Design and Implementation of Clock Gated ALU performing 16 functions using DeMux & AND Gate

M.Sathish Kumar[1],
U.G. Student
Raja College of Engineering and Technology,
Tamil Nadu

S.K.Senthil Kumar[2],
U.G. Student
Raja College of Engineering and Technology,
Tamil Nadu

R.Venkatesh[3]
Assistant Professor
Raja College of Engineering and Technology
Tamil Nadu

Abstract—— This paper presents a new method for designing an Arithmetic and Logic Unit (ALU) based on Clock Gating Technique. In this method, we propose a technique that scaling of technology and the need for higher performance and more functionality, power dissipation is becoming a major bottleneck for microprocessor designs. Because clock power can be significant in high-performance processors, we propose a clock gating (CG) technique using AND and DEMUX gates to effectively reduce clock power. It provide clock input to only one functional module that is either arithmetic or logical block, while the other is put OFF. This paper has been developed as part oflow power processor design in the platform Xilinx ISE 9.2 and synthesized on 90nm Spartan-3E FPGA. On simulator, we achieved 90.32% clock power reduction using latch based clock gating and 75.82% clock power reduction using latch free clock gating.

Index Terms—Clock gate, ALU, FPGA, clock power, dynamic power, LUT, leakage power

INTRODUCTION

Present-day, general-purpose microprocessor designs are faced with the daunting task of reducing power dissipation since power dissipation is quickly becoming a bottleneck for future technologies. Lowering power consumption is important for not only lengthening battery life in portable systems, but also improving reliability, and reducing heat-removal cost in high-performance systems.

Clock power is a major component of microprocessor power mainly because the clock is fed to most of the circuit blocks in the processor, and the clock switches every cycle.

Clock gating is a well-known technique to reduce clock power. Because individual circuit usage varies within and across applications, not all the circuits are used all the time, giving rise to power reduction opportunity. By ANDing the clock with a gate-control signal, clock gating essentially disables the clock to a circuit whenever the circuit is not used, avoiding power dissipation due to unnecessary charging and discharging of the unused circuits.

Effective clock gating, however, requires a methodology that determines which circuits are gated, when, and for how long. Clock-gating schemes that either result in frequent toggling of the clock-gated circuit between enabled and disabled states, or apply clock gating to such small blocks that the clock-gating control circuitry is almost as large as the blocks themselves, incur large overhead. This overhead may result in power dissipation to be higher than that without clock gating.

I. Related Work

Low Power ALU Design is based on application of clock gate to turn off the sub-module of ALU that is not in use by current executing instruction as decided by instruction decoder unit. According to the analysis, Clock Power consumes 55-75 percent of total chip power and will increase in the next coming generation of hardware designs at 32nm and below. Hence, reducing clock power is very important. Clock gating is a key power reduction technique used by hardware designers and is typically implemented by RTL-level HDL Simulator or gate level power analyzer tools.

$$Power = C_L \cdot (Voltage) \cdot (frequency)$$

In the equation, power is directly proportional to the square of voltage and the frequency of the clock.

A. Statement of the Problem:

Clock gating is used in VLSI circuit design to reduce dynamic power by gating off the functional unit that is not in use by current executing instructions as decided by instruction decoder unit.

II. Clock Gating

Clock Enable consumed More Power and Clock Gating consumed Less Power. According to reference [2], Power optimization, traditionally relegated to the synthesis and circuits level, now shifted to the System Level and Register-Transfer-Level. This is possible due to clock gating which switch off the inactive units of the design and reduce overall power consumption. There are many clock gating styles available to optimize power in VLSI circuits. They can be:

- Latch-free based CG design.
- Latch-based CG design.
- Flip-flop based CG design.
- Intelligent CG design.

A. *Latch-free Clock Gated ALU design*

We use an AND gate in clock gate if clock is active on the rising edge. We use an OR gate in clock gate if clock is active on the falling edge. Using idea given in [1] and [2], we develop following ALU design as shown in Fig. 1.

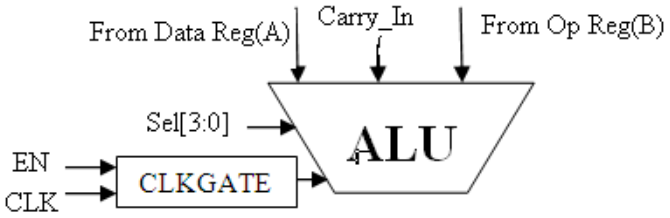

Fig. 1. Latch free clock gated design

B. *Problem in Latch-Free Clock Gated Design*

If enable signal goes inactive in between the clock pulse then gated clock terminated before his life time as shown in Fig. 2.

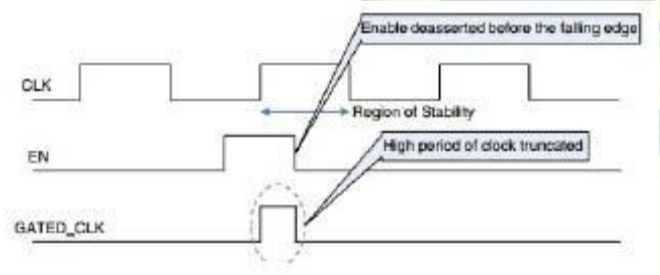

Fig. 2. Problem in latch-free clock gated design

C. *Latch Based Clock Gated ALU Design*

The latch-based clock gate consists of a level sensitive latch in design to hold the enable signal from the active edge to the inactive edge of the clock as shown in Fig. 3.

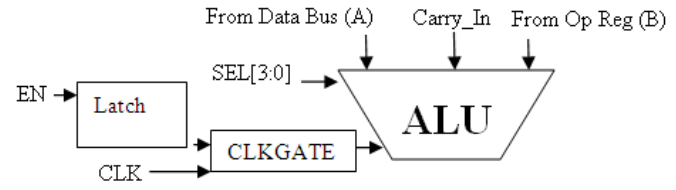

Fig. 3. Latch based clock gated ALU design

D. *Flip-Flop Based Clock gated ALU Design*

The Flip-Flop based clock gate consists of a level sensitive latch in design to hold the enable signal from the active edge to the inactive edge of the clock as shown in Fig. 4.

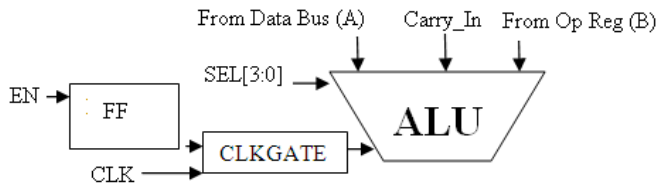

Fig. 4. Flip-flop based clock gated ALU design

III. Implementation Of ALU Performing 16 Operations

The clock gating technique is applied to a ALU performing 16 operations. The ALU designed performs arithmetic as well as logical operations. In this proposed model of ALU, the arithmetic and logical blocks are design separately. Each is provided with the two inputs and the select signals, to select a particular operation. The operations performed by the ALU are tabulated in table 1.

Table 1: ALU Operations Implemented

Select	Arithmetic Operation	Select	Logical Operation
0000	Input A	1000	AND Operation
0001	Addition	1001	OR Operation
0010	Subtraction	1010	XOR Operation
0011	Multiplication	1011	Complementary of A
0100	Division	1100	Complementary of B
0101	Modulo function	1101	NAND Operation
0110	Arithmetic Shift Right	1110	NOR Operation
0111	Arithmetic Shift Left	1111	XNOR Operation

The ALU generates 4 flags-Zero (Z), Carry (C), Sign (S), and Parity (P). Flags are not affected by the Unary Logic functions. Only the C flag is affected by the Shift function. All flags are affected by the other ALU functions.

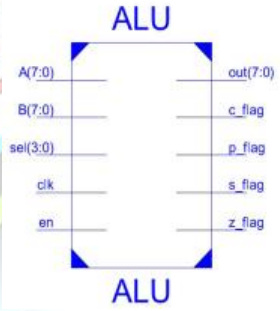

Fig. 5. Arithmetic and logic unit

Flags	Zero	1001	OR Operation
	Sign	1010	XOR Operation
	Carry	1111	XNOR Operation
	Parity		Sign Flag

The clock input is provided to the two functional modules through the gated clock. In this paper, the clock gate is implemented using AND and DEMUX logic.

A. *ALU with DEMUX Clock Gating*

The design of an ALU performing 16 operations are considered in this paper, consists of two functional blocks namely: arithmetic and logical. Each block is synchronous which means the inputs are evaluated on the occurrence of the clock. The DEMUX clock gating technique can be applied to the scenario, where only one functional unit operates while the other does not. Hence, the enable input of the DEMUX will provide the

clock signal either the arithmetic or the logical unit. The top view of the schematic is as shown in Fig. 6.

The design consists of the DEMUX, which passes the clock to either the arithmetic or the logical based on the enable input. The signal OUT0 is connected as the clock input to the arithmetic block while OUT1 to logical unit. The design is complied, simulated, and synthesized on the FPGA Spartan 3E. After the physical synthesis, it is observed that the design is implemented using standard cells.

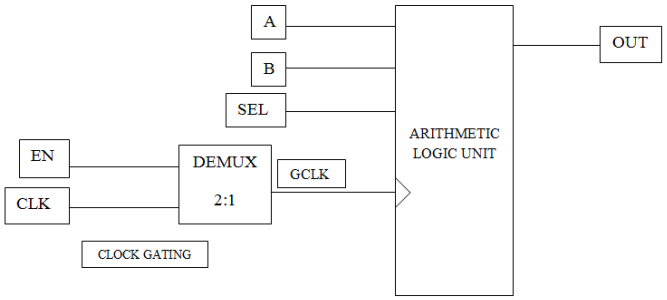

Fig. 6. Top View of the DEMUX Gated Clock

B. ALU with AND Clock Gating

In this second method of clock gating an AND gate is used to select the functional units. The clock input is provided to both the modules but either of the one is selected by the AND gate to perform the evaluation of the inputs. The top view of the design is shown in Fig. 7.

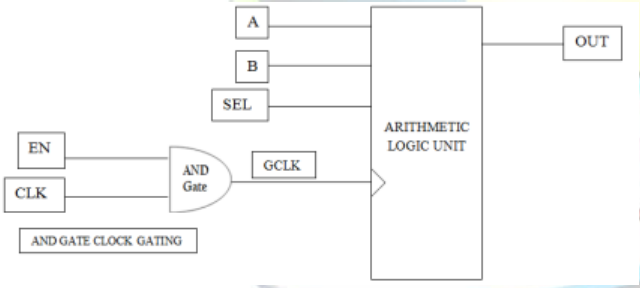

Fig. 7. View of ANDed Clock Gating

As seen from the design and implementation, the AND gate has two inputs EN1 and EN2. If EN1=EN2=1, the arithmetic block evaluated the inputs and accordingly generates the output and the flags. For all other sent of the inputs to AND gate the logical block evaluates the inputs. The device utilization is tabulated in table 2.

Table 2: Device Utilization with ANDed Clock Gating

Resource	Used	Available	Utilized (%)
IOS	80	232	34.48
Global Buffers	1	16	6.25
LUTs	88	9112	0.98
CLB Slices	22	2278	1.01
Dffs or Latches	34	18224	0.19

The design is compiled, simulated and the physical synthesis is carried out. From the plan ahead after physical implementation, the device utilization or the area obtained are different. The post synthesis generates the device utilization for the netlist as well as for the implemented. The report are represented in the Fig. 8 and 9 respectively.

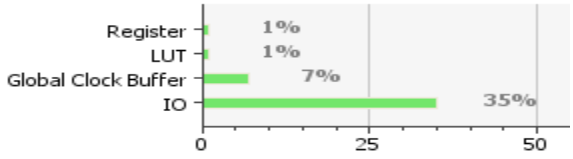

Fig. 8. Netlist Estimated Utilization in Percentage

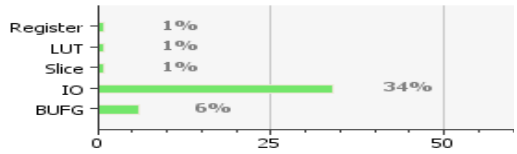

Fig. 9. Implemented Estimation in Percentage

The proposed clock gating techniques applied to an ALU performing 16 operations are simulated using the Xilinx ISE 9.2 and synthesized on 90nm Spartan-3E FPGA.

IV. Experimental Methodology

The simulated wave form of the ALU performing 16 operations using proposed clock gating is shown in the Fig. 10. From the simulated result shown in Fig. 10, it is observed that, the input is applied to A and B, the input to the select lines of logical and arithmetic functional units is also provided. The enable input EN of the DEMUX are initially made LOW, hence the output is not obtained at the OUT. Later, when the enable goes HIGH, the results are obtained at OUT. During this phase of evaluation the logical block is put to OFF state as no clock is provided. Hence, by this method of enabling a selective functional block in ON state leads to reduction of the power. In this second technique of applying the clock through the AND gate, when inputs EN1 and EN2 are HIGH, the arithmetic block is enabled, clock is provided and hence, the inputs are evaluated. For rest of the other combination of the inputs of the AND gate, will select the logical functional unit and hence, the logical block is evaluated for the given set of inputs.

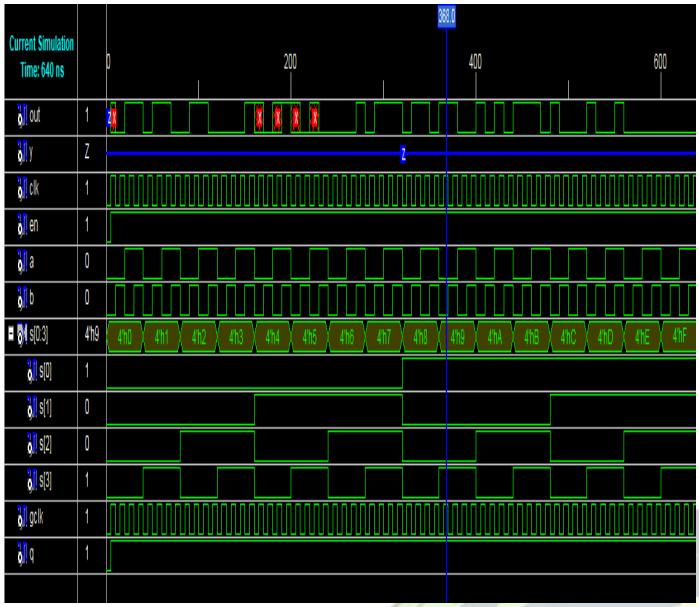

Fig. 10. Simulated waveform of the ALU performing 16 operations

V. Device Utilization

In this section, the comparison of the proposed techniques is presented. Firstly the device utilization or the gate area of the two techniques is tabulated in table 3.

Table 3: Comparison of the Device Utilization

Resource	DEMUXed clock gating	ANDed clock Gating
IOS	80	80
Global Buffers	01	01
LUTs	89	88
CLB Slices	23	22
Dffs or Latches	34	34
Number of gates	118	118
Number of instances	266	268

VI. RTL & Technology Schematic

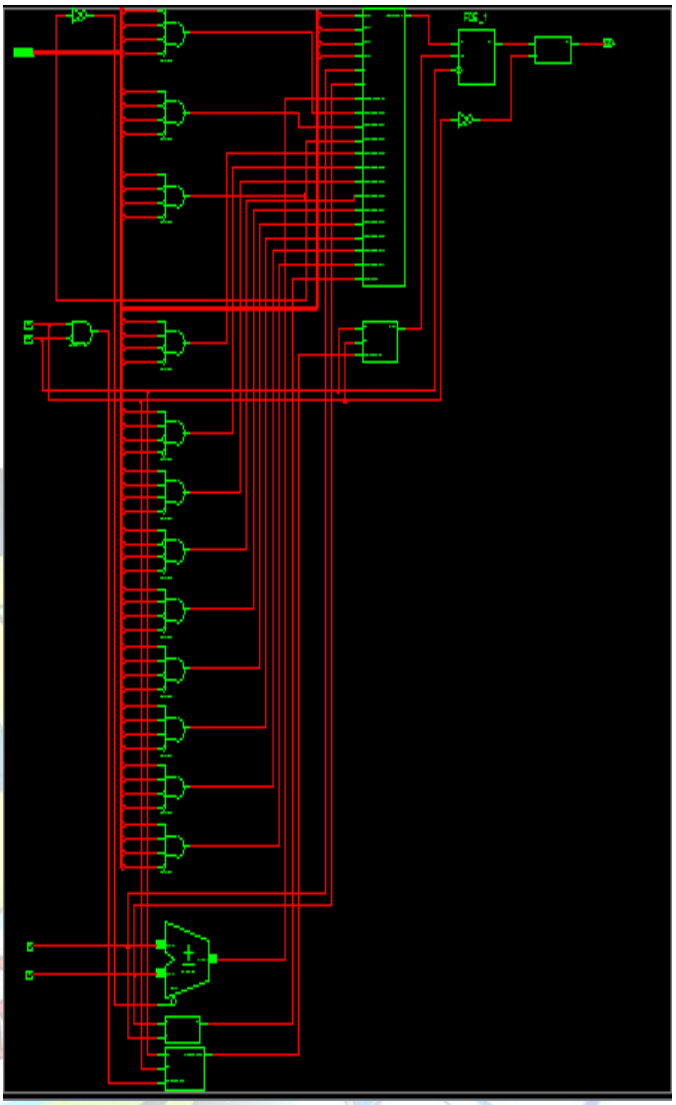

Fig. 11. RTL Schematic of ALU

RTL Schematic generated by Xilinx Synthesis Technology and save with the extension .ngr as shown in Fig 11. RTL Schematic shows a schematic representation of optimized design in terms of digital logic symbols that are not related to the targeted Xilinx FPGA device.

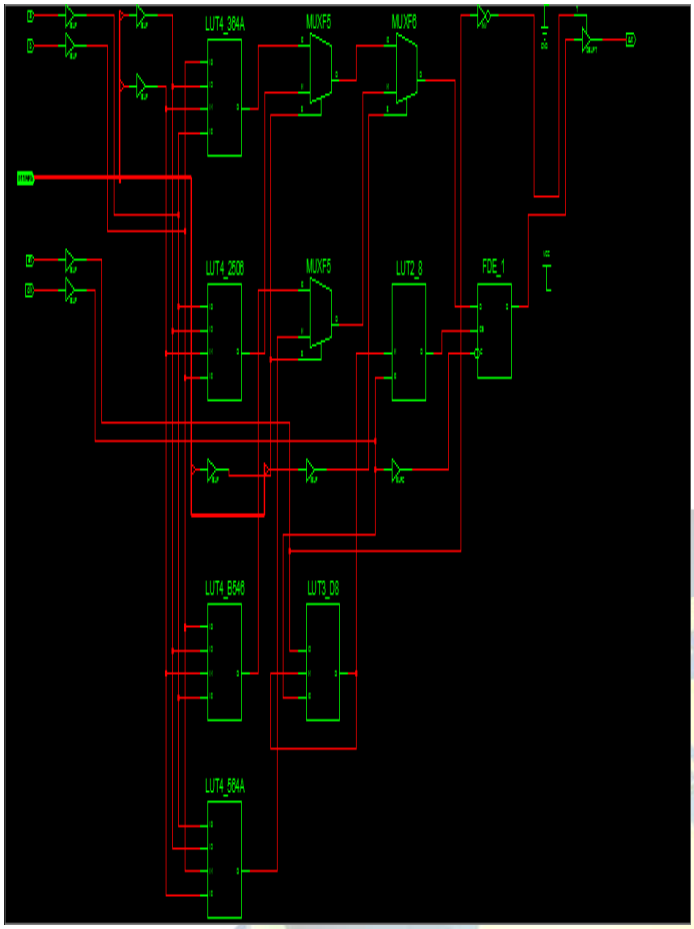

Fig. 12. Technology Schematic of ALU

Technology Schematic generated by Xilinx Synthesis Technology and save with the extension .ngc as shown in Fig 12. Technology Schematic shows a schematic representation of optimized design in terms of digital logic symbols that are not related to the targeted Xilinx FPGA device. Christo Ananth et al. [5] proposed a system in which the complex parallelism technique is used to involve the processing of Substitution Byte, Shift Row, Mix Column and Add Round Key. Using S- Box complex parallelism, the original text is converted into cipher text. From that, we have achieved a 96% energy efficiency in Complex Parallelism Encryption technique and recovering the delay 232 ns. The complex parallelism that merge with parallel mix column and the one task one processor techniques are used. In future, Complex Parallelism single loop technique is used for recovering the original message.

VII. CONCLUSIONS

An ALU performing 16 operations is design and developed for low power processor in the platform Xilinx ISE 9.2 and synthesized on 90nm Spartan-3E FPGA.

Frequency	Clock Power	Logic Power	Signal Power	IOs Power
100MHz	2mW	1mW	1mW	0Mw
1000MHz	17mW	9mW	10mW	4Mw
10GHz	168mW	48mW	88mW	41mW
100GHz	1679mW	153mW	802mW	410mW
1000GHz	16795mW	1198mW	7983mW	4099mW

In next phase using clock gating, we turn off rest 15 modules when any module is in execution then theoretical assumption is 93.75% power reduction.
Table 4 shows 90.32% clock power reduction using latch based clock gating.

Table 4: Latch based Clock Gating

Latch based Clock Gating	Total Power	Dynamic Power	Clock Power
Without Clock Gate	94mW	41mW	17Mw
With Clock Gate	77mW	25mW	2mW

Table 5 shows 75.82% clock power reduction using latch free clock gating.

Table 5: Latch Free based Clock Gating

Latch Free based Clock Gating	Total Power	Dynamic Power	Clock Power
Without Clock Gate	94mW	41mW	17mW
With Clock Gate	80mW	28mW	5mW

In this paper, the clock gating technique using DEMUXed and ANDed gating logic is applied to an ALU performing 16 operations is implemented on SPARTAN 3E FPGA. The power analysis is carried for a wide range of clock frequencies with all the resource type is performed and analyzed. From the results obtained, both can be used as clock gating techniques to reduce power. The keen observation and analysis of these two techniques, deliberate power reduction for clock, IO and dynamic. The average dynamic power reduction is 69.03% when compared between the two clock gating techniques. There is noticeable reduction in power for IO and dynamic using ANDed logic which is nearly an average of 67% and 59% respectively. The device utilization of the two techniques also highlight the area reduction using ANDed clock gating technique. This power reduction motivates the use of these clock gating techniques in low power high speed devices.

REFERENCES

[1] Mrs. Roopa Kulkarni and Dr. S.Y.Kulkarni, "Power Analysis and Comparison of Clock Gated Techniques Implemented on a 16-bit ALU", http://www.researchgate.net/publication/277005681.

[2] Peiyi Zhao, Zhongfeng Wang, - Guoqiang Hang, ―Power optimization for VLSI circuits and systems_, 10th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT), Nov 2010, pp 639 – 642.

[3] J. P. Oliver, J. Curto, D. Bouvier, M. Ramos, and E. Boemo, "Clock gating and clock enable for FPGA power reduction," in Proc. 8th Southern Conference on Programmable Logic (SPL), pp. 1-5, 2012.

[4] J. Shinde and S. S. Salankar, "Clock gating-A power optimizing technique for VLSI circuits," in Proc. Annual IEEE India Conference (INDICON), pp. 1-4, 2011.

[5] Christo Ananth, H.Anusuya Baby, "Encryption and Decryption in Complex Parallelism", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 3, Issue 3, March 2014,pp 790-795

[6] J. Kathuria, M. AyouKhan, Arti Noor,"A Review of Clock Gating Technique,"MIT International Journal of Electronics and Commmunication Engineering, Vol. 1 N0. 2 Aug 2011, Pg 106.

[7] P. R. Panda, B. V. N. Silpa, A. Shrivastava, and K. Gummidipudi, "Power-efficient System Design," Chapter 2 - Basic Low Power Digital Design, 2010.

[8] J. Castro, P. Parra, and A. J. Acosta, "Optimization of clock-gating structures for low-leakage high-performance applications," in *Proceedings of IEEE International Symposium on Efficient Embedded Computing*, pp. 3220-3223, 2010.

[9] J. Montanaro, et. al. A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor. IEEE Journal of Solid-State Circuits, (31) 11, Page 1703 – 1714, Nov. 1996.

[10] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations", *In Proc. of 27th Int'l Symp. Computer Architecture (ISCA)*, Jul. 2000, pp. 83-94.

[11] M. Gowan, L. Biro, and D. Jackson, "Power considerations in the design of the Alpha 21264 microprocessor", *In proc. of 35th Design Automation Conference (DAC)*, Jun. 1998, pp. 726-731.

[12] S. Palacharla, N. P. Jouppi, and J. E. Smith, "Complexity-effective superscalar processors", *In Proc. of 24th Annual Int'l Symp. on Computer Architecture (ISCA)*, Jun. 1997, pp. 206-218.