TOP P6 QUESTIONS AND ANSWERS

Will Pentium® Processor optimizations hold for P6?

In general terms, Pentium Processor optimizations hold for the P6. Intel does not necessarily recommend that ISVs tune existing assembly code specifically for P6. In most cases, the recommendation is to recompile optimized areas with a modern 32-bit compiler. If ISVs want ABSOLUTE performance, however, they need to hand tune for all supported processors including the P6 procesor. (See the 32-bit Optimization Guide.)

16-Bit Applications on P6:

16-bit code is implemented via larger opcode sizes due to instruction prefixes and thus has less instructions in cache each line. This is the same on Intel486™ and Pentium® Processors. Segment register loads change the processor environment and speculative data already loaded via dynamic execution may become invalid. The higher number of seg reg loads, the more performance degradation on P6 over and above an application without segment register loads.

What are the memory alignment requirements for P6?

The P6 processor follows the alignment rules of the i486™ processor; however, the impact of not aligning code on P6 is not nearly as severe on the P6 because there is a high probability that dynamic execution will have provided other instructions to perform. Misaligned cache data accesses cost significantly more cycles on the P6.

Number of execution units:

Essentially 6 user-visible execution units - 2 fp, 2 integer, a load and a store.

Compilers/Tools:

Intel's Proton compiler already has P6 optimizations (removed partial stalls). We are currently engaging with other compiler vendors. (See the section on 'Compiler Information'.)

What is 36-Bit Addressing?

This is achieved by enabling CR4.PAE=1 (Page Addr Ext.). Page Translation occurs by Bits 30&31 pointing to an entry in a page dir ptr table (a new structure pointed to by CR3). Page directory pointer table points to page directory and bits 21-29 point to entry in page directory. Page directory entry points to beginning of page table with bits 12-20 pointing to entry in that table. That entry points to page frame with bits 0-11 pointing to offset in page. Page sizes are 2M when using 32-bit addresses. (Request OS Writer's Guide if more information is required.)

What features have been added to P6 to improve Paging?

P6 has two features designed to improve Paging performance.

1. 4Mbyte code pages
2. The addition of a global page bit in the PTE that prevents the page entry in the TLB from being flushed.

What are the new instructions/enhanced instructions implemented in P6?

CMOVcc (Conditional Move) - allows graphic driver writers to implement pixel saturation algorithms and reduce amount of branches in routine. Tests integer flags and if a condition is true, move a register to a register or memory
example:

```
    ; Code that uses CMP/JMP          ; Code that uses CMOV
    cmp ECX,EAX                        cmp ECX,EAX
    ja above                           cmovbe EAX,ECX
    mov EAX,ECX
above:
```

FCMOV - Floating point conditional move controlled by integer condition codes
FCOMI, FUCOMI, FCOMIP, FUCOMIP - Compare stack top to the source and sets integer flags accordingly. Convenient for testing any of the 6 arithmetic relations as well as ordered or unordered after an FCOM or FTST operation.
example:

```
    ; on I387                         ; Code that uses FCOMI
    FCOM ST0, ST1                      FCOMI ST0, ST1
    FNSTSW AX                          Jcc label
    SAHF                               ...
    Jcc label                          label:
    ...
label:
```

String instructions - speeded up by factor of 3x
New memory type - Uncacheable, Speculatable, Write Combining (USWC) that allows write oriented frame buffers to be treated differently, i.e., writes buffered in a 32 byte line buffer and only write the data in a burst mode to the frame bufffer when necessary or full instead of doing inefficient byte oriented writes.
CPUID - Contains bits to determine P6 features (documented in OS Writer's Guide)
RDPMC - Read performance monitoring counters

(For further information, see the 32-Bit Optimizations Guide.)

Bus speed: The initial implementations will have a 66MHz system bus speed, with the P6 running at 133Mhz.

Wattage and voltage: Peak ~20 watts @ 2.9V