High Performance Microprocessor Design Methods Exploiting Information Locality and Data Redundancy for Lower Area Cost and Power Consumption

Byung-Soo Choi[1], Jeong-A Lee[2], and Dong-Soo Har[3]

[1] Ultrafast Fiber-Optic Networks Research Center
K-JIST(Kwangju Institute of Science and Technology)
1 Oryong-dong Puk-gu Gwangju, 500-712, Republic of Korea
bschoi@kjist.ac.kr
[2] Department of Computer Engineering
Chosun University
375 Susuk-dong Dong-gu Gwangju, 501-759, Republic of Korea
jalee@chosun.ac.kr
[3] Department of Information and Communications
K-JIST(Kwangju Institute of Science and Technology)
1 Oryong-dong Puk-gu Gwangju, 500-712, Republic of Korea
hardon@kjist.ac.kr

Abstract. Value predictor predicting result of instruction before real execution to exceed the data flow limit, redundant operation table removing redundant computation dynamically, and asynchronous bus avoiding clock synchronization problem have been proposed as high performance microprocessor design methods. However, these methods increase area cost and power consumption problems because of the larger table for value predictor and redundant operation table, and the higher switching activity in asynchronous bus. To resolve the problems of data tables for value predictor and redundant operation table, we have investigated partial tag and narrow-width operand methods, which have been recently proposed separately and present an efficient update method for value predictor and a table organization method for redundant operation table, respectively. To reduce excessive switching activity of asynchronous bus, we also propose a bus encoding method using frequent value cache, which reduces the same data transmissions. The proposed three methods – an efficient update method for value predictor, a table organization method for redundant operation table, and a frequent value cache for asynchronous bus – exploit information locality such as instruction and data locality as well as data redundancy. Analysis with a conventional microprocessor model show that the proposed three methods reduce total area cost and power consumption by about 18.2% and 26.5%, respectively, with negligible performance variance.

P.-C. Yew and J. Xue (Eds.): ACSAC 2004, LNCS 3189, pp. 170–184, 2004.

1 Introduction

Until a few years ago, performance improvement has been a key research issue in microprocessor design. Recently, however, the area cost and the power consumption of a microprocessor have been increased drastically as the number of transistors keeps increasing. As a result, research interest has been shifted to performance improvement while maintaining the efficiency of area cost and power consumption. In this paper, several design methods have been investigated for a high performance microprocessor with an emphasis on achieving efficient area cost and power consumption.

Among many design techniques for a high performance microprocessor, three methods are investigated such as value predictor, redundant operation table, and asynchronous dual-rail bus in this research. The value predictor predicts a result of an instruction before the instruction is actually executed. Hence dependent instructions can be executed at the same time when the instruction is executed. On the other hand, the redundant operation table stores recently executed instructions in a table and checks whether the current executable instruction is already stored in the table. In other words, the redundant operation table can skip the real execution of an instruction by a simple lookup procedure with the table, subsequently shortening the execution time of the instruction. Another alternative design technique, the asynchronous dual-rail bus is a reliable bus scheme for a complex system such as a futuristic high performance microprocessor. The asynchronous dual-rail bus can transmit data in a reliable fashion by making use of the dual-rail encoding, which combines the data and the control signals.

Analyzing the aforementioned three design methods from the area cost and power consumption points of view, several attempts are made especially to find some locality and redundancy of data used in each design method. Several information localities and data redundancies were found, which causes extra area cost and power consumption. More specifically, the value predictor and the redundant operation table store the same or a little different instructions (instruction locality), small operand values (operand data locality), and small result values (result data locality), whereas the asynchronous dual-rail bus transmits the same data items repeatedly (communication data locality). From what we observed about these localities, a conclusion was reached that each design method can be further enhanced for lower area cost and lower power consumption by exploiting such localities to reduce redundancy.

In this paper, we propose three enhanced methods as follows. First, for value predictors, we propose a method to combine the two previously proposed area cost reduction methods such as partial-tag and narrow-width methods. Second, we designed a partial resolution method to reduce the area cost of the tag fields in the redundant operation table. Third, we applied the previously proposed frequent value cache method into an asynchronous dual-rail bus to minimize the communication data redundancy.

As the last step, we investigated total area cost and power consumption reduction effects in a conventional microprocessor model. By using the proposed

methods, the total area cost and power consumption in a microprocessor model would be reduced by about 18.2% and 26.5%, respectively.

This paper is organized as follows. Section 2 describes related work as three high performance design methods, information locality, and data redundancy. The proposed area and power reduction methods for value predictor and redundant operation table are described in Section 3 and 4, respectively. Also a designed power reduction method for asynchronous dual-rail bus is explained in Section 5. Meanwhile, total area cost and power consumption reduction effects in a microprocessor model are analyzed in Section 6. Section 7 concludes this research.

2 Related Work

2.1 High Performance Design Methods

Value Predictor: Value predictors have been proposed to overcome the data dependency problems in the instruction-level parallelism by predicting a result value of an instruction before its actual execution [1], [2].
Redundant Operation Table: When the instruction-level parallelism increases, there are many side effects. One of the side effects is the increased number of redundant executions because of speculative executions due to branch predictor or value predictor. Unfortunately, speculative or redundant operations limit the performance improvement and increase the power consumption as well [3]. To overcome such negative effects, many optimization methods have been proposed [4], [5]. One typical solution is eliminating redundant operations, where redundant executions of complex operations are replaced by simple table lookup operations [6].
Asynchronous Dual-Rail Bus: Because of the steady increase of the number of components in a chip, SOC design methods have been studied intensively and will be used for a futuristic high performance microprocessor. To succeed in the market, the time-to-market and the reliability of a SOC are very important. To help the design efforts for a short design time and reliability of SOCs, asynchronous design methods [7] have been studied recently. For a reliable asynchronous bus structure in SOC designs, the dual-rail data encoding method [8] has been intensively investigated.

2.2 Information Locality and Data Redundancy

Information Locality: Information localities in a microprocessor are defined, which are related to instructions, operands of instructions, results of instructions, and communication data over bus. First, *instruction locality* is defined as a small number of instructions is repeatedly or frequently executed, and usually the instructions are located closely to each other in a given time interval. Second, *operand locality* is defined as the data value of the operand is small in most instructions and can be represented with small number of bits. Third,

result locality is defined as the results of most instructions are small which can be represented with small number of bits. Last, *communication data locality* is defined as a bus transmits the same or very similar data repeatedly or frequently in a given time interval.

Data Redundancy: Considering the above information localities, we can infer that there are data redundancy in instructions, operands of instructions, results of instructions, and communication data over bus, respectively. First, *data redundancy of instructions* is occurred when the instruction addresses in a given time interval are not so different, which can be inferred that the higher bits of instruction address are the same. Hence the higher bits of addresses of executed instructions in a given time interval are redundant. Second, *data redundancy of operands* is occurred when most operands of executed instructions in a given time interval require a small number of bits, and hence the higher bits of operands are considered as redundant bits. Third, *data redundancy of results* is occurred when the most results of executed instructions in a given time interval require a small number of bits, and hence the higher bits of results are redundant. Last, *data redundancy of communication data* is occurred when the most communication data in a given time interval are the same or similar, and hence most communications are redundant.

3 Value Predictor

3.1 Table Structure

In this research, we explain only the stride predictor for the simplicity. The stride predictor assumes that consecutive result values of an instruction have the same stride value [1]. Usually, a value predictor exploits a large data table to store required information and is referenced by the instruction address.

3.2 Combining Partial Tag and Narrow-Width Operand Method

Two Methods to Reduce Area and Power of Value Predictor: To reduce the area cost and power consumption of value predictor, two methods have been already proposed as follows.

Partial Tag Method: Instruction or data caches are usually based on a correct association between an instruction address and an indexed entry because the lookup data must be the same value as the previously stored value. In the value predictor, however, a lookup data is a prediction value so that it does not always require the correct association between a lookup address and an indexed data. Based on such a loose association between a lookup address and an indexed data, a value predictor does not necessarily use a full-tag, but can use a partial-tag, which reduces the area cost of the tag part [9]. Briefly, the full-tag method takes an address as a tag except for index bits, but the partial-tag method only uses some part of a full-tag.

Narrow-Width Operand Method: Analysis of the result values of a program shows that only a few result values require a full precision value supported by processor

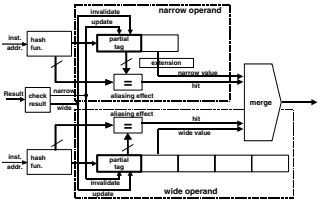


Fig. 1. Combining Method of Partial-Tag and Narrow-Width Methods

registers. Taking into account such locality, the narrow-width operand method classifies result values into two types as the narrow-width and the wide-width result values according to the required number of bits [10]. For the purpose of area cost reduction in data tables, the narrow-width operand method utilizes both the narrow-width and wide-width tables for storing the narrow-width and wide-width result values, respectively. If a result value of an instruction requires fewer bits than the predetermined number of bits, prediction information of the instruction is stored in the narrow-width table. Otherwise, prediction information is stored in the wide-width table. Because the narrow-width table stores fewer bits for each result value, it reduces the overall area cost of a data table.

Combining Partial Tag and Narrow-Width Operand Methods: To date, two area cost reduction methods for value predictors have been proposed independently. In the present research, a combining method with an efficient table-update method is proposed to minimize the performance degradation. A simple method combining these two methods is conceivable. However, such a simple superposition method decreases the performance improvement ratio because two prediction values are generated from the two tables.

We propose a new table-update method as shown in Figure 1. When the result of an instruction is classified into a narrow-width result(wide-width result), the instruction is stored in the narrow-width table(wide-width table). At the same time, the wide-width table(narrow-width table) invalidates an indexed entry if the entry contains the same partial-tag with the instruction. In short, depending on the classification result of an instruction, only one of the two tables stores the instruction, and the other table must invalidate a corresponding entry if the tag is the same with the referenced address.

3.3 Analysis

To measure the effect of the proposed area reduction method, the die size and the power consumption of value predictor are measured by using CACTI 3.2 [11]. We also investigated IPC value when the proposed method is used with a SimpleScalar [12] model and SPEC 95 [13] benchmark programs. However, as we expected, the IPC value changes very little about 1%. Hence we skip the explanation of IPC variation when the proposed method is used.

Area Cost: Table 1 describes area cost reduction ratios over the conventional stride predictor. The reduction of area cost is higher with the narrow-width

Table 1. Area Cost and Power Consumption Reduction Ratios with Different Area Reduction Methods for Stride Predictor

Area Reduction Methods	Area Cost				Power Consumption			
	Number of Entries (K)				Number of Entries (K)			
	32	16	8	4	32	16	8	4
Narrow-Width	57%	55%	56%	47%	16%	17%	27%	24%
Partial-Tag	20%	19%	20%	20%	39%	31%	46%	49%
Proposed Combining Method	***72%***	***74%***	***72%***	***68%***	***58%***	***51%***	***63%***	***68%***

method than with the partial-tag method. The reason is as follows. The reduction ratio depends on the portion of the area cost reduced by the partial-tag and the narrow-width methods. The partial-tag method can decrease the area cost of the tag part only; however, the narrow-width method can decrease the area cost of all result values. Meanwhile, the proposed combining method decreases the area cost more than other area cost reduction methods. The proposed combining method decreases the area cost by about 71% for the stride predictor.

Power Consumption: Table 1 also describes power consumption reduction ratios over the conventional stride predictor. The reduction of power consumption is higher with the partial-tag method than with the narrow-width method. The reason is as follows. The power consumption of the tag part is larger than that of data part since each tag comparison requires more power consumption. Meanwhile, the proposed combining method decreases the power consumption more than other area cost reduction methods. The proposed combining method reduces the power consumption by about 61% for the stride predictor.

4 Redundant Operation Table

4.1 Table Structure

In a redundant operation table, operands are partitioned into two parts: an index and a tag parts. Meanwhile, all operations are classified into integer or floating-point operations. Hence redundant operation tables have different structures depending upon the operation type.

4.2 Narrow-Wide-Width Table

A preliminary analysis of operands for integer and floating-point operations in a SimpleScalar [12] microprocessor with SPEC [13] benchmarks reveals that most operands can be represented with a small number of bits. A partial resolution method is proposed to exploit this characteristics. The partial resolution method eliminates the area cost to store redundant bits for consecutive 0s in the higher bits for integer operands and the lower bits for floating-point operands in the conventional wide-width redundant operation table. A wide-narrow-width redundant operation table utilizing the partial resolution method is designed as shown in Figure 2. The wide-narrow-width redundant operation table dynamically classifies operations into wide-width and narrow-width operations depending on the

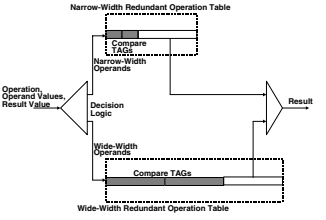


Fig. 2. Wide-Narrow-Width Redundant Operation Table

operand bit width. When the operation requires narrow-width operands, the instruction is stored in the narrow-width redundant operation table. Otherwise, the instruction is stored in the wide-width redundant operation table. Note that the concept of the partial resolution method is similar to the partial-tag method [9], which is proposed to apply for value predictors. The partial-tag method for value predictors stores imprecise tag information, but the partial resolution method for redundant operation table should store precise tag information. Hence, the partial-tag method for value predictor cannot be directly used for the redundant operation table.

4.3 Analysis

Note that we also investigated IPC value when the proposed method is used with a SimpleScalar [12] model and SPEC 95 [13] benchmark programs. However, since the IPC variance is very little, we skip the explanation of IPC variation when the proposed method is used.

Area Cost: The area cost of the conventional wide-width redundant operation table can be calculated easily. Meanwhile, the wide-narrow-width redundant operation table consists of two subsidiary predictors, hence the area cost of it is calculated by the summation of each area cost for narrow-width and wide-width tables. Based on the above considerations and methods, the relative area cost is measured as shown in Table 2. Note that the models containing above 512 entries are measured, since the redundant operation table usually requires many entries. As the table explains, the proposed partial resolution method reduces the area cost by 20%, for FP 2048-entry, at the maximum.

Power Consumption: Since the conventional wide-width redundant operation table is referred for all lookups, it can be easily calculated the dynamic power consumption of the wide-width table. On the other hand, since each subsidiary table in the wide-narrow-width redundant operation table is referred with different lookup ratios, the lookup ratio of each table should be considered. Hence the total dynamic power consumption of the proposed wide-narrow-width redundant operation table is calculated by the summation of each power consumption of narrow-width and wide-width tables considering each lookup ratios. Based on the above considerations and methods, the relative dynamic power consumption reduction ratio is measured as shown in Table 2. As the table explains, the

Table 2. Relative Area Cost and Power Consumption Reduction Ratio

Reduction Ratio over Wide-Width Table		Number of Entries	
		2048	512
Area Cost	INT	7%	9%
	FP	20%	10%
Power Consumption	INT	34%	24%
	FP	30%	31%

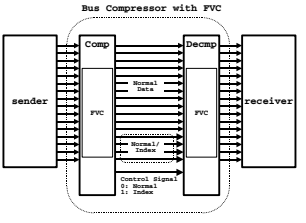


Fig. 3. Frequent Value Cache augmented Bus Scheme

proposed partial resolution method reduces the dynamic power consumption by about 34%, for INT 2048-entry, at the maximum.

5 Asynchronous Dual-Rail Bus

5.1 Frequent Value Cache

One-of-four data encoding method reduces the power consumption of the dual-rail encoding method by decreasing switching activities [14]. Meanwhile, the data pattern analysis illustrates that many data items are repeatedly transmitted in accordance with the result in [15]. Hence we can conclude that the conventional dual-rail and the previously proposed one-of-four data encoding methods waste the power when the data bus transmits the same data items repeatedly.

To reduce such waste of power, we proposed a different method, which utilizes a buffer to exploit the feature of repeatedly transmitted data item. The proposed buffer stores data items and sends an index for a data item when the data item to be sent is already stored in the buffer. Since the index requires fewer number of bits than the data itself, the wasted bandwidth or the switching activity can be decreased, resulting in low power consumption.

Figure 3 describes a frequent value cache(FVC) very briefly that stores data items of each communication. The normal *sender* and *receiver* deliver a data item with a normal fashion, while the *Comp* and *Decmp* deliver a data item by a data itself or an index of *FVC* depending on the hit of *FVC*. When a data itself is transferred, all bus lines are used; however, when an index of the data item is transferred, only the index lines are used. Thus, the index lines are used for both an index and a data item. To distinguish whether a transmitted information represents an index or a normal data item, a *control* signal is used.

5.2 Analysis

Three measures as hit ratio, switching activity reduction ratio, and power consumption reduction ratio are investigated. The hit ratio is the most important one since it decides the switching activity reduction ratio that finally determines the power consumption reduction ratio. To analyze, we investigated a memory bus in SimpleScalar model [12] and SPEC95 benchmark [13] programs.

Hit Ratio: We found the following conclusions through investigating data patterns over the above memory bus. First, even only one entry of FVC can detect 40% of the repeatedly transmitted data items. Second, over 256 entries can represent most data items.

Switching Activity: From the high hit ratio of the FVC, it is required to know how much switching activity can be reduced. In the research, only the change of signal levels between consecutive data items are measured to calculate the switching activity ratio of a bus. The normal dual-rail bus utilizes all 32-bit logical bits and each signal line causes two switchings, hence the switching activity is 32×2. Meanwhile, FVC delivers an index for a hit case and a normal data item for a miss case. In addition, the control signal changes for every communications, hence it changes two times for each communication. Therefore, the switching activity when FVC is used is calculated by Equation 1.

$$P_{hit} \times \{1 + \log(\#entry)\} \times 2 + (1 - P_{hit}) \times (1 + 32) \times 2 \tag{1}$$

Based on the above analysis, the switching activity reduction ratio of FVC over the normal dual-rail bus model is calculated by Equation 2.

$$\frac{P_{hit} \times \{1+\log(\#entry)\} \times 2 + (1-P_{hit}) \times (1+32) \times 2}{32 \times 2} \tag{2}$$

Analysis result illustrates that FVC reduces the switching activity of the conventional model by 75% at maximum. However, the switching activity reduction ratio is decreased after the maximum point because of the increased number of index bits.

Power Consumption: The total power consumption should include the power consumption of the FVC tables although the power consumption ratio of the table would be below 5% as explained in [16]. In addition, the power consumption of the bus itself should be considered as well. To measure the power consumption of FVC table and bus lines, it is assumed that 0.25 micron technology is used, and the length of the bus line is 10 mm, which follows the 2001 ITRS [17]. Power consumptions of the normal model and the FVC model are as follows:

Normal Model: The power consumption is only caused by the dual-rail bus for logical 32-bit bus lines. Based on the 0.25 micron technology, we assume that 10 mm bus lines consume about 0.4 nJ by using power measure tools.

FVC Model: The power consumption is caused by two parts as the FVC table and bus lines. To measure the power consumption of the FVC table, CACTI tool [11] is used. Since all entries should be checked at the same time, it is assumed that the table is a fully-associative content address memory. The power consumption of FVC model can be formulated as Equation 3. Specifically, the

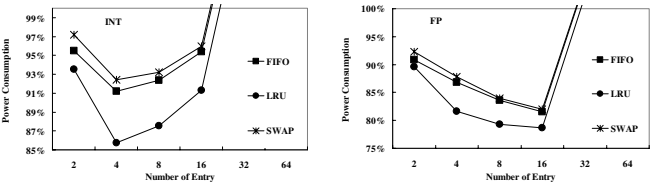


Fig. 4. Power Consumption Variation

power consumption of the FVC table is multiplied by two because FVC model requires two FVC tables for a sender and a receiver. Meanwhile, when the FVC miss, each FVC table must be updated and it consumes more power. To include this power consumption to update FVC, we include the $Miss\_Ratio$ in the equation.

$$\begin{aligned} &Table\_Power \times 2 \times (1 + Miss\_Ratio) \\ &+Bus\_Power \times Switching\_Activity\_Reduction\_Ratio \end{aligned} \tag{3}$$

Finally, it can be derived a power consumption reduction ratio of the FVC model over the normal model as shown in Equation 4.

$$\frac{Table\_Power \times 2 \times (1 + Miss\_Ratio) + (0.4nJ) \times Switching\_Activity\_Reduction\_Ratio}{0.4nJ} \tag{4}$$

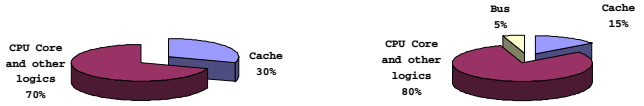
Figure 4 shows the power consumption reduction ratio when the FVC model is used. From the figure, it can be concluded that FVC reduces the total power consumption by about 14% and 22% at maximum for integer and floating-point benchmarks, respectively.

6 Analysis in a Microprocessor

Until previous sections, it have been analyzed independently the area cost and/or power consumption reduction ratios of the proposed methods for value predictor, redundant operation table, and asynchronous dual-rail bus. Meanwhile, because our main goal is to reduce the total area cost and power consumption of a high performance microprocessor, it is needed to know how much area cost and power consumption can be reduced when the proposed methods are used for each design method.

6.1 Area Cost and Power Consumption Breakdowns

Because no processor has been implemented with the value predictor, redundant operation table, and asynchronous dual-rail bus at the same time, it is required to model a futuristic microprocessor to investigate the portions of area cost and power consumption of each design method.



(a) Area Cost (b) Power Consumption

Fig. 5. Area Cost and Power Consumption Breakdowns of Alpha 21264 Model

Conventional Model: The Alpha 21264 microprocessor [18], [19] is selected to find the breakdown of die size and power consumption of major blocks such as cache and core parts. Because value predictor and redundant operation table have similar structure with the cache, it can be assumed that the area cost and power consumption of tables for value predictor and redundant operation table are calculated by the relative area cost and power consumption over the cache.
Area Cost and Power Consumption Breakdown: Alpha 21264 utilizes 128Kbyte Instruction and Data caches, which require about 30% of total area cost [18] and consumes about 15% of total power consumption [19]. Figure 5(a) and 5(b) show the breakdowns of area cost and power consumption of the Alpha 21264 model, respectively.
New Model: The new Alpha 21264 model consists of the old Alpha 21264 and other three design methods. Because of such modification of the old Alpha 21264 model, the area cost and power consumption breakdowns will be changed.
Area Cost Breakdown: The area cost of caches is about 30% and the others about 70% in the old Alpha 21264 processor. However, the value predictor and redundant operation tables add more area cost as 164Kbyte and 144Kbyte, respectively. In the old Alpha 21264 processor, 128Kbyte cache uses about 30% of total die size, hence it can be inferred that the value predictor increases the total area cost by about 38.4%, which is calculated by 30%*164/128. Also, the redundant operation table adds about 33.8% of total area cost, which is calculated by 30%*144/128. Finally, the total area cost is increased by about 72.2%, which is calculated by the summation of the extra area cost of value predictor and redundant operation table. From this total area cost increase, it should be rearranged the portion of area cost of each component as 17.4% for cache, 22.3% for value predictor, 19.6% for redundant operation table, and 40.7% for others as shown in Table 3. As shown in the table, it can be known that the portions of area cost for value predictor and redundant operation table are large, about 42%.
Power Consumption Breakdown: On the other hand, the portions of additional power consumption of value predictor and redundant operation table can be calculated by the relative power consumption over cache. It is inferred that the stride type value predictor consumes five times as much energy as cache from [4]. Hence, the value predictor consumes more energy by about 96.1%, which is calculated by 15%*(164/128)*5. Meanwhile, the redundant operation table also consumes more energy by about 16.9%, which is calculated by 15%*(144/128). From this increased total power consumption, it should be rearranged the portions of power consumption of each block as 7.1% for cache, 2.3% for bus, 45.1% for value

Table 3. Area Cost Breakdown, Reduction Ratios, Relative Reduction in the new Alpha 21264

Parts	Portion	Reduction Ratio	Relative Reduction
Cache	17.4%		
CPU Core	40.7%		
Value Predictor	**22.3%**	**64%**	**14.3%**
Redundant Operation Table	**19.6%**	**20%**	**3.9%**
Total	100%		**18.2%**

Table 4. Power Consumption Breakdown, Reduction Ratios, Relative Reduction in the new Alpha 21264

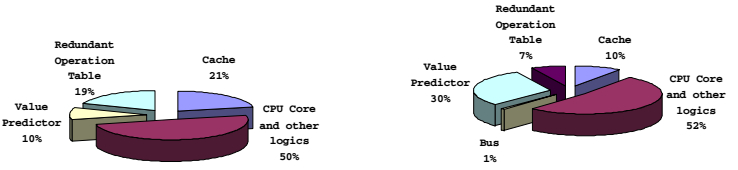
Parts	Portion	Reduction Ratio	Relative Reduction
Cache	7.1%		
CPU Core	37.6%		
Bus	**2.3%**	**14%**	**0.3%**
Value Predictor	**45.1%**	**52%**	**23.5%**
Redundant Operation Table	**7.9%**	**34%**	**2.7%**
Total	100%		**26.5%**

predictor, 7.9% for redundant operation table, and 37.6% for CPU Core as shown in Table 4. The extra power consumption caused by value predictor, redundant operation table, and asynchronous dual-rail bus is very large, about 55%.

6.2 Reduction of Total Area Cost and Power Consumption

Reduction of Total Area Cost and Power Consumption:
Area Cost Reduction: When the proposed area cost reduction methods for value predictor and redundant operation table are used, the total area cost can be reduced by about 18.2%, which is calculated by the summation of reduction ratios of area cost for value predictor (22.3% * 64% = 14.3%) and redundant operation table (19.6% * 20% = 3.9%), as shown in Table 3.
Power Consumption Reduction: Meanwhile, the proposed power consumption reduction methods can decrease the power consumption of each design method by about 52% for value predictor, 34% for redundant operation table, and 14% for asynchronous dual-rail bus, which are shown in Table 4. Therefore, the proposed area cost and power consumption reduction methods reduce the power consumption by about 23.45%(=45.1%*52%), 2.7%(=7.9%*34%), and 0.3%(=2.3%*14%), respectively, and finally the total power consumption by about 26.5% as shown in Table 4.

Area Cost and Power Consumption Breakdowns:
Area Cost Breakdown: The portions of area cost of value predictor and redundant operation table are changed as shown in Figure 6(a). As shown in the figure, the total portion of area cost for value predictor and redundant operation tables is reduced from 42% to 29%.



(a) Area Cost (b) Power Consumption

Fig. 6. Area Cost and Power Consumption Breakdowns of Area Cost Reduced Alpha 21264 Model

Power Consumption Breakdown: The portions of power consumption of value predictor, redundant operation table, and asynchronous dual-rail bus are changed as shown in Figure 6(b). As shown in the figure, it can be known that the total portion of power consumption of value predictor, redundant operation table, and asynchronous dual-rail bus is reduced from 55% to 38%.

7 Conclusion

Throughout this research, we have pointed out that the low area and power design methods should be proposed for design techniques for a high performance microprocessor. Among many techniques, three high performance design techniques have been investigated.

Analysis of information locality and related data redundancy illustrates that the area and power are wasted by the data redundancy in each high performance design method. Therefore, the information locality was exploited and tried to minimize data redundancy in each method. Finally, three different approaches have been proposed for each method respectively.

First, to reduce the waste of area cost and power consumption in a value predictor, which is caused by data redundancy in tag and data part, we proposed a combining method of previously proposed partial tag and narrow-width method with an efficient table-update method. Structural and dynamic analysis show that the proposed method reduces the area cost by about 71% and the power consumption by about 61% over the conventional value predictor. Second, for the redundant operation table, we designed a partial tag method. Although the redundant operation table wastes area and power in both tag and data parts, a redundancy minimization method only for tag part has been discussed. The proposed method reduces the area cost by about 20% and the power consumption by about 34% over the ordinal redundant operation table structure. Third, to reduce the waste of power consumption of asynchronous dual-rail bus, we utilize the frequent value cache with several circuits. Analysis results show that the proposed method decreases the power consumption of a bus in a microprocessor by about 14% for integer and 22% for floating-point data communications over a memory bus in a microprocessor.

As well, we examined how much total area cost and power consumption can be reduced when the proposed area cost reduction methods are used for each design method. This analysis confirmed that the total area cost and power consumption would be reduced by about 18.2% and 26.5%, respectively.

Acknowledgments. This work was supported in part by the Korea Science and Engineering Foundation (KOSEF) through the Ultra-Fast Fiber-Optic Networks Research Center at Kwangju Institute of Science and Technology.

References

1. Mikko H. Lipasti and John P. Shen: Exceeding the Dataflow Limit via Value Prediction, Proc. of 29th Intl. Symp. on MICRO, (1996) 226-237
2. Sang-Jeong Lee, Yuan Wang, and Pen-Chung Yew: Decoupled Value Prediction on Trace Processors, Proc. of 6th IEEE Intl. Symp. on HPCA (2000) 231-240
3. Rafael Moreno, Luis Pinuel, Silvia del Pino, and Francisco Tirado: A Power Perspective of Value Speculation for Superscalar Microprocessors, Proc. of ICCD, (200) 147-154
4. Ravi Bhargava and Lizy K. John: Latency and Energy Aware Value Prediction for High-Frequency Processors, Proc. of the 16th ICS (2002) 45-56
5. Ravi Bhargava and Lizy K. John: Performance and Energy Impact of Instruction-Level Value Predictor Filtering, Proc. of the First Workshop of Value-Prediction (2003)
6. Daniel Citron and Dror G. Feitelson: Hardware Memoization of Mathematical and Trigonometric Functions, Technical Report-2000-5, Hebrew University of Jerusalem (2000)
7. Scott Hauck: Asynchronous Design Methodologies: An Overview, Proc. of the IEEE (1995) Vol.83, No.1, 69-93
8. Tom Verhoeff: Delay-Insensitive Codes: An Overview, Distributed Computing (1988) Vol.3, 1-8
9. Toshinori Sato and Itsujiro Arita: Partial Resolution in Data Value Predictors, Proc. of ICPP (2000) 69-76
10. Toshinori Sato and Itsujiro Arita: Table Size Reduction for Data Value Predictors by exploiting Narrow Width Values, Proc. of ICS (2000) 196-205
11. Premkishore Shivakumar and Norman P. Jouppi: CACTI 3.0: An Integrated Cache Timing, Power, and Area Model, WRL Research Report 2001/2, COMPAQ Western Research Laboratory (2001)
12. Doug Burger and Todd M. Austin: The SimpleScalar Tool Set, Version 2.0, Technical Report, CS-TR-97-1342, University of Wisconsin (1997)
13. SPEC CPU Benchmarks: Standard Performance Evaluation Cooperation http://www.specbench.org/osg/cpu95
14. John Bainbridge and Steve B. Furber: Delay Insensitive System-on-Chip Interconnect using 1-of-4 Data Encoding, Proc. of ASYNC. (2001) 118-126
15. Benjamin Bishop and Anil Bahuman: A Low-Energy Adaptive Bus Coding Scheme, Proc. of the IEEE Workshop of VLSI (2001) 118-122
16. Tiehan Lv, Jorg Henkel, Haris Lekatsas, and Wayne Wolf: An Adaptive Dictionary Encoding Scheme for SOC Data Buses, Proc. of DATE (2002) 1059-1064

17. The Semiconductor Industry Association: The International Technology Roadmap for Semiconductor (2001)
18. Srilatha Manne, Artur Klauser, and Dirk Grunwald: Pipeline Gating: Speculation Control for Energy Reduction, Proc. of ISCA (1998) 122-131
19. Michael K. Gowan, Larry L. Biro, and Daniel B. Jackson: Power Considerations in the Design of the Alpha 21264 Microprocessor, Proc. of DAC (1998) 726-731