

# Architecture assignment Report

This report summarizes the execution times and speedups achieved by loop unrolling with degrees of 4, 8, 16, and 64 across three different array sizes.

## Unrolling Time Table

Array Size	Normal Time	Unrolled-4 Time	Unrolled-8 Time	Unrolled-16 Time	Unrolled-64 Time
1024	0.000003	0.000002	0.000002	0.000004	0.000005
1024*1024	0.002590	0.001570	0.001541	0.001515	0.002636
1024*1024*16	0.017450	0.012979	0.012979	0.028906	0.033658

## Unrolling Speedup Table

Array Size	normal	Unrolled-4 Speedup	Unrolled-8 Speedup	Unrolled-16 Speedup	Unrolled-64 Speedup
1024	1	1.500000	1.500000	0.750000	0.600000
1024*1024	1	1.649682	1.680727	0.850016	0.982549
1024*1024*16	1	1.335630	1.344480	0.603681	0.518450

## Observations

### 1. Execution Time:

- Loop unrolling significantly reduces execution time for smaller array sizes and lower degrees of unrolling (4 and 8).
- Larger array sizes and higher degrees of unrolling (16 and 64) sometimes increase execution time due to overheads.

### 2. Speedup:

- Speedup is most noticeable with unrolling degrees of 4 and 8, especially for larger arrays.
- Higher degrees of unrolling (16 and 64) show reduced speedups or even slowdowns due to diminishing returns and potential cache inefficiencies.

### 3. Optimal Configuration:

- Unrolling with degrees of 4 or 8 achieves the best balance between speedup and performance consistency across different array sizes.

#### Explain the way of solving :

I have used the wsl ubuntu to compile the code with disabling the optimization and I have compile it a lot of time and take the average and record it in the result tables as show above , this picture example of code compilation using ubuntu :

```
anwar@DESKTOP-OUT68RN: ~  
anwar@DESKTOP-OUT68RN:~$ gcc -O0 secsize.c -o secsize  
anwar@DESKTOP-OUT68RN:~$ ./secsize  
size = 1048576  
normal Time: 0.002315  
unrolling_4 Time: 0.001600  
Speedup (4-way unrolling): 1.446875  
unrolling_8 Time: 0.001511  
Speedup (8-way unrolling): 1.532098  
unrolling_16 Time: 0.002626  
Speedup (16-way unrolling): 0.881569  
unrolling_64 Time: 0.002251  
Speedup (64-way unrolling): 1.028432  
anwar@DESKTOP-OUT68RN:~$ gcc -O0 secsize.c -o secsize  
anwar@DESKTOP-OUT68RN:~$ ./secsize  
size = 1048576  
normal Time: 0.002371  
unrolling_4 Time: 0.001602  
Speedup (4-way unrolling): 1.480025  
unrolling_8 Time: 0.001698  
Speedup (8-way unrolling): 1.396349  
unrolling_16 Time: 0.003617  
Speedup (16-way unrolling): 0.655516  
unrolling_64 Time: 0.002710  
Speedup (64-way unrolling): 0.874908  
anwar@DESKTOP-OUT68RN:~$ gcc -O0 secsize.c -o secsize  
anwar@DESKTOP-OUT68RN:~$ ./secsize  
size = 1048576  
normal Time: 0.002175  
unrolling_4 Time: 0.001488  
Speedup (4-way unrolling): 1.461694  
unrolling_8 Time: 0.001284  
Speedup (8-way unrolling): 1.693925  
unrolling_16 Time: 0.002261  
Speedup (16-way unrolling): 0.961964  
unrolling_64 Time: 0.002228  
Speedup (64-way unrolling): 0.976212  
anwar@DESKTOP-OUT68RN:~$ gcc -O0 secsize.c -o secsize  
anwar@DESKTOP-OUT68RN:~$ ./secsize  
size = 1048576  
normal Time: 0.002590  
unrolling_4 Time: 0.001570  
Speedup (4-way unrolling): 1.649682  
unrolling_8 Time: 0.001541  
Speedup (8-way unrolling): 1.680727  
unrolling_16 Time: 0.003047  
Speedup (16-way unrolling): 0.850016  
unrolling_64 Time: 0.002636  
Speedup (64-way unrolling): 0.982549  
anwar@DESKTOP-OUT68RN:~$
```

**"I did like this for each array size"**

```
anwar@DESKTOP-OUT68RN: ~  
anwar@DESKTOP-OUT68RN:~$ gcc -O0 lastsize.c -o lastsize  
anwar@DESKTOP-OUT68RN:~$ ./lastsize  
size = 16777216  
normal Time: 0.016907  
unrolling_4 Time: 0.012808  
Speedup (4-way unrolling): 1.320034  
unrolling_8 Time: 0.012671  
Speedup (8-way unrolling): 1.334307  
unrolling_16 Time: 0.028148  
Speedup (16-way unrolling): 0.600647  
unrolling_64 Time: 0.034136  
Speedup (64-way unrolling): 0.495284  
anwar@DESKTOP-OUT68RN:~$ gcc -O0 lastsize.c -o lastsize  
anwar@DESKTOP-OUT68RN:~$ ./lastsize  
size = 16777216  
normal Time: 0.017337  
unrolling_4 Time: 0.013057  
Speedup (4-way unrolling): 1.327794  
unrolling_8 Time: 0.013053  
Speedup (8-way unrolling): 1.328200  
unrolling_16 Time: 0.029310  
Speedup (16-way unrolling): 0.591505  
unrolling_64 Time: 0.034200  
Speedup (64-way unrolling): 0.506930  
anwar@DESKTOP-OUT68RN:~$ gcc -O0 lastsize.c -o lastsize  
anwar@DESKTOP-OUT68RN:~$ ./lastsize  
size = 16777216  
normal Time: 0.018763  
unrolling_4 Time: 0.013568  
Speedup (4-way unrolling): 1.382886  
unrolling_8 Time: 0.014044  
Speedup (8-way unrolling): 1.336015  
unrolling_16 Time: 0.031189  
Speedup (16-way unrolling): 0.601590  
unrolling_64 Time: 0.035921  
Speedup (64-way unrolling): 0.522341  
anwar@DESKTOP-OUT68RN:~$ gcc -O0 lastsize.c -o lastsize  
anwar@DESKTOP-OUT68RN:~$ ./lastsize  
size = 16777216  
normal Time: 0.017450  
unrolling_4 Time: 0.013065  
Speedup (4-way unrolling): 1.335630  
unrolling_8 Time: 0.012979  
Speedup (8-way unrolling): 1.344480  
unrolling_16 Time: 0.028906  
Speedup (16-way unrolling): 0.603681  
unrolling_64 Time: 0.033658  
Speedup (64-way unrolling): 0.518450  
anwar@DESKTOP-OUT68RN:~$ █
```

**I well put the code in text file with submission.**

**Name :Anwar Baker**

**Number :12113661**