A 60fps 496mW Multi-Object Recognition Processor with Workload-Aware Dynamic Power Management

Joo-Young Kim, Seungjin Lee, Jinwook Oh, Minsu Kim, and Hoi-Jun Yoo
School of EECS, KAIST, 373-1, Guseong-dong, Yuseong-gu, Daejeon, 305-701, KOREA
trample7@eeinfo.kaist.ac.kr

ABSTRACT

An energy efficient object recognition processor is proposed for real-time visual applications. Its energy efficiency is improved by lowering average power consumption while sustaining high frame rate. To this end, the proposed processor features from all levels of chip design. In architecture level, it performs 3-stage task pipelining for high frame rate operation and workload-aware dynamic power management for low power consumption. In block level, energy efficient special purposed engines are employed while software controlled clock gating is exploited for fine-grained clock control. In circuit level, analog-digital mixed design is used to reduce power with the same performance. As a result, the 49mm^2 chip in a 0.13μm technology achieves 60fps object recognition for VGA (640x480) input with 496mW power at the supply of 1.2V. It means only 8.2mJ is dissipated per frame, which is 3.2X more energy efficient than the state of the art.

Categories and Subject Descriptors

C.1.2 [**Multiprocessors**]: parallel processors
I.4.8 [**Scene Analysis**]: object recognition

General Terms

Algorithms, Design, Management, Performance

Keywords

Multimedia processor, energy efficient object recognition, workload-aware dynamic power management

1. INTRODUCTION

Object recognition becomes a key technology for advanced visual applications such as autonomous cruise control, intelligent robot vision, and surveillance system [1-3]. Basically, most conventional object recognition mechanisms are performed by following algorithm sequences [4]. Firstly, various scale spaces for input video stream are generated by a cascade filtering approach and feature points are extracted among them using local maximum search. Then, the extracted feature points are converted to descriptor vectors that describe magnitude and orientation characteristics of them. Lastly, the final decision is made by performing nearest neighbor matching with pre-defined database composed of more than ten thousands of object vectors. Since each stage of object recognition includes considerable amount of computations, it is hard to achieve real-time operation. However, to apply it to the real-time visual applications aforementioned, a high frame rate operation, for example over 30fps, is demanded. In addition, low power operation is also required for battery limited applications such as mobile robot or handset device.

To reduce computational complexity of conventional object recognitions, the processor of [2] introduced visual attention paradigm to object recognition hardware. With cellular neural network based visual attention engine [5], it reduces internal workload of object recognition by filtering meaningless feature points. Although it achieves real-time operation, however, it has not fully considered energy efficiency in terms of hardware utilization and aggressive power management. High cost attention engine is utilized only for short time within a frame and only dynamic power is managed by clock gating.

In this work, we propose an energy efficient object recognition processor with following key features from all levels of chip design. First, workload reducing 3-stage multi-object recognition algorithm is proposed and implemented in pipelined manner in the proposed processor to achieve high frame rate operation. Second, dynamic power management with workload-aware task scheduling is performed to reduce not only dynamic power but also static and clock power dissipation. Third, software controlled clock gating is exploited for fine-grained clock control. Last, special purposed engines and analog-digital mixed mode design achieves low energy block implementation. This paper consists of six sections. Section 2 describes the system architecture of the proposed processor for high frame rate object recognition. Section 3 proposes chip power architecture and analyzes its dynamic power management including software controlled clock gating. Section 4 explains low energy building block design. In Section 5, chip implementation results and performance evaluation will be shown. Finally, conclusion is made in Section 6.

2. SYSTEM ARCHITECTURE FOR HIGH FRAME RATE OPERATION

Fig. 1 shows overall algorithm flow of the proposed multi-object recognition. It consists of 3 stages: visual perception, parallel processing, and object decision. The visual perception stage preliminary extracts static features such as intensity, color and orientation, and dynamic feature such as motion vector from the input image to generate saliency map. Then, based on this map, it determines region-of-interests (ROIs) for each object in a unit of 40x40 pixel sized tile, called grid-tile. After that, the parallel processing stage extracts feature points out of the selected ROI grid-tiles using parallel and repetitive kernel processing, and generates descriptor vectors. Lastly, the final object decision is made by iterative database matching for descriptor vectors of each object. This grid-based ROI processing, which divides an input image into over a number of small sized grid-tiles and processes the ROIs of objects based on them, enables fine-grained boundary extraction for

ISLPED'09, August 19–21, 2009, San Francisco, California, USA.
Copyright 2009 ACM 978-1-60558-684-7/09/08...$10.00.

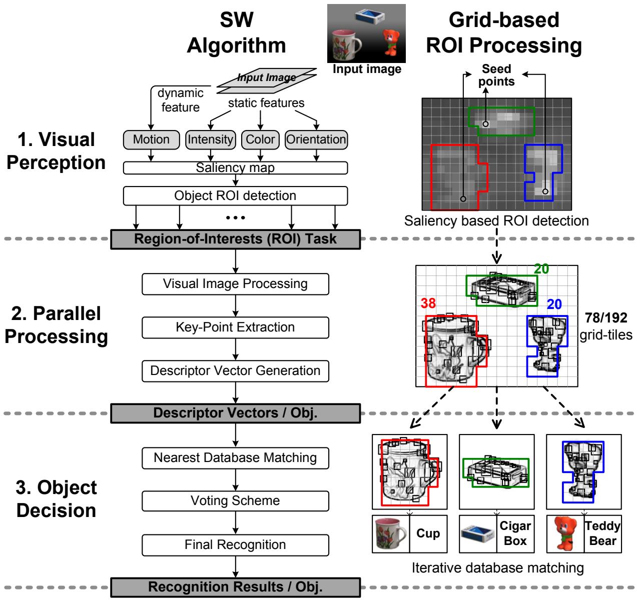
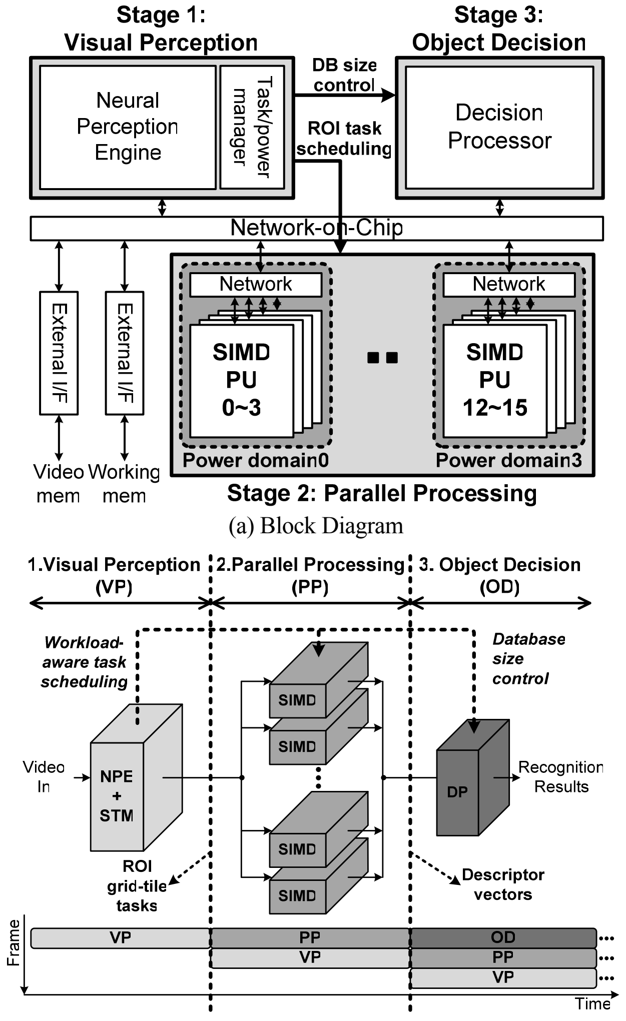


Fig. 1 3-Stage Multi-Object Recognition Algorithm

each object and results in workload reduction of input image. By applying this model to the 164 sample images of multi-object database Berkeley Segmentation Database (BSD300) [6], object ROIs of images are measured to 41% of overall area on average. This means the workloads of the following parallel processing and object decision stage can be also reduced in similar portion by processing only the selected region. As a result, workload reduction in algorithm level greatly improves energy efficiency in recognition processing by eliminating needless processing for uninterested area of the image, originally supposed to be processed.

Fig. 2 (a) shows the block diagram of the proposed recognition processor consisting of 21 IP blocks. The neural perception engine (NPE) is responsible for the visual perception stage. It extracts various features from the input image and generates ROIs for each object in a grid-tile unit. 16 SIMD processor units (SPU) are responsible for the parallel processing stage. They have 4 separated power domains and generate descriptor vectors for the selected ROIs. A decision processor (DP) performs the object decision stage with dedicated blocks for vector distance calculation and matching. A SPU task/power manager (STM) is specially devised to distribute ROI grid-tile tasks from the NPE to the 16 SPUs and to manage 4 power domains of the 16 SPUs. 2 memory interfaces are integrated to communicate with 2 external memories. To increase the parallelism and utilization of the proposed hardware, the proposed 3 object recognition stages, the visual attention, parallel processing, and object decision are executed in the pipeline as shown in Fig. 2 (b). Pipelined data are ROI grid-tile tasks and descriptor vectors for between the 1st and 2nd stage and the 2nd and 3rd stage, respectively. For efficient task pipelining, the execution times of 3 stages are controlled to be balanced. The execution time of visual perception is deterministic because it is mainly composed of repeated feature extractions and normalizations. Meanwhile, the execution times of parallel processing and object decision stage are varied according to their workloads, the number of extracted ROI grid-tiles and descriptor vectors, respectively. Therefore, to balance the execution times of 3 stages, the STM controls the execution time of the parallel processing and object decision to the one of visual perception, which is fitted to target pipeline time 16ms. To control



(a) Block Diagram

(b) 3-Stage Task Pipelining

Fig. 2 System Architecture

the execution time of the parallel processing stage, the STM performs workload-aware task scheduling. It determines the number of the operating SPUs according to the amount of measured ROI grid-tiles. Since the number of processing units increases in proportional to the workload, the execution time is kept in constant level. And the execution time is controlled by performing scheduling more aggressively or conservatively. The execution time of object decision stage is controlled by adjusting the size of applied database in vector matching process. Based on the vector matching algorithm of [7], the overall execution time of the object decision stage is estimated with the number of input descriptor vectors and the size of applied database. Using this, the execution time can be controlled by configuring database coverage rate of the DP after measuring the number of descriptor vectors from the parallel processing stage. As a result, the execution time of the 2nd and 3rd stage is balanced to 16ms, even under varying workload conditions, and the overall processor achieves 60 fps frame rate.

3. DYNAMIC POWER MANAGEMENT

In this chapter, we will discuss about the dynamic power management schemes applied to the proposed processor. Fig. 3 shows overall power management architecture. First of all,

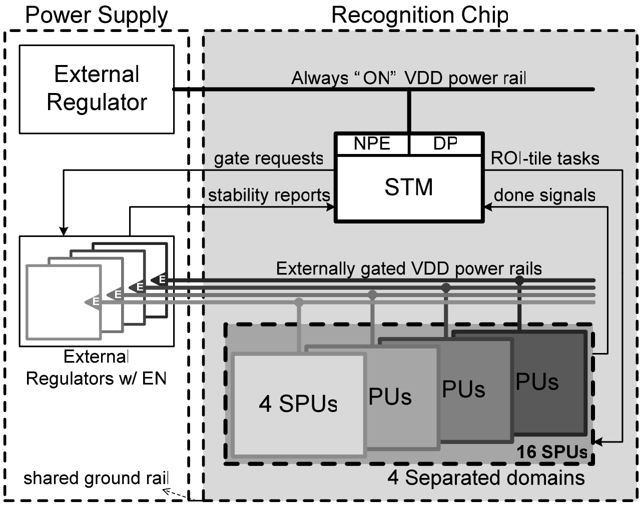
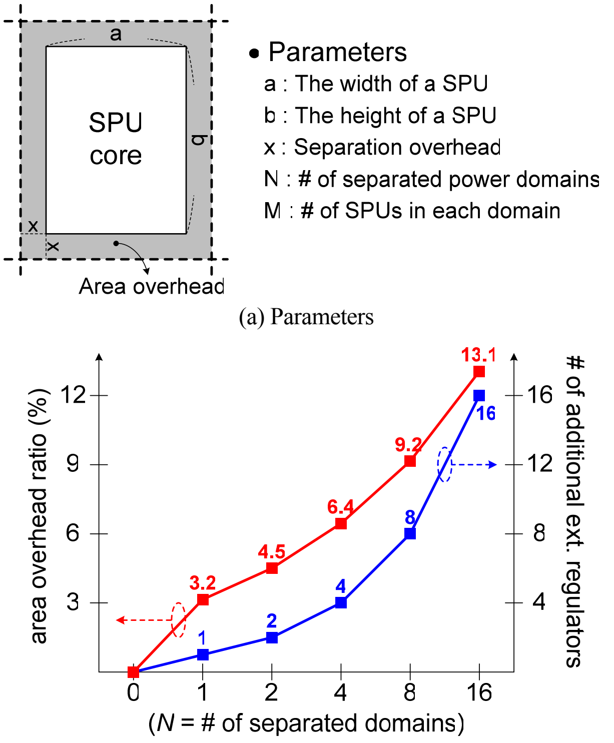


Fig. 3 Power Management Architecture

regulators for power supply of the processor exist in external and ground rails of off-chip and on-chip are all shared. In the chip, the NPE, DP, and STM are always on domain, while the 16 SPUs are in externally controlled 4 separated power domains. The STM is responsible for managing the 4 power domains of the 16 SPUs according to the workload variations of the NPE. In this power domain architecture, we will discuss following 3 issues for low power consumption of the processor. First, the overhead of power domain separation is analyzed according to the number of power domains. Second, workload-aware scheduling scheme is proposed for efficient power gating and its power reduction effect is evaluated. Third, fine-grained clock gating scheme is explained for further reduction of operating power in activated domains.

3.1 POWER DOMAIN SEPARATION OVERHEAD

Power gating is an efficient and robust way to reduce leakage power in deep sub-micron technology by disconnecting idle blocks from the power supply [8]. To avoid excessive design overheads of on-chip power gating, such as header and footer transistor addition to logic and memory blocks and gate signal generation, off-chip power gating is adopted with a simple backend process using standard library. For off-chip power gating, all we have to do in design time is to separate the power domain of chip into several pieces and generate request signals for power gating to external regulators. Therefore, major overheads of off-chip power gating are summarized as two things: area overhead in on-chip domain and additional regulators in off-chip domain. Fig. 4 (a) shows the additional spaces around the processing core required for power domain separation, which are mainly for power rings. In this figure, the additional length for domain separation is denoted as x where the width and height of the SPU core is defined as a and b, respectively. The x is a summated value that includes all required factors such as power ring space, interval distance, and margin. The number of separated power domains is N, while the number of the clustered SPU cores in same power domain is M. It is assumed that the number of SPU cores in each power domain is even. Therefore, $N{\cdot}M$ should be equal to the total number of the SPUs, or 16 in this design. It is also assumed that the ratio of the core area is kept even if the M is not the square of integer. With these configurations, the total area of N separated power domains with $N{\cdot}M$ SPUs is as following.



(a) Parameters

(b) Overheads

Fig. 4. Power domain separation

$$Total\ area:\ (\sqrt{M}\cdot a+2\cdot x)\times(\sqrt{M}\cdot b+2\cdot x)\times N$$

The area overhead of domain separation is computed as follows.

$$Overhead:\ (\sqrt{M}\cdot a+2\cdot x)\times(\sqrt{M}\cdot b+2\cdot x)\times N-M\cdot N\cdot a\cdot b$$
$$=2\cdot N\cdot(\sqrt{M}\cdot(a+b)\cdot x+2\cdot x^2)$$

Since the total number of SPU cores is 16, the number of separated power domains N can be configured to 1, 2, 4, 8, and 16, while the number of clustered SPU cores is 16, 8, 4, 2, and 1, respectively. The following equation shows the overhead ratio defined as the ratio of the area overhead and the core area.

$$Ratio=\frac{2\cdot N\cdot(\sqrt{M}\cdot(a+b)\cdot x+2\cdot x^2)}{N\cdot M\cdot a\cdot b}=\frac{(a+b)\cdot x}{2\cdot a\cdot b}\cdot\sqrt{N}+\frac{x^2}{4\cdot a\cdot b}\cdot N$$

Meanwhile, the number of additional regulators with enable is the same with the number of separated power domains N.

$$External\ overhead = N\ (regulator\ devices)$$

The graph of Fig. 4 (b) shows the internal and external overhead of power domain separation. For this, the values of a, b, and x are defined to 1.0mm, 1.7mm, and 0.04mm, respectively, which are measured from real physical implementation in a 0.13µm process.

3.2 WORKLOAD-AWARE TASK SCHEDULING

To perform power gating under the condition that the power domain of the 16 SPUs is separated into N domains, an appropriate scheduling scheme is demanded. To this end, the SPU task/power manager (STM) performs workload-aware task scheduling (WATS) when it distributes ROI grid-tile tasks to the 16 SPUs. Fig. 5 shows the flow chart of the WATS. First, the STM measures the workload of the current frame based on the number of extracted ROI grid-tiles.

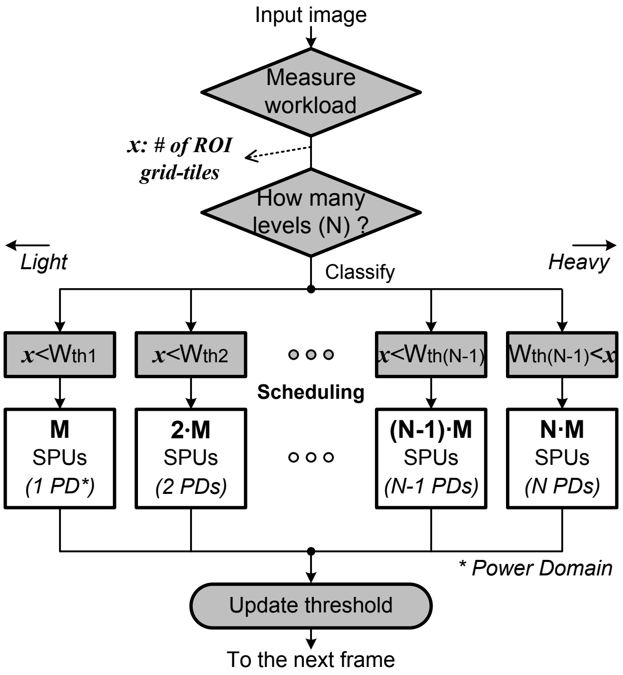


Fig. 5. Workload-Aware Task Scheduling

Then, it classifies the measured workload to one of N levels as linearly increased N-1 threshold values, where N is the number of separated power domains. Basically, the threshold values can be set to make each workload level to be equally divided. After that, the STM determines the number of operating SPUs according to the selected workload level, to be the multiple of M, which is the number of the SPUs in a domain. The threshold values can be updated to meet target pipeline time.

Since the WATS determine the number of operating SPUs in a unit of power domain, power gating can be easily applied by gating the unused power domains off. The gate request signals are generated only once per frame, considering a settling time of external regulators more than 100μs. To evaluate static power reduction effect of the WATS, following terms are defined under the condition all workload levels are equally divided.

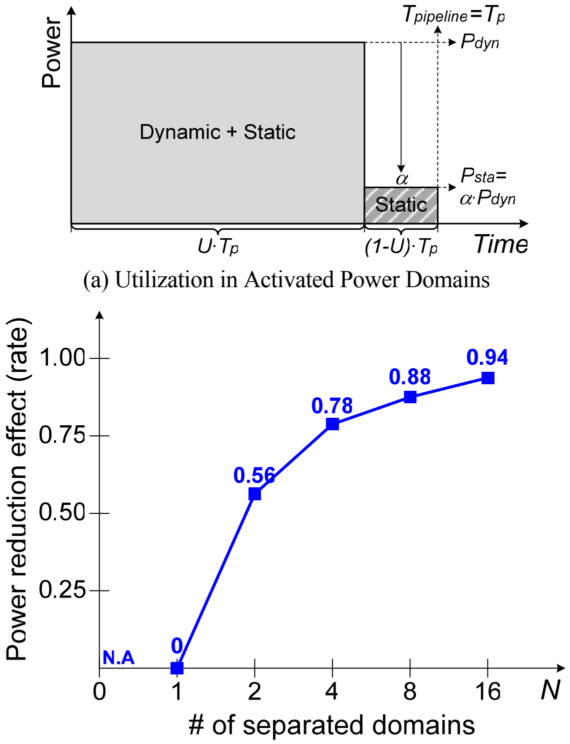
$$W = \textit{workload of current frame} \ (= \# \textit{ of ROI grid-tiles})$$
$$W_{max} = \textit{maximum workload of frame}$$
$$W_{step} = \textit{interval of one workload level} \ (= W_{max}/N)$$

The number of activated domains by the WATS is defined as

$$A = \left[\frac{W}{W_{step}} \right] + 1$$

In the activated power domains, average operating ratio of the SPUs can be simplified by utilization U as shown in Fig. 6 (a). Both dynamic power and static power is dissipated in the utilized time, while only static power is dissipated in the idle time.

$$U = \frac{W}{A \cdot W_{step}} = \frac{W}{\left(\left[\frac{W}{W_{step}} \right] + 1 \right) \cdot W_{step}}$$



(a) Utilization in Activated Power Domains

(b) Static Power Reduction Effect

Fig. 6 WATS Evaluation

Then, the total average power dissipated by the idle phase of activated power domains can be computed as follows, where P_{dyn} is dynamic power of a SPU and technology-driven α is a ratio of static power over dynamic power.

$$T_{WATS} = A \cdot M \cdot P_{dyn} \cdot \alpha \cdot (1 - U)$$

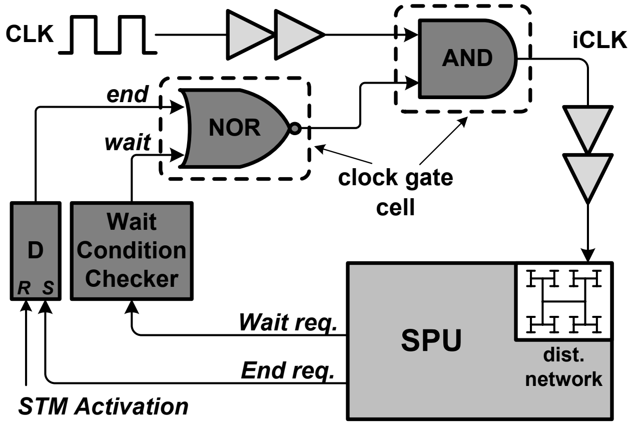
The total average power dissipation in case without the WATS is

$$T_{NORMAL} = N \cdot M \cdot P_{dyn} \cdot \alpha \cdot \left(1 - \frac{W}{W_{max}}\right)$$

Therefore, the static power reduction effect by the WATS can be measured by following equation.

$$Effect = 1 - \frac{T_{WATS}}{T_{NORMAL}} = \frac{N - A}{N \cdot \left(1 - \frac{W}{W_{max}}\right)}$$

Fig. 6 (b) shows the static power reduction effect of the WATS according to the N for realistic workloads extracted from video camcorder whose average and standard deviation of ROI grid-tiles are 79.6 and 23.9, respectively. As a result, the static power reduction effect gets stronger as the N value increases. When the number of power domains is 16, over 90% of static power can be eliminated. However, the overhead of domain separation is also sharply increased by the N as shown in Fig. 4 (b). In this trade-off relation, we determine the N value as 4, which is the most cost effective point that reduces more than 75% of static power with only 6.4% of area overhead.



(a) Block Diagram

Program end	Program wait
- Write *any_value* @ *end_addr* SC AL N : MOV r0 *end_addr* SC AL N : STRI NU r0 r0 0x0	- Write *sync_idx* @ *wait_addr* SC AL N : MOV r0 *wait_addr* SC AL N : MOV r1 *sync_idx* SC AL N : STRI NU r1 r0 0x0

(b) Program Code

Fig. 7 Software Controlled Clock Gating

3.3 SOFTWARE CONTROLLED CLOCK GATING

To minimize dynamic power consumption in activated power domains, software controlled clock gating is applied to each SPU. The block diagram of software controlled clock gating is shown in Fig. 7 (a). First of all, the clock of SPU is activated by the STM when it assigns ROI grid-tile task to the SPU. During the operations, a clock of SPU can be gated by its own two kinds of software requests. The first is *end request* occurred when the SPU has finished its assigned task. It generates hardwired *end request* by writing any value at pre-defined end address. Then, the end signal is set to 1 until the STM activates the SPU again. The end signal disconnects the external clock signal with internal clock buffers of the SPU. The second is *wait request* generated when the SPU should stop and wait for the other module's operation. It sends *wait request* by writing some value at pre-defined wait address. Different from *end request*, the SPU writes the index value to the wait address to notify which bit of a 16-bit barrier register should be checked for clock recovery. In this case, the clock is gated until all wait conditions resolved. When all wait conditions are removed by their initiators, the *wait* signal is deactivated by wait condition checker and the clock signal is automatically recovered. With the software controlled clock gating, the clock of the SPU can be controlled in a cycle level.

4. LOW ENERGY BLOCK DESIGN

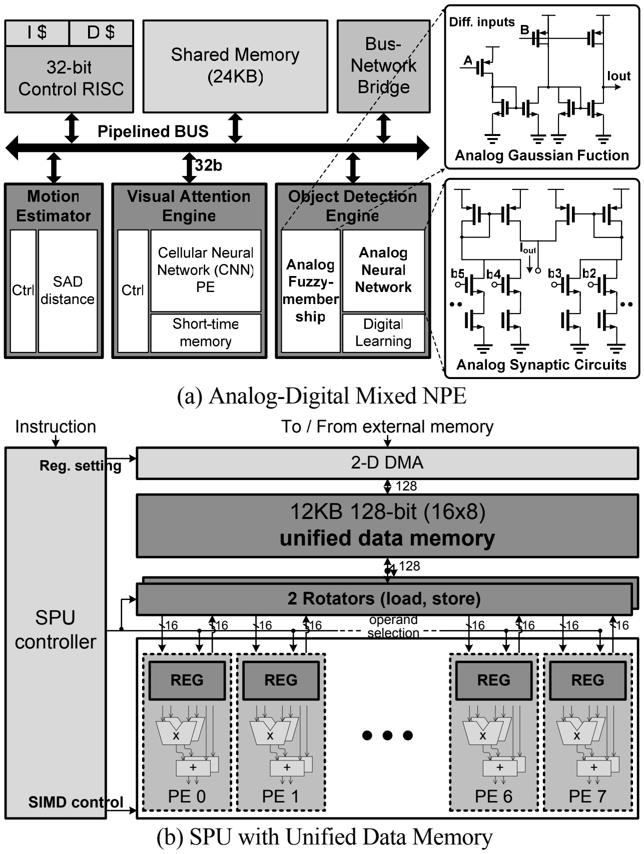
To achieve energy efficient object recognition, each individual building block of the processor should perform energy efficient processing either. To this end, low-energy oriented designs are applied for major building blocks, the NPE and 16 SPUs.

4.1 NEURAL PERCEPTION ENGINE

The visual perception algorithm [9] for the visual perception stage consists of several feature extraction processes for saliency map generation and classification processes for ROI detection. To implement this in an energy efficient way, the NPE employs a RISC processor and 3 specialized hardware blocks as shown in Fig. 8 (a). Motion estimator (ME) is employed to extract dynamic motion vectors from two sequential frames in time domain. The cellular neural network based visual attention engine (VAE) [5] extracts various static features using its collective processing on entire image pixels. And the proposed object detection engine (ODE) performs the final ROI detection using neuro-fuzzy classification. In the design of the ODE, analog-digital mixed design is employed for energy efficient processing. By implementing non-linear Gaussian membership function and neural synaptic multiply circuits in analog circuits, the area and power of the ODE is reduced by 59% and 44%, respectively, compared to those of digital-only implementations. The RISC processor takes a role in controlling the 3 dedicated engines and performing software oriented operations between the operations of dedicated engines. It also performs clock gating of the dedicated engines when they are in the idle state. As a result, the NPE achieves more than 7x energy efficient operations while running visual perception algorithm, than the mentioned RISC processor does it without the 3 dedicated hardware blocks.

4.2 SIMD PROCESSOR UNIT

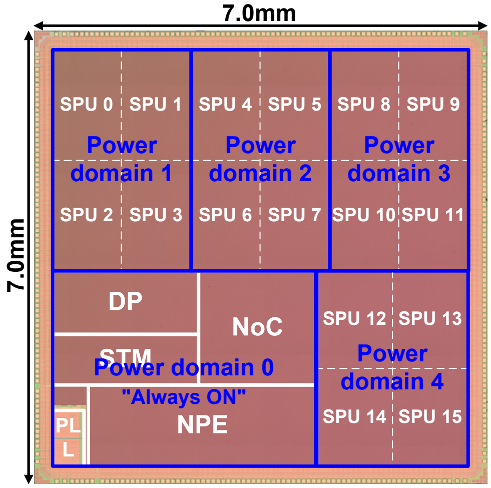
The 16 SPUs are the most dominant part of the processor in terms of area, computing performance, and power dissipation. It consists of a SPU controller, eight SIMD controlled 16b processing elements (PEs), 12KB 128-bit wide data memory with 2 rotators, and 2D DMA. For local data memory for the 8 PEs, one unified 128-bit wide memory is employed rather than 8 16-bit wide memories, since it saves area and power consumption by 30.4% and 36.4%, respectively. The data transfers between the memory and PEs are



(a) Analog-Digital Mixed NPE

(b) SPU with Unified Data Memory

Fig. 8 Low Energy Building Blocks



(a) Chip Micrograph

Process Technology	0.13mm 1P 8M CMOS	
Package	320 pin FPGA	
Power supply	1.2V core, 2.5V I/O	
Operating Frequency	200MHz IPs / 400MHz NoC	
Die Size	7mm x 7mm	
Transistor Counts	36.4M transistors 3.73M gates / 396KB SRAM	
Peak Computing Performance	16 SPUs	128 GOPS
	NPE	54 GOPS
	DP	19.4 GOPS
	Total	201.4 GOPS
Power Consumption	Peak: 695mW / Average: 496mW	
Power Efficiency	290 GOPS/W	
Target Application	Multi-object recognition	
Input Screen	VGA (640x480 pixels) video	
Recognition Frame-rate	60 fps	
Multi-object Captured	Up to 10 objects / scene	
Database Size	16384 vectors (50 objects)	

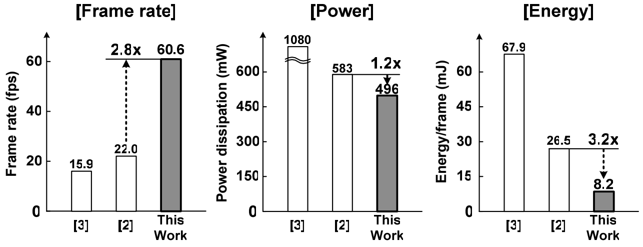
(b) Chip Features

Fig. 9 Chip Implementation Results

minimized by the program, and are always performed in a unit of line, or 128-bit data. The 2 rotators between them give flexibility in data transfers by aligning the data in 16-bit unit. Under the unified data memory, each PE has its own register file. With the independently managed status table, each of 8 PEs can be conditionally executed for the same instruction. The 2-D DMA is responsible for data transfers between external memory and internal data memory. The parallel execution hides excessive latency for external memory access.

5. CHIP IMPLEMENTATION

The recognition SoC is fabricated in 0.13µm CMOS technology. Fig. 9 shows chip micrograph and features. Its area amounts to 49mm^2 and contains 36.4M transistors including 3.73M gates and 396KB SRAM. The operating frequency is 200MHz for IP blocks and 400MHz for network-on-chip interconnection. The chip achieves 60fps object recognition for VGA (640x480) video resolution with 496mW average power consumption at the supply voltage of 1.2V. Fig. 10 shows performance comparisons with the previous recognition processors [2-3]. With a 3-stage task pipelining and fine grained ROI processing, the proposed processor achieves 60fps frame rate, which is a 2.8X higher frame rate than the previous processors, even it scales up the video resolution from QVGA to VGA. And the proposed task scheduling and dynamic power management reduces the power of the 16 SPUs by 32% and achieves 496mW total average power consumption. It is slightly decreased from the previous ones while 60fps frame rate is sustained. As a result, the proposed processor consumes only 8.2mJ energy to process a single video frame, which is 3.2X improved than the previous processors.



	[3]	[2]	This Work
Frame rate (fps)	15.9	22.0	**60.6**
Power (mW)	1080	583	**496**
Energy/frame (mJ)	67.9	26.5	**8.2**
Video resolution	QVGA (320x240)	QVGA (320x240)	**VGA (640x480)**
Applied program	Feature extraction & descriptor generation	Full object recognition w/ single attention	**Full object recognition w/ multi attention**

Fig. 10 Performance Comparisons

6. CONCLUSION

An energy efficient object recognition processor is presented. For high frame rate recognition, 3-stage multi-object recognition algorithm is proposed and efficiently pipelined in the processor. For low power operation, dynamic power management is applied with workload-aware task scheduling. For low energy building blocks, dedicated engines and analog-digital mixed mode design are used. As a result, the implemented processor consumes only 8.2mJ per frame, which is 3.2x more energy efficient than the state of the art recognition processor.

7. REFERENCES

[1] J.-Y. Kim, et al., "A 201.4GOPS 496mW Real-time Multi-Object Recognition Processor with Bio-inspired Neural Perception Engine," IEEE ISSCC, pp. 150-151, 2009.

[2] K. Kim, et al., "A 125GOPS 583mW Network-on-Chip Based Parallel Processor with Bio-inspired Visual Attention Engine," IEEE ISSCC, pp. 308-309, 2008.

[3] D. Kim, et al., "An 81.6 GOPS Object Recognition Processor Based on NoC and Visual Image Processing Memory," IEEE CICC, pp.443-446, 2007.

[4] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," ACM International Journal of Computer Vision, Vol.60, Issue 2, pp. 91-110, Jan. 2004.

[5] S. Lee, et al., "The Brain Mimicking Visual Attention Engine: An 80x60 Digital Cellular Neural Network for Rapid Global Feature Extraction," IEEE Symp. VLSI circuits, pp. 26-27, 2008.

[6] D. Martin, et al., "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics," IEEE ICCV 2001.

[7] J.-Y. Kim et al., "A 66fps 38mW Nearest Neighbor Matching Processor with Hierarchical VQ Algorithm for Real-Time Object Recognition," IEEE A-SSCC, pp.177-180, 2008.

[8] M. Keating, et al., "Low power methodology manual for system on chip design," Springer, 2007.

[9] M. Kim et al., "A 22.8GOPS 2.83mW Neuro-fuzzy Object Detection Engine for Fast Multi-object Recognition," IEEE Symp. on VLSI circuits 2009, not published yet.