Performance Analysis of Wave Pipelining in Circuits with Redundancies

R. Ozgun and F.J. Meyer

Department of Electrical and Computer Engineering, College of Engineering

Abstract In conventional pipelined circuits there is only one datawave active in any pipeline stage at any time; therefore, the clock speed of the circuit is limited by the maximum stage delay in the circuit. In wave pipelining, the clock speed depends mostly on the difference between the longest and shortest path delays. In some circuit designs there are redundant elements to make the circuit less sensitive to noise, to provide higher signal driving capability, or other purposes. Also, some circuit designs include logic to detect the early completion of a computation, or to guarantee that the worst physical path delay does not equate to the worst computational delay. Prior tools for wave-pipelined circuits do not account for such design features. This research develops a computer-aided design tool to determine the maximum clock speed for wave-pipelined circuits with redundant logic or where otherwise the internal circuit timing depends on the input signal values.

1. Introduction

Wave pipelining is an innovative design alternative to conventional pipelines—first proposed by Cotton [1], who called it *maximum rate pipelining*. In conventional pipelined circuits, the clock speed of the circuit is limited by the maximum stage delay in the circuit. Cotton observed that the rate at which a *datawave* can propagate through the circuit depends mostly on the difference between the longest and the shortest path delays. This result is a consequence of the fact that multiple datawaves can propagate from the primary inputs to the primary outputs of the circuit simultaneously, increasing the throughput, without the use of intermediate registers (see Fig. 1 and Fig. 2.).

The industry is faced with some hurdles to make this a practical design technique: retiming the circuit to decrease the difference between maximum and minimum delay times [2] [3], increasing the fault tolerance of the circuit and making it less sensitive to noise [4], delay buffer design, developing synthesis techniques and computer-aided design tools for wave pipelining [5], etc.

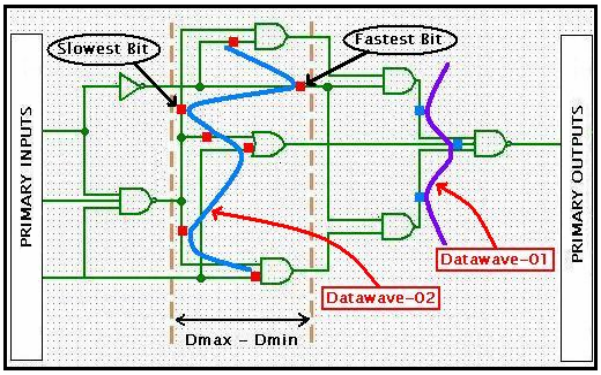

Fig. 1. A random circuit showing data bits with different speeds

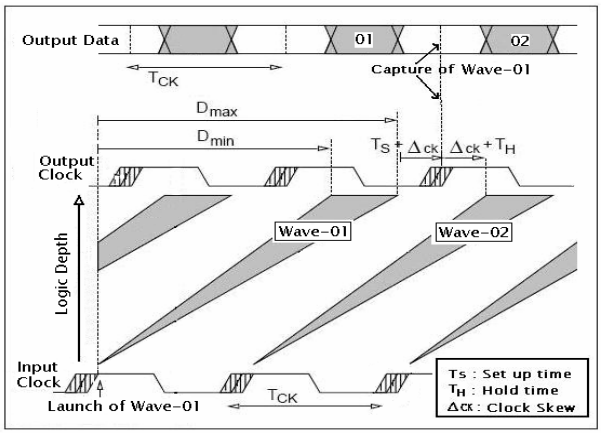

Fig. 2. Spatial diagram for wave pipelining showing delay contours

Using the timing model presented in [2], the spread of signals traveling along the longest and the shortest paths through the logic block can be approximated as *delay contours,* shown in Fig. 2. Each signal's timing is uncertain, and there is also some skew in the timing between different signals (data skew). As a consequence of this model, clock speed in wave pipelining was formulated as [5],

$$T_{CK} > (D_{MAX} - D_{MIN}) + (T_S + T_H) + 2\Delta_{CK} \qquad (1)$$

However, this is not a completely sufficient requirement. Consider a conventional pipeline. The portion of Eq. (1), T_S+T_H, corresponds to the time that a flip-flop in a conventional pipeline must be *exclusively* used by one bit of data. The portion of Eq. (1), $2\Delta_{CK}$, corresponds to the

uncertainty for the clock arriving at the flip-flops (clock skew). The portion, D_{MAX}-D_{MIN} would be just D_{MAX} in a conventional pipeline; however there would be a separate condition requiring $D_{MIN}>T_S+T_H+2\Delta_{CK}$. That additional constraint on D_{MIN} is to satisfy the exclusion requirement at the flip-flops.

In a wave pipelined circuit, all devices in the circuit must satisfy the exclusion requirement. In prior design tools, the operational model of devices was simplified so that the timing uncertainty of signals could only increase as the signal propagated through the circuit. More precise delay models reveal that signal uncertainty can fluctuate as it progresses through the circuit. Our tool analyzes this accurately. This fluctuation is why we require that Eq. (1) apply to all devices in the wave-pipelined circuit.

2. Analysis and Delay Variations

A main task of the research is how to completely characterize the delay variations, noting that wave pipeline performance is mostly determined by the difference of path delays in a circuit. Future development of this research will include buffer insertion and Boolean expression synthesis.

For a CMOS gate, there will be different delay values for different input combinations. For example, consider a 2-input NOR Gate;- when *both* inputs are set to logic 0, it will pull up by switching on two serially-connected PMOS transistors, and we will get a time constant of $2R_{mos}C_{load}$.- when *both* inputs are made logic 1 values, it will pull down by switching on both of the two parallel NMOS transistors and the time constant will be $0.5R_{mos}C_{load}$ in this case.

Thus, when calculating minimum and maximum delays, we need to take into account the fact that delay varies conspicuously depending on the input values to each device. We illustrate this using the simulation software we have developed and for the example Boolean expression in Eq. (2) implemented in two-level NAND-NAND logic.

$$y(a,b,c,d) = ab + ac + bc' + cd \qquad (2)$$

Setting $R_{mos}C_{load}$ to 60 time units, the shortest and longest paths through the circuit have delays *45* and *420*. However, these delays never occur, because it is not possible to set the inputs to all logic devices to values where they all operate at their slowest or fastest. Our simulator calculates that this implementation of this function has $D_{MIN}=140$ and $D_{MAX}=360$. One way to get delay D_{MIN} is when the input vector $<a,b,c,d>$ switches from $<0,0,0,1>$ to $<1,1,1,1>$.

Since we care mostly about the values for D_{MAX} - D_{MIN} rather than just D_{MAX}, it is possible to invent new logic design techniques that possibly increase D_{MAX} but reduce the difference. A novel design technique that we plan to investigate further is the partitioning of input spaces and the use of majority logic calculations. The Boolean expressions in Eq. (3) illustrate the technique.

$$y(a,b,c,d \mid c) = a + d \qquad (3.1)$$
$$y(a,b,c,d \mid c'+d) = b + c \qquad (3.2)$$
$$y(a,b,c,d \mid c'+d') = ac + bc' \qquad (3.3)$$

There are three regions to the input space: *c', cd,* and *cd'.* Each of the three subsidiary functions is correct in two of the three regions. In fact, we could reduce the third function to just bc' since the chosen functions for the other two spaces are accidentally both correct over a larger portion of the input space. In each region, at least two of the three functions equal *y.* Therefore, the inputs to the majority gate are *a+d, b+c,* and *ac+bc'* and the majority gate output is *y.*

Our simulator calculated D_{MIN} as *140* and D_{MAX} as *270* over all pairs of consecutive input vectors. The results show there is a potential improvement in D_{MAX} - D_{MIN} from *220* to *130*. Although D_{MAX} and D_{MIN} values are larger in the majority-based design, the technique improves throughput when the circuit is operated as a wave pipeline.

3. Conclusion

The main goal of this research was developing a computer-aided design tool to determine the delay variations for wave-pipelined circuits with redundant logic or where otherwise the internal circuit timing depends on the input signal values. We also mentioned an alternative design technique that increases the performance of wave-pipelined circuits.

4. References

[1] L.W Cotton, "Maximum-rate pipeline system," *AFIPS Proc. Spring Joint Computer Conference*, vol. 34, pp. 581-586, May 1969

[2] C. T. Gray, W. Liu, R. K. Cavin, III, "Timing Constraints for Wave-Pipelined Systems," *IEEE Trans. on Computer-Aided Design*, vol. 13, no. 8, pp. 987-1004, August 1994

[3] N. V. Shenoy, R. K. Brayton, and A. L. Sangiovanni-Vincentelli, "Minimum padding to satisfy short path constraints," in *Proceedings of the 1993 IEEE/ACM International Conference on Computer-Aided Design*, pp.156-161, November 1993

[4] T**.** Feng, N. Park, Y. Kim, F. Lombardi, F. J. Meyer, "Reliability Modeling and Assurance of Clockless Wave Pipeline," *19th IEEE International Symposium on (DFT'04),* pp. 442-450, October 2004

[5] W.P. Burleson, M. Ciesielski, F. Klass, W. Liu, "Wave-Pipelining: A Tutorial and Research Survey," *IEEE Trans. on VLSI Systems,* vol. 6, no. 3, pp. 464-474, September 1998