Adders based on Binary Stored Carry-or-Borrow representation (BSCB)

Daniel Torno, exorand® technology

Standard Carry-Look-Ahead Adder

$$\begin{cases} g_{n-1} = a_{n-1} \cdot b_{n-1} \\ p_{n-1} = a_{n-1} \oplus b_{n-1} \end{cases}$$

$$\begin{cases} s_0 = p_0 \oplus cy_{in} \\ s_1 = p_1 \oplus g_0 \oplus p_0 \cdot cy_{in} \\ s_2 = p_2 \oplus g_1 \oplus p_1 \cdot g_0 \oplus p_1 \cdot p_0 \cdot cy_{in} \\ s_3 = p_3 \oplus g_2 \oplus p_2 \cdot g_1 \oplus p_2 \cdot p_1 \cdot g_0 \oplus p_2 \cdot p_1 \cdot p_0 \cdot cy_{in} \\ cy_{out} = g_3 \oplus p_3 \cdot g_2 \oplus p_3 \cdot p_2 \cdot g_1 \oplus p_3 \cdot p_2 \cdot p_1 \cdot g_0 \oplus p_3 \cdot p_2 \cdot p_1 \cdot p_0 \cdot cy_{in} \end{cases}$$

A new term q_i is substituted to g_i

BSCB CLA Adder

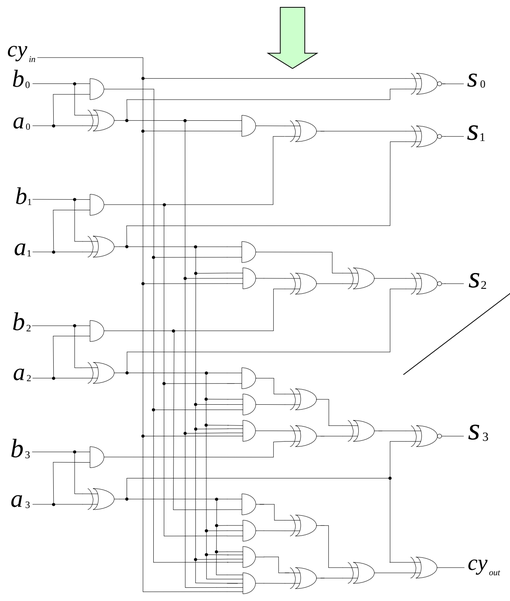
$$\begin{cases} q_0 = \overline{b_n \oplus cy_{in}} \\ q_n = \overline{b_n \oplus a_{n-1}} \\ p_n = a_n \oplus b_n \end{cases}$$

$$\begin{cases} s_0 = p_0 \oplus cy_{in} \\ s_1 = \overline{a_1} \oplus q_1 \oplus p_0 \cdot q_0 \\ s_2 = \overline{a_2} \oplus q_2 \oplus p_1 \cdot q_1 \oplus p_1 \cdot p_0 \cdot q_0 \\ s_3 = \overline{a_3} \oplus q_3 \oplus p_2 \cdot q_2 \oplus p_2 \cdot p_1 \cdot q_1 \oplus p_2 \cdot p_1 \cdot p_0 \cdot q_0 \\ cy_{out} = a_3 \oplus p_3 \cdot q_3 \oplus p_3 \cdot p_2 \cdot q_2 \oplus p_3 \cdot p_2 \cdot p_1 \cdot q_1 \oplus p_3 \cdot p_2 \cdot p_1 \cdot p_0 \cdot q_0 \end{cases}$$

BSCB Ripple-Carry Adder

Definition of a signal z_i such that:

$$\begin{cases} s_n = \overline{a_n} \oplus q_n \oplus z_{n-1} \\ z_n = p_n \cdot \left[q_n \oplus z_{n-1} \right] \end{cases}$$

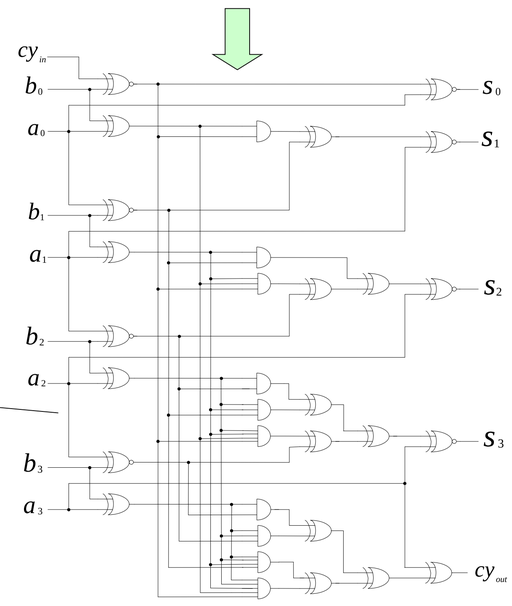


10 test vectors to detect all single stuck-at-faults

Boolean re-expression

Carry-save half-adder network

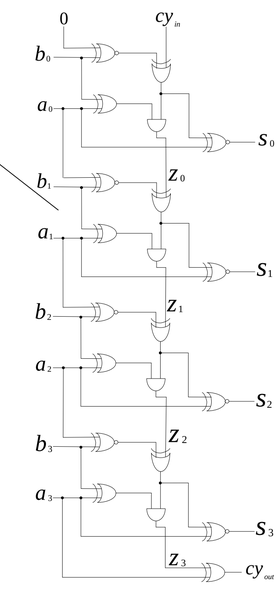
8 test vectors

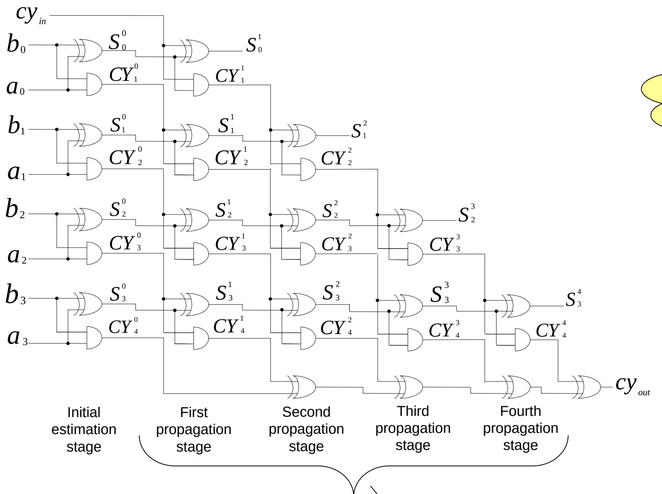


Boolean re-expression

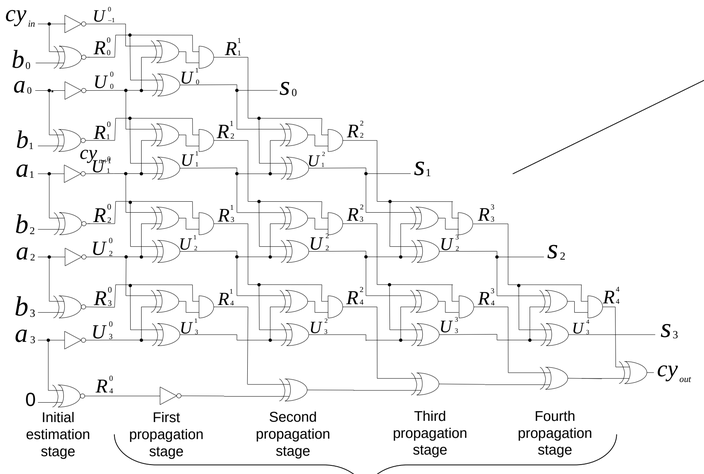
BSCB half-adder network

C-testable, 4 test vectors (versus 5 for standard implementation)





Why keeping things simple after all?



Signals	Positions					
	4	3	2	1	0	cy_{in}
a_n	0	1	0	0	0	1
b_n	0	0	1	1	1	
$U_n^0 = \overline{a_n}$	1	0	1	1	1	
$R_n^0 = \overline{b_n \oplus a_{n-1}}$	0	1	0	0	1	
$U_n^1 = U_n^0 \oplus R_n^0$	1	1	1	1	0	
$R_n^1 = \left(U_{n-1}^0 \oplus U_{n-2}^0\right).R_{n-1}^0$	1	0	0	1		
$U_n^2 = U_n^1 \oplus R_n^1$	0	1	1	0	0	
$R_n^2 = \left(U_{n-1}^1 \oplus U_{n-2}^1\right).R_{n-1}^1$	0	0	1			
$U_n^3 = U_n^2 \oplus R_n^2$	0	1	0	0	0	
$R_n^3 = \left(U_{n-1}^2 \oplus U_{n-2}^2\right).R_{n-1}^2$	0	1				
$U_n^4 = U_n^3 \oplus R_n^3$	0	0	0	0	0	
$R_n^4 = \left(U_{n-1}^3 \oplus U_{n-2}^3\right).R_{n-1}^3$	1					
S_n	1	0	0	0	0	

Inputs of the propagation stages are in the range: [0, +3]

Inputs of the propagation stages are in the range: [-1, +2]

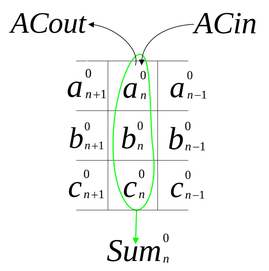
CY_{n+1}	S_n	Sum_n
0	0	0
0	1	+1
1	0	+2
1	1	+3

Expression of the partial sum Sum_n

R_{n+1}	U_n	Sum_n
0	0	0
0	1	+1
1	0	+2
1	1	-1

This range allows assumptions of input and output carries to compute the initial estimation Sum_n^0

Initial estimation for BSCB 3 bits addition



$$\begin{cases} \forall\, a_n, b_n, c_n: & -1 \le Sum_n^0 \le +2 \\ Sum_n^0 = a_n + b_n + c_n + ACin & -2 * ACout \end{cases}$$

$$\begin{cases} ACout = 1 \\ ACin = 1 \\ Sum_n^0 = a_n + b_n + c_n - 1 \end{cases} \Rightarrow b_n, c_n$$

$Sum_n^0 : \{U_n^0, R_{n+1}^0$	$a_n = 0$	$a_n = 1$
0,0	$-1: \{1,1$	$0: \{0,0$
0,1	$0: \{0,0$	$+1: \{1,0$
1,0	$0: \{0,0$	$+1: \{1,0$
1,1	$+1: \{1,0$	$+2: \{0,1$

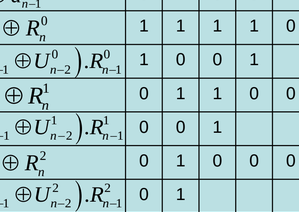
$$\Rightarrow \begin{cases} U_n^0 = \overline{a_n \oplus b_n \oplus c_n} \\ R_{n+1}^0 = \left(\overline{a_n \oplus b_n}\right) \cdot \left(\overline{b_n \oplus c_n}\right) \end{cases}$$

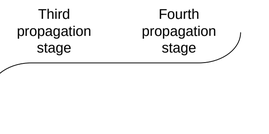
BSCB adder accumulator

Substitution:

$$\begin{cases} a_n = \overline{U_n^0} \\ b_n = R_n^0 \oplus U_{n-1}^0 \end{cases}$$

BSCB full-adder





A lot of open points:

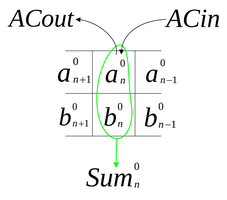
- Comparison with standard implementation in terms of power, speed and silicium area?
- -Reliability with parity check?
- -Reversible logic design?

Still hard work to be done for sure!

Initial estimations for BSCB 2 bits addition

$ACin$ = *Assumed input carry*

$ACout$ = *Assumed output carry*



$$\begin{cases} \forall\, a_n, b_n: & -1 \le Sum_n^0 \le +2 \\ Sum_n^0 = a_n + b_n + ACin & -2 * ACout \end{cases}$$

Three initial estimations

Unconditional carries:

$$\begin{cases} ACout = 0 \\ ACin = 0 \\ Sum_n^0 = a_n + b_n \end{cases} \Rightarrow b_n$$

$Sum_n^0 : \{U_n^0, R_{n+1}^0$	$a_n = 0$	$a_n = 1$
0	$0: \{0,0$	$+1: \{1,0$
1	$+1: \{1,0$	$+2: \{0,1$

$$\Rightarrow \begin{cases} U_n^0 = a_n \oplus b_n \\ R_{n+1}^0 = a_n \cdot b_n \end{cases}$$

Unconditional carries:

$$\begin{cases} ACout = 1 \\ ACin = 1 \\ Sum_n^0 = a_n + b_n - 1 \end{cases} \Rightarrow b_n$$

$Sum_n^0 : \{U_n^0, R_{n+1}^0$	$a_n = 0$	$a_n = 1$
0	$-1: \{1,1$	$0: \{0,0$
1	$0: \{0,0$	$+1: \{1,0$

$$\Rightarrow \begin{cases} U_n^0 = \overline{a_n \oplus b_n} \\ R_{n+1}^0 = \overline{a_n} \cdot \overline{b_n} \end{cases}$$

Conditional carries:

$$\begin{cases} ACout = 1 - b_{n+1} \\ ACin = 1 - b_n \\ Sum_n^0 = a_n - 1 + 2 * b_{n+1} \end{cases} \Rightarrow b_{n+1}$$

$Sum_n^0 : \{U_n^0, R_{n+1}^0$	$a_n = 0$	$a_n = 1$
0	$-1: \{1,1$	$0: \{0,0$
1	$+1: \{1,0$	$+2: \{0,1$

$$\Rightarrow \begin{cases} U_n^0 = \overline{a_n} \\ R_{n+1}^0 = \overline{a_n \oplus b_{n+1}} \end{cases}$$