

A Processor Architecture for Motion Sensing Systems using Accelerometer

Takashi MATSUO , Arif Ullah KHAN , Takashi HAMABE*,
Yoshinori TAKEUCHI , and Masaharu IMAI

Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka, 565-0871 JAPAN
{takasi-m,khan,t-hamabe,takeuchi,imai}@ist.osaka-u.ac.jp

Abstract- Microelectromechanical-systems-based accelerometers are widely used in motion sensing, in which coordinate system transform is the major process. In this study, a method for coordinate system transform is introduced, and a low-power processor architecture specialized for coordinate system transform is proposed and evaluated. Through experimental results, the proposed processor reduced energy consumption by 48.3% compared with a conventional RISC processor implementation.

I. INTRODUCTION

In rehabilitation, one of the main tasks of health care providers is checking whether motions of patients are normal or not. While health care providers have attended on patients so far, owing to aging society, the number of patients per one health care provider for medical rehabilitation is expected to increase in the near future. Therefore, it is necessary to develop devices which can assist medical rehabilitation of patients. Recently, motion sensing devices using microelectromechanical-systems (MEMS) based accelerometers have become available. By developing motion sensing devices using MEMS-based accelerometers, medical rehabilitation of patients can be made without health care providers attending on patients. While long operating time of motion sensing devices in daily life can reduce the burdens of health care providers, most of them work for only a short time. Large battery capacity could achieve long operating time. However, motion sensing devices have to be small enough so that they can measure one's motion correctly because big or heavy devices can affect frequency of one's motion which is necessary in motion sensing applications. Thus, hardware in motion sensing systems must be low power consumption.

Motion sensing systems have many processes. In our motion sensing system, step count, moving distance, and energy expenditure are estimated. It is necessary to transform coordinate systems to estimate these correctly, because the sensor coordinate is not necessarily equal to the world coordinate system. And coordinate system transform is the major process in

our motion sensing system. Main contribution of this paper is a proposal of a processor with special instructions for coordinate system transform which can achieve long operating time of motion sensing systems due to its low power consumption.

The rest of this paper is organized as follows. Section II introduces processes in our motion sensing system. Then, the proposed low-power processor is introduced in Section III. Section IV shows experimental results. Finally, Section V concludes this paper and indicates future problems.

II. PROCESSES IN MOTION SENSING SYSTEM USING ACCELEROMETER

This section introduces processes composing our motion sensing system using accelerometer. Processes composing our motion sensing system using accelerometer are follows:

- moving distance estimation
- estimation of the number of steps
- energy expenditure estimation
- coordinate system transform

In this study, we presume that there exists three contexts, that are, types of behaviors related with variance of vertical acceleration: stop, walk, and run.

A. Moving Distance Estimation

Step frequency, a.k.a. cadence, can be obtained by using discrete fourier transform of vertical acceleration [1]. Average speed V [m/s] in a time window of discrete fourier transform is estimated as

$$V = SF, \quad (1)$$

where S [m] is stride length and F [Hz] is cadence of activity. Moving distance D [m] in a time window is estimated as

$$D = VT, \quad (2)$$

where T [s] is a time window size [2].

*At present, MICRONIX Inc.

TABLE I
MET AND ACTIVITIES

MET(s)	Activity
0.9	sleep
1.0	rest
1.2	stand up
3.3	walk(81m/min)
7.0	jog
8.0	cycle
10.0	run(161m/min)

B. Estimation of the Number of Steps

The number of steps in a time window N [steps] is estimated by

$$N = FT. \quad (3)$$

C. Energy Expenditure Estimation

This study uses an energy expenditure estimation method based on Metabolic Equivalent of Task (MET), which is proposed by the American College of Sports Medicine.

C.1. MET

The MET is used as a means of expressing the intensities and energy expenditures of activities, and does not depend on a person's age or sex. One MET is the energy expended at rest, and two METs means the energy expended twice than at rest. Table I shows major activities and the corresponding MET values [3].

C.2. Energy Expenditure Estimation Method Based on MET

Energy expenditure E [kcal] can be estimated by

$$E = C_E MTW, \quad (4)$$

where C_E is a constant 2.9167×10^{-4} and M is a MET value. For example, when a 60 kg person jogs for 15 minutes (900 seconds). Energy the person expended is calculated as follows:

$$2.9167 \times 10^{-4} \times 7.0 \times 900 \times 60 = 110.25(\text{kcal}) \quad (5)$$

Energy expenditure for one day can be estimated by checking type of activity and time. However it is difficult to check a person's activity for a long time. Therefore, it is necessary to estimate the MET by using context information mentioned above.

According to [4], the METs in walking and running is a function of moving speed V [m/s]. The METs in walking and running are calculated by

$$M_w = 0.00453V + 1.2, \quad (6)$$

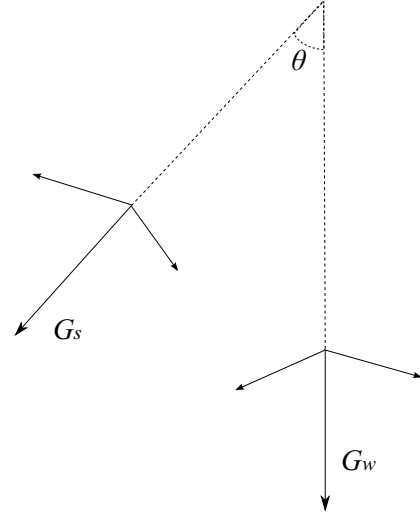


Fig. 1. The Angle Between Two Vectors of Gravitational Acceleration

and

$$M_r = 0.0155V - 4.7, \quad (7)$$

respectively, where M_w and M_r are METs in walking and running.

D. Coordinate System Transform

Since in the practical use of a motion sensor, the sensor coordinate is not necessarily equal to the world coordinate system. Fig. 1 shows relation of the angle between two vectors of gravitational acceleration. When a person is in stop, the vector of gravitational acceleration in the sensor coordinate system G_s is obtained. The angle between the vector of gravitational acceleration in sensor coordinate system G_s and the vector of gravitational acceleration in world coordinate system G_w can be computed by

$$\theta = \arccos \left(\frac{G_s \cdot G_w}{|G_s||G_w|} \right). \quad (8)$$

While one of implementation methods uses an arccosine table in order to obtain the angle θ , the table must be large for supporting high resolution.

However, there is another method which can obtain the angle θ without having a large table. Fig. 2 shows angle decision flow diagram. By calculating inner product of G_s and G_w , $\cos \theta$ is computed. Then, $|\cos \frac{\theta}{2}|$ and $|\sin \frac{\theta}{2}|$ is computed by using half-angle formulas. In order to determine signs of $\cos \frac{\theta}{2}$ and $\sin \frac{\theta}{2}$, the method rotate the vector of gravitational acceleration in the sensor coordinate system G_s using quaternion in four ways. If the rotated vector is equal to the vector of gravitational acceleration in the world coordinate system G_w , the angle θ is specified by determining signs of $\cos \frac{\theta}{2}$ and $\sin \frac{\theta}{2}$. A 3×3 matrix W for transforming sensor coordinate system to world coordinate system can be found by using the angle θ

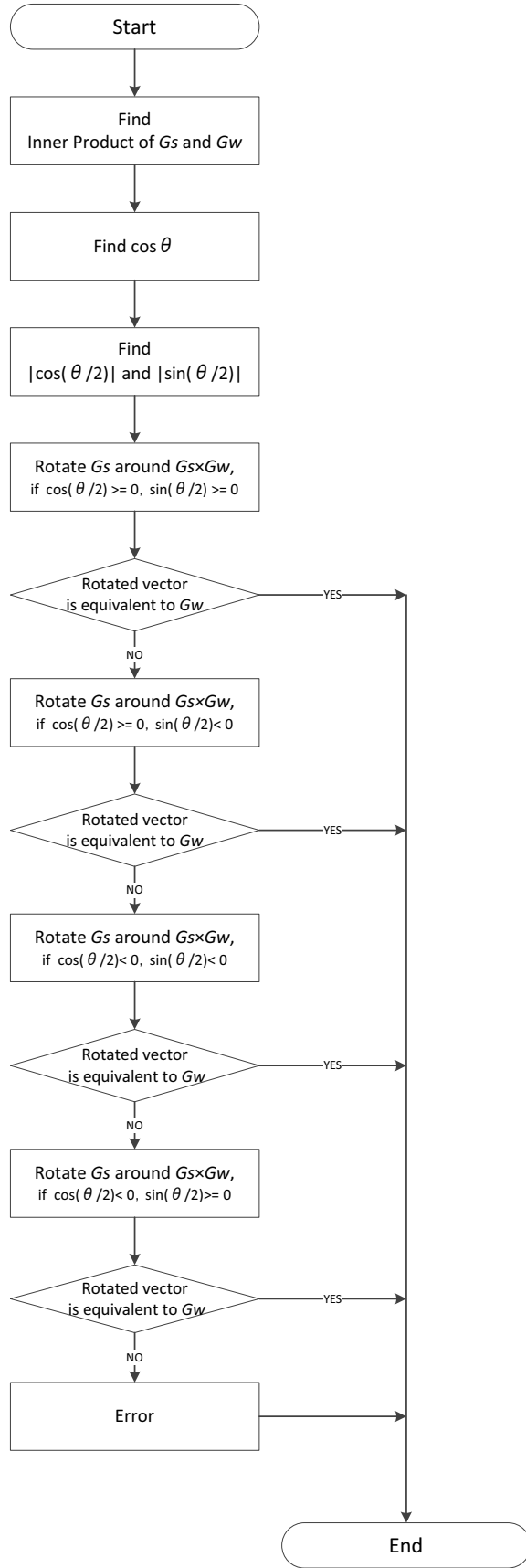


Fig. 2. Angle Decision Flow Diagram

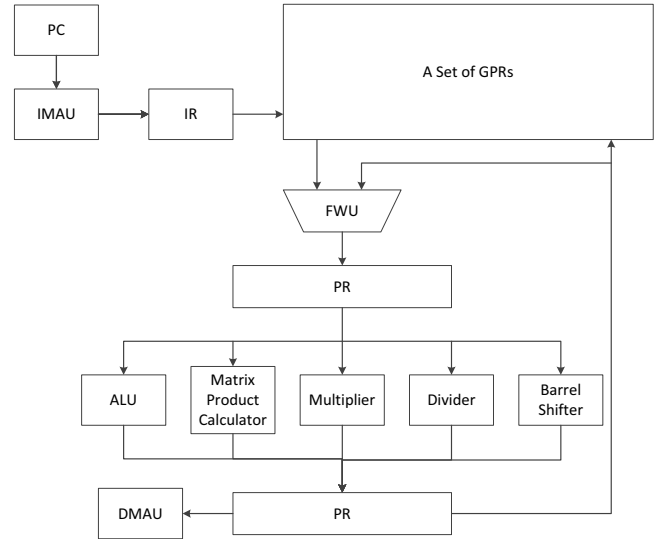


Fig. 3. Architecture of The Proposed Processor

between two vectors of gravitational acceleration G_s and G_w , where G_w is the vector of gravitational acceleration of the world coordinate system [5].

Three coordinates in world coordinate system X_w , Y_w , and Z_w are calculated as

$$\begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} = \mathbf{W} \begin{pmatrix} x_s \\ y_s \\ z_s \end{pmatrix}, \quad (9)$$

where x_s , y_s , and z_s are three coordinates in the sensor coordinate system.

III. PROPOSED PROCESSOR ARCHITECTURE FOR COORDINATE SYSTEM TRANSFORMATION

This section introduces a processor architecture specified for coordinate system transform. The proposed processor is designed by adding specific instructions and hardware to a simple RISC processor, Brownie STD 32 provided by ASIP Solutions, Inc [6]. We designed the processor using an application domain specific instruction set processor development system called ASIP Meister [7].

A. Additional Hardware

Fig. 3 shows the proposed processor architecture. A matrix product calculator specialized for inner products of 1×3 and 3×1 matrices is added to the base processor, Brownie STD 32. Moreover dedicated registers for storing nine coefficients of the matrix $\mathbf{W}$ for coordinate system transform are added.

Fig. 4 shows structure of the matrix product calculator added to Brownie STD 32. It is not possible to store a three dimensional coordinate to one general purpose register (GPR), because resolution of accelerometer is 12-bit. Thus, as shown in

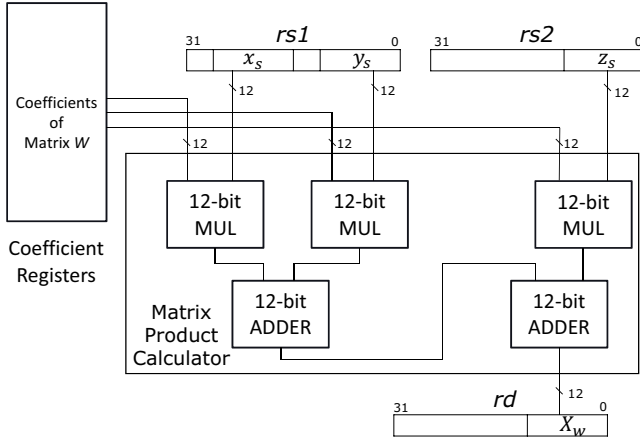


Fig. 4. Matrix Product Calculator

Fig. 4, a three dimensional coordinate of the sensor coordinate system is stored to two GPRs, $rs1$ and $rs2$. Matrix product calculator outputs a coordinate of world coordinate system, by using the following six values: a three dimensional coordinate of the sensor coordinate system stored in $rs1$, $rs2$ and three coefficients of a 1×3 matrix stored in the coefficient registers.

B. Custom Instruction for Coordinate System Transform

This section explains the behavior of the proposed custom instructions for coordinate system transform added to Brownie STD 32.

PACK

The PACK operation joins the lower 16 bits of two registers. The x-coordinate and the y-coordinate in two GPRs can be stored to one GPR.

STREG

The STREG operation stores the lower 12 bits of a register to one of the dedicated registers for matrix coefficients, and the target register is destined by operand. Fig. 5 shows storing a matrix coefficient a to the dedicated register. This operation moves one of the nine coefficients of the matrix W for coordinate system transform stored in GPR to the dedicated registers for matrix coefficients.

MATX

The MATX operation calculates the x-coordinate of world coordinate system. By using the PACK operation, the MATX operation stores the x-coordinate and the y-coordinate of the sensor coordinate system to a GPR, and stores the z-coordinate to the other GPR. In addition to three coordinates of sensor coordinate system stored in a GPR, coefficients of the 3×3 matrix W stored in one of the dedicated registers.

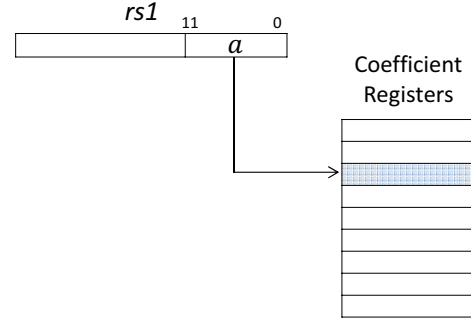


Fig. 5. Storing A Matrix Coefficient

MATY

The MATY operation calculates the y-coordinate of the world coordinate system and stores the y-coordinate to a GPR.

MATZ

The MATZ operation calculates the z-coordinate of the world coordinate system and stores the z-coordinate to a GPR.

IV. EXPERIMENTS

This section shows experimental results for evaluating the proposed ASIP for coordinate system transform.

A. How to Experiment

Evaluation is made by comparing the proposed ASIP with Brownie STD 32 and Brownie STD 32 with MAC instruction. In the experiment, execution cycle, energy consumption, area, power, and maximum frequency are compared. Operation voltages and frequencies of all three processors are 1.8V and 100MHz, respectively.

B. Experimental Result

1. Area, Power Consumption, and Maximum Frequency

Table II and III show experimental results of area, power, and maximum operating frequency with Brownie STD 32, Brownie STD 32 using MAC instruction and the proposed ASIP, in which numbers in parentheses indicate increase or decrease rate compared with Brownie STD 32.

Area was estimated from RTL description by logic synthesis using Synopsys Design Compiler and 0.18 μ m CMOS library. Power consumption was estimated by switching information of circuit obtained from gate level simulation using Mentor ModelSim.

2. Execution Cycle

Table IV shows execution cycle for motion sensing should be executed within a time window. Numbers in parentheses of Table IV indicate rates of decrease compared with

TABLE II
AREA AND POWER OF THE PROCESSORS

	Area[μm^2]	Power[$\mu W/MHz$]
Brownie STD 32	323,369	160.46
Brownie with MAC	395,642 (+22.4%)	180.51 (+12.5%)
ASIP	389,092 (+20.3%)	201.05 (+25.3%)

TABLE III
MAX FREQUENCY OF THE PROCESSORS

	Max frequency[MHz]
Brownie STD 32	348.4
Brownie with MAC	111.6 (-68.0%)
ASIP	207.9 (-40.3%)

implementation based on simple RISC instructions. By using custom instructions, the number of execution cycles is decreased by 58.7%.

3. Energy Consumption

Energy consumption is calculated from execution cycle, power consumption, and operating frequency. Let N [cycles] be execution cycles for motion sensing, P [mW] be power consumption, and f [MHz] be operating frequency. By using these symbols, energy consumption E [mJ] can be expressed by

$$E = PN \frac{1}{f}, \quad (10)$$

where $f = 100$ [MHz] in this experiment.

Table V shows energy consumption for motion sensing. Numbers in parentheses of Table V indicate decrease rates compared with Brownie STD 32. By using the proposed custom instructions, energy consumption in motion sensing process is decreased by 48.3%.

C. Discussion

The proposed processor is superior to the implementation based on simple RISC instructions in terms of energy consumption, while it is inferior in terms of power consumption. This is because execution cycle was reduced significantly by the proposed matrix product calculator.

The overhead in area, power and maximum operating frequency of the proposed processor are inferior to those of the implementation based on simple RISC instructions, however

TABLE IV
EXECUTION CYCLE FOR MOTION SENSING [cycles]

	Execution Cycle[cycles]
Brownie STD 32	371,755
Brownie with MAC	248,491 (-33.2%)
ASIP	153,378 (-58.7%)

TABLE V
ENERGY FOR MOTION SENSING [mJ]

	Energy for Motion Sensing[mJ]
Brownie STD 32	59.7
Brownie with MAC	44.9 (-24.8%)
ASIP	32.9 (-48.3%)

this is not critical for motion sensing devices, because motion sensing devices are non-invasive in general.

According to these experimental results, energy consumption for motion sensing decreases significantly by the proposed processor architecture and specific instructions for coordinate system transform.

V. CONCLUSIONS

In this paper, we introduced a method for coordinate system transform, which is necessary in motion sensing system, and proposed a processor architecture specified for matrix product calculation to transform coordinate system. In the experimental results the proposed architecture achieved reduction of execution cycles and energy consumption for motion sensing by using specific instructions for matrix product calculation.

In future, we will make a comparison with other devices like dedicated hardwares and practical processors, and then, develop a processor for motion sensing system using a 9-axis MEMS-based sensor, i.e. 3-axis accelerometer, 3-axis magnetometer and 3-axis gyroscope.

ACKNOWLEDGMENT

This work was partly supported by Regional Innovation Strategy Support Program of the Ministry of Education, Culture, Sports, Science and Technology in Japan (MEXT). We express our special thanks to Prof. Masaki Sekine from Osaka Electro-Communication University for his valuable discussion about motion sensing algorithm using accelerometer.

REFERENCES

- [1] N. Sekine, Y. Kuwae, Y. Higashi, T. Fujimoto, M. Sekine, and T. Tamura, "Improving the Accuracy of Pedometer Used by the Elderly with the FFT Algorithm," *Medicine Science in Sports Exercise*, Vol. 2006, No. 38(9), pp. 1674–81, Sep. 2006.
- [2] Y. Horita, M. Sekine, T. Tamura, Y. Kuwae, Y. Higashi, T. Fujimoto, Y. Oshima, and T. Shiga, "Development of a Step Count Algorithm for the Elderly," *Medical and Biological Engineering*, Vol. 2008, No. 46(2), pp. 283–288, Apr. 2008.
- [3] B. Ainsworth, W. Haskell, M. Whitt, M. Irwin, A. Swartz, S. Strath, W. O'Brien, D. J. Bassett, K. Schmitz, P. Emplaincourt, D. J. Jacobs, and L. AS, "Compendium of Physical Activities: an update of activity codes and MET intensities," *Medicine Science in Sports Exercise*, Vol. 2000, No. 32(9 Suppl), pp. S498–504, Sep. 2000.
- [4] N. Ryu, Y. Kawahara, A. Kobayashi, and T. Asami, "A Energy Expenditure Estimation Method for Non-Exercise Activity Thermogenesis Using Accelerometer," *IPSJ SIG Technical Report. UBI*, Vol. 2008, No. 40, pp. 67–74, May 2008.
- [5] A. Sabatini, "Quaternion-Based Extended Kalman Filter for Determining Orientation by Inertial and Magnetic Sensing," *Biomedical Engineering, IEEE Transactions on*, Vol. 53, No. 7, pp. 1346–1356, Jul. 2006.
- [6] H. Iwato, T. Hieda, H. Tanaka, A. Sato, K. Sakanushi, Y. Takeuchi, and M. Imai, "A Highly Extensible Base Processor for Short-term ASIP Design," *IEICE Technical Report. DC, Dependable Computing*, Vol. 107, No. 339, pp. 19–24, Nov. 2007.
- [7] M. Imai, Y. Takeuchi, H. Shiomi, A. Sato, and A. Kitajima, "An Application Specific Processor Development Environment : ASIP Meister," *IEICE Technical Report. DSP, Digital Signal Processing*, Vol. 102, No. 339, pp. 39–44, Oct. 2002.