Asian Journal of Information Technology 15 (11): 1785-1790, 2016
ISSN: 1682-3915

Reducing Power and Delay in Instruction Queue for Sram Based Processor Unit

[1]G. Dhanalakshmi, [2]M. Sundarambal and [2]K. Muralidharan
[1]Department of ECE, Sri Ramakrishna Engineering College,
[2]Department of EEE, Coimbatore Institute of Technology, Coimbatore, India

Abstract: The ever-growing requirements of advanced computing platform are the extension of battery lifetime for high performance microprocessors. Power minimization has become a primary concern in microprocessor design. One of the main dynamic instruction scheduling used in data path designs is Instruction Queue (IQ) which allows the out of order execution in a superscalar processor. The Instruction Queue holds decoded and renamed instructions until they issue out-of-order to appropriate functional units. It consumes the considerable amount of power in a high performance processor and this unit is responsible for 27% of total chip power dissipation in typical superscalar microprocessor. The proposed technique aims at reducing the dynamic and static power dissipation in the Instruction Queue (IQ) by using power-gated match-line sense amplifier. It also reduces sensing delay during search operation in the IQ compared to conventional IQ. The proposed design of IQ using power gated match line sense amplifier is less hardware modifiable with the least amount of redesign and verification efforts, lowest possible design risk, least hardware overhead and without significant impact on the performance.

Key words: Instruction queue, leakage power, dynamic power, CAM, SRAM, issue width, sensing delay

INTRODUCTION

High performance embedded applications in the multimedia, networking, imaging and high-end consumer application domains are increased in the market today. Since, these applications need high performance requirements, there is a gradual shift towards using more complex out-of-order superscalar embedded to meet the performance goals. The demand for low power consumption in battery powered applications is rising sharply. This is to extend the service lifetime of systems by reducing power requirement. Now a days, low power design has become a critical design consideration in high end systems due to its expensive cooling and packaging costs and lower reliability often associated with high levels of on-chip power dissipation.

Power dissipation is becoming an important factor in high performance processor design as technology scales and device integration level increases. The power dissipation becomes a major concernin deep sub-micrometer technology (65 nm and below) due to leakage and dynamic power. Several power reduction techniques have been proposed in the literaturefor individual processor units. Many attemptshave been made to design either new power-efficient units or make current architectures more power-aware.

The modern high performance processors execute instructions aggressively, processing them in each pipeline stage as soon as possible. This requires fetching and processing large number of instructions. A typical processor fetches several instructions from the memory, decodes them and dissipates them to the instruction queue. The Instruction Queue (IQ) of a super scalar processor is a complex structure which is dedicated to out-of-orderexecution. The instruction queue is liable for a significant amount of overall processor power dissipation due to its high complexity. This power varies between 25-27% of processor's totalpower dissipation (dynamic and static power) referring to the literature review (Folegnani and Gonzalez, 2001; Kucuk *et al*., 2002).

The instruction queue is a CAM+RAM-like structure which hold instructions until they can be issued out-of-order to appropriate functional units. In this structure, the source operand numbers are placed in the CAM structure while the remaining information about the instructions is placed in the RAM structure. The number of entries corresponds to the size of the instruction queue. The CAM+RAM structure is complex interms of design and verification time and does not support compaction. This structure has lower power dissipation and hence, it has the potential to reduce the average power dissipation of the Instruction Queue.

CAM and RAM structures require precharging and discharging internal high capacitance lines and nodes for every operation. The CAM needs to perform tag matching operations at every cycle. This involves driving and

Corresponding Author: G. Dhanalakshmi, Department of ECE, Sri Ramakrishna Engineering College, Coimbatore, India

clearing high capacitance taglines and also precharging and discharging high capacitance matchline nodes at every cycle. Similarly, the RAM also needs to charge and discharge its bitlinesfor read operation. The following four possible actions are associated with IQ (Homayoun and Baniasadi, 2005):

- Set an entry for every dispatched instruction
- Read an entry to issue a new instruction to a functional unit
- Wakeup the next instruction waiting in the IQ once the result getsready
- Select instructions for issue when thenumber of instructions available exceeds the processor issue limit (issue width)

The main complexity of the Instruction Queue system is the associative search during the wakeup process. All the above tasks are energy demanding and make the Instruction Queue one of the major energy consumers in the processor as shown in (Homayoun and Baniasadi, 2005; Canal and Gonzalez, 2001; Hu *et al*., 2004a; Abella *et al*., 2003; Palacharla *et al*., 1997; Ernst *et al*., 2003; Buyuktosunoglu *et al*., 2002; Aggarwal *et al*., 2004; Brown *et al*., 2001; Ernst *et al*., 2003).

Literature review There are several approaches introduced to reduce the power dissipation during the associative search which is related to wakeup logic. Folegnani and Gonzalez (2001) proposed a new technique whichavoids waking up empty entries in the Instruction Queue. Brown *et al*. (2001) proposed amethod to remove the select logic from the critical path (Brekelbaum *et al*., 2002). Homayoun and Baniasadi (2005) predicted "lazy instructions" which are spending longperiods in the Instruction Queue.

They reduced Instruction queue power dissipation by waking up lazy instruction in every two cycles. They could reduce wakeup power dissipation in the IQ by reducing the fetch rate when the number oflazy instructions in the pipeline exceeds a dynamically decided threshold (Homayoun and Baniasadi, 2005). Canal and Gonzalez (2001) proposed a scheme which schedules instructions based on their expected issue time in the IQ (Kucuk *et al*., 2002).

Raasch *et al*. (2002) suggested adapting the issue queue size and partitioned issue queues to reduce the wakeup activity. Brekelbaum *et al*. (2002) proposed a new scheduler which exploits latency instructions in orderto reduce implementation complexity. Stark *et al*. (2000) used grandparent availability time to speculate wakeup logic (Hu *et al*., 2004b). Ernst *et al*. (2003) introduced a wakeup free scheduler which relied onpredicting the instruction issue latency. Hu *et al*. (2004a, b) used wakeup-free schedulers forexplaining the design constrains result in performance loss and suggested a model to eliminate some of those constrains (Stark *et al*., 2000). The proposed work is an attempt to reduce leakage and dynamic power in the instruction queue.

MATERIALS AND METHODS

Conventional instruction Queue: The structure of the wakeup logic of an Instruction Queue is shown in Fig. 1. Tag drive lines are used to broadcast the results to all instructions waiting in the IQ. Each instruction in the IQ compares its operand tags with the broadcast tags. If a match is detected the instruction source operand gets ready. Once all source operands of an entry are becoming as ready (rdyL and rdyR flags) the instruction can enter the execution stage. Finally, the OR logic which is used for combining the results of comparators that sets the rdyL/rdyR flags.

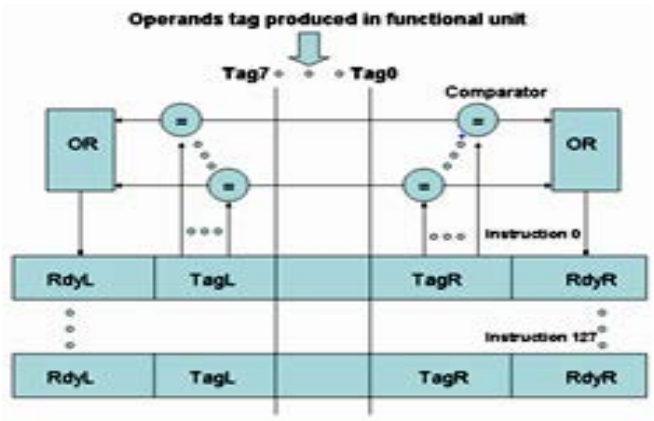

Fig. 1: Instruction queue in superscalar processor

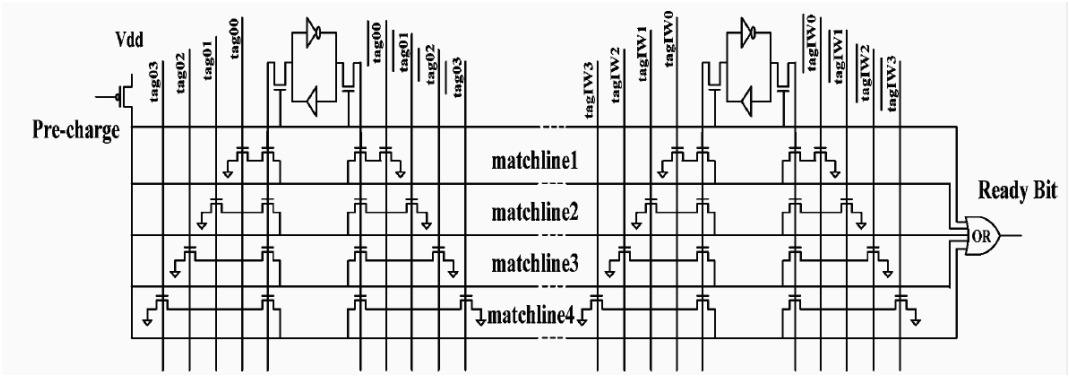

Fig. 2: Circuit implementation of instruction queue

The circuit level implementation for singlerow of the instruction queue is shown in a Fig. 2. In each cycle, all the match lines are precharged high which allows the individual bits associated with an instruction tag to be compared with the results broadcasted on the tag lines. When a mismatch occurs, the corresponding matchlines are discharged. If there is a match, the match line stays at V_{DD} which indicatesa tag match. At each cycle, there are up to four instructions broadcasted on the taglines, need four sets of one-bit comparators for each one-bit cell as shownin Fig. 2.

All four matchlines must be ored together to detect a match on any of the broadcasted tags. The result of the ORsets the ready bit of an instruction queue source operand showing that it is ready. The matchline discharge is the major energy consumption activity responsible for more than 58% of the energy consumption in the Instruction Queue (Canal and Gonzalez, 2001). The matchline must go across the entire width of the IQ, it has a large wire capacitance and adding the one-bit comparators diffusion capacitance makes the equivalent capacitance of matchline very large. Precharging and discharging this large capacitor is responsible for the majority of power consumption in the IQ.

Most of the time, all the instruction queue matchlines are discharged except one. In this configuration,among 8×4 matchlines in the instruction queue on an average in each cycle, only one is not discharged. Discharging the other matchlines will cause significant amount of power dissipation in the Instruction Queue. In other words, out of four taglines, on an average only one carries the broadcasted data from the functional units to all entries in the instruction queue. This means that precharging the other matchlines is not useful.

Proposed instruction queue: A new circuit level effective gated-power technique is used to reduceboth the static and dynamic power dissipation in the IQ and enhance the robustness of the design against process variations is proposed. Thus, robust, high-speed andlow-power sense amplifiers are highly sought-after in CAM designs. A feedback loop is employed to auto-turn off the power supply to the comparison elements and hence reduce the average power consumption and increase the speed by reducing the sensing delay.

The new proposed structure of wakeup logic of an Instruction Queue is shown in Fig. 3. Each cell has the same number of transistors as the conventional Instruction Queue. In the conventional circuit cell both sram and comparision units are powered by a same V_{DD}. The proposed circuit sram and comparision units are powered by two separate rails namely V_{DD} and V_{DDML}. V_{DDML} is controlled by a Power transistor (Px) independently and there is afeedback loop that can auto turn-off the ML current to save the power. The purpose of using two separate power rails (V_{DD} and V_{DDML}) is to completely isolate the SRAM cell from any possibility of power disturbances during compare operation.

To reduce the power dissipation there is a match line sense amplifier in the proposed circuit of an IQ. Once, the voltage on the ML reaches a certain threshold, the gated-power transistor Px is controlled bya feedback loop, this automatically turn off Px to reduce the power consumption.

Operation of the Match Line Sense Amplifier (MLSA) The circuit diagram for match line sense amplifier is shown in Fig. 4. In the MLSA, EN is set to low and the power transistor (Px) is turned off. It will make the signal C1 and ML initializedto V_{DD} and ground respectively. After that,

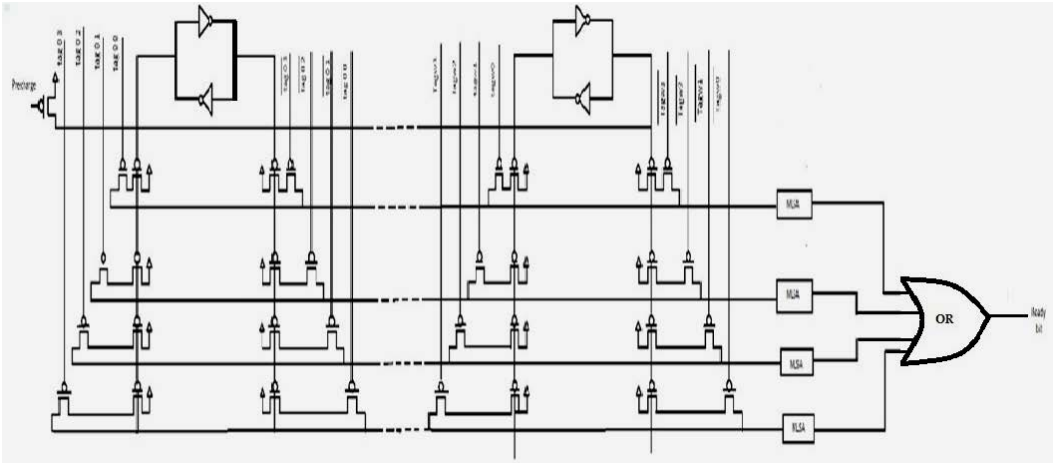

Fig. 3: Circuit diagram for the proposed IQ

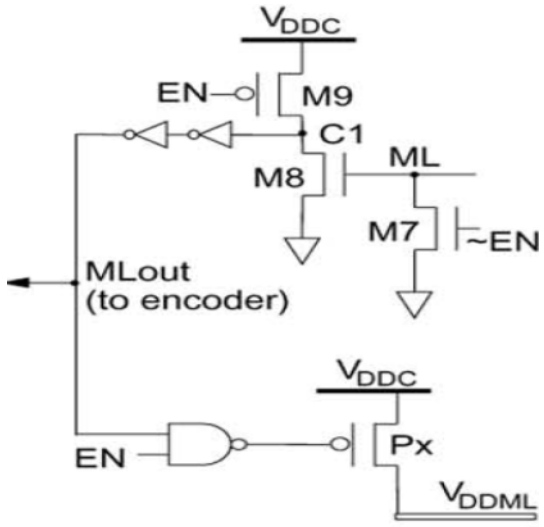

Fig. 4: Circuit diagram for the match line sense amplifier

signal EN turns high and initiates the compare phase. If one or more mismatchhappens in the cell, the ML will be charged up. All the cells of a row will share the limited current offered by the transistor (Px) during the mismatch. When the voltage of the match line reaches the threshold voltage of transistor M8.

Voltage at node C1 ispulled down. After asmall delay, the NAND2 gate is toggled and thus the power transistor Pxis turned off again. ML is not fully charged to V_{DD} but some voltage slightly above the threshold voltage of M8.

The voltage on the ML is charged to around 0.5 V only which is far below V_{DD} and hence, the power consumption is reduced. Thus, the new proposed architecture offers both low-power and high-speed operation.

Table 1: Performance comparison of conventional IQ with proposed IQ

	Average power consumption		Delay	
Supply voltage (VDD)	Conventional IQ (uW)	Proposed IQ (uW)	Conventional IQ (ns)	Proposed IQ (ns)
0.6V	3.2308	0.4764	128.17	34.72
0.7V	7.0571	0.7802	117.76	32.14
0.8V	13.7772	1.7777	113.78	30.81
0.7V	24.7134	3.1051	108.01	27.57
1.0V	41.5347	4.4683	103.67	20.87

RESULTS AND DISCUSSION

Simulation environment: The circuits for the conventional IQ and proposed IQ have been simulated using BSIM 3V3 65 nm technology on Tanner EDA tool. The testing environment is created and tested with the same input patterns on room temperature with supply voltage ranging from 0.6-1.0V.

Simulation analysis: The simulation circuit diagram for conventional IQ and proposed IQ are shown in Fig. 5 and 6, respectively. Precharge is always maintained high during the circuit testing.

Whenever the searched tag-data matched with operand tag-data, ready bit becomes high or else ready bit becomes low. The testing result of conventional IQ and proposed IQ functional wave forms are obtained during waveform simulations which are shown in Fig. 5and 6, respectively.

The comparison on the performance of the proposed IQ with the conventional IQ is shown in Table 1. The Power reduction is observed for the proposed design at 65 nm fabrication technology compared with conventional IQ.

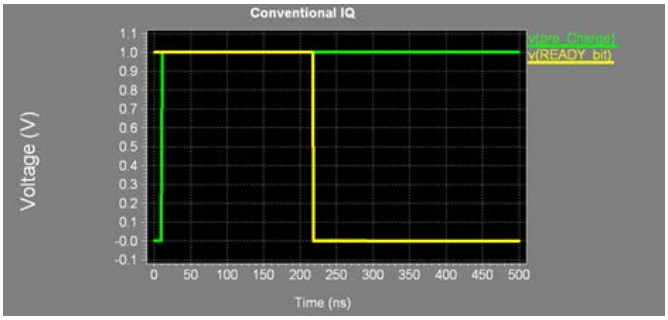

Fig. 5: Waveform of the conventional IQ

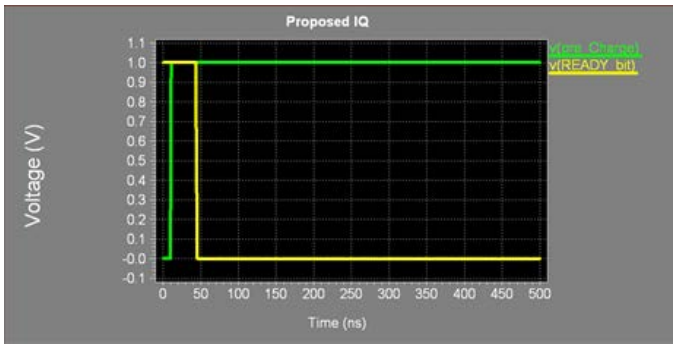

Fig. 6: Waveform for the proposed IQ

CONCLUSION

The design of instruction queue is a critical part of a superscalar out of order execution processor that dynamically schedules instructions for execution. The main complexity of the instruction queue stems from the associative search during wakeup process. Due to its high complexity, the instruction queue is responsible for a significant amount of overall processor power dissipation in the high performance processor. Instruction queue matchlines are discharged except one.

This means that precharging other matchlines is not useful. The new design of Instruction Queue prevent precharging such matchlines using power gated match line sense amplifier. It is designed using Tanner EDA tool at 65 nm technology.

In the high performance out of order superscalar processor, the test results of new design of IQ using MLSA outperforms in decrease in both static and dynamic power consumption, reducing the sensing delay to increase the performance and stability improvement of the memory cells and when compared to conventional IQ. In addition to this, the power saving technique used is technology independent one.

REFERENCES

Abella, J., R. Canal and A. Gonzalez, 2003. Power-and complexity-aware issue queue designs. IEEE. Micro, 23: 50-58.

Aggarwal, A., M. Franklin and O. Ergin, 2004. Defining wakeup width for efficient dynamic scheduling. Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors ICCD, October 11-13, 2004, IEEE, New York, USA., ISBN: 0-7695-2231-9, pp: 36-41.

Brekelbaum, E., J. Rupley, C. Wilkerson and B. Black, 2002. Hierarchical scheduling windows. Proceedings of the 35th Annual ACM/IEEE International Symposium on Microarchitecture, November 18-22, 2002, IEEE Computer Society Press, Istanbul, Turkey, ISBN: 0-7695-1859-1, pp: 27-36.

Brown, M.D., J. Stark and Y.N. Patt, 2001. Select-free instruction scheduling logic. Proceedings of the 34th ACM/IEEE International Symposium on Microarchitecture MICRO-34, December 1-5, 2001, IEEE, Austin, Texas, USA., ISBN: 0-7965-1369-7, pp: 204-213.

Buyuktosunoglu, A., D.H. Albonesi, P. Bose, P.W. Cook and S.E. Schuster, 2002. Tradeoffs in power-efficient issue queue design. Proceedings of the 2002 International Symposium on Low Power Electronics and Design, August 12-14, 2002, ACM, Monterey, California, ISBN: 1-58113-475-4, pp: 184-189.

Canal, R. and A. Gonzalez, 2001. Reducing the complexity of the issue logic. Proceedings of the 15th international conference on Supercomputing, June 18-23, 2001, Sorrento, Italy, ISBN: 1-58113-410-X, pp: 312-320.

Ernst, D., A. Hamel and T. Austin, 2003. Cyclone: A broadcast-free dynamic instruction scheduler with selective replay. Proceedings of the 30th Annual International Symposium on Computer Architecture, June 9-11, 2003, IEEE, Ann Arbor, Michigan, ISBN: 0-7695-1945-8, pp: 253-262.

Folegnani, D. and A. Gonzalez, 2001. Energy-effective issue logic. Proceedings of the 28th Annual International Symposium Computer Architecture, June 30-July 4, 2001, ACM, Gothenburg, Sweden, ISBN: 0-7695-1162-7, pp: 230-239.

Homayoun, H. and A. Baniasadi, 2005. Using lazy instruction prediction to reduce processor wakeup power dissipation. Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), March 20-22, 2005, IEEE, Austin, Texas, ISBN: 0-7803-8965-4, pp: 0-1.

Hu, J.S., N. Vijaykrishnan and M.J. Irwin, 2004a. Exploring wakeup-free instruction scheduling. Proceedings of the International Symposium on High Performance Computer Architectur (HPCA-10), February 14-18, 2004, IEEE, Madrid, Spain, ISBN: 0-7695-2053-7, pp: 232-232.

Hu, Z., A. Buyuktosunoglu, V. Srinivasan, V. Zyuban and H. Jacobson et al., 2004b. Microarchitectural techniques for power gating of execution units. Proceedings of the 2004 International Symposium on Low Power Electronics and Design, August 9-11, 2004, ACM, Newport Beach, California, USA., ISBN: 1-58113-929-2, pp: 32-37.

Kucuk, G., D. Ponomarev and K. Ghose, 2002.Low-complexity reorder buffer architecture. Proceedings of the 16th international conference on Supercomputing, June 22-26, 2002, ACM, New York, USA., ISBN: 1-58113-483-5, pp: 57-66.

Palacharla, S., N.P. Jouppi and J.E. Smith, 1997. Complexity-effective superscalar processors. Proceedings of the 24th annual international symposium on Computer Architecture, June 1-4, 1997, ACM, Denver, Colorado, USA, ISBN: 0-89791-901-7, pp: 206-218.

Raasch, S.E., N.L. Binkert and S.K. Reinhardt, 2002. A scalable instruction queue design using dependence chains. Proceedings of the 29th Annual International Symposium on Computer Architecture, May 25-29, 2002, IEEE Computer Society, New York, USA., ISBN: 0-7695-1605-X, pp: 318-329.

Stark, J., M.D. Brown and Y.N. Patt, 2000. On pipelining dynamic instruction scheduling logic. Proceedings of the 33rd Annual ACM/IEEE International Symposium on Microarchitecture, December 10-13, 2000, ACM, Monterey, California, USA., ISBN: 1-58113-196-8, pp: 57-66.