

A organização de uma cache pode ser descrita pelo tuplo (S, E, B, m). A cache é organizada em  $S = 2^s$  sets, cada set com E linhas. Cada linha consiste num bloco de dados com  $B = 2^b$  bytes.

O parâmetro m indica o tamanho do endereço da máquina, i.e., quantos bits tem um endereço, formando  $M = 2^m$  endereços únicos. Em termos de endereçamento da cache, os b bits menos significativos do endereço formam o block offset, identificando o byte endereçado dentro do bloco. Os s bits seguintes do endereço constituem o set index, identificando o set onde esse endereço mapeia. Os restantes  $t = m - (s + b)$  bits constituem a tag, que identifica de forma unívoca o endereço armazenado em cada linha da cache. O número total de linhas de uma cache é dado por  $S * E$  e a capacidade total da cache em bytes é dada por  $C = S * E * B$  (a capacidade refere-se apenas aos dados; os bits necessários para a tag e valid bit não são contabilizados).

### 1. Exercícios

1. Cada linha da tabela abaixo fornece alguns parâmetros da organização de uma cache com 2048 bytes de dados e linhas de 32 bytes, utilizada numa máquina com endereços de 32 bits. Complete a tabela com os parâmetros em falta e indique o modo de mapeamento utilizado (directo, completamente associativo ou n-way set associative – neste último caso indique o valor de n).

E	S	b	s	t	Mapeamento
4					
64					
1					

2. A tabela abaixo apresenta na coluna da esquerda uma sequência de endereços ( $m=5$ ) de acesso à memória gerados por um determinado programa. As três colunas seguintes representam 3 diferentes modos de mapeamento numa cache que usa o algoritmo de substituição LRU. Preencha estas colunas indicando em que set/linha (dentro do set) mapeia cada endereço e indicando se se trata de um cold miss, colisão ou de um hit (veja o exemplo na 1ª linha). Considere a cache inicialmente fria. Finalmente indique na última linha a miss rate observada.

Addr	(S=4, E=1,B=2,m=5)	(S=1, E=4,B=2,m=5)	(S=2, E=2,B=2,m=5)
5 (00101)	Set 2 / Linha 0 (cold miss)	Set 0 / Linha 0 (cold miss)	Set 0 / Linha 0 (cold miss)
14			
10			
29			
4			
21			
16			
5			
Miss rate:			

3. O algoritmo abaixo verifica se um determinado endereço de memória está ou não carregado na cache. Sabendo que:

- a função `bits(addr, bitLS, bitMS)` devolve o inteiro representado pelos bits que vão desde o `bitLS` (menos significativo) até ao `bitMS` (mais significativo) inclusive;
- o array `tag[s][l]` armazena a tag correspondente ao set `s` e linha `l`;
- o array `valid[s][l]` corresponde ao valid bit do set `s` e linha `l`;

caracterize esta cache relativamente aos parâmetros (`S`, `E`, `B`, `m`) e indique ainda a sua capacidade, bem como o modo de mapeamento utilizado.

```
bool in_cache (int addr) {
    int set = bits (addr, 5, 15);
    int addr_tag = bits (addr, 16, 31);
    bool hit = false, continue = true;
    for (linha = 0 ; linha < 8 && continue ; linha++) {
        if (tag[set][linha] == addr_tag) {
            continue = false;
            hit = valid[set][linha];
        }
    }
    return hit;
}
```