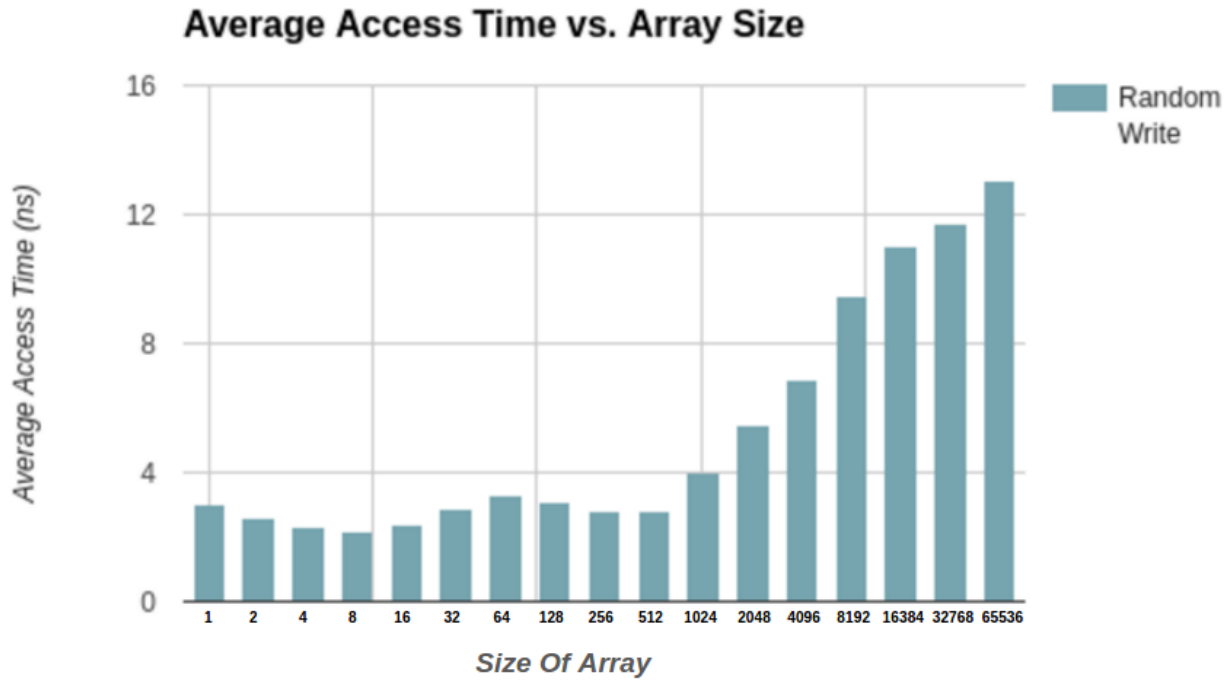
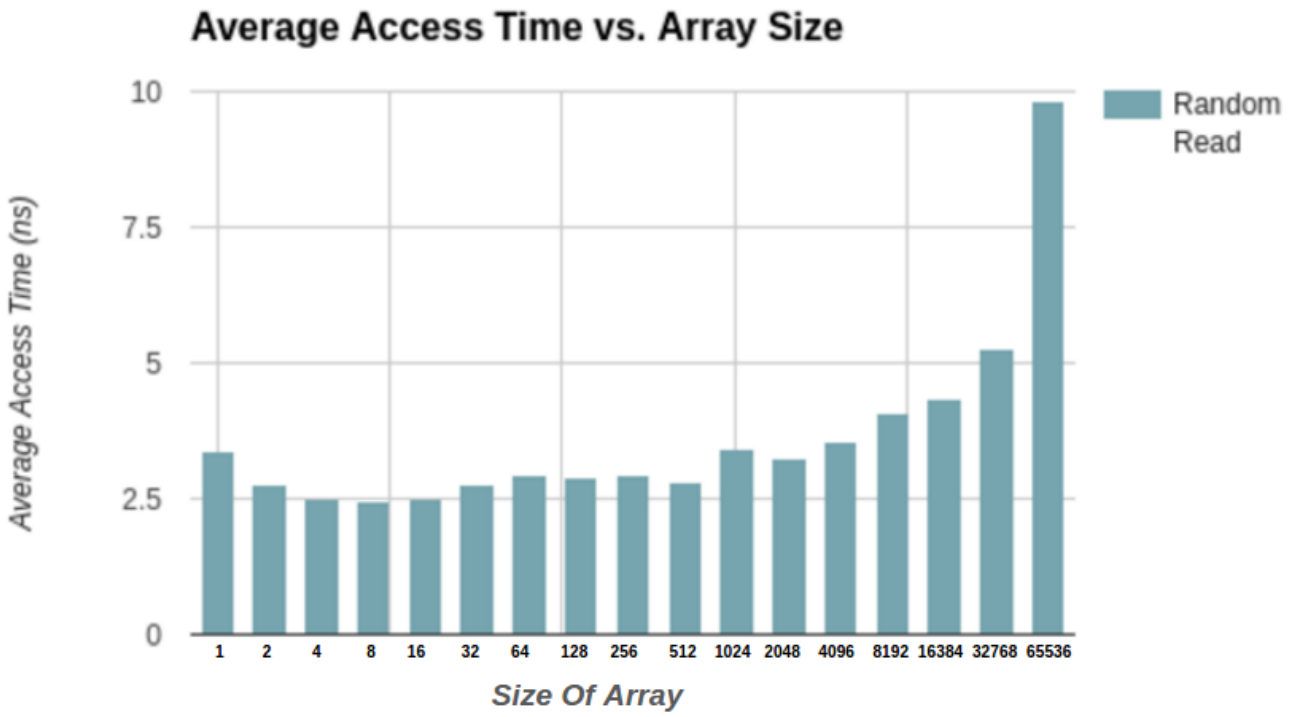
Task1:

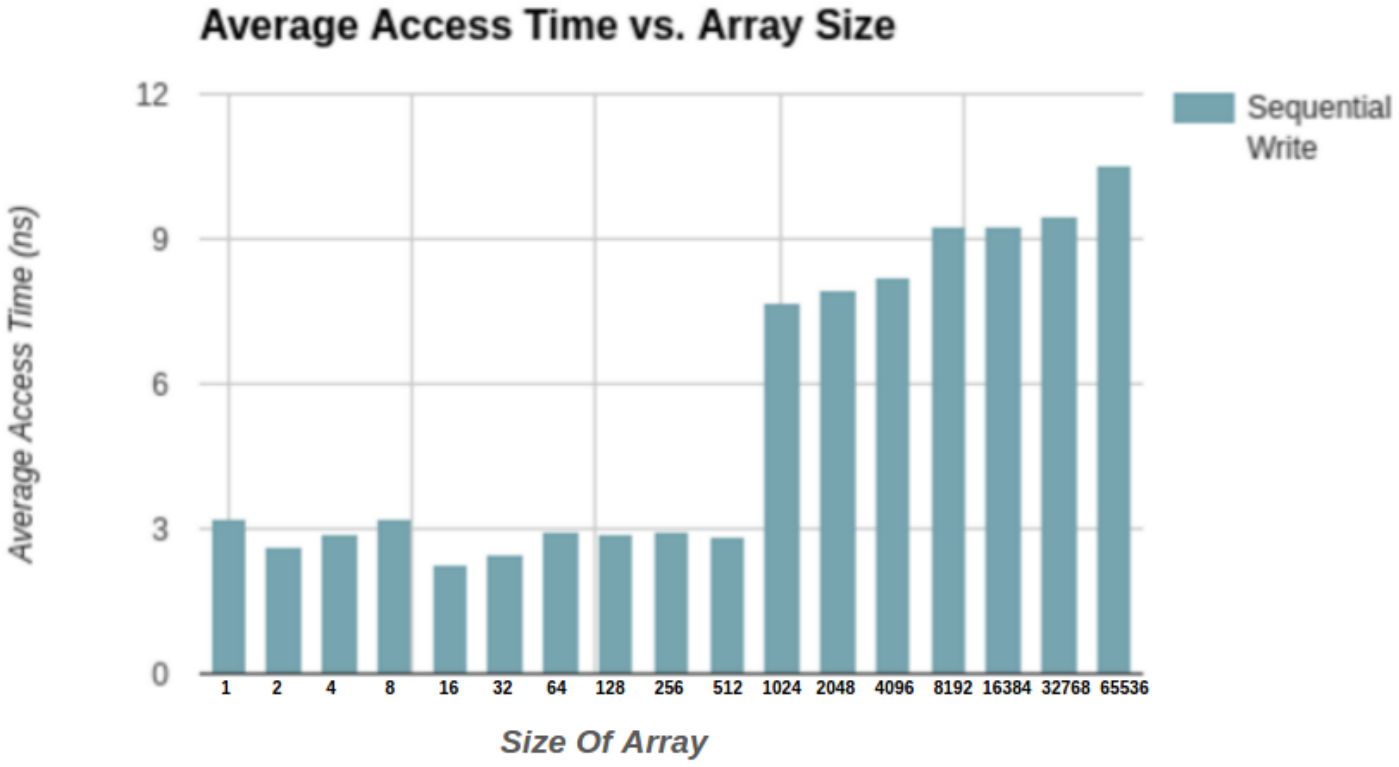
The average access time of memory reference in both read and write operations in sequential and random manner for different array sizes is as shown below in table:

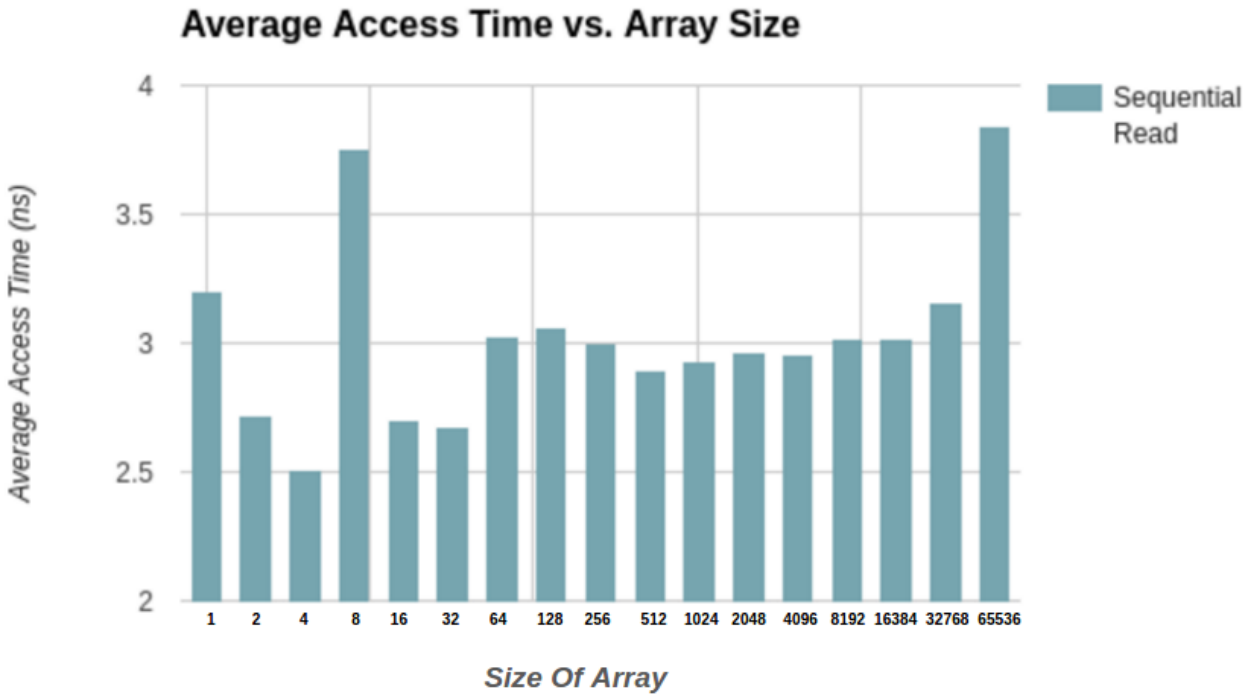
Array Size	Average Access Time (ns) - Random Write	Average Access Time (ns) - Random Read	Average Access Time (ns) - Sequential Write	Average Access Time (ns) - Sequential Read
1	3	3.36	3.2	3.2
2	2.63	2.76	2.62	2.72
4	2.335	2.51	2.915	2.51
8	2.205	2.445	3.2275	3.7575
16	2.3925	2.5125	2.24625	2.70375
32	2.85625	2.7825	2.485625	2.67125
64	3.274688	2.946563	2.964688	3.029375
128	3.054531	2.88125	2.883906	3.059687
256	2.814766	2.920234	2.931719	2.996328
512	2.774688	2.825156	2.822305	2.899023
1024	4.023223	3.429277	7.702676	2.93293
2048	5.470889	3.247187	7.950928	2.961885
4096	6.867192	3.537031	8.22125	2.953418
8192	9.470176	4.073376	9.274521	3.017766
16384	11.051451	4.354395	9.237157	3.01688
32768	11.709772	5.267745	9.499771	3.155983
65536	13.021858	9.845013	10.508946	3.845862

Graphs plotting Average Access Time vs Array Size for various read and write are as shown below:




Average Access Time vs. Array Size
Average Access Time (ns)
10
7.5
5
2.5
0
1 2 4 8 16 32 64 128 256 512 1024 2048 4096 8192 16384 32768 65536
Size Of Array
Random Read


Average Access Time vs. Array Size
Average Access Time (ns)
12
9
6
3
0
1 2 4 8 16 32 64 128 256 512 1024 2048 4096 8192 16384 32768 65536
Size Of Array
Sequential Write



a) The average access time measured represents the Latency of memory hierarchy as it measures the ***time*** taken to access a particular memory reference. The accuracy of measurement is improved by accessing the memory multiple times (i.e., having N iterations), because time taken for single access will have a very low access time and it will be quite difficult to differentiate from noise. So, iterating N times and calculating the average value will improve the performance.

b) Concept of Spatial Locality (use of data elements within relatively close storage locations) makes the access time different for linear and random access patterns. Random access pattern have very low spatial locality therefore a high access time and linear access pattern have very high spatial locality and low access time. The access time for write operation is greater than that for read operation. As write operation is limited by the size of write buffer in both write through and write back making write operation asynchronous.

c) As mentioned earlier, the memory access pattern is repeated for N iterations and from each iteration, we only collect the data from a sample instead of accessing the entire array, so as to considerably reduce the virtual to physical address mapping in TLB's. TLB is a cache in CPU that stores page table address in order to reduce the memory access time (accessing RAM for every single operation makes the CPU drastically slow). TLB stores recently accessed page table entries exploiting the locality. So, instead of accessing all the elements in the array, we only choose a single sample thereby eliminating the TLB overhead.