1. **Em todos os casos foi usado o algoritmo de substituição random**

Caso 1:

Usando mapeamento direto para ambos os casos, com tamanho de bloco igual a um, write through:

1. Para a cache de 16 palavras, temos um modo de endereçamento consistindo em: Tag de 26 bits, índice de 4 bits e 2 bits de endereçamento de byte;

Para a de 32 temos um modo de endereçamento consistindo em:Tag de 25 bits, índice de 5 bits e 2 bits de endereçamento de byte;

1. Para a cache de 16 palavras, o miss count foi de 263. Para a de 32 palavras, foi de 76. Essa redução deve-se ao fato de que, por possuir maior tamanho de cache, mais blocos podem ser armazenados na cache de 32 palavras, o que diminui a taxa de falta.

Caso 2:

1. Mapeamento direto: Tag de 26 bits, índice de 4 bits e 2 bits de endereçamento de byte;

Associativo 2-way: Tag de 27 bits, índice de 3 bits e 2 bits de endereçamento de byte;

Completamente associativo: Tag de 30 bits e 2 bits de endereçamento de byte;

1. Para o mapeamento direto, o miss count foi de 263. Para o Associativo 2-way, foi de 148. Para Completamente associativo, foi de 14. Como o endereçamento do completamente associativo todos os blocos possuem o mesmo índice a substituição era feita apenas quando a cache estivesse cheia o que fez com que o espaço fosse melhor aproveitado, melhorando a taxa de hit. O Associativo 2-way também o espaço pois existem dois possíveis locais para endereçar um bloco dentro de um conjunto o que também melhora a ocupação de cache por diminuir a necessidade de substituição

Caso 3:

1. 32 palavras com blocos de 2 palavras: Tag de 25 bits, índice de 4 bits, offset de bloco de 1 bits e 2 bits de endereçamento de byte;

32 palavras com blocos de 4 palavras: Tag de 25 bits, índice de 3 bits, offset de bloco de 2 bits e 2 bits de endereçamento de byte

1. Para a cache com blocos de 2 palavras, o miss count foi de 73. Para a cache com blocos de 4 palavras, o miss count foi de 70. Essa pouca diferença acontece porque o código não tem tanta localidade espacial para os dados, o que não possibilita que o aumento na quantidade de palavras por bloco seja tão efetivo.

**2) Em todos os casos foi usado o algoritmo de substituição random**

Caso 1:

Usando mapeamento direto para ambos os casos:

1. Para a cache de 16 palavras, temos um modo de endereçamento consistindo em: Tag de 26 bits, índice de 4 bits e 2 bits de endereçamento de byte;

Para a de 32 temos um modo de endereçamento consistindo em:Tag de 25 bits, índice de 5 bits e 2 bits de endereçamento de byte;

1. Para a cache de 16 palavras, o miss count foi de 401. Para a de 32 palavras, foi de 29. Essa redução drástica deve-se ao fato de que, por possuir maior tamanho de cache, mais blocos podem ser armazenados na cache de 32 palavras, o que diminui a taxa de falta. Como as instruções são sequenciais e o programa executa um loop, a cache de 16 bits sofre substituições constantes, o que impacta negativamente a taxa de acertos. Isso não influencia tanto a taxa da cache de 32 palavras.

Caso 2:

1. Mapeamento direto: Tag de 26 bits, índice de 4 bits e 2 bits de endereçamento de byte;

Associativo 2-way: Tag de 27 bits, índice de 3 bits e 2 bits de endereçamento de byte;

Completamente associativo: Tag de 30 bits e 2 bits de endereçamento de byte;

1. Para o mapeamento direto, o miss count foi de 401. Para o Associativo 2-way, foi de 395. Para Completamente associativo, foi de 377. Como as instruções são sequenciais e a cache é de 16 palavras, ocorrem muitas substituições de palavras. Como esperado, o resultado melhor foi o da cache Completamente Associativa, por aproveitar melhor o espaço de memória e explorar melhor a propriedade de localidade temporal.

Caso 3:

1. 32 palavras com blocos de 2 palavras: Tag de 25 bits, índice de 4 bits, offset de bloco de 1 bits e 2 bits de endereçamento de byte;

32 palavras com blocos de 4 palavras: Tag de 25 bits, índice de 3 bits, offset de bloco de 2 bits e 2 bits de endereçamento de byte

1. Para a cache com blocos de 2 palavras, o miss count foi de 15. Para a cache com blocos de 4 palavras, o miss count foi de 8. Esses números baixos se devem ao alto aproveitamento da localidade espacial das duas caches, pois as instruções são sequenciais e existem poucos desvios para quebrar esta linearidade.