Design of Area–Delay–Power Efficient Carry Select Adder

G.Ravi
M.Tech Student,
Department of ECE,
Lords Institute of Engineering & Technology,
Hyderabad, India.

B.Sanjai Prasada Rao
Associate Professor,
Department of ECE,
Lords Institute of Engineering & Technology,
Hyderabad, India.

Abstract:

In this brief, the logic operations involved in conventional carry select adder (CSLA) and binary to excess-1 converter(BEC)-based CSLA are analyzed to study the data dependence and to identify redundant logic operations. We have eliminated all the redundant logic operations present in the conventional CSLA and proposed a new logic formulation for CSLA. In the proposed scheme, the carry select (CS) operation is scheduled before the calculation of final-sum, which is different from the conventional approach. Bit patterns of two anticipating carry words (corresponding to cin=0and 1) and fixed cbits are used for logic optimization of CS and generation units. An efficient CSLA design is obtained using optimized logic units. The proposed CSLA design involves significantly less area and delay than the recently proposed BEC-based CSLA. Due to the small carry-output delay,the proposed CSLA design is a good candidate for square-root (SQRT) CSLA. A theoretical estimate shows that the proposed SQRT-CSLA involves nearly 35% less area–delay–product (ADP)than the BEC- based SQRT-CSLA, which is best among the existing SQRT-CSLA designs, on average, for different bit-widths. The application-specified integrated circuit (ASIC) synthesis result shows that the BEC- based SQRT-CSLA design involves 48% more ADP and consumes 50% more energy than the proposed SQRT-CSLA, on average, for different bit-widths.

Index Terms:

Adder, arithmetic unit, low-power design.

I. INTRODUCTION:

Low-Power,area-efficient,and high performance VLSI systems are increasingly used in portable and mobiledevices, multi standard wireless receivers, and bio medical instrumentation[1], [2]. An adder is the main component of anarithmetic unit. A complex digital signal processing (DSP)system involves several adders.

An efficient adder design essentiallyimproves the performance of a complex DSP system.A ripple carry adder (RCA) uses a simple design, but carrypropagation delay (CPD) is the main concern in this adder Carry look-ahead and carry select (CS) methods have beensuggested to reduce the CPD of adders. A conventional carry select adder (CSLA) is an RCA–RCAconfiguration that generates a pair of sumwords and output carrybits corresponding the anticipated input-carry (c=0 and 1) and selects one out of each pair for final-sumandfinal-output-carry[3]. A conventional CSLA has less CPDthan an RCA, but the design is not attractive since it uses adual RCA.

Few attempts have been made to avoid dual use ofRCA in CSLA design. Kim and Kim [4] used one RCA andone add-one circuit instead of two RCAs, where the add-one. circuit is implemented using a multiplexer (MUX). He et al.[5] proposed a square-root (SQRT)-CSLAto implement largebit-width adders with less delay. In a SQRT CSLA, CSLAswith increasing size are connected in a cascading structure. Themain objective of SQRT-CSLA design is to provide a parallel path for carry propagation that helps to reduce the overall adder delay. Ram kumar and Kittur [6] suggested a binary to BEC- based CSLA.

The BEC- based CSLA involves less logicresources than the conventional CSLA, but it has marginally higher delay. A CSLA based on common Boolean logic (CBL) is also proposed in [7] and [8]. The CBL- based CSLA of [7] involves significantly less logic resource than the conventional CSLA but it has longer CPD, which is almost equal to that of the RCA. To overcome this problem, a SQRT-CSLA based on CBL was proposed in [8]. However, the CBL- based SQRTCSLA design of [8] requires more logic resource and delay than the BEC- based SQRT-CSLA of [6]. We observe that logic optimization largely depends on availability of redundant operations in the formulation, whereas adder delay mainly depends on data dependence. In the existing designs, logic is optimized without giving any consideration to the data dependence.

In this brief, we made an analysis on logic operations involved in conventional and BEC- based CSLAs to study the data dependence and to identify redundant logic operations. Based on this analysis, we have proposed a logic formulation for the CSLA. The main contribution in this brief are logic formulation based on data dependence and optimized carry generator (CG) and CS design. Based on the proposed logic formulation, we have derived an efficient logic design for CSLA. Due to optimized logic units, the proposed CSLA involves significantly less ADP than the existing CSLAs. We have shown that the SQRT-CSLA using the proposed CSLA design involves nearly 32% less ADP and consumes 33% less energy than that of the corresponding SQRT-CSLA. The rest of this brief is organized as follows. Logic formulation of CSLA is presented in Section II. The proposed CSLA is presented in Section III and the performance comparison is presented in Section IV. The conclusion is given in Section V.

II. LOGIC FORMULATION:

The CSLA has two units: 1) the sum and carry generator unit (SCG) and 2) the sum and carry selection unit [9]. The SCG unit consumes most of the logic resources of CSLA and significantly contributes to the critical path. Different logic designs have been suggested for efficient implementation of the SCG unit. We made a study of the logic designs suggested for the SCG unit of conventional and BEC- based CSLAs of [6] by suitable logic expressions. The main objective of this study is to identify redundant logic operations and data dependence. Accordingly, we remove all redundant logic operations and sequence logic operations based on their data dependence.
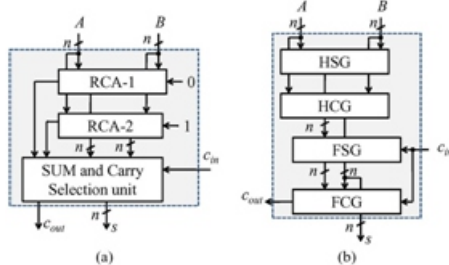
Fig. 1. (a) Conventional CSLA; n is the input operand bit-width. (b) The logic operations of the RCA is shown in split form, where HSG, HCG, FSG, and FCG represent half-sum generation, half-carry generation, full-sum generation, and full-carry generation, respectively.

A. Logic Expressions of the SCG Unit of the Conventional CSLA:

As shown in Fig. 1(a), the SCG unit of the conventional CSLA [3] is composed of two n-bit RCAs, where n is the adderbit-width. The logic operation of the n-bit RCA is performedin four stages: 1) half-sum generation (HSG); 2) half-carrygeneration (HCG); 3) full-sum generation (FSG); and 4) full carry generation (FCG). Suppose two n-bit operands are addedin the conventional CSLA, then RCA-1 and RCA-2 generaten-bit sum ($s0$ and $s1$) and output-carry ($c0out$ and c) correspondingto input-carry ($cin=0$ and $c=1$), respectively. Logic expressions of RCA-1 and RCA-2 of the SCG unit of then-bit CSLA are given

$$
\begin{aligned}
&s_{00}(i)=A(i) \oplus B(i) \quad c_{00in}(i)=A(i) \cdot B(i) \\
&(1a) s_{01}(i)=s_{00}(i) \oplus c_{01}(i-1) \quad (1b) c_{01}(i)=c_{00}(i)+s_{00}(i) \cdot c_{01}(i-1) \quad c_{00out}=c_{01}(n-1) \quad (1c) s_{10}(i)=A(i) \oplus B(i) \quad c_{10}(i)=A(i) \cdot B(i) \quad (2a) s_{11}(i)=s_{10}(i) \oplus c_{11}(i-1) \quad (2b) c_{11}(i)=c_{10}(i)+s_{10}(i) \cdot c_{11}(i-1) \quad c_{1out}=c_{11}(n-1) \\
&(2c) \text{where } c_{01}(-1)=0, \ c(-1)=1, \text{ and } 0=i=n-1.
\end{aligned}
$$

As shown in (1a)–(1c) and (2a)–(2c), the logic expressionof $\{s00(i), c0011(i)\}$ is identical to that of $\{s(i)\}$. Theseredundant logic operations can be removed to have an optimizeddesign for RCA-2, in which the HSG and HCG of RCA-1is shared to construct RCA-2. Based on this, [4] and [5] haveused an add-one circuit instead of RCA-2 in the CSLA, inwhich a BEC circuit is used in [6] for the same purpose. Since the BEC- based CSLA offers the best area–delay–powerefficiency among the existing CSLAs, we discuss here the logicexpressions of the SCG unit of the BEC- based CSLA as well.

B. Logic Expression of the SCG Unit of the BEC Based CSLA:

As shown in Fig. 2, the RCA calculates n-bit sum $s01$ andcoutcorresponding to cin=0. The BEC unit receives $s01$ from the RCA and generates (n +1)-bit excess-1 code. Themost significant bit (MSB) of BEC represents c1out, in which n least significant bits (LSBs) represent s11. The logic expressions of the RCA are the same as those given in (1a)–(1c). The logic expressions of the BEC unit of the n-bit BEC- based CSLA are given as

$$
\begin{aligned}
s_1^1(0) &= \overline{s_1^0}(0) \quad c_1^1(0) = s_1^0(0) \quad &(3a) \\
s_1^1(i) &= s_1^0(i) \oplus c_1^1(i-1) \quad &(3b) \\
c_1^1(i) &= s_1^0(i) \cdot c_1^1(i-1) \quad &(3c) \\
c_{out}^1 &= c_1^0(n-1) \oplus c_1^1(n-1) \quad &(3d)
\end{aligned}
$$

for $1 \le i \le n-1$.

We can find from (1a)–(1c) and (3a)–(3d)that, in \the case of the BEC- based CSLA, c11depends on s01, which otherwisehas no dependence on s01in the case of the conventional CSLA. The BEC method therefore increases data dependence in theCSLA. We have considered logic expressions of the conventionalCSLA and made a further study on the data dependenceto find an optimized logic expression for the CSLA.It is interesting to note from (1a)–(1c) and (2a)–(2c) that logic expressions of s01ands11are identical except the termsc01and c11since $(s01=S11 =s0)$ In addition, we find that c01 and c11 depend $\{s0,c0,cin\}$, where c0=c00 = c10 .Since c10 and c11 have no dependence on s01 and s 11 , the logic operation of c01and c11can be scheduled before s01

ands11,and the select unit can select one from the set (s01,s11) for the final-sum of the CSLA. We find that a significant amount of logic resource is spent for calculating {s01,s11}, and it is not an efficient approachto reject one sum-word after the calculation. Instead, one canselect the required carry word from the anticipated carry words{c0and c} to calculate the final-sum. The selected carry wordis added with the half-sum (s1) to generate the final-sum (s).Using this method, one can have three design advantages:1)Calculation of s01is avoided in the SCG unit; 2) the n-bitselect unit is required instead of the (n +1) bit; and 3) smalloutput-carry delay. All these features result in an area–delayand energy-efficient design for the CSLA. We have removed allthe redundant logic operations of (1a)–(1c) and (2a)–(2c) andrearranged logic expressions of (1a)–(1c) and (2a)–(2c) basedon their dependence. The proposed logic formulation for theCSLA is given as

$$s_0(i) = A(i) \oplus B(i) \quad c_0(i) = A(i) \cdot B(i) \quad (4a)$$
$$c_1^0(i) = c_1^0(i-1) \cdot s_0(i) + c_0(i) \quad \text{for } (c_1^0(0) = 0) \quad (4b)$$
$$c_1^1(i) = c_1^1(i-1) \cdot s_0(i) + c_0(i) \quad \text{for } (c_1^1(0) = 1) \quad (4c)$$
$$c(i) = c_1^0(i) \quad \text{if } (c_{in} = 0) \quad (4d)$$
$$c(i) = c_1^1(i) \quad \text{if } (c_{in} = 1) \quad (4e)$$
$$c_{out} = c(n-1) \quad (4f)$$
$$s(0) = s_0(0) \oplus c_{in} \quad s(i) = s_0(i) \oplus c(i-1). \quad (4g)$$

III. PROPOSED ADDER DESIGN

The proposed CSLA is based on the logic formulation given in (4a)–(4g), and its structure is shown in Fig. 3(a). It consists of one HSG unit, one FSG unit, one CG unit, and one CS unit CG unit is composed of two CGs (CG0 and CG) corresponding to input-carry '0' and '1'. The HSG receives two n-bit operands.

(A and B) and generate half-sum word s0and half-carry wordc0of width n bits each. Both CG0and CG1receive s from the HSG unit and generate two n-bit full-carry words cand ccorresponding to input-carry '0' and '1', respectively.The logic diagram of the HSG unit is shown in Fig. 3(b). The logic circuits of CG0 and CG1are optimized to take advantageof the fixed input-carry bits. The optimized designs of CG0andCG1 are shown in Fig. 3(c) and (d), respectively. The CS unit selects one final carry word from the two carrywords available at its input line using the control signal c.Itselects c01when cin=0;otherwise, it selects c. The CS unitcan be implemented using an n-bit 2-to-l MUX. However, wefind from the truth table of the CS unit that carry words c11and C11follow a specific bit pattern. If c10(i)='1', then c(i)=1,irrespective of s0(i) and c(i),for0 = i = n - 1. This featureis used for logic optimization of the CS unit. The optimizeddesign of the CS unit is shown in Fig. 3(e), which is composedof nAND–OR gates. The final carry word c is obtained from theCS unit. The MSB of c is sent to output as c0LSBs areXOR- ed with (n - 1) MSBs of half-sum (s0), and (n - 1)FSG [shown in Fig. 3(f)] to obtain (n - 1) MSBs of final-sum(s).The LSB ofs0is XOR- ed with cinto obtain the LSB of s.
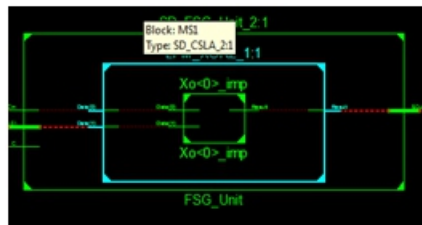
IV. PERFORMANCE COMPARISON
A. Area–Delay Estimation Method:

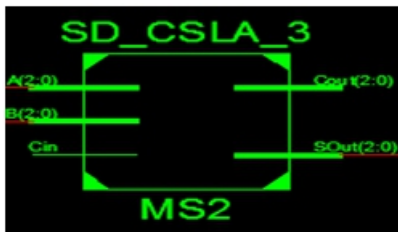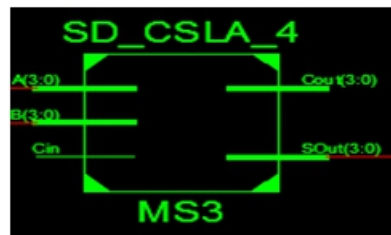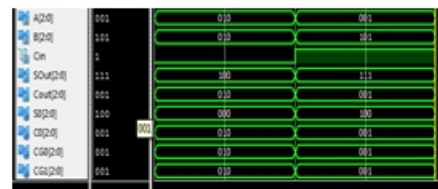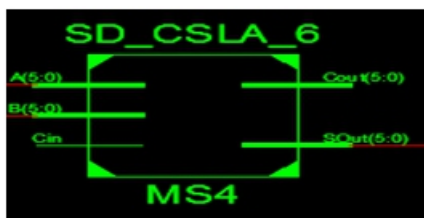We have considered all the gates to be made of 2-input AND,2-input OR, and inverter (AOI). A 2-input XOR is composed
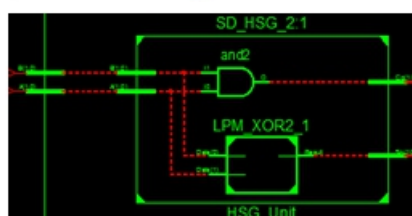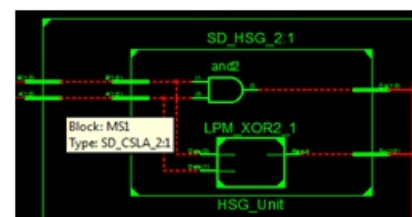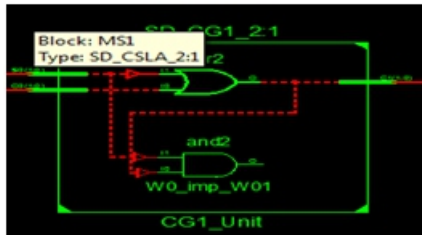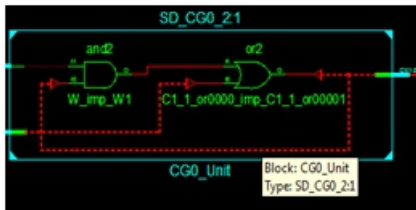
MS0

MS1

MS2

MS3
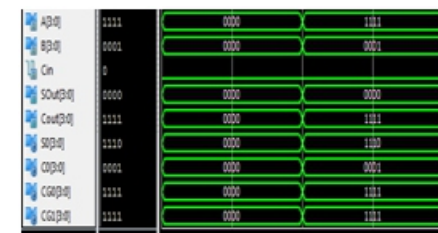
Ms4

Ms5

Ms6

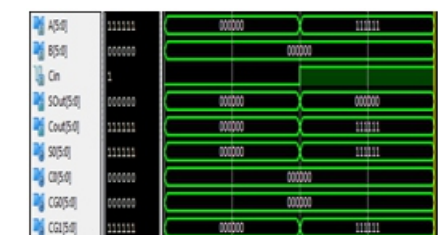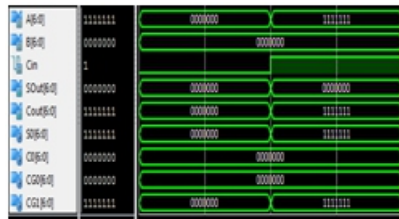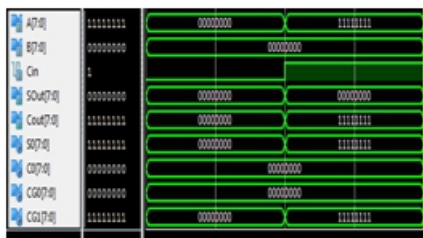## TABLE I: AREA AND DELAY OF AND, OR, AND NOT GATES GIVEN IN THE SAED 90-nm STANDARD CELL LIBRARY DATASHEET

	AND-gate	OR-gate	NOT-gate
Area (um^2)	7.37	7.37	6.45
Delay (ps)	180	170	100

AND,1OR, and 2 NOT gates. The area and delay of the 2-input AND, 2-input OR, and NOT gates(shown in Table I) are taken from the Synopsis Armenia Educational Department (SAED) 90-nm standard cell library data sheet for theoretical estimation. The area and delay of a design are calculated using the following relations:

$$A = a \cdot N_a + r \cdot N_o + i \cdot N_i \quad (5a)$$
$$T = n_a \cdot T_a + n_o \cdot T_o + n_i \cdot T_i \quad (5b)$$

Where (N_a, N_o, N_i)counts of the total design and itscritical path. (a, r, i) and (T), respectively,represent the area and delay of one (AND, OR, NOT) gate. We have calculated the (AOI) gate counts of each design for area and delay estimation. Using (5a) and (5b), the area and delay of each design are calculated from the AOI gate counts (N_a, T_o, T_i), (n_a, n_o, n_i), and the cell details of Table I.

B. Single-Stage CSLA:

The general expression to calculate the AOI gate counts of the n-bit proposed CSLA and the BEC- based CSLA of [6] and CBL- based CSLA of [7] and [8] are given in Table II of single stage design.

We have calculated the AOI gate counts on the critical path of the proposed n-bit CSLA and CSLAs of [6]–[8] and used those AOI gate counts in (5b) to find an expression for delay of final-sum and output-carry in the unit of T(NOT gate delay).
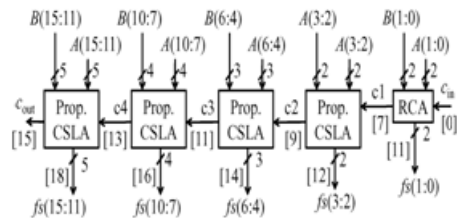

Fig. 4. Proposed SQRT-CSLA for n =16. All intermediate and output signals are labeled with delay (shown in square brackets).

whereas the CBL- based CSLA of [7] offers a single carry propagation path identical to the RCA design. Moreover, the proposed CSLA design has 0.45 ns less output-carry delay than the output-sum delay. This is mainly due to the CS unit that produces output-carry before the FSG calculates the final-sum.

C. Multistage CSLA (SQRT-CSLA):

The multi path carry propagation feature of the CSLA isfully exploited in the SQRT-CSLA [5], which is composed ofa chain of CSLAs. CSLAs of increasing size are used in theSQRT-CSLA to extract the maximum concurrence in the carrypropagation path. Using the SQRT-CSLA design, large-sizeadders are implemented with significantly less delay than asingle-stage CSLA of same size. However, carry propagationdelay between the CSLA stages of SQRT-CSLA is critical forthe overall adder delay.

Due to early generation of output-carrywith multipath carry propagation feature, the proposed CSLAdesign is more favorable than the existing CSLA designs forarea–delay efficient implementation of SQRT-CSLA. A 16-bitSQRT-CSLA design using the proposed CSLA, where the 2-bit RCA, 2-bit CSLA, 3-bit CSLA, 4-bitCSLA, and 5-bit CSLA are used. We have considered thecascaded configuration of (2-bit RCA and 2-, 3-, 4-, 6-, 7-,and 8-bit CSLAs) and (2-bit RCA and 2-, 3-, 4-, 6-, 7-, 8-, 9-,11-, and 12-bit CSLAs), respectively, for the 32-bit SQRTC-SLAand the 64-bit SQRT-CSLA to optimize adder delay. To demonstrate the advantage of the proposed CSLA design inSQRT-CSLA, we have estimated the area and delay of SQRTCSLA

using the proposed CSLA design and the BEC-basedC-SLA of [6] and the CBL-based CSLA of [7] for bit-widths16, 32, and 64. The delay of the CBL-based SQRT-CSLA [7] is significantly higher for large bit-widths than the proposed SQRT-CSLA and BEC-based SQRT-CSLA designs. Compared with SQRT-CSLA designs of [6] and [7], the proposed SQRTCSLA design, respectively, involves 35% and 72% less ADP, on average, for different bit-widths.

V. SIMULATION RESULT
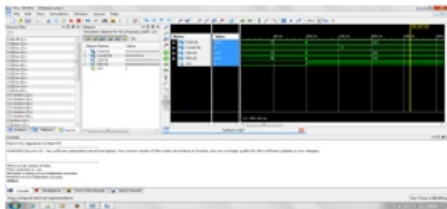
VI. CONCLUSION:

We have analyzed the logic operations involved in the conventional and BEC-based CSLAs to study the data dependence and to identify redundant logic operations. We have eliminated all the redundant logic operations of the conventional CSLA and proposed a new logic formulation for the CSLA. In the proposed scheme, the CS operation is scheduled before the calculation of final-sum, which is different from the conventional approach. Carry words corresponding to input-carry '0' and '1' generated by the CSLA based on the proposed scheme follow a specific bit pattern, which is used for logic optimization of the CS unit. Fixed input bits of the CG unit are also used for logic optimization. Based on this, an optimized design for CS and CG units are obtained. Using these optimized logic units, an efficient design is obtained for the CSLA. The proposed CSLA design involves significantly less area and delay than the recently proposed BEC-based CSLA. Due to the small carryoutput delay, the proposed CSLA design is a good candidate for the SQRT adder. The ASIC synthesis result shows that the existing BEC-based SQRT-CSLA design involves 48% more ADP and consumes 50% more energy than the proposed SQRTCSLA, on average, for different bit-widths.

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 electron-ics for 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 carryselectadder for low power application," inProc. IEEE Int. Symp. CircuitsSyst., 2005, vol. 4, pp. 4082–4085.

[6] B. Ramkumar and H. M. Kittur, "Low-power and area-efficient carry-select adder," 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 carryselect 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 selectadder 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.