



Hardware Reference Manual
Z8000 CPU Board

Central Data Corporation

1602 Newton Drive, Champaign, IL 61820 (217) 359-8010

Z8000 CPU Board

Central Data Corporation
1602 Newton Drive
Champaign, IL 61820
Phone: (217) 359-8010

Copyright (C) 1982, Central Data Corporation

TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
1	<u>General Information</u>	1
2	<u>Functional Description</u>	3
3	<u>Principles of Operation</u>	5
	Block Diagram.....	5
	The Processor.....	5
	Bus Structure.....	7
	EPROM/ROM Capability.....	11
	Programmable Timers.....	11
	Interrupt Capabilities.....	12
	9511 Arithmetic Processor Option.....	14
	I/O Addressing.....	14
	Memory Management.....	15
	I/D Mode Operation.....	18
	Memory Map Setup.....	20
4	<u>Installation/User Selectable Options</u>	22
	System Board Location.....	22
	Bus Straps.....	22
	9511 Straps.....	23
	Bus Timeout Strap.....	23
	BPRO Driver Enable Setup.....	24
	A22,A23 Straps.....	24
5	<u>Specifications</u>	25
6	<u>Schematics</u>	29

1. General Information

The Central Data Z8000 CPU Board incorporates the Zilog/AMD Z8001 16 bit microprocessor chip to form the main computer of any Multibus* system. The Z8000 runs at a speed of 4MHz, and its lines are gated to be completely Multibus compatible. This board also has full multimaster capabilities to allow the user to have several processor boards and DMA devices on the bus at one time.

The board incorporates a unique memory management circuit which divides physical memory into 2K pages. These logical pages are then mapped into physical addresses by the on board circuitry. Through this mapping process the total allowable system-wide memory is 16 megabytes. Since each 2K page of physical memory is mapped from a logical address, these pages need not be contiguous in physical memory. The mapping circuitry also checks for invalid pages to prevent a user from accessing an unmapped area. When a program sets up a map for a particular page, it can determine if it will be write protected. Later, if a program tries to write in a protected area, a trap will occur.

The CPU board can support two 2716 EPROMs to contain any monitor or bootstrap program. These PROMs are automatically selected upon initialization and can be deselected by writing an output port.

Peripheral chips used on the board include an Intel 8253 programmable interval timer, an Intel 8259A programmable interrupt controller, and optionally an AMD 9511 arithmetic processing unit. These three microprocessor support chips are used to significantly decrease the amount of work that the Z8000 has to do. The 8253 allows two timers to be set to generate interrupts. These interrupts can occur from 1us to 65ms after the interval timer is started. The interrupt controller allows the program to set various interrupt priority levels to be associated with the 8 Multibus interrupt lines.

The arithmetic processor is an option on the board which allows the user to have many complex floating point arithmetic operations done outside of the Z8000 CPU chip. The four basic floating point functions are available as well as such functions as SQRT, SIN, COS, TAN, ASIN, ACOS,

ATAN, LOG, LN, EXP, and PWR. The 9511 is operated at a speed of 2MHz. (The clock can be strapped to 4MHz when faster 9511s are available.)

* Multibus is a trademark of Intel Corporation and is used throughout this manual.

2. Functional Description

A detailed description of the Z8000 CPU board's circuitry is given in the Principles of Operation section of this manual. A brief description of the major sections of the board is presented here.

The Central processor for this board is the powerful Zilog/AMD Z8001 16-bit CPU. The processor runs at 4MHz and its architecture includes sixteen 16-bit general purpose registers, a segmented addressing structure, and a versatile instruction set. The processor also allows two modes of operation (System or Normal) which makes the board ideal for multi-user applications.

The board is fully Multibus compatible, which allows it to interface with hundreds of different boards. All bus lines conform with the IEEE Multibus proposed standard. In addition to this definition for P1 (the main connector), Central Data has defined seven pins of P2 (the auxiliary connector) to allow for address expansion, CPU single-stepping capabilities, and a non-maskable interrupt.

Two sockets are provided for up to 4K bytes of non-volatile read only memory. One PROM holds the upper byte of 16-bit words while the other PROM contains the lower byte. On reset, the PROMs are automatically enabled to execute. At any time after reset, the PROMs can be removed from addressable memory by writing to an output port.

There are two programmable timers available on the board. The Intel 8253 Programmable Interval Timer is utilized, so three timers are actually on-board; one is dedicated for use in the monitor for the single step function.

The board accepts 9 interrupt input lines. The highest priority line is NMI (non-maskable interrupt) which enters the board on pin 6 of plug P2. This interrupt cannot be disabled by software control, and it is used mainly for catastrophic failures such as power loss and parity errors. The eight vectored interrupt lines from the Multibus are routed to an Intel 8259A Interrupt Controller. With this controller, a variety of schemes can be implemented for interrupt handling.

As an option, an AMD 9511 Arithmetic Processor can be added to the Z8000 CPU board. This device can significantly increase the processing power of the board if many floating point operations are being performed.

A unique memory management circuit is on the board which is far more powerful than anything found on other microcomputer boards. This circuit allows 16 programs to each have separate memory maps so that they all can appear to have full, unobstructed address spaces consisting of 64: 64K byte segments. Furthermore, each program's segments are divided into 2K byte pages which are the basic units for allocation and management. As a program requires memory, calls to system routines will allocate additional memory to specified segments. Individual pages can optionally be write protected during the system call.

3. Principles of Operation

This chapter details the operation of the entire processor board. Any signal names in this text followed by a slash (/) indicate that the signal is active-low.

As in all Central Data schematics, a grid system is provided to help locate sources and destinations of signals. The source of any named signal will have references to all locations on the schematics where the signal is used. Each location where a signal is used, a reference is given to where it was generated.

If the location is on the same sheet as it is being referenced, it will show only a grid location (i.e. D2). If, however, the referenced signal appears on a separate page, it will have the grid location preceded by the sheet number (i.e. 2-B5).

Furthermore, if a group of signals is commonly routed together, that group may be cross-referenced together. It is not necessary that all members of the group go to each destination listed, since the purpose of the cross-reference system is only to guide a user through the schematics.

Block Diagram

Sheet 1 of the schematics is the block diagram for the Z8000 CPU Board. Each of the major sections of the board is represented on the diagram, and it can be used while reading this manual to see how the different parts of the board interact.

The Processor

The board utilizes a Zilog/AMD Z8001 microprocessor to form the central processor of any Multibus system. Full information concerning the capabilities of the Z8000 processor can be found in the manufacturer's literature for that product.

Sheet 2 of the schematics contains the processor and its associated circuitry. The Z8000 operates with only one (+5V) power supply, and requires a 4MHz clock. To generate this clock, which must swing between strict levels near the

limits of the power supply, an open collector line driver is used (IC38) which is driven by the 4MHz/ output of the clock circuitry. The 82 ohm resistor and 56pf capacitor on the output of this gate make the clock meet the Z8000 specifications, with the resistor pulling the clock line to nearly +5V when the input to the gate is low, and the gate driving the line low whenever its input goes high.

Two interrupt inputs (VI/ and NVI/) as well as the segment trap input (SEGT/) are generated on other pages and run to the processor directly. The non-maskable interrupt input (NMI/) comes from pin 6 of connector P2, which allows external interrupts to the processor for catastrophic system errors (such as power failure or parity errors).

The status lines from the processor (pins 20-23) are used to generate status signals used throughout the board. Whenever ST3 is high, a memory cycle is occurring. Since the Multibus makes no distinction for different types of memory cycles (opcode fetch, stack operation, etc.), the other processor status lines are not used to subdivide the SMEM signal. Whenever a memory cycle is not in progress, IC21 (74S138) is enabled, giving outputs which show the type of cycle in progress. Any outputs marked "N.U." are not used on the board.

Other output lines from the processor are buffered and distributed throughout the board. The WAIT/ input to the processor is driven by a D-type flip-flop (used to guarantee clock setup times) whose input allows the processor to be held up on any of four conditions, if any of the four inputs of the 74H21 AND gate go low.

The first input (GATE COMD) is low only when the board does not have control of the Multibus. The second input (BUS WAIT/) is low whenever, during an access to the Multibus, the slave board has not returned an acknowledge signal to the bus. The third input is used by the 9511 to synchronize any access to it. The last input is used to insert wait-states to guarantee setup times on the Multibus.

Sheet 3 of the schematics details the address/data buffers, CPU clock generator, and the clock state counter. The 74LS273 latches are used to hold the addresses from the AD lines of the processor. The addresses are latched in on the rising edge of MIOAS, which is generated near the bottom of the page. This signal is held low except when a memory or I/O cycle is in progress. During one of these cycles, and when AS/ is low, the 4MHz/ signal clocks the output high.

The 74LS373 tri-state latches are used to hold the data from the Z8000 during write cycles. The data is latched in at the start of T2, with the data enabled to the buffered data (BD) lines whenever the processor is in write mode. To gate data back to the processor, the 74LS244 buffers (IC66, IC69) are used. The buffered data lines are gated to the Z8000 in read mode when data strobe (DS) is high.

The processor clock signals are derived from a crystal oscillator and divider circuit. The oscillator is a feedback network, with the two 1K resistors used to bias the 74S04 inverters into their linear regions. The 100pf capacitor is used to block DC voltage to the crystal, and to provide greater stability. After buffering, the output of the oscillator is divided down to the various frequencies which are needed on the board.

The power-on initialization circuit provides a pulse when power is applied to bring the system up in the proper state. The 1N4148 diode is used to discharge the capacitor quickly when power is turned off.

The clock state generator determines which clock cycle of a machine cycle the board is in. A 74LS161 is used to count the clock cycles, with counting disabled after the tenth cycle has been reached. The counter is cleared with the AS/ signal from the processor, which is active during the first clock cycle of any machine cycle. The outputs of the counter are fed to a decoder. This decoder has one output for each clock cycle.

Bus Structure

The Z8000 CPU board is fully Multibus compatible, which allows it to interface with hundreds of boards manufactured for this bus. All bus lines conform with the IEEE Multibus proposed standard. In addition to this definition for P1 (the main connector), Central Data has defined the following pins of P2 (the auxiliary connector) to allow for added flexibility:

<u>P2 Pin</u>	<u>Name</u>	<u>Function</u>
5	STOP	CPU Stop Input
6	NMI	Non-maskable Interrupt Input
7	AS	CPU Address Strobe Output
57	A20	}
58	A21	} CPU Extended Address Lines
55 or 59	A22	}
56 or 60	A23	}

The Multibus interface is shown on sheets 4 and 9 of the schematics. Sheet 4 has the buffers for the address and data busses, as well as bus clock generation circuitry and command acknowledge gating.

The address buffers are driven whenever the GATE ADR/ signal goes low. For the upper 13 address lines, the buffers are not enabled during I/O operations to keep those lines at a logic low state. The lower 11 address line buffers are driven directly from the latched address lines of the CPU. The other lines are latched during T1 from the outputs of the page maps (see the memory management section).

The data bus buffers are used to gate data into and out of the board. Data is transferred out of the board whenever the GATE DBO/ signal is active, during write cycles when this board has control of the bus. Data is gated from the Multibus to the processor whenever a Multibus read cycle is in progress. During most transfers, both GATE DBIN LO/ and GATE DBIN HI/ are active, gating the full 16-bits of data onto the board. During I/O byte reads, however, the low-order data must be gated to both the high and low sections of the internal data bus. IC46 is used to gate the data to the high byte, while IC50 (on sheet 9) is used to gate the data to the low data lines. During this type of cycle, the GATE DBIN HI/ signal will go high preventing the upper Multibus data lines from entering the board.

It should be noted that the bytes are swapped between the Multibus and the internal data bus since the Multibus considers the lower data lines (D0-D7) to contain the high-order data, while the Z8000 uses the upper data lines for the same function.

The Z8000, during byte write operations, copies the data being written onto both the upper and lower data lines. During byte reads, however, it expects the data to be on the upper lines for even addresses, and the lower lines for odd addresses. Therefore, for memory read operations, a full word is always read (no byte memory reads are ever performed to the bus) to guarantee that the data appears on the proper data lines. I/O transfers cannot be done in the same manner, since 8-bit I/O devices always transfer data over the lower 8 Multibus data lines. Since the data may be required on either the high or low data lines of the Z8000 (depending on the even/odd status of the address), the data must be gated to both sets of lines from the lower Multibus lines.

Sheet 4 also includes an oscillator which can be used to generate the bus clocks. The frequency is 9.5MHz, and the buffered output can be run to the CCLK/ and/or the BCLK/ lines of the Multibus as selected by shorting plugs.

Finally, the circuit which waits for acknowledgment from any memory or I/O board being accessed is found on this sheet. The user can select if the advanced acknowledge signal is monitored, with any valid acknowledgment indicated by BACK going high. Whenever a Multibus cycle is active (CYCREQ/) and a bus timeout has not happened, pin 6 of IC2 will be low. When this is low with BACK, wait states are inserted indicating that the board being accessed has not completed the transfer.

Sheet 9 of the schematic contains the control logic for the Multibus interface. The 74S74 (IC10) is used to start a memory cycle to the bus. Its preset input (pin 10) is brought low to start the cycle, with the DS/ signal clearing the output after the processor has accepted the data. A memory cycle is started when all three inputs to the 74LS10 gate are high. Pin 5 is high except during segment traps, preventing a memory write cycle if write protected memory is being accessed. Pin 3 goes high whenever non-PROM memory is being accessed. Finally, pin 4 goes high if either input to the 74LS132 gate driving it goes low. Pin 12 will go low at the start of T2 on read cycles. Pin 13 will always go low at the start of T3. The net effect is that write cycles start one clock period later than read cycles, to guarantee the address and data setup time requirements of the Multibus.

I/O cycles are started when pin 12 of IC25 goes low. This occurs when address line BA15 is low, and the processor is doing an I/O cycle. The DS signal is used to gate the signal only after all setup times have been met.

For any type of Multibus cycle, pin 11 of IC6 will go low. All four gates of IC29 are used to specify the command to be sent to the bus. Two gates are selected with the MULTIBUS MEM CYC signal, while the other two use MULTIBUS I/O CYC. The R/W line is used to determine if a read or a write cycle is performed. The outputs of IC29 are routed to tri-state inverting buffers which enable the commands to the bus when the board is the current bus master.

The bottom section of the sheet contains circuitry to enable data bus buffers and to insert wait states to the processor on memory cycles. Whenever the processor is doing a read and CYC REQ/ is low pin 11 of IC13 will go low. This enables data from the low Multibus data lines to the upper data lines of the board. Whenever the board is doing a

write, and it has control of the bus (with GATE ADR high), pin 11 of IC24 will be low--enabling all 16-bits of data onto the Multibus. The gating of data from the Multibus to the lower internal data lines is done with one of two buffers. If a byte I/O read cycle is in progress, the data is routed from the low-order Multibus data line (thus duplicating the data gated to the board's upper internal data lines). If a read cycle is active, but it is not to a byte I/O address, then the low-order internal data lines are gated from the high-order Multibus data lines. Pin 6 of IC20 is used to gate data during the special case. Whenever this line is high, it will allow the GATE DBIN HI signal to gate data in the normal mode.

Due to the Multibus specification, all addresses must be setup 50ns prior to the start of any command. Also, during write commands, the data must be setup 50ns before the command is started. Since there are various delays between the Z8000 pins and the address and data lines of the Multibus, and because the Z8000 has a multiplexed address/data bus, wait states must be inserted for all cycles. The MEM WAIT/ signal is routed to the WAIT/ line of the Z8000, being driven during T2 of any I/O cycle (pin 4 of IC12), T1 of memory read cycles, or T1 and T2 of memory write cycles.

The 74LS123 in the center of the page is used for the bus timeout feature. It is triggered by the CYC REQ/ signal, and if another CYC REQ/ is not received within 1ms, the output (pin 12) will go high. This signal can be used to prevent the processor from going into an infinite wait-state when it tries to access non existant memory or I/O.

The top section of sheet 9 is the bus arbitration logic, which determines if this processor board should control the bus at any one time. The basic inputs to the circuit are the BPRN and BUSY signals from the Multibus, with its outputs being the BUSRQ, BPRO, and CBRQ. The BRPN input goes low whenever a higher priority board wants control of the bus. As soon as the current cycle is completed, the bus is released to the other board. The BUSRQ output signal is active whenever this board would like control of the bus, and is used in parallel bus priority resolution circuits. The BPRO output is high when neither this board nor any higher priority boards are requesting the bus. Finally, the BUSY signal is active whenever any board is in control of the bus, normally going inactive only for one clock cycle during bus exchanges or during a period of no bus activity.

All bus operations are done synchronous with a bus clock (BCLK). This clock is provided by one of the boards on the bus, and has no relationship with any non-arbitration logic

on the bus. If the user chooses to have the Z8000 CPU board drive the BCLK line, the frequency will be 9.5MHz.

The Z8000 CPU board releases the bus to lower priority devices only when the processor goes into a long series of internal operation cycles, which normally occur during very long instructions and when the processor is halted. During all other conditions, the board will activate its BUSRQ signal, as well as cause BPRO/ to go low, causing lower priority boards to be locked out.

The 74LS163 (IC52) counter is used to signal when five consecutive internal operation cycles have been performed by the processor. When this occurs, pin 15 goes high, inhibiting further counting and causing the BUSRQ signal to go inactive. This will allow lower priority boards to become the bus master if no higher priority boards request it.

The 74109 circuit (IC11) is used to properly gate addresses and data signals to the bus, since the addresses and write data must have setup and hold times of 50ns with respect to the command lines. The circuit provides that addresses are enabled one clock cycle before commands when the board gains control of the bus (when pin 8 of IC17 goes high). Also, when the bus must be released, either because a master of a higher priority has requested it or because five internal operation cycles have been performed, the command is removed one clock cycle before the addresses. The bus is released as soon as the current cycle is completed, with pin 8 of IC28 going low. This causes the latch following it to force the K/ input to the second flip-flop to go low. This is the trigger for the bus release. Note that when the MO/ signal from the Z8000 is low (this output is settable with special instructions) the bus cannot be released, allowing the CPU to keep access during sets of instructions which cannot be broken up.

EPROM/ROM Capabilities

Sheet 7 of the schematics contains the bootstrap PROMs and their associated circuitry. Two 2Kx8 PROMs are allowed on the board, one used for the high byte of data, and the other for the low byte. Whenever pins 18 and 20 of the 2716 PROMs are low, the data addressed is gated to the output. Pin 18 is low during any read cycle, with pin 20 being tied to the BOOT ENBL/ signal.

This BOOT ENABLE/ signal is low whenever a memory cycle is in progress (with SMEM high), and the BA15 signal is low, and the bootstrap enable flip-flop (IC14) is set. This latch's Q/ output is set when the board is initialized, and

can be cleared by writing to port HF801.

Note that since only BA15 is used for decoding the PROM, the data stored repeats itself eight times in the first half of each segment. Since the program is active only for a brief time (since it normally loads itself into RAM for execution), this causes no problems.

Also on sheet 7 is the address decoder for the process number port. Pin 13 of IC5 goes high whenever address HF803 is written, which latches the lower four data lines into the process number port (see the memory management section).

Programmable Timers

An 8253 timer is provided on the board which allows interrupts to be generated at three different times. Sheet 8 of the schematics contains the timer circuit, which is actually three independent programmable interval timers.

The timers can be programmed to generate an interrupt from one to 65,535 input clock cycles from when it is started. Two of the timers are driven with a 1MHz clock, with the other timer (which is used to single-step the processor) driven with the MREQ signal from the processor. This special case allows the interrupt to happen at a specific number of memory cycles after the counter is set.

The counter takes four I/O addresses: HE801, HE803, HE805, and HE807. The first three ports are written with the starting count value, least significant byte first, which is decremented each clock cycle--causing an interrupt when it becomes zero. The third port is the control port, which is written prior to writing the starting count values to the other ports. The values written to this port are H30, H70, or HB0 to prepare to write to counter 0, 1, and 2, respectively. If the control word is written without writing a count value, the interrupt output is dropped.

Interrupt Capability

The board accepts 9 interrupt input lines. The highest priority line is NMI (non-maskable interrupt) which enters the board on pin 6 of P2. This interrupt cannot be disabled by software control, and it is used mainly for catastrophic failures such as power loss and parity errors. The processor switches to system mode and, using the normal interrupt handling procedures, gets a new status word and program counter from the NPSAP as described in the Z8000 product specification.

Eight vectored interrupt lines originate on the Multibus, and they are routed to an Intel 8259A Interrupt Controller. With this controller, a variety of schemes can be implemented for interrupt handling. Normally, interrupt level 7 has the lowest priority while level 0 has the highest.

<u>Mode</u>	<u>Description</u>
Fully Nested	Interrupt Priority Fixed at 0 as the highest, 7 as the lowest.
Auto-Rotating	Equal Priority--each level, after receiving service, becomes the lowest priority level until the next interrupt.
Specific Priority	System program determines the highest priority. Lower levels determined in ascending sequence.
Polled	System Program examines interrupt status register and determines what action to take.

Sheet 8 of the schematics contains the interrupt controller, which takes the buffered vectored interrupt lines from the Multibus and generates a vectored interrupt signal to the Z8000. After the processor accepts the interrupt, it does a vectored interrupt acknowledge cycle. During this cycle, the interrupt controller returns a vector address which causes the processor to jump to the proper service routine.

Since the 8259 is designed for 8086 systems, it expects two INTA/ pulses on pin 26. The first INTA/ pulse is used for internal operations, and the data bus is not driven at that time. During the second pulse, however, the controller puts a vector out to the bus.

Since the Z8000 requires vectors to be separated by two (interrupt level 0 returns vector 0, level 1 requires vector 2, etc.) and the 8259 puts out a number corresponding directly to the interrupt level being serviced, the lower data lines must be shifted to properly coordinate the two devices. Care should be taken when programming the controller to take the shifted data lines into account.

Full information on programming the 8259 can be found in the Intel 8259A data sheet. The part requires two I/O ports, with addresses HF001 and HF003. A suggested initialization

routine to setup the part for normal (fully nested) mode is shown below. Note the shifted bits in each byte sent out due to the vector situation explained above.

```

LD          R0,#1F02
CLR         R1
OUTB       F001,RH0      ICW1
OUTB       F003,RL1      ICW2
OUTB       F003,RL0      ICW4
OUTB       F003,RL1      ENABLE ALL LEVELS
LDB        RLO,#20
LDK        R1,#8
LOOP OUTB   F001,RL0      ISSUE EOI COMMANDS
          DJNZ R1,LOOP

```

The interrupt outputs from the three timers as well as the service request line from the 9511 are combined to form the non-vectorized interrupt (NVI) input to the CPU. If the non-vectorized interrupts are enabled, the CPU fetches a new status word and program counter from the NPSAP.

9511 Arithmetic Processor Option

A 9511 arithmetic processor can be installed on the board in socket position 35 to give the board increased floating point processing power. This arithmetic processor is on sheet 8 of the schematics.

The 9511 has a variety of functions available including add, subtract, multiply and divide for 16-bit, 32-bit and floating point operands. Trigonometric, logarithmic, and other high level functions are also available for floating point operands only. Finally, functions are available for converting between the different operand types.

All operands are held in the 9511 by an internal stack, with the Z8000 pushing and popping from the stack by executing output and input instructions to it. Another output port in the 9511 is used to start a new command, while the corresponding input port is used to read the device's status.

The 9511's port addresses are HE001 for the stack and HE003 for the command and status. A two position strap selection allows the 9511 to be driven by a 2MHz or a 4Mhz clock. Normal 9511s are limited to the 2MHz clock frequency, although Advanced Micro Devices plans to introduce a 4MHz version in the future. Also, another strap selection determines if the SVREQ output will cause a non-vectorized interrupt to the Z8000. Normally, when a 9511 is installed, the interrupt is allowed. If the 9511 is not installed, the interrupt must be strapped OFF in order to prevent a

continuous interrupt.

The PAUSE/ output of the chip is used to synchronize the reading of data to the device. It is used as one of the WAIT/ inputs to the Z8000 to guarantee that a transfer is not completed until the 9511 has made the read data valid.

I/O Addressing

Since the Z8000 can address 65,536 I/O ports, no effort was made to conserve the number of ports used on the board. Also, I/O address decoding is not done to the fullest extent possible, allowing many I/O devices to be accessed using several addresses. It is suggested, however, that the suggested addresses are used in all programs to create uniformity.

Sheet 8 of the schematics contains the I/O address decoding circuitry, which uses a 74LS139 to divide the I/O address space between the different devices. The first half of the device (on the left) is enabled only during I/O cycles (SIO) to the upper 32,768 addresses (BA15) during the CPU's data strobe (DS). The next lower two address lines (BA14 and BA13) are used to select between the segment map, page map, and the other I/O devices on the board. Writes to the memory maps are gated with the T3 timing signal to shorten their duration.

The second half of the decoder allocates the addresses above E000 to the other I/O devices. This is done using the next two address lines (BA12 and BA11). The first addresses (starting at HE000) are allocated to the 9511, with the next addresses (HE800) used by the 8253 timer. The third available block of addresses (starting at HF000) is used by the 8259A interrupt controller, and the last block (HF800) is used to write the process number port and disable the PROMs. The RD/ and WR/ signals (which are routed to the I/O devices) are generated during T3 of any I/O cycle to addresses above HE000. The RD 9511/ signal is valid for the entire read cycle (not just T3) since read operations to that device are variable length.

Memory Management

The CPU board has a unique memory management circuit which allows 16 programs (or processes) to each have separate memory maps; they all can appear to have full, unobstructed address spaces consisting of 32: 64K byte segments. Naturally, all of the processes may not allocate 2 megabytes at one time because of the 16 megabyte limit of physical memory.

The segments for each process are divided into 2K byte pages which are the basic units for allocation and management. As a program requires memory, system calls will allocate additional memory to specified segments. Individual pages can optionally be write protected during the call.

This memory management scheme adds another dimension to the normal addressing of the system, the process number. Each of up to 16 processes on the system has a process number associated with it. This number is written into an on-board latch before the process is executed. When this is done, the memory map for that process is enabled. This process number latch (IC67) is on sheet 5 of the schematics, which contains the segment mapping circuitry.

For each memory access, the CPU places the segment number out on its SN pins one clock cycle prior to when addresses become valid. When the segment number becomes valid it is fed into the address lines of the segment RAMs along with the output from the process number latch. In this manner, the segment mapping RAM knows the segment and program number for this memory access. The memory management software assigns an unrelated number, called the segment address, to each such unique combination which is currently being used in the system. This number is the output of the segment map. There are 32 such numbers available which means that, on average, each program can have access to two segments. Because of the way the circuit is designed (with RAM), the number of segments assigned to each program is not dependent on any other program (except that the total of 32 cannot be exceeded). This means that one program could have 17 segments with the other 15 programs having one each, or any other combination totaling less than 32 total program-segment combinations used.

Since most programs will need only one segment, there should be no need to have over 32 process-segment combinations active at one time. If, for some reason, this is necessary, the system readily adapts by allowing the system software to remove certain maps from this RAM until they are needed. In this manner, one program can easily access all 32 segments at once--which is 2 megabytes of storage if each page of each segment is allocated. It should be noted that virtually every system will use far less than the 32 segments available without swapping maps in and out. The board has been designed, however, to allow larger numbers of segments with relatively little overhead.

Note that the outputs of the process number latch and the segment number from the Z8000 are routed through a series of multiplexers (74LS157s IC57-IC59). The alternate inputs to

these devices are driven from the buffered address lines. Under normal operation, the select input (pin 1) of the devices is low, allowing the process and segment numbers to run to the address inputs of the segment map RAMs. When the segment map is being setup, during I/O operations, the normal address lines are routed to the RAMs, which allows them to be written by using the proper I/O addresses (see the Memory Map Setup section below). Also when writing the segment maps, the data lines BD1-BD4 are gated to the I/O lines of the 2114 RAMs.

Only 32 segments (rather than 128) are available to each process because, when operating in System mode, the upper 64 segments are mapped to the system address space while the lower 64 segments are mapped to the process number currently in the latch. Further, the top 32 segments for each process are reserved for I/D operation (described later) In Normal mode, an access to the upper 64 segments causes a segment trap. In this manner, System mode programs can easily access the memory of any process on the system. The process number is forced to zero whenever the Z8000 accesses segments above H40. This allows the first map to be used for system memory, and leaves 15 maps for user processes.

An important point to note is that each process has its own set of segment numbers--each independent of the other processes. The process-segment combination is assigned a unique "segment address" number which takes the place of address relocation (during program loading) which is required in single address-space systems.

Segment address 31 is reserved for unused segment-map RAM locations. As described later, if this code appears during a memory access, a segment trap occurs telling the CPU that it tried to access an unmapped segment.

These segment address lines are sent to the page mapping circuitry (sheet 6) along with the upper 5 address lines from the Z8000 (AD11-AD15). Like the segment RAM, the page RAM's address lines are driven through multiplexers which allow the page RAM to be setup using known I/O addresses. With its normal address inputs, the RAM can determine exactly which 2K page of a segment in a process is being accessed. The outputs of these high speed (45ns access time) RAMs join the lower 11 Z8000 address lines to form the Multibus address lines.

Two additional outputs from these RAMs provide unmapped-page protection and write protection. If any page in a segment has not been allocated (assigned a physical address space), pin 7 of IC81 goes low. Likewise, if a write is attempted to a read-only page in Normal mode, pin 12 of IC20 goes low.

Note that system mode programs can write to protected areas without causing a trap. The last possible source for a segment trap is a Normal mode program trying to do an operation to a segment number above H40. As stated above, segment address 31 always causes a trap because all 32 of the pages for that segment are kept in the unallocated state. The user has the option when allocating a page of memory to write protect it.

If any type of segment trap occurs, pin 8 of IC25 will go high, and if an operation is in progress to a Multibus memory address, it causes a segment trap to be generated at the start of T2. This also latches the cause of the error into IC39, which is read by the Z8000 on the segment trap acknowledge cycle to determine the reason for the error. This acknowledge cycle causes pin 6 of IC7 to go low, clearing the error flip-flop and enabling the reason onto the data bus.

I/D Mode Operation

On boards which have revision B or later, or on previous boards ordered with the "I/D Mods", provisions have been made for operation of the processor in Instruction/Data mode. In this mode, user programs can run non-segmented, and yet have access to two segments--one for instructions and one for data. This mode is useful because the programs then become shorter, but still have access to 128K bytes of storage. If a non-segmented program is run without I/D mode, then only 64K bytes of storage are available.

The board runs in I/D mode when operating in Normal mode, and when accessing a segment number between H20 and H3F. Any time the system is running in System mode, or in Normal mode with a segment number between 0 and H'F, then the I/D operation is inhibited.

When running in I/D mode, the CPU status line ST2 is used to determine if an instruction or a data byte is being accessed. When this line is low, a data byte is being accessed. When high, an instruction byte is being read. The circuit change needed to implement the I/D mode is a multiplexer inserted in the SA0 line (segment address 0). When in I/D mode, the ST2 signal is used to generate SA0 which is then sent to the page maps. When not running in I/D mode, the output of the segment map RAM is used to drive SA0 (as described above).

The most important implication of this change is that for any process that is to run in I/D mode, two consecutive segment addresses must be allocated to it. For example, a process could use segment addresses 2 and 3, but not 3 and 4

(since that crosses a boundary where SA1-SA4 are different).

It is important to see that the operating system (up in segment 40, running in System mode) must still run like always (I/D mode is not available to it). Also, a user program can be loaded into segment number 0, and it will be runnable in segmented mode--and it can access segment numbers 0 through H1F. The operating system, with the process number port set right, can read the memory for any process without regard to whether it is instruction or data space. The example below helps to explain this:

The Z8000 monitor (ZMON) allocates the first 64K of RAM to its segment H40. The second physical 64K section goes to process 1, segment 0; the third section goes to process 2, segment 0, etc. for a total of 16: 64K sections.

The short program below will re-map the third and fourth 64K sections of physical RAM to be in segments H20 and H21 of process 1. These two 64K sections of RAM will remain in segment 0 of processes 2 and 3, giving the side effect of memory sharing between processes. Therefore, when locations of process 2, segment 0 are changed, the same locations of process 1, segment H20 will change. Likewise, when locations of process 3, segment 0 are changed, the changes will show up in process 1, segment H21. Naturally, you could change process 1, segments H20 and H21 and see the changes back at processes 2 and 3, segment 0.

```
BD 02      LDK R0,#2
3B 06 81 40 OUT 8140,R0   Map seg 20 of pr 1 to SA2
BD 03      LDK R0,#3
3B 06 81 42 OUT 8142,R0   Map seg 21 of pr 1 to SA3
9E 08      RET UN
```

After entering the program at 402000, step through it using the monitor. Then, the two segments of process 1 should be mapped as described above. Try changing locations and note the sharing between processes. Then enter 11 22 33 44 into the first four bytes of segment H20. Enter AA BB CC DD into the first locations of segment H21. Then enter the following program into segment H21, offset 10:

```
61 00 00 00 LD  R0,0      Read absolute address 0
5E 08 00 10 JP  UN,10     Loop
```

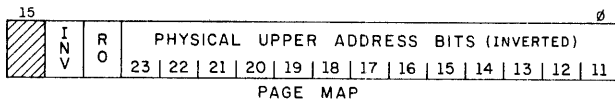
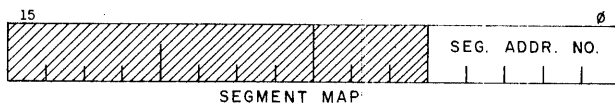
With the monitor, change the CPU mode to Normal, Non-Segmented (RE SEG 0 S/N 0). Start stepping at location 210010. Note the change in R0--it read the data from segment H20 instead of H21. Try stepping at location 200010. The garbage data will be displayed by the monitor (since it thinks you will be executing it), but when you

actually step you see that the instructions are being read from segment H21, and the data is still being read from segment H20 (clear R0 to prove it). So the monitor (or operating system, etc.) can read the instructions AND the data segments just by specifying segment H21 or H20, but when the program runs, it will only get instructions from segment H21 and only get data from segment H20.

Once again, you can still run user programs in segmented mode if they are loaded below segment H20. They will be limited, however, to H20 segments (2 megabytes).

Memory Map Setup

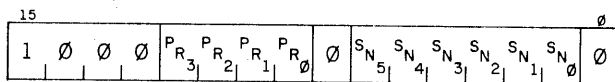
The two memory map RAMs (the segment map and the page map) are setup prior to accessing any of the associated memory locations by doing output instructions to the maps. These output instructions write the 5 bit unique segment address into the segment map and the 13 bit address and two control bits into the page map. Figure 1 shows the position of these bits in a register.



INV = 0 FOR INVALID PAGE
RO = 1 FOR READ-ONLY PAGE

Figure 1. Layout of Map Bits

The segment map output ports start at address H8000. The process number and segment number that the user wants to program should be incorporated into the address as shown in Figure 2.



PR = PROGRAM NUMBER
SN = Z8000 SEGMENT NUMBER

Figure 2. Layout of Addresses for Segment Map

4. Installation/User Selectable Options

System Board Location

The Z8000 CPU Board must be placed in the proper card position of the system in order to resolve its priority with other bus masters. In a serial bus priority resolution circuit, one designated position in the mother board has the highest priority, with positions of decreasing priority moving away from that location. In this situation, the highest priority boards (such as disk controllers, etc.) are placed in the first positions, with the CPU board(s) placed next, and any lower priority boards placed after the CPU(s). Since there are no fixed rules for determining the relative priority of various types of boards, it is left to the system designer to determine this. Most Central Data designed systems have boards inserted in the following priority:

- Cartridge/Winchester Disk Controllers
- Floppy Disk Controllers
- Intelligent I/O Boards
- CPU Boards

In systems utilizing parallel bus priority resolution circuitry, the priority of each card position is determined by a special priority resolver. Since every system can be designed differently, the system designer will have total responsibility for determining the locations of each type of board.

Note that the Z8000 CPU board normally holds the bus (will not release it for lower priority devices) unless the CPU goes into a HALT state. This means that any boards placed at a lower priority than the Z8000 board will have to wait until the CPU halts before they can gain access to the bus.

Bus Straps

Three straps are provided to determine if the board will drive the two bus clocks, and whether the board will respond to the advanced acknowledge returned by some memory and I/O boards.

The first two straps, marked BCLK and CCLK, should have shorting plugs installed if the user wishes the board to drive the corresponding clock lines of the Multibus. The Multibus specification requires one bus master per system (and only one) to drive each of these two lines, so if no other board in the system is driving either clock line, the corresponding position should be strapped.

The other bus strap (marked AACK) should be inserted if the board is to respond to the advanced acknowledge returned by some slave boards on the bus. Normally, this line is strapped to allow the fastest possible system operation.

9511 Straps

Two straps are available for the 9511 arithmetic processor. The first set, which has two positions, is marked 9511 MHZ. For normal 9511s, the position marked "2" should be jumpered with a shorting plug. If a 4MHz 9511 is being used in the board, the position marked "4" should be jumpered.

The other strapping option for the 9511 is generally used to determine if a 9511 is present on the board or not. This strap (marked 9511 INT) drives the interrupt line associated with the 9511, and also has two positions. The first (OFF) connects the 9511's interrupt line to ground, thus disabling interrupts from the device. If the board is not fitted with a 9511, this position must be jumpered, or the 9511 interrupt will always be active.

The alternate position (ON) straps the SVREQ output of the 9511 to the 9511 interrupt input signal to the processor. This is the normal position to strap if a 9511 is installed, although the user can still disable 9511 interrupts and use the 9511 in polled mode.

Bus Timeout Strap

The final user selectable option of the board determines if a reference to an invalid memory or I/O location will cause a "timeout". If the timeout feature is utilized, any reference to an invalid location, after being active more than 1ms, will force an acknowledge signal back to the processor to allow continued execution. No data transfer actually occurs, but the processor is not held up due to an invalid reference.

Due to the usefulness of this feature, it is suggested that the user jumper the AUTO TIMEOUT strap ON. If for some reason the system designer does not desire the timeout feature, the position can be strapped OFF.

BPRO Driver Enable Setup

In systems where a serial bus priority resolution circuit is used, a shorting plug should be placed over the two wire-wrap pins marked BPRO. For parallel bus priority systems, the two pins should remain disconnected. This strap is available only on boards with revision A or greater.

A22, A23 Straps

Because there was no definition of the upper four address lines when Central Data first designed its Multibus boards, they were arbitrarily placed on pins 57-60 of P2. The IEEE Multibus Specification keeps A20 and A21 on pins 57 and 58, but moves A22 and A23 to pins 55 and 56. To allow for compatibility with either old Central Data systems or the new IEEE specification, straps are provided to select which pins are used for the two upper address lines. These straps are available only on boards with revision A or greater.

5. Specifications

Word Size

Instruction--16, 32, 48, or 64 bits
Data--8 or 16 bits

Cycle Time

Fastest Instruction--750ns

Memory Capacity

On Board PROM--4K bytes (optional)
Off Board Expansion--Up to 16 megabytes in any combination of RAM, PROM, and EPROM.

I/O Addressing

<u>Port</u>	<u>Device</u>	<u>Input Function</u>	<u>Output Function</u>
0-7FF	Multibus	User Defined	User Defined
8000-9000	Seg. Map	None	Setup of Seg Map
A000-C000	Page Map	None	Setup of Page Map
E001	9511	Pop Data from Stack	Push Data on Stack
E003	9511	Read Status	Write Command
E801	8253	Read Counter 0	Load Counter 0
E803	8253	Read Counter 1	Load Counter 1
E805	8253	Read Counter 2	Load Counter 2
E807	8253	None	Write Mode Word
F001	8259A	Read IRR or ISR	Write OCW2, OCW3, or ICW1
F003	8259A	Read IMR	Write OCW1, ICW2, ICW3 or ICW4
F800	PROM Dis.	None	Disables PROM
F802	Proc. #	None	Writes Process No.

Interrupts

Non-maskable interrupt input on pin 6 of P2. Low level triggered, 100ns minimum duration. NMI interrupts are handled as specified in the Z8001 Product specification.

Vectored Interrupts from the Multibus are conditioned by an Intel 8259A. After priority has been established, an interrupt is sent to the Z8000 with a vector to the service routine.

The interrupts from the 8253 and 9511 are gated to the non-vectored interrupt input of the Z8000.

Timers

The two timers which are available for any purpose can have a timeout period of 1us to 65.535ms. The timer used is an Intel 8253.

Arithmetic Processor (optional)

Device Used: AMD 9511
Clock Speed: 2MHz (4MHz strap selectable)
Fastest floating point command execution time: 54 cycles

Interface

All P1 signals meet the IEEE Multibus proposed specification. In addition, to allow the full 24-bit address bus, CPU single-stepping capabilities, and a non-maskable interrupt, the following pins of P2 are defined.

<u>P2 Pin</u>	<u>Function</u>
5	STOP
6	NMI
7	AS
57	A20
58	A21
55 or 59	A22
56 or 60	A23

Electrical Characteristics

Vcc= +5V +5%
Vdd= +12V +5%
Icc= 4.3A typ, 5.9A max
Idd= .05A typ, .10A max (with 9511 option only)

Environmental Characteristics

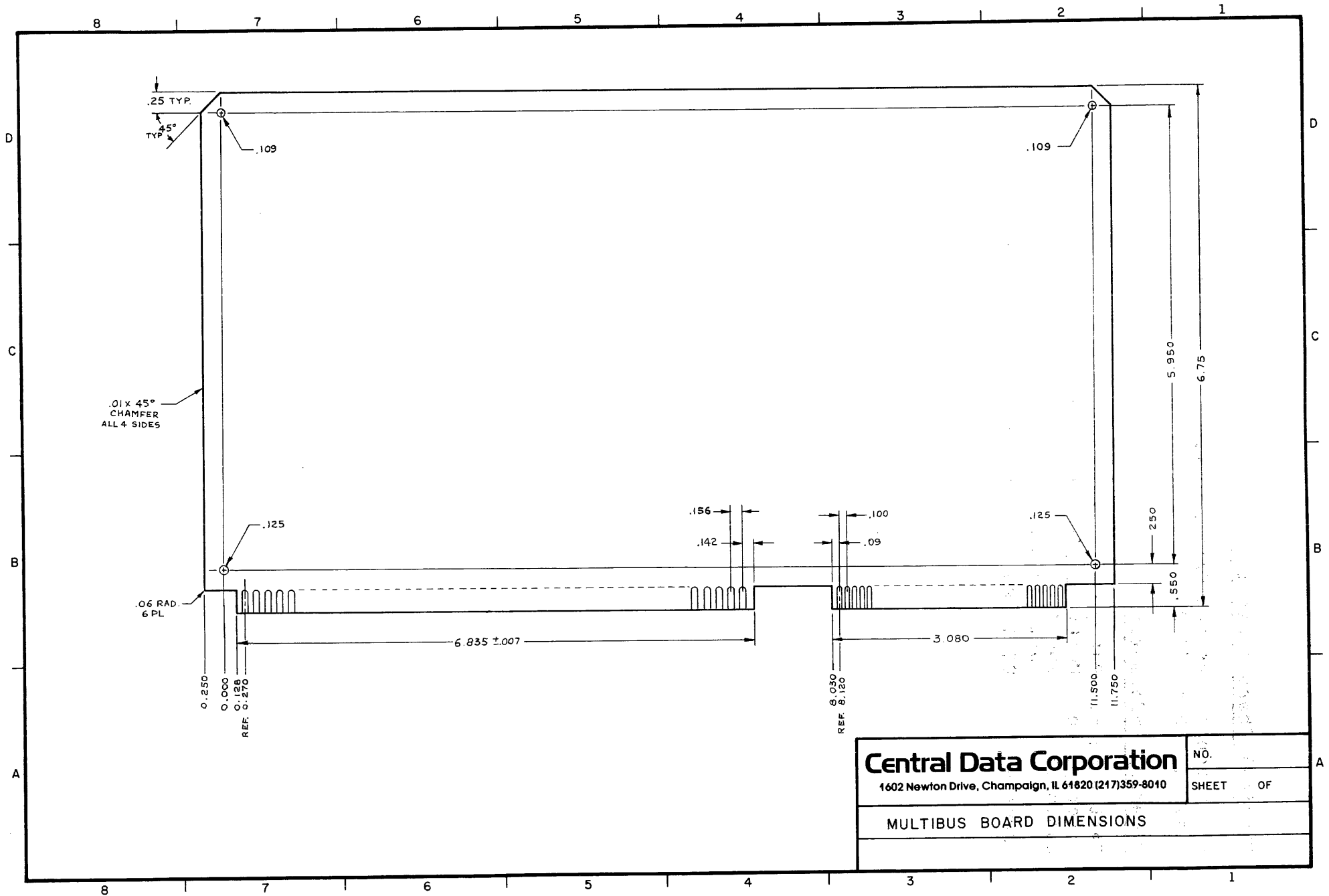
Operating Temperature--0 C to +55 C
Relative Humidity--to 90% (non-condensing)

Physical Characteristics

Dimensions: see the drawing later in the manual.
Weight: 12 oz (340gm)

Ordering Information

Part Number: B1017
Description: Multibus Z8000 CPU Board
Options: /9511 with 9511 APU
/MON with monitor PROMs
Example: B1017/MON Z8000 CPU with monitor



Central Data Corporation 1602 Newton Drive, Champaign, IL 61820 (217)359-8010	NO.
	SHEET OF
MULTIBUS BOARD DIMENSIONS	

6. Schematics

The following pages contain the schematics for the Z8000 CPU board. A full description of the circuitry is given in the Principles of Operation section of this manual.

8

7

6

5

4

3

2

1

D

C

B

A

D

C

B

A

3-INPUT 'AND' GATE



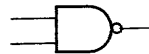
2-INPUT 'OR' GATE



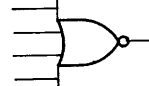
INVERTER



2-INPUT 'NAND' GATE



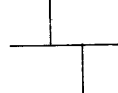
4-INPUT 'NOR' GATE



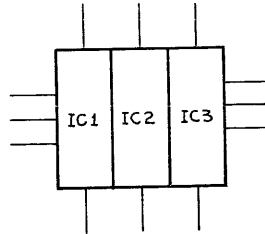
TWO LINES - NO CONNECTION



3 LINES - ALL CONNECTED



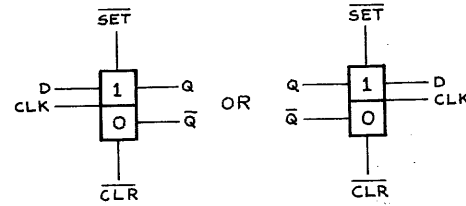
GROUP OF SIMILAR PARTS



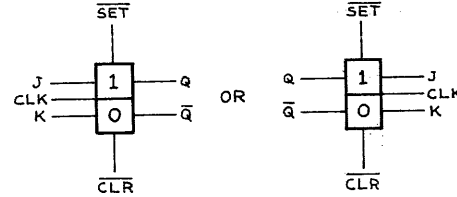
ALL LINES ENTERING ON THE SIDES ARE BUSSED TO ALL CHIPS

ALL LINES ENTERING ON THE TOP OR BOTTOM ARE SEPARATE FOR EACH CHIP

D-TYPE FLIP-FLOP



J/K FLIP-FLOP



UNMARKED ARROWS GO TO +5V

Central Data Corporation

1602 Newton Drive, Champaign, IL 61820 (217)359-8010

NO. SHEET OF

DRAWING CONVENTIONS

8

7

6

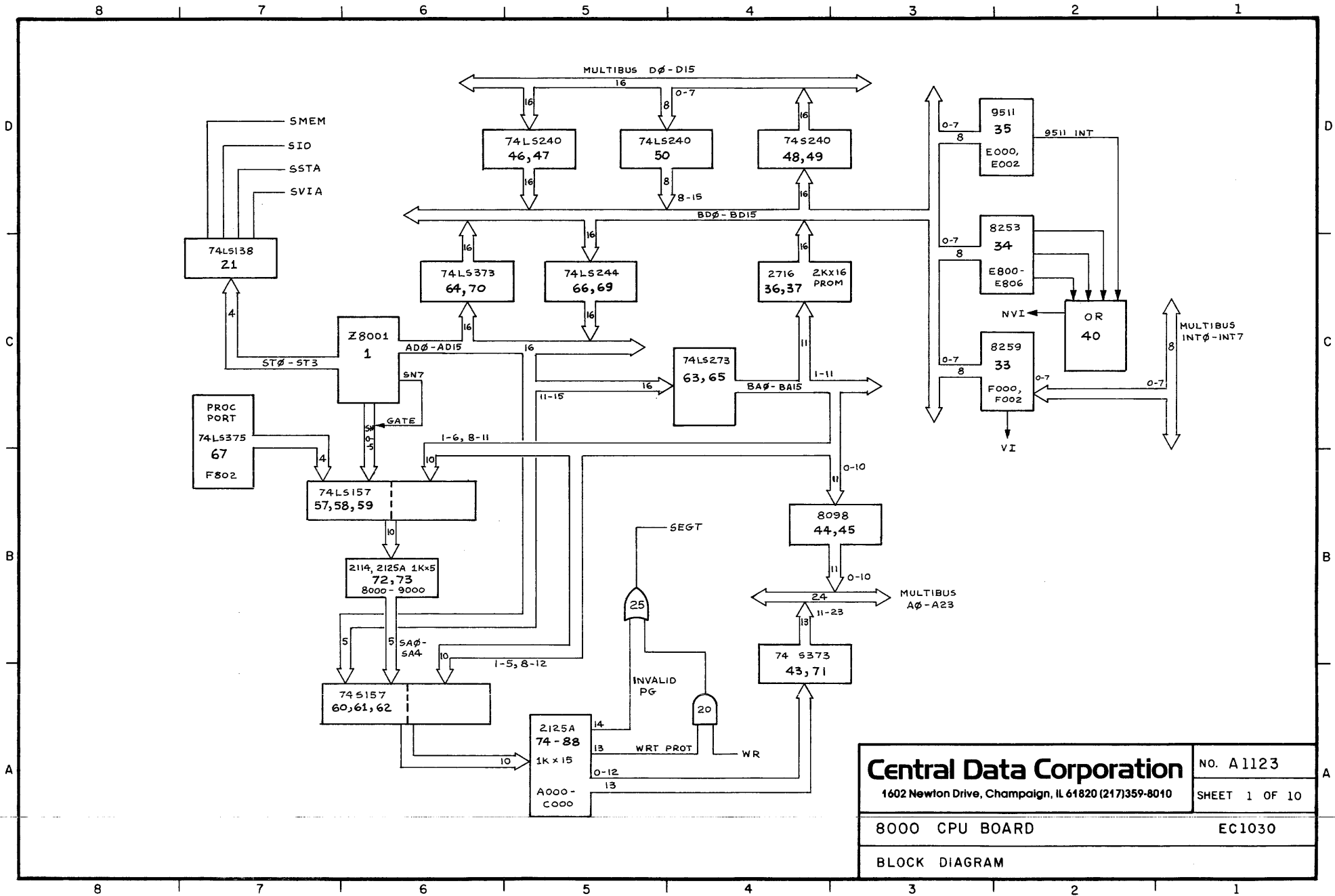
5

4

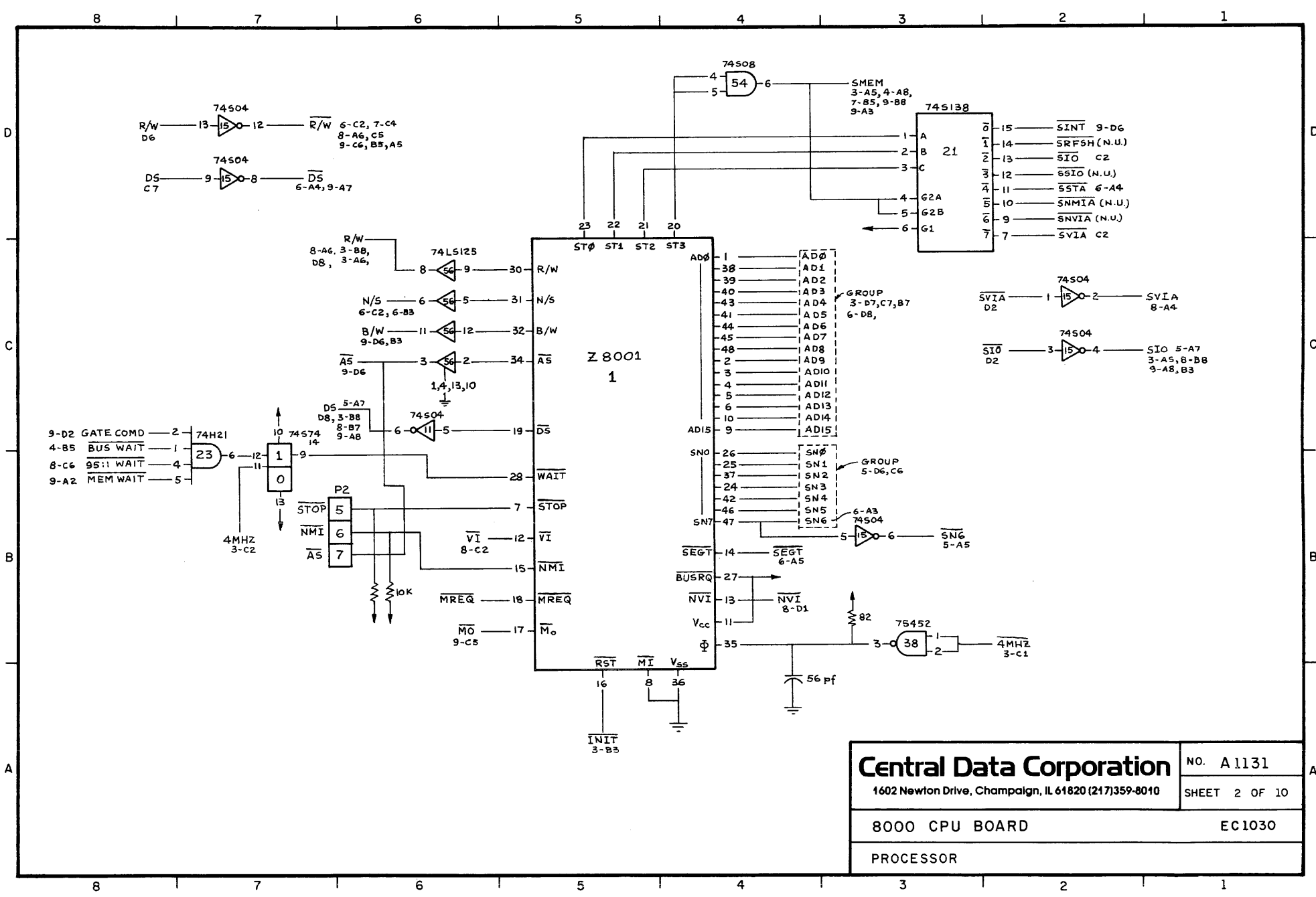
3

2

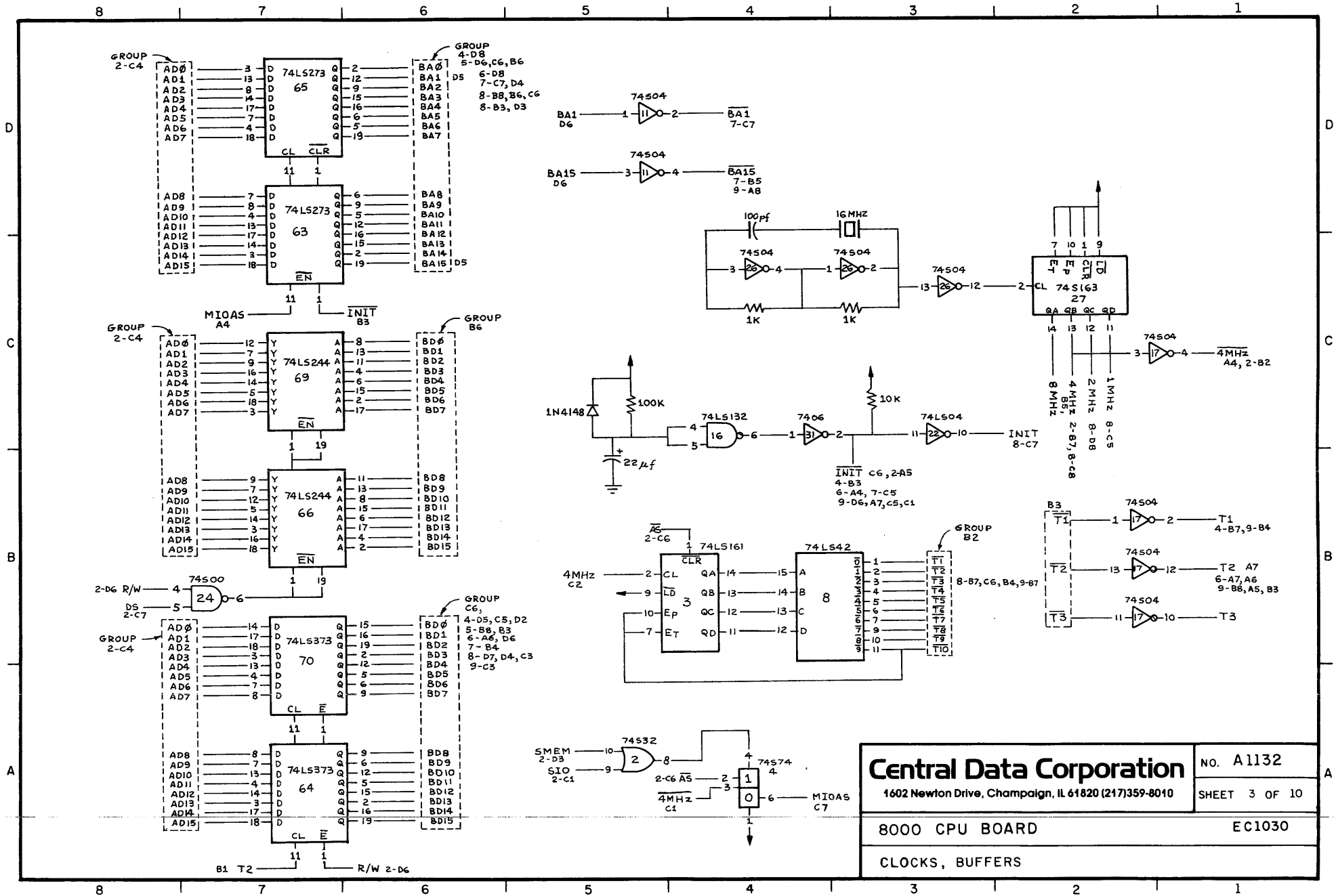
1



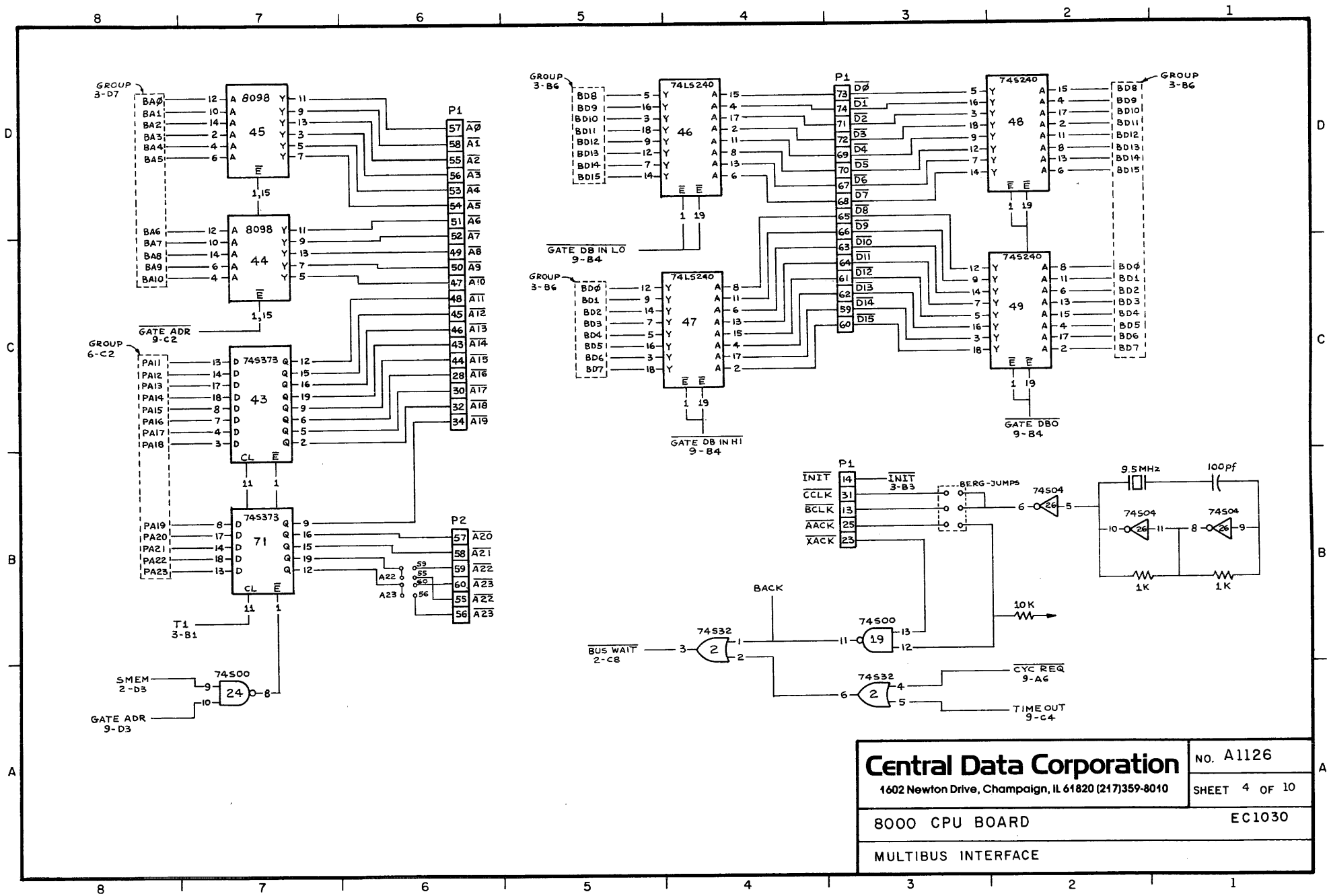
Central Data Corporation 1602 Newton Drive, Champaign, IL 61820 (217)359-8010	NO. A1123
	SHEET 1 OF 10
8000 CPU BOARD	EC1030
BLOCK DIAGRAM	



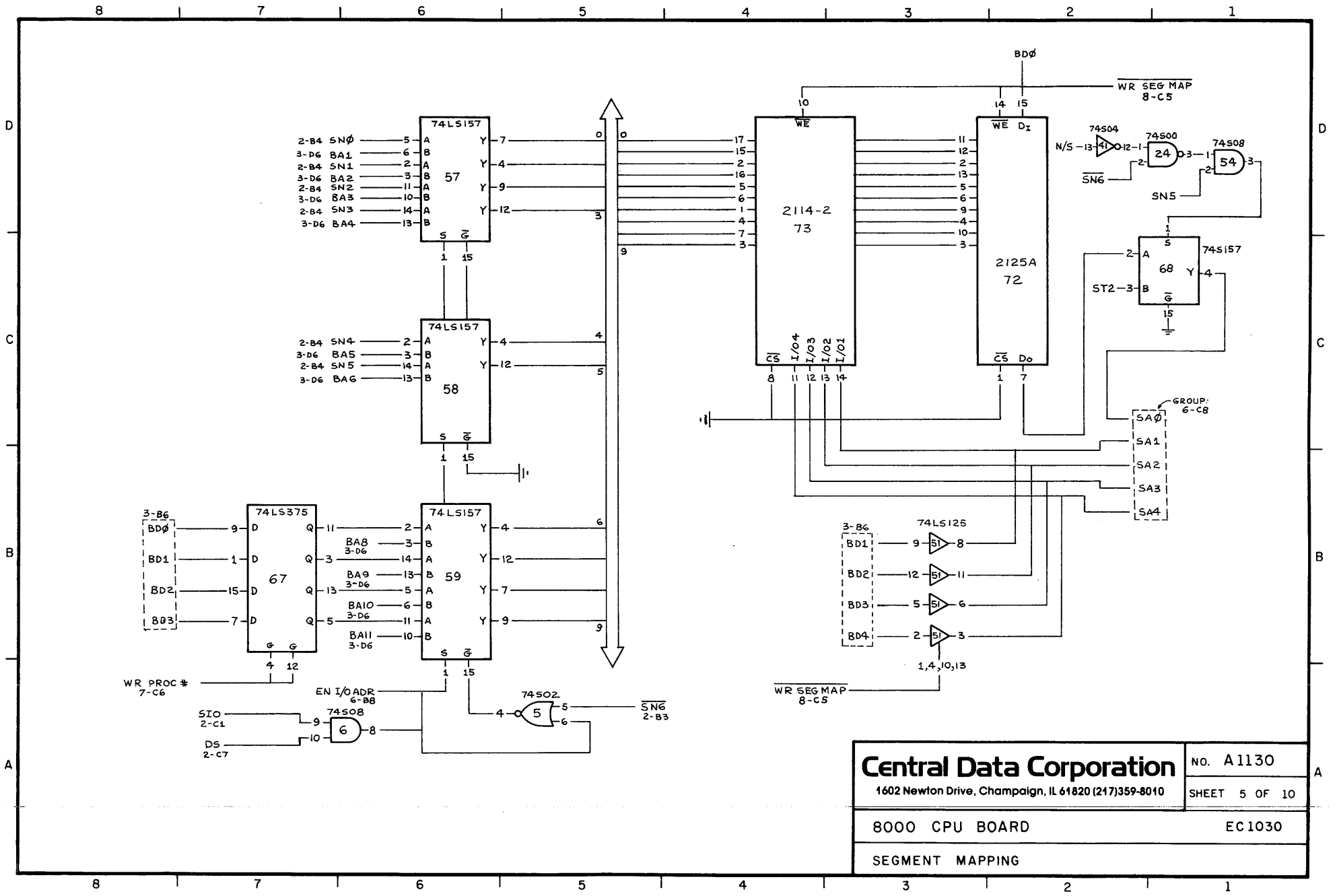
Central Data Corporation		NO. A1131
1602 Newton Drive, Champaign, IL 61820 (217)359-8010		SHEET 2 OF 10
8000 CPU BOARD		EC1030
PROCESSOR		



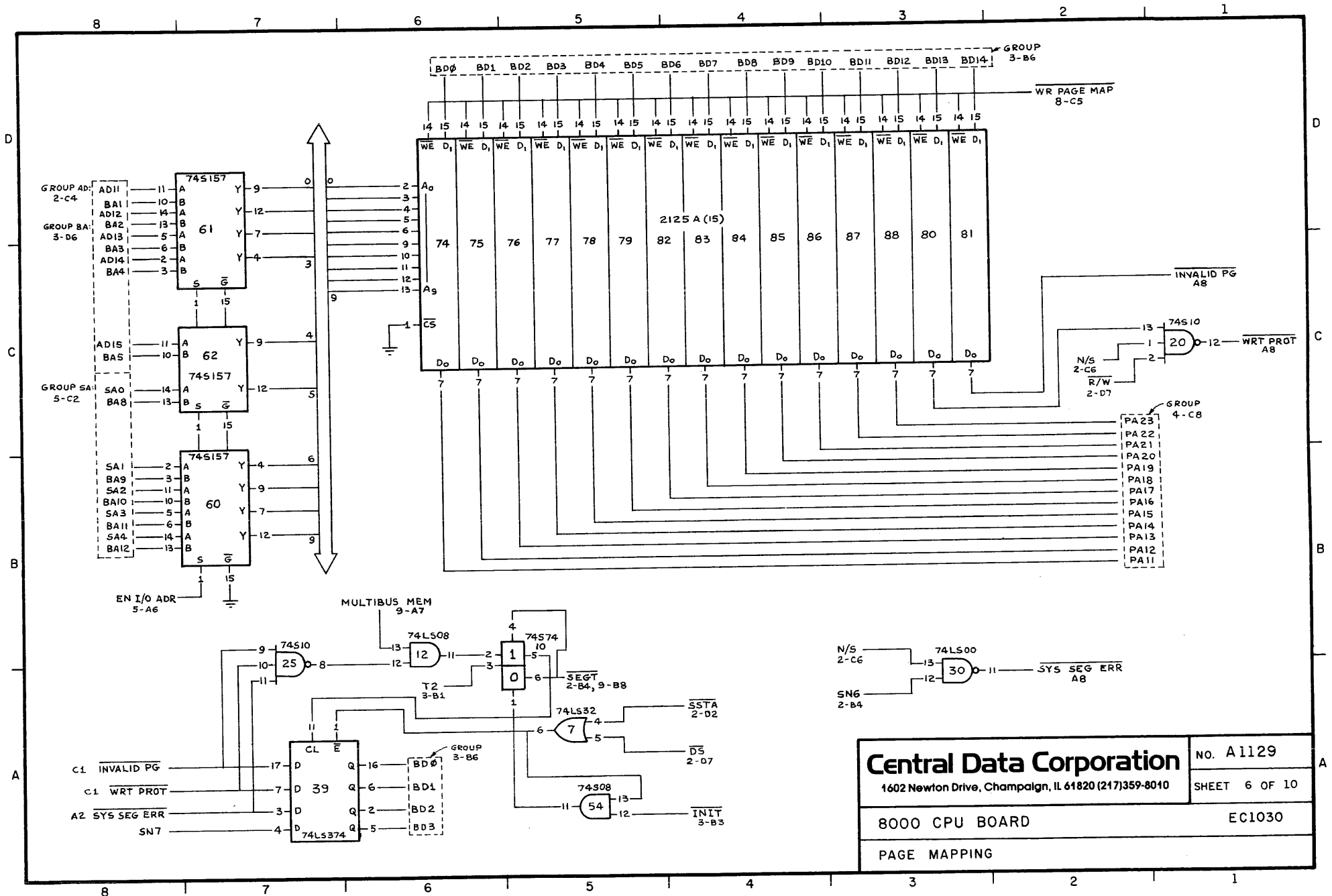
Central Data Corporation 1602 Newton Drive, Champaign, IL 61820 (217)359-8010		NO. A1132
		SHEET 3 OF 10
8000 CPU BOARD		EC1030
CLOCKS, BUFFERS		



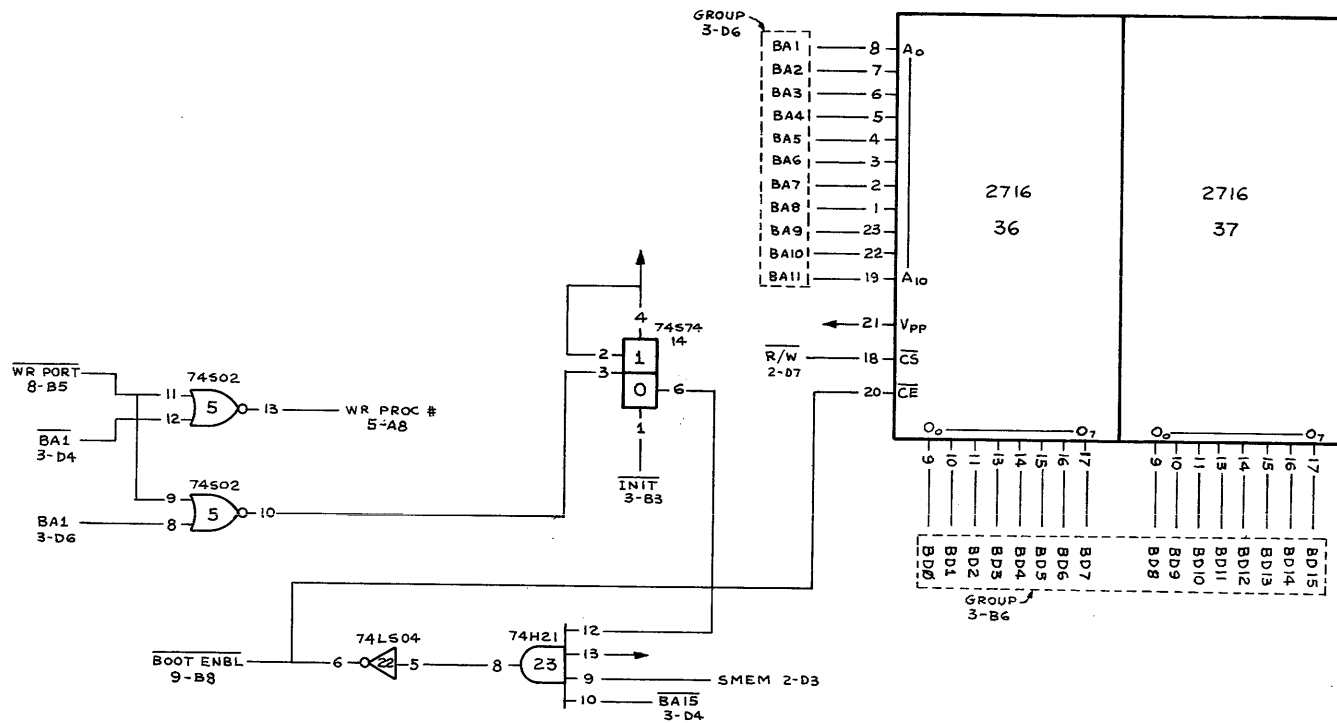
Central Data Corporation 1602 Newton Drive, Champaign, IL 61820 (217)359-8010	NO. A1126
	SHEET 4 OF 10
8000 CPU BOARD	EC1030
MULTIBUS INTERFACE	



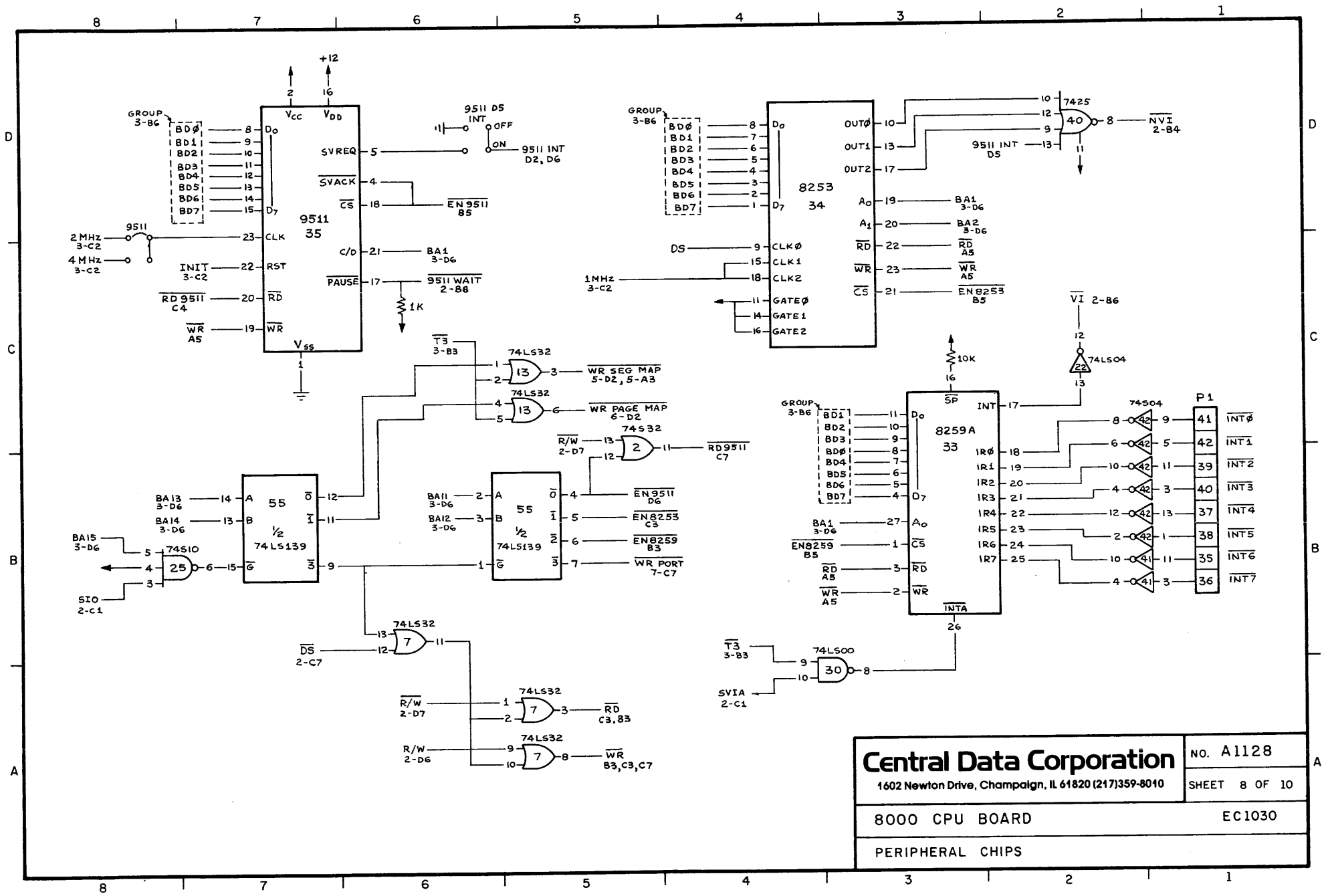
Central Data Corporation 1602 Newton Drive, Champaign, IL 61820 (217)359-8010	NO. A1130
	SHEET 5 OF 10
	8000 CPU BOARD
SEGMENT MAPPING	EC1030



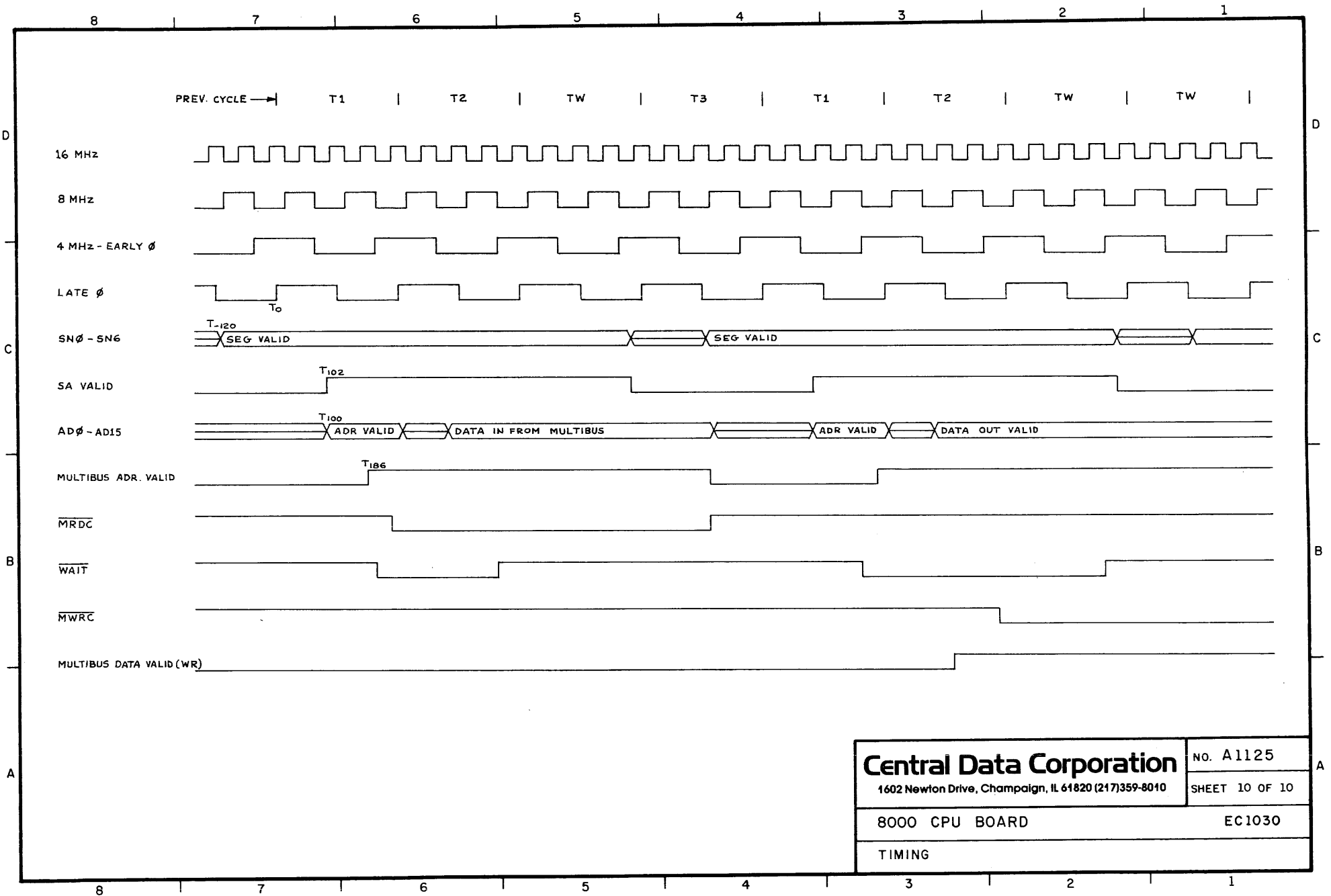
Central Data Corporation		NO. A1129
1602 Newton Drive, Champaign, IL 61820 (217)359-8010		SHEET 6 OF 10
8000 CPU BOARD		EC1030
PAGE MAPPING		



Central Data Corporation 1602 Newton Drive, Champaign, IL 61820 (217)359-8010	NO. A1133
	SHEET 7 OF 10
8000 CPU BOARD	EC1030
BOOTSTRAP PROM	



Central Data Corporation 1602 Newton Drive, Champaign, IL 61820 (217)359-8010	NO. A1128
	SHEET 8 OF 10
8000 CPU BOARD	EC1030
PERIPHERAL CHIPS	



Central Data Corporation		NO. A1125
1602 Newton Drive, Champaign, IL 61820 (217)359-8010		SHEET 10 OF 10
8000 CPU BOARD		EC1030
TIMING		

9511 MHZ

IC8



39K

74109

IC10

S74

S04

IC12



74333

S74

S04

IC17

IC18

S00

S10

S138

IC22

04

H21

S00

S10

IC33

IC34

IC35

IC36

LOW

HIGH

IC37

8259

8253

9511

2716

2716

OFF ON

IC43

7425

10K

9511

ONT

IC40

BPRO

IC42

IC41

S04

S04

5373

8908

8908

IC46

S240

S240