IBM decided to adopt structured logic testing in the early 70’s to solve what seemed to be a very large problem in testing LSI logic. Although the gate array chips only contained around 500 logic gates, the ceramic modules held 100 of these chips and a mainframe system required a minimum of 15 to 20 of these modules. Also, more than 1,000 chip designs were expected, and high quality computer generated tests were considered essential.

Our first structured design for testability included full scan for logic and some limited design rules for buried memory arrays. It was a bumpy road for the first few years, but eventually design for testability became an accepted part of IC design.

As our technology improved, the design rules for testability evolved to meet new requirements and provide enhanced testing. This included delay testing, weighted random pattern testing, and built-in self-test for both logic and memory.

When I look at today’s microprocessors, their size and complexity make them seem almost impossible to test in a cost-effective way. But many of the techniques used in the past may still be useful.

The main objective of scan design was to simplify the sequential test problem to that of testing combinational logic. This is still the foundation of structured logic test.

The stuck-at fault model, which has been in use since the late 1950s, will likely still be the primary fault model. There are at least two reasons for this longevity. First it provides $n+1$ tests for every logic gate with $n$ inputs, and second it is possible to find a test for all stuck-at faults unless there is logic redundancy.

Finally, if you try to establish structured test into your organization, it will be strongly opposed by the majority of the IC design team since no one likes new design rules. This means you must initially get the right people, including some key IC designers, to help you make structured test successful.