Digital Signal Processing Laboratory

Preeti Rao

DSP Workshop, WEL IITB

14 Dec 2010

Dept. of Electrical Engineering

IIT Bombay

Course Overview

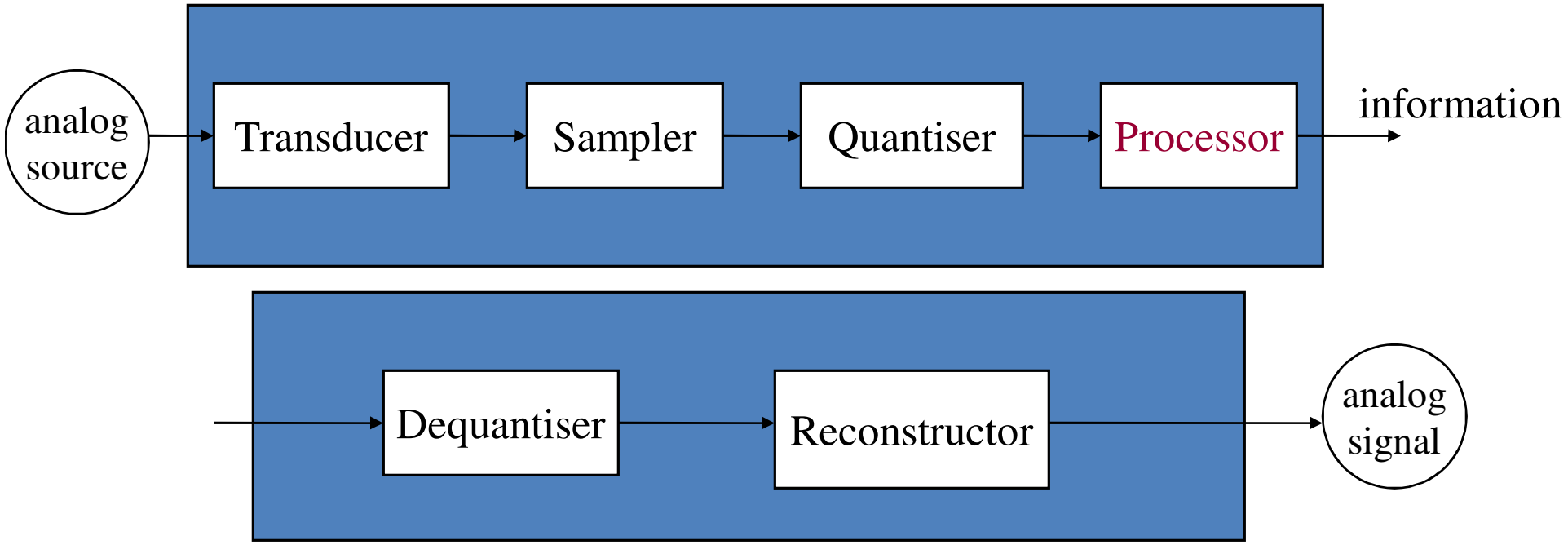
- Objective – to familiarize students with practical aspects of digital signal processing applications
- Introduction to Digital signal processors (DSP)
  - Hardware, Software
- Real-time signal processing
  - Concepts and implementation
- Platforms – Texas Instruments (TI)
  - TMS320C5510 – fixed-point processor
  - TMS320C6713 – floating-point processor

DSP applications

Filtering, coding, transmitting, estimating, detecting, analyzing, recognizing, synthesizing, recording, and reproducing signals by means of digital devices or techniques.

The term "signal" includes audio, video, speech, image, communication, geophysical, sonar, radar, medical, musical, and other signals.

Processing Analog Signals



- Hard real-time constraint

Why special processors?

- Digital Processors
  - DSP
  - General Purpose Processors (GPP)
  - Advanced RISC Machine (ARM) processors
  - Field Programmable Gate Arrays (FPGAs)
- Which one to use ? Why ?
  - How about combining them ?
- Programmable DSPs or DSP cores (embedded)
  - Focus will be on programmable DSPs

GPP Vs DSP

	Data Manipulation	Math Calculation
Typical Applications	Word processing, database management, spread sheets, operating sytems, etc.	Digital Signal Processing, motion control, scientific and engineering simulations, etc.
Main Operations	data movement ($A \rightarrow B$) value testing (*If* $A=B$ *then* ...)	addition ($A+B=C$) multiplication ($A \times B=C$)

Application: Cellular Phone

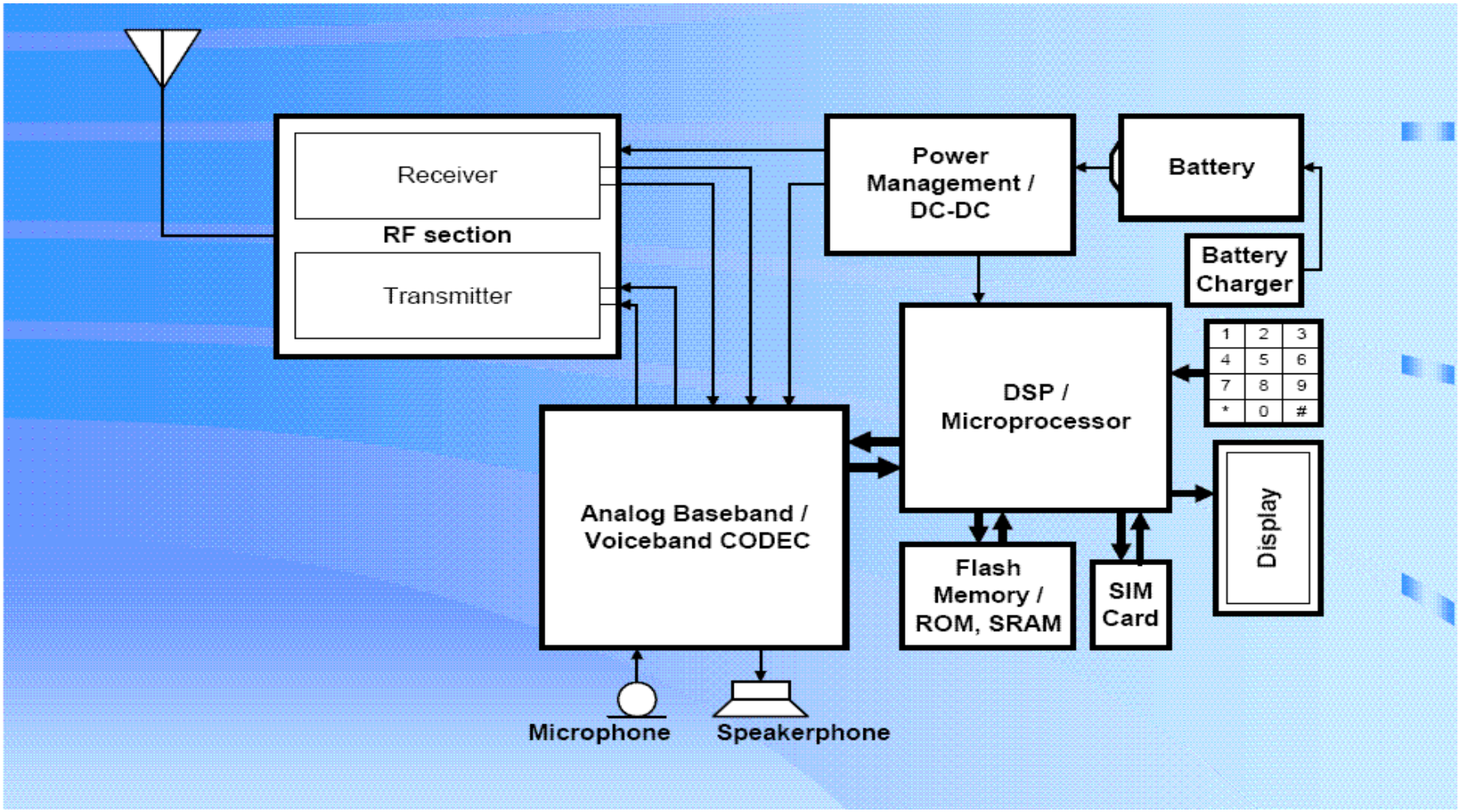


Image taken from TI's website

Application: MP3 Player

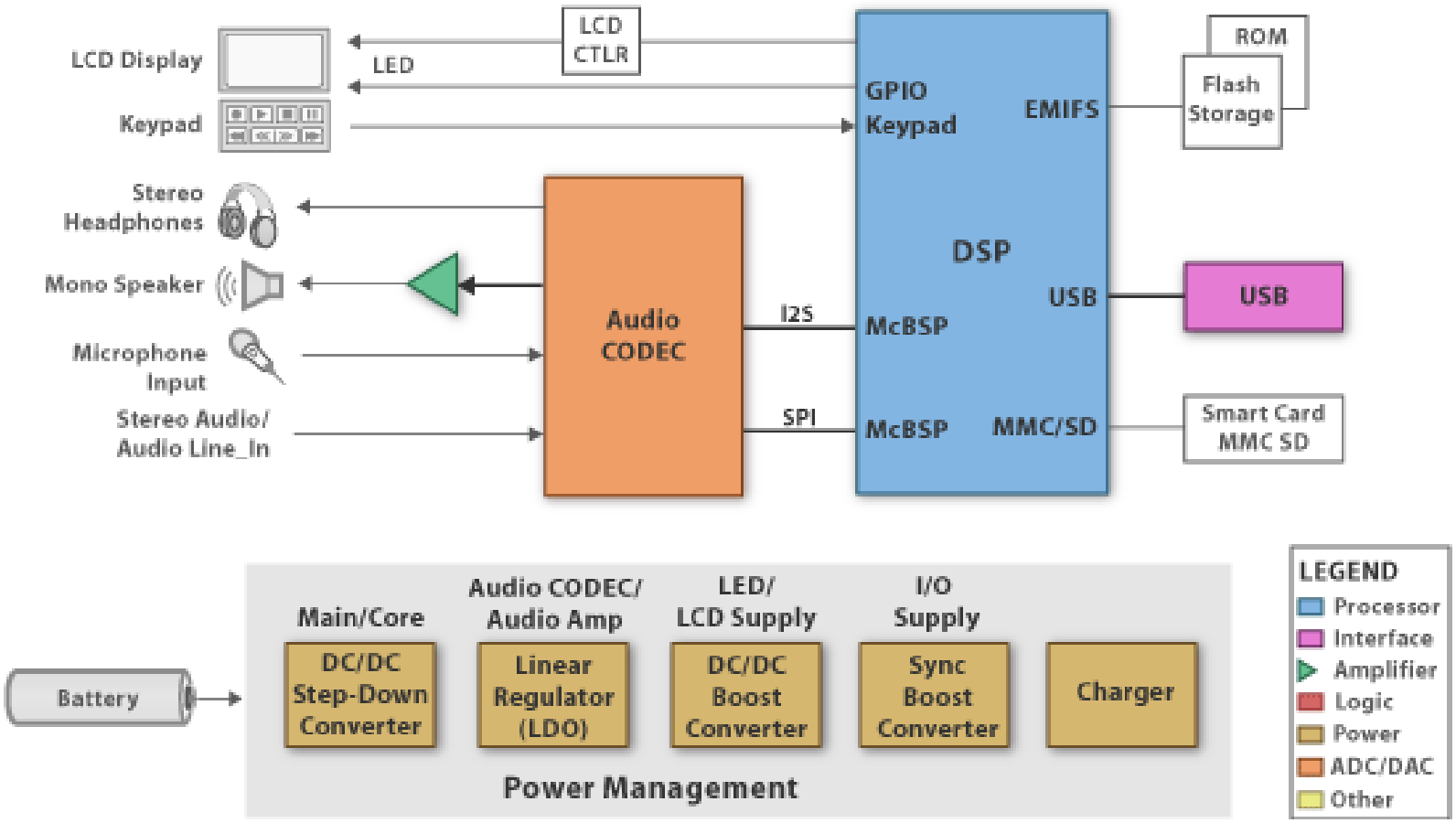
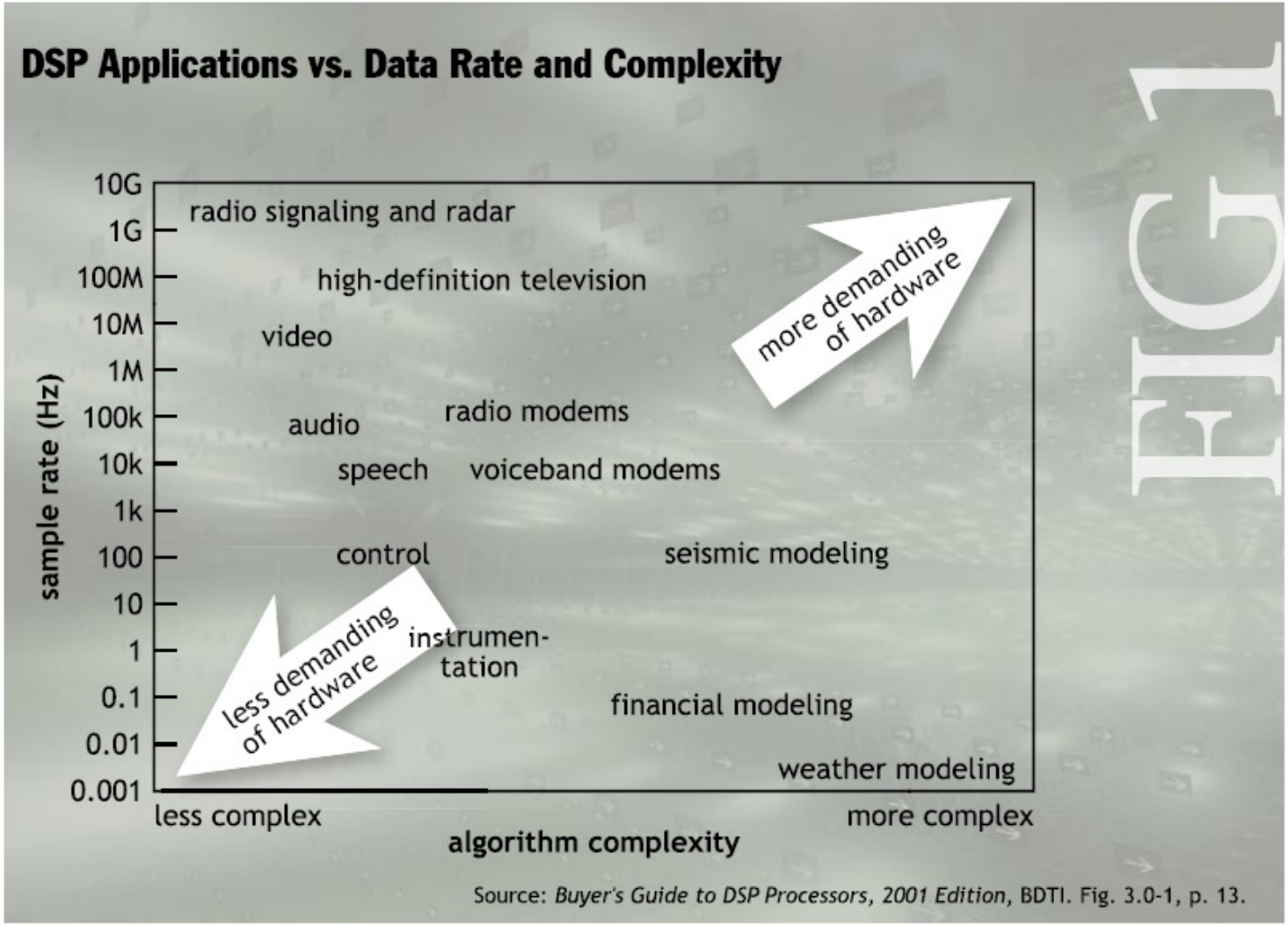


Image taken from TI's website

Complexity of Applications



Signal Processing: Core Operations

Filtering

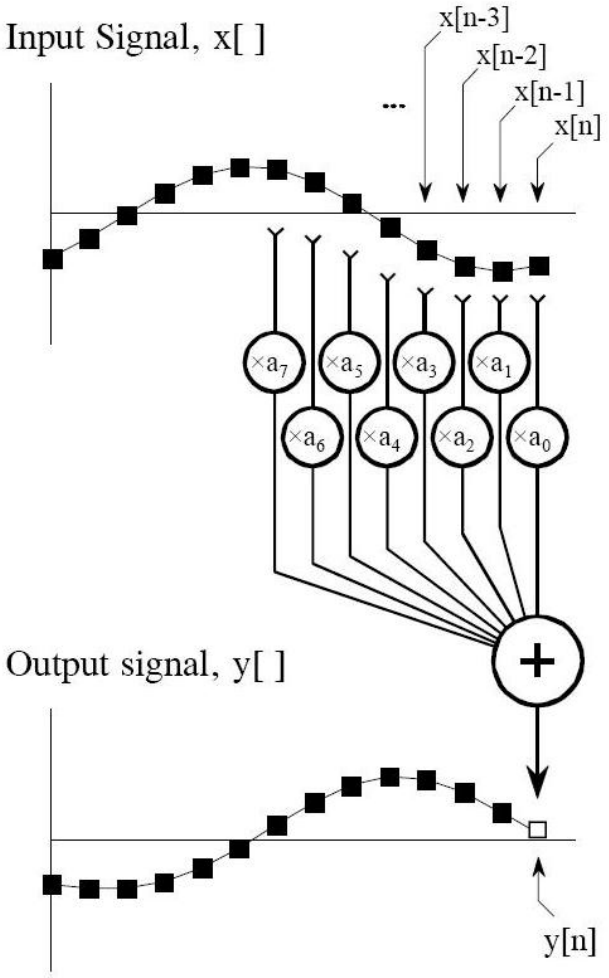
$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

Transform

$$X_S(\omega, n) = \sum_{m=-\infty}^{\infty} x(m)w(n-m)e^{-j\omega m}$$

"Sum of products"

FIR Filter


Input Signal, x[ ]
x[n-3]
x[n-2]
x[n-1]
x[n]
...
×a7
×a5
×a3
×a1
×a6
×a4
×a2
×a0
+
Output signal, y[ ]
y[n]

Sum-of-products on a GPP

- Operations in one step of filtering operation
  - Fetch instruction
  - Fetch program data
  - Fetch data
  - Multiply: shift and add
  - Accumulate
  - Store
  - Shift data in memory
- Looping overhead

DSPs

- Optimized for **arithmetic** and **data-handling** common to all signal processing applications

- A common operation:

$$D = A \times B + C$$

The multiply-accumulate (MAC) is optimally executed by the DSP

bus structure + architecture

DSP - Constraints

- Real-time operation
- Minimum power consumption
- Smaller size (chip area)
- Minimum cost

Use GPP when…

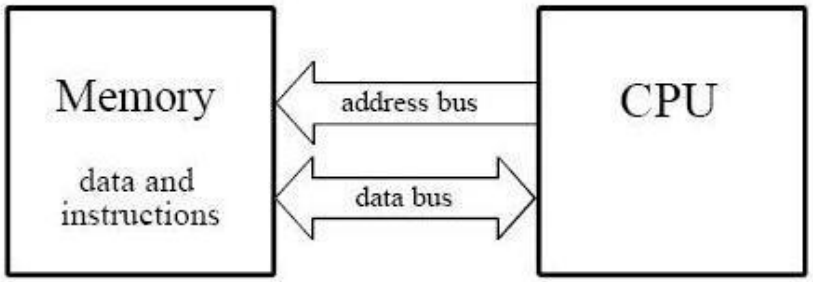
- Large memory
- Advanced operating systems
- Real-time is not a constraint

Salient Features of DSPs (1/2)

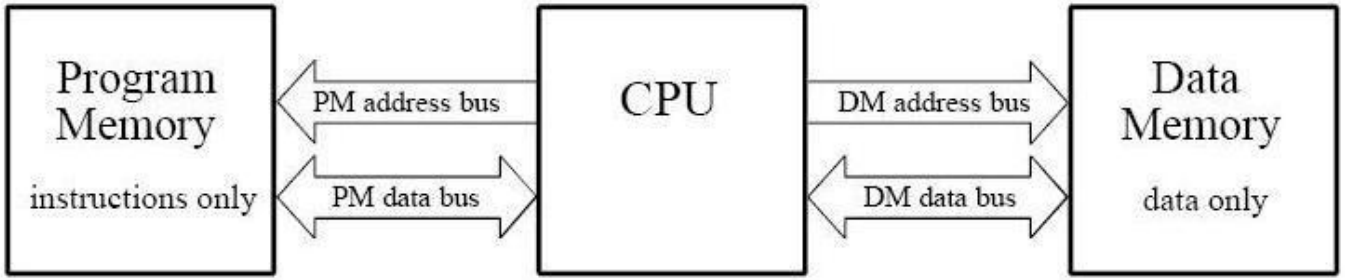
- MAC (Multiply-Accumulator)
  - Used in digital filters and algorithms involving vector dot product
- Efficient memory access
  - Harvard Architecture
  - Pipelining
  - Program Cache memory
- Streamlined input/output (I/O)

Typical Processor Architecture

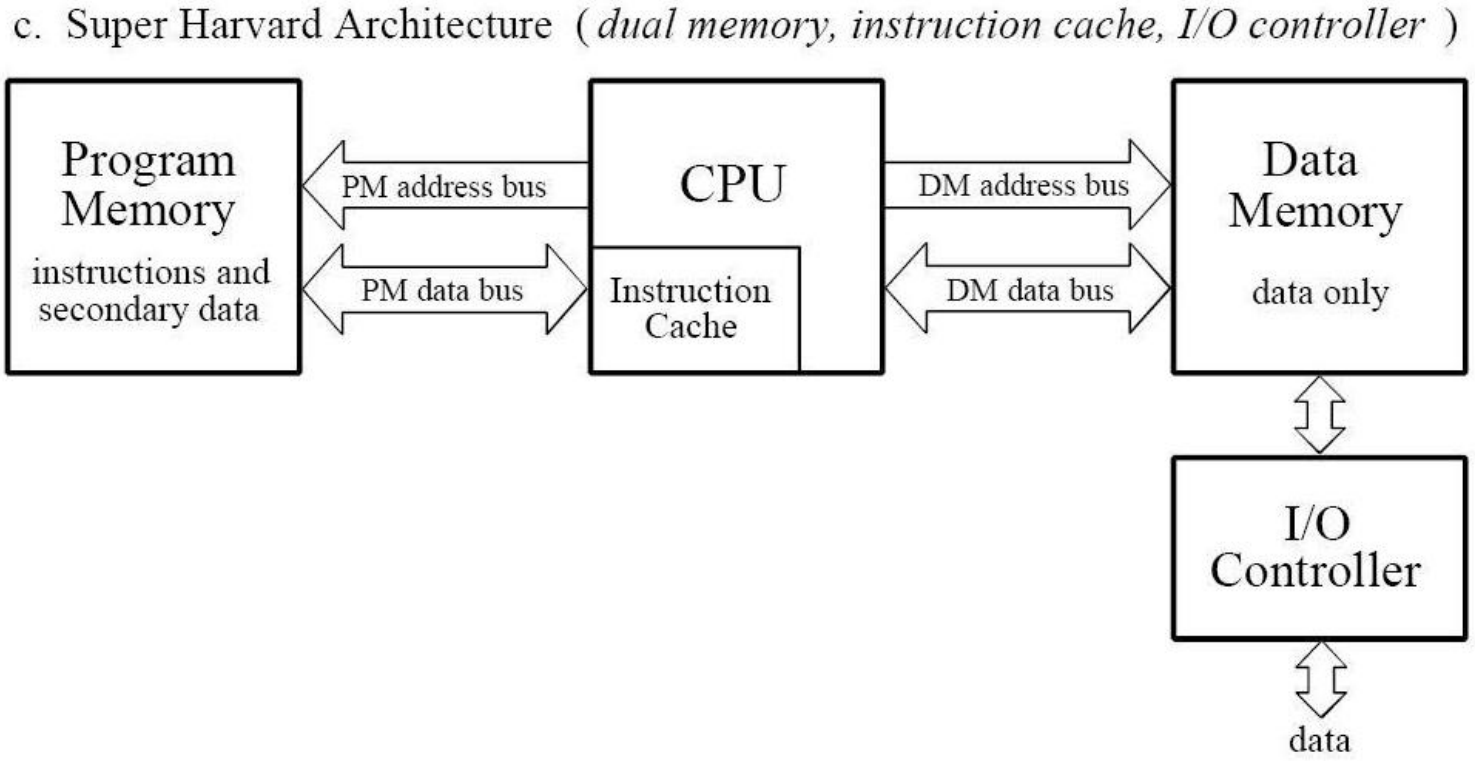
a. Von Neumann Architecture (*single memory*)



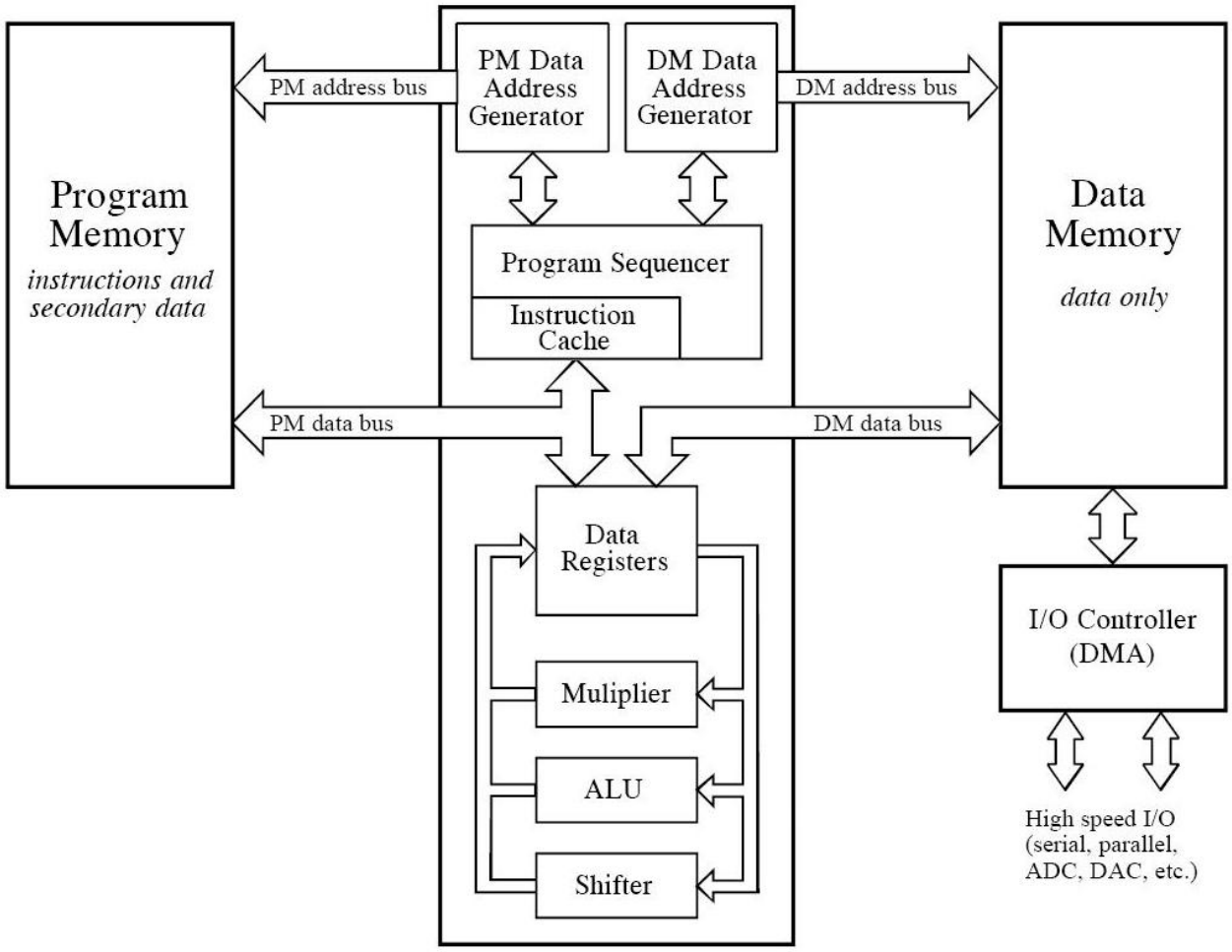
b. Harvard Architecture (*dual memory*)



Typical Processor Architecture



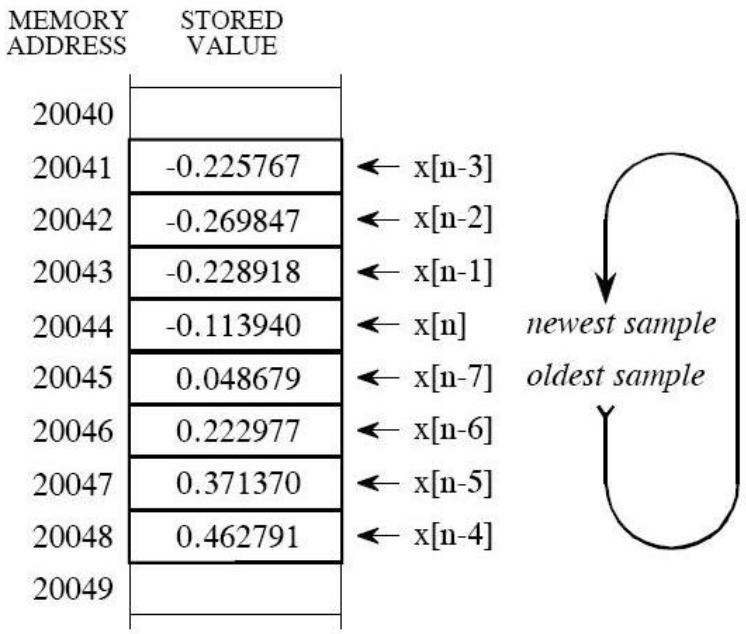
CPU Details


Program Memory
instructions and secondary data
PM address bus
PM Data Address Generator
DM Data Address Generator
DM address bus
Data Memory
data only
Program Sequencer
Instruction Cache
PM data bus
DM data bus
Data Registers
Muliplier
ALU
Shifter
I/O Controller (DMA)
High speed I/O (serial, parallel, ADC, DAC, etc.)

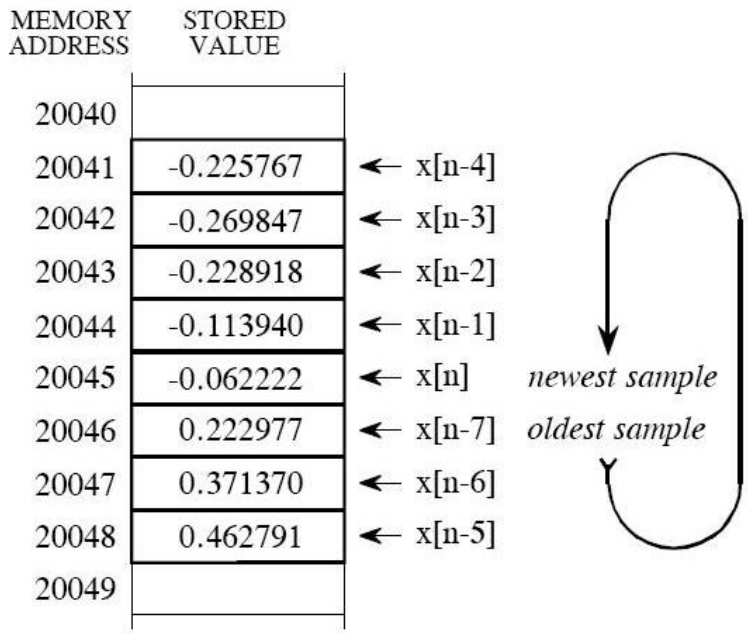
Salient Features of DSPs (2/2)

- Zero-overhead looping
  - Special instruction to implement “for-next” loop without expending any clock cycles
  - Cache memory
- Circular buffering
- Specialized instruction sets

Circular Buffering



a. Circular buffer at some instant



b. Circular buffer after next sample

Trends and Developments in DSPs

- Conventional
  - Single-cycle MAC, multiply operations
- Enhanced
  - Multiple MACs, ALUs
  - Very Long Instruction Words (VLIW)
  - Single instruction multiple data (SIMD)
- Hybrid Design
  - ARM like architectures
  - DSP + GPP
  - FPGA + DSP + ARM
- Multi-core processing

Real-time DSP projects

- Digital audio, speech and music synthesis
  - Synthesize tones of varying frequency, amplitude, phase
  - Undersample tones to see and hear aliasing
  - Linear and non-linear processing: filter, modulation
  - Play a melody as a sequence of tones
  - Design systems for audio effects: echo, reverberation, chorus, equalizer
  - Observe effects of quantization
  - Speech analysis
  - Speech coding by LPC
  - Pitch shifting, time scaling
  - 3-D virtualization of audio

Real-time DSP projects (contd.)

- Digital communication
  - Amplitude and frequency modulation
  - DTMF
  - FSK, BPSK and matched filtering

Development cycle of a DSP project

- Designing the DSP algorithm and simulation with Scilab/ Matlab using stored data. Plot/see/listen to output for verification.
- Use DSP kit development tools to write the corresponding C code. Compile, link and create executable to run on DSP.
- Apply external signal input, monitor output. Use profile, debug options as needed.

Lab session

- Synthesis of tones

  Synthesize sine waves of various frequencies, amplitudes and phases at specified sampling rate; additive synthesis of musical tones

  - Filtering