Test Compaction of Crosstalk Faults through Fault List Reordering

Shehzad Hasan, Ajoy K. Palit, Kishore K. Duganapalli, Walter Anheier
University of Bremen, ITEM, Otto-Hahn-Allee-NW1, D-28359 Bremen, Germany.
E-mails: {hasan, palit, kishore, anheier}@item.uni-bremen.de

Abstract
An algorithm is proposed for generating test patterns that produce maximal crosstalk effect on any interconnect of a circuit using existing ATPG tool and coupling influence between interconnects. The fault list is then reordered so as to reduce the total number of test patterns.

1. Introduction
As the feature sizes of deep sub-micron (DSM) chips decreases and the operating frequencies goes into multi-GHz, an interconnect starts behaving as a transmission line which give rise to Signal Integrity (SI) problems. Due to high aspect ratios of interconnects, the inherent coupling (capacitive or inductive) between adjacent interconnects give rise to crosstalk noise. Crosstalk noise may cause undesirable effects including excessive overshoot, undershoot, glitches, additional signal delay as well as signal speed-up. Crosstalk fault can thus be classified as either crosstalk induced pulse or crosstalk induced delay. In the first case, a fast switching interconnect (aggressor) can induce a short pulse or glitch on a static adjacent interconnect (victim). In the second case, a strong aggressor can induce delay in the victim line if they excite in opposite directions. If the induced noise on victim is above the threshold voltage, or if the induced delay is more than permissible, they may lead to logic failures or functionality problems in the adjacent flip-flops, or at the outputs [1]. In this paper only crosstalk induced pulses are addressed. Excitation of a crosstalk pulse on victim line can only be performed through a rising/falling transition on the aggressor therefore the time line is divided into two parts i.e. before transition and after transition. Positive glitch is defined as a temporary high pulse on a static-0 line (victim) and is tested by applying two test patterns such that aggressor excite from low to high logic (rising transition). A larger crosstalk effect is observed when more interconnects or aggressors transit simultaneously. Testing for a crosstalk pulse requires simultaneous transition of maximum possible aggressors so as to get the maximum effect of positive or negative glitch on a logic-0 or logic-1 line respectively. Interconnects’ influence on each other due to capacitive or inductive coupling is the other factor which needs to be considered. The aggressor influence can be calculated through layout tools. The task is thus to transit maximum number of aggressors having the highest influence on a victim. Reducing the number of test patterns or test set compaction is also important for saving time as well as memory space of testers. Crosstalk faults can however take no advantage of fault equivalence or fault dominance concepts of stuck-at-faults. Neither is static compaction more effective in crosstalk fault patterns as maximum aggressor effect is achieved mostly by a single pattern. Reduction is achieved by reusing the test pattern of previous fault in finding out the new fault.

2. Algorithm
Prerequisites to the algorithm include an aggressor weight matrix, which defines the relative influence of each aggressor on the victim line, through which a sorted list of aggressors is calculated. Each line is kept as victim, and all aggressors are eliminated which have a relative influence less than some threshold value. In addition to this a list of patterns for all stuck-at-faults is acquired from an ATPG program keeping the don’t-care bits unchanged. With the help of these patterns six compatibility matrices are formed i.e. $V_o A_o, V_o A_1, V_o A_2, V_o A_3, A_0, A_1$. Each row of the compatibility matrix of $V_o A_i$ specifies whether a pattern of victim line kept at logic-0 is compatible with patterns of aggressor lines kept at logic-1 where the latter patterns are not required to propagate to the output, while the patterns of victims are selected such that they excite the victim line as well as propagate their effect to at least one output. The task of generating test patterns for crosstalk faults between aggressors and victim is divided into two phases. For positive glitch faults the first phase can be defined as to find the suitable input vector(s), which will excite the victim to logic-0 and as many lines (potential aggressors) as possible to logic-0 starting with the line having most weight and propagate the victim’s effect to the output. The pattern(s) achieved is called Before Transition test pattern(s). This can be done very easily with the help of compatibility graphs and sorted list of aggressors as shown in Fig. 1. The second phase decides the real aggressors. This is done again by exciting victim to logic-0 and then in order of aggressors’ weights exciting as many lines to logic-1. The pattern achieved is called After Transition test pattern. Similarly for negative glitch faults, victim is kept at logic-1 with aggressors having falling transition. The compatibility matrix $V_i A_i$ is used for before transition pattern and $V_i A_i$ for after transition pattern.
3. Test Set Compaction

Fig 1. Flow chart of the algorithm

If an interconnect line is tested as victim for both positive and negative glitch, then four test patterns are required. The concepts of (structural) fault equivalence where one fault can be removed if its tests are exactly same as another fault, cannot be applied here because two inputs of a gate can have different aggressors and thus different patterns. Fault dominance approach of stuck at faults also different aggressors and thus different patterns. Applied here because two inputs of a gate can have its tests are exactly same as another fault, cannot be fault equivalence where one fault can be removed if patterns are required. The concepts of (structural) faults. A total of 44 patterns were required to C17 benchmark circuit was tested for all crosstalk chains of faults are combined until there are only resulting in overall reduction of test patterns. These matching pattern reduces one test pattern thus three patterns are required and each successive for another fault then instead of four patterns only for one fault matches with before transition pattern. For this the After Transition test pattern of the last fault of one sequence is combined with Before Transition test pattern of the first fault of another sequence again by making the Before Transition pattern flexible.

4. Conclusion

In this paper, a new approach for finding out maximum possible aggressor effect on a victim line was presented. Compatibility matrices were used to differentiate between potential and real aggressors. A set of patterns were obtained which gave the maximum crosstalk effect for each interconnect, next these patterns were reordered to reduce the overall test set. The algorithm was applied on C17 but is also applicable for small crosstalk-sensitive zones of other large circuits too.

Reference