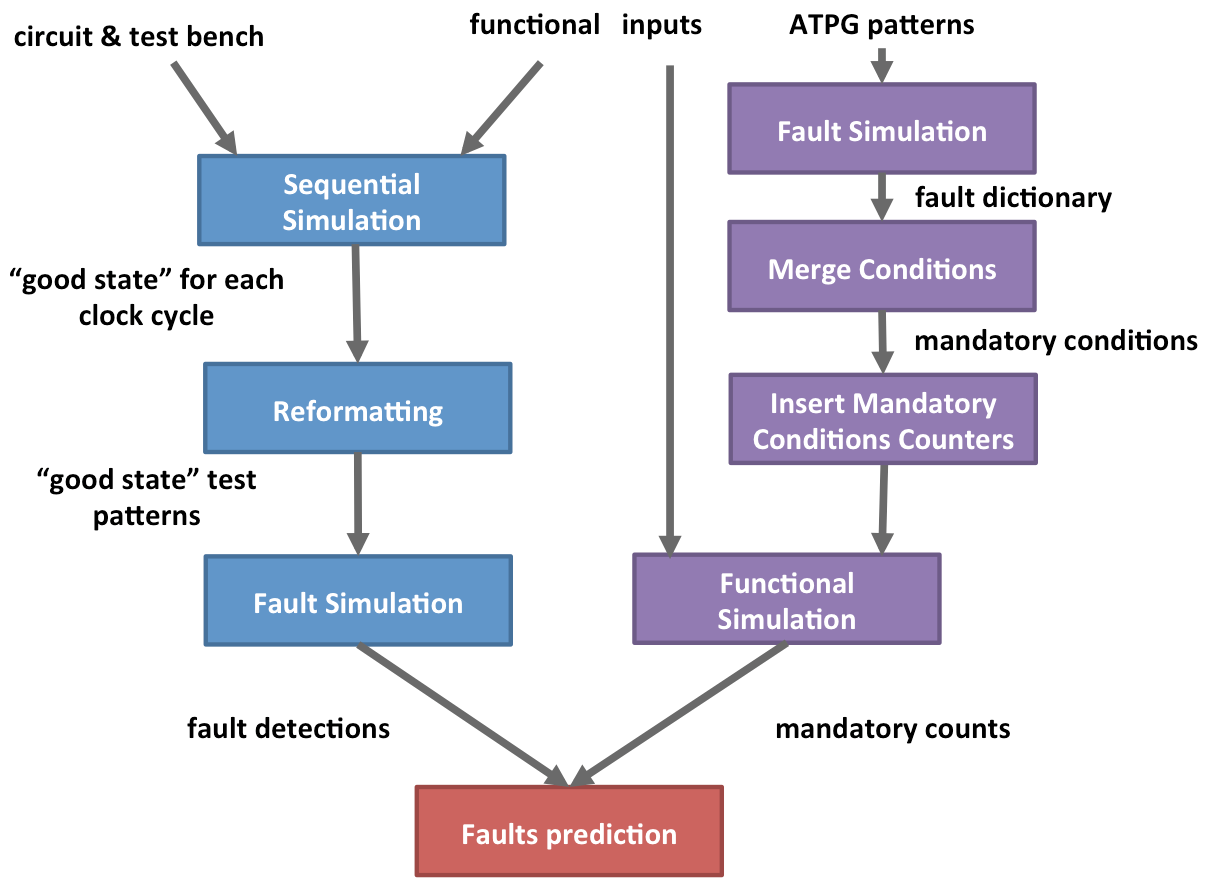
****

Figure 1. Overall flow of mandatory condition counts for potential important fault estimation

**4. Experimental Setup**

To obtain faults prediction, our experiments consisted of two primary portions ——

fault detections were achieved by a fault simulation employed “good state” patterns (the same method described in the paper [1]), and mandatory conditions occurred were counted through functional simulation. The overall flow is shown in Figure 1.

**4.1 Cell-Aware-Type Faults setup**

Based upon Cell-Aware-Type Faults definition [1], we created a single circuit for each standard cell in Synopsys® SAED\_EDK90\_CORE Digital Standard Cell Library, a set of ATPG patterns would be generated by Mentor Graphics® Tessent® for this single circuit. One of test patterns which does not exist in the set of ATPG, would be randomly chosen as the fault for this standard cell through Mentor Graphics® User Defined Fault Models (UDFM).

**4.2 Mandatory conditions**

To generate mandatory conditions, we first implemented Mentor Graphics® Tessent® to generate n-detect test sets of patterns that detects each fault at least n times. For each fault, a fault dictionary recorded the n (or more) patterns that detect this fault. Input and flip-flop bits that are the identical for all of the detection patterns are merged as the mandatory bits. Those input and flip-flop bits that differ at least one times are not mandatory and signed “x” as uncertain value. The procedure of generating a fault mandatory condition is shown in Figure 2.

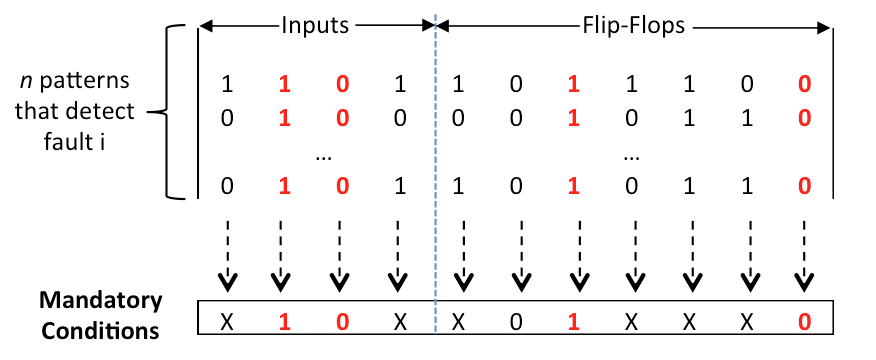


Figure 2. An example of fault mandatory conditions generating process

## **4.3 Mandatory Test Benches and Counts**

Using the mandatory conditions generated above, a test bench which added a AND gate for each particular type of fault was created for each circuits that were examined. Those AND gates that check mandatory conditions are met, and assert high if all conditions are met. They can be fed into counters to count the number of times a mandatory condition occurs. An example of such a gate (for detecting any arbitrary fault i that has n mandatory conditions) is shown figure 3. Functional inputs have been simulated based upon this test bench which retrieved the mandatory counts for each circuit. For the DES-56 circuit, the ASCII format of paper [2] as the plain text inputted for encrypted, then cipher text inputted the DES for decrypted. During simulation, cipher keys randomly chosen for every 64 bits of the plain text encrypted, the identical key for corresponding every 64 bits of the cipher text would be adopted for decrypted. Because we do not know the detail functionality of s9234 benchmark, random input assumed as normal operation during simulation.

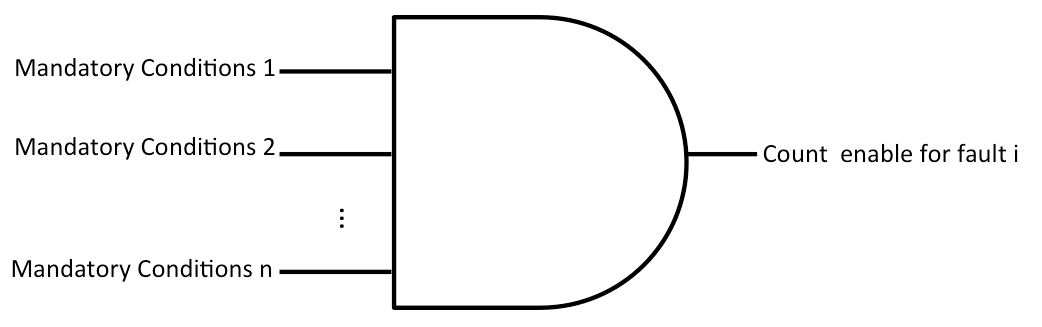


Figure 3. AND-gate for fault I with n mandatory condtions

[1] [Fanchen Zhang](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=p_Authors:.QT.Fanchen%20Zhang.QT.&newsearch=true) ; [Thornton, M.](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=p_Authors:.QT.Thornton,%20M..QT.&newsearch=true) ; [Dworak, J.](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=p_Authors:.QT.Dworak,%20J..QT.&newsearch=true) [North Atlantic Test Workshop (NATW), 2014 IEEE 23rd](http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6875428)

[2] J. Dworak, D. Dorsey, A. Wang, and M.R. Mercer, “Excitation, Observation, and ELF-MD: Optimization Criteria for High Quality Test Sets,” Proceedings of the 2004 IEEE VLSI Test Symposium (VTS’04), Napa Valley, California, April 25-29, 2004, pp. 9-15