525.446 : DSP Hardware Lab

Douglas Wenstrand

Joseph Haber

Semester Goals

- DSP Concepts
  - Become re-familiarized with some of the common signal processing topics : Convolution, Filtering, Modulation, Transforms
  - Gain a more concrete understanding of DSP topics by implementing them -- “playing”
  - Through implementation, get a chance to study the application of the above topics
    - Compression, Signal Detection / Estimation, Communications

- DSP Implementation
  - **Understand** the concept of embedded systems
  - **Understand** the architectural features of a DSP, and what types of applications they enable
  - Be able to program in C and Assembly Language to **most effectively** use the DSP to solve your real-world problems
  - Select **appropriate** DSP’s for problems
  - **Understand** how a DSP processor fits into a hardware system

Grading

- Lab Projects : 75%
  - Several assignments will be handed out throughout the semester (with defined due-dates). These assignments will involve implementation of a particular function on the hardware.

- In Class Labs / Quizzes : 25%
  - Post-lab quizzes (based on the lab)
  - Pre-lab quizzes (based on assigned reading)
  - Quick, in-class assignments

  Anytime appropriate, most of the time with warning

Topics

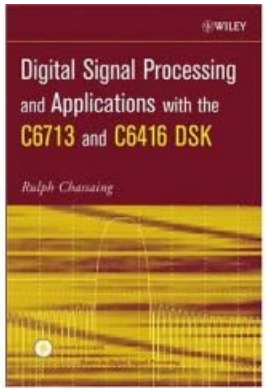
- Class will generally follow the text
  - Text is tied directly to our hardware
- Power Management
- DMA in far more detail than the book
- High Efficiency Coding in a Pipelined System
- Hardware design of a dsp-based system
- DSP in FPGA systems

Additional topics to text

Time Permitting

Administrative Details

- Class Webpage : dsp.echelonembedded.com
  - Notes and Assignments and other data

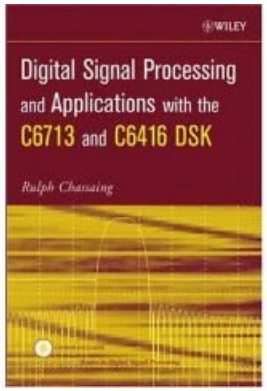


- Textbook:
  - DSP and Applications with the C6713 and C6416 DSK – Chassaing (update this!)

- Make sure you are on the class mailing list
  - dspforum@echelonembedded.com
  - We use this as a mailing list for sending information and answering questions.
  - Ask questions to the group and answer each other.
  - Emails on file will be invited to the list

Administrative Details

- Each week, lecture will generally be used to introduce topics necessary for the lab, and will generally be 1 hour or so.
  - Remainder is lab time, use it!
- Labs will be typically assigned weekly. Due date is listed on the lab
  - Extensions for travel, illness will be granted with advance request.
  - Grading is generally done via demonstration in the laboratory



As is customary in a lab class, students are allowed and expected to help one another with problems that arise. Copying of lab assignments and / or ripping large segments of code from the internet is neither expected nor allowed.

Introduction

Embedded Systems

DSP architecture

TMS320C6713 DSK

Dictionary Definitions

microprocessor - (Or "micro") A computer whose entire CPU is contained on one (or a small number of) integrated circuits.

The important characteristics of a microprocessor are the widths of its internal and external address bus and data bus (and instruction), its clock rate and its instruction set. Processors are also often classified as either RISC or CISC.

microprocessor - (Or "micro") Any CPU that is contained on a single chip.

This little chip is the heart of a computer. Often referred to as just the "processor," the microprocessor does all the computations like adding, subtracting, multiplying, and dividing. In PCs, the most popular microprocessor used is the Intel Pentium chip, whereas Macintosh computers use the PowerPC chip (developed by Motorola, IBM, and Apple).

4,8,16,32, DSP, RISC, CISC???

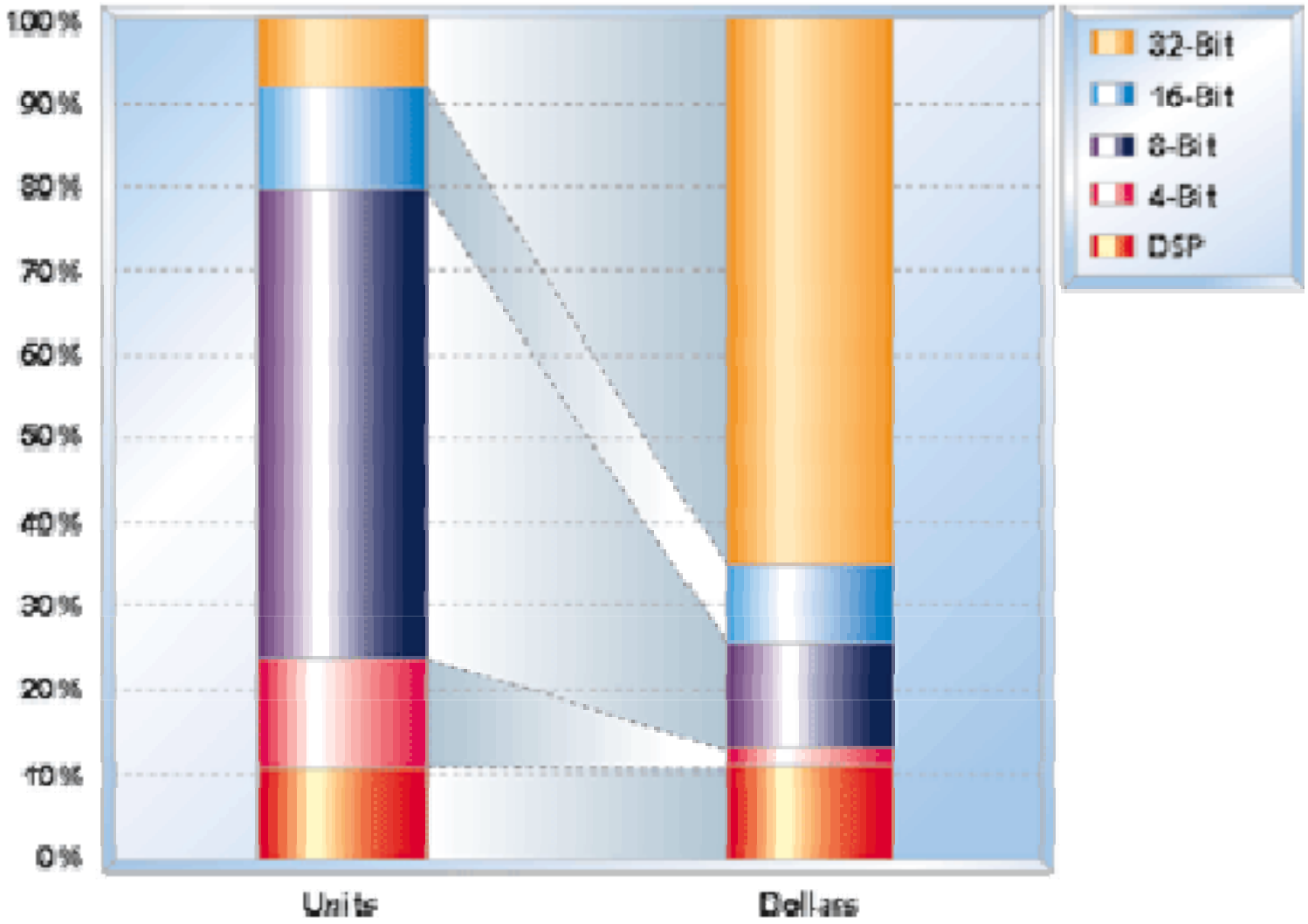
- A processor is frequently categorized based on **the width of its busses**.
- Clock Rate (i.e. at what rate does the processor execute instructions)
- Complexity of Instruction Set
  - CISC : Complex Instruction Set Computer
  - RISC : Reduced Instruction Set Computer

Interesting Facts

- Estimated that approximately 2% of all silicon sales are from processors…but 30% of the profits
- Estimated that under 2% of 32-bit processors sold end up in traditional computers
- How many of processors sold today are 32-bit (or 64 for that matter)?

http://www.embedded.com/showArticle.jhtml?articleID=9900861

Processor Breakdown by sales



Source : Embedded Systems Programming Magazine

Microprocessor Invasion

Denon Home Theater In a Box
Analog Devices SHARC DSP

Maytag “Fuzzy Logic” Washer
8051 Variant

Kodak Digital Camera
DSP/uP Combo Chip

I-River MP3 Player / FM Tuner

Wireless Phones (Samsung Pictured)
TI OMAP DSP/uP Combo Chip

Microprocessor Systems in Automobiles

The first car to use a microprocessor was the 1978 Cadillac Seville.
The chip, a modified 6802, drove the car's "Trip Computer," a flashy dashboard bauble that displayed mileage and other trivia. (source ESP : 8/03)

BMW 7-Series : 100 microprocessors on board

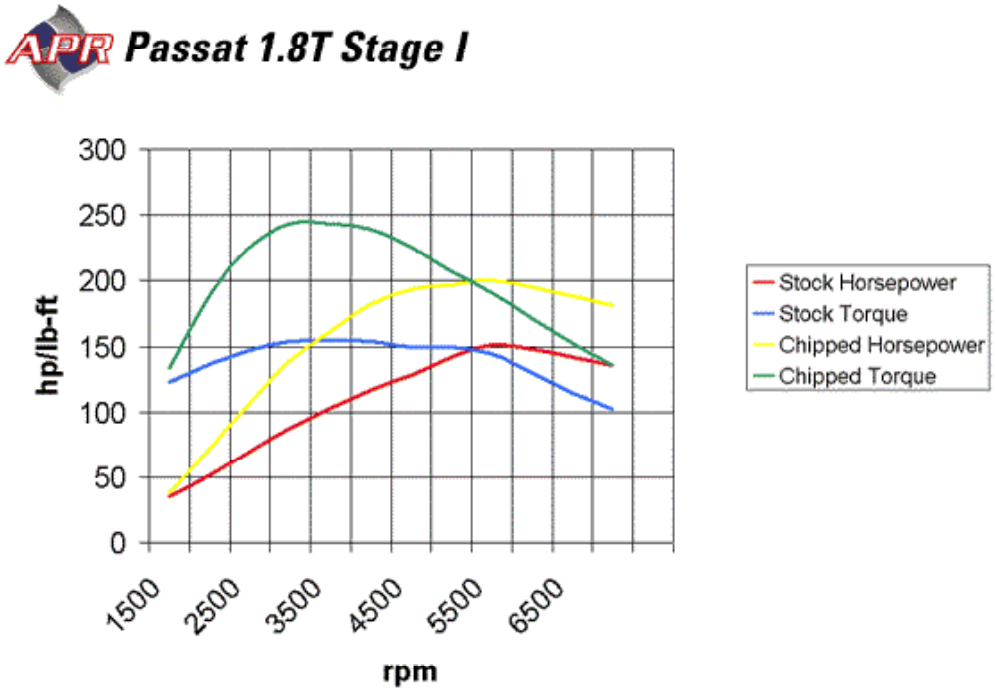
Volvo S40 : 50-60

Toyota Echo : 30-40

All Numbers Estimated
ESP (8/03)

uP’s in automobiles

- Cruise Control
- Intermittent Wipers
- Control / Memory for power seats/mirrors
- On Board Diagnostics and Readout
- Navigation System
- Engine Control (ECU)
- (etc)



DSP in Automobiles

- Navigation
- Entertainment
  - XM Radio
- Lane Departure Warnings
- Object Detection Systems
- Cell Phones
  - Speech Recognition
- Occupant Classification

Embedded Systems Characteristics

- Real-Time
  - Real, defined timing requirements for particular actions to be accomplished
- Event Driven
  - Actions of the system are in response to events, not a predefined sequence.
- Resource constrained
  - Memory Size, speed, power constrained
- Special purpose
  - Device must only perform certain well defined tasks

Example

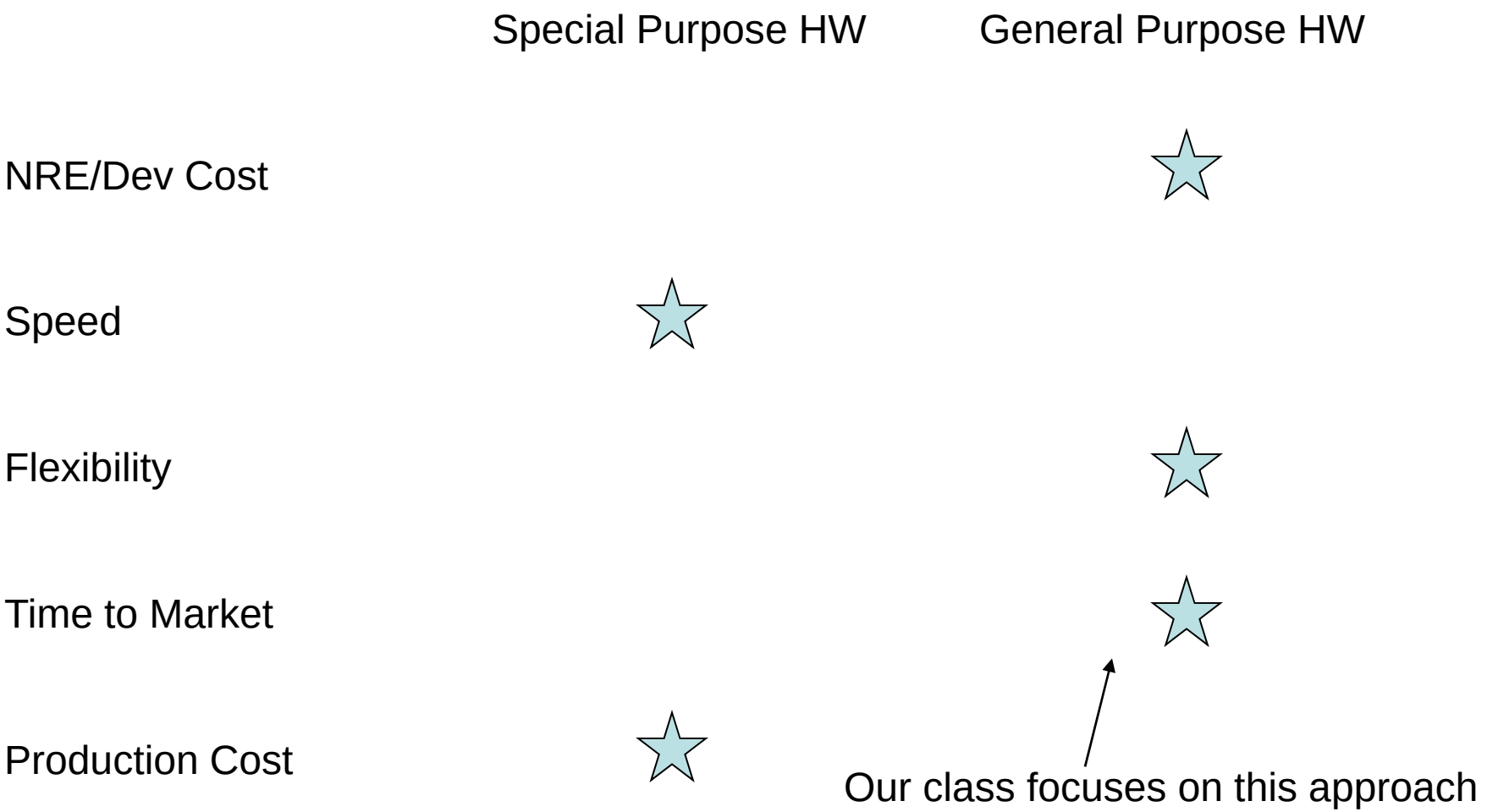
Denon Home Theater In a Box
Analog Devices SHARC DSP

- Events :
  - Button Press
  - Knob Turned
  - New Sample needed by D/A converter
  - Data block available from CD drive

Design Approaches for Digital Systems

- Special Purpose Hardware
  - Custom IC
  - ASIC
- Software Programmable Processor
  - Pentium
  - PowerPC
  - Thousands of Others
- FPGA (possibly with embedded general purpose microprocessor)

Metric Summary



A DSP is a “general purpose” processor with features specifically designed to make Signal processing applications fast and efficient

Old Style Embedded SW Design Flow

- Develop Code which is to be run on Target processor
- Since target is minimal (doesn't have much memory, I/O…etc. Code development takes place on a separate machine. (a PC)
    - Cross Compiler / Assembler
    - Simulator
- Code is then run in the target system and observed. Debugging is done based on performance, and any debug support programmed into the software.

Emulation / Debugging

One needs to be able to interactively see how code behaves in the real system. After initial code development is done, testing of the code commences on the actual target system (and usually target processor).

- In-Circuit Emulator
- Debugger Kernel / BIOS
- Background Debug Mode
- JTAG Emulation
- Debugger
  - Interactively Run Code
    - Breakpoints
    - Single Step
    - Watch Variables
  - Observe how code interacts with rest of target system

Development environment is frequently processor specific, less uniform that what you might expect with PC development tools, and more prone to bugs and errors. Since the users are engineers, these things are typically tolerated

Differences between Embedded & Traditional Programming

Micros designed for embedded market frequently include peripherals on-board

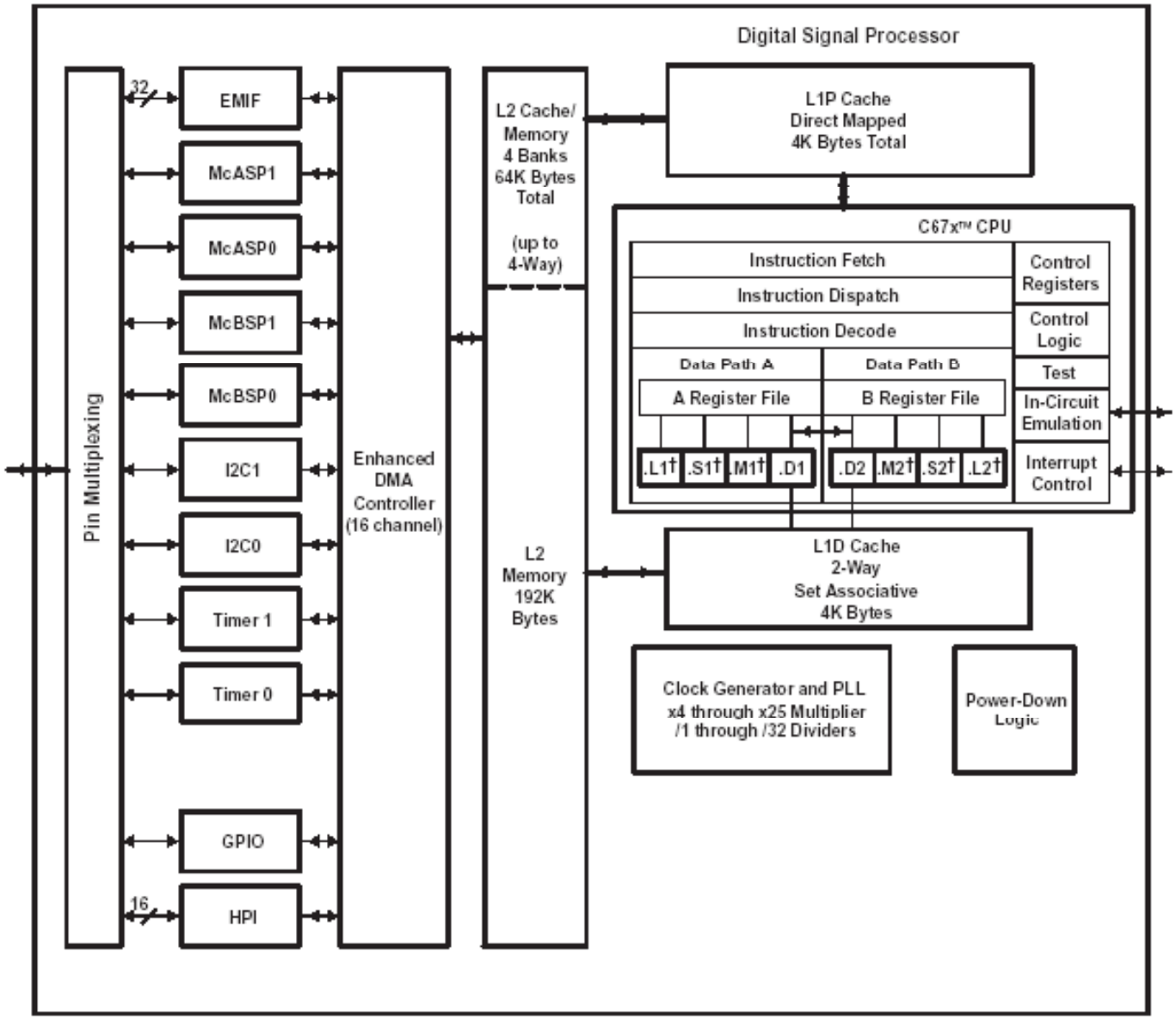
- Tailored towards a target market
- A/D Converters
- Timers / Counters
- Interrupt Controllers
- General Purpose I/O pins
- Pulse-Width Modulators
- Serial Ports (Buffered / Unbuffered)
- Bus / Protocol Interfaces (IrDA, Ethernet, USB, PCI)

Device drivers are often time consuming

•peripherals (ex: A/D, Timer, Serial Ports, etc…) generally require lots of register twiddling

•Frequently there are "App Notes" from manufacturer with simple setup code for those who just want to do the basics.

TI TMS320C6713 DSP



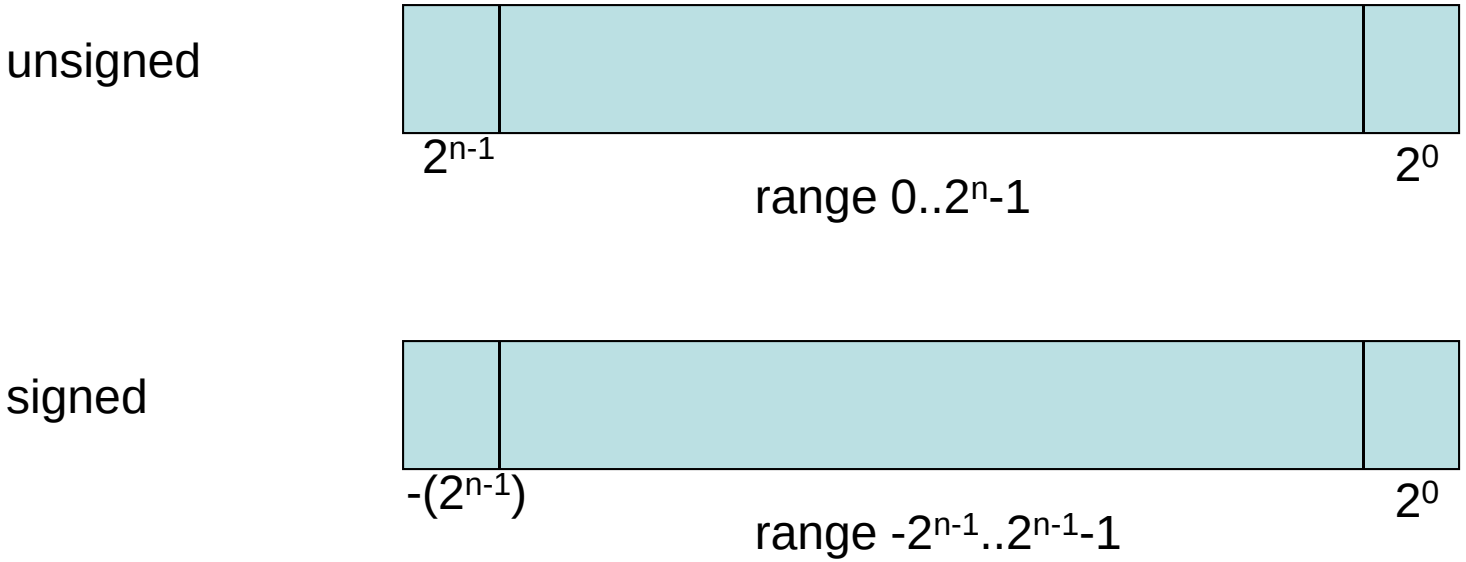
TI TMS320C6713 DSP
Key Features

- DMA Controller
- Serial Ports (I/O)
- Multiple Computation Units
- Cache
- On-chip PLL
- Host Port Interface
- Timers
- Floating Point Units

Basic Numbering Formats

- Three main numbering formats that we need to know:
  - unsigned representation
  - 2’s complement representation (for signed types)
  - floating point representations
- Fixed point representations of fractions
  - Saturating arithmetic
  - Multiplication of fractions

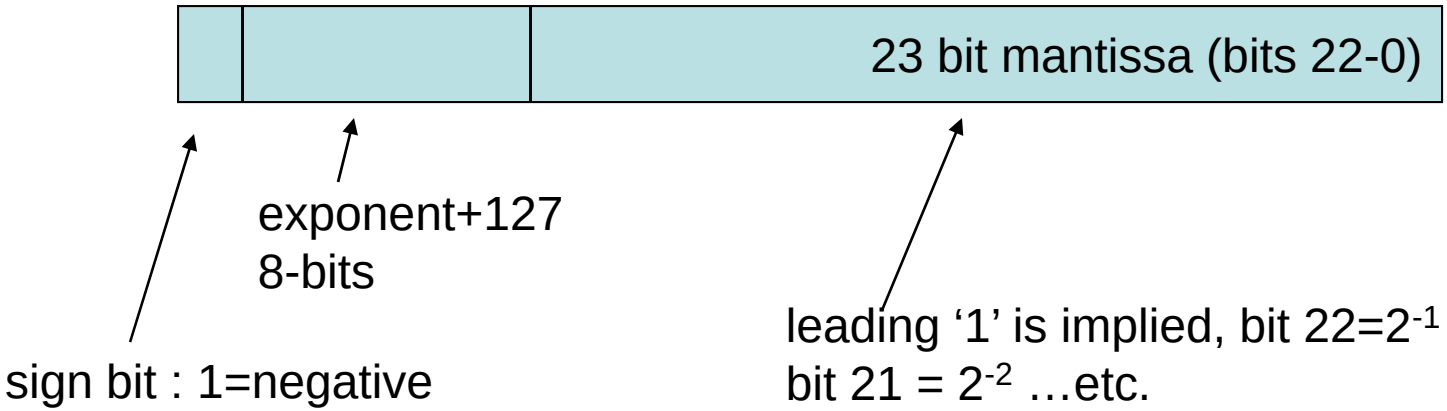
Standard signed and unsigned representations



Arithmetic operations are the same, so the same hardware handles both types

IEEE Std Floating Point Representation

- IEEE floating point numbers have three basic components: the sign, the exponent, and the mantissa. The mantissa is composed of the *fraction* and an implicit leading digit (explained below). The exponent base (2) is implicit and need not be stored.
- single (32-bit) and double (64-bit) precision floating-point standards exist



Lets represent the number 525 : 1000001101. This is like saying 1.000001101 e+9
so => 01000100000000110100000000000000

A processor with a floating point unit, or *floating point DSP*, handles these data types Natively (in hardware).

Floating Point vs. Integer

- Why Floating Point?
- Exercise
  - Evaluate power between TMS320C6713 (225MHz) and TMS320C6416T (1GHZ)
- Often with creative design (often lots of thought involved) one can do same thing with integer math.
- Floating point makes programming easier
  - DSP system designs from Matlab port over free from numbering system changes
  - Cost = Power / Speed

The TMS320C6713 DSK

The TMS320C6713 DSK

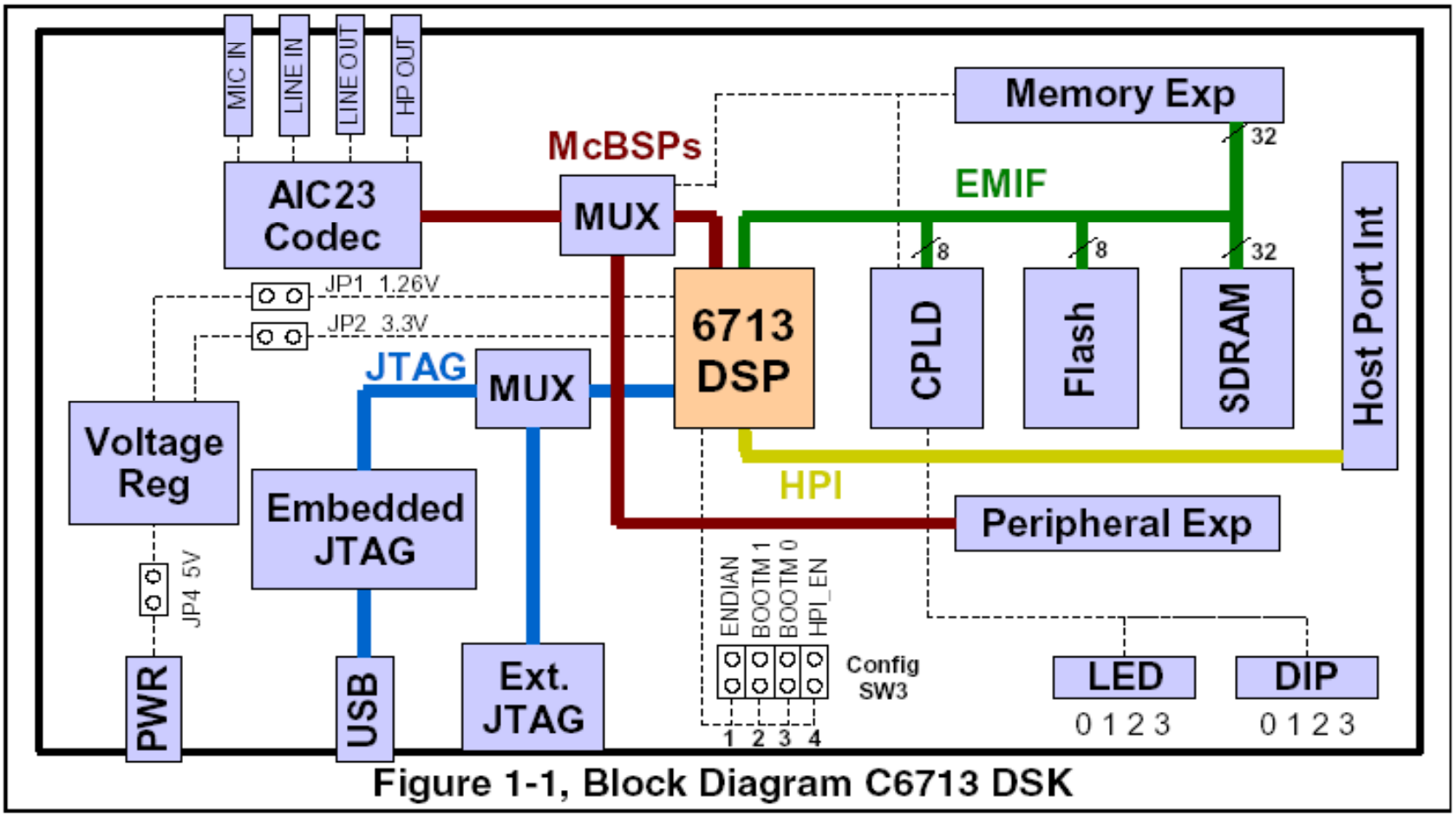


Figure 1-1, Block Diagram C6713 DSK

The TMS320C6713 DSK

- DSP “Starter Kit”
  - Most manufacturers make a board like this available for reasonable price for “design wins”
    - Unlikely that someone would design a board around a chip sight unseen.
- Contains the things most often used to exercise the DSP and verify its capabilities for your application
  - Power measurement
  - Benchmarking
- Software frequently included
  - Locked to the DSK

Agenda for Lab Time

If we are lucky, the administrative stuff and brief overview has only taken a short time, the rest of the time will be spent in the lab – K223

- Distribute DSKs
- As a group
  - Start software
  - Connect to board
  - Use the debugger to modify memory, light LEDs, read switches…etc.
  - Load a pre-built project
  - Run / Change / Compile / Re-load
- Work on Lab 1
  - A pair of headphones and/or an audio cable is useful to have for most labs