Input/Output (I/O)

- What is I/O?
- Waiting Loop
- Exceptions, or Interrupts
  - Hardware interrupt
  - Software interrupt
  - Application: Hurricane detector
  - Application: Time sharing

Exceptions

- Do your usual processing BUT when the keyboard is pressed then go get the keystroke. Then resume the usual processing
Exceptions

Main Program

Keystroke, causes an interrupt

Similar to jal

Interrupt Handler

Get Keystroke

Review jal

Main Program

jal

1. $31 = PC
2. jump to subroutine

Subroutine

jal

Exceptions

Main Program

J Handler

1. Finish completing instruction
2. PC is stored (e.g., epc)
3. PC = specific address A

Interrupt

A

J Handler

Handler

Restore PC, Return From Exception (rfe)

Exceptions

Main Program

J Handler

1. Finish completing instruction
2. PC is stored (e.g., epc)
3. PC = specific address A

Interrupt

A

J Handler

Handler

Examples:

Interrupt type | Addr |
--- | --- |
LTI | 4 |
ETI | 8 |
OV | 12 |
IMA | 16 |
IMR | 20 |

Examples:

Memory

<table>
<thead>
<tr>
<th>Addr</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>j Reset_Routine</td>
</tr>
<tr>
<td>4</td>
<td>j Handler_LTI</td>
</tr>
<tr>
<td>8</td>
<td>j Handler_ETI</td>
</tr>
<tr>
<td>12</td>
<td>j Handler_OV</td>
</tr>
<tr>
<td>16</td>
<td>j Handler_IMA</td>
</tr>
<tr>
<td>20</td>
<td>j Handler_IMR</td>
</tr>
</tbody>
</table>

Exceptions

CPU

External hardware interrupts

Level triggered interrupt (LTI)

Edge triggered interrupt (ETI)

Internal hardware interrupts

• Overflow (OV)
• Incorrect memory access (IMA)
• Incorrect math result (IMR)

Exceptions

Memory

<table>
<thead>
<tr>
<th>Addr</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>j Reset_Routine</td>
</tr>
<tr>
<td>4</td>
<td>j Handler_LTI</td>
</tr>
<tr>
<td>8</td>
<td>j Handler_ETI</td>
</tr>
<tr>
<td>12</td>
<td>j Handler_OV</td>
</tr>
<tr>
<td>16</td>
<td>j Handler_IMA</td>
</tr>
<tr>
<td>20</td>
<td>j Handler_IMR</td>
</tr>
</tbody>
</table>
**Hurricane Detector**

- Processor
- External Interrupt
- Hurricane Detector

**Memory**

- 4:
  - Hurricane
  - Hurricane

- Save reg. values
- Print "Head for the hills!"
- Restore reg values
- rfe

**Storage Area**

---

**Time Sharing**

- Processor wants to execute two programs P1 and P2 at the same time

**Memory**

- Interrupt Handler
- Storage
- P1
- P2
- Stored_$1
- Stored_$2
- Stored_$3
- Stored_PC

**CPU**

- Interrupt every 30 ms

---

**Time Sharing**

- **Initially:**
  - Storage = init val of all regs for P2
  - Stored_PC = Start_P2
  - Jump to Start_P1

- **Interrupt Handler**
  - Storage swap CPU Regs
  - swap Stored_PC rfe

- **Start_P1:**
  - P1

- **Start_P2:**
  - P2

---

**Software Exception**

- **Special instructions, when executed then an interrupt occurs**
- **Graceful way of moving from user status to supervisor status**
  - User status: instructions are executed but restrictions
  - Supervisor status: no restrictions
- **Application:** used in debuggers such as for breakpoints