

Зміст

Зміст.....	1
Теоретична частина.....	2
Практична частина.....	11

Теоретична частина

1) Питання: Ktore tvrdenie(-a) ohľadom kľucového slova generic je/su pravdivé ?

Відповіді:

■ Definuje generickú konštantu v rámci bloku entity.

■ Môže byť deklarované v bloku architektury.

☐ Je možné zmeniť hodnotu deklarovanú slovom ľubovoľne meniť počas behu simulácie modulu.

■ Môže byť vždy deklarované pred deklaráciou portov.

2) Питання: Ak (binárny) dekodér má N vstupov, koľko má výstupov?

Відповіді:

☐ N^2

☐ $N/2$

■ 2^N

☐ $2N$

3) Питання: Ktoré tvrdenia sú správne? Cislicový počítač je definovaný špecifikáciou ... (dokončte vetu s výberom z nasledujúcich možností)

Відповіді:

☐ jeho komponentov.

☐ jeho CISC a RISC komponentov.

■ jeho inštrukčne orientovanej architektúry a mikroarchitektúry.

■ jeho inštrukčne orientovanej architektúry a implementačne orientovanej architektúry.

☐ jeho inštrukčnej sady a CPU.

4) Питання: Čo je charakteristické pri uplatnení časového paralelizmu v procese spracovania prúdu inštrukcií?

Відповіді:

■ Delenie inštrukcie na dielčie kroky, ktoré sa vykonávajú súbežne.

□ Zvýšenie počtu ALU jednotiek.

■ Zreťazené spracovanie.

5) Питання: Opisanie počítačového systému sa uskutočňuje na základe určenia jeho funkcie a štruktúry. Pricom:

Відповіді:

■ Funkcie systému, vyjadrená prostredníctvom špecifikácie a opisu procesov ktoré sú v ňom definované, predstavuje pravidlo na dosiahnutie požadovaného cieľa.

□ Funkcia systému, vyjadrená prostredníctvom abstraktných alebo inžinierskych foriem zobrazenia (graty, jazyky, schémy a pod.), vyjadruje kompozíciu jeho prvkov a ich vzájomných väzieb.

□ Štruktúra systému, vyjadrená prostredníctvom špecifikácie a opisu procesov, ktoré sú v ňom definované, predstavuje pravidlo na dosiahnutie požadovaného cieľa.

■ Štruktúra systému, vyjadrená prostredníctvom abstraktných alebo inžinierskych foriem zobrazenia (graty, jazyky, schémy a pod.) vyjadruje kompozíciu jeho prvkov a ich vzájomných väzieb.

6) Питання: Ktoré obvody sú určené na uchovávanie informáci (napr, vnútorného stavu obvodu)?

Відповіді:

■ Klopné obvody

□ Kombinacie obvody

■ Preklapacie obvody

□ Pravdivostná tabuľka

7) Питання: Označte pravdivé tvrdenie, resp. tvrdenia.

Označte jednu alebo viac odpovedí:

Відповіді:

☒ V prípade automatu Moore, výstup je závislý len od vnútorného stavu automatu

☐ V prípade automatu Moore, výstup je závislý tak od vnútorného stavu automatu ako aj od aktuálneho výstupu.

☒ V prípade automatu Mealy, výstup je závislý tak od vnútorného stavu automatu ako aj od aktuálneho výstupu

☐ V prípade automatu Mealy, výstup je závislý len od vnútorného stavu automatu.

8) Питання: Hennessy a Patterson navrhujú postupovať pri návrhu pocitacovej architektury tak, aby boli dodržané 4 základné odporúčania. Čo je charakteristické pre odporúčanie "Jadro musí byť rýchle?"

Označte jednu alebo viac odpovedí:

Відповіді:

☐ Konzistentný inštrukčný formát.

☐ Rôzne inštrukčné formáty majú svoju výhodu.

☒ Menej časte a komplexnejšie príkazy sú realizované ako postupnosť jednoduchých inštrukcií.

☐ MIPS preferuje operácie nad operandmi uloženými v registroch.

9) Питання: Čo predstavuje priestorový paralelizmus?

Відповіді:

☐ Konzistentný inštrukčný formát.

☐ Úloha sa rozdelí na dielčie kroky, ktoré sa vykonávajú súbežne.

☒ Zvýšenie počtu funkčných jednotiek.

10) Питання: Pri rýchlej asociatívnej pamäti s priamym mapovaním platí:

Відповіді:

■ $S = B$, kde S definuje veľkosť bloku cache a B je veľkosť bloku v HP.

□ $S = 1$, kde S definuje veľkosť bloku cache.

□ $S = B/N$, kde S definuje veľkosť bloku cache a B je veľkosť bloku v HP a N je počet blokov v rade.

11) Питання: DNF je súčin implikantov

Відповіді:

□ pravda.

■ nepravda.

12) Питання: DNF je súčet implikantov

Відповіді:

□ pravda.

■ nepravda.

13) Питання: DNF je súčin implikantov

Відповіді:

□ pravda.

■ nepravda.

14) Питання: DNF je súčet implikantov

Відповіді:

■ pravda

□ nepravda

14) Питання: KNF je súčin implikantov

Відповіді:

□ pravda.

■ nepravda.

15) Питання: KNF je súčet implicantov

Відповіді:

☐ pravda.

☒ nepravda.

16) Питання: KNF je súčet impikantov

Відповіді:

☐ pravda.

☒ nepravda.

17) Питання: KNF je súčin implicantov

Відповіді:

☒ pravda

☐ nepravda

18) Питання: Preklápací obvod typu D je asynchrónny bistabilný PO so vstupom D a hodinovým vstupom C.

Відповіді:

☐ pravda.

☒ nepravda.

19) Питання: Ktoré tvrdenie resp. tvrdenia je/sú o príkaze wait pravdivé?

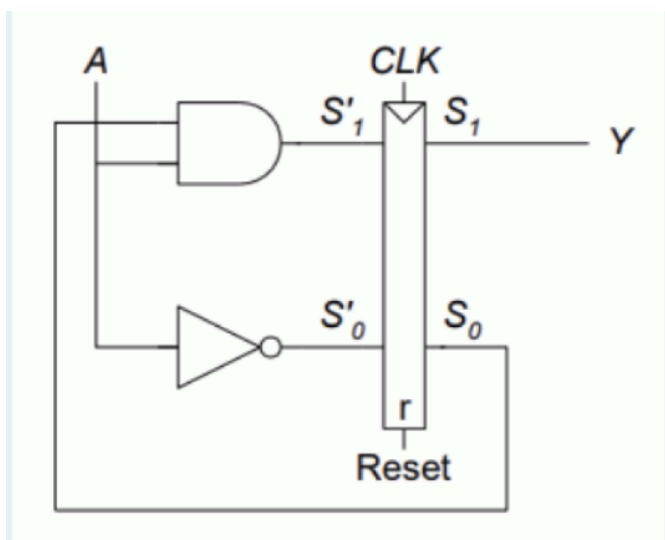
Відповіді:

☒ wait slúži len na účely simulácie

☐ wait je plne syntetizovateľným príkazom

☒ wait slúži k úplnému zastaveniu behu procesu

☐ wait má viacero foriem, ktoré ale nie je možné vzájomne kombinovať



20) Питання: Akého typu je nasledujúca KSA schéma?

Відповіді:

☒ Moore

☐ Mealy

☐ Loop

21) Питання: Abeceda hexidecimálnej číselnej sústavy je tvorená symbolmi?

Відповіді:

☐ 0,1,2,3,4,5,6,7

☒ 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

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

22) Питання: Pre logické úrovne platí :

1 alebo viac odpovedí

Відповіді:

☒ Číslicové signály majú konečný počet diskretných hodnôt, väčšinou dve; 1 a 0

☐ Číslicové signály majú konečný počet analógových hodnôt

☐ Číslicové signály majú nekonečný počet diskretných hodnôt

☐ Číslicové signály majú konečný počet diskretných hodnôt, väčšinou 10: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

23) Питання: Čo je to pretečenie? 1 alebo viac

Відповіді:

- ☐ k výsledok operácie sa rovná N, pričom N je počet zobrazovacích rádov
- ☐ Ak výsledok operácie sa rovná 0.

■ Ak výsledok operácie nie je možné reprezentovať na K bitoch, pričom K je počet zobrazovacích rádov.

- ☐ Ak výsledok operácie je možné reprezentovať na N bitoch, pričom N je počet zobrazovacích rádov.
 - ☐ Ak výsledok operácie je NaN.
-

24) Питання: Read Only Memory je volatilným typom pamäte

Відповіді:

- ☐ pravda.

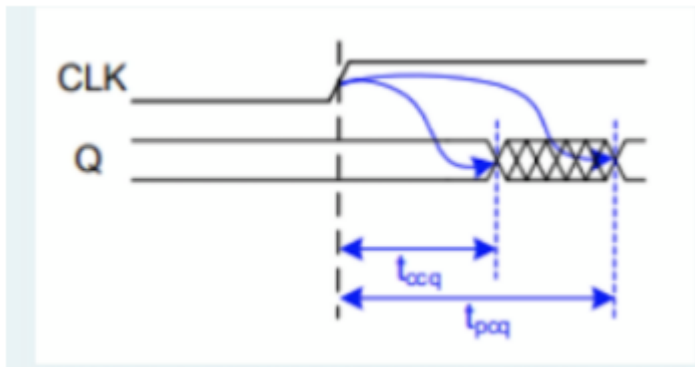
■ nepravda.

25) Питання: Random Access Memory je volatilným typom pamäte

Відповіді:

■ pravda

- ☐ nepravda
-



26) Питання: Čo je to propagačné oneskorenie?

Відповіді:

■ Je to časový úsek(začínajúci so zmenou CLK), po uplynutí ktorého je garantované, že Q má stabilnú(ustálenú) hodnotu

□ Je to časový úsek(začínajúci so zmenou CLK), po uplynutí ktorého sa začína prejavovať zmena na 1 tj. Q ešte nemusí mať ustálenú hodnotu

27) Питання: При праці з K-мапою платі : 1 alebo viac

Відповіді:

□ Každá "0"(nulový bod) musí byť zaradená aspoň do jednej pravidelnej kongurácie, ak cieľom je tvorba disjunktívnej normálnej formy

□ Preferované sú pravidelné kongurácie s minimálnym stupňom k. Pravidelnou konguráciou stupňa k sa nazýva kongurácia 2^k štvorčekov, z ktorých každý má práve k susedov

■ Preferované sú pravidelné kongurácie s maximálnym stupňom k. Pravidelnou konguráciou stupňa k sa nazýva kongurácia 2^k štvorčekov, z ktorých každý má práve k susedov

■ Každá "1"(jedničkový bod) musí byť zaradená aspoň do jednej pravidelnej kongurácie, ak cieľom je tvorba disjunktívnej normálnej formy

28) Питання: Označte pravdivý resp. pravdivé výroky : 1 alebo viac

Відповіді:

□ Hazard typu čítanie po zápise opisuje údajovú antizávislosť

■ Hazard typu zápis po čítaní opisuje údajovú antizávislosť

□ Hazard typu zápis po zápise opisuje údajovú antizávislosť

□ Hazard RAW opisuje štrukturálny hazard

29) Питання: V architektúre MIPS pri práci s funkciami : 1 alebo viac

Відповіді:

- Volajúca funkcia vykoná inštrukciu jal na volanú funkciu
 - Volaná funkcia vykoná inštrukciu jr \$ra
 - Volajúca funkcia uchováva hodnoty potrebných registrov (\$ra, niekedy aj \$t0-t9)
 - Volajúca funkcia smie prepísať obsah oboch registrov, ale nesmie prepísať pamäťové miesta, ktoré používa volaná funkcia
 - Volaná funkcia obnoví obsah uchovávaných registrov
-

31) Питання: Hodnota 'Z' predstavuje :

Відповіді:

- Stav vysokej impedancie daného uzla v obvode. Tento stav nie je možné overiť ampérmetrom
 - Stav vysokej impedancie daného uzla v obvode. Tento stav nie je možné overiť voltmetrom
- ☐ Plávajúci výstupný signál, ktorý môže mať iba hodnotu 0
- Plávajúci výstupný signál, ktorý môže mať tak hodnotu 0 ako aj 1
-

32) Питання: Nech je daná nasledujúca postupnosť inštrukcií:

1. add \$s0, \$s0, \$s1
2. lw \$t0, 0x20(\$t7)
3. addi \$s0, \$s0, -10

Ktorá alebo ktoré inštrukcie sú typu I?

Відповідь:

- addi \$s0, \$s0, -10
 - lw \$t0, 0x20(\$t7)
- ☐ add \$s0, \$s0, \$s1
-

Практична частина

33) Питання: Акý je numerický ekvivalent výrazu 6O"42" z jazyka VHDL? 1 alebo viac:

Відповіді:

■ "100010"

■ 6D"34"

□ 4D"34"

□ 6X"23"

Пояснення:

Formát: NB,,hodnota"

N = počet bitov, B = základ číselnej sústavy

NB nie je povinná položka (prednastavená je decimal)

Číslo	# Bity	Základ	Hodnota	Bin. reťazec
3B"101"	3	binary	5	101
B"11"	unsized	binary	3	11
8B"11"	8	binary	3	00000011
8B"1010_1011	8	binary	171	10101011
3D"6"	3	decimal	6	110
6O"42"	6	octal	34	100010
8X"AB"	8	hexadecimal	171	10101011
others=> '0'	n	binary	0	00...0
others=> '1'	n	binary	$2^n - 1$	11...1

Якщо нам дають число як в першому стовпчику, то перша цифра означає довжину для бінарного значення (якщо немає, то який розмір буде, то такий буде), друга цифра означає систему числення B - двійкова, O - вісімкова, D - десяткова, H - шістнадцяткова, в лапках ми маємо число в тій системі, яку нам дали. Далі ми можемо порівняти кожне число перевівши їх, наприклад, у десяткову. Якщо числа однакові, то відмічаємо як правильне.

34) Питання: Určte čas odozvy v sekundách v prípade vykonávania programu na jednocyklovom procesore, ak program pozostáva z 100 miliónov inštrukcií a perióda strojového cyklu je 570 ns.

Відповідь: 57

Пояснення: переводимо 570 наносекунд в секунди = 0,00000057 секунд. Формула часу відповіді (cas odozvy): кількість інструкцій * кількість циклів * час =
= 100000000 * 1 * 0,00000057 = 57 секунд

35) Питання: Určte čas odozvy v sekundách v prípade vykonávania programu na dvojcyklovom procesore, ak program pozostáva z 100 miliónov inštrukcií a perióda strojového cyklu je 25ns.

Відповідь: 5

Пояснення: переводимо 25 наносекунд в секунди = 0,000000025 секунд. Формула часу відповіді (cas odozvy): кількість інструкцій * кількість циклів * час =
= 100000000 * 2 * 0,000000025 = 5 секунд

36) Питання: Určte hodnotu charakteristiky reálneho čísla A vyjadreného v dekadickéj číselnej sústave a zobrazeného vo formáte jednoduchej presnosti podľa štandardu IEEE-754.

A = -27,5 (mínus dvadsaťsedem celých a päť desatín)

Pozn. Hodnota čísla sa uvádza v dekadickéj číselnej sústave.

Відповідь: 131

Для ясності: Першочергово потрібно знати, що IEEE754 складається зі знаку (перша цифра), зміщеної експоненти або характеристики (наступні 8 цифр, що нам і треба) та мантиси (інші 23 цифри). По порядку будемо знаходити кожную частину. Тобто 32 цифри — запам'ятайте.

Пояснення: Маємо число -27.5. Для визначення експоненти переводимо число 27.5 в двійкову систему — це буде 11011.1 (як це робити, знайдете самі). Далі знаходимо ПРОСТУ експоненту: для цього беремо наше двійкове число (11011.1) і зміщуємо кому вліво, допоки ПЕРЕД комою не лишиться одна цифра — 1.10111. Нам необхідно записати експоненту — те, скільки разів ми змістили кому вліво — тобто 4 рази змістили, значить чотири. Для визначення зміщеної експоненти або характеристики додаємо до 4 число 127 (константа якась, просто знайте це). У висновку маємо 131.

37) Питання: Doplňte chýbajúce operátory do VHDL príkazov tak, aby na výstupoch sum a cout bol správne vypočítaný bitový súčet a prenos do vyššieho rádu.

```
entity scitacka is
    port (a : in std_logic;      -- 1.operand
          b : in std_logic;      -- 2.operand
          cin : in std_logic;    -- vstup prenosu (carry in)
          sum : out std_logic;   -- súčet
          cout : out std_logic); -- výstup prenosu (carry in)
end;

architecture behavioral of scitacka is
begin
    sum <= a □ b □ cin;
    cout <= (a □ b) □ (cin □ (a □ b));
end;
```

Відповідь:

```
entity scitacka is
    port (a : in std_logic; -- 1.operand
          b : in std_logic; -- 2.operand
          cin : in std_logic; -- vstup prenosu (carry in)
          sum : out std_logic; -- súčet
          cout : out std_logic); -- výstup prenosu (carry in)
end;

architecture behavioral of scitacka is
begin
    sum <= a xor b xor cin;
    cout <= (a and b) or (cin and (a xor b));
end;
```

Пояснення: a і b це наш бінарний ввід (або 0, або 1), cin це наше “перенесене число”, що теж може означати 0 або 1. Під цим числом я маю на увазі те, коли ми додаємо чи множимо письмово (3 клас, математика), то ми ‘запам’ятовуємо число” – цю роль і виконує змінна cin - вона працює як таке саме число, але грає роль “чи є в нас перенесене число”. Далі маємо sum - сума яка в нас вийшла, але там або 0, або 1, бо змінна може зберігати лише так. cout нам вказує, чи є “перенесене число”.

38) Питання: Ака је репрезентација циселнеј hodnoty -7 v priamom kode?

Відповіді:

■ 10111

□ 111

□ 1110

Пояснення: Посилання для вас <https://ppi.madosonline.sk/index.php?pid=21>

39) Питання: Koľko rôznych číselných hodnôt vieme zapísať pomocou desiatich bitov?

Відповідь: 1024

Пояснення: 2 v stepení toj kilykosti bitiv, яку нас просять, тобто $10 = 2^{10} = 1024$

Питання: Nasledujúci programový kód:

```
int A(int n){
    if(n == 1){
        return 1;
    } else{
        return (n * A(n - 1));
    }
}
```

je preložený do MIPS assembly kódu a je pustený na MIPS architektúre známej z prednášok. Ako sa zmení register na zápis ak hodnota do sa do nej zapíše ak sa zavolá funkcia A(int n). Výsledok zapíšte bez medzery medzi jednotlivými číslami zápisu v tvare:

označenie_registra=zapsaná_hodnota

Pozn. Hodnotu vypíšte v desiatkovej číselnej sústave!

Відповідь: Q=120

Пояснення: Там код факторіалу на C (сказано за асемблер) і вхідне значення 5, тобто $5! = 120$. Сказано ще за ПЕВНИЙ регістр знамий з преднашок (по правильному \$v0), то я хз, але якщо глянути на цвіки деякі, то регістр Q, тому відповідь Q=120.

40) Питання: Nech \$s0 obsahuje slovo 0x45238967. Pri vykonaní nasledujúcich inštrukcií v počítači, ktorý sa riadi ukladáním slov v pamäti podľa veľkého endianu, akú hodnotu bude obsahovať register \$s0?

lw \$r0, 0(\$0)

lb \$r0, 1(\$0)

Відповіді:

☐ 0x45

☒ 0x23

☐ 0x89

☐ 0x67

Пояснення: У великому ендіані байти зберігаються від старшого до молодшого. Число 0x45238967 розташоване в пам'яті як: [0x45, 0x23, 0x89, 0x67]. Інструкція lb \$r0, 1(\$0) бере другий байт (1 позиція) — це 0x23.

41) Питання: Nech \$s0 obsahuje slovo 0x45238967. Pri vykonaní nasledujúcich inštrukcií v počítači, ktorý sa riadi ukladáním slov v pamäti podľa maleho endianu, akú hodnotu bude obsahovať register \$s0?

lw \$r0, 0(\$0)

lb \$r0, 1(\$0)

Відповіді:

☐ 0x45

☐ 0x23

☒ 0x89

☐ 0x67

Пояснення: У малому ендіані байти зберігаються від молодшого до старшого. Число 0x45238967 розташоване в пам'яті як: [0x67, 0x89, 0x23, 0x45]. Інструкція lb \$r0, 1(\$0) бере другий байт (1 позиція) — це 0x89.

42) Питання: Najmensie 4 bitove cislo v doplnkovom kode ma tvar:

Відповіді:

☒ 1000

☐ 1001

☐ 0000

☐ 1111

Пояснення: Перевести числа в десятковий вид для зручності порівняння.

43) Питання: Aka je hodnota najmensieho cisla reprezentovaného v doplnkovom kóde a vyjadreného pomocou štyroch bitov? Odpoved uveďte v desiatkovej sústave

Відповіді: -8

Пояснення: Діапазон значень в доплнковому кодi (doplňkový kód), якщо маємо чотири біти, є від -8 до +7 або ж від 1000 до 0111.

44) Питання: Nech je daná nasledujúca postupnosť inštrukcií:

```
addi $s0, $s0, 4
addi $s1, $s0, 1
sll $1, $1, 2
beq $s0, $s1, target
addi $s1, $s1, 1
```

target:

```
addi $s1, $s1, 2
```

Aku hodnotu bude mať \$s1?

Oznacte jednu odpoved:

Відповідь:

☐ 1

☐ 2

☐ 3

☐ 4

☐ 5

☒ 6

☐ 7

☐ 8

Пояснення: Перша інструкція `addi $s0, $0, 4` встановлює значення регістра `$s0` рівним 4. Друга інструкція `addi $s1, $0, 1` встановлює значення регістра `$s1` рівним 1. Третя інструкція `sll $s1, $s1, 2` зміщує бінарне число в регістрі `$s1` на два біти вліво, через що значення регістра `$s1` змінюється на 4. Четверта інструкція `beq $s0, $s1, target` перевіряє, чи значення регістрів `$s0` та `$s1` однакові. Якщо вони однакові, виконання переходить до інструкції, позначеної як `target`. Якщо ні, продовжиться наступна інструкція. У цьому випадку відбувається перехід до інструкції `target`. П'ята інструкція `addi $s1, $s1, 1` не виконується, оскільки відбувся перехід до інструкції `target`. Шоста інструкція `addi $s1, $s1, 2` додає 2 до значення регістра `$s1`, змінюючи його значення на 6.
