IEICE Electronics Express, Vol.2, No.4, 108–114

Three-dimension scheduling under multi-cycle interconnect communications

Shih-Hsu Huang,[a] **Chung-Hsin Chiang, and Chun-Hua Cheng**

Department of Electronic Engineering,

Chung Yuan Christian University, Chung Li, Taiwan, R.O.C.

a) *shhuang@cycu.edu.tw*

Abstract: The three-dimension scheduling is defined as the simultaneous application of clock selection and operation scheduling. Previous three-dimension scheduling approach does not consider the interconnect delay. However, with the advent of nanometer era, the interconnect delay may take multiple clock cycles. In this paper, we use convex programming to formulate the three-dimension scheduling problem under multi-cycle interconnect communications. Benchmark data consistently show that our approach achieves the minimum latency within an acceptable run time.

Keywords: Computer Aided Design, High Level Synthesis, Scheduling, Clock Selection and Interconnect Delay

Classification: Science and engineering for electronics

References

[1] J. Cong and X. Yuan, "Multilevel Global Placement with Retiming," *Proc. ACM/IEEE Design Automat. Conf.*, pp. 644–649, 1997.

[2] P. Chong and R. K. Brayton, "Characterization of Feasible Retimings," *Proc. IEEE Int. Workshop Logic Synthesis*, pp. 1–6, 2001.

[3] J. Cong, Y. Fan, G. Han, X. Yang, and Z. Zhang, "Architecture Synthesis Integrated with Global Placement for Multi-Cycle Communication," *Proc. ACM/IEEE Int. Conf. Comput. -Aided Des.*, pp. 536-543, 2003.

[4] S. Chaudhuri, S. A. Blythe, and R. A. Walker, "A Solution Methodology for Exact Design Space Exploration in a Three Dimensional Design Space," *IEEE Trans. VLSI Syst.*, vol. 5, no. 1, pp. 69–81, 1997.

[5] S. Boyd and L. Vandenberghe, "Convex Optimization," Cambridge University Press, 2004.

[6] C. T. Hwang, J. H. Lee, and Y. C. Hsu, "A Formal Approach to the Scheduling Problem in High Level Synthesis," *IEEE Trans. Computer-Aided Design*, vol. 10, no. 4, pp. 464–475, 1991.

DOI: 10.1587/elex.2.108
Received January 11, 2005
Accepted January 26, 2005
Published February 25, 2005

ELectronics EXpress

IEICE Electronics Express, Vol.2, No.4, 108–114

1 Introduction

There are two important inflection points in the development of nanometer technologies. One is when the average interconnect delay exceeds the gate delay, which happened during mid 1990's and led to the timing closure problem. The other is when single-cycle on-chip communication is no longer possible, which is happening now. Most existing design tools only deal with the first problem but completely lack consideration of multi-cycle interconnect communication. Some efforts [1, 2] have been paid to alleviate the performance degradation caused by global interconnects at the gate level. However, considering multi-cycle interconnect communications at the gate level has a big limitation in the design space exploration.

To further improve the circuit performance, there is a demand to consider multi-cycle communication in high-level synthesis. The RDR (regular distributed register) architecture [3] provides a regular synthesis platform for estimating the interconnect delay. Cong, Fan, Han, Yang and Zhang [3] developed a high-level synthesis system, called MCAS, for the RDR architecture. However, the MCAS system has the following two limitations. First, they restrict that each operation takes only single clock cycle. Secondly, their scheduling algorithm is based on the force-directed scheduling framework, which assumes a fixed clock length.

For a more useful and exhaustive design space exploration, the clock length should be considered as a variable. In fact, the combination of clock selection and operation scheduling has been referred to as the three-dimension scheduling problem [4]. However, previous approach [4] does not consider the interconnect delay. In this paper, we use convex programming [5] to formulate the interconnect delay into the three-dimension scheduling problem. Therefore, our approach not only allows the operations to take multiple clock cycles, but also allows the interconnect communications take multiple clock cycles. Given a prior binding and the upper bound of number of control steps, our approach guarantees minimizing the overall latency, which includes the interconnect delay. Benchmark data consistently show that our approach works well in practice.

2 Motivation

We use the data flow graph shown in Fig. 1 to illustrate our motivation. Assume that this data flow graph is executed in two-island RDR architecture: the island I_1 is a multiplier, whose computation delay is 163 time units (tu); the island I_2 is an ALU, whose computation delay is 56 tu. For the convenience of the readers, in Fig. 1, we draw the inter-islands data transfer in bold. Note that the RDR architecture facilitates the estimation of interconnect delay. Here we assume that the local interconnect delay (i.e., intra-island data transfer) is 5 tu and the global interconnect delay (i.e., inter-islands data transfer) is 180 tu.

The selection of clock length affects the latency. We give two design points in the design space exploration as below.

DOI: 10.1587/elex.2.108
Received January 11, 2005
Accepted January 26, 2005
Published February 25, 2005

(1) If we use a clock length of 86 tu, the minimum number of control steps is 6. Fig. 1 (a) gives such a design point. As a result, the minimum latency is 516 tu.

(2) If we use a clock length of 180 tu, the minimum number of control steps is 4. Fig. 1 (b) gives such a design point. As a result, the minimum latency is 720 tu.

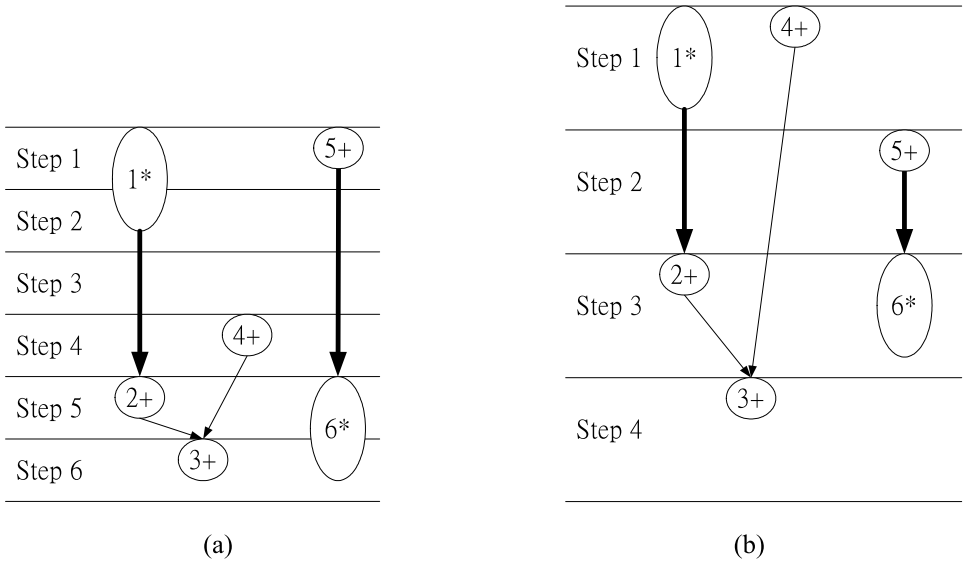


Fig. 1. A motivational example.

3 The Formulations

We use convex programming to formulate the three-dimension scheduling problem that allows multi-cycle interconnect communications. Note that, as described in [6], there are polynomial time algorithms to solve convex programming formulations.

In our convex programming formulations, we use the notation $x_{i,j,s}$ to denote a binary variable (i.e., an 0-1 integer variable). Binary variable $x_{i,j,s} = 1$, if and only if operation o_i is scheduled into control step j and take s control steps; otherwise, binary variable $x_{i,j,s} = 0$. Clearly, we have $1 \leq i \leq n$, $1 \leq j \leq t$ and $1 \leq s \leq t$, where n is the number of operations in the data flow graph and t is the upper bound of number of control steps. Thus, intuitively, the total number of binary variables is $n \cdot t^2$. However, in fact, from the ASAP (as soon as possible) calculation and ALAP (as late as possible) calculation [6], we can find that a lot of binary variables are redundant since their values are definitely 0. Therefore, we can prune these redundant binary variables without scarifying the exactness of the solution.

The constants used in our convex programming formulations are as below.

■ The value n denotes the number of operations in the data flow graph.

DOI: 10.1587/elex.2.108
Received January 11, 2005
Accepted January 26, 2005
Published February 25, 2005

- The delay of each operation o_i is D_i.
- The interconnect delay of operation o_i to operation o_k is $N_{i,k}$.
- The value E_i denotes the earliest possible control step of operation o_i. Note that, we can use the ASAP calculation to determine the value E_i for each operation o_i.
- The value L_i denotes the latest possible control step of operation o_i. Note that, given the upper bound of number of control steps, we can use the ALAP calculation to determine the value L_i for each operation o_i.
- We use I_k to denote an island, and we say that $o_i \in I_k$ if and only if operation o_i is assigned to executed by the island I_k.

Let the variable $Cycle$ denote the clock length and the variable C_{step} denote the number of control steps. The three-dimension scheduling problem under multi-cycle interconnect communications can be formulated as the following convex programming formulations.

$$\text{Minimize} \quad Cycle \cdot C_{step} \tag{Formula 1}$$

Subject to
For each operation o_i

$$\sum_{j=E_i}^{L_i} \sum_{s=1}^{L_i-j+1} x_{i,j,s} = 1 \tag{Formula 2}$$

$$\sum_{j=E_i}^{L_i} \sum_{s=1}^{L_i-j+1} s \cdot x_{i,j,s} = \left\lceil \frac{D_i}{Cycle} \right\rceil \tag{Formula 3}$$

For each dependency relation $o_i \rightarrow o_k$

$$\sum_{j=E_i}^{L_i} \sum_{s=1}^{L_i-j+1} (j \cdot x_{i,j,s}) + \frac{D_i + N_{i,k}}{Cycle} \leq \sum_{j=E_k}^{L_k} \sum_{s=1}^{L_k-j+1} (j \cdot x_{k,j,s}) \tag{Formula 4}$$

For each control step $c$and each island I_k

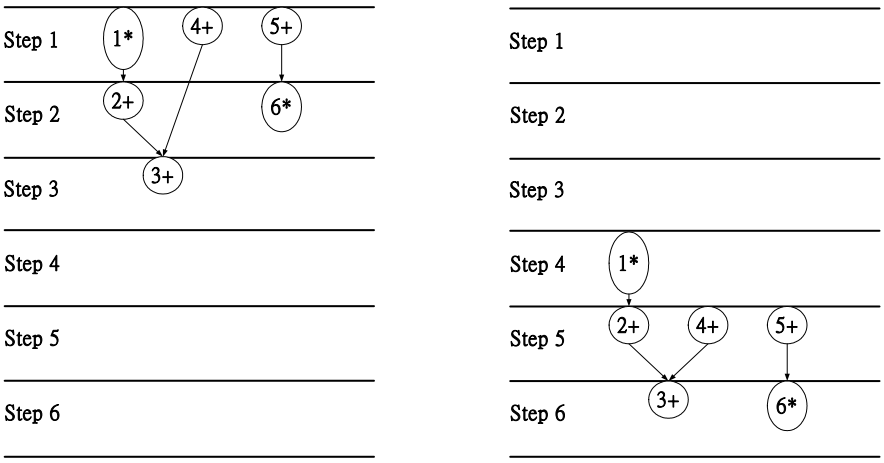
$$\sum_{o_i \in I_k} \sum_{j=E_i}^{L_i > c ? c : L_i} \sum_{s=c-j+1}^{L_i-j+1} x_{i,j,s} \leq 1 \tag{Formula 5}$$

For each operation o_i without successors

$$\sum_{j=E_i}^{L_i} \sum_{s=1}^{L_i-j+1} (j \cdot x_{i,j,s}) + \left(\frac{D_i}{Cycle} - 1 \right) \leq C_{step} \tag{Formula 6}$$

Formula 1 defines the objective function. Formula 2 states the constraint that every operation must be scheduled to a control step. Formula 3 derives the number of control steps that each operation is required to take. Formula 4 ensures that the data dependency relationships are preserved. Formula 5

DOI: 10.1587/elex.2.108
Received January 11, 2005
Accepted January 26, 2005
Published February 25, 2005



Operation	Associated Binary Variables
o_1	$x_{1,1,1}$,$x_{1,1,2}$,$x_{1,1,3}$,$x_{1,1,4}$,$x_{1,2,1}$,$x_{1,2,2}$,$x_{1,2,3}$,$x_{1,3,1}$,$x_{1,3,2}$, $x_{1,4,1}$
o_2	$x_{2,2,1}$,$x_{2,2,2}$,$x_{2,2,3}$,$x_{2,2,4}$,$x_{2,3,1}$,$x_{2,3,2}$,$x_{2,3,3}$,$x_{2,4,1}$,$x_{2,4,2}$, $x_{2,5,1}$
o_3	$x_{3,3,1}$,$x_{3,3,2}$,$x_{3,3,3}$,$x_{3,3,4}$,$x_{3,4,1}$,$x_{3,4,2}$,$x_{3,4,3}$,$x_{3,5,1}$,$x_{3,5,2}$, $x_{3,6,1}$
o_4	$x_{4,1,1}$,$x_{4,1,2}$,$x_{4,1,3}$,$x_{4,1,4}$,$x_{4,1,5}$,$x_{4,2,1}$,$x_{4,2,2}$,$x_{4,2,3}$,$x_{4,2,4}$,$x_{4,3,1}$, $x_{4,3,2}$, $x_{4,3,3}$, $x_{4,4,1}$, $x_{4,4,2}$, $x_{4,5,1}$
o_5	$x_{5,1,1}$,$x_{5,1,2}$,$x_{5,1,3}$,$x_{5,1,4}$,$x_{5,1,5}$,$x_{5,2,1}$,$x_{5,2,2}$,$x_{5,2,3}$,$x_{5,2,4}$,$x_{5,3,1}$, $x_{5,3,2}$, $x_{5,3,3}$, $x_{5,4,1}$, $x_{5,4,2}$, $x_{5,5,1}$
o_6	$x_{6,2,1}$,$x_{6,2,2}$,$x_{6,2,3}$,$x_{6,2,4}$,$x_{6,2,5}$,$x_{6,3,1}$,$x_{6,3,2}$,$x_{6,3,3}$,$x_{6,3,4}$,$x_{6,4,1}$, $x_{6,4,2}$, $x_{6,4,3}$, $x_{6,5,1}$, $x_{6,5,2}$, $x_{6,6,1}$

(c)

Fig. 2. (a) ASAP schedule (b) ALAP schedule (c) Associated binary variables.

states the constraint that each island at most executes one operation in any control step. Formula 6 states the constraint that each operation o_i must be scheduled not later than the control step C_{step}.

We use the data flow graph given in Fig. 1 to illustrate the convex programming formulations. Assume that the upper bound of control steps is 6. In the ASAP calculation and ALAP calculation, we assume that each operation takes only one control step. Fig. 2 (a) and Fig. 2 (b) give the ASAP calculation and the ALAP calculation, respectively. According to the ASAP and ALAP calculations, we can prune all the redundant binary variables. Fig. 2 (c) gives all the necessary (i.e., irredundant) binary variables associated with each operation. In the following, for each formula, we give an example to explain.

Formula 2. Using operation o_2 as an example, there is exactly one binary variable is true among all the ten binary variables associated with operation o_2. Thus, we have $x_{2,2,1} + x_{2,2,2} + x_{2,2,3} + x_{2,2,4} + x_{2,3,1} + x_{2,3,2} + x_{2,3,3} + x_{2,4,1} + x_{2,4,2} + x_{2,5,1} = 1$.

Formula 3. Using operation o_2 as an example, since the computation delay of island I_2 (i.e., the delay of an ALU operation) is 56 tu, we can use $\lceil 56/Cycle \rceil$ to determine the number of control steps that operation o_2 takes. Thus, we have $x_{2,2,1} + 2x_{2,2,2} + 3x_{2,2,3} + 4x_{2,2,4} + x_{2,3,1} + 2x_{2,3,2} + 3x_{2,3,3} + x_{2,4,1} + 2x_{2,4,2} + x_{2,5,1} = \lceil 56/Cycle \rceil$.

Formula 4. Using the data dependency relation of $o_1 \rightarrow o_2$ as an example, operation o_2 can be executed if and only if operation o_1 has completed its

DOI: 10.1587/elex.2.108
Received January 11, 2005
Accepted January 26, 2005
Published February 25, 2005

execution. If operation o_1 is schedule into control step 1, the control step that operation o_2 can start the execution should be greater than the value $1 + (D_1 + N_{1,2})/Cycle = 1 + (163 + 180)/Cycle$. Thus, we have $x_{1,1,1} + x_{1,1,2} + x_{1,1,3} + x_{1,1,4} + 2x_{1,2,1} + 2x_{1,2,2} + 2x_{1,2,3} + 3x_{1,3,1} + 3x_{1,3,2} + 4x_{1,4,1} + (343/Cycle) \leq 2x_{2,2,1} + 2x_{2,2,2} + 2x_{2,2,3} + 2x_{2,2,4} + 3_{x2,3,1} + 3x_{2,3,2} + 3x_{2,3,3} + 4x_{2,4,1} + 4x_{2,4,2} + 5x_{2,5,1}$.

Formula 5. Consider that there are two ALU operations o_4 and o_5 can be scheduled into control step 2. However, at control step 2, the number of ALU operations that can be executed in the island I_2 is at most 1. Thus, we have $x_{4,1,1} + x_{4,1,2} + x_{4,1,3} + x_{4,1,4} + x_{4,1,5} + x_{5,1,1} + x_{5,1,2} + x_{5,1,3} + x_{5,1,4} + x_{5,1,5} \leq 1$.

Formula 6. Using operation o_3 as an example, operation o_3 must be completed its execution not later then the control step C_{step}. Thus, we have $3x_{3,3,1} + 3x_{3,3,2} + 3x_{3,3,3} + 3x_{3,3,4} + 4x_{3,4,1} + 4x_{3,4,2} + 4x_{3,4,3} + 5x_{3,5,1} + 5x_{3,5,2} + 6x_{3,6,1} + (56/Cycle) - 1 \leq C_{step}$.

After solving the convex programming formulations, we have that $x_{1,1,2} = x_{2,5,1} = x_{3,6,1} = x_{4,4,1} = x_{5,1,1} = x_{6,5,2} = 1$ and the values of other binary variables are 0. The clock length $Cycle$ is 86 tu. The number of control steps C_{step} is 6. The overall latency is $86 * 6 = 516$ tu. The corresponding schedule is shown in Fig. 1 (a).

4 Experimental Results

We use the Extended LINGO Release 8.0 to solve the convex programming formulations on a personal computer with P4-2.4 GHz CPU and 512 MBytes RAM. Five benchmark circuits, including AR filter, HAL, Bandpass filter, FIR filter and Elliptic filter, are used to test the effectiveness of our approach. In our experiments, our synthesis platform is the same as the two-island RDR architecture described in Section 2.

Table I gives the characteristics of benchmark circuits and tabulates our experimental results. We assume that the upper bound of number of control steps is 30. The column *mul_op* denotes the number of multiplication operations. The column *alu_op* denotes the number of ALU operations. The column d*ependency* denotes the number of dependency relations. The column *Cycle* denotes the clock length. The column C_{step} denotes the number of control steps. The column *Latency* denotes the latency. The column *run time* denotes the CPU time in seconds that the Extended LINGO Release

Table I. Experimental results on benchmark circuits.

Circuit	Circuit Characteristics			Synthesis Results			
	mul_op	*alu_op*	d*ependency*	*Cycle* (tu)	C_{step}	*Latency* (tu)	*run time* (seconds)
AR	16	12	30	172	20	3440	3585
HAL	6	5	8	58	22	1276	531
Bandpass	11	18	30	82	26	2132	3531
FIR	7	6	12	58	25	1450	1940
Elliptic	8	26	47	118	30	3540	1158

DOI: 10.1587/elex.2.108
Received January 11, 2005
Accepted January 26, 2005
Published February 25, 2005

8.0 used to solve the convex programming formulations. Experimental data consistently show that our approach achieves the minimum latency within an acceptable run time.

5 Conclusions

This paper investigates the problem of three-dimension scheduling, i.e., the combination of clock selection and operation scheduling. A convex programming approach is proposed to minimize the overall latency. Compared with previous work [4], our approach is distinctive in that it allows multi-cycle interconnect communications. Experiments with benchmark circuits consistently show that our approach achieves the minimum latency within an acceptable run time.

6 Acknowledgments

This work was supported in part by the National Science Council of R.O.C. under contract number NSC 92-2220-E-033-001.

DOI: 10.1587/elex.2.108
Received January 11, 2005
Accepted January 26, 2005
Published February 25, 2005