UPDATE NEWS: version 1.33 - 26 Aug 98

Interrupt management in this driver has become, over time, increasingly odd and difficult to explain - this has been mostly due to my own mental inadequacies. In recent kernels, it has failed to function at all when compiled for SMP. I've fixed that problem, and after taking a fresh look at interrupts in general, greatly reduced the number of places where they're fiddled with. Done some heavy testing and it looks very good. The driver now makes use of the __initfunc() and __initdata macros to save about 4k of kernel memory. Once again, the same code works for both 2.0.xx and 2.1.xx kernels.

UPDATE NEWS: version 1.32 - 28 Mar 98

Removed the check for legal IN2000 hardware versions: It appears that the driver works fine with serial EPROMs (the 8-pin chip that defines hardware rev) as old as 2.1, so we'll assume that all cards are OK.

UPDATE NEWS: version 1.31 - 6 Jul 97

Fixed a bug that caused incorrect SCSI status bytes to be returned from commands sent to LUNs greater than 0. This means that CDROM changers work now! Fixed a bug in the handling of command-line arguments when loaded as a module. Also put all the header data in in2000.h where it belongs. There are no longer any differences between this driver in the 2.1.xx source tree and the 2.0.xx tree, as of 2.0.31 and 2.1.45 (or is it .46?) - this makes things much easier for me...

UPDATE NEWS: version 1.30 - 14 Oct 96

Fixed a bug in the code that sets the transfer direction bit (DESTID_DPD in the WD_DESTINATION_ID register). There are quite a few SCSI commands that do a write-to-device; now we deal with all of them correctly. Thanks to Joerg Dorchain for catching this one.

UPDATE NEWS: version 1.29 - 24 Sep 96

The memory-mapped hardware on the card is now accessed via the 'readb()' and 'readl()' macros - required by the new memory management scheme in the 2.1.x kernel series. As suggested by Andries Brouwer, 'bios_param()' no longer forces an artificial 1023 track limit on drives. Also removed some kludge-code left over from struggles with older (buggy) compilers.

UPDATE NEWS: version 1.28 - 07 May 96

Tightened up the "interrupts enabled/disabled" discipline

```
   in 'in2000_queuecommand()' and maybe 1 or 2 other places.
   I _think_ it may have been a little too lax, causing an
   occasional crash during full moon. A fully functional
   /proc interface is now in place - if you want to play
   with it, start by doing 'cat /proc/scsi/in2000/0'. You
   can also use it to change a few run-time parameters on
   the fly, but it's mostly for debugging. The curious
   should take a good look at 'in2000_proc_info()' in the
   in2000.c file to get an understanding of what it's all
   about; I figure that people who are really into it will
   want to add features suited to their own needs...
   Also, sync is now DISABLED by default.

UPDATE NEWS: version 1.27 - 10 Apr 96

   Fixed a well-hidden bug in the adaptive-disconnect code
   that would show up every now and then during extreme
   heavy loads involving 2 or more simultaneously active
   devices. Thanks to Joe Mack for keeping my nose to the
   grindstone on this one.

UPDATE NEWS: version 1.26 - 07 Mar 96

   1.25 had a nasty bug that bit people with swap partitions
   and tape drives. Also, in my attempt to guess my way
   through Intel assembly language, I made an error in the
   inline code for IO writes. Made a few other changes and
   repairs - this version (fingers crossed) should work well.

UPDATE NEWS: version 1.25 - 05 Mar 96

   Kernel 1.3.70 interrupt mods added; old kernels still OK.
   Big help from Bill Earnest and David Willmore on speed
   testing and optimizing: I think there's a real improvement
   in this area.
   New! User-friendly command-line interface for LILO and
   module loading - the old method is gone, so you'll need
   to read the comments for 'setup_strings' near the top
   of in2000.c. For people with CDROM's or other devices
   that have a tough time with sync negotiation, you can
   now selectively disable sync on individual devices -
   search for the 'nosync' keyword in the command-line
   comments. Some of you disable the BIOS on the card, which
   caused the auto-detect function to fail; there is now a
   command-line option to force detection of a ROM-less card.

UPDATE NEWS: version 1.24a - 24 Feb 96

   There was a bug in the synchronous transfer code. Only
   a few people downloaded before I caught it - could have
   been worse.

UPDATE NEWS: version 1.24 - 23 Feb 96

   Lots of good changes. Advice from Bill Earnest resulted
   in much better detection of cards, more efficient usage
```

of the fifo, and (hopefully) faster data transfers. The jury is still out on speed - I hope it's improved some. One nifty new feature is a cool way of doing disconnect/ reselect. The driver defaults to what I'm calling 'adaptive disconnect' - meaning that each command is evaluated individually as to whether or not it should be run with the option to disconnect/reselect (if the device chooses), or as a "SCSI-bus-hog". When several devices are operating simultaneously, disconnects are usually an advantage. In a single device system, or if only 1 device is being accessed, transfers usually go faster if disconnects are not allowed.

The default arguments (you get these when you don't give an 'in2000' command-line argument, or you give a blank argument) will cause the driver to do adaptive disconnect, synchronous transfers, and a minimum of debug messages. If you want to fool with the options, search for 'setup_strings' near the top of the in2000.c file and check the 'hostdata->args' section in in2000.h - but be warned! Not everything is working yet (some things will never work, probably). I believe that disabling disconnects (DIS_NEVER) will allow you to choose a LEVEL2 value higher than 'L2_BASIC', but I haven't spent a lot of time testing this. You might try 'ENABLE_CLUSTERING' to see what happens: my tests showed little difference either way. There's also a define called 'DEFAULT_SX_PER'; this sets the data transfer speed for the asynchronous mode. I've put it at 500 ns despite the fact that the card could handle settings of 376 or 252, because higher speeds may be a problem with poor quality cables or improper termination; 500 ns is a compromise. You can choose your own default through the command-line with the 'period' keyword.

```
------------------------------------------------
***********  DIP switch settings  **************
------------------------------------------------
```

sw1-1	sw1-2	BIOS address (hex)
off	off	C8000 - CBFF0
on	off	D8000 - DBFF0
off	on	D0000 - D3FF0
on	on	BIOS disabled

sw1-3	sw1-4	IO port address (hex)
off	off	220 - 22F
on	off	200 - 20F
off	on	110 - 11F
on	on	100 - 10F

sw1-5	sw1-6	sw1-7	Interrupt
off	off	off	15

off	on	off	14
off	off	on	11
off	on	on	10
on	-	-	disabled

sw1-8 function depends on BIOS version. In earlier versions this controlled synchronous data transfer support for MSDOS:
   off = disabled
   on = enabled
In later ROMs (starting with 01.3 in April 1994) sw1-8 controls the "greater than 2 disk drive" feature that first appeared in MSDOS 5.0 (ignored by Linux):
   off = 2 drives maximum
   on = 7 drives maximum

sw1-9	Floppy controller
off	disabled
on	enabled

I should mention that Drew Eckhardt's 'Generic NCR5380' sources were my main inspiration, with lots of reference to the IN2000 driver currently distributed in the kernel source. I also owe much to a driver written by Hamish Macdonald for Linux-m68k(!). And to Eric Wright for being an ALPHA guinea pig. And to Bill Earnest for 2 tons of great input and information. And to David Willmore for extensive 'bonnie' testing. And to Joe Mack for continual testing and feedback.

John Shifflett   jshiffle@netcom.com