

Efficient Multiplication Carry-Save on-The Fly Correction with Advanced Vedic and Baugh-Wooley Methodologies

A. Nithya¹, M. Palaniappan²

M.E. VLSI Design, Shanmuganathan Engineering College, Arasampatti, Pudukkottai, Tamil Nadu 622507, India¹

Assistant Professor, Department of ECE, Shanmuganathan Engineering College, Arasampatti,

Tamil Nadu 622507, India²

ABSTRACT: The main objective of the proposed work is implementing the 32 bit Vedic and Baugh-Wooley multiplication operation with Carry-Save methodology to reduce the average time and delay of operations. This system contain a high speed low power digital multiplier by using advantage of Vedic multiplication algorithms with a very efficient leakage control technique called multiple channel conventional multiplier technology. We have designed 16 bit Vedic multiplier using Carry-Save technology and used comparative simulation results that indicate the performance of the circuit. Vedic mathematics is a ancient Indian mathematics is very useful for doing tedious and cumbersome mathematical calculation at a very fast rate. The Vedic multiplier is approximately 10 times faster performance than the conventional multiplier architecture. Experiments show that our proposed multiplier schemes have only shorter delay with respect to a conventional Booth radix-4 multiplier schemes.

KEYWORDS: Multiplier, Booth Encoding, Partial Carry-Save, Data-path Synthesis

I. INTRODUCTION

Addition and multiplication are the key arithmetic operations in the vast majority of digital circuits and processors. On the one hand, more complex operations such as division, square root, exponential, and logarithmic computations are based on them. Signal processing, multimedia applications and even fixed point scientific calculations are also often dominated by integer addition and multiplication. Therefore, it is crucial to improve the performance of adders, multipliers and those structures that utilize them as basic blocks without incurring significant area or power overhead.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 2, February 2017

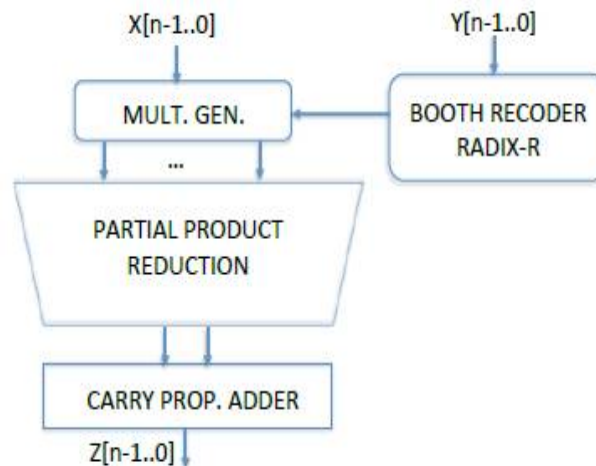


Fig. 1: Radix R Booth Multiplier

Depending on what pairs of generation and propagation signals are interconnected, several kinds of prefix adders can be implemented. All of them possess a delay in the order of $O(\log(n))$. Among these designs, the Kogge-Stone (KS) adders are the fastest prefix adders, but also the most complex ones, as their area grows with $O(n \cdot \log(n))$. On the other hand, multipliers are usually composed of a Partial Product Matrix (PPM), which accumulates the partial products and reduces them to just two operands, and a last Carry Propagate Adder (CPA) stage, which adds these two operands and produces the final result.

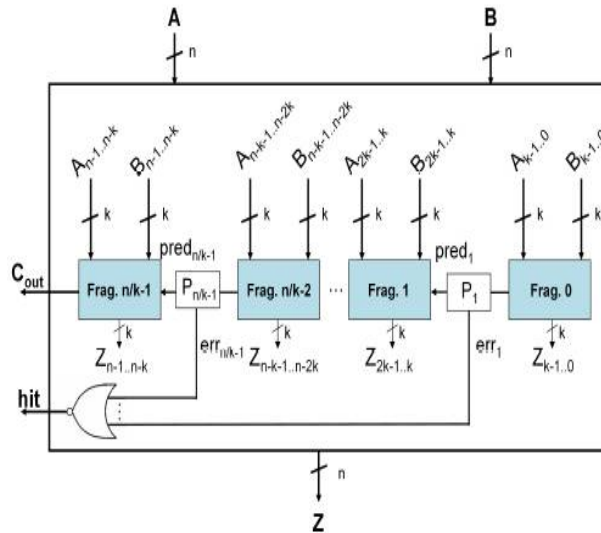
The height of the partial product matrix is usually reduced by applying a Booth recoding in radix $R = 2; > 0$. Thus the height of an $n \times n$ multiplier is reduced from n to $d(n + 1) = e$. Applying this recoding technique, some logic will be necessary to calculate the Booth multiples or subproducts. Typically $R = 2$, because given a product $X \cdot Y$, only $0X$, $1X$ and $2X$ multiples need to be generated. As can be observed, all of them can be easily calculated via shift and negation operations.

For $R > 2$, there appear hard multiples, i.e., those that are not composable with only shifts and negations, and the penalty due to their calculation exceeds the gains from reduction of the partial product matrix height. The generic structure of a Radix R Booth Multiplier is depicted in Figure 1.

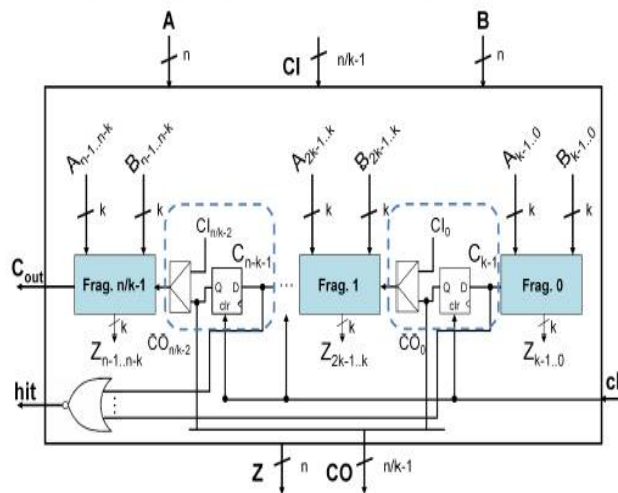
All the aforementioned designs operate with fixed latency. As an alternative, several Variable Latency Functional Units (VLFU) have appeared in the last two decades. The main idea behind these designs is to shorten the carry propagation path by predicting one or several of the intermediate carries. There is an increasing interest in the design of VLFUs. In this system we will focus on the Multispeculative Functional Units (MSFU). These units divide the n -bit adders into several k -bit fragments, predicting the value of the fragment input carries, so the results produced by them are in partial carry-save form.

Thereby, when one or two of these results need to be multiplied, they must be first reduced to a non-redundant form to utilize a conventional two input multiplier. This fact can produce several penalty cycles, stopping the data path locally or globally. In order to overcome this limitation, in this system a partial carry-save multiplier is proposed. It receives two inputs in partial carry-save form, and produces the result in partial carry-save form as well.

In this way, all the mis-predictions are corrected on-the-fly and there will be no need to stop the data path until the correct final output is calculated. We refer to this new multiplier as the On-the-fly Multispeculative Multiplier (OMSM).



(a) Generic Multispeculative Adder



(b) Multispeculative Adder in multispeculative datapaths

Fig. 2: Multispeculative Adder (MSADD) designs

II. SYSTEM ANALYSIS

Some works have recently studied the impact of dividing a FU into several sub-functional units that work in parallel in order to produce ultra fast datapaths. Authors present the partial redundant format to implement several complex operations. Authors divide the n-bit adders into k-bit fragments. These k-bit fragments are overlapped, and replicate several computations with a '0' carry-in, relying on the low probability of a (k + 1)-bit propagates signal to happen.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 2, February 2017

However, if this happens the result is incorrect. A different approach is presented, where the adders are divided into two or more independent fragments interconnected with a carry predictor, based on data correlation. The major drawback of these approaches is that every time there is a $(k + 1)$ -bit propagate signals chain or a failure in the prediction, a stall must be included in the datapath. If the controller is centralized, the whole datapath will be stopped. On the other hand, if the controller is distributed, the stall can be just partial and might be even hidden. However, even with these distributed controllers, if the failure happens in the critical path or the stalled operations cause a hazard to an operation in the critical path, the stall will be translated into extra cycles in the latency of the circuit.

In past system authors present a multi-speculative adder and a mechanism to propagate the carry predictions through addition chains, instead of stopping and correcting. Initially, the prediction is '0' for every carry-in. Afterwards, the intermediate carries are pipelined to the following addition, and correction only happens in the last addition of the chain.

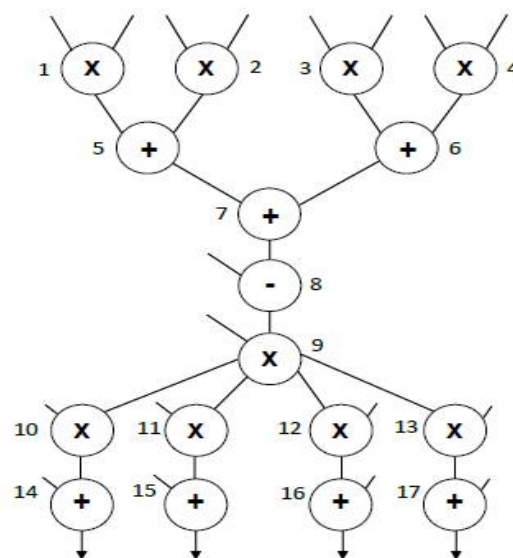


Fig. 3: LMS dataflow graph

In this addition, a misprediction then happens whenever a k -bit fragment produces a carry-out equal to '1'. This idea is extended to structures composed of additions and products, restricting the multiplications to be only found in the leaf nodes of the structure. In this way, the penalty due to possible stalls can only happen when the root operation of the structure is being executed. Nevertheless, this constraint on the location of multipliers imposes a limitation to this technique.

The reason is that in multispeculative datapaths, the results generated by an adder or multiplier (multipliers contain a multispeculative adder to implement the last stage addition) are produced in partial redundant form. So if a multiplier receives two operands in partial redundant form $X = U + A$; $Y = V + B$, the distributive property must be applied. Hence, a correct or non-redundant result must be produced before driving it to a multiplier. Our approach will tackle this problem to avoid the penalty cycles. Some previous works have dealt with total or partial redundant multipliers. The work presents a multiplier able to work with partial carry-save multiplicands and multipliers. Given n and k word and fragment bitwidths, respectively, every generated multiple after the Booth encoding requires $n + n=k$ bits, plus an extra '1' for negative multiples, which is translated into a larger 4:2 compression tree.

The floating-point unit described earlier performs the product in a Signed-Digit (SD) format, but requiring two separate 4:2 compression trees for the positive and negative vectors. Authors in past, propose a precoder stage able to deal with both redundant and non redundant multipliers, but not multiplicands. The work describes a multiplier able to receive both inputs in carry-save representation, with a complex multiples generation stage, which produces a noticeable area

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 2, February 2017

increase. Finally, the work in presents a design flow utilizing the previous multiplier. In this system we propose a multiplier able to accept both inputs in partial carry-save, without requiring any intermediate conversion, and integrated with the adders described earlier in an efficient way, which allows generating partial carry-save intermediate results as well as non-redundant circuit outputs.

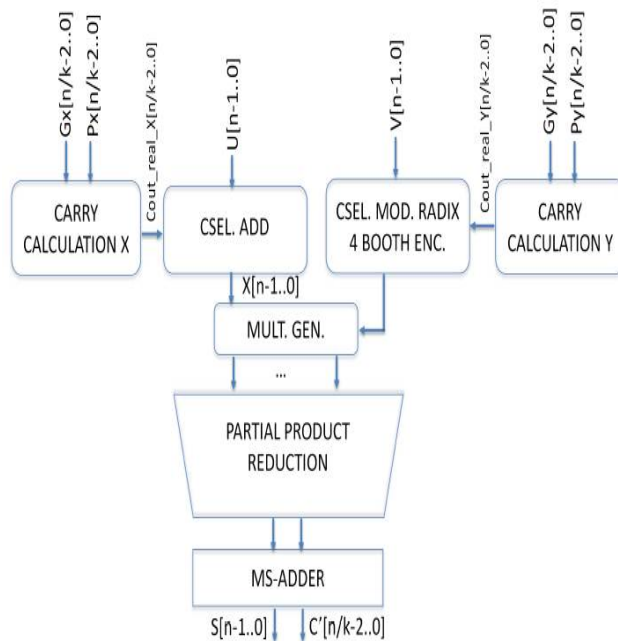


Fig. 4: On-the-fly Correcting Multispeculative Multiplier (OMSM)

II. EXISTING SYSTEM

- ✚ Existing Multiplication techniques process the input data slowly like a bit basis.
- ✚ So that the past system is suitable for only limited set of data bits to process at a time.
- ✚ We need a special mechanism to improve the efficiency of multiplication process.
- ✚ Time Consuming and practically hard to implement for large size data bits.
- ✚ Cost Expensive Process.

Disadvantages of Existing System

- ✓ More Power Consumption
- ✓ Delay rate is too high
- ✓ Memory Requirement is higher for processing
- ✓ Cost Expensive

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 2, February 2017

III. PROPOSED METHODOLOGY

- ✚ The Proposed system presents an efficient technique is used for multiplication of two binary numbers which reduce the average power, leakage power and delay.
- ✚ The proposed architecture is based on Vedic mathematics to increase the speed performance of binary multiplication which is based on 16 sutra(formula).
- ✚ This idea of designing the multiplier from Vedic mathematics is come because partial product and sum are generated in single step.
- ✚ An efficient Baugh-Wooley Design technology with carry-save method has been used for power and performance optimization of the Multiplier Operation.
- ✚ On the other hand, multipliers are usually composed of a Partial Product Matrix (PPM), which accumulates the partial products and reduces them to just two operands, and a last Carry Propagate Adder (CPA) stage, which adds these two operands and produces the final result.

Advantages of Proposed System

- ✓ Memory consumption is low in processing time.
- ✓ Delay is comparatively reduced
- ✓ Energy Consumption is reduces and low cost expensive in work.

IV. LITERATURE SURVEY

Partial redundant formats have appeared in the last decades as a way of mitigating the area overhead of total redundant formats, such as the Carry-Save (CS) or the Redundant Binary Representation (RBR), while achieving higher performance than non-redundant formats. As in any redundant system, when two operands X; Y are added or multiplied the output is in the form of $Z = S + C$, where S and C are the sum and the carry components, respectively. Some works have recently studied the impact of dividing a FU into several sub-functional units that work in parallel in order to produce ultra fast datapaths.

Authors in past present the partial redundant format to implement several complex operations. Authors in earlier stages divide the n-bit adders into k-bit fragments. These k-bit fragments are overlapped, and replicate several computations with a '0' carry-in, relying on the low probability of a (k + 1)-bit propagates signal to happen. However, if this happens the result is incorrect. A different approach is presented earlier, where the adders are divided into two or more independent fragments interconnected with a carry predictor, based on data correlation.

The major drawback of these approaches is that every time there is a (k + 1)-bit propagate signals chain or a failure in the prediction, a stall must be included in the data path. If the controller is centralized, the whole datapath will be stopped. On the other hand, if the controller is distributed, the stall can be just partial and might be even hidden. However, even with these distributed controllers, if the failure happens in the critical path or the stalled operations cause a hazard to an operation in the critical path, the stall will be translated into extra cycles in the latency of the circuit.

In past the authors present a multispeculative adder and a mechanism to propagate the carry predictions through addition chains, instead of stopping and correcting. Initially, the prediction is '0' for every carry-in. Afterwards, the intermediate carries are pipelined to the following addition, and correction only happens in the last addition of the chain. In this addition, a misprediction then happens whenever a k-bit fragment produces a carry-out equal to '1'. This idea is extended to structures composed of additions and products, restricting the multiplications to be only found in the

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 2, February 2017

leaf nodes of the structure. In this way, the penalty due to possible stalls can only happen when the root operation of the structure is being executed.

Nevertheless, this constraint on the location of multipliers imposes a limitation to this technique. The reason is that in multispeculative datapaths, the results generated by an adder or multiplier (multipliers contain a multispeculative adder to implement the last stage addition) are produced in partial redundant form. So if a multiplier receives two operands in partial redundant form $X = U + A$; $Y = V + B$, the distributive property must be applied. Hence, a correct or non-redundant result must be produced before driving it to a multiplier. Our approach will tackle this problem to avoid the penalty cycles.

Some previous works have dealt with total or partial redundant multipliers. The work presents a multiplier able to work with partial carry-save multiplicands and multipliers. Given n and k word and fragment bitwidths, respectively, every generated multiple after the Booth encoding requires $n + n-k$ bits, plus an extra '1' for negative multiples, which is translated into a larger 4:2 compression tree. The floating-point unit described in earlier stages, which performs the product in a Signed-Digit (SD) format, but requiring two separate 4:2 compression trees for the positive and negative vectors. Authors in earlier stages propose a precoder stage able to deal with both redundant and non redundant multipliers, but not multiplicands. The work describes a multiplier able to receive both inputs in carry-save representation, with a complex multiples generation stage, which produces a noticeable area increase. Finally, the work presents a design flow utilizing the previous multiplier.

V. EXPERIMENTAL RESULT

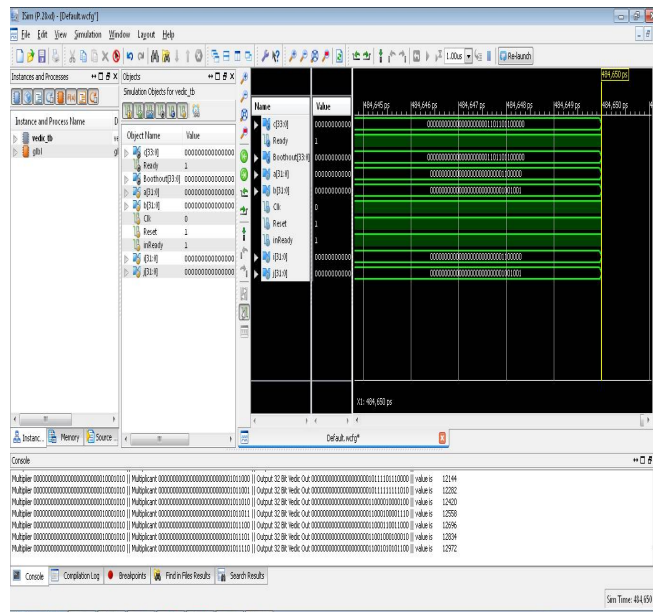


Fig.5. Multiplier Design

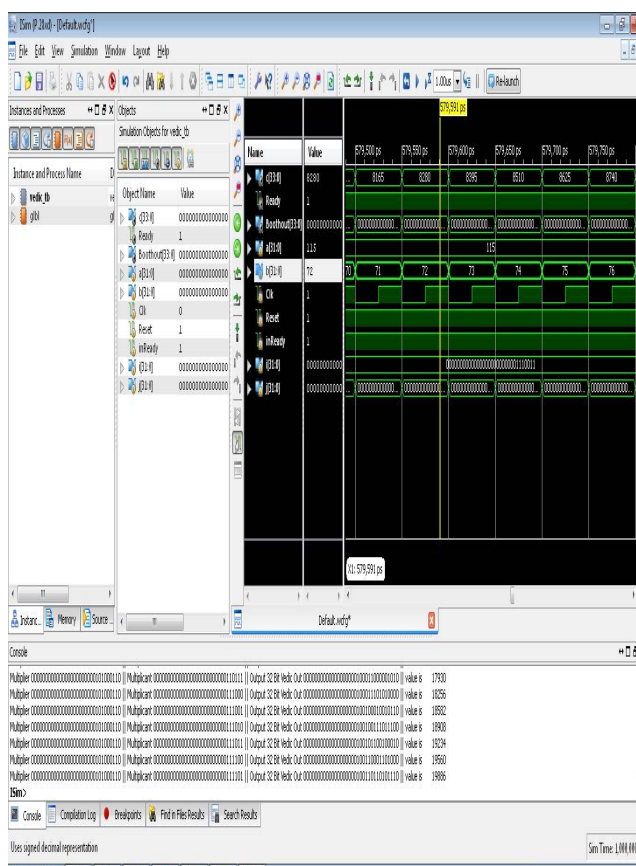


Fig.6. Vedic Output

VI. CONCLUSION AND FUTURE SCOPE

In this system we have designed and simulated a 32 bit Vedic and Baugh-Wooley multiplier with Carry-Save technique. Though using the Vedic technique the average power increases a bit but the propagation delay of the circuit reduces in a large proportion to cause a considerable amount of reduction in performance. Moreover, the use of these Vedic and Baugh-Wooley structures helps the circuit more power efficient and provides high performance in ultra low power applications. The Vedic multiplier gives about 73 %–90 % less amount of delay compared to the conventional multiplier schemes and the overall delay is reduced considerably. We have designed the circuit keeping our concentration primarily on reduction of propagation delay, dynamic average and leakage power consumption and it has been successfully shown that by using our proposed algorithm.

REFERENCES

- [1] A. A. D. Barrio, R. Hermida, and S. O. Memik, “Exploring the energy efficiency of multispeculative adders” in ICCD, 2013, pp. 309–315.
- [2] A. A. D. Barrio, R. Hermida, S. O. Memik, J. M. Mendias, and M. C. Molina, “Improving circuit performance with multispeculative additive trees in high-level synthesis” *Microelectronics Journal*, vol. 45, no. 11, pp. 1470–1479, Nov. 2014.
- [3] V. Castellana and F. Ferrandi, “Scheduling independent liveness analysis for register binding in high level synthesis” in DATE, 2013, pp. 1571–1574.
- [4] J. Hormigo, J. Villalba, and E. Zapata, “Multioperand redundant adders on fpgas” *IEEE Trans. Comput.*, vol. 62, pp. 2013–2025, Oct. 2013.
- [5] M. Erle, B. Hickmann, and M. Schulte, “Decimal floating-point multiplication,” *IEEE Trans. Comput.*, vol. 58, pp. 902–916, Jul. 2009.
- [6] A. Booth, “A signed binary multiplication technique,” *Quarterly Journal of Mechanical Applied Mathematics*, vol. 4, pp. 236–240, mar 1951.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 2, February 2017

- [7] P.-M. Seidel, L. McFearn, and D. Matula, "Binary multiplication radix-32 and radix-256," in ARITH, 2001, pp. 23–32.
- [8] A. D. Gloria and M. Olivieri, "Statistical carry lookahead adders," IEEE Trans. Comput., vol. 45, pp. 340–347, Mar. 1996.
- [9] S. M. Nowick, K. Y. Yun, A. E. Dooply, and P. A. Beerel, "Speculative completion for the design of high-performance asynchronous dynamic adders," in ASYNC, 1997, pp. 210–223.
- [10] L. Benimi, E. Macii, M. Poncino, and G. D. Micheli, "Telescopic units: A new paradigm for performance optimization of vlsi design," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 17, pp. 220–232, Mar. 1998.
- [11] J. L. et al., "An algorithmic approach for generic parallel adders," in ICCAD, 2003, pp. 734–740.
- [12] E. Ashmila, S. Dlay, and O. Hinton, "Adder methodology and design using probabilistic multiple carry estimates," in Computers and Digital Techniques, 2005, pp. 697–703.
- [13] D. Kelly and B. Phillips, "Arithmetic data value speculation," in Asia-Pacific Conference, 2005, pp. 353–366.
- [14] A. K. Verma, P. Brisk, and P. Jenne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in DATE, 2008, pp. 1250–1255.
- [15] A. Cilaro, "A new speculative addition architecture suitable for two's complement operations," in DATE, 2009, pp. 664–669.

BIOGRAPHY



Ms. Nithya. A. Studying M.E. VLSI Design in Shanmuganathan Engineering College, which is located in Arasampatti, Pudukkottai, Tamil Nadu, India.