On the Design of Novel XOR-based Attention Mechanism for Enhanced Efficiency of Transformers

Sumit Kumar Jha
jha@cs.fiu.edu
Computer Science Department
Florida International University
Miami, FL, USA

Susmit Jha
susmit.jha@sri.com
Computer Science Laboratory
SRI International
Menlo Park, CA, USA

Rickard Ewetz
rickard.ewetz@ucf.edu
Electrical and Computer Engineering
University of Central Florida
Orlando, FL, USA

Alvaro Velasquez
alvaro.velasquez@colorado.edu
Department of Computer Science
University of Colorado Boulder
Boulder, CO, USA

ABSTRACT

We present a new xor-based attention function for efficient hardware implementation of transformers. While the standard attention mechanism relies on matrix multiplication between the key and the transpose of the query, we propose replacing the computation of this attention function with bitwise xor operations. We mathematically analyze the information-theoretic properties of the standard multiplication-based attention, demonstrating that it preserves input entropy, and then computationally show that the xor-based attention approximately preserves the entropy of its input despite small variations in correlations between the inputs. Across various admittedly simple tasks, including arithmetic, sorting, and text generation, we show comparable performance to baseline methods using scaled GPT models. The xor-based computation of the attention function shows substantial improvement in power consumption, latency, and circuit area compared to the corresponding multiplication-based attention function. This hardware efficiency makes xor-based attention more compelling for the deployment of transformers under tight resource constraints, opening new application domains in sustainable energy-efficient computing. Additional optimizations to the xor-based attention function can further improve efficiency of transformers.

ACM Reference Format:
Sumit Kumar Jha, Susmit Jha, Rickard Ewetz, and Alvaro Velasquez. 2024. On the Design of Novel XOR-based Attention Mechanism for Enhanced Efficiency of Transformers. In *61st ACM/IEEE Design Automation Conference (DAC '24), June 23–27, 2024, San Francisco, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3649329.3658253

DAC '24, June 23–27, 2024, San Francisco, CA, USA

ACM ISBN 979-8-4007-0601-1/24/06…$15.00
https://doi.org/10.1145/3649329.3658253

1 INTRODUCTION

Transformer-based artificial intelligence models have demonstrated impressive and hitherto unforeseen capabilities, but their exponentially growing computational demands raise urgent questions about both operating costs and sustainability. Data centers and data networks consume massive amounts of energy [6, 18], accounting for an estimated 2%-3% of global electricity use in 2023. As human interfaces for discovering knowledge on the Internet transition from search engines to large language models, and users switch from search engines to conversational queries on models like Claude, Gemini, GPT, Grok, or public models, AI could end up consuming an even more significant portion of worldwide energy production [13, 15]. This trajectory is clearly not sustainable.

Our work seeks to tackle this challenge by redesigning the attention function that is central to large language models and other transformers. Attention enables models to focus on the relevant context by selectively weighting inputs. However, the predominant approach relies on computationally intensive matrix multiplications, which makes attention a performance bottleneck. Inspired by the similarity in information-theoretic properties of multiplication and the xor operation, we aim to overcome this limitation by substituting matrix multiplication in the attention function with xor operations during the calculation of the attention. Although simple, this change represents a significant shift from mainstream practices. We demonstrate its viability on admittedly simple examples and quantify substantial improvements in power, latency, and area needs of hardware implementation.

Prior work to improve the efficiency of the attention mechanism has focused primarily on three areas: algorithmic advancements such as sparse and linear attention to reduce token interactions [2, 21], model optimizations such as knowledge distillation and pruning to compress models [4, 9, 11], and hardware acceleration including circuit design and parallelization [1, 15]. However, these approaches do not directly address the core complexity of matrix multiplications and consequent inevitable hardware inefficiencies. In contrast, our approach reshapes the computation of the attention function based on information-theoretic insights about entropy preservation,

diverging from current approaches and paving the way for potentially useful advancements along these lines in the future. While there has been work [23] on applying xor and other Boolean gates to quantized neural networks in general, we are the first to identify that attention can be accelerated using xor-based approaches. The key contributions of our work are as follows:

(1) We present a new XOR-based attention function that can lead to the design of transformer architectures and large language models optimized for hardware efficiency, thereby introducing a new computation approach that significantly reduces the energy, latency, and hardware complexity of such models.
(2) Our work offers an information-theoretic analysis of the standard attention function and shows that the attention function preserves the entropy of at least one of its inputs under mild technical conditions, including those imposed by Ostrowski's theorem [22]. Then, we use computational analysis to show that the XOR-based attention function approximately retains the flow of information from its inputs to its output despite small variations in the correlation between the inputs.
(3) We validate our theoretical propositions through experiments such as sorting, arithmetic, and text generation, showcasing the potential of our proposed design in various admittedly simple scenarios.
(4) We show that the 16-bit XOR-based attention function is 13X more energy efficient, 5X faster, and 39X smaller in area than the multiplier-based approach.

In summary, our work reveals a promising path towards developing large language models and other transformer models that are more efficient. The principle of logic-based attention we establish can potentially shape new approaches for efficient and sustainable AI.

2 RELATED WORK

Prior work has attempted to enhance the efficiency of attention mechanisms through three main approaches: algorithmic advancement, model optimization, and hardware acceleration. Our description of the related work omits several interesting and important results due to space constraints.

2.1 Algorithmic Advancement

Several studies [2, 21] have sought to improve the computation of the attention operation using ideas about the sparsity of the attention matrix and its low rank. For example, the Sparse Transformer reduces the complexity to $O(n\sqrt{n})$ and the Reformer speeds up the attention operation to $O(n \log n)$. The Linformer uses the Johnson-Lindenstrauss lemma and an existential argument to further reduce the complexity of the operation by finding a low-dimension projection that is logarithmic in the original dimension of the attention matrix. These improvements still rely on matrix multiplication and are inherently inefficient for hardware implementation. Further, our approach is orthogonal to these ideas and may be suitable adapted for use in conjunction with them, replacing multiplication with Boolean XOR operations.

2.2 Model Optimization

Quantization and pruning coupled with knowledge distillation has been particularly helpful in enhancing the efficiency of transformer models [4, 9]. Quantization has enabled the deployment of large language models on commodity GPU hardware by deploying models with as few as 4 bits. More recently, 1-bit LLMs have been proposed that greatly push the boundaries of quantization [11, 20]. Most of these approaches to optimize models, such as knowledge distillation and quantization, will remain applicable to our method.

2.3 Hardware Acceleration

There has been a lot of work on accelerating neural networks, including transformers by creating specialized accelerators. We present a new attention function that is hardware-friendly and can be implemented efficiently on multiple hardware platforms, including existing CMOS and upcoming in-memory computing technologies [5, 8, 12].

Efforts have been made to use Boolean logic [23] for binary neural networks. In a recent approach [14], the transformer was co-designed on an FPGA using sparse attention and dynamic pipelining. The softmax function has been optimized for use in transformers using a hardware-software co-design approach [16]. Mixed-scheme quantization has been used to design FPGA solutions [19] for transformers used in computer vision. Hardware optimizations such as token pruning and other methods have been used to accelerate Bayesian transformers and transformers deployed on crossbars [3, 10].

In contrast, our approach is different because we redesign attention based on fundamental information-theoretic insights on the entropy preservation properties of the multiplication and the XOR operation. This departs significantly from current approaches and opens up new avenues for algorithmic, model, and hardware optimizations of the new attention function.

3 ATTENTION: BEYOND MULTIPLICATION

We briefly discuss the attention function, and then motivate the design of a new xor-based attention function using information-theoretic arguments about the traditional attention function that essentially involves matrix multiplication. Then, we computationally show that an xor-based attention function approximately preserves the information-theoretic properties of the traditional attention function.

3.1 Attention Mechanism

In neural networks such as transformers, the attention mechanism is used to focus on relevant sections of the inputs depending on the context of the task and thereby enable both efficient and adaptive learning. As a result of the success of the attention mechanism, transformers have become ubiquitous in large language models, vision language models, and even scientific foundation models.

Mathematically speaking, the attention computation involves three primary components:

(1) Queries: The query represents the current position in the sequence whose attention is being computed. Given an input

X, the query $Q = W_Q X$ uses a tensor W_Q that can be learned to identify the current position in the input sequence.
(2) Keys: Given a query, the key represents all the positions in the input sequence that the query must pay attention. Given an input X, the key $K = W_K X$ uses a learnable tensor W_K to identify the position that should be "attended to" by the query vector.
(3) Values: Given an input X, the value $V = W_V X$ uses a learnable tensor to identify the values that should be weighed by the computed attention to produce the final output of the attention mechanism.

Given the query, the key and the value vectors, the attention mechanism is best described as the scaled dot-product between the query and the key being used to compute a weighted version of the value vector. Formally, we say that,

$$\mathrm{A}(Q, K) = QK^T \tag{1}$$

$$\mathrm{Attn}(Q, K) = \mathrm{softmax}\left(\frac{\mathrm{A}(Q, K)}{\sqrt{d_k}}\right)V \tag{2}$$

where d_k is the dimensionality of the keys. Here, $\mathrm{A}(Q, K)$ is the attention function and Attn is the attention output. Our emphasis in this paper is the redesign of the attention function $\mathrm{A}(Q, K)$. To this end, we first study the information-theoretic properties of the classical attention function and then discover that the XOR operation shows similar properties.

3.2 Information Theoretic Motivation

We motivate our redesign of the attention function $\mathrm{A}(Q, K)$ by focusing on the information flow through this function. Ideally, an attention function may not cause a significant loss in the information content of the output as compared to the information content of the input. We denote by $J(M)$ the singular value decomposition (SVD) entropy of the matrix M. Given the singular values σ_i^* of the matrix M, its SVD entropy is computed as follows:

$$J(M) = -\frac{1}{\log k}\sum \sigma_i \log \sigma_i \tag{3}$$

Here, k is the number of singular values and $\sigma_i = \sigma_i^*/S_M$ is the normalized singular value, and $S_M = \sum \sigma_i^*$ is the sum of singular values for the matrix M.

In particular, the SVD entropy of the Q matrix is given by $J(Q) = -\frac{1}{\log k}\sum \sigma_i^Q \log \sigma_i^Q$, where σ_i^Q are the normalized singular values of Q. Similarly, the SVD entropy of a K matrix: $J(K) = -\frac{1}{\log k}\sum \sigma_i^K \log \sigma_i^K$. We recall that a matrix and its transpose have identical singular values. For the sake of simplicity, we assume that both Q and K are non-singular square matrices with a Frobenius norm of unity.

Theorem 3.1 (Ostrowski's Theorem [22]). *Let $\sigma_0^*(X), \ldots \sigma_k^*(X)$ denote the singular values of a non-singular matrix X and two n-square complex matrices A and B. Let f be a symmetric function of n real variables such that $f^{exp}(x_1, \ldots, x_n) = f(e^{x_1}, \ldots, e^{x_n})$ is increasing in each x_i and convex in the region $x_i \geq 0$, then*

$$f(\sigma_1^*(AB), \ldots, \sigma_k^*(AB)) \leq f(\sigma_1^*(A)\sigma_1^*(B), \ldots, \sigma_k^*(A)\sigma_k^*(B)) \tag{4}$$

Lemma 3.2. *The function $f(x_1, \ldots, x_n) = \sum x_i/C \log(x_i/C)$, where C is a constant, satisfies the conditions of Ostrowski's theorem.*

Proof. We will show that the function, f, satisfies the conditions of the Ostrowski's theorem:

(1) The function f is a symmetric function of n real variables: $x_1, \ldots x_n$.
(2) $f^{exp}(x_1, \ldots, x_n) = \sum e^{\frac{x_i}{C}} \log \frac{x_i}{C} = \sum \frac{x_i}{C} e^{\frac{x_i}{C}}$ is increasing in each variable x_i since its derivative is $\frac{Ce^{x_i/C} + x_i e^{x_i/C}}{C^2} \geq 0$ for all $x_i \geq 0$.
(3) $f^{exp}(x_1, \ldots, x_n) = \sum e^{\frac{x_i}{C}} \log \frac{x_i}{C} = \sum \frac{x_i}{C} e^{\frac{x_i}{C}}$ has a second derivative of $\frac{e^{x_i/C} x_i + 2ce^{x_i/C}}{C^3} \geq 0$ for all $x_i \geq 0$. Hence, $f^{exp}(x_1, \ldots, x_n)$ is convex. □

Under the technical conditions imposed by the Ostrowski's theorem, we will establish the result showing that the SVD entropy of the product is at least as large as the SVD entropy of one of the input matrices, i.e., information is retained during the computation of the attention function in traditional attention mechanisms.

Lemma 3.3. $J(A(Q, K)) \geq \min(J(Q), J(K))$

Proof. Consider a function related to the SVD entropy, $f(x_1, \ldots, x_n) = \sum x_i/S_A \log(x_i/S_A)$, where S_A is the sum of singular values of the attention function. From Theorem 3.1 and Lemma 3.2, we get the following:

$$\sum \frac{\sigma_i^*(QK)}{S_A} \log\left(\frac{\sigma_i^*(QK)}{S_A}\right) \leq \sum \frac{\sigma_i^*(Q)\sigma_i^*(K)}{S_A} \log\left(\frac{\sigma_i^*(Q)\sigma_i^*(K)}{S_A}\right) \tag{5}$$

Now, the term on the left hand side can be connected to the entropy of the attention from the definition of the attention function and the SVD entropy:

$$-(\log k) J(A) = \sum \frac{\sigma_i^*(QK)}{S_A} \log\left(\frac{\sigma_i^*(QK)}{S_A}\right) \tag{6}$$

The term on the right side of Eqn 5 can be simplified to Eqn. 7 using the fact that $\sum \sigma_i^*(QK) \leq \sum \sigma_i^*(Q)\sigma_i^*(K)$, that is, $\sum \sigma_i^*(QK) \leq \sum \sigma_i^*(Q)$ and $\sum \sigma_i^*(QK) \leq \sum \sigma_i^*(K)$ for matrices Q and K with Forbenius norm 1. We recall that singular values are always non-negative. Eqn. 7 holds true for both $S = S_k$ and $S = S_Q$.

$$\sum \frac{\sigma_i^*(Q)\sigma_i^*(K)}{S_A} \log\left(\frac{\sigma_i^*(Q)\sigma_i^*(K)}{S_A}\right) \leq \frac{\sigma_i^*(Q)\sigma_i^*(K)}{S} \log\left(\frac{\sigma_i^*(Q)\sigma_i^*(K)}{S}\right) \tag{7}$$

$$\leq \max\left(\frac{\sigma_i^*(K)}{S_K} \log\left(\frac{\sigma_i^*(K)}{S_K}\right), \frac{\sigma_i^*(Q)}{S_Q} \log\left(\frac{\sigma_i^*(Q)}{S_Q}\right)\right) \tag{8}$$

$$\leq \max\left(-(\log k) J(Q), -(\log k) J(K)\right) \tag{9}$$

Eqn. 8 follows from Eqn. 7 by using the fact that Q and K have a Forbenius norm of 1.

Putting together Eqn. 5 and Eqn. 9, we get the following:

$$J(A) \geq \min(J(Q), J(K)) \tag{10}$$

Thus, the SVD entropy of the attention function is at least as large as the smallest of the SVD entropy of the query Q and key K terms. □

3.3 Information Theory and Bitwise Operations

In the previous section, we saw that the traditional attention operation has an interesting information-theoretic interpretation as it does not lead to a loss of information. The design of our new attention function is based on our analysis of the information flow between the output and the input of bit-vector operations. A bitwise operation between two bit-vectors that causes a significant loss of information may be undesirable for use as an attention function. In Table 1, we show how different bitwise operations, such as or (nor), and (nand), and xor differently influence the flow of information from their inputs to the outputs by measuring the difference in the entropy of the inputs and the entropy of the output. The table shows inputs with 16 bits and shows results for a million samples per entry.

Table 1: Influence of Bitwise Operations on Information Flow

Operation	Bit Length	Correlation	Difference Entropy
		-0.5	-20.60%
AND	16-bits	0.0	-19.02%
		0.5	-18.07%
		-0.5	-20.61%
OR	16-bits	0.0	-19.01%
		0.5	-18.11%
		-0.9	-4.61%
		-0.5	-0.76%
XOR	16-bits	0.0	-0.15%
		0.5	-0.77%
		0.9	-4.61%

Table 1 shows how the bitwise XOR function is more effective in propagating the entropy of its inputs to the outputs, and other bitwise operators are not as effective at transmitting information from the query and the key vector to the attention function. Based on this motivation, we define a new XOR-based attention function, as defined below:

$$\mathrm{A}^{\oplus}(Q,K) = Q \oplus K^T \tag{11}$$

$$A_{ij} = \sum_{k=1}^{n} Q_{ik} \oplus (K^T)_{kj} \tag{12}$$

We discretize the query and the key vectors before computing the attention using the XOR operation above, and then employ the new attention function to weigh the value items as earlier.

$$\mathrm{Attn}^{\oplus}(Q,K) = \mathrm{softmax}\left(\frac{\mathrm{A}^{\oplus}(Q,K)}{\sqrt{d_k}}\right)V \tag{13}$$

where d_k is the dimensionality of the keys. Here, $\mathrm{A}(Q,K)$ is the attention function and Attn is the attention output.

4 EXPERIMENTAL RESULTS

We evaluate the efficacy of our XOR-based attention function by replacing the traditional attention function with this new function in small-scale variants of the GPT model, including GPT-2. Our goal is to show that the XOR-based attention function is capable of learning concepts and shows comparable performance with the traditional attention mechanism on admittedly simple tasks. We present a quantitative analysis of learning problems such as sorting and arithmetic, and the change in energy, area and latency requirements of the hardware implementation. We perform a qualitative analysis on the text generation task.

Table 2: Attention mechanism comparison – sorting.

Length of Input	Number of Distinct Integers	Attention Mechanisms	
		Multiplication	XOR
10	3	100.00%	100.00%
10	5	99.94%	99.96%
10	7	99.24%	99.84%
10	9	99.84%	99.46%
20	3	99.78%	99.78%
20	5	99.48%	97.92%
20	7	99.60%	99.46%
20	9	99.52%	98.84%

All of our experiments were performed on a single node with 112 processor cores, 1.5 TB of RAM, and 8 A100 GPUs each having 80GB of RAM. The relatively small and simple examples demonstrate that the transformer model can learn using this new XOR-based attention function. Future research will explore the effectiveness of the novel attention function within larger-scale, instruction-tuned language models, promising even more informative evaluation and deeper insights in this design.

4.1 Sorting with GPT-nano

In our first experiment, we sought to learn how to sort sequences using the new XOR-based attention mechanism that we have developed, and compare it with the traditional attention based on multiplication. The essence of the learning problem is as follows: Given a random list of integers, the model aims to produce a sorted version of that list. This basic sorting task provides a platform for understanding and evaluating the model's capability to recognize patterns and apply logical sequencing rules, as well as to evaluate the efficacy of our new XOR-based attention function.

The model, GPT-nano, is a scaled-down variant of larger GPT models, tailored for reduced complexity operations. It comprises of 3 layers with 3 attention heads each and an embedding dimension of 48, substantially smaller than its full-sized counterparts. This model is trained on a custom dataset comprising sequences of integers of fixed length, where the target is the sorted sequence. The data set is generated to ensure a blend of difficulty levels by occasionally amplifying the instances with a higher number of repeated integers, which represents scenarios that challenge the model. The data set is divided into training and test sets, with a distribution ratio that ensures 25% of the examples are reserved randomly for testing, selected via a hashing mechanism. The sequences fed into the transformer are preprocessed to contain both the input and the output concatenated, with the latter shifted by one position. Training is carried out using an AdamW optimizer with a learning rate of 10^{-4} and 10,000 iterations. During evaluation, the model is tasked with sorting unseen sequences and its predictions are compared against the ground truth.

Table 2 shows the results of sequences of length 10 and 20 involving 3 to 9 distinct integers. Traditional multiplication-based attention achieves an average accuracy of 99.68% while the XOR-based mechanism achieves an average accuracy of 99.40%.

4.2 Adding with GPT

We assessed our implementation of XOR-based attention mechanism in generative pre-trained transformer models using the addition of two inputs with a small number of digits. The objective is to investigate the efficacy of this novel attention mechanism in this arithmetic computational task, and compare its performance with the conventional multiplication-based attention mechanism.

Three GPT models were selected for this experiment: GPT-nano, GPT-mini, and GPT-2. GPT-nano is a highly compact model designed for efficient computation that was discussed earlier. The GPT-mini model is designed as a scaled-down version of larger models, it comprises of 6 layers, 6 heads, and an embedding size of 192. This model strikes a balance between computational efficiency and the ability to handle moderately complex tasks. The vanilla GPT-2 model consists of 12 layers, 12 heads, and an embedding size of 768, totaling approximately 124 million parameters. This configuration represents a fairly complex benchmark that can still be successfully trained on A100 GPUs.

The diverse nature of these models allows for an investigation into the effectiveness of the XOR-based attention mechanism across different scales of model architecture and complexity. Each model was evaluated using two distinct attention mechanisms: (i) standard multiplication-based attention as a control and (ii) XOR-based attention.

Table 3: Results by model and attention mechanism

Model	Number of Digits	Attention Mechanism	
		Multiplication	XOR
GPT-nano	2	100.0%	100.0%
	3	100.0%	100.0%
	4	100.0%	99.90%
GPT-mini	2	100.0%	100.0%
	3	100.0%	100.0%
	4	100.0%	100.0%
GPT2	2	100.0%	100.0%
	3	100.0%	100.0%
	4	100.0%	99.90%

As shown in Table 3, XOR-based attention mechanism shows comparable and sometimes equivalent performance to the traditional multiplication-based attention despite the simplicity and efficiency of the XOR operation. This clearly demonstrates that the XOR-based attention function is capable of learning for such tasks in transformer models.

We compared our new XOR-based attention function with standard multiplication-based attention in the context of neural machine translation. We utilize the GPT-2 model and the Multi30k German-English translation task as our testbed. The performance of each model is evaluated quantitatively on the basis of the loss function. The XOR-based attention does not perform well compared to the multiplication-based attention and achieves a higher loss. We attribute this to the fact that the optimization procedure for the XOR-based attention function may need to be customized.

Table 4: Metrics for Multiplication with Different Bit-widths

Bit-width	Power (mW)	Latency (ns)	Area (um^2)
4 bits	0.02	1.18	147.83
6 bits	0.06	1.51	358.08
8 bits	0.12	1.83	658.43
10 bits	0.19	2.16	1048.89
16 bits	0.52	3.88	2966.45

4.3 Shakespeare Dataset using GPT-mini

We trained the GPT-mini model with 6 layers, 6 heads and 192 dimensional embedding from scratch on the Shakespeare data set using the standard attention and our XOR-based attention function. Here is a sample response from the GPT-2 model that was trained using a multiplication-based attention mechanism.

```
O God, O God!
  SLY. A heart's no lord haim for that he seas the deed matting
    throughly.
  POST. Marry haphe all!
  FIRST OBY. What in honest fors so discove, suse my suffer iends!
  SIR JOHN. I will prosce me ithe wear he what I as true his? No
                                        Exeunt PISANIO
   BYETR. Thy slee ave the most dog.
 CASSIUS. Say, they have this soner stand.
  SICINIUS. Ay, madam, felel!
  THURIO. Neer, fie; in my past ble indure wash an thou,
    And I'll be a womine my the lork.
```

The response from the GPT-2 model accelerated using the XOR-based attention is shown below.

```
O God, O God! fellow! Who's the breast as
     makes a company? O leave a whipt? Ho! A pollot of women!
    angels! He is hear! He shath day whill sain.
  HOLASTESSS. And have to wof true he!
  CAMILLO. Who has you then better?
  HESPEED. Hath, I honest the hour of his greetingesss
    And a man saffections.
  CALIBAN. Hear you.
  SON. CHALLOTES. Here hope disciplin'd where was this
    A wand we furying.
  FRENCHMENAS. O loutse man in the dead oof,
    And much they darest moonshing all forese,
    Betwee
```

The standard and the XOR-based approach both get the structure of the response correct in terms of identifying dialogue and the specific character of the sentences in the play. However, due to the small size of the models, both of the trained models fail to achieve the kind of realistic responses that larger models, such as GPT-4, are able to produce. The experiment shows that the XOR-based attention approach has similar qualitative performance in this setting as the standard attention operation.

5 EFFICIENCY: MULTIPLICATION VS. XOR

We synthesize logic circuits using the Synopsis Design Compiler with the 45nm FreePDK45 standard cell library [17]. We employ the compile_ultra command and utilize the DC Ultra synthesis mode to generate gate-level netlists for the designs. We set the area as unconstrained to optimize for power and latency.

Table 4 shows the size of the multiplier circuit required to implement the multiplication of different bit widths. We have

Table 5: Metrics for XOR with Different Bit-widths

Bit-width	Power (mW)	Latency (ns)	Area (um^2)
4 bits	0.01	0.69	18.76
6 bits	0.02	0.69	28.14
8 bits	0.02	0.69	37.52
10 bits	0.03	0.69	46.90
16 bits	0.04	0.69	75.04

implemented the fixed bit-width multiplier to compare it with our fixed-width XOR logic. As shown in Table 4, increasing the bit width from 4 bits to 16 bits leads to a 26X increase in the power required for the computation. Similarly, the area required to perform the computation increases by a factor of 20.

Our XOR-based attention function uses bitwise XOR between the query and the key matrices. We study the size, energy, and time required to implement the XOR computation for bit vectors with varying bit widths. In Table 5, we show that the power and computation time required by the XOR operation increase linearly with the bit width of the input. This is in sharp contrast to the rapid growth in the compute energy and compute time faced by the implementation of the multiplication operation.

Table 4 and 5 show that the power requirement of the XOR implementation is 6X smaller for 8 bits and about 13X smaller than the multiplier for 16 bits. Similarly, the latency for the 16-bit XOR implementation is about 5X lower than that of the corresponding multiplier. The area needed for the XOR implementation is 39X smaller than that of the multiplier for 16 bits.

6 CONCLUSIONS

We present a new XOR-based attention function calculation that enables substantial reductions in energy consumption, latency, and area compared to the matrix multiplication-based attention function computation. We validate the potential of XOR attention in various simple tasks, including arithmetic, sorting, and text generation using scaled GPT models. Our approach realizes substantial hardware benefits, as illustrated in our prototypical evaluation on a 45 nm technology. Future work will include an evaluation of the new attention function on larger multi-modal models with more complex benchmarks, re-design of components of the attention mechanism that still use matrix multiplication, the use of automated synthesis methods [7], and exploring potential synergies with quantized LLMs [11].

ACKNOWLEDGEMENTS

The authors acknowledge support from NSF awards #2408925, #2404036, and DARPA awards FA8750-23-2-0501 and HR00112420004. This work was supported by the U.S. Army Research Laboratory Cooperative Research Agreement W911NF-17-2-0196. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the Department of Defense or the United States Government. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR) program, Reaching a New Energy Sciences Workforce (RENEW) under Award Number(s) DE-SC0024576 and DE-SC0024428.

REFERENCES

[1] Gang Chen, Shengyu He, Haitao Meng, and Kai Huang. 2020. Phonebit: Efficient gpu-accelerated binary neural network inference engine for mobile phones. In *2020 Design, Automation & Test in Europe Conference (DATE)*. IEEE, 786–791.

[2] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. 2020. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794* (2020).

[3] Hongxiang Fan, Martin Ferianc, and Wayne Luk. 2022. Enabling fast uncertainty estimation: accelerating bayesian transformers via algorithmic and hardware optimizations. In *2022 Proceedings of the 59th ACM/IEEE Design Automation Conference*. IEEE, 325–330.

[4] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2022. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*. Chapman and Hall/CRC, 291–326.

[5] Amad Ul Hassen, Dwaipayan Chakraborty, and Sumit Kumar Jha. 2018. Free binary decision diagram-based synthesis of compact crossbars for in-memory computing. *IEEE Transactions on Circuits and Systems II: Express Briefs* 65, 5 (2018), 622–626.

[6] IEA. 2023. Data centres & networks. https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks. [Accessed 2023].

[7] Susmit Kumar Jha. 2011. *Towards automated system synthesis using sciduction*. University of California, Berkeley.

[8] Sumit Kumar Jha, Dilia E Rodriguez, Joseph E Van Nostrand, and Alvaro Velasquez. 2016. Computation of boolean formulas using sneak paths in crossbar computing. US Patent 9,319,047.

[9] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351* (2019).

[10] Zhengang Li, Yanyue Xie, Peiyan Dong, Olivia Chen, and Yanzhi Wang. 2023. Algorithm-Software-Hardware Co-Design for Deep Learning Acceleration. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 1–4.

[11] Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. 2024. The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits. *arXiv:2402.17764* (2024).

[12] Jodh Singh Pannu, Sunny Raj, Steven Lawrence Fernandes, Dwaipayan Chakraborty, Sarah Rafiq, Nathaniel Cady, and Sumit Kumar Jha. 2020. Design and fabrication of flow-based edge detection memristor crossbar circuits. *IEEE Transactions on Circuits and Systems II: Express Briefs* 67, 5 (2020), 961–965.

[13] David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350* (2021).

[14] Hongwu Peng, Shaoyi Huang, Shiyang Chen, Bingbing Li, Tong Geng, Ang Li, Weiwen Jiang, Wujie Wen, Jinbo Bi, Hang Liu, et al. 2022. A length adaptive algorithm-hardware co-design of transformer on fpga through sparse attention and dynamic pipelining. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*. 1135–1140.

[15] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053* (2019).

[16] Jacob R Stevens, Rangharajan Venkatesan, Steve Dai, Brucek Khailany, and Anand Raghunathan. 2021. Softermax: Hardware/software co-design of an efficient softmax for transformers. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 469–474.

[17] James E Stine, Ivan Castellanos, Michael Wood, Jeff Henson, Fred Love, W Rhett Davis, Paul D Franzon, Michael Bucher, Sunil Basavarajaiah, Julie Oh, et al. 2007. FreePDK: An open-source variation-aware design kit. In *2007 IEEE international conference on Microelectronic Systems Education (MSE'07)*. IEEE, 173–174.

[18] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. *arXiv preprint arXiv:1906.02243* (2019).

[19] Mengshu Sun, Zhengang Li, Alec Lu, Haoyu Ma, Geng Yuan, Yanyue Xie, Hao Tang, Yanyu Li, Miriam Leeser, Zhangyang Wang, et al. 2022. Late Breaking Results: FPGA-Aware Automatic Acceleration Framework for Vision Transformer with Mixed-Scheme Quantization. In *Proceedings of the 59th Design Automation Conference (DAC)*.

[20] Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. 2023. Bitnet: Scaling 1-bit transformers for large language models. *arXiv preprint arXiv:2310.11453* (2023).

[21] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *arXiv:2006.04768* (2020).

[22] William Watkins. 1970. On the singular values of a product of matrices. *Journal of Research, National Bureau of Standards: Mathematics and mathematical physics. Section B* 74, 4 (1970), 311.

[23] Shien Zhu, Luan HK Duong, and Weichen Liu. 2020. XOR-Net: An efficient computation pipeline for binary neural network inference on edge devices. In *Proceedings of the 26th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 124–131.