

CSP-Löser und Strategien auf FPGA

Am Beispiel des N-Damenproblems



Übersicht

- **CSP-Struktur und Optimierungsansätze**
- Aufbau eines CSP-Designs
- Vergleich und Auswertung

Constraint Satisfaction Problem

Variablen: x_1, x_2, \dots, x_n

Domänen: D_1, D_2, \dots, D_n

Bedingungen: $p_k(x_{k1}, \dots, x_{kj})$



- Zählen aller gültigen Pfade
- Tiefensuche
- N-Queens, Einfärbungsproblem

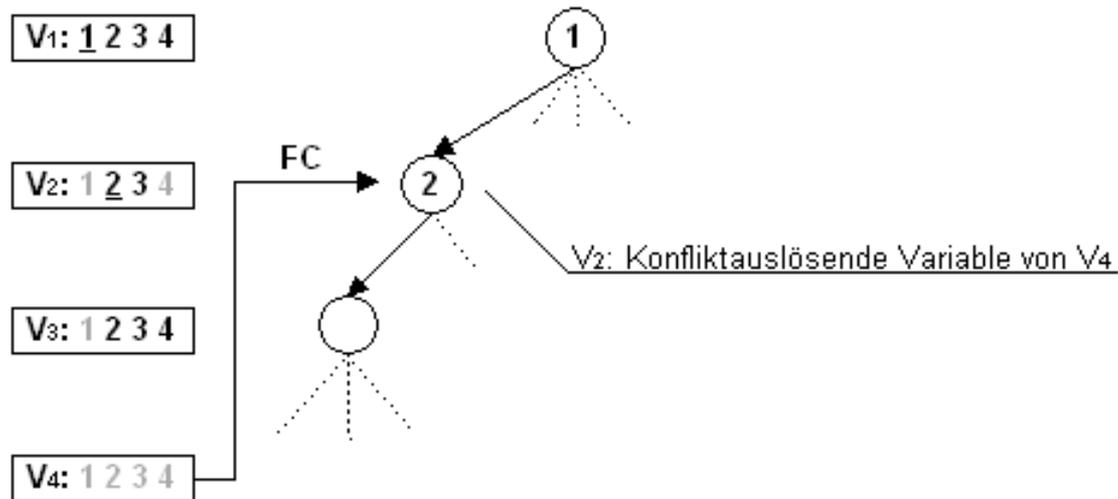
1. Optimierung: im Solver

- Informiertes Backtrack
- Forward Checking

2. Optimierung: Parallelisierung

- domänenbasierte DPS
- funktionsbasierte DPS

Strategien: Forward Checking



Parallelisierungsstrukturen

Domänenbasierte DPS →

Global

B1			
B2			
B3			
B4			

V1 V2 V3 V4

Agent 1

B1			

V1 V2 V3 V4

Agent 2

B2			

V1 V2 V3 V4

Domänenbasierte
DPS →

Global

a1	b1	c1	d1
a2	b2	c2	d2
a3	b3	c3	d3
a4	b4	c4	d4

V1 V2 V3 V4

Agent 1

a1	b1		
	b2		
	b3		
	b4		

V1 V2 V3 V4

Agent 2

a1		c1	
		c2	
		c3	
		c4	

V1 V2 V3 V4

Agent 3

a1			d1
			d2
			d3
			d4

V1 V2 V3 V4

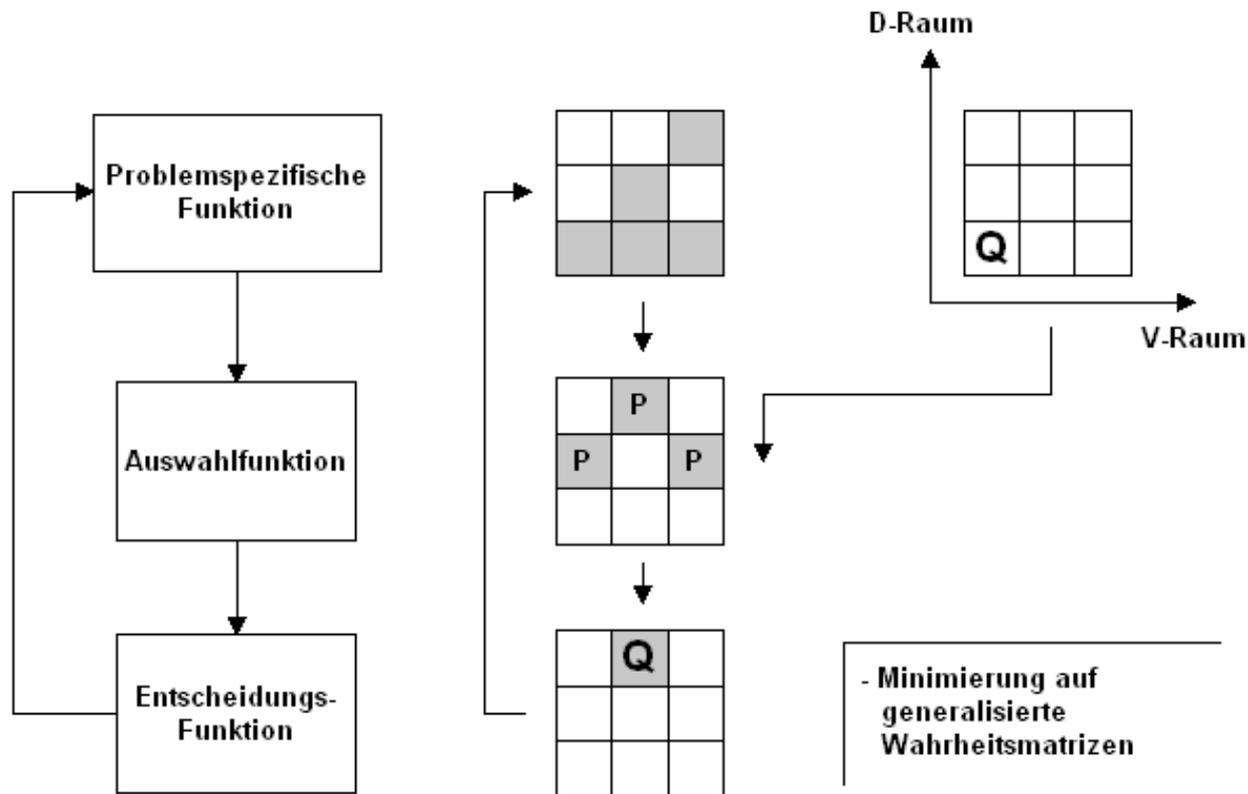
Übersicht

- CSP-Struktur und Optimierungsansätze
- **Aufbau eines CSP-Designs**
- Vergleich und Auswertung

Systementwurf

- Globale Parallelisierung (domänenbasiertes DPS)
 - Netzwerk eingebettetes System mit DB
- Lokalität von Information und Informationsverarbeitung
 - Daten- und Verarbeitungseinheiten auf einem Chip
- Parallelisierung des Lösungsentwurfs
 - Parallele Strukturen auch innerhalb eines Lösers – Kommunikation / Synchronisierung
- Logische Strukturen auf Basis von Kommunikation
 - Kommunikation zwischen den Prozessen ersetzt komplexe Operationen

Solver-Aufbau



Problemspezifische Funktion

- Input Datenbank: Zugewiesene Variablen
 - Initialisierung: Leere Menge, Vordefinierte Menge
- Output Datenbank: Abb. inkonsistenter Domänenwerte



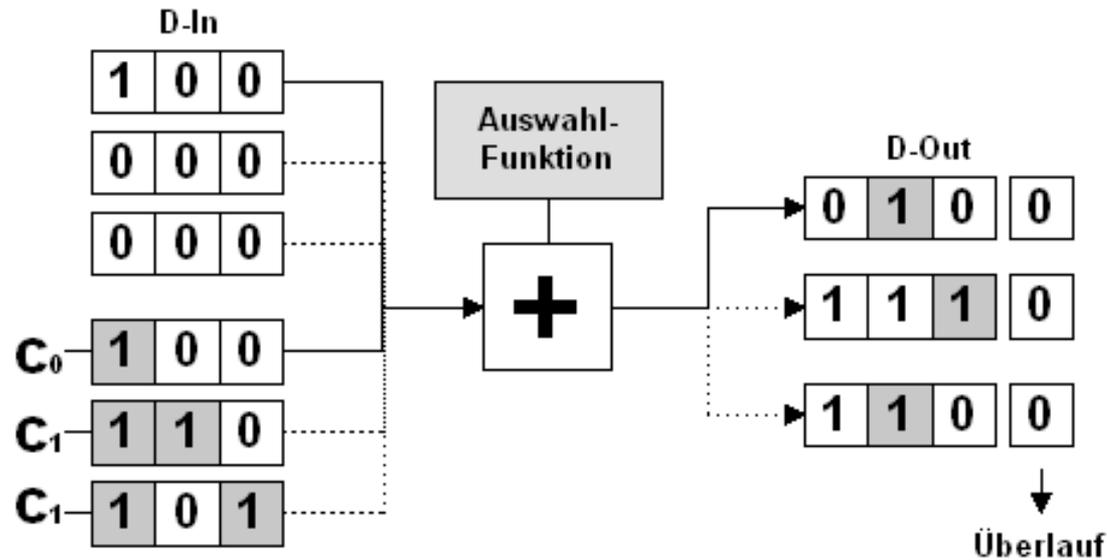
Problemspezifische Funktion

- Einzige Funktion, der Constraints bekannt sind und
- die problemspezifisch optimiert werden muss

- Informationen werden auf generalisierte Wahrheitsmatrizen abgebildet

- Nur verändertes Wissen muss in der Aktualisierung berücksichtigt werden

Auswahlfunktion

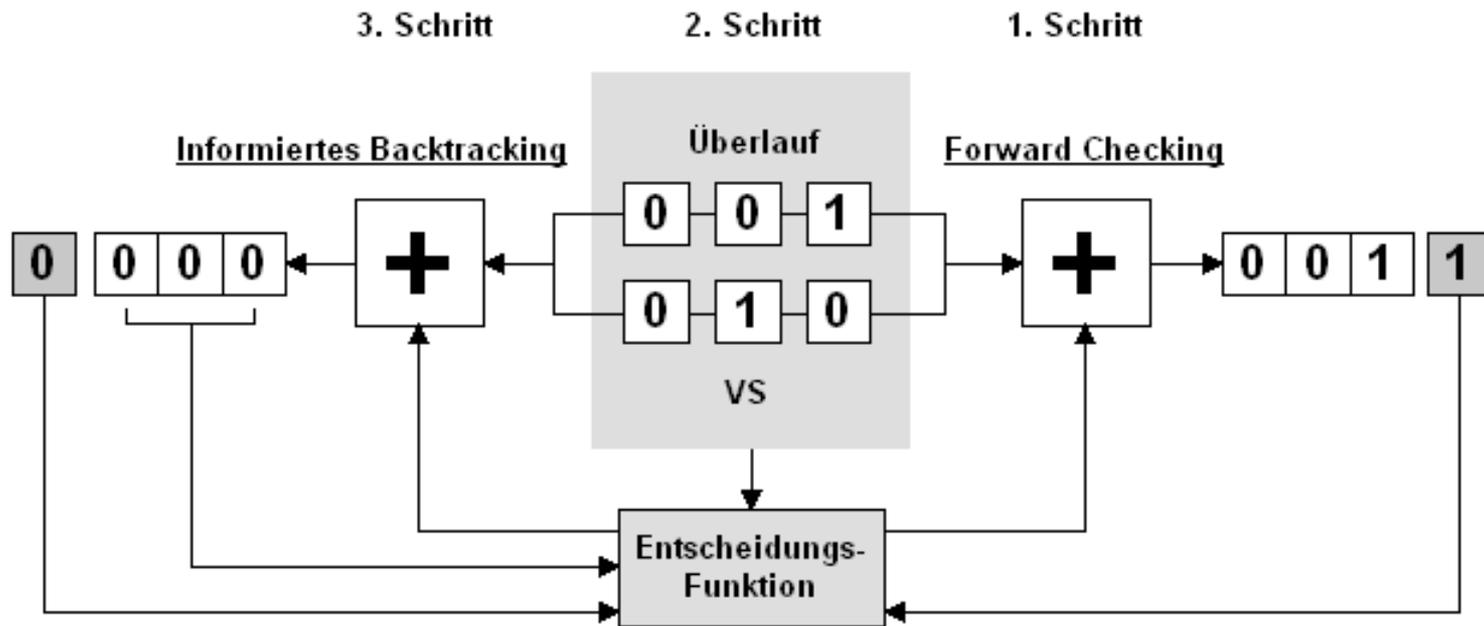


- Aufbau einer Carry-Chain
- VL Computer-Arithmetik

Auswahlfunktion

- Domänen werden parallel ausgewertet
- Sehr schnelle Kommunikationsweiterleitung
 - jedoch u.U. kritischer Pfad
 - Fertige Architektur auf FPGA
- Out1: Übertrag
- Out2: Der nächst wählbare Wert jeder Domäne

Entscheidungsfunktion



Entscheidungsfunktion

- Entscheidet die Wertzuweisung der aktiven Variable und
- die Auswahl der nächst aktiven Variable
- Teilparallelisierte Operation
- Sehr schneller lokaler Informationsaustausch
- Es wird immer eine „positive“ Entscheidung getroffen

Übersicht

- CSP-Struktur und Optimierungsansätze
- Aufbau eines CSP-Designs
- **Auswertung und Vergleich**

Beispiel: Online Computer Store

- Schnelle Zugriffszeiten
- Eingabe: Produkt oder Leistungsmerkmale
- Anzeige von ausschließlich relevanten Produkten
- Automatische Auswahl, wie preiswerteste Konfiguration

Vergleich

Auf Basis von N-Queens

P ¹	PC-Time ²	PC-Time ³	SE ⁴	FPGA-Time ⁵	SE-Factor ⁶
1	410.97s	102.74s	99.00s	0.55s	4.15
2	675.92s	168.98s	154.80s	0.86s	4.37
3	778.41s	194.60s	181.80s	1.01s	4.28
4	404.78s	101.20s	91.80s	0.51s	4.41

¹partial problems

²Intel(R) Core(TM) 2 Quad CPU Q9550 at 2.83GHz, single core use

³Intel(R) Core(TM) 2 Quad CPU Q9550 at 2.83GHz, quad core use

⁴Slice Equivalent: single core mode at 100MHz

⁵Multi core mode: Virtex-4 XC4VLX160 at 143MHz

⁶Compared with the single core mode of the Q9950

- *Optimierter Löseransatz für N-Queens verwendet
- 65 Teilprobleme pro Sekunde zu berechnen

Abschluss



- Noch Fragen?