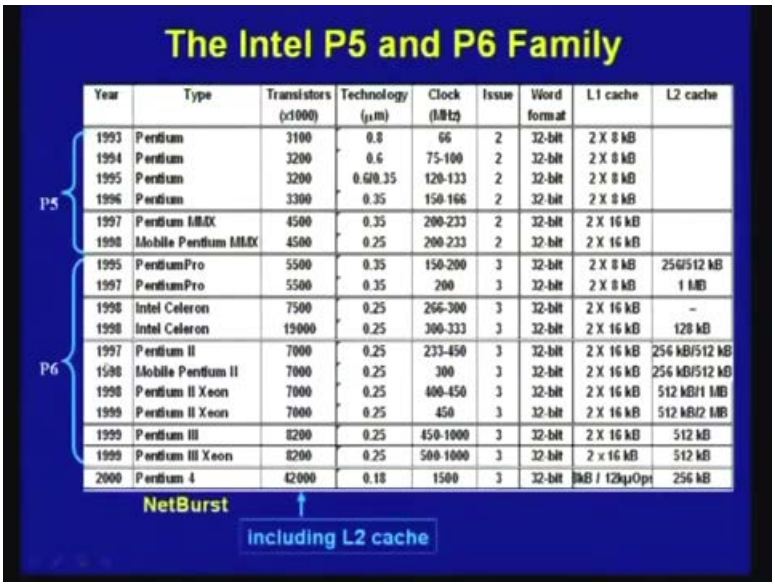
High Performance Computer Architecture
Prof. Ajit Pal
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 35
Case Studies (Contd.)

Hello and welcome to today's lecture on Case Studies, we shall continue our discussion on the evolution of Intel processors. In the last lecture, we have discussed about various processors starting with that Intel 4 0 0 4, that was introduced in the year 1971.

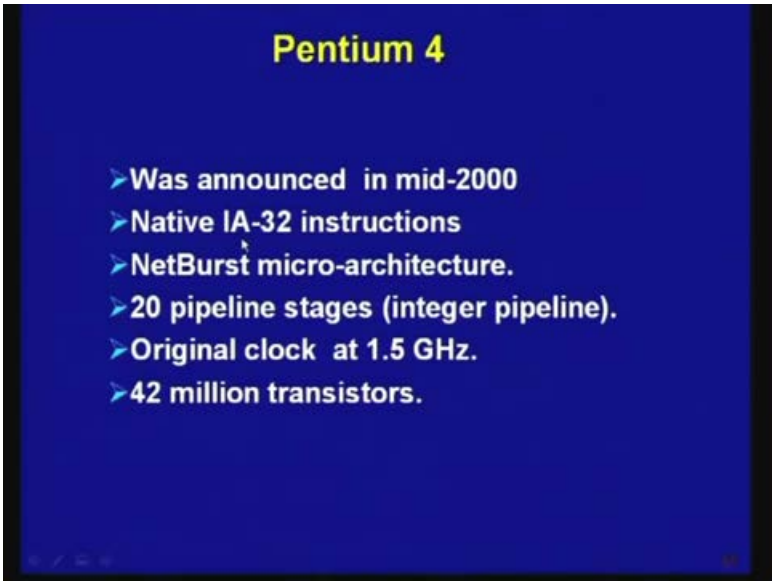
(Refer Slide Time: 01:04)



And subsequently 8 bit, 16 bit and the 32 bit processors were introduced, and we have discussed about different types of p 4, p 5 and p 6 processors like Pentium, Pentium pro, Pentium 2, Pentium 3, which are based on these p 4, p 5 and p 6 architecture. Today, we shall focus on Pentium 4, which was introduced in the year 2000, based on Netburst architecture, we shall going to the detail of this micro architecture.

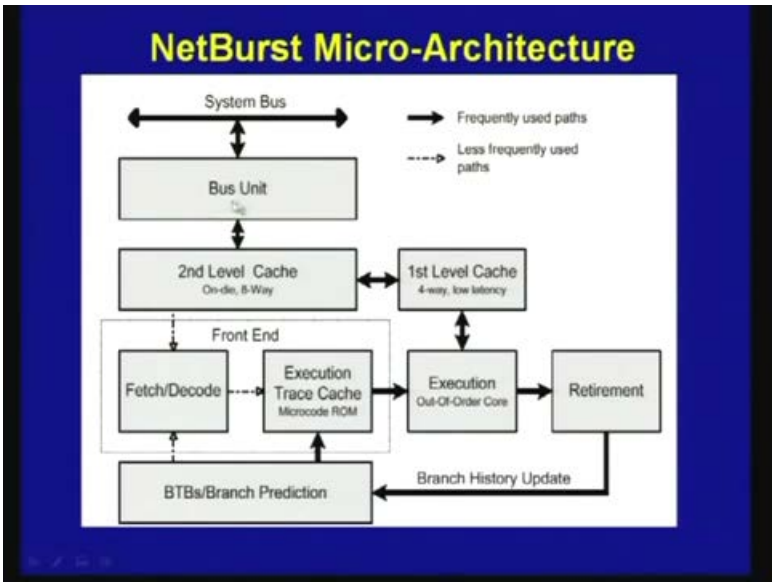
And it use it required 42 million transistors including a L 2 cache based on 0.18 micron technology, and clock speed was initially 1.5 Giga Hertz, subsequently of course, it was increased. And it is a 3 issue processor having 8 kilo byte of cache memory, and 128 k micro operations, that is being stored in their trace cache, as we shall discuss and it is having 2 level cache, L 1 and L 2, in L 2 it has got 256 kilobyte of cache memory.

(Refer Slide Time: 02:37)



So, as I mentioned it was announced in mid 2000, and it uses native IA 32 instructions; that means, it is compatible with other processors like Pentium, pro Pentium, Pentium 2 and Pentium 3. It has actually maintained the downward compatibility; that means, if you write a program, in for Pentium, it will work on Pentium 4, so all the instructions which are used in Pentium are being used in Pentium 4, that is called downward compatibility. And, as I mentioned it is based on NetBurst micro-architecture, I shall be going to details of this, and it has got twenty pipeline stages, integer pipeline stages, and it has used 1.5 Giga Hertz clock, and it uses 42 million transistors.

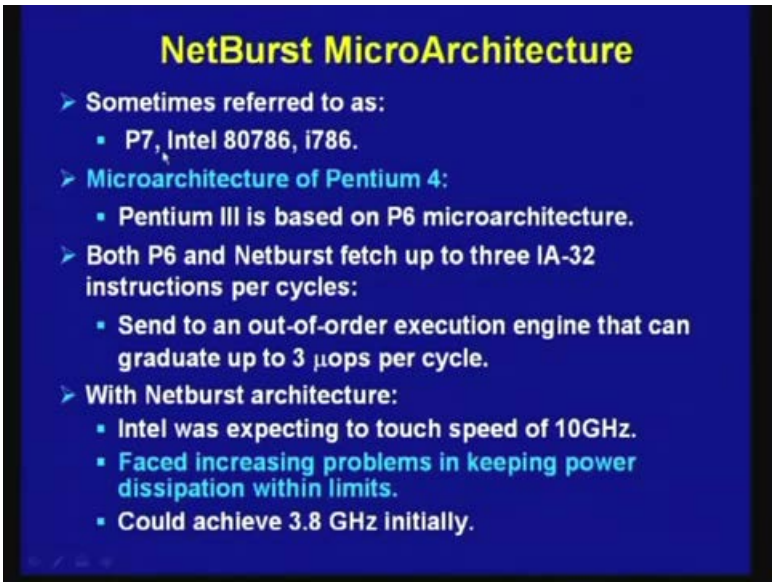
(Refer Slide Time: 03:34)



So, this is the basic micro architecture of Pentium 4, so you can see here it has got that bus interface unit, which is written here bus unit, which actually interfaces with the system bus; that means, memory and I/O devices. And it has got 2 level cache, first level cache which is 4 way having very low latency; that means, it is pretty fast, and of course, it has got second level cache, which is also on-die 8 way cache memory.

And in the front end it has got fetch the code unit, and also a special type of cache known as execution trace cache, and also it has got microcode ROM, I shall discuss about these things in more details. And execution takes place in out of order, so it is supports out of order execution, and it uses branch prediction using branch target buffer, it has got many branch target buffers. And so it does branch prediction, and the branch history is stored in those branch target buffers, and it has got execution unit out of order and the retirement unit where the writing into the registers takes place.

(Refer Slide Time: 05:08).



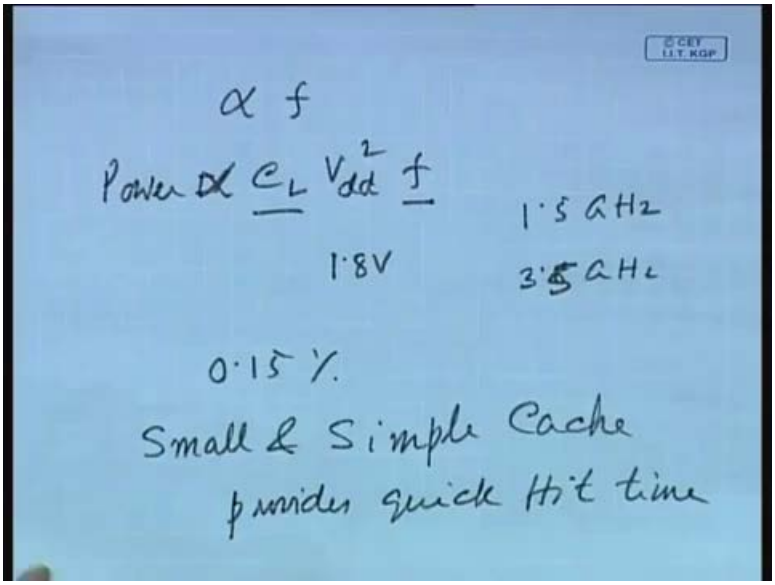
So, this gives a overview of the NetBurst architecture, and it is also I mean sometimes it was referred to us P 7, because earlier p 5 and p 6 architecture were introduced, since it was introduced after them. It was also called p 7, but later on this was not, terminology was not used much and also called Intel 80786 or i 786, so same processor called in different ways, Pentium 4 is the most commonly used name.

And so just like Pentium 3 it is also based on p 6 micro architecture, both p 6 and NetBurst fetch up to 3 IA 32 instructions per cycles. So, it fetches multiple instructions

per cycle, and send an out of order execution engine, so those instructions are sent to out of order execution in that can graduate up to 3 micro operations per cycle. So, as I already mentioned each instruction is decoded into several micro operations, minimum 2, maximum 4 micro operations, and sometimes more number of micro operations, when the instruction is very complex.

And in each you know out of order execution 3 micro operations per cycles are executed, and with this net burse architecture Intel was expecting to touch speed of 10 Giga Hertz, but unfortunately that was not feasible. Primarily, because of power dissipation, as you know when the clock frequency is increasing.

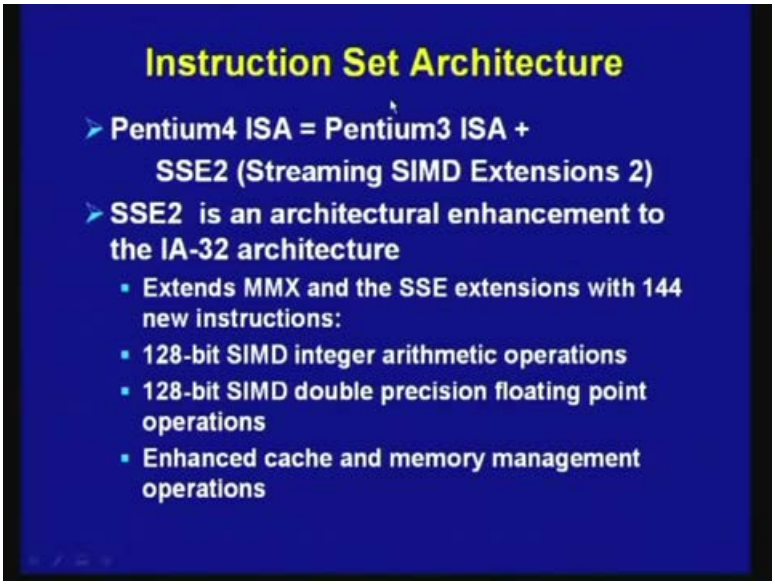
(Refer Slide Time: 06:58)



Power dissipation also increases because power dissipation is proportionate to frequency actually, more precisely C L V d d square f, that is the power dissipation, power is proportional to this, and C L actually presents overall capacitance supply voltage and that is the f is a clock frequency. And despite of reduced supply voltage as you have seen in case of Pentium 4 1.8 voltage supply, volt is used that clock frequency cannot be I mean cannot be increased to 10 Giga Hertz.

So, in spite of their effort Intel failed to do, so and they could achieve up to 3.8 Giga hertz initially, but of course as I mentioned initially 1.5 Giga Hertz processor were commercially available, subsequently they were increased to 3.8 Giga hertz or 3.5 Giga Hertz anyway.

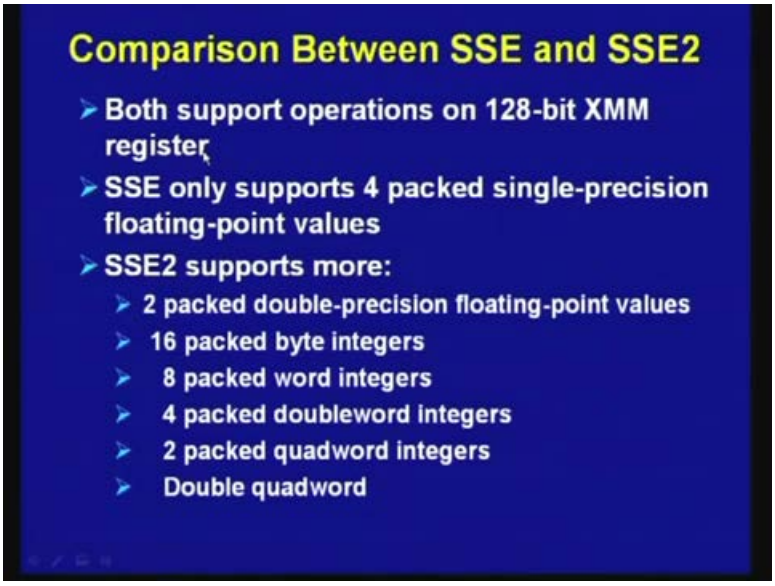
(Refer Slide Time: 08:05)



So, let us have a look at the instruction set architecture of Pentium 4, the Pentium instruction set architecture 4 comprises Pentium 3 instruction set architecture for architecture comprises Pentium 3 instructions set architecture plus SSE. SSE 2 stands for Streaming SIMD Extensions 2, so these streaming SIMD extensions 2 were a additional thing that is provided in Pentium 4, and SSE 2 is an architecture enhancement to the IA 32 architecture.

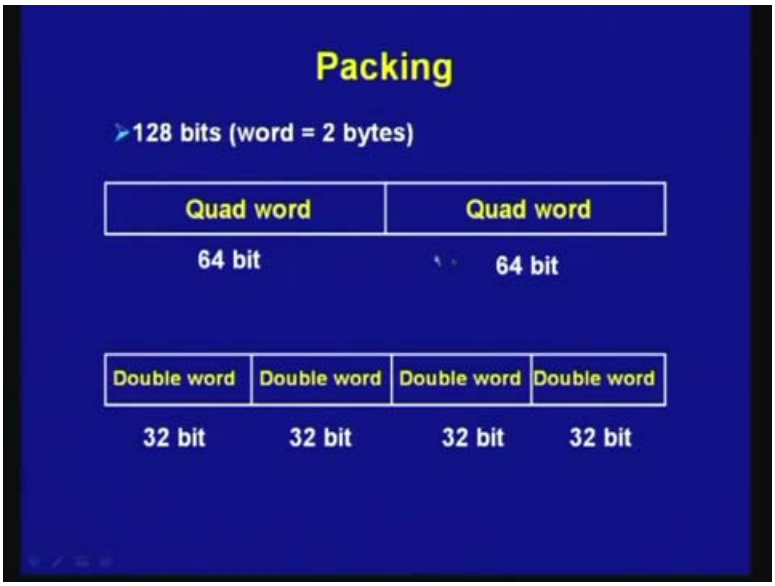
Earlier, the Pentium 3 processor was based on IA 32 architecture, but this was the additional components, so which uses it extends I mean it extends MMX and the SSE extensions with 144 new instructions, 128 bit SIMD integer arithmetic operations, 128-bit SIMD double precision floating point operations. And also it allows enhanced cache and memory management operations, so with this enhanced features; obviously, the performance of Pentium 4 is expected to be higher than Pentium 3.

(Refer Slide Time: 09:20)



So, here is the comparison between SSE and SSE 2, so both supports the operations on 128 bit XMM register, so there are different types of registers as we shall see, and there are 128 bit XMM registers available. And this SSE only supports 4 packed single precision floating point values, but in case of SSE 2 it supports much more varieties as you can see. It can support 2 packed double precision floating point values or 16 packed byte integers, or 8 packed word integers word means 2 byte 4 packed double word integers or 2 packed quadward integers and double quadward. So, these are the various variations, that is supported by SSE 2 architecture.

(Refer Slide Time: 10:14)

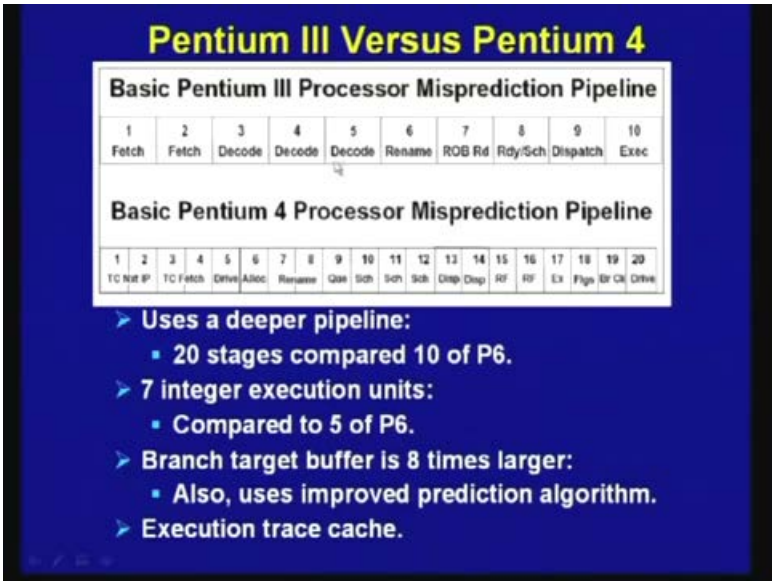


For example, it is how the packing is done that 128 bit packing is done, where each word is considered to be 2 bytes, so the quadward, double quadward here 64 bit and 64 bit each is shown. And diagram below this it has got quad double word, that is fine quad double word so each packed double word, 4 packed double word. So, you can see each of 32 bit and that can be packed in a 128 bit so; that means, this packing can be done for the purpose of increasing the memory utilization. So, in each 128 bit you can storing just one word, you can pack more number of wards, so that efficiency or utilization of the memory is more.

(Refer Slide Time: 11:22)

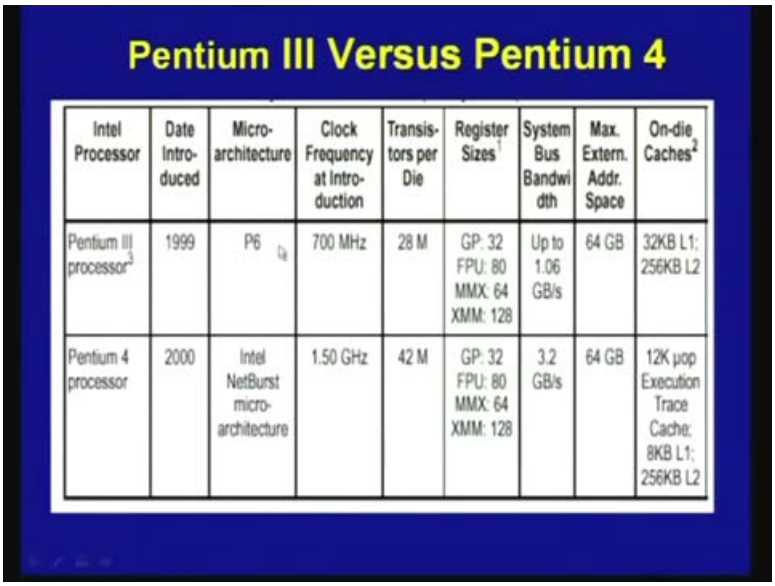
And this is the hardware support for SSE 2 those various functions, that I have mentioned corresponding to a SSE 2 is supported by the following hardware adder, and multiplier units in the SSE 2 engine are of 128 bits wide, and twice the width of in Pentium 3. In case of Pentium 3 those adder and multiplier were of 64 bits, but here it is 128 bits, and increase bandwidth in a load store for floating point values, and because it can do load and store are 128 bit. And 1 load plus 1 store can be completed between XMM register and L 1 cache in one clock cycle, so this will help in enhancing the transfer of data between the L 1 cache and the XMM register which are used for performing the operations.

(Refer Slide Time: 12:19)



So, this shows the pipeline pipelining that is being done pipeline architecture, as I mentioned Pentium 3 uses ten stage pipeline, so you can see that fetch decode, then decode and execute, and all this operations are divided into 10 stages. In case of Pentium of 3 on the other hand these are divided in 20 stages in Pentium 4, so 20 stages compared to 10 of Pentium 6, and it has got seven integer execution units compared to 5 of Pentium 6, and branch target buffer is 8 times larger. So, it uses branch target buffer, and it also uses improved prediction algorithm, which I shall discuss little later and it uses a special type of cache known as execution trace cache, which I shall also discuss in detail.

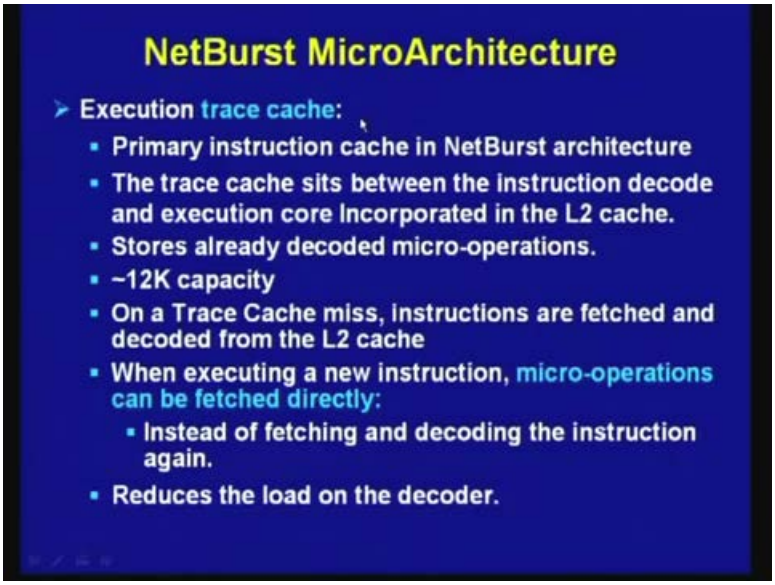
(Refer Slide Time: 13:18)



And this is another comparison we can see the micro architecture for Pentium 4 is Netburst compared to p 6 or Pentium 3, and clock frequency is enhanced from 700 Mega Hertz to 1.5 Giga Hertz. The number of transistors is more 42 million compared to 28 million of Pentium 3, the here are the different registers general purpose 32 is same fourteen point unit registers 80 same, MMX 64 same, but XMM 128 same.

So, the register size is more or less same, and this system bus bandwidth is increased from 1.06 gigabyte per second, to 3.2 gigabyte per second, so maximum external address space is also increased from I mean also same 64 gigabyte for both the cases. And on die caches sizes have increased, but with a different variety for example, it can use execution trace cache, having 128 k micro operations and it has got 2 levels, of cache L 1 8 kilobyte and L 2 256 kilobyte.

(Refer Slide Time: 14:35)



So, let me discuss in little bit detail about the execution trace cache, which is a new invention that is being used in Pentium 4, so this is the primary instruction cache used in Netburst architecture. So, this execution trace cache is essentially the instruction cache, but we shall see it is not used in the conventional way, the way the instructions are stored in a conventional cache here it is not done in the same way.

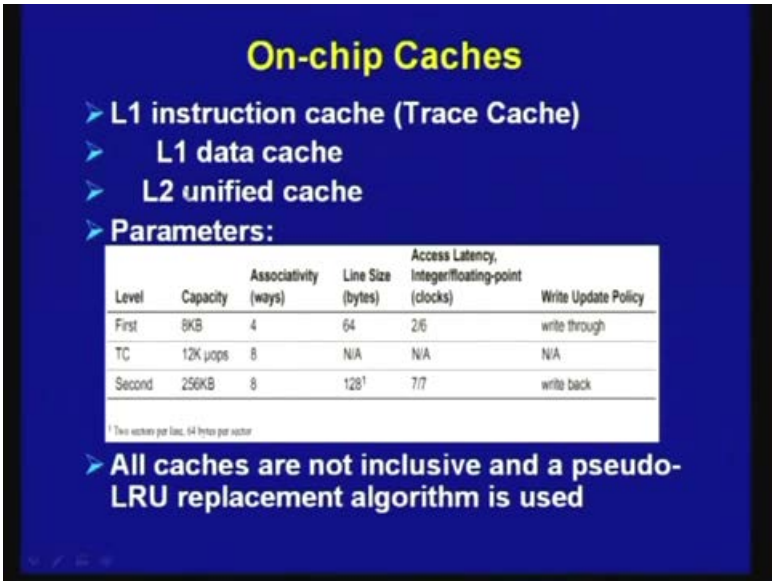
So, this trace cache sits between the instruction decode and execution core incorporated in the L 2 cache, and it stores already decoded micro operations, so here is the difference instead of storing the instructions here it stores the micro operations. That means, there is

no need for re decoding, straight away micro operations are stored, and they can be fetched and they can be executed directly.

So, conventionally instructions are stored in cache, and then they are decoded before it goes to the execution unit, but here it is done, here it is not so. So, on a trace cache means instructions are fetched and decoded from the L 2 cache, so whenever there is a hit then those micro operations are fetched directly delivered to the execution units. On the other hand whenever there is a miss the instructions are fetched from the L 2 cache, and the then they are decoded, and after decoding they are stored in the trace cache, and it has been found that the miss-rate is very small that mean point.

It has been found to be 0.015 percent so; that means, the reading from the L 2 cache is not very frequently, it occurs occasionally; that means, 0.015 percent means it occurs very infrequently. So, can then execution new instruction micro operations can be fetched directly, as I have already mentioned instead of fetching and decoding the instruction again, so this reduces the load on the decoder, so and also it makes it faster.

(Refer Slide Time: 17:12)



And here some details about the cache memories on chip cash as I told, it has got L 1 instruction cache, which is the trace cache and L 1 data cache, and L 2 unified cache. So, that level 2 cache is unified 1, where both instruction and data are stored instead of separate, separate data and instruction cache. However, the L 1 cache is separate and first

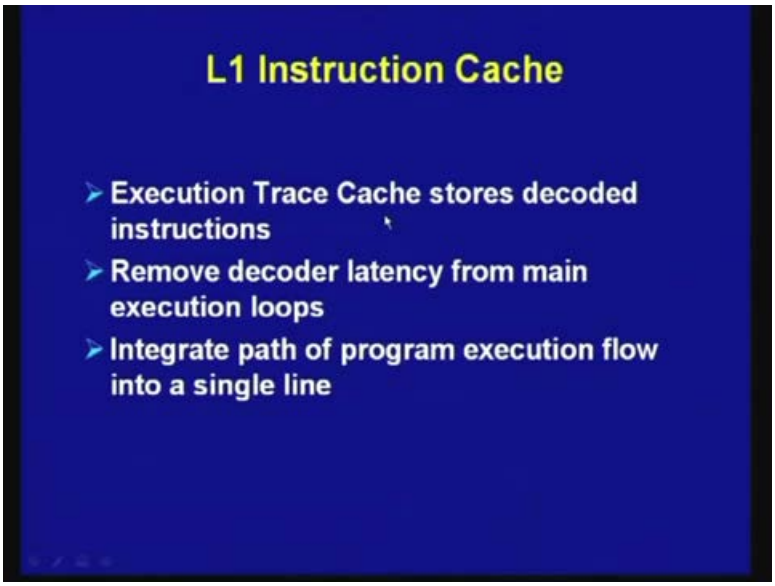
one is the trace cache and instruction trace cache, and the data cache is also there is a separate L 1 data cache.

And here is the size of the different cache's fist level cache is only 8 kilobyte, so this 8 kilobyte has been used primarily to make it small and simple. ((Refer Slide Time: 06:58)) As, we know that small and simple cache gives you provides quick hit time, which has been achieved by providing this sample small L 1 cache 8 kilobyte, which is 4 associativity and line size is 64 bytes.

And it can access latency 2-6 that is integer is 2 cycles per integer data, and on the other hand for floating point data is 6 cycles, and it uses write through. Similarly, the trace cache as I mentioned it stores the micro operations store, 12 k micro operations, it is also 8 ways associativity, and the second level cache is 256 kilobyte, which is 8 associativity and line size is 128 bytes, so 2 sectors per line it uses and 64 bytes per sector. So, here the access is 7 cycles, and it uses write back policy, so all cases are not inclusive, and a pseudo-LRU; that means, inclusion property is not strictly followed.

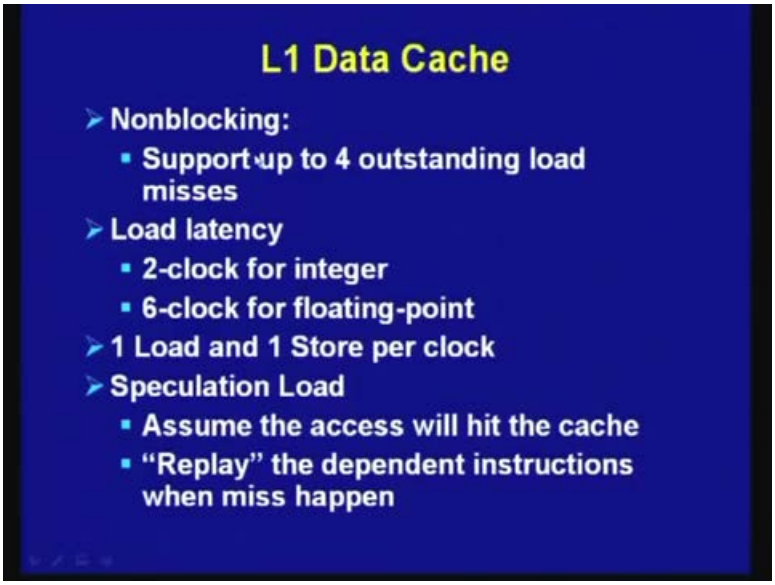
As, we know the lower level cache contains whatever is present in L 2 cache, normally presents is also present in L 1 cache, but this inclusion property is not always maintained in this processor, and it also uses pseudo-LRU replacement algorithm when replacement has to be done.

(Refer Slide Time: 20:09)



So, as I mentioned the L 1 instruction case is execution trace cache, stores the decode instructions it removes the decoder latency from main execution loops, and integrate path of program execution flow into a single line. So, this essentially enhances the execution of the instructions, because the decoder latency is not present in the execution loop, when the execution takes place there is no need for decoding is done at the time of replacement.

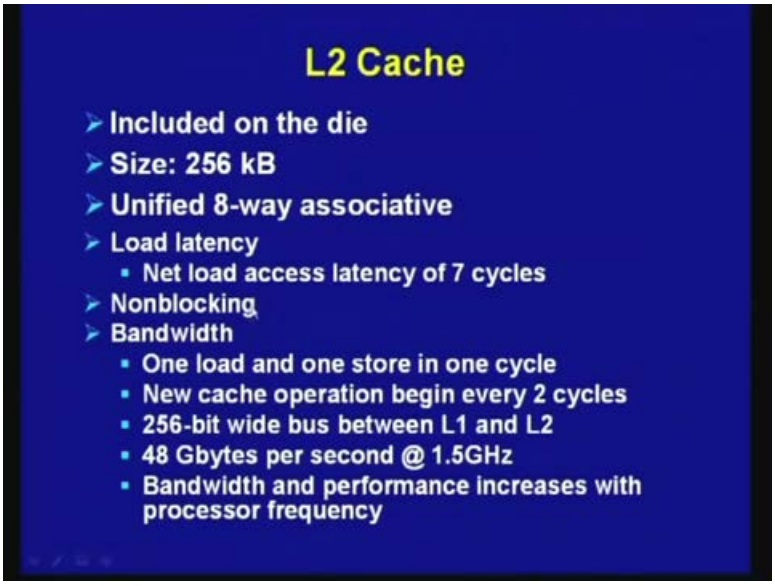
(Refer Slide Time: 20:48)



So, L 1 data cache is non blocking, non blocking means I have already explained the importance or what do you really mean by blocking. Normally, when it is blocking whenever there is a miss, then that it is being stopped then you have to read it from the lower level and then from the next higher level then you proceed, but here it is not done. So, it is supports up to four outstanding load misses; that means, it will continue to continue operations, execution and up to 4 outstanding load misses.

And as I mentioned load latency is 2 clock per integer, and 6 clock per integer for floating point, and one load and one store per clock is performed. And it also does a speculation load as assumes the access will hit the cache, and replay the dependent instructions when miss happens, so speculation load is also a new feature invention that is being used in Pentium 4.

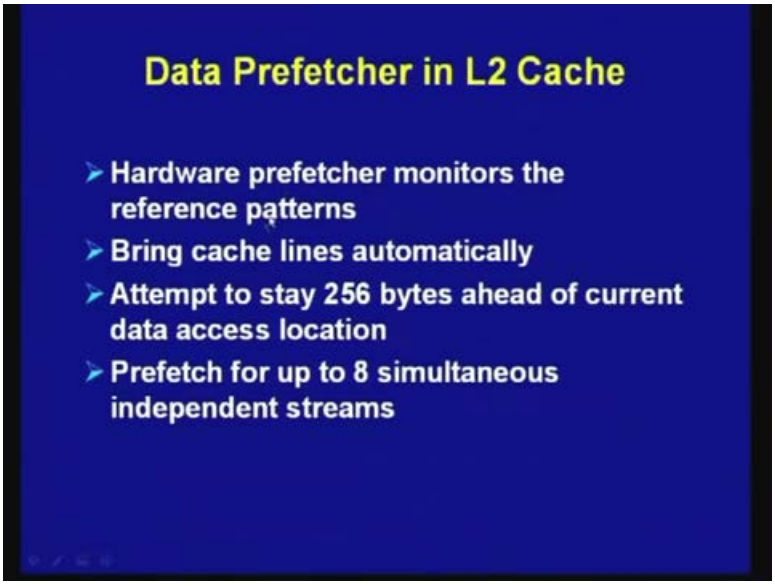
(Refer Slide Time: 22:04)



Then L 2 cache is included on die as I mentioned, then it has got 256 kilobytes of size and it uses unified 8 way associative, and load latency is 7 cycles, it is also non blocking. And bandwidth is 1 load and 1 store in 1 cycle, new cache operation begin every 2 cycles, 256 bit wide bus between L 1 and L 2, so this higher width allows quick transfer of data between L 1 and L 2.

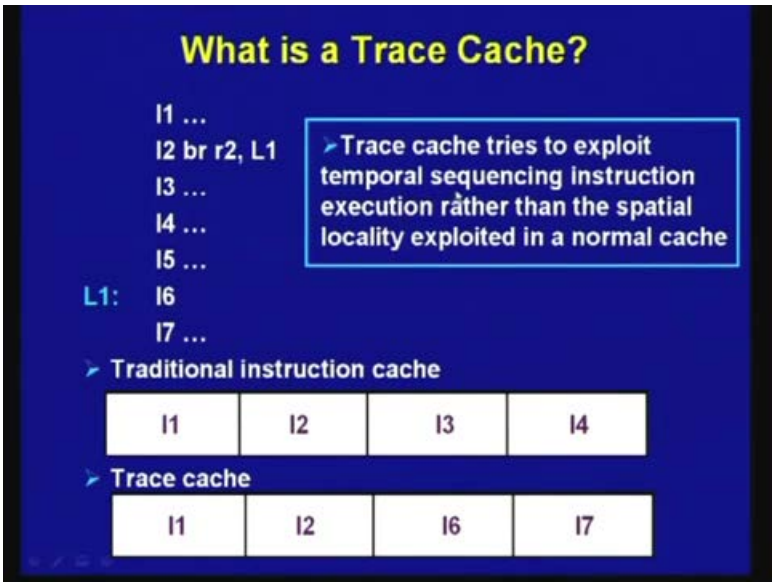
And 48 gigabytes per second, that is your external bus 1.5 gigahertz, that is the bandwidth and performance increases with processor frequency, so as the processor frequency increases this gives you higher performance as a frequency of the processor increases.

(Refer Slide Time: 23:14)



Now, it also does perfecting of instructions, so hardware prefetcher monitors the reference patterns, based on the reference patterns it brings cache lines automatically. So, from the main memory automatically the cache lines are brought using the prefetcher, and it attempts to stay 256 bytes ahead of current data access location, so this prefetching also helps you to get the data quickly instead of reading it from the memory. So, prefetch for up to 8 simultaneous independent streams, so this is another important feature, so simultaneous independent streams can be prefetched and stored in the prefetcher memory or buffering.

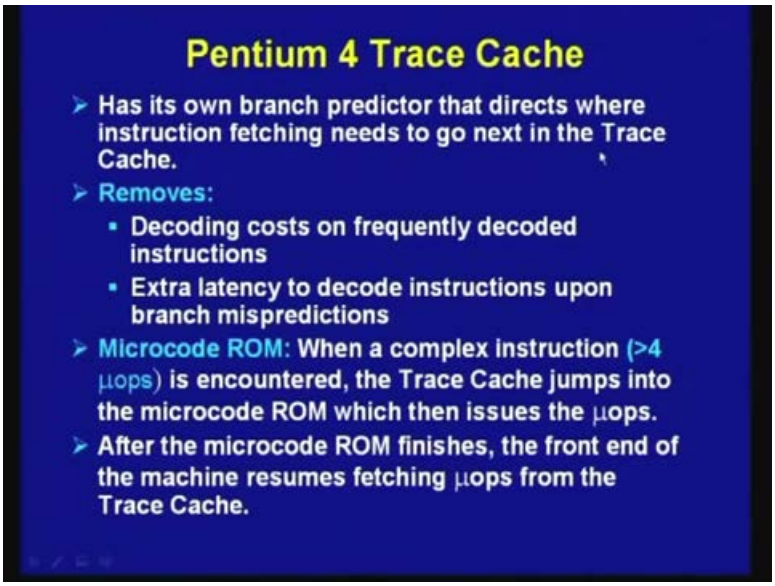
(Refer Slide Time: 24:10)



So, I was mentioning about the trace cache, trace cache tries to exploit temporal sequencing instruction execution rather than the spatial locality exploited in a normal cache, which is illustrated with the help of this simple example. Suppose, it is the instruction sequence I 1, I 2, I 3, I 4, I 5, I 6, I 7, normally in traditional instruction cache you will be storing I 1, I 2, I 3, and I 4 based on spatial locality.

So, based on spatial locality this is being done in traditional instruction cache, but in case of trace cache, which is based on temporal sequencing of instructions it will store I 1, I 2, I 1, I 2 followed by I 6 and I 7, not I 3 I 4 and I 5. Because, that satisfies the temporal based on the temporary locality, and this is how it is being done, and this is one feature of fetching the cache memory from the trace cache.

(Refer Slide Time: 25:22)

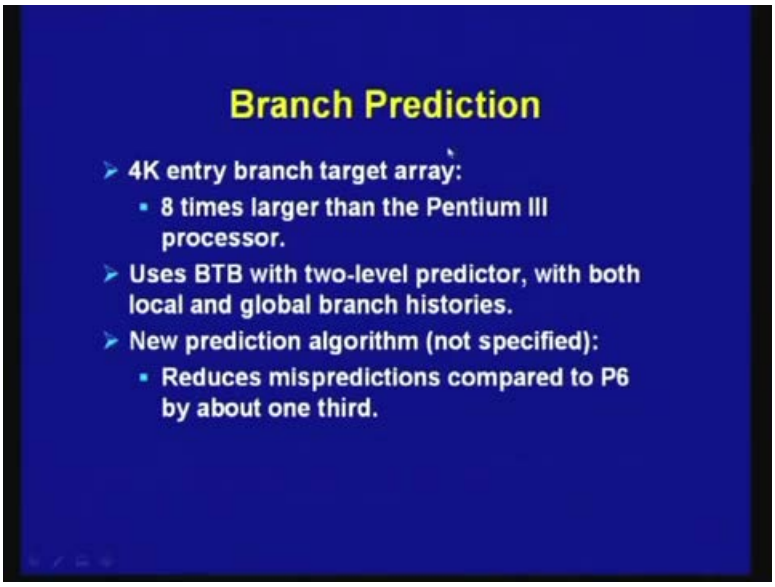


And the Pentium 4 trace cache has its own branch prediction that directs, where instruction fetching needs to go next in the trace cache, so whenever there is a trace cache miss. That branch predictor helps you to direct where the instruction fetching has to take place, so it removes the decoding costs, on frequently decoded instructions, extra latency to decode instructions upon branch mispredictions.

So, and it also uses microcode ROM, microcode ROM when a complex instruction is encountered, the trace cache jumps to the microcode ROM, which then issues the micro operations. So, this use of microcode ROM is also a new invention that is being done provided in Pentium 4, and after the microcode ROM finishes the front end of machine

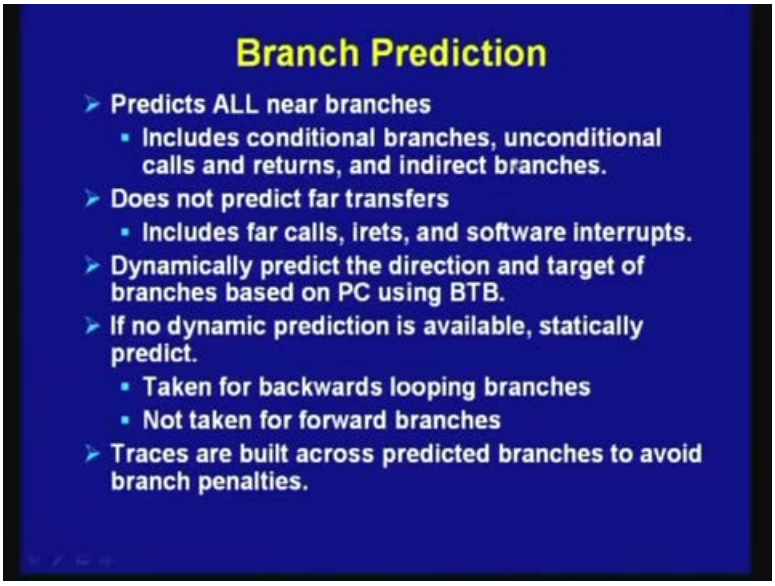
resumes fetching micro operations from the trace cache. So, this is how it combines the use of trace cache, and the microcode ROM for executing the micro operations.

(Refer Slide Time: 26:44)



Then the branch prediction it uses 4 k entry the branch target array, so this is also 8 times larger than Pentium 3 processor, and it uses the branch target buffer with 2 level predictor. 2 level predictor means it has got both local and global branch, it stores both and global branches histories, and based on that prediction is done. And it uses a new prediction algorithm unfortunately details of algorithm is not available is not known, because it is a classified type of predictional algorithm that is being used by Intel. And based on this predictional algorithm, it reduces mispredictions compared to p 6 by about one third, so that p 6 architecture which are used up to Pentium 3, it gives you much better results, so mispredictions is much reduced.

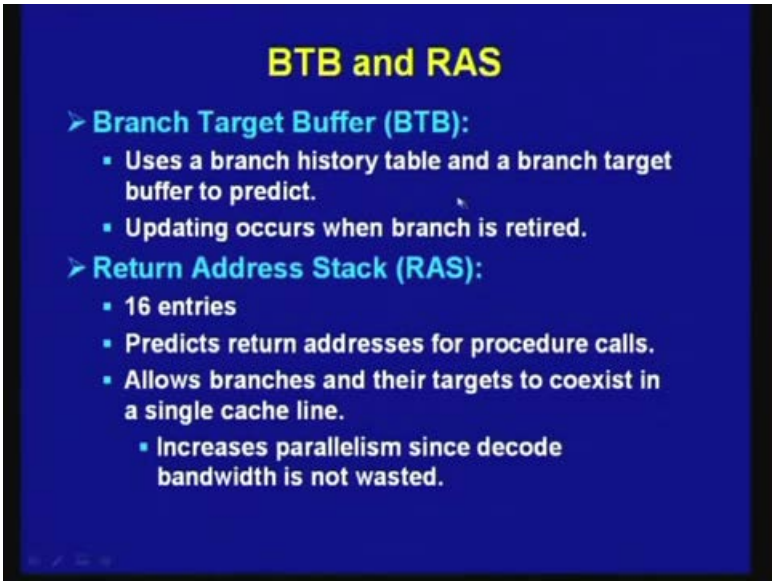
(Refer Slide Time: 27:56)



And branch prediction whenever the branch prediction is done, it predicts all near branches and it includes conditional branches, unconditional calls and returns and indirect branches. So, and it does not predict far transfers, it includes far calls interrupt returns and software interrupts, so it does not predict those far transfers only try to predict all the near branches; that means, within the page. And dynamically predict the direction and target branches based on pc using branch target buffer, so it uses branch transfer buffer which gives you the branch address.

And if no dynamic prediction is available then it does the prediction statically; that means, if that branch history either local or global is not available, then it tries to predict in a statistical way, taken for backwards looping branches and not taken for forward branches. So, it combines the prediction is dynamic in nature, it can be either based on those 2 level prediction based on local and global predictors, or whenever that dynamic prediction is not available. Then it is used as statically predict thing that branch taken or not taken, and traces are built across predicted branches to avoid branch penalties.

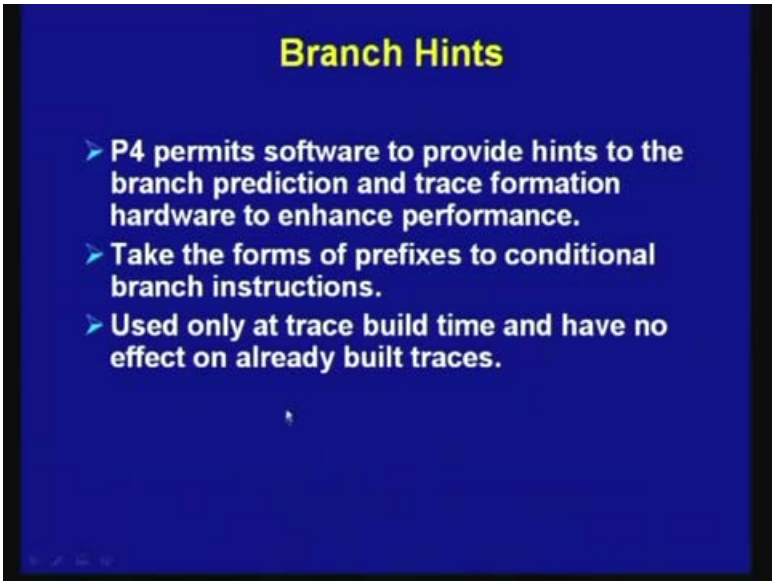
(Refer Slide Time: 29:39)



So, it uses branch target buffer uses a branch history table, and a branch target buffer to predict branch history table and branch target buffer. So, updating occurs when a branch is retired; that means, branches is found to be correct and; that means, retired means whenever it has been proved to be correct the prediction has turned out to be correct. Then only that history is stored, that branch history it is stored in branch history table and branch target buffer, and it also uses a return address stack RAS, that uses 16 entries and it predicts return addresses for proceed your calls.

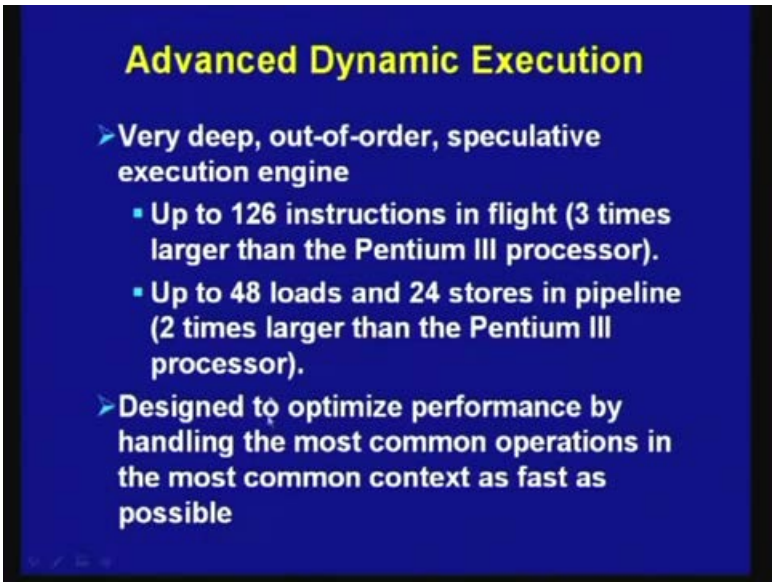
So, there are many proceed your calls and there return addresses can also stored in a return address stack, and it allows branches and their targets to coexist in a single cache line. And it increases parallelism since decode bandwidth is not wasted, so all these things are used to enhance the performance of the processor.

(Refer Slide Time: 30:53)



And this is the case when branch hints occurs, Pentium 4 permits software to provide hints to the branch prediction, and trace information hardware to enhance performance, and it takes the forms of prefixes to conditional branch instructions. So, it is used only at trace build time, and have no effect on already built traces, so this is how the I mean the software to provide the hint is used for branch prediction, and trace formation hardware, so this also goes in enhancing the performance.

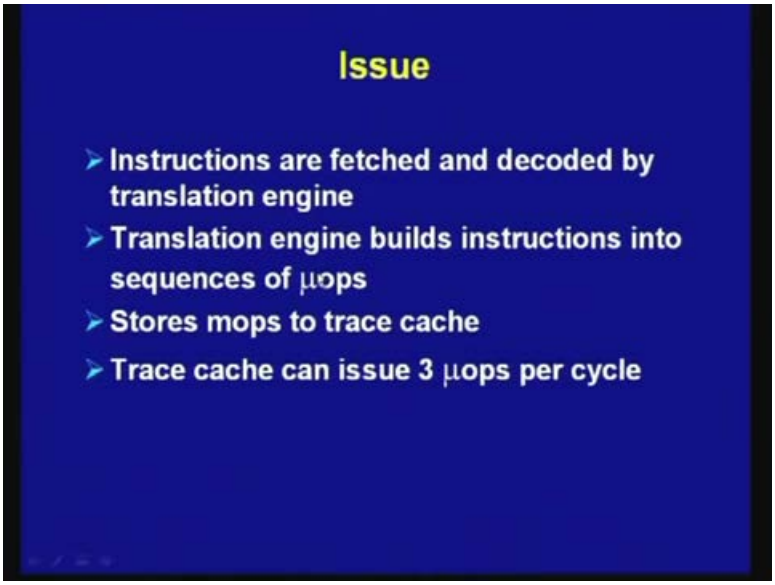
(Refer Slide Time: 31:38)



Then coming to the execution it uses advanced dynamic execution, so it is very deep out of order speculative execution engine, so we have already seen it has got 20 stage pipelines. So, it is very deep, it is considered to be very deep, and it also allows you out of order execution in a speculative way, so speculative execution engine is used, and up to 126 instructions can be in flight, which is 3 times larger than the Pentium 3 processor.

So, instructions can be in flight, and then that speculative execution engine will select dynamic schedule them for the purpose of execution, and up to 48 loads, and 24 stores in pipeline can be present, which is also 2 times larger than Pentium 3 processor. And these are designed to optimize performance by handling the most common operations in the most common context as fast as possible, so this is the basic objective of this speculative execution engine.

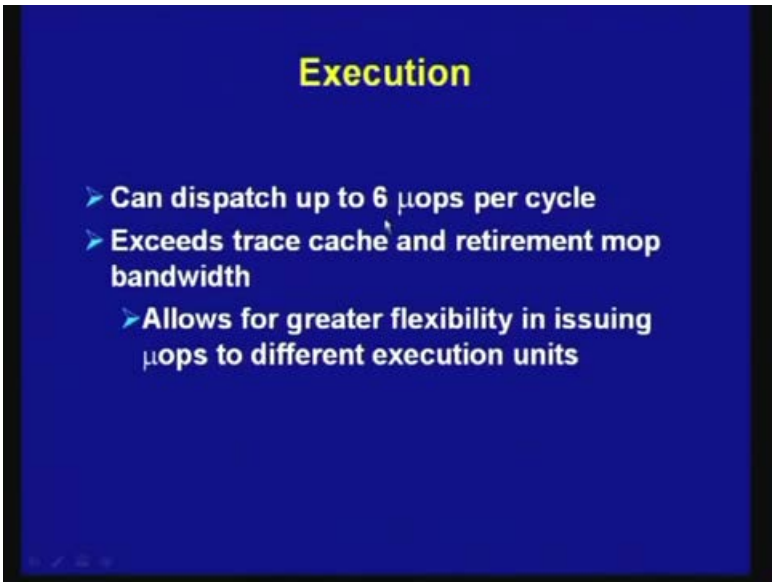
(Refer Slide Time: 33:01)



Then issue instructions are fetched and decoded by translation engine, and translation engine builds instructions into sequences of micro operations, and it stores micro operations to trace cache, as we have already mentioned. Trace cache can issue 3 micro operations per cycle three micro operations can issue per cycle in the issue stage, and execution engine can dispatch up to 6 micro operations per cycle. And it exceeds trace cache and retirement micro operation bandwidth, so it is received 3 micro operations per cycle and it can dispatch up to six micro operations per cycle. So, it allows for greater

flexibility in issuing micro operations to different execution units, we have already seen there are 7 different execution units present in the processor.

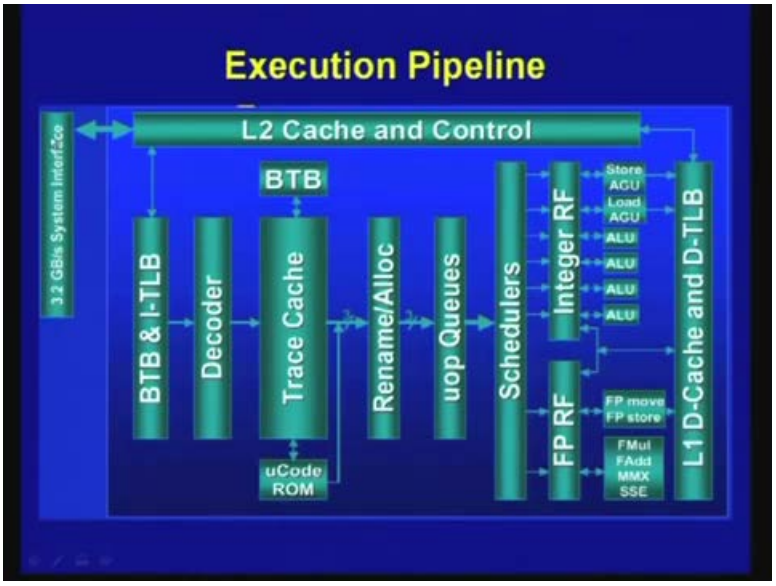
(Refer Slide Time: 34:05)



And this is the execution pipeline you can see here that system interface 3.2 gigabytes per second, system interface which is for external interface, and here is your L 2 cache from the memory it is stored in that first transfer to the L 2 cache. So, L 2 cache and control is there, then it has got branch buffer and L 1 cache, instruction TLB, and then decoder trace cache. It uses renaming of registers and instead of using ROB, and then here is your micro operation queues, then here is the schedulers which schedules those micro operations at 2 different functional units.

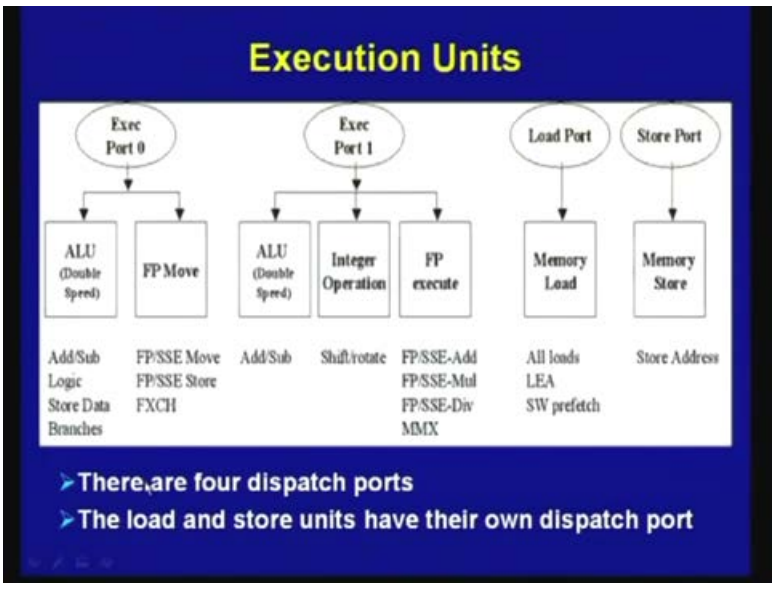
So, these are the integer functional units, and this is floating point functional units, so these are the different functional units, so this is integer register file, and this is a floating point register file. And this are the different functional units that is present, storage AGU, load AGU, ALU 1, ALU 2, ALU 3, ALU 4. So, and it has got floating point move, floating point arithmetic operations multiplication, add MMX SSE. So, you can see it has got variety functional units; that is being present, and it directly accesses the L 1 data cache and the data TLB, so this is the execution pipeline.

(Refer Slide Time: 36:05)



And these are the 20 stage pipeline that is being shown here, I am not been into the details of the these different stages cache trace, next instruction and so on, and these are the execution units.

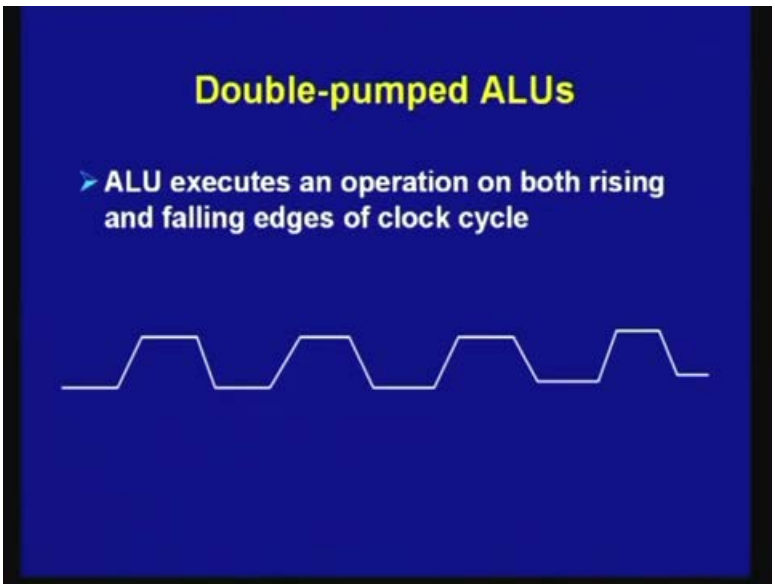
(Refer Slide Time 36:23)



So, there are 4 dispatch ports, you can see 1,2,3,4, and the load and store units have their own dispatch ports, the port 0 and port 1, they are feeding to ALU floating point move a double speed. Double speed means we shall see that double speed is achieve by perfuming the execution on both edges, and integer operation and floating point execute.

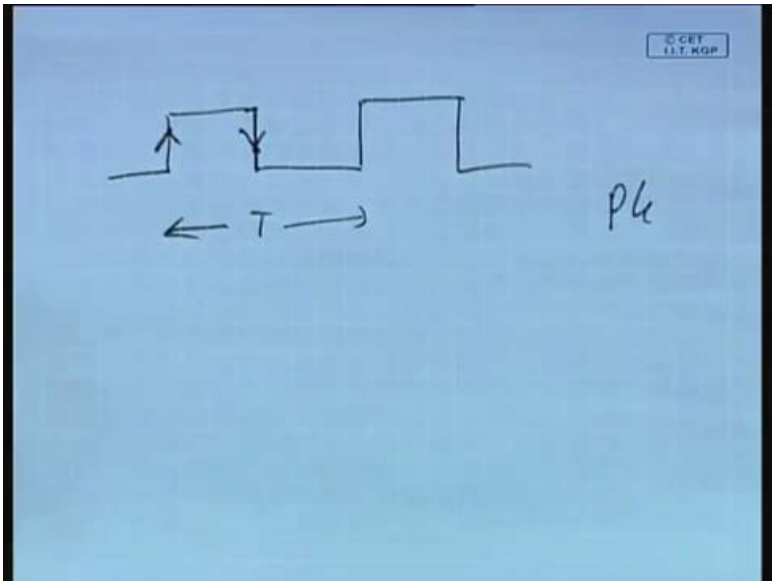
So, different operations, that is being performed by the different functional units are listed below, so addition that ALU double speed performs additions, subtraction, logical operations, store data branches. Then this floating point move performs, floating point SSE, floating point SSE stores, and floating point exchange, this ALU double speed performs additions, subtraction. And this integer operations performs shift and router operations, and this floating point execute per floating point SSE add, floating point SSE multiplied, floating point SSE division, and MMX operations, so this memory load performs all load, and that memory store performs the store operations.

(Refer Slide Time: 37:57)



And as I mentioning it uses double pumped ALU's, what is the significance of double pumped ALU's, all ALU's executes an operation on both rising and falling edges of clock cycle, normally you know we use a single clock cycle.

(Refer Slide Time: 38:18)



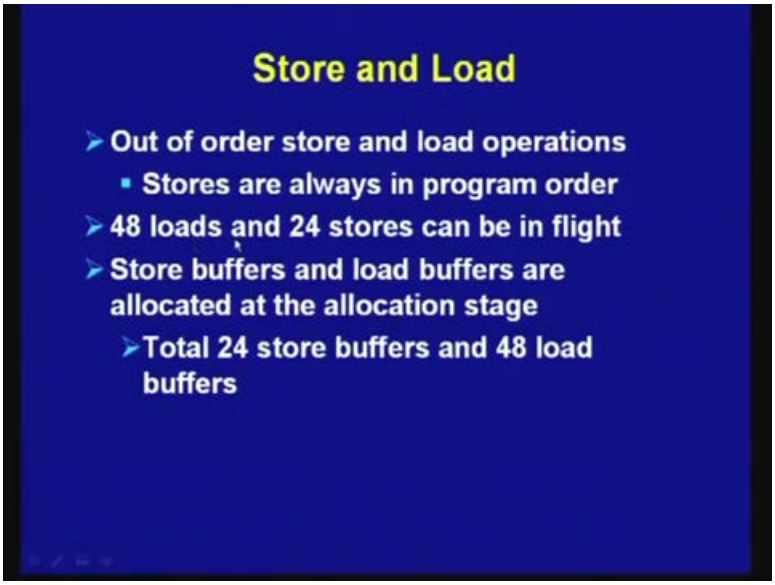
So, this is considered to be a single clock cycle, these are time period, and normally in 1 clock cycle, 1 operation is performed, but in case of Pentium 4 that execution is done, on this edge as well on this edge. As, a result the ((Refer Time: 38:43)) it is called double pumped ALU's and ALU executes an operation on both rising and falling edges of the clock cycles.

(Refer Slide Time: 38:52)

Then the retirement means at the end of execution, when the micro operations completes their operation and register as are updated and so on, it can retire 8 micro operations per

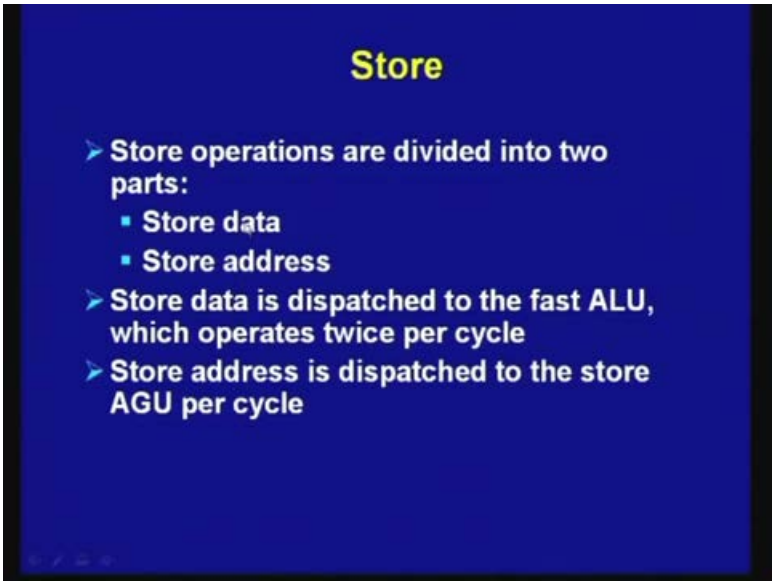
cycle. it supports precise exceptions reorder buffer to organize completed micro operations, since it uses out of order execution, it is necessary to have re order buffer, and which is done to with to organise the completed micro operations. So, that the way the results are stored follow the same order, and the same program order, so these are also keeps track of branches and send updated branch information to the branch target buffer, so this is how the retirement stage works.

(Refer Slide Time: 39:53)



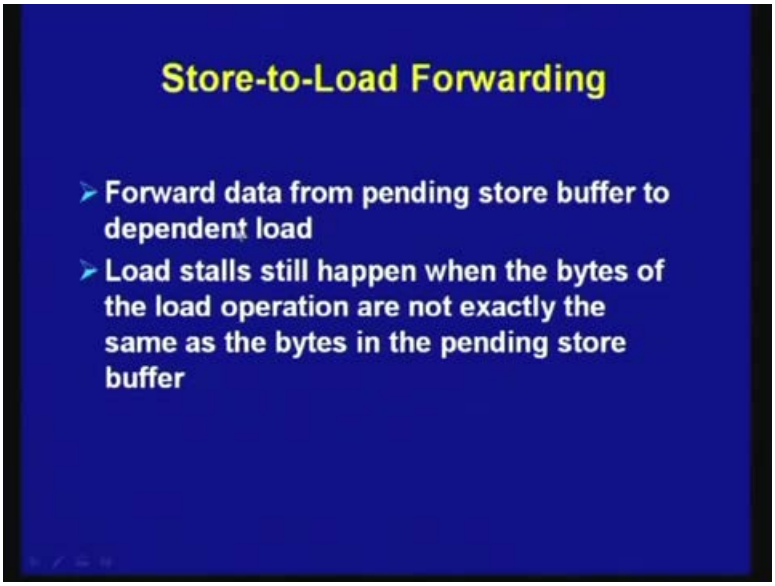
Then the store and load, store and load is also out of order store and load operations stores are always in program order, and 48 loads and 24 stores can be in flight, and store buffers and load buffers are allocated at the allocation stage. So, and total 24 store buffers and 48 load buffers are used for the purpose of storing the operands, and then their loading and storing is performed, and not in not coupled with execution of instruction, but they can go head independently.

(Refer Slide Time: 40:40)



So, store operations are divided into 2, parts store data and store address, store data is dispatched to the fast ALU and which operates twice per cycle, and store address is dispatched to the store AGU per cycle. So, this is how store operations is I mean is operation is performed for better performance.

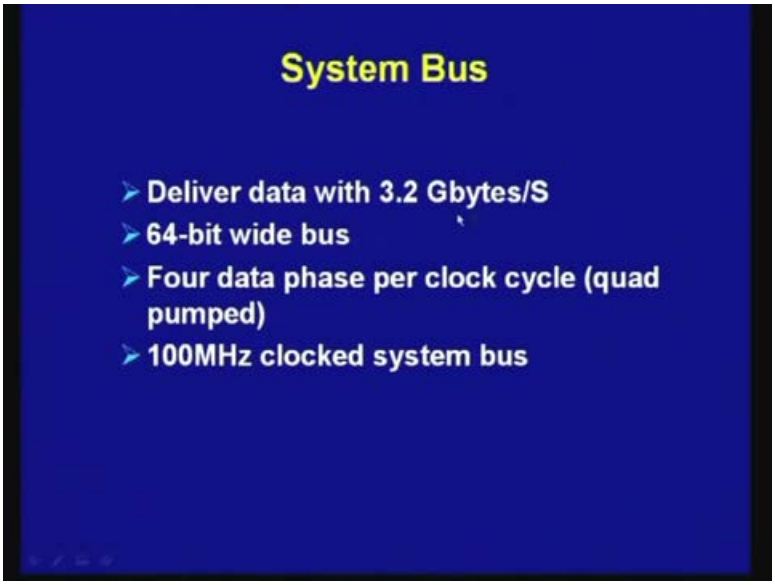
(Refer Slide Time: 41:08)



And it also does store to load forwarding ,forward data from pending store buffer to dependent load, so this forwarding helps you to I mean to the dependent loads, so forward data from the pending store. So, some buffer to the dependent load, so this is

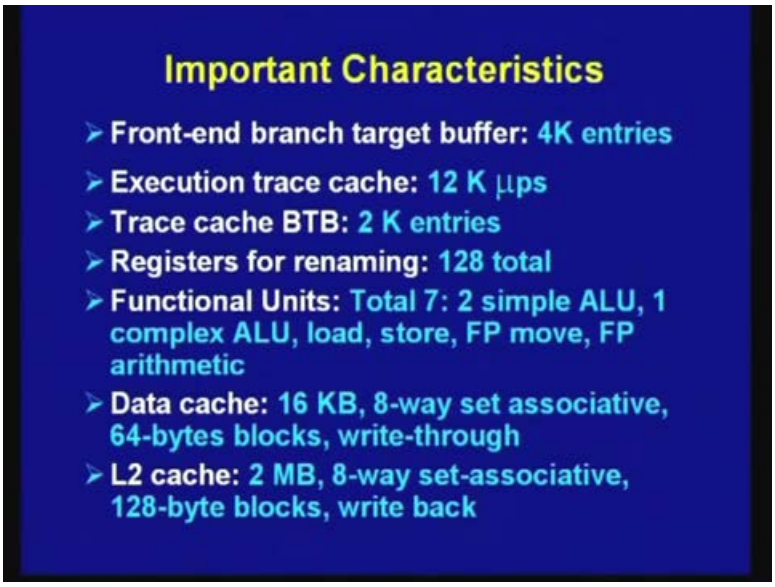
how forwarding helps you also to enhance performance, so load stalls still happen when the bytes of the load operation are not exactly, the same as the bytes in the pending store buffer, so in spite of the stalls this forwarding can help to great extent.

(Refer Slide Time: 42:04)



Coming to the last portion that is your system bus, it delivers data with 3.2 gigabytes per second, that is the system bus for external interface, and it is provides 64 bit wide bus as you are seen, and four data phases per cycle, quad pumped and 100 megahertz clock system bus, so this is a system bus.

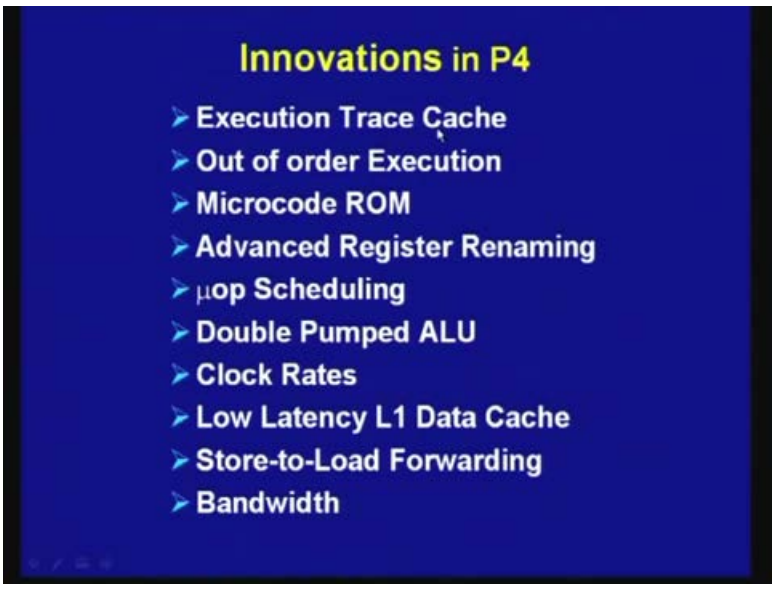
(Refer Slide Time: 42:25)



Now, coming to the important characteristics of that we have discussed, so for Pentium 4 as I have already mentioned, it uses front end branch target buffer, so that front end branch has got 4 k entries. And that execution trace cache has got 12 k micro operations, and I have already mention about it, and trace cache branch target buffer has got 2 k entries. So, you can see that it has got two different branch target buffers, 1 is front end branch target buffer, another is trace cache branch target buffer, so and then there are 128 registers are available total for the purpose of renaming.

So, it uses renaming and 128 registers are available for this purpose, and as I mentioned it has got total of 7 functional units, 2 simple ALU, 1 complex ALU. Load, store, floating point move, floating point arithmetic, as we have shown in one of our earlier diagram, then data cache is 16 kilo byte 8 way set associative 64 bytes blocks and it uses write through policy. Then L 2 cache is 2 megabyte, and for 8 way set associative 128 byte blocks, and it uses write back policy.

(Refer Slide Time: 44:00)



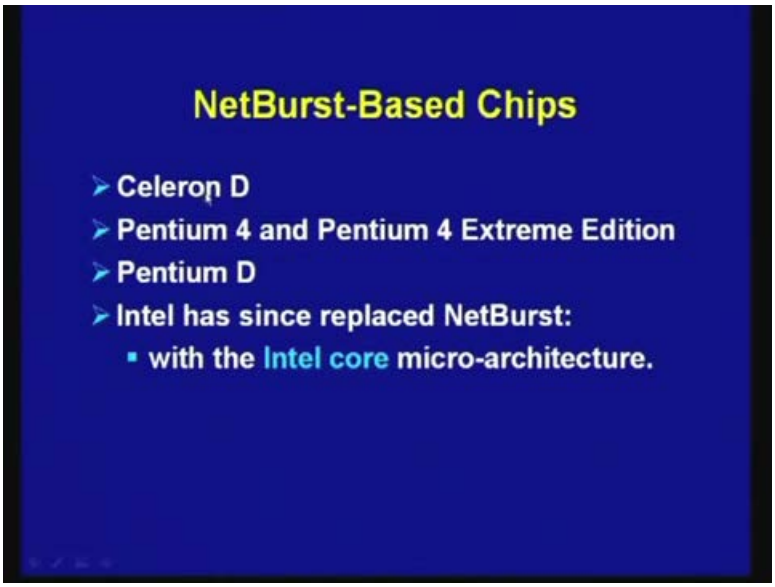
So, here is the list of different innovations, that has been used in Pentium 4, number 1 is the use of execution trace cache, we have already discussed at length the execution trace cache, then the use of out of order execution. So, out of order execution as I mean as soon as your operands are available execution take place, so it does out of order execution and which all of course, requires that retirement units.

So, that the end the register updating and everything take place in order, then microcode ROM is used here, so that the micro operations are stored in micro code ROM. And from the micro code ROM, they can be directly fetched and can be sent to execution units, and then it uses advance register renaming, and with the help of 128 registers that I have already mentioned that the registers use for renaming.

So, it uses advanced register renaming and it does micro operations scheduling rather than instruction scheduling, and it uses double pumped ALU which is also enhances the performance of the processor. Then clock rates it uses higher clock rates, because of the 20 stage pipeline, you have seen that pipeline is of 20 stages, so the clock frequency can be made higher, we know that as the number of pipelines stages is increased the clock frequency can be made higher.

So, it uses higher clock rate, and I have already mentioned about the low latency L 1 data cache by making it simple and small, and then it uses store to load forwarding, we have discussed in detail. Then the bandwidth, it gives you higher bandwidth, we have seen that L 2 to L 1 on cache bandwidth is 128 bits, so it gives you higher bandwidth and external bandwidth is 64 bit, so with these gives you higher bandwidth so you can see these are the various innovations used in Pentium 4.

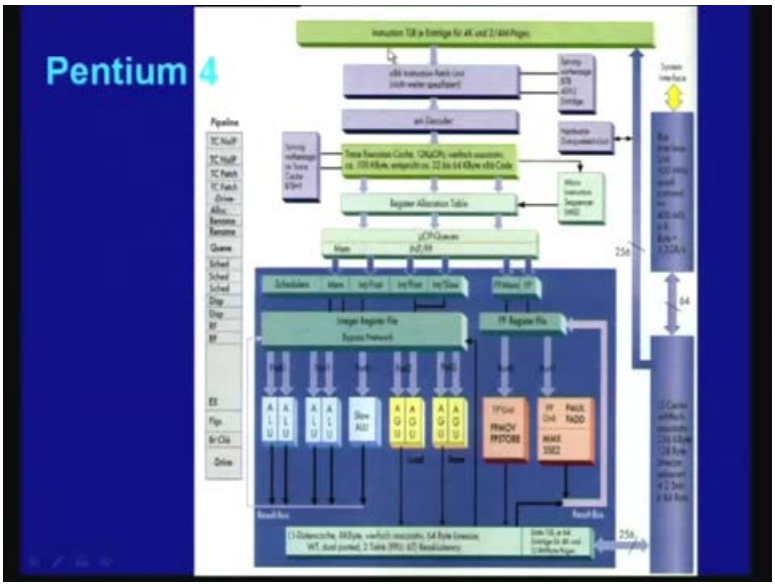
(Refer Slide Time: 46:32)



And based on this Netburst architecture, number of processors are available that Celeron D, Pentium 4, and Pentium 4 extreme edition, Pentium D. Of course, Intel has, since

replaced the Netburst, with the Intel core, micro architecture and latter on, we shall discuss about it this.

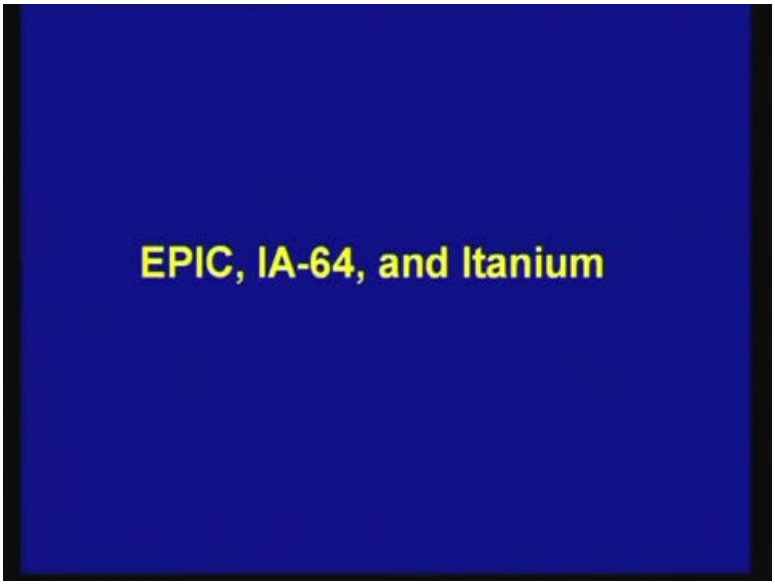
(Refer Slide Time: 46:57)



And here is the detailed micro architecture of Pentium 4, you can see this is that, here is your system interface, that bus interface unit, and here the interfaces of the 64 bytes, and this is that L 2 cache, 256 kilobytes. And the bus between L 1 cache, TLB, and L 1 cache is 256 bits in this particular diagram, and this is that instruction fetch unit, this is the TLB, and there is trace cache.

That I mention 128 but 12 k micro operations as stored in this trace cache, is that register cache allocation table, and is the micro operations queues, and these are memory and instructions, that those operations are stored. And here this scheduler, schedules the different operations, which are to be executed by those functional units, so these are the different ALU's, AGU's that floating point unit, floating point move, and floating point execution units, so and that this 20 stage pipeline is shown on the left side, so this gives you an overview of the Pentium 4 micro architecture.

(Refer Slide Time: 48:53)



So, we have come to the end of today's lecture, now we have discussed in detail the Pentium 4 micro architecture, in my next lecture I shall focus on epic or IA-64 that uses the Itanium processor by Intel, that uses 64 bit processor and that I shall discuss in my next lecture.

Thank you.