☐9 ☐9 ☐9 ☐9 ☐9 ☐9

Uvažujte datovou cache s následujícími parametry – velikost cache: 2048B, délka bloku: 64B, stupeň asociativity: 1. Cache je na začátku prázdná. Délka slova je 32 bitů. Předpokládejte, že program čte sekvenčně v jednom cyklu z paměti počínaje od adresy 0x4E4 právě 708 slov. Počet opakování cyklu je 10, takže procesor čte celkem 708×10 slov. Souběh jiných programů nepředpokládáme.

Question 1 (+1b) Kolik má výše uvedená cache setů?

1/1

128 | 0 | 2 | 256 | 45 | 10
1 | 512 | 16 | 4 | 32 | 30
42 | 20 | 64 | 1024 | 8 | 25

Question 2 (+3b) Uveďte číslo setu, do kterého se namapují data z adresy **0x4E4**. Pozor, první set má číslo (index) nula.

3/3

17 | 5 | 24 | 31 | 1252 | 11
409 | 27 | 19 | 300 | 30 | 39
31 | 21 | 88 | 1 | 13 | 42

Question 3 (+1b) Jaký bude počet **hitů** v této datové cache **při prvním průchodu cyklem**?

1/1

105 | 514 | 165 | 613 | 204 | 11
941 | 608 | 663 | 708 | 158 | 315
84 | 203 | 562 | 511 | 40 | 586

Question 4 (+1b) Jaký bude počet **missů** v této datové cache **při prvním průchodu cyklem**?

1/1

146 | 0 | 194 | 543 | 504 | 603
668 | 624 | 505 | 197 | 393 | -233
100 | 697 | 95 | 45 | 550 | 122

Question 5 (+1b) Jaký bude **celkový počet hitů** v této datové cache?

1/1

1334 | 684 | 165 | 6676 | 1142 | 5505
887 | 6801 | 7059 | 2045 | 1261 | 3793
5724 | 5936 | 2701 | 4809 | 4739 | 218

Question 6 (+1b) Jaký bude **celkový počet missů** v této datové cache?

1/1

3287 | 5938 | 6396 | 2271 | 1575 | 1356
279 | 404 | 1144 | 4379 | 21 | 5035
6193 | 6862 | 6915 | 2341 | 5819 | 5746

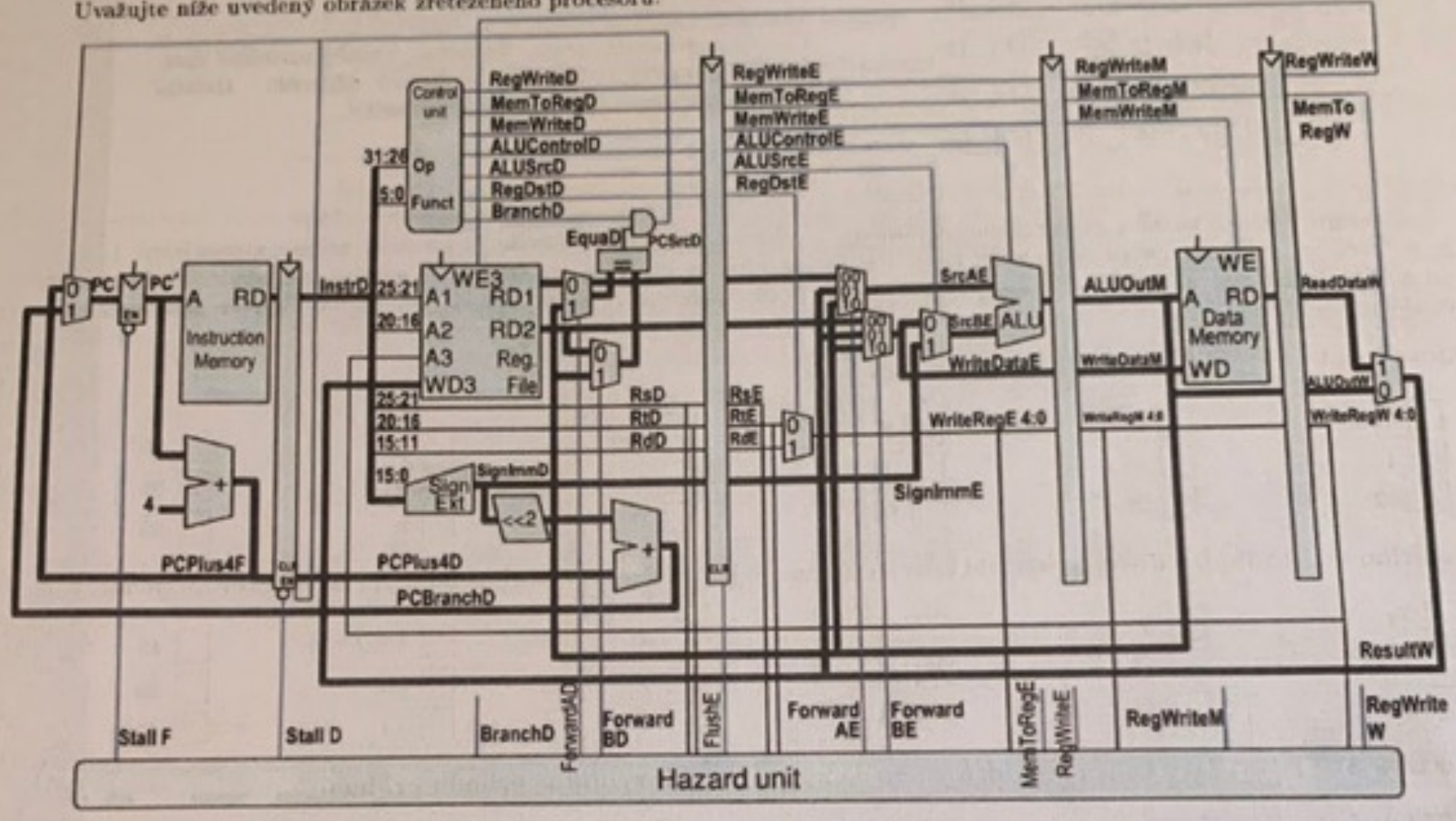
Question 7 (±1b) Obvyklou vlastností architektur CISC je, že obsahují instrukce pro snadnou práci s obsahem paměti, jako například instrukce, která přímo modifikuje obsah paměti o libovolnou konstantu (omezení rozsahem datového typu) na zadané adrese.

-1/1 Ano Ne

Question 8 (±1b) RISC procesory používají tzv. Load/Store model, v němž nelze v jedné instrukci kombinovat paměťové a aritmetické (ALU) operace – což neplatí pro procesory CISC.

0/1 Ano Ne

Uvažujte níže uvedený obrázek zřetězeného procesoru.



Označte instrukce, které procesor **nebude** schopen vykonat správně (za všech okolností a ve všech možných posloupnostech s libovolnými dalšími instrukcemi) v těchto případech:

Question 9 ♣ (+2b) Nejnižší bit signálu WriteDataE je přerušený/zablokovaný na hodnotě „0".

2/2

- beq $s, $t, C
- sw $t, C($s)
- sub $d, $s, $t
- add $d, $s, $t
- addi $t, $s, C
- lw $t, C($s)
- *None of these answers are correct.*

Question 10 ♣ (+2b) Signál ALUSrcE je přerušený/zablokovaný na hodnotě „0".

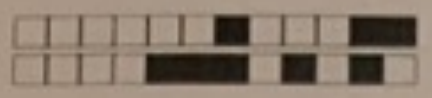
0/2

- sw $t, C($s)
- addi $t, $s, C
- beq $s, $t, C
- sub $d, $s, $t
- lw $t, C($s)
- add $d, $s, $t
- *None of these answers are correct.*

Question 11 ♣ (+2b) Nyní uvažujte případ, kdy v důsledku chyby v procesoru řídicí signály z Hazard unit ForwardAE jsou trvale nastaveny na 00. Označte ty sekvence instrukcí (fragment programu), které nebudou vykonány správně! Předpokládejte, že pipeline je před vykonáním sekvence vyprázdněna.

0/2

- sub s0,s1,s2
add s3,s4,s0
lw s1,4(s2)
- sw s0,4(s1)
sub s2,s0,s2
beq $0,s2,n1
- add s4,s0,s1
sub s0,s1,s0
add s1,s1,s3
- add s2,s1,s2
sub s4,s0,s1
beq s2,s2,n2
- *None of these answers are correct.*



Question 12
(+1b) Jaká je průměrná propustnost výše uvedeného procesoru v MIPS (Milion Instructions Per Second), pokud v důsledku mezi-instrukčních závislostí je v průměru na každých 5 užitečných instrukcí zavedena jedna bublina (bubble – jako důsledek stall, flush)? Víme, že latence jednotlivých stupňů (kombinační zpoždění mezistupňové logiky) procesoru je následující IF: 5ns, ID: 9ns, EX: 25ns, MEM: 9ns, WB: 1ns a frekvence procesoru je nastavena na maximální možnou. Zpoždění v důsledku čekání na ustálení výstupních hodnot mezistupňových registrů (clock-to-Q propagation delay: t_{pcq}), předstih před hodinovým signálem (setup time: t_{setup}), nedokonalosti rozvodu hodinového signálu (t_{skew}) a úvodní plnění pipeline zanedbejte.

1/1

31	20	38	28	15	18
33	2	55	0	30	40
17	17	25	25	10	40

Předpokládejte následující parametry paměťového subsystému procesoru: procesor je 8-bitový, umožňuje adresaci jednotlivých byte, organizace paměti je **little-endian**, velikost stránky spravované jednotkou správy paměti je 16byte a jedna položka 1-úrovňové tabulky pro překlad virtuální adresy zabírá právě 1byte. Přitom předpokládejte, že velikost stránkovací tabulky odpovídá právě jedné stránce. Výpis obsahu fyzického adresního prostoru/paměti:

0x0007

	x0:	x1:	x2:	x3:	x4:	x5:	x6:	x7:	x8:	x9:	xA:	xB:	xC:	xD:	xE:	xF:
00:	6C	77	FD	68	C2	79	91	EA	BF	3E	78	4C	BE	C3	49	65
10:	F3	79	3B	C1	00	B7	D5	1A	3C	67	58	74	D4	92	65	40
20:	DF	E7	62	C1	00	DE	5A	91	66	7F	6B	00	7C	5B	A2	DE
30:	A3	B4	6B	16	E9	B4	30	83	94	33	9C	E0	CE	98	53	00
40:	80	00	26	15	64	00	93	47	38	3E	B9	F6	DB	D1	FB	C4
50:	78	6E	B2	4D	A7	76	3F	4A	2B	65	2A	31	B9	00	99	00
60:	EA	57	CE	94	7A	B5	90	00	13	30	C6	00	00	C7	12	23
70:	75	FC	7F	E0	47	A9	6D	3A	EE	85	00	C6	89	73	8F	88
80:	92	3A	2E	E2	00	00	61	B7	8B	A1	B5	34	BE	A5	83	E9
90:	1D	37	1D	40	F9	4C	4B	1E	F1	75	31	BA	93	19	C4	45
A0:	54	87	38	00	A6	57	11	89	6C	DE	8F	00	48	58	5D	88
B0:	18	E5	F4	D8	00	5C	DA	00	00	73	A6	57	7B	00	F2	23

Nejvyšší úroveň stránkovací tabulky (hodnota PDBR – page directory base register) začíná na adrese **0x0080**. Hodnota položky stránkovací tabulky vyjma nejnižšího bitu přímo určuje adresu. Nejnižší bit položky stránkovací tabulky určuje, jestli je položka platná (0-platná, 1-neplatná). Procesor čte z virtuální adresy **0x0047**.

Question 13 (+0.5b) Kolik bitů virtuální adresy (včetně offsetu ve stránce) je jednotka správy paměti s výše uvedenými parametry schopná přeložit?

0.5/0.5

14	0	7	5	16	10
15	4	9	12	11	3
8	13	2	1	nevím	6

Question 14 (+1.5b) Jaký je offset v rámci fyzické stránky? (Odpovídejte dekadicky.)

1.5/1.5

21	7	25	47	6	58
2	0	44	1	10	17
9	1	48	1	36	20

Question 15 (+1.5b) Jaký je index v tabulce stránek? (Odpovídejte dekadicky.)

1.5/1.5

0	15	57	0	28	17
19	7	20	57	55	3
4	0	48	18	29	14

Question 16 (+2.5b) Na které fyzické adrese jsou uložena data z virtuální adresy 0x0047? (**Pokyny pro vyplnění:** Box zakřížkujte pouze pokud má být daný bit nastaven na 1. Prázdný box je považován za 0. Nejméně významný bit je úplně vpravo. Odpovídejte binárně.)

2.5/2.5

1	1	1	1	1	1	1	1	

Question 17 (+0.5b) Kolik úrovní by bylo potřeba pro stránkovací tabulku, aby byl pokrytý kompletní překlad z 8-bitové virtuální adresy na 8-bitovou fyzickou adresu? Uvádějte absolutní hodnotu počtu potřebných úrovní.

0.5/0.5

- [ ] 2
- [ ] nerealizovatelné
- [ ] 8
- [ ] 3
- [ ] 4
- [ ] 0
- [ ] 6
- [ ] 5
- [ ] 7
- [ ] 1

Question 18 (+0.5b) Pokud uvažujeme obvykle používané stránkovací tabulky, který z následujících bitů NENÍ součástí položky stránkovací tabulky?

0.5/0.5

- [ ] User/Supervisor bit: uživatelský přístup/pouze OS
- [ ] Accessed: zda jsme v nedávné době četli/zapsali
- [ ] Present bit: zda je stránka v paměti, nebo na disku
- [ ] Read/Write bit: určuje zda je stránka pouze ke čtení
- [ ] Alloc: zda se ve stránce nachází dynamicky alokovaná data (např. pomocí malloc)
- [ ] Write-through/Write-back: zapisovací strategie pro stránku
- [ ] Cache disabled/enabled: zda procesor může cachovat

Zcela jistě znáte způsob reprezentace čísel v pohyblivé řádové čárce v jednoduché přesnosti dle IEEE745. Nicméně Vaše aplikace vyžaduje, aby číslo v plovoucí řádové čárce zabíralo pouze 10 bitů. Proto jste navrhli reprezentaci, ve které má exponent 4 bity, mantisa 5 bitů. Jinak chcete zachovat všechny vlastnosti IEEE754.

Question 19 (+2b) Jak se binárně zakóduje dekadické číslo 5.71 v této reprezentaci? **(Pokyny pro vyplnění:** Box zakřížkujte pouze pokud má být daný bit nastaven na 1. Prázdný box je považován za 0. Nejméně významný bit je úplně vpravo. I pokud vychází zbytek větší než 0.5 LSB tak mantisu nahoru **ne**zaokrouhlujte.)

2/2

- [ ] 1 [ ] 1 [ ] 1 [ ] 1 [ ] 1 [ ] 1 [ ] 1 [ ] 1 [ ] 1 [ ] 1

Question 20 (+0.5b) Jak bude v této reprezentaci binárně zakódováno největší možné zobrazitelné reálné číslo?

0.5/0.5

- [ ] 1 [ ] 1 [ ] 1 [ ] 1 [ ] 1 [ ] 1 [ ] 1 [ ] 1 [ ] 1 [ ] 1

Question 21 (+0.5b) Jaká bude dekadická hodnota tohoto největšího možného čísla? Uvažujte 3 desetinné pozice.

0.5/0.5

- [ ] 16.375
- [ ] 43.000
- [ ] 109.000
- [ ] 73.250
- [ ] 179.500
- [ ] 59.375
- [ ] 24.875
- [ ] 81.875
- [ ] 90.500
- [ ] 286.875
- [ ] 252.000
- [ ] 2.750

Question 22 (+0.5b) Jak bude v této reprezentaci zakódováno nejmenší nenulové kladné normalizované číslo?

0.5/0.5

- [ ] 1 [ ] 1 [ ] 1 [ ] 1 [ ] 1 [ ] 1 [ ] 1 [ ] 1 [ ] 1 [ ] 1

Question 23 (+0.5b) Jaká bude dekadická hodnota tohoto nejmenšího nenulového kladného normalizovaného čísla? Zaokrouhlete na 3 desetinné pozice.

0.5/0.5

- [ ] 0.008
- [ ] 0.105
- [ ] 0.035
- [ ] 0.016
- [ ] 0.148
- [ ] 0.004
- [ ] 1.125
- [ ] 0.266
- [ ] 0.031
- [ ] 0.047
- [ ] 0.094
- [ ] 0.000

Question 24

(+2b) Pokud uvažujeme dekadické číslo -9 a aditivní konstantu 31, jaká bude 6-bitová binární reprezentace takto kódovaného čísla v kódu s posunutou nulou?

0/2

- [ ] 110111
- [ ] 010110
- [ ] 011010
- [ ] 110010
- [ ] 101100
- [ ] 111101
- [ ] 001100
- [ ] 110001
- [ ] 011101
- [ ] 101000
- [ ] 111111
- [ ] 110000
- [ ] 000110
- [ ] 100001
- [ ] 011100