TSTE17 System Design, CDIO

- Lecture 2
  - System Design
  - Tools
  - Intro Digital Communication

Course news

- LIPS project model described in Lecture 5
  - Thursday 10/9 10.15-12.00

Outline

- System Design
- Introduction to the design flow
- Basic approach
- What aspects are important
- What tools do we use
- Channel model

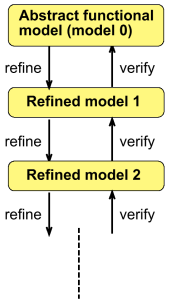
Design Flow

- System modeling incl. environment simulation
  - SDL (System Description Language)
  - C/C++
  - Java
- Algorithm (DSP) modeling
  - Matlab
  - Simulink
  - C/C++

Design Flow, cont.

- Software Implementation
  - SDL
  - C/C++
  - Java
  - Compile/debug environment
- Hardware Implementation
  - HDL + synthesis
  - C/C++ + synthesis

The Modelling Hierarchy



System Partitioning

- Map function blocks to physical units
  - Dedicated HW blocks
  - CPU SW
  - Combinations HW/SW
    - -> Performance analysis and modeling important
    - -> System simulation!
- Define interfaces between blocks
  - Bit widths
  - Latency
  - Waveforms

Example – Design of an I/O interface

- Initial model 0

  data = recieve(inport);

  transmit(outport,value);

  -> No timing or implementation information

  -> Fast => extensive system simulations possible
- Model 1

  while status /= OK { (data,status) = receive(inport) ; wait 1 clock cycle }

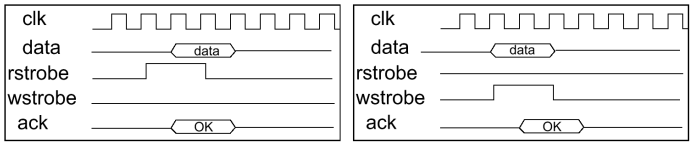
  while status /= OK { status = transmit(outport,value); wait 1 clock cycle }

  -> Some timing information

  -> No implementation information

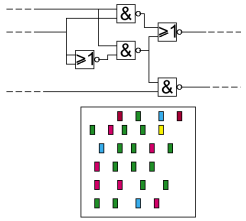
Example cont.

- Model N
  - Detailed clock accurate timing information
  - Too slow to be used in extensive system simulations



Example cont. - Model N+1

- Gate level description with timing from layout



  - Slow but accurate
  - Only simple functional verification/simulation possible

The Design Task

- Conceptual idea => real hardware
  - 50 MSamples/s input
  - Transmit over radio
  - Similar to 802.11a, DRM, or 802.16 (WIMAX)
  - Recieve and decode

(A radio channel simulator may be used)

Roles

- Customer
  - Decide on meetings
  - Need to iterate requirement specifications
  - Kent Palmkvist (kent.palmkvist@liu.se)
- Supervisor
  - Kent Palmkvist (kent.palmkvist@liu.se)
- Meetings arranged by signing up on lists outside office one day ahead or more

Group Roles

- Defined in the LIPS project model
  - Project manager
  - Documentation
  - Design
  - Testing
  - ....
- All members of the groups should have a role

Requirement specification

- Multiple versions
  - Each model level should produce a new requirement specification
    - Use simulation of the current model to find out suitable requirement for the next level
  - Use the old and add additional requirements

Design flow cont.

- First model starts with high-level blocks (behavioral)
  - Check functionality
  - Create a good testbench
  - Find limits of noise on channel (#bits in A/D & D/A)
- Second model adds details on algorithms (functional)
  - Select algorithms
  - Check wordlength (between blocks)
  - Find limits on wordlength (try to reduce)

Design flow cont.

- Third model adds bit true effects, wordlengths etc. (structural)
  - Work with fixed point numbers
  - Add timing
  - Include overflow and truncation
- Fourth model are synthesizable (Altera blocks)
  - Detailed timing
  - Generate hardware

Tool details

- Documentation available online (on the machines)
- Windows-machines do not use the same version of the tools
  - If you use windows versions, check documentation first

Simulink

- Simulates dynamic systems
- Built on top of Matlab
- All Matlab functions accessable
- Linear and non-linear systems
- Continous time and/or discrete time

Simulink Schematics

- Hierarchical schematics
- Combine blocks from libraries
- Create your own libraries
- Mask hides sublevels
- Parameter entry through masks

Simulink Programming

- S-functions describe blocks
  - Interface
  - Number of functions (start, step etc.)
  - State
- Programming languages supported
  - M-files (matlab script files)
  - C-MEX, C++, Ada, Fortran...
- Simulation model differs from VHDL

Simulink Typical blocks

- Blocks divided into libraries
  - Simulink
  - Communications Systems Toolsbox
  - DSP Systems Toolbox
  - Simulink Extras
  - Altera Blockset

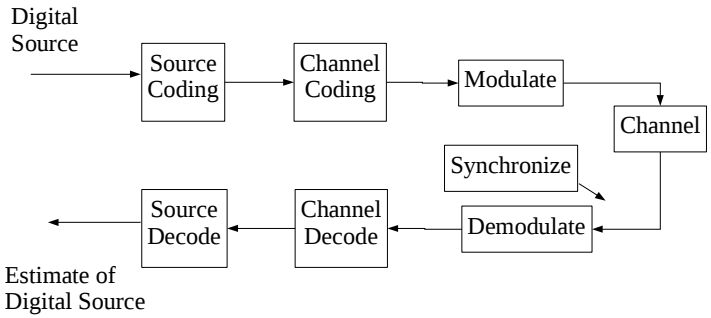
Simulink Extensions

- Large set of toolboxes
  - Finance ... Math ... Control
- Real Time Workshop (RTW)
  - Schematic -> C-programs
  - Supports DSP and general CPU

Telecom basics

- Assume knowledge about
  - Random variables
  - Discrete-Time signals
  - Discrete-Time systems
  - Linear Time-Invariant (LTI) systems
- If not, please look at the first chapter of the book (or literature in the corresponding courses)

Components of a digital communication system



Building blocks

- Source coding
  - Reduce redundancy in the source data
  - Example: English language
  - No gain if input data is statistically independent (memoryless)

Building blocks, cont.

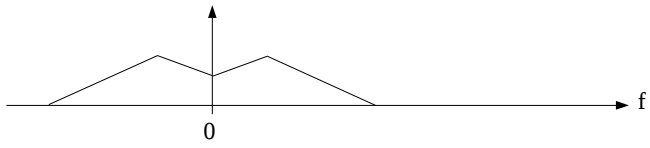
- Channel coding
  - Will be covered in a later lecture (protects data from single bit errors)
- Modulation
  - Translate a symbol into an analog signal to send over the channel
  - Multiple bits may be transmitted in each symbol
  - Rate of symbols sent is called “Baud rate”
  - Note: Baud rate NOT EQUAL TO bits/s

Building blocks, cont.

- Channel
  - Medium that stores/transfer the information
  - Wire, Radio, Optics, Magnetic media etc.
  - Usually distorts the sent information

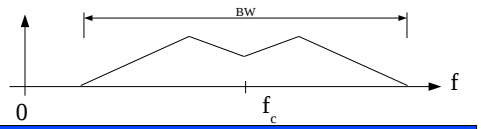
Baseband modulation

- The modulated signal is sent directly in the channel
- Typical example: Modems, ADSL, VDSL etc.
- Signal contains information at low frequencies (possibly also DC)



Band-pass modulation

- The information is embedded in a carrier frequency f_c.
- Information in carrier amplitude, phase, and/or frequency
- Assume sinosoidal carrier
- No information outside f_c +/- BW/2



Digital Communication

- Transmit symbols on to the channel
  - Use different waveforms for different symbols
  - Symbols have a symbol time equal to 1/symbol rate
- Waveform limitations
  - Amplitude limited
  - Length (time) limited
  - Bandwidth limited
  - Should be easy to generate and detect

How to encode information onto an analog waveform

- Amplitude
  - Produce a carrier, change its amplitude
- Phase
  - Generate a carrier, delay it to change phase
- Frequency
  - Vary frequency of carrier
- Shape
  - Select shapes according to symbol to send
  - Non-sinusoid will lead to use of spectrum outside the carrier frequency (e.g., square wave)

Simple cases, sinusiodal carrier

- Amplitude modulation (AM)

  $y(t) = A(t) \sin(2\pi f_c t)$
- Phase modulation (PM)

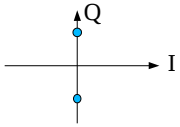
  $y(t) = \sin(2\pi f_c t + Phase(t))$
- Detect value at receiver by comparing with reference
  - Require stable and well matched reference
  - Environment affects received amplitude (attenuation)
  - Environment affects received phase (delays)
- General: $y(t) = A(t)\sin(2\pi f_c t + phi(t))$

Representing amplitude and phase: Constellations

- Transmitted symbols usually drawn as constellation
  - I (inphase) and Q (quadrature) axis describe cosine and sine terms of the carrier sine curve

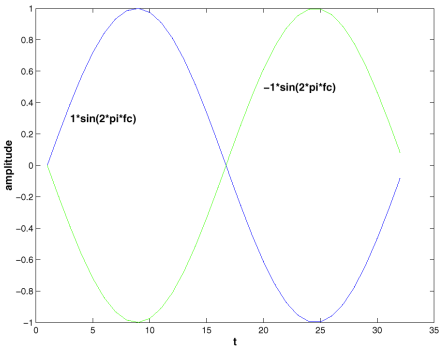
$$A\sin(2\pi f_c t + \phi) = B\sin(2\pi f_c t) + C\cos(2\pi f_c t)$$

- Polar coordinates => amplitude and phase of the carrier signal
- Example f(x,t) => f(0,t) = 1
  f(1,t) = -1



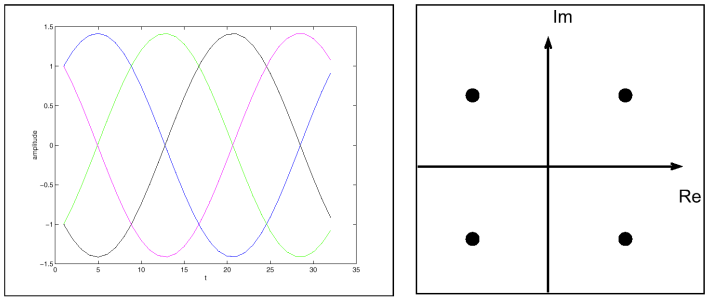
$y(t) = f(x,t)\sin(2\pi f_c t)$

Binary Phase Shift Keying (BPSK)



Phase Shift Keying (PSK)

- Constant amplitude, only phase changes (in example starts at π/4, 3π/4, 5π/4, 7π/4)



PSK cont.

- Transmitted signal

$$\cos(2\pi f_c t - \varphi_x) = \cos(\varphi_x)\cos(2\pi f_c t) + \sin(\varphi_x)\sin(2\pi f_c t)$$

$$= f_I(x,t)\cos(2\pi f_c t) + f_Q(x,t)\sin(2\pi f_c t)$$

- Rewrite using complex representation

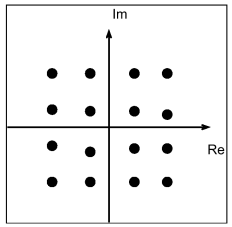
$$= f_I(x,t)\cos(2\pi f_c t) + j \cdot f_Q(x,t)\sin(2\pi f_c t) = f_{I,Q}(x,t) \cdot e^{j2\pi f_c t}$$

- $F_I(x,t)$ and $F_Q(x,t)$ is selected differently depending on modulation method

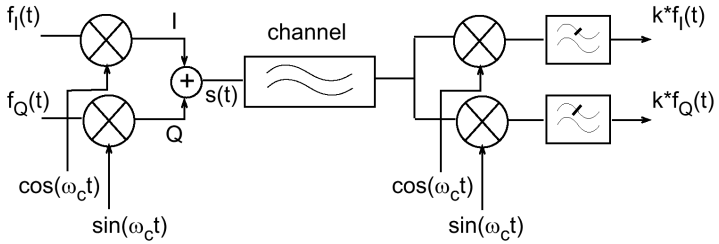
Quadrature Amplitude Modulation (QAM)

- Modulate both amplitude and phase
- Use equal distance between all points

16-QAM



Creating the modulated carrier



Transmitter Reciever

Channel Model

- Distorts the signal due to reflections and diffractions, effects are time-variant
- Multipath: The transmitted signals travels through many different paths before reaching the reciever
- Delay spread, attenuation, frequency broadening
- Simplest model treats the channel distortion as noise (Additive White Gaussian Noise, AWGN)

Channel Model, cont.

- Fading occurs when phase of the signals from two paths differs by approximately π
- Output from the channel (LTI channel model)

$$f_{rec}(x,t) = f(x,t) \otimes h(t) + n_{ch}(t) \quad \textit{or in the Fourier domain}$$

$$F_{rec}(e^{j\Omega}) = F(e^{j\Omega}) \cdot H(e^{j\Omega}) + N_{ch}(e^{j\Omega})$$

where h(t) is channel impulse response, and $n_{ch}(t)$ is the noise in the channel

- Output consists of rotated and scaled version of the input signal plus noise

Channel Model, cont.

- Common channel models are Rayleigh fading channel and Rice fading channels.
- Multipath spread (T_m) corresponds to time between first and last received signal version
- Coherence bandwidth $B_m = 1/T_m$
- If signal bandwidth << B_m => flat fading channel
- Typical indoor channel has T_m less than 200 ns (usually less than 100 ns)

Channel Model, cont.

- Channel characteristics are time-varying
- Coherence time: Duration over which the channel characteristics do not change significantly
- If coherence time > time to send a number of symbol, then the channel is a slow fading channel
- Indoor channels are frequency-selective slowly fading channels

Wireless communication, cont.

- High data rate => Large bandwidth (BW)
- BW >> Coherence bandwidth => frequency selective fading
  - Coherence bandwidth = $1/T_m$ (multipath spread)
- Possible solutions
  - Use multiple narrow carriers
  - Use equalizer to even out effect (inverse of impulse response)

Multi-carrier Multiplexing

- Reduce effects of frequency selective fading by use of multiple carriers
- Each carrier must be non-overlapping with the other carriers to enable detection of data
- Leads to inefficient use of bandwidth