International Journal of Scientific Engineering and Research (IJSER)
ISSN (Online): 2347-3878
Index Copernicus Value (2015): 62.86 | Impact Factor (2015): 3.791

Crosstalk Minimization in SOC using Genetic Algorithm

Shilpa D.R[1], Uma B.V [2]

[1]RVCE, Bangalore-560059, India
[2]RVCE, Bangalore-560059, India

Abstract: ***The era of high speed and high density designs has led the design engineers various challenges. As the technology is scaling, one of the major concerns of VLSI design is signal integrity. Crosstalk is the major cause of signal integrity, occurs due to the coupling between adjacent interconnects. Over the years many layout level crosstalk optimization techniques such as shielding, buffer insertion, and encoding-decoding schemes have been proposed. But addressing this problem at high level of design abstraction is also advisable. This work proposes high level synthesis procedures to eliminate capacitive crosstalk based on genetic algorithm. Crosstalk aware simultaneous scheduling, binding and resource allocation is performed to reduce coupling parameter. Coupling parameter is estimated by having the knowledge of data transition patterns and their correlations. The proposed genetic algorithm with ASAP and ALAP schedules as the initial solutions has reduced crosstalk by 12.2241% for 3x3 matrix multiplication.***

Keywords: Genetic Algorithm, Miller Capacitance Factor, High level Synthesis, Crosstalk.

1. Introduction

As predicted by Moore, the transistor density on the silicon chip is increasing. As expected by ITRS (International Roadmap for Semiconductors) 2013, the transistor feature size will be reduced to 5 nm by 2022[1]. This leads the interconnects to be packed very close to each other, thus increasing the coupling capacitance between them. When one of the interconnects switches it affects the data in its neighboring interconnects. This phenomenon is called Crosstalk. Increasing spacing between adjacent lines, using shield wires, ensuring the adjacent lines to switch at different times[2], bus encoding[4], buffer insertion[5] etc. are some of the methods to reduce crosstalk at physical design level. But these methods lack in flexibility as layout level modifications are cumbersome. This work proposes high level synthesis procedures for crosstalk minimization using genetic algorithm. Addressing the crosstalk problem at high level yields more efficient results than addressing it at lower (layout) level. The three important processes at high level synthesis are Scheduling, Allocation and Binding.

Genetic algorithm is one of the probabilistic search algorithms to find the optimal solution for a given problem. Starting with the initial population, GA tries to improve the fitness of the population iteratively by application of crossover and mutation. Individuals with better fitness are selected for crossover in next generation than other individuals. Iteration is carried out until certain criterion is met or for certain number of generations[9]. Authors[9] have proposed simultaneous scheduling, allocation and binding using problem space genetic algorithm (PSGA) to generate area and power optimized designs. Scheduling for a multiprocessing task to improve the time and processor utilization is proposed based on coarse grain GA[10]. Floor planning application for SoC/IP using GA is presented by Phuong and Thanh[11]. The proposed work achieves crosstalk minimization in SoCs employing Genetic algorithm in HLS procedures.

The organization of this paper is as follows. The section 1.1 explains about crosstalk and Miller Coupling Factor. Section 1.2 discusses about High Level Synthesis and section 2 describes Genetic algorithm. The next section describes the algorithm used which is followed by results, conclusion and references.

1.1 Crosstalk

Crosstalk can be better explained by aggressor victim model. Aggressor is the interconnect that influences a change in its neighbour, the victim. Dual Aggressor victim model as shown in Figure 1 is a variant where two aggressors influence the victim interconnect. If an 8 bit bus is used for data transfer, then crosstalk on bit B0 and B7 is calculated using single aggressor model while dual aggressor victim model is used for bits B1 to B6.

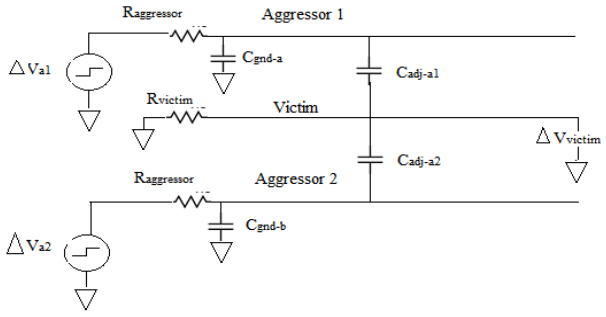

Figure 1: Dual Aggressor-Victim model

The number of coupling capacitances in the circuit, Miller Coupling Factor (MCF) is a direct estimate of crosstalk in the circuit. MCF can be calculated using the equation (1).

$$Ceff = Cg + \frac{Cc(\Delta V - \Delta A1)}{E} + \frac{Cc(\Delta V - \Delta A2)}{E} \quad (1)$$

Where Cg is the capacitance between interconnect and ground, Cc is the coupling capacitance, ΔV is the change in voltage of the victim, $\Delta A1$ and $\Delta A2$ are the change voltages of the two aggressors respectively and E is the supply

voltage. MCF for various transitions in victim and aggressors is mentioned in Table 1. This highlights that the maximum crosstalk occurs when the aggressors switch in opposite direction to victim and no crosstalk condition is considered when aggressors switch in the same direction as the victim.

Crosstalk induced in the interconnect can cause it to switch faster or slower depending on the relative variation of it and it's neighbors. This induces a delay in the interconnect which may lead to wrong functional evaluation. Further, when the victim remains constant and the aggressor switches, a noise/delay is forced on the victim. Therefore, functionality of the circuit should not be compromised and hence crosstalk reduction plays an important role.

Table 1: Coupling in the victim by dual aggressor model

Coupling Transition Patterns (A2,V,A1)	Coupling
(↑,↑,↑) (↓,↓,↓)	No Coupling
(-,↑,↑) (-,↓,↓) (↑,↑,-) (↓,↑,-)	C_c
(-,↑,-) (-,↓,-) (↑,↑,↓) (↑,↓,↓) (↓,↑,↑) (↓,↓,↑)	$2C_c$
(-,↑,↓) (-,↓,↑) (↑,↓,-) (↓,↑,-)	$3C_c$
(↑,↓,↑) (↓,↑,↓)	$4C_c$

1.2 High Level Synthesis

High level synthesis converts the behavioural specification of a design usually represented as a CDFG along with design constraints to a register transfer level netlist which is synthesized to the gate level using a logic synthesis tool. The HLS engine has to perform tasks such as compilation of the behavioural specification, scheduling the operations, binding, allocation and data path and controller generation. Scheduling is the process of determining the control step for each operation in a data flow graph. There exist many methods for scheduling a given DFG such as As Soon As Possible (ASAP), As Late As Possible (ALAP), Force Directed Scheduling (FDS), List Scheduling (LS), etc. Resource binding is the process where we assign the same resource to more than one operations which are executed at different time steps [1]. Resource binding indicates that the resources have been shared. The area of the design and the number of resources utilized can be minimized by employing resource binding. The type and number of functional resources and the number of registers required are determined in allocation process in which all the design constraints are considered.

2. Genetic Algorithm

Genetic algorithm is considered as the most sophisticated search and multi objective optimization method based on living nature. Genetic algorithm, widely known as GA mimics the natural process of evolution of life. Genetic algorithm consider a population of individuals initially which are later evolved to form new individuals in that population. The parents are subjected to crossover which generates offspring, is considered as one generation. In each generation, the two individuals with better fitness is added to the initial population of next generation.

This work proposes the use of genetic algorithm to minimize crosstalk in SOCs. A High Level Synthesis framework is developed to reduce crosstalk by simultaneous scheduling, allocation and binding. A DFG (Data Flow Graph) can be scheduled by scheduling algorithms like ASAP, ALAP, FDS etc. These scheduled DFGs are considered as the initial population for GA. In the proposed algorithm ASAP and ALAP scheduled DFGs are the initial populations and are referred as parent_1 and parent_2 which form the first generation. In a DFG with N operations, a random number less than N is chosen as the crossover point and a new generation is formed based on this crossover point which contains child_1 and child_2. Child_1 and child_2 are mutated to schedule the DFGs. Fitness based on MCF is calculated for crosstalk aware binding. The two individuals with minimum coupling (better fitness) are considered as the parents for next generation and the remaining individuals are rejected. This process can be referred as 'Survival of the fittest'. The procedure is repeated for predefined number of generations with different crossover points. The algorithm for the above is shown in Figure 2.

GA Algorithm

1. Initialize all the primary inputs of the given data flow graph with the random numbers.
2. Predecessors and successors are found for each operation based on the data dependency.
3. Initial solution is constructed by ASAP and ALAP scheduling.
4. All the operations are scheduled, allocated and bound simultaneously using Genetic Algorithm.
5. Outputs of the DFG is calculated and the DFG is printed.
6. Verilog file (.v) for the given DFG is generated

Figure 2: Algorithm used to minimize crosstalk using HLS

3. Simulation Results

Crosstalk values based on MCF for a 3x3 matrix multiplication for different algorithms with and without resource binding are tabulated in Table 2. The inputs for this example are generated using pseudo random sequence generator. The capacitive coupling of parent_1, parent_2, child_1 and child_2 for various generations is depicted in Table 3 and same is plotted in Figure 3. The crossover points considered in each generation and selection process is also indicated in Table 3.

The crosstalk values obtained for matrix multiplication clearly shows that genetic algorithm gives better results compared to ASAP or ALAP. Using genetic algorithm with ASAP and ALAP schedule as the initial solutions reduces the crosstalk by 12.2241% for 3x3 matrix multiplication as shown in Table 2. The Verilog file obtained for 3x3 matrix multiplication is synthesized using Xilinx ISE and the functionality is verified. The screenshot of the same is shown

in the Figure 4. The framework developed converts the given data flow graph to crosstalk optimized RTL design.

Table 2: Coupling in DFG of 3x3 matrix multiplication under different conditions

Initial Solution	ASAP(∑MCF)	ALAP(∑MCF)	Genetic Algorithm (∑MCF)
Without Resource Binding	589	589	589
With Resource Binding	561	546	517

Table 3: Coupling In DFG Of 3x3 Matrix Multiplication for 7 Generations

Generation Number	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Crossover point	12	10	32	42	22	25	13
Parent_1	561	539	539	523	523	517	517
Parent_2	546	546	540	523	523	523	517
Child_1	544	540	523	528	517	517	517
Child_2	539	552	523	523	529	523	517
Selected individual 1	Child_1	Parent_1	Child_1	Parent_2	Child_1	Parent_1	Parent_1
Selected individual 2	Child_2	Child_1	Child_2	Child_2	Parent_2	Child_1	Parent_2

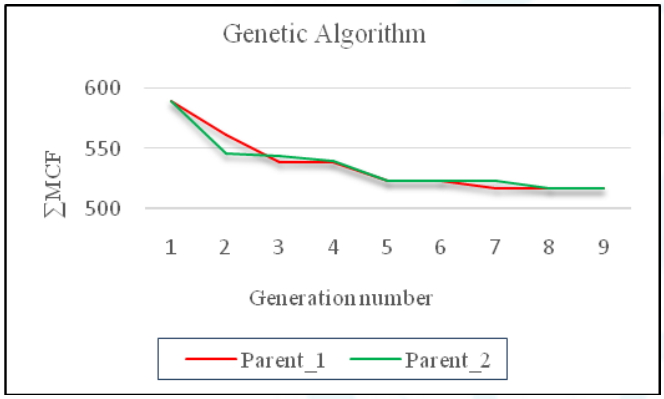

Figure 3: Fitness (coupling) of the parents at each generation

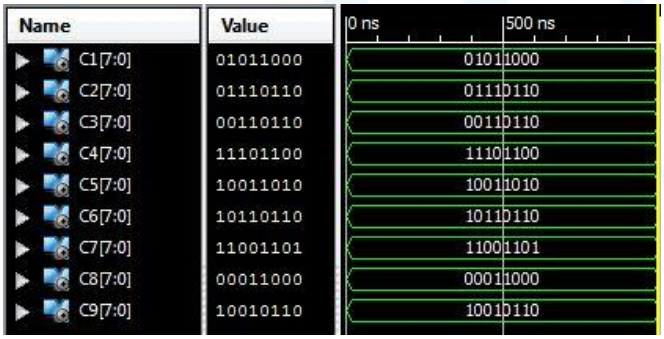

Figure 4: Simulated results of 3x3 matrix multiplication

4. Conclusions

As crosstalk is one of the major concerns of VLSI designs, handling this issue at layout level always may not give optimized result in terms of area and power. Imploring the crosstalk problem at architecture level develops a constraint for design engineers for optimization. The proposed algorithm gives better crosstalk performance for a given Data Flow Graph compromising the latency.

The coupling parameter for matrix multiplication is reduced noticeably using the proposed method. In this work, there are no constraints for resource utilization and latency which play a major role in practical applications.

Crosstalk is estimated using Miller Coupling Factor, although it is a better approximation technique, employing probabilistic method to estimate crosstalk is advisable.

References

[1] Hariharan Sankaran, "High-Level Synthesis Framework for Crosstalk Minimization in VLSI ASICs", University of South Florida Scholar Commons Graduate School Theses and Dissertations USF Graduate School, October 31, 2008.
[2] N. H. E. Weste and D. Harris, "CMOS VLSI", Pearson Addison Wesley, ISBN 0-321 -14901 -7, 2005.
[3] Hariharan Sankaran and Srinivas Katkoori, "Simultaneous Scheduling, Allocation, Binding, Re-Ordering, and Encoding for Crosstalk Pattern Minimization During High–Level Synthesis", IEEE Transactions in VLSI, VOL.19, NO. 2, February 2011.
[4] Brock J. LaMeres and Sunil P. Khatri, " Encoding based Minimization of Inductive Cross-talk for Off-chip Data Transmission",Design, Automation and Test in Europe, 2005.
[5] Proceedings of Colorado and Dept. of Electrical and Computer Engineering Boulder,pp.1318-1323 vol.2, ISBN:0769522882,2008.
[6] C. J. Alpert, A. Devgan, and S. T. Quay, "Buffer insertion for noise and delay optimization," in proc. ACM/IEEE Design Automation Conf., pp. 362-367, 1998.
[7] J.Venkateswara Rao and A.V.N.Tilak, "A Bus Encoding To Reduce Crosstalk Noise Effect In System On Chip", International Journal of VLSI design & Communication Systems (VLSICS) Vol.2, No.2,pp. 110-118 , June 2011.
[8] Sireesha Kondapalli and Dr. Giri Babu Kande, "Fibonacci Codes for Crosstalk Avoidance", Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834,p-ISSN: 2278-8735.Volume 8, Issue 3 ,pp. 09-15,AP,India, Nov. - Dec. 2013.
[9] I. Ahmed, M. K. Dhodhi and C. Y. R. Chen, "Integrated scheduling, allocation and moduleselection for design-

space exploration in high-levelsynthesis," in IEE Proc.-Compur. Digit. Tech., Vol.142, No. I, January 1995.
[10] Iotfii, H. Broumandnia, A. Lofti, "Task graph scheduling in multiprocessor system using a coarse grained genetic algorithm", proc. of the 2nd Intl. Conf. onComputer Technology and Development, Cairo, 2010, pp. 179-185.
[11] Phuong Hong Phan, Thanh Duc Tong, "Genetic Programming Approach for SoC/IP Floorplanning Applications", in proc. of the Intl. Conf. on AdvancedTechnologies for Communications, Ho Chi Minh City, 2010, pp. 291-296.