Designing a Vedic Multiplier for OFDM Synchronization Using FPGA

V.Deepika*, S.Deva Priya*

*PG student (Applied Electronics) &Electronics and communication engineering &Jayaram College of Engineering and Technology, Trichy, India.
Assistant professor & Electronics and communication engineering & Jayaram College of Engineering and Technology, Trichy, India.

Abstract. This abstract gives the designing of Vedic multiplier for Orthogonal Frequency Division Multiplexing (OFDM) synchronization using Field Programmable Gate Array (FPGA). The designing of Vedic Multiplier is based on a novel technique of digital multiplication which is quite different from the conventional method of multiplication like add and shift. Were smaller blocks are used to design the bigger one. The Vedic Multiplier is designed in Verilog HDL, as its give effective utilization of structural method of modeling. The individual block is implemented using Verilog hardware description language. The functionality of each block is verified using simulation software, Model Sim and ISE.

Keywords— Field Programmable Gate Array (FPGA), IEEE 802.16, Orthogonal Frequency Division Multiplexing (OFDM).

I INTRODUCTION

Orthogonal Frequency-Division Multiplexing (OFDM) is a method of encoding digital data on multiple carrier frequencies. OFDM have developed into a popular scheme for wideband digital communication, whether wireless otherwise over copper wires, used in application such as digital television and audio distribution, DSL Internet access, 4G mobile communications and wireless networks. OFDM is essentially identical to Coded OFDM (COFDM) and Discrete Multi-tone Modulation (DMT), and is a frequency-division multiplexing (FDM) scheme used as a digital multi-carrier modulation method. The word "coded" comes from the use of forward error correction (FEC). A large number of closely spaced orthogonal sub-carrier signals are used to carry data on several parallel data streams or channels. Each sub-carrier is modulate with a conventional modulation scheme (such as quadrature amplitude modulation or phase-shift keying) at a low symbol rate, maintain total data rates similar to conventional single-carrier modulation schemes in the same bandwidth. The primary advantage of OFDM more than single-carrier schemes is its ability to cope with severe channel conditions (for example, attenuation of high frequencies in a long copper wire, narrowband interference and frequency-selective fading due to multipath) without complex equalization filters.

Channel equalization is simplified because OFDM may be viewed as using many slowly modulated narrowband signals rather than one rapidly modulated wideband signal. The low symbol rate make the use of a guard interval between symbols reasonable, making it possible to eliminate inter symbol interference (ISI) and utilize echoes and time-spreading (on analogue TV these are visible as ghosting and blurring respectively) to complete a diversity gain, i.e. a signal-to-noise ratio development. This mechanism and facilitate the design of single frequency networks (SFNs), where several adjacent transmitters send the same signal simultaneously at the same frequency as the signals from multiple distant transmitter may be combined constructively, rather than interfering because would typically occur in a traditional single-carrier system.

Orthogonal frequency division multiplexing (OFDM) is an effective modulation technique used in both wired and wireless communication systems. Particularly, thanks to the advantages of spectral efficiency and robustness to multipath desertion, OFDM was specified for multiple applications in high bit-rate wireless transmission systems such as wireless local area networks adopt by IEEE 802.11 and metropolitan area network in IEEE 802.16d. However, OFDM performance is perceptive to receiver synchronization. Frequency equalizes cause inter-subcarrier interference and error in timing synchronization can lead to inter-symbol interference [2]. Therefore, synchronization is critical for good performance in OFDM systems.

A.Orthogonality

Theoretically, OFDM is a specialized FDM, the additional constraint being the entire carrier signals are orthogonal to each other. In OFDM, the sub-carrier frequencies are chosen so that the sub-carriers are orthogonal to each other meaning that cross-talk between the sub-channels is eliminated and inter-carrier guard bands are not essential. This greatly simplifies the design of both the transmitter and the receiver; unlike conventional FDM a separate filter for each sub-channel is not required.

The orthogonality requires that the sub-carrier spacing is \( \Delta f = \frac{f}{T_u} \) Hertz, where \( T_u \) seconds is the useful symbol duration (the receiver side window size) and \( k \) is a
positive integer, typically equal to 1. Therefore, with N subcarriers the total pass band bandwidth will be \( B = N \Delta f / (\text{Hz}) \).

The orthogonality also allows high spectral efficiency, with a total symbol rate near the Nyquist rate for the equivalent baseband signal (i.e. near half the Nyquist rate for the double-sideband physical pass band signal). Almost the whole available frequency band can be utilized. OFDM generally has a nearly ‘white’ spectrum give it benign electromagnetic interference properties with respect to other co-channel users.

A simple example: A useful symbol duration \( T_0 = 1 \text{ ms} \) would require a sub-carrier spacing of \( \Delta f = \frac{1}{1,000} = 1 \text{ kHz} \) (or an integer multiple of that) for orthogonality. \( N = 1,000 \) sub-carriers would result in a total pass band bandwidth of \( N \Delta f = 1 \text{ MHz} \). For this symbol time, the required bandwidth in theory according to Nyquist is \( N/2T_0 = 0.5 \text{ MHz} \) (i.e., half of the achieved bandwidth required by our scheme). If a guard interval is applied Nyquist bandwidth requirement would be even lower. The FFT would result in \( N = 1,000 \) samples per symbol. If no guard interval was applied, this would result in a base band complex valued signal with a sample rate of 1 MHz, which would need a baseband bandwidth of 0.5 MHz according to Nyquist. However, the pass band RF signal is produced by multiply the baseband signal with a carrier waveform (i.e., double-sideband quadrature amplitude-modulation) resulting in a pass band bandwidth of 1 MHz. A single-side band (SSB) or vestigial sideband (VSB) modulation scheme would achieve almost half that bandwidth for the same symbol rate (i.e., twice as high spectral efficiency for the same symbol alphabet length). It is however more sensitive to multipath interference.

OFDM requires very accurate frequency synchronization between the receiver and the transmitter; with frequency deviation the sub-carriers will no longer be orthogonal, causing inter-carrier interference (ICI) (i.e., cross-talk between the sub-carriers). Frequency offset is typically caused by mismatched transmitter and receiver oscillator or by Doppler shift owing to movement. While Doppler shift alone may be compensated for by the receiver, the situation is worse when combined with multipath, as reflection will appear at various frequency offset, which is a large amount harder to correct. This effect typically worsens as speed increase and is an important factor limiting the use of OFDM in high-speed vehicles. In order to mitigate ICI in such scenarios, one can shape each sub-carrier in order to minimize the interference resulting in a non-orthogonal subcarriers overlapping. For example, a low-complexity scheme referred to as WCP-OFDM (Weighted Cyclic Prefix Orthogonal Frequency-Division Multiplexing) consists in using short filters at the transmitter output in order to perform a potentially non-rectangular pulse shaping and a near perfect reconstruction using a single-tap per subcarrier equalization. Other ICI suppression techniques usually increase drastically the receiver complexity.

One key principle of OFDM is that since low symbol rate modulation schemes (i.e., where the symbols are relatively long compared to the channel time characteristics) suffer less from intersymbol interference caused by multipath propagation, it is beneficial to transmit a number of low-rate streams in parallel instead of a single high-rate stream. Since the duration of every symbol is extended, it is feasible to insert a guard interval between the OFDM symbol thus eliminating the inter symbol interference.

The guard intervals also eliminate the need for a pulse-shaping filter and it reduces the warmth to time synchronization problems.

A simple example: If one sends a million symbols per second using conventional single-carrier modulation over a wireless channel then the duration of each symbol would live one microsecond or less. These impose severe constraints on synchronization and necessitate the removal of multipath interference. If the same million symbols per second are spread with one thousand sub-channel the duration of each symbol can be longer by a factor of a thousand (i.e., one millisecond) for orthogonality with approximately the same bandwidth. Assume that a guard interval of 1/8 of the symbol length is inserted between each symbol. Intersymbol interference can be avoided if the multipath time-spreading (the time between the reception of the first and the last echo) is shorter than the guard interval (i.e., 125 microseconds). This corresponds to a maximum difference of 37.5 kilometres between the lengths of the path.

The prefix, which is transmitted during the guard interval, consists of the end of the OFDM symbol copied into the guard interval and the guard interval is transmitted follow by the OFDM symbol. The reason that the guard interval consists of a copy of the end of the OFDM symbol is so that the receiver will integrate over an integer number of sinusoid cycles for each of the multipath when it performs OFDM demodulation with the FFT. In some standards such as Ultra wideband, in the interest of transmitted power, cyclic prefix is skipped and nothing is sent during the guard interval. The receiver will then have to mimic the cyclic prefix functionality by copying the end part of the OFDM symbol and adding it to the beginning portion.

B. Channel Coding and Interleaving

OFDM is invariably used in conjunction with coding (forward error correction), and almost always uses frequency and/or time interleaving.

Frequency (subcarrier) interleaving increase resistance to frequency-selective channel condition such as fading. For example, when a part of the channel bandwidth fades, frequency interleaving ensures that the bit errors that would result from those subcarriers in the faded part of the bandwidth are spread out in the bit-stream rather than being concentrated. Similarly time interleaving ensure bit that are originally close together in the bit-stream are transmitted far apart in time, thus justifying against severe fading as would happen when travelling at high speed.

However, time interleaving is of little gain in slowly fading channels, such as for stationary reception, and frequency interleaving offers little to no benefit for narrowband channels that suffer from flat-fading (where the whole channel bandwidth fades at the same time).

The reason why interleaving is used on OFDM is to attempt to spread the errors out in the bit-stream that is presented to the error correction decoder because when such decoders are presented with a high concentration of errors the
decoder is unable to correct all the bit errors and a burst of uncorrected errors occur. A similar design of audio data encoding makes compact disc (CD) playback robust.

A classical type of error correction coding used with OFDM-based systems is convolution coding, often concatenated with Reed Solomon coding. Usually, additional interleaving (on top of the time and frequency interleaving mentioned above) in between the two layers of coding is implemented. The choice intended for Reed-Solomon coding as the outer error correction code is based on the observation that the Viterbi decoder used for inner convolution decoding produces short errors bursts when there is a high concentration of errors and Reed-Solomon codes are inherently well-suited to correcting bursts of errors.

Newer systems are usually now adopt near-optimal types of error correction codes that use the turbo decoding principle where the decoder iterates towards the desired solution. Example of such error correction coding types includes turbo cubes and LDPC code which perform close to the Shannon limit for the Additive White Gaussian Noise (AWGN) channel. Some systems that have implemented these code have concatenated them with either Reed-Solomon (for example on the Media FLO system) or BCH codes (on the DVB-S2 system) to improve upon an error floor inherent to these codes at high signal-to-noise ratios.

II PRIOR WORK

A. Design of DSP48E1-Based Correlator

The DSP48E1 Slice inside the Virtex-6 contains a multiplier followed by a configurable arithmetic unit to provide many independent functions, e.g., multiply, multiply-accumulate, multiply-add, three-input add, and more[9]. It also allows the data path to be configured for various input combinations and register stages; a three stage pipeline offers maximum performance.

The coefficients are recomputed according to the IEEE 802.16d standard. The second design supports the complex multiply-adds in a five-stage pipeline consisting of DSP48E1 Slices configured for three-stage internal pipelining. \( R_{i,Re} \) and \( R_{i,Im} \) are the real and imaginary parts of received sample respectively. \( Pr_{Re} \) and \( Pr_{Im} \) similarly represent the complex conjugation of known preamble. The pipeline registers for the \( Pr_{Re} \), \( Pr_{Im} \) are required because they are considered to be of constant value. \( Re \) and \( Im \) are the real and imaginary parts of the previous multiply-\( add \), \( MA_{n-1} \). The output of these complex multiply-adds can be expressed as \( Output = R_{i} Pr_{i} z^{-5} + z^{-4} MA_{n-1} \). It presents the pipeline structure of the correlate. The additional pipeline registers are required for handling the received sample. Adding pipeline registers should improve the performance significantly. The output of the pipelined correlates is

\[
Output = Pr_{[63]} R_{i} (z^{-3})63z^{-5} + z^{-4} \\
\times \left( Pr_{[62]} R_{i} (z^{-3})62z^{-5} + z^{-4} \left( Pr_{[61]} R_{i} (z^{-3})61z^{-5} + \cdots \\
+ z^{-4} \left( Pr_{[0]} R_{i} (z^{-5}) \cdots \right) \right) \right) \\
= (z^{-3})63z^{-5} \left( Pr_{[63]} R_{i} + z^{-1} (Pr_{[62]} R_{i} + \cdots \\
+ z^{-1} (Pr_{[0]} R_{i} \cdots \right)).
\]

B. Design of Multiplier less Correlator

The principle of multiplier less correlators is to represent coefficients and round them in the form of summed powers of 2. Hence, a shift-and-add is performed instead of multiplying by coefficients. It is expected that multiplier less correlation is more efficient, but with embedded hard multipliers in modern FPGAs, it is unclear whether they should still be considered favourable. Furthermore, synchronization accuracy must be considered. To explore this, four alternative multiplier less correlate are implemented using four coefficient sets with increasing degrees of rounding, to compare the cost and performance and evaluate against multiplier-based correlates. The coefficient sets are found by quantizing the 64 normalized preamble samples.
III PRESENT WORK

A. Vedic Multiplier

Vedic mathematics is part of four Vedas. It is part of Sthapatya-Veda which is an upa-veda (supplement) of AtharvaVeda. It covers clarification of several modern mathematical terms including arithmetic, geometry (plane, coordinate), quadratic equations, trigonometry, factorization and even calculus.

His Holiness Jagadguru Shankaracharya Bharati Krishna Teerthaji Maharaja (1884-1960) comprised all this work together and gave its mathematical explanation while discussing it for various applications. Swahiji constructed 16 sutras (formulae) and 16 Upa sutras (sub formulae) after extensive research in Atharva Veda. Observably these formulae are not to be found in present text of Atharva Veda because these formulae were constructed by Swamiji himself. Vedic mathematics is not only a mathematical wonder but also it is logical. That’s why VM has such a degree of eminence which cannot be disapproved. Due these phenomenal characteristic, VM has already crossed the boundaries of India and has become a leading topic of research overseas. VM deals with several basic as well as complex mathematical operations. Especially, methods of basic arithmetic are extremely simple and powerful.

The word „Vedic” is derived from the word ’Veda’ which means the store-house of all knowledge. Vedic mathematics is mainly based on 16 Sutras (or aphorisms) trade with various branches of mathematics like arithmetic, algebra, geometry etc. These method and ideas can be directly applied to trigonometry, plain and spherical geometry, conics, calculus (both differential and integral) and applied mathematics of various kinds. As mentioned earlier all these Sutras were reconstructed from ancient Vedic texts early in the last century. Many Sub-sutras were also discovered at the same time, which are not discussed here.

The beauty of Vedic mathematics lies in the fact that it reduces the otherwise cumbersome-looking calculations in conventional mathematics to very simple one. This is so because the Vedic formulae are claim to be based on the natural principles on which the human mind work. This is a very interesting field and presents some effective algorithms which can be applied to various branches of engineering such as computing and digital signal processing.

The multiplier architecture can be generally classified into three categories. First is the serial multiplier which emphasizes on hardware and minimum amount of chip area. Second is parallel multiplier (array and tree) which carry out high speed mathematical operations. But the drawback is the relatively larger chip area consumption. Third is serial-parallel multiplier which serves as a good trade-off between the time consuming serial multiplier and the area consuming parallel multipliers. The proposed Vedic multiplier is based on the Vedic multiplication formulae (Sutras). These Sutras have been traditionally used for the multiplication of two numbers in the decimal number system.

In this work we relate the same ideas to the binary number system to make the proposed algorithm compatible with the digital hardware.

B. Multiplication Using Urdhva Tiryakbhyam Sutra Algorithm

The multiplier is based on an algorithm Urdhva Tiryakbhyam (Vertical & Crosswise) of ancient Indian Vedic Mathematics. Urdhva Tiryakbhyam Sutra is a common multiplication formula applicable to all cases of multiplication. It literally means “Vertically and crosswise”. It is based on a narrative concept through which the generation of all partial products can be done with the simultaneous addition of these partial products. The parallelism in generation of partial products and their summation is obtained using Urdhava Tiryakbhyam explained in fig 2. The algorithm can be generalized for n x n bit number.

Since the partial products and their sums are calculated in parallel multiplier is independent of the clock frequency of the processor. Therefore the multiplier will require the same amount of time to calculate the product and hence is independent of the clock frequency. The advantage is that it reduces the need of microprocessors to operate at increasingly high clock frequencies. Even as a higher clock frequency generally results in increased processing power its disadvantage is that it also increases power dissipation which outcome in higher device operating temperature. By adopt the Vedic multiplier; microprocessors designers can easily circumvent these problems to avoid catastrophic device failure. The processing power of multiplier can easily be increased by increasing the input and output data bus width since it has a quite regular structure. Due to its regular structure it can be easily layout in a silicon chip. The Multiplier has the advantage as the number of bits increases, gate delay and area increases very slowly as compared to other multipliers. Therefore it is time, power and space efficient. It is demonstrated that this architecture is quite capable in terms of silicon area/speed [4].

To illustrate this multiplication scheme, consider the multiplication of two decimal numbers (325 * 738). Line diagram for the multiplication is shown in Fig 3.1. The digits on the both side of the line are multiplied and added with the carry from the previous step. This generates one of the bits of the result and a carry. This carry is added in the next step and hence the procedure goes on. If more than one line are there in one step all the results are added to the earlier carry. In each step least important bit acts as the result bit and all other bits act as carry for the next step. Initially the carry is taken to be zero. To make the methodology more clearly, an alternate illustration is given with the help of line diagrams in figure3 where the dot represent bit „0” or „1”.
To illustrate the multiplication algorithm, consider the multiplication of two binary numbers $a_3a_2a_1a_0$ and $b_3b_2b_1b_0$. As the result of this multiplication would be more than 4 bits, we communicate it as... $r_3r_2r_1r_0$. Line diagram for multiplication of two 4-bit numbers is shown in Fig 3 which is unknown but the mapping of the Fig 2 in binary system. For the simplicity, each bit is representing by a circle. Least significant bit $r_0$ is obtained by multiplying the least significant bits of the multiplicand and the multiplier. The process is followed according to the steps shown in figure 3.

![Figure 3: Line diagram for multiplication of two 4-bit numbers.](image)

Firstly, least significant bits are multiplied which gives the least significant bit of the product (vertical). Then the LSB of the multiplicand is multiplied with the next higher bit of the multiplier and added with the product of LSB of multiplier and next higher bit of the multiplicand (crosswise). The sum give second bit of the product and the carry is added in the output of next stage sum obtain by the crosswise and vertical multiplication and addition of three bits of the two numbers from least significant position. Next all the four bits are process with crosswise multiplication and addition to give the sum and carry. The sum is the similar bit of the product and the carry is again added to the next stage multiplication and addition of three bits except the LSB. The same operation continues pending the multiplication of the two MSBs to give the MSB of the product. For example, if in some intermediate step, we get 110, then 0 will act as result bit (referred as $r_n$) and 11 as the carry (referred as $c_n$). It should be clearly noted that can may be a multi-bit number.

The hardware realization of a 4-bit multiplier is shown in figure 4. This hardware design is very similar to that of the famous array multiplier where an array of adders is required to appear at the final product. All the partial products are calculated in parallel and the delay associated is mainly the time taken by the carry to propagate through the adders which form the multiplication array. Clearly, this is not an efficient algorithm for the multiplication of large numbers as a lot of propagation delay is involved in such cases. To deal with this problem, we now discuss Nikhilam Sutra which presents an efficient method of multiplying two large numbers.

![Figure 4: Hardware architecture of the Urdhva Tiryakbhyam sutra multiplier](image)

C. Multiplication Using Nikhilam Sutra Algorithm

Nikhilam Sutra literally means “all from 9 and last from 10”. Although it is applicable to all cases of multiplication, it is more efficient when the numbers involved are large. Since it finds out the compliment of the large number from its nearest base to perform the multiplication operation on it, larger is the original number, lesser the complexity of the multiplication. We first illustrate this Sutra by considering the multiplication of two decimal numbers (96 * 93) where the chosen base is 100 which is nearest to and greater than both these two numbers.

![Figure 2: Multiplication of two decimal numbers by urdhvatiryakbhyam.](image)

The right hand side (RHS) of the product can be obtained by simply multiplying the numbers of the Column 2 ($7*4 = 28$). The left hand side (LHS) of the product can be found by cross subtracting the second number of Column 2 from the first number of Column 1 or vice versa, i.e., $96 - 7 = 89$ or $93 - 4 = 89$. The final result is obtained by concatenating RHS and LHS (Answer = 8928) [3].
The 8x8 bit multiplier is structured using 4X4 bit blocks as shown in figure 6. In this figure the 8 bit multiplicand A can be decomposed into pair of 4 bits AH-AL. Similarly multiplicand B can be decomposed into BH-BL. The 16 bit product can be written as:

\[ P = A \times B = (AH - AL) \times (BH - BL) = AH \times BH + AH \times BL + AL \times BH + AL \times BL \]

The outputs of 4X4 bit multipliers are added accordingly to obtain the final product. Thus, in the final stage two adders are also required. Now the basic building block of 8x8 bits Vedic multiplier is 4x4 bits multiplier which implemented in its structural model.

**IV SIMULATION RESULTS**

**V CONCLUSION**

The multiplier less and the DSP Slice-based correlators significantly exceed the timing requirements for OFDM synchronization. Urdhva Tiryakbhyam Sutra is most efficient Sutra (Algorithm), giving minimum delay for multiplication of all types of numbers, either small or large. Further, the Verilog HDL coding of Urdhva Tiryakbhyam Sutra for 8x8 bit multiplication and its FPGA implementation by Xilinx Synthesis Tool on Spartan 3E kit have been done.

**ACKNOWLEDGEMENT**

First and foremost I thank GOD, the almighty who stands behind and strengthens me to complete the project successfully. I would like to express my sincere respect and gratitude towards my HOD Mrs.M.Geetha, M.Tech.,(Ph.D). Her wide knowledge, serious research attitude and enthusiasm in work deeply impressed me and taught what a true scientific research should be. I am very thankful for the support she extended to me and the freedom to express my views. Words are inadequate to express the gratitude to my beloved husband, children, Parents, and friends for their excellent and never ending co-operation.

**REFERENCES**

V.Deepika, completed her B.E in Pavender Bharadhu dasan college of Engg & Tech, she is doing her M.E Applied Electronics in Jayaram college of Engg & Tech and her research area of interest is VLSI Design.

Mrs.S.Deva Priya, completed her B.E in Arulmigu Meenakshiamman college of Engg. & tech. She completed her M.E in Embedded systems in SASTRA University and she is doing her Ph.D in Mobile Communication in Anna University Trichy. She is working in Jayaram college of Engg & Tech for about 14 years. She had attended 12 international conferences, and 10 national conferences. She is a Life member in ISTE and member in IEEE, and she published 5 books. Her research area of interest is Embedded Systems, SOC, Mobile Communication, DSP, wireless communication.