26

A0B36APO 11.6.2014

Zkouškový test z předmětu A0B36APO

Jméno a příjmení: Jonáš Šerých	Login: serycjon
Jméno cvičícího: Píša	

0,5

1. (1b) Jaké znáte sčítačky? Vyjmenujte! Znáte i nějakou rychlou sčítačku? Jedná se o obvod sekvenční nebo kombinační?

sériové paralelní 2bit 4bit 32bit ⋮ Jelínka, Alfonzie / nejrychlejší sčítačky v širším okolí / s predikcí carry – sekvenční

1

2. (1b) Vysvětlete zásadní rozdíl mezi instrukcí skoku a instrukcí volání podprogramu!

při volání podprogramu se pak dá vrátit za volání – tedy podprogram lze spustit z různých míst a po jeho ukončení se vrátit na místo příslušného volání. to u skoku nelze (vrátit lze se vrátit na nějaké jedno fixní místo)

1,5

3. (2b) Jaké znáte hazardy (v souvislosti se zřetězeným procesorem)? A dále uveďte HW techniky jakými se řeší!

- datové
- řídicí
- ?

forwarding, stalling

3

4. (3b) Uvažujte níže uvedený testovací program. Pro zjednodušení předpokládejte virtuálně adresovanou plně asociativní cache (stupeň asociativity >4) s délkou bloku 64B. Spočítejte celkový miss-rate datové cache, pokud je na počátku prázdná! Předpokládejte, že všechny proměnné s výjimkou polí jsou již v registrech procesoru, a že pole A, B a C jsou uloženy za sebou v paměti. Neuvažujte běh dalších programů!

double – 8B → 8x double do bloku

```
double A[1024], B[1024], C[1024];
for(i=0; i<1000; i+=2)
      A[i] = 10*B[i] - C[i+1];
```

500 — přednačte se na 4 cykly

125 = 5.25

125 missů pro každou proměnnou
500 přístupů

Vaše odpověď (miss-rate): ...1.25%...

5. (3b) Předpokládejte jeno-úrovňové stránkování a že potřebné stránkovací tabulky jsou již v operační paměti. Průměrný čas potřebný ke čtení položky ze stránkovací tabulky, pokud je v paměti, je 100 ns. Za účelem zrychlení se používá TLB cache, která uchovává 32 záznamů (32 párů: číslo virtuální stránky - číslo fyzické stránky), přičemž čas nalezení záznamu je 10ns. Při TLB miss-u, operační systém přistupuje do stránkovací tabulky v paměti aby získal překlad dané virtuální stránky a aktualizuje TLB (čas aktualizace TLB je zahrnut v čase přístupu do stránkovací tabulky v paměti - 100 ns). Neuvažujte tedy použití cache pro aktualizaci TLB. Velikost stránky je 4kB, adresy jsou 32-bitové. Pokud je hit rate TLB roven 0,9, jaký je průměrný čas překladu stránky?

3

$0{,}9 \cdot 10 + 0{,}1 \cdot (10 + 100) = 10 + 0{,}1 \cdot 100 = 20\ ns$

4

Vaše odpověď (průměrný čas překladu stránky): 20 ns

6. Předpokládejte adresní prostor o velikosti 64 bitů (64-bitová adresace). Určete (pokud možno odpovídejte ve tvaru mocniny dvou):

$\frac{2^{64}}{2^{39}} = 2^{64-39} = 2^{25}$

A. (0,5b) velikost adresovatelné paměti v bytech: 2^{64} ✓

B. (0,5b) počet 512 GB disků potřebných pro uložení celého adresního prostoru (G=2^{30}): 2^{25} $\left(= \frac{2^{64}\,B}{2^{9} \cdot 2^{30}\,B}\right)$ ✓

$512 = 2^9$, disk $2^{39}\,B$

C. (2b) velikost jedno-úrovňové stránkovací tabulky, pokud velikost stránky je 4MB a jedna položka stránkovací tabulky má 8B: $2^{45}\,B$ $\left(= 8B \cdot \frac{2^{64}\,B}{2^{22}\,B}\right)$ ✓

$2^3 \cdot 2^{(64-22)} = 2^{45}$

$2^{22}\,B$

D. (1b) Dále posuďte aplikovatelnost konfigurace z bodu C:

to by se nám nikam nevešlo → víceúrovňové tabulky

7. (1b) Jak vypadá položka stránkovací tabulky? - co typicky obsahuje? Vyjmenujte!

1

číslo virtuální stránky, číslo fyzické stránky, dirty bit, další flagy - WB, ...

8. (2b) Uveďte strojový kód instrukce beq (v hexadecimálním tvaru) z níže uvedeného fragmentu:

```
loop:  addi $5, $5, 1
       bne  $5, $3, loop
```

2

Formát instrukce beq je následující: 000101ss sssttttt CCCCCCCC CCCCCCCC. První operand instrukce beq se kóduje bezprostředně za operačním znakem instrukce. Instrukce realizuje operaci: `if($s != $t) go to PC+4+4*C.`

00 101000 11 1111111 1111111 0

Uveďte kód instrukce v hexadecimálním tvaru: 0x 14 A3 FF FE

Prostor pro pomocné výpočty:

C = -2

0010
1101
1110

sssss = 00101
ttttt = 00011

ccc... = 111...0

9. (2b) Porovnejte v několika málo větách PCI a PCIe! (Pomocné pojmy: komunikace, sběrnice, point-to-point, paket, sdílení, multiplex, kompatibilita)

2

PCI | PCIe

kompatibilní

sběrnice
sdílena několika zařízeními

point-to-point
full duplex
Používá packety

10. (3b) Určete návratovou hodnotu níže uvedené funkce/podprogramu, pokud vstupní hodnota funkce je 5. Použití registrů odpovídá konvenci probírané na přednáškách (O32).

```
        addi v0,$0,0
        addi t0,$0,1
L1:     beq  a0,$0,L2
        nop
        addi t1,t0,0
        add  t0,t0,v0
        addi v0,t1,0
        addi a0,a0,-1
        j L1
L2:     nop
        jr ra
        nop
```

```
v0 = 0
t0 = 1
for (a0=5; a0≠0, a0--) {
    t1 = t0;
    t0 += v0
    v0 = t1
}
return
```

a0	t0	t1	v0
5	1	–	0
5	1	1	1
4	2	1	1
3	3	2	2
2	5	3	3
1	8	5	5

2

Návratová hodnota funkce: 8,
Hodnoty regitrů bezprostředně po dokončení instrukce `jr ra`:

Obsah v0: 5, Obsah t0: 8, Obsah t1: 5, Obsah a0: 0

11. (1b) Vyjmenujte statické a dynamické propojovací sítě!

12. (2b) Vyjmenujte vrstvy ISO/OSI modelu! Na které vrstvě pracuje bridge (most) a na které router (brána)?

2

7 Aplikační
6 Prezentační
5 Session
4 Transportní
3 Network
2 Data link
1 Fyzická

bridge: 2, router: 3

13. (1b) Jakou důležitou vlastnost mají atomické operace? Vysvětlete pojem atomičnost!

(1)

jsou nedělitelné - vždy proběhnou najednou, bez možnosti, aby se něco dělo uprostřed.
důležité např u zajištění zámků pro paralelní aplikace

14. (2b) Popište hlavní rozdíl mezi tzv. polled exception handling a vectored exception handling!

(2)

Polled
obecný handler rozhoduje, kam dál
musí se obsluhovat postupně po jedné

Vectored
koukne se do tabulky a předá obsluhu
rovnou příslušnému handleru

15. (2b) Uveďte důvody (výhody) použití mikroprogramového řadiče! Jakými způsoby může být mikroprogramový řadič realizovaný?

(1)

↑ jako konečný deterministický stavový automat

může být přeprogramován → simulace jiné architektury, větší flexibilita