# EECE 491: Discrete-time Signal Processing

Mohammad M. Mansour
*Dept. of Electrical and Compute Engineering*
*American University of Beirut*

## Lecture 1: Introduction

# Administrative

- **Instructor**
  - Dr. Mohammad M. Mansour, Professor of ECE
  - Office: 508 Bechtel
  - Extension: 3597
  - Email: mmansour@aub.edu.lb
- **Send emails with Subject: EECE491**
  - Will try to reply back during office hours
- **Course Webpage on Moodle**
- **Teaching Assistant (TA)**
  - Hussein Hammoud, hah49@mail.aub.edu
- **Office hours:**
  - Mondays: 9:00 – 11:00 AM
- **Lecture hours:**
  - T-R: 2:00 – 3:30 pm in room 208 Bechtel

# My Research Area

- **Software-defined modems for 5G wireless networks**
- **High-performance communication and digital signal processing systems**
- **Capacity-approaching channel coding and applications**
- **Energy-efficient designs by algorithm, architecture and circuit co-optimization**
- **Error-resilient architectures and circuits, and applications**
- **Computer architecture**

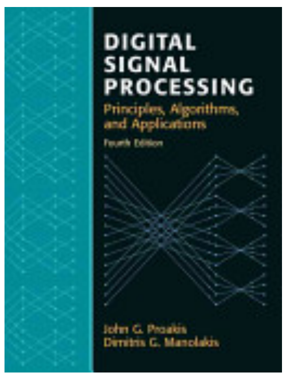
# Course Policy

- **Grading:**
  - 2 Midterms (35%)
  - Final Exam (35%)
  - Homework Assignments (10%)
  - Project (20%)

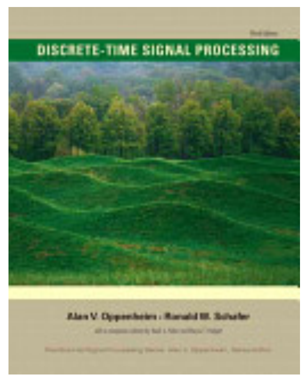
- **Projects: Individually done**
- **Lectures begin on the hour. Please come time.**
- **Exams are closed book and comprehensive.**

# Course Syllabus

- **Main Textbooks**
  - **"Digital Signal Processing", J. Proakis and D. Manolakis, Prentice Hall, 4th Ed.**
  - "Discrete Time Signal Processing", A. Oppenheim and R. Schafer, Prentice Hall, 3rd Ed.
  - "Digital Signal Processing", S. Mitra, McGraw Hill, 4th Ed.




P&M




O&S


Mitra

## Course Topics

- Signals, Systems, and Transforms (Review)
- LTI Discrete-Time Systems
- Digital Filter Structures
- FIR and IIR Digital Filter Design
- Filter Design Based on a Least Squares Approach
- DSP Algorithm Implementation
- Finite Word-length Effects
- Multirate Digital Signal Processing
- Multirate Filter Banks and Wavelets
- DSP Architectures
- Applications to Software Defined Radios and Communications Modems
- Introduction to Adaptive Filtering

# Topics and Applications

- **Digital signal processing algorithms/applications**
  - Signals, convolution, sampling (signals & systems)
  - Transfer functions & freq. resp. (signals & systems)
  - Filter design & implementation, signal-to-noise ratio
  - Quantization (embedded systems) and data conversion

- **Digital communication algorithms/applications**
  - Analog modulation/demodulation (signals & systems)
  - Digital modulation/demodulation, pulse shaping, pseudo noise
  - Signal quality: matched filtering, bit error probability

- **Digital signal processor (DSP) architectures**
  - Assembly language, interfacing, pipelining (embedded systems)
  - Harvard architecture, addressing modes, real-time programming

# Software Tools

- **Matlab**
  - DSP toolbox
- **Simulink**
- **DSP toolkits in lab**

# Signal Processing in General

- **Signal Processing**
  - Generation, transformation and extraction of information
  - Algorithms with associated architectures and implementations
  - Applications related to processing information

- **Examples:**
  - Convert one signal to another
    - Examples: Filter, generate control commands, etc.
  - Interpretation and information extraction
    - Examples: Speech recognition, machine learning

- **Real-time Signal Processing systems**
  - Guarantee delivery of data by a specific time

# *Digital* Signal Processing

- **Digital: Signal $x(t)$ is discrete both in amplitude and time**
  - Discrete samples (in time): $x[0]$, $x[1]$, $x[2]$, …
  - Discrete amplitude (quantization)
    - $x[n]$ quantized and represented as a binary number
- **Notation:**
  - Continuous-time signal: $x(t)$
  - Discrete-time signal: $x[n]$
  - Digital signal: Q(x[n])
- **Digital representation (on a computer)**
- **Discrete-time samples can be samples of a continuous-time signal:**
  - Samples taken $T$ seconds apart ($T$ = sampling period)

$$x[n] = x(t)\big|_{t = nT} = x(nT)$$

# Why Learn DSP?

- **“Swiss-Army-Knife” of modern EE**

- **Impacts all aspects of modern life**
  - Communications (wireless, internet, GPS, etc.)
  - Control and monitoring (cars, machines, etc.)
  - Multimedia (mp3, cameras, videos, restoration, etc.)
  - Health (medical devices, imaging, etc.)
  - Economy (stock market, prediction)
  - More ...

# What Can You Do with Digital Signal Processors?



***Consumer audio***

Smart meters


DSL modems

***Communications***

Wireless Wearable
Multichannel EEG

IP camera

Video conferencing

Microscopes

Mixing
board

Amp

***Pro-audio***

In-car entertainment

Tablets

MRI noise
cancelling
headphones

***Biomedical***

## What Can You Do with Digital Signal Processors?

**machine vision**

**Avionics & Defense**

**High Performance Computing**

**Video Encoding/ Decoding**

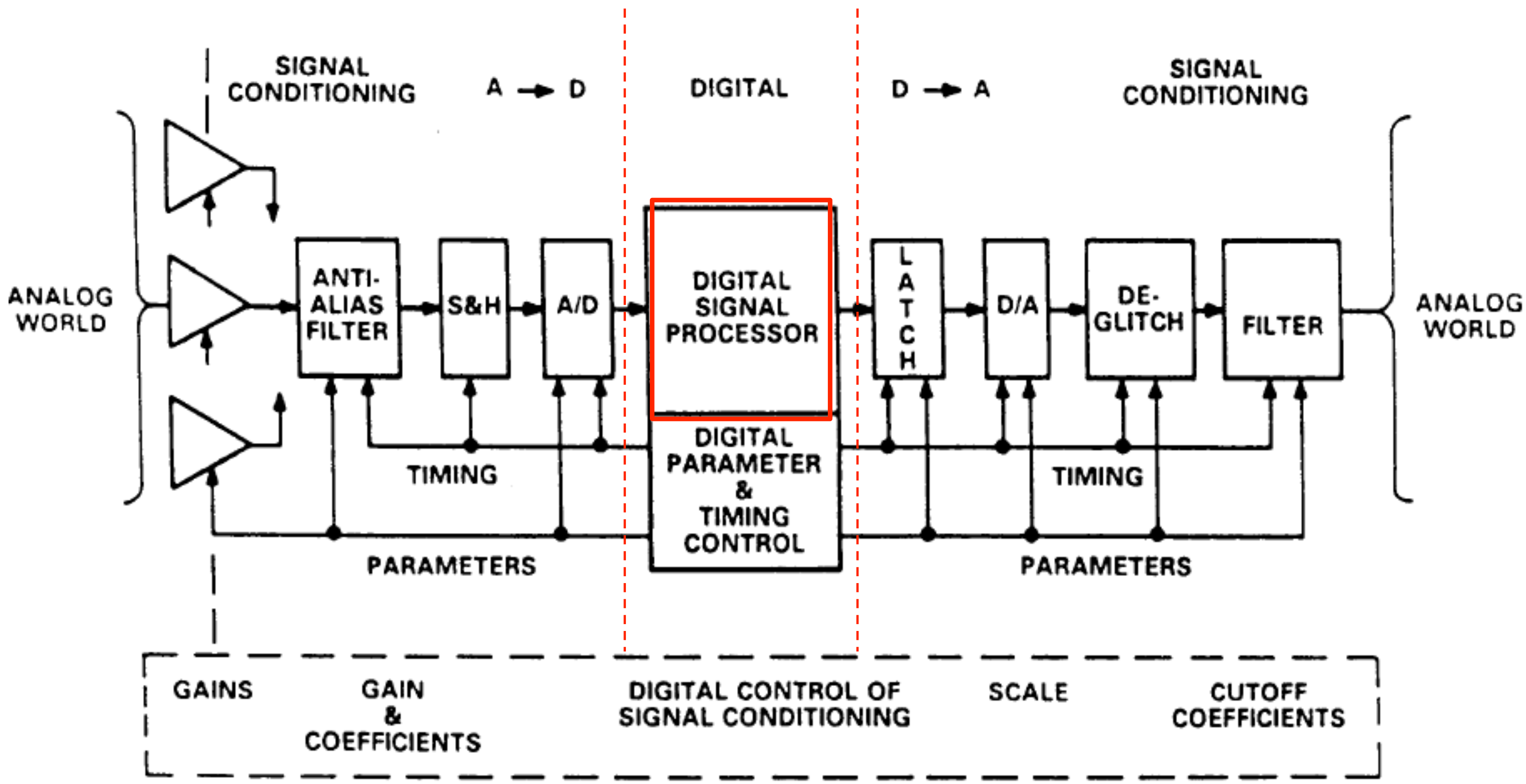
**Biometrics**

# Advantages of DSP

- **Flexibility**
- **System/implementation does not age**
- **"Easy" implementation**
  - Ex: Programmable implementation on a DSP
- **Reusable hardware**
- **Sophisticated processing**
- **Process on a computer**
- **(Today) Computation is cheaper and better**
  - Vector DSPs
  - Multi-core DSPs

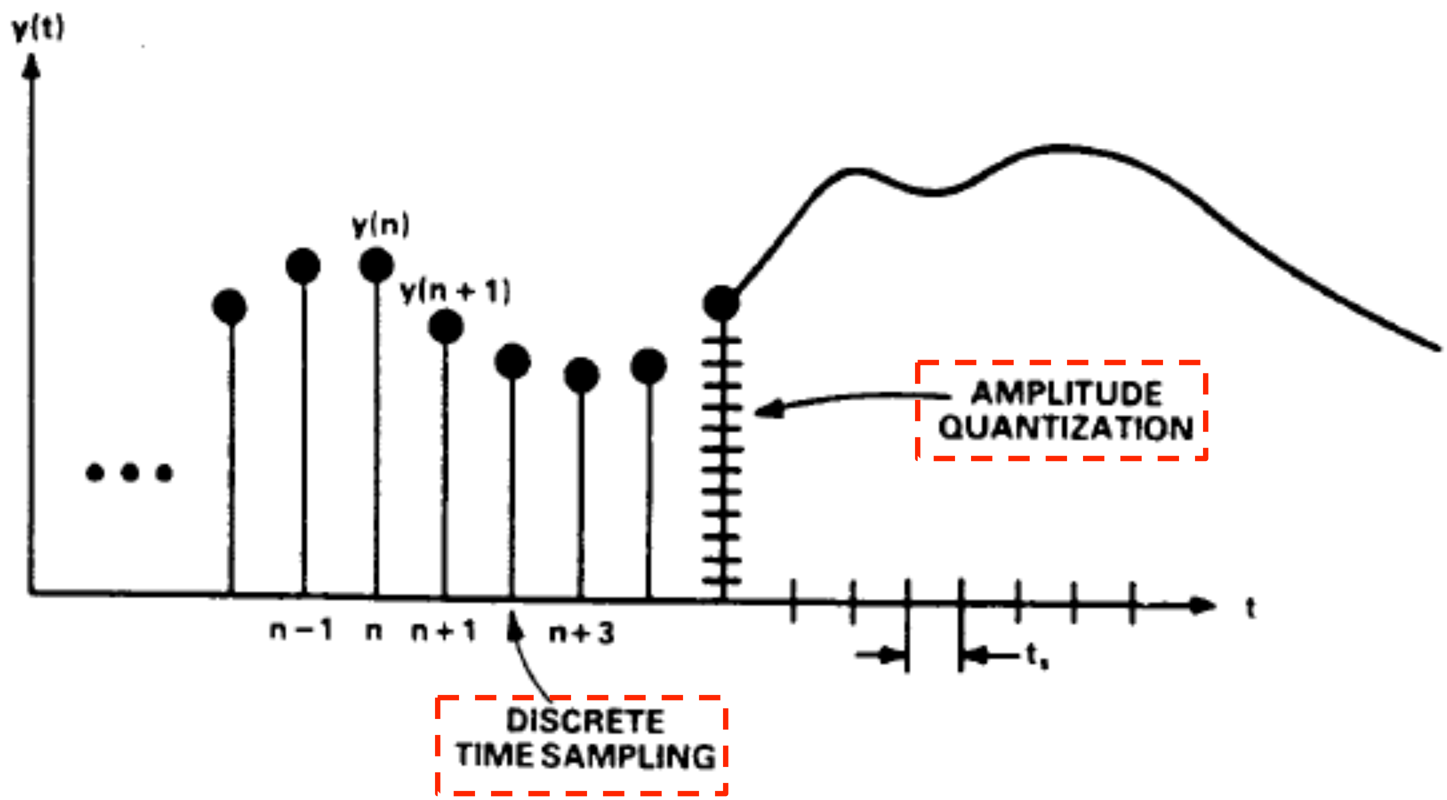
# DSP System Structure

# Key Components of a Sampled Data System

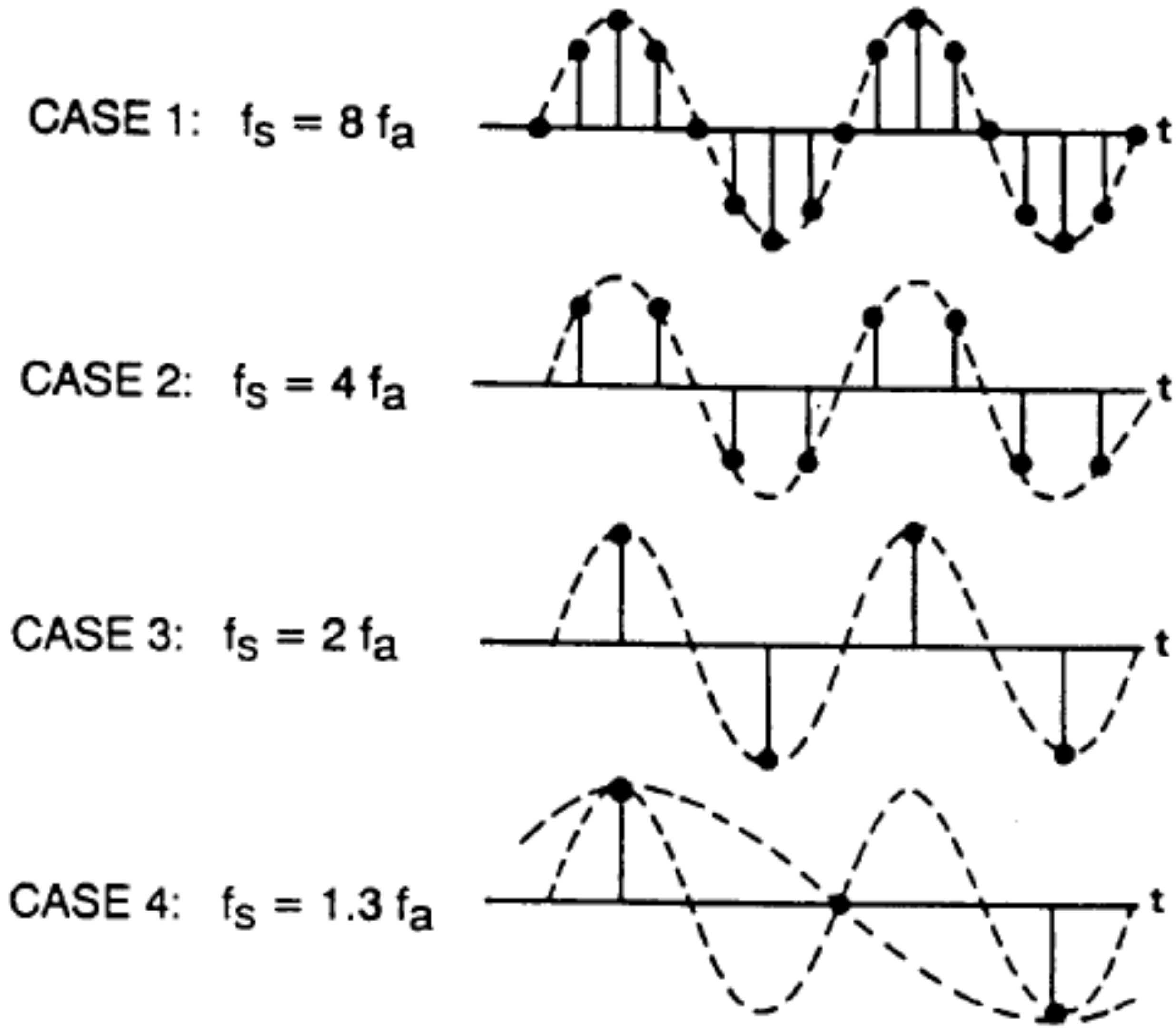



**[Source: *Analog Devices*]**

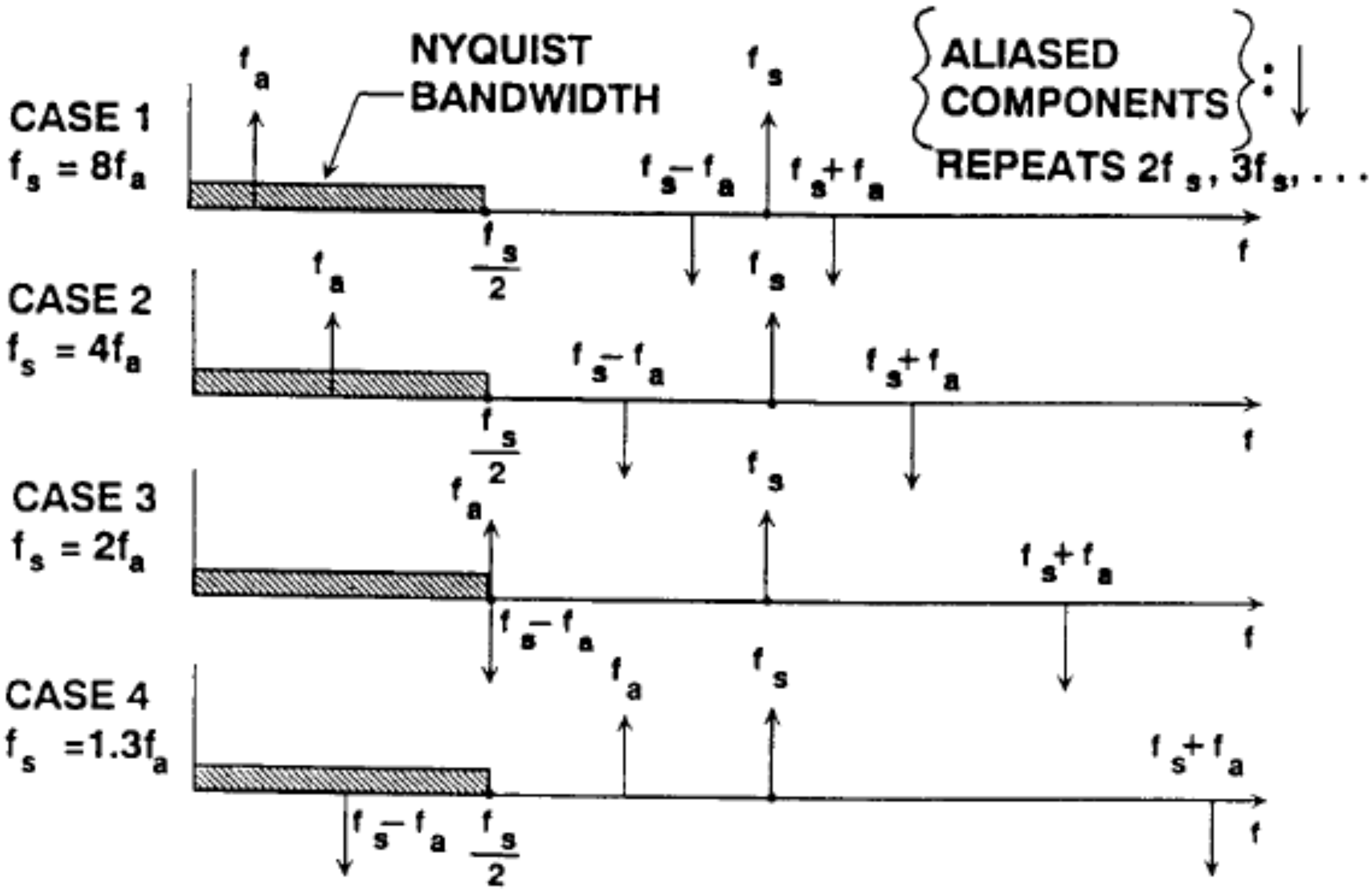
# Discrete Sampling of an Analog Signal



# Time-Domain Effects of Aliasing



# Frequency Domain Effects of Aliasing

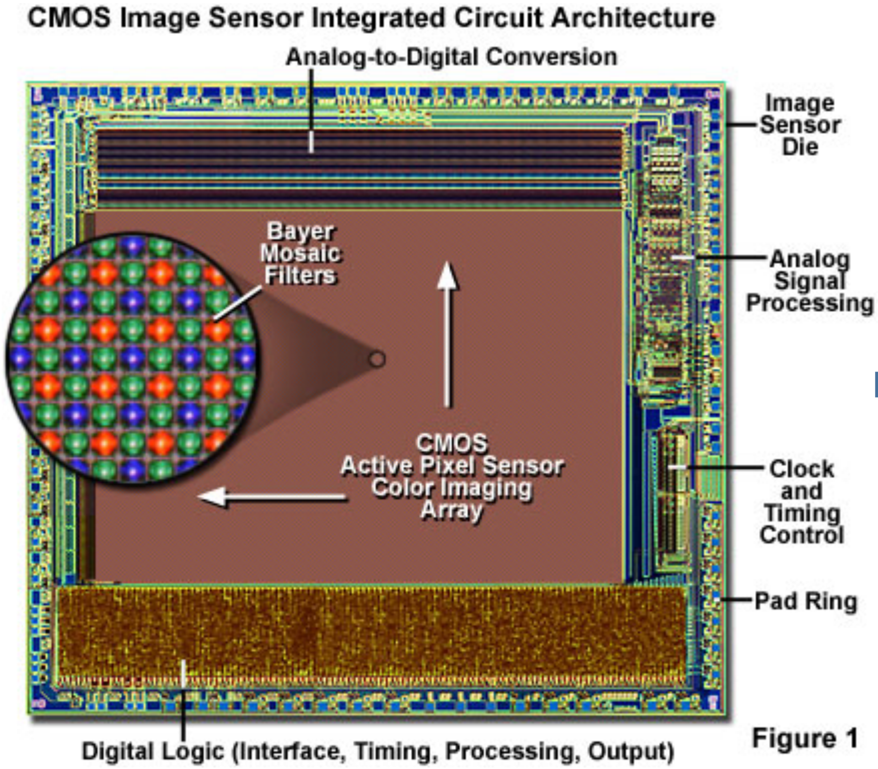


# Examples

# Example I: Audio Compression

- **Compress audio by 10x without perceptual loss of quality.**
- **Sophisticated processing based on models of human perception**
- **3MB files instead of 30MB**
  - Entire industry changed in less than 10 years!

# Example II: Digital Camera




Focus/exposure Control → Preprocessing → White balancing → Demosaic → Color transform → Post-processing → Compression

(color reconstruction)

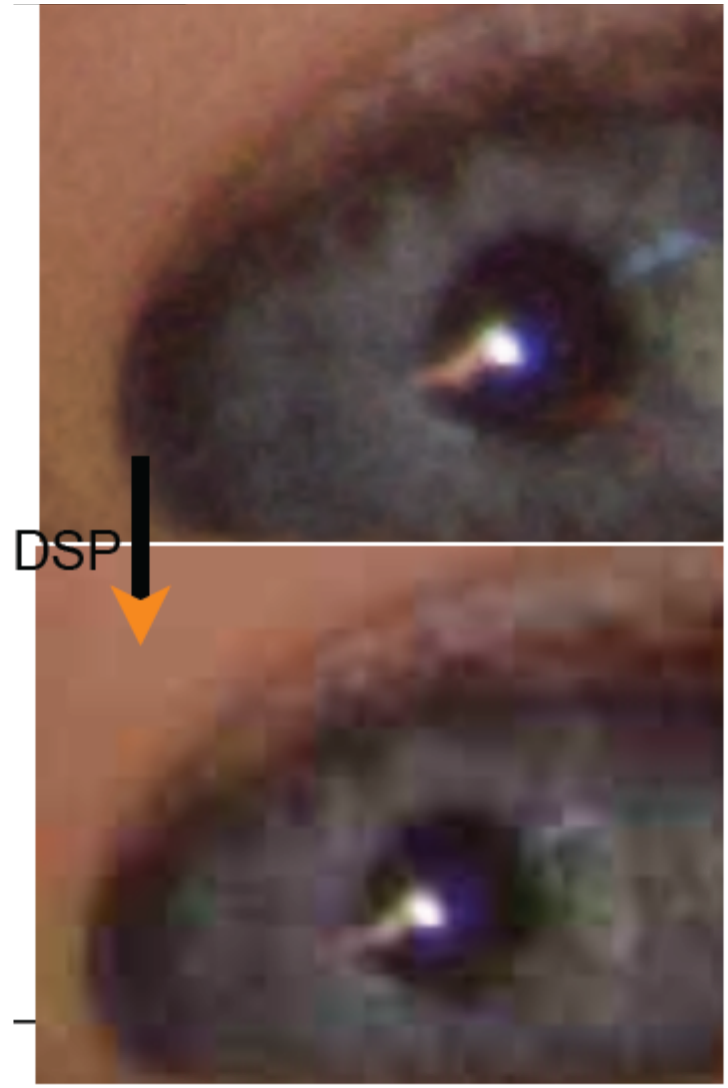
# Example II: Digital Camera (cont'd)

DSP

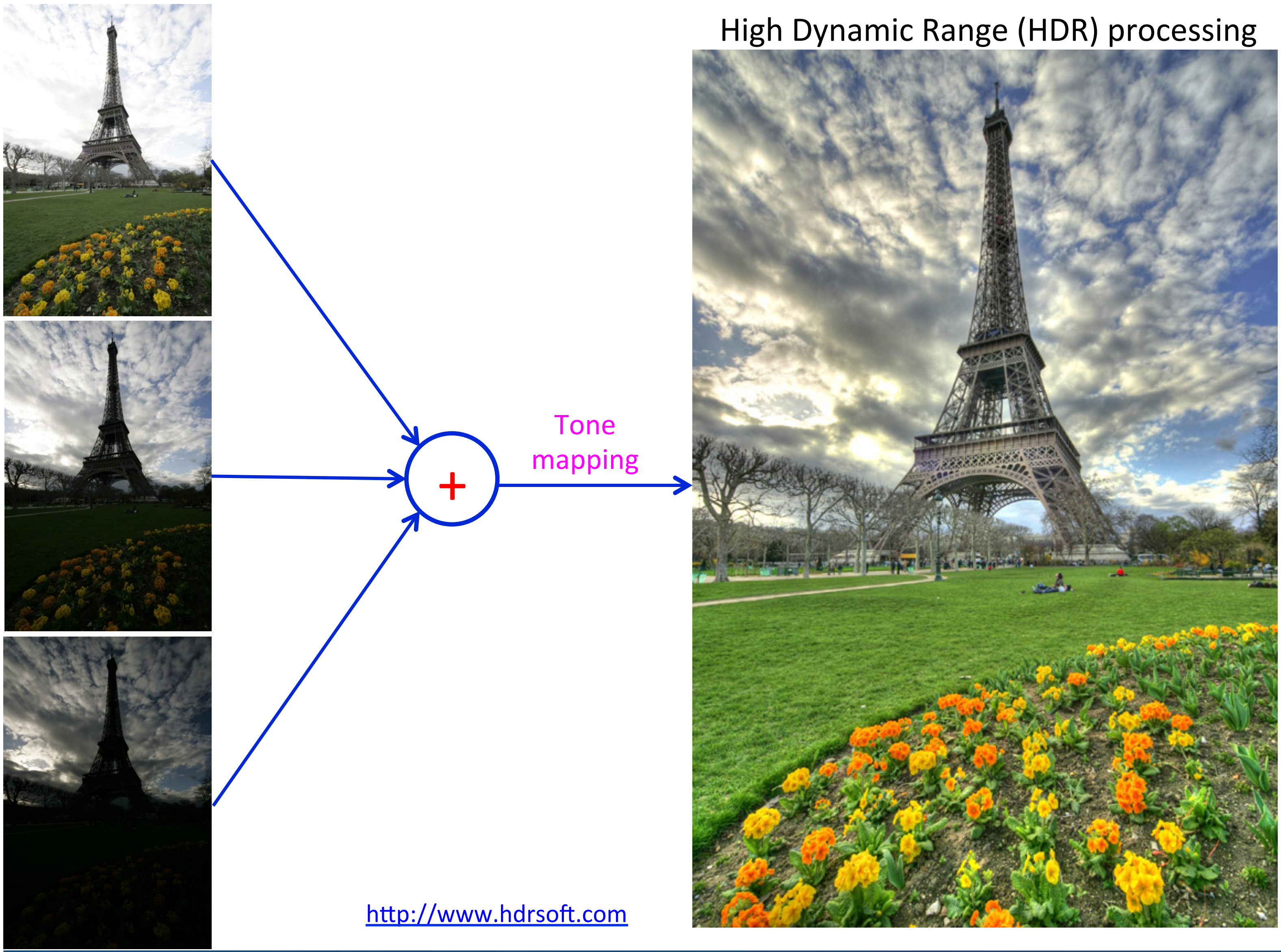
# Example II: Digital Camera (cont'd)

- **Compression of 40x without perceptual loss of quality**

- **Example of slight over-compression difference enables x60 compression!**



# Example III: Computational Photography




http://www.hdrsoft.com

# Example IV: Software Defined Radio (SDR)

- **Traditional radio:**
  - Hardware receiver/demodulators/filtering
  - Outputs analog signals or digital bits

- **Software Defined Radio:**
  - Uses RF font end for baseband signal
  - High speed ADC digitizes samples
  - All processing chain done in software

# Example IV: Software Defined Radio (cont'd)

- **Advantages:**
  - Flexibility
  - Upgradability
  - Sophisticated processing
  - Ideal Processing chain - not approximate like in analog hardware

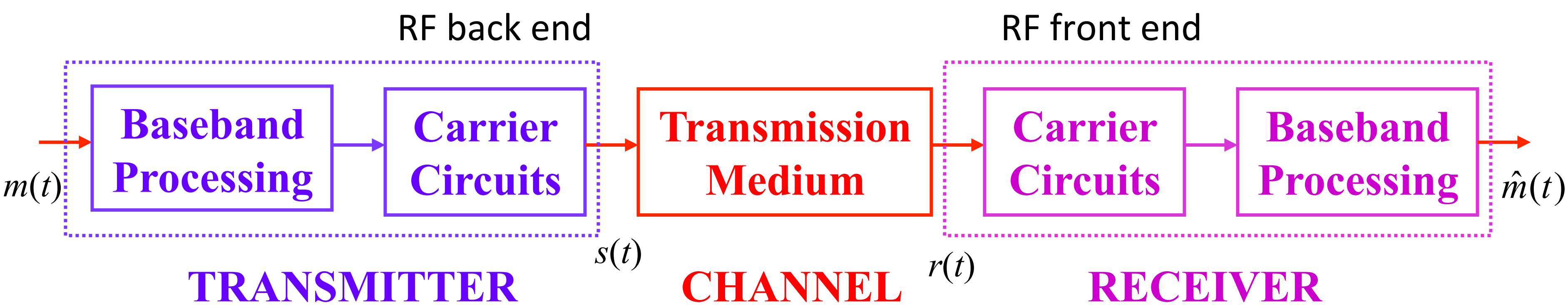
- **Already used in consumer electronics**
  - Cellphone baseband processors
  - Wifi, GPS, etc.

# Communication System Structure

# COMM System Structure

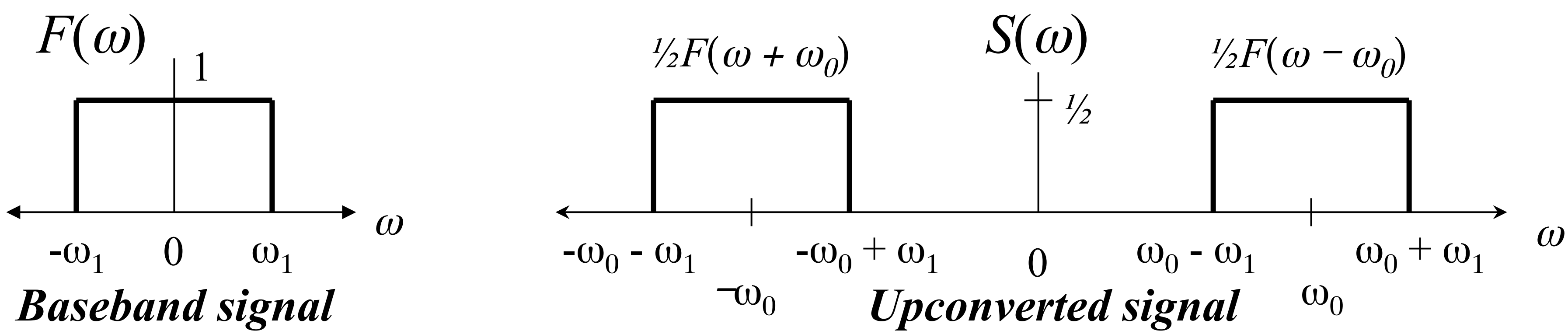
- **Information sources**
  - Voice, music, images, video, and data (message signal $m(t)$)
  - Have power concentrated near DC (called baseband signals)

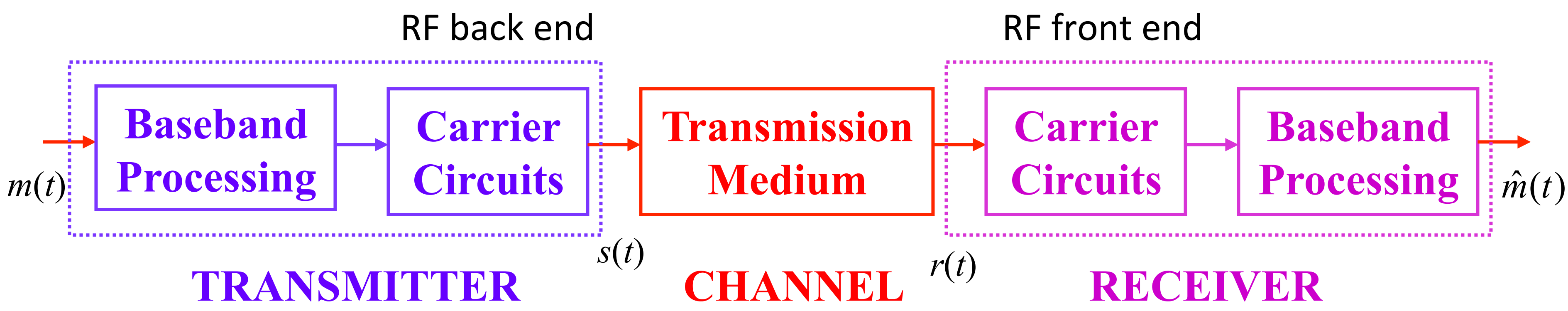
- **Baseband processing in transmitter**
  - Lowpass filter message signal (e.g. AM/FM radio)
  - **Digital**: Add redundancy to message bit stream to aid receiver in detecting and possibly correcting bit errors



# COMM System Structure

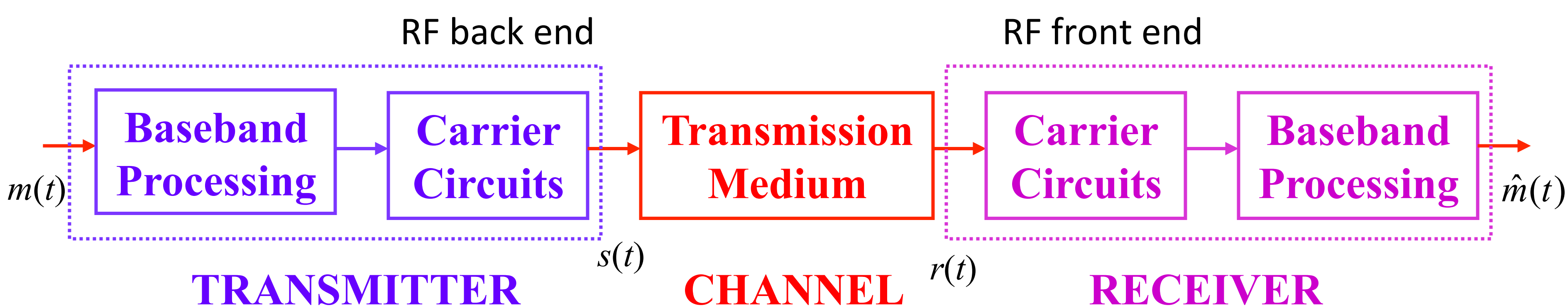
- **Carrier circuits in transmitter (RF back end)**
  - Up-convert baseband signal into transmission band
  - Apply bandpass filtering to enforce transmission in band



# COMM System Structure

- **Channel – wired or wireless**
  - Propagating signals spread and attenuate over distance
  - Boosting improves signal strength and reduces noise
- **Receiver**
  - Carrier circuits down-convert bandpass signal to baseband
  - Baseband processing extracts/enhances message signal



## Course Objective

- **Develop skills for**
  - Analyzing and synthesizing algorithms and systems that process discrete-time signals
  - Implementing these algorithms and systems