11 September 2012

Stack-180

Assembly and User Manual

Copyright 2011, 2012 TG Consulting

Table of Contents

Table of Contents (cont)

Stack-180 Introduction

The Stack-180 system was designed as a high speed CP/M-based system that can make use of some more recent hardware that has never been considered in a CP/M system before. Not only does this include multiple IDE-based disk drives, but ZIP and CDROM drives, too. A fully-implemented Stack-180 Mainboard also includes the USBWiz and the XPORT AR.

The USBWiz provides a USB host with two USB connectors capable of driving USB-based thumb drives, USB card reader, USB printers, and more. The USBWiz also carries an SD Card slot. The SD Card or USB Thumb Drive must be formatted with the FAT file system (Windows) prior to use with the USBWiz, giving the Stack-180 direct read/write access to the FAT file system. (FAT12, FAT16, or FAT32 can be used.) (The USBWiz is replaced by the ALFAT USB on Rev G0 and above mainboards. The ALFAT USB does not have the SD Card slot.)

The XPORT AR provides a 10/100BaseT connection to the Local Area Network (LAN) and is capable of operating at 100Mbps in full duplex mode. Currently, the XPORT and CP/M software provides two separate LAN-based links: a remote terminal connection and a remote printer connection, both links via the LAN. Other uses are being explored.

The CPU board provides a high speed, general purpose CPU with 32K EEPROM and 1MB RAM. Although intended as the basis for the Stack-180 CP/M system, the CPU board can also be used as a generic processing module. It mounts via two rows of 30 pins each, arranged on a .1" grid. All address, data, control, and power connections are made through these pins.

The Dual GIDE board carries the mount point for the CPU board, provides two independent IDE channels (for up to four devices,) and buffers the signals to/from the CPU board. The D-GIDE accepts +5VDC power through a 3.5" drive power connector, and passes it to the CPU board. Finally, the D-GIDE also provides a further connection to the mainboard by way of 34-pin and 26-pin connectors. The D-GIDE can mount directly to the mainboard, or ribbon cables can be used to connect the two.

The Stack-180 Mainboard provides access to the I/O subsystem. It includes a Real Time Clock, Quad UART, 16 interrupt levels, Slot addressing control ports, and general control and status ports. It also provides access to the USBWIZ and XPORT devices, as well as the four expansion slots.

With mainboard Rev G0 and above, the USBWiz OEM has been replaced by the ALFAT USB. The units are similar: both provide two USB Host ports that can operate on FAT Thumb Drive storage; both can host USB mice, keyboards, printers, and more. The ALFAT USB lacks the SD Card slot found on the USBWiz OEM. On the other hand, the ALFAT USB is cheaper, can run at the full USB 2.0 transfer speed spec and supports long filenames.

Stack-180 CPU Assembly

Assembly

Before beginning the assembly, examine the PCB solder points for any sign of oxidation. Basically, each solder point should be clean and shiny. If oxidation has begun to form, it will inhibit the soldering process and may lead to poor solder joints. To address this, you should clean the PCB prior to soldering the first component. Any number of metal polishes and electronic cleaners may suffice, but you should be careful that they do not disturb the solder resist and silk-screen legends. Be sure to clean all residue from the PCB; pay particular attention to the thru holes.

Assembly is straightforward and there are no particular difficulties that you should be aware of during assembly. Refer to the parts list and install the following:

- Install sockets for U2 through U10, leaving U1 for later since it's taller than many of the other components. The SMD RAM chips at U3 and U4 are pre-installed and are not socketed.

- Install .1ufd monolithic caps at C1 through C10, and the two 22pf caps at C20 & C21.

- Install resistors for R1-R8.

- Install the 10K resistor pack at RN1. Pin 1 on the resistor pack has a stripe or a dot; pin one of the socket is square.

- Install the 68-pin PLCC socket for U1. Pin 1 is at the center of the top row of pins, and there is an arrow inside the body of the socket pointing to it. Also note the notch on the top left corner of the PCB outline as well as the socket. Ensure the correct pin 1 alignment before soldering! The socket pins are not keyed to the PCB, so there are four ways to install it – three of them are wrong! (The CPU IC is keyed, so there is only one way to install it in the socket.)

- Install the four tantalum capacitors at C91 through C94. Pay attention to the '+' marking on the capacitors – it points towards U9. (You may receive 1ufd electrolytic capacitors instead. In this case the '-' lead is identified, and it points AWAY from U9.)

- Install the four 47ufd power supply filter caps at C101-C103. The '+' lead goes in the square hole. The Stack-180 has an optional Memory Expansion board that allows an additional 2MB of RAM to be attached. If you ever intend to do so, be sure to lay C101 and C104 on their sides - room between the CPU and Memory Expansion boards is limited.

- Install the two LEDs – GREEN at D1 for Power, and RED at D2 for HALT. The LEDs have a flat edge on the base that denotes pin 1 – it goes in the square hole (with the flat side towards the CPU.)

- Install the various header/jumpers. Do not install J1 & J2 yet. Place 2-pin jumpers at J5, J6, J9, and J10, 3-pin jumper at J3, and the ten-pin header at J4 for Terminal I/O. (J7 and J8 are not used.)

- Install the crystal at Y1. It can be either 18.432MHz or 29.4912MHz. There are two sets of holes for the crystal; use whichever set best fits the crystal leads. To prevent the underside of the 'can'-type crystal from shorting across the inner holes, leave a small space between the crystal and the PCB.

- Install the various ICs.

- To protect the pins, J1 & J2 may be installed later. When you're ready to install them, do so with the black plastic strip against the bottom of the CPU board. The MALE pins should be installed on the CPU board; the corresponding FEMALE sockets will be installed on the D-GIDE. For best fit

alignment, it is suggested that you assemble the two boards together via the J1/J2 connection, and then solder the pins to the CPU while the boards are still fit together. Be certain that the male pins are snug up against the underside of the CPU board before soldering. Then, turn the board pair over and solder the sockets to the D-GIDE board. Again, be certain the socket is pressed firmly against the D-GIDE board before soldering. This process will leave sufficient gap between the boards to ensure the components on the D-GIDE do not contact the underside of the CPU board. (Note that the capacitors on the D-GIDE board must be installed on their sides, with a 90 degree bend in the leads.)

CPU Board Headers and Jumpers

J1 and J2 are used for future expansion, providing all of the necessary CPU data, address, and control signals for further I/O decoding. Address lines A11-A19 are not provided, eliminating any possibility of memory expansion since the CPU tops out at 1MB and all of it is available on the CPU board. (Future memory expansion, beyond 1MB, is still possible via a daughterboard mounted directly to the CPU board.) Additionally, J1 and J2 carry the power and ground inputs to the PCB.

The jumpers and headers at several locations on the CPU board are used to determine the various operating parameters of the CPU.

J3 – 3-pin header. The normal position is Pins 2-3 shorted. This allows the CPU to write to the EEPROM to update the configuration block. Shorting pins 1-2 serves as a 'write protect' for the EEPROM. Initially, install a supplied shorting block on pins 1-2.

J4 – 10-pin RS-232 port for Terminal I/O via the Z8S180's internal ASCI-1 port. See discussion for J10 below.

J5 – 2-pin header. External RESET switch header. ATX case front panel switch.

J6 – 2-pin header. External power on LED header. ATX case front panel LED.

J7 & J8 – Not used.

J9 – 2-pin header. This jumper determines the state of Bit 6 when reading the CPU Control Port at 0F8H. The EEPROM uses this bit to determine the CPU clock speed for setting the ASCI baud rates and PRT reload values. If removed, values for a 29.4912MHz clock are used; if the jumper is installed, the CPU uses values for an 18.432MHz clock. 29.4912MHz is normal, so the shorting block should be removed.

J10 – 2-pin header. This jumper determines the state of Bit 5 when reading the CPU Control Port at 0F8H. The EEPROM uses this bit to determine if the mainboard is attached. If the jumper is installed, no mainboard is present; removing the jumper tells the CPU/EEPROM that the mainboard is present.

If the jumper is installed, the CPU/EEPROM will only use the on-board RS-232 port for Terminal I/O.

If the jumper is removed – indicating the mainboard is installed – the CPU will use the setting of SW1 (on the mainboard,) position 1 for Terminal I/O destination.

If SW1-1 is OFF, Terminal I/O will use the CPU board's RS-232 port via ASCI-1.

If SW1-1 is ON, Terminal I/O will be conducted via the mainboard's XPORT AR LAN adapter.

Stack Dual GIDE Assembly

Discussion

Before beginning the assembly, examine the PCB solder points for any sign of oxidation. Basically, each solder point should be clean and shiny. If oxidation has begun to form, it will inhibit the soldering process and may lead to poor solder joints. To address this, you should clean the PCB prior to soldering the first component. Any number of metal polishes and electronic cleaners may suffice, but you should be careful that they do not disturb the solder resist and silk-screen legends. Be sure to clean all residue from the PCB; pay particular attention to the thru holes.

The 40-pin right angle IDE connector headers at J3 & J4 can be configured in one of two ways: you can clip off pin 20 prior to soldering, or you can leave pin 20 in place.

With pin 20 removed, the IDE subsystem can use higher quality, 80-conductor IDE cables, either flat or round, for connecting the various IDE devices.

With pin 20 in place, J9 (Primary) and J10 (Secondary) can supply +5vdc to pin 20, providing a CF-IDE adapter with power via the IDE cable pin 20, if the adapter is designed to accept such a connection. However, doing so will eliminate the possibility of using the higher density, 80-conductor ribbon (or round) IDE cables, as well as some 40-conductor cables. (On virtually all 80-conductor - and most 40-conductor - IDE cables, pin 20 has a 'plug' in place to serve as a key, preventing the cable from being inserted incorrectly, requiring that pin 20 be clipped off of the J3 and/or J4 connectors.

Removing pin 20 will cause no problems with the IDE subsystem - it is not used in any way. Clipping the pin AFTER installation is difficult; replacing the pin after clipping is nearly impossible. We recommend that you clip this pin; all 80-conductor IDE cables – and some 40-conductor cables – have this pin blocked. After doing so, any future use of CF-IDE adapters will require that you provide a power connection directly to the adapter.

On Rev D (and above) PCBs, J9 & J10 are used to route +5vdc to IDE pin 20. On PCBs prior to Rev D, J9 & J10 are not present. You can accomplish the same thing by soldering a wire from any available +5vdc point (pin 24 on any 24-pin IC) to pin 20 on either (or both) of J3 & J4.

Assembly

Assembly is straightforward and there are no particular difficulties that you should be aware of during assembly. The 'Top' side of the PCB has the lettering. The following build sequence is recommended:

- Install sockets for U1 through U7.

- Install .1ufd monolithic caps at C1 through C7.

- Install the 10K resistor pack at RN1. Pin 1 on the resistor pack has a stripe or a dot; pin one of the socket is square.

- Install resistors for R1-R3. Pay attention to the values.

- Install the four 47ufd power supply filter caps at C8-C11. The '+' lead goes in the square hole. The capacitors will normally be placed on their side with the leads bent at 90 degrees. If you place the caps upright, there will need to be increased spacing between the D-GIDE and the CPU boards. This may be acceptable, but it is recommended that you minimize the spacing between boards.

- Install the 2x2 right angle header at J8 for external IDE activity LEDs.

- (PCB Rev D and above) Install the two 2-pin shorting headers at J9 & J10.

- After reviewing the pin 20 discussion above, decide how to configure pin 20. If you are removing pin 20, do so prior to installing the two 40-pin, right angle headers at J3 & J4. Install the headers.

- Install the two input pin headers at J6 (2x17-pin) and J7 (2x13-pin). These headers mount to the bottom of the D-GIDE board, soldered from the top, and will provide the motherboard will all needed Address, Data, and Control signals. The associated socket headers are included with the mainboard kit; if you didn't order the mainboard, they will be included with the D-GIDE in case you wish to build your own motherboard. All holes within and between the two sockets are centered on .1" spacing.

- Install the power input socket at J5. This socket provides power to both the D-GIDE and the CPU, and is not needed if you included the mainboard. When mounted on the mainboard, the D-GIDE and CPU receive power via the mainboard, so connecting a power cable to J5 is unnecessary.

- Install the two 30-pin SIP header sockets at J1 & J2. The header sockets go UP (soldered from the bottom.) Once the two SIP socket headers are in place, BUT NOT SOLDERED, you can insert the 30-pin headers in the bottom of the CPU board (the plastic carrier mates against the bottom of the CPU board.) Mate the two boards together, ensure the pin header is firmly against the underside of the CPU board, and solder the pins to the CPU board from the top. While ensuring that the SIP socket header on the D-GIDE is firmly against the PCB, solder the SIP socket headers to the D-GIDE from the bottom of the D-GIDE PCB.

- Install the various ICs. Pay particular attention to the installation of U5 & U6 – these are physically identical 24-pin GAL chips and it is easy to mix them up. U6 is labeled D-GIDE-1; U5 is labeled D-GIDE-2.

Jumpers and Headers

J1/J2 - Two rows of 1x30-pin connectors used to pass address, data, and control signals between the CPU and D-GIDE.

J3 – Primary IDE connector. 2x20 right angle connector.

J4 – Secondary IDE connector. 2x20 right angle connector.

J5 – Power connector. Used to supply +5vdc and ground to the D-GIDE and CPU boards when the mainboard is not being used.

J6 – 2x17 header mounted to the underside of the D-GIDE to pass address, data, and control signals to/from the mainboard.

J7 – 2x13 header mounted to the underside of the D-GIDE to pass address, data, and control signals to/from the mainboard.

J8 – 2x2 right angle header for external drive activity LEDs. The two rightmost pins are for the Primary adapter; the leftmost pins are for the Secondary adapter.

On Rev D (and above) PCBs:

J9 - 1x2 shorting header for supplying +5vdc to pin 20 on the Primary IDE header @ J3.

J10 - 1x2 shorting header for supplying +5vdc to pin 20 on the Secondary IDE header @ J4.

Stack Mainboard Assembly

Pre-Assembly

The Stack-180 mainboard was designed to install in any ATX or Micro ATX case, and to use an ATX power supply. It must be noted that some 'ATX' power supplies (old ones – generally pre-2000) do not follow the requirements used in the Stack-180 design, particularly as it applies to the Soft-On/Soft-Off feature. The sticking point we've seen with prototype systems has to do with the Pwr_OK signal on Pin 8 of the ATX 20-pin power connector. As long as this signal is driven to +5VDC as required by the ATX spec, Soft-On/Soft-Off should work as expected.

In general, it seems safe to say that, if your power supply has the 12V power connector required by Pentium 4 mainboards (4- or 8-pin, yellow and black wires), it probably matches the ATX12V spec, and will probably meet the requirements of the Stack-180. Another way is, if the mainboard power connector has the 24-pin option, you're probably OK (but this is less certain than the 12v power connector.)

One note of caution – proprietary power supplies may be referred to as ATX power supplies, but might not conform to the requirements for the 20-pin power connector. The 'In Case of Trouble' section details the 20-pin power connector requirements. If your power supply does not meet this standard, you MUST NOT USE IT or your mainboard may be destroyed.

There are simply too many power supplies available to be certain that yours will work as intended. Most of them meet the design requirements; some old power supplies (pre-2000) will not. The 'Initial Testing' section of this document will describe the checks needed to verify your power supply.

Mainboards Rev G0 and above have a modification to the soft-on circuit that eliminates the previous issues with the Pwr_OK signal. This means that virtually all ATX power supplies should work well with the Stack Rev G0 mainboard. For mainboards prior to Rev G0, a simple wiring change can be effected that will incorporate the revision in the G0 mainboard. Contact TG Consulting for details.

Assembly

Before beginning the assembly, examine the PCB solder points for any sign of oxidation. Basically, each solder point should be clean and shiny. If oxidation has begun to form, it will inhibit the soldering process and may lead to poor solder joints. To address this, you should clean the PCB prior to soldering the first component. Any number of metal polishes and electronic cleaners may suffice, but you should be careful that they do not disturb the solder resist and silk-screen legends. Be sure to clean all residue from the PCB; pay particular attention to the thru holes.

Assembly is straightforward and there are no particular difficulties that you should be aware of during assembly. Note that MOST ICs are installed vertically, and pin 1 is always towards the top of the board. Several ICs are installed sideways, so pin 1 is to the right side of the board. This is noted in the instructions. Also, note that the socket for SW1 is installed with Pin 1 to the bottom left as explained below.

Board orientation: Silk screen legends face up, with the 1" circular battery holder in the upper right corner. This is the 'normal' upright orientation. IN ALL CASES, the pad for Pin 1 is SQUARE.

Note that each IC socket is 'keyed' with a notch on one end. The notch indicates Pin 1 to the left of the notch. When installing multiple sockets, it is much easier if you use masking tape to hold the socket on the board so you can turn the board over and solder.

The following build sequence is recommended:

- Install sockets for the "upright" ICs (Pin 1 at the top left) at:

8 pin: K1
14 Pin: U9, U16, U21, U24
16 Pin: U22, U23
18 Pin: U26
20 Pin: U1, U2, U3, U5, U6, U15, U25
24 Pin: U8, U10

- Install sockets for the "sideways" ICs (Pin 1 at the top right) at:

(NOTE: The 8-pin socket for SW-1 is installed 'backwards' from the normal sideways mounted sockets. In this case, Pin 1 is at the bottom left. Note the notch silk-screened on the mainboard and the notch on the socket. You might consider eliminating the socket and soldering the DIP switch directly to the mainboard, as the DIP switch has a tendency to come loose if you're not very careful when moving the switches.)

8 Pin: SW1
14 Pin: U27
20 Pin: U17, U18, U19, U20
24 Pin: U7, U11

- Optional hardware:
XPORT:
14 pin: U29

HEX LED:
14 Pin: U13, U14
20 Pin: U12

At this point, all sockets have been placed except for U4, the 68-pin PLCC socket (it will be installed later.) Before continuing, go back and verify that every pin of every socket has been soldered.

- Install .1ufd monolithic bypass capacitors at each of the IC sockets previously installed (except for U13 and U14 – these are optional LED sockets - and the 8-pin sockets at K1 and SW1.) The capacitors are placed at the Pin 1 end of each socket. The capacitors connect directly to the internal power and ground planes, so soldering requires a little more heat and time.

If optional hardware is not included, DO NOT install a .1ufd capacitor for that part: C4 for USBWiz; C29 for XPORT; C12 for HEX LED.

NOTE: The RS-232 level shifters at U5 and U6 each have three .1ufd caps at C5/6, C51/61, and C52/62.

- C41/42/43/44 are .1ufd caps surrounding the 68 pin PLCC socket. Install these four caps.

- Install the six 330pf capacitors near the U5 and U6 level shifters at C53/54/55 and C63/64/65.

- Install the following resistor packs: Pin 1 of each pack is marked with a stripe or a dot. As always, Pin 1 of the header has a square pad.

RN1 – 10K ohm, 6-pin (6A103G)
RN2 – 220 ohm, 8-pin (8A221G)
RN3 – 220 ohm, 6-pin (6A221G)
RN4 – 10K ohm, 6-pin (6A103G)
RN5 – 220 ohm, 6-pin (6A221G)

- Install individual resistors as follows:

R1 – 330 Ohm (Orange, Orange, Brown)
R2 – 10K Ohm (Brown, Black, Orange)
R3 – 10K Ohm (Brown, Black, Orange)
R4 – 330 Ohm (Orange, Orange, Brown)
R5 – 10K Ohm (Brown, Black, Orange)
R6 – 47K Ohm (Yellow, Purple, Orange)
R7 – 47K Ohm (Yellow, Purple, Orange)
R8 – 47K Ohm (Yellow, Purple, Orange)
R9 – 1K Ohm (Brown, Black, Red)
R10 – 1K Ohm (Brown, Black, Red)

Optional parts include the following resistors: 220 Ohm (Red, Red, Brown) at R11 for XPORT.

- Install the battery holder at B1. Note that the '+' lead (with the extrusion) is towards the left. The battery holder leads can easily be 'cold soldered' so be certain to use enough heat for a good connection. Note that some battery holders with Rev F2 mainboards will have spacing difficulties with RN1 and hole size problems with the battery holder pins. This is due to our parts supplier changing manufacturers for the battery holder. The pins on the newer battery holders are offset to the side somewhat, causing the battery holder to intrude on RN1. To correct this, the plastic battery holder will have to be 'notched' to allow RN1 to intrude into the battery holder. Also, the pins are too wide for the mainboard holes, so they will have to be filed to fit. The pin spacing is good - the pin themselves are too wide to fit in the holes.

- Install two 1N4148 diodes at D3 and D4. In both cases, the banded end of the diode goes to the left, towards U25.

- Install three 1N4148 diodes at D7, D8, and D9. In all cases, the banded end goes to the right, towards J3.

- Install three 1N751A 5.1V Zener diodes at D2, D5, and D6. D2 and D6 have the banded end to the right; D5 has the banded end towards the top of the mainboard.

- Install the 68-pin PLCC socket at U4. It is VERY EASY to install this socket incorrectly, as it is not keyed to the PCB. In other words, it will install four different ways, and three of them are wrong. The PCB silkscreen outline has a notch in the upper left corner that corresponds to the same notch on the socket. Also, there is an arrow on the inner face of the socket; it points towards Pin 1. As usual, pin 1 on the PCB has a square pad. Ensure the correct alignment!

- Install the following jumpers and headers:

2x17 header (socket) at J1
2x13 header (socket) at J2
1x2 jumper (pins) at J6 (External power switch)
1x3 jumper (pins) at J7 (XPORT Power-on-LAN)
1x3 jumper (pins) at J11 (power supply soft-on)
1x2 jumper (pins) at J12 (RTC Battery backup)
1x3 jumper (pins) at J13 (PWR_OK Bypass)
1x2 jumper (pins) at J15 (/INT0 interrupt chain enable)
2x30 header (socket) at Slot 1
2x30 header (socket) at Slot 2

NOTE: Do not install the optional headers at: J3 for USBWiz or J16 & J17 for XPORT. They will be installed later if the option kit is included.

- Install three, 3-pin fan power headers at J9, J10, and J14. Note that the locking tab goes towards the edge of the PCB.

- Install eight 100mfd electrolytic capacitors at C100-C107. The POSITIVE lead goes in the square pad. NOTE: There is not enough room under the D-GIDE board for capacitor C104 to stand up straight. Instead, bend the leads 90 degrees and lay the cap flat on the mainboard. C105 and C106, also under the D-GIDE, do not have this problem, and can be mounted upright IF they are mounted firmly against the mainboard.

- Install the green power-on LED at D1. The flat side of the LED goes towards the bottom of the board.

- Install the yellow standby power-on LED at D10. The flat side of the LED goes towards the bottom of the board.

- Prior to installing the 20-pin power connector at J4, you must cut off the tabs on each end. The 'key' on the socket is a small ridge on one end, denoting pin 1. As usual, pin 1 on the PCB has a square pad (towards the right side of the board.) The latching tab goes towards the top of the board edge. This connector will install upside down if you do it wrong. Be careful!

- On mainboards prior to Rev G0, install the stacked DB-9 connector at J5. The mounting tabs do not provide enough holding force to keep the riser stiff against the mainboard, so you should solder the tabs in the holes provided. This will require a good bit of heat. Be careful that you don't damage the mainboard.

- On the Rev G0 mainboard, install the FEMALE DB-9 connector at J5 (the uppermost position.) Install the MALE DB-9 connector at J18 (the lower of the two positions.)

NOTE: Do not install the two 96-pin AT-style connectors at Slot3 and Slot4 until power checks are complete.

Optional Hardware Installation

- Perform the following if optional hardware kits are included:

ALFAT USB:

- Loosely install the two 4-40 posts on the top of the mainboard with ¼" 4-40 screws inserted from the bottom of the board. Don't tighten them. Insert, but do not solder the 1x18 socket header at J3. Insert the 1x18 pin header into the bottom of the ALFAT USB, but do not solder. With two more ¼" 4-40 machine screws, install the ALFAT USB daughterboard on the mainboard at J3. While holding the ALFAT USB aligned to the mainboard, tighten all four machine screws.

- Ensure the mainboard socket header at J3 is pressed firmly against the mainboard, and solder the header to the mainboard.

- In a like manner, ensure the 1x18 pin header is pressed firmly against the bottom of the ALFAT USB, and solder the header to the ALFAT USB from the top.

NOTE: You may want to use blue locktite (242 or equivalent) on the two machine screws on the bottom of the mainboard. This will ensure that future removal of the ALFAT USB does not disturb the alignment of the posts.

- Remove the ALFAT USB daughterboard from the mainboard until power supply testing is complete.

XPORT:

The XPORT kit consists of two parts: 1) the XPORT itself, and 2) the 3.3v power supply.

NOTE: The ATX power supply will always have +5v power present if it is plugged in and the rear panel power switch is on, even though main power is off. To ensure power is completely off, unplug the main AC power cord.

Rather than use ATX 3.3v power, the XPORT uses +5v Standby (+5vSB) power to enable the 'Wake on LAN' feature. This will allow the XPORT to power up the Stack 180 when your Telnet client connects to the XPORT. To do so, however, the XPORT must have power applied, even when power is off. This requires the use of +5vSB supplied by the ATX power supply. Since the XPORT runs on 3.3v, a voltage regulator is needed to develop 3.3vSB from the ATX +5vSB.

U28 is a 5v to 3.3 volt Low Drop Out voltage regulator, rated at 1 amp. It supplies 3.3vSB power to the XPORT as well as the Slot1 and Slot2 expansion slots. C108 and C109 are 47uf electrolytic capacitors, used to filter the input and output to/from the regulator.

The XPORT AR is shipped from the manufacturer without the Stack daughterboard. If you purchased the XPORT from a source other than TG Consulting, you will still require the mounting and parts kit, and you'll have to solder the XPORT to the daughterboard.

Install and solder the following:

- Install the two 47ufd electrolytic caps at C108 & C109. There is sufficient room under the USBWiz for these caps to stand upright.

- Test fit the U28 regulator to the mainboard and note where the three pins need to be bent 90 degrees. Bend the pins and check the fit. Adjust as necessary. Apply a small dab of heat sink compound (not included) to the back side of the 3.3v regulator where it mounts to the mainboard. Use a 3/8" 4-40 screw and nut to mount U28 to the mainboard. Insert the screw from the bottom of

the mainboard. When properly aligned, tighten the nut and solder U28 to the mainboard. You may want to use a drop of blue Locktite on the screw/nut as this part is non-removable.

Loosely install the two 4-40 posts on the top of the mainboard with 4-40 screws inserted from the bottom of the board. Don't tighten them. Insert, but do not solder the two socket headers at J16 and J17. With two more 4-40 machine screws, install the XPORT daughterboard on the mainboard at J16 and J17. While holding the XPORT aligned to the mainboard, tighten all four machine screws.

Ensure the two socket headers at J16 and J17 are pressed firmly against the mainboard, and solder both headers to the mainboard.

You may want to use Locktite on the two machine screws on the bottom of the mainboard. This will ensure that future removal of the XPORT does not disturb the alignment of the posts.

Remove the XPORT daughterboard from the mainboard. Do not re-install the XPORT until all mainboard initial testing is complete.

HEX LED:

All necessary components are already installed. Do not install the LEDs until initial power checks are complete.

-5VDC Option

All PC-AT, and some early ATX power supplies provided -5VDC, but it has been removed from the design standard and will not be present in current power supplies. If you have a need for -5VDC at the expansion connectors, the Stack mainboard has provisions for making it available.

For mainboard revisions prior to G0, the mainboard could use the ATX-provided -5vdc, if available. If not, the mainboard provided space for a regulator to create -5vdc from the -12vdc supply. Rev G0 and later mainboards assume the ATX power supply does not provide -5dc; the mainboard creates -5vdc from the -12vdc supply and there are no provisions for using any -5vdc provided by the power supply.

On mainboards prior to Rev G0, check the J4 power supply cable connection - if there is a white wire on pin 18, your power supply probably provides -5vdc. Check with a voltmeter. If present, you should install the '-5' jumper on the mainboard. The jumper is located under the USBWiz, slightly above and to the left of capacitor C108. This connects the power supply -5vdc output to the slots - nothing further needs to be done.

If you don't have -5VDC on J4, pin 18, do not install the '-5' jumper. You'll need to install the -5VDC regulator and two capacitors instead.

For Rev G0 and later mainboards, the "-5" jumper does not exist and you cannot use -5vdc from the power supply, even if the power supply provides it. Instead, the only option is to use -12vdc with the 7905 -5vdc regulator.

NOTE: Rev F2 and lower mainboards have a design error in the -5vdc circuit (corrected in Rev F3 and later PCBs.) The heat sink mount pad on the mainboard is grounded, but the 7905 -5vdc regulator has the mounting tab connected to the input circuit (-12vdc.) If you mount the regulator to the mainboard, you will short -12vdc directly to ground. Instead, use an insulator between the mainboard and the 7905 mounting tab, and DO NOT use a metal nut/bolt to mount the tab to the mainboard. Nylon would be acceptable.

The -5vdc option provides -5vdc to the expansion connectors if your ATX power supply does not already provide it. More than likely, you will never see a need for having -5vdc.

Installing the -5vdc Regulator Option:

On Rev F3 and earlier mainboards where the power supply does not provide -5vdc, and on all G0 and later mainboards, install the following:

- Install the two 47ufd electrolytic capacitors at C110 & C111.

If using a Rev F2 or earlier mainboard, refer to the NOTE above. DO NOT MOUNT THE REGULATOR DIRECTLY TO THE MAINBOARD!

If using a Rev F3 or later mainboard, test fit the U30 7905 regulator to the mainboard and note where the three pins need to be bent 90 degrees. Bend the pins and check the fit. Adjust as necessary. Apply a small dab of heat sink compound (not included) to the back side of the 7905 regulator where it mounts to the mainboard. Use a 3/8" 4-40 screw and nut to mount U30 to the mainboard. Insert the screw from the bottom of the mainboard. When properly aligned, tighten the nut and solder U30 to the mainboard. You may want to use a drop of blue locktite on the screw/nut as this part is non-removable.

Installing the Mainboard in the ATX Case

The mainboard was designed to fit in a standard ATX or Micro ATX case. It will also fit in an AT-style case, but the power connector will need to be replaced with an ATX-style 20-pin housing, and there may be some problems with the seven standoff mount locations.

Your case should have included ¼" standoffs on which to mount the mainboard, lifting it above the side of the case. This is the proper spacing to use so that any installed PC-Style adapters will properly fit the mounting brackets. Alternately, the hardware kit includes a set of SAE standard 6-32 standoffs, screws, and nuts. If it is necessary to use these, you will likely have to install the nut on the back side of the mainboard mount panel to obtain a secure fit. Use a drop of blue Locktite on the nut.

Remove the 'I/O Riser Bracket' from the I/O opening at the rear of the case. The mainboard external connections will not align properly with the bracket.

There are seven mount points – each should be supported with a ¼" standoff. Install the standoffs, lay the mainboard in the case, and install the seven retaining screws. Don't tighten any screws until all seven are installed.

Connect the 20-pin power cable to J4 and the front panel power switch to J6. If any case-mounted fans are present, connect them to J9, J10, and/or J14, the Fan1, Fan2, and Fan3 connectors.

Initial Mainboard Testing – Power Checks

Part of the initial power-up testing is to verify that your ATX power supply works as required by the Stack. We have found some old-model 'ATX' power supplies that do not work as required by the Stack or the ATX power supply specification, so pay particular attention to these checks before you install any ICs. Damage to the mainboard components could result!

First, check the ATX 20-pin power connector. If there is a white wire on Pin 18 (-5VDC) you may have an early ATX power supply that might have problems with the Stack. If the white wire is missing (nothing connected to Pin 18) then your power supply is probably OK.

Initial testing consists of voltage checks without having installed any ICs. This is merely to ensure that there are no shorts across the various power supplies and that the appropriate voltages are not driven too high. To accomplish this test, install the following:

- Shorting block on J11, pins 2-3 (J11 is below and to the left of the RTC battery.)
- Shorting block on J7, pins 2-3 (J7 is under the XPORT mount position.)
- Shorting block on J15 in the center of the mainboard.
- K1 Relay (LBA110.)

Temporarily install the mainboard in your ATX case, connect the 20-pin power cable, and plug in the power supply. Turn on the rear panel power switch. Main power should immediately come up.

NOTE: An ATX switching power supply requires a load in order to 'switch on'. You may need to connect a hard drive to provide sufficient load. Also, ensure the K1 relay (LBA110) is installed correctly.

Ground the negative lead on your multimeter, turn on the power supply via the rear switch (if present) and, with the positive multimeter lead, check for the +5vSB voltage.

- On U28, Pin 1 you should read +5vDC. The same voltage should be present on U26, Pin 18. This

is the 5VSB (standby) power from the ATX power supply, so the yellow LED (Stby Pwr) should be lit.

If the 5vSB check is bad, STOP! Unplug everything, remove the mainboard, and re-check everything you've done. If all else fails, refer to the 'In Case of Trouble' section to contact TG Consulting.

- On U28 Pin 3 and J16 Pin 3, you should read +3.3VDC. This is the 3.3V Standby power created by U28. If you have not installed the XPORT kit, you will not have 3.3VSB.

Shut down main power at the rear panel power switch. Shift the J11 shorting block to position 1-2 to enable the mainboard's Soft-On function.

Via the rear panel power switch, turn on main power. The mainboard yellow LED should light, but not the green LED, indicating Standby power is applied but not main power.

Install a shorting block on J6. Main power should come up, lighting the green LED.

If main power does not power up, STOP, re-check everything, and try again. If all else fails, refer to the 'In Case of Trouble' section.

The final Soft-On test requires that you remove the shorting block on J6 – main power should stay ON. If the power supply drops back to standby, your ATX supply does not meet the Soft-On/Soft-Off requirements of the Stack-180 – you will have to operate with the J11 shorting block in position 2-3 and use the rear panel power switch for power on/off. This problem has been noted with some older ATX power supplies. Newer power supplies, ATX12V V2.0 and higher, are known to work well.

NOTE: We've seen one case where a bad K1 relay caused the power supply to drop off-line as described in the above paragraph. This was with an ATX 12V V2.03 power supply - 'known to work well' caused us to look a little closer to find the bad relay rather than assume the power supply required replacing.

With main power fully energized, read the following voltages:

- On U1, pin 20 (74ABT245) you should read +5vdc. (4.75 to 5.25 is OK)
- On U5, pin 10 (SN75185) you should read -12vdc. (-10 to -12.5 is OK)
- On U5, pin 1, you should read +12vdc. (+10 to +12.5 is OK)
- On Slot 1, pin 1, you should read +3.3vdc. (3.2 to 3.4 is OK)
- On the upper hole of the '-5' jumper (left side of mainboard, between U28 and C108), you may read -5vdc. Note that many newer ATX supplies do not provide this power – if the white wire is missing from Pin 18 of the 20-pin power connector, -5vdc will not be present. If the white wire IS present and you DO read -5vdc at the upper hole on the '-5' jumper, you can install a jumper wire between the two holes to pass -5vdc to the Slot connectors. If -5vdc is NOT present, do not install the jumper wire. Refer to '-5vdc Option' and 'In Case of Trouble' for more information concerning -5vdc power.

If there is a problem, find and repair it before continuing. If all else fails, refer to the 'In Case of Trouble' section.

Assuming all voltages are good, final assembly can be completed. Note that, at this point, you cannot turn off the Stack if operating in the Soft On/Off mode (J11 shorting block in position 1-2). The CPU is required to turn off main power – so you'll have to either switch off via the back panel switch on the power supply or simply unplug the unit. This is not an issue when the system is fully assembled and running normally.

Turn off and unplug the ATX power supply.

Final mainboard assembly consists of installing the two 96-pin AT-style card connectors at Slot 3 and Slot 4. Before doing so, you should install U18 and U27 between the Slot 3 and Slot 4 connectors. With the connectors in place, installing these ICs will be somewhat more difficult.

Final Mainboard Testing

- Install the various ICs, using the schematic and board layout diagrams for reference.

NOTE: For the HEX LED display option: Pay particular attention to the orientation of the TIL-311 HEX Display LEDs. You'll notice one end has a single notch and the opposite end has two notches. The end with TWO NOTCHES denotes Pin 1 on the left and pin 14 on the right. Installing them upside down will INSTANTLY destroy them. Remember – TWO NOTCHES AT THE TOP.

NOTE: SW1 is inserted such that the '1-2-3-4' label is across the bottom. Start with all switches OFF (down.)

- Before continuing, make a final check for ICs and HEX LEDs being installed correctly. The HEX LEDs have two notches at the top of the display. All ICs have pin 1 either towards the top of the board, or towards the right side.

- Remove the J12 jumper (if installed) and install the Real Time Clock battery. The battery '+' side faces UP.

- If the XPORT is being used, install it now.

- If the USBWiz is being used, install it now.

- Install the Dual GIDE board. In the same process you used to mount the USBWiz and XPORT, use 4-40 screws and ½" standoffs to mount the GIDE to the mainboard. After alignment, you may want to use a drop of Locktite on the bottom screws to ensure proper alignment is retained when removing the D-GIDE.

- Install the mainboard in the ATX case and connect the various headers: power, external power on, etc. Your ATX case should have come with ¼" standoffs for PCB mounting. Use these to mount the PCB in 7 places. If not, use the 6-32 hardware in your mainboard kit.

- Connect the IDE devices to the D-GIDE. Note that the LOWER IDE connector is the Primary GIDE and the UPPER IDE connector is the Secondary GIDE.

- Install the CPU board on top of the D-GIDE. Note that the boards should overlap precisely, and all three boards form a fairly straight edge along the right side of the mainboard. You CAN install the CPU board upside down on the D-GIDE, but it will be offset to the left.

- Connect the Terminal, external reset and front panel power on LED connectors to the CPU board.

- Ensure the terminal (or TELNET Client software) is set for the following parameters: 9600 Baud, Flow control disabled, and Enter (or RETURN) sends CR only (not CR/LF!) If you can, set your terminal for VT-100 (or higher) emulation. If you're using the LAN-based terminal via the XPORT, make sure you connect to Port 10001 for normal operation.

- With the front panel power switch connected to J6, power up the Stack. Your terminal should receive a CPU test and initialization screen, followed by the logon screen. If the system fails to power up, refer to 'In Case of Trouble.' If garbage is displayed, verify that your terminal is properly connected and set to 9600 baud, no flow control. Perform a hardware RESET on the Stack and, if garbage is still being displayed, immediately (within 3 seconds of the hard RESET) press any key on

the keyboard – this will force the CPU to use 9600 baud, no flow control (if not already there.) If you're still getting garbage, see 'Terminal Problems'. If nothing is displayed, refer to 'Dead System.'

- Set SW1-2 to ON. This will allow the system to perform a 'Full' initialization cycle, including any GIDE- or Slot-connected devices. See SW1 Bit Positions for details.

- Enter 'S' (Setup) and enable the various ports and controllers you have installed. When complete, be sure to save the EEPROM data block via the 'Write EEPROM and ReBoot' option.

- After the re-boot, enter the 'T' (Test) command. You'll see a menu similar to this:

```
Stack-180 Boot ROM Test Routines
 Available Testing Routines (29MHz):

        A - RAM Write/Read/Verify tests
        B - IDE Drive Testing
        C - Test PRT-0 50mSec Interrupt
        D - IDE Copy Routines
Mainboard Required:
        E - Serial Port Tests
        F - Display Switch Setting data
        G - LAN Test
        H - USBWiz Test
        I - Printer Test Page - Slot4 LPT
        J - Printer Test Page - Slot3 LPT
        K - Fan Speed Display


Select (ESC to Exit) ?
```

Select the various tests that are presented. Note that the serial tests require that the selected serial connectors be plugged in to each other. Some specific tests that should be accomplished:

A – RAM Read/Write/Verify Tests. Let this test run for at least one complete cycle. Press ESC to exit the test.

B – IDE Drive Testing. Select the drive to test via Option 1 (make sure it is installed, connected, and powered up) and select the R/W/V test via Option 3. The selected drive must have been previously enabled in Setup. This is a non-destructive test, so it will not modify any data already present on the drive. Allow it to run a few minutes, then press ESC to exit the test.

C - Test PRT-0 50mSec Interrupt. This test will verify that the CPU's internal timer and interrupt system is working properly. After selecting the test, the Stack should pause for ~2 seconds, report Test Complete, and await a key press to exit the test. If you never see the 'Test Complete' message, refer to 'In Case of Trouble.'

E - Serial Port Tests. This test verifies the serial port hardware, but can only be performed if you have a jumper cable connected to both serial connectors on the back panel. The cable must be a 9-pin Male to 9-pin Female, straight-through cable. Cross-over or 'Null Modem' cables will not work. Press ESC to exit the test.

G - LAN Test. XPORT only. Selecting this test will pass continuous strings of ASCII data to your LAN-based TELNET client if the XPORT is connected via the LAN. See the XPORT Setup and Test section later in this document. If you're already running via the LAN-based terminal, this test is unnecessary. See XPORT Setup for further details.

H - USBWiz Test. USBWiz only. Selecting this test allows you to issue command strings directly to the USBWiz. Only one check is necessary – the 'vr' command as shown here:

```
USBWiz Test - Command Line interface.  ESC to exit.
Cmd: vr
USBWiz 2.29
!00

Cmd:
```

The 'vr' above is the 'Version' command to the USBWiz – it responds with the firmware version currently installed in the USBWiz. See USBWiz Setup for further details.

K – Fan Speed Display. Shows the RPM of the three case fans. Press ESC to exit the test.

Mainboard Headers and Jumpers

Jumpers at several locations on the mainboard are used to determine the various operating parameters of the mainboard components. Refer to the mainboard layout diagram for the following:

J12 (top right, beside the battery.) When installed, the battery will maintain RTC time during a loss of all power. If the ATX power supply is plugged in and in the 'On' or 'Soft Off' condition, +5VSB supplies the RTC. Otherwise, the battery maintains RTC timing.

J15 (center mainboard.) Enables the 16-level interrupt system on the mainboard. If not installed, mainboard interrupts will not be passed to the CPU.

J13 and J8 (near the USBWiz connector.) These jumpers are set to determine the USBWiz I/O parameters via the MODE0 and MODE1 pins. For normal operation, you should install only the LOWER of the two jumpers (J8) to put the USBWiz in UART I/O mode. Refer to the USBWiz Manual, Section 6.1 for further information.

J9, J10, and J14 are 3-pin fan power connectors. If you wish, you can power internal case fans with these headers. All three fans can be monitored for rotational speed via the EEPROM Test menu.

J5 (top left corner) is a stacked DB-9 connector, male over female, that provides I/O for two RS-232 ports, COMA (DTE - top) and COMD (DCE - bottom).

J6 (right edge, middle) is connected to the front panel power switch.

J7 (middle of the mainboard, left) is a 3-pin jumper used to allow or prevent the XPORT from performing the Remote Power On function. With the shorting block in position 2-3, the XPORT will not be able to power up the Stack when a remote connection is established. In position 1-2, Remote Power On is enabled; main power will be energized when your Telnet client connects to the XPORT.

J11 (below and to the left of the RTC battery) is a 3-pin jumper that is supplied in case your power supply does not strictly conform to the ATX12V V2 standard. With the shorting block in position 1-2, ATX Soft On/Off is used to control the power supply. In position 2-3, the power supply will come up to full power any time the rear panel power switch is turned on. In this case, the position of the J7 shorting block will be immaterial, and the front panel power switch connected to J6 will have no effect.

SW1 Bit Positions

SW1 (top center) is a 4-position DIP switch. These four bits are read in as bits 0-3 from port 0F0H (SwitchPort) . The physical switch body labeling is somewhat confusing, since the switches are labeled 1-4 and they control bits 0-3 respectively.

Each bit has the following definition:

```
Switch
Position
1 (Bit 0)   OFF – Use RS-232 Terminal I/O via ASCI-1
            ON  - Use LAN-Based Terminal I/O via ASCI-0 and XPORT AR

2 (Bit 1)   OFF – Simple Initialization cycle
            ON  - Full Initialization cycle

3 (Bit 2)   OFF – Mainboard has USBWiz OEM installed
            ON  - ALFAT USB is installed

4 (Bit 3)   OFF – Normal Boot – No XPORT CLI Interface
            ON  - Start XPORT CLI on Reset
```

Notes:
The SW1-1 Terminal I/O destination is a software selection – it has no control over the hardware.

SW1-2 OFF forces the CPU to bypass any Slot or GIDE initialization functions, regardless of their Setup status. SW1-2 ON allows the CPU to perform Slot and GIDE initialization functions as determined by Setup. This is used as a troubleshooting method.

SW1-3 only applies to Rev G0 and later mainboards. Prior to Rev G0, SW1-3 was undefined.

SW1-4 ON will start the XPORT Command Line Interface during the power-on/reset initialization process. The document "Command Reference.PDF" in the CD's XPORT folder lists all CLI commands. This is one method by which you can set the XPORT's Internet Protocol (IP) address. Refer to section 'XPORT Setup' for directions and further information. Note that if you are running on the LAN-based terminal via the XPORT, setting SW1-4 to ON will do nothing - you cannot use the Line 1 CLI unless you're also running on a serial terminal. In this case, the Line 2 CLI can be used from the Setup menu, option H.

Setup

CPU Setup

CPU board hardware setup is fairly simple – there are few options, with only a single Terminal I/O port and three jumpers.

The Terminal I/O port is wired directly to the CPU's ASCI-1 UART. Under software control, it can run at virtually any baud rate, from 110 to 230,400; hardware flow control is established via CTS/RTS signaling, if enabled in EEPROM Setup.

Due to a Zilog design fault in the Z180 core, the CPU is unable to detect incoming CTS signaling on ASCI-1. The board design re-routes the incoming CTS signal through GpPort, bit 4, where Bit4 = 0 signals that CTS is active, while Bit4 = 1 signals that CTS is NOT active.

The 3-pin jumper at J3 is the EEPROM Write Enable jumper. When the shorting block is in position 1-2, the EEPROM is in Hardware Write Protect. With jumper J3 in position 2-3, Hardware Write Enable is in effect. In this case, the system depends on the EEPROM's internal Software Data Protect process to prevent unwanted writes to the EEPROM.

For initial testing, J3 should be placed in position 1-2 to prevent unwanted EEPROM writes. Once system normal operation is obvious, J3 should be shunted to position 2-3 to allow the operator to run EEPROM Setup and configure the software to match the system hardware. When Setup is complete, a Write to the EEPROM can be completed. If desired, you can then return J3 to the Write Protect position, shunting across pins 1-2.

J9 is used by the CPU and EEPROM software to determine the system clock rate. With the J9 shorting block removed, the system expects a 29.4912MHz clock rate; with the J9 shorting block installed, the system operates with an 18.432MHz clock.

J10 is used by the CPU and EEPROM code to determine if the mainboard is present. With the J10 shorting block removed, the system expects the mainboard to be present and will follow the Setup Data Block assignments for hardware initialization. With J10 installed, the system assumes the mainboard is absent, and will not attempt to initialize any mainboard hardware regardless of the Setup Data Block settings. This is one troubleshooting method that can be used to minimize the system init process and perhaps make troubleshooting a little easier.

Terminal Setup

The Serial Terminal is accessed through the CPU's internal ASCI-1 port. For the initial connection, the EEPROM is programmed to use 9600 baud, eight bit characters, no parity, and 1 stop bit (8N1.) Connector J4 on the CPU board and the terminal pigtail provide a DTE port for a 'null modem' connection to a DTE terminal or PC running a terminal/TELNET program. The supplied pigtail is of the 'straight thru' type; the cable should be of the 'Null Modem' type.

The Terminal (or terminal program) must be set for 9600 baud, 8N1, and flow control disabled. If possible, your terminal should be set for VT-100 emulation. Terminals (and software) vary, but there should be a method by which you can set the terminal's response when the Carriage Return (CR) key is pressed. Some terminals/software will transmit a CR/Line Feed pair when the Carriage Return key is pressed. You must set the terminal to send only a CR when the CR key is pressed.

Setup (cont)

Dual GIDE Setup

The Dual GIDE board is based on Tilmann Reh's original GIDE design. The original 16V8 and 20V8 GALs were replaced with two 22V10 GALs to provide two complete IDE control ports that operate completely independent from one another. The two IDE data buses are common between ports; the IDE control buses are separate.

J3 - the lower of the two IDE connectors – is referred to as the Primary IDE port. J4 – the upper IDE connector – is referred to as the Secondary IDE port. The ports are identical and can be used interchangeably, with the sole exception being the port's address range. The Primary port range is 50H-5FH; the Secondary port range is 40H-4FH.

There are two jumpers on the D-GIDE for external Drive Activity LEDs at J8, one for each GIDE port. These signals are voltage/current-limited by R2 and R3, so they can be connected directly to front panel LEDs to indicate drive activity.

The Stack OS can access either of the two ports with Master and/or Slave drives. The B/P BIOS can only use one port – it can be either of the two – and can operate on both Master and Slave drives on that port.

Mainboard Setup

The Mainboard is the heart of the I/O and interrupt subsystems. In a minimal system, it hosts the Quad UART, RTC, Interrupt system, and Slot I/O decoding. Optional features include the HEX LED, USBWiz, and XPORT AR.

The 16C754 Quad UART provides the mainboard with two general purpose serial ports – one DCE and one DTE. The other two UART ports are hardwired to the USBWiz and XPORT interfaces and are not available for general serial I/O. The UART uses the system clock to set baud rates, which can range from 50 to 230,400 baud.

The Epson 72421 Real Time Clock module provides time/date data to the system. It maintains timekeeping during power down via two sources: Standby +5vSB and a battery on the mainboard. Whenever the system power supply is in Standby (plugged in and turned on via the rear panel switch), or at full power operation, time/date data is maintain by +5vSB. When the power supply is completely shut down, the battery maintains the current date/time if jumper J12 is installed. Normally, J12 should be installed, but it is recommended that you leave J12 removed until after you power up and set the RTC.

The mainboard interrupt system provides 16 levels of interrupt processing via the CPU's /INT0 input line. /INT0 is placed in Interrupt Mode 2, and a vector address is passed to the CPU by U25 during the Interrupt Acknowledge cycle. This vector, when added to the internal I register contents, provides the 16-bit 'call' address of the interrupt service routine.

Several of the 16 interrupts are dedicated to specific sources. For instance, the Quad UART 'owns' interrupt levels 6, 8, 10, & 11 for serial ports A, B, C, & D, respectively. The RTC is assigned the Level 2 Interrupt. Interrupt Level 15 is assigned to the 'power off' function (see the 'Power ON/OFF Handling' section.)

The remaining interrupts are 'available' for general use at the Slot connectors.

Setup (cont)

The /INT0 interrupt system can be totally disabled by removing the mainboard jumper on J15. This action leaves the /INT0 line tied high so no /INT0 interrupts can be passed to the CPU.

Slot I/O decoding takes two forms: Slot 1/2 and Slot 3/4. See the 'System Operation' section for a description of Slot Addressing.

XPORT Setup

The XPORT AR is normally accessed through the CPU's internal ASCI-0 port. A secondary port provides access via the Serial-B port @ 68H. If for some reason the LAN connection to the XPORT is disabled, the secondary port can provide access to the XPORT Command Line Interface (CLI).

There are three basic methods you can use to access the XPORT, change the internal programming, and set the IP Address data:

- Device Installer. Normally, you will use the Device Installer software on the CDROM. After installing the software on your Windows computer, you can access the XPORT via the LAN connection, just like any Web page.

- Command Line Interface via the Local (Serial) Terminal UART. This option provides local serial terminal access to the CLI. Cisco-like ASCII command strings are used to set the various operating modes and parameters. To open the CLI in this mode, set SW1-4 to ON, then RESET the Stack-180. The CLI will be opened immediately after the terminal UART is initialized. Note that you MUST be running on the Local (Serial) Terminal (SW1-1 must be OFF) - it is not possible to open the CLI in this mode while running on the LAN-Based Terminal. When you have finished, press ESC to exit the CLI and resume the normal start-up initialization.

- Command Line Interface via the Serial-C connection. This CLI can be used with either the local serial terminal or the LAN-Based terminal. It is accessed via the Setup menu, option K. It operates in a manner identical to the CLI via the Local (Serial) Terminal UART. When finished with the CLI, press ESC to return to the Setup menu.

The CLI command options are numerous, confusing, and difficult to deal with unless you are quite well-versed in the command language used by Cisco routers. It is recommended that you avoid the CLI whenever possible, at least initially, and rely on the Device Installer for XPORT configuration. For detailed XPORT configuration information via the CLI, refer to the XPORT documentation on the CDROM. In particular, refer to \Design Files\XPORT AR\Command Reference.PDF.

If you received the XPORT from TG Consulting, it may have already been programmed and configured for your system - you shouldn't need to perform any configuration items.

If you purchased your XPORT from another source, it will not be configured properly for the Stack-180 - you will need to import the XML file contained on the CDROM and program/configure the XPORT yourself.

The Device Installer software allows you to configure the XPORT via the Web-based interface contained within the XPORT. If you know the IP address, you can use your regular Web browser in place of Device Installer. Note that we've used Microsoft Internet Explorer for this, but early versions of Firefox may return a 400 error (version 10 works well, however.)

For now, it is recommended that you use the Device Installer software provided on the CDROM. Install the program on your Windows or Linux machine, power up the Stack-180 (making sure it is connected to your LAN), and then execute DeviceInstaller.EXE.

Setup (cont)

A first-time installation may require that you download the data files for your XPORT. This is automatic. You can download data only for the XPORT, or for all/selected Lantronix products. At a minimum, you must download the XPORT AR data files.

When the DeviceInstaller finishes initialization, it will search your network for Lantronix devices. If it fails to find the XPORT, ensure your Stack-180 is powered up and restart the search. When the search is complete, you'll see the XPORT listed in the left column under the LAN connection used to access the XPORT. Double click on the XPORT entry to open the IP-based list of XPORT devices. Normally, there will be only one device listed. Double-click on the IP address of the device you wish to configure.

Accessing the XPORT via the LAN connection requires a user name and password. By default, the user name is "admin" and the password is "PASS". Note that upper/lower case is critical.

When the 'Device Details' Web page opens, click on the 'Web Configuration' tab, then on the green right-arrow in the upper left corner. You will be prompted for the username and password - the user is 'admin' and the password is 'PASS'.

After opening the Device Status page, page down to the orange "XML" tab and click it. Then, at the top of the new page, select "Import XML Configuration Record."

Click "Browse" and locate the "xml Stack export.xml" configuration file supplied on the CDROM. It will be located in the \Stack Design\Design Files\XPORT AR\ subdirectory. Click "Import" to import the configuration file into the XPORT.

When the import is complete, click on the orange "System" tab at the bottom of the orange tab area. At the top of the page, click "Reboot". After 30 seconds, the initial Status Page will be re-displayed. Since the IP address was 'released' during the reboot, you will have to click the 'Search' button to find the new XPORT address.

Select the orange "Network" tab. Page down to take a look at the current network settings, especially the IP address. At this point, the XPORT is set to 'DHCP ON,' meaning the XPORT will receive an IP address from the DHCP Server.

If desired, you can leave the settings as-is, leaving DHCP enabled. However, this may result in the XPORT being assigned a different IP address in the future and will make connecting the LAN-Based Terminal somewhat more difficult in that you'll have to determine and specify the IP address each time you open the TELNET client software. Instead, it is recommended that you assign a STATIC IP address. This will place the XPORT at the same IP address each time you power up, so you'll never have to search for its address.

To set the XPORT to a Static IP address, note the current (dynamic) address. This address was assigned by your DHCP Server (normally a router), as were the Network Mask, Gateway, and Domain name. In the upper section of the display page, click on DHCP Client OFF, set the IP to the desired address, and copy the Network Mast, Gateway, and Domain name from the 'Current Configuration' section to the 'Network Configuration' section. Click 'Submit'.

After clicking 'Submit', page down to the orange 'System' tab, click it, and the click 'Reboot'. The XPORT will reboot with the new settings.

Setup (cont)

USBWiz Setup

The USBWiz is accessed via the mainboard's Serial-C UART port, with a base address of 70H. Mainboard jumpers J8 and J13 are used to set the USBWiz operating mode. For Stack-180 purposes, a shorting block should be installed on J8; J13 should left open. This places the USBWiz in UART Interface mode.

The USBWiz powers up at 9600 baud, 8N1, RTS/CTS Flow Control enabled. You cannot disable the USBWiz's hardware flow control - it is always used by the USBWiz. If the incoming /CTS signal is HIGH (/CTS disabled) the USBWiz will not be able to output serial data. In a similar manner, the outgoing /RTS signal is used to control the flow of incoming characters. If the /RTS signal output by the USBWiz is HIGH, you must not send any character data; you MUST wait until /RTS is LOW, and you must check it before each character is sent.

The EEPROM is coded to set the USBWiz to 230,400 baud during the initialization sequence. Lower baud rates can be used if preferred, but it is recommended that you leave Serial-C set to 230,400 baud with HFC ON to provide the maximum transfer rate. Testing has shown it to be stable and error-free under all operating conditions.

EEPROM Setup

Entering the 'S' (Setup) command at the EEPROM command line prompt will execute the Setup code. The Setup menu looks like this:

```
Setup Menu Options

 A - CPU Parameters
 B - Terminal Control Strings
 C - MotherBoard Parameters
 D - D-GIDE Parameters
 E - PC Slot3 Parameters
 F - PC Slot4 Parameters
 G - Set System Defaults
 H - XPORT CLI
 I - Display All Settings
 J - ReBoot - Discard changes
 K - ReBoot - Test changes
 L - ReBoot - Write EEPROM

Select (ESC to Exit) ?
```

Setup (cont)

Option A – CPU Setup Parameters

This option displays the current CPU parameters and allows you to set/change them as desired:

```
Current CPU Parameters:
 Baud Rate @ 115,200     HFC: On
 Soft Reset: Off
 Memory Waits: 0         I/O Waits: 1
 MainBoard: Enabled
 Terminal Delay: 300

Options:
 1 - Set Terminal Baud Rate
 2 - Toggle Terminal Flow Control
 3 - Toggle Terminal Soft Reset
 4 - Set Memory Waits
 5 - Set I/O Waits
 6 - Set Terminal Delay Loop Timer

Select (ESC to Exit) ?
```

Terminal Baud Rate refers to the internal ASCI-1 port used for Local Terminal I/O. HFC stands for Hardware Flow Control; it can be used to control the data flow to the terminal, allowing much higher baud rates. Terminal Soft Reset, if enabled, will perform a software reset of the terminal at every restart, such as when an error occurs. Wait States can be used to slow the CPU when using slow memory or I/O devices. Normally, the Stack-180 runs with zero Memory Waits and one I/O Wait. (One I/O Wait State is the minimum allowed by the CPU core.) The Terminal Delay Loop Timer is a down-count loop executed before a character is transmitted to the Local Terminal via ASCI-1. Longer delay loop counts may allow a slow terminal to operate at higher baud rates without character overrun. The combination of Baud Rate and the Delay counter allows you to fine-tune the Local Terminal serial transmission for maximum data transfer rates.

Note that, if the Terminal Baud Rate is changed in Setup, you will also have to change the baud rate used by your Terminal. It is entirely possible that you could end up with different baud rate settings on the CPU and the Terminal. If this happens, the Terminal screen will fill with garbage rather than the standard system initialization data. When all else fails, you can press any key during the first three seconds of system initialization to reset the Stack baud rate to 9600 baud. To accomplish this, set your terminal to 9600 baud, no flow control. Perform a hard reset on the Stack. When the terminal screen begins to fill with garbage, press any key. The Stack will immediately restart at 9600 baud and you should see valid startup data appear on your terminal.

The proper sequence for changing the baud rate is:

1 - Power up, enter Setup, and set the desired Baud Rate and HFC status.
2 - Exit Setup via option L - ReBoot - Write EEPROM
3 - The system will restart, but garbage will be displayed on the terminal - this is expected.
4 - Set the terminal to the same baud rate/HFC as the Stack.
5 - Perform a hardware reset (or power cycle) on the Stack.

If you still see garbage, then HFC and/or Baud Rate is not compatible - select a lower baud rate or disable HFC.

Setup (cont)

Option B – Terminal Control Strings

Terminal Control Strings are used for display formatting; Clear Screen, Normal and Reverse Video, and Terminal Soft Reset can be assigned here:

```
Current Terminal Parameters:
 Enhanced Functions:    Enabled
 Clear Screen String:   ESC, [, 2, J, ESC, [, H, NULL, NULL, NULL
 Enter Reverse Video:   ESC, [, 7, m, NULL, NULL, NULL, NULL, NULL, NULL
 Exit Reverse Video:    ESC, [, 0, m, NULL, NULL, NULL, NULL, NULL, NULL
 Terminal Soft Reset:   ESC, [, !, p, ESC, [, 0, $, ~, ESC, [, ?, 7, h, ESC, [, ?, 5,
W, NULL, NULL, NULL, NULL, NULL, NULL

Options:
  1 - Toggle Enhanced Functions
  2 - Set CLS String
  3 - Set RevVid String
  4 - Set NormVid String
  5 - Set SoftReset String

Select (ESC to Exit) ?
```

The predefined control strings are VT-100 compliant and should work fine for any VT-100 (or higher) Terminal. If display problems exist with the predefined strings, they can be disabled with the Enhanced Functions toggle. In the current EEPROM code base, the strings can only be modified by editing the source code prior to compile-time.

The CLS, RevVid and NormVid strings are a maximum of ten bytes; shorter strings are NULL-filled. The Soft Reset string is 25 bytes, also NULL-filled.

Option C – Motherboard Parameters

```
Motherboard Parameters:
  Serial A: Enabled @ 9,600    HFC: Off
  Serial D: Enabled @ 9,600    HFC: Off

  USBWiz (Serial C):       Enabled @ 230,400  HFC: On
  XPort Line 1 (ASCI-0):   Enabled @ 230,400  HFC: On
  XPort Line 2 (Serial B): Enabled @ 115,200  HFC: On

Options:
 1 - Set Serial A Params
 2 - Set Serial D Params
 3 - Set USBWiz Params (Serial C)
 4 - Set XPort Line 1 (ASCI-0)
 5 - Set Xport Line 2 (Serial B)

Select (ESC to Exit) ?
```

These options allow you to configure the Baud Rate and Flow Control status of the serial ports on the mainboard, as well as the internal ASCI-0 port within the CPU.

Setup (cont)

Option D – D-GIDE Parameters

This is used to selectively enable/disable the four GIDE devices, and set the Cylinder, Head, and Sector/Track parameters for each device (if the emulation is activated,) and to determine the Command Set (ATA or ATAPI) used by the device.

```
GIDE CHS Parameters:
 Pri Master:    Cmd Set: ATA, Real Mode - CHS parameters obtained from drive.
 Pri Slave:     Disabled
 Sec Master:    Cmd Set: ATA, Real Mode - CHS parameters obtained from drive.
 Sec Slave:     Disabled

Options:
 1 - Set Primary Master
 2 - Set Primary Slave
 3 - Set Secondary Master
 4 - Set Secondary Slave

Select (ESC to Exit) ?
```

If a device is is set to enabled with the ATA command set, the system initialization process will attempt to access the device and, if emulation is enabled, set it to the specified CHS emulation parameters. If emulation is disabled (Real Mode), it will use the default CHS parameters supplied by the device.

If the ATAPI command set is active, the device will be ignored during system initialization. This would be the case for a CDROM or ZIP device.

Enabling a device that is not physically connected and powered up will cause timeout errors during system initialization.

Option E – PC Slot3 Parameters

The Stack-180 can access/use PC-Style Multifunction adapters, providing access to the COM, LPT, Floppy, and IDE ports on the adapter. This option sets the parameters for Slot3.

```
MultiFunction Adapter in Slot3:  Enabled

 COM1:   Enabled @ 9,600    HFC: Off
 COM2:   Enabled @ 9,600    HFC: Off
 LPT:    Enabled
 Floppy: Enabled
 PcIde3 Master: Cmd Set: ATAPI
 PcIde3 Slave:  Cmd Set: ATAPI

Options:
 1 - Toggle Slot3 Adapter Enable
 2 - Set PcCom1 Params
 3 - Set PcCom2 Params
 4 - Toggle LPT Enable
 5 - Toggle Floppy Enable
 6 - Set PcIde3 Master Params
 7 - Set PcIde3 Slave Params

Select (ESC to Exit) ?
```

Setup (cont)

If the Slot3 multifunction adapter is enabled, you can selectively enable/disable the various ports on the adapter. You can also set the baud rate and HFC status of the serial ports.

Option F – PC Slot4 Parameters

Identical to the Slot3 Parameters above, except for a Slot4 multifunction adapter. You can use multifunction adapters in one or both slots, and selectively enable/disable the ports as required for your hardware situation.

```
MultiFunction Adapter in Slot4:  Enabled

 COM1:   Enabled @ 9,600    HFC: Off
 COM2:   Enabled @ 9,600    HFC: Off
 LPT:    Disabled
 Floppy: Disabled
 PcIde4 Master: Cmd Set: ATA, Real Mode - CHS parameters obtained from drive.
 PcIde4 Slave:  Cmd Set: ATA, Real Mode - CHS parameters obtained from drive.

Options:
 1 - Toggle Slot4 Adapter Enable
 2 - Set PcCom1 Params
 3 - Set PcCom2 Params
 4 - Toggle LPT Enable
 5 - Toggle Floppy Enable
 6 - Set PcIde4 Master Params
 7 - Set PcIde4 Slave Params

Select (ESC to Exit) ?
```

The Slot3 and Slot4 displays above show that both slot adapters are present and enabled. Slot3 has all ports enabled with the IDE devices set to ATAPI – these are used for CDROM and ZIP drives. Slot4 has COM and IDE ports enabled, with the IDE devices in Real mode using the ATA command set. In this case, the two physical IDE devices are Compact Flash-IDE adapters with 256MB CF modules installed. Your configuration will probably be different.

Option G - Set System Defaults

System Defaults is used to define EEPROM defaults and selected hardware parameters passed to the CP/M BIOS at boot time.

```
Current Boot Default Parameters:

 A - Default GIDE Boot:         Primary GIDE, Master, Set #0
 B - Default PcIde Boot:        Slot4 PcIDE, Master, Set #0
 C - Default Floppy Boot:       Slot3 Floppy
 D - AtApi (ZIP) Device:        Slot3 PcIDE, Master, Set #0
 E - CDROM Device:              Slot3 PcIDE, Slave
 F - Default LPT/COM Slot:      Slot3 LPT/COM
 G - Default B/P BIOS IDE Drv:  Secondary GIDE, Master
 H - Default StackOS IDE Drv:   Primary GIDE, Master
 I - Auto Boot Setting:         Disabled
 J - USBWiz Flash Port:         USB-1, LUN #: 0

Select (ESC to Exit) ?
```

Setup (cont)

Options A, B, and C of the System Defaults menu are used to determine which of the ports is considered the 'default' port. These settings are used only by the EEPROM and are not passed to the CP/M BIOS. When booting from GIDE, PcIde, or Floppy, the Stack-180 command line can take several forms:

```
G<cr> Boot the default GIDE
GP    Boot the Primary GIDE
GS    Boot the Secondary GIDE
P<cr> Boot the default PcIde
P3    Boot the Slot3 PcIde
P4    Boot the Slot4 PcIDe
F<cr> Boot the default Floppy
F3    Boot the Slot3 Floppy
F4    Boot the Slot4 Floppy
```

Option D is used to set the location (device and drive select) of the AtApi ZIP device, if installed. It can be any one of the eight possible drives (PriGide, SecGide, Slot3 IDE, or Slot4 IDE, either Master or Slave.) This information is also passed to the CP/M BIOS and used by the StackOS AtApi ZIP driver. The B/P BIOS does not make use of this data. When the AtApi ZIP drive is defined in this manner, you can also boot the drive via the A<cr> command (no optional parameters are recognized with this command.)

Option E is used in a fashion similar to Option D, except for the CDROM device. This data is also passed to the CP/M BIOS at boot time, but the BIOS does not make used of it since there are no CDROM drivers present in either BIOS. Instead, the utility program CDZSWP.COM (with the Stack-180 driver) uses it to determine the location/drive select for the CDROM.

Option F – Default LPT/COM Slot is passed to the BIOS to determine which slot is providing LPT and COM port hardware services. Both OS models use this data to set driver parameters for these ports.

Option G – Default B/P BIOS IDE Drv. If the B/P BIOS OS is booted from an IDE device, that device is the drive used by the B/P BIOS for IDE operations. However, if a Floppy is booted, B/P BIOS would not 'know' which device to use for IDE operations, so the parameter set here is passed to the B/P BIOS and is used to define the IDE base port.

Option H – Default StackOS IDE Drv. When the StackOS is booted from an IDE device, that device assumes the Ide1 (primary IDE) driver. However, when a Floppy or AtApi ZIP drive is booted, the StackOS would not 'know' which port to assign to the Ide1 driver, so the parameter set here is passed to the StackOS and is used to define the Ide1 driver base port.

Option I – Auto Boot Setting. The Stack-180 will normally power up to the EEPROM command line and await command input. Setting the Auto Boot device to something other than 'No Auto Boot' will cause the Stack-180 to automatically boot the selected device 3 seconds after system initialization is complete. The Auto Boot process is aborted by pressing any key before the 3 second delay times out.

Option J – USBWiz Flash Port. This setting is used by the EEPROM Loader to determine the source USBWiz port (SD Card, USB-0, or USB-1) that the Loader will use for reading EEPROM or Boot images prior to burning the EEPROM or writing the boot partition. If the source is set to USB-0 or USB-1, you will also be prompted for a Logical Unit # (LUN) in anticipation of your using a multi-slot card reader. This data is passed to the CP/M BIOS and is used by the BACKUP program as the default device.

Setup (cont)

Option H - XPORT CLI

The XPORT Command Line Interface is a method by which you can access the internal parameter settings of the XPORT via a Cisco-like command line. This is a complicated subject and not suited for this manual. Refer to the XPORT documentation "Command Reference.PDF" on your CDROM in the \Stack Design\Design Files\XPORT AR subdirectory.

Option I - Display All Settings

A quick reference, one-page display of most System Setup parameters.

Option J – ReBoot – Discard changes

Selecting this option causes a full restart without writing to the EEPROM. A full restart consists of placing the EEPROM back in context and conducting the restart with the original EEPROM Setup data intact.

Option K – ReBoot – Test Changes

Test Changes will cause a restart, but it is done with the current (as modified) EEPROM Setup table variables. This allows you to test any changes without committing them to the EEPROM.

Option L – ReBoot - Write EEPROM

When all System Setup parameters are established as desired, selecting Option L will write the setup data to the EEPROM for future use, and then restart the EEPROM code with the new settings. You must be certain that the CPU board jumper J3 is in position 2-3 prior to selecting this option. Otherwise, the system will 'hang' as it attempts the write without the /WR signal being applied to the EEPROM. The new settings will remain in effect, even with all power off, until changed again. When the write is complete, you can set CPU jumper J3 to position 1-2 to write-protect the EEPROM.

System Operation - CPU Board

The CPU board design consists of eight sections: CPU, Clock, RAM, EEPROM, General Purpose I/O Port, Terminal I/O, Reset Generation, and I/O Port Decoding. The board has two connectors – J1 and J2 – that carry 30 pins each, consisting of power, ground, address, data, control, and I/O signals. (Memory access signals are not passed off the board, since all memory Read/Write operations are conducted on the CPU board itself.) The 30 pin headers are arranged in a .1" grid, so the CPU board can be easily adapted to a standard grid board for hobbyist applications.

CPU
The Z8S180-33 CPU is a fully static microprocessor that can run up to 33MHz. It features a wealth of internal I/O support as well as a Memory Management Unit (MMU) that can address up to 1MB of system RAM. Dual Programmable Reload Timers, dual DMA units, dual UART channels, and Wait State Generators are built in.

On power up, the CPU begins executing the EEPROM program. One of the very first operations the CPU performs is to determine the address of the Terminal I/O port, which will be either the internal ASCI-1 port for the local terminal or the internal ASCI-0 port if LAN-Based Terminal I/O is being conducted via the XPORT. After that, the CPU performs a quick memory check to determine the number of available 32K RAM banks. A 1MB system will show 32 banks of 32K each. If, for some reason, the CPU cannot detect any memory, it will execute a HALT instruction and the HALT LED on the CPU board will energize. This should never happen; if it does, there is a memory RD/WR problem. Occasionally, the HALT LED will light if the CPU was unable to read the EEPROM code, but this is pure chance.

After the memory check, the CPU begins the init sequence for internal, D-GIDE, and mainboard I/O ports, as determined by the EEPROM Setup code block. If a port is not enabled in Setup, it will not be initialized by the EEPROM init.

If the HEX LED option is installed on the mainboard, you'll be able to track the CPU init process through the various parts of the sequence. Each major step has a specific code, so you can see where a 'hang' occurs by watching the LEDs.

Finally, the CPU enters the EEPROM monitor loop, awaiting command input. At this point, the HEX LED will display 00H signaling that all is well.

Clock
The system clock runs at 29.4912MHz. An alternate clock at 18.432MHz can be used if desired. Crystal Y1 and capacitors C20 & C21 form the basic system clock. The clock signal is fed directly into the CPU's clock input via pins 3 & 4.

EEPROM
The system EEPROM is a 32Kx8 device that can be written in-system under software control. CPU board jumper J3 is the EEPROM Write Enable jumper – in position 1-2, the EEPROM is HARDWARE write protected; in position 2-3, the EEPROM is HARDWARE write enabled. The internal EEPROM design also contains a Software Data Protection feature that, when enabled, serves as a software write protect; this function is enabled when you received the kit. By following a specific algorithm, the EEPROM can be 'write enabled', have data written to it, and then return to the 'write protect' mode. This is the process followed when writing the Setup Code block to the EEPROM, and is also used when burning the entire EEPROM in-system.

RAM
System RAM consists of one or two 512Kx8 static RAMs with less than 20nSec access time to allow 0 wait state operation. The RAM chips are Surface Mount Devices and are already in place on an otherwise bare CPU board.

System Operation - CPU Board (cont)

U5 is a 16V8-7 GAL, programmed to select between EEPROM, LOW or HIGH 512K RAMs. Provisions are built in for additional RAM banking, with up to seven more 512K RAM chips that can be 'banked' in place of the HIGH 512K RAM.

General Purpose I/O Port

The GpPort is a RD/WR port at address 0F8H, used to swap the EEPROM in/out of context, bank the HIGH 512K RAM, and to provide control signals to the rest of the system. The Read and Write functions are NOT complementary in all respects. In other words, some bits during the WRITE function have different definitions during the READ function.

READING GpPort

Bits 0, 1, 2 High 512K Bank Select bits

Bit 3 0 - Normal XPORT operation
1 - XPORT RESET
Bit 4 0 - CTS Bit for ASCI-1 (Terminal)
1 - Not Clear to Send @ Terminal during HFC
Bit 5 0 - Jumper J10 (installed) - Motherboard Not Present
1 - Jumper J10 (removed) - Motherboard Present
Bit 6 0 - Jumper J9 (installed) - Clk at 18.432MHz
1 - Jumper J9 (removed) - Clk at 29.4912MHz
Bit 7 0 - RAM + ROM Memory Map
1 - RAM Only Memory Map

WRITING GpPort

Bits 0, 1, 2 High 512K Bank Select bits

Bit 3 0 - Normal XPORT operation
1 - XPORT RESET
Bit 4 0 - (Power is ON)
1 - PowerDown
Bit 5 0 - LED Disabled
1 - LED Enabled
Bit 6 0 - RTC Disabled
1 - RTC Enabled
Bit 7 0 - 32K ROM Enable (ROM+RAM Memory Map)
1 - 32K ROM Disable (RAM Only Memory Map)

All GpPort bits reset to 0 on power up or hard reset.

Terminal I/O

CPU Board Terminal I/O is conducted via the 'Local' terminal on the CPU's internal ASCI-1 port. The port is wired as a DTE port (Data Terminal Equipment) and is expected to be connected to a DTE port (Data Terminal Equipment) via a 'Null Modem' cable. Terminal control strings used by the EEPROM and the CP/M Z-System Terminal Capabilities (TCAP) file are VT-100 compliant.

System Operation - CPU Board (cont)

Reset Generation

U10 is the MAX707 (or Sipex 707) MPU Supervisor. At initial power on, it will hold the /RESET line active (LOW) for 200mSec after the +5VDC power supply reaches +4.65V. Any time the +5VDC rail drops below +4.65VDC, it will trigger a RESET condition, and will hold the RESET until 200mSec after power rises above +4.65VDC. It also supplies a de-bounce circuit for the front panel reset switch.

I/O Port Decoding

I/O decoding is performed by U8, a 20V8 GAL rated at 10nSec. (May use a 22V10 instead.) It supplies I/O signals for Reading/Writing the GpPort on the CPU board. For system expansion purposes, it also supplies /IoRd (I/O Read), /IoWr (I/O Write), /IntAk (Interrupt Acknowledge), and three, 8-address I/O port address enables at 60H-67H, 68H-6FH, and 70H-77H. In a CPU-only or CPU + D-GIDE configuration, these three port ranges can be used for any purpose. The mainboard uses these three port ranges for three of the four Quad UART select signals.

System Operation - Dual GIDE

The Dual GIDE board provides the Stack CPU with access to four IDE devices: a Master and Slave device on each of the two IDE ports. In addition, it buffers signals from the CPU board to the mainboard, and from the mainboard to the CPU board.

U1, U2, U5, and U6 form the Dual GIDE circuit. U6 is a 22V10-15 GAL (D-GIDE-1) that decodes CPU I/O address signals to provide A0, A1, A2, /CS0, and /CS1 signals to the IDE interface. These signals are common to both IDE ports. In addition, it decodes the /Select0 and /Select1 signals that are used to distinguish accesses between the two IDE ports.

U5 is a 22V10-15 GAL (D-GIDE-2) that further decodes the /SelectX signals to provide the /RD and /WR signals for each of the two IDE ports, and controls the data gating and direction of the IDE data bus.

U1 is a 74ABT646 bi-directional data port that passes low-byte data to/from the IDE data bus. U2 is another 74ABT646 that handles the high-byte data to/from the IDE data bus.

U3, U4, and U7 are 74ABT541 octal buffers. They are used to buffer the address, control, and timing signals from the CPU to the mainboard, and the interrupt signals from the mainboard to the CPU (/Int0, /Int1, and /Int2.)

Connector J5 is a standard 3.5” disk drive power header that is used to provide +5VDC to the D-GIDE and the CPU board when used in a non-mainboard configuration. If the D-GIDE is mounted to the mainboard, +5VDC is received from the mainboard and it is not necessary to provide power to J5.

J1 and J2 are 30-pin connectors on the sides of the D-GIDE. These connectors are arranged in a .1” grid and are used to connect the CPU board to the D-GIDE.

The J6 (34 pin) and J7 (26-pin) headers are used to connect the D-GIDE to the mainboard. They are arranged in a .1” grid to make it easy to create your own version of a mainboard.

J8 is a 2x2 right-angle pin header for external IDE Activity LEDs, one for each IDE port.

The J3 (Primary) and J4 (Secondary) 40-pin, right angle connectors are the IDE data cable connections. Pin 20 should normally be removed from these connectors, since many IDE cables have a 'plug' installed in the pin 20 socket. This is used as a 'key' for ensuring the cable connector is plugged in properly.

System Operation - Mainboard

Power ON/OFF Handling

System power on/off does not use a 'normal' on-off switch. Instead, it uses the ATX power supply's Soft-On and Soft-Off feature.

With the system plugged in and the rear panel power supply switch (if present) turned on, the system is in the 'standby' state, with +5vSB (Standby) power applied to the mainboard, but all other power is off. Pressing the front panel power switch will bring up main power and the system will begin running. Pressing the front panel power switch again, however, does not directly turn off system power. Instead, it generates an interrupt. When the CPU responds to the interrupt, it will wait until it detects that the power switch has been released; it will then output a control bit to the General Purpose port, causing a shut down.

The heart of the Soft-On/Soft-Off feature is relay K1 on the mainboard. With the power supply in standby, +5vSB is applied (through 10K resistor R2) to hold the power supply's Power ON signal in a HIGH State, preventing it from coming up to full power. +5vSB is also applied to one side of the front panel power switch. When the normally open switch is pressed, +5vSB is applied through diode D7 to contactor #2 on the K1 relay, energizing it and closing the normally open contacts, applying a ground to the power supply's Power ON signal. This brings the power supply to full power operation.

Once the power supply is up and stable, it energizes the Pwr_OK line, which is tied to the normally closed contacts on K1 contactor #1. The closed contacts pass the Pwr_OK signal through diode D9 to Contactor #2, holding it energized after the front panel switch is released. The Pwr_OK can take as long as 500mSec before it is stable. For this reason, an instantaneous button press may be too short to keep the power supply energized. Press – and hold – the front panel power switch for a second or so to eliminate this potential issue.

Pressing the front panel power button again has no direct effect on relay K1. Instead, a Level 15 interrupt is generated. The CPU responds to the interrupt by waiting for the button to be released. Once it detects a button release, it will output a '1' to Bit 4 of the General Purpose I/O Port. This bit is used to energize contactor #1 on relay K1, opening the normally closed contacts and removing the Pwr_OK signal that was holding contactor #2 energized.

At this point, main power shuts down and the power supply is left in the standby state.

When a LAN-based connection to the XPORT is established, the XPORT will set Configurable Pin 11 high. The CP11 signal (LanConn+) is tied to the power-on circuit through jumper J7 pins 1-2, then through diode D8. This connection operates in the same manner as if you pressed the front panel power ON switch: the system comes to full power. To disable the remote power-on feature, set J7 to position 2-3.

When running at full power with a remote, LAN-based connection active via the XPORT, you cannot use the front panel power switch to shut down the Stack. Instead, shutdown is accomplished by ‘arming’ the shutdown circuit (EEPROM ‘X’ command or CP/M OFF.COM utility) and THEN closing the LAN connection to the XPORT. Closing the LAN connection WITHOUT ‘arming’ the shutdown circuit will simply close the TELNET terminal client - the Stack will remain powered up and awaiting commands. Re-opening the TELNET client software will allow you to pick up exactly where you left off when the TELNET client was closed.

Finally, when running a LAN-based terminal connection, if you press the front panel power switch, the Stack will execute the shutdown interrupt code and the CPU will HALT, but main power will still be applied until you close the LAN connection to the XPORT.

System Operation – Mainboard (cont)

When using a power supply that does not conform to the Soft-On/Off requirements of the Stack-180, the mainboard J11 shorting block should be installed on pins 2-3. For a fully-compatible power supply, the shorting block will be installed on pins 1-2.

Real Time Clock

The Epson 72421 Real Time Clock module is used for timekeeping. No external components are required for the module to operate, although a battery is necessary for power-down timekeeping.

The RTC has two sources of power for maintaining the current time. When main power and standby power is shut down, a battery maintains the clock in a standby state whereby timekeeping is ongoing but I/O is disabled.

If standby power is applied (that is, the power cord is plugged in and the rear panel power switch is ON) then +5vSB is applied to the RTC. This is true even if main power is energized.

Interrupts

The Z8S180 CPU can accept interrupts from two basic sources: internal and external. Internal interrupts are generated by the components internal to the CPU core, while external interrupts are generated by components external to the CPU. The CPU has twelve interrupt sources – eight internal and four external - with the following priority:

Priority	Function	Source
(1)	TRAP (Undefined Op Code Trap)	Internal
(2)	NMI (Non Maskable Interrupt)	External
(3)	INT0 (Maskable Interrupt Level 0)	External
(4)	INT1 (Maskable Interrupt Level 1)	External
(5)	INT2 (Maskable Interrupt Level 2)	External
(6)	Timer 0	Internal
(7)	Timer 1	Internal
(8)	DMA channel 0	Internal
(9)	DMA channel 1	Internal
(10)	Clocked Serial I/O Port	Internal
(11)	Asynchronous SCI channel 0	Internal
(12)	Asynchronous SCI channel 1	Internal

The Trap Interrupt (priority 1) causes a restart to 0000H, similar to a system RESET. However, if a TRAP interrupt caused the restart, the TRAP bit (bit 7) of the Interrupt/Trap Control Register (34H) is set HIGH, indicating a TRAP occurred.

The NMI interrupt (Level 2) effects a direct CALL to memory location 66H. CP/M is not compatible with the NMI interrupt address, so the /NMI signal is tied HIGH; an NMI should never occur.

With the exception of /INT0, all other interrupts are handled in a manner similar to Interrupt Mode 2, whereby a vector address is supplied internally and added to the contents of the IL Register (33H), previously loaded with the base page of the interrupt vector table on a 256-byte boundary.

The /INT1 interrupt input is dedicated to Slot3; /INT2 is dedicated to Slot4.

System Operation – Mainboard (cont)

The /INT0 interrupt is virtually identical to the original Z80 /INT input line. It can be programmed to operate in Mode 0, Mode 1, or Mode 2 interrupts. Mode 2 is used in the Stack-180. The I Register is loaded with the high order 8 bits of the vector table; the low-order 8 bits are supplied by the mainboard during Interrupt Acknowledge. Together, the low- and high-order bits form a 16-bit address that points to an entry in the Interrupt Vector Table. This entry contains the 16-bit address of the handler for the specific interrupt.

The mainboard-generated /INT0 interrupt is based on a 1-of-16 decoder, allowing up to sixteen sources for interrupts on the /INT0 line. Eight of the /INT0 interrupts are dedicated to specific mainboard hardware; eight are available for general use by Slot 1 or Slot 2 expansion boards:

Interrupt Level	Source	Interrupt Level	Source
0	Slot 1/2	8	Serial-B
1	Slot 1/2	9	Slot 1/2
2	RTC	10	Serial-C
3	COM-A	11	Serial-D
4	COM-B	12	Slot 1/2
5	Slot 1/2	13	Slot 1/2
6	Serial-A	14	Slot 1/2
7	Slot 1/2	15	Power OFF

The COM-A and COM-B interrupts are tied to the PC-AT style expansion slots 3 and 4 for use with multifunction adapters that supply serial ports.

Quad UART

The 16C754 UART at U4 is a quad universal asynchronous receiver/transmitter (UART) with 64-byte FIFOs, automatic hardware/software flow control, and data rates up to 5 Mbits/sec. Two of the four UART ports (Serial-A and Serial-D) provide general purpose serial I/O at baud rates from 50 to 230,400 baud. The remaining two serial ports (Serial-B and Serial-C) are tied directly to the XPORT and USBWiz mainboard components.

Slot 1/2 Input/Output Addressing

Slots 1 & 2 are general purpose Stack Expansion slots. In addition to the typical I/O decoding and data bus signals, they also carry a part of the address bus. Specifically, they carry buffered A0-A7 signal lines directly from the CPU. U19 and U20 are used to 'port' the address data for A8-A15 to their respective Slots. In other words, A0-A7 come from the CPU; A8-A15 come from U19 (for Slot1) or U20 (for Slot 2). You can set any desired address by simply outputting it to the Slot address port. Slot 1 is ported at 0F1H; Slot 2 is at 0F2H.

System Operation – Mainboard (cont)

Slot 3/4 Input/Output Addressing

Slots 3 & 4 are somewhat different from Slots 1/2. Since these slots are intended to mimic the AT 16-bit expansion bus, it was necessary to follow the AT 16-bit conventions. This requires a 10-bit address for an I/O range of 000H-3FFH. Although 16-bit I/O addressing is possible with the Z180, it is quite cumbersome. 8-bit addressing is much easier, faster, and takes up less code space, but to do this, we had to come up with two more I/O address bits. A simple but effective method was used whereby Slot address lines A3-A11 (8 bits) are written to the Slot Address port, and CPU address lines A0-A2 are used directly on the Slot. In this manner, the three CPU address bits are added to the Slot Address port bits to generate an 11-bit Slot address – more than sufficient for the AT bus that uses 10 bits for I/O port decoding. In this manner, you can access all of the various I/O ports on the typical multifunction adapter that carries IDE, Floppy, Serial, and Parallel ports on one card.

Data Bus Buffer

Data to/from the mainboard is buffered via the 74ABT245 bidirectional octal buffer at U1. Since all mainboard operations are I/O, it is a simple matter to allow the buffer to send/receive only when the mainboard is being accessed. The following truncated I/O map shows when the buffer is activated for input/output:

I/O Ports	Device
00H-3FH	CPU internal registers
40H-5FH	Dual GIDE
60H-F7H	Mainboard I/O
F8H	GpPort on CPU board
F9-FFH	Mainboard I/O (if IoDec GAL version 1.5 or above is installed)

As a result, the buffer is enabled whenever an I/O operation (input or output) occurs within the range of 60H-F7H or F8H-FFH. The buffered /RD signal determines the direction of data flow, either towards the CPU during I/O READ, or towards the mainboard during I/O WRITE.

USBWiz

The USBWiz is accessed through the Serial-C port in the Quad UART. Since the USBWiz always powers up at 9600 baud, the EEPROM initialization process is used to change the baud rate for the UART and the USBWiz to whatever baud rate is determined by Setup - normally 230,400 baud.

Mainboard jumpers J8 and J13 are used to set the USBWiz operating mode. Normally, J8 is installed and J13 is removed, placing the USBWiz in the UART I/O mode. From the Stack's perspective, the USBWiz is simply another character device connected to a UART - just like a modem or serial printer. The USBWiz is then controlled through specific ASCII command strings sent to it via the UART.

XPORT AR

The XPORT is accessed through the Z8S180's internal ASCI-0 port. The baud rate is determined by Setup - it defaults to 230,400 baud with handshaking enabled. This will normally never be changed, although Setup permits it.

The XPORT is programmed to operate in a 'tunnel' mode, where terminal data is passed to/from the Stack via the ASCI-0 serial port.

System Operation – Mainboard (cont)

HEX LED

The HEX LED display is intended as an aid to troubleshooting, used to provide status information. It can display any HEX value from 00H-0FFH. The EEPROM Monitor has a series of POST codes that are used to track system init during the power-on sequence.

The display can be enabled/disabled by setting/resetting a bit in the General Purpose port (GpPort) at 0F8H. Bit 5, when SET, enables the display. When RESET, the display is disabled. Care MUST be taken when writing to the GpPort! See the General Purpose I/O Port section for details.

With the display enabled, writing a value to the LED Display port (0F0H) will latch the value in U12 and display it. The current value will be retained until you overwrite it or power down the system. Disabling/enabling the display will not change the latched contents of U12.

See Appendix 1 - EEPROM POST Codes for details.

System Operation - EEPROM Loader/Burner

The EEPROM can be re-burned in place. Although this is considered a risky operation, it has been done numerous times without a single report of trouble.

There are two methods by which the EEPROM can be re-burned: the EEPROM Loader and EEPROM.COM (a CP/M utility program.)

EEPROM Loader

The EEPROM Loader command can be used to burn the EEPROM if the USBWiz is attached to the Stack. You will have the option of burning the entire EEPROM, or excluding the Setup Table from the burn. Normally, you would exclude the Setup data, but if the Setup data arrangement changes, you will be required to burn it too, and you'll be required to completely re-enter the Setup data.

You also have the option of saving the EEPROM Image or Setup data table to the USBWiz device, and of burning the Setup data table from the USBWiz device file.

The USBWiz can be used to burn the EEPROM from an image stored on a USB Thumb Drive or SD Card. The Thumb Drive or SD Card must be formatted under the FAT filesystem, so is can be read/written via Windows. (For the purposes of this discussion, the SD Card or Thumb Drive will be referred to as the USB Drive.)

The USB Drive must have the following directory/file structure:

\CPMDATA\STKEPROM\EEPROM.BIN

To burn the EEPROM.BIN file into the EEPROM, you must be sure that the USBWiz is set to use the proper device, either SD Card, USB-0, or USB-1. Verify that the System Defaults (Setup Menu, option G) has the correct selection for the USBWiz Flash Port (Option J). It can be either USB-0, USB-1, or SD Card. (USB-0 is the port closest to the USBWiz circuit board.)

Exit Setup via the 'Write EEPROM and ReBoot' selection. When the reboot is complete, select the 'L' (Loader) command. The Loader menu will display the selected USBWiz port, followed by a list of menu options. Select option 2 - Burn EEPROM Image. After the Loader finds and opens the EEPROM.BIN file, you'll be presented with the option to continue or abort. If you abort, nothing changes. Continuing the burn process will result in an EEPROM burn. Only the data bytes that are being changed will be burned.

It is critical that the burn process not be disturbed in any way. A power outage, for instance, will wreck the burn and probably render the EEPROM unusable. More than likely, the only recovery method will be to ship the EEPROM to TGConsulting for re-programming.

The time it takes to burn the EEPROM depends on the number of bytes being burned. It's a fairly slow process. The HEX LED will show the burn progress via a downcount. As an example, starting with a completely blank EEPROM, the XICOR X28HC256 will take about 1.5 minutes for a full burn, while the ATMEL 28C256 will take just over 4 minutes.

Other than the HEX LED, there is no way to measure the progress of the burn. If it takes more than about 5 minutes, something went wrong and it's quite likely that your EEPROM is unusable. At this point, a hard RESET will probably result in a dead system, although it's possible, depending on the point of failure, that some or even most of the EEPROM is still usable.

System Operation - EEPROM Burner (cont)

EEPROM.COM - CP/M Utility

The A1: user area contains the EEPROM.COM utility. With this utility, you can copy the contents of the EEPROM to the OLDEPROM.BIN file, and write the EEPROM from the contents of the EEPROM.BIN file.

As with the EEPROM Loader burn, this is a slow process. Time to completion is approximately the same as with the EEPROM Loader routine. It displays the progress via the HEX LED, and also shows the progress as each 8K bank is burned.

When using the EEPROM.COM utility, you must first write the new EEPROM.BIN file into the A1: user area. Most often, this will be accomplished by downloading a DD-compatible floppy image file, writing it to floppy, and copying the floppy-based file to your A1: user area.

If you do not have a mainboard that can access a floppy adapter, you can use a CDROM instead. In this case, you would download the EEPROM.BIN file, write it to a CDROM, and use the Stack's CDZSWP program to read the file from the CD and write it to the A1: user area. Obviously, this requires that a CDROM drive be connected to the GIDE.

Operating Systems

StackOS Introduction

The Stack Operating System presents a partially-banked Z-System-based interface with a custom BIOS that provides additional memory banking. The CCP and BDOS are completely non-banked, but some of the system segments of the Z-System are placed in Banked memory. Specifically, the NDR, RCP, and FCP are in banked memory. The main BIOS is also resident in main memory, but the drive allocation tables are in Banked memory, releasing up to 4K of RAM for the Transient Program Area.

The StackOS CCP has been modified from the stock ZCPR 3.3 CCP to recognize that the RCP, FCP, and NDR are in banked memory.

An extension to the main BIOS has been placed in banked memory. The External BIOS routines can be called in a manner similar to a standard BIOS call. External BIOS routines perform such tasks as reading the RTC, writing the LCD, performing serial port init/read/write, and many other tasks that make programming easier. The External BIOS 'owns' a 32K slice of banked memory, so there is sufficient space for major hardware programming. In general, the routines associated with hardware that does not fall under the 'typical' CP/M umbrella are placed in the External BIOS. The calling conventions for External BIOS routines are explained in Section 6.

The StackOS, being fairly large, does not provide the largest possible TPA. However, the modular approach allows for fairly easy implementation of additional hardware drivers. Also, the BIOS was designed to accept certain parameters from the boot EEPROM, allowing the BIOS hardware drivers to be 'soft-coded' in that the port base addresses are supplied by the EEPROM, rather than being hard-coded into the BIOS. For instance, the boot IDE drive address is provided by the EEPROM; the BIOS knows the address of the port it was booted from, so it uses that port for the primary IDE driver. It can be any one of the eight possible Master or Slave IDE drives. The same applies to the Floppy and ZIP drivers - their port addresses are determined by the EEPROM and passed to the BIOS at boot time.

Stack OS Build Options

Build options are selected via compile-time equates in the BIOSZ.Z80 file. In this file, you can select which of the available disk drivers will be compiled, and which of the optional routines will be included. For instance, you can selectively add the IDE, Floppy, and ZIP drivers to the BIOS, and you can selectively build with Banked Allocation tables, Fast Warm Boots, BREAK key processing, and an interrupt-driven terminal.

There are also several mainboard-only equates. These can be enabled if the mainboard is present, and include a 64K Print Spooler, HEX LED and RTC drivers, and drivers for the LPT/COM ports carried on a Slot3/4 adapter.

While the BIOSZ.Z80 file determines **if** a disk driver is to be included, file TBL#EQU.LIB determines **how many** drives are attached to the driver. For instance, the PcFlop equate in BIOSZ.Z80 can be set TRUE to include the floppy driver; the PcFlopNd equate in TBL#EQU.LIB determines the number of drives that are connected to the floppy hardware.

When you've set the various options as desired, you can execute the batch command that will build a new BIOS and store it on the selected drive, ready to boot.

The command line "`MAKEBIOS A`" will compile the BIOS, (optionally) add it to the BPSYSGEN image, and (optionally) store it on the boot tracks of Drive A:. (Note that there is no colon after the drive name in the MAKEBIOS command line.)

Operating Systems (Cont)

B/P BIOS Introduction

The B/P BIOS is a fully banked operating system. It provides numerous options and improvements over standard CP/M.

B/P BIOS Build Options

B/P BIOS is built in two stages: the Boot BIOS and the Bank BIOS. The Boot BIOS is a limited, memory-restricted system. It provides all of the essential services of a CP/M BIOS, but minimizes much of the advanced processes in order to maximize the TPA. The Bank BIOS makes use of the full Stack memory map to provide the maximum TPA space as well as the advanced features built in to the entire OS.

The build options for each of the two B/P BIOS models (Boot and Bank) are set through the use of two parameter files: DEFBOOT and DEFBANK. These files contain all of the various equates used to define the system options. After setting the various equates in the DEFBOOT file, you would then compile the BIOS with "ZEX MAKEBOOT x", where 'x' is the destination drive. In a similar manner, after setting the equates in the DEFBANK file, you would execute the "ZEX MAKEBANK x" command line (again, 'x' is the destination drive.)

Once the optional parameters are set, executing the command line ZEX MAKEBOOT x or ZEX MAKEBANK x will create each BIOS and install it on Drive x: at the appropriate locations.

Generating a Boot Drive

The method used to build a boot drive depends on the hardware installed in your system. There are three basic situations: USBWiz, Floppy, and No Mainboard Present.

All three situations require that you overwrite all or part of the boot partition. 'No Mainboard Present' will do so via a CP/M utility. Floppy disk OS updates can be effected via the CP/M command line. USBWiz updates are accomplished via the built-in EEPROM Loader code.

USBWiz

The USBWiz can be used to generate a boot drive from an image stored on a USB Thumb Drive or SD Card. The Thumb Drive or SD Card must be formatted under the FAT filesystem, so it can be read/written via Windows (or Linux.) (For the purposes of this discussion, the SD Card or Thumb Drive will be referred to as the USB Drive.)

The USB Drive must already have the following directory/file structure:

Stack OS: \CPMDATA\STACKOS\INSTALL\INSTALL.IMG
B/P BIOS: \CPMDATA\STACKBP\INSTALL\INSTALL.IMG

Operating Systems (Cont)

The "INSTALL.IMG" file contains a complete image of the entire boot partition for a specific version of CP/M, which will be referred to as drive A: after boot time. **Installing a new boot image will completely overwrite anything that was previously stored on your A: drive.** The image contains a complete copy of all Operating System source code, help files, and utilities, in addition to the OS itself.

Rather than overwrite your regular boot drive, it would be prudent to use a different drive, if only to maintain a backup - the choice is yours.

Connect the IDE drive, install the USB Drive, and power up the Stack to the EEPROM Monitor. Select 'S' (Setup) and ensure the appropriate GIDE or Slot adapter is enabled. For loading the StackOS, the boot drive can be any of the GIDE or Slot3/4 Master or Slave drives; for B/P BIOS, the boot drive should be any of the GIDE or Slot3/4 Master drives.

Next, verify that the System Defaults (Setup Menu, option G) has the correct selection for the USBWiz Flash Port (Option J). It can be either USB-0, USB-1, or SD Card. (USB-0 is the port closest to the USBWiz circuit board.)

Exit Setup via the 'Write EEPROM and ReBoot' selection. When the reboot is complete, select the 'L' (Loader) command. The Loader menu will display the selected USBWiz port, followed by a list of menu options. Select option 1 - Write OS Image to IDE. You'll be presented with another menu that requires you to select the destination IDE device. (Remember, Stack OS can use any of the four IDE ports and either Master or Slave drive; B/P BIOS can use any of the four IDE ports, but must use a Master drive.)

Next, you'll be prompted to select which of the two operating systems you wish to load. Select either S for StackOS or B for B/P BIOS.

Finally, you have the option of writing the full disk image or simply the boot tracks. Normally, you'll choose the Full Disk Image.

The Loader will begin writing the boot disk image. If you're simply writing the boot tracks, it will only take a few seconds. Writing the entire disk image will take about 8-10 minutes, depending on the size of the image. (Currently, StackOS will write about 890 blocks; B/P BIOS is about the same.)

When the write is complete, press ESC to restart the Stack, and then boot the system on whichever drive you just loaded.

Floppy

Generating a boot drive from Floppy requires that your hardware includes the mainboard and a floppy adapter with at least one drive connected. Boot the floppy. When the boot process is complete, drive A: will be the floppy you just booted, B: is the second floppy drive, and C: is the default IDE drive (as set by EEPROM Setup, System Defaults option. Note that each OS model has a different default drive associated with it.)

Operating Systems (Cont)

Use the following command lines to generate the boot IDE drive:

`WIPEDIR C:` (This will wipe the directory contents for the boot partition. If you're simply updating an existing boot drive, you **must not** use this command - it will wipe the entire directory!)

`BPSYSGEN` (The sysgen command allows you to select the source [A:] and destination [C:] drives for a boot track copy.)

`UCOPY C*:=A*:*.* /SR` (Note the odd asterisk placement. The UCOPY program will copy all files from all user areas on A: to the same filename and user areas on C:.)

When UCOPY is complete, execute BOOT.COM, then boot from the selected IDE drive.

No Mainboard Present

Building a boot drive without a mainboard restricts your options significantly: you will be limited to the use of the BACKUP.COM program. In other words, you must already HAVE a boot drive, so you can load CP/M and execute the BACKUP.COM program. Additionally, you'll need a CDROM drive connected to the Stack, and a CD Writer drive connected to your Windows (or similar) machine.

Download the boot drive image via the Internet. Unzip the file, and write the resulting INSTALL.IMG file to the root directory of a CDROM disk. It can be a re-writable disk.

With the CP/M program CDZSWP, copy the INSTALL.IMG file from the CD to a CP/M drive/user area. Since this is an image of the boot drive's content, it will be fairly large; you'll have to use an empty - or nearly empty - drive partition for temporarily holding the INSTALL.IMG file.

Normally, BACKUP will create a boot drive from a USB Thumb Drive-based image file, but it also has a special option for creating a boot drive from a drive-based image file. In this case, the destination drive CANNOT be drive A: - BACKUSB will not allow you to overwrite the boot drive. To do so could make your machine unable to boot if a problem were to occur at the wrong time.

(This section - as well as BACKUP - is still being worked on.)

Building Data Drive Partitions

Thus far, we have built the primary boot drive for the selected OS model. Next, we'll add data files to the other on-line partitions. As with building a Boot Drive, there are several ways to accomplish this, with the USBWiz being the fastest and easiest.

USBWiz

Look on the Thumb drive to find the following sub path:

CPMDATA\STACKOS\IMAGE\

There, you should find DRIVEB.IMG, DRIVEC.IMG, DRIVED.IMG, and DRIVEE.IMG.

The drive name, and which partition it belongs to, should be obvious. You might not have all four images - I forget exactly what I sent you. The important ones are D: and E:, with B: being semi-important. C: is pretty much junk - you can do whatever you want with the C: partition.

If you're good so far...

Start the program BACKUP.COM. Select option 4 for Setup, and verify that sub-option has 'STACKOS' as the machine subdir (no, don't include the quote marks.)

If that's good, also check option M and N for the correct USB port for the Thumb Drive - Option N will always be LUN #0 for a Thumb Drive or SD Card. If you're using a multi-card reader, LUNs come into play for the various slots.

If all is good, press ESC to exit. If you changed something, you'll be prompted to save the changes. You should always save the changes unless you're just playing.

At the main menu, enter 2 for Restore, 3 for Imaging, and 1 to restore 1 drive.

It will ask for the drive image to restore. This may be confusing, so read it several times. You're being prompted for the restore source IMAGE, not the restore destination DRIVE. Remember you saw DRIVEB.IMG, DRIVEC.IMG, etc? You want to enter the single letter designation for that specific file. In other words, if you want to restore the DRIVEB.IMG file, the restore image is B. If you want to restore from DRIVEE.IMG, you'll enter an E.

OK - the RESTORE SOURCE IMAGE FILENAME, which MAY NOT BE THE SAME AS THE RESTORE DESTINATION DRIVE!!!

Since the images all come from identical partition sizes (except for partition 0, or drive A:) they are interchangeable, so it is fine if you wish to restore DRIVEC.IMG to the E: partition (under StackOS. B/P BIOS uses different partition sizes, with C:, D:, and E: being 36Mb – the images from those drives will not fit on the 8MB B: partition.)

After you've entered the source filename letter, it will ask for the destination drive, and provide a default with the same letter as the source you already entered. If that's what you want, simply press return and the restore will begin.

Programming Considerations

General Purpose Port

The General Purpose Port (0F0H) provides some measure of control over certain hardware functions. In a few cases, bits being read have nothing to do with bits being written. The specific bit positions are:

Bit	Read Purpose	Write Purpose
0,1,2	512K Bank Select	Same
3	XPORT Reset status	HIGH = XPORT Reset, LO=Clear Reset
4	Terminal /CTS	HIGH = Force Power Down
5	0 - CPU Bd J10 installed 1 - CPU Bd J10 removed	HIGH = LED Enable
6	0 - CPU Bd J9 installed 1 - CPU Bd J9 removed	HIGH = RTC Enable
7	0 - EEPROM in context 1 - EEPROM out of context	Same

Bits 0, 1, and 2 provide a one-of-eight decoder to select the upper 512K RAM chip. With all bits zero, the CPU board-mounted chip (underside) is selected. The SMDMEM16 Memory Map Decoder GAL only has 5 chip select outputs available, and the optional Memory Expansion Board can only carry 4 RAM chips, so although the 3 bits can decode one-of-eight, one-of-five is the limit based on the current hardware. The end result is a 3MB Stack180, rather than a 1MB machine. (There is a StackOS High RamDisk driver for memory beyond 1MB – there is no other use for it.)

Bit 3 is used as a RESET signal for the XPORT. Setting bit 3 HIGH provides an active LOW signal to the XPORT at J17, Pin 4. The same signal is also connected to Slot 1 and Slot 2 on Pin 21. Since the XPORT is powered by +5VSB, the XPORT will normally have power applied. A system RESET, if applied to the XPORT during power up, would 'break' the XPORT's connection to your LAN, an undesirable outcome. By moving the XPORT's RESET signal to the General Purpose Port, you can force an XPORT RESET whenever necessary, usually at a time where it will not 'break' the comm stream between the XPORT and your LAN.

Bit 4 is a dual-purpose bit in that the read function is different from the write function. When reading Bit 4, you are actually reading the status of the Clear To Send input signal from the local terminal serial port. (The Z8S180 core has a bug that prevent ASCI-1 from correctly presenting the CTS signal status, so an off-chip solution was required.)

Writing a HIGH to Bit 4 forces the Soft-Off function to activate, shutting down the system. There is no delay or 'Are you sure?' message - the system will power down immediately if the Soft-On/Soft-Off function is enabled.

Bit 5 is another dual-function bit. Reading Bit 5 reflects the status of J10 on the CPU board. If bit 5 is LOW, the J10 shorting block is installed. If Bit 5 is HIGH, the shorting block is removed. The EEPROM code uses this bit to determine if the mainboard is present. If the shorting block is installed (LOW) the mainboard is not present.

Setting Bit 5 HIGH turns on the LED display. Setting it LOW disabled the display.

Bit 6 is also a dual-function bit. Reading Bit 6 reflect the status of J9 on the CPU board. With a shorting block installed on J9, Bit 6 will be LOW; the EEPROM code uses this as a flag for the CPU clock being set to 18.432MHz. If Bit 6 is HIGH (J9 shorting block not installed) the EEPROM uses settings for a 29.4912MHz CPU clock.

Writing Bit 6 to HIGH enables the Real Time Clock for I/O. The RTC is in a standby state

Programming Considerations (cont)

whenever Bit 6 is LOW.

Bit 7 is used to place the EEPROM in or out of context. Writing a LOW to Bit 7 will place the EEPROM in context; a HIGH will put SRAM in context.

Real Time Clock

The Epson 72421 RTC must be enabled prior to attempting to access the RTC. To do so, you must set Bit 6 of the GP Port to HIGH (1). When the RTC access is complete, you should reset Bit 6 back to a LOW (0) to prevent a random access from changing the time, particularly during a power-down excursion.

To enable the RTC, set Bit 6 HIGH in this manner:

```
in      a,(GpPort)          ; Read the GpPort contents
and     10000111b           ; Mask bits
or      01000000b           ; Enable Bit 6
out     (GpPort),a
```

In the example above, note that bits 6, 5, 4, & 3 were masked to 0 after reading the GpPort contents. The bits are masked because the read status of these bits could create havoc if written back to the GpPort – most likely, you do not want to write those bits as they were read. Bits 6/5/4 during read have nothing to do with bits 6/5/4 during write, and bit 3 written HI will shut down the system without warning. Always be extra careful when writing to GpPort.

Setting Bit 6 HIGH via the OR statement will enable the RTC for I/O. When RTC I/O is complete, you should disable the RTC. This is accomplished with the following code snippet:

```
in      a,(GpPort)          ; Read the port
and     10000111b           ; Mask bits 6/5/4/3
out     (GpPort),a
```

Slots 1 & 2

Expansion Slots 1 & 2 are identical with the sole exception of the Base Address Port. Writing to 0F1H (for Slot 1) or 0F2H (for Slot 2) will set the address latch for the upper 8 bits of the slot's 16-bit address bus. Bits 0-7 are tied directly to the CPU's address bus. In this manner, the upper 8 address bits can be generated, while the lower 8 bits are 'real time' CPU address bits.

The Base Address Ports for Slots 1 & 2 are Write Only.

Slots 3 & 4

Expansion Slots 3 & 4 are identical with one exception - the Slot Offset Address Port address. The original PC-XT and PC-AT used 10 bits for I/O, from 0000H-03FFH. Most Z8S180 I/O only uses 8 address bits, and the Stack only provides 8 bits to the mainboard, so a method had to be devised to account for the two missing I/O address bits. The lower 3 bits (A0, A1, A2) are 'real-time' address bits driven by the CPU. Bits 3-10 are driven by the address register for that slot: 0F3H for Slot3, and 0F4H for Slot4. With the 8-bit slot offset value in the address register, plus the 3-bit value driven by the CPU, there are 11 address bits used to drive Slots 3 & 4 I/O (0000H-07FFH).

The Slot Offset Address Ports for Slots 3 & 4 are Write Only.

Programming Considerations (cont)

HEX LED

The HEX LED is enabled and disabled in a manner similar to the RTC, but rather than permitting I/O to take place, enabling the HEX LED simply turns ON the display; disabling it will turn the display off. When off, the port will still accept new data for display, but the LEDs will not display it until they are turned on.

To enable the HEX LED, execute the following code snippet:

```
in      a,(GpPort)          ; Get the port contents
and     10000111b           ; Mask bits
or      00100000b           ; Enable the display
out     (GpPort),a
```

To disable the display, execute the following:

```
in      a,(GpPort)          ; Get port contents
and     10000111b           ; Clear 6/5/4/3
out     (GpPort),a
```

USBWiz

XPORT AR

StackOS External BIOS

The External BIOS available under the StackOS provides quick access to a number of routines that can access and control the various hardware of the Stack – routines that are not part of the normal set of CP/M support routines.

The External BIOS calling conventions are similar to those associated with a BIOS CALL. However, a BIOS CALL is always possible – the system will not run without a BIOS. The External BIOS may not be present; calling the External BIOS without it being loaded will always return an error code 2 – External BIOS Not Loaded.

The External BIOS is accessed by CALLing the entry point in low RAM: CALL ExtBios (ExtBios is defined in A4:Page0.LIB.) The entry point then performs a JP to the External BIOS dispatch routine built-in to the main BIOS. Several registers must be loaded with various parameters prior to the initial CALL to ExtBios.

The general calling parameters for External BIOS routines take the following form:

```
Entry:      C has Function #
            B has SubFunction #
            A has byte data (if required)
            HL has two-byte or word data, or an address pointer (if required)
```

Exit: A has single character return data
F has routine exit status
HL points to External Bios Buffer

Uses: A, HL – all other registers preserved.

Low RAM carries pointers to the External BIOS handler. Defined in A4:Page0.LIB, the byte at address (EBiosLd) is Z if the External BIOS is NOT loaded, and NZ if the External BIOS is loaded into Banked RAM. You can test (EBiosLd) if desired, but the test is always performed by the dispatch handler, and it will return an error (Carry Set, A=2) if the External BIOS is not loaded.

Address (ExtBios) is a 3-byte JP xxxx sequence, jumping directly into the BIOS-based dispatch handler routine.

File A4:EXTBIOS.LIB describes the register requirements for the individual routines. In all cases, the C register carries the Function and B carries the SubFunction. For instance, Real Time Clock I/O is handled by Function C=1=ClockIO; there are 4 possible SubFunctions that are determined by Register B:

1	ClkDisp	Read & display Date/Time at current cursor position
2	ClkRd	Read and return pointer to date/time string
3	ClkStor	Store current date/time in memory
4	ClkDiff	Determine difference between now and Stored time/date

To CALL the ClkDisp routine, simply load BC with the Function & SubFunction, and call the External BIOS:

```
        MACLIB      A:PAGE0.LIB
        MACLIB      A:EXTBIOS.LIB
...
        LD          BC,ClkDisp*256+ClockIO    ; C=ClockIO, B=ClkDisp
        CALL        ExtBios
```

(Don’t forget to include the two MACLIB statements to define the labels used in calling the External BIOS.)

For all ExtBios CALLS, the HL register is used to return a pointer to the 35-byte External Bios Buffer.

The ClkDisp routine displays the RTC date/time at the current cursor position, so no data needs to be returned to the caller. ClkRd, however, must return a pointer to the date/time string. In this case, the string is stored in the External Bios Buffer; HL points to this buffer.

In Case of Trouble

General Problems:

- Check your soldering. Verify no solder bridges or other shorts on any solder connection.

- Verify that all sockets, headers, connectors, and pins are properly soldered. It is not uncommon to miss a socket pin while soldering.

- Ensure all components are installed correctly. Pin 1 is always the square pin.

Power Supply Problems:

Ensure that the K1 relay is installed properly.

We have found that some ATX power supplies do not conform to the ATX specification, particularly with those prior to the ATX12V spec. Also, some proprietary power supplies might meet the ATX design requirements for form and fit, but are not wired as per the specification for the 20-pin power connector. These power supplies MUST BE AVOIDED! If your power supply's 20-pin connector does not match the following table for wire color and voltage, it does not meet the ATX specification and might be a proprietary power supply that will cause major trouble:

Pin	Name	Color	Pin	Name	Color
1	3.3V	Orange	11	3.3V	Orange
2	3.3V	Orange	12	-12V	Blue
3	COM	Black	13	COM	Black
4	5V	Red	14	PS_ON	Green (Note 3)
5	COM	Black	15	COM	Black
6	5V	Red	16	COM	Black
7	COM	Black	17	COM	Black
8	PWR_OK	Gray (Note 1)	18	-5VDC	White (Optional - Note 4)
9	+5vSB	Purple (Note 2)	19	+5VDC	Red
10	12V	Yellow	20	+5VDC	Red

Notes:
1. The PWR_OK signal is normally used to hold power applied when when releasing the front panel power switch (or when removing the jumper across J6 during testing.) This signal (or the lack thereof) is a problem with some older ATX power supplies. If this problem exists, you can use the work-around of moving mainboard J13 from 2-3 to the 1-2 position. This causes the system to use +5vdc to hold power energized rather than the PWR_OK signal. If the problem persists, you may have to operate with Soft On/Off disabled by placing a shorting block on mainboard J11, position 2-3.

2. +5vSB (standby) power is used on the mainboard to perform the Soft-Power ON function and to maintain the RTC (which is also battery-backed in case of a loss of all power.) It is also used to power the XPORT adapter for remote Wake-On-LAN, where main power is activated when a LAN connection is established with the XPORT. Finally, it also provides the +3.3vSB voltage (via U29) used by the XPORT and Slot connectors.

3. The PS_ON signal (active LOW) is sent from the mainboard to the power supply, and is used to bring up main power. With mainboard J11 in position 2-3, PS_ON is ALWAYS applied, so main power will come up as soon as the rear panel power switch is turned on.

4. -5VDC has been removed from the ATX12V spec, so this wire will likely be absent. Nothing in the Stack system uses -5VDC, so it's not required for normal operation. Some (a very few) 8- or

In Case of Trouble (cont)

16-bit XT/AT adapter boards used -5VDC. The Stack mainboard has provisions for creating -5VDC from -12VDC if it ever becomes necessary for a particular adapter board application. There is a jumper (labeled '-5') on the mainboard that will pass -5VDC to the slot connectors if your power

supply provides -5VDC. If the power supply does not provide -5VDC, the jumper should not be installed. In this case, -5V can be created from the -12VDC supply by installing U30 and C110/C111. U30 is a 7905 regulator in the TO-220 package, capable of supplying 1 amp at -5VDC; C110 and C111 are 47uf electrolytic filtering capacitors. These parts are easily obtained from most any electronic parts supplier.

Dead System

This is the worst-case scenario, since there is no on-line method by which you can test the various components to determine where the fault lies. Start with the power supply: Are the power signals as described in the 'Power Supply Problems' section above? Does it come up to full power when the front panel power button is pressed? Does it remain at full power when the button is released?

If the power supply is ATX12V compliant, it is possible that a short on one of the power rails is preventing the power supply from coming up to full power.

If you have a mainboard-based system, remove the CPU and D-GIDE boards from the mainboard, connect +5Vdc to the D-GIDE via the on-board 3.5" floppy power connector, and attempt to power up without the mainboard. A successful power-up points to a mainboard fault; otherwise, the CPU board is the likely culprit.

XPORT Problems

Programming the XPORT can be troublesome. Refer to 'XPORT Setup' and verify the proper IP address is set in the XPORT. Also verify that your TELNET software is using Port 10001 to connect to the Stack Terminal. Contact TG Consulting for detailed assistance.

USBWiz problems

Ensure mainboard jumper J8 is installed and J13 is removed. This places the USBWiz in the UART Interface mode.

Interrupt Problems

Ensure that jumper J15 is installed on the mainboard.

There are only two ways the mainboard uses interrupts when the EEPROM program is running:

- RTC 1-minute interrupt. The system prompt includes the current time; when the minute counter rolls over, the system prompt line will automatically update with the new time.

- In the Testing menu, you can run the 50mSec PRT-0 test. Start the test; it should end automatically in approximately 2 seconds.

If the RTC 1-minute interrupt works correctly, the mainboard interrupt system is responding

properly. If the PRT-0 50mSec interrupt test passes but the RTC prompt does not, you should re-init the RTC time via the CS command. Press 'C' to enter the 'Clock' mode, and 'S' to enter Clock Set. Follow the prompts to enter the current time and date.

In some cases, particularly when the RTC is first powered up from a full-off state, it's possible that the RTC is locked in an unstable condition. To clear this, use the CR command (Clock Reset) to initialize the RTC to a known state, then follow with the CS command (Clock Set) to set the current date/time.

If the PRT-0 50mSec test fails, the Programmable Reload Timer internal to the CPU may be at fault, or memory is not being accessed properly during the down-count operation.

Terminal Problems

Terminal-related problems will generally fall into one of two categories: Baud Rate or cable connections.

The default EEPROM is coded to set the terminal port to 9600 baud, no flow control. Your terminal should be set to the same. Additionally, ensure the terminal is set to send CR when CR is pressed (NOT CR/LF – a two-character CR/LF sequence will cause trouble.)

Serial cable connections have been known to cause many headaches. If your Stack appears to be running (green LEDs lit, HEX LED displaying various test codes) but you're not receiving data on the Terminal, then a likely source of trouble will be the terminal data cable. The Stack CPU kit is delivered with a 9-pin terminal pigtail wired for DTE operation. Most terminals use a DTE serial port, so a 'Null Modem' cable would normally be used between the Stack-180 and the Terminal. The same applies if you're using a Windows machine for your terminal. See 'Appendix E - Terminal Serial Connections' for additional information.

If you're terminal is displaying data, but it appears to be garbage, ensure the terminal's baud rate is set appropriately. The Stack defaults to 9600 baud on initial power up, but may have been set to a different value while running the Setup utility. If you're unable to display good ASCII data and cannot determine the proper baud rate, perform the following:

- Set the terminal to 9600 baud
- Perform a hard reset
- Within 3 seconds of the hard reset, press any key on the terminal keyboard.

This process forces the CPU to run at 9600 baud, 8N1, no handshaking.

Setup Problems

If the system seems to 'lock up' when saving the EEPROM Setup data block, ensure jumper J3 on the CPU board is connected between pins 2-3. This enables the EEPROM Write signal and allows you to save the EEPROM Setup data back to the EEPROM. To Write Protect the EEPROM, install jumper J3 between pins 1-2. DO NOT change this jumper while system power is applied!

If you're seeing IDE timeout errors during system initialization, you probably have a GIDE or Slot IDE drive enabled, but the drive is not present, causing the timeout. Ensure only drives that are connected and powered are enabled in Setup.

When all else fails, contact TG Consulting via e-mail at tgcons@cfl.rr.com.

Parts List – Stack-180 CPU

Sockets:
1 @ 68-pin PLCC
1 @ 28-pin DIP
1 @ 24-pin .3” DIP
3 @ 20-pin DIP
1 @ 16-pin DIP
1 @ 8-pin DIP

Integrated Circuits:
U1 - 1 @ Z8S180-33 CPU PLCC
U2 - 1 @ 28C256 EEPROM – pre-programmed
U3/4 - 1 or 2 512Kx8-17 SRAM (pre-installed)
U5 - 1 @ 16V8-10 GAL – pre-programmed Memory Decoder (SMDMEM 1.6)
U6 – 1 @ 74F273
U7 – 1 @ 74ABT244
U8 – 1 @ 22V10-15 GAL – pre-programmed I/O Decoder (S180IO 1.4) OR
U9 – 1 @ MAX232A – RS-232 Level Converter
U10 – 1 @ MAX707 (or SIPEX SP707CP) (or MAXIM DS1707CPA) Reset Generator

Possible substitute:
U8 - 1 @ 20V8-15 GAL - pre-programmed I/O Decoder (S180IO 1.3)

Resistors:
5 @ 10K ¼ watt – R2, R3, R6, R7, R8
3 @ 330 ¼ watt – R1, R4, R5
1 @ RN1 – 10K, 10-pin bussed

Capacitors:
10 @ .1ufd mono – C1-C10
2 @ 22pfd mono – C20, C21
4 @ 47ufd electrolytic – C101-C103
4 @ 1ufd electrolytic (or Tantalum) – C91-C94

Misc:
1 @ Crystal – 29.4912MHz
1 @ Stack CPU PCB
1 @ T1-3/4 Red LED
1 @ T1-3/4 Green LED
1 @ 3-pin straight header
4 @ 2-pin straight headers
2 @ header jumpers (shorting blocks)
1 @ 2x5 (10-pin) shrouded header
1 @ Terminal pigtail
2 @ 1x30 expansion pin headers
1 @ CDROM

Parts List – Dual GIDE

Sockets:
4 @ 24-pin .3” sockets
3 @ 20-pin sockets

Integrated Circuits:
1 @ 22V10-15 GAL, pre-programmed D-GIDE-1 @ U6
1 @ 22V10-15 GAL, pre-programmed D-GIDE-2 @ U5
2 @ 74ABT646 @ U1, U2
3 @ 74ABT541 @ U3, U4, U7

Resistors:
1 @ 10K ¼ watt resistor @ R1
2 @ 330 ohm, ¼ watt resistors @ R2, R3
1 @ 10K 6-pin bussed resistor @ RN1

Capacitors:
7 @ .1uf monolithic caps
4 @ 47ufd Electrolytic @ C8-C11

Misc:
2 @ 1x30-pin socket header @ J1, J2
2 @ 2x20 right angle pin header – IDE connectors @ J3, J4
1 @ 4-pin Power connector – 3.5” male floppy type @ J5
1 @ 2x17 straight pin header @ J6
1 @ 2x13 straight pin header @ J7
1 @ 2x2 right angle pin header external LED @ J8
2 @ 2-pin straight header @ J9 & J10 (Rev D and above)
1 @ D-GIDE PCB

Hardware:
4 @ 1/2” 4-40 F-F standoffs
8 @ 1/4” 4-40 machine screws

Other: (These are normally in the mainboard kit – included here if mainboard not ordered.)
1 @ 2x13 pin socket @ J7
1 @ 2x17 pin socket @ J6

Parts List – Stack Mainboard

Sockets:
1 @ 68-pin PLCC
4 @ 24-pin .3” DIP
11 @ 20-pin DIP
1 @ 18-pin DIP
2 @ 16-pin DIP
5 @ 14-pin DIP
2 @ 8-pin DIP
1 @ CR-2032 Battery Socket

Integrated Circuits:
U1 – 74ABT245 (or 74F245)
U2 – 74F240 Octal Buffer
U3 – 74ABT541 Octal Buffer
U4 – 16C754 Quad UART
U5 - SN75185 RS-232 level-shifter
U6 – SN75185 RS-232 level-shifter
U7 – 74ABT646 Octal Transceiver
U8 – 74ABT646 Octal Transceiver
U9 – 74F08 Quad NAND
U10 – GAL22V10-15 – Mainboard I/O Decoder – Pre-programmed. Labeled “MoBoIO 1.6”
U11 – GAL22V10-15 – Slot I/O Decoder – Pre-programmed. Labeled “SlotIO 1.1”
U15 – 74F240 Octal Buffer
U16 – 74F04 HEX Inverter
U17 – 74F273 Octal Latch
U18 – 74F273 Octal Latch
U19 – 74F273 Octal Latch
U20 – 74F273 Octal Latch
U21 – 74F04 HEX Inverter
U22 – 74F148 Interrupt Decoder
U23 – 74F148 Interrupt Decoder
U24 – 74F08 Quad NAND
U25 – 74ABT541 Octal Buffer
U26 – Epson 72421 RTC module
U27 – 74F04 HEX Inverter

Diodes:
D1 – GRN T1¾ LED
D2 – 5.1V Zener
D3 – 1N4148
D4 – 1N4148
D5 – 5.1V Zener
D6 – 5.1V Zener
D7 – 1N4148
D8 – 1N4148
D9 – 1N4148
D10 – YELLOW T1¾ LED

Parts List – Stack Mainboard (cont)

Resistors:
3 @ 10K ¼ watt – R2, R3, R5 (Brown, Black, Orange)
2 @ 330 ¼ watt – R1, R4 (Orange, Orange, Brown)
3 @ 47K ¼ watt – R6, R7, R8 (Yellow, Purple, Orange)
2 @ 1K ¼ watt – R9, R10 (Brown, Black, Red)
RN1 – 10K, 6-pin bussed (6A103G)
RN2 – 220, 8-pin bussed (8A221G)
RN3 – 220, 6-pin bussed (6A221G) (may be 270 ohm – 6A271G))
RN4 – 10K, 6-pin bussed (6A103G)
RN5 – 220, 6-pin bussed (6A221G) (may be 270 ohm – 6A271G)

Capacitors:
31 @ .1ufd mono (Labeled as '104')
8 @ 100ufd electrolytic – C100-C107
6 @ 330pfd ceramic – C53-C55, C63-C65 (Labeled as '331')

Headers and Jumpers:
1 @ 34-pin socket header (2x17) (J1) (D-GIDE plug in)
1 @ 26-pin socket header (2x13) (J2) (D-GIDE plug in)
1 @ ATX power connector (J4)
3 @ 2-pin jumper (J6, J12, J15)
2 @ 3-pin jumper (J7, J11)
3 @ Fan power connectors (J9, J10, and J14)
2 @ 2x30 slot socket headers (Slot 1, Slot 2)
2 @ 96-pin AT slot connectors (Slot 3, Slot 4)
1 @ stacked DB-9 connector (male over female) (Tyco 1734282-3)
1 @ 4-pin pin header (J8)

Hardware:
7 @ ¼" 6-32 M-F standoffs
7 @ 6-32 HEX nuts
7 @ 6-32x1/4" machine screws

Misc:
1 @ LBA110 Dual CMOS Relay (K1)
1 @ Mainboard PCB
1 @ 4-position DIP switch
1 @ Lithium Battery CR-2032
4 @ Shorting blocks
1 @ Assembly Instructions
1 @ PCB Layout Diagram
1 @ Schematic Diagram

Stack ALFAT USB Kit (Mainboard Option)

Capacitors:
C4 - .1ufd mono (Labeled as '104')

Headers and Jumpers:
1 @ 18-pin socket header (1x18) (J3 on Mainboard)
1 @ 18-pin pin header (1x18) (J3 on ALFAT USB)

Hardware:
2 @ 1/2" 4-40 F-F standoff posts
4 @ 4-40 x 1/4" machine screws
1 @ ALFAT USB daughterboard

Stack XPORT Kit (Mainboard Option)

Sockets:
1 @ 14-pin DIP @ U29

ICs:
U28 – LM3940IT (3.3 volt regulator)+
U29 – 74F04 Inverter

Resistors and Capacitors:
1 @ .1ufd mono @ C29 (Labeled as '104')
2 @ 47ufd electrolytic @ C108/C109
1 @ 220 Ohm @ R11 (Red, Red, Brown)

Headers and Jumpers:
2 @ 8-pin socket header (1x8) (J16 & J17 on Mainboard)
2 @ 8-pin pin header (1x8) (J16 & J17 on XPORT Daughterboard)

Hardware:
2 @ 1/2" 4-40 F-F standoff posts
4 @ 4-40 x 1/4" machine screws
1 @ 4-40 x 3/8" machine screw
1 @ 4-40 machine nut
1 @ XPORT AR mounted on daughterboard

Stack HEX LED Kit (Mainboard Option)

Capacitors:
C12 - .1ufd mono (labeled as '104')

Sockets:
2 @ 14-pin DIP @ U13 and U14
1 @ 20-pin DIP @ U12

ICs:
2 @ TIL311 Hex LED (U13, U14)
1 @ 74F273 (U12)

Stack -5VDC Kit (Mainboard Option)

Integrated Circuit:
1 @ 7905 (-5VDC Regulator)

Capacitors:
2 @ 47ufd electrolytic

Hardware:
1 @ 4-40 x 3/8” machine screw
1 @ 4-40 machine nut

CDROM Contents

CDRomRoot:\Stack Design Subdirectory

Absolute TELNET Subdirectory

Contains the Absolute TELNET (AT) 9.18 installation file. AT is used as the primary TELNET program for the LAN-based terminal. Hosted at www.celestialsoftware.net, AT is regularly updated and available at a good price. The current version is v9.18, dated 7 Jan 2011, and is available for download at www.celestialsoftware.net/download-selection.html. Demo, Lite, and Professional versions are all supported by the same download. The USER file is an AT keymap file for the Stack180. It redefines certain keys on the PC keyboard to provide the key sequences used by WordStar. For instance, PgUp, PgDn, Home, End, arrow keys, etc, are set to generate the required ESC sequences that WordStar uses for those functions. The same WordStar arrow key definitions are also used in the Z3CPR TCAP definition.

Boot ROM (EEPROM Source Code) Subdirectory

Contains the complete and current EEPROM source code for the Stack180. Also contains an archive of all versions, all the way back to Rev A of 08-14-06. All EEPROM versions were created and compiled using ZDS 3.68 (see below.)

Design Files Subdirectory

Contains numerous subdirectories with various hardware-related documents. Included are the USBWiz and XPORT subdirectories, which carry manufacturer-supplied data files. You'll also find subdirectories that contain There are a few specific files that should be noted:

Design Files\XPORT AR\setup_di_x86x64cd_4.3.0.3.exe - latest version (as of 2/1/2012) of the Lantronix Device Installer for the XPORT AR. Installs on your WinBox machine and connects to the XPORT via the 10BaseT LAN connection. Allows you to perform all programming required to use the XPORT with your Stack180. This install file contains all necessary installation data and does not require Internet access.

Design Files\XPORT AR\xml Stack Export.xml - This file contains the XPORT AR configuration record for proper Stack180 operation. You can easily program the XPORT AR configuration using the built-in software with this configuration file.

Design Files\ZILOG\ZDS368.exe - Zilog Developer Studio, v 3.68. An integrated development environment with Z180 cross-assembler for installation on your WinBox machine. The EEPROM source code was built under ZDS 3.68.

Design Files\PALASM - The PALASM subdirectory contains the entire MSDOS version of the PALASM 4 program, version 1.5. You would normally copy the entire subdirectory's contents to the desired location on your hard drive, then create a shortcut pointing to the \EXE\PALASM.EXE file. PALASM was used to compile all GAL source code. There is a PALASM shortcut in the root PALASM directory. This provides an example of how to set up the shortcut to point to the correct subdirectory for the GAL files. Looking at the properties, you'll see that the path to the PALASM root directory was set to C:\ProgFile\PalASM.

The command line attempts to execute the C:\ProgFile\PalASM\EXE\PALASM.EXE file and sets the working directory to C:\Computer\Stack180\GalFiles. Obviously, you would need to change the paths to reflect your file structure and the location of the PALASM program files.

GALFiles Subdirectory - Contains all source code files for the various GALs used in the Stack180. Each file was compiled via PALASM (see above.) Also has an archive of older source files.

CDROM Contents (cont.)

Tera Term Subdirectory - TeraTerm is another useful TELNET program for your WinBox when using either the serial or LAN terminal on the Stack180. The installation file is in the ttpro313.zip archive. The Stack180 file is a keyboard definition file for TeraTerm.

CDRomRoot:\Stack Design .PDF Data Files

Various PDF files detailing RAM Expansion, Schematics, this User Manual, the LCD Adapter, and a short tutorial on configuring the XPORT to provide a link to the LAN-based printer for use by CP/M.

CDRomRoot:\CPMDATA Subdirectory

The CPMDATA subdirectory is an archive of the entire hard drive contents of the Stack180. It is broken into two main archives, based on the operating system: STACKOS or STACKBP. A third section off of the main archive is the EPROM subdirectory (see below.)

STACKOS - The StackOS archive on partitions A:, B:, C:, D:, and E:.

STACKBP - The B/P BIOS archive on partitions A:, B:, C:, D:, and E:.

The two operating systems are on separate hard drives, so there is no overlap with the various partitions. In both cases, partition A: is the boot drive - it contains the operating system boot code, system utilities, help files, and all OS source code. Partition B: holds the source code to several Stack-specific utilities. Partitions C:, D:, and E: hold numerous archive libraries with a large assortment of languages, compilers, utilities, BDOS replacements, etc. In particular, a very large library of Z-System utilities is also present.

The archives consist of three basic types of files, each in its own very specific subdirectory path:

Install Images

Boot partition installation images are used by the EEPROM Loader and the USBWiz to create a boot partition on an empty hard drive, or to overwrite an existing boot partition with a new one. The installation images are located at the following paths:

CPMDATA\STACKOS\INSTALL\INSTALL.IMG for Stack OS, or
CPMDATA\STACKBP\INSTALL\INSTALL.IMG for B/P BIOS.

Disk Image Files

These files differ from Install images in that they are NOT associated with a boot partition. Instead, they are images of non-boot partitions and can be written to an empty partition through the use of the BACKUP program carried in A15: of the boot drive. BACKUP is also used to create the images.

CPMDATA\STACKOS\IMAGE\DRIVEx.IMG, where 'x' is B, C, D, etc (for STACKOS), or
CPMDATA\STACKBP\IMAGE\DRIVEx.IMG for B/P BIOS.

CDROM Contents (cont.)

Individual Files

Finally, individual files are stored in the archive via the BACKUP program:

CPMDATA\STACKOS\FILES\DRIVEx\USERxxx\filename.ext for Stack OS, or
CPMDATA\STACKBP\FILES\DRIVEx\USERxxx\fielname.ext for B/P BIOS.

Each Drive and User area has a separate subdirectory, as shown in the paths above with the lowercase 'x' characters, where drive B: would use a path subdirectory of DRIVEB and User 5 would have a subdirectory of USER005.

The STACKOS and STACKBP subdirectory names in BACKUP can be modified with the setup option from the primary menu, so you can configure the program for different paths based on the OS or machine name.

Although BACKUP can write individual files to the backup medium, it is not yet capable of restoring files from the medium to the hard drive. This is not the case for image files.

EEPROM Files

The EEPROM loader file is stored in the CPMDATA archive at CPMDATA\STKEPROM\EEPROM.BIN. The EEPROM Loader utility can read the EEPROM.BIN file and reprogram the entire EEPROM in-circuit. Although burning a new EEPROM in-circuit is considered a 'risky' operation, there has not yet been a single EEPROM returned to TG Consulting for a re-burn.

As the EEPROM code is still being updated fairly often, this process provides and excellent means by which the user can update the EEPROM on the fly, in just a few minutes. Updated EEPROM.BIN files are placed in the Download area at the TG Consulting web site whenever a new code file is generated.

An option of the Loader utility is to save the EEPROM contents to the USBWiz storage medium (SD Card or Thumb Drive). You can also save the EEPROM Setup code block, or burn the Setup Code block from a file to the EEPROM.

(The Setup Code Block is 256 bytes in the EEPROM that stores all of the Setup data tables and system variables. Usually, an EEPROM update will not require you to overwrite the Setup Code Block, which would force you to go through the entire setup process again. Occasionally, however, you will have to overwrite the Setup code block and re-start the entire Setup process. By saving the Setup Data Block to USBWiz storage, you can, if needed, quickly and easily restore the Setup code block to its previous state.)

Appendix A - EEPROM POST Codes

EEPROM POST codes are displayed on the HEX LED during EEPROM operation.

```
;---------------------------------------------------------------------
; POST Code Equates
;
;   These values are sent to the LED display at various times,
; allowing the user to determine where problems occur and to track
; the progress of the various processes.
;---------------------------------------------------------------------
Post$PwrUp        equ   0ffh  ; Displayed on power up
Post$AllOk        equ   00h   ; All OK - Init complete

;-------------------------------------
; Test sequence Post Codes
Post$TestInit     equ   01h   ; Entering Test code
Post$TestLp       equ   02h   ; Test menu loop
Post$MemTest      equ   03h   ; Entering memory test routine
Post$SwSet        equ   04h   ; Display Switch Settings
Post$Gide0M       equ   05h   ; Testing GIDE0 Master
Post$Gide0S       equ   06h   ; Testing GIDE0 Slave
Post$Gide1M       equ   07h   ; Testing GIDE1 Master
Post$Gide1S       equ   08h   ; Testing GIDE1 Slave
Post$PcIde4M      equ   09h   ; Testing PcIde4 Master
Post$PcIde4S      equ   0ah   ; Testing PcIde4 Slave
Post$PcIde3M      equ   0bh   ; Testing PcIde3 Master
Post$PcIde3S      equ   0ch   ; Testing PcIde3 Slave
Post$TestExit     equ   0fh   ; Exiting Test code

;-------------------------------------
; Setup sequence Post Code
Post$SetupEnt     equ   10h   ; Entering Setup code
Post$CpuParms     equ   11h   ; Enter CPU Parameter Setup
Post$TermParms    equ   12h   ; Entering Terminal Setup
Post$MoBoParms    equ   13h   ; Entering MoBo Setup
Post$GideParms    equ   14h   ; Entering GIDE setup
Post$SlotParms    equ   15h   ; Entering Slot setup
Post$EepromWr     equ   1fh   ; Writing EEPROM

;-------------------------------------
; Boot Sequence Post Codes (all boot devices)
; GIDE Boot Post Codes
Post$EntGideBt    equ   20h   ; Entering Gide Boot Sequence
Post$GideInit     equ   21h   ; Entering GIDE init sequence

; PcIde Boot Post Codes
Post$EntPcIdeBt   equ   23h   ; Entering PcIde Boot Sequence
Post$PcIdeInit    equ   24h   ; Entering PcIde Init sequence

; Floppy Boot Post Codes
Post$EntPcFlpBt   equ   26h   ; Entering PcFloppy boot sequence
Post$PcFlpInit    equ   27h   ; Entering Floppy init sequence

; AtApi Boot sequence Post Codes
Post$EntAtBt      equ   29h   ; Entering PcAtApi Boot sequence
Post$AtInit       equ   2ah   ; Entering AtApi Init sequence
```

Appendix A - EEPROM POST Codes (cont)

```
; Generic codes for all boot drivers
Post$SysValid     equ   2eh   ; Generic 'System Valid' Post Code
Post$Booting      equ   2fh   ; Generic 'Booting' Post Code

;-------------------------------------
; System Init sequence codes
Post$CpuInit      equ   0c0h  ; Entering CPU Initialization routine
Post$BrgInit      equ   0c1h  ; Entering BRG Init routine
Post$BrgComp      equ   0c2h  ; BRG Init Complete
Post$RamChk       equ   0c3h  ; Init RAM test
Post$KbdDelay     equ   0c4h  ; 5 second kbd Delay
Post$ResHdwr      equ   0c5h  ; Entering Reset Hardware routine
Post$PollPG       equ   0c6h  ; Polling Primary Gide
Post$PollSG       equ   0c7h  ; Polling Secondary Gide
Post$CLI          equ   0c8h  ; Entering XPORT CLI routine
Post$CLIL         equ   0c9h  ; CLI Loop in execution
Post$PollPc3      equ   0cah  ; Polling Slot3 PcIde Master
Post$PollPc4      equ   0cbh  ; Polling Slot4 PcIde Master

;-------------------------------------
; Serial Test sequence codes
Post$EntSerTest   equ   0d0h  ; Entering Serial Test Menu
Post$EntATst      equ   0d1h  ; Testing Serial A
Post$Ent31Tst     equ   0d2h  ; Testing PcCom1, Slot 3
Post$Ent32Tst     equ   0d3h  ; Testing PcCom2, Slot 3
Post$Ent41Tst     equ   0d4h  ; Testing PcCom1, Slot 4
Post$Ent42Tst     equ   0d5h  ; Testing PcCom2, Slot 4
Post$SndAT        equ   0d6h  ; Sending 'A' from Test port
Post$RcvAD        equ   0d7h  ; Receiving 'A' from Port D
Post$SndAD        equ   0d8h  ; Sending 'A' from Port D
Post$RcvAT        equ   0d9h  ; Receiving 'A' from Test port

;-------------------------------------
; Error Codes (Generic and Controller-specific)
Post$NoSys        equ   0f0h  ; Error - No System on boot disk
Post$IdeErr       equ   0f1h  ; Error - IDE Controller I/O error
Post$FlopErr      equ   0f2h  ; Error - PcFloppy Controller I/O error
Post$IdeTmOut     equ   0f3h  ; Error - IDE Timeout
Post$FlopTmOut    equ   0f4h  ; Error - PcFloppy Timeout
```

Appendix B - EEPROM Parameter Passing

The EEPROM Setup code is used to build an information table that is passed to CP/M immediately before execution is passed to the Cold Boot BIOS. The following table is taken directly from the EEPROM source code:

```
; Each drive type has a 'Boot Code' assigned, ranging from 1 to N, where:
; PcFlop4Code = 1 = PcFloppy, Slot4 Boot
; PcFlop3Code = 2 = PcFloppy, Slot3 Boot
; AtZipCode   = 3 = AtApi ZIP Boot
; PcIde4Code  = 4 = PcIde4 Boot
; PcIde3Code  = 5 = PcIde3 Boot
; PriGideCode = 6 = Primary GIDE Boot
; SecGideCode = 7 = Secondary GIDE Boot;
;
; Enter the Operating System with the following table in RAM:
;
;     Location    Info
;     0013H       Boot Port Select Code byte
;     0014H       StackOS Default IDE Drive
;     0015H       StackOS Default/Boot device (Master/Slave)
;     0016H       B/P BIOS Default IDE Port
;     0017H       B/P BIOS Default/Actual boot device (Master/Slave)
;
;     001BH       Console Re-director byte
;     001CH       AtApi Port Base address (+1 for Slave)
;     001DH       Floppy Base Port (Slot) address
;     001EH       LPT/COM Base Port (Slot3Base or Slot4Base)
;     001FH       CDROM Base Port Address (+1 for Slave)
;
;     0023H       UsbWiz Default Port
;     0024H       USBWiz Default LUN
;     0025H       # of Main RAM Banks
;     0026H       # of Banked RAM Banks
;     0027H       (SysParam) byte
;
;     002BH       LCD Interface Status
```

The data table is used to inform the BIOS of various system operating parameters. For instance, byte (0013H) is used to report the location/type of the boot drive, while byte (001BH) informs the BIOS which terminal port is being used.

The AtApi and CDROM Base Port Addresses (001CH and 001FH, respectively) use a +1 offset if the drive is the Slave. For instance, if the AtApi (ZIP) drive is the Master on Slot4 and the CDROM is the Slave on Slot4, then 0D8H (Slot4 base address) is passed for the AtApi address and 0D9H is passed for the CDROM address.

Appendix C - Slot Expansion Connectors

The four expansion slots come in two types: Stack180 Expansion in Slots 1 & 2, and PC-AT Expansion in Slots 3 & 4.

Slot 1 and Slot 2

The Slot 1/2 connectors provide access to all Address, Data, and Control signals on the Stack180 mainboard. They are nearly identical - address lines A8-A15 are different for each of the two slots. In the table below, this is shown as SxA15 for pin 29; replace the 'x' with 1 or 2, depending on the Slot in use. Thus, A15 on Pin 29 is S1A15 for Slot1, and S2A15 for Slot2. This only applies to the A8-A15 address lines; the A0-A7 address lines are identical and driven by the CPU. They are labeled SA0 for Slot A0, SA1 for Slot A1, etc.

Pin#	Signal	Pin#	Signal	Pin#	Signal
1	+3.3 VDC	21	/XPortReset	41	SxA9
2	Gnd	22	/Wait	42	/Int0L0
3	MBD7	23	+5 VDC	43	SxA8
4	BResetH	24	/DREQ0	44	+5 VSB
5	MBD6	25	Gnd	45	SA7
6	+5 VDC	26	/TEND0	46	Int0L14H
7	MBD5	27	+3.3 VDC	47	SA6
8	/BReset	28	/DREQ1	48	BE-Clk
9	MBD4	29	SxA15	49	SA5
10	-5 VDC	30	/TEND1	50	+5 VDC
11	MBD3	31	SxA14	51	SA4
12	-12 VDC	32	BPHI Clk	52	Int0L7H
13	MDB2	33	SxA13	53	SA3
14	+12 VDC	34	Int0L9H	54	Int0L5H
15	MBD1	35	SxA12	55	SA2
16	*BIoWr	36	/Int0L13	56	+5 VDC
17	MBD0	37	SxA11	57	SA1
18	*BIoRd	38	/Int0L12	58	+3.3 VSB
19	+3.3 VDC	39	SxA10	59	SA0
20	Gnd	40	/Int0L1	60	Gnd

The mainboard does not provide access to the A8-A19 address signals. Although A8-A10 are present on the mainboard, they are not connected to anything. Slots 1 and 2, however, may need to use the higher address lines for some reason. If this is the case, you can set the desired A8-A15 address bits by writing them to port 0F1H for Slot1 and port 0F2H for Slot2. The A0-A7 Slot Address lines are driven by the CPU address bus.

Mainboard Data Bus signals are only active during mainboard I/O; they are labeled MBD0-MBD7.

The Buffered I/O Read and Write signals (*BIoRd and *BIoWr) are created as result of a logical OR of the CPU's (/RD and /IORQ) and (/WR and /IORQ) signals. Thus, *BIoRd will go active (low) when both /RD and /IORQ are active (low); *BIoWr will go active (low) when both /WR and /IORQ are active (low.)

Slot 3 and Slot 4

As close as reasonable, Slot3 and Slot4 are identical to the PC-AT 16 bit slots. In terms of power, address, data, and control signals, they are identical. DMA and interrupts are handled somewhat differently. In all cases, the labels 'In' and 'Out' reference the mainboard.

Appendix C - Slot Expansion Connectors (cont)

A1	*I/O Channel Check (In)	(open)	B1	Gnd	
A2	D7 (In/Out)		B2	BResetH (Out)	
A3	D6 (In/Out)		B3	+5 Vdc	
A4	D5 (In/Out)		B4	IRQ9 (In)	(Intx)
A5	D4 (In/Out)		B5	-5 Vdc	
A6	D3 (In/Out)		B6	DRQ2 (In)	(DREQ0)
A7	D2 (In/Out)		B7	-12 Vdc	
A8	D1 (In/Out)		B8	*CARD SEL (In)	(open)
A9	D0 (In/Out)		B9	+12 Vdc	
A10	*I/O Channel Ready (In)	(open)	B10	Gnd	
A11	AENx (Out)(HI if DMA owns bus)		B11	*MEMWR (Out)	(+5)
A12	SA19 (Out)	(Gnd)	B12	*MEMRD (Out)	(+5)
A13	SA18 (Out)	(Gnd)	B13	*IOWR (Out)	(*SlotxWr)
A14	SA17 (Out)	(Gnd)	B14	*IORD (Out)	(*SlotxRd)
A15	SA16 (Out)	(Gnd)	B15	*DACK3 (Out)	(/DACKx)
A16	SA15 (Out)	(Gnd)	B16	DRQ3 (In)	(DREQ0)
A17	SA14 (Out)	(Gnd)	B17	*DACK1 (Out)	(/DACKx)
A18	SA13 (Out)	(Gnd)	B18	DRQ1 (In)	(DREQ0)
A19	SA12 (Out)	(Gnd)	B19	*REFRESH (In/Out)	(open)
A20	SA11 (Out)	(Gnd)	B20	Sys Clock (Out)	(BPHI Clk)
A21	SxA10 (Out)	(Latched)	B21	IRQ 7 (In)	(LptIntH)
A22	SxA9 (Out)	(Latched)	B22	IRQ 6 (In)	(FlopIntH)
A23	SxA8 (Out)	(Latched)	B23	IRQ 5 (In)	(Intx)
A24	SxA7 (Out)	(Latched)	B24	IRQ 4 (In)	(Int0L4H)
A25	SxA6 (Out)	(Latched)	B25	IRQ 3 (In)	(Int0L3H)
A26	SxA5 (Out)	(Latched)	B26	*DACK 2 (Out)	(/DACKx)
A27	SxA4 (Out)	(Latched)	B27	T/C (Out)	(TEnd0)
A28	SxA3 (Out)	(Latched)	B28	ALE (Out)	(+5)
A29	SA2 (Out)		B29	+5 Vdc	
A30	SA1 (Out)		B30	OSC (Out)	(open)
A31	SA0 (Out)		B31	Gnd	

C1	SBHE (In/Out)(Sys Bus HI En)	(open)	D1	*MEM CS16 (In)	(open)
C2	LA23 (In/Out)	(open)	D2	*I/O CS16 (In)	(open)
C3	LA22 (In/Out)	(open)	D3	IRQ 10 (In)	(Intx)
C4	LA21 (In/Out)	(open)	D4	IRQ 11 (In)	(Intx)
C5	LA20 (In/Out)	(open)	D5	IRQ 12 (In)	(open)
C6	LA19 (In/Out)	(open)	D6	IRQ 15 (In)	(Ide1IntH)
C7	LA18 (In/Out)	(open)	D7	IRQ 14 (In)	(Ide0IntH)
C8	LA17 (In/Out)	(open)	D8	*DACK 0 (Out)	(/DACKx)
C9	*MEMRD (In/Out)	(open)	D9	DRQ 0 (In)	(DREQ0)
C10	*MEMWR (In/Out)	(open)	D10	*DACK 5 (Out)	(/DACKx)
C11	D8 (In/Out)		D11	DRQ 5 (In)	(DREQ0)
C12	D9 (In/Out)		D12	*DACK 6 (Out)	(/DACKx)
C13	D10 (In/Out)		D13	DRQ 6 (In)	(DREQ0)
C14	D11 (In/Out)		D14	*DACK 7 (Out)	(/DACKx)
C15	D12 (In/Out)		D15	DRQ 7 (In)	(DREQ0)
C16	D13 (In/Out)		D16	+5 Vdc	
C17	D14 (In/Out)		D17	*MASTER (In)	(open)
C18	D15 (In/Out)		D18	Gnd	

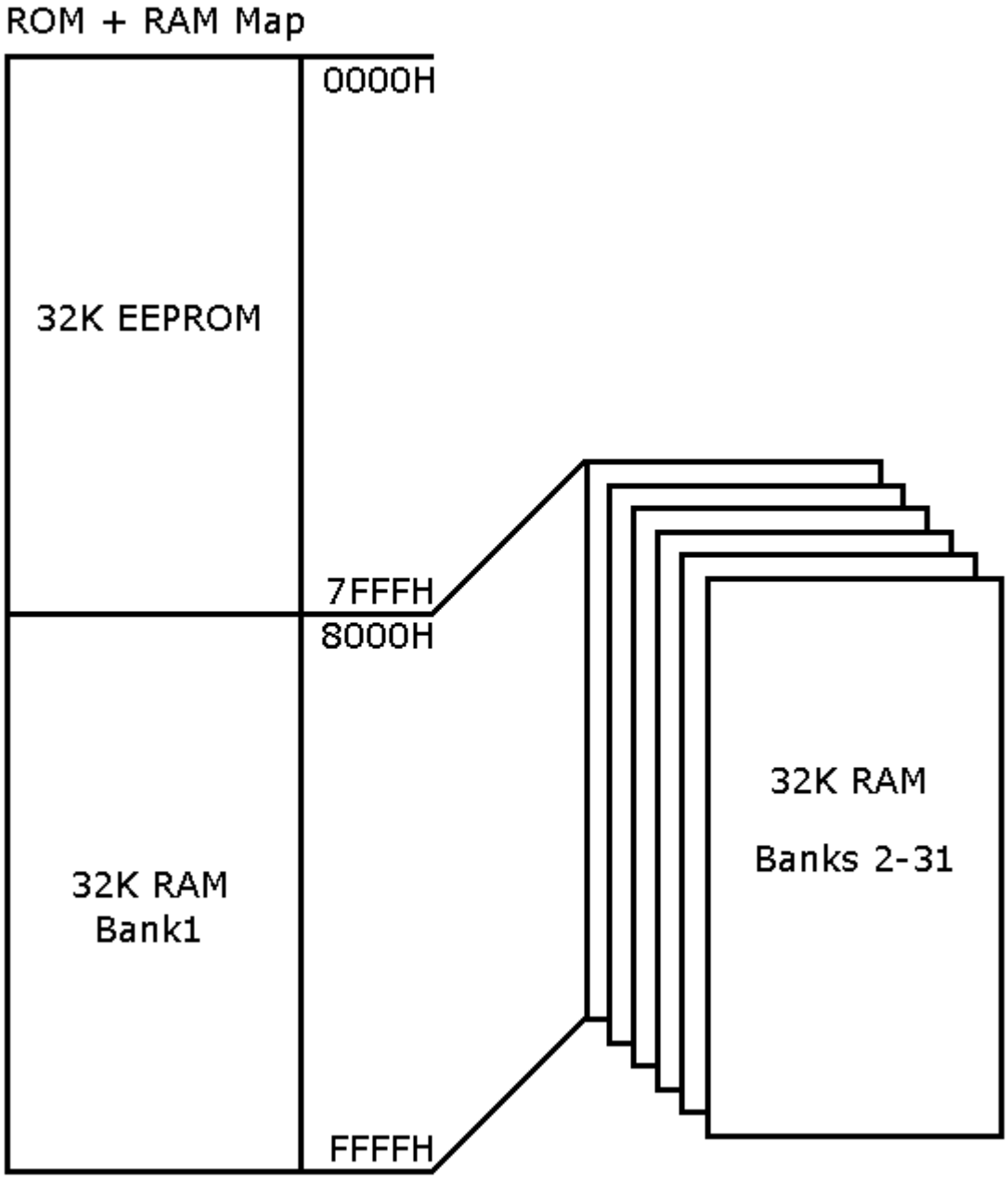
Many of the pins are not used in the current implementation. Note the pins labeled "(open)." These pins have nothing connected. A few pins have labels, but are tied either to (Gnd) or (+5). This is to ensure all I/O board input pins are at a certain logic level that means, essentially, "ignore."

Appendix D - Memory Map

The Stack-180 uses two different memory maps: one when the EEPROM program is in control, and a different map after booting into CP/M. In both cases, a 32K 'window' is used to provide access to different 32K RAM banks.

EEPROM Program Memory Map

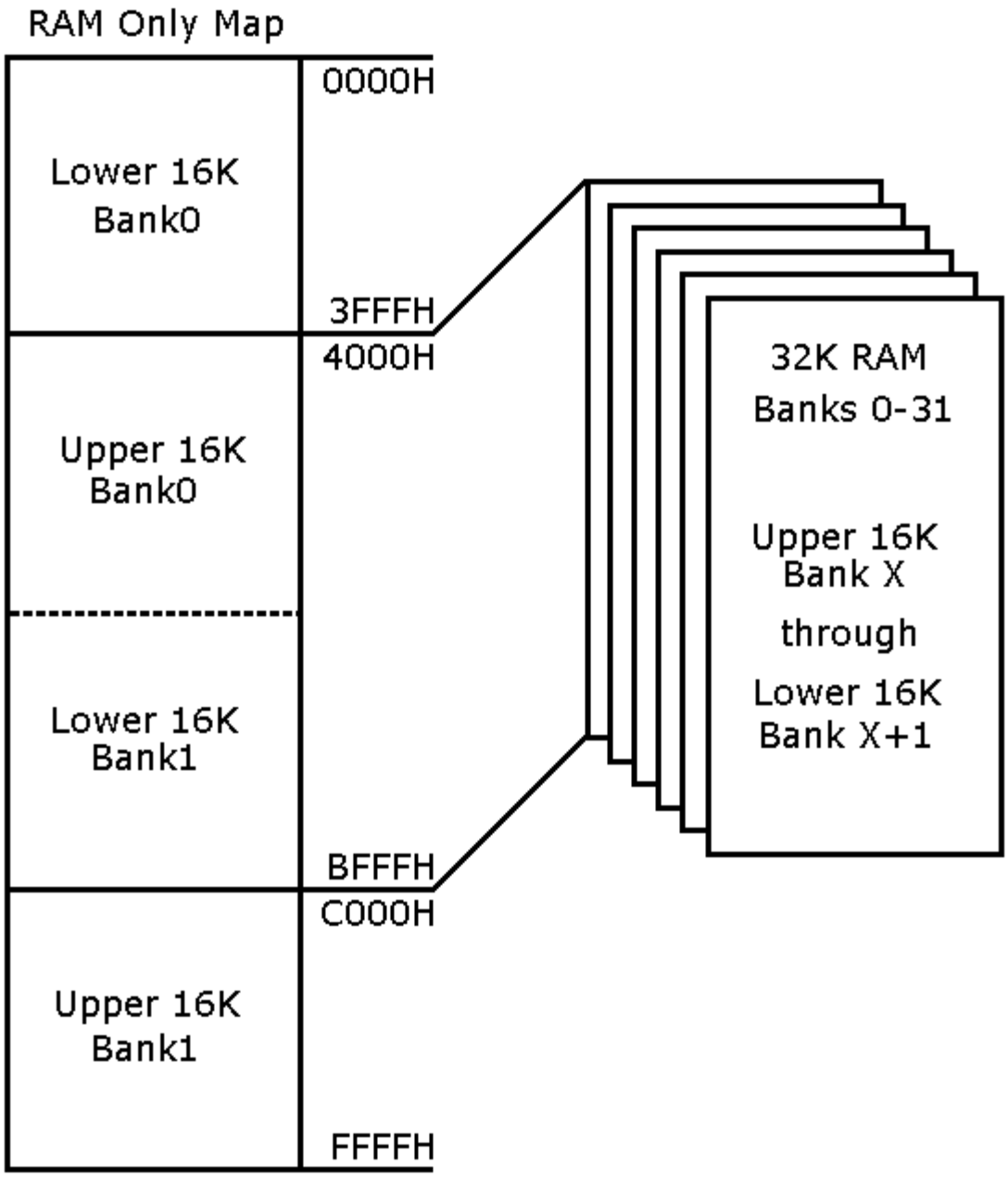
The EEPROM memory map consists of the 32K EEPROM and a 32K bank of RAM, for a total of 64K. RAM Bank 0 is out of context (replaced by the EEPROM); Banks 1-31 can be selectively placed in context from 8000H-FFFFH.



Appendix D - Memory Map (cont)

CP/M Memory Map

The memory map changes when CP/M is booted. The EEPROM is taken out of context and replaced by RAM Bank0. The 32K memory bank 'window' is shifted from the original 8000H-FFFFH area to the middle of the 64K RAM, from 4000H-BFFFH. Since CP/M resides in high memory (and 'owns' Page0 from 0000H-00FFH), this map allows the operating system to remain resident at all times, regardless of the current banking.



Appendix D - Memory Map (cont)

Extended RAM Banks

In addition to the Z8S180's MMU 1MB Base RAM banking layout shown in the above memory maps, there is also an 'extended' set of RAM banks available through the addition of the 2MB RAM Expansion option board.

The primary 1MB RAM subsystem consists of two, 512K RAM chips, labeled the LOW and HIGH RAM chips. The HIGH RAM chip (Extended RAM, Bank 0,) occupying RAM addresses 80000H-FFFFFH, can be swapped out of context and an 'extended RAM' chip placed in context. The coding of the memory map decoder GAL (U5 on the CPU board) allows for an additional seven, 512K extended RAM chips (3.5MB) to be placed in context. However, due to space constraints, the RAM expansion adapter was limited to 2MB (four, 512K chips, labeled Extended RAM, Banks 1 - 4.)

To determine which Extended RAM Bank is in context, read port 0F8H and mask out bits 3-7:

```
      in    a,(GpPort)  ; Read 0f8h port
      and   7           ; Mask out bits 3-7
```

Whatever value is now in register A (0-7) is the current Extended RAM Bank, where 0 (Bank 0) is the 512K HIGH RAM on the CPU board, and 1-7 (Banks 1-7) are the Extended Banks.

You can place the desired RAM Bank in context by writing to GpPort, but care MUST be taken to ensure you do not disturb the status of bit 7 in GpPort. For instance:

```
; Enter with new Extended RAM Bank select value (0-7) in register B
      in    a,(GpPort)  ; Get current control port status byte
      and   10000000b   ; Save RAM Map Select bit 7
                        ;  Mask all other bits to 0
      add   a,b         ; Add new Extended RAM Bank select value
      out   (GpPort),a  ; Place new bank in context
```

With the 2MB RAM Expansion Adapter, the Banked RAM selection is limited to Banks 0-4, rather than Banks 0-7, although there is nothing preventing you from selecting Banks 5-7. However, doing so will result in no HIGH RAM (80000H-FFFFFH) being in context.

The EEPROM start-up code performs a quick check of RAM capacity, reporting the number of 32K RAM Banks in the 1MB Base RAM, and the number of 32K RAM Banks in the Extend RAM. A full load of 1MB Base RAM and 2MB of Extended RAM will result in a report showing 32 Banks of 32K in Base RAM and 64 Banks of 32K in Extended RAM.

The EEPROM-based RAM Test (Testing Menu, Option A) will test all available RAM, including the Extended RAM, as reported during the start-up process.

StackOS can use the Extended RAM Banks via the High RAM RamDisk driver by setting the RamDisk and ExtRamDisk equates to TRUE before compiling the BIOS, resulting in a 1MB RamDisk in the Extended RAM. Both equates are in the A3:BIOSZ.Z80 file. Setting ExtRamDisk to FALSE will place a 512K RamDisk in the main 1MB RAM.

B/P BIOS does not have an equivalent setting for the RamDisk - it is limited to the 1MB Main RAM.

Appendix E - I/O Map

0000H – 003FH	**Reserved for Z8S180 CPU**	
0000H	CNTLA0	ASCI Control Register A0
0001H	CNTLA1	ASCI Control Register A1
0002H	CNTLB0	ASCI Control Register B0
0003H	CNTLB1	ASCI Control Register B1
0004H	STAT0	ASCI Status Register Channel A
0005H	STAT1	ASCI Status Register Channel B
0006H	TDR0	ASCI-0 Transmit Data Register
0007H	TDR1	ASCI-1 Transmit Data Register
0008H	RDR0	ASCI-0 Receive Data Register
0009H	RDR1	ASCI-1 Receive Data Register
000AH	CNTR	CSI/O Control Register
000BH	TRDR	CSI/O Trans/Rcv Data Register
000CH	TMDR0L	Timer Data Register 0L
000DH	TMDR0H	Timer Data Register 0H
000EH	RLDR0L	Timer Reload Register 0L
000FH	RLDR0H	Timer Reload Register 0H
0010H	TCR	Timer Control Register
0011H		
0012H	ASEXT0	ASCI Extension Control Reg A
0013H	ASEXT1	ASCI Extension Control Reg B
0014H	TMDR1L	Timer Data Register 1L
0015H	TMDR1H	Timer Data Register 1H
0016H	RLDR1L	Timer Reload Register 1L
0017H	RLDR1H	Timer Reload Register 1H
0018H	(R/O) FRC	Free Running Counter
0019H		
001AH	ASTC0L	ASCI Time Constant Register 0L
001BH	ASTC0H	ASCI Time Constant Register 0H
001CH	ASTC1L	ASCI Time Constant Register 1L
001DH	ASTC1H	ASCI Time Constant Register 1H
001EH	CMR	Clock Multiplier Register
001FH	CCR	CPU Control Register
0020H	SAR0L	DMA Source Address Register, Channel 0L
0021H	SAR0H	DMA Source Address Register, Channel 0H
0022H	SAR0B	DMA Source Address Register, Channel 0B
0023H	DAR0L	DMA Destination Add Reg, Channel 0L
0024H	DAR0H	DMA Destination Add Reg, Channel 0H
0025H	DAR0B	DMA Destination Add Reg, Channel 0B
0026H	BCR0L	DMA Byte Count Register, Channel 0L
0027H	BCR0H	DMA Byte Count Register, Channel 0H
0028H	MAR1L	DMA Memory Address Register, Channel 1L
0029H	MAR1H	DMA Memory Address Register, Channel 1H
002AH	MAR1B	DMA Memory Address Register, Channel 1B
002BH	IAR1L	DMA I/O Address Register Channel 1L
002CH	IAR1H	DMA I/O Address Register Channel 1H
002DH	IAR1B	DMA I/O Address MS Byte Register
002EH	BCR1L	DMA Byte Count Register, Channel 1L
002FH	BCR1H	DMA Byte Count Register, Channel 1H
0030H	DSTAT	DMA Status Register
0031H	DMODE	DMA Mode Register
0032H	DCNTL	DMA/WAIT Control Register
0033H	IVRLO	Interrupt Vector Low Register
0034H	ITC	Interrupt/Trap Control Register
0035H		
0036H	RCR	Refresh Control Register
0037H		

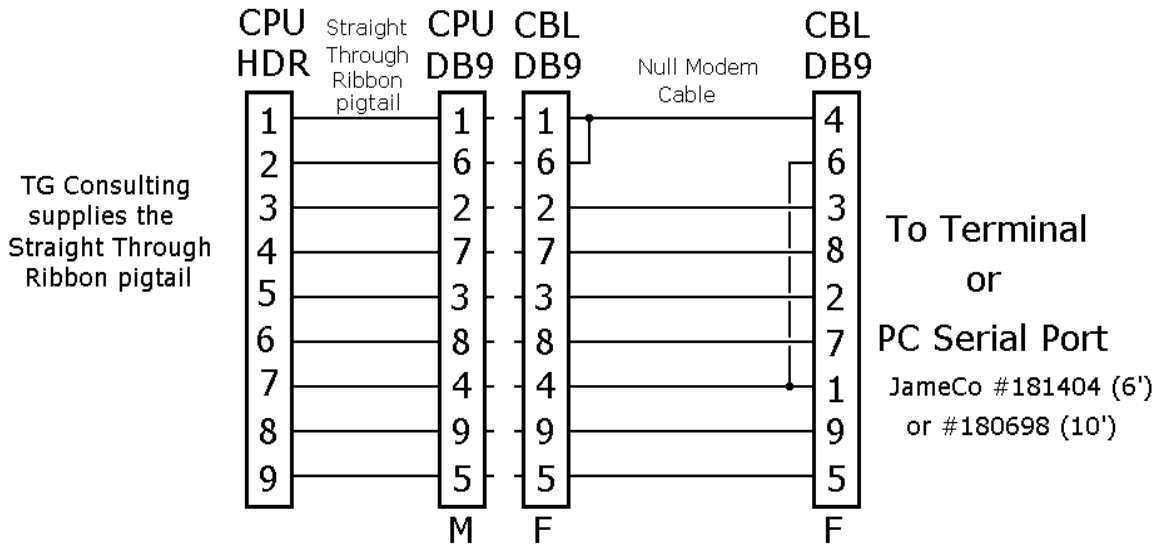
Appendix E - I/O Map (cont)

```
      0038H       CBR         MMU Common Base Register
      0039H       BBR         MMU Bank Base Register
      003AH       CBAR        MMU Common/Bank Area Register
      003BH
      003CH
      003DH
      003EH       OMCR        Operating Mode Control Register
      003FH       ICR         I/O Control Register

0040H – 004FH     Secondary GIDE Adapter
0050H – 005FH     Primary GIDE Adapter
0060H – 0067H     Mainboard Serial Port A (DTE)
0068H – 006FH     Mainboard Serial Port B (USBWiz or ALFAT USB)
0070H – 0077H     Mainboard Serial Port C (XPORT Line 2)
0078H – 007FH     Mainboard Serial Port D (DCE)
0080H – 0087H     Slot1/2 Adapter: WIZNet 4-Port LAN, Port A
0088H – 008FH     Slot1/2 Adapter: WIZNet 4-Port LAN, Port B
0090H – 0097H     Slot1/2 Adapter: WIZNet 4-Port LAN, Port C
0098H – 009FH     Slot1/2 Adapter: WIZNet 4-Port LAN, Port D

00A0H – 00B7H     Available, but not decoded

00B8H - 00BFH     Slot5 Port Range (experimental mainboard only)
00C0H - 00C7H     Slot6 Port Range (experimental mainboard only)
00C8H - 00CFH     Slot7 Port Range (experimental mainboard only)
00D0H - 00D7H     Slot3 Port Range
00D8H - 00DFH     Slot4 Port Range
00E0H – 00EFH     Epson 72421 Real Time Clock

00F0H Read        Mainboard Switch Port
00F0H Write       HEX LED output
00F1H Read        Signal Monitor Port
00F1H Write       Slot 1 Address Port
00F2H Read        Available, not decoded
00F2H Write       Slot 2 Address Port
00F3H Read        Available, not decoded
00F3H Write       Slot 3 Address Port
00F4H Read        Available, not decoded
00F4H Write       Slot 4 Address Port

00F5H Rd/Wr       PC-Slot HIGH data byte (D8-D15)
00F6H Rd/Wr       Slot3 DMA (Slots 3/5/7 on experimental mainboard)
00F7H Rd/Wr       Slot4 DMA (Slots 4/5 on experimental mainboard)
00F8H Rd/Wr       General Purpose Control Port

00F9H Rd          Available, not decoded
00F9H Wr          Available (Slot5 Address Port on experimental mainboard)
00FAH Rd          Available, not decoded
00FAH Wr          Available (Slot6 Address Port on experimental mainboard)
00FBH Wr          Available (Slot7 Address Port on experimental mainboard)

00FBH Rd          Slot1/2 Adapter: WIZNet 4-Port LAN Status (Rd only)
00FCH-00FFH       Slot1/2 Adapter: LCD Display w/8-bit LED
```

Appendix F - Terminal Serial Connections

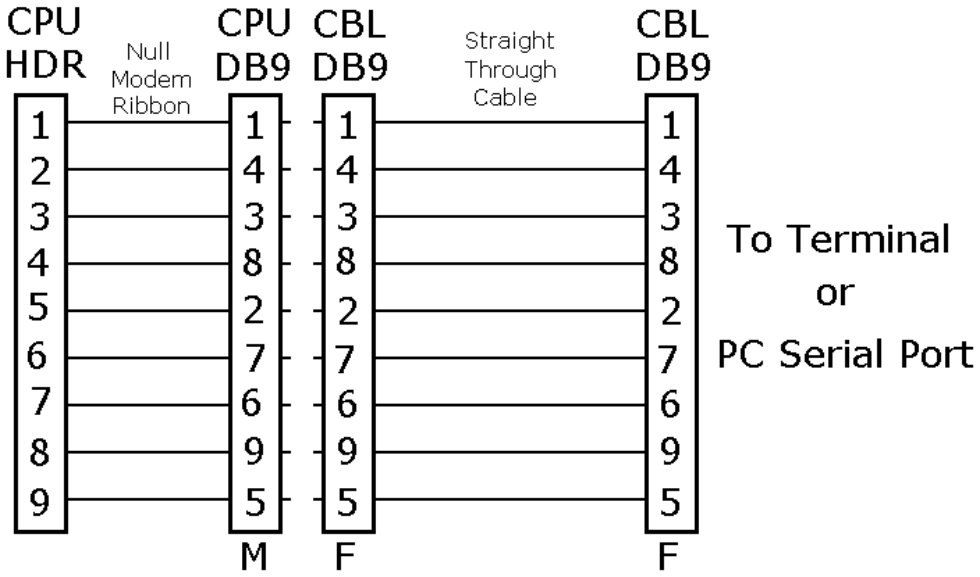
The Stack-180 CPU board is supplied with a 'straight thru' pigtail connector that brings the CPU terminal port to the rear of the case with a male 9-pin connector in a DTE (Data Terminal Equipment) format.

Your typical terminal or PC serial port is also in a DTE format, so a 'null modem' cable connection will be required. Below is a diagram that shows typical connections with a 'straight thru' pigtail and a null modem cable:



A very simple, three-wire cable (TxD, RxD & Gnd) will allow the Stack to operate, but the baud rate will be limited to something less than 38,400 baud. With the full handshaking cable setup shown above, maximum baud rates of 115,200 can be delivered while using hardware flow control.

An alternate configuration would be to use a 'null modem' pigtail with a straight thru cable:



Appendix G – About the XPORT AR

The XPORT AR is marketed by Lantronics. It is normally programmed via DeviceInstaller, a Windows software product on the Stack CDROM in the Stack180\Design Files\XPORT AR\ subdirectory. It can also be programed manually via the Command Line Interface (CLI) via either the Windows LAN connection or the Stack serial connection.

The XPORT actually has two serial connections to the Stack180: ASCI-0 (Line1) and Serial-B (Line2). A limitation of the XPORT is that each serial connection can only perform one of two basic functions at any given time: it can 'tunnel' to a remote IP address, or it can be used to connect to the Command Line Interface (CLI) within the XPORT. Under normal operation, the two serial connections operate in 'Tunnel' mode to provide LAN-based IP connections to the Terminal and the Printer, meaning the CLI is normally not connected to the Stack serial ports.

Unfortunately, without the CLI connection, you cannot re-direct the XPORT to a different IP address or change any operating parameters. This means that, with the normal implementation, the XPORT cannot be controlled 'on-the-fly' from the Stack180.

It IS possible, however, to have the Stack180 power up in CLI mode if the Stack is connected to the local serial terminal via ASCI-1. In other words, when using a local, serial terminal, you can power up with the XPORT in CLI mode, connected to the Stack via ASCI-0. Before power-up, set SW1-1 to OFF and SW1-4 to ON. When you power up, Terminal I/O will be via ASCI-1 (local serial) and the XPORT CLI will be attached to ASCI-0. You can then, via Cisco-like command strings, reprogram the various XPORT operating parameters. See the XPORT "Command Reference" .PDF document for details.

Once CLI programming via the Stack is complete, you can disable the CLI interface by setting SW1-4 to OFF and restore SW1-1 to the desired setting (Off=Local serial terminal, ON=LAN-based terminal.) Restart the Stack to return to normal operation.

A 'near on-line' method of CLI access is available via the TELNET program used to provide the LAN-based Terminal. The Stack can be configured to either LAN-based or the local serial terminal. Simply open a new TELNET session and connect to the Stack's IP address via Port 23 – this provides a TELNET connection to the CLI. Alternately, you can also use your Internet browser to connect to the Stack's IP address via Port 80 – this opens a hypertext link to the XPORT and provides the easiest method of re-programming the XPORT.

Although not a real-time control link, the Hypertext or TELNET linkages provide the next best thing.

Appendix H - Accessory Boards and Adapters

There are several add-on adapter boards and kits available to the Stack:

- 2MB Memory Expansion. System RAM can be expanded to a total of 3MB with the addition of the 2MB Memory Expansion daughterboard. The EEPROM and Memory Decoder chip are removed from the CPU board and installed on the Memory Expansion board, which is then installed on the CPU board. The additional 2MB consists of four, 512K RAM chips that are 'banked' in place of the upper 512K RAM chip on the CPU board. StackOS has a RamDisk driver that uses 1MB of the expansion; no other use for the addition RAM currently exists. The EEPROM code will detect and test the full 2MB of expansion RAM. This is not a kit – it is only available assembled and tested.

- XPORT Adapter and LCD I/O. This adapter provides two interfaces: an XPORT connection that allows the user to 'recover' the XPORT firmware when all else has failed, and a parallel connection to a 2x24 LCD display. This adapter installs in either Slot1 or Slot2.

- D-GIDE + SPI SD Card Interface. A 2-port (4 device) 16-bit IDE adapter and dual SD Card connectors for additional storage options. This is a Slot3 or Slot4 adapter.

- 4-Port LAN Connection. Provides four LAN connections to terminals, printers, etc, via a single IP with 4 port addresses. This adapter installs in either Slot1 or Slot2.