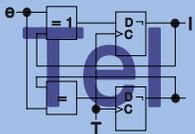


Diplomarbeit – Verteidigung

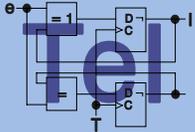
**Mikrocontrollergestützte
Selbstorganisationsprinzipien
rekonfigurierbarer
Rechnersysteme am Beispiel der
Xilinx FPGA-Architektur**

Falk Niederlein

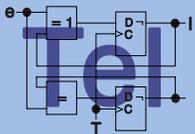
s6838029@inf.tu-dresden.de



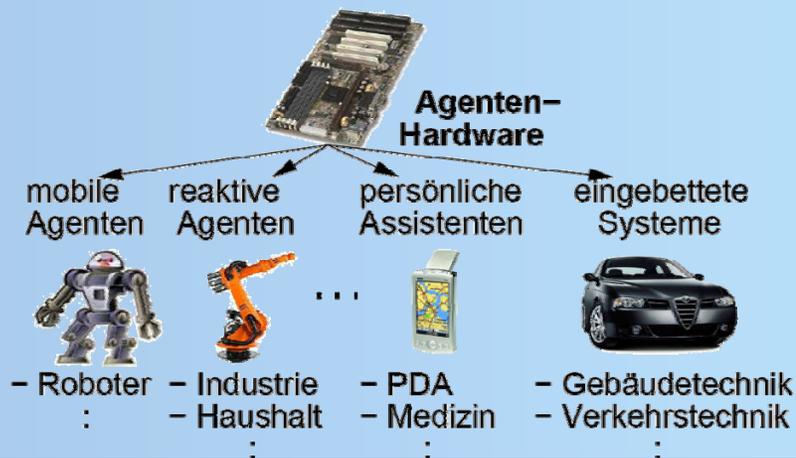
- 1 Aufgabenstellung
- 2 Motivation
- 3 Einleitung
- 4 Lösungsansätze
- 5 Umsetzung
- 6 Zusammenfassung und Ausblick



- Einarbeitung in die Aufgabenstellung und Literaturstudium zu Selbstorganisationsprinzipien rekonfigurierbarer Rechnersysteme
- Methodische Untersuchungen für eine netzwerkgestützte Rekonfiguration von Xilinx-FPGAs und Auswahl einer geeigneten Mikrocontroller-Architektur
- Implementation eines Programms als „add on“ zur Erweiterung des FPGA-Entwurfsablauf für die Realisierung einer netzwerkgestützten Download-Möglichkeit und zur Rekonfiguration der Xilinx FPGA-Architektur
- Erstellung von Beispielen und Szenarien zur Validation der entwickelten Programmlösungen und zur Demonstration der mikrocontrollergestützten Anpassungsfähigkeit rekonfigurierbarer Rechnersysteme
- Untersuchungen zur Effizienz der implementierten Programme und Applikationen und Auswertung der Implementationsergebnisse
- Dokumentation der Programme und Ergebnisse

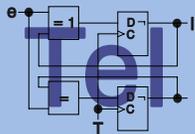


- Der andauernde Fortschritt in der Mikroelektronik führt zu immer komplexeren Systemen
- Zur Beherrschung sollen Prinzipien des „Organic Computings“ angewandt werden
- Organische Computersysteme arbeiten möglichst selbstorganisierend, dies erfordert ein adaptives und kontextbewusstes Verhalten
- Das so entstandene selbstorganisierende System stellt Dienste zur Verfügung
- Dienstbringende Systeme sind bereits in der Softwareentwicklung etabliert und sollen auch in Hardwaresystemen zu Einsatz kommen
→ Agententechnologie
- Einsatzgebiete:



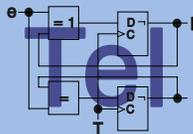
Agentensysteme und Organic Computing

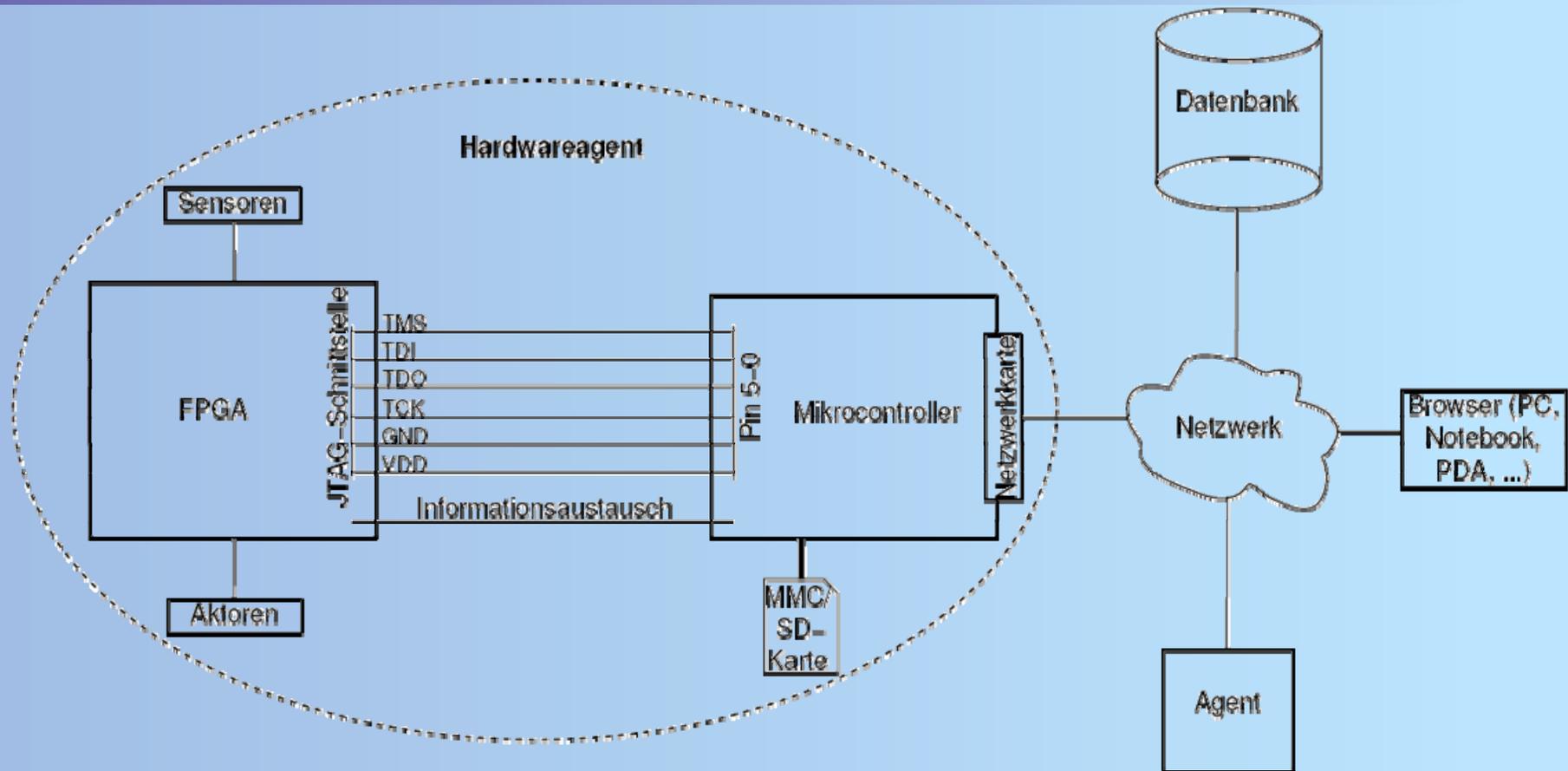
- eine solche universelle Hardware-Plattform mit OC-Eigenschaften kann zukünftig als HW-Agent existieren
- Merkmale eines Agentensystems:
 - Systeme arbeiten weitestgehend unabhängig von Benutzereingriffen
 - Lösen Aktionen aufgrund eigener Initiative aus
 - Reagieren auf Änderungen der Umwelt
 - Kommunikation mit anderen Agentensystemen
 - Lernend durch zurückliegende Entscheidungen oder Erkenntnissen
- Entwicklung eines Agentensystems mit Eigenschaften des Organic Computing zur Erhöhung der Anpassungsfähigkeit und Robustheit
- Für die Selbstorganisation werden die Self-x-Eigenschaften benötigt:
 - Selbst-konfigurierend
 - Selbst-optimierend
 - Selbst-heilend
 - Selbst-schützend



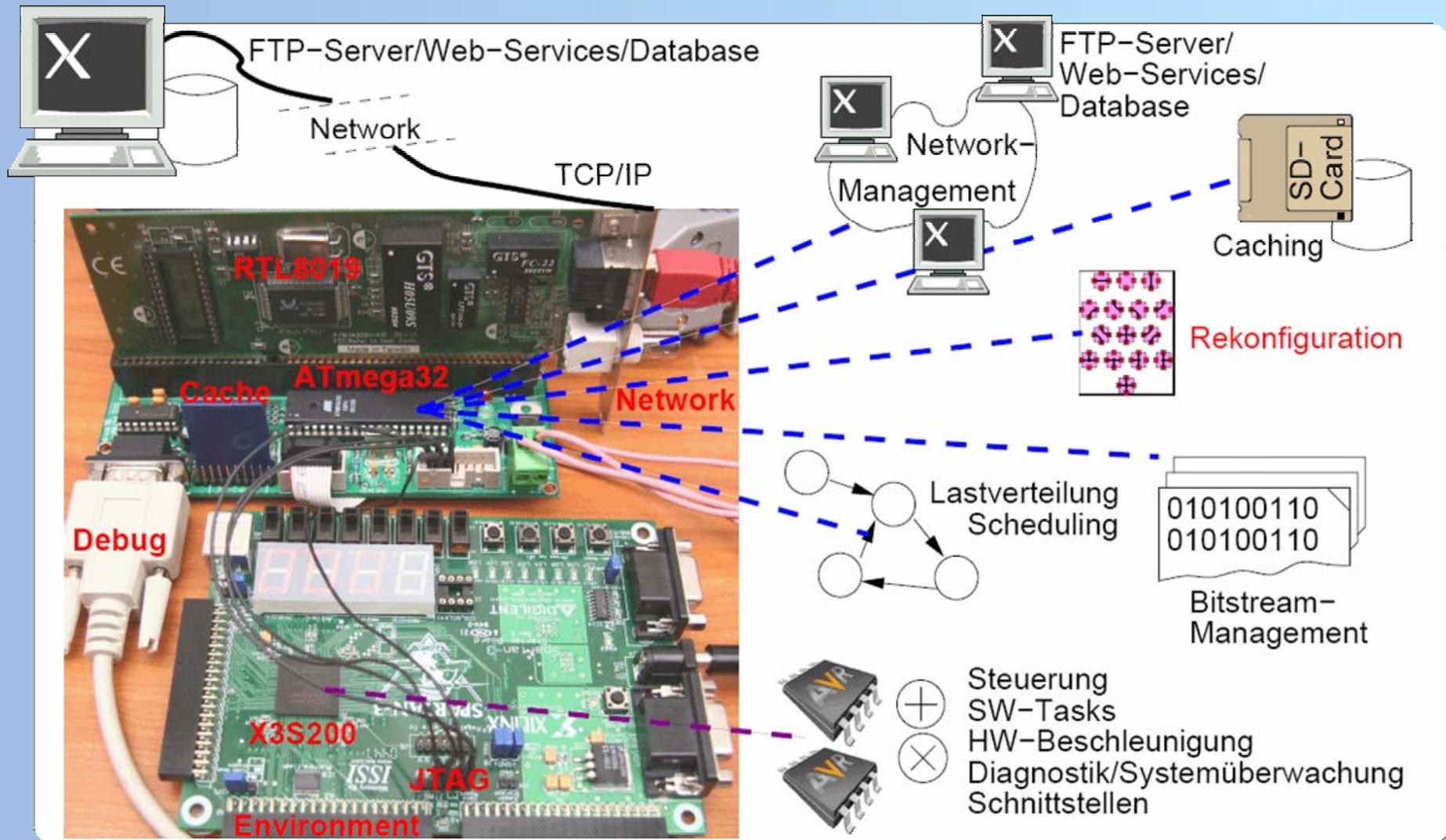
HW-Unterstützung für das Agentensystem

- Konzeption einer für Agentensysteme spezialisierten HW-Architektur, welche hohe Flexibilitätsanforderungen gerecht wird
- Einsatz von programmierbaren und rekonfigurierbaren HW-Baugruppen zur Umsetzung der Selbstorganisations- und Robustheitsanforderungen
- Rekonfigurierbare HW ist repräsentiert durch ein Xilinx-FPGA
 - Führt Task aus
 - Verschiedene Konfigurationen für unterschiedliche Tasks
 - Anpassung der Konfiguration an Umgebungsanforderung
 - HW-Beschleunigung
- Programmierbare HW ist repräsentiert durch den Mikrocontroller ATMEL ATmega32
 - Für die Verwaltung zuständig
 - Steuert den Rekonfigurationsprozess
 - Ist Schnittstelle zum Netzwerk

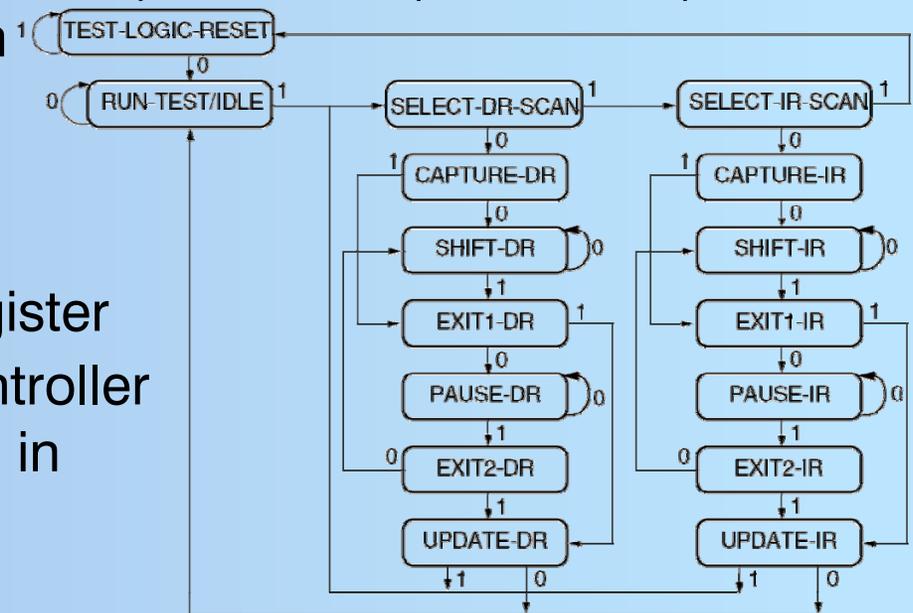




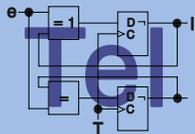
- Agentensystem bestehend aus FPGA und Mikrocontroller
- Verbunden durch Konfigurations- und Kommunikationsschnittstelle
- Netzwerkfähigkeit erlaubt Datenbankzugriff, Kommunikation mit anderen Agenten und Browseransteuerung



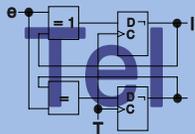
- Rekonfiguration erfolgt über die JTAG-Schnittstelle
 - JTAG ist ein weit verbreiteter Standard
 - Statische und partielle Rekonfiguration möglich
 - Mehrere Bausteine können in einer Kette zusammengeschaltet und gezielt angepasst werden
- Die Rekonfiguration wird vollständig über den Test Access Port (TAP) ausgeführt, wobei die TAP-spezifischen Pins TDI (*Test Data In*), TDO (*Test Data Out*), TMS (*Test Mode Select*) und TCK (*Test Clock*) entsprechend angesteuert werden
- JTAG compatible HW besitzt dedizierte Komponenten, u.a. den TAP, einen Steuerautomaten (TAP-Controller) und mehrere Register
- Das TMS-Pin steuert den TAP-Controller und das TDI-Pin schiebt die Daten in die Register



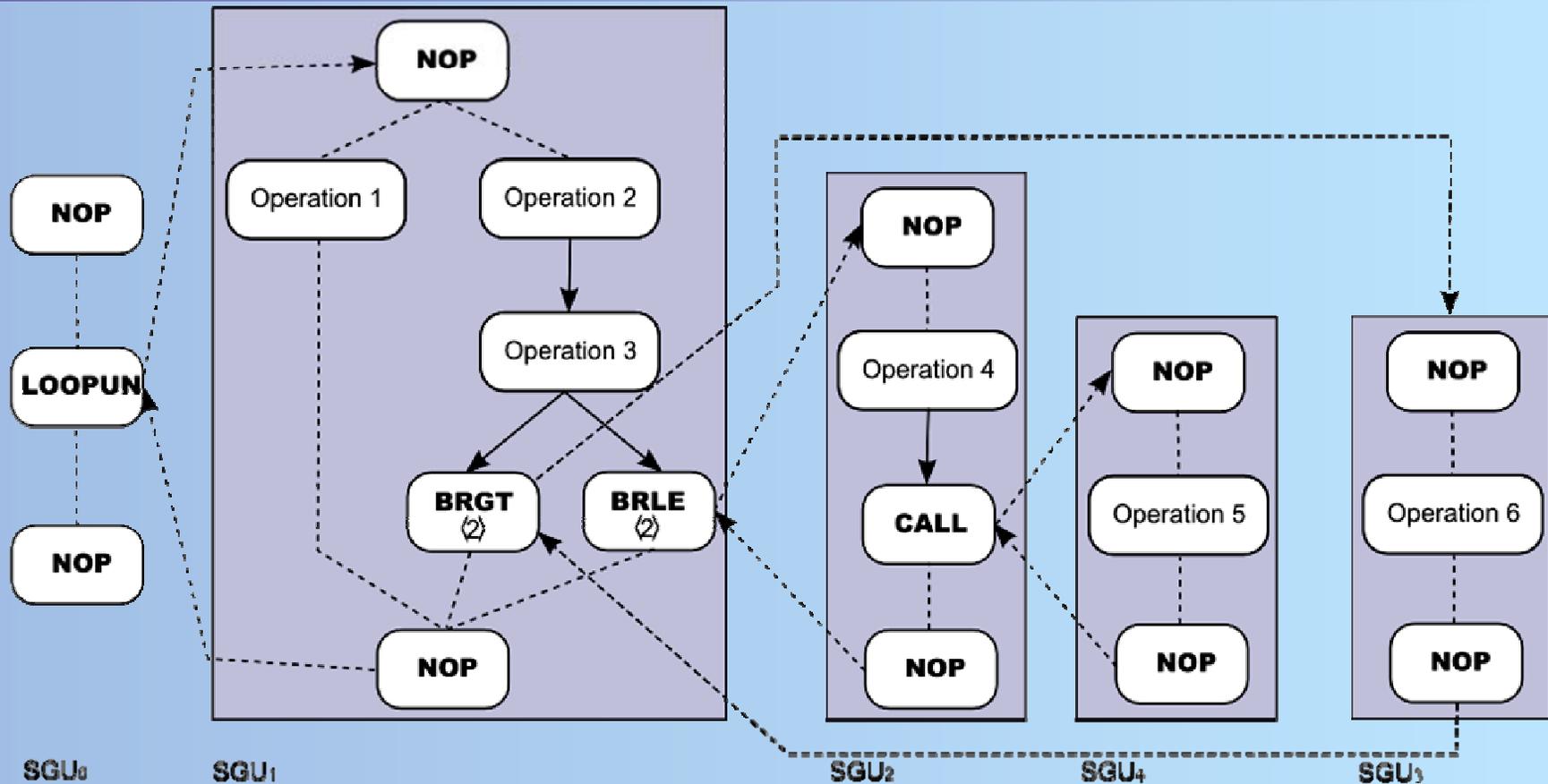
- Die Konfiguration über die JTAG-Schnittstelle erfordert das spezielle SVF- (*Serial Vector Format*) Dateiformat
- Das SVF-Dateiformat ist durch einen industriellen Standard definiert und stellt die JTAG-Befehle in portabler und kompakter Form dar
- Das Format vereinheitlicht hersteller- und bausteinspezifischen Abweichungen
- Für rekonfigurierbare Logikbausteine der Firma Xilinx erfolgt weiterhin eine Konvertierung in das XSVF- (*Xilinx Serial Vector Format*) Dateiformat
 - Optimiert für Xilinx-Bausteine
 - Binäre Form
- Die Dateien werden mit der Entwurfssoftware (iMPACT) generiert
- Der Mikrocontroller interpretiert die XSVF-Datei und steuert mit den enthalten Daten die TAP-Pins



- Das Scheduling ist auf dem Mikrocontroller untergebracht und verwaltet/steuert die Funktion des Agentensystems, welche durch die FPGA-HW realisiert werden kann
- Das Scheduling leitet den Rekonfigurationsprozess ein und sorgt für eine zielgerechte Anpassung des Agentensystems
- Das Scheduling wird durch das Prinzip der Sequenzgraphen realisiert
- Sequenzgraphen vereinen Steuer- und Datenfluss
- Der Steuerfluss enthält Steuerkomponenten für die Verzweigung und Iteration, der Datenfluss zeigt Datenabhängigkeiten
- Für die Maschinenlesbarkeit werden die Graphen in Adjazenzmatrizen abgebildet und vom Mikrocontroller interpretiert
- Das Scheduling beinhaltet zusätzlich eine Möglichkeit zum FTP-Download, zur Erhöhung der Aktualität und Erweiterbarkeit



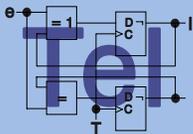
Beispiel eines Sequenzgraphen

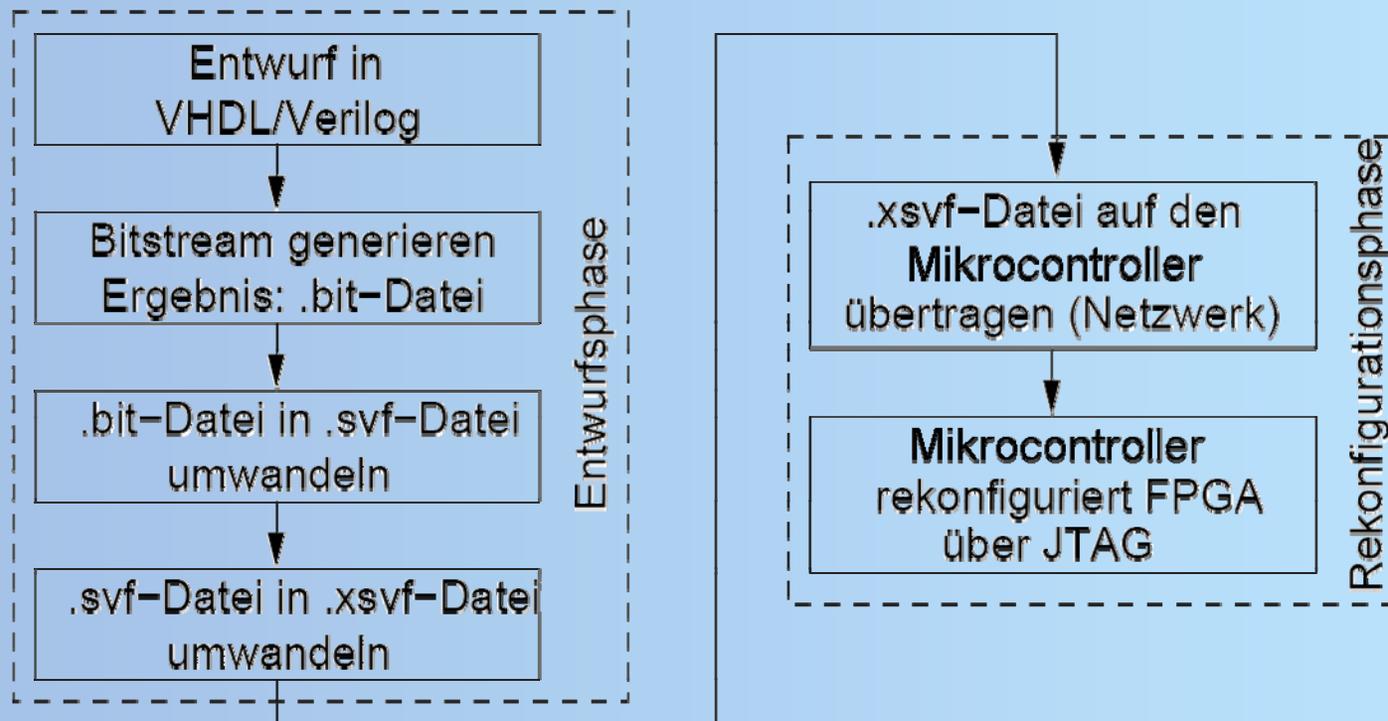


- Ein Sequenzgraph mit mehreren Einheiten
- Jede Einheit ist in einer separaten Datei gespeichert
- Steuerknoten sind: NOP (Ein- bzw. Austrittspunkt), LOOP (Iteration), BR (Verzweigung) und CALL (Modulaufruf)

- Für die Anpassung des FPGAs benötigt der Mikrocontroller Informationen über den internen Zustand
- Eine Verbindung zwischen FPGA und Mikrocontroller ist nötig
- Mindestens ein Steuerbit für die Abarbeitung von Routinen bzw. Initialisierungen
- Die Übertragung mehrerer Bits ermöglicht erst den Einsatz von Sequenzgraphen
- Verzweigungen und Iterationen anwendbar durch Vergleich der in den Sequenzgraphen enthaltenen Referenzwerte

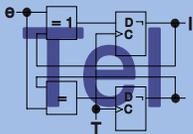
- Selbstorganisation kann nicht vollständig erfüllt werden
- Die Intelligenz des Systems kommt nicht von sich selbst, sondern wird durch die Sequenzgraphen vorgegeben
- Die für die Robustheit und Aufgabenverteilung wichtige Kommunikation zwischen den Agentensystemen ist zurzeit nicht realisiert

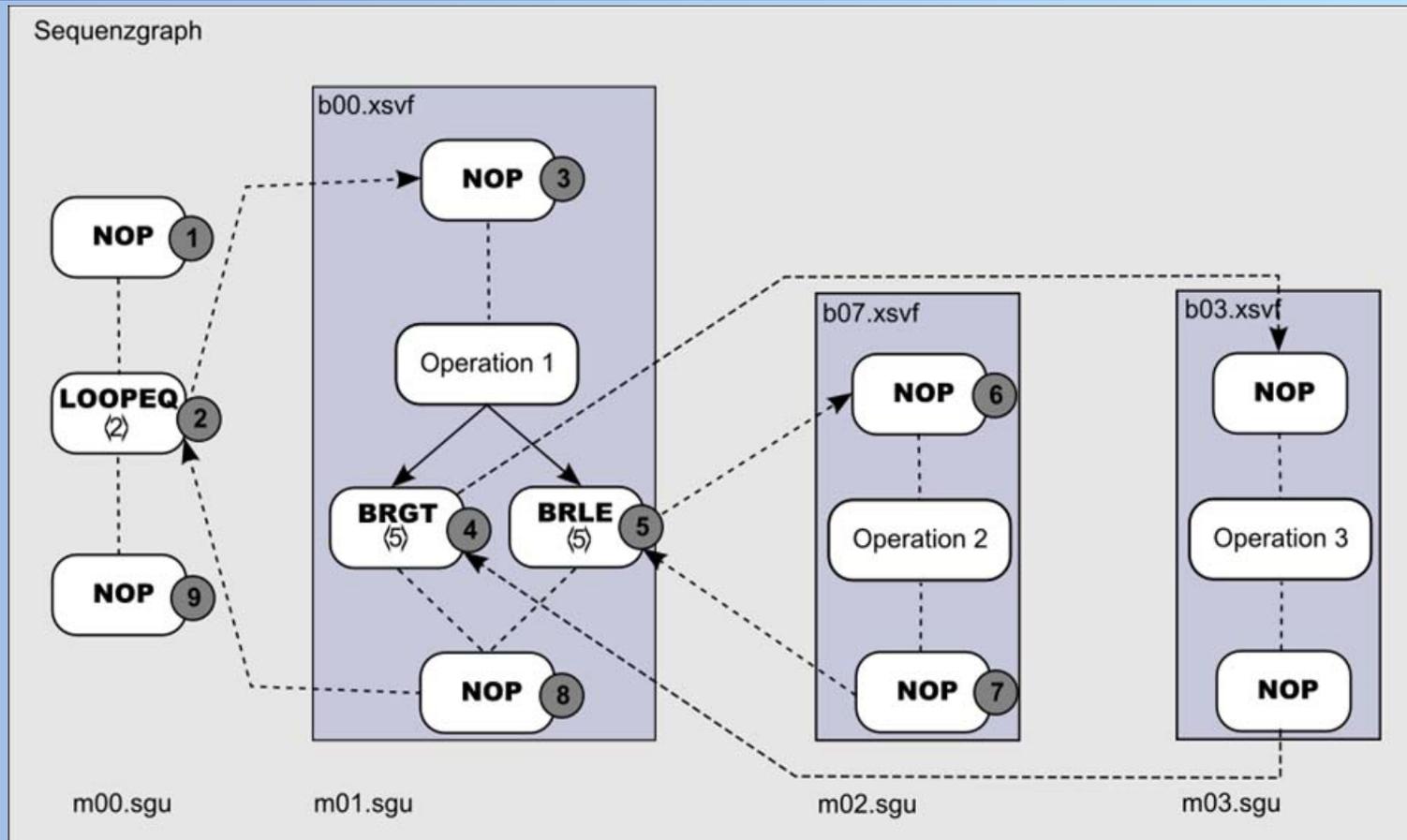




- Zwei Hauptphasen: Generierung der Entwurfsdatei (Entwurfsphase) und Rekonfiguration des FPGAs
- Die Ausführung der Rekonfigurationsphase wird durch das Scheduling auf dem Mikrocontroller gesteuert

- Die XSVF-Datei enthält alle Daten für die Rekonfiguration des FPGAs über die JTAG-Schnittstelle
- Durch Interpretation der XSVF-Datei werden die TAP-Pins TDI, TMS und TCK richtig angesteuert bzw. der Wert an TDO korrekt ausgewertet
- Die XSVF-Datei ist eine Aneinanderreihung von XSVF-Befehlen und dazugehörigen Parametern
- Die Befehle sind Steueranweisungen und beschreiben einerseits die Ansteuerung der TAP-Pins andererseits enthalten sie Informationen für die ausführende Software
- Es gibt zurzeit 22 XSVF-Befehle; nicht alle werden zur Rekonfiguration benötigt
- Die wichtigsten Befehle sind die XSDR bzw. XSIR Befehle, dessen Parameter über TDI in das Daten- bzw. Befehlsregister gelangen
- Der Algorithmus „springt“ von Befehl zu Befehl, identifiziert ihn und führt entsprechende Abarbeitungsroutinen aus
- Solange bis der letzte Befehl (XCOMPLETE) erreicht wird

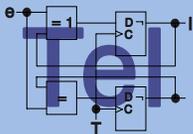




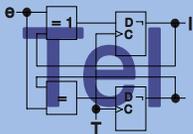
- Der Algorithmus lässt sich grob in drei Abarbeitungsschritte unterteilen
 - Ausgehend vom aktuellen Knoten den nächsten Knoten aufsuchen
 - Auswertung der Knoten
 - Den Rekonfigurationsprozess initiieren

Zusammenfassung und Ausblick

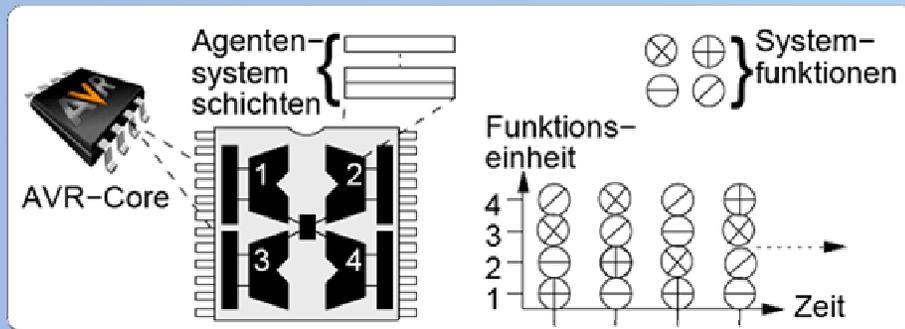
- Rekonfiguration: Implementation eines Rekonfigurationsprozesses über die JTAG-Schnittstelle
 - Scheduling: Umsetzung der Selbstorganisation über Sequenzgraphen
 - FTP-Download: als Erweiterung des Scheduling können Konfigurations- und SGU-Dateien von einem FTP-Server heruntergeladen werden
 - Verbindung: Entwicklung eines Protokolls für die Übertragung des internen Zustands des FPGAs
 - FPGA-Entwurf: zur Demonstration der Anpassungsfähigkeit des Agentensystems
-
- Verbund von Agentensystemen
 - Die „intelligente“ Erzeugung von Sequenzgraphen
 - Weiterentwicklung der Rekonfiguration
 - Einsatz leistungsfähigerer Hardware



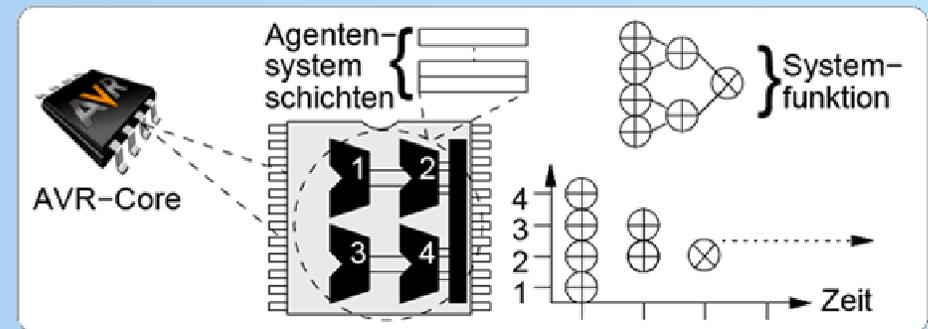
- Agenten-HW kann spezielle Verarbeitungseinheiten mit spezifisch festgelegter Funktion oder aber universelle Recheneinheiten wie beispielsweise Prozessorkerne enthalten
- dynamische Abbildung der Agentenfunktionalität, Systemschichten und der „Self-x“-Eigenschaften auf diese speziellen Verarbeitungseinheiten und Prozessorkerne unter Verwendung der rekonfigurierbaren HW
- verschiedene Möglichkeiten der funktionalen Verteilung/Lastverteilung realisierbar
 - Agentenfunktionalität wird ausschließlich durch einzelne voneinander unabhängig arbeitende Prozessorkerne realisiert
 - Lösung einer spezifischen Agentenfunktion durch mehrere Prozessorkerne
 - Abspaltung eines Prozessorkerns für Steuerungsaufgaben
 - parallele Realisierung von Steuerfunktionalität und Datenstromverarbeitung durch Integration aufgabenspezifischer Funktionseinheiten



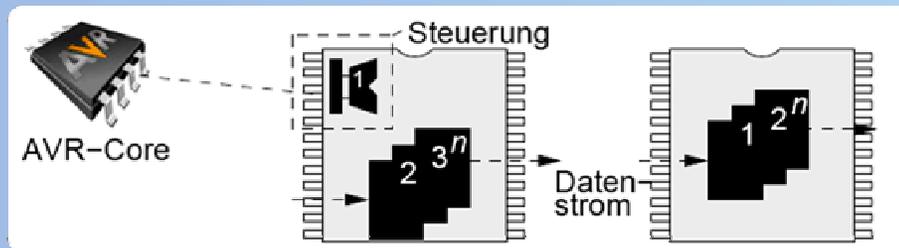
Systemadaption auf rekonfigurierbarer HW



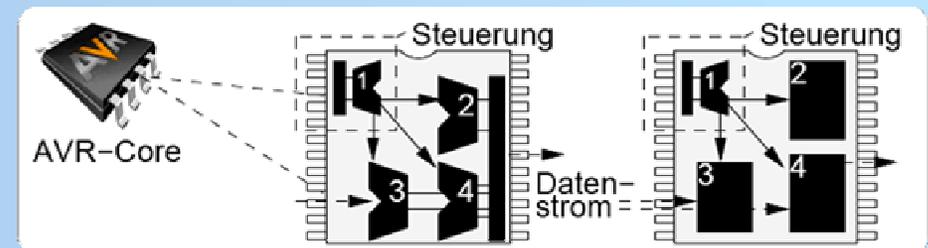
a) Agentenfunktionalität wird ausschließlich durch einzelne voneinander unabhängig arbeitende Prozessorkerne realisiert



b) Lösung einer spezifischen Agentenfunktion durch mehrere Prozessorkerne

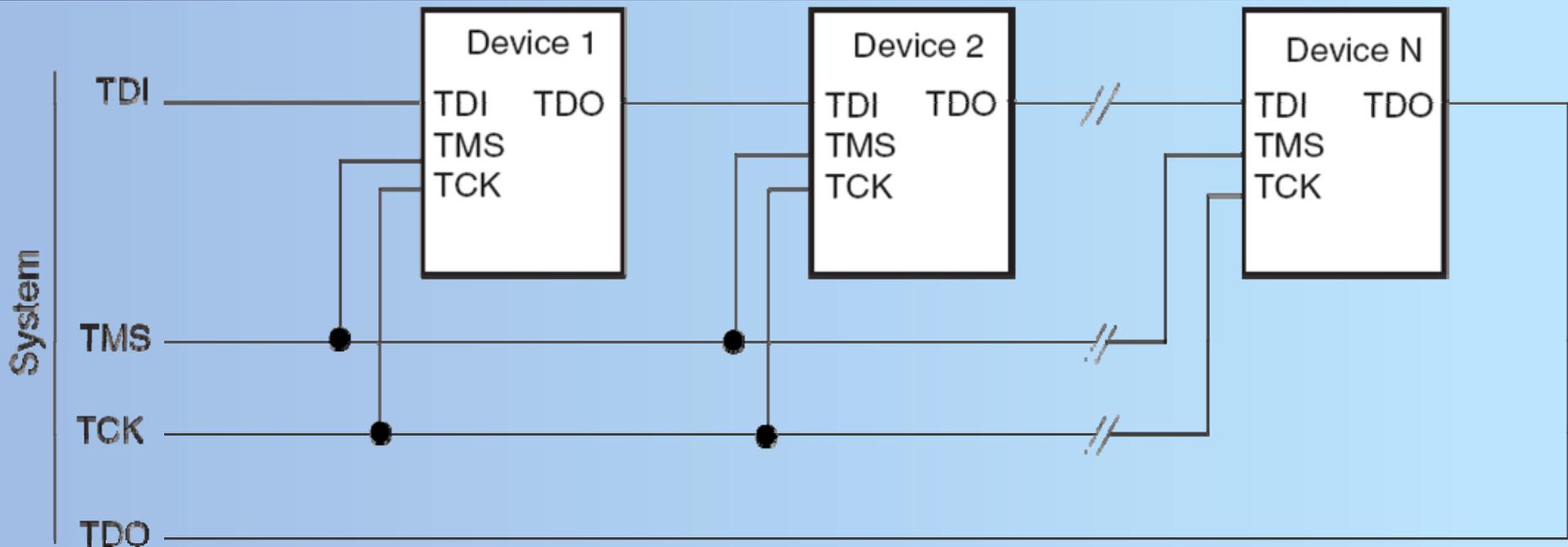


c) Abspaltung eines Prozessorkerns für Steuerungsaufgaben



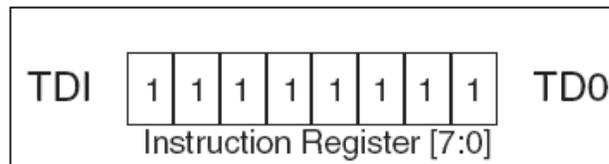
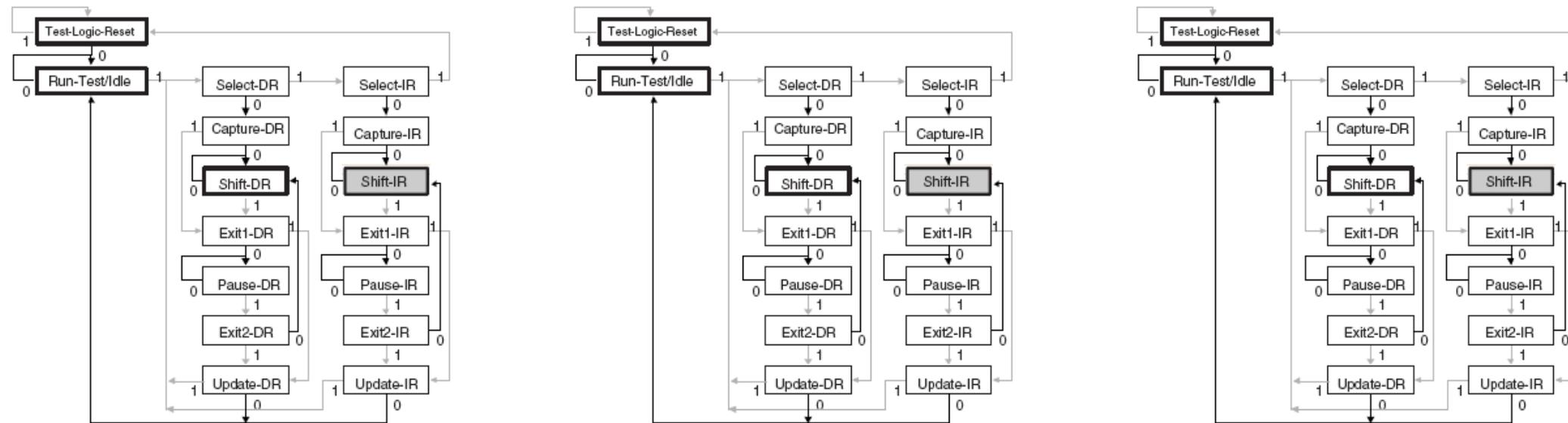
d) parallele Realisierung von Steuerungsfunktionalität und Datenstromverarbeitung durch Integration aufgabenspezifischer Funktionseinheiten

Rekonfiguration in einer Kette (1)

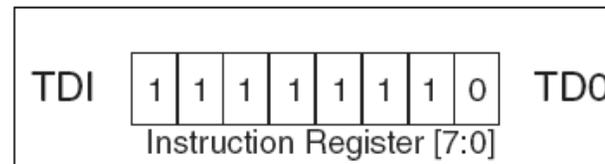


- Alle Bausteine in der Kette teilen sich TMS und TCK
- Das TDI-Signal ist mit dem TDI-Eingang des 1. Bausteins verbunden
- Das TDO-Signal des 1. Bausteins ist an den TDI-Eingang des 2. Bausteins angeschlossen, usw.
- Wegen des gemeinsam genutztem TMS-Signals befinden sich alle Bausteine im gleichem Zustand des TAP-Controllers

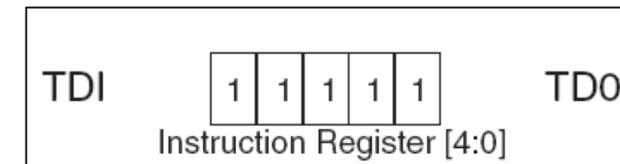
Rekonfiguration in einer Kette (2)



XC18V02



XC9572XL



XCV150

- Auswahl eines Baustein indem die anderen Bausteine einen Bypass-Befehl erhalten (alles „1“ im Befehlsregister)
- Das Datenregister wird mit dem 1-Bit langen Bypass-Register verknüpft
- SVF- bzw. XSVF-Dateien müssen dies berücksichtigen