Multimed Tools Appl (2014) 71:235–246
DOI 10.1007/s11042-013-1490-0

Interactive scheduling for mobile multimedia service in M2M environment

Anand Paul · Seungmin Rho · K. Bharnitharan

Published online: 29 May 2013

Abstract Computational load of motion estimation in advanced video coding (AVC) standard is significantly high and its more true for HDTV sequences. In this paper, video processing algorithm is mapped onto a learning method to improve machine to machine (M2M) architecture, namely, the parallel reconfigurable computing (PRC) architecture, which consists of multiple units, First, we construct a directed acyclic graph (DAG) to represent the video coding algorithms comprising motion estimation. In the future trillions of devices are connected (M2M) together to provide services and that time power management would be a challenge. Computation aware scheme for different machine is reduced by dynamically scheduling usage of multi-core processing environment for video sequence depending up complexity of the video. And different video coding algorithm is selected depending upon the nature of the video. Simulation results show the effectiveness of the proposed method.

Keywords Parallel processing · Video processing · Dynamic scheduling · Ubiquitous environment · M2M

1 Introduction

For the past 20 years several video coding standards such as MPEG-1, MPEG-2/4, H.263, H.264 [6], have been playing a significant role in digital media revolution. The recent advancement in H.264 significantly increases coding efficiency and processor computation power with inclusion

A. Paul
The School of Computer Science and Engineering, Kyungpook National University, Daegu, South Korea
e-mail: paul.editor@gmail.com

S. Rho (✉)
Department of Multimedia, Sungkyul University, Anyang-si, South Korea
e-mail: smrho@sungkyul.edu

K. Bharnitharan
Department of Electrical Engineering, Feng Chia University, Taichung, Taiwan

of variable block motion estimation and many other coding features [23], and that is very true for HDVT sequences and there is a need to efficiently process the video. Recently, the computation-aware (CA) concept is getting attention of video processing researchers. In software implementations, processors may have to support video coding of different frame rates, frame sizes, and search ranges. In hardware implementations, even if the frame rate, frame size, and search range have been clearly determined, the computation resource (e.g. operating frequency) may still be adjusted according to the battery power for portable devices.

The ability to enable and disable components, as well as tuning their performance to the workload (e.g., user's requests), is important in achieving energy efficient utilization. In this work, we will present new approaches for lowering energy consumption in both system design and utilization. The Policy presented in this thesis has been experimented using an event driven simulator, which demonstrated the effectiveness in power savings with less impact on performance or reliability [8].

The fundamental premise for the applicability of power management schemes is that systems, or system components, experience non-uniform workloads during normal operation time [4, 19]. Non-uniform workloads are common in communication networks and in almost any interactive system.

Dynamic power management (DPM) techniques achieve energy efficient utilization of systems by selectively placing system components into low-power states when they are idle [1]. As illustrated in Fig. 1. The goal of CA BMA [Block Matching Algorithm]s is to find the best block matching results in a computation-limited and computation-variant environment. The authors of [21] are pioneers of CA BMAs. They contributed a novel scheme. But it was intended for software and not feasible in hardware [3, 22].

With M2M, machines could not only retrieve information from other devices but also, to some extent, take action based on the information. Sensors that gather the information that some M2M systems transmit are becoming more widely used and thus are driving demand for the technology especially in power management area. The vision of an M2M and Internet of Things built from smart devices raises several important research questions in terms of system architecture, design and development, and human involvement [7]. The lower cost of sensors and initiatives for integrating them into larger systems are also increasing the approach's popularity. The biggest new trend is that vendors are expanding M2M into wireless technology, using radio chips or modules they can attach to almost any device or machine. Thus, M2M is gearing up for exponential

Collocated smart objects with local rule databases (1–5) form an ad hoc reasoning system in which logical queries are sent from object to object and results are passed back along the inference chain(refer Fig. 2). In this example, smart object 1 contains one rule, A if B; to evaluate B, it sends a query to objects 2 and 3, which in turn asks object 4. Reasoning chains can be limited to physical areas around the originating object. In this example, smart object 5 is beyond the distance limit set by smart object 1 (the circle indicates the maximum distance from object 1).

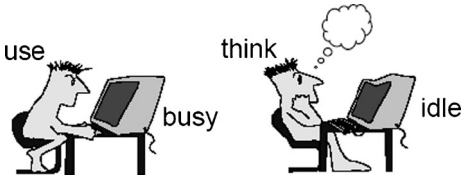


Fig. 1 An interactive systems is busy or idle depending on the user requests

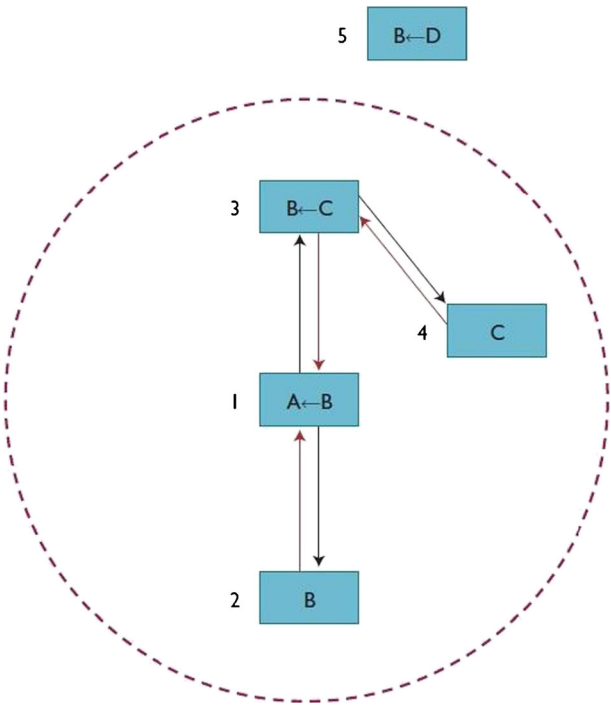


Fig. 2 Peer- to peer reasoning with smart object [7]

2 Mobile multimedia from M2M context

In order to evaluate the performance of video compres-sion coding, it is necessary to define a measure to compare the original video and the video after compressed. Most video compression systems are designed to minimize the mean square error (MSE) between two video sequences Ψ1 and Ψ2, which is defined as

$$MSE = \sigma_e^2 = \frac{1}{N}\sum_t \sum_{x,y} [\Psi 1(x,y,t) - \Psi 2(x,y,t)]^2 \quad (1)$$

where N is the total number of frames in either video sequences.

Instead of the MSE, the peak-signal-to-noise ratio (PSNR) in decibel (dB) is more often used as a quality measure in video coding, which is defined as

$$PSNR = 20\log_{10}\frac{255}{MSE} \quad (2)$$

It is worth noting that one should compute the MSE between corresponding frames, average the resulting MSE values over all frames, and finally convert the MSE value to PSNR.

If you are expected to be late, our mobile device inform our audience automatically telling them approximately how long they have to wait. Moreover, our planner can look up the traffic condition in advance and suggests what time you should leave. Sensors can monitor the traffic conditions along the routes to your destination so that you are able to select the best route to get to the venue on time [2].

2.1 Predictive coding

However, the performance is not efficient to compress the fast changing video, so other methods to remove the spatiotemporal redundancy are proposed for video compression. The most famous method is the block matching algorithm, or motion estimation and motion compensation method. The block matching algorithm divides the current frame and the previous frame into several macroblocks, comparing the blocks in the two frames and trying to search for the best matched pairs for each block [12].

The dissimilarity D(s,t) (sometimes referred to as error, distortion, or distance) between two images Ψn andΨn-1 is defined as follows

$$D(s,t) = \sum_{Vy=1}^{p} \sum_{Vx,=1}^{q} M[\Psi n(x,y), \Psi n{-}1(x+Vx, y+Vy)] \tag{3}$$

where M(u,v) is a metric that measure the dissimilarity between the two arguments u and v.

There are several types of matching criteria and two most frequently used is MSE and MAD, which is defined as follows:

- Mean square error (MSE):

$$M(u,v) = (u{-}v)^2 \tag{4}$$

- Mean absolute difference (MAD):

$$M(u,v) = |u{-}v| \tag{5}$$

A study based on experimental works reported that the matching criterion does not significantly affect the search. Hence, the MAD is preferred due to its simplicity in implementation [12].

These two measures (MSE/MAD) give us a clear idea about the video frame. And MSE/MAD of the previous frame gives more information about the current frame and its complexity. If the current frame of a video sequence is fast moving (complex) then more processor core are allotted to process else only one processor core is assigned.

2.2 Background

From our previous work, even though Deterministic Markov Non-Stationary Policies (DMNSP) gives best power optimization, we can find out other policies than DMNSP of DPM also give best power optimization with respect to different arrival of request in embedded system model [17, 18]. For example, when the requests of requester are coming in long time interval, the greedy policy can give best power optimization compare with others and our DMNSP policies. When the requests are coming in continuously without inter arrival time worst policy (always on) can give best result. To allow further improvement in the battery life of portable devices, one new energy reduction schemes will be needed which ought to predict a best and suitable policy amidst existing policy online. This call for the use of intelligent controllers which can learn itself to predict a best policy that balance workload against power.

The number of levels in a DAG defined as the “depth” of a DAG. Maximizing the speed of a DAG is equivalent to minimizing the depth of the DAG. Thus partitioning and optimization methods for the fine grained DRL/M hardware architecture will be considered. For this DRL architecture, consideration of parallel processing in temporal partitioning may

improve execution time. Many designers are interested in exploiting the inherent parallelism often present in large signal processing applications, so the trade-off between speed and silicon area with various levels of parallelism becomes an important design issue. Hence, a partitioning technique is developed with this parallel processing consideration in mind. The ability to find a minimum depth solution with duplication is necessary. Thus, a graph is partitioned into sub-graphs with a minimum depth solution. Each sub-graph consists of a set of clustering sub-graphs. Each sub-graph is selected by the greedy method, one at a time. This method will be shown to find the minimum depth partitioning of an application.

2.3 Complexity analysis

The whole partitioning procedure has a low time complexity. Before finding the MFC, all the nodes are processed by topological sorting. Hence during portioning, the nodes are visited in topological order. The time complexity of visiting the nodes is O(V.logV) where V is the number of nodes in the given graph. The task of finding each MFC includes calculating the area, sorting the area and labeling nodes. Each floor cone area is calculated by the depth first search (DFS) methodology. Hence, the time complexity of calculating the area is O(V.logV). In sorting the area, the number of floor cones is no more than the number of nodes V. The time complexity is O(V). When labeling nodes, a node is visited once. The complexity of labeling nodes is O(V). In the MFC-processing, to apply FFD the number of MFC is no more than the number of nodes V. The time complexity is O(V.logV). The minimum-depth partitioning solution selects k-MMG's in each greedy method iteration. Therefore, depth determining takes O(V2.logV) time. In conclusion, the total time complexity is bounded by O(V2.logV).

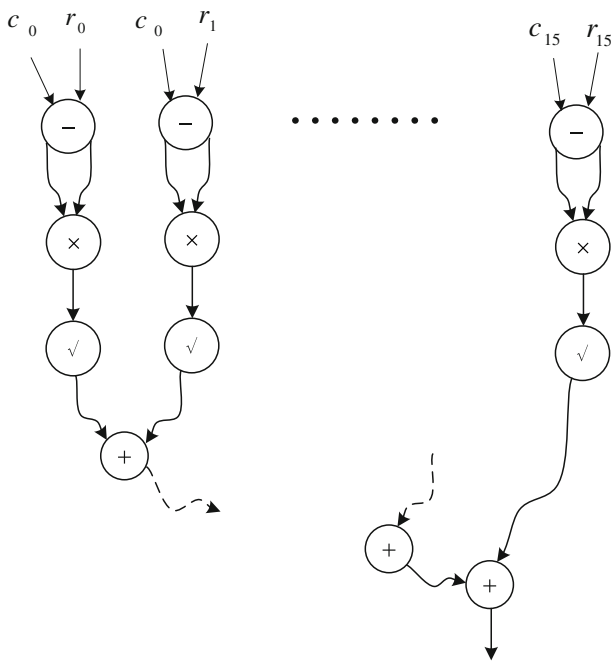


Fig. 3 The data flow graph of SAD calculation

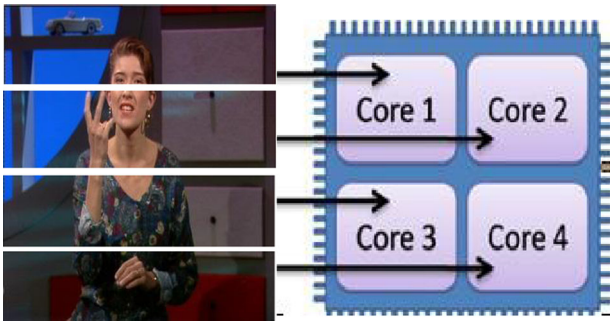


Fig. 4 Multiple core share the complexity which processing a frame

The reference pixel block is generated by displacement from the current block's location in the reference frame. The displacement is provided by the Motion Vector (MV). MV consists of is a pair (x, y) of horizontal and vertical displacement values. There are various criteria available for calculating block matching. The popular criteria of sum of absolute difference (SAD) is listed bellow

$$SAD(dx, dy) = \sum_{i=0}^{15} \sum_{j=0}^{15} \left| c_{i,j}^{x,y} - r_{i,j}^{x+dx,y+dy} \right| \tag{6}$$

where c_{ij} is a sample of the template macroblock and r_{ij} is a sample of the candidate block. The search range(SR) is 16×16, and macroblock size is 16×16. Figure 3 illustrates the SAD calculation. Current and reference block candidate are represented by c and r.

As it is required to search the whole search range to find the correct MV. First say, the search starts in x direction, Fig. 3 shows the DFG of the macroblock (MB). In this graph, there are 256 (16×16) inputs and 1 output. However, in the HDVT there are $1920 \times 1080/16 = 8100$MBs.

3 Mobile multimedia for intelligent M2M

There are also computing requirements that need to be taken into account when it comes to SDTV and HDTV, assuming we want to encode 720 p at 30 frames per second, this would require to handle 1,280×720 pixels per frame which gives us 27,648,000 pixels each second to process. That's over 27 million pixels to encode and another 27 million pixels to decode. High definition is usually done using the H.264 standard (in order to lower the bandwidth required while keeping high quality) and H.264 is quite complex to compute [13].

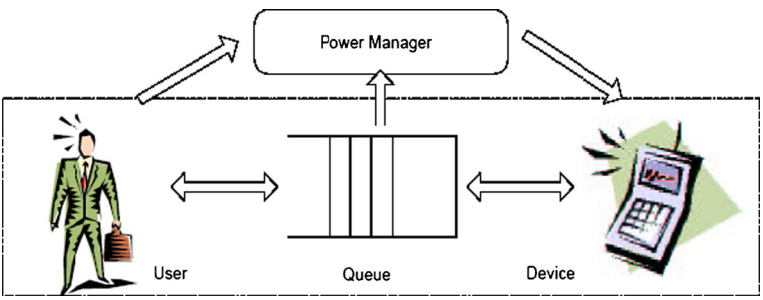


Fig. 5 Requester model

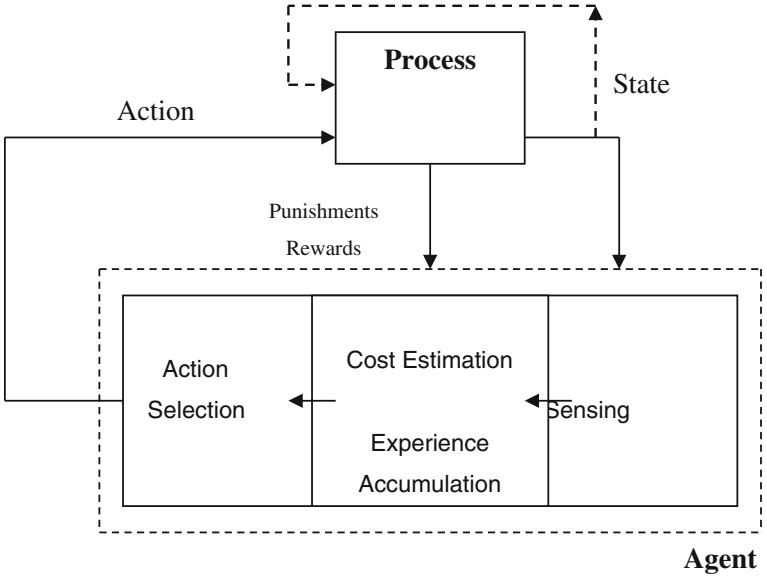


Fig. 6 RL Scheduler and decision maker

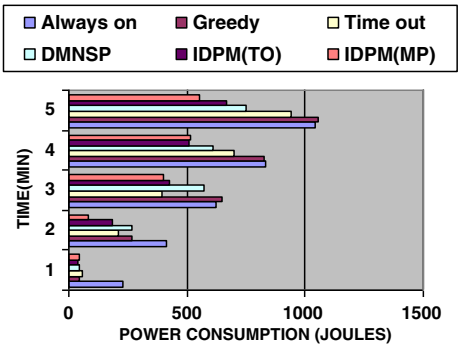
3.1 Slice partition for HDTV

Advanced video encoding techniques take each frame, split it into smaller section called slice and then analyze and encode each macro block [15]. In other words, multi-core processors can be utilized in a fairly simple way, we split the pictures into regions and encode the regions in parallel. This is not trivial because sometimes the actual processing of a macro block is dependent on previous images or its neighbor macro blocks (especially for motion estimation algorithms), but other than that, it is quite straightforward. Each core processes each slice, we use a scheduling algorithm to switch between core when video is fast and smooth [11, 14]. For illustration we use Fig. 4 (Irene-sign language sequence), though slice are not exact rectangular split.

3.2 Requester model

A special entity called "requester" that generates workloads including IO requests and computation needs. Request modeling is one essential part of power management because policies

Fig. 7 Power consumption of IBM hard disk drive under different DPM polices



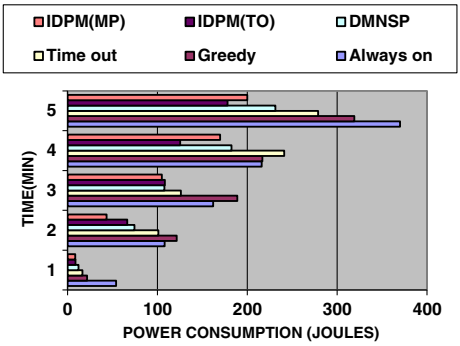


Fig. 8 Power consumption of Fijitsu hard disk drive under different DPM polices

predict future workloads based on their requester models. We consider two requester models for designing policies: single requester, multiple requesters. These models are increasingly complex and close to the programs running on realistic interactive systems like a laptop computer.

Figure 5 depicts the concept of the single-request model. The requester generates requests for the device; meanwhile, the power manager observes the requests. Based on this observation, the power manager issues commands to change the power states of the device. Some policies explicitly use this model in determining their rules to change power states [5, 20]; some other policies implicitly assume a single requester [10].

3.3 Reinforcement learning (RL) for M2M systems

Figure 6 shows how the RL agent interacts with the process. Nearly all RL methods currently in use are based on the Temporal Differences (TD) technique [16]. The fundamental idea behind it is prediction learning: when the agent receives reinforcement, it must somehow propagate it backwards in time so that states leading to that condition and formerly visited may be associated with a prediction of future consequences. This is based on an important assumption on the process' dynamics, called the Markov condition [9]: the present observation must be perfectly a conditional probability on the immediate past observation and input action. In practical terms, this means that the agent's sensors must be good enough to produce correct and unambiguous observations of the process states.

Power consumption of IBM Hard Disk Drive under different DPM polices are shown in Figs. 7, 8, and 9. In Tables 1, 2, and 3 Columns 2 to 4 respectively show the power consumption implemented by general purpose processor (GPP), parallel processing elements without the local memory for various DPM scheme.

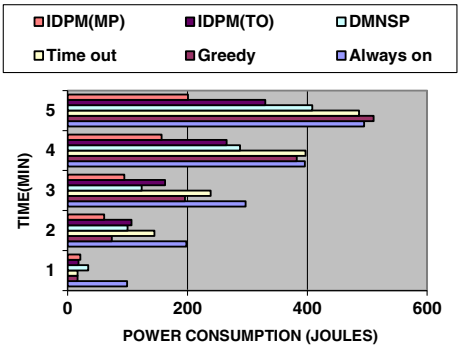


Fig. 9 Power consumption of HP-Smart-Badge under different DPM polices

Table 1 Power consumption of IBM hard disk drive under different DPM polices

MM1	Power consumption in joules				
Time(min)	1	2	3	4	5
Always on	228	416	624	832	1040
Greedy	45.66	268.2	647.9	825.2	1053.4
Time out	58.83	206.85	394.5	700.6	939.39
DMNSP	45.91	263.91	574.34	612.11	749.7
IDPM(TO)	38.99	185.88	428.7	506.76	668.31
IDPM(MP)	43.18	83.52	398.53	514.5	553.3

4 Experimental results

Our proposed algorithm can compute a theoretically unlimited number of parallel FPGA modules (DRPPU) but, for reasons of simple comparison for core to core communication, we perform simulations for parallel arrays from one to 5 DRPPU and for DRPPU areas (A_{DRPPU}) from 1,536, 2,304, 2,688, 4,992, 6,144, 6,656 to 9,280 CLB. Thus we are demonstrating the ability of our algorithm to help design massively parallel architecture. In fact, FPGA is intrinsically capable of such function but application up to the present time has been largely linear, due to lack of design tools and the habitual persistence of traditional thinking.

On the other hand as regards to dynamic power management policies are concern, all the policies suggested so far [9] have either under prediction or over prediction by which they pay performance or power penalty. Our policy makes sure that server is ON, when there is an event in the Service Requester and Service Queue. Which means that under prediction or over prediction will never occur. Performance penalty will never occur by the proposed scheme.

5 Conclusion

The addition of parallel processing techniques to reconfigurable computing has the potential to improve DSP applications. Therefore, multiple processing units arranged according to traditional parallel processing techniques are being applied for high computation and data intensive applications such as HMM. This paper has presented a minimum-depth partitioning algorithm for parallel reconfigurable computing. It is shown that application

Table 2 Power consumption of Fijitsu hard disk drive under different DPM polices

MM1	Power consumption in joules				
Time(min)	1	2	3	4	5
Always on	54	108	162	216	370
Greedy	21.55	121.38	188.9	216.81	319.11
Time out	16.5	100.99	126.03	240.99	278.7
DMNSP	12.17	74.41	107.75	182.43	231.18
IDPM(TO)	8.64	66.3	108.18	125.24	177.94
IDPM(MP)	8.54	43.19	104.7	169.69	199.8

Table 3 Power consumption of HP-Smart Badge under different DPM polices

MM1	Power consumption in joules				
Time(min)	1	2	3	4	5
Always on	99	198	297	396	495
Greedy	16.29	73.50	195.65	382.27	510.71
Time out	16.56	144.57	238.65	397.1	486.37
DMNSP	34.12	100.1	123.625	287.7	408.22
IDPM(TO)	17.99	106.35	162.47	265.43	329.94
IDPM(MP)	21.085	61.05	94.5	156.57	200.79

speed can be improved by increasing parallelism of the parallel DRC units. The resulting high-parallelism design has somewhat higher total chip area because of redundancy between parallel units. The proposed algorithm can accept arbitrary chip area constraints or maximum parallelism constraint and then optimize for speed

Good predictors should minimize the number of mispredictions. We call over prediction (under prediction) a predicted idle period longer (shorter) than the actual one. Over predictions give rise to a performance penalty, while under predictions imply power waste but no performance penalty. To represent the quality of a predictor we define two figures: safety that is the complement of the risk of making over predictions, and efficiency, that is the complement of the risk of making under predictions.

All the policies suggested so far have either under prediction or over prediction by which they pay performance or power penalty. Our policy makes sure that server is ON, when there is an event in the Service Requester and Service Queue. Which means that under prediction or over prediction will never occur. Performance penalty will never occur by our policy.

Acknowledgments This research is support by Kyungpook National University Research Fund 2012. This work was partially supported by URP-CEST 2013 [Undergraduate Research Program - Center for Embedded Software Technology], Kyungpook National University, Korea.

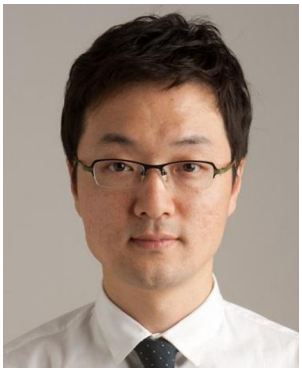
References

1. Benini L, Paleologo G, Bogliolo A, De Micheli G (1999) Policy optimization for dynamic power management. IEEE Trans Comput Aided Des 18(6):813–833
2. Chen Y-K (2012) Challenges and opportunities of internet of things. ASP-DAC, 2012, 17th Asia-South Pacific Conference Proceedings, Page 383–388, Jan 30th–Feb 2nd 2012
3. Chen LF, Lai YK (2004) VLSI architecture of the reconfigurable computing engine for digital signal processing applications. IEEE Circuits and Systems Conference, ISCAS '04. pp 937–40
4. Chung E, Benini L, De Micheli G (1999) Dynamic power management for nonstationary service requests. Design, Automation and Test in Europe, pp 77–81
5. Johnson RA (2001) Probability and statistics for engineers. Prentice hall of India
6. Joint Video Team of ITU-T and ISODEC JTC I (2003) Draft ITU-T recommendation and final draft international standard of joint video specification. (ITU-T Kec. H.264 ISO/IEC 14496.10 AVC) JVT of ISO/IEC MPEG and ITU-T VCEG , JVT – GO05
7. Kortuem G, Kawar F, Fitton D, Sundramoorty V (2010) Smart object as building blocks for the internet of things. IEEE Internet Comput 44–51
8. Lu Y, Chung E, Simuníc T, Benini L, De Micheli G (2000) Quantitative comparison of PM algorithms. Design, Automation and Test in Europe, pp 20–26

9. Lu Y-H, De Micheli G (2001) Comparing system-level power management policies. Stanford University, IEEE
10. Maestre R, Kurdahi FJ, Fernández M, Hermida R, Bagherzadeh N, Singh H (2001) Kernel scheduling techniques for efficient solution space exploration in reconfigurable computing. Special issue on modern methods and tools in digital system design. J Syst Archit 47:277–292
11. Paul A (2013) High performance for adaptive deblocking filter in H.264/AVC system. IETE Tech Rev
12. Paul A, Bharanitharan K, Wu J (2013) Algorithm and architecture for adaptive motion estimation in video processing. IETE Tech Rev 30(1):24–30
13. Paul A, Chen B-W, Bharanitharan K, Wang J-F (2013) Video search and indexing with reinforcement agent for interactive multimedia services. ACM Trans Embed Comput Syst 12(2)
14. Paul A, Jiang YC, Wang JF, Yang JF (2012) Parallel reconfigurable computing based mapping algorithm for motion estimation in advanced video coding. ACM Trans Embed Comput Syst 11(S2)
15. Paul A, Wu J, Yang J-F, Jeong J (2011) Gradient-based edge detection for motion estimation in H.264/AVC. IET Image Process 323–327
16. Qiu Q, Pedram M (1999) Dynamic power management based on continuous-time Markov decision processes. Design Automation Conference
17. Schmit H et al (2002) PipeRench: a virtualized programmable datapath in 0.18 micron technology. IEEE Custom Integrated Circuits Conference, pp 63–66
18. Singh H, Lu G, Lee M, Kurdahi FJ, Bagherzadeh N, Filho E, Maestre R (2000) MorphoSys: case study of a reconfigurable computing system targeting multimedia applications. Proceedings Design Automation Conference (DAC'00), pp 573–578, Los Angeles, California
19. Stallings W (2003) Computer organization and architecture: designing for performance. Pearson Education
20. Sutton RS, Barto AG (1998) Reinforcement learning—an introduction. MITPress, Cambridge, A Bradford Book
21. Tsai PL, Huang SY, Liu CT, Wang JS (2003) Computationaware scheme for software-based block motion estimation. IEEE Trans Circuits Syst Video Technol 13(9):901–913
22. Vissers KA (2003) Parallel processing architectures for reconfigurable systems. Design, Automation and Test in Europe Conference and Exhibition, pp 396–397
23. Wiegand T, Sullivan GJ, Bjontegaard G, Luthra A (2003) Overview of the H.264/AVC video coding standard. IEEE Trans Circuits Syst Video Technol 13(7):560–576

Anand Paul he is currently working in The School of Computer Science and Engineering, Kyungpook National University, South Korea as Assistant Professor, He got his the Ph.D. degree in the electrical engineering at National Cheng Kung University, Taiwan, R.O.C. in 2010. He worked as an Assistant Professor from 2010 to 2012 in Hanyang University, Seoul, South Korea. His research interests include Algorithm and Architecture for motion estimation in video, and Digital Video SoC design and Reconfigurable Embedded Computing. 2004–2010 he has been awarded Outstanding International Student Scholarship, and in 2009 he won the best paper award in national computer symposium, Taipei, Taiwan.

He serves as a reviewer for IEEE Transactions on Circuits and Systems for Video Technology, ACM Transactions on Embedded Computing Systems, IET Image Processing, IET Signal Processing and IET Circuits and Systems He gave invited talk in International Symposium on Embedded Technology workshop in 2012, Daegu, South Korea., He is an MPEG and M2M Focus Group Delegate member representing South Korea.



Dr. Seungmin Rho received his M.S. and Ph.D. Degrees in Computer Science from Ajou University, Korea, in Computer Science from Ajou University, Korea, in 2003 and 2008, respectively. In 2008–2009, he was a Postdoctoral Research Fellow at the Computer Music Lab of the School of Computer Science in Carnegie Mellon University. In 2009–2011, he had been working as a Research Professor at School of Electrical Engineering in Korea University. In 2012, he was an assistant professor at Division of Information and Communication in Baekseok University. Dr. Rho is currently a faculty of Department of Multimedia at Sungkyul University. His research interests include database, music retrieval, multimedia systems, machine learning, knowledge management and intelligent agent technologies. He has been a reviewer in Multimedia Tools and Applications (MTAP), Journal of Systems and Software, Information Science (Elsevier), and Program Committee member in over 15 international conferences. He has published more than 50 papers in journals and book chapters and 40 in international conferences and workshops. He has been listed in Who's Who in the World in 2007 and 2008, respectively.

K. Bharanitharan (S'07–M'09) received the PhD degree in Electrical Engineering from the National Cheng Kung University, Tainan, Taiwan, in 2009. In 2005, he won outstanding international student fellowship award at National Cheng Kung University. He serves as a reviewer for IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Very Large Scale Integration Systems, IEEE Transactions on Evolutionary Computation, IEEE Signal processing letter, IEEE Transactions on Very Large Scale Integration Systems since 2009. He has published more than 16 research papers in highly reputed journals and conferences.

His research interests include H.264/AVC video coding, HEVC, scalable video coding, image processing, multiview video coding, and associated VLSI architectures. His research works also include Multi-Core reconfigurable systems, Java based apps development and dynamic power management for advanced video coding.