EMT251 - Introduction to Integrated Circuit Design

This course is *Introduction to Integrated Circuit Design*, offered by the School of Microelectronics Engineering.

Academic Session 2009/2010

This semester (semester 2), I will be handling this course for students in the Computer Engineering and Communication Engineering programs.

Announcement

[20100423] **Final Theoretical Session @ DKR4 on Saturday, 24/04/2010 from 10am to 12pm.**

Syllabus

Week 00

- Course Briefing/Introduction
- **No LAB Sessions**

Week 01

- Lecture: History of IC and Transistor Fundamentals
- Lab: Circuit Netlist & Simulation (Mentor's Eldo & Xelga)

Week 02

- Lecture: More Netlist? Parameters?
- Lab: Circuit Schematic (Mentor's Design Architect)

Week 03

- Lecture: Combinational Logic Design
- Lab: Lab Test #1 (Circuit Netlist & Simulation)

Week 04

- Lecture: Combinational Logic Design (Part II)
- Lab: Tutorial #1 (Circuit Netlist & Simulation)
- Recap of last week's lab test:
    - analysis of output waveform

        - many students still have no idea of what we're doing - got the initial waveform, but don't know what to do
    - netlist format
        - the difference between circuit element & netlist command
    - troubleshooting & debugging
        - READ THE ERROR MESSAGE!
    - nodes & connectivity
        - connect the dots!
    - using pwl instead of pulse
        - is a cycle enough? why should we use pulse? when should we use pwl?

Week 05

- Lecture: Dynamic Logic (Maybe will add Sequential Logic Design)
- Lab: Briefing & Start of Mini-Project (The TWO Tasks)
- ***Note:*** *We will be moving our lab sessions to MKR3 starting this week! Yay!*
- **Test 1 (Theory)**: Sunday, 7 February 2010 @ DKG2/3 @ 3PM! GOOD LUCK!

Week 06

- Lecture: CMOS Fabrication, Half-Semester Summary
- Lab: Custom IC Layout (Using IC Station)

Week 07

- SEMESTER BREAK!

Week 08

- Lecture: The first 2-hour is cancelled (VC's keynote speech), so maybe a tutorial?
- Lab: No official lab hour (free access)

Week 09

- Lecture: Layout (Stick diagrams, Design rules, Design area)
- Lab: Mini Project

Week 10

- Lecture: Circuit Characterization
- Lab: DRC & LVS

Week 11

- Lecture: Circuit Characterization (more)
- Lab: Mini Project

Week 12

- Lecture: Performance Estimation
- Lab: Lab Test #2 (DRC & LVS)

Week 13

- Lecture: Performance Estimation (more) + Tutorials
- Lab: Mini Project

Week 14

- Lecture: Design Methodology + Tutorials
- Lab: Viva for Mini Project

Timetable

20091130 Okay... the timetable is out. Will publish it here once confirmed. ***20091207*** Final draft? ***20091215*** Final? ***20100104*** Really... Final One?

Lecture Hours

Group	Location	Day	Time
(3,4)	DKR2	WED	1000-1200
(3,4)	DKR5	THU	1000-1100

Lab Hours

Group	Location	Day	Time
(3)	MMIC	WED	1500-1700
(4)	MMIC	FRI	0800-1000
(4)	MMIC	FRI	1000-1200
(3)	MMIC	FRI	1500-1700

Consultation Hours

Group	Location	Day	Time
($)	SME-B10	MON	1500-1600
($)	SME-B10	TUE	1000-1200
($)	SME-B10	TUE	1500-1600
($)	SME-B10	THU	1500-1600

Note The 2-hour slot on Tuesday morning is by appointment only.

Reference

1. Weste, Neil and Harris, David. "*CMOS VLSI Design: A Circuits and Systems Perspective*". 3rd Edition, 2005. Addison Wesley. ISBN: 0-321-26977-2.

EMT251 @ UniMAP

Ramblings of an old school academics... at least that's how I look at myself (though I'm not THAT old

).

20091020 - Old School

I have been assigned to EMT251 for the coming semester. This course (*EMT251 - Introduction to IC Design*) is actually a course offered by School of Microelectronics. Looking at my background, it's only natural for me to get assigned as one of the lecturers for this course. However, I have since deviate my focus towards board level designs instead of IC level. So, I'm going to need to brush up my IC design knowledge - which I guess would be slightly difficult with the technology advances. For example, I used to work with 2µ and 1.2µ processes which was still using the good old lambda rule and where we only discuss about long channel effects  We'll see how it goes...

20091023 - EMT251 Mini Project

Okay... it seems that no details on EMT251 will be coming anytime soon. We will be having a meeting on the content of this course. In the meantime, I just had an idea about a mini-project for this course: develop a simple program (can be any language, but C comes to mind) that reads a circuit netlist and then create a basic layout and routing for that circuit. Let's say we start with a circuit with around 100 logic gates? I haven't done this before myself but I'm excited about it already. Hope we can work something out to make this happen (although I don't see it ever going to happen) 

20091125 - Lab-oriented Engineering Education

I've been thinking... the original idea behind this lab-oriented (or lab-intensive) approach being implemented in UniMAP was having lab sessions that could strengthen the theoretical knowledge being taught during lectures. However, as I have come to realized, some courses (maybe all) tend to grade the students based on the practical lab-work (i.e. get marks for completion of assignment) rather than the understanding of the theory itself. So, if this is the case, we shouldn't be surprised if the grades during final exams do not reflect the coursework marks (mainly lab-work). This course has a similar trend... the assignment during lab hours are based on the usage of tools, rather than design knowledge/analysis. Obviously, the lab-work **will not** (most of the time) be what we planned for in the first place. So, I'm planning to suggest that we induce more theoretical-related component into the lab. Still waiting for that meeting...

Mentor Graphics Tools @MKR3

The thing to do: I need to setup another IC design lab - going to EPF building in Kangar has become too much of a hassle (i.e. parking madness!).

The issue: The Mentor Graphics software that we have is the old version 2005.x (icflow, ams, ixcal, etc.) - which requires either the old Red Hat 8 or RHEL 3. Due to its dependency on the old shared libraries (unlike the Xilinx Tools I've been using), it simply refuses to run on the newer Linux distro (e.g. I've tried it on Slackware 13.0, naturally). In addition to that, I've been having problems installing RHEL3 on the machine @MKR3 (HP Compaq DC7800). It lost both its USB mouse and keyboard when rebooting from installation. Note that these peripherals worked during installation.

The solution: After finishing installation, boot using the rescue facility on the install CD (i.e. type `linux rescue` at boot prompt). Edit `/etc/modules.conf` and change:

```
alias usb-controller usb-uhci
alias usb-controller1 ehci-hcd
```

to:

```
alias usb-controller ehci-hcd
alias usb-controller1 usb-uhci
```

More issue: Now I have another problem. RHEL3 simply CANNOT detect the ethernet controller.

The solution: Download the e1000e driver. I used the older version 0.2.9.5 - simply because I know it will work (based on a forum discussion that I came across... and the above link).

Miscellaneous Issues:

- how to let user automount usb?
  - as root,

    ```
    echo "/dev/sda1  /mnt/usbdrive  vfat  noauto,owner,user  0 0" >> /etc/fstab; mkdir /mnt/usbdrive
    ```

  - i made 4 entries (inc. sdb1,sdc1,sdd1) just in case
- NIS login
  - refer to this
  - somehow the ypserv package is not on the install dvd - got this from centos 3.4 mirror
- dma is NOT enabled for SATA harddisk? (or is it a motherboard issue?)
  - append `ide0=ata66` to kernel? ⇐ doesn't work!
  - try `idebus=66` next!

Mentor EDA Installation:

Installation of Mentor Graphics tools was straight forward. I installed everything in `/opt/mentor` rather than `/EDA` set by our vendor. So, I get to learn about setting up the environment variable a bit.

Things to note:

- make sure `/etc/hosts` is present and contains line for localhost - Mentor Graphics software somehow rely on this to be correct to run!

EMT251 @MKR3

Finally, MKR3 is now officially capable of running Mentor Graphics EDA tools. In addition to that, I also managed to configure NIS server/client setup for domain login. The students are no longer tied-up to a single computer. However, I've just found out that I should have considered LDAP instead for better security features. Since this is a local subnet, I guess it should be safe for at least this semester. I'll try to find out about LDAP next.

So, just for archiving purposes, here's how I setup the NIS domain login. This is done on RHEL3 UPD4 - other distributions/releases might do things slightly different.

- To configure NIS server, check required packages
  - yp-tools, ypbind and ypserv (ypserv NOT on install DVD I created - had to get it from CentOS 3.4 mirror)
  - decide on a NIS domain name <nis-domain>, and use a static IP address <nis-ip> for the intended NIS server machine
- Customize server settings
  - most of the default settings should be usable (may not be secure, but usable)
  - set domainname

    ```
    domainname <nis-domain>
    ```

  - edit `/var/yp/Makefile` accordingly and run

    ```
    /usr/lib/yp/ypinit -m
    ```

    - somehow mail doesn't build - I simply disabled them in the Makefile (from all rules)
  - I didn't change '/etc/nsswitch.conf', `/etc/ypserv.conf` and `securenets` (using defaults)
- Start the services
  - ypserv, yppasswdd, ypbind (ypxfrd is for slave servers)
  - I've tried this without ypbind running and it still works! so, I guess it's not really needed
- To configure NIS client, check required packages
  - yp-tools and ypbind
- Set NIS domain & client settings
  - set domainname

    ```
    domainname <nis-domain>
    ```

  - run

    ```
    authconfig
    ```

    or 'Authentication' GUI interface

    - this is to setup NIS client (you need <nis-domain> and <nis-ip> here)
- Start the service
  - ypbind

Also, it's common for NIS setup to utilize NFS - no point of being able to login from any machine without having a 'portable' folder. It's relatively simple to setup NFS on RHEL3 UPD4 - other distributions/releases might do things slightly different.

- on server machine, edit `/etc/exports` and add an entry

  ```
  /home 192.168.20.0/24(rw,sync)
  ```

  - create the same name for the mount point on clients - will make your life easier
  - start the nfs server (can be done from command line OR the 'services' GUI interface)
- on client machine, edit `/etc/fstab` and add an entry

```
<server-ip>:/home  /home  nfs  user,rw  0 0
```

- naturally, we need to make sure /home directory is not filled and reserved for NFS mount
- either restart the machine or as root, type

```
mount /home
```

- we can alternatively use autofs, but I thought this is much easier

NIS on Slackware

Got this from here.

```
Slackware-NIS-mini-HOWTO
How to set up a NIS server and configure NIS clients on Slackware Linux
14-Dec-2000
David Cantrell <david@slackware.com


==============
THE NIS SERVER
==============

STEP 1:  Decide which machine will be the NIS server
----------------------------------------------------

This is usually not a very involving process, but does require some
thought.  You'll want to use a machine that is on most of the time and
does not need to reboot into other operating systems.  If you have an
existing fileserver, that would be a good choice for your NIS server.


STEP 2:  Decide on what you want NIS to serve
---------------------------------------------

NIS is fairly complex.  It allows you to distribute all kinds of
information.  Things like usernames, passwords, groups, hostnames, and
network services that should be running.  The most common use is to use
NIS as a common user logon authetication system.  The second most common
use is to have it share NFS automount maps so that users logging in on a
system will have their home directory automounted for them.

In this mini-HOWTO, I will explain how to set up your NIS server and
clients for passwd/group sharing and home directory sharing.


STEP 3:  Move home directories to sharing location
--------------------------------------------------
```

In order to make the home directory NFS shares work seamlessly across the NIS server and clients, I move them to /export/home. This is the location they will be mounted on the client, so I make them exist there on the server as well. You might not have /export on your server, so let's make it:

```
   mkdir -p /export
```

Now make the home directory location:

```
   mkdir /export/home
```

Now move all your home directories over to /export/home. You should leave "ftp" in /home, as that's where Slackware installs it and it is sometimes site specific. These commands should move all home directories to /export/home and leave ftp in /home:

```
   mv /home/* /export/home
   mv /export/home/ftp /home
```

STEP 4: Edit home directory paths on the server

Since home directories are now in /export/home, you need to edit /etc/passwd on the server to reflect those locations. If you have a user called "david", you will want to change his home directory to /export/home/david. Do this for all users on the system.

STEP 5: Write the automount maps

I usually make two automount maps for the home directories. One is called auto.master and controls how the automounter works. The other is called auto.export and will describe the /export map.

Edit /etc/auto.master and add this to it:

```
   #
   # auto.master - Master map file for NFS automounter
   #

   /export auto.export --timeout 60
```

Now edit /etc/auto.export and add this to it:

```
   #
```

```
   # auto.export - NFS automounter configuration file
   #

   # Home directories
   home -fstype=nfs <NFS/NIS server>:/export/home
```

Of course, replace <NFS/NIS server> with the hostname of your NFS/NIS server.

STEP 6: Edit the NFS exports file on the server

Now we need to make sure that /export is shared from the server. Edit /etc/exports and make sure this line is present:

```
   /export (rw,no_root_squash)
```

Save the file and restart the NFS server daemons:

```
   killall -HUP rpc.nfsd rpc.mountd
```

STEP 7: Pick a name for your NIS domain

Each NIS domain needs to have a unique name. Do not use the same name as your DNS domain. This should be something different. I will use "mynis" in this example.

Edit /etc/defaultdomain and add the name of your NIS domain:

```
   mynis
```

Save and exit. The name of your NIS domain shoulw be the only thing in that file. Once you've done that, set the NIS domain with this command:

```
   nisdomainname `cat /etc/defaultdomain`
```

STEP 8: Edit /var/yp/Makefile

The NIS server has a Makefile that controls what maps are served out by NIS. Each time you make a change (such as adding a user), you should go into /var/yp and type "make" to update the NIS information.

```
In this file we need to edit the all: target and specify the stuff we want
to share.  Here's what your all: target should look like:

   all: passwd group hosts rpc services netid protocols netgrp mail \
        shadow auto.master auto.export

Now find the line that looks like this:

   AUTO_LOCAL  = $(YPSRCDIR)/auto.local

We want to add this line just after that line:

   AUTO_EXPORT = $(YPSRCDIR)/auto.export

Now find the auto.local: target.  We need to add this target just after
that target:

   auto.export: $(AUTO_EXPORT) $(YPDIR)/Makefile
    @echo "Updating $@..."
      -@sed -e "/^#/d" -e s/#.*$$// $(AUTO_EXPORT) | $(DBLOAD) \
            -i $(AUTO_EXPORT) -o $(YPMAPDIR)/$@ - $@
      -@$(NOPUSH) || $(YPPUSH) -d $(DOMAIN) $@

Save the Makefile and exit.


STEP 9:  Edit /var/yp/securenets
--------------------------------

Edit /var/yp/securenets and change the "0.0.0.0   0.0.0.0" line to
something like this:

   255.255.255.0   192.168.1.0

You should specify your network address and submask there, this is only
an example.  This example restricts NIS access to any machine on the
192.168.1 class C subnet.


STEP 10:  Create an /etc/ypserv.conf file
-----------------------------------------

The only line you need in this file is:

   dns: yes

This line tells ypserv to resolve hostnames using a DNS server instead of
relying on NIS hostname maps (which you probably aren't using).  Save and
exit.
```

```
STEP 11:  Start ypbind
----------------------

Start ypbind:

   /usr/sbin/ypbind

Use ps to make sure it started and is running properly.  You will probably
see a master and slave process for it.


STEP 12:  Start ypserv
----------------------

Start ypserv:

   /usr/sbin/ypserv

Use ps to make sure it started and is running properly.


STEP 13:  Initialize the NIS maps
---------------------------------

We need to initialize the NIS maps with this command:

   /usr/lib/yp/ypinit -m

Fill in the blanks for all machines on your network that will be acting as
NIS servers.  At this point, you probably want to fire up rpc.yppasswdd to
make sure that users on NIS clients can run chfn and passwd.


STEP 14:  Done, making sure NIS stuff loads at bootup
-----------------------------------------------------

Your NIS server is now up and running.  You should edit /etc/rc.d/rc.inet2
and make sure the following block is uncommented:

   # Setting up NIS:
   # (NOTE: For detailed information about setting up NIS, see the
documentation
   # in /usr/doc/yp-tools, /usr/doc/ypbind, and /usr/doc/ypserv)
   #
   # First, we must set the NIS domainname.  NOTE: this is not
   # necessarily the same as your DNS domainname, set in
   # /etc/resolv.conf!  The NIS domainname is the name of a domain
   # served by your NIS server.

   if [ -r /etc/defaultdomain ]; then
```

```
     nisdomainname `cat /etc/defaultdomain`
   fi

   # Then, we start up ypbind.  It will use broadcast to find a server.

   if [ -d /var/yp ] ; then
     echo -n " ypbind"
     ${NET}/ypbind
   fi

   # If you are the NIS master server for the NIS domain, then
   # you must run rpc.yppasswdd, which is the RPC server that
   # lets users change their passwords.

   if [ -x ${NET}/rpc.yppasswdd ]; then
     echo -n " yppasswdd"
     ${NET}/rpc.yppasswdd
   fi

It's probably adviseable to run rpc.yppasswdd with this line:

   ${NET}/rpc.yppasswdd -e chsh -e chfn

This ensures that chfn and chsh work.  You also need to add this after the
ypbind if block:

   if [ -x ${NET}/ypserv ]; then
     echo -n " ypserv"
     ${NET}/ypserv
   fi

The ypserv program needs to run on the NIS server.



==============
THE NIS CLIENT
==============

STEP 1:  Set the NIS domain
---------------------------

Edit /etc/defaultdomain just like you did on the server and enter the name
of your NIS domain.


STEP 2:  Edit /etc/yp.conf
--------------------------

Open /etc/yp.conf in a text editor and make sure this line is present:
```

```
   ypserver <IP or hostname of the NIS server>

Specify either the IP address or hostname of your NIS server.  If you
don't have working hostnames, you must use the IP address here.


STEP 3:  Set the NIS domain
---------------------------

Set the NIS domain by typing this command:

   nisdomainname `cat /etc/defaultdomain`


STEP 4:  Run ypbind
-------------------

Start ypbind with this command:

   /usr/sbin/ypbind


STEP 5:  Test the NIS connection
--------------------------------

See if the NIS server is working with this command:

   ypcat passwd.byname

If you see the /etc/passwd entries from the NIS server, you're good to go.
Otherwise, go back to the NIS server section and see if you missed any
steps.


STEP 6:  Edit /etc/nsswitch.conf
--------------------------------

You need to edit /etc/nsswitch.conf and tell it to use NIS for passwd,
shadow, and group.  Change those entries to:

   db files nis

Change the automount entry to:

   files nis


STEP 7:  Enable NIS logins in /etc/passwd and /etc/group
--------------------------------------------------------

Edit /etc/passwd and add this line to the end:
```

```
   +:::::

Edit /etc/group and add this line to the end:

   +:::


STEP 8:  Create the /export directory for home directory mounting
-----------------------------------------------------------------

Make sure you have that directory:

   mkdir -p /export


STEP 9:  Create an rc.autofs script
-----------------------------------

The included rc.autofs script should be moved into /etc/rc.d.  Edit
rc.local and add this line:

   /etc/rc.d/rc.autofs


STEP 10:  Done, enable NIS stuff at bootup
------------------------------------------

Your NIS client is now configured.  Edit /etc/rc.d/rc.inet2 and uncomment
this block:

   # Setting up NIS:
   # (NOTE: For detailed information about setting up NIS, see the
documentation
   # in /usr/doc/yp-tools, /usr/doc/ypbind, and /usr/doc/ypserv)
   #
   # First, we must set the NIS domainname.  NOTE: this is not
   # necessarily the same as your DNS domainname, set in
   # /etc/resolv.conf!  The NIS domainname is the name of a domain
   # served by your NIS server.

   if [ -r /etc/defaultdomain ]; then
     nisdomainname `cat /etc/defaultdomain`
   fi

   # Then, we start up ypbind.  It will use broadcast to find a server.

   if [ -d /var/yp ] ; then
     echo -n " ypbind"
     ${NET}/ypbind
   fi
```

From:
https://www.azman.my1matrix.cc/ - **Azman @MY1**

Permanent link:
https://www.azman.my1matrix.cc/doku.php?id=archive:emt251&rev=1770522381

Last update: **2026/02/08 03:46**