Micro-Architectures Evolution in Generations of Intel's Processors

Nitin Mathur
Assistant Professor
Department of Computer Science and Engineering
Jodhpur Institute of Engineering & Technology
Jodhpur, India
nitin.mathur@jietjodhpur.com

Dr. Rajesh Purohit
Associate Professor and Head
Department of Computer Science and Engineering
M.B.M. Engineering College, J. N. V. University
Jodhpur, India
rajeshpurohitmbm@yahoo.com

***Abstract*—** Parallel computing has made great pace in recent years with no doubt. It has advanced in the last twenty years due to the emergence and migration of new workloads and usage models of mainstream computing. Intel employs parallel computing by implement instruction-level parallelism with advancement in generations of superscalar processors' micro-architecture in last two decades. Intel put their great effort in developing processor microarchitecture in superscalar processors.

Intel's vision for the evolution of architectural innovation and core competencies enabling that evolution is to achieve maximum parallelism, provide performance at its highest level. This paper will focus on comparative learn of advances in processor microarchitecture of Intel to implement instruction-level parallelism.

Index Terms— Parallelism, Superscalar, Processor micro-architecture, Intel.

I. INTRODUCTION

Parallel processing has emerged as a key enabling technology that is driven by concurrent events in modern computers. Parallel processing requires concurrent execution of many events in the computer. These concurrent events are attainable in a computer system at various processing levels. Parallelism can be applied at various levels of processing such as job, module and instruction.

Instruction-level parallelism (ILP) realized by processor architecture that speed ups execution by causing individual machine operations to execute in parallel. It is necessary to take decisions about executions of multiple operations handled by processor hardware. According to decision taking power by hardware, instruction-level parallelism architectures can be called as *Superscalar*. Superscalar architectures use special hardware to analyze the instruction stream at execution time and to determine which operations in the instruction stream are independent of all preceding operations and have their source operands available. These operations can then be issued and executed concurrently.

This paper's focus on evolution of processors of Intel after 80486 and from Intel Pentium to Intel Sandy Bridge superscalar micro-architecture. The central point of this paper is development of micro-architecture of superscalar design in different generations of Intel processors that implement instruction-level parallelism. In a textbook on computer architecture by Blaauw and Brooks [1977], authors defined the distinction between architecture and micro-architecture.

Architecture defines the functional behavior of the processor. It specifies an instruction set that characterizes the functional behavior of an instruction set processor. All software must be mapped to or encoded in this instruction set in order to execute by the processor. Every program is complied into a sequence of instructions in this instruction set. Examples of well-known architectures are IBM 360, PowerPC and Intel IA32.

An implementation is a specific design of an architecture referred to as *microarchitecture*. Architecture can have many implementations in the lifetime of that ISA. All microarchitecture of architecture can execute any program encoded in that ISA. Examples of some well-known microarchitecture are IBM 360/91, PowerPC 604 and Intel P6. Attributed associated with a micro-architecture include pipeline design, cache memories and branch predictors. Microarchitecture features are generally implemented in hardware and hidden from software.

The Pentium processor was Intel's first superscalar micro-architecture design following the popular i486 CPU family in 1993. The design started in early 1989 with the primary goal of maximizing performance while preserving

software compatibility within the practical constraint of available technology. Intel's P6 micro-architecture was designed to outperform all other x86 CPUs by a significant margin in 1995. Although it shares some design techniques with competitors such as AMD's K5, NexGen's Nx586, and Cyrix's M1, the new Intel chip has several important advantages over these competitors. In 2000, Intel launched Netburst microarchitecture of Intel's new flagship Pentium 4 processor that is basis of a new family of processors from Intel stating with the Pentium 4.

It implemented significantly higher clocks rates, internet audio and streaming video, image processing, speech recognition, 3D applications and games, multi-media and multitasking environment. It includes streaming SIMD instructions that improve performance for multi-media, content creation, scientific and engineering applications. In 2000, Intel unwrapped a new and the first IA-64 Merced microarchitecture with association of HP, showing how IA-64's EPIC design results in hardware that is both simpler and more powerful than traditional RISC or CISC processors. Gone are the complex instruction reorder buffers and register alias tables found in modern superscalar processors.

In their place are more registers, more function units, and more branch predictors. This design was big step forward for Intel in the workstation and server markets. In 2003, Intel turned desktop to mobile with Pentium M. The Intel Pentium M processor is a key component of Intel Centrino Mobile technology platform. It is Intel's first micro-architecture designed specifically for mobility. It provides outstanding mobile performance and its dynamic power management enables energy saving for longer battery life.

Intel first introduced Intel Core microarchitecture in 2006 with our 65nm silicon process technology. The first generation of this multi-core optimized microarchitecture extended the energy-efficient philosophy first delivered in the mobile microarchitecture of the Intel Pentium M processor and enhanced it with many new, leading-edge microarchitecture innovations for industry-leading performance, greater energy efficiency and more responsive multitasking.

Intel Core microarchitecture innovations include: • Intel Wide Dynamic Execution, • Intel Intelligent Power Capability, • Intel Advanced Smart Cache, • Intel Smart Memory Access, • Intel Advanced Digital Media Boost. Processors based on Intel Core microarchitecture have delivered record-setting performance on leading industry benchmarks for desktop, mobile and mainstream server platforms.

In 2008, a new microarchitecture codenamed Nehalem launched to rewriting the book on processor energy efficiency, performance and scalability. This next generation Intel microarchitecture Nehalem is a dynamically scalable and design-scalable microarchitecture. At runtime, it dynamically manages cores, threads, cache, interfaces and power to deliver outstanding energy efficiency and performance on demand. At design time, it scales, enabling Intel to easily provide versions that are optimized for server, desktop and notebook market. Intel delivered version differing in the number of cores, caches, interconnect capability and memory controller capability.

Intel's processor clock has tocked, delivering next generation architecture for PCs and servers in 2010. The new CPU is an evolutionary improvement over its predecessor, Nehalem, tweaking the branch predictor, register renaming, and instruction decoding. The big changes in Sandy Bridge target multimedia applications such as 3D graphics, image processing, and video processing. The chip is Intel's first to integrate the graphics processing unit (GPU) on the processor itself. This integration not only eliminates an external chip, but it improves graphics performance by more closely coupling the GPU and the CPU. Sandy Bridge introduced the Advanced Vector Extensions (AVX), which double peak floating-point throughput. AVX is accelerating many 3D-graphics and imaging applications. The new processor also adds hard-wired video encoding. Sandy Bridge was first appearing in desktop and notebook processors that was announced in early 2011 and branded as "2nd generation Intel Core" processors.

Instruction-level Parallelism is discussed in next section. Superscalar ILP architectures are described in section 3. Innovation learn of micro-architectures is explained in section 4 and last end with conclusion appears in section 5.

II. INSTRUCTION-LEVEL PARALLELISM (ILP)

Instruction Level Parallelism is the lowest level of parallelism. At instruction or statement level, a typical grain contains less than twenty instructions, called "*fine grain*". Depending on individual programs, fine-grain parallelism at this level range from two to thousand. The advantage of fine-grain computation lays in the excess of parallelism.. ILP can be defined by various ways. Some are as follows.

(a) *Instruction-level parallelism* defined as amount of parallelism (measured by the number of instructions) that can be achieved by issue and execute multiple instructions concurrently.

(b) *Instruction-level parallelism* may be defined as the capability to exploring a sequential instruction stream, identify independent instructions, issue multiple instructions per cycle and send to several execution units in parallel to fully utilizing the available resource [1], [11], [12], [19], [20].

III. Superscalar Ilp Architecture

The outcome of instruction-level parallel execution is that multiple operations are simultaneously in execution. It is necessary to take decision about when and whether operation should be executed. Superscalar ILP architectures schedule the instructions at run-time or execution time and executed the multiple instructions at multiple execution units simultaneously. Superscalar processors are based on sequential architecture. Superscalar machines incorporate multiple functional units to achieve greater concurrent processing of multiple instructions and higher execution throughput.

A superscalar processor makes a great effort to issue an instruction every cycle so as to execute many instructions in parallel, even though program is sequentially handed by the hardware. With every instruction that a superscalar processor issues, it must check the instruction's operands interfere with the operands of any other instruction in flight. Once an instruction is independent of all other ones in flight, the hardware must be also decide exactly when and on which available functional unit to execute the instruction. Figure 1 shows the superscalar execution. Superscalar processor rely on hardware for the scheduling the instructions which is called *Dynamic instruction scheduling*. Figure 1 shows structure of superscalar processor of degree 3 with multiple execution units. In order to fully utilize a superscalar processor of degree m must issues m instructions per cycle to execute in parallel at all times. If ILP of m is not available, stalls and dead time will result where instructions are waited for results of previous instruction [1], [10], [11], [12], [19], [20].

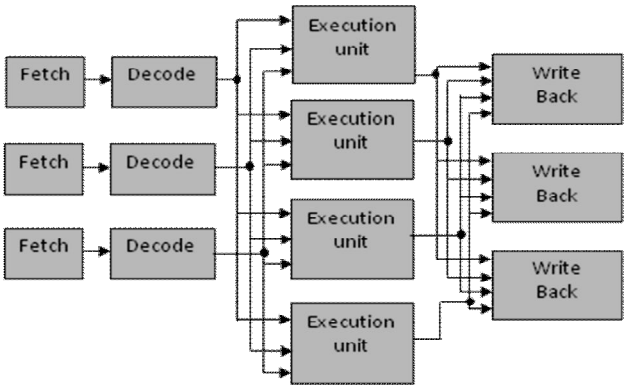

Figure 1: Superscalar Execution

IV. Evolution Revise

Throughout history, new and improved technologies have transformed the human experience. In the 20th century, the pace of change sped up radically as we entered the computing age. For nearly 40 years Intel innovations have continuously created new possibilities in the lives of people around the world.

In 1965, Intel co-founder Gordon Moore predicted that the number of transistors on a chip would double about every two years. Since then, Moore's Law has fueled a technology revolution as Intel has exponentially increased the number of transistors integrated into it processors for greater performance and energy efficiency. Figure 2 shows the evolution of Intel's micro-architecture from Pentium to Sandy Bridge.

Intel started implementation of instruction-level parallelism in its first superscalar design with Pentium in 1993. The most important enhancements over the 486 are the separate instruction and data caches, the dual integer pipelines (the U-pipeline and the V-pipeline, as Intel calls them), branch prediction using the branch target buffer (BTB), the pipelined floating-point unit, and the 64-bit external data bus. Even-parity checking is implemented for the data bus and the internal RAM arrays (caches and TLBs) [2], [3], [6], [8], [9].

Pentium was the first high-performance microprocessor to include a system management mode like those found on power-miserly processors for notebooks and other battery-based applications; Intel was holding to its promise to include SMM on all new CPUs. The integer data path is in the middle, while the floating-point data path is on the side opposite the data cache. In contrast to other superscalar designs, Pentium's integer data path is actually bigger than its FP data path. This is an indication of the extra logic associated with complex instruction support.

Intel came with P6 micro-architecture in 1996 which deep pipeline eliminates the cache-access bottlenecks that restrict its competitors to clock speeds of about 100 MHz. In addition, the Intel design uses a closely coupled secondary cache to speed memory accesses, a critical issue for high-frequency CPUs. Intel will combine the P6 CPU and a cache chip into a single PGA package, reducing the time needed for data to move from the cache to the processor.

Like some of its competitors, the P6 translates x86 instructions into simple, fixed-length instructions that Intel calls micro-operations or uops (pronounced "youops"). These uops are then executed in a decoupled superscalar core

capable of register renaming and out-of-order execution. Intel has given the name "dynamic execution" to this particular combination of features, which is neither new nor unique, but highly effective in increasing x86 performance. The P6 also implements a new system bus with increased bandwidth compared to the Pentium bus. The new bus is capable of supporting up to four P6 processors with no glue logic, reducing the cost of developing and building multiprocessor systems. This feature set makes the new processor particularly attractive for servers; it will also be used in high-end desktop PCs and, eventually, in mainstream PC products.

The P6 team threw out most of the design techniques used by the 486 and Pentium and started from a blank piece of paper to build a high-performance x86-compatible processor. The result is a microarchitecture that is quite radical compared with Intel's previous x86 designs, but one that draws from the same bag of tricks as competitors' x86 chips. To this mix, the P6 adds high-performance cache and bus designs that allow even large programs to make good use of the superscalar CPU core [6], [8], [9], [15].

In 2000, Pentium launched NetBurst microarchitecture with new features. The design goals of Intel NetBurst micro-architecture are: (a) to execute both the legacy IA-32 code and applications based on single-instruction, multiple-data (SIMD) technology at high processing rates; (b) to operate at high clock rates, and to scale to higher performance and clock rates in the future. To accomplish these design goals, the Intel NetBurst micro-architecture has many advanced features and improvements over the P6 micro-architecture.

To enable high performance and highly scalable clock rates, the major design considerations of the Intel NetBurst micro-architecture are as follows:

• It uses a deeply pipelined design to enable high clock rates with different parts of the chip running at different clock rates, some faster and some slower than the nominally-quoted clock frequency of the processor. The Intel NetBurst micro-architecture allows the Pentium 4 processor to achieve significantly higher clock rates as compared with the Pentium III processor. These clock rates will achieve well above 1 GHz.

• Its pipeline provides high performance by optimizing for the common case of frequently executed instructions. This means that the most frequently-executed instructions in common circumstances (such as a cache hit) are decoded efficiently and executed with short latencies, such that frequently encountered code sequences are processed with high throughput.

• It employs many techniques to hide stall penalties. Among these are parallel execution, buffering, and speculation. Furthermore, the Intel NetBurst micro-architecture executes instructions dynamically and out-or-order, so the time it takes to execute each individual instruction is not always deterministic. Performance of a particular code sequence may vary depending on the state the machine was in when that code sequence was entered [4], [6], [7], [8], [9].

In same year 2000, Intel designers described Merced microarchitecture as a six-wide machine, fetching and executing two bundles, or six instructions, per cycle at its peak rate. The processor uses a 10-stage pipeline to achieve high clock speeds, although designers declined to specify the target clock speed. IA-64 features such as predication, speculation, and register rotation are implemented with simple hardware structures. Dynamic structures, such as a decoupled fetch unit, non-blocking caches, and register score-boarding, avoid pipeline stalls due to level-one (L1) cache misses.

The tighter coupling between the compiler and the processor improves hardware efficiency compared with traditional RISC or x86 designs. The Merced microarchitecture processor employs a tighter coupling between hardware and software. In this design style the hardware- software interface lets the software exploit all available compilation time information and efficiently deliver this information to the hardware. It addresses several fundamental performance bottlenecks in modern computers, such as memory latency, memory address disambiguation, and control flow dependencies.

Merced constructs provide powerful architectural semantics and enable the software to make global optimizations across a large scheduling scope, thereby exposing available instruction-level parallelism (ILP) to the hardware. The hardware takes advantage of this enhanced ILP, providing abundant execution resources. Additionally, it focuses on dynamic runtime optimizations to enable the compiled code schedule to flow through at high throughput. This strategy increases the synergy between hardware and software, and leads to higher overall performance [5], [6], [8], [9].

In 2003, Intel came with Intel Pentium M architecture that is a key component of Intel Centrino. Mobile technology platform. It is Intel's first microprocessor designed specifically for mobility. It provides outstanding mobile performance and its dynamic power management enables energy saving for longer battery life. Designing a mobile processor calls for different power/performance tradeoffs than designing a traditional high performance processor. The Intel

Pentium M processor's architectures focused to achieve best performance at given power and thermal constraints.

Design tradeoffs for the mobile market are rather complicated and involve several challenges:

i. Optimizing the design for maximum performance and extended battery life. The challenge lies in how to balance between these conflicting goals.
ii. Trading performance for power. Performance features whether increasing instruction-level parallelism (ILP) or speeding up frequency. Usually increase power consumption. Power-saving features usually decrease performance. The challenge lies in figuring out how much power one can afford to lose in order to implement a performance feature.

During the definition of the Intel Pentium M processor architectures tended to use the stricter criterion in each case:

i. Reducing number of instructions per task.
ii. Reducing number of micro-ops per instruction.
iii. Reducing number of transistor switches per micro-op.
iv. Reducing the amount of energy per transistor switch.

The following statements describe several of the Intel Pentium M processor power-aware features. These features cover all the above-mentioned strategies:

i. Reducing the number of instructions per task: advanced branch prediction.
ii. Reducing the number of micro-ops per instruction: micro-ops fusion and dedicated stack engine.
iii. Reducing the number of transistor switches per micro-op: the Intel Pentium M processor bus and various lower-level optimizations.
iv. Reducing the amount of energy per transistor switch: Intel Speed Step technology [6], [8], [9], [21]

Intel commenced with Intel Core microarchitecture in 2006. The Intel® Core™ microarchitecture is a new foundation for Intel® architecture–based desktop, mobile, and mainstream server multi-core processors. This state-of-the-art multi-core– optimized and power-efficient microarchitecture is designed to deliver increased *performance* and *performance per watt*—thus increasing overall energy efficiency.

For Core microarchitecture, Intel's design goals focused on areas such as connectivity, manageability, security, and reliability, as well as compute capability. One means of significantly increasing compute capability is with Intel multi-core processors delivering greater levels of performance and performance-per-watt capabilities. The move to multi-core processing has also opened the door to many other microarchitectural innovations that will further improve performance. Intel Core microarchitecture is focused on enhancing existing and emerging application and usage models across each platform segment, including desktop, server, and mobile.

Intel has long been the leader in driving down power consumption in laptops. The mobile microarchitecture found in the Intel Pentium M processor and Intel Centrino mobile technology has consistently delivered an industry-leading combination of laptop performance, performance per watt, and battery life. Intel NetBurst microarchitecture has also delivered innovations enabling great performance in the desktop and server segments.

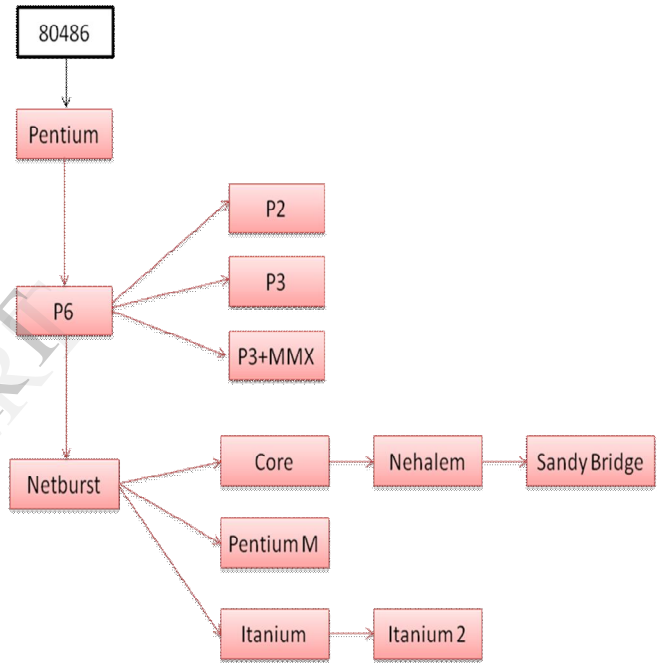

Figure 2: Evolution of Intel Micro-architecture

The Key points of Intel Core microarchitecture innovations are as follows:

• Intel Wide Dynamic Execution

Now, with the Intel Core microarchitecture, Intel significantly enhances this capability with Intel Wide Dynamic Execution. It enables delivery of more instructions per clock cycle to improve execution time and energy efficiency. Every execution core is wider, allowing each core to fetch, dispatch, execute, and return up to four full instructions simultaneously. One feature for reducing execution time is macrofusion. Macrofusion enables common instruction pairs (such as a compare followed by a conditional

jump) to be combined into a single internal instruction (micro-op) during decoding. Two program instructions can then be executed as one micro-op, reducing the overall amount of work the processor has to do. This increases the overall number of instructions that can be run within any given period of time or reduces the amount of ime to run a set number of instructions. By doing more in less time, macrofusion improves overall performance and energy efficiency.

• Intel Intelligent Power Capability

Intel Intelligent Power Capability is a set of capabilities designed to reduce power consumption and device design requirements. This feature manages the runtime power consumption of all the processor's execution cores. It includes an advanced power-gating capability that allows for an ultra fine-grained logic control that turns on individual processor logic subsystems only if and when they are needed. Additionally, many buses and arrays are split so that data required in some modes of operation can be put in a low power state when not needed.

• Intel Advanced Smart Cache

The Intel Advanced Smart Cache is multi-core optimized cache that improves performance and efficiency by increasing the probability that each execution core of a dual-core processor can access data from a higher-performance, more efficient cache subsystem. To accomplish this, Intel shares Level 2 (L2) cache between cores. By sharing L2 caches among each core, the Intel Advanced Smart Cache also allows each core to dynamically use up to 100 percent of available L2 cache. When one core has minimal cache requirements, other cores can increase their percentage of L2 cache, reducing cache misses and increasing performance. Multi-Core Optimized Cache also enables obtaining data from cache at higher throughput rates.

• Intel® Smart Memory Access

Intel Smart Memory Access improves system performance by optimizing the use of the available data bandwidth from the memory subsystem and hiding the latency of memory accesses. The goal is to ensure that data can be used as quickly as possible and is located as close as possible to where it's needed to minimize latency and thus improve efficiency and speed. Intel Smart Memory Access includes an important new capability called *memory disambiguation*, which increases the efficiency of out-of-order processing by providing the execution cores with the built-in intelligence to speculatively load data for instructions that are about to execute before all previous store instructions are executed.

• Intel® Advanced Digitial Media Boost

The Intel Advanced Digital Media Boost is a feature that significantly improves performance when executing SSE instructions. Both 128-bit SIMD integer arithmetic and 128-bit SIMD double-precision floating-point operations reduce the overall number of instructions required to execute a particular program task, and as a result can contribute to an overall performance increase. They accelerate a broad range of applications, including video, speech and image, photo processing, encryption, financial, engineering, and scientific. SSE instructions enhance the Intel architecture by enabling programmers to develop algorithms that can mix packed, single-precision floating-point and integers, using both SSE and MMX instructions respectively [6], [8], [9], [16].

In 2008, Intel introduced a new dynamically and design scalable microarchitecture that rewrites the book of energy and performance. Intel Micro-architecture (Nehalem) provides a number of distinct feature enhancements over those of the enhanced Intel Core Micro-architecture including:

1. Enhanced" processor core:
   - improved branch prediction and recovery cost from mis-prediction,
   - enhancements in loop streaming to improve front-end performance and reduce power consumption,
   - deeper buffering in out-of-order engine to sustain higher levels of instruction level parallelism,
   - enhanced execution units with accelerated processing of CRC, string/text and data shuffling.
2. Hyper-threading technology (SMT):
   - support for two hardware threads (logical processors) per core,
   - a 4-wide execution engine, larger L3, and large memory bandwidth.
3. "Smarter" Memory Access:
   - integrated (on-chip) memory controller supporting low-latency access to local system memory and overall scalable memory bandwidth (previously the memory controller was hosted on a separate chip and it was common to all dual or quad socket systems),
   - new cache hierarchy organization with shared, inclusive L3 to reduce snoop trace,
   - two level TLBs and increased TLB sizes,
   - faster unaligned memory access.
4. Dedicated Power management:
   - integrated micro-controller with embedded firmware which manages power consumption,
   - embedded real-time sensors for temperature, current, and power,

- integrated power gate to turn o_/on per-core power consumption;
- Versatility to reduce power consumption of memory and QPI link subsystems.

Nehalem implements a number of techniques to process efficiently the stream of Intel64 ISA CISC “macro-instructions" in the user code. A core internally consists of a large number of functional units (FUs) each capable of carrying out an elementary “micro-operation" (micro-op). Micro-operations having no dependencies on the results of each other could proceed in parallel if separate FUs are available. The micro-operations eventually reach the execution FUs where they are dispatched to FUs and “retire", that is, have their results saved back to visible (“architected") state (i.e., data registers or memory).

When all micro-ops of a macro-instruction retire, the macro-instruction itself retires. It is clear that the basic objective of the processor is to maximize the macro-instruction retirement rate. The fundamental approach Nehalem (and other modern processors) takes to maximize instruction completion rates is to allow the micro-ops of as many instructions as feasible, proceed in parallel with micro-op occupying independent FUs at each clock cycle. The entire process proceeds in stages, in a “pipelined" fashion. Pipelining is used to break down a lengthy task into sub-tasks where intermediate results flow downstream the pipeline stages. Complex FUs are usually themselves pipelined. A floating-point ALU may require several clock cycles to produce the results of complex FP operations, such as, FP division or square root.

The advantage of pipelining here is that with proper intermediate result buffering, we could supply a new set of input operands to the pipelined FU in each clock cycle and then correspondingly expect a new result to be produced at each clock cycle at the output of the FU. Dynamic instruction scheduling logic in the processor determines which micro-ops can proceed in parallel while the program execution remains semantically correct. Dynamic scheduling utilizes the “Instruction Level Parallelism" (ILP) which is possible within the instruction stream of a program. Another mechanism to avoid pipeline stalling is called “speculative” execution.

A processor may speculatively start fetching and executing instructions from a code path before the outcome of a conditional branch is determined. Nehalem, as other modern processors, invests heavily into pre-fetching as many instructions, from a predicted path and translating them into micro-ops, as possible. A dynamic scheduler then attempts to maximize the number of concurrent micro-ops which can be in progress (“in-flight") at a time, thus increasing the completion instruction rates. Another interesting feature of Intel64 is the direct support for SIMD instructions which increase the effective ALU throughput for FP or integer operations [6], [8], [9], [17].

Intel launched sandy bridge micro-architecture in 2010. Sandy Bridge was first appeared in desktop and notebook processors that was be announced in early 2011 and branded as “2nd generation Intel Core” processors. It will later roll into a family of server processors. A focus on power efficiency, however, enables Sandy Bridge to achieve more performance within the same limits as its predecessor. Improvements in turbo mode enable even greater performance for short time periods. For notebook computers, these improvements can significantly extend battery life by completing tasks more quickly and allowing the system to revert to a sleep state.

More recently, Nehalem-class processors “integrated” graphics into the processor, but these products actually used two chips in one package. Sandy Bridge includes the GPU on the processor chip, providing several benefits. The GPU is now built in the same leading-edge manufacturing process as the CPU, rather than an older process, as in earlier products. This change alone provides a huge improvement in performance per watt, increasing the speed of the shader engines while reducing their power. The GPU can now access the processor’s large level three (L3) cache (which Intel now calls the last-level cache or LLC). When graphics functions executed in a simple pipeline, caching was irrelevant. A modern shader-based architecture, however, can access the same data over and over as it layers one effect over another. The L3 cache short-circuits the path to main memory, providing its information much more quickly. This extra speed improves graphics performance while reducing pressure on the limited DRAM bandwidth.

As announced by Intel in 2008, AVX is a new set of x86 instruction-set extensions that logically follows SSE4. AVX increases the width of the 16 XMM registers from 128 bits to 256 bits. To implement the AVX instructions, Intel extended the width of the FP multiply unit and the FP add unit to accommodate the wider YMM registers.

Sandy Bridge contains a new component, the system agent that controls what was previously called the north bridge: the memory controller, PCI Express, display interfaces, and the DMI connection to the external south-

bridge chip (PCH). Instead of the single L3 cache used in Nehalem, Sandy Bridge divides the L3 cache into four blocks. Intel has not disclosed details, but we expect each block to be 2MB in size, totalling 8MB of L3 cache in a four-CPU processor.To improve bus bandwidth and simplify scalability, sandy Bridge implements a ring interconnect instead of the usual internal bus. Because each station in the ring connects only to the next station, the ring segments are short enough to operate at the full CPU speed. Furthermore, each ring segment can potentially carry different data.

As with Intel's other multicore processors, the frequency of each CPU core can be adjusted individually, so it can be turned on or off as needed. Even when its clock is turned off, however, the CPU burns leakage power. So, when the processor is truly idle, the voltage to the CPUs can be turned off, completely eliminating their power dissipation. The graphics unit is on a separate voltage plane, so it can be disabled for applications that do not require graphics. The system agent and north-bridge logic are on a third power plane, which allows the north bridge to refresh the display while the rest of the processor is asleep. The system agent contains the power manager—a programmable microcontroller that controls the power state of all units and controls reset functions. This approach allows power-management decisions to be made more quickly than by running a driver on one of the CPUs [6], [8], [9], [14].

V. Conclusion

Current and upcoming era of computer architecture belongs to micro-architecture evolution. Instruction-level parallelism is most appropriate technique to deal efficiently with micro-architecture issues. Superscalar design is preferred choice of leading processors designing companies. Intel started parallelism with Pentium and it continues in present and future. Intel's focuses on all the fields of processor in which it is implemented like server, user workstation etc. Then Intel turns to mobility section of processor with trade-off of performance and power. Performance of processor will depend on instruction-level parallelism provided by individual core using as well as coarse-grain parallelism supplied by multiple cores in multi-core environment and multithreaded environment. After that Intel gives new term to world "Core" and now days, Intel's focal point is multicore impression with multithreaded implementation.

References

[1] Ramakrishna Rau and Joseph A. Fisher, *Instruction Level Parallel Processing: History, Overview and Perspective*, The journal of supercomputing, 1993.

[2] Brian Case, *Intel Reveals Pentium Implementations Details*, Microprocessor Report, 1993.

[3] Donald Alpert, Dror Avnon, *Architecture of the Pentium Microprocessor*, IEEE Micro, 1993.

[4] Glenn Hinton, Dave Sager, Mike Upton, Darrrell Boggs, Doug Careman, Alan Kyker, Patrice Roussel, *The Micro-architecture of the Pentium 4 Processor*, Intel Technology Journal Q1, 2001.

[5] Harsh Sharangpani, Ken Arora, *Itanium Processor Micro-architecture*, IEEE Micro, 2000.

[6] Intel, *Itanium Processor Microarchitecture Reference for software optimization*, 2000.

[7] Intel, *Intel Pentium 4 Processor Optimization Reference Manual*, 2001.

[8] Intel, *Intel 64 and IA-32 Architectures Optimization Reference Manual*, 2012.

[9] Intel, *Intel Quick Reference Guide – Product Family*, dated: 28-05-2012.

[10] James E. Smith, Gurindar S. Sohi, *The Micro-architecture of Superscalar Processors*, 1995.

[11] John Paul Shen and Mikko H. Lipasti, *Modern Processor Design, fundamental of Superscalar Processors*, Tata Mc-Graw Hill Limited, 2005.

[12] Kai Hawng, *Advance Computer Architecture-Parallelism, Scalability, Programmability*, Mc-Graw Hill International Edition, 1993.

[13] Linley Gwennap, *Merced Shows Innovative Design*, Microprocessor Report, 1999.

[14] Linley Gwennap, *Sandy Bridge Spans Generations*, Microprocessor Report, 2010.

[15] Linley Gwennap, *Intel's P6 Uses Decoupled Superscalar Design*, Microprocessor Report, 1995.

[16] Martin Dixon, Per Hammerlund, Stephan, Ronak Singhal, *The Next-Generation Intel Core Microarchitecture*, Intel Technology Journal Volume 14 Issue 3, 2010.

[17] Michael E. Thomadakis, *The Architecture of the Nehalem Processor and Nehalem-EP SMP Platforms*, Research Report, Texas A&M University, 2011.

[18] Nitin Mathur, Rajesh Purohit, *Architectural Features of Processors to Implement Parallelism at Instruction Level in Assembly Level Language*, Proc. of the UGC national conference on New Advances in Programming Language and their implementation, 15-16 March, 2013, ISBN No. 978-81-7233-886-2.

[19] Norman P. Jouppi and David W. Wall, *Available Instruction-Level Parallelism for Superscalar and Super pipelined Machines*, Proc. Third Int. Conf. Arch. Support for Prog. Lang. and OS, ACM Press, 1989.

[20] Siamak Arya, Howard Sachs, Sreeram Duvvuru, *An Architecture for High Instruction Level Parallelism*, Proc. of the 28^{th} Annual Hawaii International Conference on system Sciences, 1995.

[21] Simcha Gochman, Ronny Ronen, Ittai Anati, Ariel Berkovits, Tsvika Kurts, Alon Naveh, Ali Saeed, Zeev Sperber, Robert C. Valentine, *The Intel Pentium M Processor: Microarchitecture and Performance*, Intel Technology Journal, Volume 7 Issue 2, 2003.