2021 IEEE International Symposium on Smart Electronic Systems (iSES) | 978-1-7281-8753-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/iSES52644.2021.00044

Efficient Design of Artificial Neural Networks using Approximate Compressors and Multipliers

KATTEKOLA NARESH
ECE Department
NIT Meghalaya and
VNRVJIET, Hyderabad
nareshkattekola@gmail.com

Dr. SHUBHANKAR MAJUMDAR
ECE Department
NIT Meghalaya
shub@nitm.ac.in

Dr. Y. PADMA SAI
ECE Department
VNR VJIET
Hyderabad, Telangana
padmasai_y@vnrvjiet.in

P. ROHITH SAI
PG Scholar
VNR VJIET
Hyderabad, Telangana
rohithsai27@gmail.com

***Abstract*—Nowadays, Artificial Neural Networks (ANNs) secured impressive results with multiple applications and approaches in various research fields, as well as image processing, face recognition and semantic segmentation. Here, the focus is to minimize the complexity of ANN hardware in keeping accuracy as a major concern. ANN is a subsystem that is approximate, in machine learning where it trains the neurons to get the relevant output according to the target value. By using this ANN, interfacing can be possible between approximate arithmetic circuits. 3:2, 4:2 compressors are designed with unique error positions, usually gives better power area and delay constraints in between 5 to 25%. The designed approximate ANN gains the design constraints in the range of 3 to 30%. The simulation results were done by using synopsys design compiler at 90nm Technology.**

Keywords— ***Approximate Computing, Artificial Neural Networks, Multipliers, Neurons, Compressors***

I. INTRODUCTION

Approximate computing (AC) is one of the techniques that increase the efficiency of the system which leverages the error resilience characteristics of the applications. Low power techniques like dynamic voltage, power gating, and frequency scaling, and power-energy aware application mapping, which are popular from older days and frequently used to gain the energy and area efficiency of the system. But these may not be sufficient to meet the requirement of the present growing computer technologies and demands. With these, the requirement in reduction of design constraints is limited and bounded to some restrictions. Precise computing doesn't offer many trade-off inefficient systems where the design size depends on the number of applications or required instructions of the design. Hence imprecise design has some of the applications which can have more scope in reducing the design constraint parameters to any extent.

The approximate Computing concept has some specific error bound, which is acceptable and always considerable for some imprecise applications. Arithmetic circuits are the major concept in any digital design which has more scope in designing and reconfiguring the design. Approximating the arithmetic circuits receiving significant attention from the IC design community. Approximate computing is a calculation technique which gives an inexact result rather than the guaranteed exact result.

Every application doesn't need an accurate computation unless it is very sensitive to errors. Even with the precedence, as the complexity increases, an optimized design with error-tolerant circuits doesn't affect the functionality in some applications. An ASIC implementation is proposed here for a specific application that doesn't depend on accurate results.

Approximate computing is a major advantage for image processing applications where there is no scope for accuracy. for image compression and image format conversion applications needed combinational blocks like adders[1], compressors[2], multipliers [2],[3] and dividers [4],[5],[6],[7],[8].

Compressors are basic circuits that are made of full adder and half adders to count the number of "ones" in the input. Several compressors are required in the partial product reduction stage which then affects multipliers. Various compressors with different approaches and low power techniques at gate level and transistor-level such as 3-2, 4-2, 5-2, and 5-3 were proposed by researchers in the last 20 years. These are useful only when the preferred multiplier is small in size. 16 & 32-bit multipliers [9] require a large number and size of compressors which is a major score in controlling the design constraints.

II. PROPOSED MODULES

A. 3:2 Approximate Compressor

The 3:2 approximate compressor is an approximate full adder where the outputs are the sum and carry. It is designed with new positioned error models named Proposed Approximate Compressor Model (PACM)-1 and 2. PACM-1 and 2 are proposed with 1 and 2 errors respectively for better efficiency which are shown in Fig.1. with error positions.

The expressions of PACM-1are

$$S=AB'+B'C+AC+A'BC' \quad (1)$$

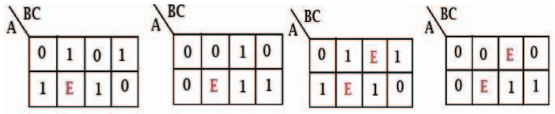
$$Cout=BC+AB \quad (2)$$

The expressions of PACM-2 are

$$S=AB'+C+A'B \quad (3)$$

$$C=AB \quad (4)$$

978-1-7281-8753-2/21/$31.00 ©2021 IEEE
DOI 10.1109/iSES52644.2021.00044



(a) (b)

Fig. 1. K-Map for a) PACM-1 & b) PACM- 2 Design of S, Cout

B. 4:2 Approximate Compressor

The 4:2 approximate compressor is introduced as PACM-3 and -4 with 3 and 4 errors respectively in the design of approximation. PACM-3 is proposed with 3 errors introduced at the 5th, 9th,15th position in the provided K-Map by keeping 0's and 1's in combinations for errors in similar places of sum and carry. Hoping to have reduced carry chains and in the PACM-4 of the proposed approximate compressor design is introduced with 4 errors at the position of 5th, 9th, 12th, and 15th in the K-Map at similar places of sum and carry. The PACM-4 occupies more carry chains when compared to PACM-3 even with more error positioning places. Fig. 2. shows PACM-3 and 4 respectively.

The expressions of PACM-3 are

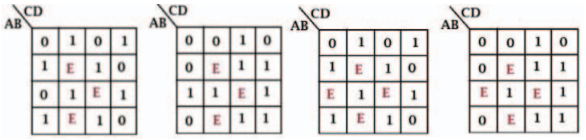
Sum= C'D+BD+AD+A'BC'+AB'C'+ABC+A'B'CD (5)

Carry=CD+BC+AC+AB (6)

The expressions of PACM-4 are

Sum= C'D+BC'+BD+AB+AC'+AD+A'B'CD' (7)

Carry= CD+BC+AC+ABD (8)



(a) (b)

Fig. 2. K-Map for a) PACM-3 & b) PACM-4 Design of S, Cout

C. Approximate Multiplier

The array multiplier is the conventional design in multiplication. Here the error positions for the compressors are introduced in the array multiplier. The design occupies two 4:2 compressors and one 3:2 compressor generally. But the approximate multiplier design is developed with all the combinations of Proposed Approximate Compressor Models (PACM)1, 2 3 and 4, and hence the Approximate Multiplier models expected are Proposed Approximate Multiplier Models (PAMM) 1, 2, 3 and 4. The Fig. 3 shows the architecture of Array Multiplier using 3:2, 4:2 compressors.

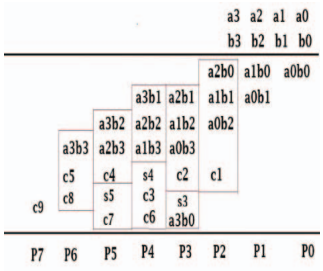


Fig. 3. 4-bit Multiplier using 4:2 and 3:2 Approximate Compressors

D. Design of ANN

Fig. 4 presents the neuron architecture using an approximate multiplier block. Neuron and ANN are implemented using proposed approximate multipliers and compressor design blocks to explore the area and latency tradeoff. Approximate Compressor based Neuron Model (ACNM) is designed using approximate compressors. The proposed multiplier design is used to realize each neuron computation in each layer, and in the second one, called Approximate Compressor Multiplier based neuron (ACMN), a single proposed approximate compressor and multiplier design is used to implement the whole ANN. Moreover, we present efficient hardware implementation of ANN architecture using approximate compressors and multipliers that the ANN hardware accuracy by reducing its complexity.

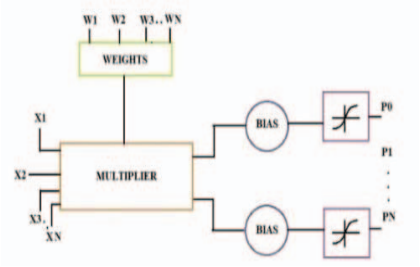


Fig. 4. Neuron using Approximate Multiplier design

The primary inputs of ANN are $X_1, X_2 \ldots X_n$. These all variables are multiplied by the relative weights at the first hidden layer during the computations.the size of multiplexers are determined for the input variables by the maximum number of inputs at all layers, and the sixe of these multiplexers aredefined by the total number of weights and bias values .

The control block, which actually represents a counter functioning block to which accumulated value is added and stored in the register R. In this architecture, the neuron calculation is obtained after n + 1 clock cycles and a common control block which denotes m and n are occupy a number of inputs and outputs of the layer, respectively. The control block synchronization input variable is multiplied by the relevant weight. The Fig. 5 shows the ANN architetcure using multiplier design.

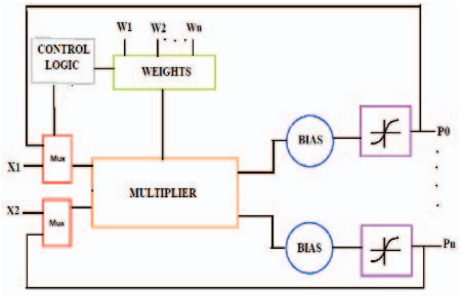


Fig. 5. ANN using Approximate Multiplier design

III. Results & Discussions

A. Design Constraints Comparison for Compressors and Multipliers.

The simulation results of the compressors and multipliers of accurate and approximate design models were done by using Synopsys design compiler at 90nm Technology.

TABLE I 3:2 Compressors

Compressor Design	Power (uW)	Area (um^2)	Delay (nS)	Power Gain	Area Gain	Delay Gain
Accurate3:2	10.09	39.62	0.24	-	-	-
PACM-1	9.18	34.01	0.19	9.0	14.1	20.8
PACM-2	7.60	28.64	0.18	24.6	27.7	25.0
SVCM [3]	7.71	29.01	0.18	23.58	26.7	25.0
Accurate4:2	24.72	95.76	0.43	-	-	-
PACM-3	23.06	83.78	0.40	6.7	12.5	7.0
PACM-4	15.20	71.71	0.38	38.5	25.1	11.6
MHCM [2]	23.32	86.63	0.40	5.66	9.53	7.0

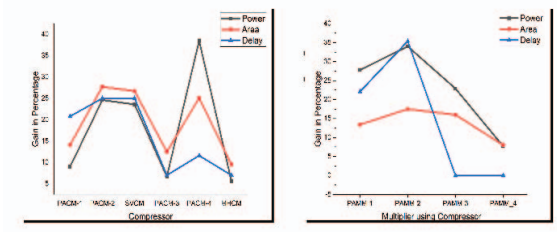
TABLE II Multiplier using Compressor

Multiplier using Compressor	Power (uW)	Area (um^2)	Delay (nS)	Power Gain	Area Gain	Delay Gain
Accurate3:2	121.05	525.29	1.13	-	-	-
PAMM-1	87.34	454.73	0.88	27.8	13.4	22.1
PAMM-2	79.90	433.25	0.73	34.0	17.5	35.4
Accurate4:2	136.45	600.84	1.01	-	-	-
PAMM-3	105.16	504.64	1.02	22.9	16.0	-
PAMM-4	125.93	552.91	1.02	7.7	8.0	-

The Table-I, II describes the design of different accurate and approximate compressors and accurate and approximate 4-bit multipliers respectively.

Here PAMM-1 and -2 are designed using PACM-1, -2 with PACM-3 respectively, and PAMM-3 and -4 designed using PACM-3 and -4 respectively. PAMM-5 and -6 designed using PACM-1, -2 with PACM-4 respectively.

The below Fig.6 shows the gains of the design constraints for Compressors and Compressor based Multipliers PACM, PAMM respectively



(a) (b)

Fig. 6. The Graph to compare gains of design constraints for a) compressors & b) Multiplier using Compressor

From the tabular results and comparisons of I, Approximate 3:2 and 4:2 compressors have gained in all design constraints. The advantage of 4:2 compressors is the power gain which is in the range of 25 to 40%. For 5:2 compressor, which is mainly based on 3:2 is also have gain in all the design constraints. Table II describes about approximate multiplier using PACM-1,2,3,4 models, which has power gain of 38% for model 4 with area gain of 25%.

B. Design Constraints Comparison for Neuron and ANN

The Table-III, IV describes the design of, ACNM, ACMN using 3:2,4:2 compressors and table-V, VI describes the design constraints of ACANN, ACMANN using compressors and multipliers. Simulation is performed by using Synopsys Design compiler under 90nm technology.

Here ACMN-1 and -2 are designed using ACNM-1, -2 with ACNM-3 respectively, and ACMN-3 and -4 designed using ACNM-3 and -4 respectively.

TABLE III Neuron using Compressor

Compressor	Power (uW)	Area (um^2)	Delay (nS)	Power Gain	Area Gain	Delay Gain
Accurate3:2	8.19	68.94	0.45	-	-	-
ACNM-1	8.08	63.33	0.39	1.3	8.1	13.3
ACNM-2	7.22	57.96	0.39	11.8	16	13.3
Accurate4:2	21.82	128.10	0.79	-	-	-
ACNM-3	17.10	104.05	0.72	21.6	18.7	8.86
ACNM-4	18.83	116.12	0.75	13.7	9.35	5.06

TABLE IV Compressor based Multiplier Neuron

Multiplier using Compressor	Power (uW)	Area (um^2)	Delay (nS)	Power Gain	Area Gain	Delay Gain
Accurate	27.10	601.11	1.51	-	-	-
ACMN-1	26.78	537.66	1.44	1.2	10.5	4.63
ACMN-2	26.63	515.18	1.25	1.7	14.3	17.2
Accurate	28.40	662.50	1.57	-	-	-
ACMN-3	27.63	566.30	1.48	2.7	14.5	5.73
ACMN-4	24.98	592.45	1.48	12.04	10.5	5.73

TABLE V Compressor based ANN

Compressor	Power (uW)	Area (um^2)	Delay (nS)	Power Gain	Area Gain	Delay Gain
Accurate3:2	48.48	495.30	0.74	-	-	-
ACANN-1	44.90	439.14	0.68	7.38	11.33	8.1
ACANN-2	40.31	385.43	0.68	16.85	22.18	8.1
Accurate4:2	128.81	907.01	0.81	-	-	-
ACANN-3	99.08	690.57	0.76	23.08	23.86	6.17
ACANN-4	126.05	799.17	0.80	2.14	11.88	1.23

TABLE VI Compressor based Multiplier ANN

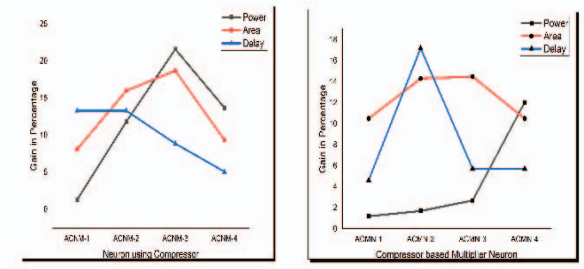
Multiplier using Compressor	Power (uW)	Area (um^2)	Delay (nS)	Power Gain	Area Gain	Delay Gain
Accurate3:2	284.0	7540.5	8.94	-	-	-
ACMANN-1	203.0	6291.4	7.31	28.5	16.5	18.2
ACMANN-2	201.9	5996.1	7.18	28.9	20.4	19.7
Accurate4:2	424.31	10583.2	13.6	-	-	-
ACMANN-3	360.11	9040.5	12.88	15.1	14.57	5.29
ACMANN-4	392.41	9336.9	13.12	7.51	11.77	3.52
ACMANN-5	344.99	8432.7	12.24	18.7	20.31	10

A feedforward ANN is implemented, which has a 3:2 compressor (with 8 inputs), a hidden layer with 10 neurons and an output layer (with 8 neurons), and a 4:2 compressor (with 16 inputs), a hidden layer with 10 neurons and an output layer with 16 neurons, and using these two compressors, a multiplier (with 16 inputs) and a hidden layer (with 16 Neurons) and an output layer (with 16 neurons). The activation functions in the hidden layer and the output layer are the symmetric saturated linear function and SoftMax, respectively.

ANN was trained using MATLAB's ANN toolbox. In the case where the training and test inputs are standardized to between -1 and 1, the weights are initialized randomly, and are adjusted using a learning method based on backpropagation to minimize the error between the actual response and the expected response. The ANN was trained with 1000 data and tested with 866 data. After training, the misclassification rate was 2.19%.

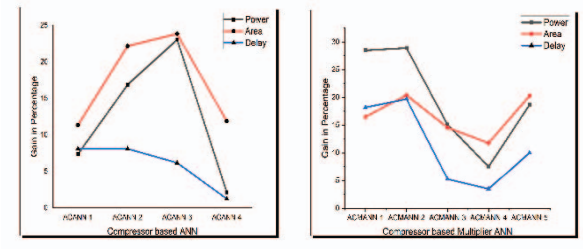
Approximate Compressor based ANN(ACANN) is implemented using models 1,2,3 and 4. Compressor based Multiplier ANN (ACMANN) designs of approximate Models are implemented using 1,2,3,4 based compressors. Model 5 is designed by using 5:2 compressor which is based on 3:2. Table III, IV, V and VI shows the design of approximate ANN at different architecture levels using 1,2,3,4 respectively.

The below Fig. 7. shows the gains of the design constraints for ACNM using 3:2 & 4:2 compressors, ACMN of compressor-based multipliers using 3:2 & 4:2 compressors. The Fig. 8 shows the gains of design constraints for ACANN using 3:2 & 4:2 compressors and Compressor based Multiplier ANN Model (ACMANN).



(a) (b)

Fig. 7. The Graph to compare gains of Neuron using a)compressor & b) Compressor based Multiplier Neuron



(a) (b)

Fig. 8. The Graph to compare gains of a) Compressor based ANN and b) Compressor based Multiplier ANN

From the tabular results and comparisons, approximate compressors-based Neuron have gained in all design constraints. The advantage of 3:2 compressors-based neuron has got the power gain which is in the range of 2 to 22%.

The design of ACMN, ACNM, ACANN and ACMANN with required possibilities are tabulated from III to VI. From the tabular column, there is no proper gain in ACANN-4 which is because of the carry chain. Due to accurate modules at the remaining stages, there is no sufficient gain in power and area. When comparing the ACANN-1, 2 designs, the submodules are efficient in all constraints, hence the gain is possible for all the constraints within the range of 5 to 25%.

IV. Conclusions

The 4-bit multipliers are designed with 3:2, 4:2 with all possible combinations of which have proposed models using K-Map. The results show that the 3:2 compressor has power, area, and delay gains of 24.6%, 27.7%, and 25% respectively. For the 4:2 compressors, the power, area, and delay gains of 38.5%, 25.1%, and 11% respectively. The approximate multiplier with the better gains from the power, area, and delay of 34%, 17.5%, and 35.4% respectively. Hence the overall gain change is between 5% to 35% from the error rate of 18.5% to 25% in all design constraints.

An ANN is designed using different multipliers, compressors of approximate logic. The design constraints are evaluated in gain format for power, area, and delay. From the possible comparisons, approximate ANN gains the power, area and delay in the maximum range of 18.7%, 20.31% and 10% respectively. Here the overall gain changes between 3 to 30% which is better in all the constraints.

References

[1] Manickam Ramasamy, G.Narmadha, S. Deivasigamani “Carry based approximate full adder for low power approximate computing,” in 2019 7th International Conference on Smart Computing & Communications (ICSCC), 2019.

[2] Minho Ha , Sunggu Lee., “Multipliers with Approximate 4–2 Compres- sors and Error Recovery Modules,” in IEEE Embedded Systems Letters, 2018.

[3] S. Venkatachalam and S. Ko, "Design of Power and Area Efficient Approximate Multipliers," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 5, pp. 1782-1786, May 2017.

[4] J. A. Arjun and S. Majumdar, "Development of Approximate Compressor Based Hybrid Dadda Multiplier for Image De-noising Applications," 2019 IEEE 16th India Council International Conference (INDICON), Rajkot, India, 2019.

[5] R. Marimuthu, Y. E. Rezinold and P. S. Mallick, "Design and Analysis of Multiplier Using Approximate 15-4 Compressor," in IEEE Access, vol. 5, pp. 1027-1036, 2017.

[6] S. Pabithra and S. Nageswari, "Analysis of Approximate Multiplier Using 15–4 Compressor for Error Tolerant Application," 2018 International Conference on Control, Power, Communication and Computing Technologies (ICCPCCT), Kannur, India, 2018.

[7] T. U. Swathi Krishna, K. S. Riyas, Y. Premson and R. Sakthivel, "15–4 Approximate Compressor Based Multiplier for Image Processing," 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2018.

[8] H. Baba, T. Yang, M. Inoue, K. Tajima, T. Ukezono and T. Sato, "A Low-Power and Small-Area Multiplier for Accuracy-Scalable Approximate Computing," 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Hong Kong, China, 2018.