

Chipimplementation einer zweidimensionalen Fouriertransformation für die Auswertung eines Sensor-Arrays

Bachelorkolloquium

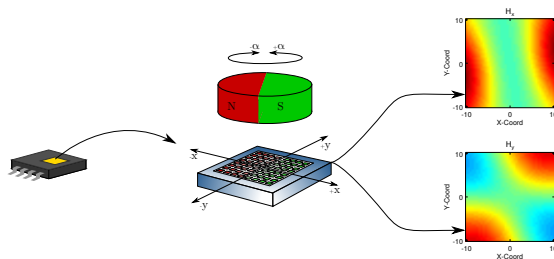
Thomas Lattmann

1. Mai 2018

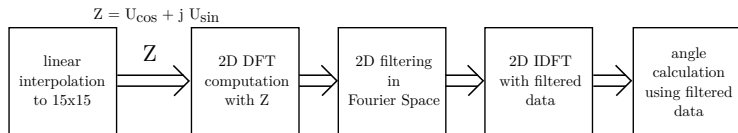
Gliederung:

- Einleitung
- Grundlagen
- Analyse und Entwurf
- Testumgebung und Evaluation
- Zusammenfassung und Ausblick

- 350 μm Prozess, EuroPractice
- Array von Magnetsensoren
- Sensoren, Signalverarbeitung & Ausgabe des digitalen Nutzsignals auf einem ASIC



Quelle: SensorAusgang.pdf, K.-R. Riemschneider + T. Schütze



Quelle: [Frequency_filtering_and_stray_field_compensation_using_2D-DFT_algorithm.pdf](#), K.-R. Riemschneider + T. Schütze

Gliederung:

- Interpretation von Dualzahlen
- Komplexe Multiplikation
- Matrixmultiplikation
- DFT und IDFT
- 2D-DFT

Mögliche Arten sind:

- positive Ganzzahldarstellung (a)
- Darstellung im Einerkomplement (b)
- Darstellung im Zweierkomplement (c)
- vorzeichenbehaftete Festkommazahlen (SQ-Format) mit u. ohne Vorkommaanteil (d)

Beispiel:

1001011010100₂

$$4096 + 512 + 128 + 64 + 16 + 4 = 4820_{10} \quad (a)$$

$$-(512 + 128 + 64 + 16 + 4) = -724_{10} \quad (b)$$

$$-4096 + 512 + 128 + 64 + 16 + 4 = -3372_{10} \quad (c)$$

$$-4 + 0,5 + 0,125 + 0,062 + 0,015625 + 0,00390625 = -3,29296875_{10} \quad \text{in S2Q10 (d)}$$

Mögliche Arten sind:

- positive Ganzzahldarstellung (a)
- Darstellung im Einerkomplement (b)
- Darstellung im Zweierkomplement (c)
- vorzeichenbehaftete Festkommazahlen (SQ-Format) mit u. ohne Vorkommaanteil (d)

Beispiel:

$$1001011010100_2$$

$$4096 + 512 + 128 + 64 + 16 + 4 = 4820_{10} \quad (a)$$

$$-(512 + 128 + 64 + 16 + 4) = -724_{10} \quad (b)$$

$$-4096 + 512 + 128 + 64 + 16 + 4 = -3372_{10} \quad (c)$$

$$-4 + 0,5 + 0,125 + 0,062 + 0,015625 + 0,00390625 = -3,29296875_{10} \quad \text{in S2Q10 (d)}$$

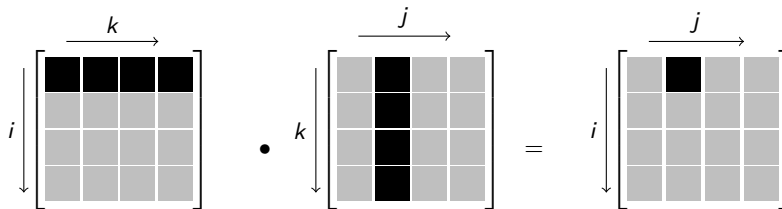
Komplexe Multiplikation sind 4 einfache Multiplikationen und 2 Additionen.

$$\begin{aligned}e + jf &= (a + jb) \cdot (c + jd) \\&= a \cdot c + j(a \cdot d) + j(b \cdot c) + j^2(b \cdot d) \\&= a \cdot c - b \cdot d + j(a \cdot d + b \cdot c)\end{aligned}$$

Wenn einer der beiden Multiplikanden keinen Imaginärteil haben, reduziert sich das zu

$$\begin{aligned}e + jf &= a \cdot (c + jd) \\&= a \cdot c + j(a \cdot d)\end{aligned}$$

Veranschaulichung der Matrixmultiplikation



DFT:

Summenschreibweise

$$X^*[m] = \frac{1}{N} \cdot \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi \frac{mn}{N}}$$

Matrixschreibweise

$$W = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} e^{-j2\pi \frac{mn}{N}}$$
$$X^* = W \cdot x$$

IDFT:

$$x[n] = \frac{1}{N} \sum_{m=0}^{N-1} X^*[m] \cdot e^{+j2\pi \frac{mn}{N}}$$

DFT:

Summenschreibweise

$$X^*[m] = \frac{1}{N} \cdot \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi mn/N}$$

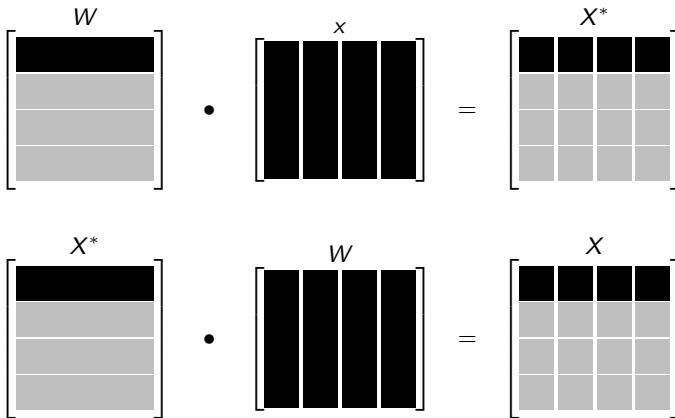
Matrixschreibweise

$$W = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} e^{-j2\pi mn/N}$$
$$X^* = W \cdot x$$

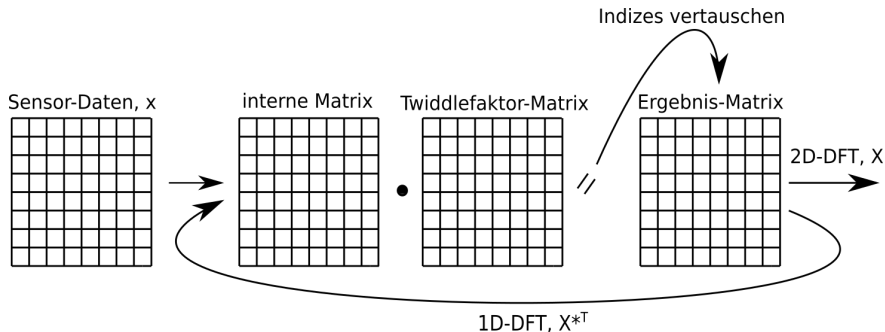
IDFT:

$$x[n] = \frac{1}{N} \sum_{m=0}^{N-1} X^*[m] \cdot e^{+j2\pi mn/N}$$

$$\begin{aligned} X &= W \cdot x \cdot W \\ &= X^* \cdot W \end{aligned}$$

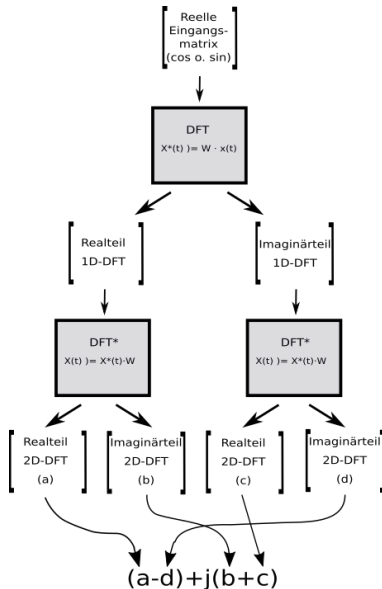


$$\begin{aligned} X &= \left((x \cdot W)^T \cdot W \right)^T \\ &= \left(X^{*T} \cdot W \right)^T \end{aligned}$$

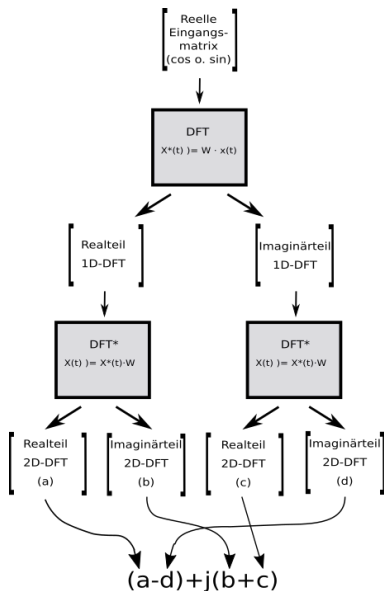


- optimierte Matrixmultiplikation mit reellen Eingangswerten
- optimierte Matrixmultiplikation mit komplexen Eingangswerten
- Fast Fouriertransformation (Butterfly-Algorithmus)
- allgemeine Matrixmultiplikation

2D-DFT mit reellen Eingangswerten zur Ausnutzung von Redundanzen



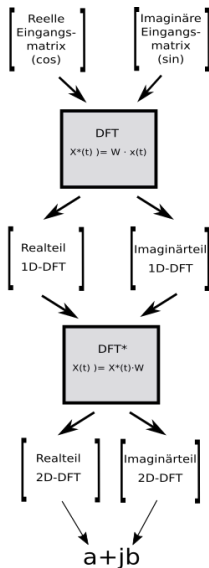
2D-DFT mit reellen Eingangswerten zur Ausnutzung von Redundanzen



	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6	A4	B4	C4	D4	E4	F4	G4	H4
7	A3	B3	C3	D3	E3	F3	G3	H3
8	A2	B2	C2	D2	E2	F2	G2	H2

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6	-A4	-B4	-C4	-D4	-E4	-F4	-G4	-H4
7	-A3	-B3	-C3	-D3	-E3	-F3	-G3	-H3
8	-A2	-B2	-C2	-D2	-E2	-F2	-G2	-H2

2D-DFT mit komplexen Eingangswerten



Methode	Anzahl reeller Multiplikationen
reelle Eingangswerte	64
komplexe Eingangswerte	128
FFT	128
allgemeine Matrixmultiplikation	4096

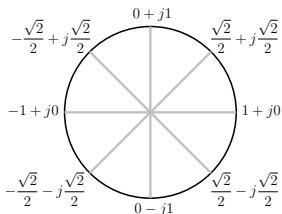
Gliederung:

- Gegenüberstellung verschiedener Größen von Twiddlefaktormatrizen
- Optimieren der 8×8 -DFT
- Konstantenmultiplikation
- Benötigte Takte
- Zustandsfolge
- Entwickeln der 2D-DFT auf Basis der 1D-DFT

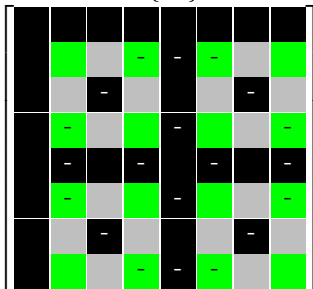
Gegenüberstellung verschiedener Größen von DFT-Twiddlefaktormatrizen

N	8	9	12	15	16
$N \times N$	64	81	144	225	256
trivial \Re	48	45	128	81	128
nicht triv. \Re	16	36	16	144	128
triv. \Im	48	21	96	45	128
nicht triv. \Im	16	60	48	180	128
\sum triv.	96	66	224	126	256
\sum nicht triv.	32	96	64	324	256
Anzahl verschiedener nicht trivialer Werte	1	7	1	13	3
Verhältnis \sum trivial / \sum nicht trivial	3	0,6875	3,5	0,3889	1

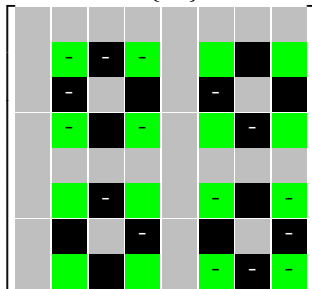
Optimierung der 8x8-DFT



$Re\{W\}$

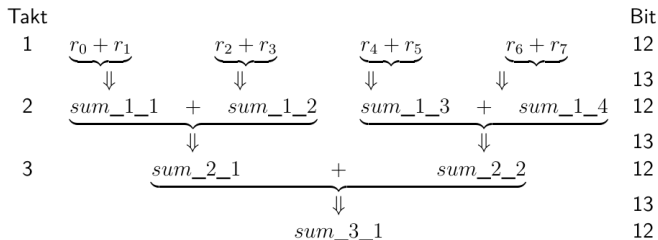


$Im\{W\}$

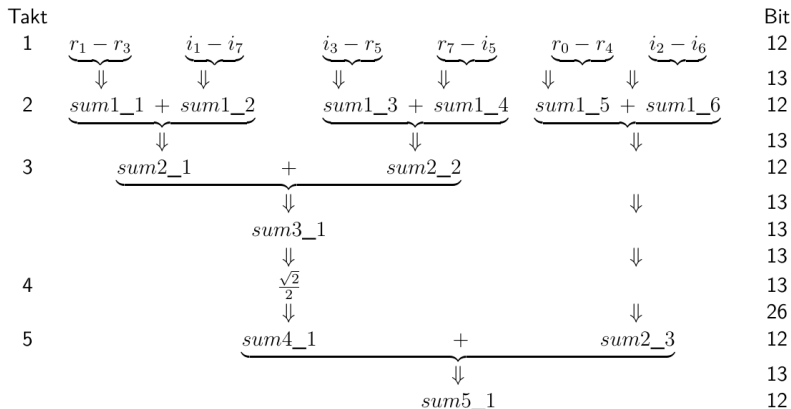


Legende: $\blacksquare = 1$ $\blacksquare = -1$ $\square = 0$ $\color{red}\blacksquare = \sqrt{2}/2$ $\color{red}\blacksquare = -\sqrt{2}/2$

1. Spalte: $r_0 + r_1 + r_2 + r_3 + r_4 + r_5 + r_6 + r_7$



2. Spalte: $r_1 - r_3 + i_1 - i_7 + i_3 - r_5 + r_7 - i_5 + r_0 - r_4 + i_2 - i_6$

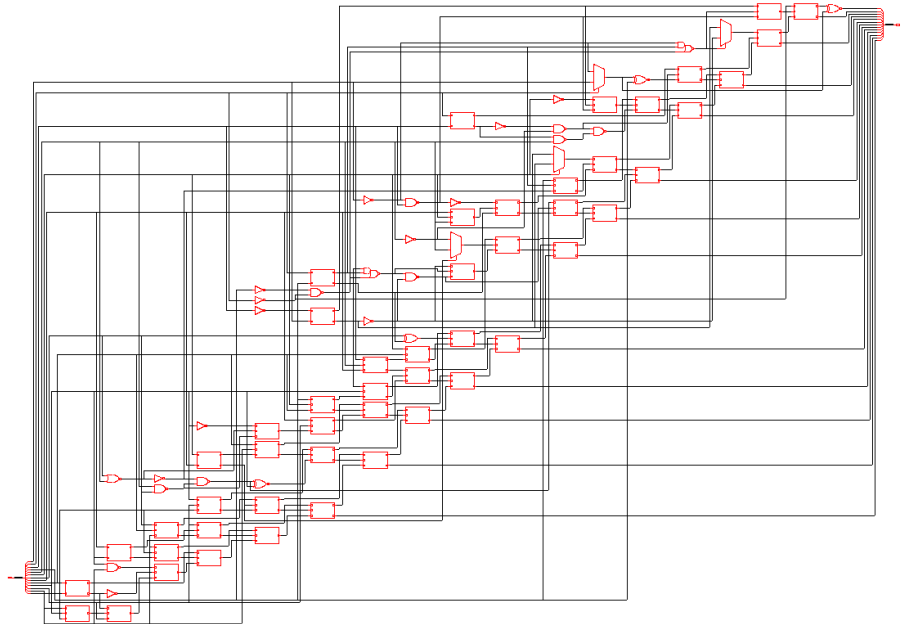


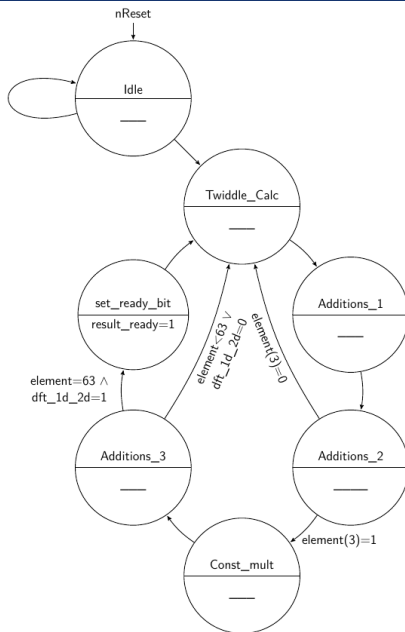
Summe der Takte für die Berechnung der 2D-DFT

Zeile	Additionen pro Element (N)	Takte pro Element ($\log_2(N)$)	Takte für Multiplikation	Summe der Takte
1	8	3	0	3
2	12	3,6	1	5
3	8	3	0	3
4	12	3,6	1	5
5	8	3	0	3
6	12	3,6	1	5
7	8	3	0	3
8	12	3,6	1	5

$$\Rightarrow \text{Summe der Takte ist } \underbrace{3 \cdot 4}_{\substack{\text{ungerade} \\ \text{Zeilen} \\ \text{aus} \\ W}} + \underbrace{5 \cdot 4}_{\substack{\text{gerade} \\ \text{Zeilen} \\ \text{aus} \\ W}} \cdot \underbrace{8}_{\substack{\text{alle} \\ \text{Spalten} \\ \text{aus} \\ \text{Input}}} \cdot \underbrace{2}_{\substack{\text{1D-DFT} \\ \text{und} \\ \text{2D-DFT}}} = 512$$

Konstantenmultiplikation mit $\frac{\sqrt{2}}{2} \simeq 0.70703125 = 0001011010100_2$

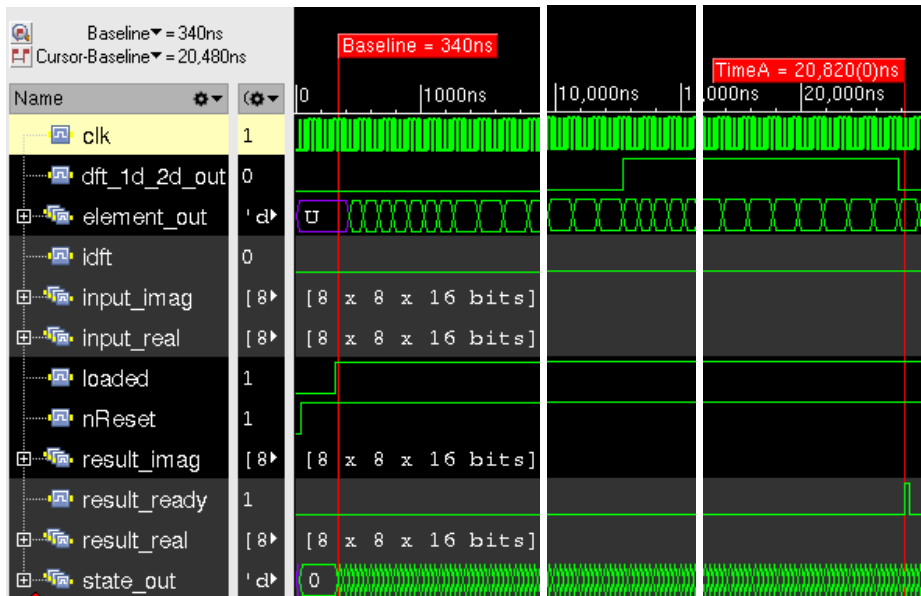




Gliederung:

- Testumgebung
- Zeitabschätzung
- Chipimplementation

- Simulation mit NC Sim und SimVision
 - ▶ nützlich für Teilfunktionen
 - ▶ Betrachtung einzelner Signalverläufe
- Automatisierung durch Shell-Skript
 - ▶ Simulation mit NC Sim und TCL-Skript
 - ▶ Berechnung mittels Matlab
 - ▶ Vergleich



Edge counts for :clk:

Number of posedges	512
Number of negedges	512
Number of edges	1024
Time high	10,240ns
Time low	10,240ns
Duty cycle	50%

Ok

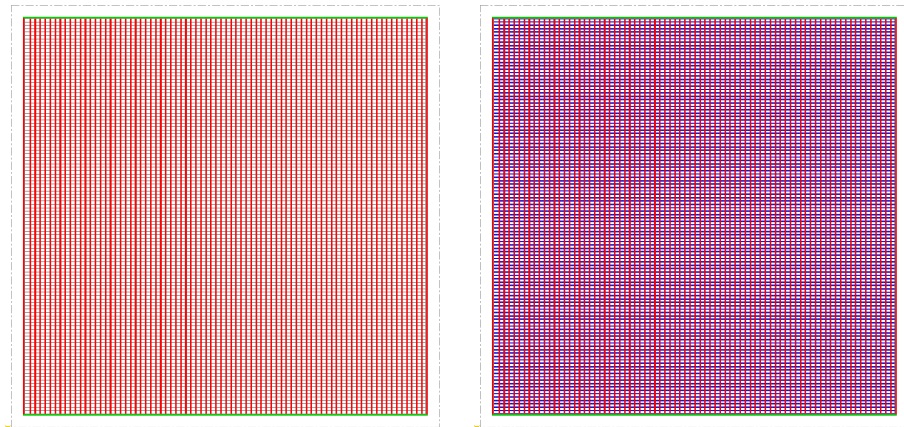
Gegeben: Systemtakt: 100 MHz, max. Drehzahl: 8000min⁻¹, Auflösung: 1°

$$\frac{8000 \text{ min}^{-1}}{60} = 133, \bar{3} \text{ sec}^{-1}$$

$$\curvearrowright 1 \text{ Umdrehung} = \frac{1}{133, \bar{3} \text{ sec}^{-1}} = 7,5 \cdot 10^{-3} \text{ sec}$$

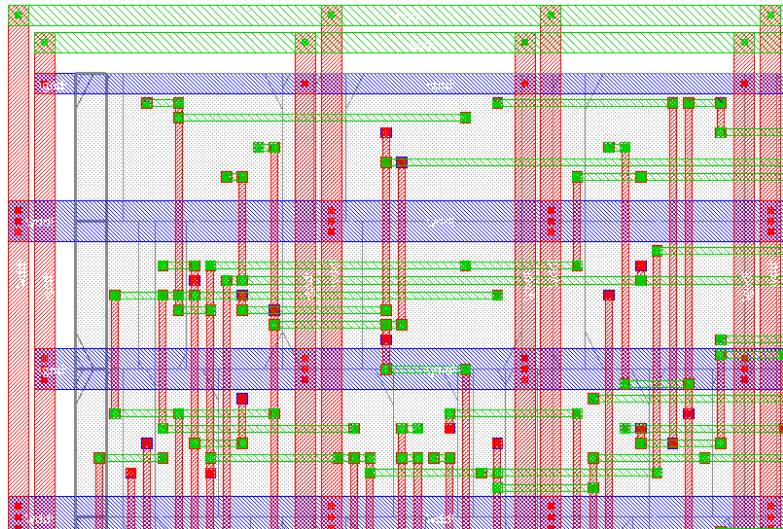
$$1^\circ \hat{=} \frac{7,5 \cdot 10^{-3} \text{ sec}}{360} = 20,83 \cdot 10^{-6} \text{ sec}$$

$$20,83 \cdot 10^{-6} \text{ sec} \cdot 100 \text{ MHz} = 2083 \text{ Takte}$$

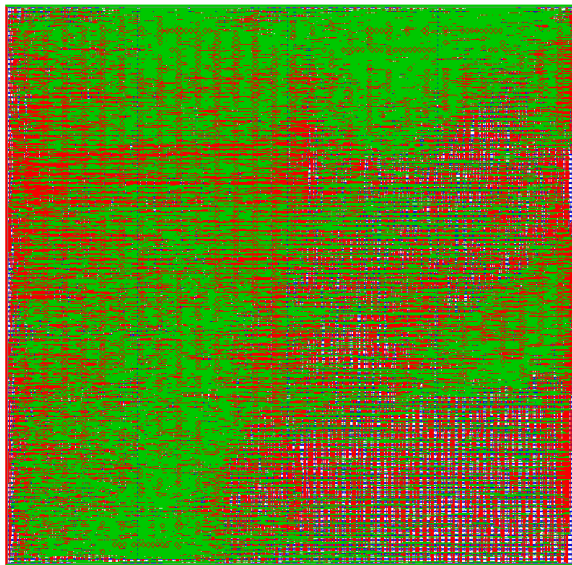


blau: Layer 1, rot: Layer 2, grün: Layer 3

Chipimplementation: Platzierung der Standardzellen



blau: Layer 1, rot: Layer 2, grün: Layer 3



Standardzellen:

15 310

Fläche:

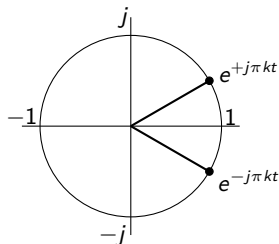
$1\,524\,960\mu\text{m}^2 = 1,5\text{mm}^2$

Prozess:

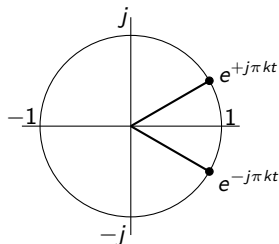
$350\mu\text{m}^2$

$$W = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} e^{-\frac{j2\pi mn}{N}}$$

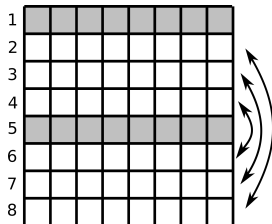
$$W^* = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} e^{+\frac{j2\pi mn}{N}}$$



$$W = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} e^{-\frac{j2\pi mn}{N}}$$



$$W^* = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} e^{+\frac{j2\pi mn}{N}}$$



- DFT als 8×8 hat sich als effizient erwiesen
- Optimierung der Multiplikationen mit der Twiddlefaktormatrix
- Kritischer Pfad scheint Konstantenmultiplikation zu sein
- Berechnung der 1D- und 2D-DFT mit selber Einheit
- Benötigte Takte liegen im realistischen Rahmen
- DFT und IDFT benötigen zusammen etwa 50% der verfügbaren Takte
- IDFT kann durch geringe Ergänzungen berechnet werden
- Wertvolle Grundlagen für die Implementation der 15×15 2D-DFT

- Reduzierung des kritischen Pfades
 - ▶ auf zwei Schaltnetze aufteilen
 - ▶ Wallace-Tree verwenden
- 15x15 mit ähnlich vielen Takten