Jolezal Filip
Mark: 25/30 (total score: 25/30)

+18/1/52+

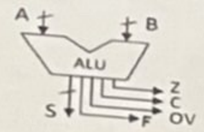
Archit

Zkouškový test, [illegible] 2021/22

← *Prosíme, vyplňte zde své [illegible] (uvedeno na kartě ISIC) [illegible] pro [illegible] použito [illegible] a příjmení.*

Čas na vypracování testu: 90 minut. Otázky označené symbolem ♣ mohou mít vícero správných odpovědí. Ostatní otázky mají pouze jednu správnou odpověď.

Uvažujte 8-bitovou ALU dle obrázku vpravo. Vstupem jsou dvě 8-bitová binární čísla A=0b11100011, B=0b11101110. 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 znamínkem).



(**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 1 (+1b) Jaký bude výsledek součtu?

1/1 1 1 1 1 1 1 1 1

Question 2 (+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 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 9 bitů. Proto jste navrhli reprezentaci, ve které má exponent 4 bity, mantisa 4 bity. Jinak chcete zachovat všechny vlastnosti IEEE754.

Question 3 (+2b) Jak se binárně zakóduje dekadické číslo 5.11 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**.)

2/2 1 1 1 1 1 1 1 1 1

Question 4 (+1b) Jak bude v této reprezentaci binárně zakódováno největší možné zobrazitelné reálné číslo?

1/1 1 1 1 1 1 1 1 1 1

Question 5 (+1b) Jaká bude dekadická hodnota tohoto největšího možného čísla? Uvažujte 3 desetinné pozice.

1/1

111.500	22.375	21.500	50.250	134.500	295.250
248.000	107.625	21.625	46.000	172.250	108.000

Question 6 (+1b) Jak bude v této reprezentaci zakódováno nejmenší nenulové kladné normalizované číslo?

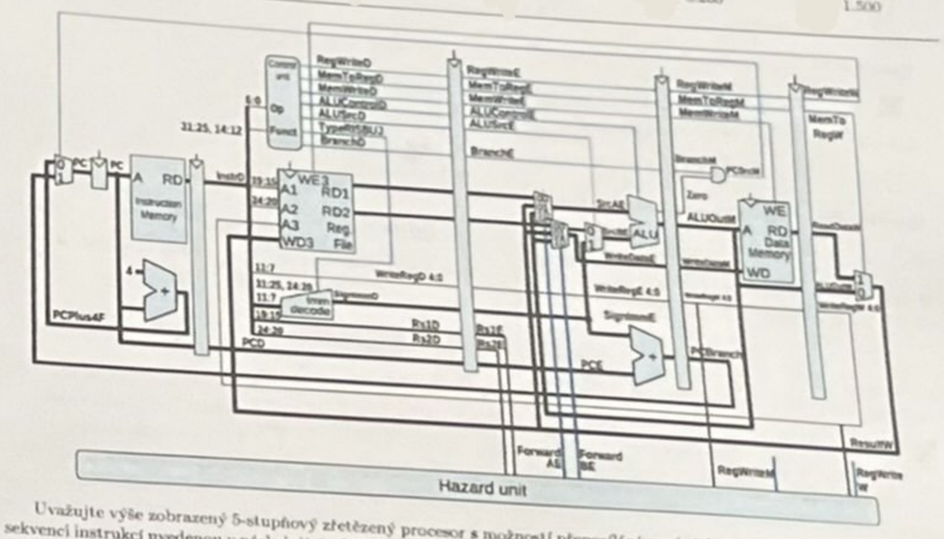
1/1 1 1 1 1 1 1 1 1 1

+18/2/51+

Question 7 (+1b) Jaká bude dekadická hodnota tohoto nejmenšího nenulového kladného normalizovaného čísla? Zaokrouhlete na 3 desetinné pozice.

1/1

0.094	0.055	0.016	0.043	0.063	0.625
0.027	0.084	0.004	0.094	0.266	1.500



Uvažujte výše zobrazený 5-stupňový zřetězený procesor s možností přeposílání nezivýsledků (forwarding) uvnitř procesoru a sekvencí instrukcí uvedenou v následující tabulce. Pro počáteční instrukce sekvence předpokládejte, že jsou již hodnoty vstupních argumentů přítomné v pracovních registrech procesoru. V následujících otázkách vyplňte jakých hodnot mají nabývat signály Forward AE a Forward BE pro jednotlivé instrukce ze sekvence instrukcí.

Pozice	Instrukce
InstA	`sub t0,a0,a1`
InstB	`srai t1,t0,31`
InstC	`and t1,t1,t0`
InstD	`addi t2,zero,0`
InstE	`sub a0,a0,t1`
InstF	`addi t3,zero,0`

instrukce	operace
add rd, rs1, rs2	rd = rs1 + rs2;
addi rd,rs1,C	rd = rs1 + C;
sub rd,rs1,rs2	rd = rs1 - rs2
slt rd,rs1,rs2	rd = (rs1 < rs2) (set less than)
slli rd,rs1,Csht	rd = rs1 << Csht (shift left logical)
srli rd,rs1,Csht	rd = rs1 >> Csht (shift right logical)
srai rd,rs1,Csht	rd = rs1 >> Csht (shift right arithmetic)

Question 8 (+0.5b)
Forward AE v InstB
`srai t1,t0,31`
0.5/0.5 0 1 2

Question 9 (+0.5b)
Forward BE v InstB
`srai t1,t0,31`
0.5/0. 0 1 2

Question 10 (+0.5b)
Forward AE v InstC
`and t1,t1,t0`
0.5/0. 0 1 2

Question 11 (+0.5b)
Forward BE v InstC
`and t1,t1,t0`
0 1 2

Question 12 (+0.5b)
Forward AE v InstD
`addi t2,zero,0`
0 1 2

Question 13 (+0.5b)
Forward BE v InstD
`addi t2,zero,0`
0 1 2

Question 14 (+0.5b)
Forward AE v InstE
`sub a0,a0,t1`
0 1 2

Question 15 (+0.5b)
Forward BE v InstE
`sub a0,a0,t1`
0 1 2

Question 16 (+0.5b)
Forward AE v InstF
`addi t3,zero,0`
0 1 2

Question 17 (+0.5b)
Forward BE v InstF
`addi t3,zero,0`
0 1 2 0.5/0.5

0.5/0.5

0.5/0.5

Question 18 (+2b) *Jakou funkci/výpočet realizuje fragment kódu analyzovaný v předchozích otázkách? Vstupy jsou v registrech a0-a3, výstup v registru a0*

2/2

Minimum ze dvou prvků	Součet čtyřech prvků	Maximum ze dvou prvků
Porovnání dvou prvků	Průměr ze čtyřech prvků	Nejmenší společný dělitel
Součin dvou prvků	Rozdíl dvou prvků	Průměr ze dvou prvků

Question 19 (±1b) Klasická PCI sběrnice je sériová.

1/1

Ne	Ano

Question 20 (±1b) PCI-e sběrnice je sériová.

1/1

Ne	Ano

Question 21 (±1b) Zřetězené zpracování instrukcí (pipelining) urychluje načtení dat z paměti.

0/1

Ano	Ne

Question 22 (±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.

1/1

Ne	Ano

Question 23
(+1b) Algoritmus FIFO v případě konfliktu platných řádek v cache nahrazuje

1/1

první/nejstarší přidaný řádek	náhodně vybraný řádek	nejméně používaný řádek
řádek s nejnižší adresou	dosud nepoužitý řádek	řádek s nejnižší hodnotou
naposled přidaný řádek	naposled použitý řádek	nejdéle nepoužitý řádek

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 24 (+2b) Za jakou dobu vykoná tento procesor uvedený program?

2/2

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

Question 25 (+3b) Jak se změní tato doba, když přidáme do systému cache paměť? Cache má miss rate 10% pro daný program a doba nalezení záznamu v cache (včetně provedení paměťové instrukce v případě hitu) je 10ns.

3/3

5000ns	8000ns	3450ns	2600ns	6750ns
3000ns	5500ns	6500ns	6000ns	3200ns
4000ns	4500ns	3500ns	5750ns	9300ns

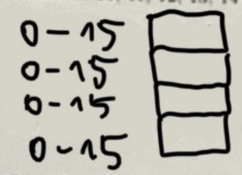
Uveďte strojový kód instrukce "sw" (v hexadecimálním tvaru) níže uvedeného fragmentu. Pokud některá z položek instrukce není specifikována nebo ze zadání úkolu nevyplývá, pak pro účely tohoto zkouškového testu může daná položka nabývat libovolnou hodnotu – ideálně je však tuto položku nevyplňovat. Operační kód použitých instrukcí je následující: addi:0b0010011, add:0b0110011, sw:0b0100011. Instrukce addi má fct3 = 0, instrukce add má funct7=0 a fct3 = 0, instrukce sw má fct3= 0b010.

```
addi x7, x7, -12
add x10, x10, x7
sw x10,-12(x17)
```

I	imm(12) 31:20		rs1(5), 19:15	fct3.14:12	rd(5).11:7	opcode(7) 6:0
R	funct7(7) 31:25	rs2(5), 24:20	rs1(5), 19:15	fct3.14:12	rd(5).11:7	opcode(7) 6:0
S	imm[11:5], 31:25	rs2(5), 24:20	rs1(5), 19:15	fct3.14:12	imm[4:0] ,11:7	opcode(7) 6:0

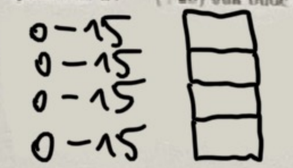
Question 26 (+2b) Jak bude zakódována horní polovina instrukce "sw" v hexadecimálním tvaru? Tzn. horních 16 bitů instrukce vyjádřené v strojovém kódu. (První cifru čísla vyplňte do prvního řádku, druhou cifru do druhého řádku atd. Hodnotám 10, 11, 12, 13, 14 a 15 odpovídá A, B, C, D, E a F.

0/2



Question 27 (+2b) Jak bude zakódována dolní polovina instrukce "sw" v hexadecimálním tvaru?

0/2



Question 28 (+1b) Datový typ float, který se používá v jazyce C se v paměti obvykle ukládá v:

1/1

dle standardu IEEE 754	dvojkovému doplňku	není specifikováno
BCD kódu	přímém kódu	inverzním kódu
Hammingově kódu	Grayově kódu	aditivním kódu