**Samostatná práce č. 3 - autorské řešení**

Akcelerace algoritmů – PV 193

**Zadání:**

Navrhněte grafy přechodů řadiče paměti pro zápis a pro čtení dat, který obsluhuje operační paměť a jednostupňovou CACHE L2 paměť dat – viz schéma. Graf přechodů navrhněte pro obě strategie (WRITE BACK a WRITE THROUGHT) synchronizace obsahu vyrovnávací paměti a operační paměti.

Principiální schéma začlenění řadiče paměti pro obsluhu **CACHE** paměti a operační paměti do struktury číslicového počítače

externí
CACHE L2

hlavní paměť

procesor

V/V řídicí
jednotka

řadič paměti

*Funkční popis:*

Řadič generuje sběrnicové cykly zápisu a čtení s respektováním jedné ze dvou ve výuce diskutovaných metod synchronizace obsahu vyrovnávacích pamětí a operační paměti. Předpokládejte, že šířka toku dat do a z operační paměti a CACHE je stejná a je čtyřikrát širší než délka slova procesoru.

Řešení:

1. Graf přechodů pro operaci **READ** v režimech WRITE BACK a WRITE THROUGHT paměti CACHE L2:

*idle*

*READ*

*wait L2 status*

*HIT*

*data into CPU from L2*

*MISS/no DIRTY*

*write old address into OP*

*MISS/DIRTY*

*read now address from OP*

*data write into L2 & CPU*& reset DIRTY

1. Graf přechodů pro operaci **WRIRE** v režimu WRITE TROUGHT paměti CACHE L2:

*idle*

wait L2 status

write into OP & reset DIRTY

*write old address into OP*

*read now address from OP*& reset DIRTY

read L2

complete data & set DIRTY

write into L2

*WRITE*

*HIT*

*MISS/no DIRTY*

*MISS/DIRTY*

1. Graf přechodů pro operaci **WRIRE** v režimu WRITE BACK paměti CACHE L2:

*idle*

wait L2 status

*write old address into OP*

*read now address from OP*& reset DIRTY

read L2

complete data
& set DIRTY

write into L2

*WRITE*

*HIT*

*MISS/no DIRTY*

*MISS/DIRTY*

### Strategie výběru adresy v CACHE pro **uvolnění**

Cílem je dosáhnout minimální frekvenci uvolňování adres – minimalizace výpadku adres. Adresa vybraná k uvolnění CACHE je označována termín **oběť**. Použití strategií uvolňování adres závisí na typu CACHE paměti. Pro jednostupňovou přímo mapovanou CACHE paměť je výběr strategií výměny adres silně omezen.

Níže popisované strategie se uplatní u plně asociativní CACHE nebo u n-cestně mapovaných CACHE.

#### Strategie OPT

*Optimal* – nahraď tu adresu, která bude znovu zapotřebí co nejpozději v porovnání s ostatními. Z teoretického hlediska se jedná o optimální strategii, ale problémem je zjistit, která adresa bude co nejpozději vyžadována a jaké příznaky tuto skutečnost indikují.

Protože nelze předvídat chování běžných programů, tak se jedná o teoretickou strategii.

#### Strategie FIFO

Tato strategie se nemůže aplikovat u přímo mapované pracuje jako kritérium používá dobu setrvání adresy v CACHE – First *in, first out* – tedy nahraď nejstarší adresu. Doba setrvání adresy v CACHE standardně není kritériem frekvence používání této stránky. Ta může být průběžně používána.

Tato strategie není standardně příliš efektivní, ale je jednoduchá na implementaci.

#### Strategien FIFO s druhou šancí

Při použití adresy se nastaví její příznak aktivity. V jednoduché verzi tato adresa získá příznak aktivity – příznaky se nesčítají. Adresa s tímto příznakem nebude uvolněna, pokud se v CACHE nachází adresa bez tohoto příznaku. Pokud je adresa uvolněna, tak příznak aktivity ztrácí. Výsledkem je, že i staré, ale často používané adresy, přežívají.

Efektivita této strategie se blíží strategii LRU.

#### Strategie LRU

Strategie „Nejméně užívaná v poslední době“ – *Least Recently Used*. Při přesné implementaci vyžaduje udržování informací o používaných adresách. Je používáno několik variant:

##### LRU řízen hodinami

Uvolňuje se adresa s nejstarším záznamem o použití. Jedná se o náročnou programovou nebo technickou realizaci prohledávání pole.

##### Zásobníková LRU

Při použití adresy se adresa nebo její index přesune na vrchol zásobníku. Prioritně uvolňovaná adresa nebo její index se nachází na dně zásobníku.

Jedná se programově náročnou aplikaci, a proto se používá technická realizace této strategie.

#### Strategie LFU

Strategie „Nahraď nejméně využívanou adresu“ – *Least Frequently Used*. Přesná implementace této strategie vyžaduje udržování informací o používaných adresách. Jedná se o náročnou implementaci pracující s čítačem přístupů Pro uchovávání počtu přístupů je nutno implementovat tzv. čítač přístupů.

#### Pseudo-LRU strategie

Nejčastěji aplikovaná strategie. Jedná se o obdobu strategie LRU. Přesnost detekce pasivity adresy je velmi snížena.

Tato strategie se často kombinuje se zjednodušenou verzí strategie LFU.

##### MFU

Strategie „Nahraď nejčastěji používanou adresu“ – *Most Frequently Used*. Uvolňuje se stránka s nejvyšší hodnotou čítače přístupů. Princip funkce spočívá na úvaze, že stránka s nejmenší hodnotou čítače přístupů byla právě zavedena do paměti, a ještě nestihla být použita.