

HIGH-PERFORMANCE PARALLEL ADDITION USING HYBRID WAVE-PIPELINING

James Levy, Jabulani Nyathi, and José Delgado-Frias
School of EECS, Washington State University
Pullman, Washington 99164-2752
(jlevy, jabu, jdelgado)@eecs.wsu.edu

Abstract—Pipelining digital systems has been shown to provide significant performance gains over non-pipelined systems and remains a standard in microprocessor design. The desire for increased performance has seen a push for deeper pipelines, as well as the introduction of pipelining schemes such as wave-pipelining and hybrid wave-pipelining. In this paper we present a hybrid wave-pipelined parallel adder that operates at 1.79 GHz, 42% performance improvement compared to that of a super-pipelined adder. The simulations have been performed using a modest 0.25 μm technology. The three stage hybrid wave-pipelined parallel adder sustains a total of 8 unrelated data waves within the pipe. Another performance benefit achieved by using the the hybrid wave-pipelining scheme is the lessening of delays associated with clock skew and clock distribution.

I. INTRODUCTION

High performance data path circuits continue to be a topic of interest especially as technologies are scaled to the nanometer regime. Adders fall under this group and have been the subject of in-depth analysis for decades. Careful optimization of adders and other data path circuits will grow in importance as methods to reduce power while maintaining or improving performance are sought. There exist numerous adder implementations each with good attributes and some drawbacks. Examples include ripple carry adders, carry look-ahead adders, carry skip adders, and carry select adders [1] just to name a few. Each one of the above mentioned examples has numerous variants designed to enhance performance. The advent of wide data paths requiring up-wards of 64-bit additions has further intensified the need for adder optimization. As the number of input bits increases, so does the delay associated with the computation of the carries. The desire to reduce the delays associated with carry propagation has resulted in novel adder architectures and implementations [2], [3], [4]. In this paper we present a pipelined parallel adder that employs the novel hybrid wave-pipelining scheme to attain significant performance gains over conventional pipelined adders. In Section 2 we briefly discuss pipelining schemes and their performance related evaluations, we also analyze data dependent delays of a select few logic gates as it relates to wave- and hybrid wave-pipelining. In Section 3 we present the results of a 32-bit hybrid wave-pipelined adder and some concluding remarks appear in Section 4.

II. PIPELINING SCHEMES

Pipelining allows unrelated data waves to be overlapped in execution leading to an increase in throughput by a factor

N for an n stage system. The scheme has shortfalls in that (i) the stage with the longest delay sets the pipeline's clock frequency. This leads to underutilization of logic since stages that complete their computations within a time shorter than the clock cycle time remain idle for some fraction of the clock cycle. (ii) Each pipeline stage operates on just one data wave, i.e. there is no overlapping of unrelated data waves within a single stage. Data must completely clear a stage and be latched at an intermediate latch before new data can be admitted into the pipe stage. (iii) the latches might require elaborate clock networks to manage skew and distribution (depending on the depth of the pipe and stage complexity).

To address some of these issues *wave-pipelining* [5], [6], [7], [8], has been revived. The wave-pipelining scheme offers faster clock rates by minimizing the delay difference between the system's short and long paths. The logic path is designed to be long enough to sustain several unrelated data waves. This enables the removal of intermediate registers which reduces clock loading. The scheme achieves a balanced pipeline by inserting buffers in the shorter data paths. A wave-pipelined system operates asynchronously locally while it externally has the appearance of a synchronous system. The speed-up achieved is similar to that of conventional pipelining, but at a higher frequency. Some of the shortfalls of wave-pipelining include (i) vulnerability to process, voltage and temperature variations [9] (ii) possibilities of overshooting the longest path's delay while padding the shorter paths, (iii) intermediate nodes becoming difficult to observe in the absence of latches.

A novel pipelining scheme termed *hybrid wave-pipelining* reported in [10], [11], [12] has been shown to further reduce the clock cycle time by re-introducing a few of the conventional pipeline registers and pipelining the clock to match the logic path delays. The hybrid wave-pipelining scheme minimizes the delay difference per stage and allows the stages to be long enough to sustain more than a single wave. Pipelining the clock offers a way to manage clock skew and distribution since the clock now experiences similar delays to those of the data. Re-introducing some of the registers of conventional pipelining allows for the intermediate nodes to become easily observable and to some degree data dispersion can be tolerated better than in wave-pipelining due to the intermediate register use. It must be noted that the registers used in hybrid wave-pipelining do not give the scheme fine granularity as in conventional pipelining since more than a single data

wave can be sustained within a stage. The approach allows for further clock cycle time reduction since delay minimization is performed per stage. The hybrid wave-pipelining scheme's short-falls include: (i) increased design time which we expect can be reduced by making the design semi-custom. Regular structures such as registers can be placed in libraries. (ii) Susceptibility to process, temperature and voltage variations.

A. Pipelined Adders

In this sub-section we reference a couple of conventionally pipelined (CP) adders and highlight the significant issues the designs address. We proceed to present the hybrid wave-pipelined (HWP) adder architecture and how it can be manipulated to form either a conventional- or wave-pipelined (WP) adder. Pipelining improves throughput at the expense of latency, however, once the pipe is filled we can expect one data item per unit of time. Some of the conventional pipelined adder designs that have been reported include one that uses overlapped clocks in an effort to eliminate sources of overhead [13]. The 4-bit carry propagate adder employs a series of three registers to equalize the delays in adding the four bits and has a three cycle latency [13]. Time borrowing is performed to shorten the critical path and the adder design has been realized in $0.35 \mu\text{m}$ technology. A 32-bit carry-select adder in $0.25 \mu\text{m}$ technology, operating at 1.67 GHz is reported in [14]. A number of pipeline registers are introduced and in a similar fashion as in [13] several of these registers are inserted in series to equalize data arrival times at adder units. The gain in speed is achieved by clocking sub-circuits faster than would be possible with a ripple carry adder. These two conventional pipelined adder architectures achieve path delay equalization by inserting registers. In the instances where these registers are used for delay equalization no logic is used between the set of registers, the output of one register connects directly to the input of the next. Figure 1 shows the register placement.

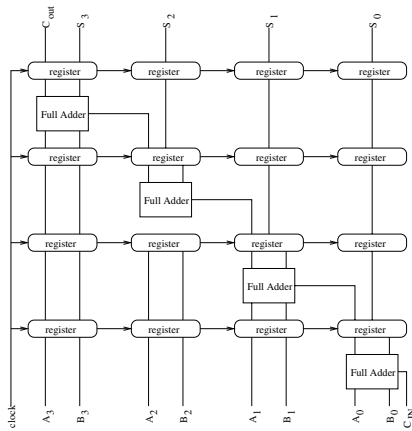


Fig. 1. Conventional pipelined adder architecture.

Introducing several registers increases the clocking overhead and skew. The adders described in [13] and [14] maybe of different types (carry propagate and carry select) however the principles leading to the use of pipelining are the same. In this

paper a parallel adder is used to present the performance gains earned due to the use of hybrid wave-pipelining. The wave-pipelining scheme of [15] eliminates the intermediate registers of conventional pipelining and can thus boast the benefit of eliminating the register overhead and associated clock skew. The hybrid wave-pipelining scheme [10], [11], [12] on the other hand uses intermediate registers but without as fine a granularity as that of conventional pipelining.

We have fashioned our hybrid wave-pipelined adder based on the wave-pipelined one reported in [15]. The equations used to describe addition are those reformulated by Brent and Kung [16] and we reproduce them here for convenience. Brent and Kung showed that the carry chain resulting from the general adder equations can be computed in parallel. The g_i and p_i terms used in their recurrence relations are expressed as follows:

$$\begin{aligned} g_i &= a_i \cdot b_i \\ p_i &= a_i \oplus b_i \end{aligned} \quad (1)$$

Brent and Kung introduced a concatenation symbol \circ and proved that:

$$(g, p) \circ (\bar{g}, \bar{p}) = (g + p \cdot \bar{g}, p \cdot \bar{p}) \quad (2)$$

In [15] it is suggested that $c_i = G_i$ for $i = 1, 2, \dots, n$, and if $i = 1$, then $(G_i, P_i) = (g_i, p_i)$. This enables for concatenation as in Equation 3.

$$(G_i, P_i) = (g_i, p_i) \circ (G_{i-1}, P_{i-1}) \text{ for } i > 1 \quad (3)$$

Using the fact that $c_i = G_i$ and computing G_i and P_i based on Equation 3 permits for the carry bits to be computed in parallel before the sum can be computed as $S_i = c_i \oplus p_i$. We note here that $s_1 = p_1$.

Figure 2 is a block diagram of the hybrid wave-pipelined adder with registers inserted to design pipe stages that permit for ease of stage delay minimization and can sustain more than a single wave within a stage.

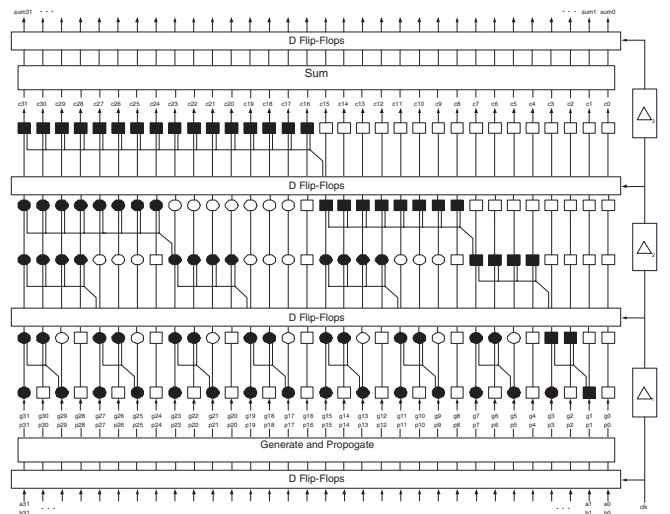


Fig. 2. HWP adder architecture.

If only the input and output registers are left while the other two intermediate latches are removed from the block diagram of Figure 2, we would have the wave-pipelined adder architecture of [15]. The major drawback of this particular hybrid wave- and the wave-pipelined adder architecture is that the fan-out of the stage that computes the $c_i = G_i$ terms grows by a factor of $2^{x/2}$. Where x denotes the number of the adder's inputs. A 32-bit adder therefore has a fan-out of 16 at this stage. For data paths larger than 32-bits wide increasing the latency at the expense of a limited fan-out becomes preferable.

The hybrid wave-pipelining scheme does not require extensive data path equalization since it can rely on the intermediate registers to perform this function. Though not a necessity, padding can further allow for shorter clock cycle times. In this study the short data paths have been padded in a similar way as that of [15]. Figure 3 shows the blocks of gates that have been used to realize the computation of the carries. Shaded and blank squares/circles on the figure map directly to those of Figure 2. In the following sub-section we will discuss data dispersion issues that have led to the use of biased NAND/AND gates in place of the standard CMOS circuits.

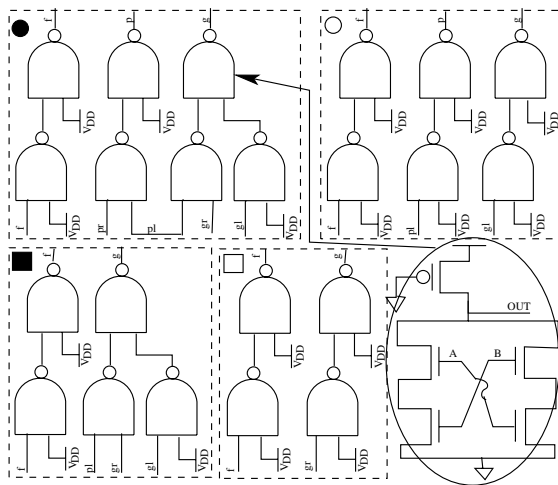


Fig. 3. Biased NAND/AND gates for carry signal computations

B. Data Dependant Delays

One of the major design issues of wave-pipelining is the need to balance data paths. The fact that there are no intermediate registers implies that data paths must be equalized. Data paths differ due to input vector patterns, fan-outs, wire delays, differing logic paths, process, temperature and voltage variations. In this subsection we discuss data dispersion due to differing input patterns. Some CMOS logic gates suffer from these data dependent delays [15]. The CMOS inverter is the easiest to design and can be designed to have a near perfect response to both a logic zero and a logic one. CMOS AND and OR gates will have different delays depending on the input signals due to the parallel pull-up or pull-down network. We provide delay statistics for both the CMOS and Biased AND gates in Table I. It is apparent from the displayed values that the biased AND gate has improved data dispersion.

TABLE I
AND GATE INPUT OUTPUT DELAYS

Cases (A,B)	Delay (ps)	
	Standard CMOS	Biased
01->11	94	71
11->00	64	88
11->01	101	85
00->11	101	73
10->11	93	71
11->10	94	85

There are some CMOS logic gates in addition to the inverter that have delays that are not heavily influenced by input data patterns. The CMOS XOR is one such logic gate and it owes this stability to the uniformity of both its pull-up and pull-down networks. The XOR boolean equations dictate that there be an equal number of series devices in the pull-up and pull-down paths.

III. 32-BIT PARALLEL ADDER RESULTS

Simulations of 32-bit parallel adders have been performed for conventional pipelining, wave-pipelining and hybrid wave-pipelining. We consider pass transistor XOR gate circuits for the conventional pipelined adder for fair comparison. We basically give the pipelined and wave-pipelined adders significant advantages over the hybrid wave-pipelined adder particularly with regard to clock generation and distribution. The clock for the pipelined adder does not experience any delays as would be the case if an elaborate clock tree is used. The fan-out of the different stages of the hybrid wave-pipelined adder has been managed differently taking into account the significantly low drive capability of future technologies.

If the conventional pipelined adder is super-pipelined its estimated delay is 800 ps. The wave-pipelined adder has a clock cycle time of 670 ps. Running the wave-pipelined adder any faster than this leads to data overrun. The hybrid wave-pipelined adder has a shorter clock cycle time than that of the other two adders. The cycle time using this scheme is 560 ps. These values are tabulated in Table II. The hybrid wave-pipelined adder runs 1.429 times faster than the conventional pipelined adder and 1.196 times faster than the wave-pipelined adder.

TABLE II
PIPELINED ADDER CLOCK CYCLE TIMES

Design	Clock cycle time (ns)	HWP Speedup
HWP	0.560	-
WP	0.670	1.196
CP 2	.8	1.429

Figure 4 shows the number of intermediate waves that stage one of the Hybrid Wave-Pipelined adder can sustain. The pipelined local clock signals clocking new data in and old data out of the stage are also shown, notice that the edges of the local clocks need not be synchronized. We have to initially fill

the stage with data before the output of the particular stage is valid thus the output remains at logic “0” until the stage computes the initial values.

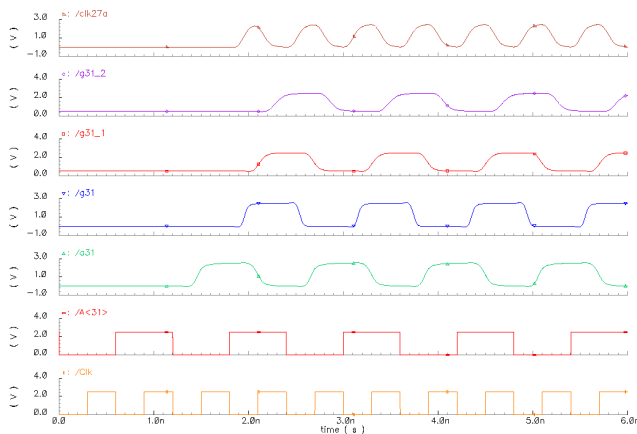


Fig. 4. HWP Adder clocks traveling with associated data.

Figure 2 shows how the registers are clocked by delayed clock signals and in the following we will provide a brief on the clocking approach. The basic idea involves delaying the clock by an amount equivalent to the stage’s delay difference ($d_{max} - d_{min}$). Each stage therefore has a local clock generated by circuitry bearing the said stage’s delay characteristics. Pipelining the clock in this manner permits each data wave to travel through the pipe stages at the same rate as its associated clock. The benefits of pipelining the clock are three-fold: (i) we eliminate the need of having an elaborate clock distribution network which might see the latches receiving the significant clock edge at differing times, (ii) should we chose to employ clock gating there is no need for additional clock pulses to flush the pipe, admitted data is guaranteed to propagate through all the pipe stages since local clocks will be generated, and (iii) there is potential for power savings “assuming” that the clock distribution network dissipates a significant amount of power. Additional clock pulses associated with attempts to flush the pipe dissipate power and they do not exist with this scheme.

It must be noted that the work on power evaluation is work in progress as of this writing. For this particular adder design inverters have been used to generate and pipeline local clocks. Simulation results show that data arrive at the registers with the associated clock signals as evidenced by the traces of Figure 4. Another important aspect of the hybrid wave-pipelining scheme is that the frequency can be slowed to allow the design to operate as a conventionally pipelined system. The benefit here could be in testing. Re-introducing intermediate latches also allows for some internal nodes to be observable.

IV. CONCLUSION

In this paper we have presented a hybrid wave-pipelined adder and shown that it outperforms both conventional- and wave-pipelined adders operating 1.429 and 1.196 times faster respectively. The three stage hybrid wave-pipelined adder is capable of sustaining 8 unrelated data waves within the pipe.

A 3-stage CP adder would only have 3 unrelated data waves within a pipe. The hybrid wave-pipelined scheme allows data to travel with its associated clock through the different pipe stages and thus allows for nodes to be observable, a difficult task to achieve in wave-pipelining. Pipelining the clock lessens clock skew (no single clock is required to drive the system’s multiple registers). In addition Hybrid Wave-Pipelining will operate at any slower clock frequency unlike Wave-Pipelining that operates on disjoint sets of points [7]. Future work will involve studying process, temperature and voltage variations and how they impact performance in the hybrid wave-pipelined scheme. The hybrid wave-pipelining approach’s power savings potential is under consideration. It must be noted that the scheme is not only limited to adders, but is applicable to other systems or subsystems that require high speed operation.

REFERENCES

- [1] N. West and K. Eshraghian, *Principles of CMOS VLSI Design: A System Perspective*, 2nd ed., Addison Wesley, NY, 1992, pp. 513-536.
- [2] P. K. Chan, M. D. F. Schlag, C. D. Thomborson, and V. G. Oklobdzija, “Delay Optimization of Carry-Skip Adders and Block Carry-Lookahead Adders,” *10th IEEE Proceedings on Computer Arithmetic*, June 26-28, 1991, pp. 154-164.
- [3] C-H. Huang, J-S. Wang, C. Yeh and C-J. Fang, “The CMOS Carry-Forward Adders,” *IEEE Journal of Solid-State Circuits*, Vol. 39, NO. 2, February 2004, pp. 327-336.
- [4] Y. Kim and L-S Kim, “64-bit carry-select adder with reduced area,” *Electronics Letters*, Vol. 37, Issue 10, May 10, 2001, pp. 614-615.
- [5] D. Wong, G. De Micheli and M. Flynn, “Designing High Performance Digital Circuits Using Wave-Pipelining: Algorithms and Practical Experiences,” *IEEE Transactions on CAD*, Vol. 12, No. 1, January 1993, pp. 25-46.
- [6] C. T. Gray, W. Liu, R. K. Cavin, III, *Wave Pipelining: Theory and CMOS Implementation*, Kluwer Academic Publisher 1994.
- [7] W. P. Burleson, M. Ciesielski, F. Klass and W. Liu, “Wave-Pipelining: A Tutorial and Research Survey,” *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, Vol. 6, No. 3, pp. 464 - 474, September 1998.
- [8] C. Gray, W. Liu, and R. Cavin III, “Timing Constraints for Wave-Pipelining Systems,” *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, Vol 13, No. 8, August 1994, pp 987-1004.
- [9] W. J. Kim and Y-B. Kim, “Automating Wave-Pipelined Circuit Design,” *IEEE Design and Test of Computers*, December-November 2003, pp. 51-58.
- [10] J. G. Delgado-Frias and J. Nyathi, “A Wave-Pipelined CMOS Associative Router For Communication Switches,” *2000 IEEE International Symposium on Circuits and Systems*, Geneva, Switzerland, pp. 391 - 394, May 2000.
- [11] J. G. Delgado and J. Nyathi “A Wave-Pipelined Network Router,” *IEEE Computer Society Workshop on VLSI 2001*, pp. 165 - 170, April 19 - 20, 2001.
- [12] J. Nyathi and J. G. Delgado-Frias, “Hybrid Wave-Pipelined Network Router,” *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, pp. 1764 - 1772, December 2002.
- [13] V. Sukumar, D. Pan, K. Buck, H. Hess, H. Li, D. Cox and M. M. Mojarradi, “Design of a pipelined Adder Using Skew Tolerant Domino Logic in a 0.35 μm TSMC Process,” *2004 IEEE Workshop on Microelectronics and Electron Devices*, September 2004, pp. 55-59.
- [14] Y. Kim, K-H.Sung and L-S. Kim, “A 1.67 GHz 32-bit Pipelined Carry-Select Adder Using the Complementary Scheme,” *2002 IEEE International Symposium on Circuits and Systems*, May 26-29, pp. I-461-I-464.
- [15] W. Liu, C. T. Gray, D. Fan, W. J. Farlow, T. A. Hughes and R. K. Cavin, “A 250-MHz Wave-Pipelined Adder in 2 μm CMOS,” *IEEE Journal of Solid-State Circuits*, Vol. 29, No. 9, September 1994, pp. 1117-1128.
- [16] R. P. Brent and H. T. Kung, “A regular Layout for Parallel Adders,” *IEEE Transactions on Computers*, March 1982, pp. 260-264.