SixPak Ultra for XT/AT bus (ISA)

ECE 1895 Final Project

Description

The original IBM PC and PC/XT is an authentic open architecture (other than the BIOS) with a relatively simple bus retroactively labeled ISA (industry standard architecture) by Compaq. The simple design of the PC mainboard means the original XT has minimal onboard devices, and the users must add on their own UART, Display Adapters, Parallel Port, RTC, Hard Disk Drive controller, Floppy Controller, and so on. The 8 slots on my PC/XT have been completely filled with various adapters.

The one critical issue with my current XT setup is the lack of a potent graphics adapter. It currently has a CGA (Color Graphics Array) adapter built by Quadram corporation in 1983. For unknown reasons, they utilized completely orthogonal wiring for the adapter with the 16k graphics SRAM located quite far away from the bus. This could be the reason that although the SRAM has a fast enough access time (100 ns), the SRAM cannot be updated fast enough by the NEC V20 when my bus is running at 7.12Mhz (clock generated by the PC-SPRINT board). This generates a lot of garbage on the screen whenever the processor does any graphics-intensive tasks under turbo mode.

The inherent problems of CGA – 8x8 text mode character, only 80x25 text mode, non-square pixels, and ugly color palettes during graphics mode are also limiting the capabilities of the machine significantly. So, in this new project, **the core capability of the SixPak Ultra is a VGA adapter built upon the famous Trident TVGA9000i VGA controller for the 8-bit XT bus architecture.**

But it would be a great waste of a slot if the only capability of the card is a graphics adapter. ISA being a true no-BS system bus (like Woz's apple II bus), we can attach 20000 devices to it without it giving any complaints. The only limitation is the interrupt/DMA and memory address space of the host system, which can be fixed through a few DIP switches connected to the bus mux and minimal glue logic.

So, the additional 5 functions added to this card will be

1. **an updated Intel 16550 (from the old 8250) UART**, which now sports a FIFO buffer and support for high speed (115200+) baud rate.
2. **an Ethernet adapter** based on the Realtek RTL8019 controller.
3. **an Adlib-compatible sound card** based on the famous Yamaha YM3012 OPL2 chip
4. **a PC8477BV-based high-density floppy controller** for diskette operations higher than that old 360k DSDD drive.
5. **an RTC clock** based on the "explosive" DALLAS DS12885 clock chip

Yes, the name takes direct inspiration from the illustrious AST SixPak plus. But as AST has been gone for over 20 years now, I think we can at least recycle this famous trademark a little bit. The ample size of a full-height, full-length ISA card will give me a lot of room to fit all these logic on board. And my previous experiences with ISA (555-project 1), Analog-Digital mixed PCB design (VDM-22) will come in handy in this case.

Design Reference

1. The VGA portion of the SixPak Ultra will be an 8-bit bus adaptation of the OEM Trident TVGA9000 ATbus (16bit ISA) graphics adapter. This will involve the reduction of address bus from 24 bits to 20 bits and data bus from 16 to 8 bits. The TVGA9000i chipset is perfectly capable of handling such a change and the only thing I'll need to modify is some routines in the VGA BIOS, namely taking out DMA channel 6 as it doesn't exist on an XTbus.
2. The Ethernet adapter uses the recommended layout from the RTL8019 datasheet.
3. The OPL2 sound card design follows the many open-source Adlib compatible card design online. I'll have to implement an interrupt selection switch on the bus driver as I only have 1 free interrupt left (int 4).
4. The floppy controller follows IBM's design for the PC/AT, the successor of the PC/XT. Modification includes a reduction from 16 to 8 bit data bus (address bus not changed as only the least significant 3 bits were used). The UART design follows the same route.
5. The RTC clock is really simple, it's a copy of many open-source design online.

Required Resources

Everything other than the main ICs for the SixPak Ultra can be procured from DigiKey. The ICs like the VGA controller, Intel's PLCC 16550, RTL8019, YM3012, PC8477BV and DS12885 will have to be purchased from eBay due to their vintage status.

Every other IC including the bus driver, mux/demux and DRAM/ROM will be replaced with their modern counterparts. I've gone through the VGA, UART and Ethernet controller to make sure I can find the correct modern chip.

The bulk of the cost will be purchasing the ICs from eBay ($30-40). The rest will accumulate up to $100 for a total of around $150.

I will also have to acquire an EEPROM writer for my BIOS extension codes (mainly to support the high-density floppy drives). I believe SERC has some of these, but I'll have to check on Monday.

Project Justification

This project marks the end of my studies on the original IBM PC architecture (the original x86 machine). The design will include 8086 machine code modifications, digital/analog mixed PCB designs, bus-based I/O etc. My previous projects like last term's 555 based IBM Joystick Controller, the PC-SPRINT clock generation board and VDM22 all prepared me well for this "Large Scale Integration" project which will be the SixPak Ultra.

This board will also be extremely popular amongst the vintage computing community. I know many people who had to give up on a 5150 or a 5160 just because of they cannot find the proper I/O, display, networking and floppy controller. This board has all of them. With just one board, you can basically get a barebones 5150 up to speed with some of the most decked out 5160s.

Schedule

As of today (Oct. 30 2022), I have completed the VGA portion and the majority of the UART/Floppy controller schematic work. I estimate to be able to finish up the schematic by next weekend (Nov 5-6) and have a workable PCB after another week (Nov 12-13).

I'll be able to produce the final BOM once the schematic is complete, so by the time the PCB is complete most of the components (probably except the YM3012 as it only ships from China) will arrive. The YM3012 should then arrive when the PCB comes back from JLC, in time to final assembly.

The presentation will be quite simple. I'll plug the card in my IBM PC/XT, wait for the machine to POST. If no POST errors were found, we'd then have to check for IRQ/DMA conflicts. From my understanding of the 5160's interrupt scheme, both the IBM double density floppy controller and my high-density controller can share interrupt 6. If that doesn't work, I'll have to remove the original IBM floppy controller and in turn lose my ability to use the 360K 5.25 inch floppy drive ☹.

If that works, we'd then need to test the VGA adapter by going MODE CO80. This will direct COMMAND.COM output to my VGA monitor (a glorious IBM P76). We can then test stuff like AutoCAD, prince of Persia, Windows 3.0, Digital Research GEM and a whole lot more.

Ethernet is relatively easy to test. I'll have to disable my Token Ring driver from loading by doing a little autoexec and config.sys menu thingy but then we should be able to load MTCP with RTL8019's packet driver. I obtained some extensive DOS networking experience mainly from the Token Ring paper implementation last term, so this should go without any hassle.

For UART, we'll load up Kermit and do some raspberry pi telnet goodness. With a high baud rate, we should finally be able to browse graphics BBS without hiccups.

The RTC would use AST's SixPak Plus driver -ASTCLOCK.COM. My design is compatible with AST's design so it should just set/configure the clock chip with ease.

For the Adlib card, David Murray's Planet X3 has some great adlib support. Games like the Ultima series and lemmings also have some good Adlib music tracks. We'll play those games in the end.

Potential Problems

I don't really see anything that can go wrong with this. With modern manufacturing capabilities and good PCB design, the card should operate with no problem under a 7.12 MHz bus clock. My worries are being unable to clean the ground much due to the aging PSU in my 5160, the adlib card is going to be quite noisy as tested by a fully functional ESS audio drive ISA card last year (had to unplug it as I ran out of slots, now you know why I'm motivated to make this card).

The possibility of nothing works is very small in this case, as all 6 functions basically just share the bus without any interconnections (except for power/gnd rails). At least one function will work, in the worst-case scenario. I should be able to identify the problem due to the highly integrated nature of the design, and it can be fixed in the next revision.