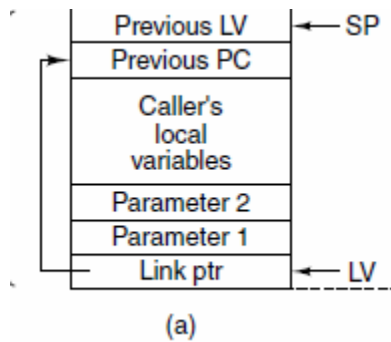


Esercizio 1

1. Discutere il funzionamento dell'IFU: in quale architettura è stata introdotta, per quale motivo, in cosa consiste, ...
2. Facendo riferimento alla figura data, immaginate un metodo A (con due parametri e alcune variabili) che invoca, nell'architettura MIC1, un metodo B con tre parametri e altre variabili. Spiegate come funziona il meccanismo di *invokevirtual*.



Esercizi IJVM 2

Si scriva un metodo IJVM `stampa(n, s)`, dove n è un numero naturale maggiore di zero e s una variabile booleana con il seguente significato: se essa vale zero, il metodo deve stampare i numeri pari tra 2 e n , se invece vale uno, allora il metodo stampa i numeri dispari compresi tra 1 ed n . Per es., `stampa(10, 0)` produce: 2, 4, 6, 8, 10; `stampa(6, 1)` genera 1, 3, 5.

Esercizio 3

- 1) Una cache contiene sia codice che dati, quindi i blocchi che contiene possono essere soggetti sia a lettura/fetch che a scrittura (cioè possono essere modificati da qualche istruzione) e in caso di scrittura si usa la politica *write back*. Supponiamo inoltre che essa sia organizzata come una cache associativa a insiemi a 2 vie con 2048 righe (essendo a 2 vie, ciascuna riga potrà ospitare 2 blocchi) e che la dimensione dei blocchi sia 32 byte. Tutte le domande seguenti si riferiscono a questa struttura di cache.
 - 1.1) In cosa consiste la politica *write back* per la gestione delle scritture, e in che modo la memorizzazione di un “dirty bit” (o bit di modifica) per ogni blocco contenuto in cache può migliorarne l’efficienza?
 - 1.2) Nell’ipotesi che gli indirizzi di memoria siano da 30 bit, come dev’essere partizionato un indirizzo nei tre campi “Tag”, “numero riga” e “offset” per poter verificare se è presente in cache? (cioè in quale posizione si trova e quanti bit occupa nell’indirizzo ciascuno dei tre campi elencati sopra? Spiegare come si può ricavare questa informazione dai dati sulla struttura della cache riportati all’inizio).
 - 1.3) Calcolare la dimensione **totale** in bit della cache (sommando ai bit di dati / istruzioni che sono contenuti nei blocchi anche quelli relativi alle informazioni associate a ciascun blocco necessarie al funzionamento della cache). Non basta indicare un valore, mostrare la FORMULA usata per ottenerlo.
 - 1.4) Se il byte di indirizzo X viene trovato nella seconda via della riga numero 0x038 della cache, dove è memorizzato il Tag 0x00CA, e supponendo che il byte cercato sia il numero 18 nel blocco dati (gli indici di byte dentro il blocco vanno da 0 a 31) qual è l’indirizzo X? (esprimere l’indirizzo X sotto forma di sequenza di 30 bit).

- 2) Scegliere 4 modalità d'indirizzamento dalla seguente tabella e spiegare quali informazioni devono essere contenute nel formato di una istruzione di linguaggio macchina per indicare dove trovare un operando quando si utilizzano tali modalità d'indirizzamento (eventualmente fare qualche esempio).

| | |
|--|--|
| a. indirizzamento dello stack | e. indirizzamento indiretto tramite registro |
| b. indirizzamento diretto | f. indirizzamento di registro |
| c. indirizzamento indicizzato | g. indirizzamento immediato |
| d. indirizzamento indicizzato esteso (base-indice) | |

TABELLA DELLE MODALITA' DI INDIRIZZAMENTO

- 3) **3.1)** Nel contesto della memoria virtuale paginata spiegare: cosa è la tabella delle pagine, cosa è il Translation Lookaside Buffer (TLB), cosa è la Memory Management Unit (MMU). Si possono utilizzare anche schemi ed esempi per rendere più chiara la spiegazione.
- 3.2)** Come avviene la traduzione da indirizzo logico a indirizzo fisico nella memoria virtuale paginata? Cosa accade se la pagina che contiene un dato indirizzo logico NON è caricata in memoria RAM?