

JCacheSim: simulatore visuale di gerarchia di memoria con interprete per programmi MIPS

Paolo Bennati, Roberto Giorgi
Dipartimento di Ingegneria dell'Informazione,
Università degli Studi di Siena

<cognome>@dii.unisi.it

Sommario

La gerarchia di memoria ricopre un ruolo essenziale nella progettazione dei moderni calcolatori ma e' anche uno degli aspetti piu' difficilmente presentabili da un punto di vista didattico. Sono particolarmente utili strumenti di visualizzazione grafica. JCacheSim e' un simulatore in grado di riprodurre il comportamento reale della cache. E' stato sviluppato sottoforma di Java Applet in modo da poter essere disponibile su qualsiasi piattaforma; tramite un meccanismo di logging, JCacheSim consente la raccolta da parte del docente di informazioni sull'attivita' svolta dagli studenti. JCacheSim e' il framework ideale per lo studio della cache durante i Corsi di Calcolatori Elettronici.

1. Introduzione

Nei moderni calcolatori, la gerarchia di memoria rappresenta uno degli aspetti didatticamente piu' ostili da presentare perche' i meccanismi che ne regolano il funzionamento sono difficilmente illustrabili se non tramite la visualizzazione di esempi o "animazioni" [Branovic, 2002] [Prete, 1994].

D'altra parte la gerarchia di memoria ha un ruolo ormai essenziale nel funzionamento di moderni calcolatori in quanto il divario di velocita' processore/memoria e' sempre piu' crescente. Poiche' la performance globale dell'intero sistema e' determinata dall'elemento piu' lento, tale gap rappresenta il cosiddetto "collo di bottiglia di Von Neumann" [Hennessy, 2004].

Assume particolare importanza la corretta progettazione dei livelli della gerarchia di memoria piu' alti, cioe' piu' vicini al processore, ed in particolare le memorie cache [Smith, 1982]. Le memorie cache altro non sono che dei buffer molto veloci che contengono copie di quelle parti di memoria piu' frequentemente e/o

recentemente utilizzata. L'esecuzione dell'applicazione trae vantaggio dalle memorie cache grazie al minor tempo di accesso.

La comprensione dei principi di funzionamento delle memorie cache rappresenta quindi un aspetto didatticamente fondamentale nei corsi di Architettura dei Calcolatori o – tradizionalmente – Calcolatori Elettronici. Il motivo per cui la memoria cache risulta efficace nel ridurre il tempo di accesso medio alle memoria risiede nello studio delle caratteristiche di localita' dell'applicazione [Giorgi, 1997].

Dato un certo programma, non e' assolutamente semplice per lo studente realizzare quelli che sono le potenzialita' di sfruttamento della memoria cache, capire quale possa essere un corretto dimensionamento della stessa, quali debbano essere le politiche locali di gestione (es. politica di scrittura, politica di rimpiazzamento).

Sono disponibili strumenti che mostrano il numero di hit e miss totali (es DINERO [Elder, Dinero IV], o altri piu' sofisticati quali SimpleScalar [Burger, 1997], ma tali strumenti si limitano a mostrare le statistiche di hit e miss e a fornire i risultati essenziali da utilizzare nella ricerca, senza consentire un'indagine dettagliata a fini didattici di quale sia la distribuzione degli accessi in cache e cosa avvenga in una particolare cella di memoria o blocco di memoria in cache. Ci si avvale in questo caso di programmi di simulazione.

Un simulatore di una qualsiasi architettura atro non e' che uno strumento in grado di riprodurre esattamente il comportamento di uno specifico sistema e conseguentemente di modellare il suo reale funzionamento (Fig. 1).

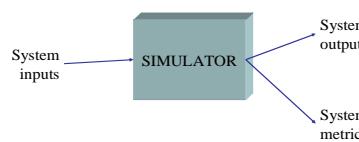


Fig. 1. Schema logico generale di un simulatore.

Con un simulatore si possono estrarre stime sulla performance del sistema ed e' possibile effettuare una completa esplorazione dello spazio di progetto e risulta piu' semplice la validazione del software prima della sua introduzione nel mercato.

JCacheSim vuole essere un contributo in questo specifico campo didattico offrendo uno strumento visuale che puo' essere utile a progettisti e ricercatori nell'acquisizione delle informazioni sull'esecuzione di un programma in un dato sistema. Puo' essere utile nel processo di scelta della migliore configurazione di memoria cache (cache size, block size, block placing policy, replacing policy, ecc...)

Lo studio delle memorie cache rappresenta elemento fondamentale nel percorso formativo di qualsiasi studente. Tuttavia non e' affatto semplice comprenderne l'esatto funzionamento, poiche' e' necessaria la conoscenza non solo di ogni singolo componente del sistema ma anche il meccanismo con cui essi interagiscono fra di loro. Per questo motivo, JCacheSim risulta essere un framework ideale per lo studio teorico durante i Corsi di Architettura dei Computer.

Sono sempre in maggior numero i corsi che in qualche modo offrono strumenti di studio on-line [International data Corporation] [Branovic, 2001]. In quest'ottica JCacheSim e' scritto in forma di Java Applet per due motivi:

- i) essere disponibile su qualsiasi piattaforma con linguaggio Java (basta un semplice browser per internet per utilizzarlo, indipendentemente, ad es. dal sistema operativo);
- ii) poter raccogliere statistiche sulle sperimentazioni svolte dallo studente. Riguardo a quest'ultimo aspetto, lo studente deve identificarsi e verranno raccolti dati sull'uso effettivo dello strumento di simulazione.

2. Caratteristiche di JCacheSim

JCacheSim e' uno strumento in grado di simulare il comportamento reale del sottosistema di memoria cache in modalita' grafica. Consente, semplicemente modificando i parametri nell'interfaccia grafica, di impostare specifiche configurazioni di sistema. Inoltre, lo studente puo' inserire un proprio programma scritto in Assembly MIPS, o utilizzare alcuni degli esempi predisposti. Un semplice assemblatore interno si occupa di effettuare il controllo sintattico e la traduzione in linguaggio macchina.

JCacheSim e' stato sviluppato in linguaggio Java e puo' essere installato su un qualsiasi WebServer opportunamente configurato. Questo ne garantisce un'ampia accessibilita' da parte degli utilizzatori oltre ad un ovvio semplice interfacciamento con essi.

Una volta compilato il codice, il simulatore MIPS puo' eseguire il programma in tre diverse modalita':

- *Single Operation*: viene eseguita una singola operazione MIPS;
- *Single Step*: ogni singola operazione e' scomposta nelle sue parti di fetch ed execute e l'utente puo' seguire step-by-step il comportamento della cache;
- *Program*: l'intero programma compilato viene sequenzialmente eseguito automaticamente.

Durante tutta l'esecuzione del programma il contenuto di ogni singola cella di memoria e' visibile, rendendo semplice la comprensione dell'associazione fra indirizzi di memoria e indirizzi in cache. Alla fine i risultati della simulazione sono mostrati sottoforma di valori numerici e di grafici con i quali e' possibile esplorare ad esempio la localita' degli accessi di un determinato programma inserito dallo studente.

3. Esecuzione di JCacheSim

Durante l'utilizzo dello strumento, l'utente e' guidato tramite una interfaccia intuitiva. Come primo passo e' necessario impostare la configurazione del sistema. Si tratta di impostare la lunghezza di parola del sistema (32 o 64 bit) la configurazione della cache e la dimensione della memoria principale (Fig. 2).

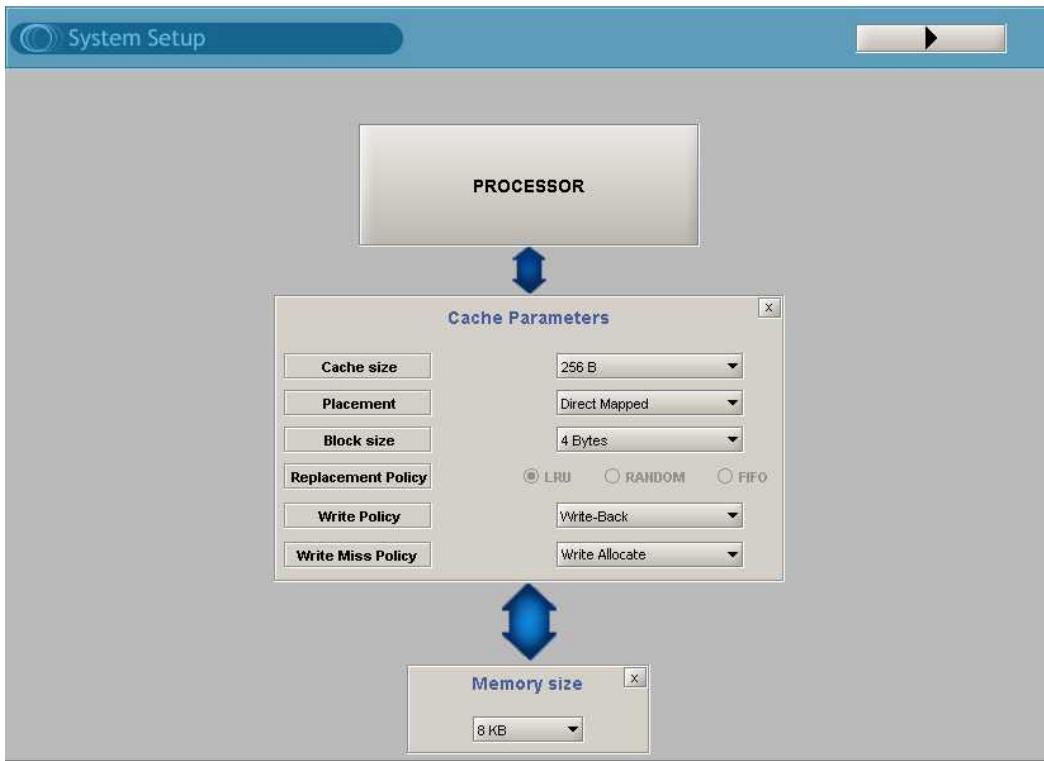


Fig. 2. Setup del sistema

La seconda fase consiste nella compilazione di un codice in formato assembler MIPS. E' possibile caricare dei codici di esempio oppure utilizzare una finestra di editing. L'interfaccia riporta sulla parte sinistra per comodita' l'elenco delle istruzioni modellate nel compilatore (Tabella 1) con la sintassi da utilizzare.

Category	Instruction
Arithmetic	add, addi, addu, sub, subu, nop, mult, mthi, mtlo
Logic	and, nor, or, sll, srl, sllv, sriv, xor
Data transfer	lb, lw, lui, sb, sw
Unconditional branch	beq, bne, slt, sltu
Conditional branch	j, jal, jr
Interrupt management	nop
Floating point instruction	add.s, bclf, bclt, c.eq.s, c.lt.s, c.le.s, div.s, lwc1, mfcl, mtcl, mul.s, sub.s, swc1

Tabella 1. Set di istruzioni supportato dal compilatore.

Con l'apposito pulsante si avvia la compilazione; eventuali errori sono opportunamente segnalati nell'apposito spazio dell'interfaccia (Fig. 3).

Lo step successivo e' quello della simulazione vera e propria. L'utente puo' decidere come eseguire il codice precedentemente compilato secondo le tre modalita' previste. Ogni azione dell'esecuzione e' visualizzata e animata graficamente. Sulla parte sinistra e' riportato il contenuto della memoria cache e sulla destra invece quello della memoria principale. In alto, ogni istruzione e' descritta in dettaglio. Sono inoltre presenti due sezioni riassuntive con il codice da eseguire ed il contenuto dei registri. Nel caso di esecuzione Single Step o Single Operation ogni istruzione e' anche descritta da un'apposita finestra di messaggio (Fig. 4).

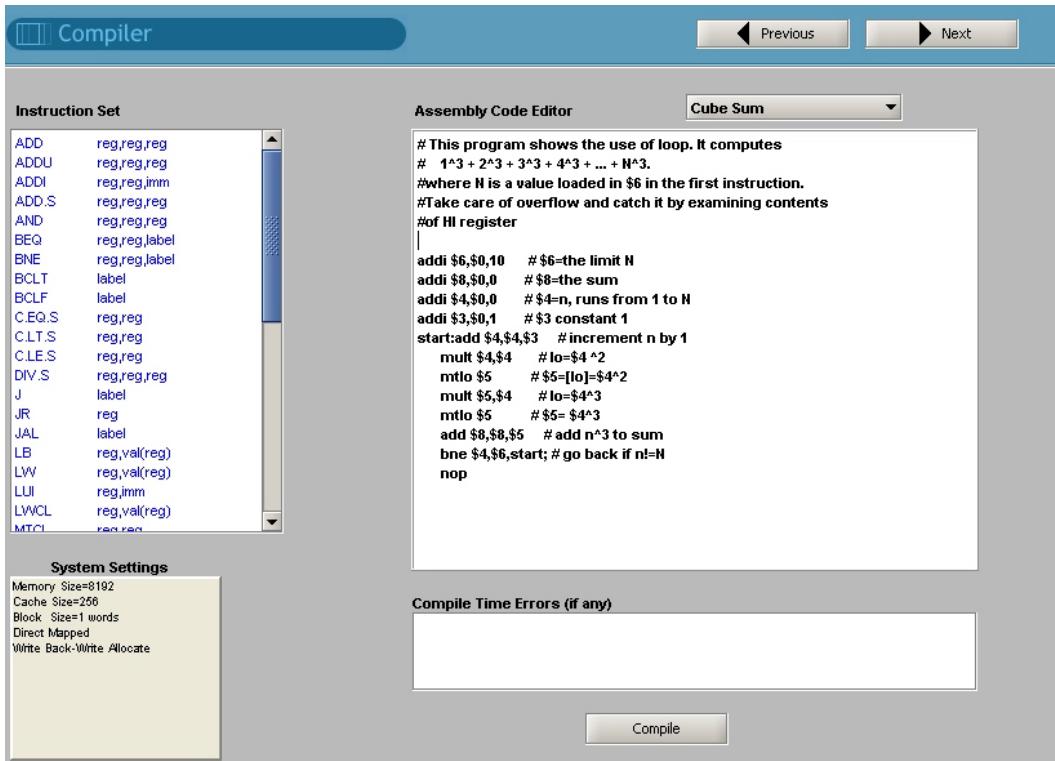


Fig. 5. Il compilatore di jCacheSim

A simulazione terminata, e' possibile visualizzare le statistiche attraverso una ulteriore schermata. Tale schermata e' suddivisa in tre sottosezioni:

- nella prima schermata sono presentati i parametri generali della CPU, del bus, della memoria principale e della cache: tipo e numero di istruzioni eseguite, utilizzo del bus, tipologia di accessi alla memoria oltre a miss-rate e hit-rate;
- nel secondo pannello e' presentato un grafico con il numero di read/write in rapporto al numero totali degli accessi; inoltre e' presente anche una tabella che mostra le statistiche di tutte le operazioni di accesso alla memoria;
- nella terza finestra sono presenti alcuni grafici che riportano dettagliatamente le operazioni eseguite in ogni blocco (Fig. 5).

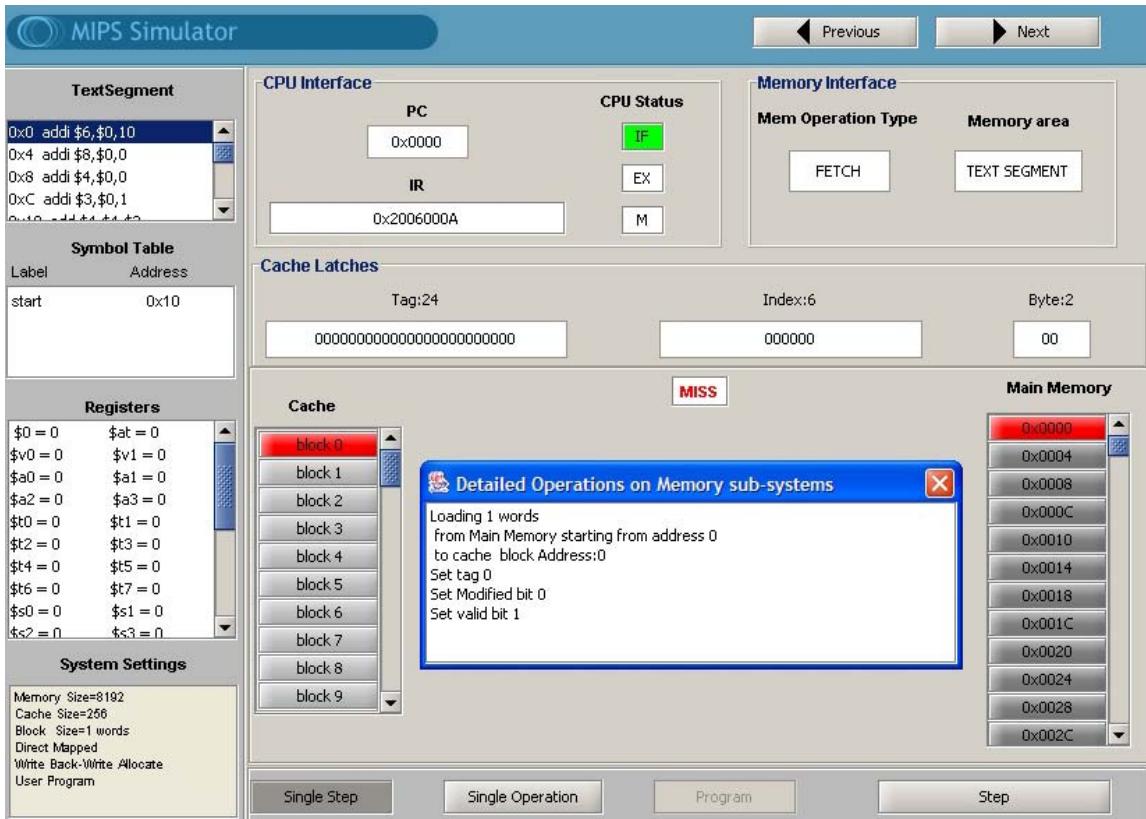


Fig. 4. L'interfaccia della sezione di simulazione.

3. JCacheSim come strumento didattico

JCacheSim costituisce uno strumento di supporto allo studio dei sistemi di elaborazione durante i Corsi di Architettura dei Calcolatori. E' un tool semplice ed intuitivo che guida passo-passo lo studente nella configurazione del sistema, nella compilazione del codice e nella simulazione dello stesso. Le statistiche sono facilmente visualizzabili grazie ai numerosi grafici disponibili al termine della simulazione.

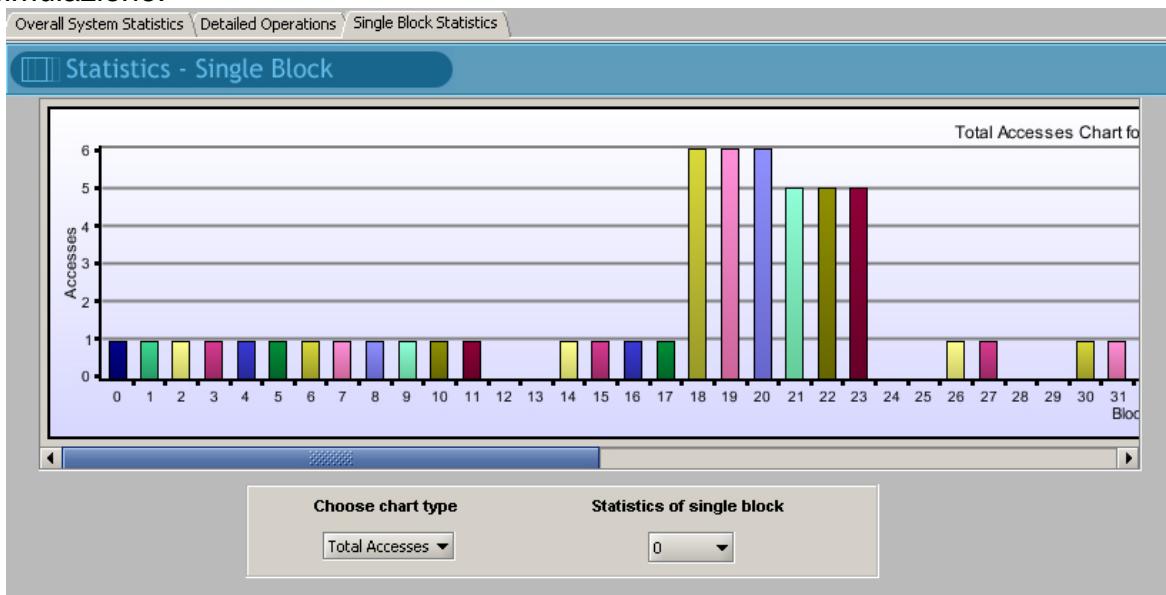


Fig. 5. Statistiche del singolo blocco.

JCacheSim risulta essere un valido aiuto anche ai docenti, che solitamente incontrano notevoli difficoltà nello spiegare argomenti come la gerarchia di memoria,

il linguaggio assembler e, piu' in generale, l'interazione tra tutti i sottosistemi del calcolatore. Con JCacheSim i docenti possono organizzare dimostrazioni in laboratorio per illustrare quanto spiegato durante le lezioni teoriche.

Poihe' JCacheSim funziona tramite Web ed implementa al suo interno un meccanismo di identificazione dell'utente (Fig. 6) e raccolta delle statistiche di utilizzo, il docente potrà seguire l'utilizzo del tool fatto da ogni singolo studente, il tempo speso da ognuno di essi nelle varie parti del simulatore.



Fig. 6. Meccanismo di Login.

4. Conclusioni e sviluppi futuri

In questo articolo e' stato presentato JCacheSim, un simulatore di visuale di sistemi MIPS. Tale simulatore e' in grado di assemblare un programma scritto in linguaggio assembler MIPS e di simularne l'esecuzione. Al termine vengono fornite dettagliate statistiche sull'applicazione eseguite e mostrate in dettaglio tutte le operazioni che si svolgono nella gerarchia di memoria.

Il simulatore e' stato sviluppato in linguaggio JAVA e generato come applet java: risulta per questo essere facilmente utilizzabile via web.

JCacheSim puo' risultare particolarmente utile ai docenti, studenti universitari, poiche' consente di specificare e variare una grande varietà di parametri del sistema e di visualizzare il comportamento interno di componenti architetturali del calcolatore.

Il simulatore e' gia' disponibile alla pagina web:

<http://www.dii.unisi.it/~giorgi/jcachesim>.

5. Riferimenti bibliografici

Giorgi R., Prete, C. A., Prina G.: Cache memory design for embedded systems based on program locality analysis, Proceedings of the 1997 International Conference on Microelectronics Systems Education (MSE '97), pp. 0016

Hennessy J. L., Patterson D. A: Computer Organization and Design, 3rd edition, Morgan Kaufman/Elsevier, 2004

Smith A. J.: Cache memories. ACM Computing Surveys, 14(3):473–530, 1982.

Elder, J., M.D. Hill. Diner IV Trace-Driven Uniprocessor Cache Simulator. <http://www.cs.wisc.edu/~markhill/DinerIV>.

- Burger D., Austin T. M.: The SimpleScalar tool set, version 2.0. Technical Report 1342, University of Wisconsin--Madison Computer Sciences Department, 1997
- Branovic I. and Giorgi R., Prete C.A.: Web-based training on Computer Architecture: The case for JCachesim. IEEE Workshop on Computer Architecture Education (WCAE-02), Anchorage, AK, USA, pp. 56-60,2002.
- International Data Corporation: Distance Learning in Higher Education: Market Forecast and Analysis, 1999-2004.
- Prete, A.: Cachesim: A graphical software environment to support the teaching of computer system with cache memories. Proceedings of 7-th SEI Conference on Software Engineering Education, Springer-Verlag, January 1994.
- Branovic, I., Milutinovic, V.: Tutorial on Advances in Internet-based Education, (<http://galeb.etf.bg.ac.yu/~vm/tutorials>), School of Electrical Engineering, University of Belgrade, Serbia, Yugoslavia, 2001.