CASE STUDY: CACHE STRUCTURE IN INTEL PENTIUM 1993

Meet Doshi[1] 1812078
[1]Department of Electronics Engineering
[1] K.J. Somaiya College of Engineering, Vidyavihar, 400077, Maharashtra, India

CONTENTS

I Introduction 1
I-A **Today's Pentium** . . . 1
I-B **Original P5 Micro-architecure Pentium** . . . 1

II L1 CACHE 1
II-A **CACHE BASICS** . . . 1
II-B **CACHE ORGANIZATION** . . . 1
II-C **CACHE STRUCTURE** . . . 2
II-C1 **Cache Operating Modes** . . . 2
II-C2 **Cache Flushing** . . . 2
II-C3 **Data Cache Consistency Protocol (MESI Protocol)** . . . 2
II-C4 **Cache Mapping** . . . 3

III CONCLUSION 3

References 3
[1]

[1]
Date: September 30, 2020
Corresponding authors:
Meet Doshi(email: meet10@somaiya.edu)

CASE STUDY: CACHE STRUCTURE IN INTEL PENTIUM 1993

Meet Doshi[1] 1812078
[1]Department of Electronics Engineering
[1] K.J. Somaiya College of Engineering, Vidyavihar, 400077, Maharashtra, India

I. Introduction

A. *Today's Pentium*

Pentium is a brand used for a series of x86 architecture-compatible microprocessors produced by Intel since 1993. In their form as of November 2011, Pentium processors are considered entry-level products that Intel rates as "two stars",meaning that they are above the low-end Atom and Celeron series, but below the faster Intel Core lineup, and workstation Xeon[1].

B. *Original P5 Micro-architecure Pentium*

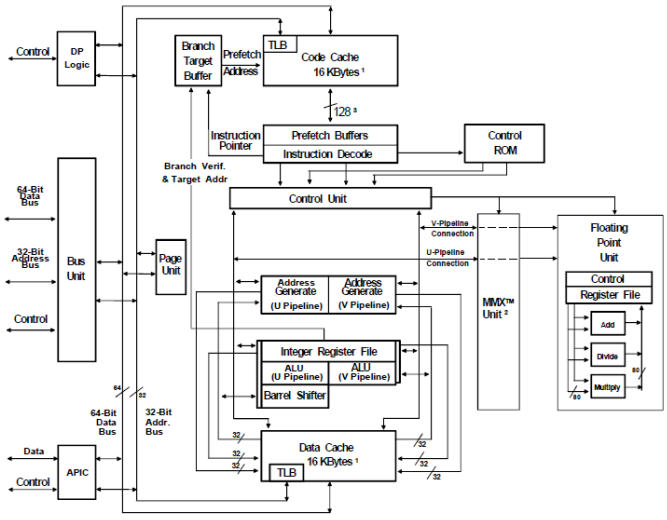
The Pentium processor is the successor to the Intel486 processor. Originally released with a 66MHz clock speed, it is a 16-bit based superscalar processor capable of executing two instructions in parallel during a single clock. It uses a CISC (Complex Instruction Set Computer) type instruction set, and uses the little-endian type format to store bytes in memory. Since all x86 architectures are upward compatible from x86 upto Pentium IV, P5 architecture is also based on the original 80386 IA-32 architecture. It was called the first superscalar architecture since it had 2 whole pipes called the U-Pipe and the V-Pipe. It had a 32 bit ALU and two such ALU's. But there was only one Floating Point unit of Extended double precision IEEE format. One of the best features of Pentium I was introduction to simple but very reliable hit rate branch prediction technique. The address bus was 32 bits, meaning it could address byte addressable memory of upto 4 GB. The data bus can transmit 64 bits(This doubling of the data bus width meant that twice the amount of information previous chips could manage was read with each memory fetch)[3]. Pentium also incorporates a 5 stage pipeline for lower CPI.
The general terms "Pentium processor" and "Pentium processor family" are used throughout this volume to refer to both the Pentium processor (75/90/100/120/133/150/166/200)MHz.

II. L1 CACHE

A. *CACHE BASICS*

The Pentium processor includes separate code and data caches integrated on chip to meet its performance goals. Each cache on the Pentium processor with MMX technology is 16 Kbytes in size, and is 4-way set associative. The caches on the Pentium processor (75/90/100/120/133/150/166/200) are each 8 Kbytes in size and 2-way set-associative. Each cache has a dedicated Translation Lookaside Buffer (TLB) to translate linear addresses to physical addresses. The Pentium processor data cache is configurable to be writeback or writethrough on a line-by-line basis and follows the MESI protocol. The data cache tags are triple ported to support two data transfers and an inquire cycle in the same clock. The code cache is an inherently write protected cache. The code cache tags of the Pentium processor (75/90/100/120/133/150/166/200) are also triple ported to support snooping and split-line accesses. The Pentium processor with MMX technology does not support split line accesses to the code cache. As such, its code cache tags are dual ported. Individual pages can be configured as cacheable or non-cacheable by software or hardware. The caches can be enabled or disabled by software or hardware[2].

Figure 1 Pentium I Architecture[2]



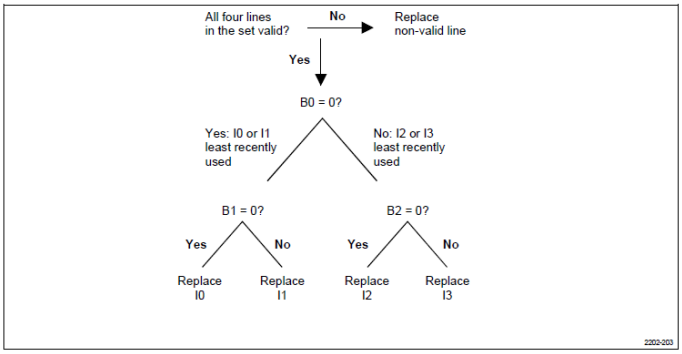
If you look at the above Figure you can clearly see how the superscalar architecture has two different cache with each having their own TLB(look-aside) to avoid translation time of linear address to physical address.

B. *CACHE ORGANIZATION*

The caches have been designed for maximum flexibility and performance. The data cache is configurable as writeback or writethrough on a line-by-line basis. Memory areas can be defined as non-cacheable by software and external hardware. Cache writeback and invalidations can be initiated by hardware

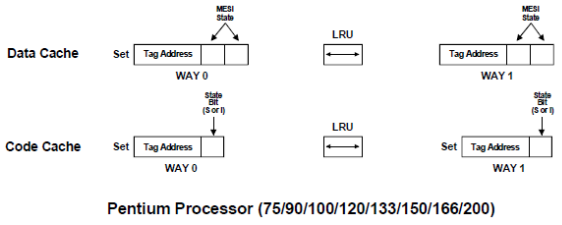
or software. Protocols for cache consistency and line replacement are implemented in hardware, easing system design. On the Pentium processor (75/90/100/120/133/150/166/200), each of the caches are 8 Kbytes in size and each is organized as a 2-way set associative cache. There are 128 blocks in each cache, 64 blocks in one set, each block containing 2 lines (each line has its own tag address). Each cache line is 32 bytes wide. The Pentium processor with MMX technology has two 16 Kbyte 4-way setassociative caches the with a cache line length of 32 bytes. In the Pentium processor (75/90/100/120/133/150/166/200), replacement in both the data and instruction caches is handled by the LRU mechanism which requires one bit per set in each of the caches. The Pentium processor with MMX technology employs a pseudo-LRU replacement algorithm which requires three blts per set in each of the caches. When a line must be replaced, the cache will first select which of I0:I1 and I2:I3 was least recently used. Then the cache will determine which of the two lines was least recently used and mark it for replacement. This decision tree is shown in Figure 2

Figure 2 Pseudo-LRU Cache Replacement Strategy[2]



The data cache consists of eight banks interleaved on 4-byte boundaries. The data cache can be accessed simultaneously from both pipes, as long as the references are to different cache banks. A conceptual diagram of the organization of the data and code caches is shown in Figure 3. Note that the data cache supports the MESI writeback cache consistency protocol which requires 2 state bits, while the code cache supports the S and I state only and therefore requires only one state bit.

Figure 3 Conceptual Organization of Code and Data Caches[2]



C. CACHE STRUCTURE

The instruction and data caches can be accessed simultaneously. The instruction cache can provide up to 32 bytes of raw opcodes and the data cache can provide data for two data references all in the same clock. This capability is implemented partially through the tag structure. The tags in the data cache are triple ported. One of the ports is dedicated to snooping while the other two are used to lookup two independent addresses corresponding to data references from each of the pipelines. The instruction cache tags of the Pentium processor (75/90/100/120/133/150/166/200) are also triple ported. Again, one port is dedicated to support snooping and the other two ports facilitate split line accesses (simultaneously accessing upper half of one line and lower half of the next line). The storage array in the data cache is single ported but interleaved on 4-byte boundaries to be able to provide data for two simultaneous accesses to the same cache line.Each of the caches are parity protected. In the instruction cache, there are parity bits on a quarter line basis and there is one parity bit for each tag. The data cache contains one parity bit for each tag and a parity bit per byte of data. Each of the caches are accessed with physical addresses and each cache has its own TLB (translation lookaside buffer) to translate linear addresses to physical addresses.

1) *Cache Operating Modes*

The operating modes of the caches are controlled by the CD (cache disable) and NW (not writethrough) bits in CR0. For normal operation and highest performance, these bits should both be reset to “0.”

Table 1 Modes of Operation

CD	NW	Description
1	1	Read hits access the cache,Write misses access memory,etc
1	0	Read hits access the cache, Write hits update the cache, etc
0	1	General protection Exception
0	0	Write hits update the cache Only writes to shared lines and write misses appear externally

2) *Cache Flushing*

The on-chip cache can be flushed by external hardware or by software instructions. Flushing the cache through hardware is accomplished by driving the FLUSH# pin low. This causes the cache to writeback all modified lines in the data cache and mark the state bits for both caches invalid. The Flush Acknowledge special cycle is driven by the Pentium processor when all writebacks and invalidations are complete.

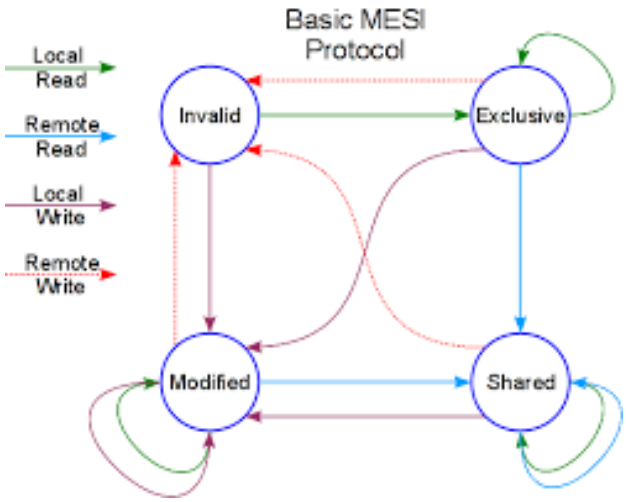
3) *Data Cache Consistency Protocol (MESI Protocol)*

The Pentium processor Cache Consistency Protocol is a set of rules by which states are assigned to cached entries (lines). The rules apply for memory read/write cycles only. I/O and special cycles are not run through the data cache. Every line in the Pentium processor data cache is assigned a state dependent on both Pentium processor generated activities and activities generated by other bus masters (snooping). The Pentium processor Data Cache Protocol consists of four states that define whether a line is valid (HIT/MISS), if it is available in other caches, and if it has been MODIFIED. The four states are the M (Modified), E (Exclusive), S (Shared) and the I (Invalid) states and the protocol is referred to as the MESI protocol. A definition of the states is given below:

Table 3 MESI Protocol

State	Description
M - Modified:	An M-state line is available in ONLY one cache and it is also MODIFIED (different from main memory). An M-state line can be accessed (read/written to) without sending a cycle out on the bus.
E - Exclusive:	An E-state line is also available in ONLY one cache in the system, but the line is not MODIFIED (i.e., it is the same as main memory). An E-state line can be accessed (read/written to) without generating a bus cycle. A write to an E-state line will cause the line to become MODIFIED.
S - Shared:	This state indicates that the line is potentially shared with other caches (i.e. the same line may exist in more than one cache). A read to an S-state line will not generate bus activity, but a write to a SHARED line will generate a write through cycle on the bus. The write through cycle may invalidate this line in other caches. A write to an S-state line will update the cache.
I - Invalid:	This state indicates that the line is not available in the cache. A read to this line will be a MISS and may cause the Pentium processor to execute a LINE FILL (fetch the whole line into the cache from main memory). A write to an INVALID line will cause the Pentium processor to execute a write-through cycle on the bus.

Figure 3 MESI Protocol



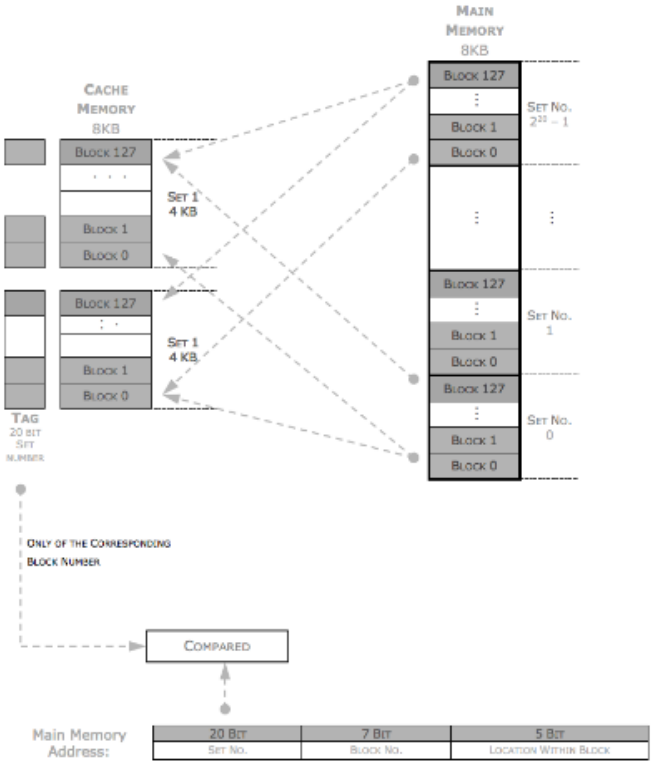
4) Cache Mapping

Table 2 Cache Mapping Sizes

Size of Main Memory	32 bits 4 GB
Size of Cache Memory	8 Kb Code Cache 8 kb Data Cache
Mapping Type	2- Way Set Associative
Number of Sets in Cache Memory	2
Number of Sets in Main Memory	$2\backslash^{20}$
No. of Blocks in one set in Cache and Main Memory	64
Total No. of Blocks in Cache	128
Number of Cache Lines in one block	2
Size of Cache Line	32 bytes
Size of Block	64 bytes
Size of Set	4 Kbytes
Size of Cache	8 Kbytes

The above table shows size of each component in cache mapping.

Figure 4 Two Way Set Associative Mapping



III. CONCLUSION

In this case study we took a close look at how the built in L1 cache of Pentium 1 processor. We also looked at the P5 Micro architecture developed by Intel which was the first step towards modern day superscalar processors. The cache implementation is similar to those of MIPS architecture developed by John L Hennessy. The Modified Exclusive Shared Invalid(MESI) Protocol maintains consistency of the cache and provides flexibility to it. We also looked at how memory mapping is done on both code and data caches and why they are separately provided with each having separate TLB of it's own. In a whole, the Pentium was a big breakthrough for the modern day processor line up.

REFERENCES

[1] https://en.wikipedia.org/wiki/Pentium#Pentium
[2] Pentium® Processor Family Developer's Manual 1997
[3] https://en.wikipedia.org/wiki/P5_(microarchitecture)
[4] https://en.wikichip.org/wiki/intel/pentium
[5] https://encyclopedia2.thefreedictionary.com/Pentium+1
[6] Pentium® Processor Software Developer's Manual 1997
[7] https://en.wikipedia.org/wiki/MESI_protocol