



Verteidigung zur Studienarbeit:

Bewertung innovativer Addiererstrukturen für  
breite Operanden auf aktuellen FPGA-  
Plattformen

Markus Krause

Dresden, 19.05.2016



## Gliederung

1. Einführung
2. Addiererarten
3. Design-Strukturen
4. Testaufbau
5. Ergebnisse
6. Fazit

## Einführung

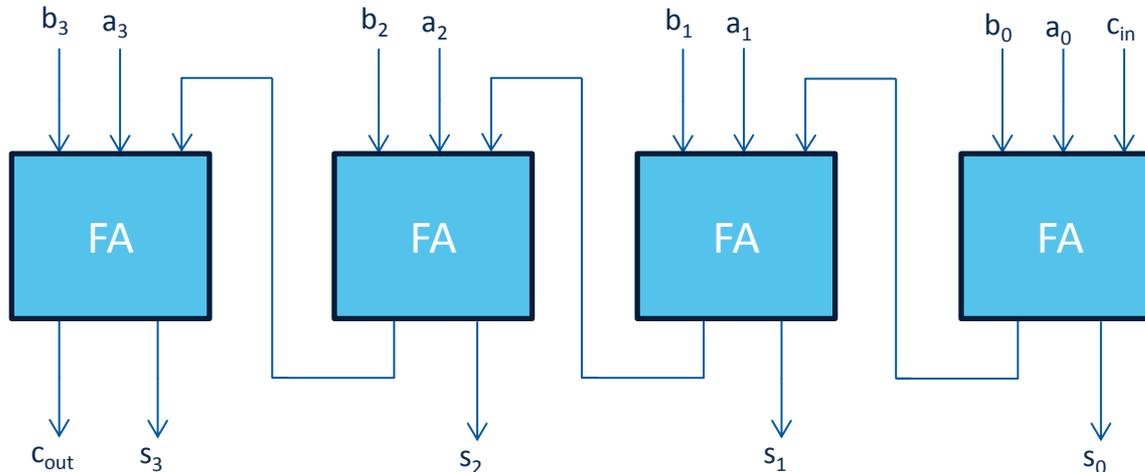
- **Problemstellung und Motivation:**
  - die binäre Wortaddition ist eine der wichtigsten arithmetischen Operationen
  - interner Übertrag wird schnell zum kritischen Pfad digitaler Schaltungen
  - Problem bei sehr breiten Operanden wie z.B. in der Kryptographie
  - daher Einsatz von Carry-Chains
  - Ziel ist der Vergleich von neuen innovativen Addiereransätzen

## Einführung

- **Aufgabenstellung:**
  1. Literaturstudium zu Realisierungen breiter Addierer
  2. Begutachtung der Syntheseresultate auf Xilinx und Altera FPGAs
  3. Entwicklung von skriptbasierten, automatischen Explorationsprozessen
  4. Quantitative Messung

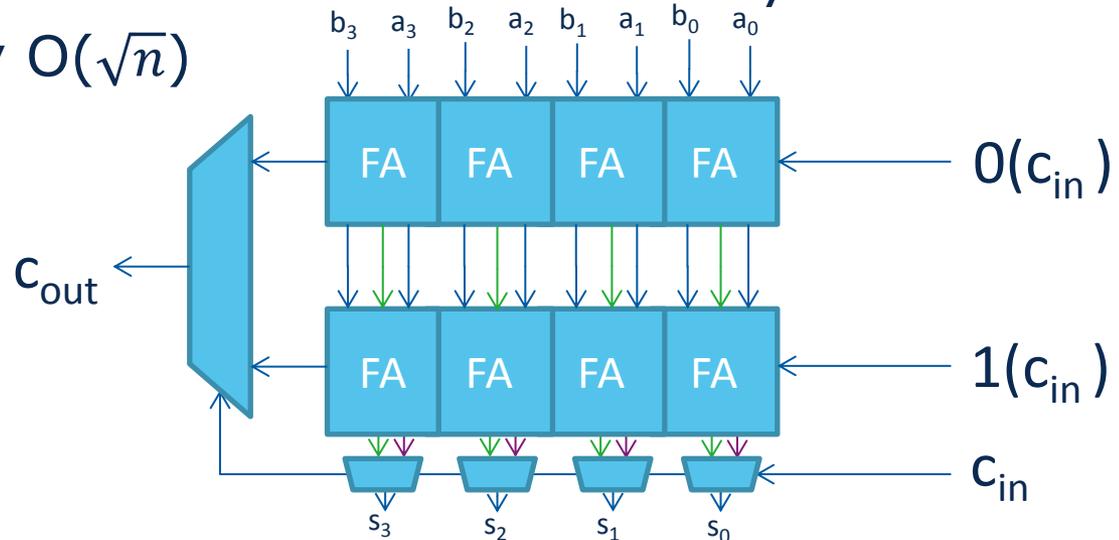
## Addiererarten

- Ripple-Carry-Adder:
  - einfacher regulärer Aufbau
  - Verkettung von 1-Bit-Volladierern
  - der Übertrag *rieselt* durch den Addierer
  - lineares Delay  $O(n)$



## Addiererarten

- Carry-Select-Adder:
  - spekulatives Berechnen beider Blocksummen und Carries
  - mit Hilfe von Multiplexern zutreffende Ergebnisse für Summe und Carry auswählen
  - Delay  $O(\sqrt{n})$

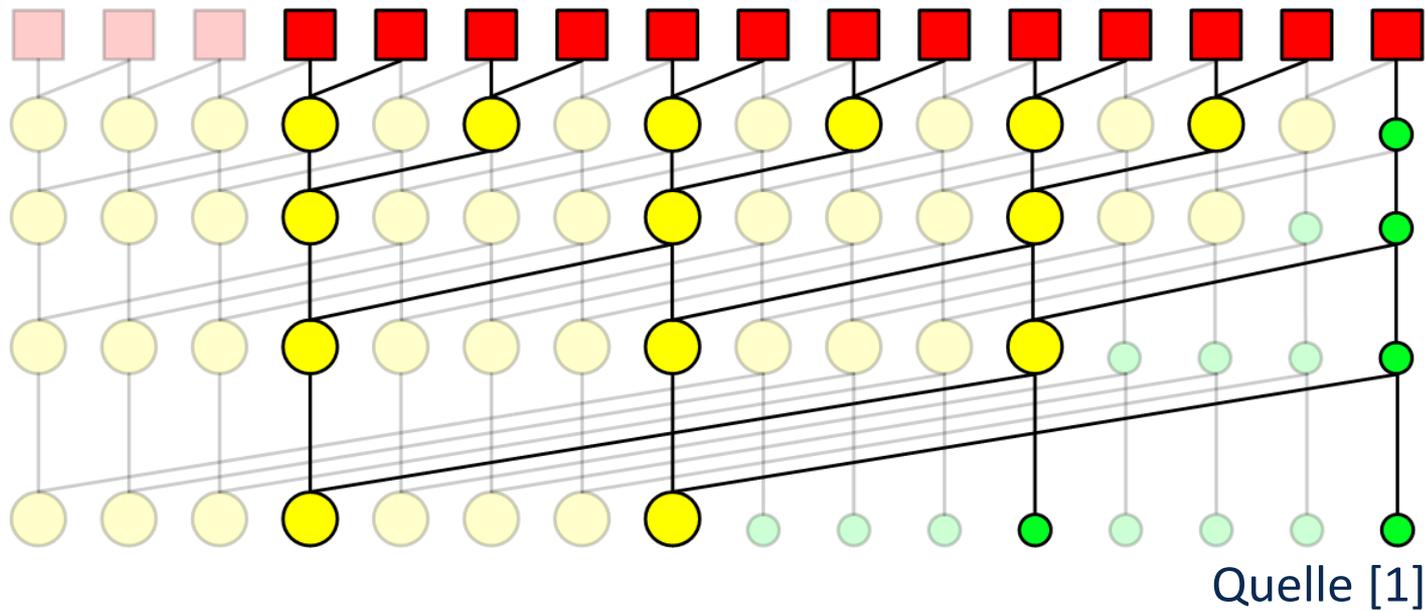


## Addiererarten

- **Parallel-Prefix-Network:**
  - assoziative Verknüpfung der einzelnen Komponenten
  - ermöglicht durch die Verallgemeinerung der Generate- und Propagate-Signale
  - in der Theorie am schnellsten mit einem Delay  $O(\log(n))$

## Addiererarten

- Parallel-Prefix-Netzwerk:
  - Kogge-Stone-PPN



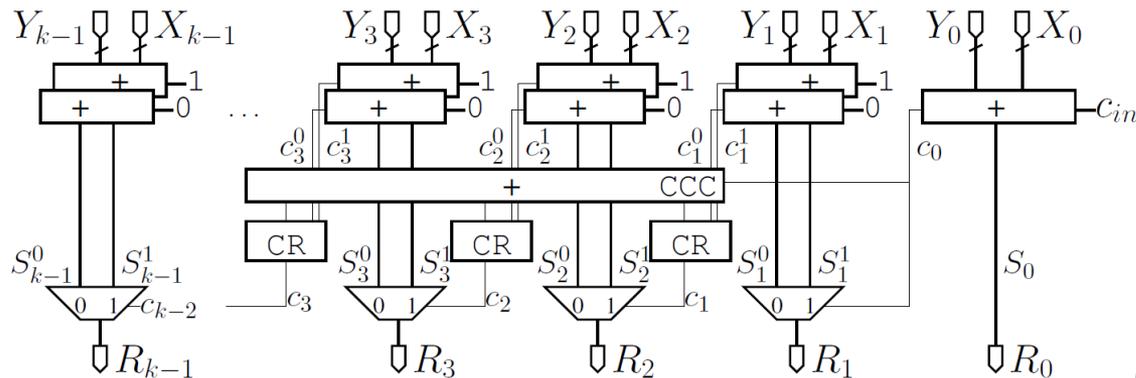
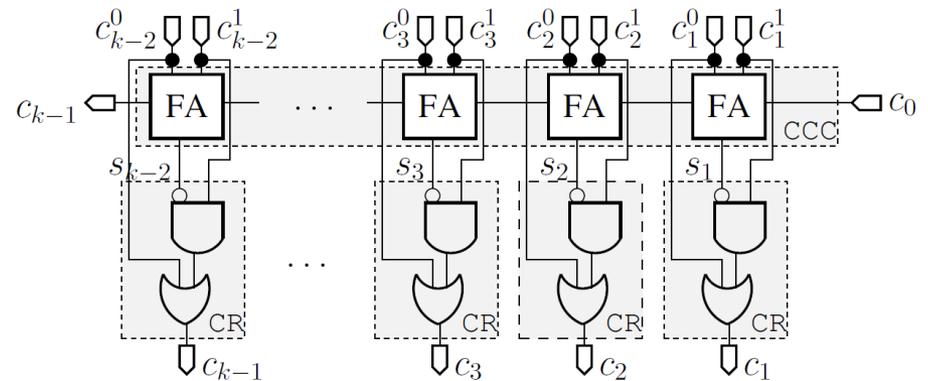
## Design-Strukturen

- **Add-Add-Multiplex:**
  - an den CSelA angelehnt
  - verwendet an Stelle der MUX-Chain einen Carry-Computation-Circuit und Carry-Recovery
  - nutzt schnelle Carry-Chains, statt langsamer General-Purpose-Leitungen

# Design-Strukturen

- Add-Add-Multiplex:

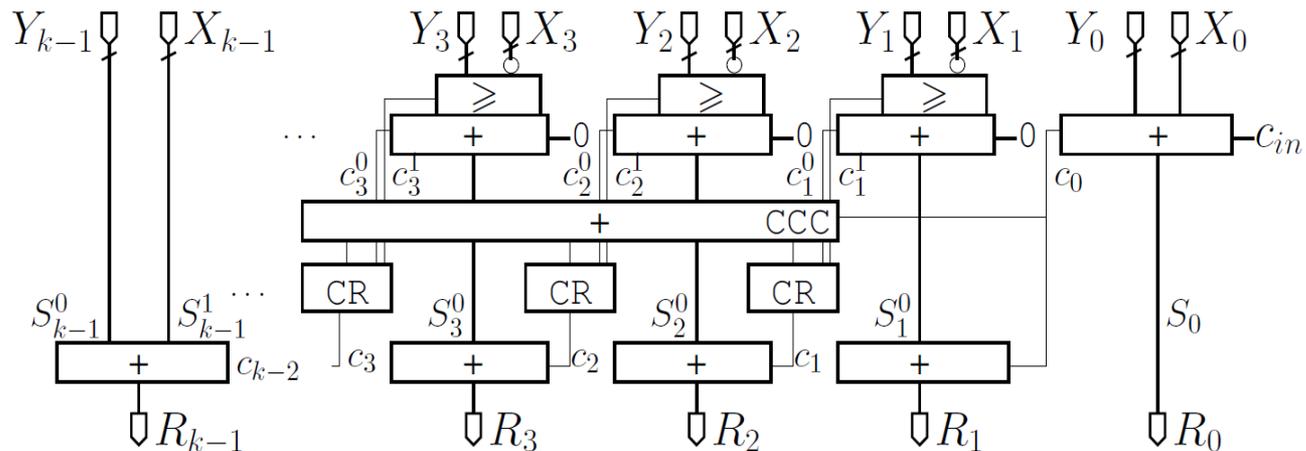
- $c_k = c_k^0 + \bar{s}_k c_k^1$



Quelle: [2]

## Design-Strukturen

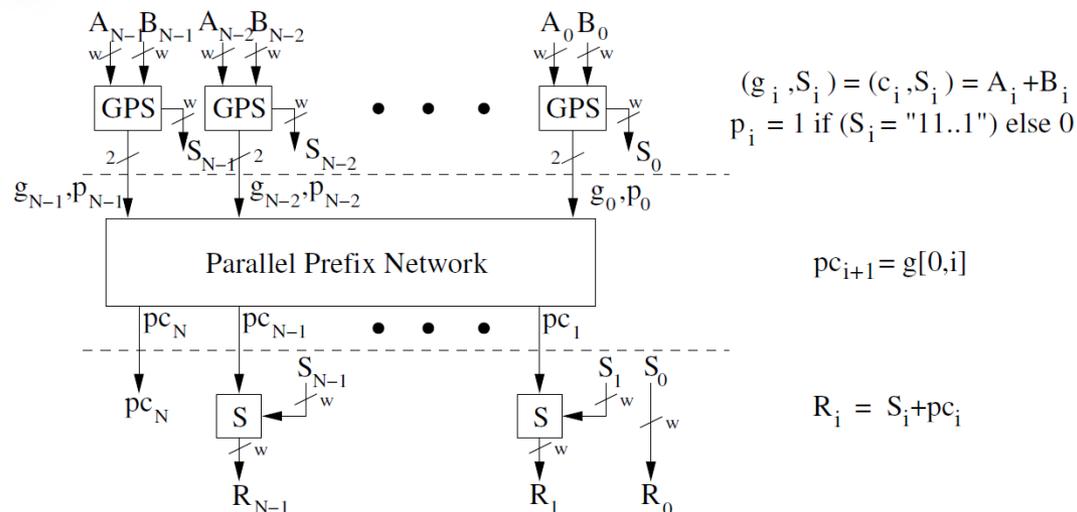
- Compare-Add-Increment:
  - sehr ähnlich zum AAM aufgebaut
  - Ein- und Ausgänge unterscheiden sich leicht
  - Vergleicher für  $c_k^1$  und Addierer für  $R_k$



Quelle [2]

## Design-Strukturen

- Novel High-Radix PPN-Adder:
  - grundsätzlicher Aufbau wie CSelA
  - verwendet Wörter statt Einzel-Bits
  - PPN Größe reduziert sich um  $N = \frac{n}{w}$



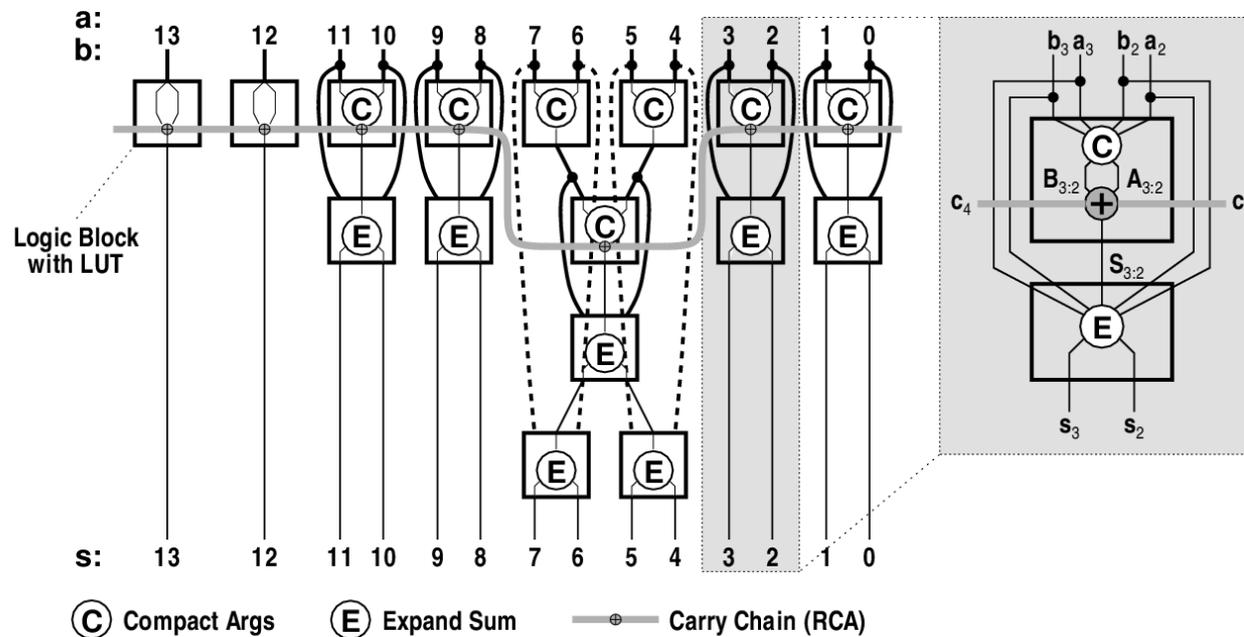
Quelle: [3]

## Design-Strukturen

- **Carry-Compact-Adder:**
  - unterscheidet sich stärker von den anderen Adder-Designs
  - ist bereits ab 50 Bit Wortbreite schneller als Standard RCA
  - nutzt Grouping
  - Compact- und Expand-Teil

# Addiererstrukturen

- Carry-Compact-Adder:



Quelle: [4]

## Testaufbau

- **Verwendete Software:**
  - ISE Design Suite 14.7
  - Vivado 2015.4
  - Quartus II 13.1 und 15.0
- Einsatz verschiedener Versionen aufgrund von fehlender Abwärtskompatibilität

## Testaufbau

- **Getestete Devices:**
  - Sowohl aktuelle Architekturen als auch ältere zur besseren Vergleichbarkeit

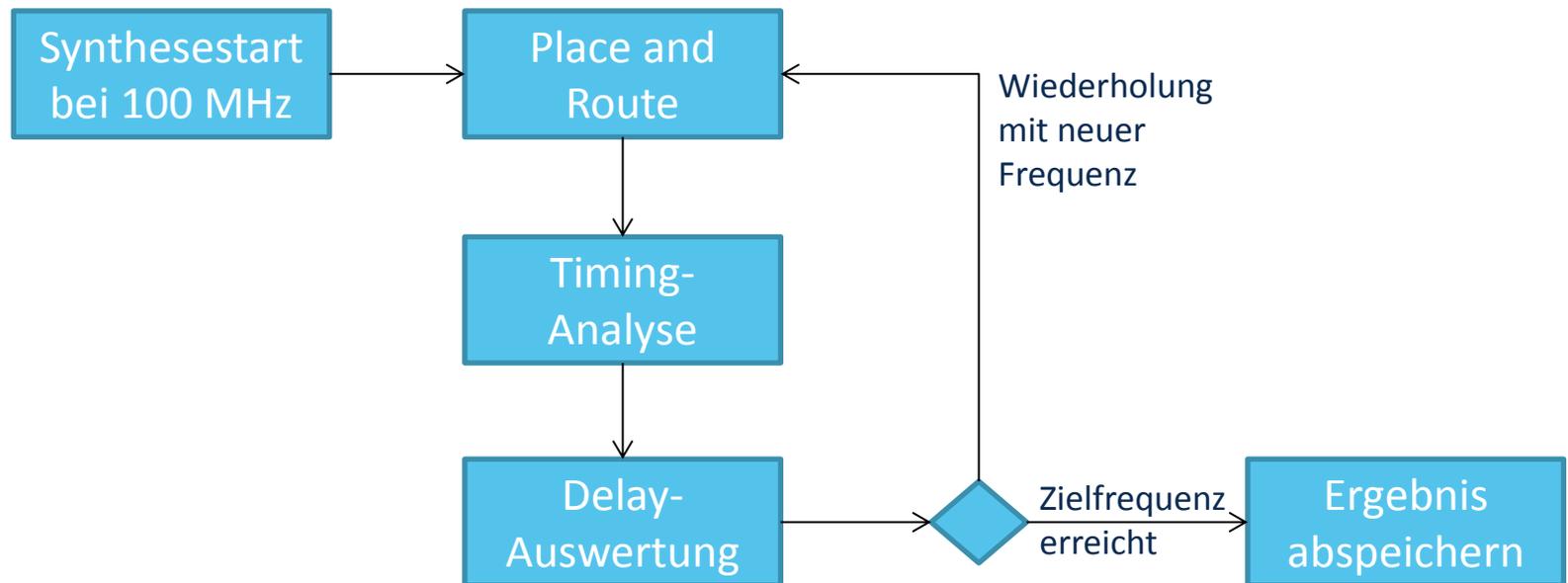
Hersteller	FPGA-Familie	Device-ID
Xilinx	Virtex 6 Virtex 7 Kintex 7	XC6VIX75T-1ff784 XC7VX485TFFG1761-2 XC7K325TFFG900-2
Altera	Cyclone V Stratix V	5CGXBC4C6F23C7 5SGXMA9K2H40C2

## Testaufbau

- Programmablauf:
  - Ausgangsbasis ist Shell-Skript
  - legt Unterordner mit jeweils angepassten Parametern an
  - 90 mögliche Kombinationen
    - Bspw.:
      - 256 Bit Breite
      - Carry-Select-Design
      - CCC als Skipping-Struktur
      - AAM als Architektur

## Testaufbau

- Programmablauf:
  - Tcl oder Shell-Skript (ISE)



## Testaufbau

- Programmablauf:
  - Besonderheiten:
    - bei ISE kein Tcl-Support
    - für Quartus angepasster Work Flow
    - bei CCA Anpassung von Parameter L
    - LUT6\_2 Komponente von Xilinx für CCA gut geeignet, unter Altera normal implementiert

## Ergebnisse

- Allgemeine Daten:
  - deutlicher Geschwindigkeitsgewinn
  - teilweise 5 mal so schnell wie RCA
  - Taktfrequenzen von um die 180 MHz bei 2048 Bit Breite
  - Area-Delay-Product trotz höherem Ressourcenverbrauch besser als bei RCA
  - neuere Devices schneller als Ältere

## Ergebnisse

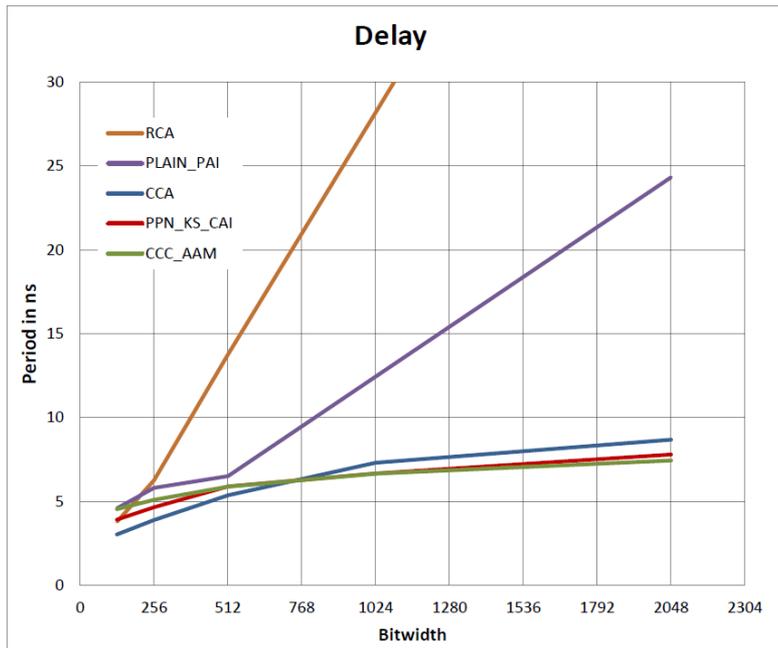
- Addierer-Vergleich:
  - RCA weit abgeschlagen
  - alle CSel-Varianten liegen nah beieinander
  - AAM und CAI am schnellsten bei den Architekturen
  - Unterschiede zwischen CCC und PPN nur marginal
  - CCA weist Besonderheit bei Flächenverbrauch auf

## Ergebnisse

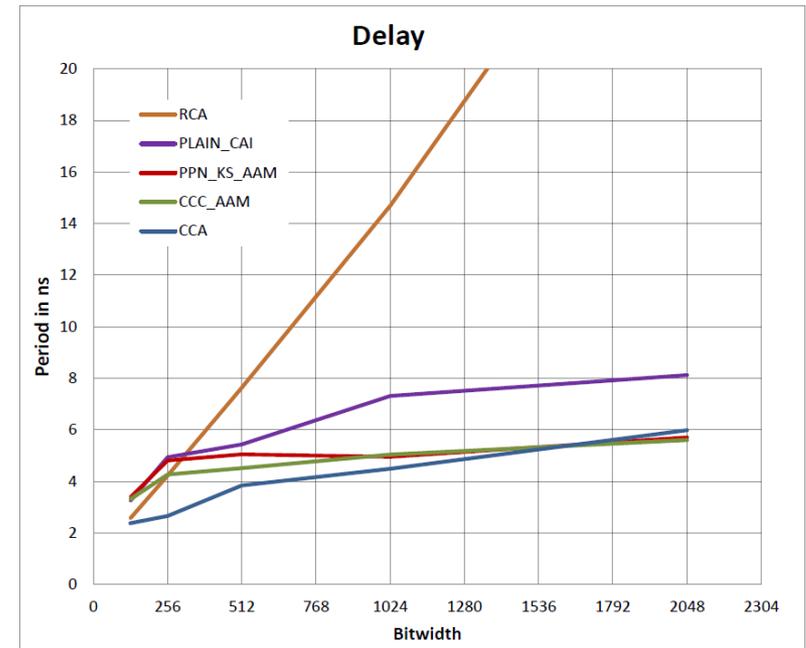
- **FPGA-Vergleich:**
  - Virtex 7 und Kintex 7 sehr ähnlich
  - Virtex 6 benötigt mehr Fläche für CselA
  - Cyclone V kann RCA für 2048 Bit nicht synthetisieren
  - bei Altera ALMs statt LUTs, schwieriger Vergleich zu Xilinx
  - RCA bei Altera deutlich schneller

# Ergebnisse

## Virtex 6

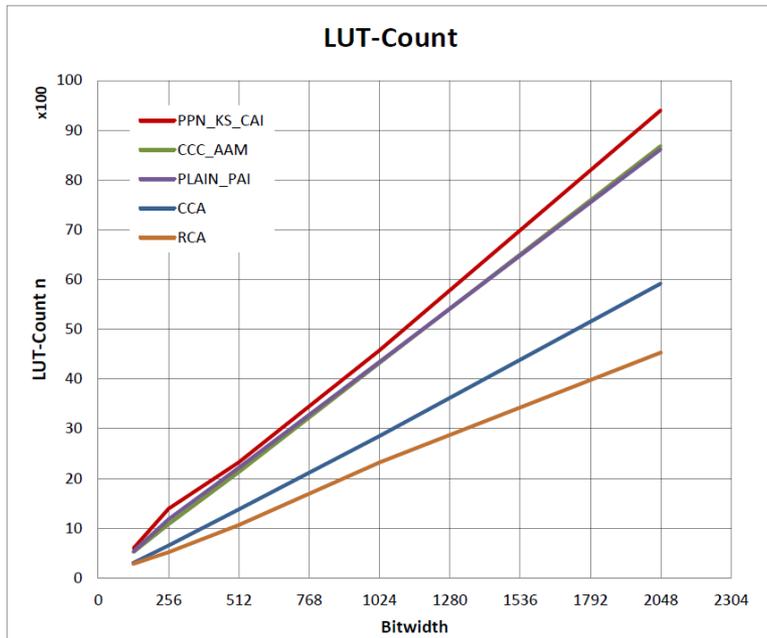


## Virtex 7

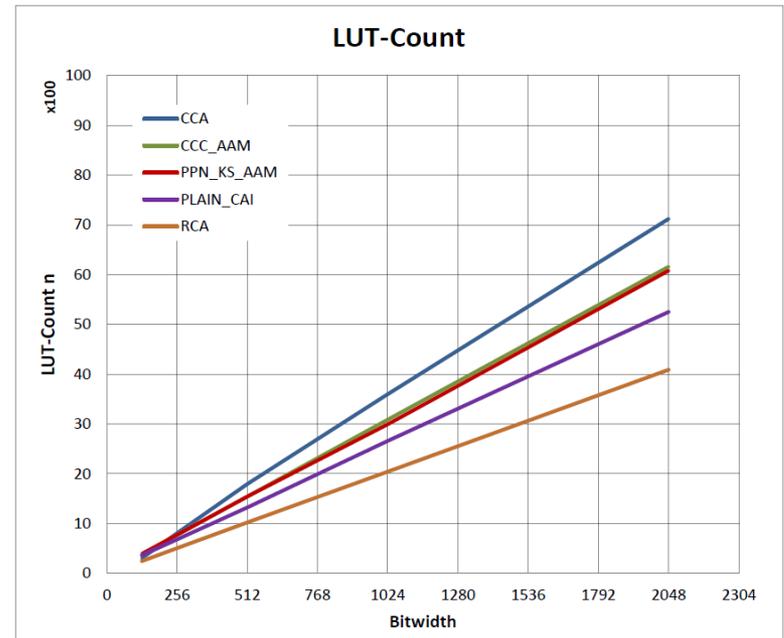


# Ergebnisse

## Virtex 6

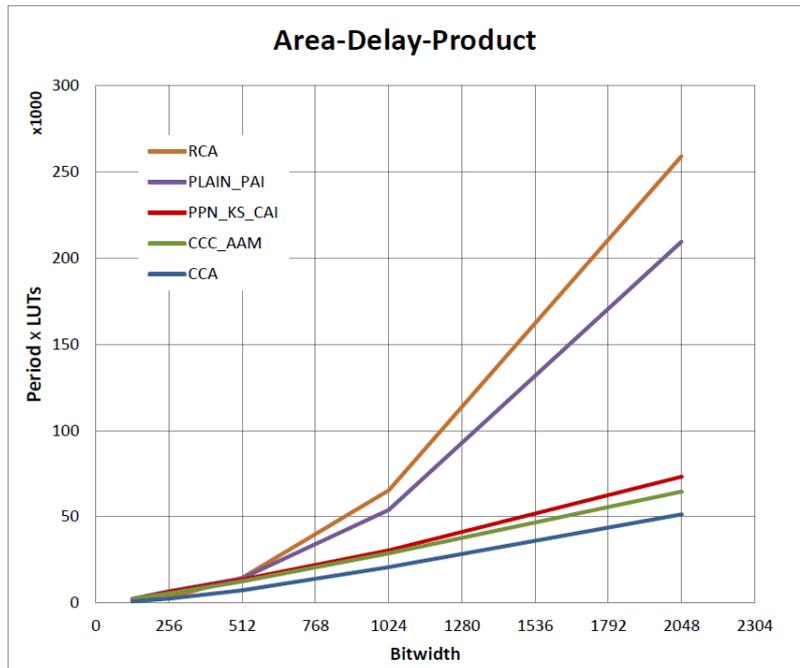


## Virtex 7

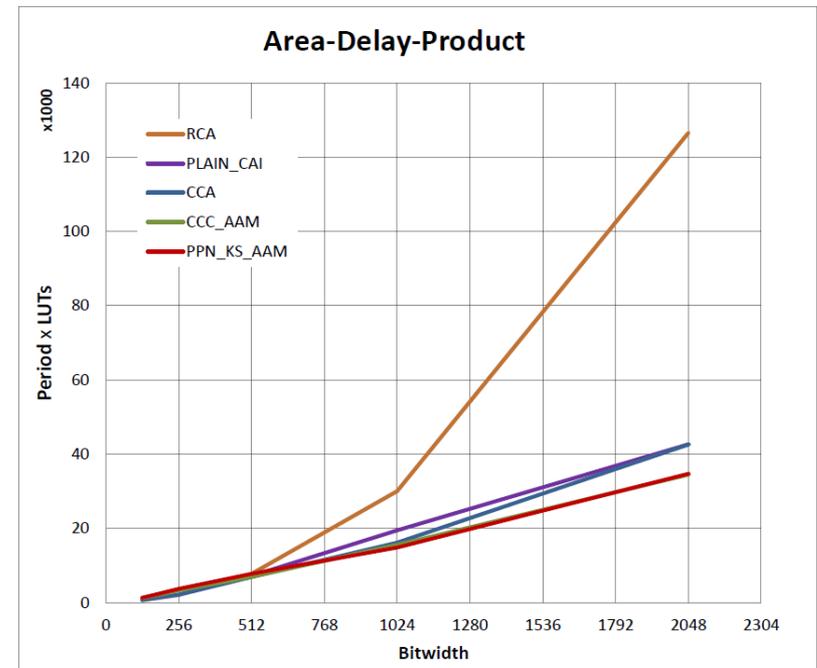


# Ergebnisse

## Virtex 6

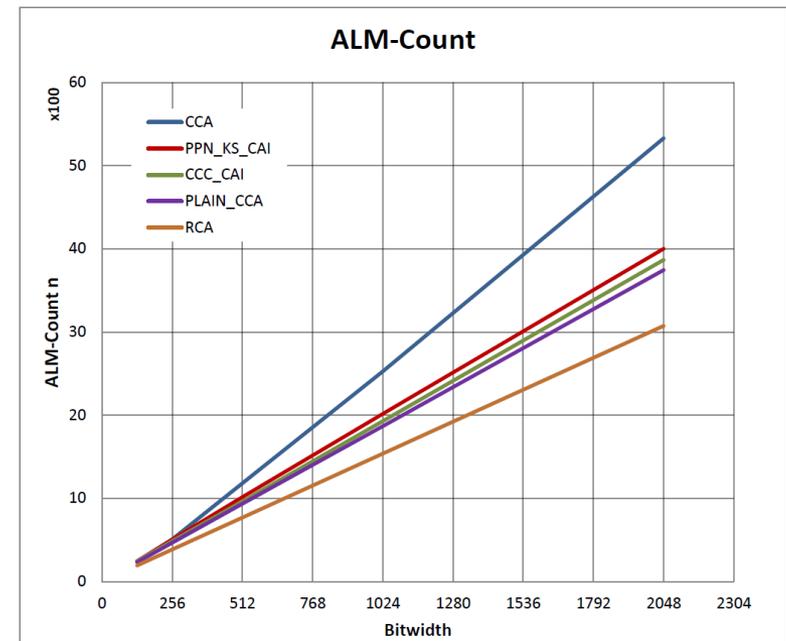
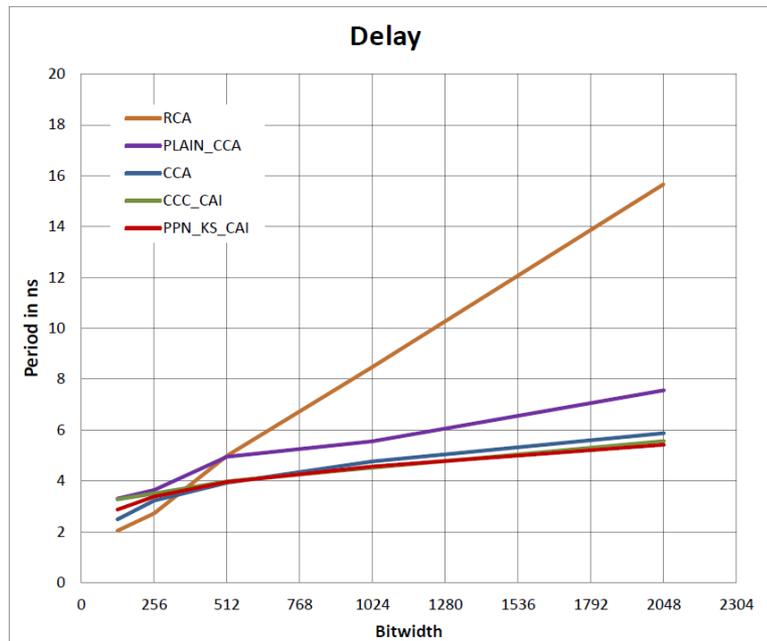


## Virtex 7



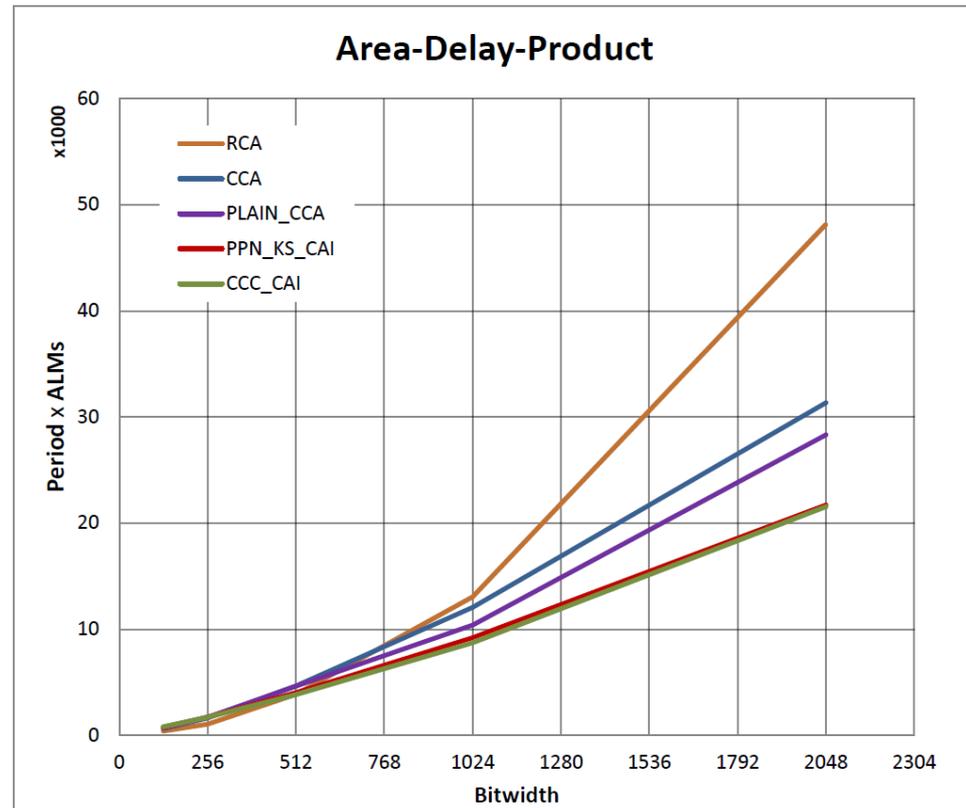
# Ergebnisse

- Stratix V:



# Ergebnisse

- Stratix V:



## Fazit

- keines der Designs hebt sich deutlich vom Rest ab
- beste Wahl abhängig von Device und Bitbreite
- unterschiedliche Einheiten bei HW-Verbrauch machen Vergleich dahingehend schwierig
- generell sehr gute Alternativen zum Standard-RCA vorhanden

## Quellenverzeichnis

- [1] Wikipedia Eintrag zum Kogge-Stone-Adder
- [2] H. Nguyen, B. Pasca, and T. Preusser, "FPGA-specific arithmetic optimizations of short-latency adders," in 2011 International Conference on Field Programmable Logic and Applications (FPL)
- [3] M. Rogawski, E. Homsirikamol, and K. Gaj, "A Novel Modular Adder for One Thousand Bits and More Using Fast Carry Chains of Modern FPGAs"
- [4] T. Preußner, M. Zabel, and R. Spallek, "Accelerating Computations on FPGA Carry Chains by Operand Compaction" in 2011 20th IEEE Symposium on Computer Arithmetic
- [5] Wikipedia Eintrag zu Ripple-Carry-Adder
- [6] T. Preußner, "Vorlesungsfolien Computerarithmetik"
- [7] Kris Gaj, "Conditional-Sum Adders and Parallel Prefix Network Adders. FPGA Optimized Adders" Vorlesung von 2014, George Mason University
- [8] IP Core Library - Published and maintained by the Chair for VLSI Design, Diagnostics and Architecture, Faculty of Computer Science, Technische Universität Dresden, Germany



**»Wissen schafft Brücken.«**