

L1.1                      Goal                      Wed. February 6, 2002

### Design and Implement Complex Digital Systems.

Use a Hardware Design Language (VHDL).

Implement with Multiple Existing Integrated Circuits.

Prior Digital Design Experience is not Necessary.

Prerequisite is Something to do with Circuit Theory.

6.004 is NOT a Prerequisite.

Take 6.004 before 6.111 or

Take 6.004 after 6.111 or

Take both in the same term.

L 1.2                      Objectives and Outcomes                      Wed. February 6, 2002

On completion of 6.111 students will have confidence in their abilities to conceive and carry out a complex digital systems design project in a team of two or three people. More broadly, they will be ready to handle substantial, challenging design problems. In particular, students will be able to:

1. explain the elements of digital system abstractions such as digital logic, Boolean algebra, flip-flops, finite-state machines (FSMs), and microprogrammed systems.
2. design simple digital systems based on these digital abstractions, and the "digital paradigm" including discrete, sampled information.
3. use basic digital tools and devices such as digital oscilloscopes, PALs, PROMs, and VHDL.
4. work in a design team that can propose, design, successfully implement, and report on a digital circuit design project.
5. communicate the purpose and results of a design project in written and oral presentations.

L 1.3                      Approach                      Wed. February 6, 2002

#### Knowledge

Theory

Examples

Design Rules

Guidelines

#### Environment

Lab Space

Oscilloscopes and Logic Analyzers

Programming equipment, computers, and design software

#### Challenges

Quizzes

Problem Sets

Lab Exercises

Project

L 1.4                      Project                      Wed. February 6, 2002

#### Lab 1:

Find the Digital Lab and wire something.

Learn about equipment: oscilloscopes and logic analyzers.

Program and test a PAL (PLD).

#### Lab 2:

Design and Implement a complicated FSM.

Use VHDL to program a CPLD.

#### Lab3:

Design exercise using microprogramming

(may be replaced with a gate array laboratory)

Good prototype for the final project

L1.5 Final Project Wed. February 6, 2002

Unstructured Assignment  
Unstructured Solution  
You and the staff negotiate a proposal  
Proposal Conference  
Design Review(s)  
Early  
Detailed  
Design Presentation

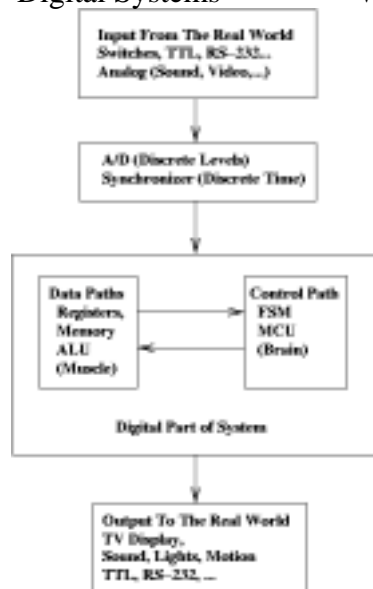
Staff will provide  
Help with design, debugging and testing  
Encouragement  
Praise (as success evolves)

L 1.6 Grading and Collaboration Wed. February 6, 2002  
We start with a number and then discuss everyone, especially performance in labs and project.

Quizzes (20%)  
Problem sets (10%)  
Lab exercises (35%)  
Project (35%)  
As you see, this is primarily a lab course.

Cooperation  
Please be civil and don't hog resources such as computers.  
Do not collaborate with anyone on quizzes.  
Do not copy anything from anyone else.  
  
Collaborate on how to do problem sets and labs,  
then do them individually.  
Do NOT copy anything from anyone else!  
  
Collaboration on the project is welcome.  
Get help from anyone who will help you.  
Cooperate and collaborate with your project partners.  
Copy anything you want (with attribution).

L 1.7 Digital Systems Wed. February 6, 2002



L 1.8 Implementation Wed. February 6, 2002

To implement digital circuits, we  
Start with Gates: AND, OR, NAND, NOR, NOT, etc.

These blocks are implemented by SSI (requires the most wiring).

Progress to Building Blocks: Registers, Counters, Shift Registers,  
Multiplexors, Selectors, etc.

These blocks are implemented by MSI (requires less wiring).

Progress to PALs (PLDs), CPLDs, Gate Arrays.

We need a language (e.g., VHDL) so these blocks can be programmed  
(requires the least wiring).

L 1.9 VHDL Wed. February 6, 2002

## VHSIC Hardware Description Language

Language to express digital systems

Structural  
Behavioral  
Timing

Rich and powerful language  
Basic standard environment  
Supports both

Hardware concepts and  
Software concepts

L 1.10 VHDL Example Wed. February 6, 2002

```
-- Massachusetts (Obsolete) Stoplight Example
library ieee;
use ieee.std_logic_1164.all;
entity check is port(
  r, y, g: in std_logic;
  ok: out std_logic);
end check;
architecture logical of check is
  signal t1, t2, t3: std_logic;
begin
  t1 <= r and (not g); -- doesn't matter whether y is on or off
  t2 <= y and (not g); -- doesn't matter whether r is on or off
  t3 <= (not r) and (not y) and g; -- g on alone
  ok <= t1 or t2 or t3;
end logical;
```



L 1.11 VHDL Characteristics Wed. February 6, 2002

### Advantages

- Shorter Design Cycle
- Improved Design Quality
- Vendor and Technology Independence
- Lower Design Cost
- Design Management
- Simulation and Verification of Design

### Disadvantages

- A Change of Culture
  - Away from Schematic-based Design
  - Towards Language-based Design
- Cost of Getting Started
  - Selecting and Paying for Tools
- Debugging Design Problems

L 1.12 Misconceptions Wed. February 6, 2002

Just code it in VHDL and the synthesis tool will design the logic.

VHDL code that simulates the same way will synthesize to the same set of gates.

The best style is OBJECT ORIENTED.

Depends on the objects (always does)

Synthesis just can't be as good as a design done by a human!

Shades of assembly language vs. a higher level language

L 1.13      What Can be Synthesized      Wed. February 6, 2002

Combinational functions

Mux, encoder, decoders  
Comparator  
Parity generator  
Adder, ALU  
Miscellaneous logic

Counter based functions

Counters  
MAR, FIFO

Register and latch functions

Control logic

Sequencers, FSM  
Synchronizers

L 1.14      Observations      Wed. February 6, 2002

VHDL is a programming language.  
There are many good and bad programs.

Functionality is important.  
BUT not enough! Style is important.

Good design is critical.

Synthesis is hard.  
Fitter program takes clues from your VHDL code.

Decomposition of a large design into smaller, understandable sub-parts is essential.

L 1.15      Design for Testing      Wed. February 6, 2002

Design can be fun.

Testing is work.

Testing is more important than design.

Untested designs are rarely good designs.

Verification by simulation is often the ONLY way parts of a design can be checked. Physical realizations often do not allow access to internal signals.

Almost all of us make mistakes; testing helps catch mistakes.

L 1.16      How To Use VHDL      Wed. February 6, 2002

Design before typing,

at least a little!

Start with a working VHDL file,

modify it to do what you want to do.

Compile files as they are typed in.

Don't wait until design is completely entered before compiling and/or fitting the VHDL code.

Take one step at a time.

L 1.17      Lab Hours – Disk Storage      Wed. February 6, 2002

Please be out by the indicated time.

Instrument room personnel get paid by the hour.

Please do NOT take advantage of them.

The usual times are:

Monday through Friday – 9:00 AM to 11:45 PM

Saturday – 12:00 Noon to 5:45 PM

Sunday – 12:00 Noon to 11:45 PM

Hours for Holidays, Spring Break, etc. will be posted.

**Disk Storage (in addition to your Athena account)**

Unlimited (well, no quota) disk storage is available on sunpal[12]. Get your computer account on sunpal[12] by logging in to sunpal[12] in the Digital Lab, 38–600. Use your Athena name and password. This will automatically give you an account on sunpal[12] and create your home directory. Your sunpal[12] home directory is available on the Ultra5s in the Digital Lab. The sunpal machines are located on the near right side of the lab as you enter the lab. If you have trouble, ASK.

L 1.18                      6.111 Software                      Wed. February 6, 2002

Use 'setup 6.111' not 'add 6.111'

so /mit/6.111/.attachrc is sourced.

From PCs in the Digital Lab (or from any Athena PC)

'setup 6.111'

'ultra5' (to login to the least lightly loaded Ultra 5 in the Digital Lab)

Names are athpal0 – athpal5 and eecs–ath–33.

'setup 6.111' (on the Ultra 5)

L 1.19      Turn in Your Schedule Sheet      Wed. February 6, 2002

Fill in the form on the last page of the handout. **ONLY** your Tuesday schedule is important.

Extra handouts will be in the filing cabinet in the back left corner of the Digital Lab – 38–600.

Recitation Assignments will be posted by Friday on the bulletin board in the Digital Lab and on the 6.111 web page.

//<http://web.mit.edu/6.111/www/s2002>

Also linked to the department web page

Pick up lab kits starting Thursday at 1 pm.