ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 1, March 2016

DESIGN OF ADD MULTIPLY OPERATOR USING SUM TO MODIFIED BOOTH ALGORITHM FOR SIGNED AND UNSIGNED BITS

Hariharan.M, Dr.H.Mangalam <br/>
PG Scholar, Department OF ECE, Head Of The Department, Department of ECE<br/>
Sri Krishna College of Engineering and Technology, Kuniamuthur P.O., <br/>
Coimbatore-641008, Tamil Nadu, India. <br/>
hariece38@gmail.com,mangalam@skcet.ac.in

***Abstract*—*DSP applications are widely used in electronic appliances which performs complex arithmetic operations. Enactment of DSP system is based on the verdict of their design of MAC (Multiply accumulator) and AM (Add Multiply) unit which uses only primitive resources. The proposed technique is based on optimizing the design of FAM (Fused Add and Multiply) unit for increasing the performance of DSP processor. A new recoding technique (S-MB) is proposed for direct recoding of sum of two numbers in modified booth form. This proposed S-MB algorithm is simple, structured and can be easily modified for either signed or unsigned numbers. It can also be implemented for odd or even number of bits. This proposed technique yields considerable improvement when compared to existing fused add multiply unit in terms of critical path delay, hardware complexity and power consumption.*

KeyWords: Fused Add Multiply, Multiply Accumulator, Critical Path, Sum to Modified Booth algorithm.

1. INTRODUCTION

1.1 *ARITHMETIC OPERATION IN DSP*

Digital Signal Processing is most widely used in modern consumer electronics for the domain of multimedia, communication etc. Digital signal processing (DSP) is the mathematical manipulation of an information signal to modify or improve it in some way. It is characterized by the depiction of discrete time, discrete frequency, or other discrete domain signals by a categorization of numbers or symbols and the processing of these signals. The goal of DSP is typically to measure, filter and/or poultice continuous real-world analog signals. The first step is usually to convert the signal from an analog to a digital form, through sampling and then digitizing it using an analog-to-digital converter (ADC), which cracks the analog signal into a stream of numbers.

However, often, the prerequisite output signal is another analog output signal, which requires a digital-to-analog converter (DAC). Even if this process is more multifaceted than analog processing and has a discrete value range, the bid of computational power to digital signal processing provides many advantages over analog processing in many applications, such as error revealing and correction in transmission as well as data compression. Typical DSP applications carry out a large quantity of arithmetic operations as their implementation is based on computationally intensive kernels, As expected, the enactment of DSP systems is inherently affected by decisions on their design regarding the allocation and the design of arithmetic units.Recent research activities illustrate substantial performance improvements in the field of arithmetic optimization. The author in [1], [11] have shown that the design of arithmetic components combining manoeuvres which share data, can lead to significant performance improvements. Multipliers play an important role in today's DSP and numerous other applications. Based on the observation that an addition can often be subsequent in multiplication (e.g.,symmetric FIR filters), the Multiply-Accumulator (MAC) and Multiply-Add (MAD) units were hosted [3] leading to more effective implementations of DSP algorithms compared to the conventional ones, which use only primitive resources[8]. With improvements in technology, many researchers have tried and are trying to design multipliers which offer either of the succeeding design targets – high speed, low power consumption, regularity of layout and hence less area or even arrangement of them in one multiplier thus making them suitable for various high speed, low power and compact VLSI implementation.

All Rights Reserved © 2016 IJARTET

Add and Shift algorithm is the mutual multiplication method . Several architectures have been proposed to optimize the enactment of the MAC operation in area, critical path delay or power consumption [5]. As noted in, MAC components escalation the elasticity of DSP data path synthesis as a large set of arithmetic operations can be efficiently mapped onto them. In parallel multipliers number of partial products to be added is the main parameter that determines the performance of the multiplier.

Except the MAC/MAD operations, many DSP applications are based on Add-Multiply operations (e.g. FFT algorithm). The candid design of the AM unit, by first allocating an adder and then output of the AM unit is given to multiplier input, increases significantly both area and critical path delay of the circuit. Depending on the necessities of application, DSP is implemented on general purpose computers or embedded processors that may include specific microprocessors called digital signal processors. Targeting an optimized design of AM operators,[2] fusion techniques are engaged based on the direct recoding of the addition of two numbers in its Modified Booth (MB) form[7].Thus, the carry-propagate adderor carry-look-ahead adder of the conventional AM design is excluded resulting in considerable gains of performance.

1.3 *BOOTH MULTIPLIER*

A special expansion of the pre-processing step of the recoder is needed in order to handle operands in carry-save representation. Christo Ananth et al. [6] proposed a system which contributes the complex parallelism mechanism to protect the information by using Advanced Encryption Standard (AES) Technique. AES is an encryption algorithm which uses 128 bit as a data and generates a secured data. In Encryption, when cipher key is inserted, the plain text is converted into cipher text by using complex parallelism. Similarly, in decryption, the cipher text is converted into original one by removing a cipher key. The complex parallelism technique involves the process of Substitution Byte, Shift Row, Mix Column and Add Round Key. The above four techniques are used to involve the process of shuffling the message. The complex parallelism is highly secured and the information is not broken by any other intruder.

This work focuses on the efficient design of FAM operators, aiming the optimization of the recoding structure for direct modelling of the MB form into sum of two numbers (Sum to MB – S-MB). More specifically, A novel recoding method is anticipated which decreases the critical path delay and condenses area and power consumption.

The projected*S-MB* algorithm is structured, simple and can be easily modified in order to be smeared either in signed or unsigned numbers, which embrace of odd or even number of bits. Three alternative structures of the proposed *S-MB* approach by means of conventional and signed-bit Full Adders and Half Adders as structure blocks are also proposed.

2.EXISTING METHODS

2.1 *CONVENTIONAL DESIGN OF THE AM OPERATOR*

In this paper, we focus on AM units which implement the operation. The conventional design of the AM operator involves that its inputs are first driven to an adder and then the input and the sum are driven to a multiplier. The drawback of using an adder is that it inserts a significant delay in the vital path of the AM. As there are carry signals to be propagated inside the adder, the critical path depends on the bit-width of the inputs. To facilitate decline in the delay, a Carry-Look-Ahead (CLA) adder can be used which stillraises the area occupation and power dissipation. An optimized design of the AM operator is based on the fusion of the adder and the MB encoding unit into a solitary data path block by direct recoding of the sum to its MB representation.

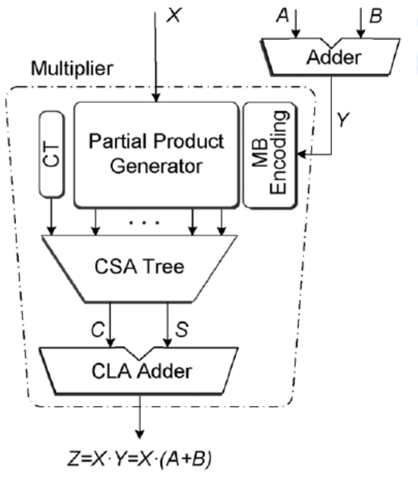


Figure 2.1: AM operator based on conventional design

The fused Add-Multiply (FAM) component includes just one adder at the end (final adder of the parallel multiplier). As a result,

All Rights Reserved © 2016 IJARTET

significant area savings are experiented and the critical path delay of the recoding process is reduced and decoupled as of the bit-width of its inputs. In this work, we present a new technique for direct recoding of two numbers in the MB representation of their sum is presented.

2.2 *MODIFIED BOOTH FORM*

Modified Booth (MB) is a customary form used in multiplication. It is a redundant signed-digit radix-4 encoder coding technique. Its main benefit is that it diminishes by half the number of partial products in multiplication comparing to any other radix-2 demonstration. Let us consider the multiplication of 2's complement numbers x and y with each number consisting of n = 2k bits. The multiplicand y can be signified in MB form as:

$$Y = \langle y_{n-1} y_{n-2} \cdots y_1 y_0 \rangle_{2's} = -y_{2k-1} \cdot 2^{2k-1} + \sum_{i=0}^{2k-2} y_i \cdot 2^i$$

$$= \langle \mathrm{y}_{k-1}^{MB} \mathrm{y}_{k-2}^{MB} \cdots \mathrm{y}_1^{MB} \mathrm{y}_0^{MB} \rangle_{MB} = \sum_{j=0}^{k-1} \mathrm{y}_j^{MB} \cdot 2^{2j} \quad (1)$$

$$\mathrm{y}_j^{MB} = -2y_{2j+1} + y_{2j} + y_{2j-1}. \quad (2)$$

$$\mathrm{y}_j^{MB} \in \{-2, -1, 0, +1, +2\}, 0 \leq j \leq k-1$$

Digits correspond to the three consecutive bits y_{2j+1}, y_{2j} and y_{2j-1} with one bit overlapped and considering that $y_{-1} = 0$. Table 1 shows how they are formed by summarizing the MB encoding technique.

Table 2.1: MODIFIED BOOTH ENCODING TABLE

Binary			y_j^{MB}	MB Encoding			Input carry
0	0	0	0	0	0	0	0
0	0	1	+1	0	1	0	0
0	1	0	+1	0	1	0	0
0	1	1	+2	0	0	1	0
1	0	0	-2	1	0	1	1
1	0	1	-1	1	1	0	1
1	1	0	-1	1	1	0	1
1	1	1	0	1	0	0	0

Each digit is represented by three bits named,*one* and *two*. The sign bit shows if the digit is negative (s = 1) or positive (s = 0) Signal *one* shows if the supreme value of a digit is equal to 1 or not . Signal *two* shows if the absolute value of a digit is equal to 2 or not. Using these three bits we calculate the MB digitsby the following relation:

$$\mathrm{y}_j^{MB} = (-1)^{s_j} \cdot [one_j + 2 \cdot two_j]. \quad (3)$$

The Boolean equations of the MB encoding signals are as given below and the gate level schematic is shown in figure 2.2

$$one_j = y_{2j-1} \oplus y_{2j}$$

$$two_j = (y_{2j+1} \oplus y_{2j}) \cdot \overline{one_j}$$

$$s_j = y_{2j+1}$$

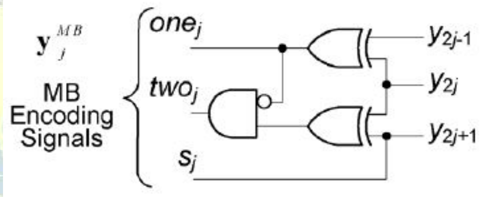


Fig. 2.2: gate-level schematic for the implementation of the MB encoding signals

2.3 *FAM IMPLEMENTATION*

In the FAM design presented in Fig.3.3, the multiplier is a parallel one based on the MB algorithm. Let us consider the product x and y . The term y is encoded based on the MB algorithm and multiplied with x.

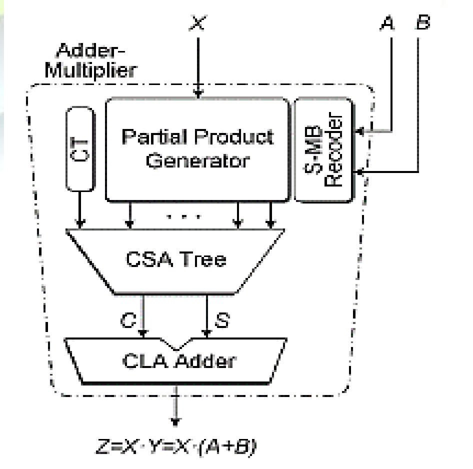


Fig 2.3: AM operator based on the fused design with direct recoding

All Rights Reserved © 2016 IJARTET

Both x and y consist of n = 2k bits and are in 2's match form. Equation describes the generation of the partial products:

$$PP_j = X \cdot \mathbf{y}_j^{MB} = \overline{p}_{j,n}2^n + \sum_{i=0}^{n-1} p_{j,i} \cdot 2^i. \qquad (4$$

The creation of the i-th bit of the partial product is based on the next logical expression and Fig. 2.4 illustrates its implementation at gate level.

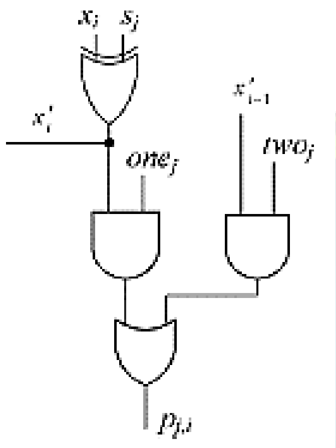


Fig. 2.4: Generation of the i-th bit of the partial product pp$_j$

For the computation of the least and the most significant bits of the partial product we consider respectively $x_{-1} = 0$ and $x_n = x_{n-1}$

$$p_{j,i} = ((x_i \oplus s_j) \wedge one_j) \vee ((x_{i-1} \oplus s_j) \wedge two_j). \quad (5)$$

Note that in case that n = 2k+1 the number of the resulting partial products is [n/2] + 1 = k + 1 and the most significant MB digit is created based on sign extension of the initial 2's complement number. After the partial products are produced, they are added, properly weighted, through a Wallace Carry-Save Adder (CSA) tree with the Correction Term (CT) which is known by the following equations:

$$Z = X \cdot Y = CT + \sum_{j=0}^{k-1} PP_j \cdot 2^{2j} \qquad (6)$$

$$CT = CT(low) + CT(high) = \\ = \sum_{j=0}^{k-1} c_{\text{in},j} \cdot 2^{2j} + 2^n \left(1 + \sum_{j=0}^{k-1} 2^{2j+1}\right) \qquad (7)$$

Finally, the carry-save output of the Wallace CSA tree is fed to a fast Carry Look Ahead (CLA) adder to form the final result X = Z . Y as shown in Fig. 2.3.

3.PROPOSED METHOD

3.1*DEFINING SIGNED-BIT FULL ADDERS AND HALF ADDERS FOR STRUCTURED SIGNED ARITHMETIC*

In *S-MB* recoding technique, recoding of the addition of two consecutive bits of the input A(a2j, a2j + 1) with two consecutive its of the input B(b2j, b2j + 1) into one MB digit is done.Three bits are included in forming a MB digit. The most significant bit of them is negatively weighted while the two least significant bits of them have positive weight. Consequently, in order to alter the two aforementioned pairs of bits in MB form, we need to use signed-bit arithmetic. For this purpose, we extend a set of bit-level signed Half Adders (HA)and Full Adders (FA) considering their inputs and outputs to be signed.

More exclusively, in this work, we use two types of signed HAs which are referred as HA* and HA**. Tables 3.1 – 3.3 are their truth tables and in Fig.3.1a and 3.1b we present their corresponding Boolean equations.

HA*

$$c = p \vee q$$
$$s = p \oplus q$$

Fig 3.1a: Boolean equations and schematics for signed HA*

Considering that p,q are the binary inputs and c,s are the outputs (carry and sum correspondingly) of a HA* which realizes the relation where the sum is considered negatively signed. the output takes one of the values of a HA* which implements the relation 2c – s = p + q. where the sum is considered negatively signed the output takes one of the values {0, +1, +2}.

All Rights Reserved © 2016 IJARTET

Table. 3.1: HA* Basic Operation

Inputs		Output value	outputs	
P(+)	Q(+)		C(+)	S(-)
0	0	0	0	0
0	1	+1	1	1
1	0	+1	1	1
1	1	+2	1	0

In Table 3.2, the dual implementation of HA* is described in which the signs of all inputs and outputs ae reversed and the corresponding outputs are given.

Table. 3.2 HA* dual operation

Inputs		Output value	outputs	
P(+)	Q(+)		C(+)	S(-)
0	0	0	0	0
0	1	-1	1	1
1	0	-1	1	1
1	1	-2	1	0

Table 3.3 gives the truth table of HA** basic operation and Fig. 3.1(b) show the schematic of HA** which implements the relation 2 . c – s = - p + q. and influences a negative (p) and a positive (q) input resulting in the output values { -1, 0, +1}.

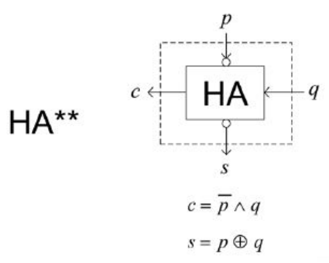


Fig 3.1b: Boolean equations and schematics for signed HA**

Table 3.3 HA** basic operation

Inputs			Output value	outputs	
P(+)	Q(+)	C(+)		C(+)	S(-)
0	0	0	0	0	0
0	0	1	+1	1	1
0	1	0	-1	0	1
0	1	1	0	0	0
1	0	0	+1	1	1
1	0	1	+2	1	0
1	1	0	0	0	0
1	1	1	+1	1	1

Two types of signed FAs are designed whose truth table are presented in Table 3.4 and 3.5.Fig 3.2 shows their schematic.

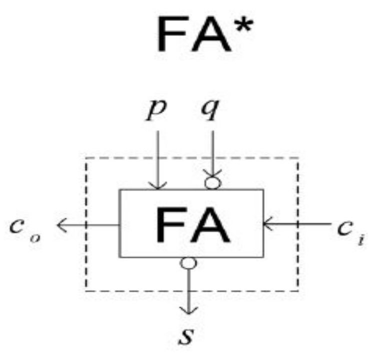
Table 3.4: FA* Operation

Inputs			Output Value[1]	Outputs	
P(+)	Q(+)	C(+)		C(+)	S(+)
0	0	0	0	0	0
0	0	1	+1	1	1
0	1	0	-1	0	1
0	1	1	0	0	0
1	0	0	+1	1	1
1	0	1	+2	1	0
1	1	0	0	0	0
1	1	1	+1	1	1

Table 3.5 : FA** Operation

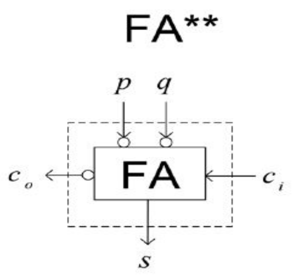
Inputs			Output values	outputs	
P(-)	Q(-)	C_i(+)		C_o(-)	S(+)
0	0	0	0	0	0
0	0	1	+1	0	1
0	1	0	-1	1	1
0	1	1	0	0	0
1	0	0	-1	1	1
1	0	1	0	0	0
1	1	0	-2	1	0
1	1	1	-1	1	1

The schematics drawn in Fig. 3.2(a) and Fig 3.2(b) show the relation of FA* and FA** with the conventional FA. Assuming that p,q and c_i are the binary inputs and c_0,s are the output carry and sum respectively, FA* implements the relation $2c_o – s = p – q + c_i$ where the bits s and q are considered negatively signed (Table 3.4, Fig. 3.1(a)).



$$c_o = ((p \vee \overline{q}) \wedge c_i) \vee (p \wedge \overline{q})$$
$$s = p \oplus q \oplus c_i$$

Fig . 3.2a: Boolean equations and schematics for signed FA*



$$c_o = ((p \vee q) \wedge \overline{c}_i) \vee (p \wedge q)$$
$$s = p \oplus q \oplus c_i$$

Fig . 3.2b: Boolean equations and schematics for signed FA**

Table 3.4 (truth table of FA*) shows that the output values of FA* are {-1,0,+1,+2}. In the case of FA**, the two inputs p,q are negatively signed and FA** realizes the relation $-2 . c_o - s = -p - q + c_i$ (Table 3.4, Fig. 3.2(b)). The output values become {-2, -1, 0, +1} . As shown in Fig. 3.2, the signed FAs are implemented using the conventional FA with the negative inputs and outputs inverted.

3.2 *PROPOSED S-MB RECODING TECHNIQUES*

Both conventional and signed HAs and FAs in order to design and explore three new alternative schemes of the *S-MB* recoding technique. Each of the three methods can be simply applied in either signed (2's complement representation) numbers which consist even number of bits. In all schemes it is considered that both inputs A and B are in 2's complement form and consist of 2k bits in case of even. Targeting to transform the sum of A and B (Y = A + B) in its MB representation ,The bits a_{2j}, a_{2j+1} and b_{2j}, b_{2j+1} are considered as the inputs of the recoding cell in order to get the output.

3.2.1 *S-MB1 RECODING SCHEME*

The first method of the projected recoding technique is consigned as *S-MB*1 and is illustrated in detail in Fig.3.3 for even bit-width of input numbers. As seen in Fig. 3.3 the sum of A and B is given by the relation:

$$Y = A + B = \mathbf{y}_k \cdot 2^{2k} + \sum_{j=0}^{k-1} \mathbf{y}_j^{MB} \cdot 2^{2j}$$
$$\text{where} \quad \mathbf{y}_j^{MB} = -2s_{2j+1} + s_{2j} + c_{2j}. \qquad (8)$$

The encoding of the MB digits is based on the analysis of Section 3.2 Both bits s_{2j+1} and s_{2j} are extracted from the recoding cell of Fig. 3.3.

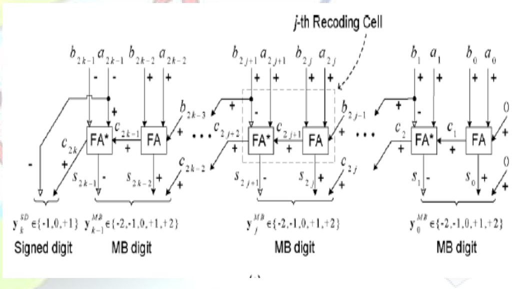


Fig 3.3: *S-MB*1 recoding scheme for even number of bits

A conventional FA with inputs a_{2j},b_{2j} and b_{2j-1} produces the carry and the sum

$$c_{2j+1} = (a_{2j} \wedge b_{2j}) \vee (b_{2j-1} \wedge (a_{2j} \vee b_{2j}))$$

$$s_{2j} = a_{2j} \oplus b_{2j} \oplus b_{2j-1}$$

As the bit needs s_{23+1} to be negatively signed,a FA* (Table 3.4, Fig. 3.2(a)) is added with inputs a_{2j+1} , b_{2j+1} (-) and c_{2j+1} which produces the carry c_{2j+2} and the sum s_{2j+1}(-).

$$c_{2j+2} = (a_{2j+1} \wedge \overline{b}_{2j+1}) \vee (c_{2j+1} \wedge (a_{2j+1} \vee \overline{b}_{2j+1}))$$
$$s_{2j+1} = a_{2j+1} \oplus b_{2j+1} \oplus c_{2j+1}. \qquad (9)$$

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 1, March 2016

The FA* as negatively signed while it is also used with positive sign as an input carry of the successive recoding cell.The preliminary values $b_{-1} = 0$ and $c_0 = 0$ is considered.The most significant digit (MSD) of

$$c_{2j+2,2} = c_{2j+1} \wedge \overline{(a_{2j+1} \oplus b_{2j+1})}$$
$$s_{2j+1} = a_{2j+1} \oplus b_{2j+1} \oplus c_{2j+1}. \qquad (15)$$

the*S-MB*1 recoding scheme is formed for the even bit-width of A and B Fig 4.3, In this case, the MSD is a signed digit and is specified by the next algebraic equation:

$$y^{SD}_{k,even} = -a_{2k-1} + c_{2k}. \qquad (10)$$

The critical path delay of *S-MB*1 recoding scheme (Fig. 3.2) is constant in respect to the input bit-width and is specified by the equation:

$$T_{S-MB1} = T_{FA,carry} + T_{FA^*,sum} \qquad (11)$$

3.2.2 *S-MB2 RECODING SCEME*

The second approach of the proposed recoding technique, *S-MB*2, is described in Fig.3.4 for even bit-width of input numbers. The initial values $c_{0,1} = 0$ and $c_{0,2} = 0$ is considered, The digit $y_j^{MB}, 0 \leq j \leq k-1$ are formed based on s_{2j+1}, s_{2j} and $c_{2j,2}$.As in the *S-MB*1 recoding scheme, conventional FA to produce the carry c_{2j+1} and the sum s_{2j} . The inputs of the FA are a_{2j}, b_{2j} and $c_{2j,1}$. The bit $c_{2j,1}$ is the output carry of a conservative HA which is piece of the (j - 1) recoding cell and has the bits a_{2j-1},b_{2j-1} as inputs.

Fig . 3.4 : *S-MB*2 recoding scheme for even number of bits.

The bit s_{2j+1} is the output sum of a HA* (basic operation – Table 3.1, Fig. 3.1(a)) in which c_{2j+1} and the sum produced by a conventional HA with the bits a_{2j+1},b_{2j+1} as inputs.

The HA* is used in order to produce the negatively signed sum s_{2j+1} and its outputs are given by the following Boolean equations:

$$c_{2j+2,2} = c_{2j+1} \vee (a_{2j+1} \oplus b_{2j+1})$$
$$s_{2j+1} = a_{2j+1} \oplus b_{2j+1} \oplus c_{2j+1}. \qquad (12)$$

In case that A and B comprise of even number of bits (Fig. 4.3), a_{2n-1} and b_{2n-1} are negatively weighted and the conventional HA of the (n -1) recoding cell is replaced by the dual HA* analyzed in Table 3.2. The MSD is a signed digit and is given by the relation:

$$y^{SD}_{k,even} = -c_{2k,1} + c_{2k,2}. \qquad (13)$$

The critical path delay of *S-MB*2 recoding scheme is calculated as follows:

$$T_{S-MB2} = T_{HA,carry} + T_{FA,carry} + T_{HA^*,sum} \qquad (14)$$

where $T_{HA,carry}$ and $T_{FA,carry}$ are the delays of shaping the output carry of a conservative HA and FA respectively and $T_{HA^*,sum}$ the delay of forming the sum of a signed HA*.

3.2.3 *S-MB3 RECODING SCHEME*

The third structureexecuting the projected recoding method is *S-MB*3. It isillustrated in detail in Fig. 3.5 for even bit-width of input numbers. consider that $c_{0,1} = 0$and $c_{0,2} = 0$.The digits $y_j^{MB}, 0 \leq j \leq k-1$, based on s_{2j+1}, s_{2j} and $c_{2j,2}$. The bit is now the output carry of a HA* (basic operation – Table 3.1,Fig. 3.1(a)), which belongs to the (j-1) recoding cell and has the bits a_{2j-1},b_{2j-1} as inputs. The negatively signed bit is produced by a HA** (Table 3.3, Fig. 3.1(b)) in which the output sum (negatively signed) of the HA* of the recoding cell with the bits a_{2j+1},b_{2j+1} as inputs.

All Rights Reserved © 2016 IJARTET

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 1, March 2016

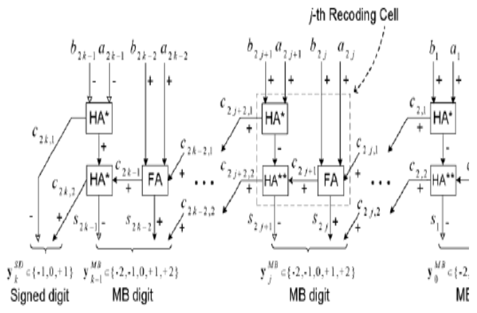


Fig. 3.5. *S-MB*3 recoding scheme for even number of bits.

The carry and sum outputs of the HA** are specified by the nextBoolean equations:

In case that both A and B consists of even number of bits (Fig.3.3), a_{2k-1} and b_{2k-1} are negatively weighted and the dual implementation of the HA*Table 3.2, Fig. 4.1(a) in the (k-1) recoding cell. Thus, the production sum of the HA* turn into positively weighted and the HA** that in detail in Fig.3.6 for even bit-width of input numbers.

follows has to be substituted with a HA*. The most significant digits for even bit-width of A and B, are formed as in *S-MB*2 recoding method.

$$c_{2j+2,2} = c_{2j+1} \wedge \left(\overline{a_{2j+1} \oplus b_{2j+1}}\right)$$
$$s_{2j+1} = a_{2j+1} \oplus b_{2j+1} \oplus c_{2j+1}. \qquad (15)$$

The important path delay of *S-MB*3 recoding scheme is calculated as follows:

$$T_{S-MB3} = T_{HA^*,carry} + T_{FA,carry} + T_{HA^{**},sum} \qquad (16)$$

Where $T_{HA^*,carry}$ and $T_{FA,carry}$ are the delays of shaping the output carry of a signed HA* and FA respectively and $T_{HA^{**},sum}$ is the delay of forming the sum of a signed HA**.

3.2.4 S-MB 2 RECODING FOR UNSIGED BITS

The first method of the projected recoding technique is consigned as *S-MB*1 and is illustrated

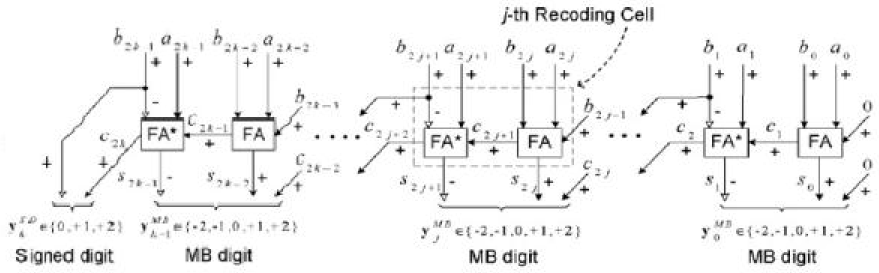


Fig. 3.6: Implementation of the MSD of the *S-MB*1 recoding scheme in case of unsigned input numbers for even

3.2.4 S-MB 2 RECODING FOR UNSIGED BITS

The first method of the projected recoding technique is consigned as S-MB1 and is illustrated in detail in Fig.3.7 for even bit-width of input numbers

In case that the input numbers and are unsigned, their most significant bits are positively signed.

All Rights Reserved © 2016 IJARTET

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 1, March 2016

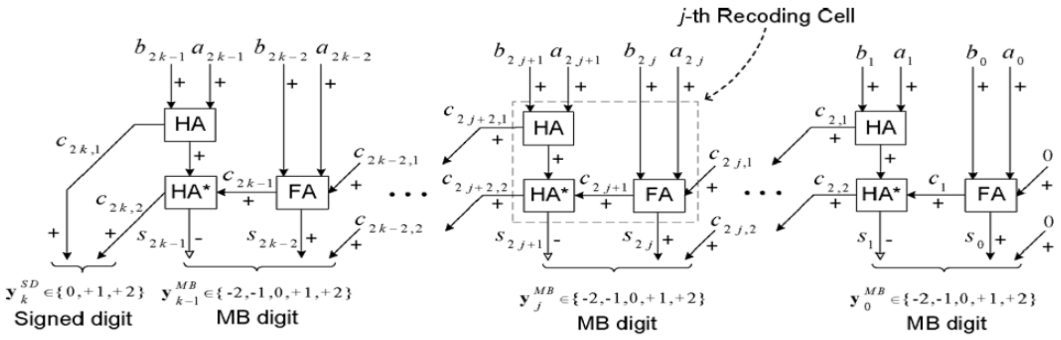


Fig. 3.7: Implementation of the MSD of the *S-MB*1 recoding scheme in case of unsigned input numbers for even

3.2.4 S-MB 3 RECODING FOR UNSIGED BITS

The first method of the projected recoding technique is consigned as S-MB3 and is illustrated in detail inFig.3.8 for even bit-width of input numbersIn case that the input numbers and are unsigned, their most significant bits are positively signed.

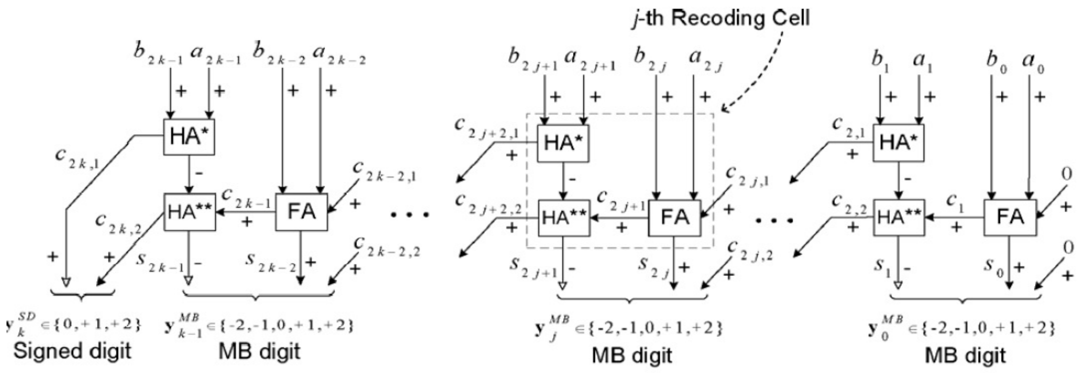


Fig. 3.8: Implementation of the MSD of the *S-MB*1 recoding scheme in case of unsigned input numbers for even

4. RESULT

The synthesis result are done using XILINX ISE.

Table 4.1 Power and Frequency Analysis

POWER (mW)	FREQUENCY			
	500 MHZ	600 MHZ	800 MHZ	1000 MHZ
EXISTING	96	109	134	159
S MB 1	90	101	124	147
S MB 2	93	105	128	152
S MB 3	88	99	121	143

Table 4.2 reveals the percentage power reduction of proposed technique when compared to the existing technique. It can be observed that S-MB 3 technique provides a higher reduction in power compared to other technique

Table 4.2 percentage power reduction

POWER (mW)	FREQUENCY			
	500 MHZ	**600 MHZ**	**800 MHZ**	**1000 MHZ**
S MB 1	6.14%	7.32%	7.86%	7.92%
S MB 2	3.24%	3.42%	4.12%	4.52%
S MB 3	8.18%	10.02%	10.17%	10.51%

All Rights Reserved © 2016 IJARTET

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 1, March 2016

Table 4.3 Area and Delay Calculation

	TIME DELAY (ns)	TOTAL EQUIVALENT GATE COUNT
EXISTING	14.128	2255
S MB 1	5.123	2189
S MB 2	3.797	2153
S MB 3	5.128	2171

Table 4.3 shows the area and delay calculation for existing and proposed technique

5. CONCLUSION

This project focuses on optimizing the design of the Fused-Add Multiply (FAM) operator.A structured technique for the direct recoding of the sum of two numbers in MB form is proposed.Three alternative designs of the proposed *S-MB* recoder are implemented and compared with the existing ones. The proposed recoding schemes, when they are incorporated in FAM designs, yield considerable performance improvements in comparison with the most efficient recoding schemes found in literature.

FUTURE WORK

The Proposed S-MB algorithm can be implemented in variable latency speculating booth and further it will reduce the time delay in partial product. Future it is applied with filters.

REFERENCES

[1] A. Amaricai, M. Vladutiu, and O. Boncalo, "Design issues and implementations for floating-point divide-add fused," IEEE Trans. Circuits Syst. II–Exp. Briefs, vol. 57, no. 4, pp. 295–299, Apr. 2010.

[2] J. D. Bruguera and T. Lang, "Implementation of the FFT butterfly with redundant arithmetic," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 43, no. 10, pp. 717–723, Oct. 1996.

[3] J. J. F. Cavanagh,Digital Computer Arithmetic. NewYork:McGraw- Hill, 1984.

[4] L.-H. Chen, O. T.-C. Chen, T.-Y.Wang, and Y.-C. Ma, "A multiplication-accumulation computation unit with optimized compressors and minimized switching activities," in Proc. IEEE Int, Symp. Circuits and Syst., Kobe, Japan, 2005, vol. 6, pp. 6118–6121.

[5] O. Kwon, K. Nowka, and E. E. Swartzlander, "A 16-bit by 16-bitMAC design using fast 5: 3 compressor cells," J. VLSI Signal Process. Syst., vol. 31, no. 2, pp. 77–89, Jun. 2002.

[6] Christo Ananth, H.Anusuya Baby, "High Efficient Complex Parallelism for Cryptography", IOSR Journal of Computer Engineering (IOSR-JCE), Volume 16, Issue 2, Ver. III (Mar-Apr. 2014), PP 01-07

[7] O. L. Macsorley, "High-speed arithmetic in binary computers," Proc. IRE, vol. 49, no. 1, pp. 67–91, Jan. 1961.

[8] S. Nikolaidis, E. Karaolis, and E. D. Kyriakis-Bitzaros, "Estimation of signal transition activity in FIR filters implemented by a MAC architecture," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 19, no. 1, pp. 164–169, Jan. 2000.

[9] B. Parhami, Computer Arithmetic: Algorithms and Hardware Designs. Oxford: Oxford Univ. Press, 2000.

[10] A. Peymandoust and G. de Micheli, "Using symbolic algebra in algorithmic level DSP synthesis," in Proc. Design Automation Conf., Las Vegas, NV, 2001, pp. 277–282.

All Rights Reserved © 2016 IJARTET