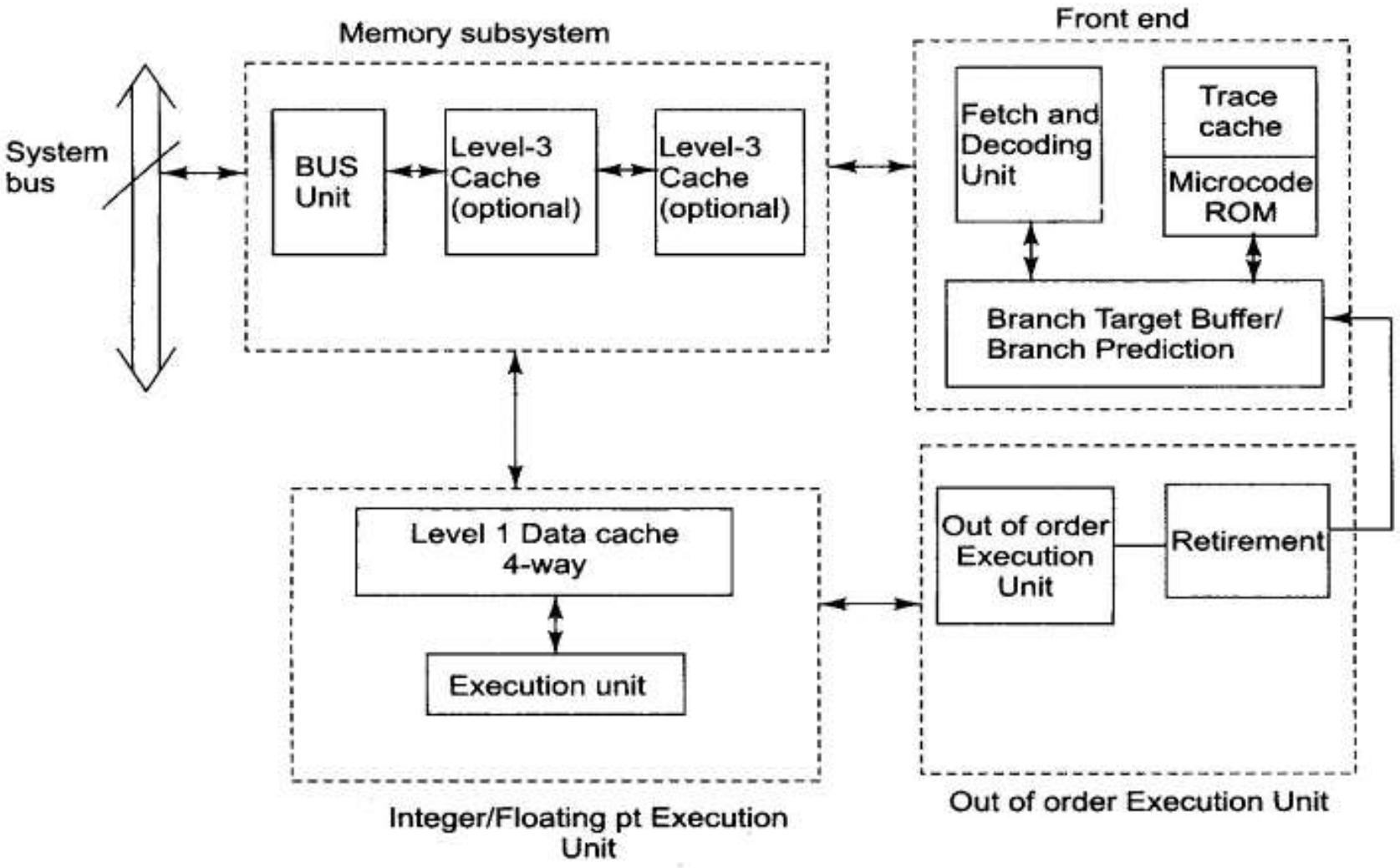
Pentium 4 – Processor of the New Millennium

Salient Features of Pentium – 4

Pentium 4 microprocessor arrived in the scene in June, 2000. After Pentium PRO processor, designed using P6 micro-architecture, which was released in 1995, Pentium-4 is the next x86 processor from Intel. This new processor with Pentium-4 Net Burst architecture utilizes all the features of earlier P6 architecture of Pentium 3 and includes many more. Some of the features of Pentium 4 are as follows:

(i) It is based on NetBurst microarchitecture
(ii) It has 42 million transistors, fabricated using 0.18 micron CMOS process.
(iii) Its die size is 217 sq. mm, and power consumption is 50W
(iv) Clock speed varies from 1.4 GHz to 1.7 GHz. At 1.5 GHz the microprocessor delivers 535 SPECint2000 and 558 SPECfp 2000 of performance.
(v) It has hyper-pipelined technology—Its pipeline depth extends to 20 stages.
(vi) In addition to the L1 8 KB data cache, it also includes an Execution Trace Cache that stores up to 12 K decoded micro-ops in the order of program execution.
(vii) The on-die 256KB L2-cache is non-blocking, 8-way set associative. It employs 256-bit interface that delivers data transfer rates of 48 GB/s at 1.5 GHz.
(viii) Pentium-4 NetBurst microarchitecture introduces Internet Streaming SIMD Extensions 2 (SSF2) instructions. This extends the SIMD capabilities that MMX technology and SSF techonology delivered by adding 144 new instructions. These instructions includes 128-bit SIMD integer arithmetic and 128-bit SIMD double-precision floating-point operations.
(ix) It supports 400 MHz system bus, which provides up to 3.2 GB/s of bandwidth. The bus is fed by dual PC800 Rambus channel. This compares to the 1.06 GB/s delivered on the Pentium-III processor's 133-MHz system bus.
(x) Two Arithmetic Logic Units (ALUs) on the Pentium 4 processor are clocked at twice the core processor frequency. This allows basic integer instructions such as Add, Subtract, Logical AND, Logical OR, etc. to execute in a half clock cycle.
(xi) Advanced dynamic execution.

Net burst Micro Architecture for Pentium – 4



Block Diagram of Pentium 4 Microarchitecture

Net burst Micro Architecture for Pentium – 4

- Front End Module
  - The Front End Module of Pentium 4 processor contains
    - IA 32 Instruction decoder,
    - Trace Cache
    - Microcode ROM
    - Front end branch Predictor

Net burst Micro Architecture for Pentium – 4

- IA 32 Instruction decoder
  - The instructions supported by Pentium 4 are variable length and are supported by many different addressing modes.
  - The role of instruction decoder to decode these instructions concurrently and translate them in to micro operations known as µops.
  - A single instruction decoder decodes one instruction per clock cycle.
  - Some instructions are translated into single µops while others are translated into multiple numbers of µops .
  - In case of a complex instruction, when the instruction needs to be translated into more than four µops , the decoder usually does not decodes such instructions.
  - Rather it transfers the task to a Microcode ROM.

Net burst Micro Architecture for Pentium – 4

- Trace Cache (TC)
  - The basic function of front end module is to fetch the instructions to be executed, decode them and feed decoded instruction to the next module, which is the out of order execution module.
  - The instructions are first decoded into basic micro operations known as µops, and the stream of decoded instructions are fed to a level - 1 (L1) instruction cache.
  - This special instruction cache is known as Trace Cache, which is a special feature of Pentium micro architecture.
  - It is special because it does not store the instructions but decoded stream of instructions, i.e micro operations or µops, thus enhancing the execution speed considerably.

Net burst Micro Architecture for Pentium – 4

- Microcode ROM
  - When some complex instructions like interrupt handling or string manipulation etc. appear, Trace cache transfers the control to a micro code ROM, which stores the µops corresponding to these complex instructions.
  - When control is passed to the microcode ROM, the corresponding µops are issued.
  - After the µops are issued by the microcode ROM, the control goes to the trace cache once again.
  - The µops delivered by the trace cache and the microcode ROM are buffered in a queue in an orderly fashion.
  - The resultant flow of µops is next fed to the execution engine.

Net burst Micro Architecture for Pentium – 4

- Front end branch Predictor in Pentium 4
  - The other important unit in the Front end is the branch Prediction logic unit.
  - This unit predicts the locations from where the next instruction bytes are fetched.
  - The predictions are made based on past history of the program execution.
  - The earlier generation processor follows simple branching strategy.
  - When the processor comes across a branch instruction, it evaluates the branch condition.
  - The condition evaluation may involve a complex calculation, which may consume time and the processor has to wait till the condition is computed and thereafter it decides whether to take the branch or not.

Net burst Micro Architecture for Pentium – 4

- Branch Prediction
  - The modern day fast processors cannot wait till the branch condition is evaluated to decide whether to take a branch since this will unnecessarily slow down the speed of execution.
  - These processors take the strategy of speculating whether the branch condition will be satisfied.
  - Pentium, for example, makes a guess about the branching using strategy called speculative execution.
  - This strategy involves making a guess at which direction the branch is going to be taken and then branching at the new branch target even before the branching condition is actually evaluated.
  - Many strategies have been suggested for speculative prediction and the guess is made used one of these branch prediction strategies.

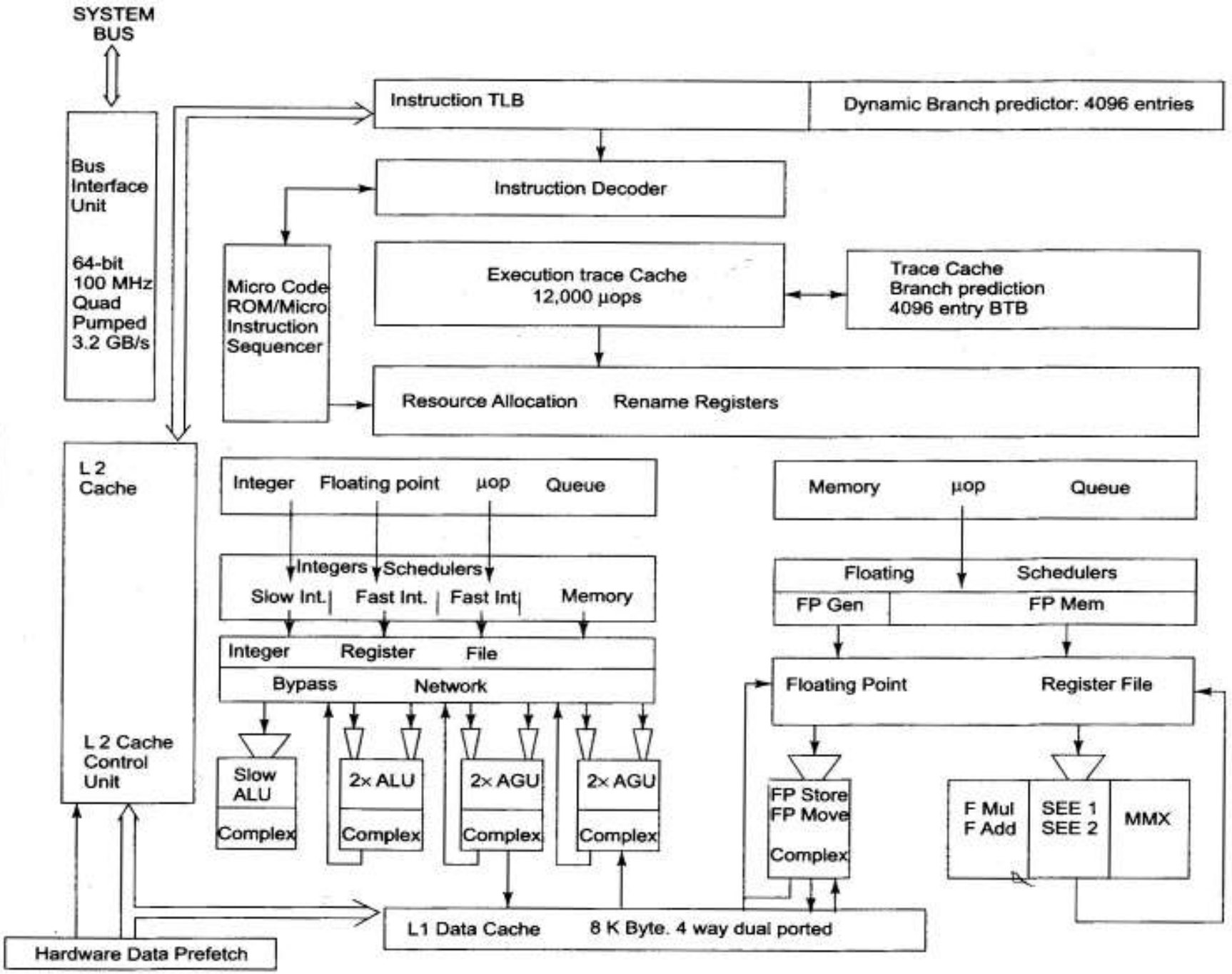
Net burst Micro Architecture for Pentium – 4

- If, however the processor incorrectly predicts a branch it may lead to severe problem.
- In case of incorrect prediction, these instructions are fetched from the wrong branch predictions and may be executed for incorrectly for a wrong speculation.
- In such a case, the pipeline has to be flushed of the erroneous speculative instructions and results.
- After flushing out the wrong instructions, the instructions from the correct branch address are fetched, and executed.
- Flushing the pipeline of instructions and results is expensive and produces a delay of several cycles.
- The delay depends on the level of pipelining in the processor.
- Also there is a delay associated with loading the new instruction stream.
- The resulting delay invariably degrades the system performance significantly, if such erroneous predictions take place often.

Net burst Micro Architecture for Pentium – 4

- As the length of pipeline in a processor increases, the degradation also increases proportionately.
- In case of a wrong prediction, Pentium – 4 with 20 stages will have to wait for considerable number of cycles, while new instructions are loaded from the cache.
- The P4 has minimum loss of 19 clock cycles due to each erroneous prediction.
- This is the loss incurred when the code resides in the L1 cache.
- The loss will be higher if the correct branch is not found in the L1 cache, since in that case the data has to be fetched from L2 cache.

D



Detail Architecture of Pentium Processor

Instruction Translation Look Aside Buffer (ITLB) And Branch Prediction

- If there is a trace cache miss, then instructions bytes are required to be fetched from the L2 cache.
- These are next decoded into μops to be placed in the Trace cache (TC).
- The instruction Translation look aside Buffer (ITLB) receives the request from the TC to deliver a new instruction, and it translates the next instruction pointer address to a physical address.
- A request is sent to the L2 cache, and instruction bytes are returned.
- These bytes are placed into streaming buffers, which hold the byte until they are decoded.

Instruction Translation Look Aside Buffer (ITLB) And Branch Prediction

- Since there are two logical processors there are two ITLBs.
- Thus each logical processor has its own ITLB and its own instruction pointer to track the progress of instruction fetch for each of them.
- Now suppose both the logical processors request the access of L2 cache, the instruction fetch logic performs arbitration based on which processor request has arrived first.
- Accordingly, it sends requests to the L2 cache and grants the request of the first processor.
- It, however, reserves at least one request slot for each logical processor.
- In this way, both logical processors can access and fetch data from L2 cache without any conflict.

Instruction Translation Look Aside Buffer (ITLB) And Branch Prediction

- Before the instructions are decoded, they are stored in streaming buffers.
- Thus each logical processor has its own set of 64 byte streaming buffers, which store the instruction bytes and subsequently they are dispatched to the instruction decode stage.

Out of order Execution Engine

- allocation, register renaming, scheduling, execution
- Allocator logic –
  - 126 reorder buffer entries
  - 128 integer, 128 floating point physical registers
  - 48 load and 24 store buffer entries
- register rename
  - rename onto machine's physical registers
  - 8 general purpose registers --→ expanded to use 128 physical registers
  - Register alias table
- Instruction Scheduling
  - 5 schedulers

Rapid Execution Module and Memory Subsystem

RAPID EXECUTION MODULE

Pentium 4 has two ALUs (Arithmetic Logic Unit) and two AGUs (Address Generation Unit), which run at twice the processor speed. This implies that the ALUs in a 1.4 GHz processor works at 2.8 GHz. The doubled speed of these units means twice the number of instructions being executed per clock cycle.

Arithmetic and Logic Unit is responsible for carrying out all integer calculations (add, subtract, multiplication, division) and logical operations. AGUs are primarily used to resolve indirect mode of memory addressing. As can be comprehended, these units are quite important for high-speed processing which includes frequent fetching of instructions and arithmetic calculations.

MEMORY SUBSYSTEM

The memory subsystem involving virtual memory and paging is briefly described below.

Paging and Virtual Memory

With the flat or the segmented memory model, linear address space is mapped into the processors physical address space either directly or through paging when using direct mapping (paging disabled), each linear address has a one-to-one correspondence with a physical address. Linear address bits are sent out on the processor's address lines without translation.

When using IA-32 architecture's paging mechanism (paging enabled) linear address space is divided into pages which are mapped to virtual memory. The pages of virtual memory are then mapped as needed into physical memory when an operating system or execution uses paging. The paging mechanism is transparent to an application program. All that the application sees is linear address space.

In addition, IA-32 architecture's paging mechanism includes extension that support:

- Page Address Extensions (PAE) to address physical address space greater than 4G Bytes.
- Page Size Extension (PSE) to map linear address to physical address in 4 M bytes page.

Cache

The access to DRAM main memory is often very slow. To enhance the speed of data access fast SRAM caches are used to reduce this latency