PC532E Users Guide

4. August 2010

Content

1. What is the PC532E ?

The PC532E is a successor of the PC532. The „E“ has two meanings : first meaning is Emulation and the second meaning is Enhanced. The Emulation put all peripherals of the PC532 into a Field Programmable Gate Array (FPGA) of ALTERA. The goal is that version 1.5.3 of NetBSD is able to run on the machine without modifications. The enhancements will allow the use of modern hardware like IDE and USB. For this purpose the operating system software has to be modified.

2. Limits of Emulation

The effort for emulation was limited to the point when NetBSD was running. For example the number of serial devices was set to one SCN2681 and from this device only channel A was implemented for operation. Channel B is limited to what the operating system needs to boot succesfully. Channel A supports only two baud rates. The setting for 9600 baud is implemented, every other setting will run the serial channel at 57600 baud. Other settings like the number of data bits or stop bits are fixed to 8 data bits and 1 stop bit. If really necessary the FPGA can be modfied to support other values. The SCSI interface is reverse enginered by watching which commands are executed during the boot process and normal operation. This was limited to one hard disk drive. The FPGA translates all SCSI operations into IDE operations. If a command is not supported by IDE the FPGA generates a useful response for the operating system. The size of the SCSI drive is limited to 8GB due to the translation of 24 bit LBAs from SCSI to IDE.

3. Planned Enhancements

The planned enhancements are native IDE use and USB support. The FPGA allows the CPU to direct access the IDE interface. To make use of it NetBSD needs a driver. The already existing drivers for IDE can not be used because of different hardware. Therefore a new driver has to be implemented. As soon as the new driver is available the translator hardware SCSI-to-IDE inside the FPGA can be taken out. This frees resources which can be used for other tasks. One topic high on the agenda is USB support. The plan is to implement an UHCI or OHCI device and to use the existing USB driver of NetBSD.

4. Known Issues of Emulation

There is one known issue with the emulation. The operating system sents the message :

„ncr0: timeout waiting for final SCI_DSR_DREQ“

The occurence is very low. And it was never observed that any problems occured after this message. The disk drive system continued to work properly. It seems that the number of interrupts may influence the number of occurences. This was observed during downloads on the serial port.

5. Custom Hardware

The CPU board has one socket for a PS2 DIMM of EDO type. The capacity can be 16 MB in one rank (one RAS) or 32 MB with two ranks (two RAS). The supply voltage is 5 V. There are different DIMM versions available with 4 k or 2 k refresh. The PC532E supports only 2 k refresh. The page size is therefore 8 kB.
A 2.5 inch parallel ATA (IDE) drive with a height of 9.5 mm can be attached to the connector on the PC532E IO board. An external dc power supply with 10 to 15 volt and a capacity of 1.5 A minimum is required.

6. Boot process of the PC532E

The CPU board of the PC532E has no boot rom device. Therefore a part of DRAM is used for the boot code. The system supports 16 MB or 32 MB DRAM. From this memory always the last 64 KB block is used for the boot code. In normal operation the CPU is not able to access the last block of the DRAM. Therefore the operating system will detect (DRAM-Size) – 64 KB as available main memory. Booting the computer will need a boot rom in any way and the PC532E uses its FPGA for this purpose. Inside the FPGA a small 2 KB ROM is implemented which contains the so-called „micro-boot“ code. This micro-boot will load 64 KB of boot code from the IDE drive. The LBA of the boot code is fixed to x'100_0000. This is the first block after the 8GB SCSI drive range. The micro-boot code uses IDE direct and will initiate the IDE drive for the correct data transfer method used during emulation. After loading the boot code the micro-boot switches to the normal boot process. This means that the boot code is now accessible for the CPU from address 0. The switch will happen 30 seconds after power-up or by pressing the space key. Most of the other keys will also switch but some of them are used for loading a new boot code. See the chapter „loading of boot code“. The PC532E starts with a transmission speed of 9600 baud for the serial port.

7. „E“ features of the current implementation

The CPU can access some new features compared to the PC532. The following table gives an overview.

Startaddress	Feature	Description
x'E000_0000	IDE Register access	Direct access of IDE interface at the register level
x'E400_0000	IDE DMA access	Fast data transfer to/from IDE drive
x'E800_0000	Micro-boot ROM	2 KB ROM containing micro-boot code
x'E900_0000	CONFIG register	Configuration of various system features
x'EA00_0000	IDE DMAC	DMA controller for IDE access

Addressmap of new PC532E features

The address range to access a feature is always between startaddress and up to the startaddress – 1 of the next feature. This will result in mirroring a feature, i.e. the micro-boot rom is mirrored 8192 times.
The interface between the FPGA and the CPU is 32 bits wide. Some registers can be written also with bytes because they use the BE (Byte Enable) signal from the NS32532 CPU. Reading is always possible with byte, word and double-word data packets. The description of registers is always done in 32 bits.
The interrupt vector number 4 for the SCSI drive is also used for the direct IDE drive access. Both drive access methods can be mixed when each access sets the required control bits in the hardware and interrupt vectors in the software. The IDE drive itself needs no different setup. Always Multiword DMA transfer mode 2 is used with the READ DMA and WRITE DMA commands.

8. Description of CONFIG Register

The CONFIG register implements five bits to control important functions of the PC532E. Access is normally done only during the micro-boot phase.

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-

Byte 3 of CONFIG register

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-

Byte 2 of CONFIG register

15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	BIGM

Byte 1 of CONFIG register

7	6	5	4	3	2	1	0
-	-	-	-	UCODE	ENAU	HRST	LEDC

Byte 0 of CONFIG register

The CONFIG register is reset to all „0“. It is only writable. Byte-Enable can be used.

BIGM	BIG Memory, 1 = 32 MB DRAM, 0 = 16 MB DRAM . The micro-boot code will determine DRAM size and set this bit accordingly. BIGM is using an own byte to allow independent modifying of the other bits.
UCODE	Micro-boot ROM use : 0 = micro-boot ROM mirrored at address 0 1 = micro-boot ROM only accessible at address x'E800_0000.
ENAU	Enable Automatic SCSI-to-IDE translation : 0 = no automatic translation of SCSI commands. If the CPU accesses IDE directly this bit must be set to 0. 1 = translation hardware active. This is the normal case for NetBSD 1.5.3
HRST	Hard disk drive Reset : 0 = active . 1 = normal operation
LEDC	LED control : 0 = LED is on . 1 = LED on/off set from IDE usage
-	not implemented , write 0 for future compatibility

9. Access of IDE registers

The registers of the IDE specification are mapped into the memory of the PC532E CPU. The IBM PC uses IO-addresses for this purpose.

Addr-Offset	Byte 3	Byte 2	Byte 1	Byte 0
0	-	-	Data [15:8]	Data [7:0]
4	-	-	-	Feat./Err-Reg
8	-	-	-	Sector Count
12	-	-	-	Sector Number
16	-	-	-	Cylinder Low
20	-	-	-	Cylinder High
24	-	-	-	Device + Head
28	-	-	-	CMD/Status
56 (x'38)	-	-	-	Alternate Status

Addressmap of IDE registers

Access of offset values in the range of 0 – 63 not shown will result in undefined operation. The SCSI-to-IDE emulation uses only the commands READ DMA (x'C8) , WRITE DMA (x'CA) and FLUSH CACHE (x'E7). The micro-boot code uses in addition IDENTIFY DEVICE (x'EC) and SET FEATURES (x'EF). For a detailed description of the IDE interface please read a disk drive data sheet or look into the web for further information.

10. IDE DMA access

To accellerate data transfer from/to the IDE drive a method similar to the SCSI access can be used. During SCSI access the CPU is responsible for the data transfer. In the original PC532 this transfer was done byte by byte. In the PC532E the bus between the CPU and the FPGA is 32 bit wide. For a read operation the FPGA will first collect the required number of bytes from the drive and then serve the request from the CPU. IDE is even more advantageous because the data channel is 16 bit wide and only two accesses are needed to get 32 bit. The same will happen during write operation. The CPU writes a double-word and the FPGA splits the data in two 16 bit words for the IDE interface. A direct IDE data transfer of the CPU can use the same behaviour if it accesses inside the address range x'E400_0000 – x'E7FF_FFFF. Please note that a DMA mode must be selected in the IDE drive and an IDE access with DMA capabiliy must be used. Otherwise the CPU will hang up (no READY signal) .

11. IDE DMA Controller : DMAC

The DMAC is a powerful tool to free the software from the task of moving data around. In addition it is much faster compared to software. The transfer speed between the sector buffer inside the FPGA and the DRAM is 100 MB/s (25 MHz * 4 Byte). The speed between the drive and the FPGA is 12.5 MB/s with Multiword DMA Mode 2. The DMAC has two registers to control its operation.

Addr-Offset	Register	Description
0	DADDR	Addresspointer to DRAM + R/W-Flag
4	TCOUNT	Number of Bytes to transfer

Addressmap of DMAC registers

Both registers are only writable as 32 bit double words.

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	**DA[24]**
23	22	21	20	19	18	17	16
DA[23]	**DA[22]**	**DA[21]**	**DA[20]**	**DA[19]**	**DA[18]**	**DA[17]**	**DA[16]**
15	14	13	12	11	10	9	8
DA[15]	**DA[14]**	**DA[13]**	**DA[12]**	**DA[11]**	**DA[10]**	**DA[9]**	**DA[8]**
7	6	5	4	3	2	1	0
DA[7]	**DA[6]**	**DA[5]**	**DA[4]**	**DA[3]**	**DA[2]**	-	**WR**

DADDR Register

The DADDR register is only writable. It's content after reset is undefined.

DA[24:2] The address DA covers the maximum DRAM amount of 32 MB. Addresses are always set on double-word boundaries. DA can even point to the last 64 KB of the DRAM to write/read the bootcode.
WR WR=1 is a WRITE transfer to the IDE drive , WR=0 is a READ from the IDE drive.
- not implemented, write „0“ for future compatibility

31	30	29	28	27	26	25	24
„0“	**„0“**	**„0“**	**„0“**	**„0“**	**„0“**	**„0“**	**„0“**
23	22	21	20	19	18	17	16
„0“	**„0“**	**„0“**	**„0“**	**„0“**	**„0“**	**TC[17]**	**TC[16]**
15	14	13	12	11	10	9	8
TC[15]	**TC[14]**	**TC[13]**	**TC[12]**	**TC[11]**	**TC[10]**	**TC[9]**	**„0“**
7	6	5	4	3	2	1	0
„0“	**„0“**	**„0“**	**„0“**	**„0“**	**„0“**	**„0“**	**„0“**

TCOUNT Register

The TCOUNT register can be read or written. After reset it's content is 0.

TC[17:9] The transfer count in bytes. The bits [8:0] can not be set. Therefore transfer count is always a multiple of 512 B which is the sector size of an IDE drive. A maximum of 261632 bytes can be transfered.
„0“ this bits are always read as „0“ and can not be set. Write „0“ for future compatibility.

12. Using an IDE Drive with DMAC

The DMAC is a feature which can only be used when the automatic translation from SCSI to IDE is switched off. Therefore bit ENAU of the CONFIG register has to be set to „0“. Accessing an IDE drive with the help of the DMAC is straightforward. First the IDE command is written in the drive's registers. Next the address pointer to the DRAM is set. The last register to be written is the transfer count register TCOUNT. Now the DMAC hardware is active and waiting for the drive to signal a dma request. The software driver will enable the interrupt and exit. The drive always sent an interrupt at the end of the command to the host. The host should first check wether TCOUNT is 0. In case of a READ command it is possible that the content is x'200. This means that one block of data has still to be written to the DRAM. The interrupt service routine (ISR) should then wait in a small loop until TCOUNT is 0. A WRITE should never show this behavior. If any other number except 0 and x'200 is found in TCOUNT something has gone wrong. In this case the ISR should terminate the DMAC by writing 0 to TCOUNT. After checking TCOUNT the ISR should check the status register of the drive for errors. For DMA accesses the DRQ bit in the IDE status register is not valid.
Although it is not done in an operating system it is possible during program development that an IDE register access is performend when the DMAC is activated (TCOUNT>0) . This is of high risk for the system stability because the current hardware in the FPGA may not work correctly in this situation.

13. Writing an IDE driver for NetBSD

It is important to notice that the DMAC works with real memory addresses. In case of NetBSD the operating system is working in virtual mode and may see linear addresses for buffers which are in reality not linear. This may create a significant overhead and may split one transfer in multiple accesses.
The drive characteristics can be fetched by the command „IDENTIFY DEVICE“, opcode x'EC. This command uses the PIO transfer method. There are drives from FUJITSU which know a command „IDENTIFY DEVICE DMA“, opcode x'EE, which other drives don't know. A 160 GB Western Digital drive is an example for this type. If „IDENTIFY DEVICE“ is used it is recommended that the IDE interface is set to PIO mode 2 to access the drive. Maybe it is helpful to look in the actual NetBSD code which is already supporting IDE drives.

14. Loading of Boot Code

To load a new boot code (or the first boot code ever) the micro-boot code supports two methods. The „crc“-method uses the protocol defined in the download programm of the NetBSD version 1.5.3. It uses a crc word at the end of the transmission. The „hex“-method uses INTEL Hex-Format which contains a checksum in each line. To start a download the user has to type either „crc“ or „hex“ followed by carriage-return after power-up. After successfull download the PC532E writes the boot code to disk and starts to run the code. If there was a transmission error the PC532E stops working. Transmission speed during micro-boot is fixed to 9600 baud. A boot code size of less than 64 KB is possible. The micro-boot code always writes and reads 64 KB from disk.

15. The perfect Download

The download program which is described in NetBSD 1.5.3 does not use a feedback during transmission. It was observed that larger files (10 MB or more) could not be downloaded succesfully. What happened in most cases was a ring buffer overrun. To avoid this the program was modified. Now a synchronization character („!“) is sent each time when (len & 127) == 0. „len“ is the number of bytes to receive and counts downward. The host is waiting for the synchronization character. With this method download was always successfull even when NetBSD showed the one known issue during transmission.