17 June 2006, 21:37 #1

bluea

Posts: n/a

Gayle Hardware Registers

Hello, is there any information about the Gayle hardware registers?

I am trying to implement an AT-IDE interface. I already know the IDE-registers at 0xda0000 and 0xda1000 from different Linux docs. But an A1200 Kickstart 40.68 is ignoring the idedrive at all. If I am running the A1200 Kick none of the above addresses are accessed.

I think the Kickstart checks at first if Gayle is available... but how?

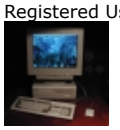
But after extensive search I did not find any more information about that.

Is there somebody here who can help??

Thanks in advance!
Thomas

17 June 2006, 23:23 #2

patrik
Registered User



Join Date: Jan 2005
Location: Umeå
Age: 41
Posts: 652

Maybe there is some info in this project, which claims to give the A500 an A1200 compatible IDE-interface.

18 June 2006, 23:57 #3

ganralf
Registered User

Join Date: May 2006
Location: Germany
Posts: 97

Gayle also has some registers for PCMCIA stuff and Interrupts. If you are already in the linux source take a look into /source/include/asm-m68k/amigayle.h. You should also look into AmigaOS' card.resource autodocs and /resource/card.h.

I guess, exec looks for a Gayle in those registers and if it doesn't find one scsi.device doesn't even start.

My findings on Gayle's registers so far are:

Code:

```
0xda0000        // Data
0xda0006        // Error | Feature
0xda000a        // Sector Count
0xda000e        // Sector Number
0xda0012        // Cylinder Low
0xda0016        // Cylinder High
0xda001a        // Device / Head
0xda001e        // Status | Command
0xda101a        // Control
0xda8000        // Gayle Status
0xda9000        // Gayle INTREQ
0xdaa000        // Gayle INTENA
0xdab000        // Gayle Config
```

21 June 2006, 17:55 #4

Toni Wilen
WinUAE developer

Join Date: Aug 2001
Location: Hämeenlinna/Finland
Age: 47
Posts: 25,301

A1200 Kickstart seems to only access Gayle IDE registers after some kind of Gayle detection code: (log from WinUAE with A1200 KS3.1 ROM, format is address, data, length of access)

45-921 [4 225x011]: GAYLE_WRITE 00DE109A=FFFFBFFF (2) PC=00F80480
45-921 [4 225x011]: GAYLE_WRITE 00DE1000=00000000 (1) PC=00F804AE
45-921 [4 225x011]: GAYLE_READ 00DE1000 PC=00F804CA
45-921 [4 225x011]: GAYLE_READ 00DE1000 PC=00F804CA
45-921 [4 225x011]: GAYLE_READ 00DE1000 PC=00F804CA
45-921 [4 225x011]: GAYLE_READ 00DE1000 PC=00F804CA
45-921 [4 225x011]: GAYLE_READ 00DE1000 PC=00F804CA
45-921 [4 225x011]: GAYLE_READ 00DE1000 PC=00F804CA
45-921 [4 225x011]: GAYLE_READ 00DE1000 PC=00F804CA
45-921 [4 225x011]: GAYLE_READ 00DE1000 PC=00F804CA
45-921 [4 226x035]: GAYLE_WRITE 00DE109A=FFFFBFFF (2) PC=00FB706E
45-921 [4 226x035]: GAYLE_WRITE 00DE1000=00000000 (1) PC=00FB70B0
45-921 [4 226x035]: GAYLE_READ 00DE1000 PC=00FB7042
45-921 [4 226x035]: GAYLE_READ 00DE1000 PC=00FB7042
45-921 [4 226x035]: GAYLE_READ 00DE1000 PC=00FB7042
45-921 [4 226x035]: GAYLE_READ 00DE1000 PC=00FB7042

A4000 ROM does access A4000 IDE registers without testing anything.

19-625 [105 226x006]: GAYLE_WRITE 00DD203A=FFFFFFA0 (1) PC=00F8B0DE
19-625 [105 226x006]: GAYLE_READ 00DD203E PC=00F8AEEE
19-625 [105 184x007]: GAYLE_READ 00DD203A PC=00F8AEEE
19-625 [105 111x008]: GAYLE_READ 00DD2026 PC=00F8AEEE
19-625 [105 038x009]: GAYLE_READ 00DD2032 PC=00F8AEEE
19-890 [118 226x001]: GAYLE_READ 00DD203E PC=00F8AEEE
19-890 [118 141x002]: GAYLE_READ 00DD203A PC=00F8AEEC
19-890 [118 005x003]: GAYLE_READ 00DD2026 PC=00F8AEEC
19-890 [118 096x003]: GAYLE_READ 00DD2032 PC=00F8AEEC

(this interested me because I am going to implement A1200/A4000-compatible IDE emulation soon..)

08 July 2006, 23:47 #5

bluea

Posts: n/a

Thank you!!
This was really helpful. Unfortunately my tracer was a little buggy. So I was not able to catch the accesses at 0xde1000.

The "Gayle-check" is: (hope its helpful for Toni)

write 00h to 0xde1000
read byte 0xde1000 with bit 7 set
read byte 0xde1000 with bit 7 set
read byte 0xde1000 with bit 7 cleard
read byte 0xde1000 with bit 7 set

Then the Kickstart accepts Gayle and accesses HD. Works even with 37.300 (A600).

trace looks like this:
p00fb6b80t00daa000 1w 80808080
p00fb7074t00de109a 2w bfffbfff
p00fb70b4t00de1000 1w 00000000
p00fb7044t00de1000 1r ffff0000
p00fb7044t00de1000 1r ffffc000
p00fb7044t00de1000 1r 7fff0000
p00fb7044t00de1000 1r ffffc000
p00fb70d4t00da2018 1w a0a0a0a0
p00fb70d8t00da2010 1r 34ff34ff
P00fbcf98T00da9000 1r 7fffffff
P00fbcfa4T00daa000 1r 00000000

(p=pc, t=target addr, 1=byte 2=word, r=read w=write, data transferred format: bbxxxxxx wwwwxxxx)

@ganralf:

do you have bit-assignments for the Gayle registers (0xda8000,0xda9000,0xdaa000,0xdab000)?

regards,
Thomas

09 July 2006, 18:07 #6

ganralf
Registered User

Join Date: May 2006
Location: Germany
Posts: 97

Good you asked, that gave me the chance to sort my scraps and clearify things. Almost everything is for PCMCIA. Just IRQ handling ist left for IDE. I guess the best bet for the rest would be to ignore writes a return 0 on reads.

0xda900 Gayle INTREQ:
0x80 IDE
0X02 IDE1ACK (Slave)
0x01 IDE0ACK (Master)

If a Interrupt (Level 2) occurs and it is caused by an IDE Device Gayle INTREQ IDE bit 7 is set. I'm not sure if, the corresponding IDExACK will be set. When done with interrupt handling these bits will be set to 0 by the device driver.

0xdaa000 Gayle INTENA:
0x80 IDE

Setting bit 7 of Gayle INTENA enables ATA Interrupts.

So your trace reads:
> p00fb6b80t00daa000 1w 80808080
Enable IDE Interrupts
.
.
.
> p00fb70d4t00da2018 1w a0a0a0a0
Select device 0
> p00fb70d8t00da2010 1r 34ff34ff
Device's Cylinder Low register is 0x34. (Should have been 0 after reset)
> P00fbcf98T00da9000 1r 7fffffff
> P00fbcfa4T00daa000 1r 00000000
We're here in card.resource's irq handler. Which checks for a PCMCIA Int.

As usual for Amiga adresses are not fully decoded. Kickstart uses the following adresses for IDE. The above mentioned adresses are used by Linux. If you want the project to be compatible with Linux you should implement a similiar incomplete decoding.

0xda2000 Data
0xda2004 Error | Feature
0xda2008 SectorCount
0xda200c SectorNumber
0xda2010 CylinderLow
0xda2014 CylinderHigh
0xda2018 Device/Head
0xda201c Status | Command
0xda3018 Control

0xde1000's MSB should actually be interpreted as a 8 bit serial shift register, which reads 0xd0. Fat Garys (A3000,A4000) have the very same mechanism at 0xde1002 this register is called GaryID (see: http://www.thule.no/haynie/research/...ocs/a3000p.pdf).

Hope that helps and good luck on your project.

« Previous Thread | Next Thread »

Thread Tools

Show Printable Version

Similar Threads

Thread	Thread Starter	Forum	Replies	Last Post
Gayle in A500+	bebek	Hardware mods	14	25 May 2019 01:36
Gayle datasheet	Toni Wilen	Coders. General	9	09 December 2017 21:00
Overheating Gayle	tygre	support.Hardware	9	16 May 2011 06:04
Using FPU registers?	oRBIT	Coders. General	16	26 April 2010 14:34
Gayle Datasheet: Thanks	Oli_HD	support.Hardware	2	06 March 2008 00:43

Posting Rules

You **may not** post new threads
You **may not** post replies
You **may not** post attachments
You **may not** edit your posts

BB code is **On**
Smilies are **On**
[IMG] code is **On**
HTML code is **Off**

Forum Rules

Forum Jump
support.Hardware Go

All times are GMT +2. The time now is 22:09.

-- EAB3 skin

Archive - Top