08 Communication

08.02 I/O synchronization

- Synchronization issues
- Program-controlled
- Interrupt
- Direct memory access

Synchronization issues

- Accessing a peripheral can take thousands of clock cycles
- The service time of a peripheral device can be of millions of clock cycles
- The arrival time of an external input is usually unpredictable

- Pipelines are designed assuming that memory accesses can be performed in a few clock cycles. If this is not the case, the pipeline is stalled
- Pipeline stalls cannot solve all synchronization issues

- I/O read/write instructions cannot be handled as memory read/write instructions

Synchronization paradigms

- Program-controlled I/O
  - Synchronization is achieved by polling the status register of the device
- Interrupt-controlled I/O
  - Synchronization is achieved by means of hardware signals that allow the peripheral device to inform the CPU
- Direct memory access (DMA)
  - Data transfer between memory and I/O devices is performed without involving the CPU

Program-controlled I/O

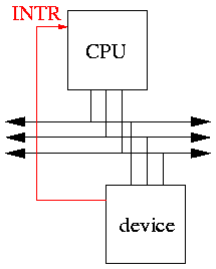
- The process that needs to access a peripheral explicitly waits by polling the status register

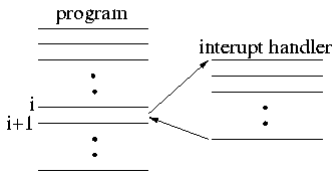
```
...
while (peripheral.status != READY) {}
...
```

- Keeps the processor busy while waiting for a peripheral
- Keeps the peripheral BUS busy to read the status register

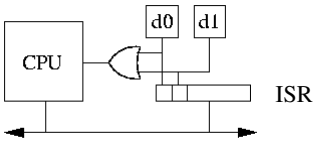
Interrupt-controlled I/O

- The device uses an extra wire to notify an event to the CPU

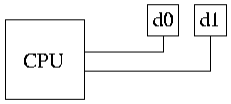




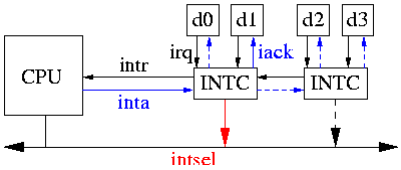
Handling multiple interrupts



(a)



(b)



(c)

Direct Memory Access