

# Speedup of a Large Word-Width High-Speed Asynchronous Multiply and Accumulate Unit

Liang Zhou (Member IEEE) and Scott C. Smith (Senior Member IEEE)

Department of Electrical Engineering  
University of Arkansas  
Fayetteville, Arkansas, USA  
[lxz011@uark.edu](mailto:lxz011@uark.edu) and [smithsco@uark.edu](mailto:smithsco@uark.edu)

**Abstract**—This paper develops a new high-speed architecture for asynchronous Multiply and Accumulate (MAC) feedback circuitry, resulting in a speedup of 1.72 over the previous fastest  $72+32 \times 32$  asynchronous MAC in the literature.

**Keywords**—NULL Convention Logic (NCL); Gate-Level Piplining (GLP); NULL Cycle Reduction (NCR); computer arithmetic

## I. INTRODUCTION

An asynchronous delay-insensitive  $72+32 \times 32$  Multiply and Accumulate (MAC) unit was designed in [1], utilizing the NULL Convention Logic (NCL) paradigm [2], which was shown to be the fastest asynchronous MAC in the literature. The MAC performs a  $32\text{-bit} \times 32\text{-bit}$  fixed-point fractional multiplication, accepting (signed  $\times$  signed), (signed  $\times$  unsigned), and (unsigned  $\times$  unsigned)  $2^s$  complement operands; the product can be added to or subtracted from the 72-bit accumulator, and includes a multiply only option. The outputs are the 72-bit  $2^s$  complement result along with a bit to detect overflow. The resulting design, shown in Fig. 1, consists of a 5-stage bit-wise pipelined feed-forward multiplication portion and a NULL Cycle Reduced accumulation feedback loop, which achieves an average cycle time,  $T_{DD}$ , of 8.6 ns, using a 3.3V  $0.5\mu\text{m}$  CMOS process.

As detailed in [1], the MAC's throughput is limited by the accumulation feedback loop, shown in Fig. 2, which consists of a Carry-Save Adder (CSA) to add the two feed-forward multiplication partial products (PPs) with the feedback accumulator, followed by a 71-bit Ripple-Carry Adder (RCA) to generate the final accumulate value by summing the two PPs output from the CSA, and finally overflow calculation circuitry. The MAC's throughput was optimized by maximizing the throughput of the feedback loop, by pipelining it into 4 stages and then applying the NULL Cycle Reduction (NCR) technique [3], and subsequently pipelining the feed-forward portion to achieve the same, or slightly better, performance as the throughput-limiting feedback loop, which maximized performance for the entire MAC while minimizing area. Hence, if the feedback loop throughput could be increased, the throughput for the entire MAC could then be increased by re-pipelining the feed-forward portion.

Note that a RCA is used instead of a Carry-Lookahead Adder (CLA), because NCL circuit performance is based on average-case delay, not worst-case delay like in synchronous circuits; RCAs and CLAs have the same average-case delay,  $O(\log N)$  for an  $N$ -bit adder; and an NCL RCA requires substantially less area compared to the equivalent CLA. Hence, an RCA is used because it reduces area without decreasing throughput, as detailed in [4]. Also note that all registers and demultiplexers in this paper are assumed to be reset to NULL, unless otherwise specified.

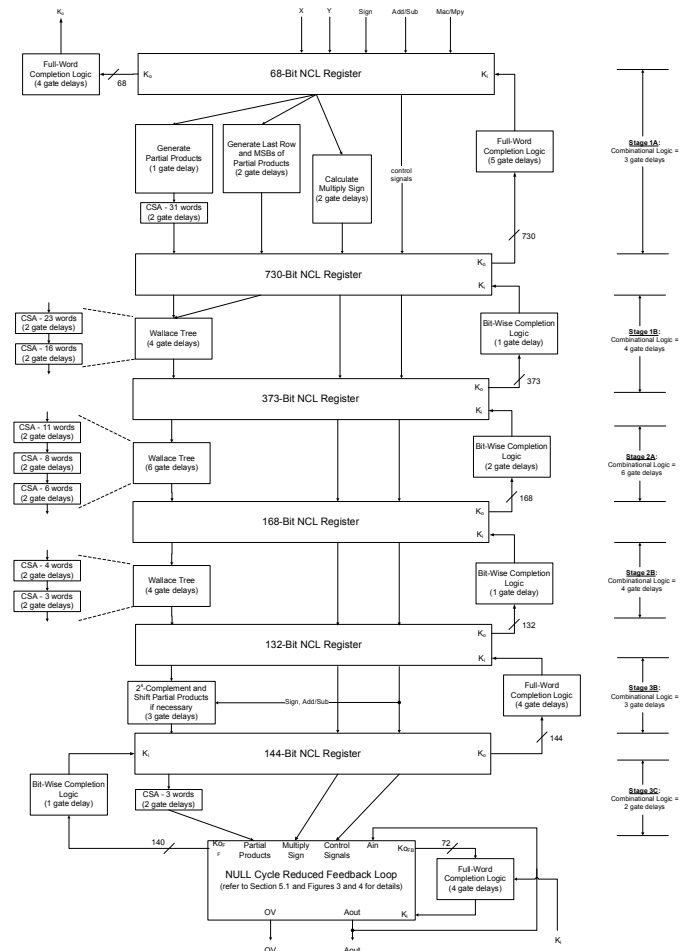


Figure 1.  $72+32 \times 32$  MAC designed in [1].

## II. REDESIGNING THE MAC FEEDBACK LOOP

Instead of calculating the final accumulator value in the feedback loop using a CSA followed by a RCA, as shown in Fig. 2, the accumulator can be fed back as two PPs in Carry-Save form, and the final accumulation performed in a subsequent feed-forward RCA. This will reduce the main portion of the accumulation feedback loop to only 2 CSAs instead of a CSA followed by a 71-bit RCA. Since the RCA is now moved out of the feedback loop, it can be fully pipelined. After final accumulation, another feedback loop is required to calculate the overflow output. Optimizing the throughput of each portion of this new MAC feedback (i.e., accumulate feedback, feed-forward RCA, and overflow feedback), shown in Fig. 3, to be at least as fast as the slowest portion, and then re-pipelining the feed-forward multiplication portion, will maximize throughput for the entire MAC.

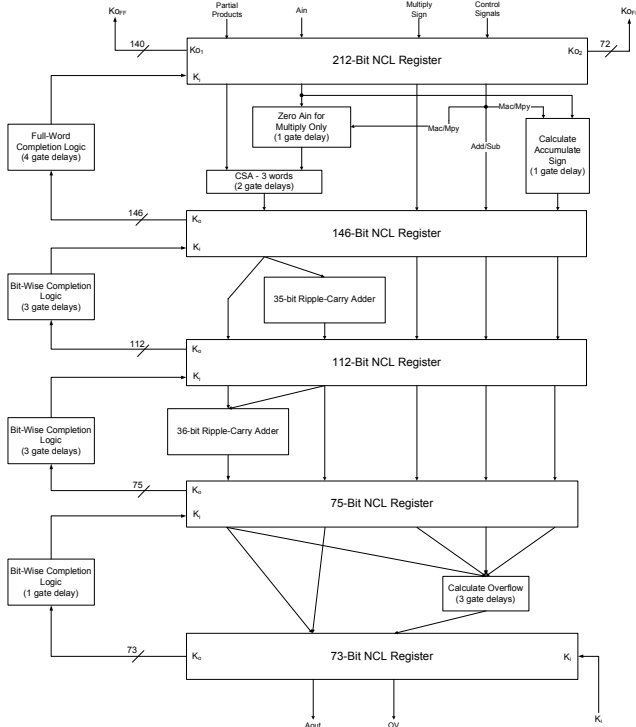


Figure 2. Accumulate feedback circuitry from [1].

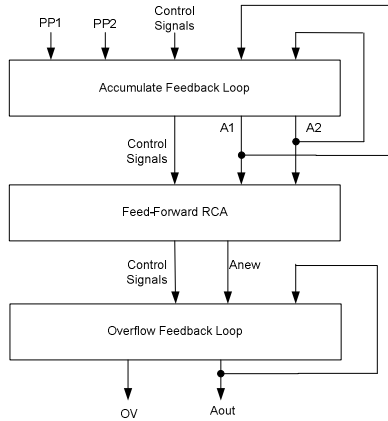


Figure 3. Redesigned MAC feedback.

### A. Accumulate Feedback

As shown in Fig. 4, the first portion of the new MAC feedback, the accumulate feedback loop, now feeds back the accumulator as two PPs in carry-save form,  $A1$  and  $A2$ , which are zeroed when the multiply only option is selected. This new feedback loop also contains two CSAs, the first which sums the two PPs from the feed-forward multiplication,  $PP1$  and  $PP2$ , with  $A1$ , and the second which sums the first CSA output with  $A2$ , generating the new accumulator value,  $A1$  and  $A2$ , in carry-save form.  $A1$  and  $A2$  are then fed back to the beginning of the accumulate feedback loop and passed to the next stage in the redesigned MAC feedback, the feed-forward RCA.

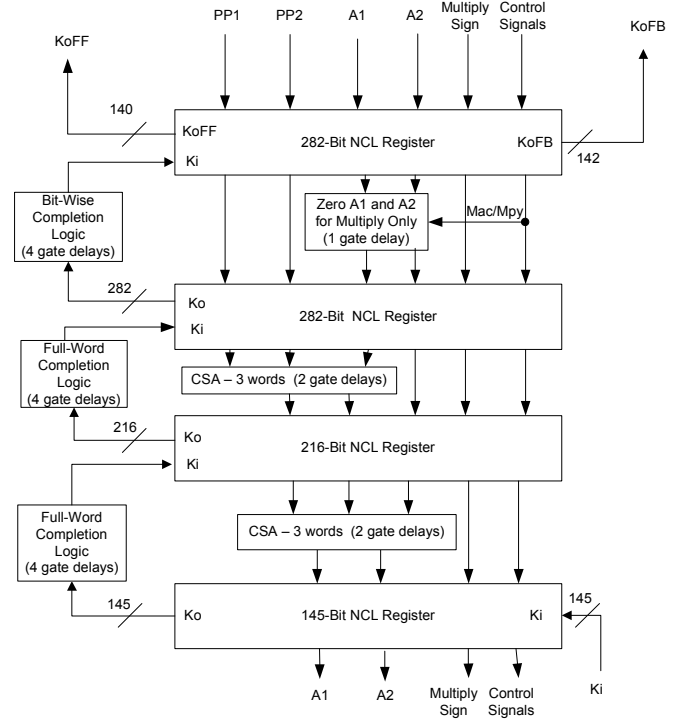


Figure 4. Redesigned accumulate feedback circuitry.

The initial  $T_{DD}$  for this new accumulate feedback loop was 9.1 ns; hence it required further optimization. Bit-wise completion [5] was applied to the first stage to both increase throughput and reduce area. The second and third stages utilize full-word completion [5], because using bit-wise completion will require more gates and was not required to make the accumulate feedback loop any faster to maximize overall MAC performance, since this feedback loop is no longer the throughput-limiting stage.

As in [1], the NCR technique [3] was applied to the accumulate feedback loop to maximize performance, resulting in a  $T_{DD}$  of 5.0 ns. NCR increases throughput of an NCL system by decreasing the circuit's NULL cycle time, without affecting its DATA cycle time, by successively partitioning input wavefronts such that one circuit processes a DATA wavefront, while its duplicate processes a NULL wavefront. The first DATA/NULL cycle flows through the original circuit, while the next DATA/NULL cycle flows through the duplicate circuit. The outputs of the two circuits are then multiplexed to form a single output stream.

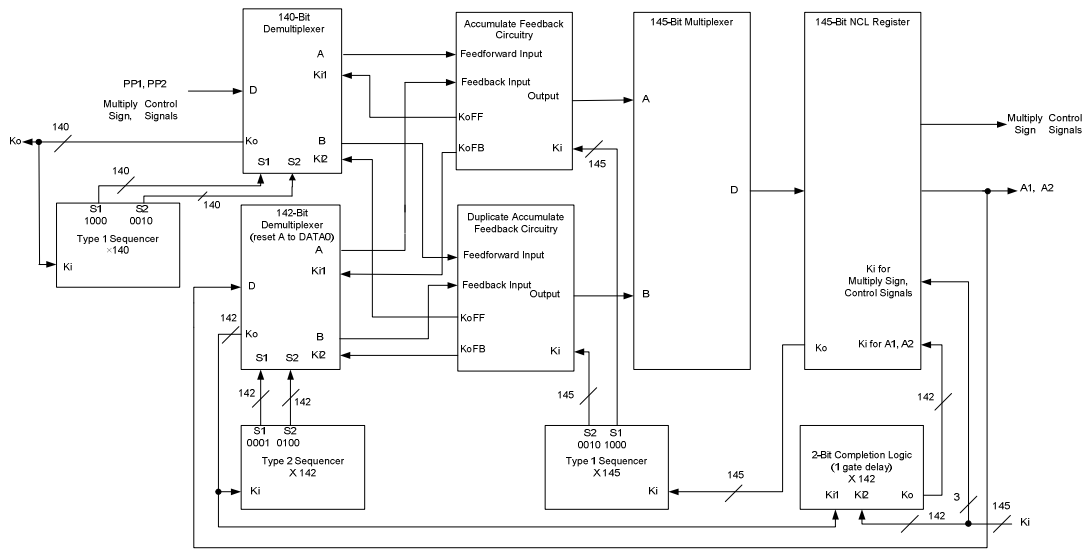


Figure 5. NULL Cycle Reduced accumulate feedback loop.

Bit-wise NCR, as shown in Fig. 5, was utilized such that each bit in the Demultiplexers and output registers of the original and duplicate circuits has its own corresponding Sequencer, which allows the feed-forward and feedback wavefronts to flow through the NCR architecture more independently, as explained in [3]. Additionally, a nonfunctional stage was included at the output to maximize performance, required to mitigate the effect of adjacent stage DATA propagation delay in determining throughput, as explained in [3, 5].

Since NCR is being applied to a feedback loop, the feedback Demultiplexer must be reset to DATA0, instead of NULL, to initialize the accumulator to zero upon reset. This requires the corresponding Sequencers to be modified to output a cycle of 0001 and 0100 for  $S1$  and  $S2$ , respectively, instead of the normal sequence of 1000 and 0010, because after reset this Demultiplexer will request a NULL wavefront, such that the first state for  $S1S2$  should be 00 to allow the NULL wavefront to pass. Since the first DATA flows to output  $A$ , the second DATA should flow to output  $B$ ; hence, the second state for  $S1S2$  should be 01. In Fig. 5, the original Sequencer is denoted as Type 1, and its modified version is denoted as Type 2. The Type 2 Sequencer is implemented by changing the reset state of the TH33 gates in the Type 1 Sequencer [3], since the Type 2 sequencer is just one cycle ahead of its Type 1 counterpart. The  $A1$  and  $A2$  outputs of the nonfunctional register are both fed back to the input of 142-Bit Demultiplexer and passed to the 70-bit RCA; hence, their  $Ki$  request signals must be generated using 2-input completion logic to combine the  $Ko$  request signals from the Demultiplexer and RCA.

### B. Feed-Forward Ripple Carry Adder

The 70-bit RCA is pipelined into 35 stages of 2-bit RCAs, utilizing bit-wise completion. The worst-case delay of a 2-bit RCA is 3 gate delays; and the bit-wise completion requires 1 gate delay. Therefore,  $T_{DD}$  is 8 gate delays and simulated to

be 4.9 ns. Coarser pipelining the RCA into 22 stages of 3-bit RCAs plus 2 stages of 2-bit RCAs resulted in a  $T_{DD}$  of 6.4 ns, which would have decreased overall MAC throughput.

### C. Overflow Feedback

Overflow depends on the type of operation being performed (i.e., signed  $\times$  signed, signed  $\times$  unsigned, or unsigned  $\times$  unsigned multiplication and adding to or subtracting from the accumulator or multiplication only), as well as the most significant bit (MSB) of both the previous and the new accumulator value. Hence, overflow is calculated in a feedback loop following the RCA, where the MSB of the new accumulator value,  $A_{new}$ , is calculated. As shown in Fig. 6, the redesigned overflow feedback circuitry consists of a function that zeros the previous accumulator's sign bit if multiply only is selected (no overflow can occur in this case), and the Calculate Overflow function.

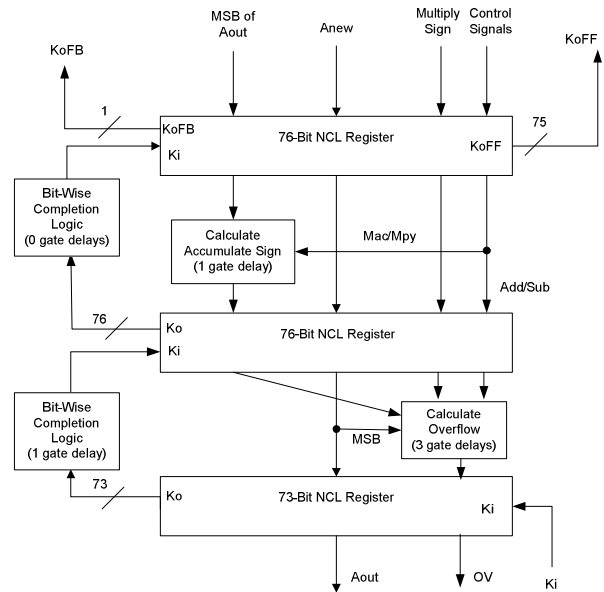


Figure 6. Redesigned overflow feedback circuitry.

The initial  $T_{DD}$  for this new overflow feedback loop was 10.0 ns; hence it required further optimization. Bit-wise completion [5] was utilized to both increase throughput and reduce area. Note that the completion delay of the first stage is 0 because the Calculate Accumulate Sign only produces 1 bit, such that its corresponding  $K_O$  is directly connected to the  $K_i$  signals of its inputs, which does not require any completion detection circuitry.

As shown in Fig. 7, bit-wise NCR was applied, similar to its application to the Accumulate Feedback Circuitry described in Section II.A, resulting in a  $T_{DD}$  of 5.0 ns. Since the Overflow Feedback Circuitry is the last stage in the MAC pipeline, a nonfunctional registration stage was not required; however, if this MAC is utilized as part of a larger design, where its output feeds into another circuit, adding the nonfunctional register may be necessary to maximize performance.

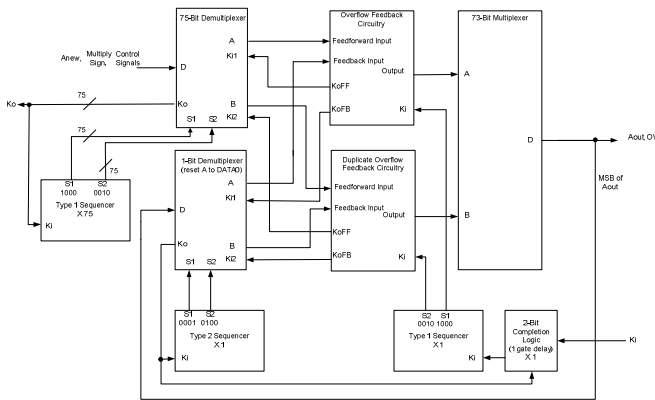


Figure 7. NULL Cycle Reduced overflow feedback loop.

### III. REDESIGNING THE MAC FEED-FORWARD PIPELINE

After optimizing the feedback circuitry, as detailed in Section II, the throughput of the feed-forward portion must be optimized to be at least as fast as the redesigned MAC feedback. Referring to Fig. 1, the feed-forward portion was optimally pipelined using the Gate-Level Pipelining (GLP) method [5], resulting in 7 stages with a  $T_{DD}$  of 14 gate delays, utilizing bit-wise completion, except for the  $2^5$  Complement and Shift PPs stage that used full-word completion because bit-wise completion did not increase throughput and required more gates. This decreased the feed-forward  $T_{DD}$  to 7.8 ns; hence, additional optimization was needed to reduce its  $T_{DD}$  to that of the redesigned feedback circuitry, 5.0 ns.

The first stage, consisting of Generate PPs and Calculate Multiply Sign, and the second last stage,  $2^5$  Complement and Shift PPs, were the throughput limiting stages, with a  $T_{DD}$  of 12 and 14 gate delays, respectively. The other stages, all consisting of CSAs, could have been further pipelined, utilizing bit-wise completion, to achieve a  $T_{DD}$  of 6 gate delays. Hence, NCR was applied to the two slow stages, and the CSAs were re-pipelined to the minimal  $T_{DD}$  of 6 gate delays. The resulting design consisted of an NCR PP generation first stage, followed by 8 stages of bit-wise pipelined CSAs, an NCR  $2^5$  Complement and Shift PPs stage, and one final bit-wise pipelined CSA stage, which resulted in a  $T_{DD}$  of 4.9 ns.

## IV. SIMULATION RESULTS

The optimized feed-forward pipeline and feedback circuitry were connected together and the overall MAC simulated with Mentor Graphics ModelSim, using the same 3.3V 0.5 $\mu$ m static NCL CMOS library and the same testbench as in [1], to compare the overall MAC speedup. The testbench operation is

$$A_{out} = \sum_{i=0}^N (X_i \times Y_i), \text{ where } X_i = X_0 + (2^{-21} \times i) \text{ and}$$

$Y_i = Y_0 + (2^{-11} \times i)$ , with  $N$  chosen to be 255. This allows for a variety of computations to be performed such that any unusually short or long operations will not significantly skew the average cycle time.  $X_0$  and  $Y_0$  were randomly selected as  $X_0 = A61C039Dh = -0.702270077076$  and  $Y_0 = F0046718h = -0.124865639955$ ; and (signed  $\times$  signed) multiplication was selected. As expected, the outputs were the same as the previous MAC; and the overall  $T_{DD}$  was 5.0 ns, equivalent to the  $T_{DD}$  of the slowest stage, a speedup of 1.72 compared to the previous  $T_{DD}$  of 8.6 ns [1]. However, this significant speedup comes at the expense of a 3.58X increase in gates, from 21,154 [1] to 75,664.

## V. CONCLUSION

In this paper the fastest  $72+32 \times 32$  asynchronous MAC in the literature [1] was redesigned by utilizing a new high-speed feedback architecture for the accumulate circuitry, resulting in a speedup of 1.72 at the cost of a 3.58X increase in area. This significant area increase was due in large part to the multiple uses of NCR. If this MAC did not include as many options (e.g., only performed  $2^5$  complement multiplication, and always added the result to the accumulator), the  $2^5$  Complement and Shift PPs and complex Overflow Feedback stages would be eliminated, both of which required NCR; hence, the overhead would be substantially reduced, and the throughput would be slightly increased as well. For future comparisons, this new MAC was also simulated using a 1.8V 0.18 $\mu$ m static NCL CMOS library [6], resulting in a  $T_{DD}$  of 3.8 ns, making this the fastest  $72+32 \times 32$  asynchronous MAC in the literature.

## REFERENCES

- [1] S. C. Smith, "Development of a Large Word-Width High-Speed Asynchronous Multiply and Accumulate Unit," *Elsevier's Integration, The VLSI Journal*, Vol. 39/1, pp. 12-28, September 2005.
- [2] K. M. Fant, S. A. Brandt, "NULL Convention Logic: A Complete and Consistent Logic for Asynchronous Digital Circuit Synthesis," *International Conference on Application Specific Systems, Architectures, and Processors*, pp. 261-273, 1996.
- [3] S. C. Smith, "Speedup of NULL Convention Digital Circuits Using NULL Cycle Reduction," *Elsevier's Journal of Systems Architecture*, Vol. 52/7, pp. 411-422, July 2006.
- [4] S. C. Smith, R. F. DeMara, J. S. Yuan, M. Hagedorn, and D. Ferguson, "NULL Convention Multiply and Accumulate Unit with Conditional Rounding, Scaling, and Saturation," *Elsevier's Journal of Systems Architecture*, Vol. 47/12, pp. 977-998, June 2002.
- [5] S. C. Smith, R. F. DeMara, J. S. Yuan, M. Hagedorn, and D. Ferguson, "Delay-Insensitive Gate-Level Pipelining," *Elsevier's Integration, the VLSI Journal*, Vol. 30/2, pp. 103-131, October 2001.
- [6] <http://comp.uark.edu/~smithsco/VHDL.html>