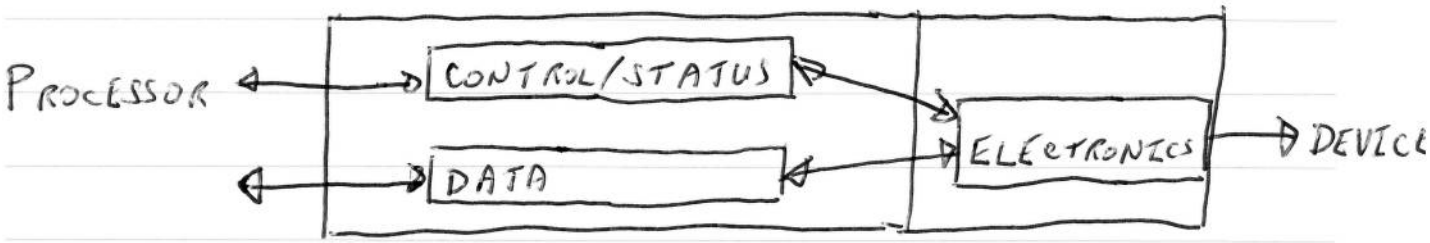
INTERFACING I/O DEVICES

I/O CONTROLLER

- CONTROL/STATUS REGISTERS
- DATA REGISTERS
- DEVICE ELECTRONICS



CONTROL / STATUS REGISTERS

- CPU TELLS DEVICE WHAT TO DO — WRITE TO CONTROL REGISTER
- CPU CHECKS WHETHER TASK IS DONE — READ THE STATUS REGISTER

DATA REGISTERS

- CPU TRANSFERS DATA TO/FROM DEVICE

DEVICE ELECTRONICS

- PERFORMS ACTUAL OPERATION
  PIXELS TO SCREEN,
  BITS TO/FROM DISK
  CHARACTERS FROM KEYBOARD

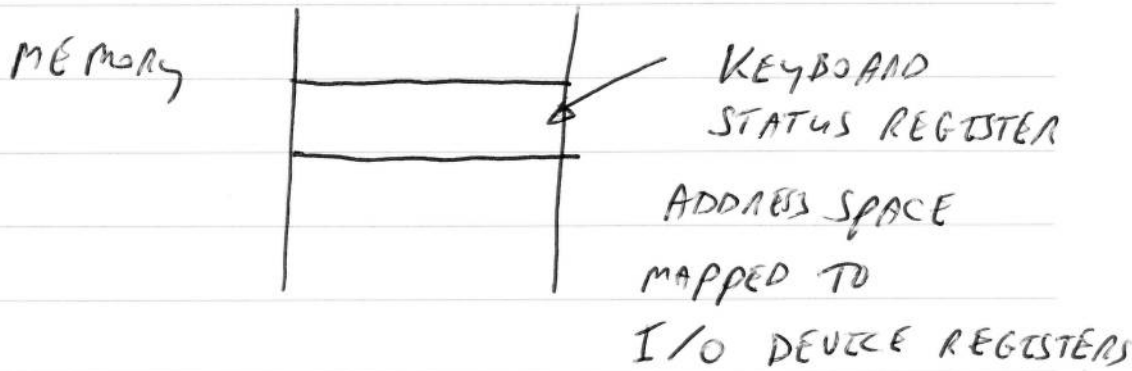
MEMORY MAPPED VS I/O INSTRUCTIONS

I/O INSTRUCTIONS

- DESIGNATE DEDICATED INSTRUCTIONS FOR I/O E.G. IN, OUT } INTEL

MEMORY MAPPED

- ASSIGN A MEMORY ADDRESS TO EACH DEVICE REGISTER
- USE DATA MOVEMENT INSTRUCTIONS
  E.G. LD ST
  LOAD VALUE STORE VALUE
  FOR CONTROL AND DATA TRANSFER
- HARDWARE INTERCEPTS THESE ADDRESSES



- POLLING
- INTERRUPTS
- DMA

POLLING

PROCESSOR CONTINUALLY CHECKS THE DEVICE STATUS REGISTER UNTIL

- NEW DATA ARRIVES OR
- DEVICE IS READY FOR NEXT DATA

POLLING EXAMPLES

- COST OF A POLL IS 400 CLOCK CYCLES.
- CPU IS 500 Mhz
- HOW MUCH OF THE CPU TIME, AS A PERCENTAGE IS NEEDED TO POLL?

— 1. A mouse that requires 30 polls per second?

CPU: 500 Mhz ⇒ 500,000,000 (PULSES OF CLOCK CYCLES / SECOND

Mouse – 30 polls / second

CPU: 500Mhz = 500,000,000 cycles / second
Poll Mouse : 400 clock cycles for 1 poll
30 polls x 400 clocks = 12000 clocks
IN EVERY SECOND IS SPENT BY THE CPU POLLING THE MOUSE.

$$\frac{12000}{500,000,000} \times \frac{100}{1} = 0.0024\%$$ of time spent polling mouse

3. Hard Disk Calculation

A hard disk that sends 16 bytes at a time and achieves 4MB/second

4MB / 16 = (approx) 4,000,000 / 16 = 250,000

250,000 (16 byte) packets / second.
This means 250000 polls that will be done every second.

250,000 polls × 400 cycles for a poll
= 100,000,000 cycles

$\frac{100,000,000}{500,000,000} \times \frac{100}{1}$ = 20% of time spent polling the hard disk

Interrupts

Processor is told by the device when to look. It is interrupted by the device.

- Dedicated wires on the bus are assigned for interrupts i.e. lines
- When a device wants to send an interrupt it asserts the corresponding line
- The processor checks for interrupts after each instruction is executed. This requires "zero time" as it is done in parallel with instruction execution.

(6)

IF AN INTERRUPT IS PENDING.
(I.E. A LINE IS ASSERTED) THE PROCESSOR

1. SAVES THE PC AND PERHAPS SOME REGISTERS
2. SWITCHES TO KERNEL MODE (I.E. PRIVILEGED MODE)
3. JUMPS TO A LOCATION IN MEMORY WHICH STORES THE INTERRUPT VECTOR TABLE.

AT THIS POINT THE OS TAKES OVER.