DSP in Communications and Signal Processing

Dr. Kandeepan Sithamparanathan
Wireless Signal Processing Group,
National ICT Australia

Overview

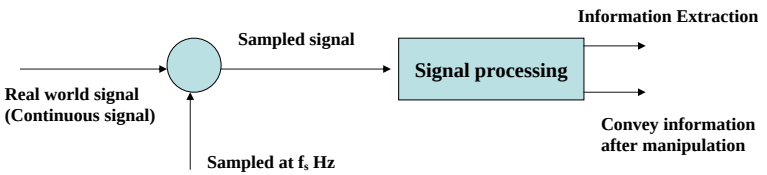
- Introduction to digital signal processing
- Introduction to digital signal processors (DSP)
- Application space – communications and signal processing
- Hardware –
  - Texas instrument's C6000 processors
  - Texas Instrument's DSP Starter Kits - DSK
- Software tools – Texas Instrument's CCS
- Project example – sonic modem

Introduction to Digital Signal Processing

- Real world signals are continuous in nature
- Signals contain information, which need to be extracted or conveyed
- Signals need to be processed for information extraction or analysis
- Signals need to be processed and modified to convey the information (at the end information will be extracted)
- Continuous signals can be processed either
  - In the continuous domain – processing with analog components
  - In the discrete domain – processing with digital components
- For discrete processing, continuous signals are sampled and discretised

Introduction to Digital Signal Processing

- The signals need to be sampled, such that the information in the signal is not lost or changed
- Nyquist's sampling theorem $f_s > 2B$
- Sampled signal spectrum repeats every f_s (Hz)



- Computers/processors that are used for processing signals are called Digital Signal Processors or DSP

Introduction to Digital Signal Processors

- Digital signal processors (DSP) are specially designed for signal processing purposes (i.e. number crunching) for embedded applications
- Was initially used in military and space applications
- Evolved into commercial applications over the past 15 years
- More parallel processing capabilities
- Efficient Arithmetic and Logical Units (ALU)
- Multiple peripheral interfaces
- Efficiently designed for real-time applications with real time operating systems

Introduction to Digital Signal Processors

- General purpose processors are not designed for real-time applications
- General purpose microprocessors are designed to do bulk processing
- DSP performance has increased by almost 40% over the past 10 years
- The DSP architecture and design are in its 4th generation
  - more onboard memory
  - VLIW architecture
  - Floating point processing as well as fixed point processing
  - High speed processors
  - Single cycle multiplication and arithmetic operations
- General purpose processors are catching up with DSP like processors, with enhanced capabilities

Introduction to Digital Signal Processors

- Some of the general purpose processors use DSP for real time processing
- Market for DSP processors in 1997 – $3 billion
- Development boards for research and education purposes (high speed sampling with wider bandwidth)
- Development boards used as prototypes for demonstrators
- Current DSP development boards include FPGA
- Testbeds for communications and signal processing applications are based on DSP based development boards
- Software based, reusable, easy to use boards

Application Space

- Telecommunications
  - Cellular phones : *Low power consumption*
  - Video conferencing : *More onboard memory*
  - Pagers and cordless phones: *Low power consumption*
  - Satellite communications: *Low power consumption for onboard systems*
  - Digital receiver and modem designs: *High speed and low power consumption*
  - Cellular base stations/ satellite earth terminals: *High speed*
- Other/signal processing applications
  - Medical applications and instrumentations (ultra sound)
  - Speech processing and automated voice systems
  - Digital cameras and MP3 players
  - Music synthesis
  - Radar and sonar applications

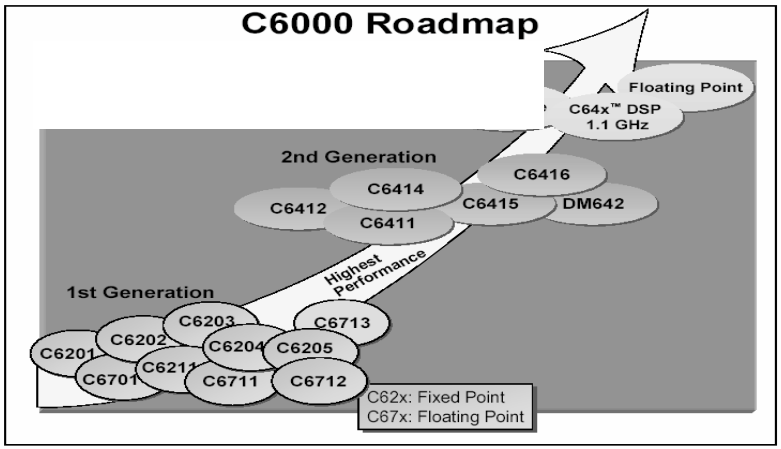
Introduction to Digital Signal Processors

- Common tasks for a DSP
  - Filtering
  - Coding (Turbo coding, Viterbi Coding)
  - Speech signal processing (speaker recognition, speech recognition, speaker tracking)
  - Signal synthesis
  - Implementing complex mathematical functions for nonlinear signal processing
  - Modulation, synchronisation and detection

Introduction to Digital Signal Processors

- How to choose a DSP for you need ?
  - Cost
  - Easy development process
  - Tools for development
  - Power consumption
  - Memory usage
  - Processor speed
  - Peripheral support

Hardware – TI's C6000 DSP



By courtesy of Texas Instruments inc , Ref[1]

TI's C6000 DSP and Peripherals

- Viterbi Coprocessor (C6416)
- Turbo Coprocessor (C6416)
- Timers and Counters
  - 2 or 3 timers, able to count internal external clocks
  - Capable of generating interrupts for CPU/EDMA
- General purpose input output – could observe signals at pin levels
- Boot loader – operates upon reset, or power-up
- Ethernet Interface
- UTOPIA for ATM connection

TI's C6000 DSP and Peripherals

- Hardware Interrupts – HWI
  - Allows synchronisation with the external world
  - CPU has 12 configurable interrupts
  - 1 global enable/disable interrupt switch
  - Interrupt source may EDMA, codecs, external pins etc
- Parallel peripherals
  - Allows a master slave operational mode by letting another DSP/PC access the memory
  - Host port Interface
  - XBUS
  - PCI

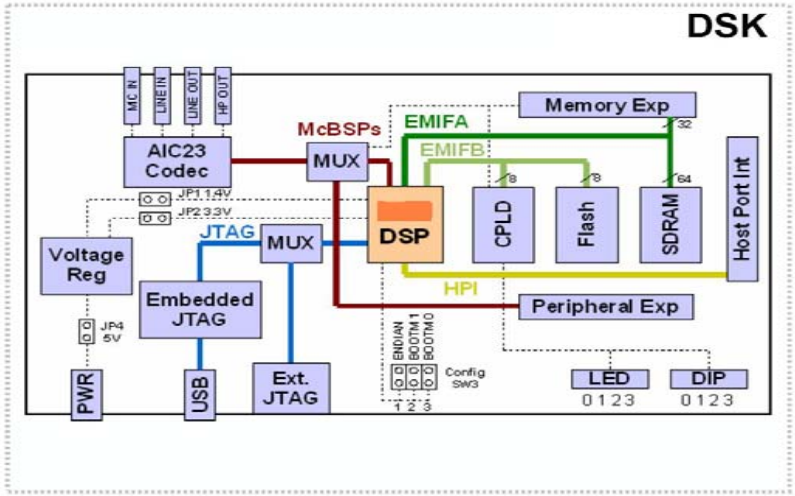
TI's C6000 DSP and Peripherals

- EDMA – Enhanced DMA
  - One of the main features of the C6000 family for its high level performances
  - Not using the EDMA means not getting the maximum performance out of the C6000 device
  - Enables data transfer between external and internal memory
  - 16 EDMA channels on C67 devices and 64 EDMA channels on C64 devices
- EMIF – External Memory Interface
  - Interface between CPU and external memory or EDMA and external memory
  - Access to Asynch and Synch memories

TI's C6000 DSP and Peripherals

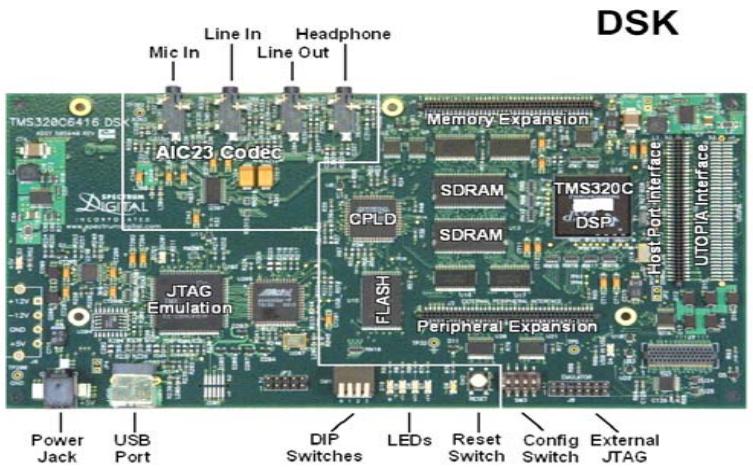
- McBSP: Multi-channel buffered Serial Port
  - Used for serial communications
  - Commonly used to connect to any serial codecs (auido codecs)
  - Upto 100Mbps speed
  - Two sections for transmit and receive, giving a fully duplex system
- McASP
  - Includes all McBSP features and many more
  - Designed for multi-channel audio processing, such as 5.1 surround sound
  - Up to 8 stereo lines (16 channels)

TI's C6713 DSP Starter Kit - DSK



By courtesy of Texas Instruments inc, Ref[1]

TI's C6713 DSP Starter Kit - DSK



By courtesy of Texas Instruments inc, Ref[1]

TI's C6713 DSP Starter Kit - DSK

C6416 / C6713 DSK Features

- **TMS320C6416 DSP:** 600MHz, fixed-point, 1M Byte internal RAM
  or
  TMS320C6713 DSP: 225MHz, floating-point, 256K Byte internal RAM
- **External SDRAM:** 16M Bytes,
  C6416 – 64-bit interface
  C6713 – 32-bit interface
- **External Flash:** 512K Bytes, 8-bit interface
- **AIC23 Codec:** Stereo, 8KHz –96KHz sample rate, 16 to 24-bit samples; mic, line-in, line-out and speaker jacks
- **CPLD:** Programmable "glue" logic
- **4 User LEDs:** Writable through CPLD
- **4 User DIP Switches:** Readable through CPLD
- **3 Configuration Switches:** Selects power-on configuration and boot modes
- **Daughtercard Expansion I/F:** Allows user to enhance functionality with add-on daughtercards
- **HPI Expansion Interface:** Allows high speed communication with another DSP
- **Embedded JTAG Emulator:** Provides high speed JTAG debug through widely accepted USB host interface

TI's C6713 DSP Starter Kit - DSK

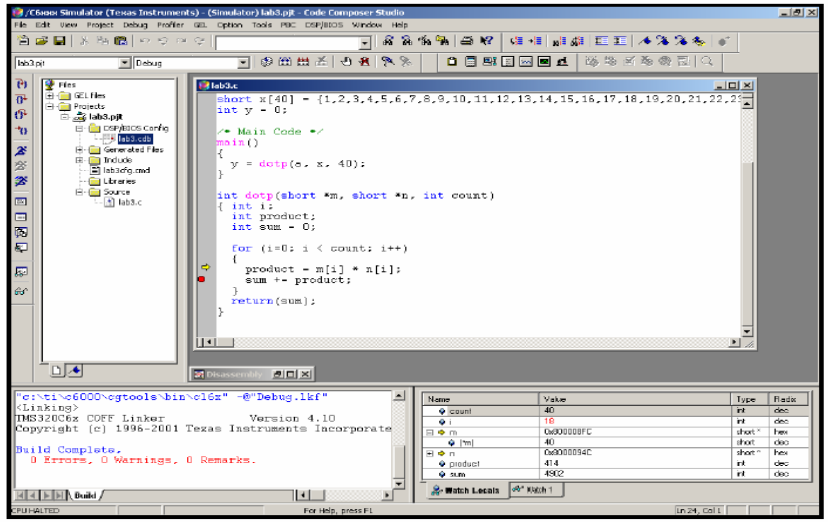
AIC23 – DSK onboard stereo audio codec

- Four external pins: line-in, line-out, microphone and headphone
- 3.5mm audio stereo pins
- Variable sampling frequency from 8khz to 96kHz derived form a 12MHz clock
- Circuit includes analog components resulting in thermal noise
- The sampled input signal is passed through an interpolator, modulator and a decimator for improved signal top noise ratio performances
- Uses sigma-delta technique for the discretisation process, which is a nonlinear process
- Sampled data are available in 2's complement format
- Data transfer from and to the audio codec is performed by the McBSP

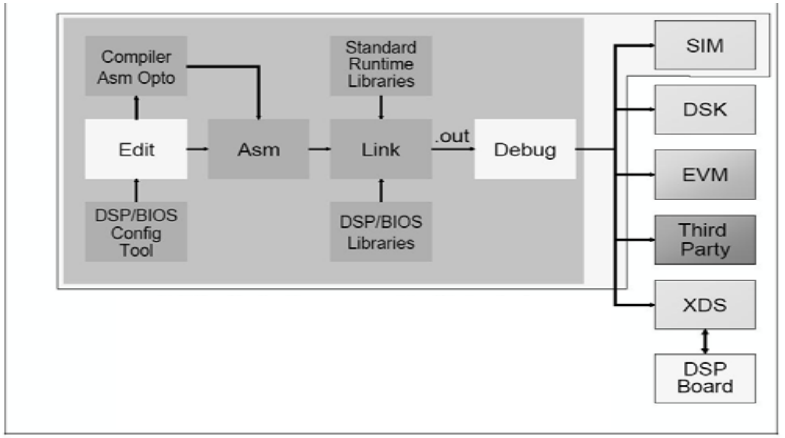
Software Tools for Development

- Texas Instruments provides its own software development tool called the Code Composer Studio – CCS
- CCS runs on the host PC
- CCS is an Integrated Development Environment (IDE)
  - Codes could be entered and edited in CCS
  - Codes can be built and compiled to make an executable
  - Target DSP can be run and stopped from CCS
  - Codes can be debugged at assembly level
  - Visualisation tools, to view memory
- The license to the DSK version of CCS is the Development board itself (a hardware key for the software)
- CCS includes the compiler assembler and the linker

TI's CCS - Snap Shot



TI's Code Composer Studio

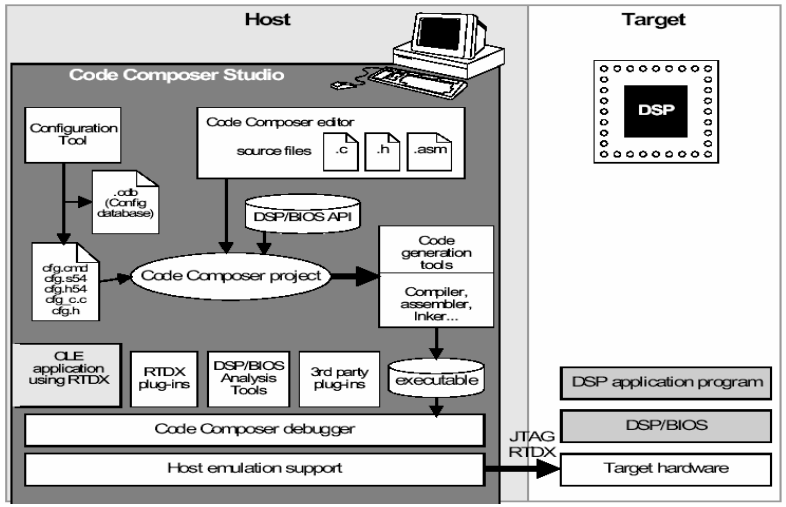


By courtesy of Texas Instruments inc, Ref[1]

Software Development Tools

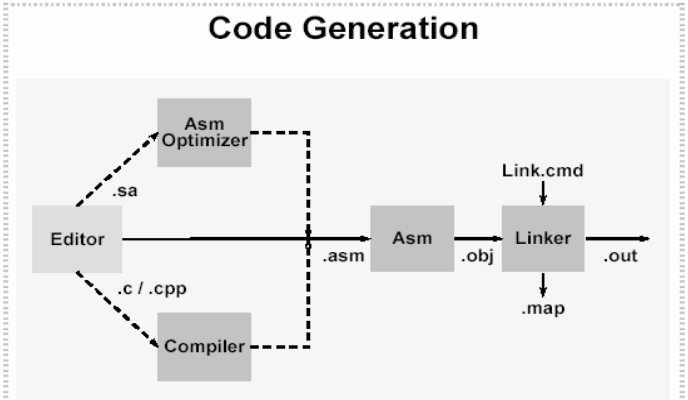
- CCS also includes the DSP/BIOS
- DSP/BIOS is a real time operating system developed for TI's DSPs
- DSP/BIOS consists of three main features
  - Real time scheduler
  - Real time Capture analysis
  - Real time Input/Output
- DSP/BIOS makes life easy for the application developer
- DSP/BIOS Kernel could be altered if required
- User has the option to right codes for the DSP instead of using the DSP/BIOS, which might be more efficient in some instances

DSP/BIOS



By courtesy of Texas Instruments inc, Ref[1]

TI's Code Composer Studio



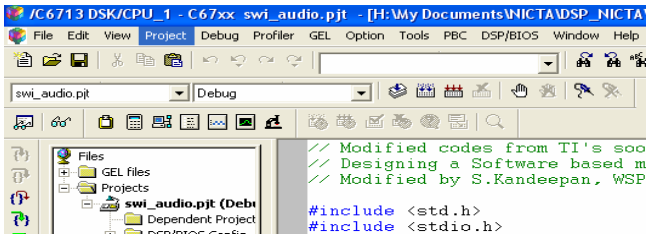
By courtesy of Texas Instruments inc, Ref[1]

Code Generation

- Codes are written in C or C++. With extensions .c or .cpp
- Codes can also be generated in Assembly or linear Assembly, with .asm extension or .sa extension
- Linear Assembly codes are incorporated using the Asm Optimizer
- Two modes for compilation in CCS
  - Debug mode
  - Optimization mode
- Assembler creates the object file for the linker
- Linker generated the executable .out file to be loaded on to the DSP
- .MAP file is a report file generated from the linking process

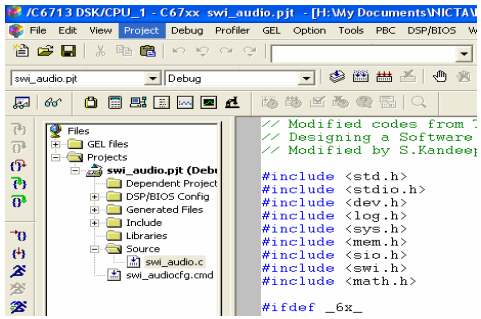
Project Development

- **Step1:** Connect the DSK board to the host PC via the USB interface and power up the DSK
- **Step2:** Start CCS (note – CCS would not start without the DSK board)
- **Step3:** Go to the 'project' tab in CCS and open the project called 'swi_audio.pjt'. All the file associated with the project will be loaded, and could be viewed on the left hand side of the CCS window.



Project Development

- **Step4:** Go to the 'source' folder under the swi_audio project in your left side window on the CCS, then double click on the 'swi_audio.c' file to open it on the main window. You can go through the source codes to get some understanding of how the DSP is programmed.



Step5: Build the project (that is compile and link the codes), by clicking on the tool shown below with the red arrows on it.

Project Development

- **Step6:** Now, load the program to the DSP by going to 'File' tab and clicking on the Load Program. Now the program is loaded into the DSP and the DSP is ready to perform the instructions given in the program. Note that the program cannot be loaded if there are errors in the compilation process (i.e. the building process)
- **Step7:** Now, run the DSP by clicking on the run action on the left side panel. When the DSP is running, the message "CPU Running" is displayed at the bottom left hand side of the CCS window. You can stop the DSP as shown below. When the DSP is not running the message "DSP Halted" is displayed at the bottom.

Project Task

A Simple Binary FSK based modem design

- **Objective –** You have to design a modem based on Binary FSK modulation. The audio codec on the DSK has got the sampling frequency set at 48kHz, therefore use frequency tones of 1200Hz and 2400Hz for the FSK modulation. You could transmit these tones through the 'Lineout' terminal of the DSK using a speaker and receive the signal using a microphone, directed to the speakers, connected to the 'Microphone' terminal of the DSK. Design a simple receiver/detector for the BFSK that you designed. You can synchronise the transmission by transmitting some known sequence of data in order to get coherent detection, and also achieve timing synchronisation
- You can use a single DSK board (using loop back techniques) or use two DSK boards depending on the availability. Note that you can connect only one DSK per PC.

Advanced Task

Software Implemented Digital Phase Locked Loop

- Design a Digital Phase-Locked Loop (DPLL) for tracking down the fundamental frequency of a symmetric periodic signal
- The DPLL is implemented in software hence also known as Software Implemented PLL or SPLL
- Most fundamental element of software radio design
- If not familiar with DPLL, refer to [DPLL]

Summary

- Introduction to DSP and its applications
- Introduction to DSP Implementation
- DSP usage increases rapidly with emerging embedded applications
- Introduction to TI's DSP Starter Kit – DSK
- Future work on DSP design and implementation with WSP @ NICTA

Wrap up

- Ongoing/future DSP projects @ WSP, NICTA
  - Software defined radio
  - Efficient Receiver designs
  - Equalisation and synchronisation techniques, and Coding techniques
  - DSP based wireless test-bed design
  - Channel measurements
  - Multi-channel DSP based test-bed
  - MIMO applications: channel measurement, receiver design, synchronisation techniques etc

Reference:

[1] Texas Instruments Tutorials and Training manuals

The imagination driving Australia's ICT future.

Thank you

For more details contact
Email: Kandeepan.Sithamparanathan@nicta.com.au
Phone: 61256251