Scheduling to Timing Optimization for a Novel High-level Synthesis Approach

Ling Li[1], Teng Wang[1], Ziyi Hu[1], Xin'an Wang[1]*, Xu Zhang[1]

[1]Key Lab of Integrated Microsystem, Peking University Shenzhen Graduate School, Shenzhen 518055, China
* Email: wangxa@ szpku.edu.cn

Abstract

Traditional IC design methodology based on standard cells shows its limitation on design efficiency, which can not satisfy the needs for shorter time-to-market and more advanced functionality of IC products. To solve this problem, a novel high level synthesis method named operator design method is proposed. In this paper, a scheduling scheme to timing optimization for operator design method is proposed, which is carried out based on the attributes of the operand in the operation and dependence of the operations. The experiment results proves the feasibility and the efficiency of the operator design method, and obtains a 65% faster data-processing capacity and 30% reduction in hardware cost than that of SPARK tool of University California at San Diego.

Keywords-high-level synthesis; operator design method; rapid design; scheduling

1. Introduction

The rapid development of IC manufacturing technology has brought higher desires to IC design. Traditional low abstraction design levels can't meet with the requirement of design efficiency and production performance [1]. Design technology that can be used in rapid designed and automatable implementation is seriously in need, which in turn allows designers to explore the design space efficiently and rapidly.

High-Level Synthesis (HLS) technology, the key technique of designing complex structure such as CPU and DSP, can speed up IC design largely. Currently, the goal of HLS technology is RTL-level design, which reduces the needed source code by 10 times [2]. Scheduling, recognized as a key step in HLS, tries to balance the number of operations of every different type executed per clock cycle. One simple scheduling approach is to schedule all the operations as soon as possible (ASAP) [3][4][5][6]. The converse approach is to schedule the operations as late as possible (ALAP). The force-directed scheduling (FDS) [7] schedules operations based on their urgency and mobility respectively. There are several other types of scheduling approaches which either iteratively reschedule the design [8] or schedule operations along the critical path first through the behavioral description [9]. However, HLS has not been widely accepted in the engineering practice, which is primarily due to the quality loss when compared with manual designs and the various limits while using HLS [10], and in most cases, the intelligent strategies which these scheduling algorithms typically focus on to automate the optimization of one or more objective functions under a set of designs, is not reachable, and thus, some hardware waste appears [11].

Key Lab of Integrated Microsystems Science and Engineering Applications of Peking University Shenzhen Graduate School has been conducting a continuous research on the methodology about how to speed up IC design, and also has proposed a novel IC design methodology named Operator Design Method[12]. After a series of steps like algorithm analysis, generating operator structure graph, aggregation, time-marking, verification and optimization, C or MATLAB code can be easily translated into operator language, and hardware structure can be generated automatically [13]. In this paper, a standard flow aiming at computer aided analysis and transforming for operator design method is described and a novel scheduling scheme to timing optimization is proposed due to increasing design complexity and time-to-market pressure.

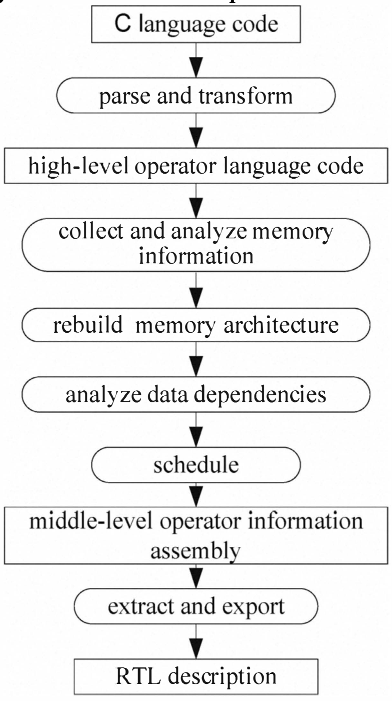

C language code
parse and transform
high-level operator language code
collect and analyze memory information
rebuild memory architecture
analyze data dependencies
schedule
middle-level operator information assembly
extract and export
RTL description

Figure 1. standard design flow of operator method

2. Principle of Operator Design Method

A standard flow aiming at computer aided analysis and transforming for operator design method, a novel HLS method is proposed, as shown in Fig. 1 In this paper several key steps will be picked up and discussed in detail.

Key Step 1: Parse and transform.

In this step, a sort of computer language, such as C programming language, is the input of the design flow. The task of this step is to analyze each segment of the high level language, pre-compile and transform code, which is completed by language identification tool. So a novel language named high-level operator language is proposed, balancing the difference between high level language and hardware language, in which some of segment control configurations, data types (such as array, pointer, structure, etc.), and operators in C programming language are got rid of, the form of function defining and transferring is modified, macro-define is replaced, and the ports of input and output are picked up.

Key Step 2: Collect and analyze memory information.

In the high-level operator language, two types of variables, mem and addr, are added, which are mapped from data arrays and pointers. The task of this step mainly focuses on collecting and analyzing the information about mem and addr variables to frame the memory architecture according to a set of rules.

Key Step 3: Rebuild memory architecture.

Based on the information about data array and pointer collected in step2, the memory architecture will be designed, which includes the memory word width and the connection form when it is shared between two or more function modules. The address should be calculated according to a set of rules because the memory word width of the primary code is changed.

Key Step 4: Analyze data dependencies and schedule.

The task of this step is to pick-up key operations, analyze the data dependencies, schedule operations, and extract timing information, completed by language identification tool, so that hardware transform was performed conveniently on the analysis results. Scheduling is a connecting link between the preceding and the following in transforming high-level operator language into hardware language.

The time-marking is extracting timing information from data dependencies and application requirement, as well as the constraint of the technology and frequency in operator design library. After that the timing information is drew into operation-control flow graph.

Key Step 5: Extract and export.

According to the aggregation of memory information, data information, key operation information, and operation-control flow graph, Register Transfer Level (RTL) description can be generated under certain rules.

3. Scheduling algorithms

3.1 Concept of Scheduling

For the purpose of time saving and resource sharing , scheduling algorithm was developed in HLS flow. Scheduling, considered as the bridge of hardware design and algorithm description, is to determine the time step or clock cycle in which each operation in the design is executed. Scheduling always leads to shorter schedule lengths and higher resource utilization to improve design performance.

Scheduling algorithm distributes the different operations of the whole algorithm into the appointed cycle and tries to balance the number of operations of every different type executed per clock cycle for code parallelization according to data dependency and constraint requirement, so as to reduce the whole time of instruction code and achieve the optimization or better scheduling result which meets with the constraint requirement.

Resource and time are the main factors of all constraint requirements that should be considered carefully [14]. The problem of planar optimizing is resolved by two kinds of iterative technique in scheduling. Resource-constrained scheduling, one kind of iterative technique, uses the fixed number of operation resource, and achieves the goal that completes the algorithm using the least state number, so as to obtain the optimization result by modifying the constraint requirements making use of iterative technique. Time-constrained scheduling, the other kind of iterative technique, uses the fixed number of state, and achieves

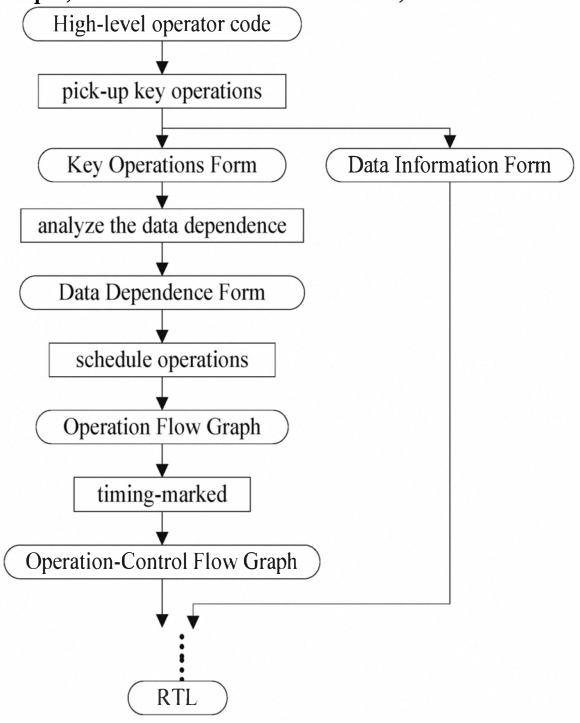


Figure 2. The flow of scheduling

the goal that completing the algorithm using the least operation resource, so as to obtain the optimization result by modifying the constraint requirements making use of iterative technique.

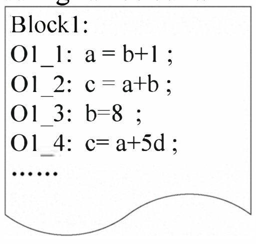
As shown in Fig.2, based on the key operations picked up from the high-level operator code and the data dependency form, all of the operators are rearranged. According to the principles of maximizing parallelization and avoiding data conflict, partial operations of lower level are inserted into the executing cycles of the upper operations from bottom up, constrained by the data (and possibly control) dependencies between the operations, for the sake of achieving the final scheduling result.

3.2 The Strategy of Proposed Scheduling Algorithm

1) Prophases work for scheduling

In order to schedule the operations expediently, the prophases work of scheduling is prerequisite, picking up the key operations and dividing the operator block, which increases the scope for applying parallelizing optimizations in the scheduling phase.

a) Picking up key operations. Ignoring data defining, parameter defining, data in always-block and address data operation, which are transferred to the next level by data information file for transforming into RTL, other operators about data operation should be picked up as the key operations, shown in Fig.3. There are two kinds of operations that can be considered the key operations: integer variable and memory. The information of memory is supplied after rebuilding memory architecture based on the mem and addr information which are collected from the data arrays and pointers in C language code. Finally, a key operations form which is the foundation of scheduling can be built.



```
Block1:
O1_1:  a = b+1 ;
O1_2:  c = a+b ;
O1_3:  b=8  ;
O1_4:  c= a+5d ;
……
```

Figure 3. The key operations form

b) Dividing operations into blocks. According to the keywords, such as decision-else-enddecision, loop-endloop, case-endcase the operations are divided into different blocks, named as "ON_n", shown in Fig.3, of which O is the initial of operation, representing the operation, N points out the block number the operations belongs to, and n indicates the operation number in the block. After picked up, all operations are divided into different blocks based on the different functions, and every block can be scheduled as a whole.

2) Key steps of scheduling

In this section the scheduling algorithm constrained by data dependencies and hardware resource allocations is divided into three steps. A more precise definition of scheduling steps can be given as follows.

operand	operation output	operation input	new address
a	O1_1	O1_2,O1_4	
b	O1_3	O1_1,O1_2	
c	O1_2,O1_4		
d		O1_4	
……	……	……	……

Figure 4. analyze the data dependence

a) Analyzing data dependence. Firstly, analyze the operand in the key operations. As to the integer variable, they can be picked up directly. As for the memory data, they must be transformed based on some rules before picked up. Secondly, analyze the memory configuration and change the address if needed which has a significant impact on both the system performance and the scheduling result. Finally, make the list including the information that which operation the operand is in, placing the operand as output or input, shown in Fig.4.

b) Scheduling operations. To schedule operations, the operations are first analyzed to decide whether they are in serial or parallel order based on two rules. One is the global rule, which schedules the different blocks and decides the executing order of different blocks. The other is the local rule, which schedules the different operations and decides the executing order of different operations according to data dependencies so as to save time and resources, shown in Fig.5.

The secondary scheduling puts the emphasis on the data of the memory, and can also optimize the result of the former scheduling further.

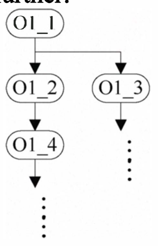


Figure 5. scheduling operations

c)Time-marking. The operation flow graph is already generated after secondary scheduling, and the executive cycle can be calculated conforming to some rules, such as reading data from memory costs two cycles and the

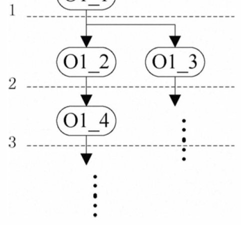


Figure 6. timing-marked

operation of DECISION doesn't cost any cycle. In this step, the operation-control flow graph can be generated, which is applied in transforming into RTL, shown in Fig.6.

4. Experimental Results

For validating the proposed scheduling algorithm, a target algorithm of Inverse Discrete Cosine Transform, which is also a testing pattern for the SPARK tool of University California San Diego [15], is implemented with the operator design method. Based on the results of the high-level operator language, the analysis of data dependency, data storage structure and control structure was carried on, and further scheduling and hardware transformation was performed on the analysis results.

Both implementation results of this paper and SPARK tool were synthesized with SMIC 0.13um CMOS technology with a clock frequency of 100MHz (10ns), 200MHz (5ns), and 250MHz (4ns) respectively and the performance comparison is presented in Table I.

Table 1. Performance Comparison

		This design	SPARK design
Process(nm)		130	130
Area(um^2)	10ns	43151	61763
	5ns	43370	66989
	4ns	53378	71413
Cycle needed dealing with 100 groups of data		24*100=2400	68*100=6800

According to Table I, because of analyzing the data dependence and scheduling the operations after collecting the memory information and rebuilding the memory architecture, the data-processing capacity of our design is much faster than that of the SPARK design. In particular, the design structure generated by the proposed scheduling scheme in this paper obtains a 65% faster data-processing capacity and 30% reduction in hardware cost than that of SPARK tool.

5. Conclusions

In this paper, scheduling to timing optimization in operator design method is proposed and has been applied to implementing the hardware of Inverse Discrete Cosine Transform while obtaining a 65% faster data-processing capacity and 30% reduction in hardware cost than that of SPARK tool of University California San Diego. The implementation proves the practicability and efficiency of the scheduling algorithm, and demonstrates the importance of the scheduling algorithm in saving the data-processing time and reducing the hardware resources.

References

[1] Yangyuan Wang, Yongwen Wang, :China's IC Industry Development - from the country of consumption to the power industry, pp. 241. Science Press, Beijing, 2008. (in Chinese).

[2] Wang F,Sun G Y, Xie Y. A variation aware high level synthesis framework,Proceeding of the 2008

[3] T. J. Kowalski and D. E. Thomas. The VLSI design automation assistant: What's in a knowledge base. In Design Automation Conference, 1985.

[4] C.J. Tseng and D.P. Siewiorek. Automated synthesis of data paths in digital systems. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, July 1986.

[5] P. Marwedel. A new synthesis for the mimola software system. In Design Automation Conference, 1986.

[6] H. Trickey. Flamel: A high-level hardware compiler. IEEE Transactions on Computer–Aided Design, March 1987.

[7] P.G.Paulin and J. P. Knight. Force-Directed Scheduling for the Behavioral Synthesis of ASIC's. IEEE Transactions on CAD, 8(6):661–678, June1989.

[8] I.-C. Park and C.-M. Kyung. Fast and near optimal scheduling in automatic data path synthesis. In Design Automation Conference, 1991.

[9] A.C. Parker, J. Pizarro, and M. Mlinar. MAHA: A program for datapath synthesis. In Design Automation Conference, 1986.

[10] Liangwei Ge, Song Chen, Takeshi Yoshimura, "Automatic Implementation of Arithmetic Functions in High-Level Synthesis".2008 IEEE.

[11] María C. Molina, Rafael Ruiz-Sautua, José M. Mendías, and Román Hermida, "Bitwise Scheduling to Balance the Computational Cost of Behavioral Specifications". IEEE Transactions on Computer–Aided Design OF Integrated Circuits and Systems, VOL.25, NO.1, January 2006.

[12] Dai Peng, Wang Xin'an, Zhang Xing, "A Novel Reconfigurable Operator Based IC Design Methodology for Multimedia Processing". 2009 IEEE.

[13] Ziyi Hu, Jianhong Peng, "Implementation of intra prediction in H.264 based on a novel design methodology". IEEE CSAE 2011.

[14] LIN Younlong. Recent development in high level synthesis [J] . ACM Transactions on Design Automation of Electronic Systems , 1997 , 2 (1) : 2 – 21.

[15] Sumit Gupta, Rajesh K. Kuputa, et al. SPARK: A Parallelizing Approach to the High-Level Synthesis of Digital Circuits, Boston: Kluwer Academic Publishers, 2004:145-171.