Digital Signal Processing on TMS320C6713 DSK: Implementation

Ketki P. Kshirsagar

Department of Electronics and Telecommunication, Vishwakarma Institute of Information Technology, Pune, India

E-mail: ketki.kshirsagar@viit.ac.in

Abstract

Signal process ideas area unit usually given during a very mathematical and abstract format. A typical resolution is to produce an active laboratory as example applications of abstract ideas. However, hardware-based digital signal process (DSP) laboratories–that area unit generally incorporated into senior-level signal process courses–sometimes emphasize programming the DSP chip instead of exploring algorithms and applications. This paper is a report on the familiarization method of the TMS320C6713 and, therefore, the implementation of digital signal process comes. This provides us added advantage of simply writing codes on MATLAB and implementing it on the DSP processors.

Keywords: *Digital signal processing, TMS320C6713, features, MATLAB tool,*
DSP processor

INTRODUCTION

Digital signal process is one amongst the core technologies, in apace growing application areas, like wireless communications, audio and video process and industrial management. The quantity and sort of merchandise that embrace some type of digital signal process has full-grown dramatically over the previous couple of years. DSP has become a key element, in several of the buyer, communications, medical and industrial merchandise that implement the signal process exploitation microprocessors, Field Programmable Gate Arrays (FPGAs), Custom ICs etc. attributable to increasing quality of the on top of mentioned applications, the variability of the DSP-capable processors has dilated greatly. DSPs are processors or microcomputers whose hardware, software, and instruction sets are optimized for high-speed numeric process applications, a necessary for process digital knowledge, representing analog signals in real time.

The DSP processors have gained augmented quality thanks to the assorted benefits like reprogram ability within the field, cost-effectiveness, speed, energy potency etc. Digital signal processors like the TMS320C6x (C6x) family of processors are like quick special-purpose microprocessors with a specialized kind of design and an instruction set applicable for signal process. The C6x notation is employed to designate a member of TX Instruments' (TI) TMS320C6000 family of digital signal processors. The design of the C6x digital signal processor is incredibly compatible for numerically intensive calculations. Supported a very-long-instruction-word (VLIW) design, the C6x is taken into account to be TI's most powerful processor. DSP techniques are terribly undefeated thanks to the event of cheap package and hardware support. For instance, modems and speech recognition is more cost-effective victimization DSP techniques.

DSP's over Analog Circuits

- Can implement complex linear or nonlinear algorithms.
- Can modify easily by changing software.
- Reduced parts count makes fabrication easier.
- High reliability.

DSP's and Microprocessors

The general purpose computers are performing two tasks:

- Data Manipulation.
- Mathematical Calculations.

All the microprocessors are capable of doing these tasks, however, it is tough to form a tool which might perform each the functions optimally, thanks to the concerned technical tradeoffs just like the size of the instruction set, however, interrupts are handled etc. equally DSPs are designed to perform the mathematical calculations required in Digital Signal process. Knowledge manipulation involves storing and sorting of data. As an example, data processing programs will a basic task of storing, organizing and retrieving of the knowledge. This can be achieved by moving knowledge from one location to a different and testing for inequalities. Whereas arithmetic is sometimes utilized in this sort of application, it is sporadic and does not considerably have an effect on the general swiftness. Additionally, to playing mathematical calculations terribly quickly, DSPs should even have a foreseeable 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 [2–4].

Feature of DSP's

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 [5].

MACs and Multiple Execution Units

The MAC operation is useful in DSP algorithms that involve computing a vector dot product, such as digital filters, correlation, and Fourier transforms.

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.

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.

Dedicated Address Generation Unit

The dedicated address generation units also help speed up the performance of the arithmetic processing on DSP.

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.

Feature of TMS320C6713 DSK [6, 7].

- A Texas Instruments TMS320C6713 DSP operating at 225 MHz.
- An AIC23 stereo codec.

- 16 Mbytes of synchronous DRAM.
- 512 Kbytes of non-volatile Flash memory.
- 4 user accessible LEDs and DIP switches.
- Software board configuration through registers implemented in CPLD.
- Configurable boot options.
- Standard expansion connectors for daughter card use.
- JTAG emulation through on-board JTAG emulator with USB host.
- Interface or external emulator.
- Single voltage power supply (+5V).

The DSP on the 6713 DSK interfaces to on-board peripherals through a 32-bit wide EMIF (External Memory Interface). The SDRAM, Flash and CPLD are all connected to the bus. EMIF signals are also connected daughter card expansion connectors which are used for third party add-in boards.

Programming on TMS320C6713

CCS provides an IDE to include the computer code tools. CCS includes tools for code generation, like a C compiler, a computer program, and a linker. It is graphical capabilities and supports real time debugging. It provides an easy-to-use computer code tool to make and rectify programs shown in Figure 1.

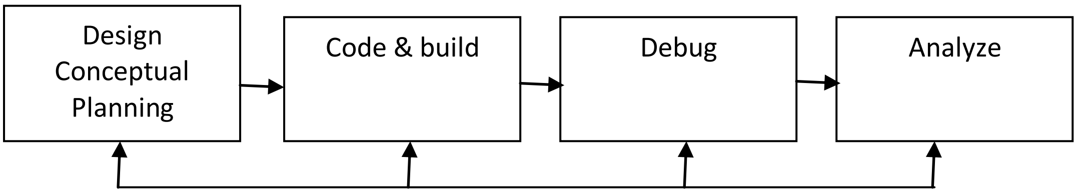

Fig. 1: *Simplified Code Composer Studio IDE Development Flow.*

Use the USB cable to connect the DSK board to the USB port on the PC [8–10]. Use the 5-V power supply included with the DSK package to connect to the +5-V power connector on the DSK to turn it on. Install CCS with the CD-ROM included with the DSK, preferably using the c:\C6713 structure. You will be working with a number of files with different extensions. They include in include file eg. file.pjt, file.c etc. Then include C6713 support files. Then create new project and compiling into machine language. Finally, to run the program, load the program into the DSP. Go to File_Load Program. Load the executable file (.out) that the compiler generated (generally in the Debug directory of the project). Then use MATLAB simulink model which is shows model system in MATLAB coding (shown in Figure 2).

Problems faced during the implementation of Chassaing/kehtarnavaz

Examples:

- During the compilation, the compiler can’t find some header (.h) files

Solution:-

Add C:\CCStudio_v3.1\C6000\dsk6713\include to the search path

- During Compilation, you get warnings about “far calls” to data?

Solution:-

Set the memory model to “data = far”

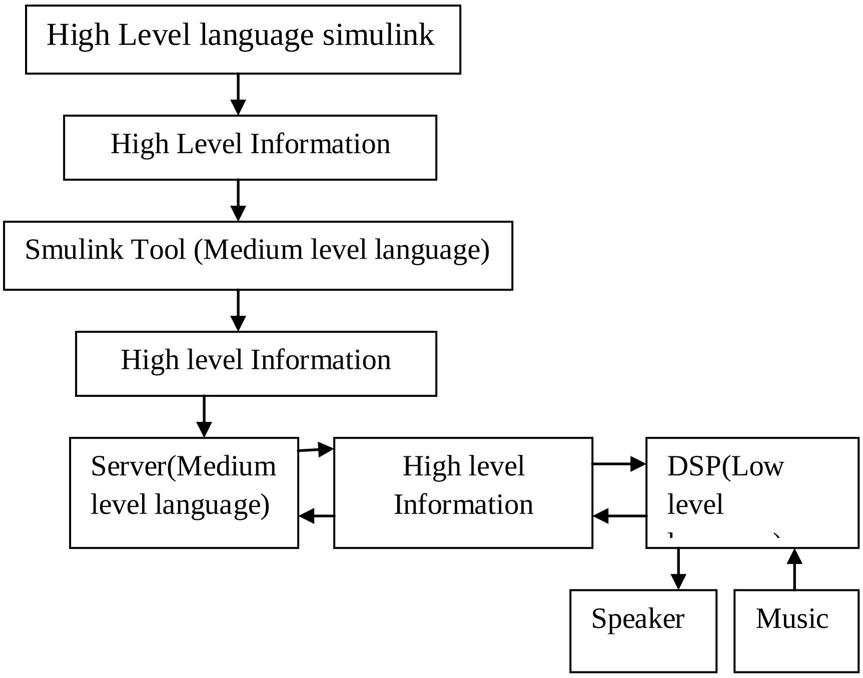
High Level language simulink
High Level Information
Smulink Tool (Medium level language)
High level Information
Server(Medium level language)
High level Information
DSP(Low level
Speaker
Music

Fig. 2: *Steps in the Modeling of a System Using Simulink.*

Implementation

For any application writing code, function sends a single 16 bit sample to the codec. In this case the same data is sent out twice, once to the left channel and once to the right channel. The codec is configured to accept data at a rate of 48,000 stereo samples per second. The serial port is used to transmit data to the codec at a much slower rate than the DSP can process data. It accepts data 16 bits at a time and shifts them out slowly one at a time. The write function returns a 1 if the write is completed successfully or a 0 if the serial channel is busy. The while () loop around the writes waits while the serial port is busy so program can be synchronized to the data rate of the codec.

For creating a project in CCS: type the code for any application, create a project in CCS, save project, add the project file, add the required library files to project, scan file dependencies, set appropriate compiler option after completion build and run project.

The project can now be built and run.

Select Project! Rebuild all or press the toolbar with the three down arrows. This compiles and assembles the source file(s). The resulting object files are then linked with the library files. This creates an executable file .out that can be loaded into the C6713 processor and run.

Select File! Load Program in order to load .out to the DSK.

CCS can be used to plot the current output data stored in the buffer out_buffer.

Select View!Graph!Time/Frequency. Change the Graph Property. The starting address of the output buffer is out_buffer. The other options can be left as default.

Choose a fast Fourier transform (FFT) order so that the frame size is 2 orders. Press OK and verify that the FFT magnitude plot.

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. 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. 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. 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.

REFERENCES

1. Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing, Second Edition,* California Technical Publishing; 1999.
2. Berkeley Design Technology, Inc. The Evolution of DSP Processors. World Wide Web. http://www.bdti.com/articles/evolution.pdf; 2006.
3. Berkeley Design Technology, Inc. Choosing a Processor: Benchmark and Beyond", World Wide Web. http://www.bdti.com/articles/20060301_TIDC_Choosing.pdf; 2006.
4. University of Rochester. DSP Architectures: Past, Present and Future. World Wide Web. http://www.ece.rochester.edu/research/wcng/papers/CAN_r1.pdf; 2006.
5. Gene Frantz. Digital signal processor trends. *Proceedings of the IEEE Micro*. 2000 20(6): 52–59p.
6. Texas Instruments. *TMS320VC5510/5510A, Fixed-Point Digital Signal Processors, Data Manual,* Dallas, TX; 2006.
7. Texas Instruments. *TMS320C62X/C67X, Programmers' Guide,* Dallas, TX; 1999.
8. Texas Instruments. *TMS320C6000, Peripherals, Reference Guide,* Dallas, TX; 2001.
9. Texas Instruments. Inc *TMS320C55x, Technical Overview,* Dallas, TX; 2000.
10. Texas Instruments. *TMS320C6713B, Floating-Point Digital Signal Processors, Data Sheet,* Dallas, TX; 2006.