

Clock Cycle Selection Considering Interconnection Delay in High Level Synthesis*

H. Mecha, M. Fernández, R. Hermida

Dpto. Informática y Automática, Univ. Complutense de Madrid

E-mail: hortem@eucmax.sim.ucm.es Fax 34-1-394-46-87 Tel. 34-1-394-44-69

Abstract

This paper presents a method to select the clock cycle in a High Level Synthesis process. The main novelty is that it takes into account the interconnection delay. This is important to settle securely the right electrical behavior of the final circuit. It uses a simple model based on point to point interconnections. The selection algorithm considers the dead times of the operators, that influence the total execution time, and the number of control steps, that settles the control unit size.

1: Introduction

In High Level Synthesis (HLS) the selection of the clock cycle is very important because it influences the overall execution time (ET) and the size of the Control Unit (CU) of a design. Scheduling and Hardware Allocation (HA) depend on the clock cycle, so it is important to spend some time trying to obtain a good value, in order to drive the design process to a good solution. If the design constraints are not met because the clock cycle was not properly selected, it is necessary to try a new value, to schedule again the graph and to allocate the hardware. This may take too much time.

In this selection it is very important to pay attention to the interconnection delay. In technologies around one micron, the interconnections capacitances come to the same magnitude as transistors capacitances, and therefore, the signal propagation delay becomes an important factor in the total ET of any operation.

This paper presents a method to find the clock period considering this factor, taking into account only point to point interconnections. The presented technique leaves space for improvements, such as consideration of multimodule interconnections. A basic design model, including all the elements of the data-path (DP) and the signal propagation delay, is presented.

2: Previous work

Clock cycle computation is an important task often neglected in HLS literature. The usual approach is to use a fixed clock cycle specified by the user [1], without considering

*Supported by Spanish Government grant TIC 96/1071 and European Community program HCM CHRX-CT94-0459

its influence over the area and performance of the final circuit. Most systems [2], [3], [4] use techniques for estimating clock cycles that only examine the ET. For example Gajski presents in [4] the Clock Slack Minimization Algorithm to calculate a clock cycle which would result in the maximum utilization of the Functional Units (FUs), but the CU can become unacceptably large. Moreover, the interconnection delay is not considered, so the results are not realistic.

In [5] the selection is performed for a Data Flow Graph (DFG) scheduled and allocated, but once again, the interconnection delay is not considered. Also, the new selection may need a new scheduling and then a new HA, so the selected clock cycle may not be optimal.

Only Weng [6] takes into account interconnection delay, but only when the floorplanning has been done, in order to modify the design when the time constraints have not been met. This can lead to circuits that do not match users constraints. Our approach tries to solve these problems calculating a cycle time that leads to an optimal performance without increasing the CU area, and bearing in mind the interconnection delay. In this way it is possible to guarantee the correct timing simulation in the design process.

In section 3 the design model is presented. Next, the interconnection delay estimation for an interconnection with a length L is explained. Section 5 shows a method to include this estimation in a HLS process. Finally, one example and the conclusions are presented.

3: Design model

The design model used by our system is shown in Figure 1, where only point to point interconnections are allowed. One level of multiplexers is assumed at the register inputs and another one at the FUs inputs. In this model, we consider multiplexers implemented with two levels of gates, so their delay is approximately constant and independent on the number of inputs. A typical operation involves operands being read from registers, an operation performed on them, and the result stored in a register. The operation delay can go through several control steps. This delay is computed as:

$$delay = t_{oper} + t_{store} + t_{prop} + 2 * t_{mux}$$

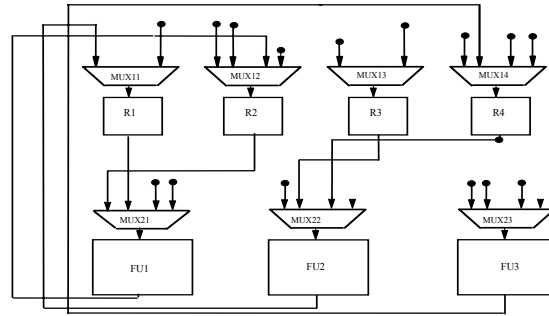


Figure 1. Design Model

t_{oper} is the delay of the operator and depends on the FU that will implement it; t_{mux} is the delay of a multiplexer and t_{store} the delay of the register. These three delays depend on the library and they are known. Also, t_{prop} represents the signal propagation, and it depends on four delays: two of them from the multiplexers outputs to the registers (t_{mux_REG}) or FUs (t_{mux_FU}); the other two from the registers (t_{REG_FU}) or FUs

(t_{FU_REG}) outputs to FUs or registers respectively, through multiplexers if necessary. These delays are unknown and the methods to estimate them are explained next.

4: Interconnection delay estimation

In technologies around 1μ interconnection delay is comparable to functional delay [6], so it is important to estimate it before calculating the cycle time. In order to estimate the interconnection delay, we need to find a physical model to represent the interconnection. As a basis for our study we start with a simple model where interconnections are only point to point. Consequently, an interconnection is a line with one end point driven by a MOSFET and the other connected to the gate of another MOSFET. For current technologies, the resistance and capacitance of the line can not be considered zero. Although there are some different models to represent it, the best one matching the interconnection behavior is Π_3 , that permits to estimate the delay with less than 3% error [7]. This model is represented in figure 2, where R and C are the total resistance and capacitance of the wire, C_t is the capacitance of the load transistor and R_t is the equivalent resistance of the driving transistor. If the width for an interconnection is fixed, and r and c are the resistance and capacitance of a wire per length unit then: $R=r*L$ and $C=c*L$, where L is the interconnection length.

To calculate the interconnection delay we use $t_{0.9}$, that is the time delay from the moment a step voltage V_i is applied to the source point (1) until V_o at node (2) reaches the value $0.9V_i$. To compute $t_{0.9}$ it is necessary to study the circuit frequency response [8]. The transfer function for this model is:

$$H(s) = \frac{V_o(s)}{V_i(s)} = \frac{F}{As^4 + Bs^3 + Ds^2 + Es + F}$$

where A , B , D , E and F are functions of C , R , R_t and C_t , and s is the complex frequency.

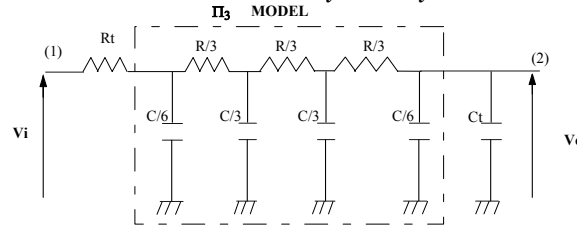


Figure 2. Π_3 Model

The transfer function denominator is a fourth order polynomial in s , so it has four roots, the poles of $H(s)$. Their values depend on the relationship $C_t * R_t / C * R$, and therefore on the interconnection length and on the technology. The time domain response of the circuit depends on the poles position. There is a **dominant pole** whenever a pole is much lower than the others, that are at least a decade away. It can be proved [8] that the dominant pole determines the time domain response, and therefore the system can be approximated by a first order system.

It is interesting to know the variation of the two most significant poles, as a function of the relation $C_t * R_t / C * R$. Taking into account this goal, we have performed a study based on typical values of C_t and R_t from ES2 Library [9], a standard cells library for CADENCE, and a realistic interconnection length range. We have taken as typical values for 1μ technology, $R_t=1000\Omega$, $C_t=0.05\text{pF}$, $r=0.04\Omega/\mu$ and $c=0.09\text{fF}/\mu$. In this study L ranges from 10000μ to 1μ , that are the values of the interconnection lengths for typical

SAN examples. It has been obtained that in all cases the dominant pole frequency is at least 25 times lower than the other poles, so it is possible to model the system as a first order one to estimate $t_{0.9}$. In a first order system, the response in terms of time is: $V_o(t) = V_i(1 - e^{-\sigma t})$ where σ is the dominant pole. To compute $t_{0.9}$ we take $V_o(t)$ as $0.9V_i$, and so $t_{0.9} = \ln(10)/\sigma$.

The computation of the poles in a fourth order system and, therefore, its transformation into a first order system, are of constant complexity. Besides, the transmission line delay estimation is also of constant complexity, when R , C , R_t and C_t are known. This solution is valid for the range of L from 10000μ to 1μ . Nevertheless, for longer interconnections, the relationship between the two most significant poles could be lower, and then this method would not be suitable. However, these longer interconnections have excessive delays, and, consequently, the clock cycle of the circuit should be very long in order to settle securely on a right electrical behavior. Usually, the designer uses different methods to avoid these long interconnection delays, for example making wider interconnections to decrease the resistance, or using buffers that accelerate the signal transmission [10]. Both solutions lead to interconnection lengths where the dominant pole method can be applied.

But the interconnection delay depends on its length, and so, it is necessary to calculate the value of L we should use to compute the signal propagation delay for all the operations in the DFG. As each interconnection length is different, in next section we will estimate a value to settle securely the right circuit electrical behavior.

5: Interconnection delay estimation in HLS

The values for C_t and R_t only depend on the design technology, but R and C also depend on the interconnection length. In section 3 we showed that signal propagation in the execution of an operation involves four delays. Two of these delays are transmissions from multiplexers to FUs or registers (t_{mux_FU} and t_{mux_REG}). Studying the results of the placement and routing algorithms, we can see that, in many cases, the multiplexers are placed very closely to the FUs or to the registers they are connected to. The length of these interconnections (that we have called **close** interconnections in [11]), L_{close} is estimated as the average standard cell length, that is a known value.

The other two kinds of interconnections, t_{REG_FU} and t_{FU_REG} can belong to any of the two types of point to point interconnections, **close** or **distant** in the terminology of [11]. We can not know the real interconnection length until placement and routing have taken place, so we need to perform an estimation. A possible solution to secure the right electrical behavior of the circuit, is performing an estimation of the **worst case** interconnection length. This maximum length, L_{max} , sets the maximum delay of an interconnection. It is equal to the half perimeter of the circuit, and it depends on the total circuit area by: $L_{max} = 2\sqrt{Area}$. Therefore, the interconnection delay for these two kinds of interconnections is estimated as the delay of an interconnection L_{max} , and then: $t_{prop} = 2 * t(L_{max}) + 2 * t(L_{close})$

But L_{max} depends on the circuit area, that is not known until HA has taken place, and the interconnection delay should be estimated before scheduling, in order to select a right clock cycle, so we need to estimate the total circuit area. However, any estimation of the circuit area before scheduling can not give acceptable results because the information of

the circuit is incomplete. We propose another solution to solve this problem, that has been included in our HLS system [12]:

A.- Estimating the signal propagation delay t_{prop} . An initial clock cycle is estimated without considering the interconnection delay, and using the algorithm presented in [13], that considers the DFG, the module library and the user constraints. Then, a list scheduling is performed, using for implementing each operation the fastest module in the library, and the maximum area for this scheduling is estimated. With this value, we estimate the maximum value of the half perimeter of the circuit $L_{maxList_Sche}$, and the signal propagation delay is calculated with: $t_{prop}=2*t_{0,g}(L_{maxList_Sche})+2*t_{0,g}(L_{close})$.

B.- Estimating the clock cycle, scheduling and HA. In this step the clock cycle is selected using [13], but now taking into account the interconnection delay estimated before. Then a new scheduling and a HA are performed, and the final area of the circuit, A_{final} , is calculated using [11] with an error lower than 5%.

C.- Calculating t_{prop} and selecting the final clock cycle. Using A_{final} , it is possible to calculate the half perimeter of the circuit L_{final} , and to obtain the new signal propagation delay t_{prop_new} : $t_{prop_new}=2*t_{0,g}(L_{final})+2*t_{0,g}(L_{close})$

This value can be used to generate t_{dif} that is the difference between the value estimated in phase A and the new value $t_{dif}=t_{prop}-t_{prop_new}$. It is possible to decrease the clock cycle estimated in phase B in a value equal to t_{dif} divided by the maximum number of steps that any FU in the circuit needs to execute an operation. In this way, the scheduling and HA performed before are valid, though the clock cycle changes.

On the other hand, we consider that the control unit works in parallel with the data-path, and so it is only necessary to take into account the delay of the control signals. This implies the addition of the delay of one interconnection to the clock cycle. With these assumptions it can be guaranteed the correct time simulation of the circuit.

6: Example

To show the goodness of this algorithm we present the results for the Fifth Order Elliptical Filter [14]. The module library is presented in table I. In this example we will assume that area minimization is more important than time minimization. In order to select the clock cycle considering interconnection delay, we follow the phases presented in section 5.

module	op	area	delay
adder	+	0,054mm ²	30ns
adder	+	0,21mm ²	24ns
multiplier	*	2,3mm ²	138ns

Table I Module Library

A.- Estimating signal propagation delay t_{prop} . According to [13], we have to introduce a parameter α , that weights time priority versus area priority. As area minimization is more important than ET minimization, we choose $\alpha=0.3$. For this value, the module library presented in table I and the algorithm presented in [13], we obtain a clock cycle of 69ns. After list scheduling and allocation we obtain a total area equal to

8,6 mm², and $L_{maxList_Sche} = 5,858\text{mm}$, with an associated delay equal to 1,412ns. The delay of a short interconnection for 1 μ technology is $t_{0,9}(L_{close}) = 0,127\text{ns}$. Then:

$$t_{prop} = 2 * 1,412\text{ns} + 2 * 0,127\text{ns} = 4\text{ns}$$

B.- Estimating the clock cycle, scheduling and HA. The new clock cycle obtained for this library considering the signal propagation delay is: $t_{cycle} = 71\text{ns}$.

C.- Calculating t_{prop} and selecting the final clock cycle. After HA we obtain $A_{final} = 4,6\text{mm}^2$, so $L_{final} = 4,294\text{ mm}$. The delay associated to this length interconnection is $t_{0,9}(L_{final}) = 1,051\text{ns}$, and $t_{prop_new} = 2,356\text{ns}$. Consequently $t_{dif} = 4\text{ns} - 2,356\text{ns} = 1,644\text{ns}$. The maximum number of steps for any FU in the data-path is 2, so it is possible to decrease the clock cycle in $t_{dif}/2 = 0,822\text{ns}$. This value is lower than 1, so the new clock cycle would be: $t_{cycle_new} = t_{cycle}$. In order to take into account the control signal delay, we should add $2,356\text{ns} \approx 3\text{ns}$ to the final clock cycle.

7: Conclusions

In this paper a new method to estimate the clock cycle considering interconnection delay has been presented. The model used is a basic one, with only point to point interconnections, and a fixed delay for the multiplexers. The interconnection delay has been estimated using the worst case length, so it is possible to guarantee the right electrical behavior. Our current work is related with the treatment of a more realistic model, with multimodule interconnection and a set of different hierarchical multiplexers covering all possible number of inputs.

8: References

- [1] C. Wang, K.K.Parhi, "High Level DSP Synthesis Using Concurrent Transformations, Scheduling and Allocation" IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 14, n° 3, pp. 274-295, March 1995
- [2] A.C. Parker, T. Pizzaro and M. Mlinar, "MAHA: A Program for Datapath Synthesis", in Proc. of the Designs Automation Conference, pp 461-466, 1986.
- [3] R. Jain, M.J.Mlinar, A. Parker, N. Park "Area Time Model for Synthesis of Non-pipelined Designs" Proc. IEEE International Conference on Computer-Aided Design, ICCAD-88, Nov. 1988, pp. 48-51
- [4] S. Narayan, D. Gajski "System Clock Estimation based on Clock Slack Minimization" EuroVHDL 92 Hamburg Sep 1992, pp 66-71
- [5] S. Parameswaran, P. Jha, N. Dutt "Resynthesizing controllers for minimum execution time" Proc. Second Asian Pacific Conference on Hardware Description Languages, APCHDL 94, pp. 111-117, Japan.
- [6] J. Weng, A. Parker, "3D Scheduling: High Level Synthesis with Floorplanning", 28th ACM/IEEE Design Automation Conference, 1991, pp. 668-673
- [7] T. Sakurai "Approximation of Wiring Delay in MOSFET LSI" IEEE Journal of Solid-State Circuits, Vol. SC-18, n° 4, pp. 418-426, Aug. 1983
- [8] B. C. Kuo "Automatic Control Systems" Prentice-Hall International Editions
- [9] ES2 ECPD10 Library Databook, 1993
- [10] N.H.E. Weste, K. Eshraghian "Principles of CMOS VLSI Design" 2nd Edition, cap. 4, 1994
- [11] H. Mecha, M. Fernández, F. Tirado, J. Septien, D. Mozos, K. Olcoz "A Method for Area Estimation of VLSI Integrated Circuits in High Level Synthesis", IEEE Trans. on CAD of Integrated Circuits and Systems, vol. 15, n°2, pp. 258-265, Feb. 1996.
- [12] J. Septién, D. Mozos, F. Tirado, R. Hermida, M. Fernández, H. Mecha "FIDIAS: an Integral Approach to High Level Synthesis" IEE Proceedings Circuits, Devices and Systems, vol. 142, n°4, pp. 227-235, Agosto 1995

- [13] H. Mecha, M. Fernández, R. Hermida, D. Mozos, K. Olcoz "Clock Cycle Estimation Based on Dead Time and Control Unit Area Minimization" Microprocessing and Microprogramming, vol. 40, Ed. North-Holland, pp. 821-824., Sept. 1994, Holland.
- [14] High Level Synthesis Benchmarks, 1991