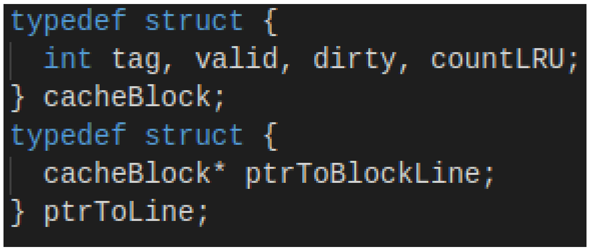
Summary ECE 3056 LAB 2

A. Data structures



```
typedef struct {
  int tag, valid, dirty, countLRU;
} cacheBlock;
typedef struct {
  cacheBlock* ptrToBlockLine;
} ptrToLine;
```

Two data structures (both structs) are used to implement the cache. The first struct defines the cache block which includes the tag, dirty bit, valid bit and the LRU timer – countLRU. The second struct defines a pointer to the block lines in the cache block.

Initially, I am setting up entire cache block based on the no. of cache lines in the block. I am also setting tag, valid, dirty, lru counter to zero since the cache is empty in the beginning. Once the cache block is allocated, you can begin accessing the cache. Once the cache is accessed, you can update the total number of accesses. Now you need to extract the no. of offset bits and index bits to find the tag value. Once tag is found, you can extract the index as well. Once the indexing parameters are explained, you update the contents of the block depending on whether if you hit or miss.

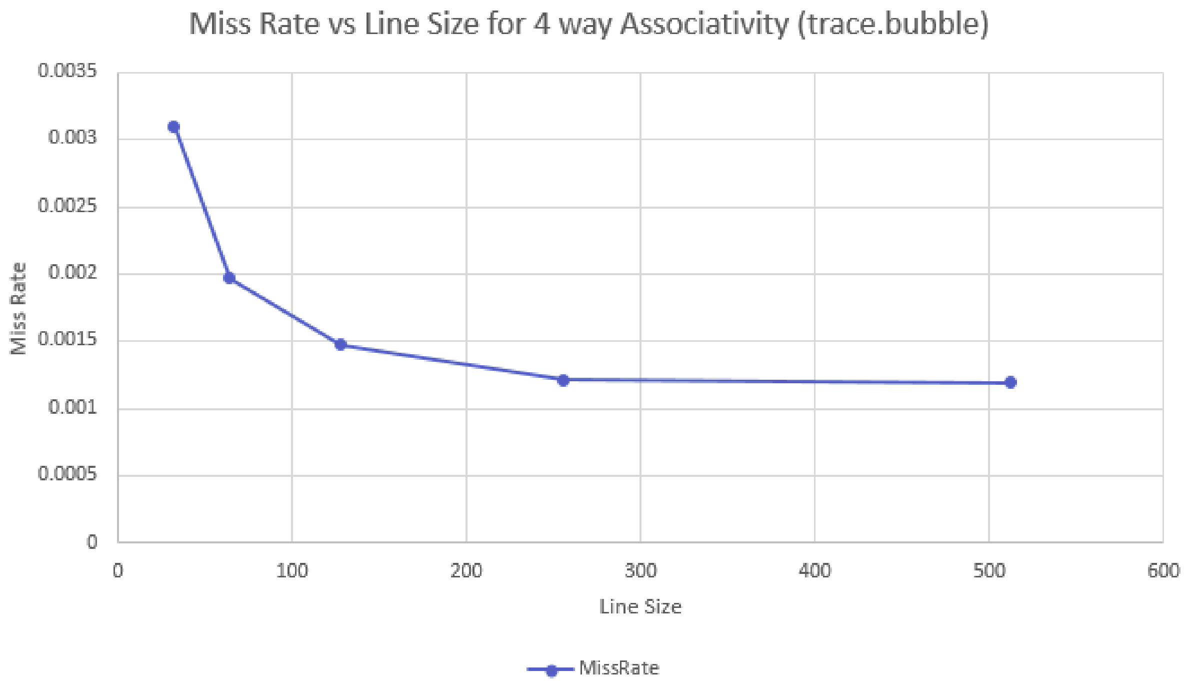
B.

Part1 Graphs

Here are the plots of the miss rate vs. the line size for line sizes of 32 bytes to 512 bytes with a fixed associativity of 4 for each trace.

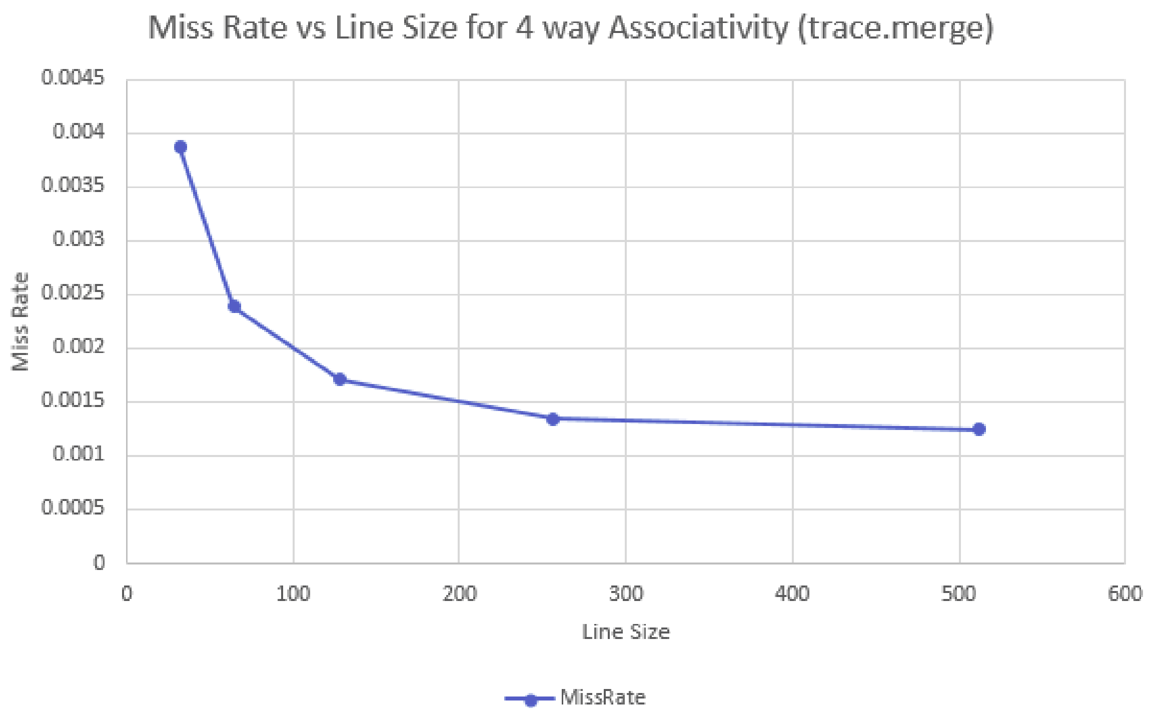
1. Data and plot for 4 way associativity using trace.bubble

Block Size	MissRate
32	0.003102805
64	0.001969839
128	0.001471765
256	0.001216321
512	0.001192754



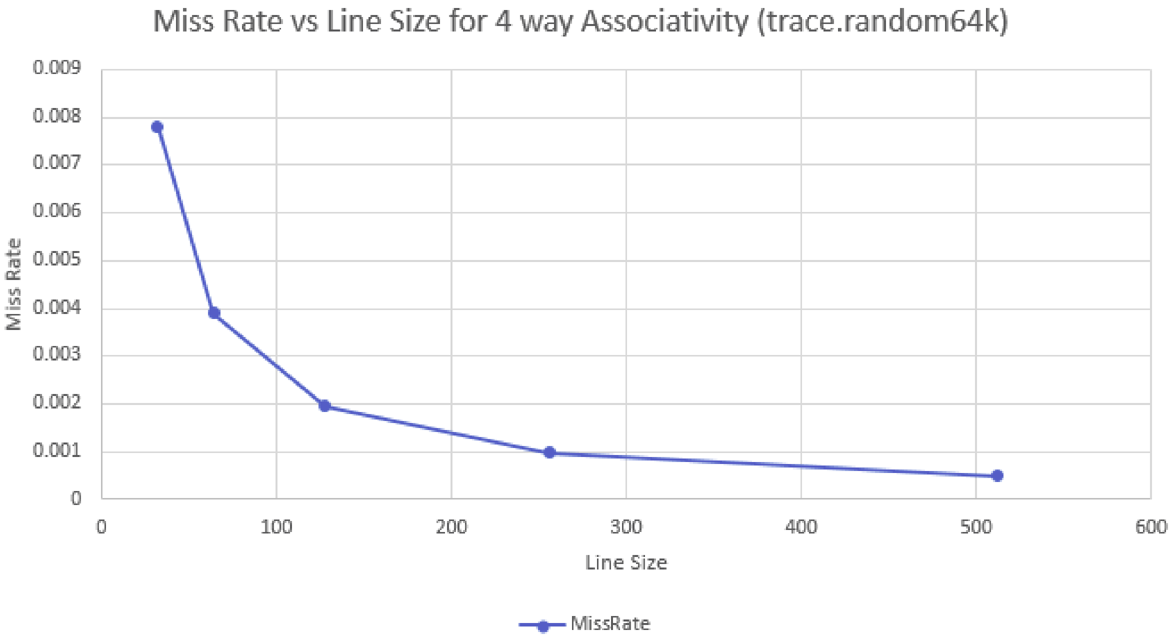
2. Data and plot for 4 way associativity using trace.merge

Block Size	MissRate
32	0.003868369
64	0.002393198
128	0.00171858
256	0.001345457
512	0.00125273



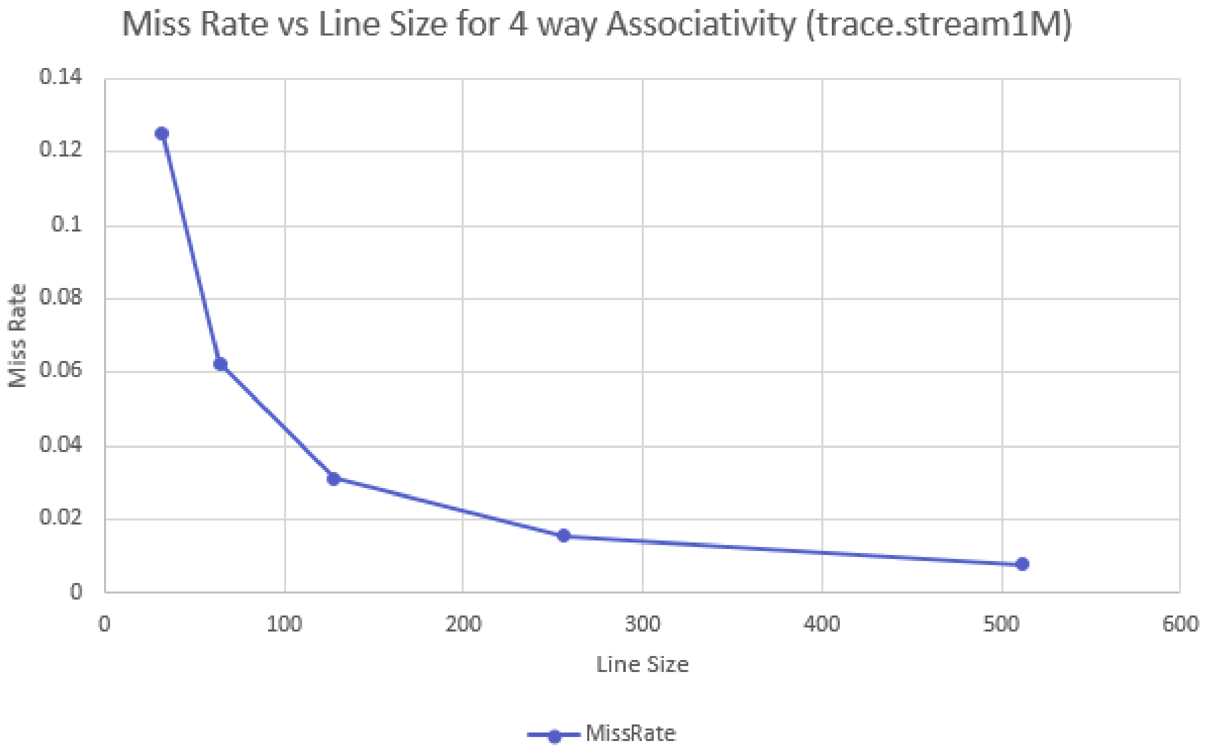
3. Data and plot for 4 way associativity using trace.random64k

Block Size	MissRate
32	0.0078125
64	0.00390625
128	0.001953125
256	0.000976563
512	0.000488281

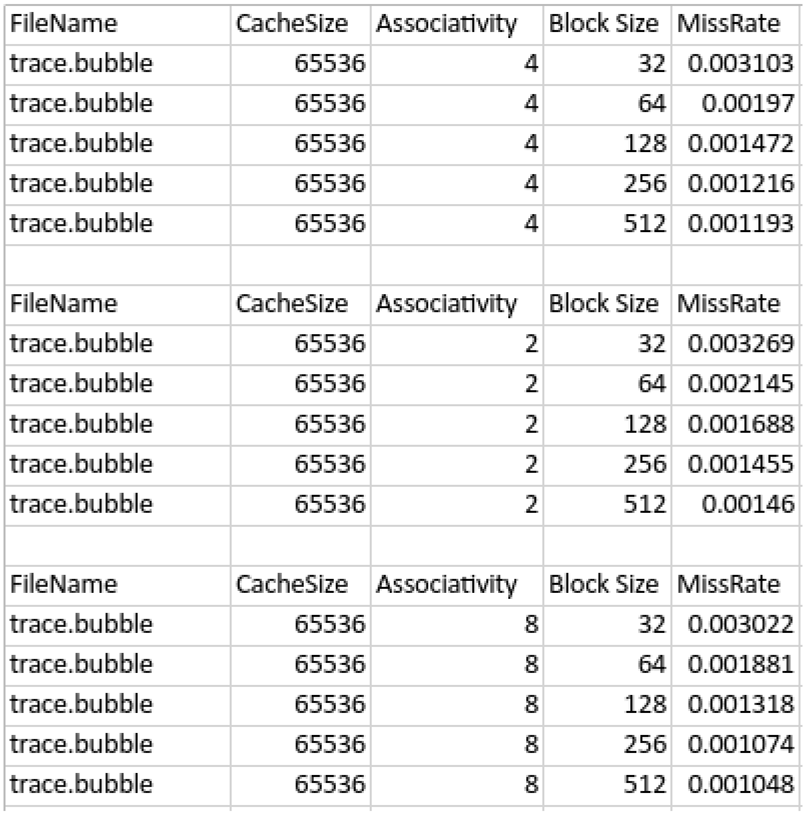


4. Data and plot for 4 way associativity using trace.stream1M

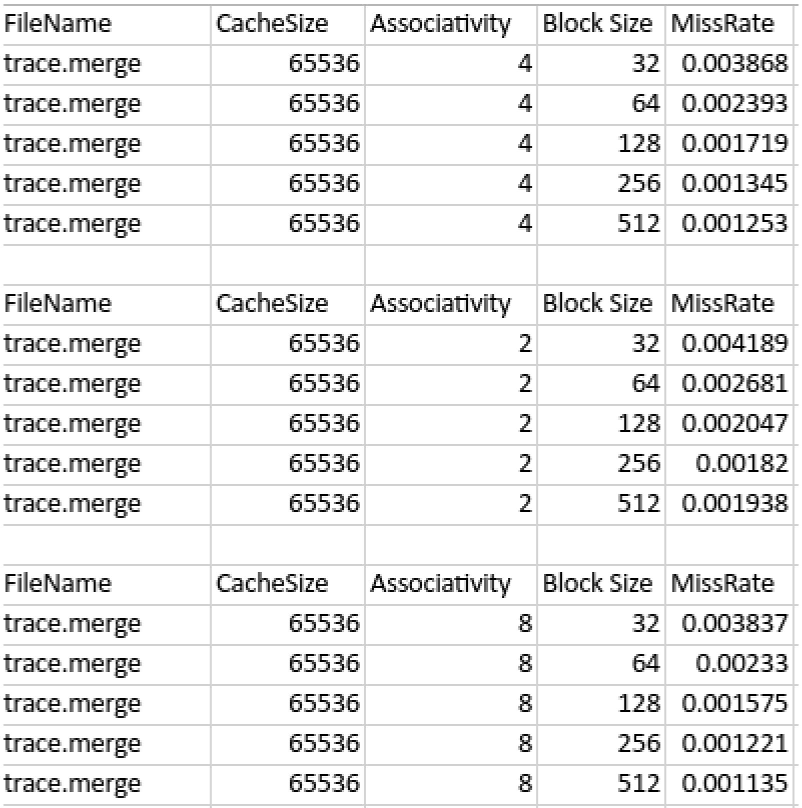
Block Size	MissRate
32	0.125
64	0.0625
128	0.03125
256	0.015625
512	0.0078125



Part2 Miss Rate Data for all traces using 64 KB cache size 2, 4, 8 way associativity and cache block size of 32, 64, 128, 256, 512 bytes.



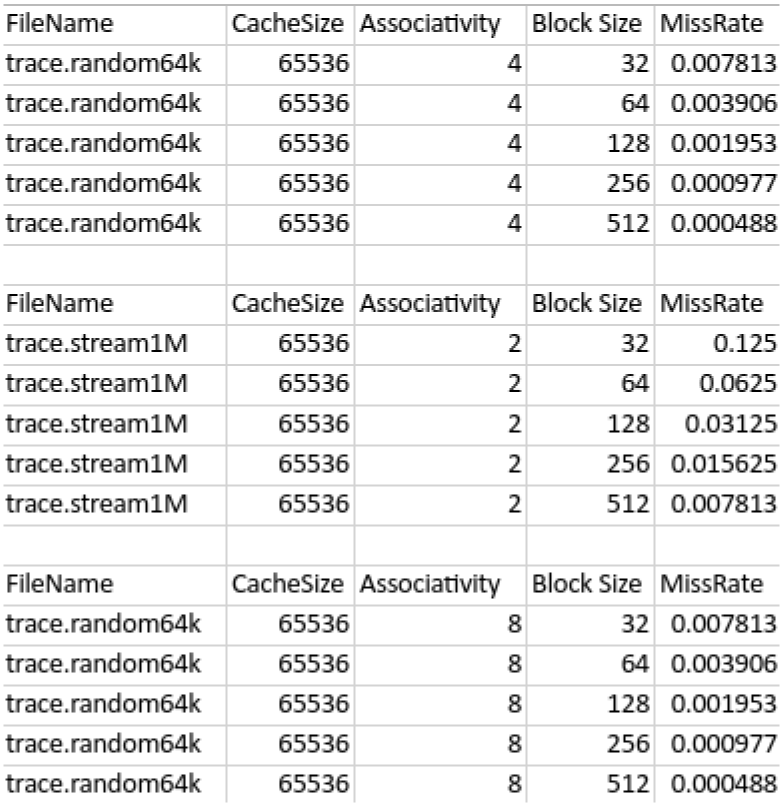
FileName	CacheSize	Associativity	Block Size	MissRate
trace.bubble	65536	4	32	0.003103
trace.bubble	65536	4	64	0.00197
trace.bubble	65536	4	128	0.001472
trace.bubble	65536	4	256	0.001216
trace.bubble	65536	4	512	0.001193
FileName	CacheSize	Associativity	Block Size	MissRate
trace.bubble	65536	2	32	0.003269
trace.bubble	65536	2	64	0.002145
trace.bubble	65536	2	128	0.001688
trace.bubble	65536	2	256	0.001455
trace.bubble	65536	2	512	0.00146
FileName	CacheSize	Associativity	Block Size	MissRate
trace.bubble	65536	8	32	0.003022
trace.bubble	65536	8	64	0.001881
trace.bubble	65536	8	128	0.001318
trace.bubble	65536	8	256	0.001074
trace.bubble	65536	8	512	0.001048



FileName	CacheSize	Associativity	Block Size	MissRate
trace.merge	65536	4	32	0.003868
trace.merge	65536	4	64	0.002393
trace.merge	65536	4	128	0.001719
trace.merge	65536	4	256	0.001345
trace.merge	65536	4	512	0.001253

FileName	CacheSize	Associativity	Block Size	MissRate
trace.merge	65536	2	32	0.004189
trace.merge	65536	2	64	0.002681
trace.merge	65536	2	128	0.002047
trace.merge	65536	2	256	0.00182
trace.merge	65536	2	512	0.001938

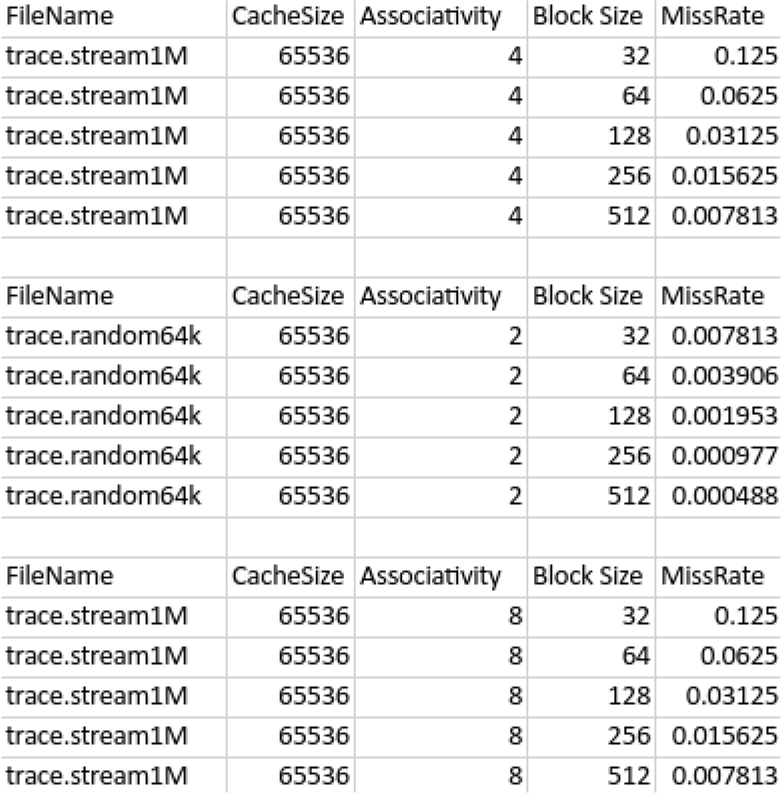
FileName	CacheSize	Associativity	Block Size	MissRate
trace.merge	65536	8	32	0.003837
trace.merge	65536	8	64	0.00233
trace.merge	65536	8	128	0.001575
trace.merge	65536	8	256	0.001221
trace.merge	65536	8	512	0.001135



FileName	CacheSize	Associativity	Block Size	MissRate
trace.random64k	65536	4	32	0.007813
trace.random64k	65536	4	64	0.003906
trace.random64k	65536	4	128	0.001953
trace.random64k	65536	4	256	0.000977
trace.random64k	65536	4	512	0.000488

FileName	CacheSize	Associativity	Block Size	MissRate
trace.stream1M	65536	2	32	0.125
trace.stream1M	65536	2	64	0.0625
trace.stream1M	65536	2	128	0.03125
trace.stream1M	65536	2	256	0.015625
trace.stream1M	65536	2	512	0.007813

FileName	CacheSize	Associativity	Block Size	MissRate
trace.random64k	65536	8	32	0.007813
trace.random64k	65536	8	64	0.003906
trace.random64k	65536	8	128	0.001953
trace.random64k	65536	8	256	0.000977
trace.random64k	65536	8	512	0.000488



FileName	CacheSize	Associativity	Block Size	MissRate
trace.stream1M	65536	4	32	0.125
trace.stream1M	65536	4	64	0.0625
trace.stream1M	65536	4	128	0.03125
trace.stream1M	65536	4	256	0.015625
trace.stream1M	65536	4	512	0.007813

FileName	CacheSize	Associativity	Block Size	MissRate
trace.random64k	65536	2	32	0.007813
trace.random64k	65536	2	64	0.003906
trace.random64k	65536	2	128	0.001953
trace.random64k	65536	2	256	0.000977
trace.random64k	65536	2	512	0.000488

FileName	CacheSize	Associativity	Block Size	MissRate
trace.stream1M	65536	8	32	0.125
trace.stream1M	65536	8	64	0.0625
trace.stream1M	65536	8	128	0.03125
trace.stream1M	65536	8	256	0.015625
trace.stream1M	65536	8	512	0.007813

Part2 Calculate Overall Miss Rate, Read Miss Rate, Write Miss Rate

Based on the Miss rates of associativity 2,4,8 with cache size of 64KB and cache block sizes 32, 64, 128, 256, 512 across all 4 trace files, I concluded that the best configuration that minimizes the sum of the overall miss rate across all traces is the 8 way association with 512 byte cache block size.

Now I calculated the Overall Miss Rate, Read Miss Rate and Write Miss Rate for 8 way association, 512 Byte cache block size, 64 KB cache size across all trace files. The write miss rate, for example, can be calculated by (Writebacks – No. Of Write Misses)/(Accesses – No. Of Writes)

FileName	No. Of Writes	No. of Write Misses	ReadMissRate	WriteMissRate	Overall Miss Rate
trace.bubble	656333	618	0.00017102	0.000941595	0.001048029
trace.merge	1133678	1026	0.000198327	0.000905019	0.001134607
trace.random64k	0	0	0.000488281	0	0.000488281
trace.stream1M	0	0	0.0078125	0	0.0078125

Part3 Calculate the volume of write-back traffic in bytes for each trace using the best overall configuration that minimizes the sum of the overall miss rate across all traces.

To calculate the write back traffic in bytes, you can use the formula: 512*Writebacks

FileName	WriteBack Traffic
trace.bubble	812544
trace.merge	1189888
trace.random64k	0
trace.stream1M	0

Part4 Compute the total memory access volume for each trace using the best overall configuration that minimizes the sum of the overall miss rate across all traces.

To compute the total memory access volume, you can use 512*Misses

FileName	Total Mem Access Volume
trace.bubble	3392512
trace.merge	4460544
trace.random64k	65536
trace.stream1M	1048576

Total NO. Of bytes referenced can be calculated by 512*Accesses

FileName	Total Number of Byte Referenced
trace.bubble	3237039616
trace.merge	3931356160
trace.random64k	134217728
trace.stream1M	134217728

Using Total No. Of Bytes refereced – Total Mem Access Volume, you can find the Maim memory access saved

FileName	Main Memory Access Saved
trace.bubble	3233647104
trace.merge	3926895616
trace.random64k	134152192
trace.stream1M	133169152