TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Informatik Institut für Technische Informatik, Professur für VLSI-Entwurfssysteme, Diagnostik und Architektur

Erweiterung der Trace-Hardware eines Mikroprozessors für die Fehlerinjektion und -beobachtung

Vorstellung der Diplomarbeit

Marco Gunia
marco.gunia@tu-dresden.de

28. November 2013

Inhalt

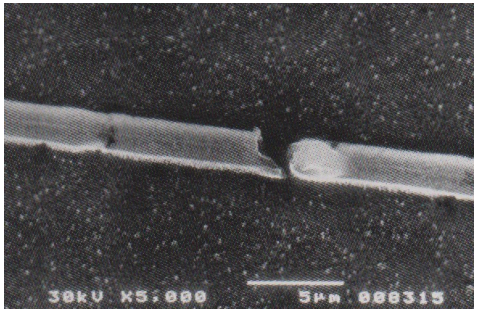
- Einleitung
- Aufgabenstellung
- Aktueller Stand
- Verbleibende Aufgaben

Einleitung

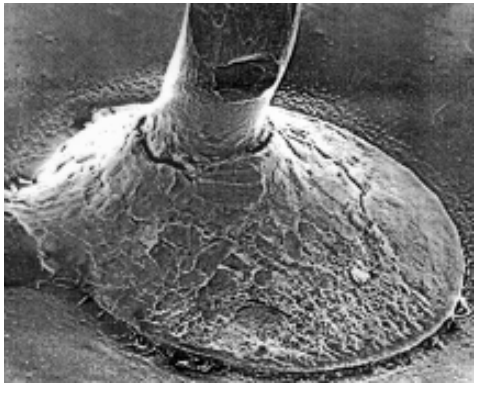
Fehler in digitalen Schaltungen

Fehler in digitalen Schaltungen unterteilen sich in

- Permanente Faults und
  - → Defekte in der Metallisierung,
  - → Defekte in den Diffusionsgebieten,
  - → Defekte in den Isolatoren,
  - → Defekte beim Chip Bonding,
  - → Defekte bei der el. Kontaktierung,
  - → Entwurfsfehler und
  - → Alterungseffekte.
- Transiente Faults.
  - → Externe Strahlung,
  - → Rauschen und
  - → Elektromagetische Störungen.



Quelle: [Hill04]



Quelle: [Wol07]

Einleitung

Trace

Klassische Debug-Techniken zur Fehlersuche

- Simulation,
- In-Circuit Emulator oder
- Software Instrumentation.

Nachteile klassischer Techniken:

- Genauigkeit,
- Intrusivität und
- Postmortem Debugging.

→ Tracing beschreibt den Prozess der Protokollierung der Zustandsänderung eines Systems während der Programmausführung. Das Resultat wird Trace bezeichnet. [Ale09]

Aufgabenstellung

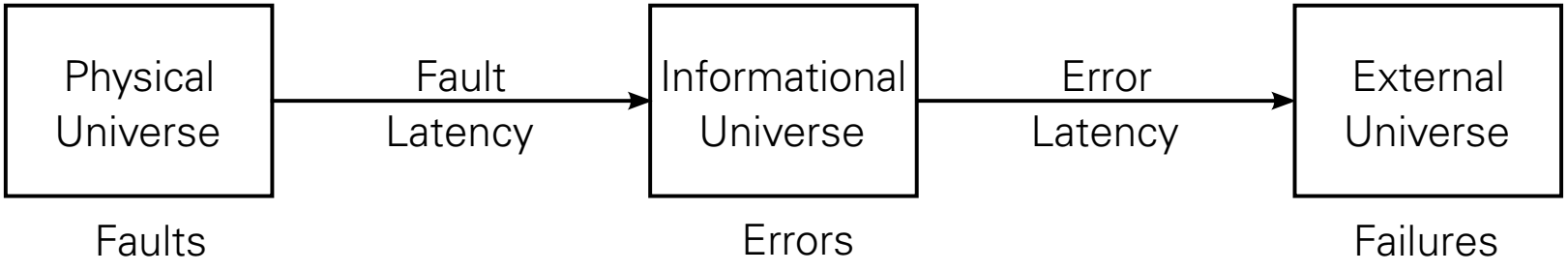
Das Ziel der Arbeit ist die kontrollierte Nachbildung sowohl transienter als auch permanenter Fehler in einem Mikoprozessor ohne Unterbrechung der Programmausführung durch eine Erweiterung vorhandener Trace-Hardware.

- Literaturstudium zu Fehlermodellen, Fehlerinjektion und Fehlernachweis,
- Literaturstudium zu Trace- und Debug-Hardware in aktuellen Mikroprozessoren,
- Analyse der Anforderungen an eine Erweiterung der Trace-Infrastruktur,
- Analyse der Beobachtbarkeit der injizierten Fehler mittels Trace,
- Entwurf und prototypische Implementierung der Trace-Erweiterung mit Parametrierung,
- Test der Fehlerinjektion und -beobachtung am konkreten Beispiel und
- Bewertung und Dokumentation der erzielten Ergebnisse.

Grundlagen

Fehler in digitalen Schaltungen

Es wird unterschieden zwischen Faults, Errors und Failures.



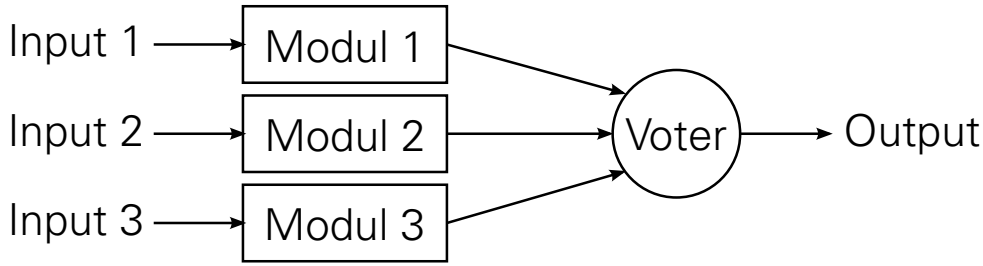
Unterscheidung zwischen Faults, Errors and Failures (Quelle: [Joh89])

Grundlagen

Fehler in digitalen Schaltungen

Begriffe:

- Fehlermodelle,
- Fehlerinjektion,
  - → Physikalische Fehlerinjektion,
  - → Hardwarefehlerinjektion,
  - → Softwarefehlerinjektion und
  - → Simulationsbasierte Fehlerinjektion.
- Fehlertolerante Systeme und
  - → Fault Avoidance,
  - → Fault Masking und
  - → Fault Tolerance.
- Redundanz.
  - → Hardware,
  - → Information und
  - → Zeit.



Triple Modular Redundancy (Quelle: [Joh89])

Aktueller Stand

Folgende Arbeiten wurden vor Beginn der Diplomarbeit durchgeführt:

- Entwurf des ZiLOG Z80 Prozessors in Verilog mithilfe von Standardzellen.

Folgende Aufgaben wurden im Rahmen der Diplomarbeit bearbeitet:

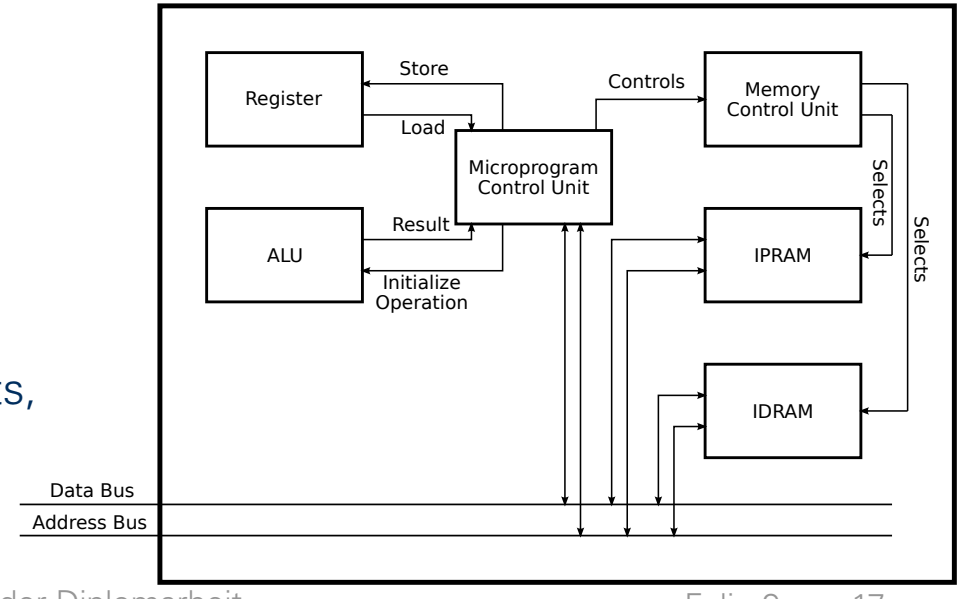
- Präsentation von Trace-Hardware in aktuellen Mikroprozessoren,
- Synthese des Z80 auf einem Xilinx Virtex 5,
- Einarbeitung in das zur Verfügung gestellte Trace-Framework und dessen Integration in den Prozessor,
- Vorstellung von Fehlermodellen, Fehlerinjektion und Fehlernachweis,
- Untersuchung von Alternativen für die Fehlerinjektion und
- Entwicklung einer Methode zur Fehlerinjektion in den Z80.

Aktueller Stand

ZiLOG Z80

Der ZiLOG Z80 beschreibt einen 8-Bit CISC-Mikroprozessor, der abwärtskompatibel zum 8080 von Intel konzipiert wurde. Ausgewählte Eigenschaften:

- General Purpose Register,
  - Sieben Register und ein Flagregister sowie deren Schattenregister.
  - Für 16-Bit Operationen Konkatenation von 8-Bit Registern zu 16-Bit Registern.
- Weitere Register und
  - Befehlszähler,
  - Stackpointer und
  - Indexregister.
- Interrupts.
  - Maskierbare und nichtmaskierbare Interrupts,
  - Laden des nachfolgenden Befehls aus Speicher oder über externen Datenbus.

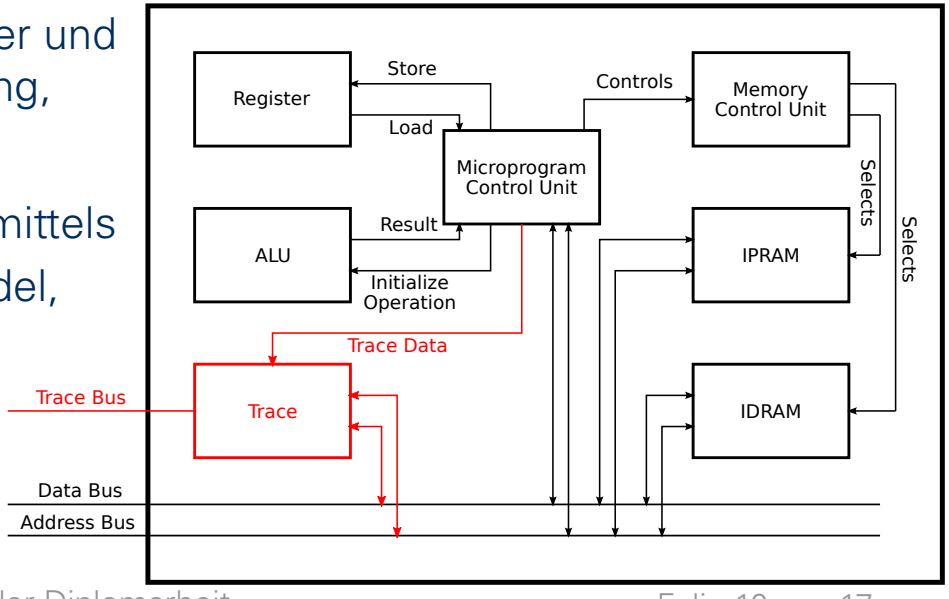


Aktueller Stand

Integration der Trace-Hardware in den Z80

Das Trace-Modul wurde von Herrn Stefan Alex im Rahmen seiner Diplomarbeit für SHAP konzipiert. Es ist durch folgende Eigenschaften gekennzeichnet:

- Off-Chip-Trace unter Nutzung der Gigabit-Ethernet-Schnittstelle,
- Protokollierung von Instruktionen, Daten, Nachrichten und Statistiken,
- Auswahl zwischen zyklusakkurater und nicht-zyklusakkurater Aufzeichnung,
- Cross-Trigger-Funktionalität,
- Kompaktierung der Trace-Daten mittels
  - Program-Flow-Change-Model,
  - XOR-, Trim- oder Differenz-kompression.
- Bereitstellung der Host-Software zur Dekodierung.

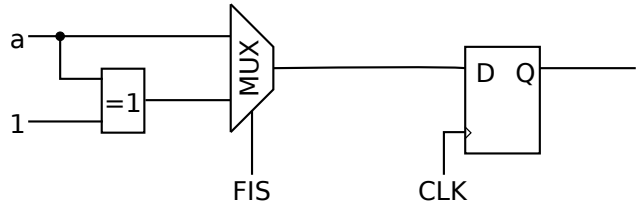


Aktueller Stand

Fehlermodelle und Fehlerinjektion

Im Rahmen der Arbeit werden

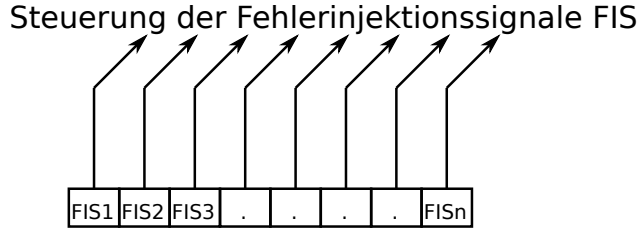
- Fehlermodelle auf Schalter-, Gatter- und Registertransferebene vorgestellt, wobei sich
- die Nachbildung auf *Stuck-at-0*, *Stuck-at-1* und *Bit-Flips* auf Registertransferebene beschränkt.



Bit-Flip-Fehlermodell (Quelle: [Shi08], [Shi13])

Zur Fehlerinjektion stehen folgende Verfahren zur Verfügung:

- Physikalische Fehlerinjektion,
- Softwarefehlerinjektion,
- Simulationsbasierte Fehlerinjektion und
- Hardwarefehlerinjektion.
  - → Rekonfigurationsbasierte Verfahren und
  - → Instrumentierungsbasierte Verfahren.



Maskenregister zur individuellen Auswahl des entsprechenden Faults (Quelle: [Civ+01b])

Aktueller Stand
Hardwarefehlerinjektion

Instrumentierungsbasierte Verfahren:

- Hinzufügen von Schaltungsteilen zum Entwurf für die Fehlerinjektion.

Rekonfigurationsbasierte Verfahren:

- Klassische Rekonfiguration erzwingt die Veränderung des Quellcodes und somit eine eigene Synthese, Place & Route und Bitfile-Generierung für jeden Fault.
- Partielle Rekonfiguration umfasst die teilweise Veränderung des Bitstreams zur Laufzeit.
  - → Modifikation von LUTs oder
  - → Initialisierung von Flipflops.

Vergleich instrumentierungsbasierter und rekonfigurationsbasierter Verfahren:

- Durchführung mehrerer Experimente in Echtzeit mittels instrumentierungsbasierter Verfahren möglich, aber
- Keine zusätzliche Hardware für rekonfigurationsbasierte Verfahren notwendig.

Aktueller Stand

Fehlerinjektion in den Z80

Bedingungen an die Fehlerinjektion:

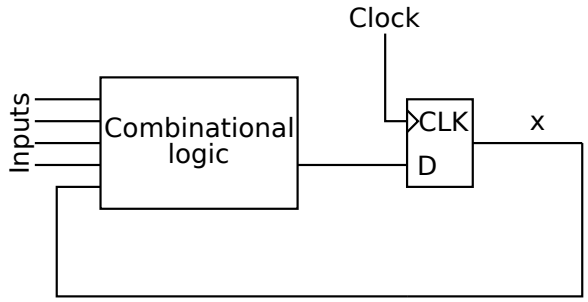
- Keine Veränderung des Zeitverhaltens und
- Synthesefähigkeit.

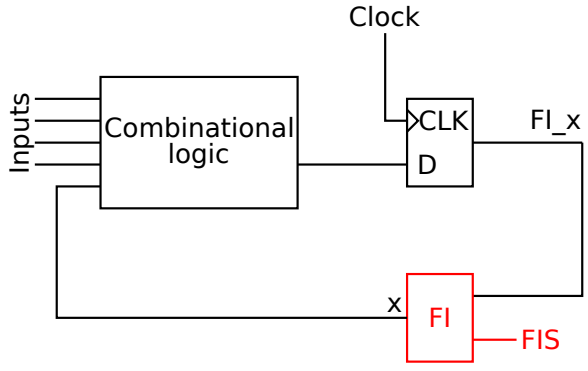
Verilog-Einschränkungen:

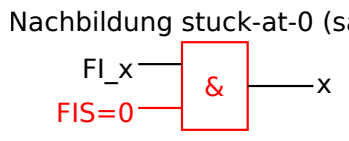
- Zuweisung an ein Signal stets nur in einem Prozess und
- Default-Anweisung für Signale am Anfang des Prozesses nicht spezifikationskonform.

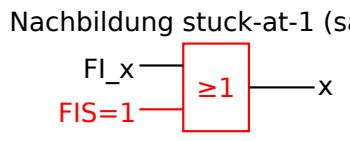
Lösung:

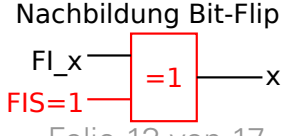
- Neue Signale an Flipflop-Ausgängen,
- Rückführung auf ursprüngliche Signale über Fehlerinjektionslogik und
- Realisierung von „dauerhaften“ Bit-Flips mittels Signalzuweisung an Flipflop in jedem Takt.









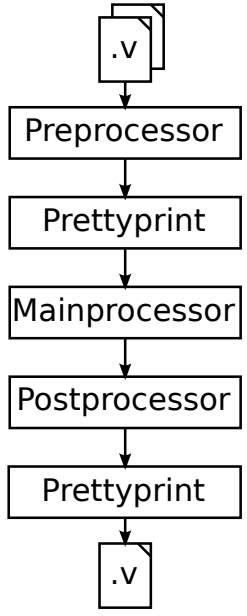


Aktueller Stand

Fehlerinjektion in den Z80

Den Ausgangspunkt bildet eine synthetisierbare Verilog-Beschreibung. Zur Integration der Fehlerinjektion wird der Quellcode in folgenden Schritten modifiziert:

- Preprocessor,
  - → Verändert die Hardwarebeschreibung entsprechend den im Quellcode vorhandenen Präprozessoranweisungen.
- Prettyprint,
  - → Gruppiert Anweisungen und
  - → Formatiert den Quelltext.
- Mainprocessor und
  - → Erfasst die für die Fehlerinjektion relevanten Signale und deren Eigenschaften (z.B. Signaltyp und -breite) und
  - → Ergänzt die Hardwarebeschreibung um Kennzeichner zur Integration der Logik zur Fehlerinjektion.
- Postprocessor.
  - → Interpretiert die vom Mainprocessor ergänzten Kennzeichner,
  - → Erweitert die Beschreibung um Signale und Logik zur Fehlerinjektion.



Verbleibende Aufgaben

Folgende Aufgaben sind zu bearbeiten:

- Nachweis der aufgetretenen Fehler mittels Trace und
- Implementierung von Fehlertoleranzmaßnahmen und Evaluation.

Vielen Dank für die Aufmerksamkeit!

Quellen

[Ale09] Alex, S.: *Entwurf und Implementierung einer parametrierbaren Trace-Hardware am Beispiel der SHAP-Mikroarchitektur*
Diplomarbeit, TU Dresden: Institut für Technische Informatik, 2009

[Civ+01b] Civera, P. u.a.: *FPGA-based Fault Injection for Microprocessor Systems*
Proceeding of the 10th Asian Test Symposium, 2001

[Hil04] Hilleringmann, U.: *Silizium-Halbleitertechnologie.*
Wiesbaden: Teubner Verlag, 4. Auflage, 2004

[Joh89] Johnson, B.W.: *Design and Analysis of Fault-Tolerant Digital Systems*
Addison-Wesley Publishing Company, 1989

[Wol07] Wolter, K.-J.: *Vorlesungmsskript Aufbau und Verbindungstechnik 1*
TU Dresden: Institut für Aufbau- und Verbindungstechnik der Elektronik, 2007