A Credit Seminar Report
On

Digital Signal Processors: Current Trends and Architectures

By:

Bhavik Shah
(06307907)

Guided by:
Prof. V. M. Gadre
Department of Electrical Engineering,
IIT Bombay
Acknowledgements

I would like to express my deep gratitude to Prof. V. M. Gadre for his profound guidance and support which helped me understanding the nuances of the seminar work. I am thankful to him for his timely suggestions which helped me a lot in completion of this report.

I would also like to extend my sincere thanks to all the members of the TI-DSP lab for their help and support.

Bhavik Shah
14th November, 2006.
Abstract

Digital signal processing is one of the core technologies in rapidly growing application areas such as wireless communications, audio and video processing and industrial control. The report presents an overview of DSP processors and the trends and recent developments in the field of Digital Signal Processors. The report also discusses the architectural features of the two different kinds of DSPs one floating-point and one fixed-point.

The first chapter presents the differences of traditional microprocessor and a DSP. It also presents some common architectural features shared by different DSPs for an efficient processing of DSP algorithms.

The second chapter mainly deals with the recent changes in the DSP architectures, the evolution and trends in DSP processor architecture. The new generation of processors with VLIW and superscalar structures and processors with hybrid structures are studied in brief here.

The fourth and fifth chapters take a review of the two different kind of digital signal processors namely i.e. floating-point and fixed-point processors.

The concluding chapter takes a review of the report and gives a look to further trends.
Contents

1. Introduction and Overview of Digital Signal Processors 1
   1.1 Introduction 1
   1.2 Difference between DSPs and other Microprocessors 1
   1.3 Important features of DSPs 2
      1.3.1 MACs and Multiple execution units 2
      1.3.2 Circular buffering 3
      1.3.4 Dedicated Address Generation Units 4
      1.3.5 Zero overhead looping 4
      1.3.6 Data formats 4
      1.3.7 Specialized Instruction Set 4

2. Digital Signal Processors: Trends and Developments 6
   2.1 First generation conventional 6
   2.2 Second generation Enhanced conventional 7
   2.3 Third generation Novel design 8
   2.4 Power consideration 9
   2.5 Fourth generation Hybrids 9
   2.6 Benchmarking DSPs 10

3. Architecture and Peripheral of TMS320C67x 12
   3.1 Introduction 12
   3.2 Architecture of C67xx 13
      3.2.1 Central Processing Unit 13
      3.2.2 General purpose Register files 14
      3.2.3 Functional Units 14
      3.2.4 Memory System 14
   3.3 Peripherals of TMS32067xx 16
      3.3.1 Enhanced DMA 16
      3.3.2 Host Port Interface 17
      3.3.3 External Memory Interface 17
      3.3.4 Multichannel Buffered Serial Port 18
      3.3.5 Timers 19
      3.3.6 Multichannel Audio Serial Port 20
      3.3.7 Power Down Logic 20

4. Overview of Fixed-point processor TMS320C55x 21
   4.1 Introduction 21
   4.2 Architectural Features 21
   4.3 Low Power Enhancements 24
      4.3.1 Parallelism 24
      4.3.2 Alternate Computational Hardware 24
4.3.3 Memory Access 24
4.3.4 Automatic Power Management 25
4.3.5 Power Down Flexibility 25

4.4 Peripherals of TMS320C55x 25
  4.4.1 Clock Generator with PLL 26
  4.4.2 DMA Controller 26
  4.4.3 Host Port Interface 27

5. Conclusion 28

References 29
Chapter 1

Introduction and
Overview of Digital Signal Processors

1.1 Introduction

Digital signal processing is one of the core technologies, in rapidly growing application areas, such as wireless communications, audio and video processing and industrial control. The number and variety of products that include some form of digital signal processing has grown dramatically over the last few years. DSP has become a key component, in many of the consumer, communications, medical and industrial products which implement the signal processing using microprocessors, Field Programmable Gate Arrays (FPGAs), Custom ICs etc.

Due to increasing popularity of the above mentioned applications, the variety of the DSP-capable processors has expanded greatly. DSPs are processors or microcomputers whose hardware, software, and instruction sets are optimized for high-speed numeric processing applications, an essential for processing digital data, representing analog signals in real time. The DSP processors have gained increased popularity because of the various advantages like reprogram ability in the field, cost-effectiveness, speed, energy efficiency etc.

1.2 Difference between DSPs and Other Microprocessors

Over the past few years it is seen that general purpose computers are capable of performing two major tasks.

(1) Data Manipulation, and
(2) Mathematical Calculations

All the microprocessors are capable of doing these tasks but it is difficult to make a device which can perform both the functions optimally, because of the involved technical trade offs like the size of the instruction set, how interrupts are handled etc. As a broad generalization these factors have made traditional microprocessors such as Pentium
Series, primarily directed at data manipulation. Similarly DSPs are designed to perform the mathematical calculations needed in Digital Signal Processing, [1].

Data manipulation involves storing and sorting of information. For instance, a word processing program does a basic task of storing, organizing and retrieving of the information. This is achieved by moving data from one location to another and testing for inequalities (A=B, A<B etc.). While mathematics is occasionally used in this type of application, it is infrequent and does not significantly affect the overall execution speed. In comparison to this, the execution speed of most of the DSP algorithms is limited almost completely by the number of multiplications and additions required.

In addition to performing mathematical calculations very rapidly, DSPs must also have a predictable execution time, [1]. Most DSPs are used in applications where the processing is continuous, not having a defined start or end. The cost, power consumption, design difficulty etc increase along with the execution speed, which makes an accurate knowledge of the execution time, critical for selecting proper device, as well as algorithms that can be applied. DSPs can also perform the tasks in parallel instead of serial in case of traditional microprocessors.

<table>
<thead>
<tr>
<th>Typical Applications</th>
<th>Data Manipulation</th>
<th>Math Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Word processing, database management, spread sheets, operating systems, etc.</td>
<td>Digital Signal Processing, motion control, scientific and engineering simulations, etc.</td>
</tr>
<tr>
<td>Main Operations</td>
<td>data movement ((A \rightarrow B))</td>
<td>addition ((A+B=C))</td>
</tr>
<tr>
<td></td>
<td>value testing ((If A=B then ...))</td>
<td>multiplication ((A \times B=C))</td>
</tr>
</tbody>
</table>

*Figure 1.1: Data Manipulation Vs. Mathematical calculation.*

### 1.3 Important features of DSPs

As the DSP processors are designed and optimized for implementation of various DSP algorithms, most processors share various common features to support the high performance, repetitive, numeric intensive tasks.

#### 1.3.1 MACs and Multiple Execution Units

The most commonly known and used feature of a DSP processor is the ability to perform one or more multiply-accumulate

*Figure1.2: MAC units are helpful in such FIR filters*
operation (also called as “MACs”) in a single instruction cycle. The MAC operation is useful in DSP algorithms that involve computing a vector dot product, such as digital filters, correlation, and Fourier transforms. The MAC operation becomes useful as the DSP applications typically have very high computational requirements in comparison to other types of computing tasks, since they often must execute DSP algorithms (such as FIR filtering) in real time on lengthy segments of signals sampled at 10-100 KHz or higher. To facilitate this DSP processors often include several independent execution units that are capable of operating in parallel.

1.3.2 Efficient Memory Access

DSP processors also share a feature of efficient memory access i.e. the ability to complete several accesses to memory in a single instruction cycle. Due to Harvard architecture in DSPs, i.e. physically separate storage and signal pathways for instructions and data, and pipelined structure the processor is able to fetch an instruction while simultaneously fetching operands and/or storing the result of previous instruction to memory. In some recently available DSPs a further optimization is done by including a small bank of RAM near the processor core, often termed as L1 memory, which is used as an instruction cache. When a small group of instructions is executed repeatedly, the cache is loaded with these instructions thus making the bus available for data fetches, instead of instruction fetches.

1.3.3 Circular Buffering

The need of processing the digital signals in real time, where in the output (processed samples) have to be produced at the same time at which the input samples are being acquired, evolves the concept of Circular Buffering. For instance this is needed in telephone communication, hearing aids, radars etc. Circular buffers are used to store the most recent values of a continually updated signal. Circular buffering allows processors to access a block of data sequentially and then automatically wrap around to the beginning address exactly the pattern used to access coefficients in FIR filter. Circular buffering also very helpful in implementing first-in, first-out buffers, commonly used for I/O and for FIR delay lines.

1.3.4 Dedicated Address Generation Unit

The dedicated address generation units also help speed up the performance of the arithmetic processing on DSP. Once an appropriate addressing registers have been configured, the address generation unit operates in the background. (i.e. without using the main data path of the processor). The address required for operand access is now formed by the address generation unit in parallel with the execution of the arithmetic instructions [1]. DSP processor address generation units typically support a selection of addressing modes tailored to DSP applications. The most common of these is register-indirect
addressing with post-increment, which is used in situations where a repetitive computation is performed on data stored sequentially in memory. Some processors also support bit-reversed addressing, which increases the speed of certain fast Fourier transform (FFT) algorithms.

1.3.5 Zero-Overhead looping

DSP algorithms typically spend the majority of their processing time in relatively small sections of software that are executed repeatedly; i.e. in loops. Hence, most DSP processors provide special support for efficient looping. Often, a special loop or repeat instruction is provided which allows the programmer to implement a for-next loop without expending any clock cycles for updating and testing the loop counter or branching back to the top of the loop. This feature is often referred to as “zero-overhead looping”.

1.3.6 Data Format

Digital signal processors are largely classified based on the native arithmetic used in the processor. The classification is done as fixed-point processors and floating-point processors. The name itself indicates the native data type supported by the processor. The DSPs which use fixed point arithmetic represent the numbers as integers or as fractions in a fixed range (usually -1.0 to +1.0). In the processors which use floating point arithmetic the numbers are represented by a mantissa and an exponent format, [2].

Floating point arithmetic is a more flexible and general mechanism than fixed point. The floating point arithmetic provides the designers with a larger dynamic range, because of which the floating point processors are easy to program compared to the corresponding fixed point processors. The programmers do not have to be concerned about the dynamic range and precision issues while operating with the floating point processors. Though the fixed point processors have a disadvantage of complexity in programming due to limited dynamic range, they offer an advantage of low cost and less power consumption over the floating point counter parts.

1.3.7 Specialized Instruction Sets

The instruction sets of the digital signal processors are designed to make maximum use of the processors’ resources and at the same time minimize the memory space required to store the instructions. Maximum utilization of the DSPs’ resources ensures the maximum efficiency and minimizing the storage space ensures the cost effectiveness of the overall system.

To ensure the maximum use of the underlying hardware of the DSP, the instructions are designed to perform several parallel operations in a single instruction, typically including fetching of data in parallel with main arithmetic operation. For achieving minimum storage requirements the DSPs’ instructions are kept short by
restricting which register can be used with which operations and which operations can be combined in an instruction.

Some of the latest processors use VLIW (very long instruction word) architectures, where in multiple instructions are issued and executed per cycle. The instructions in such architectures are short and designed to perform much less work compared to those of conventional DSPs thus requiring less memory and increased speed because of the VLIW architecture.
Chapter 2

Digital Signal Processors:
Trends and Developments

Even though DSP processors have undergone dramatic changes over past couple of decades, there are certain features central to most DSP processors as discussed in the earlier chapter. In this chapter we look at the trends and developments in the field of Digital Signal Processors and their architectures.

The general DSP architectures can be divided into four generations, as per their evolution, which are discussed in following sections.

2.1 First Generation Conventional

This class of architecture represented the first widely accepted DSP processors. These processors were designed around Harvard architecture with separate data and program memory, [4]. They issue and execute one instruction per clock cycle, and use the complex, multi-operation type of instructions. These processors typically include a single MAC unit and an ALU, but these could only perform fixed-point computations. The software that accompanied the chips had specialized instruction sets and addressing modes for DSP with hardware support for software looping.

This group of processors includes TMS320C10 from Texas Instruments and ADSP-2101 from Analog Devices. These processors generally operate at 20-50MHz frequency, and provide good DSP performance while maintaining very modest power consumption.

Figure 2.1: First generation Conventional processor architecture


2.2 Second Generation Enhanced Conventional

In the next stage of development the processors retained much of the design of first generation but with added features such as pipelining multiple ALUs and accumulators to enhance performance. While these processors are code compatible with their predecessors, they provide speedup in the operations. Enhanced conventional DSP processors typically have wider data buses to allow them to retrieve more data words per clock cycle in order to keep the additional execution units fed. These hardware enhancements are combined with an extended instruction set that takes advantage of the additional hardware by allowing more operations to be encoded in a single instruction and executed in parallel, [4]. Figures 2.2 and 2.3 compare the execution units and bus architecture of first and second generation Lucent Technologies DSPs (DSP 16XX and DSP 16XXX respectively.).

![Figure 2.2: Lucent DSP 16XX (Conventional DSP)](image)

![Figure 2.3: Lucent DSP 16XXX (Enhanced Conventional DSP)](image)

Increases in cost and power consumption due to the additional hardware are largely offset by increased performance, allowing these processors to maintain cost-performance and energy consumption similar to those of previous generation.

Additionally peripheral device interfaces, counters and timer circuitry, important data acquisition etc are also incorporated in the DSP processor. The TMS320C20 from Texas Instruments and Motorola DSP 56002 are members of this generation of processors. The TMS320C20 combines both a pipelined architecture and Auxiliary Register Arithmetic Unit (ARAU). In addition to it the on chip RAM can be configured either as data or program memory. ARAU can provide address manipulation as well as compute 16-bit unsigned arithmetic thereby reducing some load on central ALU.
2.3 Third Generation Novel Designs

Enhanced conventional DSP processors provide improved performance by allowing more operations to be encoded in every instruction, but because they follow the trend of using specialized hardware and complex, compound instructions, they suffer from some of the same performance problems as conventional DSPs: they are difficult to program in assembly language and they are unfriendly compiler targets. To achieve high performance and creating an architecture that lends itself to the use of compilers some newer DSP processors use “multi-issue” approach. These generation DSP processors execute single instruction multiple data (SIMD), very long instruction word (VLIW) and superscalar operations.

SIMD allows one instruction to be executed on many independent groups of data, [5]. For SIMD to be effective, programs and data sets must be tailored for data parallel processing, and SIMD is most effective with large blocks of data. In DSP, SIMD may require a large program memory for rearranging data, merging partial results and loop unrolling. Figure 2.4 shows a way of implementing SIMD.

Making effective use of processors’ SIMD capabilities can require significant effort on the part of the programmer. Programmers often must arrange data in memory so that SIMD processing can proceed at full speed, and they may also have to reorganize algorithms to make maximum use of the processors’ resources.

VLIW processors issue a fixed number of instructions either as one large instruction, or in a fixed instruction packet, and the scheduling of these instruction is performed by the compiler. For VLIW to be effective there must be sufficient parallelism in straight line code, to occupy the operation slots. Parallelism can be improved by loop unrolling to remove branch instructions and to use global scheduling techniques. If the loops cannot be sufficiently unrolled, then VLIW causes a disadvantage of low code density.
Superscalar processors on the other hand can issue varying number of instructions per cycle and can be scheduled statically by the compiler or dynamically by the processor itself. Thus superscalar designs hold code density advantages over VLIW, because the processor can determine if the subsequent instructions in a program sequence can be issued during execution, in addition to running unscheduled programs.

The VLIW architectures’ disadvantage of code density and code compatibility were tried to be eliminated by involving the features of CISC and RISC processors in DSP processor. This evolved the hybrid architectures called explicitly parallel instruction computing (EPIC) and variable length execution sets (VLES).

The latest DSP family from Texas Instruments, TMS320C64x, combines both VLIW and SIMD in one architecture known as VelociTI. This scheme improves the performance of VLIW by allowing execution packets (EP) to span across 256-bit fetch packet boundaries. Each EP consists of a group of 32-bit instructions and EP can vary in size.

2.4 Power Considerations

Power has been a major concern especially in hand held devices like mobile phones etc. Power dissipation is lowered as parallelism is increased with the use of multiple functional units and buses. As a result the power usage is reduced when memory access is minimized. An optimum code density also saves power by scaling the instruction size to just the required amount. The burst-fill instruction cache in the Texas Instruments' TMS320C55x family is flexible enough to be optimized based on the code type. This mechanism improves the cache hit ratio and reduces memory accesses. The core processor can also dynamically and independently control the power feeding the on chip peripherals and memory arrays. If these arrays and peripherals are not used, the processor switches them to a low power mode and brings them back to full power when an access is initiated without any latency. Additionally the power is saved based on the supported data types as well. The TMS320C5x family of DSPs supports only integer operations instead of floating point operations in ‘C64x and ‘C67x family, there by compromising on increased code complexity and reduced performance.

2.5 Fourth Generation Hybrids

The content of audio and video data on internet is increasing day by day. The problem lies in processing the audio and video on the users’ end. This gives rise to the need for signal processing in a computer based environment, sometimes simultaneously with general purpose computing, or in some cases, to be capable of processing some digital signals but not wanting to incur some extra cost of a dedicated DSP chip. Devices such as these process control signals as well. The TMS320C5x family of DSPs supports only integer operations instead of floating point operations in ‘C64x and ‘C67x family, there by compromising on increased code complexity and reduced performance.
additional DSP instructions offload the processing from the general purpose processor core. The first processor in this generation of processor was SH-DSP from Hitachi. Figure 2.5 shows the simplified SH-DSP family processor architecture. The important point to be noted is the difference in the bus architectures. The general purpose processor core implements the von Neumann architecture where as the DSP core implements Harvard. The integer MAC operations in this generation processors are carried out by the general purpose processor core while the DSP core processes the more complex DSP instructions.

![Figure 2.5: Simplified SH-DSP family processor architecture](image)

The I, X, Y and Peripheral buses are the four internal buses through which the core communicate with the memories and peripheral devices. The I bus is comprised of a 32-bit address and data bus known as IAB and IDB respectively. This bus is used by both CPU and DSP core to access any memory block, i.e. X, Y or external. The X and Y bus is only accessible to the DSP core for the on-chip X and Y memories and each bus has a 15-bit address bus and a 16-bit data bus. This address bus is actually padded with a zero in the LSB position since memory accesses are aligned on word lengths. Lastly the peripheral bus transmits bidirectional data to the I bus via the bus state controller (BSC).


2.6 Benchmarking DSP

Due to various generations and various manufacturers of DSP processors, it becomes important to choose a most appropriate DSP processor for the application for which we intend to use the processor. This results in need of quick and accurate comparison of processors’ DSP performance. As the architectures diversify, it becomes more difficult to compare performance of the processors. Benchmarking becomes important on such cases to evaluate the processors’ performance.

The performance of the processor is measured based on the following five criteria.

- Cycle count
- Execution time
- Energy Consumption
- Memory use
- Cost-performance

In the past some simple and popular metrics used for the evaluation of processor performance were MIPS (Million Instructions Per Second), MOPS (Million Operations Per Second), MMACs (Millions of Multiply Accumulate), MFLOPS (Million Floating Point Operations Per Second) etc. However with the improvements in the architecture of the DSPs and introduction of various new architectures like VLIW, SIMD etc, these Metrics become less reliable, [3].

To overcome this drawback one more benchmarking technique was used to evaluate the performance of the processor by execution of a full application on the various DSPs. Even this technique results in inaccurate results as the benchmarking becomes application dependent. The main factor that causes this technique to fail is the ill-definition of the application itself. Also this method is costly and time consuming and it can not measure the performance of the processor itself independent of the system environment in which the processor is working.

Thus to avoid all these drawbacks, rigorously defined, uniformly written and optimized group of small applications are used for benchmarking the DSP performances. These applications include real block FIR filter, complex block FIR filter, 256-point FFT etc. These applications help determining the performance of the various DSP processors based on the above mentioned criteria. These benchmarks are developed by an independent organization called Berkeley Design Technology, Inc (BDTi) and the results of these are accepted worldwide.
Chapter 3

Architecture and
Peripherals of TMS320C67x

3.1 Introduction

In the previous chapter we had a glimpse of the general features of, different generation of processors and their architectures. The TMS320C6x are the first processors to use velociTI architecture, having implemented the VLIW architecture. The TMS320C62x is a 16-bit fixed point processor and the ‘67x is a floating point processor, with 32-bit integer support. The discussion in this chapter is focused on the TMS320C67x processor. The architecture and peripherals associated with this processor are also discussed.

In general the TMS320C6x devices execute up to eight 32-bit instructions per cycle. The ‘67x devices core consist of ‘C6x CPU which has following features.

- Program fetch unit
- Instruction dispatch unit
- Instruction decode unit
- Two data paths, each 32-bit wide and with four functional units
- The functional units consist of two multiplier and six ALUs
- Thirty-two 32-bit registers
- Control registers
- Control logic
- Test, emulation, and interrupt logic.
- Parallel execution of eight instructions.
- 8/16/32-bit data support, providing efficient memory support for a variety of applications.
- 40-bit arithmetic options add extra precision for computationally intensive applications.
3.2 Architecture of TMS320C67xx

The simplified architecture of TMS320C6713 is shown in the Figure 3.1 below. The processor consists of three main parts: CPU, peripherals and memory.

![Figure 3.1: Simplified block diagram of TMS320C67xx family](image)

### 3.2.1 Central Processing Unit

The CPU contains program fetch unit, instruction dispatch unit, instruction decode unit. The CPU fetches advanced very-long instruction words (VLIW) (256 bits wide) to supply up to eight 32-bit instructions to the eight functional units during every clock cycle. The VLIW architecture features controls by which all eight units do not have to be supplied with instructions if they are not ready to execute. The first bit of every 32-bit instruction determines if the next instruction belongs to the same execute packet as the previous instruction, or whether it should be executed in the following clock as a part of the next execute packet. Fetch packets are always 256 bits wide; however, the execute packets can vary in size. The variable-length execute packets are a key memory-saving feature, distinguishing the C67x CPU from other VLIW architectures. The CPU also contains two data paths (Containing registers A and B respectively) in which the processing takes place. Each data path has four functional units (.L, .M, .S and .D). The functional units execute logic, multiply, shifting and data address operation. Figure 3.2 shows the simplified block diagram of the two data paths.

![Figure 3.2: TMS320C67X data path](image)
All instructions except loads and stores operate on the register. All data transfers between the register files and memory take place only through two data-addressing units (.D1 and .D2). The CPU also has various control registers, control logic and test, emulation and logic. Access to control registers is provided from data path B.

### 3.2.2 General Purpose Register Files

The CPU contains two general purpose register files A and B. These can be used for data or as data address pointers. Each file contains sixteen 32-bit registers (A0-A15 for file A and B0-B15 for file B). The registers A1, A2, B0, B1, B2 can also be used as condition registers. The registers A4-A7 and B4-B7 can be used for circular addressing. These registers provide 32-bit and 40-bit fixed-point data. The 32-bit data can be stored in any register. For 40-bit data, processor stores least significant 32 bits in an even register and remaining 8 bits in upper (odd) register.

### 3.2.3 Functional Units

The CPU features two sets of functional units. Each set contains four units and a register file. One set contains functional units .L1, .S1, .M1, and .D1; the other set contains units .D2, .M2, .S2, and .L2. The two register files each contain sixteen 32-bit registers for a total of 32 general-purpose registers. The two sets of functional units, along with two register files, compose sides A and B of the CPU. Each functional unit has two 32-bit read ports for source operands and one 32-bit write port into a general purpose register file. The functional units .L1, .S1, .M1, and .D1 write to register file A and the functional units .L2, .S2, .M2, and .D2 write to register file B. As each unit has its own 32-bit write port, all eight ports can be used in parallel in every cycle. The .L, .S, and .M functional units are ALUs. They perform 32-bit/40-bit arithmetic and logical operations. .S unit also performs branching operations and .D units perform linear and circular address calculations. Only .S2 unit performs accesses to control register file. Table 3.1 describes the functional unit along with its description.

### 3.2.4 Memory System

The memory system of the TMS320C671x series processor implements a modified Harvard architecture, providing separate address spaces for instruction and data memory.

The processor uses a two-level cache-based architecture and has a powerful and diverse set of peripherals. The Level 1 program cache (L1P) is a 4K-byte direct-mapped cache and the Level 1 data cache (L1D) is a 4K-byte 2-way set-associative cache. The Level 2 memory/cache (L2) consists of a 256K-byte memory space that is shared between program and data space. 64K bytes of the 256K bytes in L2 memory can be configured as mapped memory, cache, or combinations of the two. The remaining 192K bytes in L2 serve as mapped SRAM.
<table>
<thead>
<tr>
<th>Functional Unit</th>
<th>Description</th>
</tr>
</thead>
</table>
| .L unit (.L1, .L2) | 32/40-bit arithmetic and compare operations  
Left most 1, 0, bit counting for 32 bits  
Normalization count for 32 and 40 bits  
32 bit logical operations  
32/64-bit IEEE floating-point arithmetic  
Floating-point/fixed-point conversions |
| .S unit (.S1, .S2) | 32-bit arithmetic operations  
32/40 bit shifts and 32-bit bit-field operations  
32 bit logical operations  
Branching  
Constant generation  
Register transfers to/from the control register file  
32/64-bit IEEE floating-point compare operations  
32/64-bit IEEE floating-point reciprocal and square root  
reciprocal approximation |
| .M unit (.M1, .M2) | 16 x 16 bit multiplies  
32 x 32-bit multiplies  
Single-precision (32-bit) floating-point IEEE multiplies  
Double-precision (64-bit) floating-point IEEE multiplies |
| .D unit (.D1, .D2) | 32-bit add, subtract, linear and circular address calculation |

*Table 3.1: Functional Units and Descriptions*

Figure 3.3 shows the memory structure in CPU of TMS320C67x. The external memory interface (EMIF) connects the CPU and external memory. This is discussed in section 3.3.

![Figure 3.3: Memory structure in CPU of TMS320C67x](image-url)
3.3 Peripherals of TMS320C6713

The TMS320C67x devices contain peripherals for communication with off-chip memory, co-processors, host processors and serial devices. The following subsections discuss the peripherals of ‘C6713 processor.

3.3.1 Enhanced DMA

The enhanced direct memory access (EDMA) controller transfers data between regions in the memory map without interference by the CPU. The EDMA provides transfers of data to and from internal memory, internal peripherals, or external devices in the background of CPU operation. The EDMA has sixteen independently programmable channels allowing sixteen different contexts for operation.

The EDMA can read or write data element from source or destination location respectively in memory. EDMA also provides combined transfers of data elements such as frame transfer and block transfer. Each EDMA channel has an independently programmable number of data elements per frame and number of frames per block.

The EDMA has following features:

- Background operation: The DMA operates independently of the CPU.
- High throughput: Elements can be transferred at the CPU clock rate.
- Sixteen channels: The EDMA can keep track of the contexts of sixteen independent transfers.
- Split operation: A single channel may be used simultaneously to perform both receive and transmit element transfers to or from two peripherals and memory.
- Programmable priority: Each channel has independently programmable priorities versus the CPU.
- Each channel’s source and destination address registers can have configurable indexes for each read and write transfer. The address may remain constant, increment, decrement, or be adjusted by a programmable value.
- Programmable-width transfers: Each channel can be independently configured to transfer bytes, 16-bit half words, or 32-bit words.
- Authentication: Once a block transfer is complete, an EDMA channel may automatically reinitialize itself for the next block transfer.
- Linking: Each EDMA channel can be linked to a subsequent transfer to perform after completion.
• Event synchronization: Each channel is initiated by a specific event. Transfers may be either synchronized by element or by frame.

3.3.2 Host-Port Interface

The Host-Port Interface (HPI) is a 16-bit wide parallel port through which a host processor can directly access the CPUs memory space. The host device functions as a master to the interface, which increases ease of access. The host and CPU can exchange information via internal or external memory. The host also has direct access to memory-mapped peripherals.

The HPI is connected to the internal memory via a set of registers. Either the host or the CPU may use the HPI Control register (HPIC) to configure the interface. The host can access the host address register (HPIA) and the host data register (HPID) to access the internal memory space of the device. The host accesses these registers using external data and interface control signals. The HPIC is a memory-mapped register, which allows the CPU access.

The data transactions are performed within the EDMA, and are invisible to the user.

3.3.3 External Memory Interface (EMIF)

The external memory interface (EMIF) supports an interface to several external devices, allowing additional data and program memory space beyond that which is included on-chip.

The types of memories supported include:

• Synchronous burst SRAM (SBSRAM)

• Synchronous DRAM (SDRAM)

• Asynchronous devices, including asynchronous SRAM, ROM, and FIFOs. The EMIF provides highly programmable timings to these interfaces.

• External shared-memory devices

There are two data ordering standards in byte-addressable microcontrollers exist:

- Little-endian ordering, in which bytes are ordered from right to left, the most significant byte having the highest address.

- Big-endian ordering, in which bytes are ordered from left to right, the most significant byte having the lowest address.
The EMIF reads and writes both big- and little-endian devices. There is no distinction between ROM and asynchronous interface. For all memory types, the address is internally shifted to compensate for memory widths of less than 32 bits.

### 3.3.4 Multichannel Buffered Serial Port (McBSP)

The C62x/C67x multichannel buffered serial port (McBSP) is based on the standard serial port interface found on the TMS320C2000 and C5000 platforms. The standard serial port interface provides:

- Full-duplex communication
- Double-buffered data registers, which allow a continuous data stream
- Independent framing and clocking for reception and transmission
- Direct interface to industry-standard codecs, analog interface chips (AICs), and other serially connected A/D and D/A devices
- External shift clock generation or an internal programmable frequency shift clock
- Multichannel transmission and reception of up to 128 channels.
- An element sizes of 8-, 12-, 16-, 20-, 24-, or 32-bit.
- μ-Law and A-Law companding.
- 8-bit data transfers with LSB or MSB first.
- Programmable polarity for both frame synchronization and data clocks.
- Highly programmable internal clock and frame generation.

The Fig 3.4 shows the basic block diagram of McBSP unit.

Data communication between McBSP and the devices interfaced takes place via two different pins for transmission and reception – data transmit (DX) and data receive (RX) respectively. Control information in the form of clocking and frame synchronization is communicated via CLKX, CLKR, FSX, and FSR. 32-bit wide control registers are used to communicate McBSP with peripheral devices through internal peripheral bus. CPU or DMA write the DATA to be transmitted to the Data transmit register (DXR) which is shifted out to DX via the transmit shift register (XSR). Similarly, receive data on the DR pin is shifted into the receive shift register (RSR) and copied into the receive buffer register (RBR). RBR is then copied to DRR, which can be read by the CPU or the DMA controller. This allows internal data movement and external data communications simultaneously.
3.3.5 Timers

The 'C62x/C67x has two 32-bit general-purpose timers that can be used to:

- Time events
- Count events
- Generate pulses
- Interrupt the CPU
- Send synchronization events to the DMA controller

The timer works in one of the two signaling modes depending on whether clocked by an internal or an external source. The timer has an input pin (TINP) and an output pin (TOUT). The TINP pin can be used as a general purpose input, and the TOUT pin can be used as a general-purpose output.

When an internal clock is provided, the timer generates timing sequences to trigger peripheral or external devices such as DMA controller or A/D converter respectively. When an external clock is provided, the timer can count external events and interrupt the CPU after a specified number of events.
3.3.6 Multichannel Audio Serial Ports (McASP)

The ‘C6713 processor includes two Multichannel Audio Serial Ports (McASP). The McASP interface modules each support one transmit and one receive clock zone. Each of the McASP has eight serial data pins which can be individually allocated to any of the two zones. The serial port supports time-division multiplexing on each pin from 2 to 32 time slots. The C6713B has sufficient bandwidth to support all 16 serial data pins transmitting a 192 kHz stereo signal. Serial data in each zone may be transmitted and received on multiple serial data pins simultaneously and formatted in a multitude of variations on the Philips Inter-IC Sound (I2S) format, [10].

In addition, the McASP transmitter may be programmed to output multiple S/PDIF IEC60958, AES-3, CP-430 encoded data channels simultaneously, with a single RAM containing the full implementation of user data and channel status fields. The McASP also provides extensive error-checking and recovery features, such as the bad clock detection circuit for each high-frequency master clock which verifies that the master clock is within a programmed frequency range.

3.3.7 Power-Down Logic

Most of the operating power of CMOS logic is dissipated during circuit switching, from one logic state to another. By preventing some or all of the chip’s logic from switching, significant power savings can be realized without losing any data or operational context. Power-down mode PD1 blocks the internal clock inputs at the boundary of the CPU, preventing most of its logic from switching, effectively shutting down the CPU. Additional power savings are accomplished in power-down mode PD2, in which the entire on chip clock structure (including multiple buffers) is halted at the output of the PLL. Power-down mode PD3 shuts down the entire internal clock tree (like PD2) and also disconnects the external clock source (CLKIN) from reaching the PLL. Wake-up from PD3 takes longer than wake-up from PD2 because the PLL needs to be relocked, just as it does following power up.
Chapter 4

Overview of
Fixed-Point Processor TMS320C55x

4.1 Introduction

The previous chapter covered a brief discussion about the architecture and peripherals of, an important family of Digital Signal Processors, from Texas Instruments, known as TMS320C67x. The important feature of these processors is its high performance due to the floating point data type support, but the important disadvantage being less power efficient. In this chapter we take a brief overview of one more important class of Digital Signal Processor family from Texas Instruments, TMS320C55x, fixed point processors. These processors are less efficient compared to the earlier one as regards to the performance but are highly power efficient as they support only integer data types. Also these devices are cheaper than the floating point counter part.

The C55x family of processors is optimized for power efficiency, low system cost, and best-in-class performance for tight power budgets, [9]. The ’C55x core delivers twice the cycle efficiency of its predecessor C54x family through a dual-MAC (multiply-accumulate) architecture with parallel instructions, additional accumulators, ALUs, and data registers.

Due to the high power efficiency, the processor family finds immense applications in various wireless handsets, portable audio players, digital cameras, personal medical devices (e.g. Hearing Aids) etc.

4.2 Architectural Features

The TMS320C5510, fixed-point digital signal processors (DSPs) are based on the TMS320C55x DSP generation CPU processor core, [6]. The C55x DSP architecture achieves high performance and low power through increased parallelism and total focus on reduction in power dissipation. The CPU supports an internal bus structure composed of one program bus, three data read buses, two data write buses, and additional buses dedicated to peripheral and DMA activity. These buses provide the ability to perform up
to three data reads and two data writes in a single cycle. In parallel, the DMA controller can perform up to two data transfers per cycle independent of the CPU activity.

The C55x CPU provides two multiply-accumulate (MAC) units, each capable of 17-bit x 17-bit multiplication in a single cycle. A central 40-bit arithmetic/logic unit (ALU) is supported by an additional 16-bit ALU. Use of the ALUs is under instruction set control, providing the ability to optimize parallel activity and power consumption. These resources are managed in the address unit (AU) and data unit (DU) of the C55x CPU. The C55x DSP generation supports a variable byte width instruction set for improved code density. The instruction unit (IU) performs 32-bit program fetches from internal or external memory and queues instructions for the program unit (PU). The program unit decodes the instructions, directs tasks to AU and DU resources, and manages the fully protected pipeline. Predictive branching capability avoids pipeline flushes on execution of conditional instructions. The 5510/5510A also includes a 24K-byte instruction cache to minimize external memory accesses, improving data throughput and conserving system power.

The table 4.1 shows the key architectural features and benefits of the C55x family of processors, and Figure 4.1 shows the simplified architecture of C55x, [9].

<table>
<thead>
<tr>
<th>Features</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 32 x 16-bit Instruction buffer queue</td>
<td>Buffers variable length instructions and implements efficient block repeat operations</td>
</tr>
<tr>
<td>Two 17-bit x17-bit MAC units</td>
<td>Execute dual MAC operations in a single cycle</td>
</tr>
<tr>
<td>One 40-bit ALU</td>
<td>Performs high precision arithmetic and logical operations</td>
</tr>
<tr>
<td>One 40-bit Barrel Shifter</td>
<td>Can shift a 40-bit result up to 31 bits to the left, or 32 bits to the right</td>
</tr>
<tr>
<td>One 16-bit ALU</td>
<td>Performs simpler arithmetic in parallel to main ALU</td>
</tr>
<tr>
<td>Four 40-bit accumulators</td>
<td>Hold results of computations and reduce the required memory traffic</td>
</tr>
<tr>
<td>Twelve independent buses:</td>
<td>Provide the instructions to be processed as well as the operands for the various computational units in parallel —to take advantage of the 'C55x parallelism.</td>
</tr>
<tr>
<td>– Three data read buses</td>
<td></td>
</tr>
<tr>
<td>– Two data write buses</td>
<td></td>
</tr>
<tr>
<td>– Five data address buses</td>
<td></td>
</tr>
<tr>
<td>– One program read bus</td>
<td></td>
</tr>
<tr>
<td>– One program address bus</td>
<td></td>
</tr>
<tr>
<td>User-configurable IDLE Domains</td>
<td>Improve flexibility of low-activity power management</td>
</tr>
</tbody>
</table>

Table 4.1: Features and Benefits of the 'C55x architecture
Following is the brief description about the main blocks.

1) **Instruction buffer unit** – This unit buffers and decodes the instructions that make up the application program. In addition, this unit includes the decode logic that interprets the variable length instructions of the ’C55x. The instruction buffer unit increases the efficiency of the DSP by maintaining a constant stream of tasks for the various computational units to perform.

2) **Program flow unit** – The program flow unit keeps track of the execution point within the program being executed. This unit includes the hardware used for efficient looping as well as dedicated hardware for speculative branching, conditional execution, and pipeline protection. This hardware is vital to the processing efficiency of the ’C55x as it helps reduce the number of processor cycles needed for program control changes such as branches and subroutine calls.

3) **Address data flow unit** – This unit provides the address pointers for data accesses during program execution. The efficient addressing modes of the ’C55x are made possible by the components of the address data flow unit. Dedicated hardware for managing the five data buses keeps data flowing to the various computational units. The address data flow unit further increases the instruction level parallelism of the ’C55x
architecture by providing an additional general-purpose ALU for simple arithmetic operations.

4) **Data computation unit** – This unit is the heart of the DSP, and performs the arithmetic computations on the data being processed. It includes the MACs, the main ALU, and the accumulator registers. Additional features include a barrel shifter, rounding and saturation control, and dedicated hardware for efficiently performing the Viterbi algorithm, which is commonly used in error control coding schemes. The instruction level parallelism provided by this unit is key to the processing efficiency of the ’C55x.

### 4.3 Low Power Enhancements

As discussed earlier, the C55x processor is optimized for power efficiency. Along with being a fixed-point processor there are other hardware optimizations done in the processor to achieve more efficient power consumption. These are now discussed in subsequent sub-sections.

#### 4.3.1 Parallelism

The C55x family of processors provides higher performance and lower power dissipation by increased parallelism. This is achieved by including two MAC units, two ALUs and multiple read/write buses. These enhancements allow processing of two data streams, or one stream at twice the speed, without the need to read coefficient values twice. Due to this memory access for a given task is minimized, thus improving power efficiency and performance.

#### 4.3.2 Alternate Computational Hardware

The C55x architecture provides flexibility in performing computational tasks. The architecture has two arithmetic/logic units (ALUs), one 40-bit ALU and one 16-bit ALU. The 40-bit ALU is used for primary computational tasks and the 16-bit ALU is used for smaller arithmetic/logic tasks. The flexible instruction set provides the capability to direct simpler computational or logical/bit-manipulation tasks to the 16-bit ALU which consumes less power. This redirection of resources also saves power by reducing cycles per task since both ALUs can operate in parallel.

#### 4.3.3 Memory Access

Memory accesses, both internal and external, can be a major contributor to power dissipation. Minimizing the number of memory accesses necessary to complete a given task furthers the goal of minimizing power dissipation per task. In the C55x family processors the program fetches are performed as 32-bit accesses rather than 16-bit accesses in its predecessors. In addition to this the instruction set of C55x processors is
variable-byte-length which means that each 32-bit word fetch actually retrieves more than one instruction. The variable length instructions improve the code density and conserve power by scaling the instruction to the amount of information needed.

4.3.4 Automatic Power Management

The C55x core CPU actively manages the power by automatic switching of the internal blocks between normal and low-power modes. When an on chip memory array is not being accessed it is automatically switched to a low power mode. The array is switched to the normal operation if any access request, for that array, arrives. The array returns to the low-power mode if no further accesses to that array are requested, and remains in the same mode until it is needed again. The similar power control is also done for the on chip peripherals. Thus, only a few of the on chip blocks are actually consuming power at any point of time which ensures maximum power efficiency.

4.3.5 Power Down Flexibility

The processor divides the architecture in many user controllable IDLE domains. These domains are the sections of the hardware which can be selectively enabled or disabled using software. When disabled the particular domain enters in to a very low power IDLE state. In the IDLE state the domain still maintains the contents of the register and memory. The domain returns to normal operation as soon as it is enabled again. This feature allows the user to have maximum control over the power consumption of the device. The main domains which consume maximum power and which can be put to IDLE state by software are: the CPU, the DMA controller, the peripherals, the external memory interface (EMIF), the instruction cache, and the clock generation circuitry, [9].

4.4 Peripherals of TMS320C5510

The TMS320C5510 digital signal processors mainly contain the following peripherals, [11].

- Clock generator with PLL
- Direct memory access (DMA) controller
- Host port interface (HPI)
- Multichannel buffered serial port (McBSP)
- Power management / Idle configurations
- Timer, general-purpose

We take a brief overview of the features of some of the important peripherals in the following subsections.
4.4.1 Clock Generator with PLL

The DSP clock generator supplies the DSP with a clock signal that is based on an input clock signal connected at the CLKin pin. Included in the clock generator is a digital phase-lock loop (PLL), which can be enabled or disabled. The clock generator can be configured to create a CPU clock signal that has the desired frequency. The clock generator can be operated in one of the two modes.

In the bypass mode, the PLL is bypassed, and the frequency of the output clock signal is equal to the frequency of the input clock signal divided by 1, 2, or 4. Because the PLL is disabled, this mode can be used to save power.

In the lock mode, the input frequency can be both multiplied and divided to produce the desired output frequency, and the output clock signal is phase-locked to the input clock signal. The lock mode is entered if the PLL ENABLE bit of the clock mode register is set and the phase-locking sequence is complete. During the phase-locking sequence, the clock generator is kept in the bypass mode.

4.4.2 DMA Controller

The DMA controller has the following important features:

- Operation that is independent of the CPU.
- Four standard ports, one for each data resource: internal dual-access RAM (DARAM), internal single-access RAM (SARAM), external memory, and peripherals.
- An auxiliary port to enable certain transfers between the host port interface (HPI) and memory.
- Six channels, which allow the DMA controller to keep track of the context of six independent block transfers among the standard ports.
- Bits for assigning each channel a low priority or a high priority.
- Event synchronization. DMA transfers in each channel can be made dependent on the occurrence of selected events.
- An interrupt for each channel. Each channel can send an interrupt to the CPU on completion of certain operational events.
- Software-selectable options for updating addresses for the sources and destinations of data transfers.
- A dedicated idle domain. The DMA controller can be put into a low-power state by turning off this domain. Each multichannel buffered serial port (McBSP) on the C55x DSP has the ability to temporarily take the DMA domain out of this idle state when the McBSP needs the DMA controller.
4.4.3 Host Port Interface (HPI)

The HPI provides a 16-bit-wide parallel port through which an external host processor (host) can directly access the memory of the DSP. The host and the DSP can exchange information via memory internal or external to the DSP and within the address reach of the HPI. The HPI uses 20-bit addresses, where each address is assigned to a 16-bit word in memory.

The DMA controller handles all HPI accesses. Through the DMA controller, one of two HPI access configurations can be chosen. In one configuration, the HPI shares internal memory with the DMA channels. In the other configuration, the HPI has exclusive access to the internal memory. The HPI cannot directly access other peripherals’ registers. If the host requires data from other peripherals, that data must be moved to memory first, either by the CPU or by activity in one of the six DMA channels. Likewise, data from the host must be transferred to memory before being transferred to other peripherals.

Figure 4.2 shows the position of HPI in the host DSP system.
Chapter 5

Conclusion

There are many applications for which the Digital Signal Processor becomes an ideal choice as they provide the best possible combination of performance, power and cost. Most of the DSP applications can be simplified into multiplications and additions, so the MAC formed a main functional unit in early DSP processors. The designers later incorporated more features, like pipelining, SIMD, VLIW etc, in the processors to deliver improved performance.

There has been a drive to develop new benchmarking schemes as the improvement in the processor architecture made the earlier benchmarking schemes, obsolete and less reliable.

Power issues are gaining importance as DSP processors are incorporated in to handheld, mobile and portable devices. This leads to development of an important class of DSP processors namely fixed-point processors.

Based on the current trends seen in the DSP processor development we may predict that the manufacturers will follow the path of general purpose processors. With new IC manufacturing technologies available we may expect to see more on-chip peripherals and memory; and in fact the system on chip may not be too far away.
References


