Texcelerate

On-Device AI Assistant for Text and Code Completion

Andrew Liao, Amelia Heller, Anirudh Prakash

Background

Sending data to the cloud can be risky!

- Developers at companies working on confidential projects risk leaking sensitive code
- Journalists or artists using AI tools for drafting may expose unpublished work to external servers
- Many people work in industries (healthcare, legal, etc) that have strict data regulations that restrict cloud processing

Requirements

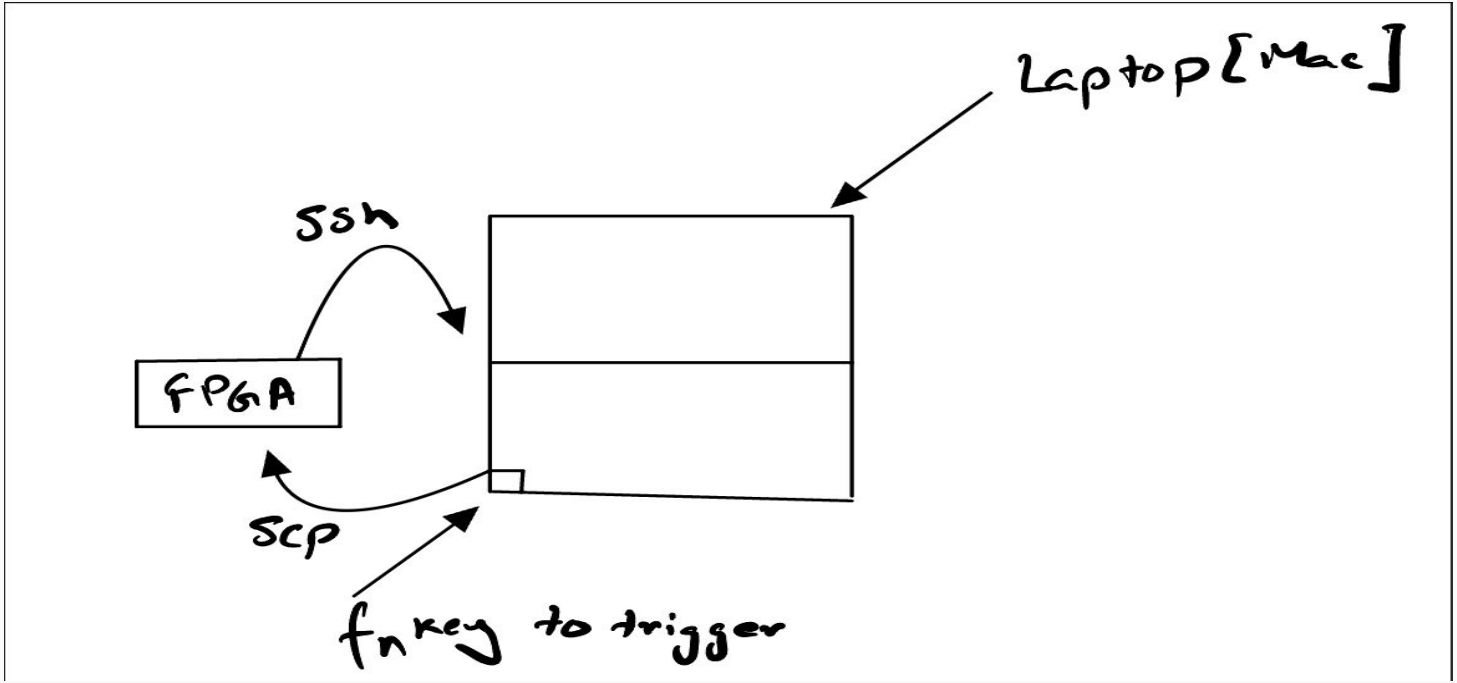
1. **Reading speed level output (10 tokens/sec, time to first token < 250ms)**
2. **Streaming should be reversible for natural user interface.**
3. **Provide context as needed to the system so as to not degrade output quality.**
4. **Power consumption and token generation speed lower than CPU and GPU**

Our Solution

Smart Assistant for Text and Code Suggestions: A plug-in FPGA accelerator to enable faster, more energy efficient, automatic text and code completion for users ranging from creative professionals to software engineers.

Efficient, Fast and Privacy Preserving: Uses a BitNet-based architecture optimized for hardware acceleration, maintaining on-device privacy tasks like code suggestion and creative writing.

Solution Approach



Solution Approach – Motivation

1) Bitnets are used to ensure low memory bandwidth and processing requirements.
2) A wirelessly connected FPGA is used for acceleration, this potentially could be used to provide acceleration to multiple clients. This significantly improves utilization of the core.
3) LUT based acceleration was proven to provide over 50% acceleration on GPUs by Microsoft

Technical Challenges

Architectural Improvements:

We need to ensure the system meets required reading speed (10 tokens/sec)

Lookup table-based acceleration may encounter limitations due to the FPGA's LUT capacity.

Technical Challenges

Integration:

Ensure that we can provide the necessary context as required—either by developing our own text editor/plugin or by utilizing a reconfigurable one.

We will need to work with HLS tools and synthesis techniques for Xilinx based FPGAs – Vivado, Vitis, etc.

Risks

1. Limited FPGA iteration speed

2. Insufficient time for user interface development

3. Communication between user and FPGA is essential but out of project scope

Ethical Concerns

1) Biased outputs – we plan to evaluate on known benchmark like CrowS-Pairs.

2) Hallucinations – we plan to evaluate this on the TruthfulQA benchmark.

3) The FPGA system may glitch and affect the user's device – This we plan to mitigate with a limited power output.

Testing and Verification

Model Correctness: Ensure the model avoids hallucinations – CrowS and TruthfulQA

Performance Infrastructure: Ensure that our token/second and time to first token counters are accurate – Use large outputs and manual timing

Power and Performance Improvements: Verify that our architecture has surpassed power and timing benchmark metrics –over 10 tokens/sec and under 500mW of power

Texccelerate

Anirudh Prakash, Amelia Heller, Andrew Liao

Project start: **Wed, 1/29/2025**

Display week: **1**

TASK	ASSIGNED TO	PROGRESS	START	END
Project Planning				
Decide on FPGA		80%	1/29/25	2/5/25
Decide UI/UX Structure		0%	2/5/25	2/7/25
Decide Benchmark Softcores		50%	1/29/25	2/12/25
Choose Target Model		60%	1/29/25	2/3/25
ML Model				
Test existing BitNet Model		100%	1/29/25	2/1/25
Select Text model for quantize		50%	1/31/25	2/5/25
Quantize text model		0%	2/5/25	2/19/25
Modify inference code for CPU soft core deploymen		0%	2/20/25	2/27/25
Modify inference code for GPU soft core deploymen		0%	2/28/25	3/14/25
Modify inference code for FPGA deployment		0%	3/15/25	3/29/25
FPGA Acceleration				
Choose CPU softcore for reference		100%	1/29/25	2/5/25
Choose GPU softcore for reference		0%	2/6/25	2/20/25
Implement Unified Performance Counter		0%	2/21/25	2/28/25
Synthesize CPU/GPU soft cores		0%	2/5/25	2/28/25
Decide FPGA Architecture + RTL		0%	2/25/25	3/11/25
Verify Texccelerate RTL		0%	3/12/25	3/19/25
Synthesize Texccelerate on FPGA		0%	3/20/25	4/3/25
Synthesize Texccelerate on FPGA		0%	4/4/25	4/18/25
FPGA Interface, UI/UX				
Boot Linux on FPGA hard core		0%	2/19/25	2/22/25
FPGA to computer UART framework		0%	2/22/25	3/4/25
FPGA PS to PL communication framework		0%	3/5/25	3/15/25
Stream PMU metris through UART		0%	3/16/25	3/23/25
UI/UX Software Interface		0%	3/24/25	4/7/25
Whole System Integrated testing		0%	4/8/25	4/22/25