EECS 318 CAD
Computer Aided Design

LECTURE 1:
DSP Architectures

Instructor: Francis G. Wolff
wolff@eecs.cwru.edu
Case Western Reserve University

This presentation uses powerpoint animation: please viewshow

High Growth Portable Markets

Cellular Phones

- 69% Subscriber Growth
  - 325M in '99
  - 500M in '02
  - upgrades

Portable PCs

- CAGR = 14% ('99 to '02)
  - growth from 20Mu to 26Mu

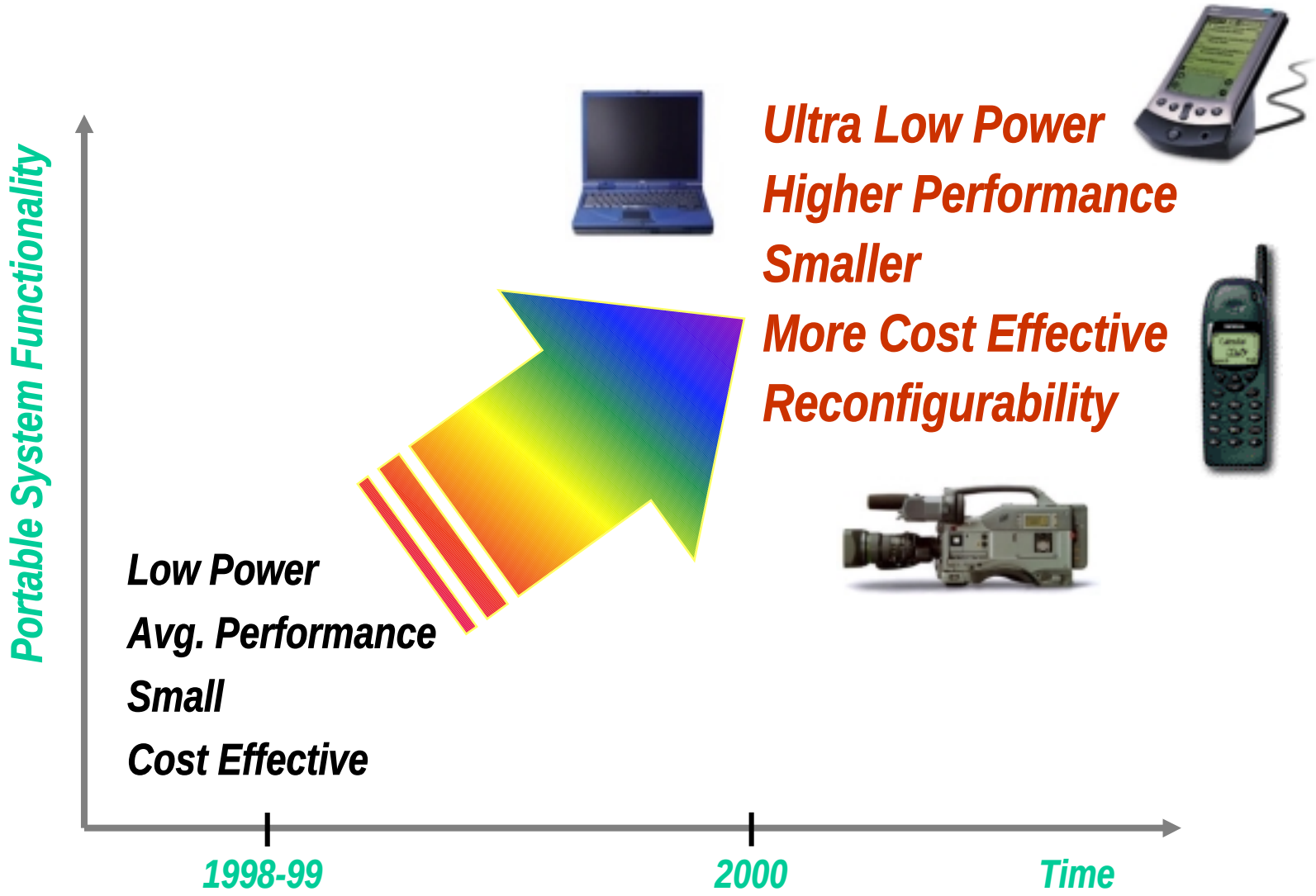
PDAs

- CAGR = 32% ('99 to '02)
  - growing to 14M units/yr in '02

CAGR=Calculated Growth

ref: www.xilinx.com

Portable Applications Demanding “More with Less”



Portable Applications: DSP markets

Industry analysts expect the Internet audio market to grow to more than 30 million units by 2003.

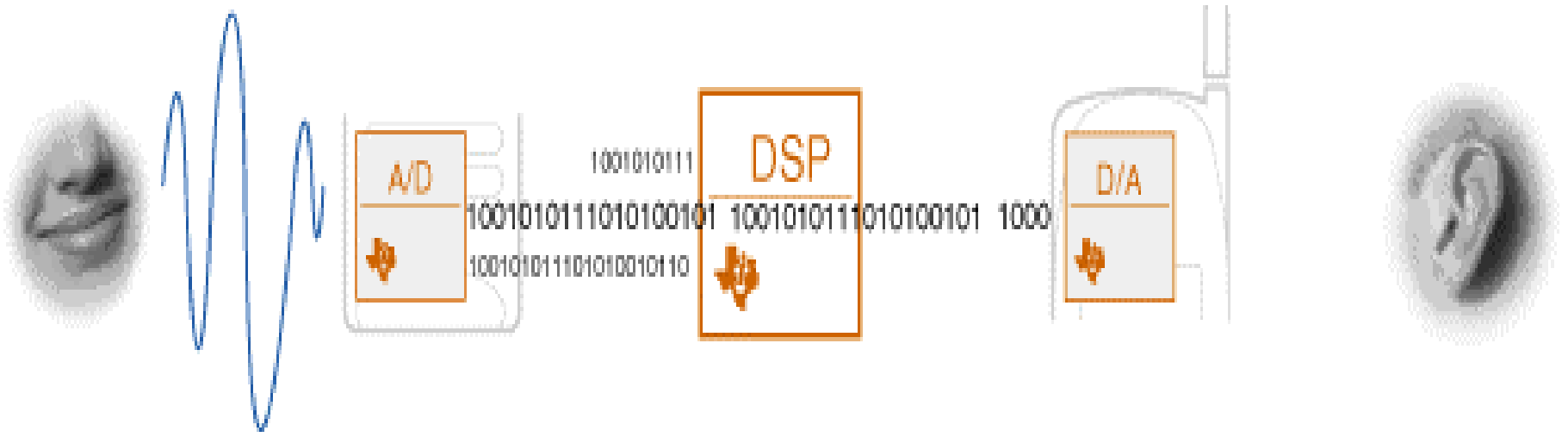
Analysts expect Internet telephony to be a $4 billion market by 2002.

Portable Applications: DSP markets

Eight of the world's top 10 wireless network equipment providers have chosen TI DSPs.

TI forecasts that within five years, advanced wireless devices will reach 15% of the total digital cellular phone market.

Typical DSP application



When you speak, your voice is picked up by an analog sensor in the cell phone's microphone.

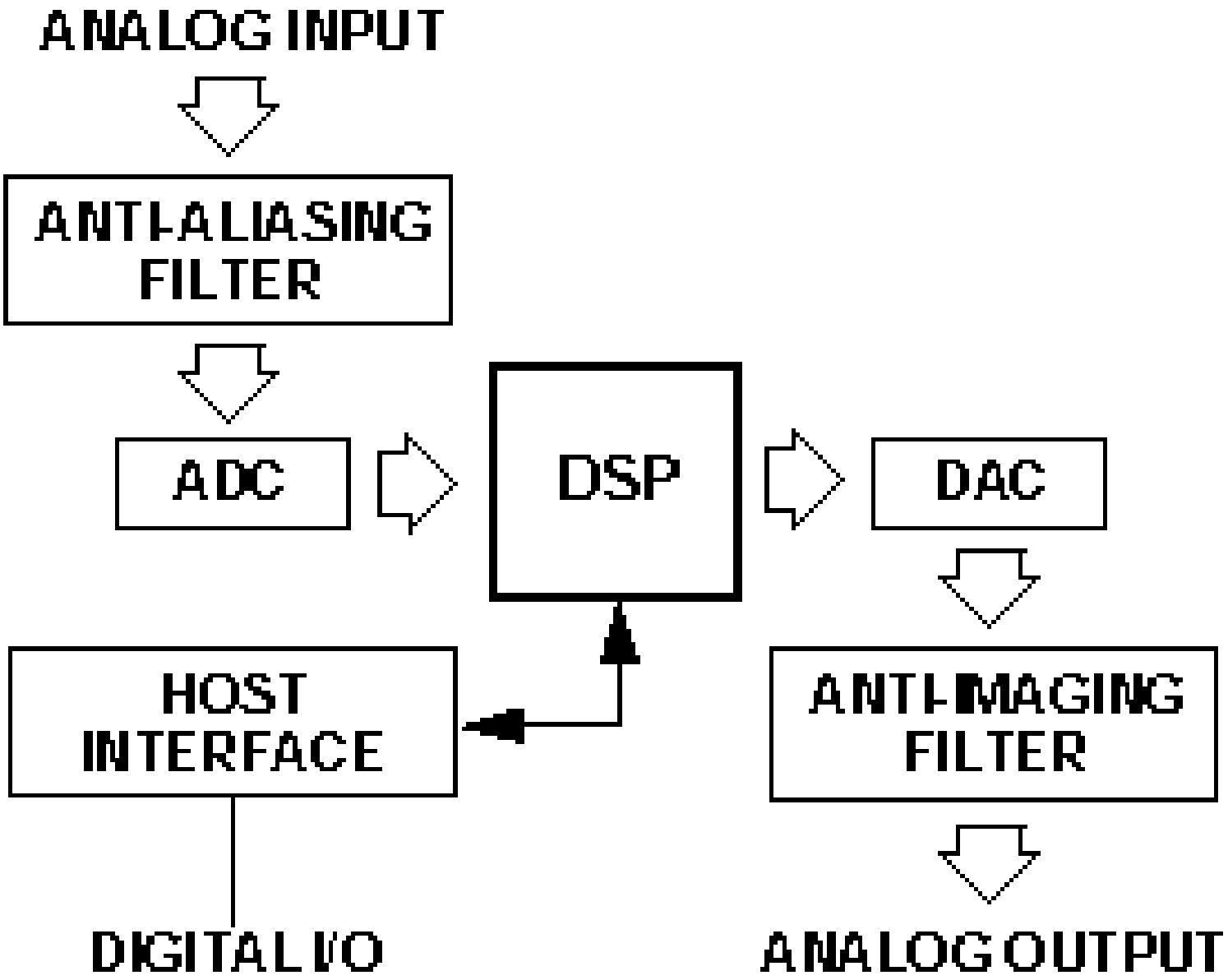
An analog-to-digital converter chip converts your voice, which is an analog signal, into digital signals, represented by 1s and 0s.

The DSP compresses the digital signals and removes any background noise.

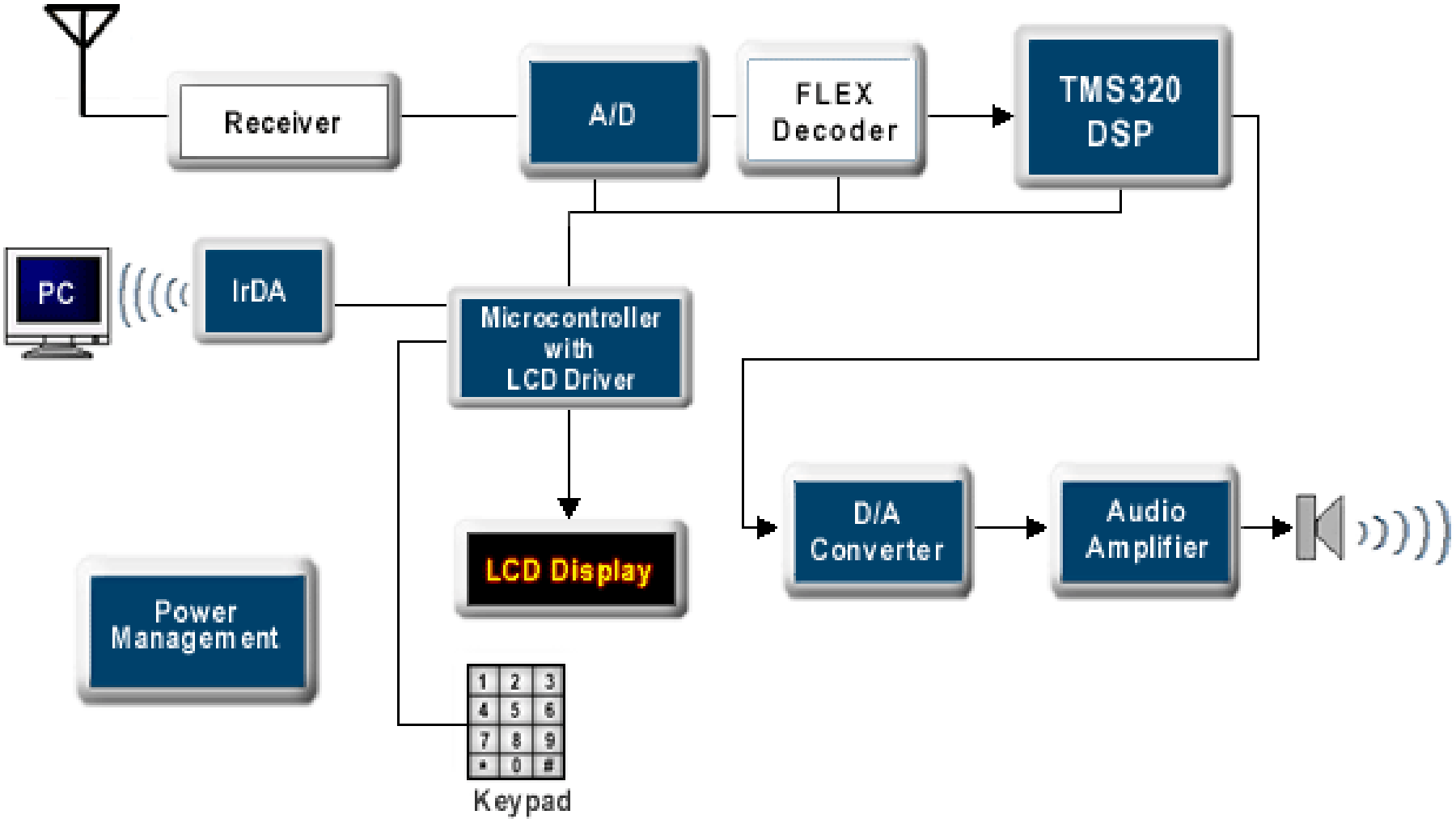
In the listener's cell phone, a digital-to-analog converter chip changes the digital signals back to an analog voice signal.

Your voice exits the phone through the speaker.

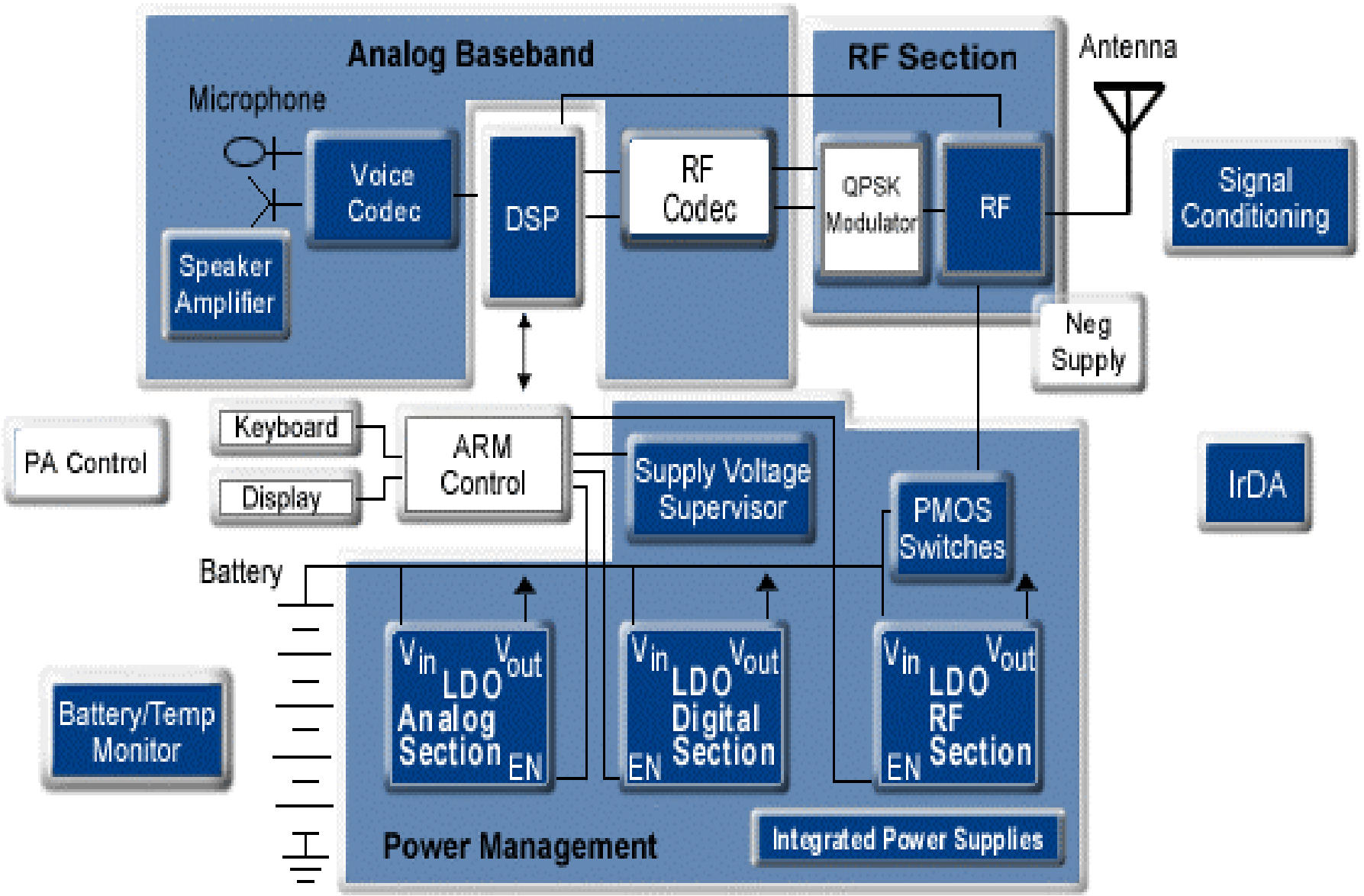
The basic Digital Signal Processing architecture



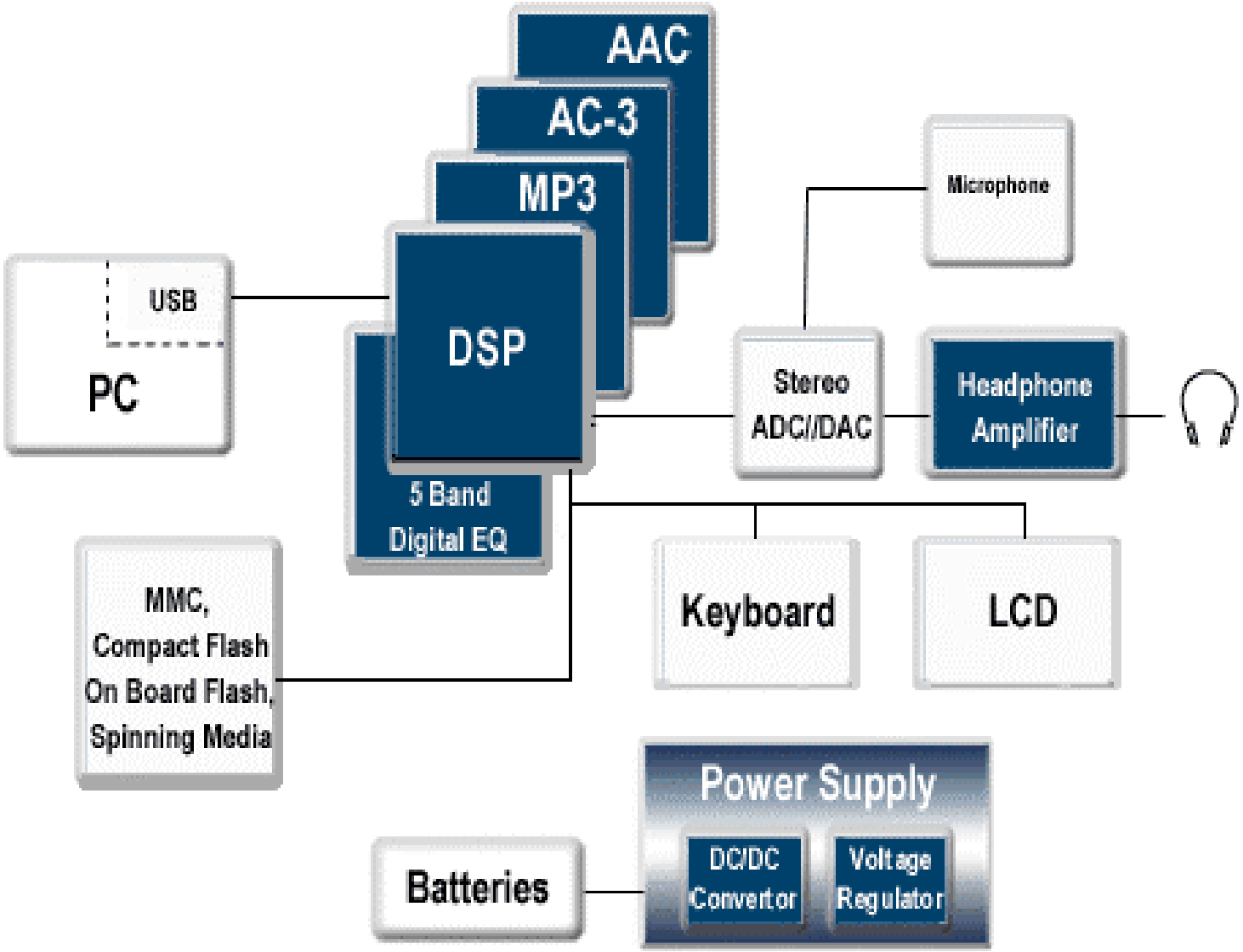
DSP System: Digital Pager



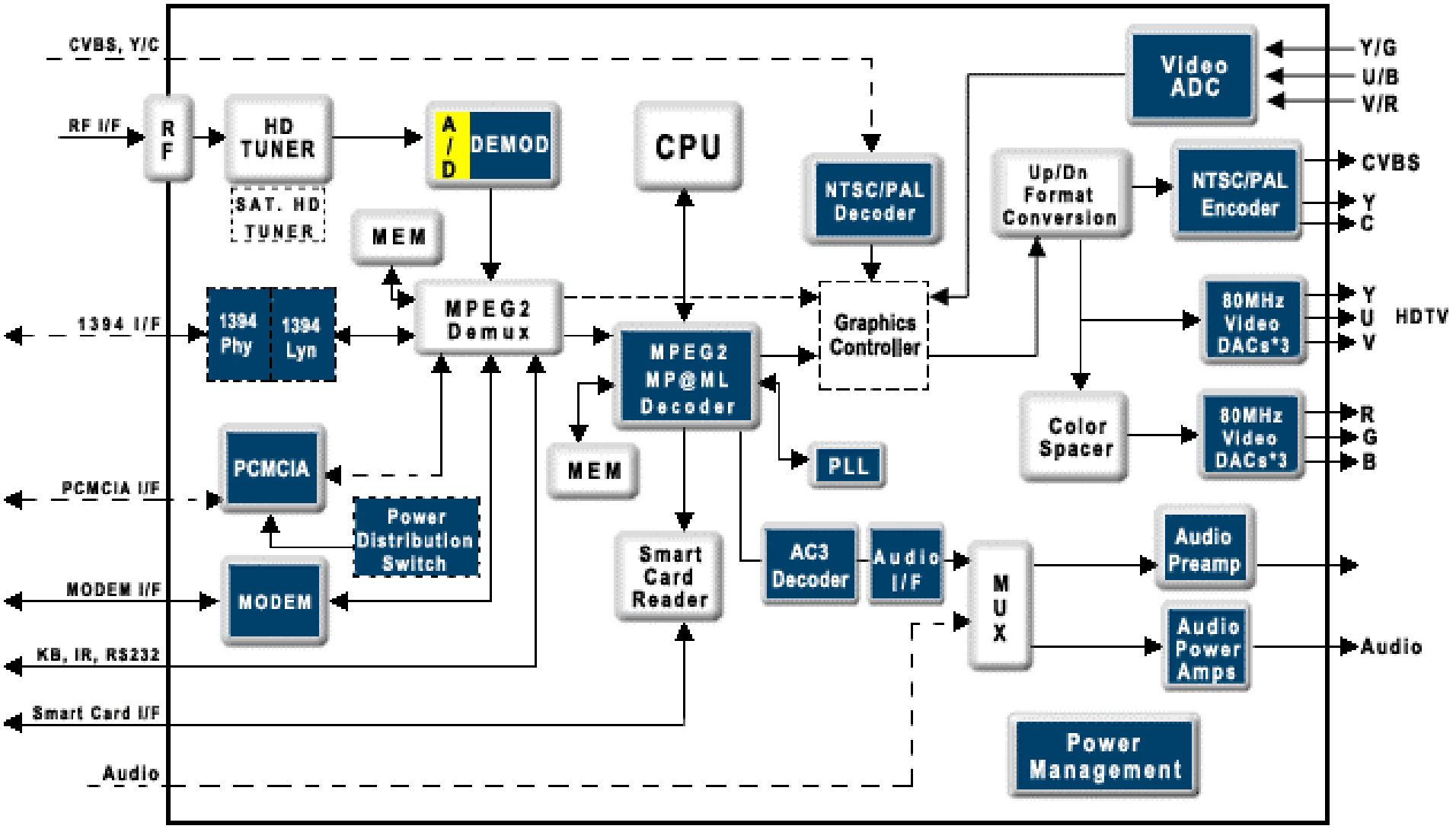
DSP Systems: Cell Phone



DSP Systems: Internet Audio System



DSP Systems: HDTV



TI DSP History: Modem applications

1982 TMS32010, TI introduces its first programmable general-purpose DSP to market

- **Operating at 5 MIPS.**
- **It was ideal for modems and defense applications.**

1988 TMS320C3x, TI introduces the industry's first floating-point DSP.

- **High-performance applications demanding floating-point performance include voice/fax mail, 3-D graphics, bar-code scanners and video conferencing audio and visual systems.**
- **TMS320C1x, the world's first DSP-based hearing aid uses TI's DSP.**

TI DSP History: Telecommunications applications

1989 TMS320C5x, TI introduces highest performance fixed-point DSP generation in the industry, operating at 28 MIPS.

- **The ‘C5x delivers 2 to 4 times the performance of any other fixed-point DSP.**

- **Targeted to the industrial, communications, computer and automotive segments, the ‘C5x DSPs are used mainly in**
  - **cellular and cordless telephones,**
  - **high-speed modems,**
  - **printers and**
  - **copiers.**

TI DSP History: Automobile applications

1992 DSPs become one of the fastest growing segments within the automobile electronics market.

The math-intensive, real-time calculating capabilities of DSPs provide future solutions for

- **active suspension,**
- **closed-loop engine control systems,**
- **intelligent cruise control radar systems,**
- **anti-skid braking systems and**
- **car entertainment systems.**

Cadillac introduces the 1993 model Allante featuring a TI DSP-based road sensing system for a smoother ride, less roll and tighter cornering.

TI DSP History: Hard Disk Drive applications

1994 DSP technology enables the first uniprocessor DSP hard disc drive (HDD) from Maxtor Corp.

the 171-megabyte PCMCIA Type III HDD.

- **By replacing a number of microcontrollers, drive costs were cut by 30 percent while battery life was extended and storage capacity increased.**

- **In 1994, more than 95 percent of all high performance disk drives with a DSP inside contain a TI TMS320 DSP.**

1996 TI's T320C2xLP cDSP technology enables Seagate, one of the world's largest hard disk drive (HDD) maker, to develop the first mainstream 3.5-inch HDD to adopt a uniprocessor DSP design, integrating logic, flash memory, and a DSP core into a single unit.

TI DSP History: Internet applications

1999 Provides the first complete DSP-based solution, for the secure downloading of music off the Internet onto portable audio devices, with Liquid Audio Inc., the Fraunhofer Institute for Integrated Circuits and SanDisk Corp.

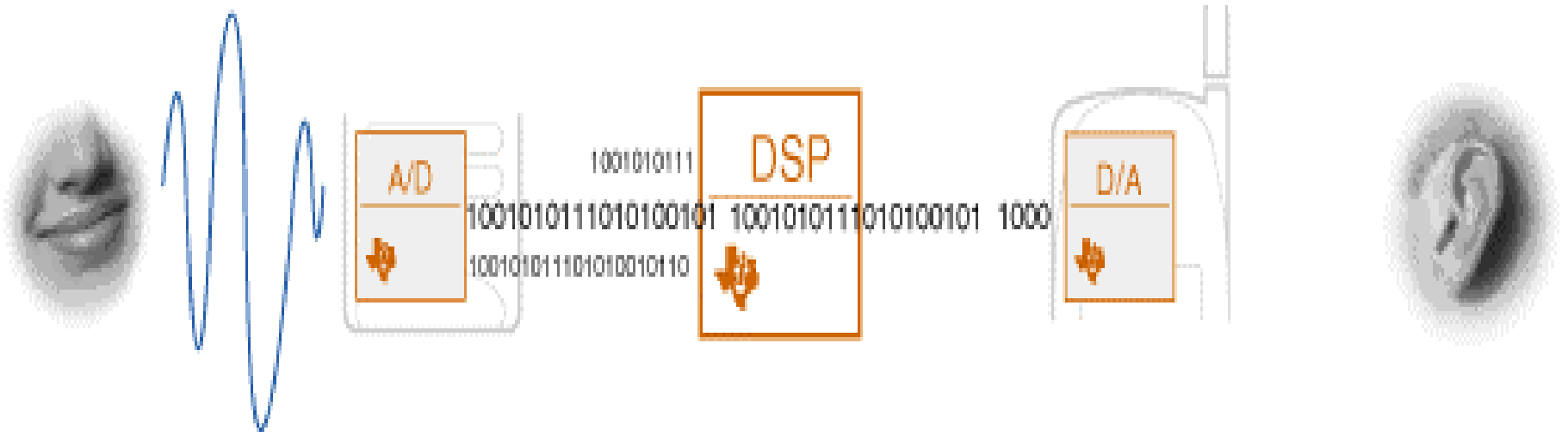
Announces that SANYO Electric Co., Ltd. will deliver the first Secure Digital Music Initiative (SDMI)-compliant portable digital music player based on TI's TMS320C5000 programmable DSPs and Liquid Audio's Secure Portable Player Platform (SP3).

Announces the industry's first 1.2 Volt TMS320C54x DSP that extends the battery life for applications such as cochlear implants, hearing aids and wireless and telephony devices.

Outline

- DSP architectural basics
- Improved performance through increased parallelism
  - Allowing more operations per instruction
    - Enhanced conventional DSPs
    - Single-instruction, multiple-data (SIMD)
  - Issuing multiple instructions per instruction cycle
    - VLIW DSPs
    - Superscalar DSPs
- CPUs with SIMD extensions
- DSP/microcontroller hybrids

Review: Typical DSP application



When you speak, your voice is picked up by an analog sensor in the cell phone's microphone.

An analog-to-digital converter chip converts your voice, which is an analog signal, into digital signals, represented by 1s and 0s.

The DSP compresses the digital signals and removes any background noise.

In the listener's cell phone, a digital-to-analog converter chip changes the digital signals back to an analog voice signal.

Your voice exits the phone through the speaker.

Basic Digital FIR Filter Equation

The two most common real-time digital filters are

(1) Finite Impulse Response (FIR) filter
(2) Infinite Impulse Response (IIR) filter

The basic FIR filter equation is $y[n] = \sum h[k] * x[n-k]$

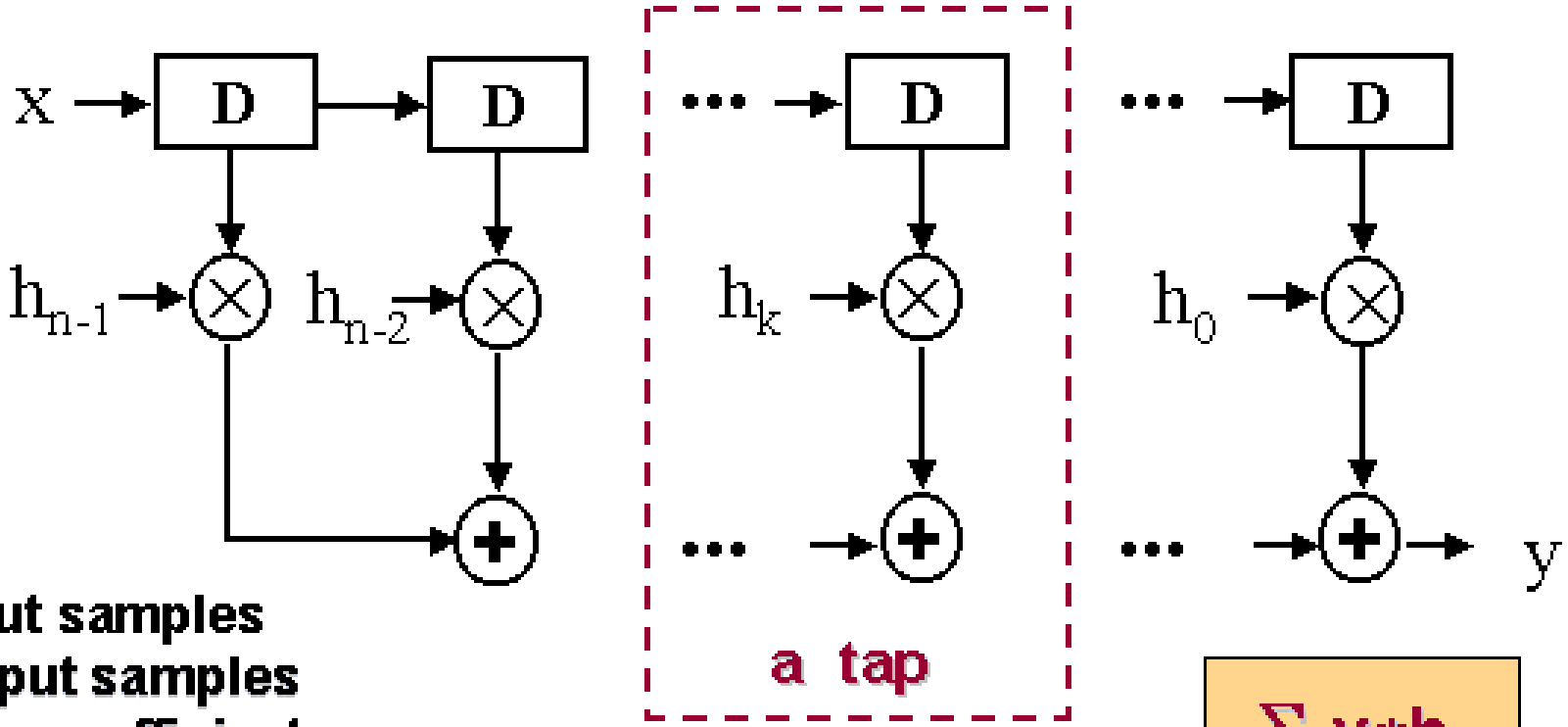
where h[k] is an array of constants

and note that this equation uses only adds and multiplies!

In the C language

```
y[n] = 0.0;
for (k = 0; k < N; k++)
    y[n] = y[n]   +   h[k] * x[n-k];
```

A Motivating Example:
FIR Filtering



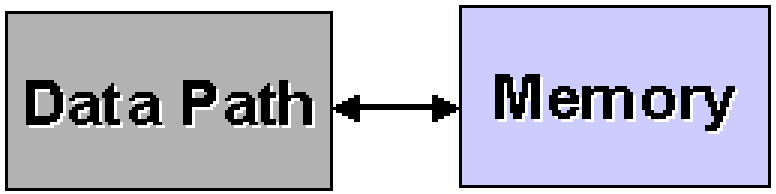
x = input samples
y = output samples
h = filter coefficients
D = unit time delay

Σ x*h

Filter characteristics governed by number of taps, selection of filter coefficients.

FIR Filter on a Typical GPP

```
loop:
        mov     *r0,x0
        mov     *r1,y0
        mpy     x0,y0,a
        add     a,b
        mov     y0,*r2
        inc     r0
        inc     r1
        inc     r2
        dec     ctr
        tst     ctr
        jnz     loop
```

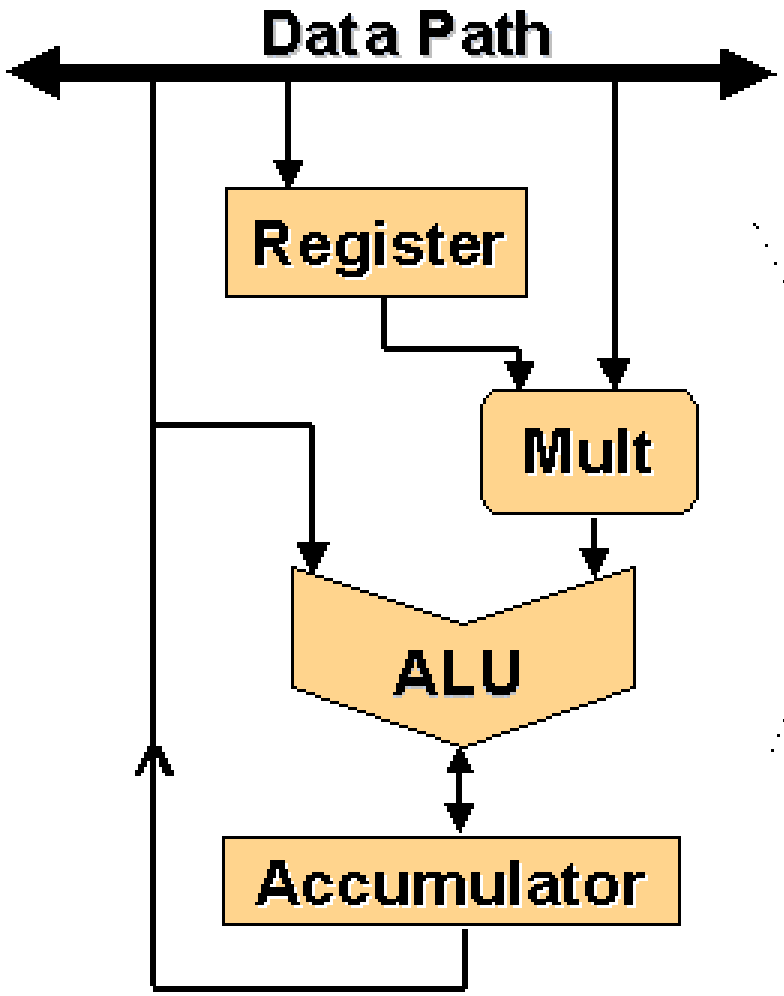


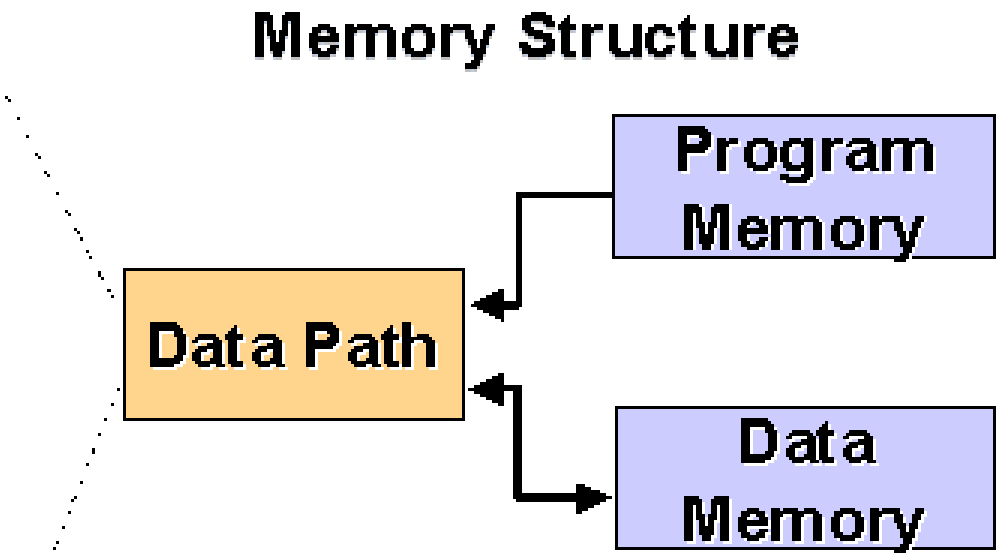
Problems:

- Memory bandwidth bottleneck
- Control code and addressing overhead
- Possibly slow multiply

(Computes one tap per loop iteration)

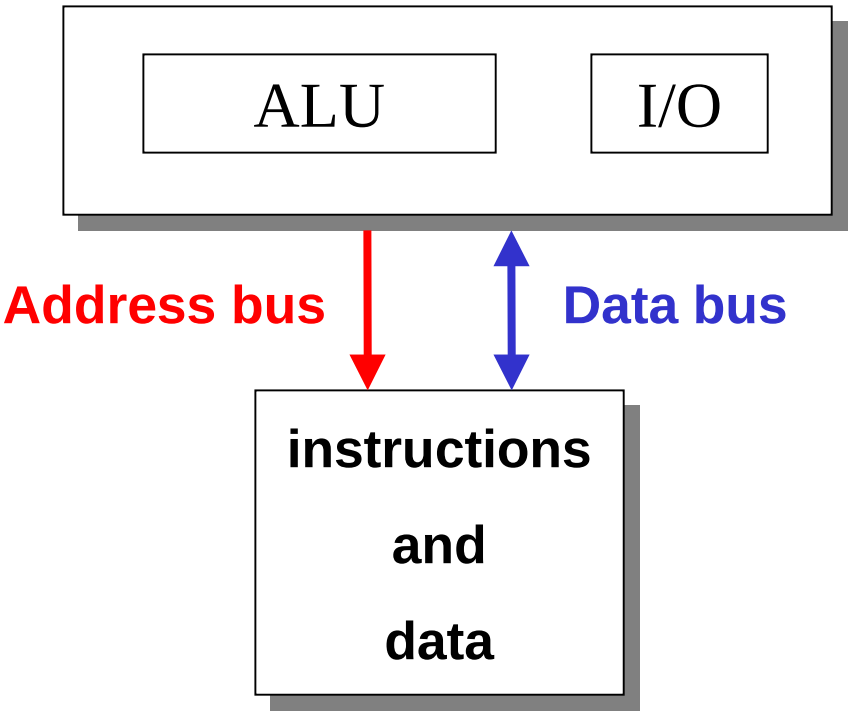
Early DSP Architecture

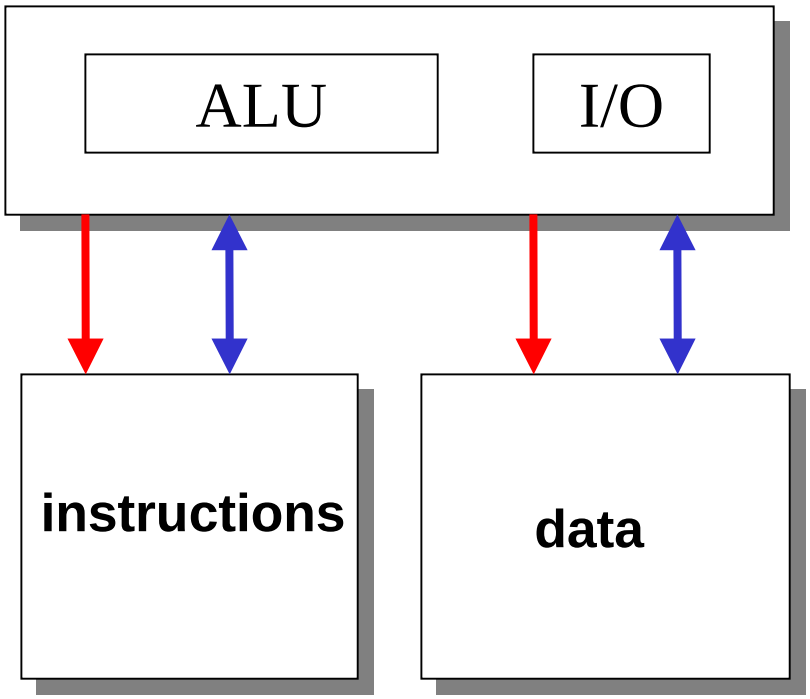




Early Improvements
(1) Datapath Structure
(2) Memory Structure

Memory Stuctures: Von Neuman & Harvard





Von Neuman architecture **Area efficient but requires higher bus bandwidth because instructions and data must compete for memory.**

Harvard architecture **was coined to describe machines with separate memories. Speed efficient: Increased parallelism.**

FIR Filter on a Conventional DSP

A single DSP machine instruction does the following

(1) MAC (2) Fetch next MX0 (3) Fetch next MY0 (4) inc I0 & I4

```
DO dotprod UNTIL CE;
dotprod:
 MR=MR+MX0*MY0(SS),MX0=DM(I0,M0),MY0=PM(I4,M4);
```

MAC: Multiply & Accumulate

MR: multiply Register

DM: Fetch the next data value MX0 from Data Memory

PM: Fetch the next filter tap constant MY0 from Program Memory

Harvard Arch: allows parallel memory read of MX0=Filter Values and MY0=Filter Contants

BDTi

DSP - Architecture Characteristics

DSP architecture is optimized to solve one problem well

- **Digital filters (FIR, IIR, and FFTs)**
- **In Real-Time**

Architecture features added to speed up this problem

MAC: multiply & accumulator, speedup FIR tap
Circular buffer: speedup shifting FIR delay registers
RISC based: single clock per instruction
Harvard Architecture: separate instruction & data
Word orientated

Disadvantages (not a general purpose processor, GPP)

slow character processing
No multi-user operating system support
No virtual memory, no translate look-a-side tables
No memory page protection (Read, Write, Execute)

DSP - MAC Architecture Characteristics

MAC: multiply & accumulator: MAC = A*B +C

Carry-save adders (csa) sums 3 numbers efficiently!

by allowing three values to be computed we can take advantage of the csa technique.

Thus the MAC is not a separated multiplier and adder but a integrated singular design.

This allows easy implementation of $y[n] = \sum c[k] * x[n-k]$

Hence, less area and faster than a separate multiplier and adder.

Baseline: "Conventional DSPs"

- Common attributes:
  - 16- or 24-bit fixed-point (fractional), or 32-bit floating-point arithmetic
  - 16-, 24-, 32-bit instructions
  - One instruction per cycle ("single issue")
  - Complex, "compound" instructions encoding many operations
  - Highly constrained, non-orthogonal architectures

Baseline: "Conventional DSPs"

- Common attributes (cont.):
  - Dedicated addressing hardware w/ specialized addressing modes
  - Multiple-access on-chip memory architecture
  - Dedicated hardware for loops and other execution control
  - Specialized on-chip peripherals and I/O interfaces
  - Low cost, low power, low memory usage

Increasing Parallelism

- Boosting performance beyond the increases afforded by faster clock speeds requires the processor to do more work in every clock cycle. How?
- By increasing the processors' parallelism in one of the following ways:
  - Increase the number of operations that can be performed in each instruction
  - Increase the number of instructions that can be issued and executed in every cycle

Reduced Instruction Set Computer

RISC - Reduced Instruction Set Computer

- By reducing the number of instructions that a processor supports and thereby reducing the complexity of the chip,
- it is possible to make individual instructions execute faster and achieve a net gain in performance
- even though more instructions might be required to accomplish a task.

RISC trades-off

instruction set complexity for instruction execution timing.

RISC Features

- **Large register set: having more registers allows memory access to be minimized.**
- **Load/Store architecture: operating data in memory directly is one of the most expensive in terms of clock cycle.**
- **Fixed length instruction encoding: This simplifies instruction fetching and decoding logic and allows easy implementation of pipelining.**

All instructions are register-to-register format
except Load/Store which access memory

All instructions execute in a single cycle
save branch instructions which require two.

Almost all single instruction size & same format.

Complex Instruction Set Computer

CISC - Complex Instruction Set Computer

Philosophy: **Hardware is always faster than the software.**

Objective: **Instruction set should be as powerful as possible**

With a power instruction set, fewer instructions needed to complete (and less memory) the same task as RISC.

- **CISC was developed at a time (early 60's), when memory technology was not so advanced.**
- **Memory was small (in terms of kilobytes) and expensive.**

But for embedded systems, especially Internet Appliances, memory efficiency comes into play again, especially in chip area and power.

Comparison

CISC	RISC
Any instruction may reference memory	Only load/store references memory
Many instructions & addressing modes	Few instructions & addressing modes
Variable instruction formats	Fixed instruction formats
Single register set	Multiple register sets
Multi-clock cycle instructions	Single-clock cycle instructions
Micro-program interprets instructions	Hardware (FSM) executes instructions
Complexity is in the micro-program	Complexity is in the compiler
Less to no pipelining	Highly pipelined
Program code size small	Program code size large

Which is better

RISC

Or

CISC

?

Analogy (Chakravarty, 1994)

Construct a 5 foot wall

Method A: a large amount of small concrete blocks.

Method B: a few large concrete blocks.

Which method is better?

The amount of work done in either method is equal

Method A: more blocks to stack but easier and faster to carry.

Method B: fewer blocks to stack but the process is slowed by the weight of each block.

The distinction is in the dispersal of the work done.

RISC versus CISC

RISC machines: SUN SPARC, SGI Mips, HP PA-RISC

CISC machines: Intel 80x86, Motorola 680x0

What really distinguishes RISC from CISC these days
lies in the architecture and not in the instruction set.

CISC occurs whenever there is a disparity in speed

- **between CPU operations and memory accesses**
- **due to technology or cost.**

What about combining both ideas?

Intel 8086 Pentium P6 architecture
is externally CISC but internally RISC & CISC!

Intel IA-64 executes many instructions in parallel.