AN–709
Application Note

MECL 10,000 ARITHMETIC ELEMENTS MC10179, MC10180, MC10181

by
Tom Balph
Computer Applications

The MECL 10,000 arithmetic functions include a 4-bit arithmetic unit, a dual adder/subtractor, and a lookahead carry block. This application note describes the devices and shows their operation in larger system configurations.

MOTOROLA *Semiconductor Products Inc.*

MECL 10,000 ARITHMETIC ELEMENTS
MC10179, MC10180, MC10181

INTRODUCTION

The designer of a high speed digital system may have the need to perform arithmetic operations on binary words. The MECL 10,000 family of integrated circuits has several arithmetic elements among its many versatile functions. These include the MC10180 dual adder/subtractor, the MC10181 4-bit arithmetic logic unit/function generator, and the MC10179 lookahead carry block.

Arithmetic operations on binary numbers use two basic single-bit computing elements. These elements are the full adder (FA) and the half adder (HA). To describe the adders use the following nomenclature:

A = augend
B = addend
S = sum
C_{in} = input carry
C_{out} = output carry

The more basic element is the half adder. It is just a two input element producing the sum and output carry of the augend and addend. (The augend and addend are the two bits being added together). The adder equations are:

$$S = A \oplus B = A\overline{B} + \overline{B}A \quad (1)$$

$$C_{out} = A \cdot B \quad (2)$$

If the half adder is extended to include the addition of an input carry, the result will be the full adder. The FA adds augend, addend, and input carry producing sum and output carry. The FA equations are:

$$S = A \oplus B \oplus C_{in} = \overline{C}_{in}(A\overline{B} + \overline{A}B) + C_{in}(AB + \overline{A}\overline{B}) \quad (3)$$

$$C_{out} = (AB) + (AC_{in}) + (BC_{in}) \quad (4)$$

The symbolic representations of the HA and FA are shown in Figure 1. These two adders are the basic configurations around which the MECL arithmetic elements are designed. Addition and subtraction along with variations on both use the FA and HA as the basic computational tool.

It is interesting to note an unusual feature of the full adder equations. The sum and output carry equations are the same in positive and negative logic. If the sum

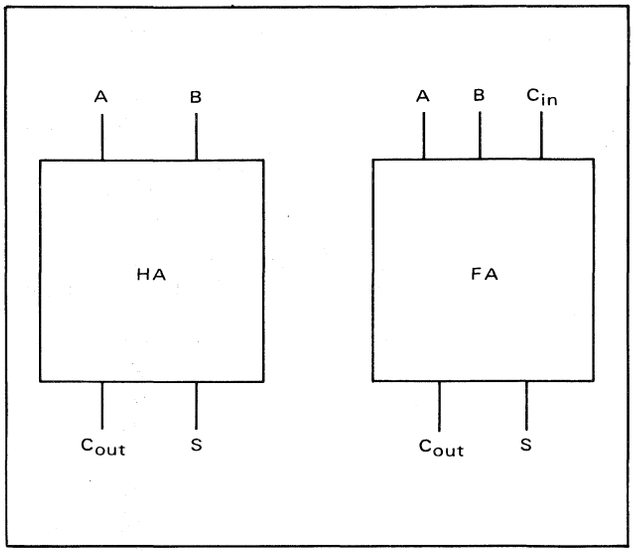


FIGURE 1 – Symbolic representations of the half adder and full adder.

equation is known in positive logic (i.e., the more positive level is a "one" or true state), the negative logic equation can be found. Using the same variable assignments, to convert to negative logic (i.e., the less positive level is a "one" or true state), substitute the complements of the variables into the equation. Equation (3) becomes in negative logic:

$$\overline{S} = C_{in}(\overline{A}B + A\overline{B}) + \overline{C}_{in}(\overline{A}\overline{B} + AB) \quad (5)$$

Addendum "A" demonstrates the development of equation (5) to

$$S = \overline{C}_{in}(A\overline{B} + \overline{A}B) + C_{in}(AB + \overline{B}\overline{A})$$

Which equates to the original sum equation in positive logic.

For the carry-out equation, the negative logic equation becomes:

$$\overline{C}_{out} = \overline{C}_{in}\overline{A} + \overline{C}_{in}\overline{B} + \overline{A}\overline{B} \quad (6)$$

after substituting the complements of the variables.

Simplifying equation (6) as in Addendum "B" yields

$$C_{out} = C_{in}A + C_{in}B + AB$$

Equation (6) is the same as (4) therefore, the positive and negative logic equations for C_{out} are the same. This result will be used later on to describe the MECL 10,000 arithmetic elements in both logic definitions.

The MC10180, MC10181, and MC10179 each have different capabilities. The following device descriptions will explain operation of each device.

Device Description

MC10180

The MC10180 is a dual full adder/subtractor. Each adder is provided with inputs of operand A, operand B, (operand refers to the binary word being added or subtracted) and carry-in and with outputs of sum, $\overline{sum}$, and carry out. The add/subtract operation is determined by two select lines common to each adder. The mode of operation is selected as shown in Figure 2 which provides the block diagram and function select table for the MC10180.

In examining operation of the device more closely (Figure 3), the configuration is logically a full adder with an exclusive-NOR function on operand A and on operand B. Either operand may be selectively inverted or rippled-through to the adder. In this manner, the subtraction of either operand can be provided.

The positive logic equations for the MC10180 are:

$$S = \overline{C}_{in}(\overline{A}'B' + A'\overline{B}') + C_{in}(A'B' + \overline{A}'\overline{B}') \quad (7)$$

$$C_{out} = C_{in}A' + C_{in}B' + A'B' \quad (8)$$

where

$$A' = \overline{A \oplus Sel_A} = A \odot Sel_A \quad (9)$$

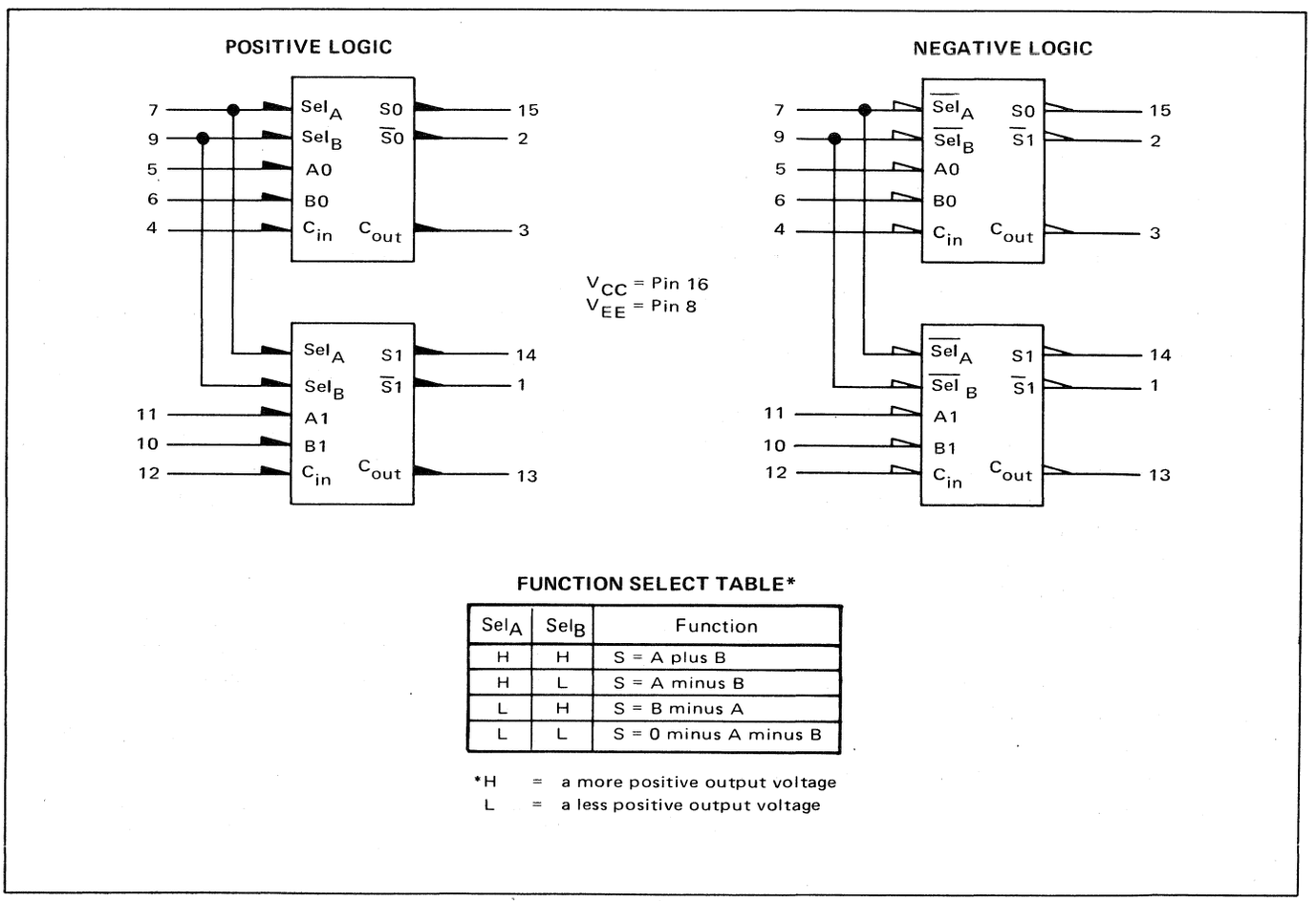
$$B' = \overline{B \oplus Sel_B} = B \odot Sel_B \quad (10)$$

Equations (7) and (8) are for the sum and carry-out of the full adder in terms of C_{in}, A′, and B′. A′ and B′ are the operands A and B modified by the exclusive-NOR function.

It was shown that the full adder equations are similar in both logic definitions. Because the positive and negative logic equations for the FA are the same, the MC10180 may be used similarly in both logic definitions. However, the negative logic equations for A′ and B′ must become:

$$A' = A \oplus \overline{Sel_A} = A \odot Sel_A \quad (11)$$

$$B' = B \oplus \overline{Sel_B} = B \odot Sel_A \quad (12)$$



FUNCTION SELECT TABLE*

Sel_A	Sel_B	Function
H	H	S = A plus B
H	L	S = A minus B
L	H	S = B minus A
L	L	S = 0 minus A minus B

*H = a more positive output voltage
L = a less positive output voltage

FIGURE 2 – Block diagram and function select table for MC10180.

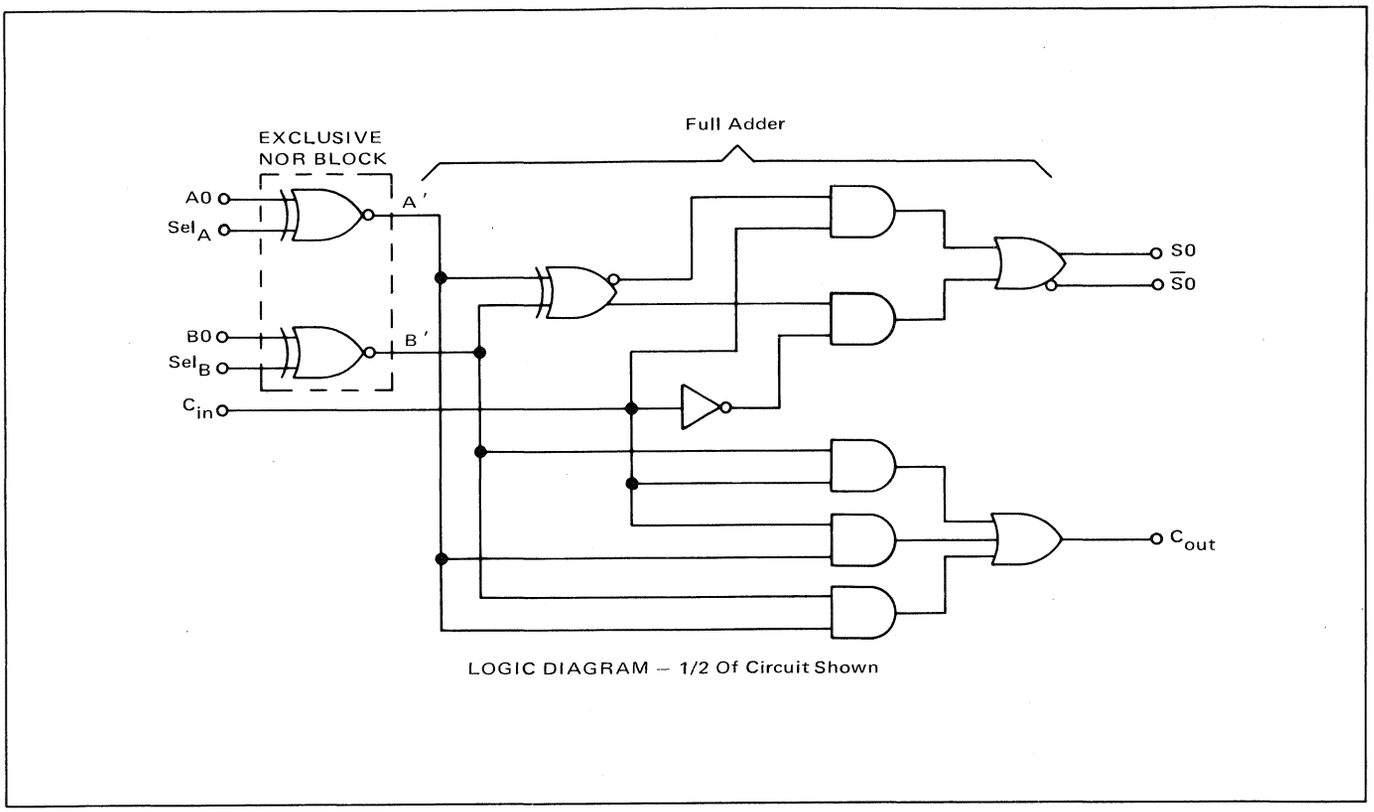


FIGURE 3 – Positive Logic Diagram for MC10180

The exclusive-NOR function becomes an exclusive-OR function in negative logic. Therefore, the select lines are redefined to $\overline{Sel_A}$ and $\overline{Sel_B}$ for negative logic. Equations (7) and (8) still hold.

MC10181

The MC10181 is a more complex device than the MC10180. The MC10181 is an arithmetic logic unit capable of performing 16 logic operations and 16 arithmetic operations on two 4-bit operands (Figure 4). It has inputs for a 4-bit operand A, a 4-bit operand B, and carry-in and has outputs for a 4-bit function-out, carry-out, and group carry propagate and carry generate.

The logic or arithmetic operation of the ALU is determined by internal logic controlled by the mode input. In the arithmetic mode, full internal lookahead carry is incorporated to determine the 4-bit function-out and carry-out. The mode control disables the internal carry circuitry for logic mode operation, and carry is ignored. In the logic mode, the function-out is generated on a bit per bit basis.

Each mode of operation has 16 function capability (32 total). The four select lines (S_0 through S_3) determine which function will be performed on operands A and B in either mode of operation.

When the mode control is in the more positive (high) logic state, the MC10181 is in the logic mode of operation and can perform any of 16 possible logic operations on two variables. These functions are especially useful for computer processor applications where the processor must execute commands such as generation of all 1's or 0's, "mask" the contents of a register, OR or AND the contents of two registers, or similar logic operations.

The arithmetic mode is also composed of 16 functions. These functions perform various forms of addition and subtraction on operands A and B. The different functions are useful for a variety of operations. As examples, the A times 2 function may be used for an arithmetic shift in a processor, the A minus 1 function may do a decrement operation, the two's complement of minus 1 may be generated, and the A plus 0 function may either ripple-through word A or increment word A by adding a carry-in.

The operation of the device may be seen more clearly by examining the negative logic diagram of Figure 4A. For each bit, two internal functions may be defined:

$$G_i = (S_2 A_i \overline{B}_i) + (S_3 A_i B_i) \quad (13)$$

$$P_i = A_i + S_0 B_i + S_i \overline{B}_i \quad (14)$$

where i = 0 through 3. G_i and P_i are the operands A and B modified by the S_0, S_1, S_2, and S_3 control lines. The two functions are independent of the mode control and hold for both modes of operation.

The function-out is then written in terms of G_i, P_i, and carry. In the logic mode (M = high state), the function-out is dependent only on G_i and P_i as the carry logic is disabled. The negative logic equation for function-out becomes:

$$F_i = \overline{(G_i \oplus P_i)} = G_i \odot P_i \quad (15)$$

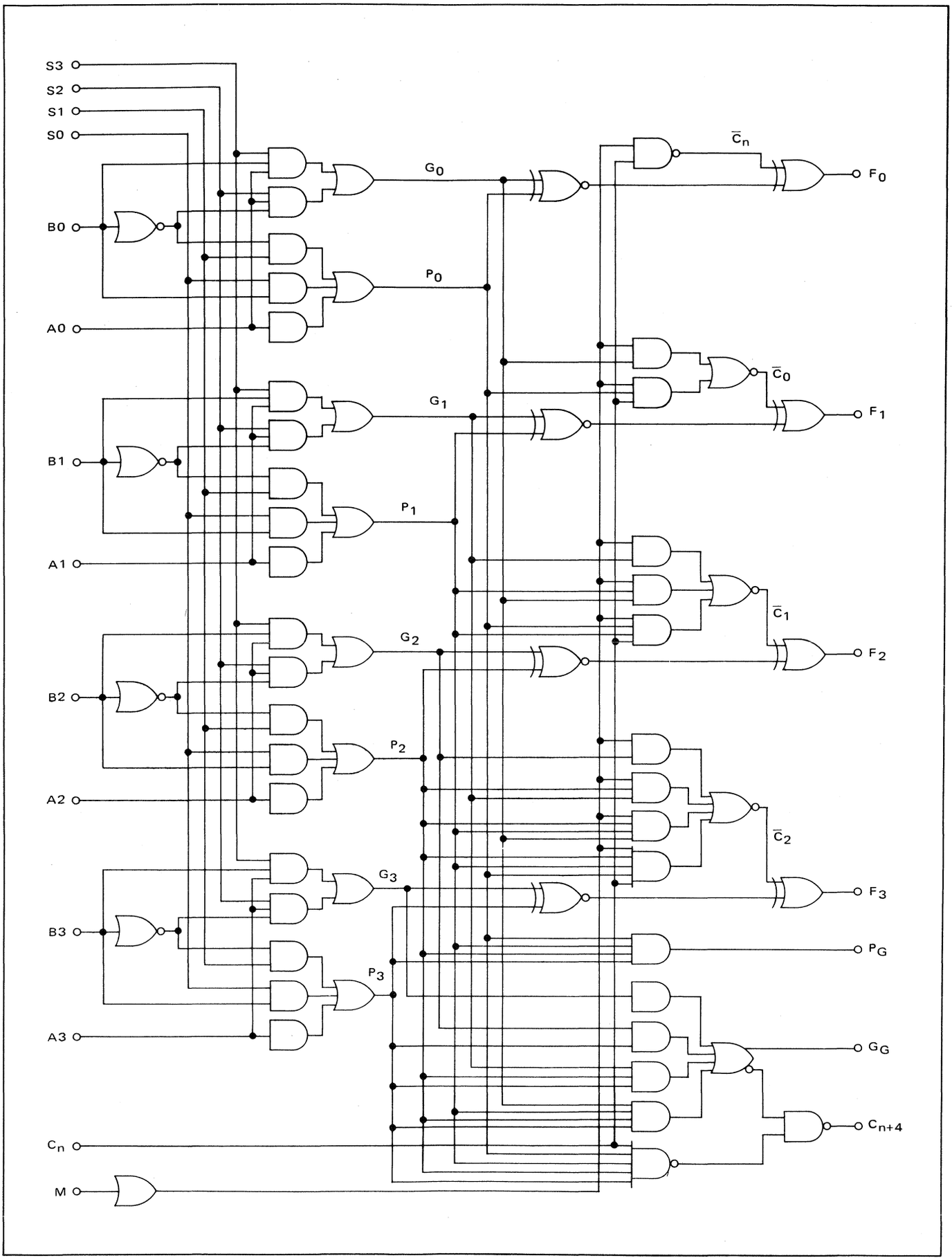


FIGURE 4A – MC10181 NEGATIVE LOGIC DIAGRAM

For arithmetic operation the internal lookahead carry is not inhibited. The equation then becomes:

$$F_i = \overline{(G_i \oplus P_i)} \oplus \overline{C}_{i-1} \quad (16)$$

where

$$\overline{C}_{i-1} = \overline{G_{i-1} + (P_{i-1}\,\overline{C}_{i-2})} \quad (17)$$

Equation (17) is the expression for carry in terms of lookahead techniques. Lookahead techniques may be extended to secondary levels beyond the basic MC10181. The group carry propagate and carry generate are provided for this purpose to be used with the MC10179. That is:

$$P_G = P_0P_1P_2P_3 \quad (18)$$

$$G_G = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 \quad (19)$$

$$C_{n+4} = G_G + P_3P_2P_1P_0C_n \quad (20)$$

Secondary lookahead with the MC10179 can significantly reduce add times on long words. More will be said later on lookahead carry arithmetic.

The MC10181 has been described in terms of negative logic. For positive logic operation, the functions become altered as shown in Figure 4B. The 16 functions for logic operation are altered due to the redefinition in logic levels. However, arithmetic operation is somewhat similar in positive and negative logic. The differences in the arithmetic functions are due to the logical expressions contained within arithmetic equations.

To show this, use the conditions of S_0 = H, S_1 = H, S_2 = H, S_3 = L, and M = L. The arithmetic expression for negative logic is:

F = A plus (A·B)

The equivalent arithmetic function in positive logic is:

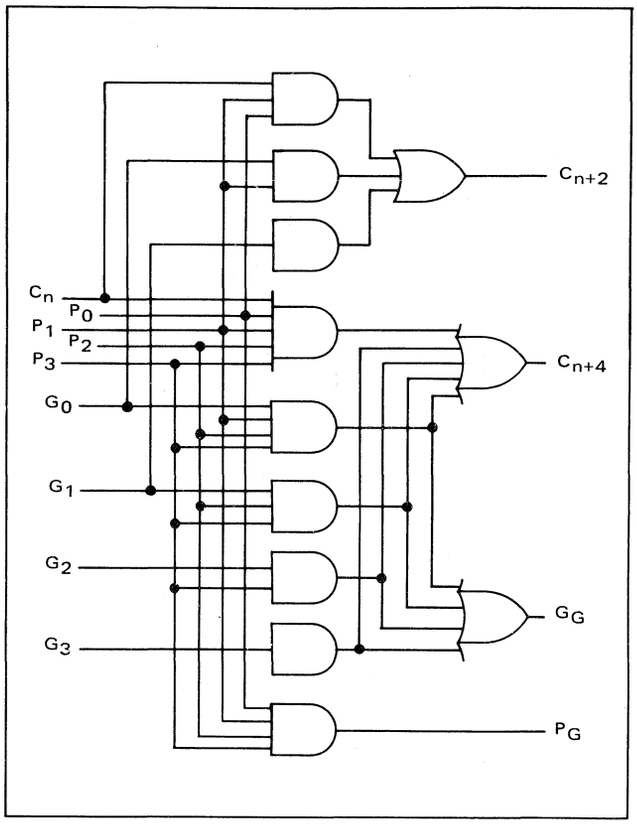


FIGURE 5 – Negative Logic Diagram for MC10179.

F = A plus (A + B)

The only difference in the two functions is that the logical expression A·B changes to A + B.

MC10179

The MC10179 is a logic block that performs the lookahead carry function for the previously described MC10180 and MC10181. The carry generate and carry propagate inputs are used to produce carry outputs external to the adders. The simultaneous production of

POSITIVE LOGIC

Function Select $\overline{S3}$	$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	Logic Functions M is High C = D.C. F	Arithmetic Operation M is Low C_n is low F
L	L	L	L	$F = \overline{A}$	F = A
L	L	L	H	$F = \overline{A} + \overline{B}$	F = A plus (A • $\overline{B}$)
L	L	H	L	$F = \overline{A} + B$	F = A plus (A • B)
L	L	H	H	F = Logical "1"	F = A times 2
L	H	L	L	$F = \overline{A} \bullet \overline{B}$	F = (A + B) plus 0
L	H	L	H	$F = \overline{B}$	F = (A + B) plus (A • $\overline{B}$)
L	H	H	L	$F = A \odot B$	F = A plus B
L	H	H	H	$F = A + \overline{B}$	F = A plus (A + $\overline{B}$)
H	L	L	L	$F = \overline{A} \bullet B$	F = (A + $\overline{B}$) plus 0
H	L	L	H	$F = A \oplus B$	F = A minus B minus 1
H	L	H	L	F = B	F = (A + $\overline{B}$) plus (A • B)
H	L	H	H	F = A + B	F = A plus (A + $\overline{B}$)
H	H	L	L	F = Logical "0"	F = minus 1 (two's complement)
H	H	L	H	$F = A \bullet \overline{B}$	F = (A • $\overline{B}$) minus 1
H	H	H	L	F = A • B	F = (A • B) minus 1
H	H	H	H	F = A	F = A minus 1

NEGATIVE LOGIC

Function Select S3	S2	S1	S0	Logic Functions M is High F	Arithmetic Operation M is Low C_n of LSB must be High F
L	L	L	L	$F = \overline{A}$	F = A minus 1
L	L	L	H	$F = \overline{A + B}$	F = A plus (A + $\overline{B}$)
L	L	H	L	$F = \overline{A} \bullet B$	F = A plus (A + B)
L	L	H	H	F = Logical "0"	F = A times 2
L	H	L	L	$F = \overline{A \bullet B}$	F = (A • B) minus 1
L	H	L	H	$F = \overline{B}$	F = (A • B) plus (A + $\overline{B}$)
L	H	H	L	$F = A \oplus B$	F = A plus B
L	H	H	H	$F = A \bullet \overline{B}$	F = A plus (A • $\overline{B}$)
H	L	L	L	$F = \overline{A} + B$	F = (A • $\overline{B}$) minus 1
H	L	L	H	$F = A \odot B$	F = A minus B minus 1
H	L	H	L	F = B	F = (A • $\overline{B}$) plus (A + B)
H	L	H	H	F = A • B	F = (A • $\overline{B}$) plus A
H	H	L	L	F = Logical "1"	F = minus 1 (two's complement)
H	H	L	H	$F = A + \overline{B}$	F = (A + $\overline{B}$) plus 0
H	H	H	L	F = A + B	F = (A + B) plus 0
H	H	H	H	F = A	F = A plus 0

*F outputs of ALU are one's complement of function listed below.

FIGURE 4B – MC10181 Function Select Table

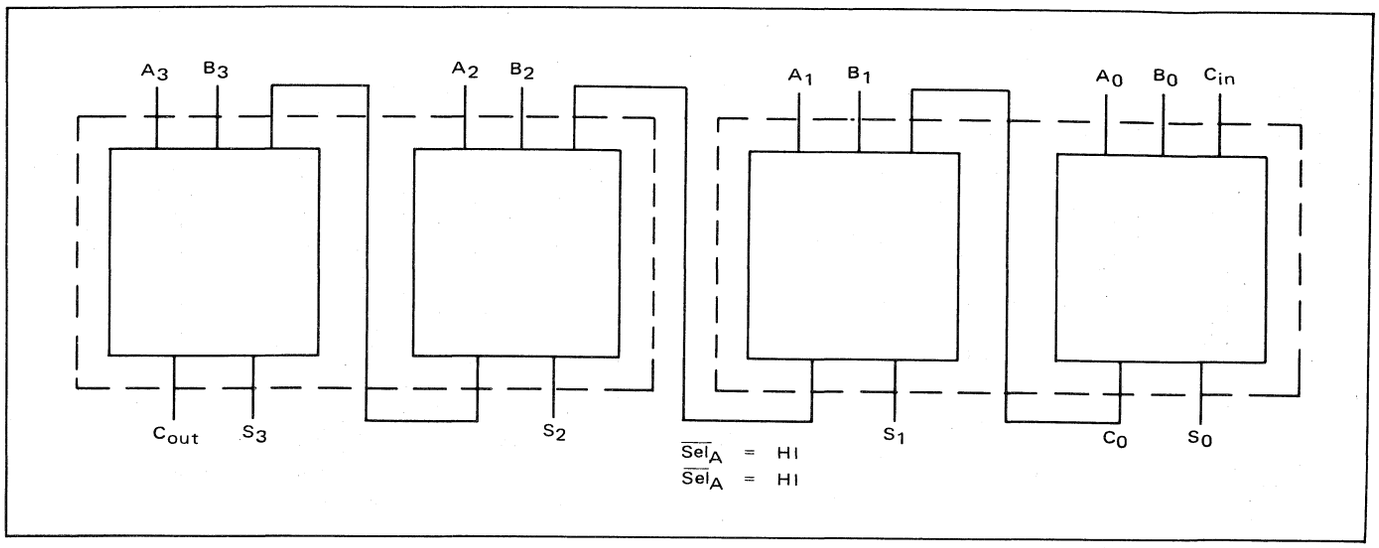


FIGURE 6 – A ripple 4-bit parallel binary adder using two MC10180's.

carry external to the adder greatly reduces add times on long words. Techniques will be shown using both the MC10180 and the MC10181 with the MC10179. The logic diagram of the MC10179 is shown in Figure 5.

The negative logic equations for the MC10179 are:

$$P_G = P_0P_1P_2P_3 \quad (21)$$

$$G_G = (G_0P_1P_2P_3) + (G_1P_2P_3) + (G_2P_3) + G_3 \quad (22)$$

$$C_{n+2} = (C_nP_0P_1) + (G_0P_1) + G_1 \quad (23)$$

$$C_{n+4} = (C_nP_0P_1P_2P_3) + (G_0P_1P_2P_3) + (G_1P_2P_3) + (G_2P_3) + G_3 \quad (24)$$

Although the device has been shown in terms of negative logic, the MC10179 may also be used in terms of positive logic definition. The lookahead carry adders shown in the application note apply for both logic definitions.

PARALLEL ADDERS

A parallel binary adder is used to sum two numbers of n bits. A parallel adder requires n full adders for two n-bit numbers if carry is rippled between adders. Figure 6 illustrates this technique using the MC10180.

The MC10181 may also be used rippling carry between devices. The MC10181 is basically a parallel arithmetic block, 4 bits wide. The device can be used as a parallel adder with carry rippled between devices (every 4 bits). Figure 7 shows this technique.

Another type of parallel adder is the carry lookahead configuration. The basic principle of lookahead addition is the examination of the inputs of each adder stage and the simultaneous production of the proper carries for each of the stages. The overall add time can be significantly reduced from that of a ripple adder.

In a simple block form, the lookahead adder is shown in Figure 8. The group of input bits (number A and number B) is applied to the carry logic block together with the initial carry. The carry outputs are then provided

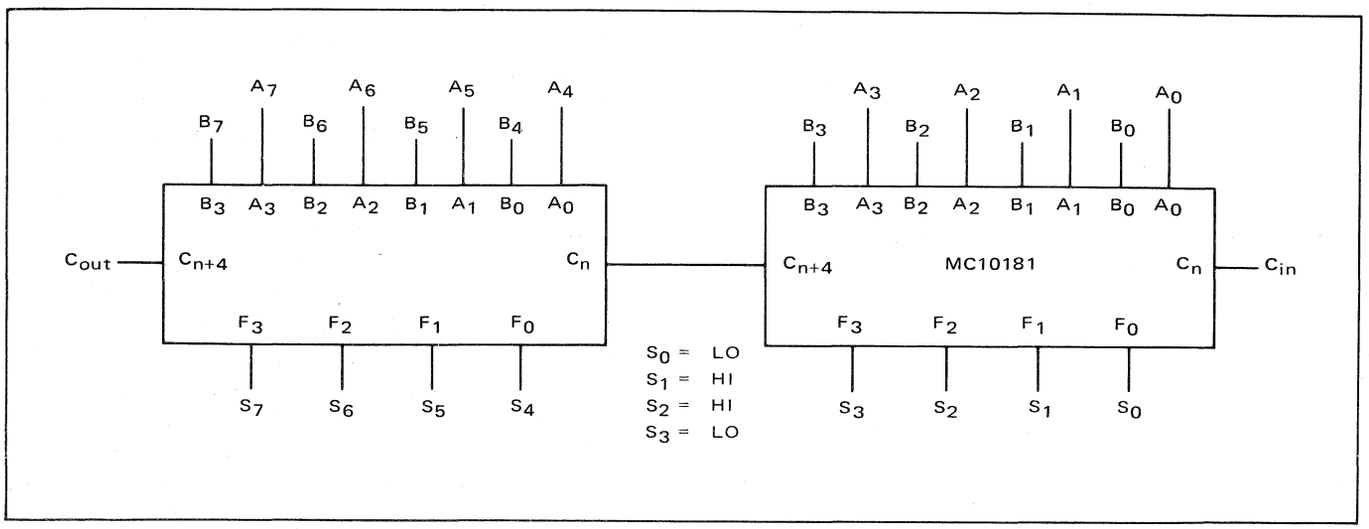


FIGURE 7 – An 8-bit parallel adder using the MC10181 with carry rippled between devices.

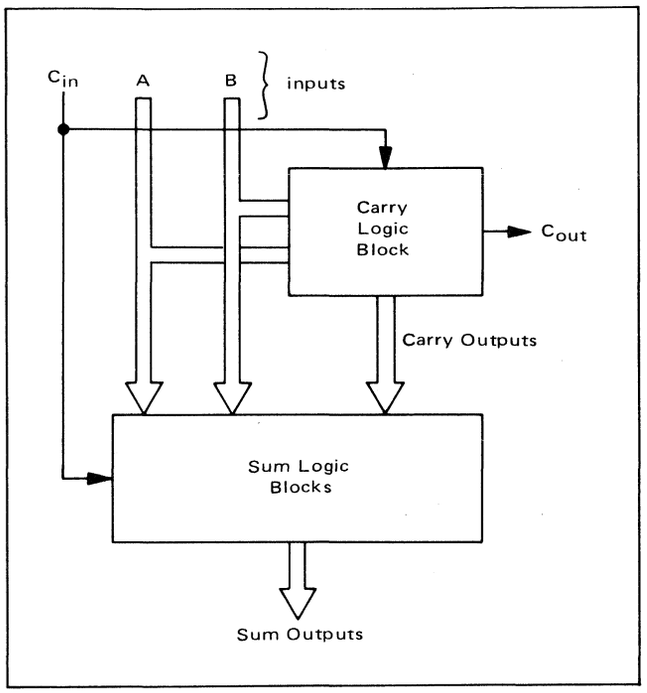


FIGURE 8 – Carry lookahead adder in block form.

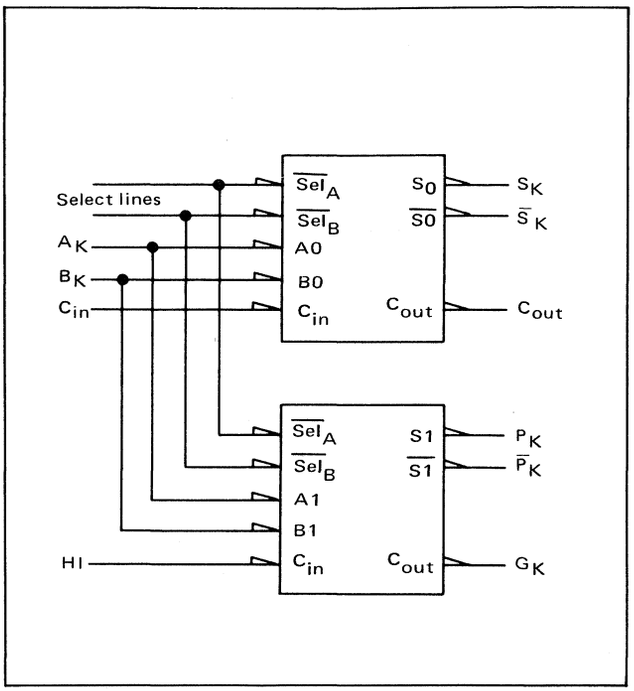


FIGURE 9 – MC10180 connected as a fast carry adder, producing carry propagate and carry generate.

to the sum logic blocks to form the proper sum outputs.

Two new symbols are defined for carry lookahead addition:

$$G_K = A_K \cdot B_K \tag{25}$$

$$P_K = A_K \oplus B_K \tag{26}$$

G_K is the symbol for the carry output bit generated at the Kth bit position and is called the carry generate. The carry generate is produced only when there is a true input on both the A_K and B_K inputs. P_K is the symbol for propagate, which indicates that a carry will be produced if there is a true input on the A_K or the B_K lines and a carry is present from the previous stage. Notice that the equations for G_K and P_K are equivalent to the HA sum and output carry equations.

At the Kth bit position, a carry bit will be generated according to the equation:

$$C_K = G_K + (P_K C_{K-1}) \tag{27}$$

where C_{K-1} is the carry from the previous stage. This equation may be expanded in terms of the carry input to the parallel adder:

$$C_K = G_K + (P_K G_{K-1}) + (P_K P_{K-1} G_{K-2}) + \ldots .$$

$$\ldots . + (P_K P_{K-1} P_{K-2} \ldots . P_1 G_0) \tag{28}$$

where $G_0 = C_{in}$ (initial carry in)

The carry at each stage of the parallel adder can then be produced as a combination of propagate and generate terms. The carry need not ripple through each stage of adder.

The sum block for each sum bit must add an input carry, augend, and addend. The sum equation is:

$$S_K = A_K \oplus B_K \oplus C_{K-1} \tag{29}$$

which is equivalent to the sum equation of the FA.

LOOKAHEAD ADDITION WITH THE MC10180

It is observed that equations (25) and (26) for G_K and P_K are equivalent to equations (1) and (2) for output carry and sum for the HA. It can also be observed that equations (3) and (4) for the FA reduce to equations (1) and (2) for the HA if input carry is held to the zero state. Therefore, a full adder can be used to produce G_K and P_K for the associated addend and augend bits.

A fast carry adder for lookahead addition can then be made utilizing the MC10180. One half of the dual adder is used as a summing block, and one half of the device is used to produce carry generate and carry propagate. Figure 9 illustrates this technique.

A 4-bit parallel lookahead carry adder may be connected as shown in Figure 10 using the MC10180 and the MC10179. The MC10179 generates carry every second bit, therefore carry is rippled between every other adder.

For longer adders the basic 4-bit configuration of Figure 10 would be repeated. Figure 11 shows simplified block diagrams of 8-bit and 16-bit configurations.

Carry lookahead may be extended to higher levels by using the group generate and group propagate outputs of the MC10179 to produce carry at a secondary level. This

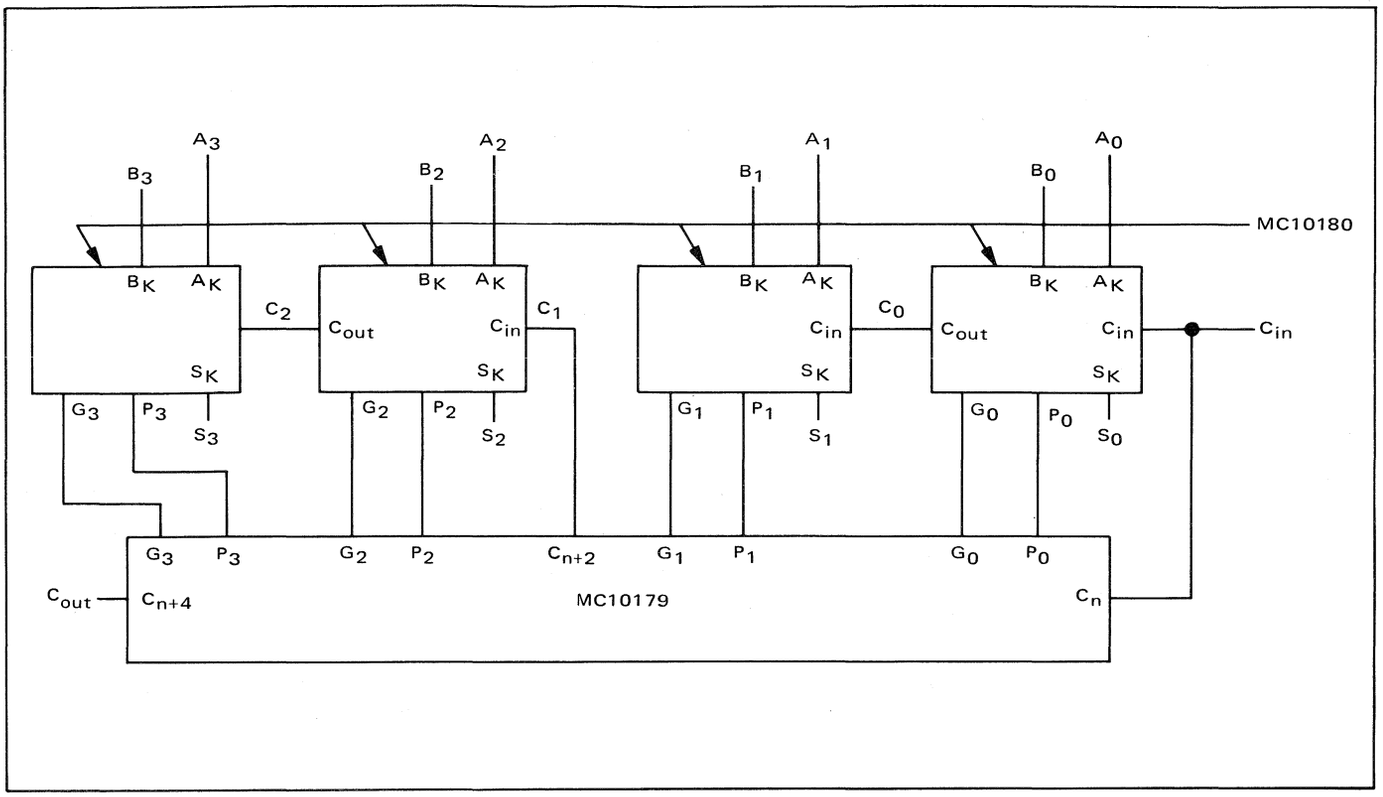


FIGURE 10 – A 4-bit parallel lookahead carry adder using the MC10179 and MC10180's (connected as in Figure 9).

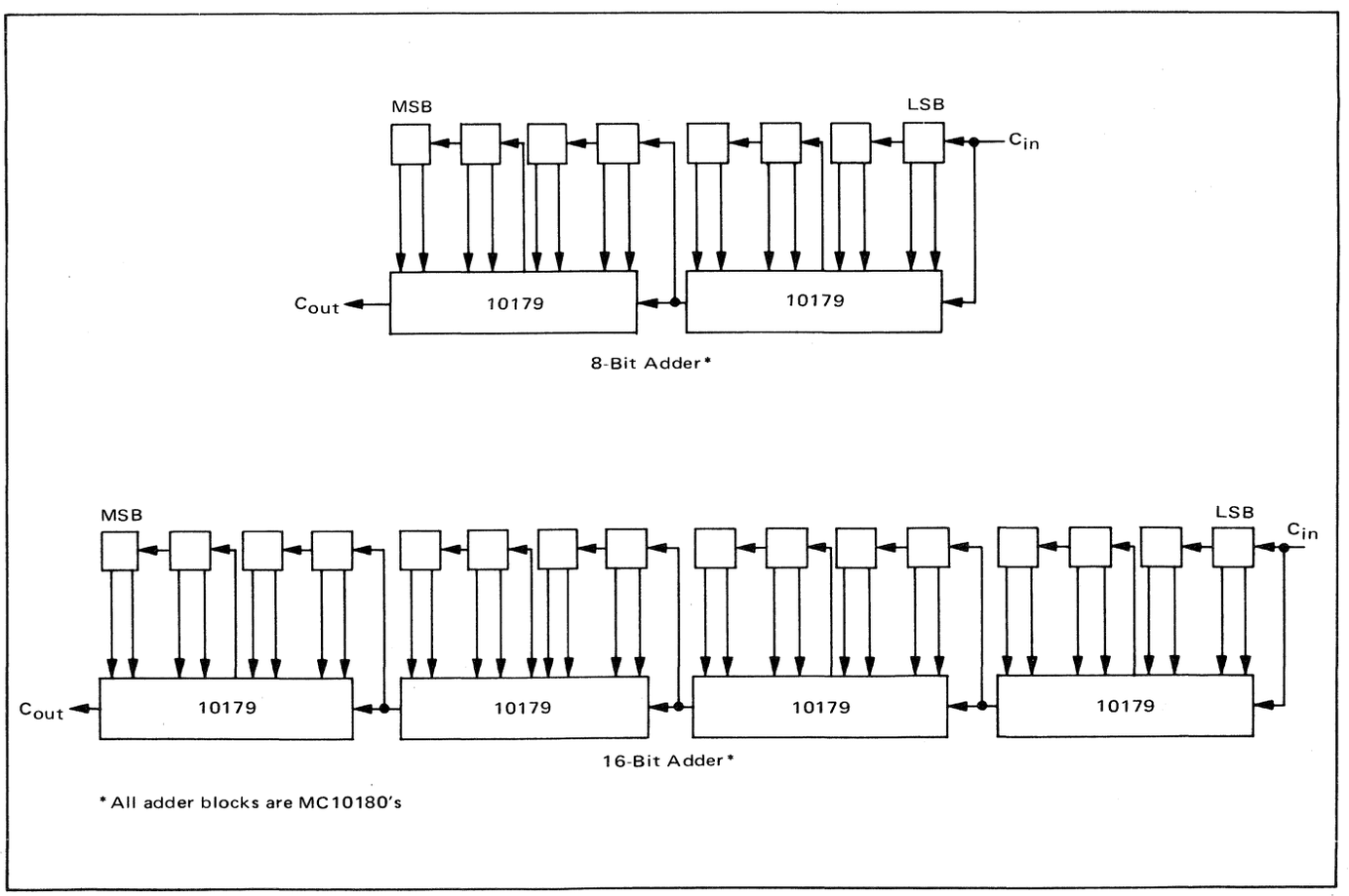


FIGURE 11 – Block diagrams of 8-bit and 16-bit lookahead carry adders using the MC10180 and MC10179

technique gains a time advantage only with word lengths above 16 bits.

The adders shown using the MC10180 are not as fast or as efficient in package count as adders using the MC10181. The advantage of using the MC10180 is that carry is available for each adder bit. Also, the MC10180 is in the 16-pin dual-in-line package for PC board layouts that cannot use the 24-pin package required by the MC10181.

LOOKAHEAD ADDITION WITH THE MC10181

The MC10181 uses lookahead carry internal to the device to do arithmetic operations. In the arithmetic mode of operation for addition of A and B (S0 = L, S1 = H, S2 = H, S3 = L), equations (13) and (14) become:

$$G_i = A_i \cdot B_i \qquad (30)$$

$$P_i = A_i + B_i \qquad (31)$$

Equation (30) is the same as equation (25) for G_K, although equation (31) differs from equation (26) for P_K. It can be shown that equation (31) contains the set of terms of equation (26) and therefore produces carry propagate (although function A + B contains a redundant term). The propagate and generate functions are then used to sum operands A and B and input carry. The internal carry is formed using lookahead logic as shown by equation (17).

The MC10181 with internal lookahead adders is in itself a very fast arithmetic unit. When the MC10181 is incorporated with the MC10179 even faster performance is possible for long words. Use of the MC10179 is advantageous for word lengths longer than 16 bits.

The MC10179 in this application performs secondary level lookahead as the MC10181 performs the first level internally. The group propagate and group generate outputs are utilized by the MC10179 to produce the carry out. Block diagrams of 16-bit and 32-bit adders with the MC10181 and MC10179 are shown in Figure 12.

COMPARISON OF ADDER PERFORMANCE

Several parallel adder configurations have been shown using the MC10180 and MC10181. Figure 13 gives a table

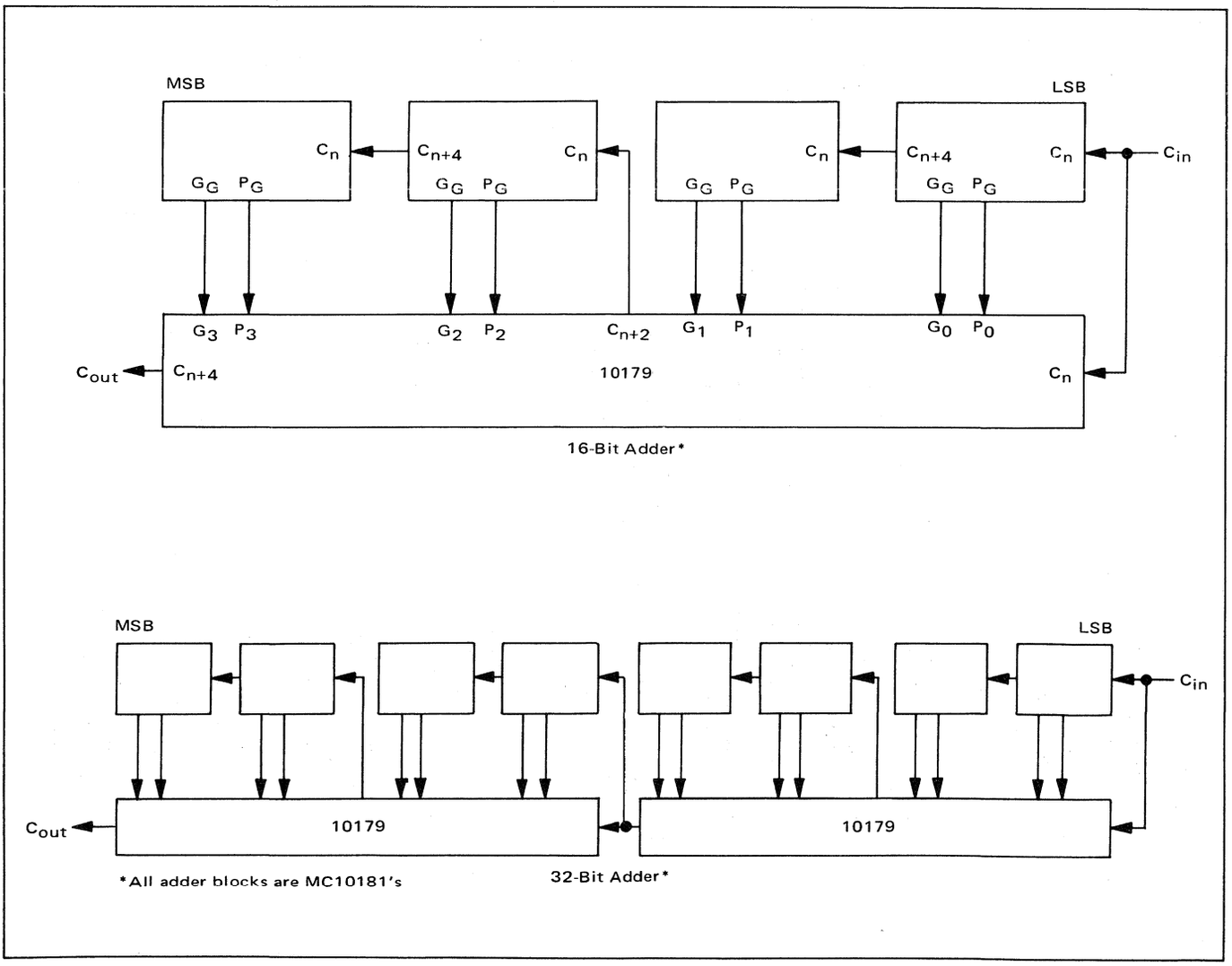


FIGURE 12 – Block diagrams of 16-bit and 32-bit lookahead carry adders using the MC10181 and MC10179

No. of Bits	Configuration	Total Add Time (nsec)	Package Count		
			MC10179	MC10180	MC10181
4	Ripple	11		2	
	Ripple	8			1
	Lookahead	12	1	4	
8	Ripple	20		4	
	Ripple	11			2
	Lookahead	15	2	8	
16	Riple	38		8	
	Ripple	17			4
	Lookahead	21	4	16	
	Lookahead	16	1		4
24	Ripple	55		12	
	Ripple	23			6
	Lookahead	22	7	24	
	Lookahead	16	1		6
32	Ripple	73		16	
	Ripple	30			8
	Lookahead	25	10	32	
	Lookahead	19	2		8
64	Ripple	132		32	
	Ripple	54			16
	Lookahead	31	20	64	
	Lookahead	25	4		16

FIGURE 13 – Typical add times for various parallel adders using the MC10180, MC10181, and MC10179.

of typical addition times for these techniques for various bit lengths. The trade-offs of different types of parallel adders will determine which adder is best for a particular application.

The adders using the MC10180 have the advantage of access to each individual carry bit. The disadvantages of the MC10180 adders are increased package count and longer add times.

The MC10181 has the highest performance in any parallel technique (both ripple and lookahead) at minimum package count. This is due to the internal lookahead carry and 4-bit capability of the device.

SIGNED 2'S COMPLEMENT ARITHMETIC

The binary adders described can be used also for signed 2's complement arithmetic including addition and subtraction. Subtraction is performed by the addition of the 2's complement of the subtrahend to the minuend (for A-B, A = minuend and B = subtrahend). The 2's complement of a number is found by inverting the number bits and adding one to the number, that is, the 2's complement of B = $\overline{B}$ plus 1.

The MC10180 and MC10181 can also perform parallel subtraction. In either case, the subtrahend (normally word B) is inverted by use of the control lines to the device. The C_{in} to the adder is then forced to a "1" state, and subtraction is performed by addition of the 2's complement of the number.

The MC10180 also has the ability to invert either operand A or B. In this manner, various operations as A-B, B-A, or -A-B may be performed with the basic adder configuration.

The MC10181 is more versatile in the arithmetic configuration than just for addition or subtraction. Many variations on A + B or A-B can be performed with the 16 arithmetic function capability of the device. Such functions as A minus 1, A plus 0, and A times 2 (A plus A) are useful for implementing various computer instructions. Also, as mentioned earlier arithmetic functions containing logical expressions (F = A plus (A + B)) may prove useful in particular applications.

CONCLUSION

The MC10180, MC10181, and MC10179 are versatile building blocks that may be used in a variety of arithmetic operations. The devices have different advantages that determine which is most suitable for a particular application.

The MC10181 has 32 function capability, is 4-bits wide, is very fast, and comes in a 24-pin package. The characteristic of the device that may be detrimental to some applications is that the internal carry bits are not available for use.

The MC10180 is a fast dual adder. Its advantages lie in applications where single bit adders are useful. These applications include serial adders and parallel adders where the individual carry bits must be available. The device also is in a 16-pin package. When compared to configurations using the MC10181, the MC10180 adder will be slower and requires a higher package count.

The MC10179 lookahead carry block may be used in configurations with both the MC10180 and MC10181. Addition time on long words can be reduced significantly using this device with the MECL 10,000 adders.

ADDENDUM "A"

$$\overline{S} = C_{in}(\overline{A}B + A\overline{B}) + \overline{C}_{in}(\overline{A}\overline{B} + AB) \quad (5)$$

$$S = \overline{C_{in}(\overline{A}B + A\overline{B}) + \overline{C}_{in}(\overline{A}\overline{B} + AB)}$$

$$S = \left[\overline{C}_{in} + [(A + \overline{B}) \cdot (\overline{A} + B)]\right] \cdot \left[C_{in} + [(A + B) \cdot (\overline{A} + \overline{B})]\right]$$

$$S = \left[\overline{C}_{in} + (\cancel{A\overline{A}} + AB + \overline{B}\overline{A} + \cancel{B\overline{B}})\right] \cdot \left[C_{in} + (\cancel{A\overline{A}} + A\overline{B} + B\overline{A} + \cancel{B\overline{B}})\right]$$

$$S = \left[\overline{C}_{in} + (AB + \overline{B}\overline{A})\right] \cdot \left[C_{in} + (A\overline{B} + \overline{A}B)\right]$$

$$S = \cancel{\overline{C}_{in}}C_{in} + \overline{C}_{in}(A\overline{B} + \overline{A}B) + C_{in}(AB + \overline{B}\overline{A}) + (AB + \overline{B}\overline{A})(A\overline{B} + \overline{A}B)$$

Noting that:

$$(AB + \overline{B}\overline{A})(A\overline{B} + \overline{A}B) = (A \odot B)(A \oplus B) =$$
$$(A \odot B)\overline{(A \odot B)} = 0$$

The equation becomes

$$S = \overline{C}_{in}(A\overline{B} + \overline{A}B) + C_{in}(AB + \overline{B}\overline{A}) \quad (5)$$

ADDENDUM "B"

$$\overline{C}_{out} = \overline{C}_{in}\overline{A} + \overline{C}_{in}\overline{B} + \overline{A}\overline{B}$$

$$C_{out} = \overline{\overline{C}_{in}\overline{A} + \overline{C}_{in}\overline{B} + \overline{A}\overline{B}}$$

$$C_{out} = (C_{in} + A)(C_{in} + B)(A + B)$$

$$C_{out} = (C_{in} + C_{in}B + C_{in}A + AB)(A + B)$$

$$C_{out} = C_{in}A + C_{in}AB + \cancel{C_{in}A} + AB + C_{in}B + \cancel{C_{in}B} + \cancel{C_{in}AB} + \cancel{AB}$$

$$C_{out} = C_{in}A + C_{in}AB + C_{in}B + AB$$

$$C_{out} = C_{in}A + C_{in}B + AB \quad (6)$$

REFERENCES

1. "MECL Integrated Circuits Data Book," Motorola Inc., November, 1972.
2. "MECL System Design Handbook," Motorola Inc., 2nd Edition, December, 1972.
3. AN-488, "High-Speed Addition Using Lookahead Carry Techniques," Motorola, 1969.
4. Y. Chu, "Digital Computer Design Fundamentals," New York, McGraw-Hill, 1962.
5. I. Flores, "The Logic of Computer Arithmetic," New Jersey, Prentice-Hall, 1963.

MOTOROLA Semiconductor Products Inc.

Printed in Switzerland