Energy Characterization of Tiny AI Accelerator-Equipped Microcontrollers

Yushan Huang*
Imperial College London
London, UK
yushan.huang21@imperial.ac.uk

Taesik Gong*
UNIST
Ulsan, South Korea
taesik.gong@unist.ac.kr

SiYoung Jang
Nokia Bell Labs
Cambridge, UK
siyoung.jang@nokia-bell-labs.com

Fahim Kawsar
Nokia Bell Labs & University of Glasgow
Cambridge, UK
fahim.kawsar@nokia-bell-labs.com

Chulhong Min
Nokia Bell Labs
Cambridge, UK
chulhong.min@nokia-bell-labs.com

Abstract

Tiny AI accelerators are seamlessly integrated into wearable devices due to their small form factor, enabling human sensing applications to run solely on wearables. However, despite this potential, the energy characterization of these tiny AI accelerators has been hardly studied, which is a key enabler for realizing such applications in our daily lives. In this paper, we present a comprehensive analysis of the energy characterization of ultra-low power microcontrollers using MAX78000 manufactured by Analog Device. We detailed the hardware components and their supported power configurations. We then conducted extensive benchmarks at micro and macro levels. For micro-level benchmarks, we evaluated the power/energy consumption under individual system configuration involved in each operation–sensing, AI inference, computation, memory I/O, and idle. For macro-level benchmarks, we analyzed the impact of system-wide configurations on overall energy consumption of end-to-end application pipelines. Our findings offer valuable insights into energy optimization for wearable systems with on-device and human-centered sensing technologies.

CCS Concepts

• **Hardware** → **Power and energy**; • **Computer systems organization** → **Embedded systems**.

Keywords

Energy, Machine Learning, Tiny AI Accelerator, Microcontrollers

ACM Reference Format:
Yushan Huang, Taesik Gong, SiYoung Jang, Fahim Kawsar, and Chulhong Min. 2024. Energy Characterization of Tiny AI Accelerator-Equipped Microcontrollers. In *International Workshop on Human-Centered Sensing, Networking, and Multi-Device Systems (HumanSys '24), November 4–7, 2024, Hangzhou, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3698388.3699628

*This work was done while the authors were affiliated with Nokia Bell Labs.

HumanSys '24, November 4–7, 2024, Hangzhou, China

ACM ISBN 979-8-4007-1300-2/24/11
https://doi.org/10.1145/3698388.3699628

1 INTRODUCTION

Tiny AI accelerators such as Analog MAX78000 [1], ARM Ethos-U65 [2], and GreenWaves GAP9 [3], have significantly shrunk the physical boundaries of artificial intelligence (AI), bringing AI capabilities closer to us than ever before. Designed to operate on microcontrollers (MCUs), these accelerators feature extremely small form factor-*e.g.*, MAX78000: 8mm×8mm-, thereby being seamlessly integrated into wearable devices. This miniaturization enables AI workloads to run directly on wearables [4], offering exciting opportunities such as reduced latency and enhanced privacy preservation by processing data locally.

With integration into wearable technology, tiny AI accelerators show great potential for advancing human-centered sensing applications. By leveraging various on-body sensors, wearable devices equipped with these AI accelerators can continuously process rich user context in real-time and deliver situational services on the fly without relying on smartphones. For example, an attention alert application can monitor surrounding visual events through smart glasses and provide haptic alerts on a ring. However, this increased computational capability introduces significant challenges, particularly regarding *energy efficiency*. Energy characterization of mobile and embedded accelerators have been studied in previous works [5, 6]. While a few benchmark studies exist for tiny AI accelerators integrated into MCUs [7], they primarily focus on energy benchmarking for different models and lack systematic studies characterizing their energy consumption under different system operational conditions.

We presented a comprehensive energy characterization of MCUs equipped with tiny AI accelerators, aiming to facilitate the development of energy-efficient human-centered sensing systems. In this paper, our study focuses on the MAX78000 board, which integrates an Arm Cortex-M4F core, a RISC-V core, and a convolutional neural network (CNN) accelerator into a single package. To facilitate our analysis, we first detailed the hardware components and their supported power configurations. We then conducted extensive benchmarks and characterizations at both micro and macro levels. At the micro level, we evaluated the impact of each hardware component's system configuration on the energy consumption of

Table 1: System configuration variables involved in the experiment.

	Core Configuration Control		Frequency Management			CNN Accelerator Quadrant Control		
	Processor	Operation Mode	System Clock	System Divider	CNN Divider	Number of Activated Nodes	CNN Node Contiguity	CNN Boost
Sensing	✓	✓	✓	✓				
Inference	✓	✓	✓	✓	✓	✓	✓	✓
Computation	✓	✓	✓	✓				
Memory I/O	✓	✓	✓	✓	✓	✓	✓	✓
Idle	✓		✓	✓				

primitive software operations such as sensing, inference, and computation. At the macro level, we analyzed end-to-end application pipelines to assess the impact of system-wide configurations on overall energy consumption. Our findings provide valuable insights for optimizing energy usage in human-centered sensing and wearable devices equipped with tiny AI accelerators.

2 BACKGROUND

We first describe the experimental hardware platform, followed by a description of the application pipeline, and outline the configurable system parameters.

2.1 MAX78000 Description

Recently, many tiny AI accelerators have been introduced, such as Analog MAX78000 [1], ARM Ethos-U65 [2], and GreenWaves GAP9 [3]. However, most of them are not commercially available or have limited control over their underlying operations. Therefore, we selected Analog MAX78000 as our experimental platform as it offers comprehensive open-source tools and documentation, making it ideal for exploring energy consumption characteristics.

The MAX78000 is an MCU equipped with a CNN accelerator designed for ultra low-power neural networks. It has two processors, an ARM Cortex-M4F and RISC-V, both capable of serving as the main processor, with 512 kB flash and 128 kB SRAM. The CNN accelerator features 4 nodes per group and 4 groups per quadrant, totaling 64 nodes, where each node handles convolution tasks with shared memory for input and activation. This CNN accelerator has dedicated memory: 512 kB input memory, 442 kB of weight memory and 2 kB of bias memory. Figure 1 provides an overview of MAX78000.

2.2 Application Pipeline

The application pipeline on wearables has several operations, especially including sensing and inference, designed to understand human contexts. This pipeline is often executed continuously, with intervals to deliver situational services. To analyze the energy characterization of the MAX78000, we considered a set of common operations for human sensing.

Sensing refers to the process of collecting data from sensors. For instance, this could involve capturing images using a camera or recording sound using a microphone.

Inference is a operation which involves executing the CNN model on the CNN accelerator for tasks like image classification or object detection.

Computation (COMP) refers to operations handled specifically by the processor, such as processing raw data from sensors, performing calculations like Softmax, and some on-processor memory operations.

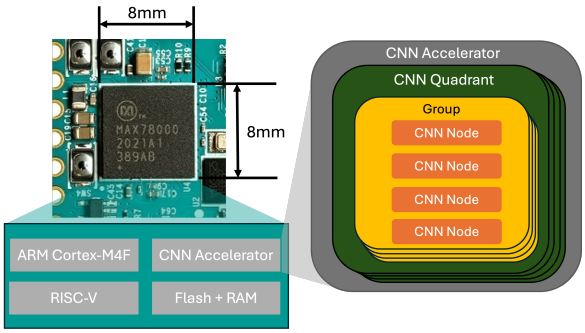

Figure 1: Overview of MAX78000.

Memory I/O (MIO) involves operations between the processor and the CNN accelerator, such as loading model parameters from SRAM to the CNN accelerator or transferring inference results back to SRAM after inference.

Idle refers to waiting periods where the MCU is not actively executing tasks, often entering a low-power state while awaiting the next operation or input.

2.3 Configuration Description

The MAX78000 offers several configurable system parameters that affect energy performance during application execution. We especially focus on three control categories: core configuration control, frequency control, and CNN accelerator quadrant control. Each category is further divided into several subcategories, resulting in a total of eight system configuration variables, as shown in Table 1. The table also indicates which controls are available during each operation of the application pipeline.

Processor. Many MCU devices feature one or more processing units. For instance, MAX78000 is equipped with ARM Cortex-M4F and RISC-V, both of which can operate as main processing unit. However, the architectural differences of the two may result in varying power consumption even under the same workload. Hence, we explored their power and energy consumption differences by selecting a processor type for each operation of the pipeline.

Operation Mode. The MAX78000, like other modern MCUs used in low-power devices (*e.g.*, wearable, IoT, and embedded systems), features several operational modes designed to optimize power consumption. For example, in ACTIVE mode, all system components are fully operational, while SLEEP mode places the ARM or RISC-V in a retention state to reduce power consumption. However, since modes like Low-Power (LPM) and Ultra-Low-Power (UPM) modes affect other configurations such as memory and clocks (see Table 2), we focus on ACTIVE and SLEEP modes only (except 'idle' operation) to accurately capture their impact.

Table 2: MAX78000 operation modes.

Processor Mode	ARM	RISC-V	Oscillators	Memory	CNN Quadrants	CNN RAM	Peripherals
ACTIVE	On	On	All Available	Available	Active, Configurable	Active, Configurable	Available
SLEEP	Retention	On/Retention	All Available	Available	Active, Configurable	Active, Configurable	Available
LPM	Retention	On/Retention	ISO, IBRO ERTCO, INRO	0,1: Retention 2,3: Available	Active, Configurable	Active, Configurable	Available
UPM	Retention	Retention	IBRO, ERTCO INRO	Retention	Optionally off	Selectable Retention	Retention

Clock and Dividers. The operating frequency (*OPE_Freq*) of processors is a critical factor affecting the energy performance of MCUs and is determined by two key parameters: system clock (*SYS_CLK*) and the clock divider (*SYS_DIV*). The relationship between the parameters is as follows:

$$OPE\_Freq = \frac{SYS\_CLK}{2^{SYS\_DIV}}$$

The system clock provides timing signals to the processor, CNN accelerator, peripherals, and other system components, ensuring operations at a correct speed and timing. While MAX78000 supports multiple clock sources as system clocks, we experimented with most commonly used Internal Primary Oscillator (IPO, 100 MHz) and Internal Secondary Oscillator (ISO, 60 MHz) for all operations, as shown Table. 1.

The clock divider serves as a mechanism to scale down the base system clock frequency by dividing it with a specific divider value, thereby controlling the operating speed of various components within the system. In MAX78000, the entire system frequency is controlled via the system divider while CNN accelerator frequency can be further controlled by CNN divider. For these experiments, we configured 1, 2, and 4 (out of 8 choices) for both, as higher divider results in long latency, which is suitable for real-world scenarios.

Number of Activated Nodes. The MAX78000 features a unique configuration that allows developers to select the number of active nodes in the CNN accelerator, which impacts energy performance. Developers can customize the number of active nodes through register settings based on the application's needs. In our experiments, we configured the accelerator to operate with the minimum required number of CNN nodes and with all 64 nodes active.

CNN Node Contiguity. The unique architecture of the CNN accelerator, where nodes are bound by groups and quadrants, presents a valuable opportunity for investigating energy performance. Since data is shared within each group, discontinuous node activation may lead to additional transactions between memory spaces within the CNN accelerator, potentially impacting energy efficiency. Assuming f represents activation, and 0 represents deactivation of a group, for our experiment, we considered continued (*e.g.*, $ff00$) and discontinued (*e.g.*, $f0f0$) node activation.

CNN Boost. The MAX78000 features an external CNN boost circuit designed to provide additional power to the CNN accelerator. While the internal Single-Input Multiple-Output (SIMO) power supply is sufficient under moderate workloads, it may experience voltage drops (brown-out) during transient over-current conditions when peak computing power is required. The CNN boost circuit supplements the SIMO power supply, ensuring stable operation of the CNN accelerator by preventing power-related failures. We conducted experiments to compare the power/energy consumption with/without CNN boost.

3 ENERGY CHARACTERIZATION

We conducted micro and macro levels' experiments. The micro-level experiments provide a detailed investigation of the energy characterization in each operation, helping us identify the optimal system configuration for each operation. The macro-level experiments are used to validate the combined effect of the system configurations on the overall power/energy consumption of the entire pipeline.

In the micro-level experiments, we evaluated individual system configuration involved in each operation, measuring power/energy consumption. We considered two scenario for each operation: sensing, inference, COMP, and MIO. Based on the results of the micro-level experiments, we selected the system configuration variables that had the most significant impact on power/energy, then used them for the macro-level experiments. We added a *Best* system configuration that shows the lowest *energy consumption* for each application, and compared it with the default configuration.

We measured the **average power usage (*P*)** using Monsoon High Voltage Power Monitor [8] with 50 Hz sampling rate and 3.0 V of the input voltage (U). For the **energy consumption (*E*)**, we first measured the average power P, then measured the latency t by the internal clock on the MAX78000, and finally calculated the energy consumption by $E = P \times t$. To obtain stable values, we only utilized the values recorded after running for 1 min. All results are reported as the average of ten experiments.

3.1 Micro Benchmark

For micro benchmark, as Table. 1 shows, each operation is evaluated with consideration of core configuration control, frequency management, and CNN accelerator quadrant control, either partially or entirely. When experimenting with a particular system configuration variable, all other system configuration variables remain at the default settings. The default configuration is: processor = ARM, operation mode = ACTIVE, system clock = IPO, system divider = 1, CNN divider = 1, number of activated nodes = part, CNN node contiguity = No, CNN Boost = No, and using default busy-waiting idle implementation.

Sensing. We considered two scenarios: capturing images with a camera and recording audio with a microphone, both using the integrated sensors. Sensing operation primarily involves core configuration control and frequency management. The results are shown in Fig. 2. For energy consumption, which depends on power and latency, using (a) ARM as the main processor, (b) SLEEP mode, (c)ISO clock, and (d) a system divider of 2 reduces energy cost. Compared

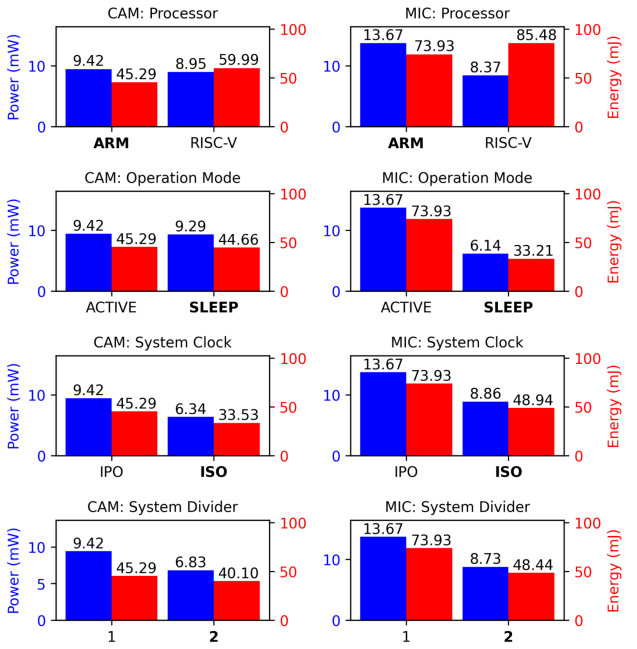

Figure 2: Micro-level evaluation: sensing operation. Bold fonts are more energy-efficient configurations.

to RISC-V as the main processor, ACTIVE mode, IPO clock, and a system divider of 1, these configurations reduced energy by an average of 1.24X, 1.62X, 1.43X, and 1.33X, respectively.

Inference. We considered two models: a 4-layer simple CNN network and a more complex 18-layer Unet network [9] (which includes both encoder and decoder structures). We repeated the simple CNN model inference 1,000 times and the Unet model 100 times. We experimented with control configurations as mentioned in Table 1 for the inference operation. As shown in Figure 3, the results show that the processor type has a minimal impact on power/energy, with only difference under 0.1% on average between ARM and RISC-V. Interestingly, setting the main processor to SLEEP mode during inference operation is more efficient than ACTIVE mode, reducing the power/energy by 7.29% on average. This is because faster system clocks, system dividers, and CNN dividers result in higher power, but much lower latency, thereby resulting in lower energy consumption. For CNN accelerator quadrant control, activating minimum required number of nodes reduced power/energy consumption by 1.68X. However, node contiguity and CNN boost only have minimal reductions. These reductions may stem from the higher data transfer efficiency of continuously activated nodes, as well as the more stable voltage provided by the CNN Boost, which reduces performance fluctuations caused by voltage variations and shortens processing time.

Computation. We considered two cases: Fibonacci calculation and Softmax calculation with 100 network nodes. We calculated the 34th Fibonacci number, and repeated the Softmax calculation 50,000 times. The COMP operation primarily involves core configuration control and frequency management, with the results shown in the Table 3. The results for processor, system clock, and system

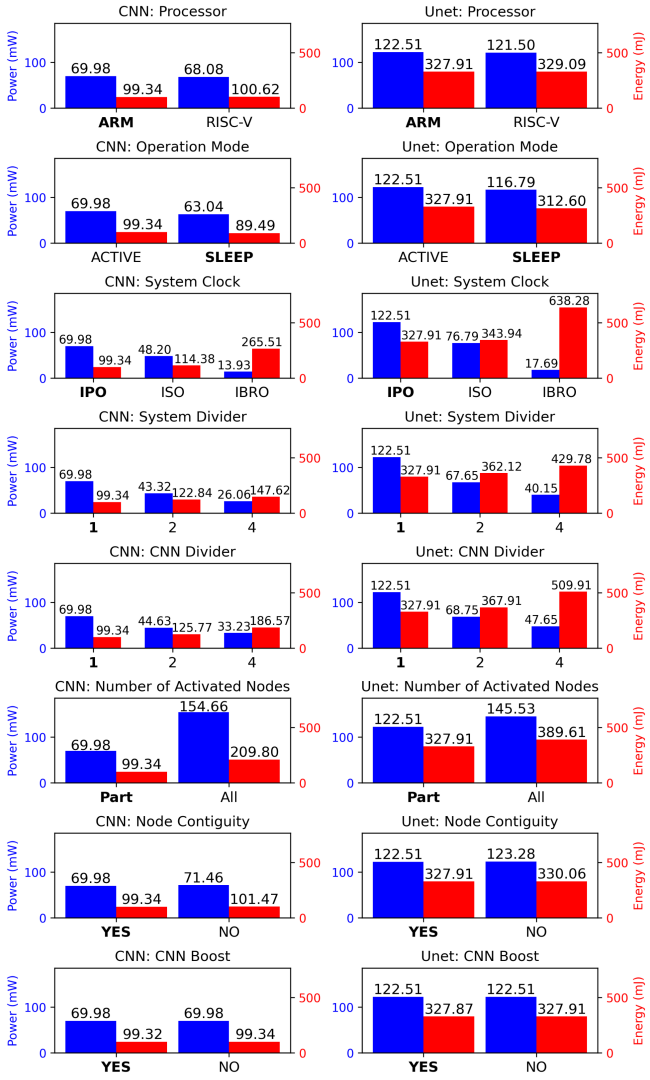

Figure 3: Micro-level evaluation: inference operation.

divider are similar to those in the inference operation. However, for operation mode, since the COMP operation involves tasks running on the main processor, it cannot be put into SLEEP mode. Therefore, we tested the impact of the secondary processor's mode on power/energy consumption while the main processor is executing tasks. As Table 3 shows, when ARM is the main processor, if RISC-V is activated alongside ARM, there is marginal difference in power/energy consumption, with only 4.31% increase. However, when RISC-V is the main processor, the differences in power/energy consumption between ACTIVE and SLEEP is much more significant, increased by 32.56%. These findings suggest that when executing tasks requiring only one processor, it is best to activate only the main processor (ARM has higher power but lower energy consumption, while RISC-V shows the opposite). For tasks involve both processors, whether through cooperation or parallel computing, choosing ARM as the main processor and RISC-V as the secondary processor is more energy-efficient.

Table 3: Micro-level power (mW) and energy (mJ) evaluation: computation operation.

		Fibonacci		Softmax	
		Power	Energy	Power	Energy
Processor	**ARM**	13.96	37.76	14.29	30.83
	RISC-V	8.65	56.06	9.53	64.28
Operation Mode (Main: ARM)	**SLEEP**	13.96	37.76	14.29	30.83
	ACTIVE	14.72	39.82	14.80	31.93
Processor Mode (Main: RISC-V)	**SLEEP**	8.65	56.06	9.53	64.28
	ACTIVE	12.89	83.54	14.06	94.85
System Clock	**IPO**	13.96	37.76	14.29	30.83
	ISO	9.02	40.66	9.09	32.82
System Divider	**1**	13.96	37.76	14.29	30.83
	2	8.83	47.77	8.91	38.45
	4	6.38	68.78	6.42	55.40

Table 4: Micro-level power (mW) and energy (mJ) evaluation: memory I/O operation.

		CNN		Unet	
		Power	Energy	Power	Energy
Processor	**ARM**	26.14	64.13	26.10	70.46
	RISC-V	23.12	90.55	23.69	105.81
Processor Mode (Main: ARM)	**SLEEP**	26.14	64.13	26.10	70.46
	ACTIVE	26.68	65.40	27.05	73.03
Processor Mode (Main: RISC-V)	**SLEEP**	23.12	90.55	23.69	105.81
	ACTIVE	27.90	109.27	27.59	123.23
System Clock	**IPO**	26.14	64.13	26.10	70.46
	ISO	17.28	70.90	18.11	81.84
System Divider	**1**	26.14	64.13	26.10	70.46
	2	16.21	79.42	17.04	92.04
	4	12.01	86.79	11.77	127.14
CNN Divider	**1**	26.14	64.13	26.10	70.46
	2	19.33	64.39	20.37	74.67
	4	15.25	78.27	15.02	84.79
Number of Activated Nodes	**Part**	26.14	64.13	26.10	70.46
	All	26.14	64.13	26.10	70.46
Node Contiguity	**YES**	26.14	64.13	26.10	70.46
	NO	26.14	64.13	26.10	70.49
CNN Boost	**YES**	26.14	64.13	26.10	70.46
	NO	26.14	64.10	26.10	70.44

Memory I/O. We explored a 4-layer CNN and an 18-layer Unet, including their weight, bias, and input loading/unloading. We repeated this operation 500 times for the CNN and 100 times for the Unet. The results are shown in Table 4. The patterns in MIO operations regarding core configuration control and frequency management are similar to those in the inference and COMP operations, and the patterns for node contiguity and CNN boost are also similar to those in the inference operations, thus we omit the analysis. One notable difference is that, unlike the inference operation, the 'Number of Activated Nodes' has almost no impact in the MIO operation. This is because, even when all nodes are activated, the number of nodes involved remains fixed, resulting in no noticeable impact on power and energy consumption.

Idle. We first measured the power consumption of the default busy-waiting implementation. As Table 5 shows, using RISC-V as the main processor, along with slower system clocks and dividers, results in lower power consumption. However, busy-waiting keeps all components active for continuous signal-waiting, causing high energy usage. Table 2 lists MAX78000's power-saving modes. SLEEP, LPM, and UPM modes offers deeper sleep states, reducing the number of active components accordingly. We evaluated two wake-up mechanisms: the Wake-Up Timer (WUT) and the Low-Power Timer (LPT), since both of which support waking up the MAX78000 from the above non-ACTIVE states. Table 6 shows the power consumption for different idle implementations. The results indicate that UPM mode provides the lowest power consumption among all implementations, at only 2.02 mW, as it activates the fewest system components. Moreover, regardless of whether UPM mode is entered via ARM or RISC-V or is woken up by WUT or LPT, the power consumption remains approximately 2.02 mW.

Table 5: Micro level power (mW) evaluation: idle phrase

Processor		System Clock		System Divider		
ARM	RISC-V	IPO	ISO	1	2	4
13.11	10.76	13.11	8.60	13.11	8.42	6.19

Table 6: The power (mW) of idle implementations

	Default	Wake-Up Source	SLEEP	LPM	**UPM**
ARM	13.11	WUT	6.11	5.33	2.02
		LPT	6.11	5.33	2.02
RISC-V	10.76	WUT	6.41	5.42	2.02
		LPT	6.41	5.42	2.02

Table 7: The 'Best' configuration for working energy.

	Processor	Processor Mode	System Clock	System Divider
Sensing	ARM	SLEEP	ISO	2
Comp_1	ARM	ACTIVE	IPO	1
MIO_1	ARM	ACTIVE	IPO	1
Inference	ARM	SLEEP	IPO	1
MIO_2	ARM	ACTIVE	IPO	1
Comp_2	ARM	ACTIVE	IPO	1
Idle	ARM	UPM	ISO	2
Part CNN Nodes, CNN Divider = 1, WUT_UPM				

3.2 Macro Benchmark

After conducting a micro-level analysis of each operation, we proceeded to a macro-level analysis using an end-to-end speech recognition application (using KWS20 model [10]). We assume each pipeline execution cycle lasts 2.5 seconds (total time), with the idle time determined by the total duration of the operations in an application. Because under the default system configuration, the working-to-idle time ratio is about 1:3, closely matching real-world conditions.

Based on the results of micro-level experiments, we selected the system configuration variables that have significant impacts: processor (ARM or RISC-V), operation mode (ACTIVE or SLEEP), system clock (IPO or ISO), system divider (1 or 2), CNN divider (1 or 2), number of activated nodes (part or all), and idle implementation. We used the pre-determined default configuration of MAX78000 as a baseline: processor = ARM, operation mode = ACTIVE, system clock = IPO, system divider = 1, CNN divider = 1, number of activated nodes = all, and using default busy-waiting idle implementation. The default configuration is maintained throughout all operations. We also designed a configuration that results in the lowest working

Table 8: The time (T), power (P) and energy (E) evaluation for the speech recognition experiment.

	Default			Best		
	T (us)	P (mW)	E (mJ)	T (us)	P (mW)	E (mJ)
Sensing	398759	13.69	5.459	429413	3.68	1.580
COMP_1	46018	12.83	0.590	46018	12.83	0.590
MIO_1	132435	25.27	3.347	132435	25.27	3.347
Inference	14631	106.73	1.562	14631	98.81	1.446
MIO_2	68	10.77	0.001	68	10.77	0.001
COMP_2	1292	16.73	0.022	1292	16.73	0.022
Idle	1906797	15.60	29.746	1876143	2.02	3.790
Total Energy	-	-	40.726	-	-	**10.775**
Working Energy	-	-	10.980	-	-	**6.985**

energy consumption during the execution of an application pipeline, labeled as 'Best', the configuration is shown in Table 7.

The results in Table 8 show that the proportion of energy consumption is different across the operations of the pipeline. For instance, in the operations of idle, sensing, MIO, and inference account for a higher percentage, indicating that particular attention should be paid to optimizing energy consumption in practical applications. Compared the 'Best' with the 'Default' configurations, the main differences between them are: the system clock and system divider during the sensing and idle operations, the operation mode during the sensing, inference and idle operations, as well as the idle implementation. These differences indicate that the default configuration of the MAX78000 only optimizes parts of the operations, including COMP_1, MIO_1, inference, MIO_2, and COMP_2, but does not consider the entire end-to-end process, neglecting the sensing and idle operations and the optimal configuration of the operation mode. In addition, simply applying the static system configuration to the entire pipeline is not energy-efficient, as the optimal energy-efficient configuration differs for each operation. The 'Best' configuration takes into account every operation in the pipeline, and dynamically adjusts the system configuration for each operation to minimize energy consumption. Compared to the 'Default' configuration, the 'Best' configuration reduces total energy by 3.78X and working energy by 1.57X. In addition, in terms of latency, the 'Best' configuration is only slightly slower than the default configuration during the sensing operation. For application scenarios that involve waiting periods (*e.g.*, when task execution is intermittent), the slight increase in sensing operation latency has a negligible impact.

4 LIMITATIONS AND FUTURE WORKS

In this section, we discuss some limitations of this study, as well as the future works.

Limited experimental MCUs. In this study, we have conducted experiments on the MAX78000. Other MCUs, such as ARM Ethos-U65 [2] and GreenWaves GAP9 [3], still require further exploration in future work. However, we believe that the findings of this study are general to other MCUs. This is because the energy characterization observed in MAX78000 tend to be influenced by fundamental factors such as workload characteristics, data flow, and hardware architecture—elements that are similar across a range of MCUs.

Limited experimental scenarios/cases. In the micro-level experiments, we have considered two scenarios for each operation, while in the macro-level experiments, we have used speech recognition as an example. Nevertheless, we do not cover all scenarios and sensor types (*e.g.*, accelerometers). In future work, we plan to further explore the energy characterization of AI accelerators across a broader range of application scenarios and sensors.

System and algorithm co-design to reduce energy. In this study, we primarily analyzed energy characterization from a system perspective and observed that there is a trade-off between energy, latency, and power. Balancing this trade-off requires considering the specific scenario, as each scenario may have particular requirements for energy, latency, and power. In such cases, these requirements, along with the trade-off, present an optimization problem. In the future, we plan to co-design system and algorithm to achieve the optimal balance of energy, latency, and power.

5 CONCLUSION

In this paper, we conducted a variety of benchmarks to analyze the energy characterization of the ultra-low-power DNN accelerator, MAX78000. First, we performed a micro-level analysis by analyzing each operation of the pipeline independently, followed by a macro level analysis, treating the pipeline as a whole. The system configurations cover three aspects and eight variables, where we evaluated the impact of these variables on power and energy consumption. Additionally, we discovered that the most energy-efficient system configuration is dynamic for each operation and thus proposed the most energy-efficient configuration for a speech recognition example. Beyond these data and findings, our benchmark study offers valuable insights for the development of wearable devices equipped with tiny AI accelerators and efficient human-centered sensing technologies.

References

[1] Maxim Integrated. Max78000. 2024. https://www.analog.com/en.

[2] ARM. Ethos-u65. 2024. https://www.arm.com/products/silicon-ip-cpu/ethos/ethos-u65.

[3] Greenwaves. Ultra low power gap processors. 2024. https://greenwaves-technologies.com/gap9_processor/.

[4] Taesik Gong, Si Young Jang, Utku Günay Acer, Fahim Kawsar, and Chulhong Min. Synergy: Towards on-body ai via tiny ai accelerator collaboration on wearables, 2024.

[5] Mattia Antonini, Tran Huy Vu, Chulhong Min, Alessandro Montanari, Akhil Mathur, and Fahim Kawsar. Resource characterisation of personal-scale sensing models on edge accelerators. In *Proceedings of the First International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things*, pages 49–55, 2019.

[6] Nicholas D Lane, Sourav Bhattacharya, Petko Georgiev, Claudio Forlivesi, and Fahim Kawsar. An early resource characterization of deep learning on wearables, smartphones and internet-of-things devices. In *Proceedings of the 2015 international workshop on internet of things towards applications*, pages 7–12, 2015.

[7] Arthur Moss, Hyunjong Lee, Lei Xun, Chulhong Min, Fahim Kawsar, and Alessandro Montanari. Ultra-low power dnn accelerators for iot: Resource characterization of the max78000. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pages 934–940, 2022.

[8] Monsoon Solutions Inc. Monsoon high voltage power monitor. 2024. https://www.msoon.com/.

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[10] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.