



## Potpourri

- Serial communication links
- FFTs
- RFID
- Useful Tools

## Interconnects dominate performance and power in digital systems

IBM blue gene/L



## Interconnects dominate performance and power in digital systems

IBM blue gene/L



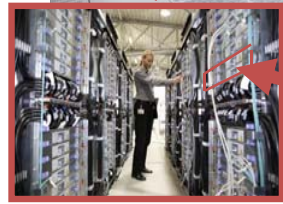
## Interconnects dominate performance and power in digital systems

IBM blue gene/L



# Interconnects dominate performance and power in digital systems

IBM blue gene/L



6.111 Fall 2016



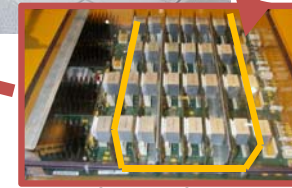
Lecture 13

# Interconnects dominate performance and power in digital systems

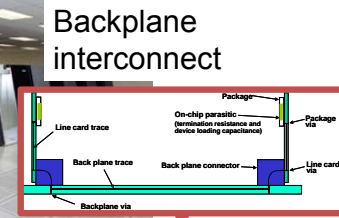
IBM blue gene/L



6.111 Fall 2016



Lecture 13



# Interconnects dominate performance and power in digital systems

IBM blue gene/L



6.111 Fall 2016



Lecture 13



compute cards

# Interconnects dominate performance and power in digital systems

IBM blue gene/L

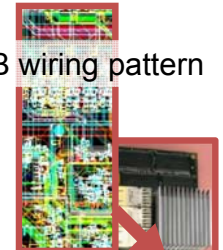


6.111 Fall 2016



node cards

Lecture 13

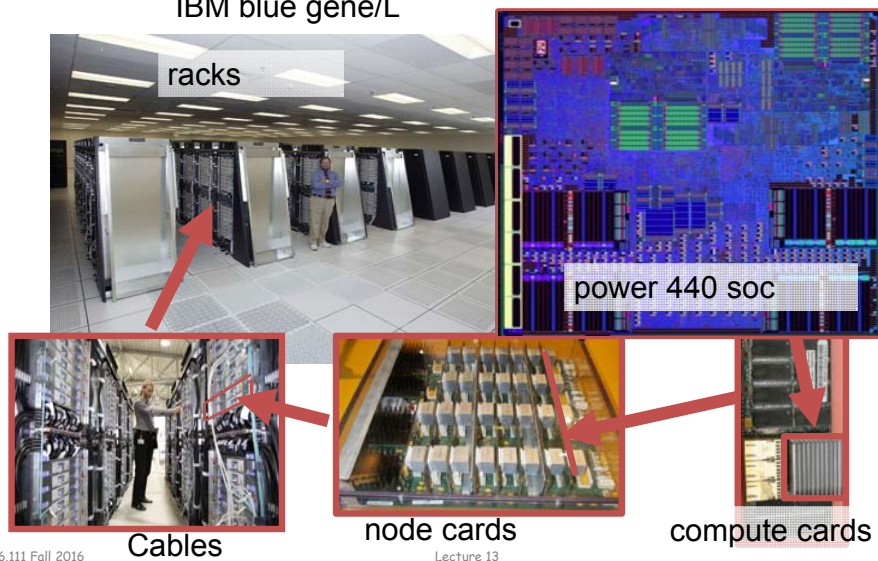


compute cards



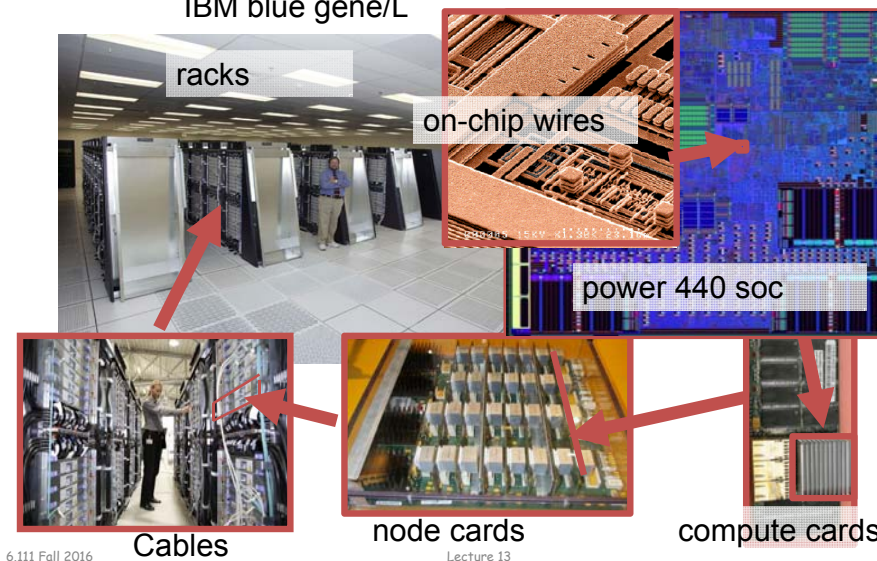
# Interconnects dominate performance and power

IBM blue gene/L

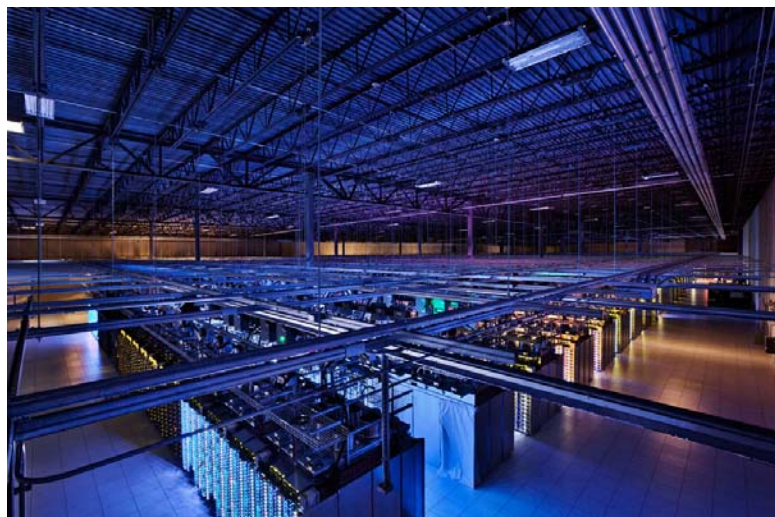


# Interconnects dominate performance and power

IBM blue gene/L



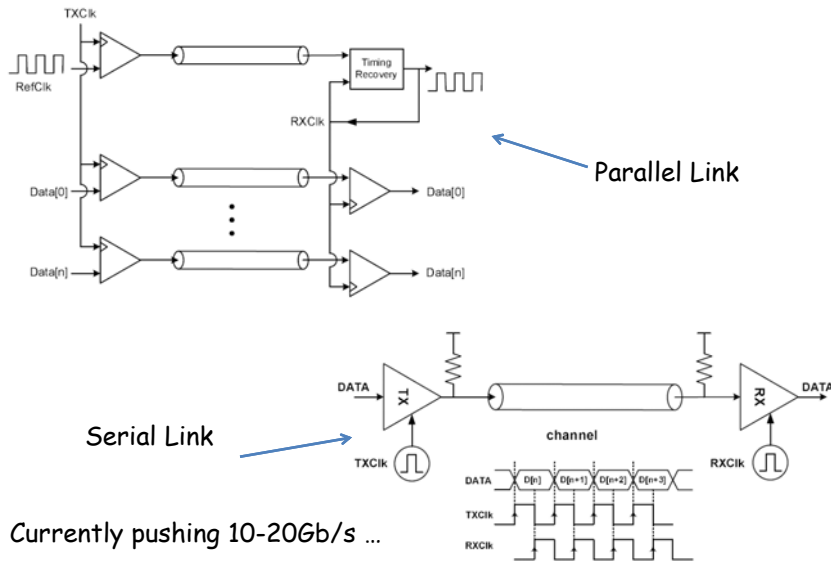
# Google Server Farm



# Backups - Tapes



## Serial and Parallel Links



## Serial Communications

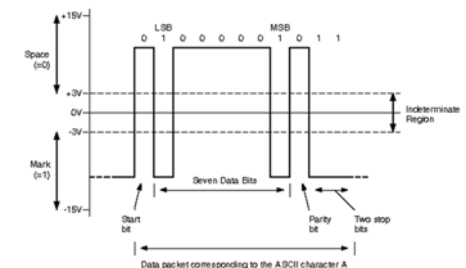
- Sending information one bit at a time vs. many bits in parallel
  - Serial: good for long distance (save on cable, pin and connector cost, easy synchronization). Requires "serializer" at sender, "deserializer" at receiver
  - Parallel: issues with clock skew, crosstalk, interconnect density, pin count. Used to dominate for short-distances (eg, between chips).
  - BUT modern preference is for parallel, but independent serial links (eg, PCI-Express x1,x2,x4,x8,x16) as a hedge against link failures.
- A zillion standards
  - Asynchronous (no explicit clock) vs. Synchronous (CLK line in addition to DATA line).
  - Recent trend to reduce signaling voltages: save power, reduce transition times
  - Control/low-bandwidth Interfaces: SPI, I<sup>2</sup>C, 1-Wire, PS/2, AC97
  - Networking: RS232, Ethernet, T1, Sonet
  - Computer Peripherals: USB, FireWire, Fiber Channel, Infiniband, SATA, Serial Attached SCSI

## RS232 (aka "serial port")

- Labkit: simple bidirectional data connection with computer.
- Characteristics
  - Large voltages => special interface chips (1/mark: -12V to -3V, 0/space: 3V to 12V)
  - Separate xmit and rcv wires: full duplex
  - Slow transmission rates (1 bit time = 1 baud); most interfaces support standardized baud rates: 1200, 2400, 4800, 9600, 19.2K, 38.4K, 57.6K, 115.2K
  - Format
    - Wire is held at 1/mark when idle
    - Start bit (1 bit of "0" at start of transmission)
    - Data bits (LSB first, can be 5 to 8 bits of data)
    - Parity bit (none, even, odd)
    - Stop bits (1, 1.5 or 2 bits of 1/mark at end of symbol)
    - Most common 8-N-1: eight data bits, no parity, one stop bit

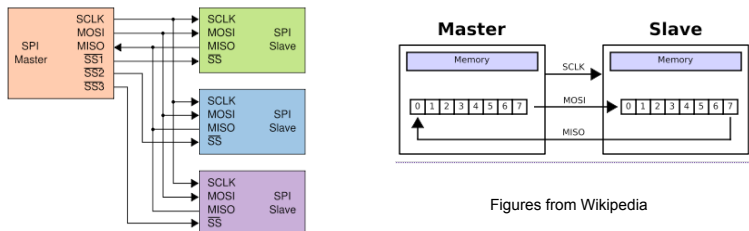
## RS232 interface

- Transmit: easy, just build FSM to generate desired waveform with correct bit timing
- Receive:
  - Want to sample value in middle of each bit time
  - Oversample, eg, at 16x baud rate
  - Look for 1->0 transition at beginning of start bit
  - Count to 8 to sample start bit, then repeatedly count to 16 to sample other bits
  - Check format (start, data, parity, stop) before accepting data.



# SPI (Serial Peripheral Interface)

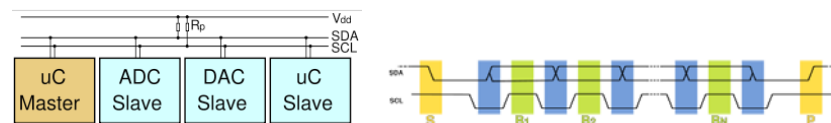
- Simple, 3-wire interface + devices selects
  - SCLK generated by master (1-70MHz). Assert data on one edge, sample data on the other. Default state of SCLK and assignment of edges is often programmable.
  - Master Out Slave In (MOSI) data shifted out of master register into slave register
  - Master In Slave Out (MISO) data shifted out of slave register and into master register
  - Selects (usually active low) determine which device is active. Assertion often triggers an action in the slave, so master waits some predetermined time then shifts data.



Figures from Wikipedia

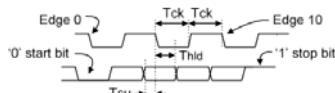
# I<sup>2</sup>C (Inter-Integrated Circuit)

- 2 open-drain wires (SCL = clock, SDA = data)
- Multiple-master, each transmission addresses a particular device, many devices have many different sub-addresses (internal registers)
- Format (all addresses/data send MSB first):
  - Sender: Start [S] bit (SDA↓ while SCL high)
  - Sender: One or more 8-bit data packets, each followed by 1-bit ACK
    - Data changed when SCL low, sampled at SCL↑
    - Receiver: Active-low ACK generated after each data packet
  - Sender: Stop [P] bit (SDA↑ while SCL high)
- SCL and SDA have pullup resistors, senders only drive low, go high-impedance to let pullups make line high (so multiple drivers okay!)
  - Receiver can hold SCL low to stretch clock timing, sender must wait until SCL goes high before moving to next bit.
  - Multiple senders can contend using SDA for arbitration

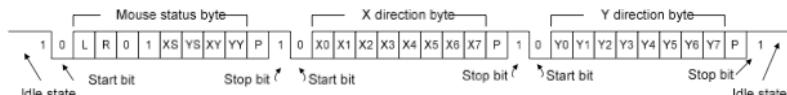


# PS/2 Keyboard/Mouse Interface

- 2-wire interface (CLK, DATA), bidirectional transmission of serial data at 10-16kHz
- Format
  - Device generates CLK, but host can request-to-send by holding CLK low for 100us
  - DATA and CLK idle at "1", CLK starts when there's a transmission. DATA changes on CLK↑, sampled on CLK↓
  - 11-bit packets: one start bit of "0", 8 data bits (LSB first), odd parity bit, one stop bit of "1".
  - Keyboards send scan codes (not ASCII!) for each press, 8'hF0 followed by scan code for each release
  - Mice send button status, Δx and Δy of movement since last transmission



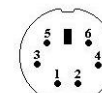
| Symbol           | Parameter                | Min  | Max  |
|------------------|--------------------------|------|------|
| T <sub>CK</sub>  | Clock time               | 30us | 50us |
| T <sub>SU</sub>  | Data-to-clock setup time | 5us  | 25us |
| T <sub>HLD</sub> | Clock-to-data hold time  | 5us  | 25us |



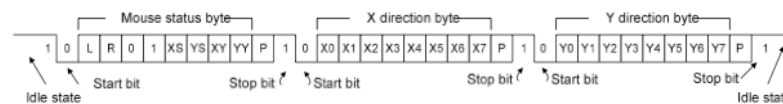
Figures from digilentinc.com

# PS/2 Keyboard/Mouse Interface

- 2 signal wire interface (CLK, DATA), bidirectional transmission of serial data at 10-16kHz



| Pin | Signal | In/Out |
|-----|--------|--------|
| 1   | Data   | Out    |
| 2   | N/C    |        |
| 3   | Ground |        |
| 4   | +5V    |        |
| 5   | Clock  | Out    |
| 6   | N/C    |        |



Figures from digilentinc.com



# IDE Bus - Serial ATA (SATA)

| 40 Pin IDE Connector PinOut |                 |       |                 |
|-----------------------------|-----------------|-------|-----------------|
| Pin #                       | Signal Function | Pin # | Signal Function |
| 1                           | Reset           | 2     | Ground          |
| 3                           | Data 7          | 4     | Data 8          |
| 5                           | Data 6          | 6     | Data 9          |
| 7                           | Data 5          | 8     | Data 10         |
| 9                           | Data 4          | 10    | Data 11         |
| 11                          | Data 3          | 12    | Data 12         |
| 13                          | Data 2          | 14    | Data 13         |
| 15                          | Data 1          | 16    | Data 14         |
| 17                          | Data 0          | 18    | Data 15         |
| 19                          | Ground          | 20    | Key             |
| 21                          | DMARQ           | 22    | Ground          |
| 23                          | DIOV-           | 24    | Ground          |
| 25                          | DIOV+           | 26    | Ground          |
| 27                          | IORDY           | 28    | CSEL            |
| 29                          | DMARK-          | 30    | Ground          |
| 31                          | INTRQ           | 32    | IOCS16-         |
| 33                          | DA1             | 34    | FDIAG-          |
| 35                          | DA0             | 36    | DA2             |
| 37                          | CS1FX-          | 38    | CS3FX-          |
| 39                          | DASP-           | 40    | Ground          |

SATA

| Pin | Name |
|-----|------|
| 1   | GND  |
| 2   | A+   |
| 3   | A-   |
| 4   | GND  |
| 5   | B-   |
| 6   | B+   |
| 7   | GND  |

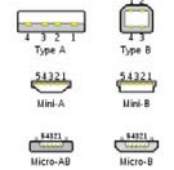
2-wire (+,-) for high-speed

SATA 1: 1.5Gb/s  
SATA 2: 3Gb/s  
SATA 3: 6Gb/s

# USB - Universal Serial Bus

- USB 1.0 (12 Mbit/s) introduced in 1996, USB 2.0 (480 Mbit/s) in 2000, USB 3.0 (5 Gbit/s) in 2012, USB-C 2016.
- Created by Compaq, Digital, IBM, Intel, Northern Telecom and Microsoft.
- Uses differential bi-direction serial communications

Type A USB 2.0 - 4 pins

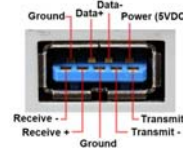


Type A & B Pinout

| Pin | Name | Cable color | Description |
|-----|------|-------------|-------------|
| 1   | VCC  | Red         | +5 V        |
| 2   | D-   | White       | Data -      |
| 3   | D+   | Green       | Data +      |
| 4   | GND  | Black       | Ground      |

Mini/Micro Pinout

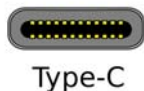
| Pin | Name | Color | Description  |
|-----|------|-------|--|
| 1   | VCC  | Red   | +5 V   |
| 2   | D-   | White | Data -   |
| 3   | D+   | Green | Data +   |
| 4   | ID   | none  | permits distinction of Micro-A- and Micro-B-Plug<br>Type A: connected to Ground<br>Type B: not connected |
| 5   | GND  | Black | Signal Ground  |



USB 3.0

# USB - C

- Universal connector for power and data - first product MacBook Air - one and only port!
- Symmetrical - no orientation
- Supports DisplayPort, HDMI, power, USB, and VGA. Uses differential bi-direction serial communications
- Supplies up to 100W power
- New adapters required for DisplayPort, HDMI, power, USB, and VGA - omg!



Type-C



Actual renderings courtesy of Foscoion. USB Type-C™ and USB-C™ are trademarks of USB Implementers Forum.

Figure 2-1 USB Type-C Receptacle Interface (Front View)

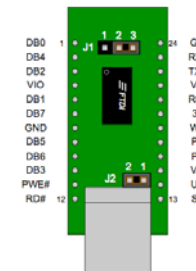
| A1  | A2   | A3   | A4   | A5   | A6 | A7 | A8   | A9   | A10  | A11  | A12 |
|-----|------|------|------|------|----|----|------|------|------|------|-----|
| GND | TX1+ | TX1- | Vbus | CC1  | D+ | D- | SBU1 | Vbus | RX2- | RX2+ | GND |
| GND | RX1+ | RX1- | Vbus | SBU2 | D- | D+ | CC2  | Vbus | TX2- | TX2+ | GND |
| B12 | B11  | B10  | B9   | B8   | B7 | B6 | B5   | B4   | B3   | B2   | B1  |

# USB (Universal Serial Bus)

- 2-wire (D+,D-) for high-speed, bidirectional polled transmission between master and addressable endpoints in multiple devices. Full speed (12Mbps) and High speed (480Mbps) data rates.
- Multi-level tiered-star topology (127 devices, including hubs)
- FTDI UM245R USB-to-FIFO module for bidirectional data transfer using a handshake protocol, also asynchronous "bit-bang" mode with selectable baud rates.
  - 24-pin DIP module, wire to user pins
  - Drivers for Windows workstations in lab



Figures from ftdi.com

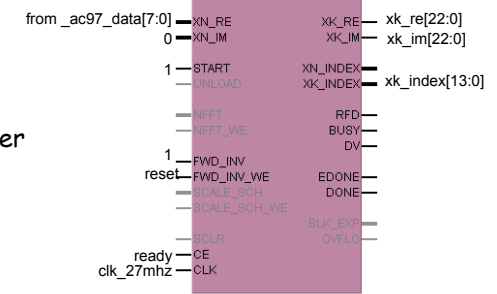


# Audio Feature Extraction

- Most features are best recognized in the frequency domain
- Use Discrete Fourier Transform
  - Algorithm used: Fast Fourier Transform (FFT)
  - Input: N data values acquired at sample frequency  $\omega_s$ 
    - Nyquist rate is  $\omega_s/2$
  - Output: N complex values representing DFT coefficients in the frequency range  $-\omega_s/2$  to  $+\omega_s/2$ .
    - Each value covers a frequency range of  $\omega_s/N$
    - Indices  $(0, (N/2)-1)$  are for frequencies  $i*(\omega_s/N)$
    - Indices  $(N/2, N-1)$  are for frequencies  $-\omega_s/2 + (i - N/2)*(\omega_s/N)$
  - If N is even, output is symmetric, so we can calculate magnitude using only positive frequencies. Magnitude  $\approx \sqrt{r^2 + i^2}$  constant factors.
- Example
  - Audio data from AC97 sampled at 8kHz
  - 2048 data points => 2048-point FFT
  - 2048 complex results, each result covers  $8k/2048 = 4Hz$  range

# FFT example - Labkit

- IP wizard will build a N-point FFT module
  - WARNING: they're big!
- In theory, there are two operating modes (select at build time)
  - "pipelined" where you get a complex value out for every sample you send the module - runs continuously
  - "burst" where you load up N samples, wait a while and get your answer while loading the set of samples.
- To use FFT, use sample Verilog
- Demo: audio spectrum analyzer
  - Uses "pipelined" mode
- 44 page datasheet



# FFT - Nexys4

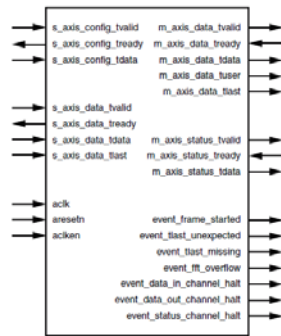
- IP core uses AXI4 protocol
- 97 page datasheet

Table 3-6: Data Input Channel TDATA Fields

| Field Name | Width    | Padded | Description  |
|------------|----------|--------|--|
| XN_RE      | $b_{in}$ | Yes    | Real component ( $b_{in} = 8 - 34$ ) in two's complement or single precision floating-point format.      |
| XN_IM      | $b_{in}$ | Yes    | Imaginary component ( $b_{in} = 8 - 34$ ) in two's complement or single precision floating-point format. |

Table 3-9: Data Output Channel TDATA Fields

| Field Name | Width | Padded              | Description  |
|------------|-------|---------------------|--|
| XK_RE      | $b_k$ | Yes - sign extended | Output data: Real component in two's complement or floating-point format. (For scaled arithmetic and block floating-point arithmetic, $b_k = b_{in}$ . For unscaled arithmetic, $b_k = b_{in} + \log_2(\text{maximum point size}) + 1$ . For single precision floating-point $b_k = 32$ .)                       |
| XK_IM      | $b_k$ | Yes - sign extended | Output data: Imaginary component in two's complement or single precision floating-point format. (For scaled arithmetic and block floating-point arithmetic, $b_k = b_{in}$ . For unscaled arithmetic, $b_k = b_{in} + \log_2(\text{maximum point size}) + 1$ . For single precision floating-point $b_k = 32$ .) |



# AXI Protocol

- Separate data and address connections for reads and writes: simultaneous, bidirectional data transfer.
- Useful for memory mapped applications

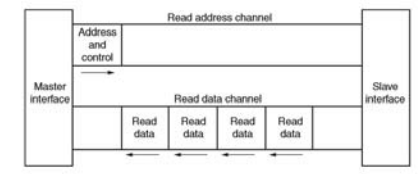


Figure 1-1: Channel Architecture of Reads

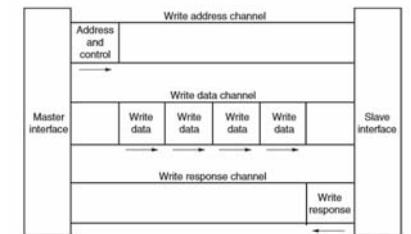


Figure 1-2: Channel Architecture of Writes

## FFT of AC97 data

To process AC97 samples:

- use Pipelined mode (input one sample in each cycle, get one sample out each cycle).
  - FFT expects one sample each cycle, so hook READY to CE so that FFT only cycles once per AC97 frame
- use Unscaled mode, do scaling yourself
  - Number of output bits = (input width) + NFFT + 1
  - NFFT is  $\log_2(\text{size of FFT})$
- let number of FFT points = P, assume 48kHz sample rate
  - there are P frequency bins
  - positive freqs in bins 0 to (P/2 - 1)
  - negative freqs in bins (P/2) to (P-1)
  - each bin covers (48k/P)Hz
  - Use XK\_INDEX to tell which bin's data you're getting out
  - Typically you want magnitude =  $\sqrt{xk\_re^2 + xk\_im^2}$

## Iterative Sqrt module

```
// takes integer square root iteratively
module sqrt #(parameter NBITS = 8, // max 32
              MBITS = (NBITS+1)/2)
    (input wire clk, start,
     input wire [NBITS-1:0] data,
     output reg [MBITS-1:0] answer,
     output wire done);
    reg busy;
    reg [4:0] bit;
    // compute answer bit-by-bit, starting at MSB
    wire [MBITS-1:0] trial = answer | (1 << bit);

    always @(posedge clk) begin
        if (busy) begin
            if (bit == 0) busy <= 0;
            else bit <= bit - 1;
            if (trial*trial <= data) answer <= trial;
        end
        else if (start) begin
            busy <= 1;
            answer <= 0;
            bit <= MBITS - 1;
        end
    end

    assign done = ~busy;
endmodule
```

## Tools

- Labkit hardware with sample Verilog
  - NTSC Camera - display BW images
  - ZBT Memory - high speed memory two 512Kx36 banks
  - Alphanumeric data with hex display
  - Compact Flash - 128Mbits non-volatile memory
- Nexys4 hardware with sample Verilog
  - VGA Camera
  - SD card read/write
- Application support
  - Sound - Matlab script: convert wav files to AC97 8bit COE file
  - Images - Matlab script: convert BMP COE field
  - USB PC-Labkit data transfer
- GIT - Shared project team repository with version control

## FPGAs @ Home

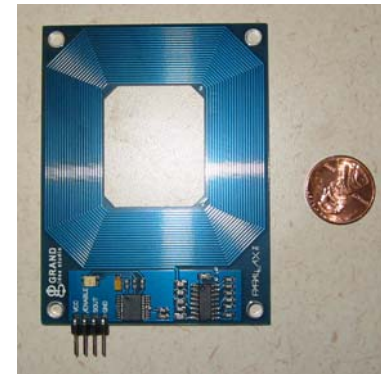
- 6.111 labkit: the Lexus (but an old one) of FPGA protoboards
    - XC2V6000 (67,586 LUT/FFs, 144 BRAMs)
  - Three affordable alternatives (lots more out there)
    - Basys 2 Board ([www.digilentinc.com](http://www.digilentinc.com))
      - \$99 = Spartan 3E-250 (4,896 LUT/FFs, 12 BRAMs)
      - Switches, buttons, leds, 4-digit seven-segment display
    - Basys 3 Board ([www.digilentinc.com](http://www.digilentinc.com))
      - \$79 = Artix-7 (20,800 LUT/FFs, 10 BRAMs)
      - Switches, buttons, leds, 4-digit seven-segment display
    - Nexys4-DDR Board
      - \$159 = Artix-7 (63,400 LUT, 270 BRAMs)
      - 450Mhz clock, audio
- 1 BRAM = 18Kb



# RFID- Radio Frequency Identification

- Used to provide remote interrogation/identification.
  - Frequency bands:
    - 125 - 134 kHz [MIT ID]\*
    - 13.56 MHz [US Passports]\*
    - 400 - 960 MHz UHF [EZPASS 915mhz ~ 1 mw]\*\*
    - 2.45 GHz
    - 5.8 GHz
- \* excitation/broadcast powered  
 \*\* battery powered

# 125khz RFID



125khz transmitter

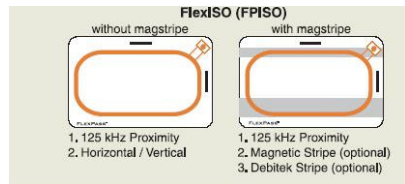


Receiver  
 Powered by 125khz broadcast signal

# RFID Internals

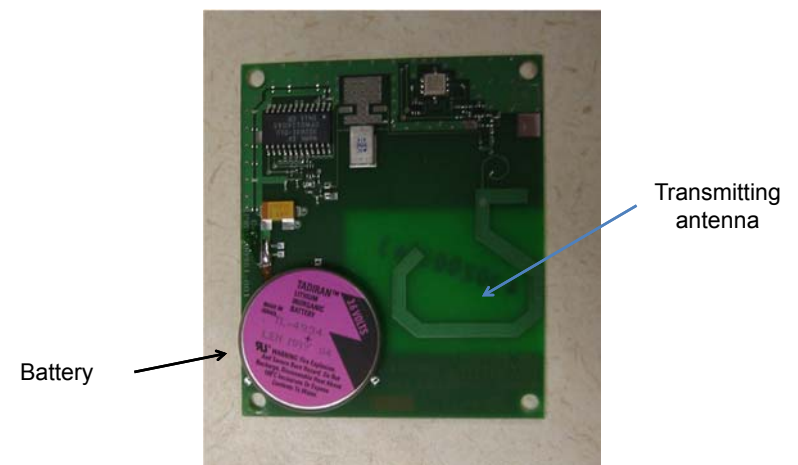
## Legend

- Contact smart chip module
- 125 kHz proximity antenna and chip

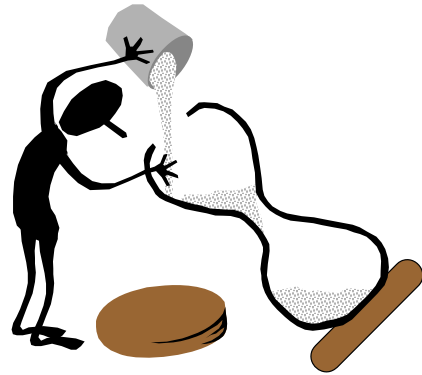


\*[http://groups.csail.mit.edu/mac/classes/6.805/student-papers/fall04-papers/mit\\_id/#specs](http://groups.csail.mit.edu/mac/classes/6.805/student-papers/fall04-papers/mit_id/#specs)

# EZ-pass Internals



See you in lab!



Final project represents 72 hours of credit, so you should average 2-3 hours/day of work on your project assuming you give yourself the occasional day off...