EasyChair Preprint

№ 1388

Temperature Sensitive Microprocessor Design (To Reduce Heat Generation and improve performance)

Tamanna Afroze and S. M. Farhad

August 9, 2019

Temperature Sensitive Microprocessor Design

To Reduce Heat Generation and improve performance

Abstract — **Microprocessors are designed with very tiny microchips and heat induced due to operation makes the chip deteriorate their performance in many extents. Heat causes a portion of chip-area to get heated which degrades operation of many applications in chip-level. This work wants to make a watcher to watch the applications running in pipeline, and then by utilizing slack time in hardware level this work wants to improve performance of the processor. In this paper, this work proposes two new heat-control mechanisms to improve performance, one is at operation-level and the other is at architectural-level. At operation-level, this work proposes a prediction mechanism to predict the useful operations inside the microprocessor that performs as a sink for heat dissipation. At architectural-level, this work proposes a drain system for heat dissipation. The proposed prediction and drain mechanisms will reduce heat generation and thereby increase performance. This work has simulated the proposed system using Matlab and observed that the system works perfectly well. Java program has been devised to take care of fault tolerance and fault detection.**

Keywords- Heat detection, heat control, power dissipation, drain system, runtime, fault tolerance, fault detection.

I. INTRODUCTION

Microprocessors are the base part of a computer of our day-to-day computing system. They are the core part in the name of computing and communication engineering and large-scale data integration system except some embedded systems. The necessity of computing-power is growing day-by-day in many dimensions. Due to operations performed by processors heat is generated. Heat generation limits processor performance and limits the number of transistors incorporated in a single chip. Research on how to reduce the generated heat, is gaining popularity due to its many aspects of improvement [1, 2, 3, 4, 5, 6, 7].Different layer of microprocessor including clocked pulse is also playing significant role in performance improvement. As the performance improvement continues so does the heat generation both in chip and architectural level [6]. Each of these works has addressed several aspects of improving performance that have induced heat. Bit-partitioning mechanism improves performance, but the proposed chip-level architecture increases fabrication area [5]. Moreover, the super-scaling operation like decoder spacing expansion in this system requires logic states to remain active simultaneously that in turn increases heat at chip-level.

Time may come when microprocessors will be separated from the whole machine and every computing will be done in distributed way, otherwise, recently known as cloud computing. We are envisioning being making the processor chip contingent with heat control and heat dissipation. From our high school physics, we know the very important equation of heat and power relation, that is,

$$\boldsymbol{H = Pt}$$

Where H is heat, P is Power and t is time.

Very little work addressed the issue of operational techniques for heat controlling. Architecture-level issues controlling by *register-transistor logic* is discussed in [15]. We addressed the logical gates' operation for performance improvement and thermal awareness. Some operations can be predicted early which can increase performance and also for long distance travel or for propagation delay induced by many gate transition heat can increase which often causes programs to run slowly.

Microchip's architecture is designed using universal gates, that is, NAND and NOR gates. If we can design our universal gates in such a way that common logics come under a simple union of an integrated chip, and heat dissipation on the chip is attached within that particular chip in such a way that, we are designing dynamically controlled heat detector by detecting power. In this way, we can make our processor chips cool and, simultaneously longevity may increase. Concurrently, it will give runtime improvement or working time improvement. Runtime is the time which is the active time of a particular operation or a particular mechanism. It indicates the operation's improvement or how much time a circuit remains active which in turn makes the circuit heated. This work has shown how runtime affects chip's output in operation and heat control.

NAND and NOR gates are designed in the chip level by the use of inverter logic gate. If we can add some logic for inverters, so that inverter gates will be aware of temperature and will take necessary operation for temperature minimization, then we can achieve some good logic for temperature minimization with low cost. A new method of heat detection with RTL logic in the chip-level layer absorbing the generated heat is presented in [3]. Changing clock frequency and checking delay or overhead of operation is essential in performance improvement [5]. But extra logic for this operation may increase time. We want to predict some operation early to check the operation of some gates and then make the performance improved, heat reduction simultaneously.

Use of different metal instead of silicon is also a think for temperature management. Not always we can use other metal with optimum cost. Some research is necessary for gaining

some good effect. This thought is motivated by the general observation of our daily activities. If we store water in the bucket, the water remains cold or hot depending on the surrounding environment. So, if we can flow some thermally cold metal beside the chips or tiny gates it can significantly reduce temperature with improving performance of running application though.

In this paper, we propose two mechanisms that reduce the heat inside processor. For circuit level optimization, we need to check the level of minimum voltage, which can keep the logic states without attaining the highest voltage levels. We can add capacitor for this purpose. In the second methodology, if we can add different metal instead of the silicon chip for storing the voltage levels we can get an optimum result for temperature management.

The main message of this paper we want to deliver is that smart viewing of applications and runtime improvement of those running application can significantly improve temperature reduction process and simultaneously improve performance. In this work we have also considered smart error and fault detection with tolerable fault tolerance. Watcher needs to take care of performance of the running application, and tries to find out the slack time and stall times to inject other necessary instruction checking to improve performance.

Rest of the paper is organized as follows: Section II points out the related works. Section III illustrates our proposed techniques in details, and finally the paper concludes in Section IV.

II. RELATED WORKS

With the advent of newer techniques of multiprocessor design, temperature and power has become a crucial issue in chip designing. Every design focus on power consumption, power dissipation and heat tolerance to improve the microchip designing keeping the performance of program execution better and reliable. Power and heat management is not only related to microchip designing mechanisms, but also essential for network-chip designing process.

Most of the work addressed the internal parameters of microchips. Simultaneously, register file, memory store/load operation, reading registers, and cache architecture also came into the heat control and detection mechanism.

Heat control by using separate hardware architecture and bit-partitioning method is illustrated in [1]. The author considered an extra efficient mechanism by inducing newer memory chips for accessing register files. Bit-partitioned Register File (BPRF) considered their designing mechanism from basic cache organization mechanism. It is designed for, in fact, designed based on a conventional dynamically scheduled superscalar processor. They showed how much energy is consumed in the separated bank of register files, and bit partitioned method, while preserving early de-allocation of registers usage for processors performance. Energy, that is, otherwise related to power consumption greatly improves performance of microprocessor in this methodology.

In [3], the authors considered heat detection and control mechanism for Architectures. They designed the hot-leakage mechanism for the micro-architecture going inside into the main digital logic designing gates, like NAND or NOR's CMOS fabrication level. They showed how temperature leak can be controlled in caches by using several techniques, like, lowering the Quiescent V_{dd}, multiple threshold CMOS, Drowsy caches, hot-leakage parameters. Dynamic thermal management by monitoring chip-wide temperature at run-time and dynamically inducing power reduction schemes is discussed in [2].Reducing register ports for memory read/write operation in explained in [4]. Thermal relationship and thermal management for subarrayed data cache has been discussed in [2, 13, 22].

Skadron proposed most of the temperature control mechanisms in literature [3, 11, 15, 19]. His several publications show many different improvements for heat control. In [15], the authors showed a micro-architecture, which is temperature aware. In brief, they showed chip-level hardware techniques for good illustration of both the benefits and challenges of runtime thermal management. They named the architecture as hot-spot. In this paper, they showed how to find out the het-induced chip and then to reduce the heat of the detected chip by using RTL circuits. In [8, 9], they showed die area for fabrication of microchips and several parameters that are necessary for die area fabrication.

Heat has a direct relation to time, which has been already said in introduction of this paper. [10] Takes attention of time-variant design issues for micro-architecture.

The deep level of micro-architecture is illustrated in [5, 6, 7, 8, 12]. The logic gates have induced delay. Any state transition causes delay for propagation from input to output [5]. Totem pole for inverter circuit has been induced in the CMOS logic design in order to decrease propagation delay. Now-a-days, we do not have a single processor in our computer. This work has considered simultaneous multithreaded processors, chip multiprocessors, and many cores. Induced heat and their checking and minimization are discussed in [6, 7, 8].

In [17], the author showed power, thermal view for multicore. They described the temperature issue as spatial distribution. They showed how heat is detected for the floorplan of the system. It is important to care about the geometric characteristics of the floorplan. They summarize all those impacting on chip heating as follows:

1. *Proximity of hot units*. If two or more hotspots come close,this will produce thermal coupling and therefore raise the temperature locally.

2. *Relative positions of hot and cold units*. A floorplan interleaving hot and cold units will result in lower global power density (therefore lower temperature).
3. *Available spreading silicon*. Units placed in such a position, that limits its spreading perimeter, will result in higher temperature, e.g. the units placed in a corner of the die.

Finally, we want to conclude this section by saying something about fabrication of silicon chips. Now-a-days, different fabrication methods are used for chip design. In [15], they said some very important step of the fabrication process. They said, chip today are typically packaged with the die placed against a spreaded plate, often made of Aluminum, copper, or some other highly conductive metal, which is in turn placed against a heat sink of Aluminum or copper that is cooled by fan. In introduction section, we said about using different metal for designing the fabrication of chips or for designing the drain system for heat control. In strategy section, we will say in detail what things are important for our proposed mechanism.

III. TEMPERATURE SENSITIVE DESIGN

Our principal goal is the make the processor design using some separator logic gate. Most of the part of microprocessor are designed by universal gates NAND and NOR. These gates are designed by basic gates and including an inverter circuit in front of the gates. We can add heat dissipation mechanism in the inverter circuit. We think of making the gates design with a different type of metal, and the chip can sustain 3 logic states instead of 2. It will improve voltage switching easier and fast, and we can drain the tri-state easily. From power calculation we can check the heat, as we know, Power equation is:

$$\boldsymbol{P = VI}$$

Where P is Power, V is Voltage, and I is Current.

How the temperature is related to these mathematical equations we will describe in later of this section. Briefly we can say, the less time the chip will remain on (bit state 1), the less will be the voltage and current flow. That way power will reduce. When power reduces it will decrease heat. Moreover, we have said already, the chip will remain on for shorter period. So, time will be small. We want to make the change in bit level in less than nano-scale level. We want to optimize our performance of the running application, that is, we want to reduce runtime. In this way, we want to contribute in supercomputing. Our programming simulation will tell details about it and will show light inside the nano architecture.

Main design logic is tested to keep in mind about following improvement of microchips.

1. Register induced die expansion controlling
2. Heat reduction by using different metal
3. Performance improvement
4. Reduction of gate delay
5. Checking heat control inducing sampling switch
6. Modifying heated area detection mechanism
7. Predicting, checking and controlling thread operation for Simultaneous Multithreaded Processors
8. Core switching heat transfer for many core, multi core and chip multiprocessors
9. Prediction about load/store operation to minimize heat transfer

Temperature aware Microprocessor designing work [15] addressed the issue of detecting thermally hot area and making changes by using RTL. Register needs larger fabrication area while we are designing using buffer to occupy smaller area in fabrication. This is one of the most important issues addressed in this work.

We want to design heat reduction by using different metal. We can design the substrate of the fabrication die using metals which has property of become cooled easily and quickly. We want to make prediction logic for some operations which will improve performance of microprocessor, reduce propagation delay of gates and will improve memory searching mechanism for load/store operation.

The model we simulated in Matlab is included in figure 1 and the corresponding code is included in Appendix section. The main thing we want to show by simulation is the runtime of the proposed system. Our proposed system comprises of the buffer and the sampling switch which we also named as “system” for simulation. We want to proof that the gate delays of the proposed system do not increase runtime of the existing system. And this proposition is compared in STEP 4 and Step 5 of the simulation. This work compared the proposition from STEP 1 to STEP 5 using different inputs. Firstly, this work has given 0 and 1 as input to check whether the system works finely or not. Later we provided a constant 1 as an input to check the runtime and output as the 0 output of the main circuit conflicts with the default output of the system.

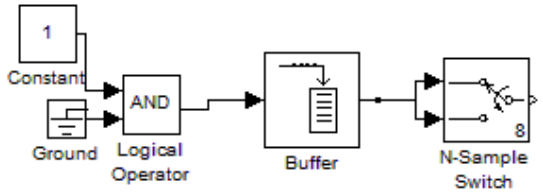
We want to simulate a logic gate’s output which is input for the buffer. We are considering universal gates in this respect as design of chip comes with universal gates. That’s why, we are considering AND gate in the simulation as a symbol for control logic’s output of Microprocessor.

The mechanism of Figure 1 is illustrated as follows. Register occupies larger die-area. We want to shrink the size of die fabrication area using buffer. This is a variation of [15]. The logical output of the logic gates will be directed to the input of buffer. Buffer will store the data and these data will be directed to operation by N-sample switch. As buffer is storing data, it will minimize induced heat by bit-partitioned method. So, it is an important variation of [1]. Thought may come that sampling-switch may grow the die-size. But designing sampling switch is very easy by using a XOR gate.

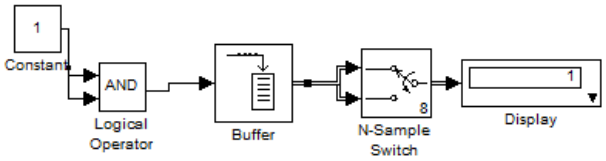
Proposed System:

STEP 1: Simulation runs for 10s and the circuit works well. In this step this work has just tested the system using some

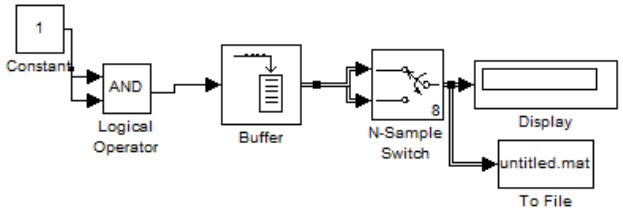
basic terminology, like AND logic gate with input 1 and 0, buffer to store the value, and switch to relay the output to the another system.



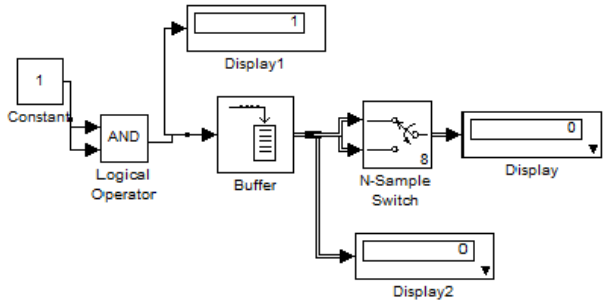
STEP 2: Simulation runs for 10s and the output of the simulation is showed in the display box. Buffer and sampling switch works perfectly according to the theoretical way.



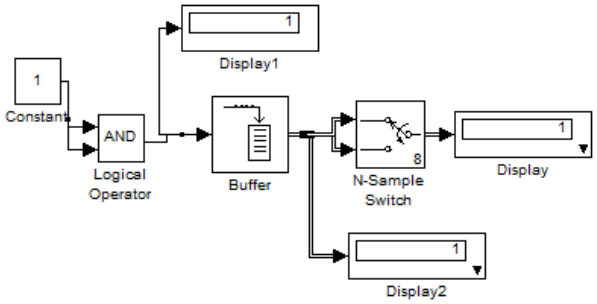
STEP 3: Simulation runs for 5s and output transferred to a file and also in the display box. The output of N-Sample switch cannot be transferred to the file, and output cannot reach to display. It is only for testing purpose.



STEP 4: Simulation runs for 0.05s and we do not get the desired output. AND gate works and the time finishes. The output of AND gate cannot be stored in buffer for why we don't get any output in "display" and "display1".



STEP 5: Simulation runs for 0.5s and we get desired outputs in the appropriate display box.



After simulating these stepwise processes we have found our basic proposed system which is as follows:

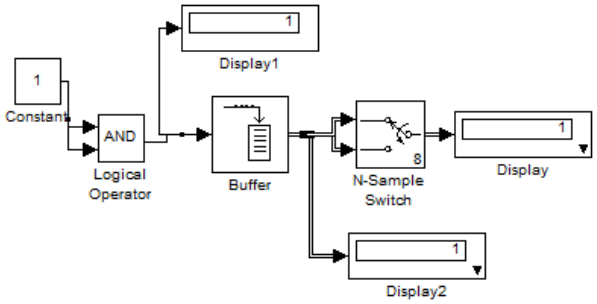


Figure 1: The basic circuit to replace inside chip for heat control

Comparison with bit-interleaved Mechanism:
We are designing *the bit-interleaved scheme* for our proposed mechanism and the working principle of the bit-interleaved scheme maintained using the "display" box. The logical operator gives the output which is compared with the bit-position to store and stored in the buffer and sampling switch is delivering the output comparing the set-bit and the level zero. The comparison gates are the building block of the bit-interleaved system. We are using same input and same simulation time to check the working methodologies. Our proposed system resides in this mechanism. The bit interleaved system according to our design is as follows:

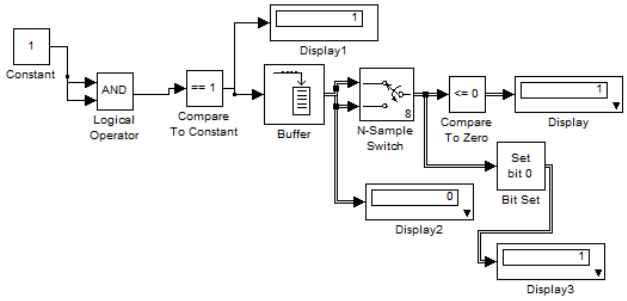
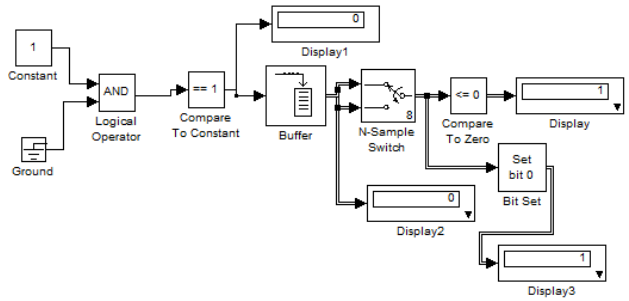
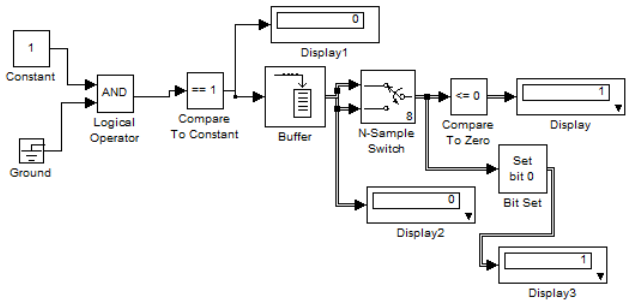


Figure 2: Bit Interleaved System

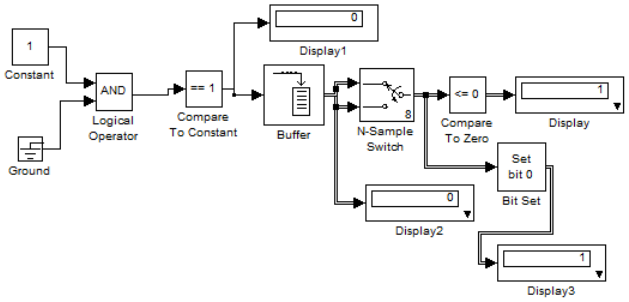
STEP 1: Simulation time 10s. This is used to check the system. "Display" block at every point is showing the output.



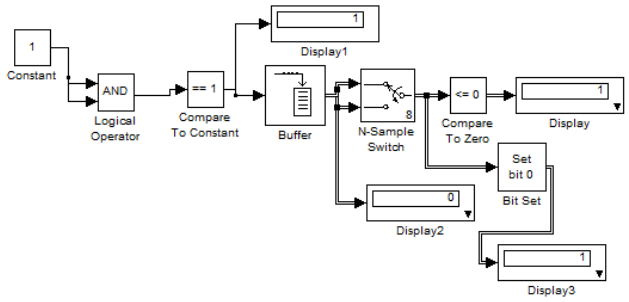
STEP 2: Simulation time 5s. It is used to compare with our system.



STEP 3: Simulation time 0.5s. System ends at the "Display 2" block. Our system works till final output. We get better output.



STEP 4: Simulation Time 0.05s. System give simulation's output till the "Display1" block output. Rest of the system gives default output. This system needs much area in fabrication and because of digital blocks of comparison, it gives default wrong output. As many blocks need to active for a large amount of time heat increases in the system.



Logic gate induces propagation delay. So, we can also use flip-flop type switch/latch for better performance improvement. Latch has a problem of giving transient output while input changes. Not only transient output, latch has delay incurred due to operation [5]. So, we can check the current and previous output from the operation of Flip-Flop. JK Flip-flop is good for this case of keeping current and previous state change. Operating table of JK Flip-flop is illustrated in Appendix.

In order to keep track of logical operation we feed the input of logic gates in the buffer input in a reformed way. We want to design a special prediction mechanism for logical gate's input in order to minimize gate delay. Typical logic gates propagation delay differs from 20ns to 40 ns. In order to design the prediction logic and feedback logic, we want to introduce JK Flip-Flop's characteristic function. Previous state is documented by Q_t(which is output of AND gate in the Figure 1 and in the Figure 4 (U3A)), and J, K, and Qt determine next state in the prediction logic circuitry. Simultaneously, we want to change the trigger, which is, clock input. We want to change the clock trigger in such a way that it remains active for 5ns to 8ns. If the clock can be designed in this way, then it will make some improvement to the gate delay. Gate delay will be reduced a little bit as rest of the input of the flip-flop is already there for action.

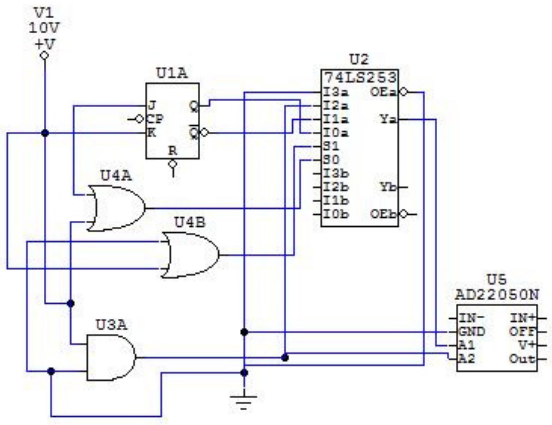


Figure 3: The simple prediction logic

In the above figure (Figure 3), this work has shown the prediction logic, input and output on time t and time (t+1). The K input is for reset and this input will be fixed at constant voltage source. The J input will be fixed at the output of the logic gate which is U3A for Figure 3 on time t.

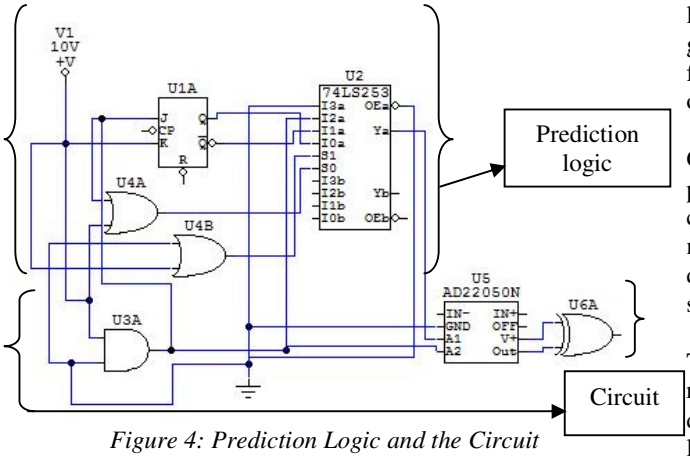


Figure 4: Prediction Logic and the Circuit

Illustration of Figure 4:
The upper portion indicated by a line in Figure 4 is the prediction logic and the lower portion is the circuit for which prediction is designed. The buffer in the circuit (AD22050N) is used for retrieval of output of the prediction/actual output. The input of buffer A1 is the output of prediction logic and the input of buffer A2 is the actual output of the operating circuit. The operating circuit can be anything like output of ALU, memory calculation output for load/store operation, branch prediction logic, task switching operation, control logic for program execution, etc. Buffer and the xor gate works as a sink which absorbs heat and reduces heat generation by storing for a short interval of time.

The gate U1A is the JK Flip-flop. The gate U2 is a multiplexor. The gate U4A and U4B is used for designing the selectors input for the multiplexor. These gates, namely U1A, U4A, U4B, and U2 are used for designing the prediction logic for the subsystem. The input of J of gate U1A is output of gate U3A on time t. The input of K of gate U1A is constant positive voltage, V_{dd}. The inputs of U4A are J and first input (I1) of the gate U3A.The inputs of U4B are K and second input (I2) of the gate U3A. The output of gate U4A is the Selector S0 (Least Significant Bit) of multiplexor U2 and the output of gate U4B is the Selector S1 (Most Significant Bit) of multiplexor U2.

U4A		U4B		U2	
J	I1	K	I2	S0	S1
0	0	1	0	0	1
0	1	1	1	1	1
1	0	-	-	1	1
1	1	-	-	1	1

Table 1: Operational table for prediction logic

From the table above we can see that selectors (S0 S1) will select input I3a of multiplexor for most of the time which is predicted value for the circuit, and sometimes I2a which is actual output of the circuitry at time (t+1). Depending on the logic circuitry's design we choose the optimum output for the gates as predicted output. If transient input occurs in any case for a little period of time it will select multiplexor's input I0a or I1a which are output on time t.

Gate U3A, U5 and UGA is the basic circuit for which the prediction logic is designed. Gate U3A is for designing the control logic. By control logic we mean any decision from microprocessor's operation. Gate UGA is an XOR gate which designs the circuitry for N-Sample switch of the basic design system.

The advantage of this system is that we are capable of reducing propagation delay of logic gates, as we are determining the output of time (t+1) on time t. It makes the logic gates to remain active for less period of time and in this way we can reduce the power consumption by logic gates and heat generation will be minimized. In this way operational-level output detection minimizes total heat generation for the entire circuits and whole microprocessor. This work has checked the output of this system in a very tiny measure level in *Circuit Maker Simulator* to detect whether prediction get lost or gives the correct result and works as a sink. We have got the result that output level is operational, which means our predictor detects output of the logical operation for many cases and with the clock change it gives prediction. Buffer works perfectly as sinks, and this work has both **performance improvement and heat reduction**.

Recently, SMT (Simultaneous Multithreaded Multiprocessor) and CMP (Chip Multiprocessors) are gaining popularity. If we can predict the exact output of the control circuitry it will reduce thread excitation unnecessarily and will get benefit from resource utilization which can be extended as future work.

Applicability:
This type of mechanism is not only necessary for control-logic design of microprocessors, but also is an essential subsystem for network-on-chip. We can leave some work for Operating System. We can design a patch for thread's activity to control heat in case of Simultaneous Multithreaded Multiprocessors.

Fault Analysis of this work:
Analysis of fault detection and fault tolerance is a necessary work for this type of architecture modification. The main things we want to take care in this work is checking fault tolerance can be done whether smartly. The main points we want to point out in this work are as follows:

1. The motivation behind the research work is to find out the propagation error within combinational circuits and sequential logic gates.
2. To consider stuck at 0/1 as a boon not as a danger for the circuits.
3. Analysis of dependability of applications and instructions with respect to fault tolerance systems.

With this end in view we modified our previous proposed design in the following way:

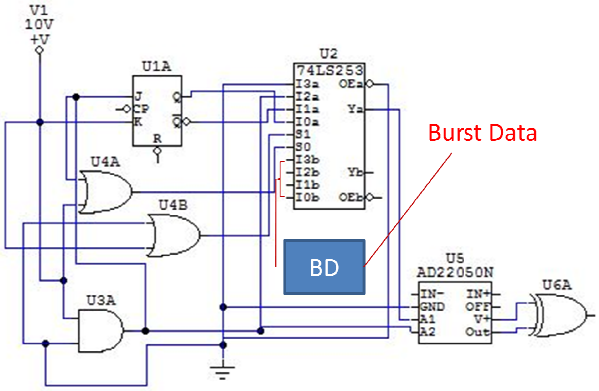


Figure 5: Checking fault and mechanism to tolerate fault.

Some simple examples of fault tolerance is included with the following figures and description:

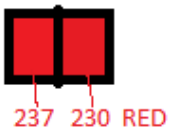


Figure 6: RGB values bit interpretation checking in order to fault detect

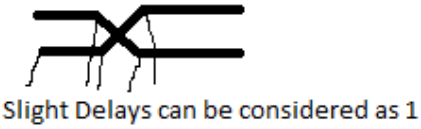


Figure 7: Waveform shows data in operation and detection of its slight delay in order to reduce fault.

Figure 8: Slight change to detect edge in order to detect fault.

In this way, we can detect slight change and we can omit the change of bit levels and fixed the value of bits to either 0/1 to reduce bit-level fluctuations and in that way we can improve performance of the processors. Simultaneously, the less on/off of the processors chips will be made, the less the temperature increment will happen.

IV. Conclusion

In this paper, we propose two techniques to reduce the generated heat to microprocessor. The first technique is the prediction mechanism that predicts the required operations which intern works as a sinkhole and thus reduces the heat. The second technique is to fabricate the processor with new metal that works as a heat drain system reducing significant heat. Our proposed techniques reduce the generated heat in the microprocessor and thus improve performance.

We will further extend our work for development of an integrated system. Considering each and every operation inside microprocessor and designing a complete control logic that improves performance and reduces heat/temperature increment. Though the work considers different architectures of microprocessor, it will be extended specific further for Simultaneous Multithreaded Multiprocessor and Chip Multiprocessors.

References

[1] M. Kondo and H. Nakamura, "A Small, Fast and Low-Power Register File by Bit-Partitioning," Proceedings of the 11th Int'l Symposium on High-Performance Computer Architecture,2005,pp. 1-10.

[2] J. Hu, K. John, and S. Wang, "Thermal-Aware Subarrayed Data Cache Microarchitectures," International Journal of Intelligent Control and Systems, Vol.13, No. 4, December 2008, pp. 251-263.

[3] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan, "HotLeakage: A Temperature-Aware Model of Subthreshold and Gate Leakage for Architects," University of Virginia Department of Computer Science Tech. Report CS-2003-05.

[4] I. Park, M. D. Powell, and T. N. Vijaykumar, "Reducing Register Ports for Higher Speed and Lower Energy", In Proceedings of MICRO, 2002.

[5] M.S. Hrishikesh, N. P. Jouppi, K. I. Farkas, D. Burger, S. W. Keckler, and P. Shivakumar, "The Optimal Logic Depth Per Pipeline Stage is 6 to 8 FO4 Inverter Delays", In the proceedings of the 29th International Symposium on Computer Architecture.

[6] J. Donald and M. Martonosi, "Temperature Aware Design Issues for SMT and CMP Architectures", Workshop on Complexity-Effective Design, 2004.

[7] J. Donald and M. Martonosi, "Techniques for Multicore Thermal Management: Classification and New Exploration", ACM SIGARCH Computer Architecture News, 2006.

[8] Sheng-Chih Lin, N. Srivastava and K. Banerjee, "A Thermally –Aware Methodology for Design –Specific Optimization of Supply and Threshold Voltages in Nanometer Scale ICs", International Conference of Computer Design, 2005.

[9] G. H. Loh, Y. Xie, and B. Black, "Processor Design in 3D Die-Stacking Technologies," IEEE Computer Society, 2007. Pp. 31-48.

[10] H. Yu, Yu Hu, C. Liu, and Lei He, "Minimal Skew Clock Embedding Considering Time Variant Temperature Gradient," In the Proceedings of ISPLD, 2007, Austin, Texas, USA.

[11] Z. Qi, B. H. Meyer, W. Huang, R. J. Ribando, K. Skadron, M. R. Stan, "Temperature-to-Power Mapping," In the Proceedings of ICCD, 2010.

[12] S. Borkar, "Thousand Core Chips-A Technology Perspective," In the Proceedings of DAC, 2007, San Diego, California, USA.

[13] J. K. John, J.S. Hu , and S. G. Ziavras, "Optimizing the Thermal Behavior of Subarrayed Data Caches," International Conference of Computer Design, 2005.pp. 625-630.

[14] S. Borkar, "Design Challenges of Technology Scaling," IEEE Micro, 1999. Pp. 23-29.

[15] K. Skadron,M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-Aware Microarchitecture," International Symposium on Computer Architecture, 2003.

[16] O. S. Unsal, J. W. Tschanz, K. Bowman, V. De, X. Vera, A. Gonzales, O. Ergin, "Impact of Parameter Variations on Circuits and Microarchitecture," In the Proceedings of IEEE Computer Society, 2006, pp. 30-39.

[17] M. Monchiero, R. Canal, and A. Gonzalez , "Design Space Exploration for Multicore Architectures: A Power/Performance/Thermal View," In the Proceedings of ICS, 2006, Queensland, Australia.

[18] Moris Mano, *Digital Design,* Third Edition.

[19] K. Sankaranarayanan, S. Velusamy, M. Stan, and K. Skadron,"A case for thermal-aware floorplanning at themicroarchitectural level."*Journal of Instruction LevelParallelism 8* , 2005, pp. 1-16.

[20] Web Tools, Online Physics Material Tutorial.

[21] Matlab, Mathwork's Simulation Tool.

[22] B. Zhai, S. Pant, L. Nazhandali, S. Hanson, J. Olson, A. Reeves, and M. Minuth, "Eenergy-Efficient Subthreshold Processor Design ," In IEEE Transactions on Very Large Scale Integration (VLSI) Systems, VOL. 17, No. 8, August 2009.

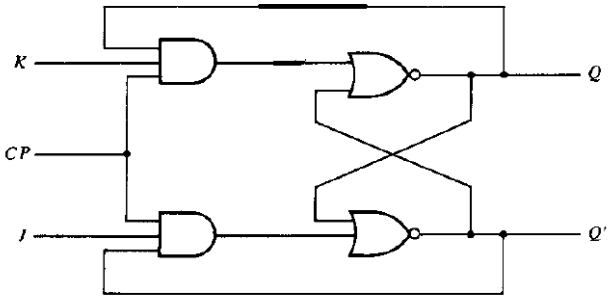
[23] C. Hamacher, Z. Vranesic, and S. Zaky, "*Compputer Organization*", Mc-Graw Hill, Fifth Edition.

[24] D. A. Patterson, and J.L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, Elseiver, Third Edition.

[25] D. A. Patterson, and J.L. Hennessy, *Computer Architecture: A Quantitive Approach*, Elseiver, Fourth Edition.

[26] I. Hossain, and B. K. Gunturk, "High Dynamic Range Imaging for Non-Static Scenes", SPIE Electronic Imaging Conference, 2011.

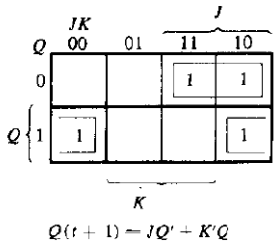
APPENDIX: OPERATION OF JK FLIP-FLOP



(a) Logic diagram

Q	J	K	Q(t + 1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

(b) Characteristic table



$Q(t+1) = JQ' + K'Q$

(c) Characteristic equation

Figure 5: JK Flip-Flop's Operating Table

The J-K flip-flop is versatile and is a widely used type of flip-flop. The J and K designations for the inputs have no known significance except that they are adjacent letters in the alphabet. The functioning of the J-K flip-flop is identical to that of the S-R flip-flop in the SET, RESET and no-change condition of operation. The difference is that the J-K flip-flop has no invalid state as does the S-R flip-flop.

Figure 4 shows the basic internal logic for a positive edge-triggered J-K flip-flop. Notice that it differs from the S-R edge-triggered flip-flop in that the Q output is connected back to the input of gate where K input is feed, and the Q' output is connected back to the input of gate where J input is feed. A J-K flip-flop can also be of negative edge-triggered in which case the clock input is inverted.

As you can see, on each successive clock spike, the flip-flop changes to the opposite state. This mode is called toggle operation.

Matlab generated code for the model this work simulated:

```
function importfile(fileToRead1)
%IMPORTFILE(FILETOREAD1)
%  Imports data from the specified file
%  FILETOREAD1:  file to read

%  Auto-generated by MATLAB on 14-Aug-2016 14:05:32

% Import the file
newData1 = load('-mat', fileToRead1);

% Create new variables in the base workspace from those fields.
vars = fieldnames(newData1);
for i = 1:length(vars)
    assignin('base', vars{i}, newData1.(vars{i}));
end


```