

WAVE PIPELINES VIA LOOK-UP TABLES

Eduardo I. Boemo, Sergio López-Buedo, and Juan M. Meneses

E.T.S.I. Telecomunicación - Universidad Politécnica de Madrid
28040 Ciudad Universitaria. Madrid - España. e-mail: ivan@die.upm.es

ABSTRACT

Look-up tables (LUTs) allow the delay of digital blocks with different types of gates or different logic depth to be equalized; thus, they could be a useful building block for the construction of wave pipelined circuits. In this paper, this alternative is explored by using a RAM-based FPGA. An experimental LUT-based wave pipeline 7-bit array multiplier has been constructed. The main results, for an intentionally skewed clock synchronization strategy, show that it is possible to obtain throughputs as high as 80 MHz with 8 waves running in a 13-LUT logic depth combinational circuit. The prototype presents a continuous range of frequency operation and exhibits an acceptable dependence with power supply variations. In terms of fast-prototyping, wave pipelining on FPGAs allows the designers to obtain a unique combination of high-throughput and minimum-latency.

1. INTRODUCTION

The construction of maximum-rate circuits or wave pipelines is centered on the equalization of all path delays. It allows several waves to travel through the circuit without interference, with a clock period smaller than the delay of the maximum path [1]. In a wave pipeline, the throughput is just limited by the difference between the maximum and minimum path delay, plus the clock skew, the rise/fall time and setup/hold values of the registers [2]. In principle, this technique speeds up a combinational circuit without increasing either the synchronization power (due to the avoidance of intermediate registers) or the spurious activity power (due to the inherent path equalization), or the initial latency (due to the maximum delay of the datapath not being increased by the insertion of intermediate registers).

Some recent applications of wave pipelines are [3]- [5]. These circuits are implemented using special ECL and CMOS cells designed to achieve gates with data-independent propagation delay. However, to the best of our knowledge, the usefulness of look-up tables as elements to achieve the equalization have not been reported in scientific literature. In this paper, this alternative is explored by using commercial RAM-based FPGAs.

Even considering that registers are "free" in most FPGA chips, these devices exhibit some advantages for researchers interested in wave pipeline topics: LUTs mask the delay of different logic functions, and also have been designed as data-independent delays as possible in order to improve simulation accuracy [6]; the architecture exhibits high regularity that leads to delay equalization; the knowledge *a priori* of each FPGA element delay (wire segments, LUTs, and other interconnection resources) makes possible the

performance of path equalization; powerful layout editors exist; and finally, the fast design cycle and reprogrammability of this technology allows lots of prototypes to be built, adjusted, measured, and compared without significant cost.

2. WAVE PIPELINING USING LUTs

The construction of a wave pipeline requires both non data-dependent delay technologies as well as exact models rather than conservative worst-case delay specifications. However, it is possible to make use of commercial technologies if a categorical matching strategy [4] is adopted. It allows the final path unbalance due to different gaps between model and actual delay of each circuit element to be minimized. Applied to FPGAs, categorical matching leads to composing each path with the same number of LUTs, *pips*, *magicboxes*, etc., avoiding any trade between LUT and wire delays. As a consequence, in an FPGA-based wave pipeline all bits pass through the same number of LUTs, in the same way that all bits pass the same number of registers in a conventional pipeline.

In order to analyze the feasibility of LUT-based wave pipelines, the array multiplier proposed by H. Guild [7], [8] has been selected as case-study for the experiments. The operands were limited to 7-bit, the maximum array size that fitted into the FPGA chip selected (a Xilinx XC4005PC84-6) for the equalization strategy adopted.

The equalization task was separated into successive steps. First, each array cell was fitted into one CLB (*configurable logic block*), and then, the placement was carried out maintaining the layout as similar as possible to the topology; that is: spatial regularity as a previous step towards delay equalization. Then, chains of additional LUTs were used just to delay the least significant output results (the fastest paths). After completing the placement, the routing of a set of local wires between neighbouring cells was carried out, and then this pattern was copied in the rest of the array, transforming homogeneous wiring into delay equalization, but also reducing errors during the manual layout process.

However, the equalization of circuits with both global and local communications is inherently difficult. The extra delay of global lines must be equalized by increasing the delay of the local ones. In this case, the global communications were implemented using horizontal FPGA *long lines* which have a delay between 3 and 4 ns for the chip selected. To balance such a delay, all local communications were routed through one, two or three *magicboxes*, depending on the load of the global line of each segment. The use of half *long lines* to implement the last array global communication was not considered in order to maintain the regularity.

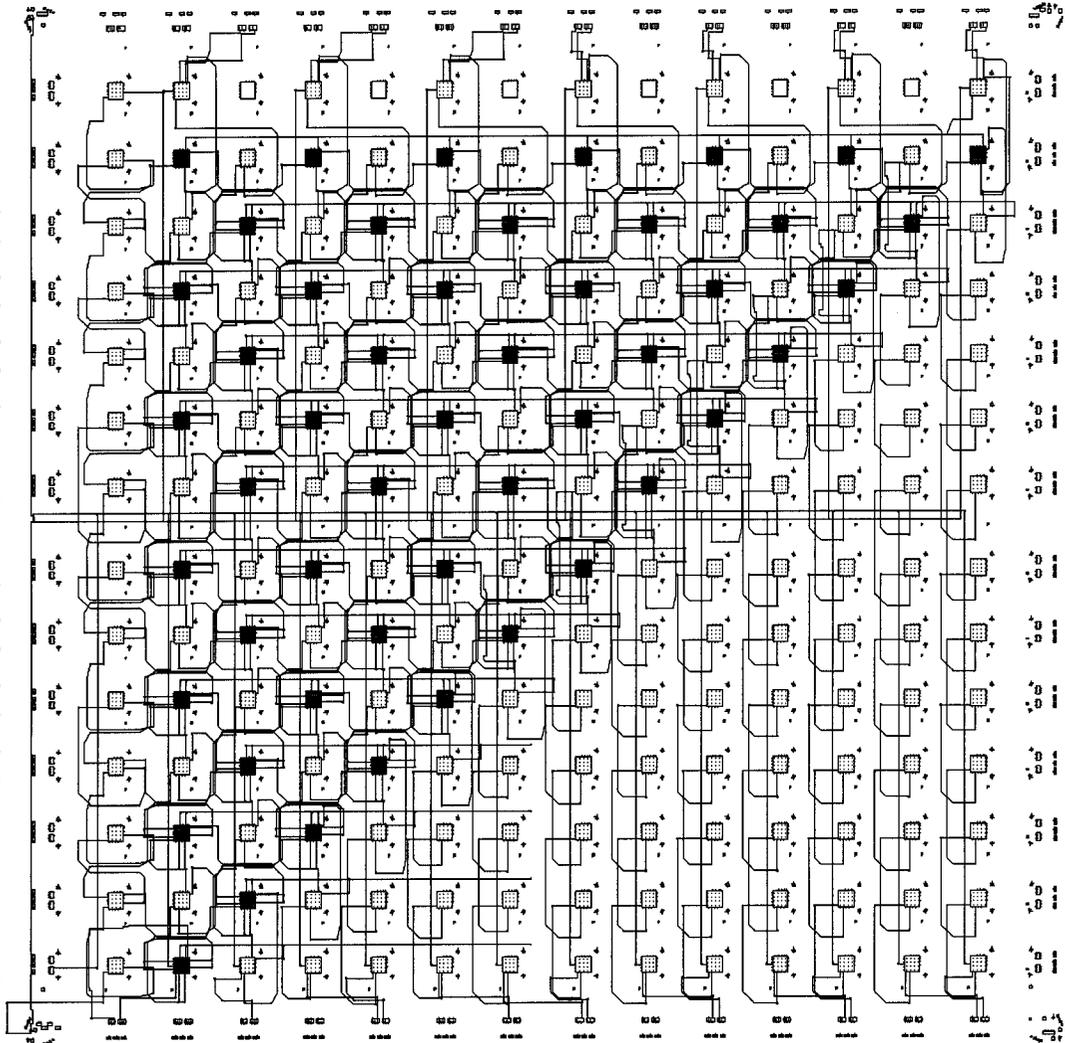


Fig.2: Wave pipeline 7-bit Guild array multiplier in a XC4005PC84

After the placement and routing, the last step is to distribute the clock (all the I/O are registered). The first versions were synchronized using single-phase clocking. It made the timing process straightforward (due to the standard clock lines and buffers of the FPGA being utilized), and also allowed the wave pipeline to be embedded into a synchronous system, sharing the common clock signal. However, the main drawback of a single-phase clocked wave pipeline is a set of frequency bands where the circuit does not work [9], as well as a closer dependence on power supply and temperature variations. The main results of these experiments were presented in [10].

The clock delay process was performed in two phases. The gross adjustment was made leaving the outputs unregistered, and measuring the relative position of clock edges and data waves. Then, a fine adjustment was made by increasing or reducing the wiring delay in steps of around 1 ns, until an optimal combination of bandwidth and ΔV_{cc} operation range was obtained. The main problem found was the data dependent delay of wiring, which spoiled the clock duty cycle. Best result corresponded to a 14-LUT clock path that minimized the

routing delay. The skewed clock line was ended on a clock buffer in order to drive the output registers. It was also connected to an output pad to synchronize the logic analyzer.

In Fig.1 the actual final layout is shown. The colour of the original picture has been modified in order to separate the CLBs in: processing (dark gray), delaying (light gray), and unused (white). The first CLB column corresponds to the clock delay path.

3. EXPERIMENTAL RESULTS

All measurements were taken at room temperature. Input data included a 2^{16} random vectors as well as a set of numbers that produce the toggle of almost all the output. This second sequence facilitated the detection of both zero-clocking and double-clocking. Chip dissipation was improved by the addition of a heat sink with forced airflow. The principal prototype characteristics are summarized in Table I. All results correspond to measured data otherwise specified.

Topology	Guild 7-bit
Technology	Xilinx XC4005PC84-6
Synchronization strategy	Intentionally skewed clock
Partitioning, placemet and routing	Full manual
Bandwidth	0 to 80 MHz continous
Number of CLBs	190
Number of registers	28
Logic depth	13 LUTs
Number of waves	8 @ 80 MHz
Maximum data path unbalance	1.9 ns (simulation)
Latency	95 ns @ 80 MHz
Average power consumption (random vectors)	3485 mW @ 80 MHz

Table I: Main prototype characteristics.

The maximum speed, 80 MHz, ended up 10 times higher than the value predicted by the timing analyzer tool (*xdelay*), based on long path delay calculation - the factor between simulation result and real frequency of operation was measured, for the same chip, as 1.3 for non-equalized conventional pipelines, as is depicted in Fig.2. Some versions ran up to 82.5 MHz at the expense of accentuated power supply voltage dependence.

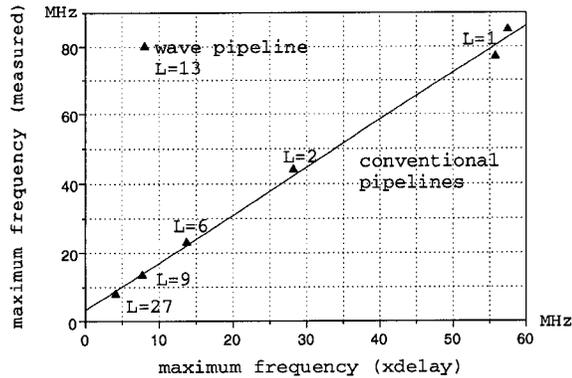


Fig.2: Simulated versus measured maximum frequency (L=logic depth in LUTs)

The circuit makes use of 190 CLBs: 49 to datapath processing, 127 for datapath equalization and 14 for clock delaying. The logic depth is 13 LUTs and all I/O pads are registered. Maximum unbalance between all input-to-output paths resulted 1.9 ns (simulation). The delay of a typical path can be broken down to: 62% for LUTs, 34% for wires, and finally, 4% for registering (simulation).

A comparison in terms of power consumption shows that the skewed clock wave pipeline gives the worst value. It consumes more than both a fine-grain conventional pipeline and the previous single-phase wave version (Fig.3). All prototypes have equal off-chip power, because of they were implemented using exactly the same FPGA chip, pads, vectors, and PCB. Moreover, both single-phase and skewed clock wave pipelines have almost identical datapath layouts; thus, the power consumption gap between them can be assigned to the chain of extra CLBs utilized to delay the output clock in the latter.

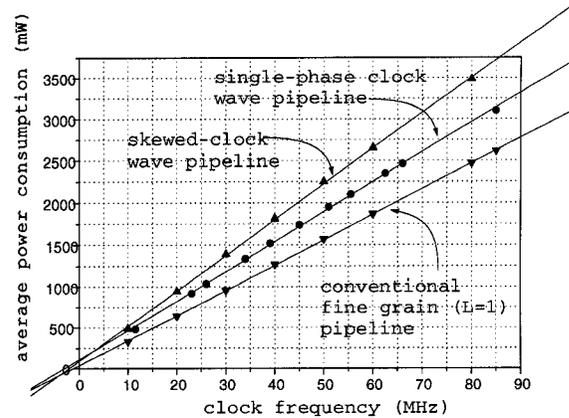


Fig.3: Power consumption versus clock frequency

The extra power consumption of wave pipelines respect to conventional pipelines depends on three factors: datapath spurious activity, the number of registers, and extra equalization logic. Even considering that wave pipelines reduce spurious activity (for example, a non-equalized combinational array resulted in a maximum of 40 intermediate values between two consecutive correct products: ten times the number measured for the wave array with the output registers removed), the reduction effect is even bigger on a fine-grain conventional pipeline, that has just 1-LUT logic depth. Moreover, the main advantage of wave pipelining in terms of power consumption, the reduction in the number of registers (28 instead of 278 corresponding to a conventional fine-grain pipeline) is not significant: in the technology selected the synchronization power is not dominant [11].

In terms of reliability, the skewed-clock wave pipeline exhibited an acceptable dependence with supply voltage variations (Fig.4). Maximum test voltages were limited to 5.5 v in order to avoid excessive power consumption. At 80 MHz, the range of operation is between 4.56 to 5.5 v, better than the previous single-phase clock wave pipeline prototype (4.88 to 5.13 v), and similar to the range corresponding to a conventional pipeline version of 85 MHz of bandwidth (4.57 to 5.5 v), although the advantage of the conventional pipeline become important as the frequency operation was decreased.

The main positive consequence of wave pipelining was the small latency achieved: 95 ns @ 80 MHz. Thus, a unique combination of high throughput and low latency was obtained. Considering that, for n=7 bits, the array has 13 cells in the longest path, and each of them requires one CLB (the number of outputs of each Guild cell does not allow the designer to fit two of them in a CLB), a classic pipeline must

have 13 stages in order to reach a throughput over 80 MHz; it would imply a latency nearly twice that corresponding to the wave version.

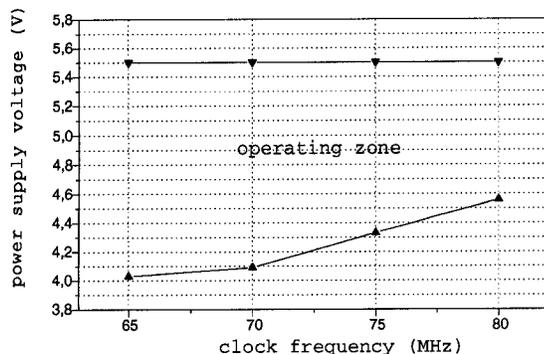


Fig.3: Power supply voltage variation range versus frequency.

4. CONCLUSIONS

The feasibility of LUT-based wave pipelines has been demonstrated. The equalization methodology presented, a combination of categorical matching and maximum regularity, allows us to discard path unbalance as the main factor that limits the speed on current FPGAs.

From a fast prototyping perspective, there is no significant advantage in terms of power consumption, speed, or reliability in respect to classic pipelining; the distinctive result of the wave effect is just the achievement of a unique combination of high speed and low latency: the first similar to conventional fine-grain pipelines, and the second practically equal to the combinational circuit.

Finally, a LUT-based FPGA-like architectures oriented to supporting a wave pipeline running mode could be more effective in terms of occupation, if extra buffers with propagation delay matched with LUT delay were included.

ACKNOWLEDGES

This research has been supported by the CICYT of Spain under contract TIC92-0083.

REFERENCES

- [1] L. Cotten, "Circuit Implementation of High-Speed Pipeline Systems", *Proc. Fall Joint Computer Conference (AFIPS)*, 1965.
- [2] D. Wong, "Techniques for Designing High-Performance Digital Circuits Using Wave Pipelining", Tech. Rep. No. CLS-TR-92-508, Stanford U: February 1992.
- [3] W. Burleson, C. Lee and E. Tan, "A 150 MHz Wave Pipelined Adaptive Digital Filter in 2 μ m CMOS", *VLSI Signal Proc. VII, Ed. J. Rabaey*, pp.296-305. New York: IEEE Press, 1994.
- [4] W. Liu, T. Gray, D. Fan, W. Farlow and T. Hughes, "A 250-MHz Wave Pipelined Adder in 2- μ m CMOS", *IEEE Journal of Solid-State Circuits*, Vol.29, No.9, pp.1117-1127. Sept. 1994.
- [5] F. Klass and M. Flynn, "A 16x16-bit Static CMOS Wave-Pipelined Multiplier", *Proc. ISLASS 94*, pp.143-146. IEEE Press 1994.
- [6] Application Note "The Tilde De-Mystified", *The Programmable Logic Array Data Book*. Xilinx Inc. 1994.
- [7] H. Guild, "Fully Iterative Fast Array for Binary Multiplication and Addition", *Electronic Letters*, pp.263, Vol.5, No.12, June 1969.
- [8] T. Hallin and M. Flynn. "Pipeline of Arithmetic Functions". *IEEE Trans. on Computer*, pp.880-886. August 1972.
- [9] C. Gray, W. Liu and R. Cavin, "Wave Pipelining: Theory and Implementation", Kluwer Academic Publishers. 1992.
- [10] E. Boemo, S. López-Buedo, and J. Meneses, "The Wave Pipeline Effect on LUT-based FPGA Architectures", *Proc 1996 ACM SIGDA FPGA Workshop*, Monterey, California, February 1996 (in press).
- [11] E. Boemo, G. González de Rivera, S. López-Buedo and J. Meneses, "Some Notes on Power Management on FPGAs", *Proc. Fifth Int. Workshop on Field Programmable Logic and Applications*, pp.149-157, Oxford, U.K. Ed.: W. Moore & W.Luk. Berlin: Springer-Verlag 1995.