Real-Time DSP

ECE 5655/4655 Lecture Notes

© 1998 – 2014
Mark A. Wickert
Portions "Courtesy of Texas Instruments"
# Table of Contents

## Overview of Real-Time Digital Signal Processing

- Introduction ......................................................... 1–1
  - A Brief Description of the Course 1–1
  - Background Requirements 1–4
- A Brief History of DSP in the Context of Real-Time Processing .......... 1–4
  - Great Moments in DSPS History 1–6
  - Who Are the Players? 1–7
  - DSP Integration (Frantz) 1–7
  - Communications and Wireless 1–8
- DSP Hardware Design Issues ........................................ 1–11
- DSP System Design Flow (Marven & Ewers) .......................... 1–12
- Course Laboratory Foundations ...................................... 1–13
  - Software 1–14
  - Test Equipment 1–14

## The TMS320C6x Family: Hardware and Software

- Introduction ......................................................... 2–1
  - C62/64 Product Specifications 2–2
  - C67 Product Specifications 2–3
- C6x Family Feature Overview ...................................... 2–4
- C6x Architecture Overview ......................................... 2–5
  - A High-Level Look 2–5
  - C64x™ Enhancements Include: 2–6
  - A Detailed Look at the C6x Architecture: 2–7
  - Functional Unit Operations 2–8
  - C62/67 and C64 Data Paths 2–9
  - The Internal Bus Structure 2–10
  - C6x Internal Buses for 6201 and 6701 2–10
- Internal Memory ..................................................... 2–11
  - C6x11 Double Level Memory 2–11
  - Cache versus RAM 2–12
  - Internal Memory Summary 2–13
- External Memory Interface (EMIF) .................................. 2–13
  - SDRAM 2–14
  - SBSRAM 2–14
  - Asynchronous Memory 2–14
Analog Input and Output

Introduction ................................................. 4–1
Analog I/O Basics ............................................. 4–1
Analog I/O on the C6x ........................................ 4–3
  Interrupt Steps 4–3
  Source Selection 4–4
  Interrupt Vector Table 4–6
  Enabling Interrupts 4–10
  Interrupt Service Routines in C 4–13
  Serial Port Communication 4–15
On-Board TLC320AD535 Telephony Codec .................. 4–20
  A Simple Interrupt Program 4–23
PCM3006 Stereo Daughter Card Codec ....................... 4–29
  Daughtercard Physical Layout (Mike Morrow) 4–31
  Supplied Software (Mike Morrow) 4–32
  MATLAB API (Mike Morrow) 4–42
TI PCM 3003 Audio Daughter Card .......................... 4–48
  Default Jumper Settings 4–49
  Jumper Descriptions 4–50
  Board Schematics 4–52
  A Simple Interrupt Program 4–55
6713 DSK with On-Board TLV320AIC23
Stereo Audio Codec ........................................... 4–60
  AIC23 Portion of the 6713 DSK 4–65
Fixed-Point Considerations

Introduction ................................................................................. 5–1
C6x Data Types .............................................................................. 5–2
Q-Format Number Representation .............................................. 5–2
Multiplicative Overflow ................................................................. 5–5
Accumulative Overflow ................................................................. 5–10
  Saturating the Result 5–11
  Use Guard Bits 5–13
  Overflow Allowed by Design 5–14
Floating Point .................................................................................. 5–15
  Single Precision 5–15
  Double Precision 5–16
  Addition 5–16

Code Optimization and Advanced C

Introduction .................................................................................. 6–1
C Runtime Environment ................................................................. 6–1
  Stack Pointer 6–1
  Global Pointer 6–2
  Memory Management 6–4
  Use of Volatile 6–5
Optimizing C Code ......................................................................... 6–6
  Memory Aliases 6–6
  C Intrinsic Functions 6–6
Intrinsics Reference Guide ............................................................... 6–9

Real-Time FIR Digital Filters

Introduction .................................................................................. 7–1
Basics of Digital Filter Design ......................................................... 7–2
  Overview of Approximation Techniques 7–4
Basic FIR Filter Topologies ............................................................ 7–5
Overview of FIR Filter Design ............................................. 7–7
  Why or Why Not Choose FIR? 7–7
  FIR Design Using Windowing 7–7
  Example: Use of the Hanning Window 7–10
  Lowpass Design 7–14
  Optimum Approximations of FIR Filters 7–17
MATLAB Basic Filter Design Functions .............................. 7–19
  Windowed FIR Design From Amplitude Specifications 7–20
  Using MATLAB’s fdatool .............................................. 7–23
  Example: Windowed FIR Design 7–23
  Example: Equiripple FIR Design 7–28
  Writing C Coefficient Files 7–31
Filter Implementation Code Examples ................................ 7–35
  A Simple Fixed-Point Implementation 7–35
  A Simple Floating Point Implementation 7–40
  Direct Export Into Code Composer 7–42
Circular Addressing ....................................................... 7–45
  Circular Buffer in C 7–45
  C Circular Buffer 7–46
Performance of Interrupt Driven Programs .......................... 7–55
  Traditional Code Timing Approach 7–56
  Using Counters in the Wait Loop 7–57
MATLAB CCS Link Example ............................................. 7–61

Real-Time IIR Digital Filters

Introduction ............................ ................................. 8–1
Basic IIR Filter Topologies ................................................ 8–1
  Direct Form I 8–2
  Direct Form II 8–3
  Cascade Form 8–5
  Parallel Form 8–6
  Transposed Forms 8–8
  Pole-Zero Lattice 8–8
Digital Sinusoidal Oscillators ........................................... 8–8
Overview of IIR Filter Design ......................................... 8–11
  IIR Approximation Approaches 8–11
  Converting to 8–13
  Classical Design from Analog Prototypes 8–18
  Obtaining the Second-Order Section Coefficients 8–19
MATLAB Design Functions ............................................. 8–19
  MATLAB Filter Design Examples 8–21
  Writing C Coefficient Files 8–33
Filter Implementation Code Examples .............................................. 8–35
A Simple Floating-Point Cascade of Biquads Implementation 8–35
Direct Export Into Code Composer 8–39
Coupled Form Oscillator in C 8–41
MATLAB CCS Link Example .......................................................... 8–45
Command Line Interfacing 8–45
MATLAB GUI Interfacing 8–52

Adaptive Filters
Introduction ................................................................. 9–1
Wiener Filter .............................................................. 9–2
Adaptive Wiener Filter. ....................................................... 9–4
  Least-Mean-Square Adaptation 9–6
  Adaptive Filter Variations 9–8
Adaptive Line Enhancement .................................................... 9–9
C6x Code Examples. ......................................................... 9–13
  A Two Channel Input Signal + Signal or Signal + Noise Canceller 9–13
  Adaptive Line Enhancer 9–18

Real-Time Fast Fourier Transform
Introduction ................................................................. 10–1
DTFT, DFT, and FFT Theory Overview ..................................... 10–1
  The DTFT 10–2
  The DFT and its Relation to the DTFT 10–2
Simple Application Examples ................................................... 10–5
  Spectral Analysis 10–5
  Transform Domain Linear Filtering 10–9
Radix 2 FFT ........................................................................ 10–10
  Decimation-in-Time 10–11
  Decimation-in-Frequency 10–15
  Computing the IDFT 10–15
Frame Processing .................................................................... 10–16
Radix-2 FFT Implementation on the C6x ................................. 10–17
  A C Based Real-Time FFT 10–17
  Transform Domain Filtering using a Linear Assembly Based FFT and IFFT 10–21

Real-time Analysis and Scheduling
Introduction ................................................................. 11–1
Features ............................................................... 11–1
The DSP/BIOS API Modules ........................................... 11–3
A Case Study: Audio Player with DTMF ............................... 11–4
   Run them together under `main()` : 11–5
   A second solution is to use two interrupts under `main()` : 11–5
   Interrupt driven state machine 11–6
   The DSP/BIOS Solution 11–7
   Getting Started with DSP/BIOS 11–10

**Communications Applications**

Introduction ................................................................. 12–1
   Summary of Factors Relating to Implementation Choices 12–2
Transmitting Signals ..................................................... 12–3
   Baseband/IF Transmitter 12–3
   Complex Baseband Transmitter 12–6
   Practical Pulse Shaping 12–8
Receiving Signals ......................................................... 12–15
   Complex Envelope Representation 12–15
   Standard I-Q Demodulation 12–16
   Using the Hilbert Transform 12–17
System Application: A DSP Based Costas Loop for Coherent Carrier Recovery .................................................. 12–23
   MATLAB Simulation 12–28
   Real-Time Implementation on the C6711 DSK 12–32
Commercial Receiver Architectures ................................. 12–42

**Reference Frameworks**

Introduction ................................................................. 13–1

**Simulink Embedded Target**

Introduction ................................................................. 14–1
Overview of Real-Time Digital Signal Processing

Introduction

In this first chapter we provide motivation for the topics to be addressed in this course. Before going any further let us first give a short description of the course and the assumed background for the course

A Brief Description of the Course

• A course in real-time DSP brings together the following:
  – Continuous- and discrete-time systems theory (in particular knowledge from a first DSP course)
  – Software engineering concepts
  – Microprocessor programming and hardware interfacing
• The interest in doing this stems from the increase in real-time DSP applications headed for the consumer market, and the ever improving device VLSI designs for implementing powerful DSP microprocessors
- Texas Instruments has created the following long list of DSP real-time application areas

<table>
<thead>
<tr>
<th>Automotive</th>
<th>Consumer</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive ride control</td>
<td>Digital radios/TVs</td>
<td>Disk drive control</td>
</tr>
<tr>
<td>Antiskid brakes</td>
<td>Educational toys</td>
<td>Engine control</td>
</tr>
<tr>
<td>Cellular telephones</td>
<td>Music synthesizers</td>
<td>Laser printer control</td>
</tr>
<tr>
<td>Digital radios</td>
<td>Pagers</td>
<td>Motor control</td>
</tr>
<tr>
<td>Engine control</td>
<td>Power tools</td>
<td>Robotics control</td>
</tr>
<tr>
<td>Navigation and global positioning</td>
<td>Radar detectors</td>
<td>Servo control</td>
</tr>
<tr>
<td>Vibration analysis</td>
<td>Solid-state answering machines</td>
<td></td>
</tr>
<tr>
<td>Voice commands</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anticollision radar</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>General-Purpose</th>
<th>Graphics/Imaging</th>
<th>Industrial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive filtering</td>
<td>3-D rotation</td>
<td>Numeric control</td>
</tr>
<tr>
<td>Convolution</td>
<td>Animation/digital maps</td>
<td>Power-line monitoring</td>
</tr>
<tr>
<td>Correlation</td>
<td>Homomorphic processing</td>
<td>Robotics</td>
</tr>
<tr>
<td>Digital filtering</td>
<td>Image compression/transmission</td>
<td>Security access</td>
</tr>
<tr>
<td>Fast Fourier transforms</td>
<td>Image enhancement</td>
<td></td>
</tr>
<tr>
<td>Hilbert transforms</td>
<td>Pattern recognition</td>
<td></td>
</tr>
<tr>
<td>Waveform generation</td>
<td>Robot vision</td>
<td></td>
</tr>
<tr>
<td>Windowing</td>
<td>Workstations</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instrumentation</th>
<th>Medical</th>
<th>Military</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital filtering</td>
<td>Diagnostic equipment</td>
<td>Image processing</td>
</tr>
<tr>
<td>Function generation</td>
<td>Fetal monitoring</td>
<td>Missile guidance</td>
</tr>
<tr>
<td>Pattern matching</td>
<td>Hearing aids</td>
<td>Navigation</td>
</tr>
<tr>
<td>Phase-locked loops</td>
<td>Patient monitoring</td>
<td>Radar processing</td>
</tr>
<tr>
<td>Seismic processing</td>
<td>Prosthetics</td>
<td>Radio frequency modems</td>
</tr>
<tr>
<td>Spectrum analysis</td>
<td>Ultrasound equipment</td>
<td>Secure communications</td>
</tr>
<tr>
<td>Transient analysis</td>
<td></td>
<td>Sonar processing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Telecommunications</th>
<th>Voice/Speech</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1200- to 33 600-bps modems</td>
<td>Faxing</td>
<td>Speaker verification</td>
</tr>
<tr>
<td>Adaptive equalizers</td>
<td>Line repeaters</td>
<td>Speech enhancement</td>
</tr>
<tr>
<td>ADPCM transcoders</td>
<td>Personal communications systems (PCS)</td>
<td>Speech recognition</td>
</tr>
<tr>
<td>Cellular telephones</td>
<td>Personal digital assistants (PDA)</td>
<td>Speech synthesis</td>
</tr>
<tr>
<td>Channel multiplexing</td>
<td>Speaker phones</td>
<td>Speech vocoding</td>
</tr>
<tr>
<td>Data encryption</td>
<td>Spread spectrum communications</td>
<td>Text-to-speech</td>
</tr>
<tr>
<td>Digital PBXs</td>
<td>Video conferencing</td>
<td>Voice mail</td>
</tr>
<tr>
<td>Digital speech interpolation (DSI)</td>
<td>X.25 packet switching</td>
<td></td>
</tr>
<tr>
<td>DTMF encoding/decoding</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Echo cancellation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This course is about the use of general purpose digital signal processing microprocessors for solving signal processing problems in real-time

The course focus will be on using the Texas Instruments (TI) C6x family of fixed/floating processors, and in particular implementing programs written in assembly, linear assembly, and ANSI C, for the TI OMAP-L138 (ARM Cortex-A9 + C6748 floating-point DSP) ZOOM™ OMAP-L138 Experimenter Kit (TMDSEXPL138 $495 at www.ti-estore.com)
- A lower cost version of this board that does not include the touch screen is available for ~$150 (where?)

Other platforms that will be explored this semester include:
- TI C6713 DSK (digital signal processing starter kit available for ~$395.00 at www.ti-estore.com)
- TI C55x fixed-point family via the VC5505/VC5515 eZDSP USB Stick ($49 at www.ti-estore.com)
- TI OMAP-3530 (ARM Cortex-A8 + C64x+ DSP) Beagle-Board-xM ($149 from links at http://beagleboard.org/)

The course will start out considering general signal processing applications of real-time DSP and the associated programming issues
- Later in the course, a focus applications area will be wireless communication system design using DSP algorithms
• In addition to the text book, we will be using TI training materials from the courses
  – TMS320C6000 DSP Workshop (normally a 4 day course)
  – TI C6000/OMAL-L138 Teaching Materials DVD v3
• The course meeting time will be used for lecturing and laboratory time (about 60/40) using hardware/software development tools, primarily Code Composer Studio (4.2 & 5.1)

**Background Requirements**

• The required background for all students taking the course is an introductory graduate or junior/senior level undergraduate course in DSP and experience programming in ANSI C
• Knowledge of ANSI C is required in order to develop real-time algorithm in a high level language (HLL)
• TMS320C6x assembly and linear assembly language programming will be covered in the course
• Eventually, and in practice, we will be using a combination of C and assembly, e.g., mixed language programming.
• Programming the PC host in C/C++, or perhaps some other language, may be useful for developing a user interactive application for the final project
  – A MATLAB API (application programming interface) for the DSK is also available
• Familiarity with test equipment e.g., signal generators, digital scopes, and a spectrum/network analyzer would be helpful
Course Perspective

Signals & Systems

Modern DSP

Real-Time DSP

Inform/ Coding

Comm Sys II

Comm Lab

Prob & Statistics

Statistical Signal Process

Spread Spectrum

Wireless Networking

Optical Comm

Satellite Comm

Wireless & Mobil Com

PLL & Freq Syn

Random Signals

Detect/ Estimation

Spectral Estimation

Comm Networks

Radar Systems

Image Processing

Undergraduate Engineering Curriculum

Senior/1st Year Graduate Signals & Systems Courses

Other Graduate Signals & Systems Courses Offered on Demand/Indep. Study
A Brief History of DSP in the Context of Real-Time Processing

- From a systems engineering point of view DSP stands for digital signal processing, but in the world of real-time hardware, systems people often refer to digital signal processing solutions (DSPS)

- The decades of DSP/DSPS have brought the following

<table>
<thead>
<tr>
<th>Decade</th>
<th>Characteristic</th>
<th>$/MIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘60s</td>
<td>University Curiosity</td>
<td>$100–$1,000</td>
</tr>
<tr>
<td>‘70s</td>
<td>Military Advantage</td>
<td>$10–$100</td>
</tr>
<tr>
<td>‘80s</td>
<td>Commercial Success</td>
<td>$1–$10</td>
</tr>
<tr>
<td>‘90s</td>
<td>Consumer Enabler</td>
<td>$0.10–$1</td>
</tr>
<tr>
<td>Beyond</td>
<td>Expected Part of Daily Life?</td>
<td>$0.01–$0.10</td>
</tr>
</tbody>
</table>

• Another way of looking at this is (Frantz)

<table>
<thead>
<tr>
<th>Processing</th>
<th>Processors</th>
<th>Processing Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1980</td>
<td>1990</td>
<td></td>
</tr>
<tr>
<td>Technology</td>
<td>Product</td>
<td>Technology</td>
</tr>
<tr>
<td>What is DSP?</td>
<td>How do I create a product?</td>
<td>How do I solve problems?</td>
</tr>
</tbody>
</table>

• We are well beyond the time line shown above, what has been happening in the last 12 years?
  – DSP is definitely mainstream in most everything electronic
  – general purpose DSP processors are only part of the picture; FPGA drives the government/military applications side, and ARM processors drive the consumer product side
  – In the last few years we have seen more multi-core devices
  – The open media application platform, OMAP-L138 used for this course is an example of this
Great Moments in DSPS History

• **1976** – DSP is used to simulate a voice in the educational product “Speak and Spell”

• **1982** – TI announces details on the TMS32010 which executes at 5 MIPS

• **1985** – DSP is used in a modem for the first time

• **1986** – Lotus automotive uses DSP in active suspension and noise abatement in racing cars

• **1988** – First DSP hearing aid introduced

• **1991** – First TI sponsored educators conference

• **1993** – Cadillac automotive introduces a DSP-based ride control system

• **1995** – TI implements the *On-line DSP Lab™* for Web based testing of DSP applications

• **1997** – 15 years of DSPs for TI and
  – The introduction of the TMS320C6x family of DSPs 1,600 MIPS performance (C62x)
  – The 1-v barrier in power consumption reached
  – In late 1997 the first C6x floating point part is announced (C67x) with 1 GFLOPS performance

• **2012**?

---

A Brief History of DSP in the Context of Real-Time Processing

DSP Vendor Market Share for 2006

- TI increased by 17% in 2006 over 2005; 17% increase in cellular shipments and 9% in the DSP market

DSP Markets for 2006

Source: Forward Concepts, 2007
DSP Integration (Frantz)

Table 1.2: DSP Integration over three decades.

<table>
<thead>
<tr>
<th></th>
<th>Typical 1982 DSP</th>
<th>Typical 1992 DSP (97)</th>
<th>Typical 2002 DSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Die size</td>
<td>50 mm</td>
<td>50 mm</td>
<td>50 mm</td>
</tr>
<tr>
<td>Technology size</td>
<td>3 µ</td>
<td>0.8 (.35) µ</td>
<td>0.18 µ</td>
</tr>
<tr>
<td>MIPS</td>
<td>5</td>
<td>40 (100)</td>
<td>2000</td>
</tr>
<tr>
<td>MHz</td>
<td>20</td>
<td>80 (200)</td>
<td>500</td>
</tr>
<tr>
<td>RAM (words)</td>
<td>144</td>
<td>1 K</td>
<td>16 K</td>
</tr>
<tr>
<td>ROM (words)</td>
<td>1.5 K</td>
<td>4 K</td>
<td>64 K</td>
</tr>
<tr>
<td>Price</td>
<td>$150.00</td>
<td>$15.00</td>
<td>$1.50</td>
</tr>
<tr>
<td>Pwr. Dissp.</td>
<td>250 mW/MIP</td>
<td>12.5 mW/MIP (1)</td>
<td>0.1 mW/MIP</td>
</tr>
<tr>
<td>Transistors</td>
<td>50 K</td>
<td>500 K</td>
<td>5 M</td>
</tr>
<tr>
<td>Wafer Size</td>
<td>3” (75 mm)</td>
<td>6” (150 mm)</td>
<td>12” (300 mm)</td>
</tr>
</tbody>
</table>

- In Spring 2000 TI also announced the C55x family, which offers 0.05 mW/MIPS and 600–800 MIPS
- Late 2004, C67 at 1350 MFLOPs at $36.50, 800 MFLOPs at $21.07
- Late 2004, 1GHz C64 which offers fixed-point processing for communications and imaging applications (eight 32-bit instructions/cycle, 28 operations/cycle, 8000 MIPS
**A Brief History of DSP in the Context of Real-Time Processing**

<table>
<thead>
<tr>
<th>Microcontrollers (MCUs)</th>
<th>ARM®-Based Processors</th>
<th>Digital Signal Processors (DSPs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-bit ultra-low power MCUs</td>
<td>32-bit ARM Cortex™-M3 MCUs</td>
<td>DSP DSP+ARM</td>
</tr>
<tr>
<td>32-bit real-time MCUs</td>
<td>ARM Cortex-A8 MPUs</td>
<td>Multi-core DSP</td>
</tr>
<tr>
<td>32-bit ARM Cortex™-M3 MCUs</td>
<td>C5000™</td>
<td>Ultra Low power DSP</td>
</tr>
<tr>
<td>MSP430™</td>
<td>C2000™</td>
<td></td>
</tr>
<tr>
<td>Delfino™ Piccolo™</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C5000™</td>
<td>C6000™</td>
<td></td>
</tr>
<tr>
<td>Sitara™ ARM® Cortex™-A8 &amp; ARM9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C6000™ DaVinci™ video processors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integra™</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-core</td>
<td></td>
<td></td>
</tr>
<tr>
<td>software &amp; Devices</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programmer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Board Support Package</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software &amp; Dev. Tools</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **MPUs – Microprocessors**
- **C6000™**: 24,000 MMACS
- **C5000™**: Up to 300 MHz +Accelerator
- **C5000™**: Up to 128KB ROM
- **C6000™ DaVinci™**: USB, ENET, PCIe, SATA, SPI
- **C6000™**: Floating/Fixed Point
- **C6000™ DaVinci™**: Industrial computing, Calculations, Portable data terminals
- **C6000™**: $5.00 to $200.00
- **C5000™**: 2.000 MMACS
- **C5000™**: Cache
- **C5000™**: RAM, ROM
- **C5000™**: SRIO, EMAC, DP, PCIe
- **C5000™**: Floating/Fixed Point
- **C5000™**: $5.00 to $200.00
- **C5000™**: Telecom test & meas, media gateways, base stations
- **C5000™**: $40 to $200.00
- **C5000™**: Audio, Voice, Media
- **C5000™**: Biometrics
- **C5000™**: Medical, Biometrics
- **C5000™**: $3.00 to $10.00
Chapter 1 • Overview of Real-Time Digital Signal Processing

– C67x floating-point family

New OMAP-L138, TMS320C6748/6/2 power-efficient processors offer unmatched connectivity options for industrial and communication applications

With these four new processors from TI based on the OMAP-L13x/C674x platforms, developers can:

- Access flexible networking and interface options (SATA, uPP, Ethernet MAC and more)
- Optimize their designs for power and performance utilizing the industry’s lowest power floating-point DSP
- Develop and design products easier with low-cost development tools (starting at $149) and pin-for-pin and software compatible options.

All the devices feature:

- C674x fixed- and floating-point DSP core which provides high-precision and wide dynamic range as well as higher performance, efficiency and low power
- Power management software with DVFS to enable various power down modes
- Complementary analog products including the TI TPS65070 power management device with integrated 10-bit, 4-channel ADC, touch screen controller and battery charger implements all sequencing and default options and supports the devices’ power modes
- Object code compatible with all TMS320C67x+™ and TMS320C64x+™devices so developers can leverage existing software code
- Pin-for-pin compatibility (C6742, C6746, C6748 and OMAP-L138) allows for the ability to expand an entire product portfolio using the same hardware and software platform

The OMAP-L138 applications processor consists of an ARM9 and C674x fixed and floating point DSP combined with interface options for SATA, Universal Parallel Port (uPP), Ethernet MAC, USB 2.0HS, USB1.1 FS, Video Input/Output, LCD Controller and much more.

The TMS320C6748/6/2 processors consist of a C674x fixed- and floating-point DSP combined with a variety of networking and interface features including many, if not all, of those found in the OMAP-L138.
– C64x fixed-point multi-core family

TMS320C64x DSPs Performance Value DSPs

C64x Performance Value DSPs offer speeds up to 1 GHz offer a wide range of performance and peripheral options.

- Up to 1GHz / 8000 MMACS of processing performance
- Up to 1MB of L2 memory on-chip
- Starting at $10 (1ku quantity)
- Smallest package C64x DSP achieved with compact 16mm x 16mm footprint
- Significant BOM savings from on-chip integration of EMAC, PCI, HPI, McBSP, McA 12C, UART, PWM, GPIO, Utopia, EMIF, and 64-bit timers.
- C64x+ core enables 20% higher cycle performance compared to the previous generation C64x core
- 20-30% smaller code size from 16-bit compact instructions and SP LOOP buffer

C64x™ DSP Applications

- Networked applications, printers, scanners, copiers, Software Defined Radio, pub safety, medical diagnostics, industrial, Military/Aero/Space, communications and emerging.

– Multi-core versions

TMS320C647x Multicore DSPs

Devices

TMS320C647x™ high-performance DSP

Options of up to 6 integrated cores on a single die
Highest performing multicore DSP with 4.2 GHz performance
Includes industry’s best power efficiency with 3 GHz at 0.15 mW/MIPS
100% code compatible with TI’s single core DSPs based on C64x or C64x+ cores
TI’s SmartReflex™ technology reduces device power consumption by tweaking the core voltage once the chip is integrated into the system.
Serializer/deserializer (SERDES) interfaces
SGMII Ethernet MAC (EMAC)
Antennae Interface (AIF)
Serial Rapid I/O (SRIO)
Up to 4.8 MB of L1/L2 RAM
TI’s fastest DDR2 memory interface
C66x fixed/float dual-core family

TMS320C66x high-performance multicore DSPs combine fixed- and floating-point capabilities, delivering innovation and performance at unmatched levels

The industry's highest performing multicore DSPs combine both fixed- and floating-point capabilities for the first time in the TMS320C66x generation of scalable devices. The new generation includes three new multicore processors, the TMS320C6672, TMS320C6674 and TMS320C6678, and a four-core TMS320C6670 communications system-on-a-chip (SoC).

The devices deliver up to 10 GHz of performance at the lowest power levels, making them suitable for developing products for applications such as mission critical, medical and high-end imaging, test and automation, high-performance computing and core networking.

TMS320C66x high-performance multicore DSPs key features include:

- Coprocessing, fixed and floating-point operation and optimized inter-element communication due to TI's KeyStone architecture
- Variety of processor options integrating up to eight C66x cores, including:
  - Value-optimized, dual-core C6672 in 1 GHz or 1.25 GHz speed options
  - Power-optimized C6674 with four cores or C6678 with up to eight cores. Both are available in 1 GHz, or 1.25 GHz speeds
  - Performance-optimized, four core C6670 SoC, in 1 GHz or 1.2 GHz speeds
- Industry's first 10 GHz DSP featuring 320 GMACs and 160 GFLOPs of fixed- and floating-point performance
- 100% code compatibility with C64x™ fixed-point and C67x™ floating-point devices
- Multicore entitlement with features such as Multicore Navigator, Layer 1/2 acceleration, PCI Express, and improved memory architecture
- Full range of peripherals and power optimization techniques
- Field-hardened compiler and development tools, robust ecosystem and numerous hardware vendor partnerships
Design considerations:

- Ease of programming shortens development time by weeks or months resulting in reduced time to market.
- Energy efficient DSP devices as low as 38mW standby, 467mW active power consumption.
- Floating point instructions for applications that require high precision, wide dynamic range.
- Connectivity peripherals for BOM cost reduction, including Ethernet MAC, UPP, SATA, USB, VPIF, etc.
- Code compatibility across full line of C6000 DSPs – C64x, C64x+, C67x and C674x DSPs.
- Industrial/automotive temp support from -40 to 125 deg C.

Applications such as:

- Medical imaging
- Test and measurement
- Military
- Audio
- Communication
- Industrial

A Brief History of DSP in the Context of Real-Time Processing
Chapter 1 • Overview of Real-Time Digital Signal Processing

**DSP**
- C645x
- C641x
- C674x
- C642x
- C672x
- C671x

- DSP with focus on intensive Signal Processing
  - Power-efficient fixed/floating pt DSP devices; 38mW to 467mW
  - High-performance DSP devices; up to 8000 MMAC @ 1GHz
  - Excels at communications, industrial, military, medical, T&M and audio processing

**DSP + ARM**
- C6L138/OMAP-L138
- C6L137/OMAP-L137

- DSP with integrated Controller/Host Processing
  - Low power (38mW standby; <520mW typical active) fixed and floating point DSP with ARM core
  - Ideal for industrial, audio and communications apps

**DSP/ARM + Video**
- DM6467/T
- OMAP3530/25
- DM644x
- DM3xx
- DM643x
- DM64x

- Video and Imaging Processors
  - High performance multi-format video up to 1080p
  - Ideal for Video, Imaging and Vision applications

Featured Product
A Brief History of DSP in the Context of Real-Time Processing

– C55x fixed-point family

**TMS320C55x™ DSPs - featuring the TMS320C5505, TMS320C5504, TMS320C5514 and TMS320C5515 DSPs**

- 100% code-compatible with C55x devices
- Broad portfolio of the industry’s most power-efficient 16-bit DSPs with standby power as low as 0.15 mW/MHz and performance up to 600 MIPS
- Lowest industry standby power greatly extends battery life
- Software compatible with all C5000™ DSPs
- Easy-to-use software and development tools speed time-to-market

**C55x DSP Applications**

- Feature-rich, miniaturized personal and portable products; 2G, 2.5G and 3G cell phones and basestations; digital audio players; digital still cameras; electronic books, voice recognition, GPS receivers; fingerprint / pattern recognition; wireless modems; headsets; biometrics.
Communications and Wireless

- In the past two key application areas, telephone line modems and cellular voice/data communications, have had a major impact on DSPs sales
- In 1994 when the V.34 modem standard (28.8 kbps) was approved fixed point (integer) DSPs became a popular design route
  - The V.34 standard allowed for data rates from 2400 to 28,800 bps
  - The modulation scheme at 28,800 is a high-order quadrature amplitude modulation (QAM) employing shell mapping, trellis encoding, nonlinear precoding, and soft decision Viterbi decoding
  - All of the above and backward compatibility (lower rate schemes) can be performed by one 50 MIPS fixed point DSP
  - In late 1994 these modems cost ~$400, today we know that 28,800 modems (dial-up) are not used much at all
- In the cellular telephone arena fixed point DSPs are widely used in the portable handsets
  - For example the TI C55x series is popular because it has features such as low power dissipation, integration of

---

memory and peripherals, integrated Viterbi accelerator, and small packaging, just to name a few

- Other vendors have tailored their DSP product lines for the high volume cellular handset markets as well

- DSPs are also utilized in VoIP and cellular basestations

- In a portable device market, such as cell phones, the efficiency of a particular processor can be measured in mW/algorithm, which is obtained via

$$\frac{\text{mW}}{\text{algorithm}} = \frac{\text{mA}}{\text{MIPS}} \times \frac{\text{MIPS}}{\text{algorithm}} \times \text{voltage}$$

- MIPS/algorithm requirements for an IS95 handset

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Implementation</th>
<th>MIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlator</td>
<td>Hardware</td>
<td>5</td>
</tr>
<tr>
<td>Automatic frequency control (AFC)</td>
<td>Hardware</td>
<td>5</td>
</tr>
<tr>
<td>Automatic gain control (AGC)</td>
<td>Hardware</td>
<td>5</td>
</tr>
<tr>
<td>Transmit filter</td>
<td>Hardware</td>
<td>30</td>
</tr>
<tr>
<td>128-pt FFT</td>
<td>Software</td>
<td>1</td>
</tr>
<tr>
<td>Viterbi decoder (length 9, rate 1/2)</td>
<td>Software</td>
<td>6</td>
</tr>
<tr>
<td>Vocoder (8-Kbps Qualcomm code excited linear prediction (QCELP))</td>
<td>Software</td>
<td>20</td>
</tr>
<tr>
<td>Vocoder (enhanced variable rate coder (EVRC))</td>
<td>Software</td>
<td>30</td>
</tr>
</tbody>
</table>
DSP Hardware Design Issues\textsuperscript{1}

- Hardware alternatives, FPGA, ASIC
- Using a general purpose DSP
- Fixed-Point DSPS
  - Fixed-point arithmetic
  - Quantization effects and scaling in fixed point
  - Typically a lower power design
  - Typically a lower cost design
- Floating Point DSPS
  - A more rapid design due to the ease of floating point algorithm implementation
  - With the powerful C6x family a few floating point calculations can easily be software emulated on the C62x or C64x
  - With the C67x the floating point units are available, so efficient floating point code can be easily written
- Accessing memory resources
- Integration of peripheral devices
- A very practical alternative today is to consider a hybrid FPGA/general purpose DSP architecture

DSP System Design Flow (Marven & Ewers)

Application

Define System Requirements

Select DSP Device

Software

Code Writing

Debug Code

Hardware

Schematic

Prototype

System Integration

System test and Debug
• Application Design
  – Software design
  – Generating assembler source and using C where possible
  – Testing the code
  – Hardware design
  – System integration

Course Laboratory Foundations

• Of primary focus is the ZOOM OMAP-L138/C6748 DSK
  (open multimedia application platform = OMAP)

• TI TMS320C6713 DSP Starter Kit (DSK)
  – In the past we have used the C6713 DSK with CCS 3.3;
    this target is supported by the text, and can be programmed
    using CCS 4.2 (in theory CCS 5.3 but have not tried yet)

• In the fixed-point arena, we will also spend some time with
  the C55x platform on CCS 5.3, specifically either the C5505
  or C5515

• The OMAP3530 based Beagleboard is available for the final
  project, for those interested in embedded Linux development

• In the future we may have a second daughter card which is
  suitable for software radio work involving in-phase/quadrature
  (I/Q) sampling and/or zero intermediate frequency (IF)
  processing
• Zero IF processing is becoming more popular in wireless chip sets, so this front-end processing approach and others will hopefully be part of this class in the future

Software
• TI integrated development environment *Code Composer Studio*; CCS 5.3 Platinum (all cores) which supports TI DSPs and MSP430 devices
  – The above IDE is based *Eclipse*
• MATLAB release 2012b with the following enhancement for DSP
  – Signal processing toolbox
  – DSP System Toolbox, (includes Filter Design Toolbox)
  – Fixed-Point Toolbox
  – The text supported *MATLAB Real-Time Interface* (documentation to be released very soon)
• *GoldWave* or *Audacity* shareware for .wav audio file manipulation

Test Equipment
• Agilent two-channel 300 MHz digital scope (2 Gsps)
• Agilent function/arbitrary waveform generators; 15 MHz and 80 MHz models
• Agilent 10 Hz – 500 MHz spectrum/vector network analyzers