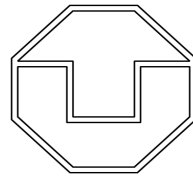


TECHNISCHE UNIVERSITÄT DRESDEN  
FAKULTÄT ELEKTROTECHNIK  
INSTITUT DER GRUNDLAGEN DER ELEKTROTECHNIK UND ELEKTRONIK

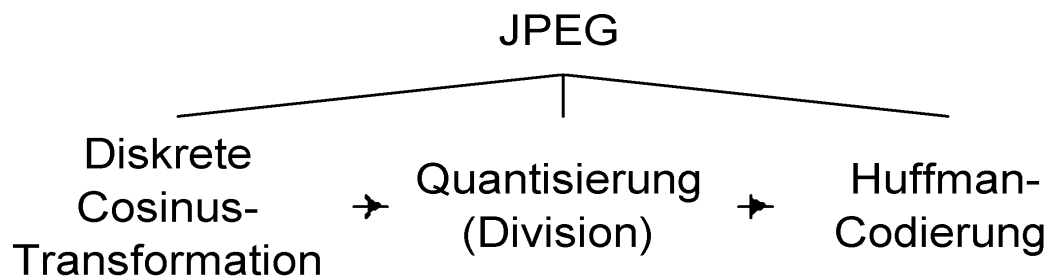
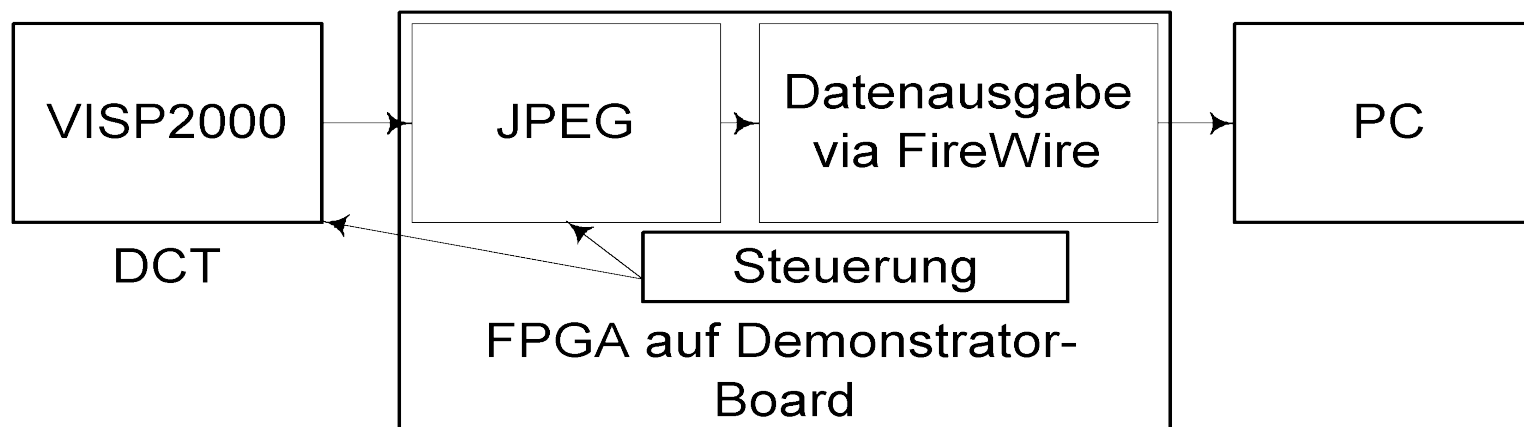


Präsentation der Studienarbeit

# FPGA-Implementierung der Quantisierung und Codierung des JPEG-Kompressionsalgorithmus

Ralf Hildebrandt

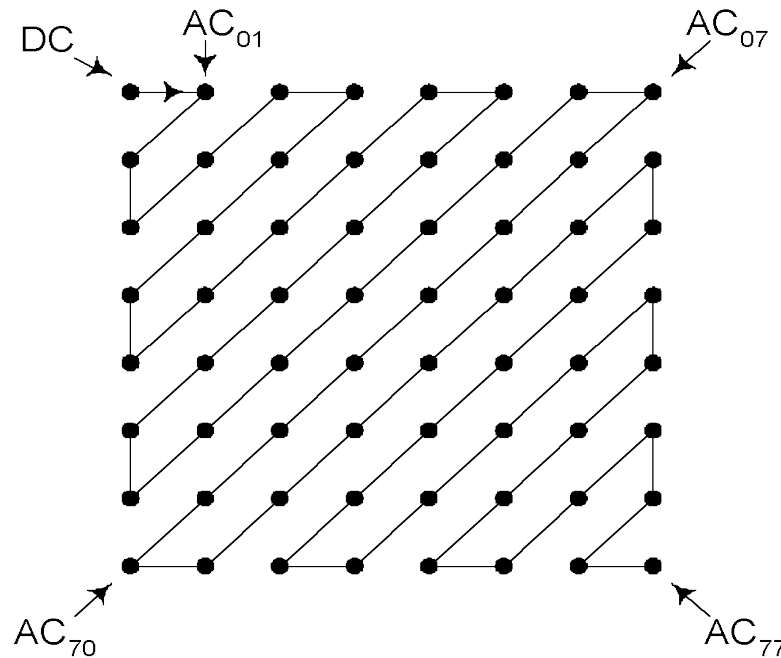
# Problemüberblick



# Die DCT

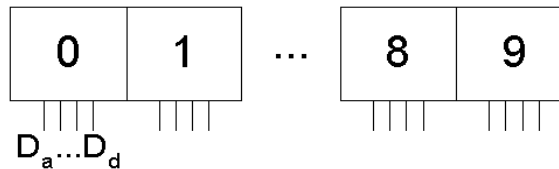
$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}$$

$$C(u) = C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{wenn } u, v = 0 \\ 1 & \text{sonst} \end{cases}$$

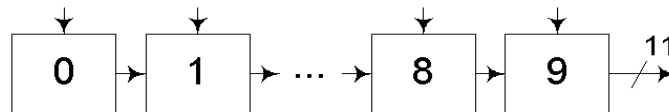


## Das Ausgabeschema des VISP2000

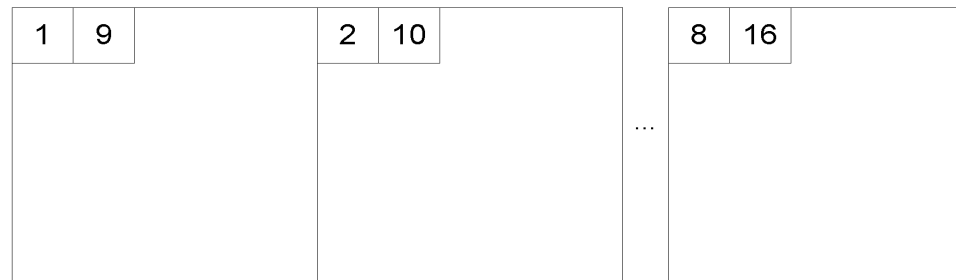
- ADC des VISP2000 arbeitet pipelined (10 Stufen)
- Ausgang: 2x4 Komparatorergebnisse



- Realisierung der RSD-Korrektur (10 Stufen) extern - in Hardware



- Die so errechneten Zahlenwerte (Helligkeitsinformation bzw. hier: DCT-Koeffizienten) erhält man in der Reihenfolge des folgenden Schemas:

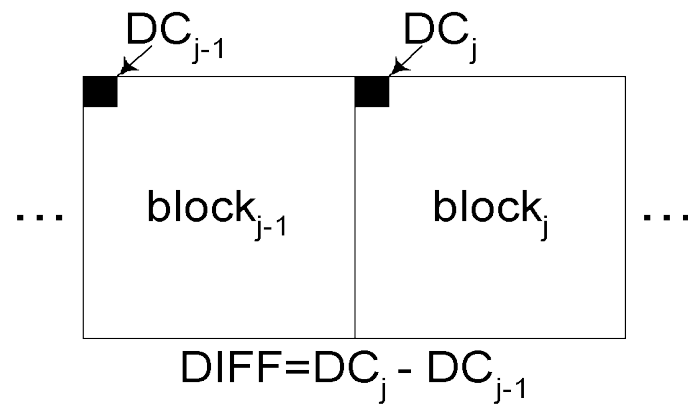


# Die Quantisierung

$$F^Q(u, v) = \text{integer round} \left( \frac{F(u, v)}{Q(u, v)} \right)$$

- 64 Quantisierungskoeffizienten  $Q(u, v)$
- Realisierung als Multiplikation mit den Inversen der Quantisierungskoeffizienten
- Abspeichern der 64 invertierten Quantisierungskoeffizienten
- Nutzung des Wallace-Tree-Multiplizierer-Prinzips zum Erreichen der notwendigen Geschwindigkeit bei moderater Fläche

# Die Codierung



Kategorie	Wertebereich
0	0
1	-1 ; 1
2	-3,-2 ; 2,3
3	-7,...,-4 ; 4,...,7
...	...
11	-2047,...,-1024 ; 1024,...,2047

## Die Wechselkoeffizienten:

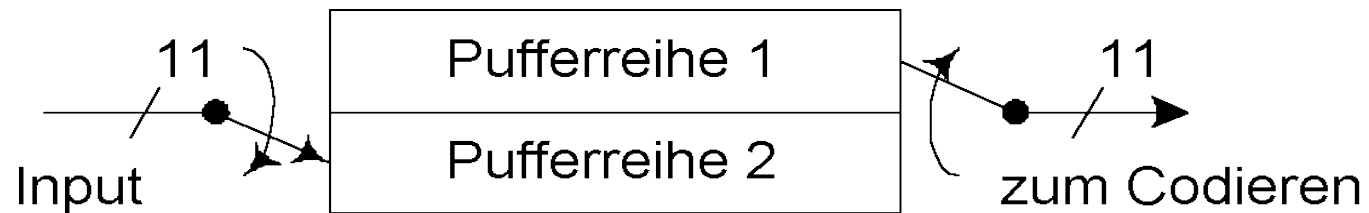
Zig-Zag Index	1	2	3	4	5	6	7	8	9	10	11	...	63
AC-Coefficient	0	0	0	0	-14	0	0	+1	0	0	0	...	0

run=4  
Kategorie=4
run=2  
Kategorie=1

- DC: Vergabe eines Codewortes für jede Kategorie
- AC: Vergabe eines Codewortes für jede Kombination aus „run“ und Kategorie
- Anhängen so vieler Bits, wie die Kategorie groß ist
- festes Huffman-Codewortalphabet

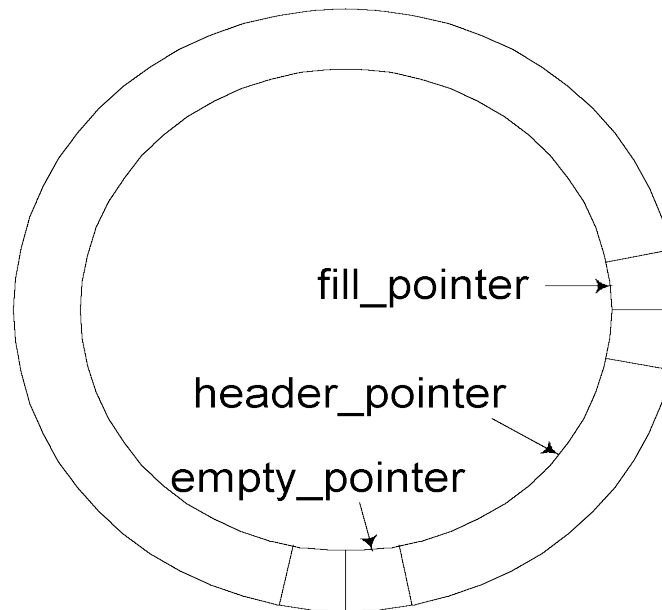
## Prinzip der Realisierung der Codierung:



- Gleichzeitiges Lesen und Schreiben => dual-port-RAM
- 2 Bildreihen, bestehend aus 8x8-Blöcken gepuffert => 11264 Bit
- sequentielles Durchlaufen des Puffers => Verwendung eines schnelleren Taktes, als der „Pixeltakt“ (Der Takt wird durch die Ansteuerung des VISP2000 erzeugt und ist um den Faktor 10 höher als der „Pixeltakt“.)

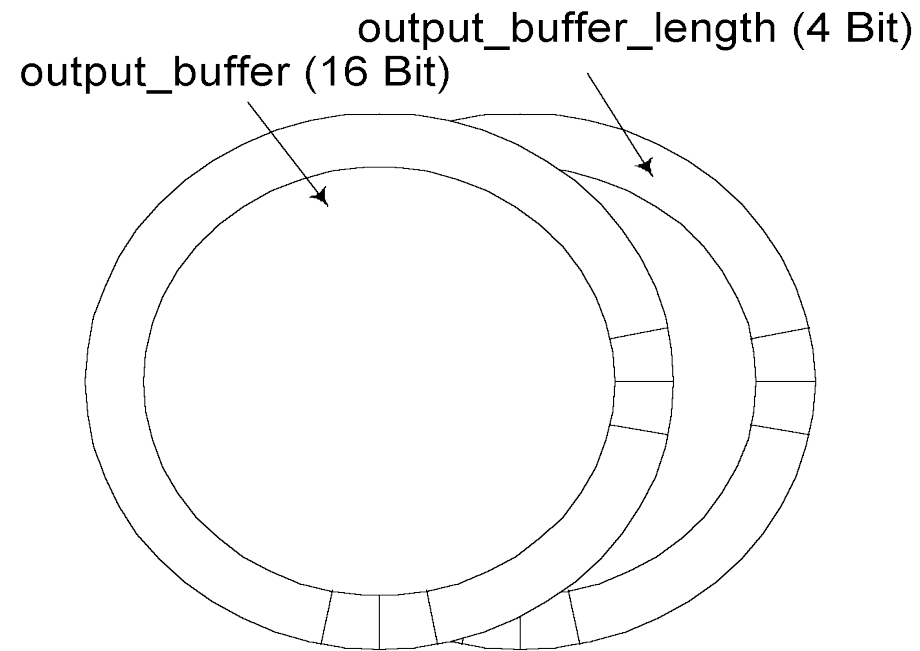


# Ausgangspufferung



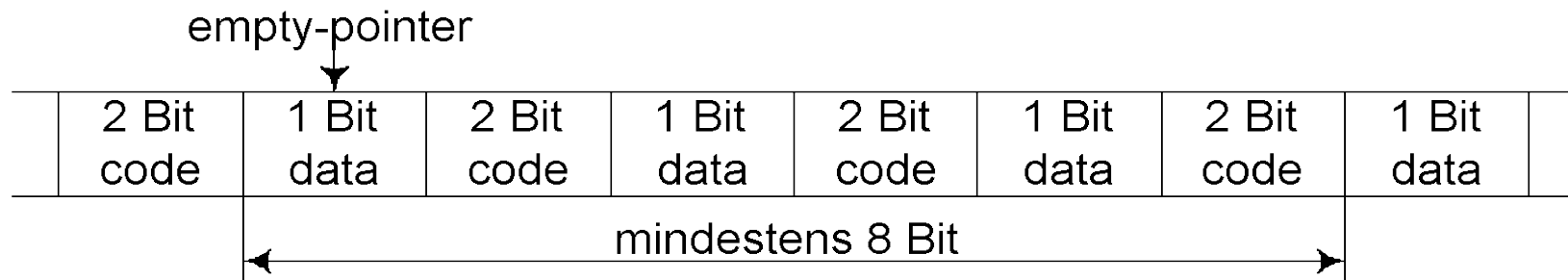
- Segment => „Schublade“ mit 16 Bit Wortbreite
- Speicherung des Codewortes („run“/Kategorie) in einer Schublade und der zusätzlichen Bits in einer anderen => dual-port-RAM nötig, da gleichzeitig

- Notwendigkeit zur Speicherung der Information, wie viele Bits in einer „Schublade“ gültig sind



## Zur Datenausgabe

- Übergabe von 8 Bit parallel an die Ausgabestufe für die FireWire-Schnittstelle
- worst case bei der Puffer-Füllung:



- => Größe des Ring-Puffers: 512 „Schubladen“ (16 Bit Daten und 4 Bit Information, wie viele Bits gültig sind => 10240 Bit)

## Prinzipien bei der Realisierung

- Flip-Flop basiertes Design, da die FPGA nur Flip Flops und keine Latches zur Verfügung stellt
- Eingangsstufe stellt Synchronisationssignale für alle nachfolgenden Stufen bereit
- Verwendung von Block-RAM auf der FPGA, welcher in variabler Konfiguration zur Verfügung steht und damit Möglichkeit zur Verwendung von dual-port-RAM
- feste Quantisierungskoeffizienten und festes Huffman-Codewortalphabet