prices, it is important to remember,

First, processor prices are continually falling.

Second, prices are strongly dependent on quantity.

Price of DSP is always related to characteristic of the processor such as:

- On-chip memory,
- Performance
- Flexibility
- Ease of Development
DSP Kit

32-bit floating-point DSP from Texas Instruments
TMS320C67x Family

- The TMS320C67x family is derived from the TMS320C62x family of 16-bit fixed-point DSP processors

- The TMS320C67x support
  - all the instruction set of TMS320C62x,
  - IEEE 754 32-bit single precision and 64-bit double precision arithmetic
  - and 32-bit fixed-point arithmetic

- Applications targeted
  - Home audio
  - 3D graphics
  - Medical Imaging
  - Radar
  - Speech recognition
Architecture

- The TMS320C67x contains
  - Two floating-point data paths
  - Each data path contains 2 ALUs, a multiplier and an adder/subtractor for address generation
  - The ALU support both integer and floating-point operations
  - The multiplier can perform 16 16 and 32 32 integer multipliers as well as 32-bit and 64-bit floating point multiplies
  - Each data path contains a register file of sixteen 32-bit general purpose registers
  - To support 64-bit floating-point arithmetic, pairs of adjacent registers can be used to hold 64-bit data
The memory system implement a modified Harvard architecture providing separate address space for instruction and data.

- The TMS320C67x uses a
  - 32-bit address bus for fetching instructions with a 256-bit data bus
  - 32-bit address bus for fetching data with a 64-bit data bus

- TMS320C67x allows up to two data move instructions to be executed in parallel with other instructions
Addressing

- The supported addressing modes are:
  - Register-direct
  - Register-indirect
    - The address register modification options include pre-increment/decrement by a short immediate or by the contents of any register
  - Immediate data
  - Modulo addressing
Pipeline

- The TMS320C67x pipeline consists of 16 stages.
- Instructions are always fetched eight at a time via the 256-bit instruction bus.
- The group of 8-instructions are called “fetch packets”
- The TMS320C67x support variable-length execution packets, then can execute from one to eight instructions in parallel
- The processor does not check execution packets for resource contention.
- Consequently, hand-written assembly code may introduce resource conflicts that produce unwanted behavior
Instruction Set

- The instruction set is a RISC 32-bit instruction width, uniform register sets, and extremely regular.
- The TMS320C67x is highly parallel architecture, obtaining maximum performance often requires the programmer to schedule instructions carefully.
- This can be a challenge due to the complex architecture and long, variable instructions latencies.
- Texas Instruments assembly optimizer tools and C compiler simplify code development by automating the scheduling and parallelization processes.
Benchmark

Execution Time (Microseconds)

ADSP-21161N (100MHz) | TMS320C6713 (225MHz) | Renesas 240 MHZ
Benchmark

Energy Consumption for 256-FFT

- **ADSP-21161N (100MHz, 1.8V)**
- **TMS320C6713 (200MHz, 1.2V)**
- **Renesas (200 MHZ, 1.5V)**
Conclusion

- TMS320C67x achieves a very good processing speed, particularly considering its low cost.
- It also has better cost-performance results than its closest competitors.
- Also, it offers support for double-precision processing at speed cost.
- The TMS320C67x benefits form well-supported development infrastructure of Texas Instrument. However, the family is relatively difficult to program at assembly language level.
- TMS320C67x benefits from extensive third party support including development boards, emulators, applications board and software libraries.