BIPOLAR AND MOS LOGIC CIRCUITS

Logic is the science that deals with the principles of reasoning. Reasoning can be either valid or faulty, and a statement can be either true or false. In electronic circuits, this translates into a device being on or off or an output voltage being high or low. The work of Claude Shannon, based on the earlier theoretical work of George Boole, established an algebra of logic. Boolean algebra describes logical operations on binary state variables such as *and, or,* and *exclusive or,* as well as the *invert* (*not*) function. Electronic circuits, defined by the Webster's New Collegiate Dictionary, as "an assemblage of electronic components: hookup," implement each of the basic logic functions, which in turn can be interconnected to achieve more complex arithmetic or control operations.

BIPOLAR LOGIC CIRCUITS

Bipolar junction transistors (BJTs) along with resistors and diodes can be used to create electronic circuits that perform Boolean logic functions. Bipolar refers to transistors that have two polarities of charge carriers inside: electrons and holes. These were the first transistors developed commercially, starting in 1949. Junctions of p-type and n-type semiconductors were formed from silicon with small amounts of impurities to produce the BJT structure. Two types of BJTs are possible: *pnp* and *npn*, which refer to the arrangement and types of the semiconductor junctions.

Historical Background

The first bipolar logic circuits were implementation with discrete transistors on small circuit boards or as hybrids. In 1956, the first integrated circuits (ICs) were designed, which consisted of resistors, diodes, and bipolar transistors, all interconnected on chip by metal lines. Bipolar logic ICs have been designed in several styles or families, some of which are obsolete today. The first logic families were resistor–transistor logic (RTL) and diode–transistor logic (DTL). Later, as the cost of adding transistors to a circuit became secondary, the number of *active devices* (transistors) was increased to replace the resistors and diodes.

Transistor–transistor logic (TTL) was introduced in the early 1960s and was quickly accepted around the world, becoming the most popular logic family. Most mainframe computers of the 1970s were designed based on TTL logic chips, and only the TTL's density and speed changed significantly over the course of many years. Later, Schottky barrier diodes (SBDs) were used to prevent saturation in the bipolar transistors, which reduced power consumption and improved switching speed. Saturation in a BJT describes the condition when the base-collector junction becomes forward biased and conducts current. Saturated transistors slow down considerably. TTL used a single 5 V power supply, which became the de facto standard for logic circuits. Only recently has this begun to change to 3.3 V. Digital circuits based on TTL are still widely sold today. TTL held the largest market share until the early 1980s, when complementary metal–oxide–silicon (CMOS) began to overtake it.

For a period of time in the 1970s, integrated injection logic (IIL), also known as merged transistor logic, was a successful logic family. It promised the highest speeds and levels of integration. However, the processing complexity and the inferior speed–power product, compared to modern emitter-coupled logic (ECL) designs, have relegated IIL technology to historical interest.

ECL, a derivative of current-mode logic (CML), has been popular for over two decades. It made its first commercial appearance in 1962, introduced by Motorola as MECL1, and within a few years was capable of 1 to 2 ns propagation delays. In CML, whose name derives from the current steering realized by differential amplifiers, the transistors do not saturate by design. ECL logic is fairly dense, since several differential amplifiers are stacked on top of each other and share a single current source. ECL is used in high-speed computers

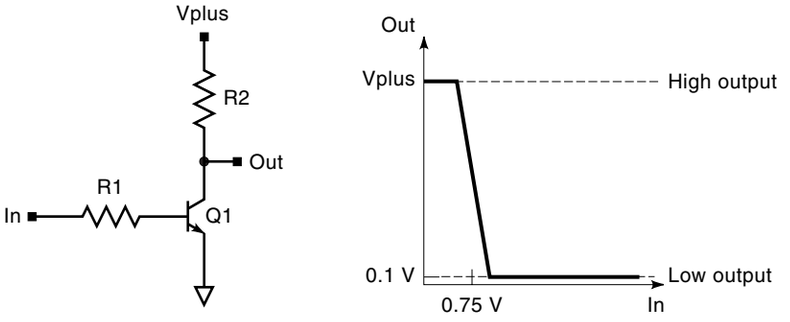

Figure 1. Simple inverter circuit and voltage transfer curve.

(such as the Crays), in instrumentation, and in communications. Typical propagation delays for ECL logic today are less than 100 ps per gate, and the maximum frequency of operation is greater than 10 GHz.

Nonthreshold logic (NTL) is a relatively new nonsaturating logic family that uses common-emitter (CE) stages with large emitter resistors to prevent saturation. The layouts of some NTL designs are very small; others have achieved some of the highest speed and speed–power records.

Bipolar CMOS (BICMOS) was introduced in the middle 1980s as an optimal technology containing both high-speed and high-density devices. The current drive of bipolar transistors could be used to drive long metal lines or large capacitances, which metal–oxide–silicon (MOS) is poor at doing. On the other hand, CMOS had made it possible to integrate very large numbers of logic gates, and memory cells, on a single IC chip, compared to bipolar.

Another new logic family, called current mirror control logic (CMCL), is claimed to achieve the highest speed–power product of any technology. It operates at less than 2 V and is derived from CML by removing the current sources and replacing them with current mirrors.

Bipolar Logic Circuits Operation

The most fundamental property of logic circuits is that their transfer characteristics (e.g., voltage transfer function from input to output) are such that they *clamp* at the output. Their gain is high enough so that a minimum input voltage swing will causes the output to swing fully on or off. This *limiting* action allows a large number of logic circuits to function simultaneously, for example in a microprocessor, without suffering any errors. The amount of signal degradation that can be tolerated at the input of a logic circuit is called its noise immunity, and it is proportional to the circuit gain.

Figure 1 shows a simple inverter circuit consisting of a common emitter (CE) transistor with its load resistor R2, and a current limiting resistor R1 at its input. Also shown is a simplified voltage transfer curve. When the input voltage increases from zero to the normal *on* voltage of the base-emitter junction (V_{be}, 0.75 V), the transistor begins to turn on and the output level begins to drop. The low output voltage will be reached when the transistor saturates, resulting in a level close to zero (about 0.1 V or less). This voltage is lower than what is needed to turn on the next stage, and therefore it is a logic zero.

A simple RTL circuit is shown in Fig. 2 as well as curves indicating changes in time at the inputs and the corresponding output change. No limiting resistors are shown for the inputs A and B, although these are normally necessary; they serve the secondary purpose of preventing one transistor from taking all the current coming from the power supply. RTL is a saturating logic family, and its transistors take longer to come out of saturation (turn off) than to go into saturation. Therefore its rising output voltage occurs with more delay than its falling voltage.

In Fig. 3, we show a two-input DTL *nand* gate. The input threshold of this circuit is equal to two diode drops ($2V_{be}$), since the input voltage goes up one diode and down three diodes. Only when both inputs are above the turn-on threshold does the output go low, corresponding to the *nand* function. The advantage of DTL over RTL is that the inputs do not draw any current when they are high, because of the reverse bias at their diodes. The resistor at the base of the output transistor speeds up its turnoff.

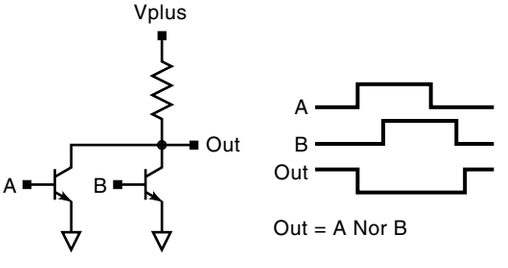

Figure 2. Resistor–transistor logic (RTL) *nor* gate. Timing waveforms show Boolean relationship.

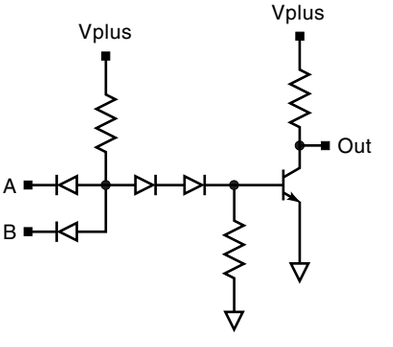

Figure 3. Diode–transistor logic (DTL) *nand* gate. Output is high for all input combinations except $A = 1$ and $B = 1$.

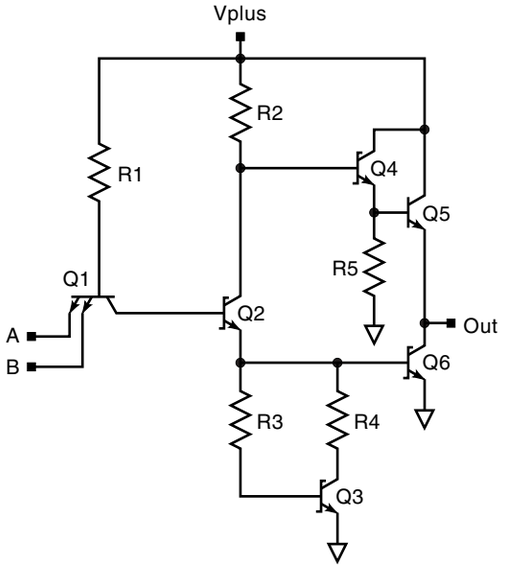

Figure 4. Schottky transistor–transistor logic (TTL) *nand* gate with totem-pole output and squaring circuit.

The Schottky TTL *nand* gate shown in Fig. 4 has similar operation to the DTL implementation except that the input diodes have been merged into two emitters at the input transistor. The output stage is composed of a *totem-pole* configuration driven by a phase-splitter transistor (Q2). The output is pulled up by a Darlington connect pair. Schottky diodes connected internally across base–collector junctions prevent those transistors that might saturate from doing so. The additional two resistors and a transistor circuit connected at the base of the output transistor (Q3, R3, and R4) are known as a *squaring circuit,* and it is used to sharpen the voltage transfer function. The TTL output levels are particularly troublesome, since they require driving large capacitive loads and uncontrolled (impedance) transmission lines; they must also source and sink current in a historically asymmetric way.

IIL circuits are similar to RTL circuits except that all resistors are replaced by *pnp* current sources, realizing an all-transistor design style. Figure 5 shows that the current sources are applied to the inputs of transistors and that the gates consist of the parallel connection of several transistors.

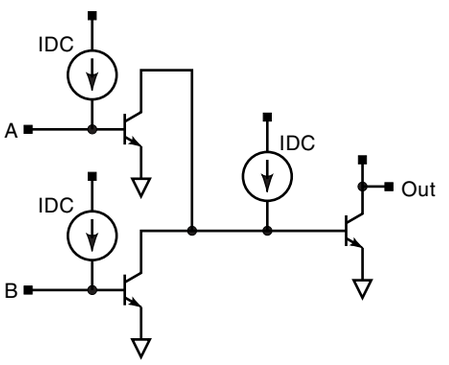

Figure 5. Integrated injection logic (IIL): *pnp* transistors make up the current sources. All-transistor design.

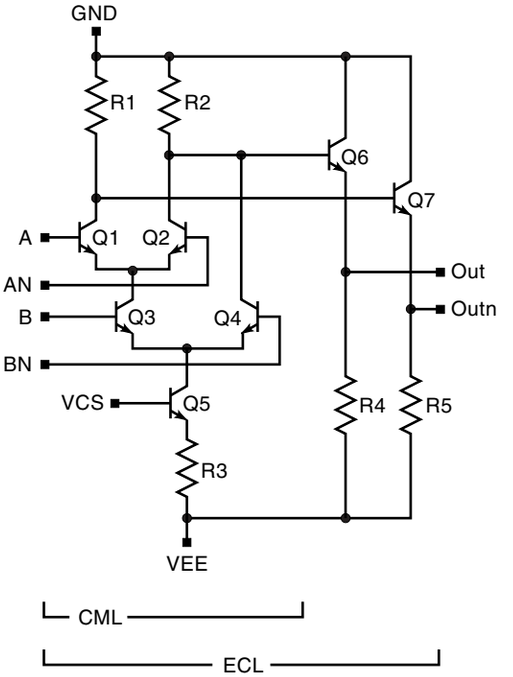

Figure 6. Emitter-coupled logic (ECL) *and* gate with full differential inputs and outputs. The addition of an emitter follower turns CML into ECL.

Because CE stages are loaded by current sources, the saturation is light. The *pnp*'s are integrated with the *npn*'s in a single *tub.* Multiple emitters on the *pnp* side further compress the layout. One shortcoming of IIL was that it required special processes to make *npn* transistors that could operate upside down, with the collector on top, and also special steps to make *pnp* transistors.

ECL/CML circuits are based on the differential amplifier. Figure 6 shows an ECL *and* gate consisting of two differential amplifiers stacked on top of each other. The addition of emitter follower outputs turns a CML circuit into an ECL circuit. In the figure, the emitter followers are simply loaded by resistors, but current sources are more commonly employed. At the bottom of the differential amplifiers, transistors Q5 implements a current source that is biased from a common voltage V_{cs}. Since the so-called tail current is constant, the amount of voltage swing developed across load resistors R1 and R2 is well controlled and saturation prevented. The bases of the differential amplifiers connect to dc levels such that their base–collector junctions do not saturate. There is normally a voltage difference of V_{be} (a diode drop) between signals A and AN and between signals B and BN, so that the emitter of Q1, for example, makes the collector of Q3 be no lower than the base of Q3. The *and* operation is verified by noting that only when both Q1 and Q3 are on will the current flow into R1, making the output OUTN low, and OUT high.

Although Fig. 6 shows fully differential inputs and outputs, it is common to use single-ended versions of both. This requires connecting one side of each differential amplifier to an appropriate threshold voltage level. These levels are usually generated by the same circuit that provides the bias to the current sources, and are typically separated from each

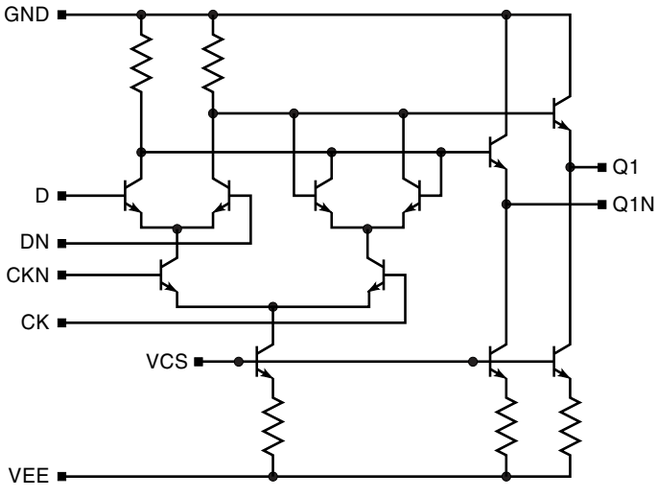

Figure 7. ECL latch. Outputs latch on CK high.

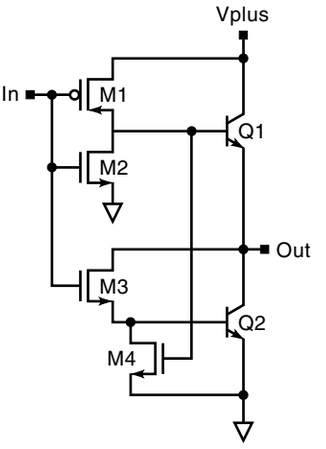

Figure 9. Standard BICMOS inverting buffer. All-transistor design requires no resistors. Bipolar transistors provide high current drive for large (or unknown) loads.

other by diode drops. The bias circuit is often a *bandgap* voltage regulator, which is capable of operating with great immunity from temperature and supply voltage variations.

An ECL latch is an interesting circuit that shows some of the special characteristics of this logic family. Figure 7 shows that two differential amplifiers are stacked on top of a third, all three sharing the same current source. When current flows through the input differential pair, on CKN high, the outputs Q and QN simply follow the inputs D and DN. When CK selects the current to flow though the differential pair on the right, it latches to the voltage that was previously presented at its inputs. This differential pair has its outputs cross-connected to its inputs, and also connected to the collectors of the input differential pair. The figure also shows emitter followers with constant current source loading.

Figure 8 shows a two-input nonthreshold logic gate. The NTL transistors do not saturate, since the current demand is proportional to the input voltage. However, this makes the transfer function fairly linear, resulting in lower gain and a soft voltage transfer curve. This is not desirable for logic circuits, but can be alleviated by adding parallel capacitors across the emitter resistors. In NTL the maximum collector current is predictable, and selected so that the logic voltage swing does not saturate the next stage. Emitter followers between stages improve interstage loading and the ability to drive more logic inputs (fanout).

As an example a standard BICMOS inverting buffer, which can be used to drive large capacitive loads, is shown in Fig. 9. The two *npn* transistors in combination can pull up and pull down larger currents, and faster, than MOS devices of similar area. The CMOS inverter, M1 and M2, drive the top *npn* in opposite phase to the drive presented to the bottom *npn*. The bottom *npn* is both buffered and kept out of saturation by the *n*-channel MOS (NMOS) device connected across its base and collectors. The CMOS inverter also drives a fourth NMOS transistors, M4, that speeds the turnoff of the bottom bipolar transistor.

An example of a current mirror control logic (CMCL) xor gate can be seen in Fig. 10. The removal of current sources in

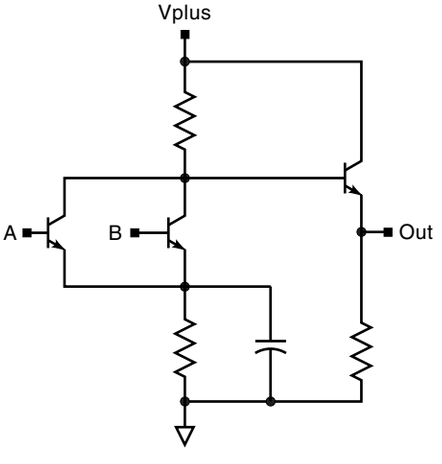

Figure 8. Nonthreshold logic (NTL), including speedup capacitor.

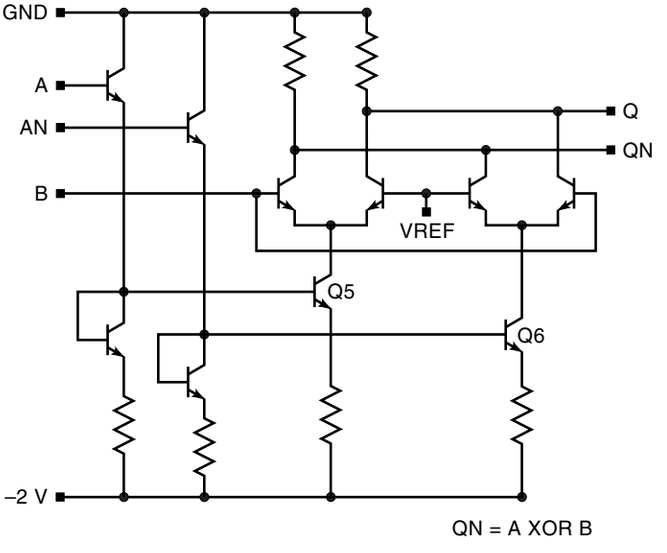

Figure 10. Current mirror control logic (CMCL) *xor* gate.

Table 1.

Parameter	BJT	MOS
g_m/I	Highest, $1/V_T = 1/(26\text{ mV})$	1/(250 mV)
$1/f$	Best	Poor
V_{offset}	Very good, $\exp(V_{be}/V_t)$	Poor (surface effects)
R_{in}	Medium, base current	Infinite
R_b	Yes, limits F_{max}	No
Switch	Not ideal	Yes
Complementary device	Lateral PNP, 200 MHz	PMOS, 40% speed
F_t	Function of current	Function of $V_{gs} - V_t$

an ECL circuit allows it to operate from 2 V. As shown, two levels of logic, corresponding to signals A and AN, and B, are possible. The differential signals A and AN swing about 0.5 V and drive the *master* side of two separate current mirror circuits. When A is high, the current through Q5 is on, while AN, being low, produces almost no current in Q6. The signal B is applied against a reference voltage that is regulated against supply voltage changes.

Comparison of BJT with Other Technologies

At the highest speeds, bipolar logic survives the CMOS onslaught; it also enjoys the advantage of being differential and semianalog. The reasons why bipolar technology continues to be so important will become apparent from the comparison of the advantages and disadvantages of both BJTs and MOS devices shown in Table 1.

Bipolar transistors have a tremendous (exponential) gain and high current drive capability, allowing small devices to develop large gain and to drive loads of considerable size, by contrast to MOS. In addition, the small offset voltage of bipolar differential pairs is good for logic at small signal swings. Fully differential signals can have smaller levels, with the additional benefit of better common-mode signal and noise rejection, resulting in better noise immunity.

Small logic swing (the way to the future for all logic families) can naturally take bipolar logic to lower supply voltages and higher speeds. Speed and power improve with reduced voltage swing. In addition, the much lower $1/f$ frequency corner of BJTs, related to its bulk properties, makes them much superior in applications where memory effects are detrimental, as in data transmission and test equipment.

Because bipolar logic is built around a lot of the same elements used in analog design (BJTs, resistors, diodes, capacitors, and even inductors), it lends itself nicely to mixed signal design, combining analog and digital on the same circuit. For example, some of the key bipolar and BICMOS analog functions, such as analog-to-digital converters, digital-to-analog converters, phase-locked loops, and disk-drive electronics, contain a mix of logic circuits.

Concerning GaAs metal semiconductor field effect transistors (MESFETs), all of the above comparisons apply, since their operation is very similar to that of MOS transistors, except that the higher mobility (at low electric fields) of GaAs gives it an advantage. GaAs ICs also are manufactured on a semiinsulating substrate that gives reduced signal loading and improved crosstalk. However, competing directly with GaAs logic, Si–BJT processes are cheaper, more stable (statistical corners), better at low temperature, lower in noise for data transmission (dc-coupled), and nontoxic.

New silicon–germanium (SiGe) bipolar technologies are challenging even the most advanced processes in other materials, since they achieve very high cutoff frequencies (e.g., 75 GHz) and have other improved transistor parameters. Since 1995, the circuit speed performance of SiGe has not only exceeded GaAs MESFETs but almost equaled that of other more exotic technologies such as HEMT (high electron mobility transistor) and HBT (heterojunction bipolar transistor). Finally, silicon-on-insulator and higher resistivity substrates will remove the last obstacle in the competition for the high-speed and mixed-signal markets, namely, the substrate's capacitance, loss, and cross-talk.

CMOS LOGIC CIRCUITS

Although bipolar processes are used for many high-speed designs, CMOS has become the dominant technology for most of today's digital ICs. Lower power and higher integration densities are two reasons for the success of the CMOS process. Field effect transistors (FETs) were conceived as early as 1925, more than two decades before the invention of the bipolar transistor. However, development of fabrication processes to support the manufacture of FETs did not occur until the 1960s. The earliest MOS processes were based on single transistor types. The CMOS fabrication process allows both types of MOSFETs, *p*-channel and *n*-channel, to be fabricated together. Schematic symbols for an *n*-channel MOS transistor and a *p*-channel MOS (PMOS) transistor are shown in Fig. 11. Circuits consisting of NMOS and PMOS transistors can be used to perform boolean logic functions.

Classical CMOS Gates

In order to understand how CMOS logic circuits behave, a model for the operation of an individual FET is needed. The simplest model of an NMOS transistor is that of a switch that closes when the logic level on the gate is high, passing the logical state on the input to the output. With a low on the gate, the switch modeling the NMOS transistor is open. For modeling a PMOS transistor, the switch is open when the gate is driven with a high and closed when the gate is driven with a low. Boolean logic functions such as *nand, nor,* and *invert* can be easily implemented by networks of PMOS and NMOS transistors as shown in Fig. 12.

Series connections of transistors perform a logical *and* operation, whereas parallel connections perform a logical *or* operation. For the *nand* gate in Fig. 12, two NMOS transistors are in series between the output (y) and ground (the source of a low logic level). When both x_1 and x_2 are asserted high, the output is pulled to ground, thus passing a low voltage level to

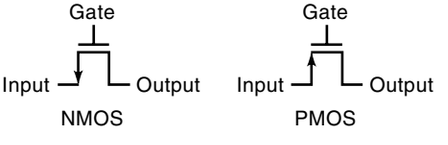

Figure 11. MOS transistor schematics. As shown, the input is arbitrarily connected to the transistor source, and the output to the drain. The gate controls the switch action.

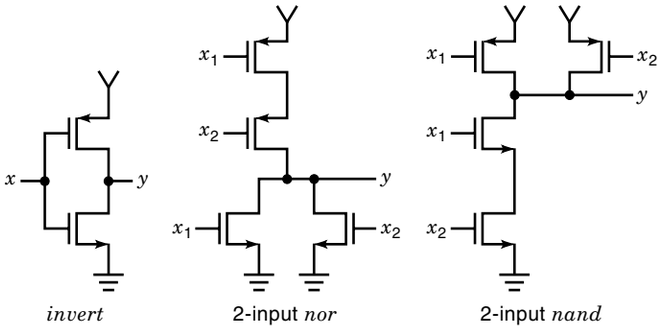

Figure 12. Basic logic gates are constructed by parallel and series combinations of NMOS and PMOS transistors.

the output. The two PMOS transistors are connected in parallel between the high-voltage supply and the output. When either x_1 is low or x_2 is low, a path is closed between the high supply and the output, passing a high voltage level to the output. The Boolean operation of a *nand* function is thus performed.

To understand why PMOS transistors are used to pass high levels to the output while NMOS transistors are used to pass low levels, the switch model must be modified to take account of the threshold voltage. The switch modeling an NMOS transistor is actually closed only when the difference between the voltage on the gate, V_g, and the voltage on the source, V_s, is greater than a threshold voltage, V_t. If an NMOS transistor is used to pass a low level, then V_g is always greater than V_t and the switch is always closed. If, however, an NMOS transistor were used to pass a high level, the output would rise until the voltage was V_t less than the gate voltage, at which point the switch would open and the output would cease to rise. NMOS transistors will pass good 0's, but degrade the voltage level of a 1. PMOS transistors, by similar arguments, pass good 1's, but poor 0's.

The operation of digital CMOS circuits can be described by Boolean algebra. The normal boolean expression describing the *nand* function is given by $y = \overline{x_1 x_2}$, which conveys that both x_1 and x_2 must be high before $y = 0$. An alternative Boolean expression for the *nand* function is $y = \overline{x}_1 + \overline{x}_2$, which implies that if either x_1 or x_2 is low, then $y = 1$. Both concepts are needed to describe the CMOS *nand* gate fully.

Let $y = x(0)$ be read as "y is equal to x passing a 0," meaning that when $x = 1$, a 0 is passed to the output. Also, let $y = x(1)$ be "y is equal to x passing a 1," meaning that when $x = 1$, a 1 is passed to the output. The CMOS *nand* function can then be fully described by the following expression:

$$y = x_1 x_2(0) + \overline{x}_1(1) + \overline{x}_2(1) \tag{1}$$

This can be read as "y is equal to $x_1 x_2$ passing a 0 or $\overline{x}_1$ passing a 1 or $\overline{x}_2$ passing a 1." This notation conveys the meaning that when x_1 is high and x_2 is high, the output is low, and if either x_1 or x_2 is low, the output is high. The first term of Eq. (1) implies the series connection to ground of two NMOS transistors. The second and third terms imply a parallel connection of PMOS transistors.

The concept of passing 1's and 0's can be used to derive other classical CMOS gates. The derivation of the two-input *nor* gate will be used as an example. The information in a truth table describing the logical function of a two-input *nor* gate can be entered into a Karnaugh map and the 1's and 0's can be grouped together as shown in Fig. 13. These groupings imply that a 1 must be passed to the output under the input condition $\overline{x}_1 \overline{x}_2$ and that a 0 must be passed to the output when $x_1 = 1$ or when $x_2 = 1$. Using the pass notation introduced,

x_1	x_2	y
0	0	1
0	1	0
1	0	0
1	1	0

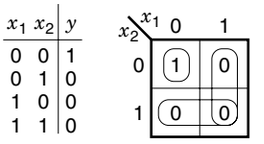

Figure 13. *Nor* truth table and Karnaugh map with groupings for both the PMOS network and the NMOS network.

$$y = \overline{x}_1 \overline{x}_2(1) + x_1(0) + x_2(0) \tag{2}$$

Equation (2) describes the network of MOS transistors needed. The first term is a logical *and* operation. This implies a series path between the output and the high supply, since both x_1 and x_2 must be low to pass a 1 to the output. Since p-channel transistors pass good 1 levels and the switch model closes when the gate is driven by a low level, a pair of series PMOS gates are used. The next two terms imply parallel paths passing 0's when x_1 or x_2 is high. Parallel n-channel transistors are therefore used. Figure 12 shows the classical CMOS *nor* gate with series PMOS transistors between the output and the high supply and parallel NMOS transistors between the output and the low supply.

Digital MOS ICs can be designed using exclusively *nand, nor,* and *invert* gates; however, the cost of an IC is proportional to the size of the die, and this design methodology might not result in the most economical solution. Several logic families have been developed to reduce the number of transistors required to implement a given Boolean logic expression. As an example, the implementation of the Boolean function described by the truth table in Fig. 14 will be described for several static logic families. Using basic logic gates, two two-input *nand* gates and one three-input *nand* gate would be needed to implement the design example, requiring a total of 14 transistors.

x_1	x_2	x_3	x_4	y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

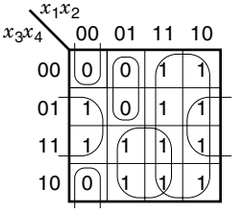

Figure 14. Design example used for comparison of several CMOS logic families.

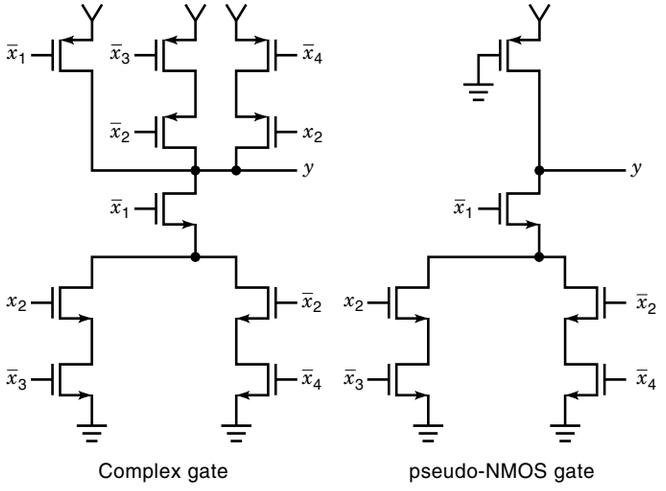

Figure 15. Complex gate implementation using fully complementary networks and a pseudo-NMOS approach.

A complex gate with just one node can perform the same function. Using a Karnaugh map, the required PMOS and NMOS networks to implement a complex CMOS gate can be derived in the same manner that the *nor* gate was derived. For the Karnaugh map groupings shown in Figure 14,

$$y = \overline{x}_1[x_2\overline{x}_3(0) + \overline{x}_2\overline{x}_4(0)] + x_1(1) + x_2x_3(1) + \overline{x}_2x_4(1) \qquad (3)$$

Using NMOS transistors to pass 0's and PMOS transistors to pass 1's, and remembering that a PMOS switch is closed when driven with a low logic level, we obtain the 10-transistor solution shown on the left in Fig. 15. This reduction in transistor count motivates cost-conscious circuit designers to find alternative implementations, especially for logic blocks that will be placed many times on a digital IC.

Pseudo-NMOS Gates

Another alternative approach reaches back into history to reduce the transistor count even further. In the 1970s, MOS processes were dominantly single-transistor processes. An NMOS processes had typically two types of n-channel FETs: an enhancement transistor similar to the NMOS transistor available in today's CMOS process, and a depletion transistor. If the gate and source of the depletion device were tied together (self-sourced), the transistor could be modeled as a resistor. The model for the NMOS enhancement transistor also need to be expanded to include the concept of resistance. When $V_g - V_s > V_t$ and the drain voltage V_d is such that $V_d - V_s < V_g - V_s - V_t$, then the closed switch modeling the NMOS transistor has the property of resistance.

In the NMOS process, Boolean logic was implemented with an NMOS network identical to the one shown in Fig. 15, while the PMOS network was replaced by a single, self-sourced depletion transistor connected to the high supply. When the input state was such that $y = 1$, the NMOS network's path to ground would be broken and the output would charge to the high supply through the resistance of the depletion NMOS transistor. When the inputs were such that $y = 0$, then the output would be a voltage division between the resistor representing the depletion transistor pullup and the resistor representing the enhancement NMOS transistors in the pulldown path. The value of the resistance of a FET is inversely proportional to the width of the transistor. The voltage value representing a low logic level could be controlled by the ratio of the widths of the depletion and enhancement NMOS transistors. The enhancement NMOS transistors were sized to be of much smaller resistance than the depletion transistor setting a low voltage for the low logic level.

This NMOS structure can be imitated in today's CMOS process, by replacing the depletion pullup transistor with either a PMOS transistor whose gate is tied to a logic low (*pseudo-NMOS*), or an NMOS transistor whose gate is driven with a bias voltage that is greater than the supply voltage plus V_t (*modified pseudo-NMOS*). The pseudo-NMOS structure is shown in the right half of Fig. 15 and requires just six transistors. A modified pseudo-NMOS structure would also require only six transistors.

There is a cost associated with the pseudo-NMOS approaches. When $y = 0$, a current flows between the supply voltage and ground. This current flow is minimized by making the pullup resistance just low enough to maintain the desired speed of operation for the circuit, but still represents enough current to be the reason that CMOS processes replaced the NMOS and PMOS processes of the 1970s. Although entire chip designs have been based on this logic style, it is normally used in conjunction with dynamic techniques, which will be discussed in a following section.

Pass Gates

The concept of passing 1's and 0's helps to understand the operation of static CMOS gates. The limitation of passing only 0's and 1's requires the use of more transistors than necessary for many Boolean functions. This constraint is lifted for pass transistor gates. The first grouping, in the Karnaugh map of Fig. 16, is $\overline{x}_1\overline{x}_2$, passing the variable x_4. The second is $\overline{x}_1x_2$, passing x_3. The resulting equation is

$$y = \overline{x}_1[\overline{x}_2(x_4) + x_2(x_3)] + x_1(1) \qquad (4)$$

The four-transistor circuit shown on the right side of Fig. 16 implements the desired function. Since the variables x_3 and x_4 can be either 1's or 0's, the voltage level of 1's will be degraded at y and must therefore be restored. One approach for passing good levels would be to use an NMOS and a PMOS transistor in parallel (a transmission gate) whenever a variable is passed. The NMOS device would pass a good 0, while the PMOS transistor would pass a good 1. If transmission gates were used, seven transistors would be required. The second approach would be to actively restore the output level at the output of the pass network.

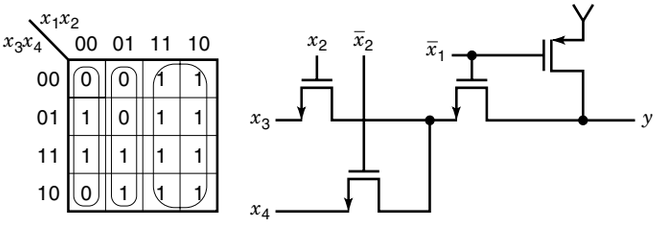

Figure 16. Pass transistor design eliminates the constraint of passing only 1's and 0's.

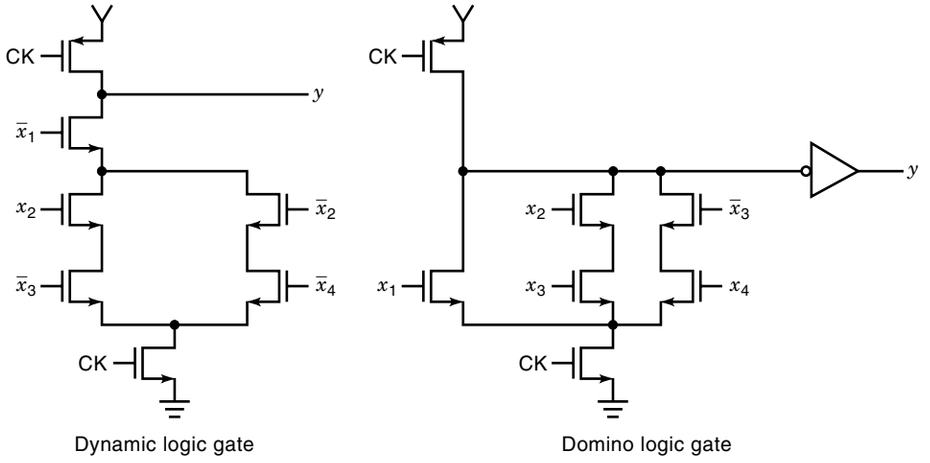

Figure 17. Dynamic logic structures. A basic precharge evaluate structure (left) is compared with a domino logic structure.

The logic of this design example is to perform the function of a synchronous set and load/hold when used to drive the *d* input of a master–slave flip-flop. Since an array of 100 of these circuits was needed on an actual design and since active level restoration of the 1 level was inherent in the master section of the flip-flop, the pass transistor design shown in Fig. 16 was used, saving 1000 transistors in comparison with the *nand*-gate approach.

Dynamic Gates

Unlike static logic, in which there is a path between the output and either the high supply or ground for every input state, dynamic logic relies on storage of charge on the capacitive load seen at the output to maintain the logic level for periods of time. The time base for storage is normally established by a clock. The output of dynamic logic is also valid only during a portion of the charge storage time. A dynamic logic gate that implements the design example of Fig. 14 is shown on the left side of Fig. 17. When the clock CK is low, *y* is precharged to a 1 level. When the clock transitions high, the NMOS transistor at the bottom of the NMOS network evaluates the state of the inputs and conditionally discharges the output to ground. If there is no path to ground through the NMOS array, the output remains high. This high voltage level is stored on the capacitance seen at the output node. Over time this level will degrade due to parasitic leakage paths. This leakage establishes a lower limit on the rate of the clock signal. This limit could be eliminated by adding a PMOS transistor, like the pseudo-NMOS pullup device of Fig. 15, with a resistance just low enough to overcome the effects of leakage. A total of seven transistors are required to implement the design example.

There are two design considerations for dynamic logic gates. The output of the dynamic gate of Fig. 17 is floating or at a high impedance when storing a high level. The logic level is stored in the form of charge on the capacitance of the node. This floating node is susceptible to charge sharing introduced by crosstalk from adjacent lines. Charge sharing occurs between two physically adjacent conductors due to the parasitic capacitance formed by the node metals separated by an the insulating layer. When an adjacent node voltage changes, charge is coupled through this parasitice capacitance to the otherwise unrelated node. Charge can be moved off of the node by charge sharing, reducing the voltage level of the stored logic 1. Charge sharing across the gate capacitance of transistors within the NMOS pulldown structure also can degrade the output 1 level if the inputs transition while clock is high. The second consideration occurs when one dynamic gate drives the input of a second dynamic gate. A race exists between when the clock transitions high and the first dynamic gate evaluates low. This race occurs because both outputs are precharged high when the clock is low and evaluate when the clock is high. With the inputs to the NMOS transistors of the second stage high due to the precharge of the first stage, the second node begins to fall even though the first stage might transition low, requiring the second stage to remain high.

Due to the charge sharing and inherent race condition of dynamic gates, they are not generally used for random logic. When the NMOS and PMOS arrays for implementing a desired function are large, the effort required to ensure proper operation is worthwhile. Dynamic logic is often used, for example, to implement the address decode logic of a memory circuit.

In order to overcome the race condition and external crosstalk susceptability of dynamic logic, the domino logic configuration shown on the right side of Fig. 17 can be used. The inverter on the output node isolates the high-impedance node from charge-sharing effects external to the node. Also, since precharging the internal node to a 1 sets the output to a 0, no race exists between cascaded stages. The first state evaluates, and if the input transitions, it may cause the second to transition, and so on in a domino effect. The NMOS network is simply designed to produce the inverse of the function desired at the output of the inverter.

Future Challenges for CMOS

The new set of constraints presented by today's submicrometer CMOS processes are opening for debate again design questions long thought answered. MOS processing technologies underwent a evaluation in the past. NMOS, with inherent speed advantages, replaced PMOS as the dominant MOS technology in the early to mid 1970s. CMOS, with inherent power advantages, replaced NMOS processes in the late 1970s and early 1980s. The speed and density of the circuits being designed at that time had risen to the point where packaging technology could no longer deal with the power lev-

els generated by NMOS designs. Today we are faced once again with the same power dilemma. Architecture, logic, circuit, and fabrication techniques are evolving to reduce power while maintaining or increasing performance. As the submicrometer geometries continue to shrink, previous assumptions about the choice of design methodologies and families are no longer valid. For instance, the area of a circuit is now a much stronger function of interconnect than transistor count. Off-state leakage is no longer negligible and can in the case of large memory cores actually dominate as the main component of static power. Switching power, the power associated with charging and discharging capacitive loads, increases with capacitance. This is allowing the design tradeoff between capacitance and static power to be reevaluated. As with most engineering disciplines, VLSI logic design will continue to evolve.

BIBLIOGRAPHY

A. R. Alvarez, *BICMOS Technology and Applications,* Norwell, MA: Kluwer Academic, 1989.

A. Barna, *VHSIC: Technologies and Tradeoffs,* New York: Wiley-Interscience, 1981.

M. I. Elmasry, *Digital Bipolar Integrated Circuits,* New York: Wiley-Interscience, 1983.

S. H. K. Embabi, A. Bellaouar, and M. I. Elmasry, *Digital BICMOS Integrated Circuit Design,* Norwell, MA: Kluwer Academic, 1993.

E. D. Fabricius, *Introduction to VLSI Design,* New York: McGraw-Hill, 1990.

L. A. Glasser and D. W. Dobberpuhl, *The Design and Analysis of VLSI Circuits,* Reading, MA: Addison-Wesley, 1985.

L. J. Herbst, *Monolithic Integrated Circuits,* Oxford: Clarendon Press, 1985.

Integrated Circuit Engineering Corporation, *STATUS 1997,* Scottsdale, AZ: ICE, 1997.

K. Kishine, Y. Kobayashi, and H. Ichino, A high-speed, low-power bipolar digital circuit for Gb/s LSI's: Current mirror control logic, *IEEE J. Solid-State Circuits* **32**:, 215–221, 1997.

P. L. Mathews, *Choosing and Using ECL,* London: Granada Publishing, 1983.

R. Meyer, *Advanced Integrated Circuits for Communications,* Course ECE242 Notes, Berkeley: Univ. of California 1994.

J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective,* Upper Saddle River, NJ: Prentice-Hall, 1996.

H. Rein and M. Moller, Design considerations for very-high-speed Si-bipolar IC's operating up to 50 Gb/s, *IEEE J. Solid-State Circuits,* **31**: 1076–1090, 1996.

M. I. Rocchi (ed.), *High Speed Digital IC Technologies,* Norwood, MA: Artech House, 1990.

M. Shoji, *CMOS Digital Circuit Technology,* Englewood Cliffs, NJ: Prentice-Hall, 1988.

N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective,* 2nd ed. Reading, MA: Addision-Wesley, 1993.

S. Wolf, *Silicon Processing for the VLSI Era: Volume 2,* Sunset Beach, CA: Lattice Press, 1990.

GERMAN GUTIERREZ
Silicon Wave, Inc.

STERLING WHITAKER
NASA Institute for Advanced Microelectronics

BIPOLAR LOGIC CIRCUITS. See BICMOS LOGIC CIRCUITS.