Software-basierender Selbsttest von Prozessorkernen unter Verlustleistungsbeschränkung

Jun Zhou, Hans-Joachim Wunderlich

Institut für Technische Informatik
Universität Stuttgart
Pfaffenwaldring 47
70569 Stuttgart, Deutschland
zhoujn@informatik.uni-stuttgart.de
wu@informatik.uni-stuttgart.de

Software-basierender Selbsttest (SBST) von Prozessoren wird schon seit Jahrzehnten verwendet, da er erhebliche Vorteile bietet, wie z. B. die Wiederverwendbarkeit in allen Stadien des Lebenszykluses des Systems, der Verzicht auf kostspielige Tester und geringe oder sogar gar kein Aufwand für „Design for Test“. Die größten Nachteile sind die unzureichende strukturelle Fehlerabdeckung sowie relativ lange Testanwendungszeiten. In jüngster Zeit wurden große Fortschritte bei der Kombination der Erzeugung von strukturellen Testmustern und SBST gemacht. Dieser Ansatz macht das „Peak Power Problem” weniger dringend, da der Test im Systemmodus und nicht in einem speziellen Testmodus durchgeführt wird, der zu einer erhöhten Schaltaktivität führt. Der durchschnittliche Energiebedarf muss jedoch immer noch optimiert werden. Geringer Speicherbedarf, kurze Ausführungzeiten sowie eine hohen Fehlerabdeckung sind für den effizienten Einsatz der SBST Methode unerlässlich.

In diesem Beitrag wird eine strukturelle SBST Methode präsentiert, welche diese drei Parameter gleichzeitig optimiert. Zuerst stellen wir einen Algorithmus vor, mit dem ein effektives Testprogramm mit minimalem menschlichem Eingreifen synthetisiert werden kann. Die konventionellen Techniken zur Energieoptimierung der Software können an dieser Stelle nicht angewandt werden, da mit ihnen das Ziel verfolgt wird, die Semantik eines Programms beizubehalten, während bei SBST hingegen die strukturelle Fehlerabdeckung beibehalten werden muss. Um dieser speziellen Anforderung gerecht zu werden, stellen wir anschließend unseren Optimierungsprozess vor, der das Verhalten des eingebetteten Tests analysiert, um dieselbe Fehlerüberdeckung zu erhalten, und dabei Freiheitsgrade bei der Befehlsreihenfolge und Don’t Cares verwendet , um Energie zu sparen. Schließlich wird die Methode auf Gatterebene validiert. Den Ergebnissen zufolge kann die Energie um 40% und die Testdauer um 30% reduziert werden ohne an Fehlerabdeckung einzubüßen.