

Embedded Machine on FPGA

A Reconfigurable System based on Kahn Process Networks

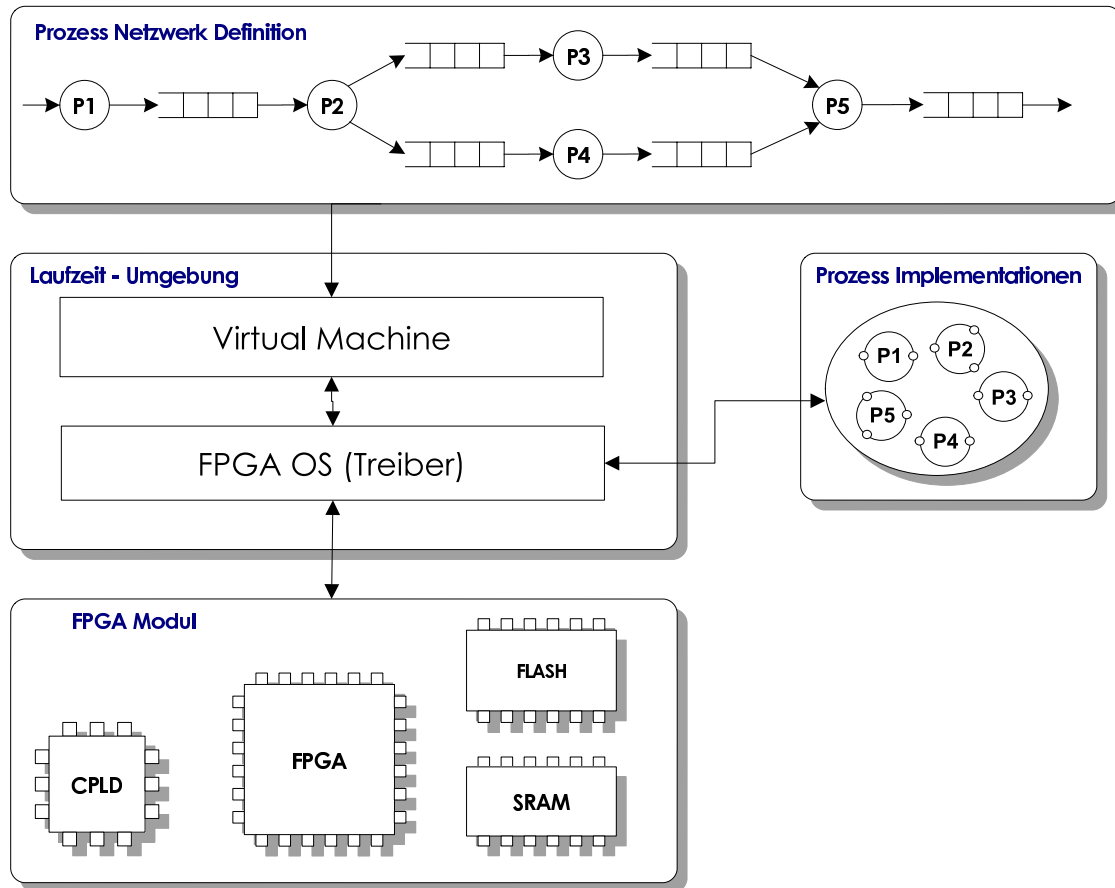
Master's Thesis
Roman Plessl

Betreuung: Matthias Dyer
Leitung: Prof. Dr. Lothar Thiele

Inhalt

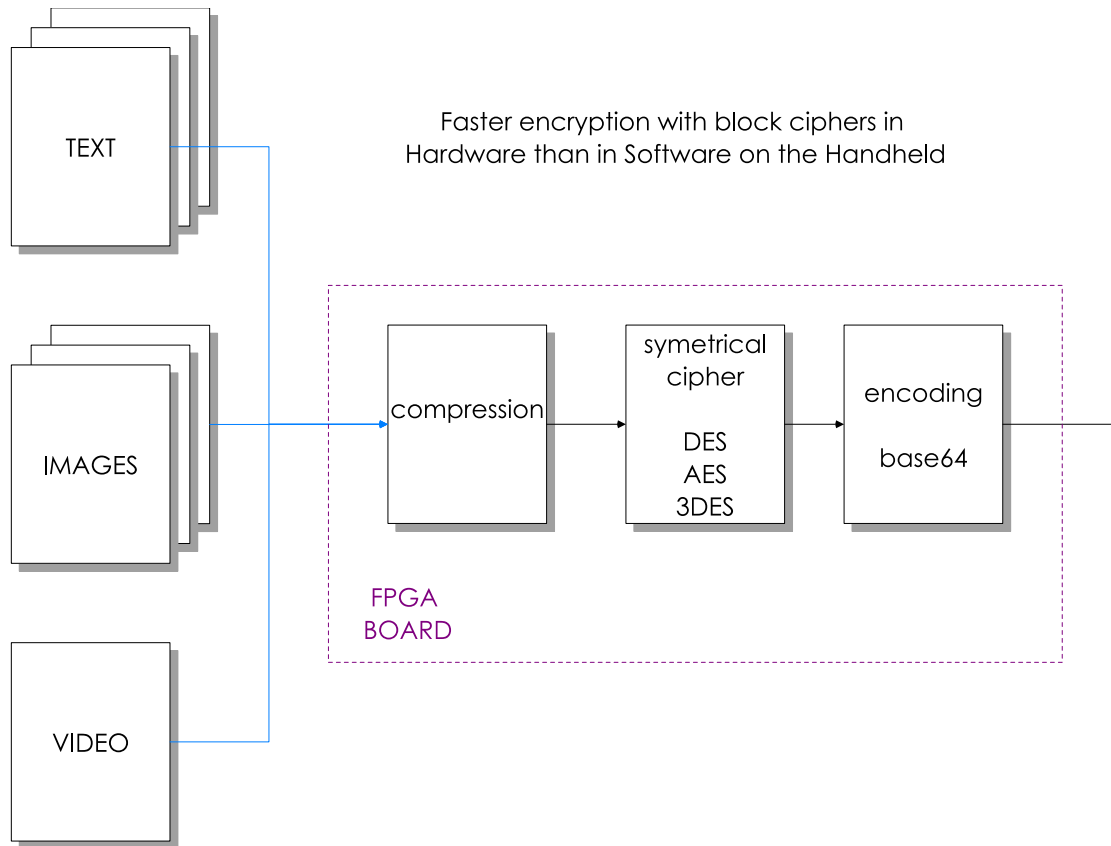
- Problemstellung und Übersicht
- Ziel der Arbeit und Anwendungen des Ansatzes
- Plattform
- Abbildung des Systems und Systemübersicht
 - Slotmanager - Ausführungsumgebung
 - Loader auf FPGA und CPLD
 - Speichern von internen Zuständen
- Resultate und Erfahrungen
- Ausblick
- Demonstration und Fragen

Problemstellung und Übersicht



- Motivation
- Kahn Prozess Netzwerk interpretierte «*coordination language*»
- Prozess Implementationen *synthetisierte Tasks*
- Laufzeit - Umgebung
 - Virtual Machine (Interpreter für das Kahn Prozess Netzwerk)
 - FPGA OS (Treiber für das FPGA Modul)
- FPGA - Modul

Anwendungen des Ansatzes



- Einsatzgebiete von Kahn Prozess Netzwerken
 - «Streaming Applications» (z.B. Audio, Video, Verschlüsselung)
 - Datafluss orientierte Probleme
- Performante Rechnungen welche in Hardware deutlich schneller ablaufen als in Software
 - symetrische Verschlüsselung (AES, DES)
 - Filter (Bildverarbeitung)
- Parallelisierbare Aufgaben
 - Mustererkennung (Bit-Muster)
 - Filter (FIR-Filter)

Ziel der Arbeit

- Entwicklung eines Frameworks zur Ausführung von Hardware Tasks eines Kahn Prozesses Netzwerkes auf einem FPGA
- Erstellung eines «Ein-Task-Slot» Systems mit definierten Schnittstellen für einen Kahn Prozess
- Abbildung von Input- und Output Ports eines Prozesses auf vorhandene beschränkte Speicherressourcen.
- Ausarbeitung einer Kommunikation mit dem Scheduler welcher in Software läuft
- Entwicklung eines Mechanismus zum Speichern und Rückladen von Prozess Kontexten

Plattform:

- IPAQ (unter Windows CE laufend) und BTNODEFPGA

Tools:

- Xilinx ISE
- Synplicitys Synplify Pro
- Microsoft Embedded Visual C++
- Perl

Hardware - BTNODEFPGA



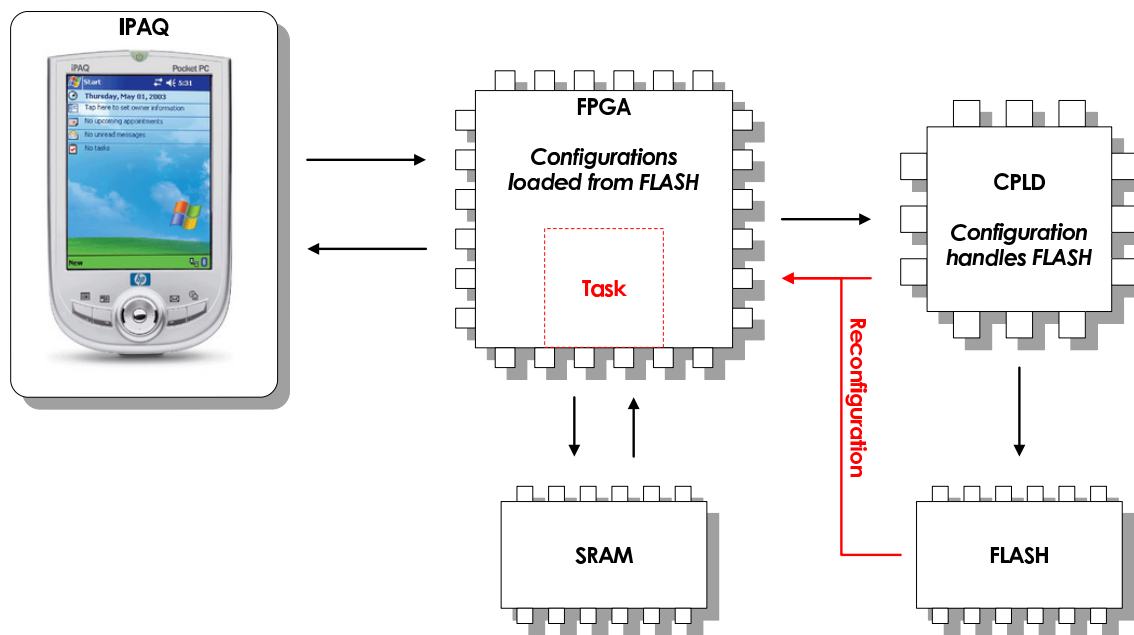
- FPGA: *Xilinx Spartan II*, 200k gates
- CPLD: *Xilinx Coolrunner*, 384 Macrozellen
- FLASH: 8 MBit, 16 x 64 kByte Sektoren
- SRAM: 4 MBit, 256k x 16 Bit
- Design entworfen für mobilen Einsatz
 - 18 MHz Clock
 - Geringe Leistungsaufnahme
«low power» Design
 - entsprechend verwendete Bauelemente
- I/O Pins am FPGA (IPAQ)
- I/O Pins am CPLD (BTNODE)

Hardware - IPAQ und «IPAQ Bridge»



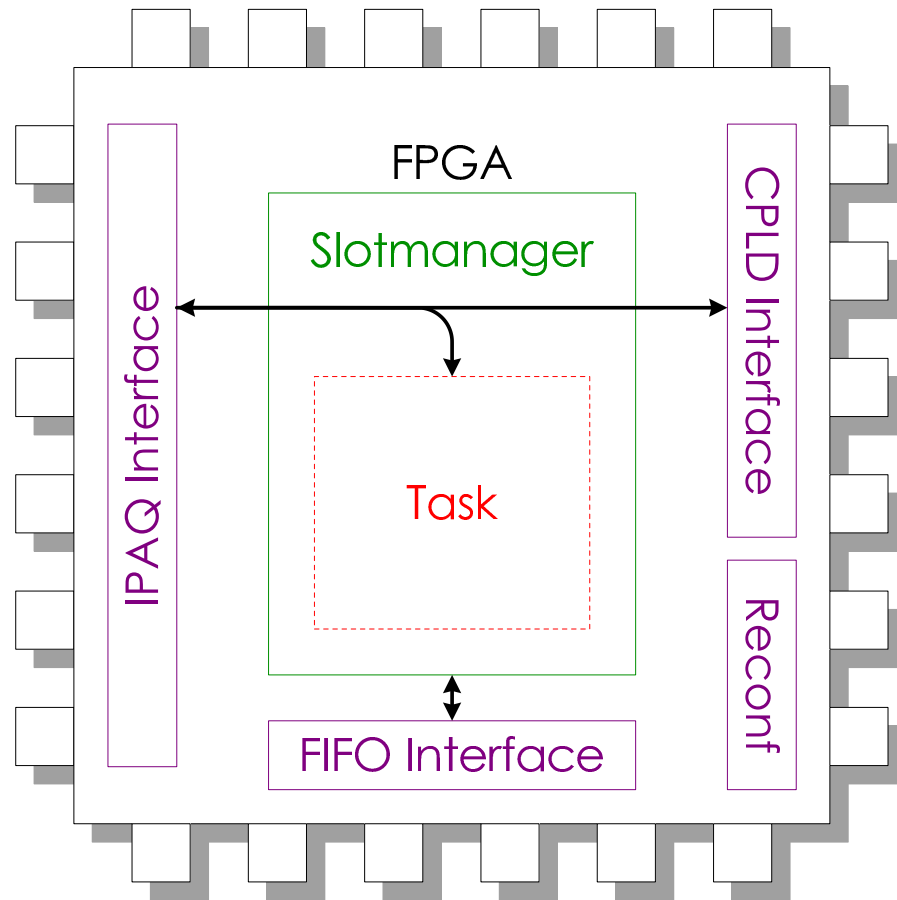
- IPAQ unter Windows CE laufend
- «IPAQ Bridge»
- Erweiterung verbunden mit Expansion Slot
- Asynchrones SRAM Interface
- Zugriff mittels Memory Mapped I/O Speicherregion

Abbildung des Systems und Systemübersicht



- Framework
- Task - Slot
- FIFO Implementierung in SRAM
- CPLD für FLASH Steuerung
- FLASH ist Speicher für Bitstreams
- Kommunikation von Events zum und vom IPAQ (Scheduling)
- Bitstream Ladeprozess

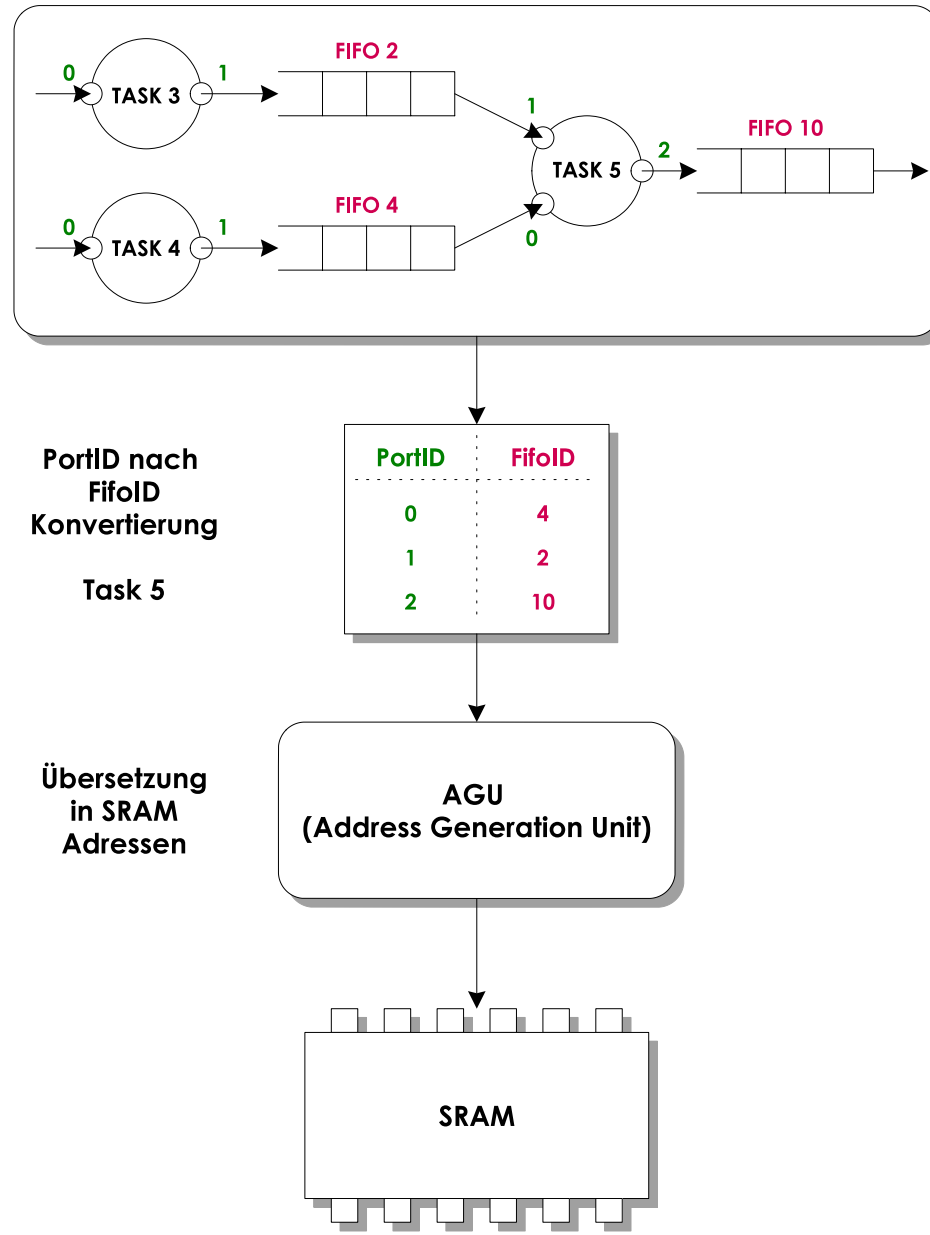
Slotmanager - Ausführungsumgebung



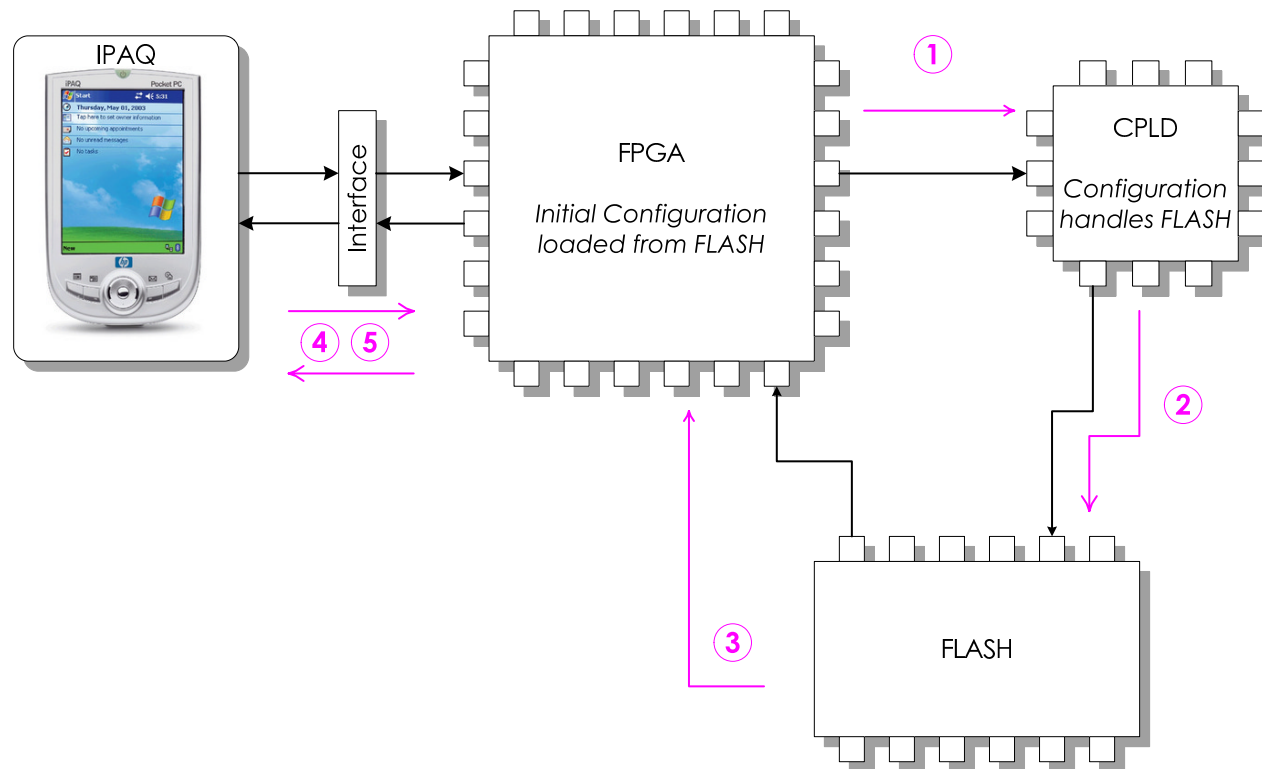
- Ausführungsumgebung für einen Task
- IPAQ Kommunikation über 24-Bit Interface
- 16-Bit FIFO Interface
- PortID → FifoID → SRAM Adressen
- Kommunikation IPAQ → FLASH bleibt bestehen
- Daten des Slotmanagers:
 - *Verwendete Slices:*
838 / 2352 (35%)
 - *Flip-Flops:*
736

Slotmanager

Konversion PortID zu SRAM Adressen



Loader auf FPGA und CPLD

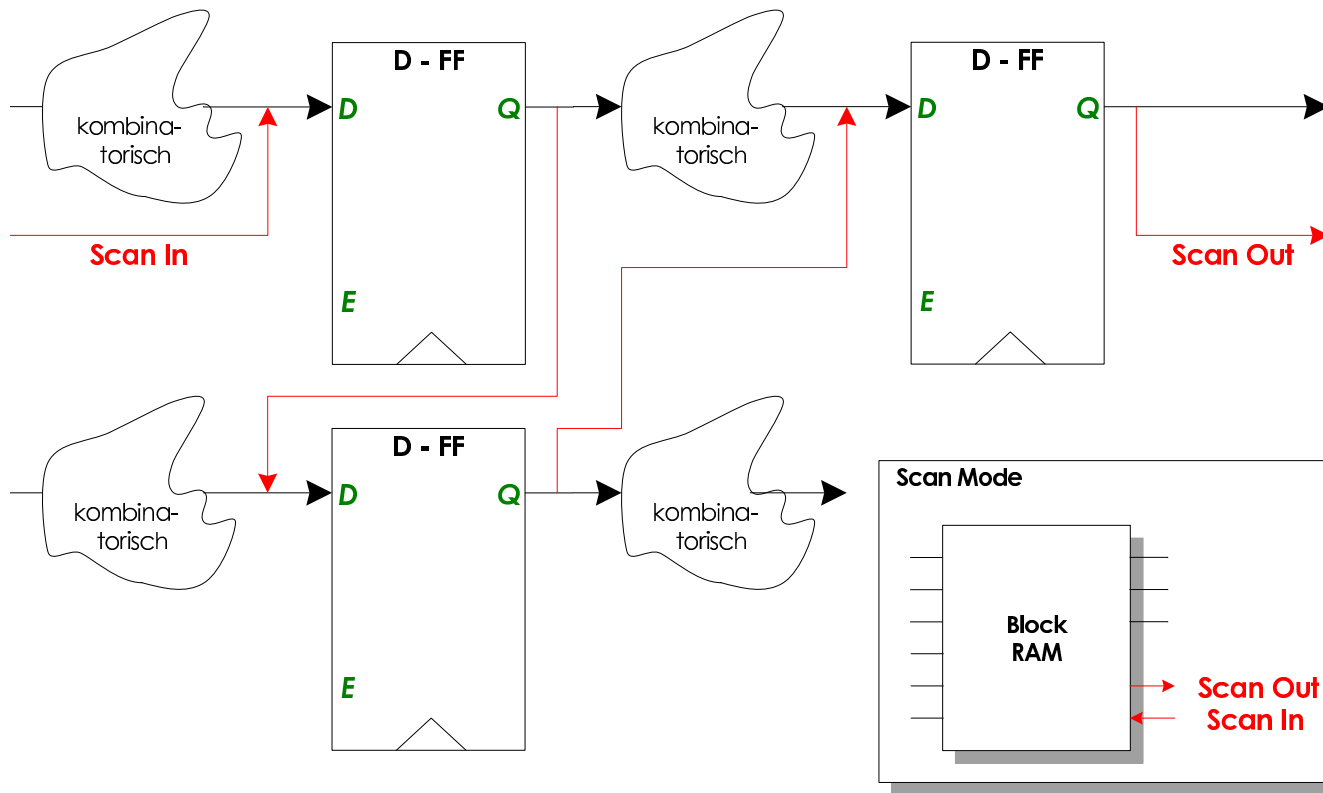


- Momentan separates Design um einen ausführbaren Task in das System zu laden.
- FLASH kann maximal 5 verschiedene Bitstreams speichern
- Reconfigurationszeiten:
FPGA → FLASH: 7.8 s
FPGA → FLASH: 1.6 s (min.)
FLASH → FPGA: 9 ms
- FPGA:
Verwendete Slices:
 267 / 2352 (11%)
- CPLD:
Verwendete Macrozellen:
 202 / 384 (53%)

Speichern von internen Zuständen - «Scan-Chain»

- Kann Prozesse besitzen einen internen Zustand (Kontext) welcher bei Task-Switch (→ Scheduler) gespeichert werden muss
- Möglichkeiten diese internen Zustände zu speichern:
 - «Readback» Funktion der Bauelemente
 - Explizites Speichern durch den Programmierer
 - «Scan-Chain»: Nachahmung eines Ansatzes aus dem VLSI-Design (Design for Test)
- Eine «Scan Chain» bildet ein grosses lineares Schiebe-Register, welches alle (gewünschten) Flip-Flops eines Tasks miteinander verbindet. Dieses Schieberegister wird vom Slotmanager aktiviert und manipuliert. Die Zustand-Bits werden seriell eingelesen, bzw. hinausgeschoben.
- Vorteil dieses Ansatzes
 - automatisierte Erstellung
 - Speicherung der Zustände im laufenden Slotmanager
 - gewisse Unabhängigkeit von der FPGA-Architektur

Speicher von internen Zuständen

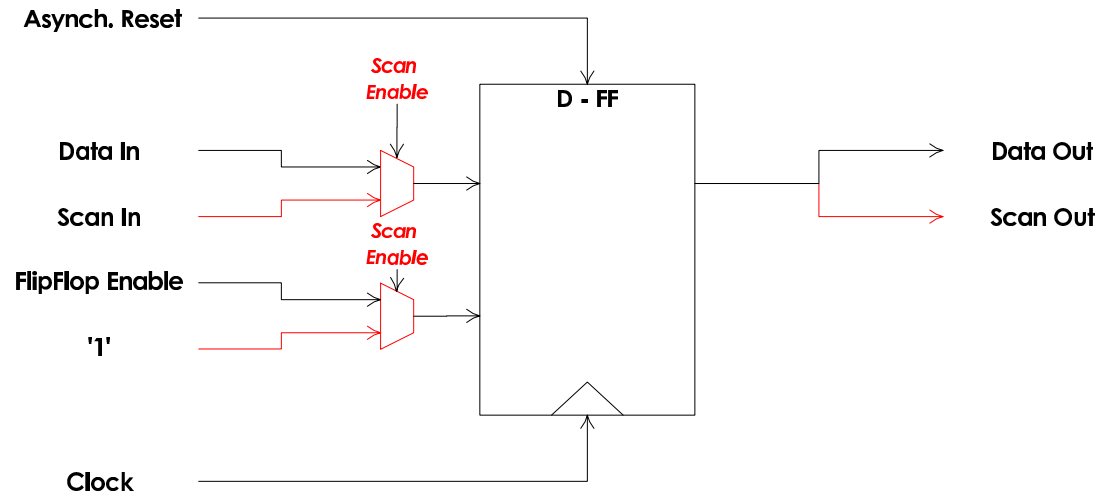


Im «Scan-Mode» werden die kombinatorischen Schaltungsteile überbrückt.

Speicherung der Zustände auf dem IPAQ – Kommunikation mit dem Task mittels eines BlockRAMs.

- IPAQ Interface
- Scan Interface

Speichern von internen Zuständen - Ersetzen der Flip-Flops



- Ersetzen der ausgewählten Zustand Flip-Flops mit ihrem «scanbaren» Äquivalent
- Flip-Flops werden eigentlich nicht ersetzt sondern mit zusätzlicher kombinatorischen Logik versehen
- Einfügen der Änderung in der Netzliste nach der Synthese
- Task Entwickler muss nichts von diesem Speichermechanismus wissen

Resultate und Erfahrungen

Resultate

- Erweiterung des IPAQs um einen «computation» Knoten
- Funktionierende Rekonfiguration des System vom IPAQ aus
- Proof-of-Concept für die Ausführung eines Kahn Prozess Netwerks auf diesem Knoten
- Konzepte und Implementationen («Scan Chain», virtuelle FIFOs, Kahn Task Cores)

Erfahrungen

- Designs mit mehreren Bauteilen sind anspruchsvoll im Entwurf und schwierig zu testen
- Beschränkung der verwendeten Komponenten (FPGA, CPLD)
- Vorzüge eines «brain pools»

Ausblick auf zukünftige Arbeiten

- Erweiterung auf mehrere Task-Slots mit partieller Reconfiguration
- Integration des Slotmanagers und der ausgeführten Tasks in eine Windows CE Applikation
- Arbeiten im Bereich des Scheduling von Kahn Prozess Netzwerken
- Debugging des Slotmanagers

Demonstration

FILM AB!

Fragen

Für Fragen stehe ich Ihnen gerne zur Verfügung