

Hardwarebeschreibung

Digital-Design

Prof. Dr.-Ing. habil. Jürgen Kampe

High-level Synthese

1. Datenfluss und Steuerfluss.

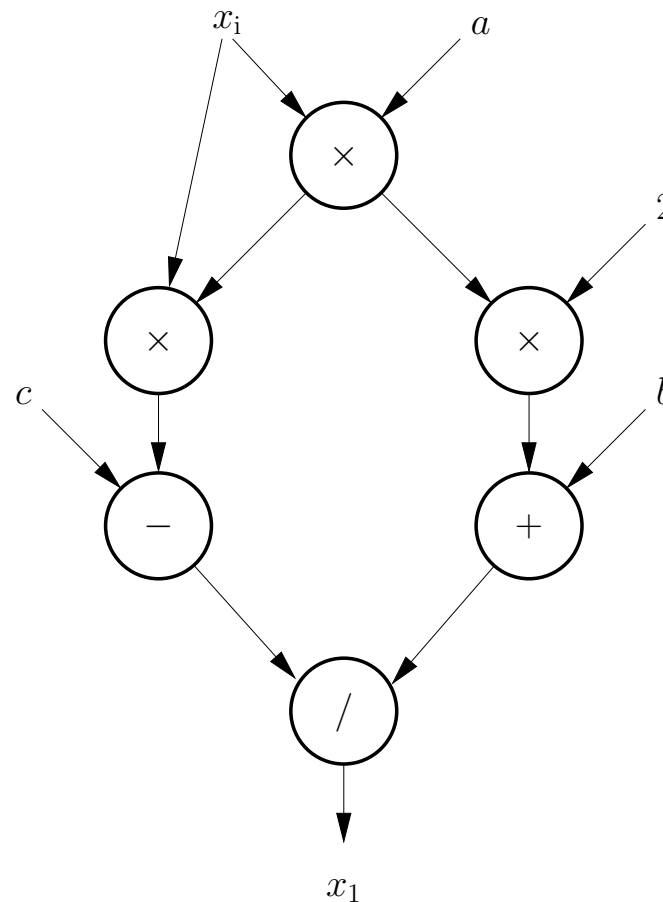
Eine quadratische Funktion $y = ax^2 + bx + c$ ist im Intervall $0 \leq x \leq 255$ definiert. Mit Hilfe des Iterationsverfahrens nach Newton sollen die Nullstellen der Funktion in Abhängigkeit der Parameter a, b, c berechnet werden. Es sei vorausgesetzt, dass reelle Nullstellen existieren, die alle im angegebenen Intervall liegen. Da die Startlösung nicht auf dem Scheitelpunkt liegen darf, werden die Intervallgrenzen benutzt.

Ein das Problem beschreibender VHDL-Quelltext könnte wie folgt aussehen:

```
quadzero: process(a,b,c)
  variable x1,x2,xi integer range 0 to 255;
begin
  x1:=0;                                -- Startwert
  xi:=255;                              -- Vorgänger-Wert weitab
  while(x1/=xi) loop
    xi:=x1;                             -- Vorgänger speichern
    x1:=(a*xi*xi-c)/(2*a*xi+b);          -- Iterationsschritt  $x - \frac{f(x)}{f'(x)}$ 
  end loop;
  x2:=255;                              -- 2. Startwert für 2. Lösung
  xi:=0;
  while(x2/=xi) loop
    xi:=x2;
    x2:=(a*xi*xi-c)/(2*a*xi+b);
  end loop;
end process;
```

Bereichsüberschreitungen von Zwischenergebnissen und Rundungsfehler sind zu vernachlässigen. Für diesen Algorithmus soll der Daten- und der Kontrollflussgraph mit den zugehörigen Links entwickelt werden.

2. Scheduling.



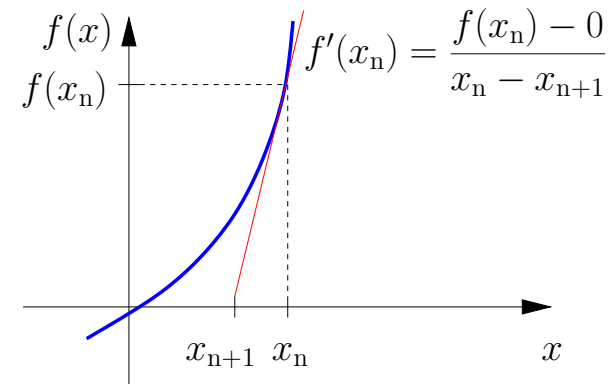
Für die Berechnung, die im oben abgebildeten DFG spezifiziert ist, soll die Anzahl der benötigten Register sowie die Zuordnung aller Variablen inklusive Zwischenergebnisse auf die einzelnen Register bestimmt werden. Dazu sind die Operationen zunächst nach dem ASAP-Scheduling anzuordnen. Die Multiplikation und die Division benötigen zwei Takte, alle anderen (inkl. Shift) einen Takt.

Führen Sie eine Lebensdaueranalyse durch. Berücksichtigen Sie dabei, dass die Parameter a , b und c im nächsten Zyklus wieder benötigt werden.

Newton'sches Näherungsverfahren:

```
quadzero: process(a,b,c)
  variable x1,x2,xi  integer range 0 to 255;
  begin
    x1:=0;
    xi:=255;
    while(x1/=xi) loop
      xi:=x1;
      x1:=(a*xi*xi-c)/(2*a*xi+b);
    end loop;
    x2:=255;
    xi:=0;
    while(x2/=xi) loop
      xi:=x2;
      x2:=(a*xi*xi-c)/(2*a*xi+b);
    end loop;
  end process;
```

Idee: Linearisierung der Funktion in einem Ausgangspunkt (Bestimmung der Tangente); Verwendung der Nullstelle der Tangente als verbesserte Näherung der Nullstelle der Funktion.



$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x - \frac{ax^2 + bx + c}{2ax + b} = \frac{ax^2 - c}{2ax + b}$$

Daten- und Steuerfluss:

```
quadzero: process(a,b,c)
  variable x1,x2,xi integer range 0 to 255;
  begin
    x1:=0;
    xi:=255;
    while(x1/=xi) loop
      xi:=x1;
      x1:=(a*xi*xi-c)/(2*a*xi+b);
    end loop;
    x2:=255;
    xi:=0;
    while(x2/=xi) loop
      xi:=x2;
      x2:=(a*xi*xi-c)/(2*a*xi+b);
    end loop;
  end process;
```

Datenflussgraph

Abfolge von Operationen:

Knoten: Operationen des Algorithmus,

Kanten: kausale Daten-Abhängigkeiten.

⇒ parallele Verknüpfung der Operationen.

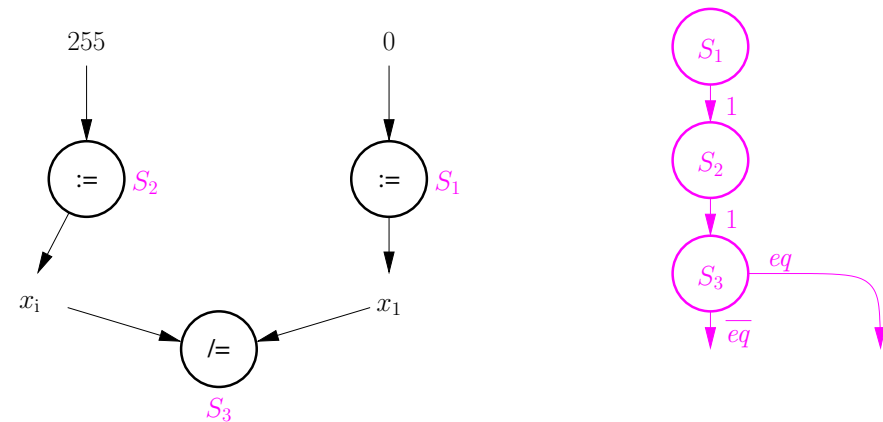
Steuerflussgraph

Abfolge von Zuständen:

Knoten: Zustände repräsentieren Zeitschritte, in denen die Operationen ausgeführt werden,

Kanten: Zustandswechsel, ggf. mit Bedingungen.

⇒ sequentielle Abarbeitung.

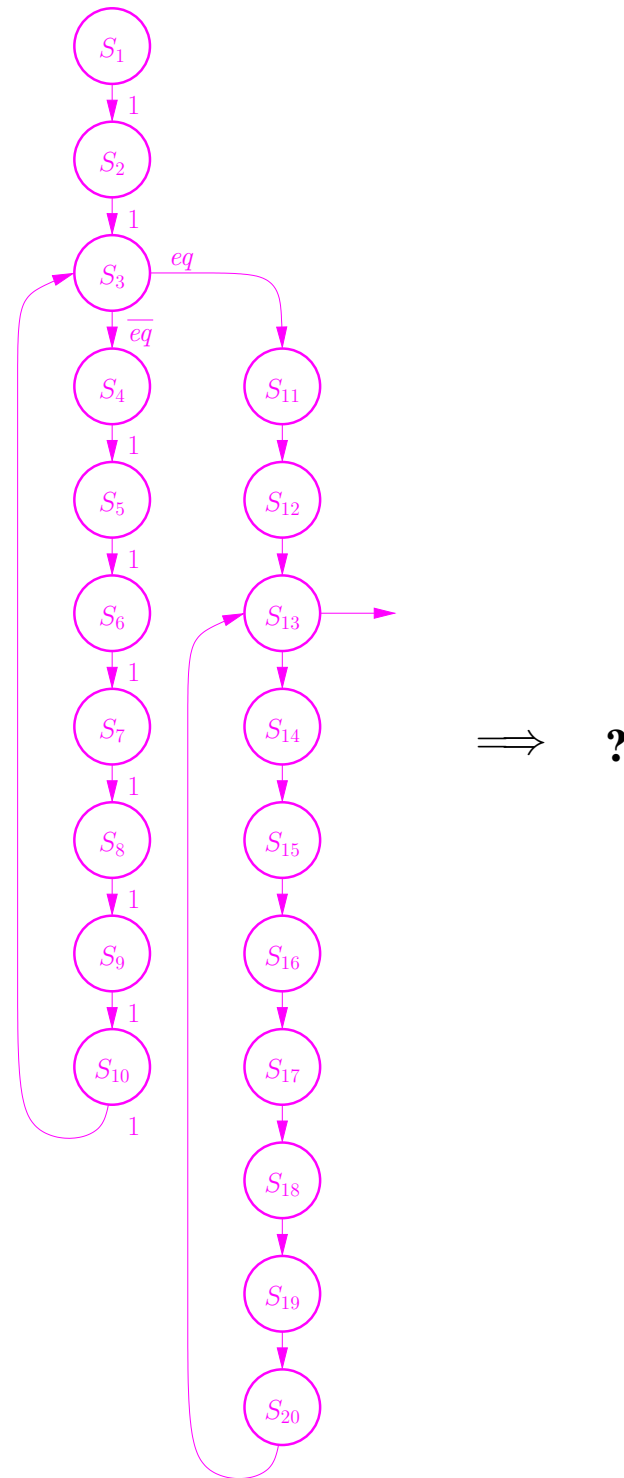


?

Daten- und Steuerfluss:

Steuerfluss für die ideal parallele Realisierung:

- Zusammenfassen unverzweigter Zustandsfolgen zu Superzuständen (*superstates*).



ASAP-Scheduling für *superstate* SS_3 :

Ermittlung der tatsächlichen Parallisierbarkeit und der Mindestanzahl der Taktschritte:

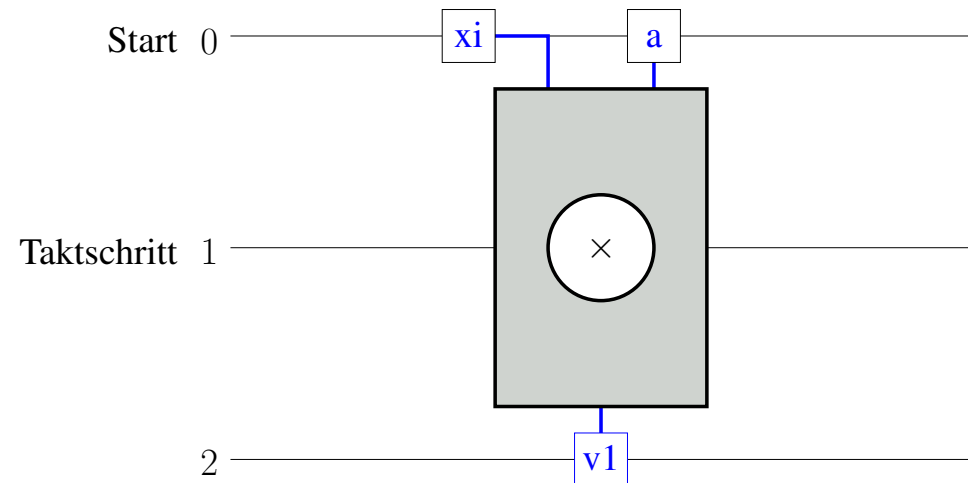
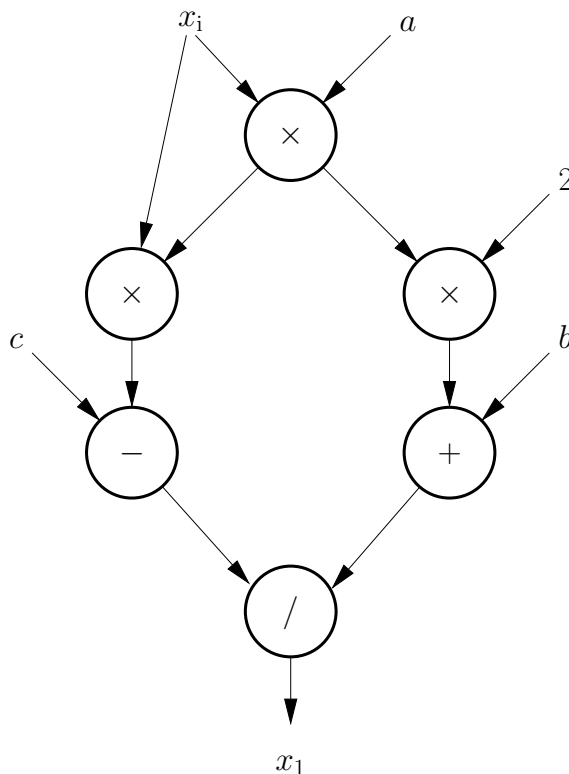
- In einem getakteten System kann eine Operation erst dann ausgeführt werden, wenn die Operanden zur Verfügung stehen.

ASAP (*as soon as possible*): Eine Operation wird eingeplant, sobald alle Operanden zur Verfügung stehen.

ALAP (*as late as possible*): Eine Operation wird erst dann eingeplant, wenn ihr Ergebnis benötigt wird.

- ASAP und ALAP liefern die gleiche minimale Anzahl an Taktschritten für eine parallele Realisierung.
- Vorgaben der Hardware:

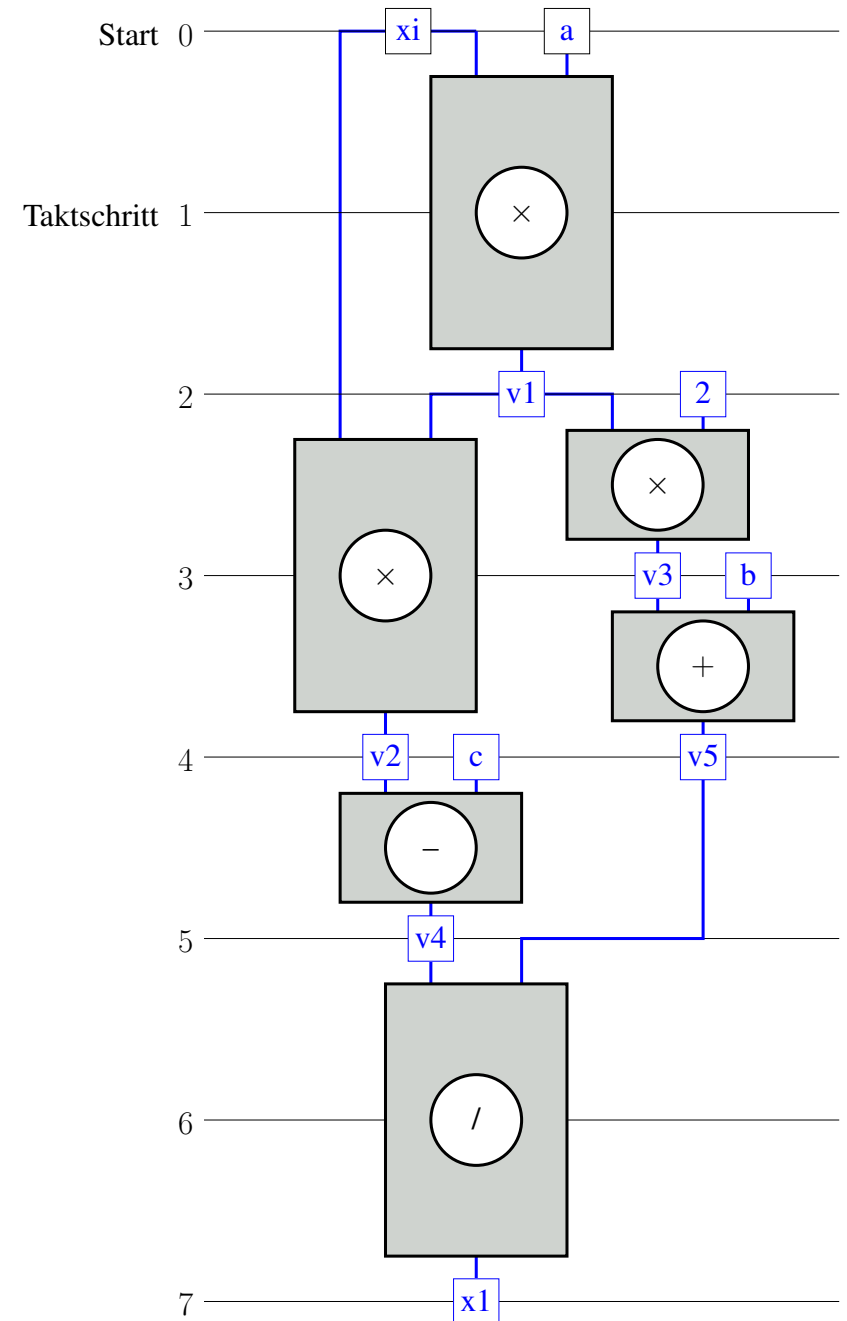
Multiplikation und Division benötigen 2 Taktschritte, alle anderen Operationen 1 Taktschritt.



Lebensdaueranalyse:

Wie lange sind die Variablenbelegungen erforderlich?

Takt	a	b	c	xi	x1	v1	v2	v3	v4	v5
0	L	L	L	L						
1	L	L	L	L						
2	L	L	L	L						
3	L	L	L	L						
4	L	L	L							
5	L	L	L							
6	L	L	L							
7	L	L	L							



Bestimmung der Registerzahl:

Lebensdaueranalyse:

Takt	a	b	c	xi	x1	v1	v2	v3	v4	v5
0	L	L	L	L						
1	L	L	L	L						
2	L	L	L	L		L				
3	L	L	L	L		L		L		
4	L	L	L				L			L
5	L	L	L						L	L
6	L	L	L						L	L
7	L	L	L		L					

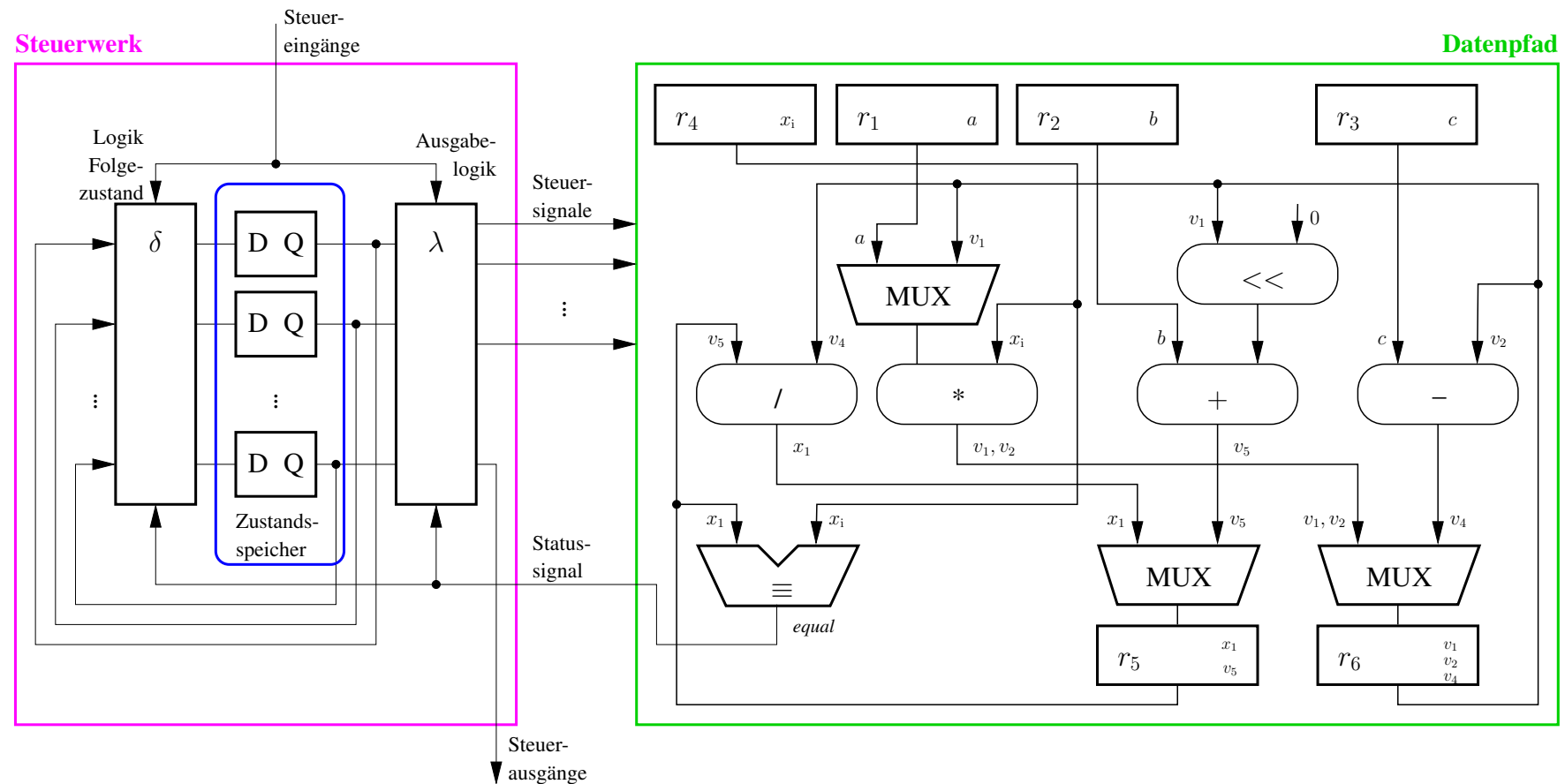
Registerzuordnung:

Register	a	b	c	xi	x1	v1	v2	v3	v4	v5
r_1	×									
r_2		×								
r_3			×					?		
r_4				×						

Es werden ? Datenregister benötigt.

Ergänzung Allocation

- Abschließend werden die Datenoperationen den realen Verarbeitungselementen zugeordnet.
- Es ergibt sich als Register-Transfer-Struktur folgende Realisierung:



Als Steuerworte sind die Register-Takte zur Steuerung der Datenübernahme und die Auswahl-Adressen für die Multiplexer erforderlich.

Verzeichnis der Präsentationen

High-level Synthese

Aufgabenstellung (1)	5. Seminar HB: 1
Aufgabenstellung (2)	5. Seminar HB: 2
Ergänzung: Allocation	5. Seminar HB: 3
	5. Seminar HB: 4

Verzeichnis der Präsentationen

Präsentationen: 1