(8 Punkte) Gegeben ist ein Programm mit Funktionsaufrufen. Die Variable `x` ist global definiert.

(a) Bei jedem Funktionsaufruf wird die Rücksprungadresse (Zeilennummer) am Stack abgelegt. Der Stackpointer (SP) zeigt zu Beginn auf die Adresse 0xFFFF. Ermitteln Sie den Stackpointer und den Stackinhalt nach Ausführung der Funktion `Program Start()`.

Hinweis: Beachten Sie, dass nicht mehr benötigte Stack-Einträge am Stack bleiben und bei nachfolgenden Stack-Zugriffen überschrieben werden. Weiters, dass Variable `x` global definiert ist.

```
    Program Start() {
0:      x = 0;
1:      u();
2:      v();
3:      exit;
    }

    function v() {
4:      if x > 0 then {
5:          x = x-1;
6:          v(x);
        }
7:      return;
    }

    function u() {
8:      x = 3;
9:      return;
    }
```

1 : FF=2
8:
9: pop FF
2: push 3
4 x = 3
5 x = 2
6 : push 7
4 : x = 2
5 : x = 1
6 : push(7)
4 : x = 1
5 : x = 0
6 : push(7)
4:
7: pop(7)
7: pop(7)
7: pop(7)
7. pop(3)

	Stack
0xFFF9	
0xFFFA	
0xFFFB	
0xFFFC	0x7
0xFFFD	0x7
0xFFFE	0x7
0xFFFF	0x3

	Register
SP	0xFFFF

(b) Kreuzen Sie an, ob es sich um wahre oder falsche Aussagen handelt.
(richtig: +1 Punkte, falsch: -1 Punkte, keine Antwort: 0 Punkte)

wahr	falsch	
○	⊗	`If`-Anweisungen können bei Prozessoren mit Pipelining zu Structural Hazards führen.
○	⊗	Bei einer RISC Prozessorarchitektur stehen tendenziell mehr Maschinenbefehle zur Verfügung als bei einer CISC Prozessorarchitektur.
○	⊗	Eine CISC Prozessorarchitektur arbeitet tendenziell mit einer höheren Taktrate als eine RISC Prozessorarchitektur.

3. *(10 Punkte)* Sie arbeiten mit einem Prozessor, der eine fünfstufige Pipeline besitzt: *Instruction Fetch* (IF), *Instruction Decode and Read Registers* (ID), *Execute* (EXE), *Read or Write Memory* (MEM) und *Write Back Registers* (WB). Bedingt durch die Pipelinestruktur kann es zu *RAW Data-Hazards* kommen, welche durch Einfügen von NOPs (No Operations) vermieden werden. Register werden in der ersten Takthälfte beschrieben und in der zweiten Takthälfte gelesen. Auf dem Prozessor wird folgendes Programm ausgeführt:

```
SUB  R4,  R5,  R6     # subtrahiere R6 von R5, Ergebnis in R4
SRL  R0,  R4,  1      # right shift von R4 um eine Stelle, Ergebnis in R0
LW   R1,  Mem[0x69]   # lade Wert aus Speicheradresse 0x69 in R1
SW   R1,  Mem[R4]     # schreibe R1 auf Speicheradresse in R4
DIV  R7,  R8,  R9     # dividiere R8 durch R9, Ergebnis in R7
```

(a) Zeichnen Sie die Belegung der Pipeline für das gegebene Programm unter der Annahme, dass die Pipeline am Beginn und am Ende leer ist. Vergessen Sie nicht an den entsprechenden Stellen genügend NOP Operationen einzufügen.

Zeit ↓	IF	ID	EXE	MEM	WB
1	SUB				
2	NOP	SUB			
3	NOP	NOP	SUB		
4	SRL	NOP	NOP	SUB	
5	LW	SRL	NOP	NOP	SUB
6	NOP	LW	SRL	NOP	NOP
7	NOP	NOP	LW	SRL	NOP
8	SW	NOP	NOP	LW	SRL
9	DIV	SW	NOP	NOP	LW
10	SW	DIV	SW	NOP	NOP
11		SW	DIV	SW	NOP
12			SW	DIV	SUB
13				SW	DIV

(b) Ordnen Sie die Instruktionen so um, dass möglichst wenige NOP Operationen notwendig und alle Abhängigkeiten berücksichtigt sind.

SUB
LW
DIV
SRL
SW

Technische Grundlagen der Informatik			Test 3 23.06.2017 90 Minuten **Gruppe A**
Matrikelnr.	Nachname	Vorname	Unterschrift

Deckblatt sofort ausfüllen und unterschreiben!

Bitte deutlich und nur mit **Kugelschreiber** schreiben. Verwenden Sie keinen Tipp-Ex oder dergleichen. Unleserliche Antworten werden nicht gewertet!

Geben Sie bei Rechenaufgaben den **Lösungsweg** an!

Es sind keine Hilfsmittel zugelassen. Dies inkludiert Bücher, Mitschriften, Ausdrucke von Folien, Smartphones, Taschenrechner etc.

Zusatzblätter werden nicht akzeptiert!

Bei **Ankreuzfragen** werden Minuspunkte auf Teilaufgaben übernommen. Das Minimum je Gesamtaufgabe beträgt 0 Punkte.

1	[10]	[ ]
2	[12]	[ ]
3	[10]	[ ]
4	[8]	[ ]
5	[15]	[ ]
6	[10]	[ ]
7	[8]	[ ]
8	[7]	[ ]
9	[10]	[ ]
10	[10]	[ ]
Summe	[100]	[ ]

---

1. *(10 Punkte)* Beantworten Sie folgende Fragen zu Speicher, Caches und Pipelining.

   (a) Im Zusammenhang mit Pipelining unterteilt man Hazards in *Data Hazards*, *Control Hazards* und *Structural Hazards*. Ordnen Sie jeder der folgenden Maßnahme zur Reduktion von Hazards jene Art von Hazards zu, die damit reduziert werden kann.

   - Erweiterung der Hardware: Structural Hazards
   - Data Forwarding: Data Hazards
   - Branch Prediction: Control Hazards

   (b) Ordnen Sie folgende Speichertechnologien absteigend (langsamster Speicher zuerst) nach ihren Zugriffszeiten: *Solid State Drive (SSD)*, *Dynamic RAM (DRAM)*, *Static RAM (SRAM)*, *Hard Disk Drive (HDD)*

   slowest fastest

   HDD, SSD, DRAM, SRAM

2. *(12 Punkte)* Beantworten Sie folgende Fragen zu Speicherzugriffen und Stacks.

(a) Welche der folgenden Instruktionspaare stellen korrekte Implementierungen der Stack Operationen *push(R)* und *pop(R)* dar? Wenn ein Instruktionspaar *push(R)* und *pop(R)*, in beliebiger Reihenfolge, darstellt, schreiben Sie unter die entsprechende Instruktion *push(R)* bzw. *pop(R)*. Stellt ein Instruktionspaar nicht beides, *push(R)* und *pop(R)*, dar, so streichen Sie das komplette Instruktionspaar durch.

$memory[SP-] \leftarrow R$	$R \leftarrow memory[+SP]$
push(R)	pop(R)
$R \leftarrow memory[SP-]$	$memory[-SP] \leftarrow R$
$R \leftarrow memory[+SP]$	$memory[SP+] \leftarrow R$
$memory[SP+] \leftarrow R$	$R \leftarrow memory[SP-]$

(b) Gegeben sind nachfolgende Instruktionen. Schreiben Sie jeweils rechts neben der Instruktion die verwendete Adressierungsart hin.

Hinweis: In der Vorlesung sind folgende Adressierungsarten betrachtet worden: Direct-Addressing Mode, Immediate Mode, Indirect-Addressing Mode, Register-Indirect Mode und Register Mode.

$R1 \leftarrow R2$	Register-Mode
$R3 \leftarrow -1$	Immediate Mode
$R0 \leftarrow memory[0x500]$	Direct-Addressing Mode
$R3 \leftarrow memory[R4]$	Register-Indirect Mode
$R7 \leftarrow memory[memory[0x666]]$	Indirect Addressing-Mode

(c) Welche der Abkürzungen *LIFO, LILO, FIFO* und *FILO* beschreibt die Funktionsweise eines Stacks und wofür steht die Abkürzung?

LIFO = Last in First out

5. *(15 Punkte)* Ein Prozessor mit einer Adresslänge von 24 Bit und einer Datenwortlänge von 8 Bit hat einen 16 KiB großen *4-way Set-Associative Cache* mit einer Blockgröße von 16 Byte.

$2^4 \cdot 2^{10}$ 2^4

(a) Aus wievielen Cache-Sets besteht dieser Cache?

$2^4 \cdot x = 2^{14} \Rightarrow x = 2^{10}$ $\frac{2^{10}}{2^2} = 2^8$ 2^8 Sets

256 Sets

(b) Berechnen Sie für den gegebenen Cache die Längen von Tag, Offset und Index in Bit.

Tag-Länge: $24 - 12 = 12$ Bit

Index-Länge: 8 Bit

Offset-Länge: $\frac{2^4}{1} = 2^4 \Rightarrow 4$ Bit

(c) Angenommen, das Cache-Set 18 ist vor jedem der unten angegebenen Lesezugriffe folgendermaßen belegt:

Tag	Valid	Dirty
FF0	1	0
003	1	1
7E8	0	0
3FF	1	0

Welche Zugriffe führen zu einem *miss*, welche zu einem *hit*? Kreuzen Sie entsprechend an.
(richtig: +1 Punkte, falsch: -1 Punkte, keine Antwort: 0 Punkte)

Tag | Index

`FF0181`	☒ hit	○ miss
`3EE18C`	○ hit	☒ miss
`003180`	☒ hit	○ miss
`7E8183`	○ hit	☒ miss

(d) Auf welche Strategie bei Schreiboperationen kann man aufgrund der im vorigen Teilbeispiel angegebenen Verwaltungsinformationen schließen?
(richtig: +2 Punkte, falsch: -2 Punkte, keine Antwort: 0 Punkte)

○ Write Through ☒ Copy Back ○ Random Write

(e) Wieviele Adressen des Hauptspeichers sind dem selben Cache-Set zugeordnet?

16 Adressen

128 - 160
32 - = 240

255.255.255.240

CSMA/CA für Halbduplex ist sich ein Lacke

7. *(8 Punkte)* Kreuzen Sie an, ob es sich um wahre oder falsche Aussagen handelt.
(richtig: +2 Punkte, falsch: -2 Punkte, keine Antwort: 0 Punkte; Minimum: 0 Punkte)

wahr	falsch	
○	⊗	Der *Time to live* Eintrag (Hop-Count) im IPv4 IP-Header wird bei jedem Routing um eines erhöht.
○	⊗	Carrier Sense Multiple Access with Collision Detection (CSMA/CD) berechnet korrekte Prüfsummen.
⊗	○	Bei einer Halbduplex-Übertragung können Daten abwechselnd, aber nicht gleichzeitig, in beide Richtungen übertragen werden.
⊗	○	MAC-Spoofing betrifft einen Sicherheitsaspekt am OSI-Layer 2

---

8. *(7 Punkte)* Gegeben ist die folgende IPv6-Adresse in minimaler Notation:

```
fe80::17:a1ed:9f91:bbc/64
```

(a) Geben Sie die vollständige Adresse mit allen führenden Nullen an.

(b) Welche Bedeutung hat fe80: am Anfang der IP?

(c) Ist eine MAC-Adresse in die IPv6 Adresse codiert worden und wie haben Sie dies erkannt?

128 : 160
32 : 240
64
16 : 80

(d) Können alle möglichen IP-Adressen im Subnet für Hosts verwendet werden, und wenn nicht, welche Adressen sind ausgenommen?

Nein da: Netzname und Broadcastadresse

(e) Wie adressieren Sie in IPv6 den Localhost?

128 = 160
32 = 240
64
16 = 80

255.255.255.240

9. *(10 Punkte)* Sie bekommen von Ihrem ISP (Internet Service Provider) ein /28 Netzwerk zugewiesen. Der Default-Gateway befindet sich im zugewiesenen Netzwerk aber physisch beim ISP und hat die niedrigste Host-IP-Adresse im Netzwerk. Ihr Webserver ist ebenfalls im gleichem Netzwerk und hört auf die IP 130.6.4.164.

1111 0000 | 0000 1111 | 255-64-16 | 255 - 80 = 175
1010 0100 | 1010 0100 | 255-80
128 32 160 | 128 32 | 1010 1111

(a) Wie lauten die Netzwerk- und die Broadcast-Adresse des zugewiesenen Netzwerkes?

Netzadr: IPv4SCP: 130.6.4.160

Broadcast: IPv4SCP: 130.6.4.175

(b) Weisen sie einem weiteren Host eine Netzwerkkonfiguration (IP-Adresse, Subnetmask, Gateway) zu, sodass dieser mit dem Internet kommunizieren kann.

128, 130.6.4.162

Gateway: 130.6.4.161

(c) Sie begehen einen Konfigurationsfehler und weisen Ihrem neuen Host aus der Aufgabe (b) ein /27 Netzwerk anstelle eines /28 Netzwerkes zu. Kann dieser Host dennoch mit Ihrem Webserver kommunizieren? Welche Fehlersymptome sind zu erwarten?

ja da Pakete fehlgeleitet werden können

Paket könnte weggeschmissen werden

(d) Angenommen, Sie haben Daten via SCP über TCP, IPv4 und Ethernet übertragen und drei Ethernetpakete mit je 120 Byte erhalten. Wie viele Nutzdaten in Byte wurden versendet, wenn jeder Header 20 Byte groß war und TCP die Paketgröße beschränkte.

Nutzdt: 1 Paket ohne SCP Header: 120-20-20-20 = 60 Byte => 2 Paket 160 Byte Nutzdaten

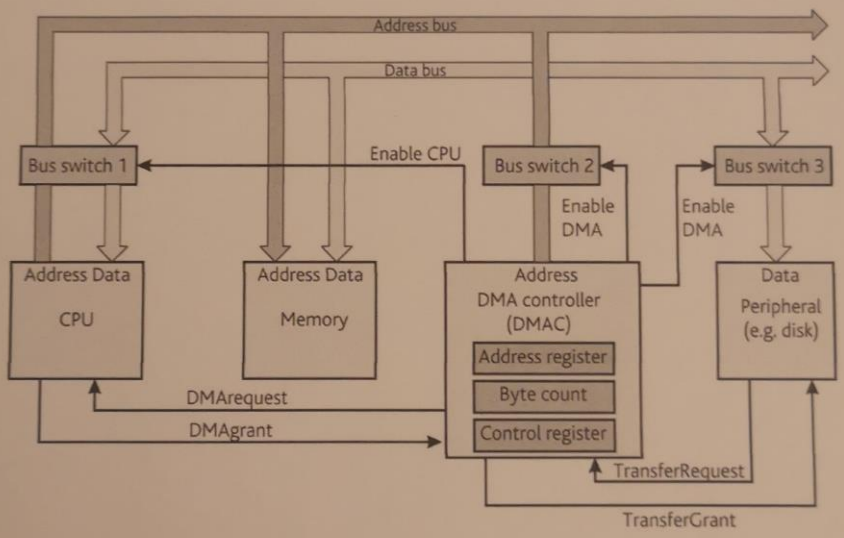
1 Paket mit SCP Header: 120-4-20 = 80 Byte (80) => 3 Pakete 200 Byte Nutzdaten

10. *(10 Punkte)* Peripherie

(a) Kreuzen Sie an, ob es sich um wahre oder falsche Aussagen handelt.
(richtig: +2 Punkte, falsch: -2 Punkte, keine Antwort: 0 Punkte)

wahr	falsch	
○	⊗	Ein Backup wird durch ein Raid obsolete.
⊗	○	SLC Flashspeicher kann schneller als MLC Flashspeicher beschrieben werden.

(b) Ein peripheres Gerät will Daten mittels eines DMA Controllers kopieren. Tragen Sie alle erforderlichen Singale in der richtigen Reihenfolge in die untenstehende Tabelle ein, um dem peripheren Gerät die Datenübertragung zu erlauben. Falls Signale gleichzeitig geschalten werden, schreiben Sie diese in einer Zeile.



Reihenfolge	Signal
1	TransferRequest
2	DMA Request
3	DMA Grant
4	Enable CPU / DMA
5	Transfer Grant

(c) USB 3.1 (SuperSpeed+) verwendet zur Leitungscodierung 128b132b. Wie groß ist der Overhead in MB wenn 128 GB an Nutzdaten via USB 3.1 kopiert werden? Geben Sie den Lösungsweg an.

(10 Punkte) Schreiben Sie ein Micro16-Programm, das den Inhalt von Register R0 um die in Register R1 angegebene Stellenzahl n nach links rotiert. Das modifizierte Datenwort soll am Ende wieder im Register R0 gespeichert werden. Gehen Sie davon aus, dass die im Register R1 gespeicherte Zahl eine positive ganze Zahl darstellt.

Beispiele:
R0 = 11001100 10010101, R1 = 00000000 0000000 R0 = 11001100 10010101
R0 = 11001100 10010101, R1 = 00000000 0000001 R0 = 10011001 00101011
R0 = 11001100 10010101, R1 = 00000000 0000010 R0 = 00110010 01010111
R0 = 11001100 10010101, R1 = 00000000 0000011 R0 = 01100100 10101110
R0 = 11001100 10010101, R1 = 00000000 0000100 R0 = 11001001 01011100

- Micro-Code:

```
:loop
(R1); if Z goto. end
R1 ← R1 + -1
(R0); if N goto, eins
R0 ← Sh(R0)
goto. loop
:eins
R2 ← 1
R0 ← Sh(R0)
R0 ← R0 + R2
goto. loop
:end
```

(c) Wodurch unterscheidet sich ein Computer mit einer *Von-Neumann-Architektur* von einem Computer mit einer *Harvard-Achitektur*?

Harvard getrennt Programm und Daten speichern

(d) Gegeben sei ein Prozessor der Caching verwendet. Die Speicheradressen sind wie folgt strukturiert: 0xTTIIOO. Jedes T, I und O ist jeweils Platzhalter für eine volle hexadezimale Ziffer. T steht für *Tag*, I für *Index* und O für *Offset*. Die Adressierung erfolgt Byte-weise und die Größe des Caches beträgt 64 KiB.

i. Wieviele Sets hat dieser Cache?

$2^8 = 256$ Byte

ii. Mit welcher Blockgröße arbeitet der Cache?

$2^8 \cdot 2^0 = x \Rightarrow 2^8$ Byte $= 256$ Byte

iii. Um welche Art von Cache handelt es sich (direct-mapped, N-way-set-associative, fully-associative)? Im Falle eines N-way-set-associative Caches geben Sie ein gültiges N an.

$64 \text{ KiB} = 2^{16}$

$2^{16} = 2^8 \cdot x \cdot 2^8 \Rightarrow x = 2^0 \Rightarrow$ Direct-mapped