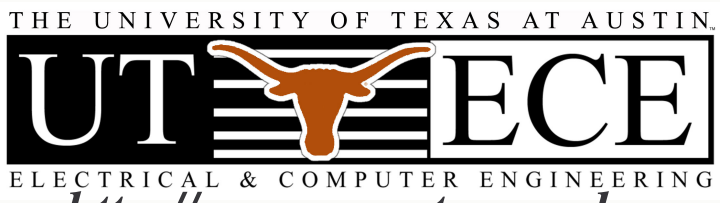
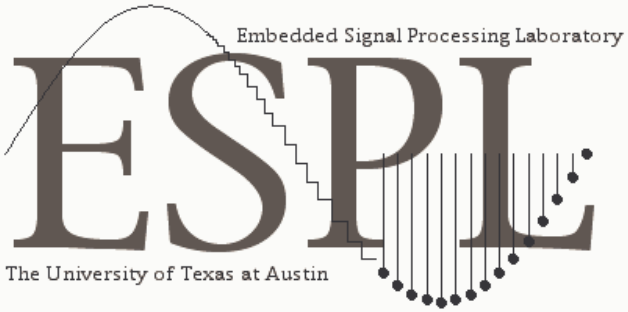
Embedded Software Systems

Prof. Brian L. Evans



http://www.ece.utexas.edu



http://signal.ece.utexas.edu

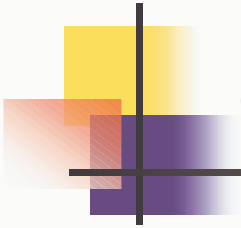
Wireless Networking & Communications Group

http://www.wncg.org

CENTER FOR PERCEPTUAL SYSTEMS
The University of Texas at Austin

http://www.cps.utexas.edu

January 21, 2004



Outline

- Introduction
- Programmable Digital Signal Processors
- Electronic Design Automation
  - Methods and Tools
  - Dataflow Models
  - Process Networks
- Communication Systems
  - General Structure
  - ADSL Transceiver Block Diagram

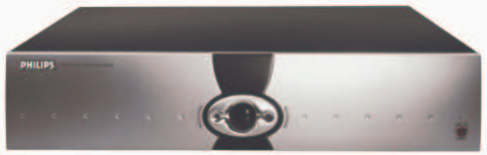
Overview

- What are embedded systems?
  - Computers masquerading as non-computers

Casio Camera Watch

Nokia 7110 Browser Phone

Sony Playstation 2



Philips TiVo Recorder

Philips DVD player

Slide courtesy of Prof. Stephen A. Edwards of Columbia University

Embedded System Challenges

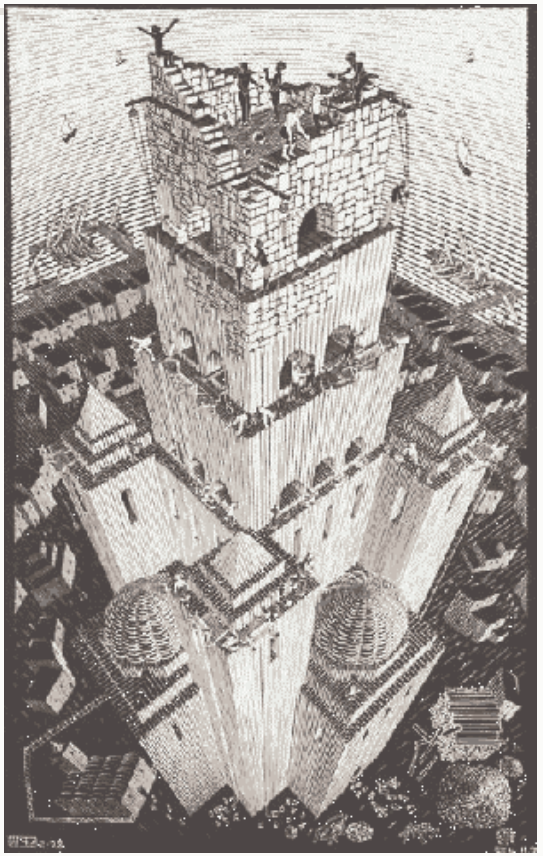
- Differs from general-purpose computing
  - Real-time constraints
  - Power constraints
  - Exotic hardware
  - Concurrency
  - Control systems
  - Signal processing
  - User interface
  - Laws of physics

SR-71

Slide courtesy of Prof. Stephen A. Edwards of Columbia University

The Role of Languages

- Language shapes how you solve a problem
- Java, C & C++ designed for general-purpose systems programming
- Do not address timing, concurrency
- Domain-specific languages are much more concise
- Problem must fit the language



M. C. Escher, Tower of Babel

Slide courtesy of Prof. Stephen A. Edwards of Columbia University

Course Topics

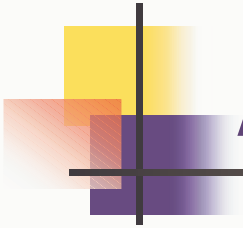
- Programming languages
  - Procedural programming: Assembly and C
  - Object-oriented programming: C++ and Java
- Real-time operating systems
  - Concurrency
  - Meeting deadlines
- Modeling systems
  - Dataflow languages
  - Synchronous/reactive languages
- Modeling environments
  - Discrete-event models

Pre-requisites

Algorithms

Object-oriented software design

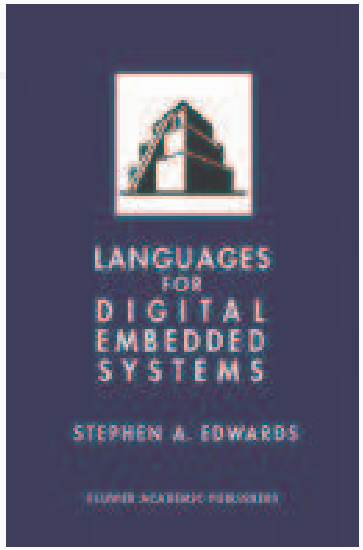
Embedded software implementation



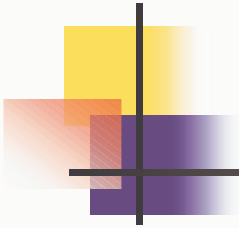
A Few Related Courses

- EE380L-5 Engineering Programming Languages (Fall)
- EE382C-8 Methodologies of Hardware/Software Codesign (Spring, odd years)
- EE382M High-Level Synthesis (Spring, even years)
- EE382N Parallel Computer Architecture (Fall)
- EE382N-11 Distributed Systems (every year)
- EE382N-14 High-Speed Computer Arithmetic (Fall)
- CS388S Formal Semantics and Verification
- CS392C Methods/Tech. for Parallel Programming
- CS395T Real-Time Systems

Course Textbooks



- Stephen A. Edwards, *Languages for Digital Embedded Systems*, Kluwer, 2000 (Required)
  - Survey of field
  - Balanced software/hardware coverage
- Shuvra S. Bhattacharyya, Praveen K. Murthy, and Edward A. Lee, *Software Synthesis from Dataflow Graphs*, Kluwer, 1996 (Optional)
  - Synchronous Dataflow (SDF) model of computation
  - Scheduling SDF graphs onto single processors
  - Was the textbook for the course before 2002



Course Goals

- Breadth
  - Knowledge of many different languages
  - Languages embody design methodologies
  - Broader knowledge, bigger "bag of tricks"
- Depth
  - Big design project
  - Gives you in-depth experience with one of the languages

Grading

- Calculation of numeric grades
  - 20% midterm #1
  - 20% midterm #2 (not cumulative)
  - 10% homework (four assignments)
  - 50% project (progress towards publishable research)
- Project
  - Project idea – due in two weeks
  - Project white paper – due in four weeks
  - Literature survey talk – week before Spring Break
  - Literature survey report – week after Spring Break
  - Final presentation – final week of lecture
  - Final project report – due after "dead" days

Past average GPA is 3.53

www.UTLife.com

No final exam

20% of reports are published

Examples of Good Project Reports

- Computer Architecture
  - David Armstrong, 2002, "Architectural Considerations for Network Processor Design"
  - Deepu Talla, 1999, "Evaluating Programmable VLIW and SIMD Architectures for DSP and Multimedia Applications"
- Design Automation Tools
  - Gregory Allen and David Schanbacher, 1997, “Beamforming with Process Networks/Pthreads”
  - "Hugo Andrade and Scott Kovner, 1998, “Software Synthesis from Dataflow Models for Embedded Software Design in the G Programming Language and the LabVIEW Development Environment”

Handout K

Handout T

Examples of Good Project Reports

- Application-Specific

  Matthew Felder and Jimmy Mason1997, "Efficient Dual-Tone Multiple-Frequency Detection Using the Non-Uniform Discrete Fourier Transform"

  Thomas Holme and Karen Watkins, 1998, "Optimal Architectures for Massively Parallel Implementation of Hard Real-time Beamformers"

  Koichi Sato, 2002, "Designing Intelligent Surveillance Camera System"

- All literature survey and final reports and presentations are available on class Web site

Academic Integrity

- Homework assignments
  - Discuss homework questions with others
  - Be sure to submit your own *independent* solution
  - Turning in two identical (or nearly identical) homework sets is considered *academic dishonesty*
- Project reports and presentations
  - Should only contain work of those named on report
  - If any other work is included, then reference source
  - Copying information from another source without giving proper reference and quotation is *plagiarism*
- Why does academic integrity matter? *Enron!*

Instructional Staff

- Prof. Brian L. Evans

  Research: embedded real-time signal and image processing systems, electronic design methods and tools

  Office hours: MW 2:00 – 3:30 PM, ENS 433B, 232-1457

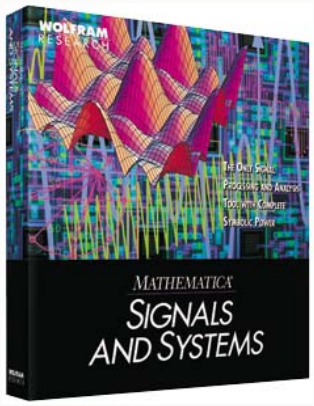
- Mr. Ming Ding (Grader)

  Research: communication system design

  Will hold office hours during the two days before a homework assignment is due

On My Way to Austin…

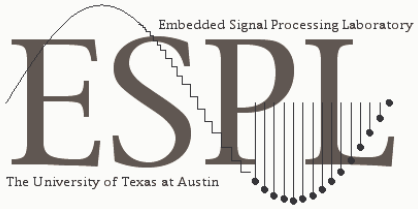
1987-1993

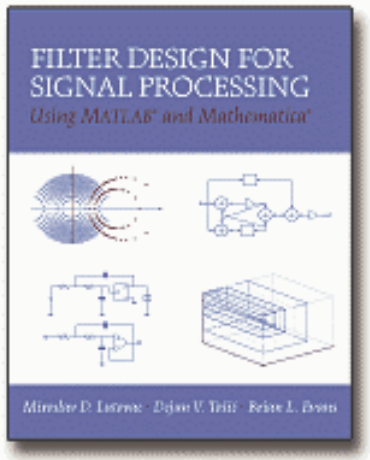


1993-1996

- Signals and Systems Pack
  - Symbolic analysis of signals and systems in Mathematica
  - By product of my PhD work
  - On market since 1995
- Ptolemy Classic
  - Mixes models of computation
    - Untimed dataflow
    - Process network
    - Discrete-event
  - Untimed dataflow synthesis
  - Source code powers Agilent Advanced Design System

Embedded Signal Processing Lab





- Develop and Disseminate
  - Theoretical bounds on signal/image quality
  - Optimal and low-complexity algorithms using bounds
  - Algorithm suites and fixed-point, real-time prototypes
- Analog/Digital IIR Filter Design for Implementation
  - Butterworth and Chebyshev filters are special cases of Elliptic filters
  - Minimum order does not always give most efficient implementation
  - Control quality factors

Students & Alumni

Embedded Signal Processing Laboratory
ESPL
The University of Texas at Austin

ADSL/VDSL Transceiver Design

Ph.D. students: **Dogu Arifler**
Ming Ding

Ph.D. graduates: **Güner Arslan** (Cicada)
Biao Lu (Schlumberger)
Milos Milosevic (Schlumberger)

Wireless Communications

Ph.D. students: **Kyungtae Han**
Zukang Shen

MS students: **Ian Wong** (NI Summer Intern)

Ph.D. graduate: **Murat Torlak** (UT Dallas)

MS graduates: **Srikanth K. Gummadi** (TI)
Amey A. Deosthali (TI)

Real-Time Imaging

Ph.D. students: **Gregory E. Allen** (UT Applied Research Labs)
Serene Banerjee

MS students: **Vishal Monga**

Ph.D. graduates: **Thomas D. Kite** (Audio Precision)
Niranjan Damera-Venkata (HP Labs)

MS graduates: **Young Cho** (UCLA)

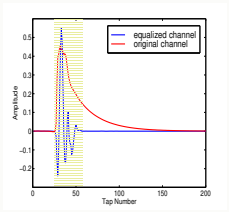


Image Analysis

Ph.D. graduates: **Dong Wei** (SBC Research)
K. Clint Slatton (University of Florida)
Wade C. Schwartzkopf (Integrity Applications)

Wireless Networking and Comm Group: http://www.wncg.org

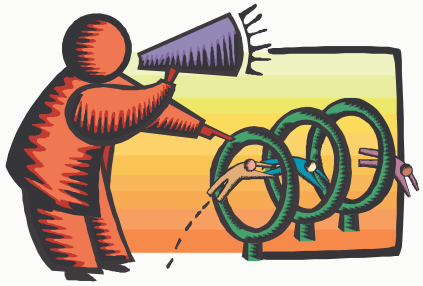
Center for Perceptual Systems http://www.cps.utexas.edu

Digital Signal Processors (DSPs)

- For real time (*guaranteed delivery*)
- Fixed-point DSPs for high-volume products
  - Battery-powered: cell phones, dial-up modems, portable MP3 players, digital still cameras, and digital video (e.g. TI C5000)
  - Wall-powered: ADSL modems, VDSL modems, cell phone basestations, modem banks, laser printers, video conferencing systems (e.g. TI 6200, C6400)
- Floating-point DSPs for low-volume products and feasibility analysis on fixed-point DSPs
- TI 45%, Agere 25%, Mot 10%, 8% Analog

Digital Signal Processor Architecture

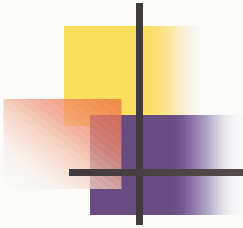
- Harvard architecture: program/data memory separated and can be accessed on same cycle
- Word size: 16, 20, 24, or 32 bits
- Programmer must manage memory
  - 32-128 kwords data/program on chip
  - On-chip data cache rare (TI C6000)
  - No support for virtual memory



- Predictable input/output: deterministic interrupt service routine latency (e.g. 11 cycles on TI C6000)

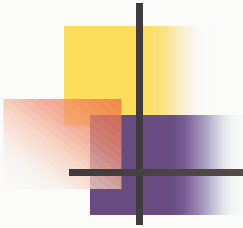
Digital Signal Processor Architecture

- Deterministic, no-overhead looping
- Single instruction cycle multiply unit(s)
- No-overhead addressing modes in hardware
  - Modulo addressing for circular buffers, e.g. filters
  - Bit-reversed addressing, e.g. fast Fourier transforms (not available on TI C6000)
- Native number formats
  - Integer: binary point on far right of bit pattern
  - Fractional: binary point just right of sign bit
  - Floating-point: could emulate on fixed-point DSPs



Drawbacks to Programming DSPs

- General drawbacks
  - Limited on-chip memory
  - Poor C compiler performance
- Fixed-point issues
  - Non-standard C extensions for fractional data
  - Converting floating-point programs to fixed-point
  - Manual tracking of binary point prone to error
- Conventional DSPs
  - No byte addressing (needed for image/video)
  - Limited addressable memory on fixed-point DSPs



Electronic Design Automation

- Specification, simulation, and synthesis

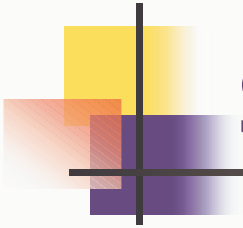
Programming languages	Concurrency
Dataflow models	Process network
Scheduling	Software synthesis
Discrete-event models	Cosimulation

- Evaluate/build embedded system designs in
  - Ptolemy Classic from UC Berkeley
  - Ptolemy II from UC Berkeley
  - Advanced Design System from Agilent
  - LabVIEW from National Instruments

Dataflow Models

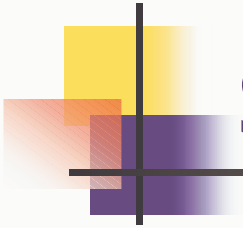
Examples in modern design automation tools

Electronic Design Automation Tool	*Dataflow Models*	*Example Application*
Agilent Advanced Design System	Synchronous Dataflow, Timed Synchronous Dataflow	Mixed analog, digital, and RF communication systems (data transmission subsystem)
Co-Centric System Design Studio	Cyclostatic Dataflow	Periodic digital systems, e.g. data converters, MP3 decoder, digital baseband communications
LabVIEW	Homogeneous Dynamic Dataflow, Process Network	Mixed analog and digital data acquistion and processing systems
UC Berkeley Ptolemy Classic and Ptolemy II	Synchronous Dataflow, Boolean Dataflow, Dynamic Dataflow	Periodic and aperiodic digital systems



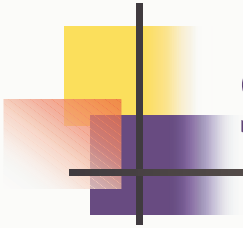
Synchronous Dataflow [Lee 1986]

- Arcs: one-way first-in first-out queues
- A block is enabled for execution when enough tokens are available on all inputs
  - Source blocks are always enabled
- When block executes, it always produces and consumes the same fixed amount of tokens
  - Consumed data is dequeued from arc
- Flow of data through graph may not depend on values of data
- Delay is a property of an arc
  - Delay of *n* samples means that *n* tokens are initially in the queue of that arc



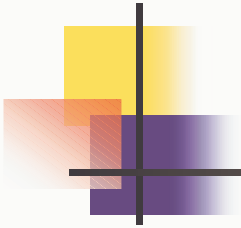
Synchronous Dataflow

- Systems are *determinate*
  - History of tokens produced on communication channels do not depend on the execution order
  - May be executed sequentially or in parallel with the same outcome
- Scheduling
  - Load balancing to make sure that all tokens produced can be consumed: linear complexity
  - Find a periodic schedule
    - List scheduling: worst-case is exponential complexity
    - Heuristics to minimize buffer size: cubic complexity



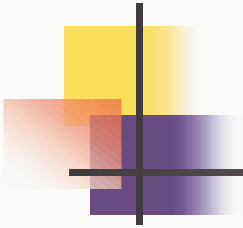
Synchronous Dataflow Modeling

- Signal Processing
  - Finite impulse response filters
  - Infinite impulse response filters
  - Fast Fourier transform
  - Multirate systems and filter banks
- Communication Systems
  - Sinusoidal modulation and demodulation
  - Pulse shapers
  - Transmission subsystem
- Inappropriate for data-dependent graphs, e.g. baud rate negotiation at modem startup



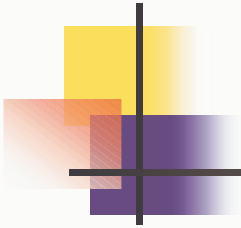
Process Network [Kahn 1974]

- A set of concurrent processes that communicate through network of one-way infinite first-in first-out (FIFO) queues
- Reads from queues are blocking
  - If the queue is empty, the process will suspend until there is enough data in the queue.
  - When a process blocks, the scheduler will not run the process until enough data becomes available.
- Writes to the queues are non-blocking



Process Network

- A process is either *enabled* or *blocked* waiting for data on only one of its input channels
- Systems are *determinate*
  - History of tokens produced on communication channels do not depend on the execution order
  - May be executed sequentially or in parallel with the same outcome
- Supports recurrence and recursion
- Formal mathematical representation: processes are functions that map streams into streams



Process Network

- Turing complete: questions of termination and bounded buffering are undecidable
- Undecidable (in finite time) if process network
  - Terminates
  - Requires bounded memory
- Signal processing: run for infinite time
- Scheduler can find a bounded memory solution using infinite time [Parks 1995]
  - Ptolemy Process Network domain
  - UT Austin Computational Process Network framework in C++

    http://www.ece.utexas.edu/~allen/PNSourceCode/

Communication Systems

- Information sources
  - Message signal $m(t)$ information source to be sent
  - Possible information sources include voice, music, images, video, and data
- Basic structure of an analog communication system is shown below

Transmitter

- Signal processing
  - Lowpass filtering
  - In digital communications, redundancy added to message bit stream for error detection in receiver
- Carrier circuits
  - Multiplying input by sinusoid at carrier frequency, e.g. FM station such as 94.7 MHz

Channel

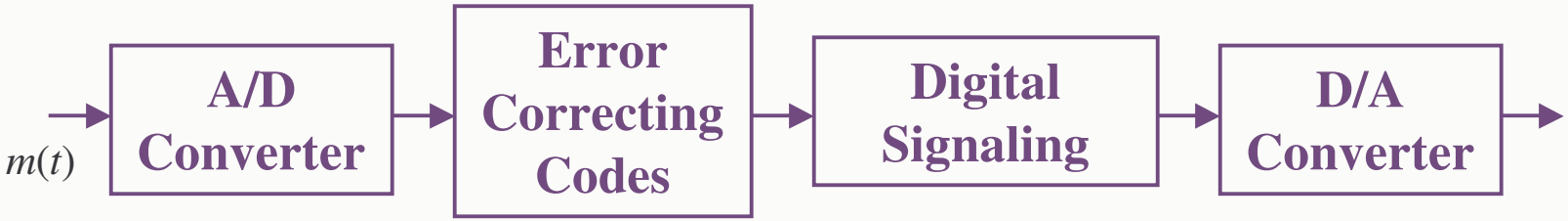
- Transmission medium
  - Wireline (twisted pair, coaxial, fiber optics)
  - Wireless (indoor/air, outdoor/air, space)
- Propagating signals experience a gradual degradation over distance
- Boosting improves signal and reduces noise, e.g. repeaters

Receiver and Information Sinks

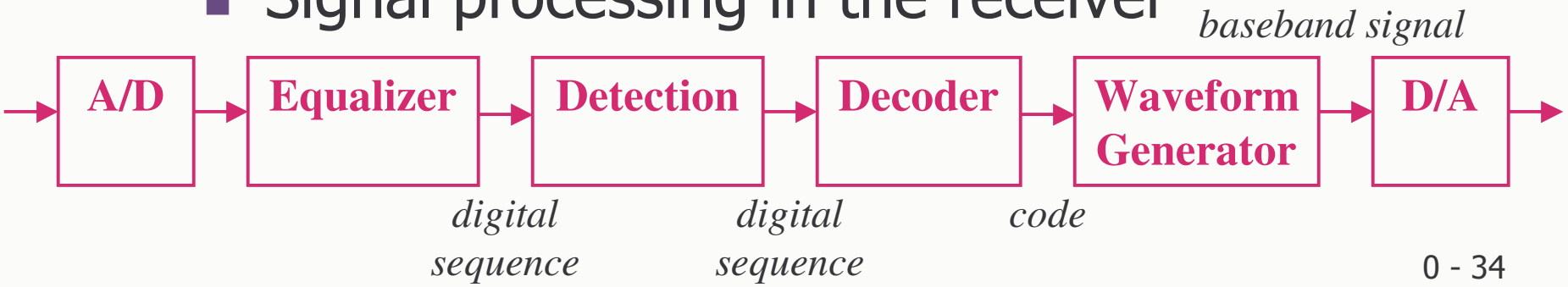
- Receiver
  - Carrier circuits undo effects of carrier circuits in transmitter, e.g. demodulate from a bandpass signal to a baseband signal
  - Signal processing subsystem extracts and enhances the baseband signal
- Information sinks
  - Output devices, e.g. computer screens & speakers

Hybrid Communication Systems

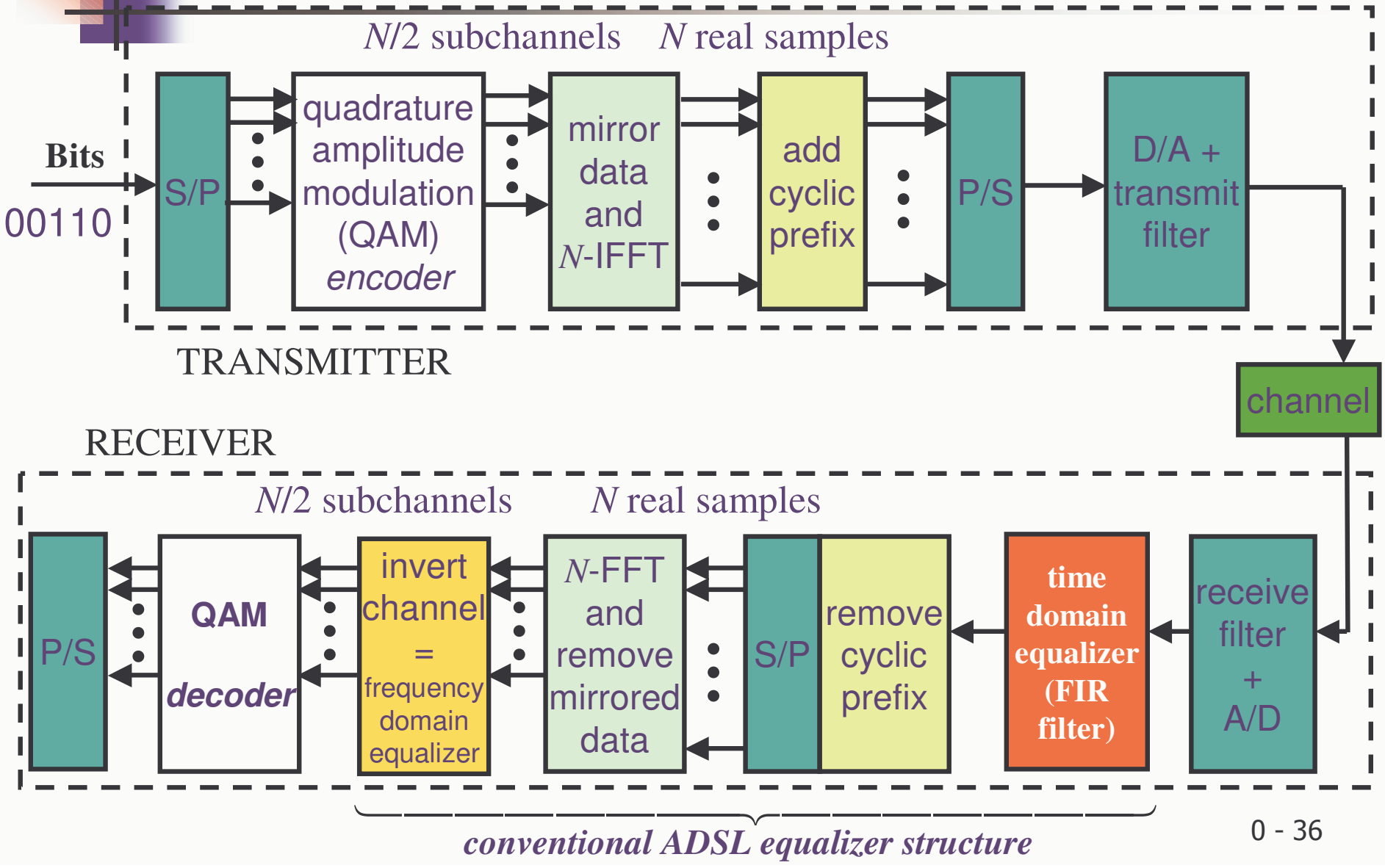
- Mixed analog and digital signal processing in transmitter and receiver
  - Message signal digital broadcast over analog channel (e.g. compressed speech in cell phones)
- Signal processing in the transmitter



- Signal processing in the receiver



ADSL Transceiver

- Asymmetric Digital Subscriber Line modem
  - Line driver (single chip)
  - Transceiver: analog front end + digital baseband
- Sampling rate: 2.208 MHz (real time)
- Bit error rate: 10^{-7} (Reed-Solomon codes)
- Symbol rate: 4,000 symbols/s
- Frame is symbol plus redundant information
- Single frame transmission (low delay)

ADSL Transceiver: Data Transmission