Low Power Cubic Computation Unit Using Vedic Sutras

P Aatmica, V Ranjith, I Nimalsurya and Ramesh S R
Department of Electronics and Communication Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, India
E-Mail : aatmica1998@gmail.com, ajitranjith98@gmail.com, nimalsurya1903@gmail.com, sr_ramesh@cb.amrita.edu

Abstract. In this era of ever blooming technology, devices are developed to be more compatible and handy for the usage of people. Diminishing on-chip power has been the major concern in this nanotechnology period. Very-large-scale integration (VLSI) is the practice of developing an integrated circuit by combining several hundreds of transistors or other devices in a single chip. VLSI allows numerous functions to be added on in a single chip. Each process has inherently higher dynamic and leakage currents. Power dissipation of devices is the main concern when it applies to portable devices. Having high on-chip power degrades the devices reliability and its lifetime. There are various methods of reducing power usage by employing different algorithms, architectures, circuit logics and technology like threshold reduction. This work presents a cubic architecture designed using Vedic sutras from Vedic mathematics. It is compared with conventional architecture. Square of a binary number can be determined by using Dwanda yoga sutra of Vedic mathematics. It focuses at low power design of the architecture. This work shows that the proposed architecture has 21.53% and 46.56 % decrease in area and power respectively.

1. Introduction
The squaring and cubing operations find its application in many areas of digital signal processing like discrete Fourier transform, convolution, digital filters and so on. The squaring and cubing operations in the DSP are commonly performed using the conventional multipliers. The hardware implementation of these multipliers leads to a complex architecture which requires more gates, hence more area and more power. The conventional multipliers also have a large delay. To overcome these shortcomings, algorithm is designed using the Vedic multipliers.

Vedic mathematics is a known to be a good alternative to conventional mathematics. Vedic mathematics basically consists of 16 sutras and all the operations can be performed using those sutras. The 16 sutras are elaborated in [1]. In [2], an efficient parallel multiplier and accumulator (MAC) unit based on Vedic mathematics is presented. Vedic mathematics utilizes the Urdhva-tiryagbhym sutra for the multiplier design. Vedic mathematics simplifies the complex calculations by employing simple techniques. The multiplication using these sutras was implemented in 8085 and 8086, and it resulted in appreciable saving in processing time [3]. One of the sutras for cubing operation is Anurupya sutra, and the implementation of the sutra with two different architectures, one array structured and one tree structured has resulted in less area, delay and power [4]. Another sutra for cubic computation is Yavadunam sutra, which converts the cube of large number into cube of smaller number and addition operation [5]. Conventional approach for squaring and cubing is presented in [6]. An efficient booth multiplier has been proposed in [7].

The squaring operation is done using the Dwanda Yoga sutra which uses the Duplex property. This sutra has been implemented in [8] and it has shown that the carry propagation delay is greatly reduced. The Urdhva-tiryakbyham sutra is used to multiply any two numbers and it has been shown that the vertical critical delay is reduced when compared to the conventional multipliers [9].
The paper proposes an efficient cubic architecture based on the Dwanda Yoga sutra and Urdhva-tiryakbhyam sutra. The Dwanda Yoga sutra is used for the squaring of the given number and the result is multiplied again with the given number using Urdhva-tiryakbhyam sutra which gives the cube of the number.

The rest of the paper is organized as follows: Section II provides the basic overview of Vedic mathematics. Section III discusses about the conventional approach for the cubic architecture. In section IV, an overview of methodology of this work is provided. Section V provides us the results and section VI draws the conclusion.

2. Vedic Mathematics
The Vedic mathematics is the system of mathematics employing from ancient method of mathematics with calculations using unique methodologies and techniques which provides an easy solution for any type, be it an algebra, trigonometry, geometry, arithmetic and many mathematical problems. A scholar of Sanskrit, philosophy, history and mathematics, Sri Bharati Krishna Tirthaji (1884-1960), who rediscovered the mathematical formulae from the ancient Indian scripture which formed the base for the Indian Vedic mathematics to be rebuilt and reused for the calculations which is more faster than the normal calculation systems that are available. He gave us sixteen formulae known as sutras.

2.1 Vedic Sutras
The Sutras used for squaring are Yavadunam sutra, EkadhikenaPurvena sutra, and Dwanda Yoga sutra. The sutras used for cubing are Yavadunam and Anurupya.

2.2 Vedic Multiplier
Multipliers are used for the purpose of the multiplication in the circuits where it greatly reduces the computation required for operation and does the summation of the partial products. Of all the basic arithmetic operations we know, multiplications takes more time and consumes more power. There are many types of multipliers such as array multiplier, Wallace tree multiplier, carry save adder multipliers and so on. But for wide bit multiplications require Vedic multipliers for faster computation.

3. Conventional Approach
The squaring and multiplication is done by using conventional multipliers employing conventional mathematics from the beginning of the engineering stream. Architectures have been designed and employed to perform the binary squaring and cubing using the microprocessors along with basic conventional binary arithmetic operations. But the major drawback in this approach is that the processing speed is low, area is large and consumes more power. Power consumption is more for wider bit calculations and also the processing speed decreases with wider bits of calculations. Low power design are to be employed for saving power consumption.
Table 1. Conventional Cube Using Conventional Multiplication

| Cube of 8 bit number | Calculations Performed |
|----------------------|------------------------|
| **Multiplicand** =   | 1 1 1 1 1 1 1 1        |
| (255)                |                        |
| **Multiplier** =     | 1 1 1 1 1 1 1 X        |
| (255)                |                        |

| Intermediate result = | 1 1 1 1 1 1 0 0 0 0 0 0 0 1 |
| (65025)               | 1 1 1 1 1 1 1 1 X          |

| Final result =        | 1 1 1 1 1 0 1 0 0 0 0 0 1 0 1 1 1 1 1 1 1 |
| (16581375)            | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |

4. Methodology
4.1 Dwanda Yoga

The Dwanda Yoga sutra is used for squaring operation and it uses the Duplex (D) property. The duplex of a number is obtained by the sum of the twice the product of the outermost pair and then the next outer pair inner to the outermost pair and so on till zero pairs are left. For the odd number of digits, the middlemost number does not have a pair and in that case, the middlemost number without the pair is squared. Duplex (D) property is explained in the example given below.

\[
\begin{align*}
D(x) &= x^2 \\
D(xy) &= 2x \cdot y \\
D(xyz) &= (2x \cdot z) + y^2 \\
D(wxyz) &= (2w \cdot z) + (2x \cdot y) \\
D(vwxyz) &= (2v \cdot z) + (2w \cdot y) + x^2
\end{align*}
\]

where \( x, xy, xyz, wxyz, vwxyz \) are numbers.

The square of a number can be found using the dwanda yoga as below. For example, the square of the respective numbers say ‘xy’, ‘xyz’, ‘wxyz’ is given by

\[
\begin{align*}
(xy)^2 &= D(x) \mid D(xy) \mid D(y) \\
(xyz)^2 &= D(x) \mid D(xy) \mid D(yz) \mid D(z) \\
wxyz)^2 &= D(w) \mid D(wx) \mid D(wxy) \mid D(wxyz) \mid D(xyz) \mid D(yz) \mid D(z)
\end{align*}
\]

In the example of \((xy)^2\)

Let \( D(x) = A \), \( D(xy) = B \), \( D(y) = C \).
\((xy)^2 = EABC\), each of the duplexes (i.e) A, B, C should contain only one digit. If any of the duplexes has a number with more than one digit, the remaining digits should be carried on to the next duplex and added with it and the process continues. ‘E’ is the carry generated from ‘A’.

4.2 UrdhvaTiryakbhyam Sutra

UrdhvaTiryakbhyam sutra in Vedic mathematics is a general method for multiplying any two numbers. Urdhva-Tiryāk literally translates to ‘vertically and crosswise’. UrdhvaTiryakbhyam is a fast multiplication method where the multiplication of more than one digit is not allowed and the rest of the operations are additions. The steps for multiplication of two three-digit numbers using UrdhvaTiryakbhyam sutra is given below.

Step 1: Vertical multiplication of the rightmost digit of the two numbers.
Step 2: Cross product of the rightmost 2 digits of the two numbers and addition of their products.
Step 3: Cross product of the outermost digits of the two numbers and vertical multiplication of the middle digits of the two numbers and addition of all these products.
Step 4: Cross product of the leftmost 2 digits of the two numbers and the addition of their products.
Step 5: Vertical multiplication of the leftmost digit of the two numbers.

The answer from each of these steps should contain only one digit except the last step. So, the remaining digits should be carried on to the answer of the next step and the process continues.

Example: 153 x 135

| 153 | 2) 5.3 | 3) 1.53 |
| 135 | 3 | 3 |
| (15) | (25 + 9 = 34) | (5 + 3 + 15 = 25) |
| 4) 1.5 | 5) 1 | (3 + 5 = 8) |
| (1) | | |

Figure. 1. Multiplication Using UrdhvaTiryakbhyam

Let the final result be \(P_5 P_4 P_3 P_2 P_1 P_0\)

\[ P_0 = 5 \text{ (carry } \rightarrow 1) ; \]
\[ P_1 = 1 + 34 = 5 \text{ (carry } \rightarrow 3) ; \]
\[ P_2 = 3 + 23 = 6 \text{ (carry } \rightarrow 2) ; \]
\[ P_3 = 2 + 8 = 0 \text{ (carry } \rightarrow 1) ; \]
\[ P_4 = 1 + 1 = 2 \text{ (carry } \rightarrow 0) . \]

So the final result is 20655.

4.3 Algorithm

The algorithm for finding the cube of a four-bit binary number is given below.

Let the four-bit number be A3 A2 A1 A0, say \((1001)_2\). The first part of the computation is finding the square of the number using the Dwanda Yoga sutra as explained below.

\[
D \ (A3) = (A3 \times A3) = A \\
D \ (A3 \ A2) = 2 \times A3 \times A2 = B \\
D \ (A3 \ A2 \ A1) = (2 \times A3 \times A1) + (A2 \times A2) = C \\
D \ (A3 \ A2 \ A1 \ A0) = (2 \times A3 \times A0) + (2 \times A2 \times A1) = D \\
D \ (A2 \ A1 \ A0) = (2 \times A2 \times A0) + (A1 \times A1) = E
\]
\[ D(\text{A1 A0}) = (2 \times \text{A1} \times \text{A0}) = F \]
\[ D(\text{A0}) = \text{A0} \times \text{A0} = G \]

\[ Z = \text{Carry from } D(\text{A3}) \]

\[ \begin{array}{cccccc}
\text{A3} & \text{A2} & \text{A1} & \text{A0} \\
\times & \text{A3} & \text{A2} & \text{A1} & \text{A0} \\
\hline
\text{Z} & \text{A} & \text{B} & \text{C} & \text{D} & \text{E} & \text{F} & \text{G} \\
\hline
\text{P7} & \text{P6} & \text{P5} & \text{P4} & \text{P3} & \text{P2} & \text{P1} & \text{P0} \\
\hline
\text{Final Result} \\
\end{array} \]

**Figure. 2. 4-Bit Multiplication**

The final result is obtained after all the carries from each of the duplexes are added to the next duplex as explained before.

Example: \((1001)_2^2\)

\[
\begin{align*}
D(1) &= 1 \times 1 = 1 = A \\
D(10) &= (10 \times 1 \times 0) = 0 = B \\
D(100) &= (10 \times 1 \times 0) + (0 \times 0) = 0 = C \\
D(1001) &= (10 \times 1 \times 1) + (10 \times 0 \times 0) = 10 = D \\
D(001) &= (10 \times 0 \times 1) + (0 \times 0) = 0 = E \\
D(01) &= (10 \times 0 \times 1) = 0 = F \\
D(1) &= 1 \times 1 = 1 = G \\
\end{align*}
\]

**Figure. 3. Square of a 4-Bit Number Using Dwanda Yoga**

So, \((1001)_2^2 = 01010001\).

The second part of the computation involves the multiplication of the result obtained from the first part and the given number using UrdhvaTiryakbhyam sutra as given below.

\[ \begin{array}{cccccc}
P_7 & P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0 \\
\times & \text{A3} & \text{A2} & \text{A1} & \text{A0} \\
\hline
R_{11} & R_{10} & R_9 & R_8 & R_7 & R_6 & R_5 & R_4 & R_3 & R_2 & R_1 & R_0 \\
\hline
\text{Final Result} \\
\end{array} \]

**Figure. 4. Final Result**
Example : (01010001) x (1001):

\[\begin{array}{c|c}
1 & 01010001 \\
1 & 1001 \hline
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
1 & 00000000 \\
1 & 100010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\
0 & 00000000 \\
0 & 01010001 \\

\end{array}\]

\textbf{Figure. 5.} Multiplication Using UrdhvaTiryakbhyam for Binary Number

The result is as follows,

\[R0 = 1; R1 = 0; R2 = 0; R3 = 1; R4 = 1; R5 = 0; R6 = 0; R7 = 1; R8 = 0; R9 = 1; R10 = 0; R11 = 0.\]

So, \((1001)3 = 00101001101\)

\section*{5 Results and Discussion}

The designs were coded in VHDL language. The simulations were performed using Modelsim Altera. Synthesis of the design was done using Synopsys Design Compiler. All the technology cells were mapped to SAED 90 nm.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
Parameter & Conventional & Vedic \\
\hline
Cells & 781 & 631 \\
\hline
Area (\(\mu\text{m}^2\)) & 8495 & 6666 \\
\hline
Dynamic Power (\(\mu\text{W}\)) & 1140 & 607 \\
\hline
Leakage Power (\(\mu\text{W}\)) & 27.93 & 21.59 \\
\hline
Nets & 797 & 640 \\
\hline
Total Power (\(\mu\text{W}\)) & 1169.6 & 616.68 \\
\hline
\end{tabular}
\caption{Synthesis Results of the 8-bit number}
\end{table}

From the above table it is evident that the proposed cubic computation unit yields a drastic reduction in power and area. Also the number of nets used in the design is reduced.

\section*{6 Conclusion}

In this work, a cubic architecture has been proposed using Vedic mathematics. The proposed architecture has been analysed in terms of area and power. The proposed cubic architecture performs better than conventional architecture in terms of power and area. Hence, the proposed cube architecture may be useful for low power applications, Digital signal processing applications, cryptographic applications, image processing, and so on.
References

[1] Maharaja J S S B K T 2009 Vedic mathematics (Delhi : MotilalBanarsidass Publishers Pvt. Ltd)

[2] Jithin S and Prabhu E 2015 Parallel Multiplier–Accumulator Unit based on Vedic Mathematics in ARPN Journal of Engineering and Applied Sciences 10 pp 3608-3613

[3] Chidgupkar P D and Karad M T 2004 The implementation of Vedic algorithms in digital signal processing Glob. J. Eng. Educ. 8 pp 153–158

[4] Ramalatha M Thanushkodi K Deena Dayalan K et. al. 2009 A novel time and energy efficient cubing circuit using Vedic mathematics for finite field arithmetic Proc. of 09th Int. Conf. on Advances in Recent Technologies in Communication and Computing pp 873–875

[5] Barik R K and Pradhan M2017 Efficient ASIC and FPGA implementation of cube architecture in IET Computers & Digital Techniques 11 pp 43-49

[6] Liddicoat A A and Flynn M 2000 Parallel square and cube computations Proc. Of 34th Int. Conf. on Signals, Systems and Computers 2 pp 1325–1329

[7] DurgaPavan M V and Ramesh S R 2018 An Efficient Booth Multiplier Using Probabilistic Approach Int. Conf. on Communication and Signal Processing (ICCSP), Chennai pp 0365-0368

[8] Kumar B D and Bharathi M 2012 A High Speed And Efficient Design For Binary Number Squaring Using Dwandwa Yoga Int. Journal of Advanced Research in Computer Engineering & Technology, (IJARCET ) 1 pp 476-479

[9] Bansal Y and Charu M 2016 A novel high-speed approach for 16 × 16 Vedic multiplication with compressor adders Comput.Electr. Eng. 49 pp 39–49