A 1.5V CMOS High-Speed 16-bit÷8-bit Divider Using the Quotient-Select Architecture and True-Single-Phase Bootstrapped Dynamic Circuit Techniques Suitable for Low-Voltage VLSI

C. C. Yeh, J. H. Lou and J. B. Kuo
Rm. 338, Dept. of Electrical Eng., National Taiwan University
Roosevelt Rd., Sec. 4, #1, Taipei, Taiwan 106-17

***Abstract*—This paper reports a 1.5V high-speed 16-bit÷8-bit divider circuit using the quotient-select architecture and true-single-phase bootstrapped dynamic circuit techniques. Based on a $0.8\mu m$ CMOS technology, the speed performance of this 16-bit÷8-bit divider circuit is improved by 45% as compared to the divider using the non-restoring iterative architecture and the domino dynamic logic circuits without the bootstrapped technique.**

I. Introduction

Division is an important function in a CPU arithmetic unit. Enhancing the speed performance of a divider circuit is critical in raising the speed performance of a VLSI CPU [1]-[3]. Fig. 1 shows the block diagram of a 16-bit÷8-bit divider circuit using a non-restoring iterative architecture [1]. A 16-bit dividend and an 8-bit divisor are assumed to be positive and smaller than 1. As in a standard binary division operation, successively right-shifted values of the divisor are subtracted from or added to the the dividend. Also, for next-generation deep-submicron CMOS VLSI technology, low supply voltage is the trend. For sub-$0.1\mu m$ CMOS technology, 1.5V is necessary. At a supply voltage of 1.5V, the speed performance of CMOS dynamic logic circuits such as NORA [4], domino, Zipper is better than that of CMOS static ones as a result of reduced internal parasitic capacitances. However, as the serial fan-in is large, its associated propagation delay may increase drastically, which is especially serious at a low supply voltage. In this paper, by using a 1.5V CMOS dynamic logic circuit with a bootstrapper technique, a 1.5V high-speed CMOS bootstrapped 16-bit÷8-bit divider using the quotient-select architecture is reported. It will be shown that the speed performance of this 16-bit÷8-bit divider circuit is improved by 45% as compared to the divider using the non-restoring iterative architecture and the domino dynamic logic circuits without the bootstrapped technique.

This work is supported under R.O.C. National Science Council Contracts #84-2622-E002-008, 011 & #85-2215-E0002-024.

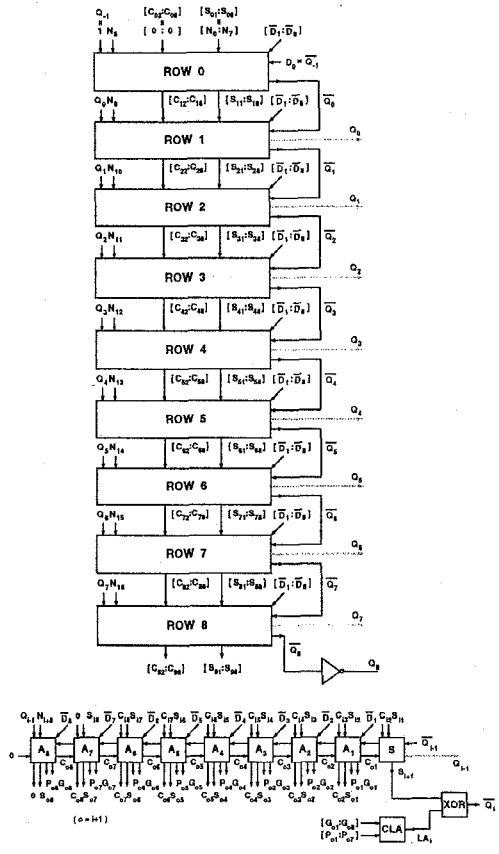

Fig. 1. The architecture of a 16-bit÷8-bit non-restoring iterative divider.

II. Conventional Non-Resoring Iterative Divider

In a conventional 16-bit÷8-bit non-restoring iterative divider, eight 'quo tient rows' are required. In each quotient row, there are A (adder) cells, an S (sign) cell, and a CLA (carry look-ahead) cell. Each A cell is composed of a full adder and a control signal implemented by an EXCLUSIVE-OR gate. The control signal decides whether an add or a subtract to be performed. In the jth A cell in the (i-1)th r ow, using a full adder, its associated sum and carry signals at the output are related to the sum and carry signals of the previous row as:

$$S_{i,j} = (\bar{Q}_{i-2} \oplus \bar{D}_j) \oplus S_{i-1,j+1} \oplus C_{i-1,j+2}, \quad (1)$$

$$C_{i,j} = (\bar{Q}_{i-2} \oplus \bar{D}_j)(S_{i-1,j+1} + C_{i-1,j+2}) + S_{i-1,j+1} C_{i-1,j+2}, \quad (2)$$

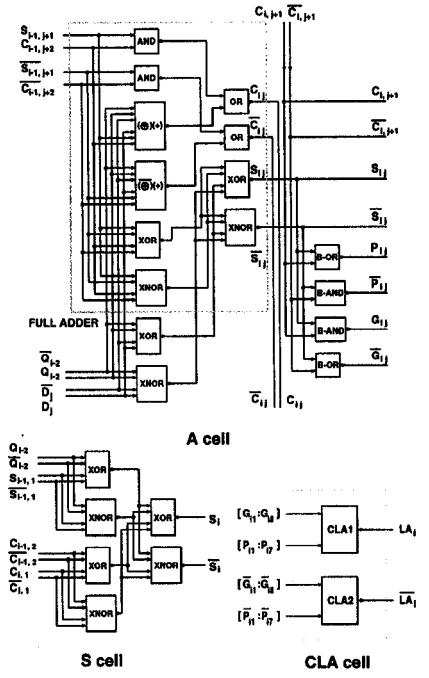

Fig. 2. The functional blocks of the ith quotient row in the 16-bit÷8-bit non-restoring iterative divider.

where $\bar{Q}_{i-2}$ is the negative of the (i-2)th-bit quotient, and $\bar{D}_j$ is the negative of the jth-bit dividend. In addition, propagate and generate signals for the carry look-ahead circuit have been produced:

$$P_{i,j} = [(\bar{Q}_{i-2} \oplus \bar{D}_j) \oplus S_{i-1,j+1} \oplus C_{i-1,j+2}] + C_{i,j+1}, \quad (3)$$

$$G_{i,j} = [(\bar{Q}_{i-2} \oplus \bar{D}_j) \oplus S_{i-1,j+1} \oplus C_{i-1,j+2}] \cdot C_{i,j+1}. \quad (4)$$

The S cell, which contains the sum portion of the A cell, is used to compute the sign for each row:

$$S_i = (\bar{Q}_{i-2} \oplus S_{i-1,1} \oplus C_{i-1,2}) \oplus C_{i,1}. \quad (5)$$

Eqs. (1)-(4) are applicable for the cells not at the top and left boundaries ($2 \le i \le b_q + 1$ and $1 \le j \le b_d - 1$). For the cells at the top and left boundaries ($i = 1$ or $j = b_d$), where b_q is the quotient bit number, and b_d is the divisor bit number, Eqs.(1)-(4) should be modified to include appropriate boundary conditions as shown in Fig. 1. Note that N_i is the ith bit dividend. As shown in Fig. 2, the CLA cell is used to compute the final carry signal according to the following formula:

$$Y_{i,j} = G_{i,j} + P_{i,j} Y_{i,j+1}, \quad j = 1, ..., b_d, \quad (6)$$

$$LA_i = Y_{i,1}. \quad (7)$$

Using an EXCLUSIVE-OR logic gate, the ith quotient bit at the output is low if either output from the S cell or t he CLA cell is 1.

The speed performance of a non-restoring iterative divider is determined by the speed of the propagate and generate signals of the A-cells, the delay time of the LA_i signal in the CLA cell and the speed of producing the quotient bit $\overline{Q_i}$ of the XOR in the quotient row. After the quotient bit of a quotient row ($\overline{Q_i}$) is produced, its value is transferred to the next row. Then, the quotient bit of the next row ($\overline{Q_{i+1}}$) is computed. No quotient bit of the next row can be computed until the quotient bit of the previous row is obtained.

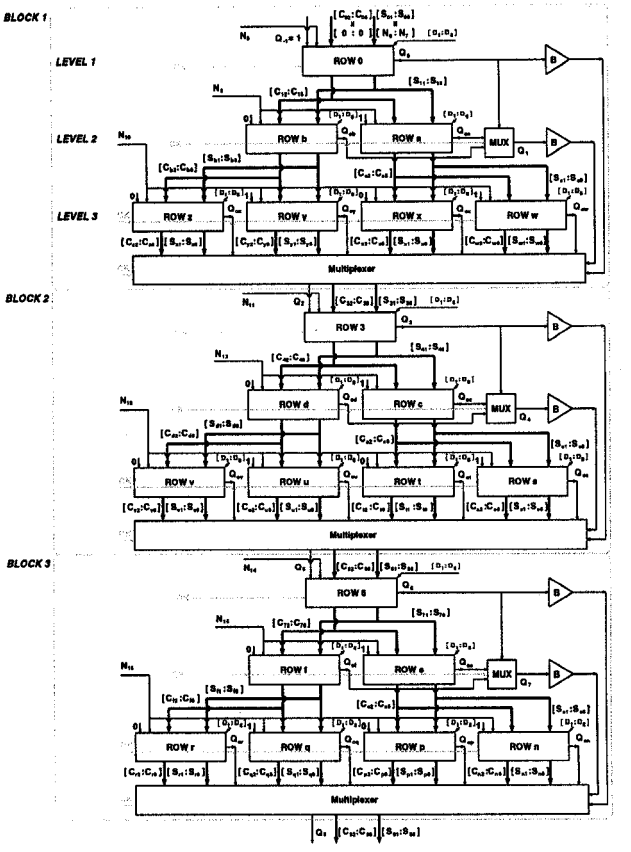

Fig. 3. Block diagram of the 1.5V 16-bit÷8-bit divider using the quotient-select architecture and true-single-phase bootstrapped dynamic circuit techniques.

III. The Parallel-Out Quotient-Select Divider

As shown in Fig. 3, in the 3-bit parallel-out quotient-select architecture, instead of waiting for the quotient bit from the previous row, three quotient blocks have been used to produce the nine output quotient bits almost simultaneously. In each block, three levels of quotient rows have been arranged. For example, under the quotient row of the first level, at the second level there are two quotient rows: a and b. At the third level, there are four quotient rows: w, x, y, and z. The second and the third levels of quotient rows have been arranged to produce three output quotient bits: Q_0, Q_1, and Q_2 simultaneously. In these seven quotient rows in three levels, the input quotient bits of rows 0, a, b, w, x, y, and z have been designated as 1, 1, 0, 1, 0, 1, and 0, respectively.

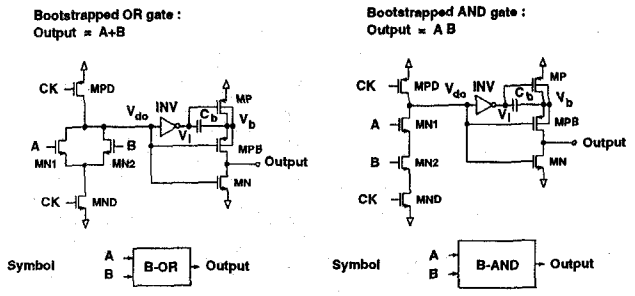

Fig. 4. The 1.5V CMOS bootstrapped dynamic logic circuits including the CMOS bootstrapper circuit.

The seven individual output quotient bits of each row in three levels are Q_0, Q_{oa}, Q_{ob}, Q_{ow}, Q_{ox}, Q_{oy}, and Q_{oz}, respectively. The inputs to the first block are the dividend bits: N_0-N_8. The output quotient bit of row 0— Q_0 may be 0 or 1. The sum and the carry signals produced by row 0 are transferred to rows a and b such that Q_{oa} and Q_{ob} can be computed immediately without waiting for the generation of Q_0. Another dividend bit N_9 is used as the input to both the quotient rows a and b. Then, the output quotient bit— Q_1 is equal to Q_{oa} or Q_{ob} depending on Q_0 by the multiplexer— MUX. If Q_0 is 1, MUX outputs Q_{oa} as Q_1. If Q_1 is 0, MUX outputs Q_{ob} as Q_1. The outputs of the second-level quotient rows—$S_{a1} - S_{a8}$, $C_{a2} - C_{a8}$;$S_{b1} - S_{b8}$, $C_{b2} - C_{b8}$ are used as inputs to the third level. In addition, another dividend bit N_{10} is used as an input to the third level.

The output quotient bit Q_2 of the third level is determined by a similar decision criterion as in the second level. Under the third level of the quotient rows, a multiplexer is used to select the sum and the carry of the first block— S_{32}-S_{38}, C_{31}-C_{38}. The second block uses the outputs from the first block to generate the output quotient bits: Q_3, Q_4, and Q_5. The third block generates the output quotient bits: Q_6, Q_7, and Q_8.

The speed of the 16-bit÷8-bit divider with the quotient-select architecture is determined by the propagation delay of each of the three blocks as shown in Fig.3. The propagation delay of producing the output quotient bits Q_0, Q_1, and Q_2 of the first block is mainly determined by the propagati on delay of the sum and the carry signals ($S's$, $C's$) associated with each quoti ent row in all three levels in the first block. Although there are three levels in the first block, the speed of producing Q_1 and Q_2 is not substantially slower than that of producing Q_0 since the critical component of the propagation delay in producing the three output quotient bits— Q_0, Q_1 and Q_2 is on the adder circuit in the A cell. Therefore, the speed of generating Q_0, Q_1 and Q_2 is about identical. Similar situations exist for Q_3, Q_4 and Q_5 for the second block and Q_6, Q_7 and Q_8 for the third block.

As a result, the speed of the 16-bit÷8-bit divider with the quotient-select architecture is about three times faster

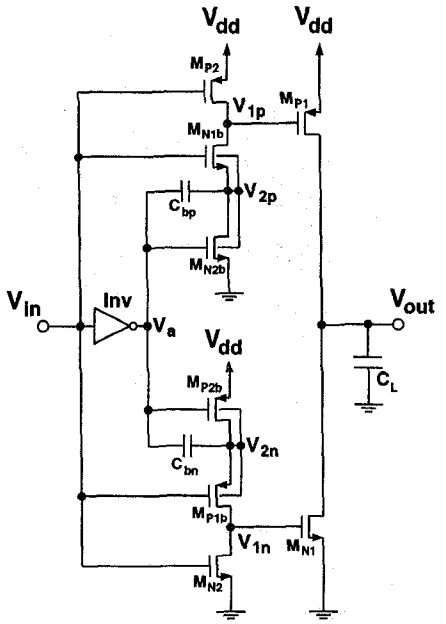

Fig. 5. The 1.5V CMOS buffer circuit using the bootstrapper technique.

as compared to that with the conventional non-restoring iterative architecture. Fig. 4 shows the 1.5V CMOS bootstrapped dynamic logic circuits including the CMOS bootstrapper circuit. As CK is low, it is the precharge period of bootstrapper circuits. During the precharge period, the internal node (V_{do}) is precharged to V_{dd}, and the output voltage V_{out} is predischarged to ground via MN. The bootstrap capacitor (C_b) is charged to V_{dd}— the left side is grounded and the right side is at $V_b = V_{dd}$. During the precharge period, the right side of the bootstrap capacitor is separated from the output since MPB is off. As CK turns high, it's the logic evaluation period. During the logic evaluation period, MPD, MP, M N turn off. During the logic evaluation period, the internal node voltage V_{do} is determined by inputs A and B. If both A and B are high, V_{do} is pulled low and V_I is high. Owing to the charge in the bootstrap capacitor, V_b will be bootstrapped to over V_{dd}— the internal voltage overshoot. In addition, as MPB turns on, V_{out} is pulled high to over V_{dd}. In the CLA cell, owing to the 1.5V bootstrapped CMOS dynamic logic circuit, the signal swing of the input signals—$P's$ and $G's$ exceeds 2V. As a result, the switching speed of the CLA cell is enhanced.

Fig. 5 shows the 1.5V CMOS buffer (B) circuit using the bootstrapper technique[5]. During the pull-up transient, the operation of the full-swing bootstrapped CMOS buffer circuit is divided into two periods regarding the bootstrap capacitor C_{bp}: (1) the charge build-up period and (2) the bootstrap period. Prior to the pull-up transient, the input is at 0V and at the output of the inverter V_a is at 1.5V. Therefore, M_{N1b} and M_{N2} are off ; M_{N2b} is on. At the output of the buffer, V_{out} is at 0V. On the other hand, M_{N2b} and C_{bp} of the bootstrap segment are separated from the M_{P2} and M_{P1} of the fundamental seg-

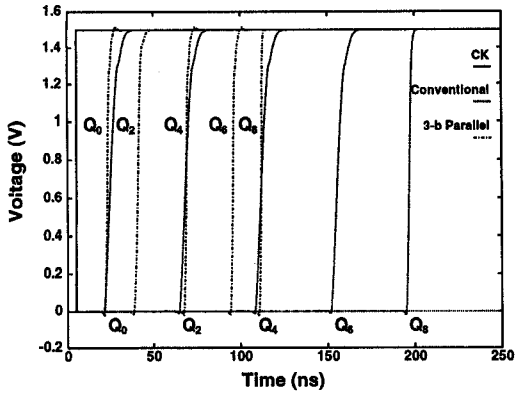

Fig. 6. Transient waveform of the 1.5V 16-bit÷8-bit divider using the quotient-select architecture and the true-single-phase bootstrapped dynamic circuit techniques.

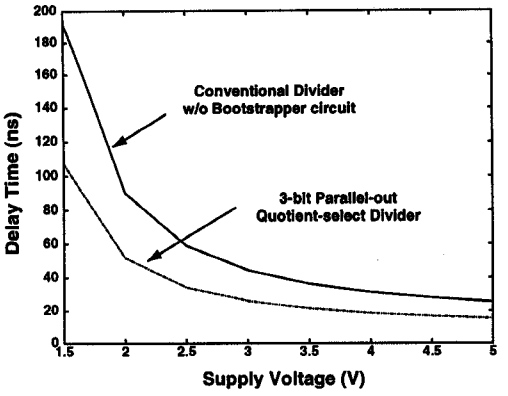

Fig. 7. Delay time vs supply voltage of the true-single-phase CMOS bootstrapped divider.

ment. As a result, the bootstrap capacitor C_{bp} has charge of $1.5C_{bp}$ Coulomb After the input ramp-up period, the right side of the bootstrap capacitor C_{bp} is disconnected from ground since M_{N2b} is off. Instead, it's connected to the gate of M_{P1} since M_{N1b} is on. Due to the voltage change at the left side of the bootstrap capacitor C_{bp}, the right side of the bootstrap capacitor C_{bp} changes to below 0V— the internal voltage undershoot. As a result, the output voltage can switch at a faster pace since the gate of M_{P1} is driven at below 0V. Pull-down transient has a complementary configuration.

IV. Results and Discussion

Fig. 6 shows the transient waveform of the 1.5V 16-bit÷8-bit divider using the quotient-select architecture and the true-single-phase bootstrapped dynamic circuit techniques. The load at the quotient bit output is 0.1pf. At a supply voltage of 1.5V, the propagation delay of the output quotient bit Q_8 is 107ns for the 16-bit÷8-bit divider using the 3-bit parallel-out quotient-select architecture. Compared with the propagation delay of the divider using the conventional non-restoring iterative architecture—192ns, a speed enhancement of 1.8x has be en reached, which is less than 3x as expected. This is due to the fact that the quotient-select architecture is not fully "parallel-processing". The propagation delays of two consecutive quotient rows of a block are differed by the delay in a full adder. In addition, the extra delay due to the multiplexer and the buffer also contributes to the shrinkage in the speed enhancement.

Fig. 7 shows the delay time vs. supply voltage of the true-single-phase CMOS boo tstrapped divider. As shown in the figure, regardless of the supply voltage, a consistent improvement in the speed for the 16-bit÷8-bit divider using the 3-bit parallel-out quotient-select architecture over the the one using the conv entional non-restoring iterative architecture can be seen.

V. Conclusion

In this paper, a 3-bit parallel-out quotient-select architecture has been studied. In fact, for a large-size divider system such as a 64-bit÷32-bit divider, a 8-bit parallel-out quotient-select architecture can be used to further enhance the speed performanc e. The more bits used in the parallel-out quotient-select structure, the more improvement in speed can be expected. However, a larger die area is also needed.

References

[1] M. Cappa and V. C. Hamacher, "An Augmented Iterative Array for High-Speed Binary Division," *IEEE Trans. Computers*, Vol. C-22, No. 2, pp.172-175, Feb. 1973.

[2] R. Stefanelli, "A Suggestion for a High-Speed Parallel Binary Divider, " *IEEE Trans. Computers*, Vol. C-21, No. 1, pp.42-55, Jan 1972.

[3] A. Vandemeulebroecke, E. vanzieleghem, T. Denayer, and P. G. A. Jespers, "A New Carry-Free Division Algorithm and its Applications to a Single-Chip 1024-b RSA Processor, " *IEEE J. Solid-State Circuits*, Vol. 25, No. 3, pp. 748-756, June 1990.

[4] N. F. Gonzales and H. J. DeMan, "NORA: A Racefree Dynamic CMOS Technique for Pipelined Logic Structures," *IEEE J. Solid-State Circuits*, Vol. 18, pp. 261-266, June 1983.

[5] J. H. Lou and J. B. Kuo, "A 1.5V Full-Swing Bootstrapped CMOS Large Capacitive-Load Driver Circuit Suitable for Low-Voltage CMOS VLSI," *IEEE J. Solid-State Circuits*, Vol. 32, No. 1, pp. 119-121, Jan. 1997.