Table 1. Characteristics of TTL Logic Families

Parameter	74	74H	74L	74S	74LS	74FCT
Delay (ns)	10	6	33	3	10	3
Power (mW)	2	22	1	19	2	0.5
Input current (mA)	1.6	2	0.18	2	0.36	0.01
Output current (mA)	16	20	3.6	20	8	48
Dc fanout	10	10	20	10	22	4800

TRANSISTOR–TRANSISTOR LOGIC

Transistor–transistor logic (TTL) is a family of electronic logic used in the construction of computers and other digital systems. It is based on the characteristics of bipolar transistor integrated circuits, and it is well suited to the manufacture of silicon integrated circuits. The TTL family was one of the first series of integrated circuits to achieve high-volume production. TTL has been the prevailing logic standard since its introduction by Texas Instruments (TI) in the mid 1960s.

Need and Growth: The Space Race

TTL logic provided major improvements in speed, power, size, and reliability over previous technologies. This was particularly important in the mid 1960s during the space race. This period, between the Sputnik satellite launch by the USSR in 1957 and the landing on the moon by the United States in 1969, became a race between the two countries to be the first to land on the moon. It was a serious race because it had military implications. A key technology, missile guidance, was the same as for intercontinental ballistic missiles (ICBMs). Missile guidance and control requires precision. The rocket must closely follow its required path at very high speeds to be usable. The computers and logic to provide this control must combine high speed of measurement and calculation with small size, low power, and high reliability. TTL provided a significant improvement to this technology.

TTL as a Logic Standard

The TTL logic family defined an electrical specification for digital logic. Logic parts that met this specification were called *TTL-compatible*. By the 1970s, almost all digital components were TTL-compatible. This included all logic parts, memories, and microprocessors. Those that were not TTL-compatible had interface parts so they could be designed into TTL-based systems.

TTL Logic Families

TTL consists of a set of families of compatible logic parts. The first edition of the *TTL Data Book* from TI in 1973 (1) already describes five TTL families.

Logic devices can be characterized by their delay and the power required per gate. Delay and power can be traded, to some degree. To achieve a given speed, a gate requires a certain minimum amount of power. Decreasing the delay requires more power, and decreasing the power will increase the delay. TTL was first introduced as three families, each designed for different combinations of speed and power per gate. These were the 74 family, the 74L family, and the 74H family. The 74 had standard speed and power, the 74L low power, and the 74H high speed. The 74 family combined 12 ns typical delay with 10 mW per gate. The 74L family combined 35 ns delay with 1 mW per gate, and the 74H family combined 6 ns delay with 22 mW per gate.

The number of TTL families increased with time as processes improved and new circuit techniques were introduced. Examples include 74S, 74LS, 74AS, 74ALS, and 74F. Recent examples based on CMOS technology include 74C, 74HCT, 74ACT, and 74FCT. Each new family offered a better tradeoff of speed versus power.

Table 1 lists key characteristics of a few of these families for comparison, and Buchanan discusses TTL families in Ref. 2.

TTL Logic Part Numbering Standard

In addition to constituting an electrical standard for logic, TTL led to a de facto part numbering standard for TTL-compatible logic parts. TI, the primary vendor of TTL logic parts, introduced the standard. An example TTL part number is 74H00. The 74 indicates the commercial temperature range, the H indicates the specific TTL family, and the 00 indicates a specific part—a quad two-input NAND gate. TI defined most of the initial parts of the TTL families, and its part numbering scheme quickly became the standard. All 7400 parts, of whatever family, are interchangeable: they have the same function, package types and electrical connection pattern. Data books from various TTL vendors (1,3–6) show the wide use of this numbering scheme. Also see Wakerly (7) and Buchanan (2).

Military and Commercial Grades

Each family has two classes of parts defined by their temperature range: military and commercial. The military parts have a 54 prefix and are guaranteed to work from −55°C to +125°C. Commercial parts have a 74 prefix and are guaranteed from 0°C to 70°C. In addition to guaranteed operation over a wider temperature range, military parts have additional testing for reliability. They are subjected to *burn-in,* which means that the parts are operated at high temperature for a period of time, typically 168 h, and then tested. Burn-in increases reliability in that marginal parts typically fail in the first few hours of operation. Parts that have survived burn-in are more reliable than equivalent parts that have not. These requirements were codified by the US military in the MIL-M-38510 and MIL-STD-883 procurement specifications. These are presented in Ref. 1.

TTL Utility: The Basic Logic Gate Concept

When they were introduced, TTL offered logic gates that were smaller, faster, lower-powered, and more reliable than previous technologies. TTL sold against discrete gates and other

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering.* Copyright © 1999 John Wiley & Sons, Inc.

logic families such as diode transistor logic (DTL), but quickly displaced the other logic parts because of the nature of digital, or Boolean, logic. Computers and other digital systems are composed of Boolean logic and memory elements. Gates implement the logic elements, and latches implement the memory elements. All possible Boolean logic functions can be created by combinations of one basic gate: the NAND gate or the NOR gate. The NAND gate is a combination of a logical AND and an inverter. The NOR gate is a combination of a logical OR and an inverter. Although equivalent, the NAND is favored over the NOR in TTL logic for secondary considerations of speed and historical familiarity. Latch memory elements can be composed of four NAND gates. As a result, any logic system can be designed using combinations of NAND gates, as discussed by Wakerly in Ref. 7 and Buchanan in Ref. 2. Digital systems prior to TTL were typically built using NAND gates made from discrete components. TTL offered a smaller, faster, lower-powered, more reliable NAND gate. Since any logic function could be built with combinations of NAND gates, designers could design systems using TTL as soon as a few basic NAND gate types were available.

The TTL Silicon Integrated Circuit Revolution

TTL entered a market prepared for it. Prior to 1963, digital logic was made from combinations of individually mounted and connected, or *discrete,* components such as resistors, capacitors, diodes, and transistors. Each company made its own logic, so there was no prevailing standard. However, most companies made similar gates that combined silicon diodes and germanium transistors. Silicon transistors were not readily available nor applicable at the time. The resulting gate was similar in characteristics to the coming TTL gates except for signal polarity. The best germanium transistors were of the *pnp* type that used negative voltages, rather than the *npn* type using positive voltages favored by silicon. Since the gates were similar except for voltage polarity, existing designs could be easily converted to TTL logic.

TTL brought several immediate advantages to such a conversion. Gates built from discrete components and germanium transistors were relatively large, slow, and sensitive to temperature. A typical germanium-based design in 1963 using the popular 2N1300-series transistors from TI and others occupied 25 cm^2 (2 $in.^2$) of printed circuit board space and had a logic delay of 1 μs. This was very acceptable, because these gates replaced mechanical-relay-based logic that took much more space and had delays of several thousand microseconds. High- speed gates using 2N964 transistors from Motorola required the same printed circuit board area but had a delay of 0.10 μs, or 100 ns. See Ref. 8 for a discussion of discrete-component gate design.

The first TTL logic gate, the SN7400, combined four two-input NAND gates in a single package that occupied less than 3 cm^2 (0.5 $in.^2$) of board space and had a logic delay of 12 ns. The result was a factor-of-16 improvement in printed circuit board space and a factor-of-8 improvement in speed for the same function.

Silicon and the Printed Circuit Revolution

TTL entered the market just as the electronics industry was converting to automated manufacturing, and it benefited from this timing. Prior to 1960, electronics was hand built. Individual components such as resistors and capacitors were soldered to terminal strips and vacuum tube sockets. The strips and sockets were wired together manually, using wires cut to length and soldered, in the same manner as other electrical wiring. Electronic wiring was revolutionized by the introduction of the printed circuit board, which automated the wiring process. A printed circuit board is a sheet of fiberglass or plastic, typically 1.6 mm ($\frac{1}{16}$ in.) thick, with thin sheets of copper bonded to either side of it. Ink patterns are printed on the copper, and the board is immersed in acid to etch away the exposed copper. Thus the name "printed circuit board." Only the copper that is covered by the ink remains. It becomes the copper wiring that interconnects electronic components installed on the board. Holes are drilled in the board, and electronic components are installed on the board by inserting their leads into the holes in the board. This insertion can be done by machine. The components are soldered to the copper patterns using a wave soldering machine that immerses one side of the board in liquid solder for a short period of time. This solders all the components at once. The result is a board full of completely interconnected electronic components made by automated processes: the wiring is fixed by printing, and the components are inserted and batch-soldered by machine. Wakerly (7) describes this in more detail.

The printed circuit board revolutionized the electronic industry because it radically reduced the labor content and made the product cost independent of the complexity of the wiring. It allowed high-volume, low-cost electronic products. The first example of this was the humble transistor radio. These radios could be manufactured cheaply in the millions because the printed circuit board allowed them to be made almost as easily as printing a book.

Printed circuit technology also makes a qualitative change. Previously, increasing the circuit complexity increased the expense because each additional wire represented a labor expense to install it. With printed circuit and integrated circuit technology, the cost of a system is essentially independent of its complexity. It is only a function of the cost of the components and printed circuit boards themselves. The cost to assemble and wire the components on the printed circuit card is small in relation to the component cost.

Integrated circuits such as TTL extended the concept of manufacturing by printing to printing wires and transistors on silicon. The cost of such a silicon device is determined by the number of square millimeters of silicon required to manufacture it, not by the number of transistors or the complexity of wiring on the silicon. Wakerly (7) discusses the relation between printed circuit and integrated circuit technology.

The combination of printed circuit boards and integrated circuits proved a powerful economic engine. Customers buy complexity. A more complex and powerful computer is worth more money than a less powerful one. However, printed circuit technology makes the cost independent of complexity. The result is a race to make money by increasing complexity, since a more complex device will sell for more money but cost the same amount to make as a less complex one.

Incentives to Convert to TTL

The improvement in printed circuit board space was an effective incentive to buy TTL instead of existing discrete-component logic. One of the selling features of TTL when it was

introduced was that it provided two flip-flops in a single package. A flip-flop is composed of two latches, requiring eight or more NAND gates to implement, as described by Wakerly in Ref. 7. The flip-flop is important because it is the most commonly used memory element in digital system design. Two flip-flops in a single package were equivalent to 16 gates and represented significant improvement in size over existing logic. Each TTL gate was 16 times smaller than a gate made from discrete components. The dual flip-flop thus represented a total size reduction of 256. This size advantage meant a cost saving, because fewer printed circuit boards were required to implement a design. It also provided savings in testing, because the TTL gates and flip-flops were pretested by the TTL vendor. Discrete-component logic had to be tested by the manufacturer, then combined into a system and tested again.

Speed improvement was another incentive to buy TTL. Faster systems sold for more money. Also, higher speed could often be used by designers to reduce cost by using sequential methods in place of direct methods. In a sequential method, a small amount of logic is used to implement a larger, more complex function by doing the logic as a sequence of small steps. Each step in the sequence requires a small amount of logic, and this same logic is used for all steps. For example, a large logic function could be implemented in ten steps by a small amount of logic perhaps one-tenth the size of the original logic. This is a powerful method. Sequential logic is the basis for the utility of digital computers: they can implement any mathematical function, however complex, as a sequence of instructions using one set of logic. If you introduce a logic family such as TTL that is 10 times as fast as the logic it replaces, you can often choose whether it is to be ten times as fast, or have one-tenth the cost, or any point in between. See Wakerly (7) for a discussion of sequential logic design.

Silicon-based TTL also provided significant improvement in temperature sensitivity, which was very important in the military and aerospace applications driving the market at the time. Logic designs at that time were based on germanium transistors, and germanium transistors do not work at high temperatures. All transistors have a maximum usable temperature. Joyce and Clarke discuss these limits in Ref. 9. Germanium transistors are typically limited to 60°C or less. Silicon transistors will work reliably up to 125°C. Military and aerospace hardware must be able to work reliably at high temperatures, exceeding 100°C in some cases. Silicon transistors in general and TTL logic in particular provided this capability.

TTL integrated circuits also provided a significant increase in product reliability over prior technologies. Integrated circuits increased equipment reliability because they reduced the number of components by providing more function in a single component. The mean time between failures (MTBF) is proportional to the failure rate of the components in the system and inversely proportional to the number of components. Wakerly (7) discusses this in more detail. Reliability is particularly important in military and aerospace applications, where a failure in the field can be prohibitively expensive. It is important in commercial equipment for similar reasons: field repair of failed equipment is very expensive, both in actual expenditure and in reputation.

TTL as Unit Logic

TTL introduced standardization in logic design. It introduced the concept of unit logic, logic that could be easily combined to form complex systems. It had good characteristics as a unit logic:

- Unit delay
- Unit loads
- High fanout and fan-in
- The ability to combine AND and OR functions in the same circuit

TTL also created an electrical interface standard for logic. Parts from different families and from different vendors could be used together with confidence.

Unit delay means that all TTL gates have similar delay, and this delay is independent of how the gate is used, to a first approximation. With unit delays, you do not have to consider the electrical characteristics of the gates or the logic signals to design a logic system.

Unit loads mean that each gate input puts a single unit load on the gate output that is driving it. Each gate output is rated at 10 loads, for all standard gates and logic functions. This is called the *fanout* of the gate. The unit load concept makes it is easy to calculate whether a gate output rating is being exceeded: you merely add up the unit loads of the gates the output is driving. TTL provided a relatively high fanout of 10 loads, meaning that one gate output can drive 10 other gate inputs. This means that you can assemble large networks of gates without having to worry excessively about overloading any of the gates.

High fan-in means that you can have a large number of inputs on a single gate. The TTL 7430 has eight inputs, and TTL is capable of many more. High fan-in reduces the gate count. In many designs, you need many inputs to a single gate. High fan-in capability means that you can do this with one gate. If you do not have high fan-in, you must create a complex of gates instead, using several gates instead of one to achieve the same result. Additional gates mean additional components, power, and delay, so high fan-in capability offers obvious advantages.

TTL also has the ability to combine AND and OR functions in a single gate. This capability is similar to high fan-in: it allows you to implement a logic function in one gate that would otherwise require several. An example is the 7453. This part combines four AND gates, a four-input OR gate, and an inverter in a single part. This single AND–OR–invert (AOI) gate would otherwise require five NAND gates to implement. Wakerly (7) discusses these capabilities in more detail.

Competition: RTL, DTL, ECL

TTL was not the first silicon integrated circuit logic family, but it became the most popular. The competition was RTL, ECL, and DTL.

RTL, or resistor–transistor logic, was introduced by Fairchild Semiconductor as the first production silicon integrated circuit logic family. It was simple in form, but had significant limitations, as described by Rabaey (10). It was slow, had limited fanout, and was sensitive to temperature and electrical noise. In addition, the speed and noise sensitivity of RTL were degraded when you increased the number of output loads. Compared to germanium discrete logic, it was better in speed and density but worse in fanout, loading, and noise sensitivity.

ECL, or emitter-coupled logic, was based on the linear differential amplifier, as described by Rabaey (10). It was introduced by Motorola Semiconductor and was very fast—significantly faster than other families, including TTL. Its disadvantages were that it was different in form and signal levels from preexisting logic, it used small signals and was therefore sensitive to electrical noise, and it was relatively power-hungry, requiring more power per gate than other logic forms. Its high speed and noise sensitivity required more careful design than other logic families. This limited its usefulness for general-purpose designs, which in turn limited its market penetration. However, its speed superiority made it popular for high-performance applications such as very high-speed supercomputers. Its use in the IBM System/360 Model 91 is described in Ref. 11, for example.

DTL, or diode–transistor logic, was very similar to TTL in design and performance. The Fairchild 930 series of DTL actually preceded TTL, and TTL is effectively an improvement on DTL. The characteristics and performance of the two families are similar. However, TTL has better speed, fanout, and manufacturing costs for equivalent manufacturing technologies. TTL could directly replace DTL in most cases with increased performance.

TTL Evolution and Levels of Integration: SSI, MSI, LSI, VLSI

After its introduction in the mid 1960s, TTL rapidly evolved. The first edition of the *TTL Data Book* (1), published by TI in 1973, contained five families with over 100 different part types. The growth pattern was simple but powerful. New parts were introduced that combined increasing numbers of gates into useful logic modules. These new parts became classified into groups defined by their equivalent gate count. *Small-scale integration* (SSI) comprised logic that combined up to 100 gates in a single package. *Medium-scale integration* (MSI) referred to parts that combined 100 to 1000 equivalent gates. *Large-scale integration* (LSI) referred to parts between 1000 and 10,000 gates, and *very large-scale integration* (VLSI) referred to parts with above 10,000 equivalent gates. These classifications are not exact, but provide a general guideline for discussing part complexity.

The SSI and MSI categories were useful for expressing complexity in the early days of TTL. However, at the LSI level of 1000 gates and above, specific part types emerged, and these parts became known more by their type and characteristics than by the term LSI or VLSI. Examples include bit-slice *register and arithmetic–logic units* (RALUs), programmable read-only memories (PROMs), random access memories (RAMs), first-in–first-out (FIFO) buffer memories, universal asynchronous receiver transmitters (UARTs), and the most significant logic part, the microprocessor.

TTL Continued Evolution: CMOS TTL

The evolution of TTL continues to the present. One significant improvement was the introduction of high-speed TTL families based on complementary metal oxide semiconductor (CMOS) technology. In 1985, Integrated Device Technology introduced a new form of TTL-compatible logic, FCT, as described in Ref. 5. This logic family was based on CMOS technology, and it had two distinct advantages: it was much faster and consumed much less power than previous of TTL families based on bipolar technology. Buchanan compares CMOS and bipolar TTL families in Ref. 2.

CMOS-based logic was not new. The 4000 family of CMOS logic was developed by RCA in the early 1970s as a medium-speed (50 ns to 200 ns) but very low-power logic family. TTL-compatible versions were introduced by other vendors, notably the 74C series by National Semiconductor. CMOS logic had the advantage of very low power, but it was also slow and had limited ability to drive standard TTL logic families. It was popular in industrial and aerospace applications, but it was not as widely used as TTL because of its low speed.

The high speed of the FCT family overcame this objection, with the result that new families of TTL are based on fast CMOS technologies. The change to CMOS occurred because the speed of CMOS technology improved more rapidly than that of bipolar technology. See Rabaey (10) and Mead and Conway (12) for discussions of these technologies. FCT represented the point where CMOS passed bipolar technology. CMOS technology has been so successful at implementing high-speed logic that almost all new TTL families are based on it. Indeed, CMOS technology has almost totally displaced bipolar technology in all logic parts. For example, all new microprocessors and memories are designed in CMOS technology.

Competition from Programmable Logic: PLDs and Logic Gate Arrays

As TTL has evolved, it has also encountered competition from programmable logic and logic gate arrays. A *programmable logic device* (PLD) is an array of TTL gates that is effectively wired after it is manufactured and shipped to the customer. The first widely accepted PLD was introduced by Monolithic Memories (MMI) in 1976 as the *programmable array logic* (PAL) device. A PAL consists of an array of programmable AND gates with a fixed array of OR gates that is fully interconnected by electrical fuses, as described by Birkner (13) and Burton (14). The PAL is wired by using a programming device to blow all the fuses except the ones desired. The remaining fuse pattern defines the interconnections of the gates. Many other manufacturers copied MMI and made similar devices. "PAL" was trademarked by MMI, so "PLD" was adopted to refer to these devices. Since the PAL is programmed in the field (i.e., at the customer rather than at the manufacturer), the full name for these devices is *field-programmable logic devices,* or FPLDs. The advantages of the PLD are that you can create just the custom TTL device you need in a single integrated circuit and that you can change the wiring of the gates without having to redesign the printed circuit board, as described by Bolton in (15). PLDs have a standard architecture of programmable AND gates and fixed OR gates, and provide integration at the SSI and MSI levels. *Complex PLDs,* or CPLDs, provide integration at the MSI level. A CPLD is a collection of PLDs on a single chip with additional fuses to provide wiring between the PLDs.

A *gate array* is an array of logic gates on an integrated circuit that are connected together by user-defined metal wiring as almost the last step in the manufacturing process. The metal wiring is defined by one of the 12 to 20 masks used in the manufacturing process. Since only one mask is custom to a particular design, the development cost of this integrated circuit is much lower than for a complete integrated circuit.

For example, a custom integrated circuit may cost $250,000 to $500,000 in development costs, whereas a gate array may cost $10,000 to $50,000. The result is a custom integrated circuit with most of the cost advantages of a standard integrated circuit. Wakerly discusses this in Ref. 7. Gate array capability as a technology became available with the first integrated circuits, but gate arrays as a formal technology emerged only in the early 1980s. Gate arrays allow a relatively free form of gate interconnect, using the programmable metal layer, and provide integration at the LSI and VLSI levels.

Field-programmable gate arrays, or FPGAs, are a hybrid of the PLD and the gate array. FPGAs were introduced by Xilinx in the late 1980s. They attempt to combine the free form of interconnect of the gate array with the field programmability of PLDs, with varying degrees of success and economic tradeoffs. FPGAs provide integration at the LSI and VLSI levels.

PLD, logic gate array, and FPGA devices compete directly with traditional TTL logic design. Traditional TTL logic design uses the printed circuit board as the user-programmable element to connect TTL-family SSI, MSI, and LSI integrated circuits together. The PLD, logic gate array, and FPGA devices simply take TTL integration to its logical conclusion: putting the whole logic design on a single chip to maximize the benefits of VLSI integration—high speed, low power, minimum size, and low cost. Gate arrays of one million gates and larger are in production, allowing large designs to be implemented on a single chip. PLDs, gate arrays and FPGAs compete with traditional TTL logic families, but they are still TTL-compatible. A PLD, gate array, or FPGA design results in a custom, user-defined TTL VLSI integrated circuit.

TTL as Glue Logic

Today, gate arrays and other user-programmed TTL devices such as PLDs and FPGAs are used to implement the majority of TTL logic designs. However, traditional TTL logic devices remain popular for utility logic. Large VLSI devices such as gate arrays, microprocessors, and memories often require SSI and MSI support logic for interconnection. Designers use TTL devices to provide this connection: to "glue together" the VLSI devices. TTL used this way is called *glue logic.* Examples of typical TTL glue logic devices include bus drivers, decoders, multiplexers, and clock drivers. The term glue logic arises because the focus in these systems is on the VLSI components. The SSI and MSI components used for interconnect are considered in relation to their service to the VLSI components, as "glue" to connect between them. However, from a logic perspective, the SSI and VLSI components are just TTL devices with different internal functions.

Designers use glue logic to provide signal drive and to expand the connection capability of VLSI circuits. A designer will use TTL bus drivers to allow a VLSI chip to drive many other devices. The TTL bus drivers are more powerful than the drivers available on most VLSI chips, and the combination of the VLSI chip and TTL driver will be faster than the VLSI chip driving the same signals alone. Also, the power dissipation associated with driving many devices is moved from the VLSI chip to the TTL chip. This can allow the VLSI chip to drive many loads without exceeding its power limits. Alternatively, decreasing power dissipation on a VLSI chip such as a CPU can allow it to run at a higher speed for the same power dissipation.

Designers also use glue logic to expand the connection capability of VLSI chips by allowing the limited number of pins available on VLSI devices to be more effective. For example, a VLSI chip can drive one of 16 output lines directly by allocating 16 pins for this function, or it can drive four pins and rely on two external 74138 TTL decoder chips to expand a four-bit binary code into a one-of-16 code. The result is a saving of 12 pins. If you can save enough pins on an VLSI device such as a gate array, you can use a cheaper package and reduce production cost. Alternatively, you can use the saved pins for additional system functions to add value. Wakerly (7) and Buchanan (2) discuss this type of logic design.

TRANSISTOR–TRANSISTOR LOGIC CIRCUIT DESIGN

TTL is based on a particular circuit design. The characteristics of TTL logic families and the specifications for TTL compatibility derive from the original circuit design. TTL circuit design can be viewed as an improvement over prior logic circuit designs. To understand the design features of TTL circuits, we will examine a DTL gate design and compare it with an equivalent TTL gate. A DTL gate is used as the basis for comparison because it is both simple and representative of prior logic gate technology. Also, the signal levels for DTL and TTL are similar, which allowed TTL to replace DTL logic easily. Wakerly (7) and and Buchanan (2) describe the circuit design of various TTL families.

Logic Circuit Components: Diodes and Transistors

To have a more detailed understanding of how logic circuits work, we need to review how diodes and transistors work in these circuits, particularly silicon diodes and transistors. The following discussion presents conceptual models of diodes and transistors sufficient for use in understanding their activities in TTL circuits. For a more thorough and accurate understanding of junction diodes and transistors, see Rabaey (10), Middlebrook (16), Mattson (17), Neudeck (18), and Streetman (19).

Semiconductor Diodes

A diode conducts current primarily in one direction. If a voltage of 0.70 V or greater is applied across a silicon *pn* junction diode in the forward direction, a current will flow. This current will be in the milliampere range for the diodes we are considering. If a voltage is applied across a silicon diode in the reverse direction, only a small leakage current will flow, in the nanoampere range for the diodes we are considering. This leakage current will be essentially constant for any voltage we apply in the reverse direction that is less than the diode breakdown voltage. Figure 1 shows the relationship between voltage and current for a silicon *pn* junction diode.

Diode Direct Current Characteristics. Silicon *pn* junction diodes have a typical voltage drop of 0.70 V when operated at their design current and a temperature of 23°C. This voltage drop consists of 0.65 V of pure diode forward voltage drop plus 0.050 V of resistive losses. The voltage drop of a semiconductor diode changes very little with a change in current. In-

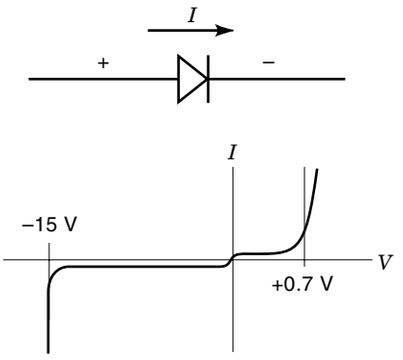

Figure 1. Relation between voltage and current for a typical silicon *pn* junction diode. The current scale is dependent on the diode size.

creasing the current by a factor of 10 will increase the voltage drop by only 60 mV, less than 10%. Likewise, decreasing the current by a factor of 10 will decrease the voltage drop by only 60 mV.

Figure 2 shows a conceptual electrical diagram of the direct current (dc) characteristics of a semiconductor diode. It consists of an ideal diode; a voltage source V_F, which represents the threshold voltage for diode forward conduction; a resistor R_S, which represents the internal ohmic resistance of the diode; and a current source I_R, which represents the reverse leakage current. These diode characteristics depend on the technology used to make the diode. Silicon diodes have a typical V_F of 0.65 V and an I_R of a few nanoamperes at room temperature. Voltage across R_S increases this to 0.7 V for typical designs. Germanium diodes have a V_F of 0.35 V and a reverse leakage of a few microamperes. Silicon Schottky diodes also have a V_F of 0.35 V and a reverse leakage of a few microamperes, even though they are based on silicon technology. Although the gross dc electrical characteristics of germanium and Schottky diodes are similar, their construction and high-frequency characteristics are very different.

Diode Temperature Characteristics. Diode voltages and currents vary with temperature. For silicon diodes, the forward voltage drop decreases with increasing temperature at the rate of −2.3 mV/°C. Likewise, the reverse voltage leakage current is an exponential function of temperature. It approximately doubles for each 10°C increase in temperature. The strong temperature dependence of reverse leakage current is the primary factor that limits the maximum temperature of operation of diodes and transistors. When the reverse leakage

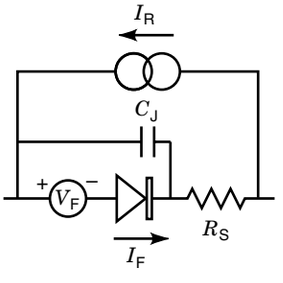

Figure 2. Simplified electrical model of a *pn* junction diode.

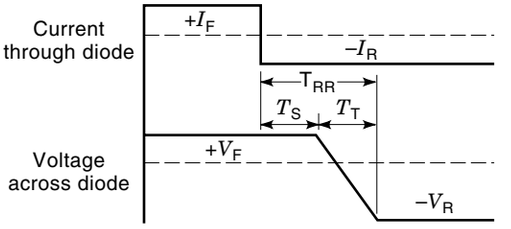

Figure 3. Timing diagram of silicon diode turnoff characteristics.

current approaches the forward current, you no longer have a diode.

Diode Alternating Current Characteristics: Switching Times. Semiconductor *pn* junction diodes have a finite turnoff time. This is called the reverse recovery time T_{RR} and is the most significant alternating current (ac) characteristic of diodes used in logic circuits. When the current through the diode is changed from forward to reverse, the diode continues to conduct current for the reverse recovery time. After this recovery time, the diode turns off, and reverse voltage quickly builds up across it. T_{RR} is defined as the time for the diode to turn off at a specified current, where the reverse current applied to turn off the diode is equal to the forward current. A typical value of T_{RR} is 4.0 ns for a 1N914 silicon diode. Figure 3 shows a timing diagram of the turnoff waveform of a diode. Note that T_{RR} consists of two components: the saturation time T_S and the transition time T_T.

Semiconductor *pn* junction diodes require some time to turn off because they build up internal charge Q_S in the semiconductor material during the forward current phase. This charge is proportional to the forward current through the diode. It keeps the diode conducting, and it must be removed before the diode can turn off. The time needed to remove this charge is the T_S component of T_{RR}. This time is thus inversely proportional to the turnoff current. Increasing the turnoff current decreases the turnoff time, and vice versa. T_{RR} is specified for a reverse current, equal to the forward current, because the stored charge is proportional to the forward current. There is an upper limit to the turnoff time, which is governed by the minority carrier lifetime of the material used to make the diode. The shorter the minority carrier lifetime, the shorter the turnoff time of the diode.

Schottky diodes, because of their different construction, do not have a significant stored-charge turnoff characteristic. They also have a lower forward voltage, 0.35 V typical, as opposed to 0.70 V for silicon *pn* junction diodes, as discussed by Rabaey (10). These characteristics are used to advantage in Schottky TTL families, as discussed by Wakerly (7) and Buchanan (2).

Semiconductor diodes also have a junction capacitance C_J. This capacitance is a nonlinear function of the voltage across the diode. The capacitance under forward bias is much larger than the capacitance under reverse bias.

Transistors

A bipolar transistor is a current amplifier. A current flow between the base and emitter terminals in the forward direction will cause a larger current to flow between the collector and emitter terminals. The ratio of collector current to base cur-

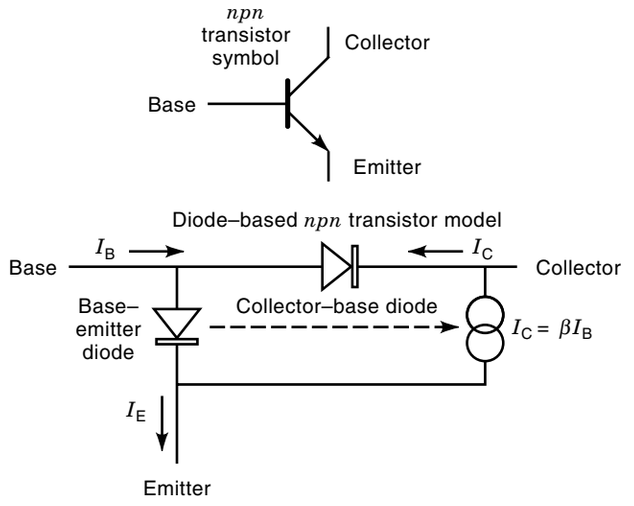

Figure 4. Simplified diode-based conceptual model of an *npn* bipolar transistor.

rent is the *current gain* β of the transistor. A typical value for β is 100.

Bipolar transistors come in two types: *npn* and *pnp*. Of these, *npn* transistors are positive-oriented: a positive current into the base will cause a positive current to flow from the collector to the emitter. When the transistor is on, the base is more positive than the emitter. When off, the collector is more positive than the base or the emitter. *pnp* transistors are similar, but with opposite polarities. A current flow out of the base will cause a current flow from the emitter to the collector. When on, the base of a *pnp* transistor is more negative than its emitter. When off, the collector is more negative than the base or the emitter. Joyce and Clarke discuss circuit design with transistors in Ref. 9.

Both *npn* and *pnp* transistors can be made of silicon or germanium. However, *pnp* transistors have the highest performance in germanium technology, and *npn* transistors have the highest performance in silicon technology. For this reason, silicon circuit design tends to favor the use of *npn* transistors and positive voltages.

Diode Model of Transistor. For our purposes, the transistor can be modeled by two diodes and a current source. Figure 4 shows a conceptual diagram of a diode-based *npn* transistor model. In this model, forward current flow in the base–emitter diode will cause a current flow between the collector and the emitter. If the base–emitter diode is reverse biased, no current will flow in the collector other than leakage current in the collector–base diode.

Note that this conceptual model is greatly simplified, specifically as an aid to understanding TTL logic circuit design. A significant amount of detail has been purposely omitted. For more thorough and accurate models, see Rabaey (10) and Mattson (17).

Transistor Direct Current Characteristics. In a transistor, the forward voltage drop of the base–emitter diode is a function of the collector current rather than the base current. This is because the transistor equations derive the base–emitter voltage as a function of the base current times β, which is equal to the collector current.

Transistor Circuits: Diode Connection. Transistors can be connected as diodes. We will find this useful in discussing TTL circuits. Figure 5 shows an example of a diode-connected transistor. This circuit is an example of a transistor used as a diode.

Forward voltage applied to the diode-connected transistor appears across the emitter–base junction. As the voltage builds up to 0.7 V, current flows between the base and emitter, causing a larger current to flow between collector and emitter. If a constant current is applied, the base–emitter forward voltage will stabilize at the point where the sum of the base and collector currents equals the applied current.

Recall that the base-to-emitter voltage of a forward-biased transistor is equivalent to the forward voltage of a diode with a forward current equal to the transistor collector current. On connecting the collector and base of a transistor together, the single collector–base terminal conducts both the collector and base currents. This results in a two-terminal device with characteristics of a *pn* junction diode. The forward voltage of the simulated diode is the base-to-emitter voltage of the transistor, and is equivalent to the forward voltage of a diode at a current equal to the collector current. Since the collector and base currents are combined, the forward voltage is equivalent to the forward voltage of a diode. The diode-connected transistor also works as a diode in the reverse direction. When the emitter–base diode is reverse biased, the transistor is turned off and no current flows in the device except leakage current in the base–emitter diode.

Transistor Circuits: Emitter Follower. Another useful transistor circuit configuration used in TTL circuits is the emitter follower. Figure 6 shows an *npn* transistor in an emitter follower circuit with a graph of the input and output voltages as a function of time for an increasing voltage ramp. In the emitter follower circuit, the output voltage V_{OUT} will be equal to the input voltage V_{IN}, minus a 0.7 V drop for the base–emitter forward voltage for V_{IN} between 0.7 V and V_{CC}.

Assume that V_{IN} is at some middle voltage between zero and V_{CC}, such as 2.7 V for V_{CC} equal to 5.0 V. Initially, current will flow from V_{IN} through the base-emitter junction and resistor R_1 to ground. The base current will cause a larger collector current between the collector and emitter, equal to β times the base current. The collector current as well as the base current will flow into R_1. The additional collector current would cause V_{OUT} to rise if the base current were constant.

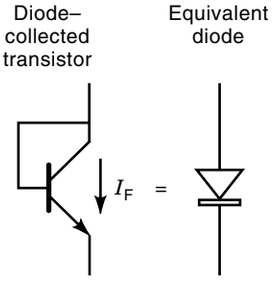

Figure 5. Diode-connected *npn* transistor circuit.

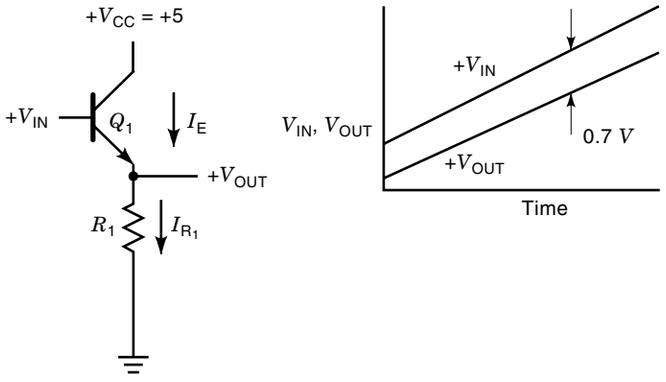

Figure 6. *npn* emitter follower transistor circuit.

However, as V_{OUT} rises for a constant V_{IN}, the base-to-emitter voltage of Q_1 will decrease. As a result, the base-to-emitter voltage will stabilize at a point where the sum of the base and collector currents just equals the current through R_1. This voltage is approximately 0.7 V for silicon transistors at their design current, and will be near that voltage over a wide range of current through R_1. This is because a small (60 mV) change in base-to-emitter voltage will cause a tenfold change in collector current. Conversely, a tenfold change in current through R_1 will cause only a 60 mV change in the base-to-emitter voltage.

The emitter follower circuit has interesting characteristics. The voltage drop between V_{IN} and V_{OUT} is approximately constant, independent of the current in R_1 over a wide range. This means that V_{OUT} will accurately follow V_{IN} with a 0.7-V drop. This is why it is called an emitter follower, because the emitter voltage follows the input voltage. If the load is a capacitor instead of a resistor, the emitter follower will supply current to the capacitor until it charges up to V_{IN} minus 0.7 V. The current decreases as the voltage rises. The output voltage continues to rise asymtotically toward V_{IN}, but at an exponentially decreasing rate. In typical digital circuits, the output voltage rises to within a few tenths of a volt of V_{IN} as determined by leakage currents in the load attached to V_{OUT}.

Transistor Circuits: Saturated Switching. The most significant use of transistors in logic is as a switch. Figure 7 shows an *npn* transistor in a switching circuit along with a plot of the output voltage as a function of input base current. In this circuit, base current supplied to transistor Q_1 causes collector current to flow through the load resistor R_L. As the base current increases from zero to its full value, the output voltage drops from V_{CC} to nearly zero.

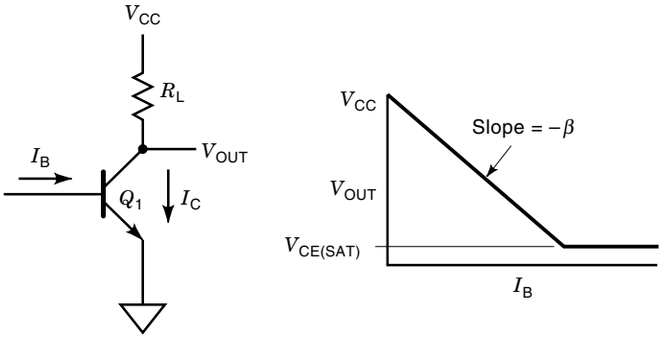

Figure 7. *npn* switching transistor circuit.

In switching circuits, base current is supplied in excess of the minimum required to bring the output to zero. This is called the *excess base current.* The excess base current does not cause a continued decrease in output voltage, because the collector–base diode in our diode-based model limits the turn-on of the transistor. If the collector voltage is more positive than the base voltage, the collector–base diode is reverse biased. It conducts only leakage current. If the collector voltage is less than the base voltage, the collector–base diode is forward biased and conducts. If the forward voltage across the collector–base diode is large enough, some of the current being supplied to the base flows through the collector–base diode to the emitter rather than through the base–emitter diode. This reduces the current into the base, reducing the current in the collector. The circuit balances itself so that the collector–base diode takes all the excess current into the base terminal above the base current required to support the collector current. The base–emitter diode current required to support the collector current is called the *active base current.* A transistor in this condition, where there is forward current flow in both the base–emitter and collector–base diodes, is said to be *saturated,* or operating in the saturated region. As a switch, it is said to be as turned on.

The difference in forward voltages between the base–emitter diode and the collector–base diode defines the collector-to-emitter voltage in the saturated condition. This difference is of the order of 100 to 200 mV for typical silicon transistors and is called the *saturation voltage* $V_{CE(SAT)}$. It is a function of the design of the transistor. The difference has several sources. For example:

1. The base–emitter forward voltage is determined by the collector current, while the collector–base forward voltage is determined only by the collector–base diode current. In a typical transistor in the saturation region, the excess base current in the collector–base diode may be $\frac{1}{20}$ of the collector current. This would yield a difference of 75 mV if the diodes were the same size.
2. The collector–base diode is effectively larger than the base–emitter diode. It has a lower current density and therefore lower voltage for the same current.
3. The base–emitter diode typically has a high series resistance, increasing its forward voltage as a result.

Transistor Temperature Sensitivity. Transistor voltages and currents vary with temperature. As in silicon diodes, the base–emitter and collector–base diode forward voltage drops decrease with increasing temperature at the rate of -2.3 mV/°C. Likewise, their reverse-voltage leakage currents are an exponential function of temperature, approximately doubling for each 10°C increase in temperature. The strong temperature dependence of the reverse leakage current is the primary factor that determines the maximum temperature of operation of diodes and transistors. When the reverse leakage current of the collector–base diode approaches the

active base current, you have a transistor that will turn on by itself.

The forward current gain β of transistors is also temperature-sensitive. It increases with increasing temperature. At low temperatures, β is also low. The degradation of β at low temperatures is the primary limiting factor that determines the minimum operating temperature of bipolar transistor circuits.

Transistor ac Characteristics: Switching Times. Transistors have finite turn-on and turnoff times. These times depend on the transistor design and the circuit that uses it. The transistor turn-on time is inversely proportional to the base drive current. Figure 8 shows a timing diagram of transistor turn-on and turnoff. In this diagram, a positive base current into transistor Q_1 causes current flow between the collector and emitter of Q_1. This current flow causes a voltage drop across resistor R_L, which causes the output voltage V_{OUT} to decrease from V_{CC} to the saturation voltage of Q_1 when Q_1 is fully on.

For a more complete discussion of transistor switching delay times, see Mattson (17).

Turn-on Time. To turn on Q_1, you drive a positive base current I_B into Q_1. This is called the *turn-on current.* The base drive current must first charge the emitter–base diode capacitance C_{EB} to the diode forward voltage for conduction before current can flow through the emitter–base junction and cause collector current to flow. The time required to do this is the *delay time* T_D in the timing diagram. The base current must then be sufficient to drive the desired collector current and to discharge the collector–base capacitance C_{CB} as the transistor turns on and the collector-to-base voltage decreases. The collector current includes both the dc component through R_L and the current to discharge the load capacitance C_L. The time required to do this is the *rise time* T_R in the timing diagram.

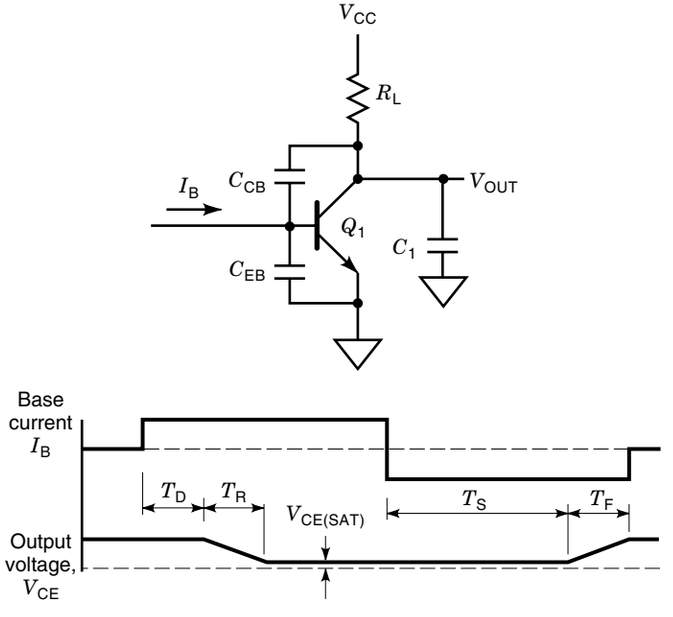

Figure 8. *npn* switching transistor timing diagram.

Forced β. The turn-on current must be larger than the dc base current required to support the dc output current. The dc output current in this example is the current through resistor R_1 when the output voltage is equal to the transistor saturation voltage of a few tenths of a volt. The required base current is this current divided by the current gain β of the transistor. The excess current drives the capacitances and controls the turn-on time. The excess current for a switching-transistor circuit design is specified indirectly by specifying the total base current for the switch, the *design turn-on current.* This current is generally much larger than the required dc base current. The ratio of the design turn-on current to the dc collector current is called the *forced* β. A typical forced β value is 10. The forced β is used in design because the actual β is a widely varying function of temperature and the manufacturing process.

Turnoff Time. To turn off Q_1, you drive negative base current from Q_1. This is called the *turn-off current.* The turn-off current removes stored charge from the base region of the transistor. In concept, it removes stored charge from the collector–base diode. The collector–base diode acquires charge from excess base current. The excess base current that sped up the turn-on of the transistor injects stored charge into the collector–base diode when the transistor is fully turned on. The turnoff current removes this charge. The time required to do this is the *storage time* T_S in the timing diagram. While the charge is being removed, the collector–base diode is still conducting and the transistor remains in saturation. Once the stored charge has been removed, the turnoff current discharges the base–emitter and collector–base diode junction capacitances, and the output rises to V_{CC}. The time required to do this is called the *fall time* T_F in the timing diagram.

Gold Doping to Reduce Turnoff Time. The transistor saturation time can be reduced by doping the base region of the transistor with gold. Gold doping reduces the minority carrier lifetime of the base region, reducing the natural turnoff time of the collector–base diode. However, reducing the minority carrier lifetime of the base region also reduces the β of the transistor. Therefore, gold doping reduces turnoff time at the expense of lower transistor β. Gold doping was used to decrease switching times in the original TTL circuits because TTL is less sensitive to β than other logic circuits such as DTL.

Schottky Transistor Turnoff Time. Transistor switches can be improved by adding a Schottky diode between the collector and the base of the transistor. Figure 9 shows a diagram of such a transistor and its switching performance. The Schottky diode D_1 has a lower forward voltage than the collector–base pn junction silicon diode. As a result, D_1 turns on before the collector–base diode. Because D_1 turns on first and supports the excess base current, the collector–base diode never turns on and builds up no stored charge. Since the Schottky diode has no significant stored charge, the stored charge switching time T_S is reduced to zero, as shown in the timing diagram. This is a significant improvement in switching transistor performance, as discussed by Streetman (19).

Note that because the Schottky diode has a lower voltage drop than the collector–base diode, the saturation voltage $V_{CE(SAT)}$ of Schottky transistors is higher than that of conven-

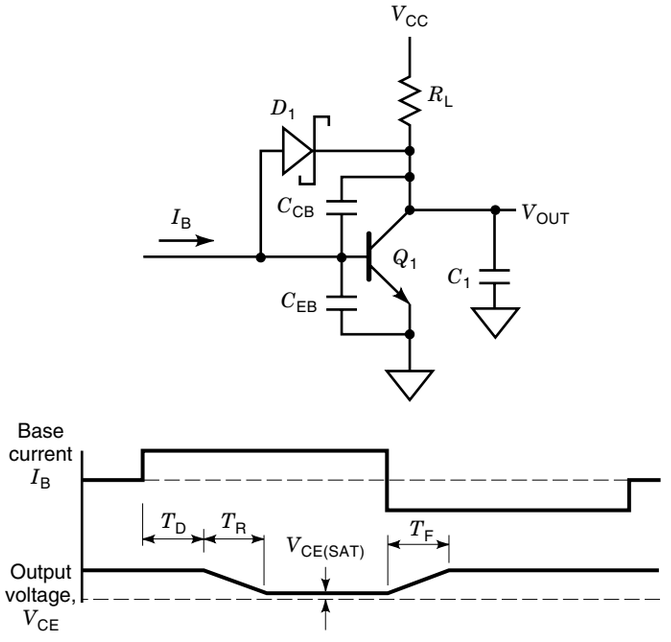

Figure 9. Schottky *npn* switching transistor timing diagram.

tional transistors. It is typically 0.35 V, versus 0.2 V for a conventional transistor. As a result, Schottky TTL circuits have maximum output voltages of 0.5 V, versus 0.4 V for a conventional output.

DTL NAND Gate Circuit

Figure 10 shows a circuit diagram of a two-input DTL NAND gate. A logic 1 is defined as a signal voltage greater than 1.4 V, and a logic 0 as a voltage less than 1.4 V. If both signals are at a logic 1, the output is at logic 0; if either input is at logic 0, the output is at logic 1.

DTL Operation: Output Low

When inputs A and B are both higher in voltage than 1.4 V, diodes D_1 and D_2 are reverse biased and conduct no current except for minor leakage current. Current flows from +6 V through R_1, D_3, and D_4 into the base of Q_1. A smaller current flows out of the base through R_2 to −6 V, and the net current into the base of Q_1 is the difference of these currents. Current flow into the base of Q_1 causes current flow between its collector and emitter. This current will flow through R_3, causing the output voltage to fall to nearly zero. In this case, Q_1 is said to be turned on.

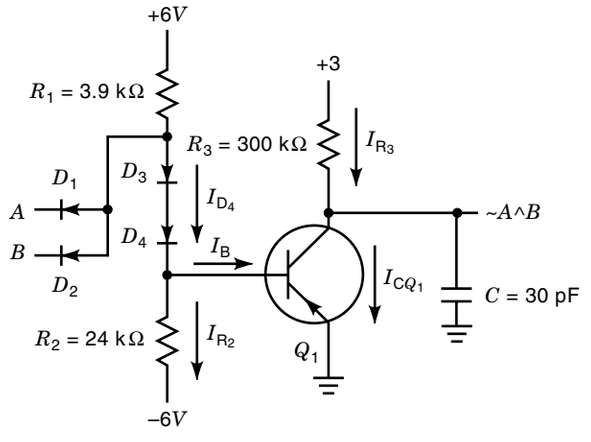

Figure 10. DTL circuit diagram.

DTL Operation: Output High

If either input A or B is lower than 1.4 V, current flows from +6 V through R_1 and D_1 or D_2, whichever input is lowest in voltage. A smaller current also flows through D_3, D_4, and R_2 to −6 V, and the current through D_1 or D_2 will be the difference of these currents. The base–emitter junction of Q_1 is reverse biased, and no base current will flow into Q_1. With no base current flowing into Q_1, no collector current flows except for minor leakage, and the output rises to +3 V. In this case, Q_1 is said to be turned off.

DTL History

The DTL circuit example shown is a typical example of discrete silicon transistor DTL gate designs. For example, it is similar to the SLT logic family designed and used by IBM in the mid to late 1960s in its System/360, 1130, and 1800 series computers. Such a gate may have a propagation delay of 25 ns or less using 1965 technology. The circuit design is similar for germanium transistor DTL designs, using a *pnp* germanium transistor, reversed diode polarities, and reversed power supply polarities.

Silicon integrated circuit DTL devices, such as the 930 DTL series by Fairchild Semiconductor in the mid 1960s, use the same basic circuit design with some modifications: they use a single 5 V supply and no negative supply, i.e., resistor R_2 is connected to ground. The forward voltage of the emitter–base diode, V_{BE}, of Q_1 across R_2 supplies the turnoff current. While Q_1 is on, it has an emitter–base voltage of approximately 0.70 V. When the base current drive to Q_1 is removed by shunting the current from R_1 to ground through D_1 or D_2, V_{BE} remains at 0.70 V while Q_1 is still conducting. It remains at approximately 0.70 V until the stored charge in the base of Q_1 has been removed and the C_{EB} and C_{CB} capacitances have been discharged. DTL silicon integrated circuits use a single 5 V supply for convenience, at the expense of higher power and degraded performance compared to the three power supply design shown.

DTL Circuit Design Characteristics

DTL circuit design centers around the transistor, as discussed in Walston and Miller (8). Two cases must be considered: when the inputs are high and the output is driven low, and when an input is low and the output is driven high. Transistor Q_1 drives the outputs low. Q_1 must drive three currents: the load current consisting of one or more DTL gates, the current in the output resistor R_3, and the current to drive the load capacitance C_L from high to low. The load capacitance is the sum of the input capacitances of the gates being driven and the stray capacitance of the traces on the printed circuit board that connect the output to the gates being driven. The 30 pF shown is a typical value.

DTL Circuit: Output Low

The circuit shown, when its input is driven to zero, requires a current I_{IL} given by

$$I_{IL} = \frac{6 - 0.7}{R_1} - \frac{6 - 0.7}{R_2} = 1.259 - 0.221 = 1.038\,\text{mA} \quad (1)$$

If 10 gates are driven, the total gate current is 10.38 mA. The load resistor R_3, requires 10 mA when the output is driven to 0 V, for a total load current of 20.38 mA.

The base current to the transistor when both inputs are high is given by

$$I_{BL} = \frac{6 - 2 \times 0.7}{R_1} - \frac{6 + 0.7}{R_2} = 1.179 - 0.280 = 0.899\,\text{mA} \quad (2)$$

The forced β for the transistor is therefore given by

$$\beta = 20.38/0.899 = 22.67 \quad (3)$$

The forced β is the minimum value of β for the transistor to meet the dc requirements for output drive. The actual β of the transistor must be significantly higher than this value for design margin and to provide excess base current for fast switching.

DTL Circuit: Output High

When either input is low, the output is driven high. In this case, the transistor is turned off, and the load resistor, R_3, supplies current to drive the gate inputs high and to charge up the load capacitance. The input diodes on the DTL gates typically have low leakage of a microampere or less, so there is little dc current to drive.

However, the load resistor must supply all the current to charge the load capacitance. The time to charge this capacitance is given by the RC time constant of the load resistor and the load capacitance. This time constant is 9.0 ns for the 300 Ω load resistor and 30 pF stray capacitance shown.

DTL Circuit: Speed Considerations

The transistor in this DTL circuit turns on faster than it turns off. This means that the propagation delay of the gate is determined by the turnoff time of the transistor and the RC time constant of the load resistor and output stray capacitance. If the transistor turns off in 14 ns and the delay provided by the RC time constant of the load resistor and load capacitance is 9 ns, for a total of 23 ns, the RC time constant of the load resistor accounts for 39% of the total delay.

The load RC time constant is an important factor in this type of simple DTL circuit design. The load resistor is made as small as possible, consistent with other tradeoffs. In the circuit shown, a separate 3 V supply is used for the load resistor. The relatively low voltage of the supply allows a small load resistor for low RC time constant while minimizing the power dissipation in the load resistor when the output is driven low.

Since turnoff of the transistor is critical, R_2 provides turnoff current to remove the stored charge in the base of the transistor during turnoff. A separate −6 V supply provides for the turnoff current drive through R_2. The high voltage of the turnoff supply makes the turnoff current nearly constant during the turnoff interval when the base-to-emitter voltage is changing as the transistor turns off.

DTL Summary: Good Points and Problems

DTL logic gates provided the following benefits of a good logic family:

1. Basic logic gates from which any logic function can be assembled
2. Unit delay per gate, allowing simple calculation of logic delays
3. Unit loads for inputs, allowing simple calculation of output loading
4. High fan-in capability to reduce gate count
5. Ability to combine AND and OR functions in a single gate
6. High fanout, allowing simple logic design without having to worry about loading

But DTL also had the following weak points:

1. High power dissipation due to the output pullup resistor when the output is low
2. Multiple power supplies versus higher output power dissipation
3. Propagation delay affected by output capacitive loading
4. High transistor β required for high fanout
5. Different manufacturing requirements for diodes and transistors

TTL: AN IMPROVEMENT ON DTL

TTL was an improvement on DTL that solved some of the problems with DTL. Figure 11 shows a diagram of a two-input TTL NAND gate. This circuit is for the 7400 NAND gate described in the TI *TTL Databook* (1). The circuit configuration of TTL is similar to DTL but with some significant differences. The TTL circuit example shown is equivalent to the 7400 TTL gate design introduced by TI in the early 1960s. This gate had a guaranteed propagation delay of 22 ns or less at 25°C.

TTL Operation: Output Low

If both A and B inputs are higher in voltage than 1.4 V, the gate output is low. If both A and B are high, the base–emitter diodes of Q_4 and Q_5 are reverse biased, and current from R_1 flows through the collector–base diodes of Q_4 and Q_5 into the base of Q_2. This base current causes Q_2 to turn on until it saturates. Its saturated collector-to-emitter voltage is approximately 0.1 V. When Q_2 is turned on, the current from R_2 in addition to the base current into Q_2 flows into the base of Q_1, less the current in R_4. This turns on Q_1 until it saturates with an output voltage of approximately 0.1 V.

Note that turning on Q_2 removes the base current drive to Q_3. In the on state, the base voltage of Q_1 is approximately 0.7 V and its collector voltage is approximately 0.1 V. The base voltage of Q_3 is approximately 0.8 V: the 0.7 V at the base of Q_1 plus the 0.1 V of saturated collector-to-emitter volt-

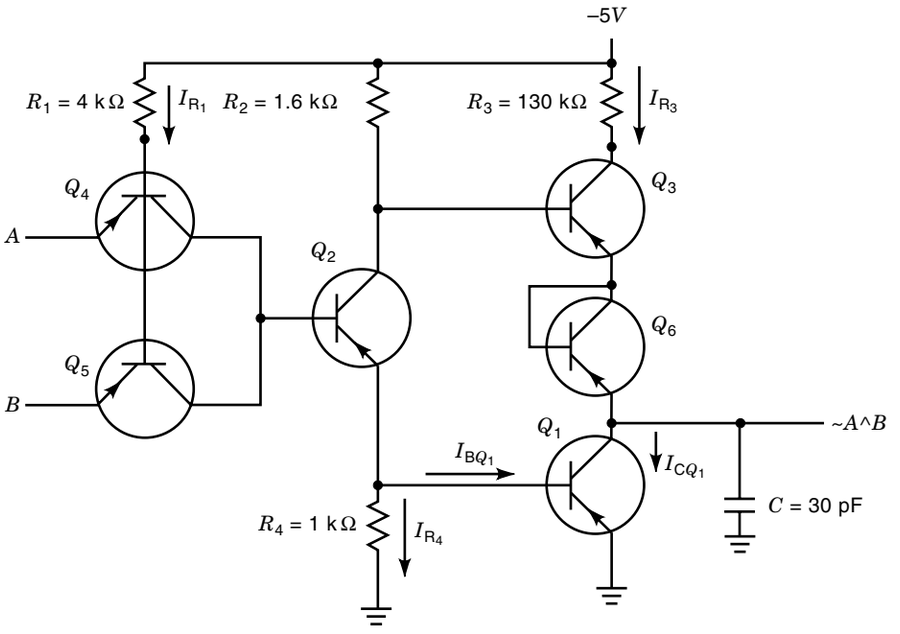

Figure 11. TTL circuit diagram.

age of Q_2. Without Q_6, the base-to-emitter voltage of Q_3 would be approximately 0.7 V, sufficient for forward bias and base-to-emitter current conduction. However, the addition of Q_6 effectively eliminates any significant current from this possible conduction path when Q_2 is on. The 0.7 V difference between the voltage at the base of Q_3 and the output voltage is now distributed between the base–emitter diode of Q_3 and that of Q_6. Assuming the voltages split evenly between the two devices, the result is 0.35 V across each of the base–emitter junctions of Q_3 and Q_6. The resulting current is a factor of $10^{-0.35/0.060}$, or approximately 10^{-6}, lower than the nominal current through these devices when they are conducting.

TTL Operation: Output High

If either input A or B is lower than 1.4 V, current flows from +5 V through R_1 and the base–emitter junction of Q_4 or Q_5, whichever input is lower in voltage. This causes Q_4 or Q_5 to turn on. The collector voltage of Q_4 and Q_5 falls to approximately 0.1 V above the lower emitter voltage of Q_4 or Q_5. Since this is also the base voltage of Q_2, it receives no drive and is turned off. Also, any leakage into the base of Q_2, such as collector-to-base leakage current of Q_2, flows through Q_4 or Q_5, whichever one is turned on. With Q_2 turned off, there is no base current drive to Q_1, and Q_1 is turned off. Resistor R_4 conducts any leakage current into the base of Q_1 to ground.

When Q_2 is off, R_2 provides base current drive to Q_3, and it conducts current. The current through Q_3 flows from +5 V through R_3, Q_3, and Q_6 to the output and into C_L. Resistor R_3 limits the current available to charge C_L. When the voltage on C_L is near zero, Q_3 turns on, connecting R_3 to C_L through the diode-connected transistor Q_6. The current into C_L charges it towards +5 V. When the output voltage is approximately 1.4 V below +5, or 3.6 V, Q_3 turns off. Specifically, it transitions from the saturated switching state to the linear emitter follower state.

In the emitter follower state, Q_3 maintains the output voltage at a minimum 3.6 V or higher. If the output voltage is slightly less than 3.6 V, current flows in Q_3 until the output rises to 3.6 V. If the output voltage is higher than 3.6 V, Q_3 is turned off, and no current flows into the output.

Emitter Follower Output Drive

The combination of Q_2 and R_3 provide an emitter follower drive. This is also called a totem pole output circuit, since Q_3 appears above Q_1 in TTL circuit schematics. In the emitter follower state, the collector–base diode of Q_3 is not forward biased, and the current in Q_3 is determined by its emitter to base voltage. If the output voltage is more than 1.4 V below +5 V, the diode-connected transistor Q_6 conducts. It operates in the same way as a silicon diode with a voltage drop of approximately 0.7 V. Subtracting the 0.7 V drop of Q_6 from 1.4 V, we have a base-to-emitter voltage of 0.7 V at Q_3, and current flows between the base and emitter of Q_3. If the base-to-emitter voltage of Q_3 is 60 mV higher than 0.7 V, the current in Q_3 increases by a factor of 10; if it is 60 mV less than 0.7 V, the current in Q_3 decreases by a factor of 10. As C_L charges and the output voltage rises above 3.6 V, the current in Q_3 drops rapidly toward zero. It stabilizes at a value equal to the output current, if any, plus any leakage current through Q_1.

TTL Improvements over DTL

TTL provided solutions to the problems of DTL mentioned at the end of the preceding section. It replaced the load resistor of DTL with an emitter follower transistor as an active pullup. This attacked problems 1, 2, and 3. When the output is high, the emitter follower provides a high current to charge the external load capacitance but will automatically reduce the current when the capacitance is charged. When the out-

put is low, only the current from the base drive resistor R_2 has to be shunted to ground to turn off the emitter follower. Also, the current through R_2 now provides the drive to the output transistor for driving the output low, rather than being unused as the current in the output resistor was in DTL. The emitter follower eliminates problem 1 by replacing the output power resistor.

The output resistor in DTL is largely responsible for imposing the speed–power tradeoff. One method of obviating this tradeoff is to use a separate, low-voltage power supply for the output resistor. This is the power supply referred to in problem 2. TTL greatly reduces this problem by using the emitter follower active pullup, effectively eliminating the need for a separate output power supply.

The low-to-high transition time of DTL is proportional to the RC time constant of the output load resistor and the capacitance of the load it drives. Increasing the capacitive loading of the output of a DTL gate significantly increases its delay, which is problem 3. The emitter follower output of TTL provides a high transient current to drive external capacitive loads in low-to-high output transitions. This significantly reduces the sensitivity of the output to capacitive loading, greatly reducing problem 3.

DTL gate design requires a high β in the output transistor. This requirement is problem 4. As shown in the DTL example, a fanout of 10 requires a minimum transistor β of 22.67. This minimum β must be exceeded over all combinations of voltage and temperature for the circuit to provide this fanout. That requirement may be difficult to meet at low voltages or low temperatures. TTL significantly reduced it by using two transistors to drive the output. In the TTL circuit, Q_2 drives Q_1 to generate the output current. The output-transistor β requirement of the TTL circuit is met by the product of the β of Q_2 and the β of Q_1. A β of 5 for Q_1 and for Q_2 results in an effective β of 25 for the pair. This greatly reduces the requirement for high β and solves problem 4.

Diodes and transistors in integrated circuits have different manufacturing requirements for best performance. This is problem 5. Manufacturing technology has imposed technical and economic limitation on integrated circuits ever since they were first introduced in the 1960s. An integrated circuit manufacturing process that creates transistors also creates resistors as a by-product. Diodes, however, are different from transistors and require different manufacturing methods for them to perform well. This requires a choice between additional manufacturing steps to combine optimum diodes and transistors on the same integrated circuit, and manufacturing diodes and/or transistors with less than optimum characteristics. TTL solved this problem by eliminating the diodes. All components are transistors or resistors, eliminating problem 5. If a diode is needed, a diode-connected transistor is used.

TTL Manufacturing Technology: The Multiemitter Transistor

TTL replaced the input diodes of DTL with transistors. The input transistors in the TTL circuit, Q_3 and Q_4 in the circuit shown in Fig. 11, are shown as individual transistors with their bases and collectors connected together. However, they are implemented in silicon as a unique structure, the multiemitter transistor. A single transistor has an emitter, base, and collector. These are arranged in vertical regions in integrated circuit design, with the collector as the bottom region, the base as the next region, and the emitter as the top region. The emitter is typically quite small for best performance. A multiemitter transistor has common collector and base regions but individual emitters. Figure 12 shows a diagram of this configuration.

The three-emitter transistor shown is equivalent to three separate transistors with their base and collector regions connected by metal wiring. The multiemitter transistor provides the same function, but in a significantly smaller area, since no special wiring is required to connect the base and collector regions. The small size of the multiemitter transistor translates directly into economic benefit for the chip: smaller size means both a larger number of integrated circuits per wafer and a higher percentage yield of these circuits, resulting in lower production cost per integrated circuit.

Additional Logic Functions in TTL: Wire OR and Open Collector Drive

DTL, both the discrete and integrated circuit versions, provides two additional circuit functions in addition to simple NAND gates. These are the wire-OR and AND–OR gates. When the outputs of two DTL gates are connected together, the common output can be taken low by either gate. If either gate is active, the common output will be low. This is called the wire-OR configuration because the output will be low if either the first or the second gate is active.

Wire-OR gates are used for *data buses* as well as simple logic. Many devices share a data bus, and any of several devices can drive it. The wire OR provides a simple connection mechanism to allow a selected device to drive the bus.

DTL gates used in a wire-OR configuration do not use internal output resistors, but use a single external output resis-

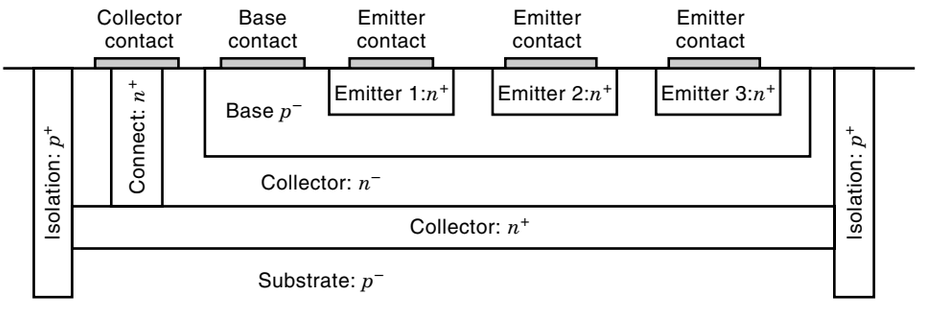

Figure 12. Multiemitter planar silicon transistor structure.

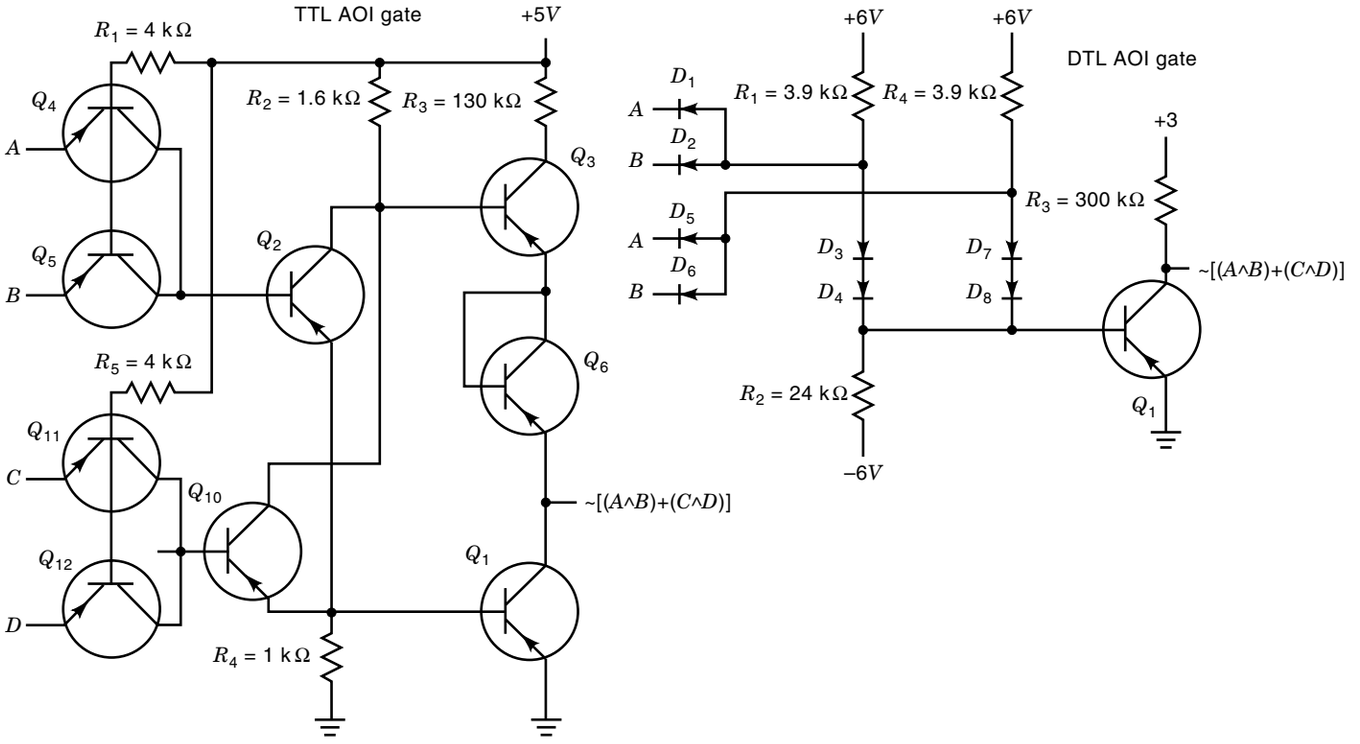

Figure 13. DTL and TTL AND–OR–invert gate circuits.

tor to which all wire-ORed gates are tied. These gates are called *open-collector output gates* because the collector of the output transistor is open, that is, has no resistor connected to it. Open-collector TTL devices are also available for wire-OR service. These devices are equivalent to a standard TTL gate with the emitter follower output driver disconnected. A TTL wire-OR system configuration is equivalent to the DTL version in that a single output resistor provides the current drive for the low-to-high transition of the bus.

Additional Logic Functions in TTL: AND–OR Gates

Both DTL and TTL have the ability to perform AND and OR functions in the same gate. This capability is used in many designs to reduce the gate count and propagation delay required to implement a logic function. Figure 13 shows equivalent AND–OR–invert functions implemented in TTL and DTL. Wakerly discusses this in Ref. 7.

In the DTL case, if A and B are both high, resistor R_2 supplies drive current to the base of Q_1 through diodes D_3 and D_4. Likewise, if C and D are both high, resistor R_4 supplies drive current to the base of Q_1 through diodes D_7 and D_8. In the TTL circuit, if inputs A and B are both high, resistor R_1 supplies current to the base of Q_2, turning it on and connecting resistor R_2 to the base of Q_1. Resistor R_2 supplies base current to transistor Q_1, turning it on. Similarly, if inputs C and D are both high, transistor Q_{10} turns on and resistor R_2 supplies current to the base of Q_1, turning it on. Each circuit provides an AND–OR–invert (AOI) function. If either A and B are both high or C and D are both high, the output will be low. The circuits shown are for two two-input AND functions and a two-way OR function. The number of AND inputs and the number of OR inputs can be expanded by adding diodes or transistors and resistors as required.

THE TTL LOGIC INTERFACE STANDARD

The popularity of TTL established it as a de facto logic interface standard, and the interface specifications for the first TTL gates, the 7400, became that standard. These specifications are given in Table 2 and discussed by Wakerly (7) and Buchanan (2). Figure 14 shows a plot of TTL gate output voltage versus input voltage with the corresponding specification points.

TTL Threshold Voltage

These specifications derive from the TTL circuit design. The input threshold voltage is typically 1.4 V, approximately equal to the sum of the base-to-emitter voltages of Q_1 and Q_2. This voltage is a function of temperature, dropping to 1.1 V at +125°C and rising to 1.7 V at −55°C, as described in Ref. 1. The input threshold voltage is the input voltage where the output is at threshold voltage, halfway between a TTL high and a TTL low. The input threshold voltage is not explicitly specified; however, the other input specifications refer to it indirectly.

Transition Region and Noise Margin

The input voltages guaranteed to be acceptable as a TTL low and TTL high are 0.8 V and 2.0 V, respectively. These volt-

Table 2. Specifications for TTL Interfaces

Parameter	Symbol	Value	Comments
Input threshold voltage[a]	V_T	1.4 V	Typical (not specified)
Input voltage for TTL low[b]	V_{IL}	+0.8 V	Maximum
Input voltage for TTL high[b]	V_{IH}	+2.0 V	Minimum
Forbidden region[b]		V_{IL} to V_{IH}	+0.8 V to +2.0 V
Transition time[a]		50 ns	Maximum, $V_{IL} \leftrightarrow V_{IH}$
Input current at TTL low	I_{IL}	1.6 mA	Maximum at 0 V in
Input current at TTL high	I_{IH}	−40 μA	Maximum at 2.4 V
Output voltage for TTL low[b]	V_{OL}	0.4 V	Maximum at 16 mA
Output voltage for TTL high[b]	V_{OH}	2.4 V	Minimum at −400 μA
Output short-circuit current	I_{OS}	55 mA	Maximum at 0 V out

[a] Not specified. The values shown are typical values expected by users and vendors.
[b] Values show define TTL compabitility.

ages provide a symmetrical 0.6 V margin between these voltages and the threshold voltage. This margin absorbs variations of the threshold voltage with temperature. The region between these voltages is the transition region, because the input signal passes through this region only during a transition from low to high or high to low. While it is in this region, it is neither a TTL high nor a TTL low.

Input signals are allowed to pass through the transition region, but not allowed to stay there. The expected transition time through this region is not specified but expected to be 50 ns or less. Long transition times can cause excess system noise and power dissipation. TTL gates are typically sensitive to small changes in their input voltage near the threshold voltage. If the input signal spends a long time near the input threshold voltage, the TTL gate can become a high-frequency noise amplifier and/or oscillator and draw excessive power while doing so.

The guaranteed TTL output voltages for a TTL high and TTL low are 2.4 V and 0.4 V, respectively. These voltages are 0.4 V further from threshold than the input levels, providing a 0.4 V guaranteed noise margin between TTL inputs and outputs. Actual TTL output voltages are typically 0.2 V for a TTL low and 3.5 V for a TTL high, providing an actual noise margin of 0.6 V for TTL low signals and 1.5 V for TTL high signals. Buchanan discusses TTL dc and ac noise margins in Ref. 2.

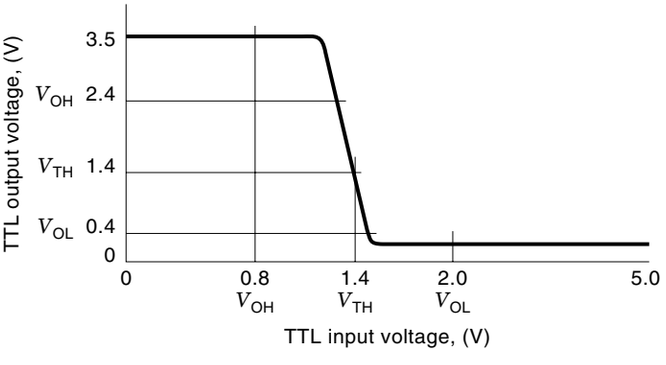

Figure 14. TTL input and output signal voltage relationships.

Note that the actual noise margin is asymmetrical, with more than twice as much margin for TTL high signals as low. This leads to the common practice in TTL system designs of selecting the TTL high level as the inactive signal level and TTL low as the active level. The result is that system noise problems are most likely to occur when you are looking at the signals, that is, when the logic is active and doing something. This makes it easier to find and fix noise problems.

Input Leakage and Inverse β

Each TTL gate draws current through its inputs in operation. If the A input in the TTL circuit diagram is at a TTL low, current from resistor R_1 flows out of the circuit through the transistor Q_4 to ground rather than into the base of Q_2. The TTL low input current is specified at 1.6 mA maximum. If the A input is high, base emitter leakage current for transistor Q_4 flows into the circuit. This current is specified as 40 μA maximum.

The input leakage current for a TTL high has an interesting source, the *reverse* β of the input transistors, such as transistor Q_4 in Fig. 11. An *npn* transistor such as Q_4 is manufactured in silicon by using alternating n and p layers to form the emitter, base, and collector. These layers are designed to provide the transistor with high performance in its normal operation, with the collector–base diode reverse biased and current flow through the emitter base diode to cause a high collector-to-emitter current. However, transistors can be used in the reverse mode, with the emitter base diode reverse biased and current flow through the collector–base diode causing current flow between the emitter and collector. This generally results in a poor transistor, with a β of much less than 1, called the reverse β. However, the reverse β is not zero. The relatively high input leakage current of 40 μA allows for the reverse β effect, where current flow through the collector–base diode of Q_4 causes a reverse β leakage current through its reverse-biased emitter–base junction.

The input transistors Q_4 and Q_5 used in the 7400 series also had low breakdown voltages, due in part to the reverse β effect. Input leakage is nominal if driven by a 7400-series gate to its nominal TTL high level of 3.5 V. Connecting an input directly to the +5 V power supply can cause excess leakage in some cases, and even burn out the input transistor for the gate. A common practice is to connect unused inputs through a resistor to +5 V. The resistor limits the current flow into the gate and prevents high leakage currents and possible input destruction. Later TTL families solved this problem and eliminated the consequent current limitation. However, this practice that the 7400 family initiated is still used in most TTL designs.

Output Specifications and Fanout

The TTL output specifications are designed for a minimum fanout of 10 gates. The output low voltage is specified at a current of 16 mA, equivalent to the input low current for 10

gates. Likewise, the output high voltage is specified at a current of 400 μA also equivalent to the input high current for 10 gates.

Dc and ac Fanout

The dc fan out is defined as the ratio of the output drive current to the input current. This is a minimum of 10 by convention. It can be much larger for devices with high output drive and low input current. For example, CMOS TTL has a dc fanout in the thousands. In practice, however, designers limit the fanout to 10 or less to prevent the gate outputs from being slowed down by excessive capacitive loading. The recommended fanout limit for speed considerations is called the ac fan out. The ac fanout is a system design guideline, and is defined as the ratio of the maximum recommended capacitance to be driven by a gate to the input capacitance of a gate. The input capacitance is given by the part specifications, but the maximum output capacitance is a system design decision to prevent speed degradation by high-capacitance loads. A 50 pF maximum output capacitance and 5 pF input capacitance results in an ac fanout of 10. Buchanan discusses this in more detail in Ref. 2.

Short-Circuit Current

The output short-circuit current, I_{OS}, is valuable for its existence more than for its actual value. The short-circuit current is defined as the maximum current that flows through the output emitter follower if the output is connected to ground. The existence of this specification implies that the output can be grounded temporarily without harming the TTL device. The specification includes the caveat that only one output should be grounded at a time, and only for a maximum of one second. However, it does guarantee that accidentally grounding the output does not destroy the device. This is a valuable practical advantage in system design and debug, where an output may be accidentally (or purposely) grounded during test.

TTL LOGIC FAMILY VARIATIONS AND IMPROVEMENTS

The TTL logic interface standard developed from the interface specifications of the first TTL devices, the 7400 family. Additional TTL families have been introduced for higher speed and/or lower power. Each family maintains TTL compatibility by using the same voltage levels for the TTL input and output signals, but each family might have gate input currents and output drive currents that differ from their 7400 equivalents. However, the output drive currents are matched to the input currents to maintain a fanout of 10 or more within the family. For example, the low-power 74L family has input currents and output drive currents that are one-fifth of their 7400 equivalents.

All true TTL families are, by definition, TTL-compatible. However, mixing families requires some care. In mixing 7400 and 74L00 gates, each 7400 gate appears as five gate loads to a 74L00 device. This limits the 74L00-to-7400 fanout to 2 for the 74L00 devices. Likewise, each 74L00 load appears as one-fifth of a load to a 7400 device, resulting in an effective 7400-to-74L00 fanout of 50. The input and output current specifications for each family must be examined to determine the loading effects when the two are connected. Buchanan discusses TTL family specifications and inter family communication in Ref. 2.

TTL Circuit Improvements: Darlington Output Drive

The 7400 circuit design was copied in most TTL families, but with variations and improvements. The 74H and following families replaced the diode-connected transistor Q_6 with a full transistor. This configuration has the emitter of Q_5 driving the base of Q_6, and Q_6 driving the output. This is called a Darlington configuration. It results in a stronger drive capability for the emitter follower output and less sensitivity of the circuit to the β's of these transistors, as shown in the circuit for the 74H00 in the *TTL Data Book* (1).

TTL Circuit Improvements: Tristate Drivers

National Semiconductor introduced the tristate driver in the early 1970s to improve the ability of TTL devices to drive buses. Until this time, buses were driven by open-collector drivers with a single pullup resistor for the bus. Buses implemented in this manner had the same problems as DTL logic: long risetime of low-to-high signals if the resistor current was small, or high power dissipation if the resistor current was large.

TTL overcame the problem of long rise time of signals in logic by using an emitter follower driver to pull up the output in low-to-high transitions. This meant that the output was actively driven for both low and high output values. However, this also meant the outputs of TTL gates could not be connected together. If one gate was driving low and the other was driving high, the result was a high current flow from the emitter follower of the high-driving device to the low-driving device.

Tristate drivers allow TTL outputs to drive buses. They provide the benefit of the fast low-to-high risetime of TTL while avoiding the high currents that flow when the outputs of TTL gates are connected together. Tristate drivers have three output states, as their name implies: high, low, and high-impedance. In the high-impedance state, both the output pulldown transistor Q_1 and the emitter follower transistor Q_5 are turned off by having their base drive current removed. The result is that no current other than leakage will flow to or from a disabled output, whether the output is driven high, low, or in between. A tristate gate in this third condition is called *disabled.* In a bus configuration, only one gate should be driving the bus at any time. Tristate drivers meet this condition by having all gates driving the bus be disabled except the selected bus driver. Tristate drivers brought the advantages of TTL outputs to buses: higher-speed active drive for low-to-high transitions, and elimination of the high dc power dissipation in an external pullup resistor. Wakerly (7) describes tristate drivers in more detail.

TTL Technology Improvements: Schottky TTL Families

The original TTL design has been the object of many improvements in manufacturing technology and circuit design. One of the first and most effective improvements was the introduction of Schottky diodes and transistors, as Wakerly describes in Ref. 7.

Schottky transistors do not have the saturation turnoff delay of conventional transistors. Schottky-transistor-based TTL designs are therefore faster than their conventional counterparts. Since Schottky transistors do not have saturation delay, they do not require gold doping to reduce this delay, and they do not suffer the resulting transistor β reduction. Schottky transistors benefit in both speed and current gain relative to the transistors used in the 7400 family. Schottky technology was introduced in the 74S and 74LS TTL families in the 1970s. Schottky TTL families quickly became the primary TTL logic standards, replacing the older 74, 74H, and 74L parts based on gold-doped technology.

The performance advantages of Schottky TTL were matched by manufacturing advantages. By a lucky coincidence, the manufacture of Schottky diodes and transistors requires no special processing steps beyond those used in conventional silicon integrated circuit manufacture. A Schottky diode is made by depositing a metal such as aluminum onto a lightly doped region of silicon. The same aluminum metal used for interconnecting the transistors on a chip was used to form the Schottky diodes. Figure 15 shows how the Schottky clamp diodes were implemented in the silicon as an extension of the metal contact to the collector.

These diodes performed well, required little space, and eliminated the gold doping step. As a result, better product yields were obtained.

The small size, high speed and ease of manufacture of the Schottky diode led to a hybrid of DTL and TTL, as implemented in the 74LS family. The 74LS family used Schottky diodes for the input AND function, replacing Q_3 and Q_4 in the original TTL circuit. However, it retained the remainder of the circuit, from Q_2 onward. The easily manufacturable Schottky diodes overcame the diode manufacturing problem of DTL. Their small size allowed them to compete successfully with transistors for the input AND gate in these designs. And by combining these diodes with a conventional TTL drive circuit, all the other advantages of the TTL circuit design were retained.

TTL Technology Improvements: CMOS TTL

In 1985, the 74FCT family based on CMOS technology began another major improvement in TTL logic. The 74FCT family was 35% faster than previous families and used very little power, as described in the Integrated Device Technology *High Performance CMOS Data Book* (5). Logic based on CMOS technology has some very valuable features. CMOS logic uses power only when it is switching, i.e., when the inputs and outputs are changing state from high to low or low to high. During the time it is not switching, it requires almost no power. For example, a 7400 device uses 4 mA when all outputs are high and 12 mA when all outputs are low. An equivalent 74FCT00 typically requires less than 100 μA for either case, 40 to 120 times less than the 7400. The result is logic that draws power only when you use it. This does not mean that CMOS logic draws no significant power. In typical high-speed systems with lots of logic activity, the power drawn by CMOS logic may be 10% of the power drawn by an equivalent part using bipolar technology, but that is still a significant saving.

Another advantage of CMOS logic is that its gates draw almost no input current, only leakage currents of a few nanoamperes. Coupled with a high-current output drive capability (48 mA to 64 mA for 74FCT logic), this results in a large dc fanout of over 1000. However, the ac fanout is still typically limited to 10 for a 50 pF design maximum for the output capacitance and a 5 pF input capacitance.

An additional advantage of CMOS logic is low sensitivity to radiation, such as encountered in space. Radiation such as gamma rays and alpha particles gradually degrades bipolar transistors by disrupting the crystal lattice they use for transistor operation. This shows up as a degradation of transistor β and an increase in leakage currents. CMOS logic is much less sensitive to these effects. Ordinary well-designed CMOS logic can withstand ten times the accumulated radiation dose of bipolar logic, and CMOS logic specially designed for radiation resistance is almost immune to it. Also, improvements to CMOS in the form of reduced size and gate oxide thickness further decrease its radiation sensitivity. As a result, CMOS logic with its low power and low sensitivity to radiation make it ideal for space applications such as satellites.

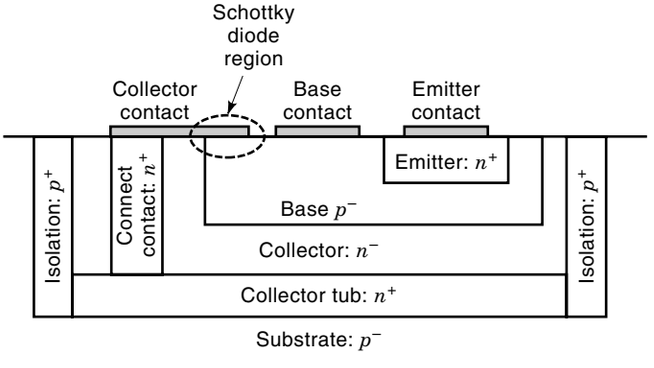

Figure 15. Schottky planar silicon transistor structure.

PERFORMANCE OF TTL FAMILIES

The performance of a TTL family is a function of its transistor technology, its circuit design (including its speed–power tradeoff point), its operating temperature, and its packaging. The design goal for a TTL family is a desired speed and power, somewhere between the maximum speed for a reasonable power and a reasonable speed for the minimum power.

Bipolar Technology and Speed–Power Product

The transistor technology and to some extent the circuit design establish the speed-versus-power potential of the family, called the speed–power tradeoff. The speed–power tradeoff for bipolar transistor technologies is captured in the speed–power product, described by Wakerly (7). This is the gate delay in nanoseconds times the power required per gate in milliwatts. The result is measured in picojoules. The best technology is the one with the lowest speed–power product, which will give the lowest delay for a given power or the lowest power for a given delay.

The speed–power product indicates that speed and power are inversely related. However, the relation is nonlinear. Increasing the power per gate increases speed only up to a point of diminishing returns. Further increases in power buy pro-

gressively smaller increments in speed up to some maximum defined by the underlying transistor technology. As a result, the speed–power product is not constant. It increases at high power levels as you approach the speed limit of the technology.

CMOS TTL Speed and Power

Bipolar transistors are current amplifiers. They require current flow into the base to cause collector current flow, so they have an implied quiescent dc power. This, coupled with the gain–bandwidth product of the transistors, defines the speed–power product of a TTL family based on them. CMOS transistors are voltage-driven devices. They are a type of voltage-driven resistor. If voltage is applied, then the resistance is low and current can flow. If no voltage is applied, then the resistance is high and very little current will flow. As a result, TTL families using CMOS technology have no significant quiescent current. This means that there is no inherent speed–power tradeoff in CMOS TTL families.

In practice, the operating current of CMOS TTL devices is not zero. It is a combination of frequency, leakage and other currents due to second-order effects. However, these currents are not a primary factor in CMOS TTL family design.

BiCMOS TTL: Bipolar–CMOS Hybrid

Bipolar and CMOS transistors can be manufactured on the same integrated circuit, at the expense of some extra process steps. Circuits that use both bipolar and CMOS devices are called BiCMOS circuits. These circuits combine the advantages of both types of transistors, as discussed by Rabaey (10). Bipolar transistors are good analog amplifiers and allow relatively precise control of signals, but they require base drive current and are somewhat slow to turn on and off because they are minority carrier devices. CMOS transistors require no standby current, and they turn on and off quickly because they are majority carrier devices; however, they are relatively poor analog amplifiers because of their low voltage gain per stage. BiCMOS designs try to combine the best features of both types: the precision signal control of bipolar devices for noise control with the high speed and low power of CMOS devices. BiCMOS TTL families have had some success in this area, notably the TI 74ABT family.

BiCMOS has some potential technical advantages but some manufacturing disadvantages. It competes directly with pure CMOS technology. A new BiCMOS TTL family can offer speed and noise advantages over prior families; however, this often leads to an equivalent family based on pure CMOS available later, at a lower price and in higher volume.

TTL Thermal Performance

TTL propagation delay varies with temperature according to its implementation technology. Propagation delay is separated into two categories: that due to a low-to-high transition of the output, and that due to a high-to-low transition of the output. TTL logic designers use the worse of the two specifications when estimating the performance of their designs.

Bipolar TTL Thermal Performance. In bipolar technology families such as 74, 74H, 74L, 74S, and 74LS, the propagation delay for a high-to-low transition of the output increases as the temperature decreases, while the low-to-high delay increases as the temperature increases. For example, the high-to-low delay of the 7400 increases from 7 ns at 25°C to 10 ns at −55°C, and the low-to-high delay increases from 10 ns at 25°C to 15 ns at 125°C, as documented in Ref. 1. Changes in propagation delay with temperature for bipolar (and BiCMOS) TTL are governed by four effects: (1) the base-to-emitter forward voltage decreases with increasing temperature, (2) the β of bipolar transistors increases with increasing temperature, (3) the storage delay of transistors increases with increasing temperature, and (4) silicon resistors increase in resistance with increasing temperature.

When the temperature decreases, the high-to-low propagation delay increases due to decreased output current from Q_1. The output current decreases primarily due to the decrease of the β of Q_1 with decreasing temperature. This is a direct effect: lower β means lower output current available to drive capacitance for a given base current. A secondary source of decrease in output drive is a decrease in base drive to Q_1 due to the increase in base emitter voltages of Q_1 through Q_4 increase with increasing temperature. A base-to-emitter voltage of 0.7 V at 25°C becomes 0.9 V at −55°C. The increase in base-to-emitter voltages of Q_1, Q_2, and Q_3 decreases the voltage developed across R_1, thus decreasing the current drive to Q_2. The voltage across R_1 will decrease by 0.6 V, from $5 - (3 \times 0.7) = 2.9$ V to $5 - (3 \times 0.9) = 2.3$ V, a decrease of 30%. Also, the voltage across R_2 will decrease by 0.2 V from 4.3 V to 4.1 V, a decrease of 5%. This decrease in voltage is somewhat offset by the decrease in resistance of the silicon resistors R_1 and R_2 with decreasing temperature.

When the temperature increases, the low-to-high propagation delay increases with increasing temperature. The increased delay has two sources: increase in resistance of the silicon resistors and increase in storage delay in the output transistor Q_1. High temperature increases the β and the storage time of Q_1. These two effects are related. A higher β means that the transistor is more sensitive to stored charge. Higher temperature means a higher resistance value for the internal base resistance of Q_1 and of R_4 that removes the stored charge. The result is a longer storage delay at high temperatures.

Temperature changes modify the low-to-high and high-to-low delays differently in bipolar TTL. The resulting delay increases with both increasing and decreasing temperature. As a compromise, TTL families are designed to have their minimum propagation delay for both delays at room temperature, 25°C, with increasing delay with deviations from this temperature.

CMOS TTL Thermal Performance. CMOS TTL devices have different temperature effects because the CMOS transistors are different. Propagation delays in CMOS devices increase uniformly with temperature. This is because the current drive of CMOS transistors decreases with increasing temperature. The delay of TTL devices in a representative CMOS process may increase by 30% as the temperature increases from 25°C to 125°C. The propagation delay also decreases with decreasing temperature due to increasing drive current. This is a mixed blessing. Decreasing propagation delay also means decreased rise and fall times, which mean increased noise and ringing of the circuit traces connected to the outputs. CMOS

device system noise problems generally occur at low temperatures.

Propagation Delay versus Power Supply Voltage

Decreasing the power supply voltage increases the propagation delay for TTL devices. Lower power supply voltage means lower internal drive currents to the transistors and corresponding lower drive currents to the external load capacitance to drive it high or low. This effect is typically small. For example, a 5% change in voltage may cause a 5% change in propagation delay. Large effects are effectively prevented. TTL families have a required power supply voltage tolerance of ±10% for the specifications to be guaranteed, and most families require ±5% tolerance. If your power supply voltage is outside these tolerances, none of your specifications are guaranteed to be valid.

Power versus Frequency

The power required by a TTL gate increases with frequency. The increased power is required to charge and discharge internal and external capacitance. The current required is the product of the capacitance, the charge and discharge voltage, and the frequency. In practice, almost all of the increase in current with frequency is due to charging and discharging the external capacitive load, even for small capacitive loads. This current is given by

$$I_{CC(AC)} = QF = V_{OH}C_L F \tag{4}$$

where

$I_{CC(AC)}$ = power supply current as a function of frequency (mA)
V_{OH} = TTL output voltage for a TTL high (V)
C_L = load capacitance (pF)
F = output frequency, low to high (MHz)

This equation gives the average current due to frequency. The peak transient currents are much higher, perhaps as much as 10 times as much. For example, a TTL gate operating at 10 MHz frequency may charge its output capacitance to V_{OH} in 10 ns, and it will do this every 100 ns. This means that the peak current is 10 times the average current.

Power Supply Decoupling Capacitors

High peak currents in the TTL device mean high peak current demands from the power supply. The power supply for the TTL device must have a low impedance so that the voltage does not drop during the peak current demand. A common and necessary practice in TTL logic design is to provide large decoupling capacitors for each of the TTL devices to supply this transient current. These capacitors supply the peak currents, preventing transient voltage loss and increase in propagation delay due to the reduced voltage. These capacitors also reduce electrical noise on the board and power supply lines by keeping the peak current surges local to the TTL device. Johnson and Graham discuss decoupling design in Ref. 20, as does Buchanan in Ref. 2.

HIGH-PERFORMANCE TTL

High-performance digital systems, such as computer systems, require high-performance TTL. The speed of the TTL logic can determine the speed of the system. Higher-speed systems are worth more than lower speed systems, so higher-speed TTL is worth more than lower-speed TTL. This provides economic incentive to maximize the TTL speed. Buchanan describes high-performance TTL system design in Ref. 2.

Clock Speed versus Propagation Delay

The performance of a digital system is determined by its clock speed in megahertz. The clock speed is defined as the reciprocal of the clock period: the shorter the clock period, the higher the clock frequency. The clock period is limited to a minimum value by the propagation delay of the gates that make up a system. A rule of thumb is that the clock period should be ten gate delays or longer. This allows up to 10 gate delays between one clock and the next. Systems with fewer delays per clock period are difficult to design, and systems with more delays per clock are easier to design but may be wasting valuable performance potential. For example, a system using 7400-series TTL with a maximum propagation delay of 22 ns per gate should have a clock period of 220 ns or longer, with a corresponding clock frequency of 4.5 MHz or less.

Elements of Propagation Delay

The speed of a TTL gate is determined by its propagation delay, which is measured as shown in Fig. 16 for a NAND gate. Propagation delay is measured from the time the input crosses 1.5 V until the time the output crosses 1.5 V. This is T_{DHL} for a high-to-low transition of the output and T_{DLH} for a low-to-high transition. The worst-case propagation delay of the gate is the larger of these two numbers.

The propagation delay has several related elements. When the input signal starts changing, there is a transition delay before the input signal crosses the 1.5 V threshold. This is T_{TR} for rising signals and T_{TF} for falling signals. Nothing happens in the gate during this transition delay, because the input has not yet changed state from a TTL low to a TTL high or vice versa. When the input crosses the 1.5 V threshold, the internal circuitry of the gate can begin to generate the corresponding output transition. It takes some time to do this, which is the internal delay T_{INT}. After the internal delay, the output begins its transition. The output transition takes some time before it crosses the 1.5 V threshold. This is T_{TR} for rising signals and T_{TF} for falling signals. The propagation delay of the gate is the sum of the internal delay and the output rise or fall transition delay, T_{TR} or T_{TF}.

Propagation Delay Limits

To decrease the propagation delay of the gate, we must decrease the various elements of the propagation delay. There are limits to how much we can do so. The most direct way to decrease the delay to decrease the internal delay T_{INT}. This delay is determined primarily by the TTL transistor technology and somewhat by the TTL circuit design. It has decreased from approximately 8 ns for 7400 devices to 2 ns or less for current TTL technologies.

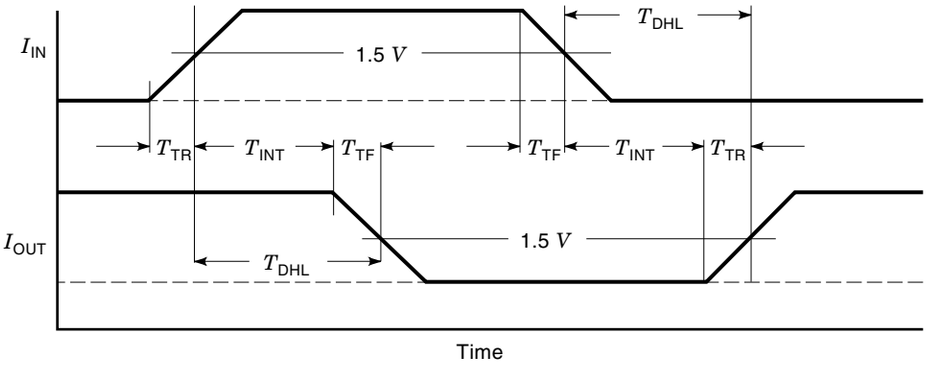

Figure 16. TTL gate timing diagram showing propagation delay elements.

Reducing the transition delays is not simple. The output transition time is defined as the time for a signal to change from a low to a high or a high to a low. More precisely, it is measured as the time between the 10% and 90% points of the transition. Output transition times must be limited to a minimum value as determined by noise conditions. Short rise and fall times cause high levels of electrical noise in printed circuit board traces due to reflection and ringing of the signals. This electrical noise causes errors in the signals if it is large enough. TTL gates with output transition times of less than 3 ns generate too much noise to be usable in most designs. A 3 ns transition time means a delay of approximately 1.5 ns for T_{TR} or T_{TF}. This limits the propagation delay of the gate to a minimum of 1.5 ns even if the internal delay is zero. Since the internal delay cannot be zero, typical delays for fast gates are in the 2.5 ns to 4 ns range.

Origin of Transition Delays

Transition delays are the result of how logic signals are supposed to affect logic gates. The conventional specification is that the output of a gate should not change state until the input has crossed the threshold voltage. With this interpretation, a gate will ignore input noise as long as it does not cross the threshold voltage. Each gate therefore reduces noise: noise at its input does not propagate to its output. It also causes transition delay because the output cannot begin to change until the input has crossed the threshold voltage.

Input Hysteresis. Another description of how logic signals are supposed to affect logic gates is that the output must not change until the input has crossed the transition region. This means that the input signal must fully cross the transition region before the gate output changes. Gates with the characteristic that the input must go past threshold in both directions before the output changes have *input hysteresis.* This is specified by the *hysteresis value,* defined as the positive-going threshold value minus the negative-going threshold value. Input hysteresis improves noise rejection because the noise must go beyond the threshold voltage before the output can change. Many gates incorporate some hysterisis for improved noise rejection. It also eliminates noise amplification and oscillation in the case where the gate input voltage drifts slowly into the transition region near the gate threshold voltage, as discussed in Wakerly (7). In this case, the noise must be larger than the hysterisis value to propagate through the gate.

Avoiding Transition Delays

Transition delays can be avoided. For example, a wire has no transition delay because its output follows its input. When its input begins to change, its output begins to change. It does not wait until the input reaches a threshold voltage before it changes its output. However, the cost of eliminating this delay is the corresponding elimination of automatic noise rejection. Two TTL-compatible devices use this approach: the bus switch and the ultrafast buffer.

Bus Switch. The bus switch is a low-resistance (5 Ω typical) switch that can connect the output of one gate to the input of another. It was introduced in 1991 by Quality Semiconductor as the QuickSwitch, and is discussed in its data book (6). Since its output follows its input, it is like a wire: it has no transition delay. You can use it to connect the driving gates on one bus to receiving gates on another bus. This is tying two buses together, and the name bus switch derives from this application. It has the advantage of providing a nearly zero-delay connection between gates.

Ultrafast Buffer. Another device that avoids transition delay is the ultrafast buffer, introduced by MicroLinear in 1994 as their part number ML74244, as described in Ref. 21. The ultrafast buffer is a linear amplifier with a gain of 1.0. Since it has a gain of unity, its output follows its input. This allows it to have a propagation delay of less than 1.5 ns, whereas the delay of fastest comparable buffer is 3.8 ns.

High-Speed Signals and Noise-Limited Design

Electrical noise can limit the speed of TTL designs, as discussed by Johnson and Graham (20) and Buchanan (2). High-speed systems are synchronous systems, with few exceptions. A *synchronous system* is a clock-driven system. The speed of the system is determined by the speed of the clock. The clock drives flip-flops. In such a system, new data are clocked into flip-flops, logic gates combine the outputs of the flip-flops to determine their next values, and the new values are clocked into the flip-flops. The logic gates must generate the next values from the current values before the next clock. The re-

sulting signal must be valid and unchanging before it can be clocked into a flip-flop.

Trace Ringing and Reflection. When a gate drives a printed circuit board trace, it can generate noise. Figure 17 shows a diagram of one gate driving another through a printed circuit board trace. The trace has inductance and capacitance, and the gate input has capacitance. This forms a resonant circuit. Note that the trace inductance and capacitance are distributed, while the gate input capacitance is not. The circuit shown is representative for short trace lengths. Very long traces look more like transmission lines, and the ringing effects become reflection effects. However, the nature and effect of the noise and the methods of reducing it are similar.

When the output of the driving gate at A transitions, the signal at the receiving gate input B consists of a damped sine wave imposed on the signal of the driving gate. The damped sine wave causes the signal at B to go below 0 V, then above 0 V. The portion below zero is called the undershoot, and the portion above is called the overshoot. If the overshoot is large enough, it violates the TTL logic low level in this case. However, if you wait long enough, the sine wave noise will decay to a safe level. The time you must wait is the ringing interval. Ringing represents an additional propagation delay.

The ringing interval is a function of the gate output transition time, the resonant circuit, and the gate output resistance. The output resistance determines the damping factor, or decay rate, of the resonant circuits. Increasing the output transition time decreases the undershoot and overshot amplitude. This decreases the ringing time, because it decreases the number of ringing cycles to achieve a safe signal level. However, increasing the transition time also increases the transition delay. The result is a compromise, with output transition times of 2 ns to 4 ns being typical.

Increasing the resonant frequency of the resonant circuit reduces the amplitude of the noise for a given transition time. It also reduces the ringing time, because the period of the resonant frequency is less. Changing the resonant circuit to increase its frequency means changing the layout of the printed circuit board to reduce trace inductance and capacitance. There is a set of techniques for improving printed circuit board layouts to reduce noise, as discussed by Buchanan (2). The first of these is to reduce the trace length as much as possible. Others include running the traces over a ground plane and making them wider to reduce their inductance. A related technique is to use the smallest packages available for the components. This minimizes the inductance of the leads in the package and allows the packages to be placed close together for minimum trace length. There is an obvious limit to these improvements: the traces have to be long enough to connect the components.

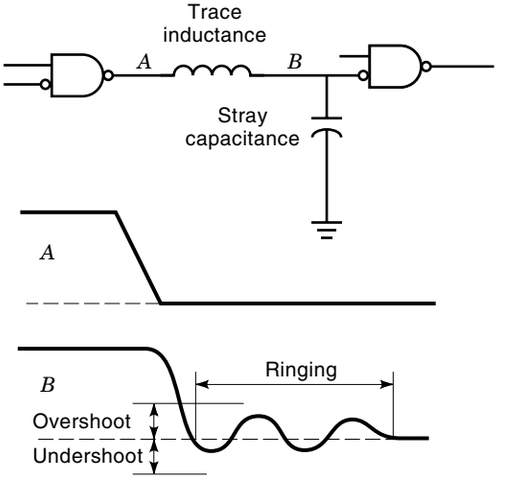

Figure 17. Circuit and timing diagrams of printed circuit trace ringing.

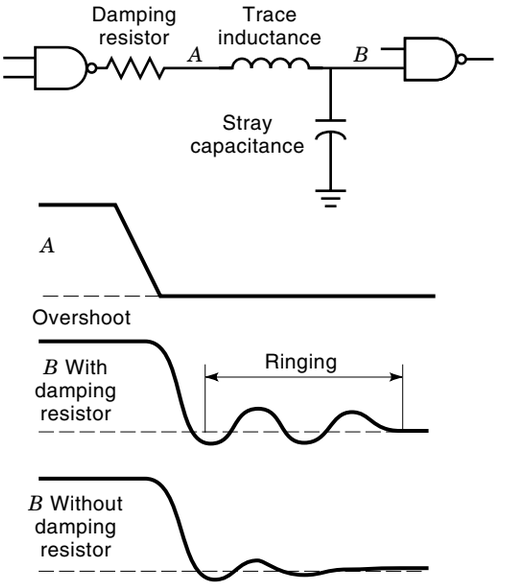

Figure 18. Circuit and timing diagrams showing effect of a damping resistor on trace ringing.

A third method of reducing the ringing interval is to increase the damping factor of the resonant circuit by adding series resistance. A common technique from the earliest days of TTL design is to add a 25 Ω to 33 Ω resistor in series with the driving gate, as shown in Fig. 18. The added resistor reduces undershoot and overshoot, and it reduces the ringing time by reducing the time to safe signal levels. This damping resistor was designed into some TTL integrated circuits, notably the AM2965 and AM2966 by Advanced Micro Devices (AMD). More recently, a full CMOS TTL family, the 74FCT2000 series, was introduced by Quality Semiconductor with damping resistors on all outputs.

Ground Bounce Noise. Ground bounce is a noise source related to ringing. Figure 19 shows a diagram of a circuit with ground bounce. A ringing voltage is injected from one circuit into another in the same package due to inductance in their common ground wire. This is discussed by Johnson and Graham (20) and Buchanan (2) and as an application note in the Quality Semiconductor data book (6). In this circuit, when Q_1 turns on and discharges the voltage in the load capacitance, the discharge current flows through the ground lead inductance. This current pulse causes a voltage pulse across this inductance, at B, proportional to the load capacitance and ground lead inductance, and inversely proportional to the transition time of Q_1. This pulse is the ground bounce pulse.

If Q_2 is turned on and not changing, the pulse appears at its output, C.

Bus driver circuits, such as the 74FCT244 which has eight bus drivers in one package, are the circuits that commonly have ground bounce problems. In their case, seven of their outputs (Q_1's) turn on at once, while one output (Q_2) remains low and transmits the ground bounce pulse. For a 74FCT244 in a PDIP package driving 50 pF loads, the load capacitance is $7 \times 50 = 350$ pF and the ground inductance is 13 nH. The resulting ground bounce pulse may be as much as 1.5 V, with a resonant frequency of approximately 70 MHz.

You can decrease ground bounce noise by several methods. First, minimize the output voltage of the gates. The TTL high output voltage charges the load capacitance. The lower the voltage, the less charge in the capacitor and the smaller the current pulse in the load inductance. The second method for reducing ground bounce is to use small packages with low lead inductance. This reduces the initial pulse and increases the resonant frequency. However, note that the circuit resonant frequency and ground bounce noise both scale as the square root of the change of the inductance. Another technique is to use resistors on all outputs. The resistors provide rapid damping of the resonant ground bounce noise, in the same manner as they damp trace ringing.

Package Pushout Delay. Ground bounce has an effect on gate delay called *package pushout delay*. During the ground bounce pulse, the voltage available to the integrated circuit is reduced by the magnitude of the pulse. This increases the delay of all circuits by reducing the drive available for all transistors. The result is that the delay for all gate outputs switching is longer than for one output switching. This added delay is called package pushout, because it is related to the package lead inductance. The package pushout can be 1 to 2 ns. Smaller packages with lower inductances have lower package pushout values. The problem is that package pushout is seldom specified. TTL is specified and tested with one output switching. To know the worst-case delay, you need to estimate the package pushout for the circuit you are using. As an interesting note, resistor-output TTL parts are often slightly faster than their resistorless counterparts, because they have less ground bounce and lower resulting transient voltage loss.

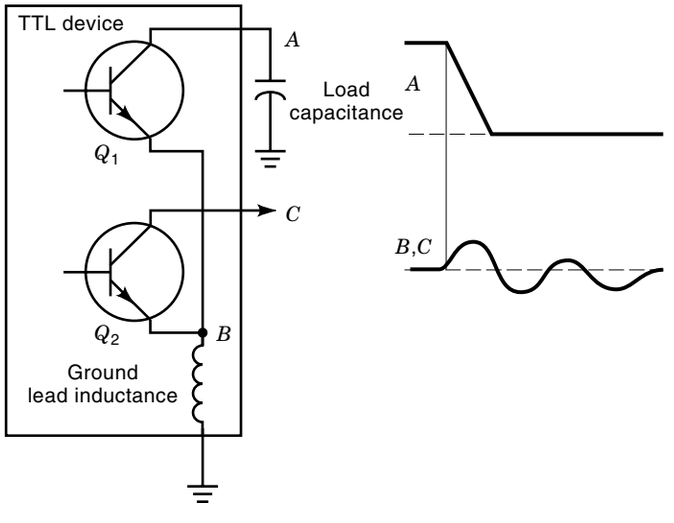

Figure 19. Diagram of a circuit with ground bounce.

TTL LOGIC DESIGN TOOLS

TTL became the standard logic for implementing digital systems, and new logic design methods developed as its use became widespread and the complexity of systems increased. The first methods used pencil and paper. Logic was drawn on paper using symbols for gates and lines for the wires that connect them. These drawings were converted to *net lists,* that is, lists of integrated circuit pins to be connected by wires. A set of pins to be connected together by wires is called a *net.* Each net is driven by one gate. The total number of nets equals the total number of gate outputs that are used.

Net lists were initially used to wire together gates by hand by wrapping wires around the pins of integrated circuit sockets. This was called *wire wrap* technology. Later, net lists were used to specify the traces on printed circuit boards. The net list determined the traces needed to connect the pins of the integrated circuits on the board.

Net-List Checking

A design in net-list form can be checked for errors by a computer. Each connection in a net list is identified as a gate input or output. Each valid net has one and only one gate output connected to it. The list is checked for nets that have more than one output and nets that have no output. Either is an error. These simple programs are valuable: they catch many clerical errors.

Schematic Capture and Printed Circuit Layout Programs

Drawing logic schematic diagrams on paper and converting them to net lists is a lengthy manual process. Computer programs were developed that allow a designer to enter a design using a graphic editor, called a schematic capture program. Once entered, the net list is generated automatically from the graphical design data entered. The schematic capture program has other advantages. When you change the schematic, you do not have to redraw it: the computer does it for you. Also, logic errors that would result in net-list errors can be found and corrected quickly. In fact, they can be found while you are drawing the schematic.

Printed circuits were initially designed manually. The traces on the printed circuit board were drawn on a piece of paper, called a layout. This is exacting work because what is on the layout will be printed on the printed circuit board in manufacturing. Computer programs called printed circuit layout programs were developed to simplify and streamline this process. A printed circuit layout program is a graphic editor for printed circuits. It has several advantages. Layouts can be modified without having to redraw the layout, and the computer is more precise than a manual layout. Also, these programs can check the layout against the net list. Some programs have automatic layout generators, called autorouters, that can automatically generate all the traces, given the net list and a list of the components to be connected.

PLDs and Logic Design Automation

The PLD extended the concept of logic design automation. When John Birkner invented the programmable array logic (PAL) device at Monolithic Memories (MMI) in 1976, he invented a market as well as a device. The PAL was not the first programmable logic device. But when MMI introduced the PAL, it also introduced the PALASM program for programming the PAL, and MMI made PALASM freely available to the public by publishing the Fortran code for it, as given by Birkner in Ref. 13. The combination of the device and a complete, automated design system for programming it allowed the PAL to become the market leader in programmable logic.

The PALASM Program and Logic Synthesis

PALASM promoted a new TTL design method: design by logic equation. To program a PAL, you entered logic equations into a PALASM program, and the programming hardware would program the PAL to implement those equations. The PALASM method of logic design by equation was the first widely used example of logic synthesis. Terms in the equations create logic gates and wire them together, but the equations themselves look nothing like symbolic gates. The PALASM program synthesizes logic gates from the equations. From this beginning sprang the more advanced high-level logic design languages in use today, such as Verilog and VHDL. For an example of the Verilog high-level design language, see Thomas and Moorby (22).

The PALASM Program and Logic Simulation

In addition, PALASM provided another crucial feature: logic checking. In addition to entering the equations for the logic, you could enter test patterns for the logic. The test patterns consisted of input patterns and the expected outputs for those patterns. The designer could check the accuracy of the logic equations before programming the PAL. Designers quickly learned to do so.

The logic checking portion of the PALASM program advanced the concept of computer testing of logic designs before manufacture. It performs testing by simulating the logic and comparing the expected results against the simulation. The simple tests performed by PALASM have evolved into sophisticated logic simulators, automatic test vector generators, and Verilog test benches that allow designers to test, evaluate, and improve large logic designs before converting them to hardware.

THE FUTURE OF TTL

TTL was introduced as a logic gate product line. Because of its wide use, it became an industry standard. Its electrical input and output characteristics became an interface standard. Devices that met this standard were called TTL-compatible. TTL devices were most prevalent when digital systems, most notably computers, were built using TTL SSI, MSI, and LSI devices. As the pace of integration introduced with TTL continued, designs moved from using small, standard TTL devices to PLDs, FPGAs, gate arrays, and microprocessors. TTL devices as design elements have moved into the background. They now fill niches as glue logic and as support logic for their larger progeny. In this role, they remain important. They serve needs not easily or practically served otherwise.

TTL will continue to improve as long as higher performance can command higher prices. Future evolution of TTL will be more closely tied to packaging and signal integrity. TTL silicon is now faster than the traces that connect the circuits. Signal integrity is now the challenge to increasing speed. Faster circuits will require more attention to package design, to printed circuit board traces and other interconnect methods, and to the analog characteristics of the circuits that communicate over those traces.

BIBLIOGRAPHY

1. Annonymous, *The TTL Data Book for Design Engineers,* Dallas: Texas Instruments, Inc., 1973.
2. J. E. Buchanan, *CMOS / TTL Digital Systems Design,* New York: McGraw-Hill, 1990.
3. Anonymous, *Schottky and Low-Power Schottky Data Book,* 2nd ed., Sunnyvale, CA: Advanced Micro Devices, 1977.
4. Anonymous, *TTL Data Book,* Mountain View, CA: Fairchild Camera and Instrument Corp., 1978.
5. Anonymous, *High Performance CMOS Data Book,* Santa Clara, CA: Integrated Device Technology, Inc., 1986.
6. Anonymous, *Quality Semiconductor 1993 Data Book,* Santa Clara, CA: Quality Semiconductor, 1993.
7. J. F. Wakerly, *Digital Design: Principles and Practices,* 2nd ed., Englewood Cliffs, NJ: Prentice-Hall, 1994.
8. J. A. Walston and J. R. Miller (eds.), *Transistor Circuit Design,* New York: McGraw-Hill, 1963.
9. M. V. Joyce and K. K. Clarke, *Transistor Circuit Analysis,* Reading, MA: Addison-Wesley, 1961.
10. J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective,* Upper Saddle River, NJ: Prentice-Hall, 1996.
11. M. Flynn and P. Low, Some remarks on system development, *IBM J. Res. Develop.,* **11** (1), 1967.
12. C. Mead and L. Conway, *Introduction to VLSI Systems,* Reading, MA: Addison-Wesley, 1980.
13. J. Birkner, *Programmable Array Logic Handbook,* Sunnyvale, CA: Monolithic Memories, 1978.
14. V. L. Burton, *The Programmable Logic Device Handbook,* Blue Ridge Summit, PA: TAB Books, 1990.
15. M. Bolton, *Digital Systems Design with Programmable Logic,* Workingham, England: Addison-Wesley, 1990.
16. R. D. Middlebrook, *An Introduction to Junction Transistor Theory,* New York: Wiley, 1957.
17. R. H. Mattson, *Basic Junction Devices and Circuits,* New York: Wiley, 1963.
18. G. W. Neudeck, *The Bipolar Junction Transistor,* 2nd ed., Reading, MA: Addison-Wesley, 1989.
19. B. G. Streetman, *Solid State Electronic Devices,* 4th ed., Englewood Cliffs, NJ: Prentice-Hall, 1995.
20. H. W. Johnson and M. Graham, *High-Speed Digital Design: A Handbook of Black Magic,* Upper Saddle River, NJ: Prentice-Hall, 1993.
21. Anonymous, *Micro Linear Corporation Data Book,* San Jose, CA: Micro Linear Corp., 1995.
22. D. E. Thomas and P. R. Moorby, *The Verilog Hardware Description Language,* 2nd ed., Norwell, MA: Kluwer Academic Press, 1995.

DAVID C. WYLAND
The Wyland Group, Inc.