International Journal of Mechanical Engineering & Computer Sciences, Vol.1, Issue 1, Jan-Jun, 2017, pp 55-63


www.arseam.com

# FUZZY BASED LEAKAGE- AND TEMPERATURE AWARE INSTRUCTION-LEVEL SCHEDULING FOR VLIW

**M. Lordwin Cecil Prabhaker[1], M. Balasubramani[2]**
**K. Manivannan[3]**

[1]Department of Electronics and Communication Engineering,
Anna University- University College of Engineering Dindigul, Tamilnadu, India

[2]Department of Electrical and Computer Engineering, Dilla University, Dilla, Ethiopia.

[3]Department of Computer Science & Engineering,
PSNA College of Engineering and Technology, Dindigul, Tamilnadu, India

**ABSTRACT**

The substantial increase in the leakage component of the total processor energy consumption is due to the miniaturization of devices and the ensuing decrease in the threshold voltage. A large fraction this leakage energy consumption in the functional units is attributed by relatively simpler issue logic and the presence of a large number of function units in the VLIW and the clustered VLIW architectures. However, because of the inherent variations in the ILP of the programs functional units are not fully utilized in the VLIW architectures. Due to the contentions for the limited number of slow inter-cluster communication channels which lead to many short idle cycles, this under utilization is even more pronounced in the context of clustered VLIX architectures. In this paper, we develop a fuzzy based real-time instruction level loop scheduling technique to reduce leakage energy consumption and look up table based temperature-aware workload balance for applications with loops on VLIW architecture. We first prove that the scheduling problem with the minimum leakage energy consumption within a timing constraint is NP-complete. Then, Fuzzy based leakage energy consumption and look up table based temperature-aware workload balance )algorithm is designed to repeatedly regroup a loop based on rotation scheduling , and decrease leakage energy integrating with leakage power reduction mechanism. We conduct experiments on a set of DSP benchmarks based on the power model of the VLIW processors. The results show that our algorithm achieves significant leakage energy saving compared with list scheduling.

***Keywords:*** VLIW architecture, Fuzzy based leakage energy consumption, temperature-aware Scheduling.

## I. INTRODUCTION

At present, many electronic systems require integrated dedicated components that are specialized to perform a task or a limited set of tasks. These are called Application Specific Integrated Circuits (ASICs). The Very long instruction word or VLIW refers to a processor architecture designed to take advantage of Instruction level parallelism (ILP). Energy consumption has become a critical issue in the design of embedded systems. As feature size shrinks, leakage energy contributes an ever increasing fraction of the total energy consumption.

A leakage energy and peak temperature reduction by compiler-assisted instruction-level scheduling is focused for stream processors. A leakage- and temperature aware design flow is presented, which is composed of a performance-oriented schedule to minimize working time. A leakage-aware rescheduling (LARS) algorithm that

concentrate operations to fewer FUs so that other FUs can be shut-down for leakage energy reduction, and a temperature aware algorithm that balance workloads among FUs for peak temperature reduction. The LARS algorithm makes every effort to concentrate operations to higher-priority FUs for increasing power-gated cycles without performance loss.

The objective of our work is

- A leakage- and temperature-aware design flow is presented to assist the compiling of instruction-level VLIW scheduling. And two scheduling algorithms are proposed for the design flow.
- The leakage-aware rescheduling algorithm and temperature-aware workload balance algorithm can reduce the leakage energy and peak temperature.
- The main objective is to reduce the leakage energy without performance loss using Fuzzy based instruction level loop scheduling technique.
- To reduce the peak temperature without performance loss using Look up table based temperature-aware workload balance technique.

## II. EXISTING METHODS

The temperature profile, process variation, and the transistor count that all have strong impact on the leakage power distribution of a processor. The framework is based on architecture similar to the Intel Itanium IA64 and it is extended to simulate its power when implemented in 65nm technology [1]. In the proposed temperature aware design the number of devices per unit area is scaling up faster than the power density is scaling down [2], [23]. This requires a cooling solutions to keep the chip and its local hotspots cool. Furthermore, high temperature slows integrated circuits because of degraded carrier mobility. It also accelerates multiple chip-failure mechanisms such as electro migration and negative bias temperature instability (NBTI), because the wear out rate that has an exponential temperature dependence [7]. The static leakage power is primarily an exponential function of temperature. . There is also the possibility of thermally induced security vulnerabilities, such as denial of service. Temperature-aware design can reduce these problems. In [3] the very long instruction word processors (VLIW) allow executing multiple instructions in parallel manner. At the same time the VLIW compilers optimize the source code for maximum performance by grouping as many instructions in parallel. This work addresses the problem of thermal-awareness in VLIW compilers. The proposed technique called temperature-aware instruction binding technique (TemplB) that effectively binds the instructions executed in parallel to the coolest possible functional units for a given schedule. It generates for each instruction in a scheduled instruction word and a priority queue of the coolest functional units that can execute the instruction, and it rebinds it to the coolest possible unit, considering the temperature and the power consumed by the instruction. Temperature-aware offline and online techniques to reduce leakage energy by idle time distribution of each task [5]. Large-scale integration with deep sub-micron technologies has led to high power densities and high chip working temperature. At the same time, leakage energy has become the dominant energy consumption source of circuits due to reduced threshold voltages [12], [13]. Due to close interdependence between temperature and leakage current and temperature has become a major issue to be considered for power-aware system level design techniques. The issue of leakage energy optimization through temperature aware idle time distribution (ITD) is addressed. An analytical temperature analysis approach which is used inside the energy optimization loop. An offline ITD technique is used to optimize leakage energy consumption, where only the static idle time is distributed. To account for the dynamic slack, an online ITD technique is used where both static and dynamic idle time are considered.

## III. SYSTEM MODEL

Despite of the computation intensity of media applications, still a big fraction of FUs are idle or not frequently used due to the inherent operation dependency and resource constraints. Therefore, idle intervals of FUs can be combined intensively in both temporal and spatial dimensions so that more idle FUs are powered down and less switch

overheads are induced [17]. Although low power design can reduce chip temperature by reducing the average power density, localized peak temperature does not decrease necessarily. Some low power mechanisms even conflict with temperature management, because temperature is influenced not only by power, but also by area and power distribution. In these situations temperature-aware algorithm is needed to reduce peak power density.

**3.1 CLUSTER:**

The cluster consists of many functional units. They are,

- Adders
- Multipliers
- Divider

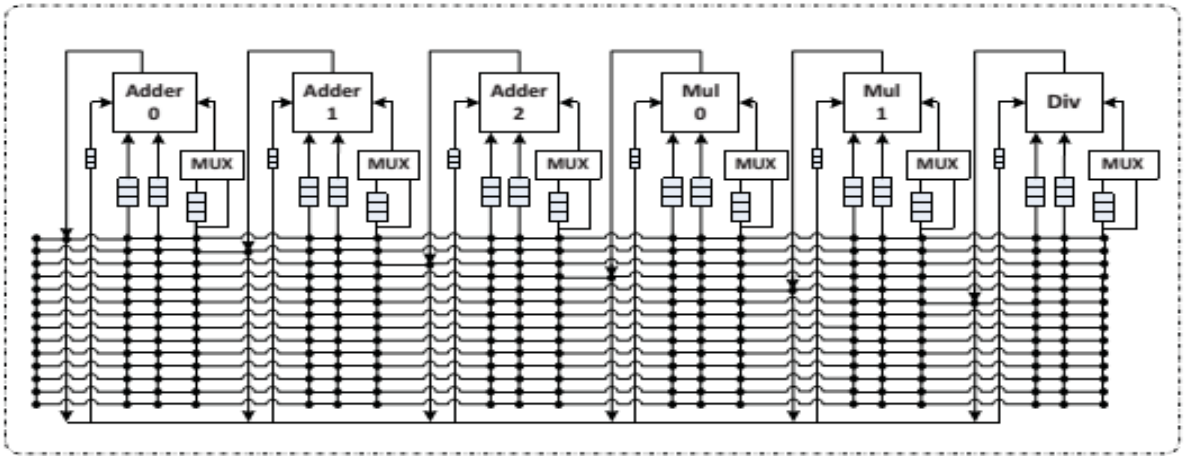



Figure 3.1: Architecture of a cluster

In a typical VLIW architecture of a stream processor, there are four clusters working in single instruction multiple data In a cluster, the various FUs which are controlled by statically scheduled VLIW instructions issued by the microcontroller. Distributed register files are utilized and the outputs of all FUs can be connected to other register within the cluster is shown in Fig.4.2. This eliminates operand replications when moving operations to other FUs.

**3.1.1 ADDER:**

The operands are A and B. A bit carried in from the previous less significant stage is Cin. A component in a cascade of adders, which add 8, 16, 32, etc. bit binary numbers, is the Full adder.

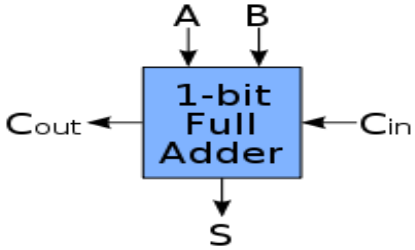



Figure 3.2 Full adder

The circuit produces a two-bit output, carry output and sum typically represented by the signals $C_{out}$ and S, where

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = (A \cdot B) + (C_{in} \cdot (A \oplus B))$$

**3.1.2 MULTIPLIER:**

A binary multiplier is to multiply two binary numbers. An AND operation compares two binary values. If both values are high, the result of the AND operation is high. If any one of the values is low, the result is low. If both of the values are low, the result is low.

**TABLE 4.2: TRUTH TABLE FOR MULTIPLIER**

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**3.5 DESIGN FLOW:**

The proposed design flow is composed of three steps. In the first step, a performance-oriented schedule is performed and the first scheduling result SCHD1 is generated. Operations are scheduled to the earliest possible cycle and the execution time is minimized.

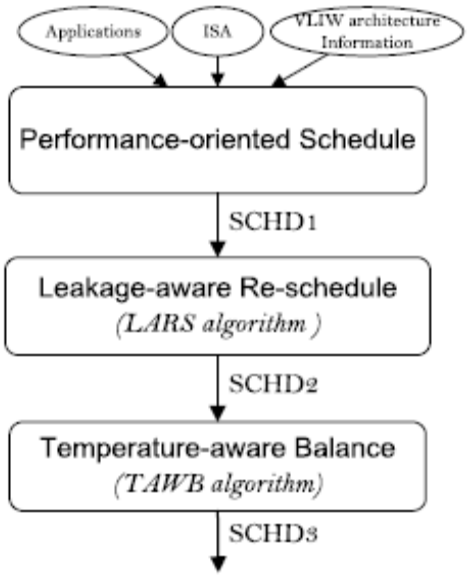



Figure 3.3: Leakage- and temperature-aware design flow

In the second step, SCHD1 is rescheduled to SCHD2 by a LARS algorithm. The operations are concentrated temporally and spatially, and then the leakage energy can be reduced by shutting down the FUs that are not used for a long period. However, the localized temperature rises due that the FUs, which are concentrated by operations, and it have heavy workloads

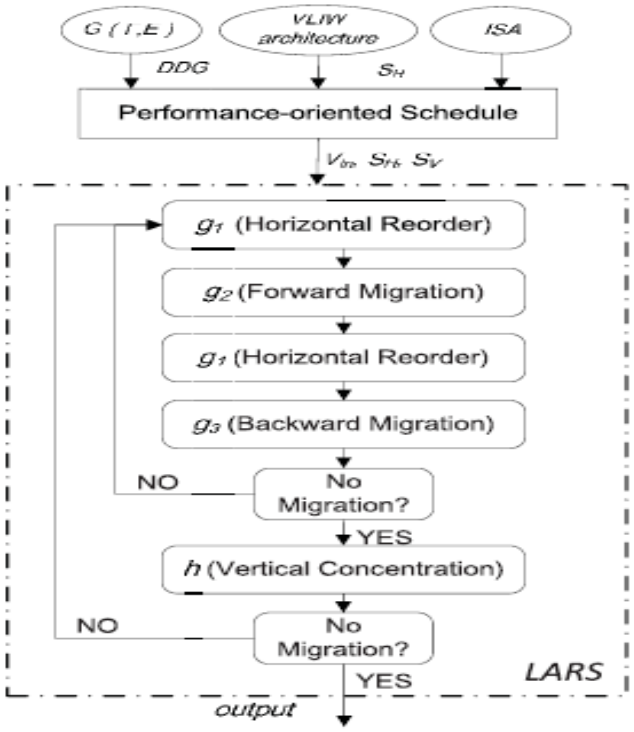



Figure 3.4 Framework of LARS algorithm

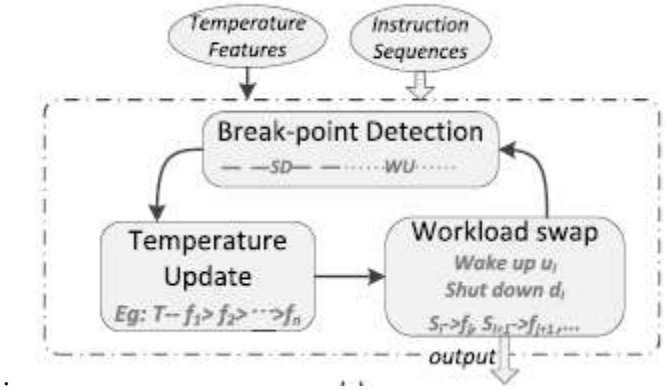



Figure 3.5 TAWB algorithm framework

To alleviate the temperature increase the third step, the temperature-aware balancing algorithm is used to balance the workloads among FUs and generates the final schedule SCHD3 Rather than attempting to model a system mathematically; FL incorporates a simple, rule-based IF X AND Y THEN Z approach to solving control problem. Relying on an operator's experience rather than their technical understanding of the system, the FL model is empirically-based.

For example, terms like "IF (process is too cool) AND (process is getting colder) THEN (add heat to the process)" or "IF (process is too hot) AND (process is heating rapidly) THEN (cool the process quickly)" are used rather than dealing with temperature control in terms such as "SP=500F"."T <1000F", or "210C <TEMP <220C". These terms

are very descriptive of what must actually happen and are imprecise.

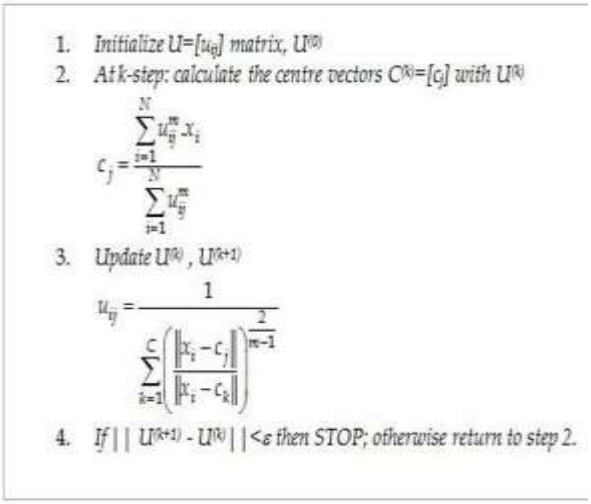



Figure.3.6 Fuzzy Based Loop Scheduling Algorithm framework

## 4. IMPLEMENTATION

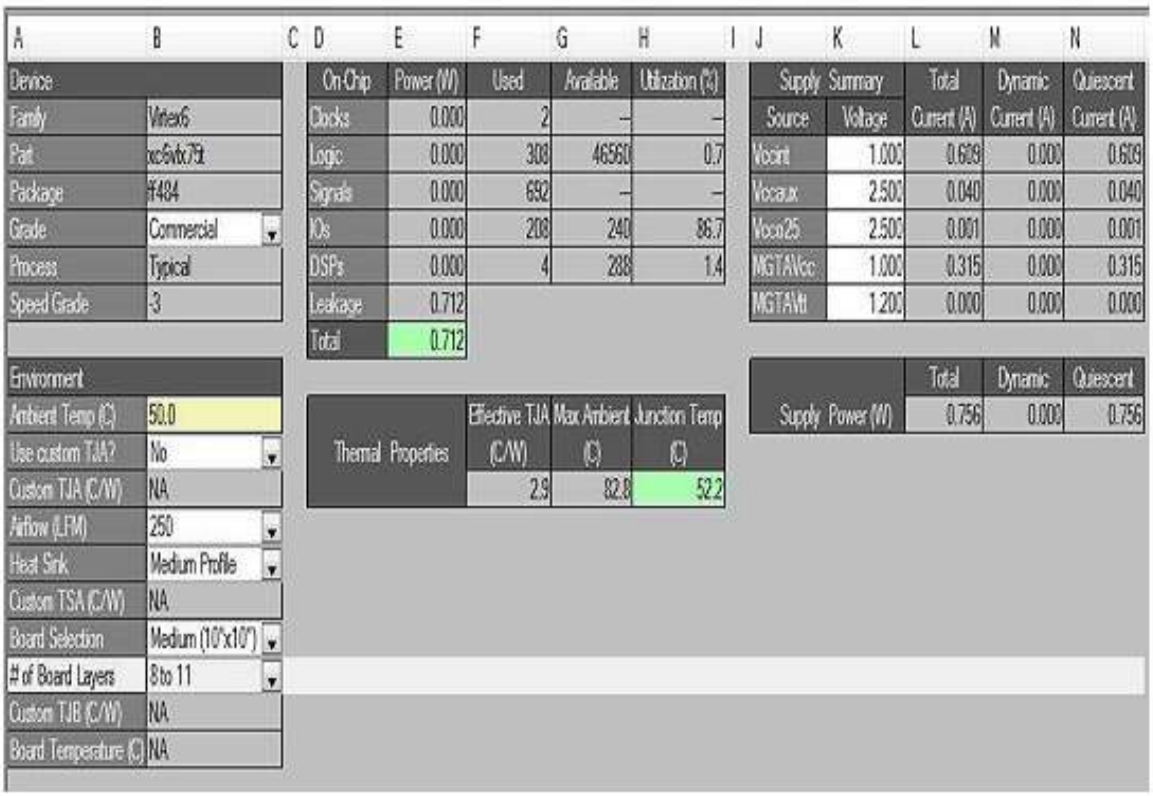
The cluster architecture is formed by using 3 – Adders, 1 –divider and 2 –Multiplexers. The Fuzzy Based LARS & TAWB Algorithm is implemented on cluster based VLIW Architecture. In a cluster, the various FUs which are controlled by statically scheduled VLIW instructions issued by the microcontroller. Distributed register files are utilized and can be the outputs of all FUs connected to other register within the cluster. This controller contains three sub modules. Modules are Instruction memory, Instruction Decoder and PG Controller. It gives the control signals through PG bits into the cluster. The temperature increase is modelled on a cycle-by-cycle basis for an FU of continuous execution. The input signal for Fuzzy logics contains the positive, negative and zero logic signals and clock signals are common. For power analysis the Xilinx12.1and Modelsim 5.5 is used to implement the cluster architecture.

## 5. RESULTS AND DISCUSSION

The power consumption for VLIW architecture is given by,

***Power Consumption = Total Power Leakage* X *(Sum of Utilization/100) %***

This Power Consumption takes place only on idle stage. The calculated total leakage power for LARS and TAWB is 0.712 W and the Sum of utilization is 92.6W. Hence the power consumption is 0.659W. For fuzzy based LARS and TAWB the leakage power consumption is same but the Sum of utilization is calculated as 88.8W. So the power consumption while using this scheduling algorithm is 0.632W. Hence by using Fuzzy logic the power degradation was reduced up to 0.027(W). The below Fig5.1 shows that fuzzy based LARS and TAWB Power analyser window.



| Device | |
|---|---|
| Family | Virtex6 |
| Part | xc6vlx75t |
| Package | ff484 |
| Grade | Commercial |
| Process | Typical |
| Speed Grade | -3 |

| Environment | |
|---|---|
| Ambient Temp (C) | 50.0 |
| Use custom TJA? | No |
| Custom TJA (C/W) | NA |
| Airflow (LFM) | 250 |
| Heat Sink | Medium Profile |
| Custom TSA (C/W) | NA |
| Board Selection | Medium (10"x10") |
| # of Board Layers | 8 to 11 |
| Custom TJB (C/W) | NA |
| Board Temperature (C) | NA |

| On-Chip | Power (W) | Used | Available | Utilization (%) |
|---|---|---|---|---|
| Clocks | 0.000 | 2 | – | – |
| Logic | 0.000 | 308 | 46560 | 0.7 |
| Signals | 0.000 | 692 | – | – |
| IOs | 0.000 | 208 | 240 | 86.7 |
| DSPs | 0.000 | 4 | 288 | 1.4 |
| Leakage | 0.712 | | | |
| Total | 0.712 | | | |

| Thermal Properties | Effective TJA (C/W) | Max Ambient (C) | Junction Temp (C) |
|---|---|---|---|
| | 2.9 | 82.8 | 52.2 |

| Supply Summary | | Total | Dynamic | Quiescent |
|---|---|---|---|---|
| Source | Voltage | Current (A) | Current (A) | Current (A) |
| Vccint | 1.000 | 0.609 | 0.000 | 0.609 |
| Vccaux | 2.500 | 0.040 | 0.000 | 0.040 |
| Vcco25 | 2.500 | 0.001 | 0.000 | 0.001 |
| MGTAVcc | 1.000 | 0.315 | 0.000 | 0.315 |
| MGTAVtt | 1.200 | 0.000 | 0.000 | 0.000 |

| Supply Power (W) | Total | Dynamic | Quiescent |
|---|---|---|---|
| | 0.756 | 0.000 | 0.756 |

Figure.5.1. Fuzzy based LARS&TAWB power consumption.

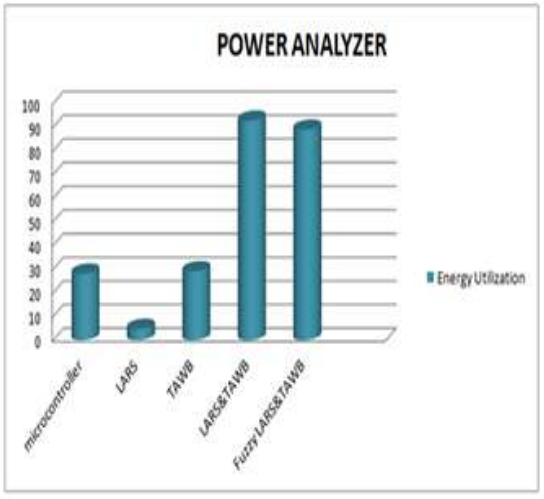



Fig 5.2 Power Consumption in W

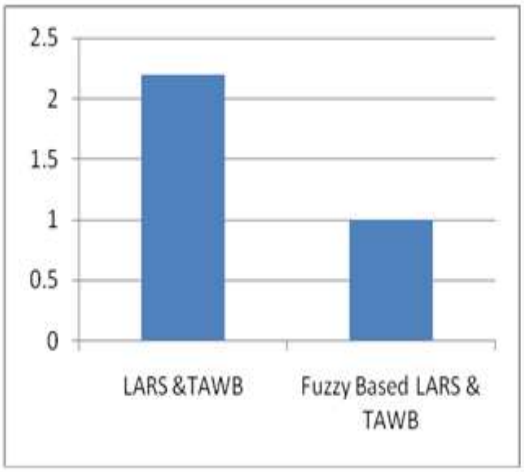



Figure 5.3. Speed Comparison between LARS and TAWB and Fuzzy based LARS and TAWB

The Inference about Power Analyzer between the modules and energy utilization is shown in Fig 5.2. The graph shows that the power consumption of fuzzy based LARS and TAWB approach is reduced about 4.2%. Fig 5.3 shows

that performance of fuzzy based LARS and TAWB is compared with existing scheduling approach, which shows that the speed is increased about 10% and the delay is reduced about 60% and the area is also reduced.

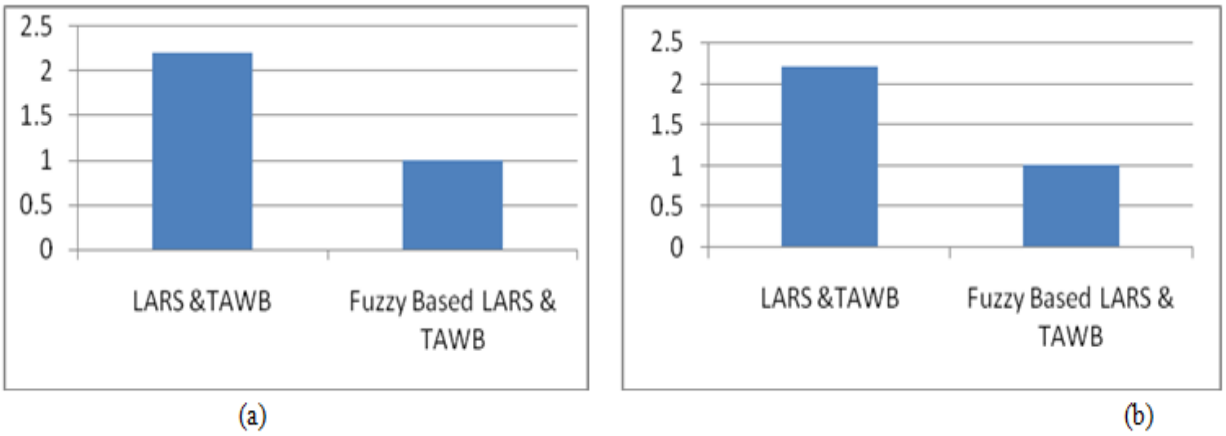



Figure 5.4. Performance Analysis (a)Delay in µs (b) Area in nm

## CONCLUSION

A leakage and temperature aware design flow is proposed for addressing the leakage energy optimization and temperature reduction issues simultaneously. Two algorithms are presented for the design flow by compiler-assisted instruction-level scheduling. First, the LARS algorithm is proposed, which concentrates operations spatially and temporally, so that more FUs can be power-gated for leakage energy optimization. It is performed after a performance oriented scheduling algorithm and brings no performance loss. The transient temperatures of the FUs are evaluated and swap the FU workloads at each swap cycle, so that workloads are balanced and the peak temperature is decreased. It brings no leakage energy increase while peak temperatures are decreased. The Fuzzy based instruction level loop scheduling technique and Look up table based temperature-aware workload balance technique that can reduce the leakage energy and peak temperature more efficiently. This can be used for future work.

## REFERENCES

[1] Ayala, J.L., Apavatjrut, A.,Atienza, D., Lopez-Vallejo, M.,(2007) "Exploring Temperature-Aware Design of Memory Architectures in VLIW Systems", Innovative architecture for future generation high-performance processors and systems,pp.81-87.

[2] Bao, M., Andrei, A., Eles, P., and Peng, Z.,(2012) "Temperature-aware idle time distribution for leakage energy optimization", IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 7, pp. 1187–1200.

[3] Chaparro,P., Gonzalez,J.,Gonzalez,A.,(2004)"Thermal-aware clustered microarchitectures", Computer Design VLSI in Computers and Processor,PP. 48 .

[4] Chengmo Yang,Orailoglu, A.,(2009) "Processor reliability enhancement through compiler- directed register file peak temperature reduction",Dependable Systems & Network,pp.468 – 477.

[5] Fang-Ju Lin , Herming Chiueh,(2006) "A Micro-architecture Simulator for Multimedia Stream Processor" Electronics, Circuits and Systems, pp.768 – 771.

[6] 6. Goto, M., Sato, T., (2004) "Leakage energy reduction in register renaming", Distributed Computing Systems Workshops, pp. 890 – 895.

[7] Huang, W., Allen-Ware, M., Carter J.B, Cheng, E., Skadron, K., and Stan M.R. (2011) "Temperature-aware architecture: Lessons and opportunities",IEEE Micro, vol. 31, no. 3, pp. 82–86.

[8] H.Su, F.Liu, A.Devgan,E.Acar,and S.Nassif,(Aug 2003)"Full chip leakage estimation considering power supply and temperature variations," in Proc.int.Symp.Low Power Electron pp.78-83.

[9] InchoonYeo, EunJungKim,(2008) "Hybrid dynamic thermal management based on statistical characteristics of multimedia applications" Low Power Electronics and Design,pp321 – 326.

[10] J.Srinivasan and S.V.Adve.,(2003)"Predictive dynamic thermal management for multimedia applications," in Proc.ACM/IEEE Int.conf.pp.109-120.

[11] Kumar,A., Li Shang , Li-Shiuan Peh , Jha N.K.,(2008) "System-Level Dynamic Thermal Management for High-Performance Microprocessors",Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions. Volume:27,no.1, pp.96 – 108.

[12] L.Benini, A.Bogliolo, and G.De Micheli,( June 2000)"A survey of design techniques for system –level dynamic power management," IEEE Trans.pp.299-316.

[13] M.Mutyam, F.Li, V.Narayanan, N.Kandemir,(June 2006) "Compiler-directed thermal management for VLIW functional units" pp.163-172.

[14] Nagpal, R., Srikant Y.N.,(2007) "Register File Energy Optimization for Snooping Based Clustered VLIW Architectures", Computer Architecture and High Performance Computing, pp.161 – 168.

[15] R. W. Lewis, P. Nithiarasu, and K. N. Seetharamu.,(2005) Fundamentals of the Finite Element Method for Heat and Fluid Flow. New York, NY.

[16] Schafer, B., Lee, Y., and Kim, T.,(2007) "Temperature-aware compilation for VLIW processors", in Proc. Int. Conf. Embedded Real Time Comput.Syst. Appl., pp. 426–431.

[17] Shan Cao,Zhaolin Li,Fang Wang and Shaojun Wei,(2014) "Compiler-Assisted Leakage and Temperature-Aware Instruction-Level VLIW Scheduling,"VOL.22.No.6.51

[18] Sheng Wei, Zheng, J.X., Potkonjak, M., (2013)"Aging-based leakage energy reduction in FPGAs", Field Programmable Logic and Applications (FPL), pp.1-4.

[19] Tiantian Liu,Orailoglu, A., Xue, C.J., Minming Li (2011) "Register allocation for simultaneous reduction of energy and peak temperature on registers" Design, pp.16.

[20] Tsai, Y.F., Ankadi, A.H., Vijaykrishnan, N., Irwin, M.J and Theocharides, T.(2004) "ChipPower: An architecture-level leakage simulator",in Proc. IEEE Int. SOC Conf, pp. 395–398.

[21] Wen-Wen Hsieh,Ting-Ting Hwang,(2009) "Thermal-aware post compilation for VLIW architectures", Design Automation Conference,pp. 606 – 611.

[22] Xiangrong Zhou , Chenjie Yu ,Petrov, P.,(2008) "Compiler-driven register re-assignment for register file power-density and temperature reduction"Design Automation,pp 750 – 753.

[23] Y.Han, I.Koren, and C.A.,Moritz.(2005)"Temperature aware floorplaning" in Proc.2nd Workshop Temperature Aware Comput.

[24] Zhang, W.,Vijaykrishnan, N., Kandemir, M. , Irwin, M.J. (2001) "Exploiting VLIW schedule slacks for dynamic and leakage energy reduction" ,Microarchitecture, pp.102 – 113.

[25] Zhang, W., Kandemir, M., Vijaykrishnan, N., Irwin, M.J.,(2003) "Compiler support for reducing leakage energy consumption",Design, Automation,pp. 1146 – 1147.