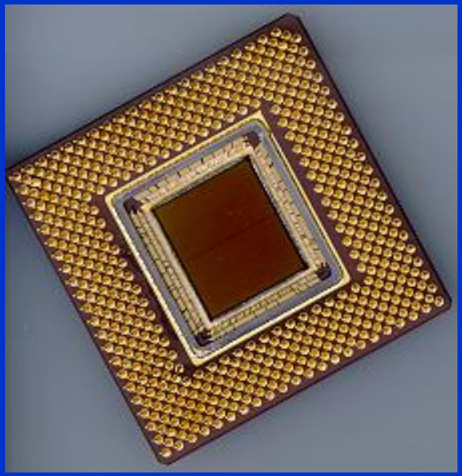


ECE 477 Digital Systems Senior Design Project

Module 1 – Documentation Guidelines and Project Proposal Formulation

Instructional Objectives

- To learn the basic requirements for maintaining lab notebooks
- To review basic principles of engineering design
  - what it is
  - who does it
  - how it is best learned
- To learn the basic requirements for the project proposal

Module 1 Outline

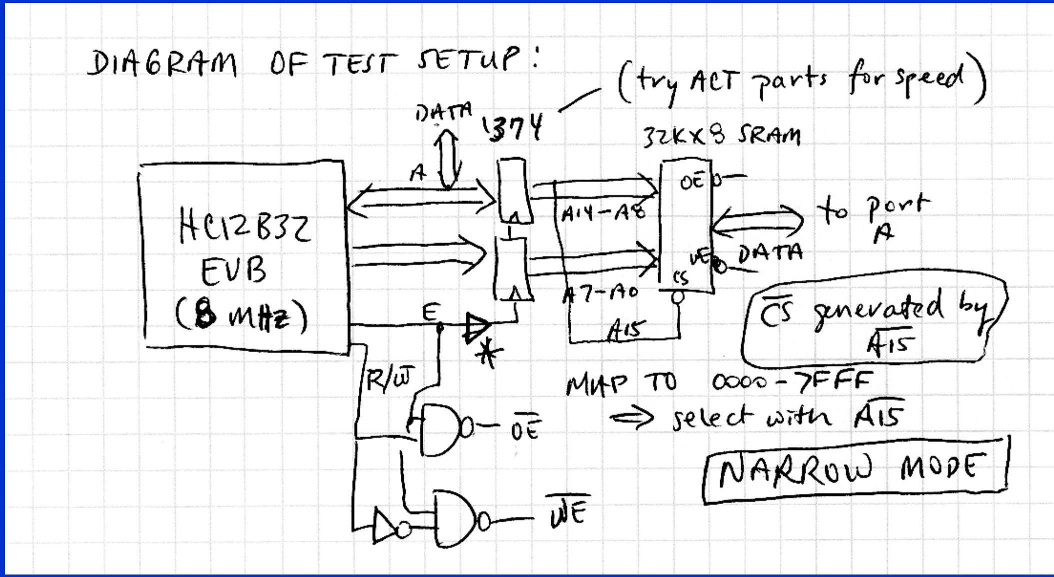
- Lab notebooks
- Software documentation guidelines
- Hardware documentation guidelines
- Engineering design
- Design project proposals
  - Preliminary
  - Final

Lab Notebooks – ***Individual***

- Used to keep track of all observations, modifications, "buggy behavior", etc., for a product under development
- All observations, modifications, redesigns, comments, notes to yourself, etc., should be *documented as they occur*, *NOT* "after the fact"
- Each *individual* will be required to maintain an on-line lab notebook on their team's project webpage
- This is *NOT* a BLOG !!!!

Individual Lab Notebook Entries

Entries will be done in HTML, but may include scanned sheets and digital pictures as well as links to hardware or software source files



Most important thing is to keep it current!

Lab Notebook Evaluation Form

Evaluation:

Component/Criterion	Score	MPY	Points
Level of technical detail – enough detail should be included for others to reproduce your work, and not simply a “blog” (i.e., not be a log of time spent in class or what you ate at a meeting, etc.)	0 1 2 3 4 5	X 3	
Pictures, drawings and diagrams – should be included where appropriate, very large images should be linked (to thumbnails of those images)	0 1 2 3 4 5	X 1	
Update record – daily entries should be made *as work is done* (not days “after the fact”), and should reflect steady, consistent progress	0 1 2 3 4 5	X 2	
Weekly summaries – weekly summaries should be a concise summary of the major accomplishments for the preceding Sunday – Saturday period	0 1 2 3 4 5	X 1	
Technical Writing Style and Clarity – writing style should be professional and concise as well as employ good grammar, sentence structure, etc. – what you wrote should be readily understandable to one “skilled in the art”	0 1 2 3 4 5	X 3	
		TOTAL	**/ 50**

Scoring:

5	*Excellent – among the very best notebooks completed this semester*
4	*Good – all requirements were amply satisfied*
3	*Average – some areas for improvement, but all basic requirements were satisfied*
2	*Below average – some basic requirements were not satisfied*
1	*Poor – very few of the requirements were satisfied*
0	*Unacceptable*

Lab Notebook Evaluation

- Level of technical detail
  - Can we reproduce what you did from what is written in your notebook?
  - No "blogs"
- Appropriateness
  - Is the content *directly* related to the project?
  - No "personal remarks" (what you ate, etc.)
- Pictures, diagrams, etc.
  - Need to aid in understanding work done
  - Thumbnail large images to reduce inline image size (generally do not need high-res images for most lab notebook entries)

Lab Notebook Evaluation

- Update Record
  - must be within 2-3 days of actual work
- Weekly summaries
  - 2-3 sentences highlighting main points of week's work
- Technical writing style
  - use 3rd person
  - future employers may see your notebook
- Use a revision control system (of your choice, e.g., `git`, `svn`, `cvs`, `rcs`) to document your update record

On-Line References – Group

- In addition to individual lab notebooks, each group should maintain on-line hardware and software _documentation_
  - data sheets
  - application notes
  - schematics
  - code listings
- Use a version control system (your choice) to manage updates and track changes

Hardware Documentation

- Include text describing changes made to each revision of the schematic
- Include information about test setups and prototyping breadboards
- Include any “debugging notes” or general observations useful to the hardware portion of the team
- Update schematic as you prototype and *prototype as much as possible of your schematic*

Software Documentation

- Include diagnostic routines developed
  - – note which SW version they work with
- As part of header of each software module, include complete history of modifications

; **Title:** *(a descriptive title of this program)*
; **Version: m.n** *(version number, following specified guidelines)*
; **Filename:** *(name of file)*
; **Author(s):** *(person who has written code)*
; **Purpose/Function of Program:** *(a brief paragraph describing the overall function performed)*
; **How Program is Run on Target System:** *(include starting address, etc.)*
; **Date Started:** *(when you first started to write this program)*
; **Update History:** *(list of changes made since first written, date of each change)*

Engineering Design

- Concerned with finding ways to build an object that performs a prescribed function – “applied intuition”
- Difference between engineering and science
  - evaluate tradeoffs among a variety of alternatives
  - orderly subdivision of problem into parts
  - iterative and recursive techniques based on analysis and synthesis

Engineering Design

- Engineers make decisions based on facts available at time of implementation (e.g., parts availability, customer specification)
- A wise engineer tries to *anticipate* future requirements, allows for future expansion without hindering current design criteria.
- Many choices to be made, both technical and non-technical; another factor is design for manufacturability – i.e., can the design be altered to reduce the cost of manufacturing and testing?

Engineering Design

- Who does it? ("working definition")

A *Computer Engineer* has knowledge of both the *theoretical and practical* aspects of how to *analyze, design, and implement* computer *hardware and software*…

Which leads to an *intuition* of how computer hardware and software works, the *synergy* between them, and what solution(s) are *"most practical" or "best"* given a set of *design constraints*.

They are therefore *able to design systems* based on the *analysis of tradeoffs* among a *variety of different implementation strategies* afforded by current technology.

Engineering Design

- How is it learned?
  - FACT: Very little learning occurs as a result of <u>listening to a lecture</u> about how to solve design problems (although careful note-taking promotes active encoding of information)
  - FACT: Very little learning occurs as a result of <u>reading about</u> how to solve design problems

Engineering Design

- How is it learned?
  - FACT: Very little learning occurs as a result of <u>watching someone else</u> solve design problems
  - FACT: We typically forget 80-90% of what we <u>hear</u>, but remember 80-90% of what we actually <u>do</u>
  - OBSERVATION: We remember 99-100% of what we “screw up”

Engineering Design

- Conclusions:
  - design capability is not "innate" – rather, it develops through lots of *experience* and *failures*
  - *creativity* is a key ingredient
  - the completion of a comprehensive design project (e.g., ECE477) is very beneficial – more realistic if done as a team effort
  - the technical material changes rapidly, but the *basic approaches* to problem solving and system design can serve an engineer well for many years

Preliminary Project Proposal (HW 0)

Team # ____ Proposed Project Name: ____

Team Members (#1 is Team Leader):

#1: ______________________________ Areas of Expertise: _____________________________

#2: ______________________________ Areas of Expertise: _____________________________

#3: ______________________________ Areas of Expertise: _____________________________

#4: ______________________________ Areas of Expertise: _____________________________

Project Description:

Describe your proposed design project idea and the motivation behind it. Summarize the basic functionality of the finished product and the intended application. Estimate the cost of completing your prototype for this course. Indicate how each team member is expected to contribute to the overall project, given their areas of expertise.

Competitive Analysis:

Identify three commercial products, open source projects, and/or patents ("prior art") that are similar to the project you are proposing. Describe the competitive advantages you plan to incorporate in your design relative to the prior art.

Patent Basics

- **Primary purpose of patent is to protect intellectual property:**

Congress shall have the power . . . to promote the progress of science and useful arts, by securing for limited times to authors and inventors the exclusive right to their respective writings and discoveries.

U.S. Constitution - Article 1, Section 8

Patent Basics

- **Patent = Right to Exclude**
  - **Prevent others from making, using or selling your invention**
  - **Right to exclude ≠ Right to make**
    - **May not own all required components**
    - **Need for licensing**
  - **Patent is *not* needed to manufacture a product**
  - **Duration = 20 years from filing date**

Patent Basics

- **Patent requirements:**
  - **Useful**
  - **Novel**
  - **Non-Obvious**

- **Patent contains:**
  - **Written description**
  - **Drawings/schematics**
  - **Claims**

Patent Basics

- **Applying for a patent requires**
  - **extensive search of prior art**
  - **protracted negotiation with assigned patent officer**
  - **patent lawyer**
- **Many on-line patent search engines are available – see www.freepatentsonline.com or google.com/patents**
- **The official web site of the U.S. Patent Office is – www.uspto.gov (printed copies of "old" patents available for a small fee)**

Patent Basics

- Best time to search for possible patent infringement is at product conception stage
- If the possibility of infringement exists, either:
  - <u>eliminate</u> infringing function from your design
  - <u>modify</u> your design so that the infringing function is performed in a "substantially different way"
- Types of infringement
  - literal
  - doctrine of equivalents

Literal Infringement

- **Exactly same function performed exactly the same way**
- **Should be “obvious”**
- **Either eliminate this function from your design -or- obtain license/pay royalty fee to use this function**
- **Note that “simply adding additional features” does <u>not</u> eliminate infringement**

Doctrine of Equivalents

- Substantially same function performed substantially the same way
- Hypothetical examples
  - for fastening pieces of wood together, does a screw perform “substantially the same function” in “substantially the same way” as a nail?
  - for reproducing recorded music, does a vinyl LP perform “substantially the same function” in “substantially the same way” as a CD?
  - for the purpose of recognizing an access code (PIN) , does software running on a microcontroller perform “substantially the same function” in “substantially the same way” as a state machine realized with discrete flip-flops and gates?

Doctrine of Equivalents

- **Need a clear understanding of <u>mechanism</u> ("way") – the <u>mechanism</u> is the *function* of interest in analyzing infringement liability under the doctrine of equivalents**
  - **hypothetical #1: "fastening" is not what is "patented"; rather, the fastening <u>mechanism</u> (ribbing on nail vs. threads on screw)**
  - **hypothetical #2: "music reproduction" is not what is "patented"; rather, the music reproduction <u>mechanism</u> (needle vibration following molded groove vs. optically reading digital data from pits molded into plastic disc)**
  - **hypothetical #3: "recognizing an access code" is not what is "patented"; rather, the access code recognition <u>mechanism</u> (software running on a generic embedded microcontroller vs. discrete custom hardware realizing a state machine)**

Doctrine of Equivalents

- **Stated another way, is the "software" implementation of <u>any</u> function the equivalent of a (digital) hardware implementation of that function under the doctrine of equivalents?**
- **Answer: "hard to say" – this is why there are patent lawyers!**
- **No case to date where software ruled the equivalent of hardware *per se* – but have been cases where <u>functions</u> ("algorithms") of hardware and software devices ruled as performed in "substantially the same way"**

Avoiding Infringement

- **Designing around**
- **Licensing**
  - **Straight license**
  - **Cross-license**
- **Acquiring subject patent**
- **Declaratory judgment action**
- **Ceasing manufacturing**

Final Project Proposal (HW 1)

Team Code Name: ______________________________ **Group No.** ______

Team Members (#1 is Team Leader):

#1: ______________________ **Areas of Expertise:** ______________________

#2: ______________________ **Areas of Expertise:** ______________________

#3: ______________________ **Areas of Expertise:** ______________________

#4: ______________________ **Areas of Expertise:** ______________________

Project Abstract:

Describe what your project is and what it does. Describe the competitive advantages you plan to incorporate in your design relative to the prior art.

Design/Functionality Overview:

Describe your proposed design project and the motivation behind it. Summarize the basic functionality of the finished device and the intended application. Estimate the cost of completing your prototype. Indicate how each team member is expected to contribute to the overall project, given their areas of expertise.

Final Project Proposal (HW 1)

- Basic Project Success Criteria – a list of *landmarks*, <u>common to all projects</u>, that outline a path toward successful completion:
  - create a bill of materials and order/sample all parts needed for the design
  - develop a complete, accurate, readable schematic of the design
  - complete a layout for the printed circuit board used in the project
  - populate and debug the design on said printed circuit board
  - package the finished product and demonstrate its functionality in a stand-alone setting

Final Project Proposal (HW 1)

- Project-Specific Success Criteria – a list of *functionality specifications*, specific to your project, that will be used as the basis for evaluating successful completion of the project requirements
- Examples
  - An ability to send/receive/decode IR signals
  - An ability to lock/unlock a door in response to a card swipe
  - An ability to control speed/direction of a robot
  - An ability to steer robot away from obstacles
  - An ability to remotely control household appliances using DTMF signals sent over a telephone line

Final Project Proposal (HW 1)

Project-Specific Success Criteria:

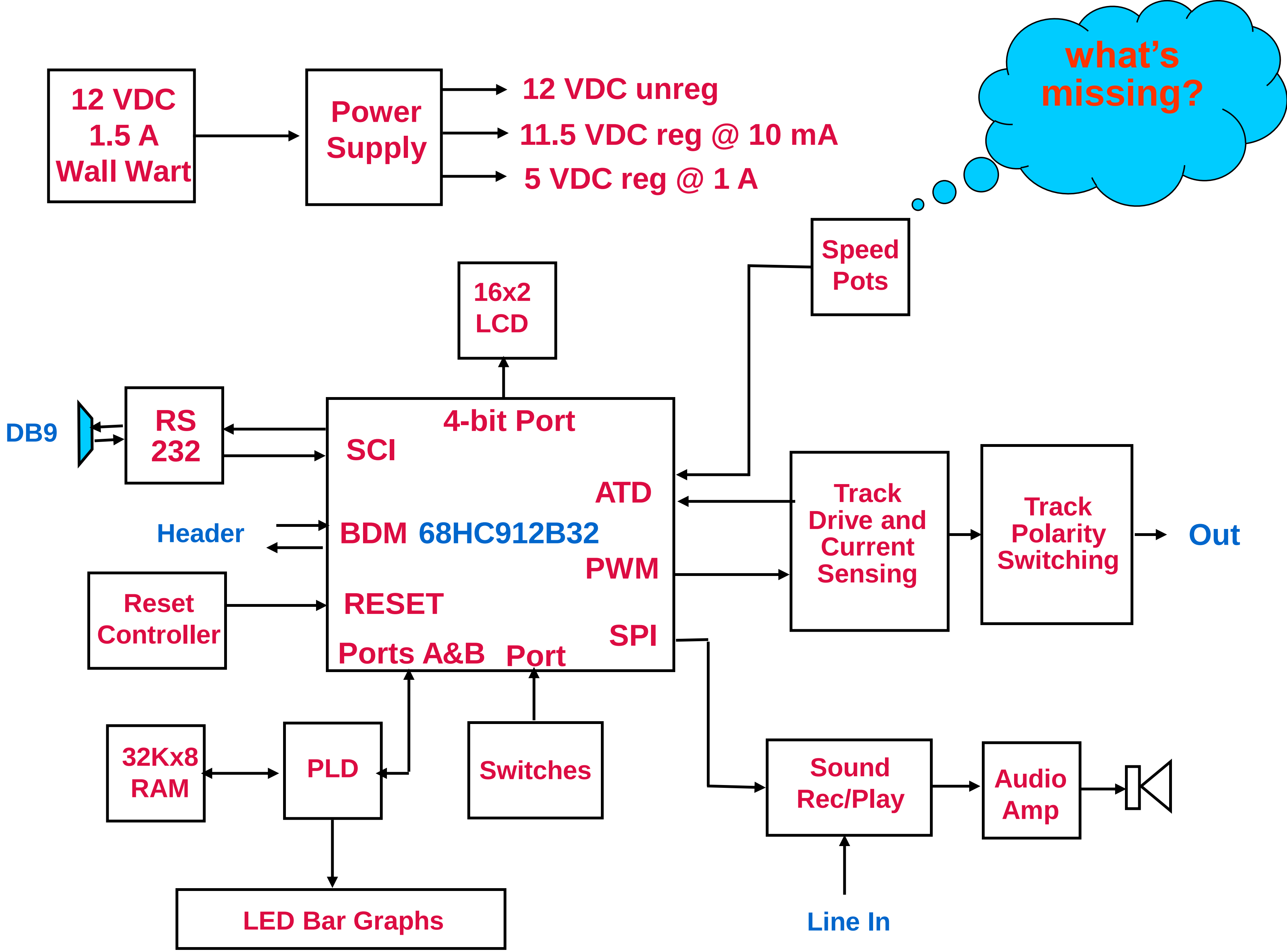
1. An ability to…
2. An ability to…
3. An ability to…
4. An ability to…
5. An ability to…

Block Diagram:

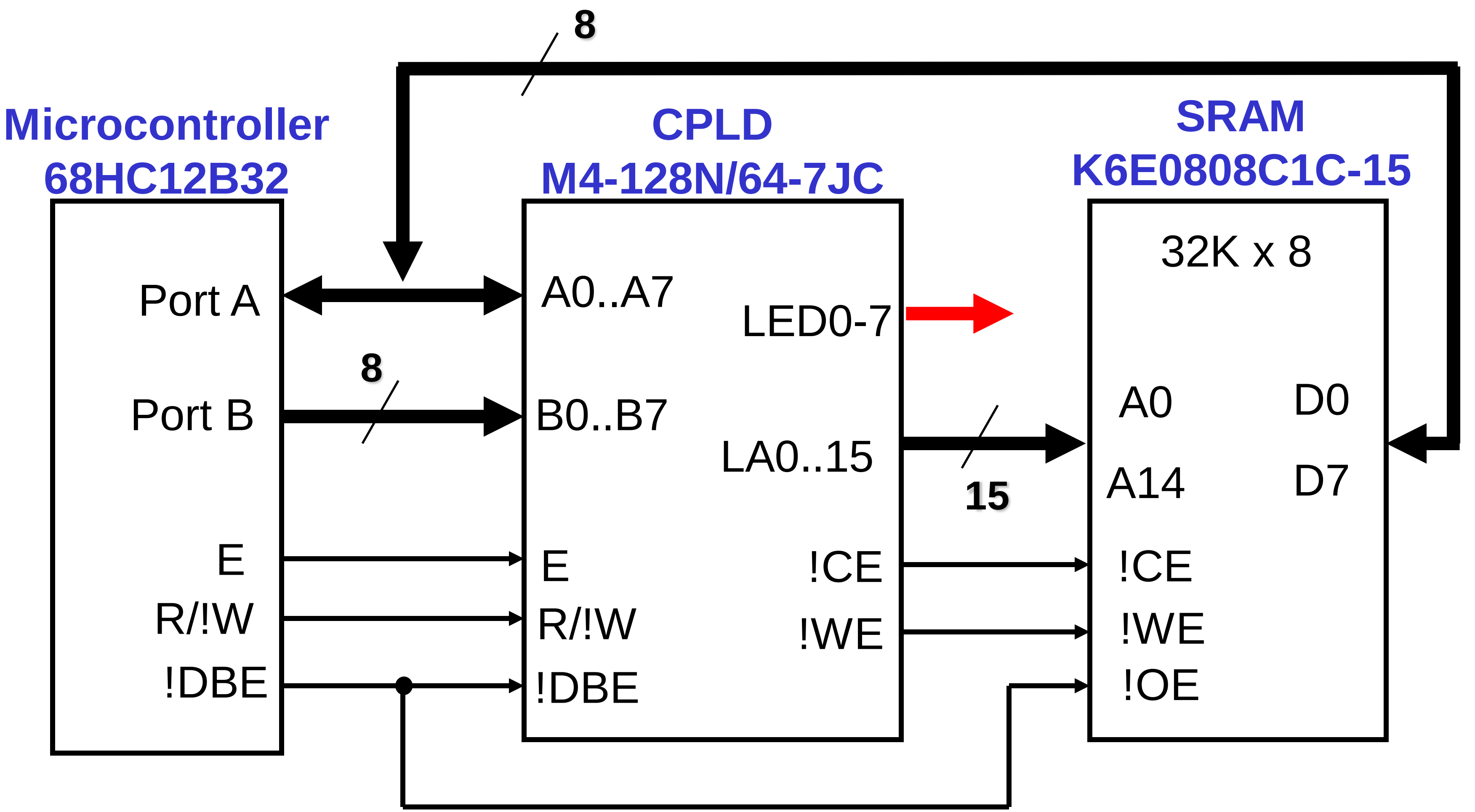
Insert a hardware block diagram here.

System Block Diagram

- A "high level" view of your overall system, depicting the major functional blocks and illustrating their interconnection
- Include bus widths on appropriate connections – very important for constraints analysis
- Continuously "refine" as project progresses – keep it up-to-date, it can be very useful in writing and debugging software



Sample Block Diagram



Interface Between HC12, CPLD, and Samsung SRAM

Final Project Proposal (HW 1)

Division of Labor:

Design Component Homework		***Professional Component Homework***	
Packaging Design and Specs		Design Constraint Analysis/Parts List	
Circuit Schematic and Narrative		Patent Liability Analysis	
Printed Circuit Board Layout		Reliability and Safety Analysis	
Software Narrative and Listing		Social/Political/Environmental Analysis	

__Each__ team member should take responsibility for __one__ Design Component Homework and for __one__ Professional Component Homework – note that these will count toward __individual__ student grades.

IMPORTANT: Examine the Course Calendar and note the due dates for each report. Consider the workload and due dates in your other courses before assuming these assignments.