Klausur: Rechnerarchitektur

SoSe 2013

19.7.2013

1 Multiple Choice

1.1

a) Die ALU enthält einen Akkumulator, der zur Aufnahme von Operanden dient.

b) Der Datenprozessor einer CPU ist zuständig für das Entschlüsseln von Befehlen und zur Steuerung der Ausführung.

c) Bevor die ALU aktiv wird (innerhalb eines Befehlszyklus), muss zunächst das Ergebnis des Dekodierers vorliegen.

d) Durch Anwendung von Pipelining werden einzelne Maschinenbefehle schneller ausgeführt.

e) Der sog. von-Neumann-Flaschenhals tritt auf, wenn der Speicher deutlich schneller ist als die Abarbeitungszeit innerhalb der CPU.

1.2

a) Eine Boolsche Funktion kann allgemein n binäre Eingaben auf m binäre Ausgaben abbilden mit $n \neq m$.

b) Ein Schaltnetz kann sowohl eine Boolsche Funktion als auch eine Schaltfunktion realisieren.

c) NOR-Bausteine reichen aus, um jede Boolsche Funktion durch Schalter zu realisieren.

d) Ein Decoder hat 2^n Eingabewerte und bildet diese auf n Ausgänge ab.

e) Ein MUX erlaubt das Selektieren eines Eingabewertes mittels einer Steuerleitung.

1.3

a) Jede Technik der Fehlererkennung erlaubt auch die anschließende Korrektur des Fehlers.

b) Durch Hinzunahme von Prüfbits zu Speicherwerten kann Fehlererkennung ermöglicht werden.

c) Der Hamming-Abstand d gibt an, dass max. d-1 Einzelbitfehler auftreten können, sodass der Fehler noch erkannt wird.

d) Zur Fehlerkorrektur von x Fehlern wird Code benötigt, dessen Wörter mind. den Abstand 2x+1 haben.

e) Der Hamming-Algorithmus setzt solche Bits als Paritätsbits, deren Index eine Zweierpotenz ist.

1.4

a) Die grundlegende Annahme von Cache-Speichern ist die temporale und spatiale Lokalität.

b) Die Größe von Cache-Zeilen entspricht immer der Größe von CPU-Registern.

c) Das valid bit v von Cache-Einträgen gibt an, ob enthaltene Daten aktuell benötigt werden.

d) Der Vorteil des Split-Caches ist die Vergrößerung der Bandbreite ohne Vergrößerung der Latenz.

e) Tags von Cache-Zeilen dienen der Speicherung bisheriger Zugriffe.

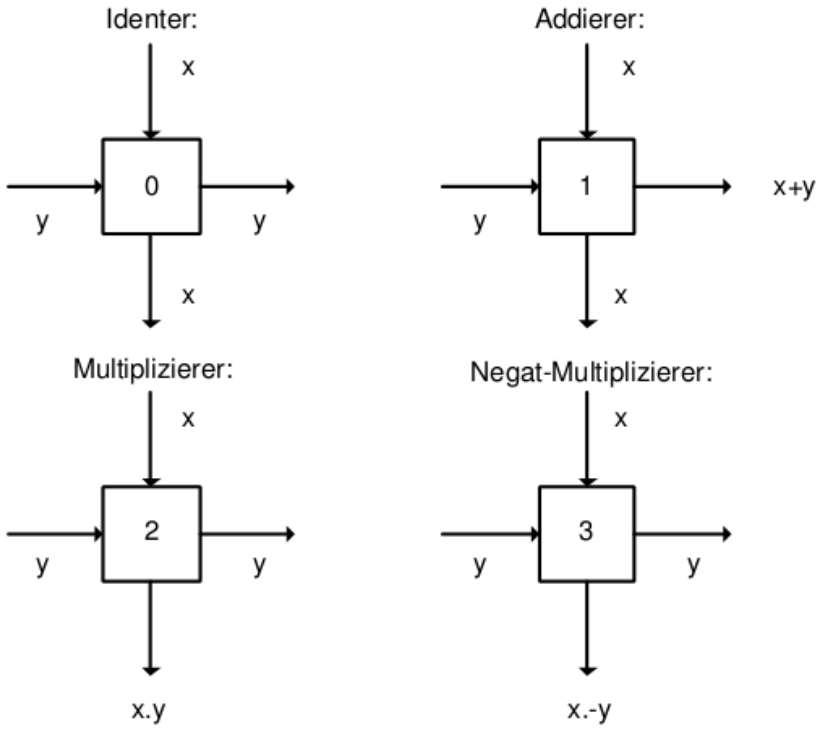
2 PLA

a)

Erläutern Sie das Konzept von PLA's.

b)

Ein PLA besitzt folgende Elemente:



Zeichnen Sie eine NOT-Schaltfunktion auf, wobei nur das NOR-Gatter verwendet werden darf. Auf Basis dieser Überlegungen soll anschließend das Schaltbild für den Typ-3: Negat-Multiplikations-Baustein mittels NOR-Gatter entwickelt werden. Verwenden Sie ausschließlich NOR-Gatter dazu.

c)

Begründen Sie, warum es empfehlenswert ist, eine Boolsche Funktion zunächst in DNF zu überführen, um sie auf PLA's zu implementieren.

d)

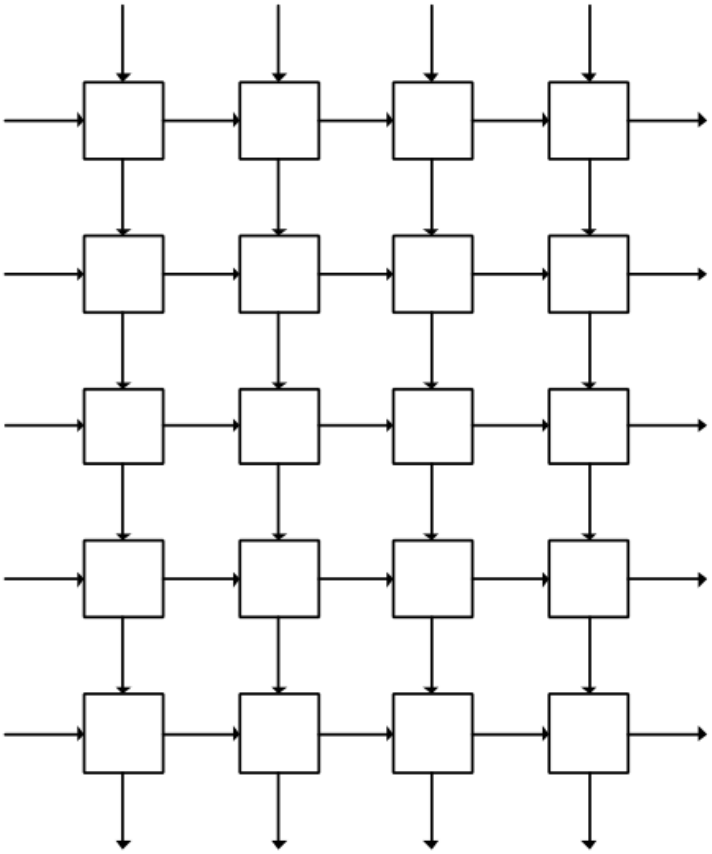
Ein normierter PLA besteht aus einer UND-Ebene und einer ODER-Ebene. Erklären Sie deren Aufgabe je kurz.

Ausgehend von einem 6x5 PLA (Zeilen x Spalten): Wie viele Zeilen umfassen die UND- und die ODER-Ebene jeweils, wenn durch den PLA eine 5-stellige Boolsche Funktion realisiert werden soll. Wie viele Produktionen darf die boolsche Funktion maximal besitzen?

e)

Gegeben ist die Funktion f(a,b,c) = (-a.-b)+(a.-b.c)+(-a.c)

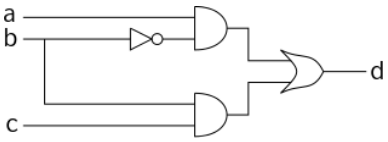
Realisieren Sie die Funktion im folgenden PLA:



3

a)

Gegeben ist folgendes Schaltnetz, das die boolsche Funktion f(a,b,c)=d implementiert.



Leiten Sie aus dem Schaltnetz die Definition der Funktion ab und geben Sie diese in disjunkter Form an. Implementiert das Schaltnetz einen bekannten logischen Baustein? Wenn ja, welchen?

b)

Gegeben ist die Wahrheitstabelle von $g(x_1, x_2, x_3, x_4)$, wobei einige Ergebnisse unbestimmt sind (□).

x_1	x_2	x_3	x_4	$g(x_1, x_2, x_3, x_4)$
0	0	0	0	1
0	0	0	1	0
0	0	1	0	□
0	0	1	1	□
0	1	0	0	1
0	1	0	1	0
0	1	1	0	□
0	1	1	1	□
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	□
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	□

Minimieren Sie die Funktion graphisch mit Karnaugh und geben Sie die minimierte Funktion in disjunkter Form an.

c

Gegeben ist die boolsche Funktion $h(x_1, x_2, x_3, x_4) = x_1. - x_2.x_3.x_4 + -x_1.x_2.x_3.x_4 + x_1.x_2.-x_3.x_4 + -x_1.x_2.-x_3.x_4 + -x_1.x_2.x_3.-x_4 + -x_1.-x_2.x_3.-x_4 + -x_1.-x_2.-x_3.-x_4$
Liegt die Funktion in disjunkter oder konjunkter Form vor?

4 Zweierkomplement

a)

Geben Sie zwei Vorteile vom Zweierkomplement gegenüber der Vorzeichen-/Betragsdarstellung in Rechnern an.

b)

Beantworten Sie folgende Fragen im Bezug auf die 2er-Komplement-Darstellung ganzer Zahlen bei der Verwendung von 8 Bits inklusive Vorzeichenbit.

i) die größte darstellbare Binärzahl

ii) die größte darstellbare Dezimalzahl

iii) die kleinste darstellbare Binärzahl

iv) die kleinste darstellbare Dezimalzahl

v) Nulldarstellung

c)

Folgende Zahlen sind gegeben:

i) $x = (27)_{10}$

ii) $y = (-30)_{10}$

Konvertieren Sie beide Zahlen in 2er-Komplement. (6 Bits)

d)

Gegeben sind die Zahlen u=100110 und v=101111 im 2er-Komplement (6 Bits). Addieren Sie beide Zahlen. Der Rechenweg soll erkennbar sein. Hat ein Überlauf stattgefunden?

e)

Wandeln sie folgende Zahl (IEEE) in Dezimaldarstellung um. (Die Darstellung als Bruch ist auch in Ordnung.)

S	Exponent	Signifikant
1	01111101	1011 0000 0000 0000 0000 000

5 Addiernetze

a)

Geben Sie die Wahrheitstafel eines Halbaddierers und dessen Schaltfunktion in minimierter Form an.

b)

Zeichnen Sie das Schaltnetz eines Halbaddierers. Beschriften Sie dabei Ein- und Ausgänge mit x,y,R,U.

c)

Erklären Sie die zusätzlichen Ein- und Ausgänge eines Volladdieres gegenüber einem Halbaddierer. Wozu wird ein Volladdierer benötigt?

d)

Was ist der Nachteil von Addiernetzen für n-stellige Dualzahlen bezüglich der Ausführungsdauer, wenn Volladdierer und Halbaddierer für die Konstruktion verwendet werden.

e)

Zeichnen Sie das Schaltnetz eines Volladdierers. Der Haldaddierer darf dabei als Symbol dargestellt werden.

f)

Zeichnen Sie das Schaltbild eines Ripple-Carry zur Aufnahme von zwei 3-Bit Zahlen. Als Bauteile werden die Symbole HA und VA verwendet.

g)

Was ist der Unterschied zwischen dem Carry-Save und dem Ripple-Carry? Welche Vor- und Nachteile bestehen bezüglich Zeitverhalten und Schaltaufwand?

6 Flip-Flops

a)

Erklären Sie kurz den Unterschied zwischen Latches und Flip-Flops.

b)

Welche Eigenschaften besitzen Flip-Flops, die für den Bau von Computern so wichtig sind?

c)

Gleichzeitiges Setzen von R und S kann beim R-S-Flip-Flop zu einem undefinierten Zustand führen. Wann genau trifft das Problem im Bezug auf den Steuertakt zu?

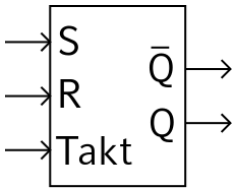
d)

Skizzieren Sie das Schaltwerk eines R-S-Flip-Flops. Kennzeichnen Sie ihn mit S,R,C,Q,$\overline{Q}$.

e)

Der R-S-Flip-Flop hat den Nachteil, dass R=1 und S=1 undefiniert ist. Der D-Flip-Flop hat dieses Problem nicht, und ist dehalb eine Erweiterung des R-S-FLip-Flops.

Erweitern Sie das Symbol zu einem D-Flip-FLop. Kennzeichnen Sie dabei die Steuerleitung mit D=Delay.

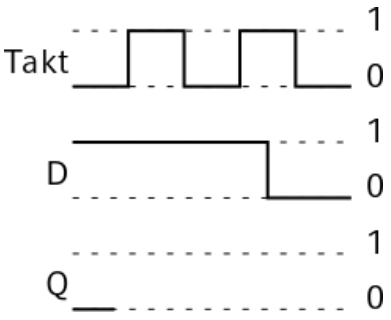


f)

Stellen Sie die Zustandstabelle eines D-FLip-Flops auf. Q^* sei dabei der alte Wert von Q. X, wenn der Zustand unerlaubt ist.

g)

Vervollständigen Sie das Impulsdiagramm eines D-Flip-Flops. Dabei wird angenommen, dass der D-Flip-Flop bei steigender Flanke schaltet und die Bausteine ohne Verzögerung schalten.



7 SPIM

a)

Jeder Zeile mit *#Kommantar-Nr:* einen Kommentar zuordnen. Nicht alle werden benötigt.

i) Ergebnis := 0

ii) Ergebnis = Ergebnis-1

iii) Ergebnis = Ergebnis + i

iv) i := 0

v) i = i + 1

vi) 4 = print_str

vii) Sprung zum Anfang der Schleife

viii) Verlasse Schleife, falls i >N

ix) Verlasse Schleife, falls i <N

Code:

```
1  .data
2  sMsg = .asciiz "Ergebnis_berechnet:\n"  # Fuer Ergebnisanzeige
3  .text
4  main:
5          li      $v0,5                   # 5=read_int
6          syscall
7          move    $t0,$v0                 # N=$t0
8          li      $t1,0                   # Kommentar-Nr:
9          li      $t2,0                   # Kommentar-Nr:
10 while:
11         blt     $t0,$t1,elihw           # Kommentar-Nr:
12         add     $t2,$t2,$t1             # Kommentar-Nr:
13         addi    $t1,$t1,1               # Kommentar-Nr:
14         b       while
15
16 elihw:
17         li      $v0, 4                  # Kommentar-Nr:
```

```
18          la      $a0, sMsg
19          syscall
20          li      $v0,1                   # 1: print_int
21          move    $a0, $t2
22          syscall
```

b)

Was berechnet das Programm?

c)

Skizzieren Sie den Inhalt des Stacks nachdem alles ausgeführt wurde. Die Position des Stackpointers soll auch gekennzeichnet werden. Welches Problem tritt bezüglich der Operation in Zeile 4 und 5 auf?

```
1           li      $t0,1
2           li      $t1,2
3           li      $t2,3
4
5           sw      $t0,($sp)
6           sw      $t1,($sp)
7
8           addi    $sp,-12
9           sw      $t0,12($sp)
10          sw      $t1,8($sp)
11          sw      $t2,4($sp)
```

d)

Gegeben ist folgendes Programm, das die Unterprozedur dbl aufruft, die den Wert des übergebenen Integer verdoppelt. Der Aufruf erfolgt über Call-by-Value.

```
1           .data
2           x: word 23
3           .text
4
```

```
5           main:   lw      $a0,x           # Wert von Adresse x in $a0
6                   jal     dbl
7                   sw      $v0,x
8                   jal     exit
9
10          dbl:    move    $t0,$a0
11                  add     $t0,$t0,$t0
12                  move    $v0,$t0         # Speichere Ergebnis in $v0
13                  jr      $ra
14
15          exit:
```

Ändern Sie das Programm so, dass der Aufruf über Call-by-Reference erfolgt. Dazu dürfen maximal 3 Befehle ergänzt werden.

```
1           .data
2           x: word 23
3           .text
4
5           main:
6                   jal     dbl
7
8                   jal     exit
9
10          dbl:
11                  add     $t0,$t0,$t0
12
13                  jr      $ra
14
15          exit:
```