C-DAC Four Days Technology Workshop

ON

Hybrid Computing – Coprocessors/Accelerators
Power-Aware Computing – Performance of Applications Kernels

hyPACK-2013
(Mode-1:Multi-Core)

Lecture Topic:

Multi-Core Processors : Multi-Core Architecture Part-I

Venue : CMSD, UoHYD ; Date : October 15-18, 2013

An Overview of threading

Lecture Outline

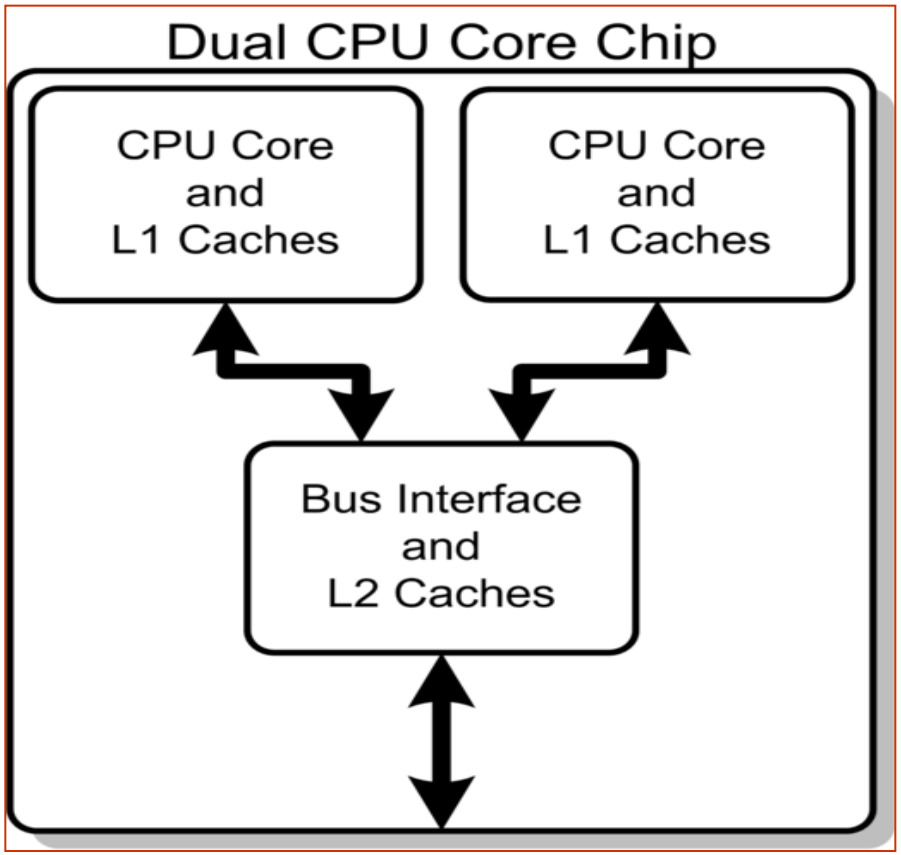
Following Topics will be discussed

- Introduction
- Understanding of Intel Multi-Core Architectures
- Multithreading, Superthreading & HyperThreading
- Case Studies & Examples

Dual-Core Processor

Conceptual diagram of

- A dual-core CPU, with
- CPU-local Level 1 caches, and
- Shared, on-chip Level 2 caches

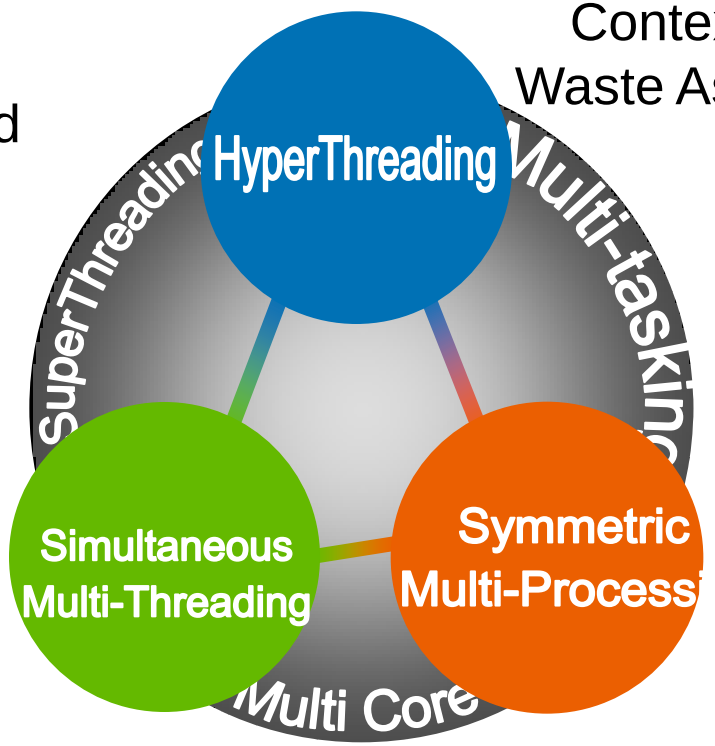


An Overview of threading

SMP and Cluster Platforms based on

Industry Standard Servers

Single Threaded CPU
Preemptive vs. co-operative Multitasking
Context, process and Thread
Waste Associated with Threads
Time Slicing
I/O Threads



Implementing Hyper-threading
- Caching & SMT

Implementing Hyper-threading
- Replicated
- Partitioned
- Shared

Source : http://www.intel.com ; htpp://www.amd.com; Reference : [6], 29], [31]

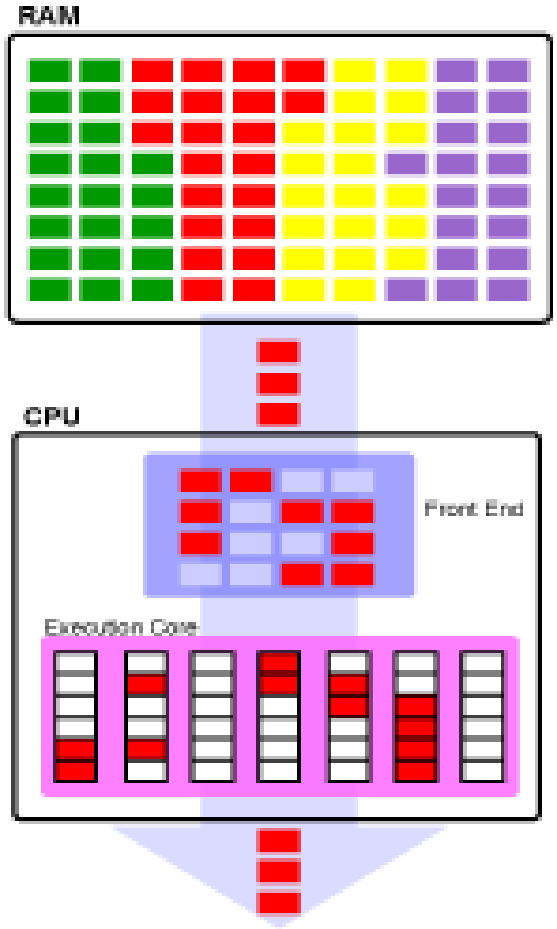
Multi-threading

- Understand more about multi-threading
- Understand more about multi-processing system
- Understand Hyper-threading
- The concept of **Simultaneous Multi-Threading** (**SMT**) – known as "**Hyper-threading**")
  - Multi-Tasking – Earlier Generation CPU's handle **Multi-tasking**
  - **SMT** – allows CPU to run more than one program at the same time

Conventional Multi-threading

❖ Out of Order Execution

- Not execute sequentially
- Reschedule the instructions
- Arrange them back after execution
- Looks as if an ordered sequential stream of instruction is executing
- Only CPU knows about execution order
- Still execution block is not fully utilized



❖ Time Slicing

❖ Empty Pipe-line stages; CPU can issue upto to four instructions per clock cycle.

Source : http://www.intel.com

Reference : [6], [29], [31]

Co-operative Multitasking

❖ Co-operative Multi-tasking

- Running programs to co-operate with each other and with OS in order to share the CPU among themselves in a fair and equitable manner.
  - Multiple program are loaded into memory
  - Both h/w and OS are involved
  - OS rapidly switch between running programs within a time frame(time slice).
  - Some time job with long execution time suffer from starvation
  - Still some execution block is utilized
- Designated Time Slice for execution of Program
- Problems : Some programs - Monopolize the CPU – System Grid to halt

Pre-emptive Multitasking

❖ Pre-emptive Multi-tasking

- Strictly enforced rules and kicks each program of the CPU once its time slice is up.
  - Memory Protection – OS makes sure that each program makes use the memory space allocated to it and it alone.
  - Instructions are fetched, decoded, and re-ordered & executed.
- Program is walled-off/Only one program on the system

Co-operative Multi-tasking & Pre-emptive Multi-tasking

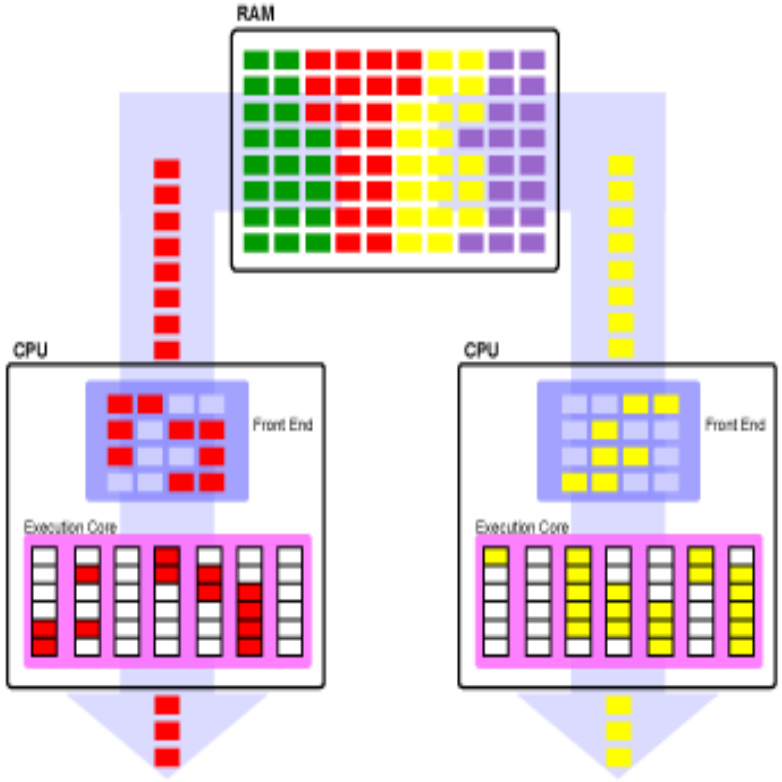
- Context Switches : currently executing process's context, flushing the CPU, and loading he next process's context
- A context Switch for a full-fledged multi-threaded process will obviously take a lot longer than a context switch for an individual thread within a process.
- Hardware support
- Wasting a number of CPU cycles
- Concept of Thread switches

An Overview of threading

- Issues & Challenges
  - Process, context and thread
  - Multi-tasking: Each program has a mind of its own
    - Pipeline bubbles
    - Number of Instructions per clock cycle
  - Time slicing – up-restore- context switches
  - The concept of “State” – Halting /Restoring
  - Wasting number of number of CPU cycles…
  - Execution efficiency improves ?

Symmetric Multiprocessors (SMPs) : Issues

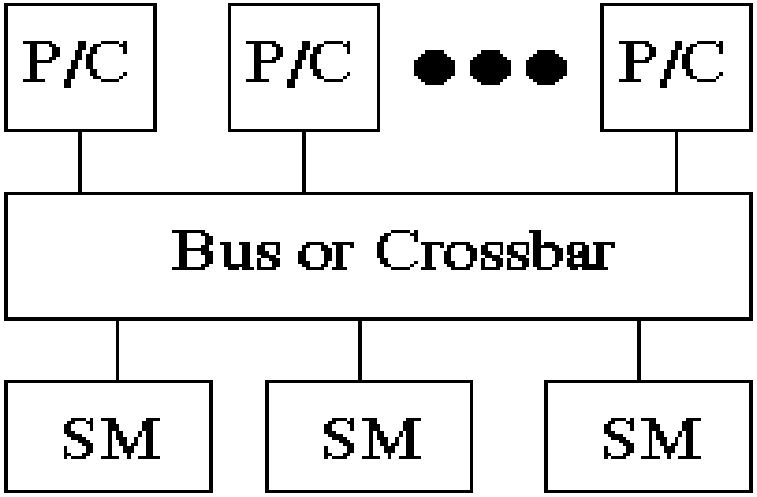
- Two processors is involved.
- OS schedule two processes for execution by either CPU
- Not allowed to monopolize
- Less waiting time
- Number of empty execution slots doubled
- Efficiency - No improvement in CPU utilization
- Time Slicing Issues



Reference : [6], [29], [31]

Symmetric Multiprocessors (SMPs) : Issues

(Contd…)



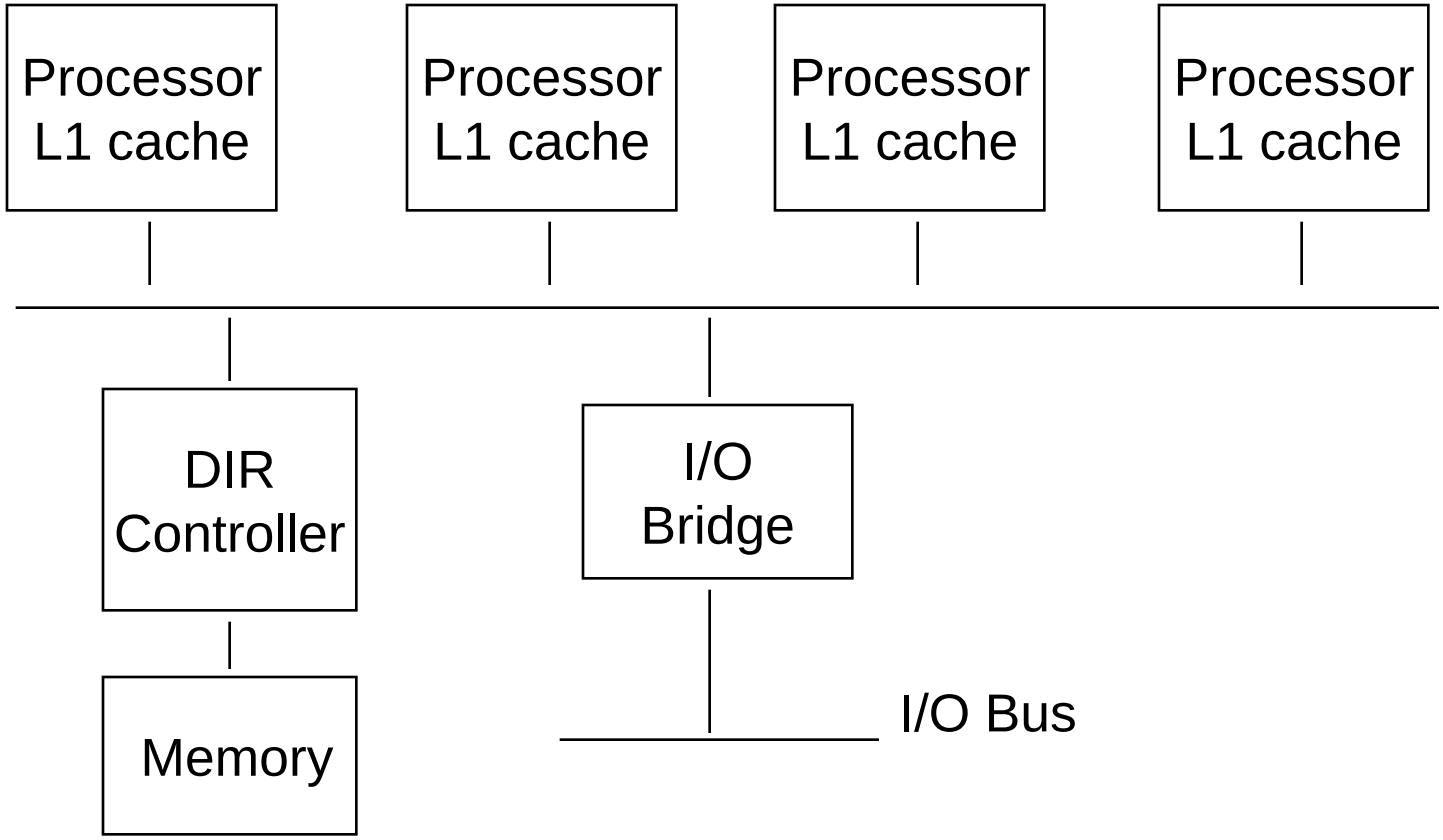
P/C : Microprocessor and cache; SM : Shared memory

- Uses commodity microprocessors with on-chip and off-chip caches.
- Processors are connected to a shared memory through a high-speed snoopy bus

Symmetric Multiprocessors (SMPs) : Issues

(Contd…)

All processors see same image of all system resources



Symmetric Multiprocessors (SMPs) : Issues (Contd…)

- On Some SMPs, a crossbar switch is used in addition to the bus.
- Scalable upto:
  - 4-8 processors (non-back planed based)
  - few tens of processors (back plane based)
- Equal priority for all processors (except for master or boot CPU)
- Memory coherency maintained by HW
- Multiple I/O Buses for greater Input Output

Symmetric Multiprocessors (SMPs) : Issues

(Contd…)

- Bus based architecture :
  - Inadequate beyond 8-16 processors
- Crossbar based architecture
  - multistage approach considering I/Os required in hardware
- Clock distribution and HF design issues for backplanes

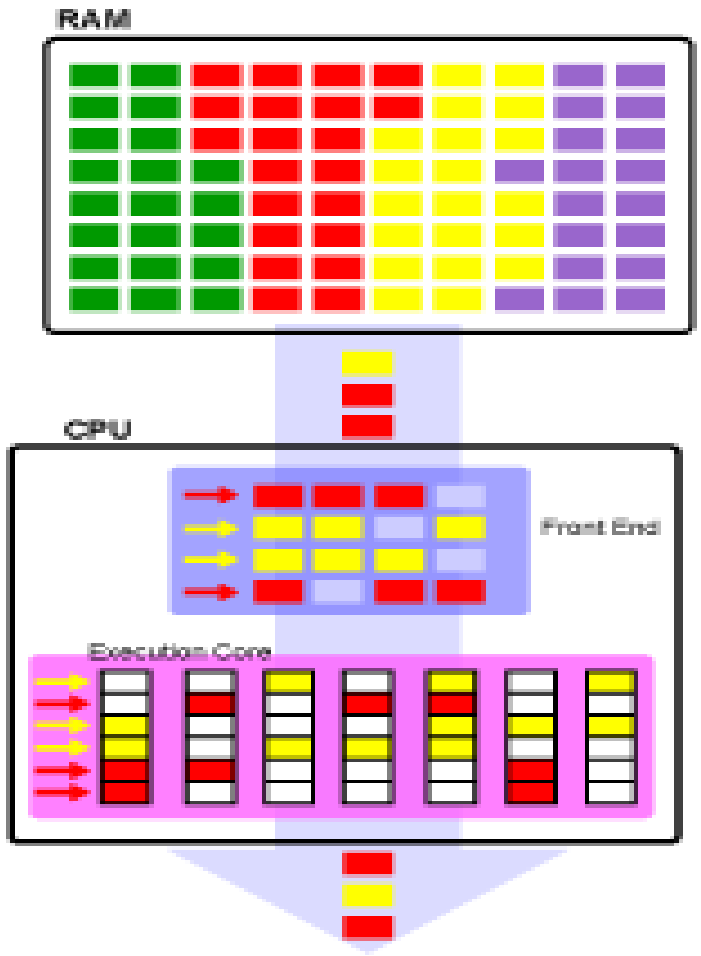
Commercial Symmetric Multiprocessors (SMPs) : IBM R690

- Limitation is mainly caused by using a centralized shared memory and a bus or cross bar interconnect which are both difficult to scale once built.

Superthreading with Multi threaded processor

Time-slice multi threading

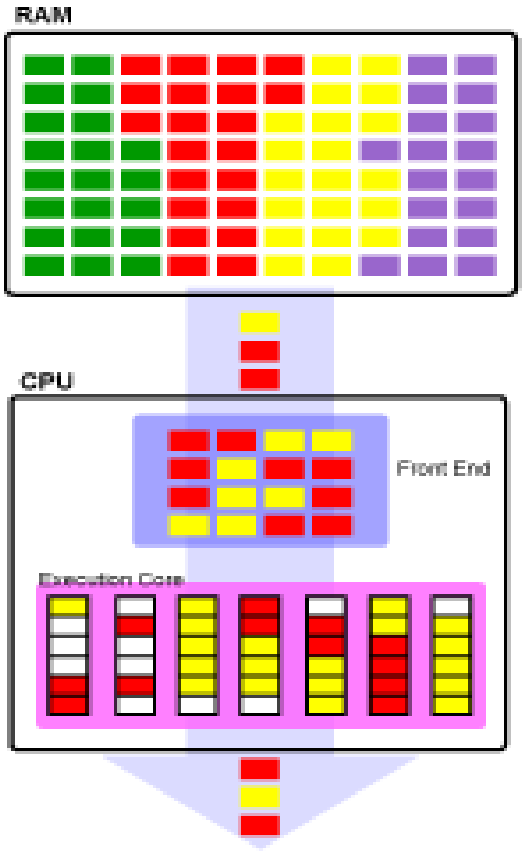
- Processor that use this technique is called multi threaded processor.
- Capable of executing more than one thread at a time.
- Front end (instruction queue) issue four instruction simultaneously
- All four instruction comes from single thread only.
- Deals with memory access latency but don't deals with poor instruction level parallelism



Reference : [6], [29], [31]

Introduction Hyperthreading

- Takes superthreading to next lev
- Hyper threading is simple super out restriction that all instruction from same thread only
- Its strength is it allows the sched maximum flexibility to fill executi
- Make efficient use of available e resources by keeping the execut busier

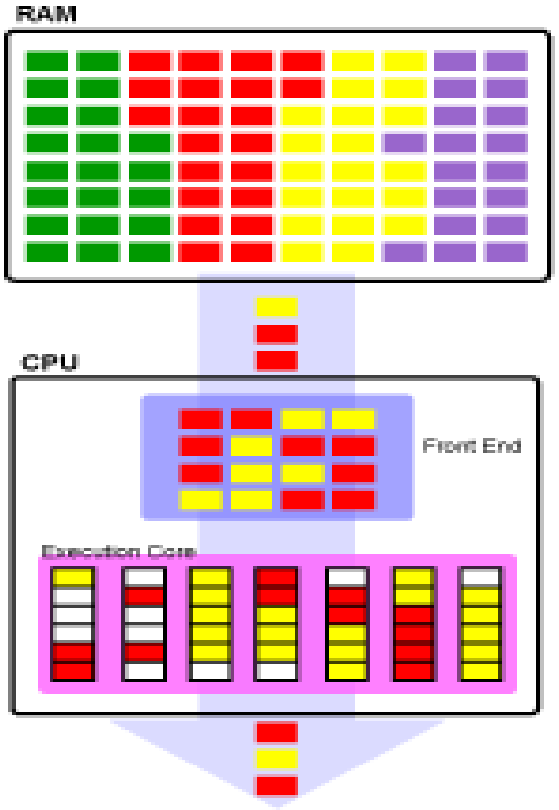


- The implementation of HT technology was the first SMT solution for general processors from Intel.

Source : http://www.intel.com ; Reference : [6], 29], [31]

Hyper threading :The next step introduction

- Hyper threading is simple super threading with out restriction that all instruction should come from same thread only
- Its strength is it allows the scheduling logic maximum flexibility to fill execution slots.
- Make efficient use of available execution resources by keeping the execution core busier



- Inside a processor with HT technology, two threads share resources from a single core - Threads are referred to as logical processors.

Source : http://www.intel.com ; Reference : [6], 29], [31]

Implementing Hyperthreading

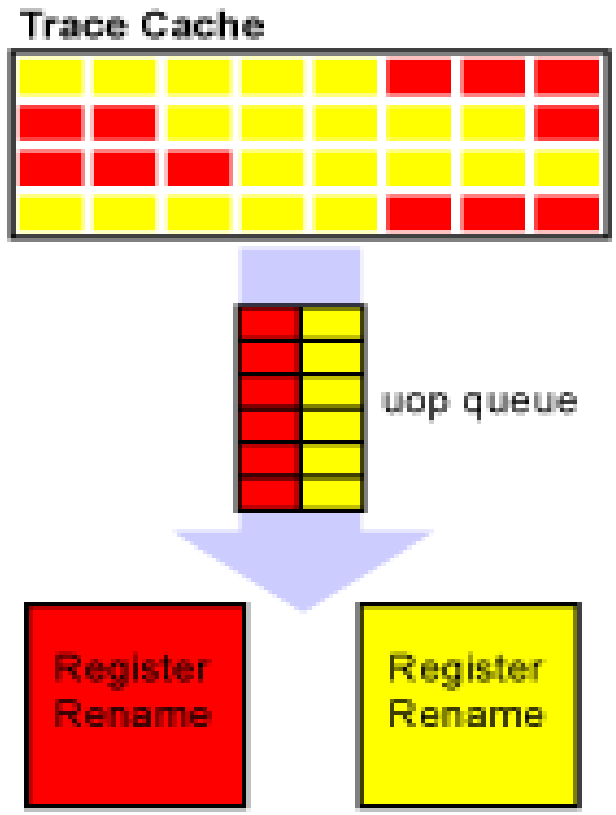
Hyper threading is implemented by dividing processor micro architectural resources into three types.

- Replicated :
  - Instruction Pointer
  - various other architectural register
- partitioned :
  - Re-order buffers
  - Load/Store buffers
- shared :
  - Caches : trace cache, L1, L2, L3
  - Micro Architectural Registers
  - Execution units

Source : http://www.intel.com
Reference : [6], 29], [31]

Hyper-threading : Partitioned Resources

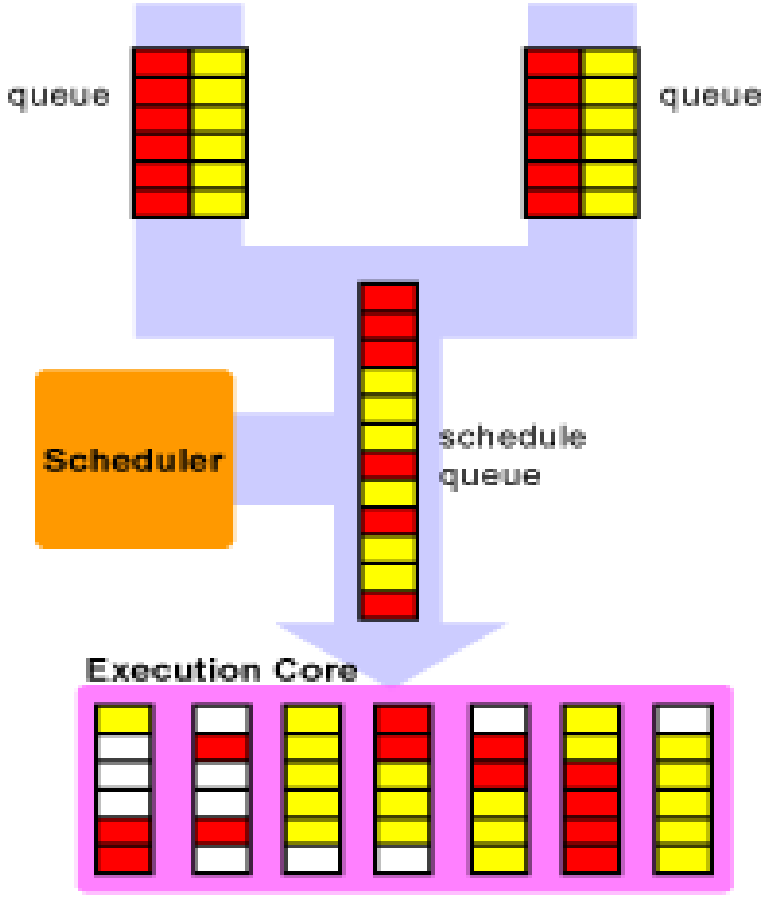
- The most partitioned resources are mostly to be found in the form of queues that decouple the major stages of the pipeline from one another.
- There are two kind of partitioned resources
  - 1) Statically partitioned
  - 2) Dynamically partitioned



Source : http://www.intel.com ; Reference : [6], 29], [31]

Hyper-threading : Partitioned Resources

- Xeon's scheduling queues are dynamically partitioned in order to keep one logical processor from monopolizing them
- There is limitation on number of entries that each logical processor can use



Source : http://www.intel.com ; Reference : [6], 29], [31]

Hyper-threading : Partitioned Resources

- Shared resources are at the heart of Hyper-threading
- More resources that can be shared between logical processors ,the more efficient hyper-threading can be squeezing the maximum amount of computing power achieved
- Some shared resources are integer unit,floating point unit,register file etc.

Source : http://www.intel.com ; Reference : [6], 29], [31]

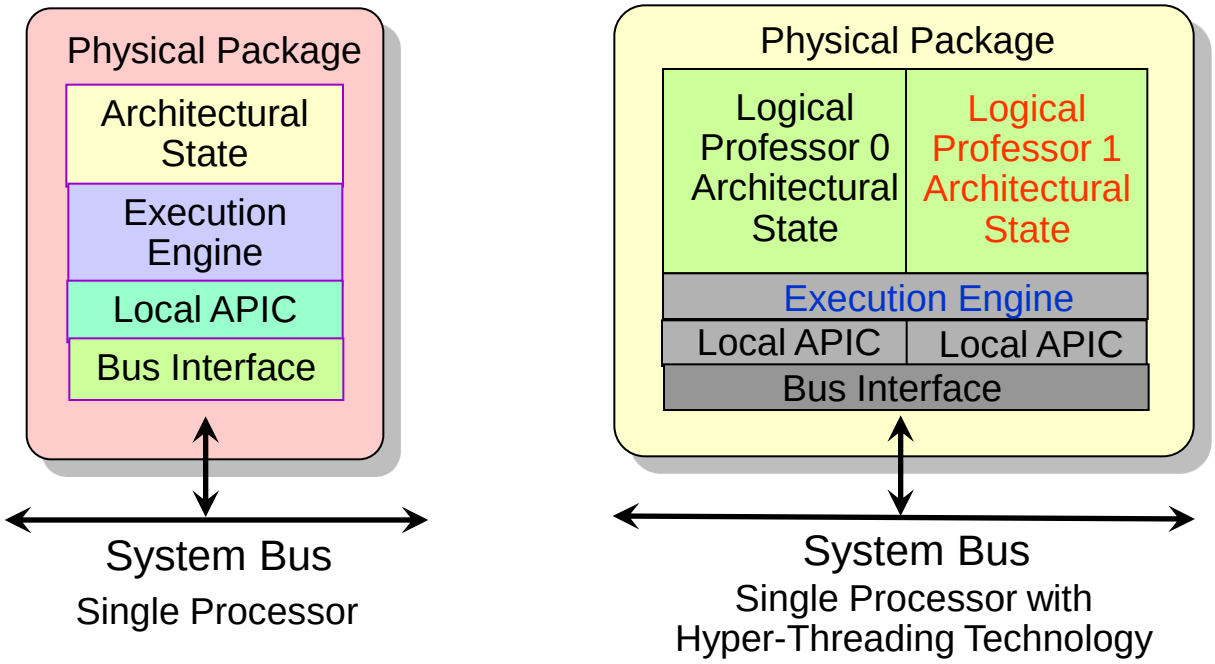
Hyper-threading : Partitioned Resources

- Cache coherency problems associated with SMP all but disappear.
- Both logical processor on an SMT system share same resources.
- However, since both logical processors share the same cache the prospect of cache conflicts increase.

Source : http://www.intel.com ; Reference : [6], 29], [31]

Hyper-threading : Partitioned Resources

- Hyper-threading (HT) technology is a hardware mechanism where multiple independent hardware threads get to execute in a single cycle on a single super-scalar processor core.

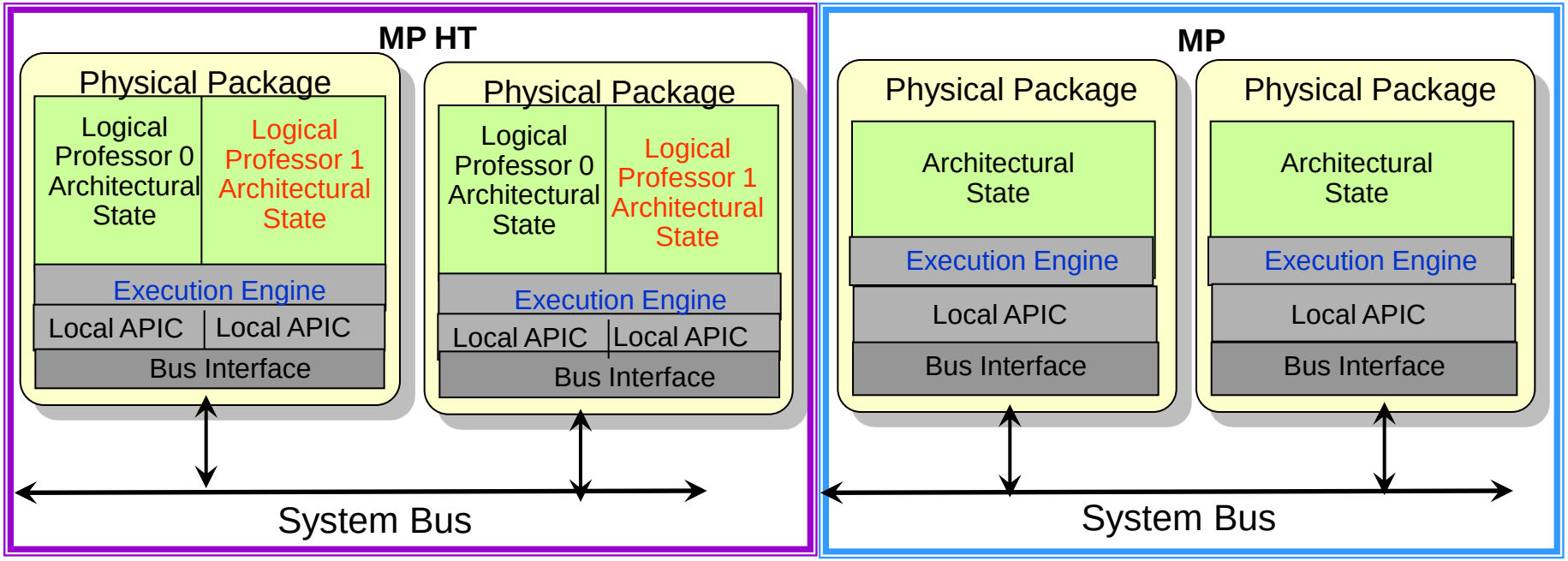


Single Processor System without Hyper-Threading Technology and Single Processor System with Hyper-Threading Technology

Source : http://www.intel.com ; Reference : [6], [10], [19], [23], [24],[29], [31]

Hyper-threading Technology

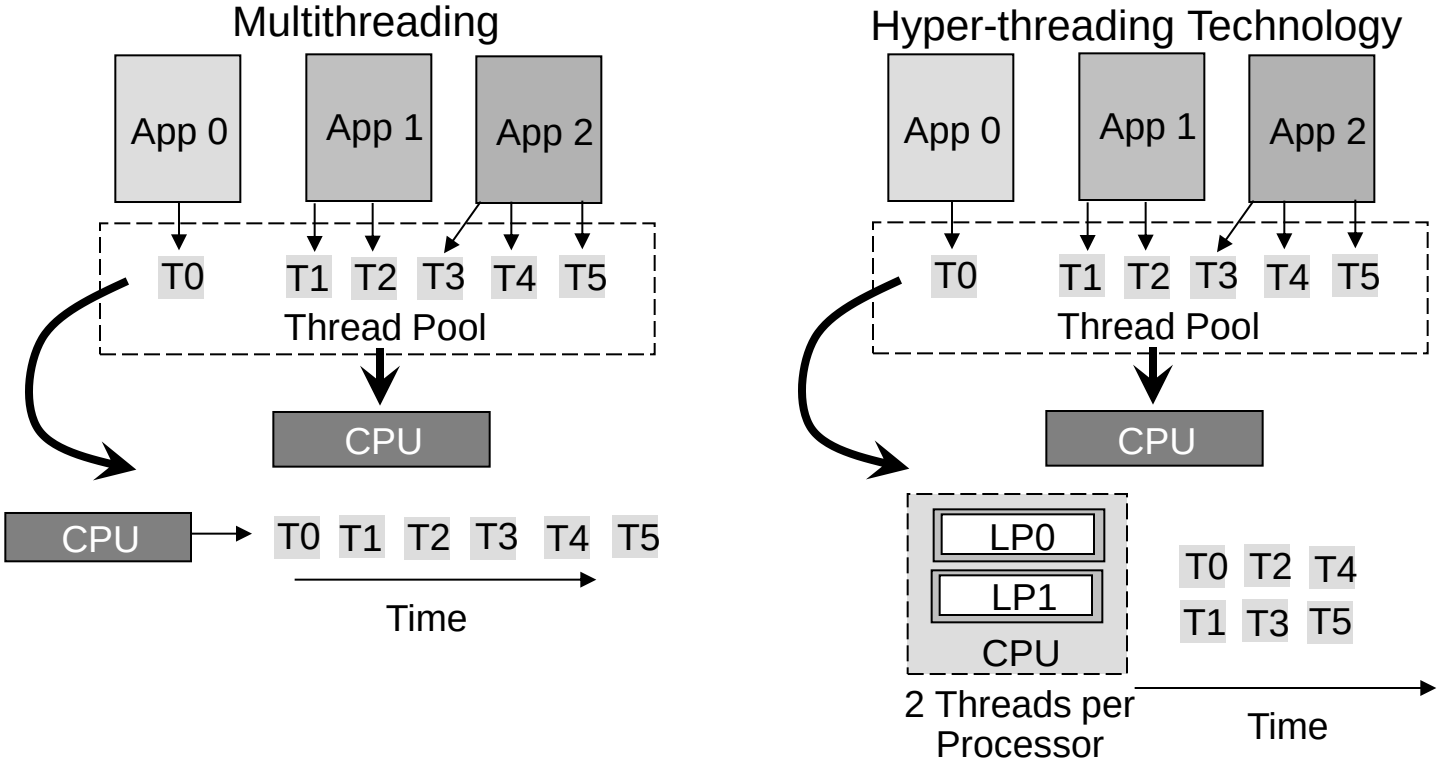
- Multi-processor with and without Hyper-threading (HT) technology



Source : http://www.intel.com ; Reference : [6], [29]. [31]

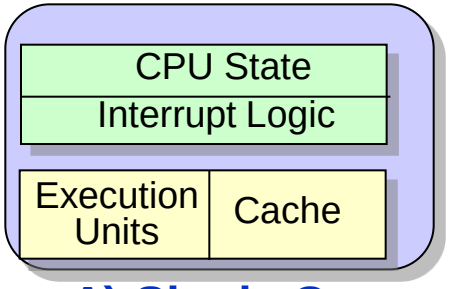
Multi-threaded Processing using Hyper-Threading Technology

- Time taken to process *n* threads on a single processor is significantly more than a single processor system with HT technology enabled.

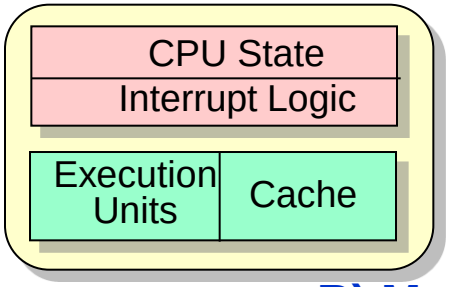


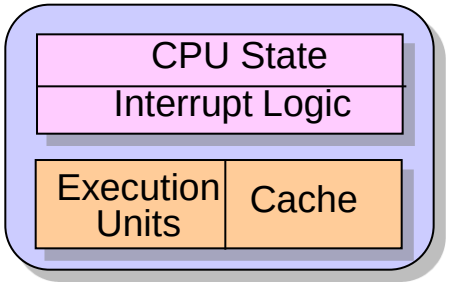
Source : http://www.intel.com ; Reference : [6], [29], [31]

Simple Comparison of Single-core, Multi-processor, and multi-Core Architectures

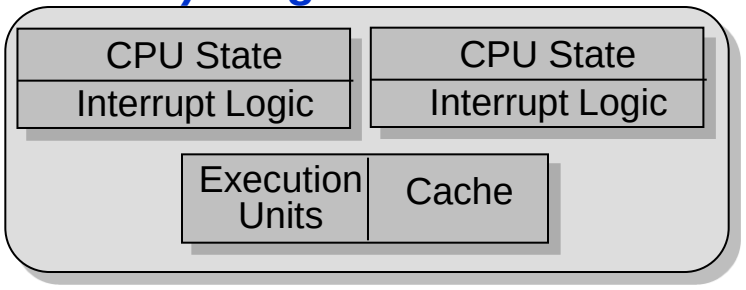


A) Single Core

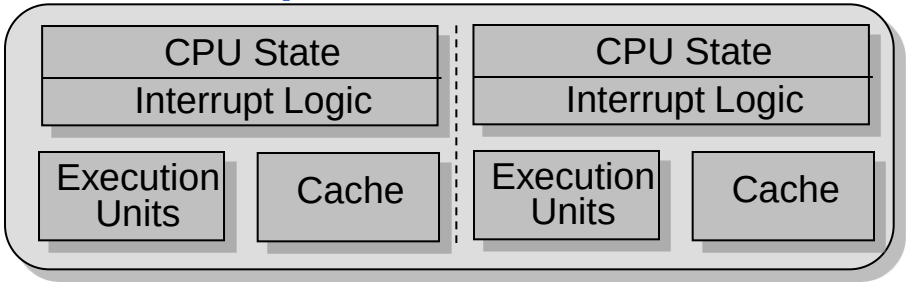




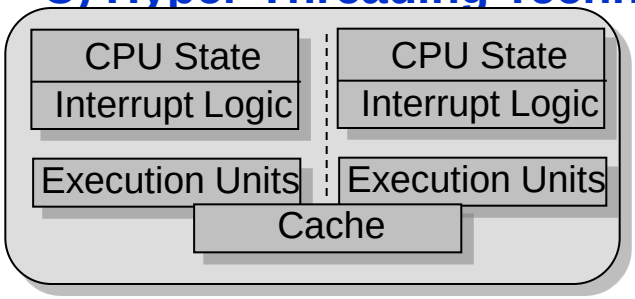
B) Multi Processor



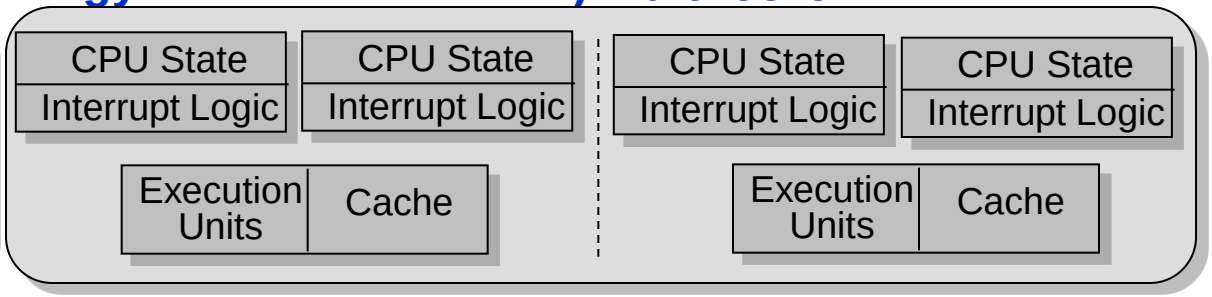
C) Hyper-Threading Technology



D) Multi-core



E) Multi-core with Shared Cache

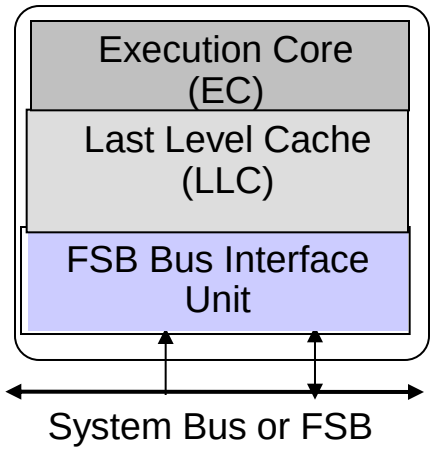


F) Multi-core with Hyper-threading Technology

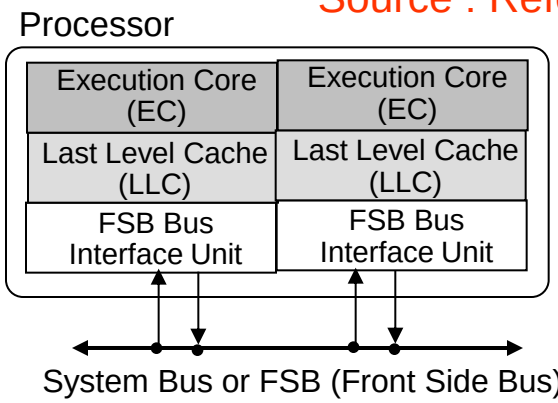
Source : http://www.intel.com ; Reference : [4],[6], [29], [31]

Multi-Core Processor Configurations

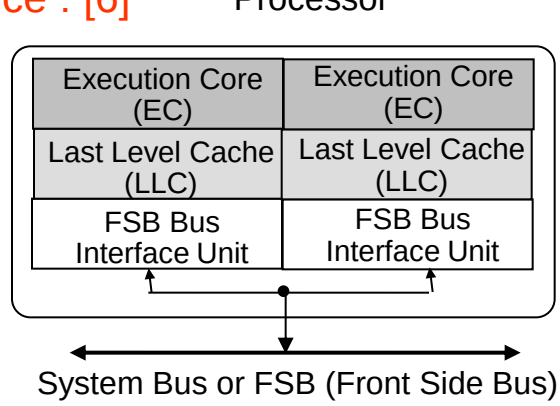
Source : Reference : [6]



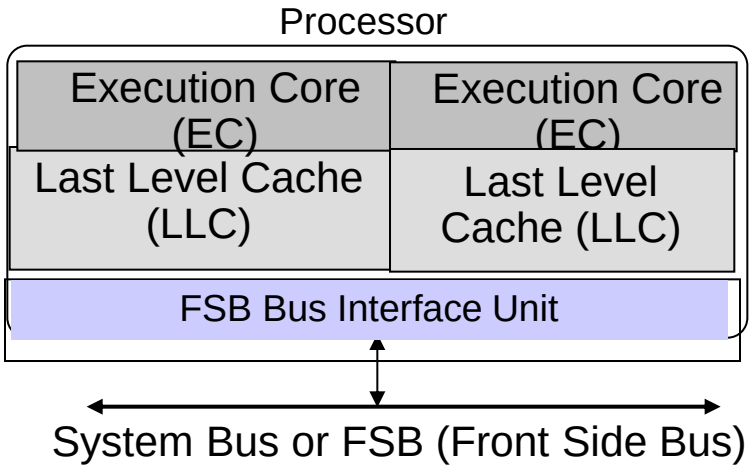
(a) Single Core Processor



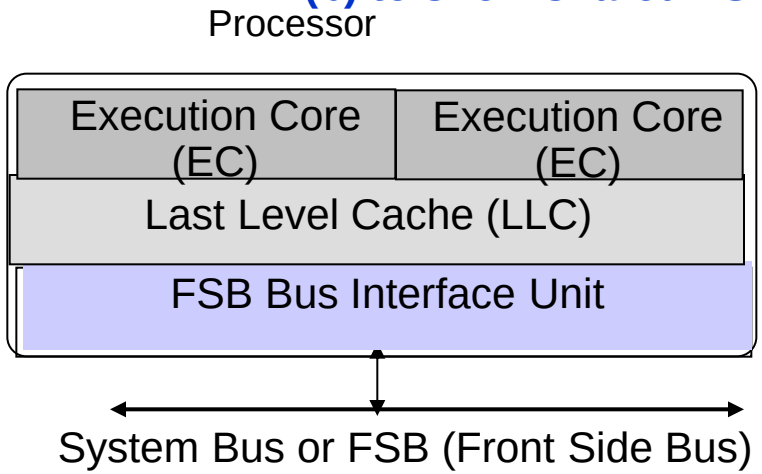
(b) Multi-Core Processor with Two Cores and Individual FSB



(c) Actual representation of (b) to show shared FSB



(d) Multi-Core Processor with Two Cores and Shared FSB



(e) Multi-Core Processor with Two Cores and Shared LLC and FSB

Multi Core : Programming Issues

- Out of Order Execution
- Preemptive and Co-operative Multitasking
- SMP to the rescue
- Super threading with Multi threaded Processor
- Hyper threading the next step (Implementation)
- Multitasking
- Caching and SMT

Conclusions

- An overview of Threading
- An overview of Superthreading & HyperThreading
- Multi Core Processor Configurations of various Systems
- An Overview of Multi-Core Architecture : Part I

References

1. Andrews, Grogory R. **(2000),** Foundations of Multithreaded, Parallel, and Distributed Programming, Boston, MA : Addison-Wesley

2. Butenhof, David R **(1997),** Programming with POSIX Threads , Boston, MA : Addison Wesley Professional

3. Culler, David E., Jaswinder Pal Singh **(1999),** Parallel Computer Architecture - A Hardware/Software Approach , San Francsico, CA : Morgan Kaufmann

4. Grama Ananth, Anshul Gupts, George Karypis and Vipin Kumar **(2003),** Introduction to Parallel computing, Boston, MA : Addison-Wesley

5. Intel Corporation, **(2003),** Intel Hyper-Threading Technology, Technical User's Guide, Santa Clara CA : Intel Corporation Available at : http://www.intel.com

6. Shameem Akhter, Jason Roberts **(April 2006),** Multi-Core Programming - Increasing Performance through Software Multi-threading , Intel PRESS, Intel Corporation,

7. Bradford Nichols, Dick Buttlar and Jacqueline Proulx Farrell **(1996),** Pthread Programming O'Reilly and Associates, Newton, MA 02164,

8. James Reinders, Intel Threading Building Blocks – (**2007**) , O'REILLY series

9. Laurence T Yang & Minyi Guo (Editors), (**2006**) *High Performance Computing - Paradigm and Infrastructure* Wiley Series on Parallel and Distributed computing, Albert Y. Zomaya, Series Editor

10. Intel Threading Methodology ; Principles and Practices Version 2.0 copy right **(March 2003),** Intel Corporation

References

11. William Gropp, Ewing Lusk, Rajeev Thakur **(1999),** Using MPI-2, Advanced Features of the Message-Passing Interface, The MIT Press..

12. Pacheco S. Peter, **(1992),** Parallel Programming with MPI, , University of Sanfrancisco, Morgan Kaufman Publishers, Inc., Sanfrancisco, California

13. Kai Hwang, Zhiwei Xu, (**1998**), Scalable Parallel Computing (Technology Architecture Programming), McGraw Hill New York.

14. Michael J. Quinn (**2004**), Parallel Programming in C with MPI and OpenMP McGraw-Hill International Editions, Computer Science Series, McGraw-Hill, Inc. Newyork

15. Andrews, Grogory R. **(2000),** Foundations of Multithreaded, Parallel, and Distributed Progrmaming, Boston, MA : Addison-Wesley

16. SunSoft. Solaris multithreaded programming guide. SunSoft Press, Mountainview, CA, **(1996),** Zomaya, editor. Parallel and Distributed Computing Handbook. McGraw-Hill,

17. Chandra, Rohit, Leonardo Dagum, Dave Kohr, Dror Maydan, Jeff McDonald, and Ramesh Menon, **(2001)**,Parallel Programming in OpenMP San Fracncisco Moraan Kaufmann

18. S.Kieriman, D.Shah, and B.Smaalders **(1995),** Programming with Threads, SunSoft Press, Mountainview, CA. 1995

19. Mattson Tim, **(2002),** Nuts and Bolts of multi-threaded Programming Santa Clara, CA : Intel Corporation, Available at : http://www.intel.com

20. I. Foster **(1995,** Designing and Building Parallel Programs ; Concepts and tools for Parallel Software Engineering, Addison-Wesley (1995)

21. J.Dongarra, I.S. Duff, D. Sorensen, and H.V.Vorst **(1999),** Numerical Linear Algebra for High Performance Computers (Software, Environments, Tools) SIAM, 1999

References

22. OpenMP C and C++ Application Program Interface, Version 1.0". **(1998),** OpenMP Architecture Review Board. October 1998

23. D. A. Lewine. *Posix Programmer's Guide:* **(1991),** Writing Portable Unix Programs with the Posix. 1 Standard. O'Reilly & Associates, 1991

24. Emery D. Berger, Kathryn S McKinley, Robert D Blumofe, Paul R.Wilson, *Hoard : A Scalable Memory Allocator for Multi-threaded Applications* ; The Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX). Cambridge, MA, November (**2000)**. Web site URL : http://www.hoard.org/

25. Marc Snir, Steve Otto, Steyen Huss-Lederman, David Walker and Jack Dongarra, (**1998**) *MPI-The Complete Reference: Volume 1, The MPI Core, second edition* [MCMPI-07].

26. William Gropp, Steven Huss-Lederman, Andrew Lumsdaine, Ewing Lusk, Bill Nitzberg, William Saphir, and Marc Snir (**1998**) *MPI-The Complete Reference: Volume 2, The MPI-2 Extensions*

27. A. Zomaya, editor. Parallel and Distributed Computing Handbook. McGraw-Hill, **(1996)**

28. OpenMP C and C++ Application Program Interface, Version 2.5 (**May 2005**)", From the OpenMP web site, URL **: http://www.openmp.org/**

29. Stokes, Jon 2002 Introduction to Multithreading, Super-threading and Hyper threading *Ars Technica*, October **(2002)**

30. Andrews Gregory R. 2000, Foundations of Multi-threaded, Parallel and Distributed Programming, Boston MA : Addison – Wesley (**2000)**

31. Deborah T. Marr , Frank Binns, David L. Hill, Glenn Hinton, David A Koufaty, J . Alan Miller, Michael Upton, "Hyperthreading, Technology Architecture and Microarchitecture", Intel (**2000-01)**

Thank You

- *Any questions ?*