# 6.111 Final Project

# Final Project: Schedule

- **Choose teams** of two or three. A single person project requires special approval by lecturer.
- **Project Abstract** (Due Nov 1 in class)
  - Start discussing project ideas with the 6.111 staff
  - Abstract should be about 1 page (clearly state the work partition)
- **Proposal Conference** with TAs (Nov 5). Bring your proposal with you.
- **Block diagram Conference** with TAs (Nov 12)
  - Review the major components in the system and your overall design approach
  - Each group in discussion with TA creates a **checklist of deliverables** (i.e., what we can expect to be demonstrated) – by the design presentation time (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** (in 34-101) on Nov 17, 19, 22
  - Each group will make a 10min electronic presentation (~10 PowerPoint slides or viewgraphs)
  - **Everyone is required to attend all days** (not just the days you are presenting) – this will count in your participation grade. Each student will write comments (anonymous) which will be provided to the group as feedback.
  - Example: S2004 FROGGER presentation slides
- **Final project presentations** and video taping (Dec 6/7)
- **Final project report** (in electronic format, which will be published with permission on the course website) due Dec 8 by 5PM (no late project check-offs or reports accepted)
- See website for 6.905 additional units signup

# Choosing A Topic

- You only have 5 weeks total (once your proposal abstract is turned in) to do this project. Be realistic in what you take on.
  - It is important to complete your project.
  - It is very difficult to receive an "A" in the class without completing the final project.
- The complexity should be equal or greater than Lab3 for each student.
- Quite often you will need to include analog building blocks (video, wireless, motors, etc.). However, keep in mind that this is a digital class and your design should demonstrate digital design principles.
- 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)

# Grading (35 points Total)

- Report and Presentation (5 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 (15 points)
  - Did you complete what your promised (i.e., graded by the checklist)
- Complexity, Innovation, Risk (5 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