A RRAM-based High Energy-efficient Accelerator Supporting Multi-modal Tasks for Virtual Reality Wearable Devices

Xin Zhao, Zhicheng Hu, Zilong Guo, Haodong Fan, Xi Yang, Jing Zhou and Liang Chang
University of Electronic Science and Technology of China
Chengdu, China

ABSTRACT

Virtual reality (VR) wearable devices can achieve immersive entertainment by fusing multi-modal tasks from various senses. However, constrained by the short battery life and limited hardware resources of the VR devices, running multiple tasks simultaneously with different modals is difficult. In this paper, we propose an energy-efficient accelerator that supports Multi-modal Tasks for VR devices, namely MTVR. We present a multi-task computing solution based on the flexible multi-task computing core design and efficient computing unit allocation strategy, which simultaneously achieves efficient work of multi-modal tasks. We design an early exit detector to skip invalid calculations, greatly saving energy. In addition, a fine-grained tiny value skip method at multiplier and adder levels is proposed to save energy further. We provide a hybrid RRAM and SRAM memory access scheme, reducing the external memory access (EMA). Through experimental evaluation, the multi-task computing core achieves an average computational utilization of 95%. When the invalid input ratio is 90%, energy saving brought by the early exit detector can reach 88%. The tiny value skip method further achieved 13% energy saving. Hybrid memory access scheme obtains 98.9% EMA reduction. We deployed the MTVR accelerator in FPGA and self-designed RRAM, achieving energy efficiency of 3.6 TOPS/W, higher than other single-task accelerators.

1 INTRODUCTION

Virtual reality (VR) wearable devices can achieve immersive experiences by mobilizing multiple senses [1]. For example, in terms of vision, clearer visual effects can be brought by super-resolution (SR) [2–4]. Keyword spotting (KWS) can realize human-computer interaction by voice commands at speech senses [5, 6]. At physiological signal perception, the electrocardiogram (ECG) can monitor the user's heartbeat during immersive gaming entertainment to ensure safety [7]. Combining SR, KWS, and ECG tasks from different modals can obtain a better immersive experience. However, limited by the battery life and hardware resources of edge-end VR wearable devices, it has become a vital challenge to efficiently support the simultaneous operation of multiple tasks in different modals to achieve an immersive experience.

To address the above issues, on the one hand, the domain-specific chip can support one designated application with good performance. By cascading several domain-specific chips, multiple tasks can be achieved simultaneously but results in high device cost and power consumption. On the other hand, adopting a single general-purpose chip, such as a CPU, to run different tasks sequentially via the time division multiplexing principle can reduce the device cost, but the work efficiency is low and it is difficult to meet the real-time requirements. Combining the excellent performance of domain-specific chips with the cost and power advantages of a single chip is a potential solution for VR wearable devices. However, no previous work considered efficiently running multiple different tasks simultaneously on a single chip for VR wearable devices.

In this work, we propose a high energy-efficient accelerator for VR wearable devices, called MTVR, which can simultaneously support multiple tasks in different modals, including SR, KWS, and ECG tasks. The specific contributions are listed as follows:

- We present a multi-task computing solution. Through flexible multi-task computing core design and efficient computing unit allocation strategy, multiple tasks can work efficiently in the computing core simultaneously. In addition, high-performance and low-power working modes are supported based on actual requirements. Furthermore, a fine-grained tiny value computing skip method is provided to further save energy consumption.
- We design a multi-task early exit mechanism. For different tasks, the lightweight early exit detector can directly exit calculations of preprocessing and neural network inference based on invalid input and current state, which greatly saves energy consumption in practical scenarios.
- We provide a hybrid RRAM and SRAM memory access scheme. On the one hand, RRAM is adopted to store the read-only parameters. On the other hand, for the output feature maps generated during the inference, caching them both in SRAM and RRAM can reduce external memory access a lot. In addition, benefitting from the endurer design, RRAM can maintain a longer lifetime.
- We have designed the multi-task accelerator MTVR and deployed it in FPGA and self-designed RRAM to verify the feasibility of the proposed architecture.

2 PRELIMINARY

In this work, we mainly optimize and deploy three *multi-modal tasks* (hereafter abbreviated as *multi-task*) on VR devices: SR, KWS, and ECG tasks. **SR** is an image reconstruction technique that can achieve mapping from low-resolution images to high-resolution ones [8]. In VR wearable devices, SR is adopted for image quality improvement and thus enhanced experience. Two metrics are employed to evaluate the SR, where peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) are used to evaluate image reconstruction quality, and frame rate (fps) is used to evaluate reconstruction speed. **KWS** is a speech recognition task employed in VR devices for human-computer interaction by speech commands. **ECG** is used for health monitoring in VR devices. The immersive entertainment will be stopped and an alarm will be issued when an abnormal heartbeat is detected. Accuracy is an important metric for both KWS and ECG tasks. In addition, the network scale of the SR task is much larger than that of KWS and ECG tasks, which bottlenecks the computation and memory access.

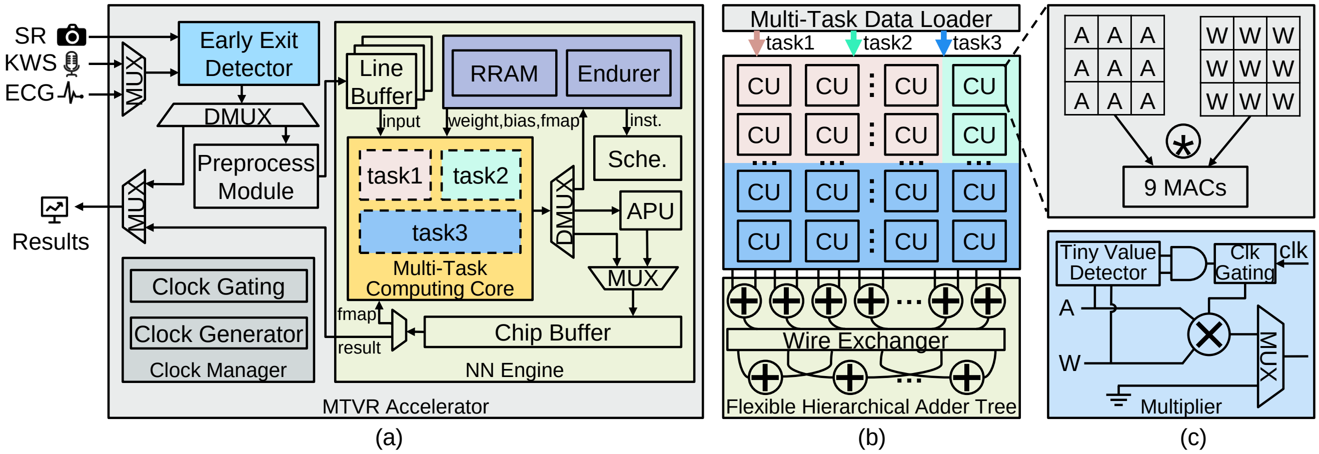


Figure 1: MTVR architecture. (a) Overall MTVR architecture, where Sche. indicates Scheduler. (b) Multi-task computing core, consisting of the multi-task data loader, CU array, and flexible hierarchical adder tree. (c) CU and multiplier.

3 MTVR ARCHITECTURE

3.1 MTVR Architecture Overview

Figure 1(a) shows the proposed multi-task architecture MTVR, which mainly consists of four parts: early exit detector, preprocess module, NN engine, and clock manager. The early exit detector takes the inputs of SR, KWS, and ECG tasks to determine whether the subsequent preprocessing and inference will continue or exit directly (in Section 3.3) and switches clocks of the preprocess module and NN engine through the clock gating module in the clock manager. The preprocessing module and NN engine complete the preprocessing and neural network inference for multiple tasks. The clock manager supports the dynamic switching of clocks for different modules and the generation of various clocks.

3.1.1 Preprocess Module. It supports the preprocessing of SR, KWS, and ECG tasks. The SR task completes the bilinear interpolation of Cb and Cr channels in the preprocessing module, and the Y channel is fed into the NN engine for more accurate computation. In the KWS task, speech signal feature (FBank) extraction is carried out, including pre-emphasis, framing, windowing, FFT, Mel filtering, etc. For the ECG task, preprocessing operations mainly contain denoising, R-peak detection, and signal segmentation.

3.1.2 NN Engine. The NN Engine implements the neural networks inference, consisting of the line buffer group, RRAM with embedded endurer, chip buffer, scheduler, multi-task computing core, and APU (Auxiliary Processing Unit). For memory, the line buffer group stores input feature maps, and each line buffer stores one row of feature map data. For 1D data inputs in KWS and ECG, only one line buffer is needed to complete input data sliding and updating. For 2D data input in SR, cascading multiple line buffers based on kernel size can complete 2D data sliding and updating. In addition, the line buffer adopts the dual port design, which simultaneously receives preprocessed input data and sends it to the multi-task computing core, enabling fully pipelined dynamic data updates and thus reducing latency caused by data updates. As for RRAM, it stores read-only data such as weights and instructions. In addition, it also works with the chip buffer to cache the generated output feature maps during the inference (in Section 3.4). The endurer is designed to improve the lifetime of RRAM. For the calculation part, SR, KWS, and ECG tasks are simultaneously calculated in the multi-task computing core (in Section 3.2). In addition, nonlinear operations such as activation and pooling are achieved in APU. The scheduler completes layer-wise control of calculations and memory access by executing instructions from RRAM.

3.1.3 Calculation Process. The original inputs of SR, KWS, and ECG tasks are first transmitted to the early exit detector. Due to the lower KWS and ECG sampling rate than the SR, a shared input channel is adopted to save input bandwidth. The early exit detector determines whether the input is valid. If it is invalid, it exits directly to end the current calculation. On the contrary, continue with subsequent preprocessing and inference. For valid inputs, the preprocessed input data is sent to the line buffer group. The multi-task computing core receives the input and weight from the line buffer group and RRAM, respectively. The scheduler reads instructions from RRAM and starts the calculation simultaneously. The output feature maps of each layer are temporarily stored in the chip buffer or RRAM. Output results until finishing the calculations.

3.2 Multi-task Computing Core

Figure 1(b) shows the structure of the multi-task computing core, consisting of a multi-task data loader, CU array, and flexible hierarchical adder tree. The multi-task data loader is a router with multiple switches that can deliver input data to a specified CU (computing unit). Based on the network scale of SR, KWS, and ECG tasks, the CU array with the size of 14×14 is divided into three sub-arrays, independently completing the inference of their respective tasks. The flexible hierarchical adder tree is composed of an inverted pyramid-shaped 8-level adder cluster, each level of which contains ($14^2/2^n$) adders. Adjacent two-level adder clusters are connected through a wire exchanger, allowing any combination between the front and rear adders. According to the different allocations of CU sub-arrays, the flexible hierarchical adder tree can be quickly combined into a matching sub-adder tree topology structure through the wire exchanger.

3.2.1 CU. Figure 1(c) indicates that a CU can complete 9 MAC operations, and different types of calculation can be supported

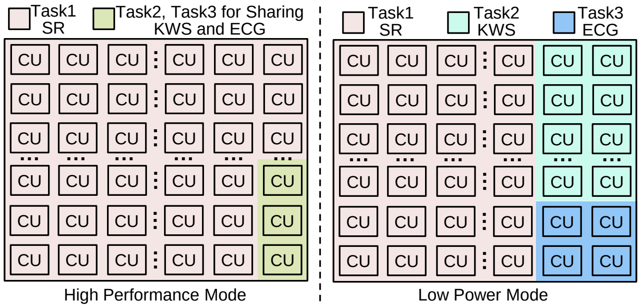


Figure 2: Dual work mode. MTVR can work in the high-performance mode and low-power mode.

based on the input routing of the multi-task data loader. For 2D convolutions, one CU can complete one 3×3 convolution or nine 1×1 convolutions, with 100% computational utilization. When the kernel size is greater than 3×3, combining multiple CUs completes large-scale convolution. For example, a 5×5 convolution can be achieved by adopting 3 CUs, with a 92.6% computational utilization. Similarly, multiple 1D convolutions are completed when the kernel length is less than 9. On the contrary, combining multiple CUs completes the larger-length convolution. For element-wise multiplication, the calculation is achieved by skipping adders in CUs. For unused adders, clock gating turns corresponding clocks off for energy saving.

Due to the redundancy of neural networks, the multiplication of tiny values can be skipped with negligible loss. Figure 1(c) shows that we add a tiny value detector in the multiplier. When one of the input values is smaller than the preset threshold, the multiplication is skipped, and the clock gating unit turns off the clock, reducing computational energy consumption. The specific value of the threshold is determined during network training. Adders adopt the same detection mechanism.

3.2.2 Dual Work Mode. According to different allocations of the CU array, MTVR can work in high-performance or low-power modes to meet the needs of practical scenarios, as shown in Figure 2. For high-performance mode, achieve higher performance by maximizing CU computational utilization. Observing that the network scale of ECG and KWS tasks is much smaller than that of SR task, which can make KWS and ECG share the same CUs by time division multiplexing method to improve the busy ratio of CUs. In addition, the sampling rates of ECG and KWS further limit their throughput. Network scale and sampling rate are considered for optimal allocation of CU sub-arrays. For low-power mode, we try to lower the frequency as much as possible while meeting the real-time performance of SR tasks and the actual minimum detection requirements of KWS and ECG (i.e. inference times per second), thereby reducing computational power consumption.

3.2.3 CU Allocation. Algorithm 1 illustrates the CU allocation strategy for the optimal CU sub-array division in both low-power and high-performance modes. The input is network structure, work mode, and sampling rate scope of SR, KWS, and ECG tasks, where K_1, K_2, C_{in}, C_{out} indicate kernel size, input channel, and output channel. The output is the optimal CU allocation scheme. The procedure can be divided into three steps. ❶ For each layer of the network, we first determine the CU allocation in the kernel dimension to achieve the maximum computational utilization (KerUti). Then, further calculate the input parallelism P_{in} and output parallelism P_{out} in the C_{in} and C_{out} dimensions (ChUti). For ease of expression, the P_{in} and P_{out} set is represented by G. ❷ For different work modes, constraints are determined based on the sampling rate and network scale for subsequent optimal search (Constrain). ❸ Based on the above conditions, we conduct layer search and task search to determine the optimal allocation of CU sub-arrays $(\xi, G)^{task\_best}_{layer\_best}$ for each task (LayerSearch, TaskSearch).

Algorithm 1: CU Allocation

```
Input: NetStructure($K_1, K_2, C_{in}, C_{out}$), WorkMode,
       SampleRate
Output: Optimal CU allocation $(\xi, G)^{task\_best}_{layer\_best}$
$\xi, P_{out}, P_{in} \leftarrow N^*$;
for $task \leftarrow 1$ to 3 do
    for $layer \leftarrow 1$ to $n$ do                                // Step 1
        $\xi^{task}_{layer} \leftarrow \text{KerUti}_{max}(K_1, K_2)$;
        $[P_{out}, P_{in}]^{task}_{layer} \leftarrow \text{ChUti}_{max}(C_{in}, C_{out}, \xi)$;
        $G^{task}_{layer} \leftarrow [P_{out}, P_{in}]^{task}_{layer}$;
    end
    NetScale ← Compute(NetStructure)                                // Step 2
    if WorkMode == high performance then
        $Ct \leftarrow$ Constrain(SampleRate$_{max}$, NetScale);
    end
    else if WorkMode == low power then
        $Ct \leftarrow$ Constrain(SampleRate$_{min}$, NetScale);
    end
                                                                    // Step 3
    $(\xi, G)^{task}_{layer\_best} \leftarrow \text{LayerSearch}(Ct, [\xi, G]^{task}_{layer})$;
end
$(\xi, G)^{task\_best}_{layer\_best} \leftarrow \text{TaskSearch}([\xi, G]^{task}_{layer\_best})$;
```

3.3 Early Exit Detector

As shown in Figure 3(a), the input of SR, KWS, and ECG tasks is first transmitted to the early exit detector to determine whether the input is valid. If it is invalid, directly exit the subsequent calculation. Figure 3(b) shows the improved EAR-based (Eye Aspect Ratio) early exit detection scheme for the SR task. Detecting the human eyes in the open or closed state determines whether to continue subsequent SR calculation. When the eye is closed, it directly exits the inference. Equation (1) shows the original EAR computing, and we simplified the calculation as shown in Equation (2). Distance calculation can be simplified as differences in horizontal and vertical coordinates because the human eye is stationary relative to the VR device after wearing it. Specifically, we first calculate the vertical coordinate differences (i.e. $(y_{P_2} - y_{P_6})$ and $(y_{P_3} - y_{P_5})$) and the horizontal coordinate differences (i.e. $(x_{P_1} - x_{P_4})$). Then, shift, addition, and multiplication operations are carried out. Finally, a comparison is executed to determine the eye state. Figure 3(b) illustrates the detailed circuit. Complex calculations such as square root

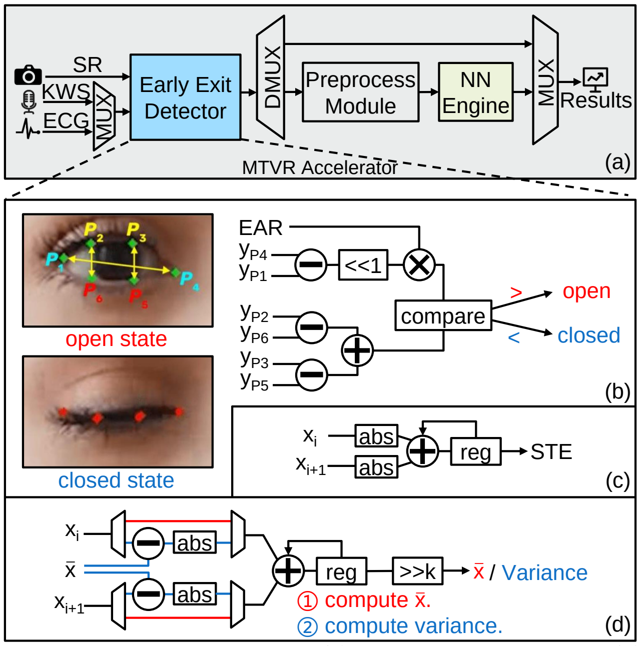


Figure 3: Early exit detector. (a) Early exit mechanism. (b) SR early exit circuit. (c) KWS early exit circuit. (d) ECG early exit circuit.

and division are eliminated after optimizations while maintaining the same detection effect.

$$EAR = \frac{P_2P_6 + P_3P_5}{2P_1P_4} = \frac{||P_2 - P_6|| + ||P_3 - P_5||}{2\,||P_1 - P_4||} \tag{1}$$

$$\{EAR \times 2(x_{P_1} - x_{P_4})\} > \{(y_{P_2} - y_{P_6}) + (y_{P_3} - y_{P_5})\} \tag{2}$$

Figure 3(c) demonstrates the optimized STE-based detection scheme for the KWS task by detecting the short-term energy (STE) of the input speech signal to determine whether the sampled speech is valid. The STE scheme first calculates the sum of input squares for multiple frames and then compares it with the threshold to determine whether the input is valid. To reduce computational overhead, we transformed square calculation into absolute value calculation and achieved the same detection effect, as shown in Equation (3). In addition, there is a half overlap area between frames, so we only calculate the nonoverlapping parts, which can further reduce the calculation by half. After optimizations, only the sum of absolute value for half-frames is needed.

$$STE = \sum_{i=1}^{N}(x_i)^2 \approx \sum_{i=1}^{N}\left|x_i'\right| \tag{3}$$

We provide an approximate variance-based early exit detection scheme for the ECG task, as shown in Figure 3(d). The variance remains small for continuous normal heartbeat values, and when there is an abnormal heartbeat, the variance changes significantly. Therefore, variance can be used to determine the subsequent calculations. As shown in Equation (4), we divide each N heartbeat data into a group and calculate its approximate variance, where N is a power exponent of 2, which can convert division calculation into shift operation; Square calculation is simplified to the calculation of absolute value. Figure 3(d) shows the detailed circuit, due to the calculation of addition and shift involved in achieving both the mean and variance, these circuits are shared to reduce area overhead. The optimal thresholds for the above three schemes are determined through network training.

$$Variance = \frac{\sum_{i=1}^{N}(x_i - \overline{x})^2}{N} \approx \frac{\sum_{i=1}^{N}|x_i - \overline{x}|}{N} \quad (N = 2^k) \tag{4}$$

3.4 Hybrid RRAM and SRAM Memory Access Scheme

To reduce EMA, SR, KWS, and ECG task weights can be stored in the on-chip memory, and RRAM is a suitable solution due to its high memory density. Therefore, we embed RRAM into the MTVR architecture to store read-only parameters such as weights and instructions. In addition, the nonvolatile feature of RRAM avoids parameters updating from off-chip memory to on-chip RRAM before each power-on inference, reducing EMA and improving initialization speed.

Figure 4(b) shows the hybrid memory access between RRAM and SRAM chip buffer with the multi-task computing core. For different tasks, the generated output feature maps during the inference require on-chip buffers with various capacity requirements for storage. However, due to the low-density feature of SRAM and the limited hardware resources of VR wearable devices, it is difficult to integrate a large-capacity SRAM on the chip to meet the requirements of various scale tasks. Traditional solutions temporarily store partial output feature maps to off-chip memory, but EMA bottlenecks system performance.

Based on this issue, we present a hybrid SRAM chip buffer and RRAM memory access scheme to achieve all on-chip inference. Specifically, it can be divided into two situations, as shown in Figure 4(a). ❶ For large-sized output feature maps, they are divided into multiple tiles based on the memory bandwidth, and then the SRAM chip buffer priority principle is adopted to temporarily store as many tiles as possible in the chip buffer, while the remaining tiles are temporarily stored to the RRAM. ❷ For multi-branch structures, based on the priority of feature map usage order, the first used output feature map is temporarily stored in the SRAM chip buffer, and the later used output feature maps are temporarily stored in the RRAM. The above methods can greatly reduce EMA and improve

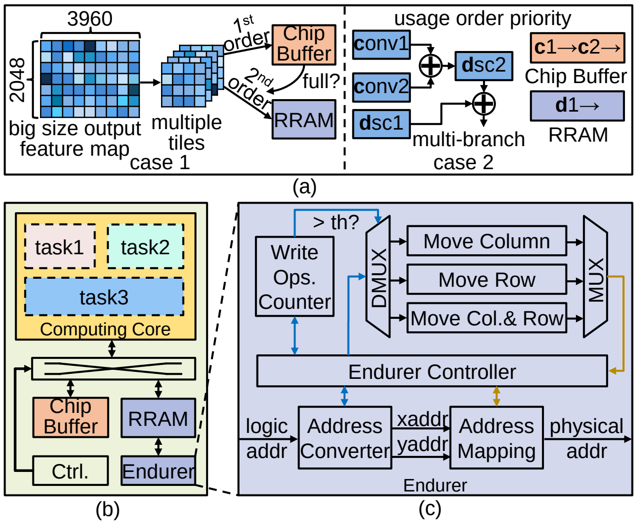


Figure 4: Hybrid RRAM and SRAM memory access. (a) Two cases for hybrid access. (b) Hybrid access scheme. (c) Endurer.

RRAM utilization. However, due to the low endurance of RRAM, we propose an endurer design, as shown in Figure 4(c), which improves the lifetime of RRAM 2.37 times by balancing address movement. Specifically, the write operation counter calculates RRAM write times when it reaches the threshold, activating the address movement and selecting a movement method, where the data on frequent write addresses is moved to other addresses to balance the write frequency of each address. Then, the address mapping unit completes mapping the logic address to the physical address.

4 EXPERIMENT AND EVALUATION

4.1 Experiment Setup

The proposed MTVR architecture is designed based on Verilog HDL, where 12 Mb RRAM uses the IP provided by TSMC, and SRAM is generated based on the ARM Memory Compiler. The power and area metrics are synthesized under the Design Compiler at the TSMC 22nm process technology. We design an in-house simulator to evaluate the MTVR architecture.

4.2 Utilization Analysis

Based on the proposed multi-task computing core and efficient CU allocation strategy, SR, KWS, and ECG tasks can work with high computational utilization, as shown in Figure 5. In high-performance mode, all three tasks have the highest computational utilization, and the utilization of the SR task can reach 98.1%. The average computing utilization of all tasks reaches 95% and 94% in high-performance and low-power modes, respectively.

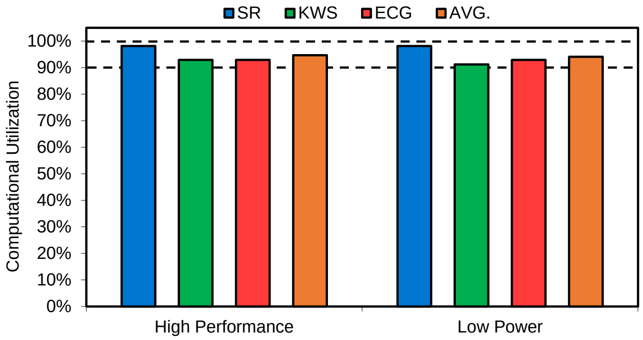


Figure 5: Computational utilization in high-performance mode and low-power mode, respectively.

4.3 Energy Analysis

Figure 6(a) shows that the energy saving brought by the tiny value skip mechanism is 13%. The early exit scheme can greatly save energy consumption by skipping preprocessing and neural network inference. Figure 6(b) tests the energy savings obtained by different ratios of invalid inputs in all inputs. When invalid inputs account for 90%, it can bring 88% energy savings.

4.4 EMA Analysis

Through the hybrid RRAM and SRAM memory access, achieving all on-chip storage during inference for different tasks is possible, greatly reducing EMA. Figure 7 shows the EMA reduction in different SR-based works. Our proposed hybrid memory access scheme can achieve a 98.9% EMA reduction, except for the inevitable EMA caused by the original input and the final calculation results output. And it's still 3.1% higher than the SOTA work, indicating the effectiveness of the proposed memory access scheme.

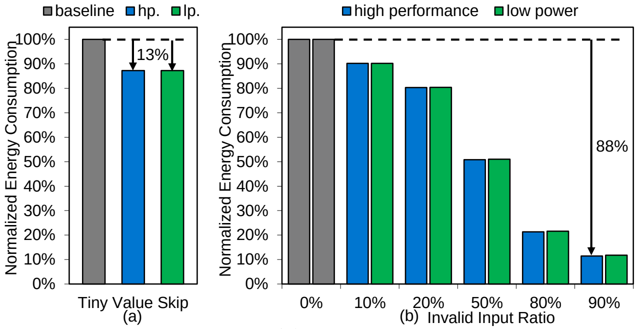


Figure 6: Energy saving. (a) Energy consumption for tiny value skip. (b) Energy consumption for early exit.

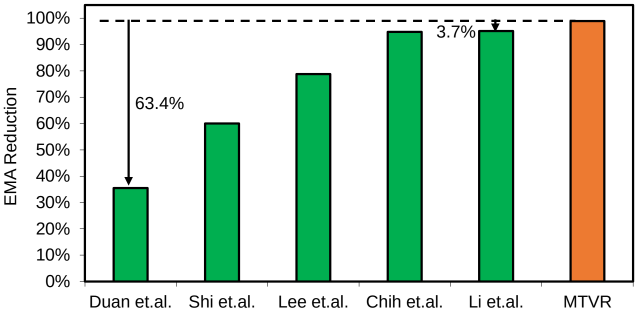


Figure 7: EMA reduction comparison. The other five works are from references [3], [10], [4], [11] and [2], respectively.

4.5 Architecture Comparison

Table 1 shows the comparison with other works. Due to the absence of a multi-task architecture based on SR, KWS, and ECG tasks, we selected the relevant single-task work. For KWS tasks, our architecture MTVR has an accuracy of 90.32%, which is higher than other works. For ECG tasks, MTVR accuracy is higher than TDPRO [9]. For SR tasks, under the Set5 dataset, we have the highest PSNR and SSIM, that is 37.44dB and 0.9592, respectively. In addition, in high-performance mode, MTVR achieved the highest frame rate of 68.9 fps. MTVR can support different working modes, with a power consumption of only 40.5mW in low-power mode, greatly extending running time while meeting real-time requirements. For system energy efficiency, our proposed MTVR has the highest energy efficiency of 3595.9GOPS/W, demonstrating the efficiency of our proposed architecture.

4.6 System Verification

Figure 8 demonstrates the system verification to validate the MTVR architecture. The MTVR architecture, except for RRAM, is deployed on the FPGA, and the RRAM part is deployed separately in the RRAM chip we designed. Our designed RRAM chip has a read speed of 8 Gb/s, and the lifetime improves 2.37 times, benefiting from the endurer design. In the next work step, we will perform complete system tape-out verification on the MTVR architecture.

4.7 Scalability Analysis

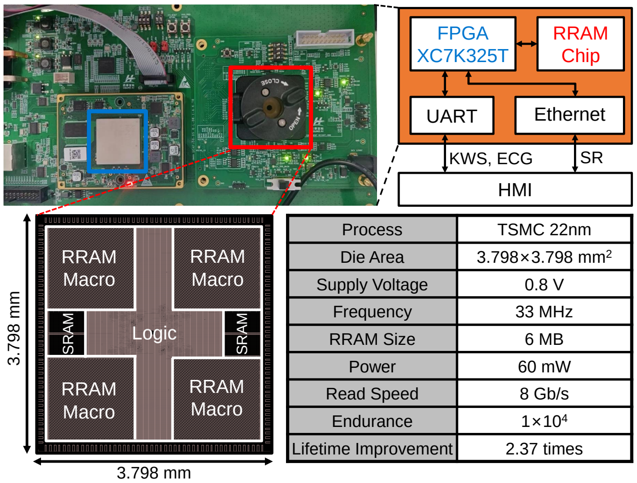
This work mainly considered three modal tasks: SR, KWS, and ECG. Due to the flexible feature of the multi-task computing core and allocation strategy, the MTVR can easily deploy other tasks,

Table 1: Comparison of different hardware architecture

Works		Single Task						Multi Task	
		KWS-based		ECG-based		SR-based		MTVR (Proposed)	
		Guo et.al. [6]	Chiang et.al. [5]	BioAIP [7]	TDPRO [9]	SRNPU [4]	Uarch [3]	High Performance	Low Power
Technology (nm)		65	28	65	55	65	28	22	
Supply Voltage (V)		0.90	0.90	0.75	1.00	1.10	0.81	0.81	0.72
Frequency (MHz)		75	1	1	1	200	156	200	90
Area (mm^2)		6.2	1	1.74	N/A	16	7	2.78	
SRAM Size (KB)		8	28	73	7.5	572	777	117	
Dataset		GSCD	GSCD	MIT-BIH	MIT-BIH	Set5	Kvasir$^{(1)}$	GSCD (KWS) MIT-BIH (ECG) Set5 (SR)	
KWS ECG	No. of Classes	10	10	5	5	-	-	10 (KWS), 5 (ECG)	
	Acuuracy	90.20%	89.76%	99.16%	98.60%	-	-	90.32% (KWS) 98.80% (ECG)	
SR	PSNR (dB)	-	-	-	-	37.06	42.68	37.44	
	SSIM	-	-	-	-	0.9565	0.9699	0.9592	
	Frame Rate (fps)	-	-	-	-	31.8	60.0	68.9	31.0
Power (mW)		N/A	89.5×10^{-3}	46.8×10^{-3}	11.4×10^{-3}	211.0	110.0	100.6	40.5
Energy/Inference (mJ)		3.36	0.0143	2.25×10^{-3}	2.77×10^{-3}	6.64	1.83	1.45	1.31
Energy Efficiency (GOPS/W)		1462.5$^{(2)}$	2950.0$^{(2)}$	534.1	704.2	1100	2264.5	3222.0	3595.9

$^{(1)}$ The original Kvasir dataset is further modified, which is unknown for the detailed dataset composition.

$^{(2)}$ The precision is normalized to 8 bits.



Process	TSMC 22nm
Die Area	3.798×3.798 mm^2
Supply Voltage	0.8 V
Frequency	33 MHz
RRAM Size	6 MB
Power	60 mW
Read Speed	8 Gb/s
Endurance	1×10^4
Lifetime Improvement	2.37 times

Figure 8: System verification, including system test board, chip photo of RRAM, and corresponding summary table.

such as detection, segmentation, etc. In future work, we will consider the multi-task requirements in other scenarios and migrate our proposed MTVR architecture to various applications.

5 CONCLUSION

This work proposes an efficient accelerator MTVR that can simultaneously run multi-modal tasks. According to the requirements of practical application scenarios, MTVR can work in high-performance or low-power modes. To save energy, we have designed an early exit mechanism. The hybrid memory access scheme is designed to reduce external memory access. The experimental results show that the MTVR accelerator achieved an energy efficiency of 3.6 TOPS/W. In future work, we will further explore the application of MTVR accelerators in other multi-task scenarios.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant No. 62104025 and State Key Laboratory of Computer Architecture (ICT, CAS) under Grant No. CARCHB202117. Liang Chang is the corresponding author.

REFERENCES

[1] I. Wohlgenannt, A. Simons, and S. Stieglitz, "Virtual reality," *Business & Information Systems Engineering*, vol. 62, pp. 455–461, 2020.

[2] Z. Li, S. Kim, D. Im, D. Han, and H.-J. Yoo, "An 0.92 mj/frame high-quality fhd super-resolution mobile accelerator soc with hybrid-precision and energy-efficient cache," in *2022 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 2022, pp. 1–2.

[3] X. Duan, Y. Chen, M. Li, Y. Rong, R. Xie, and J. Han, "Uarch: A super-resolution processor with heterogeneous triple-core architecture for workloads of u-net networks," *IEEE Transactions on Biomedical Circuits and Systems*, 2023.

[4] J. Lee, J. Lee, and H.-J. Yoo, "Srnpu: An energy-efficient cnn-based super-resolution processor with tile-based selective super-resolution in mobile devices," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 3, pp. 320–334, 2020.

[5] Y.-H. Chiang, T.-S. Chang, and S. J. Jou, "A 14 μj/decision keyword-spotting accelerator with in-sramcomputing and on-chip learning for customization," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 9, pp. 1184–1192, 2022.

[6] R. Guo, Y. Liu, S. Zheng, S.-Y. Wu, P. Ouyang, W.-S. Khwa, X. Chen, J.-J. Chen, X. Li, L. Liu *et al.*, "A 5.1 pj/neuron 127.3 us/inference rnn-based speech recognition processor using 16 computing-in-memory sram macros in 65nm cmos," in *2019 Symposium on VLSI Circuits*. IEEE, 2019, pp. C120–C121.

[7] J. Liu, Z. Zhu, Y. Zhou, N. Wang, G. Dai, Q. Liu, J. Xiao, Y. Xie, Z. Zhong, H. Liu, L. Chang, and J. Zhou, "4.5 bioaip: A reconfigurable biomedical ai processor with adaptive learning for versatile intelligent health monitoring," in *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 64, 2021, pp. 62–64.

[8] L. Chang, X. Zhao, and J. Zhou, "Adas: A high computational utilization dynamic reconfigurable hardware accelerator for super resolution," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 16, no. 3, pp. 1–22, 2023.

[9] L. Chang, S. Yang, Z. Chang, H. Fan, J. Zhou, and J. Zhou, "Tdpro: Time-domain-based computing-in memory engine for ultra-low power ecg processor," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2023.

[10] B. Shi, Z. Tang, G. Luo, and M. Jiang, "Winograd-based real-time super-resolution system on fpga," in *2019 International Conference on Field-Programmable Technology (ICFPT)*. Tianjin: IEEE, 2019, pp. 423–426.

[11] C.-Y. Chih, S.-S. Wu, J. P. Klopp, and L.-G. Chen, "Accurate and bandwidth efficient architecture for cnn-based full-hd super-resolution," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. Florence: IEEE, 2018, pp. 1–5.