Introduction to Digital Hardware and Timing Diagrams

Cmpe121, Winter 2002
S.C. Petersen

The design of digital hardware requires choosing and interconnecting actual devices that need to be carefully evaluated for electrical and timing compatibility. All digital circuits are basically analog circuits specifically designed to implement digital behavior. Thus, a timing diagram is also fundamentally analog in nature. It will show the actual voltage corresponding to defined digital states - as well as everything else in between. Karnaugh maps, next-state tables and the like are inappropriate to this purpose because they are not designed or capable of showing specific electrical behavior associated with practical digital logic devices.

We begin by defining what a timing diagram is: a two-dimensional mapping of an electric signal that represents digital states versus time. Such states can be represented either by current or voltage. Since voltage is the most common, our discussion will center exclusively around it. Essentially, this is the same thing one would see by observing a real-time oscilloscope waveform of the voltage on some pin of a particular logic device (note that *logic analyzers* are special purpose oscilloscopes that usually don't show the actual signal being analyzed but an idealization of it stored in memory after sampling and processing for repetitive but not real-time presentation and analysis).

Defining Logic Levels and Noise Margins

Making an analog circuit mimic digital behavior might seem easy today, but it wasn't always so. One of the earliest successful attempts was a family of devices constructed with discrete (separate) components consisting of resistors and transistors that operated between saturation and cutoff; the family was aptly called *resistor-transistor-logic* or just RTL for short. It was slow, large and very bulky. One military manufacturer, for example, supplied nor and nand gates in epoxy-potted plug-in rectangular plastic containers having about the same volume as a single C-cell battery today. Although we might grin at this, students should understand that engineers approached the problem of inventing these circuits the same way you would: by working with existing components and materials technology. Most advances in the art are incremental rather than revolutionary. Of course the revolutionary idea was the invention of the epitaxial integrated circuit or classic *IC* which enabled us to put analog circuits on a single slab of doped semiconductor material, like n or p-type silicon. Incremental changes quickly followed. Common discrete components that were the bread-and-butter of analog designs, like resistors and capacitors are very costly in terms of die space to implement in silicon, so they were largely eliminated by developing ways to directly connect the really needed transistor components. Ways were also invented to substitute needed resistors with easily made transistors etc. What resulted was the first commercially viable family of digital logic devices known as *transistor-transistor-logic* or TTL. The classic 74-series TTL family (54-series is the military equivalent) was the first of what would eventually become a large set of electrically different device families all having the same 74-series packaging and pin connections.

For example, consider the ubiquitous 7400, a 14-pin chip with four independent 2-input nand gates. The following list of families using the same package and pinout is not complete but gives you an idea of the variety of the 74-series.

7400	The original "vanilla" TTL device family.
74L00	Low-power; early and now obsolete.
74H00	High speed; early and now obsolete.
74S00	Schottky; early high-speed version, mostly obsolete but still around.
74LS00	Low power schottky; best of 74L and 74S, probably most commonly used overall.
74ALS00	Advanced low power schottky; improved LS version.
74F00	Fast; very high speed family.
74C00	First CMOS version with 74-series pinouts
74HC00	High-speed CMOS.
74HCT00	High-speed CMOS with TTL logic voltages
74AC00	Advanced CMOS.
74ACT00	Advanced CMOS with TTL logic voltages (sometimes called FACT)

Choosing a logic family is based on constraining factors, like cost, speed, fan-out, power and interface compatibility. Often we engineers simply use one because we are already very familiar with it. This is especially true with the LS and HCT families.

TTL assigns *ideal* logic levels to +5V (logic High or “1”) and 0V (logic Low or “0”). The non-ideal nature of all circuit implementations cause deviations from ideal voltages that must be understood and considered in any realistic design. Outputs will be less than exactly +5.0 V or more than 0V. These non-ideal voltages must still be correctly interpreted as being representative of a “1” or “0”. Fig. 1 below shows two digital inverters connected together with graphs comparing the output characteristics V_O of the first inverter with the input characteristics V_I of the second inverter. These voltages are defined as follows:

$V_{OH\,\min}$ is the *minimum* output voltage possible in the logic high state. It tells us that if the output isn’t ideal (5.0V), at least it well never be less than this.

$V_{OL\,\max}$ is the *maximum* output voltage possible in the logic low state. It tells us that if the output isn’t ideal (0.0V), at most it will never exceed this limit.

$V_{IH\,\min}$ is the *minimum* input voltage that will properly represent a logic high. Anything greater than this all the way to +5.0V will be properly understood.

$V_{IL\,\max}$ is the *maximum* input voltage that will properly represent a logic low. Anything less than this down to 0V will be properly understood.

Since $V_{OH\,\min}$ is greater than $V_{IH\,\min}$, the difference defines a margin for error. Unwanted variations at an input can be caused, for example, by inter-wire/trace coupling due to mutual inductance, missing or inadequate supply bypassing or poor power distribution design. These unwanted and hence *noisy* variations can readily be seen on an oscilloscope. Formally, the error margins for both logic states is called *noise margin* and is an important figure of merit for a logic family. Without adequate noise margins, it would impossible to build logic devices and have them work reliably.

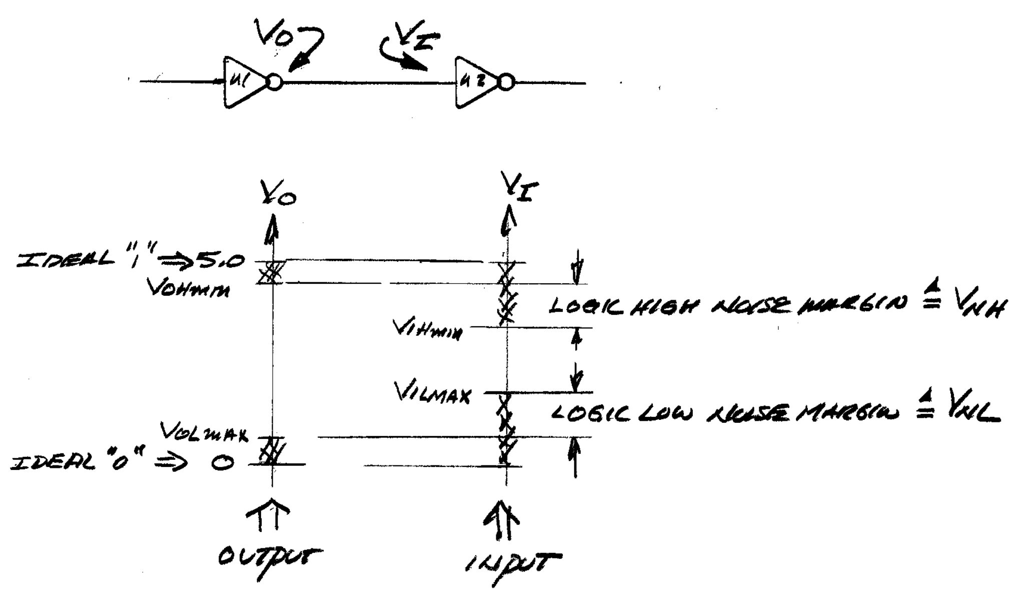


Fig.1. Non-ideal voltages and noise margins.

Logic families designed with bipolar junction transistors (BJT’s) have lower noise margins than do families based on complementary metal-oxide semiconductors (CMOS). As an exercise, compare the noise margins for two families: 74ACT (a CMOS family) and 74LS (a BJT family). You will see that CMOS has much better noise margins.

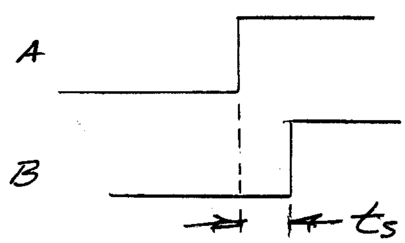
CMOS logic devices are capable of working at different logic high and low voltages depending on attached power supply voltages. TTL is confined to +5 and 0V, although newer "3V" logic designed for portable low-power applications is pushing this limit down. CMOS also draws less quiescent (or static) power than TTL since it draws power only during the short interval when it changes logic states. Unlike TTL, CMOS outputs will swing very close to either supply with an improvement in noise margin.

Several hybrid families have been developed expressly to work with CMOS logic: 74C and 74HC, for example. These incorporate internal CMOS hardware and some have BJT blocks as well. Variations that are compatible with TTL logic levels have a "T" suffix, like 74HCT or 74ACT. These families can mix directly with TTL families, like LS.

Basic Timing Parameters

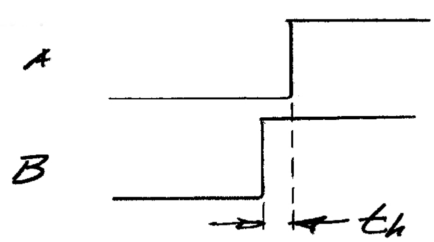
At this point, we are ready to discuss the basic electrical parameters necessary to design, analysis and understanding of electrical digital devices. These are defined as follows.

Setup time, t_{setup} or t_s **:** The time a digital signal *A* must be stable and unchanging prior to another digital signal of interest *B*.



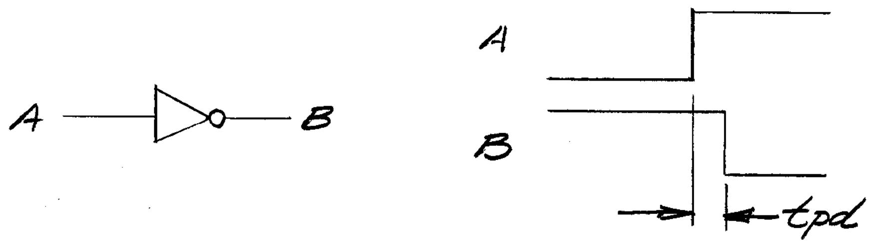
Setup time is most often associated with activity of digital signals immediately before a clock event when these signals must be stable and ready as necessary inputs to clocked circuits, especially latches.

Hold time, t_{hold} or t_s : The time a digital signal *A* must be stable and unchanging following the change of another digital signal of interest *B*.



This parameter is very important with synchronous state machines employing feedback logic that can change as a result of the clock. Propagation delay usually insures that this parameter is satisfied.

Propagation delay time, t_{pd} : The time it takes to assert at an input a digital signal *A* and see its immediate result at an output of interest.



All digital circuits, whether gates, buffers or inverters take time to respond to signals changing at their input(s). High speed devices have smaller propagation delays than slow speed devices and will be affected by load capacity, particularly for CMOS. For example, 74ACT has a delay of about 6.0 ns (50pF load capacity) while 74LS has about 22.0 ns (15 pF load capacity).

Rise time, t_{rise} or t_r : The time it takes for a digital signal to change from a stable logic low voltage to a stable logic high voltage.

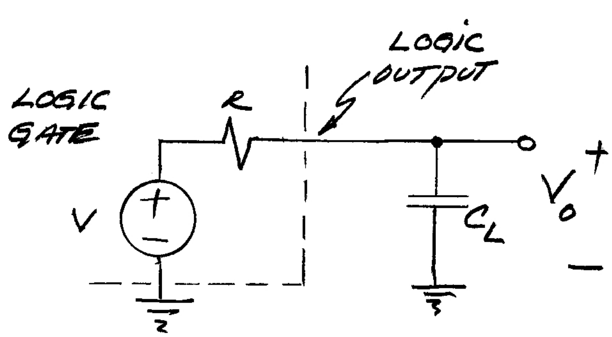
Fall time, t_{fall} or t_f : The time it takes for a digital signal to change from a stable logic high voltage to a stable logic low voltage.

Together, rise and fall times effectively set an upper bound on the maximum speed of a digital device. Since rise and fall waveforms are actually exponential, an idealized approximation depicts them as linear ramps that change at a constant rate from one digital level to the next. Where on this ramp we assign our low and high points to be follows several different conventions. Which one is used on a particular timing diagram depends on what information the diagram is trying to capture or convey.

Using the *end points* really represent the ideal voltages that map to logic “1” or logic “0”. In reality, a logic device will interpret a voltage as a “1” or “0” over a guaranteed range (see noise margin discussion earlier). The most common convention is to use two symmetrical intermediate points on the ramp, like 10% - 90% or 15% - 85% that will always fall within the guaranteed voltage ranges for high or low when connected to an input of the *same logic family*.

Relationship of Transition Times to Output Load Capacity

This discussion should provide insight into the importance of the load capacity a digital device is connected to and the development of the linear ramp approximation universally used in timing diagrams. The model we use for this purpose is quite simple: the logic device’s output is modeled as a voltage source with necessary internal resistance which is connected to an equivalent load capacity. This is shown below as a simple RC circuit.



Time-Domain analysis

To show the analog transition, we excite the circuit with a step change in voltage at its input and observe the response at the output.

Let $v_1(t) = u(t)$ a unit step. Then, $v_2(t) = (1 - e^{\frac{-t}{RC}})$. This relationship is plotted in fig. 2.

This result follows quickly from elementary network analysis, and can easily be shown using the Laplace Transform as follows (intermediate math details are omitted for brevity).

$$L\{v_1(t)\} = \frac{1}{s} \text{ so } v_2(s) = \frac{1}{s} \cdot \frac{1}{s\tau + 1}.$$

Noting the inverse transform pair:

$$L^{-1}\left\{\frac{1}{s-a} \cdot \frac{1}{s-b}\right\} = \frac{e^{bt} - e^{at}}{b-a}, \quad a \neq b.$$

Letting $a = 0$ and $b = -\frac{1}{\tau}$, we can write $v_2(t)$ as

$$v_2(t) = L^{-1}\left\{\frac{1}{s-0} \cdot \frac{1}{s+\frac{1}{\tau}} \frac{1}{\tau}\right\} = \frac{e^{\frac{-t}{\tau}} - e^{0t}}{0 - \frac{1}{\tau}} \frac{1}{\tau} = 1 - e^{\frac{-t}{\tau}} \text{ with } \quad \tau = RC$$

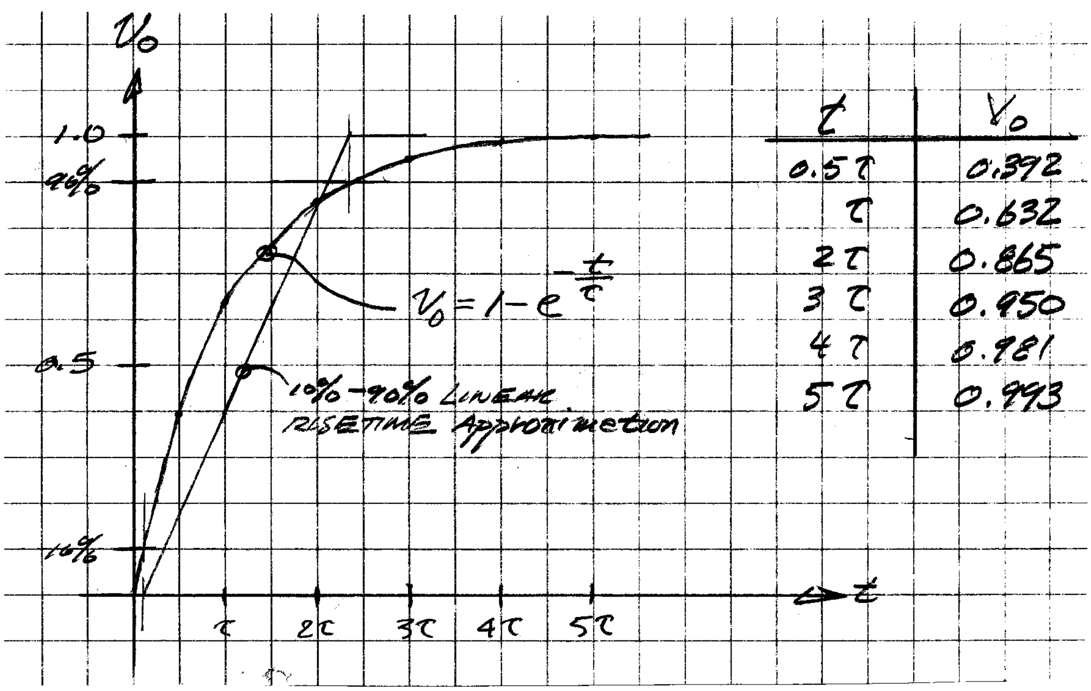


Fig.2. Plot of response to a unit step input.

Sinusoidal analysis

If we want to use an oscilloscope to view digital rise and fall signals, it must be fast enough to show them. The "speed" of an oscilloscope is rated by its –3dB analog bandwidth. This is defined as the frequency of a sinusoidal voltage that will display with its magnitude decreased by $1/\sqrt{2}$. Although this is an analog figure of merit, we can easily relate it to digital risetime.

Under sinusoidal excitation, the RC circuit has a –3dB corner low-pass frequency f_c and a magnitude response given by

$$\left|\frac{v_2(j\omega_c)}{v_1(j\omega_c)}\right| = \frac{1}{\sqrt{2}} = \frac{1}{|j\omega_c\tau+1|} = \frac{1}{\sqrt{1-(\omega_c\tau)^2}} .$$

It follows that

$$\tau = \frac{1}{2\pi f_c} = 0.1592\frac{1}{f_c} .$$

This relation is very useful for finding the corner frequency knowing the time-constant as well. The question at this point though is how do we relate τ to transition time when all we know is the low-pass corner frequency? From the graph shown above in fig. 2, it should appear reasonable to choose a linearized approximation to be somewhere between τ and 3τ . Calculating the time it takes to change from 10% to 90% (a universal common definition of rise or fall-time noted earlier) of the maximum voltage, we obtain $10\% \Rightarrow 0.1054\tau$ and $90\% \Rightarrow 2.305\tau$, so

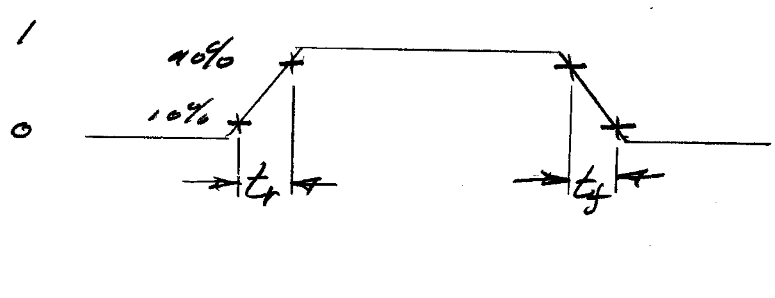
$$\Delta\tau(10\%-90\%) = 2.305\tau - 0.1054\tau = 2.2\tau \text{ or } \frac{0.35}{f_c}$$

This linear risetime approximation result is also shown on fig. 2. Moreover, 35% of the inverse bandwidth is often given as the minimum transition that can actually be seen on an oscilloscope with maximum analog bandwidth f_c . This means, for example, that an oscilloscope having a 100 MHz bandwidth could be used to display signals changing no faster than about 3.4 ns, whereas a 40 MHz scope could display rise or fall times no faster than about 9 ns minimum.

For our purpose, what this discussion establishes is that as the effective total load capacity connected to the output of a digital device increases transition times will also increase. This is why maximum rise and fall times are always specified at some particular load capacity; typically 30 pF. When the load is less than this, the device will actually switch faster. Conversely, if the load capacity exceeds the test value, the device will switch slower than the specification. Students should be aware of the fact that simply placing a scope probe on an output pin will typically increase the loading by 8 to 15pF and therefore modify the actual rise or fall time being observed.

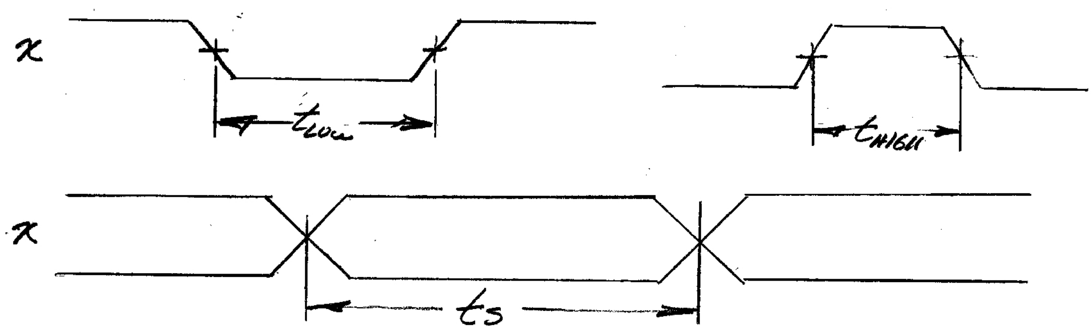
Example:
Calculate the rise and fall times for a digital gate having an output resistance of 100 Ohms connected to an equivalent load capacity of 30 pF. Plot and label these transitions on a timing diagram.

Time constant $\tau = RC = 100(30x10^{-12}) = 3ns$, so $t_r = 2.2\tau = 6.6ns$



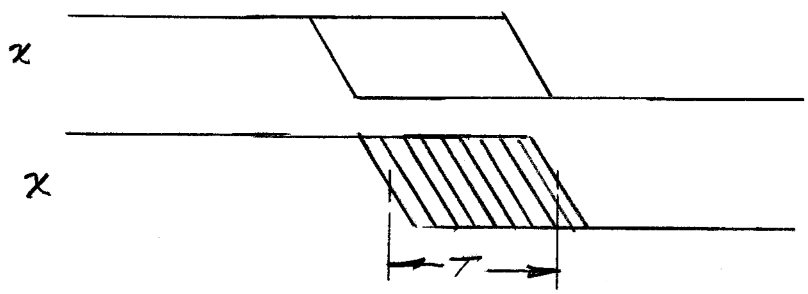
Note that rise and fall times need not be the same, since the internal resistance of the output circuit can, and often is, different for the two cases.

Besides the 10%-90% (or other %) points, the center 50% point is also often used. Two examples are shown below.

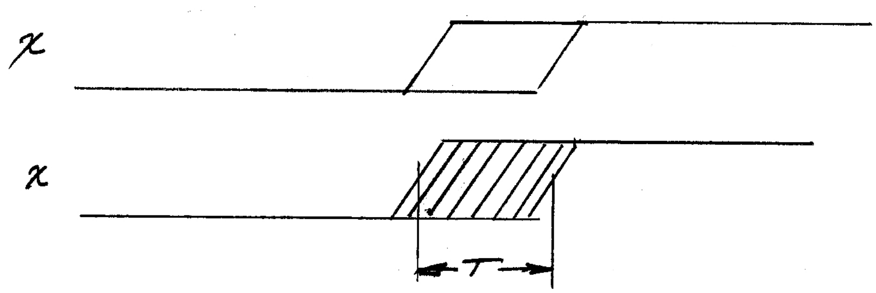


Common Graphic Conventions in Timing Diagrams

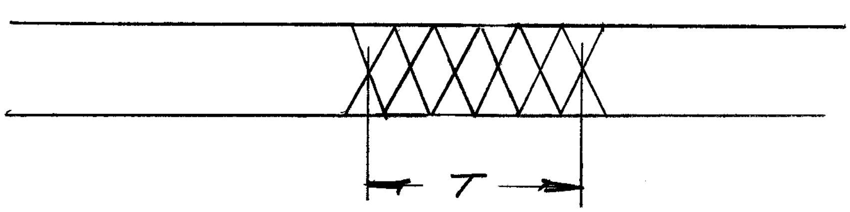
Low-going transitions. Prior to transition interval T, x is high. Sometime during T it goes low.



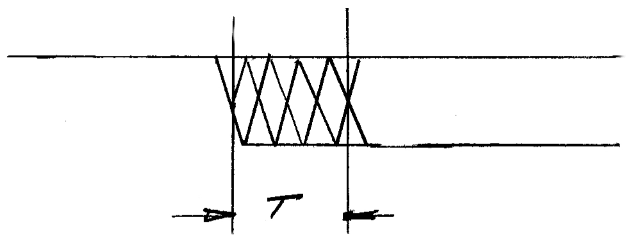
High-going transitions. Same as above but the logic is reversed.



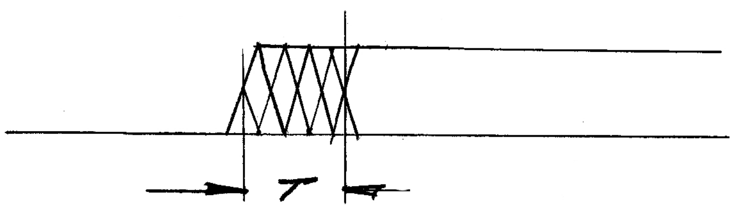
High or low-going transitions. Combining the two earlier examples, we can show a stable logic state prior to a transition interval T during which it may or may not change, followed finally by a stable logic state.



High transition to either high or low.



Low transition to either high or low.



Hi-impedance or Tri-State. Tri-state is not a logical state, but an electrical "state" of disconnection. Without it we couldn't implement digital buses. This condition of disconnection is easily modeled by a simple switch S_1 whose open or closed condition is itself under digital control S. When S is closed y follows x, but when S is open y appears to be an open circuit or high impedance. As shown, a mid-line depicts this condition of disconnection at output y.