Borrow Select Subtractor for Low Power and Area Efficiency

G Keshav and *V S Kanchana Bhaaskaran
School of Electronics Engineering, Vellore Institute of Technology Chennai
Chennai, India,
**vskanchana@ieee.org*

***Abstract*— Adders and subtractors are quintessential circuit modules in design of processors, which determine the speed of processors. The processing speed of subtraction is limited by the sequential borrow bit propagation from LSB to MSB, which in turn depends on the number of bits of operands (subtrahend and minuend). This paper presents two architectures of modified borrow select subtractor that consume lower power with increased area efficiency. The modifications carried out in the logical flow of subtraction process by using blocks with lower number of logic gates lead to a smaller number of gates, thus resulting in less device count, lower area and lower power dissipation. Validation of architectures are carried out using Cadence Virtuoso® simulations on UMC 90nm technology library. Comparison with existing architectures demonstrates the comparative advantages of proposed methods.**

Keywords— Borrow Select Subtractor, Low power Subtractor, Area Efficient Subtractor, BSLS, Adder-Subtractor

I. Introduction

Adders, subtractors and multipliers are the essential building blocks of processors [1] [2] [3] [4] [5] [6]. Normally, subtraction is done using adder-subtractor module, which can practically result in slowing down arithmetic operation, since the same hardware has to be used for both addition and subtraction processes using additional control signals. The commonly used Ripple Borrow Subtractor (RBS) used for subtraction of unsigned numbers possesses a simple architecture. However, performance of RBS is limited by borrow propagation time incurred from Least Significant Bit (LSB) to Most Significant Bit (MSB). In other words, the delay of RBS depends on binary word length. Borrow Select Subtractor (BSLS) architecture is proposed here to overcome the limitations of existing RBS with multiple RBS circuits [1]. This method generates partial difference and borrow using multiplexers (MUX) and the final difference and borrow is selected. As the BSLS utilizes multiple RBS circuits in it, it is proved normally area inefficient. The conventional method of subtraction for signed numbers uses two's complement method using addition. For addition process, references [7] [8] [9] [10] [11] [12] and [13] use adders. Note that [12] has lower power operational advantage and power delay product (PDP). Here, PDP is the product of power and delay.

The proposed architectures are compared with the conventional two's complement method found in literature, basically employing an adder [14] [15]. The adder given in [12] has been considered for comparison of two's complement method against the proposed architectures. For subtraction of signed numbers, a variant of BSLS [1] is proposed. The primary focus is on reducing area and power consumption, which is achieved by using logic blocks with fewer gates occupying less area even while aiming for same logic functionality. Section II presents BSLS in brief. Section III portrays the modified architecture of BSLS-BLO and the gate count calculation. Section IV describes the modified architecture of BSLS-RBHS and its gate count. Section V depicts the performance analysis and comparison in terms of power, delay and gate count. Section VI provides the summary and conclusion of the present work.

II. Overview of BSLS

Fig. 1 shows the structure of an half subtractor, implemented using an XOR gate, a NOT gate and an AND gate. While producing the output, half subtractor structure incurs a maximum latency of 2. Difference D incurs a latency of 1 and generation of borrow B incurs a latency of 2. Fig. 2 shows half adder implemented using an XOR gate and an AND gate. In order to produce sum S and carry C, this half adder incurs a maximum of one latency. Fig. 3 shows internal structure of a full subtractor realized using 2 XOR gates, 2 NOT gates, 2 AND gates and 1 OR gate. The full subtractor incurs a maximum latency of 4. Realization of difference D incurs a latency of 2 and borrow B introduces a latency of 4.

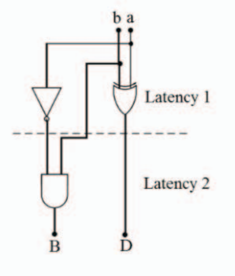

Fig. 1. Half Subtractor [1]

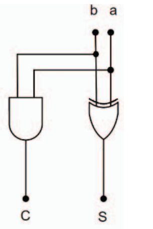

Fig. 2. Half Adder [1]

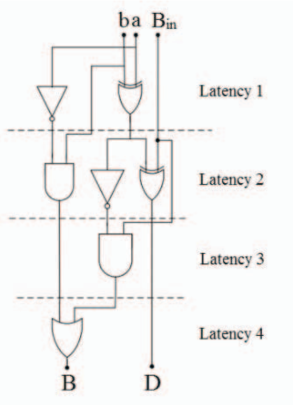

Fig. 3. Full Subtractor Gate Structure [1]

2159-3477/20/$31.00 ©2020 IEEE
DOI 10.1109/ISVLSI49217.2020.000-4

Fig. 4 depicts the structure of 2-bit Binary to Excess-1 Converter (BEC) implemented using 1 XOR, 1 AND and 1 NOT gate. The 16-bit BSLS for unsigned numbers consists of five logic groups. Each group computes the difference and borrow for different number of bits, as shown in Table I. Each functional group except first group has 2 n-bit RBS, n-bit BEC and a (2n+2):(n+1) multiplexer. The first group has only one two-bit RBS. The minuend and subtrahend are the inputs to RBS. This accounts to generation of the difference bit output when the borrow from the previous stage is 0. The minuend is increased by one using the BEC block. The subtrahend and increased minuend is given as input to RBS. This accounts to generation of difference bit when the borrow from previous stage is 1. The difference and borrow bits are available for borrow as 0 and 1. The correct output is selected based on the borrow bit obtained from the previous stage using the MUX.

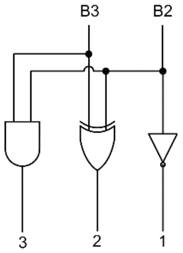

Fig. 4. 2-bit BEC [1]

TABLE I. FUNTIONAL TABLE FOR GROUP AND BITS

Group #	Number of bits #
1	2
2	2
3	3
4	4
5	5
Total bits	16

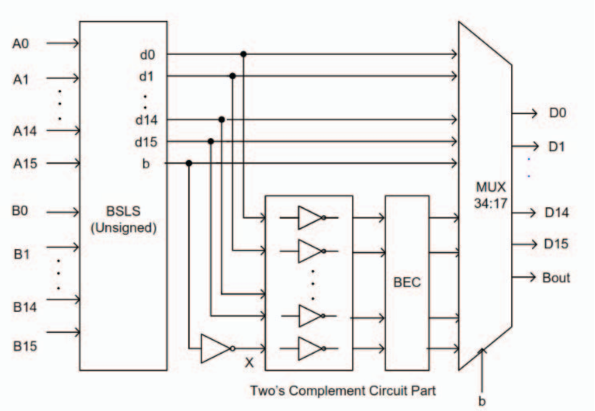

Fig. 5. Signed Borrow Select Subtractor [1]

The difference between the subtrahend and the minuend is in 2's complement form when the subtrahend is greater than minuend. To convert the two's complement form into signed magnitude form, additional blocks are required. When borrow output from unsigned BSLS is 0, the result is positive. Hence, the output doesn't necessitate the conversion to signed magnitude form. Similarly, as borrow output is 1, the result is negative. Then, the output of borrow-out is complemented and 2's complement is performed on entire result in signed magnitude form. Fig. 5 depicts the signed borrow select subtractor.

III. MODIFIED BSLS USING BLO (BSLS-BLO)

The proposed architecture replaces the second n-bit RBS and BEC-1 of BSLS using (n+1)-bit Binary-Less-One (BLO) logic in all groups. It uses Binary-Less-One (BLO) logic as shown in Fig. 6 for 3-bit BLO logic. Table II shows the functional table. The 3-bit BLO has been designed using Boolean expressions (1), (2) and (3). 4-bit BLO, 5-bit and 6-bit BLO structures can all be realized by using their respective Boolean expression forms.

$$X0 = \sim B0 \tag{1}$$

$$X1 = B1 \odot B0 \tag{2}$$

$$X2 = B2 \oplus \sim(B0 + B1) \tag{3}$$

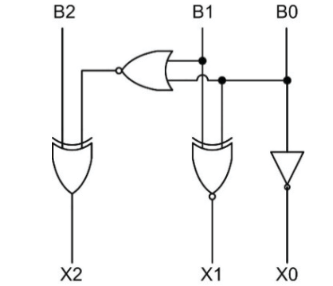

Fig. 6. 3-bit Binary-Less-One (BLO)

TABLE II. FUNCTIONAL TABLE OF 3-BIT BLO

B[2:0]	X[2:0]	B[2:0]	X[2:0]
000	111	100	011
001	000	101	100
010	001	110	101
011	010	111	110

The proposed Binary-Less-One (BLO) logic accomplishes the same operation of generating the difference and borrow assuming that the B_{in} from the previous stage is 1. This is analogous to the adder in [9] but the internal components are different. It takes the difference and borrow when the B_{in} is 0 as input and subtracting one from the received difference and borrow. The output of the BLO logic is the same as that of difference and borrow when B_{in} is 1. Fig. 7 shows the architecture of the modified Borrow Select Subtractor circuit configured using Binary-Less-One logic (BSLS-BLO). The Binary-Less-One logic is employed in all groups except the group 1 to compute difference and borrow bits. The partial difference and borrow bits obtained from first n-bit RBS, considering B_{in}=0 and Binary-Less-One (BLO) considered as B_{in}=1 are given as input bits to multiplexer, which selects the actual difference and borrow based on B_{in} generated from previous group. The first group has only one two-bit RBS, while, each of the functional groups except the first consists of one n-bit RBS, (n+1)-bit BLO and a (2n+2):(n+1) multiplexer.

A full subtractor normally requires 2 numbers each of XOR, NOT and AND gates and one OR gate. A half subtractor can be realized using one each of XOR, NOT and AND gates. To realise 2:1 MUX, one NOT gate, 2 AND gates and one OR gate are required. First 3-bit processing in RBS requires one half subtractor and 2 full subtractors. Thus, the first 3-bit RBS requires 5 XOR gates, 5 NOT gates, 5 AND gates and 2 OR gates. The 4-bit BLO is realized using 2 each of XOR, NOT and NOR gates and one XNOR gate. 8:4 MUX selects actual difference and borrow bits, and this comprises of four 2:1 multiplexers, containing 4 NOT, 8 AND and 4 OR gates.

Total gate count required to implement the group 3 structure in BSLS-BLO is thus the sum of counts, namely, 1) first 3-bit RBS, 2) 4-bit BLO and 3) 8:4 MUX. Therefore, the total number of gates employed to implement group 3 of BSLS-BLO is 7 XOR, one XNOR, 11 NOT, 13 AND, 6 OR and 2 NOR gates. Totally 40 gates are needed for group 3 in BSLS-BLO. The gate counts for the other groups have also been calculated and tabulated in Table III for comparison.

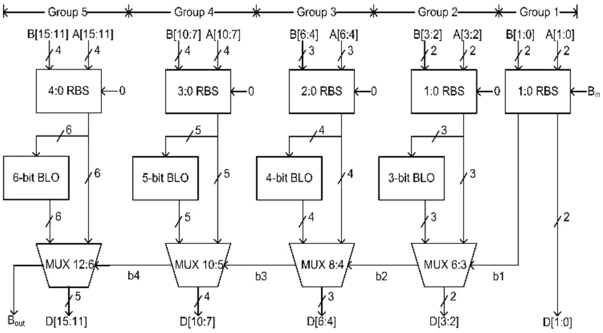

Fig. 7. Modified Borrow Select Subtractor using Binary-Less-One Logic

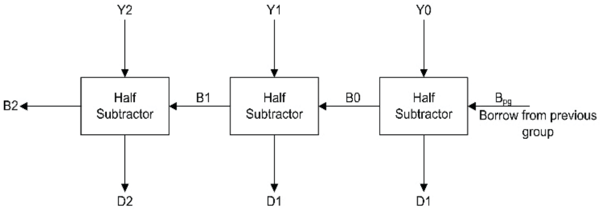

Fig. 8. 3-bit Ripple Borrow Half Subtractor (RBHS)

TABLE III. GATE COUNT FOR UNSIGNED BSLS USING BLO

Parameter	XOR	XNOR	NOT	AND	OR	NOR	Total
Group 1	4	0	4	4	2	0	14
Group 2	4	1	7	9	4	1	26
Group 3	7	1	11	13	6	2	40
Group 4	10	1	15	17	8	3	54
Group 5	13	1	19	21	10	4	68
Total	38	4	56	64	30	10	202

The modified 16-bit BSLS-BLO used for unsigned numbers requires 38 XOR, 4 XNOR, 56 NOT, 64 AND, 30 OR and 10 NOR gates. Totally 202 gates have been employed while constructing modified 16-bit BSLS using BLO for unsigned numbers. The 16-bit borrow select subtractor architecture for signed numbers has 16-bit borrow select subtractor, two's complement circuit and a 34:17 MUX. The two's complement circuit contains 18 NOT, 16 XOR and 15 AND gates. The 34:17 MUX has seventeen 2:1 MUX. Therefore, 34:17 MUX module contains 17 NOT, 34 gates and 17 OR gates. Thus, the modified 16-bit borrow select subtractor using BLO for signed numbers incurs only 320 gates in total.

IV. MODOFIED BSLS USING RIPPLE BORROW HALF SUBTRACTOR (BSLS-RBHS)

A second variant of the BSLS has been proposed in this Section, which replaces the second n-bit RBS and BEC-1 of BSLS or the n-bit BLO in BSLS-BLO using n-bit ripple borrow half subtractor (RBHS) in all groups. It employs Ripple Borrow Half Subtractor (RBHS) as shown in Fig. 8, as depicted for a 3-bit ripple borrow half subtractor (RBHS). It is analogous to the adder in [10]. The n-bit ripple borrow half subtractor takes the input from n-bit RBS with B_{in} as 0 for corresponding group. Depending on the borrow from previous group (B_{pg}), operation of RBHS varies. If borrow is 0, then the received input is passed as it is to output. Else, (B_{pg} = 1), output is produced by subtracting one from input. This is achieved by use of half subtractors. Table IV shows functional table of 3-bit RBHS. An n-bit RBHS comprises of n half subtractors. The input Y to each half subtractor is from n-bit RBS with B_{in}=0. The other input to the half subtractor is borrow from previous group or from previous half subtractor. The borrow output (B_{out}) from n^th^ half subtractor is the final borrow out of the n-bit RBHS.

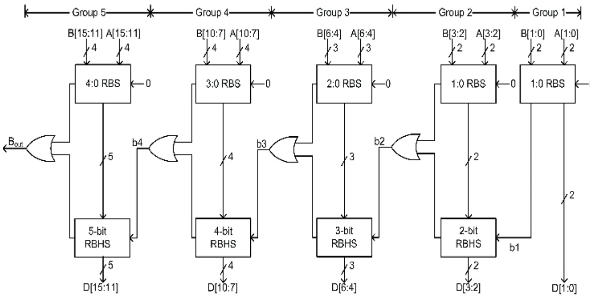

Fig. 9. Modified Borrow Select Subtractor using Ripple Borrow Half Subtractor

TABLE IV. FUNTIONAL TABLE OF 3-BIT RBHS

Bpg	Y[2:0]	B2	D[2:0]	Bpg	Y[2:0]	B2	D[2:0]
0	000	0	000	1	000	1	111
0	001	0	001	1	001	0	000
0	010	0	010	1	010	0	001
0	011	0	011	1	011	0	010
0	100	0	100	1	100	0	011
0	101	0	101	1	101	0	100
0	110	0	110	1	110	0	101
0	111	0	111	1	111	0	110

TABLE V. GATE COUNT FOR UNSIGNED BSLS USING RBHS

Parameter	XOR	NOT	AND	OR	Total
Group 1	4	4	4	2	14
Group 2	5	5	5	2	17
Group 3	8	8	8	3	27
Group 4	11	11	11	4	37
Group 5	14	14	14	5	47
Total Gates	42	42	42	16	142

Fig. 9 shows the structure of the modified borrow select subtractor circuit using ripple borrow half subtractor (BSLS-RBHS). Ripple borrow half subtractor is used in all groups except group 1 to compute difference and borrow bits. The partial difference and borrow bits obtained from first n-bit RBS, considering B_{in}=0 and the partial difference and borrow is given as input to the n-bit RBHS. Based on the borrow bit B_{in} generated from the previous group, the difference and borrow bit values have been computed. The borrow-out bit B_{out} from both n-bit RBS and n-bit RBHS are given as input to the OR gate. The output of the OR gate is the borrow-out bit B_{out} for the current group and borrow-in for the succeeding group. In this architecture, the (2n+2):(n+1) multiplexer in each group has been replaced with a OR gate. This results in reduction of number of gates. Each functional group except first group has one n-bit RBS, n-bit RBHS and an OR gate. The first group has only one two-bit RBS.

Conventionally, a full subtractor requires 2 XOR gates, 2 NOT gates, 2 AND gates and 2 OR gates. A half subtractor is used to be realized using one XOR gate, one NOT gate and one AND gate. First 3-bit processing in RBS requires one half subtractor and 2 full subtractors. Therefore, the first 3-bit RBS requires 5 XOR gates, 5 NOT gates, 5 AND gates and 2 OR gates. It may be noted that Group 3 of BSLS-RBHS contains 3-bit RBS and 3-bit RBHS. The 3-bit RBHS has been realized using 3 XOR gates, 3 NOT gates and 3 AND gates. Total gate count required to implement group 3 is the sum of gate counts required to implement first 3-bit RBS, 3-bit RBHS and a OR gate. Thus, the total number of gates employed to implement the group 3 is 8 XOR gates, 8 NOT gates, 8 AND gates and 3 OR gates. Altogether, 27 gates are required for group 3. Similarly, the gate count for the other groups are calculated and tabulated in Table V for due comparisons.

The modified 16-bit BSLS-RBHS designed for unsigned numbers requires 42 XOR gates, 42 NOT gates, 42 AND gates and 16 OR gates. Overall, 142 gates are required for constructing the modified 16-bit BSLS using RBHS for unsigned numbers. The 16-bit borrow select subtractor architecture for signed numbers has 16-bit borrow select subtractor, two's complement circuit and a 34:17 MUX. The two's complement circuit contains 18 NOT gates, 16 XOR gates and 15 AND gates. The 34:17 MUX has seventeen 2:1 MUX. Therefore, 34:17 MUX consists of 17 NOT gates, 34 AND gates and 17 OR gates in its making. Therefore, the modified 16-bit borrow select subtractor using RBHS for signed numbers incurs two hundred and sixty gates in total.

V. Results and Discussions

The proposed architectures have been designed under the same design environment for comparing the results justifiably. Table VI represents the power, delay, power delay product (PDP), device count comparison for unsigned operation.

TABLE VI. Comparison of Unsigned Subtractors

Word Size	Architecture	Power (uW)	Delay (ps)	PDP (fJ)	Device Count #
4-bit	RBS	26.01	310.6	8.078	28
	BSLS	37.70	232.8	8.776	51
	BSLS-BLO	33.71	232.8	7.847	40
	BSLS-RBHS	26.83	265.7	7.128	31
8-bit	RBS	50.05	647.1	32.387	56
	BSLS	91.84	344.6	31.648	128
	BSLS-BLO	73.85	344.6	25.448	94
	BSLS-RBHS	55.05	477.1	26.264	68
16-bit	RBS	92.59	1320.0	122.218	112
	BSLS	210.50	567.4	119.437	282
	BSLS-BLO	142.40	567.4	80.797	202
	BSLS-RBHS	107.30	900.0	96.570	142

The power consumed by the modified BSLS-BLO is reduced when compared to BSLS, while holding on to the same delay as that of the BSLS for different word size. The result is shown in Fig. 10 and Fig. 11. It can also be inferred that the delay of BSLS-RBHS is more when compared to BSLS but the power is less when compared to other architectures for different word size. PDP values of 4, 8, and 16-bit BSLS-BLO is lesser by 10.5%, 19.5%, 32.3%, respectively than BSLS as shown in Figs. 12 and 13. The PDP of 4, 8 and 16-bit BSLS-RBHS is less by 18.7%, 17.0% and 19.1%, respectively, when compared to BSLS. The PDP of BSLS-RBHS for 4-bit is found lower when compared to BSLS-BLO. On the other hand, for 8-bit and 16-bit word sizes, the PDP increases. It is also validated that the BSLS-BLO outperforms the BSLS-RBHS in terms of PDP as the word size increases.

The modified BSLS-BLO for 4, 8 and 16-bit need 11, 34 and 80 fewer gates, respectively, when compared to BSLS. This accounts to 21.5%, 26.5% and 28.3% reduction in number of gates employed, respectively, as shown in Fig. 13. The modified signed BSLS-RBHS for 4-bit, 8-bit, 16-bit utilises 20, 60 and 140 fewer gates, respectively, when compared to BSLS, accounting to 39.2%, 46.8% and 49.6% reduction in gate counts. For 4-bit words, it is found preferable to use BSLS-RBHS, due to the fact that it has lower PDP and device count when compared to counterparts. For increased word size, preference can be made between BSLS-BLO and BSLS-RBHS since the BSLS-BLO claims advantage of lower PDP and BSLS-RBHS claims the advantage of lower device count.

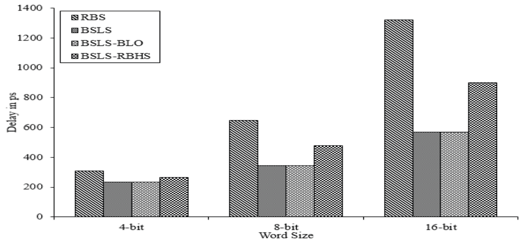

Fig. 10. Unsigned Delay Comparison

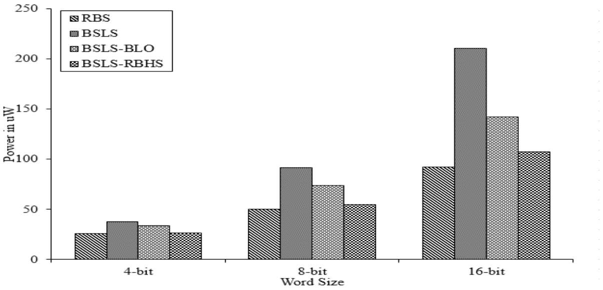

Fig. 11. Unsigned Power Comparison

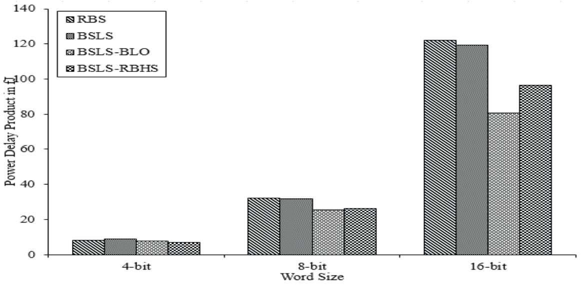

Fig. 12. Unsigned PDP Comparison

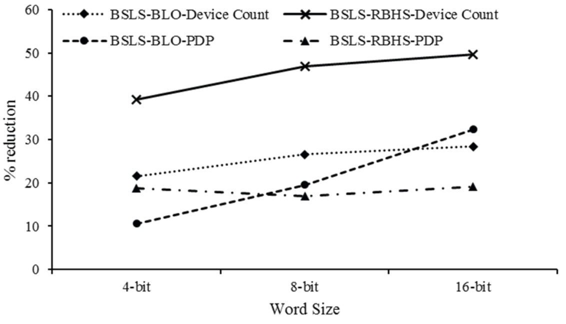

Fig. 13. Percentage Reduction in Device Count, PDP of Unsigned Architectures

Table VII illustrates the power, delay, Power-Delay-Product PDP and total device count for signed operation of subtractor circuit. Typical NMOS and PMOS devices have been used to obtain the results. Modified signed BSLS-BLO and signed BSLS-RBHS enjoy reduction in power when compared to the BSLS and the conventional 2's complement method for different word size as shown in Fig. 14.

Fig.15 depicts delay of signed architectures for 4-bit, 8-bit and 16-bit. The delay of signed BSLS-BLO is more or less equal to the delay of BSLS and it is less than 2's complement method for 8-bit and 16-bit. The delay of signed BSLS-RBHS is more when compared to BSLS for different word size. However, when compared to conventional 2's complement method, delay is more for 4-bit and 8-bit.

The power delay product of 4-bit, 8-bit and 16-bit signed BSLS-BLO is lower by 16.3%, 18.6% and 22.0%, respectively, when compared to BSLS as represented in Fig. 16 and Fig. 17. The power delay product of 4-bit, 8-bit and 16-bit BSLS-RBHS is reduced by 24.1%, 17.9% and 6.7%, respectively, when compared to BSLS as represented in Figs. 16 and 17. The PDP of signed BSLS-RBHS for 4-bit is less than that of BSLS-BLO. However, 8-bit and 16-bit word sizes the PDP increases due to the increased complexity in architecture. It is also observed that signed BSLS-BLO outperforms the signed BSLS-RBHS in terms of PDP as the word size increases.

TABLE VII. COMPARISON OF SIGNED SUBTRACTORS

Word Size	Architecture	Power (uW)	Delay (ps)	PDP (fJ)	Device #
4-bit	2's Complement	73.13	408.4	29.866	103
	BSLS	62.02	427.7	26.525	85
	BSLS-BLO	51.86	427.7	22.180	74
	BSLS-RBHS	43.97	457.6	20.120	65
8-bit	2's Complement	137.70	700.4	96.445	169
	BSLS	147.30	575.4	84.756	190
	BSLS-BLO	119.80	575.4	68.932	156
	BSLS-RBHS	98.380	706.5	69.505	130
16-bit	2's Complement	324.60	1216.0	394.713	333
	BSLS	347.10	874.5	303.538	400
	BSLS-BLO	270.50	874.5	236.552	320
	BSLS-RBHS	234.80	1205.0	282.934	260

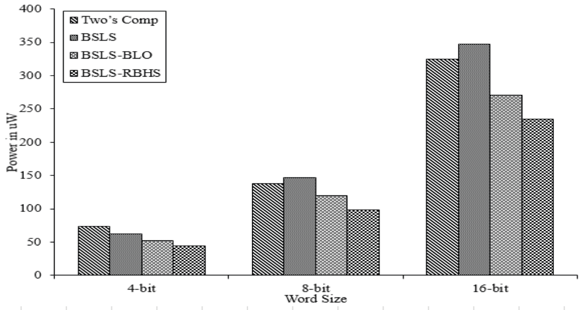

Fig. 14. Signed Power Comparison

The modified signed BSLS-BLO for 4-bit, 8-bit and16-bit requires 11, 34 and 80 fewer gates, respectively, when compared to BSLS. This accounts to 12.9%, 17.8% and 20.0% reduction in the number of gates used, respectively, as shown in Fig. 17. The modified signed BSLS-RBHS for 4-bit, 8-bit and 16-bit utilizes 20, 60 and 140 fewer gates, respectively, when compared to BSLS. This accounts to 23.5%, 31.5% and 35.0% reduction in the number of gates needed respectively, as shown in Fig. 17.

When compared to the conventional 2's complement method, the delay of BSLS is lower for 8-bit and 16-bit architectures, however, at the cost of increased power. In case of proposed BSLS-BLO, the delay and power is comparatively lower when compared to the conventional 2's complement method. In case of the second proposed architecture, BSLS-RBHS, the delay in case of 4-bit and 8-bit is more when compared to two's complement method and BSLS. And, the power consumed is lower when compared to other architecture counterparts. The delay incurred by 16-bit BSLS-RBHS is lower than that of 16-bit two's complement method. However, the delay experienced is greater than that of BSLS and BSLS-BLO.

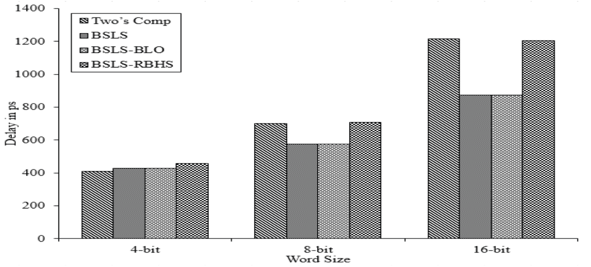

Fig. 15. Signed Delay Comparison

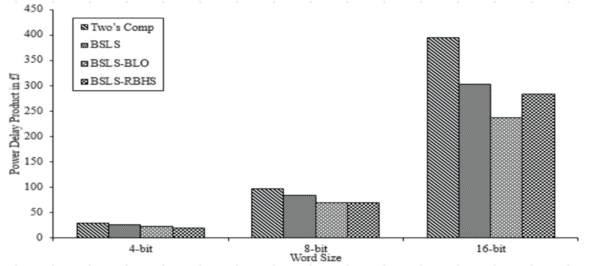

Fig. 16. Signed PDP Comparison

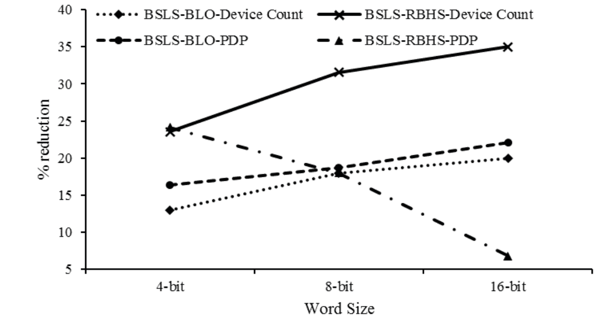

Fig. 17. Percentage Reduction in Device Count, PDP of Signed Architectures

Table VIII presents data pertaining to power consumption, delay and power delay product (PDP) under various process corners, viz. Fast-Fast, Fast-Slow, Slow-Fast, Slow-Slow for 16-bit unsigned RBS, BSLS, BSLS-BLO and BSLS-RBHS architectures. It is observed that under all the process corners, power consumed by the modified BSLS-BLO and modified BSLS-RBHS is less than that incurred by BSLS and the conventional method of two's complement, thus validating our approach. For the above corner cases, delay for BSLS-BLO is found to be equal to BSLS and less than that of RBS and delay is more for BSLS-RBHS than BSLS and is less than that of RBS. Under all process corners, power consumption is lower for BSLS-BLO and BSLS-RBHS architectures when compared to BSLS, thus validating design efficiency. The PDP for proposed BSLS-BLO and BSLS-RBHS is less than BSLS and RBS. Comparison against BSLS-BLO and BSLS-RBHS, PDP of BSLS-BLO is less in all process corners.

TABLE VIII. COMPARISON OF UNSIGNED SUBTRACTORS UNDER DIFFERENT PROCESS CORNERS

Process Corner	Architecture	Power (uW)	Delay (ps)	PDP (fJ)
Normal (TT)	RBS	92.59	1320.0	122.218
	BSLS	210.50	567.4	119.437
	BSLS-BLO	142.40	567.4	80.797
	BSLS-RBHS	107.30	900.0	96.570
FF	RBS	169.00	1100.0	185.900
	BSLS	396.30	479.5	190.025
	BSLS-BLO	268.20	479.5	128.601
	BSLS-RBHS	202.70	743.1	150.626
FS	RBS	93.85	1325.0	124.351
	BSLS	214.30	578.0	123.865
	BSLS-BLO	144.90	578.0	83.752
	BSLS-RBHS	108.80	905.1	98.474
SF	RBS	361.4	1318.0	476.325
	BSLS	386.3	559.1	215.980
	BSLS-BLO	300.0	559.1	167.730
	BSLS-RBHS	260.3	899.7	234.191
SS	RBS	81.25	1646.0	133.737
	BSLS	181.00	702.0	127.062
	BSLS-BLO	122.10	702.0	85.714
	BSLS-RBHS	93.03	1137.0	105.775

TABLE IX. COMPARISON OF SIGNED SUBTRACTORS UNDER DIFFERENT PROCESS CORNERS

Process Corner	Subtractor Type	Power (uW)	Delay (ps)	PDP (fJ)
Normal (TT)	2's Complement	324.60	1216.0	394.713
	BSLS	347.10	874.5	303.538
	BSLS-BLO	270.50	874.5	236.552
	BSLS-RBHS	234.80	1205.0	282.934
FF	2's Complement	567.5	1026.0	582.255
	BSLS	623.1	742.7	462.776
	BSLS-BLO	486.2	742.7	361.100
	BSLS-RBHS	421.4	1005.0	423.507
FS	2's Complement	323.8	1214.0	393.093
	BSLS	350.8	893.0	313.264
	BSLS-BLO	273.1	893.0	243.878
	BSLS-RBHS	236.9	1218.0	288.544
SF	2's Complement	361.4	1224.0	442.353
	BSLS	386.3	859.3	331.947
	BSLS-BLO	300.0	859.3	257.790
	BSLS-RBHS	260.3	1198.0	311.839
SS	2's Complement	274.0	1508.0	413.192
	BSLS	290.1	1076.0	312.147
	BSLS-BLO	223.2	1076.0	240.163
	BSLS-RBHS	192.7	1508.0	290.591

Table IX presents the comparative details of power consumption, delay and power delay product under various process corners for 16-bit signed Two's complement, BSLS, BSLS-BLO and BSLS-RBHS architectures. Under all the process corners, the PDP of proposed BSLS-BLO and BSLS-RBHS is less when compared to the Two's complement architecture and BSLS. Comparison between BSLS-BLO and BSLS-RBHS shows that BSLS-BLO outperforms BSLS-RBHS in terms of delay and power delay product in all process corners.

VI. CONCLUSION

This paper presents the design aspect, architecture and comparative operational advantages of modified borrow select subtractor using BLO and modified borrow select subtractor using RBHS operating at the same performance parameters, even while incurring lower power consumption even while maintaining the speed performance without incurring any adverse impact on the delay of the system. The performance comparison of modified BSLS-BLO and BSLS-RBHS with borrow select subtractor found in literature, in terms of power, PDP (Power-Delay Product) and device count depicts the advantages of the BSLS-BLO and BSLS-RBHS. The number of gates employed is less in both the modified architectures that the counterpart architectures. This design approach also leads to reduced number of transistors utilized, lower area and reduced power consumption. Under all the process corners, the BSLS-BLO and BSLS-RBHS outperform the borrow select subtractor found in the literature in terms of power delay product.

REFERENCES

[1] Amit Maruti Kunjir and V S Kanchana Bhaaskaran, "A high speed borrow select 16-bit subtractor", The Patent Office Journal Appl. 2868/CHE/2014 A, Jan. 22, 2016.

[2] L.E.M. Bckenbury and W.Shao, "Lowering Power in an Experimental RISC processor", Microprocessor and Microsystems, pp. 360-368, 2007.

[3] V Jayaprakasan, S Vijayakumar, V S Kanchana Bhaaskaran, "Evaluation of the Conventional vs. Ancient Computation methodology for Energy Efficient Arithmetic Architecture", Int. Conf on Process Automation, Control and Computing (PACC), 2011, 20-22.

[4] Kore Sagar Dattatraya, Belgudri Ritesh Appasaheb, Ramdas Bhanudas Khaladkar and V. S. Kanchana Bhaaskaran, "Low Power High Speed and Area Efficient Binary Count Multiplier", Journal of Circuits, Systems, and Computers, Vol. 25, No. 4 (2016) 1650027.

[5] B. R. Appasaheb and V.S.Kanchana Bhaaskaran, "Design and Implementation of an Efficient Multiplier Using Vedic Mathematics and Charge Recovery Logic", Proc. of Int. Conf. on VLSI, Communication, Advanced Devices, Signals & Systems and Networking (VCASAN-2013), Lecture Notes in Elec Engg, 258, July 2013, Chap 16, Pp. 101-108.

[6] Dalal Rutwik Kishor and V.S. Kanchana Bhaaskaran "Low Power Divider Using Vedic Mathematics", Third International Conference on Advances in Computing, Communications and Informatics (ICACCI-2014), 24-27 Sept. 2014 in Delhi, Pp. 575-580, 978-1-4799-3080-7/14.

[7] O. J. Bedrij, "Carry-select adder", IRE Transactions on Electronics & Computers, pp.340–344, 1962.

[8] T. Y. Ceiang and M. J. Hsiao, "Carry-Select Adder Using Single Ripple Carry Adder," Electronics Letters, V. 34, No. 22, Pp. 2101–03, Oct. 1998.

[9] B. Ramkumar and Harish M Kittur, "Low-Power and Area-Efficient Carry Select Adder", IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 2, pp.371-75, Feb. 2012.

[10] Kore Sagar Dattatraya and V. S. Kanchana Bhaaskaran, "Modified Carry Select Adder using Binary Adder as a BEC-1,", European Journal of Scientific Research V.103, no.1, pp.156-164, Jan. 2013.

[11] Y. Kim and L.-S. Kim, "64-Bit Carry-Select Adder with Reduced Area," Electronics Letters, Vol. 37, No. 10, pp. 614–615, May 2001.

[12] B. Srinivasa Ragavan, B. P. Bhuvana and V. S. Kanchana Bhaaskaran, "Low power 64-bit carry select adder using modified exnor block", Journal of Engg. and Applied Sciences, Vol.10, pp. 17294, Dec, 2015.

[13] Samiappa Sakthikumaran, S. Salivahanan, V. S. Kanchana Bhaaskaran, V. Kavinilavu, B. Brindha and C. Vinoth "A Very Fast and Low Power Carry Select Adder Circuit", 3rd International Conference on Electronics Computer Technology - ICECT 2011, Pp. 273-276, April 8 - 10, 2011.

[14] J. M. Rabaey, Digital Integrated Circuits - A Design Perspective. Upper Saddle River, NJ:Prentice-Hall, 2001.

[15] M. M. Mano, Digital Design 3rd ed., Upper Saddle River, NJ:Prentice-Hall, 2002.