Question 1 (±1b) Zapnutím stránkování se zrychlí přístup do paměti.

Ne Ano

Question 2 (±1b) Klasická paralelní PCI sběrnice dovoluje full-duplex přenos dat.

Ano Ne

Question 3 (±1b) V případě komunikace s periferi za použití přerušení (interrupt-driven) vnitřní logika procesoru je periodicky sleduje stavový bit/registr dané periférie a v případě potřeby zareaguje (vyvolá přerušení), což je v protikladu metodou tzv. "polling-u", kdy program v procesoru neustále ve smyčce sleduje dané zařízení a reaguje na jeho stav.

Ne Ano

Question 4 (±1b) PCI-e sběrnice je sériová.

Ano Ne

Uvažujte 5-stupňový zřetězený procesor s možností přeposílání mezivýsledků (forwarding) uvnitř procesoru. Určete, v kterém cyklu **poslední** instrukce níže uvedeného fragmentu kódu začne vykonávání (IF), a ve kterém cyklu bude tato instrukce dokončena (WB). Předpokládejte, že první instrukce projde přes IF, ID, EX, M a WB v cyklech 1, 2, 3, 4 a 5 (v tabulce jsou naznačeno). Do tabulky si můžete zakreslit plnění pipeline i pro zbývající instrukce.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
sub s4,s1,t1	IF	ID	EX	M	WB													
lw t2,0x4(s4)		IF	ID	EX	M	WB												
addi s2,t1,0x3			IF	ID	EX	M	WB											
add s4,s4,0x4				IF	ID	EX	M	WB										
sw s2,0(s4)					IF	ID	EX	M	WB									

Question 5 (+1.5b) Ve kterém cyklu bude přinesena/načtena (IF) poslední instrukce z tabulky?

7 5 16 11 9 12

6 14 8 15 13 10

Question 6 (+1.5b) Ve kterém cyklu bude poslední instrukce z tabulky procházet stupněm WB?

13 15 17 12 8 19

16 10 14 18 11 9

Adresní prostor procesu (32bit Linux) je rozdělen na několik částí, mezi kterými jsou: Text segment (elf), Data segment, BSS segment, Heap, Stack a další. Napište, co v těchto vyjmenovaných částech můžeme nalézt. Do které části (nebo některé z uvedených) budou ukládána data alokovaná pomocí operátoru „new" a pomocí funkce „malloc()"?

Globální proměnné
Statický alokovaná data
Dynamicky linkované knihovny
Vyhrazeno pro OS

Question 8 (+0.4b) Co se ukládá do Data segment?

Statický alokovaná data
Program
Vyhrazeno pro OS
Dynamicky linkované knihovny

Question 9 (+0.4b) Co se ukládá do BSS segment?

Lokální proměnné a data funkcí
Překlady virtuálních adres
Inicializované globální nebo statické proměnné
Dynamicky alokovaná data

Question 10 (+0.4b) Kam alokuje malloc()?

Text seg.
BSS seg.
Heap
Stack
Data seg.
Jinam

Question 11 (+0.4b) Kam alokuje operátor new?

Heap
Stack
Jinam
Text seg.
Data seg.
BSS seg.

Předpokládejte procesor s taktovací frekvencí 1GHz a program čítající 1000 instrukcí. Vykonání každé instrukce trvá jeden takt vyjma instrukcí přístupu do paměti. Pro jednoduchost předpokládejte, že jeden přístup do paměti (spolu s vykonáním dané paměťové instrukce) trvá vždy 100ns. V uvedeném programu 25% instrukcí jsou přístupy do paměti.

Question 12 (+2b) Za jakou dobu vykoná tento procesor uvedený program?

- [x] 25750ns
- [ ] 32350ns
- [ ] 20800ns
- [ ] 22450ns
- [ ] 51500ns
- [ ] 21500ns
- [ ] 58000ns
- [ ] 65450ns
- [ ] 41600ns
- [ ] 45000ns
- [ ] 42500ns
- [ ] 37000ns
- [ ] 50000ns
- [ ] 38000ns
- [ ] 30500ns

Question 13 (+3b) Jak se změní tato doba, když přidáme do systému cache paměť? Cache má míru zásahu 50% pro data a program a doba nalezení záznamu v cache (včetně provedení paměťové instrukce v případě hitu) je 10ns.

- [ ] 6000ns
- [ ] 5500ns
- [ ] 5000ns
- [ ] 3450ns
- [ ] 6750ns
- [ ] 2600ns
- [ ] 6000ns
- [ ] 9300ns
- [ ] 3000ns
- [ ] 4300ns
- [ ] 8000ns
- [ ] 3200ns
- [ ] 6500ns
- [ ] 3500ns
- [ ] 1575ns

Question 14

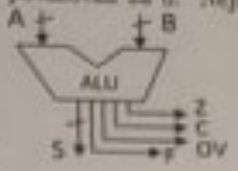
(+2b) Pokud uvažujeme dekadické číslo -10 a aditivní konstantu 15, jaká bude 6-bitová binární reprezentace tohoto kódovaného čísla v kódu s posunutou nulou?

- [ ] 111011
- [ ] 100001
- [ ] 000101
- [ ] 010101
- [ ] 001110
- [ ] 100011
- [ ] 101110
- [ ] 110010
- [ ] 001111
- [ ] 100010
- [ ] 101100
- [ ] 001101
- [ ] 100100
- [ ] 001111
- [ ] 111111

Question 15 (+2b) Datový typ int, který se používá v jazyce C se v paměti obvykle ukládá v:

- [ ] Grayově kódu
- [ ] inverzním kódu
- [ ] Hammingově kódu
- [ ] BCD kódu
- [ ] dvojkovém doplňku
- [ ] dle standardu IEEE 754
- [ ] aditivním kódu
- [ ] přímém kódu
- [ ] není specifikováno

Uvažujte 8-bitovou ALU dle obrázku vpravo. Vstupem sou dvě 8-bitová binární čísla A=0b11101101, B=0b10111111. Výstupem je 8-bitový součet A+B a indikace Z (zero - výsledek je nulový), C (carry - přenos na nejvýznamějším bitu) a OV (overflow - přetečení pro čísla se znaménkem). (**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.)



Question 16 (+1b) Jaký bude výsledek součtu?

1 1 1 1 1 1 1 1

Question 17 (+1b) Jak budou nastaveny příznaky Z, C a OV a bit F, pokud F realizuje logickou funkci $F = \overline{Z} \cdot \overline{C} \cdot \overline{OV}$? Boxy vyplňujte v uvedeném pořadí zleva doprava, tj. první box zleva představuje Z, druhý C, třetí OV a čtvrtý F.

1 1 1 1

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 3 bity, mantisa 6 bitů. Jinak chcete zachovat všechny vlastnosti IEEE754.

Question 18 (+2b) Jak se binárně zakóduje dekadické číslo 7.59 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 nezaokrouhlujte.)

1 1 1 1 1 1 1 1 1

Question 19 (+0.5b) Jak bude v této reprezentaci binárně zakódováno největší možné zobrazitelné reálné číslo?

1 1 1 1 1 1 1 1 1

Question 20 (+0.5b) Jaká bude dekadická hodnota tohoto největšího možného čísla? Uvažujte 3 desetinné pozice.

15.875	27.000	128.000	34.250	25.500	476.500
117.625	22.625	287.000	67.000	209.375	117.500

Question 21 (+0.5b) Jak bude v této reprezentaci zakódováno nejmenší nenulové kladné normalizované číslo?

1 1 1 1 1 1 1 1 1

Question 22 (+0.5b) Jaká bude dekadická hodnota tohoto nejmenšího nenulového kladného normalizovaného čísla? Zaokrouhlete na 3 desetinné pozice.

0.066	0.250	0.023	0.006	0.010	0.014
0.047	0.063	0.031	0.141	0.328	1.125

Předpokládejte následující parametry paměťového subsystému procesoru: procesor je 16-bitový, umožňuje adresaci notlivých byte, organizace paměti je **little-endian**, velikost stránky spravované jednotkou správy paměti je 32byte a je položka 2-úrovňové tabulky pro překlad virtuální adresy zabírá právě 2byte. Přitom předpokládejte, že jak velikost adresní části tabulky, tak vlastní tabulky odkazů na stránky odpovídá právě jedné stránce. Výpis obsahu fyzického adresního toru/paměti:

	x0:	x1:	x2:	x3:	x4:	x5:	x6:	x7:	x8:	x9:	xA:	xB:	xC:	xD:	xE:	xF:
00:	64	8E	98	3E	8A	C7	C6	86	63	7F	A3	AB	25	3D	BC	B5
10:	2D	EF	8E	CC	BC	4C	DD	7F	1D	00	CD	F6	19	F3	2B	FA
20:	49	1F	A4	8E	CF	5F	00	66	FA	37	12	4E	68	3F	E8	F2
30:	D4	61	EE	9E	DF	44	52	94	42	6D	55	63	18	F2	BD	1C
→ 40:	23	00	D2	BA	80	00	9E	5B	DC	33	3E	F9	32	91	70	AC
50:	DF	F1	53	3F	1D	53	AD	CB	E9	EF	41	56	4D	00	D2	3E
60:	A0	99	7A	47	AB	75	5C	88	AA	2A	72	00	00	69	DA	00
70:	17	34	85	18	9F	1A	C4	40	48	00	39	16	76	74	91	49
80:	CE	E7	12	20	1A	59	4C	98	32	10	00	D8	B6	CA	1D	00
90:	9D	16	62	23	C6	CD	00	00	C8	7D	00	F5	93	C2	41	50
A0:	49	68	E5	7C	46	62	37	00	E0	86	93	2C	69	70	00	A8
B0:	31	BC	A4	5A	49	14	73	B2	EE	31	F5	BF	D4	00	A4	43

Nejvyšší úroveň stránkovací tabulky (hodnota PDBR – page directory base register) začíná na adrese **0x0040**. 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 **0x0572**.

Question 23 (+0.75b) Kolik bitů virtuální adresy (včetně offsetu ve stránce) je jednotka správy paměti s výše uvedenými parametry schopná přeložit?

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

Question 24 (+1b) Jaký je offset v rámci fyzické stránky? (Odpovídejte dekadicky.)

33 | 23 | 26 | 16 | 18 | 28
32 | 26 | 16 | 3 | 0 | 16
19 | 34 | 28 | 30 | 10 | 2

Question 25 (+1b) Jaký je index v tabulce stránek v první úrovni? (Odpovídejte dekadicky.)

39 | 26 | 26 | 1 | 2 | 37
50 | 1 | 3 | 16 | 13 | 20
26 | 24 | 1 | 21 | 13 | 17

Question 26 (+1b) Jaký je index v tabulce stránek v druhé úrovni? (Odpovídejte dekadicky.)

54 | 11 | 58 | 13 | 32 | 21
64 | 1 | 0 | 12 | 18 | 30
7 | 3 | 14 | 58 | 7 | 13

Question 27 (+2.5b) Na které fyzické adrese jsou uložena data z virtuální adresy 0x0572? (**Pokyny pro vyplnění:** Křížkujte pouze pokud má být daný bit nastaven na 1. Prázdný box je považován za 0. Nejméně významný bit je vpravo. Odpovídejte binárně.)

Question 28 (+0.75b) Kolik úrovní by bylo potřeba pro stránkovací tabulku, aby byl pokrytý kompletní překlad z 1 virtuální adresy na 16-bitovou fyzickou adresu? Uvádějte absolutní hodnotu počtu potřebných úrovní.

3 | 6 | 8 | 5 | 0
4 | 2 | 7 | nereali