Variable Bitspace of Variable Precision Processor

Sukemi
Student of Doctoral Program in Universitas Indonesia, Depok, Depok City, West Java, Indonesia
sukemi@ilkom.unsri.ac.id; kemi_dilla@yahoo.co.id

Anak Agung Putri Ratna and Harry Sudibyo
Department of Electrical Engineering Universitas Indonesia, Depok, Depok City, West Java, Indonesia
ratna@eng.ui.ac.id, harisudi@ee.ui.ac.id

***Abstract*—This research is purposed to increase computer function into a time driven to support real time system. This purposed would the processor can work according to determined time variable and can work optimally in a certained deadline. The first approachment to design some of processor that has a different bit space 64, 32, 16 and 8 bits. Each processor will be separated by selector/arbiter priority of a task. In addition, the design of the above processors are designed as a counter with varying levels of accuracy (variable precision computing). The selection is also done by using statistical control in the task are observed by the appearance of controller mounted on the front of the architecture bitspace the second approach above. The last approach to 'add' certainty in the form of interval arithmetic precision cutting task that can be the upper bound and lower bound of the area (bounds).**
These four approachment can be structured orthogonally or stand alone into a processor/several processors by introducing a new classifier that serves as a selector or a task arbiter. The result from the four approaches proves that the processor is strutured with different bitspace able to work optimally on a variable time and time deadlines are fast, and accurate completion of tasks in support of real time.

***Index Terms*—time driven, task arbiter, variable precision computing, interval arithmetic.**

I. INTRODUCTION

Time driven is a processor which is structured by variable time bait/input so that it can work according to determined time variable and can work optimally in a certained deadline [1].

The processor to be strutured has a different bitspace configuration/architecture (64, 32, 16 and 8 bits). These processor will include an arbiter/selector which directed to bitspace above in accordance with the task to be solved optimally.

The problem of designing systems with arbitration requires the designer to clearly understand the behaviour of typical arbiters in their interaction with users and resources.

Selected to be published in Journal of Computer (JCP, ISSN 1796-203X) September 28, 2013; Copyright Transfer Agreement October 17, 2013; Extended Number 156, TelSaTech 2013. +62 81310071719; +62 8128121453; +62 811856384.

doi:10.4304/jait.5.2.65-70

Moreover, in order to design a new arbiter the designer should come up with a formal specification of the arbiter, by defining its architecture and algorithm, to be able to verify its correctness and possibly synthesize its logic implementation. This kind of task requires an appropriate description language, which should be capable of describing the various dynamic scenarios occuring in the overall system in a compact and functionally complete form. Unfortunately, traditional digital design notations such as timing diagrams, although adequate for simple sequential behaviours cannot easily capture asynchronous arbitration in a formally rigorous and precise way. Such behaviour can be inherently concurrent, at a very low signalling level, and may involve various kinds of choice, both deteministic and nondeteministic.

The structure of the processor configuration is done with a step-by-step approach beginning with classifying or even cutting the mandatory and optional part-task of a scheduling, a processor that has an ability as a numerator priorities of a task. The second approachment, to design a processor that has ability as a computer with different of accuracy levels. Furthermore, work optimally of a processor can be executed by observing the emergence of the task occurences. This approachment is employing statistic control, which is by detecting the statistic variables such as mean, deviation, etc. By using this approachment, all tasks that the emergence will be observed with coprocessor-help which is placed in the front section of bitspace architecture of the second approachment above. As a consequence, a selector is needed to determine whether the data/task are suitable for the second or the third approachment. Precision realiability on the third approachment can not be guaranteed, whereas the intended processor must have high-speed of computing and high-accuracy characteristics. In fact, both of quantity above have a tradeoff. The last approachment is carried out by adding precision certainty in the form arithmetic interval that is able to cut the data/task into a form of upper and lower border from the bounds.

Computer architecture has represented multiply extensive bit form 8 bit, 16 bit, 32 bit to 64 bit. However, in fact, the data that will be acquainted, translated, and processed is not always in a specific architecture, but also in various architectures which need statistic control.

Another approachment is also already been executed

according to interval data or interval arithmetic. This approachment represents a processor model which cut upper and lower border of numerical data. However, this approachment makes the processor as a selector and increases the process-finishing-time. As the result, real time system can not be achieved.

Architecture model here in after which have been checked by previous researcher [2][3][4] in the form of precision variable. But just remain to be burdened by the problem of computing of speed and it accuration process decrease by processor.

This research also offers a new concept that changes the arbiter functions previously used as a regulator of the bus or the bus arbiter is more popularly called. Arbiter will function as a controller or a selector of task which is named as task arbiter.

II. PREVIOUS RESEARCH

Arbiter is known as bus controller with centralized method that is responsible for the allocation of time to module parts or CPU parts. This kind arbiter using Daisy Chain method wherein bus control passes from one bus master to the next one, then to the next and so on with priority level. The working of this arbiter is called as bus arbiter. So far, no one has offered arbiter function to controlled of task or task arbiter [1].

Task Arbiter is a component that works much like the bus arbiter. Arbiter component is the beginning of the next govern the use of resources, in this case the variable precision ALU 8 bit, 16 bit, 32 bit and 64 bit according to the type and size of the incoming task. So a little different from the bus arbiter arbitration only set of resources on the same bit width, for example, 32 bit or 64 bit only. So in addition to regulate arbitration task types also noticed bits wide incoming task before being processed in the ALU. With the use of these components, the expected time of execution of a task or set of tasks more quickly in real time.

Time Limitation in real time system is covered through process scheduling algorithm in operation system level. If the available time needed by a task is shorter, then the task execute will be failed. Computation Imprecise technique [5] divides each task into two parts are mandatory and optional part.

MSB-First (Most Significant Bit-First) Computation Technique [6] is numerical computation technique that perpetrates computation process started from the highest-value bit first, different from general computation technique that starts the computation from the lowest-value first. By using and applying this MSB-First Technique in answering arithmetic unit level in the beginning execution, high accuracy can be obtained.

Generally, accuracy determination of arithmetic operation result can not be carried out before operation execution is done. Interval-Arithmetic Technique [7][8] extends two operation results that are the lower and the upper border of the real numeric value. Combination of MSB-First and Interval-Arithmetic Technique hands over the need of producing answer with determine and high accuracy in the beginning process of arithmetic instruction execution.

At first, Variable-Precision technique is based on the need to process numeric data with various precision width so that numeric computation process of hardware needed [9][10] can be optimal. This technique enables to perpetrate computation of different data precision width.

The previous research also examined the arithmetic architecture to real time needed, however this architecture does not have a wide (bitspace) selection of bits that match the diversity of data/task incoming to be processed so that the real time system more secure. As well as the research that has been done by Kerlooza. YY.,2010 [11] has developed a research Mora-Mora., 2006 [12], which also did not provide a wide selection of bits (64, 32, 16 and 8 bit).

Finally, to support architecture-4 models above have priority weighting the arbiter/selector. This is a new concept [1] that is offered in this paper and combined architecture together with all 4 models above.

III. RESEARCH METHODOLOGY

The initial preparation of this research includes the study of literature on computer architecture (state of the art) from various journals and proceedings, as well as the functionality and architecture of arbitrators that are popular today. Then followed the determination of the specifications of the system to be planned and illustrates a block diagram of the system in general.

In this research proposal designed a new computer architecture that serves as the arbiter of control or steering selector bits to a particular space-based simulators Quartus II 9.0 SP2 FPGA that can potentially be implemented in high-precision scientific computing applications (high-precision) and provide the ability significantly higher than a regular CPU (general-purpose) [4][13]. However, FPGA still exists is used to accelerate variable precision application [13].

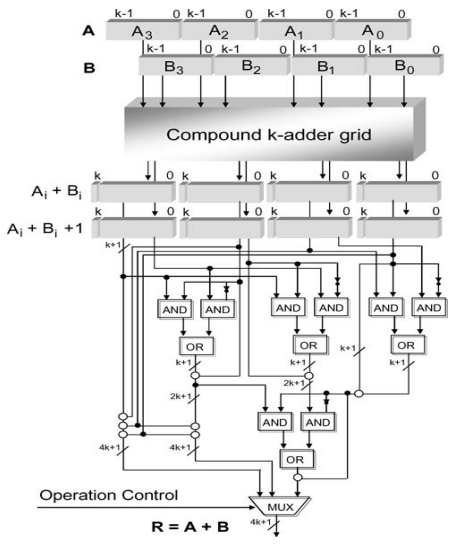

Figure 1. Flexible adder [12]

Arbiter and wide bitspace arithmetic prototype units will be built in the FPGA simulator software Quartus II

9.0 SP2 type. The simulator software has been successfully implemented on the paper being before [Kerlooza.YY.,2010]. This study and answer the originator of the previous topics on Variable Precision Processor [3][4][12].

Arithmetic adder [12] as prototype processor in Fig. 1 above will include an arbiter/selector that serves as selector of the models suitability to input task to obtain a speed of computing is much faster. The arbiter to be built is the liaison and steering as well as track/path for processing the data available on bitspace processor. The arbiter hardware pursued first tested through a special simulator on a prototype board (protoboard). This simulator is also a supporting part which clarify the function of the computer as a time driven.

Fig. 2, below illustrates a circuit of prototype processor with the arbiter/selector proposed as a selector bit space to be selected for optimal task execution process.

This prototype has arrangement of the components as previous research [3] that the register file consists of two memory units: a 64-word by 24-bit header memory, and a 256-word by *m*-bit significand memory. Besides that, the header word memory will stores the sign, and length of the variable-precision numbers.

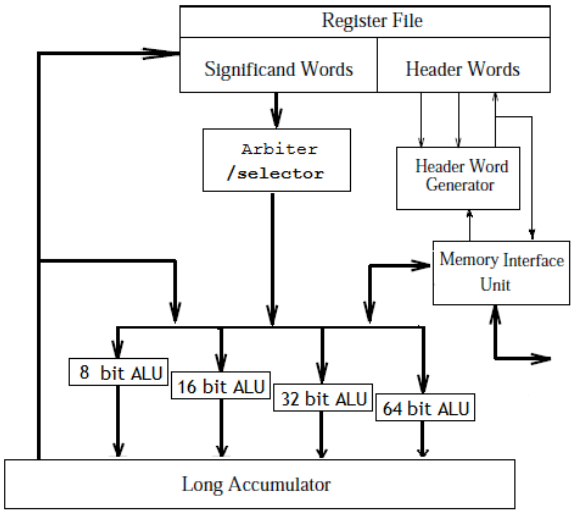

Figure 2. Prototype processor with arbiter.

The significand memory stores the significand words of the variable-precision numbers from most significant to least significant. The significand memory has two read ports and two write ports. The header memory consists of two read ports and one write port. The arbiter/selector consits of three input and eight output. Each output arbiter/selector is connecting to variable bitspace ALU. The long accumulator stores intermediate variable precision results. It functions as a long fixed point register that limits roundoff error in variable-precision adder arithmetic operations.

The new model architecture had released [12] as a adjustable computing but this architecture have not a arbiter/selector to select time and precison are optimal. The header word generator forms the header words of the variable-precision numbers. It consists of an exponent adder, and logic for producing the sign, and length of the variable-precison result. It also allows two variable-precision numbers to be compared, based on their signs. Output from the header word generator go to the header word memory or to the memory interface unit. The memory interface unit reads data and instructions from memory and writes data to memory. Outpus from the memory interface unit go to header word generator, and the bitspace-adder arithmetic.

In the Fig. 3. below illustrates a schematic adder circuit built and simulated in Altera Quartus II and Cyclone II devices (EP2C35F672C6). Arbiter Precision multi bitspace ALU is designed to function by using a control signal *ctl* to determine the size desired precision of a mathematical operation. The control signals used by the control circuit is located between the ALU components match the figure 3. N is the data width instructions, eg 8bit, 16bit, 32bit, 64bit or 128bit. While *n* is the block size of the granulation ALU used to process the mathematical instruction. If the operand width is 64bit and 8bit block size granulation is then there will be 6 blocks control with control signals, respectively, and 8 blocks granulation ALU. Control block controls the size of the desired precision of each operation based on the value 0 or 1 on the control signal. If the value is 0 then between granulation ALU connected to a 2n-bit precision. Conversely, if the value is 1 then the cut ALU inter granulation so that the precision of n-bit only. So if the control signal is 0 all the precision of N-bit and vice versa for n-bit precision.

Simulation results showed that the different bitspace have given execution time results every variable precision designed or desired by the programmer or developer has a significant time difference.

The initial hypothesis, improvement of system performance can be improved if the arbitrators [14] have chosen bitspace so that the data/task that has an arithmetic instruction execution can be had fairly high degree of accuracy. In the previous paper, it was mentioned that the execution of instructions with MSB techniques [Kerlooza.YY.,2010] provide results-between with a certain degree of accuracy is obtained since the beginning of the process execution. The higher increase in the accuracy of the between-results generated, the system will be more robust to conditions less processing time. Obviously, the implementation is done by inserting a time limit and accuracy in the instruction format. In contrast, conventional computational calculation accuracy of the answers obtained from the arithmetic instruction is done at the end of the process execution.

Most of the basic operations of measuring the size of 1 bit. This size is considered as the smallest unit size (granulation). This kind of operation is called bit-to-bit operator. However, with sub-parallel techniques [Mora-Mora.,2006] by enlarging the size of granulation operations into k-bit at a time, then the arithmetic instruction execution time becomes shorter.

Unfortunetly, granulation size is directly proportional to the complexity of the hardware.

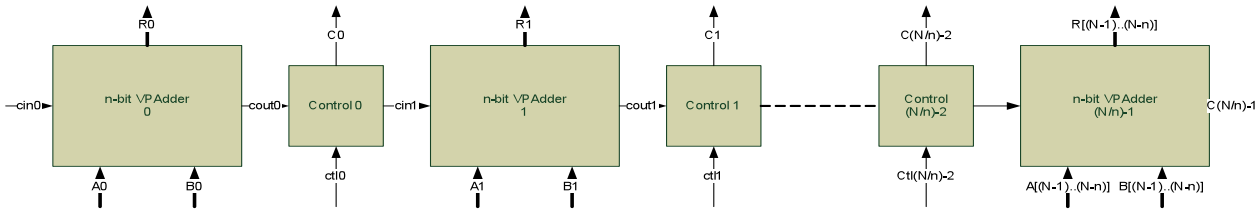

Figure 3. Variable Precision Adder

In this paper, the implementation of an arithmetic operator is also done by using Look-Up Table (LUT) available on the FPGA Quartus II, 9.0. LUT is used as an alternative implementation of the algorithm in hardware level because of technological advances integrated circuit (IC) that containts a lot of transistors.

Adder methode in this paper is based on the scheme MFIBVP [Kerlooza.YY.,2010] and carry-select adder [Mora-Mora.,2006] which produces an answer with accuracy that varies according to the time available. Furthermore, the incorporation of the new concept summation of the arbiter and the two above techniques are able to produce optimized resolution data/task that is timeless and accuracy.

IV. EXPERIMENTAL RESULT

Initial experimental results in the form of a series of logic gates which represent imprecise computation computational methods, variable precision, MSB and interval arithmetic. The gate circuit is simulated by Quartus II software 9.0 SP2 and generate signals sum representation with time and precision constraints.

From Table I below, indicates that the task will be processed on different bitspace generate greater value or not optimal. So it needs sorting task or task arbiter to direct bitspace appropriate so that optimality is reached. Moreover, in order to design a new arbiter the designer should come up with a formal specification of the arbiter, by defining its architecture and algorithm, to be able to verify its correctness and possibly synthesize its logic implementation. This kind of task requires an appropriate description language, which should be capable of describing the various dynamic scenarios occuring in the overall system in a compact and functionally complete form.

TABLE I.
TIME EXECUTION ON VARIABLE TASK IN VARIABLE-BITSPACE OF VP.

Task (bit)	Variable Bitspace of VP (bit)			
	64	32	16	8
64	80ns	X	X	X
32	80ns	60ns	X	X
16	80ns	60ns	27.5ns	X
8	80ns	60ns	27.5ns	7.5ns

In other types of software tools with different circuit Adder series but have the same function.

Simulation results of the adder circuit in fig. 3. given in the table II below.

TABLE II.
TIME EXECUTION ON VARIABLE TASK IN THE VARIABLE-BITSPACE OF VP IS MODIFIED.

n	N			
	8	16	32	64
4	2.72ns	4.56ns	8.24ns	30.70ns
8	2.48ns	5.03ns	9.15ns	34.02ns
16	X	5.21ns	9.68ns	35.96ns
32	X	X	9.79ns	37.02ns
64	X	X	X	38.34ns

In Table II above shows that the different bitspace for the same task or granulation produces a longer execution time.

Simulation results in Table I and II above, have differences due to software performances itself, in order to optimize the ability of the clock to the diversity of tasks that need to be serviced every stage of the process. This makes a fundamental difference to the two above as software tools that can simulate the adder function that is built specifically to have a variable precision at different bitspace. This work is done in order to get increased speed while still maintaining the desired precision.

Future work, design arbiter must be able to direct bitspace architecture built on a separate order to achieve optimality and hardware implementation requires estimates in the area of integrated circuits from a 1.0 micron CMOS standard cell library [15] on Table III below.

TABLE III.
AREA ESTIMATES (*mm2*) [14]

Unit	64-bit VPIAP	32-bit VPIAP	16-bit VPIAP
Multiplier	49.4	15.2	4.9
Adder	4.2	2.1	1
Significant Memory	35.5	178	8.8
Header Memory	3.3	33	3.3
Long Accumulator	13.1	5.6	2.5
Shifter	8.2	3.9	1.9
Operand Selector	1.6	8	0.4
Header Generator	1.4	1.4	1.4
Memory Interface	2.5	2.5	2.5
Control Logic	8.3	8.3	8.3
Latches	4.8	2.6	1.4
Pads, Space, etc	79.4	38.1	21.8
Total	212	102	58

These hardware requirements and area estimates needs if we built-in LSI Logic, but we can reduce if try to use FPGA tipe Quartus II 9.0 Sp2 protype processor.

The delay of each functional unit is computed by

taking the worst case delay of the critical path and adding 25 percent for process variations and clock skew [3]. Delay and cycle time estimates are given in Table IV.

TABLE IV.
DELAY AND CYCLE TIME ESTIMATES (*ns*) [3]

Unit	64-bit VPIAP	32-bit VPIAP	16-bit VPIAP
Multiplier	20	14	10.5
Adder	18.4	13.8	9.3
Significant Memory	18.4	13.8	9.3
Header Memory	8.2	7.4	7
Long Accumulator	6.6	6.6	6.6
Shifter	7.2	6.8	6.4
Operand Selector	8.9	8.2	7.8
Header Generator	3.2	3.2	3.2
Memory Interface	2.4	2.4	2.4
Control Logic	6	6	6
Latches	9.5	9.5	9.5
Pads, Space, etc	2	2	2
Total	22	16	12.5

In the future research Mora-Mora.,2006 after Michael J. Schulte.,1998 had writed delay estimation in table V for main gate and logic blocks in the arithmetic.

The hardware requirement, area estimates, delay estimation for 8-bit ALU is under 16-bit.

TABLE V.
DELAY ESTIMATION FOR MAIN GATES AND LOGIC BLOCK [12].

Logic block	Delay (r_t)
AND gate	0.5
OR gate	0.5
MUX block	0.5
CSA	1
k-adder (k=8)	3
k-multiplier (k=8)	6
k-division (LUT 8 bit address	3.5
k-square root (LUT 8 bit address)	3.5

V. CONCLUSIONS

Processor with architecture of variable precision which different bitspace can be realized with arbiter as selector and steering to optimal bitspace.

ACKNOWLEDGMENTS

This research has been funded by a Doctoral Dissertation Grant through the Directorate General of higher education ministries of education and culture of Indonesia. Year of 2013, through research institutes of Sriwijaya University, Palembang. 13000, Indonesia.

REFERENCES

[1] Sukemi, Sudibiyo, H., Ratna, A.A.P. Priority based computation a study on paradigm Shift on real time computation, Proceeding of CyberneticsCom, Bali: IEEE, (2012).
[2] Michael J. Schulte and E. E. Swartzlander, Jr., A variable-precision interval arithmetic processor. Proceedings of International Conference on Application Specific Array Processors, San Francisco: (1994) 248-258.
[3] Michael J. Schulte, Variable-precision, interval arithmetic processors. Department of Electrical Engineering and Computer Science. Lehigh University, Bethlehem, PA: (1996).
[4] Michael J. Schulte and E. E. Swartzlander. Jr., A family of variable-precision interval arithmetic processors, IEEE Transaction on Computer, 49 (5) (2000).
[5] Liu, Jane W.S:Lin;Kwei-Jay;Shih, wei-Kuan;et al, Algorithm for scheduling imprecise computation. In: computer. 24(5) (1991) 56-58.
[6] N. Asger Munk and K. Peter. MSB-First Digital serial arithmetic. Journal of Universal Computer Science. 1(7) (1995) 527.
[7] Moore, R.E., Interval arithmatic and automatic error analysis in digital computing. Dept. of mathematics of mathematics, Stanford of university, Stanford, California (1962).
[8] Boche, Ray E., An operational interval arithmetic. IEEE-Illinoi inst. of tech. Nortwestern univ., univ. of illinois. Abstrac of a paper given at national electronics confr. (1963).
[9] Chartres, Bruces A, Automatic controlled precision calculation. Journal of the ACM. 13(3)(1996) 386-403.
[10] Hull, T.E; Cohen, M.S; Hall, C.B., Specifications for a variable-precision arithmatic coprocessor. Proceedings of 10th IEEE symphosium on computer arithmatic, (1991) 127-131.
[11] Kerlooza, Y,Y., Gondokaryono, Y,S., and Mulyana, A., MSB-first interval-bounded variable-precision real time arithmatic Unit. Submitted to ICT Journal, ITB, (2010).
[12] Mora-Mora, Higinio; Mora-Pascual, Jerenimo; Garcia-Chamigo, Juan Manuel; et al., Real-time aritmetics unit. Real-Time Systems, 34, Kluwer Academic Publishers (2006) 53-79.
[13] Mark L. Chang and Scott Hauck, Variable precision analysis for FPGA synthesis. University of Washington. Department of Electrical Engineering. Seattle. (2004).
[14] Kinniment, J. David, Syncronization and Arbitration in Digital System., J. Wiley, England. (2007).
[15] LSI Logic Corporation, 1.0-micron Cell-based Product

Born in Palembang year of 1966. Sukemi was educated UNSRI S1 Electrical Engineering (1992). Then continued at the Institute Technology Bandung at Bandung for Magister degree (1999) and Doctor Engineering (2010 until now). Sukemi has teaching in the Departement of Computer System University of Sriwijaya since 2006 to the level of S1. First research was proposed about shift of paradigm computation system in Cyberneticscom2012 IEEE Indonesion section then continued at TelSaTech2013 for the same topics.

Prof . Harry, born in Madiun year of 1952 was educated UI S1 Electrical Engineering (1979). Then continued at the Universite Pierre et Marie Curie France for DEA degree (1984) and Doctor Engineering (1987). His post - doctoral training mileage at Universitat Duisburg Essen, Germany

(2000). Has been teaching in the Department of Electrical Engineering since 1982 to the level of S1, and S2. Taught courses include: Basic Electronics , Microelectronics , and CADVLSI. Research cooperation at home and abroad had he done include: Batch III study URGE , project , Director General of Higher Education entitled " Studies on the Integration on Microchip Antenna and Transceiver Circuit for Wireless Communication ".

Anak Agung Putri Ratna born in Jepang year of 1961. Has studied in Departement of Electrical Engineering UI at 1986 and continued in Waseda University for Doctoral Program. A.A.P. Ratna has teaching in University of Indonesia from 1987 till now. His research area focus in computer architecture, computer networking and e-learning model.

Call for Papers and Special Issues

Aims and Scope

JAIT is intended to reflect new directions of research and report latest advances. It is a platform for rapid dissemination of high quality research / application / work-in-progress articles on IT solutions for managing challenges and problems within the highlighted scope. JAIT encourages a multidisciplinary approach towards solving problems by harnessing the power of IT in the following areas:

- **Healthcare and Biomedicine** - advances in healthcare and biomedicine e.g. for fighting impending dangerous diseases - using IT to model transmission patterns and effective management of patients' records; expert systems to help diagnosis, etc.
- **Environmental Management** - climate change management, environmental impacts of events such as rapid urbanization and mass migration, air and water pollution (e.g. flow patterns of water or airborne pollutants), deforestation (e.g. processing and management of satellite imagery), depletion of natural resources, exploration of resources (e.g. using geographic information system analysis).
- **Popularization of Ubiquitous Computing** - foraging for computing / communication resources on the move (e.g. vehicular technology), smart / 'aware' environments, security and privacy in these contexts; human-centric computing; possible legal and social implications.
- **Commercial, Industrial and Governmental Applications** - how to use knowledge discovery to help improve productivity, resource management, day-to-day operations, decision support, deployment of human expertise, etc. Best practices in e-commerce, e-commerce, e-government, IT in construction/large project management, IT in agriculture (to improve crop yields and supply chain management), IT in business administration and enterprise computing, etc. with potential for cross-fertilization.
- **Social and Demographic Changes** - provide IT solutions that can help policy makers plan and manage issues such as rapid urbanization, mass internal migration (from rural to urban environments), graying populations, etc.
- **IT in Education and Entertainment** - complete end-to-end IT solutions for students of different abilities to learn better; best practices in e-learning; personalized tutoring systems. IT solutions for storage, indexing, retrieval and distribution of multimedia data for the film and music industry; virtual / augmented reality for entertainment purposes; restoration and management of old film/music archives.
- **Law and Order** - using IT to coordinate different law enforcement agencies' efforts so as to give them an edge over criminals and terrorists; effective and secure sharing of intelligence across national and international agencies; using IT to combat corrupt practices and commercial crimes such as frauds, rogue/unauthorized trading activities and accounting irregularities; traffic flow management and crowd control.

The main focus of the journal is on technical aspects (e.g. data mining, parallel computing, artificial intelligence, image processing (e.g. satellite imagery), video sequence analysis (e.g. surveillance video), predictive models, etc.), although a small element of social implications/issues could be allowed to put the technical aspects into perspective. In particular, we encourage a multidisciplinary / convergent approach based on the following broadly based branches of computer science for the application areas highlighted above:

Special Issue Guidelines

Special issues feature specifically aimed and targeted topics of interest contributed by authors responding to a particular Call for Papers or by invitation, edited by guest editor(s). We encourage you to submit proposals for creating special issues in areas that are of interest to the Journal. Preference will be given to proposals that cover some unique aspect of the technology and ones that include subjects that are timely and useful to the readers of the Journal. A Special Issue is typically made of 10 to 15 papers, with each paper 8 to 12 pages of length.

The following information should be included as part of the proposal:

- Proposed title for the Special Issue
- Description of the topic area to be focused upon and justification
- Review process for the selection and rejection of papers.
- Name, contact, position, affiliation, and biography of the Guest Editor(s)
- List of potential reviewers
- Potential authors to the issue
- Tentative time-table for the call for papers and reviews

If a proposal is accepted, the guest editor will be responsible for:

- Preparing the "Call for Papers" to be included on the Journal's Web site.
- Distribution of the Call for Papers broadly to various mailing lists and sites.
- Getting submissions, arranging review process, making decisions, and carrying out all correspondence with the authors. Authors should be informed the Instructions for Authors.
- Providing us the completed and approved final versions of the papers formatted in the Journal's style, together with all authors' contact information.
- Writing a one- or two-page introductory editorial to be published in the Special Issue.

Special Issue for a Conference/Workshop

A special issue for a Conference/Workshop is usually released in association with the committee members of the Conference/Workshop like general chairs and/or program chairs who are appointed as the Guest Editors of the Special Issue. Special Issue for a Conference/Workshop is typically made of 10 to 15 papers, with each paper 8 to 12 pages of length.

Guest Editors are involved in the following steps in guest-editing a Special Issue based on a Conference/Workshop:

- Selecting a Title for the Special Issue, e.g. "Special Issue: Selected Best Papers of XYZ Conference".
- Sending us a formal "Letter of Intent" for the Special Issue.
- Creating a "Call for Papers" for the Special Issue, posting it on the conference web site, and publicizing it to the conference attendees. Information about the Journal and Academy Publisher can be included in the Call for Papers.
- Establishing criteria for paper selection/rejections. The papers can be nominated based on multiple criteria, e.g. rank in review process plus the evaluation from the Session Chairs and the feedback from the Conference attendees.
- Selecting and inviting submissions, arranging review process, making decisions, and carrying out all correspondence with the authors. Authors should be informed the Author Instructions. Usually, the Proceedings manuscripts should be expanded and enhanced.
- Providing us the completed and approved final versions of the papers formatted in the Journal's style, together with all authors' contact information.
- Writing a one- or two-page introductory editorial to be published in the Special Issue.

More information is available on the web site at http://www.academypublisher.com/jait/.