

8 EIDE/ATA-2/ATA-3 INTERFACE

The EIDE/ATA-2/ATA-3, or for short ATA interface, supports four ATA busses without external logic, see Figure 8-1 on page 75. Each bus is capable of accessing two ATA units making a total of eight. More than eight units can be accessed using external logic.

An ATA unit (e.g. a hard-disk or a CD-rom player) is controlled via a set of registers in each unit, which can be read and/or written. By writing to these registers the ATA unit can be made to perform commands such as reading or writing data from or to a disk.

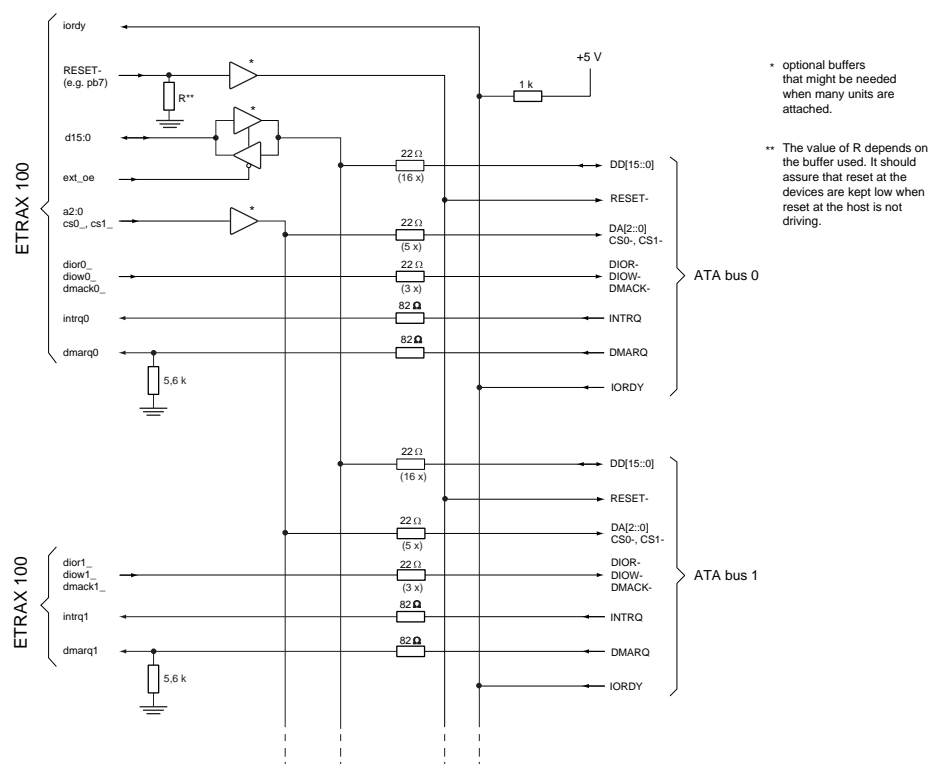


Figure 8-1 How to connect the ATA bus.

The reset signal of the ATA interface (RESET- (Device reset)) is not supported by the hardware of the ETRAX 100, but must be chosen from an available pin and handled by the software.

8.1 DATA TRANSFER

The ATA interface of ETRAX 100 can be driven either by internal DMA or by internal registers. The ATA unit uses programmed input/output (PIO) for transferring commands and also for transferring data if the unit does not have a DMA handshaking protocol.

The internal DMA of ETRAX 100 must not be confused with the DMA handshaking on the ATA bus that many ATA units use. The ETRAX 100 DMA can be used both when directly accessing the registers in the ATA unit (PIO, programmed input/output) and when using the DMA handshaking protocol of the unit.

8.1.1 Programmed Input/Output (PIO)

The commands to the ATA unit are always transferred to the unit by PIO. They are written directly into the command registers. Data can be transferred in the same way. The command or the data to be transferred is written into an internal register in ETRAX 100. Data, fetched from the ATA unit, is written into another register in ETRAX 100.

8.1.2 ATA DMA Handshaking

Most new ATA units use DMA handshaking to transfer data to and from a unit. The unit sets its DMARQ signal high when it is ready to receive or deliver data. If the amount to be transferred to or from the unit is large, it is possible that the unit neither can accept nor produce all data in one burst. It will then lower its DMARQ for periods of time during the transfer. It may take some time until the unit is able to accept or produce more data after it has lowered its DMARQ, so this time can be used to talk to another ATA unit. This is made possible through an interrupt produced when DMARQ goes low (ata_dmaend interrupt).

8.1.3 ETRAX 100 Register Access

When writing or reading a register in an ATA unit an internal register in ETRAX 100 is used. When writing to the register a transfer is started. Only one transfer is made on one access. The result, e.g. the data transferred from the ATA unit, is found in another register.

8.1.4 ETRAX 100 DMA Access

An alternative to register access is to let the internal DMA of ETRAX 100 drive the ATA interface. The internal DMA is then used to transfer data to and from the ATA unit. A transfer counter is used. For each word or byte transferred the counter is decremented. When the counter reaches zero the transfer is stopped

and, if data was transferred to ETRAX 100, an end-of-packet is signalled to ETRAX 100 DMA.

8.2 INTERRUPTS

The ATA interface has nine interrupts:

- `ata_irq0`, `ata_irq1`, `ata_irq2`, and `ata_irq3`
If enabled, an interrupt is generated whenever an IRQ signal on any of the ATA buses is one. There is one interrupt for each of the four buses.
- `ata_drq0`, `ata_drq1`, `ata_drq2`, and `ata_drq3`
If enabled, an interrupt is generated whenever a DRQ signal on any of the ATA buses is one. There is one interrupt for each of the four buses.
- `ata_dmaend`
If this is enabled it gives an interrupt whenever the DMARQ of the selected ATA bus is zero. This makes it possible to detect DMA burst ends.

8.3 TIMING

If PIO is used, the time for transferring one piece of data varies between 140 ns and 600ns depending on what mode the ATA is put in.

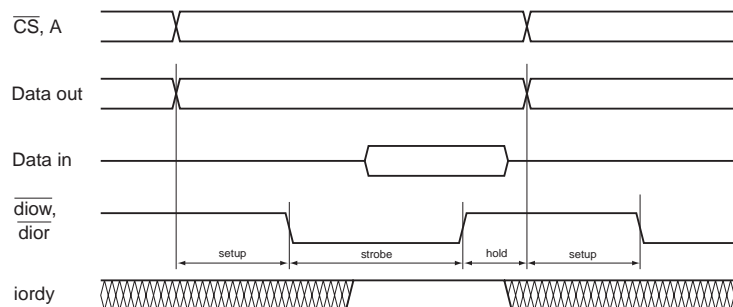


Figure 8-2 PIO timing

When DMA handshaking is used, the time varies from 120 ns to 480 ns depending on the mode.

8 EIDE/ATA-2/ATA-3 Interface

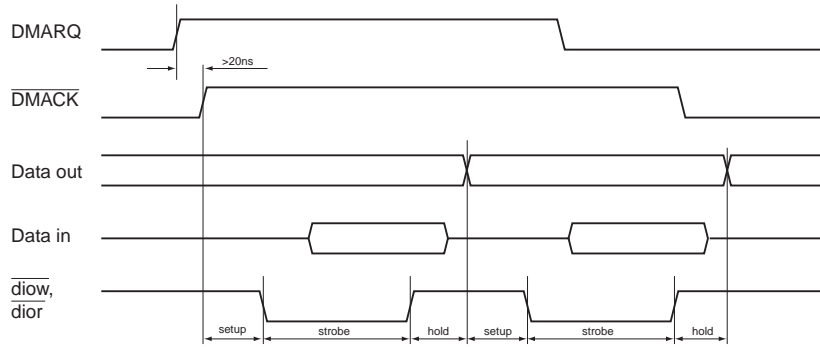


Figure 8-3 DMA multiword timing