TECHNISCHE UNIVERSITÄT DRESDEN

Implementierung eines Carry-Compact-Adders auf einem FPGA

von Matthias Brinker

Dresden, 22.06.2015

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Gliederung

01. Theorie
02. Implementierung auf einem FPGA
03. Leistung
04. Fazit

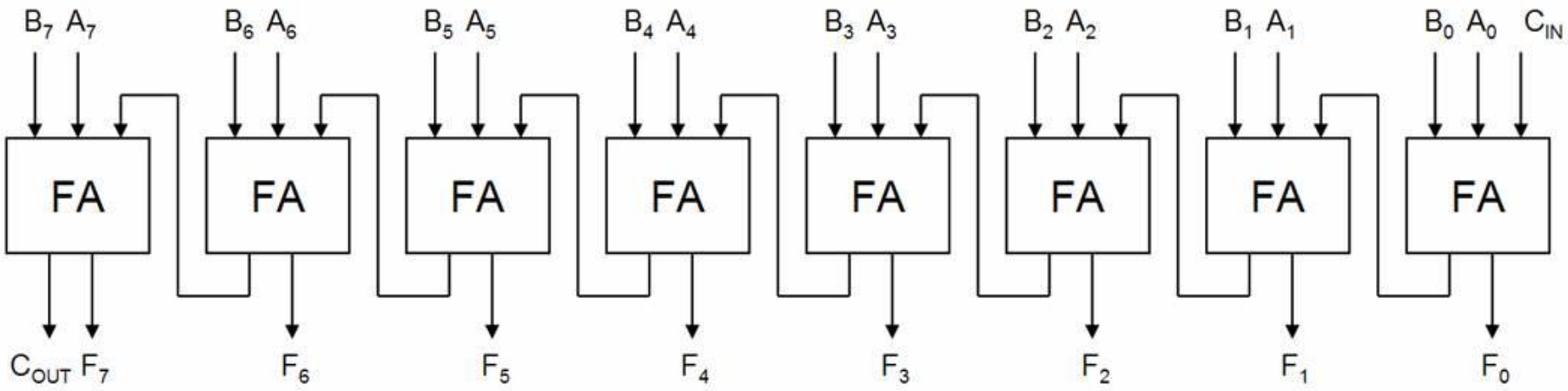
01 Theorie

Ziele eines Addierers

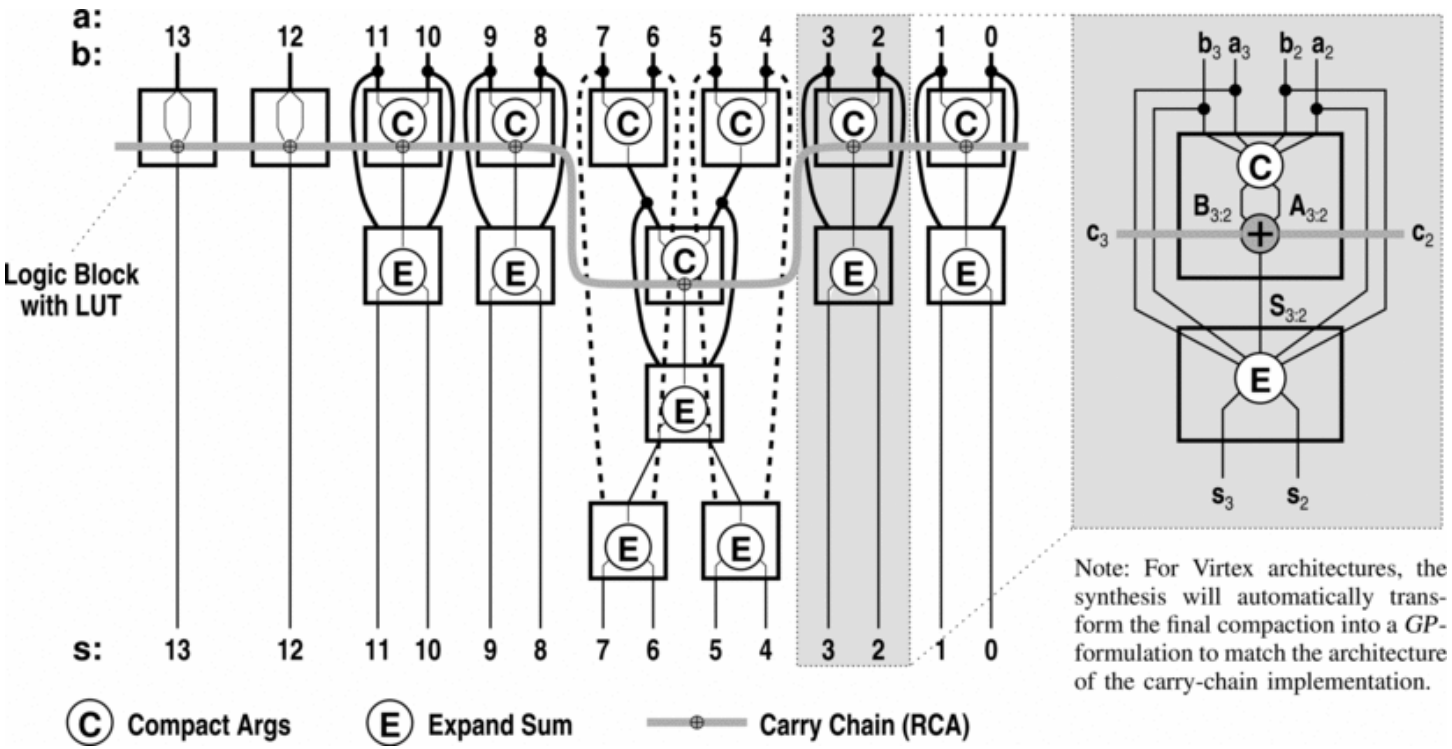
- Möglichst kurzen kritischen Pfad
  -> Hohe Frequenz

- Möglichst wenig Hardwareaufwand
  -> Wenig Fläche auf dem Chip

Ripple Carry Adder



Carry Compact Adder



Begriffserklärung

g_i : Generate

- Die Addition liefert einen Übertrag

-> a_i und b_i sind 1

p_i : Propagate

- Die Addition kann einen Übertrag liefern

-> entweder a_i oder b_i ist 1

-> Eingehender Übertrag = Ausgehender Übertrag

Begriffserklärung

$G_{i:j}$: Blockgenerate

- Der gesamte Block generiert einen Übertrag

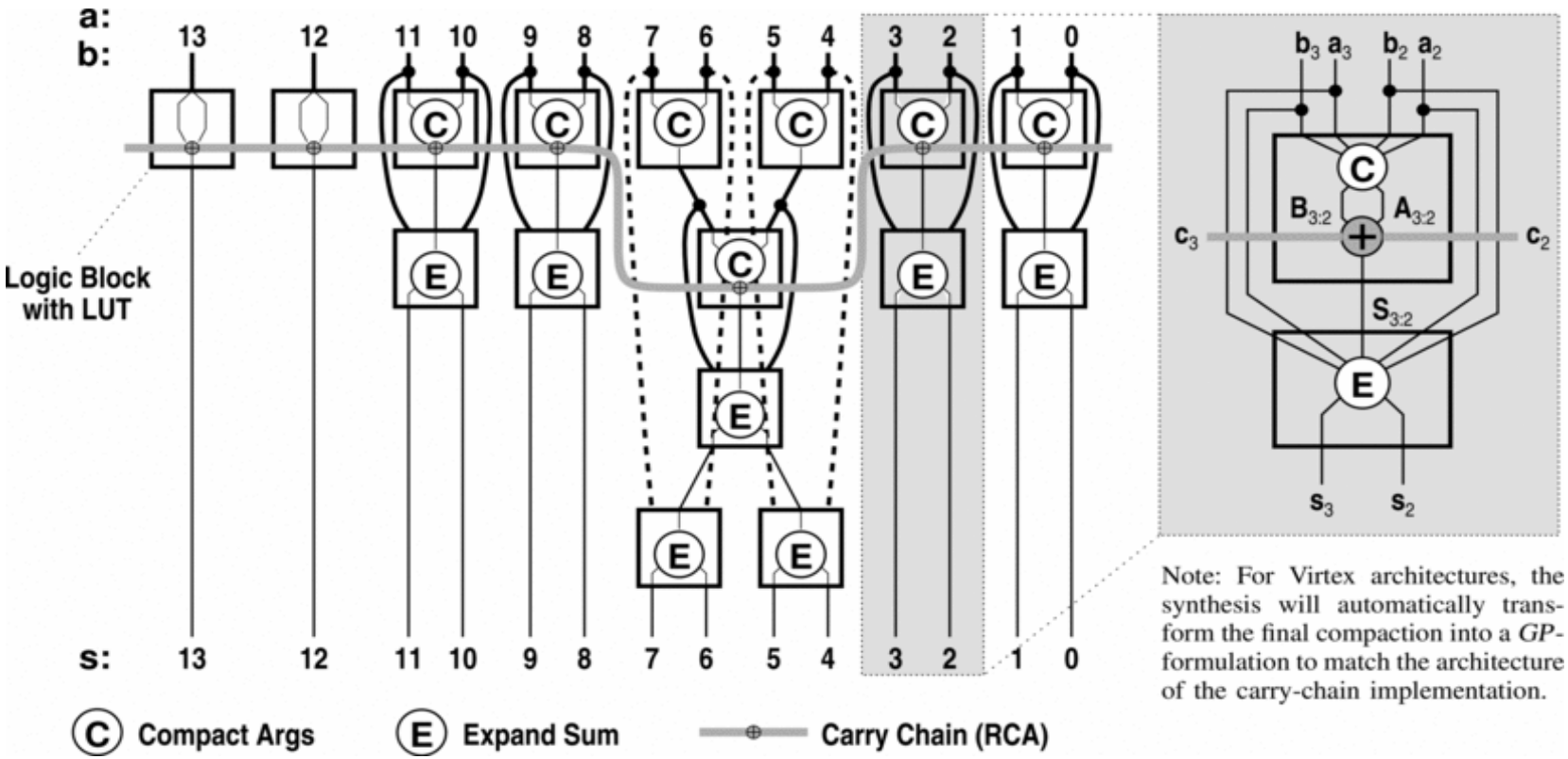
$P_{i:j}$: Blockpropagate

- Alle Propagate-Signale (in dem Intervall) sind 1

Defnition von $A_{i+1:i}$ und $B_{i+1:i}$

g_{i+1}	p_{i+1}	b_i	a_i	$G_{i+1:i}$	$P_{i+1:i}$	Case	$B_{i+1:i}$	$A_{i+1:i}$
0	0	*	*	0	0	K	0	0
0	1	0	0	0	0	K	0	0
0	1	0	1	0	1	P	0	1
0	1	1	0	0	1	P	1	0
0	1	1	1	1	0	G	1	1
1	*	*	*	1	0	G	1	1

02 Carry Compact Adder



Berechnung des Carrys

$$S_{i+1:i} = P_{i+1:i} \oplus c_i$$

$$c_i = S_{i+1:i} \oplus P_{i+1:i}$$

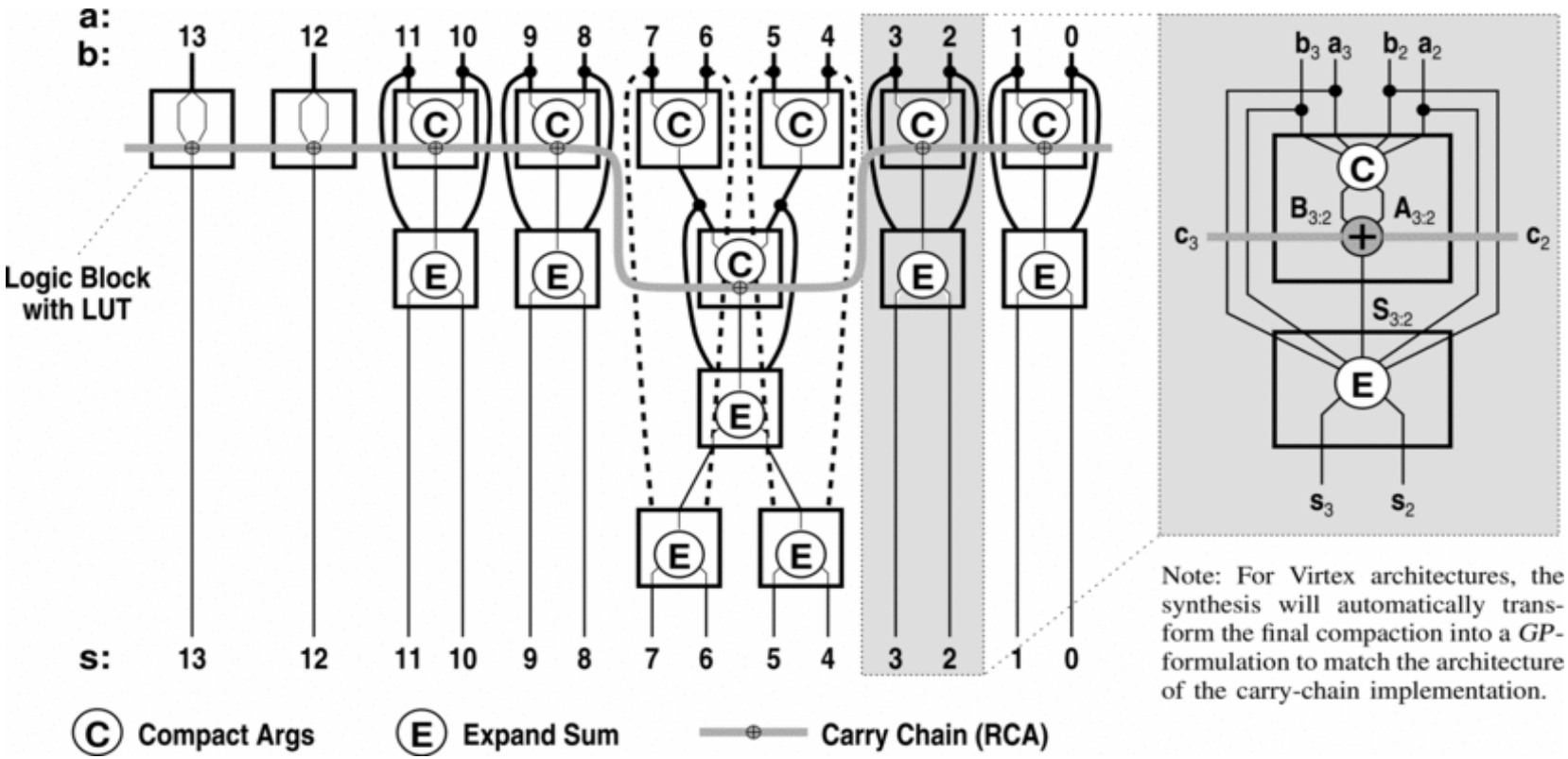
$$c_i = S_{i+1:i} \oplus ((b_{i+1} \oplus a_{i+1})(a_i \oplus b_i))$$

Berechnung der expandierten Summen

$$s_i = p_i \oplus c_i$$
$$= (a_i \oplus b_i) \oplus c_i$$

$$s_{i+1} = p_{i+1} \oplus c_{i+1}$$
$$= p_{i+1} \oplus (g_i + p_i c_i)$$
$$= (a_{i+1} \oplus b_{i+1}) \oplus (a_i b_i + (a_i \oplus b_i)\, c_i)$$

02 Carry Compact Adder

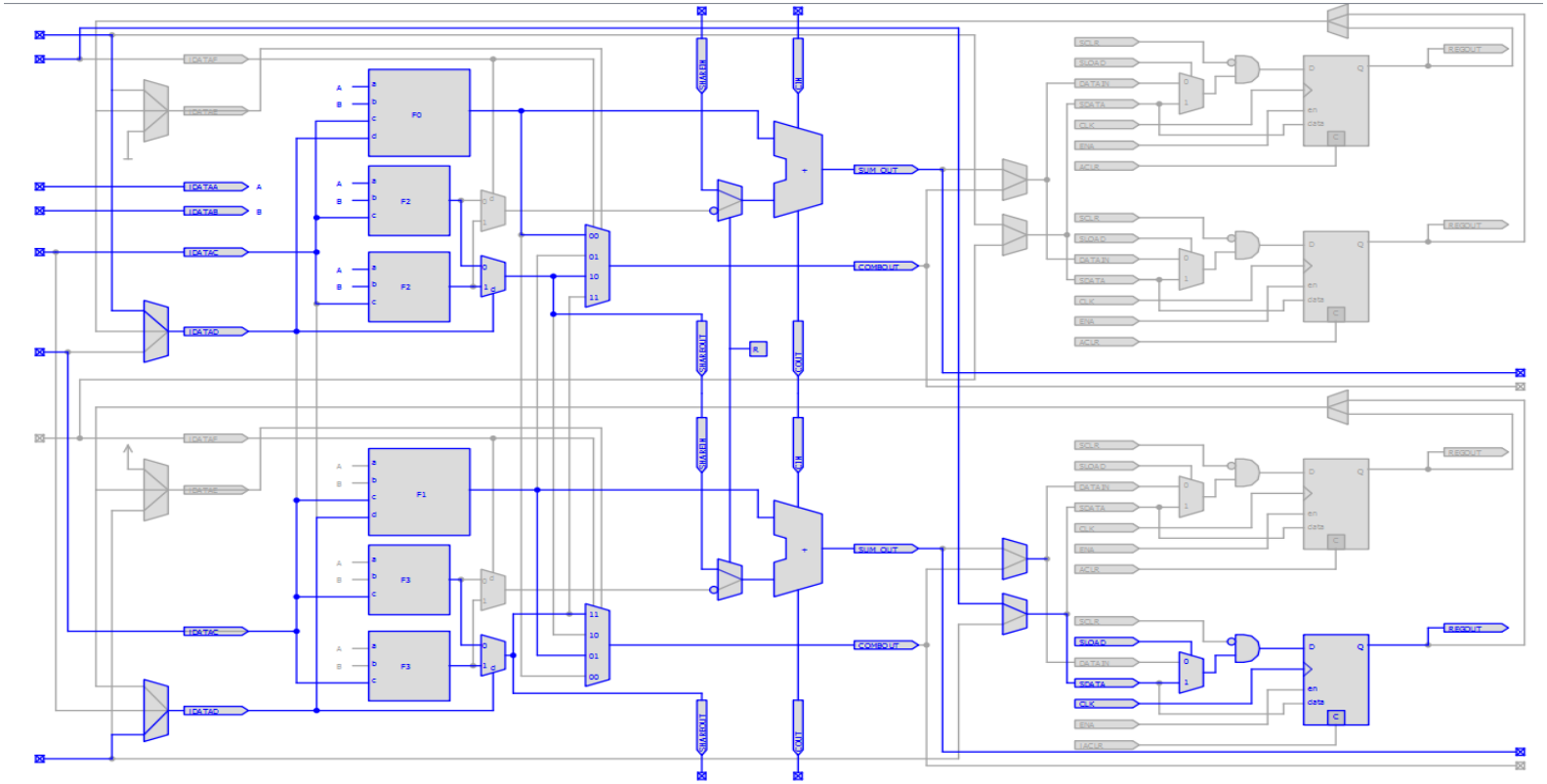


02 Implementierung auf einem FPGA

Implementierung auf einem FPGA

Warum?

ALM-Struktur eines Cyclone V Architektur



Implementierung auf einem FPGA

Warum?

- Schnelle Carry-Line
- LUTs eignen sich sehr gut für das Kompaktieren und Expandieren

LUT-Primitive

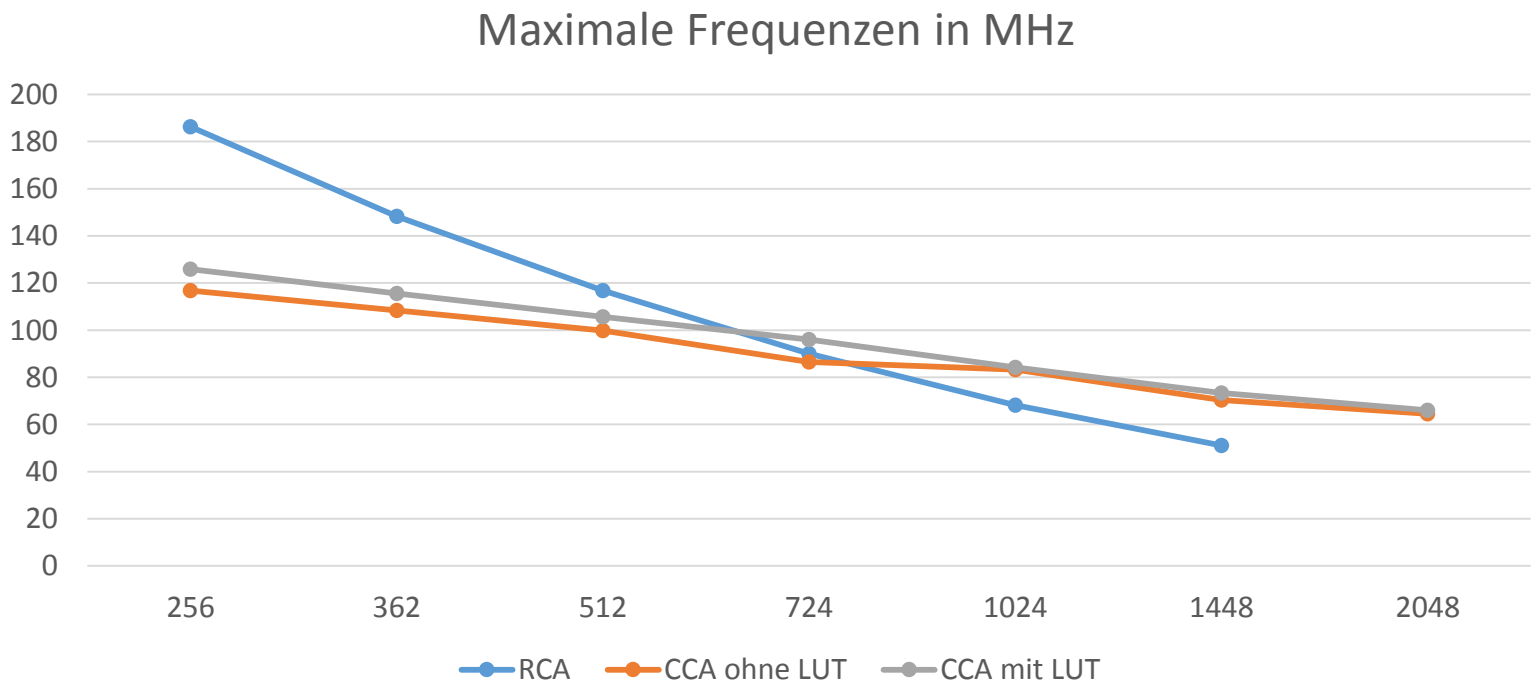
- LUT_INPUT und LUT_OUTPUT definieren die Grenzen einer LUT

03 Leistung

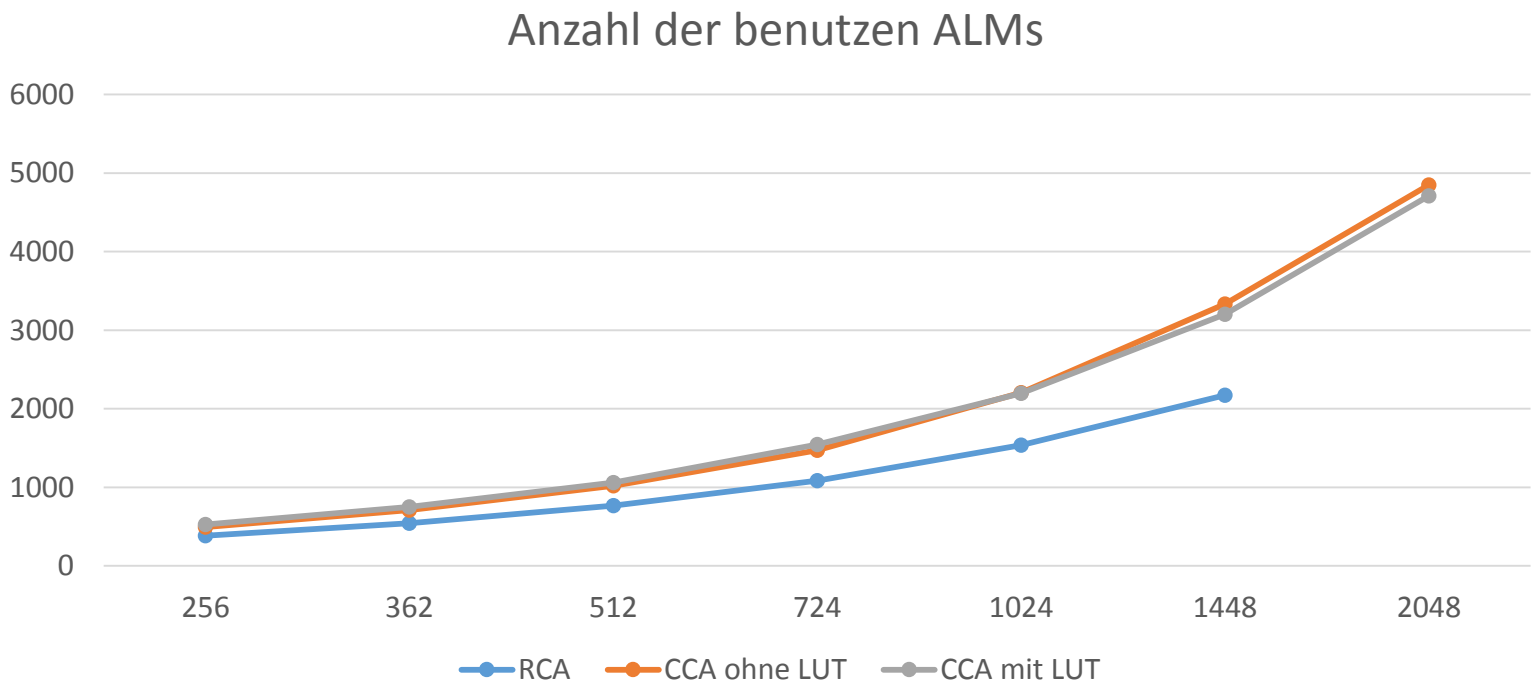
Messreihen

- RCA
- CCA ohne LUT-Primitive
- CCA mit LUT-Primitive

Verlauf der Frequenzen



Verlauf der ALMs



04 Fazit

Fazit

- Ist bei großen Bitbreiten schneller als die Implementierung eines RCAs in Quartus
- Braucht mehr Ressourcen
- Ab einer Bit-Breite von ~724 lohnt sich der Carry-Compact-Adder

»Wissen schafft Brücken.«

RCA

Bitbreite	MHz	ALMs
256	186,33	384
362	148,27	543
512	116,73	768
724	90,04	1086
1024	68,14	1536
1448	51,12	2172
2048		

CCA ohne LUT-Primitive

Bitbreite	MHz	ALMs
256	116,78	492
362	108,3	711
512	99,85	1019
724	86,55	1473
1024	83,31	2205
1448	70,39	3333
2048	64,56	4846

CCA mit LUT-Primitive

Bitbreite	MHz	ALMs
256	125,94	526
362	115,59	749
512	105,63	1059
724	96,03	1543
1024	84,85	2218
1448	73,37	3204
2048	66	4708

Quellen

http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=5992114

http://www.ece.unm.edu/faculty/pollard/classes/338/lademo/LookAheadDemo.htm