

# Konstantenmultiplikation

## Hauptseminar technische Informatik

Martin Thümmel

Institut für technische Informatik TU Dresden

21.06.2018

$$\begin{array}{r} \underline{1001011 \times 1100010} \\ 1100010 \\ + \quad 1100010 \\ + \quad 0000000 \\ + \quad 1100010 \\ + \quad 0000000 \\ + \quad 0000000 \\ + \quad 1100010 \\ \hline 1110010110110 \end{array}$$

(a) Multiplikationstabelle für zwei beliebige Zahlen

$$\begin{array}{r} \underline{1001011 \times 1100010} \\ 1100010 \\ + \quad 1100010 \\ + \quad 1100010 \\ \hline 1110010110110 \end{array}$$

(b) Multiplikationstabelle mit der Konstanten 142

Abbildung:  $113 \cdot 142$  in Binärdarstellung.

- 1 CSD
- 2 Addierergraphen
- 3 Weitere Verfahren
- 4 Erweiterte Problemstellungen
- 5 Zusammenfassung und Ausblick

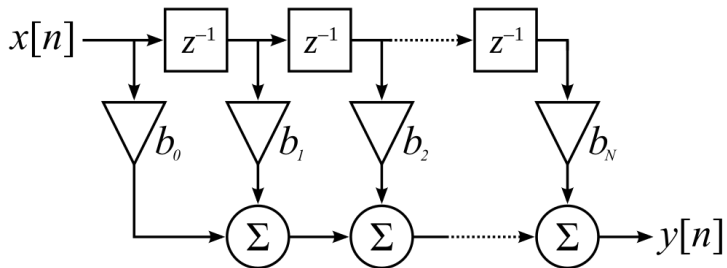


Abbildung: FIR (finite impulse response) Filter

# Mögliche Ziele der Konstantenmultiplikation

- Reduktion der benötigten Transistorenanzahl
- Reduktion der Berechnungszeit
- Steigerung des Durchsatzes
- Reduktion des Energieverbrauchs

# CSD

- Verwendung von Subtraktionen
- Ersetzungscode
- SD-Darstellung einer Zahl  $s$  durch

$$s = \sum_{k=0}^{N-1} c_k 2^k, \quad c_i \in \{-1, 0, 1\}$$

- CSD: SD-Darstellung ohne Teilfolgen  $11, \bar{1}1, 1\bar{1}, \bar{1}\bar{1}$ , wobei  $\bar{1} \equiv -1$

- Umwandlung von 477 in Binärdarstellung nach CSD

$$\begin{aligned}
 477 &= [111 \underbrace{0111}_{[100\bar{1}]_{SD}} 01]_{SD} \\
 &= [ \underbrace{1111}_{[1000\bar{1}]_{SD}} 00\bar{1}01 ]_{SD} \\
 &= [1000\bar{1}00\bar{1}01]_{SD}.
 \end{aligned}$$

- Statt 6 nun 3 Additionen



- Linearer Aufwand zur Generierung der CSD-Darstellung
- Im Durchschnitt 1/3 weniger Additionen als Binärdarstellung
- Für  $N$ -Bitwort maximal  $\frac{N+2}{2}$  Additionen
- Grund: Verwendung von Subtraktionen

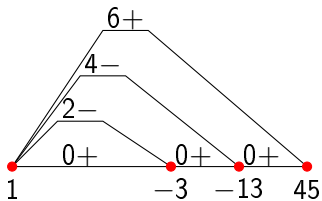
# Addierergraphen

- CSD nicht immer optimal
- $45 = 64 - 16 - 4 + 1$ , also 3 Additionen in CSD
- Mit  $45 = 5 \cdot 9$ :

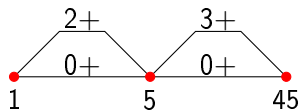
$$5x = 2^2x + 2^0x, \quad 45x = 2^3(5x) + 2^0(5x)$$

- Berechnung mit 2 Additionen möglich
- Grund: Verwendung von Zwischenergebnissen

# Einige Addierergraphen



(a) CSD-Verfahren



(b) Optimaler Addierergraph

Abbildung: Addierergraphen für Multiplikation mit 45

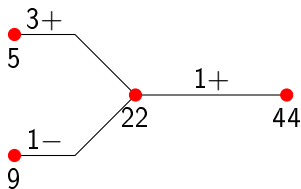
Vollständige Suche über alle Addierergraphen

- Ermittlung aller Addierergraphen bis zu  $k$  Additionen
- Suche über alle möglichen Belegungen von Kanten
- Dynamische Programmierung mittels Lookup-Tabelle

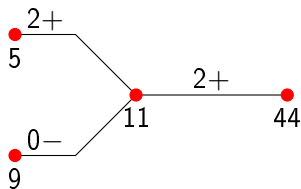
Weitere Optimierungen möglich, da

- Äquivalenzklassen von Addierergraphen
- Schieben und Addieren distributiv
- Erkennung mancher suboptimaler Ergebnisse durch DP

# Einschränkung auf ungerade Knotenwerte



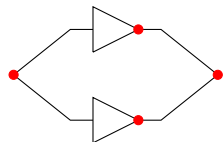
(a) Knotenwert 22



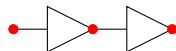
(b) Knotenwert  $22/2 = 11$

Abbildung: Transformation der Kantenbelegung und Erzeugung eines ungeraden Knotenwertes

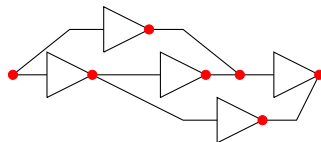
# Klassifikation von Addierergraphen



(a) Additiv



(b) Multiplikativ



(c) Leapfrog

Abbildung: Beispiele von verschiedenen Addierergraphklassen

Addierer	Addierergraphen	Addierergraphklassen
0	1	1
1	1	1
2	2	2
3	7	5
4	32	15
5	193	54
6	1444	227
7	13228	1162

**Tabelle:** Anzahlen an Addierergraphen und - klassen für gegebene Addiereranzahl



# Ergebnisse vollständige Suche

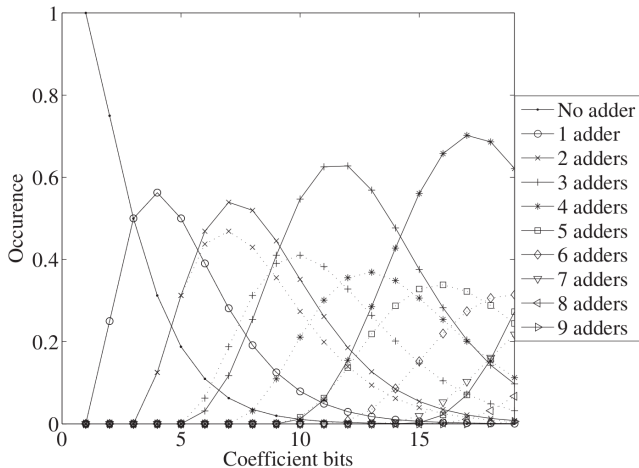


Abbildung: Relative Häufigkeit der minimal benötigten Anzahl an Addierern für verschiedene Wortlängen

## Weitere Verfahren

# CSE (Common Subpattern Elimination)

- Startpunkt: CSD-Darstellung
- Ziel: Wiederverwendung von Teilausdrücken
- Suche über beschränkte Anzahl an SD-Darstellungen
- Beispiel: 2 statt 3 Additionen

$$105 = [1010\bar{1}001]_{SD} = [10011001]_{SD}.$$

- Kubische Laufzeit in Wortlänge

- Aufteilen der Binärzahl in Blöcke der Länge  $r$
- Berechnung und Addition der benötigten Blockergebnisse
- Beispiel: 5 statt 7 Additionen

$$\begin{aligned}10599 &= 2 \cdot 2^{12} + 4 \cdot 2^9 + 5 \cdot 2^6 + 4 \cdot 2^3 + 7 \\ &= 2 \cdot 2^{12} + 4 \cdot 2^9 + 5 \cdot 2^6 + 5 \cdot 2^3 - 1 \\ &= 2 \cdot 2^{12} + 4 \cdot 2^9 + 6 \cdot 2^6 - 3 \cdot 2^3 - 1 \\ &= 2 \cdot 2^{12} + 5 \cdot 2^9 - 2 \cdot 2^6 - 3 \cdot 2^3 - 1 \\ &= 3 \cdot 2^{12} - 3 \cdot 2^9 - 2 \cdot 2^6 - 3 \cdot 2^3 - 1.\end{aligned}$$

$$10599x = (3x)2^{12} - (3x)2^9 + x2^7 - (3x)2^3 - x$$

- Lineare Laufzeit in Wortlänge

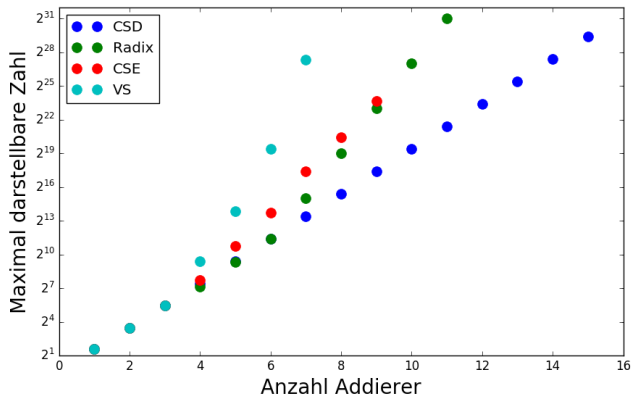


Abbildung: Erste nicht darstellbare Zahlen für verschiedene Addiereranzahlen

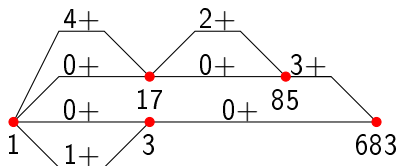
WL	CSD	Radix- $2^r$	CSE	Volls. Suche
8	2,788 (9,4%)	2,864 (12,4%)	2,547 (0,0%)	2,547 (0,0%)
16	4,444 (12,1%)	4,512 (13,8%)	4,071 (2,7%)	3,964 (0,0%)
24	7,111 (39,2%)	6,799 (33,1%)	5,433 (6,3%)	5,110 (0,0%)

**Tabelle:** Durchschnittliche Anzahl an benötigten Addierern

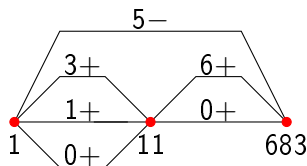
# Erweiterte Problemstellungen

# Ternäre Addierer

- Duale und ternäre Addierer in FPGAs ungefähr gleichaufwändig
- Addierergraphen mit ternären Additionen
- Beispiel: 2 statt 4 Addierer



(a) Verwendung dualer Addierer



(b) Verwendung ternärer Addierer

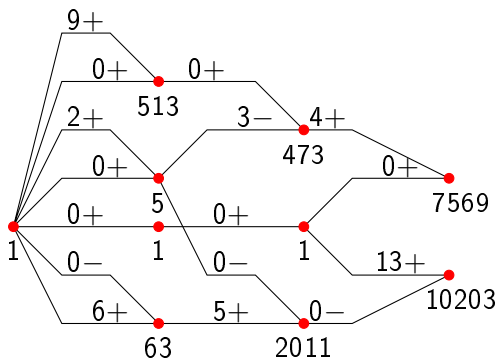
Abbildung: Addierergraphen zur Multiplikation mit 683



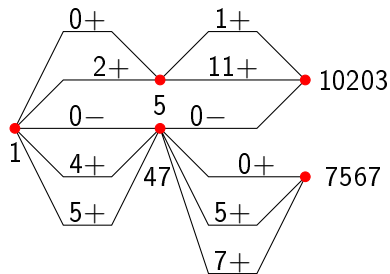
# MCM (Multiple Constant Multiplication)

- MCM zum Beispiel bei Fouriertransformation
- Wiederverwendung von Zwischenergebnisse für mehrere Endergebnisse
- Pipelining
- Optimierung z.B. via vollständige Suche oder ähnlich CSE

# MCM Beispiel



(a) Dualer Addierergraph



(b) Ternärer Addierergraph






Abbildung: Multiplikation mit Konstanten 7569 und 10203




Meist mehrere Realisierungen mit minimaler Anzahl an Additionen  
Metriken:

- Abarbeitungszeit
- Pipelining
- Energieverbrauch

## Zusammenfassung und Ausblick

- Betrachtung der Konstantenmultiplikation notwendig – hilfreiche Verfahren existieren
- Optimale Suchverfahren bis 32 Bit Wortlänge
- Heuristische Verfahren praktisch beliebige Wortlängen
- Untersuchung von erweiterten Problemstellungen und Metriken

-  A.G. Dempster, M.D Macleod, *Constant multiplication using minimum adders*, IEE Proc.-Circuits Devices Syst., Vol. 141, No. 5, October 1994
-  O. Gustafsson, A.G. Dempster, K. Johansson, M.D. Macleod, L. Wanhammar, *Simplified Design of Constant Coefficient Multipliers*, Circuits Systems Signal Processing, Vol . 25, No . 2, 2006, pp . 225–251
-  M. Kumm, M. Hardieck, J. Willkomm, P. Zipf, U. Meyer-Baese, *Multiple Constant Multiplication with Ternary Adders* 23rd International Conference on Field Programmable Logic and Applications, FPL 2013 - Proceedings
-  M. Kumm, O. Gustafsson, M. Garrido, P. Zipf, *Optimal Single Constant Multiplication using Ternary Adders* IEEE Transactions on Circuits and Systems II: Express Briefs, 2016
-  A. K. Oudjia, N. Chaillet, *Radix-2<sup>r</sup> Arithmetic for Multiplication by a Constant* IEEE Transactions on Curcuits and Systems II: Express Briefs, 2014

-  V. Lefevre, *Multiplication by an Integer Constant* Institut National de Recherche en Informatique et en Automatique, N 4192, 2001
-  J. Thong, N Nicolici, *An Optimal and Practical Approach to Single Constant Multiplication* IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 30, No 9, 2011
-  Y. Voronenko, M. Püschel, *Multiplierless Multiple Constant Multiplication* ACM Transactions on Algorithms, Vol. 3, 2007