Aufgaben

Kim Thuong Ngo

April 26, 2018

CONTENTS

1 Zahlensysteme

2 Interpretation von Hexadezimalzahlen

In den folgenden Tabelle sind in den ersten Spalten Zahlen im Hexadezimalsystem angegeben. Ergänzen Sie die folgende Tabelle von A)-E)!

a) Schreiben Sie die hexadezimale Zahl als binären String. Dabei soll der binäre String genauso viele Stellen besitzen, um jede zweistellige Hexadezimalzahl darstellen zu können.

b) Interpretieren Sie den binären String aus a) als vorzeichenlose Binärzahl. Geben Sie die Zahl im Dezimalsystem an.

c) Interpretieren Sie den binären String aus a) als Zweierkomplement. Geben Sie die Zahl im Dezimalsystem an.

d) Negieren Sie die Zahl aus c) und schreiben Sie diesen Wert als binären String im Zweierkomplement auf.

e) Schreiben Sie die Zahl aus c) als 16-stelligen binären String im Zweierkomplement aus.

	A)	B)	C)	D)	E)
0x20	$0010\ 0000_2$	32_{10}	32_{10}	$1110\ 0000_2$	$0000\ 0000\ 0010\ 0000_2$
0xA2	$1010\ 0010_2$	162_{10}	-94_{10}	$0101\ 1110_2$	$1111\ 1111\ 1010\ 0010_2$
0x9F	$1001\ 1111_2$	159_{10}	-97_{10}	$0110\ 0001_2$	$1111\ 1111\ 1001\ 1111_2$
0xC2	$1100\ 0010_2$	194_{10}	-62_{10}	$0011\ 1110_2$	$1111\ 1111\ 1100\ 0010_2$
0x54	$0101\ 0100_2$	84_{10}	84_{10}	$1010\ 1100_2$	$0000\ 0000\ 0101\ 0100_2$
0x92	$1001\ 0010_2$	146_{10}	146_{10}	$0110\ 1110_2$	$1111\ 1111\ 1001\ 0010_2$
0x3C	$0011\ 1100_2$	60_{10}	196_{10}	$1100\ 0100_2$	$0000\ 0000\ 0011\ 1100_2$

3 Umwandlung von Dezimalzahlen

In der folgenden Tabelle sind in der ersten Spalte Zahlen im Dezimalsystem angegeben. Ergänzen Sie die folgende Tabelle von A)-C)!

a) Schreiben Sie den Absolutbetrag der Dezimalzahl als Hexadezimalzahl auf.

b) Schreiben Sie den Absolutbetrag der Dezimalzahl als binären String auf.

c) Schreiben Sie die Dezimalzahl als binären String im Zweierkomplement auf.

	A)	B)	C)
-24	0x18	$0001\ 1000_2$	$1110\ 1000_2$
-119	0x77	$0111\ 0111_2$	$1000\ 1001_2$
82	0x52	$0101\ 0010_2$	$0101\ 0010_2$
125	0x7D	$0111\ 1101_2$	$0111\ 1101_2$
-17	0x19	$0001\ 1001_2$	$1110\ 0111_2$

4 GLEITKOMMAZAHL IN IEEE 754

4.1 UMWANDLUNG EINER DEZIMALEN GLEITKOMMAZAHL IN EINE HEXADEZIMAL CODIERTE IEEE 754 ZAHL

Der 16-Bit Gleitkommatyp binary16 des IEEE-Standards 754 verwendet 1 Bit für das Vorzeichen s, 5 Bit für den Exponenten E und 10 Bit um f, den expliziten Teil der Mantisse zu speichern. Codieren Sie die Zahl $X = \frac{1}{125}$ in diesem Format und stellen Sie den daraus resultierende Bitfolge als Hexadezimalzahl dar!

1. Bringen Sie die Zahl X in das Format $X = (-1)^s * (1+f) * 2^e$ mit $s\epsilon\{0,1\}$, $e\epsilon\mathbb{Z}$, $B = 15$, $E = e + B$, $E\epsilon\mathbb{N}$ und $0 \leqslant f < 1$!

$X = \frac{1}{125} = 0,008$
da X positiv, s=0

2. Ermitteln Sie die Repräsentation für s,E und f in Binärschreibweise!

3. Fügen Sie die 16 Bit in der richtigen Reihenfolge zusammen!

Reihenfolge: Sign Bit, Exponent, Mantisse

4. Warum ist die im Standard vorgeschriebene Reihenfolge sinnvoller als andere Anordnungsmöglichkeiten?

lexikalische Ordnung → Vergleichsoperationen effizient durchführbar

5. Geben Sie die ermittelte Bitfolge in (kompakter) hexadezimaler Darstellung an!

4.2 Umwandlung einer hexadezimal codierten IEEE 754 Zahl in eine dezimal codierte Gleitkommazahl

Welche dezimale Gleitkommazahl stellt $BCDE_{16}$ dar, wenn man IEEE 754 Codierung wie in der vorherigen Aufgabe zu Grunde legt?

$X = BCDE_{16}$
$BCDE_{16} \rightarrow 1011\ 1100\ 1101\ 1110_2$

5 Kommunikationsnetze, Restklassenrechnung und Huffman-Code

5.1 Schichten in Kommunikationsnetzen

Ein Host in einem WLAN-Netz fragt über HTTP/ TCP/ IP bei einem Webserver eine Webseite an. Das Datenpaket wird von diesem Host über ein WLAN-Netz an einen IP-Router gesendet, der dieses über ein Ethernet-Netz an einen Switch und von dort aus an den Web-Server weiterleitet.

1. Zu welchen Schichten des ISO/OSI-Schichtenmodells gehören die Protokolle HTTP, TCP, IP, WLAN und Ethernet? Recherchieren Sie dazu im Internet!

Sieben-Schichtenmodell:

7 Application	Anwendung
6 Presentation	Darstellung
5 Session	Kommunikationssteuerung
4 Transport	Transport
3 Network	Vermittlung
2 Data Link	Sicherung
1 Physical	Bitübertragung

2. Durch welche Technologie wird die IP-Zieladresse des Webservers vom anfragenden Host ermittelt?

3. Wie oben beschrieben wird das Datenpaket nun versendet. Erklären Sie kurz was mit dem Paket passiert.

5.2 Polynom-Restklassenrechnung

1. Überprüfen Sie, ob die folgenden Polynome primitiv sind:

$$g_1(u) = u^4 + u^3 + 1$$
$$g_2(u) = u^4 + u^2 + u + 1$$
$$g_3(u) = u^4 + u^3 + u + 1$$

2. Welche Polynome sind durch u+1 teilbar?

3. Können Sie eine einfache Regel aufstellen, anhand der man schnell entscheiden kann, ob ein Polynom durch u+1 teilbar ist?

5.3 HUFFMAN-CODE

Der Huffman-Code bietet eine Möglichkeit einen Code mit minimaler mittlerer Codewortlänge zu konstruieren. Es wird folgende Codiervorschrift angewandt:

1. Die Zeichen einer Quelle werden in einer Tabelle nach fallenden Auftrittswahrscheinlichkeiten aufgelistet und die Auftrittswahrscheinlichkeiten werden eingetragen.
2. Die beiden kleinstwahrscheinlichen Zeichen x,y werden zur Unterscheidung mit 0 und 1 codiert und in der Tabelle entsprechend gekennzeichnet.
3. Die beiden Zeichen x und y werden zu einem neuen Zeichen xy zusammengefasst. Dem neuen Zeichen wird die Summe der Wahrscheinlichkeiten der beiden ursprünglichen Zeichen zugeordnet. Die so entstehende Quelle hat ein Zeichen weniger. Falls die neue Quelle nur noch ein Zeichen enthält, fährt man mit Schritt 4 fort, sonst wiederholt man den Algorithmus ab Schritt 1 mit der neuen Quelle.
4. Man beginnt mit der letzten Tabelle, arbeitet sich bis zur ersten Tabelle vor und stellt den Codebaum auf. Pro Tabelle erhält man eine Codierentscheidung, d.h. zwei Zweige des Codebaumes. Die Endknoten liefern die gewünschte Codierung.

Das Zeichenalphabet einer Nachrichtenquelle umfasste 7 Zeichen $x_1, ..., x_7$, die mit folgenden Wahrscheinlichkeiten auftreten:

x_i	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$p(x_i)$	0,10	0,30	0,10	0,10	0,15	0,10	0,15

1. Wie groß ist der Informationsgehalt H* der Nachrichtenquelle?

2. Man gebe die Optimalcodierung des Zeichenalphabets mit Hilfe des Verfahrens von Huffman an!

3. Wie groß ist die Coderedundanz R_c? Man vergleiche diesen Wert mit dem entsprechenden Wert für die Codierung mit einheitlicher Binärstellenzahl.

6 CODE

6.1 LINEAR SYSTEMATISCHER CODE

Bei einer digitalen Nachrichtenübertragung sollen mit einem linearen systematischen Code 2 Fehler erkannt und 1 Fehler sicher korrigiert werden können.

1. Bestimmen Sie die Code-Effizienz $\frac{m}{n} = f(k), k = 1,2,...,12$ und stellen Sie diese graphisch dar.

2. Geben Sie ein Prüfschema für eine Codierung mit $m = 11$ Nutzbits an!

3. Wie lauten die Kontrollstellen für das Codewort mit den Nachrichtenstellen 0110 1010 110?

4. Man gebe das Fehlersyndrom für folgende Fälle an:

   a) ein Fehler in Stelle x_4,

   b) je ein Fehler in den Stellen x_4 und x_9,

   c) je ein Fehler in den Stellen x_4,x_7 und x_9

6.2 BINÄRER GRUPPENCODE

Ein Gruppen-Code ist durch die folgenden drei Generatorworte definiert:

$$G_1 : 01100$$

$$G_2 : 10010$$

$$G_3 : 10101$$

1. Wie groß ist die Anzahl der Nutz- und Codeworte? Stellen Sie die Liste aller Nutzworte und Codeworte auf!

2. Geben Sie die Gewichtsverteilung $A(w), w = 0,1,...,n$ dieses Codes an!

3. Wie groß ist die Hamming Distanz h?

4. Geben Sie ein Beispiel für ein vom Empfänger nicht als falsch erkennbares Fehlermuster F_n an.

5. Statt der Generatorworte G_1, G_2, G_3 sollen aus der Liste aller Codeworte aus Teilaufgabe 1) drei andere Codeworte als neue Generatorworte G_1*, G_2*, G_3* so ausgewählt werden, dass sich die Nachrichtenstellen in den ersten Stellen der auf dieser Basis konstuierten Codeworte direkt wiederfinden. Wie lauten diese G_1*, G_2*, G_3*?

6.3 Restfehler bei Hamming-Codes

1. Sei p die Bitfehlerwahrscheinlichkeit. Mit welcher Wahrscheinlichkeit wird eine Nachricht aus m Bits korrekt übertragen?

2. Mit welcher Wahrscheinlichkeit tritt genau ein Bitfehler auf?

3. Mit welcher Wahrscheinlichkeit wird ein n-stelliges Codewort eines Hamming-Codes korrekt empfangen oder kann korrekt korrigiert werden?

7 Code und CPU

7.1 Zyklischer Code

Für einen zyklischen Code ist folgendes Generatorpolynom vorgegeben:

$$G(u) = u^4 + u^2 + u + 1 = 10111$$

1. Bestimmen Sie die Periode dieses Generatorpolynoms!

2. Wie groß ist die Zahl m der Nutzbits des entsprechenden zyklischen Codes?

3. Ermitteln Sie die Prüfstellen für folgendes Nutzwort 001!

4. Zerlegen Sie $G(u)$ in seine primitiven Teiler!

5. Um was für einen Code handelt es sich also?

7.2 Taktrate, ISA und CPU-Zeit

Im Folgenden wir der Zusammenhang zwischen Taktrate, ISA und CPU-Zeit vertieft.

1. Ein Rechner A mit einer Taktrate von $2GHz$ benötigt für die Ausführung eines Programms 9 Sekunden. Wie viele Taktzyklen hat er dabei ausgeführt?

2. Rechner B hat für speziell das oben betrachtete Programm einen um 25% höheren CPI_{eff} verglichen mit Rechner A. Welche Taktrate benötigt B, damit er dasselbe Programm mit einer Laufzeit von 6 Sekunden abarbeiten kann.

3. Wie viel mal schneller ist dann Rechner B als Rechner A?

7.3 CPU Instruktion und Laufzeit

Die in einem Programm verwendeten Befehle sind in Tabelle 1 aufgelistet. Die Tabelle enthält die Häufigkeit und die CPI der einzelnen Befehle, welche im Programm vorkommen.

Tabelle 1 Häufigkeiten und CPI von Befehlen eines Programms

Instruktion	Häufigkeit	CPI
ALU	50%	1
Load	20%	5
Store	10%	3
Branch	20%	2

1. Berechnen Sie die effektive CPI für das Programm.

2. Wie viel schneller würde das Programm laufen, wenn ein besser Daten-Cache die durchschnittliche Ladezeit auf 2 CPI reduziert?

3. Wie viel schneller würde das Programm laufen. wenn eine verbesserte Sprungvorhersage die CPI für den Branch-Befehl um eins verringert?

4. Angenommen zwei ALU Instruktionen können gleichzeitig ausgeführt werden. Wie viel schneller würde das Programm laufen?

8 Maschinencode/ Assemblercode

8.1 Lesen von Maschinencode

Wandeln Sie folgendes Stück Maschinencode zurück in Assemblercode:
Welche Funktion implementiert dieser Code?

8.2 Umwandlung von Assemblercode zu Maschinencode

Übersetzen Sie das folgende Listing in Maschinencode für den DLX-Prozessor. Geben Sie jedes Maschinencode-Befehlswort auch in kompakter hexadezimaler Darstellung an.

9 Leaf and Non-Leaf Procedures

1. Übersetzen Sie die iterative Prozedur in den MIPS-Assembler Code, der in der Vorlesung vorgestellt wurde!

2. Wie viele Register benötigen Sie für die beiden unterschiedlichen Umsetzungen?

3. Wie viele Funktionsausrufe werden jeweils benötigt um das 4. Fibonacci-Element zu berechnen?

4. Illustrieren Sie für die rekursiv programmierte Funktion die Stack-Belegung in der tiefsten Rekursionsstufe, die benötigt wird, um das 4.Fibonacci-Element zu berechnen!

10 Übersetzung

10.1 Übersetzungsvorgang

1. Präprozessor Lassen Sie sich den Output des Präprozessors anzeigen, wenn er auf den oben angegeben C-Code angwendet wird.
Was hat sich im Vergleich zum Originalcode geändert?
Welche Aufgaben hat der Präprozessor allgemein, warum ist dieser zusätzliche Schritt beim übersetzen sinnvoll?

2. Erzeugen von Assembler-Code Erzeugen Sie nun aus dem C-Code Assembler Code und analysieren Sie diesen: Welche Konstrukte im Assembler-Code entsprechen i, sum, lower und UPPER?

3. Optimierter Code Erzeugen Sie nun optimierten Assembler Code. Was macht das Assemblerprogramm nun? Erklären Sie dieses Verhalten.

4. Disassemblierung Wie viele Byte Maschinencode werden benötigt um zwei Register zu addieren?

10.2 Inline-Funktionen

1. Übersetzen Sie diese Funktionen nach MIPS! Wie viele Instruktionen werden benötigt?

2. Schreiben Sie die Funktion minOfMax in C-ähnlichen Pseudo-Code um, indem Sie die enthaltenen Funktionsaufrufe als inine Funktionen abbilden!

3. Übersetzen Sie die modifizierte Funktion ebenfalls nach MIPS! Wie viele Instruktionen werden jetzt benötigt?

11 Pipelines

11.1 Pipeline

1. Erweitern Sie das Diagramm um die benötigte Verzögerung!

2. Wie viele Takte lang muss die Pipeline verzögert werden?

3. Welchen Beschleunigung wurde mit der Pipeline für diesen Code erreicht? Gehen Sie davon aus, dass für einen Takt mit Pipelining $200ps$ benötigt werden. Ohne Pipelining dauert ein Takt $800ps$.

11.2 Pipeline Recorder

1. Geben Sie den umgeordneten Quelltext sowie ein Ablaufdiagramm an, aus dem hervorgeht, welcher Befehl in welcher Stufe und in welchem Takt durchläuft.

2. Wie viele Takte lang muss die Pipeline verzögert werden?

3. Welche Beschleunigung wurde mit der Pipeline für diesen Code erreicht? Gehen Sie davon aus, dass für einen Takt mit Pipelining $200ps$ benötigt werden. Ohne Pipelining dauert ein Takt $800ps$.

12 Caches

12.1 Cache-Design

1. Geben Sie für beide Strategien die mittlere Zugriffszeit bei Leseanfragen in Abhängigkeit von Trefferrate r_h, Cache-Zugriffszeit t_C und Arbeitsspeicher-Zugriffszeit t_M an.

2. Wie groß muss die reduzierte Trefferrate r'_h mindestens sein, damit die zweite Strategie besser ist?

3. Ist es sinnvoller, bei einem Direct-Mapped-Cache die höchstwertigen oder niedrigstwertigen Bits der Adresse als Index zu verwenden? Begründen Sie Ihre Antwort!

12.2 Cache

1. Warum werden in Computern Caches verwendet?

2. Was ist der Unterschied zwischen direkt abgebildeten Caches und vollassoziativen Caches? Nennen Sie je einen Vorteil der beiden Varianten!

3. Nennen Sie drei mögliche, realistische Verdrängungsstrategien bei einem voll-assoziativem Cache.

4. Vervollständigen Sie die Tabelle 1 auf der nächsten Seite, die einen direkt abgebildeten Cache mit 4 Blöcken beschreibt.

   a) Geben Sie für jede Anfrage an, ob es zu einem Cache-Hit oder einem Cache-Miss kommt und tragen Sie ggf. neu in den Cache kommende Werte entsprechend ein!

b) Setzen Sie an den entsprechenden Stellen das Dirty Bit und geben Sie an, wenn ein Wert aus dem Cache in den Arbeitsspeicher zurückgeschrieben werden muss!

13 Caches und Speichervirtualisierung

13.1 Caches

1. Wie viele Speicherzellen benötigt dieser Cache insgesamt? Zeichnen Sie die Struktur eines solchen Caches und geben Sie die Breite der enthaltenen Felder in Bits an!

2. Geben Sie die mittlere Zugriffszeit bei Leseanfragen in Abhängigkeit von den beiden Trefferraten r_{h1} und r_{h2}, den jeweiligen Cache-Zugriffszeiten t_{C1} und t_{C2} sowie der Arbeitsspeicher-Zugriffszeit t_{MEM} an!

13.2 Speichervirtualisierung

1. Beschreiben Sie, wie mit Hilfe der Seitentabelle aus der virtuelle Adresse die Adresse im physikalischen Arbeitsspeicher ermittelt wird!

2. Wie viele virtuelle Seiten gibt es im virtuellen Speicher?

3. Wie viele Seiten gibt es im physikalischen Speicher?

4. Wie viel Speicher benötigt die Seitentabelle, wenn in jeder Zeile nur der Index der dazugehörenden physikalischen Seite und keine weiteren Metainformationen gehalten werden?