

Re: Parallel port hardware

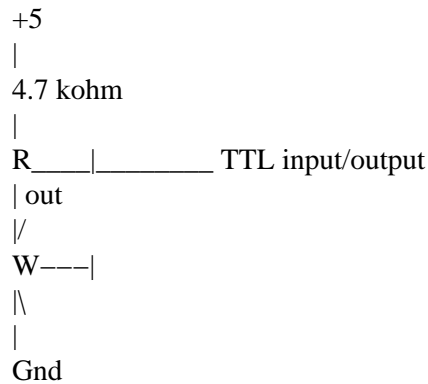
Source: <http://sci.tech--archive.net/Archive/sci.electronics.design/2007-10/msg00569.html>

- *From:* Roger Hamlett <rogerspamignored@xxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 04 Oct 2007 09:01:17 GMT
-

petrus bitbyter wrote:

"Jon Slaughter" <Jon_Slaughter@xxxxxxxxxxxxx> schreef in bericht
[news: QMMi.66\\$LD2.34@xx](mailto:QMMi.66$LD2.34@xx)

Does anyone know how the open collector control port in the SPP is



I can't seem to find anything that shows how the port is read from and I'm trying to understand its behavior when reading based on it. The above is just what I think it probably is but I could be wrong. When reading from the port W must be 0 and then R follows out. But this isn't quite true because R seems to latch W to 1 and I have to write W = 0 to allow R to follow out again.

Basically I'm trying to setup an input and output line into a control port pin. I read that I have to use all the pins as either input or output but not mixed? This is probably because I have to reset the pin and I cannot do them individually? If this is the case then I have to use a status port pin to read the line when it is working as an input?... which makes it more complicated.

This is where I'm getting most of my info:

<http://www.beyondlogic.org/spp/parallel.htm>

and this is specifically where I'm having trouble:

Re: Parallel port hardware

"An external 4.7k resistor can be used to pull the pin high. I wouldn't use anything lower, just in case you do have an internal pull up resistor, as the external resistor would act in parallel giving effectively, a lower value pull up resistor. When in high impedance state the pin on the Parallel Port is high (+5v). When in this state, your external device can pull the pin low and have the control port change read a different value. This way the 4 pins of the Control Port can be used for bi-directional data transfer. However the Control Port must be set to xxxx0100 to be able to read data, that is all pins to be +5v at the port so that you can pull it down to GND (logic 0)."

Not sure why I have to really do all port pins for bi-direction. (Obviously I can't read and write at the same time but I think I can interleave them to get what I want)

Any ideas?

Thanks,
Jon

I still want to lay my hands on the original IBM hardware manual. One of the reasons is that parallel port. Nevertheless I saw several "compatibel" schematics all with several differences. One thing is clear to me: The output lines were never meant to do input. I have the schematic of a printerport that had no inputs but the status lines. Others had "inputs" on control- and data lines but they were only meant to read back the status of that outputs. In the old days inputs used to be SN74LS14 inverting Schmidt triggered buffers, as were the read back inputs of the control lines. The control line outputs used to be SN7406 inverting open collector buffers. As the control lines are open collector, you can use their read back inputs for real input when you drive that control lines high... most of the time. I have a schematic in front of me in which the read back of the /INIT control is taken from the input of the SN7406 buffer rather than from its output. So you will never read back the real status of that printer pin. No need to say it will fail as an input as well.

The schematic in the chapter
"Using The Parallel Port to Input 8 Bits"
will fail in this case.

So if you want to stay on the safe side, don't use output pins for input. If you have to, you will have to check the printerport involved for every (type of) computer. Once you'll have to do so, it will be worthwhile to check for other properties of the printerport at hand. Almost all but the oldest computers have printerports that somehow can do bidirectional data transfer. If you have the choice, use EPP ports (or USB :)

petrus bitbyter

The oldest 'hardware' IBM motherboard tech-ref, I have, is the one for the original 'AT'. Have the BIOS ones for an XT, but can't see the hardware ones.

However the parallel port, is described in the manual for the video card (wasn't on the motherboard on these machines). The output lines, use a 74LS374 latch (or 373, on the multi-io cards). The output enable, is permanently grounded, making the data lines 'output only'. This was the 'SPP' port.

Re: Parallel port hardware

Devices that performed 'input' from these ports, used 'nibble wide' 4 bit transfers, using the four input status lines for the return direction.

The first IBM, to have bidirectional printer pins, was the PS/2. This doesn't have the automatic latch cycle of the EPP port, or the DMA abilities of the ECP port, but allowed the main data pins to be read as well as written. This is the 'bidirectional' port.

Then you have the latter EPP, and ECP ports.

Most chips generating 'modern' printer ports, can be programmed to emulate the SPP mode, but few emulate the simple bidirectional mode of the PS/2.

There used to be a standard 'mod' for the old printer cards, of cutting pin1 on the output driver chip (the output enable line), and connecting it to a spare latch on an LS174 on the card, allowing the direction to be switched. The outputs on the parallel port, were not 'open collector' drives.

Best Wishes

.