

Fakultät Elektrotechnik und Informationstechnik, Vodafone Stiftungslehrstuhl für Mobile Nachrichtensysteme

Verteidigung der Diplomarbeit:

Analyse und Entwicklung einer Befehlssatzerweiterung für die Beschleunigung von Bitmap-Indizes großer Datenbanksysteme

Erik Laux

Betreut durch: O. Arnold, T. Karnagel und B. Schlegel

Unter der Leitung von: Prof. G. Fettweis,

Prof. W. Lehner



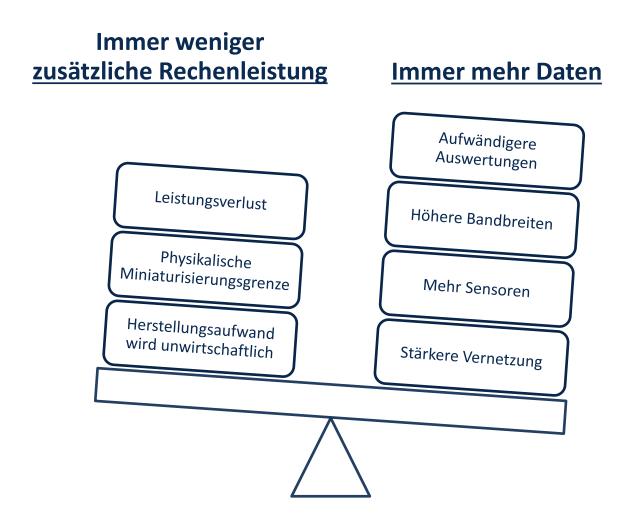


Gliederung der Präsentation

- 1. Einleitung und Motivation
- 2. Vergleich der Kompressionsalgorithmen
- 3. Detailbetrachtung für WAH
- 4. Experimentelle Resultate
- 5. Zusammenfassung und Ausblick



Leistungsoptimierung trotz physikalischer Grenzen





Mögliche Ansätze zum Lösen des Konfliktes

Anforderungen

- Energieverbrauch
- Latenz
- Speicherbandbreite
- Performance

Architekturansätze Spezialisierter Beschleuniger Koprozessor EinprozessorSystem Multi-Core Many-Core Anzahl Rechenkerne

Inhalt der Diplomarbeit:

Beschleunigung einer datenintensiven Anwendung im DB Bereich:

Bitmap Index – Fokus: Verarbeitung

- Parallelisierung auf Operationsebene mittels SIMD
- Hardware/Software Co-Design Ansatz mit spezialisierten Instruktionen



Ursprungsrelation:

Name	Geschlecht	Beruf			
Arthur	männlich	Student			
Elizabeth	weiblich	Professor			
Julia	weiblich	Student			
Robert	männlich	Gärtner			



Bitmap Indizes (2 von 5):

lst Männlich?	lst Student?
1	1
0	0
0	1
1	0

Datenverarbeitung mit einfachen Bitoperationen → Einsatz in großen DBMS

Problem: Speicheraufwand bei hoher Attributkardinalität

Lösung in DBMS: Kompression



2. Vergleich der Kompressionsalgorithmen Run-Length Encoding als Grundlage

- Verlustfreier Kompressionsalgorithmus
- Geeignet um längere Bitfolgen zu komprimieren
- Geeignet um neue Bitfolgen "anzuhängen"
- **Prinzip**: gleiche Bits werden zusammengefasst
 - → Lange Folgen werden effizient komprimiert
 - → Nur bei Veränderungen mehr Speicherplatz verwendet
- Beispiel: 0000 0000 1111 0000 0000 0000 → 0-8 1-4 0-12
- Wünschenswert: Möglichkeit zur direkten Weiterverarbeitung

2. Vergleich der Kompressionsalgorithmen

Word-Aligned Hybrid Compression (WAH)

Vektor mit relevanten Bitpositionen:

31	30	290
----	----	-----

Codeworte:

- 1. Literal 31 Bit Nullen und Einsen gemischt 31:0
- 2. 1-Fill Folge von 31 Einsen 31:1, 30:1
- 3. 0-Fill Folge von 31 Nullen 31:1, 30:0

Name	Beispiel	WAH Version
Literal	0x01004113	0x01004113
1-Fill	0x7FFFFFFF	0xC000001
0-Fill	0x0000000	0x80000001



2. Vergleich der Kompressionsalgorithmen

<u>Position List Word-Aligned Hybrid Compression (PLWAH/Concise)</u>

Vektor mit relevanten Bitpositionen:

31 30 29...25 24...0

WAH Abwandlung:

Zusätzlich "Sparse" – spezielles Literal mit einem gesetzten Bit
→ Position wird, sofern vorhanden, in Bits 29...25 der Fills integriert

Beispiel – Sparse folgt auf 0-Fill:

Vorher: 0 Fill ---- 0x80000001

Eingangsvektor: 0x00000002 → Bitposition: 2

→ Bitshift: 1

Ausgangsvektor(bin): 10000010 00000000 00000000 00000001



2. Vergleich der Kompressionsalgorithmen

Compressed Adaptive Index Format (COMPAX)

Vektor mit relevanten Bitpositionen:

31	30	29	280
----	----	----	-----

WAH Abwandlung:

Kein 1-Fill, dafür Spezialwort für LFL und FLF

Codeworte:

1. Literal 31:1

2. 0-Fill 31:0, 30:0

3. Literal-0-Fill-Literal 31:0, 30:0, 29:1

4. 0-Fill-Literal-0-Fill 31:0, 30:1, 29:0



Tensilica Instrution Extrension - Design Flow

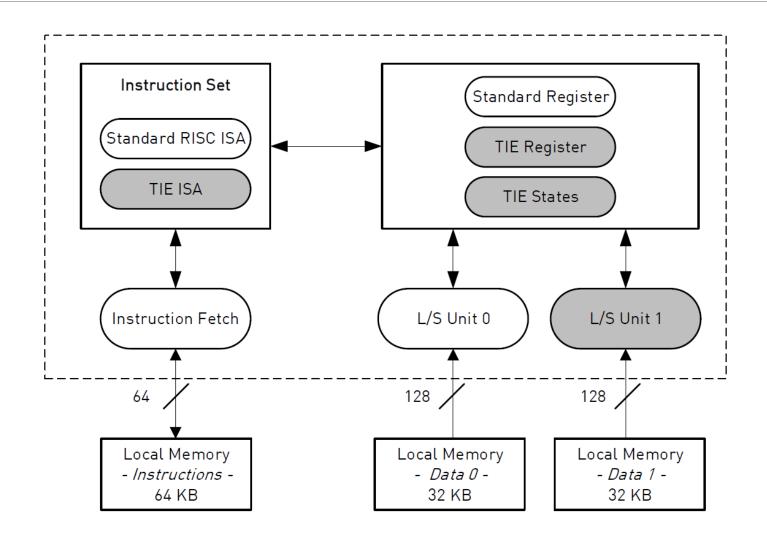
- Profiling
- Identifizierung von Hot-Spots

- TIE Instruktionen
- Optimierung von Fläche und Geschwindigkeit

- Synthese
- Optimierung des kritischen Pfades





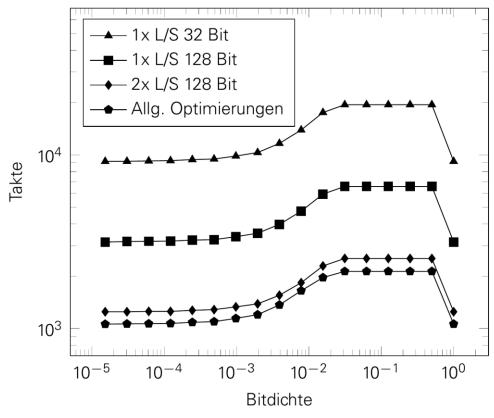




Fortschritt der Implementierungsvariante Vergleich der Entwicklungsversionen

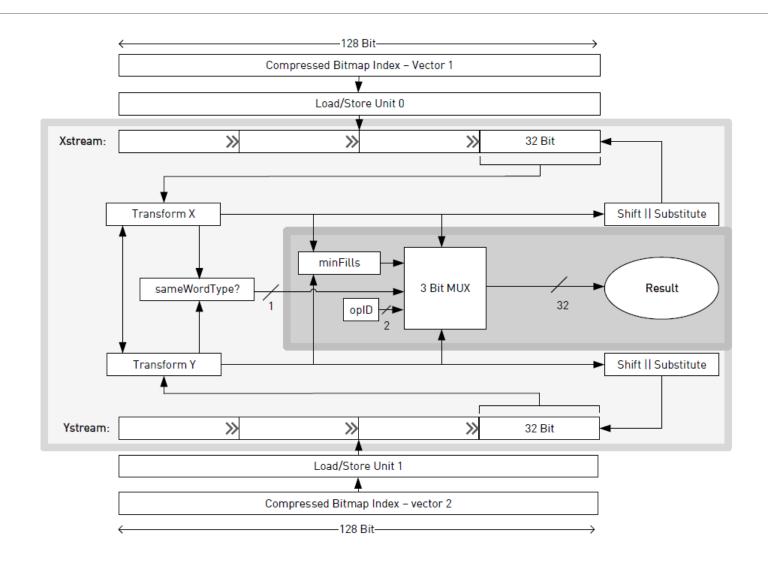
Ausgangsbedingungen

- WAH AND-Verknüpfung von 2 Bitmap Indizes
- 50.000 Bit unkomprimierte Basisdaten, gleichverteilt
- Index 1 variable Bitdichte / Index 2 feste Bitdichte bei 0,01



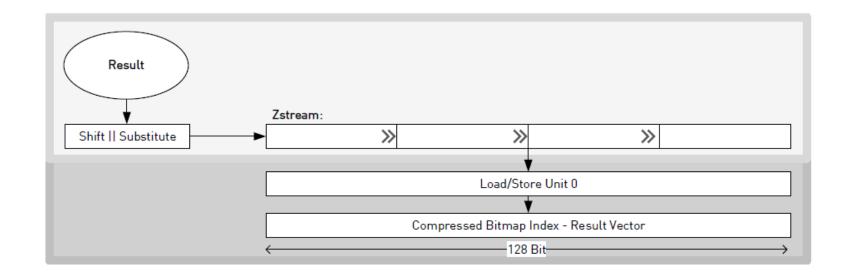


Finale Architektur der WAH Verarbeitungsimplementierung mit Pipelining 1/2





Finale Architektur der WAH Verarbeitungsimplementierung mit Pipelining 2/2





4. Experimentelle Resultate – Hardware

Syntheseergebnisse:

basierend auf Synthese mit 65nm low-power TSMC

Funktionalität	Logikfläche [mm²]	Maximale Frequenz [MHz]
Basisprozessor	0,164	555
WAH	0,419	510
WAH & PLWAH	0,397	455
WAH & PLWAH & COMPAX	0,390	384

Größenanteile am Gesamtprozessor (ca. 223.000 Gates):

Bestandteil	Anteil			
Basisprozessor	20,3 %			
WAH	37,1 %			
PLWAH	21,2 %			
COMPAX	21,4 %			
Summe	100,0 %			



Experimentelle Resultate - Vorbetrachtung

Zentrale Messgröße:

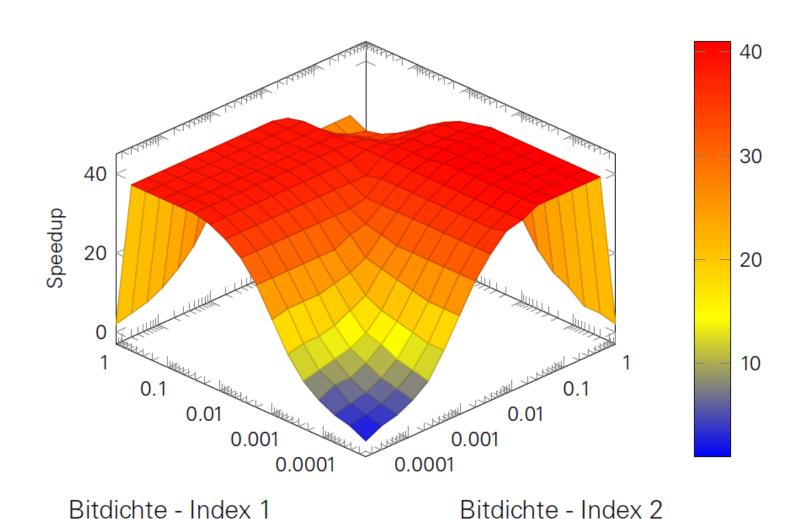
$$Speedup = rac{[Laufzeit C - Code] cycles}{[Laufzeit TIE - Code] cycles}$$

Ausgangsbedingungen:

- 2 Bitmap Indizes mit jeweils variabler Bitdichte
- 50.000 Bit unkomprimierte Basisdaten, gleichverteilt
- Simulationen mit 256 Messpunkten
- Logische AND-Verknüpfung exemplarisch gewählt



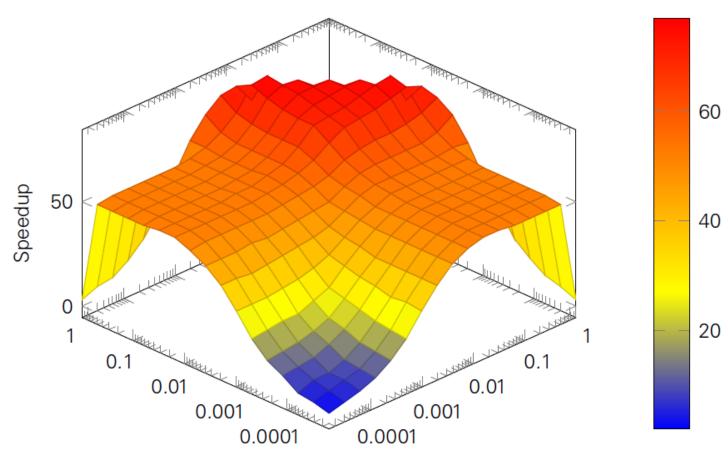
Schnitt von 2 WAH Bitmap Indizes



Verteidigung der Diplomarbeit: TIE Befehlssatzerweiterung für Bitmap Indizes



Schnitt von 2 PLWAH Bitmap Indizes

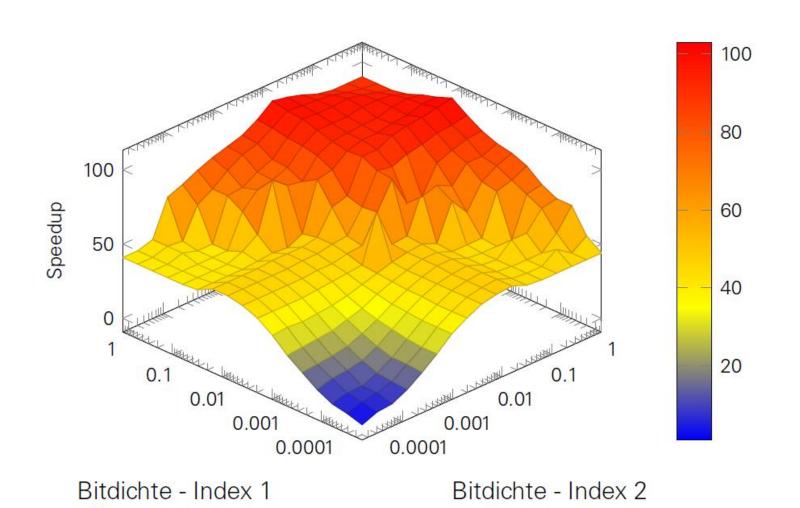


Bitdichte - Index 1

Bitdichte - Index 2



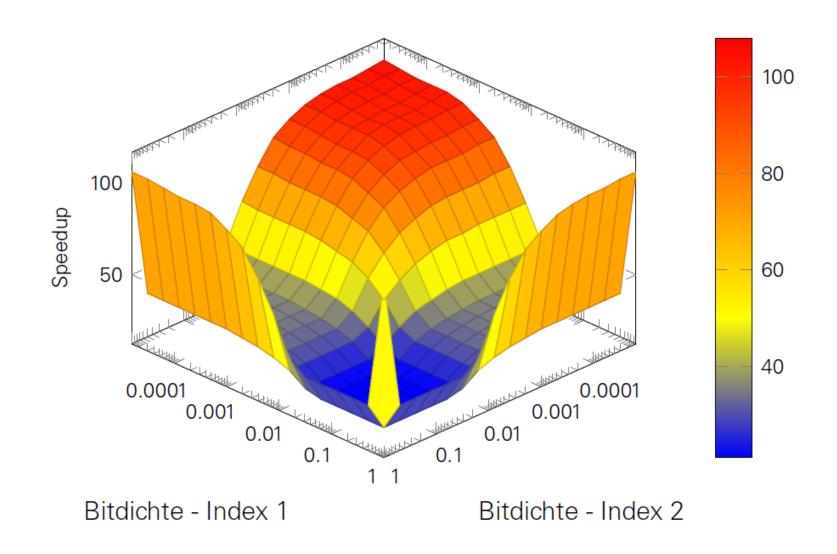
Schnitt von 2 COMPAX Bitmap Indizes



Verteidigung der Diplomarbeit: TIE Befehlssatzerweiterung für Bitmap Indizes

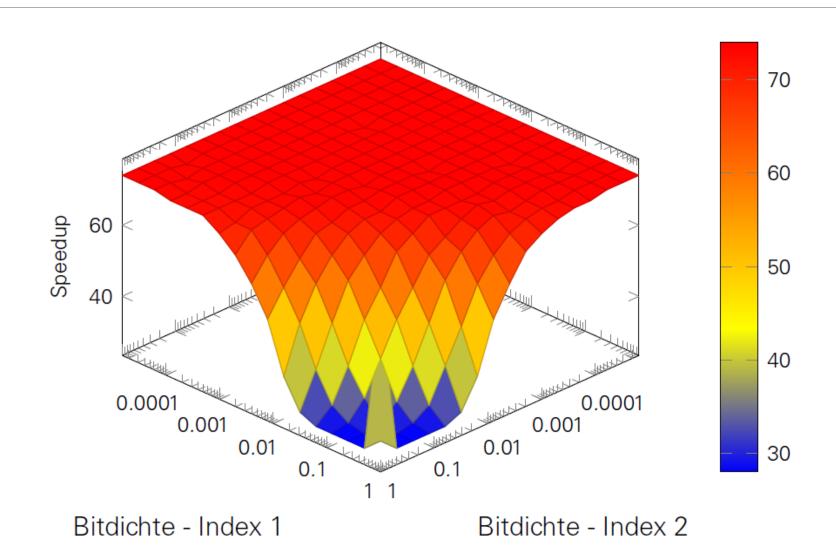


WAH Kompression eines Bitmap Index





Dekompression eines WAH Bitmap Indizes nach Bilden der Schnittmenge



5. Zusammenfassung und Ausblick

Zusammenfassung:

- TIE vor allem f
 ür einfache Operationen interessant (kritischer Pfad)
- Beschleunigung der Verarbeitung komprimierter Indizes erfolgreich
- Insbesondere Kompression und Dekompression interessant
- Synthetisierung der Operationen für alle Algorithmen erfolgreich

Ausblick:

- Beschleunigung weiterer Datenbankoperationen interessant
- Insbesondere einfache Algorithmen mit hohem Datendurchsatz



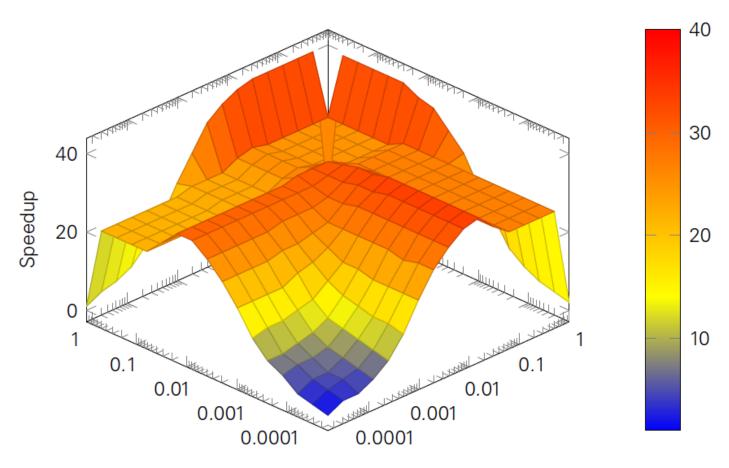








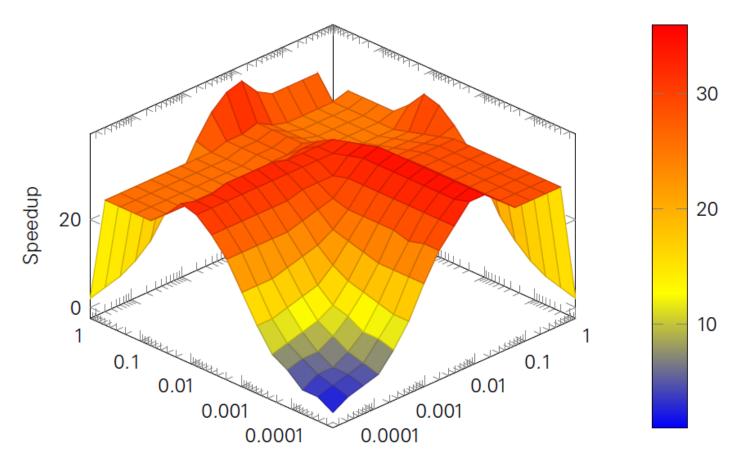




Bitdichte - Index 1

Bitdichte - Index 2

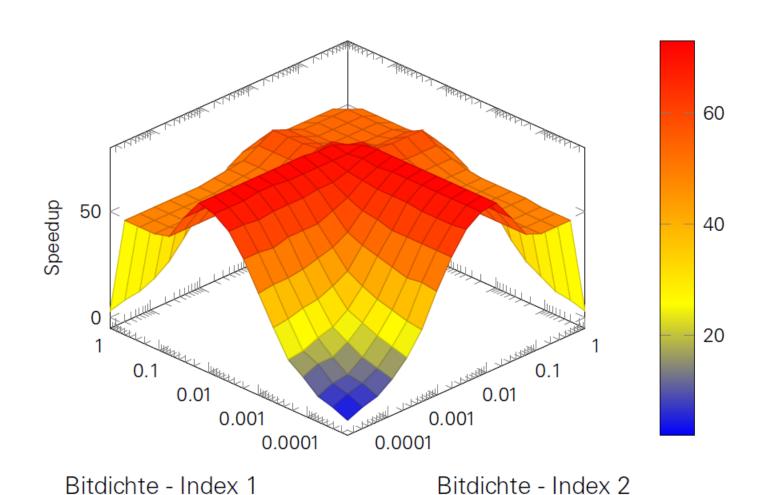




Bitdichte - Index 1

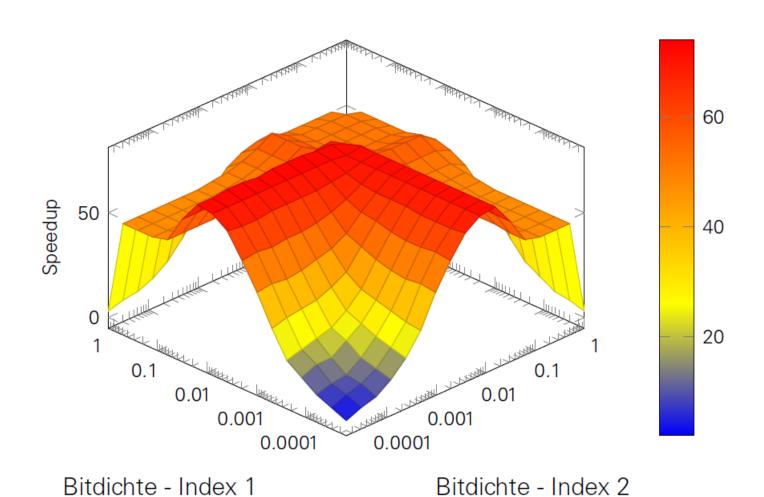
Bitdichte - Index 2





Verteidigung der Diplomarbeit: TIE Befehlssatzerweiterung für Bitmap Indizes

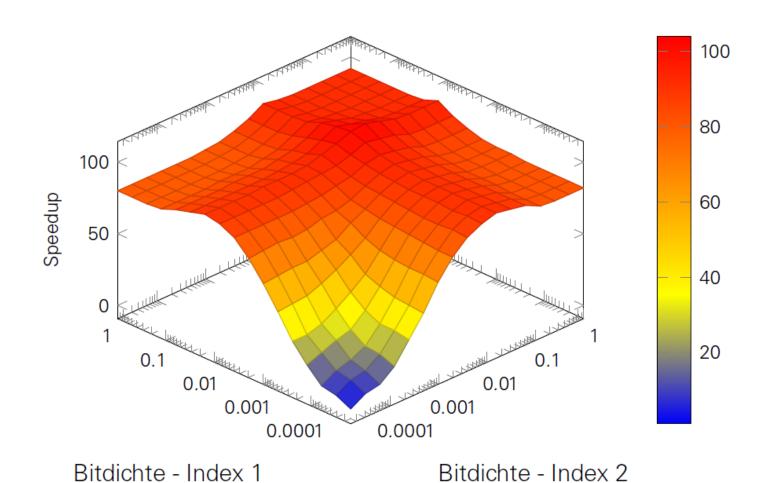




Verteidigung der Diplomarbeit: TIE Befehlssatzerweiterung für Bitmap Indizes

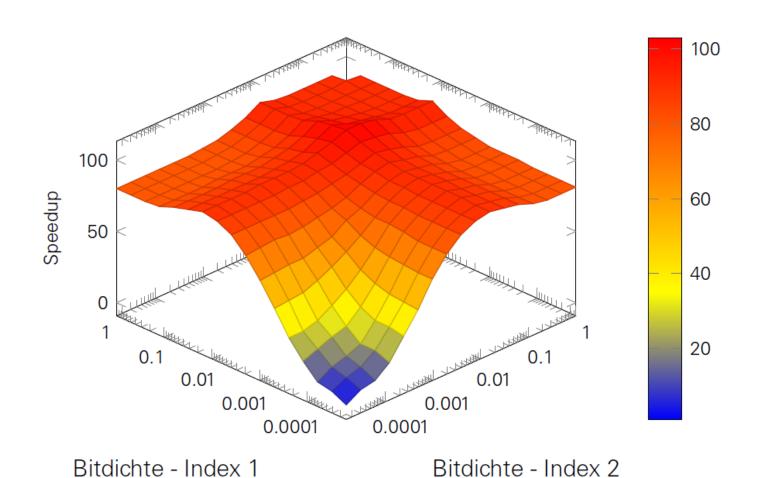






Verteidigung der Diplomarbeit: TIE Befehlssatzerweiterung für Bitmap Indizes

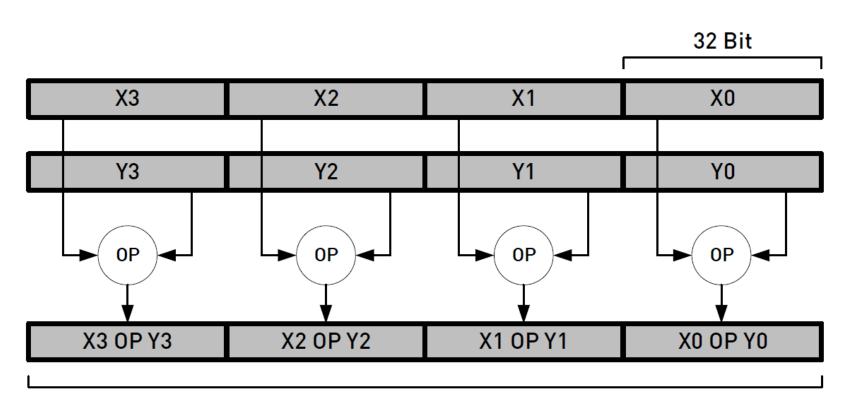




Verteidigung der Diplomarbeit: TIE Befehlssatzerweiterung für Bitmap Indizes







128 Bit



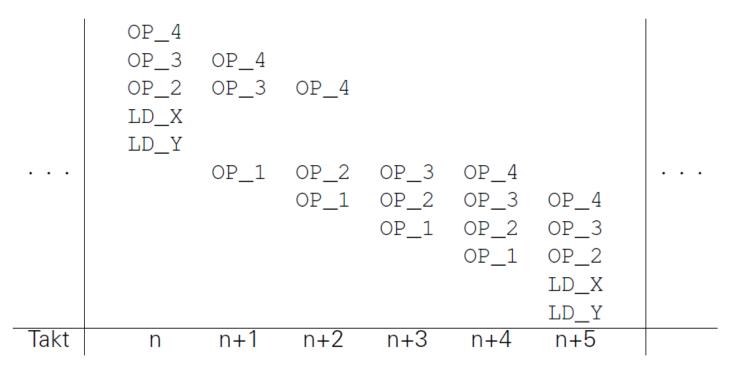
Vergleich der untersuchten Algorithmen

Eigenschaft	BBC	WAH	PLWAH	EWAH	COMPAX
Wortbreite [Bit]	8	32/64	32/64	32	32
Signalwörter	\boxtimes			\boxtimes	
Zustandslose Verarbeitung		\boxtimes	\boxtimes		\boxtimes
Sequentieller Datenzugriff		\boxtimes	\boxtimes		\boxtimes



Pipelining Allgemein und bei WAH

Befehl 1	 IF	ID	EX	MA	WB					
Befehl 2		IF	ID	EX	MA	WB				
Takt	1	2	3	4	5	6	7	8	9	10







Kompressionsraten

