High Speed Gate Level Synchronous Full Adder Designs

PADMANABHAN BALASUBRAMANIAN[§] and NIKOS E. MASTORAKIS[¶]
[§]School of Computer Science,
The University of Manchester,
Oxford Road, Manchester M13 9PL,
UNITED KINGDOM.
padmanab@cs.man.ac.uk
[¶]Department of Computer Science,
Military Institutions of University Education, Hellenic Naval Academy,
Piraeus 18539,
GREECE.
mastor@hna.gr

Abstract: - Addition forms the basis of digital computer systems. Three novel gate level full adder designs, based on the elements of a standard cell library are presented in this work: one design involving **XN**OR and **m**ultiplexer gates (XNM), another design utilizing **XN**OR, **A**ND, **I**nverter, **m**ultiplexer and **c**omplex gates (XNAIMC) and the third design incorporating **X**OR, **A**ND and **c**omplex gates (XAC). Comparisons have been performed with many other existing gate level full adder realizations. Based on extensive simulations with a 32-bit carry-ripple adder implementation; targeting three process, voltage and temperature (PVT) corners of the high speed (low V_t) 65nm STMicroelectronics CMOS process, it was found that the XAC based full adder is found to be delay efficient compared to all its gate level counterparts, even in comparison with the full adder cell available in the library. The XNM based full adder is found to be area efficient, while the XNAIMC based full adder offers a slight compromise with respect to speed and area over the other two proposed adders.

Key-Words: - Combinational logic, Full adder, High performance, Standard cells, and Deep submicron design.

1 Introduction

A binary full adder is often found in the critical path of microprocessor and digital signal processor data paths, as they are fundamental to almost all arithmetic operations. It is the core module used for many essential operations like multiplication, division and addresses computation for cache or memory accesses and is usually present in the arithmetic logic unit and floating point units. Hence, their speed optimization carries significant potential for high performance applications.

A 1-bit full adder module basically comprises of three input bits (say, *a*, *b* and *cin*) and produces two outputs (say, *sum* and *cout*), where '*sum*' refers to the summation of the two input bits, '*a*' and '*b*', and *cin* is the carry input to this stage from a preceding stage. The overflow carry output from this stage is labeled as '*cout*'.

Many efficient full custom transistor level solutions for full adder functionality have been proposed in the literature [1] – [10], optimizing any or all of the design metrics viz. speed, power and area. In this paper, our primary focus is on realizing high performance full adder functionality using readily available off-the-shelf components of a standard cell library [11]. Hence, our approach is semi-custom rather than being full custom. This article primarily focuses on the novel design of full adders at the logic level and also highlights a comparison with many other existing gate level solutions, from performance and area perspectives. The inferences from this work may be used for further improvement of full adder designs at the transistor level. Apart from this, this article is also intended to provide pedagogical value addition.

The remaining part of this paper is organized as follows. Section 2 describes the various existing gate level realizations of a 1-bit binary full adder. The three newly proposed full adder designs are mentioned in section 3. Section 4 gives details about the simulation mechanism and results obtained. Finally, we conclude in the next section.

2 Existing Gate Level Adder Designs

The truth table of a binary 1-bit full adder is given in Table 1 and the fundamental equations governing

the full adder's sum and carry outputs are given below. The sum output corresponds to the exclusive-OR operation performed on all the input bits, while the carry output basically conforms to majority logic, in that, if any of the two inputs have a similar logic state, then the carry output is assigned the same state.

$$sum = a \oplus b \oplus cin \quad (1)$$

$$cout = a \bullet b + b \bullet cin + a \bullet cin \quad (2)$$

Table 1. Truth table of a full adder

Inputs			Outputs	
a	***b***	***cin***	***sum***	***cout***
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

2.1 Full adder based on minimum sum-of-products form

A minimum sum-of-products (MSOP) form of a Boolean function can be obtained using a standard two-level logic minimizer: Espresso [12]. Though the logic realization corresponds to an AND-OR two-level logic format, inverting buffers required for the primary inputs have to be separately accounted for. A 1-bit full adder, based on MSOP forms for sum and carry outputs, is shown in fig. 1.

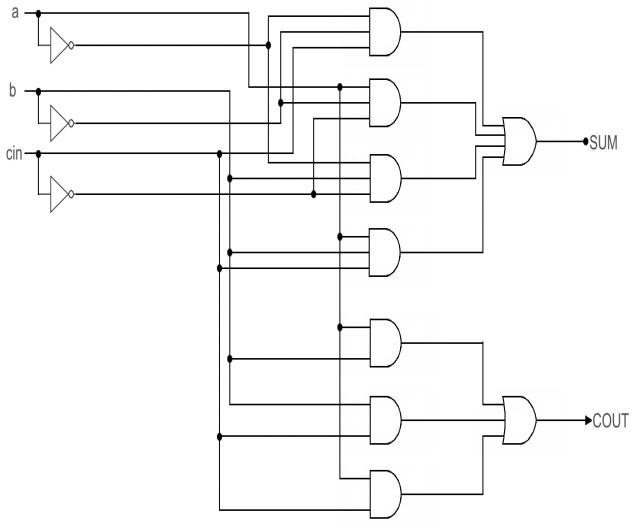

Fig. 1. Full adder realization corresponding to conventional AND-OR logic based on two-level logic minimization, with input inverters

2.2 Full adder implementation using two half adder modules

A traditional full adder implementation based on two half adder modules [13] is shown in fig. 2.

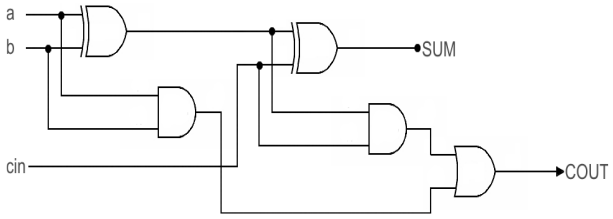

Fig. 2. 1-bit full adder based on two half adders

2.3 Full adder design incorporating logic sharing

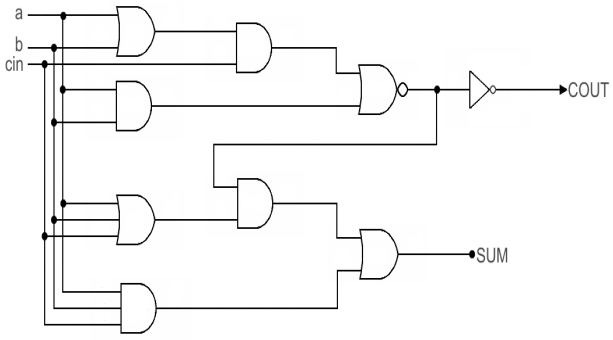

Fig. 3. Full adder design with logic sharing

The full adder design incorporating logic sharing, as described in [14] is depicted by fig. 3. A slight modification is done, in that the NOR gate followed by the NOT gate to realize the sum output in the original circuit has been replaced by an OR gate.

2.3 Full adder realization predominantly using 2:1 MUXes

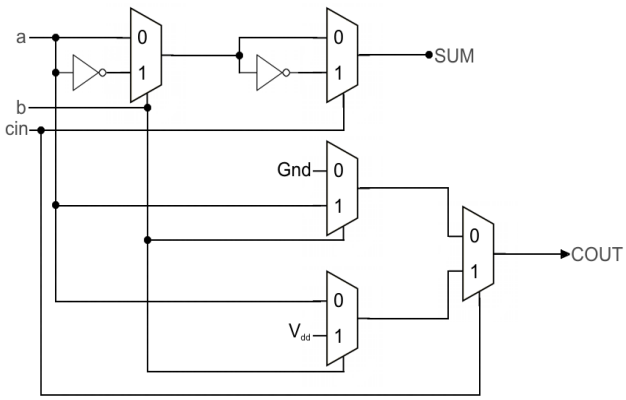

Fig. 4. Full adder realization predominantly based on 2:1 MUXes

A binary full adder realization mainly employing 2:1 MUXes is highlighted in [4]. Due to the non-availability of appropriate inverted input MUX cells to directly implement the XOR functionality, inverters were used in addition to the standard MUX cells to facilitate XOR logic realization. Excepting this, the rest of the full adder logic appears similar to that of [4], as shown in fig. 4.

2.5 Full adder designs using XNOR and XOR gates for sum logic

A full adder design employing two stages of XNOR gates for the sum logic [7] - [9] is shown in fig. 5, while that employing two successive stages of XOR gates for the sum logic is depicted by fig. 6.

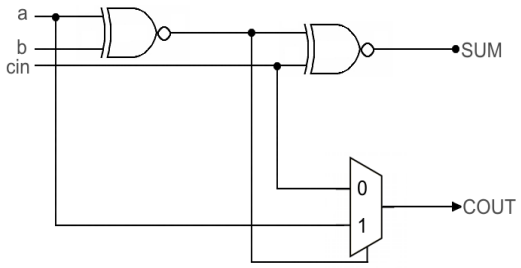

Fig. 5. Full adder using XNOR gates and a MUX

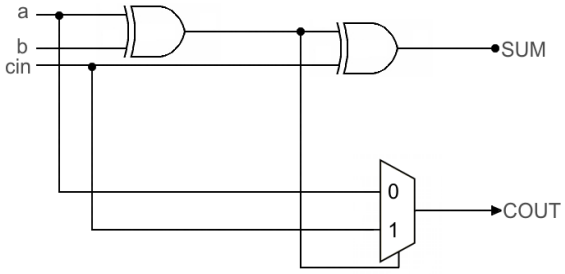

Fig. 6. Full adder using XOR gates and a MUX

2.6 Centralized full adder design

The centralized full adder design mentioned in [8] and [9] is portrayed by fig. 7.

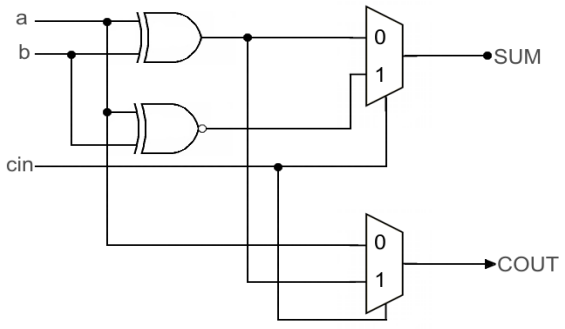

Fig. 7. Gate level centralized full adder design

2.7 Shannon's theorem based full adder

The design of a gate level binary full adder [10], based on Shannon's decomposition theorem and multiplexing control input technique for the carry and sum outputs, is given in fig. 8.

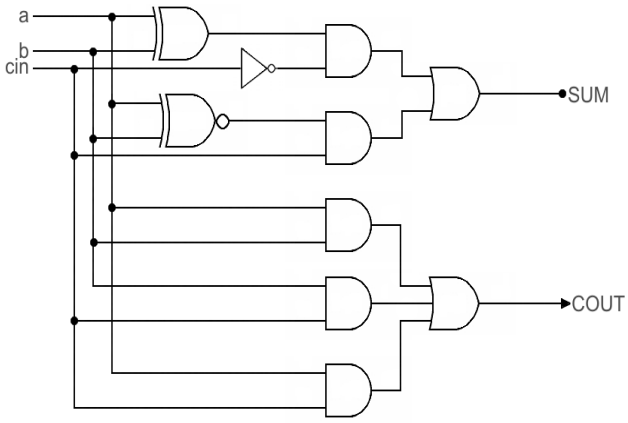

Fig. 8. Gate level full adder design by employing Shannon's theorem and multiplexing control inputs

2.8 Library's full adder cell

The internal details of the full adder cell, which forms a part of the commercial library [11], could not be commented upon in this article and so only the block schematic of the same is given below in fig. 9. The inputs and outputs are listed therein.

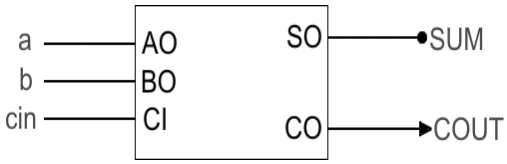

Fig. 9. Black box model of the library's full adder

3 Proposed full adder designs

Three full adder designs are proposed in this work. The first design employs only XNOR gates and 2:1 MUXes. However, one of the two non-inverting 2:1 MUXes has an inverted input. This design is portrayed by fig. 10 and shall be referred to as XNM based full adder. The second design employs XNOR, AND, NOT, inverted input 2:1 MUX and complex gates (XNAIMC based full adder), as shown in fig. 11. The complex gate used is the AO12 cell, which performs the function, $f = xy + z$, where 'x' and 'y' are the inputs to the AND logic and 'z' is the separate input to the OR logic. The last adder design is a refinement of the full adder design mentioned in section 2.2, in that, the AND gate of the second half adder module and the OR gate

associated with the carry output are replaced by a single complex gate, namely AO12 cell. This design, referred to as XAC based full adder, is shown in fig. 12.

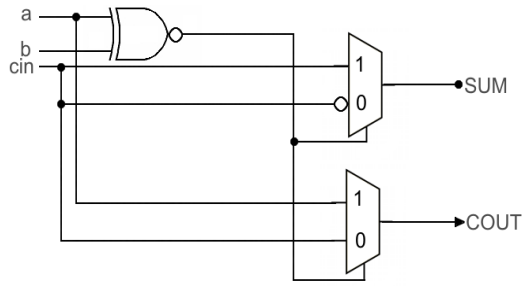

Fig. 10. Proposed XNM based full adder

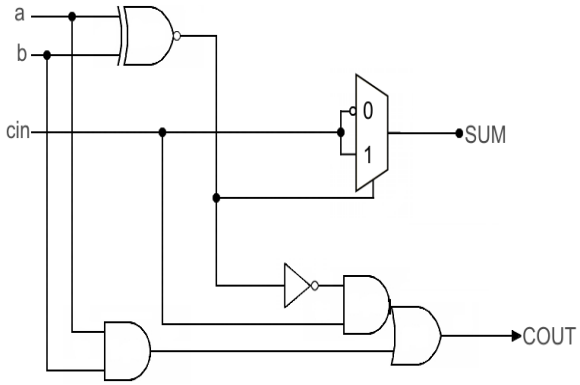

Fig. 11. Proposed XNAIMC based full adder

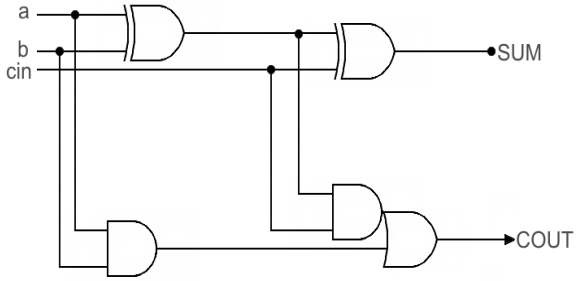

Fig. 12. Proposed XAC based full adder

Amongst all the other gate level full adders, the XAC based full adder realization was found to be the fastest. This is substantiated by the maximum operating speed values, highlighted in figures 13, 14 and 15. The expressions for the *sum* and *carry* outputs of the XAC based full adder are as follows. The Boolean product term, $(a \oplus b)\bullet cin$ in (4), has been implemented by the complex gate, AO12 cell.

$$sum = (a \oplus b) \oplus cin \qquad (3)$$

$$cout = a\bullet b + (a \oplus b)\bullet cin \qquad (4)$$

4 Simulation Method and Discussion of Results

The various 1-bit full adder realizations mentioned previously, were all described in cell level Verilog HDL, so that the physical implementation would be in exact conformity with the logical description. They were then instantiated to implement a 32-bit RCA, consisting of a series cascade of 32 full adder stages. The simulations were performed targeting a nominal case (typical case), best case and worst case PVT corner of the 65nm STMicroelectronics bulk CMOS process [11]. The simulation results purely reflect the performance and area metrics of the combinatorial adder logic and do not consider any sequential components. This sets the tone for a fair comparison of various full adder designs. The library has been inherently optimized for low power applications. The library corresponding to a supply voltage of 1.10V with low V_t was chosen, for which all the different process corner specifications exist. Static timing analysis and cells area estimation were done within the Synopsys PrimeTime environment. All the adder's inputs have the driving strength of the minimum sized inverter in the library, while their outputs possess fanout-of-4 (FO-4) drive strength. Appropriate wire loads were selected automatically for timing evaluation purpose.

Tabular columns 2, 3 and 4 depict the performance of various full adder modules for 32-bit addition, based on a ripple carry adder (RCA) topology. Table 5 reports the area measure for different 32-bit RCAs. The percentage figures mentioned within brackets highlight the percentage increase in the design metric for the other adders in comparison with the optimal adder. For e.g. in Tables 2, 3 and 4, we find that the proposed XAC based full adder features the least critical path (or least data path) delay amongst all other adders and the percentage figures for all other adders indicate their relative delay degradation compared to the proposed XAC based full adder for 32-bit addition.

From Tables 2, 3 and 4, it is evident that the full adder present in the library suffers delay degradation over the proposed XAC based full adder, on an average, across all the three process corners by 36.9%. However, the commercial library's full adder cell was found to occupy the least area in comparison with all other adders, with the next best area-efficient adder suffering from at least 1.33× increase in cells area in comparison with it; the proposed XAC based full adder reporting 29.3% more area increase over it.

Table 2. Experimental results corresponding to a nominal case PVT corner for 32-bit addition

(V_{dd} = 1.10V, $T_{Junction}$ = 25°C)

Design style	**Critical path delay (ns)**
Minimized SOP based full adder design [12]	5.41 (82.77%)
Half adders based full adder design [13]	4.28 (44.59%)
Full adder design embodying logic sharing [14]	5.78 (95.27%)
MUXes based full adder design [4]	3.73 (26.01%)
XNOR-XNOR based full adder design [7] - [9]	3.46 (16.89%)
XOR-XOR based full adder design [7] - [9]	3.50 (18.24%)
Centralized full adder [8] [9]	3.75 (26.69%)
Shannon's theorem based full adder [10]	5.17 (74.66%)
Commercial library's full adder [11]	4.11 (38.85%)
Proposed design – XNM based full adder	3.59 (21.28%)
Proposed design – XNAIMC based full adder	3.12 (5.41%)
Proposed design – XAC based full adder	2.96

Table 3. Experimental results corresponding to a best case PVT corner for 32-bit addition

(V_{dd} = 1.10V, $T_{Junction}$ = -40°C)

Design style	**Critical path delay (ns)**
Minimized SOP based full adder design [12]	3.92 (84.04%)
Half adders based full adder design [13]	3.11 (46.01%)
Full adder design embodying logic sharing [14]	4.23 (98.59%)
MUXes based full adder design [4]	2.55 (19.72%)
XNOR-XNOR based full adder design [7] - [9]	2.44 (14.55%)
XOR-XOR based full adder design [7] - [9]	2.47 (15.96%)
Centralized full adder [8] [9]	2.56 (20.19%)
Shannon's theorem based full adder [10]	3.75 (76.06%)
Commercial library's full adder [11]	2.90 (36.15%)
Proposed design – XNM based full adder	2.54 (19.25%)
Proposed design – XNAIMC based full adder	2.26 (6.10%)
Proposed design – XAC based full adder	2.13

Table 4. Experimental results corresponding to a worst case PVT corner for 32-bit addition

(V_{dd} = 1.10V, $T_{Junction}$ = 150°C)

Design style	**Critical path delay (ns)**
Minimized SOP based full adder design [12]	7.76 (83.89%)
Half adders based full adder design [13]	6.16 (45.97%)
Full adder design embodying logic sharing [14]	8.25 (95.50%)
MUXes based full adder design [4]	5.22 (23.70%)
XNOR-XNOR based full adder design [7] - [9]	5.02 (18.96%)
XOR-XOR based full adder design [7] - [9]	5.07 (20.14%)
Centralized full adder [8] [9]	5.24 (24.17%)
Shannon's theorem based full adder [10]	7.43 (76.07%)
Commercial library's full adder [11]	5.72 (35.55%)
Proposed design – XNM based full adder	5.21 (23.46%)
Proposed design – XNAIMC based full adder	4.47 (5.92%)
Proposed design – XAC based full adder	4.22

Table 5. Area metric for a 32-bit RCA implementation based on various full adder modules

(Common area measure, as performances have alone been analyzed with respect to different corners)

Design style	**Cells area (μm^2)**
Minimized SOP based full adder design [12]	1148.16 (3.83×)
Half adders based full adder design [13]	532.48 (1.78×)
Full adder design embodying logic sharing [14]	798.72 (2.67×)
MUXes based full adder design [4]	765.44 (2.56×)
XNOR-XNOR based full adder design [7] - [9]	399.36 (1.33×)
XOR-XOR based full adder design [7] - [9]	399.36 (1.33×)
Centralized full adder [8] [9]	532.48 (1.78×)
Shannon's theorem based full adder [10]	948.48 (3.17×)
Commercial library's full adder [11]	299.52
Proposed design – XNM based full adder	399.36 (1.33×)
Proposed design – XNAIMC based full adder	515.84 (1.72×)
Proposed design – XAC based full adder	465.92 (1.72×)

Though the proposed XNM based full adder reports the least area measure among the proposed adders, in line with those corresponding to XNOR-XNOR and XOR-XOR based full adder designs, it comes at the expense of an increase in the delay metric over the proposed XAC based full adder by 21.3%, on an average, considering all the targeted PVT corners. The maximum data path delay of the proposed XNAIMC based adder is somewhat closer to the proposed XAC based adder, but it occupies 10.7% more area than the latter realization.

It might be interesting to note that the classical full adder (constructed using two half adder modules) reports increase in worst case delay over the proposed XAC based full adder by a whopping 45.5%, on an average, when considering all the process corners. In terms of the operating frequency, this translates to a similar ratio of speed advantage for the proposed XAC based adder over the conventional full adder design, with respect to all the PVT corners considered. This is in addition to the fact that the proposed XAC based full adder features 12.5% lesser area consumption than the classical full adder.

Figures 13, 14 and 15 highlight the maximum operating frequency of the various 32-bit RCAs, where the maximum operating speed is given by the reciprocal of the longest data path delay. It is clearly evident that the proposed XAC based full adder exhibits the highest operating frequency amongst all the other adders, including the other two proposed adders. Among all the other gate level adders (i.e. excepting the proposed adders), the XNOR-XNOR based full adder design is faster. Nevertheless, the proposed XAC based adder reports an improvement in speed over the XNOR-XNOR based adder by 16.9%, 14.6% and 18.9% across the typical, best and worst case library corners respectively. In comparison with the commercial library based full adder, the proposed XAC based adder is found to be speed efficient by 36.9%, on an average, across all the three process corners. The above observations clearly emphasize the role played by complex gates in effecting high speed solutions, whilst minimizing area occupancy.

5 Conclusions

Three novel gate level full adder designs have been presented in this work: XNM, XNAIMC and XAC based full adders. Amongst these, the XAC based full adder achieves the highest speed, while the XNM based full adder occupies the least area. The XNAIMC based adder is somewhat closer to the XAC based adder with respect to delay. It is worth noting that the proposed XAC based full adder is faster than even the full adder cell present in a commercial standard cell library.

Apart from advancing fundamental research in data path element designs, this research article encompasses considerable pedagogical value. It is anticipated that the inferences from this work are likely to facilitate better transistor level solutions for full adder functionality compared to those which currently exist.

References:

[1] N. Zhuang and H. Wu, A New Design Of The CMOS Full Adder, *IEEE Journal of Solid-State Circuits*, Vol. 27, No. 5, May 1992, pp. 840-844.

[2] N.H.E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design – A Systems Perspective*, 2nd Edition, Addison-Wesley Publishing, MA, USA, 1993.

[3] R. Shalem, E. John and L.K. John, A Novel Low-Power Energy Recovery Full Adder Cell, *Proc. ACM Great Lakes Symposium on VLSI*, pp. 380-383, 1999.

[4] M. Margala, Low-Voltage Adders For Power-Efficient Arithmetic Circuits, *Microelectronics Journal*, Vol. 30, No. 12, December 1999, pp. 1241-1247.

[5] A.M. Shams, T.K. Darwish and M.A. Bayoumi, Performance Analysis Of Low-Power 1-Bit CMOS Full Adder Cells, *IEEE Trans. on VLSI Systems*, Vol. 10, No. 1, February 2002, pp. 20-29.

[6] M. Zhang, J. Gu, C.H. Chang, A Novel Hybrid Pass Logic With Static CMOS Output Drive Full Adder Cell, *Proc. IEEE Intl. Symposium on Circuits and Systems*, pp. 317-320, 2003.

[7] Y. Jiang, A. Al-Sheraidah, Y. Wang, E. Sha and J.-G. Chung, a Novel Multiplexer-Based Low-Power Full Adder, *IEEE Trans. on Circuits and Systems – II: Express Briefs*, Vol. 51, No. 7, July 2004, pp. 345-348.

[8] S. Goel, S. Gollamudi, A. Kumar and M. Bayoumi, On The Design Of Low-Energy Hybrid CMOS 1-Bit Full Adder Cells, *Proc. 47th IEEE Intl. Midwest Symposium on Circuits and Systems*, vol. II, pp. 209-212, 2004.

[9] S. Goel, A. Kumar and M.A. Bayoumi, Design of Robust, Energy-Efficient Full Adders for Deep Submicrometer Design Using Hybrid-

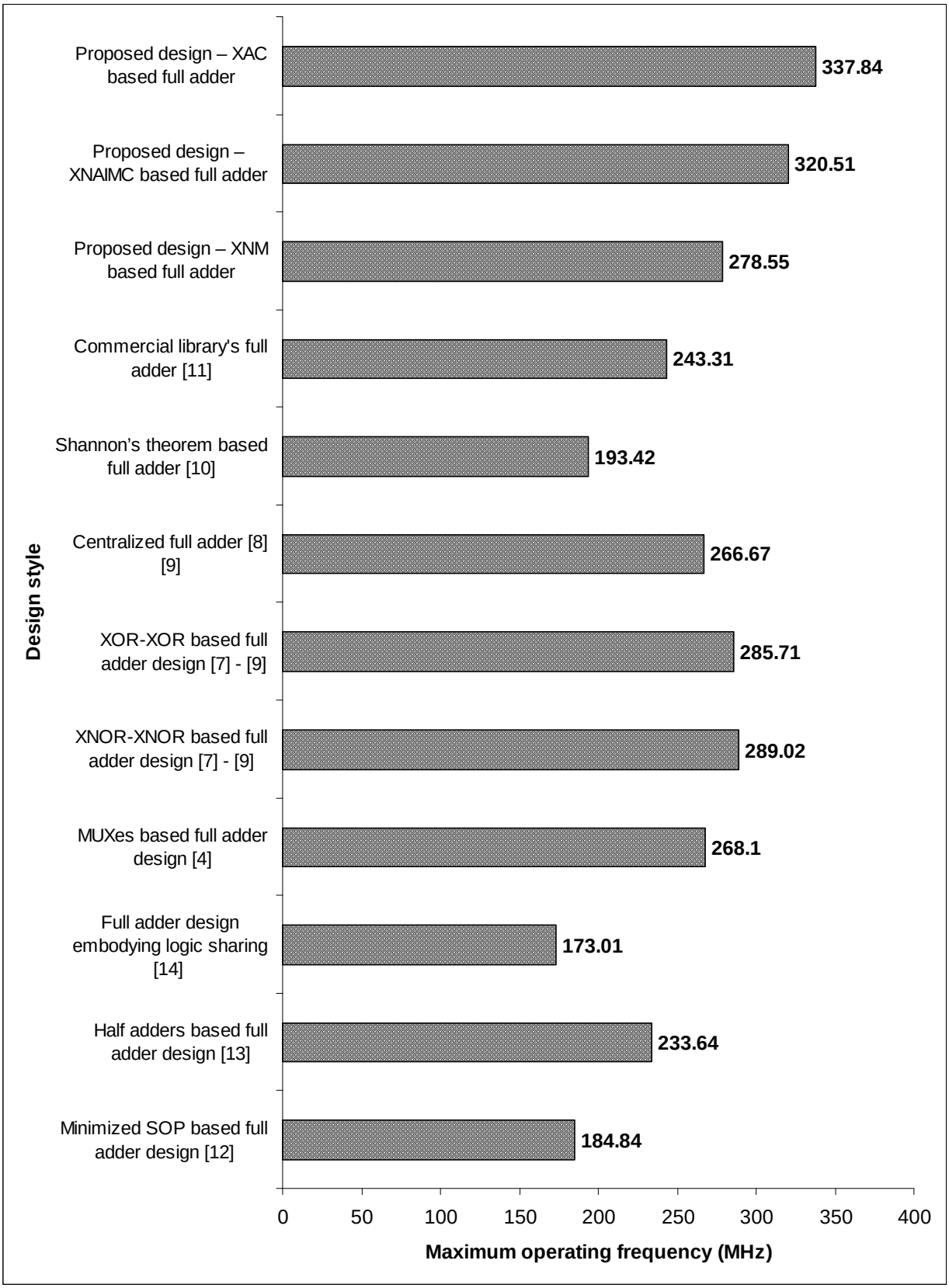

Fig. 13. Speed of operation of a 32-bit RCA, realized using different full adder modules for a nominal (typical) case library specification, with FO-4 drive strength

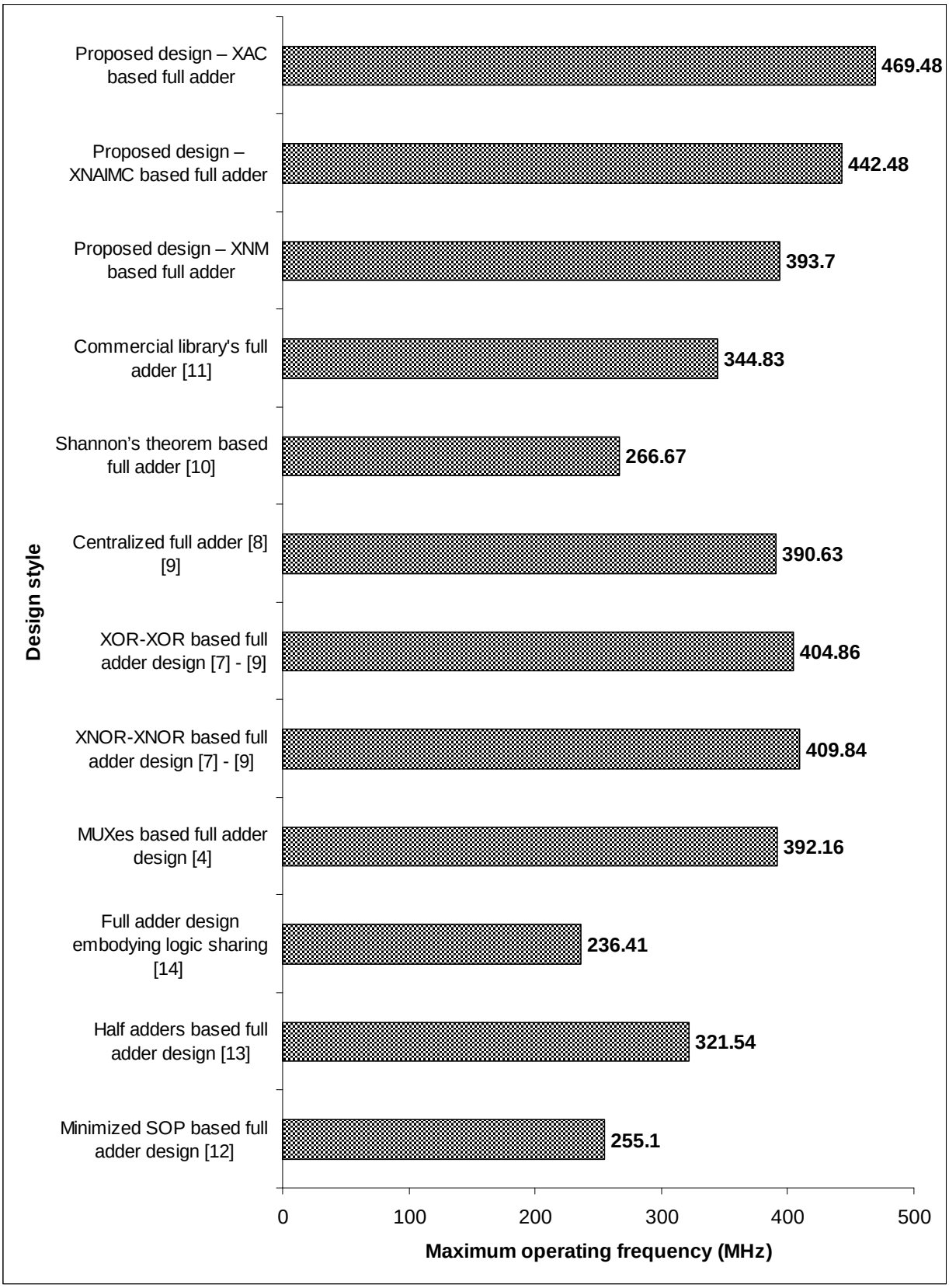

Fig. 14. Speed of operation of a 32-bit RCA, constructed using different full adder modules for a best case library specification, with FO-4 drive strength

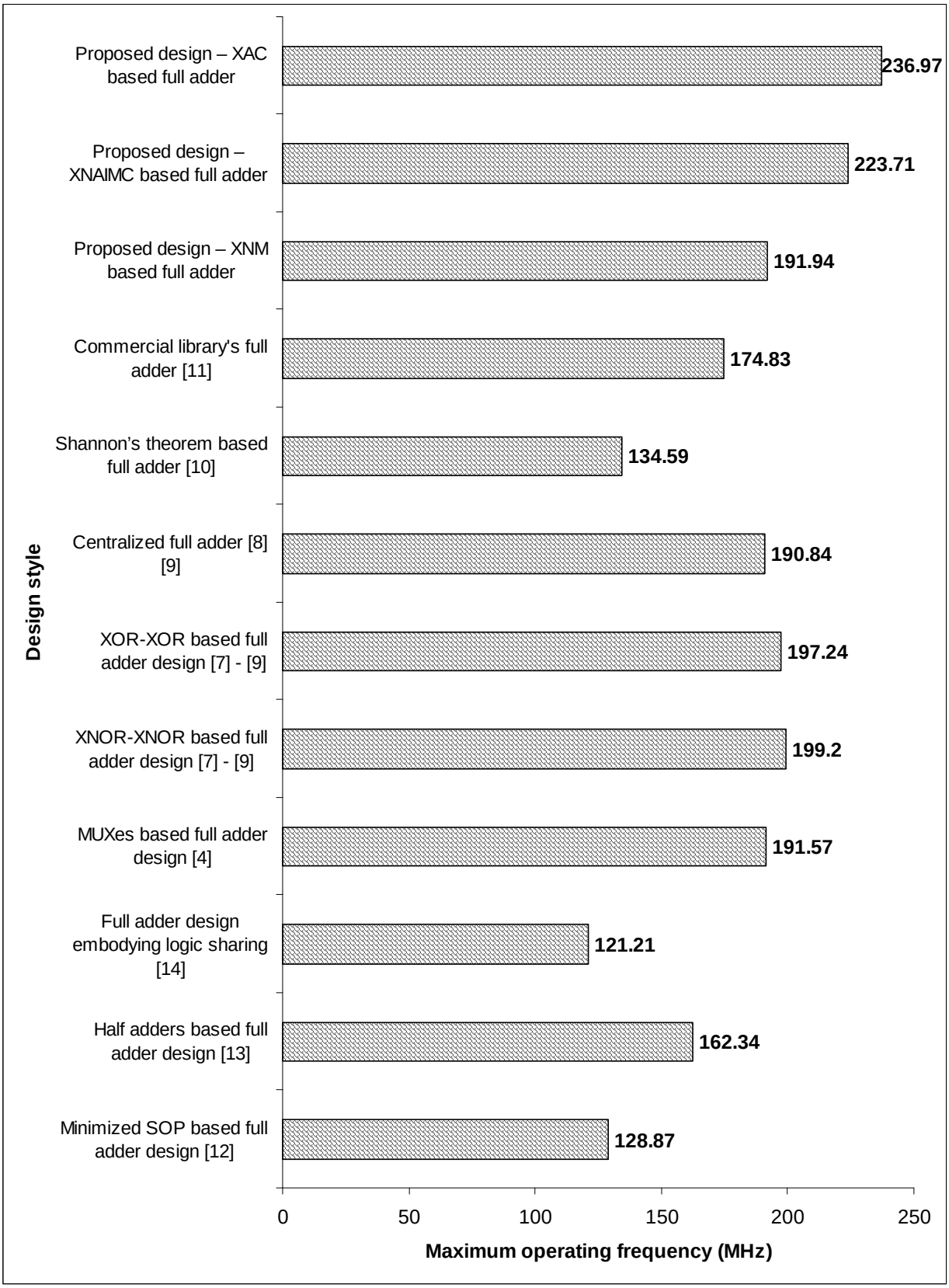

Fig. 15. Speed of operation of a 32-bit RCA, implemented using different full adder modules for a worst case library specification, with FO-4 drive strength

CMOS Logic Style, *IEEE Trans. on VLSI Systems*, Vol. 14, No. 12, December 2006, pp. 1309-1321.

[10] C. Senthilpari, A.K. Singh and K. Diwakar, Design Of A Low-Power, High-Performance, 8×8 Bit Multiplier Using A Shannon-Based Adder Cell, *Microelectronics Journal*, Vol. 39, No. 5, May 2008, pp. 812-821.

[11] STMicroelectronics CORE65LPLVT_1.10V Version 4.1 Standard Cell Library, *User Manual and Databook*, July 2006.

[12] R.K. Brayton, A.L. Sangiovanni-Vincentelli, C.T. McMullen and G.D. Hachtel, *Logic Minimization Algorithms for VLSI Synthesis*, Kluwer Academic Publishers, MA, USA, 1984.

[13] M. Morris Mano, *Digital Design*, 3rd Edition, Prentice-Hall Inc., NJ, USA, 2002.

[14] J.P. Uyemura, *CMOS Logic Circuit Design*, 1st Edition, Springer, 1999.