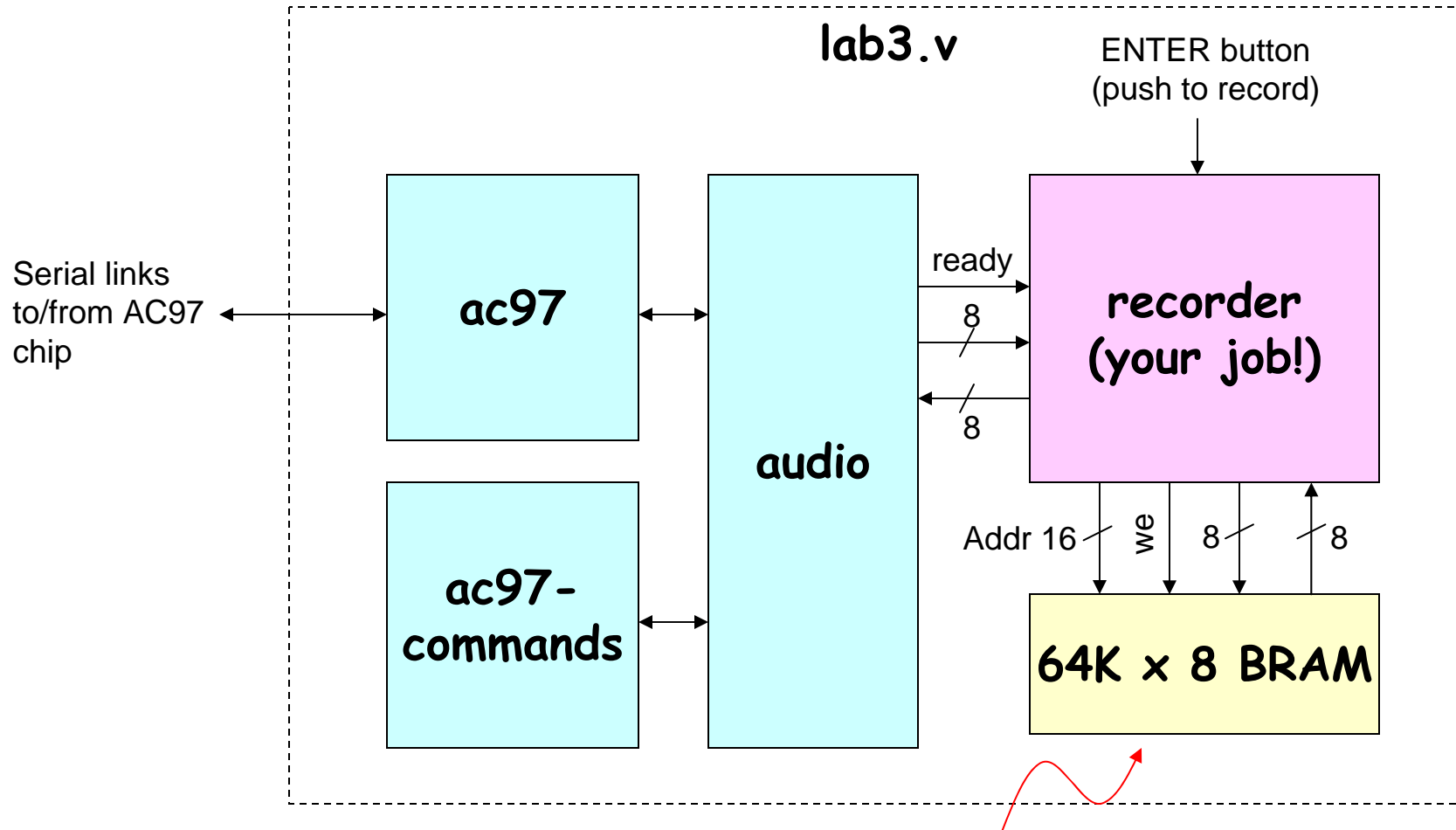


# Lab 3 overview

Assignment: build a voice recorder that records and plays back 8-bit PCM data @ 6KHz

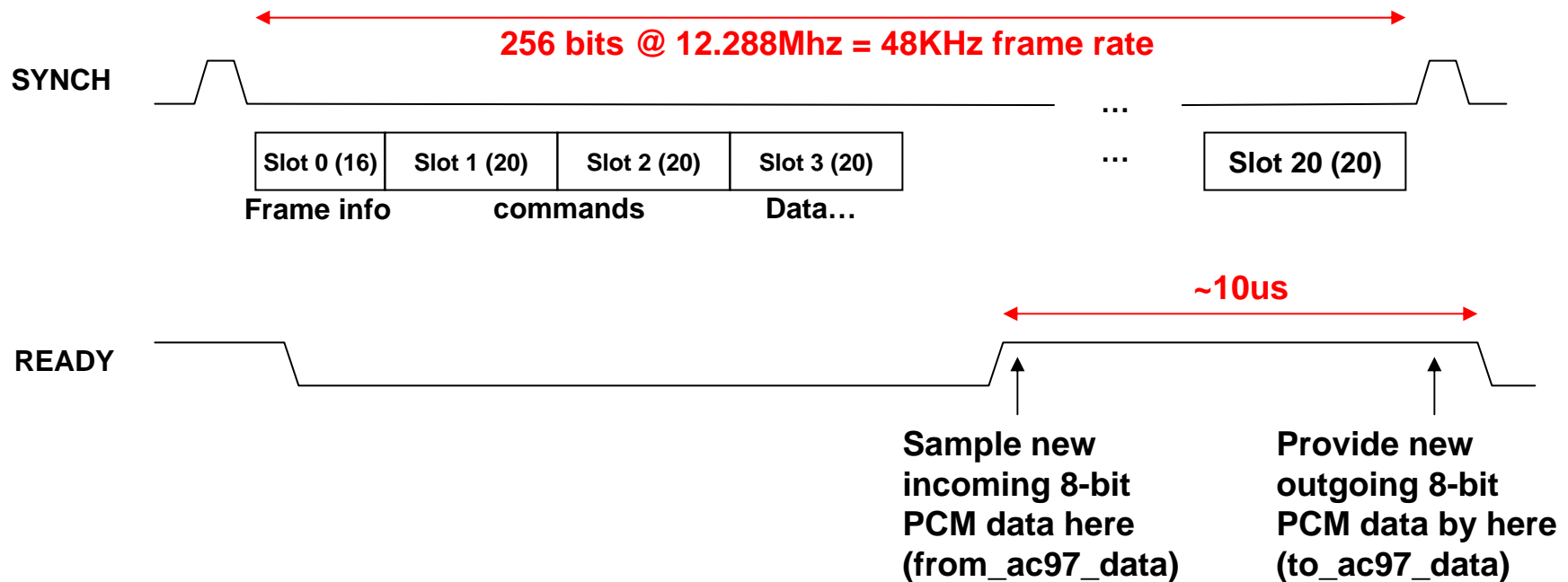
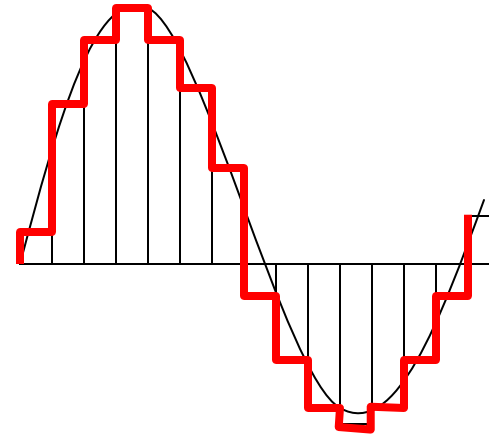


About 11 seconds of speech @ 6KHz

# AC97: PCM data

PCM = pulse code modulation

Sample waveform at fixed intervals, encode results as an N-bit signed number. For our AC97 chip, N = 18.



# 6.111 Final Project



# Final Project: Schedule

- **Choose teams** of two (or maybe three). A single person project requires approval of lecturers.
- **Project Abstract** (due Oct. 31, submit on-line)
  - Start discussing ideas now with 6.111 staff
  - About 1 page long, clearly identify who's doing what
- **Proposal Conference** with TAs (by Nov. 4)
  - Bring your proposal with you *and* submit on-line
- **Block Diagram Conference** with TAs (by Nov. 11)
  - Review major components and overall design approach
  - Each group in discussion with TA creates a **checklist of deliverables** (i.e., what we can expect each team member to demonstrate) - finalized by Nov 17.
  - Specify the device components you need to acquire (*small* budget allocated for each project if component does not exist in the stock room). Get approval from the 6.111 staff and your TA will contact John Sweeney to obtain the parts.

# Schedule (cont'd.)

- **Project Design Presentation** to staff (Nov 14 - 16)
  - Each group will make a 15 min electronic presentation (~10 slides)
  - Submit PDF on-line, will be posted on website
  - Example: S2004 FROGGER presentation slides
- **Final Project Presentations** (Dec 12 - 14)
  - Videotaped and posted on-line with your permission
- **Final Project Report** (Dec 14, 5p)
  - Submit PDF on-line, will be posted on website
  - Sorry, no late checkoffs or reports will be accepted

# Choosing A Topic

- You only have 6 weeks total (once your proposal abstract is turned in) to do this project.
  - **It is important to complete your project.**
  - It is very difficult to receive an "A" in the class without having something working for the final project.
- The complexity for each team member should be greater than the complexity of the lab assignments.
- Some projects include analog building blocks or mechanical assemblies (infrared, wireless, motors, etc.). However, keep in mind that this is a digital design class and your design will be evaluated on its digital design aspects.
- Complexity, risk and innovation factor.
  - We will give credit to innovative applications, design approaches
  - More complex is not necessarily better
- Look through previous projects for inspiration (see website)

# Some Suggestions

- **Be ambitious!**
  - But choose a sequence of milestones that are increasingly ambitious (that way at least part of your project will work and you can debug features incrementally).
  - But don't expect 400Mhz operating frequencies, etc.
- **It's motivating if there's something to see or hear**
  - Video and graphics projects are fun (and with the new labkit basic video input and output are pretty straightforward which means you can concentrate on the processing)
  - Audio/Music is low-bandwidth, so it's easy to do interesting processing in real-time (real-time is harder with video).

# Grading (40 points Total)

- Report and Presentation (10 points)
- Problem Definition and Relevance, Architecture, Design methodology (10 points)
  - What is the problem
  - Why is it important
  - System architecture and partitioning
  - Design choices and principles used
  - Style of coding
  - All of the above should be stated in the project and report
- Functionality (10 points)
  - Did you complete what you promised (i.e., graded by the checklist)
- Complexity, Innovation, Risk (10 points)



# Design Rules

- Use hierarchical design
  - Partition your design into small subsystems that are easier to design and test.
  - Design each sub-system so they can be tested individually.
  - When appropriate, use Major/Minor FSMs.
- Use the same clock edge for all edge-triggered flip-flops
  - Beware of clock skew, don't gate the clock
- Avoid problems from 'glitches'.
  - **Always assume that combinational logic glitches**
  - Never drive a critical asynchronous control signal (register clock, write enable) from the output of combinational logic.
  - Ensure a stable combinational output before it is sampled by CLK.
  - When needed, create glitch-free signals by registering outputs.

# Design Rules (cont'd.)

- **Avoid tristate bus contention by design**
- **Synchronize all asynchronous signals**
  - Use two back to back registers
- **Use memory properly**
  - Avoid high Z address to SRAM when CE is asserted.
  - Avoid address changes when WE is true.
  - Make sure your write pulse is glitch free.
- **Power supply can be noisy**
  - Use bypass capacitors to filter noise
- **Chip-to-chip communication**
  - Beware of noise (inductance)
  - Might need to synchronize signals
  - Can also use "asynchronous" protocols

# How to Make Your Project Work (see website)

- Sections that are particularly relevant are:
  - Wiring Errors
  - Care and Feeding of the Power Supply
  - Unused Inputs
  - Behavior of Ungrounded Parts
  - Tri-State Logic Signals
  - Handling CMOS Parts
  - Wire Routing
  - Clock Distribution
  - Gating the Clock
  - RAM Write Pulses
  - Synchronizer Errors
  - Testing Strategies
  - Driving High Current Devices

Document Courtesy of  
Tom Knight and Don  
Troxel