Differential current switch—High performance at low power

by E. B. Eichelberger
S. E. Bello

The recent IBM System/390™ announcement includes six high-performance air-cooled models that use a low-power variation of emitter-coupled logic (ECL). This new logic family, called differential current switch (DCS), uses differential signal pairs to represent logic signals, and combines two-level cascode logic with dotting to implement a complete set of logic circuits. These DCS circuits are described in detail, and the relative value of the DCS and ECL logic families is discussed extensively.

Introduction

Although emitter-coupled logic (ECL) continues to dominate as the primary technology for water-cooled System/390™ products, a variation has been developed that permits high performance in an air-cooled environment. This circuit family, called differential current switch (DCS), is the primary means for implementing logic in the six high-performance air-cooled Enterprise System/9000™ products (Models 190, 210, 260, 320, 440, and 480). These are collectively designated machine type 9121. Another paper in this issue [1] provides some detail regarding the application of DCS to the 9121 design.

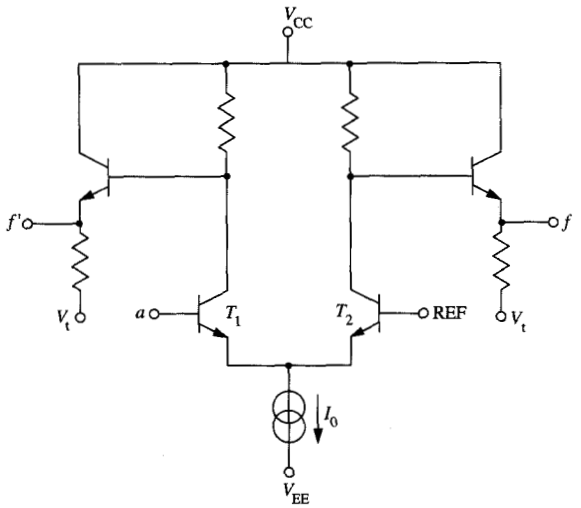
This paper describes DCS, including many of the circuits used to implement key logic functions. The value and limitations of DCS are discussed, and the specific form of implementation on an ECL gate array (masterslice) is also described.

DCS

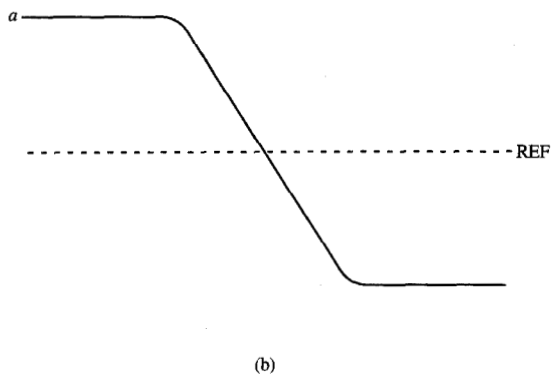
The basic concept of DCS is not new; it has seen limited use, especially in bipolar RAM and in custom bipolar chip designs. In 1975 Breuer [2] published an excellent paper on the value of this circuit family and showed superior delay and power results for a variety of functions. Yuan [3] described a circuit compiler for up to ten cascode levels of DCS in 1985. In 1986 Buckley et al. [4] described a large 32-bit processor chip that used DCS circuits with cascode trees containing up to eight levels.

The primary difference between this paper and previously published work is the focus on a gate array design using a limited number of two-level cascode logic circuits. Also, these logic circuits make use of dot logic as well as N-way selectors and use only one type of differential signal between circuits.

In normal ECL, current is steered in one of two directions by two transistors, one of which has a fixed reference voltage applied to its base (**Figure 1**). The voltages shown in the circuit diagrams are the standard



(a)



(b)

Figure 1

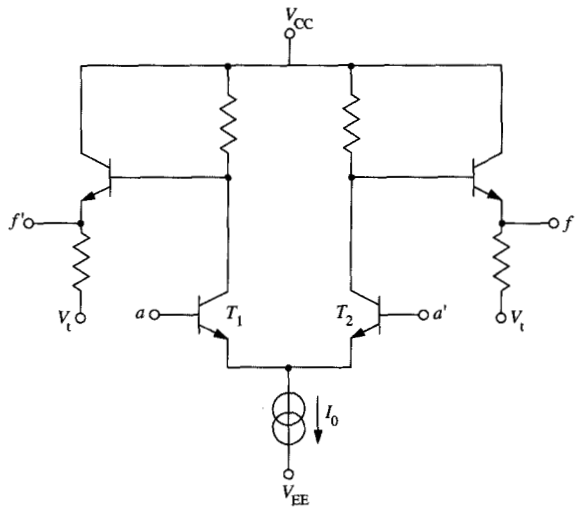
Emitter-coupled logic (ECL): (a) circuit diagram; (b) signal diagram.

ECL voltages: $V_{CC} = 1.4$ V, $V_t = -0.7$ V, $V_{EE} = -2.2$ V. In DCS the circuit is almost identical, but the current-steering transistors have signals of opposite polarities applied to their bases (**Figure 2**). Although this requires a pair of input wires as compared to a single input wire for ECL, it has many positive features.

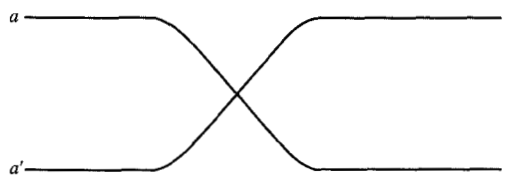
Probably the most important feature is that the nominal DCS signal swing can be reduced to less than 50% of the ECL signal swing. In both cases a voltage difference, V_0, between the base of T_1 and the base of T_2, must be maintained to reliably switch the current from one transistor to the other. [The DCS signal swing is less than 50% of the ECL swing because it does not have to provide noise margins that compensate for the variations across the chip of V_{CC}, REF (reference signal), and the current source, I_0.] This reduced signal swing results in better performance and allows acceptable performance and delay variation at very low power levels.

Another significant characteristic of the emitter-follower differential signal is shown in **Figures 3** and **4**. For ECL logic (Figure 3), the rising signal is controlled by the active emitter follower, and is very fast. Unfortunately, the falling signal is determined by the emitter-follower resistor and can be very slow, especially for low-power ECL circuits. For DCS circuits (Figure 4) both transitions are the same, and are dominated by the rising transition of an emitter follower. This improves performance and reduces delay variation, thus allowing the use of very low-power DCS circuits.

Finally, the noise sensitivity of DCS is worth discussing. It is evident from Figure 2 that any noise on V_{CC} is transmitted to both output signals f and f' in both CS and DCS. However, because the DCS outputs feed



(a)



(b)

Figure 2

Differential current switch (DCS): (a) circuit diagram; (b) signal diagram.

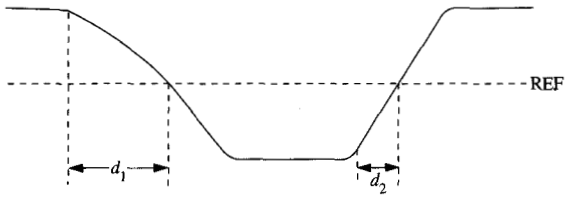


Figure 3

ECL—loading delay.

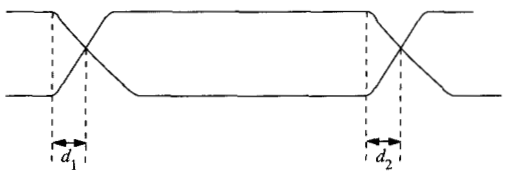


Figure 4

DCS—loading delay.

differentially to succeeding stages, they are self-canceling for the noise. The single ECL signal, however, propagates the noise. Also, during switching, the V_{CC} current change for ECL is an order of magnitude larger than for DCS (**Figure 5**), introducing delta I noise. Coupled noise for DCS is also less than half that for ECL because of the reduced signal swing.

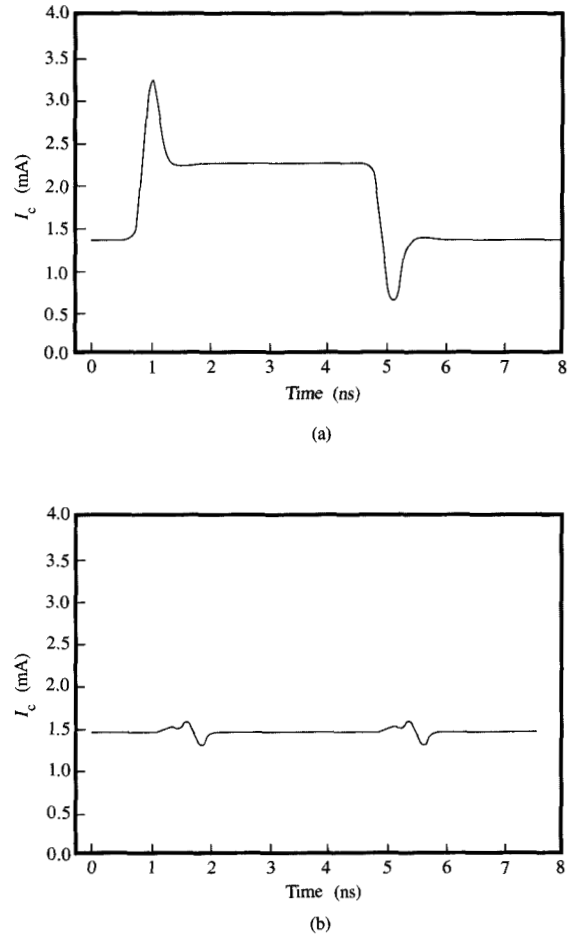


Figure 5

V_{CC} current: (a) for ECL; (b) for DCS.

DCS logic circuits

Although the simple powering circuits for DCS and ECL are almost identical (Figures 1 and 2), DCS logic circuits are considerably different from ECL logic circuits. This is due to the fact that the DCS differential signal pair can only be used to switch a current source, as shown in Figure 2. The logic must therefore be performed with the switched current.

• *Two-way selector*

One way of achieving a useful logic function is to use a two-level cascode circuit, as shown in **Figure 6**. In this case, one DCS input signal b, b' is translated to a lower voltage by using two emitter followers (T_3 and T_4). These translated signals then feed the bottom transistor pair, T_1 and T_2, and switch current either to T_5 and T_6 or to T_7 and T_8. These transistor pairs are in turn controlled by two other DCS signals a_1, a_1' and a_2, a_2'. This results in the very useful two-way selector function, with signal b serving as the selecting input.

The logic function performed by this cascode circuit is determined by the input conditions that cause the current I_0 to flow through the resistor R_1'. This happens when a_1 and b are both positive or when a_2 and b' are both positive. Thus,

$$f = a_1 b + a_2 b'. \tag{1}$$

The current I_0 being switched in this circuit is generated by the transistor T_0 and emitter resistor R_0. The signal V_X feeding the base of T_0 controls the voltage across R_0, which in turn controls the value of I_0.

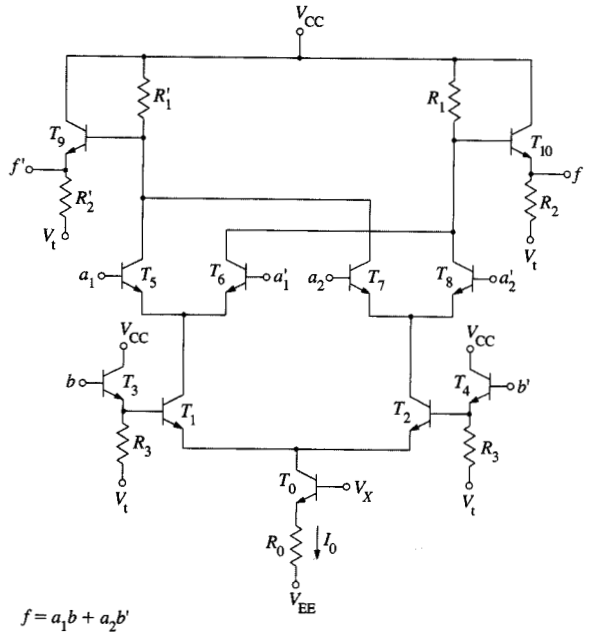


Figure 6

DCS two-way selector.

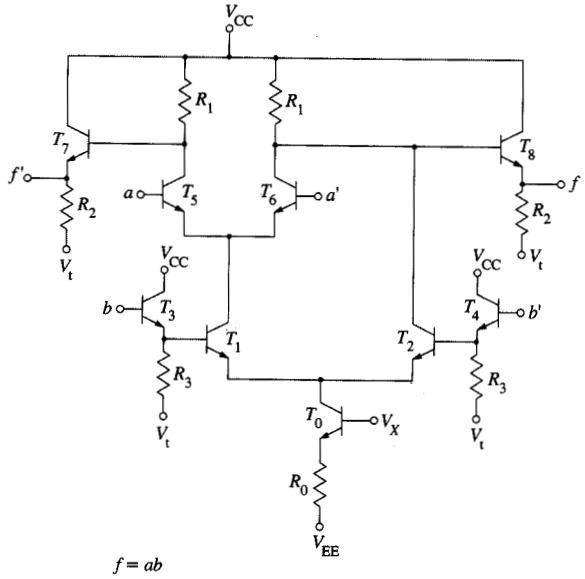


Figure 7

DCS multifunction circuit.

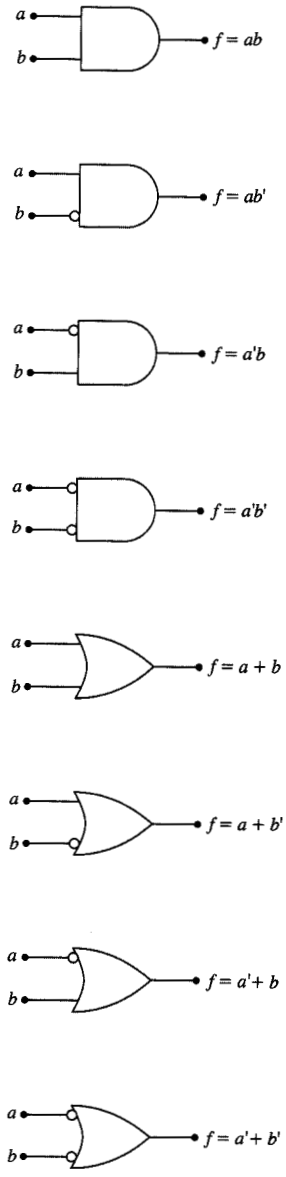


Figure 8

Multifunction circuit logic functions.

- *Exclusive-OR*

This selector circuit can also be used as an exclusive-OR circuit by feeding the a_1 input with the same signal pair that feeds a_2, except with inputs swapped. That is, if

$$a_1 = a'_2 , \tag{2}$$

then

$$f = a_2'b + a_2b'. \quad (3)$$

This use of the selector as an exclusive-OR circuit demonstrates a convenient feature of DCS, which is that the complement of any DCS signal can be obtained by swapping the two wires carrying the signal.

Finally, a few words about the input-translate emitter followers (T_3 and T_4). Although this configuration adds four components and some power consumption to the DCS circuit, it allows the logic signals feeding the upper and lower inputs of the cascode circuit to have identical voltage levels. This solves a major problem in designing with DCS and reduces the number of circuit types. Also, because the input emitter follower has a low capacitance load, its power can be made very small.

• *Multifunction*

A third DCS logic circuit, which we call the "multifunction" circuit, is shown in **Figure 7**. With the inputs and outputs as shown, the circuit performs the logic functions

$$f = ab. \quad (4)$$

Using the technique of complementing the input or output signal by swapping wires, the circuit can implement the eight functions shown in **Figure 8**. These functions, combined with the exclusive-OR function of Figure 6, represent all possible nontrivial functions of two variables.

• *Dot logic*

Probably the most useful logic function in ECL is the emitter-follower OR dot. It is fast, and substantially reduces power. The same dot function can be implemented between two or more DCS circuits with the same advantages. Unfortunately, there are two significant differences for DCS. **Figure 9** shows two multifunction circuits dotted together to form a logical OR on the output. The f output is an emitter-follower dot as in ECL, but the f' output requires a collector dot between the transistors fed by a_1 and a_2.

This collector dot performs a positive AND dot function of the negative outputs of two multifunction circuits. This has the added complexity of a second dot connection, and has a small impact on performance due to the collector node sensitivity to capacitance. To minimize this performance impact, the dotted circuits should be close together, and the collector dot net should be as short as possible.

In spite of these problems, the DCS dot function is very important; it substantially improves performance, reduces power, and reduces circuit count. Although Figure 9 shows a dot of only two circuits, it is possible to dot many DCS circuits together as in ECL. Of course, as the number of circuits becomes large, the capacitance of the collector node adds significantly to the delay time.

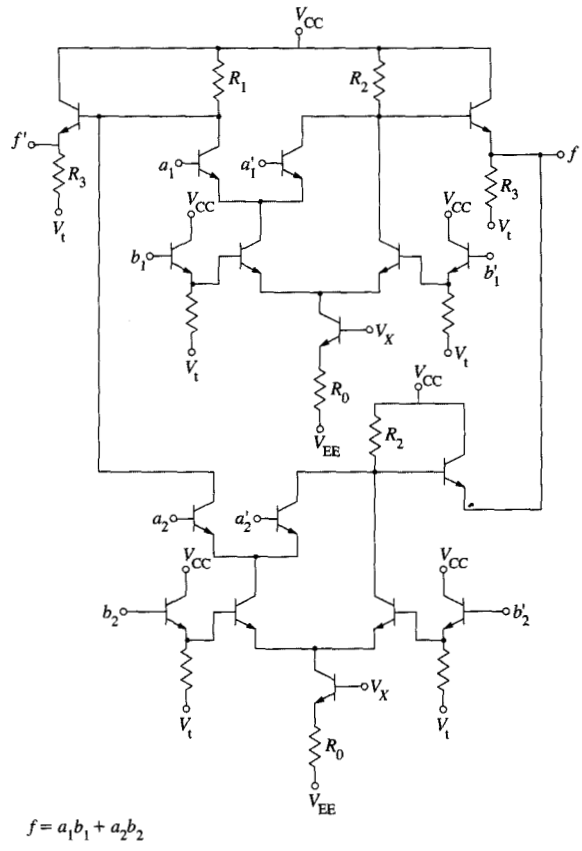


Figure 9

Two multifunction circuits with output OR dot.

Because all DCS circuits can be dotted together, there are a rich variety of logical functions possible with the type of dot shown in Figure 9. It is also possible to implement more complex dot functions by utilizing collector AND dots that feed emitter-follower OR dots for both the true and the complement outputs. In general, any logic function can be implemented in this way, but as the two-level dot functions become more complicated, the number of DCS circuits feeding the dots can become quite large. Thus, only relatively simple two-level dot functions are worth implementing in this way. Two such functions are a two-way exclusive-OR dot and a two-way selector dot.

• *N-way selector and selector driver*

Another very useful function that can be implemented with two-level cascode logic is an N-way selector. An example

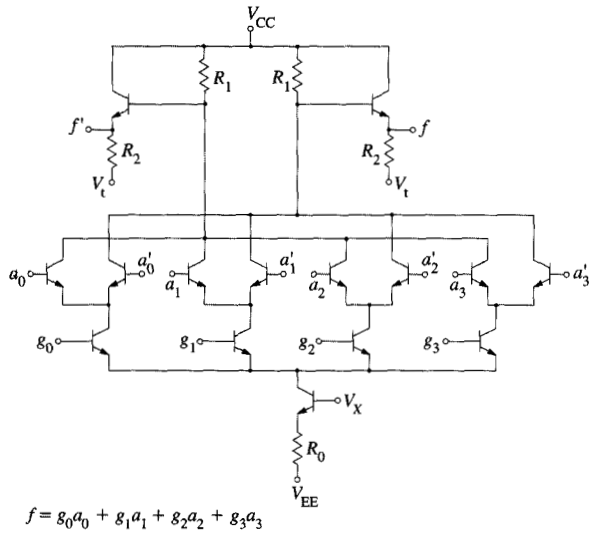


Figure 10

Four-way selector (requires four-way selector driver).

of such a circuit for $N = 4$ is shown in **Figure 10**. For this circuit four gating signals g_0, g_1, g_2, and g_3 must be generated by a selector-driver circuit that always has exactly one of the four outputs positive while the other three outputs are negative.

An example of such a four-way selector driver is shown in **Figure 11**. This circuit uses collector AND dots on the outputs of four simple DCS circuits to generate the one-out-of-four signals to drive the selector. The required lower voltage levels for the selector driver are obtained by connecting all four collector resistors R_2 to a single shift resistor R_1. The sum of the currents of all four current sources flows through R_1, thus shifting the output signals negative.

The four-way selector (Figure 10) operates in a manner similar to the two-way selector. When one of the lower gating inputs, say g_0, is positive, the current from the current source flows through the transistor controlled by g_0 and allows the corresponding DCS input, in this case a_0, to control the current and determine the state of the circuit differential outputs f and f'.

• *Latch*

There is no doubt that the most valuable application of DCS circuits is in the implementation of a D-latch (polarity hold latch). The function of a D-latch is to store data and to allow system clocks to control the operation of the logic network. The operation of the D-latch (**Figure 12**) is simple. When the clock signal c is "on," the latch output L is controlled by the data input d. When the clock turns "off," the value at the data input is stored in the latch, and the latch output is stable at this value as long as the clock is "off." Since the clock input c to the latch is driven by a clock driver circuit, it can be designed at the lower voltage level to avoid the need for an input-translate emitter follower.

Although the D-latch is a simple circuit, its operating characteristics are extremely important in system performance. The D-latch has a critical race if it is designed using standard logic circuits. This race occurs as the clock turns off, and can cause the latch to malfunction if it is not designed correctly. The cascode DCS latch (Figure 12) simply does not have a race problem. As the clock turns off (with d positive), the current switches from T_1 and T_4 to T_2 and T_6, and continues to flow through the collector resistor R_{11}, which feeds the output emitter follower T_7 and keeps it negative. This in turn ensures that T_5 is kept off, and prevents any of the current from flowing to the collector resistor R_{12}, feeding T_8. This ensures that

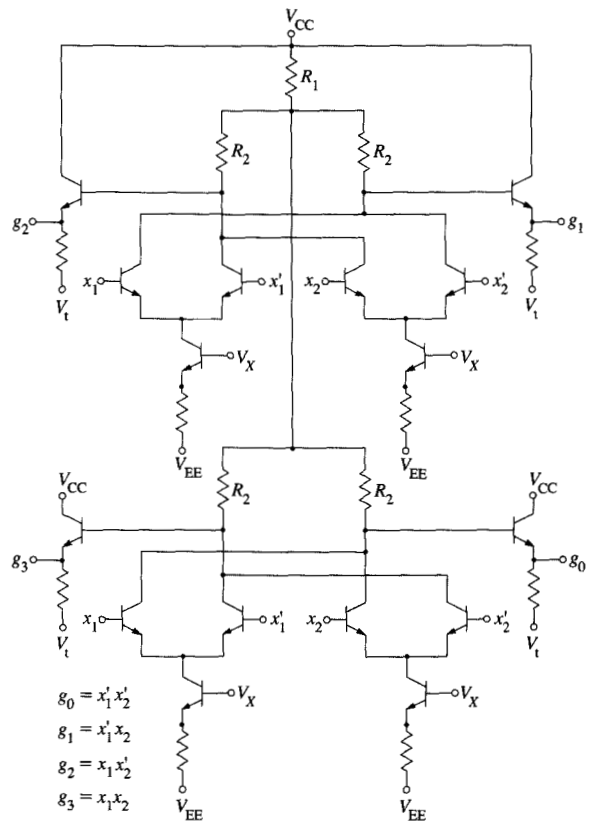


Figure 11

Four-way selector driver.

the differential latch output signal always remains positive even though there is a small positive transient on the negative latch output as the clock turns off. This transient is most often caused by the "turning on" of transistor T_6 and is not large enough to cause any problem.

Although the D-latch shown in Figure 12 has only one data input, it is possible to create latches with two or more data inputs by using the techniques described for N-way selectors (Figure 10).

- *Convert circuits*

The DCS circuits shown in this section all have DCS input and output signals. There will, in general, be a mixture of ECL circuits and DCS circuits in a system, if for no other reason than to communicate between chips (i.e., off-chip drivers have ECL swings). Thus, there is a requirement for one or more types of circuits with ECL inputs and DCS outputs, as well as for circuits with DCS inputs and ECL outputs. All of the DCS circuits shown in this section could also be designed to have ECL output levels. All ECL logic circuits with both true and complement outputs can also be designed to have DCS output levels. Thus, the only problem is deciding how many translate (convert) circuits to design.

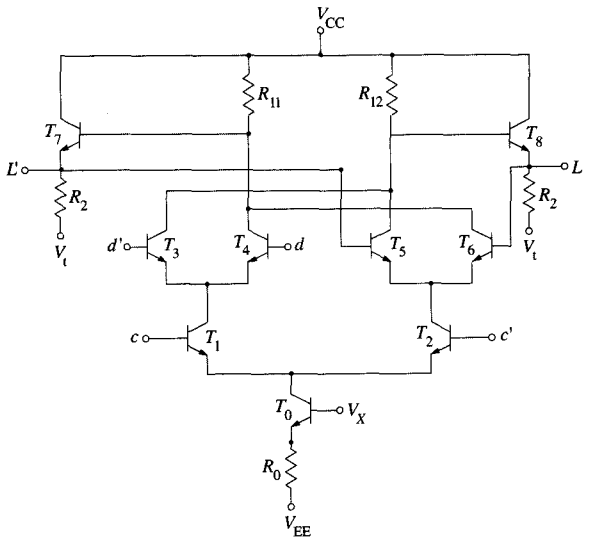


Figure 12

D-latch.

DCS implementation

The DCS circuits described in the preceding section were implemented on a gate array containing 5200 cells. The cells in the gate array were designed to support both ECL and DCS circuits. Although this required more components per cell and resulted in a loss of circuit density, it had the advantage of being able to mix both ECL and DCS circuits on a single chip, and also simplified design. In retrospect this was a good trade-off. The cells were designed to support a four-input, two-output ECL circuit, and two or more cells were used for all DCS circuits. This two-to-one area ratio seemed to work quite well from both the circuit design and wirability points of view.

ECL vs. DCS

How does DCS compare with ECL in performance, power, and circuit density? This question is not easy to answer; in fact, it has multiple answers depending on what function is being designed and how well it is designed in both ECL and DCS. Fortunately, significant data exist, since over 95% of the circuits in these systems were DCS, and the function is similar to that of a comparable ECL system.

Performance can best be evaluated by machine cycle time or critical path delays. For both of these measurements, the performance of the package and off-chip drivers also has a major impact. Thus, to compare logic circuit families the best method is to compare critical paths that do not involve package delays. When this is done for a number of functions, the DCS and ECL delays are usually about the same, with a few exceptions where DCS is significantly faster. In general, the DCS circuits are faster, but the critical paths tend to have more stages of logic with DCS. The critical path delays can be influenced very strongly by the skill of the logic designer. Also, the DCS cascode latch is significantly faster than its ECL counterpart, and has superior "set plus hold" characteristics (period of time during which data input must not change before and after the clock turns off).

Power is easier to evaluate than performance, and it is the most important attribute of DCS. DCS designs operate at lower power for two reasons:

1. Most DCS circuits use fewer current sources, and lower-power emitter followers, than do equivalent ECL circuits.
2. DCS circuits can be designed to operate at very low power. Such circuits can then be used wherever higher performance is not required.

The very low-power DCS circuits were used extensively in the 9121 design, since the CMOS cache dictated the cycle time. This resulted in an internal logic circuit power of less than half what it would have been for an ECL design.

DCS is not as good as ECL in circuit density. The fact that the two-input OR gate requires twice as much area in DCS as in ECL is enough to convince most engineers that

DCS area overhead is unacceptable. However, actual experience yielded a surprising result: The actual cell count for the DCS-designed 9121 machine was only 20% larger than for the ECL design, and almost all of the 20% overhead is accounted for in the circuits used to convert between ECL and DCS signals.

Some of the factors that minimized DCS circuit density overhead are as follows:

1. The DCS latch is about half the size of the logically hazard-free ECL latch.
2. The exclusive-OR and selector functions are about the same size as their counterpart ECL circuits.
3. There are no inverters required in DCS.
4. High-performance ECL designs frequently use a large number of high-fan-in circuits to reduce the stages of delay in the critical paths. The DCS designs can achieve the same performance with more stages of delay, thus using fewer circuits.

It is also probably worth noting that the 20% increase in cell count (chip area) was not a problem in the 9121 design because the chip designs were "I/O bound" and only used about 50% of the available circuits on the gate array chips.

Summary

The IBM System/390 family of processors includes six high-performance air-cooled models that use a new low-power variation of ECL logic. This new logic family, called DCS (differential current switch), uses differential signal pairs to represent logic signals and combines two-level cascode logic with dotting to generate a new set of DCS logic circuits. These circuits are described in detail, and their relative value in comparison with ECL is discussed.

DCS logic allowed a high-performance system to be designed at less than half the power required for an equivalent ECL design, with no significant loss of performance, and with a circuit density overhead of about 20%. The new circuits were implemented on a gate array that supports both ECL and DCS logic.

Acknowledgments

The authors wish to thank the original members of the Kingston DCS team, including Rolf Bergenn, Bill Chu, Matt Graf, John Hoffman, John Ludwig, Rick Rizzolo, and Vito Russo, for early design of the circuit, the cell, and the design automation system. We would also like to thank Arnold Weinberger and Bill Heller from Poughkeepsie for important insights into logic efficiency and wirability, and the Fishkill team of Arnie Barish, Leo Freeman, Bob Incerto, Mark Mayo, Rocco Robortaccio, and many others for this early evaluation and innovative inclusion of DCS into the ECL master slice. Finally, we would like to acknowledge the high level of technical assistance we received in the Kingston, Poughkeepsie, and East Fishkill laboratories.

References

1. V. L. Gani, M. C. Graf, R. F. Rizzolo, and W. F. Washburn, "IBM Enterprise System/9000 Type 9121 Air-Cooled Processor Technology," *IBM J. Res. Develop.* **35,** 342–351 (1991, this issue).
2. D. R. Breuer, "Applications of Low-Level Differential Logic," *Proceedings of the IEEE International Solid-State Circuits Conference*, 1975.
3. Bob Yuan, "A Circuit Compiler for CML Stack Synthesis," *VLSI Design* (January 1985).
4. F. Buckley et al., "A Bipolar 326 Processor Chip," *Proceedings of the IEEE International Solid-State Circuits Conference*, 1986.

Received January 21, 1991

Edward B. Eichelberger *IBM Data Systems Division, Neighborhood Road, Kingston, New York 12401.* Dr. Eichelberger is an IBM Fellow and the manager of the Advanced VLSI Technology and Testing Department at the Kingston facility. He earned his B.S. degree from Lehigh University and his Ph.D. in electrical engineering from Princeton University. He joined IBM in Endicott, New York, in 1956 and has worked in areas of VLSI chip design, circuit design, design automation, and design for testability. Dr. Eichelberger has received three IBM Division Awards and two IBM Corporate Awards. He has reached his seventh IBM Invention Plateau, and holds 20 U.S. patents. Dr. Eichelberger is a Fellow of the Institute of Electrical and Electronics Engineers, and shared the 1989 IEEE W. Wallace McDowell Award with T. W. Williams.

Stephen E. Bello *IBM Data Systems Division, Neighborhood Road, Kingston, New York 12401.* Mr. Bello is an IBM Senior Technical Staff Member with expertise in VLSI, processors, design automation, and parallel processing. He received a B.E.E.E. degree from New York University in 1972 and an M.S. in management of technology from the Massachusetts Institute of Technology in 1988. Mr. Bello has received two Outstanding Innovation Awards, an Outstanding Technical Achievement Award, a Division Award, and a Management Excellence Award, and he has been elected to the IBM Academy of Technology.