

Fakultät für Mathematik und Informatik

Lehrgebiet Parallelität & VLSI

**Kurse 1608 "Computersysteme I"
1609 "Computersysteme II"**

Lösungsvorschläge zur

Klausur im WS 2006/2007

Es sind insgesamt 100 Punkte zu erreichen

Aufgabe 1 (10 Punkte)

Sie haben eine große Anzahl von 4-Bit Addierern und eine große Anzahl von Multiplexern aller möglichen Breiten zur Verfügung. Sie sollen hieraus einen 16-Bit Addierer nach dem Conditional-Sum Prinzip aufbauen.

a) Wie viele der 4-Bit Addierer brauchen Sie, wenn Sie keine Optimierungen vornehmen? Begründen Sie kurz Ihre Antwort.

9, denn für einen n -bit Addierer braucht man 3 $n/2$ -bit Addierer, also $3 \cdot 3$ $n/4$ -bit Addierer. $n=16$, $n/4=4$.

b) Teilweise werden bei der Berechnung in a) von 4-Bit Addierern gleiche Berechnungen durchgeführt. Wie viele Addierer benötigen Sie, wenn Sie solche Replikationen eliminieren? Begründen Sie kurz Ihre Antwort.

7, denn die Addition der obersten 4-Bit, mit Eingangsübertrag 0 und 1, kommt jeweils 2mal vor.

Aufgabe 2 (4 Punkte)

Welche der folgenden Aussagen sind wahr?

- a) Ein Minimalpolynom kann auch Monome enthalten, die keine Primimplikanten sind.
- b) Ein Minimalpolynom kann auch Monome enthalten, die keine Kernimplikanten sind.
- c) Ein Minimalpolynom besteht stets aus allen Primimplikanten einer Funktion.
- d) Ein Minimalpolynom enthält stets alle Kernimplikanten einer Funktion.

Aufgabe 3 (4 Punkte)

Welche der folgenden Dezimalzahlen sind im dualen Zweierkomplement mit 6 Stellen einschließlich Vorzeichenbit darstellbar?

- a) -32 da mit 6 Stellen im 2-Komplement Dezimalzahlen von -32 bis +31 dargestellt werden können.
- b) +32
- c) -60
- d) +30 da mit 6 Stellen im 2-Komplement Dezimalzahlen von -32 bis +31 dargestellt werden können.

Aufgabe 4 (14 Punkte)

Gegeben sei ein Synchronzähler nach Abbildung 1.

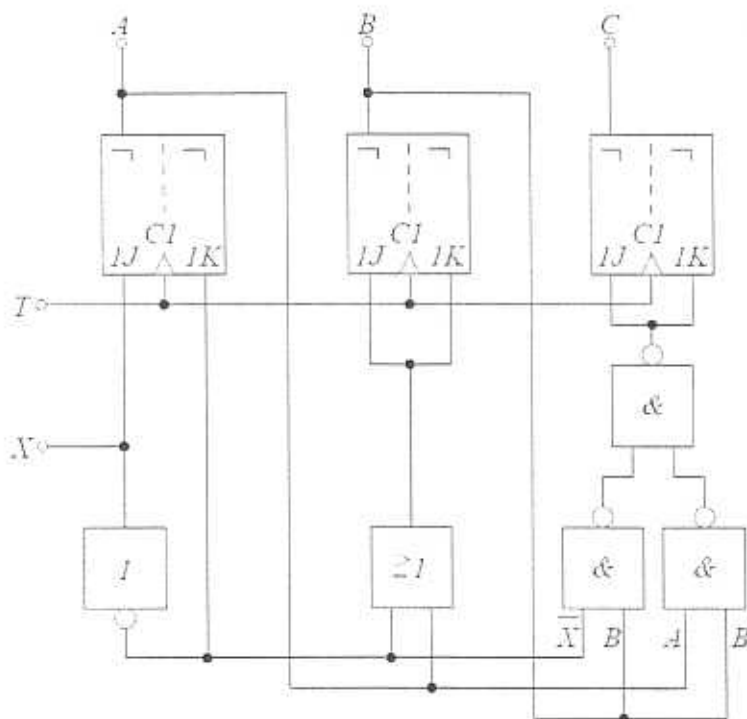


Abbildung 1: Synchronzähler mit drei Ausgängen

- a) Bestimmen Sie ausgehend vom Startzustand $A=B=C=0$ den Zählzyklus für $X=0$ und für $X=1$!
- b) Zeichnen Sie den zugehörigen Zustandgraphen!

a) Die Analyse des Schaltnetzes liefert uns die Schaltfunktionen für die Vorbereitungseingänge J und K der Flipflops.
Die Schaltfunktionen sind:

$$J_A = X \quad K_A = \bar{X}$$

$$J_B = K_B = A \vee \bar{X}$$

$$J_C = K_C = \overline{\bar{X} B \wedge \bar{A} \bar{B}} = \bar{X} B \vee A B$$

Mit der Schaltfunktion der JK-Flipflops können wir die Folgezustandstabelle aufstellen:

X	J_A	K_A	J_B	K_B	J_C	K_C	A_n	B_n	C_n	A_{n+1}	B_{n+1}	C_{n+1}
0	0	1	1	1	0	0	0	0	0	0	1	0
0	0	1	1	1	1	1	0	1	0	0	0	1
0	0	1	1	1	0	0	0	0	1	0	1	1
0	0	1	1	1	1	1	0	1	1	0	0	0
1	1	0	0	0	0	0	0	0	0	1	0	0
1	1	0	1	1	0	0	1	0	0	1	1	0
1	1	0	1	1	1	1	1	1	0	1	0	1
1	1	0	1	1	0	0	1	0	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	0	0

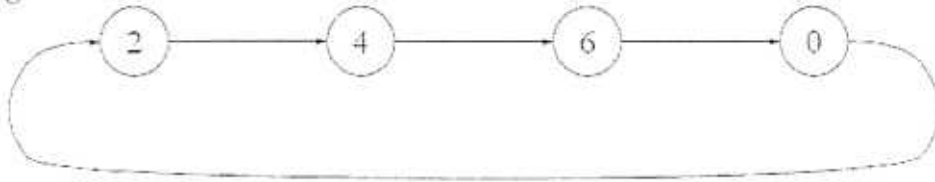
Die eingerahmte Zeile innerhalb der Tabelle, kennzeichnet einen Sonderfall. Der Anfangszustand $ABC = 000$ ist kein Zustand der Zählfolge für $X = 1$. Der Zähler wiederholt seine Zählfolge mit dem Zustand 100. Der Zustand 000 wird also nicht mehr erreicht. Die Zählzyklen lauten somit (wenn $A \hat{=} 2^0, B \hat{=} 2^1, C \hat{=} 2^2$):

für $X = 0$: 0-2-4-6-0- usw.

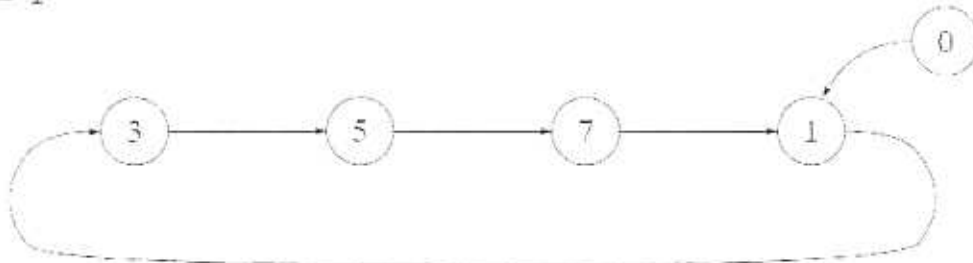
für $X = 1$: 1-3-5-7-1- usw.

b) Zustandsgraph

für $X = 0$



für $X = 1$



Aufgabe 5 (18 Punkte)

Entwerfen Sie ein komplexes Schaltwerk, das die Fakultät berechnet (vgl. Abbildung 2). Zum Aufbau des Operationswerkes stehen Register (mit steuerbarer Taktausblendung), Multiplexer, Multiplizier- und Addier-Schaltnetze mit entsprechenden Wortbreiten zur Verfügung. Der Eingangsvektor X und der Ausgangsvektor Y sind Dualzahlen. Sobald $\text{Start}=1$ wird, startet die Berechnung. Ein gültiges Ergebnis an Y wird durch $\text{Gueltig}=1$ angezeigt.

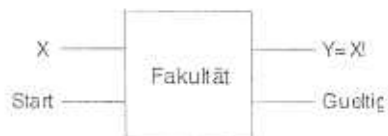
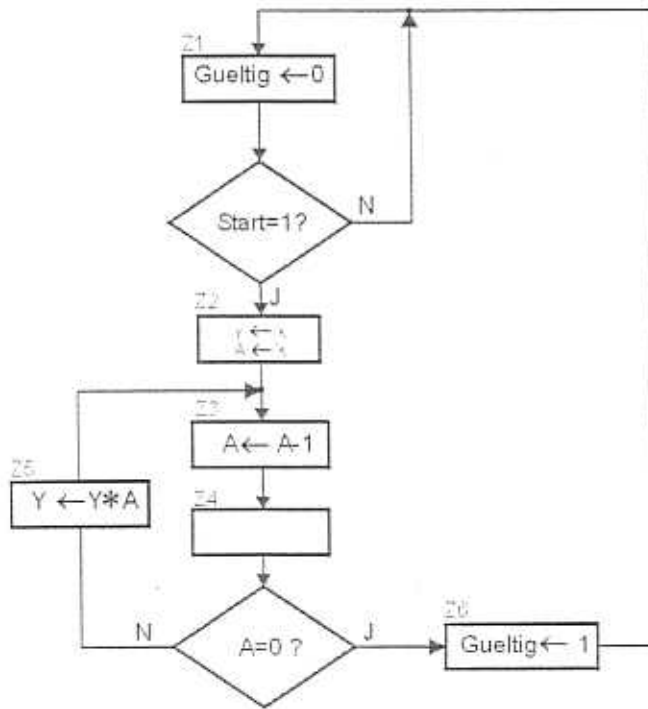


Abbildung 2: Komplexes Schaltwerk zur Berechnung der Fakultät

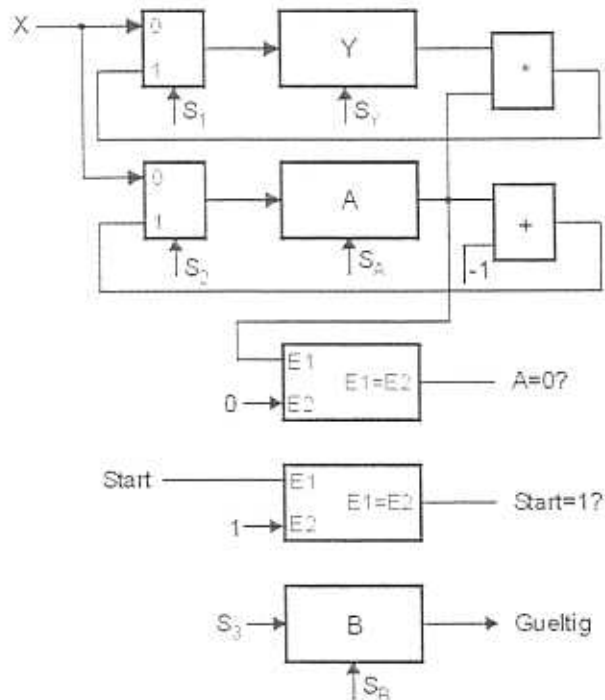
- a) Entwerfen Sie zunächst ein ASM-Diagramm, das mit den o. g. Operationen auskommt!

a) ASM-Diagramm



b) Entwerfen Sie ein geeignetes Operationswerk!

b) Operationswerk



c) Geben Sie für das Operationswerk nach b) für alle vorkommenden Zustände die benötigten Steuervektoren an!

c) Steuervektoren

	SY	SA	SB	S1	S2	S3
Z1	0	0	1	X	X	0
Z2	1	1	0	0	0	X
Z3	0	1	0	X	1	X
Z4	0	0	0	X	X	X
Z5	1	0	0	1	X	X
Z6	0	0	1	X	X	1

Aufgabe 6 (8 Punkte)

Gegeben ist ein Rechner, bei dem entweder jeweils 32 KByte Daten- und Instruktionscaches (Split-Cache) oder ein 64 KByte unified Cache für Daten und Instruktionen untergebracht werden können. Die Caches haben jeweils eine Zugriffszeit von 1 Takt, die Miss penalty ist jeweils 10 Takte. Die Miss rates für ein Benchmark, bei dem jeweils die Hälfte der Zugriffe auf Instruktionen und Daten erfolgen, finden sich in folgender Tabelle:

Größe	Data cache	Instruction cache	Unified Cache
32 KB	5%	1%	
64 KB			4%

a) Wie groß ist die mittlere Zugriffszeit in Taktzyklen für beide Varianten?

Split-Cache: $0,5 * (1 + 5/100 * 10) + 0,5 * (1 + 1/100 * 10) = 1,3$
 Unified Cache: $1 + 4/100 * 10 = 1,4$

b) Ändert sich die Reihung der beiden Varianten bzgl. der mittleren Zugriffszeit, wenn man für die Miss Penalty einen Wert ≥ 100 wählt?

Nein.

c) Ändert sich die Reihung der beiden Varianten bzgl. der mittleren Zugriffszeit, wenn 90% der Speicherzugriffe auf Daten zugreifen und sich die Miss rates nicht verändern?

Ja.

Split-Cache: $0,9 * (1 + 5/100 * 10) + 0,1 * (1 + 1/100 * 10) = 1,46$

Unified Cache: wie a)

Aufgabe 7 (3 Punkte)

Welche der folgenden Aussagen für virtuelle Speicherverwaltung treffen zu?

a) Bei einer segment-basierten virtuellen Speicherverwaltung ohne zusätzliche Seitenverwaltung kann ein Segment größer sein als der Hauptspeicher.

b) Speicherseiten können als Segmente fester Größe betrachtet werden.

c) Es gibt Speicher-Seiten, die nicht ausgelagert werden dürfen.

d) Virtuelle Speicherverwaltung ermöglicht eine einfache Relozierbarkeit von Programmen.

Aufgabe 8 (5 Punkte)

Ein Programm hat einen sequentiellen Anteil von $a=25\%$.

a) Welchen Speedup kann man nach Amdahls Gesetz mit $n=3$ Prozessoren erreichen?

2 lt. Formel

b) Wie viele Prozessoren muss man mindestens einsetzen, um 75% des nach Amdahl maximal möglichen Speedups erreichen zu können?

max. Speedup lt. Amdahl = 4, Speedup 3 erreicht bei $n \geq 9$ Prozessoren

Aufgabe 9 (3 Punkte)

Ein Parallelrechner hat eine Startupzeit T_s , die 100-mal so groß ist wie die Transferzeit pro Wort T_w . Ein Programm für diesen Rechner besteht aus k Berechnungsrunden, die jeweils von einer Kommunikation gefolgt werden. Es gibt keine Idle-Zeiten. Das Programm kann auf zwei Arten realisiert werden: In Version 1 werden $k=10$ Runden benötigt, die Berechnungszeit pro Runde ist $T_R = 100 \cdot T_w$, und jede Nachricht enthält $L=10$ Worte. In Version 2 werden $k=20$ Runden benötigt, die Berechnungszeit pro Runde ist $T_R = 40 \cdot T_w$, und jede Nachricht enthält $L=5$ Worte. Welche der folgenden Aussagen sind wahr?

- a) Die Berechnungszeit ist in Version 1 höher.
- b) Die Kommunikationszeit ist in Version 1 höher.
- c) Die Ausführungszeit ist in Version 1 höher.

a) In Version 1 ist Ausf.zeit = $10 \cdot 100 \cdot T_w = 1000 \cdot T_w$, in Version 2 = $20 \cdot 40 \cdot T_w = 800 \cdot T_w$, also ist die Aussage wahr.

b) Die Kommunikationszeit in Version 1 ist $10 \cdot (100 \cdot T_w + 10 \cdot T_w) = 1100 \cdot T_w$, in Version 2 = $20 \cdot (100 \cdot T_w + 5 \cdot T_w) = 2100 \cdot T_w$, also ist die Aussage nicht wahr.

c) Die Ausführungszeit ist in Version 1 = $1000 \cdot T_w + 1100 \cdot T_w = 2100 \cdot T_w$, in Version 2 = $800 \cdot T_w + 2100 \cdot T_w = 2900 \cdot T_w$, also ist die Aussage nicht wahr.

Aufgabe 10 (4 Punkte)

Warum werden in CISC-Prozessoren die Maschinenbefehle durch Mikroprogrammierung implementiert? Kreuzen Sie zutreffende Aussagen an!

Durch Mikroprogrammierung werden Taktzyklen eingespart.

Es können nur Maschinenbefehle implementiert werden, die eine konstante Anzahl von Taktzyklen erfordern.

Maschinenbefehle, die eine unterschiedliche Anzahl von Taktzyklen erfordern, können einfacher implementiert werden.

Mikroprogrammierung erlaubt die Implementierung komplexer Adressierungsarten.

Mikroprogrammierte Leitwerke arbeiten schneller als festverdrahtete Leitwerke.

Die Befehle können kompakt, d.h. mit wenigen Bits, codiert werden. Kurze Opcodes haben einen geringeren Hauptspeicherbedarf.

Aufgabe 11 (8 Punkte)

Gegeben sei eine DLX-Mikroarchitektur mit 5-stufiger Pipeline (IF-ID-EX-MEM-WB) und ein Programm mit 12 Befehlen. Es wird angenommen, dass keinerlei Pipelinekonflikte bestehen.

a) Nach wie vielen Taktzyklen ist das Programm beendet?

16 Taktzyklen

b) Der Prozessor sei mit einer Frequenz von 3,2 GHz getaktet. Wie lange dauert die Ausführung des Programms?

5 ns

c) Wie lautet die allgemeine Gleichung für die Anzahl der Taktzyklen T bei p Pipelinestufen und n Befehlen?

$T = p + (n-1)$

d) Wir nehmen nun an, dass zwischen dem 1. und 2. Befehl des Programms ein Lese-nach-Schreibe-Konflikt (RAW) besteht und dass keine Bypass-Hardware vorhanden ist. Um wie viele Taktzyklen verlängert sich die Programmausführung?

2 Taktzyklen

Aufgabe 12 (8 Punkte)

Gegeben sei folgendes DLX-Programm:

```
1: DIV R2, R5, R8
2: SUB R9, R2, R7
3: XOR R5, R14, R6
4: MUL R11, R9, R5
5: BEQZ R10, #0, R12
6: OR R8, R15, R2
```

Geben Sie alle Befehlspaare an, zwischen denen ein

a) Lese-nach-Schreibe-Konflikt (RAW),

1-2, 3-4, 2-4, 1-6

b) Schreibe-nach-Lese-Konflikt (WAR),

1-3, 1-6

c) Schreibe-nach-Schreibe-Konflikt (WAW) und

Es gibt keine WAW-Konflikte

d) ein Steuerfluss-Konflikt besteht!

5-6

Aufgabe 13 (11 Punkte)

Gegeben sei folgendes Programm-Fragment für einen Superskalar-Prozessor:

```
ADD R1, R2, R3
LW R4, (R5)
SUB R7, R1, R9
MUL R5, R4, R4
SUB R1, R12, R10
SW (R13), R14
OR R15, R14, R12
```

Wir betrachten im Folgenden n-fach Superskalar-Prozessoren. Alle Ausführungseinheiten können jede der im Programm vorkommenden Operationen ausführen. Alle Operationen können in einem Taktzyklus ausgeführt werden.

a) Wie viele Taktzyklen werden maximal auf einem 2-fach Superskalar-Prozessor benötigt? Geben Sie die in jedem Taktzyklus ausgeführten Befehle an!

Bei 2-facher Superskalarität werden 4 Taktzyklen benötigt (ohne Pipelinelatenz).

ADD R1, R2, R3	LW R4, (R5)
SUB R7, R1, R9	MUL R5, R4, R4
SUB R1, R12, R10	SW (R13), R14
OR R15, R14, R12	

b) Klassifizieren Sie die Art des in der Lösung von a) verwendeten Scheduling (In-Order oder Out-of-Order) und begründen Sie dies kurz!

In a) wurde In-Order-Scheduling verwendet.

c) wie a) allerdings mit 4-facher Superskalarität!

Bei 4-facher Superskalarität werden 2 Taktzyklen benötigt (ohne Pipelinelatenz).

ADD R1, R2, R3 SUB R7, R1, R9	LW R4, (R5) MUL R5, R4, R4	SW (R13), R14 SUB R1, R12, R10	OR R15, R14, R12
----------------------------------	-------------------------------	--------------------------------------	---------------------

Beachte den WAR-Konflikt zwischen den SUB R7 ... und SUB R1 ... !
Daher müssen diese beiden Befehle gleichzeitig verplant werden. Dazu wird der SW-Befehl vorgezogen, d.h. eine Ausführung in 2 Taktzyklen ist nur bei Out-of-Order Scheduling möglich.

d) Klassifizieren Sie die Art des in der Lösung von c) verwendeten Scheduling (In-Order oder Out-of-Order) und begründen Sie dies kurz!

In c) wurde Out-of-Order-Scheduling verwendet.