



Low power binomial coefficient architecture for unused spectrum detector

O. Vignesh¹ · H. Mangalam²

Received: 23 February 2017 / Revised: 17 September 2018 / Accepted: 20 September 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

Recently, low power architecture is the major requirement in wireless devices. Cognitive radio is the most significant technology to realize the problem of choosing the temporarily unused channels (Sarijari et al. in *EURASIP J Wireless Commun Netw* 2015:221, 2015). In this paper, the binomial coefficient (nC_k) architecture is proposed for finding the possibilities of detecting the unused spectrum in the cognitive radio network. A novel factorial algorithm is designed using a modified parallel multiplier with Vedic multiplication method (Maharaja in *Vedic mathematics*, Motilal Banarsidass Publishers Pvt Ltd, Delhi, 2001). Compared to existing factorial algorithm (Saha et al. in *Microelectron J* 42:1343–1352, 2011), the proposed novel factorial algorithm can perform low power, high speed and occupy less number of logic elements because of the simple data-path reduction method using decoder circuit. A new binomial coefficient architecture is designed using the proposed factorial algorithm. The proposed modules are designed and implemented in Altera Cyclone II FPGA family.

Keywords Novel factorial algorithm · Modified parallel multiplication · Binomial coefficient architecture · Vedic multiplication · Altera FPGA

1 Introduction

The hardware implementation of cognitive radio network is a very challenging task in recent technologies. Cognitive Radio (CR) is the smart radio network technology that can automatically detect unused channels in the wireless spectrum and avoid the interference to other used channels [4]. The primary incumbent users get affected due to the interference of the spectrum if high signal to noise ratio (SNR) signals are allocated. So, the CR is required for detecting the low SNR signals [5]. The iterated discrete wavelet transform (DWT) approach and Eigen value-based approach are used to estimate the primary user's SNR for spectrum sensing in CR [6, 7]. The cooperative spectrum sensing scheme used the Binomial Theorem to sense the

different types of spectrum. In Binomial Theorem, the binomial coefficient (nC_k) is the important block in cognitive radio network. The binomial coefficient is used to determine the possibility of a chance to detect the unused spectrum from wireless network [8]. The VLSI implementation of factorial and binomial coefficient is used for the low power high-speed architecture of CR networks.

In contemporary manufacturing technology, VLSI architecture with low power, less number of data path elements and area are needed for computer arithmetic processing [9]. The computation of factorial is essential in calculating the combination and permutation of natural number. The combination algorithm is called as binomial coefficient. For example, the combination of n and k is $nc_k = n! / [(n - k)! * k!]$ and the value of factorial is $n! = n * (n - 1) * (n - 2) * \dots * 1$, that is, $n!$ is obtained by multiplying the series of numbers descending from n to 1 or ascending from '1 to n '; note that $0!$ is equal to 1. There are plenty of multipliers available in digital system design but an optimized multiplier is needed to obtain high performance with a minimum switching activity [10]. To attain a high speed factorial circuit, the parallel multiplier

✉ O. Vignesh
vicky6058@gmail.com

¹ Department of Electronics Engineering, MIT Campus, Anna University, Chennai, India

² Department of ECE, Sri Ramakrishna Institute of Technology, Coimbatore, India

is used. The dynamic power consumption is a major problem in VLSI design. So, the reduction of dynamic power is an important issue to be addressed. Also, reduction of the data-path element in the architecture helps to reduce the power consumption. The Vedic multiplier is one of the earliest multipliers that provide high performance with less number of LUTs.

2 Factorial using increment method

The block diagram of the existing factorial algorithm is illustrated in Fig. 1. The factorial calculation is computed by using five major blocks like zero detector, incrementer, comparator, counter and parallel multiplier.

2.1 Zero detector

The zero detector is used to find out whether the given input number is equal to zero or not. If the input is zero, output directly displayed as one and factorial calculation is skipped, since the value of zero factorial ($0!$) is 1. On the other hand, if the input is non zero, then the zero detector produces output that is the same as the input. The output of the zero detector is given by $Y = Y_n, Y_{n-1}, Y_{n-2} \dots Y_0$ where $Y_n = X_n, Y_{n-1} = X_{n-1}, Y_{n-2} = X_{n-2}$ and so on and $Y_0 = X_0 + \text{not}(Y_n, Y_{n-1}, Y_{n-2} \dots)$ where $X_n, X_{n-1}, X_{n-2} \dots X_0 = X$ is the input.

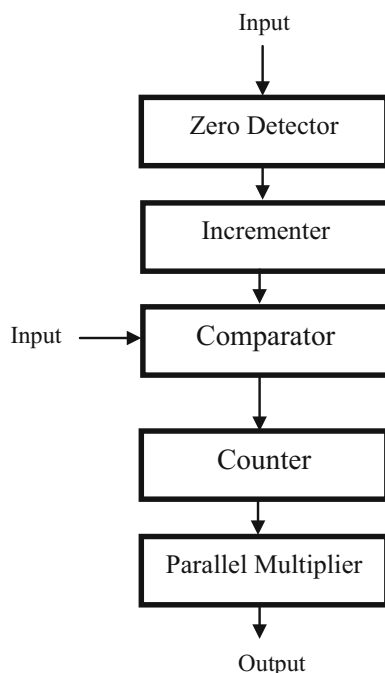


Fig. 1 Factorial using incrementer method

2.2 Incrementer

The output from the zero detector is fed to the incrementer. The initial value of the incrementer is set as zero. It increments by X times where X is the decimal equivalent of the input [9, 11].

2.3 Comparator and counter

The comparator compares the given input (X) and output of incrementer block (A). If the given input number is greater than the output of the incrementer block ($X > A$), then incrementer continues to increment until the incremented value is equal to the given number. When both numbers are equal ($X = A$), then the incrementer block stops the increment process [12]. During each comparison, the comparator provides the clock signal to the counter and hence the counter output is equal to the incremented value. The counter circuit is used to set the value in the register, which is the input of the parallel multiplier.

2.4 Parallel multiplication

The output of the counter is given as input to the parallel multiplier in each step of the increment process. In this work, a 4-bit parallel multiplier is considered, which consists of four stages and computes up to 15 factorial. The four stages consist of eight 4-bit multipliers, four 8-bit multipliers, two 16-bit multipliers and one 32-bit multiplier. There are 16 numbers of 4-bit binary numbers that can be multiplied in this parallel method as illustrated in Fig. 2. The first input is always “0001” to avoid the multiplication by zero. If the input is less than 15, then “0001” is set in the remaining registers of the parallel multiplier. For example, if $4!$ is computed, the first four inputs are set in four registers as 0100, 0011, 0010 and 0001 ($4 \times 3 \times 2 \times 1$), and the remaining 12 register values are set as “0001”. The Vedic multiplier is used in parallel multiplication block. Urdhva Triyagbhyam (UT) Sutra is a general multiplication method from Vedic mathematics. The meaning of Urdhva Triyagbhyam is Vertically Crosswise. For example, let us consider two inputs 21 and 32. The result of 21×32 is 672, which is calculated by UT Sutra method as shown in Fig. 3. All these multipliers consume more power during the running time of the circuit.

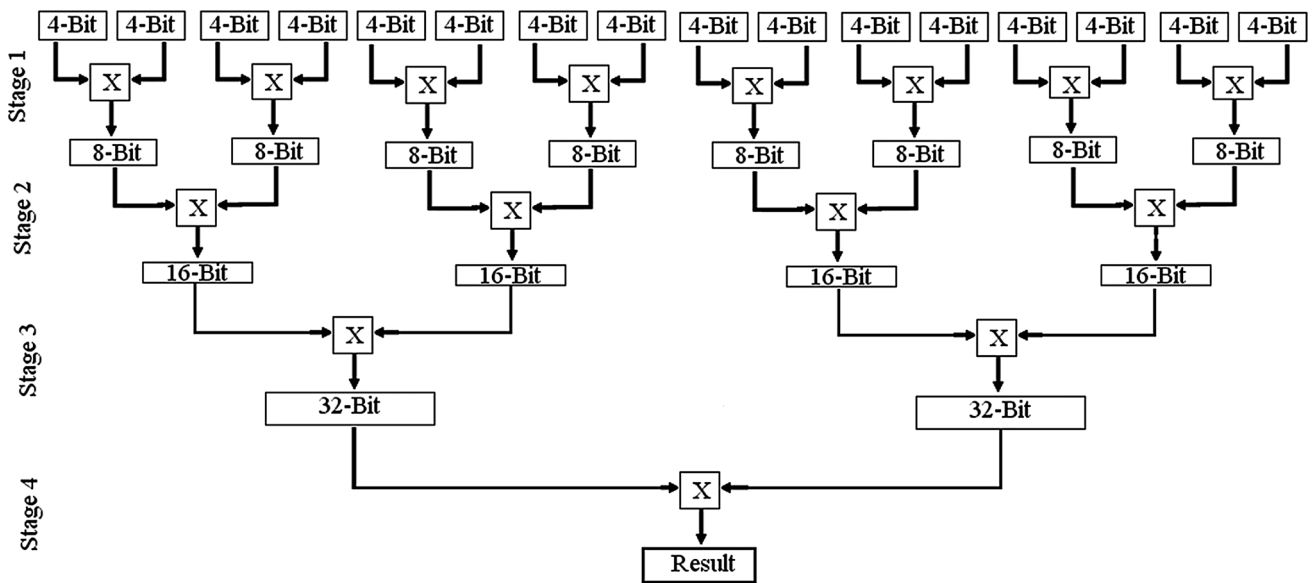


Fig. 2 Parallel multiplication

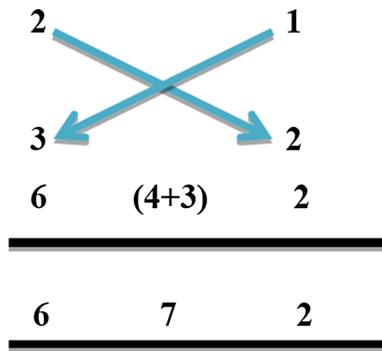


Fig. 3 Vedic multiplication

3 Proposed factorial using decoder method

The proposed factorial algorithm is performed by using a decoder block. The incrementer, comparator and counter blocks are replaced by the decoder. So, the number of LUT's is reduced in the new proposed factorial algorithm. The architecture of parallel multiplication is reconstructed by using 2:1 multiplexer and a tri-state buffer. The 4-bit architecture of factorial using decoder method is shown in Fig. 4.

3.1 Decoders

Two types of the decoders are used for factorial and binomial coefficient architecture. For 4-bit binomial coefficient architecture, both decoders are 4 to 16-bit circuits. The decoder1 generates the number of '1's from Least Significant Bit (LSB) according to the given input of factorial algorithm and also it generates '0's for remaining

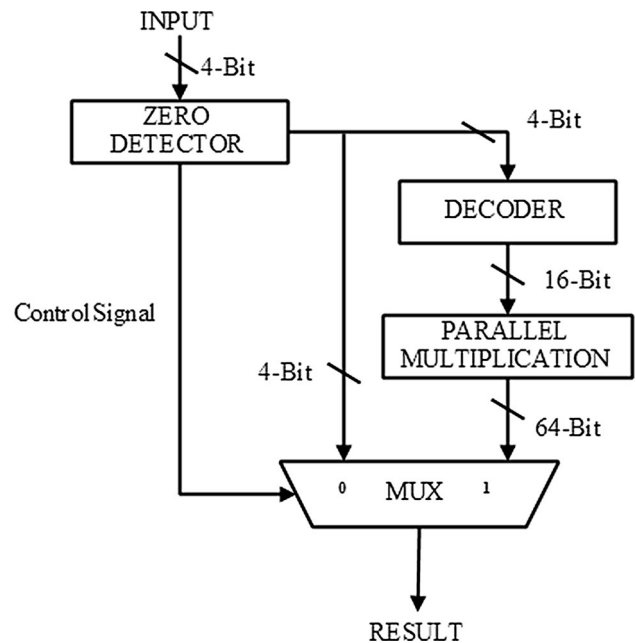


Fig. 4 Factorial using decoder method

digits. For example, if the factorial input is 5, then the decoder circuit generates five "1"s from LSB and remaining all "0"s, like '0000000000011111'. Similarly, the decoder2 circuit generates the number of '0's from LSB according to the subtracted $(n - k)$ value of input n and k and also it generates '1's for remaining digits, which is used for the binomial coefficient. For example, if the input is $n = 12$, $k = 4$, $(n - k) = 8$ then the decoder2 generates eight '0's from LSB and remaining all '1's, like '11111100000000'. The truth table of both decoders is shown in Table 1.

Table 1 Truth table of decoders

Input	Decoder1 output (s15–s0)	Decoder2 output (s15–s0)
0000	0000000000000000	1111111111111111
0001	0000000000000001	1111111111111110
0010	0000000000000011	1111111111111100
0011	0000000000000111	1111111111111000
0100	0000000000001111	1111111111110000
0101	0000000000111111	1111111111100000
0110	0000000001111111	1111111111000000
0111	0000000011111111	1111111110000000
1000	0000000111111111	1111111100000000
1001	0000001111111111	1111111000000000
1010	0000011111111111	1111110000000000
1011	0000111111111111	1111100000000000
1100	0001111111111111	1111000000000000
1101	0011111111111111	1110000000000000
1110	0011111111111111	1100000000000000
1111	0111111111111111	1000000000000000

3.2 Tri-state buffer

The normal buffer has only two output modes high or low, but the tri-state buffer has three different output modes like low, high and high impedance [13]. The three ports of Tri-state buffer are namely input, output and enable ports. The Tri-state output is high, when the enable and input are high otherwise, it remains in idle condition. In this algorithm, the tri-state buffer allows the input to modified parallel multiplication.

3.3 Modified parallel multiplication

The computational complexity is in general higher for multipliers. Designing optimal multipliers is a challenging task. This section presents a low power modified parallel multiplier module that is designed by using a tri-state buffer and multiplexer.

Figure 5 shows the multiplier module with Tri-state buffer for the modified parallel multiplier. The inputs S_1 and S_2 are the outputs from the decoder1 circuit. They are fed to the enable input of the tri-state buffers and the select line of 2:1 multiplexer. S_1S_2 can have only two possible combinations namely 10 and 11. If S_1S_2 is 10, then multiplication operation is skipped and first register values are stored in 8-bit register LSB position via a multiplexer. If S_1S_2 is 11, then the multiplied value is stored in the 8-bit register.

The conventional 4-bit parallel multiplier multiplies the values of all the 16 registers, which causes high dynamic power consumption. But, in the proposed modified parallel

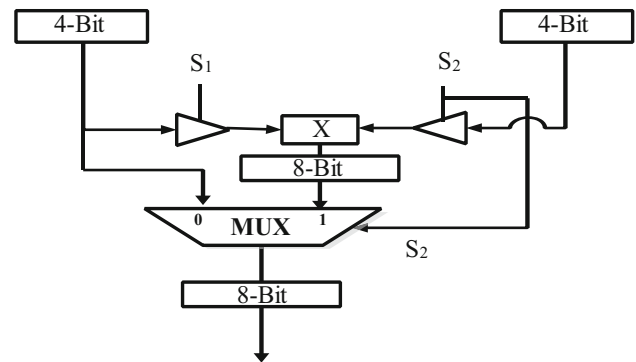


Fig. 5 Multiplier module with tri-state buffer

multiplier, the contents of required registers alone are multiplied based on the factorial inputs. This approach greatly reduces the power consumption. For example, factorial input 12 in conventional parallel multiplication requires the multiplication of $1 \times 1 \times 1 \times 1 \times 12 \times 11 \times \dots \times 2 \times 1$, but in modified parallel multiplication, only $12 \times 11 \times \dots \times 2 \times 1$ multiplications are required. The tri-state buffer makes their Most Significant Bit (MSB) places of registers in idle condition according to the input. Suppose, the input is less than 8 factorial in 4-bit modified parallel multiplier, then 50% of the hardware data-paths in the architecture is in idle condition that saves most of the power when compared to the conventional parallel multiplier.

The architecture of modified parallel multiplication is shown in Fig. 6. It comprises of 15 registers that store 0001 to 1111 values and these 4-bit register values are fed as inputs to modified parallel multiplication through the tri-state buffer. There are four stages involved in multiplication operation which is similar to the conventional parallel multiplier. The first stage consists of fifteen 4-bit registers and seven multiplication modules. The decoder1 outputs S_1 to S_{13} allow the stored 4-bit binary values to be multiplied and again stored in the 8-bit register. In the second stage, the decoder1 outputs S_2, S_6, S_{10} and S_{14} are allowed to multiply the 8-bit binary values which are the outputs of the first stage and second stage that are stored in the 16-bit register. The third stage S_4 and S_{12} and the fourth stage S_8 are used as control signals to multiply the pervious stage outputs and store respective outputs in registers.

4 Proposed binomial coefficient for cognitive radio

In this paper, a new factorial algorithm for cognitive radio networks application is proposed and it is shown Fig. 7. Cognitive radio searches the free space channel for effective transmission. In wireless communication, different

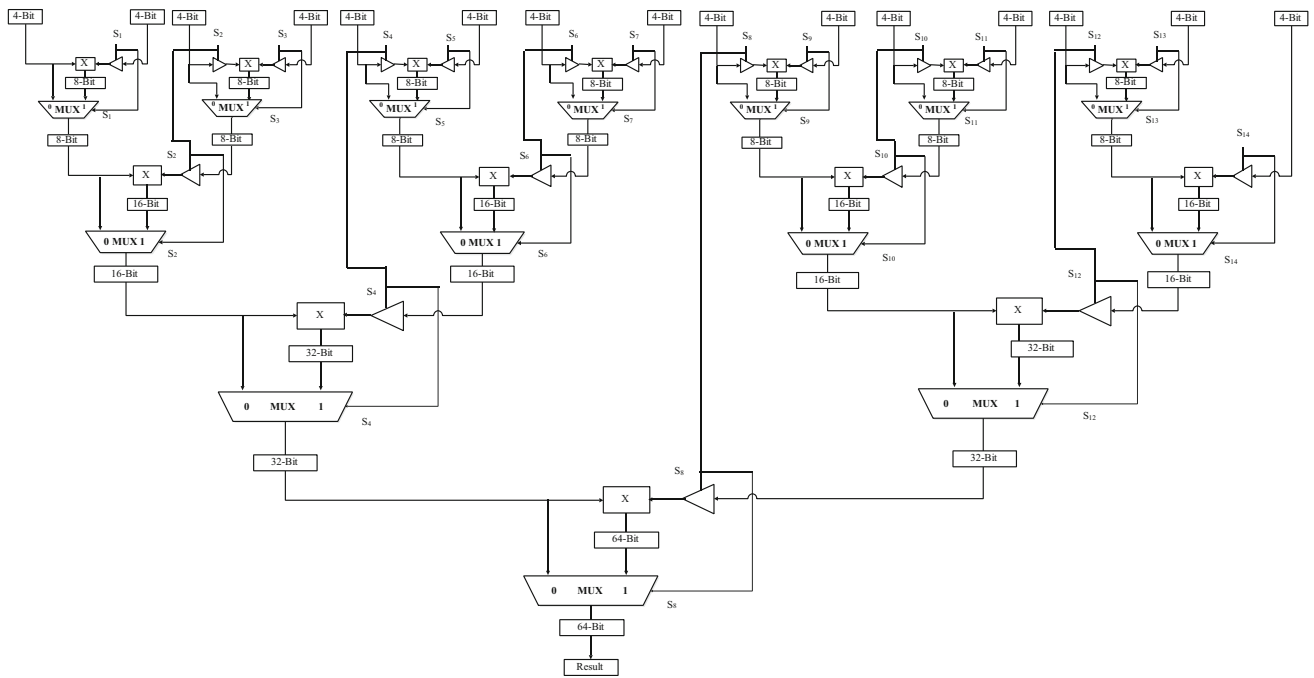


Fig. 6 Modified parallel multiplication

types of spectrum are involved like satellite signals, broadcast radio (AM and FM), infrared, Wi-Fi, mobile communication (2G, 3G, 4G) and TV signal [5]. To find the spectrum detection possibilities in wireless communication, the binomial coefficient is used. The binomial coefficient $\binom{n}{k}$ is calculated using the formula [8] given by

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{1}$$

where n is given input and k is factor interaction.

The effective process of choosing the free channels is to avoid repetition of the channels and order need not be considered. The full factorial (n!) gives all possible choices but the requirement is to find the fractional that is only non-repeated and unordered channels. The permutation (${}^n P_k$) is used to find the non-repeated channels which consider the order whereas the combination (${}^n C_k$) (also known as the binomial coefficient) is used for the non-repeated and unordered channels. For example, if ‘abcdef’ are the spectrum types, then out of this six, only three are required to be chosen. Then the combination $\{({}^6 C_3) = 20\}$ provides the non-repeated and unordered twenty choices. The calculation of permutation (${}^n P_k$) is given below

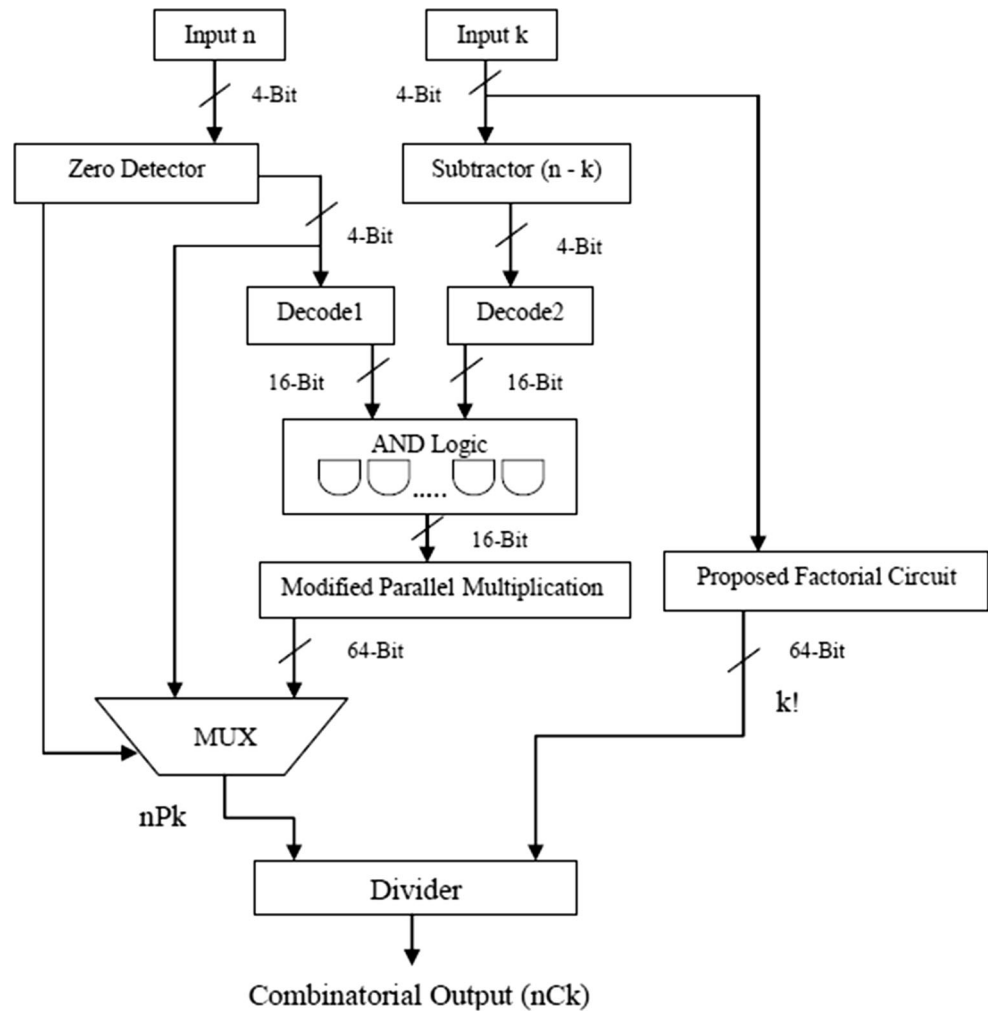
$$({}^n P_k) = \frac{n!}{(n-k)!} \tag{2}$$

The calculation of combination [14] is

$$({}^n C_k) = ({}^n P_k) \times \frac{1}{k!} \tag{3}$$

$$({}^n C_k) = \frac{n!}{(n-k)!} \times \frac{1}{k!} = \frac{n!}{k!(n-k)!} \tag{4}$$

Figure 7 illustrates the proposed binomial coefficient Architecture with proposed factorial. The permutation (${}^n P_k$) is computed without dividing the (n) factorial and (n - k) factorial using the decoder2 circuit. The decoder1 circuit generates ones from LSB according to ‘n’ value and the decoder2 circuit generates zeros from LSB according to n - k value. The outputs of the both decoder1 and decoder2 circuits are fed to the bitwise AND logic block, that produces a 16 bit control signal which is fed to the modified parallel multiplier. For an example, if the inputs are n = 13, k = 7 and (n - k) = 6, then the decoder1 output is “0001111111111111” and the decoder2 output is “1111111111000000”. The bitwise AND logic gives “0001111111000000”, then it allows multiplying 13 × 12 × 11 × 10 × 9 × 8 × 7 only. The control signals S₇–S₁₃ of tri-state buffer (S₁₅–S₀) are alone enabled and the permutation (${}^n P_k$) output is produced. Then, the binomial coefficient (${}^n C_k$) is computed by dividing the permutation (${}^n P_k$) by k!.

Fig. 7 Proposed binomial coefficient architecture

5 Results and discussion

The existing and proposed architectures are implemented using Altera Cyclone II FPGA (EP2C50F672C6) device. The power and delay performance of both architectures are simulated using Altera Quartus II—Power play power analyzer and Classic timing analyzer tools. Table 2 shows the comparative analysis of multipliers. It can be observed that Vedic multiplier [10] gives the best performance results and hence this multiplier is used in both architectures.

The performance analysis of both existing and proposed circuit modules is shown in Table 3. The existing circuit modules are implemented using parallel multiplication, factorial using increment method and binomial coefficient using existing factorial algorithm. The proposed circuit modules incorporate modified parallel multiplication, factorial using decoder method and binomial coefficient using new factorial algorithm. The results revealed that modified parallel multiplication method provide less power and delay, but slight increase in area. The factorial using

Table 2 Comparison of multiplier circuits

Multipliers	No. of bits	LEs	Delay (ns)	Power (mW)
Array multiplier	4×4	31	18.216	111.67
	8×8	169	29.359	113.52
	16×16	758	44.707	117.45
	32×32	3194	78.701	124.92
Wallace multiplier	4×4	33	14.525	111.77
	8×8	173	28.507	113.54
	16×16	766	39.653	117.87
	32×32	3232	76.863	124.48
Vedic multiplier	4×4	31	15.524	111.17
	8×8	163	28.442	113.23
	16×16	710	41.193	117.15
	32×32	2941	80.438	123.46

decoder method provides reduction in power, delay and area compared to incrementer method. The binomial coefficient architecture uses three factorial modules to

Table 3 Comparison of power, area and delay

Circuit modules		Power (mW)	Frequency (MHz)	Period (ns)	Area in terms of no. of LUTs
Existing circuit modules [3]	Parallel multiplication	123.18	61.2895	16.316	465
	Factorial using increment method	133.13	14.4793	69.064	2292
	Binomial coefficient using existing factorial algorithm	163.35	1.08172	924.457	7142
Proposed circuit modules	Modified parallel multiplication	118.95	65.4151	15.287	546
	Factorial using decoder method	121.25	18.5512	53.905	1391
	Binomial coefficient using new factorial algorithm	136.64	1.11966	893.13	5029

compute n factorial, $(n - k)$ factorial and k factorial. But, the proposed binomial coefficient computes only two factorial since ' n ' and ' $n - k$ ' factorial are combined using decoder2 and bitwise AND logic blocks. This reduction of factorial modules results in reduction in power, delay and area due to less number of logic elements.

6 Conclusion

A novel hardware efficient algorithm for the 4-bit binomial coefficient architecture is proposed in this paper. This architecture can be used as a key part for measuring the possibilities of detecting free channels in wireless networks. This is achieved through the use of a modified parallel multiplier and novel factorial algorithm. The proposed modified parallel multiplier has resulted in 3.4% reduction in power and 6% improved in speed when compared with the conventional parallel multiplier. Similarly, the proposed factorial computation using decoder method has resulted in 8.9% reduction in power, 21.9% speed enhancement and 39.3% reduction in the area compared to the conventional factorial computation using the increment approach. Likewise, the proposed binomial coefficient architecture saves 16% of power and 29.5% of the area with a speed improvement of about 3.4% compared to the conventional architecture. The proposed binomial coefficient architecture can be used in signal processing applications like binomial distribution systems.

References

- Sarijari, M. A., Abdullah, M. S., Janssen, G. J. M., & Veen, A. J. V. D. (2015). On achieving network throughput demand in cognitive radio-based home area networks. *EURASIP Journal on Wireless Communications and Networking*, 2015, 221.
- Maharaja, J. S. S. B. K. T. (2001). *Vedic mathematics*. Delhi: Motilal Banarsidass Publishers Pvt Ltd.
- Saha, P., Banerjee, A., Dandapat, A., & Bhattacharyya, P. (2011). ASIC design of a high-speed low power circuit for factorial calculation using ancient Vedic mathematics. *Microelectronics Journal*, 42, 1343–1352.
- Yin, W., Ren, P., Cai, J., & Su, Z. (2013). Performance of energy detector in the presence of noise uncertainty in cognitive radio networks. *Wireless Networks*, 19, 629–638.
- Quan, Z., Shellhammer, S. J., Zhang, W., & Sayed, A. H. (2009). Spectrum sensing by cognitive radios at very low SNR. In *Proceeding of the GLOBECOM 2009—2009 IEEE global telecommunications conference*, Honolulu, HI (pp. 1–6).
- Al-Hmood, H., Abbas, R. S., Masrub, A., & Al-Raweshidy, H. S. (2013). An estimation of primary user's SNR for spectrum sensing in cognitive radios. In *Third international conference on innovative computing technology (INTECH 2013)*, London (pp. 479–484).
- Sharma, S. K., Chatzinotas, S., & Ottersten, B. (2013). Eigenvalue based SNR estimation for cognitive radio in presence of channel correlation. In *2013 IEEE global communications conference (GLOBECOM)*, Atlanta, GA (pp. 1107–1112).
- Maleki, S., Chepuri, S. P., & Leus, G. (2013). Optimization of hard fusion based spectrum sensing for energy-constrained cognitive radio networks. *Physical Communication*, 9, 193–198.
- Bi, S., Gross, W. J., Wang, W., Al-Khalili, A., & Swamy M. N. S. (2005). An area-reduced scheme for Modulo $2n-1$ addition/subtraction. In *Proceeding of the IEEE ninth international database engineering and application symposium* (pp. 396–399).
- Mehta, P., & Gawali, D. (2009) Conventional versus Vedic mathematical method for hardware implementation of a multiplier. In *Proceedings of the IEEE international conference on advances in computing, control, and telecommunication*, Kerala (pp. 640–642).
- Huang, C. H., Wang, J. S., & Huang, C. Y. (2001). A high-speed CMOS incrementer/decrementer. In *Proceeding of the IEEE international symposium on circuits and systems*, Sydney (Vol. 4, pp. 88–91).
- Veeramachaneni, S., Krishna, M. K., Avinash, L., Reddy, P. S., & Srinivas, M. B. (2007). Efficient design of 32-bit comparator using carry look-ahead logic. In *Proceedings of the IEEE northeast workshop on circuits and systems*, Montreal (pp. 867–870).
- Badel, S. & Leblebici, Y. (2007). Tri-state buffer/bus driver circuits in MOS current-mode logic. In *2007 Ph.D. research in microelectronics and electronics conference*, Bordeaux (pp. 237–240).
- Carlitz, L. (1967). Some functional equations related to binomial coefficient summations. *Journal of Combinatorial Theory*, 3, 93–97.



O. Vignesh received M.E. degree in VLSI Design from Anna University Regional Centre Coimbatore in 2011 and B.E. Electronics and Communication Engineering degree from Bharath Niketan Engineering College in 2009 and he is pursuing Ph.D. degree in Anna University Chennai. His research areas include computer arithmetic VLSI circuits, VLSI implementation of image and signal processing and VLSI-based crypto processing.



H. Mangalam has obtained B.E. Degree in Electronics and Communication Engineering, M.E. Degree in Applied Electronics and Ph.D. in the faculty of Information and Communication Engineering with specialization in VLSI Design and Testing. She has 28 years of experience in Academia serving in varied positions in reputed engineering colleges in Coimbatore, Tamilnadu. Currently, she is working as a Professor in ECE, Sri Ramakrishna Institute of Technology, Coimbatore, Tamilnadu. She has to her credit a lot of publications in reputed journals. She is a member in Professional bodies like IEEE, IE (India), ISTE and SSI.