To return from an exception- or interrupt-handler procedure, the handler must use the IRET (or IRETD) instruction. The IRET instruction is similar to the RET instruction except that it restores the saved flags into the EFLAGS register. The IOPL field of the EFLAGS register is restored only if the CPL is 0. The IF flag is changed only if the CPL is less than or equal to the IOPL. See Chapter 3, "Instruction Set Reference, A-M," of the Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 2A, for a description of the complete operation performed by the IRET instruction.

If a stack switch occurred when calling the handler procedure, the IRET instruction switches back to the interrupted procedure's stack on the return.

The privilege-level protection for exception- and interrupt-handler procedures is similar to that used for ordinary procedure calls when called through a call gate (see Section 4.8.4, "Accessing a Code Segment Through a Call Gate"). The processor does:

- Load the CS, EIP, ESP, the EFLAGS register, the CS, EIP, ESP of the interrupted procedure, and the handler's stack on the return.

## Exception and Interrupt-Handler Procedures

### Figure 5-2: Page Fault Code

<table>
<thead>
<tr>
<th>Error Code</th>
<th>EFLAGS</th>
<th>CS</th>
<th>EIP</th>
<th>ESP After Transfer to Handler</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| ESP Before Transfer to Handler | EFLAGS | CS | EIP | SSP
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Stack Usage with No Privilege-Level Change

### Stack Usage with Privilege-Level Change

### Interrupted Procedure's Stack

### Handler's Stack

- If a page fault is caused by a page-level protection violation, the fault was caused by a non-present page.
- The fault was caused by a page-level protection violation.
- The access causing the fault was a read.
- The access causing the fault was a write.
- The access causing the fault originated when the processor was executing in supervisor mode.
- The access causing the fault originated when the processor was executing in user mode.

### Reserved Bit Violation

- The fault was not caused by a reserved bit violation.
- The fault was caused by reserved bits set to 1 in a page directory.

### Instruction Fetch

- The fault was not caused by an instruction fetch.
- The fault was caused by an instruction fetch.

---

If a stack switch occurred when calling the handler procedure, the IRET instruction switches back to the interrupted procedure's stack on the return.

- Load the CS, EIP, ESP, the EFLAGS register, the CS, EIP, ESP of the interrupted procedure, and the handler's stack on the return.