



Zwischenpräsentation zur Studienarbeit:

Bewertung innovativer Addiererstrukturen für
breite Operanden auf aktuellen FPGA-
Plattformen

Markus Krause

Dresden, 08.12.2015



Gliederung

1. Einführung
2. Addiererstrukturen
3. Aktueller Stand
4. Ausblick

Einführung

- **Problemstellung und Motivation:**
 - die binäre Wortaddition ist aufgrund des internen Übertrags häufig der kritische Pfad
 - Problem bei sehr breiten Operanden wie z.B. in der Kryptographie
 - daher Einsatz von Carry-Chains
 - Realisierung schneller, breiter Addierer

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

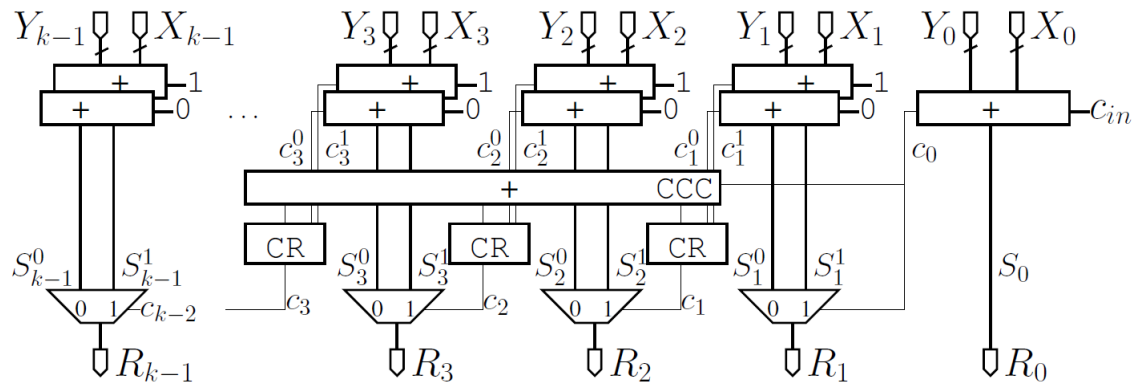
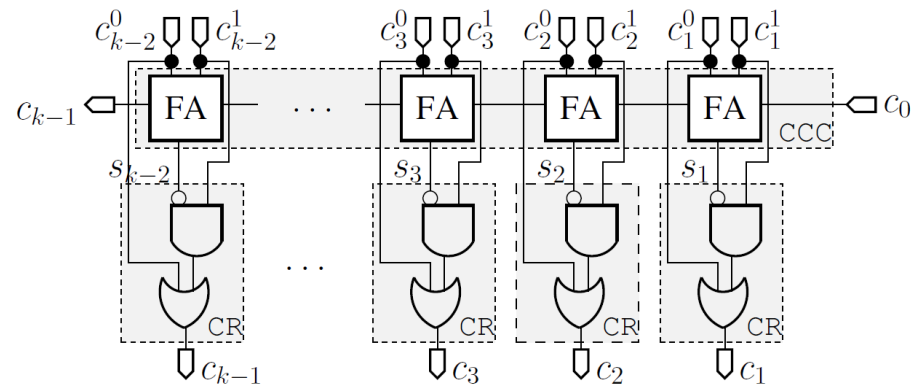
Addiererstrukturen

- Carry Select:
 - spekulatives Berechnen beider Blocksummen und Carries
 - mit Hilfe von Multiplexern zutreffendes Ergebnis auswählen
 - z.B. Add-Add-Multiplex (AAM)
 - arbeitet aber statt der MUX-Chain mit Carry-Computation-Circuit (CCC) und Carry-Recovery (CR)

Addiererstrukturen

- Carry Select: AAM

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



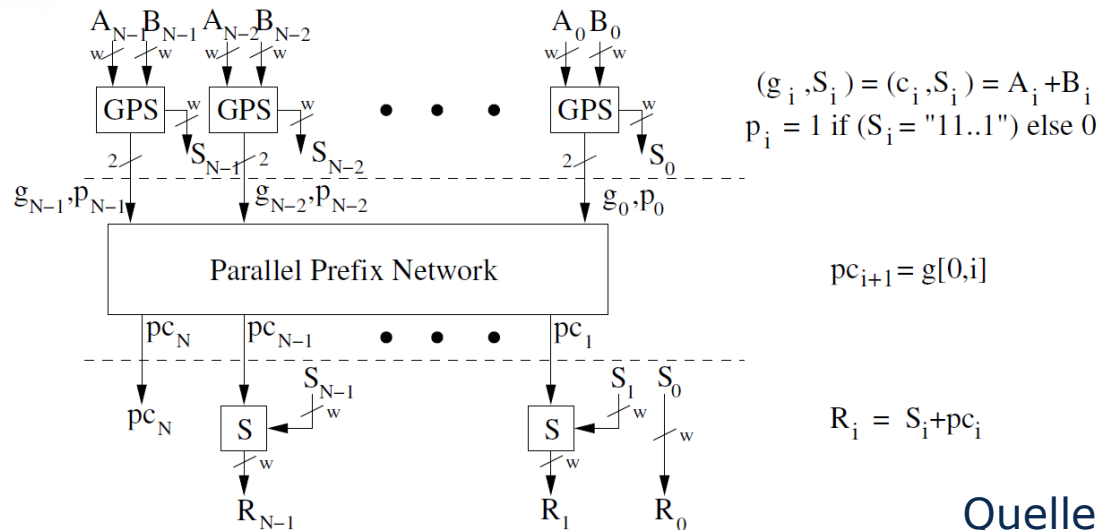
Quelle: [1]

Addiererstrukturen

- High Radix Parallel Prefix:
 - ähnlicher Aufbau wie Carry Select
 - theoretisch schneller als CslA
 - $O(\log(n)) < O(\sqrt{n})$
 - der Novel-HR-PPN-Adder als eine Möglichkeit auf FPGA

Addiererstrukturen

- High Radix Parallel Prefix: Novel-PPN
 - Verwendet Wörter statt Einzel-Bits
 - PPN Größe reduziert sich um $N = \frac{n}{w}$



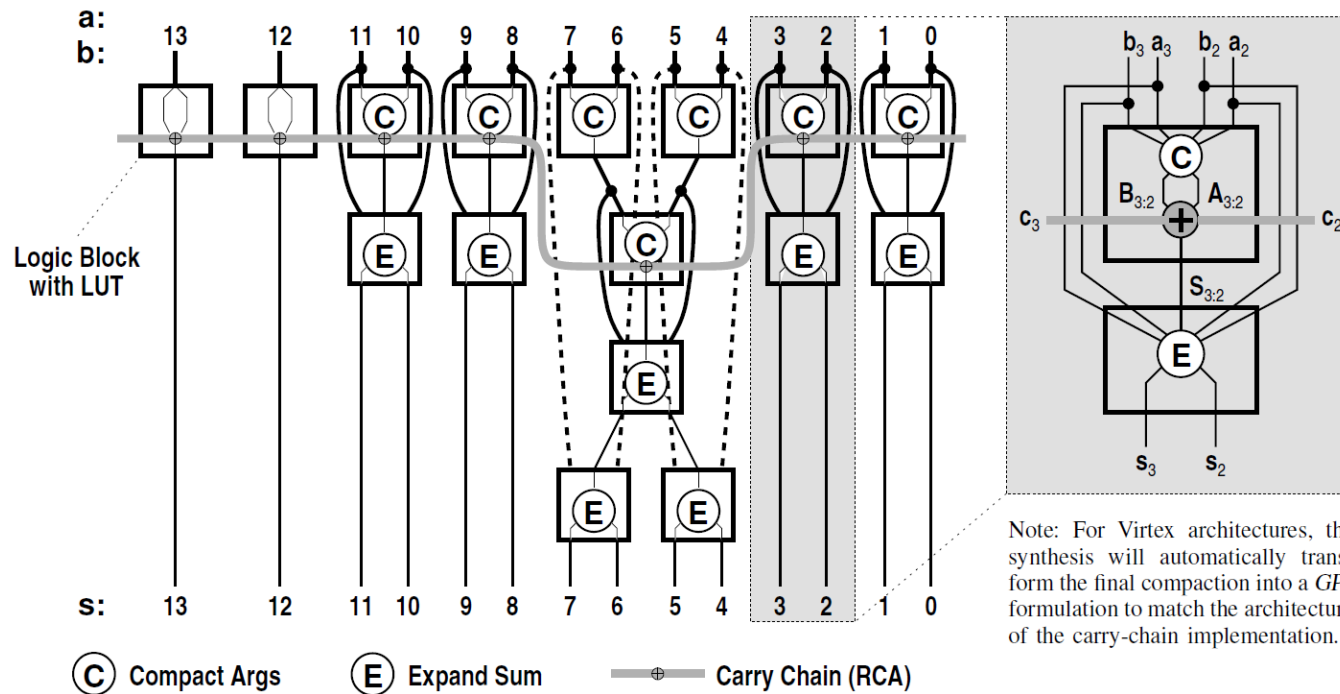
Quelle: [3]

Addiererstrukturen

- Carry Compact:
 - unterscheidet sich stärker von den anderen beiden Addern
 - ist bereits ab 50 Bit Wortbreite schneller als Standard RCA
 - nutzt Grouping

Addiererstrukturen

- Carry Compact:



Note: For Virtex architectures, the synthesis will automatically transform the final compaction into a *GP*-formulation to match the architecture of the carry-chain implementation.

Quelle: [2]

Addiererstrukturen

- Carry Compact: Sum-Expansion

- $S_{i+1:i} = P_{i+1:i} \oplus c_i$
- $c_i = S_{i+1:i} \oplus ((b_{i+1} \oplus a_{i+1})(b_i \oplus a_i))$
- $s_i = (b_i \oplus a_i) \oplus c_i$
- $s_{i+1} = (b_{i+1} \oplus a_{i+1}) \oplus (b_i a_i \oplus (b_i \oplus a_i)c_i)$

Aktueller Stand

- Literaturstudium:
 - soweit abgeschlossen
- Syntheseresultate:
 - für Xilinx begutachtet
- Skriptentwicklung:
 - für Xilinx fast abgeschlossen

Ausblick

- Syntheseresultate:
 - für Altera betrachten
- Skriptentwicklung:
 - für Altera entwickeln
- Quantitative Messung
- Dokumentation

Quellenverzeichnis

- [1] 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)
- [2] T. Preußer, M. Zabel, and R. Spallek, "Accelerating Computations on FPGA Carry Chains by Operand Compaction" in 2011 20th IEEE Symposium on Computer Arithmetic
- [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] A. Cilaro, "Variable-Latency Signed Addition on FPGAs"
- [5] Xilinx Command Line Tool Users Guide
- [6] T. Preußer, "Vorlesungsfolien Computerarithmetik"



»Wissen schafft Brücken.«