University of California, Santa Cruz
Electrical Engineering Department

A Note on the Proper Documentation of Engineering Schematics

EE174, Intro to EDA Tools
S.C. Petersen

I. Wiring Diagrams, Schematics and Engineering Schematics

A wiring diagram is a particular kind of electrical schematic. It is simply a graphic mechanical drawing showing the interconnection of all devices and components that implement an engineering design. Integrated circuits are typically represented as abstract block diagrams with pins arranged as they actually appear on corresponding physical devices. Often these blocks are drawn as rectangles or squares to mimic real device packaging. Capacitors, resistors and other components are represented with their common abstract electrical graphic symbols. The aim is to make the schematic look as much like the actual wiring as possible. The problem with these drawings is that no effort whatsoever has been made to include elements of the underlying otherwise non-physical abstract engineering design that originally went into the circuits' creation to begin with. These kinds of schematics are common with industrial application engineering notes (because those who produce them don't know any better) and, without instruction are the way students invariably begin trying to document electronic circuits. The sole value of this kind of graphic it to provide information pertinent to wiring circuit elements together. Indeed, some professionals believe this is the proper way to draw them – hence the reason they appear in industrial application notes. Generally, though, at least as engineers we are also interested in learning something about the underlying engineering design as well. Deliberately modifying a schematic that is merely a wiring diagram results in what I like to call an *engineering schematic.*

Hence, engineering schematics are a conceptual superset of simple wiring diagrams. They convey the same basic interconnection information, but because the components and wires are arranged differently, significant conceptual understanding of much of the engineering design can also become discernable; they are meant to be intelligible, *i.e*, to be read and used by engineers. In this Note I will use schematics and engineering schematics to interchangeably refer to the same entity, and for a schematic which is merely a wiring diagram I will call a wiring diagram. Engineering schematics are particularly useful in prototype and experimental work because they are easy to troubleshoot from. On the other hand, wiring diagrams merely show how components are connected on a circuit board or other particular construction format, but convey little if any engineering content.

Students invariably confuse engineering schematics and wiring diagrams. While there is a one-to-one relation between them, only one easily allows you to make technical deductions about a circuit's behavior while the others' sole value is to verify point-to-point physical interconnections and perhaps how to arrange actual parts. For some strange reason, students almost always want to draw wiring diagrams and use them exclusively. As you will see, this often makes troubleshooting difficult or impossible. Let's look at a simple example to illustrate the difference.

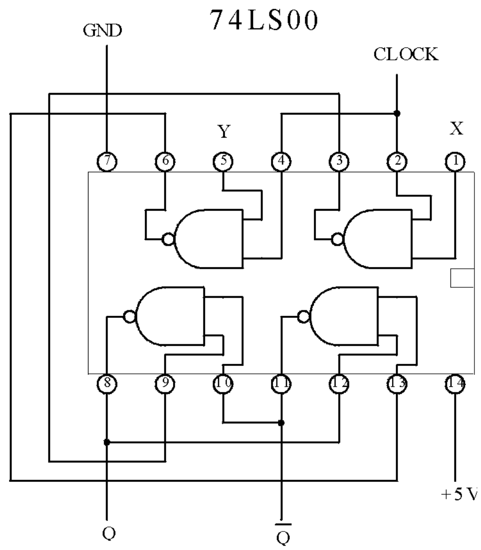

Fig.1. A wiring diagram.

Fig. 1 shows an example of a simple wiring diagram as a student actually wired it on a digital breadboard showing the top view of a 74LS00 TTL Quad Nand package. By convention, the top view always locates pin 1 in the upper left corner of the actual physical *package* with successive pin numbers increasing clockwise around the chip. Notice that except for the wiring interconnections, no other information is conveyed except perhaps what you might conclude from the signal names.

Fig. 2 shows this same circuit redrawn as an engineering schematic, which also reflects the way the chip was actually wired.

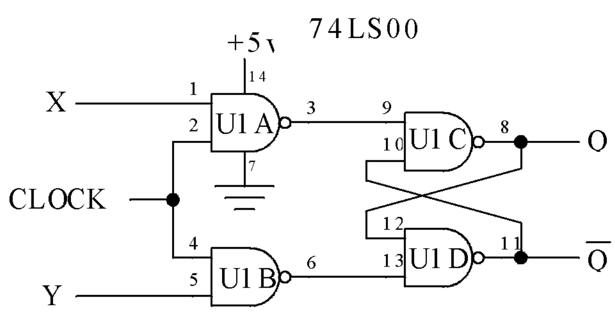

Fig.2. An engineering schematic.

Clearly, considerably more information is also evident, such as its function as a memory block. Having completed CMPE100, we instantly recognize it as an asynchronous nand latch. Although the pin numbers convey the same information as the wiring diagram, their spatial location with respect to the actual package is lost. Moreover, we don't know whether each nand gate is in fact part of the same package, or four separate packages. To handle this identification problem each gate is associated with a particular packge by the "U" number convention that assignes unique unit numbrs to physical packages.   Hence, unit package 1, "U1" refers to a physical device having four identical sub-components abstracted as symbolic 2-input nand gates and denoted by arbitrary sequential subscript alphabetic assignments, beginning with A. Engineering schematics should be strongly favored for troubleshooting over wiring diagrams.

II. Hardware Description Languages and Engineering Schematics

Hardware description languages, like Verilog or VHDL were created primarily to facilitate the realization of hardware in a format more easily read and implemented by a computer than are graphic schematics, which are meant to be read and interpreted first by humans. Hence, hardware elements are expressed in a specially formatted sequential listing of components and their interconnections. Accordingly, self-documenting details are included as written comments interspersed in the listing of components, properties and their interconnections.

Schematics created by engineers skilled in HDL's have their own unique tell-tale characteristics that result in documents as unreadable as the wiring diagram format. Components are often translated as block diagrams surrounded with disconnected wires, known as "nets", having only their net names displayed. This collage of disembodied blocks carries as little design information as a wiring diagram does.

III. Schematic Graphic Guidelines

Guidelines have developed over the years for producing well-drawn engineering schematics that enhance their engineering design and troubleshooting self-documentation properties. Unless you have a good reason for deviating from them (and we sometimes do), adhere to the following:

- Signal flow should move from left to right across the page with inputs on the left and outputs on the right.

- Use the "unit number" convention for assigning a unique package identifier to each chip. For example, the 74LS00 in fig. 2 is a single integrated circuit (also sometimes referred to as a "chip" or "package") containing four independent Nand gates, so the chip is assigned identifier U1 with its internal gates identified by letter suffixes: U1A, U1B … etc. Many variations of this convention exist in industry. Only one of the common gates need show the power connections (in this case, pins 7 and 14). Power pins are often omitted as a further abstraction, but you should *always include them*, both as a reminder that chips need power to run, and to make your engineering schematics complete – especially when bypass capacitors are connected to them to suppress power supply noise (these are unwanted power voltage variations that can cause glitches and other problems).

- Electric potentials (voltages) should increase as you move from the bottom to the top of a page. For example, fig.2 shows the +5V supply upwards while the ground pin (0 volt reference) is downwards (where "ground" or the 0 Volt reference node really is).

- When using **block diagrams** to show a component, the same left-to-right signal flow and potential orientation should be adhered to where possible. Additionally, *every signal* should be labeled *inside* the block with pin numbers (if appropriate) on the outside. Students (and some textbook authors!) often draw block diagrams without labeling the associated interconnecting wires. Sometimes this is self-evident from the context, but should not generally be relied on. Examples of poorly drawn block diagrams are MSI, LSI and VLSI parts, such as counters and CPUs.

- Annotate all passive components with unique instance designators. A partial list of common designators is listed below:
  - Resistors should use prefix “R”, such as R1, R2, R36a, etc.
  - Capacitors would similarly be C1, C2, etc.
  - Inductors: L1, L2 …
  - Switches: SW1, SW2 …
  - Connectors (or “jacks”) and plugs are assigned usually assigned J or P prexfixes.

Always begin any actual experimental circuit work with a complete and readable engineering schematic that accurately reflects what *has actually been built.* This allows you to reason *from the schematic* while troubleshooting, taking data or making modifications as you go along. Wiring diagrams (fig. 1) can also be quite useful, but they should never substitute for a good well-documented engineering schematic. Annotate your schematic as you build the circuit. If you don't do this, it will be nearly impossible to troubleshoot after it inevitably fails to work the first time. For example, this would naturally include pin numbers to accurately document how devices are connected, and whether bypass capacitors were placed across a chip’s power pins.