# Short Papers

## Some Experiments About Wave Pipelining on FPGA's

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

***Abstract*— Wave pipelining offers a unique combination of high speed, low latency, and moderate power consumption. The construction of wave pipelines is benefited by the use of gates and buffers with data-independent delays and the knowledge of the interconnection delays. These two features are present in several SRAM-based field programmable gate arrays (FPGA's): look-up tables (LUT's) allow the designer to mask the delay of different gates and combinational functions, and the timing characteristics of each wire segment are *a priori* known. This work describes a set of experiments about wave pipelining on FPGA's. The results show that a 13-LUT logic depth circuit mapped on an XC4005PC84-6 runs as high as 85 MHz (single phase clocking) or 80 MHz (intentionally skewed clocking), exhibiting a latency of 95 ns. This high throughput/latency ratio is unattainable using classic pipelining.**

***Index Terms*— Arithmetic, high performance, low-power design, performance tradeoffs.**

### I. Introduction

PIPELINING is one of the most powerful methods in high-speed digital design. The evolution and main concepts of this technique can be reviewed in [1]–[7]. In short, pipelining allows the designer to increase the throughput $N$ times by dividing the circuit in $N$ stages running concurrently. However, a concrete fine-grain implementation exhibits a more modest value of speed-up, usually between $N/2$ and $N/3$, in spite of the technology utilized or the skill of the designers. In actual high-speed pipelines, the number of extra registers inserted in the datapath is several times the gate count; the clock period is mainly limited by the setup and propagation delays of the registers; and the latency is significantly increased. As a consequence, the increment of speed in a classic pipeline follows a law of diminishing returns: each additional logic depth reduction implies that the number of registers is almost duplicated to obtain a small speedup. Probably, the first technologist that experienced the empirical obstacles of pipelining was Henry Ford. He divided the Model-T motor assembly into 48 operations and obtained a speed-up by a factor of three; meanwhile, the montage of the magneto was split into 29 parts, allowing the time to be reduced only from $20'$ to $13'10''$ [8].

Manuscript received March 31, 1997; revised November 1, 1997. This work was supported by the CICYT of Spain under Contract TIC95-0971.

E. I. Boemo and S. López-Buedo are with the Laboratorio de Microelectrónica, E.T.S. Informática, Universidad Autónoma de Madrid, Madrid 28049 Spain.

J. M. Meneses is with the Grupo de Arquitecturas Digitales, E.T.S.I. Telecomunicacion, Universidad Politecnica de Madrid, Madrid 28040 Spain.

Publisher Item Identifier S 1063-8210(98)02953-9.

The above limitations are also present in field programmable gate array (FPGA) implementations: in a high-speed pipeline, the fraction of the stage delay corresponding to wiring and registering can be as much as the 80% of the clock period, even taking into account the fact that the routability is benefited from a low pin/CLB ratio (most of the CLB's are employed only for registering). As a result, just an small fraction of the period is used to transform the data. This situation is illustrated in Fig. 1. The chart shows the clock period composition corresponding to a set of 8-bit Guild multipliers [9], pipelined with different logic depth [5] (this topology, also analyzed in [2], [3], is depicted in Section III). They were implemented in an XC3090PC84-100 [10] using default placement-routing options.

Registers are included in pipelines to guarantee a separation between different data, so that the fastest bits of a new datum do not reach the slower bits of the previous one. However, the same result can be obtained if all circuit paths have the same delay. This alternative leads to the wave pipeline (WP) technique or maximum rate circuits. The principles of WP were enunciated in [11] and analyzed in [12], but reached renewed interest after Flynn reintroduced the technique in [13]. The WP effect is based on the equalization of all paths in order to allow several waves of data to travel along the circuit with a separation several times smaller than the maximum combinational delay. In this case, the minimum clock period is not limited by the longest path, but by the difference between the maximum and minimum path delays, plus the clock skew, the rise/fall time, and the setup/hold time of the I/O registers [14]. The avoidance of intermediate registers allows the WP to maintain the latency of the original circuit.

Recent applications of WP are diverse: digital filters [15], adders [16], memories [17], multipliers [18], [19], and other arithmetic modules [20]. All these circuits have been implemented using special ECL or complementary metal–oxide–semiconductor (CMOS) gates designed to achieve fixed propagation delays. In this paper, the application of the WP technique in commercial look-up table (LUT)-based FPGA's is explored. In the next section, the main concepts and drawbacks of WP are reviewed. In Section III the experimental results are summarized.

### II. Wave Pipelines and FPGA's

The use of a single-phase clock on WP gives rise to an important drawback: a set of frequency bands where the circuit does not work. This phenomenon occurs every time the

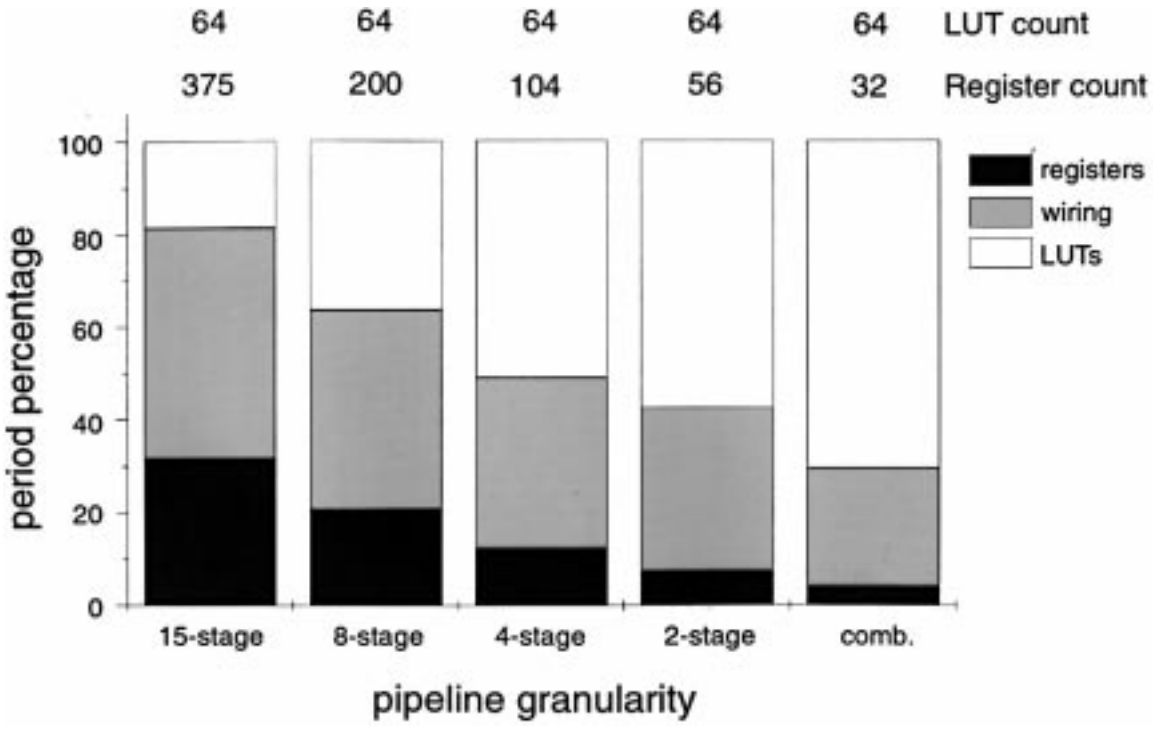


Fig. 1. Clock period composition as function of the pipelining degree in an 8-bit Guild multiplier.

operation mode (that is, the number of waves that run together inside the circuit) is changed. Moreover, each new operation band is narrower than the previous one. As a consequence, parameters that affect propagation delay such as power supply voltage, process variations, or temperature establish a limit in the practical number of waves. Detailed studies in single-phase WP synchronization can be found in [21]–[23].

The intentionally skewed clock strategy [21] was proposed by Gray *et al.* as an alternative to overcome the above problem. In this scheme, the output data is registered using a new edge, that lags behind the input clock a time equal to the longest datapath delay plus the setup time. This guarantees that the results will be correctly captured: a clock edge will always follow the arrival of every wave. This method leads to more robust WP circuits as well as facilitating the multiple-stage WP synchronization. However, it violates the synchronous design principle by introducing a new clock signal.

FPGA's exhibit several characteristics suitable for wave pipelining: 1) LUT's hide the delay of different logic functions, and also have been designed with similar rise and fall times to improve simulation accuracy [24], 2) the architecture exhibits a high regularity that leads to delay equalization, 3) the knowledge *a priori* of each FPGA element delay (wire segments, LUT's, and other interconnection resources) makes the equalization task possible, 4) powerful layout editors exist, and finally, 5) the fast design cycle and reprogramability of this technology allows prototypes to be built, measured, and adjusted without significant cost.

The use of the WP technique in FPGA's could seem inappropriate: since registers and clock distribution elements are included in the chips, they are "free" components and their use does not produce an area penalty, in contrast to other VLSI technologies. However, two peculiarities of WP's justify an exploration of this technique in FPGA's: the high-throughput/latency ratio attainable and the potential power reduction; considering that the equalization not only eliminates registers and clock lines, but also diminishes spurious activity. In addition, the WP effect can be accidentally produced in an FPGA if the circuit has nearly the same number of LUT's in all paths. This situation could lead the designer to an erroneous perception of the speed achieved and limitations of the prototype.

The implementation of WP's requires exact models rather than the min–max specification of VLSI foundries. Nevertheless, it is possible to make use of commercial technologies by applying a strategy called categorical matching [16]. It allows the path unbalance caused by the different grades of modeling accuracy of each circuit component (gates, wires, vias, etc.) to be minimized. Using this approach, all data in a FPGA-based WP must pass through the same number of LUT's, in the same way that all data passes through the same number of registers in a conventional pipeline. Moreover, all paths should be composed of the same number of other FPGA elements: "pips," "magicboxes," "long lines," [10] and so on, as far as possible.

The experiments presented in this work are focused on single-stage WP's. In this case, the longest path is not modified; just the delay of the shorter ones are increased to achieve the balance. Thus, the minimum latency characteristic of the WP technique can be fully exploited. Considering that WP is just another way of trading speed for additional area and design time (as well as exhibiting lower reliability than conventional pipelines), its potential benefits and drawbacks must be evaluated for each technology. In this way, the main

goal of this paper has been to determine if wave pipelining can constitute a useful technique in the area of fast prototyping.

## III. Experimental Results

The WP effect on FPGA's was accidentally observed by the authors of this paper during an exploration of the dependence between glitches, logic depth and power consumption in conventional pipelines. During these experiments [26], [27], a fine-grain pipeline multiplier (one LUT between registers) was manually partitioned and placed on an XC3090PC84-100. The routing was done automatically using the default parameters of the tools. After finishing the circuit, all the registers were eliminated using the Xilinx layout editor [25], but their associate LUT in each CLB was maintained (the direct input pin "di" was not used). In this way, a combinational circuit like the original pipeline but with the same number of LUT's (instead of registers) in all paths was obtained. As a result, not only a low-power but also a high-speed prototype was obtained. The Xilinx static timing analyzer showed a delay of 168.6 ns in the longest path, a value that corresponds to a bandwidth close to 6 MHz; however a throughput of 27 MHz was measured. Moreover, an unstable higher band from 34 to 35.6 MHz was also detected and several vectors were correctly multiplied even at 43 MHz. The experiments showed the feasibility of wave pipelining in FPGA's (a possibility that was also mentioned by F. Klass in the 1994 Stanford seminar EE385B) but also led to an XC3000-based pipeline paradox: registers constitute the scaffolding to construct a high-speed block; but if they are removed after the implementation process, the circuit still holds part of its high-speed attribute.

In this paper, a complete methodology for the construction of WP on the XC4000 series [10] is presented. In all cases, Guild multipliers have been selected as a case-study. This circuit consists of an array of $n^2$ cells composed of an AND gate plus a full-adder. Each of them have four inputs and two outputs and can be fitted into one CLB. The 7-bit version is shown in Fig. 2.

Departing from a common equalized datapath, two WP prototypes were constructed and measured using a Xilinx XC4005PC84-6 chip: the first with intentionally skewed clock, and the second using single-phase synchronization. All the layout procedure (partitioning, placement and routing) was performed manually using the Xilinx layout editor [25]. The equalization strategy was based on the transformation of spatial regularity into delay equalization. First, each array cell was mapped into one CLB. Then, they were placed as similarly as possible to the original topology, in order to take advantage of the uniformity in both the circuit array and the FPGA structure. This method automatically led to a coarse delay balance, but constrained the word length to 7 bits.

During the routing process, the main source of unbalance was caused by the different combination of global and local lines in the paths. This problem was solved by assigning the horizontal global wires to long lines. To compensate the delay of long lines, all the local lines were routed through two magicboxes, using a regular pattern of selected segments of around 3 ns that was repeated along the array. In this way, a highly equalized layout was created, with a constant 13-LUT logic depth in all paths. The total occupation was 182 CLB's. The maximum unbalance between all the I/O paths, determined using the Xilinx timing analyzer, resulted in less than 2.1 ns, whereas the longest path exhibited a delay of 135.3 ns. Without the WP effect, this value would limit the throughput to below 8 MHz. Unlike conventional pipeline prototypes, in this case most of the path delay (59%) corresponds to LUT's, a fact that is the key to the low latency of the circuits.

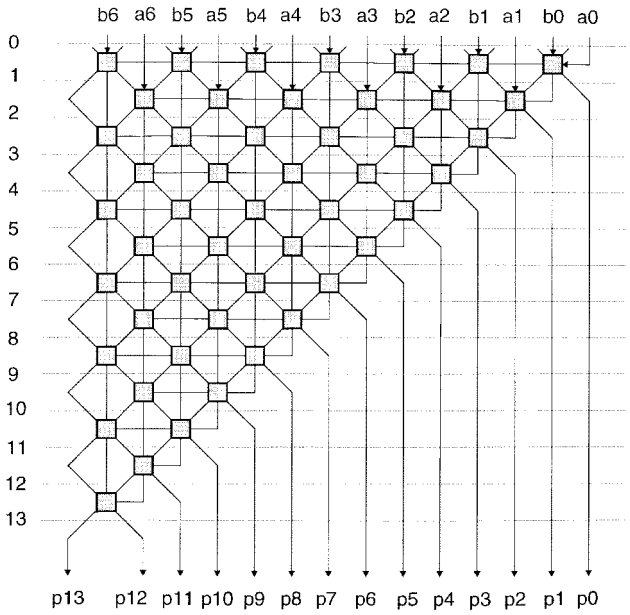


Fig. 2. A 7-bit Guild multiplier. Equitemporal lines are shown.

The final layout is depicted in Fig. 4. In spite of the transformations accomplished, the similarity to the Guild topology of Fig. 2 is evident. The color of the original picture has been modified in order to show the CLB functions: processing (dark gray), delaying (light gray), and unused (white). Using this layout, the two WP versions with different clocking schemes were constructed. The use of FPGA's offered a unique opportunity to perform a an accurate comparison between the synchronization schemes: both WP circuits have exactly the same layout, pads, chip sample, PCB, and output loads.

For the intentionally skewed clock prototype, the first column of CLB's was used to lag the clock, creating a new path whose delay matched the datapath one. The clock wiring process was performed in two phases, taking full advantage of the reconfiguration features of FPGA's. The first adjustment was made leaving the outputs unregistered, and measuring the relative position of clock edges and data waves using a high-speed oscilloscope. Then, fine tuning was done by increasing or reducing the skewed clock wiring in steps of around 1 ns, until an optimal combination of bandwidth and tolerance to Vcc fluctuations was obtained. After that, the output registers were enabled again by modifying each IOB using the layout editor. The main problem found was the different rise and fall times of the interconnection resources, which spoiled the delayed clock duty cycle. The best result corresponded to a 14-LUT clock path that minimized the weight of the routing delay. Finally, the clock delay chain was ended in a dedicated

clock buffer in order to drive the output registers. It was also connected to an output pad to trigger the logic analyzer used to test the prototype. The synchronization path made use of eight extra CLB's.

The transformations of the skewed-clock circuit to get the second WP with single-phase clocking were straightforward. The same layout was employed, but the column of CLB's to delay the clock was eliminated. Instead of them, one of the eight FPGA dedicated clock lines was used to control all the I/O registers.

All measurements were made using $2^{16}$ random vectors as well as a set of 16 operands that produce the toggle of almost all the output pins. The second type of data facilitated the detection of phenomena like double-clocking and zero-clocking [28]. The prototypes operated as fast as fine grain pipelines, but using 28 registers instead of 278. Considering the nature of FPGA's, this reduction is not significant in terms of chip size, but definitely affects the latency achieved. The highest operational frequency band (measured) for the single-phase clocked design was between 83–85 MHz, a value similar to that obtained using conventional pipelining in the same chip sample. The intentionally skewed clock prototype exhibited a continuous range of operation up to 80 MHz. The circuits ran more than ten times faster than the value predicted by the timing analyzer tool, whereas the factor between simulation results and the actual frequency of operation was measured for the same chip as 1.3 for nonequalized pipelines.

The actual I/O delay resulted in 95 ns. It was measured in a version of the layout with the output registers removed. A maximum of eight waves running inside the circuit without interference was determined by simultaneously sampling the input vectors and the output results. Therefore, the latency of the prototype is nine clock cycles if the output registers are included. Considering that, for $n = 7$ bits, the Guild topology has 13 cells in the longest path, and each of them requires one CLB, a fine-grain classic pipeline must have 13 stages to run over 80 MHz. However, the circuit would have a latency of 14 clock cycles. Pipelining every two cells could reduce the latency to eight clock cycles, but reaching 80 MHz would imply a nonrealistic timing budget for the chip selected. Thus, wave pipelining offers the FPGA end-user a unique high-speed low-latency combination.

In terms of area, the equalization cost is high taking into account that a hand-made classic pipeline could be fitted in less CLB's (a 13-stage, 7-bit, Guild array requires 278 registers). However, the final number of CLB's in the wave version is a consequence of two choices: regularity and categorical matching. Both strategies simplify the equalization task but require additional logic. In principle, the use of extra routing to balance LUT delays could reduce the number of CLB's occupied even less than the quantity needed by the equivalent classic pipeline. However, an equalization without categorical matching would require a previous characterization of all chip elements to be performed.

The analysis of WP in terms of power consumption demonstrates the relationship between low spurious activity and path balance. For instance, sampling the output at 400 MHz, a maximum of 40 intermediate values between two consecutive

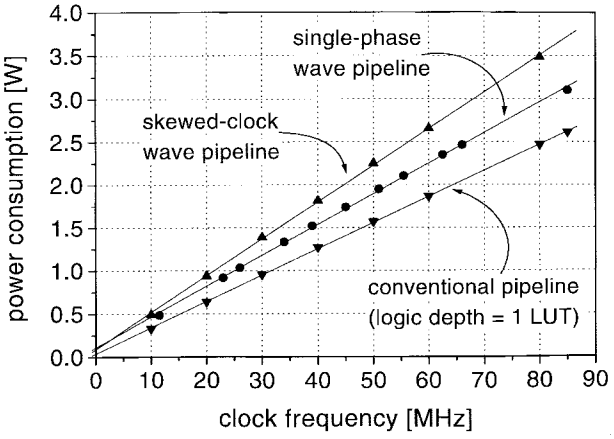


Fig. 3. Power consumption versus frequency.

results were measured in an equivalent nonequalized combinational array: ten times the number observed in a balanced version with the same logic depth. Thus, the experiments confirm that speed and power consumption are correlated at the implementation level: both conventional and wave pipelining increase the speed and reduce the power simultaneously. However, the wave prototypes consumed a little more than fine-grain pipeline arrays (Fig. 3).

The main disadvantage of WP is its strong dependence on power supply voltage variations, especially in the single-phase version. For example, at 84 MHz (the middle of the last frequency band), changes in power supply below 4.88 V or above 5.13 V produced erroneous outputs. On the contrary, the intentionally skewed clock WP was less sensitive to power supply variations. At 80 MHz, it withstood Vcc variations from 4.56 to 5.5 V (upper voltage was limited to 5.5 V to avoid an excessive power consumption). This tolerance was similar to that observed in equivalent classic pipelines.

Previous circuits have been full-manually implemented (partitioning, placement, and routing). However, the WP effect was also observed in automatic place and route implementations, whenever the arrays have the same number of LUT's in all paths (this possibility was suggested to the authors by an anonymous reviewer of the FPGA'96 Symposium). This fact can be explained considering that, if the logic depth is high, the path delays are dominated by the LUT's. Although some wires may have a large delay, the sum of the interconnection delays will always be lower than the sum of the LUT ones, considering that an array exhibits a Pareto–Levy distribution of wiring delays [27]. Thus, in some cases, the WP effect can be spontaneously produced in an FPGA. In that situation, the designer must be aware of this possibility in order to correctly interpret the nature of the better performance obtained. To illustrate the magnitude of this problem, a 7-bit combinational Guild multiplier with registered I/O was manually partitioned so that it had 13 LUT's in all its paths. This circuit was automatically placed and routed, using the default options of the tool. The main results were as follows: the longest path delay was 120.9 ns, and the shortest one was 109.4 ns. The circuit worked up to 64 MHz (measured), with five waves inside it: more than seven times the speed predicted by the

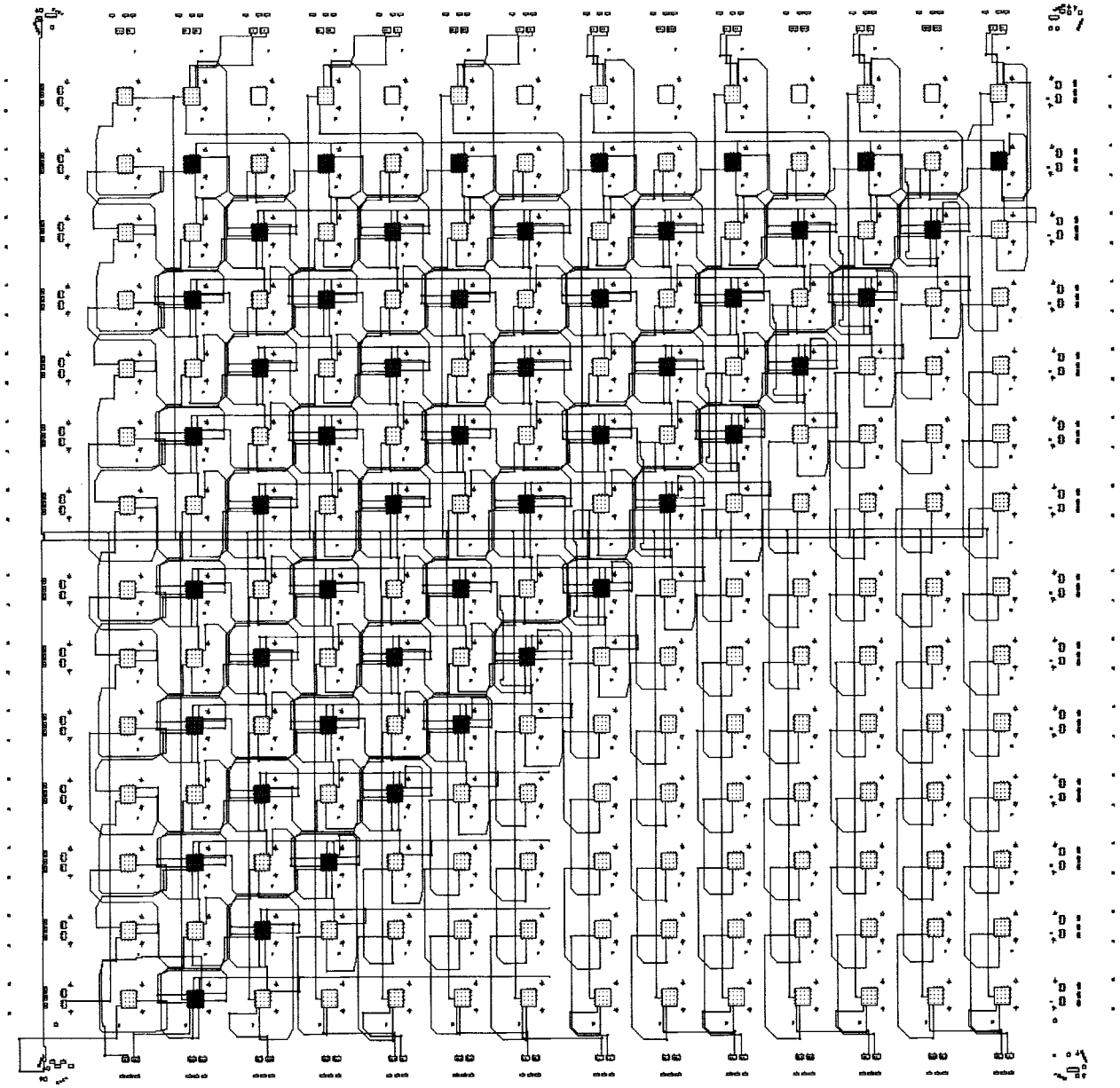

Fig. 4. XC4005 wave pipeline array multiplier layout (intentionally skewed clock version).

timing analyzer tool. The highest operation band was very narrow: just 1 MHz.

## IV. Conclusions and Future Work

The feasibility of constructing WP's using LUT's has been demonstrated, and their distinctive characteristic of low-latency, high-throughput, and reasonable power consumption (slightly higher than the conventional pipelining one) has been obtained using commercial chips and tools. Additionally, it has been confirmed that the wave pipelining effect can even be achieved using an automatic place and route, as long as the circuit has nearly the same number of LUT's in all paths.

Two synchronization alternatives have been explored. Single-phase clocking allows the designer to embed a WP block as part of a synchronous system, but its main disadvantage is the power supply and temperature dependence. Where endurance against environmental variations is a must, an intentionally skewed clocking allows the WP to tolerate similar power supply variations as the classic pipelines.

From the fast prototyping perspective, WP's offer the designer the only alternative for getting high speed (similar to that of fine-grain conventional pipelines), and low latency (similar to that of the combinational version of a given topology). This can help the gap with other relatively faster VLSI technologies to be reduced, when FPGA's are used for chip emulation. In that case, the higher sensitivity of single-phase WP's to temperature, power supply and run-to-run variations is not significant, considering that just one laboratory prototype will be needed. However, the designer must take into account the principal obstacle of single-phase WP: the narrow ranges of valid clock frequencies and the complexity of their prediction from data book information.

Designers interested in verifying the feasibility of WP in other FPGA technologies can perform a simple preliminary experiment. First, a set of LUT's must be cascaded in order to implement a 1–bit wide path crossing the chip. Each LUT must be configured as an inverter. After that, using an oscilloscope, the input–output delay can be measured by introducing a low-frequency square wave. Then, the input frequency must be gradually increased until its shape at the output pad appears highly distorted. This effect is caused by differences between the rise and fall times of the elements that compose the path. As a consequence, the last frequency value indicates the bandwidth of a perfectly equalized WP in that chip.

As a final remark, the efficiency of wave pipelining in FPGA's could be improved by the manufacturer if extra buffers with a delay similar to the LUT were included in future chips. A set of these elements uniformly distributed along the chip would allow the designer to equalize some paths without consuming extra LUT's. This would not only make the circuit faster, but would also reduce the spurious activity, and therefore the power consumption.

## REFERENCES


[1] L. Cotten, "Circuit implementation of high-speed pipeline systems," in *Proc. Fall Joint Comput. Conf.*, 1965, pp. 489–504.

[2] T. Hallin and M. Flynn, "Pipeline of arithmetic functions," *IEEE Trans. Comput.*, pp. 880–886, Aug. 1972.

[3] J. Deverell, "Pipeline iterative arithmetic arrays," *IEEE Trans. Comput.*, pp. 317–322, Mar. 1975.

[4] R. Jump and S. Ahura, "Effective pipeline of digital systems," *IEEE Trans. Comput.*, vol. C-27, pp. 855–865, Sept. 1978.

[5] C. Hauck, C. Bamji, and J. Allen, "The systematic exploration of pipelined array multiplier performance," in *Proc. ICASSP 85*, New York: IEEE Press, 1985, pp. 1461–1464.

[6] M. Hatamian and G. Cash, "A 70-MHz 8-bit $\times$ 8 bit parallel pipelined multiplier in 2.5-$\mu$m CMOS," *IEEE J. Solid-State Circuits*, vol. SC-21, pp. 505–513, Aug. 1986.

[7] T. Noll, D. Schmitt, H. Klar, and G. Enders, "A pipelined 330-MHz multiplier," *IEEE J. Solid-State Circuits*, vol. SC-21, pp. 411–416, June 1986.

[8] G. Burrel, Ed., *Crónica de la Técnica.* Barcelona, Spain: Plaza & Janes, 1989, pp. 524–525.

[9] H. Guild, "Fully iterative fast array for binary multiplication and addition," *Electron. Lett.*,vol. 5, no. 12, p. 263, June 1969.

[10] Xilinx Inc., *The Programmable Logic Data Book*, 3rd ed. San Jose, CA: Xilinx, Inc., 1994.

[11] S. Anderson, J. Earle, R. Goldschmidt, and D. Powers, "The IBM system/360 model 91 floating point execution unit," *IBM J. Res. Development*, vol. 11, pp. 34–53, Jan. 1967.

[12] L. Cotten, "Maximum-rate pipeline systems," in *Proc. Sprint Joint Comput. Conf.*, 1969, pp. 581–586.

[13] D. Wong, G. De Micheli, and M. Flynn, "Designing high-performance digital circuits using wave pipelining," in *Proc. VLSI 89*, G. Musgrave and U. Lauther Eds. North Holland, The Netherlands: Elsevier Science, 1990, pp. 241–252.

[14] D. Wong, "Techniques for designing high-performance digital circuits using wave pipelining," Stanford Univ., Tech. Rep. CLS-TR-92-508, Feb. 1992.

[15] W. Burleson, C. Lee, and E. Tan, "A 150 MHz wave pipelined adaptive digital filter in 2 $\mu$ CMOS," in *VLSI Signal Proc. VII*, J. Rabaey, Ed. New York: IEEE Press, 1994, pp. 296–305.

[16] W. Liu, T. Gray, D. Fan, W. Farlow, and T. Hughes, "A 250-MHz wave pipelined adder in 2-$\mu$m CMOS," *IEEE J. Solid-State Circuits*, vol. 29, pp. 1117–1127, Sept. 1994.

[17] K. Nowka and M. Flynn, "Wave pipelining of high performance CMOS static RAM," Stanford Univ., Tech. Rep. CLS-TR-94-615, Jan. 1994.

[18] D. Ghosh and S. Nandy, "Design and realization of high-performance wave-[ipelined 8$\times$ 8 multiplier in CMOS technology," *IEEE Trans. VLSI Syst.*, vol. 3, pp. 36–48, Mar. 1995.

[19] F. Klass and M. Flynn, "A 16 $\times$ 16-bit static CMOS wave-pipelined multiplier," in *Proc. ISCAS 94*. New York: IEEE Press, pp. 143–146.

[20] M. Flynn, K. Nowka, G. Bewick, E. Schwarz, and N. Quach, "The SNAP project: Toward sub-nanosecond arithmetic," in *Proc. 12th Symp. Comput. Arithmetic*, 1995, pp. 75–82.

[21] C. Gray, W. Liu, and R. Cavin, *Wave Pipelining: Theory and CMOS Implementation*. New York: Kluwer Academic, 1994.

[22] C. Chang, E. Davidson, and K. Sakallah, "Maximum rate single-phase clocking of a closed pipeline including wave pipelining, stopability and startability," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 1526–1545, Dec. 1995.

[23] W. Lam, R. Brayton, and A. Sangiovanni-Vicentinelli, "Valid clock frequencies and their computation in wavepipelined circuits," *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 791–807, July 1996.

[24] Xilinx Inc., "The tilde de-mystified," in *The Programmable Logic Data Book*, 2nd ed. Xilinx Inc., San Jose, CA: 1994, p. 9.3.

[25] *XACT Reference Manual*, Xilinx Inc., San Jose, CA, 1995.

[26] E. Boemo, S. Lopez-Buedo, G. Gonzalez de Rivera, and J. Meneses, "On the usefulness of pipelining and wave pipelining as low-power design technique," in *Proc. PATMOS95 Fifth Int. Workshop*, pp. 252–263, Oldenburg, Germany: Oct. 1995.

[27] E. Boemo, "Contribución al diseño de arrays VLSI con paralelismo de grano fino," Ph.D. dissertation, School of Telecommun. Eng., Tech. Univ. Madrid, Spain, Jan. 1996 (http://www.ii.uam.es/~ivan/papers/htm).

[28] J. Fishburn, "Clock skew optimization," *IEEE Trans. Comput.*, vol. 39, pp. 945–951, July 1990.