A Low-Power Sensor Hub SoC Design with an Intelligence Boost Engine for IoT Applications

Gi-Ho Park, Minkwan Kee, Hyun-su Seon,
Kyungbae Kim*, Cheolho Jeong*, Sungho Kwon*,
Jongsung Lee*

MPSoc ’16, Nara, Japan

Sejong University / *Standing-egg Inc.

SEJONG UNIVERSITY
Department of Computer Engineering

Standing egg

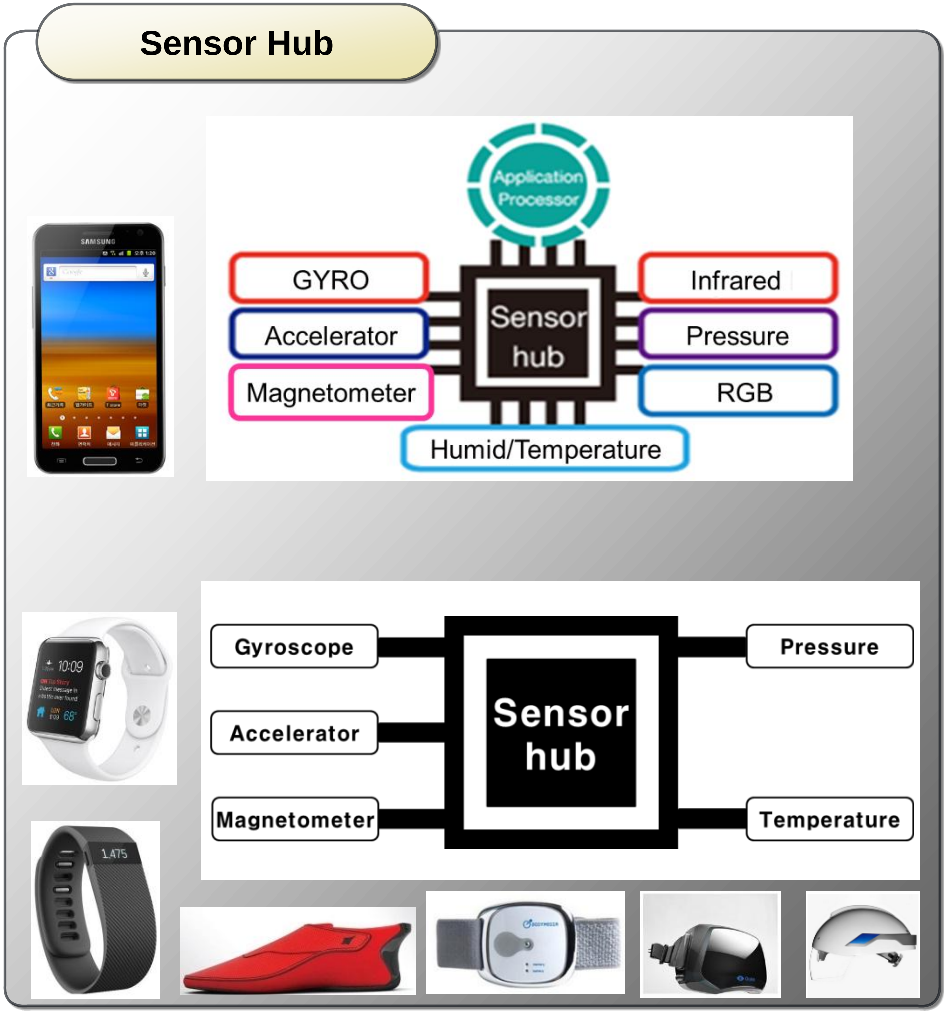
Outline

- Introduction
- Architecture of Sensor Hub SoC
- Intelligence Boost Engine (IBE)
- Power Mode, Sensor Interface Controller (SIC) and Management Scheme
- Implementation
- Conclusion and Future Works

What's Sensor Hub?

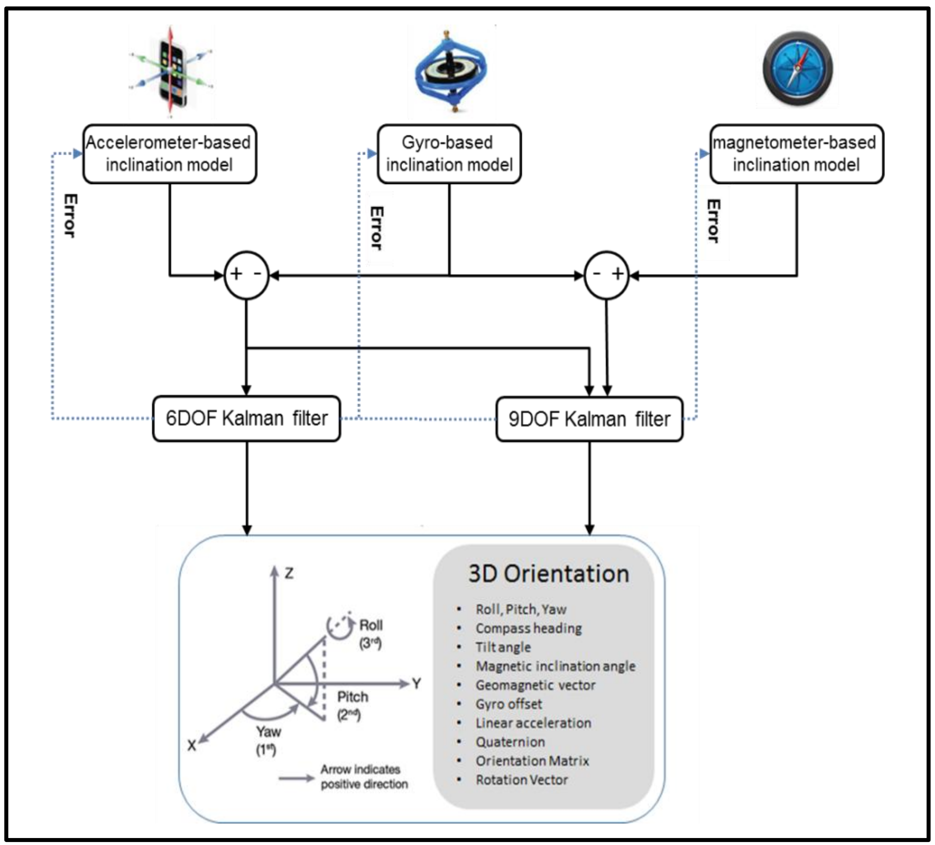
- Sensor hub
  - A device that connects with various sensors, controls and manipulates sensor data
  - Performs sensor fusion and other operations using the data generated by various sensors
  - Can work with AP or independently

- Role of sensor hub
  - Sensor calibration
  - Sensor data monitoring
  - Sensor noise filtering
  - Sensor data processing
    - Sensor fusion
  - Recognition
    - Motion detection

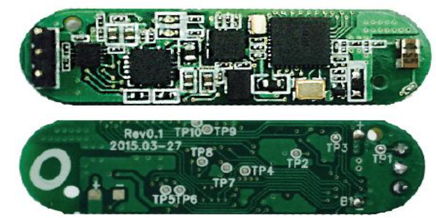


Major Target Applications of Sensor Hub

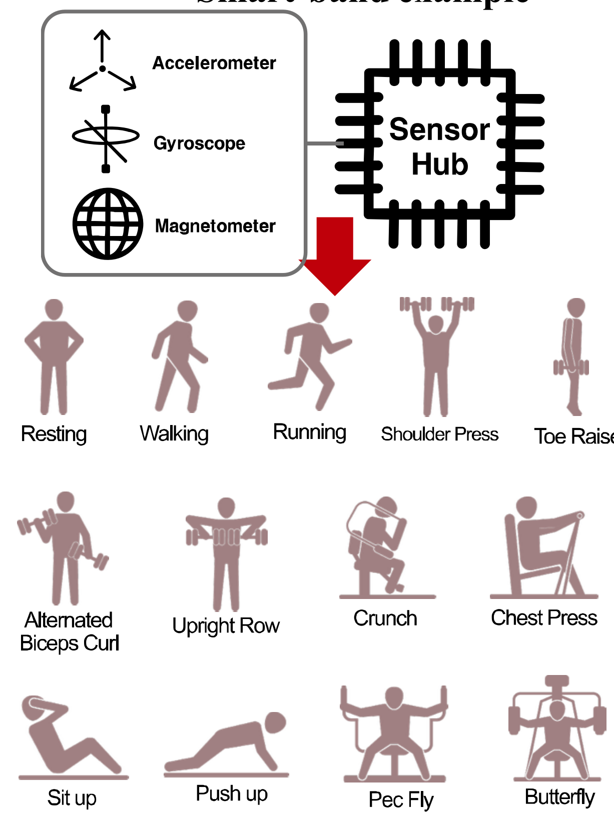
- **Sensor Fusion**
  - Acquire precise data by using different types of sensors



- **Motion Detection**



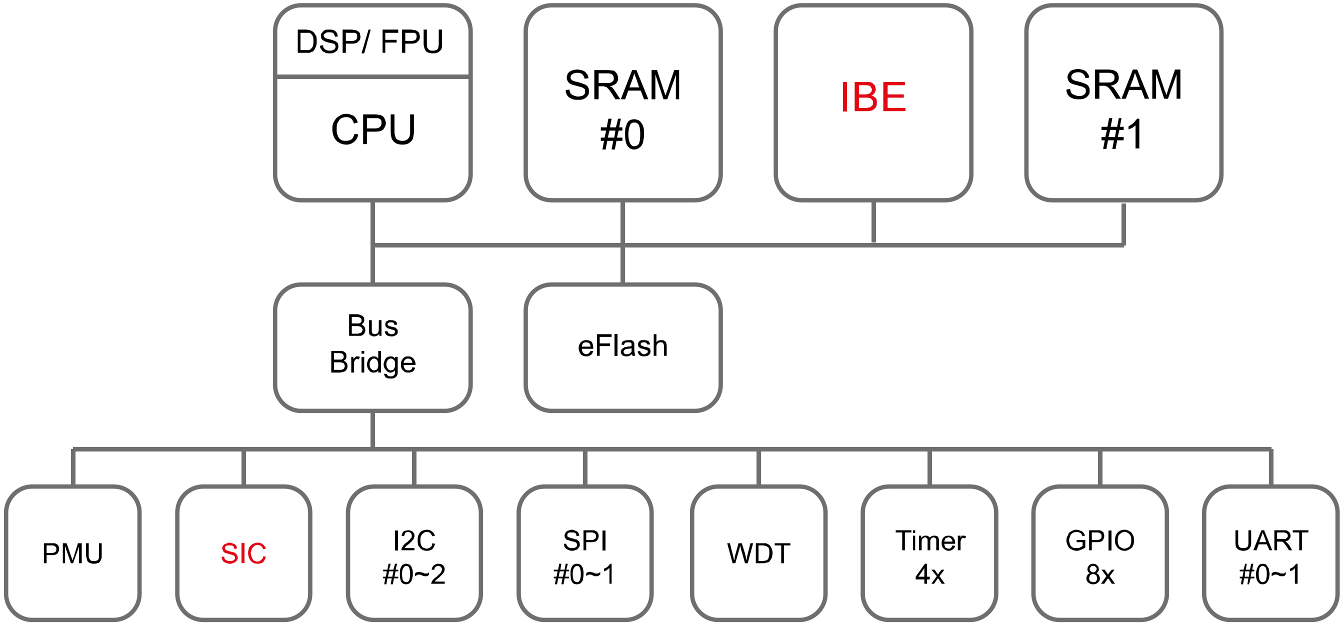
Smart-band example



A Low Power Sensor Hub (SLH-200) SoC Design

- SLH-200: Sensor Hub SoC for sensor fusion and IoT applications
  - 32bit RISC CPU integrated with FPU (ARM Cortex M4F)
  - 128KB of SRAM , 3.4Mb of embedded FLASH
  - Various peripherals (UART, GPIO, SPI & I2C)
  - Support max. 150MHz clock frequency
    - In power-down mode, could be operates with external clock source
  - Intelligence Boost Engine (IBE)™
    - A floating-point hardware accelerator for sensor fusion and machine learning algorithms
    - Support single precision 12-length vector operations – mul, add, mac, etc.
  - Support various power control modes
    - Normal/low-power/sleep/down-active/power-down mode
  - Sensor Interface Controller (SIC)™
    - Support sensor data monitoring feature in down-active/power-down mode
    - It can monitor sensors even when the CPU power is switched off

Overall System Architecture of SLH-200



Brief Specification	
CPU	ARM Cortex M4F
Memories	64KB SRAM x 2 (128KB), 3.4Mb Embedded FLASH Memory
Peripherals	Two SPI/ Three I2C / Two UART/ 8 GPIO/WDT/ 4 Timer
Others	IBE(Intelligent Boost Engine), SIC (Sensor Interface Controller)

Intelligence Boost Engine™ (IBE)

- Intelligence Boost Engine™ (IBE)
  - A H/W accelerator to perform sensor fusion and embedded machine learning algorithms
  - Support single precision floating-point data type
  - Accelerate kernels of sensor hub applications
    - Sensor fusion algorithm based on Kalman filter
    - Embedded machine learning algorithms (SVM, KNN)
  - Optimization feature for matrix element with zero value
    - Detects zero element in matrix and skips multiplication step
  - Consists of 12 MAC units, input matrix buffers, and control unit
  - Operates with relatively simple control mechanism
  - Can be operated at the same clock frequency of CPU

Application kernel analysis for IBE architecture design

- Target applications
  - Sensor fusion algorithms[1]
  - Motion detection[2],[3]

- Analysis of sensor fusion algorithm
  - Kernel functions : matrix multiplication, transpose
  - Data size : up to 12x12 matrix

Function name	Execution time
Rotation matrix	11%
Sensor estimation & Update measurement matrix	1%
Kalman gain	42%
Error estimation & Error corrections	5%
Error covariance matrix	35%
Re-create the noise covariance matrix	4%

Kernel code - Kalman gain

```
for (k = 0; k < 9; k++)
{
if ((*pfQw9x9ik != 0.0F) && (*pfC3x9jk != 0.0F))
{
if (*pfC3x9jk == 1.0F)
*pftmpA9x3ij += *pfQw9x9ik;
else if (*pfC3x9jk == -1.0F)
*pftmpA9x3ij -= *pfQw9x9ik;
else
*pftmpA9x3ij += *pfQw9x9ik * *pfC3x9jk;
}
```

SEJONG UNIVERSITY
Department of Computer Engineering

Application kernel analysis

- Analysis of motion detection
  - Detect motion using machine learning algorithm (SVM, KNN, etc.)
  - Kernel functions : vector operation & etc.

SVM functions

% time	call	name
94.13	194688198	Kernel::k_function
5.76	16281	svm_predict_values
0.13	16281	svm_load_model
0.00	23917	readline

Kernel

Operation code	Operation type
sum += px->value * py->value	vector-vector multiplication and accumulation
param.gamma*dot(x,y)+param.coef0	scalar operation

for(int t=times; t>0; t/=2)
{
if(t%2==1) ret*=tmp;
tmp = tmp * tmp;
}
	Square operation

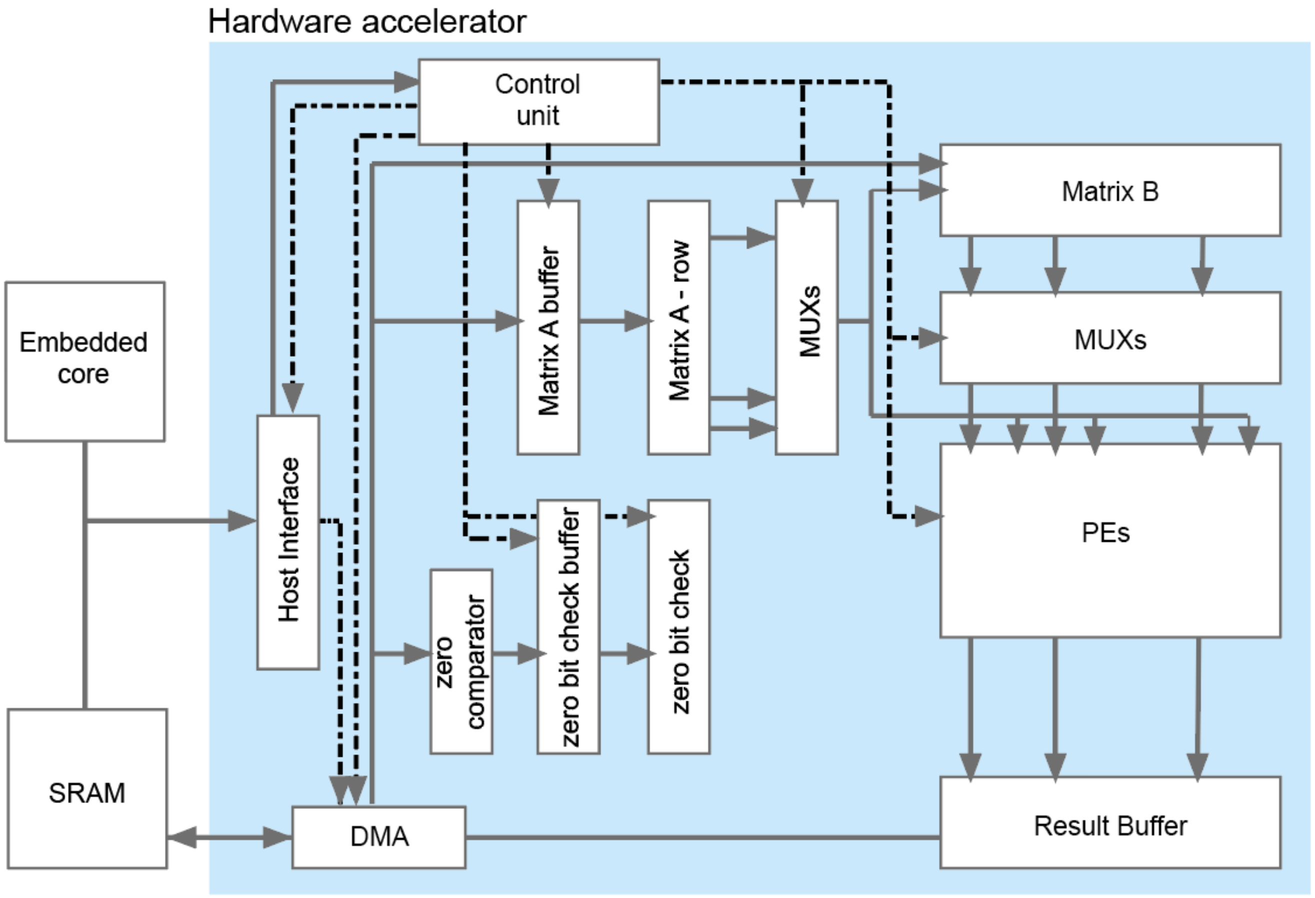
Supporting operations of IBE

- Mode 0~2 : Sensor fusion algorithm & general matrix operations
- Mode 3~4 : General vector operations

Mode #	Operation
0	Matrix A * Matrix B
1	Matrix A^T
2	Matrix A * Matrix B^T (SVM linear mode)
3	Vector * Scalar
4	R+=Vector1 * Vector2

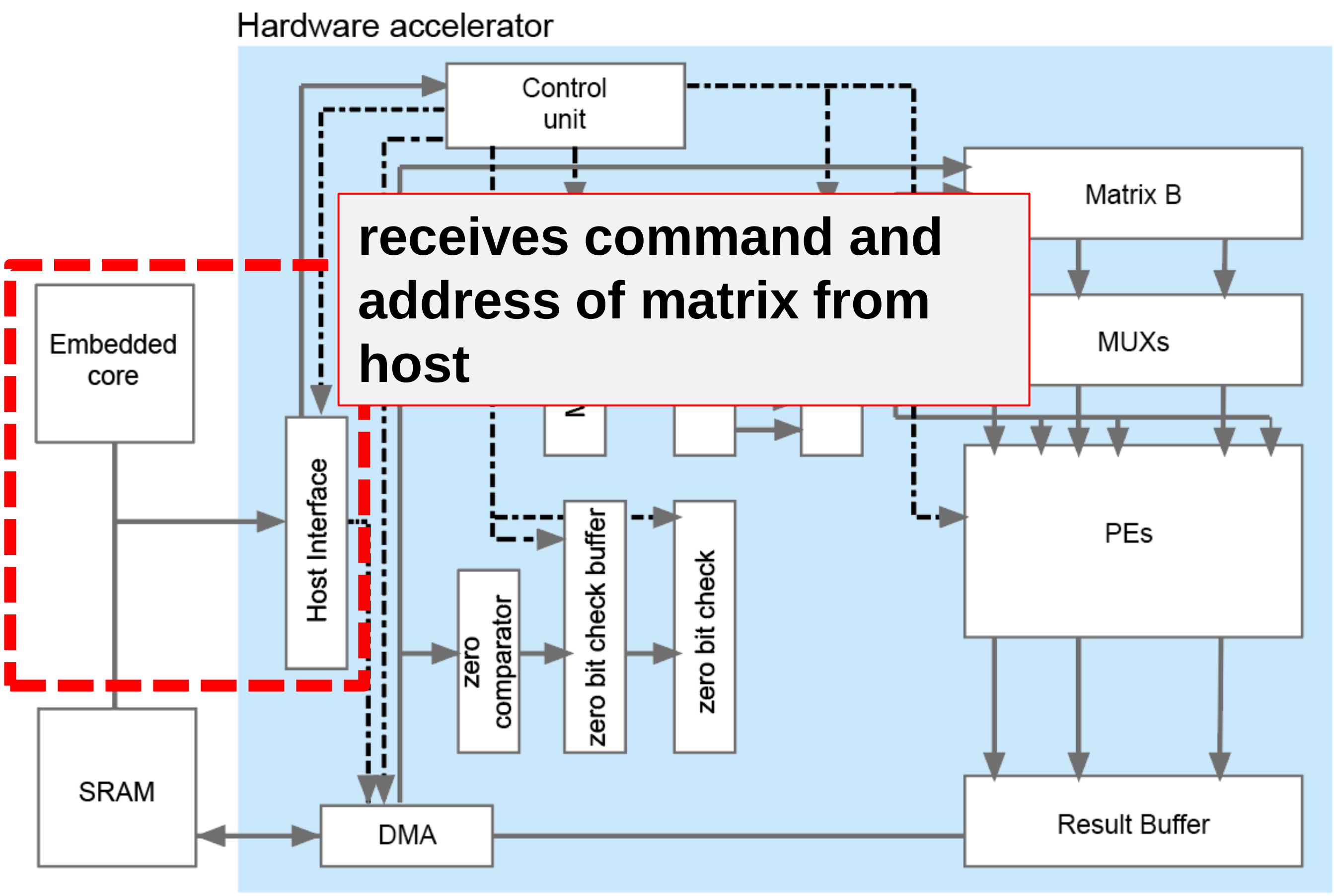
Overall Architecture of IBE

- IBE architecture
  - Processing Element(PE)s, input matrix buffers and control unit



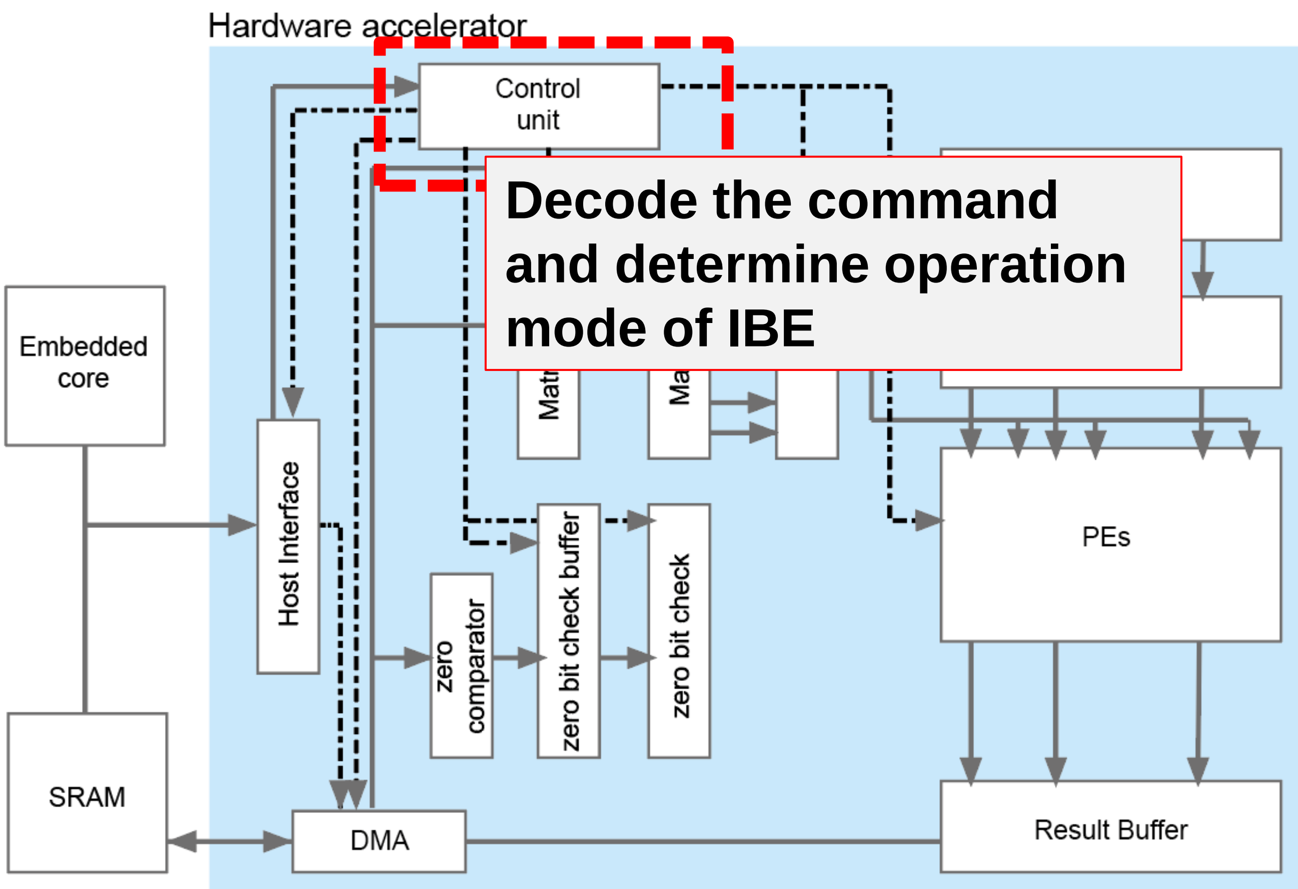
Overall Architecture of IBE

- IBE architecture
  - Host interface of IBE



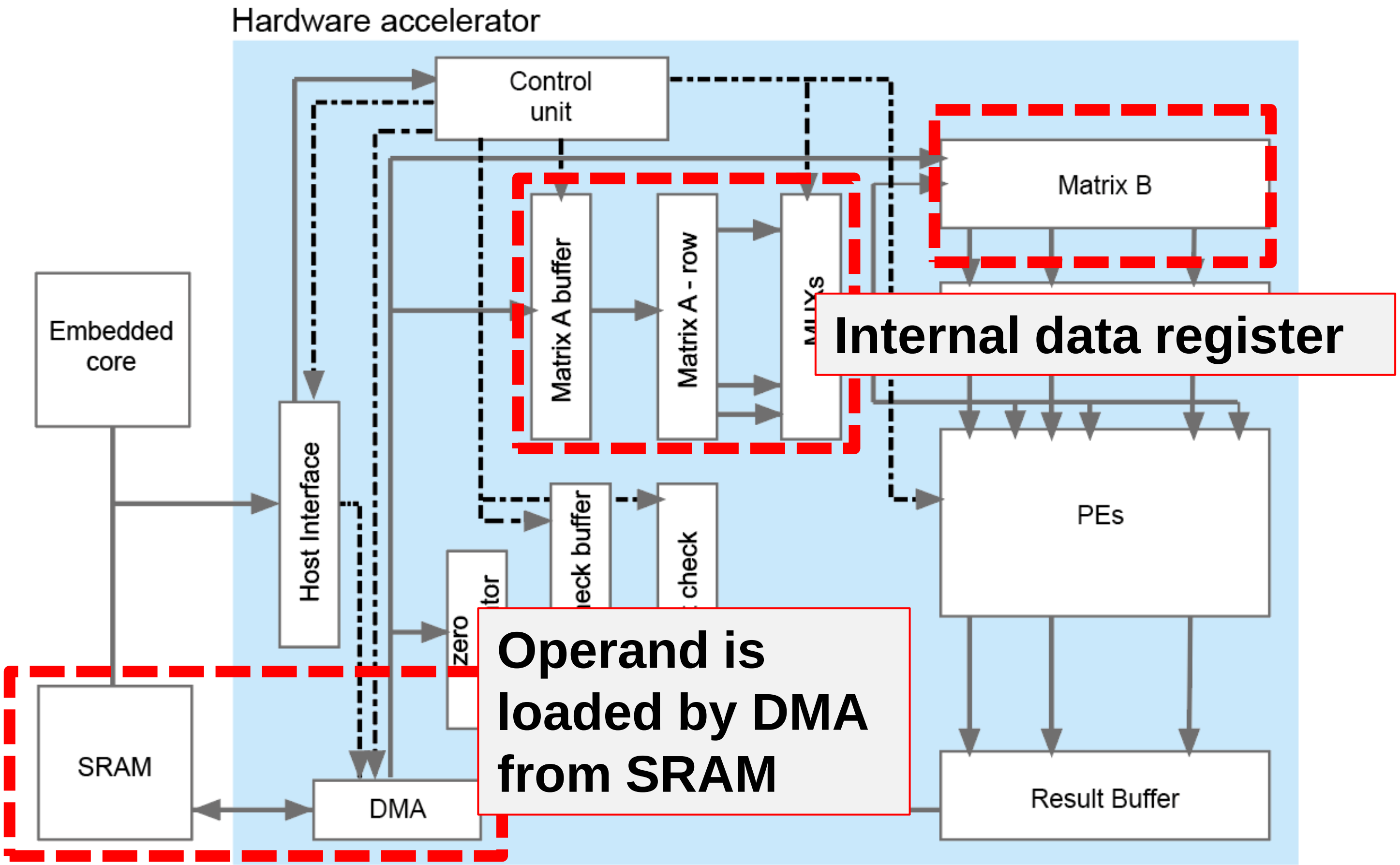
Overall Architecture of IBE

- IBE architecture
  - Control unit of IBE



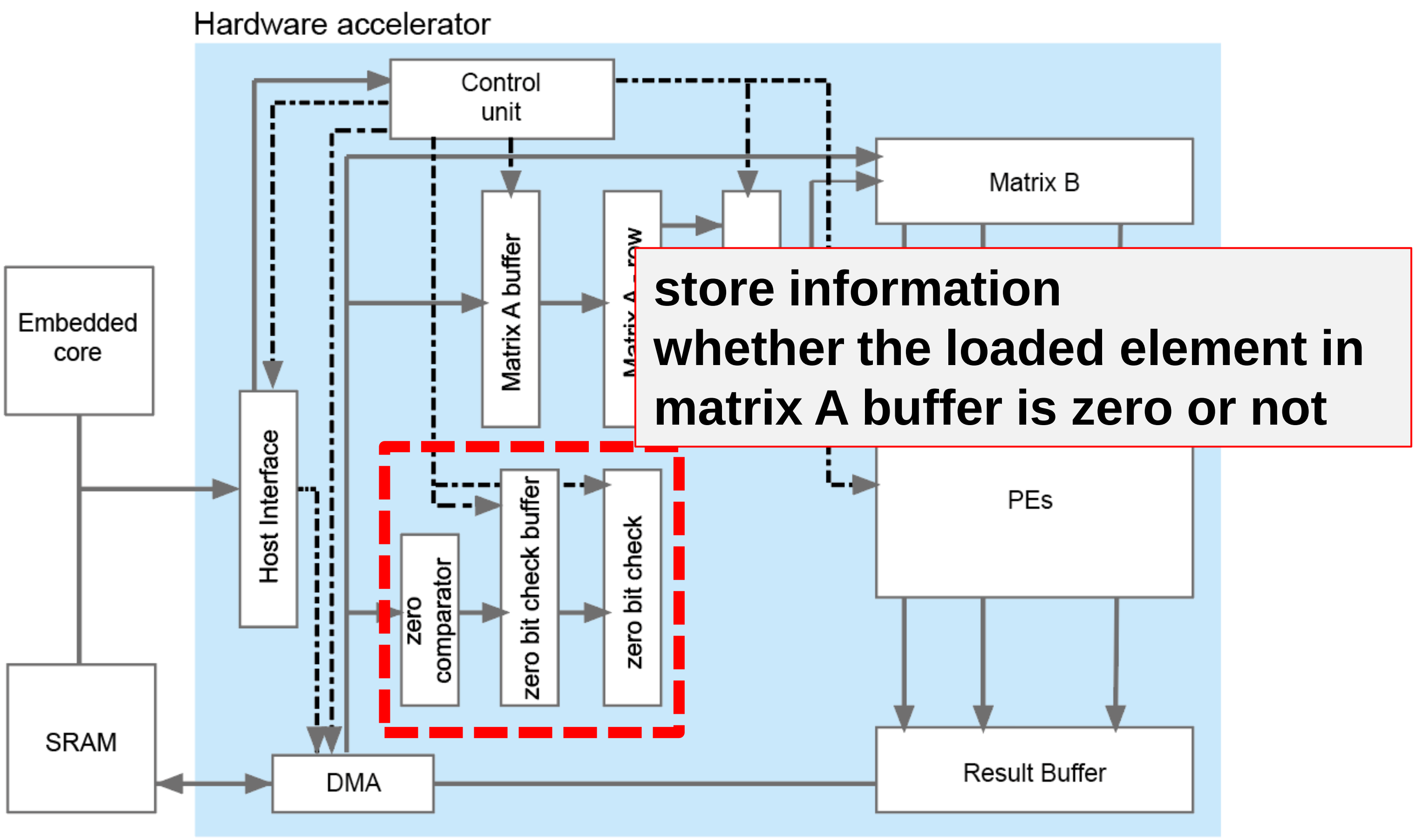
Overall Architecture of IBE

- IBE architecture
  - DMA and internal registers



System Architecture - IBE

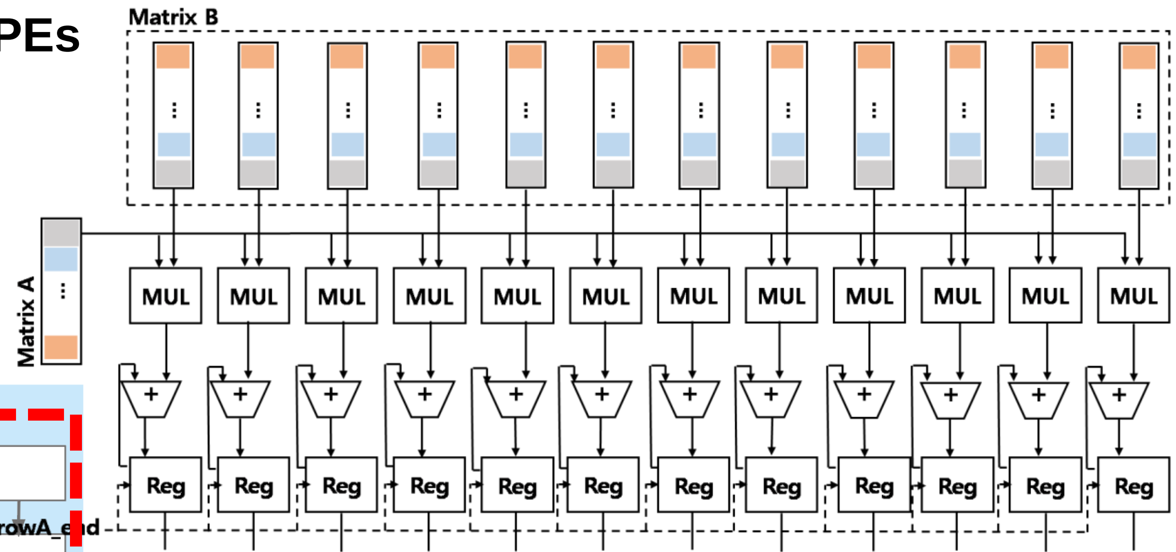
- IBE architecture
  - Zero bit check buffer for skipping kernel operation of Sensor fusion algorithm

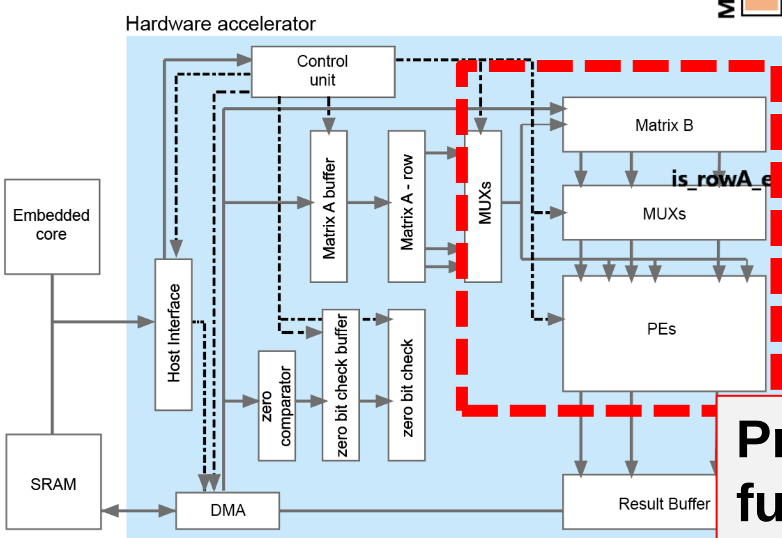


Overall Architecture of IBE

- IBE architecture
  - Processing elements (PEs)
  - Consists of 12 MACs

PEs

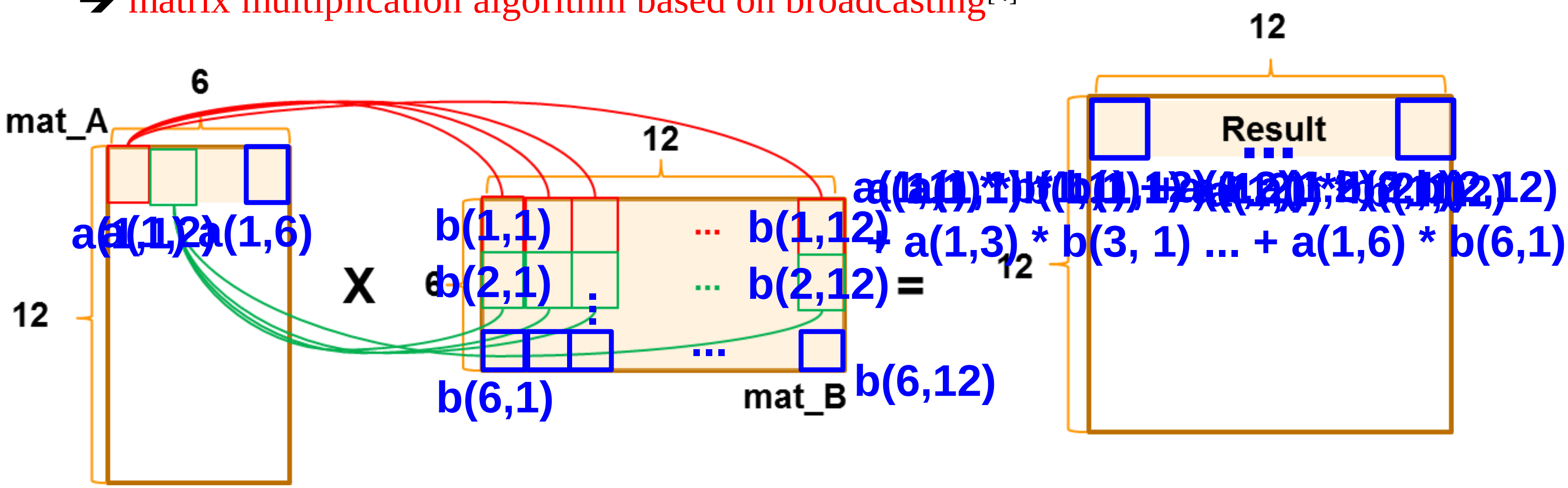




Process kernel of Sensor fusion, SVM, KNN, etc.

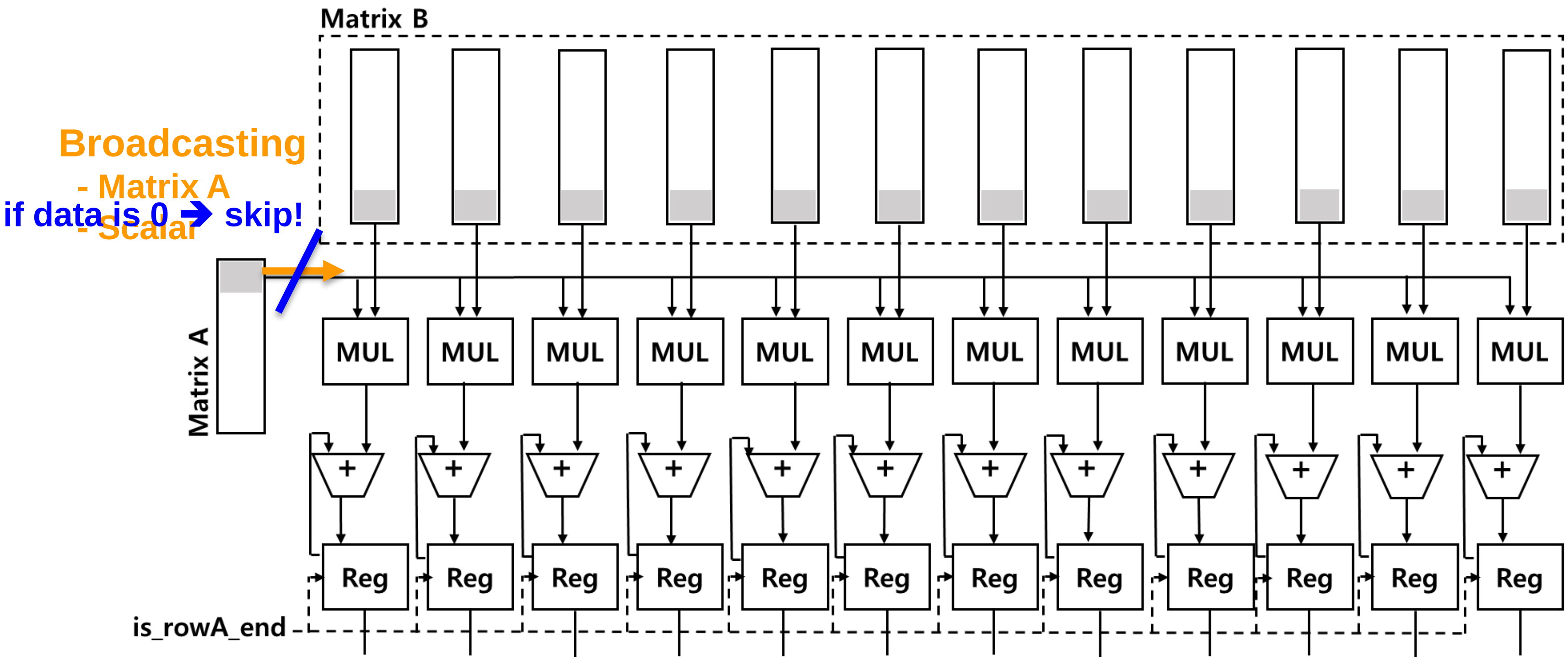
Matrix multiplication in IBE

- Broadcasting based matrix multiplication algorithm
  - Consider various matrix multiplication algorithms to support sparse matrix multiplication and simplicity of HW
  - Zero skipping algorithm
  - Vector * vector operation
  - Scalar * vector operation
  - ➔ matrix multiplication algorithm based on broadcasting[4]



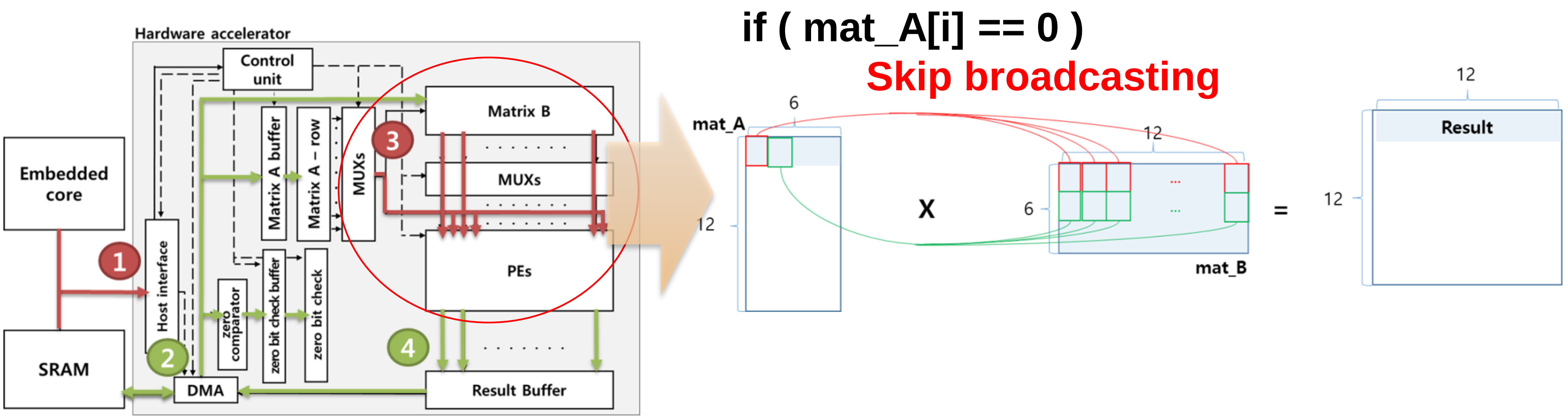
Zero skipping and scalar * vector operation in IBE

- Zero skipping and vector * scalar operation with broadcasting based multiplication HW
  - Zero skipping can be implemented easily by preventing the broadcasting the element
  - Supporting scalar * vector by broadcasting the scalar value



IBE operations – Sensor fusion algorithm

- PE arrays with 12 MACs for matrix multiplication in sensor fusion
  - Optimize the hardware resources based on 6 or 9 axis sensor fusion algorithm
    - Can support 12 x 12 matrix-matrix multiplication (matrix size is up to 12x12 in Sensor fusion)
  - Using zero skipping to reduce the unnecessary multiplication of sparse matrix
    - Zero value in operand matrix A : average 56% in sensor fusion application
- Support matrix transpose



Final IBE operation modes with SVM/KNN operations

- Mode 0~2 : Sensor fusion algorithm & general matrix operations
- Mode 3~4 : general vector operations
- Mode 5~6: SVM/KNN specific operation modes (perform a series of operations for executing SVM/KNN kernel operation)

Mode #	Operation
0	Matrix A * Matrix B
1	Matrix A^T
2	Matrix A * Matrix B^T (SVM linear mode)
3	Vector * Scalar
4	R+=Vector1 * Vector2
5	SVM polynomial mode
6	KNN mode

SVM polynomial mode operations
R1+= Vector1 * Vector2
R2 = Scalar1 * R1 + Scalar2
R3 = $R2^{Scalar3}$

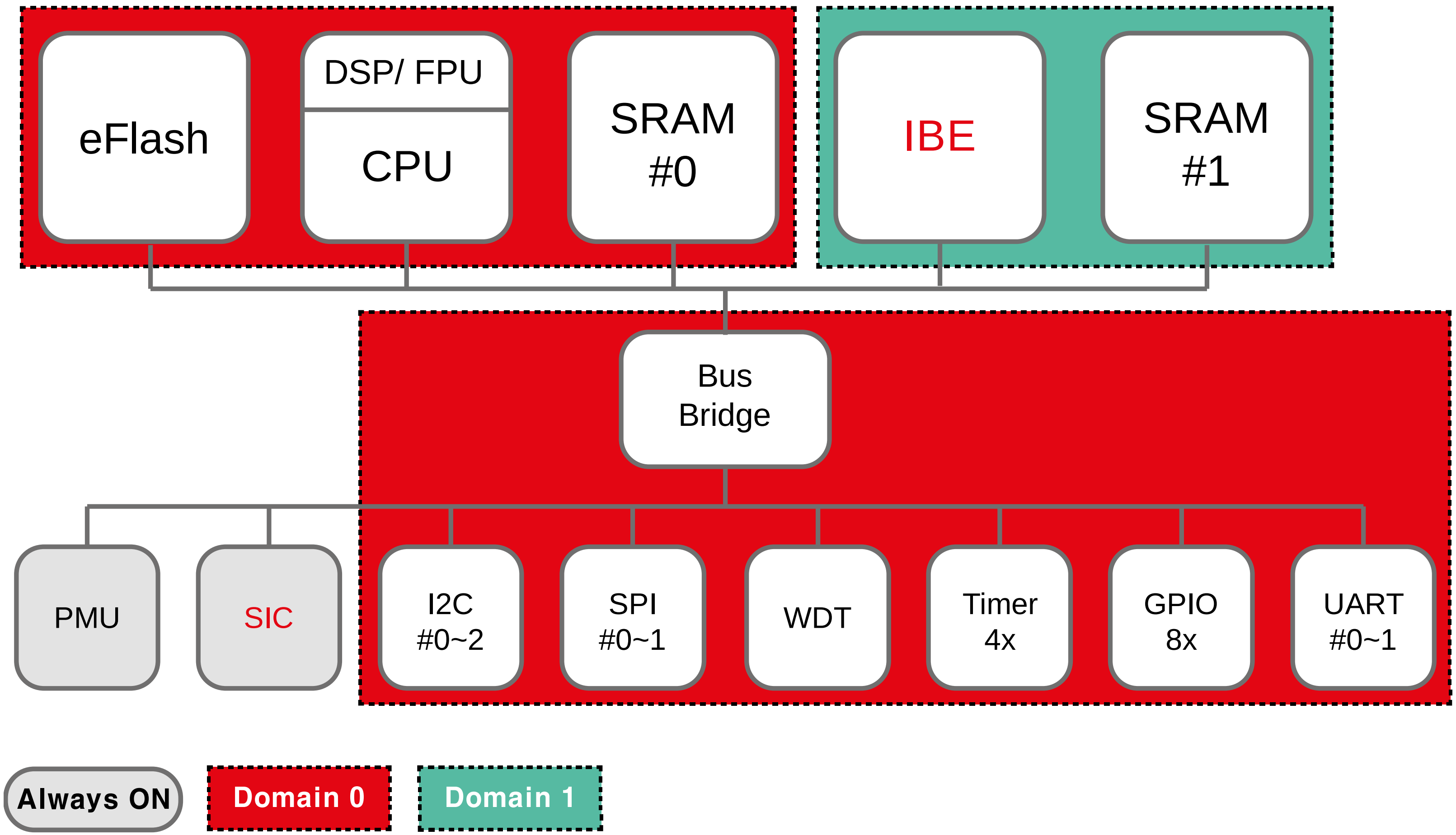
KNN mode operations
R1 = Vector1 – Vector 2
R2 += $R1^2$

Performance Improvement of IBE

- Speedup of each mode

Mode #	Operation	Baseline	IBE	Speedup
0	Matrix A * Matrix B	17485	821	21.30
1	Matrix A^T	1869	534	3.5
2	Matrix A * Matrix B^T (SVM linear mode)	17485	842	20.77
3	Vector * Scalar	2339	579	4.04
4	R+=Vector1 * Vector2	2131	592	3.60
5	SVM polynomial mode	3490	658	5.30
6	KNN mode	3751	620	6.05

Power Domain of SLH-200



Power Modes

- Define 5 power modes
  - User can control all individual IP clock gating
  - User can control power of different domains

Run Mode	Description	CPU wakeup	PMU clock control	PMU power control
Normal mode	CPU normal operation	N/A	N/A	eFlash
Low-power mode	CPU sleep mode SIC monitoring enable	Exception, Ext. Event, SIC	CPU, SRAM#0	eFlash/ IBE/SRAM#1
Sleep mode	CPU sleep mode SIC stand-by	Exception, Ext. Event, SIC	CPU, SRAM #0/#1	eFlash/ IBE/SRAM#1
Down-active mode	CPU not active SIC monitoring enable	SIC (reboot)	N/A	eFlash/ IBE/ CPU/SRAM#0,1
Power-down mode	CPU not active SIC stand-by	SIC (reboot)	N/A	eFlash/ IBE/ CPU/SRAM#0,1

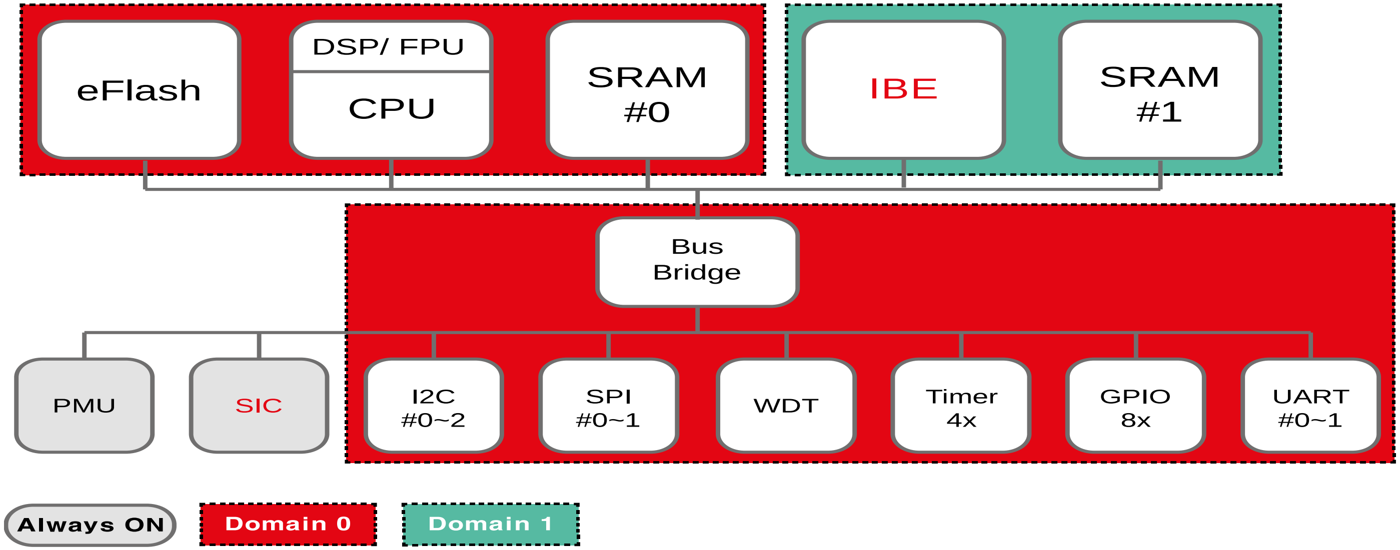
Power Modes

- 5 different power modes
  - Normal mode
    - Normal operation of SoC for sensor fusion
  - Low-power mode
    - Sleep mode in CPU, Active in SIC for sensor monitoring
    - Applies partial clock-gating and partial domain power-gating
  - Sleep mode
    - Sleep mode in CPU, Stand-by in SIC for (SIC) command receiving
    - Applies partial clock-gating and partial domain power-gating
  - Down-active mode
    - Switch off in CPU, Active in SOC for sensor monitoring
    - Applies whole domain power-gating
  - Power-down mode
    - Switch off in CPU, Stand-by in SIC for (SIC) command receiving
    - Applies whole domain power-gating and retention mode in SIC SRAM

Objective of Sensor Interface Controller (SIC)

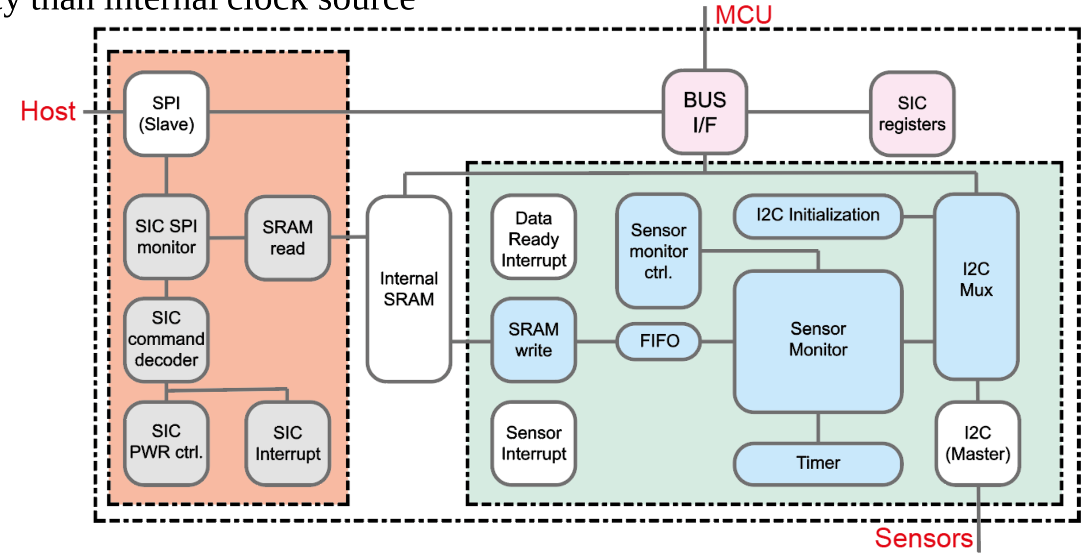
- Objectives
  - Minimize power consumption when CPU block is not used for sensor fusion
  - Just for sensor data monitoring case, most of H/W blocks is not required except CPU and some peripheral (SPI or I2C)
  - However, the CPU with DSP/FPU consumes lot of energy
    - Most common design choice would be dual CPU (ex. Big-Little configuration)

→ Use a specialized hard-wired state machine base architecture, SIC (Sensor Interface Controller) for sensor monitoring

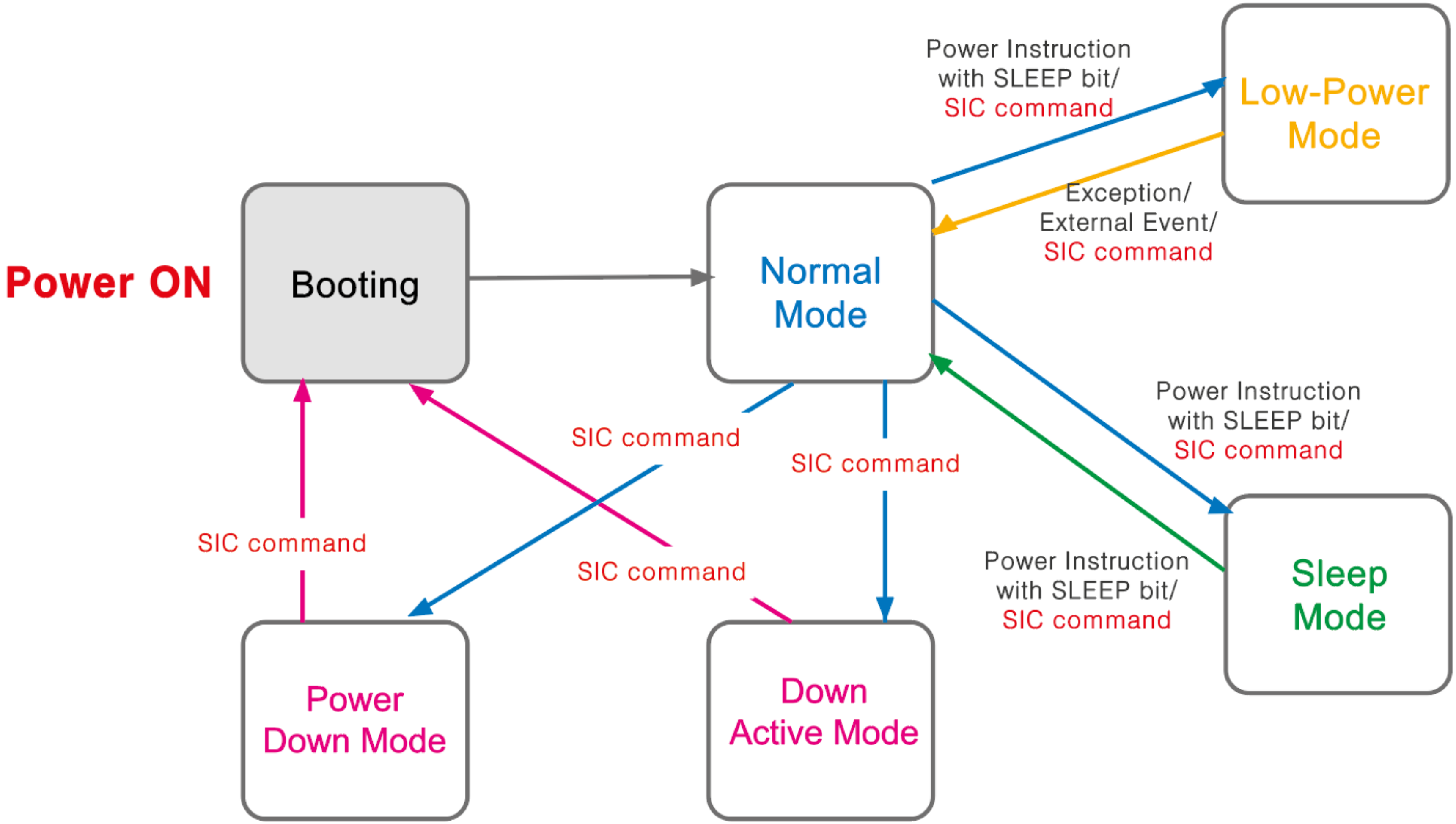


Sensor Interface Controller (SIC)

- Sensor Interface Controller™ (SIC)
  - A dedicated sensor data monitoring H/W block instead of CPU
  - Can monitors sensors via internal I2C channel
  - Can communicate with HOST CPU via internal SPI channel and with Internal MCU via Internal SRAM
  - Includes independent SRAM for sensor data storage
  - Provides SIC command for sensor monitoring, sensor data transfer, SIC control, and power control of SoC
    - Host CPU should control the running mode of sensor hub
  - User can switch off the power of CPU, SRAM, and IBE if not necessary
  - User can change clock source of SIC with external clock source – usually, has lower clock frequency than internal clock source



Run Mode Switching



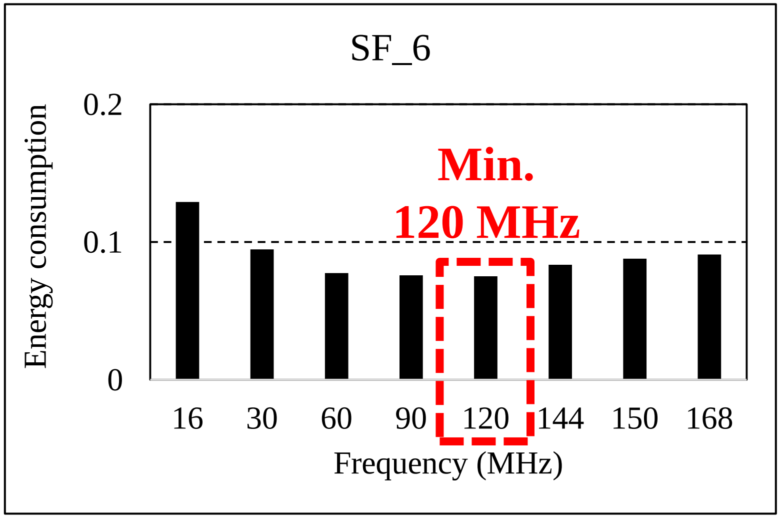
Power management scheme for SLH-200

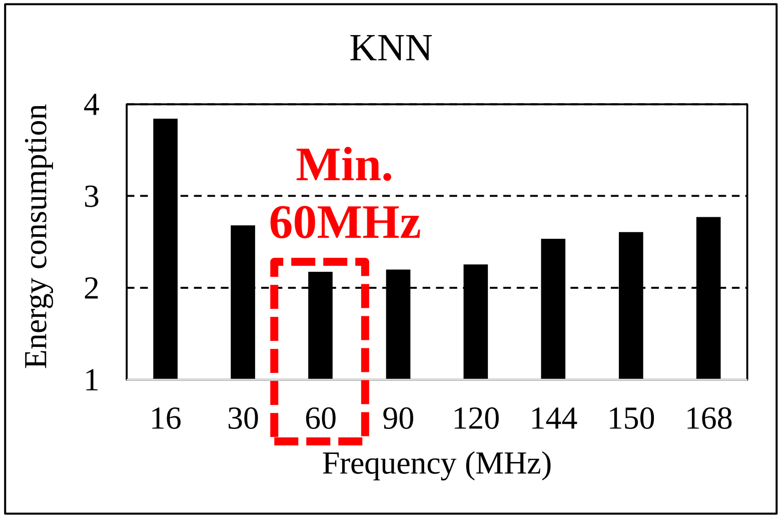
- Effectiveness of Conventional DVFS schemes for a sensor hub
  - Conventional DVFS scheme like Interval-based DVFS[5][6][7] and Task-based DVFS[8][9] , are not suitable for sensor hub
    - Keeping deadline is most important in a sensor hub
    - Sensor hub applications tend to have very short execution time and long deadline
      - Increasing execution time too much based on the deadline can also increase energy consumption

→ A simple profiling-based PM scheme is proposed.

Profiling-based power management scheme for SLH-200

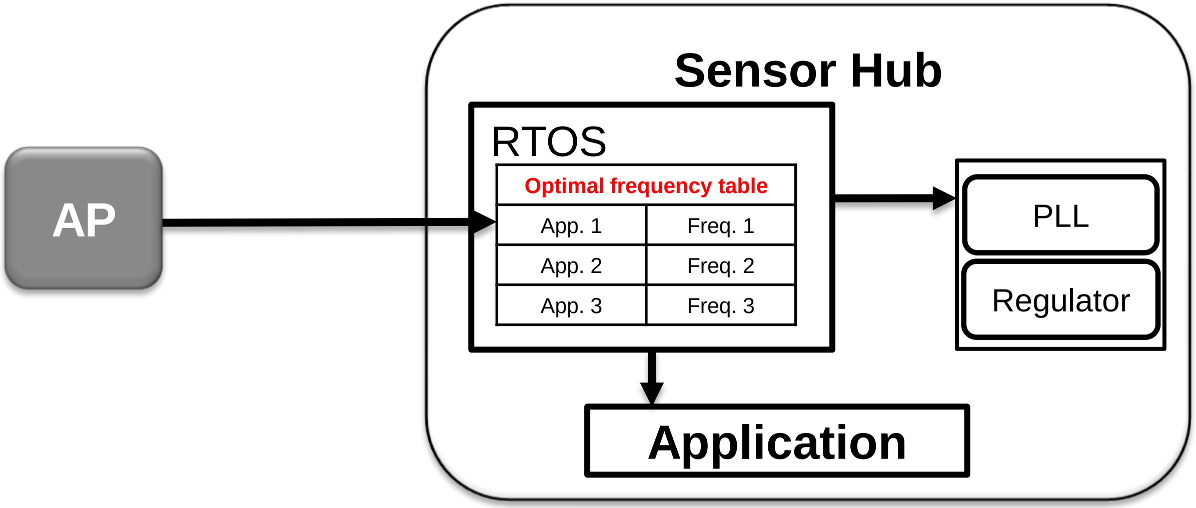
- Profiling-based power management scheme for SLH-200
  - Sensor hub performs pre-defined set of programs/applications
  - Profiling information about the target applications can be effectively used for power management scheme
  - Using profiling, we try to find the optimal operating frequency that can minimize the energy consumption of the applications in the target sensor hub SoC





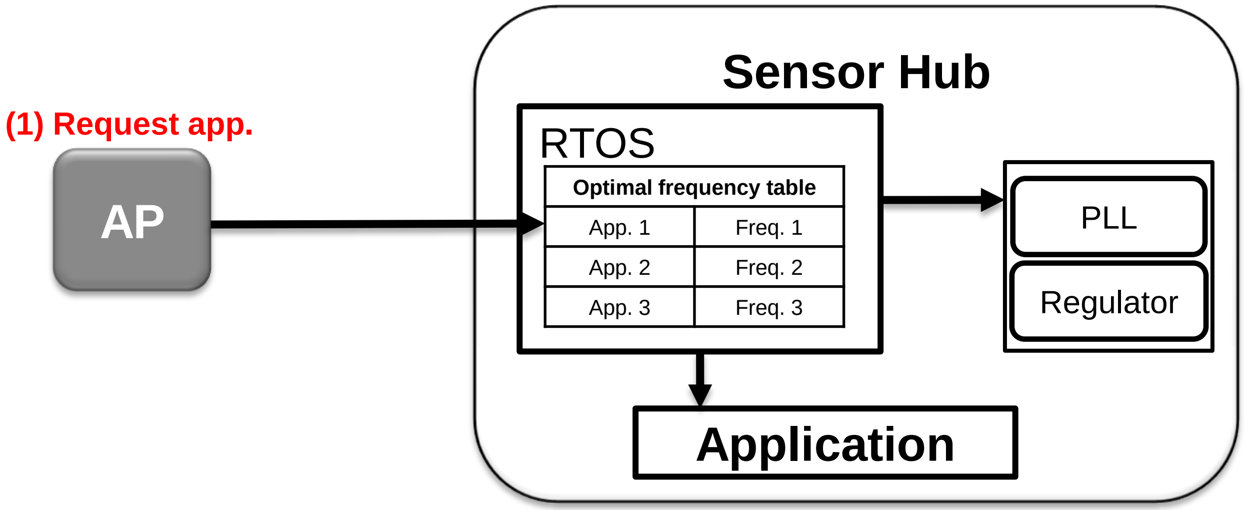
Performing profiling-based power management scheme

- Profiling-based power management scheme for SLH-200
  - RTOS stores the optimal operating frequency for each application that will be executed in Sensor hub in the optimal frequency table



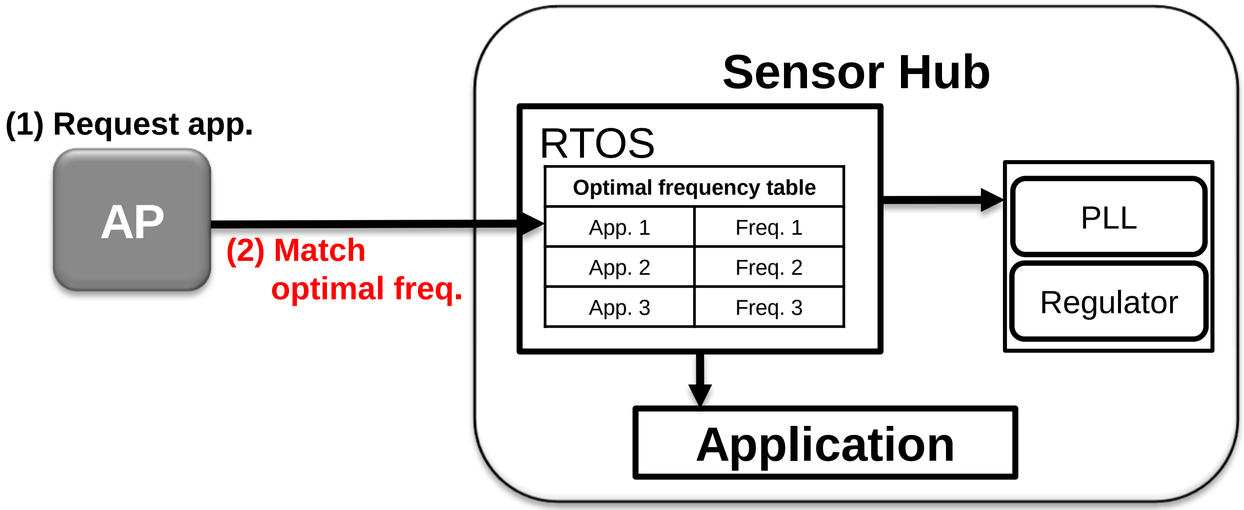
Performing profiling-based power management scheme

- Profiling-based power management scheme for SLH-200
  - RTOS stores the optimal operating frequency for each application that will be executed in Sensor hub in the optimal frequency table
  1. Main processor request to execute application in a sensor hub



Performing profiling-based power management scheme

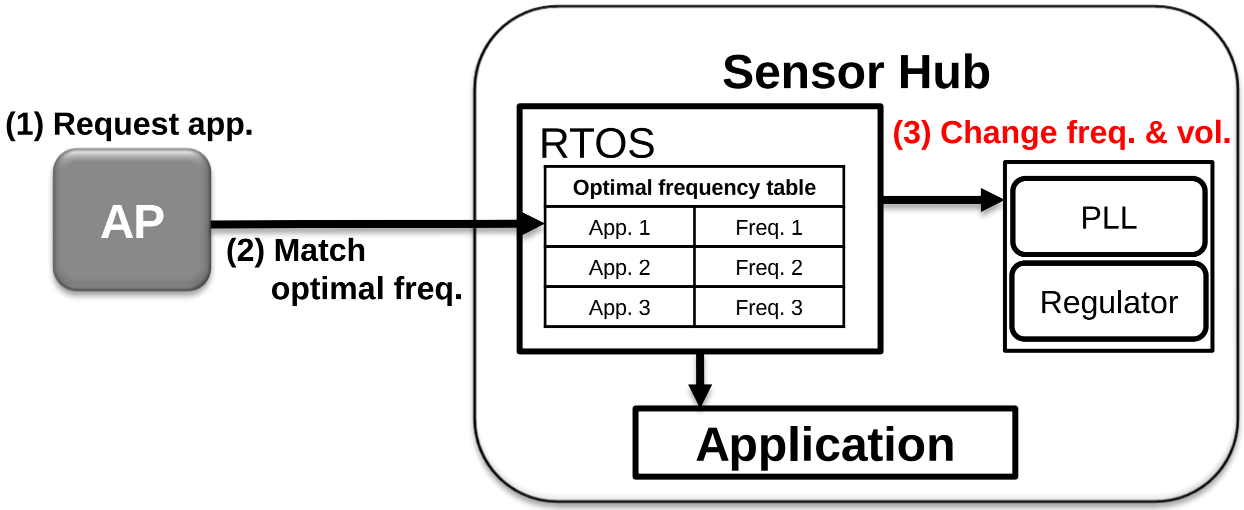
- Profiling-based power management scheme for SLH-200
  - RTOS stores the optimal operating frequency for each application that will be executed in Sensor hub in the optimal frequency table
  1. Main processor request executed application in a sensor hub
  2. Check the optimal frequency for the application in optimal frequency table



Performing profiling-based power management scheme

- Profiling-based power management scheme for SLH-200
  - RTOS stores the optimal operating frequency for each application that will be executed in Sensor hub in the optimal frequency table

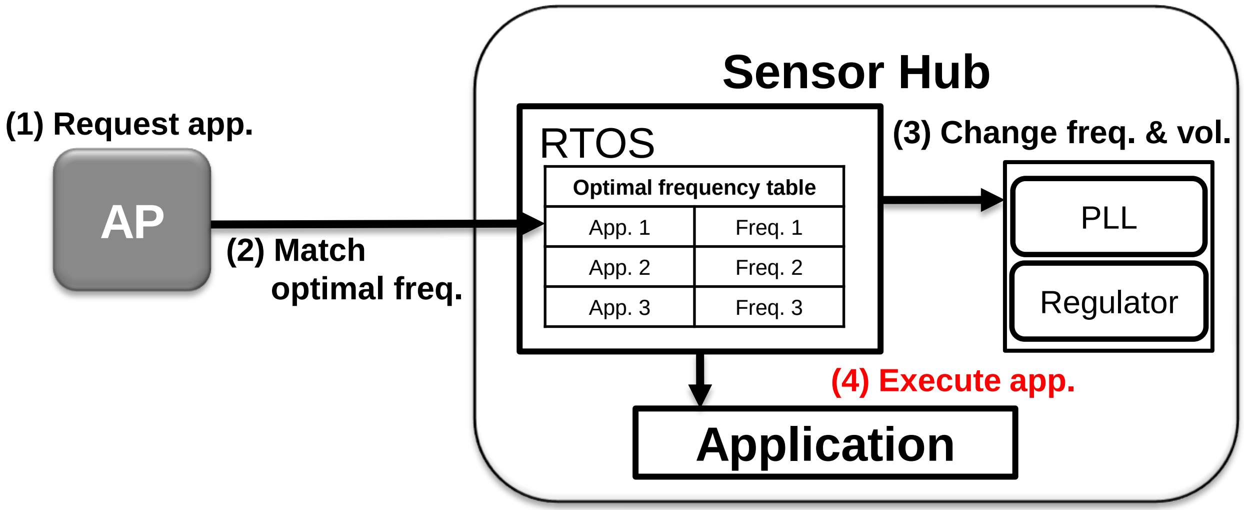
1. Main processor request executed application in a sensor hub
2. Check the optimal frequency for the application in optimal frequency table
3. Change frequency and voltage based on the value of the optimal frequency



Performing profiling-based power management scheme

- Profiling-based power management scheme for SLH-200
  - RTOS stores the optimal operating frequency for each application that will be executed in Sensor hub in the optimal frequency table

1. Main processor request executed application in a sensor hub
2. Check the optimal frequency for the application in optimal frequency table
3. Change frequency and voltage based on the value of the optimal frequency
4. Execute application at the optimal frequency



Applications for power management scheme

- Sensor hub applications for evaluation of PM

Sensor fusion[1]	Machine learning for activity monitoring	Indoor navigation
• Sensor data management • Motion detection	• Motion detection • Activity monitoring	• Context-awareness
Sensor fusion – 6 DOF • Kalman filter • Accel. + Gyro. • Rotation value	**KNN[3]** • Unnecessary learning • Easy to implement	**OpenShoe Project[11]** • Accel. + Gyro. • User location information
Sensor fusion – 9 DOF • Kalman filter • Accel. + Gyro. + Mag. • Orientation value	**SVM[10]** • learning + classification • PC: learning Sensor hub: classification	

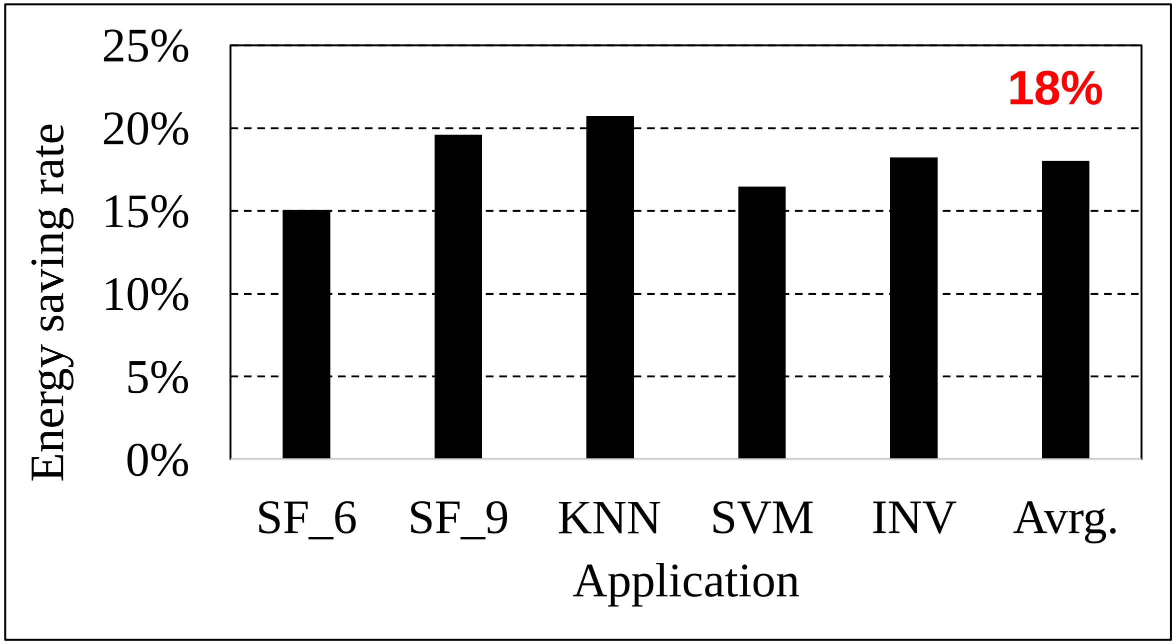
Experiment environments for PM

- STMicro CotexM4F based evaluation board[11]

Core	ARM 32 bit – M4 CPU with FPU
Frequency	Up to 168 MHz
Flash memory	Up to 2 MB
SRAM	Up to 256 KB
Memory accelerator	Instruction prefetch queue Instruction cache memory Data cache memory
Voltage regulator	Power scale1: max 168 MHz Power scale2: max 144 MHz Power scale3: max 120 MHz
Sensor sampling rate	200 Hz (5 ms) Sensor: accelerometer, gyroscope, magnetometer

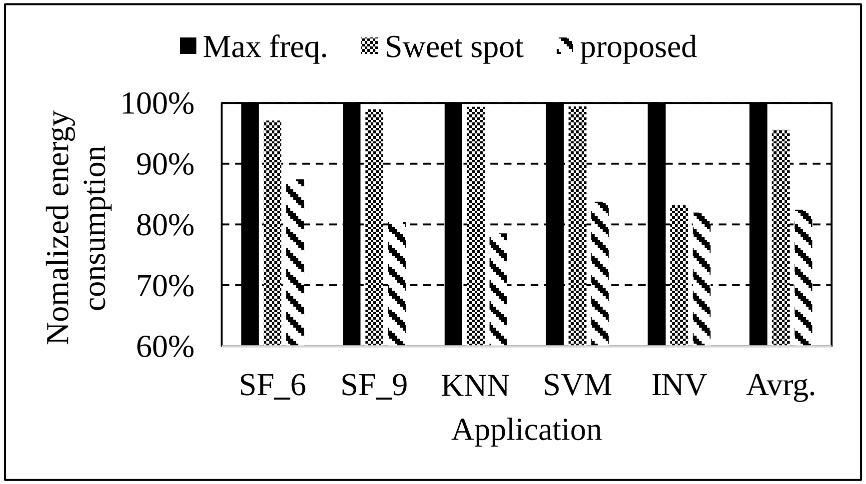
Energy savings of proposed PM scheme

- Energy consumption of proposed scheme
  - Compare energy consumption that applications are executed in max frequency
  - Proposed profiling-based PM reduces about 15% ~ 21% of the energy consumption
  - In KNN, energy consumption in proposed scheme is saved 21%
  - The average energy saving in proposed scheme is about 18%



Comparing with previous PM scheme for sensor hub

- Compare previous DVFS scheme in energy consumptions
  - Sweet spot scheme[9]
    - DVFS scheme for a sensor hub
    - Computation function is executed high frequency(168 MHz), simple function is executed relatively low frequency(120 MHz)
  - Comparing max frequency, sweet spot scheme consumed energy of 96%
  - Proposed scheme consumed energy of 82%
    - Because proposed scheme execute application at optimal frequency for total duration



Application	Computation time (ms)	Acquisition time (ms)
SF_6	0.46 (94.2%)	0.02 (5.8%)
SF_9	1.28 (97.8%)	0.02 (2.2%)
KNN	12.90 (99.2%)	0.10 (0.8%)
SVM	20.38 (99.0%)	0.19 (1.0%)
INV	0.23 (81.1%)	0.05 (18.9%)
Sweet spot	High frequency	Low frequency
Proposed.	**Optimal frequency**	

Voltage/Frequency scaling overheads

- Scaling overhead
  - Proposed scheme just change frequency only once when main processor request the execution of a specific application
  - Proposed scheme don’t change frequency during the execution of the application

→Scaling overhead is very low in proposed PM scheme

  - Sweet spot scheme requires various number of scaling per iteration as shown in the table.
  - For the Indoor navigation application, requires 400 scaling per a second

Application	Sweet spot The number of scaling	Proposed scheme The number of scaling
SF_6	2 per iteration	1
SF_9	2 per iteration	1
KNN	8 per iteration	1
SVM	10 per iteration	1
INV	2 per iteration	1

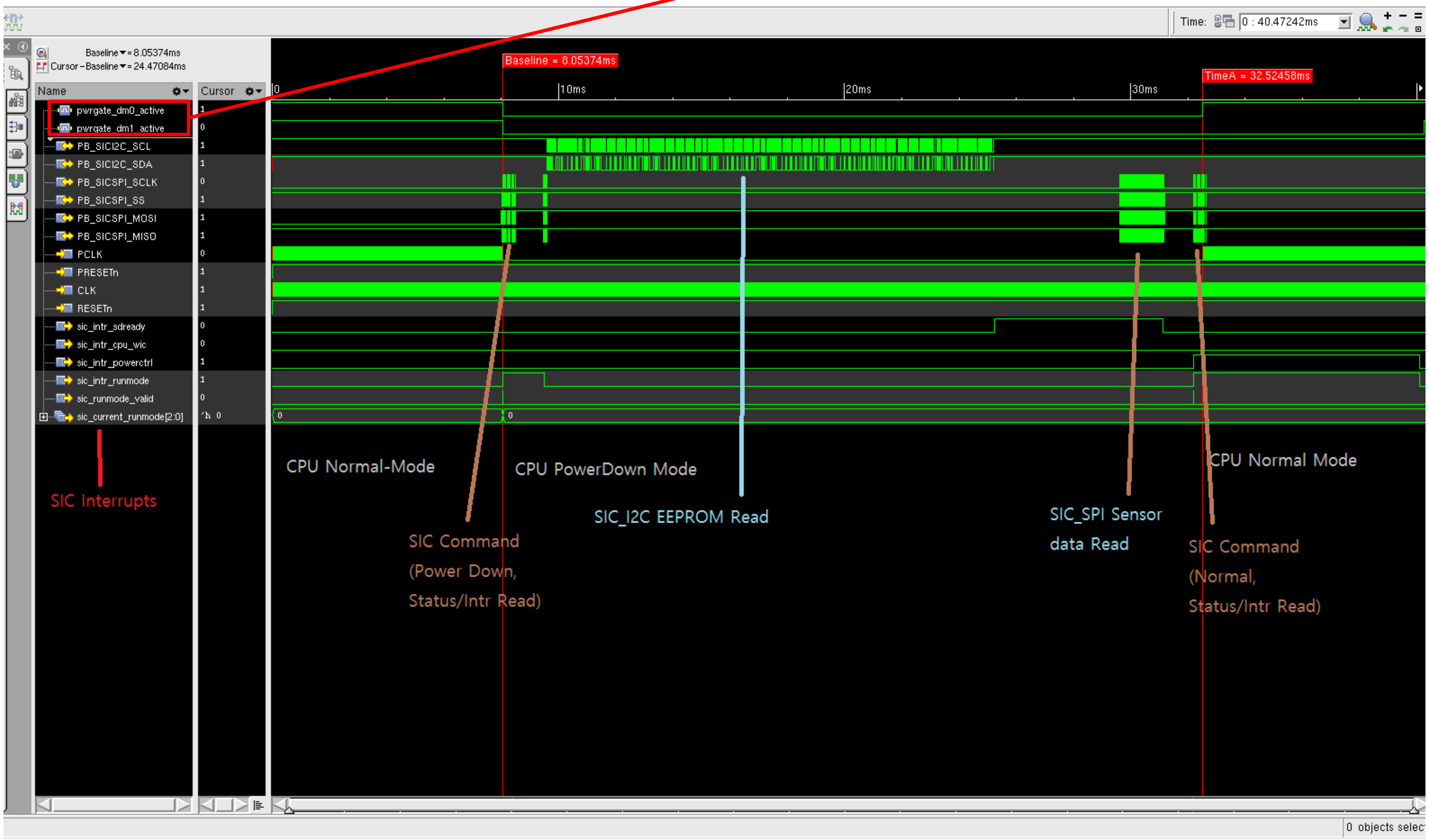
*Function execution frequency
SF_6 & SF_9: 8Hz / KNN & SVM: 1Hz / INV: 200Hz

Functional and power simulation

- RTL simulation
  - Used for individual IP development
- Platform simulation
  - Performs full-chip simulation
  - Simulation environment is composed of CPU/SRAM/embedded FLASH/Bus-matrix/Peripherals/IBE/SIC
  - Basic testing sequence takes 0.5 hours or more
- Timing simulation
  - Performs full-chip simulation with Netlist, SDF file
- Power simulation & power estimation
  - Performs full-chip simulation with Netlist, SDF, and VCD file
  - Only measure a peak power consumption
  - Results is 21.296mA/150MHz, 1.852mA/25MHz (w/o PAD)

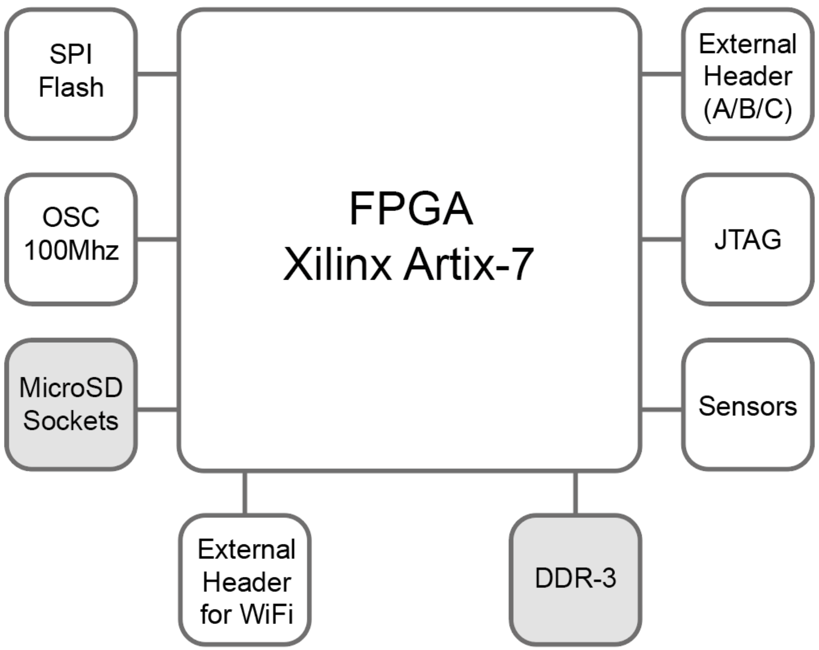
Power control simulation

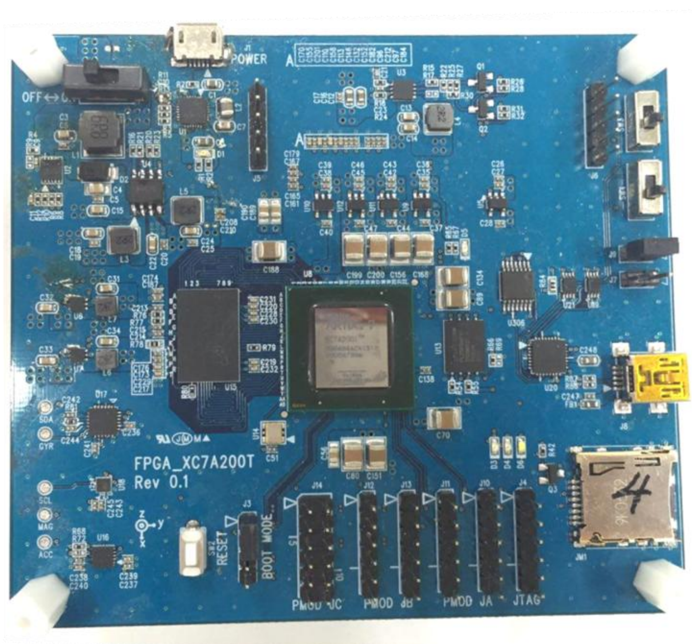
- **Power simulation waveform to perform power control through SIC**



FPGA board based testing

- FPGA testing
  - Main function test
    - CPU/DSP/FPU/SRAM/Flash/Peripherals test
  - SIC/Power control test
    - Sensor data monitoring, Sensor data transfer, & Power control test

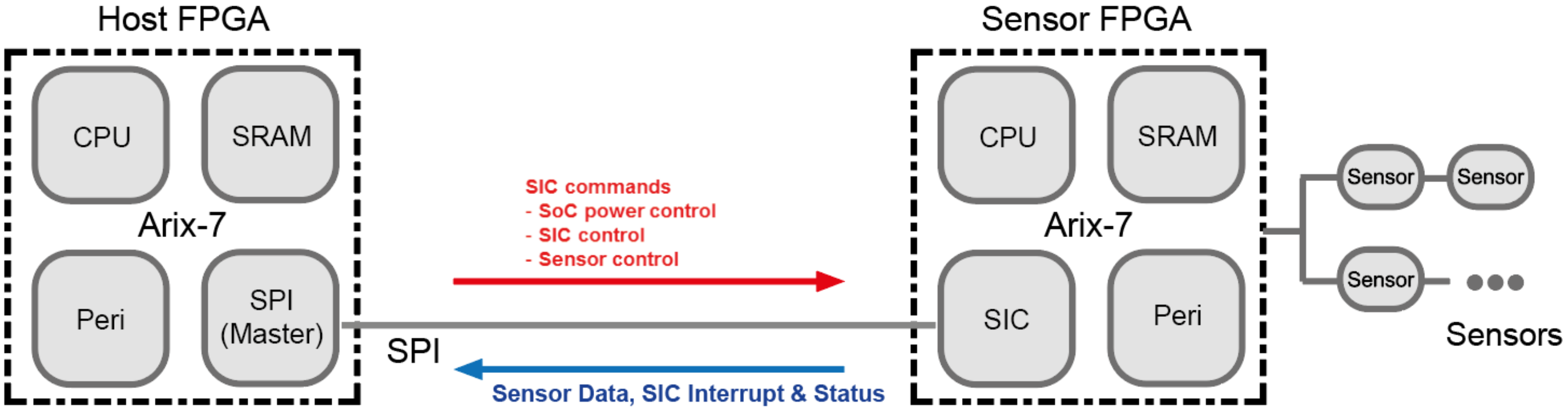




SIC/Power control testing with FPGA board

- FPGA testing for SIC/Power control
  - Two FPGA board : One for host, the other for sensor monitor
    - Sensor FPGA : Sensor monitor/Sensor data read/transfer, SIC command test
    - Host FPGA : SIC command transfer via SPI, Sensor data receive test
      - SPI read/write, SIC commands (run-mode, read sensor data, sic interrupt status, and etc.)

FPGA Board Specification		
FPGA	Xilinx Artix-7 XC7A200T	
DDR-3	Micron MT41K256M16HA	NOT USE
bit download	Spansion S25FL256SAGNFI00	
Sensors	SGA-100/ AK8975C/ ITG-3200	



Power simulation results

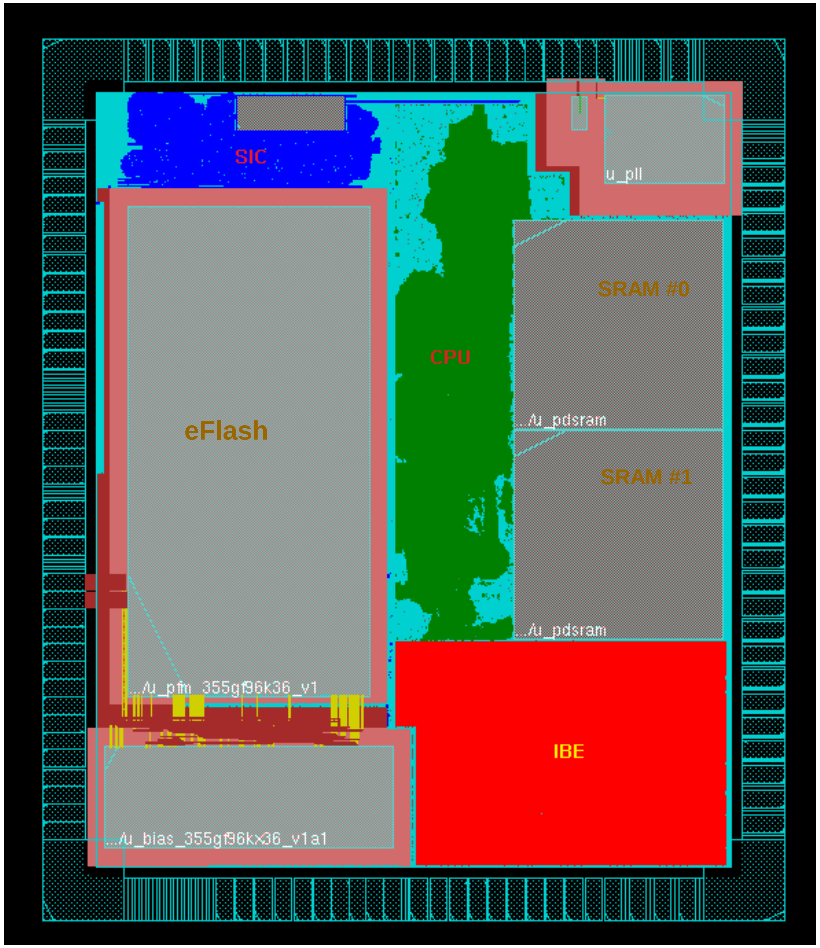
* Library Condition: SS/1.08V/125C

power-up (CPU @ 150MHz)		slh200	func	ibe	sic	pmu	pad
Power (mW)	Internal	15.100	10.400	3.220	1.570	0.050	2.700
	Switching	19.200	8.080	2.850	0.186	0.027	8.820
	Leakage	0.330	0.272	0.111	0.017	0.000	0.036
	Total	34.600	18.800	6.180	1.780	0.077	11.600
Current (mA)	Internal	12.391	9.630	2.981	1.454	0.046	0.909
	Switching	12.581	7.481	2.639	0.172	0.025	2.970
	Leakage	0.284	0.252	0.103	0.016	0.000	0.012
	Total	25.202	17.407	5.722	1.648	0.071	3.906

power-down (SIC @ 25MHz)		slh200	func	ibe	sic	pmu	pad
Power (mW)	Internal	3.010	0.000	0.000	0.198	0.032	2.710
	Switching	10.300	0.000	0.000	0.019	0.020	8.870
	Leakage	0.321	0.262	0.109	0.018	0.000	0.037
	Total	13.600	0.262	0.109	0.234	0.053	11.600
Current (mA)	Internal	1.190	0.000	0.000	0.183	0.030	0.912
	Switching	4.311	0.000	0.000	0.017	0.018	2.987
	Leakage	0.275	0.243	0.101	0.016	0.000	0.013
	Total	5.758	0.243	0.101	0.217	0.049	3.906

Implementation

*** SLH-200 Layout**

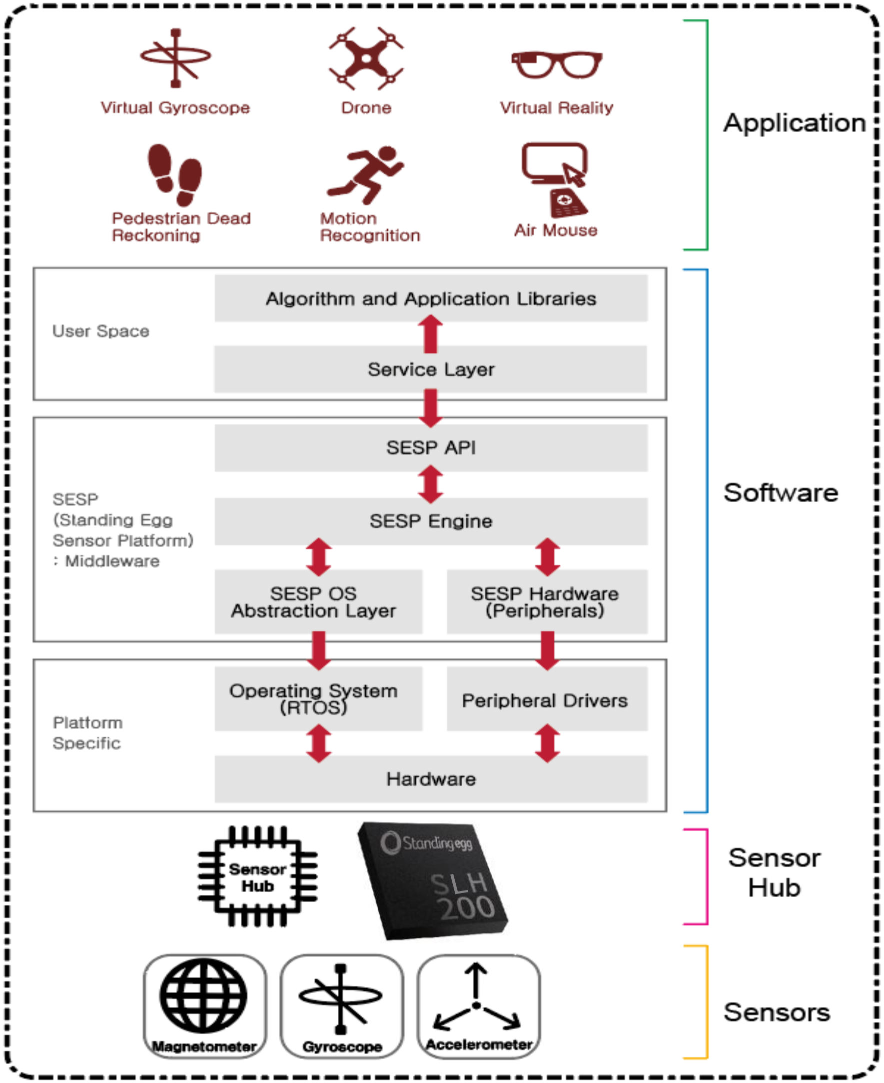


SLH-200 spec.	Description
Process	Global Foundry 55nm LP MPW
Voltages/Packages	Core 1.2V, I/O 3.3V, 80pin QFP
Die size	2112 × 2512 um^2
SRAM #0, #1	About 356,000 um^2 × 2
CPU	About 325,000 um^2 (≈ 301K gates)
IBE	About 490,000 um^2 (≈ 453K gates)
SIC	About 89,000 um^2 (≈ 83K gates with 1KB Dual-port SRAM)
Peripherals	About 74,000 um^2 (≈ 69K gates)
CPU	32bit RISC CPU/DSP/FPU
Max Frequency	150MHz
SRAM	128KB (64KB × 2 SRAM)
Flash	3.4Mb embedded Flash
Peripheral	Two SPI/ Three I2C / Two UART/ 8 GPIO/ WDT/ 4 Timer
SIC	One SPI/One I2C / 1KB Dual-port SRAM

Conclusion and Future Works

- A Low power sensor hub SoC (SLH-200)
  - Intelligence Boost Engine (IBE)
  - Sensor Interface Controller (SIC)
  - Power management scheme for sensor hub
- Sample chip will be available in September, 2016
  - Performance & power-consumption with evaluation board
  - The real application execution and test
  - The effectiveness of IBE & SIC
  - IBE performance such as processing time improvement
  - Power-consumption measurement of different running modes
- Enhancing IBE & SIC
  - Support additional algorithms for sensor hub
  - SIC interrupt notification method to Host CPU
- Power optimization
  - Power-gating on PAD, and etc.
  - Enhancing profiling-based PM with SIC

Future Works – Sensor Platform



Key Feature

- **Applications:**
  - Virtual gyroscope
  - Drone
  - Virtual reality
  - Pedestrian dead reckoning
  - Motion recognition
  - Air mouse
- **Software:**
  - System and software level customizability
  - SESP – Standing Egg Sensor Platform (Middleware)
- **Sensor hub: SLH200**
  - ARM cortex M4F
  - SIC + IBE
- **3 Sensors in one package:**
  - tri-axial 14-bit accelerometer
  - tri-axial 16-bit gyroscope
  - tri-axial 16-bit magnetometer

Reference

[1] NXP Semiconductor, "Sensor Fusion Library for Kinetis MCUs," Data Sheet, Rev 5.0, 2015. 9

[2] Chih-Chung Chang and Chih-Jen Lin, LIBSVM – A Library for Support Vector Machines, https://www.csie.ntu.edu.tw/~cjlin/libsvm/

[3] Standing Egg Inc., http://standing-egg.co.kr

[4] Vinay B. Y. Kumar, Siddharth Joshi, Sachin B. Patkar et al., "FPGA Based High Performance Double-Precision Matrix Multiplication," 22nd International Conference on VLSI Design, 2015. 6

[5] J. R. Lorch, et al, "Operating System Modifications for Task-Based Speed and Voltage Scheduling," the 1st International Conference on Mobile Systems, Applications, and Services, pp. 215-229, 2003.

[6] M. Weiser, et al, "Scheduling for reduced CPU energy," International Symposium on Operating Systems Design and Implementation, pp. 13-23, 1994.

[7] K. Govil, et al, "Comparing Algorithms for Dynamic Speed-Setting of a Low-Power CPU," International Conference on Mobile Computing and Networking (MOBICOM), pp. 13-25, 1995.

[8] K. Flautner, et al, "Automatic Performance Setting for Dynamic Voltage Scaling," International conference on Mobile Computing and Networking (MOBICOM), pp. 260-271, 2001.

[9] A. Fuks, "Sensor-Hub Sweet-Spot Analysis for Ultra-Low-Power Always-on Operation," VLSI Circuits, pp. C154-C155, 2015.

[10] SVM-light, http://svmlight.joachims.org

[11] OpenShoe, http://openshoe.org

[12] STMicroelectronics, "STM32F439ZI," Data Sheet, Rev 9, 2016.

Acknowledgements

Thanks to Jong Sam Park, Kwang Jin Kim, Seulgi Oh, Gustavo Adrian Ruiz Sanchez, Michael Vogl (Standing Egg Inc.) and Hong-yeol Lim, Seungjin Lee (Sejong University) for their effort for this project.

This work was supported by the Technology Innovation Program, 10049503, “Development of a low-power sensor processing MCU that consumes under 146uA/MHz for mobile smart device” funded By the Ministry of Trade, industry & Energy (MI, Korea).

Keit

한국산업기술평가관리원

Korea Evaluation Institute of Industrial Technology