Intel 80486 CPU Enhancements (1989)
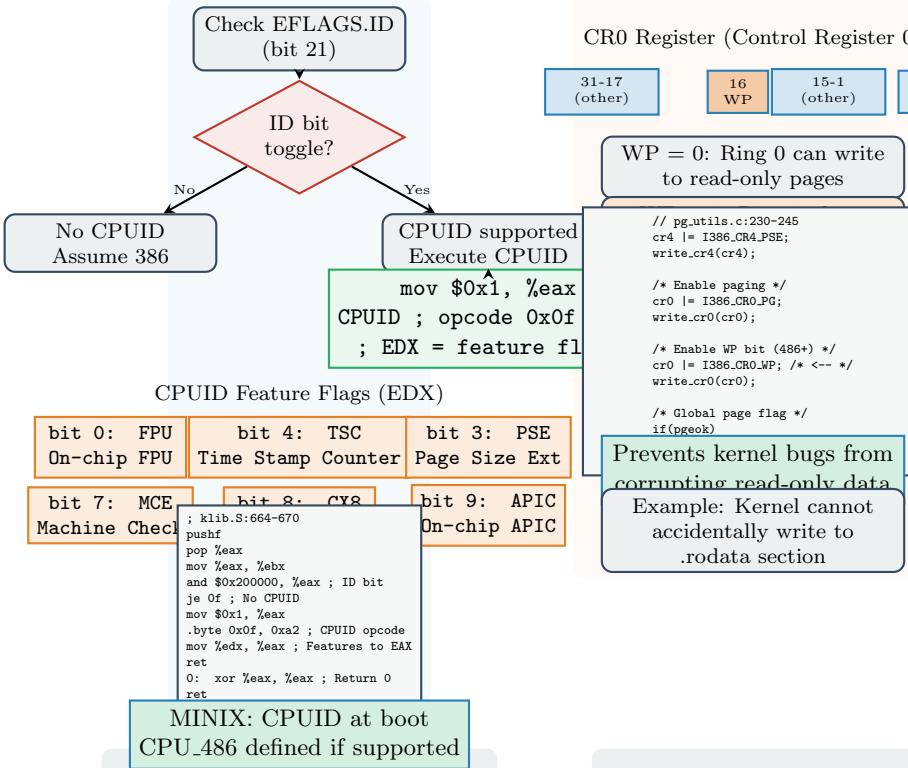
New Features Beyond 386: CPUID, WP Bit, Cache, Atomic Instructions

Source files:
specialreg.h:122-157 (CPUID flags), klib.S:664-670 (CPUID impl),
pg_utils.c:237 (WP bit), cpufunc.h:74 (WBINVD), atomic.S:67-142 (CMPXCHG/XADD)

1. CPUID Instruction

First CPU feature detection mechanism

Check EFLAGS.ID
(bit 21)

ID bit
toggle?

No → No CPUID
Assume 386

Yes → CPUID supported
Execute CPUID

```
mov $0x1, %eax
CPUID ; opcode 0x0f
; EDX = feature fl
```

CPUID Feature Flags (EDX)

| bit 0: FPU | bit 4: TSC | bit 3: PSE |
| On-chip FPU | Time Stamp Counter | Page Size Ext |

| bit 7: MCE | bit 8: CX8 | bit 9: APIC |
| Machine Check | | On-chip APIC |

```
; klib.S:664-670
pushf
pop %eax
mov %eax, %ebx
and $0x200000, %eax ; ID bit
je 0f ; No CPUID
mov $0x1, %eax
.byte 0x0f, 0xa2 ; CPUID opcode
mov %edx, %eax ; Features to EAX
ret
0:  xor %eax, %eax ; Return 0
ret
```

MINIX: CPUID at boot
CPU_486 defined if supported

2. CR0.WP Bit (Write Protect)

Enforces read-only pages in Ring 0

CR0 Register (Control Register 0)

| 31-17 (other) | 16 WP | 15-1 (other) | 0 PE |

WP = 0: Ring 0 can write to read-only pages

```
// pg_utils.c:230-245
cr4 |= I386_CR4_PSE;
write_cr4(cr4);

/* Enable paging */
cr0 |= I386_CR0_PG;
write_cr0(cr0);

/* Enable WP bit (486+) */
cr0 |= I386_CR0_WP; /* <-- */
write_cr0(cr0);

/* Global page flag */
if(pgeok)
```

Prevents kernel bugs from corrupting read-only data

Example: Kernel cannot accidentally write to .rodata section

3. Cache Control

8 KB on-die L1 cache (unified)

WBINVD
Write-back and invalidate cache

INVLPG <vaddr>
Invalidate TLB entry

WBINVD:
1. Write dirty lines to RAM
2. Invalidate all cache

INVLPG:
Invalidate single TLB entry

```
// cpufunc.h:74
void wbinvd(void);

// cpufunc.h:55
void invlpg(vaddr_t);

// Usage:
wbinvd(); // Full flush
invlpg(0xC0000000); // One page
```

MINIX: INVLPG preferred over CR3 reload for single-page invalidation

4. New Atomic Instructions

CMPXCHG <dest>, <src>
Compare and exchange

If EAX == dest:
dest := src, ZF := 1

```
; atomic.S:80-93
_atomic_and_32_nv:
  movl 4(%esp), %edx
  movl (%edx), %eax
0:
  movl %eax, %ecx
  andl 8(%esp), %ecx
  lock
  cmpxchgl %ecx, (%edx)
  jnz 1f
  movl %ecx, %eax
  ret
1:  jmp 0b
```

5. XADD Instruction

XADD <dest>, <src>
Exchange and add
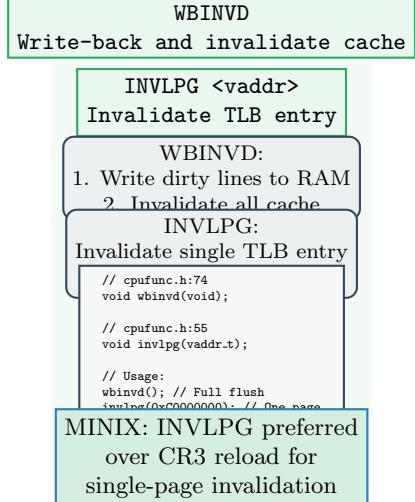
temp := dest
dest := dest + src

```
; atomic.S:62-70
_atomic_add_32_nv:
  movl 4(%esp), %edx
  movl 8(%esp), %eax
  movl %eax, %ecx
  lock
  xaddl %eax, (%edx)
  addl %ecx, %eax
  ret
```

6. BSWAP Instruction

BSWAP <reg32>
Byte swap (endianness)

0x12345678
becomes

```
// byte_swap.h:42-49
static __inline uint32_t
__byte_swap_u32_variable(
  uint32_t x)
{
  __asm volatile (
    "bswap %1"
    : "=r" (x)
    : "0" (x));
  return (x);
}
```

486 Summary: First CPU with runtime feature detection (CPUID),
8KB on-die cache, WP bit for kernel memory protection, atomic instructions (CMPXCHG, XADD),
and cache control (WBINVD, INVLPG). MINIX fully supports all features.