

# Multi Core Architecture

## Lab Assignment 5

### GPU and CUDA 2

Group members,

Shriraam Mohan <[shriraam.mohan@mailbox.tu-berlin.de](mailto:shriraam.mohan@mailbox.tu-berlin.de)>

Tamilselvan Shanmugam <[tamilselvan@mailbox.tu-berlin.de](mailto:tamilselvan@mailbox.tu-berlin.de)>

Due date: 13 Feb 2015, Thursday

## Design decisions:

### #1: 32x threads per block

GPGPU SIMD width is 32. GPU packs the block in multiples of 32 threads. To exploit all the SIMD units, we have organized our block size to be multiples of 32.

### #2: Grid size of 8x8

From the first assignment, we observed that the optimal grid size is 8x8. So, maintaining the grid size 8x8 will give the best performance. In four letter password cracker, we couldn't stick to 8x8 grid dimension, because of the gpgpusim limitations of threads per block. So we were forced to change the dimension. But, setting 8x8 grid size to our application will further boost the performance.

### #3: Loop unrolling

Control flow can affect the performance drastically. We have unrolled the for loops in the password cracker application using `#pragma unroll`.

### #4: Function in-lining

Same as above. Forced the function calls to be *in-line* to maintain the control flow. Syntax is `__forceinline__`

### #5: memcpy

Written memcpy by hand to avoid overhead involved in function call and memory copy process.

### #6: shift instead of mul:

We have replaced multiplication operation with shift operation in MD5 calculation. This will potentially save some cycles.

## Run time:

	CPU version (cycles)	GPU version (cycles)	GPU version # Instructions	GPU version improvement (# times faster)
3 letter	59286761	39320	18330370	1507
4 letter	1466718389	825810	487168748	1776
6 letter	981082582902			

## Passwords

3 letter	4 letter
1. abc	1. blue
2. aaa	2. fire
3. xyz	3. tong
4. bla	4. dark
5. zzz	5. cool
6. cpu	6. qian
7. mcs	7. temp
8. aes	8. qwer
9. cpp	9. pass
10. gpu	10. test

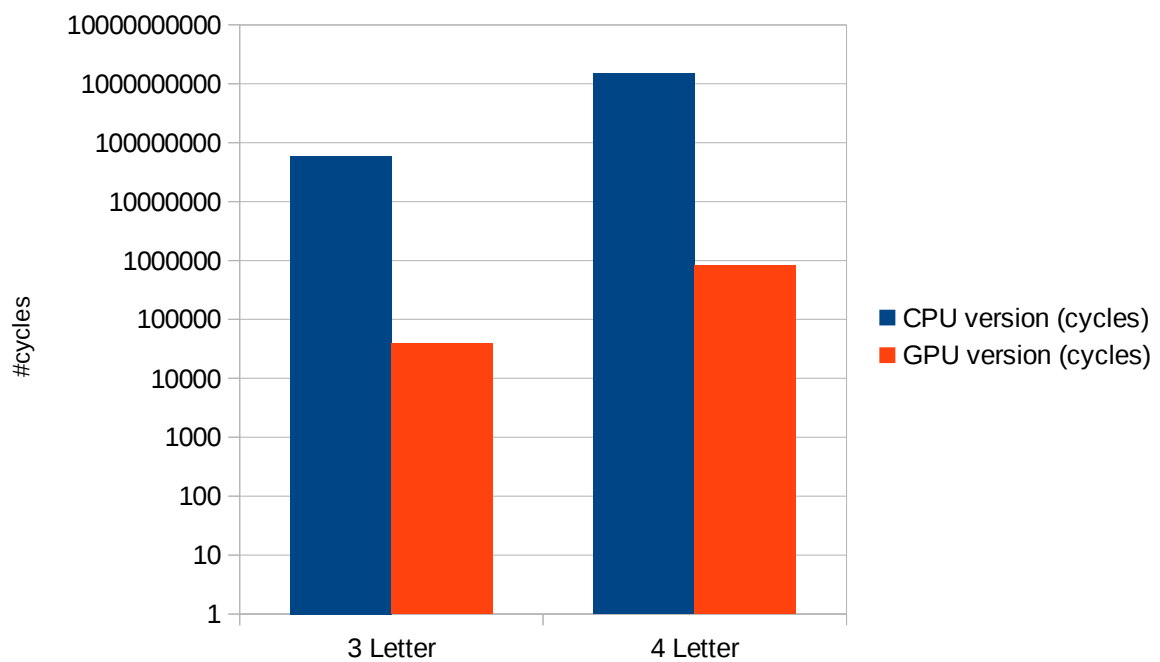


Figure.1. Comparison of CPU cycles vs GPU cycles in log scale

## Commands to run our application:

```
nvcc -arch sm_11 -o out md5gpu.cu
```

```
nvcc -ptx -arch sm_11 md5gpu.cu
```

P.S: The simulation for 6 letter password cracker is running from the afternoon till I'm writing this report. We hope it gives satisfactory results in real GPU.