A 25mm^2 SoC for IoT Devices with 18ms Noise-Robust Speech-to-Text Latency via Bayesian Speech Denoising and Attention-Based Sequence-to-Sequence DNN Speech Recognition in 16nm FinFET

Thierry Tambe[1]

En-Yu Yang[1], Glenn G. Ko[1], Yuji Chai[1], Coleman Hooper[1], Marco Donato[2]

Paul N. Whatmough[1,3], Alexander M. Rush[4], David Brooks[1], and Gu-Yeon Wei[1]

[1]Harvard University, Cambridge, MA, [2]Tufts University, Medford, MA,

[3]ARM, Boston, MA, [4]Cornell University, New York, NY

Self Introduction

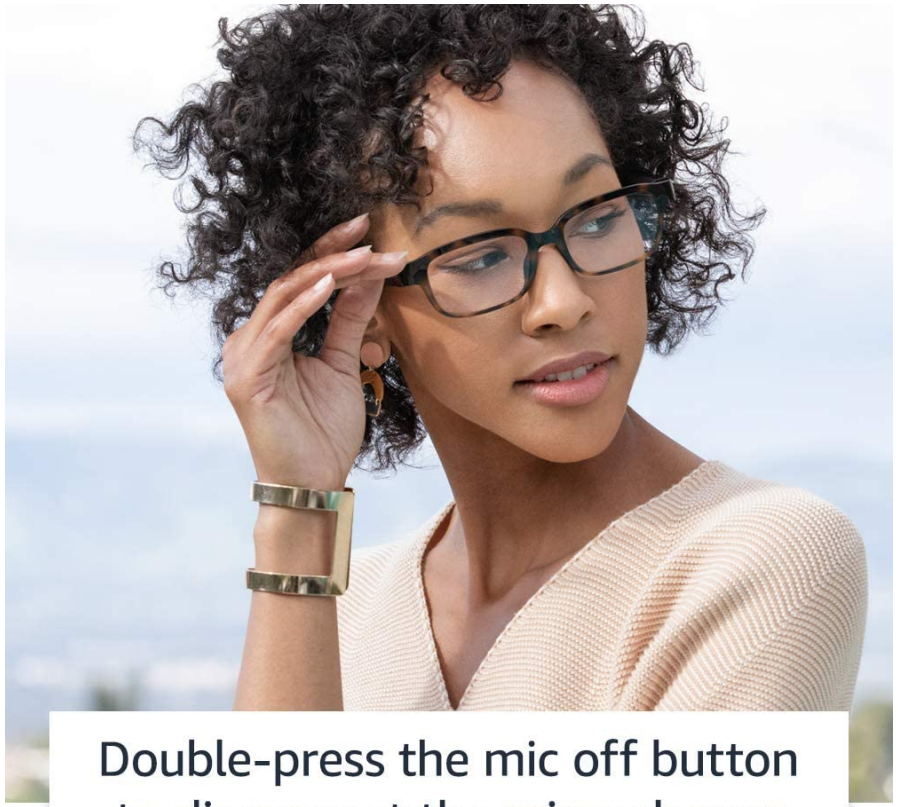
- Thierry Tambe is an EE PhD student at Harvard University.
- Current research interests focus on designing algorithms, energy-efficient and high-performance hardware accelerators and systems for machine learning and natural language processing in particular.
- Thierry was a staff engineer at Intel, Hillsboro, OR, USA (2012-2017) designing various analog/mixed-signal architectures for high-bandwidth memory and peripheral interfaces on Xeon and Xeon-Phi HPC SoCs.
- B.S. (2010) and M.Eng. (2012) both in Electrical Engineering from Texas A&M University.

Motivating Vision of the Future

Cloud and/or smart phone based *personal AI assistant* with speech-based conversational AI interfaces

Polyphonic adaptability *and* linguistic context understanding is of paramount importance

Amazon's echo frame



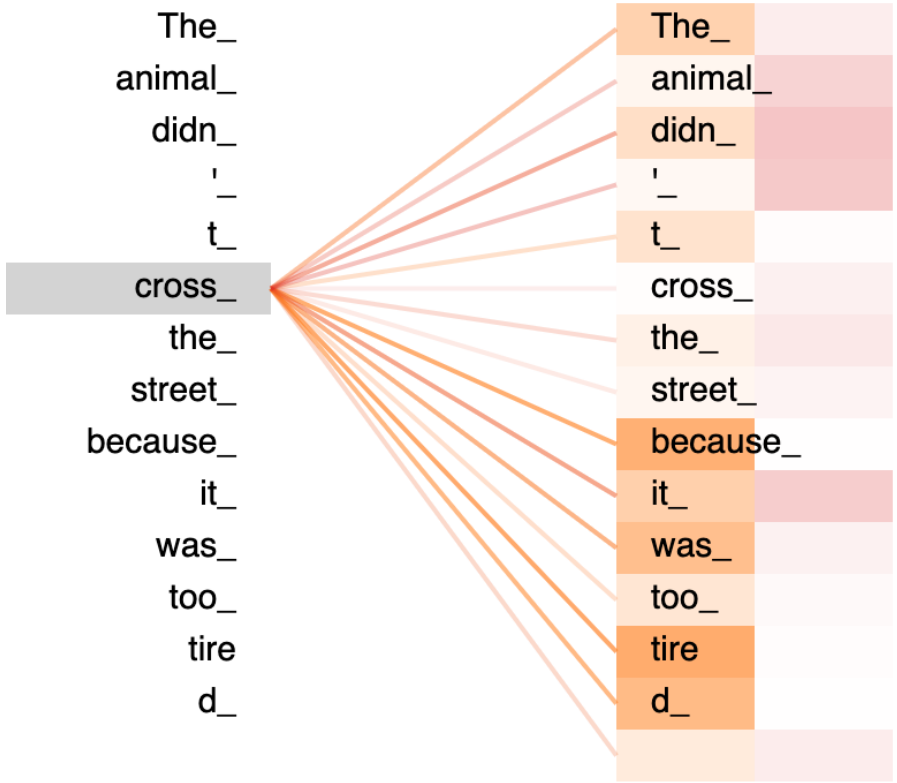
Always-ON KWS

On-demand (or perpetual) ASR

Numerous NLP tasks, e.g., translation and text-to-speech

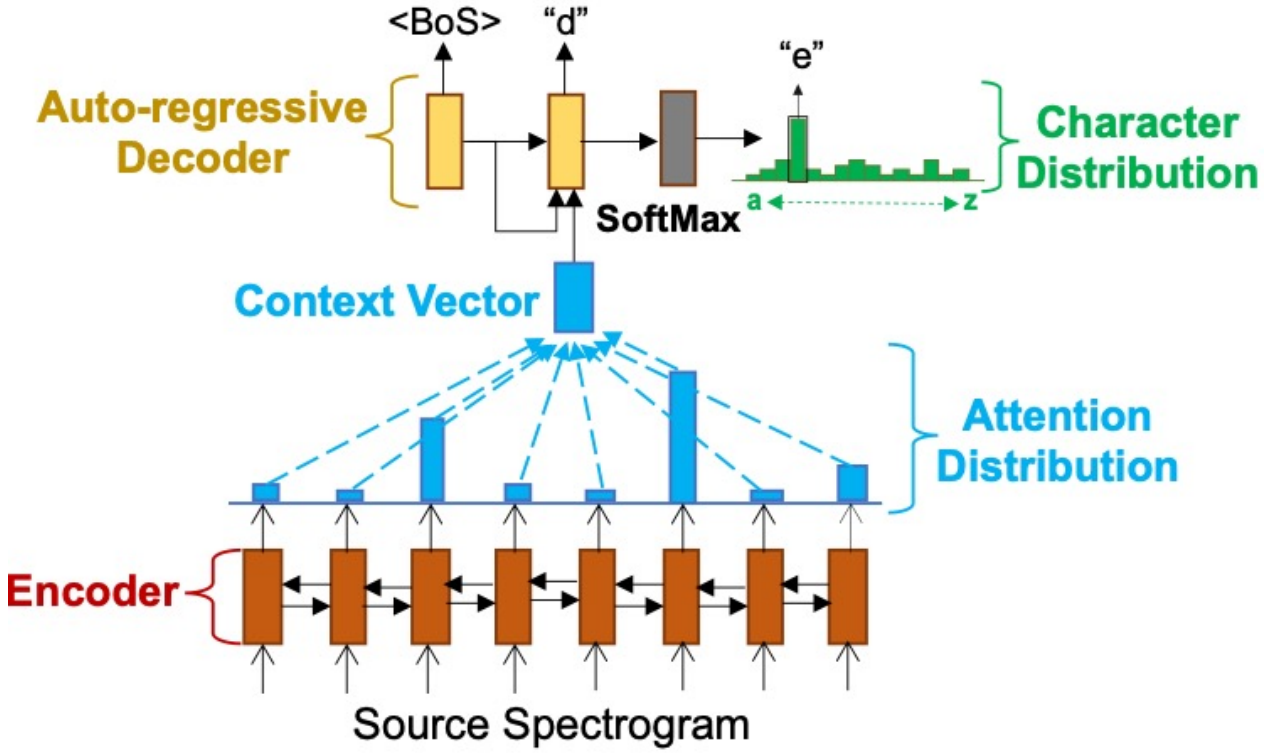
Three Main Messages

1) Accelerating the Attention Mechanism



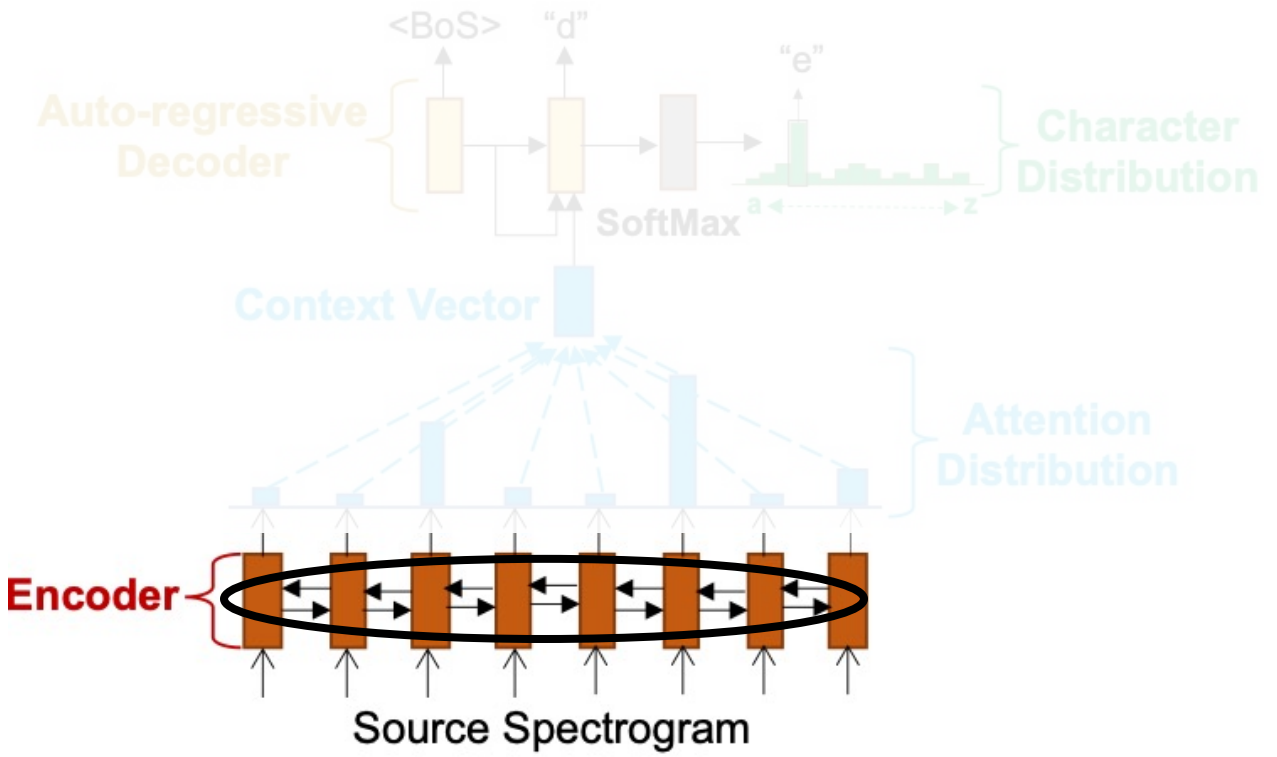
- **The attention mechanism is nowadays the most effective neural building block in NLP**

1) Accelerating the Attention Mechanism



- **Accelerating attention poses challenges and opportunities unique from CNNs and RNNs**

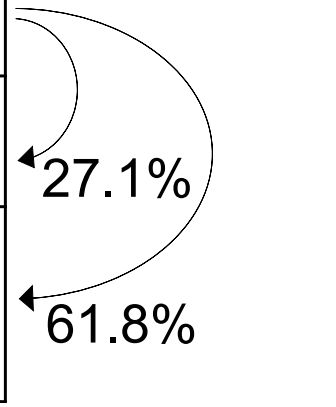
Further context learning with Bidir. RNNs



- **Bidirectional RNNs capture context by concatenating forward and backward time steps**

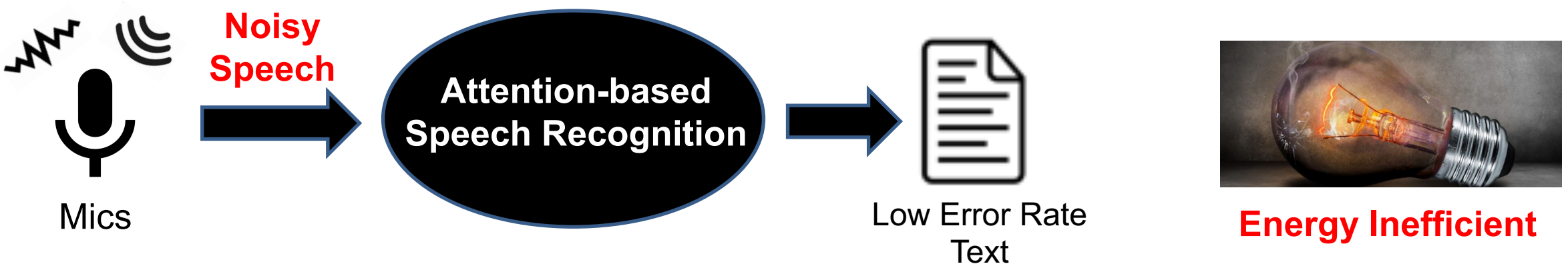
Attention + Bidir. RNN Benefits

	WER (LibriSpeech)
Unidir. LSTM w/o Attention #Params=3.9M **(Conventional)**	27.6
Unidir. LSTM w/ Attention #Params=3.3M	20.11
Bidir. LSTM w/ Attention #Params=3.5M **(this work)**	10.54

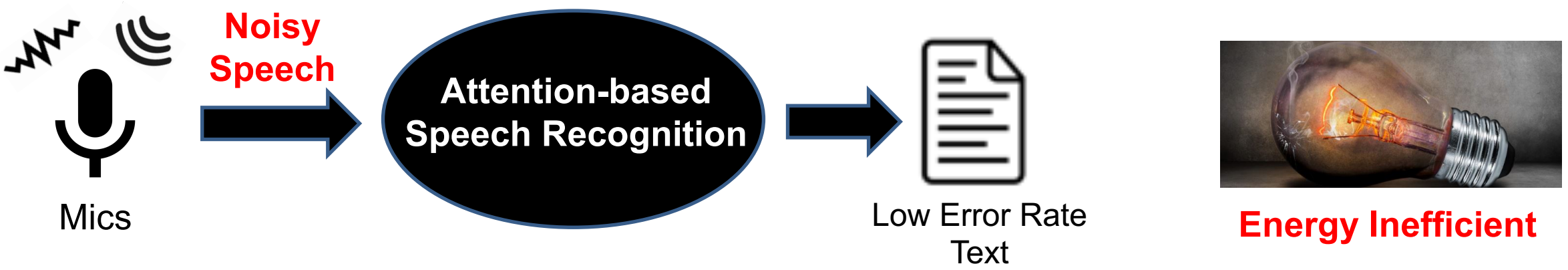


- **Attention-based bidirectional RNNs can improve ASR accuracy by up to 62% compared to a simpler unidirectional RNN**

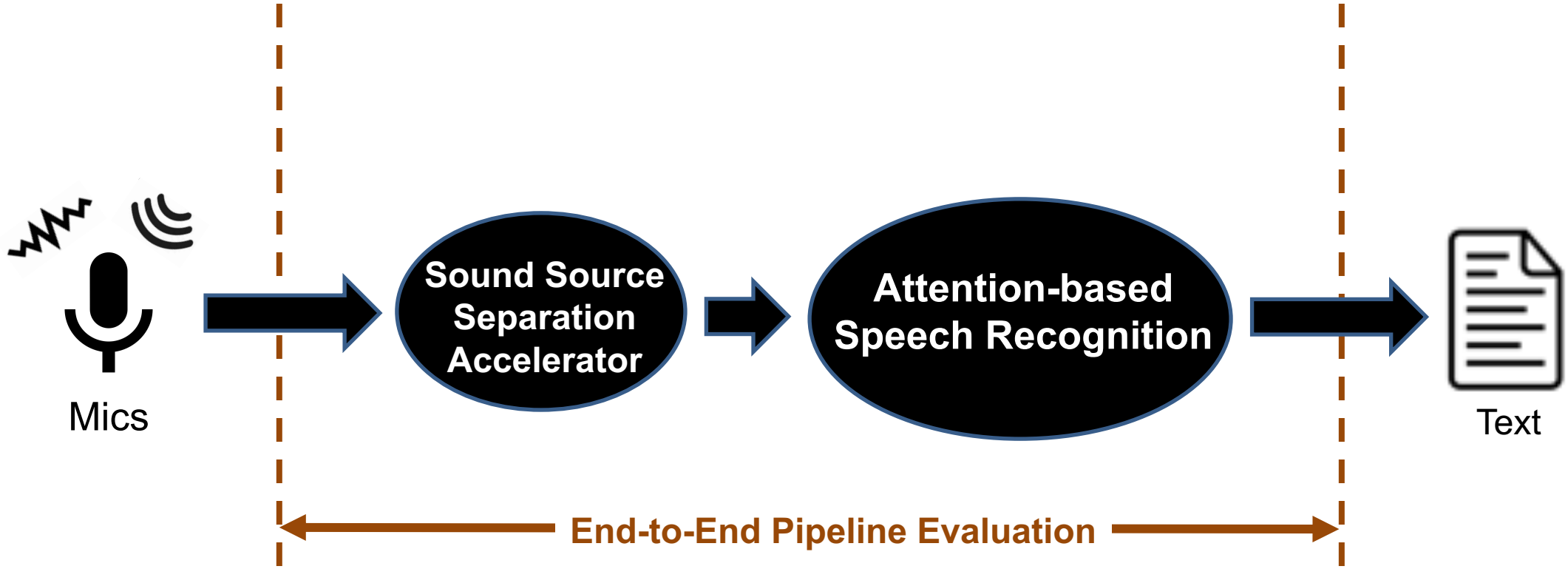
2) Benefit from Pre-Recognition Denoising



2) Benefit from Pre-Recognition Denoising



3) End-to-End Evaluation



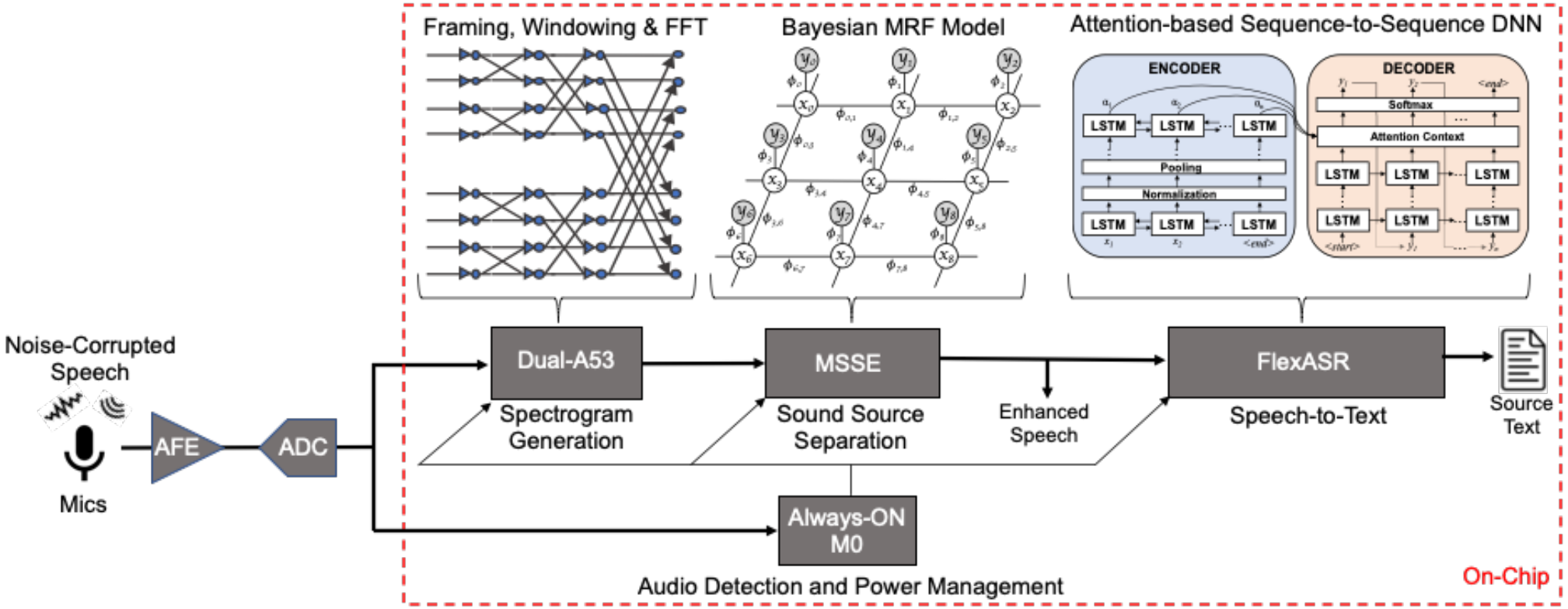
Outline

- Motivation
- **Speech-Enhancing ASR**
  - Functional Pipeline
  - 16nm SoC Architecture
  - Markov Source Separation Engine (MSSE)
  - Attention-based Seq2Seq Accelerator (FlexASR)
    - FlexASR Processing Element
    - FlexASR Multi-Function Global Buffer
- **Chip Measurement Results**
- **Summary**

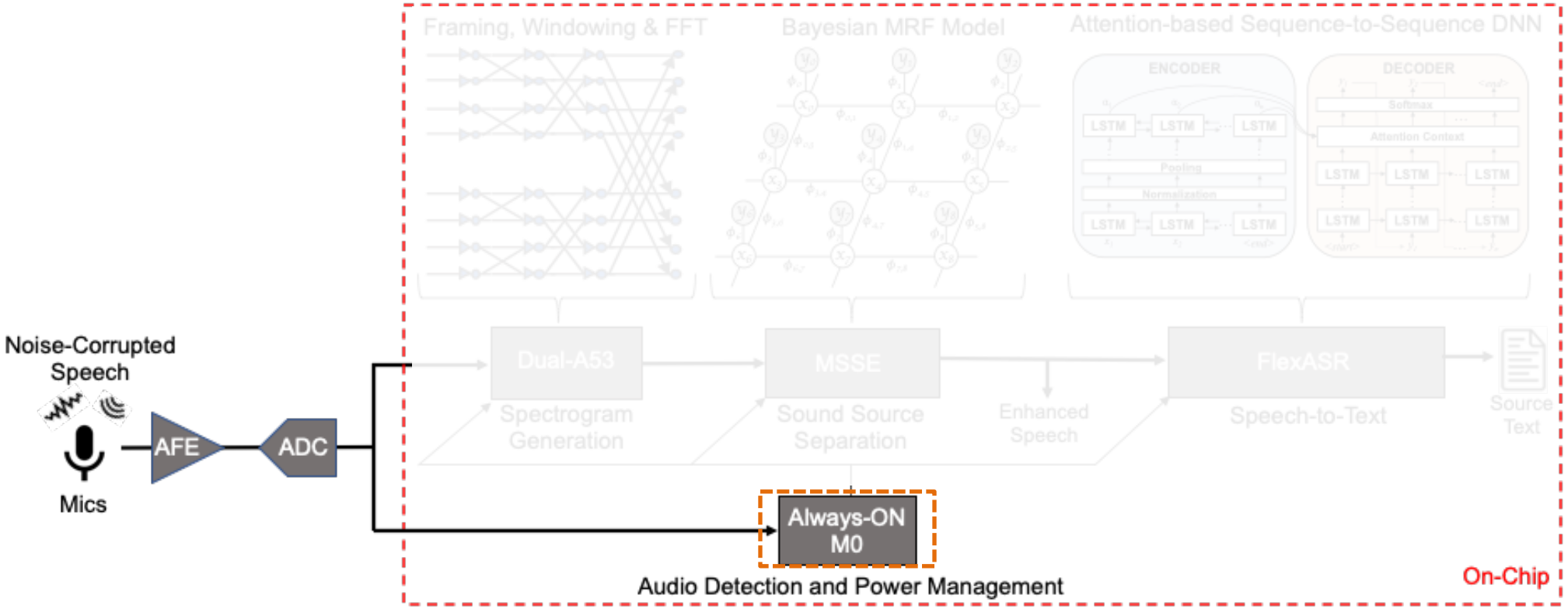
Outline

- Motivation
- **Speech-Enhancing ASR**
  - **Functional Pipeline**
  - 16nm SoC Architecture
  - Markov Source Separation Engine (MSSE)
  - Attention-based Seq2Seq Accelerator (FlexASR)
    - FlexASR Processing Element
    - FlexASR Multi-Function Global Buffer
- Chip Measurement Results
- Summary

Speech-Enhancing ASR Pipeline

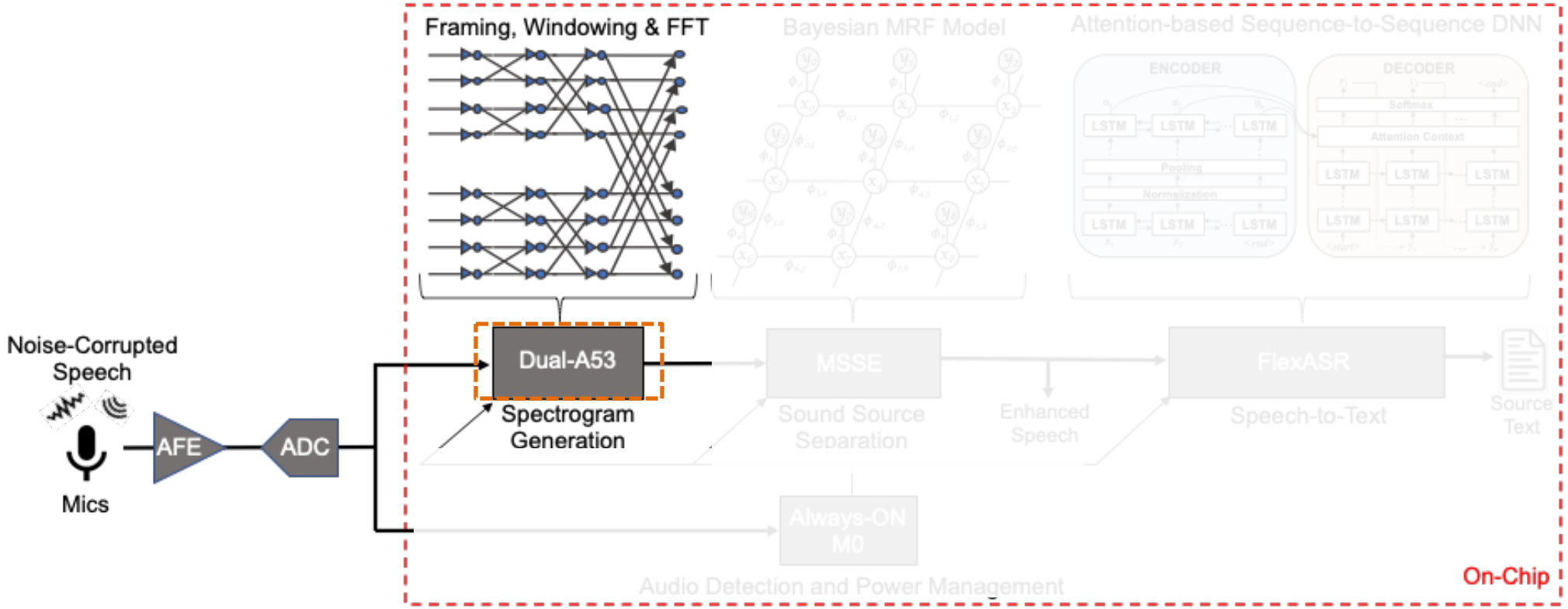


Speech-Enhancing ASR Pipeline



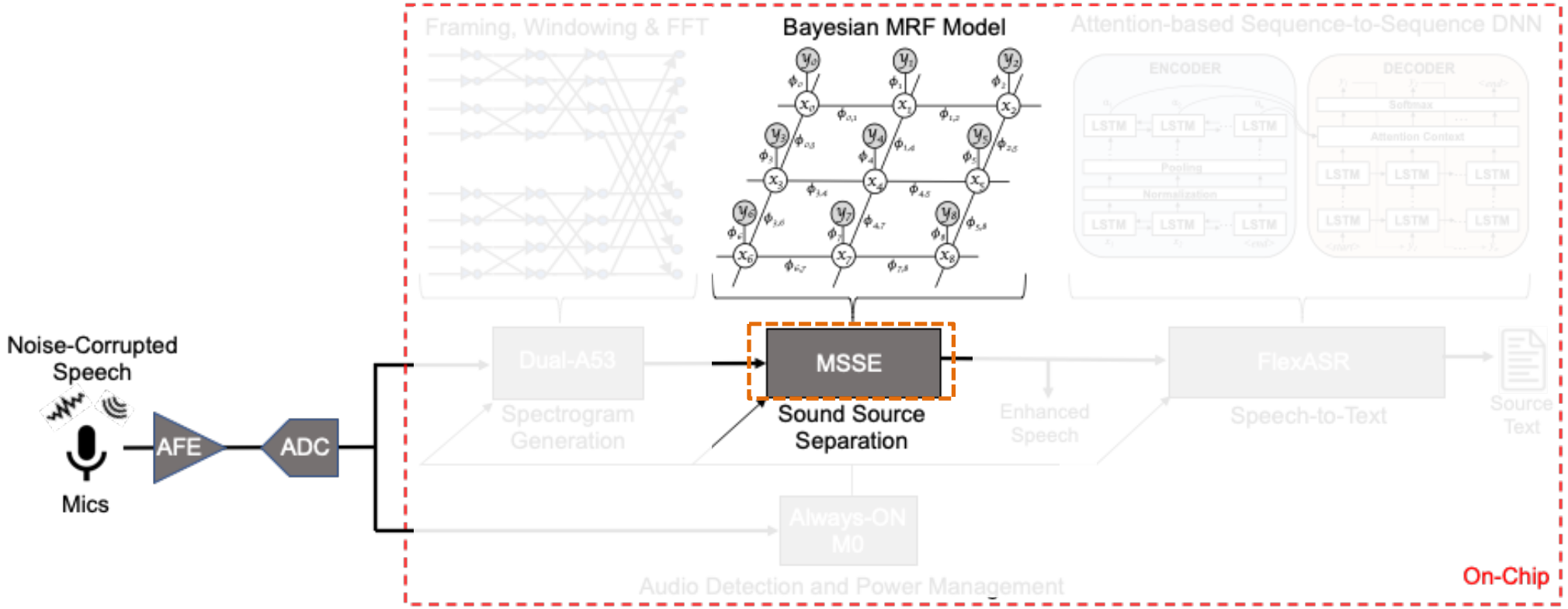
- **M0: monitors incoming audio amplitudes and subsequently boots accelerators**

Speech-Enhancing ASR Pipeline



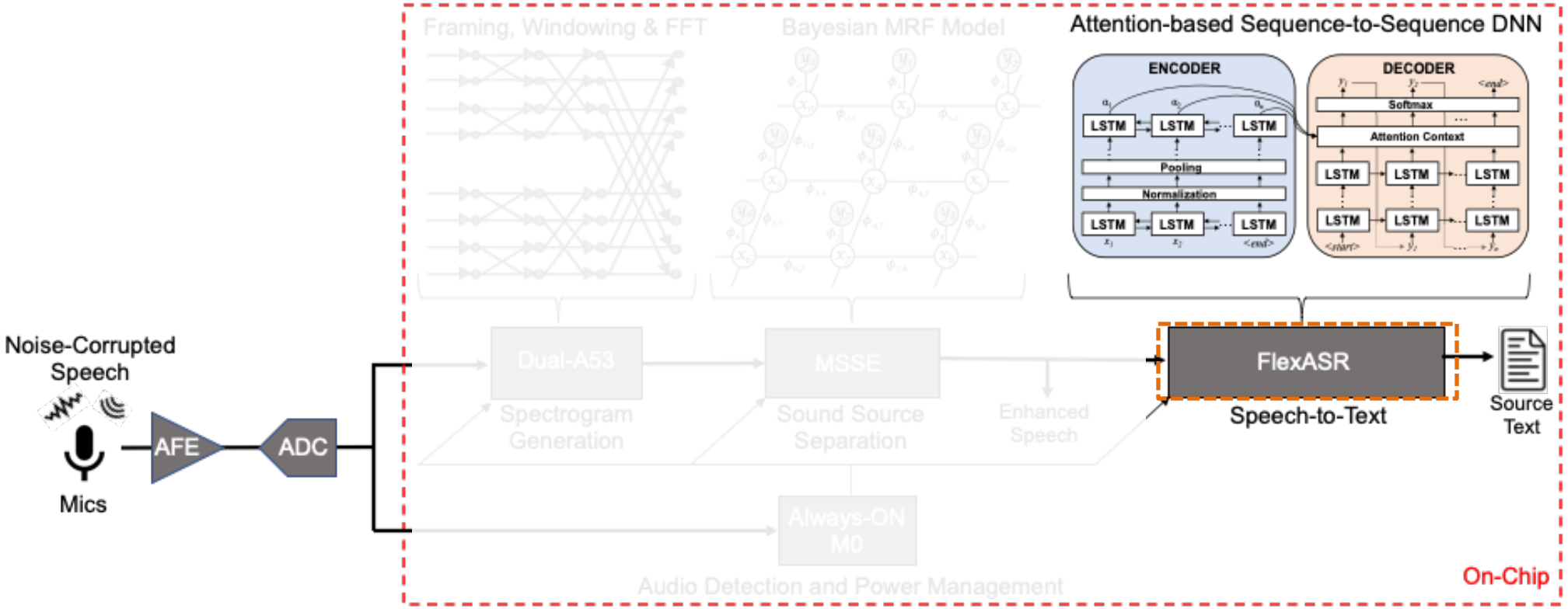
- **M0: monitors incoming audio amplitudes and subsequently boots accelerators**
- **Dual A53: performs feature extraction tasks (framing, windowing, 1024-pt FFT)**

Speech-Enhancing ASR Pipeline



- **M0: monitors incoming audio amplitudes and subsequently boots accelerators**
- **Dual A53: performs feature extraction tasks (framing, windowing, 1024-pt FFT)**
- **MSSE: optimized for unsupervised speech enhancement via Gibbs sampling**

Speech-Enhancing ASR Pipeline

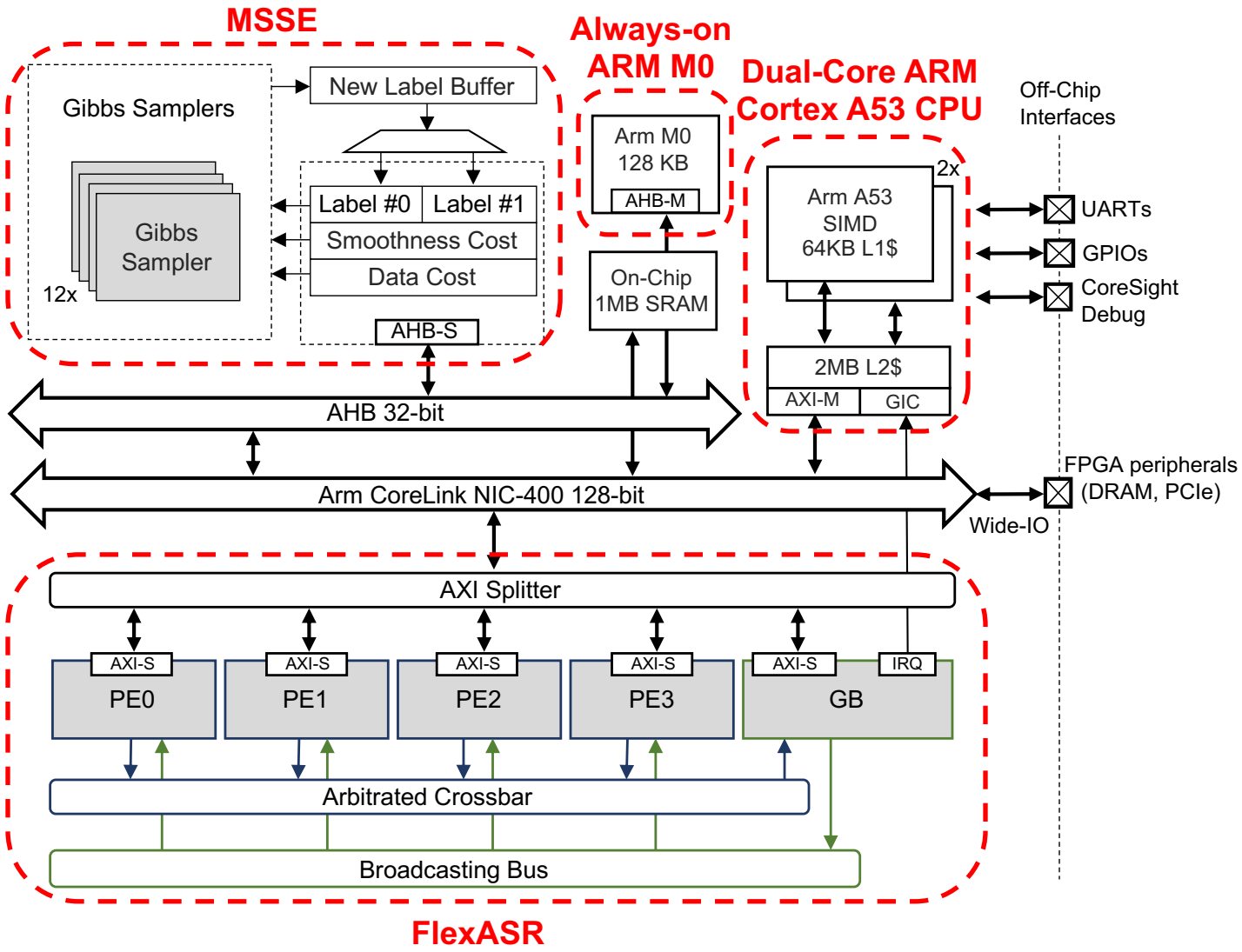


- **M0: monitors incoming audio amplitudes and subsequently boots accelerators**
- **Dual A53: performs feature extraction tasks (framing, windowing, 1024-pt FFT)**
- **MSSE: optimized for unsupervised speech enhancement via Gibbs sampling**
- **FlexASR: optimized for large vocabulary attention-based bidirectional RNNs**

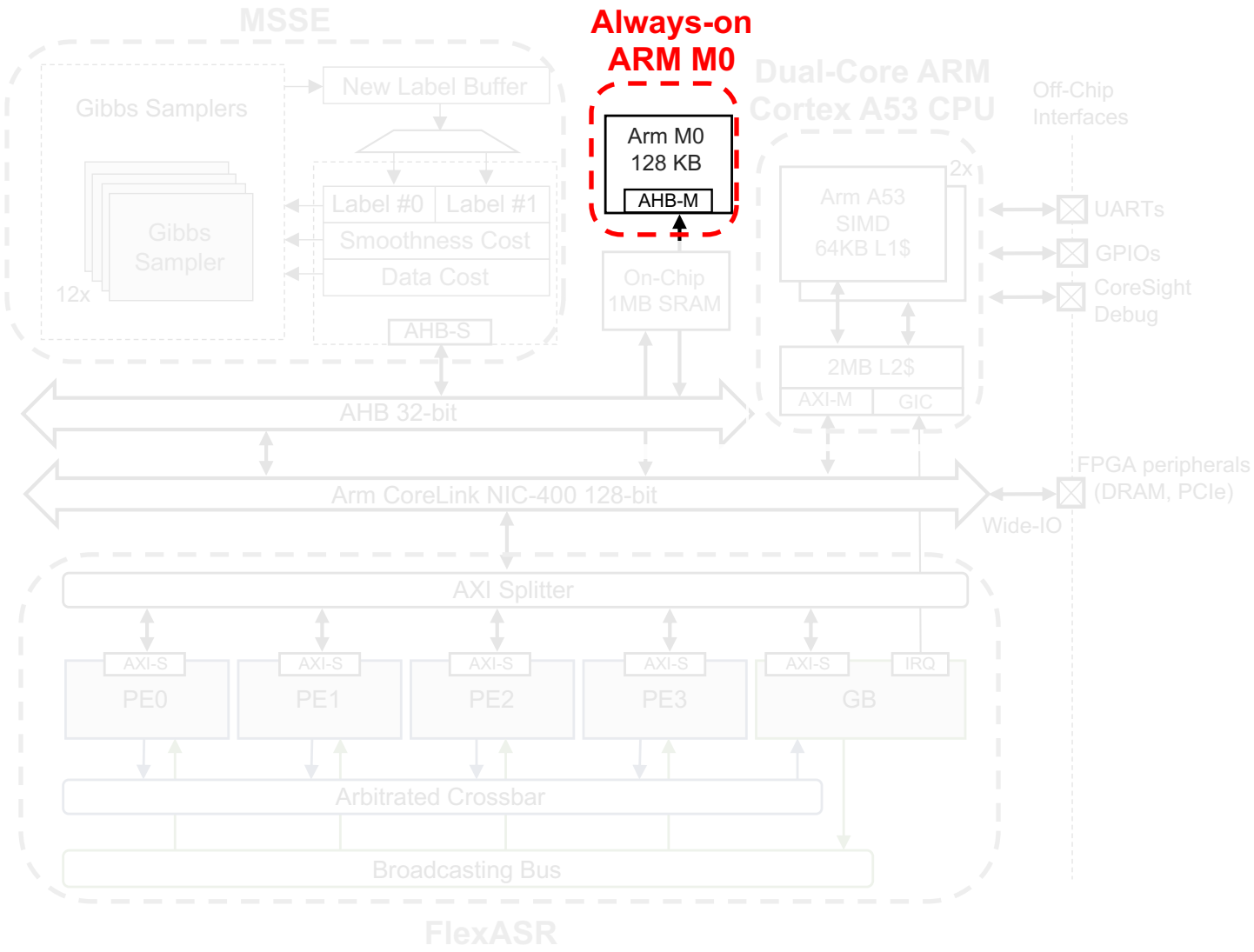
Outline

- Motivation
- **Speech-Enhancing ASR**
  - Functional Pipeline
  - **16nm SoC Architecture**
  - Markov Source Separation Engine (MSSE)
  - Attention-based Seq2Seq Accelerator (FlexASR)
    - FlexASR Processing Element
    - FlexASR Multi-Function Global Buffer
- Chip Measurement Results
- Summary

SoC Architecture

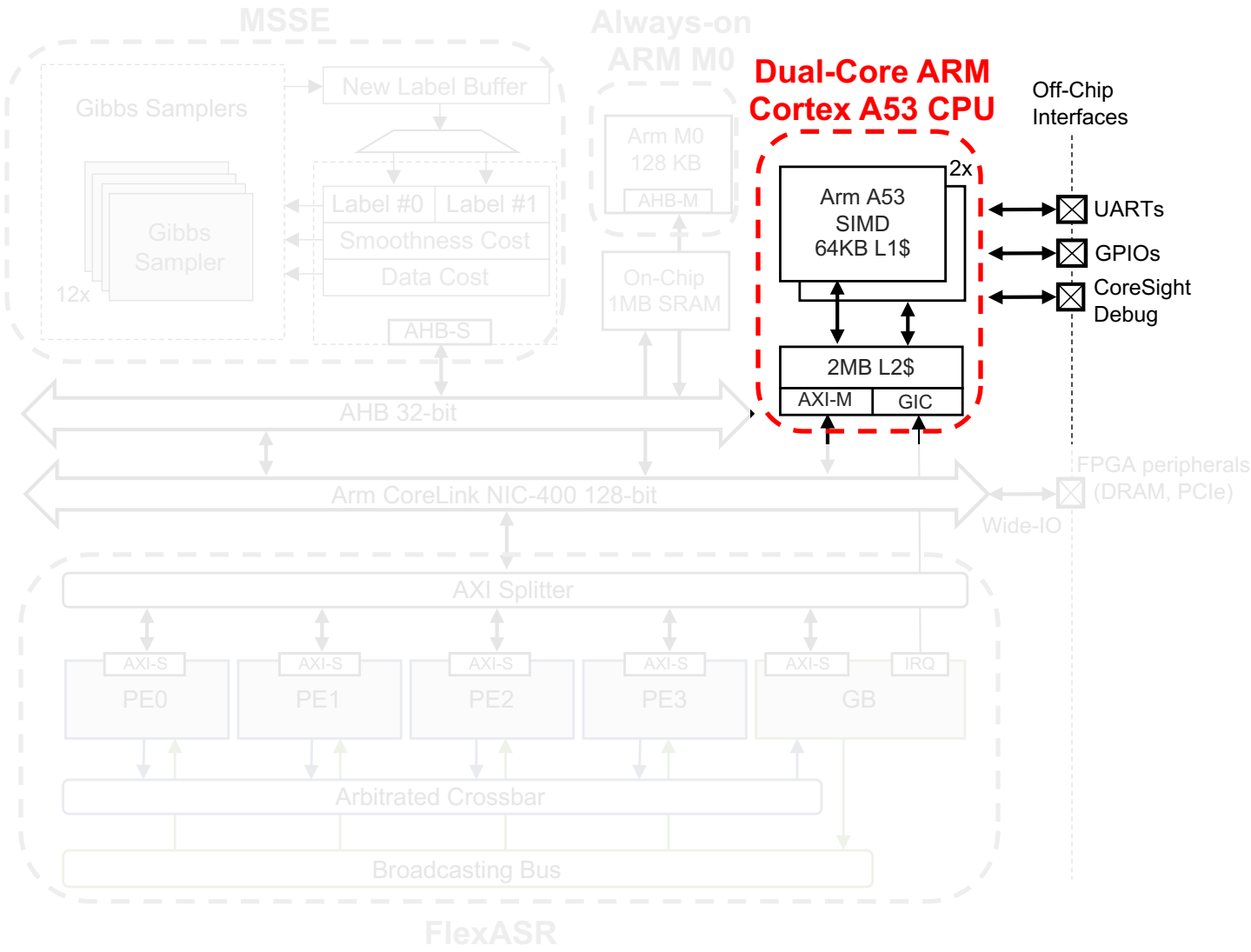


SoC Architecture



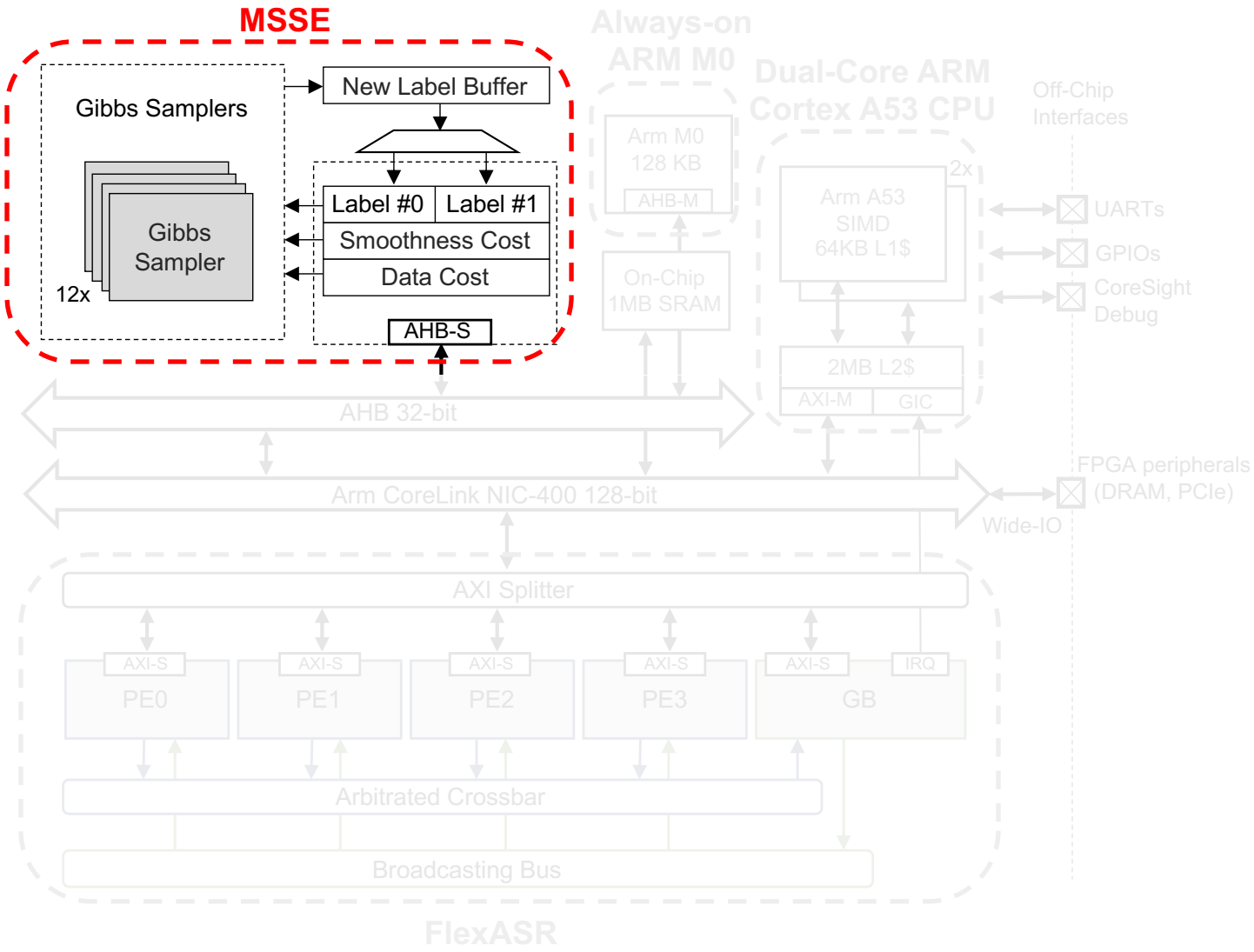
- **Always-on ARM M0**

SoC Architecture



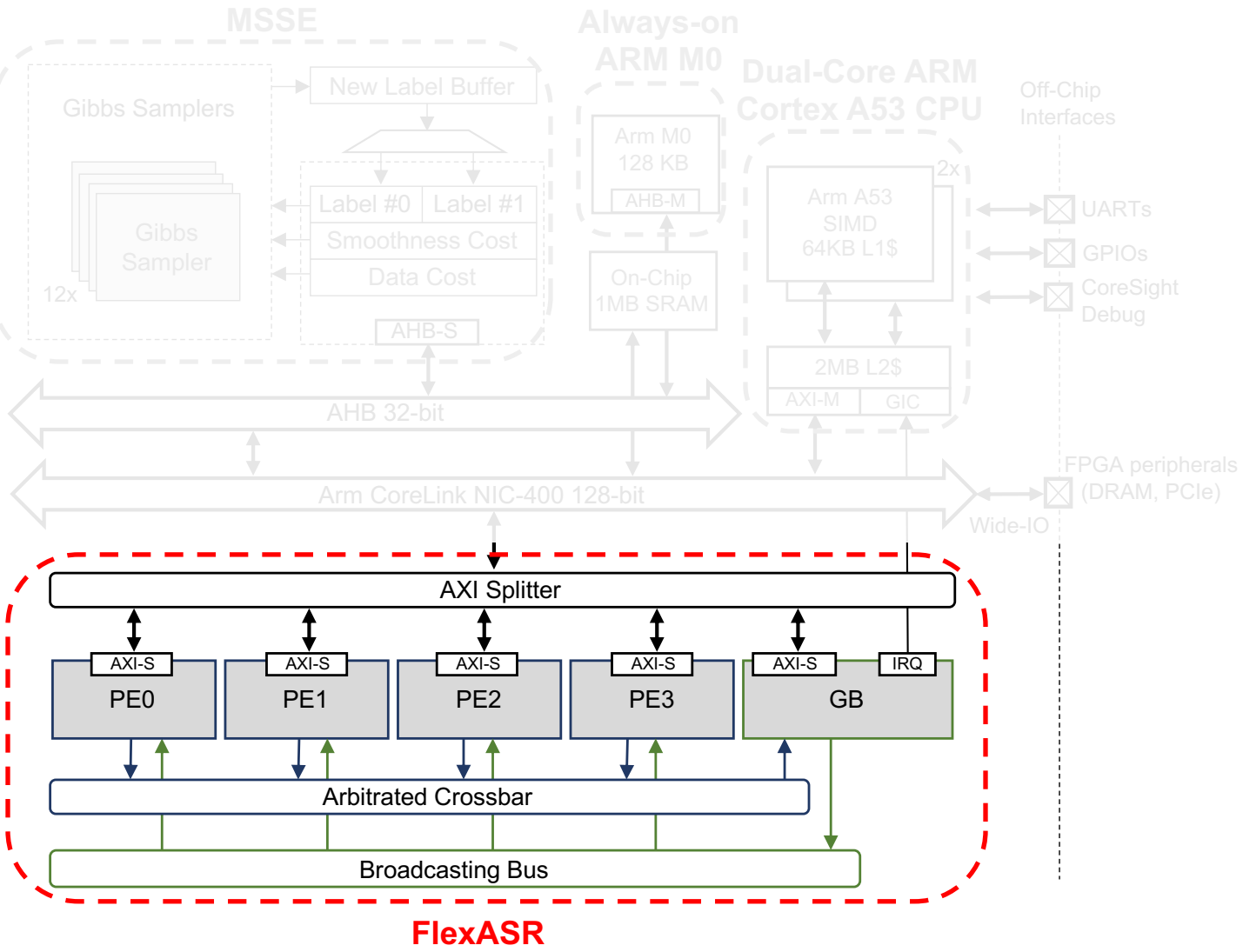
- **Always-on ARM M0**
- **Dual A53 with 2MB L2 cache**

SoC Architecture



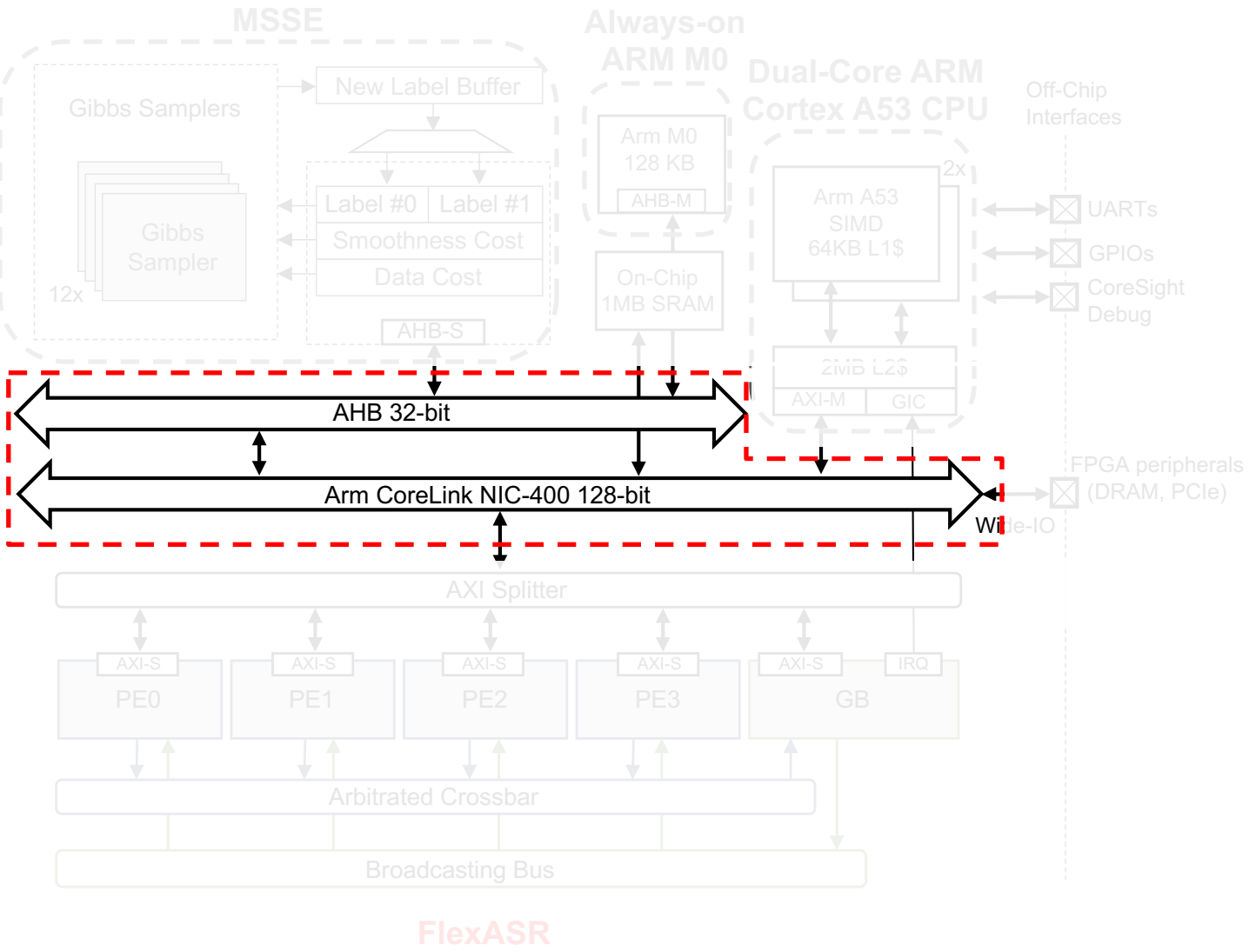
- **Always-on ARM M0**
- **Dual A53 with 2MB L2 cache**
- **MSSE: 12 parallel Gibbs samplers**

SoC Architecture



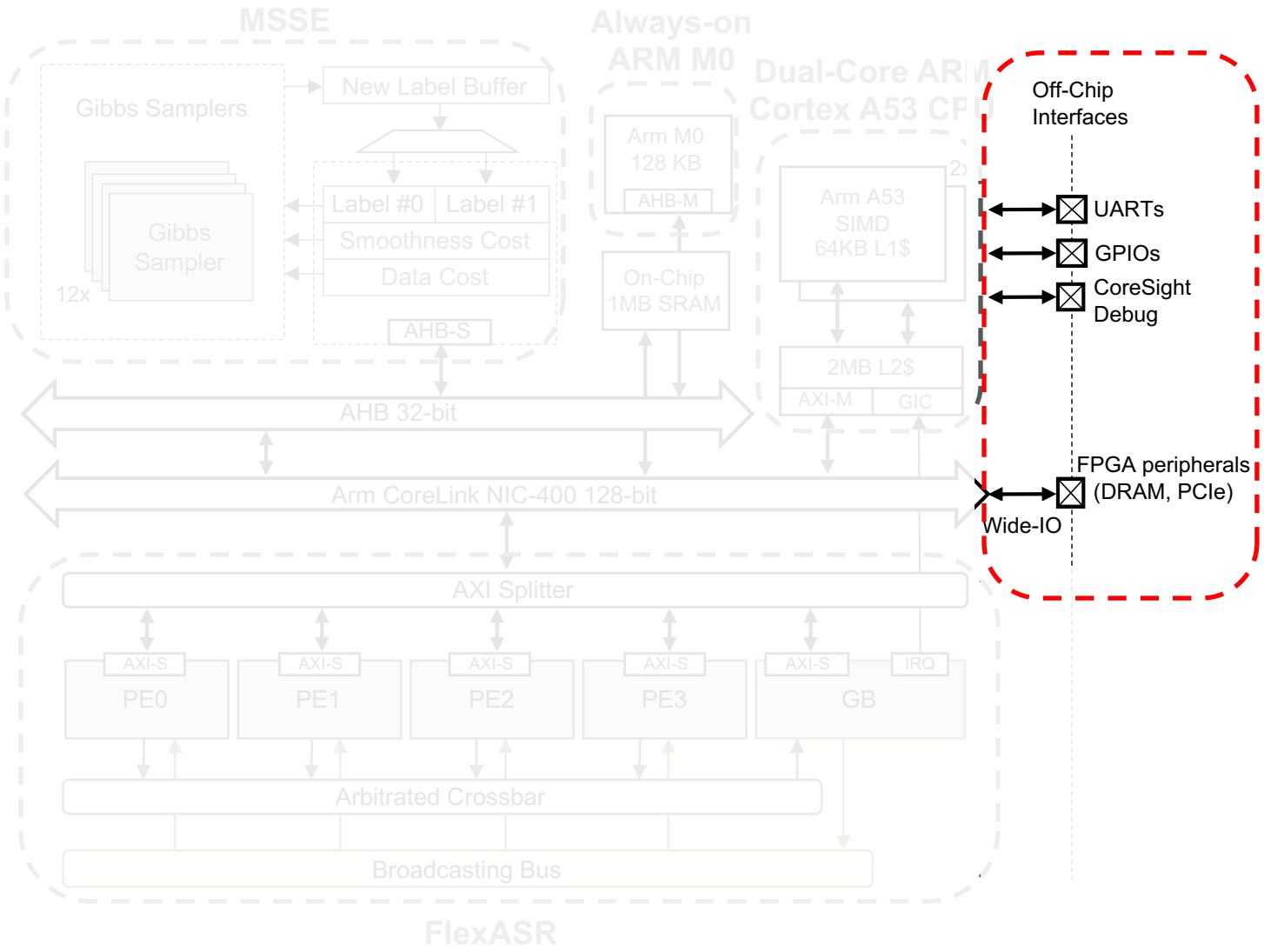
- **Always-on ARM M0**
- **Dual A53 with 2MB L2 cache**
- **MSSE: 12 parallel Gibbs samplers**
- **FlexASR: 4 processing elements and a multi-function global buffer**

SoC Architecture



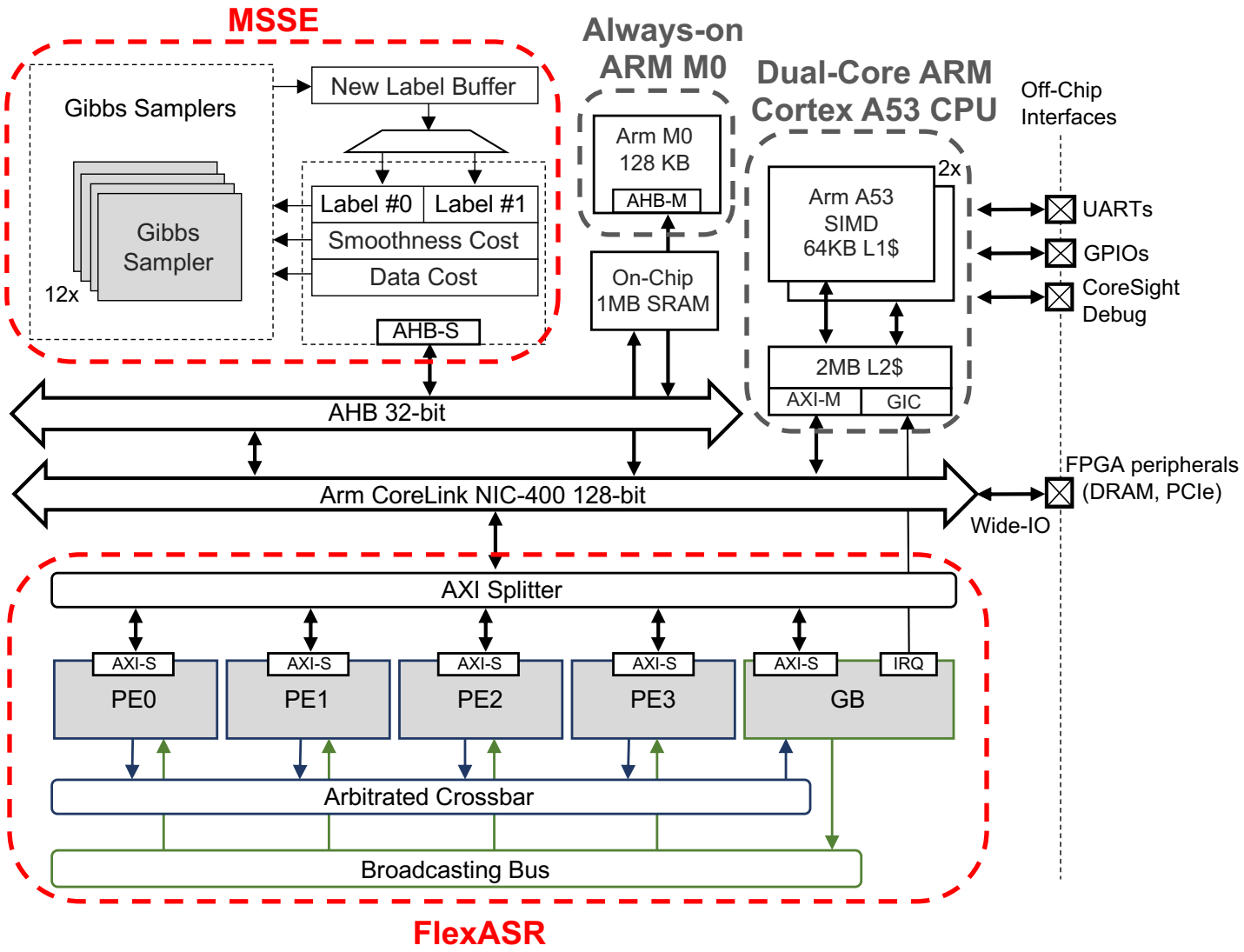
- **Always-on ARM M0**
- **Dual A53 with 2MB L2 cache**
- **MSSE: 12 parallel Gibbs samplers**
- **FlexASR: 4 processing elements and a multi-function global buffer**
- **128-bit AXI and 32-bit AHB NoCs**

SoC Architecture



- **Always-on ARM M0**
- **Dual A53 with 2MB L2 cache**
- **MSSE: 12 parallel Gibbs samplers**
- **FlexASR: 4 processing elements and a multi-function global buffer**
- **128-bit AXI and 32-bit AHB NoCs**
- **128-bit Wide-IO to FPGA and testing support**

SoC Architecture

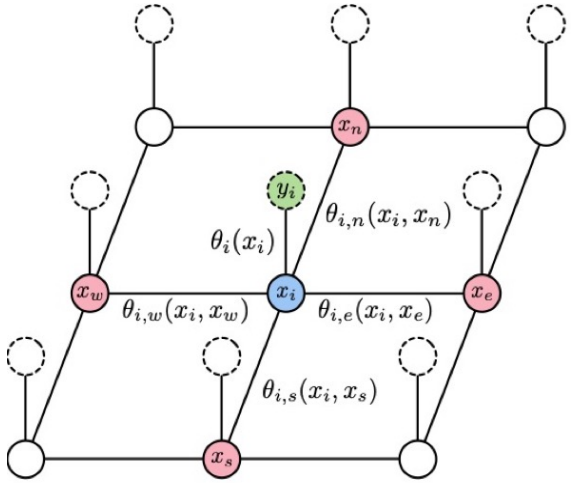


- **Always-on ARM M0**
- **Dual A53 with 2MB L2 cache**
- **MSSE: 12 parallel Gibbs samplers**
- **FlexASR: 4 processing elements and a multi-function global buffer**
- **128-bit AXI and 32-bit AHB NoCs**
- **128-bit Wide-IO to FPGA and testing support**

Outline

- Motivation
- **Speech-Enhancing ASR**
  - Functional Pipeline
  - 16nm SoC Architecture
  - **Markov Source Separation Engine (MSSE)**
  - Attention-based Seq2Seq Accelerator (FlexASR)
    - FlexASR Processing Element
    - FlexASR Multi-Function Global Buffer
- Chip Measurement Results
- Summary

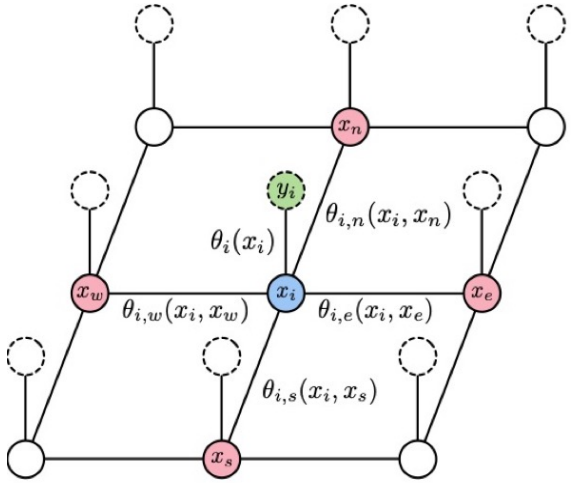
Gibbs Sampling Inference



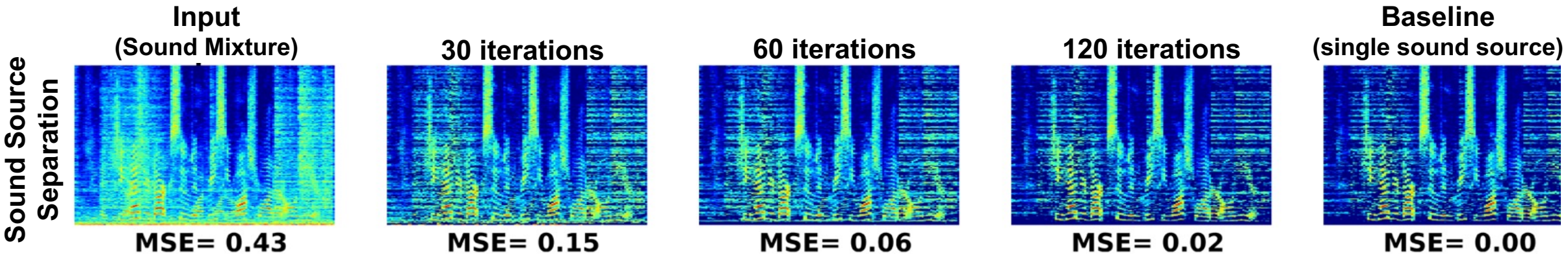
- Nodes representing input speech features
- Nodes representing output labels corresponding to feature locations
- Node being sampled
- Observed node
- Neighbor labels

[2]: G. Ko et al., A 3mm2 Programmable Bayesian Inference Accelerator for Unsupervised Machine Perception using Parallel Gibbs Sampling in 16nm, VLSI Symposium, 2020

Gibbs Sampling Inference

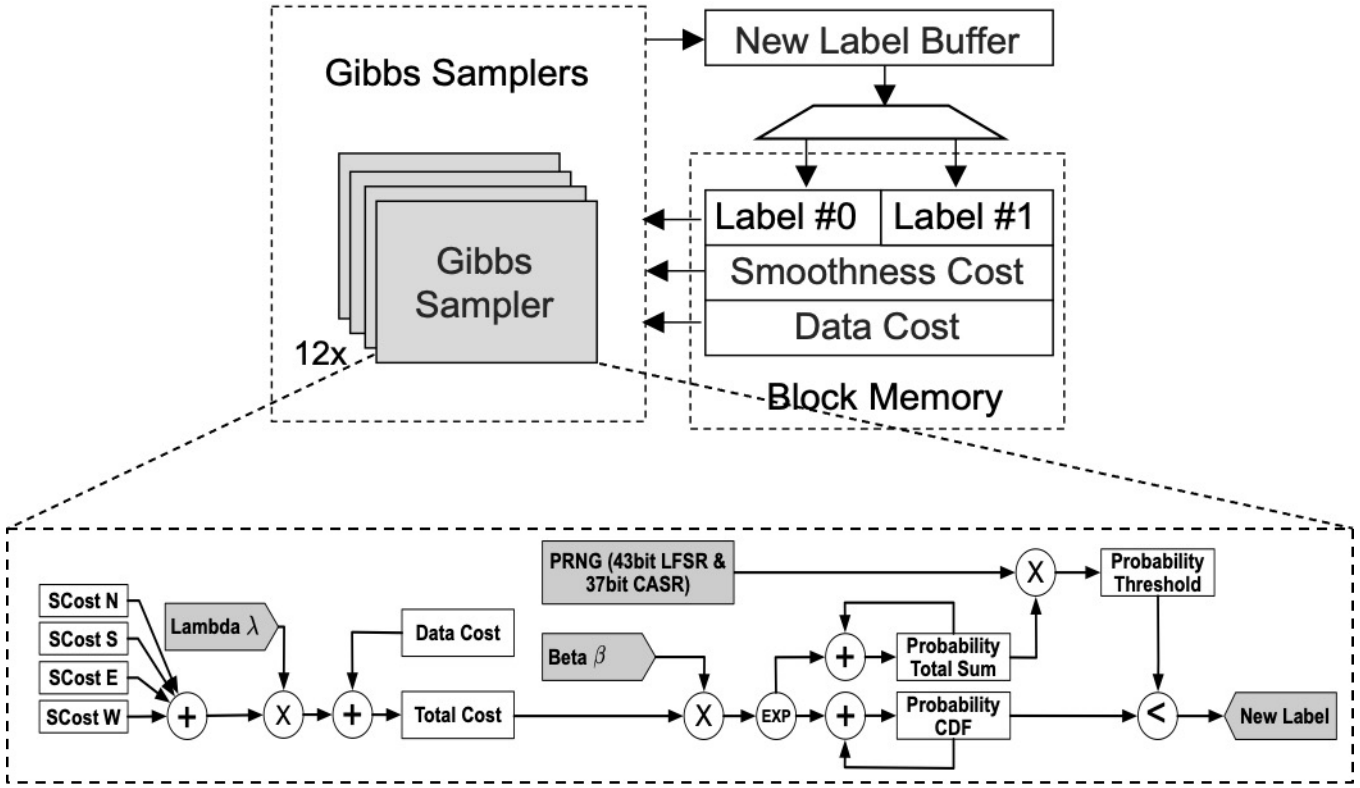


- Nodes representing input speech features
- Nodes representing output labels corresponding to feature locations
- Node being sampled
- Observed node
- Neighbor labels



[2] G. Ko et al., A 3mm2 Programmable Bayesian Inference Accelerator for Unsupervised Machine Perception using Parallel Gibbs Sampling in 16nm, VLSI Symposium, 2020

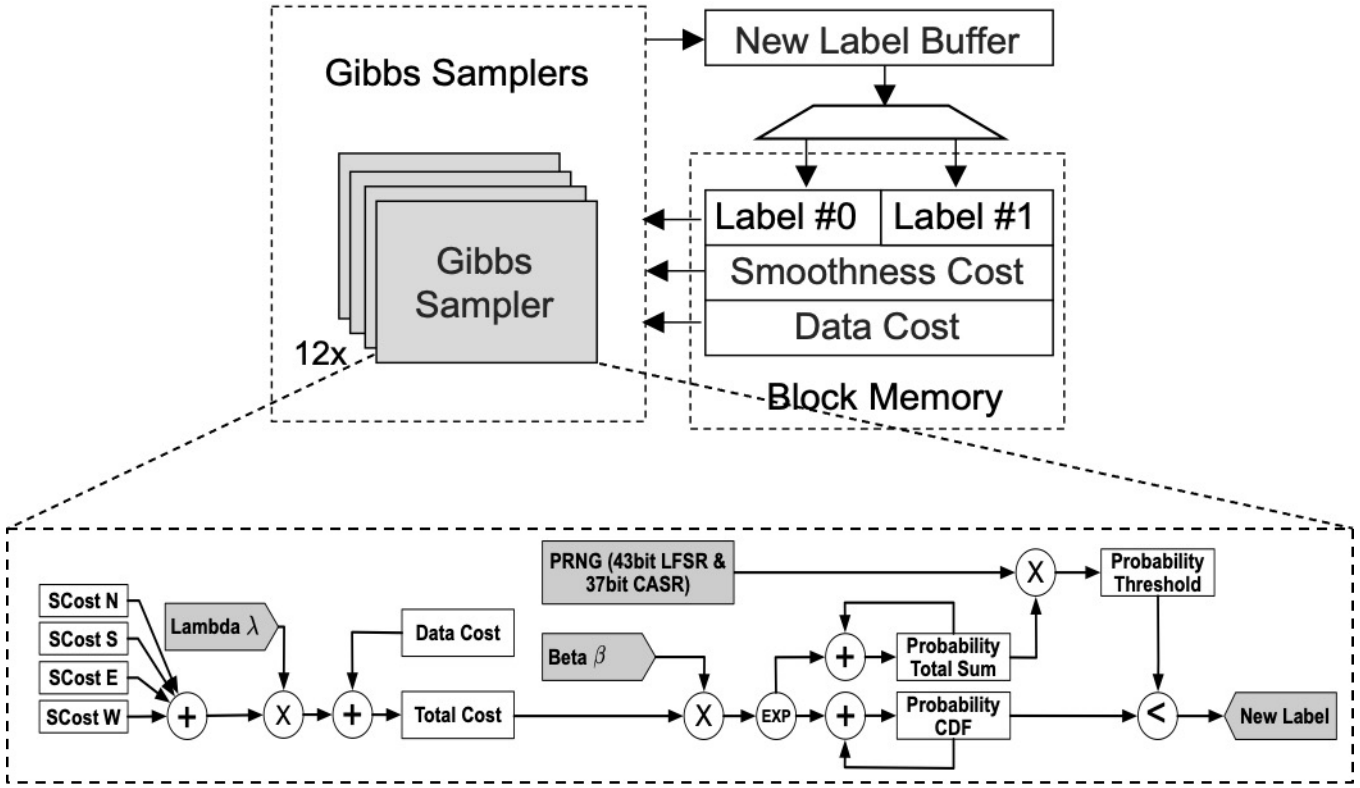
MRF Sound Source Separation Engine (MSSE)



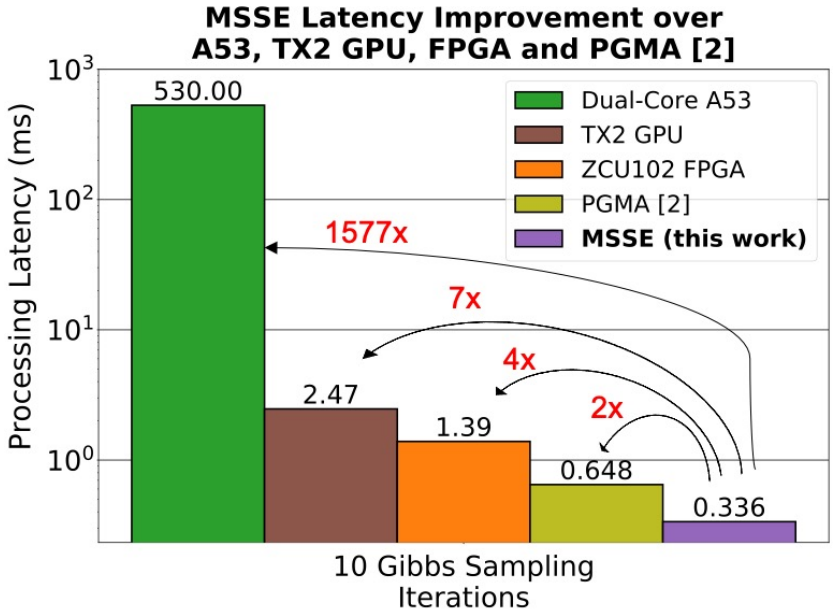
- **Highly optimized for sound source separation while PGMA [2] is a general-purpose Bayesian inference accelerator**
  - **Only supports binary labels**

[2] G. Ko et al., A 3mm2 Programmable Bayesian Inference Accelerator for Unsupervised Machine Perception using Parallel Gibbs Sampling in 16nm, VLSI Symposium, 2020

MRF Sound Source Separation Engine (MSSE)



- **Highly optimized for sound source separation while PGMA [2] is a general-purpose Bayesian inference accelerator**
  - **Only supports binary labels**
  - **2x speedup over PGMA [2]**

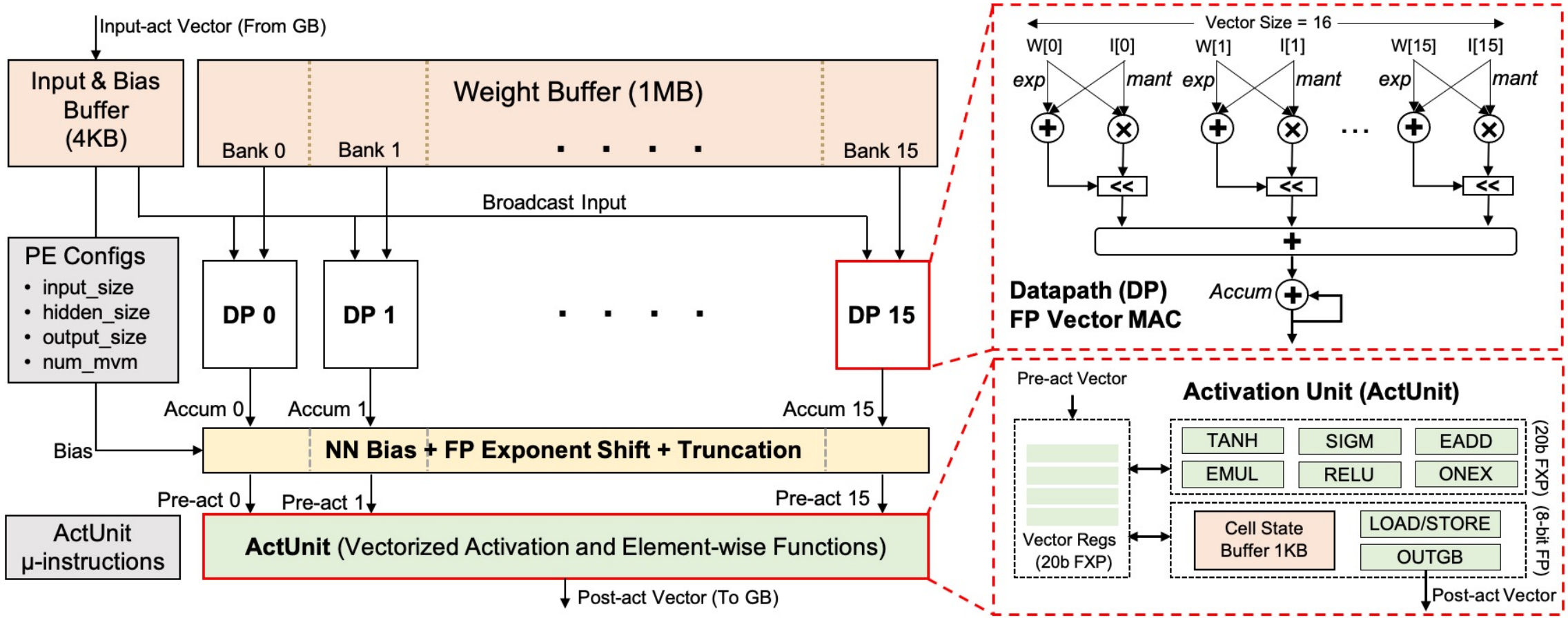


[2] G. Ko et al., A 3mm2 Programmable Bayesian Inference Accelerator for Unsupervised Machine Perception using Parallel Gibbs Sampling in 16nm, VLSI Symposium, 2020

Outline

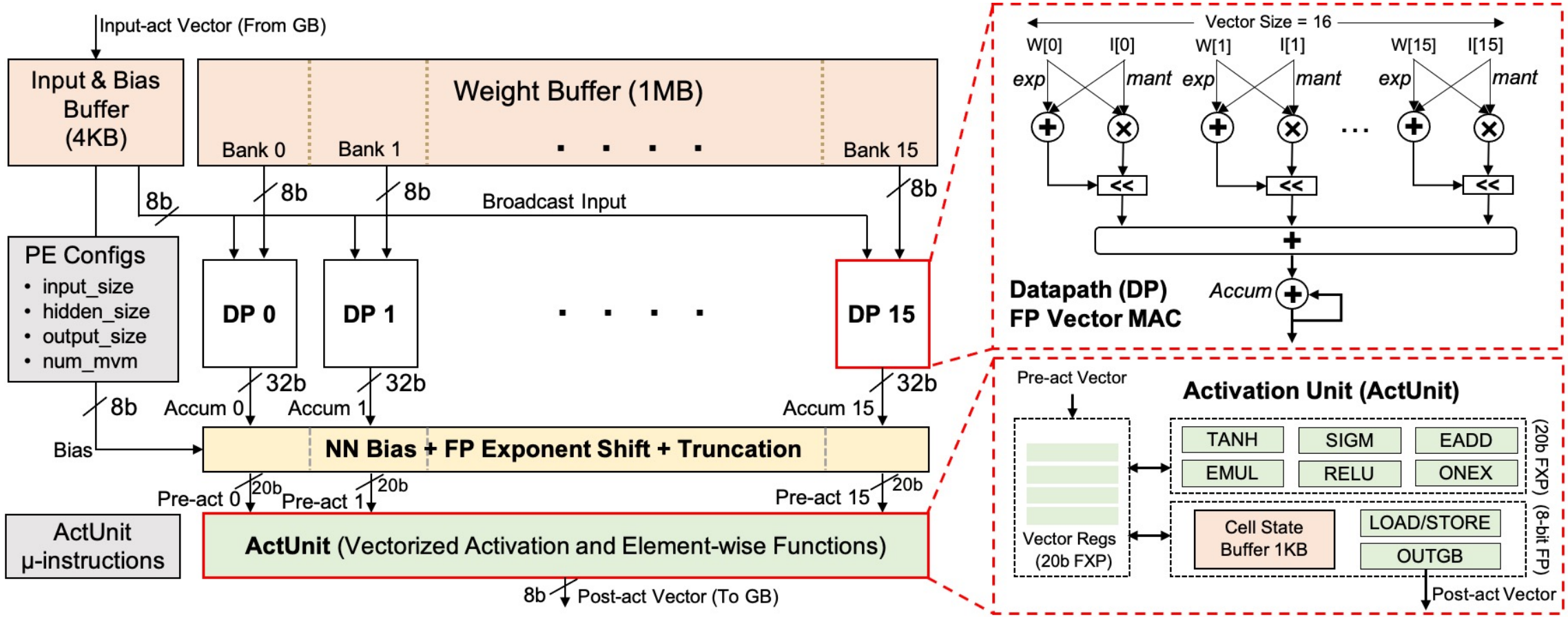
- Motivation
- **Speech-Enhancing ASR**
  - Functional Pipeline
  - 16nm SoC Architecture
  - Markov Source Separation Engine (MSSE)
  - **Attention-based Seq2Seq Accelerator (FlexASR)**
    - **FlexASR Processing Element**
    - FlexASR Multi-Function Global Buffer
- Chip Measurement Results
- Summary

FlexASR Processing Element



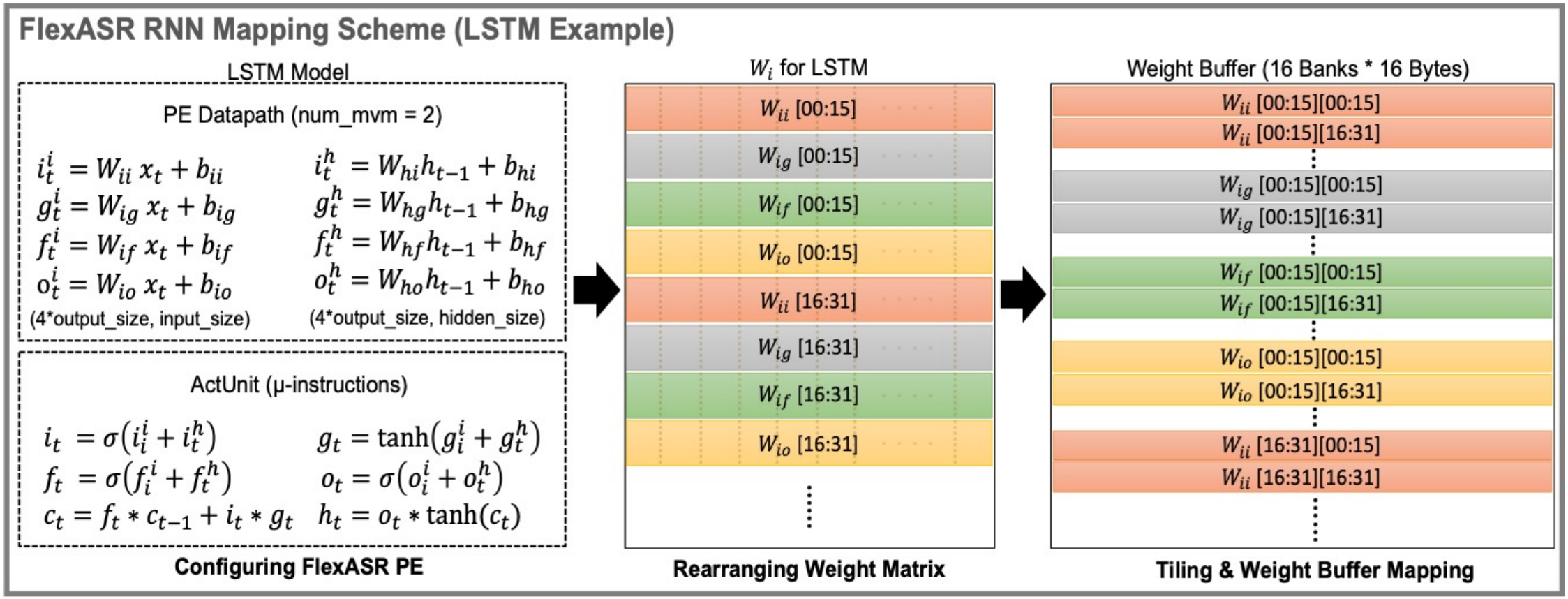
- **Processing element utilizes a floating-point datapath for high accuracy and high dynamic range computations**

FlexASR Processing Element



- **8-bit weight / activation precision with additional support for 4-bit indexes via weight clustering (2x compression)**

Weight Tiling in FlexASR PE

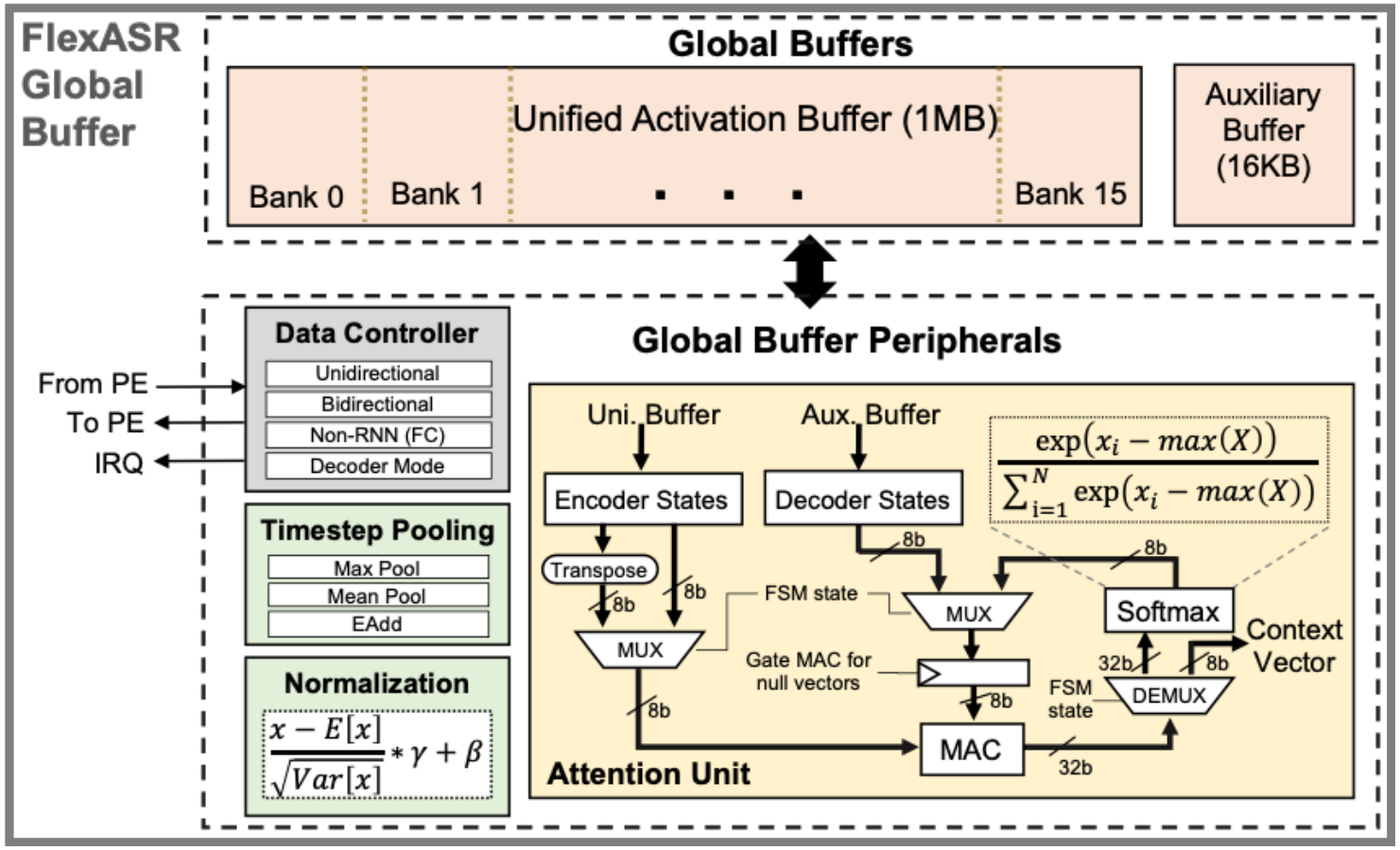


- **16-by-16 RNN weight tiles are reordered and interleaved in weight buffer as shown above in order to ensure hazard-free computation in the activation unit**

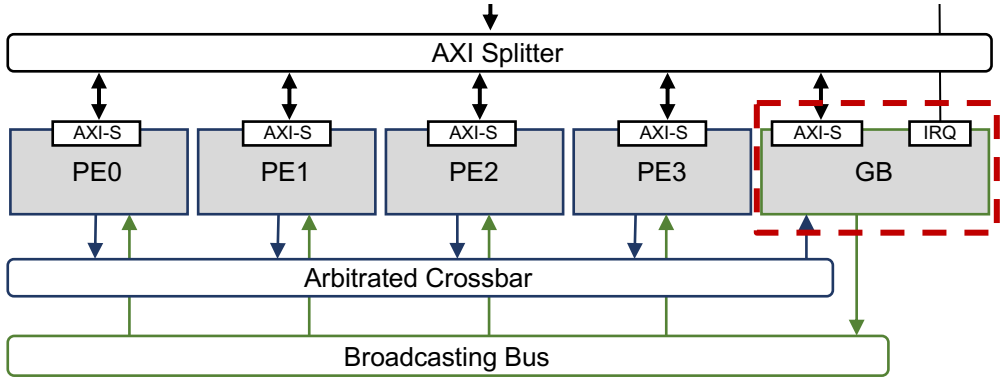
Outline

- Motivation
- **Speech-Enhancing ASR**
  - Functional Pipeline
  - 16nm SoC Architecture
  - Markov Source Separation Engine (MSSE)
  - **Attention-based Seq2Seq Accelerator (FlexASR)**
    - FlexASR Processing Element
    - **FlexASR Multi-Function Global Buffer**
- Chip Measurement Results
- Summary

FlexASR Multi-Function Global Buffer

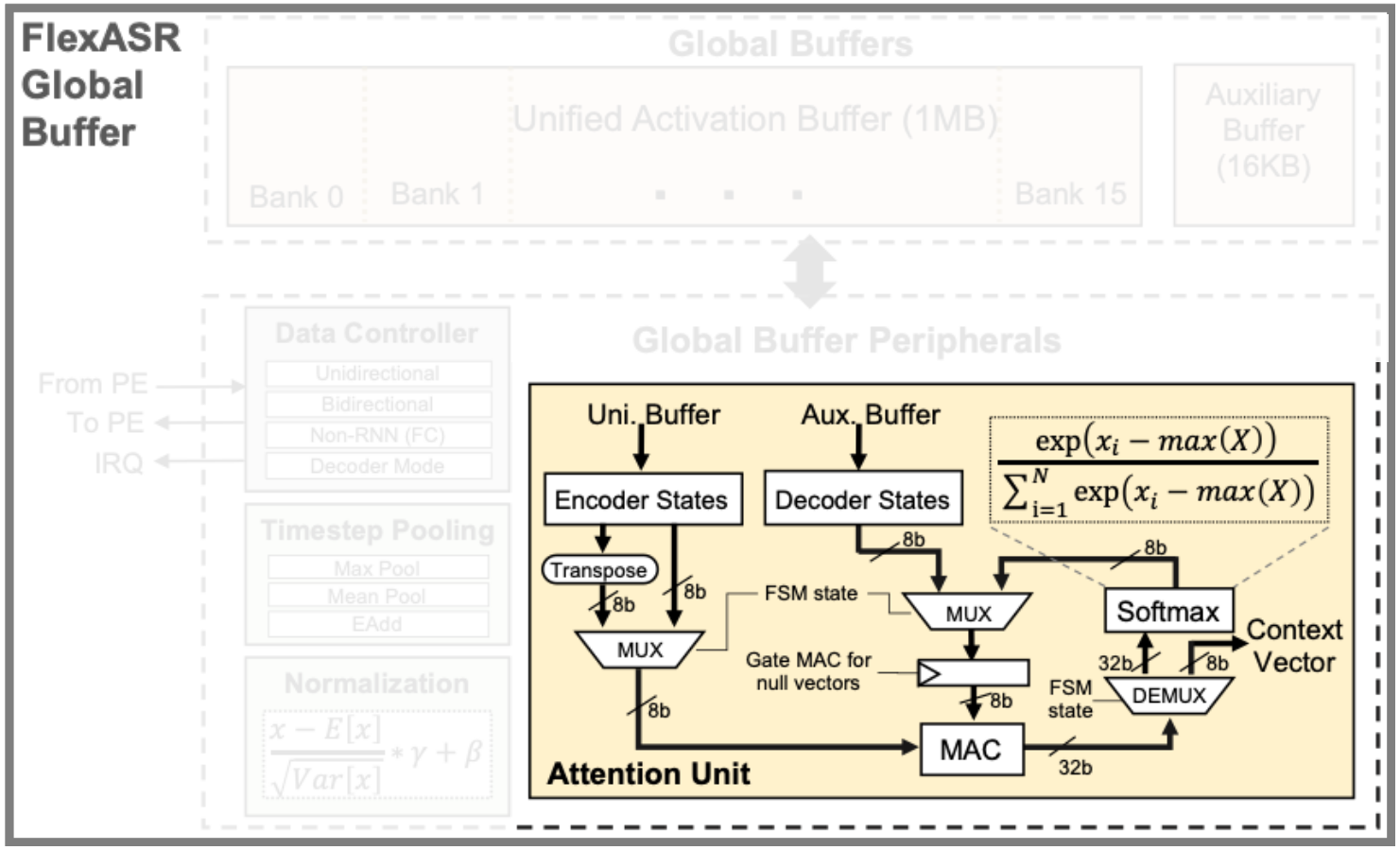


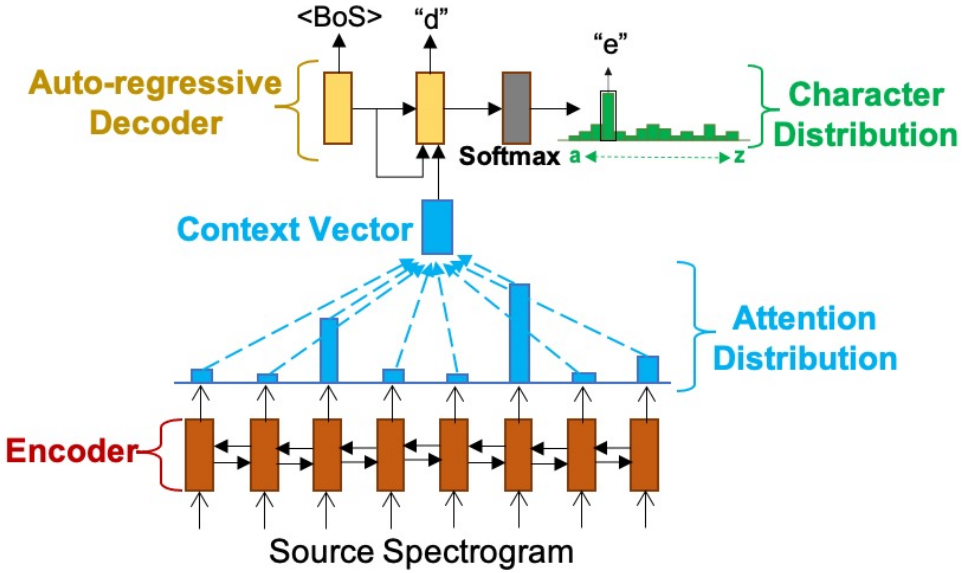
- **Global Buffer (GB) collects and unifies partial RNN outputs from PEs**



FlexASR architecture highlighting its global buffer unit

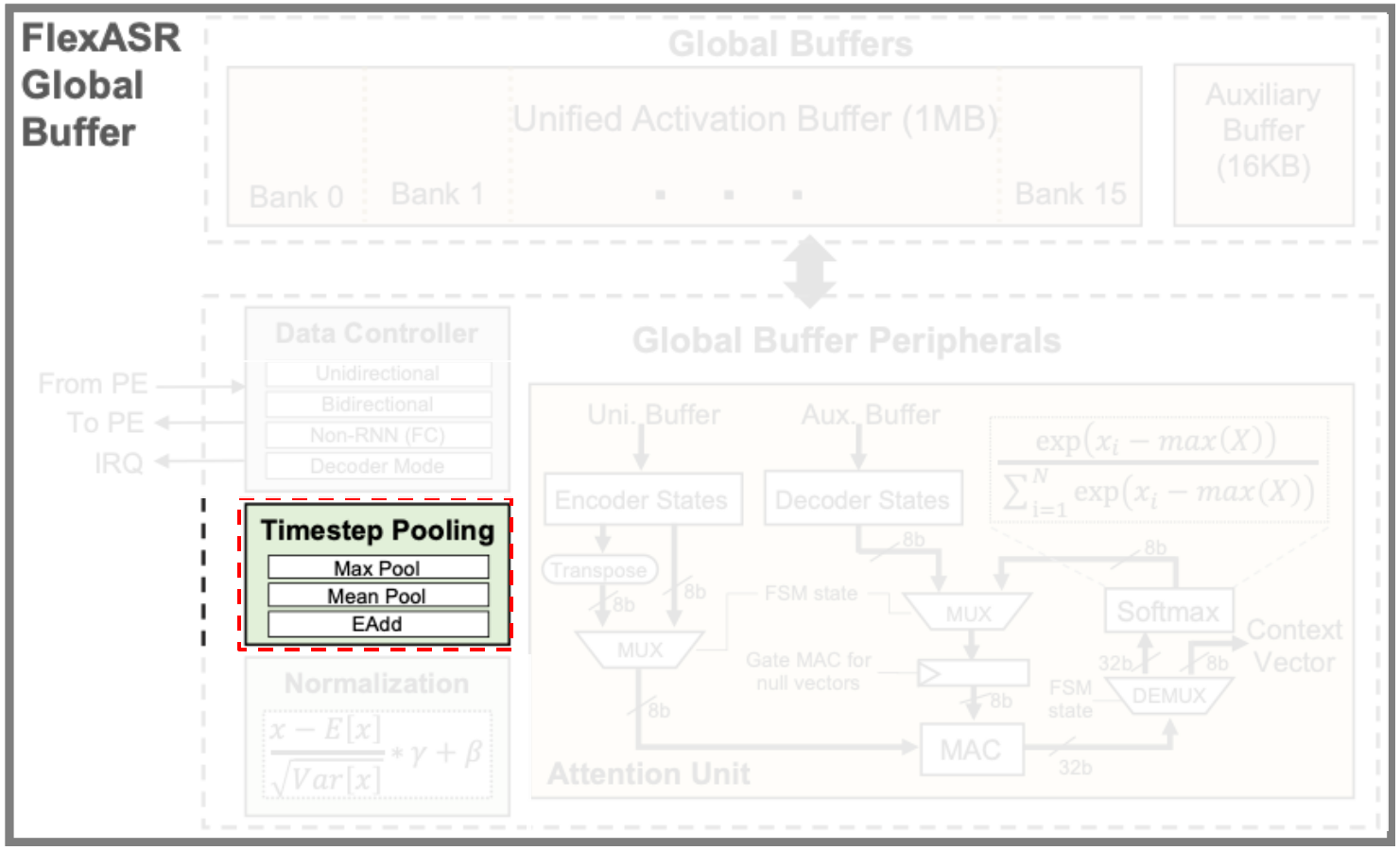
FlexASR GB: Attention Mechanism





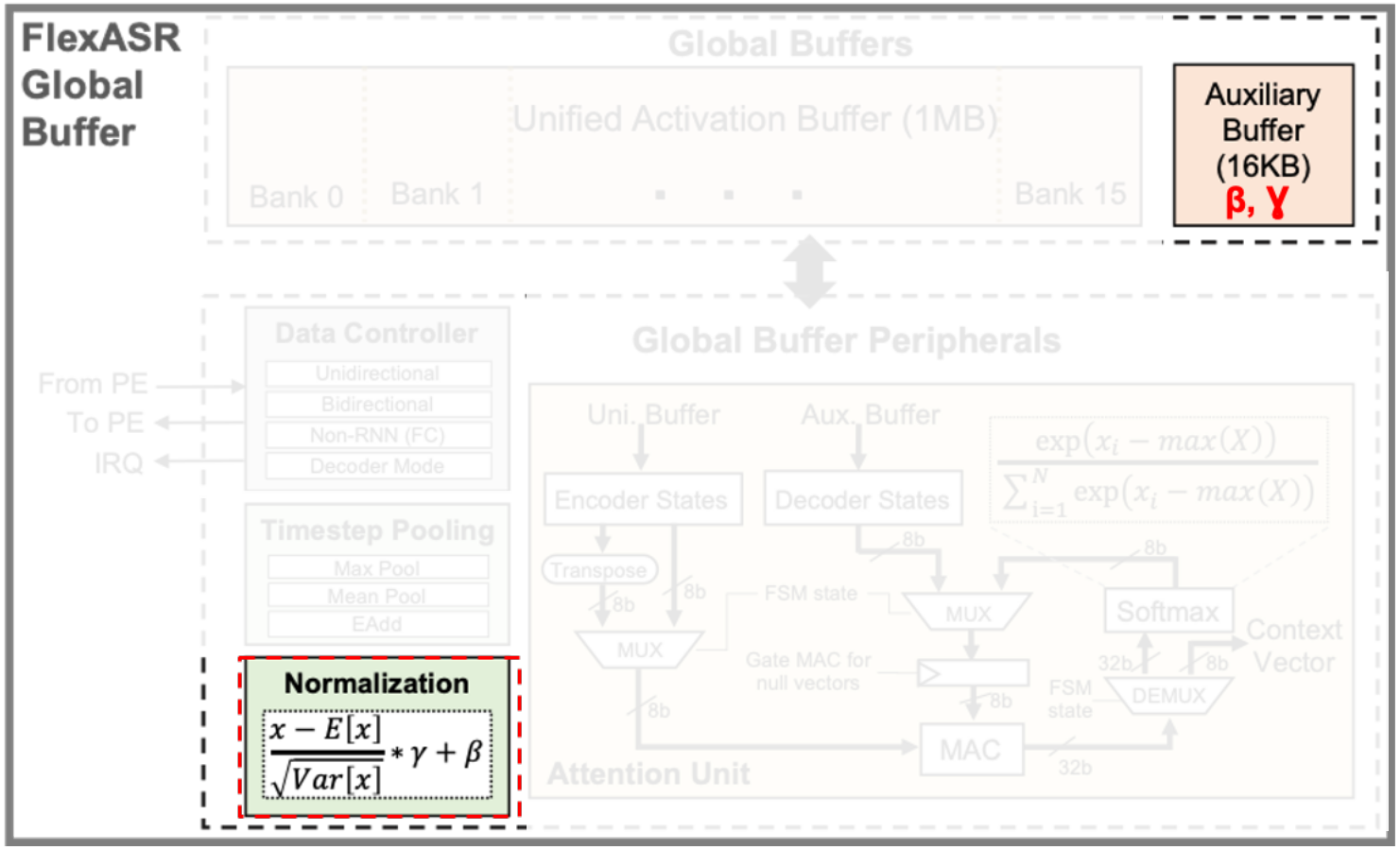
- **Attention mechanism**
  - **Computes numerically stable version of Softmax**
  - **MAC operations are skipped for null decoder states**
    - **Saves energy**

FlexASR GB: Time Step Pooling



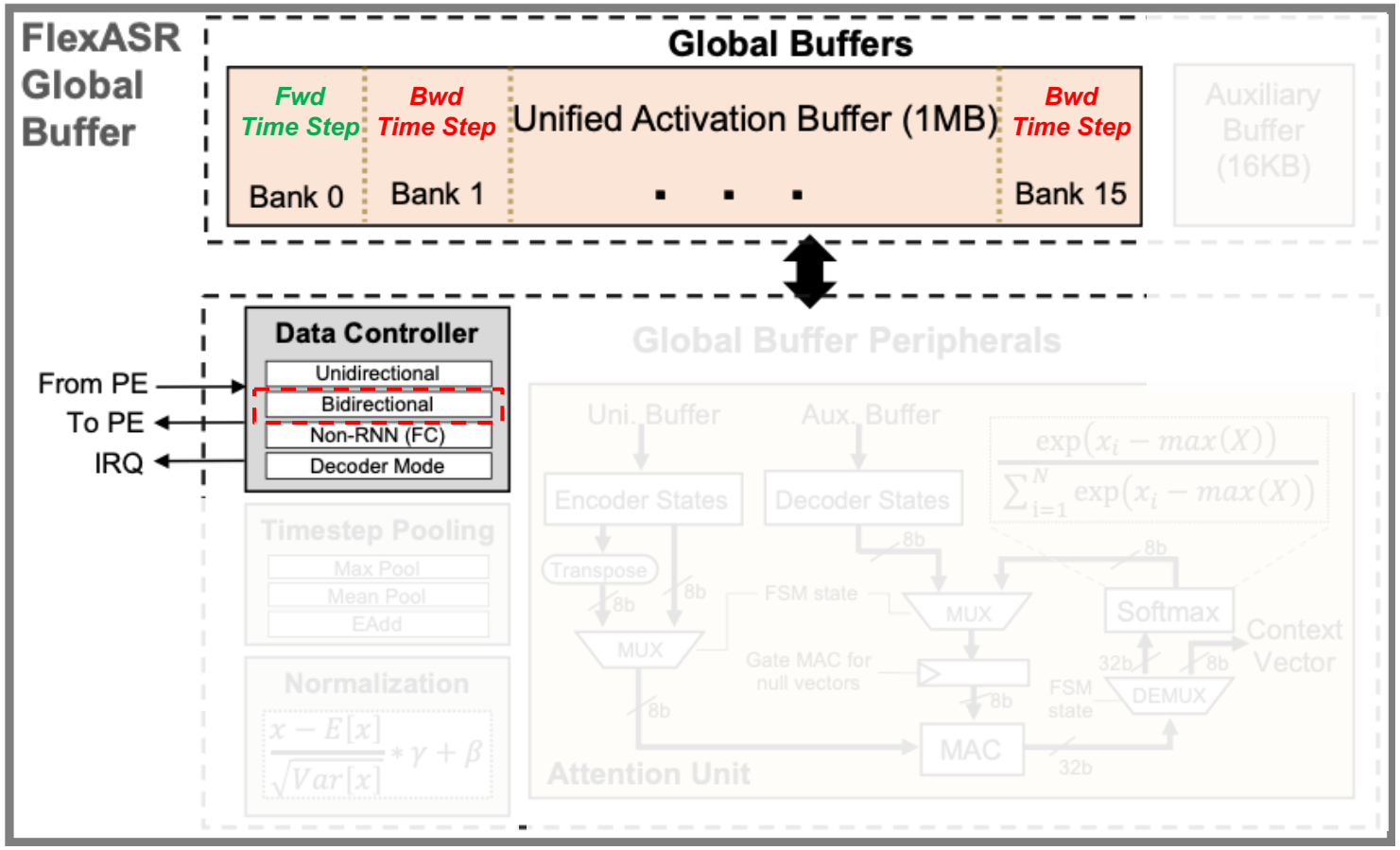
- Attention mechanism
- Time step pooling
  - Mean, Max and Element-Wise Addition

FlexASR GB: Layer Normalization



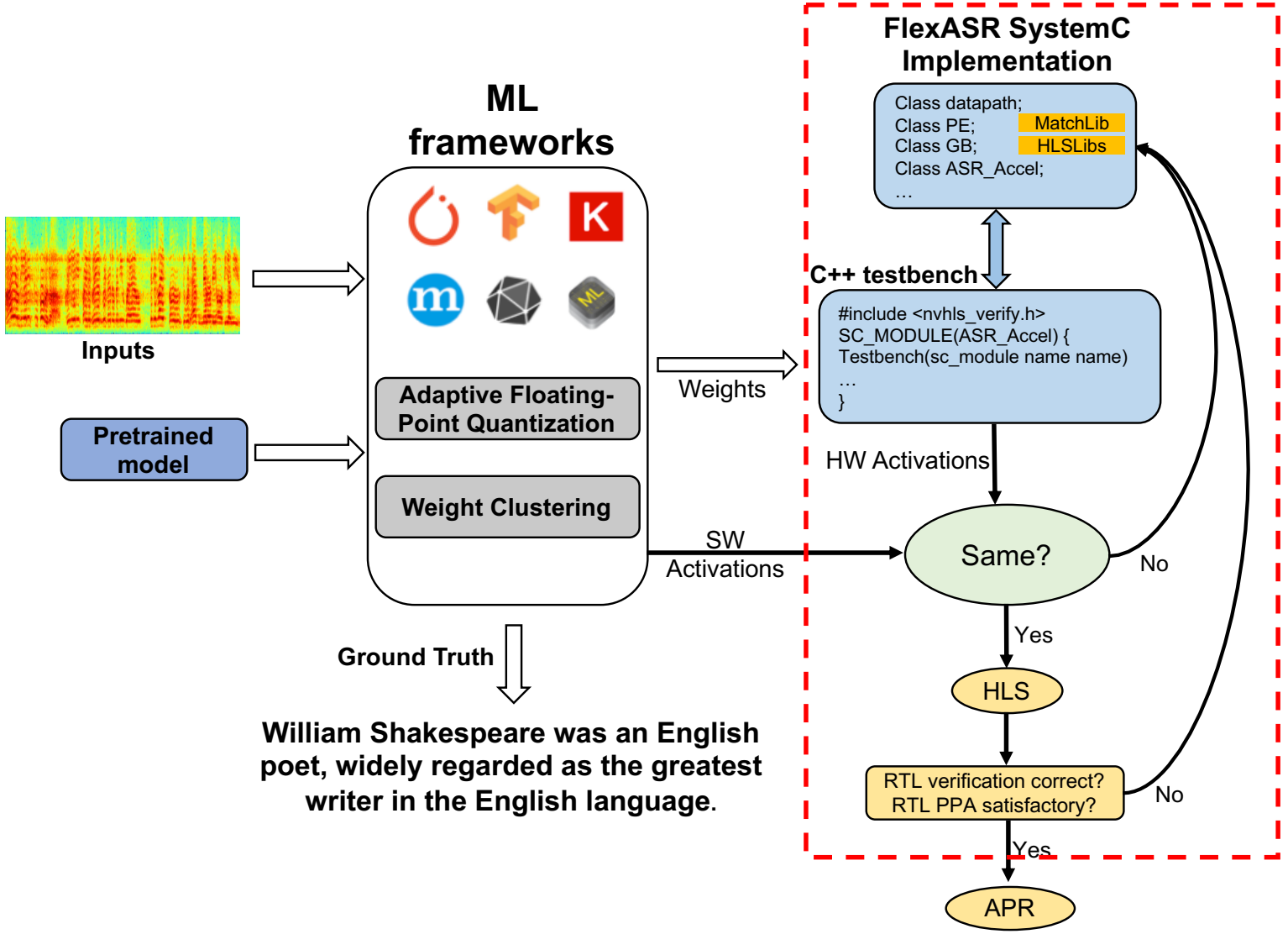
- Attention mechanism
- Time step pooling
- Layer Normalization
  - β and γ parameters stored in the auxiliary buffer

FlexASR GB: Bidir. RNN Operation



- Attention mechanism
- Time step pooling
- Layer Normalization
- Bidirectional RNN operation
  - Stripes forward and backward time steps across alternate banks in the unified activation buffer

FlexASR SW/HW Co-Design and Verification Flow

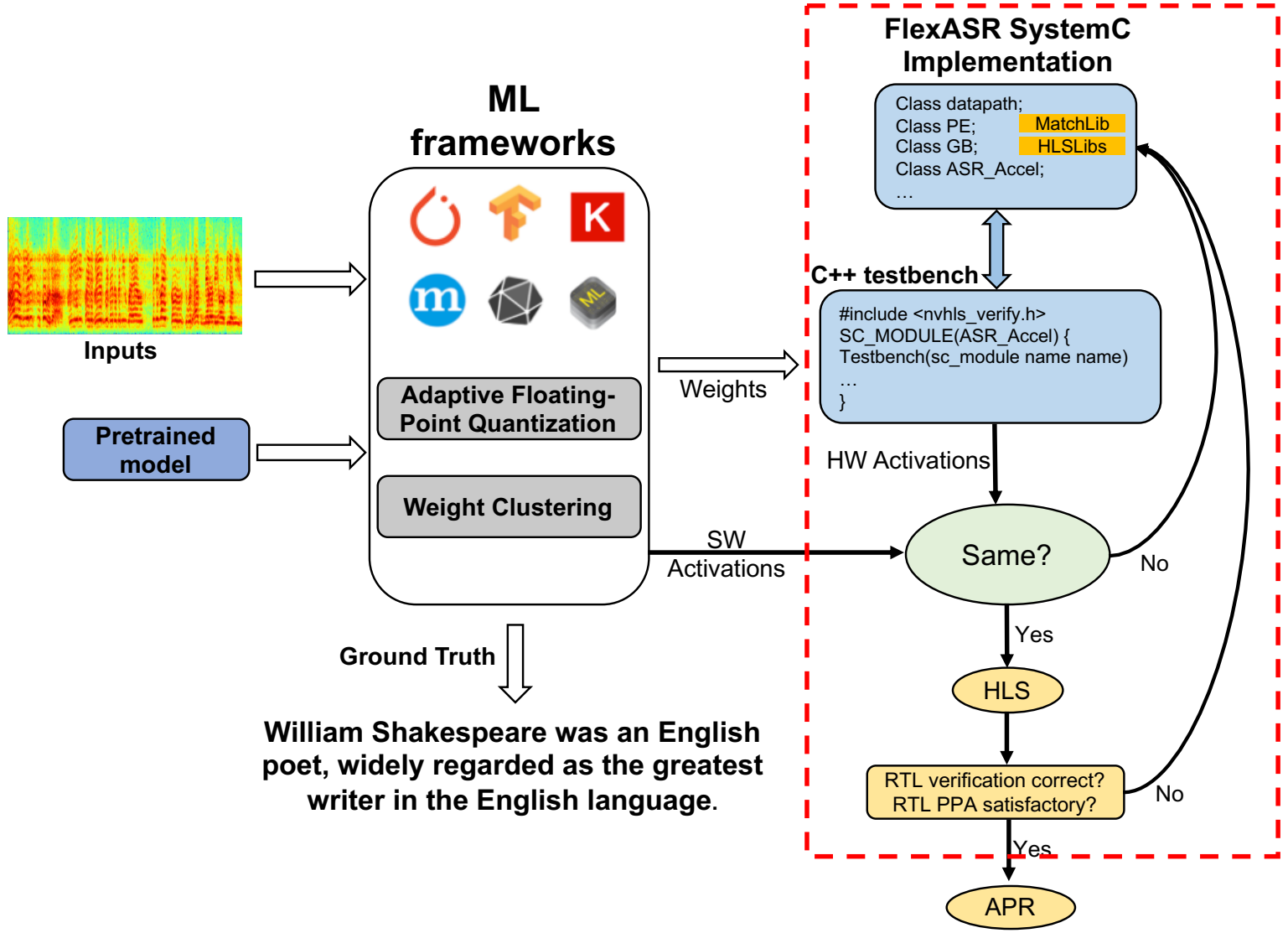


Looped inside Catapult HLS environment

Tunable HW Parameters

- # of PEs
- MAC vector size
- Scratchpad size
- Datapath precision
- # of pipeline stages
- # of initiation intervals

FlexASR SW/HW Co-Design and Verification Flow



Looped inside Catapult HLS environment

Tunable HW Parameters

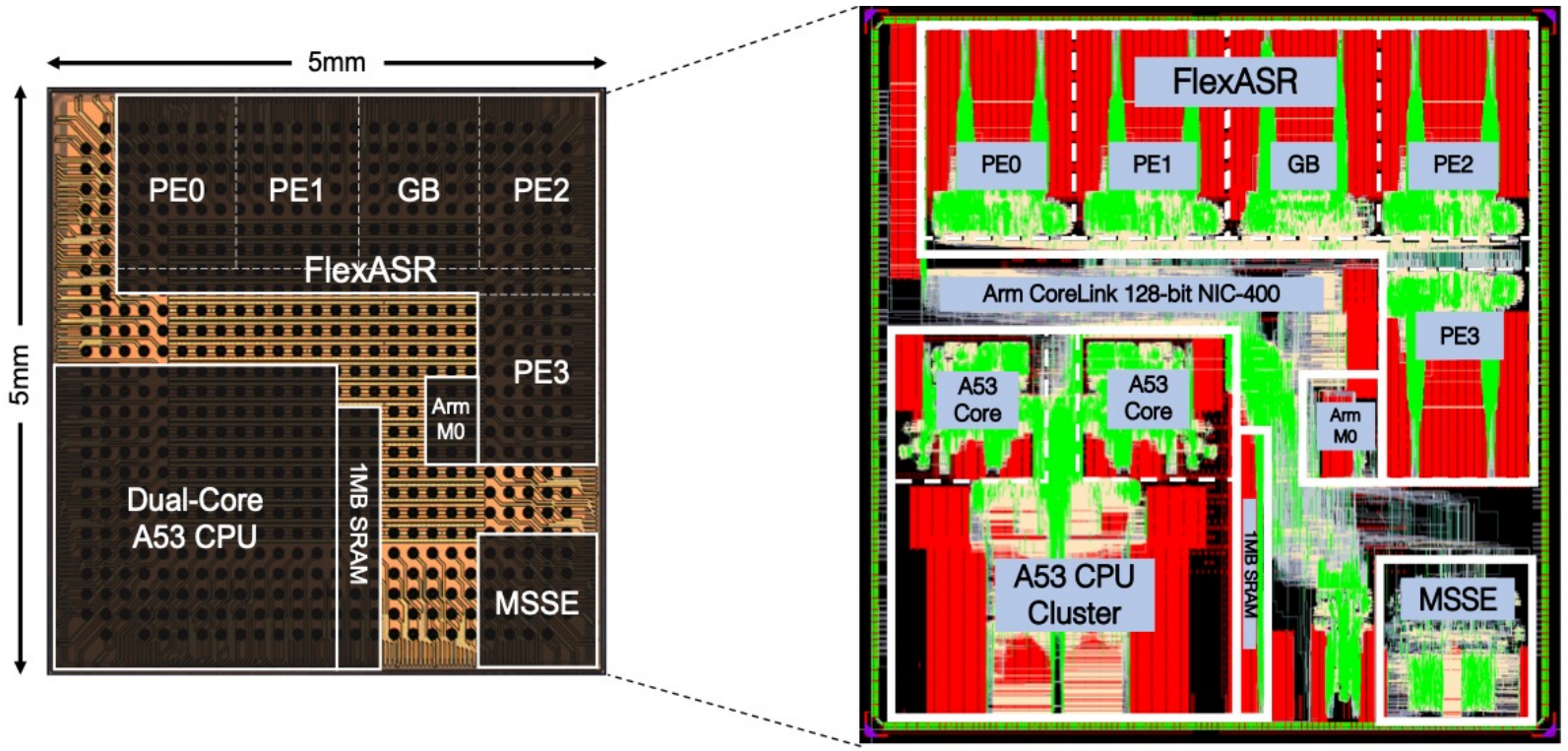
- # of PEs
- MAC vector size
- Scratchpad size
- Datapath precision
- # of pipeline stages
- # of initiation intervals

From start of SystemC coding to tapeout in less than 4 months!

Outline

- Motivation
- Speech-Enhancing ASR
  - Functional Pipeline
  - 16nm SoC Architecture
  - Markov Source Separation Engine (MSSE)
  - Attention-based Seq2Seq Accelerator (FlexASR)
    - FlexASR Processing Element
    - FlexASR Multi-Function Global Buffer
- **Chip Measurement Results**
- Summary

16nm Test Chip

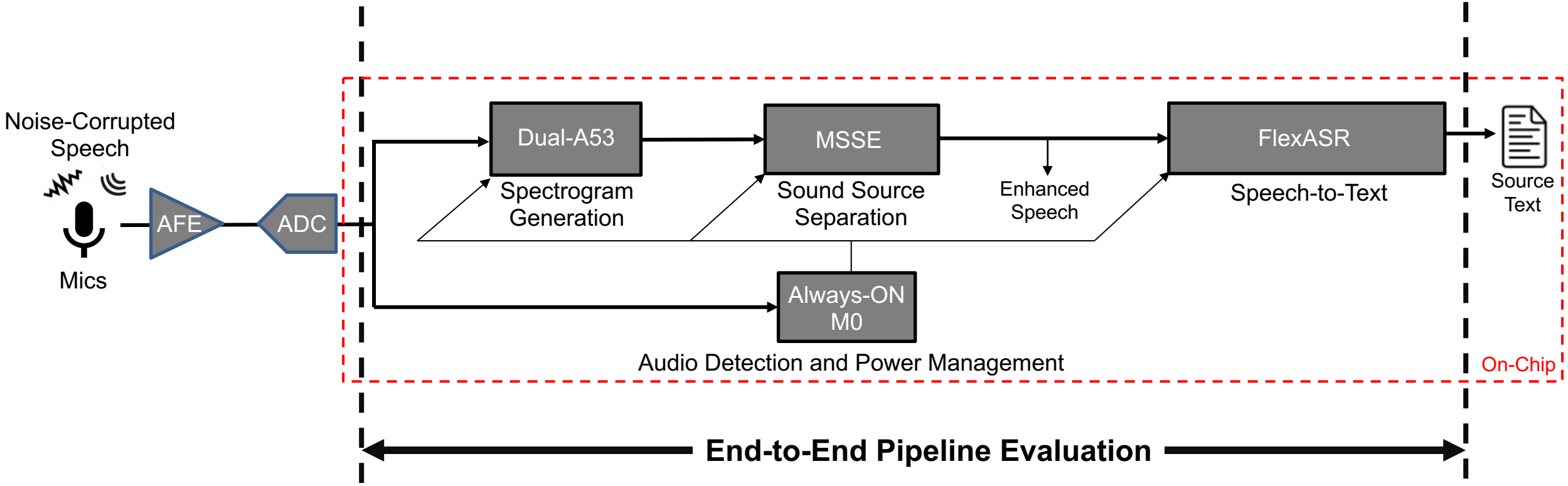


Technology	TSMC 16nm FFC
Die Area	25mm^2
Total SRAM	9.8MB
Gate Count	11M
Clock Domains	6
Power Domains	5
Supply Voltage	0.55 – 1V
Packaging	Flip-chip BGA-672

Memory Breakdown (in MB)

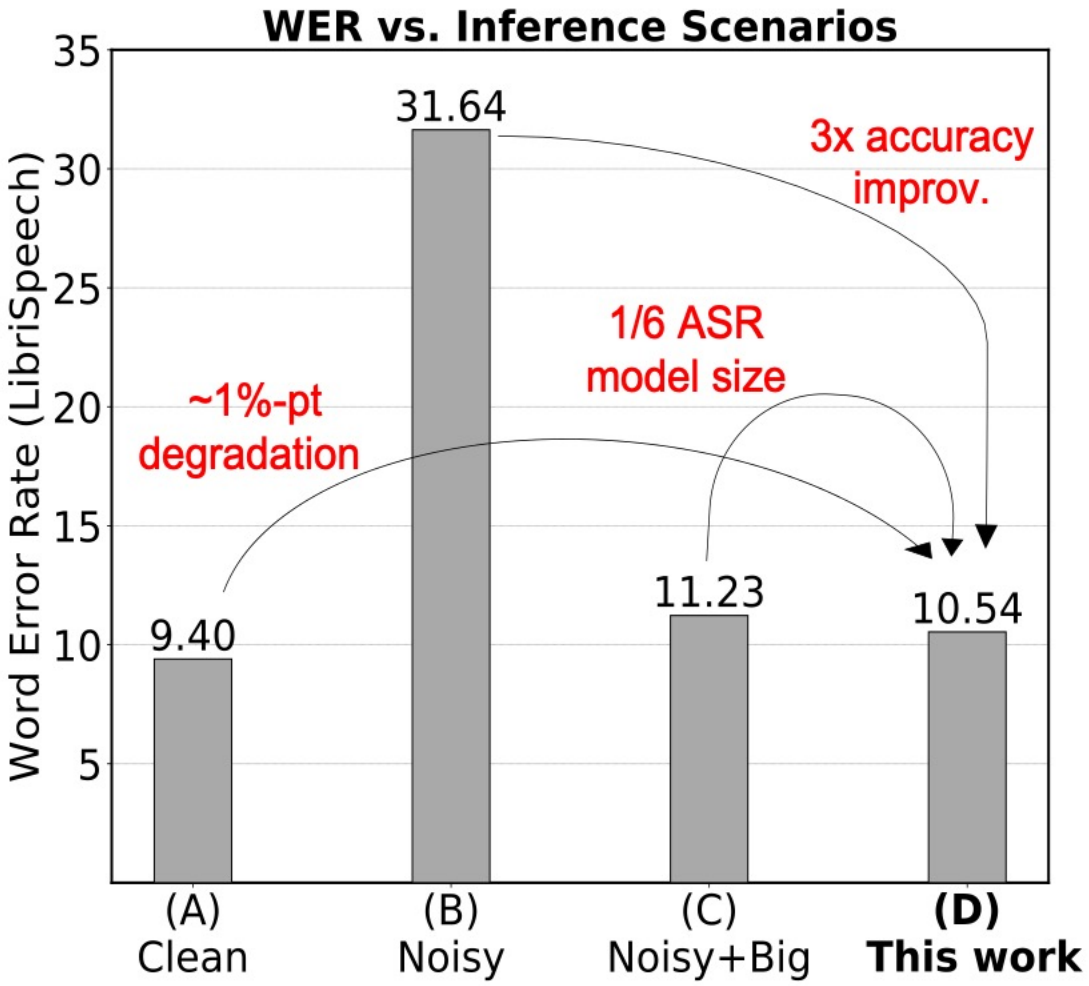
FlexASR	PEs	4.2
	GB	1.1
MSSE		0.103
ARM Dual A53		2.41
ARM M0		0.128
SoC Top Level		1.8
Total		9.8

End-to-End Evaluation



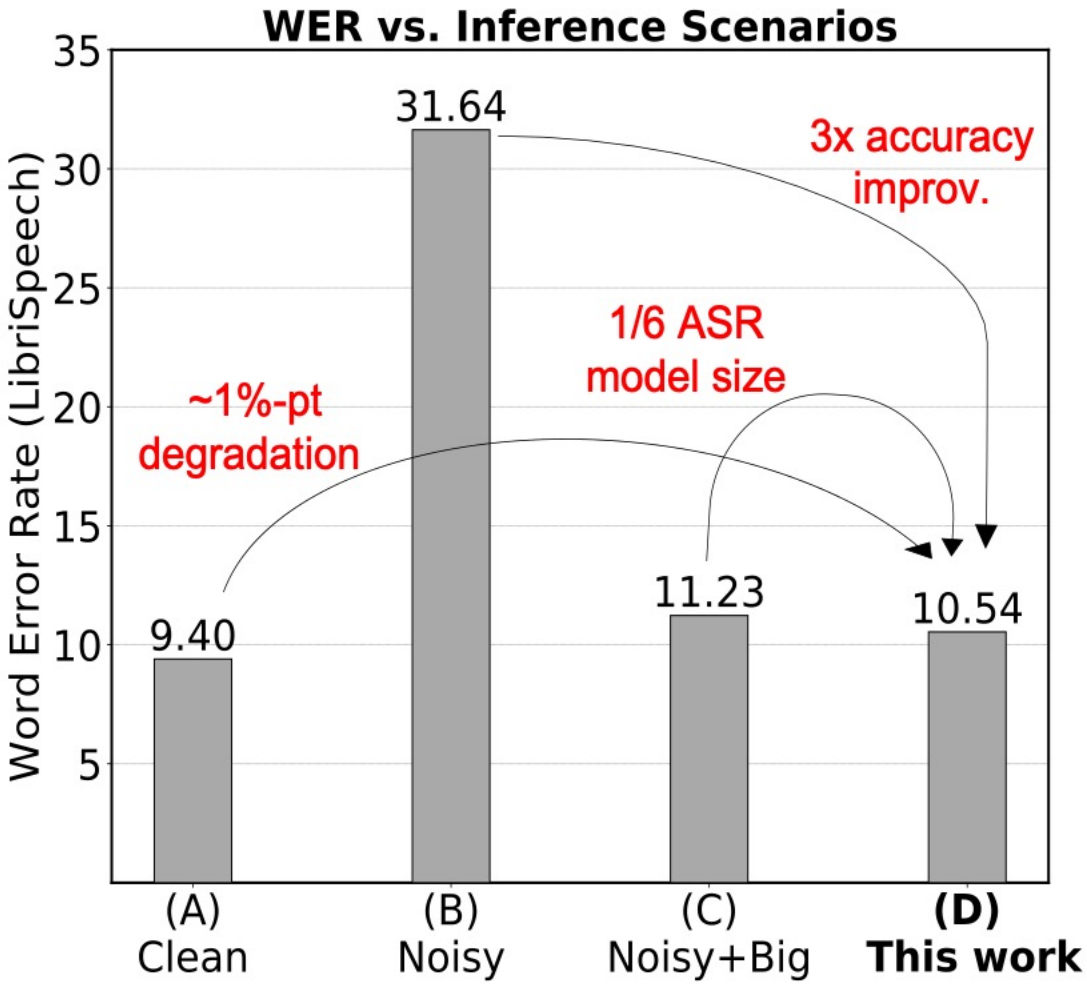
ASR Accuracy

Inference Scenarios	ASR Model Size (MB)	Model fit on-chip?	Noise Resistant?
(A) Noiseless audio of the speaker	3.5	Yes	No
(B) Noise mixed with the speaker's voice at 0.9dB SNR	3.5	Yes	No
(C) Large ASR model trained with a noise-corrupted LibriSpeech dataset	22	No	Yes
(D) Speech-Enhanced ASR Pipeline (This work)	3.5	Yes	Yes



ASR Accuracy

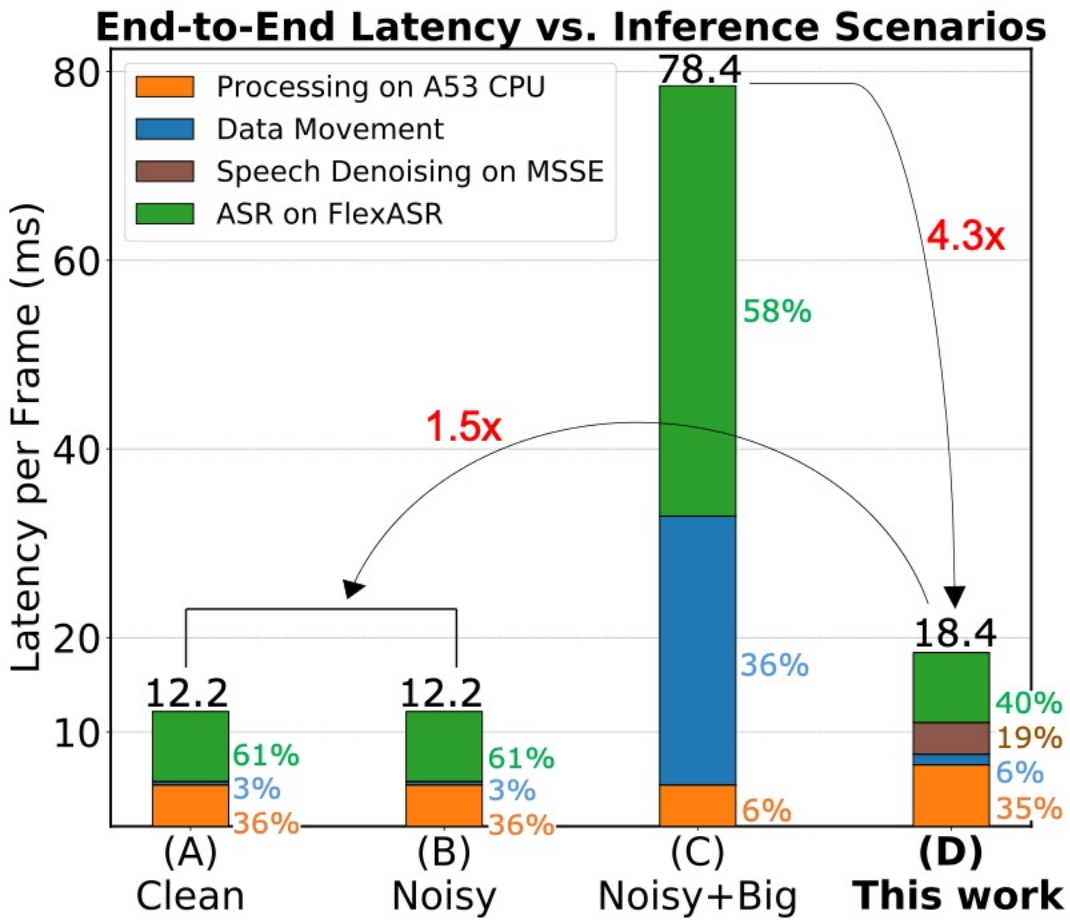
Inference Scenarios	ASR Model Size (MB)	Model fit on-chip?	Noise Resistant?
(A) Noiseless audio of the speaker	3.5	Yes	No
(B) Noise mixed with the speaker's voice at 0.9dB SNR	3.5	Yes	No
(C) Large ASR model trained with a noise-corrupted LibriSpeech dataset	22	No	Yes
(D) Speech-Enhanced ASR Pipeline (This work)	3.5	Yes	Yes



- **Speech-enhancing pipeline allows much smaller ASR models to be stored on-chip**
  - **Obviates very inefficient strategy of scaling up the DNN model in order to achieve noise robustness**

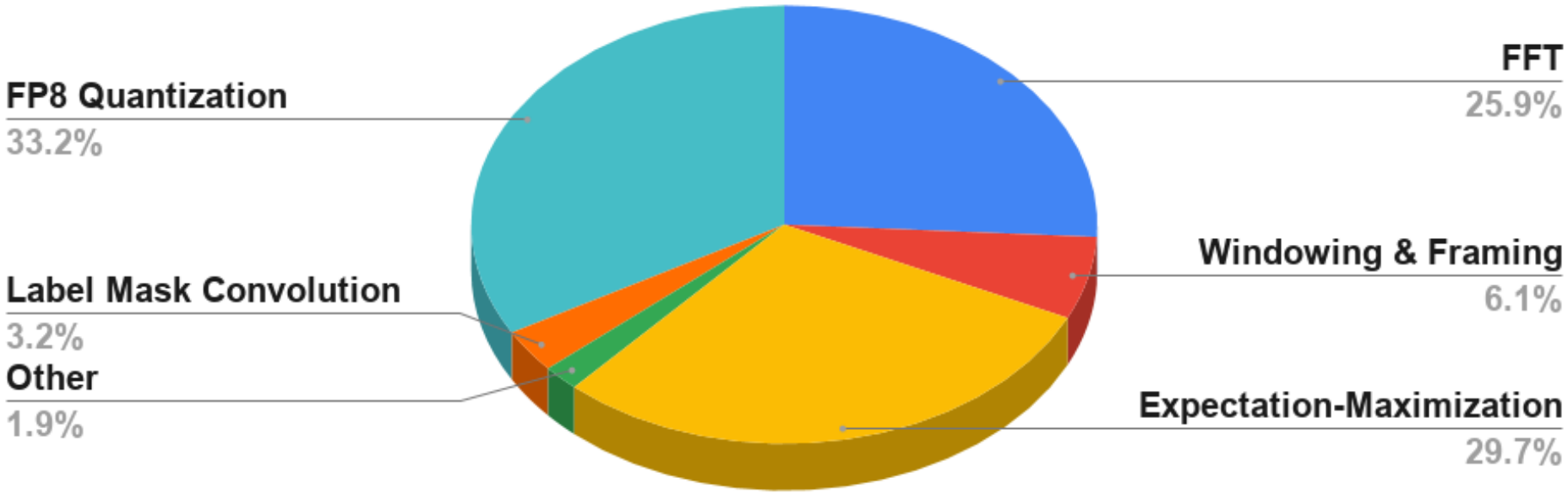
End-to-End ASR Latency

Inference Scenarios	ASR Model Size (MB)	Model fit on-chip?	Noise Resistant?
(A) Noiseless audio of the speaker	3.5	Yes	No
(B) Noise mixed with the speaker's voice at 0.9dB SNR	3.5	Yes	No
(C) Large ASR model trained with a noise-corrupted LibriSpeech dataset	22	No	Yes
(D) Speech-Enhanced ASR Pipeline (This work)	3.5	Yes	Yes



- **Speech-enhancing ASR Pipeline is:**
  - **4.3x faster compared to the scaled-up DNN approach (c)**
  - **Speech Denoising and ASR account for 19% and 40% of latency**

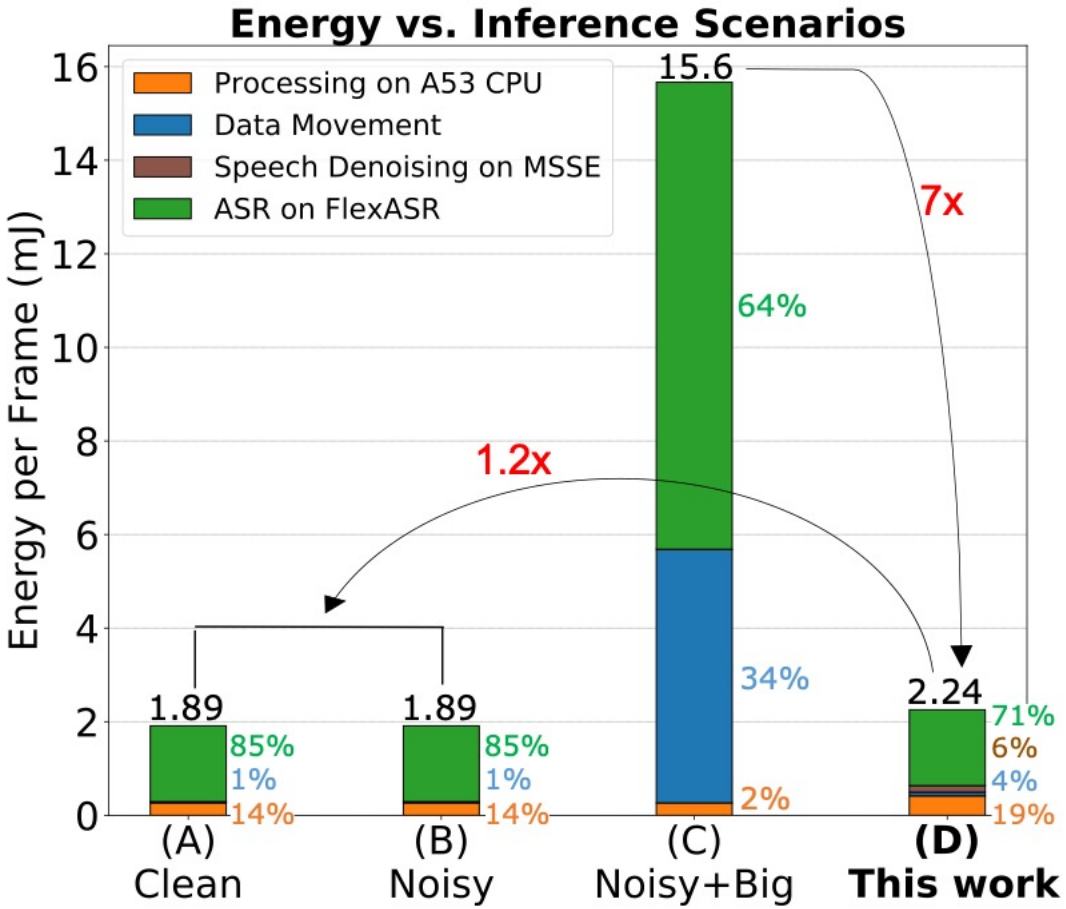
Breakdown of CPU Work



- **FFT, Windowing and Framing:** spectrogram synthesis
- **FP8 Quantization:** 32-bit fixed-point to 8-bit floating-point conversion needed for input to FlexASR
- **Expectation-Maximization Algorithm:** outer loop to update the MRF distribution after Gibbs sampling
- **Label Mask Convolution:** *clean* speech extraction from MSSE binary label mask
- **Other:** instruction dispatch, IRQ handling, misc. application logic
- **All tasks are vectorized on the A53 where applicable**

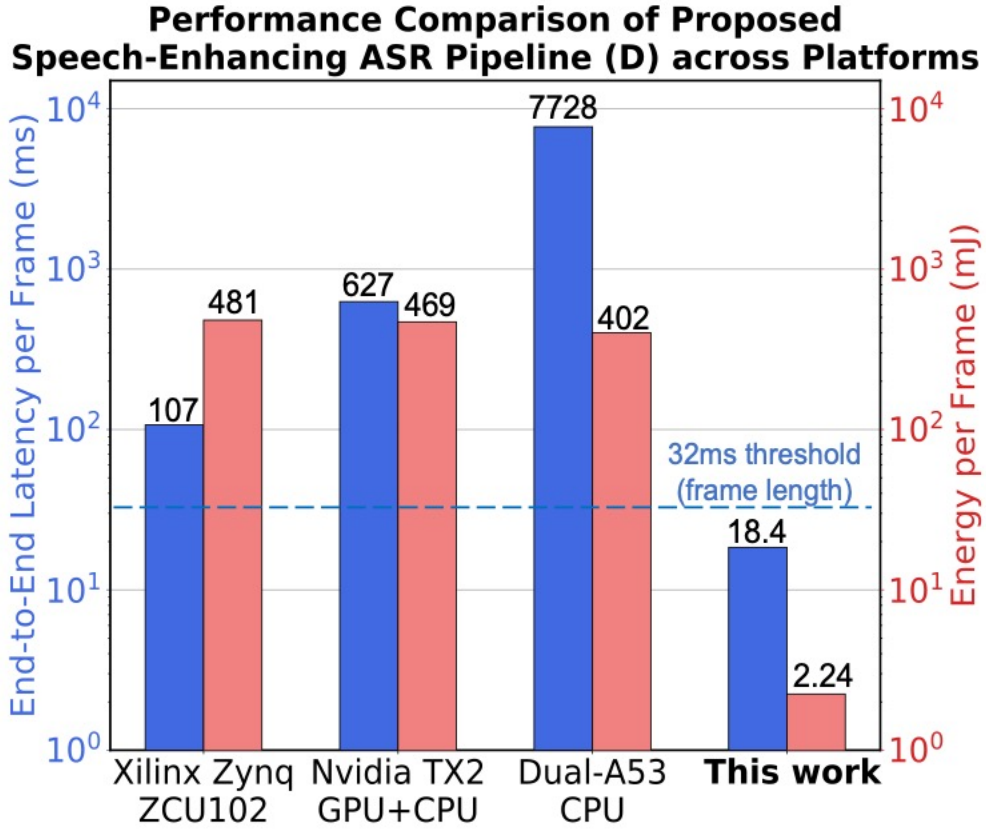
End-to-End ASR Energy

Inference Scenarios	ASR Model Size (MB)	Model fit on-chip?	Noise Resistant?
(A) Noiseless audio of the speaker	3.5	Yes	No
(B) Noise mixed with the speaker's voice at 0.9dB SNR	3.5	Yes	No
(C) Large ASR model trained with a noise-corrupted LibriSpeech dataset	22	No	Yes
(D) Speech-Enhanced ASR Pipeline (This work)	3.5	Yes	Yes



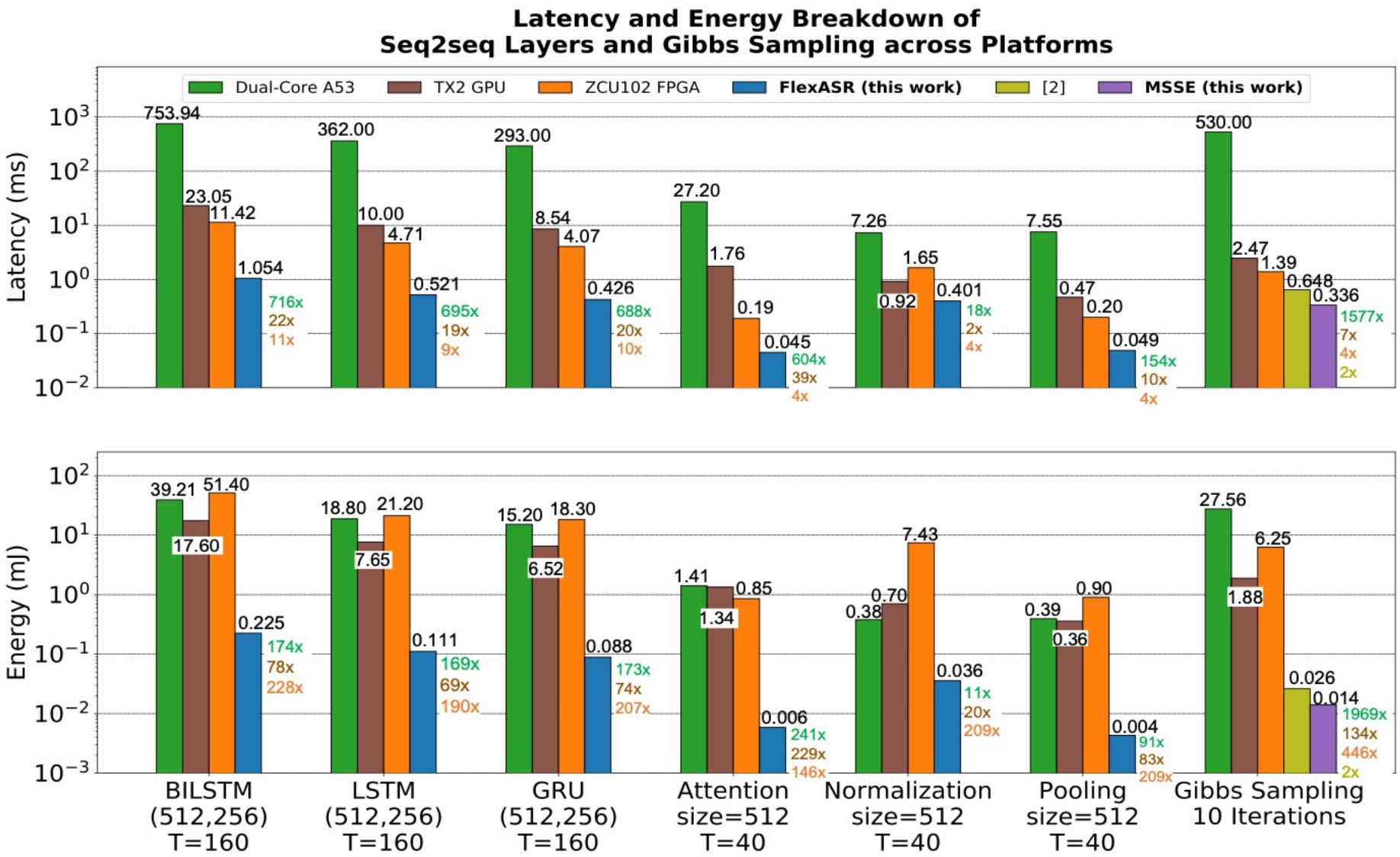
- **Speech-enhancing ASR Pipeline is:**
  - **7x more energy-efficient compared to the scaled-up DNN approach (c)**
  - **ASR dominates energy consumption**

Platform Comparison



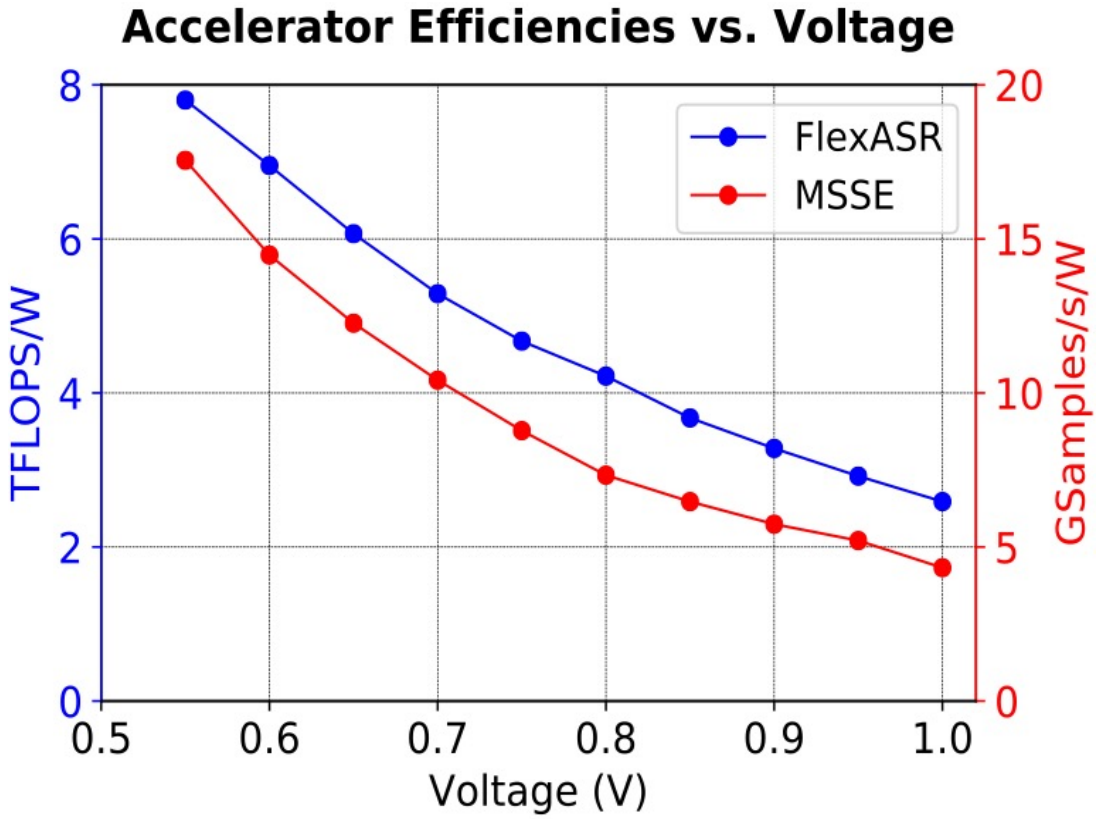
- **Speech-enhancing pipeline achieves real-time performance unlike commercial edge platforms despite substantial energy expenditures**

Per-Layer Platform Comparison



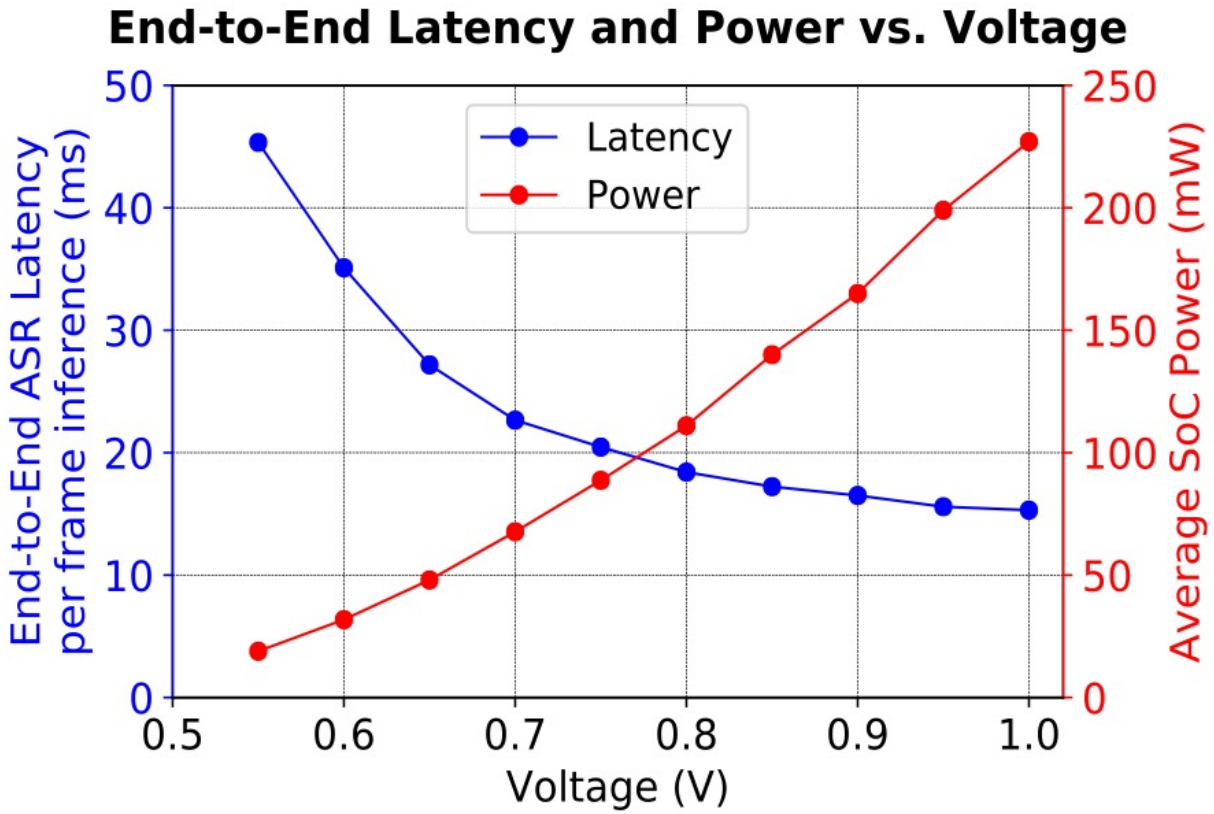
- **FlexASR: 4x – 716x faster, 11x – 228x more energy efficient**
- **MSSE: 2x – 1577x faster, 2x – 1969x more energy efficient**

Accelerator Efficiencies



- As voltage sweeps from 0.5V to 1.0V
  - MSSE: 4.33 – 17.6 GSamples/s/W
  - FlexASR: 2.6 – 7.8 TFLOPs/W

Benefits with Scaled Vdd



- As voltage sweeps from 0.5V to 1.0V
  - Latency: 45ms to 15ms
  - SoC Power: 19mW to 227mW

Comparison Table

	[4]	[5]	[6]	[7]	**This work**
Technology	65 nm	65 nm	65 nm	28 nm	16 nm
Core Dimension	9.6 mm^2	6.2 mm^2	2.6 mm^2	1.3 mm^2	21.8 mm^2
Application	ASR	KWS	KWS	ASR	**Speech Denoising, ASR**
Algorithm	HMM	RNN	RNN	CNN	**Bayesian MRF + Attention-based RNN**
On-Chip Speech Denoising	No	No	No	No	**Yes (7.3 dB SDR)**
Dataset (Vocabulary Size)	News 2 (145k words)	Smart Home (11 words)	GSCD (30 words)	TIMIT (6k words)	**LibriSpeech (200k words)**
Datatype	4-12b FxP	1b FxP	4b/8b FxP	1b FxP	Denoising: 32b FxP ASR: 8b FP
Total SRAM	730 KB	18 KB	105 KB	52 KB	9.8 MB
Supply Voltage	0.6 V – 1.2 V	0.9 V – 1.1 V	0.6 V – 1.2V	0.57 V – 0.9 V	0.55 V – 1.0 V
Frequency	3 – 86 MHz	5 – 75 MHz	250 kHz – 12.5 MHz	2.5 – 50 MHz	130 – 775 MHz
Latency per Frame	/	0.127 ms	16 ms	0.5 ms – 25 ms	**15 – 45 ms**
Power	7.78 mW @ 0.9V/40MHz	26 mW @ 0.9V/75MHz	18.3 uW @ 0.6V/250 KHz	1.42 mW @ 0.58V/20MHz	111 mW @ 0.8V/Fmax

- **First work to demonstrate on-chip support for denoised, large-vocabulary, attention-based ASR with competitive latency**

Outline

- Motivation
- Speech-Enhancing ASR
  - Functional Pipeline
  - 16nm SoC Architecture
  - Markov Source Separation Engine (MSSE)
  - Attention-based Seq2Seq Accelerator (FlexASR)
    - FlexASR Processing Element
    - FlexASR Multi-Function Global Buffer
- Chip Measurement Results
- **Summary**

Summary

- Attention-based bidirectional RNNs enables significant WER improvement by enforcing context understanding
- Noise-isolating ASR is essential for edge/IoT applications
- A 16nm SoC executing an end-to-end speech-enhancing ASR pipeline is developed featuring:
  - A programmable accelerator for seq2seq bidirectional RNNs with attention
  - A Markov source separation engine accelerator for speech denoising
- Measurements on test chip show:
  - 18ms end-to-end per-frame latency enabling real-time performance
  - 2.24mJ end-to-end per-frame energy
  - Pipeline obviates very inefficient strategy of scaling up DNN model size to achieve noise robustness

Open-Source Releases

- FlexASR HW architecture and simulator will be publicly released by end of February 2021 at this GitHub repository: https://github.com/harvard-acc/FlexASR
- FlexASR leveraged several SystemC/C++ IPs from MatchLib: https://github.com/NVlabs/matchlib
- Development and verification of this test chip leveraged several hardware IPs and tools from the CHIPKIT framework: https://github.com/whatmough/CHIPKIT

Acknowledgements

- This work is supported in part by JUMP ADA, DARPA CRAFT and DSSoC programs, NSF Awards 1704834 and 1718160, Intel Corp., and Arm Inc.
- We thank B. Khailany, R. Venkatesan, B. Keller, and Y. Shao (Nvidia); and U. Gupta, L. Pentecost, and V. Reddi (Harvard); and S. Garg (Mentor) for helpful discussions.

Thank You