TECHNISCHE UNIVERSITÄT DRESDEN

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

Virtualisierung von FPGA-Ressourcen mittels partieller dynamischer Rekonfiguration für konkurrierende Nutzerdesigns

Verteidigung der Belegarbeit

Paul R. Genßler – paul.genssler@tu-dresden.de

Dresden, 3. Dezember 2015

1. Motivation
2. Implementierung
3. Ergebnisse
4. Zusammenfassung und Ausblick

Einsatzbereich von FPGAs

- **F**ield **P**rogrammable **G**ate **A**rray
- Einführung 1984 durch Xilinx
- Meist mit SRAM konfigurierte CLBs und PSMs, IOBs
- Hardmacros wie On-Chip Speicher, DSPs, PCIe
- Sehr flexibel, viele spezialisierte Anwendungsgebiete
  - Netzwerkinfrastruktur
  - Biologie
  - Finanzwesen

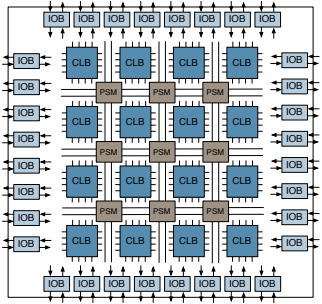


Bild: Oliver Knodel (2011). "Implementierung des Genom-Alignments auf modernen hochparallelen Plattformen". Diplomarbeit. TU Dresden

1 Motivation

Cloud Computing

Definition

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Peter Mell und Timothy Grance (2011). *The NIST Definition of Cloud Computing*.

FPGAs in der Cloud

- Einfacher und schneller Zugang
  - Keine (physische) Einrichtung, niedrige Einstiegshürde
  - Kürzere Turnaround Time dank Wiederverwendung
- Die Ressource FPGA
  - Mehr Rechenkraft für Datencenter
  - Skalierung möglich
- Vision: Transparente Beschleunigung von Nutzerprogrammen durch High Level Synthese

1 Motivation

Reconfigurable Common Cloud Computing Environment

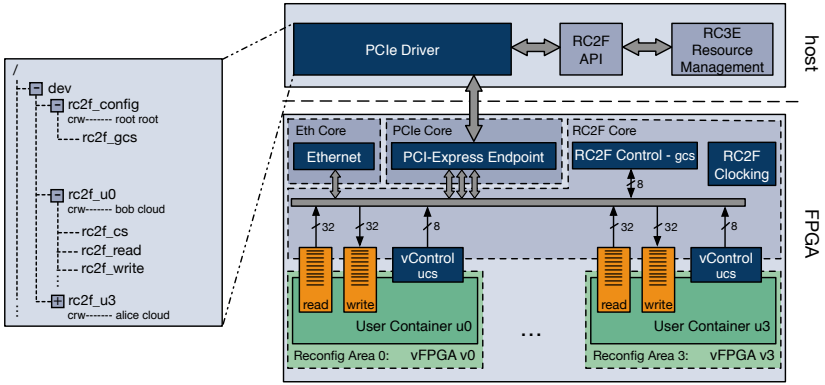


Bild: Oliver Knodel und Rainer G Spallek (2015). "Computing Framework for Dynamic Integration of Reconfigurable Resources in a Cloud". In: *Euromicro Conference on Digital System Design (DSD)*. IEEE, S. 337–344

1 Motivation

Reconfigurable Common Cloud Computing Environment

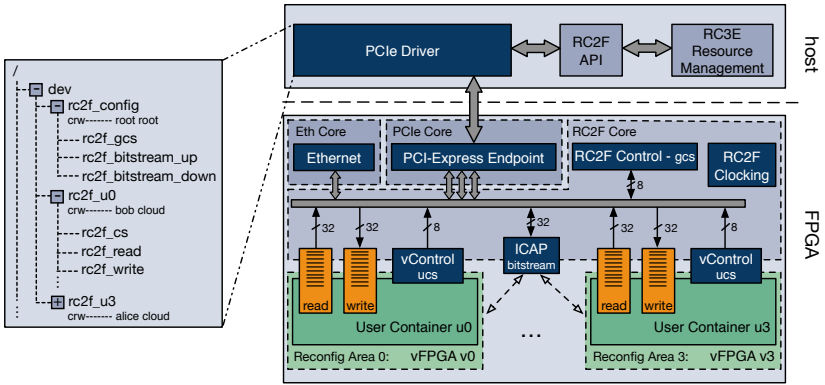


Bild: Oliver Knodel und Rainer G Spallek (2015). "Computing Framework for Dynamic Integration of Reconfigurable Resources in a Cloud". In: *Euromicro Conference on Digital System Design (DSD)*. IEEE, S. 337–344

Andere Arbeiten

Implementation	Bit Breite	Frequenz	Daten Quelle	Controller
Xilinx (PLB)	32, 64, 128[1]	100	PLB Bus	MicroBlaze
Xilinx (AXI)	32[1]	100	AXI Lite Bus	MicroBlaze
Blodget u. a., 2003	8[2]	50[2]	OPB Bus	MicroBlaze
Claus u. a., 2008	32	100	DDR2	FSM
Liu u. a., 2009	32	100	Block RAM	HWICAP
Vatsolakis u. a., 2014	32	125	PCIe	Host CPU & DMA
Diese Arbeit	32	100	PCIe	FSM

[1] Busbreite des Controllers
[2] Maximum auf dem Virtex II

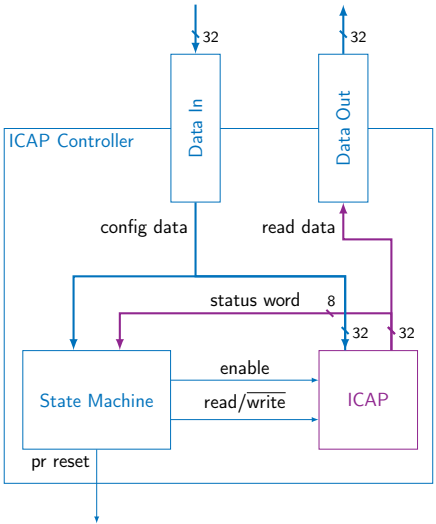
Reconfigurable Cloud Computing Framework Erweiterung

- Keine Unterstützung von mehreren unabhängigen Nutzern
  - Klare Trennung von Static und Nutzer Design
  - Definition Schnittstelle, Aufteilung der Ressourcen
  - Automatisierung des Toolflows
- Keine schnelle (Re)konfiguration
  - Implementierung eines ICAP Controller
  - Nutzung von PCIe

2 Implementierung

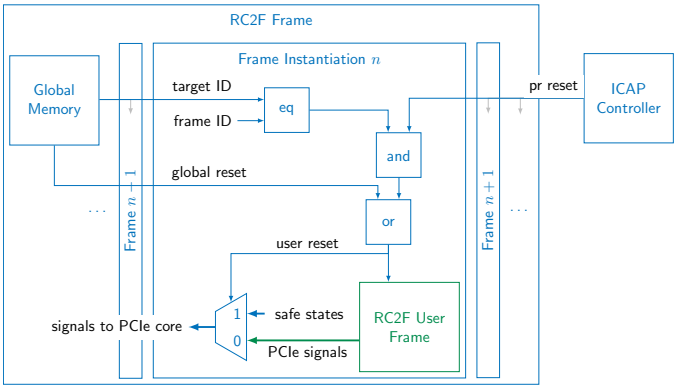
ICAP Controller

- Schnittstelle zwischen PCIe Interface und ICAP
- Steuerung implizit durch Host-Datein und ICAP Zustand
- Signal *pr reset* für Entkopplung der Nutzerdesigns
- Erweiterung zum Auslesen der Konfiguration möglich



2 Implementierung

Entkopplung eines Nutzer Designs

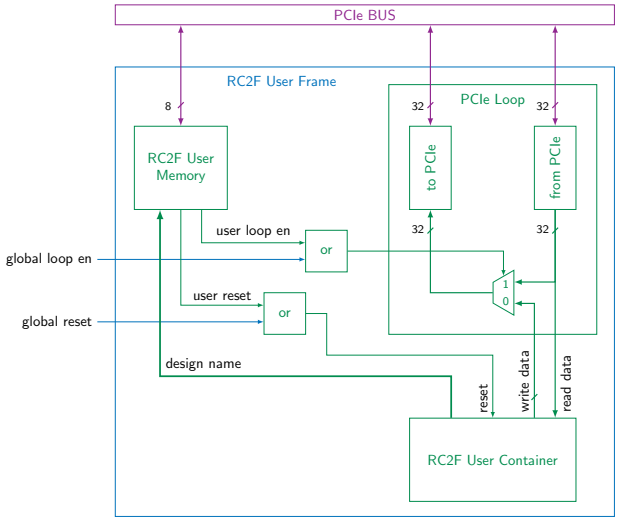


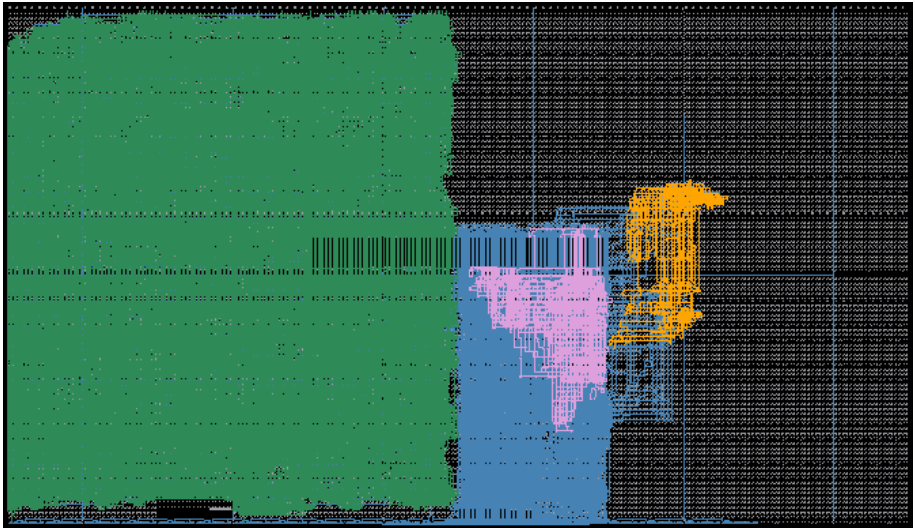
Definierte Werte der PCIe Schnittstelle während der Rekonfiguration / Reset

2 Implementierung

RC2F User Frame

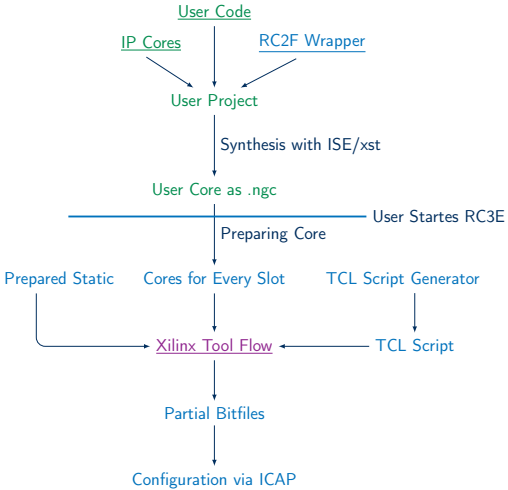
- Fest definierte Schnittstelle zum Gesamtdesign
- Einstiegspunkt für erfahrene Nutzer
- RC2F User Container mit einfacherem Interface





Erweiterer RC2F Flow

- Vereinfachung des komplexen PR Flow
- Nur ein Kommando zum Starten nötig
- Nutzer entwickelt unabhängig vom RC2F Framework
- Zeitersparnis durch Wiederverwendung möglich



3 Ergebnisse

Kompression des Bitfiles

- Verkleinerung des Bitfiles auf Kosten der Laufzeit
- Turnaround Zeit meist nicht so wichtig
- Einfache Aktivierung durch "-c" für bitgen
- Alle Wert für einen Virtex 6 XC6VLX240T

Auslastung	Niedrig	Hoch	Sehr Niedrig	Hoch
Bitfile Größe	8,81 MiB	8,81 MiB	4,15 MiB	3,43 MiB
Komprimiert	5,07 MiB	6,24 MiB	0,48 MiB	2,59 MiB
Einsparung	42,47 %	29,16 %	88,52 %	24,57 %
Verlängerung Flow	9,67 %	1,13 %	8,21 %	−0,09 %

Speedup von PCIe und ICAP gegenüber JTAG

$$\begin{aligned} \text{JTAG} &: 1\,\text{bit} * 12\,\text{MHz} = 1{,}5\,\text{MB/s} \\ \text{PCIe und ICAP} &: 32\,\text{bit} * 100\,\text{MHz} = 400{,}0\,\text{MB/s} \end{aligned}$$

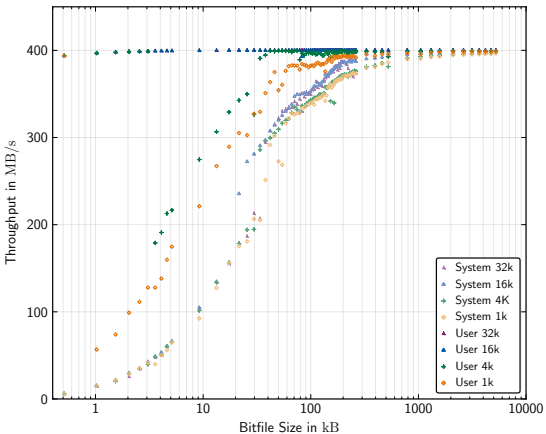
	JTAG	ICAP		Speedup	
Bitfile Size	System	System	User	System	User
0,5 MB	0,16 MB/s	386,8 MB/s	399,9 MB/s	2461,97	2545,87
1,8 MB	0,32 MB/s	396,5 MB/s	400,0 MB/s	1256,34	1267,34
5,9 MB	0,52 MB/s	398,5 MB/s	400,0 MB/s	769,45	772,42

- JTAG Overhead → nur 35 % des theoretischen Datadurchsatzes
- PCIe und ICAP fast frei von extra Steuersignalen → über 99 %

3 Ergebnisse

Datendurchsatz mit dedizierten PCIe Bus

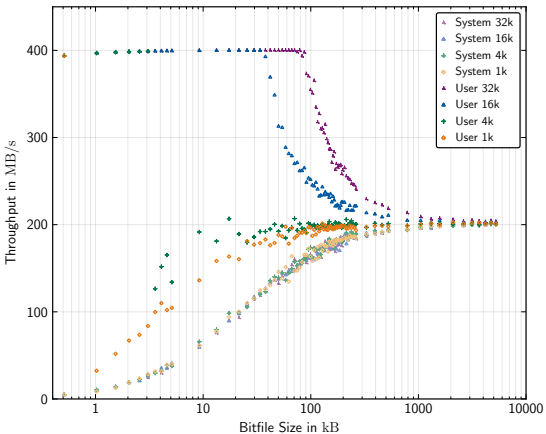
- Systemzeit: Datei auf Host ist geöffnet
- Userzeit: *pr reset* Signal liegt an
- Einfluss der FIFO Größe stärker bei kleinen Bitfiles
- Je größer desto besser



3 Ergebnisse

Datendurchsatz mit geteiltem PCIe Bus

- Rekonfiguration des ersten Nutzerdesigns
- Auslastung des Bus durch Berechnung des zweiten Nutzeres
- Signifikater Einfluss der FIFO Größe auf kleine Bitfiles
- Vorteil nur bei Bitfiles bis 500 kB



3 Ergebnisse

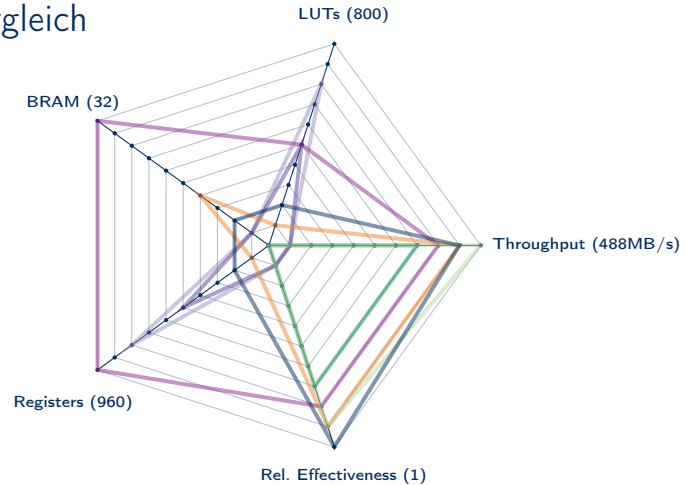
ICAP Controller Vergleich

Implementation	Datendurchsatz MB/s	Effektivität %	Daten Quelle	Registers	LUTs	BRAMs
Xilinx (PLB)	8,48	2,12	NA	746	799	1
Xilinx (AXI)	9,10	2,28	NA	477	502	1
Claus u. a., 2008	295,40	73,85	DDR2	NA	NA	NA
Liu u. a., 2009	371,40	92,85	BRAM	963	469	32
Vipin und Fahmy, 2012	399,80	99,95	DDR3	74	38	8
Vatsolakis u. a., 2014	488,00	97,60	PCIe	NA	NA	NA
Diese Arbeit	398,46	99,61	PCIe	251	177	2

3 Ergebnisse

ICAP Controller Vergleich

- Xilinx (PLB)
- Xilinx (AXI)
- Claus u. a., 2008
- Liu u. a., 2009
- Vipin und Fahmy, 2012
- Vatsolakis u. a., 2014
- Diese Arbeit



Einfluss auf Turnaround Zeit

- Einfluss von Wiederverwendung auf map Phase erkennbar
- Allgemein längere Laufzeit

Design	v6_u2_L8BT			v6_u2_BTBS10		
Flow	Usual	DPR	Increase	Usual	DPR	Increase
overall	215 s	1084 s	4,04	2044 s	2888 s	0,41
ngdbuild	12 s	20 s	0,67	107 s	122 s	0,14
map	94 s	352 s	2,74	1563 s	980 s	−0,37
par	58 s	267 s	3,60	228 s	962 s	3,22
bitgen	51 s	445 s	7,73	150 s	824 s	4,49

Zusammenfassung

- Großes Potenzial von FPGAs in der Cloud
- Aufteilung des FPGA auf mehrere unabhängige Nutzer
- Leicht zu nutzendes High- oder Low-Level Interface
- Speedup von bis zu 2500 gegenüber JTAG
- Maximaler Datendurchsatz des ICAP Controllers
- Längere Turnaround Zeiten

Ausblick

- Übertakten des ICAP: Hansen u. a. (2011) erreichen 550 MHz
- DDR3 Speicher Anbindung
- Anhalten des Nutzerdesigns und Transfer in andere Region
- Automatische Aufteilung des FPGAs nach Backasch u. a. (2014)

Backasch, Rico u. a. (2014). “Identifying homogenous reconfigurable regions in heterogeneous FPGAs for module relocation”. In: *ReConFigurable Computing and FPGAs (ReConFig), 2014 International Conference on*. IEEE, S. 1–6.

Blodget, Brandon, Scott McMillan und Patrick Lysaght (2003). “A lightweight approach for embedded reconfiguration of FPGAs”. In: *Design, Automation and Test in Europe Conference and Exhibition, 2003*. IEEE, S. 399–400.

Claus, Christopher u. a. (2008). “A multi-platform controller allowing for maximum dynamic partial reconfiguration throughput”. In: *Field Programmable Logic and Applications, 2008. FPL 2008. International Conference on*. IEEE, S. 535–538.

Hansen, Simen Gimle, Dirk Koch und Jim Torresen (2011). “High speed partial run-time reconfiguration using enhanced ICAP hard macro”. In: *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on*. IEEE, S. 174–180.

Liu, Ming u. a. (2009). “Run-time Partial Reconfiguration speed investigation and architectural design space exploration”. In: *Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on*, S. 498–502. DOI: 10.1109/FPL.2009.5272463.

Vatsolakis, Charalampos, Kyprianos Papadimitriou und Dionisios Pnevmatikatos (2014). “Enabling Dynamically Reconfigurable Technologies in Mid Range Computers Thro-ugh PCI Express”. In: *HiPEAC Workshop on Reconfigurable Computing (WRC), Vienna*.

Vipin, K. und S.A. Fahmy (2012). “A high speed open source controller for FPGA Partial Reconfiguration”. In: *Field-Programmable Technology (FPT), 2012 International Conference on*, S. 61–66. DOI: 10.1109/FPT.2012.6412113.

Turnaround Zeit für einen Nutzer

Design	v6_u1_BS2			v6_u1_BS10		
Flow	Standard	DPR	Increase	Standard	DPR	Increase
overall	796 s	1513 s	0,90	1960 s	5691 s	1,9
ngdbuild	39 s	46 s	0,18	102 s	119 s	0,17
map	581 s	908 s	0,56	1528 s	3200 s	1,09
par	102 s	253 s	1,48	203 s	1910 s	8,41
bitgen	74 s	306 s	3,14	127 s	462 s	2,64

Auslastung

	Registers		LUTs		BRAMs	
Design	Standard	DPR	Standard	DPR	Standard	DPR
v6_u1_BS2	23 708	23 711	18 744	19 357	16	16
v6_u1_BS10	54 375	54 378	43 180	44 066	16	16
v6_u2_L8BT	5092	5225	5175	5710	28	36
v6_u2_BTBS10	54 857	55 069	44 112	44 380	16	24
v6_u2_BS9BS10	102 765	102 771	81 402	81 044	28	28