


L14: Quiz Information and Final Project Kickoff

Quiz

- Quiz Review on Monday, March 29 by TAs
  - 7:30 P.M. to 9:30 P.M.
  - Room 34-101
- Quiz will be Closed Book on March 31st (during class time, Location, Walker Gym, 50-340)
  - Covers Problem Sets 1-3, Lectures 1-12, Labs 1-3
- Topics to be covered
  - Combinational Logic: Boolean Algebra, Karnaugh Maps, MSP, MPS, dealing with don't cares
  - Latches and Edge Triggered Registers/Flip-flops
    - Understand the difference between latches, registers and unclocked memory elements (e.g., SR-Flip Flop)
    - Different memory types: SR, D, JK, T
    - Understand setup/hold/propagation delay and how they are computed
  - System Timing (minimum clock period and hold time constraint)
    - Impact of Clock skew on timing
  - Counters and simple FSMs (understand how the '163 and '393 work)
  - FSM design (Mealy/Moore, dealing with glitches)
  - Major and Minor FSM construction
  - Combinational and sequential Verilog coding
    - Continuous assignments, blocking vs. non-blocking, etc.

Quiz (cont.)

- Tri-states basics
- Dealing with glitches
  - When are glitches OK?
  - How do you deal with glitches in digital system design? (registered outputs, appropriate techniques to gate a clock, etc.)
- Arithmetic
  - Number representation: sign – magnitude, Ones complement, Twos complement
  - Adder Structures: Ripple carry, Carry Bypass Adder (Don't worry about Carry lookahead adder details)
  - False Paths and Delay Estimation
  - Shift/add multiplier, Baugh-Wooley Multiplier (Twos complement multiplication)
- Memory Basics
  - Understand differences between DRAM vs. SRAM vs. EEPROM
  - Understand timing and interfacing to the 6264
- Analog building blocks
  - Basic Op Amps Circuits
  - D to A: R-2R ladder, Thermometer coding
  - A to D: Successive Approximation architecture, Flash
  - Understand the timing of the AD670 and AD558
- FPGA architectures
  - Programmable architecture (SRAM based, anti-fuse based, EEPROM based programmability)
  - How to map logic to LUT based architectures

Schedule

- Project Abstract (Due April 5th in class)
  - Start discussing project ideas with the 6.111 staff
  - Abstract should be about 1 page (clearly state the work partition)
- Work in teams of two or three. A single person project requires special approval by me.
- Proposal Conference with TAs (April 7-9). Bring your proposal with you.
- Block diagram conferences with TAs (April 12-16)
  - 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) – by the design presentation time (April 21st).
  - 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 21, 23, 26, 28
  - Each group will make an electronic presentation (power point 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.
- Final project presentations and video taping (May 11/12)
- Final project report (in electronic format, which will be published with permission on the course website) due May 13 by 5PM (no late project check-offs or reports accepted)
- See project information handout 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 larger 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

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)

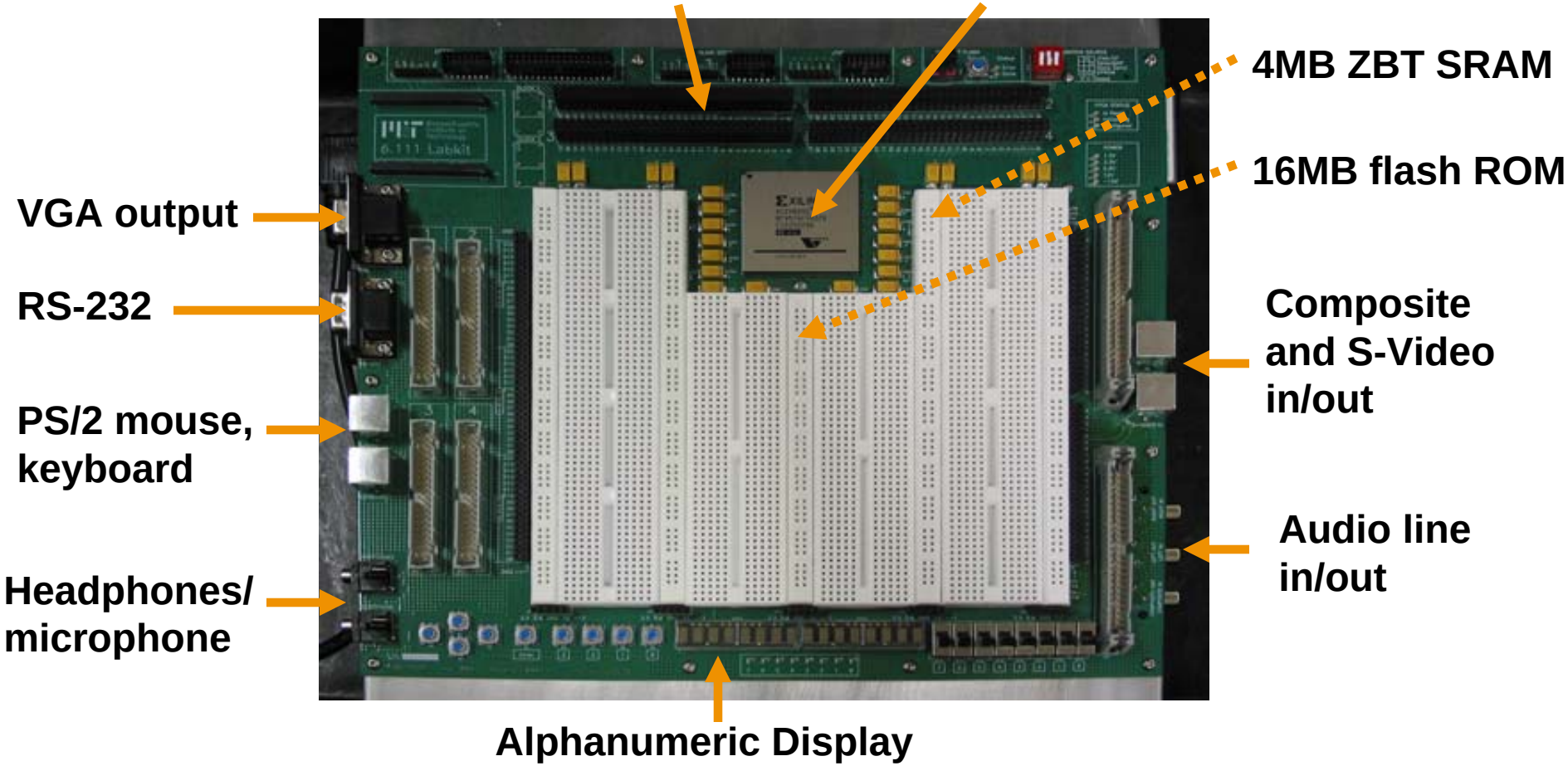
Design Rules

- 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.
- 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.

Design Rules - 2

- 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

New Labkit



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

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