Optimized Design of 4-Bit Carry Select Adder with Low Area

Swetha telukula
Department of Electronics and Communication Engineering
BABA Institute of Technology and Sciences Visakhapatnam, Andhra Pradesh, India
telukulasweatha902@gmail.com
P.V.J Raj Kumar
Department of Electronics and Communication Engineering
BABA Institute of Technology and Sciences Visakhapatnam, Andhra Pradesh, India
pvjrajkumar@gmail.com
Bhaskara Rao Doddi
Department of Electronics and Communication Engineering
BABA Institute of Technology and Sciences Visakhapatnam, Andhra Pradesh, India
bhaskarvarmad@gmail.com

Abstract-Adder is an important module in main block of CPU. Carry select adder generally needs multiplexer where one data line can be hardware required for input carry of '0' and other can be for carry input of '1'. Here we proposed Carry select adder by optimizing the logic in few blocks and modifying the logic in other blocks. Complementary structures of AND, OR has used which normally leads to the logic optimization. With respect to the existing design, half sum carry generation unit has given 28% reduction in the MOSFETS, In Carry unit with zero selection optimization is nearly 33% with respect to the MOSFETS. Carry unit with one selection there was 14% reduction in MOSFETS. Full carry unit block has been optimized for about 33% with respect to the area. Full Sum unit block was modified but 0% reduction was possible with respect to the area. Total MOSFETS reduction for 4-bit is 21.7% and in equal proportion optimization is possible for N-bit.

I. INTRODUCTION

Minimized area, power, delay and cost of VLSI systems are the need for the industry whether the design is in soft IP or hard IP[1],[2]. A conventional carry select adder generates a dual of sum words and carry words [3]. A conventional CSLA has high speed than an RCA, but the design needs to sacrifice power as well as area. Few designs have been proposed to avoid massive area. Kim and Kim [4]circuit has proposed with the implementation of many to one line selecting circuit.

Design was proposed with large size adders with high speed by using a square-root (SQRT)[5].The main intention of SQRT-CSLA design is to reduce the number of levels required to increase the overall speed of theadder. Ramkumar and Kittur [6] suggested a binary toBEC-based CSLA. The BEC-based CSLA has less hardware than the existing CSLA, but it has more delay in comparison. A CSLA based on common Boolean logic is also proposed in [7] and [8]. The CBL-based CSLA of [7]involves significantly less hardware blocks than the existing CSLA but it has more delay, which is nearer to RCA .To enhance the speed, a SQRT-CSLA based on CBL was proposed in [8]. However, the CBL-based SQRT CSLA design of [8] requires more hardware and delay than the BEC-based SQRT-CSLA of [6]. Design in [10] used AND , Or, NOT and XOR for their design.

II. CARRY SELECT ADDER

Carry select adder has five blocks namely half sum generation unit, carry block when selection input is '0', carry block when selection input is '1', full carry unit and full sum unit.Half sum generation unit produced propogate output and generate part of carry output[10]. Proposed half sum generation unit will produce complement of propogate and complemented generate part of carry output. carry block when selection input is '0' produced carry outputs with OR and AND gates [10]. Proposed carry block when selection input is '0' will produce carry outputs with complement of OR gates. Carry block when selection input is '1' produced carry outputs with AND and OR gates [10]. Proposed carry block when selection input is '1' will produce carry outputs with complement version of OR gates. Full Carry block produced carry outputs with OR and AND gates [10]. Proposed full carry block will produce carry outputs with complemented version of AND gates. Full Sum block produced sum outputs with XOR gates [10]. Proposed full Sum block will produce outputs with complement version of XOR gates.

Proposed equations for N-bit carry select adder are mentioned below.

$C0= ((Cin.C_1^1(0))' . C_1^0(0))'$……..(1)

$C1= ((Cin.C_1^1(1))' . C_1^0(1))'$……..(2)

$C2= ((Cin.C_1^1(2))' . C_1^0(2))'$ …….(3)

$C3= ((Cin.C_1^1(3))' . C_1^0(3))'$ …….(4)

$CN= ((Cin.C_1^1(N))' . C_1^0(N))'$ ……..(5)

$S_0(0)$=(A0 XNOR B0)……. …….(6)

$S_0(1)$=(A1 XNOR B1) …….……(7)

$S_0(2)$=(A2 XNOR B2) ……..….(8)

$S_0(3)$=(A3 XNOR B3) ……..(9)

$S_0(N)$=(AN XNOR BN) ……..(10)

S0= (S_0(0) XNOR Cin) ……...(11)

S1= (S_0(1) XNOR C_0) …….(12)

S2= (S_0(2) XNOR C_1) ……...(13)

S3= (S_0(3) XNOR C_2) ……...(14)

SN= (S_0(N) XNOR C_{N-1}) ……..(15)

Equations from 1 to 5 are the full carry outputs in which C_1^1(N) denotes carry output when initial carry in is '1' for bit N and C_1^0(N) denotes carry output when initial carry in is '0' for bit N. Equations from 6 to 10 denotes half sum outputs in which S_0(N) corresponds to operation on bit N operands of A and B. This equations generate complemented versions of propagate. Equations from 11 to 15 denotes full sum outputs in which SN corresponds to operation on bit N operands of S_0 and bit N-1 operand of C. This equations generate uncomplemented versions of Sum.

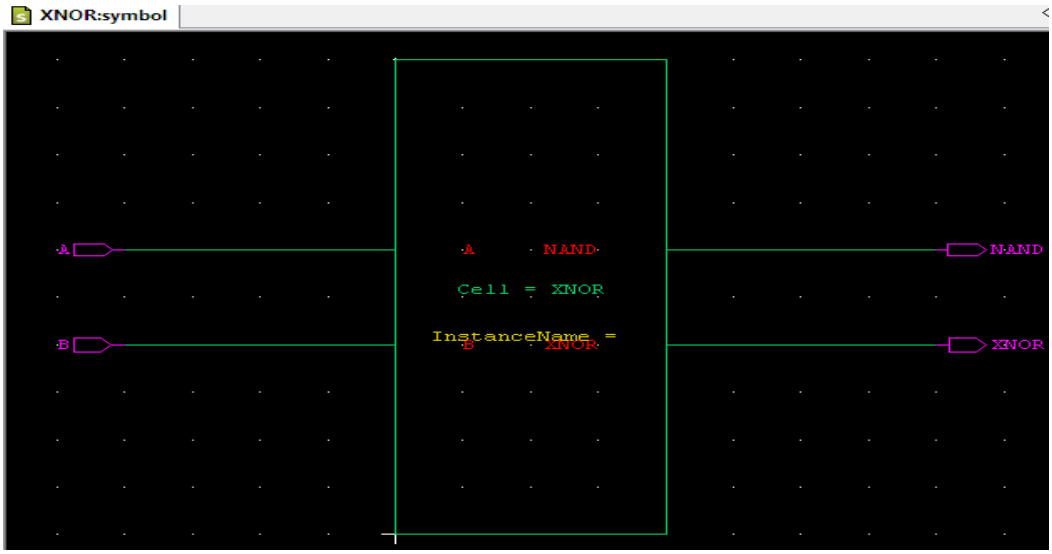

Figure 1. Symbol of XNOR.

Fig.1 indicates symbol of complemented XNOR in which it has two inputs and two outputs.

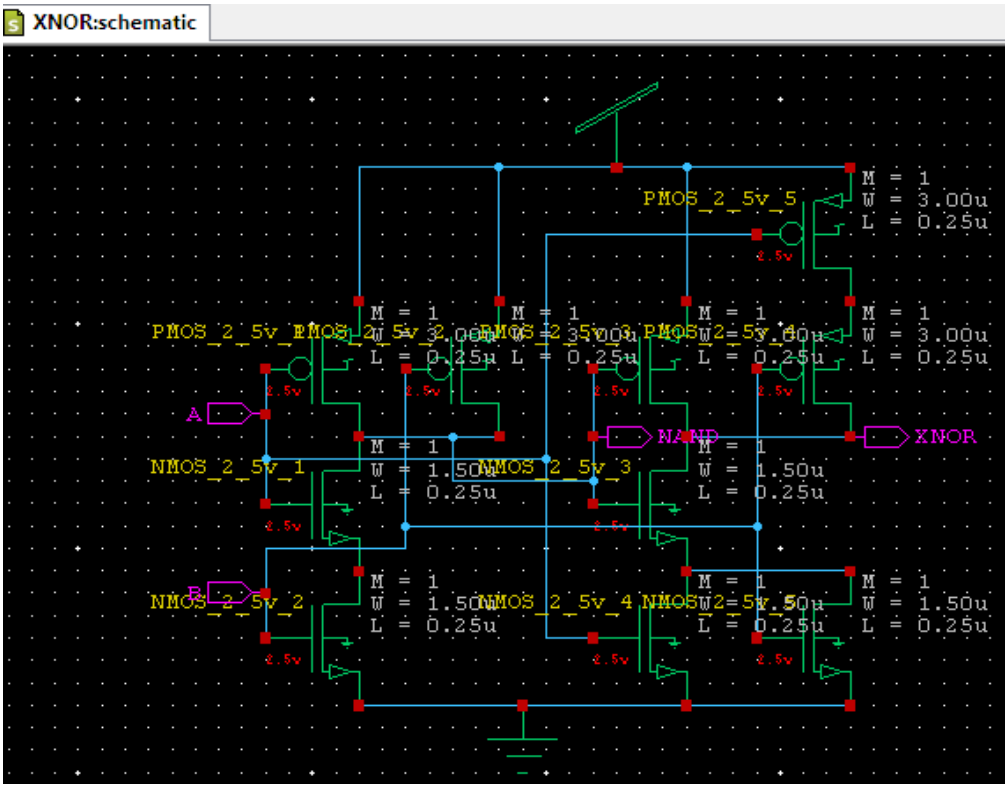

Figure 2. Circuit of XNOR.

Fig.2 indicates the circuit with two demarcation lines and it has 6 paths out of which only four are active paths and they are the possible input combinations of two inputs and it has the capability to generate complement version of AND and complement version of XOR which is the key for optimization

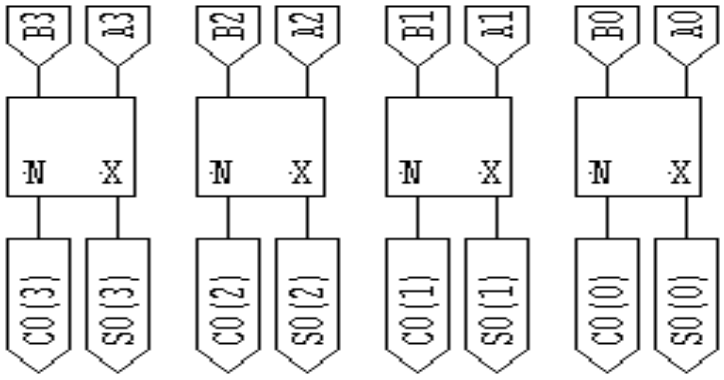

Figure 3. Circuit of Half Sum Carry Generation.

Fig.3 shows quad copies of half sum carry generation unit which is required for designing four bit carry select adder.

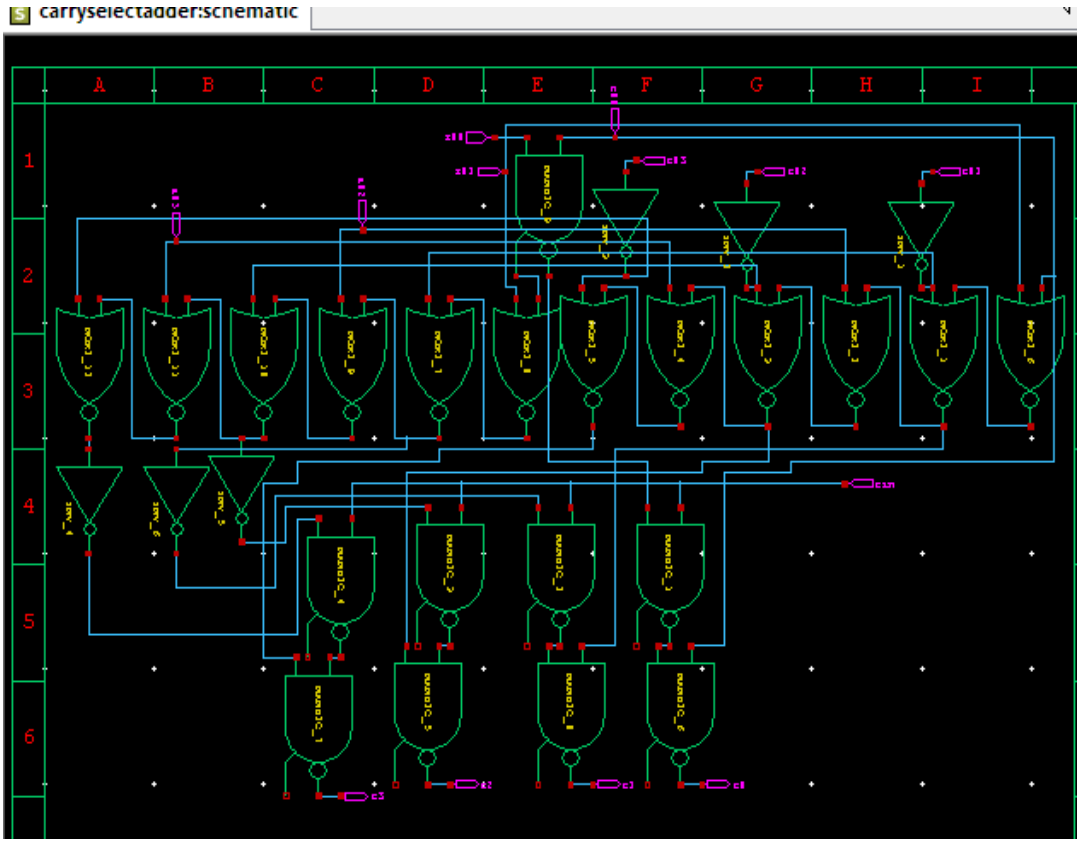

Figure 4. Circuit of full Carry Block.

Fig.4 indicates that complemented version of OR gate is needed to generate complemented carry outputs. Not gates are also needed to generate uncomplemented version of generates.There are 9 inputs out of which four inputs are hardware outputs of half sum carry generation XNOR block and four inputs are hardware outputs of half sum carry generation NAND block. One more input is initial carry in. There are four full carry outputs generated from this block and the overall hardware requires complemented version of OR, complemented version of AND and NOT gates to generate carry outputs. This block generates uncomplemented version form of carry outputs.

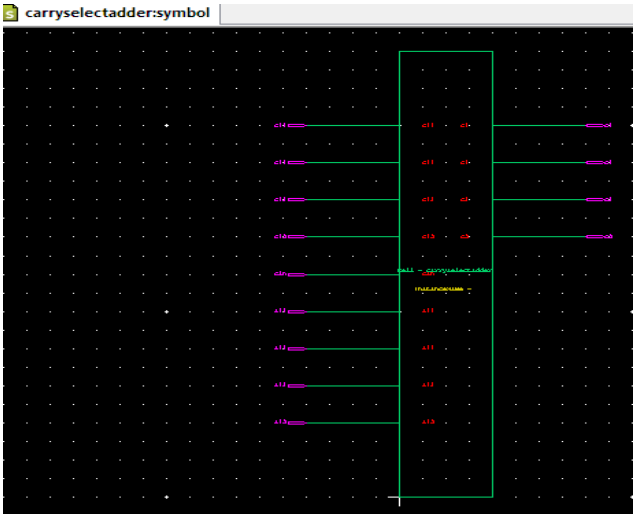

Fig.5 Symbol of full Carry Block.

Fig.5 indicates full carry block for four bit which has four outputs and 7 inputs together with generate and propagate inputs.

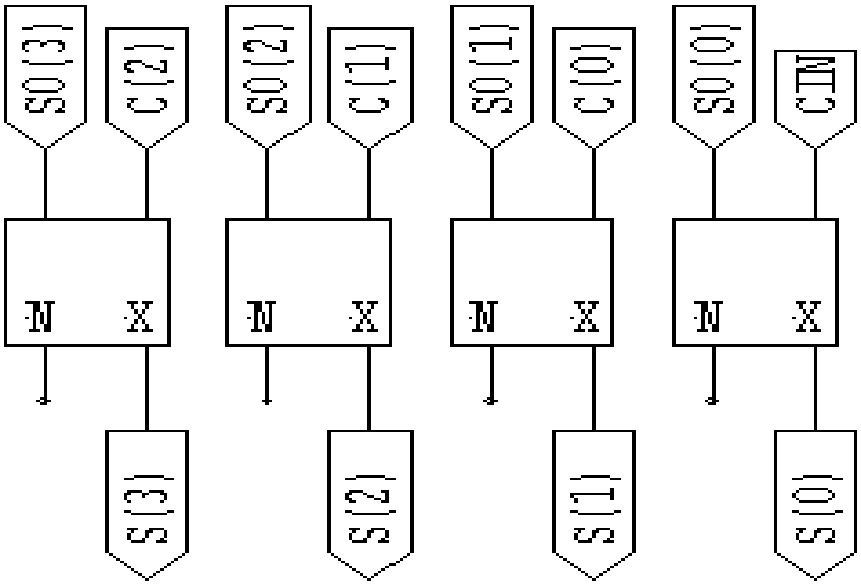

Figure 6. Circuit of Sum Block.

Fig.6 indicates that complement version of XOR gate is needed to generate full Sum outputs. This block generates uncomplemented form of sum outputs. It has only one level in which it has two outputs and the worst case delay of the two outputs are considered as the overall delay of this cell.

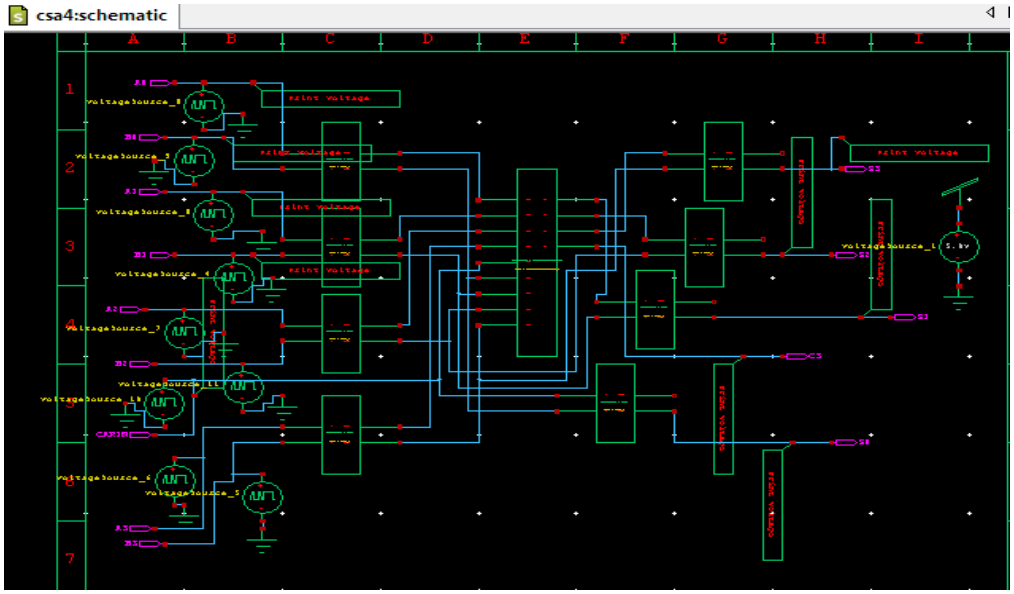

Figure 7. Circuit of Carry Select Adder Block.

Fig.7 indicates that Half Sum Carry Generation outputs are given to the inputs of carry for cin=0 and carry for cin=1.Outputs of Half Sum Carry Generation are also given to the SUM unit. Outputs of carry for sel=0 and carry for sel=1 will be the inputs of full carry outputs. Outputs of the full carry outputs will be the inputs of the full sum unit. It requires three levels of hardware to generate the output of full carry outputs and four levels of hardware to generate full sum outputs.

III. PROPOGATE/GENERATE

Propogate and generate will play key role in designing carry select adder but whether to design propogate or complemented version of propogate and whether to design generate or complemented version of generate is important in optimization point of view. Propogate requires hardware of XOR and complemented version of propogate requires as expected hardware of XNOR and similarly generate requires hardware of AND gate and it's complemented version requires NAND gate. If we design a cell such that it produces more than one usefull output is also the prime factor for optimization. To design the full sum output of full adder whether we go for XOR of all the three inputs or XNOR of three inputs is also the considerable factor in optimization point of view.

IV. PERFORMANCE ANALYSIS

Table 1. Comparison of Gates/Cells in Two Designs

S.No	Gate types/Cells used	Design in reference10	Proposed design
1	Half sum generation unit	XOR, AND(Gates)	XNOR(Cell)
2	Carry unit with zero carry input	AND, OR(Gates)	NOR, NOT(Gates)
3	Carry unit with one carry input	OR, AND(Gates)	NAND, NOT, NOR(Gates)
4	Carry unit	AND, OR(Gates)	NAND(Gate)
5	Sum unit	XOR(Gate)	XNOR(Gate)

Table1 shows that in proposed half sum generation unit XNOR cell has been used which can generate two outputs is the prime factor for optimization. Proposed design used NOR, NAND, NOT and XNOR. In the existing design XOR gates were used to design sum of full adder and proposed design used XNOR to design sum of full adder.

Table 2. Comparison of transistors in Two Designs

S.No	Transistor count(N-Bit)	Design in reference10	Proposed design
1	Half sum generation unit	16N	10N
2	Carry unit with zero carry	12(N-1)	10(N-1)

	input		
3	Carry unit with one carry input	((12N)-6)	((10N)-4)
4	Carry unit	12N	8N
5	Sum unit	10N	10N

Table2 shows that 6N number of transistors can be reduced in proposed half sum generation unit. Proposed Carry unit with zero carry input reduces 2(N-1) number of transistors. Proposed Carry unit with one carry input reduces (2N-2) number of transistors. Proposed Carry unit reduces 4N number of transistors. Proposed sum unit does not reduce any number of transistors.

Table 3. Comparison of Total Area in Two Designs

Area(N-bit)	Design in reference10	Proposed design
Transistors	62N-18	48N-14

Table3 shows there is considerable reduction in transistor count by (14N-4) and this will be very useful for large word lengths.

V. CONCLUSION

Proposed design has been optimized with respect to the area constraint of VLSI. one out of five blocks in the carry select adder has not been optimized in comparison with the existing design. Proposed design also has uniform structure such that it will be helpful if we go for ASIC implementation . Design approach can be extendable to N-bit. (14N-4) number of transistors have been optimized which leads to minimized power consumption. Optimization depends upon how affectively we manipulate the intermediate output values to make sure that in the final output we achieve exactly what we desire.

ACKNOWLEDGEMENT

This material is based upon work supported by the students of Baba Institute of Technology and sciences. Any opinions,findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily react the views of BITS.

REFERENCES

[1] K. K. Parhi, VLSI Digital Signal Processing. New York, NY USA:Wiley, 1998.

[2] A. P. Chandrakasan, N. Verma, and D. C. Daly, "Ultralow-power electronicsfor biomedical applications," Annu. Rev. Biomed. Eng., vol. 10, pp. 247–274, Aug. 2008.

[3] O. J. Bedrij, "Carry-select adder," IRE Trans. Electron. Comput.,vol. EC-11, no. 3, pp. 340–344, Jun. 1962.

[4] Y. Kim and L.-S. Kim, "64-bit carry-select adder with reduced area," Electron. Lett., vol. 37, no. 10, pp. 614–615, May 2001.

[5] Y. He, C. H. Chang, and J. Gu, "An area-efficient 64-bit square root carryselect adder for low power application," in Proc. IEEE Int. Symp. Circuits Syst., 2005, vol. 4, pp. 4082–4085.

[6] B. Ramkumar and H.M. Kittur, "Low-power and area-efficient carry-selectadder," IEEE Trans. Very Large Scale Integr. *(VLSI) Syst.*, vol. 20, no. 2,pp. 371–375, Feb. 2012.

[7] I.-C. Wey, C.-C. Ho, Y.-S. Lin, and C. C. Peng, "An area-efficient carry select adder design by sharing the common Boolean logic term," in Proc.IMECS, 2012, pp. 1–4.

[8] S.Manju and V. Sornagopal, "An efficient SQRT architecture of carry select adder design by common Boolean logic," in Proc. VLSI ICEVENT, 2013,pp. 1–5.

[9] B. Parhami, Computer Arithmetic: Algorithms and Hardware Designs,2nd ed. New York, NY, USA: Oxford Univ. Press, 2010.

[10]Basant Kumar Mohanty, Sujit Kumar Patel, "Area–Delay–Power Efficient Carry-Select Adder" IEEE Transactions on Circuits and Systems—II: Express Briefs, Vol. 61, no. 6, June 2014.