Conformal Equivalence

- Introduction

Designing a specification and making sure the protocol never leads to a deadlock while achieving its intended function cannot be verified by simulation alone. Even by some means if we do exhaustive simulation, it doesn't guarantee a complete coverage of the state space and needs a formal approach to verify specification under all permissible delay behaviors. This problem is extensive in asynchronous designs, as a hazard might manifest into a failure under a very particular set of delays.

Given a circuit implementation of this specification, we need to verify that this circuit is hazard free under all permissible delay behaviors as well. The system verification challenge is then to verify that this circuit conforms to the specification in terms of functional implementation.

- Project goal

The initial goal of the project is to implement an algorithm which can model and verify the circuit under all permissible delay models. The algorithm will read a circuit implementation structure file (with extension *.prs) as input wherein the gates of the circuit are described for every output excitation(rise/fall) and their relation to input excitations. The algorithm will build a State Graph (SG) for every gate in the design with all the reachable states modeled. Based on the relations derived from inputs and outputs, all the unexpected output transitions which lead to failure state are modeled as well. Once all the gates are modelled in state graph, we will build a single State Graph for the entire circuit by pruning out redundant and failure state transitions which are unlikely to happen.

If time permits, the algorithm will be extended to build a state graph for the specification as well. The algorithm will read the specification from a (*.lpn) file describing the petri net transitions for the input and output excitations. The system verification goal is then to verify if the circuit state graph conforms to the state graph implemented for specification.