Memoria Cache

Andrei-Leonard PÂRVAN, Marius CARP

Transylvania University of Brașov, România,
Department of Electronics and Computers, Faculty of Electrical Engeneering and Computer Science,
Politehnicii nr. 1, Brașov, România, e-mail: andrei.parvan@student.unitbv.ro

Un calculator are trei sisteme logice: sistemul central de procesare, care procesează datele conform unor instrucțiuni specifice, sistemul de memorare și stocare care stochează instrucțiunile și datele(poate fi vorba despre memorie volatilă sau de un sietm de stocare permanentă, precum hard-disk-ul.), iar cel de-al treilea este sietemul de intrare-ieșire, responsabil pentru comunicația dintre calculator și entități externe.

I. Introducere

Procesorul, în organizarea sa clasică , obține performanța maximă de viteză când în fiecare perioadă/tact de ceas, Tclk, este alimentat cu informație, care poate fi stocată în memorie. Accesarea acestei informații din memoria principală (SDRAM) necesită un timp de ordinul sutelor de tacte(numit latența memoriei) , ceea ce ar înseamna că procesorul trebuie să se oprească până când primeşte informația necesară. Pentru a se elimina această oprire a funcționării µP, se introduce între memoria principală şi µP un buffer/(“rezervor”) cu timp de acces redus, un nivel (L1) suplimentar de memorie, sau două niveluri (L1+ L2), care formează subsistemul **memorie cache**.

Prin introducerea/prezența unei memorii cache se ”ascunde” latența memoriei, dar nu se elimină .Subsistemul memorie cache este situat între µP (CPU) și magistralele din sistemul principial. Cuvântul adresă generat de procesor se trimite simultan la memoria cache și la memoria principală, iar dacă cuvântul căutat se află în cache, informația necesară se obține de aici într-un timp scurt, iar adresarea în continuare la memoria principală nu se mai efectuează. Accesarea la memorie principală se va efectua numai când cuvântul căutat nu s-a aflat în cache, evident cu întârziere de sute de tacte.

Memoria cache este o memorie SRAM, care în raport cu memoriei principală (DRAM), este de capacitate redusă și prezintă un timp de acces 1-2 Tclk . Rațiunea şi eficiența memoriei cache în “ascunderea” latenței memoriei principale se bazează pe:

– **localizarea (în timp şi spațiu)** existentă în programele rulate pe calculator

– **incluziunea informației stocate, L1⊂ L2⊂M⊂HD**

II.Operația de citire a informației

Procesorul trimite la memoria cache adresa locației din memoria principală în care se află cuvântul căutat şi dacă în cache există conținutul locației (cuvântul căutat) de la adresa solicitată din memorie, atunci acest cuvânt este transferat din cache în procesor în 1-2 tacte de ceas (*transfer pe cuvânt*). Dacă acest cuvânt nu este în cache, se accesează nivelul următor (memoria primară/principală) pentru care se consumă timp care poate fi chiar sute de tacte de ceas; dacă cuvântul căutat se află în memoria principală atunci blocul din memorie, în care se află cuvântul respectiv, este transferat şi înscris în memoria cache, de unde, apoi se citeşte cuvântul de adresă căutat şi se aduce în procesor. Dacă blocul căutat nu se află nici în memoria principală atunci se accesează nivelul următor (memoria secundară, hard disk), iar pagina de pe hard disk în care se află blocul căutat este transferată şi înscrisă în memoria principală, iar din memoria principală din pagina în care se află blocul care conține cuvâtul căutat este transferat şi înscris în memoria cache, de unde cuvântul căutat (din blocul adus în cache) se trimite la procesor; tot acest proces de transfer consumă chiar milioane de tacte de ceas. O pagină poate cuprinde, sute sau chiar mii de blocuri.

III.Operația de înscriere a informației

Procesorul trimite la cache adresa din memoria principală unde cuvântul generat de procesor trebuie înscris. Pentru înscriere există două metode: write-through şi write-back.

Prin **write-through,** cuvântul data generat de procesor este înscris întâi la adresa din memoria cache şi apoi se continuă cu înscrierea şi în memoria principală; deoarece se face acces pentru înscriere și la memoria principală se consumă un timp similar cu cel consumat prin negăsirea cuvântului de la procesul de citire.

Prin metoda **write-back** se înscrie cuvântul generat de procesor numai în memoria cache, urmând ca înscrierea și în memoria principală să se realizeze doar atunci când blocul respectiv, din memoria cache în care s-a înscris cuvântul generat de procesor, este trimis înapoi la memoria principală (aceasta se realizează doar când respectivul bloc din cache este înlocuit de către un alt bloc adus din memoria principală); evident că prin write-back, până la înscrierea și în memoria principală a cuvântului generat de procesor, conținutul blocului din memoria cache şi conținutul blocului de aceeaşi adresă din memoria principală nu este acelaşi.

Dacă la înscriere există miss în cache, adică blocul în care se află locația de adresă din memoria principală la care procesorul trebuie să înscrie nu a fost adus încă în memoria cache, atunci se caută blocul respectiv în memoria principală, se înscrie cuvântul generat de procesor în memoria principală, iar blocul respectiv se aduce în memoria cache, ceea ce este un proces consumator de multe sute de tacte de ceas.

IV.Tipuri de mapare

Există o multitudine de posibilități de definire a aplicației/mapării între

mulțimea numerelor blocurilor din memoria principală și mulțimea liniilor/intrărilor/blocurilor din memoria cache. Practic, sunt realizabile doar trei aplicații/mapări care determină următoarele modalități de organizare/structurare pentru memoriile cache: memorie cache cu mapare directă, memorie cache asociativă şi memorie cache set-asociativă.

Memoria cache cu mapare directă

Regula de mapare, pentru o organizare de memorie cache cu mapare directă, permite conținutului unui bloc din memorie, **număr bloc din memorie**, să fie plasat doar într-o singură intrare/linie din numărul total de intrări ale memoriei cache, iar acea adresă/intrare **în memoria cache** a blocului respectiv se determină cu relația următoare:

adresa intrării în memoria cache = (Număr bloc din memorie) *modulo* (număr intrări din cache)

Memoria cache complet asociativă

Memoria complet asociativă (sau full-asociativă) se situează la extrema opusă în raport cu memoria cu mapare directă. Dacă la memoria cu mapare directă un bloc din memoria principală poate fi mapat/plasat doar într-o singură intrare a memoriei cache, la cea full-asociativă un bloc din memoria principală poate fi plasat în oricare intrare din memoria cache. Pentru că blocul de memorie poate fi plasat oriunde în cache, nu mai trebuie căutată intrarea/liniei din cache în care se plasează blocul.

Memoria cache set asociativă

Memoria set asociativă se situează intermediar în raport cu cea full asociativă şi cea cu mapare directă. Memoria set asociativă prezintă un număr de seturi; fiecare dintre aceste seturi conține un număr n de căi/linii de intrare, iar aceste n căi ale unui set formează o memorie full-asociativă cu n intrări; o astfel de organizare este referită ca memorie set asociativă cu n-căi (n-way set-associative). Pentru o memorie cache cu un număr total de m intrări, fiecare dintre seturile sale având n căi, numărul total de seturi (segmente de mici memorii asociative) este egal cu raportul m/n. Fiecare set al memoriei cache este selectat de subcâmpul index din cuvântul de adresă, la fel ca si la memoria cache cu mapare directă, iar adresa setului (dintre cele m/n) din memoria cache set asociativă, în care va fi plasat un bloc din memoria principală, se determină conform relației următoare:

adresa setului din cache = (număr bloc din memorie) *modulo* (numărul seturilor din cache).

V.Referințe

1.Richard Stacpoole, Tariq Jamil - Cache Memories - https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=839642

2.Gheorghe Toacșe, Brașov-2020, Curs-Arhitectura și organizarea microprocesoarelor