TECHNISCHE UNIVERSITÄT DRESDEN

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

Realisierung einer MC-basierten Optionspreisberechnung mit FloPoCo

Christian Skubich

Dresden, 11.02.2015

Gliederung

1. FloPoCO

- **Flo**ting **Po**int **Co**res, but not only
- Generator für angepasste Arithmetikkerne
- eigenes Floating Point Format:

Exception	Sign	Exponent	Fractional
2 Bit	1 Bit	konfigurierbar	konfigurierbar

Generierbare Komponenten

Standardkomponenten
z.B. +, x, /, sqrt, exp, …

Festkomma Funktionen
z.B. $\sin(x \cdot \frac{\pi}{2})$

Floating Point Pipeline
z.B. $e^{a \cdot 0.5 + b^2}$

Testbenches

Konvertierung Festkomma ⇔Fließkomma ⇔IEEE

Komplexe Zahlen

…

Konfigurationsmöglichkeiten

- Bitbreite
- Zielfrequenz
- Wahl zwischen Implementierungen
- Signed/Unsigned
- ...

2.1 Black-Scholes

- Modell zur Bewertung von Finanzoptionen (1973 von Black und Schole)
- Monte-Carlo Simulation

for i= 1 to M

set $S_i = S_0 e^{(r-0.5\sigma^2)T+\sigma\sqrt{T}\varepsilon_i}$

set $P_i = e^{-rT}\max(S_i - E, 0)$

end

set $P_{mean} = \frac{1}{M}\sum_{i=1}^{M} P_i$

set $P_{var} = \frac{1}{M-1}\sum_{i=1}^{M}(P_i - P_{mean})^2$

for i= 1 to M

set $P_i = \max\left(S_* e^{b\varepsilon_i} + c, 0\right)$

set P_{mean} += P_i

end

set P_{mean} /= M

for i= 1 to M

set P_{var} += $(P_i - P_{mean})^2$

end

set P_{var} /= $M - 1$

Abgeleitete Parameter:

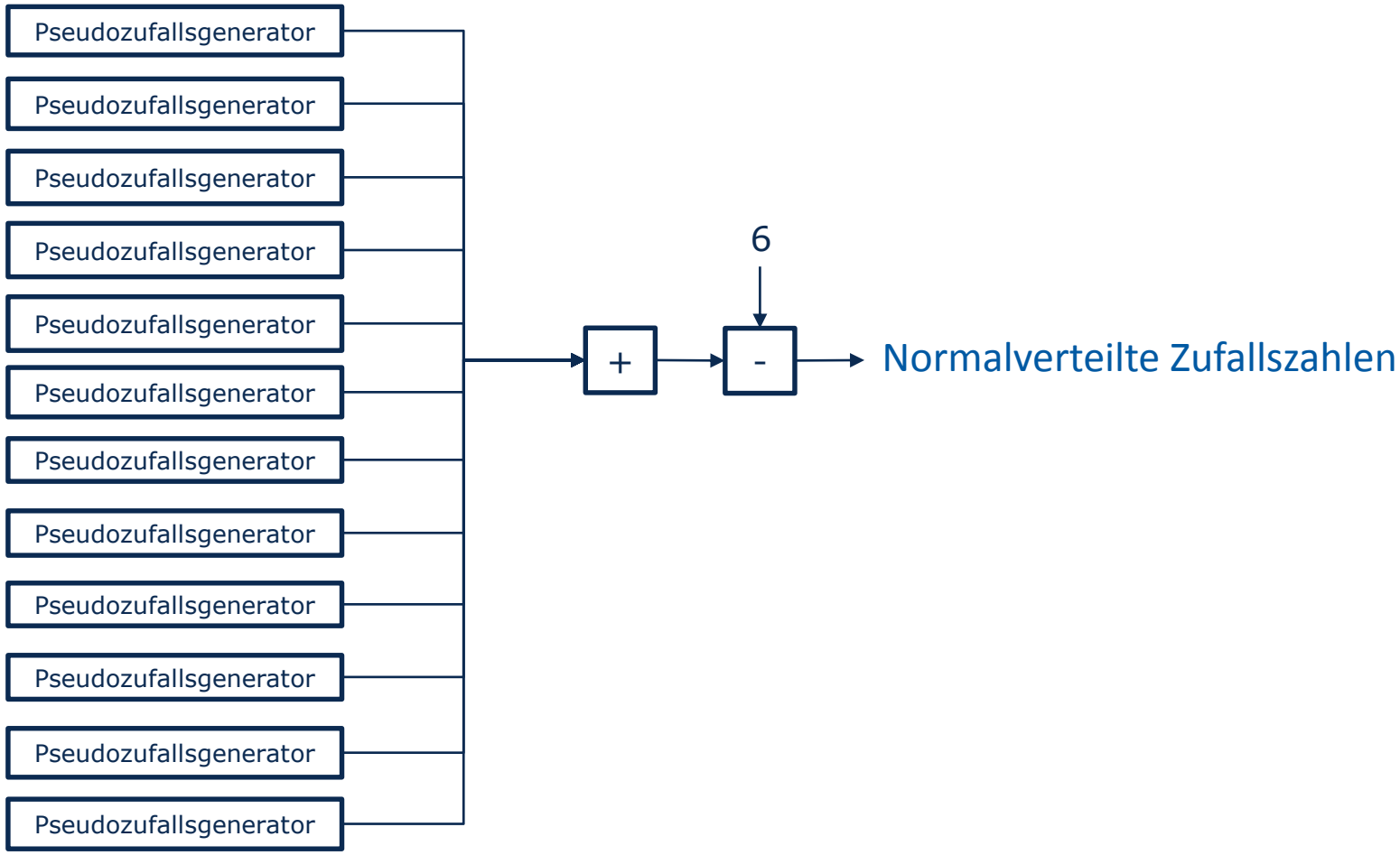
$S_* = S_0 e^{-0.5b^2}$

$b = \sigma\sqrt{T}$

$c = -e^{-rT}E$

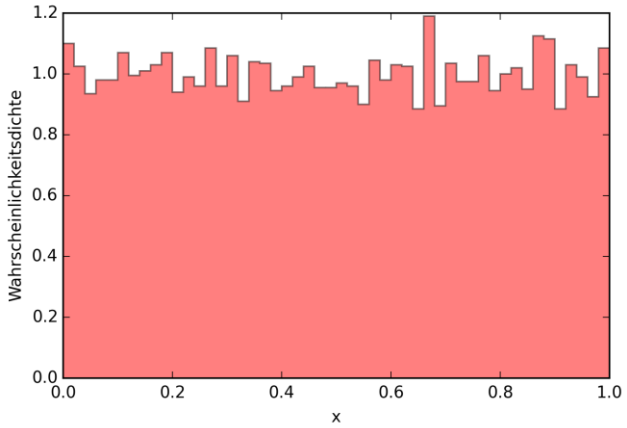
2.2 Zufallserzeugung

- Ziel: Generierung der normalverteilten ε_i für MC-Simulation
- Gegeben: Gleichverteilter Pseudo-Zufall

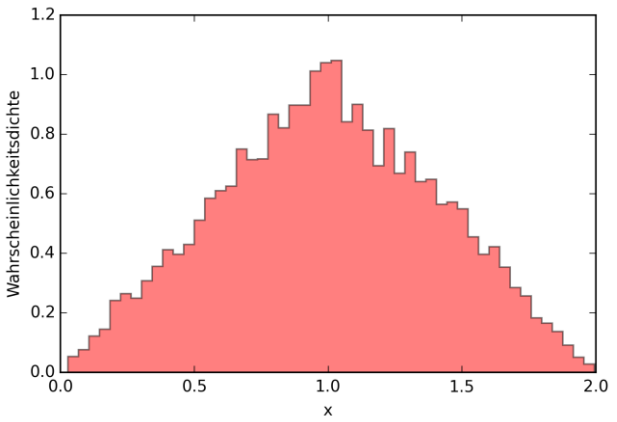

Pseudozufallsgenerator
Pseudozufallsgenerator
Pseudozufallsgenerator
Pseudozufallsgenerator
Pseudozufallsgenerator
Pseudozufallsgenerator
Pseudozufallsgenerator
Pseudozufallsgenerator
Pseudozufallsgenerator
Pseudozufallsgenerator
Pseudozufallsgenerator
Pseudozufallsgenerator
+
6
-
Normalverteilte Zufallszahlen

Was passiert bei diesem Algorithmus?

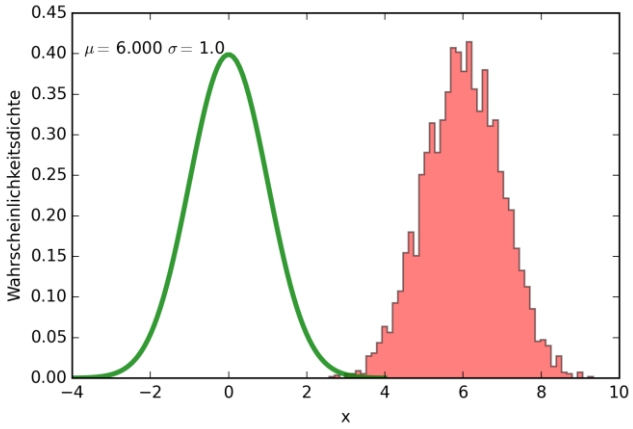
Gleichverteilung



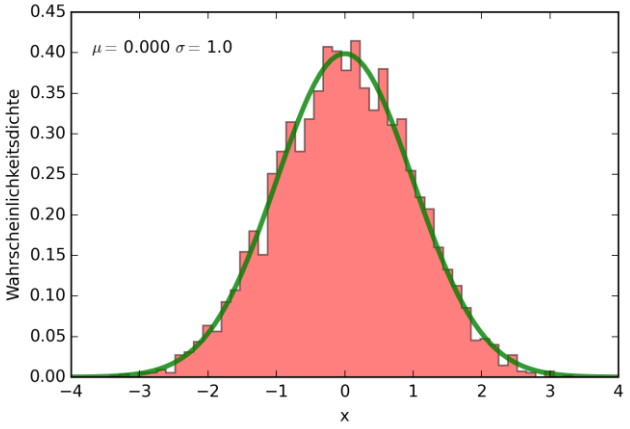
Summe von 2 Gleichverteilungen



Summe von 12 Gleichverteilungen



Korrektur des Erwartungswertes



3.1 Datenpfad

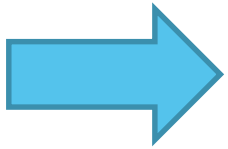
- Single Precision (8 Bit Exponent, 23 Bit Fractional Part)
- Generierung für Double Precision nicht möglich

for i= 1 to M

set $P_i = \max\left(S_* e^{b\varepsilon_i} + c, 0\right)$

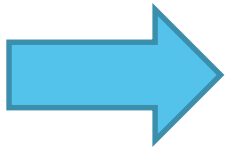
set P_{mean} += P_i

end



Automatische Pipeline, MUX & Akkumulator

set P_{mean} /= M

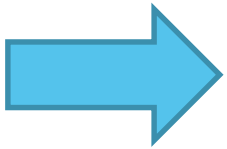


Dividierer

for i= 1 to M

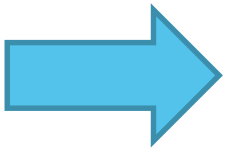
set P_{var} += $(P_i - P_{mean})^2$

end



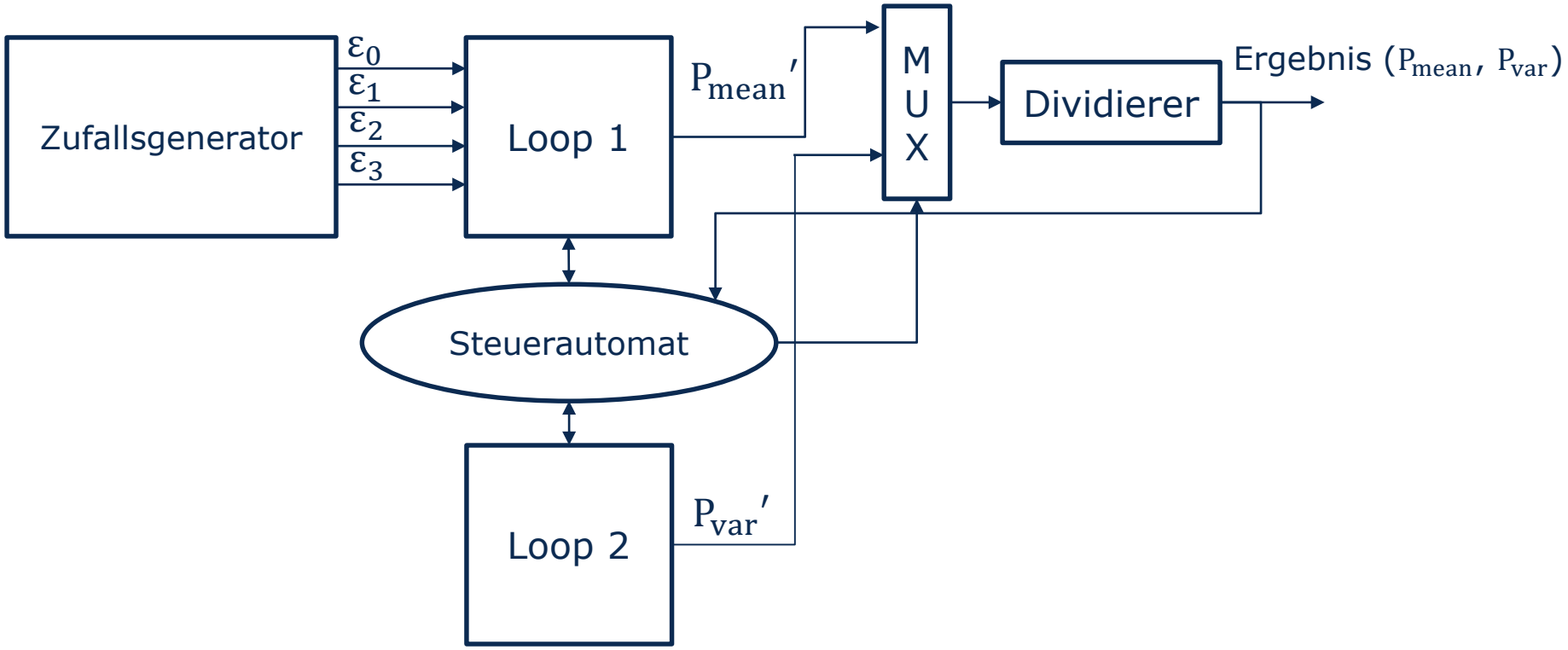
Addierer, Quadrierer & Akkumulator

set P_{var} /= $M - 1$



Dividierer

Top


Zufallsgenerator
ε_0
ε_1
ε_2
ε_3
Loop 1
P_{mean}'
M
U
X
Dividierer
Ergebnis (P_{mean}, P_{var})
Steuerautomat
Loop 2
P_{var}'

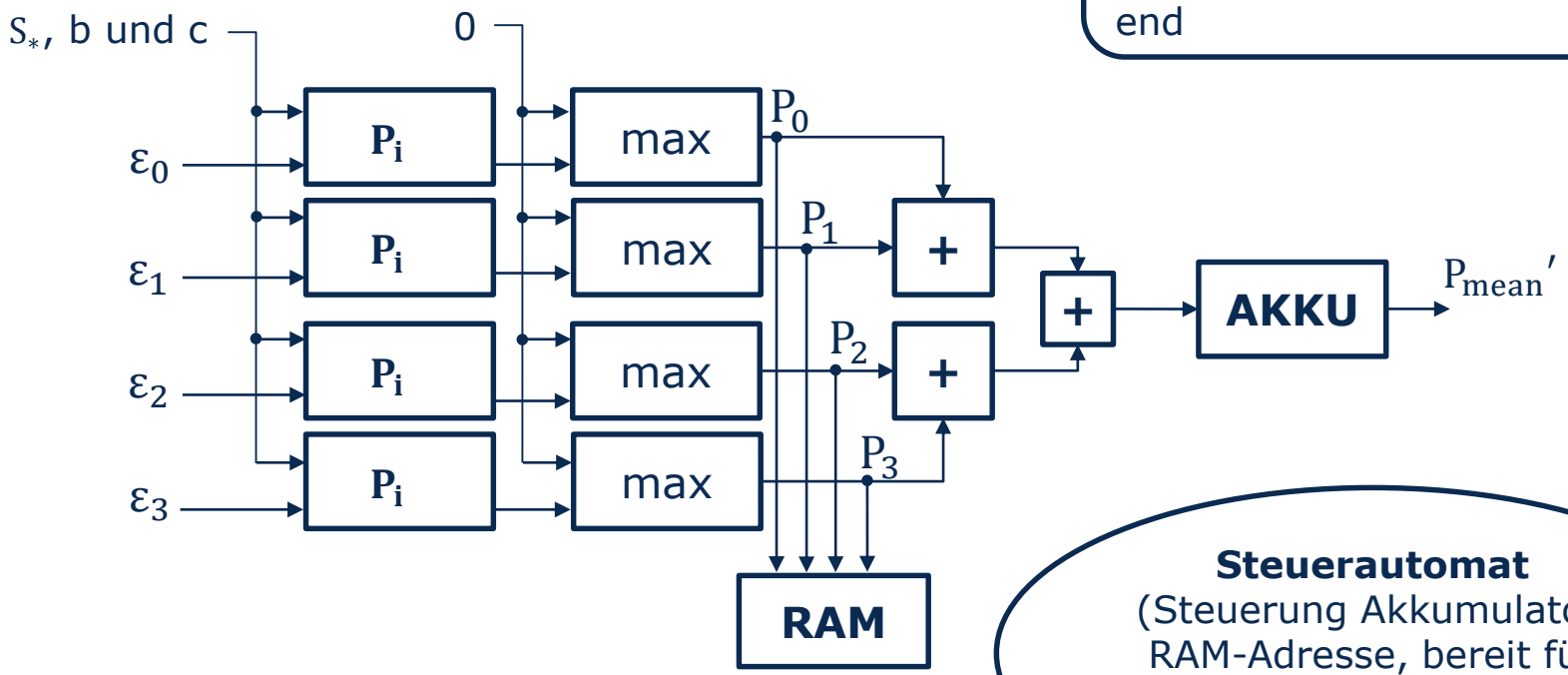
Loop 1

for i= 1 to M

set $P_i = \max\left(S_* e^{b\varepsilon_i} + c, 0\right)$

set $P_{mean} \mathrel{+}= P_i$

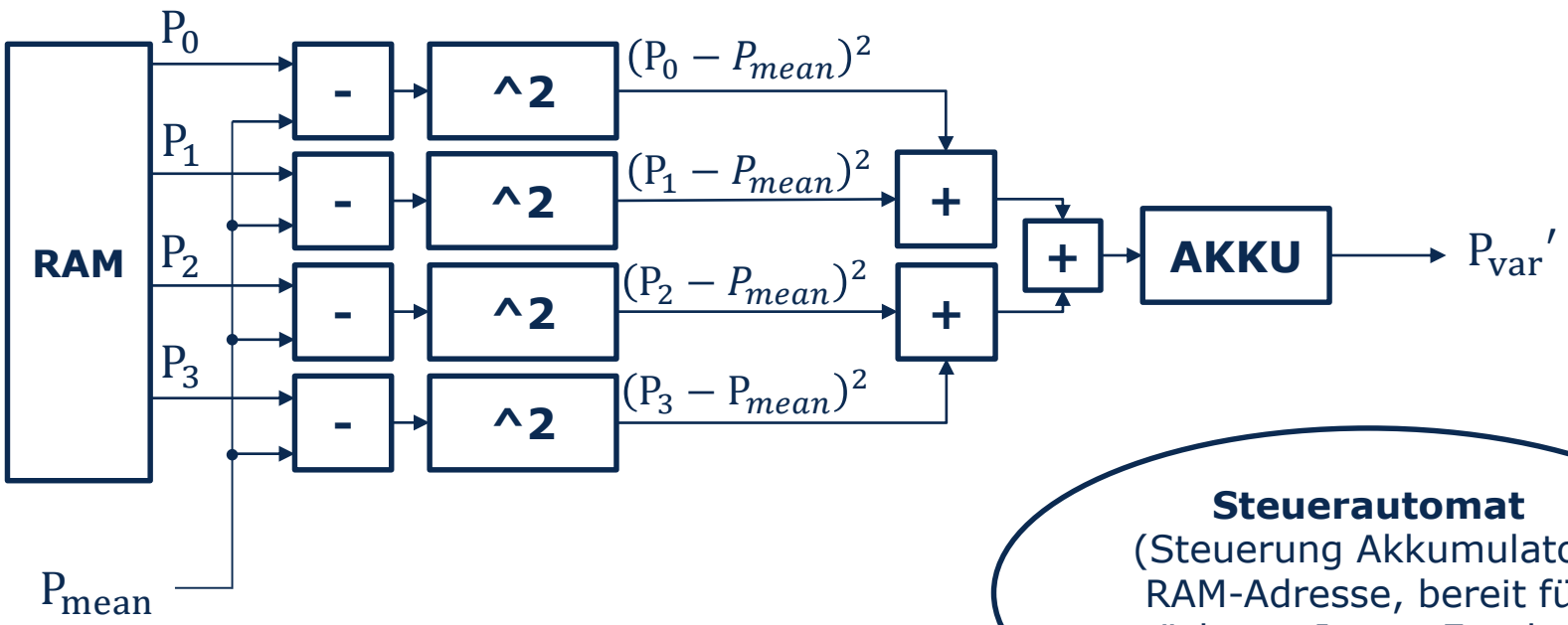
end



Steuerautomat
(Steuerung Akkumulator, RAM-Adresse, bereit für nächsten Input, Ergebnis bereit)

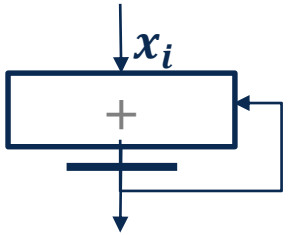
Loop 2

```
for i= 1 to M
        set P_var += (P_i−P_mean)^2
end
```



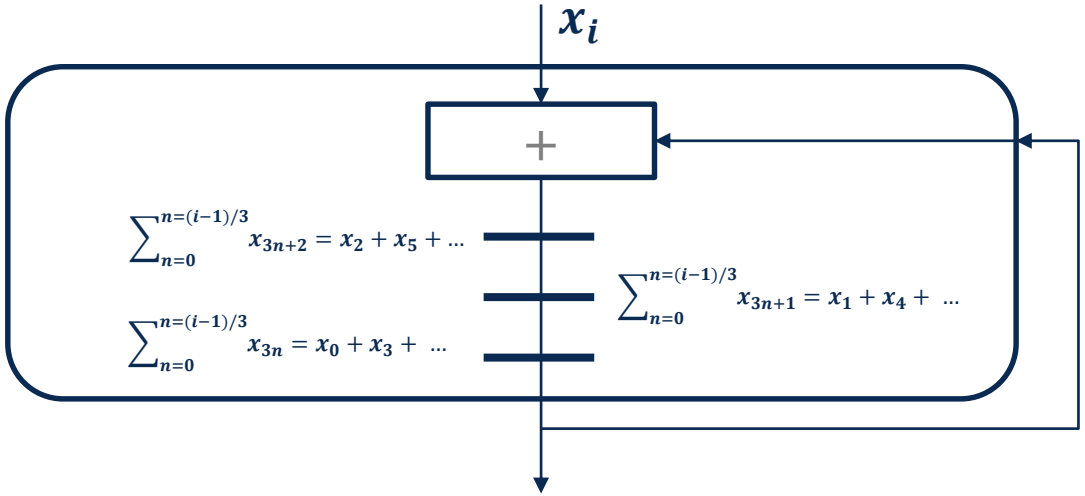
Steuerautomat
(Steuerung Akkumulator, RAM-Adresse, bereit für nächsten Input, Ergebnis bereit)

Floating Point Akkumulator



$$\sum_{n=0}^{n=i-1} x_n = x_o + x_1 + \ldots + x_{i-1}$$

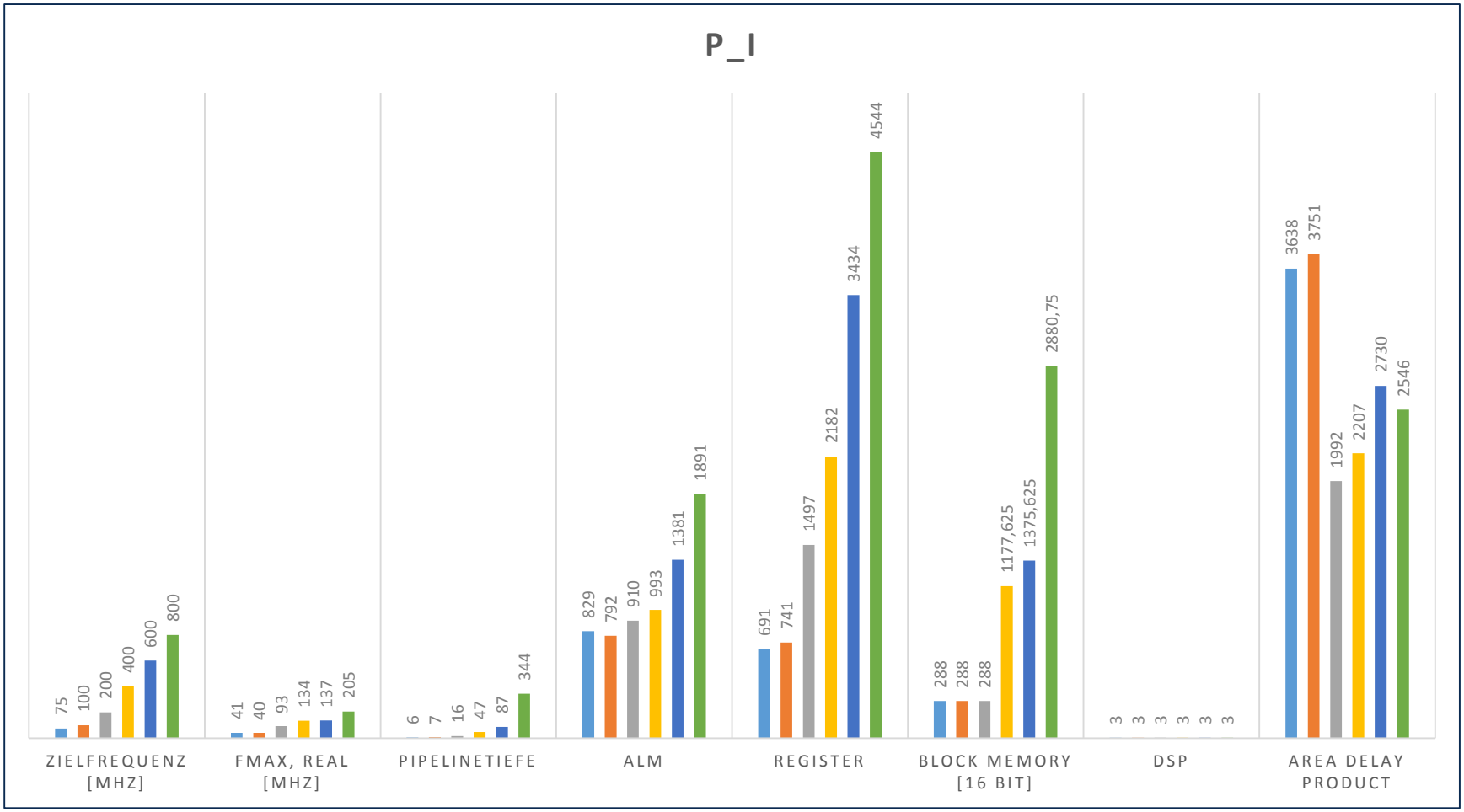
Problem: Floating Point Addierer nicht kombinatorisch:

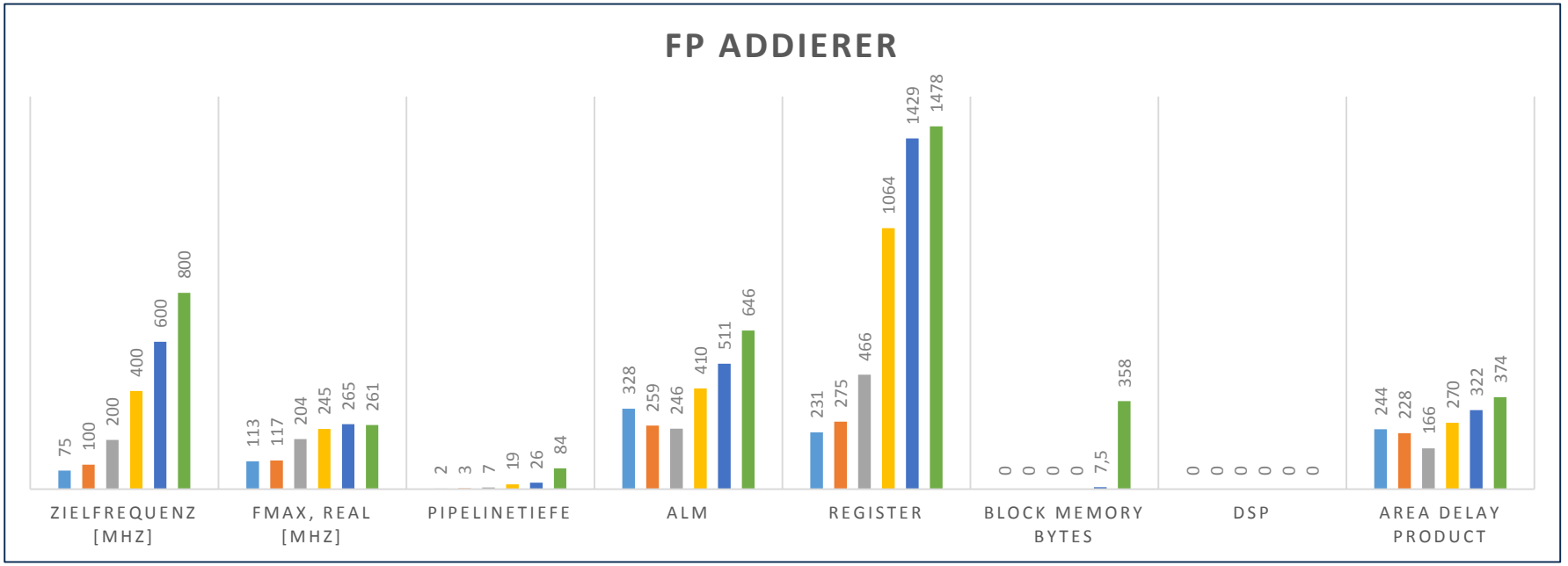


3.2 Auswahl der Komponenten

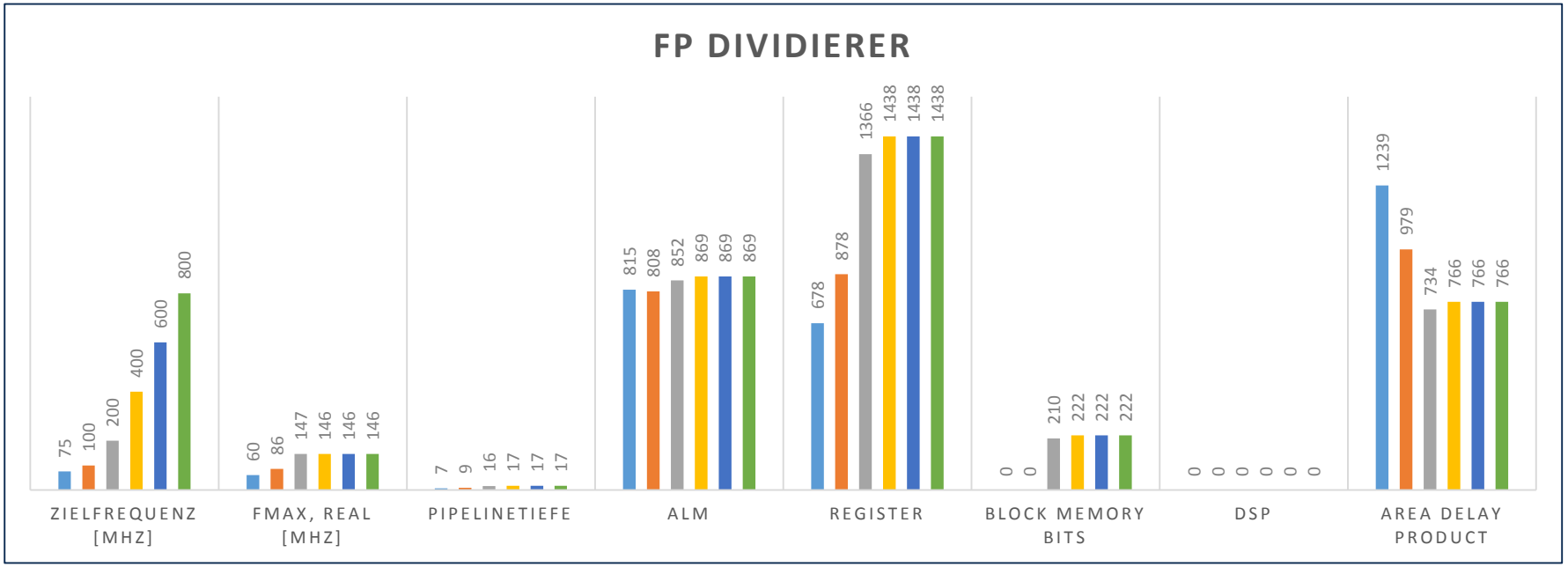
Bewertungskriterien

- f_{max}
- Ressourcenbedarf
- Area Delay Product:
  $\sqrt[4]{ALM \cdot Register \cdot Block\ Memory \cdot DSP} \cdot \frac{1}{f}$

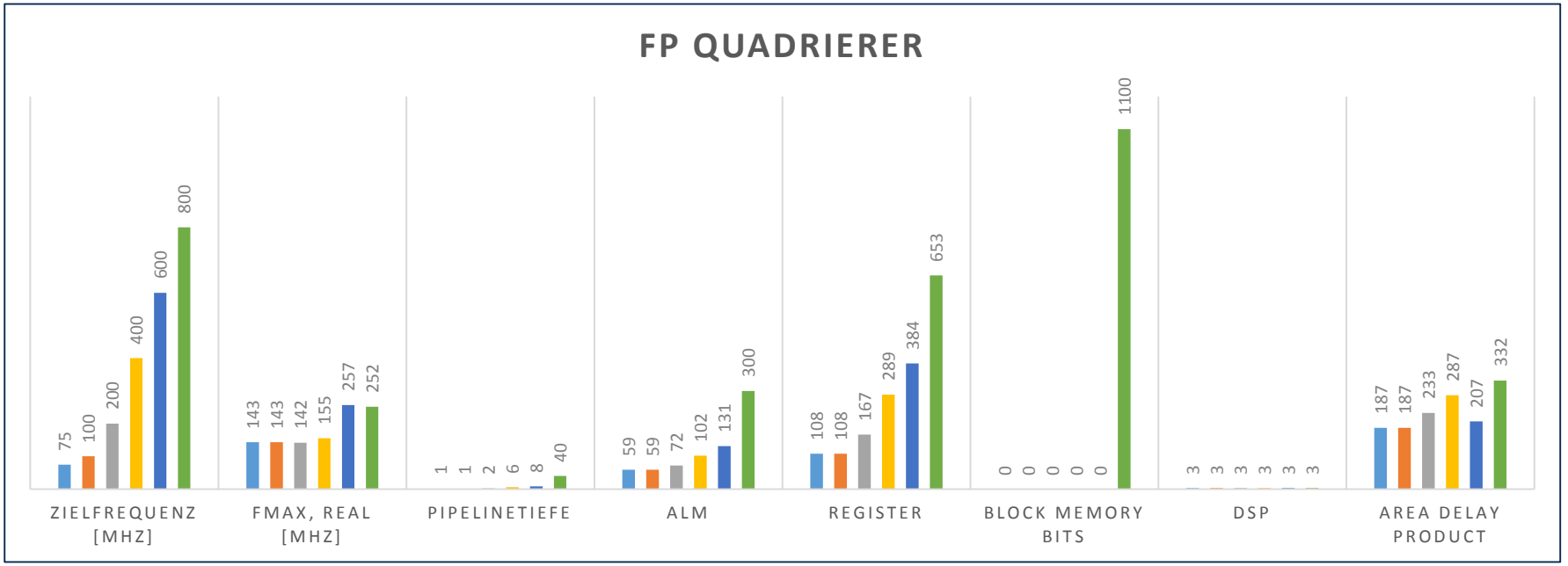

P_I
75
100
200
400
600
800
41
40
93
134
137
205
6
7
16
47
87
344
829
792
910
993
1381
1891
691
741
1497
2182
3434
4544
288
288
288
1177,625
1375,625
2880,75
3
3
3
3
3
3
3638
3751
1992
2207
2730
2546
ZIELFREQUENZ [MHZ]
FMAX, REAL [MHZ]
PIPELINETIEFE
ALM
REGISTER
BLOCK MEMORY [16 BIT]
DSP
AREA DELAY PRODUCT



	Pipelinetiefe	ALM	Register	DSP	f_{max} [MHz]	Area Delay Product
FloPoCo Addierer	7	246	466	0	204	1.66
Altera Addierer (1)	7	378	454	0	163	2.54
Altera Addierer (2)	14	456	786	0	209	2.86

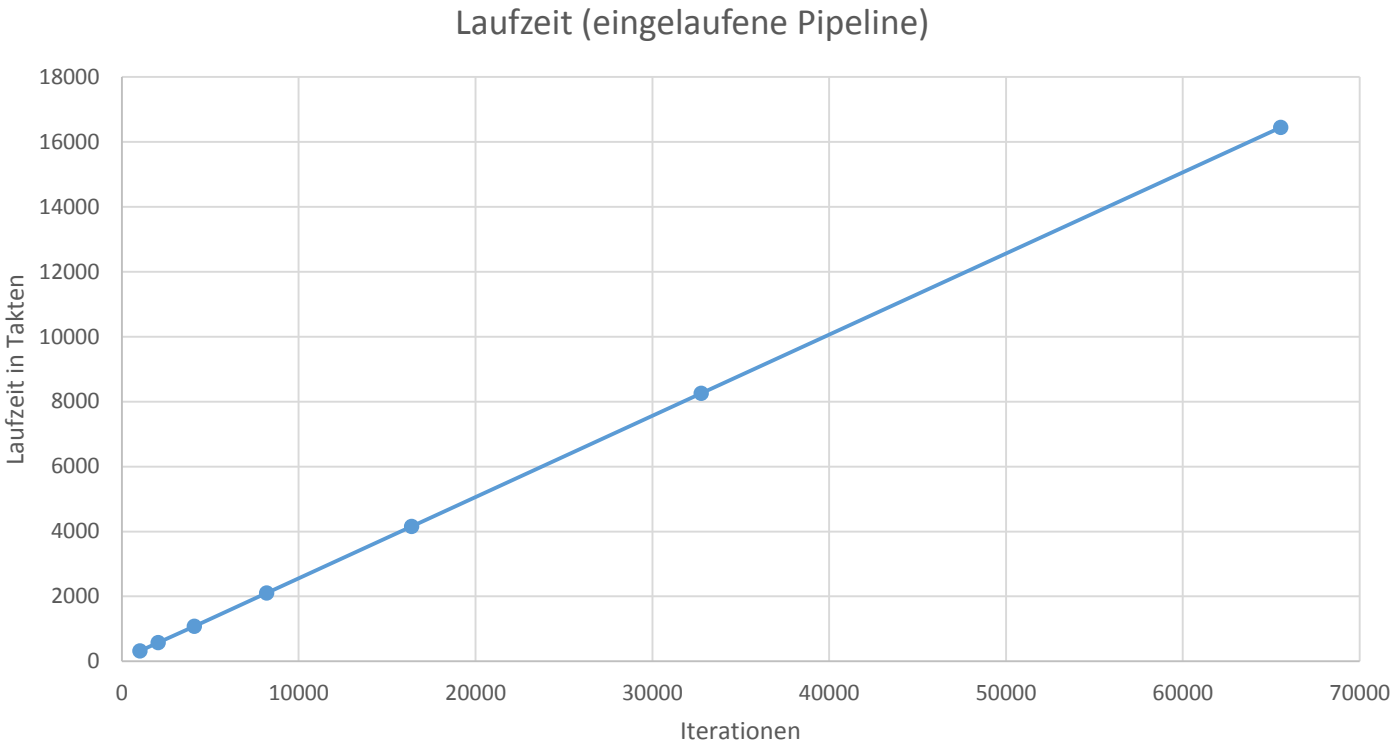


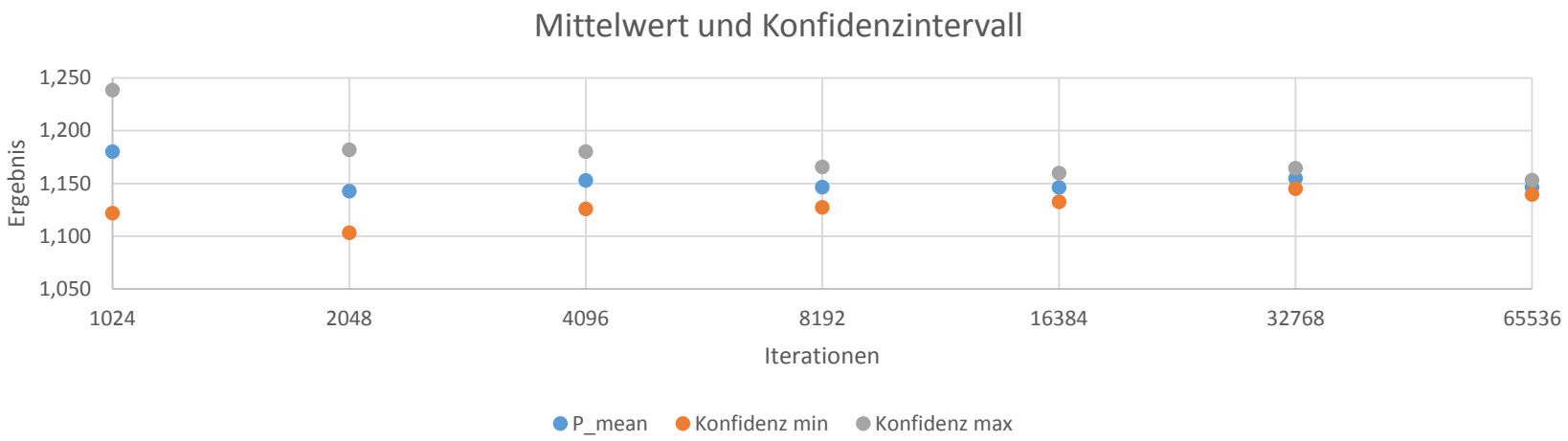
	Pipelinetiefe	ALM	Register	Block Memory Bits	DSP	f_{max} [MHz]
FloPoCo Dividierer	16	852	1366	210	0	147
Altera Dividierer	14	265	759	4608	6	243

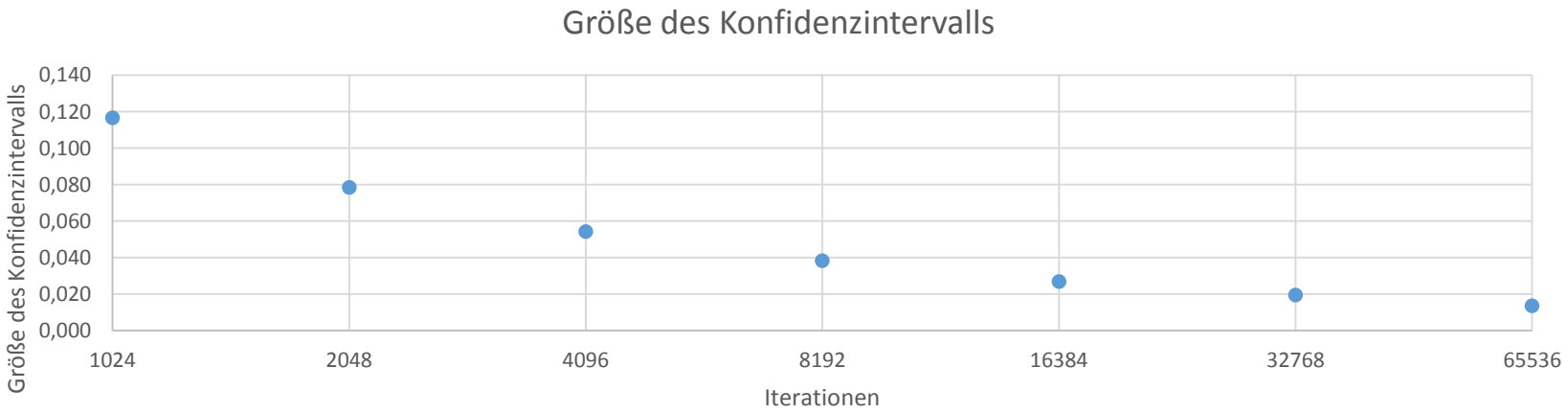


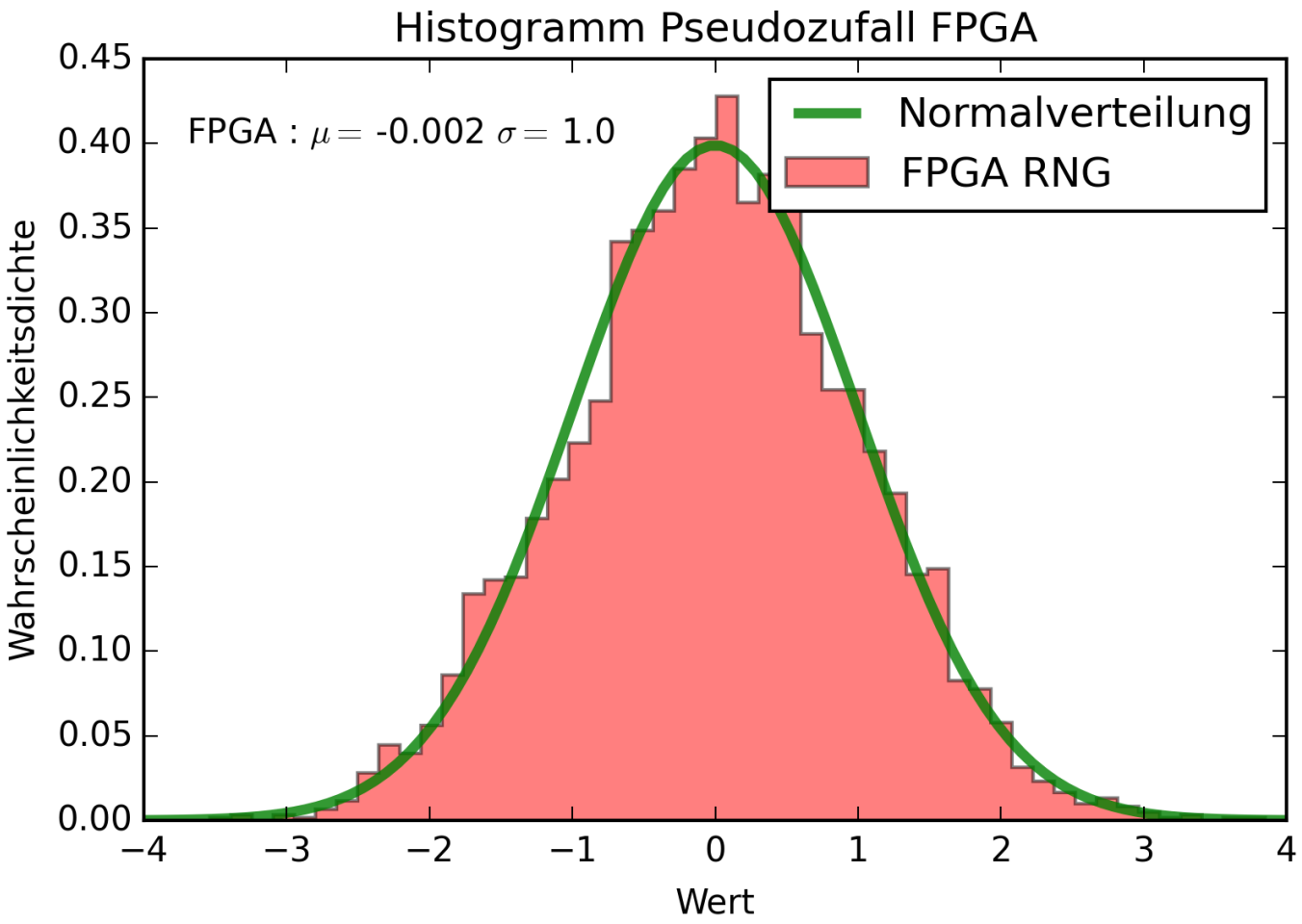
	Pipelinetiefe	ALM	Register	DSP	f_{max} [MHz]	Area Delay Product
FloPoCo Quadrierer	8	131	384	3	257	0.20
Altera Quadrierer	6	117	279	1	261	0.12

4. Ergebnisse




Mittelwert und Konfidenzintervall
Ergebnis
1,250
1,200
1,150
1,100
1,050
1024
2048
4096
8192
16384
32768
65536
Iterationen
P_mean
Konfidenz min
Konfidenz max


Größe des Konfidenzintervalls
Größe des Konfidenzintervalls
0,140
0,120
0,100
0,080
0,060
0,040
0,020
0,000
1024
2048
4096
8192
16384
32768
65536
Iterationen


Histogramm Pseudozufall FPGA
FPGA : $\mu =$ -0.002 $\sigma =$ 1.0
Normalverteilung
FPGA RNG
Wahrscheinlichkeitsdichte
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−4
−3
−2
−1
0
1
2
3
4
Wert

Ressourcenbedarf und Performance

	ALM	Register	Block Memory Bits	DSP
Ressourcenbedarf absolut	10119	13056	2162838	24
Ressourcenbedarf relativ	32 %	(keine Angabe)	53 %	28 %

- f_{max}: 91 MHz
- Performance auf GPU Niveau
- Spielraum für weiteres Loop Unrolling
- Block Memory Bedarf ungünstig

Problem: Speicher

- Maximal ca. 65000 Iterationen möglich

Lösungsmöglichkeiten

- Größerer FPGA
- SDRAM (4450 KBit vs 64 MB)
- Erneute Berechnung der Zwischenergebnisse (statt Speicherung)

Mögliche Optimierungen

- m konstant
- Zwischenergebnisse neu berechnen statt speichern
- $P_i = S_* \cdot e^{b \cdot \varepsilon_i} + c$ manuell erstellen und optimieren

Fazit

- Direkter Vergleich FloPoCo – Altera IP Cores: durchwachsen
- Implementierung im Vergleich zu CUDA konkurrenzfähig

Quellen

- Computing in Science & Engineering 6, 72 (2004); doi: 10.1109/MCSE.2004.62 http://dx.doi.org/10.1109/MCSE.2004.62
- Altera Floating-Point Megafunctions User Guide http://www.altera.co.jp/literature/ug/ug_altfp_mfug.pdf
- com.lang.c FAQ http://c-faq.com/lib/gaussian.html
- FloPoCo User Manual: http://flopoco.gforge.inria.fr/flopoco_user_manual.html