

# FAMOUS: Flexible Accelerator for the Attention Mechanism of Transformer on UltraScale+ FPGAs



Download This

Poster

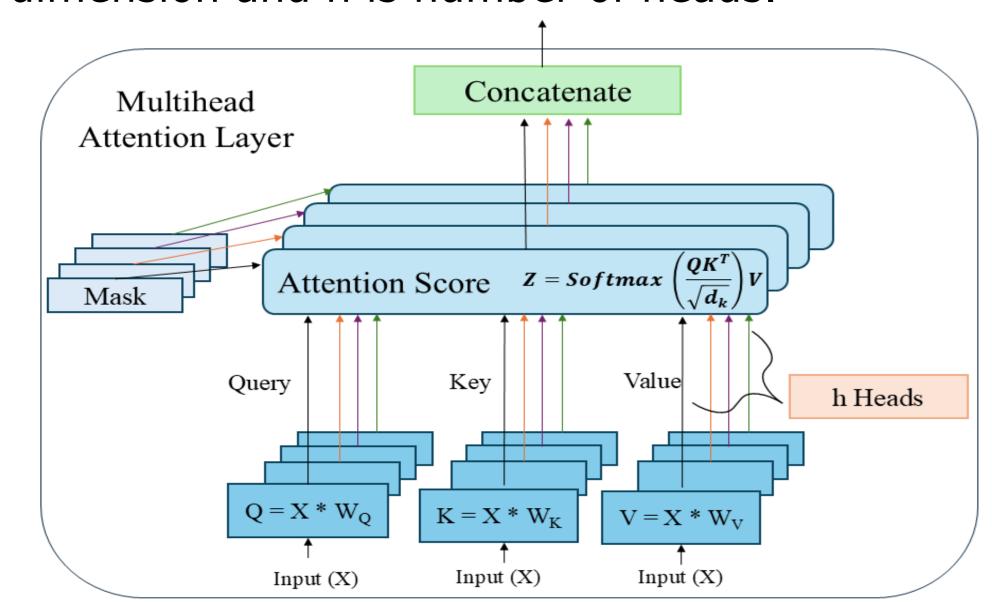
Ehsan Kabir, Md. Arafat Kabir, Austin R. J. Downey, Jason D. Bakos, David Andrews, Miaoqing Huang

#### Introduction

- Transformer neural networks (TNN) have demonstrated significant advancements in natural language processing, machine translation, computer vision.
- A remarkable feature named multi-headed attention (MHA) mechanism enables a high level of computational parallelism for both the training and inference phases.
- Suitable for acceleration on hardware like FPGAs, due to FPGA's high degree of parallelism, low latency, and energy efficiency.
- Most of the FPGA or ASIC-based accelerators for TNN have specialized sparse architecture for a specific application. Thus, they lack the flexibility to be reconfigured for a different model during runtime.
- We applied efficient tiling and wrote efficient high level synthesis (HLS) code to increase parallelism for dense computations of MHA.

### Background

The input sequence X is linearly mapped into Query (Q), Key (K), Value (V) matrices using weights and biases. The parameter dk = dmodel/h is the 2nd dimension of Q and K. dmodel is a hyperparameter called embedding dimension and h is number of heads.

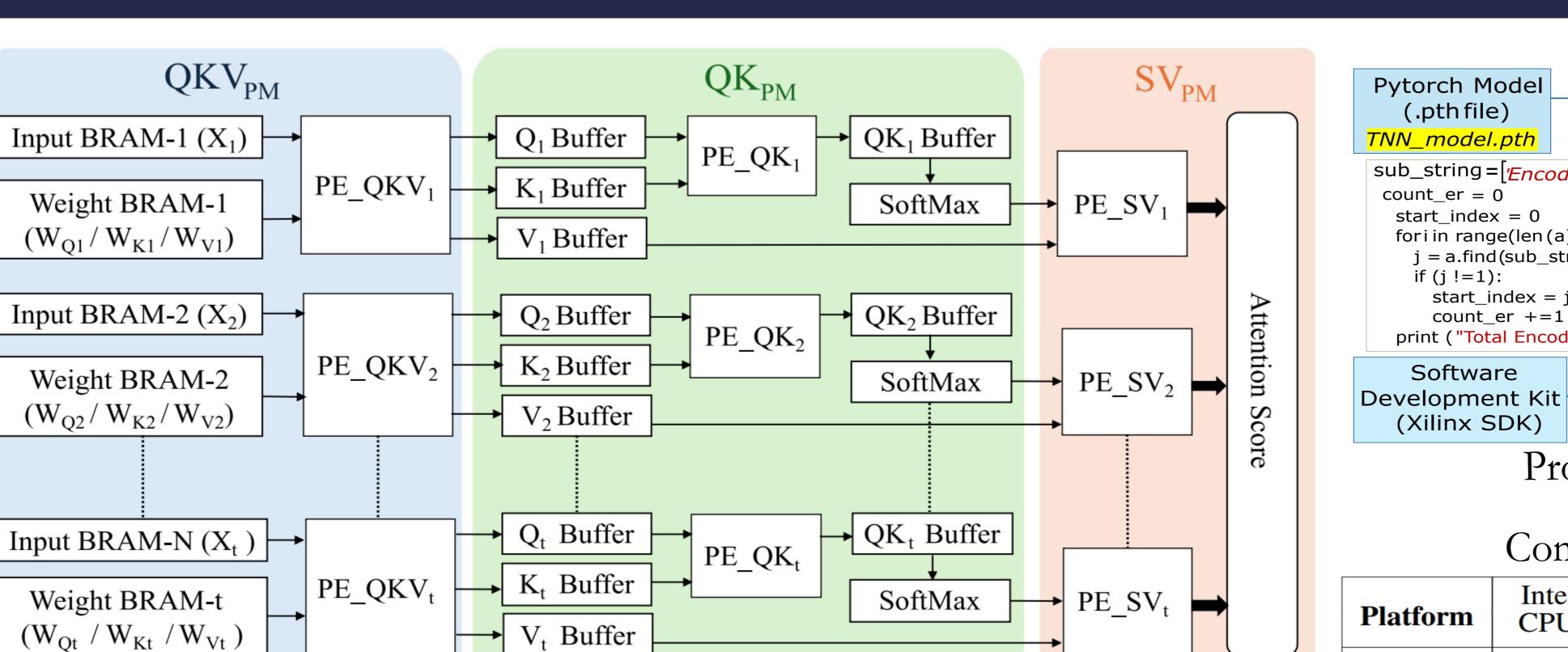


#### Accelerator Architecture

- Designed with C in Vitis HLS.
- Three main processing modules:QKVpM, QKpM and SVpM. QKVpM: Generates Q, K, V matrices. QK<sub>PM</sub>: Matrix-matrix multiplication operations between the Q and K matrices.
- SV<sub>PM</sub>: Matrix-matrix multiplication operations with V and the output from QK<sub>PM</sub>.

## Accelerating the Computation within the Attention Layer of the Transformer

- A novel architecture ensuring high BRAM and DSP utilization for efficient parallel processing with low latency.
- An efficient tiling of weight matrices to accommodate large models in on-chip memory.
- 3.28× and 2.6× faster than the Intel Xeon Gold 5220R CPU and NVIDIA V100 GPU respectively.
- 1.3× faster than the fastest state-of-the-art FPGA-based accelerator.



Accelerator Architecture for Attention Mechanism

#### Comparison with Other FPGA Accelerators

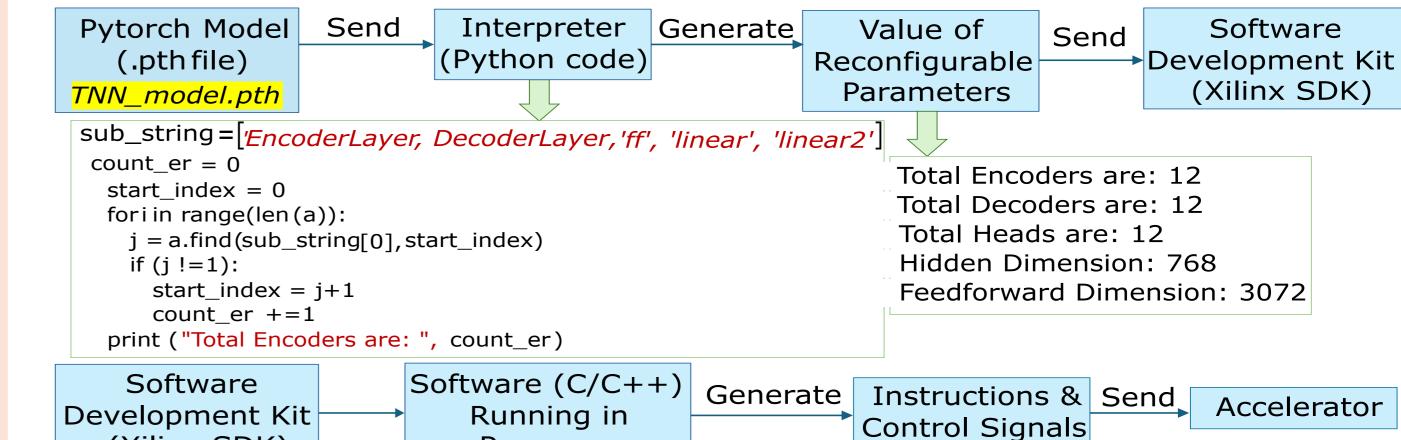
Works	Calabash [6]	Lu et al. [9]	Ye et al. [8]	Li et al. [7]	Peng et al. [4]	<b>FAMOUS</b>
<b>FPGAs</b>	Xilinx VU9P	Xilinx VU13P	Alveo U250	Xilinx VU37P	Alveo U200	Alveo U55C
Method	HDL	HDL	HDL	HLS	HLS	HLS
<b>DSPs</b>	4227	129	4189	1260	623	4157
BRAMs	640	498	1781	448	_	3148
GOPS	1288	128	171	72	97	623
Latency (ms)	0.239 <sup>a</sup>	0.8536 <sup>b</sup>	0.642	1.5264	1.706 <sup>c</sup>	0.494

<sup>&</sup>lt;sup>a</sup> Q, K, V matrix computation time ignored.

Load inputs into the	Load Values of Each Tile into Weight BRAM								
Input BRAM	Tile 1	Tile 2	Tile 3	Tile 4	Tile 5	Tile 6			
$\begin{bmatrix} X_{00} & X_{01} & X_{02} \\ X_{10} & X_{11} & X_{12} \end{bmatrix} \times$	$\begin{bmatrix} W_{00} \\ W_{10} \end{bmatrix}$	$W_{01} W_{11}$	$W_{02} \\ W_{12}$	$W_{03} W_{13}$	$W_{04} \\ W_{14}$	$\begin{bmatrix} W_{05} \\ W_{15} \end{bmatrix}$			
	$[W_{20}]$	$W_{21}$	$W_{22}$	$W_{23}$	$W_{24}$	$W_{25}$ ]			
1 <sup>St</sup> Iteration: Computation with 1 <sup>St</sup> Tile Computation output of 1 <sup>St</sup> Tile									
$\begin{bmatrix} X_{00} & X_{01} & X_{02} \\ X_{10} & X_{11} & X_{12} \end{bmatrix} \times \begin{bmatrix} W_{01} & W_{02} & W_{03} \\ W_{03} & W_{03} & W_{03} \end{bmatrix}$	$\begin{bmatrix} V_{00} \\ V_{10} \\ V_{20} \end{bmatrix} =$		$X_{00}W_{00}$	$X_{0} + X_{01}$ $X_{0} + X_{11}$	$W_{10} + M_{10} + M$	$X_{02}W_{20} \\ X_{12}W_{20}$			
2 <sup>nd</sup> Iteration: Computation with 2 <sup>nd</sup> Tile									
Ti	le 2		Compu	tation o	atput of	<sup>2nd</sup> Tile			
$\begin{bmatrix} X_{00} & X_{01} & X_{02} \\ X_{10} & X_{11} & X_{12} \end{bmatrix} \times \begin{bmatrix} V_{01} & V_{02} & V_{03} \\ V_{01} & V_{02} & V_{03} \end{bmatrix}$	$\begin{bmatrix} V_{01} \\ V_{11} \\ V_{21} \end{bmatrix} =$		$(X_{00}W_{01}X_{10}W_{01})$	$X_{1} + X_{01}$ $X_{1} + X_{11}$	$W_{11} + M_{11} + M$	$X_{02}W_{21} \ X_{12}W_{21}$			
n <sup>th</sup> Iteration: Computation with n <sup>th</sup> Tile Computation output of n <sup>th</sup> Tile									
$\begin{bmatrix} X_{00} & X_{01} & X_{02} \\ X_{10} & X_{11} & X_{12} \end{bmatrix} \times \begin{bmatrix} X_{01} & X_{02} \\ X_{10} & X_{11} & X_{12} \end{bmatrix}$	$\begin{bmatrix} W_{01} \\ W_{11} \\ W_{21} \end{bmatrix} =$		$\begin{bmatrix} X_{00}W_0 \\ X_{10}W_0 \end{bmatrix}$	$X_{0n} + X_{0n} + X_{1n}$	$_{1}^{1}W_{1n} + _{1}^{1}W_{1n} +$	$\begin{bmatrix} X_{02}W_{2n} \\ X_{12}W_{2n} \end{bmatrix}$			
Final Matrix = Output for $1^{st}$ tile + Output for $2^{nd}$ tile + + Output for $n^{th}$ tile									

Final Matrix = Output for 1<sup>st</sup> tile + Output for 2<sup>nd</sup> tile + .....+ Output for n<sup>th</sup> til

Tiling Technique



Process for Incorporating Programmability

Processor

Comparison with Other Acceleration Platforms

	1					
Platform	Intel E5 CPU [6]	NVIDIA V100 GPU [7]	Intel Xeon CPU [8]	NVIDIA P100 GPU [8]	FAM (Alveo U5	OUS 5C FPGA)
Topologies	64, 768, 12	64, 512, 4	64, 512, 8	64, 512, 4	64, 768, 8	64, 512, 8
GOP	0.308	0.11	0.11	0.11	0.308	0.11
Latency (ms)	1.1	1.5578	1.96	0.496	0.94	0.597
GOPS	280	71	56	221	328	184

Overall Result for MHA Accelerator

				) <b>( )</b>				110001010	1001			
est no.	Sequence Length	<b>Embedding Dimension</b>	Number of Heads	Tile Size	FPGA	Data Format	DSPs	BRAMs 18k	LUTs	FFs	Latency (ms)	GOPS
#1			8		Alveo	Obje Grad	4157 (46%)	3148 (78%)	1284782 (98%)	661996 (25%)	0.94	328
#2	64	768	4	64							1.401	220
#3			2		0330						2.281	135
		510			4.1						0.505	101
#4		512			Alveo	Shit fixed	4157 (46%)	3148 (78%)	1284782 (98%)	661996 (25%)	0.597	184
#5	64	256	8	64	U55C						0.352	312
11.6	120										2	214
#6	128	128 32 768	768 8	64	Alveo	Obit Grad	4157 (46%)	3148 (78%)	1284782 (98%)	661996 (25%)	2	314
#7	32				l						0.534	285
#8	16				0330						13	16

<sup>&</sup>lt;sup>b</sup> Time adjusted for 8 attention heads.

<sup>&</sup>lt;sup>c</sup> Time extracted for attention mechanism from a full transformer.