Point-to-point Data Link, Collaborative Whiteboard, and Voice Conference

MIT 6.111 Final Project
Scott B. Ostler and Michael F. Robbins

Project Goals

- Transmit digital data over a noisy analog channel
- Demonstrate digital signal processing techniques and error detection/correction
- Build packet engine capable of interfacing pseudo-serial channel to multiple application modules
- Implement applications at different necessary quality-of-service levels

Project Overview

- Analog channel: visible light, indoors, over LED/photodiode pairs
- Encoding: Orthogonal Frequency Division Multiplexing (OFDM), symbol redundancy
- Applications:
  - Voice conference (AC97 codec, microphone & headseat)
  - Shared whiteboard (VGA monitor & PS/2 mouse)

Modes of Operation

- Local Mode
  - no channel hardware. packet engine handles the connection.
- Loopback Mode
  - one labkit over real channel (uni-directional)
  - applications are effectively split between sending and receiving parts
- Link Mode
  - two labkits over real channel (bi-directional)

About OFDM

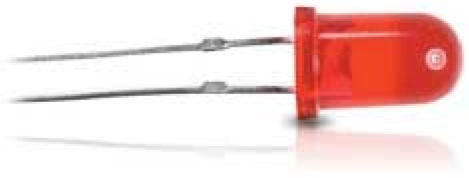
- Orthogonal Frequency Division Multiplexing
  - basically, turning particular frequency bins of an FFT on or off!
  - very good resistance to DC ambient and to narrowband noise (as is lots of environmental noise, i.e. lighting, monitors, etc.)
  - bit errors become “bin errors”
- OFDM is used in ADSL, 802.11g, WiMax, and other modern communications links

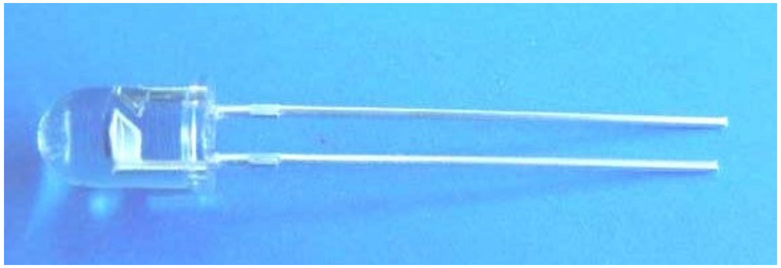
Channel Timing Details

- 32 frequency bins (to make purely real function) ==> 64 samples, played at a rate of 1 sample/us.
- The 64 samples are played twice, such that the receiving station – with any phase offset – will get at least one received frame that is purely the desired bin.
- Guard frame allows partial phase detection.

Physical Channel

- Transmitter: Generic red LED
- Receiver: Everlight PD333-3C-H0-2L
- Fairly simple op-amp circuits to do:
  - current control loop for LED
  - transimpedance amplifier for PD

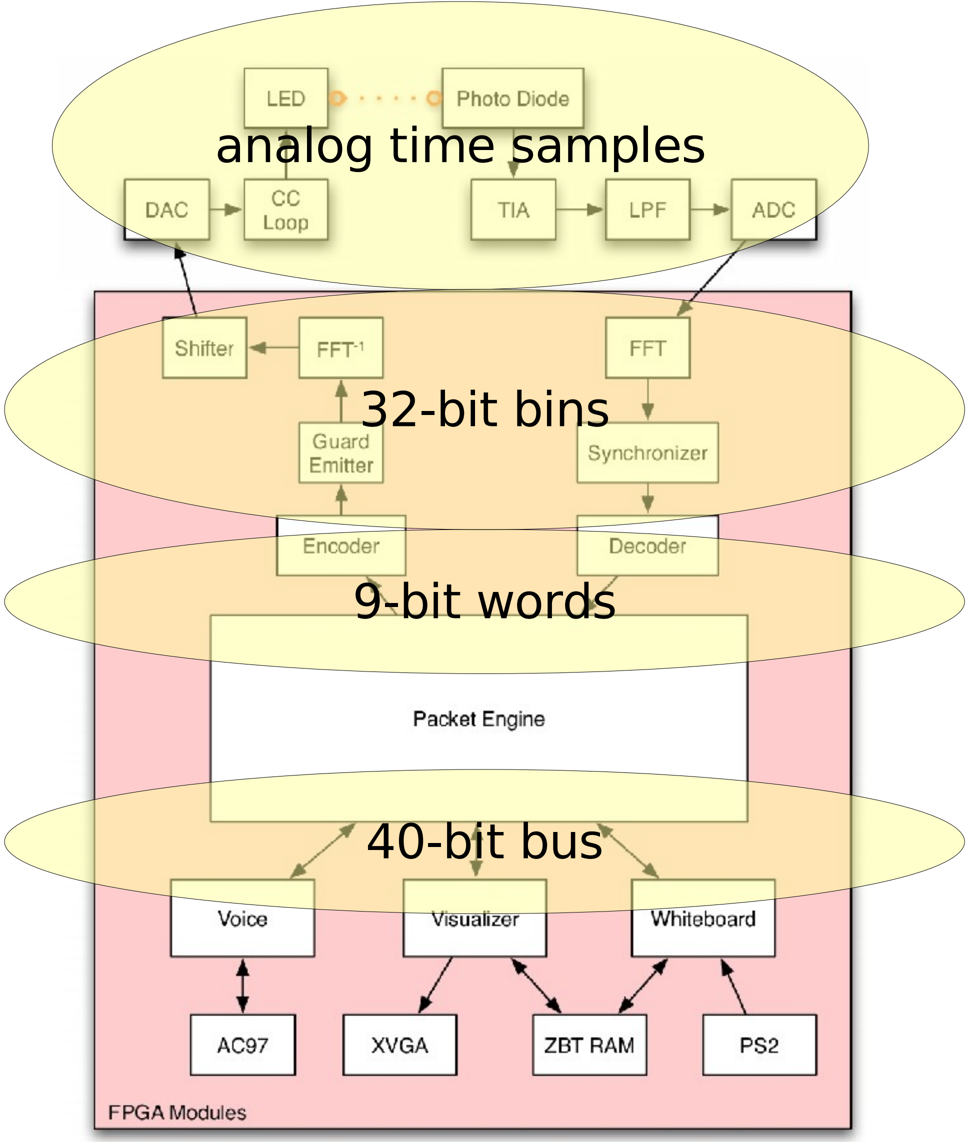




Analog <--> Digital Front End

- requirements: 1MS/s, serial interface, low pin-count package
- AD7476: 1MS/s, SPI serial output, 12-bit resolution (6 pin SOT package)
- AD5399: 1.25MS/s, Serial input, 12-bit resolution (10 pin MSOP package)
  - samples of both are enroute from ADI
- Bonus: digitally-controlled gain (2nd output from DAC + analog multiplier)

Information Flow



Encoder and Shifter

- Encoder: combinational
  - 9-bit word from Packet Engine gets converted into 32-bit representation of which FFT bins are on/off
  - Each bit from the 9-bit word is placed in 3 frequency bins, yielding redundancy
- Shifter: clocked @ 27MHz w/ 1MHz enable
  - module runs FFT^{-1} on bins and sends a new sample once every 1us.

FFT and Threshold

- Because samples played twice, we don't care about phase – only magnitude.
- Stores mag^2 from guard frame frequencies (needed for Frame Synchronizer anyway)
- Uses half of the smallest mag^2 values of the guard frequencies as the 0/1 threshold.
- Outputs 32-bit bins to synchronizer.

Frame Synchronizer

- Recognize guard frame (2 of 3 consensus on particular frequency bins)
- Get bin magnitudes^2 directly from FFT, to decide between leading/lagging partial guard frames
- Once guard frame is found, pass even-numbered frames on to Decoder, and discard odd-numbered ones.

Decoder

- Combinational module to make best guess at transmitted 9-bit word.
- Essentially inverts the scheme used in the encoder, where multiple frequency bins correspond to the same bit. (2 of 3 rule)
- Bonus: a more advanced encoding to tolerate more “bin errors”

Packet Engine

- Converts between variable bit Application packets and 9bit words
- Handles startword checksums
- Periodically pauses sending normal 9-bit words and instructs encoder to send guard frame

Voice & AC97

- 40bit bus to Packet Engine
- Sample at 4KHz, 8bit
- Buffer incoming packets to smooth playback
- Buffer and transmit outgoing packets in groups of five or more to reduce transmission overhead (start word and checksum)

Whiteboard & PS/2

- Turns PS/2 Mouse instructions into drawing instructions
- Writes to ZBT framebuffer during vblank
- 3bits/pixel, 1024x768, ~2 MB
- Basic Functionality: pixel-wide pencil, clear button
- Bonus: extra tools, color, brush size

Visualizer & XVGA

- Uses XVGA module from Lab 5, ZBT memory
- Basic: display Whiteboard framebuffer and controls
- Bonus: Overlay packet information / detected error statistics from Packet Engine

Project Risk Management

- Optical component difficulties
  - Alternative: replace with (analog) wire
- Analog front-end difficulties
  - Alternative: replace with bus of time samples
- FFT difficulties
  - Alternative: reimplement with SINE lookup
- Audio throughput
- Packet processing complexity

Approximate Schedule

- Thanksgiving
  - Testable applications (Local mode) *(Scott)*
  - Working chain Encoder --> Digital (in FPGA) loopback --> Decoder *(Mike)*
- November 24th
  - Packet engine *(Scott and Mike)*
- December 4th
  - Analog loopback *(Mike)*
  - Obtain second labkit *(Scott)*

Summary

- Several complicated components
- Risk mitigation strategies exist for most-likely issues
- Ultimately:
  - a project that shows off the FPGA with lots of real-world-applicable pieces and challenges
  - end product with “wow, neat” factor
  - lots of room for future system improvement