Hodnocení

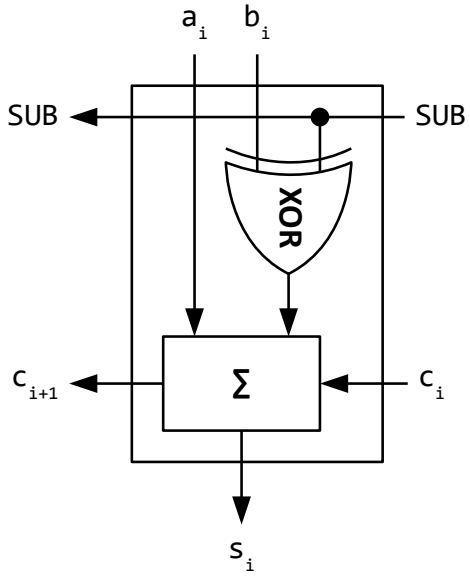
Za úplné a správné zodpovězení otázky je možné získat nejvýše počet bodů uvedený v hranatých závorkách. Rozsah odpovědi by měl pokrývat otázku samotnou, případně bezprostředně související problematiku, pokud je zřejmé, že odpověď by měla triviální charakter. Volba rozsahu odpovědi a její formulace velkou měrou vypovídá o porozumění tématu.

Celkem je možné získat 20 bodů, přičemž výsledné hodnocení je odvozeno z počtu získaných bodů. Známka 1 je podmíněna ziskem alespoň 17 bodů, známka 2 ziskem alespoň 13 bodů a známka 3 ziskem alespoň 10 bodů.

Otázka 1 [2]

Vysvětlete co se stane, když při zpracování instrukce procesorem nastane výjimka. Co musí datová cesta/řadič zajistit, aby mohly být výjimky obslouženy a co taková obsluha obnáší?

Otázka 2 [2]



Odvoďte pravdivostní tabulku pro vstupy a výstupy 1-bitové ALU znázorněné na obrázku (**vstupy** SUB **a** c_i **považujte za spojené**, tj. ALU bude rozlišovat pouze 3 vstupní hodnoty). Pomocí hradel (a poloviční sčítačky) nebo logických funkcí vyjádřete vnitřní strukturu úplné sčítačky uvnitř ALU.

Otázka 3 [2]

Vysvětlete a případně nakreslete, jak by bylo nutné upravit jednocyklovou datovou cestu na obrázku 1 (na straně 2), aby podporovala instrukci

jal JumpAddr

s následující interpretací:

Reg [31] = PC + 4; PC = (PC + 4) [31:28] | Shl2 (JumpAddr)

Uveďte ohodnocení řídících signálů nutné pro vykonání instrukce.

Otázka 4 [1]

Kdy může nastat strukturální hazard v pipeline? Pokud je takový hazard v pipeline možný, jakým způsobem se zajistí, že neovlivní správnost programu?

Otázka 5 [2]

Předpokládejte, že procesor MIPS s klasickou 5-stupňovou pipeline s fázemi IF-ID-EX-MA-WB, kompletním forwardingem a prediktorem skoků typu *always taken* vykonává následující posloupnost instrukcí:

```
        add $1, $5, $3
Label1: sw  $1, 0 ($2)
        add $2, $2, $3
        beq $2, $4, Label1 ; Not taken
        add $5, $5, $1
        sw  $1, 0 ($2)
```

Nakreslete pipeline diagram pro uvedenou posloupnost instrukcí, ve kterém vyznačíte zpracování jednotlivých instrukcí v čase. Předpokládejte, že podmíněné skoky se vykonávají ve fázi EX, a že procesor **MÁ** *delay sloty.*

Otázka 6 [1]

Jaká je přesnost predikce prediktorů typu *always-taken* a *always-not-taken* pro následující posloupnost výsledků stejné instrukce podmíněného skoku:

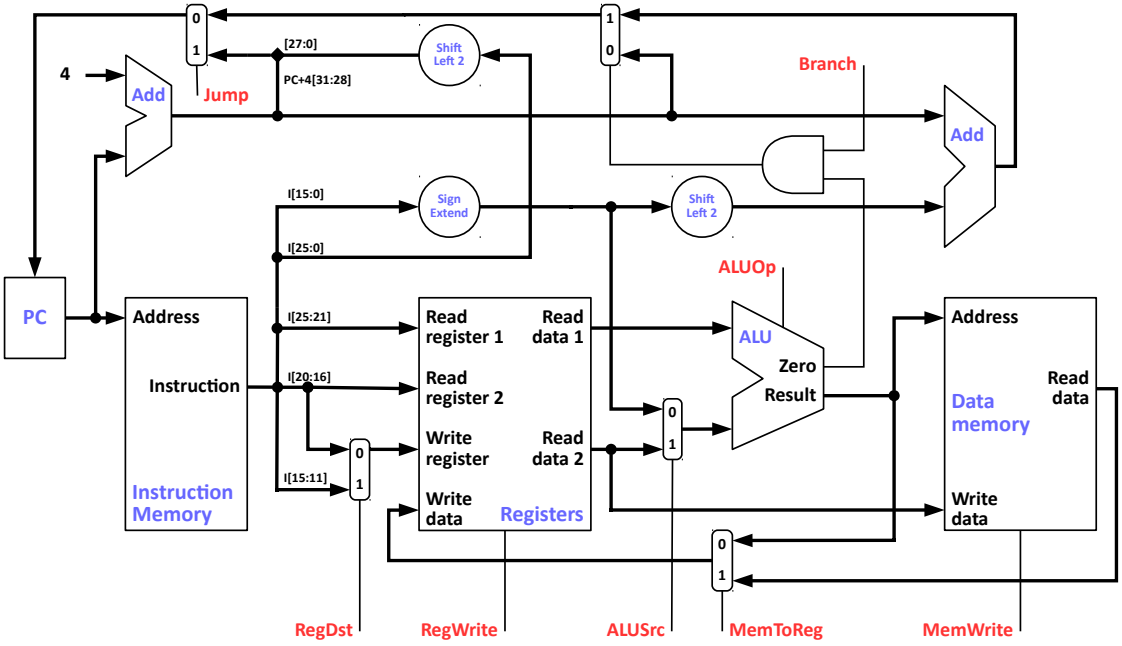
T, T, T, NT, NT

Otázka 7 [1]

Zpožďování pipeline v důsledku špatně předpověděných výsledků podmíněných skoků zvyšuje CPI. Jakého zrychlení by se dalo dosáhnout s 2-bitovým prediktorem, pokud bychom nahradili polovinu instrukcí větvení za speciální instrukci vykonávanou pomocí ALU? Předpokládejte, že správně i špatně predikované instrukce větvení jsou nahrazovány se stejnou pravděpodobností. Uvažujte následující instrukční mix a přesnost predikce 95%.

registrové	beq	jmp	lw	sw
30%	10%	5%	35%	20%

Uvažujte klasickou 5-stupňovou pipeline s fázemi IF-ID-EX-MA-WB a vyhodnocením podmíněného skoku ve fázi EX. Předpokládejte, že v programu nejsou žádné datové hazardy, a že se nevyužívají branch delay sloty.



Obrázek 1

Otázka 8 [2]

Vysvětlete princip funkce superskalárních procesorů s dynamickým plánováním instrukcí v pipeline. Proč je možné dosáhnout vyšší výkonnosti ve srovnání s procesory se skalární pipeline?

Otázka 9 [1]

Jsou data v tzv. sekundární paměti (pevný disk, SSD) přímo použitelná (mohou být načtena do registru nebo použita jako operand aritmetických operací) procesorem? Odpověď zdůvodněte.

Otázka 10 [2]

Vysvětlete co je kapacitní výpadek (*capacity miss*) cache, kdy k němu dochází, jak je možné tento typ výpadků rozlišit od jiných a jak je jejich četnost ovlivněna architekturou cache, resp. paměťové hierarchie. Proč se u nových procesorů kapacita L1 cache (v podstatě) nezvětšuje?

Otázka 11 [2]

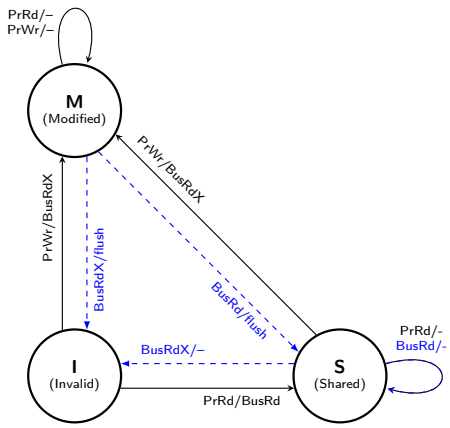
Vysvětlete a na jednoduchém příkladu (stačí pseudokód) demonstrujte, co je to *false sharing*, kdy k němu dochází a jakým způsobem je možné ho odstranit.

Otázka 12 [2]

Přístupy do paměti programu běžícího na více procesorech způsobily následující posloupnost operací nad jednou konkrétní cache line:

1. **P2** write
2. **P0** read
3. **P1** read
4. **P0** write
5. **P2** read

Předpokládejte, že víceprocesorový systém používá write-back cache a koherenční protokol MSI, jehož přechodový diagram je znázorněn na obrázku 2. Pro každou operaci uveďte, jak budou reagovat a vzájemně komunikovat řadiče cache jednotlivých procesorů a jak se bude měnit stav cache line na jednotlivých procesorech.



Obrázek 2