L14: Final Project Kickoff

Schedule - I

- **Form project teams – this week (nothing to turn in)**

- **Project Abstract (Due April 10th in 38-107 by 1PM)**
  - Start discussing project ideas with the 6.111 staff
  - Each group should meet with Prof. Chandrakasan to discuss the scope of project and breakdown of tasks
  - Abstract should be about 1 page (clearly state the work partition) – a polished abstract will be published on the course website once your project has been finalized. Sample abstracts can be viewed at http://web.mit.edu/6.111/www/s2005/PROJECT/projectgroup.html

- **Work in teams of two or three. A single person project requires special approval by the 6.111 staff.**

- **Proposal Conference with TAs (April 12-14). Bring your detailed proposal with you (3-5 pages including figures).**

- **Block diagram conferences with TAs (April 19-21)**
  - Review the major components in the system and your overall design approach
  - Each group in discussion with TA, creates a deliverables checklist (i.e., what we can expect to be demonstrated) – (Due May 1st in class).
  - 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.

- **Project Design Presentation (in 34-101) on April 24, 26, 28, May 1**
  - ☐ Each group will make an electronic presentation (power point or PDF)
  - ☐ 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 presenting group as feedback.

- **Final project check-off (with teaching staff) on May 16: 30 minutes**

- **Final project presentations and video taping on May 17 (Wednesday): ~3-4 minutes (videos to be posted on the course website with permission)**

- **Final project report (in electronic format, which will be published with permission on the course website) due May 18 by 5PM (no late project check-offs or reports accepted)**

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 larger than lab4 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 and innovation factor.**
  - **We will give credit to innovative applications, design approaches**
  - **More complex is not necessarily better**

Grading (37 points Total)

- **Report and Presentation (6 points)**

- **Problem Definition and Relevance, Architecture, Design methodology (7 points)**
  - ☐ What is the problem and 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 presentation and report

- **Functionality (17 points)**
  - ☐ Did you complete what your promised (i.e., graded by your customized checklist)

- **Innovation (7 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

- **Avoid problems from 'glitches'**
  - Always assume that combinational logic glitches
  - Never clock a register from the output of combinational logic
  - Never drive a critical control signal such as write enable from the output of combinational logic
  - Ensure a stable combinational output before it is sampled by CLK.
  - Create glitch-free signals by
    - Registering outputs
    - Gating the clock

- **Avoid tristate bus contention by design**

- **Synchronize all asynchronous signals**
  - ☐ **Use two back to back registers**

- **Use memory properly**
  - ☐ **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 handout)

- **Read all of the handout**
  - □ It is 'old' but all of it is good advice

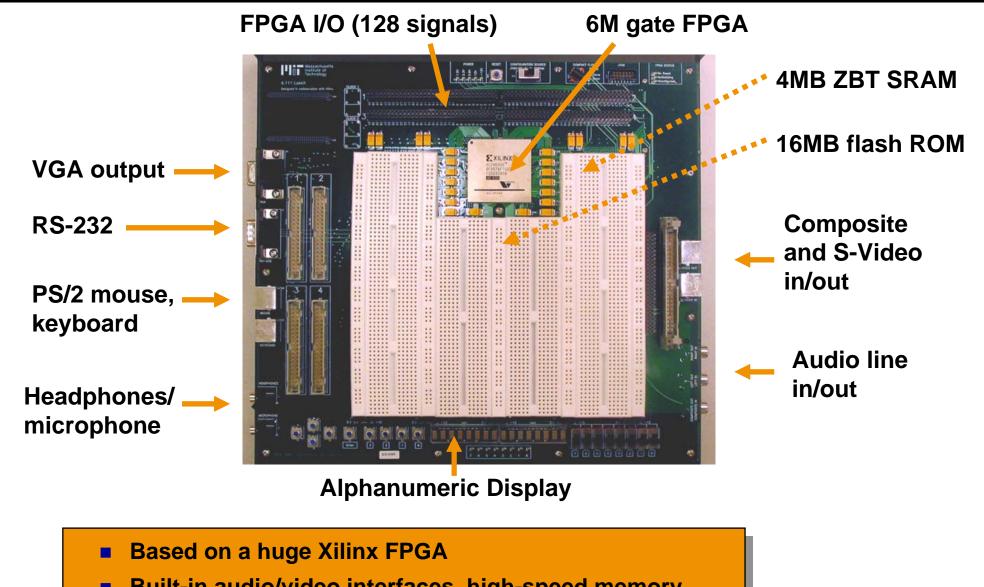- **Sections that are particularly relevant are:**
  - □ Wiring Errors
  - □ Care and Feeding of the Power Supply
  - □ Unused Inputs
  - □ Behavior of Ungrounded Parts
  - □ Tri-State Logic Signals
  - □ Handling CMOS Parts
  - □ Wire Routing
  - □ Clock Distribution
  - □ Gating the Clock
  - □ RAM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

Document Courtesy of Tom Knight and Don Troxel

Labkit

FPGA I/O (128 signals)

6M gate FPGA

4MB ZBT SRAM

16MB flash ROM

VGA output

RS-232

PS/2 mouse, keyboard

Headphones/ microphone

Composite and S-Video in/out

Audio line in/out

Alphanumeric Display

- Based on a huge Xilinx FPGA
- Built-in audio/video interfaces, high-speed memory
- Supports moderately high-speed designs (50-100MHz)