IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 8, Issue 6 (Nov. - Dec. 2013), PP 32-38
www.iosrjournals.org

A Proficient Thermal Based Prediction Unit in Multi Core System

Navya Vipin [1], Shivayya Gadag[2]
[1](Telecommunication Engg, MVJ College of Engg Bangalore/ VTU Belgaum, India)
[2](Telecommunication Engg, MVJ College of Engg Bangalore/ VTU Belgaum, India)

Abstract: *Multi core system has become a popular Execution platform for many Real time systems. The system is continue to shrink down due to thermal hotspots and temperature gradients, which Are significantly impact system parameters like Reliability, performance, cost and leakage power To improve these parameters in Real time systems/ multi core system a Temperature prediction management (TPM) technique is required for mission/safety critical application running on a multi core chip to avoid possible thermal hazards while meeting the application timing constraints.*

In this paper, we propose an efficient framework for Temperature prediction management (TPM) with time constraint model (Tracker & WDT) in many core systems where the balanced thermal profile can be achieved by proactive task migration among neighboring cores. And based on temperature prediction unit task migration will be performed among neighboring processing elements. Compared with existing proactive task migration technique ,our approach effectively minimize the thermal hotspots, less migration overhead with negligible performance overhead and it increase the processing speed of multi core systems.

Keywords: *Temperature prediction unit ,watch dog timer, multi agent ,prediction, task migration***.**

I. Introduction

With the unprecedented number of transistors integrated on a single chip, the current multi-core technology may soon progress to hundreds or thousands of cores era [1]. Examples of such system are the 80-tile network-on-chip that has been fabricated and tested by Intel and Tilera's 64 core TILE64 processor [9] .While the multi core or many-core technology delivers extraordinary performance, they have to face the significant power and thermal challenges. The increasing chip complexity and power density elevate peak temperatures of chip and unbalance the thermal gradients. Raised peak temperatures reduce lifetime of the chip, deteriorate its performance, affect the reliability and increase the cooling cost.

Dynamic thermal management (DTM) approaches such as core throttling or stalling which are widely used in computer systems usually have negative impact on the performance. The adverse positive feedback between leakage power and raised temperature creates the potential of thermal runaway .When mapped on a many-core system; diverse workload of applications may lead to power and temperature imbalance among different cores. Such temporal and spatial variations in temperature create local temperature maxima on the chip called the hotspot [4], [. An excessive spatial temperature variation, which is also referred to as the thermal gradients, increases clock skews and decreases performance and reliability.

Many dynamic thermal management techniques such as clock gating, dynamic voltage, and frequency scaling, thread migration have been proposed for multi-core systems. All these techniques aim to ensure the system running under a fixed safe temperature constraint [2],[3],[5],[6],[7],[8],Most of these existing techniques is centralized approaches.

They require a controller that monitors the temperature and workload distribution of each core on the entire chip and make global decisions of resource allocation. Such centralized approaches do not have good scalability. First of all, as the number of processing elements grows, the complexity of solving the resource management problem grows exponentially. Second, a centralized resource management unit that monitors the status and issues DTM commands to each core generates huge communication overhead in many-core architecture, as communication between the central controller and cores will increase exponentially with the number of cores. Such overhead will eventually affect the speed of data communication among user programs and also consume more power on the interconnect network. Finally, as the size and the complexity of the many-core system increase the communication latency between the central controller and the cores increases, this leads to a delayed response and sub-optimal control.

In this project, a framework of Temperature prediction management (TPM) is designed where balanced thermal profile can be achieved by proactive thermal throttling as well as thermal-aware task migrations among neighboring cores. The framework has a low cost agent residing in each processing element (PE). The agent observes the workload and temperature of the PE while exchanging tasks with its nearest neighbors through negotiation and communication. The goal of the proposed task migration is to match the PE's heat removal capability to its workload (i.e., the average power consumption) and at the same time creates a

good mix of high power (i.e., "hot") tasks and low power (i.e., "cool") tasks running on it. As each agent monitors only the local PE and communicates with its nearest neighbors, the proposed framework achieves much better scalability than the centralized approach. This project refers to the technique as Temperature prediction management (TPM) migration (TPM-M) as it aims at balancing the workload and temperature of the processors simultaneously.

II. Aim Of Project Work

The aim of the project is to design distributed thermal Management framework for many core systems. In this framework, no centralized controller is needed.

III. Justification Of The Project

1. To design Multi core systems with inputs like clock, reset, present temperature, data-in, neighbors' temp/data/ migration_ requests etc.
2. To design master slave communication between each cores.
3. To verify the task migration performed over cores.

IV. Problem Formulation

In this Project, a framework of Temperature prediction management is designed. The framework has a low cost agent residing in each processing element (PE).The agent observes the workload and temperature of the PE while exchanging tasks with its nearest neighbors through communication. The goal of the proposed task migration is to match the PE's heat removal capability to its workload i.e., the average power consumption and at the same time create a good mix of high power tasks and low power tasks running on it. As each agent monitors only the local PE and communicates with its nearest neighbors, the proposed framework achieves much better scalability than the centralized approach.

V. Scope Of Project Work

The Temperature prediction management policy achieves almost the same performance as a global management policy when the tasks are initially randomly distributed. Compared with existing proactive task migration technique, this approach generates less hotspot, less migration overhead with negligible performance overhead.

VI. Specifications

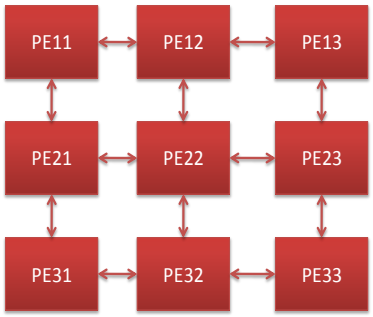
SOFTWARE TOOLS:

1. Implementation Language: Verilog
2. Simulation Tool: ISIM
3. Synthesis Tool: XST, Xilinx ISE 13.2

HARDWARE TOOL:

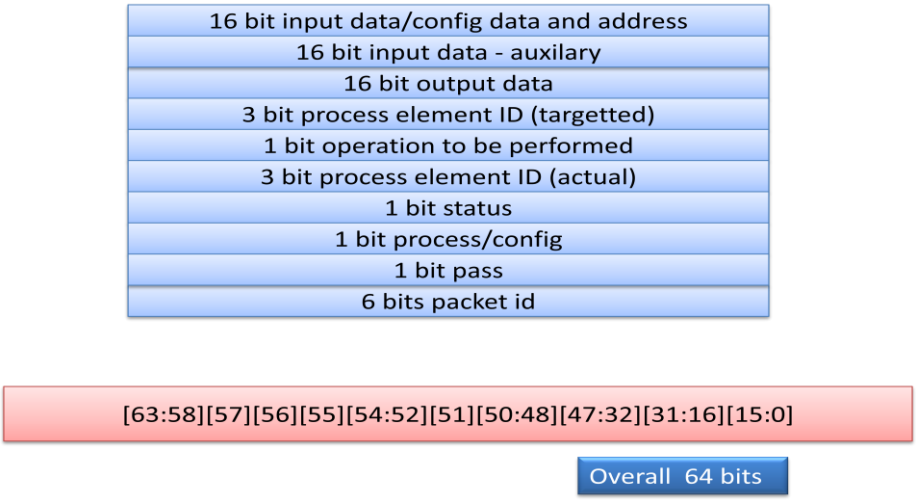
1. FPGA Implementation on: SPARTAN 3A Kit

Network of processing elements



The network of processing elements consists of nine PEs and all are internally connected to each other.PE11 is connected to PE12 and PE21, PE12 is connected to PE11, PE13 and PE22, PE13 is connected to PE12 and PE23.Similarly other PEs are connected to its neighbour PEs. The PE22 is connected to four processing elements those are PE12, PE21, PE23 and PE32.

Processing element is a generic term used to reference a hardware element that executes a stream of instructions. In some programming environments, each workstation is viewed as executing a single instruction stream; in this case, a processing element is a workstation. A different programming environment running on the same hardware, however, may view each processor or core of the individual workstations as executing an individual instruction stream.



The above figure shows the packet format, each packet consists of 64 bits which is divided into ten sections. A task is encapsulated as a packet. Task in this project is to encrypt or decrypt data.

1. **PROCESSING UNIT**

S-AES – Simplified Advanced Encryption Standard. Processing unit does encryption and decryption of the data based on S-AES.

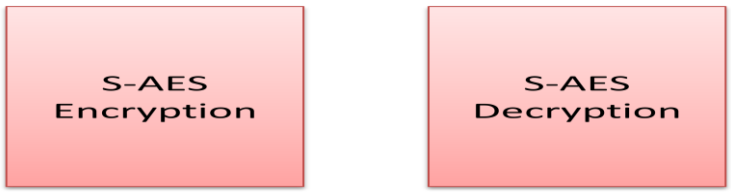
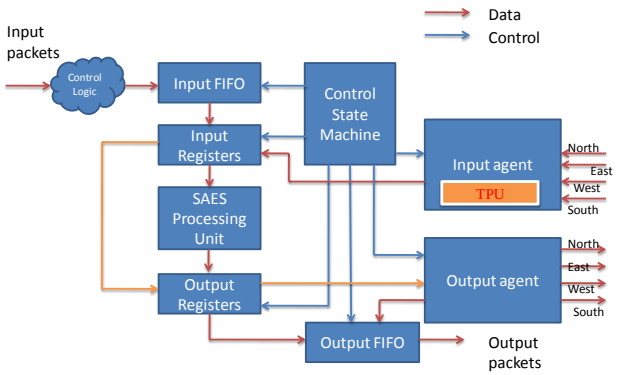


Fig Processing Unit

The processing unit consists of S-AES encryption and decryption and it process the data based on 51st bit of data packet. If 51st bit of data packet is high then the processing unit encrypts the data otherwise decrypt the data.

VII. Design Of Processing Element

Processing Element Design



Accepting packets into FIFO for processing. Data packet is accepted if the temperature of the core is less than the specified threshold temperature. Here the threshold temperature is set to 90.Input FIFO is required because input data can come at a rate different from the processing speed. Hence it should be buffered in the FIFO.
State machine reading the packet from the input FIFO and writing into the registers. Registers are in-turn read by SAES processing unit for processing. Upon completion of the process control state machine issues a signal to frame a new packet with newly processed data and status and then the signal is issued to capture the output of the processing unit to Output FIFO.

OUTPUT AGENT

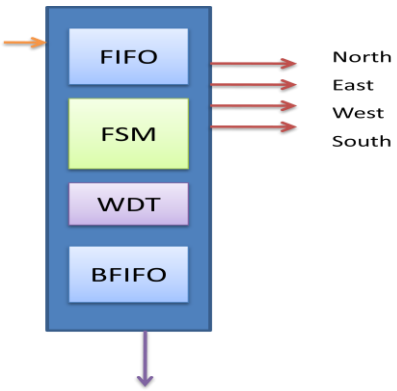


Fig Output Agent

The output agent act as master which send request to its processing element in all 4 directions. Initially packets will be stored in FIFO and then it is sent with the control of FSM and it wait for a grant for a particular time interval which set by WDT. If there is a grant then the output agent sends the packet for processing otherwise packet will be sent to BFIFO.

INTPUT AGENT

Master slave connections between input and output agents of any two processing elements. It is true for all directions.

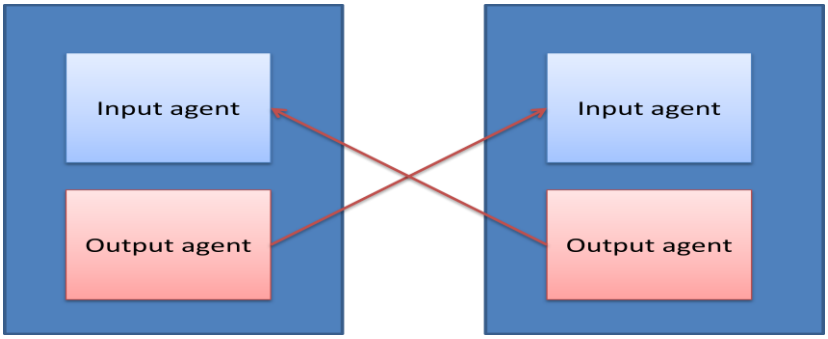


Fig 6.12 Connection between input and output agent

The above figure shows the connection between input and output agent. The output agent of a PE act as master and input agent of a PE act as slave. The output agent send request to the input agent for packet processing. If it gets a grant from input agent then the packet will be processed by the respective input agent. Each PE consists of input and output agent to act as slave and master respectively.

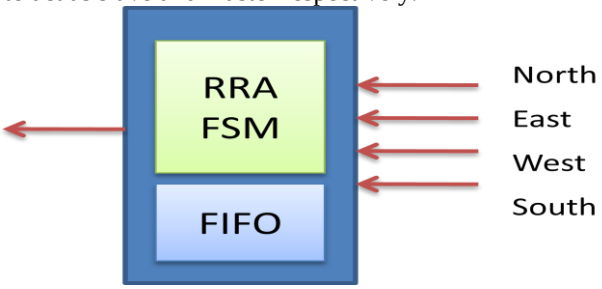


Fig Input Agent

The input agent act as slave which gets request from neighboring processing element in all 4 directions. It accepts or rejects the request depends on the temperature. When the temperature is less than or equal to

threshold, it accept the request based on Round Robin Algorithm and process the packet but when the temperature is greater than the threshold then the request will be stored in FIFO.

VIII. Temperature Processing Unit

Temperature processing unit (TPU) is introduce in this paper work, in order to predict the temperature of each processing element (PEs) and inputs to the TPU are clk,reset,grant provided to its neighboring PEs and main stream data acceptance. it consists of Tracker & WDT,where tracker is a register which keeps track of acceptance provided to its neighboring PEs as well as the input accepted from the main stream. If grant is ensued to any of its neighboring PE or accepting data from its main stream then the temperature rises, which indicates that its workload is more. in WDT reset timeout constant in order to predict the temperature in specified time interval. If none of the input conditions are satisfied then the WDT reset to its initial value.

IX. Simulation Results

1. Processing Element act as Slave

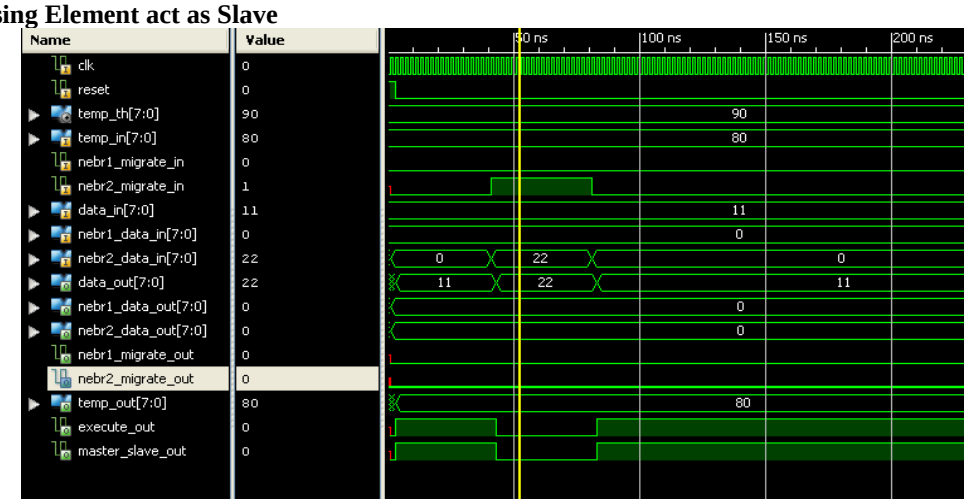


Fig.: Processing Element act as Slave

In this case input temperature is less than the threshold temperature hence it act as a slave and performs the task for neighbor 2 as there is a migration request from it.

2. Processing Element act as Master

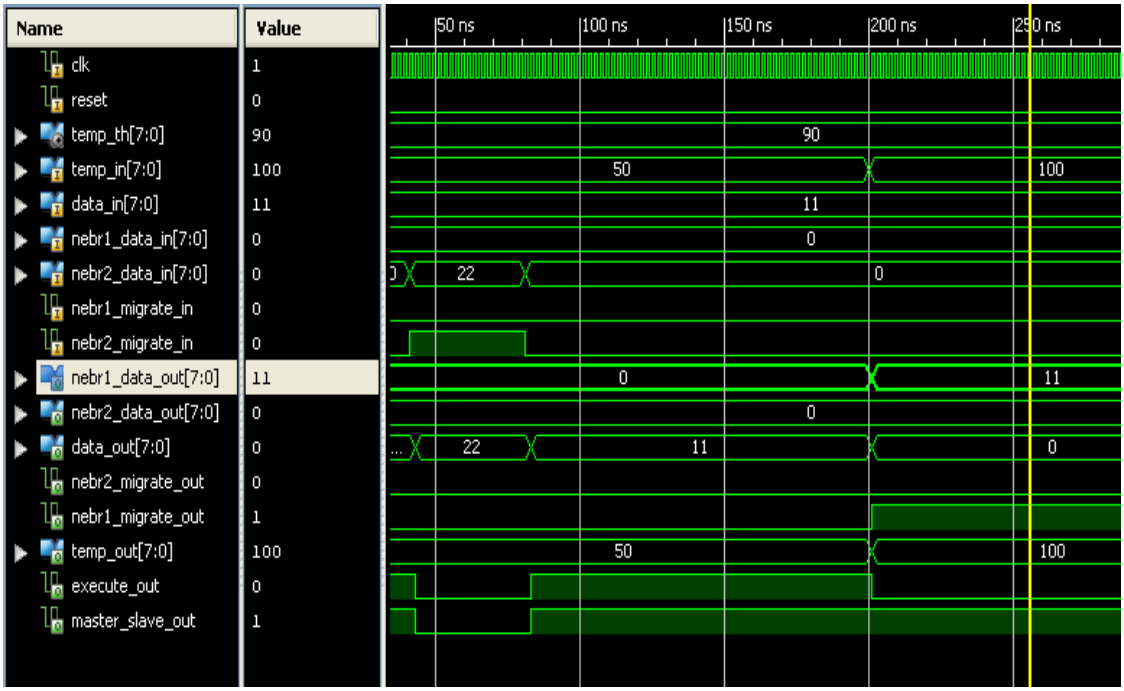


Fig.: Processing Element act as Master

In this case input temperature is greater than the threshold temperature hence it act as a master then check for neighbor 1 and 2 migration request which is zero hence neighbor 1 performs the task and give the result .

PROCESSING ELEMENT FINAL RESULT

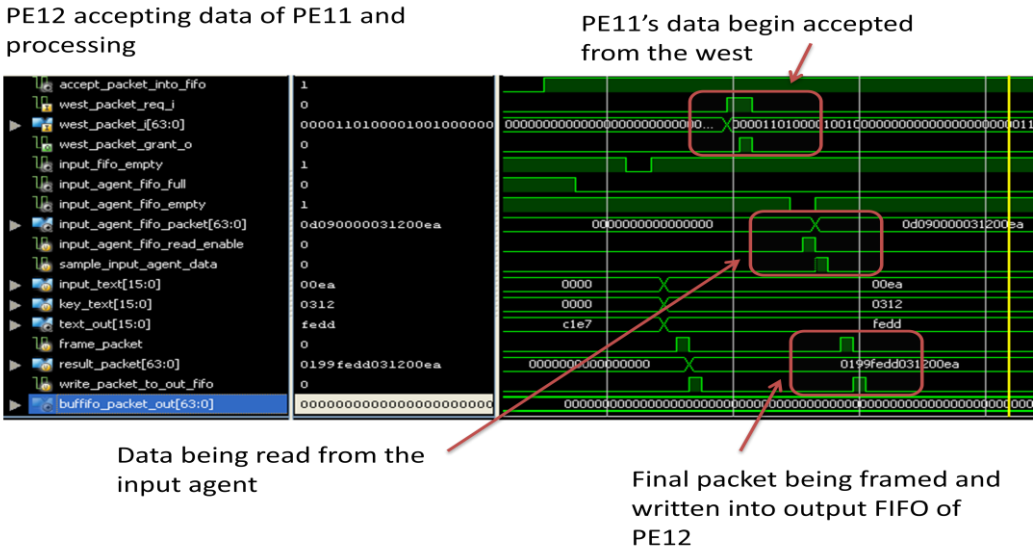


Fig: Processing Element Final Result

X. Conclusion

In this paper , we proposed a temperature based prediction unit for nine-core system. In this framework no centralized controller is required. Each core has an input and output agents which monitors the core temperature, communicates and negotiates with neighboring agents to migrate and distribute tasks evenly across the system. We also proposed temperature based prediction unit that can be used not only for future temperature prediction but also for agents to evaluate the rewards of proposed migration offers.

References

[1] J. Howard, S. Dighe, Y. Hoskote, S. Vangal, D. Finan, G. Ruhl, D. Jenkins, H. Wilson, N. Borkar, G. Schrom, F. Pailet, S. Jain, T. Jacob,S. Yada, S. Marella, P Salihundam, V. Erraguntla, M. Konow, M. Riepen, G. Droege, J. Lindemann, M. Gries, T. Apel, K. Henriss, T.Lund- Larsen, S. Steibl, S. Borkar, V. De, R. Van DerWijngaart, and T.Mattson, "A 48-Core IA-32 message-passing processor wit DVFS in 45 nm CMOS," in *Proc. Int. Solid-State Circuits Conf. (ISSCC)*, 2010,pp 108–109.

[2] T. Ebi, M. Al Faruque, and J. Henkel, "TAPE: Thermal-aware agentbased power economy for multi/many-core architectures," in *Proc. Int.Conf. Comput.-Aided Design (ICCAD)*, 2009, pp. 302–309.

[3] R. Jayaseelan and T. Mitra, "Dynamic thermal management via architectural adaption," in *Proc. Design Autom. Conf. (DAC)*, 2009, pp.484–489.

[4] D.Wentzlaff, C. Gruenwald, N. Beckmann, K.Modzelewski, A. Belay,L. Youseff, J.Miller, and A. Agarwal, "A Unified operating system for clouds and manycore: FOS," MIT-CSAIL-TR-2009-059, Nov. 2009.

[5] W. Huang, M. Stan, K. Sankaranarayanan, R. Ribando, and K.Skadron, "Many- core design from a thermal perspective," in *Proc.Design Autom. Conf. (DAC)*,2008, pp. 746–749.

[6] S. Borkar, "Thousand core chips—A technology perspective," in *Proc.Design Autom. Conf. (DAC)*, 2007, pp. 746–749.

[7] A. Coskun, T. Rosing, and K. Whisnant, "Temperature aware task scheduling in MPSoCs," in *Proc. Design Autom. Test Euro. (DATE)*,2007, pp. 1659–1664.

[8] P. Michaud, A. Seznec, D. Fetis, Y. Sazeides, and T. Constantinou,"A study of thread migration in temperature-constrained multicores,"*ACM Trans. Arch. Code Optim. (TACO)*, vol. 4, no. 2, pp. 9-1–9-28,Jun. 2007.

[9] X. Chen, C. Xu, R. Dick, and Z. Mao,"Performance and power modelling in a multi programmed multi-core environment," in *Proc. Design. Autom. Conf. (DAC)*, 2010, pp. 813–818

Navya Vipin received the BE degree in Telecommunication Engg from VTU Belgaum,India,in 2005 and pursuing the M.Tech degree from the department of Electronics and Communication Engg, VTU Belgaum ,India. She is currently working as Asst.Professor with the department of Telecommunication Engg,MVJCE,Bangalore,India Her research interest includes VLSI design and embedded system and optimization for multi core system.

Shivayya Gadag received the BE degree in Electronics and Communication Engg from VTU Belgaum,India,in 2010 and received M.Tech Degree from the department of Electronics and Communication Engg, VTU Belgaum ,India.He is currently working as Asst.Professor in the department of Telecommunication Engg,MVJCE,Bangalore,India. His research interest includes VLSI design and embedded system and optimization for multi core system.