International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Comparative Performance Evaluation of Different 8-Bit Approximate Multipliers for Computer Vision Applications

Kuldeep Singh[1], Ajay Kumar[1], Manisha Jangra[1] , Ritu Boora[1], Vinod Kumar[1]
[1]Department of Electrical and Electronics Engineering,
Guru Jambheshwar University of Science and Technology, Hisar, Haryana, India
kuldeepsingh@gjust.org; ajaypunia506@gmail.com; manisha0586@gmail.com; rituboora@gmail.com, Vinodspec@yahoo.co.in

Abstract— Approximate multiplication is attractive for error-resilient computer-vision datapaths because multiplication dominates many convolution, filtering, gradient, and feature-extraction kernels. This work experimentally compares four unsigned 8-bit multiplier architectures—an exact reference, an ordinary column-truncated multiplier, a fixed-bias truncated multiplier, and a three-boundary-column compensated (BWC-3) multiplier—under a common registered FPGA benchmark. All architectures are evaluated on a Xilinx Zynq-7000 XC7Z020-CLG484-1 device using a 14-point post-route timing sweep from 50 to 200 MHz. The reported implementation data show markedly different timing–area–power trade-offs. The exact design maintains non-negative slack through 180.995 MHz, whereas ordinary truncation and fixed bias close timing through 200 MHz; BWC-3 closes only through 133 MHz in the measured sweep because its adaptive correction logic lengthens the critical path. At 200 MHz, fixed bias requires 68 LUTs and 0.021 W dynamic power compared with 84 LUTs and 0.029 W for the exact reference. The results demonstrate that approximation quality and FPGA implementation efficiency must be considered jointly when selecting multiplier cores for vision-oriented hardware.

Index Terms—Approximate computing, truncated multiplier, fixed-bias compensation, BWC-3, FPGA, Zynq-7000, timing closure, computer vision, power-delay product.

I. INTRODUCTION

Today in the emerging intelligent autonomous systems, surveillance applications, medical imaging technology, human-machine interaction, edge computing and Internet of Things platforms computer vision has become the fundamental capability and requirement. A large volumes of pixel data is being processed in the repeated operations, including convolution, filtering, feature extraction, transform computation, correlation, and neural-network inference in the execution of advance vision algorithms. In the all the above operations multiplication is contributing a significant cost due to it's multiply-accumulate units. In the real time processing at the edge increases this multiplication core demand further largely for embedded cameras, autonomous vehicles, portable wearable devices and systems and there is a limit for energy need, silicon area acquired, and power dissipation. However, there are various techniques and architectural parallelism for scaling these requirements to improve the throughput, and optimize the energy and hardware overhead involved in exact arithmetic. Therefore, the approximate computing is emerged as an effective design model for making use of the inherent error ability to set back to its original state in many image and vision workloads [1]-[3]. The use of approximate computing in computer vision applications act in accordance with several attributes of visual data and algorithms. The approximate arithmetic operations must therefore be estimated using both circuit-level model efficiency and application-level architecture accuracy [4]-[6].

Multipliers block architecture are more engaging objects for approximation due to their partial-product creation and the product term reduction systems which contribute to large amount of logic depth and switching activity. The literature describe various techniques include partial-product truncation, approximate compressors, simplified reduction trees, logarithmic multiplication, and error-compensation circuits can be used for approximate computations. The truncated multipliers remove lower partial products and provide better hardware savings, secondly, the un-compensated truncation multiplier generates mainly the negative error as all the deleted unsigned partial products are nonnegative in values. The fixed-bias approach minimize these systematic error by inserting a constant correction, and a data-dependent compensation scheme to select operand or boundary-column information to evaluate the discarded contribution [7]-[9].

The 8-bit operand width is highly relevant to resource-constrained vision systems. The 8-bit arithmetic widely used in low-power inference accelerators where eight-bit pixels rest common in grayscale and color-image channels, and integer quantization. The verification and characterization of error behavior at this level become practical for precision, exhaustive evaluation of all 65,536 unsigned input pairs, rather depend on a limited random sample. Hence, a credible

performance analysis of the system is must to connect the exhaustive numerical analysis with representative computer-vision experiments and post-synthesis hardware measurements [10-11].

In the literature there is an absence of comparative assessment of the multiplier over a consistent control basis. Maximum of the studies report area, delay, and power under different platforms or frameworks, and other compare a DSP-mapped multiplier study against LUT-based customized approximate logic computation. Such incomprehensive studies don't clearly state whether an observed advantage comes from the approximation technique or from the experimental platform [12-14]. In this study we addresses that issue by enclosing every multiplier core within the identical registered boundaries conditions that is, input registers, a purely combinational multiplier core, and an output register. Each architecture is exercise and synthesized separately to minimize the chances of logic sharing and cross-hierarchy optimization during the processing. Thus, the article presents a comprehensive comparative performance evaluation of different 8-bit unsigned approximate multipliers that may be used for computer-vision datapaths. The experimental study exercised an exact partial-product reference, ordinary truncation, fixed-bias compensation, and a special category of boundary- weighted-compensation design model which progressively have deeper observation of omitted partial-product columns. The performance of the hardware resources utilized is estimated under the integrated implementation flow using logic utilization, register count, critical-path delay, maximum operating frequency, dynamic and total power, power-delay product, and area-delay product.

The main contribution of study is a reproducible framework for identifying when a design offers a meaningful accuracy-efficiency advantage. It establishes similar sequential and synthesis boundaries for all the multiplier architectures. The evaluation is deliberated to direct designers in selecting an 8-bit multiplier according to the error tolerance, throughput, energy budget, and implementation constraints of the target vision system.

II. MULTIPLIER ARCHITECTURES MODEL

Here, every multiplier core is kept placed between similar input and output registers as depicted in the functional model diagram in Fig. 1. The mathematical function for the computation of the exact 16-bit product is expressed as $P_EX = \Sigma_i \Sigma_j a_i b_j 2^{(i+j)}$, for i,j=0,…,7. Here, all 64 terms contribute to the result.

In case-II, the ordinary architecture with CUT = 6, suppresses the partial products satisfying condition $i+j<6$ while maintaining a 16-bit output follows $P_OT = \Sigma_(i+j\geq 6) a_i b_j 2^{(i+j)}$ arithmetic. The discarded contribution from the out is given by D_CUT and at the end provides the signed error $e_OT=P_OT-P_EX=-D_CUT$; this approximation process is basically one-sided.

The fixed-bias truncated design inserts one constant correction significant component at the truncation boundary in the process and the final result is expressed as $P_FB = P_OT + 2^{CUT} = P_OT + 64$. This economical design for hardware do a correction shifts in the ordinary truncation error toward zero without introducing operand-dependent compensation logic.

The BWC-3 compensated multiplier follows the three most-significant discarded columns for computations. If n1, n2, and n3 variables are the population counts of columns CUT−1, CUT−2, and CUT−3, respectively, then $q=(4n1+2n2+n3+4)>>3$ and $P_BWC3=P_OT+q\cdot 2^{CUT}$. This adaptive correction improves arithmetic accuracy but increases the logic depth and routing demand in hardware implementation. The experimental methodology for the unique set-up is expressed through the diagram shown in Fig. 1.

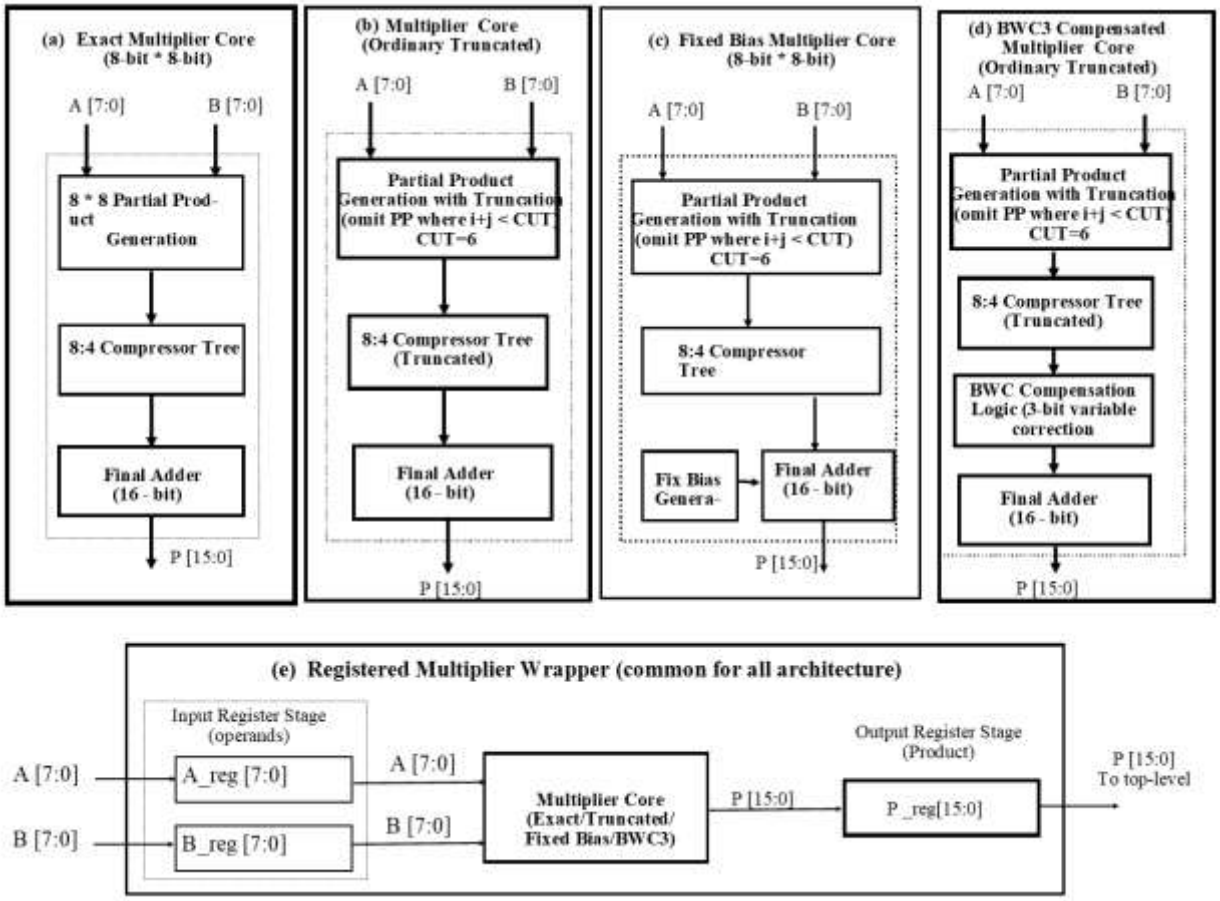


Fig. 1. Functional block diagrams of all 8-bit multipliers: a) Exact, b) Ordinary Truncated, c) Fixed-Bias, d) BWC3 Compensated Core, and Registered Multiplier wrapper.

III. RESULTS AND DISCUSSION

During the experimental process the reported results are presented in the table-I, II after post-route measurements for the XC7Z020-CLG484-1 target at 14 target frequencies: 50, 62.5, 75.09, 87.5, 100, 112.5, 125, 133, 150, 162.6, 175.131, 180.995, 190.5, and 200 MHz. For each run, the recorded quantities are worst negative slack (WNS), critical delay, slack-derived Fmax, total and dynamic power, LUTs, FFs, I/O, PDP, and ADP. Timing closure is identified conservatively by WNS ≥ 0. TABLE I shows the timing-closure and 200-MHz implementation summary for the designed models.

TABLE I TIMING-CONSTRAINTS AND 200-MHz IMPLEMENTATION SUMMARY

Architecture	Highest target with WNS≥ 0 (MHz)	WNS@200	Tcrit@200	Fmax est.@200	Pdyn@200	LUT@200	PDP@200	ADP@200
Exact	180.995	-0.132	5.132	194.86	0.029	84	0.1488	431.09
Ordinary	200.000	0.036	4.964	201.45	0.020	67	0.0993	332.59
Fixed Bias	200.000	0.012	4.988	200.48	0.021	68	0.1047	339.18
BWC-3	133.000	-2.438	7.438	134.44	0.022	111	0.1636	825.62

Table II expresses the central implementation trade-off for the parametric values of all four models. As the results show ordinary truncation and fixed bias retain positive slack values at 200 MHz, while the exact reference marks into negative slack value at 190.5 MHz. but BWC-3 measured negative WNS at 150 MHz and remains negative thereafter in the runs. Hence, the adjustable correction in values subject to conditions is not free.

TABLE II REPORTED POST-IMPLEMENTATION RESULTS

Arch.	MHz	WNS	Tcrit	Fmax est.	Pdyn	LUT	PDP	ADP
Exact	50.000	12.584	7.416	134.84	0.007	71	0.0519	526.54
Exact	62.500	8.725	7.275	137.46	0.009	71	0.0655	516.52
Exact	75.090	6.035	7.295	137.08	0.011	71	0.0802	517.95
Exact	87.500	4.407	7.023	142.39	0.012	71	0.0843	498.63
Exact	100.000	3.501	6.499	153.87	0.014	71	0.0910	461.43
Exact	112.500	2.030	6.860	145.77	0.016	71	0.1098	487.06
Exact	125.000	1.466	6.534	153.05	0.018	71	0.1176	463.91
Exact	133.000	0.800	6.719	148.83	0.019	71	0.1277	477.05
Exact	150.000	0.560	6.110	163.67	0.022	71	0.1344	433.81
Exact	162.600	0.192	5.958	167.84	0.023	80	0.1370	476.64
Exact	175.131	0.304	5.406	184.98	0.025	81	0.1351	437.89
Exact	180.995	0.233	5.292	188.96	0.026	75	0.1376	396.90
Exact	190.500	-0.094	5.357	186.67	0.027	82	0.1446	439.27
Exact	200.000	-0.132	5.132	194.86	0.029	84	0.1488	431.09
Ordinary	50.000	12.870	7.130	140.25	0.005	54	0.0357	385.02
Ordinary	62.500	9.093	6.907	144.78	0.006	54	0.0414	372.98
Ordinary	75.090	6.287	7.043	141.98	0.008	54	0.0563	380.32
Ordinary	87.500	4.541	6.889	145.16	0.009	54	0.0620	372.01
Ordinary	100.000	3.092	6.908	144.76	0.010	54	0.0691	373.03
Ordinary	112.500	2.492	6.398	156.30	0.011	54	0.0704	345.49
Ordinary	125.000	1.311	6.689	149.50	0.013	54	0.0870	361.21
Ordinary	133.000	0.923	6.596	151.61	0.014	54	0.0923	356.18
Ordinary	150.000	0.204	6.466	154.66	0.015	54	0.0970	349.16
Ordinary	162.600	0.104	6.046	165.40	0.017	55	0.1028	332.53
Ordinary	175.131	0.321	5.389	185.56	0.018	59	0.0970	317.95
Ordinary	180.995	0.323	5.202	192.23	0.019	63	0.0988	327.73
Ordinary	190.500	0.170	5.093	196.35	0.020	66	0.1019	336.14
Ordinary	200.000	0.036	4.964	201.45	0.020	67	0.0993	332.59
Fixed Bias	50.000	12.855	7.145	139.96	0.005	56	0.0357	400.12
Fixed Bias	62.500	8.640	7.360	135.87	0.006	56	0.0442	412.16
Fixed Bias	75.090	6.029	7.301	136.97	0.008	56	0.0584	408.86
Fixed Bias	87.500	4.543	6.887	145.20	0.009	56	0.0620	385.67
Fixed Bias	100.000	3.355	6.645	150.49	0.010	56	0.0664	372.12
Fixed Bias	112.500	2.613	6.277	159.31	0.012	56	0.0753	351.51
Fixed Bias	125.000	1.274	6.726	148.68	0.013	56	0.0874	376.66
Fixed Bias	133.000	0.910	6.609	151.31	0.014	56	0.0925	370.10
Fixed Bias	150.000	0.268	6.402	156.20	0.016	56	0.1024	358.51
Fixed Bias	162.600	0.304	5.846	171.06	0.017	60	0.0994	350.76
Fixed Bias	175.131	0.123	5.587	178.99	0.018	58	0.1006	324.05
Fixed Bias	180.995	0.194	5.331	187.58	0.019	64	0.1013	341.18

Arch.	MHz	WNS	Tcrit	Fmax est.	Pdyn	LUT	PDP	ADP
Fixed Bias	190.500	0.171	5.092	196.39	0.020	67	0.1018	341.16
Fixed Bias	200.000	0.012	4.988	200.48	0.021	68	0.1047	339.18
BWC-3	50.000	9.760	10.240	97.66	0.005	85	0.0512	870.40
BWC-3	62.500	5.907	10.093	99.08	0.007	85	0.0707	857.90
BWC-3	75.090	3.382	9.948	100.52	0.008	85	0.0796	845.58
BWC-3	87.500	1.871	9.559	104.61	0.010	85	0.0956	812.51
BWC-3	100.000	0.697	9.303	107.49	0.011	85	0.1023	790.76
BWC-3	112.500	0.205	8.685	115.14	0.012	86	0.1042	746.91
BWC-3	125.000	0.128	7.872	127.03	0.014	91	0.1102	716.35
BWC-3	133.000	0.047	7.472	133.83	0.015	97	0.1121	724.78
BWC-3	150.000	-0.605	7.275	137.46	0.017	113	0.1237	822.08
BWC-3	162.600	-1.036	7.186	139.16	0.018	109	0.1293	783.27
BWC-3	175.131	-1.665	7.375	135.59	0.020	118	0.1475	870.25
BWC-3	180.995	-1.959	7.484	133.62	0.020	103	0.1497	770.85
BWC-3	190.500	-2.031	7.294	137.10	0.022	112	0.1605	816.93
BWC-3	200.000	-2.438	7.438	134.44	0.022	111	0.1636	825.62

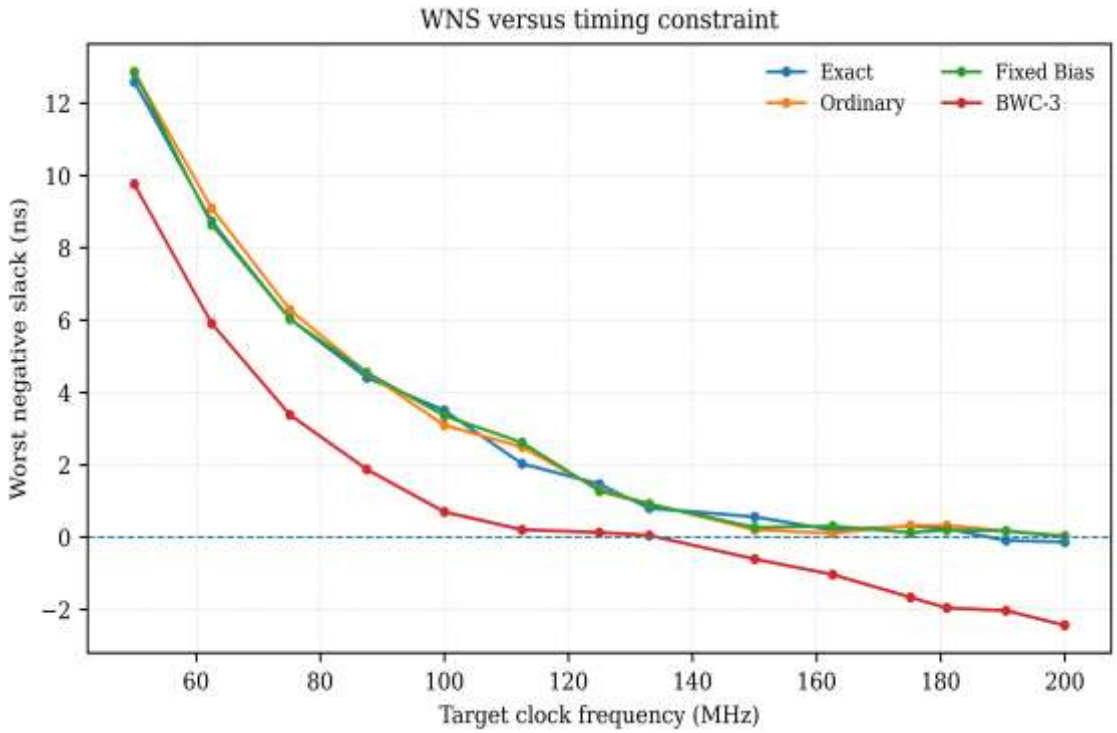


Fig. 2. WNS versus target frequency for the four multiplier architectures.

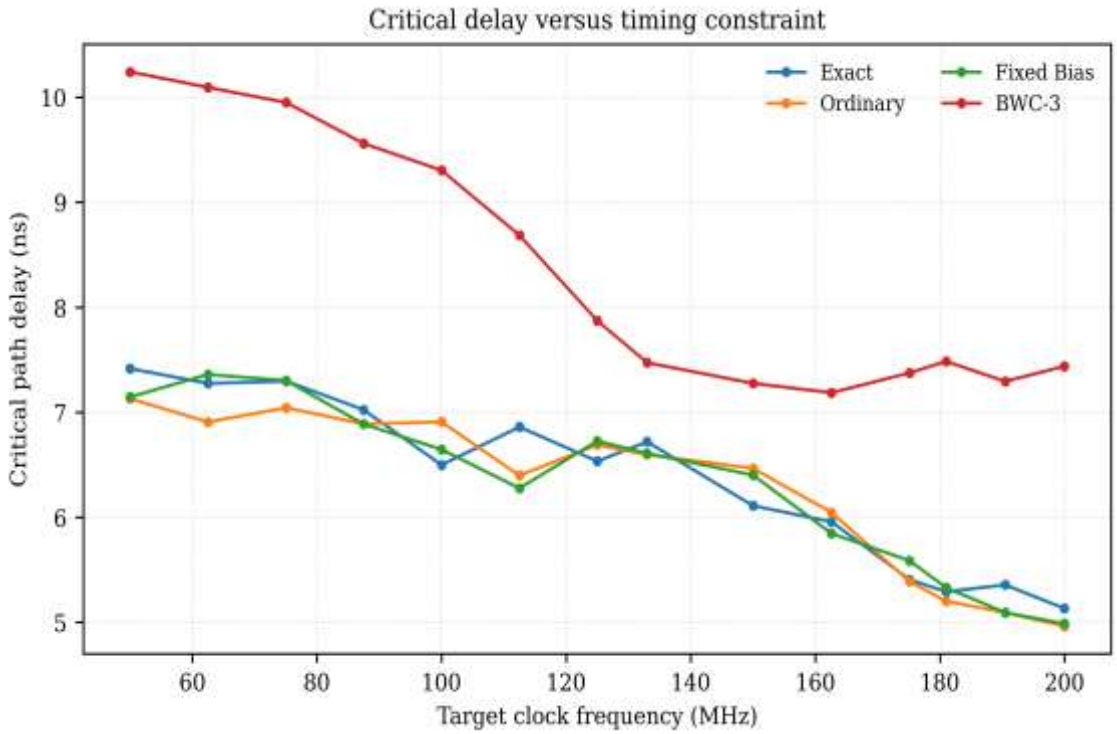


Fig.3. Post-implementation critical delay versus target frequency.

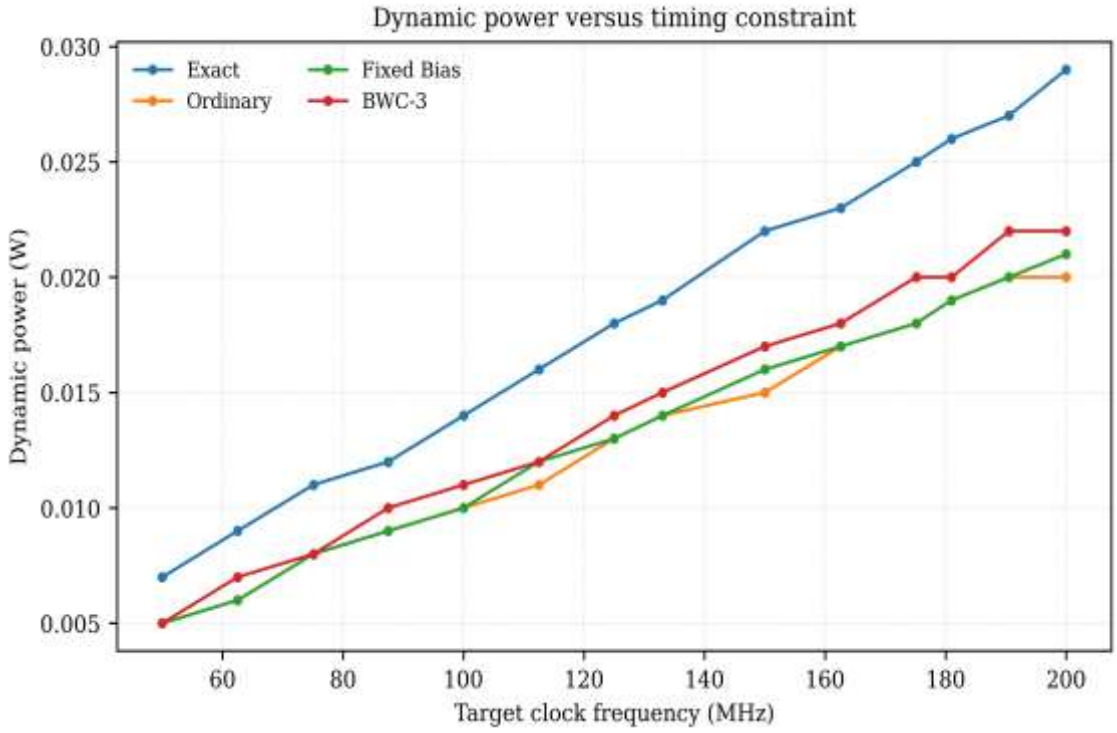


Fig. 4. Dynamic power versus target frequency.

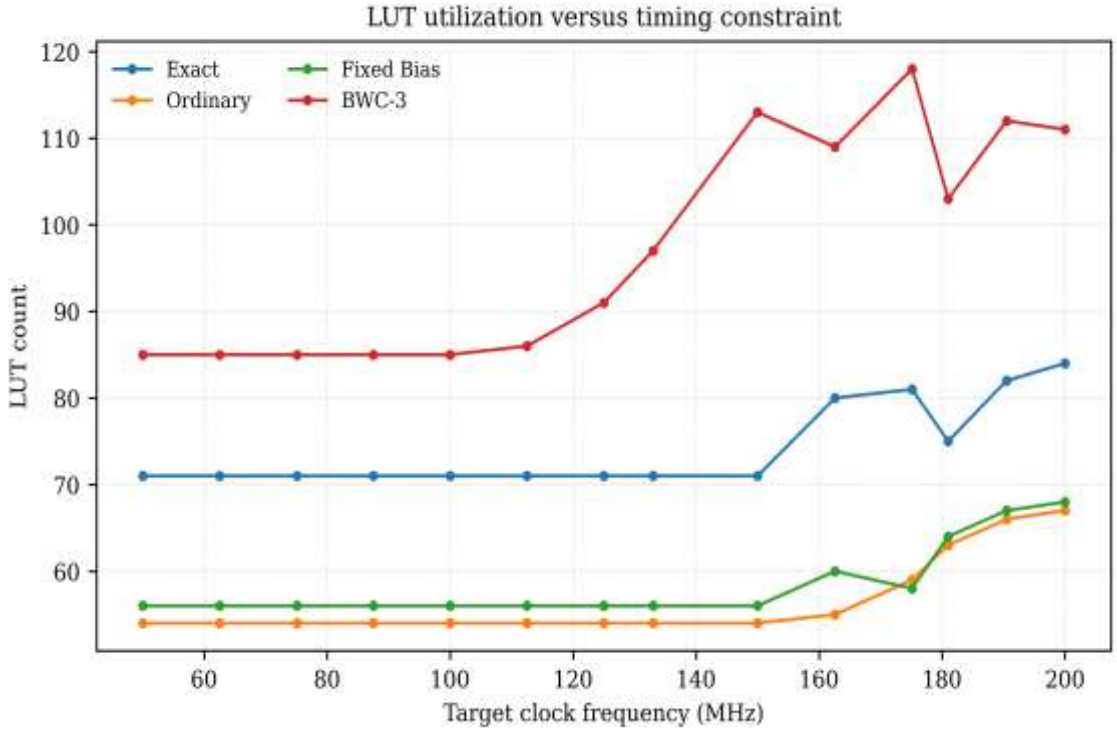


Fig. 5. LUT utilization versus target frequency.

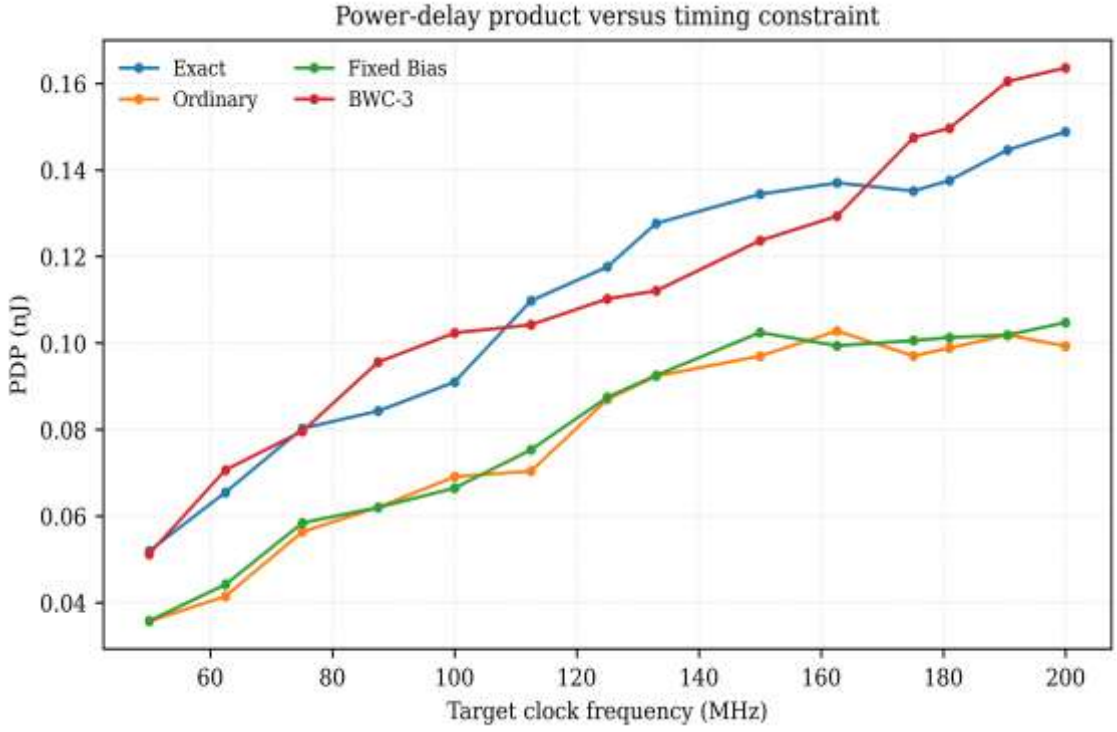


Fig. 6. Power-delay product versus target frequency.

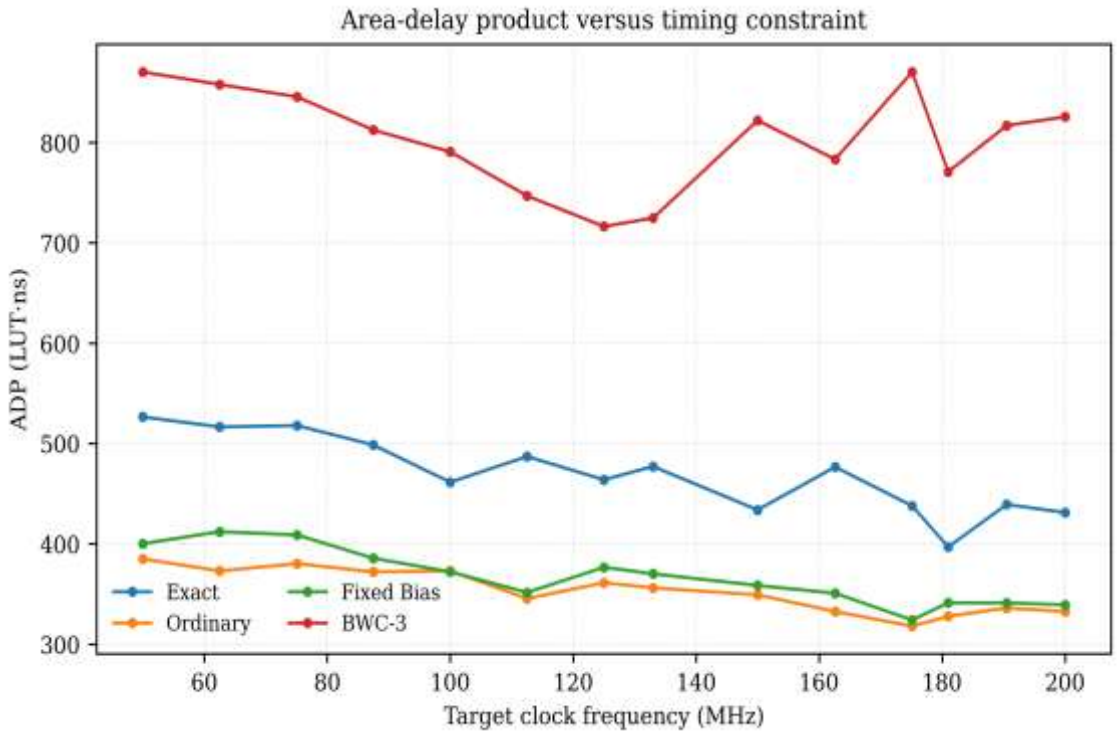


Fig. 7. Area-delay product versus target frequency.

The Fig. 2-7, presents why a multi-frequency methodology is more informative than a single implementation point for the timing sweep runs. The exact reference meets timing-closures through 180.995 MHz (WNS = 0.233 ns) but reports −0.094 ns at 190.5 MHz and −0.132 ns at 200 MHz. the ordinary truncation and fixed bias multiplier remains positive at 200 MHz with WNS = 0.036 ns and 0.012 ns values respectively. The BWC-3 model is positive through 133 MHz (0.047 ns) and whereas it becomes negative at 150 MHz (−0.605 ns) remains forever. For the practical design models the apparent Fmax estimates at failed target constraints should not be considered as timing closure.

At 200 MHz, ordinary truncation utilizes 67 LUTs, fixed bias 68 LUTs, compared with 84 LUTs in case of exact reference. The fixed bias therefore decrease LUT count by 19.0% and dynamic power by 27.6% relative to the exact model design at the same target constraint. And the PDP is 0.1047 nJ versus 0.1488 nJ for the exact reference, while ADP elevated down from 431.09 to 339.18 LUT•ns. The ordinary truncation multiplier model is slightly have less area than fixed bias, as expected from the absence of correction logic. In the case of BWC-3, it requires 111 LUTs at 200 MHz and has a longer 7.438-ns critical path, it showing that the accuracy improvement is purchased with substantial implementation cost in the present hardware realization.

IV. CONCLUSION

A unique14-point FPGA timing sweep was planned to compare the exact, ordinary truncated, fixed-bias, and BWC-3 8-bit multipliers using integrated wrapper. The measurements reveal a clear trade-off among timing robustness, and implementation hardware cost. The ordinary truncation multiplier is the smallest design and meets timing at 200 MHz, fixed bias also preserving 200-MHz timing closure and retaining lower LUT, dynamic-power, PDP, and ADP values compare with the exact reference. The BWC-3 architecture have the highest arithmetic accuracy, but its additional compensation logic limits measured timing closure to 133 MHz and increases LUT/ADP cost for hardware. For high-throughput vision architecture datapaths, the observed data thus identified fixed bias as a balanced implementation point, while BWC-3 is better for accuracy demanding operations.

REFERENCES

[1] N. Petra, D. De Caro, V. Garofalo, E. Napoli, and A. G. M. Strollo, “Truncated binary multipliers with variable correction and minimum mean square error,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 6, pp. 1312–1325, Jun. 2010, doi: 10.1109/TCSI.2009.2033536.
[2] V. Garofalo, N. Petra, and E. Napoli, “Analytical calculation of the maximum error for a family of truncated multipliers providing minimum mean square error,” IEEE Trans. Comput., vol. 60, no. 9, pp. 1366–1371, Sep. 2011, doi: 10.1109/TC.2010.236.

[3] N. Petra, D. De Caro, V. Garofalo, E. Napoli, and A. G. M. Strollo, "Design of fixed-width multipliers with linear compensation function," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 58, no. 5, pp. 947–960, May 2011, doi: 10.1109/TCSI.2010.2090572.
[4] G. Zervakis, K. Tsoumanis, S. Xydis, D. Soudris, and K. Pekmestzi, "Design-efficient approximate multiplication circuits through partial product perforation," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 24, no. 10, pp. 3105–3117, Oct. 2016, doi: 10.1109/TVLSI.2016.2535398.
[5] S. Vahdat, M. Kamal, A. Afzali-Kusha, and M. Pedram, "TOSAM: An energy-efficient truncation- and rounding-based scalable approximate multiplier," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 27, no. 5, pp. 1161–1173, May 2019, doi: 10.1109/TVLSI.2018.2890712.
[7] Y. Wu, C. Chen, W. Xiao, X. Wang, C. Wen, J. Han, X. Yin, W. Qian, and C. Zhuo, "A survey on approximate multiplier designs for energy efficiency: From algorithms to circuits," ACM Trans. Design Autom. Electron. Syst., vol. 29, no. 1, Art. no. 23, pp. 1–37, Jan. 2024, doi: 10.1145/3610291.
[8] L. Sayadi, S. Timarchi, and A. Sheikh-Akbari, "Two efficient approximate unsigned multipliers by developing new configuration for approximate 4:2 compressors," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 70, no. 4, pp. 1649–1659, Apr. 2023, doi: 10.1109/TCSI.2023.3242558.
[9] S. Shakibhamedan, N. Amirafshar, A. S. Baroughi, H. S. Shahhoseini, and N. TaheriNejad, "ACE-CNN: Approximate carry disregard multipliers for energy-efficient CNN-based image classification," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 71, no. 5, pp. 2280–2293, May 2024, doi: 10.1109/TCSI.2024.3369230.
[10] Z. Li, S. Zheng, J. Zhang, Y. Lu, J. Gao, J. Tao, and L. Wang, "Adaptable approximate multiplier design based on input distribution and polarity," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 30, no. 12, pp. 1813–1826, Dec. 2022, doi: 10.1109/TVLSI.2022.3197229.
[11] P. Alamuri, U. A. Kumar, V. Vannuru, and S. E. Ahmed, "Improved approximate multiplier architecture for image processing and neural network applications," Microprocessors Microsyst., vol. 101, Art. no. 104909, Sep. 2023, doi: 10.1016/j.micpro.2023.104909.
[12] P. Anguraj and K. Thiruvenkadam, "Design and realization of area-efficient approximate multiplier structures for image processing applications," Microprocessors Microsyst., vol. 102, Art. no. 104925, 2023, doi: 10.1016/j.micpro.2023.104925.
[13]. Kuldeep Singh, Ajay Kumar, Meenakshi, Rajesh Singh, Vinod Kumar, "FPGA Performance Characterization of an 8-Bit Ordinary Truncated Multiplier under Progressive Timing Constraints," International Journal of Scientific Research in Engineering and Management, vol. 10, no. 9, pp. 1-7, Sep 2026. https://doi.org/10.55041/IJSREM67221
[14]. Kuldeep Singh, Vinod Kumar, & Ajay Kumar. (2026). A Multi-Frequency Performance Evaluation of an 8-Bit Fixed-Bias Truncated Approximate Multiplier Design on FPGA. International Research Journal of Innovations in Engineering and Technology - IRJIET, 10(9), 92-99. Article DOI https://doi.org/10.47001/IRJIET/2026.109010