THE PERFORMABILITY ANALYSIS FOR DIFFERENT KIND OF ADDERS

Daniela Elena Popescu
Computer Science Department, University of Oradea, Str.Armatei Române nr.5, Oradea, România
depopescu@uoradea.ro

Abstract: The paper is the result of the author's activities concerning the testing and the design for testability for Computer arithmetic systems. Integer addition is the simplest operation and the most important. Despite the simplicity of addition, there isn't a best way to perform high-speed addition. There are three techniques that are in current use: carry lookahead, carry skip and carry select. The paper focuses on a comparative estimations of adders through the angle of the performability indicator.

1. INTRODUCTION

In order to compare designs, we have developed a formula for the test:performance ratio of each adder design. These results are intended to be used only comparatively – as indicators of relative merit, not as absolute measures. Performance is calculated as the inverse of the number of gate delays required to perform the operation at hand; so small delays mean high performance (A single gate delay will be denoted be τ). Test is calculated in terms of the number of tests implied for complete testing of these structures.

The measures used are not necessarily the most accurate. For performance, a better estimate of operational times might be obtained by assuming that τ is also sufficient for the operation of a two level gating structure – in some circuits, the delay through a two-level AND-OR structure is about the same as that through one level of gating – or by taking into account the fact that the delays through different types of gates vary according to the technology, the fan-in, and so forth. In any case, gate-delay may not necessarily be the best indicator of performance: for example, signal transmission through lengthy interconnections can have a great effect on timing. Nevertheless, all these "more accurate" measures are dependent on the technology used for the realization and are therefore difficult to generalize.

2. THE PERFORMANCE

In (Popescu 1998a) is determined the operational time for different kinds of adders. The results are summarized in Table 1 that realizes the comparing under this aspect of the performance of arbitrary size adders.

Table 1

Adder	Time (τ)
CLA pur	4
CSlA	6
PyA	$2\log_2 n+1$
CdSumA	$2\log_2 n+2$
SRCLA (m=4, M=4)	$\frac{n}{8}+6$
(m=2, M=8)	$\frac{n}{8}+6$
(m=4, M=8)	$\frac{n}{16}+6$
CCA (media)	$2\log_2 n+4$
SBCLA	$6\cdot\sqrt[3]{n}-2$
BCLA	$4\sqrt{n}-2$
CSkA pe un nivel (optimal)	$\approx 4\sqrt{n}$
PyCLA	$2\cdot\log_2 n$
RCLA (m=4)	$\frac{n}{2}+1$
(m=8)	$\frac{n}{4}+1$
RCA	$2\cdot n-1$

The results of this table are just simple orientated due to the above mentioned aspects.
For special cases, it may happen that the theoretical optimum values for some parameters, such as block size, may not be practical; for example the value may not be integral. For these reasons, and because the formulae are not easily compared at a glance, we shall instead look at concrete values already computed in (Popescu 1998a).
There are some general observations that must be remembered in the Table 1 data interpretations:
The carry ripple adder (RCA) appears to be an extremely bad design. It does, however, have a very simple and regular structure, which makes it attractive for a VLSI implementation. Moreover,

when combined with a Manchester carry-chain, the resulting raw performance ia sufficient good to make implementation worthwhile.

Table 2

Adder	Number of tests
CCA (the average)	8
RCA	8
The pure CLA	$\frac{(m+1)\cdot(m+2)}{2}+1$
RCLA (m)	$\frac{(m+1)\cdot(m+2)}{2}+1$
CSlA (m)	$\frac{(m+1)\cdot(m+2)}{2}+1$
BCLA (m)	$\frac{(m+1)\cdot(m+2)}{2}+7$
One level CSkA	$(10+2\cdot m)$
Multilevel CSkA	$(11+2\cdot m+2\cdot M)$
PyCLA	$8+14\cdot\log_2\frac{n}{2}$
SRCLA (m, M)	$\frac{M\cdot(M+3)}{2}+\frac{m\cdot(m+3)}{2}$
SBCLA (m, M)	$\frac{M\cdot(M+3)}{2}+\frac{m\cdot(m+3)}{2}+8$

- The carry-skip Adder (CSkA) (Omondi 1996) which are the next cheapest adders, give their best performance when non-uniform block sizes are used; however, varying block sizes means that some regularity is lost, and this may make the adder less suitable for realization in some technologies or with of-the-shelf components. If, however, uniform-size blocks are used in conjunction with a Manchester carry-chain, then the resulting structure has both regularity and good performance. This has been the basis of a number of VLSI realizations.
- The pure carry-lookahead adder is theoretically the fastest adder, but it is expensive, and the required fan-in and fan-out make it impractical for any but small adders. For large adders, the bit-stages must be grouped into blocks, and possibly superblocks, that permits the carry-lookahead to be combined with some other technique. The RCLA, which combines carry rippling with carry-lookahead, is a simple version of such an adder (with one level of blocking) and gives reasonable performance for small and medium-sized adders. It has a regular structure, in so far as it uses units (blocks) of the same structure, and the interconnectivity between blocks is not very high. Adding a superblock level to the RCLA yields the SRCLA, which is an extremely good adder for large words. But, unlike its one-level counterpart, the SRCLA requires at least two different types of basic unit and a much higher number of interconnections. The BCLA, SBCLA and ISBCLA are three related designs in the block-lookahead idea. They have operational times that are worse that those of the RCLA and SRCLA, but they cost less and may, therefore, be useful where a medium-sized or large adder is to be implemented and cost is an issue. Overall, the carry-lookahead adders have been the most popular adders for high performance arithmetic-and-logic units.
- On the basis of the above table, the pyramidal adder appears to be an excellent adder: both its operational time and cost are relatively low. It does, however, have fan-in and fan-out requirements that, in its pure form, are likely to render it impractical for any but smalls adders. Combining pyramidal addition with some other technique can yield a more practical adder with good performance and cost but without the fan-in and fan-out problems. The adder is also highly amenable to pipelining.
- The conditional-sum and carry select adders have good performance across the whole range of adder widths but at high cost and, more importantly, with high fan-in (carry-select) and high fan-out (both adders) requirements. As with other designs that suffer from problems, the use of another technique is helpful. On the other side, the both types of summons are pretender to *pipelining*.
- The carry-completion adder has a good average operational time, high regularity, and relatively low cost. Nevertheless, its worst-case performance time on the operands makes it unsuitable for many implementations. One can, however, envisage a carry-completion adder using Manchester carry-chain to improve the raw performance. Given the renewed interest in asynchronous machines, this adder may yet find a niche in the space of useful adders.

3. NUMBER OF TESTS

On the bases of some calculation expression for the number of tests for complete testing of the singular stuck-at faults (Abramovici 1996) (Hidao 1990) for the summing structures that were analyzed (in the context of using the C-testability property) and determined in (Popescu 1999) (Popescu 1997a) (Popescu 1997b) (Popescu 1998a) (Popescu 1998c) (Popescu 1998d) (Popescu 1997c) (Popescu 1998e), we complete the dates from Table 2 which realizes the comparison under this aspect of adders of arbitrary dimension.

Going on, we must note some general observations resulted from the formulae of Table 2:

- The RCA and CCA structures are the most easily tested.

- The testing of the RCLA and CS1A summing structures is made with the same number of tests, which is less than that involved by the BCLA testing and by the testing of adders organized in superblocks
- The complete testing of the single stuck at faults for the CSkA implies a number of tests higher than that necessary for testing the CLA organized in blocks, but less that that necessary for testing the CLA organized in supeblocks.

Table 3

Adder	Performability
CCA (media)	$8 \cdot (2 \cdot n - 1)$
RCA	$16 \cdot (\log_2 n + 2)$
CLA pur	$4 \cdot \left(\frac{(m+1) \cdot (m+2)}{2} + 1 \right)$
RCLA (m)	$2 \cdot \left(\frac{n}{m} + 1 \right) \cdot \left(\frac{(m+1) \cdot (m+2)}{2} + 1 \right)$
CSlA (m)	$\frac{(m+1) \cdot (m+2)}{2} + 1$
BCLA (m)	$2 \cdot \left(m + \frac{n}{m} - 1 \right) \cdot \left(\frac{(m+1) \cdot (m+2)}{2} + 7 \right)$
CSkA pe un nivel	$\left(2 \cdot \frac{n}{m} + 4 \cdot m - 5 \right) \cdot (10 + 2 \cdot m)$
CSkA multinivel	$\left(4 \cdot m + 4 \cdot M + \frac{2 \cdot n}{m \cdot M} - 12 \right) \cdot (11 + 2 \cdot m + 2 \cdot M)$
PyCLA	$2 \cdot \log_2 n \cdot \left(8 + 14 \cdot \log_2 \frac{n}{2} \right)$
SRCLA (m, M)	$2 \cdot \left(\frac{n}{m \cdot M} + 3 \right) \cdot \left(\frac{M \cdot (M+3)}{2} + \frac{m \cdot (m+3)}{2} \right)$
ISBCLA (m, M)	$2 \cdot \left(\frac{n}{m \cdot M} + m - 1 \right) \cdot \left(\frac{M \cdot (M+3)}{2} + \frac{m \cdot (m+3)}{2} + 8 \right)$
SBCLA (m, M)	$2 \cdot \left(\frac{n}{m \cdot M} + M + m - 1 \right) \cdot \left(\frac{M \cdot (M+3)}{2} + \frac{m \cdot (m+3)}{2} + 8 \right)$

It is important to note that for some kind of needs, the theoretical best value of some parameters (the blocks size, for example) it could be happened not to be touched since they don't result such as integral value. Because of this it is imposed as a criterion of comparison the *n* value used in the practical implementation of these summing structures.

On the other hand, the data from the table must be considered also in the context of technological implementation of the summing structures; there are some problems related with the fan-out, the fan-in, the integration size etc. These problems were not taken account.

4. THE PERFORMABILITY

With a view of some universal comparisons of different summing structures (at the producers level) we define the ratio of performability as being given by the result of inverse multiplication between operation performance ratio and the number of necessary tests for a complete testing for detection of the single stuck-0 (or 1)at faults.

$$Performability = \frac{1}{(performance) \cdot (no.tests)}$$

Table 3 shows the calculated formulae for performability for different kinds of adders. These results can be used as a global indicator for characterizing the adders' testability and performance.

5. CONCLUSIONS

On the base of the functional analyze of summing structures we elaborate in this paper the calculus formulae for their performance in operation; the formulae were determined in terms of the number of traversed gates by the activated signals of the summing background process.

On the other hand, by using of the spaced repeated properties of the summing structures and applying to these structures the C-testability concept – realizing when is necessary a reconfiguration of the adder in

order to obtain the C-testability – in addition with that of the minimal testing to the level of repeatable cells, we succeed by a meticulousness analyze of the structures, to obtain the minimal test sets for complete testing them in order to detect the single stuck at 0 (or 1) faults.

For a global characterization in the view of dependability of summing structures, we introduced in this paper a new indicator: the performability ratio for which we gave in Table 3 the calculations to the level of each adder structure.

REFERENCES

Abramovici, M., Breuer, A.D.Friedman, 1996, M.A.:Digital Systems And Testable Design, Computer Science Press,

Hidao Fujiwara, 1990, Logic Testing And Design For Testability, Computer Systems Series The MIT Press

Popescu, D., 1998a, "Contribu□ii la testarea asistat□ de calculator □i sinteza testabil□ a echipamente-lor de calcul cu fiabilitate sporit□", tez□ de doctorat

Popescu, D., Popescu, C., 1999, The C-testability Analyze of the Carry Skip Adder (CSkA) Proceedings of Oradea EMES' 99, May 27-39, Oradea, Romania

Popescu, D., Popescu, C., 1997a, An algorithm for determining the minimal test set for the Carry-Lookahead Adder (CLA) - The University of Oradea Annals, 1997, the Electrotechnical fascicle, Section D, pp.87-92

Popescu, D., Popescu, C., 1997b, The C-testability Analyze of the Block Carry-Lookahead Adder (BCLA) - The University of Oradea Annals, the Electrotechnical fascicle, Section D, pp.81-86

Popescu, D., Popescu, C., 1998b, The C-testability Analyze of the Ripple-Block Carry-Lookahead Adder (RCLA) - International Symposium on Systems Theory Robotics, Computers & Process Informatics, Craiova, Section Computer Science & Engineering

Popescu, D, 1998c, The C-testability analyze for the Carry Completion Adder (CCA)- A&Q98, International Conference and Quality Control, QUALITY, Design and Development, vol. Q, Cluj, pp. 117-121

Popescu, D., Popescu, C., 1998d, The C-testability analyze for an hybrid Pyramidal Carry-Lookahead Adder - FEI International Conference Electronic Cpmputers & Informatics'98, 8-9 Oktober, Slovakia

Popescu, D., Popescu, C., 1997c, The analyze on the test pattern generated by a multiple input signature register - The University of Oradea Annals, the Electrotechnical fascicle, Section D, pp.85-89

Popescu, D., Popescu, C., 1998e, The pyramidal Carry Lookahead Adder (PyCLA) - Transactions on Automatic and Computer Science, Special Issue Dedicated on Technical Informatics CONTI'98, October 29-30, Timi□oara, pp. 151-159

Omondi A.R., 1996, Computer Arithmetic Systems Algorithms, Architecture and Implementation, Prentice Hall International.