EE 241 Digital Design Spring, 2018

Texts: John F. Wakerly, Digital Design, Principles & Practices, 4th ed., Prentice Hall, 2006, or later edition (required); Capilano Computing, Logicworks, 4th or 5th ed. (recommended)

Scheduled class times: MWF 12-12:50 in SLC222

Lab: F 10-11:50 (L1) or 2-3:50 (L2) in SLC238

Instructor: John B. Gilmer Jr. Office hours: TBD. Office: SLC220 Phone: 4885
<http://www.jbgilmer.com/EE241/EE241.htm >

Bulletin description:

EE-241. Digital Design

Credits: 3

The electronics of digital devices, including Bipolar TTL and CMOS, digital logic functions (e.g., AND, OR, INVERT), Boolean algebra, combinational logic, minimization techniques, digital storage devices, synchronous sequential design, state machines, programmable logic. Three one-hour lectures and one two-hour lab per week.

Background:

The technology of computers is a hierarchy of disciplines ranging from solid state physics up to application dependent fields such as operations research and simulation. Each rests on the field below it. Electronics depends on solid state physics, in that solid state physics defines device behaviors, but these are abstracted into simplified models such as ideal components in the field of electronics. In like manner, Digital Design concerns fundamental electronic circuits called gates that perform logical operations, allowing most electronics considerations to be abstracted away. In logic circuits we will pass ones and zeros rather than voltages and currents. Digital Design also concerns itself with using these gates to perform more complicated logical functions such as addition, data storage, counting, and such. Above Digital Design in the hierarchy, Computer Organization is concerned with how to organize these logical components to build a machine capable of computation, that will execute instructions at the microcode or assembly language level. Yet higher on the technology totem pole are the computer language and systems technologies and applications that concern computer science as well as computer engineering.

EE241 Digital Design concerns the design and use of these electronic circuits, gates and latches, that are the basis for digital machines. The design of digital machines using these elementary circuits to perform more complex functions is the focus of this course, but we will also study the electronics of the components that make up these gates. Theoretical aspects such as Boolean logic and finite state machines will be covered, as well as practical aspects of using available logic devices. We will use simulation tools to design and test circuits, and will consider programmable technologies for more efficient implementation than with discrete gates and latches.

There is no specific course prerequisite. (That is about to be changed; in the future EE283 will be a prerequisite.) Students will be learning some practical knowledge of electrical and electronic principles, so some knowledge of physics and electricity is necessary. The laboratory component is an integrated part of the Digital Design course. Laboratory exercises planned will complement the instruction in the lecture. Sometimes a topic will be first raised in the lab, at other times in the classroom. (The separate EE298T T: Digital Electronics, for 1 credit, is a supplemental special course to cover the digital electronics content in EE241 that is missing in most Digital design courses that students attempt to transfer in from other schools.)

Course Outcomes:

1. Be able to perform binary number conversions and arithmetic.
2. Be able to design and analyze electronic digital gate circuits.
3. Be able to analyze and design combinational circuits, utilizing Boolean algebra and related techniques, and report on design development in a formal report.
4. Be able to analyze and design circuits with components at the MSI level of complexity.
5. Be able to analyze and design synchronous sequential circuits.
6. Gain exposure to issues related to more complex sequential design.

Education, not Training:

It is important to understand what this course is and is not. It is not training to make you a completely ready-to-go digital design expert, ready to go to industry and be instantly productive. In industry, you would be mostly programming in Verilog (a Hardware Description Language), or writing specifications, or designing fault analysis patterns, or any of many tasks that require very specific training beyond what this course can cover. Rather, the intent is to give you a good understanding of the principles upon which all digital technology is based. We use old 1970's era technology in the lab, the 74xx logic devices, because they are simple and robust enough so that we can use them in the lab and gain a practical understanding of what this subject is about. One of the difficulties with modern technology is it is so often inaccessible. You can't get inside to measure or observe, or make modifications. You probably cannot even get schematics or other design details much of the time. We can with this old stuff. The "hands-on" experience you get here will inform your insights and judgement, as you increasingly find yourself working with more abstract tools like simulation and hardware design tools later.

Schedule:

Week	Date	Topics covered	Reading *	Tests
1	Jan 16-19*	Overview, Numbers and binary arithmetic	Chapter 1,2.1-6+	
2	Jan 22-26	The electronics of Digital Gates	Chapter 3.1-11	
3	Jan 29-Feb 2	The electronics of gates, mapping to binary logic	continued	
4	Feb 5-9	Combinational logic design principles	Sections 4.1-4.2	
5	Feb 12-16	Karnough maps, minimization, hazards	Sections 4.3, 4.5	test #1
6	Feb 19-23	Intro to Design Languages: ABEL, VHDL	Sections 4.6-4.7,5.3	
7	Feb26-Mar 2	Doing it right: Documentation, timing	Section 5.1-5.2	
8	Mar 12-16	More complex functions, Combinational MSI	Sections 5.4-5.10,6.1	
9	Mar 19-23	Synchronous design: Latches, clocking, analysis	Sections 7.1-3	test#2
10	Mar 26-28*	Synchronous design: Synthesis, Design	Sections 7.4-7.7	
11	Apr 2-7	Synchronous design: Decomp'n, Programming	Sections 7.8, 7.11	
12	Apr 10-14	Sequential programmable logic, documentation	Sections 8.1-8.3	
13	Apr 16-20	Sequential functions and MSI	Sections 8.4-8.9	test#3
14	Apr 23-27	State Machine design and Memory	Chapter 9, 10	
15	Apr 30	Review	notes	
		Final Examination	(all material)	exam

* Dates marked "*" are short weeks.

* The chapter assignments, based on Wakerly 4th edition, may not be correct for later editions.

Laboratory sessions:

The laboratory work will be performed in the Laboratory SLC 238. Note that in this course lab exercises are individual. You do not work in partnerships, although another student will share the lab station and the use of the laboratory equipment. The lab sessions are relatively informal, with students expected to work independently to accomplish their lab assignments. In a very few cases instruments may need to be shared. For the exercise using the logic analyzers, students need to work in teams due to equipment limitations. (We will probably be using the National Instruments "Virtual Instrument" logic analyzers, which are currently in SLC224.) The instructor will be available to open the laboratory during scheduled office hours. If you are using the lab outside normal hours, please be considerate of others who may be in the lab, and leave the lab at least as clean as you found it. There will probably be some additional scheduled "open lab" times. SLC238 has not been used for EE241 for quite a while. The equipment will be new to you, but it should not be difficult to familiarize yourself with the oscilloscope (4 channels, and color!) and the power supply (be sure to set the "6V" supply to 5V!). The meters are similar to those in SLC125. We will probably not need the signal generators. The SLC238 lab provides more space than SLC224, which is very helpful.

We will be starting our laboratory work at the component level building blocks: diodes and transistors. By assigning high and low voltage levels symbolic values, most commonly 0 and 1, we can represent logic values, and build electronic circuits to do elementary functions such as the "AND" of signals A and B. Collections of signals can represent numbers using the binary arithmetic system. During this lab exercise we will explore how these simple circuits can be combined in useful ways. Ultimately, we will be building a "data path," a data handling unit that is an important step toward the general purpose digital computer. This has in the past been a scrolling display that presents a message chosen by the student. As a final exercise, we will likely build a sequential controller. It is expected that the sequential controller will be done using Field Programmable Gate Arrays (FPGA's), programmed using the Altera "Quartus 2" Integrated Design Environment (IDE) and most likely the "Verilog" hardware description language.

Scheduled Lab exercises:		**Report**
1. Jan 19	Lab orientation, Diodes, transistors, and simple gates	none
2. Jan 26	Design a gate, and characterize it; Device characteristics	Informal#2(same day)
3. Feb 2	Mystery chips: what are they?	Informal#3(same day)
4. Feb 9	Design, simulate, build, demonstrate a 7 segment decoder	
Feb 16	demonstration of above (& start on #5)	Demonstrate #4
5. Feb 23	Busses and switching logic: multiplexed display	**Formal** report #4
Mar 2	continuation of above	Demonstration #5
6. Mar 16	Logic Analyzer used to examine multiplexed display	Informal (next week)
7. Mar 23	Programmable logic used to implement 7 segment decoder	Informal (next week)
8. Apr 7	Data path to build a scrolling multiplexed display	Informal (next week)
9. Apr 14	Demonstrate scrolling display; Sequential Controller	(Informal #7/Demo)
Apr 20	Continue Sequential Controller (demo Apr 28)	Informal (next week)
Apr 27	(Make-up day, possibly review, or demo extras)	

Students will need to be prepared with a design prior to arriving in the lab for a given assignment. Do not expect to arrive at the lab and only then start on the design. That wastes valuable lab time, which should be spent primarily on debugging and demonstrating your circuit. (This is especially important for Lab #2 and Lab #3, which have reports due at the end of the session. A lot of credit is lost by students not completing the assignment! You need to be ready and prepared.) If the lab exercise requires that you come with prepared material (such as an initial design or specification), and you do not have it, you may be excluded from the laboratory.

Students are expected to come to the laboratory sessions bringing with them whatever is needed to do the exercise including a voltmeter (such as the DMM from EE283) and parts kits. You should have obtained an inexpensive voltmeter as part of the Lab Kit for EE283 Measurement Lab, as well as a variety of parts that will be used in this class as well as perhaps Mechatronics. (If you don't have what is needed, we'll be sure you get it.) Other parts will be distributed as needed. The full kit might not be ready for distribution at the first lab, but the parts will be supplied as soon as possible. If you have an extra Voltmeter, bring it, since having two will help, especially with Lab #2. The initial lab exercises are appended to this syllabus.

While most of the components and literature needed will be available in the laboratory or in the library, there are a few items which the student may want to purchase. These include a knife. (An Exacto knife with sharply pointed diagonal blade seems to work best. This helps dig broken wires out of your breadboard when necessary, and can be helpful for stripping insulation.) You may decide that you would like a soldering iron; it is possible to obtain one for about $25 or less. A 25 watt iron is probably best; don't try using 250 Watt guns or propane torches to solder IC's. Temperature controlled irons cost more but give better results. If you buy solder, be sure it is the resin core type, with small diameter, rather than the heavy acid core stuff used to patch radiators.

Grading:

Tests will cover all material through the previous week. Tests will normally be on Friday of the week listed, unless an announcement is made setting the date differently. Tests are open book, open notes, use of calculators is permitted. They are hard. You will not have time to look stuff up in your book very much, and still complete the test. Be well prepared. In addition to the tests listed, there will be up to three pop quizzes. These will generally be given on the day or the day after a homework exercise is collected. (If no pop quiz is given, that proportion of the grade will be allocated to the tests.)

Grading Allocation:

3 tests at 14% each :	42%	6 informal reports/demos at 3%	18%
Up to 3 pop quizzes totaling	6%	Formal lab report (#4)	6%
final examination:	25%	Class Participation	3%

All material will be graded on a basis of 0-100, with most graded material allowing for grades higher than 100 with bonus questions (usually up to 10% extra) considered. On tests and the examinations some questions may be "compensated" if large numbers of students miss them (indicating possibly a badly posed question or inadequate coverage of the topic in class). On such questions, some proportion of the "lost" credit will be returned. This is the only form of "curving" of grades in the course. All written work is expected to be neat and well presented. A penalty of up to 20% will be assessed for poor presentation on any written work.

The grades from all work will be weighted as given in the above table, totaled, and converted into the Wilkes 4.0 scale grading system using the following conversion:

93+: 4.0	83-87: 3.0	70-76: 2.0	60-64: 1.0
88-92: 3.5	77-82: 2.5	65-69: 1.5	below 60: 0.0

The laboratory exercises will be graded based on submission of materials at the end of the lab session (for some informal reports and demonstrations) or at the beginning of the following lab session (for some informal reports and the formal report). Don't miss lab or class to work on a report! It will be late anyway. Assignments are to be completed and submitted on time. Late assignments will only be accepted in exceptional circumstances, after contacting the instructor concerning the particular case and obtaining agreement. Most of the reports are informal, with only selected materials such as a schematic or table of measurements included. The Lab assignment includes instructions on what is required in these cases.

Informal reports: Selected materials, or fill-in-the-blanks on forms provided
Formal report: A full written report including description of approach, design, results, etc.
Demonstration: Schematic + Demonstrate the correct operation to the instructor in the laboratory. If a demonstration fails to work, an informal report consisting of the schematic/ circuit diagram and description of the problem is to be submitted instead. A correct demonstration receives a grade of 100 if there are no deductions for such things as messiness, improper color coding, lack of bypass capacitors, etc. A demonstration that functions but has flaws or is not fully correct will receive a lower grade. A demonstration that fails can be re-worked for the following week, and is to be accompanied by an informal report, though the grade will reflect the failure to function on time. A readable schematic of usable quality must be available for inspection during the demonstration, so that oscilloscope traces can be observed for certain signals as part of the demonstration grading. The schematic will be collected. An unsatisfactory schematic can result in a deduction of credit. (See the section on schematics in the Engineering Laboratory Reports Manual.)

Homework and Class Participation:

Homework will be assigned, and collected. There will be about 5 to 7 such homework assignments. On the due date, the homework will be collected and a solution set passed out. Questions and discussion concerning the homework will follow. The submitted homework will not be graded. It will be reviewed, and on a selective basis some students or some questions may be examined in detail. The only effect on a student's grade from homework is in the "class participation" category. I will be tracking who does the homework and the degree of seriousness with which it is taken. Along with attendance and tardiness, homework submission will be considered when assigning a number for a subjective 3% of the grade for "class participation." Class participation will also consider lab attendance and effort, including any reports to the class describing your work. (The specifics of any presentations we might do is not yet determined.) A deduction of 1% of the class participation grade unless the student sends an email message to the instructor before class time on January 19 that the student has read this syllabus. If you don't, that's the equivalent of 1 point off your final grade average.

Laboratory Reports:

The purpose of a report is to communicate effectively the results of the exercise. Part of the purpose of the formal lab report is to give the student an opportunity to demonstrate an ability to organize and format a report effectively to convey the essentials of the results concisely. Thus, the formal lab reports are expected to be relatively short and to the point, with a minimum of text and procedure. But, you do need text! Tables and graphs and equations with no text walking the reader through the process or results is not a formal lab report! The lab report is not to include unnecessary material such as a detailed account of all the steps that were required to connect components to build the circuit; the schematic conveys that information quite adequately. Generally, any extra details not needed to follow the main process of the report are given in a summary form, or in an appendix if extensive. Examples of intermediate results or techniques that may be left out or put in an appendix are test circuits, unusual voltage observations, or aberrant behavior. Inclusion of excessive material can have a negative effect on the grade given. You should conform to the usual conventions for figure and table numbering and titles, the use of callouts, proper marking and labeling on graphs, and such. See the *Engineering Laboratory Reports Manual*, of which you should have a copy. (A copy will be provided if you don't have one. The document is also online at < http://www.jbgilmer.com/LabManual/LabManual.htm>.) If you are in doubt about laboratory report issues, ask.

Part of performing the exercise and writing the formal report is determining the format for the presentation of data (if necessary) and results. This is typical of the problem facing a working engineer, where in the workplace there often is no specific format or form for a report. In their reports, students should conform to engineering practices as described in the texts for this class and in the references. For example, truth tables, pinout diagrams, and logic circuits should all conform to such standards. A verbal description is only necessary to the extent that the tabular and symbolic descriptions fail to convey some essential point, or discussion is needed to clarify or describe some point, for example the overall function of a complex circuit.

Every lab report (including informal ones) should include a very brief "success" statement that the circuit operated correctly or that the desired phenomenon was observed, giving the date correct operation was observed by the instructor, if applicable. Should the exercise not be successful, this section will instead describe the manner in which the circuit or experiment failed, and include an analysis of the causes of failure and corrective actions necessary. Such a "failed" lab exercise report can be resubmitted within a week in revised form for additional credit if further work after the due date results in success. Failure to mention successful operation will be graded as a circuit that fails to operate, in addition to a deduction for not including such a statement. Sloppy work will be appropriately rewarded. In cases where the overall report is far short of expectations, it will be "rejected" and awarded a zero grade.

All submissions are expected to be prepared with computer prepared graphics and texts. Note that the "Logicworks" simulation package is a good tool for preparing schematics. (ORCAD/PSpice can also be used for this purpose, but I like the Logicworks artwork better.) Students are expected to use this and other computer based tools for their assignments. However, the "Logicworks" circuit may sometimes be less than completely adequate, and you may need to do a final edit in a graphics program before pasting a figure into your report. For example, the Logicworks common anode LED display does not show the needed common connection to Vcc (5 Volts) which needs to be made. It also does not show connections to power for gates and such. A "binary switch" used in Logicworks is NOT an acceptable representation

of a pushbutton or DIP switch and resistor circuit used to supply a 1 or 0 to a digital circuit. You can summarize grounds and power for all IC's, including the display, in a table in a corner of the schematic, in order to avoid cluttering your schematic with power connections. You should normally use discrete gate parts rather than multiple gate IC parts in Logicworks so that the schematic circuit will be clearer. Be sure to label all parts, e.g. U1, U2, etc. for IC's. This is especially important for IC's which have multiple gates in different places in your schematic. You must indicate pin numbers as well. Remember, clarity and ease of understanding for the reader is the key objective.

Informal reports should conform to the general principles described above, but the material to be submitted is limited to that explicitly called for in the Lab assignment.

Collusion and help on assignments:

All graded material handed in is to be the student's own work. For this purpose, the homework exercises are ungraded. A student may get help from another student or anyone else in matters of technique or background, but answers and written text are not to be copied. Your submission should include no code or schematics, etc. directly or indirectly copied from anyone else's material, except for materials supplied to the class or from library materials or other approved and generally available sources. When you do include any kind of copied material (including that from the sources mentioned), you must identify it as copied and identify the source. You must attribute it properly. Evidence of copied material not given proper attribution may be rewarded by zero grades on the concerned material or more serious sanction if the situation merits such action. If you are in doubt about whether some form of help to or from another student is allowed or not, ask me. At the least send an e-mail to explain. Whoever receives the help should acknowledge it. It may even be worth some extra credit to the person giving help if the help given is proper and correct. Many lab problems will have individual variations, so that there should be no problem giving or receiving general help from someone.

Notes:

A loose-leaf notebook of old class notes will be kept in the library on reserve. This will include lecture materials used, worked homework assignments, and test solutions from previous offerings of the course. You are not obligated to copy any of this; it is merely meant to be helpful. Any material that is really needed will be distributed in the form of handouts in class (e.g. data on certain semiconductor devices).

Attendance:

Class and Laboratory session attendance is not optional. Roll will be called at the beginning of class. Missing a week's worth of classes (for class / lab sessions) or two laboratory sessions, unless explicitly excused, is reason to award a zero grade for the class. If you expect to miss a class, inform me by email or phone message immediately, and if you cannot reach me, in an emergency contact the engineering administrative assistant Lisa Colavitti at (408-4810).

Useful References:

1. TTL Handbook (there are various versions.) This reference describes almost all of the SSI and MSI (Small and Medium Scale Integration) TTL integrated circuits that you might wish to use, including pinouts, electrical characteristics, and miscellaneous other data. This is a very useful volume. It was required once in the past, but has not been made mandatory more recently

to hold down costs. Some data of this type is included in the author's web site, www.ddpp.com. Data on certain devices will be distributed in this class or the lab, but won't be comprehensive. Technical data on integrated circuits is generally easily found online, and you will be able to use the lab computers for this purpose. (But there is typically one per two students. You can bring a laptop though, or use other digital devices with web access. Parts for the kits are usually supported by documentation at Jameco's web site.

2. Various other books and references may be made available in the Lab.
3. Engineering Laboratory Reports Manual

Logicworks:

We will be using the digital simulation software Logicworks as a design tool in this course. (My understanding is that this program is no longer available for new purchase, but we will use it anyway.) We will be using version 5. Version 4 is also available, and is acceptable for most purposes. It is possible some older versions are still floating around, and someone may try to foist one on you as a substitute. Don't be deceived! Versions earlier than 3 do not have busses, which will be handy later in the course. Version 3 is much more awkward in handling programmable devices. The 4 version is available for either the Mac or Windows (On the same CD). Its documentation is in some ways superior to that of version 5, and can be useful in understanding both of the more recent versions. Version 5 is a bit easier to use for programmable devices, but is unavailable for the Mac. We want everyone to be able to obtain a version you prefer for your computer (if you have one) or for your use on a Wilkes computer in the lab or SLC126 or student lounge (if you do not have one). It is, of course, most convenient to have it on your own computer. Logicworks is also available online from electronic sources. Get to know it well. Libraries of parts that are not included in the basic package are available, and I will supply these if needed. In addition, other useful stuff is available at the Capilano web site. I understand that some versions have bugs for which patches are available from Capilano. Be sure to get any for your copy. One advantage of Logicworks 5 is that it supports VHDL programming. VHDL is a hardware description language. Normally we have used "WINCUPL" instead because it better supports the programmable devices (GAL's) that we will use. But this is one reason to possibly prefer version 5 over 4. However, we will use FPGA's but will likely use the "Verilog" language instead of VHDL. My understanding is that version 5 is more stable on Windows computers. Old versions of Logicworks 4 are available used through Amazon or Abe's. Be sure to get a copy with the disk if you do that.

WinCUPL, Quartus 2:

We will use WinCUPL to program GAL 16V8 devices to implement combinational logic. Each of the eight cells in these devices can implement a fairly complex Boolean expression, so that the one programmable device can substitute for several discrete gate devices for the seven-segment decoder. WinCUPL is a bit "buggy" but it is free, and we can usually overcome debugging and logistics issues without too much difficulty for the combinational logic exercise. Later, we will use much more complicated Field Programmable Gate Arrays (FPGA's) for the last sequential logic exercise. These devices have the equivalent of millions of gates on them. We will be using Quartus 2 to program the FPGA's. Last year we had some technical issues with the "Byteblaster" programming devices which should be resolved by the time we get to this exercise. Further information will be developed and distributed later. (When we are doing programmable logic, you might want to bring your own laptop. Bring a thumb drive too.)

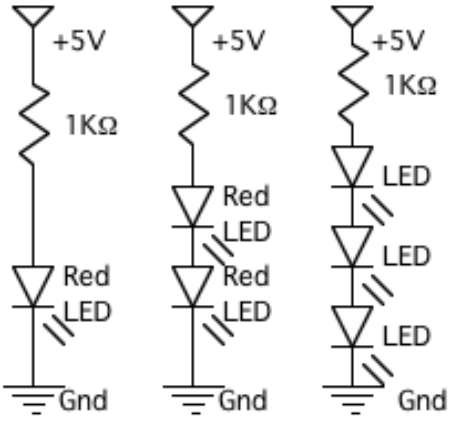
EE 241 Digital Design Lab Assignment # 1
No report required (To be performed Jan 19, 2018)

The primary purpose of this Lab session is familiarization with some of the equipment in the lab, and some of the more basic components. Each student should have a solderless breadboard with a kit of parts to be used in this and later laboratory exercises. The lab session will conclude with independent examination of some of the more basic parts and a few simple circuits.

Equipment to be familiar with: 1. NI Virtual Instrument 2. Solderless breadboard

Components to examine: Light Emitting Diode (LED), resistor, diode, bipolar transistor

1. Form the simple circuits shown at right, adding LED's (try yellow or green too), while observing what happens. Try adding a fourth series LED. (Note that we mean "circuit" in a practical sense; we generally do not clutter schematics with the power and ground networks.)

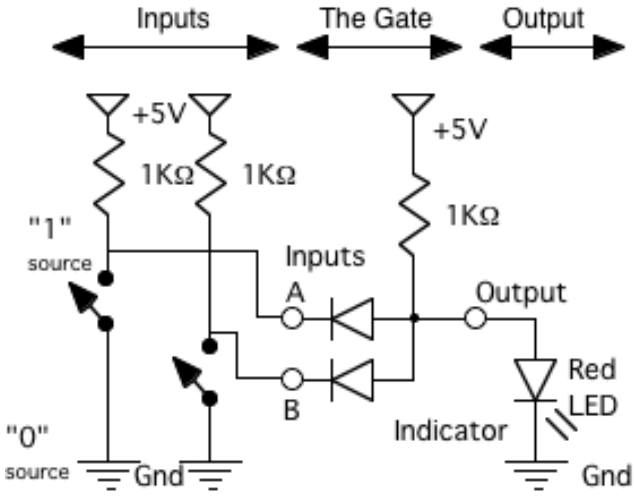


What do you see? Why? Using your voltmeter, measure the voltage across an LED and the resistor in each case. How much current is flowing in each case? What happens if an LED is backwards?

The first circuit at left can be used as a "probe". Use the top end (with bottom at ground) to test for a high voltage (usually a logic "1"). Or, use the low end (with top at 5V) to test for a low voltage (a logic "0").

Note that we must **always** put the resistor or some other component in series with the LED to limit current, or we will have a dead LED. (Why?)

2. Using signal diodes (1N4148, not 1N4004 or zeners), form the circuits shown below. Give "0" to both inputs of the "gate" circuit (by pressing both pushbuttons or equivalent). Give "1" (but not "0") to both. Try one of each.

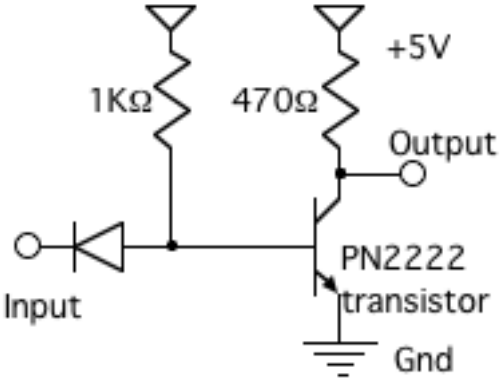


What do you observe? What would we call this simple gate: OR, NOT, AND, NOT-AND, NEVER, ALWAYS, ... if we let a "1" mean "true" and "0" mean "false"?

What voltage is a "1" at the input? A "0"? What voltage is a "1" at the output (check with and without the LED)? A "0"? What are the limitations of this kind of gate? (Would we be able to cascade them?)

Note that with current flowing through a "normal" silicon (signal) diode, you get about a .6v to .7v voltage drop. (With germanium diodes, which are not normally used for digital circuits, we get

closer to .2V. The reasons have to do with semiconductor physics, which we won't get into here. LED's are made of Gallium Arsenide, and have a higher forward voltage drop than silicon, about 1.6+ volts (red) or more for green or yellow.



3. Build the circuit shown at right.

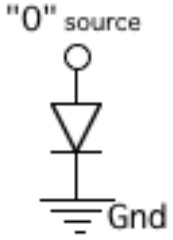
Try connecting it to a "1" (as shown earlier, through a resistor), then a "0", while observing the output. What do you see? Why? What would we call this gate?

The transistor is acting like a switch. A small current through the "base" causes a much larger current through the "collector". How much current can this gate "source" (when the pull-down transistor is "off") into a load at 2V? How much current can this gate "sink" (when the pull-down transistor is "on") if the collector current is 100 times the base current, if there is a load able to supply that much current? Assume the Load is at .8V. (Does that matter?)

4. Things to think about:

How would we form a gate that performs the OR function? (See if you can build one.)

How can we use the gates from parts 2 and 3 in combination? What would that do? What would we get if we connected two such gates in series? Would they work properly? Try building an AND/OR gate: Z=AB+CD (Z is on if A and B are both on, OR if C and D are both on.)



Suppose a "zero" is actually higher than ground level, say about .7 volts instead. The circuit at right illustrates. Would our gates still work correctly? (If you have time, try it.) Note the similarity of this "0" to the output of the gate in part 2 when one of the inputs is low.

For lab exercise #2, you will be putting to use much of what you have observed here.

Be sure you look over the parts in your kit, and be familiar with those you will be using in each of the upcoming exercises. Have them identified and ready for the start of the lab. You don't want to waste 20 minutes trying to find the right parts. You will need some way of keeping things organized. Small plastic boxes with compartments help.

Read the handout descriptions of all the parts you will be using before every lab.

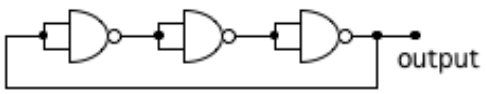
EE 241 Digital Design Lab Assignment # 2 Due at end of lab period, Jan 26, 2018

The purpose of this lab exercise is to gain a better understanding of the diode and transistor parts, and the circuits that we have when they are assembled to form a "gate". We then look at IC's that do the same thing, and their characteristics. The student should design a bipolar NAND gate prior to coming to the lab, for use in this exercise. You should build it ahead of time. Some of the wiring for later parts can be done ahead of time too. There is a lot to do here, and anything you can do ahead of time gives you a better chance to finish. Use the attached pages for your report. Turn in the report at the end of the Lab session. Be sure you do not omit anything. If you have tables of collected data, you may attach them to the lab report.

1. Design and build a 2 input NAND gate using diode, bipolar transistor, and resistor parts. Confirm that it works properly using the "0" and "1" sources of the previous lab exercise. With both inputs connected together, and the output connected to a load of 1K resistors to both ground and the 5V supply, plot output voltage versus input voltage for your gate. (Use a potentiometer to vary the input voltage, while you observe the output voltage with your meter. Connect the ends of the pot to ground and Vcc (to Vcc through a 100 Ohm resistor), with the slider being connected to your circuit input. Adjust the pot to vary the input voltage. Then, after recording an output voltage, measure the corresponding input voltage. It helps to borrow a meter (or use an oscilloscope) to allow two measurements to be made at the same time, of both input and output voltages. Include this table in your report, as well as the graph. You should have at least several data point, spaced more closely when the output is changing. What are "good" values (give a voltage range) for "0" and "1"? For each, determine a reasonable upper or lower bound for being valid. Show “excluded regions” on your Vout vs Vin graph that correspond to these V_{IL}, V_{IH}, V_{OL}, and V_{OH} specifications that you have chosen.

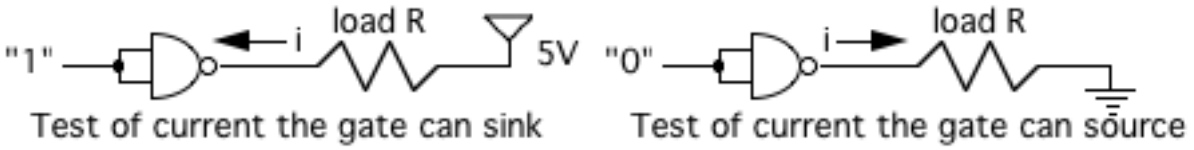
2. Repeat #1 above using the 74LS00. Connect both inputs (e.g. pins 1, 2) together for one gate as the input, and take the corresponding gate output as the output voltage. Load it as above.

3. Form a ring oscillator with your 74LS00 and also one of: 74S00, 74C00 or 7400, 74HC00, or 74L00 parts, as provided, and record the frequencies. (The actual parts available varies from year to year with availability. Different students should have different devices.) The logic families used (LS, S, C, etc.) may vary. The lab instructor will give particulars as necessary.) A ring oscillator is formed by connecting the inputs of each gate together to form an inverter, then connecting three of them head to tail in a ring. (Think about it: why not four?) What is the implied typical gate speed for each? (Think carefully about this; try going through a complete cycle with 1's and 0's on the circuit diagram.) How does that compare with published data for these families? (Look at typical, maximum delay data.) Note that the pinouts of all the devices labeled either 74xx00 and 74xx04 etc. are the same, so you can pop one out of your circuit and pop another one in, if you have been careful not to trap the IC with wires passing over it. (All the different families for the same logic device have the same pin arrangement, with just a few exceptions.) Be sure to bypass the power supply!



Ring oscillator

4. (extra credit: 10 points) We are interested in how much current the 74LS00 can sink with a zero output (and a "1" input), with the output voltage staying below .4 V. How much current can it sink with a 180Ω resistor to the 5V supply as a load; does this cause the output voltage to swing higher than it should? Try it also with 1K and 470 Ohm loads. (For a bit more extra credit, find the resistance that gives exactly .4V out. This is another place where you may make good use of a potentiometer.) We want to also do the same thing for the opposite logical condition. Does the performance of the device conform to data sheet specifications?

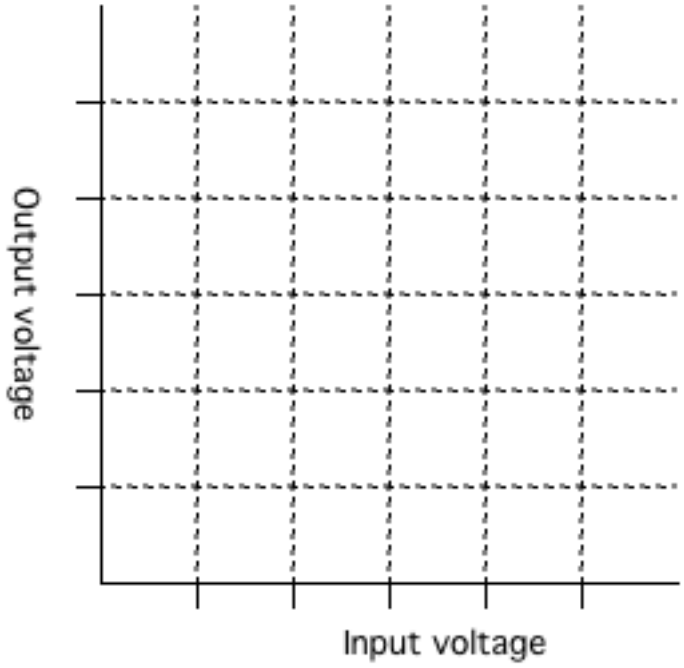


Test of current the gate can sink Test of current the gate can source

Lab #2 Informal Report Name:_____

1. Schematic of your NAND gate (including the test voltage source and load):

(include data table)



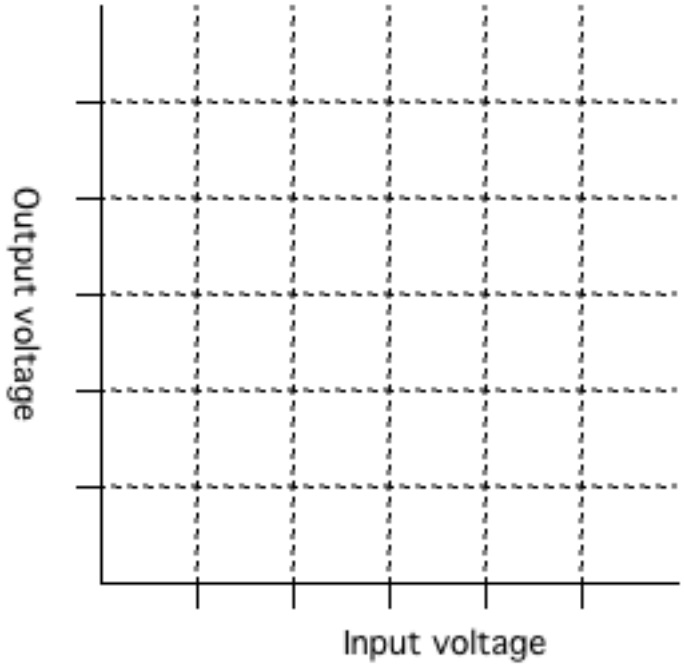
Did it work properly? Explain.

What are good voltage values for: 1 input: 1 output:

0 input: 0 output:

Why these values?

2. For 74LS00 (include data table):



What are good voltage values for: 1 input: 1 output:

0 input: 0 output:

Why these values?

3. Ring oscillator

Device: 74LS00 _____(other)

Measured oscillator frequency: _____ _____

Calculated Gate delay: _____ _____

Specified Gate delay (typical): _____ _____

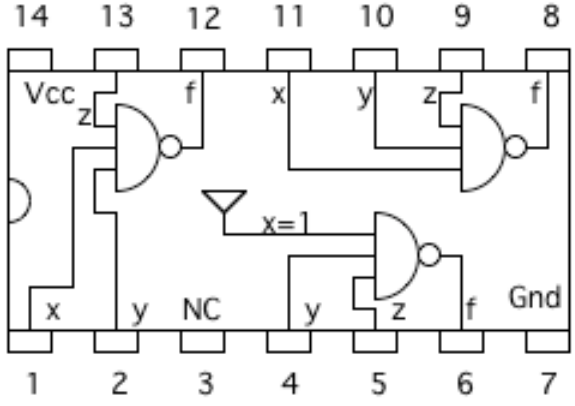
Specified Gate delay (maximum): _____ _____

Show work for frequency, gate delay:

Extra credit answers (#5): attach

EE 241 Digital Design Lab Assignment # 3
Due Feb 2, 2017 at the end of the lab period

This assignment is to discover the Boolean function performed by three randomly selected TTL (Transistor-Transistor Logic) SSI (Small Scale Integration) integrated circuits. All IC's are combinational logic (no flip-flops). Three devices are selected from the assortment made available: one each from boxes "A", "B", and "C". Power connections are Vcc (5V) at pin 14, ground at pin 7. Determine which pins are inputs and which are outputs. Then look for a pattern in how changes in the input values cause the output logic values to change. In most cases an IC will have a number of different gates. Your results submitted should include a truth table for each kind of gate (as it actually is on your device), and a pinout diagram of the chip as a whole. Note that you do NOT have to identify the chip by part number; some would be very hard to identify this way. For example:



All gates:

x	y	z	f
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Two 3 input NAND gates and one 2 input NAND gate

Hints: See your EE241 textbook for a schematic of a typical TTL gate. Note that if an input is not connected (left floating) it is, in effect, "on" or "1" (high). An input that is simply floating will register about 1.7 volts on a meter, actually in the range where it is neither legitimately on or off, but it will register "on" for our purposes. Outputs will be either "on" (logical 1, about 3.5 volts) or "off" (logical 0, less than .5 volts). Thus, with a voltmeter you can tell inputs from outputs. Connecting inputs to ground temporarily with a short piece of wire causes them to go to logical 0 ("off"). A voltmeter, or simply an LED with series resistor, can be used to indicate the output state as the inputs change. A few of the IC's are more difficult than the others: they have "open collector" outputs with no internal "pull up". Others have various defects. You must correctly identify the actual functioning of a chip (with the defect) to receive full credit. A few of the defects are particularly subtle, such as shorts between inputs for a NAND gate or a defective pull-down transistor for the output. If you correctly identify one of these faults, there will be a bit of extra credit. If you miss it, there will be a small penalty (a point).

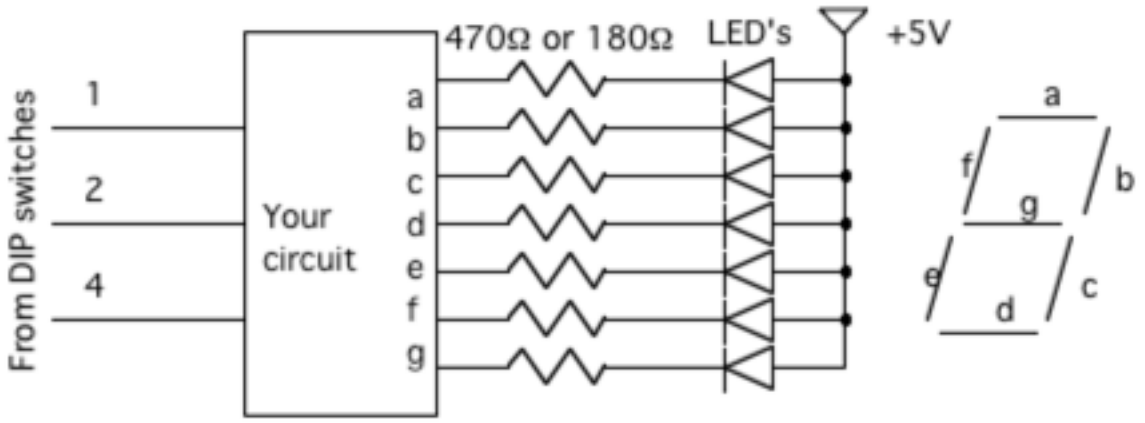
For extra credit: You can identify the device family, e.g. "standard TTL", "L", "LS", "S", etc. Oscillator ring speed and current sinking and sourcing at inputs and outputs can be used for this purpose.

You should prepare for this lab by trying the same exercise ahead of time for some of the known chips in your kit, such as the 7400, 7403, 7432, and 7411. See if you can figure them out without using documentation, and build a truth table for them as you will need to do for this lab.

Grading: A and B devices 40 points; C device 20 points; extra credit for hardest ones.

EE 241 Digital Design Laboratory
Lab Assignment # 4 (Demonstrate Feb 16, Formal report due Feb 23 2017)

You are assigned to design the combinational logic for a display, which is to use 7 segment LED's. The input is in "octal", coded on three binary signals forming the input bus. The 3 bit words can have the values 0 to 7, but coded in a peculiar manner (which makes each student's problem different). Your circuit must convert the octal digits into the seven (or 8) signals necessary to drive the display segments. The figure below illustrates the design problem:



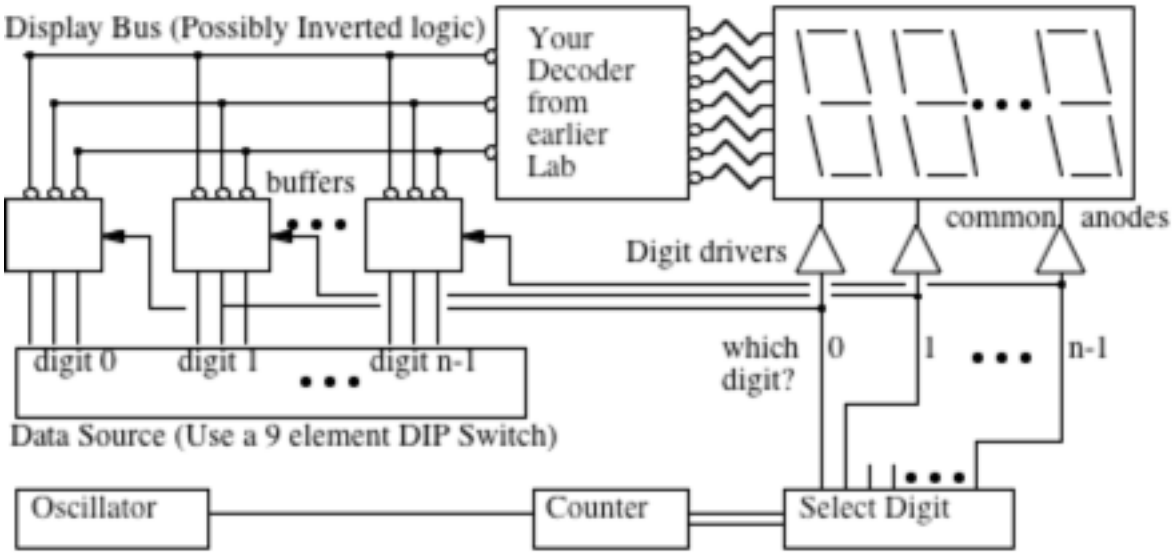
The conversion to the value to be displayed is a code to be developed by each student. We will be using this same circuit later for the multiplexed display, so choose a series of eight different characters that can make up a message or a few words that could be displayed by the LED's. Later we will extend the technique to allow a longer message. If nothing better comes to mind, use your email address, rendered as best you can with a 7 segment display. The code for each student must be different. You must also select a code such that no segment duplicates the logic function of another, or is a constant "1" or "0". (If you find this is the case, you must change your code.) Do not duplicate symbols; all 8 must be different.

Design, simulate (with Logicworks), and then build the combinational logic circuit. Show the Karnough maps used. Use Logicworks to test the design and show that it will work correctly before you build it. You would like to minimize the cost of the circuit as measured by the number of integrated circuits, gates, and inputs. Build the logic from the integrated circuits available, and demonstrate it with your 7 segment LED display. (Note that an output of a low voltage is needed to turn on the LED for a common anode display.)

The formal report is to include, among other things, your code (the specification), a truth table, your logical design in the form of Boolean equations and Karnough maps, your circuit (as a proper schematic, prepared using Logicworks and printed in a usable format), and a conclusion. Schematics should conform to standard usage and conventions, with part and pin numbers identified. Comment on any particularly interesting or innovative features. Give the cost of the circuit in terms of the number of gates, gate inputs, and the number of chips, as an appropriately formatted table. The Conclusions section must state whether it worked correctly, when it was demonstrated, and to whom (the instructor). The schematic must be prepared with Logicworks, and all material included must be prepared using computer word processing and graphics. Remember that the Abstract is NOT a substitute for an introduction. You should write it last. (The laboratory report should stand alone without the abstract.)

EE 241 Digital Design Laboratory Exercise #5 Demonstrate on March 2, 2017

Your Octal to 7 Segment decoder is to be part of a larger system that displays a multi-digit value. Only one octal decoder will be used, with various digits multiplexed to it. The output will appear on an array of common anode displays. While only one digit is lit at any one time, if they are scanned rapidly enough that it will appear that all are lit. The circuit's structure is shown below:



Design and build such a circuit that will display 9 bits worth of data on 3 digits of a display. Use for input a 9 position DIP-switch. For our case, n=3, but the above design accommodates an arbitrary number of digits. We will later expand it to 4, so save space on your breadboard for a fourth digit, or more if you desire. (A case can be made for 8 digits to convey a meaningful alpha text message, including a space between two words.) For the display, you will use the three separate digit devices, with all of the "a" segments wired together, "b", etc. Note that the digit anodes cannot be adequately driven directly by TTL levels, which have insufficient Voltage. You must design a simple transistor “high end” digit driver that sources or sinks enough current to light the digit adequately. (Digit drive signals enter the digits from the bottom in the diagram above.)

Test the design using a simple counter circuit that produces the digit drive signals sequentially as shown above. (Include this circuit on your breadboard.) If you are an "even" numbered student, scan the digits left to right. If an "odd" numbered student, scan them right to left. (Use Wilkes ID.) One third of the class will use 7403 open collector logic for multiplexing, one third will use 74LS241 and/or 244 bus drivers, and one third will use 74LS153 multiplexers. Demonstrate that this circuit works correctly. You must have an acceptable schematic (hand drawn is OK) to show the instructor and turn in when you demonstrate the circuit. It should show pin and part numbers. If it does not work on time, submit a schematic and a short description of particularly interesting points, description of how it behaves, your assessment of the reason it does not operate correctly, and the way to fix the problem.

Typical faults that cause loss of credit include dim displays (usually due to improperly designed high end drivers), unequal brightness of different numbers, e.g. "1" is brighter than "8", time gaps when no digit is displayed, and "bleeding" of current into display segments that are supposed to be off, and inadequate bypass capacitors. Loss of credit for a messy, incomplete, or an inadequate schematic is also likely. When you do your demo, have an oscilloscope fired up and handy (with two probes ready) so the instructor can probe interesting parts of your circuit.

Be very careful with the LED displays. Burning out one segment ruins an entire digit. As a matter of being extra safe, put a 100 ohm resistor in series with the digit drive until you are sure it is operating correctly. Using a counter makes this a sequential circuit, which is sensitive to noise. Use bypass capacitors! Your circuit will tend to do strange things if not adequately protected.

Be aware that leakage current through your transistor digit drivers may prevent the driver from turning completely "off", so segments appear to all be dimly lit if not on. This can be fixed with a better driver design. (You want bias to reduce Vbe to below .6 volts when off.)

A couple of final comments:
This project is a system of circuits. There are several different sub-circuits that are each separate circuits that can be designed and tested individually, before you try to combine them into one system. One of those component circuits is the 7 segment decoder designed in the previous lab exercise. You've already tested that. Another is the multiplexer. You should build and test that, perhaps using three LED's to show the output. That way you can see if the 9 switches display properly depending on the multiplexer select inputs. The digit drivers (that turn on and off the individual digits) are an electronics design problem; they can be built and tested with individual LED's. You can then integrate the multiplexer, decoder, and display, using manually controlled multiplexer selects and digit selects, to see if all that works. The oscillator - counter can be designed as a separate step. Finally, you need to specify (truth table) and design a combinational circuit that will assert the correct values to the digit and multiplexer select signals, given the counter value. You can test it separately too. Finally, you can then integrate it all together and see it work, or continue debugging if it does not. The oscilloscope will be a useful debugging tool, since now the time dimension is important, and you need to observe sequence. Building a system of separate circuits is inherently a complex undertaking. This is an important step forward in the level of complexity and sophistication of the systems you design and debug.