Interrupt Driven I/O

- Asynchronously notify CPU when I/O device needs service

- So this is done via an INTERRUPT REQUEST LINE (IRQ). Think of it as a wire.

- We must go to the service routine after interrupt signal is raised. How? Think: Linkage

- When do we check for interrupts? (Store results phase)

- Must modify our RBISR:

- What do we need to save?: PC, GPRs, ...?
MMIO in HARDWARE

Diagram:

- MDR
- 4:1
- ADDR CTL
- MAR
- KBDR
- MEM
- what if:
  
  LD 22, VA | IRQ!
  Bzp Target

- can't lose our condition codes! also, what privilege are we at?

  Processor Status Word (PSW)

  ![PSW Diagram]

  current privilege level (0/1)

  sometimes called "cpl" user/os

- so: [PSW, PC, old sp, old fp]

  regs can be saved by service routine.

- "where do we save them? the stack!"

- "how do we get back? add a new instr:
  RTI (return from interrupt)"

- ![Stack Diagram]
<table>
<thead>
<tr>
<th>IN/OUT</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MMIO</td>
<td></td>
</tr>
<tr>
<td>/proc/iomem</td>
<td>/proc/ioports</td>
</tr>
<tr>
<td>/proc/interrupts</td>
<td></td>
</tr>
<tr>
<td>lspci</td>
<td></td>
</tr>
</tbody>
</table>