



KV Computer Microsystems Teil III

Bindung und FSMD-Synthese

Andreas Koch

FG Eingebettete Systeme und ihre Anwendungen
Informatik, TU Darmstadt

Sommersemester 2005

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



- 1 Organisatorisches
- 2 Bindung
 - Modell
 - ILP-basierte Lösung
 - LEFTEDGE Algorithmus
- 3 Registerbindung
- 4 Verbindungssynthese
- 5 Steuerwerksynthese
 - Festverdrahtet
 - Microcode
 - Kompaktierung
 - Hierarchie
- 6 Zusammenfassung

Organisatorisches

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssynthese

Steuerwerksynthese

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassung



- Gesonderte Anmeldung bis zum 6.7.2005 erforderlich
 - **Unabhängig** von Prüfungssekretariat
 - **Auch** für Diplomis, Nebenfächler, etc.
 - http://www.vlsi.informatik.tu-darmstadt.de/student_area/klausur/?klausurid=6
 - Auch von den Vorlesungsseiten verlinkt
- Termin: 13.07.2005, 9:00-11:30 Uhr
 - Echte Klausurdauer: 120 Minuten
- Erlaubte Hilfsmittel: **Keine!**
 - Aufgabenstellung kann aber Extrahinweise enthalten
 - Alles Papier wird gestellt
- Raumaufteilung: Listen hängen ca. 3 Tage vorher aus

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



- Ordnet Operationen konkrete Instanzen des Ressourcetyps zu
- Kann durchgeführt werden
 - Vor
 - Während
 - Nach (\leftarrow hier betrachtet)
- ... Ablaufplanung
- Gemeinsame Ressourcennutzung bei nicht ressource-beschränkten Ablaufplänen
 - Im anderen Fall: Ressourceanzahlen bereits während Ablaufplanung bestimmt
- Ergebnis ist Grundlage für Verbindungssynthese

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



- Ein Ressourcetyp kann unterschiedliche Operationen abdecken
- Wir betrachten ressourcedominierte Schaltungen
- Eine Operation verbleibt während ihrer gesamten Ausführungszeit auf derselben Instanz

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



- Sequenzgraph $G_S(V, E)$
- v_0 und v_n spielen keine Rolle mehr (NoOps)
- Ressourcenzuordnung $T : V \rightarrow R$, $R = \{r_k : 1 \leq k \leq n_{res}\}$
- Gesucht **Bindung** $\beta : V \rightarrow R \times \mathbf{N}$
 - Zuordnung einer Operation an eine Instanz einer Ressource

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu

Kompatible Operationen



Zwei Operationen können die selbe Ressourceinstanz nutzen, wenn sie ...

- ... den gleichen Typ haben und
- ... nicht zur gleichen Zeit ablaufen

Dann heissen sie **kompatibel**

Kompatibilität

Zwei ablaufgeplante Operationen v_i und v_j sind *kompatibel* wenn gilt:

$$T(v_i) = T(v_j) \wedge ((t_i + d_i \leq t_j) \vee (t_j + d_j \leq t_i))$$

Konflikt

Zwei ablaufgeplante Operationen stehen in *Konflikt* zueinander, wenn sie nicht kompatibel sind.

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



Kompatibilitätsgraph G_+

Ungerichteter Graph $G_+(V, E)$ mit den Operationen als Knoten $V = \{1, \dots, n_{ops}\}$ und Kanten

$$E = \{\{v_i, v_j\} : v_i \text{ ist kompatibel zu } v_j\}$$

Konfliktgraph G_-

Ungerichteter Graph $G_-(V, E, k)$ mit den Operationen als Knoten $V = \{v_i : T(v_i) = k \wedge 1 \leq i \leq n_{ops}\}$ und Kanten

$$E = \{\{v_i, v_j\} : v_i \text{ steht in Konflikt zu } v_j\}$$

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

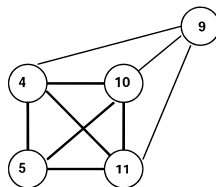
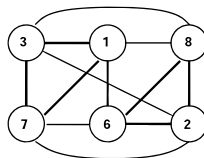
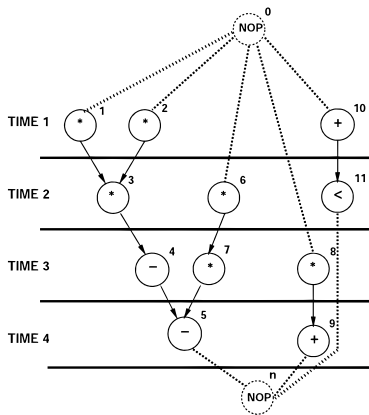
Microcode

Kompaktierung

Hierarchie

Zusammenfassu

Beispiel: Kompatibilitätsgraph



Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

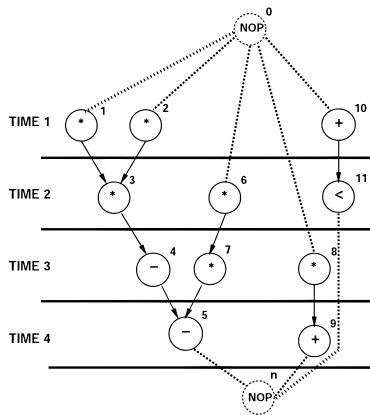
Microcode

Kompaktierung

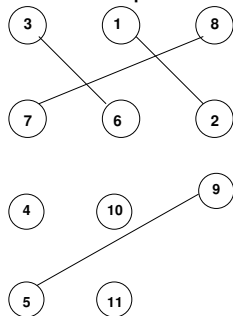
Hierarchie

Zusammenfassu

Beispiel: Konfliktgraphen je Ressource



k=1: Multiplizierer



k=2: ALUs

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu

Eigenschaften von G_+

- Kompatibilitätsgraph hat mindestens n_{res} verschiedene disjunkte Teilgraphen
- Untereinander kompatible Operationen bilden Clique
- Ziel von Bindung: Minimiere die Zahl von Ressourcen
- Kompatible Ressourcen können sich Instanz teilen
- Also: Suche nach möglichst wenigen, dafür aber möglichst grossen disjunkten Cliques
- Grösste Menge untereinander kompatibler Ressourcen ist **maximale Clique**
- Minimalzahl benötigter Cliques um ganz V zu partitionieren ist $\kappa(G_+(V, E))$



Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

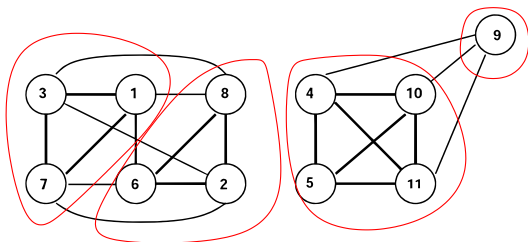
Microcode

Kompaktierung

Hierarchie

Zusammenfass

Beispiel: Maximale Cliques



$$\kappa(G_+(V, E)) = 4$$

$\{v_1, v_3, v_7\}$, $\{v_2, v_6, v_8\}$, $\{v_4, v_5, v_{10}, v_{11}\}$, $\{v_9\}$

Ergebnis: 2 Multiplizierer und 2 ALUs

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



- Kompatible Operatoren sind *nicht* durch Kanten verbunden
 - Solche Operatoren bilden *unabhängige* Mengen
- Durch Kanten verbundene Operatoren müssen auf unterschiedlichen Instanzen realisiert werden
- Idee: Löse das Graphfärbungsproblem, Farben $c(v)$ entsprechen Instanzen
 - $\forall \{v_i, v_j\} \in E : c(v_1) \neq c(v_2)$
- Gesucht: Färbung mit minimaler Anzahl $\chi(G_-(V, E, k))$ von Farben

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

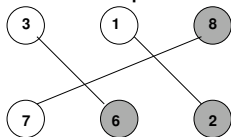
Hierarchie

Zusammenfassu

Beispiel: Minimale Graphfärbung

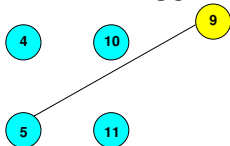


k=1: Multiplizierer



$$\chi(G_-(V, E, 1)) = 2, \\ \{v_1, v_3, v_7\}, \{v_2, v_6, v_8\},$$

k=2: ALUs



$$\chi(G_-(V, E, 2)) = 2 \\ \{v_4, v_5, v_{10}, v_{11}\}, \{v_9\}$$

Ergebnis: 2 Multiplizierer und 2 ALUs

$$\sum_{k=1}^{n_{res}} \chi(G_-(V, E, k)) = \kappa(G_+(V, E))$$

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



Lösung beider Probleme

- Partitionierung durch maximale Cliques
- Einfärbung mit minimaler Farbenzahl

in \mathcal{NP} für den allgemeinen Fall

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



- a_k Anzahl von Instanzen des Ressourcetypes k
- $B = \{b_{i,r} : 1 \leq i \leq n_{ops}, 1 \leq r \leq a_k\}$
- $b_{ir} = 1$, wenn Operation v_i auf Instanz r des Ressourcetypes k ausgeführt wird, sonst 0
- Bedeutet Bindung: $\beta(v_i) = (k, r)$
- $X = \{x_{i,l} : 1 \leq i \leq n_{ops}, 1 \leq l \leq \lambda\}$
- $x_{i,l} = 1$, wenn Operation v_i in Schritt l gestartet wird, 0 sonst

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



Für jeden Ressourcetype k muss gelten:

- 1 Jede Operation v_i muss auf genau einer Instanz ausgeführt werden

$$\sum_{r=1}^{a_k} b_{i,r} = 1 \quad , \forall 1 \leq i \leq n_{ops}$$

- 2 Auf jeder Instanz r kann im Zeitschritt l nur eine Operation v_i ablaufen

$$\sum_{\{v_i: T(v_i)=k\}} b_{i,r} \sum_{m=l-d_i+1}^l x_{i,m} \leq 1 \quad , \forall 1 \leq l \leq \lambda, 1 \leq r \leq a_k$$

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu

Beispiel: ILP Bindung



Annahmen: Ressourcetypen 1 (=Mult) und 2 (=ALU), $d_i = 1$
Aus Schedule: Instanzanzahlen $a_1 = a_2 = 2$, $\lambda = 4$

Hier gerechnet für Multiplizierer $\{v_1, v_2, v_3, v_6, v_7, v_8\}$

Jede Operation auf genau einer Instanz

$$b_{1,1} + b_{1,2} = 1$$

$$b_{2,1} + b_{2,2} = 1$$

$$b_{3,1} + b_{3,2} = 1$$

$$b_{6,1} + b_{6,2} = 1$$

$$b_{7,1} + b_{7,2} = 1$$

$$b_{8,1} + b_{8,2} = 1$$

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

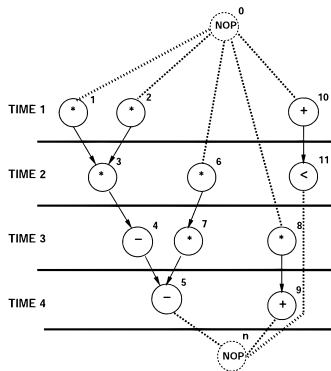
Hierarchie

Zusammenfassu

Beispiel: ILP Bindung



Vorgegebener Ablaufplan



x_{il} durch Scheduling bereits festgelegt, Werte einsetzen

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



Auf jeder Instanz nur eine Operation pro Zeitschritt

$$b_{1,1} + b_{2,1} \leq 1 \quad : l = 1 \qquad b_{1,2} + b_{2,2} \leq 1 \quad : l = 1$$

$$b_{3,1} + b_{6,1} \leq 1 \quad : l = 2 \qquad b_{3,2} + b_{6,2} \leq 1 \quad : l = 2$$

$$b_{7,1} + b_{8,1} \leq 1 \quad : l = 3 \qquad b_{7,2} + b_{8,2} \leq 1 \quad : l = 3$$

Keine Multiplikation mehr in Zeitschritt $l = 4$



Organisatorische

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu

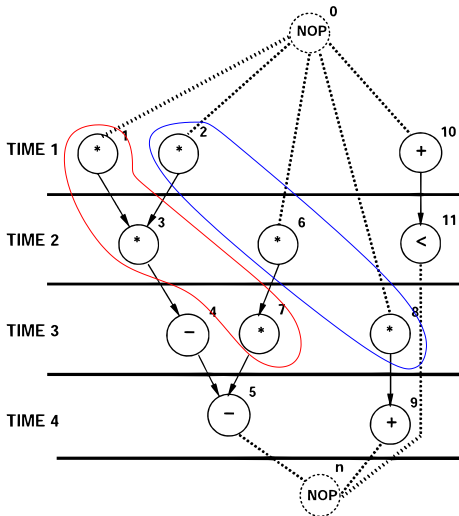
Demo mit Ip_solve

Beispiel: ILP Bindung



Ergebnis aus ILP-Solver

b11	0
b12	1
b21	1
b22	0
b31	0
b32	1
b61	1
b62	0
b71	0
b72	1
b81	1
b82	0



Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



- Lösung von ILP ist in \mathcal{NP}
- Nicht überraschend, andere allgemeine Lösungen sind auch in \mathcal{NP}
- Aber:
 - Im Fall von **Intervallgraphen** lässt sich das Problem in polynomieller Zeit exakt lösen!

Intervallgraph

Ein ungerichteter Graph $G(V, E)$ heisst Intervallgraph genau dann, falls jedem Knoten $v_i \in V$ ein Intervall $[l_i, r_i)$, mit $l_i, r_i \in \mathbf{Z} \wedge l_i \leq r_i$, zugeordnet werden kann und eine Kante $\{v_i, v_j\} \in E$ genau dann existiert, wenn sich die Intervalle $[l_i, r_i)$ und $[l_j, r_j)$ überlappen.

Konfliktgraphen G_- sind Intervallgraphen.

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



- Eingabe: Liste I von Intervallen
- Ausgabe: Überlappungsfreie Farbzuoordnung der Intervalle
- Idee
 - 1 Sortiere Intervalle nach aufsteigender Untergrenze
 - 2 Gehe Liste durch und weise nicht-überlappenden Intervallen die gleiche Farbe zu
 - 3 Nimm die nächste Farbe und wiederhole für ungefärbte Intervalle

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE
Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



Optimales überlappungsfreies Einfärben von Intervallen I

LEFTEDGE(I)

```
1  Sortiere Elemente von  $I$  aufsteigend nach  $l_i$  in Liste  $L$ ;  
2   $j = 0$ ;  
3  while  $L \neq \emptyset$  do  
4       $S = \emptyset$ ;  
5       $r = 0$ ; /* rechter Rand von Elementen in  $S$  */  
6      while  $\exists i \in L : l_i \geq r$  do  
7           $i =$  Element aus  $L$  mit kleinsten  $l_i \geq r$ ;  
8           $S = S \cup \{i\}$ ;  
9           $r = r_i$ ;  
10          $L = L \setminus \{i\}$ ;  
11  
12          $j = j + 1$ ;  
13         for  $i \in S$  do  
14              $c(i) = j$ ; /* Farbe des Intervalls setzen */
```

Organisatorische

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE
Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

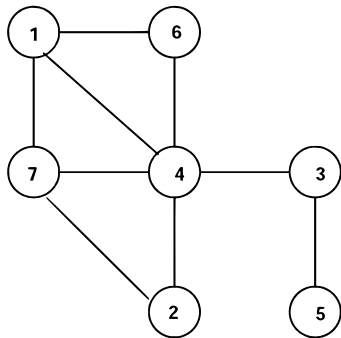
Hierarchie

Zusammenfassu

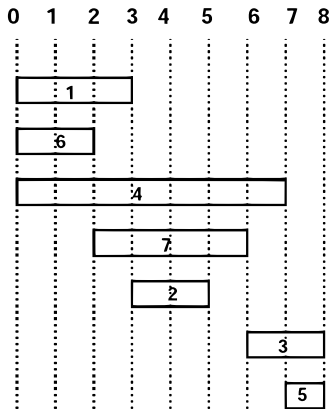
Beispiel: LEFTEDGE Algorithmus



Eingabe



(a)



(b)

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE
Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

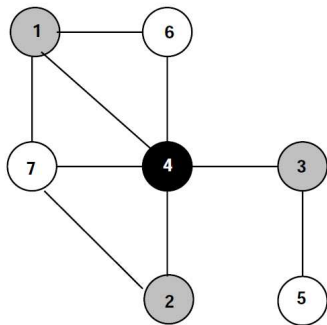
Hierarchie

Zusammenfassu

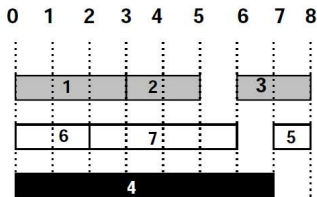
Beispiel: LEFTEDGE Algorithmus



Ausgabe



(c)



(d)

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE
Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



- Jede Kante zwischen zwei Operatoren braucht Variablen für Datentransfer
- Lebensdauer einer Variablen
 - Geburt** Zeitpunkt an dem Wert an Operatorausgang anliegt
 - Tod** Letzter Zeitpunkt an dem der Wert an einem Operatoreingang benötigt wird
- Variablen müssen für ihre Lebensdauer gespeichert werden.
- Unterschiedliche technische Realisierungsmöglichkeiten
 - Speicher: Ein- oder Multi-Port
 - Register: Alle parallel zugreifbar ← hier betrachtet

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



- Register speichern ...
 - Eingabewerte
 - Zwischenergebnisse
 - Ausgabewerte
- Vereinfachung hier
 - Dedizierte Register für Ein- und Ausgabewerte
- Aber optimierbar: Register für Zwischenergebnisse

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



- Gegeben: Ein Ablaufplan
- Daraus bestimmbar: Lebenszeiten
- Überlappende Lebenszeitintervalle implizieren separate Register
- Lösung mit Konfliktgraph und Einfärbung
 - Knoten: Variablen
 - Kanten: Überlappende Lebenszeiten
 - Gesucht: Minimale Anzahl von Registern für Zwischenergebnisse
 - Lösen mit LEFTEDGE-Algorithmus

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

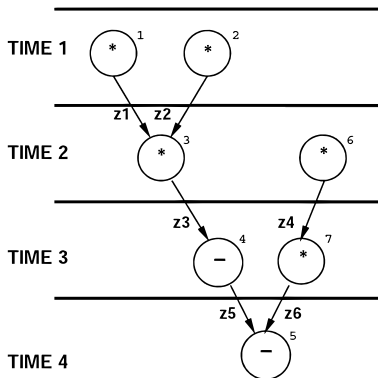
Microcode

Kompaktierung

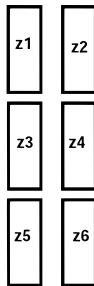
Hierarchie

Zusammenfassu

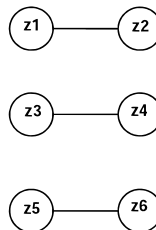
Beispiel: Registerbindung



(a)



(b)



(c)

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



Erstellen von Verbindungen

- zwischen Ressourceinstanzen und Registern
- zwischen Registern
- zur Schnittstelle zum Steuerwerk
- zu den Ein-/Ausgabe-Ports zum Restsystem

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

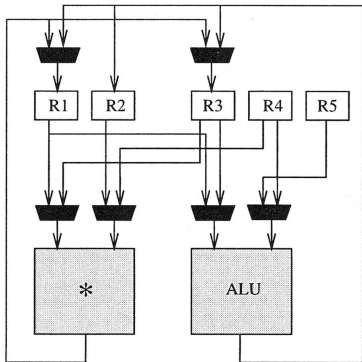
Hierarchie

Zusammenfassu

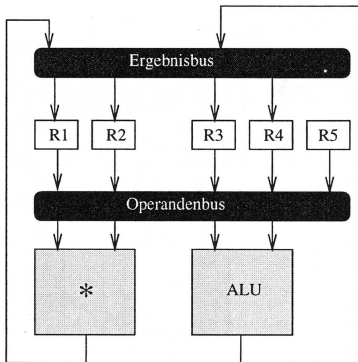
Verbindungsarchitekturen



a) multiplexerbasierte Architektur



b) busorientierte Architektur



Mehr Parallelität ./.. einfacherer Aufbau

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

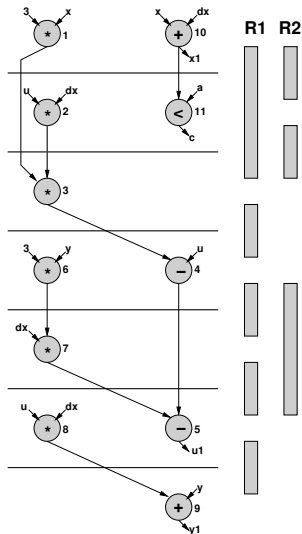
Microcode

Kompaktierung

Hierarchie

Zusammenfassu

Beispiel: Verbindungssynthese



Ablaufplan mit 1
Multiplizierer, 1 ALU

Diesmal: Mit
Ein-/Ausgabewerten und
Registerlebenszeiten

Organisatorische

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

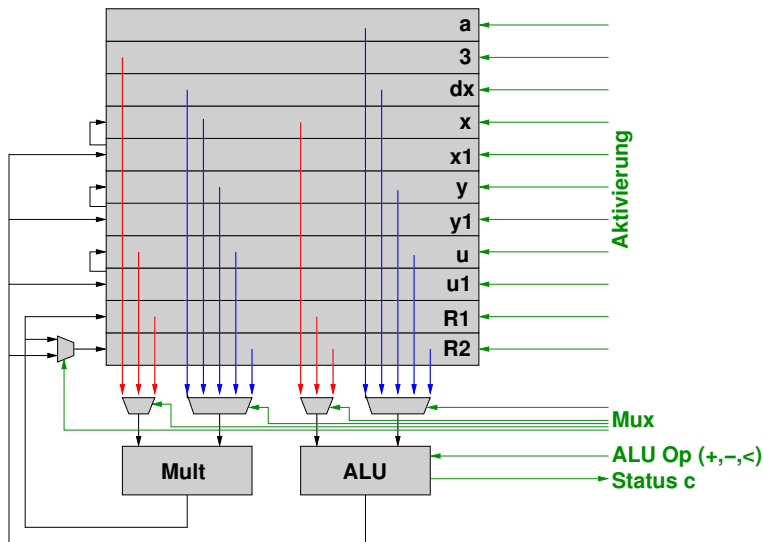
Microcode

Kompaktierung

Hierarchie

Zusammenfassu

Beispiel: Verbindungssynthese



Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



- Verhaltenssicht: Synchrone FSM
- Zunächst: Vereinfachte Sicht
 - Flache Sequenzgraphen
 - Datenunabhängige Verzögerungen
- Auf unterschiedliche Arten realisierbar
 - Festverdrahtete FSM
 - Microcode (ROM, PLA)
 - Verteilte FSM

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

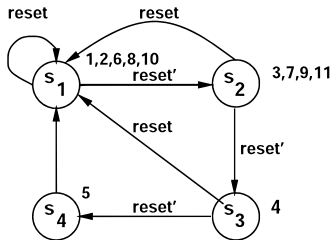
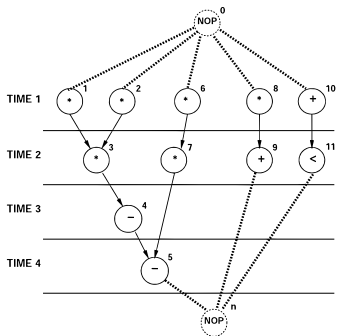
Hierarchie

Zusammenfassu

Beispiel: Festverdrahtete FSM



Annahme: Dedizierte Ressourcen mit Verzögerung 1.



Direkt in Flip-Flops und Gatter umsetzbar.

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



- Kleine Programme, gespeichert in ROMs oder PLAs
- Horizontaler Microcode
 - Ein Bit pro Aktivierungs/Steuersignal (z.B. Muxe)
 - Ein Microcode-Wort pro Zeitschritt
 - Maximale Parallelität → niedrige Latenz
 - Breite Worte (mehr Verdrahtung)
- Vertikaler Microcode
 - Ein kodierte Microcode-Wort pro Ressource
 - Schmalere Worte (weniger Verdrahtung)
 - Auch mehrere Worte pro Zeitschritt erforderlich
 - Weniger Parallelität

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

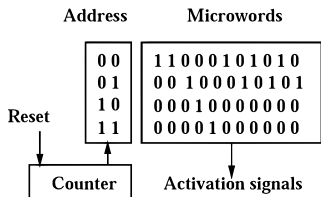
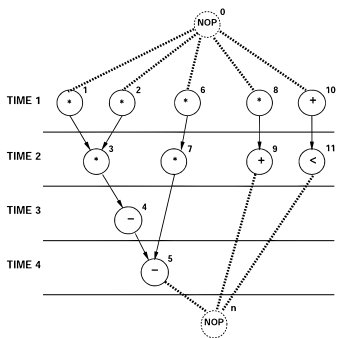
Microcode

Kompaktierung

Hierarchie

Zusammenfassu

Beispiel: Horizontaler Microcode



Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssystem

Steuerwerksynthese

Festverdrahtet

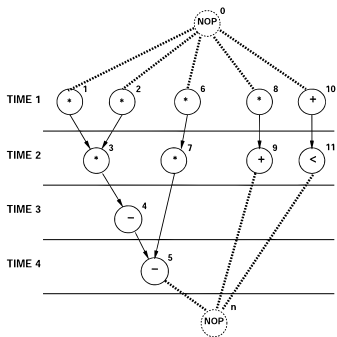
Microcode

Kompaktierung

Hierarchie

Zusammenfassung

Beispiel: Vertikaler Microcode



Microwords

```
0001
0010
0110
1000
1010
0011
0111
1001
1011
0100
0101
```

Decoder

Activation signals

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssystem

Steuerwerksynthese

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassung



- Beobachtung: Horizontaler Microcode enthält viele Nullen
- Breite Microworte lohnen sich nur bei echter Parallelität
- Finde Teile mit paralleler Ausführung
 - Steuere diese parallel an (horizontaler Ansatz)
- Für sequentielle Teile
 - Benutze kompaktere kodierte Darstellung (vertikaler Ansatz)

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



Ein Ansatz:

- Baue Konfliktgraph von parallelen Steuersignalen auf
 - Kante existiert bei paralleler Ausführung
- Dann minimales Einfärbeprobem lösen
 - **Kein Intervallgraph mehr**
 - Heuristik verwenden (Verfahren hier nicht behandelt)
- Jede Farbe entspricht einer Gruppe von Steuersignalen
- Innerhalb der Gruppe: Sequentielle Ausführung
- Gruppen untereinander: Parallele Ausführung

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

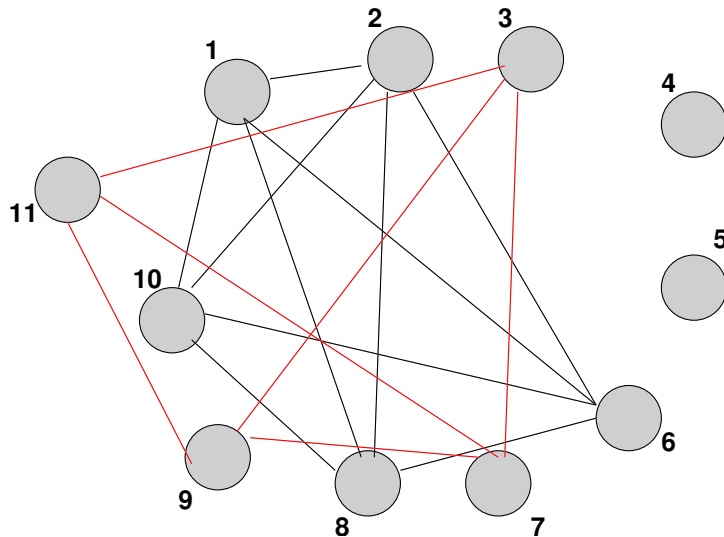
Hierarchie

Zusammenfassu

Beispiel: Microcode-Kompaktierung



Konfliktgraph



Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

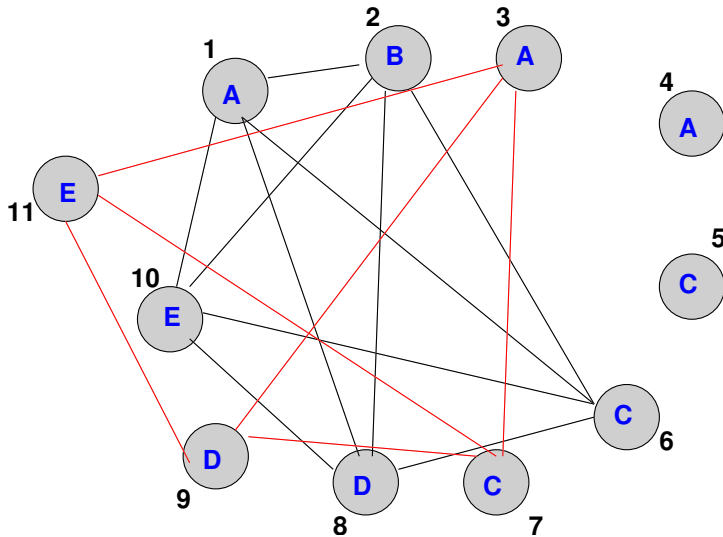
Hierarchie

Zusammenfassu

Beispiel: Microcode-Kompaktierung



Minimal eingefärbter Konfliktgraph



Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu

Beispiel: Microcode-Kompaktierung



field	op	code
A	1	01
A	3	10
A	4	11
B	2	1
C	6	01
C	7	10
C	5	11
D	8	01
D	9	10
E	10	01
E	11	10

Beachte: Braucht Code für NoOp, hier 00.

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

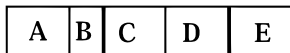
Zusammenfassu

Beispiel: Microcode-Kompaktierung

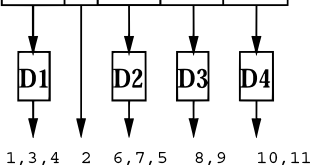
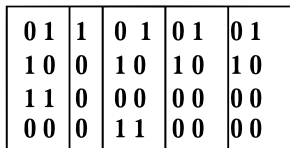


Damit jetzt schmalere Microworte **ohne** Parallelitätsverlust

Microword format



Microwords



Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

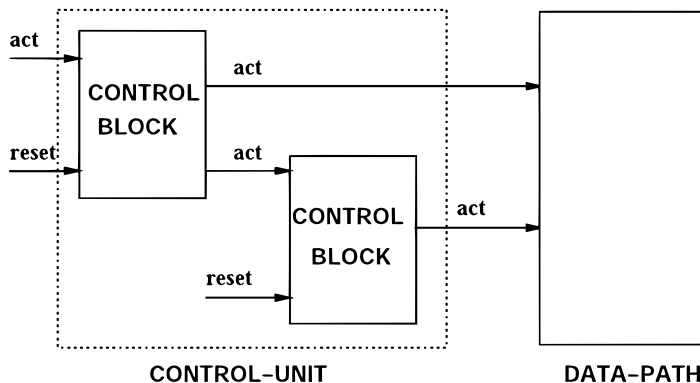
Hierarchie

Zusammenfassu

Hierarchische Steuerwerke



Für hierarchische Sequenzgraphen



Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

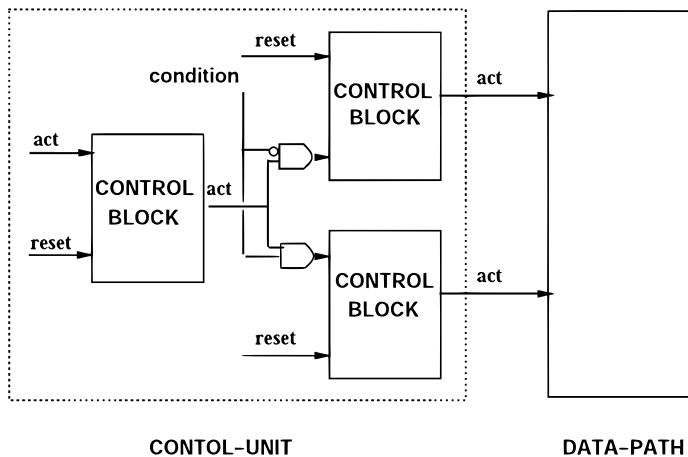
Hierarchie

Zusammenfassu

Hierarchische Steuerwerke



Mit Verzweigung



Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu



- Grundlagen der Architektursynthese
- Modelle
- Ablaufplanung
- Bindung
- Konnektivitätssynthese
- Steuerwerksynthese

Noch nicht

- Logiksynthese
- Sequentielle Optimierung
- Bibliotheksabbildung
- **Layoutsynthese** ← im Wintersemester :-)

Organisatorisch

Bindung

Modell

ILP-basierte Lösung

LEFTEDGE

Algorithmus

Registerbindung

Verbindungssyn

Steuerwerksynt

Festverdrahtet

Microcode

Kompaktierung

Hierarchie

Zusammenfassu