Aufgabe II-1: Fragen zur Rechnerarchitektur (12 P)

a) Geben Sie an, ob die folgenden Aussagen wahr (W) oder falsch (F) sind:

W F

◯ ◯ 1) Pipelining ist eine Mikroarchitekturtechnik.

◯ ◯ 2) Superskalarität ist eine Architekturtechnik.

◯ ◯ 3) Die Adresse des Speicherwortes zeigt bei einem wortadressierbaren Speicher mit Big-Endian-Anordnung immer auf das höchstwertige Byte.

◯ ◯ 4) Bei dem Befehl *Shift Logical Right* (SLR) wird als oberstes Bit immer eine Null eingefügt.

◯ ◯ 5) Die effektive Adresse ergibt sich bei der indizierten Adressierung als Summe von Indexregister und einem im Befehlswort stehenden Offset.

◯ ◯ 6) Der Sprungzieladress-Cache ist ein kleiner Cache-Speicher, auf den in der ID-Stufe zugegriffen wird und der mehrere Tupel bestehend aus Sprungadresse, Sprungzieladresse und Vorhersagebits enthält.

◯ ◯ 7) Eine typische Eigenschaft eines CISC-Prozessors ist die hohe Anzahl an universell nutzbaren Registern.

◯ ◯ 8) In der Operandenbereitstellungsphase werden der *Arithmetic Logic Unit* (ALU) die Operanden aus dem Daten-Cache bereitgestellt.

◯ ◯ 9) Durch Forwarding können Datenkonflikte aufgelöst werden, indem spätere (und jeweils unabhängige) Befehle im Programmverlauf vorgezogen werden.

◯ ◯ 10) Bei der Block-Interleaving-Technik werden die Befehle *eines* Kontrollpfades so lange wie möglich direkt hintereinander ausgeführt.

b) Geben Sie an, wie die **Latenz** und der **Durchsatz** einer Pipeline definiert sind.

c) Im Kurstext haben Sie die drei Befehlsformate (R-Typ, I-Typ und J-Typ) des DLX- bzw. MIPS-Prozessors kennengelernt. Geben Sie für jeden der drei Typen zwei Befehle an, welche diesem Typ zugeordnet werden können. Stellen Sie außerdem für jeden Typen dar, in welche Felder ein 32-Bit-Befehl unterteilt wird und geben Sie deren Bezeichner an. (Diese Aufgabe muss je nach belegter Kursversion **entweder** für den DLX- **oder** den MIPS-Prozessor gelöst werden!)

d) Geben Sie an, was man unter den Prinzipien der **örtlichen Lokalität** und **zeitlichen Lokalität** versteht.

Lösungsvorschläge

a) Die Aussagen 1, 3, 4 und 10 sind wahr.

b) Die Latenz einer Pipeline ist definiert als die Anzahl an Takten, die ein Befehl benötigt, um alle Stufen einer Pipeline zu durchlaufen.

Der Durchsatz einer Pipeline ist definiert als die Anzahl der Befehle, welche die Pipeline in einem Takt verlassen können.

c) siehe Kap. 1.3.2 (alte Kursversion) und Kap. 1.4 (neue Kursversion)

d) zeitliche Lokalität: ein Zugriff auf gleiche Daten bzw. gleichen Programmcode kommt häufig vor. Die Wahrscheinlichkeit, dass auf bereits zugegriffene Daten/-Programmcode erneut zugegriffen wird, steigt daher an.

örtliche Lokalität: erfolgt ein Zugriff auf eine bestimmte Adresse, so ist es wahrscheinlich, dass auch auf benachbarte bzw. nachfolgende Adressen in Zukunft zugegriffen wird.

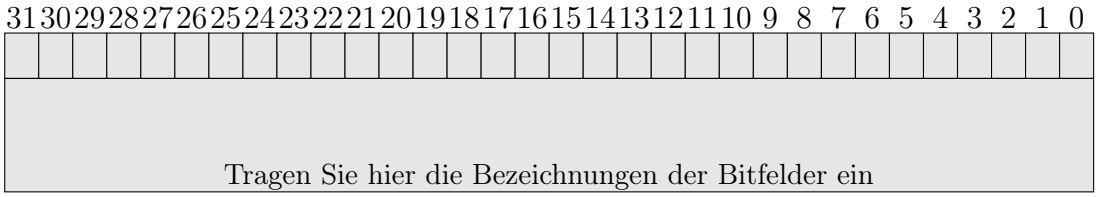
Aufgabe II-2: Gleitkommadarstellung (6 P)

Gegeben sei die Dezimalzahl $Z_{10} = -40{,}375$.

a) Stellen Sie die Zahl Z_{10} als gebrochene, normalisierte Zahl Z_{32} im 32-bit-Format des IEEE-754-Standards dar und tragen Sie dazu die entsprechenden Werte für Vorzeichen, verschobenen Exponenten und Mantisse in das folgende Schema ein:

$Z_{32} = (-1)^{\cdots\cdots} \cdot 2^{(\cdots\cdots)_{10}} \cdot (\ldots\ldots \bullet \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots)_2$

b) Tragen Sie die Zahl Z_{32} in binärer Darstellung in den folgenden Bitrahmen ein und kennzeichnen sowie bezeichnen Sie die unterscheidbaren Bitfelder.



c) Geben Sie die Zahl Z_{32} als Hexadezimalzahl Z_{16} an.

$Z_{16} = \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$

Hinweis:
Die Indizes $\cdots_2$, $\cdots_{10}$, $\cdots_{16}$ und $\cdots_{32}$ kennzeichnen jeweils Zahlen im Binär-, Dezimal- sowie Hexadezimal-System bzw. im 32-Bit-IEEE-Format.

Lösungsvorschläge

a) $Z_{32} = (-1)^1 \cdot 2^{(5)_{10}} \cdot (1 \bullet 010000110..0)_2$

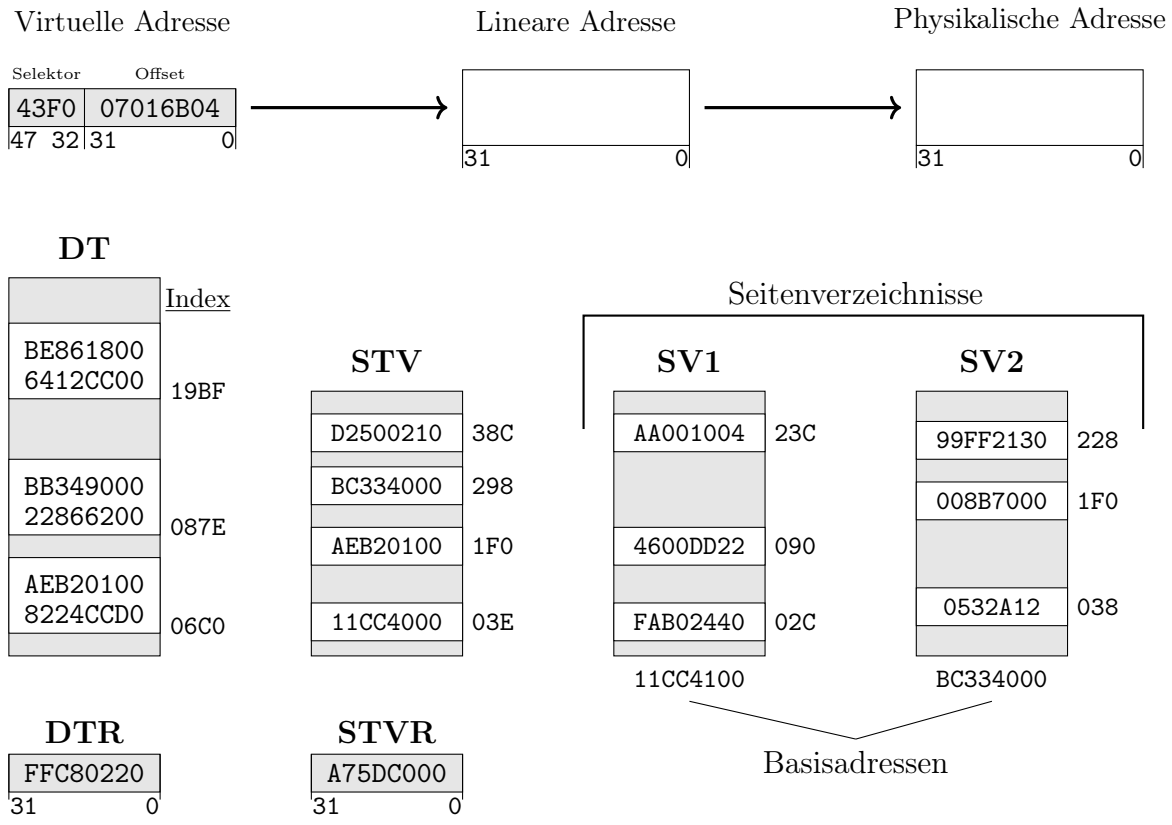
b) Die Bezeichnungen sind s: Vorzeichen, e: versch. Exponent, m: Mantissenteil.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
s	e								m																						

c) $Z_{16} = \text{C2218000}_{16}$

Aufgabe II-3: Virtuelle Speicherverwaltung (20 P)

In dieser Aufgabe wird die virtuelle Speicherverwaltung der x86-Prozessoren betrachtet. Es werden dabei sowohl eine Segmentierung als auch ein Seitenwechselverfahren angewendet. Die Adressbreite beträgt 32 Bit und die Seitengröße 4 kByte. Die unten aufgeführte Abbildung zeigt für einen aktiven Prozess den relevanten Ausschnitt der Speicherbelegung mit Deskriptortabelle (DT), Seitentabellenverzeichnis (STV) und zwei Seitentabellen (SV1 und SV2). Außerdem sind die Belegungen vom Deskriptortabellen-Basisregister (DTR) sowie vom Seitentabellenverzeichnis-Basisregister (STVR) angegeben. Alle Werte der Abbildung sind bis auf die Bitgrenzen im Hexadezimalformat angegeben.



a) Die oberen 13 Bit des Selektors der virtuellen Adresse werden als Index verwendet, um einen der 8 Byte großen Einträge in der Deskriptortabelle zu selektieren. Geben Sie an, wie viele Einträge die Deskriptortabelle insgesamt enthalten kann und wie groß die Tabelle (in Byte oder kByte) maximal wird.

b) Die unteren 4 Byte des selektierten Deskriptors ergeben addiert mit dem Offset der virtuellen Adresse die lineare Adresse. Berechnen Sie diese und tragen Sie sie in das oben stehende freie Feld ein.

c) In der Abbildung sind jeweils die unteren 12 Bit der Adressen sowohl für die STV als auch die SV1/SV2 angegeben. Für den Zugriff auf das STV werden die oberen 20 Bit des STVR mit den oberen 10 Bit der linearen Adresse (31..22) sowie zwei Nullen (00_b) konkateniert. Der selektierte Eintrag in dem STV enthält die Basisadresse des auszuwählenden Seitenverzeichnisses. Die nächsten 10 Bit der linearen Adresse (21..12) werden ebenfalls mit zwei Nullen (00_b) konkateniert und zur Selektion eines Eintrags im soeben ausgewählten Seitenverzeichnis verwendet. Die physikalische Adresse setzt sich dann aus den oberen 20 Bit des Eintrags im Seitenverzeichnis konkateniert mit den unteren 12 Bit der linearen Adresse zusammen. Bestimmen Sie auf diese Weise die physikalische Adresse und tragen Sie diese in das entsprechende Feld ein.

d) Erläutern Sie, wie das Seitenwechselverfahren durch ein im Kurs genanntes Vorgehen beschleunigt werden kann. Welche zusätzliche Funktionseinheit wird hierfür benötigt und was speichert diese ab?

e) Bestimmen Sie die Anfangsadressen für folgende Tabelleneinträge (Nummerierung hier im Dezimalsystem!):

Eintrag Nr. 12_{10} in der DT .....

Eintrag Nr. 64_{10} im STV .....

Eintrag Nr. 16_{10} im SV1 .....

Lösungsvorschläge

a) Max. Anzahl: 13 Bit des Selektors zur Auswahl → 8192 mögliche Deskriptoren.
Max. Größe: 8 Byte/Deskriptor · 8192 Deskriptoren = 64 KB oder 65536 Byte.

b) Der Index in die DT beträgt `087E`.
Die lineare Adresse ergibt sich zu `22866200 + 07016B04 = 2987CD04`

c) Die oberen 10 Bit der virtuellen Adresse konkateniert durch zwei Nullen lauten `298`. Die Basisadresse des Seitenverzeichnisses lautet daher `BC334000`, was der Basisadresse von SV2 entspricht. Die nächsten 10 Bit der linearen Adresse ergänzt durch zwei Nullen lauten `1F0` und verweisen konkateniert mit der Basisadresse von SV2 auf den Eintrag `008B7000`. Die physikalische Adresse setzt sich dann durch die oberen 20 Bit des Eintrags sowie die unteren 12 Bit der linearen Adresse zusammen zu `008B7D04`.

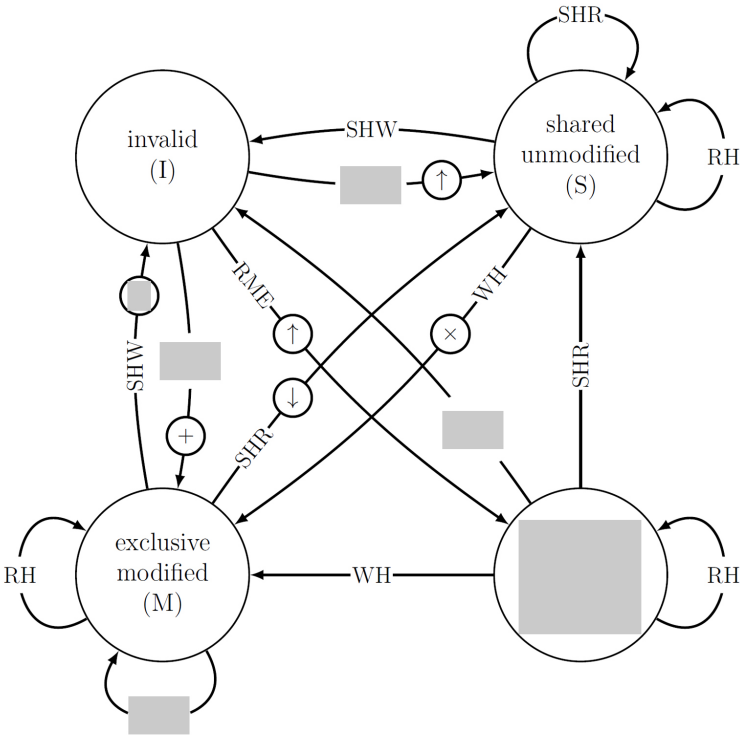
d) Das Seitenwechselverfahren kann durch einen vollassoziativen Cache-Speicher (Translation Lookaside Buffer, TLB) beschleunigt werden. Dieser speichert als Tag die oberen 20 Bit der virtuellen Adresse und als Inhalt die oberen 20 Bit der physikalischen Adresse. Solange ein Cache-Hit vorliegt, kann der zeitaufwendige mehrfache Speicherzugriff bei der Seitenverwaltung dann entfallen.

e) Die Basisadresse der DT lautet `FFC80220` und die Einträge sind jeweils 8 Byte lang. Der 12. Eintrag befindet sich daher an der Anfangsadresse `FFC80220 + B*8 = FFC80278`.
Die Basisadresse des STV lautet `A75DC000` und die Einträge sind jeweils 4 Byte lang. Der 64. Eintrag befindet sich daher an der Anfangsadresse `A75DC000 + 3C*4 = A75DC0F0`.
Die Basisadresse des SV1 lautet `11CC4100` und die Einträge sind jeweils 4 Byte lang. Der 16. Eintrag befindet sich daher an der Anfangsadresse `11CC4100 + F*4 = 11CC413C`.

Aufgabe II-4: Cache-Kohärenz-Protokoll (12 P)

a) Geben Sie an, was man unter den Begriffen **Konsistenz** und **Kohärenz** in Bezug auf die Speicherhierarchie versteht.

b) Ergänzen Sie im unten stehenden MESI-Zustands-Diagramm die sieben leeren und grau hinterlegten Felder.



RH	read hit		veränderte Cache-Line zurückschreiben
RMS	read miss, shared		
RME	read miss, exclusive		
WH	write hit		lesen einer Cache-Line zum Ändern
WM	write miss		
SHR	snoop hit on read		Cache-Line laden
SHW	snoop hit on write *oder* lesen einer Cache-Line zum Ändern		

c) Es sei ein System mit zwei Prozessoren gegeben. Jeder der beiden Prozessoren verfügt dabei über einen eigenen, in Von-Neumann-Architektur realisierten L1-Cache. Weitere Cache-Ebenen existieren nicht. Beide Prozessoren teilen sich einen Hauptspeicher und sind an diesen über einen gemeinsam genutzten Speicher-Bus angeschlossen. Zur Gewährleistung der Cache-Kohärenz wird das MESI-Protokoll eingesetzt. Jeder der L1-Caches besitzt vier Cache-Blöcke und ist als vollassoziativer Cache-Speicher mit LRU-Ersetzungsstrategie realisiert. Bei mehreren freien oder ungültigen Cache-Blöcken werden diese von unten an gefüllt. Zur Vereinfachung wird angenommen, dass sowohl die Cache-Blöcke als auch die Cache-Lines durchnummeriert sind. Die genaue Adressaufteilung in Tag/Index/Byteauswahl wird dabei in dieser Aufgabe nicht betrachtet.

Es finden nacheinander folgende Speicherzugriffe statt:

Prozessor-Nr.	Cache-Line	Lesen/Schreiben
1	1	Schreiben
1	4	Lesen
2	5	Schreiben
2	2	Lesen
1	2	Lesen
1	5	Schreiben
2	4	Schreiben

Geben Sie in der folgenden Tabelle den zeitlichen Verlauf der Cache-Inhalte an, der sich durch die oben angegebenen Speicherzugriffe sowie die Verwendung des MESI-Protokolls ergibt. Hierfür sollen in den Einträgen sowohl die Nummer der aktuell enthaltenen Cache-Line als auch die Abkürzung des Zustandes entsprechend des MESI-Protokolls eingetragen werden. Die initialen Belegungen und die Belegung nach dem ersten Zugriff sind als Beispiel vorgegeben.

Prozessor 1				Prozessor 2			
Block 1	Block 2	Block 3	Block 4	Block 1	Block 2	Block 3	Block 4
E/4	E/6	S/5	I/-	E/1	S/5	I/-	I/-
E/4	E/6	S/5	M/1	I/-	S/5	I/-	I/-

ENDE

Lösungsvorschläge

a) Ein System heißt konsistent, wenn alle Kopien eines Speicherwortes in der Speicherhierarchie stets identisch sind. Ein System heißt cache-kohärent, wenn ein Lesezugriff immer den Wert des letzten Schreibzugriffs auf das Speicherwort liefert.

b) siehe Skript.

c) Lösung:

E/4	E/6	S/5	M/1	I/-	S/5	I/-	I/-
E/4	E/6	I/-	M/1	I/-	M/5	I/-	I/-
E/4	E/6	I/-	M/1	E/2	M/5	I/-	I/-
E/4	E/6	S/2	M/1	S/2	M/5	I/-	I/-
E/4	M/5	S/2	M/1	S/2	I/-	I/-	I/-
I/-	M/5	S/2	M/1	S/2	M/4	I/-	I/-