

SESSION: 2018/19 DIET 1

Module Title: Games Programming 3

Module Code: MHI622946

Level: 4

Module Leader: Dr. Richard Holden

Individual Marks Summary:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Component | A: Environment set-up | B: Game Coding | C: Persistence | D: Report |
| Component % | 5 | 40 | 15 | 40 |

|  |  |  |
| --- | --- | --- |
| Student Name | Student ID | Student E-mail |
|  |  |  |

Student Declaration

This piece of work is not plagiarized. It is my own original work and has not been submitted elsewhere in fulfilment of the requirements of this or any other award.

Signature: Date:

Contents

[Moore’s Law 3](#_Toc531951954)

[Multi-Core Technology 3](#_Toc531951955)

[Parallelism 3](#_Toc531951956)

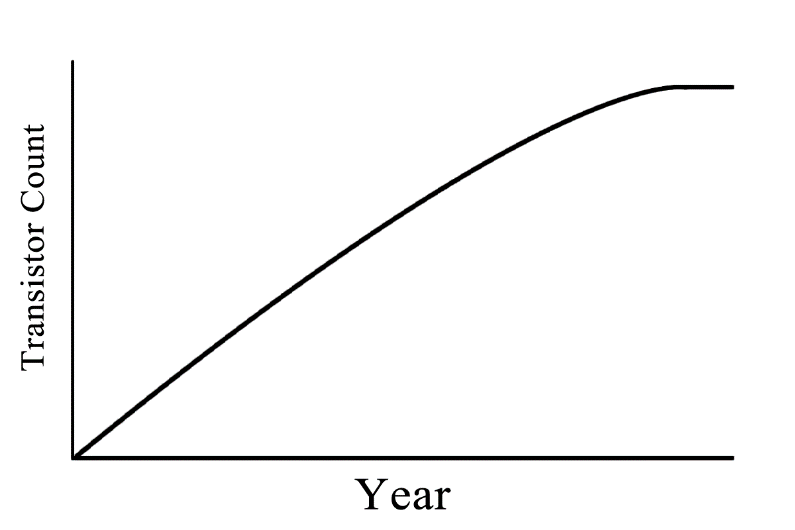
[Multi-Threading 3](#_Toc531951957)

[References 4](#_Toc531951958)

# Moore’s Law

Moore’s Law (ML) is the observation that the transistor quantity and the performance of integrated circuits doubles every year (Moore, 1995; Schaller, 1997; Thompson e Parthasarathy, 2006; Mack, 2011). This is due to the shrinking of transistors, which enabled chips to be fitted with twice as many transistors each year.

## End of Moore’s Law

However, the effect of ML appears to be slowly coming to a halt, as chip transistor capacity comes to a plateau. This issue arises due to transistor sizes reaching such microscopic sizes, that shrinkage can no longer be facilitated. One study by (Kish, 2002) claims that the limitation of transistor density is caused by the increasing challenge of supplying the correct voltage to transistors of such diminishing sizes. In order to match the pace of advancing game hardware requirements, major chip manufacturers, such as AMD and Intel, have taken the approach of implementing dual and multi core technology to their products (Tulip *et al.*, 2006).

# Multi-Core Technology

Multi-Core technology represents innovative advancements such as multi-core processors, which are defined as a singular processor containing two or more execution cores. Interest in multi-core technology heightened as transistor counts grew to the billions, resulting in much greater power requirements. By segmenting chips into multiple cores, groups of transistors can be assigned their own computations. These cores can execute tasks in parallel to other cores on the chip, resulting in lower power requirements. As stated by (Gepner e Kowalik, 2006), multi-core technology is an important innovation as it drives forward multi-threading and parallelism. Through this advancement, multi-core technology can be exploited to split computational loads, thus completing tasks with more efficiency.

## Parallelism

Parallelism involves the execution of multiple tasks in a computer system simultaneously, this works in contradiction to sequential programming, which incorporates the execution of tasks in an arranged order. Despite modern computers now shipping with multiple processors, often the system cannot fully take advantage of the hardware due to restrictions in the design of software. As presented by (Olszewski *et al.*, 2009) , multithreaded programs which provide shared memory provide interweaved accesses to shared data. (Ranger *et al.*, 2007) highlights the need for software to be programmed parallelly to exploit multi-core processors, as they become more common.

### Task Parallelism and Data Parallelism

Task parallelism is a form of parallelism focused on executing different operations in a parallel fashion to fully exploit the processing unit resources. Task parallelism can be executed synchronously or asynchronously. Synchronized task parallelism oversees tasks finishing in a singular clock cycle. The asynchronous model allows tasks to run and complete at their own pace.

Data parallelism focuses on performing the same operation, split across multiple cores, with a shared data source. This is beneficial as an operation can be executed by multiple cores simultaneously, which reduces the time required to complete, providing efficiency for both the developer and end-user.

# Multi-Threading

Multi-threading is a computation model which allows the existence of many threads to execute a single process. This can help to break singular tasks down into smaller subtasks, which is independently executed by each separate thread. Although threads may share resources of the process, referencing data which is being altered by another thread can cause issues. Therefore, clear design of the software’s structure critical.

# Relevance to Game Development

## Advantages to multi-threading in Game Development

Applying these methods and approached to the development of computer games, can bring huge benefits to both the developer and end-user. Developing a multithreaded game can bring significant performance improvements, while also enhancing the contents of the game. By enabling multithreading, a feature such as particle effects can be implemented to be rendered across multiple cores. This enables the particles to be rendered in a much more efficient manner, resulting in higher, more stable frame rates. Scalable parallelism allows the distribution of a task across the available cores of a system, which provides increased performance for users with multicore processors. Taking the particle effect example, users with high-end hardware will have an improved experience, while single-core processors can bypass these rendering complexities which their hardware may not facilitate without being stretched.

A recent study by (Asaduzzaman *et al.*, 2014), discovered that a multithreaded synchronous game engine operated more than 14x faster than it’s single-threaded counterpart, in a game with multiple physics based objects and collision detection. Their findings originated from using a multithreaded synchronous model which incorporated data parallelism process collision detections. These findings justify a need for parallel programming and multithreading in modern games titles which exponentially look to replicate the physics and nature of the real world.

## Implementing multi-threading in games

Using mutex to allow threads to share resources in a non-simultaneous manner

Task parallelism indepdendence

# Conclusion

Why we are for multithreading

Challenges it brings

Note

<https://stackoverflow.com/questions/469092/how-independent-are-threads-inside-the-same-process>

# References

ASADUZZAMAN, A.; LEE, H. Y.; GUMMADI, D. Impact of Thread Synchronization and Data Parallelism on Multicore Game Programming. Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA), 2014, The Steering Committee of The World Congress in Computer Science, Computer …. p.1.

GEPNER, P.; KOWALIK, M. F. Multi-core processors: New way to achieve high system performance. Parallel Computing in Electrical Engineering, 2006. PAR ELEC 2006. International Symposium on, 2006, IEEE. p.9-13.

KISH, L. B. End of Moore's law: thermal (noise) death of integration in micro and nano electronics. **Physics Letters A,** v. 305, n. 3-4, p. 144-149, 2002. ISSN 0375-9601.

MACK, C. A. Fifty years of Moore's law. **IEEE Transactions on semiconductor manufacturing,** v. 24, n. 2, p. 202-207, 2011. ISSN 0894-6507.

MOORE, G. E. Lithography and the future of Moore's law. Integrated Circuit Metrology, Inspection, and Process Control IX, 1995, International Society for Optics and Photonics. p.2-18.

OLSZEWSKI, M.; ANSEL, J.; AMARASINGHE, S. Kendo: efficient deterministic multithreading in software. **ACM Sigplan Notices,** v. 44, n. 3, p. 97-108, 2009. ISSN 1605584061.

RANGER, C. et al. Evaluating mapreduce for multi-core and multiprocessor systems. High Performance Computer Architecture, 2007. HPCA 2007. IEEE 13th International Symposium on, 2007, Ieee. p.13-24.

SCHALLER, R. R. Moore's law: past, present and future. **IEEE spectrum,** v. 34, n. 6, p. 52-59, 1997. ISSN 0018-9235.

THOMPSON, S. E.; PARTHASARATHY, S. Moore's law: the future of Si microelectronics. **Materials today,** v. 9, n. 6, p. 20-25, 2006. ISSN 1369-7021.

TULIP, J.; BEKKEMA, J.; NESBITT, K. Multi-threaded game engine design. Proceedings of the 3rd Australasian conference on Interactive entertainment, 2006, Murdoch University. p.9-14.