A Bipolar Population Counter Using Wave Pipelining to Achieve 2.5× Normal Clock Frequency

Derek C. Wong, *Member, IEEE*, Giovanni De Micheli, *Senior Member, IEEE*, Michael J. Flynn, *Fellow, IEEE*, and Robert E. Huston, *Member, IEEE*

***Abstract*—Wave pipelining is a technique for pipelining digital systems that can increase clock frequency in practical circuits without increasing the number of storage elements. In wave pipelining, multiple coherent waves of data are sent through a block of combinational logic by applying new inputs faster than the delay through the logic. The throughput of a 63-b CML population counter was increased from 97 to 250 MHz using wave pipelining. The internal circuit is flowthrough combinational logic. Novel CAD methods have balanced all input-to-output paths to about the same delay. This allows multiple data waves to propagate in sequence when the circuit is clocked faster than its propagation delay.**

***Key Words*—Wave pipelining, maximal-rate pipelining, digital circuits, computer-aided design, rough tuning, fine tuning, computer design, hardware, power optimization.**

I. Introduction

WAVE pipelining is a design method that can boost the pipeline rate of a system without using additional registers or latches.

In ordinary pipelined systems, there is one "wave" of data between register stages. When a new set of values is clocked into one set of registers, the values are allowed to propagate to the next set of registers before the first set is clocked again.

In contrast, wave pipelining is the use of multiple coherent "waves" of data between storage elements (see Fig. 1). This is achieved by clocking the system faster than the propagation delay between registers. In this method, the data values at the first set of registers are changed before the old data values have propagated to the next set of registers. If all the propagation paths from the combinational circuit's inputs to outputs have approximately the same delay, each wave propagates uniformly to the outputs without interfering with adjacent waves. The capacitances in the combinational logic circuit are used to store values for pipelining.

Manuscript received October 2, 1991; revised January 7, 1992. This work was supported by the Center for Integrated Systems at Stanford University and by the National Science Foundation under Contract MIP88-22961 using equipment provided by NASA under Contract NAGW 419. G. De Micheli was supported by NSF, DEC, and AT&T under a Presidential Young Investigator award.

D. C. Wong, G. De Micheli, and M. J. Flynn are with the Computer Systems Laboratory, Departments of Electrical Engineering and Computer Science, Stanford University, Stanford, CA 94305-4055.

R. E. Huston is with LTX Corporation, Trillium Division, San Jose, CA 95134.

IEEE Log Number 9107366.

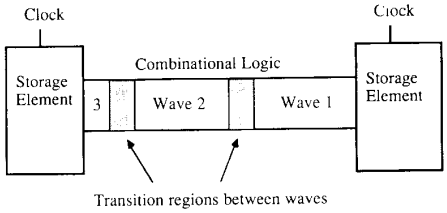


Fig. 1. In wave pipelining, multiple coherent waves of data are sent through combinational logic acting as a pipeline.

In processor designs, wave pipelining can be applied effectively to increase the pipeline rate of arithmetic units such as floating-point multipliers.

Wave pipelining can provide the following possible advantages:

- lower power, area, and delay by using fewer stages of storage elements;
- very high rates of pipelining without the added delay of storage elements dominating the input-to-output delay;
- wide applicability to all pipelined digital systems.

The following disadvantages exist:

- specialized design algorithms are needed to equalize the length of all paths;
- system level design can be more difficult. A wave-pipelined chip must be run at the proper clock frequency; it will not work at higher or lower rates. When a system using multiple wave-pipelined chips is assembled, the chips must have matched speeds;
- delay buffers may be required to lengthen some short paths. This increases the area of some circuits.

In this paper, we describe a bipolar LSI chip that achieves 2.5 times the normal clock frequency without the use of additional storage elements. This is the first published bipolar LSI chip that uses wave pipelining and the first one designed using CAD algorithms developed specifically for designing wave-pipelined circuits.

Our research goal has been to develop the necessary analytical tools and design techniques to actually build a wave-pipelined chip in VLSI, identifying and solving the necessary practical problems en route. The results of this research are theory as well as practical methods for designing circuits using wave pipelining.

We have focused on the following areas:

- analyzing technologies and clocking limits;
- designing the algorithms and developing the necessary CAD tools to automatically balance the delays in combinational logic circuits;
- designing, building, and testing a sample chip design.

This paper principally describes the third part of our research effort. For more details on our work, please refer to [9]–[11].

Research in wave pipelining combines a number of distinct disciplines including clocking theory, CAD, technology analysis and selection, and circuit design. Other work in this field has been performed by Anderson *et al.*, who first described the concept as used in the floating-point unit of the IBM 360/91 [1]. Cotten provided the first theoretical analysis of wave pipelining [2]. Lin and Xia designed and implemented an experimental computer using wave pipelining in its arithmetic units [8]. Wave pipelining was also used in a CMOS time-division multiplexer circuit [12]. Ekroot developed a theory of wave pipelining [4]. Assuming gates and modules with fixed delays, he developed linear programs to determine where to insert delay elements to balance the circuit. Also, he compared the minimum clock period using wave pipelining and ordinary pipelining. Two groups have recently been developing the methods of applying wave pipelining to CMOS circuits [5], [7]. Finally, another research group has been developing placement-and-routing algorithms for laying out wave-pipelined circuits [6].

II. The Wave-Pipelining Principle

The maximum pipeline rate is limited by technological parameters. Clocking the circuit at a frequency above the limit would mix the waves of data together.

The minimum clock period for a wave-pipelined circuit is bounded by the following constraint [10], [11][1]:

$$t_{CP} > \Delta t_P + 2 \cdot \Delta C + t_S + t_H + t_{RF} \qquad (1)$$

where

- t_{CP} clock period,
- t_P worst-case propagation time of the longest path in the combinational logic,
- Δt_P maximum difference between the longest and shortest path delays over worst-case design, process, and environment,
- ΔC worst-case uncontrolled clock skew,
- t_S setup time for edge-triggered registers (for transparent latches, t_S should instead equal the transparent period),
- t_H hold time for either edge-triggered registers or transparent latches,
- t_{RF} worst-case rise or fall time (10% to 90% voltage swing) at the last logic stage. This term prevents waves from colliding at a node with substantial rise/fall delay.

[1]A second constraint described in [10] and [11] ensures that two waves cannot collide at any node within the logic itself. Typically, the second constraint bounds the clock period less stringently than the above constraint.

In contrast, the normal clock period t_{CPN} without using wave pipelining would be bounded by

$$t_{CPN} > t_P + t_{SU} \qquad (2)$$

(plus possibly clock skew depending on the design technique) where t_{SU} is the setup time of the storage element.

By reducing Δt_P to a small fraction of t_P, the clock period t_{CP} using wave pipelining can be made much smaller than t_{CPN}. The path variation Δt_P arises from several sources: path differences due to design, process, and temperature-induced variations within one chip, and data-dependent delay variations. Process and temperature-induced variations are unavoidable, but their effects are limited within one chip. Data-dependent delays can be limited by selecting the proper technology. Using our algorithms, the worst-case Δt_P can be reduced to 15–30% of t_P in suitable technologies.

Wave pipelining can be applied to any technology, but the benefits are greater in some technologies than others. The ideal technology would have the same gate delays independent of the data patterns applied to the inputs. Data-dependent delay variations increase Δt_P and hence reduce the degree of wave pipelining that is possible. Individual gates in some common logic families have the following approximate data-dependent variations [7], [11]:

- ECL, CML, and super-buffered ECL without stacked (cascode) structures—about 5–10%;
- ECL logic using stacked structures—about 20%;
- static CMOS using two-input NAND only—about 25%.

In addition, crosstalk between signal lines, especially due to coupling capacitances, can affect delays in any technology [11]. It is very possible that the worst-case variation does not sum over a number of levels of logic because no input patterns can cause all gates along a path to have the shortest possible or longest possible delays. As found by Klass and Mulder [7] in the simulations of a 4-b adder, the delay variations tend to average out over a number of gate levels due to the logic function of the circuit.

III. Algorithms for Designing Wave-Pipelined Circuits

To make all the paths have approximately the same delay, special CAD algorithms and tools have been developed. These CAD tools take an ordinary design containing imbalanced delays and apply the following two algorithms described elsewhere ([9]–[11]).

1) Rough tuning is a graph algorithm that inserts a minimal number of active delay elements (noninverting buffers) into the circuit to pad out short paths that cannot be balanced by simply adjusting gate delays. In many cir-

cuits, balancing all path delays requires inserting some delay elements. Consider a circuit example of two gates connected in series where the output of each gate is also a global output of the circuit. In this case, the delay from the inputs to the first gate's output is always faster than the delay to the second gate's output. To balance this circuit, rough tuning would add a delay buffer between the output of the first gate and the first global output.

2) Fine tuning balances path delays by adjusting the speeds of gates in the circuit. This can be done by changing parameters such as resistor values and transistor sizes in gates. The fine-tuning algorithm solves a linear program to set all the gate parameters to minimize the imbalance in input-to-output path delays. In addition, the linear program also minimizes the power consumption of the circuit while keeping all paths shorter or equal to a user-specified delay. The fine-tuning algorithm can thus be used to optimize power versus delay for both ordinary and wave-pipelined circuits.

The methods are designed to lengthen short paths to approximately equal the length of the critical path(s). The critical path(s) are never lengthened using these methods.

IV. Test Chip Design

In order to fully test the wave-pipelining concept, we have designed a demonstration chip. This chip has been designed to be a demonstration of wave pipelining rather than a commercial circuit. The logic circuit performs a population counter function: the circuit takes 63 parallel inputs and outputs the number of ONES in that vector as a binary number. This function is similar to a section of a high-speed combinational multiplier. The number of levels of logic is only slightly shorter than that required for a 64 × 64 multiply.

As shown in Fig. 2, the design is split into two major sections. The first is a carry-save adder tree that takes 63 input lines and converts them into two 6-b numbers. The adder tree is implemented using 3–2 counters. The second section is a 6-b carry-propagate adder that adds the two 6-b numbers and produces the population count as a 6-b number. The adder is implemented using a basic carry lookahead scheme.

The circuit is a combinational logic circuit with 21 levels of logic and a nominal longest path delay of 8.5 ns for the core logic plus 1 ns for the output pin drivers. After tuning, the path length difference due to design is about 1.1 ns excluding the effects of differences in rising versus falling delays and data-dependent delays. The total delay variation Δt_P includes these effects plus process and temperature variation within the chip.

The circuit has been designed in a commercial BiCMOS process from Signetics called Qubic 1 [3]. One local metal layer and two global metal layers are provided. The minimum-sized n-p-n transistors have the following parameters: f_T = 13 GHz, A_{EM} = 1 × 2 μm, C_{BE} = 6 fF, C_{BC} = 6 fF, and C_{CS} = 35 fF.

The circuit is implemented in single-level CML using a standard cell technique. As exemplified by Fig. 3, all the logic cells are OR/NOR gates designed using a single level of current switches. The gates are single-ended rather than differential and use a voltage swing of 500 mV. The complete chip has 800 logic gates plus input and output buffers and voltage reference generators.

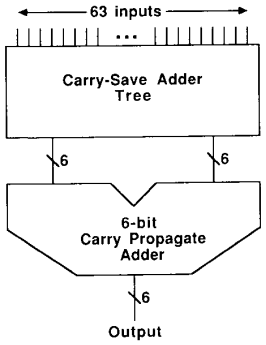


Fig. 2. 63-b population counter chip.

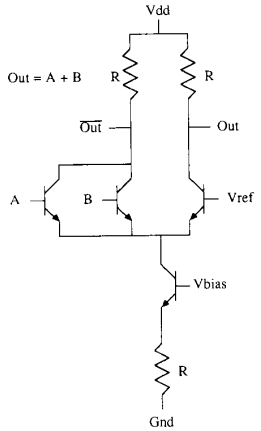


Fig. 3. A two-input OR/NOR gate in CML.

The individual logic cells are similar to the conceptual layout shown in Fig. 4. The GND and V_{DD} lines run horizontally in metal 1, and the logic signal lines, which are not shown in the diagram, run vertically in metal 2. Minimum-sized n-p-n transistors are used throughout. Note that the cell uses groups of four resistors of value 2.5, 1.5, 0.5, and 0.5 kΩ. Any combination of these can be wired in series to produce a resistor of any value between 0.5 and 5 kΩ in 0.5-kΩ increments. This allows each gate's current to be tuned without affecting the placement and routing of the overall circuit.

The supply voltage is 5 V, but could be reduced to 3 V since no stacked current switches are used. The nominal current consumption of the logic circuit, excluding I/O's and voltage generators, is 207 mA.

A. Design Flow

We used the following design flow:

1) First, a schematic netlist and the gate layout cells were designed.

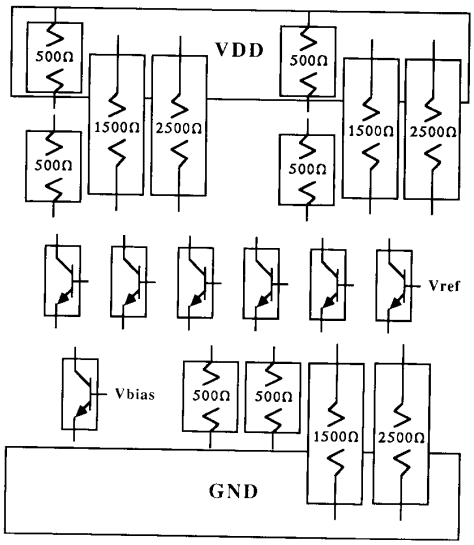


Fig. 4. Conceptual cell layout.

2) The circuit was rough-tuned.
3) A layout of the core logic circuit was made from the circuit schematics using a commercial automatic placement and routing tool.
4) Parasitic capacitances on each net were extracted.
5) A final fine-tuning pass was performed. Since all the power levels of each gate occupy the same dimensions, the power levels of each gate can be changed without requiring any changes in the placement and routing.
6) Power buses, bias voltage generators, input/output buffers, and pads were added using manual layout techniques.

The commercial place-and-route tool is designed to minimize total area. The lengths of nets having the same fanout can vary substantially, causing a corresponding variation in capacitance. The final fine-tuning pass compensates for this variation, so that the remaining imbalance in delays is kept small.

The final chip has 54 pads including the voltage reference and power/ground pins. The number of input pads was reduced from 63 to 16 by wiring a few logic inputs to each pin. This reduction simplifies packaging and testing the prototype chip, while still allowing 2^{16} input patterns to be applied.

A photomicrograph of the chip is shown in Fig. 5. The core logic is 2.5 × 3.8 mm, and the chip is about 4 × 6 mm total. The inputs are on the left, and the outputs are on the right. The power/ground and reference inputs are along the top and bottom sides. The core array is composed of about 25 rows of cells separated by routing channels.

At present, the circuit has been fully designed, simulated, fabricated, and tested. As discussed in the next section, the tests of the manufactured chips indicate that they are functional and support wave pipelining.

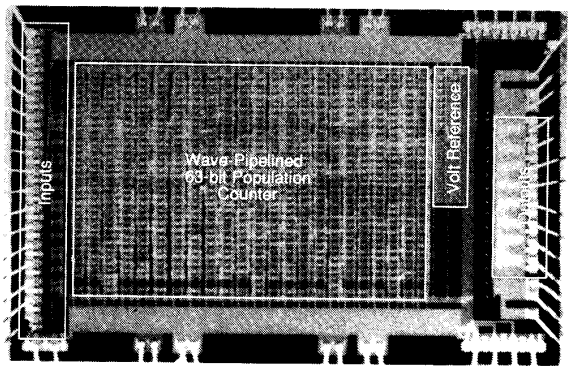


Fig. 5. Photomicrograph of LSI chip that demonstrates wave pipelining.

Following the design and fabrication of this circuit, we also studied an alternative design of the demonstration circuit including the use of stacked structures and emitter followers. Stacked structures are equivalent to two levels of OR/NOR gating where the second level has a fan-in limit of about 3 (using a 5-V power supply). For logic functions such as 3–2 counters which fit well into a stacked structure, the number of gating levels is cut in half, thus reducing the latency. For a population counter, the use of stacked structures does result in substantially lower input-to-output latency. Wave pipelining can still be applied, but the maximum number of waves is reduced due to the stacked structures. But since the circuit latency is lower, we found that the wave-pipelining frequency in this alternative design would actually be equal to or higher than the frequency of the actual demonstration chip despite using fewer waves (see [11] for more details).

V. Testing the Demonstration Chip

The wave-pipelining demonstration chips have been tested using a Trillium Delta-Master. Some key specifications of the Trillium Delta-Master are as follows:

- the inter-pin skew is guaranteed ≤ 0.5 ns for all events worst case; in practice, a principal engineer at Trillium estimates that the skew is typically closer to 100 ps;
- the tester's time step resolution is 20 ps, but subject to the above skew.

Seventy-two chips were packaged from two wafers from the same fabrication process run. Twenty-six chips passed a 20 000 vector functional test at 40 MHz.

A. Wave-Pipelining Speed Tests

To test at faster than 160 MHz, a special mode of the Trillium tester was used. The test used a special programming mode called return-to-complement in which Trillium pin drivers automatically switch to the complement of the programmed value halfway in a machine cycle. By using this, the effective maximum vector rate is doubled from 160 to 320 MHz. Each original input vector is valid for one half of a cycle followed by its exact complement

for the second half of the cycle. The total pattern size is the original 20 000 times 2 or 40 000 vectors of data. In a 63-b population counter, inverting all inputs causes all outputs to also invert. Since the new count is 63 minus the old count, the new output is always a bit-wise inversion of the old output.[2] Therefore, this test is strenuous since all outputs switch simultaneously for each inversion.

The 40 000 vector sequence has been applied at various vector rates up to 320 MHz. The maximum wave-pipelining frequencies for the 26 chips are as follows:

- one chip at 235 MHz (clock period = 4.250 ns);
- four chips at 242 MHz (clock period = 4.125 ns);
- nineteen chips at 250 MHz (clock period = 4.000 ns);
- two chips at 258 MHz (clock period = 3.875 ns).

This is 2.4 to 2.65 times faster than the ordinary pipelining frequency of 97 MHz supported by the original, unbalanced circuit. By using wave pipelining, the clock frequency has been more than doubled without using any additional registers.

Fig. 6 is an oscilloscope trace showing one input and one output pin during the test applied at 250 MHz. Since the propagation delay is about 9.0 ns for this part, more than two waves of data are stored within the combinational logic.

B. Delay Test

Fig. 7 is a histogram for the 26 chips showing the maximum propagation delay from inputs to outputs. This diagram shows that the maximum propagation delay for chips from these two wafers ranges between 9.0 and 9.9 ns.

Fig. 8 is a histogram for the 26 chips showing the minimum time that each wave of data is valid at the outputs for waves applied at 200 MHz. Every wave of data is valid for at least this period of time before the next wave begins to arrive. This pulse width measurement shows the actual valid period after accounting for the worst-case path delay difference Δt_P, the tester's inter-pin skews, the rise/fall time t_{RF}, and some tester parameters analogous to t_S and t_H.

C. Wave-Pipelining Reliability Tests

To test the reliability under voltage variations, Shmoo plots have been generated. Shmoo plots are two- or three-dimensional plots where parameters such as voltage or clock frequency are varied along the axis dimensions. A solid square or cube at a particular set of coordinates indicates that the device has passed the test under the operating conditions indicated by the coordinates (e.g., clock frequency = 240 MHz, V_{EE} = −5.1 V). Usually, there is a solid pass region in the plot surrounded by a fail region representing extreme operating conditions.

[2]This bit-wise inversion property holds for population counters with $2^N - 1$ inputs. For example, it works for 63 inputs but not for 62 or 64 inputs.

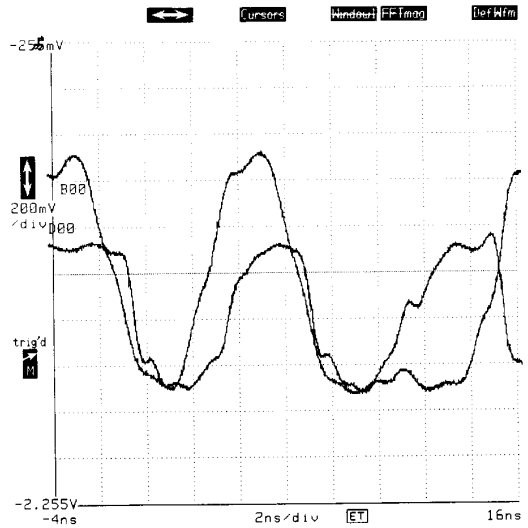


Fig. 6. Oscilloscope trace of wave pipelining at 250 MHz. B00 is the input, and D00 is the output.

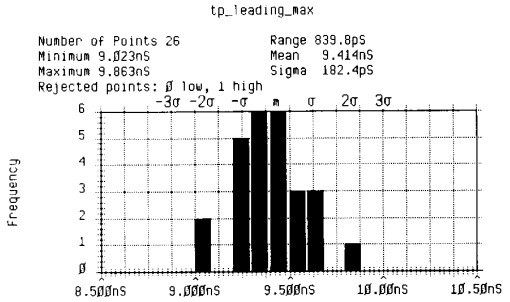


Fig. 7. Histogram of maximum propagation delays from inputs to outputs. Twenty-six chips were tested. All times are in nanoseconds.

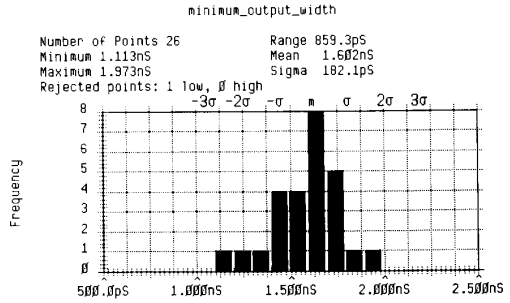


Fig. 8. Histogram of the width of the valid period for waves of data. Twenty-six chips were tested. All times are in nanoseconds.

In the test fixture, V_{DD} is connected to 0 V and GND is connected to a negative voltage supply called V_{EE}. In the plot shown in Fig. 9, the supply voltage V_{EE} (driven by the power supply called *PS2* in the plot) is varied between −4.0 and −6.0 V, and the clock frequency is varied between 80 and 320 MHz. Since the bias voltage generators are referenced to an externally supplied analog voltage, this supplied voltage is made to track variations in V_{EE} so

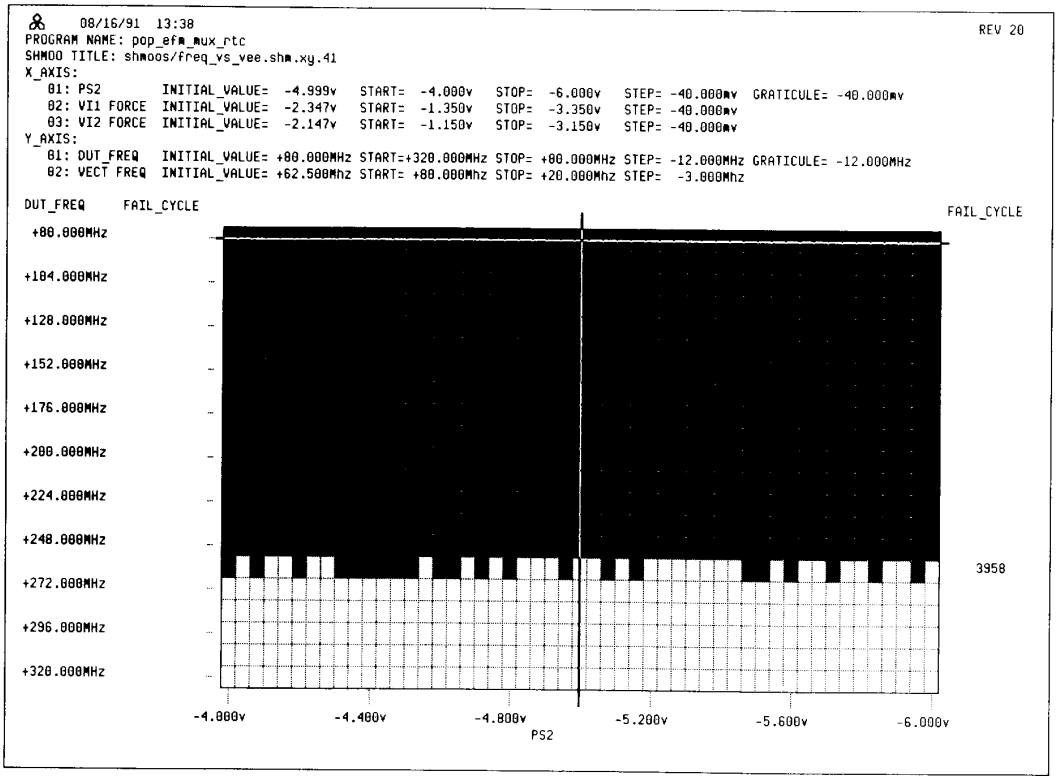


Fig. 9. Shmoo plot of the operating frequency DUT_FREQ versus supply voltage V_{EE} (called *PS2* in the plot). The internal voltage swing is kept constant.

that the internal ECL voltage swing remains at the designed 0.5 V. The plot shows that the part passes the test for the entire V_{EE} range between -4.0 and -6.0 V up to the maximum operating frequency of the part.

We have performed some limited tests of temperature variations on a few of the functional chips. High temperatures increase the maximum propagation delay by about 0.5 ns and increase the worst-case difference between longest and shortest path delays by 0.25 ns. Theoretically, this should cause the minimum clock period to increase by about 0.25 ns.

D. Parametric Measurements

We have detected one aberration in voltage levels that does not affect the ability of the chip to demonstrate wave pipelining. In this case, the chip outputs are not swinging as high as they were designed and simulated to do. The regular 100K ECL swing is about -0.9 to -1.65 V. As shown in Figs. 10 and 11, the output voltage range on our 26 chips is between -1.12 and -1.08 V for logic high, and -1.68 to -1.63 V for logic low using test vectors at 40 MHz. This should not affect our ability to test wave pipelining. The Trillium tester has been configured to threshold the chip's outputs at the middle of the actual output voltage swing.

The oscilloscope trace shown in Fig. 6 confirms these voltage swings are essentially unchanged at 250 MHz.

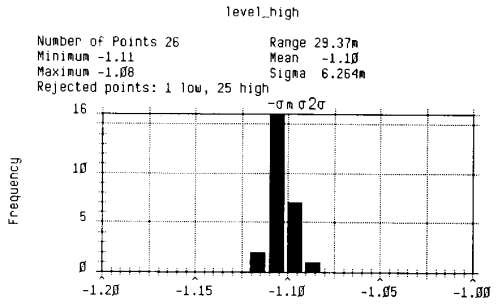


Fig. 10. Histogram of logic high voltage at output pins using 40-MHz test. Twenty-six chips were tested.

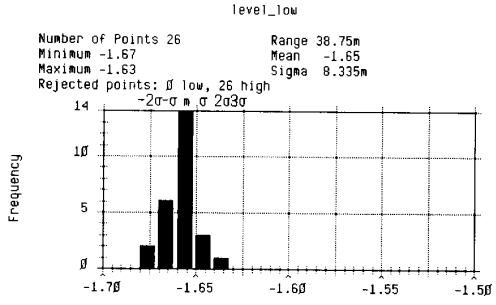


Fig. 11. Histogram of logic low voltage at output pins using 40-MHz test. Twenty-six chips were tested.

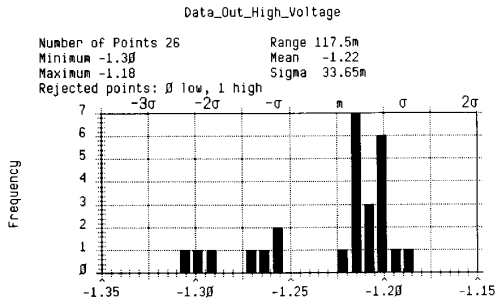


Fig. 12. Histogram of logic high voltage at output pins using 200-MHz test. Twenty-six chips were tested.

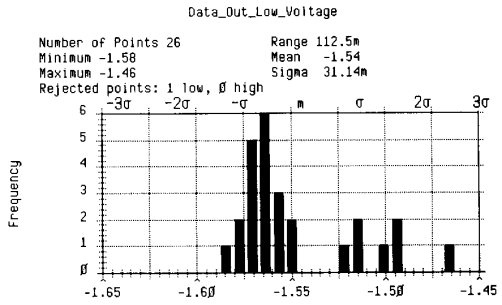


Fig. 13. Histogram of logic low voltage at output pins using 200-MHz test. Twenty-six chips were tested.

However, if the Trillium tester is used to find the worst-case threshold values for the output high and low voltages, the result is a reduced guaranteed voltage swing at 200 MHz compared to 40 MHz. Figs. 12 and 13 are the guaranteed voltage levels at 200 MHz. The Trillium tester finds the lowest logic low and highest logic high thresholds such that all 40 000 patterns pass.

At higher frequencies, crosstalk and other noise effects increase in the test fixture as well as within the chip causing the guaranteed voltage levels to decrease. Oscilloscope traces taken during high-frequency tests show significant noise and reflections on the signal lines of the test fixture which carry the chip's outputs. Also, as a particular part approaches its maximum frequency limit, the guaranteed voltage swing might decrease slightly because the worst-case patterns might not have time to swing fully due to slow worst-case output rise/fall times of up to 2.1 ns. This might explain why some parts have less safety margin in the voltage levels than others at 200 MHz. For wave-pipelined designs that include storage elements on the same chip, slow pin drivers would not affect the wave-pipelining frequency.

E. The Wave-Pipelining Limit in the Demonstration Chip

The minimum clock period is about 4 ns according to the chip tests. The wave-pipelining limit is

$$t_{CP} > \Delta t_P + 2 \cdot \Delta C + t_S + t_H + t_{RF}. \quad (3)$$

This constraint can be used to estimate Δt_P. In the case of the demonstration chip connected to the Trillium tester, the estimated individual parameters are as follows.

- Δt_P is a number which accounts for the difference between longest and shortest paths; for this chip, Δt_P is equal to 1.1 ns (from the design) plus data-dependent delays, fabrication, and temperature variations within one chip, and differences between rising and falling delays.
- ΔC corresponds to the inter-pin skew of the tester, which is worst-case 0.5 ns but could be closer to 0.1 ns in practice.
- t_S and t_H are small because the tester uses comparator circuits rather than storage elements to examine the chip outputs. The exact values analogous to t_S and t_H are not known.
- t_{RF} = 2.1 ns according to oscilloscope measurements. Since it is possible for the chip to pass the wave-pipelining tests with somewhat less than the full output voltage swing, part of t_{RF} might not be included at the fastest clock period.

Although the parameters are not known exactly, the approximate values should be consistent with the observed minimum clock period. Even if t_{RF} were discounted by 50% to 1.05 ns, ΔC were 0.1 ns, and t_S and t_H were zero, Δt_P would have to be less than 2.75 ns.

If on-chip storage elements were used and the parameters were t_{RF} = 0.5 ns, t_S = 0.5 ns, t_H = 0 ns, and $2 \cdot \Delta C$ = 0.5 ns, then the corresponding minimum clock period would be 4.25 ns if Δt_P were 2.75 ns. This is comparable to the minimum clock period of 4.0 ns observed using the Trillium tester. The setup using the Trillium tester has a larger t_{RF}, but a smaller t_S, t_H, and probably ΔC, so that the differences approximately cancel overall.

VI. Summary

Wave pipelining is a practical method for increasing pipeline frequency without using additional storage elements. In our research, we have developed a general CAD methodology for designing wave-pipelined circuits which is applicable to current and future technologies. This methodology has now been successfully applied to design an LSI chip using wave pipelining.

Using the fine- and rough-tuning algorithms, a 63-b population counter has been designed in CML technology using a commercial BiCMOS process. The layout has been performed using an automatic placement-and-routing methodology. Since the tuning software adjusts delays based on extracted capacitance values after layout, a well-balanced circuit can be designed without having to tightly control capacitances during placement and routing. The chip is designed to support wave pipelining at 2 to 3 times the normal clock frequency. Circuit simulations confirm that the input-to-output paths have close to the same delay, thus enabling a high degree of wave pipelining.

Tests prove that the fabricated chips do support wave pipelining. All 26 chips that passed low-speed functional

tests support wave pipelining at about 2.5 times the normal clock frequency. The chips successfully pass test sequences of 40 000 vectors at wave pipelining frequencies. Furthermore, the chips have been shown to support wave pipelining robustly over supply voltage and temperature fluctuations. Tests indicate that Δt_P appears to be less than 2.75 ns compared to the propagation delay t_P of 9.5 ns.

VII. Conclusions

Based on the research in this topic to date, wave pipelining has been demonstrated to be a feasible technique for designing circuits. In our research, a fabricated ECL/CML circuit has been successfully tested to verify wave pipelining. Other researchers report that certain types of CMOS circuits have been designed and in some cases fabricated and tested to demonstrate wave pipelining [5], [7].

After applying the tuning algorithms, the variation in path delay not including process/temperature variations and data-dependent variations is 12% of the delay of the longest path in the demonstration circuit. This circuit has been designed using cell-based, auto-place and route layout methods. Circuits designed with a more regular layout offer better control of the net capacitances and would probably have even smaller path delay variation after applying the CAD techniques.

During rough tuning, those circuits which have many short paths require more delay buffers than circuits which have few short paths. For instance, we have found that minimal padding is required for Wallace tree structures while more substantial padding is required for adder circuits.

Acknowledgment

G. Bewick provided much useful advice on ECL chip design and was the main contributor in studying the alternative design of the demonstration chip including stacked structures. R. Kao provided some data on input-slope dependent delay variations. Philips Research and Signetics in Sunnyvale provided use of their equipment and generous help in designing and fabricating the chip. CAD tools were provided by Mentor Graphics, Inc. P. Baltus, K.-S. Tan, V. Akylas, and several others provided much technical design help. U. Bar-Gadda, R. Razouk, J. Kosteleck, D. MacArthur, and C. Skeeters provided assistance in fabricating the chip. LTX/Trillium, Inc. provided great assistance and free use of their test equipment. G. Labonville and T. Berry at Trillium helped extensively in testing the demonstration chips.

References

[1] S. Anderson, J. Earle, R. Goldschmidt, and D. Powers, "The IBM system/360 model 91 floating point execution unit," *IBM J. Res. Develop.*, pp. 34-53, Jan. 1967.

[2] L. Cotten, "Maximum rate pipelined systems," in *AFIPS Proc. Spring Joint Comput. Conf.*, 1969, pp. 581-586.

[3] J. L. de Jong *et al.*, "Single-polysilicon layer advanced super high-speed Bi-CMOS technology," in *Proc. IEEE Bipolar Circuits Technology Meeting* (Minneapolis, MN), Sept. 1989, pp. 182-185.

[4] B. Ekroot, "Optimization of pipelined processors by insertion of combinational logic delay," Ph.D. dissertation, Elec. Eng. Dept., Stanford Univ., Stanford, CA, Sept. 1987.

[5] C. T. Gray *et al.*, "Theoretical and practical issues in CMOS wave pipelining," in *Proc. VLSI '91* (Edinburgh, Scotland), Aug. 1991, pp. 9.2.1-9.2.10.

[6] D. Joy and M. Ciesielski, "Placement for clock period minimization with multiple wave propagation," in *Proc. 28th Design Automation Conf.* (San Francisco, CA), June 1991, pp. 640-643.

[7] F. Klass and J. M. Mulder, "CMOS implementation of wave pipelining," Delft Univ. Technol., Delft, Netherlands, Tech. Rep. 1-68340-44(1990)02, Dec. 1990.

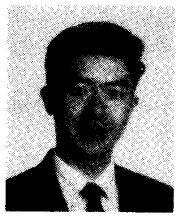
[8] Q. Lin and P. Xia, "The design and implementation of a very fast experimental pipelining computer," *J. Comput. Sci. Technol.* (Beijing), vol. 3, no. 1, pp. 1-6, 1988.

[9] D. Wong, G. De Micheli, and M. Flynn, "Inserting active delay elements to achieve wave pipelining," in *Proc. ICCAD '89* (Santa Clara, CA), Nov. 1989, pp. 270-273.

[10] D. Wong, G. De Micheli, and M. Flynn, "Designing high-performance digital circuits using wave pipelining," in *Proc. VLSI '89* (Munich, W. Germany), Aug. 1989, pp. 241-252.

[11] D. Wong, "Techniques for designing high-performance digital circuits using wave pipelining," Ph.D. dissertation and Comput. Syst. Lab. Tech. Rep. CSL-TR-92-508, Elec. Eng. Dept., Stanford Univ., Stanford, CA, Aug. 1991.

[12] C. Zukowski and K. Shum, "A matched-delay CMOS TDM multiplexer cell," in *Proc. IEEE Int. Conf. Comput. Design* (Rye Brook, NY), pp. 352-355, Oct. 1988.



Derek C. Wong (S'88-M'92) received the B.S. degree in electrical engineering and computer sciences from the University of California at Berkeley in 1985. He received the M.S. and Ph.D. degrees in electrical engineering in 1986 and 1991, respectively, from Stanford University, Stanford, CA. He is currently a post-doctoral fellow in the Department of Electrical Engineering at Stanford University.

Since 1988 he has conducted research principally on techniques for designing circuits using wave pipelining. His dissertation research resulted in novel CAD algorithms for designing wave-pipelined circuits and an LSI chip demonstrating wave pipelining. He has also published a new algorithm for high-speed division and an architecture for high-speed logic resolution. His current technical interests are high-speed digital systems, computer organization, CAD, and multimedia.

Giovanni De Micheli (S'79-M'83-SM'89) received the Dr.Eng. degree, summa cum laude, in nuclear engineering from the Politecnico di Milano, Italy, in 1979, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California, Berkeley, in 1980 and 1983, respectively.

He is Associate Professor of Electrical Engineering and, by courtesy, of Computer Science at Stanford University, Stanford, CA. From 1984 to 1986 he worked at the IBM T.J. Watson Research Center, Yorktown Heights, NY, where he was Project Leader of the Design Automation Workstation group. Previously he held positions at the Department of Electronics of the Politecnico di Milano, Italy, and at Harris Semiconductor, Melbourne, FL. His research interests include several aspects of the computer-aided design of integrated circuits with particular emphasis on automated synthesis, optimization, and verification of VLSI circuits. He is co-editor of the book *Design Systems for VLSI Circuits: Logic Synthesis and Silicon Compilation* (Martinus Nijhoff, 1987). He was also co-director of the Advanced Study Institute on Logic Synthesis and Silicon Compilation, held in L'Aquila, Italy, under the sponsorship of NATO in 1986 and in 1987.

Dr. De Micheli is an Associate Editor of the IEEE Transactions on Circuits and Systems and of *Integration: The VLSI Journal*. He was Technical and General Chairman of the International Conference on Computer Design in 1988 and 1989, respectively. He has served as a member of the technical committees of the ICCD, ICCAD, and DAC Conferences. He served also as a member of the executive committee of the New York Chapter of the Computer Society in 1985 and 1986. He received the 1987

Best Paper Award of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN and a Best Paper Award at the 20th Design Automation Conference, in June 1983.

Michael J. Flynn (M'56-SM'79-F'80) is a Professor of electrical engineering at Stanford University, Stanford, CA. His experience includes ten years at IBM Corporation working in computer organization and design. He was also a faculty member at Northwestern University and Johns Hopkins University, and the director of Stanford's Computer Systems Laboratory from 1977 to 1983.

Prof. Flynn has served as vice president of the Computer Society and was founding chairman of CS's Technical Committee on Computer Architecture, as well as ACM's Special Interest Group on Computer Architecture.

Robert E. Huston (M'64) received the B.S.E.E. degree with highest honors in 1963 from the University of Missouri, Rolla, and the M.S.E.E. degree in 1964 from Purdue University, West Lafayette, IN.

He has been employed in the design and application of semiconductor automatic test equipment since 1967. From 1967 to 1980 he was with Fairchild Systems (now Schlumberger) engaged in automatic test equipment systems design and applications, holding the position of Applications Director since 1978. From 1980 to 1983 he was with GenRad Semiconductor Test. In 1983 he co-founded Trillium Corporation, now a division of LTX Corporation, in San Jose, CA. LTX Corporation designs and manufactures digital and linear automatic test equipment. He was awarded the position of Fellow at LTX in 1988 and is an officer of the corporation. He is presently engaged in applications research for high-frequency testing of digital VLSI components.