Bondwell

BW 2

TECHNICAL SERVICE MANUAL

PREFACE

Objective

This technical manual describes the maintenance, troubleshooting and operating principle of the Bondwell 2 portable lap size personal computer.

Federal Communication Commission Radio Frequency Interference statement:

Warning: This equipment has been certified to comply with the limits for a Class A computing device, pursuant to Subpart J of Part 15 of FCC rules. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with Class A limits may be attached to this computer. Operation with non-certified perpherals is likely to result in interference to radio and TV reception.

First edition: August 31, 1985.

This document contains the latest information available at the time of publication and is for internal use only. BHL, reserves the right to modify the contents of this material at any time. Therefore, before using this document, consult your nearest dealer or BHL for the information that is applicable and updated.

The information in this manual is intended for service technicians who are required to carry out subassembly troubleshooting; and for interested parties who need to understand the design and operation of the Bondwell 2 lap size.

When performing adjustments and troubleshooting, the following documentation should be available in addition to this manual.

1) BONDWELL 2 USER'S MANUAL

   - Target audience : end user (service technician should be familiar with the use of the BW2)

2) TEAC FD-35 MICRO FLOPPY DISK DRIVE MAINTENANCE MANUAL

   - Target audience : service technician performing adjustments and troubleshooting on the TEAC Single Side/Double Density micro-floppy disk drive.

CONTENTS

CHAPTER 1 SYSTEM OVERVIEW

1.1 INTRODUCTION

The Bondwell 2 is a standalone lap size portable computer taken all the advantage of its predecessor, the BW12/14 series, but being much smaller in size. Making use of the compact size (640 x 200 dots) LCD display and the 3 1/2 in microfloppy disk drive allowing the concept of the BW2 becoming a reality.

All the essential feature of any advance computer can be found on the BW2. Not only it consists of the LCD module and the 360 micro-floppy disk drive as mentioned above. The keyboard is located on the main unit, a built-in rechargeable battery allowing continuous operation up to eight hours without mains supply, a RS232 serial interface port, one external disk drive interface port, one parallel printer interface port and a gold-finger expansion slot located on the bottom of the computer allowing cartridge-based peripherals to be added to further enhancing the BW2.

1.2 SYSTEM BOARD

The main system PCB is located horizontally on the left hand part of the base unit. It consists of the vital functional area such as :

CPU, I/O logic, BOOT ROM, RAM, VRAM, I/O ports etc.

The BW2 utilize the Z80L as its CPU which operates on 2 MHz.

The BOOT ROM is a 4K bytes 2732. It contains the cold start loader, self test routine and character generating pattern.

The BW2 uses 8 dynamic RAM 4164 which operates on 200ns.

The system board contains all circuitary for interfacing external I/O devices. The following I/O Map gives the corresponding address for accessing a particular I/O device.

PORT MAP

00H : PPI (82c55) port A

01H : PPI (82c55) port B

02H : PPI (82c55) port C

03H : PPI (82c55) control reg.

10H : TIMER (82c53) counter reg. 0 TX RX USART

11H : TIMER (82c53) counter reg. 1 LCD - Contr.

12H : TIMER (82c53) counter reg. 2 MTRON

13H : TIMER (82c53) control reg.

20H : LCD CONTROLLER data reg.

21H : LCD CONTROLLER instruction reg.

40H : USART (Serial I/O) (82c51) data reg.

41H : USART (Serial I/O) (82c51) control/status reg.

50H : Printer output port

60H : Floppy disk controller command/status reg.

61H : Floppy disk controller track reg.

62H : Floppy disk controller sector reg.

63H : Floppy disk controller data reg.

Fig. 1.1 Ports and Memory Maps

1.3 POWER BOARD

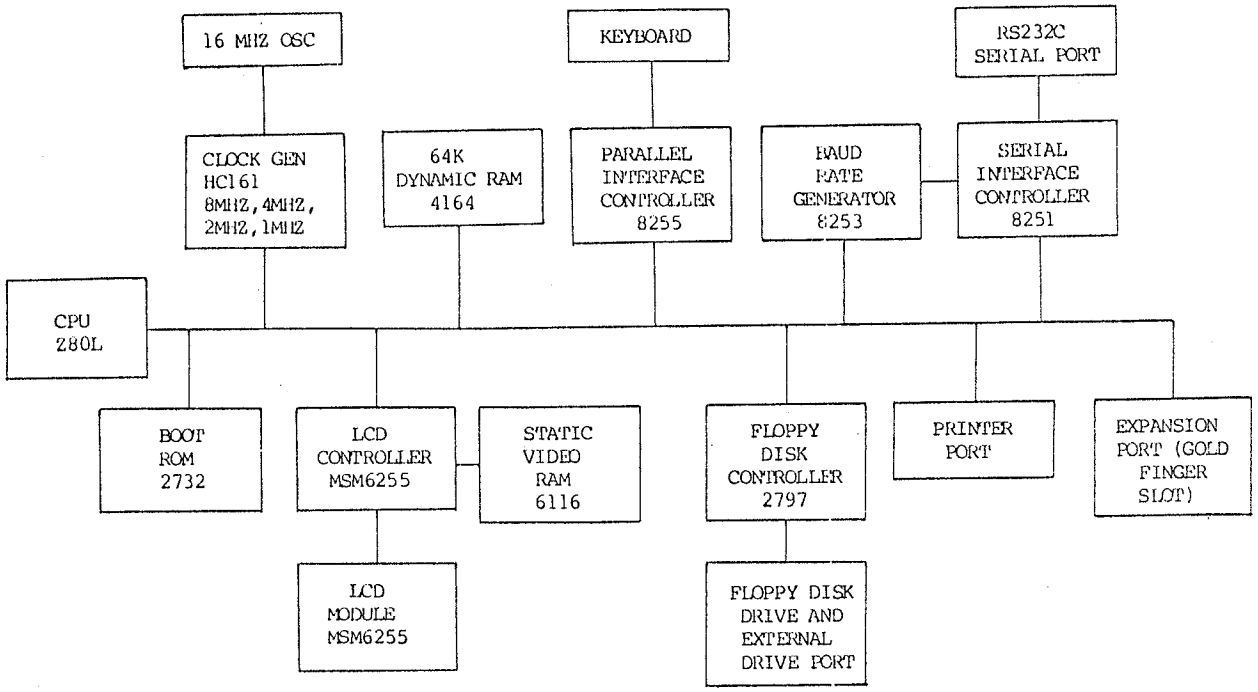
The power unit of the BW2 is situated on the rear right hand side of the base / unit. It contains a rechargeable LEAD ACID battery and a power board. The power board is placed vertically adjacent to the battery for better cooling purpose.

The power board consists of switching circuitaries to convert the +12V battery to +5V and -13.0V output for the main board and LDC module respectively while the regulated +12V is supplied to the microfloppy disk drive.

CHAPTER 2 THEORY OF OPERATION AND CIRCUIT DESCRIPTION

2.1 INTRODUCTION

In this chapter, you will find detail circuit description of the BW2. Read through the chapter before going to repair/service the BW2.



Bondwell 2 Functional Block Diagram

2.2 CIRCUIT DESCRIPTION

2.2.1 CENTRAL PROCESSING UNIT (CPU)

The BW2 uses the Zilog Z80L low power CMOS microprocessor ICI as its CPU. The Z80L is an 8-bits microprocessor having 16 address lines, various clock and control lines.

It operates on 2MHz signal from the clock divider IC26. The $\overline{RST}$ ($\overline{PIRESET}$) signal comes from the power board and upon this signal the CPU resets and clears all P.C. register and restarts at location OOOOH. All the address and data buses of the CPU are directly connected to the memory and I/O devices.

The $\overline{WAIT}$ input is pulled high disabling its function because the 64K dynamic RAM on-board main memory needs no wait state. The $\overline{BUSREQ}$ is also pulled high to disable because there is no need to set the various address, data and system control lines to high impedance state as there is no other bus master in the system (external device which the BW2 operates on). The $\overline{BUSACK}$ has no connection to any devices and is alway find to be in a high state because $\overline{BUSREQ}$ being high constantly.

The major control signals of the Z80L are the interrupts, halt, I/O request, memory request, read/write and clock lines.

When the FDC, DWG A5, requests for data transfer. The CPU $\overline{HALT}$ goes low to enable $\overline{NMI}$, the CPU is interrupted to allow specific function routine to be performed.

The system control buses, $\overline{MI}$, $\overline{IORQ}$, $\overline{RD}$ and $\overline{WR}$ is connected to the I/O logic (DWG A2) for the selection of the various I/O devices and enabling Read/Write operation to and from the system memory.

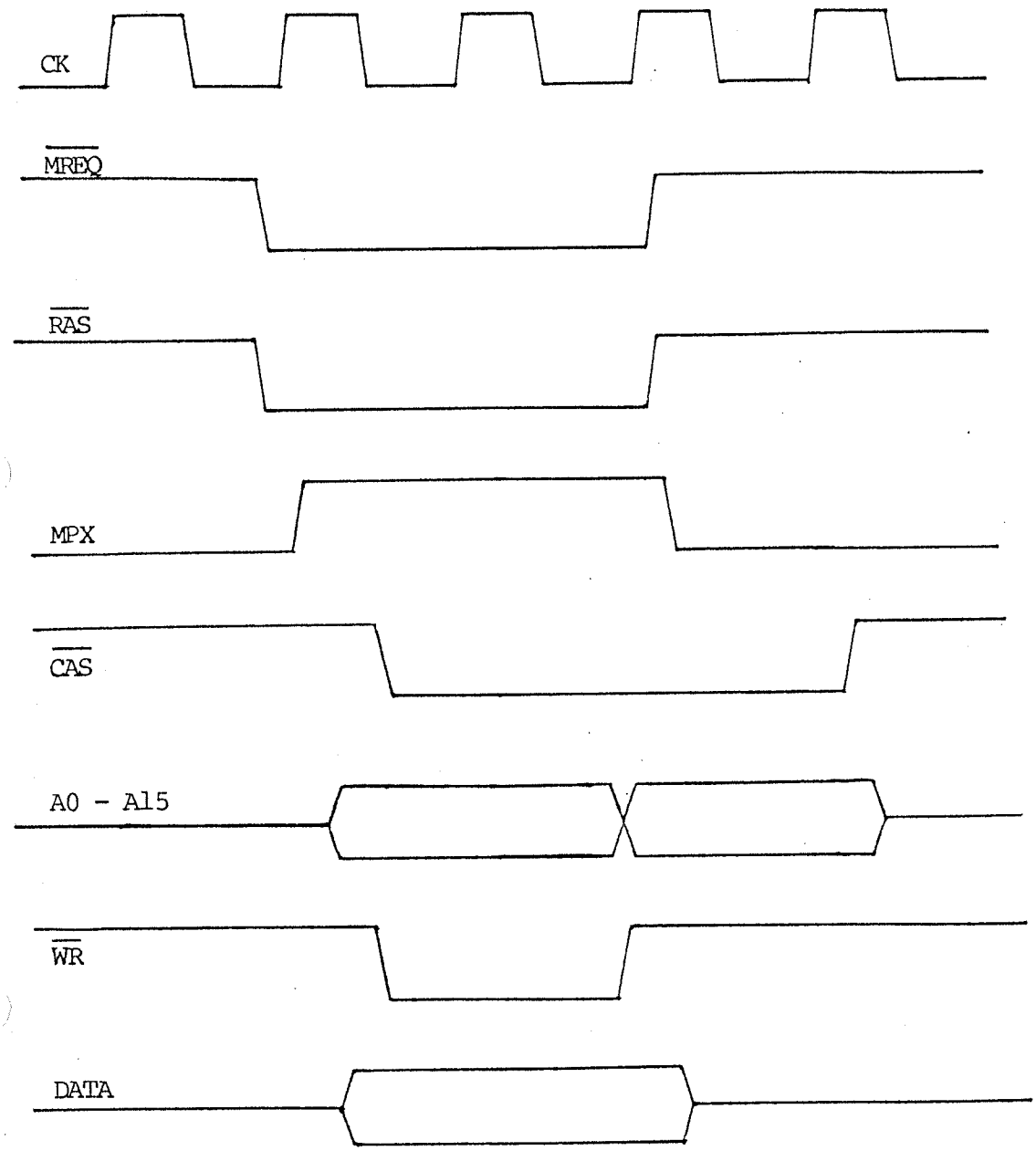


Fig. 2.1 Memory Request Timing Diagram

2.2.2 BIOS ROM

The BIOS ROM (BOOT ROM), IC8 DWG A1, is a 4K Bytes 450ns 2732 EPROM containing both the diagnostic and initialization program.

The lower 2K addresses store the cold start loader for the system while the upper 2K addresses contain the character pattern generator.

The output enable $\overline{OE}$ and chip enable $\overline{CE}$ is connected together and $\overline{ROM}$ is selected by the bank selector HC138 (IC58 DWG A5) following the power-up routine and for character generation. The advantage of connecting $\overline{OE}$ and $\overline{CE}$ is that current consumption of the 2732 ROM is dropped to standby mode when it is not operating.

2.2.3 POWER-UP INITIALIZATION

Fig 2.2 shows the power up initialization sequence.

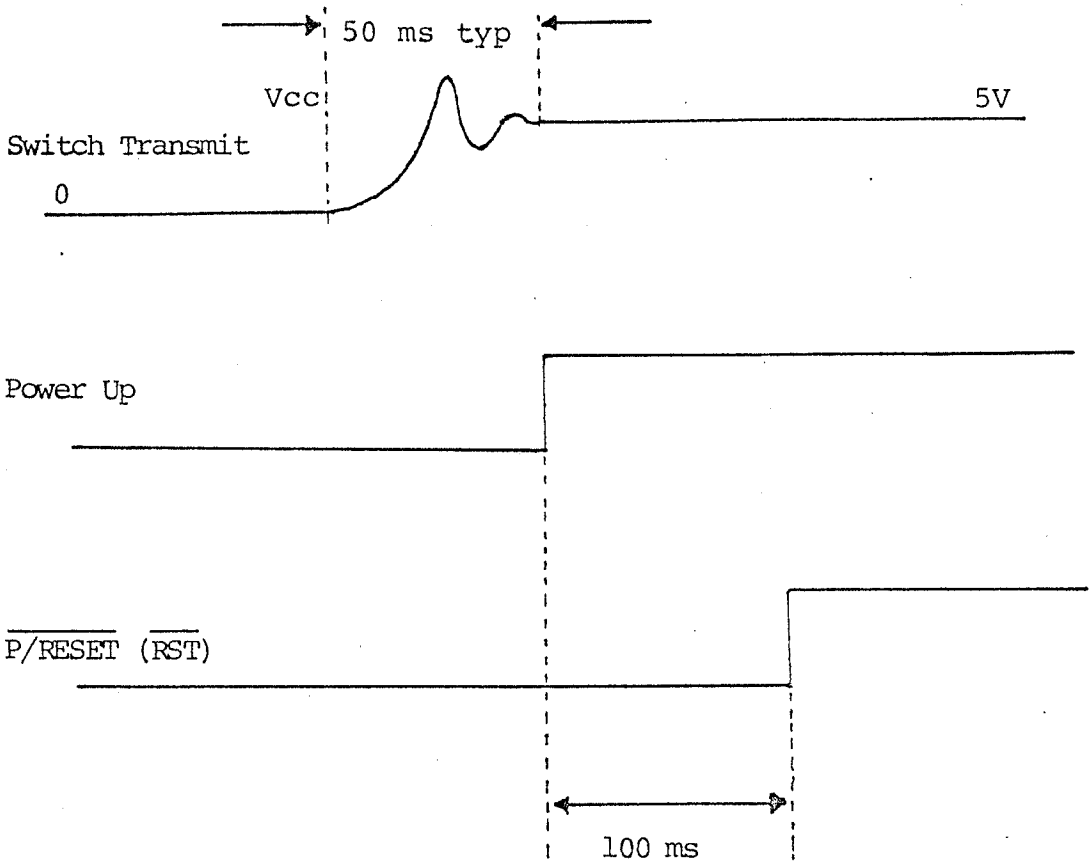


Fig. 2.2 Power Up Sequence

2.2.4 REFRESH & I/O LOGIC

The logic ICs used in the BW2 are mostly high speed CMOS. The reason of using them is mainly their low current consumption feature.

The signal provided to the dynamic RAM (main memory) is the Row Address Select $\overline{RAS}$, the Column Address Select $\overline{CAS}$ and the address Multiplex MPX.

The $\overline{RAS}$ is equal to the $\overline{MREQ}$ from the CPU. The $\overline{RAS}$ and MPX is made to synchronize with the CPU clock using two flip-flops 74HC74, IC28. MPX changes from low to high when it is triggered by the rising edge of t2 in a memory fetch cycle while $\overline{CAS}$ changes from high to low when triggered by the falling edge of t2.

In order to tailor for the timing of Z80 to match those of the BW2 family I/O chips, the falling edge of the $\overline{RD}$ and $\overline{WR}$ is delayed to generate $\overline{IORD}$ and $\overline{IOWR}$ respectively. Moreover, the chip select of each chip is generated by delaying the rising edge of the $\overline{IORQ}$. All these are made synchronous with the CPU clock by using two D-type flip-flops 74HC74 (IC29).

Also, in the BW2, the address decoder 74HC138 (IC23, DWG A2) provides individual address selection of the peripheral I/O devices. The system can gain access to the I/O devices by calling their corresponding addresses as shown in Fig 1.1, the port and memory map. The address decoder is enable by a low on $\overline{IORQ}$ and low on the address line A7.

2.2.5 DYNAMIC RAM

The 64K bytes dynamic RAM is built-up of eight 4164 (200ns). The upper 32K bytes bank (8000H to FFFFH) belongs to the common bank while the lower 32K bytes bank (0000H to 7FFFH) belongs to the switched bank. There are altogether eight switched banks, Fig. 2.4 Memory Map, controlled by the bank decoder (IC58 HC138). In these eight banks, bank 0 is the lower 32K bytes dynamic RAM, bank 1 is the VRAM bank and bank 7 is the ROM bank which is the BOOT ROM.

The two 74HC157, IC24 and IC25, are there to provide multiplexing function to address the high and low order 16 bits system address bus through the control signal MPX.

The Din and Dout of each of the 4164 RAM is connected together to facilities efficient circuit design. Therefore early write method is used in BW2 to allow the R/W of data. The $\overline{RD}$ signal from the CPU is connected to the $\overline{WE}$ pin of the RAM chip via an inverter, the direction of data flow is determined by the line.

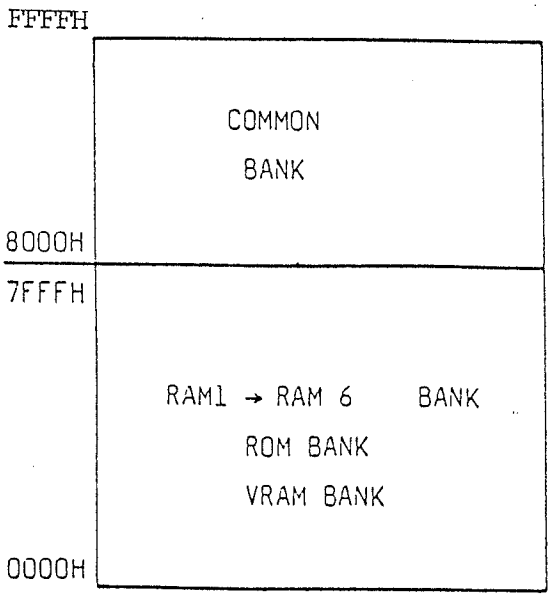


Fig 2.4 Memory Map of Bondwell 2

2.2.6 VRAM & LCD DISPLAY

The VRAM used in the BW2 are the static RAM HM6116 (IC50 to IC57), they can be directly accessed by the CPU when DIEN of the LCD controller (pin 39, MSM6255, IC49) is low. The 16K VRAM is addressed by MA0 to MA13 from 6255 where MA0 to MA10 are directly connected to the memory chips and MA11 to MA13 are connected to the address decoder (HC138, IC48) which enables the memory chips one at a time.

In the later version, two HM6264P are used instead of the eight 6116P.

The data access between the VRAM and the CPU is done through a tri-state transceiver (HC245, IC47). The transceiver is enabled by a low on $\overline{G}$ (pin 19, IC47) which is signified by a low on $\overline{VRAM}$. Data transfer is then possible after the chip is activated. Direction of the data flow is controlled by the logic level of DIR. When DIR is low, data flow from B input (CPU) to A output (VRAM), when DIR is high, data flow from A input (VRAM) to B output (CPU).

The ADF of the LCD controller (pin 40) is pulled high to allow address A0 - A13 output to MA0 - MA13 while $\overline{DIV}$ is set low for external clock application (16MHz to pin 65).

Data is input to the LCD controller through the input RD0 to RD7 directly from VRAM bus which is then fed to the LCD module via the display data parallel output (pin 33 - 36).

The LCD module is connected to the LCD controller (IC49) with a 12 way flat cable. The LCD module is a high resolution 640 x 200 dots matrix display.

The driving method of the LCD is ODD - EVEN mode which consists of 4 data lines from the 6255 (UD0 - UD3). Where UD0 and UD2 control the even dot bits of the LCD display while UD1 and UD3 control the odd dot bits. UD1 and UD2 take care of the upper half of the screen. UD3 and UD4 take care of the lower half of the screen. The Frame, DF, Load, CP are the signals for screen control and refreshing.

For detail of the LCD controller, please refer to Chapter 4 Section 4.4.

2.2.7 PARALLEL I/O

The parallel I/O in Bondwell 2 is a 82C55 which functions to select memory bank, scan keyboard, select drive and control several other I/O signals.

PC0 to PC2 are decoded by a HC138 to select one of the 8 memory banks. In these 8 banks, RAM1 is the system RAM bank which has address from 0H to 7FFFH. VRAM is the 16K bytes LCD refresh memory. RAM2 to RAM6 are connected to the gold-finger slot for expansion. The ROM bank selects the boot-ROM. PC0 to PC2 are pulled high so that when power up, ROM bank is selected.

PC3 is the only port that have no connection PC4 receives the BUSY signal from external printer and PC5 receives a motor feed back signal generated by the motor on signal. PC6 senses the data carrier detect signal from RS232 port. PC7 detects the write protect signal generated from the floppy disks.

PA0 to PA3 is decoded by a decimal decoder 7445 (IC42) to drive the row address of the keyboard. PB0 to PB7 read the scanned keyboard to see if any key being pressed. The matrix of the keyboard is in the Appendix A.

PA4 and PA5 is the drive select signal to the floppy drives. PA6 selects between RS232 port at the back of the computer and that at the gold-finger. PA7 sends a stroke signal to the printer port, which indicates the validity of data to the printer.

2.2.8 FLOPPY DISK CONTROLLER

The 2797 (IC5) floppy disk controller (FDC) is an NMOS device, its current consumption is typically 70mA. In fact the FDC is used only when the floppy disk drive is being accessed. Therefore a power saving feature is incorporated so that if the floppy is idle for more than 10 minutes, the power supply to the FDC is cut. When accessing the floppy disk is needed, the motor-on $\overline{\text{MTRON}}$ (DWG A6) signal is first issued. The power board recognises the motor-on signal as an indication of accessing floppy, it then turns on the power of the FDC. Upon power up, the self-reset circuit of the FDC issues a reset delay to itself. This will cause the FDC to reinitialize itself.

The data transfer rate of micro floppy disk drive to 250K bits/sec and the clock generator by 2797 is 1 MHz. Distinguished from other FDCs, 2797 contains an internal data separator and write precompensation circuit. The Test (pin 22) line is used to adjust both data separator and precompensation. When Test = 0, the WD (pin 31) line is internally connected to the output of the write precomp one-shot. Adjustment of the WPW (pin 33) line can then be accomplished. A second one-shot tracks the precomp setting at approximately 3:1 to insure adequate write data pulse widths to meet drive specifications.

Data separation is also adjusted with Test = 0. The TG43 (pin 29) line is internally connected to the output of the read data one-shot, which is adjusted via the RPW (pin 18) line. The DIRC (pin 16) line contains the read clock output (250KHz). The VCO trimming capacitor (pin 26) is adjusted for centre frequency. For detail description of 2797 calibration, please refer to testing procedures of BW2 in Chapter 3.

In the Bondwell 2, the CPU clock frequency is 2 MHz. When writing or reading data to or from the floppy disk drive, the worse case that the CPU has to serve the FDC is 27 usec when reading, and 23 usec when writing. These figures are not sufficient for the 2 MHz CPU to perform transferring one byte of data and increment and decrement various counters. So the method of interrupt I/O is used instead of programmed I/O.

Pin 38 DRQ is a data request signal. When the FDC wants a byte transfer, this signal is used to interrupt the CPU. Pin 39 INTRQ is interrupt request signal which goes high when a command execution in the FDC is finished. Ordinarily, CPU interrupt is disabled at the time of power up and is only enabled in the format disk program to indicate the termination of formatting one track.

2.2.9 SYSTEM CLOCK GENERATOR

The Bondwell 2 incorporates an 16 MHz crystal to produce the necessary clock frequency for the various system devices. The invertor HCU04 is used together with the crystal to produce the 16 MHz clock signal which is fed to the LCD controller and to a 4-bit binary counter which divide the signal into 8MHz, 4MHz, 2MHz and 1MHz. 4MHz is used by the Programmable Interval Timer 82C53. The 2MHz clock is used by the USART and the CPU (Ver. 1.0) while the 1MHz is the clock signal supplied to the Floppy Disk Controller.

2.2.10 PROGRAMMABLE INTERVAL TIMER (PIT)

The programmable timer of the Bondwell 2 is a 82C53 which has three 16 bit counters inside.

Counter 0 is used as the transmit $\overline{TXC}$ and receive $\overline{RXC}$ clock of 82C51 USART. OUT 0 is a square wave output dividing the 4MHz CLK 0 input by the counter value. It determines the transfer speed of the transmit/receive data.

CLK 1 is derived from the LCD controller (pin 24, IC49, DWG A4) and has clock signal value of 11 KHz feeding to counter 1. After divided by the counter value, it is a 3Hz square wave signal at OUT 1. This 3Hz output is directed to the power on LED indicator through IC38. When the battery goes low, the comparator on the power board send out a low voltage signal $\overline{LVOP}$. As a result the LED blinks to show battery-low status.

The output of counter 1 is also the input of counter 2, CLK 2. Counter 2 is a monostable output working as a motor on $\overline{MTRON}$ signal which gives a second delay for the motor to turn off after $\overline{MTRON}$ goes high, Fig 2.5 Counter 2 is also the signal for turning on the power of the $\overline{FDC}$ after it has been shut down.

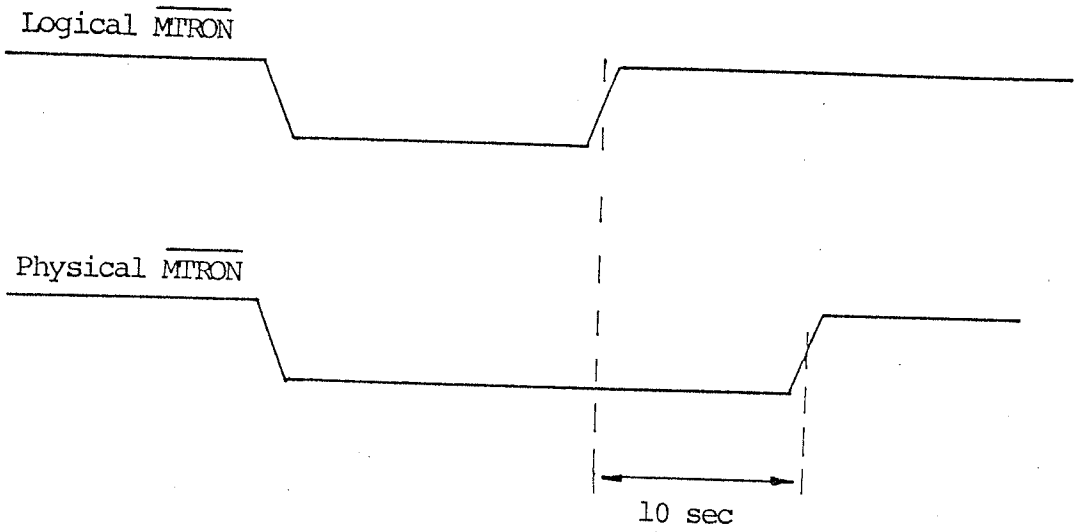


Fig. 2.5 MTRON Delay

2.2.11 PRINTER AND SERIAL I/O

The printer interface is performed by a HC273 (IC27) and two buffer ICs (IC37, 38). When data is latched to the HC273, the 8255 (IC4) sends a $\overline{STB}$ pulse to the printer indicating the validity of data at the printer port. The BUSY signal tells the 8255 the status of the printer.

The serial I/O in Bondwell 2 is performed by a 82C51 USART. The USART receives parallel data from the CPU and transmits serial data after conversion. The device also receives serial data from external devices and transmits parallel data to the CPU after conversion. The data lines of the USART are all level shifted by the line driver 1488 and line receiver 1489 (IC43 and IC44) to meet the standard RS232C specifications.

The SELECT signal from the 8255 choose between the input and the output mode of the RS232C port. When the SELECT is high, the B inputs of the IC45 is connected to the Y outputs (which is turn enable receiving function of the RS232C port). When SELECT goes low, gates A are connected to Y output gates (and RS232C port enters into transmitting mode).

2.2.12 POWER BOARD

The power in the Bondwell 2 is supplied by a built-in rechargeable battery of 12V 2.6AH. 3.4AH External power adaptor can be connected to the battery to recharge it through the D.C. jack at the back of the computer. Two switching circuit convert the 12V to 5V and -13V.

A voltage detector ICL8211 (IC1) detects the voltage level of the battery. If the battery drops below 10.5V (or any preset value), the LVOP (pin 4) goes to a low level, which together with the 3Hz pulse from teh PIT 8253 makes the power-on LED blinks to indicate battery low state. Detail of how to calibrate the low volt reference value is shown in Chapter 3.

The 5V output of the system is supplied through a L296 (IC2). The L296 is a monolithic switching regulator. The L296 regulation loop consists of a sawtooth oscillator, error amplifier, comparator and the output stage. An error signal is produced by comparing the output voltage with a precise 5.1V on-chip reference Zener Zap trimmed to ± 2%. This error signal is then compared with the sawtooth signal to generate the fixed frequency pulse width modulated pulses which drive the output stage. Precision and frequency stability of the loop is adjustable by an external RC network connected to pin 9.

The -13V output of the Bondwell 2 is supplied by a TL497AC (IC3) which is a fixed-on-time variable-frequency switching voltage regulator control circuit. The on time is programmed by an external capacitor connected between the frequency control pin (pin 3) and ground. The output voltage is controlled by an external resistor ladder network (R9, R10 and VR2). The -13V is solely for the supply to the LCD module and on-board 1488 chip.

The +12V supply is regulated through the high current linear regulator 7812 (IC5) which is, at the same time, the pre-regulator for the input of TL497AC. Though in the earlier version, it may be noticed that the +12V supply is taken directly from the battery and the pre-regulator for IC3 is 78L12.

The BW2 energy saving circuitary is constructed by using timer TLC555 (IC4) and it's associated components.

When the $\overline{\text{EN5V*}}$ goes low, the O/P (pin 3) of IC4 goes high which turns on the transistor T2 closing the relay RRD51A05. The +5V* is so connected to the +5V rail. The +5V* is the power supply to the FDC and the FDC driver/receiver.

When the $\overline{\text{EN5V*}}$ goes high turning off the transistor T1. The capacitor C18 is then changed up through R16. The capacitor is initially discharged when T1 is turned off when $\overline{\text{EN5V*}}$ is low. When the voltage of C18 is greater than two-third of 5V, the output of the TLC555 will go low which turns off the transistor T2 as well as the relay. The path from the +5V to the +5V* is opened cutting off the power supply to the FDC and FDC driver/receiver. The time delay to turn off +5V* is determined by the value of R16 and C18 and is typically 5 minutes.

CHAPTER 3 SYSTEM TROUBLESHOOTING

3.1 INTRODUCTION

This chapter provides all information necessary to troubleshoot the Bondwell 2 Lap Size Computer on the subsystem level. However, the following initial checks must be performed before the troubleshooting begins.

1) Use software which is known to be good in order to determine if the problem is hardware or software related.

2) Perform a visual check prior to digging into the circuitry. That is, check for blown fuse, broken or burned components, damaged connectors, intermittent switches, etc. and other obvious defects.

3.1.1 INITIAL CHECKS

A self-test routine resided in the BIOS ROM is automatically performed each time the power is turned ON.

After ensuring the Bondwell 2 is properly set-up, switch on power (Power Switch is located on the left hand side of machine) for the system unit. The self-test routine begins to run.

Normally the monitor should display :

"64K RAM is installed"
and the system will urge you to put in a system diskette to load system.

If all system components are in full functional order. The display will show "pass all testes loading system".

The self-test routine resides within the BIOS ROM covers the areas of System RAM, VRAM, PIO, TIMER, USART, LCD controller and FDC.

If the system encounters any fault in the testing procedure, whether it is caused by defected components or any other reasons. The system will fail the self-test routine and display an error message of the related area will be displayed on the screen.

The message shown on screen might be one (or more) of the followings :-

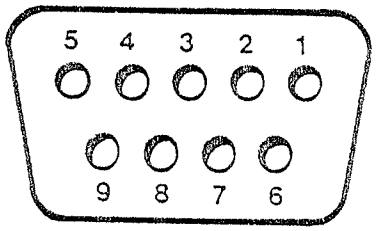
1. System RAM test fail
2. VRAM test fail
3. LCD controller test fail
4. PPI test fail
5. Counter test fail
6. USART test fail
7. FDC test fail
8. DRIVE A fail

The above messages indicates that the system fails to communicate with the particular device when they are called upon. They are aimed to provide a general direction where the trouble lies.

However, before starting any troubleshooting procedure, one must make sure whether the fault is a hard failure or not. This can be done simply by turning the power off and switch-on the power again. If the symptom persists, then start trouble-shoot the main board or the concerned area by following the troubleshooting flow-chart in the next section.

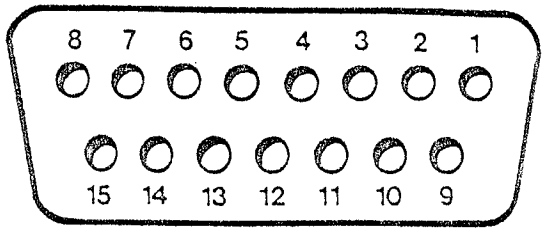
3.2 EXTERNAL DEVICE CONNECTOR WIRING DIAGRAM

3.2.1 RS-232C SOCKET PIN ASSIGNMENT (FRONT VIEW)



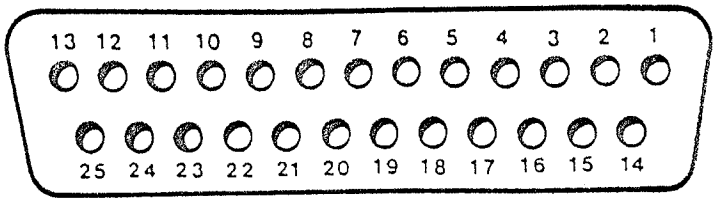
PIN NUMBER	SERIAL
1	GND
2	$\overline{\text{TRANSMIT DATA}}$
3	$\overline{\text{RECEIVE DATA}}$
4	REQUEST TO SEND
5	CLEAR TO SEND
6	DATA SET READY
7	GND
8	DATA CARRIER DEFECT
9	DATA TERMINAL READY

3.2.2 CENTRONICS-TYPE PRINTER PORT PIN ASSIGNMENT (FRONT VIEW)



PIN NUMBER	SIGNAL
1	$\overline{\text{STROBE}}$
2	DATA 0
3	DATA 1
4	DATA 2
5	DATA 3
6	DATA 4
7	DATA 5
8	DATA 6
9	DATA 7
10	GND
11	BUSY
12	GND
13	GND
14	GND
15	GND

3.2.3 DISK DRIVE EXPANSION PORT PIN ASSIGNMENT (FRONT VIEW)



PIN NUMBER	SIGNAL
1	+12V
2	+5V
3	+5V
4	$\overline{\text{INDEX}}$
5	$\overline{\text{DRIVE SELECT 1}}$
6	$\overline{\text{DIRECTION}}$
7	$\overline{\text{STEP}}$
8	$\overline{\text{WRITE DATA}}$
9	$\overline{\text{WRITE GATE}}$
10	$\overline{\text{TRACK ØØ}}$
11	$\overline{\text{WRITE PROTECT}}$
12	$\overline{\text{READ DATA}}$
13	SIDE SELECT

14	+12V
15	+12V
16	+5V
17	$\overline{\text{DRIVE}}$ SELECT Ø
18	$\overline{\text{MOTOR ON}}$
19	$\overline{\text{READY}}$
20	GND
21	GND
22	GND
23	GND
24	GND
25	GND

3.2.4 EXPANSION SLOT PIN ASSIGNMENT

J18 EDGE CON.			J19 EDGE CON.
5V	1	26	12V
D3	2	27	12V
A1	3	28	A3
A2	4	29	A4
$\overline{\text{CTSB}}$	5	30	A5
$\overline{\text{RST}}$	6	31	A6
$\overline{\text{MODSEL}}$	7	32	A7
16MHZ	8	33	A8
$\overline{\text{IORQ}}$	9	34	A9
$\overline{\text{RD}}$	10	35	A10
D0	11	36	A11
D1	12	37	A12
D2	13	38	A13
A0	14	39	A14
D4	15	40	$\overline{\text{RAM6}}$
D5	16	41	$\overline{\text{RAM5}}$
D6	17	42	$\overline{\text{RFSH}}$
D7	18	43	$\overline{\text{WR}}$
DCDB	19	44	SELECT
$\overline{\text{DTRB}}$	20	45	$\overline{\text{RAM2}}$
$\overline{\text{RTSB}}$	21	46	$\overline{\text{RAM3}}$
$\overline{\text{DSRB}}$	22	47	$\overline{\text{RAM4}}$
TXDB	23	48	$\overline{\text{SLOT}}$
RXDB	24	49	GND
GND	25	50	5V

3.3 FAULT ISOLATING FLOW CHART

Whenever experiencing any kind of malfunction on the BW2. The following fault-isolating flow chart and the Remarks in the following page enable a quick and systematic approach to locate and correct the fault.

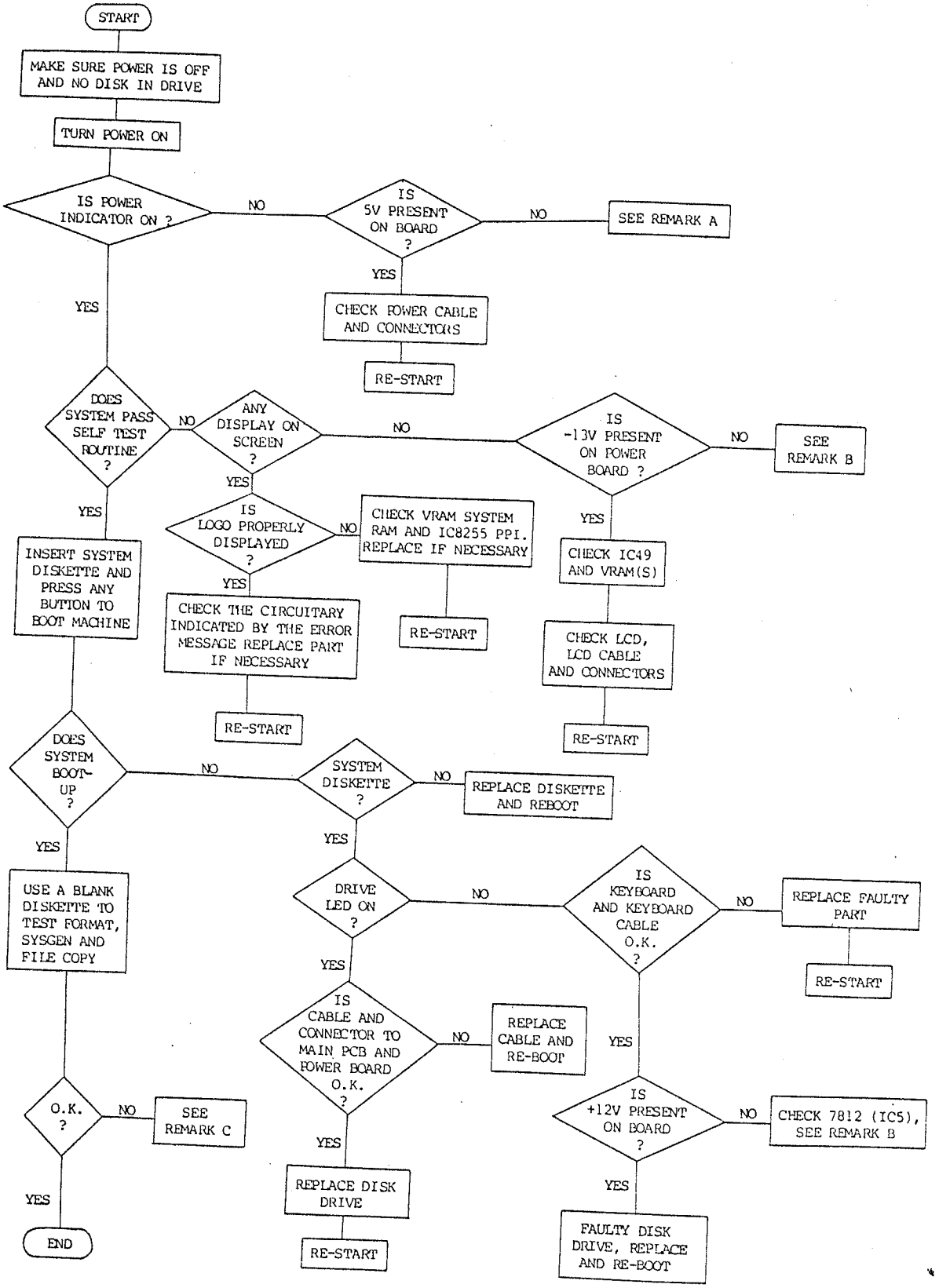


Fig. 3.1 Fault Isolating Flow Chart

REMARKS :

A. i) Check switching regulator pin 11 of IC2 L296 for oscillation. The output should be a saw tooth waveform of 60KHz.

ii) Check pin 2 of the above IC, the output should be a 60KHz square waveform.

B. i) Check the output of IC5 7812 (PCB version 1.5) for +12V.

ii) Check pin 3 of IC3 TL497AC switching regulator for a periodic output in the following form.

iii) Check also the output of pin 7 for square wave.

C. i) Check the timing frequency and pulse width of the variable capacitor J14 and the two variable resistors J15 and J16 is of the correct specification and calibrate accordingly.

Adjust J15 (10KVR) so that the pulse width on FDC pin 31 is 250ns + 10ns.

Adjust J16 (50KVR) so that the pulse width on FDC pin 29 is 500ns + 20ns.

Adjust J14 (VC) to set the frequency of pin 16 FDC to 250KHz + 1KHz.

ii) Check the +5V * to FDC and varify the signal source from the power board and the Programmable Interval Timer 82C53 IC6.

3.4 GENERAL TROUBLESHOOTING TECHNIQUE

1. When going to the suspected area of the main PCB, you should perform output to input check until the bad signal is located.

2. If the symptom indicated is not very obvious or is difficult in deciding where to start troubleshooting, then you should start with the clock generator and CPU as shown in Schematic DWG A1 and check for signal activity.

3. You could swap parts or even a sub-assembly to simplify a complicated problem.

3.5 DISASSEMBLY/ASSEMBLY PROCEDURE

1. Unscrew the four screws at the bottom of machine and the two at the inside of the carrying handle mounting plate.

2. Separate the machine top and bottom cabinets by pressing inward to unlock the hooks which are located on the bottom cabinet rim.

   HINT : it is easier to undo the hook on the left hand side (direction key side) of the machine first. Fig. 3.2 shows the location of the screws and hooks.

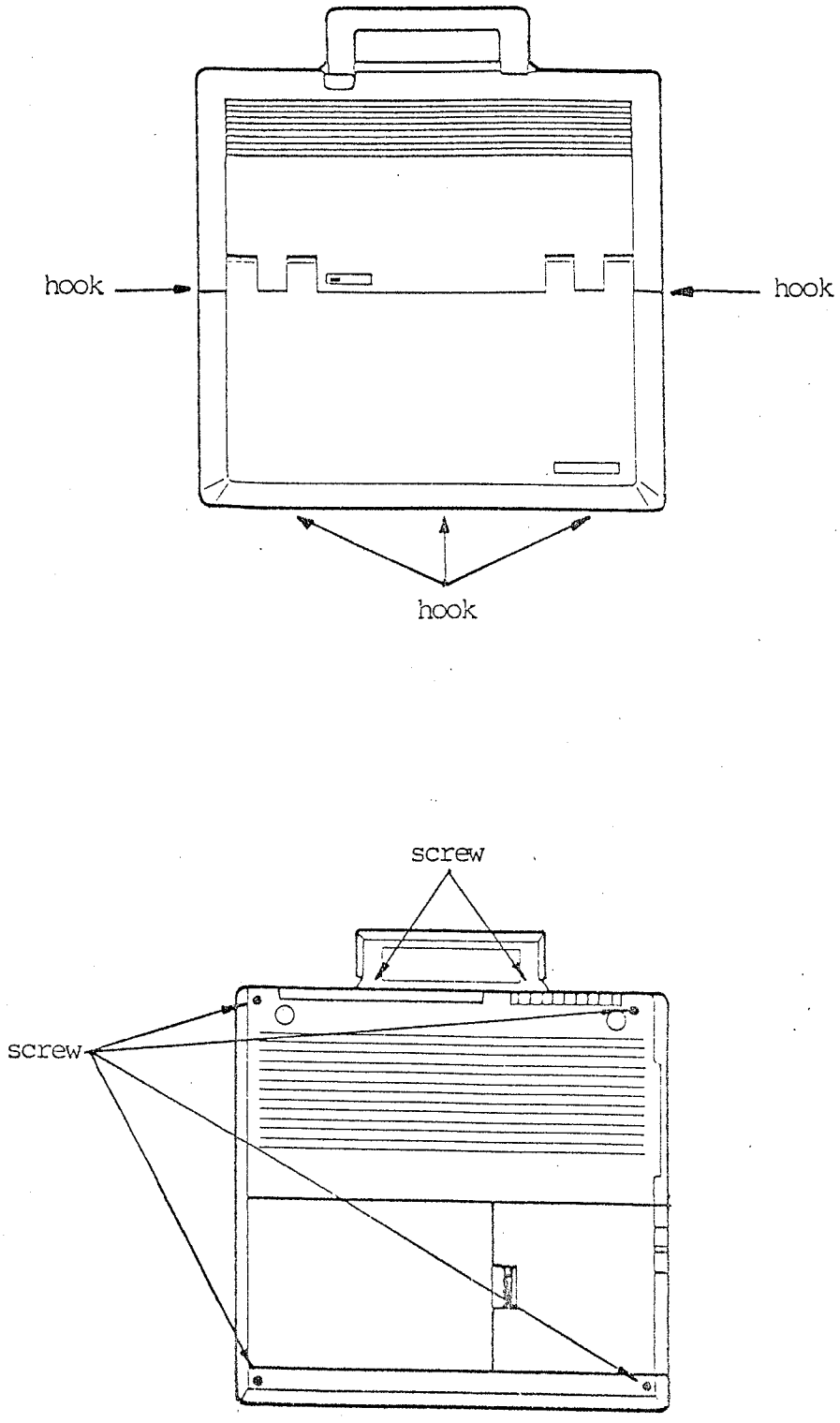


Fig. 3.2 Screws and Hooks Location of Bondwell 2

3. Rest the machine on its base and gently lift up the top cabinet.

4. Disconnect the keyboard cable from the keyboard assembly P.C.B.

5. Disconnect the LCD module connecting cable from the main P.C.B. mounted on the base of main unit.

6. Undo the screw that hold the 'Power On' LED indicator and release it from its housing.

   Top cabinet can be completely free from the base unit.

7. To separate the keyboard unit, release the four screws on the keyboard template (metal plate) that hold the unit to the top cabinet. Keyboard can then be free from top cabinet housing.

8. On the inside of the top cabinet there are two screws (with compression spring) that hold the LCD module to the machine. Undo these screws to release the module from cabinet.

9. Rest the LCD module unit on a flat surface with LCD facing upward. Undo the six screws and the four hex screws mounted on the front plate. Put away the LCD front cabinet and direct access to the LCD module is possible.

Bottom Unit :-

10. unscrew the four screws that hold the microfloppy disk drive and disconnect all the sockets to the disk drive. Free the drive unit from bottom cabinet.

11. Release the rechargeable battery by undoing the two screws that holds the battery by the metal fastening plate. At the same time, gently lift off the power supply board (the small p.c.b. situated vertically adjacent to the battery) together with the power supply socket.

12. Undo all the screws on the main board and handle mounting plate that hold the main p.c.b. to the base of bottom cabinet. Gently lift up the p.c.b. and disassembly procedure is completed.

13. Re-assembly procedure is reverse to disassembly procedure.

3.6 BONDWELL 2 HARDWARE SPECIFICATION

3.6.1 MAIN BOARD

The diagnostic on the BOOT ROM, as mentioned in the earlier section, must be passed, which include :

System RAM test,
Video RAM test,
PPI test,
USART test,
TIMER test,
FDC test,
LCD CONTROLLER test, and
Disk Drive A test.

3.6.2 POWER CONSUMPTION

The power consumption of main board with keyboard, LCD module and floppy disk drive disconnected.

+5V - less than 400 mA
+12V - less than 30 mA
-12V - less than 31 mA

3.6.3 CLOCK FREQUENCY

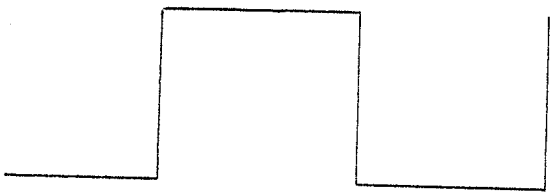
16 MHz (IC33 pin 6) - ± 0.1%
4 MHz (IC33 pin 13) - ± 0.1%
2 MHz (IC33 pin 12) - ± 0.1%
1 MHz (IC33 pin 11) - ± 0.1%

Output level : high $\geq$ 4.5V
low $\leq$ 0.2V

3.6.4 WRITE PULSE WIDTH, READ PULSE WIDTH AND DATA RATE

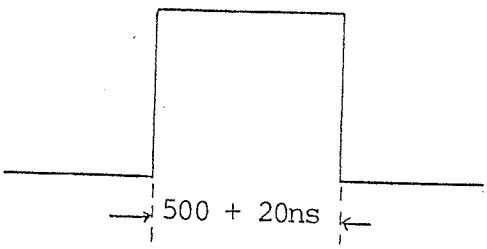
When $\overline{\text{TEST}}$ pin (FDC 2797 pin 22) strobe low :
Note : Steady state - 30 seconds after power on.

pin 16 of FDC - 250KHz ± 1KHz (at steady state)
pin 16 of FDC - 250KHz ± 10KHz (at any other time)

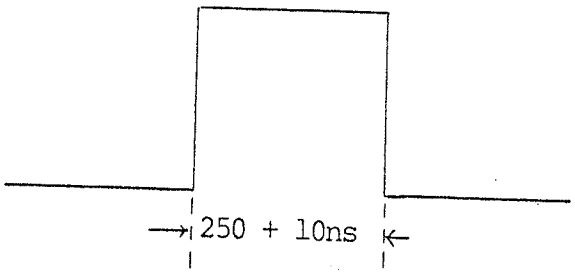


250 ± 1KHz at steady state

Pin 29 of FDC - 500ns ± 20ns



Pin 31 of FDC - 250ns ± 10ns



3.6.5 FDC (2797 IC5) TESTING AND ADJUSTMENT PROCEDURE

- Jumper, J30 should be set to VCC at power up.
- After power on, switch J30 to the other position so that FDC pin 22 (TEST) is connected to GND.
- Observe FDC pin 31 with a CRO.
- The CRO is set to 50ns/div and 0.5V/div.
- Adjust J15 (10KVR) so that the pulse width on FDC pin 31 is 250ns ± 10ns.
- Set the CRO to 100ns/div and 0.5V/div
- Observe FDC pin 29.
- Adjust J16 (50 KVR) so that the pulse width on FDC pin 29 is 500ns ± 20ns.
- Use a frequency counter to measure the frequency on FDC pin 16.
- Adjust J14 (variable cap) to set the frequency to 250KHz ± 1KHz.
- Switch the jumper back to the original position.
- Power off.

3.6.6 BONDWELL 2 SWITCHING POWER SUPPLY TESTING AND CALIBRATING PROCEDURES :-

i. Calibration spec. of switching power supply module

* The following tests are under the conditions of having DUMMY LOAD and HEAT SINK.

- supply voltage : 9.5V min. to 16V max.
  +12.00V nominal
- line regulation (5V§2A) : 20mV (typ.)
- load regulation (5V§4A) : 15mV (typ.)
- output rise time : 100ms (typ.)
- RESET delay time : 100ms (typ.)
- current spec. : (max. operating)
  +5V §2A
  +12V §700mA
  -13V §-45mA
- output current limit :
  +5V --> 4A (protected)
  -13V --> -230mA (protected)
- (1) pin (11) of L296 : 60KHz ± 10KHz Sawtooth
  pin (2) of L296 : 60KHz ± 10KHz square wave
  (2) pin (3) of TL497AC : ramp waveform
  pin (7) of TL497AC : square waveform
  <Detail waveforms are shown in page 2>
- nominal output voltage :
  #1 +12V §700mA
  #2 +5V §2A
  #3 -13V §-45mA
- output voltage variation :
  +5V (4.9V to 5.2V) o/c
  +5V (4.75V to 5.25V) §2A
  +12V (9.5 to 16V) §700mA
  -13V (-12.00V to -13.50V) §-45mA
- output ripple :
  +5V §2A 150mV p-p
  -13V §-45mA 150mV p-p
- heat sink temperature : under 90 °C (operating)

- PCB dimensions : 11.7cm x 6.9cm

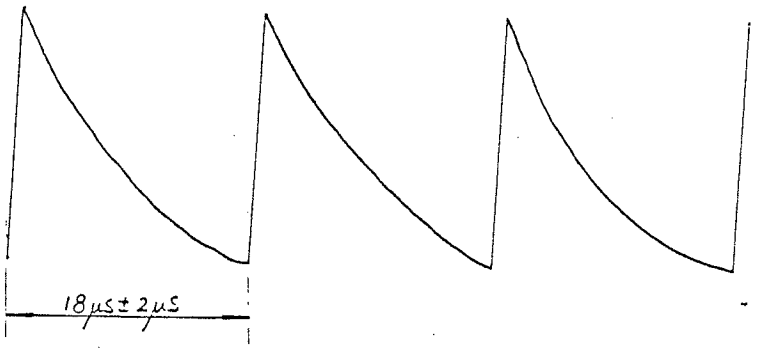
- Final test for LVOP SHOULD obtain the following results :

  LED flashing voltage : 9.6V to 10.2V
  LED lit voltage : 11V to 11.6V

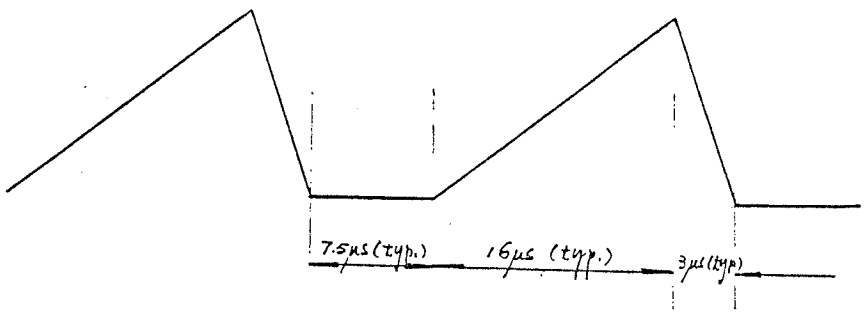
  (When switching on the whole unit HANDBOOK, the LED must lit at all trials.)

Detail waveforms :

a) At pin(11) of L296,



b) At pin(3) of TL497AC,



ii. Calibrating Procedures :

- a) #3 -13V output voltage calibration :

  1) Set supply voltage of power board to 15V ±0.5V.

  2) With the output at full load (-45mA), trim the output voltage to -13V ±0.1V.

  3) Vary the supply voltage to 9.5V +0.2V, output voltage SHOULD NOT be less than -12.00V.

- b) Low voltage calibration :

  1) Set supply voltage of power board to 9.80V sharp with a DVM.

  2) Probe pin (4) of IC1(ICL8211), LVOP, and use a plastic screw driver to trim VR1 in one direction to obtain a 'logic 1' (4.5V min).

  3) With LVOP STILL at 'logic 1', trim VR1 in reverse direction very slowly until pin (4) of IC1 JUST switch to 'logic 0'.

  4) Fix the value of VR1 for subsequent testes.

* Final test for LVOP SHOULD obtain the following results :

LED flashing voltage (V_L) : 9.6V to 1.2V
LED lit voltage (V_H) : 11.0V to 11.6V

iii. Testing procedure for low volt output, LVOP :

a) Set supply voltage to +13V ±0.5V with a DVM.

b) Probe pin (4) of IC1(ICL8211) to note a 'logic 1'.

c) Gradually decrease supply voltage to power board until pin (4) JUST switches to 'logic 0'.

d) Check the value of the supply voltage, V_L
It must be : 9.6V < V < 10.2V

e) In the same way, increase the supply voltage gradually until pin (4) JUST switches back to 'logic 1'.

f) Check the value of the supply voltage, V_H
It should be : 11.0V < V < 11.6V

3.6.7 DISK DRIVE

1. Power Consumption

+12V	: Standby 1mA (max.)		
	Average at read/write	:	85mA (typ.)
			180mA (max.)
	Average at seek	:	200mA (typ.)
			260mA (max.)
	Peak at spindle motor start	:	300mA (max.)
+5V	: Average at read /write	:	140mA (typ.)
			170mA (max.)
	Average at seek	:	100mA (typ.)
			120mA (max.)

2. Motor speed : 300 r.p.m. $\pm$ 1%

3. Alignment specification (using a commercially available alignment disk).

Radial alignment lobe rate : 80% min
Index to data burst : 200 usec + 100 usec
Hysteresis : 20% max
Head azimuth angle : $\pm$ 18°
Asymmetry (track 0, track 39) : 600 nsec

CHAPTER 4 PROGRAMMABLE DEVICES

This chapter describes some of the programmable devices in the Bondwell 2 from the programming point of view. All the necessary information are included. However, the reader should be familiar with the Bondwell 2 and the Z-80 assembly language programming.

4.1 8253, THE PROGRAMMABLE INTERVAL TIMER

To the programmer, the 8253 counter/timer appears as four memory locations or I/O ports. These are selected as follows :

$\overline{CS}$	A0	A1	Function
0	0	0	Select counter/timer 0
0	0	1	Select counter/timer 1
0	1	0	Select counter/timer 2
0	1	1	Select Control Register

Since the Read Control signal RD is tied permantly to +5V by hardware, the 8253 is limited to the following manipulation :

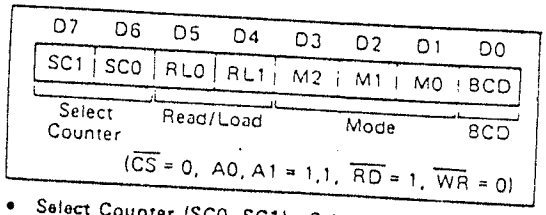
$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A1	A0	Function
0	1	0	0	0	Load counter/timer 0
0	1	0	0	1	Load counter/timer 1
0	1	0	1	0	Load counter/timer 2
0	1	0	1	1	Write Mode Word
0	1	1	X	X	No Operation, 3-state

The modes for each counter are programmed by selecting each timer-register address and writing the correct control word for format is shown in the following chart.

4.1.1 8253 PROGRAMMABLE INTERVAL TIMER CONTROL WORD

Control Word and Count Value Program

Each counter operating mode is set by control word programming. The control word format is outlined below.



($\overline{CS}$ = 0, A0, A1 = 1,1, $\overline{RD}$ = 1, $\overline{WR}$ = 0)

- **Select Counter (SC0, SC1):** Selection of set counter

SC1	SC0	Set Contents
0	0	Counter = 0 selection
0	1	Counter = 1 selection
1	0	Counter = 2 selection
1	1	Illegal combination

- **Read/Load (RL1, RL0):** Count value Reading/Loading format setting

RL1	RL0	Set Contents
0	0	Counter Latch operation
0	1	Reading/Loading of Least Significant byte (LSB)
1	0	Reading/Loading of Most Significant byte (MSB)
1	1	Reading/Loading of LSB followed by MSB

- **BCD:** Operation count mode setting

BCD	Set Contents
0	Binary Count (16-bits Binary)
1	BCD Count (4-decades Binary Coded Decimal)

- **Mode (M2, M1, M0):** Operation waveform mode setting

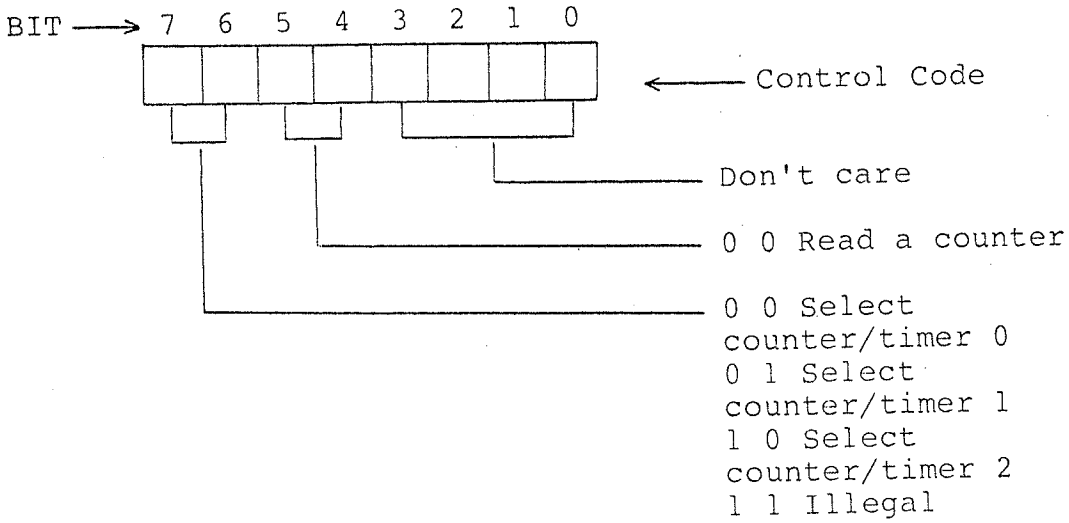
M2	M1	M0	Set Contents
0	0	0	Mode 0 (Interrupt on Terminal Count)
0	0	1	Mode 1 (Programmable One-Shot)
x	1	0	Mode 2 (Rate Generator)
x	1	1	Mode 3 (Square Wave Generator)
1	0	0	Mode 4 (Software Triggered Strobe)
1	0	1	Mode 5 (Hardware Triggered Strobe)

x denotes "not specified".

4.1.2 MONITORING 8253 COUNTER/TIMER OPERATION

The 8253 counter/timer has no Status register. Therefore, operation is monitored by reading the contents of any counter register, at any time. Below shows a reading technique.

If you want to read a counter's contents while it is decrementing, then you must write the following command code to the counter/timer.



Writing in the following instruction sequence, you will read the contents of counter on the fly.

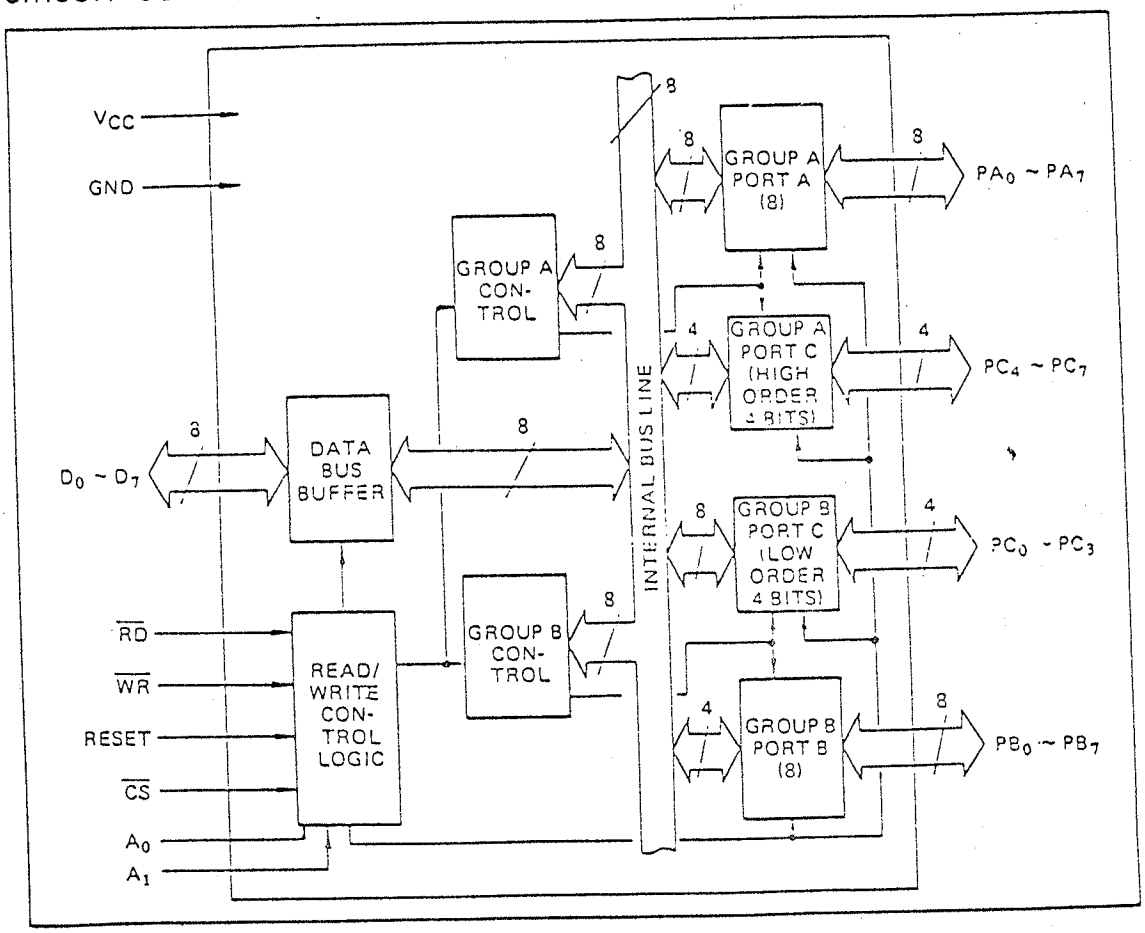
```
LD      A, 40H ; output control code to read
OUT     (7), A ; counter 1 contents
IN      A, (5) ; Read counter 1 contents and
LD      C, A   ; store in BC
IN      A, (5)
LD      B, A
```

4.2 82C55, CMOS PROGRAMMABLE PERIPHERAL INTERFACE

This device provides effective means to interface peripheral equipment such as the keyboard and a high speed printer to the Z80 CPU through the 24-bit I/O pins, which is equivalent to three 8-bit bidirectional buses, and the control lines.

The following diagram shows the circuit configuration.

CIRCUIT CONFIGURATION



BASIC FUNCTIONAL DESCRIPTION

Group A and Group B

When setting a mode to a port having 24 bits, set it by dividing it into two groups of 12 bits each.

Group A: Port A (8 bits) and high order 4 bits of port C (PC7 ~ PC4)

Group B: Port B (8 bits) and low order 4 bits of port C (PC3 ~ PC0)

Mode 0, 1, 2

There are 3 types of modes to be set by group as follows:

Mode 0: Basic input operation/output operation (Available for both groups A and B)

Mode 1: Stroke input operation/output operation (Available for both groups A and B)

Mode 2: Bidirectional bus operation (Available for group A only)

When used in mode 1 or mode 2, however, port C has bits to be defined as ports for control signal for operation ports (port A for group A and port B for group B) of their respective groups.

Port A, B, C

The internal structure of 3 ports is as follows:

Port A: One 8-bit data output latch/buffer and one 8-bit data input latch

Port B: One 8-bit data input/output latch/buffer and one 8-bit data input buffer

Port C: One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input)

Single bit set/reset function for port C

When port C is defined as output port, it is possible to set (to turn to high level) or reset (to turn to low level) any one of 8 bits individually without affecting other bits.

OPERATIONAL DESCRIPTION

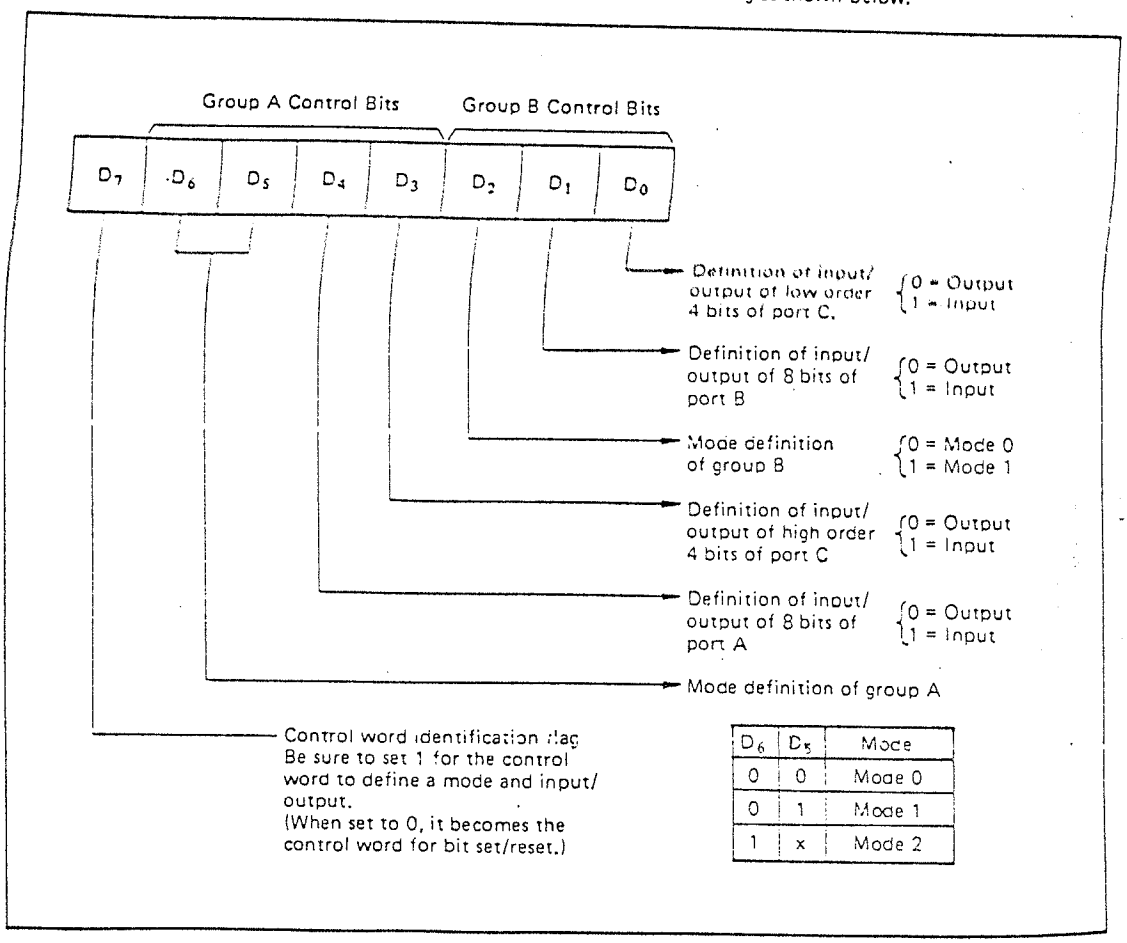
Control Logic

Operations by addresses and control signals, e.g., read and write, etc. are as shown in the table below:

Operation	A1	A0	$\overline{CS}$	$\overline{WR}$	$\overline{RD}$	Operation
Input	0	0	0	1	0	Port A → Data Bus
	0	1	0	1	0	Port B → Data Bus
	1	0	0	1	0	Port C → Data Bus
Output	0	0	0	0	1	Data Bus → Port A
	0	1	0	0	1	Data Bus → Port B
	1	0	0	0	1	Data Bus → Port C
Control	1	1	0	0	1	Data Bus → Control Register
Others	1	1	0	1	0	Illegal Condition
	x	x	1	x	x	Data bus is in the high impedance status.

Setting of Control Word

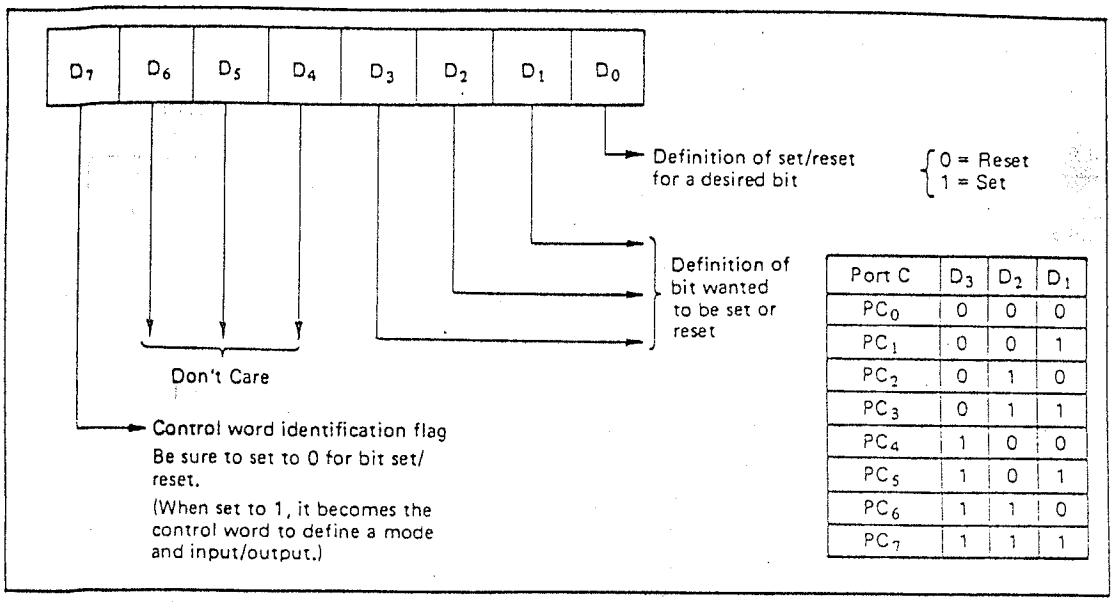
The control register is composed of 7-bit latch circuit and 1-bit flag as shown below.



D_6	D_5	Mode
0	0	Mode 0
0	1	Mode 1
1	x	Mode 2

Bit Set/Reset Function

When port C is defined as output port, it is possible to set (set output to 1) or reset (set output to 0) any one of 8 bits without affecting other bits



Port C	D_3	D_2	D_1
PC_0	0	0	0
PC_1	0	0	1
PC_2	0	1	0
PC_3	0	1	1
PC_4	1	0	0
PC_5	1	0	1
PC_6	1	1	0
PC_7	1	1	1

Interrupt Control Function

When MSM82C55A-5 is used in mode 1 or mode 2, the interrupt signal for CPU is provided. The interrupt request signal is output from port C. When the internal flip-flop INTE is set beforehand at this time, the desired interrupt request signal is output. When it is reset beforehand, however, the interrupt request signal is not output. The set/reset of the internal flip-flop is made by the bit set/reset operation for port C virtually.

Bit set → INTE is set → Interrupt allowed

Bit reset → INTE is reset → Interrupt inhibited

Operational Description by Mode

1. Mode 0 (Basic input/output operation)

Mode 0 makes MSM82C55A-5 operate as a basic input port or output port. As no control signal such as interrupt request, etc. is required in this mode. All of 24 bits can be used as two-8-bit ports and two 4-bit ports. Sixteen combinations are then possible for inputs/outputs. The inputs are not latched, but the outputs are.

Type	Control Word								Group A		Group B	
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Port A	High Order 4 Bits of Port C	Port B	Low Order 4 Bits of Port C
1	1	0	0	0	0	0	0	0	Output	Output	Output	Output
2	1	0	0	0	0	0	0	1	Output	Output	Output	Input
3	1	0	0	0	0	0	1	0	Output	Output	Input	Output
4	1	0	0	0	0	0	1	1	Output	Output	Input	Input
5	1	0	0	0	1	0	0	0	Output	Input	Output	Output
6	1	0	0	0	1	0	0	1	Output	Input	Output	Input
7	1	0	0	0	1	0	1	0	Output	Input	Input	Output
8	1	0	0	0	1	0	1	1	Output	Input	Input	Input
9	1	0	0	1	0	0	0	0	Input	Output	Output	Output
10	1	0	0	1	0	0	0	1	Input	Output	Output	Input
11	1	0	0	1	0	0	1	0	Input	Output	Input	Output
12	1	0	0	1	0	0	1	1	Input	Output	Input	Input
13	1	0	0	1	1	0	0	0	Input	Input	Output	Output
14	1	0	0	1	1	0	0	1	Input	Input	Output	Input
15	1	0	0	1	1	0	1	0	Input	Input	Input	Output
16	1	0	0	1	1	0	1	1	Input	Input	Input	Input

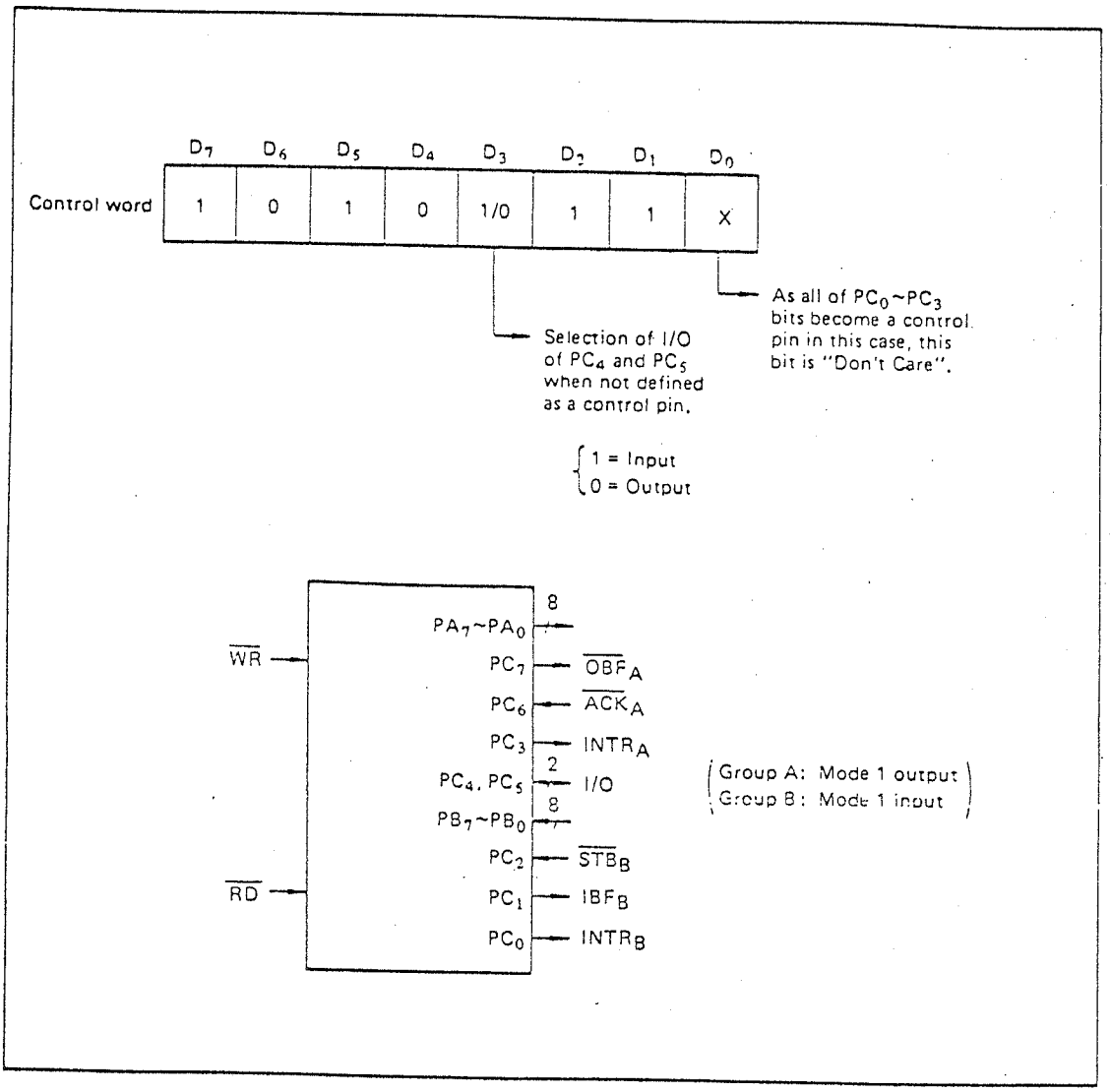
Note: When used in mode 0 for both groups A and B

Port C Function Allocation in Mode 1

Combination of Input/Output / Port C	Group A: Input Group B: Input	Group A: Input Group B: Output	Group A: Output Group B: Input	Group A: Output Group B: Output
PC_0	$INTR_B$	$INTR_B$	$INTR_B$	$INTR_B$
PC_1	IBF_B	$\overline{OBF}_B$	IBF_B	$\overline{OBF}_B$
PC_2	$\overline{STB}_B$	$\overline{ACK}_B$	$\overline{STB}_B$	$\overline{ACK}_B$
PC_3	$INTR_A$	$INTR_A$	$INTR_A$	$INTR_A$
PC_4	$\overline{STB}_A$	$\overline{STB}_A$	I/O	I/O
PC_5	IBF_A	IBF_A	I/O	I/O
PC_6	I/O	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$
PC_7	I/O	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$

Note: I/O is a bit not used as the control signal, but it is available as a port of mode 0.

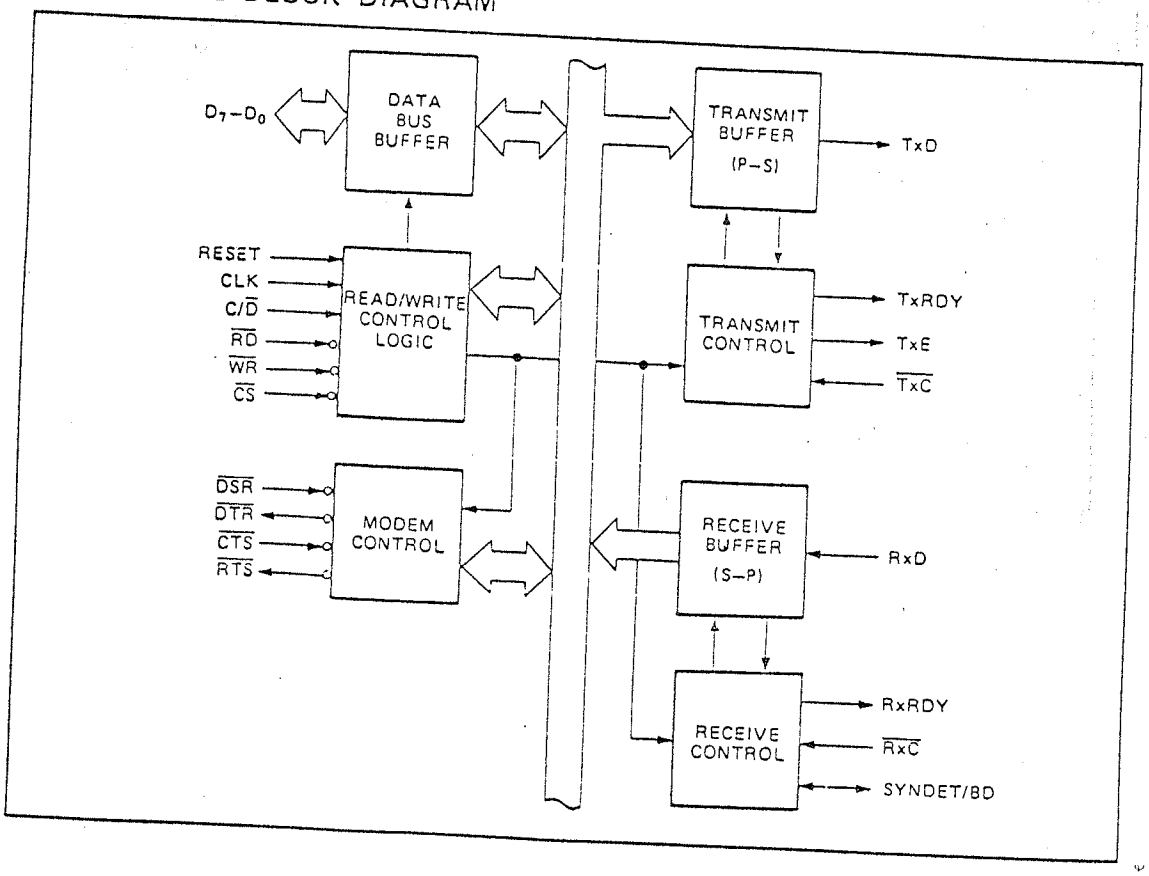
Examples of the relation between the control words and pins when used in mode 1 is shown below:
(a) When group A is mode 1 output and group B is mode 1 input.



4.3 82C51 USART (UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER)

The functional block diagram of the USART is shown in the following figures.

FUNCTIONAL BLOCK DIAGRAM



FUNCTION

Outline

MSM82C51A's functional configuration is programed by the software.

Operation between MSM82C51A and CPU is executed by program control. Table 1 shows the operation between CPU and the device.

Table 1 Operation between MSM82C51A and CPU

$\overline{CS}$	C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	
1	X	X	X	Data bus 3-state
0	X	1	1	Data bus 3-state
0	1	0	1	Status → CPU
0	1	1	0	Control word ← CPU
0	0	0	1	Data → CPU
0	0	1	0	Data ← CPU

It is necessary to execute a function-setting sequence after resetting on MSM82C51A. Fig. 1 shows the function-setting sequence.

If the function was set, the device is ready to receive a command, thus enabling the transfer of data by setting a necessary command, reading a status and reading/writing data.

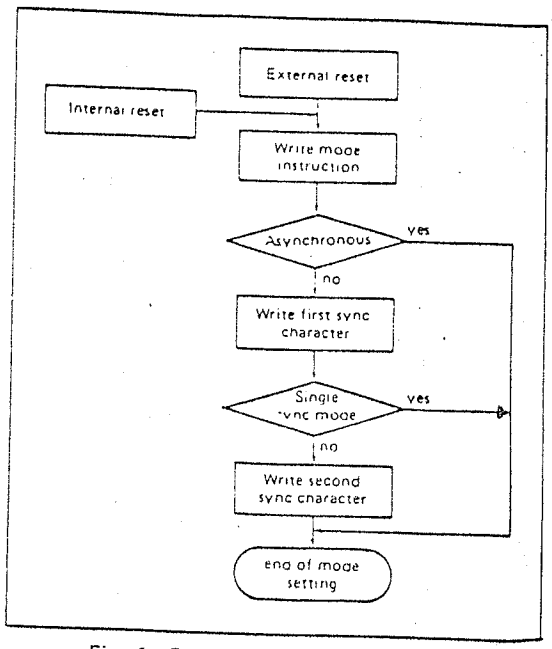


Fig. 1 Function-Setting Sequence (Mode Instruction Sequence)

Control Words

There are two types of control word.

1. Mode instruction (setting of function)
2. Command (setting of operation)

1) **Mode Instruction**

Mode instruction is used for setting the function of MSM82C51A. Mode instruction will be in "wait for write" at either internal reset or external reset. That is, the writing of control word after resetting will be recognized as "mode instruction."

Items to be set by mode instruction are as follows:

- Synchronous/asynchronous mode
- Stop bit length (asynchronous mode)
- Character length
- Parity bit
- Baud rate factor (asynchronous mode)
- Internal/external synchronization (synchronous mode)
- No. of synchronous characters (synchronous mode)

The bit configuration of mode instruction is shown in Fig.'s 2 and 3. In the case of synchronous mode, it is necessary to write one- or two-tyte sync characters.

If sync characters were written, a function will be set because the writing of sync characters constitutes part of mode instruction.

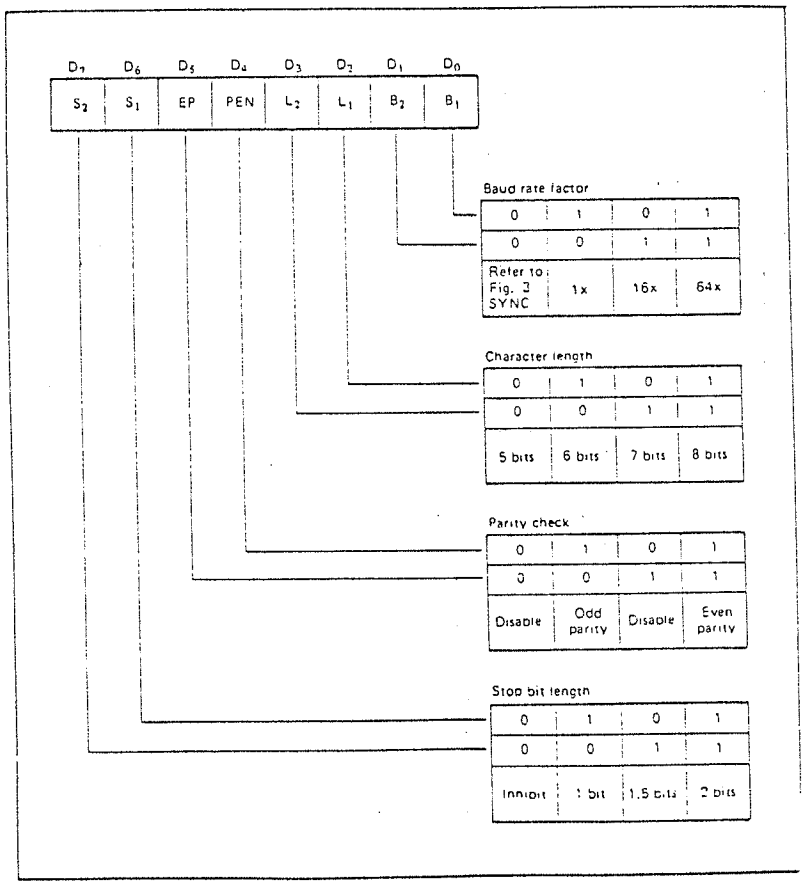


Fig. 2 Bit Configuration of Mode Instruction (Asynchrnous)

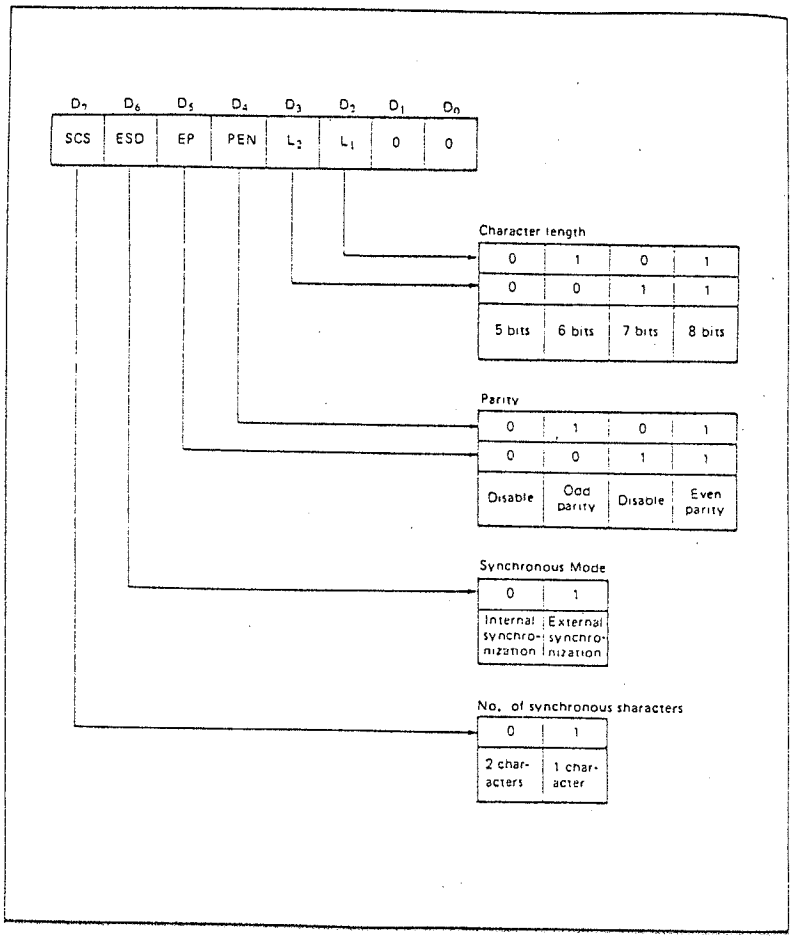


Fig. 3 Bit Configuration of Mode Instruction (Synchronous)

2) **Command**

Command is used for setting the operation of MSM82C51A.

It is possible to write a command whenever necessary after writing mode instruction and sync characters.

Items to be set by command are as follows:

- Transmit Enable/Disable
- Receive Enable/Disable
- $\overline{DTR}$, $\overline{RTS}$ Output of data.
- Resetting of error flag.
- Sending of break characters
- Internal resetting
- Hunt mode (synchronous mode)

The bit configuration of a command is shown in Fig. 4.

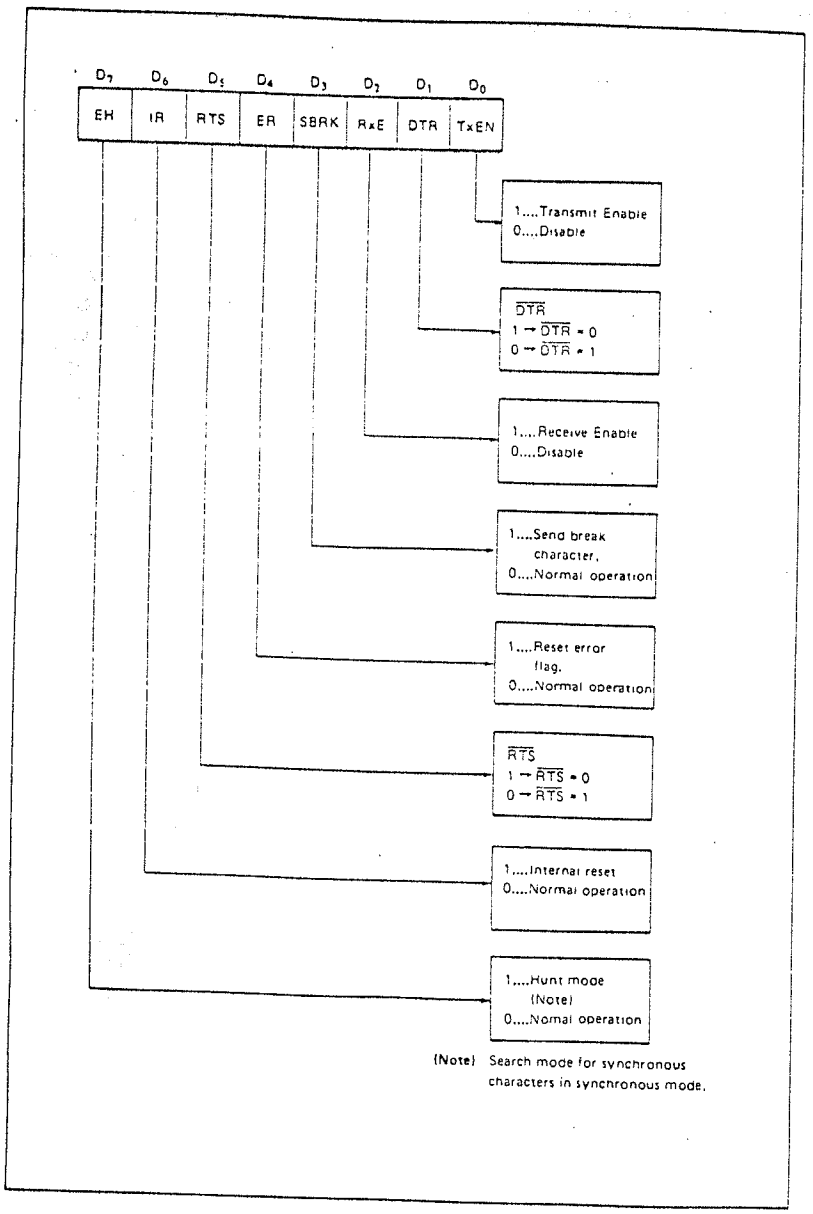


Fig. 4 Bit Configuration of Command

Standby Status

It is possible to put MSM82C51A in "standby status" for the complete static configuration of CMOS.

It is when the following conditions have been satisfied that MSM82C51A is in "standby status."

(1) $\overline{CS}$ terminal shall be fixed at Vcc level.

(2) Input pins other than $\overline{CS}$, D_0 to D_7, $\overline{RD}$, $\overline{WR}$ and C/$\overline{D}$ shall be fixed at Vcc or GND level (including SYNDET in external synchronous mode).

Note When all outputs current are 0, ICCS specification is applied.

4.4 MSM6255 LCD CONTROLLER

The following is the full list of the control code and instruction register of the MSM6255.

(1) Mode control

By writing "OOH" in the instruction register, the mode control register can be assigned.

Register	A0	D7	D6	D5	D4	D3	D2	D1	D0	
Instruction register	1	0	0	0	0	0	0	0	0	Read-able
Mode control register	0	0	MODE DATA							

D6	D5	D4	D3	D2	D1	D0	Output system		
I/O	I/O	I/O	I/O	0	0	0	Normal	1-bit serial	Character display
				1	0		ODD/EVEN		
				x	1		4-bit parallel		
				x	1				
				0	0	0	Normal	1-bit serial	Graphics
				1	0		ODD/EVEN		
				x	1		4-bit parallel		
				x	1				

Blinking time | Cursor ON/OFF | Cursor blinking | Display ON/OFF | ODD/EVEN | 4-bit parallel/1-bit serial | Mode

Display ON/OFF
1: Display ON
0: Display OFF

Cursor blinking

D5	D4	
0	0	Cursor OFF
0	1	Cursor OFF
1	0	Cursor ON
1	1	Cursor blinking

Blinking time
1: 16 frames
0: 32 frames
} Half of blinking frequency

(2) Character pitch setting

Register	A0	D7	D6	D5	D4	D3	D2	D1	D0	
Instruction register	1	0	0	0	0	0	0	0	1	Read-able
Character pitch register	0	(V_p-1)				0	(H_p-1)			Read-able

H_p represents the number of bits indicated in one byte indication data sent from RAM. The value of H_p is the following five types.

H_p	D2	D1	D0	
4	0	1	1	Horizontal character pitch 4
5	1	0	0	" 5
6	1	0	1	" 6
7	1	1	0	" 7
8	1	1	1	" 8

(3) Character number setting (per line)

	A0	D7	D6	D5	D4	D3	D2	D1	D0	
Instruction register	1	0	0	0	0	0	0	1	0	
Character number register	0	0	(H_z-1)							Read-able

Assuming the total dot number to the horizontal direction of the display is N_H,

$$N_H = H_p \times H_N \text{ , where } H_N = 2 \text{ to } 128.$$

The maximum value of N_H = 8 x 128 = 128 bytes = 1,024 dots.

(4) Time sharing number setting (display duty)

Register	A0	D7	D6	D5	D4	D3	D2	D1	D0	
Instruction register	1	0	0	0	0	0	0	1	1	Read-able
Time sharing register	0	(N_x-1)								

N_x = 1 to 256

(5) Cursor shape setting

Register	A0	D7	D6	D5	D4	D3	D2	D1	D0	
Instruction register	1	0	0	0	0	0	1	0	1	Read-able
Cursor position register	0	$(C_{pu}-1)$				$(C_{pd}-1)$				Read-able

In the character display (character mode), on lines from C_{pu} to C_{pd}, the cursor is indicated. For example, when $C_{pu} = C_{pd} = 8$ is specified while 5 x 7 dot font is used, the cursor is displayed below a character.

The length of the cursor to the horizontal direction becomes the same as the character pitch to the horizontal direction : H_p. If Cpu, Cpd $\leq V_p$, the cursor is indicated with the highest priority. whereas, Cpu, Cpd $> Vp$, the cursor is not indicated. This setting is invalud in the graphic mode.

(6) Display start lower address setting

Register	A0	D7	D6	D5	D4	D3	D2	D1	D0	
Instruction register	1	0	0	0	0	0	1	0	1	Read-able
Indication start address register (lower byte)	0	Lower start address								Read-able

(7) Display start upper address setting

Register	A0	D7	D6	D5	D4	D3	D2	D1	D0	
Instruction register	1	0	0	0	0	0	1	1	0	Read-able
Indication start address register (upper byte)	0	Upper start address								Read-able

The display start address shows an address of the RAM which stores data indicated at the left end and the most upper position. The start address is composed of upper and lower 8 bits (16 bits in total) in the graphic mode. In the character display mode, it is composed of the lower 6 bits (D5 to D0) of the upper address and the lower 8 bits (14 bits in total). In the case, the upper 2 bits of the upper address are ignored.

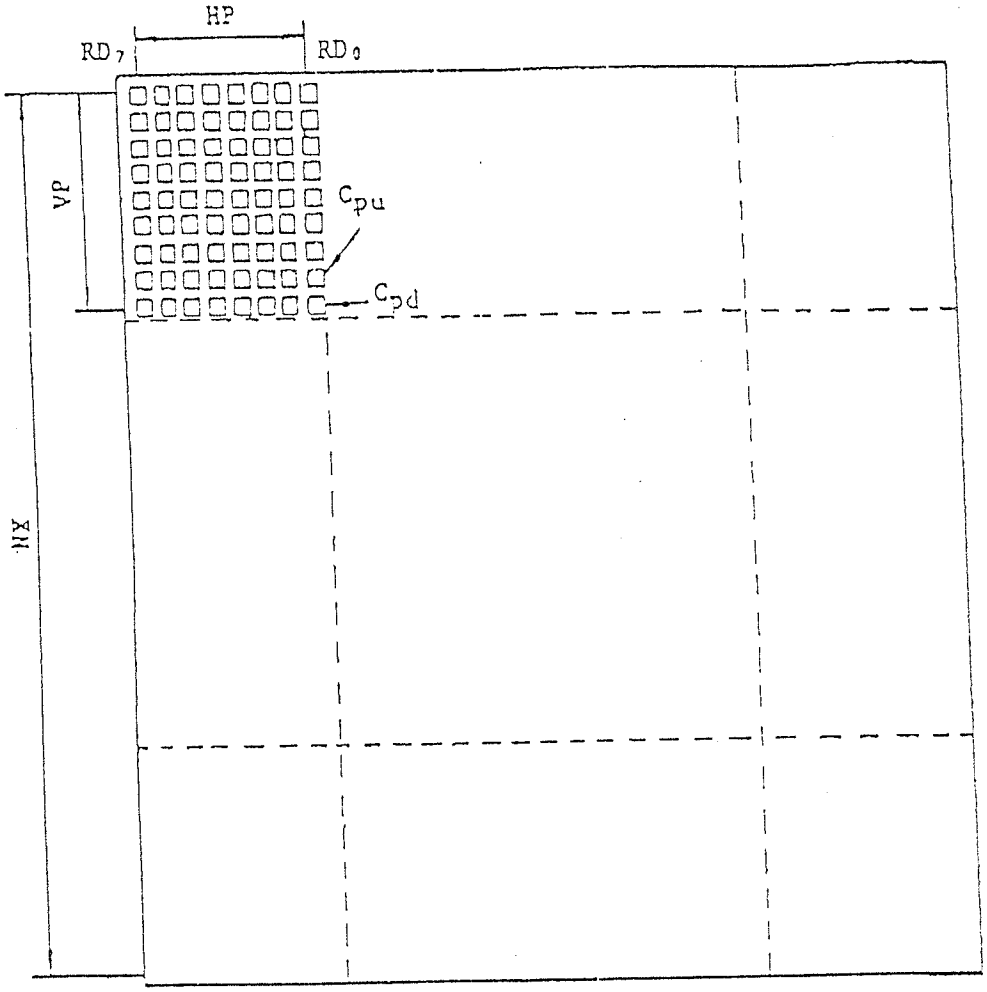
(8) Cursor's lower address setting

Register	A0	D7	D6	D5	D4	D3	D2	D1	D0	
Instruction register	1	0	0	0	0	0	1	1	1	Read-able
Cursor address register (lower byte)	0	(Lower cursor address)								

(9) Cursor's address setting

Register	A0	D7	D6	D5	D4	D3	D2	D1	D0	
Instruction register	1	0	0	0	0	1	0	0	0	Read-able
Cursor address register (upper byte)	0	(Upper cursor address)								Read-able

By means of this instruction, the value of the cursor address is written in the cursor address register. In the graphic mode, the cursor is indicated at the position specified by the cursor address register.



Symbol	Name	Meaning	Value
Hp	Horizontal character pitch	Pitch of character to the horizontal direction	6 to 8 dots
H_N	Horizontal character number	Number of characters/words per line	2 to 128 characters
Vp	Vertical character pitch	Pitch of character to the vertical direction	1 to 16 dots
Cpu	Cursor start position	A position where the cursor starts display	1 to 16 lines
Cpd	Cursor end position	A position where the cursor stops display	1 to 16 lines
N_x	Vertical direction line number	Display duty	1 to 256

CHAPTER 5

SPARE PARTS PRICE LIST

ORDER NO	DESCRIPTION	FOB HK (US$)
IC04HC	IC 74HC04	0.30
IC04HCU	IC 74HCU04	0.30
IC08HC	IC 74HC08	0.30
IC138HC	IC 74HC138	0.50
IC1488	IC 1488	0.40
IC1489	IC 1489	0.40
IC157HC	IC 74HC157	0.50
IC161HC	IC 74HC161	0.60
IC245HC	IC 74HC245	1.10
IC273HC	IC 74HC273	0.90
IC2797	FLOPPY DISK CONTROLLER	9.50
IC32HC	IC 74HC32	0.30
IC6255	MSM6255GS LCD CONTROLLER	9.40
IC74HC	IC 74HC74	0.40
IC80L	LOW POWER Z80L CPU 2.5MH	5.20
IC82C51	MSM 82C51ARS USART	2.60
IC82C53	MSM82C53-5RS TIMER	2.60
IC82C55	MSM82C55A-5RS PPI	2.60
IL07	IC 7407	0.30
IL14LS	IC 74LS14	0.20
IL38LS	IC 74LS38	0.20
IL45	IC 7445	0.50
IM2732-45	IC2732 EPROM 450NS	2.40
IM416420	4164 D. RAM 200NS	0.60
IM611620	6116 STATIC RAM 200NS	1.80
MC32016	16MHZ X'TAL	0.50
SPBW0201-02	MAIN PBC VER 1.6	7.30
IA78L12	78L12 REGULATOR	0.30
IC555C	CMOS IC555	0.30
ICL296	ICL296 SWITCHING REGULAT	3.00
ICL8211	ICL8211 VOLT DESTESTOR	1.50
ICTL497AC	ICTL497AC SWITCHING RGLR	1.10
ID0006	IN5822 SWITCHING DIODE	0.50
IT0004	TRANSISTOR 9014B	0.10
IT0012	TRANSISTOR 9015B	0.10
MS023	5V REED RELAY SPST	1.00
RV2501H	5KOHM VR (H TYPE)	0.10
RV3201H	20K OHM VR (H TYPE)	0.10
SPBW0202-01	POWER PCB BOARD	0.30

DL003	2X5 LED RED	0.10
MP021-2	BW02 MALE ADAPTOR JACK	0.10
SPBW0203-01	KEYBOARD PCB	1.10
WN020001	CABINET TOP	2.20
WN020002	CABINET BOTTOM	2.20
WN020003	DISPLAY FRONT CABINET	0.50
WN020004	DISPLAY BACK CABINET	0.60
WN020005	HINGE	0.10
WN020006	HINGE PLASTIC BUSHING	0.10
WN020007	HANDLE	0.20
WN020008	I/O DOOR	0.10
WN020009	DISPLAY LOCK HOOK	0.10
WN020011	CARTRIDGE TOP	0.40
WN020017-1	BW2 MODULE LENS	1.10
YSBW0201	3.5" FLOPPY DISK DRIVE	86.00
YSBW0202	LCD MODULE CG640200G	123.40
YSBW0203	BATTERY 6V 3AH RECHARGE	5.30

Effective Date : 1st September, 1985.

CHAPTER 6 CIRCUIT SCHEMATIC AND COMPONENT LAYOUT

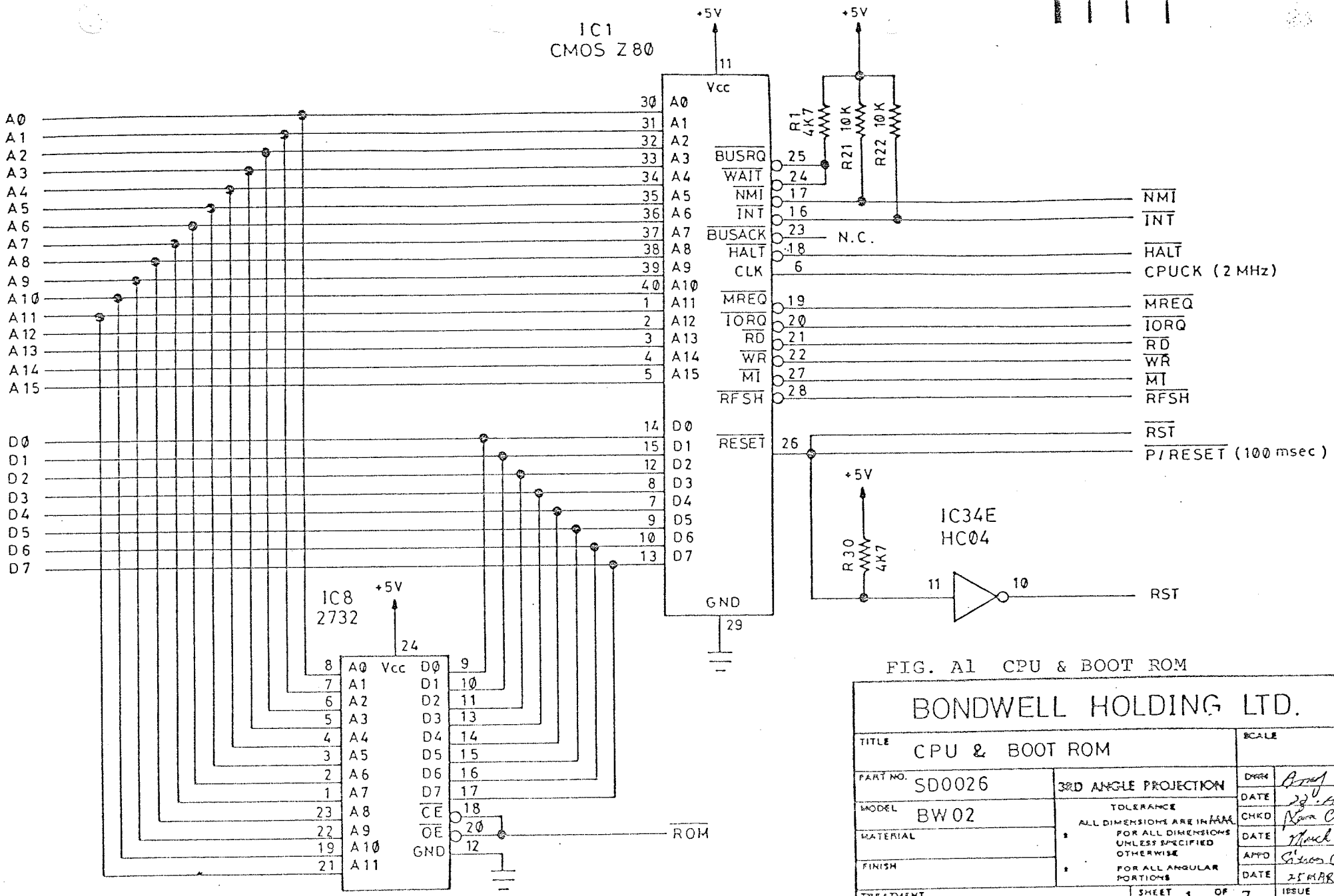


FIG. A1 CPU & BOOT ROM

BONDWELL HOLDING LTD.

TITLE: CPU & BOOT ROM			SCALE
PART NO. SD0026	3RD ANGLE PROJECTION	DRWN	Anny Yeung
MODEL BW02	TOLERANCE ALL DIMENSIONS ARE IN MM	DATE	22.MAR.85
MATERIAL	± FOR ALL DIMENSIONS UNLESS SPECIFIED OTHERWISE	CHKD	Kwan Chi Kwan
FINISH	± FOR ALL ANGULAR PORTIONS	DATE	March 22, 1985
TREATMENT	SHEET 1 OF 7	APPD	Gibson Chang
		DATE	25 MAR 85
		ISSUE	

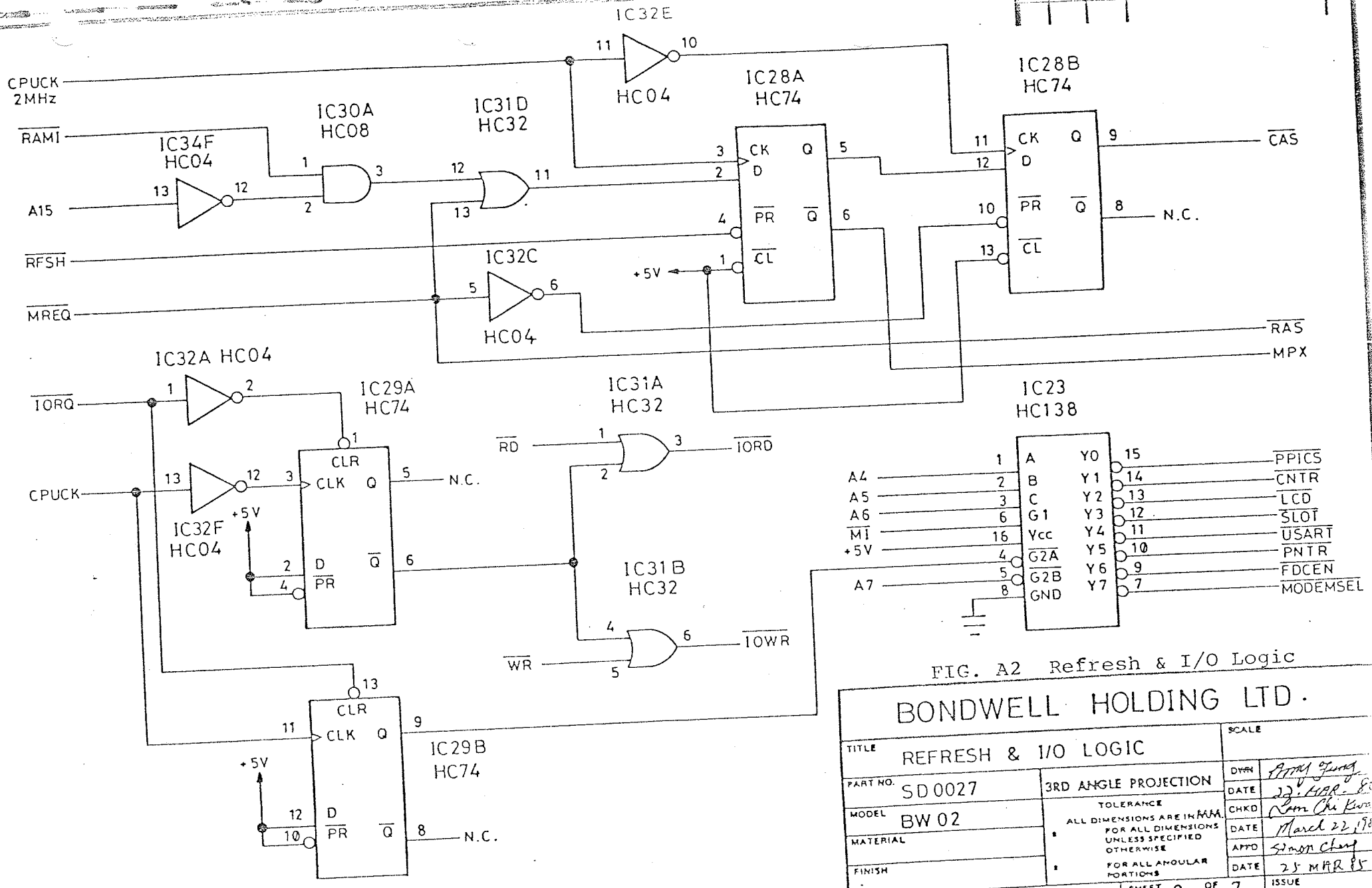


FIG. A2 Refresh & I/O Logic

BONDWELL HOLDING LTD.

TITLE: REFRESH & I/O LOGIC		SCALE	
PART NO. SD0027	3RD ANGLE PROJECTION	DWN	Benny Tsang
MODEL BW02	TOLERANCE: ALL DIMENSIONS ARE IN MM. FOR ALL DIMENSIONS UNLESS SPECIFIED OTHERWISE ±	DATE	22 MAR 85
MATERIAL	FOR ALL ANGULAR PORTIONS ±	CHKD	Tom Chi Kwan
FINISH		DATE	March 22, 1985
TREATMENT	SHEET 2 OF 7	APPD	Simon Chang
		DATE	25 MAR 85
		ISSUE	

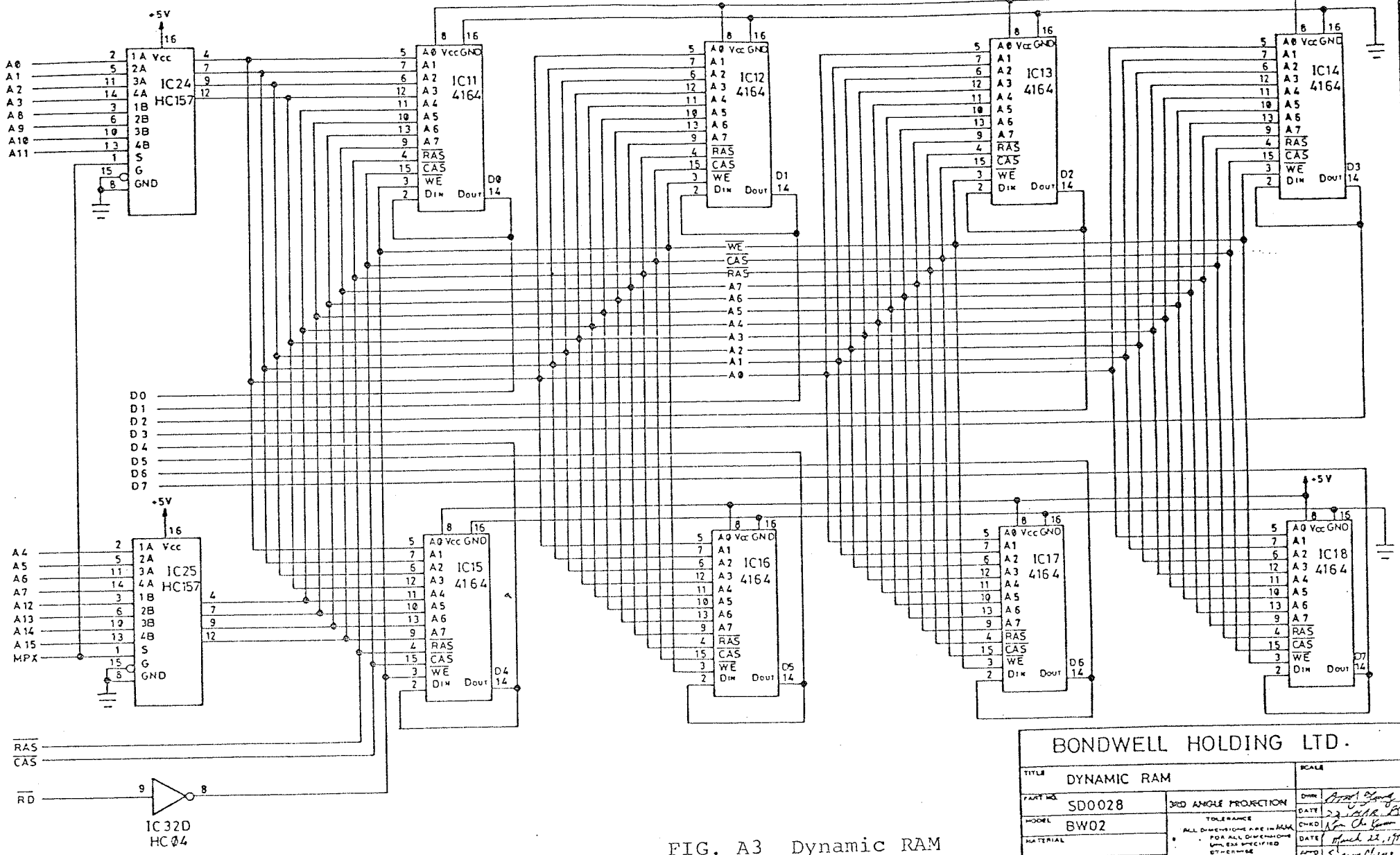


FIG. A3 Dynamic RAM

BONDWELL HOLDING LTD.

TITLE: DYNAMIC RAM		SCALE
PART NO: SD0028	3RD ANGLE PROJECTION	DWN: Arpad Kovacs, 23 MAR 82
MODEL: BW02	TOLERANCE: ALL DIMENSIONS ARE IN MM UNLESS SPECIFIED OTHERWISE. FOR ALL DIMENSIONS ± . FOR ALL ANGULAR POSITIONS ±	CHKD: Ian Chapman, DATE: March 22, 1983
MATERIAL		APPD: Simon Christ, DATE: 25 MAR 82
FINISH		
TREATMENT	SHEET 3 OF 7	ISSUE

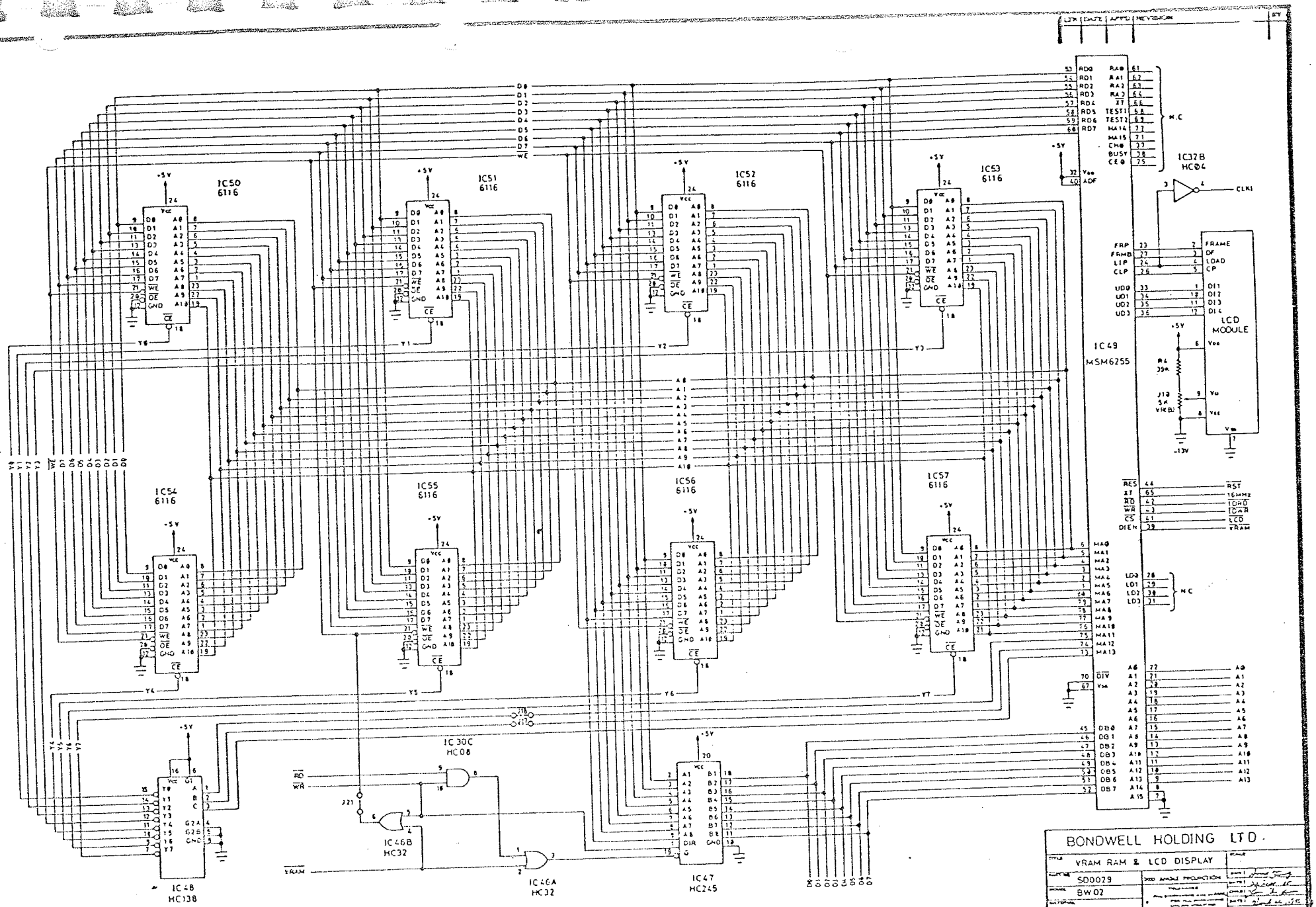


FIG. A4 VRAM & LCD Display

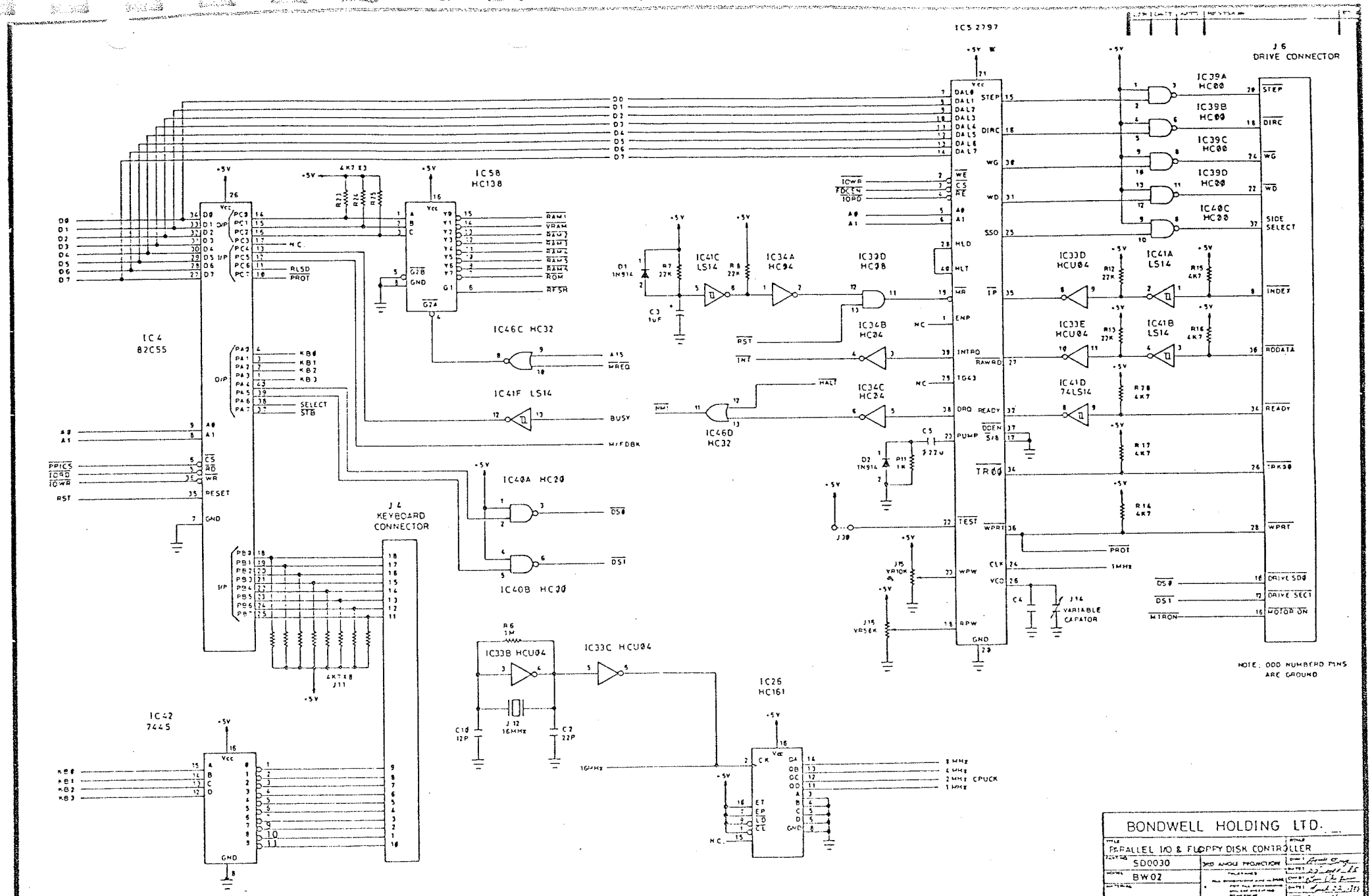
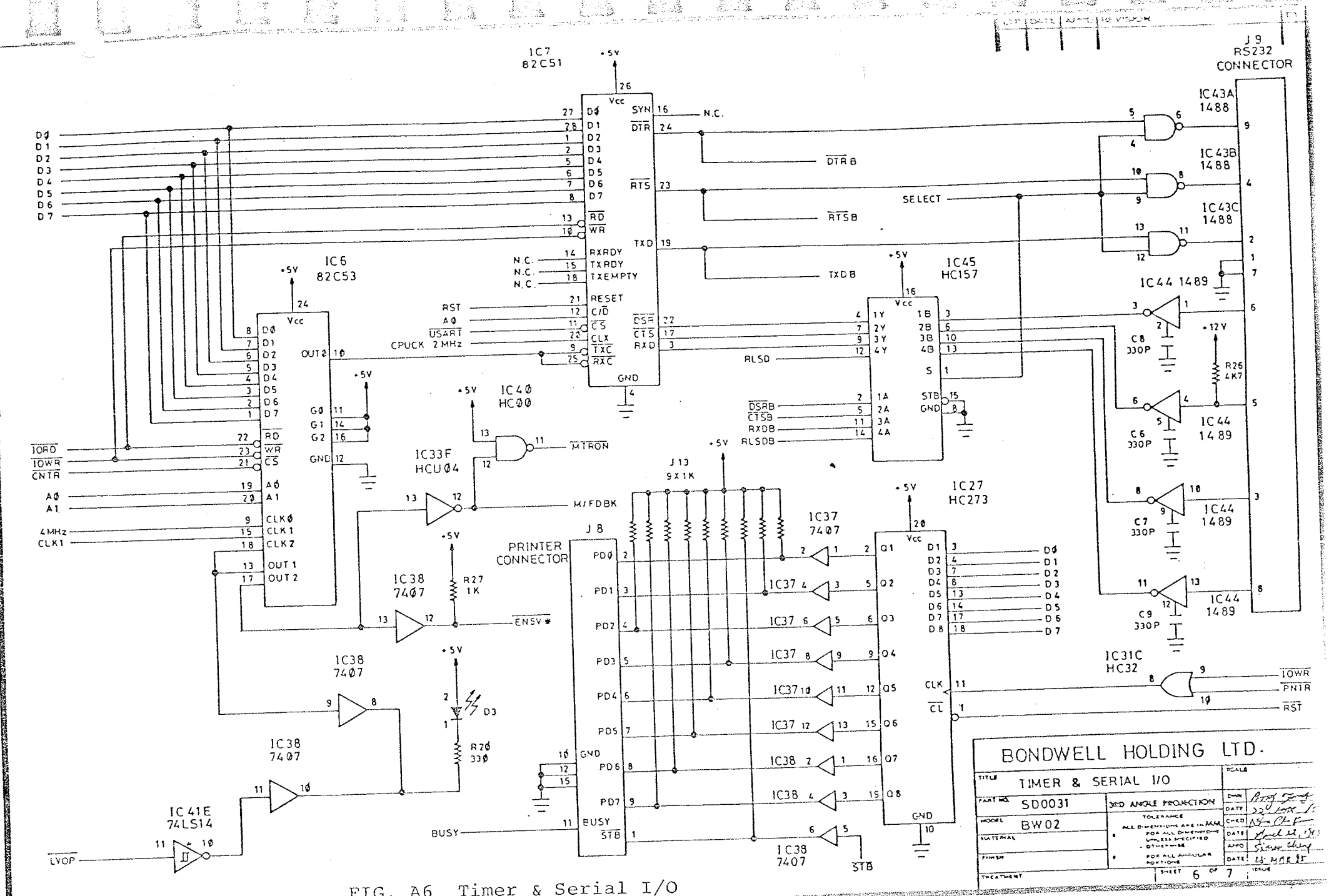


FIG. A5 Parallel I/O & Floppy Disk Controller



BONDWELL HOLDING LTD.

TITLE	TIMER & SERIAL I/O
PART NO.	SD0031
MODEL	BW02
MATERIAL	
FINISH	
TREATMENT	
3RD ANGLE PROJECTION	
TOLERANCE	ALL DIMENSIONS ARE IN MM. FOR ALL DIMENSIONS UNLESS SPECIFIED OTHERWISE ± FOR ALL ANGULAR ADDITIONS
SCALE	
DRWN	Barry Young
DATE	22 MAR 85
CHKD	Alan Chapman
DATE	March 22, 1985
APPD	Steve Ashley
DATE	22-MAR-85
SHEET	6 OF 7
ISSUE	

FIG. A6 Timer & Serial I/O

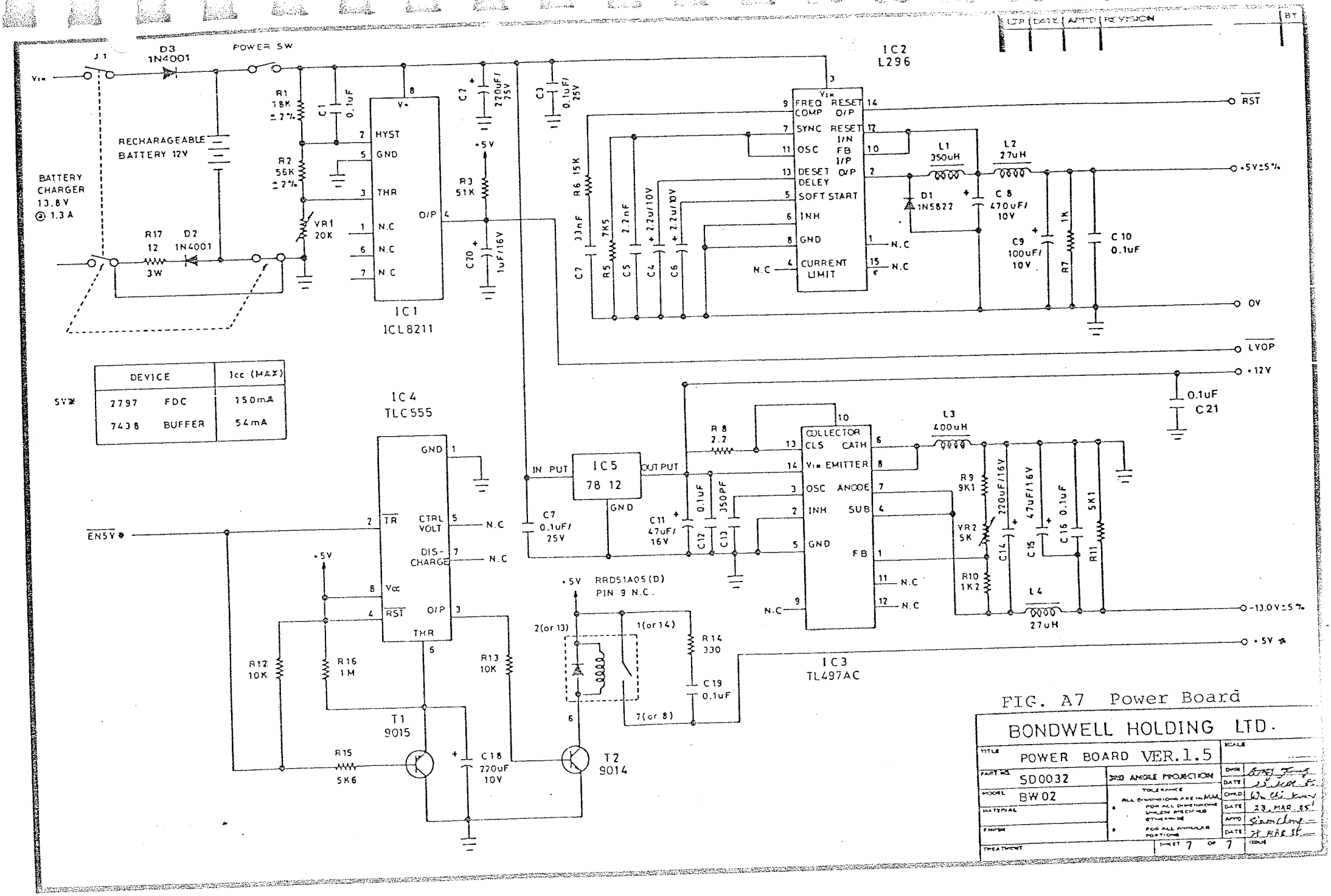
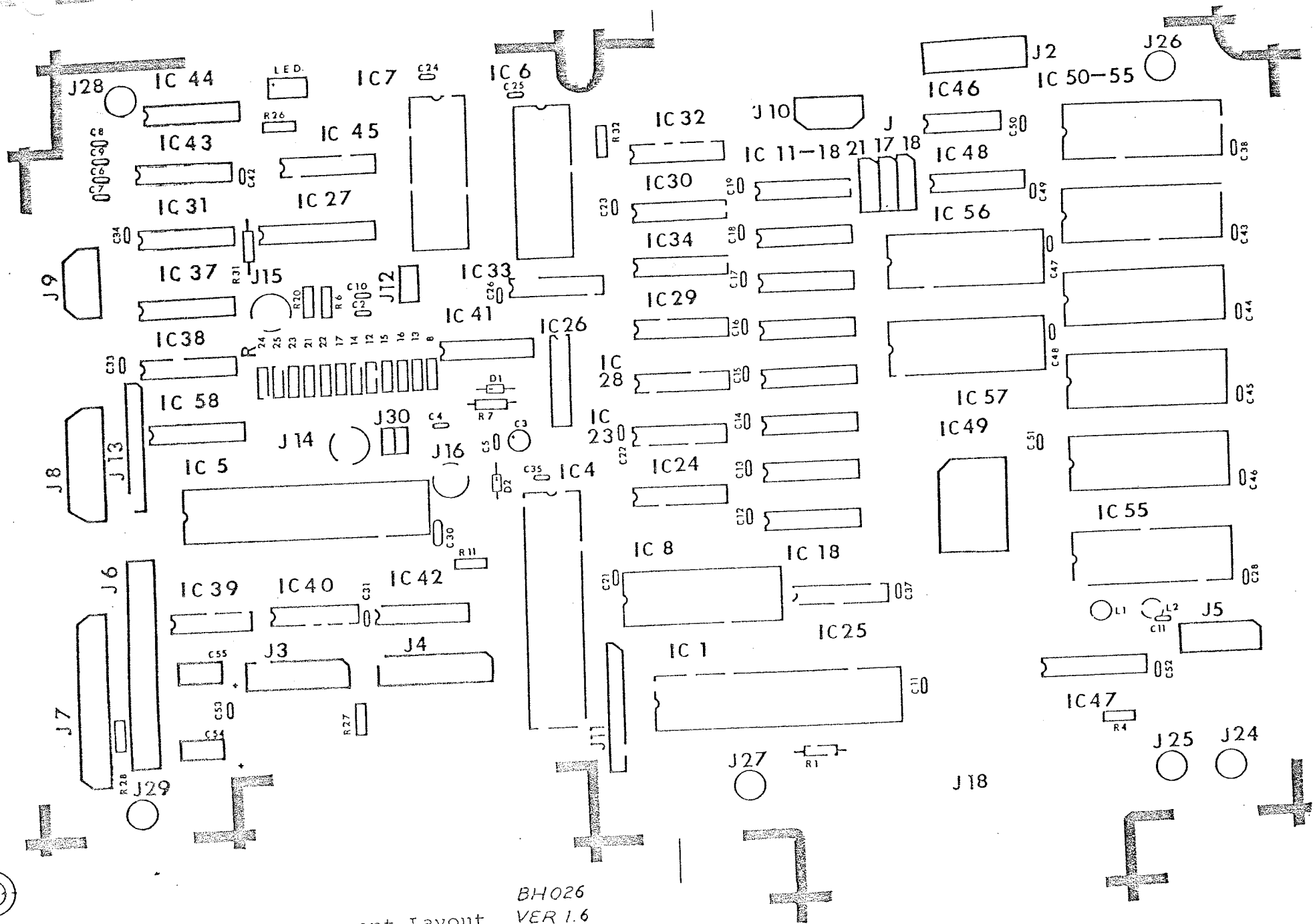


FIG. A7 Power Board

BONDWELL HOLDING LTD.

TITLE	POWER BOARD VER.1.5		SCALE
PART NO.	SD0032	3RD ANGLE PROJECTION	DRWN
MODEL	BW02	TOLERANCE	DATE 23 MAR 85
MATERIAL		ALL DIMENSIONS ARE IN MM	CHKD
FINISH		FOR ALL DIMENSIONS UNLESS SPECIFIED OTHERWISE	DATE 23 MAR 85
TREATMENT		FOR ALL ANGULAR POSITIONS	APPD
			DATE 24 APR 85
		SHEET 7 OF 7	ISSUE



BH026
VER 1.6

FIG. B1 Main PCB Component Layout

DRAWN BY : LAM CHI KWAN

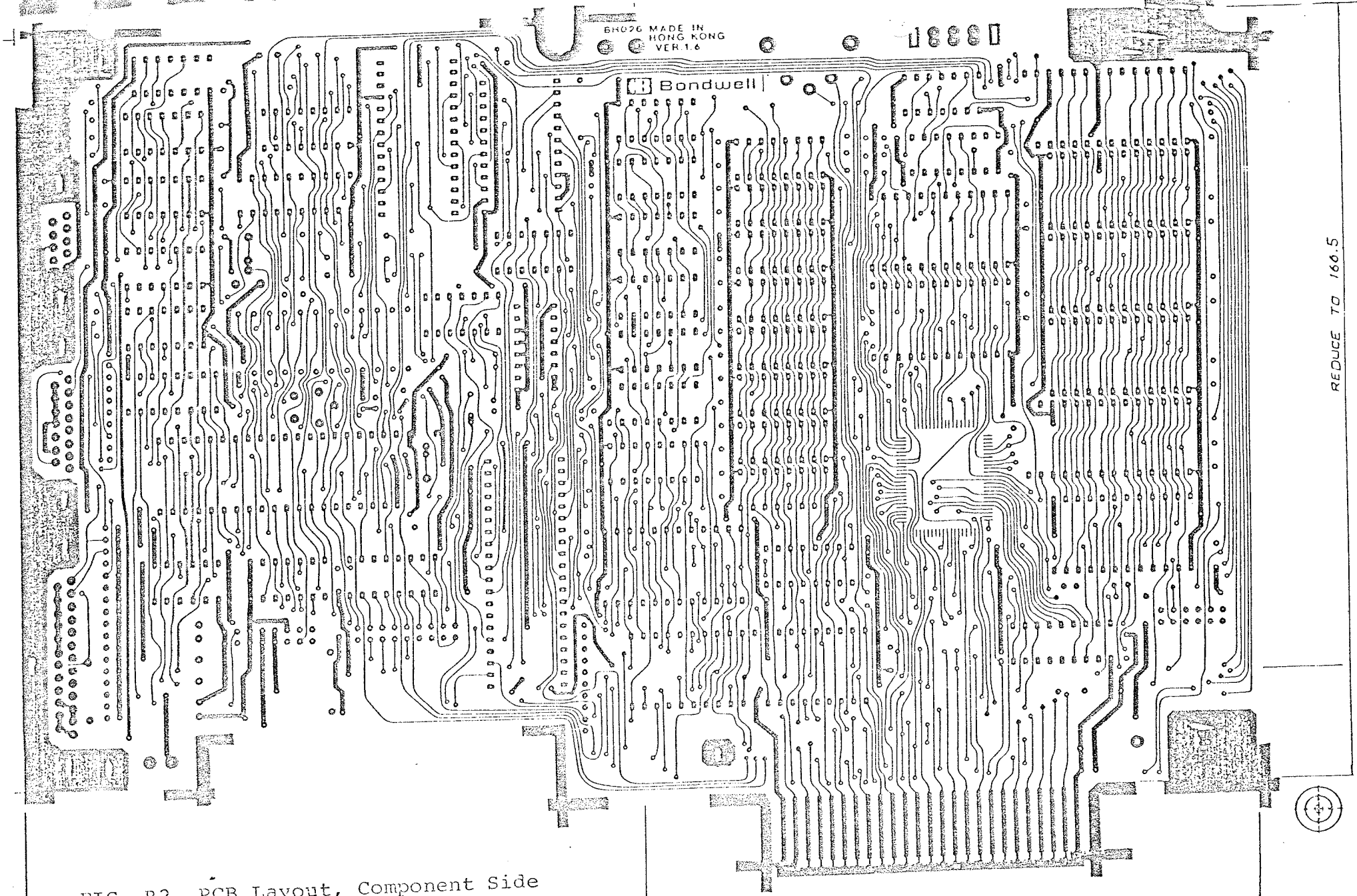
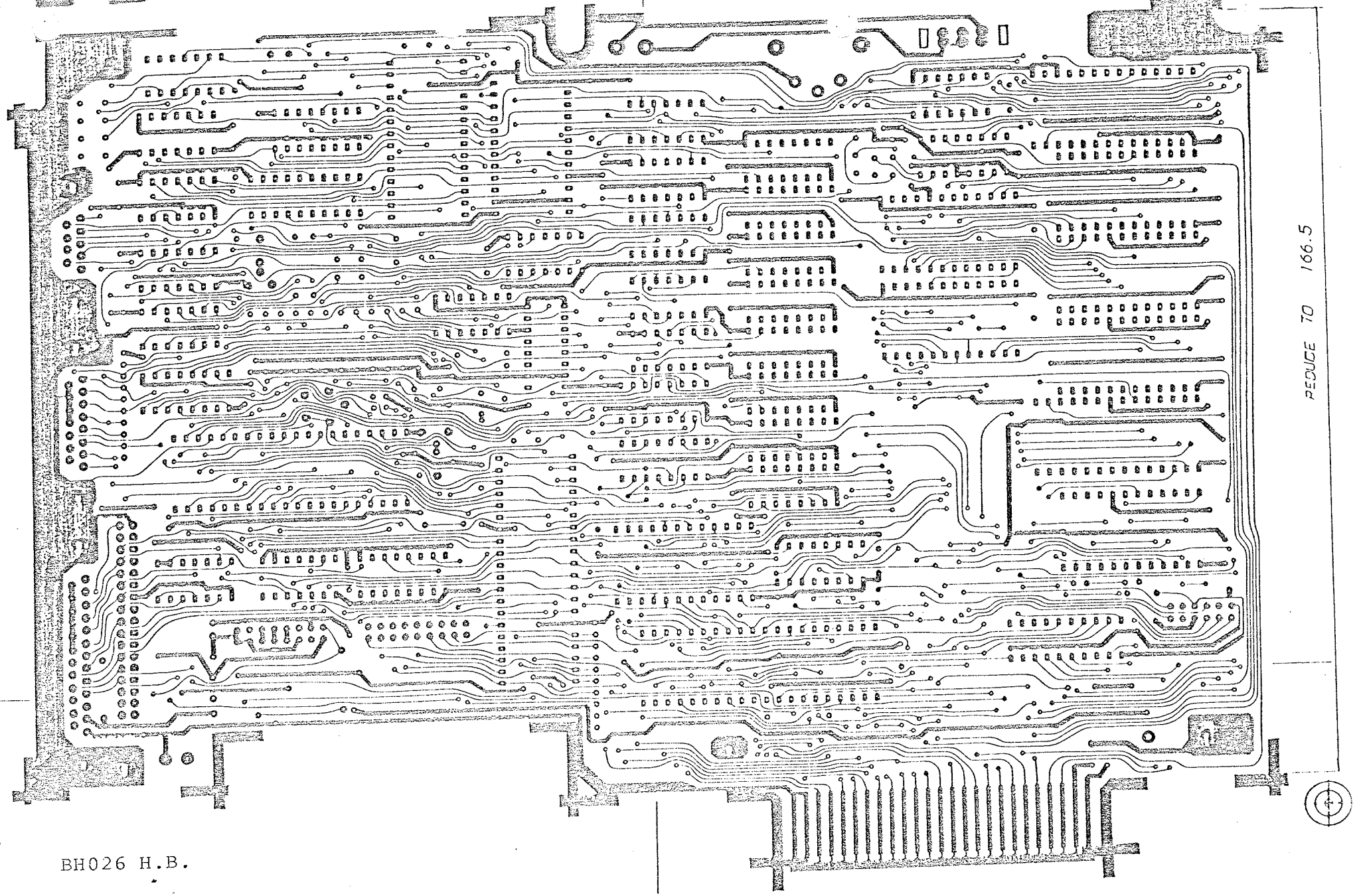


FIG. B2 PCB Layout, Component Side

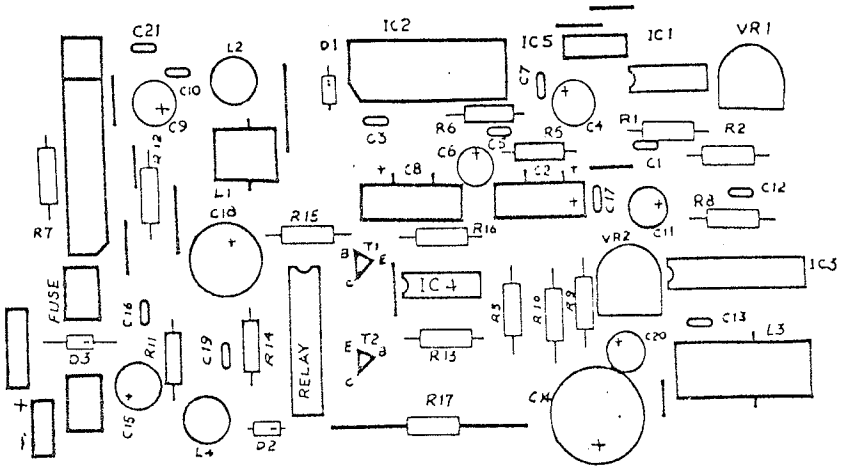
DRAWN BY : LAM CHI KWAN



BH026 H.B.

FIG. B3 PCB Layout, Solder Side

DRAWN BY : LAM CHI KWAN



BH026 POWER BOARD
VER 1.5

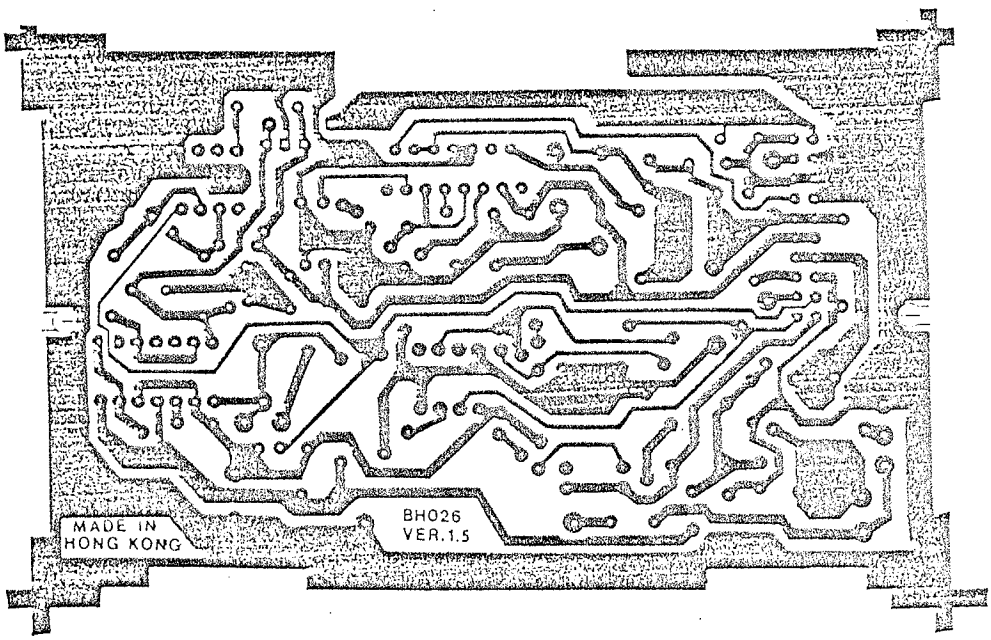


FIG. B4 Power Board Component & PCB Layout

DRAWN BY : LAM CHI KUEN

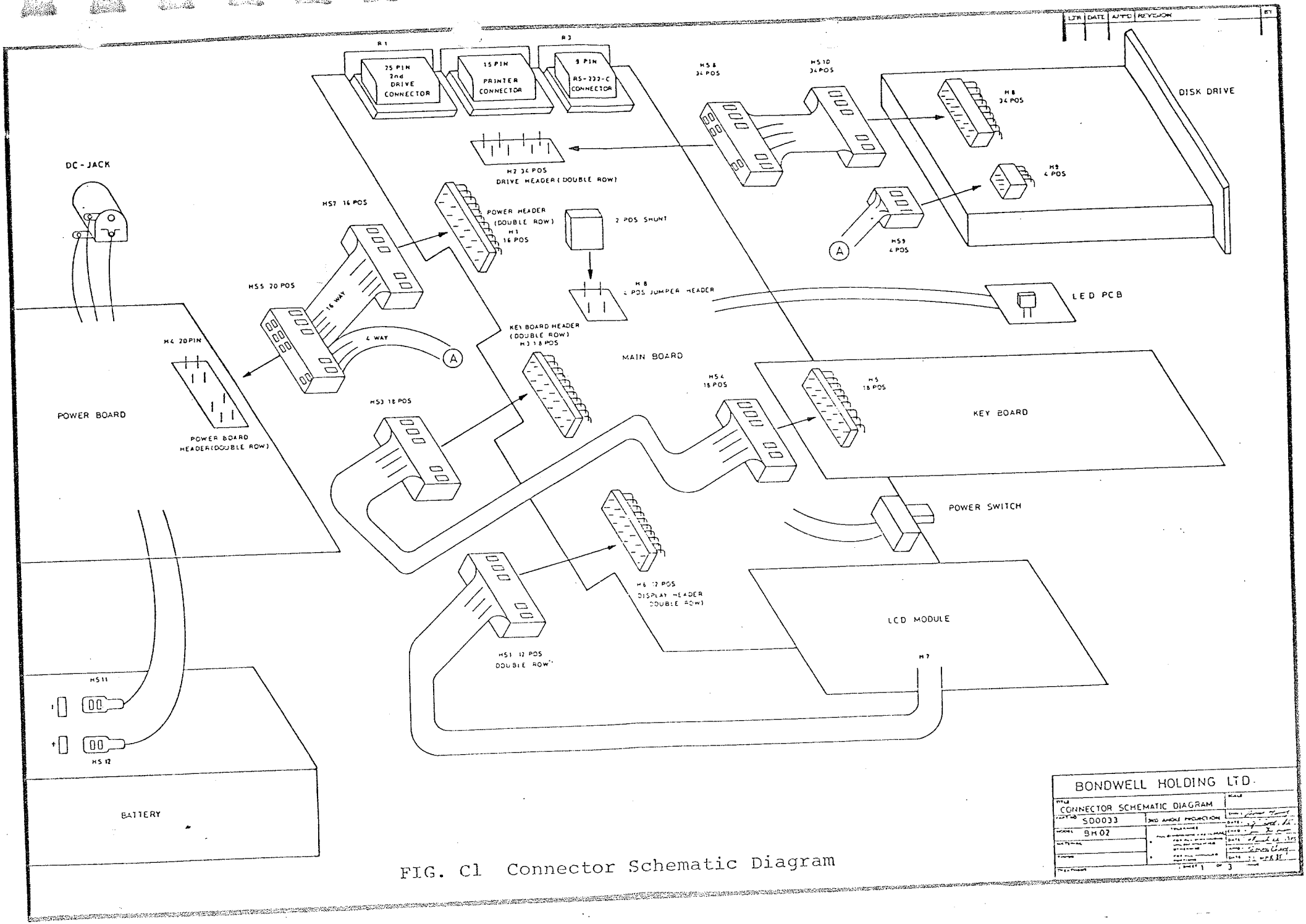


FIG. C1 Connector Schematic Diagram

DRAWN BY : LAM CHI KUEN

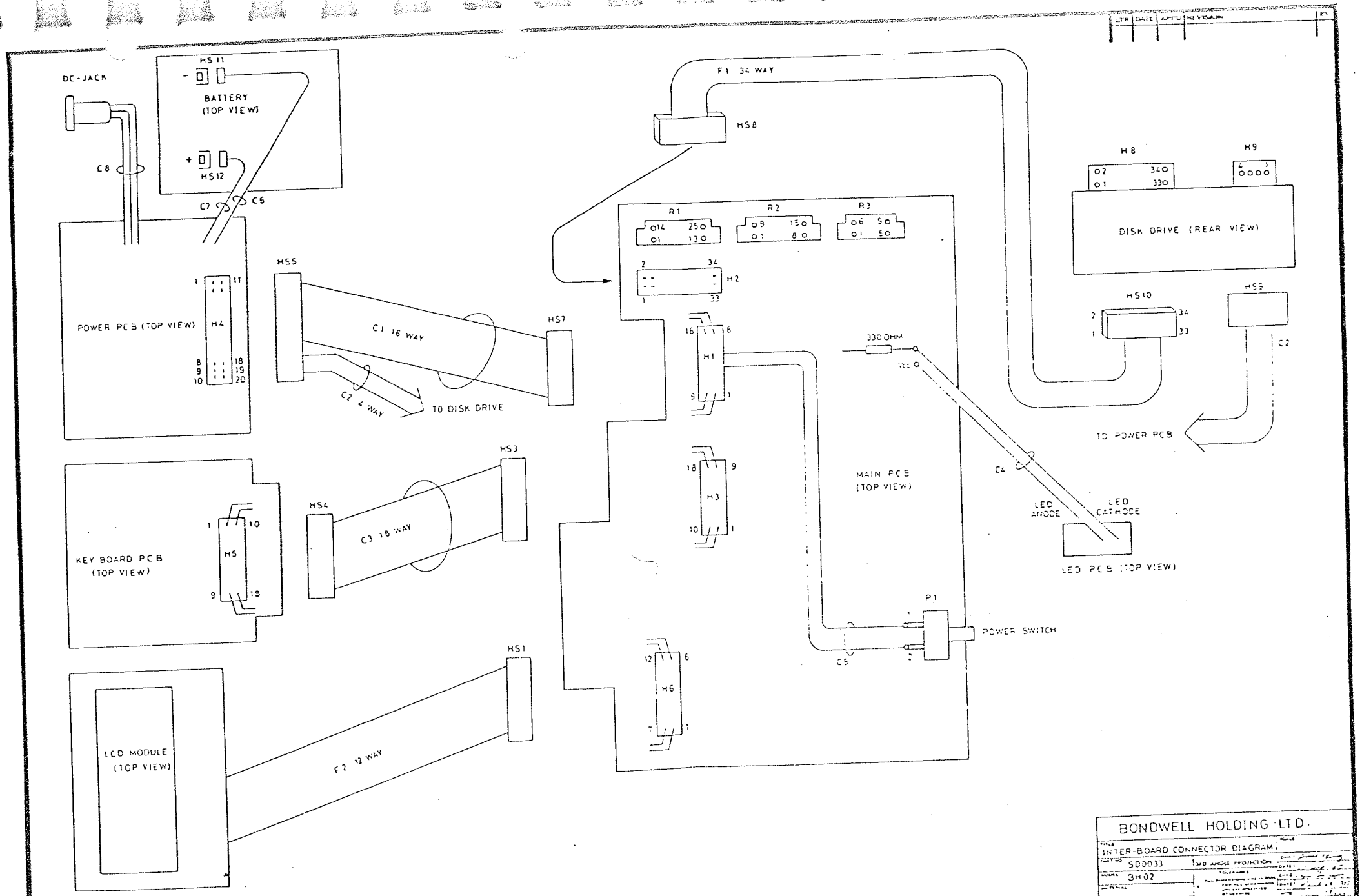


FIG. C2 Inter-Board Connector Diagram

RAM CARD for BW2 V.1.0

LTR	DATE	APP'D	REVISION	BY

H4 PIN NO.	H1 PIN NO.	H9 PIN NO.	SIGNAL
1	1		ENV5V *
2	2		-12 V
3	3		5 V
4	4		J 1
5	5		12V
6	6		5 V
7	7		$\overline{RST}$
8	8		GND
9		1	5V
10		2	GND
11	9		5V *
12	10		LVOP
13	11		GND
14	12		J 1
15	13		12V
16	14		5 V
17	15		GND
18	16		GND
19		3	GND
20		4	12V

H3 PIN NO.	H5 PIN NO.	SIGNAL
1	1	OUT 8
2	2	OUT 7
3	3	OUT 6
4	4	OUT 5
5	5	OUT 4
6	6	OUT 3
7	7	OUT 2
8	8	OUT 1
9	9	OUT Ø
10	10	OUT 9
11	11	PB7
12	12	PB6
13	13	PB5
14	14	PB4
15	15	PB3
16	16	PB2
17	17	PB1
18	18	PBØ

H6 PIN NO.	H7 PIN NO.	SIGNAL
1	6	-5V
2	4	LIP
3	2	FRP
4	7	GND
5	9	Vo
6	11	UD2
7	5	CLP
8	3	FRMB
9	1	UDØ
10	8	-12V
11	10	UD1
12	12	UD3

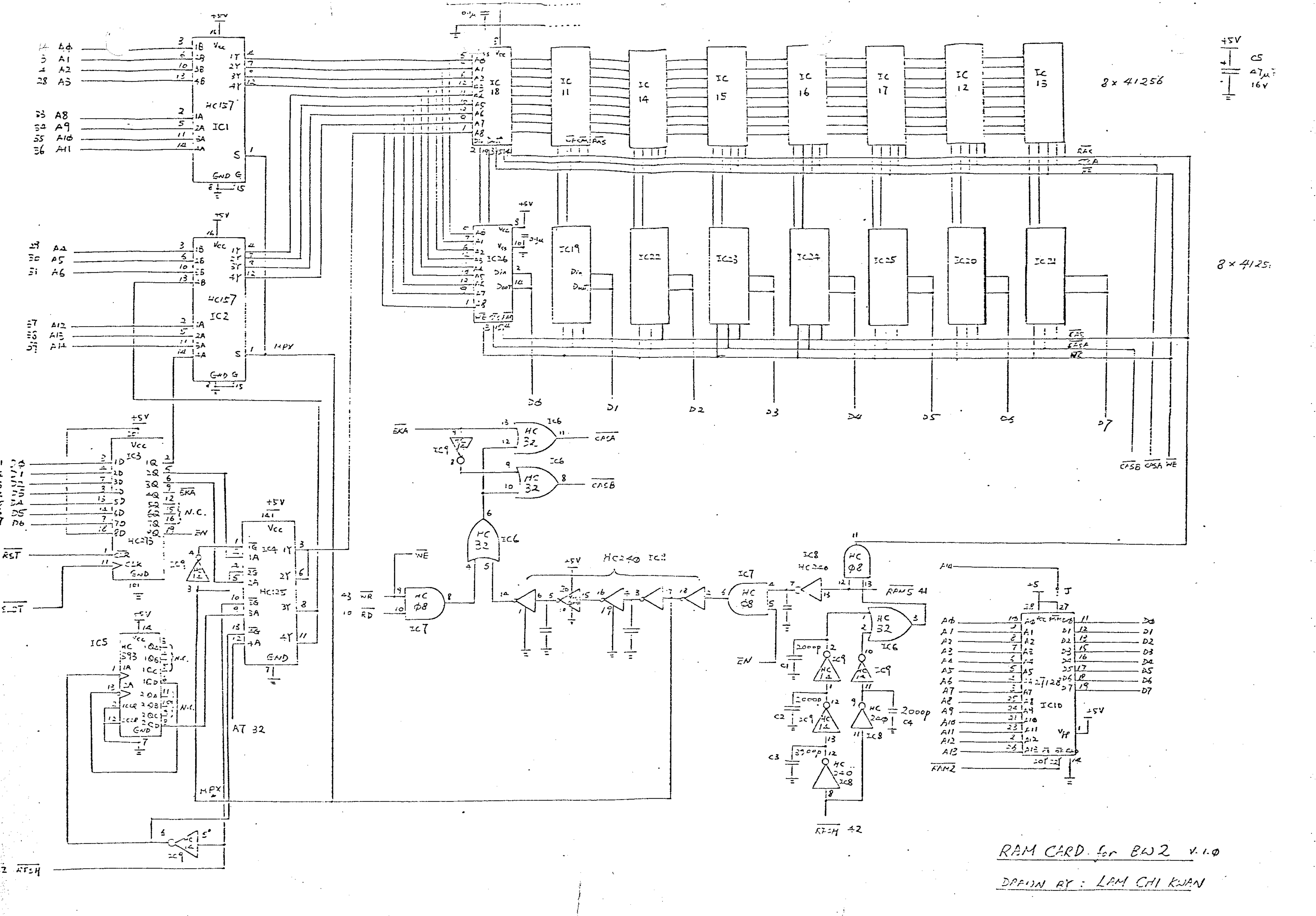
HS1 12 POSITION HOUSING DOUBLE ROW
HS3 18 POSITION HOUSING DOUBLE ROW
HS4 18 POSITION HOUSING DOUBLE ROW
HS5 20 POSITION HOUSING DOUBLE ROW
HS7 16 POSITION HOUSING DOUBLE ROW
HS8 34 POSITION FLAT CABLE CONNECTOR
HS10 34 POSITION FLAT CABLE CONNECTOR
HS9 4 POSITION HOUSING SINGLE ROW
HS11 TERMINEL RECEPTACLE
HS12 TERMINEL RECEPTACLE

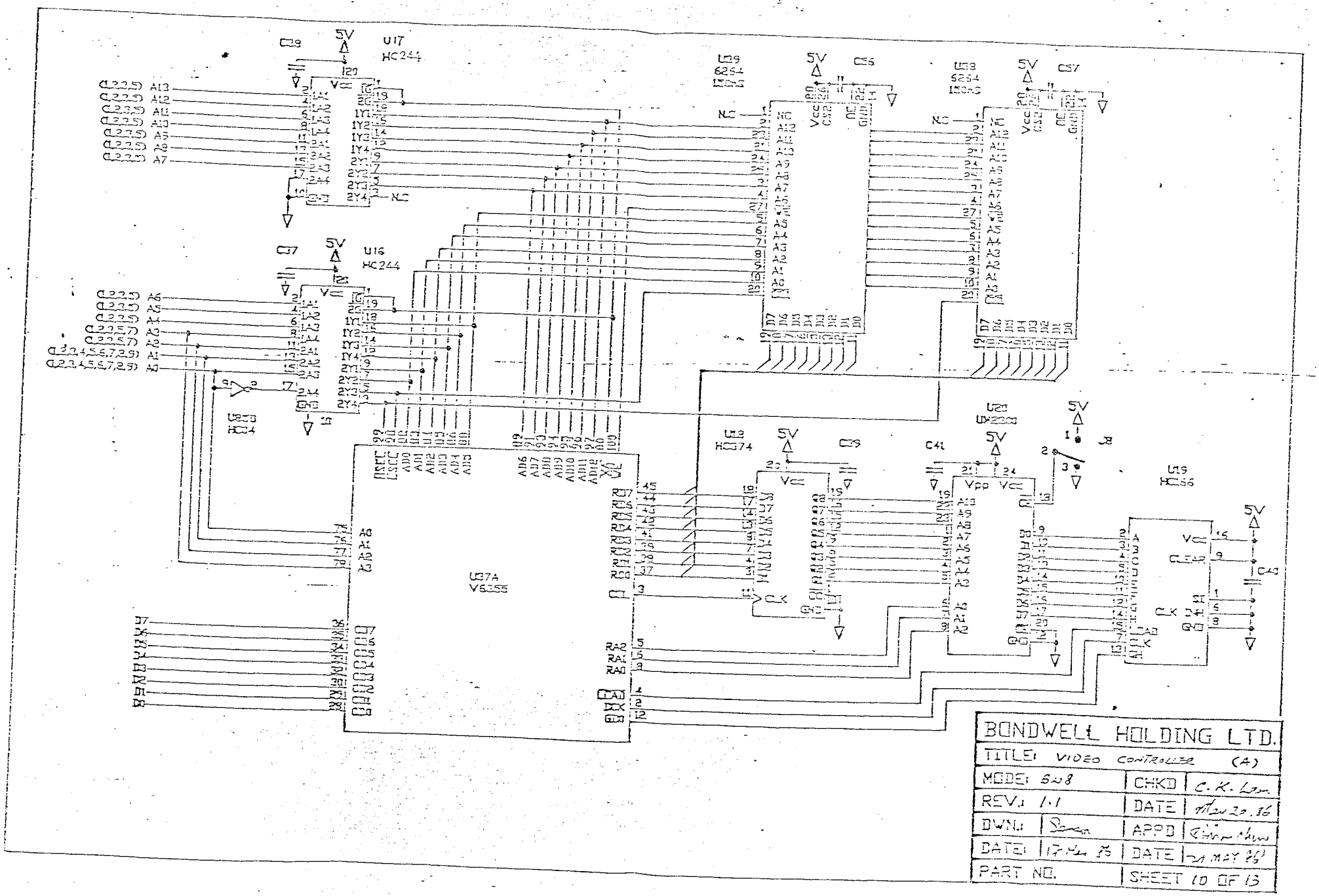
FIG. C3 Inter-Board Connector Table

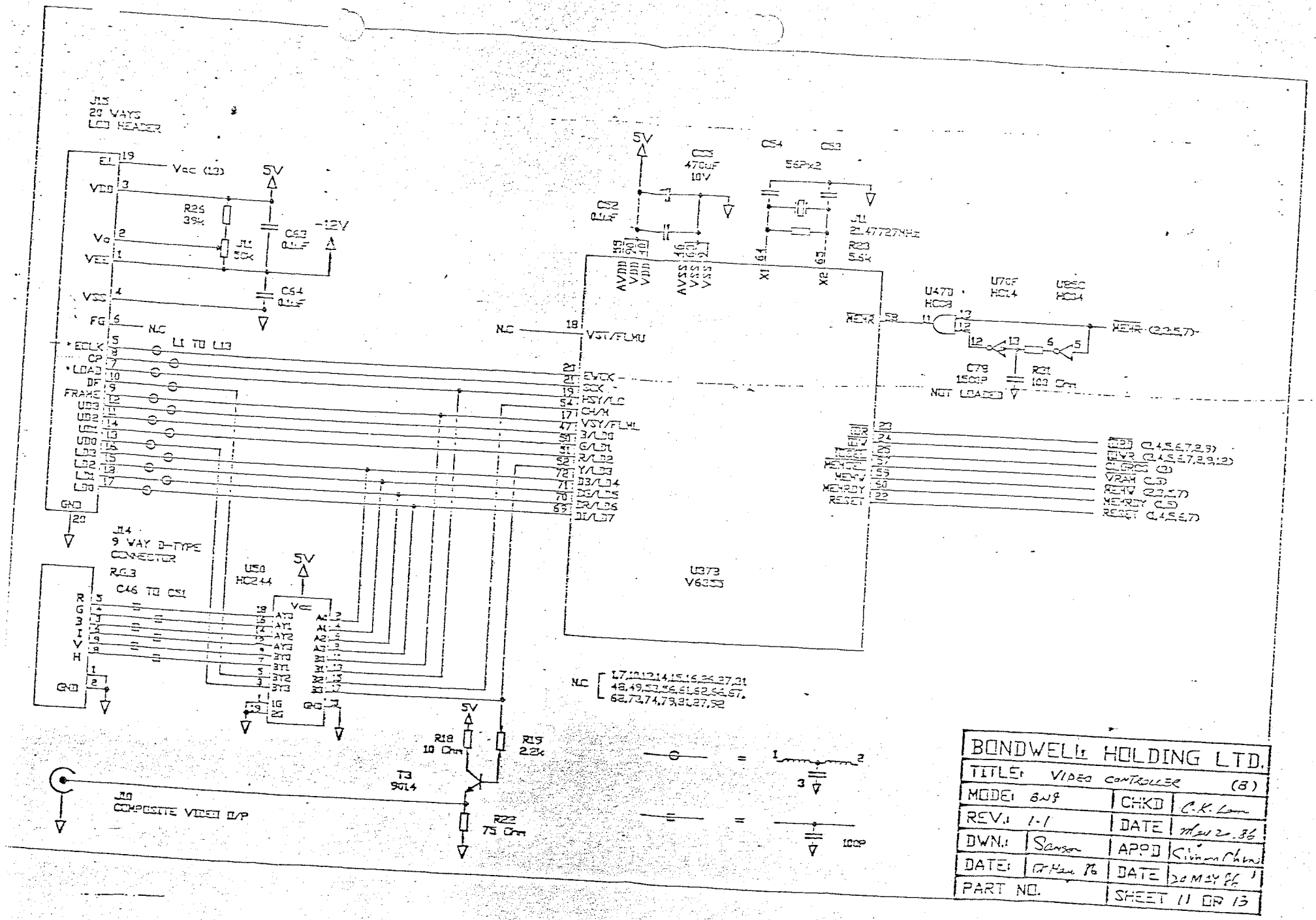
BONDWELL HOLDING LTD.

TITLE: INTER-BOARD CONNECTION TABLE		SCALE
PART NO: SD0035	3RD ANGLE PROJECTION	DWN: [illegible] DATE: 23 MAR 85
MODEL: BH02	TOLERANCE: ALL DIMENSIONS ARE IN MM; * FOR ALL DIMENSIONS UNLESS SPECIFIED OTHERWISE; * FOR ALL ANGULAR POSITIONS	CHKD: [illegible] DATE: March 23 1985
MATERIAL		APPD: Simon Stone DATE: 24 MAR 85
FINISH		
TREATMENT	SHEET 3 OF 3	ISSUE

RAM CARD for BW2 V.1.0


8 x 41256
8 x 4125
HC157
IC1
IC2
HC273
HC125
HC240
RAM CARD for BW2 V.1.0
DRAWN BY : LAM CHI KWAN


U17
HC244
U16
HC244
U7A
V6155
BONDWELL HOLDING LTD.
TITLE: VIDEO CONTROLLER (A)
MODE: 6x8
CHKD
C. K. Lam
REV: 1.1
DATE
DWN:
APPD
DATE:
17 Mar 86
DATE
24 MAY 86
PART NO.
SHEET 10 OF 13



BONDWELL HOLDING LTD.			
TITLE: VIDEO CONTROLLER (B)			
MODEL: B/S		CHKD	C.K. Lam
REV.: 1-1		DATE	May 26 86
DWN.:	Samson	APPD	Kinman Chan
DATE:	07 May 86	DATE	20 MAY 86
PART NO.		SHEET 11 OF 13	

A P P E N D I X A

Keyboard Matrix

Keyboard Matrix

	X0	X1	X2	X3	X4	X5	X6	X7
Y9	CAP LOCK							
Y8	SHIFT		SPACE BAR	Z	A	Q	2	F1
Y7	CTRL	= –	¦ \	X	S	W	3	F2
Y6	ⓐ	P	CURSOR UP	C	D	E	4	F3
Y5	1			V	F	R	5	F4
Y4	ESC			B	G	T	6	F5
Y3	TAB	* :	↵	N	H	Y	7	F6
Y2	CURSOR DOWN	· ~ ∧	{ [	M	J	U	8	F7
Y1	CURSOR RIGHT	+ ;	}]	< ,	K	I	9	F8
Y0	CURSOR LEFT	⇦	? /	> .	L	O	0	DEL

A P P E N D I X B

Bondwell 2 BIOS Listing

```
;*****************************************************
;CP/M version 2.2 GRAPHIC BIOS RELEASE 1.0 for
;project BH-026 Handbook
;edited by: LAM CHI KWAN          MARCH, 1985
;*****************************************************
;
CCP     EQU     BIOS-1600H      ;base of ccp
BDOS    EQU     CCP+806H        ;base of bdos
CDISK   EQU     0004H           ;current disk no. 0=a,.
IOBYTE  EQU     0003H           ;intel i/o byte
RESET   EQU     0000H           ;system reset address
;-----------------------------------------------------
;
PPICREG EQU     03H             ;82c55 control reg.
PPIA    EQU     00H             ;82c55 port a
PPIB    EQU     01H             ;82c55 port b
PPIC    EQU     02H             ;82c55 port c
;
CNTCREG EQU     13H             ;82c53 control reg.
CNT0    EQU     10H             ;82c53 counter reg. 0
CNT1    EQU     11H             ;82c53 counter reg. 1
CNT2    EQU     12H             ;82c53 counter reg. 2
;
UTDATA  EQU     40H             ;82c51 data reg.
UTCREG  EQU     41H             ;82c51 control reg.
UTSREG  EQU     41H             ;82c51 status reg.
;
PRINTER EQU     50H             ;printer output port
;
FDCCREG EQU     60H             ;FDC command reg.
FDCSREG EQU     60H             ;FDC status reg.
TR      EQU     61H             ;FDC track reg.
SR      EQU     62H             ;FDC sector reg.
DR      EQU     63H             ;FDC data reg.
;
;-------memory bank---------------------------
;
RAM1    EQU     0       ;ram bank 1
VRAM    EQU     1       ;vram bank
RAM2    EQU     2       ;ram bank 2
RAM3    EQU     3       ;ram bank 3
RAM4    EQU     4       ;ram bank 4
RAM5    EQU     5       ;ram bank 5
RAM6    EQU     6       ;ram bank 6
ROM     EQU     7       ;rom bank
BKMASK  EQU     0F8H    ;bank mask
BKIVMS  EQU     07H     ;bank inverse mask
;
;-------80 coloum x 25 line lcd addr.---------
;
CROM    EQU     800H    ;character rom
LCDIST  EQU     21H     ;lcd instruction reg.
LCDDAT  EQU     20H     ;lcd data reg.
CSRADL  EQU     07H     ;cursor low addr.
CSRADH  EQU     08H     ;cursor high addr.
```

```
VSTADL  EQU     05H     ;vram start addr. low
VSTADH  EQU     06H     ;vram start addr.high
CSRON   EQU     7DH     ;LCD MODE WORD CSR. ON
CSROFF  EQU     4DH     ;LCD MODE WORD CSR. OFF
;
;-------control characters--------------------
;
CR      EQU     0DH     ;carriage return
LF      EQU     0AH     ;line feed
BS      EQU     08H     ;back space
CURRHT  EQU     0CH     ;cursor right
CURUP   EQU     0BH     ;cursor up
CURHOM  EQU     1EH     ;home cursor
BKSPAC  EQU     08H     ;back space
ENDGRF  EQU.    00H     ;end graphic mode
ESC     EQU     1BH     ;escape
SPACE   EQU     20H     ;space
;
;-------esc characters-----------------------
CLEARS  EQU     '*'     ;clear screen & home
CLINE   EQU     'T'     ;clear to end of line
CLEND   EQU     'Y'     ;clear to end of screen
DLINE   EQU     'R'     ;delete line
ILINE   EQU     'E'     ;insert line
STRGRF  EQU     'G'     ;start graphic
INV     EQU     'I'     ;start inverse char.
ENINV   EQU     'N'     ;end inverse char.
;
;-----------------------------------------------
;cp/m to host disk constants
;
BLKSIZ  EQU     2048            ;cp/m block size
HSTSIZ  EQU     256             ;hst disk sec size
HSTSPT  EQU     18              ;phy. secs/trk
HSTBLK  EQU     HSTSIZ/128      ;hst buff/cp/m sec
CPMSPT  EQU     HSTBLK*HSTSPT   ;cp/m secs/trk
SECMSK  EQU     HSTBLK-1        ;sec mask
SECSHF  EQU     1       ;sec shift log2(hstblk)
;-----------------------------------------------
MTRON   EQU     90H     ;set drive motor on
MTROFF  EQU     19      ;oneshot 5s then motor off
MFDBK   EQU     5       ;motor feedback pc5
DRIVE0  EQU     10H     ;drive select 0 pa4
DRIVE1  EQU     20H     ;drive select 1 pa5
DISDRV  EQU     0CFH    ;disable both drive
SEEK    EQU     1CH     ;seek command with verify
RDSEC   EQU     88H     ;rd 1 sector command ,sso=0
WRSEC   EQU     0A8H    ;wr 1 sector command ,sso=0
RECAL   EQU     0CH     ;move rd/wr head to trk 0
RETRY   EQU     10      ;10 retry
NMI     EQU     0066H   ;nonmaskable addr.
ENBNMI  EQU     3       ;enable nmi, pc3
;
;-----------------------------------------------
;bdos constants on entry to write
```

```
;
WRALL   EQU     0       ;write to allocate
WRDIR   EQU     1       ;write to directory
WRUAL   EQU     2       ;write to unallocated
;
;---------------------------------------------
;
PBUSY   EQU     4       ;printer busy flag pc4
PSTB    EQU     7       ;strobe printer data out
;
;---------------------------------------------
;
RPKEY1  EQU     80H     ;repeat key time 1
RPKEY2  EQU     20H     ;repeat key time 2
MASKPA  EQU     11110000B   ;mask ppia
;---------------------------------------------
NSECTS  EQU     1600H/256       ;wboot sec count
;---------------------------------------------
;jump vector for individual subroutines
;
BIOS:
        JP      CBOOT           ;cold start
WBOOTE: JP      WBOOT           ;warm start
        JP      CONST           ;console status
        JP      CONIN           ;console char. in
        JP      CONOUT          ;console char. out
        JP      LIST            ;list char. out
        JP      PUNCH           ;punch char. out
        JP      READER          ;reader char. out
        JP      HOME            ;move head to home
        JP      SELDSK          ;select disk
        JP      SETTRK          ;set track number
        JP      SETSEC          ;set sector number
        JP      SETDMA          ;set dma address
        JP      READ            ;read disk
        JP      WRITE           ;write disk
        JP      LISTST          ;return list status
        JP      SECTRAN         ;sector translate
SWBANK: JP      SWBANK          ;switch bank ----DELETED---
;---------------------------------------------
FUNTBL:
F1:     DB      'DIR',20H,0,0,0,0,0,0,0,0,0,0,0,0
F2:     DB      'ERA',20H,0,0,0,0,0,0,0,0,0,0,0,0
F3:     DB      'TYPE',20H,0,0,0,0,0,0,0,0,0,0,0
F4:     DB      'REN',20H,0,0,0,0,0,0,0,0,0,0,0,0
F5:     DB      'SAVE',20H,0,0,0,0,0,0,0,0,0,0,0
F6:     DB      'STAT',20H,0,0,0,0,0,0,0,0,0,0,0
F7:     DB      'A:',0,0,0,0,0,0,0,0,0,0,0,0,0,0
F8:     DB      'B:',0,0,0,0,0,0,0,0,0,0,0,0,0,0
F9:     DB      'PIP',20H,0,0,0,0,0,0,0,0,0,0,0,0
F10:    DB      'ED',20H,0,0,0,0,0,0,0,0,0,0,0,0,0
F11:    DB      'ASM',20H,0,0,0,0,0,0,0,0,0,0,0,0
F12:    DB      'LOAD',20H,0,0,0,0,0,0,0,0,0,0,0
F13:    DB      'DDT',20H,0,0,0,0,0,0,0,0,0,0,0,0
F14:    DB      'DUMP',20H,0,0,0,0,0,0,0,0,0,0,0
```

```
F15:    DB      'SUBMIT',20H,0,0,0,0,0,0,0,0,0
F16:    DB      'SETUP',0,0,0,0,0,0,0,0,0,0,0
;
BAUDOD: DB      00000101B       ;default 300 baud
UARTOD: DB      01111010B       ;usart 1 stopbit
                ;DATA MASK      ;even parity,7bit/char.
                                ;16x
BITMSK: DB      01111111B       ;7 bit /char.
;-----------------------------------------------------
;fixed data tables (IBM compatible 3.5" disk)
;disk parameter header
DPBASE:
DPB0:   DW      0000H   ;no sec. translate
        DW      0000H   ;bdos use location
        DW      0000H   ;bdos use location
        DW      0000H   ;bdos use location
        DW      DIRBF   ;addr. of 128 byte dir buf
        DW      DPBLK   ;addr. of disk para block
        DW      CHK00   ;change disk chack addr.
        DW      ALL00   ;allocation vector
;
DPB1:   DW      0000H   ;no sec. translate
        DW      0000H   ;bdos use location
        DW      0000H   ;bdos use location
        DW      0000H   ;bdos use location
        DW      DIRBF   ;addr. of 128 byte dir buf
        DW      DPBLK   ;addr. of disk para block
        DW      CHK01   ;change disk chack addr.
        DW      ALL01   ;allocation vector
;
;disk parameter block, common to all disks
;
DPBLK:  DW      36              ;log. sec. per track
        DB      4               ;block shift (2k block)
        DB      15              ;block mask
        DB      1               ;extent mask
        DW      174             ;total block-1(less sys trk)
        DW      127             ;directory max
        DB      0C0H            ;alloc 0
        DB      0               ;alloc 1
        DW      32              ;check size
        DW      2               ;track offset
;
;-----------------------------------------------------------------
;
CBOOT:  LD      A,0FH   ;enable rom
        OUT     (PPIC),A
        JP      RESET           ;jump to reset address
CBOOTE: JP      BOOT
;
;
;-------------------------------------------------
;
WBOOT:  DI
        LD      SP,80H
```

```
        LD      A,(HSTWRT)
        OR      A
        CALL    NZ,WRHST
        XOR     A                       ;load ccp, bdos
        LD      (HSTDSK),A
        LD      (HSTTRK),A
        LD      (HSTSEC),A
        LD      (DR0TRK),A              ;home drv a
        LD      A,MTRON
        OUT     (CNTCREG),A
        IN      A,(PPIA)
        OR      10H
        OUT     (PPIA),A
        CALL    HOLD
        LD      A,0CH                   ;recal command
        OUT     (FDCCREG),A
        CALL    HOLD
;$$$$$$$$$$$$$$$$$$$$$$$
        JP      GOCPM                   ;FOR DEBUG ONLY
;$$$$$$$$$$$$$$$$$$$$$$$
;
NEXT:   LD      HL,CCP-128
        LD      (DMAADR),HL
        LD      B,NSECTS
LOAD:   PUSH    BC
        CALL    RDHST                   ;rd to hstbuf
        LD      A,(ERFLAG)
        OR      A
        JR      Z,NOLOER
        JR      WBOOT                   ;disk load err
NOLOER: LD      DE,(DMAADR)
        LD      HL,HSTBUF
        LD      BC,256
        LDIR
        LD      (DMAADR),DE
        LD      A,(HSTSEC)
        INC     A
        LD      (HSTSEC),A
        CP      18
        JR      NZ,NEXTS
        XOR     A
        LD      (HSTSEC),A
        INC     A
        LD      (HSTTRK),A
NEXTS:  POP     BC
        DJNZ    LOAD

        CALL    RDHST                   ;READ THE LAST RECORD
        LD      HL,HSTBUF
        LD      DE,(DMAADR)
        LD      BC,128
        LDIR
;
GOCPM:  LD      A,0C3H                  ;jmp instruction
        LD      (0000H),A               ;for jmp to wboot
```

```
        LD      HL,WBOOTE       ;wboot entry point
        LD      (0001H),HL
        LD      (0005H),A       ;for jmp to bdos
        LD      HL,BDOS         ;bdos entry point
        LD      (0006H),HL
        LD      BC,0080H        ;default dma address is 80h
        CALL    SETDMA
        LD      SP,0080H        ;boot rom cannot do this
        XOR     A
        LD      (HSTACT),A      ;host buffer inactive
        LD      (UNACNT),A      ;clear unalloc count
;----------DELETED IN GRAPHIC BIOS ----------
;       LD      HL,AUTO         ;set autorun
;       LD      DE,CCP+7
;       LD      BC,9
;       LDIR
;----------DELETED IN GRAPHIC BIOS ----------
        LD      A,(DRV1FL)
        OR      A
        JR      Z,AACTIV
        LD      A,(CDISK)       ;get current disk number
        CP      02
        JR      C,VDRIVE
AACTIV: XOR     A
VDRIVE: LD      C,A             ;send to the ccp
        XOR     A
        LD      (DRV1FL),A      ;recal drv b
;---------- CHANGED IN GRAPHIC BIOS ----------
        LD      HL,(CCPETY)
        LD      DE,CCP+3
        LD      (CCPETY),DE
        JP      (HL)
;-----------------------------------------------
;console status, return 0ffh if
;character ready, 00h if not
;
CONST:  LD      HL,KEYBUF       ;save old buffs.
        LD      DE,KEYBUF2
        LD      BC,4
        LDIR
;
        CALL    SCAN
        LD      A,(KEYBUF)
        CP      -1
        JR      NZ,KEYAVL
        LD      A,(KEYCNT)
        OR      A
        JR      NZ,KEYAVL
                                ;switch back
;       LD      HL,KEYBUF2      ;ctrs.
        LD      DE,KEYBUF
        LD      BC,4
        LDIR
;
        LD      A,(KEYBUF)
```

```
        CP      -1
        JR      NZ,KEYAVL
        LD      A,(KEYCNT)
        OR      A
        RET     Z
KEYAVL: LD      A,-1
        RET
;
;-----------------------------------------------
;console input. return keycode in a
;
CONIN:  LD      A,(KEYBUF)
        CP      -1
        JR      Z,NXTCHK
        LD      HL,KEYBUF
        LD      (HL),-1
        RET
NXTCHK: LD      A,(KEYCNT)
        OR      A
        JR      Z,RDKBD
        DEC     A
        LD      (KEYCNT),A
        LD      HL,(KEYPTR)
        LD      B,(HL)
        INC     HL
        LD      (KEYPTR),HL
        LD      A,(HL)
        OR      A
        JR      NZ,RETKEY
        LD      (KEYCNT),A
RETKEY: LD      A,B
        RET
RDKBD:  LD      HL,00E0H        ;delay 3msec
        CALL    HOLDE
        CALL    SCAN
        JR      CONIN
;
;-----------------------------------------------
;scan keyboard and return keycode in acc,
;keybuf, keyptr, keycnt
;
SCAN:   LD      HL,NEWTBL+1     ;read 9 rows
        LD      C,8
        IN      A,(PPIA)
        AND     MASKPA
        LD      B,A
        LD      E,-1
SCANKB: LD      A,B
        OR      C
        OUT     (PPIA),A
        IN      A,(PPIB)
        LD      (HL),A
        AND     E
        LD      E,A
        INC     HL
```

```
        DEC     C
        JP      P,SCANKB
        INC     E
        JP      Z,VALID
;
        XOR     A
        LD      (CNTRL),A
        LD      (SHIFT),A
        LD      (CAPS),A
;
        LD      A,B             ;caps?
        OR      9
        OUT     (PPIA),A
        IN      A,(PPIB)
        RRA
        JR      C,CKCNTL
        LD      (CAPS),A
CKCNTL: LD      A,B
        OR      7               ;check cntrl
        OUT     (PPIA),A
        IN      A,(PPIB)
        RRA
        JR      C,CKSHFT
        LD      (CNTRL),A
CKSHFT: LD      A,B
        OR      8               ;check shift
        OUT     (PPIA),A
        IN      A,(PPIB)
        RRA
        JR      C,CHKEND
        LD      (SHIFT),A
        LD      A,(CNTRL)
        OR      A
        JP      NZ,INVALI
;
CHKEND: LD      HL,NEWTBL
        LD      (HL),-1
        INC     HL
        SET     0,(HL)
        INC     HL
        SET     0,(HL)
;
        LD      B,10            ;compare 10 rows
        LD      DE,NEWTBL
        LD      HL,OLDTBL
CMPRE:  LD      A,(DE)
        LD      C,(HL)
        XOR     C
        AND     C
        JR      NZ,DEBOUN
        INC     DE
        INC     HL
        DJNZ    CMPRE
;
        LD      (FLAG1),A       ;process old key
```

```
        LD      (FLAG2),A
        LD      B,10            ;compare 10 rows
        LD      DE,NEWTBL
        LD      HL,OLDTBL
COMPRE: LD      A,(DE)
        LD      C,(HL)
        OR      C
        CPL
        OR      A
        JR      NZ,MEMKEY
CONTIN: INC     DE
        INC     HL
        DJNZ    COMPRE          ;process old key
        LD      A,(FLAG2)
        OR      A
        JR      Z,VALID
        LD      A,(TEMP)
        LD      C,A
        LD      A,(NPRESS)
        CP      C
        JR      NZ,VALID
        LD      HL,TIMER
        INC     (HL)
        LD      A,(FLAG3)
        OR      A
        JR      NZ,TIME2
        LD      A,(TIMER)
        CP      RPKEY1
        JR      C,VALID
        LD      (FLAG3),A
        JR      RESTMR
TIME2:  LD      A,(TIMER)
        CP      RPKEY2
        JR      C,VALID
        JR      RESTMR
;
MEMKEY: LD      C,A
        LD      A,(FLAG2)
        OR      A
        JR      NZ,VALID
        INC     A
        LD      (FLAG2),A
        PUSH    BC
        PUSH    DE
        PUSH    HL
        LD      A,C
        CALL    FNDKEY
        LD      (TEMP),A
        POP     HL
        POP     DE
        POP     BC
;&&&&&& release 1.1 &&&&&&&&
        LD      A,(FLAG4)       ;see if bad key
        OR      A
        JR      NZ,INVALI
```

```
;&&&&&& release 1.1 &&&&&&&&
        JR      CONTIN
VALID:  CALL    UPDATE
INVALI: XOR     A
        LD      (KEYCNT),A
        DEC     A
        LD      (KEYBUF),A
        RET
;
DEBOUN: CALL    FNDKEY          ;find key-code
        LD      C,A
;&&&&&&& release 1.1 &&&&&&&
        LD      A,(FLAG4)       ;see if bad key
        OR      A
        JR      NZ,INVALI
;&&&&&& release 1.1 &&&&&&&&
        LD      A,(FLAG1)
        OR      A
        JR      NZ,SCNDRD
NSCN:   LD      A,-1
        LD      (FLAG1),A
        LD      A,C
        LD      (NPRESS),A
        LD      HL,180H         ;delay 5msec
        CALL    HOLDE
        JP      SCAN
;
SCNDRD: LD      A,(NPRESS)      ;npress=newkey?
        CP      C
        JR      NZ,NSCN
        XOR     A
        LD      (FLAG1),A
        LD      (FLAG3),A
RESTMR: XOR     A
        LD      (TIMER),A
;
        LD      A,(NPRESS)
        CP      128
        JR      NC,FUNKEY
        XOR     A
        LD      (KEYCNT),A
        LD      A,(NPRESS)
        LD      (KEYBUF),A
        LD      HL,-1
        LD      (KEYPTR),HL
EXCHG:  CALL    UPDATE
FINISH: LD      A,(KEYBUF)
        RET
FUNKEY: SUB     128
        ADD     A,A
        ADD     A,A
        ADD     A,A
        ADD     A,A
        LD      E,A
        LD      D,0
```

```
        LD      HL,FUNTBL
        ADD     HL,DE
        LD      (KEYPTR),HL
        LD      A,16
        LD      (KEYCNT),A
        LD      A,-1
        LD      (KEYBUF),A
        JR      EXCHG
;
;-----------------------------------------------
;find keycode subroutine. pass a, b
;output--keycode in a
;
FNDKEY: LD      C,A
;
;&&&&&& release 1.1 &&&&&&&&&
        XOR     A
        LD      (FLAG4),A       ;no bad key yet
;&&&&&& release 1.1 &&&&&&&&&
;
        DEC     B
        LD      A,B
        ADD     A,A
        ADD     A,A
        ADD     A,A
        LD      E,A
        LD      D,0
        LD      A,(SHIFT)       ;true shift?
        OR      A
        JR      NZ,ASCII2
        LD      HL,ASCII-1
        ADD     HL,DE
FNDPOS: RLC     C
        INC     HL
        JR      NC,FNDPOS
        LD      A,(CNTRL)
        OR      A
        JR      Z,NOCNTL
        LD      A,(HL)
        CP      60H
        JR      C,BADKEY
        CP      128
        JR      NC,BADKEY
        AND     1FH
        RET
NOCNTL: LD      A,(CAPS)
        OR      A
        JR      Z,GOOUT
        LD      A,(HL)
        CP      61H
        RET     C
        CP      7BH
        RET     NC
        AND     5FH
        RET
```

```
;
ASCII2: LD      HL,SHFTBL-1
        ADD     HL,DE
RDPOS:  RLC     C
        INC     HL
        JR      NC,RDPOS
GOOUT:  LD      A,(HL)
        RET
;
;&&&&&&& release 1.1 &&&&&&&&
BADKEY: LD      A,-1
        LD      (FLAG4),A
        RET
;&&&&&&& release 1.1 &&&&&&&&
;
;-----------------------------------------------
;
ASCII:  DB      7FH,30H,6FH,6CH,2EH,2FH,08H,13H
        DB      135,39H,69H,6BH,2CH,5DH,3BH,04H
        DB      134,38H,75H,6AH,6DH,5BH,5EH,18H
        DB      133,37H,79H,68H,6EH,0DH,3AH,09H
        DB      132,36H,74H,67H,62H,-1,-1,1BH
        DB      131,35H,72H,66H,76H,-1,-1,31H
        DB      130,34H,65H,64H,63H,05H,70H,40H
        DB      129,33H,77H,73H,78H,5CH,2DH,-1
        DB      128,32H,71H,61H,7AH,20H,-1,-1
        DB      -1,-1,-1,-1,-1,-1,-1,-1
;
SHFTBL: DB      7FH,5FH,4FH,4CH,3EH,3FH,08H,13H
        DB      143,29H,49H,4BH,3CH,7DH,2BH,04H
        DB      142,28H,55H,4AH,4DH,7BH,7EH,18H
        DB      141,27H,59H,48H,4EH,0DH,2AH,09H
        DB      140,26H,54H,47H,42H,-1,-1,1BH
        DB      139,25H,52H,46H,56H,-1,-1,21H
        DB      138,24H,45H,44H,43H,05H,50H,60H
        DB      137,23H,57H,53H,58H,7CH,3DH,-1
        DB      136,22H,51H,41H,5AH,20H,-1,-1
        DB      -1,-1,-1,-1,-1,-1,-1,-1
;
;-----------------------------------------------
;subroutine to copy newtbl to oldtbl
;
UPDATE: LD      HL,NEWTBL
        LD      DE,OLDTBL
        LD      BC,10
        LDIR
        RET
;-----------------------------------------------
;GRAPHIC CONSOLE OUT FOR BH-026 HANDBOOK
;CREATED BY C.K.LAM        DATE: FEB 28,1985
;-----------------------------------------------
;FEATURES ADDED:---
;
;ESC 0  GRAPHIC MODE, CLEAR SCREEN (16K VRAM), VSTART=0, HOME TEXT CU
```

```
;       HOME GRAPHIC CURSOR.  (BOTH CURSOR DEFAULT ON)
;ESC 1  ON TEXT CURSOR
;ESC 2  OFF TEXT CURSOR
;ESC 3  ON GRAPHIC CURSOR
;ESC 4  OFF GRAPHIC CURSOR
;ESC 5 Y X  GRAPHIC CURSOR POSITION
;ESC 6  DOT ON, UPDATE GRAPHIC CURSOR
;ESC 7  DOT OFF, UPDATE GRAPHIC CURSOR
;
;-----------------------------------------------
;CONSOLE OUT, PASS (C)
;
CONOUT: LD      IY,0
        ADD     IY,SP
        LD      SP,STACK2
        IN      A,(PPIC)
        LD      (BANKBF),A
        LD      A,(ESC5)
        OR      A
        JP      NZ,G05A
        LD      A,(ESCFLG)      ;esc active?
        OR      A
        JR      NZ,ESCACT
        LD      A,(GRFFLG)      ;graphic mode?
        OR      A
        JR      NZ,GRAPH
        LD      A,C
        LD      HL,CTLTBL
COMPAR: LD      B,(HL)
CNEXT:  INC     HL
        CP      (HL)
        INC     HL
        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        JR      Z,LDJMP
        DJNZ    CNEXT
LDJMP:  EX      DE,HL
        JP      (HL)            ;goto diff. routine
;
ESCACT: LD      A,(ESCFLG)
        DEC     A
        LD      (ESCFLG),A
        CP      2
        JR      NZ,NOT2
        LD      A,C
        CP      '='
        JP      Z,RTPATH
        XOR     A
        LD      (ESCFLG),A
        LD      A,C
        LD      HL,ESCTBL
        JR      COMPAR
;
NOT2:   OR      A
```

```
        JR      Z,CHXY
        LD      A,C
        LD      (CSNY),A
        JP      GOPATH
CHXY:   LD      A,C
        SUB     32
        CP      80
        JR      NC,CRYRTY
        LD      (CUSRX),A
        LD      A,(CSNY)
        SUB     32
        CP      25
CRYRTY: JP      NC,RTPATH
        LD      (CUSRY),A
        CALL    INCSR
        JP      UDCUSR
;
GRAPH:  LD      A,C
        CP      ENDGRF
        JP      NZ,K08          ;display char.
        XOR     A
        LD      (GRFFLG),A
GOPATH: JP      RTPATH
;
CTLTBL: DB      12              ;count
        DB      CR              ;carriage ret.
        DW      K01
        DB      LF              ;line feed
        DW      K02
        DB      CURRHT          ;cursor right
        DW      K03
        DB      CURUP           ;cursor up
        DW      K04
        DB      CURHOM          ;home cursor
        DW      K05
        DB      ESC             ;escape
        DW      K06
        DB      BKSPAC          ;back space
        DW      K07
        DB      1AH             ;form feed for kaypro
        DW      E01
        DB      18H             ;clear to eol for kayp.
        DW      E02
        DB      17H             ;clear to eos for kayp.
        DW      E03
        DB      07H             ;bell
        DW      RTPATH
        DB      SPACE           ;space or char.
        DW      K08
;
ESCTBL: DB      17              ;count
        DB      CLEARS          ;clear screen & home
        DW      E01
        DB      CLINE           ;clear to end of line
        DW      E02
```

```
        DB      CLEND           ;clear to end of screen
        DW      E03
        DB      DLINE           ;delete line
        DW      E04
        DB      ILINE           ;insert line
        DW      E05
        DB      STRGRF          ;start graphic
        DW      E06
        DB      INV             ;start inverse char.
        DW      E07
        DB      ENINV           ;end inverse char.
        DW      E08
        DB      0CH             ;formfeed
        DW      E01
        DB      '1'             ;TEXT CURSOR ON
        DW      G01
        DB      '2'             ;OFF TEXT CURSOR
        DW      G02
        DB      '3'             ;GRAPHIC CURSOR ON
        DW      G03
        DB      '4'             ;OFF GRAPHIC CURSOR
        DW      G04
        DB      '5'             ;GRAPHIC CURSOR POSITION
        DW      G05
        DB      '6'             ;DOT ON
        DW      G06
        DB      '7'             ;DOT OFF
        DW      G07
        DB      SPACE           ;no match
        DW      E09             ;then return
;
;-----------------------------------------------
;
K01:    CALL    INCSR
        XOR     A               ;carriage ret
        LD      (CUSRX),A
        JP      UDCUSR
;
K02:    CALL    INCSR
K02A:   LD      A,(CUSRY)       ;line feed
        CP      24
        JR      Z,BOTTOM
        INC     A
        LD      (CUSRY),A
        JP      UDCUSR
BOTTOM: LD      DE,(VSTART)
        LD      HL,640
        ADD     HL,DE
        EX      DE,HL
        LD      BC,25*640
        ADD     HL,BC
        LD      BC,640
        CALL    CLLINE
        LD      A,D
        AND     3FH
```

```
        LD      D,A
        LD      C,LCDDAT
        LD      A,VSTADL        ;start addr.
        OUT     (LCDIST),A
        OUT     (C),E
        LD      A,VSTADH
        OUT     (LCDIST),A
        OUT     (C),D
        LD      (VSTART),DE
        JP      UDCUSR
;
K03:    CALL    INCSR
K03A:   LD      A,(CUSRX)       ;cursor right
        CP      79
        JR      NZ,MVRHT
        XOR     A
        LD      (CUSRX),A
        JR      K02A
MVRHT:  INC     A
        LD      (CUSRX),A
        JP      UDCUSR
;
K04:    LD      A,(CUSRY)       ;cusor up
        OR      A
        JP      Z,RTPATH
        DEC     A
        LD      (CUSRY),A
        CALL    INCSR
        JP      UDCUSR
;
K05:    CALL    INCSR
K05A:   XOR     A               ;home cursor
        LD      (CUSRY),A
        LD      (CUSRX),A
        JP      UDCUSR
;
K06:    LD      A,3             ;escape
        LD      (ESCFLG),A
        JP      RTPATH
;
K07:    LD      A,(CUSRX)       ;cusor back
        DEC     A
        JP      P,BSNEXT
        LD      A,(CUSRY)
        OR      A
        JP      Z,RTPATH
        DEC     A
        LD      (CUSRY),A
        LD      A,79
BSNEXT: LD      (CUSRX),A
        CALL    INCSR
        JP      UDCUSR
;
K08:    LD      A,(GRFFLG)      ;display char.
        OR      A
```

```
        LD      A,C
        LD      (WORD),A
        JR      NZ,GRF
        RES     7,A             ;mask off bit 7
GRF:    LD      L,A
        LD      H,0
        ADD     HL,HL
        ADD     HL,HL
        ADD     HL,HL
        LD      DE,CROM
        ADD     HL,DE
        LD      DE,WORDBF
        LD      BC,8
        LD      A,(BANKBF)      ;rom bank
        AND     0F8H
        OR      ROM
        OUT     (PPIC),A
        LDIR
;
        LD      A,(GRFFLG)      ;inverse video?
        OR      A
        JR      NZ,CKIFLG
        LD      A,(WORD)
        OR      A
        JP      M,INVMOD
;
CKIFLG: LD      A,(INVFLG)
        OR      A
        JR      Z,WRVRAM
INVMOD: CALL    SETINV
;
WRVRAM: CALL    UPSCRN
        JP      K03A
;
E01A:   CALL    CHVRAM
        LD      HL,3FFFH        ;clear screen
CL:     XOR     A               ;home cusor
        LD      (HL),A
        DEC     HL
        LD      A,H
        OR      L
        JR      NZ,CL
        LD      (VSTART),HL
        XOR     A
        LD      C,LCDIST
        LD      B,VSTADL
        OUT     (C),B
        OUT     (LCDDAT),A
        INC     B
        OUT     (C),B
        OUT     (LCDDAT),A
        RET
;
E02:    CALL    CTEOL           ;clear to end
        JR      RRT             ;of line
```

```
;
CTEOL:  LD      A,(CUSRX)       ;clear to end
        NEG                     ;of line
        ADD     A,80
        JP      Z,RTPATH
        LD      C,A
        LD      B,0
        LD      HL,(CUSRPO)
        LD      A,8
CLMORE: PUSH    AF
        PUSH    BC
        PUSH    HL
        CALL    CLLINE
        POP     HL
        POP     BC
        POP     AF
        LD      DE,80
        ADD     HL,DE
        DEC     A
        JR      NZ,CLMORE
        RET
;
E03:    CALL    CTEOL           ;clear to end
        LD      A,(CUSRY)       ;of screen
        ADD     A,0E8H          ;24-cusry
        NEG
        JP      Z,RTPATH
        CALL    COMPUT
        LD      B,H
        LD      C,L
        LD      DE,(VERTIC)
        LD      HL,640
        ADD     HL,DE
        CALL    CLLINE
RRT:    JP      UDCUSR          ;
;
E04:    XOR     A               ;delete line
        LD      (CUSRX),A
TRANSF: LD      A,(CUSRY)
        ADD     A,0E8H
        NEG
        JR      Z,DELAST
        CALL    COMPUT
        LD      B,H
        LD      C,L
        LD      DE,(VERTIC)
        LD      HL,640
        ADD     HL,DE
        CALL    CHVRAM
LOA:    PUSH    BC
        LD      A,(HL)          ;rd vram
        NOP                     ;screen noise
        NOP
        LD      (DE),A
        INC     HL
```

```
        INC     DE
        POP     BC
        DEC     BC
        LD      A,B
        OR      C
        JR      NZ,LOA
        EX      DE,HL
        JR      LASTLN
DELAST: LD      HL,(VERTIC)
        JR      LASTLN
;
E05:    XOR     A               ;insert line
        LD      (CUSRX),A
        LD      A,(CUSRY)
        ADD     A,0E8H
        NEG
        JR      Z,DELAST
        CALL    COMPUT
        LD      B,H
        LD      C,L
        LD      HL,25*640-1
        LD      DE,(VSTART)
        ADD     HL,DE
        PUSH    HL
        LD      HL,24*640-1
        ADD     HL,DE
        POP     DE
        CALL    CHVRAM
MVDATA: PUSH    BC
        LD      A,(HL)
        NOP                     ;screen noise
        NOP
        LD      (DE),A
        DEC     HL
        DEC     DE
        POP     BC
        DEC     BC
        LD      A,B
        OR      C
        JR      NZ,MVDATA
LASTLN: LD      BC,640
        CALL    CLLINE
        JR      UDCUSR
;
E06:    LD      A,01            ;start graphic
        LD      (GRFFLG),A
        JR      RTPATH
;
E07:    LD      A,01            ;set inverse
        LD      (INVFLG),A
        JR      RTPATH
;
E08:    XOR     A               ;end inverse
        LD      (INVFLG),A
E09:    JR      RTPATH
```

```
;
;
;-----------------------------------------------
;clear vram, pass hl, bc
;
CLLINE: CALL    CHVRAM  ;change to vram
CLEAR:  LD      A,H
        AND     00111111B
        LD      H,A
        XOR     A
        LD      (HL),A
        INC     HL
        DEC     BC
        LD      A,B
        OR      C
        JR      NZ,CLEAR
        RET
;-----------------------------------------------
;calculate cusor addr. and o/p to lcd
;pass cusry, cusrx, vstart
;
UDCUSR: LD      A,(CUSRY)       ;(640xcusry)+560
        CALL    COMPUT          ;+cusrx
        LD      DE,(VSTART)
        ADD     HL,DE
        LD      (VERTIC),HL
        LD      DE,(CUSRX)
        ADD     HL,DE
        LD      (CUSRPO),HL
        LD      DE,560
        ADD     HL,DE
        LD      C,LCDDAT        ;o/p csr addr.
        LD      A,CSRADL
        OUT     (LCDIST),A
        OUT     (C),L
        LD      A,CSRADH
        OUT     (LCDIST),A
        OUT     (C),H
        CALL    INCSR
RTPATH: LD      A,(BANKBF)
        OUT     (PPIC),A
        LD      SP,IY
        RET
;-----------------------------------------------
;inverse char. pat. in cusrpo
;
INCSR:  NOP
        CALL    CHVRAM
        LD      HL,(CUSRPO)
        LD      IX,WORDBF
        LD      DE,80
        LD      B,8
RDPAT:  LD      A,(HL)
        LD      (IX),A
        ADD     HL,DE
```

```
        INC     IX
        DJNZ    RDPAT
        CALL    SETINV
        CALL    UPSCRN
        RET
;-----------------------------------------------
;subroutine to set char. pattern in wordbf to inverse mode
;
SETINV: LD      B,8             ;set inverse
        LD      HL,WORDBF
SETI:   LD      A,(HL)
        CPL
        LD      (HL),A
        INC     HL
        DJNZ    SETI
        RET
;-----------------------------------------------
;subroutine to update pattern on screen from wordbuf
;
UPSCRN: LD      HL,(CUSRPO)
        LD      IX,WORDBF
        LD      B,8
        LD      DE,80
        CALL    CHVRAM
WRPATN: LD      A,H
        AND     00111111B
        LD      H,A
        LD      A,(IX)
        LD      (HL),A
        INC     IX
        ADD     HL,DE
        DJNZ    WRPATN
        RET
;-----------------------------------------------
;subroutine to change to vram bank
;
CHVRAM: LD      A,(BANKBF)      ;change to vram
        AND     0F8H            ;bank
        OR      VRAM
        OUT     (PPIC),A
        RET
;-----------------------------------------------
;subroutine to multiply acc. by 640
;
COMPUT: LD      D,A
        LD      E,0
        LD      H,D
        LD      L,E
        SLA     D
        SRL     H
        RR      L
        ADD     HL,DE
        RET
;-----------------------------------------------
E01:    CALL    E01A    ;ACTIVATE GRAPHIC MODE
```

```
        XOR     A
        LD      (DOTPOSY),A
        LD      (DOTPOSX),A
        LD      (DOTPOSX+1),A
        LD      (DOTBYTE),A
        LD      (DOTBYTE+1),A
        LD      A,0C9H          ;OFF GRF CUSR
        LD      (GRFCSR),A
        LD      A,80H
        LD      (DOTBIT),A
        JP      K05A
;
G01:    LD      A,(INCSR)       ;ON TEXT CURSOR
        OR      A
        JR      Z,ENG01
        XOR     A
        LD      (INCSR),A
        CALL    INCSR
        LD      C,21H
        XOR     A
        OUT     (C),A
        DEC     C
        LD      A,CSRON
        OUT     (C),A
ENG01:  JP      RTPATH
;
G02:    LD      A,(INCSR)       ;OFF TEXT CURSOR
        OR      A
        JR      NZ,ENG01
        CALL    INCSR
        LD      A,0C9H
        LD      (INCSR),A       ;"RET" INST.
        LD      C,21H
        XOR     A
        OUT     (C),A
        DEC     C
        LD      A,CSROFF
        OUT     (C),A
        JR      ENG01
;
G03:    LD      A,(GRFCSR)      ;ON GRF CUSR
        OR      A
        JR      Z,ENG01
        XOR     A
        LD      (GRFCSR),A
        CALL    GRFCSR
        JR      ENG01
;
G04:    LD      A,(GRFCSR)      ;OFF GRF CUSR
        OR      A
        JR      NZ,ENG01
        CALL    GRFCSR
        LD      A,0C9H
        LD      (GRFCSR),A
        JR      ENG01
```

```
;
G05:    LD      A,3             ;MOVE GRF CUSR
        LD      (ESC5),A
        CALL    GRFCSR
        JR      ENG01
G05A:   DEC     A
        LD      (ESC5),A
        CP      2
        JR      NZ,XVALH
        LD      A,C
        CP      200             ;Y RANGE
        JR      C,YRANGE
        LD      A,200
YRANGE: LD      (DOTPOSY),A
        JR      ENG01
XVALH:  OR      A
        JR      Z,XVALL
        LD      A,C
        LD      (DOTPOSX+1),A
        JR      ENG01
XVALL:  LD      A,C
        LD      (DOTPOSX),A
        LD      A,(DOTPOSX+1)
        CP      2H
        JR      C,XRANGE
        JR      NZ,OUTRAN
        LD      A,C
        CP      80H             ;X RANGE
        JR      C,XRANGE
OUTRAN: LD      BC,27FH
        LD      (DOTPOSX),BC
XRANGE: LD      A,(DOTPOSX)
        AND     07H
        LD      B,A
        INC     B
        XOR     A
        SCF
SETDOT: RR      A
        DJNZ    SETDOT          ;DOTBIT IN ACC.
        LD      (DOTBIT),A
        LD      DE,(DOTPOSX)
        LD      B,3
DIVID8: SRL     D
        RR      E
        DJNZ    DIVID8
        PUSH    DE
        LD      A,(DOTPOSY)
        LD      L,A
        LD      H,0
        ADD     HL,HL
        ADD     HL,HL
        ADD     HL,HL
        ADD     HL,HL   ;*16
        PUSH    HL
        ADD     HL,HL
```

```
        ADD     HL,HL   ;*64
        POP     DE
        ADD     HL,DE   ;*80
        POP     DE
        ADD     HL,DE   ;DOTBYTE
        LD      (DOTBYTE),HL
        CALL    GRFCSR
ENG05:  JP      RTPATH
;
G06:    CALL    CHVRAM
        LD      HL,(DOTBYTE)    ;DOT ON
        LD      A,(DOTBIT)
        OR      (HL)
        LD      (HL),A
        JR      ENG05
;
G07:    CALL    CHVRAM          ;DOT OFF
        LD      A,(DOTBIT)
        CPL
        LD      HL,(DOTBYTE)
        AND     (HL)
        LD      (HL),A
        JR      ENG05
;
;-----------------------------------------------
;TOGGLE COLOR OF GRF. CUSOR AT DOT POS Y,X
;
GRFCSR: RET
        CALL    CHVRAM
        LD      HL,(DOTBYTE)
        LD      DE,80
        XOR     A
        SBC     HL,DE
        LD      B,6
        LD      A,(DOTPOSY)     ;TOGGLE COLOR OF
        DEC     A               ;GRF. CUSOR AT
        CP      -1              ;DOT POSY,X
        JR      Z,LOHALF
UPBIT:  DEC     A
        CP      -1
        JR      Z,LOHALF
        PUSH    AF
        LD      DE,80
        XOR     A
        SBC     HL,DE
        LD      C,(HL)
        LD      A,(DOTBIT)
        XOR     C
        LD      C,A
        LD      (HL),C
        POP     AF
        DJNZ    UPBIT
;
LOHALF: LD      HL,(DOTBYTE)
        LD      DE,80
```

```
        ADD     HL,DE
        LD      B,6
        LD      A,(DOTPOSY)
        INC     A
        CP      200
        JR      Z,LFHALF
LOBIT:  INC     A
        CP      200
        JR      Z,LFHALF
        PUSH    AF
        LD      DE,80
        ADD     HL,DE
        LD      C,(HL)
        LD      A,(DOTBIT)
        XOR     C
        LD      C,A
        LD      (HL),C
        POP     AF
        DJNZ    LOBIT
;
LFHALF: LD      HL,(DOTBYTE)    ;LEFT HALF
        DEC     HL
        LD      C,(HL)
        INC     HL
        LD      D,(HL)
        LD      A,(DOTBIT)
        LD      B,0
SHTRT:  SRL     C
        RR      D
        RR      E
        INC     B
        SRL     A
        JR      NC,SHTRT
        LD      A,0FEH
        XOR     D
        LD      D,A
SHTLF:  SLA     E
        RL      D
        RL      C
        DJNZ    SHTLF
        LD      (HL),D
        PUSH    HL
        LD      HL,(DOTPOSX)
        LD      A,H
        OR      L
        POP     HL
        JR      Z,RTHALF
        DEC     HL
        LD      (HL),C
;
RTHALF: LD      HL,(DOTBYTE)    ;RIGHT HALF
        INC     HL
        LD      E,(HL)
        DEC     HL
        LD      D,(HL)
```

```
        LD      A,(DOTBIT)
        LD      B,0
SHTLFT: SLA     E
        RL      D
        RL      C
        INC     B
        SLA     A
        JR      NC,SHTLFT
        LD      A,7FH
        XOR     D
        LD      D,A
SHTRHT: SRL     C
        RR      D
        RR      E
        DJNZ    SHTRHT
        LD      (HL),D
        PUSH    HL
        PUSH    DE
        LD      HL,(DOTPOSX)    ;DOTPOSX=639?
        LD      DE,0FD81H
        ADD     HL,DE
        POP     DE
        POP     HL
        RET     C
        INC     HL
        LD      (HL),E
        RET
;-----------------------------------------------
;list character from register c
;
LIST:   LD      A,(IOBYTE)      ;lst:=?
        BIT     7,A             ;=tty,crt?
        JP      Z,CONOUT
        BIT     6,A             ;=ull
        JR      NZ,PTPOUT
        LD      A,C
        OUT     (PRINTER),A
;
BUSY:   NOP                     ;screen noise
        IN      A,(PPIC)
        BIT     4,A
        JR      Z,BUSY
        CALL    LDELAY
        IN      A,(PPIA)        ;set stb low
        AND     7FH
        OUT     (PPIA),A
        CALL    LDELAY
        OR      80H             ;set stb high
        OUT     (PPIA),A
        RET
;-----------------------------------------------
;return list status (0 if not ready, 0FF if ready)
;
LISTST: LD      A,(IOBYTE)      ;chk lpt?
        AND     11000000B
```

```
        CP      10000000B
        JR      Z,LPTST
LSTRDY: LD      A,0FFH          ;all others ready
        RET
LPTST:  IN      A,(PPIC)
        BIT     4,A
        JR      NZ,LSTRDY
        XOR     A
        RET
;-----------------------------------------------
;punch character from register c
;
PUNCH:  LD      A,(IOBYTE)      ;tty?
        AND     11000000B
        JP      Z,CONOUT
;
PTPOUT: IN      A,(UTSREG)      ;tx rdy?
        RRA
        JP      NC,PTPOUT
        LD      A,C
        OUT     (UTDATA),A
        RET
;-----------------------------------------------
;read character into register a from reader device
;
READER: LD      A,(IOBYTE)      ;tty?
        AND     11000000B
        JP      Z,CONIN
;
TSTRX:  IN      A,(UTSREG)      ;rx rdy?
        BIT     1,A
        JP      Z,TSTRX
        IN      A,(UTDATA)
        LD      HL,BITMSK
        AND     (HL)
        RET
;-----------------------------------------------
;move to the track 00 position of current drive
;
HOME:   LD      C,00H           ;select track 0
        CALL    SETTRK
        LD      A,(HSTWRT)      ;pending write?
        OR      A
        JP      NZ,HOMED
        LD      (HSTACT),A
HOMED:  RET
;-----------------------------------------------
;select disk given by register C
;
SELDSK: LD      HL,0000H        ;error return code
        LD      A,C
        CP      02H             ;must be 0
        RET     NC              ;no carry if 3,4,5,..
        OR      A
        JR      Z,SELDR0
```

```
        LD      A,(DRV1FL)      ;1st time sel drv 1?
        OR      A
        JR      NZ,SELDR1
        IN      A,(PPIA)        ;see if drv1 ready
        LD      E,A
        OR      20H
        DI
        OUT     (PPIA),A
        LD      A,MTRON
        OUT     (CNTCREG),A
        CALL    HOLD
        IN      A,(FDCCREG)
        RLA
        JR      C,NDRV
        LD      A,0D8H          ;FORCE INTERUPT
        OUT     (FDCCREG),A
        CALL    LDELAY
        LD      A,0D0H
        OUT     (FDCCREG),A
        CALL    LDELAY
        LD      A,0CH           ;recal
        OUT     (FDCCREG),A
        CALL    HOLD
        CALL    HOLD
        LD      A,01
        LD      (DRV1FL),A
        DEC     A
        LD      (DR1TRK),A
        LD      A,E
        OUT     (PPIA),A
        LD      A,MTROFF
        OUT     (CNT2),A
SELDR1: LD      A,C
        LD      (SEKDSK),A
        LD      HL,DPB1
        RET
NDRV:   CALL    DISSEL
        LD      HL,0
        RET
SELDR0:
;disk number is in the proper range
;compute proper disk parameter header address
        LD      A,C             ;C=disk number 0,1
        LD      (SEKDSK),A
        LD      HL,DPB0
        RET
;---------------------------------------------------
;set track given by register c
;
SETTRK: LD      A,C
        LD      (SEKTRK),A
        RET
;---------------------------------------------------
;set sector given by register c
;
```

```
SETSEC: LD      A,C
        LD      (SEKSEC),A
        RET
;---------------------------------------------------
SECTRAN: LD     H,B     ;translate sector number bc
        LD      L,C
        RET
;---------------------------------------------------
;set dma address given by registers b and c
;
SETDMA: LD      L,C             ;low order address
        LD      H,B             ;high order address
        LD      (DMAADR),HL     ;save the address
        RET
;---------------------------------------------------
;the read entry point
;
READ:   XOR     A
        LD      (UNACNT),A
        INC     A
        LD      (READOP),A      ;rd operation
        LD      (RSFLAG),A      ;rd 1 hstsec
        INC     A               ;into hstbuf
        LD      (WRTYPE),A      ;treat as unalloc
        JP      RWOPER
;
;---------------------------------------------------
;the write entry point
;
WRITE:  XOR     A
        LD      (READOP),A      ;wr operation
        LD      A,C             ;wrtype in c
        LD      (WRTYPE),A
        CP      WRUAL           ;wr unalloc?
        JR      NZ,CHKUNA
;wr to unalloc, set params
;
        LD      A,BLKSIZ/128    ;next unalloc rec
        LD      (UNACNT),A
        LD      A,(SEKDSK)
        LD      (UNADSK),A      ;unadsk=sekdsk
        LD      A,(SEKTRK)
        LD      (UNATRK),A      ;unatrk=sectrk
        LD      A,(SEKSEC)
        LD      (UNASEC),A      ;unasec=seksec
;
;check for wr to unalloc sec
;
CHKUNA: LD      A,(UNACNT)      ;unalloc remain?
        OR      A
        JR      Z,ALLOC         ;skip if not
;
;more unalloc rec remain
;
        DEC     A               ;unacnt-1
```

```
        LD      (UNACNT),A
        LD      A,(SEKDSK)      ;same disk?
        LD      HL,UNADSK
        CP      (HL)            ;sekdsk=unadsk?
        JR      NZ,ALLOC        ;skip if not
;
        LD      A,(SEKTRK)      ;sektrk=unatrk?
        LD      HL,UNATRK
        CP      (HL)
        JR      NZ,ALLOC        ;skip if not
;
        LD      A,(SEKSEC)      ;same sec?
        LD      HL,UNASEC
        CP      (HL)            ;seksec=unasec?
        JR      NZ,ALLOC        ;skip if not
;
;match, move to next sec for future ref.
;
        INC     (HL)            ;unasec+1
        LD      A,(HL)          ;end of trk?
        CP      CPMSPT
        JR      C,NOOVF         ;skip if no overflow
;
;overflow to next trk
;
        LD      (HL),0          ;unasec=0
        LD      HL,UNATRK       ;unatrk+1
        INC     (HL)
;
;match found, mark as unnecessary pre-rd
;
NOOVF:  XOR     A
        LD      (RSFLAG),A      ;no pre-read
        JR      RWOPER
;
;not an unalloc rec, may need pre-read
;
ALLOC:  XOR     A
        LD      (UNACNT),A      ;unacnt=0
        INC     A               ;may need pre-rd
        LD      (RSFLAG),A      ;so, rsflag=1
;
;-----------------------------------------------
;common code for read and write follows
;enter here to perform the rd/wr
;
RWOPER: XOR     A
        LD      (ERFLAG),A      ;no error yet
        LD      A,(SEKSEC)      ;compute hstsec
        SRL     A               ;for hstsec=256
        LD      (SEKHST),A      ;store hstsec
;
;active hstsec?
;
        LD      HL,HSTACT       ;hst buf active?
```

```
        LD      A,(HL)
        LD      (HL),1          ;always become 1
        OR      A               ;was it already?
        JR      Z,FILHST        ;fill hst if not
;
        LD      A,(SEKDSK)
        LD      HL,HSTDSK
        CP      (HL)            ;sekdsk=hstdsk?
        JR      NZ,NMATCH
;
        LD      A,(SEKTRK)
        LD      HL,HSTTRK
        CP      (HL)            ;sektrk=hsttrk?
        JR      NZ,NMATCH
;
        LD      A,(SEKHST)
        LD      HL,HSTSEC
        CP      (HL)            ;sekhst=hstsec?
        JR      Z,MATCH         ;skip if match
;
NMATCH: LD      A,(HSTWRT)      ;is wr pending
        OR      A               ;flag set?
        CALL    NZ,WRHST        ;if yes wr to disk
;
;may have to fill the hst buf
;
FILHST: LD      A,(SEKDSK)
        LD      (HSTDSK),A
        LD      A,(SEKTRK)
        LD      (HSTTRK),A
        LD      A,(SEKHST)
        LD      (HSTSEC),A
        LD      A,(RSFLAG)      ;need to read?
        OR      A
        CALL    NZ,RDHST        ;yes, if 1
        XOR     A
        LD      (HSTWRT),A      ;no pending write
;
;copy data to or from buf
;
MATCH:  LD      A,(SEKSEC)      ;mask least sig.
        AND     SECMSK          ;bit of seksec
        LD      L,A             ;HL=A
        LD      H,0
        LD      B,7
MUL128: ADD     HL,HL
        DJNZ    MUL128
;
;********* AMENDED IN GRAPHIC BIOS ***********
;       ADD     HL,HL           ;HL*128
;       ADD     HL,HL
;       ADD     HL,HL
;       ADD     HL,HL
;       ADD     HL,HL
;       ADD     HL,HL
```

```
;       ADD     HL,HL
;********* AMENDED IN GRAPHIC BIOS ***********
        LD      DE,HSTBUF
        ADD     HL,DE
;
;hl has hst buf addr. plus offset
;
        LD      DE,(DMAADR)     ;get/put cp/m data
        LD      BC,128          ;length of move
        LD      A,(READOP)      ;which way?
        OR      A
        JR      NZ,RWMOVE       ;skip if read
;
;wr operation, mark and switch direction
;
        LD      A,1             ;set pending write
        LD      (HSTWRT),A
        EX      DE,HL           ;source/dest swap
RWMOVE: LDIR                    ;(hstbuf)<>(dmaadr)
;
;data has moved to/from hstbuf
;
        LD      A,(WRTYPE)      ;wr type
        CP      WRDIR           ;to directory?
        LD      A,(ERFLAG)      ;in case of error
        RET     NZ              ;no more process
;
;clear hstbuf for dir wr
;
        OR      A               ;error?
        RET     NZ              ;skip if so
        XOR     A
        LD      (HSTWRT),A      ;buf written
        CALL    WRHST
        LD      A,(ERFLAG)
        RET
;
;---------------------------------------------
;
;wrhst performs the phy. write to hst disk
;rdhst performs the phy. read from hst disk
;
;hstdsk=host disk#, hsttrk=host trk#,
;hstsec=host sec#. write "hstziz" bytes
;from hstbuf and return err in erflag
;return erflag non-zero if error
;
WRHST:  XOR     A               ;init rd indica
        LD      (RDIND),A
        JR      RDWR            ;phy wr to disk
;
;hstdsk=host disk#, hsttrk=host track#,
;hstsec=host sec#. read "hstziz" bytes
;into hstbuf and return err in erflag
;return erflag non-zero if error
```

```
;
RDHST:  LD      A,0FFH          ;init rd indica
        LD      (RDIND),A
;
RDWR:   DI
        LD      A,RETRY         ;retry count
        LD      (RTYCNT),A
        XOR     A
        LD      (ERFLAG),A
REREAD: IN      A,(PPIC)        ;m/fdbk=1?
        BIT     MFDBK,A
        JR      NZ,MTON         ;if yes goto mton
ONMTR:  LD      A,MTRON         ;set motor on
        OUT     (CNTCREG),A
        CALL    HOLD            ;delay 0.5 sec
;
MTON:   LD      A,MTRON         ;motor on again
        OUT     (CNTCREG),A
;
PWRUP:  LD      A,0D8H          ;force interupt
        OUT     (FDCCREG),A
        CALL    LDELAY          ;delay 16 usec
        LD      A,0D0H          ;force interupt
        OUT     (FDCCREG),A
        CALL    LDELAY
        CALL    WAIT
        LD      A,(HSTDSK)      ;select drive 1?
        CP      01H
        JR      NZ,DSK0         ;if no go to dsk0
;
        IN      A,(PPIA)
        OR      DRIVE1          ;select drive 1
        OUT     (PPIA),A
        LD      A,(DR1TRK)      ;hsttrk=dr1trk?
        OUT     (TR),A
        LD      B,A
        LD      A,(HSTTRK)
        CP      80
        JP      NC,RDERR
RT1:    LD      (DR1TRK),A
        CP      B
        JR      Z,UDSR
        OUT     (DR),A
        JR      FINDTK
;
DSK0:   IN      A,(PPIA)
        OR      DRIVE0
        OUT     (PPIA),A
        LD      A,(DR0TRK)      ;hsttrk=dr0trk?
        OUT     (TR),A
        LD      B,A
        LD      A,(HSTTRK)
        CP      80
        JP      NC,RDERR
RT2:    LD      (DR0TRK),A
```

```
        CP      B
        JR      Z,UDSR
        OUT     (DR),A
;
FINDTK: IN      A,(FDCSREG)     ;FDC busy?
        RRA
        JP      C,RDERR         ;busy go err
;
        LD      A,SEEK          ;seek track
        OUT     (FDCCREG),A
        LD      HL,1250         ;delay 15 msec
        CALL    HOLDE
        CALL    WAIT            ;seek over?
        AND     10011001B       ;error occour?
        JP      NZ,RDERR        ;if yes, go and exit
UDSR:   LD      A,(HSTSEC)      ;update SR in FDC
        CP      18
        JP      NC,NORTY
        OUT     (SR),A
NOTRDY: CALL    WAIT
        BIT     7,A
        JR      NZ,NOTRDY
;-----------------------
        LD      A,(RDIND)       ;see if rd or wr
        OR      A
        JR      Z,WROP          ;no, then it's wr
;-----------------------
RDOP:   LD      HL,NMI          ;save nmi content
        LD      DE,INSTBF
        LD      BC,6
        LDIR
        LD      HL,RSERVE
        LD      DE,NMI
        LD      BC,6
        LDIR
;
        LD      A,RDSEC         ;load rd sec command
SINGL:  LD      HL,HSTBUF       ;set destin. addr.
        LD      B,0             ;transfer 256 bytes
        LD      C,DR            ;FDC data register
        LD      IX,RDWAIT
;
        OUT     (FDCCREG),A     ;rd sec command
RDWAIT: HALT
        JP      (IX)
ALLRD:  POP     DE
;
        LD      HL,INSTBF
        LD      DE,NMI
        LD      BC,6
        LDIR
;
        CALL    WAIT            ;check error
        AND     10011001B
        JP      Z,DISSEL
```

```
GOERR:  JR      RDERR
;-----------------------
WROP:   IN      A,(PPIC)        ;wprot?
        RLA
        JR      NC,NORTY
        LD      HL,NMI          ;save nmi content
        LD      DE,INSTBF
        LD      BC,6
        LDIR
        LD      HL,WSERVE
        LD      DE,NMI
        LD      BC,6
        LDIR
;
        LD      A,WRSEC         ;load wr sec command
ONESID: LD      HL,HSTBUF       ;set source addr.
        LD      B,00            ;transfer 256 bytes
        LD      C,DR            ;FDC data register
        LD      IX,WRWAIT       ;mask status reg.
;
        OUT     (FDCCREG),A     ;wr sec command
WRWAIT: HALT
        JP      (IX)
ALLWR:  POP     DE
;
PRTCT:  LD      HL,INSTBF       ;restore content
        LD      DE,NMI
        LD      BC,6
        LDIR
;
        CALL    WAIT            ;check error
        AND     11011001B
        JP      Z,DISSEL
;-----------------------
RDERR:  LD      A,RECAL         ;restore drive
        OUT     (FDCCREG),A
        CALL    LDELAY
        CALL    WAIT
        LD      A,(HSTDSK)
        RRA
        JR      C,RESDV1
        XOR     A
        LD      (DR0TRK),A
        JR      TSTRTY
RESDV1: XOR     A
        LD      (DR1TRK),A
TSTRTY: LD      HL,RTYCNT       ;get retry count
        DEC     (HL)
        JP      NZ,REREAD
NORTY:  LD      A,01H           ;return error
        LD      (ERFLAG),A
DISSEL: IN      A,(PPIA)        ;dis-select drive 0,1
        AND     DISDRV
        OUT     (PPIA),A
        LD      A,MTROFF        ;oneshot 5sec
```

```
        OUT     (CNT2),A
        RET
;-----------------------------------------
;subroutine to rd FDC status reg. and wait
;until busy flag (bit 0) is reset
;
WAIT:   IN      A,(FDCSREG)     ;rd stat. reg.
        BIT     0,A
        JR      NZ,WAIT
        RET
;-----------------------------------------
;instructions to be copied to nmi
;
RSERVE: INI                     ;6 bytes
        RET     NZ
        JP      ALLRD
;
WSERVE: OUTI                    ;0066H-0067H
        RET     NZ              ;0068H
        JP      ALLWR
;
;-----------------------------------------
HOLD:   LD      HL,0000H        ;delay 0.7 sec
HOLDE:  DEC     HL              ;12 usec/loop
        LD      A,H
        OR      L
        JR      NZ,HOLDE
        RET
;-----------------------------------------
;subroutine to delay a few micro
LDELAY: PUSH    IX              ;delay
        POP     IX
        PUSH    IX
        POP     IX
        RET
;----- DELETED IN GRAPHIC BIOS ------------------
;subroutine to change bank pass c, acc
;if c=0ff ret current bank in acc
;SWCHBK:        LD      A,C
;       INC     A
;       IN      A,(PPIC)
;       JR      NZ,CHBANK
;       AND     BKIVMS
;       RET
;CHBANK: AND    BKMASK
;       OR      C
;       OUT     (PPIC),A
;       RET
;------- CHANGED IN GRAPHIC BIOS ------------------
;NEWTBL: DB     -1,-1,-1,-1,-1
;        DB     -1,-1,-1,-1,-1
;OLDTBL: DB     -1,-1,-1,-1,-1
;        DB     -1,-1,-1,-1,-1
;----this six ptr/ctr must be contiguous----------
KEYBUF: DB      -1      ;keycode                    ö
```

```
KEYCNT: DB      0       ;no. of keycode                 ö
KEYPTR: DW      0       ;point addr.of key-code         ö
KEYBUF2: DB     -1      ;alternate key buf              ö
KEYCNT2: DB     0       ; * no. of keycode              ö
KEYPTR2: DW     0       ; * point addr. of key-codeö
DOTBIT:  DB     10000000B ;POS. OF DOT IN DOTBYTE
;------ DELETED IN GBIOS ----------------------
;
;AUTO:  DB      7,'AUTORUN',0
;-----------------------------------------------
CCPETY: DW      CCP
HSTBUF  EQU     $                       ;HOST BUFFER
;
BOOT:   DI
        LD      A,0C0H                  ;init ppi
        OUT     (PPIA),A
        LD      A,41H                   ;init modem
        OUT     (70H),A
;
;********** DELETED IN RELEASE 1.1 ************
;       LD      A,0D0H                  ;recal drive 0
;       OUT     (PPIA),A
;       LD      A,MTRON
;       OUT     (CNTCREG),A
;       LD      A,01H
;       LD      (RTYCNT),A              ;retry cnt
;       CALL    HOLD
;       CALL    RDERR
;********** DELETED IN RELEASE 1.1 ************
        XOR     A
        LD      HL,FBEGIN
        LD      B,FLENGTH
FDATA:  LD      (HL),A
        INC     HL
        DJNZ    FDATA
        DEC     A
        LD      B,20
FDATA2: LD      (HL),A
        INC     HL
        DJNZ    FDATA2
;********** DELETED IN GBIOS 1.0 **************
;       LD      E,05H                   ;init lcd
;       LD      C,LCDIST
;       OUT     (C),E
;       XOR     A
;       OUT     (LCDDAT),A
;       INC     E
;       OUT     (C),E
;       OUT     (LCDDAT),A
;       INC     E
;       OUT     (C),E
;       OUT     (LCDDAT),A
;       INC     E
;       OUT     (C),E
;       OUT     (LCDDAT),A
```

```
;********** DELETED IN GBIOS 1.0 **************
;       LD      HL,HEADING      ;print heading
;PRINT: LD      A,(HL)
;       OR      A
;       JR      Z,ENPRIN
;       LD      C,A
;       PUSH    HL
;       CALL    CONOUT
;       POP     HL
;       INC     HL
;       JR      PRINT
        LD      C,1AH           ;CLEAR SCREEN
        CALL    CONOUT
ENPRIN: LD      A,94H           ;init iobyte
        LD      (IOBYTE),A
        XOR     A               ;select drive zero
        LD      (CDISK),A
        JP      WBOOT
;-----------------------------------------
;message to be print out in cboot
;
;HEADING: DB    1BH,'*',0AH,0AH,' CP/M VERSION 2.2  Graphic BIOS 1.0',
;       DB      ' Copyright 1985 by Digital Research',0DH,0AH,0
;-----------------------------------------
;
;DIRBF  128 scratch directory area
;
        ORG     HSTBUF+256      ;beginning of data area
DIRBF:  DS      128
ALL00:  DS      24              ;allocation vector 0
ALL01:  DS      24              ;allocation vector 1
CHK00:  DS      32              ;check vector 0
CHK01:  DS      32              ;check vector 1
;
;-------UNINITIALIZE DATA--------------
;
SEKTRK: DS      1       ;seek track no.
SEKSEC: DS      1       ;seek sector no.
;
HSTDSK: DS      1       ;host disk no.
HSTTRK: DS      1       ;host track no.
HSTSEC: DS      1       ;host sector no.
;
SEKHST: DS      1       ;hold the phy. sec no.
HSTACT: DS      1       ;host active flag
;
UNACNT: DS      1       ;no. of unalloc rec left
UNADSK: DS      1       ;last unalloc disk
UNATRK: DS      1       ;last unalloc track
UNASEC: DS      1       ;last unalloc sec
;
RSFLAG: DS      1       ;read sec flag
READOP: DS      1       ;1 if read operation
WRTYPE: DS      1       ;write operation type
DMAADR: DS      2       ;last dma address
```

```
;
CNTRL:   DS      1          ;nonzero if ctrl pressed
SHIFT:   DS      1          ;nonzero if shift pressed
NPRESS:  DS      1          ;new pressed key-code
TEMP:    DS      1          ;put temp. old key
;
CSNY:    DS      1          ;temp. buffer
WORD:    DS      1          ;store ascii from bdos
WORDBF:  DS      8          ;store char. pattern
BANKBF:  DS      1          ;store prev. bank
;
RDIND:   DS      1          ;phy. rd or wr indicator
RTYCNT:  DS      1          ;hold current try no.
INSTBF:  DS      9          ;hold instruction in nmi
INTBF:   DS      4          ;hold instruction in int
ERFLAG:  DS      1
FLAG4:   DS      1          ;indicate no bad key
;
FBEGIN   EQU     $
FLAG1:   DS      1
FLAG2:   DS      1          ;zero if only 1 oldkey
FLAG3:   DS      1          ;timing indicator
TIMER:   DS      1          ;timer
CAPS:    DS      1          ;nonzero if caplock
CUSRX:   DS      2          ;cusor x-position
CUSRY:   DS      2          ;cusor y-position
CUSRPO:  DS      2          ;cusor top coordinate
VERTIC:  DS      2          ;curnt line top left
VSTART:  DS      2          ;vram starting addr.
ESCFLG:  DS      1          ;indica esc typed
GRFFLG:  DS      1          ;indica graphic mode
INVFLG:  DS      1          ;inverse flag
SEKDSK:  DS      1          ;seek disk no.
HSTWRT:  DS      1          ;write pending flag
DR0TRK:  DS      1          ;current pos. of head
DR1TRK:  DS      1          ;current pos. of head
DRV1FL:  DS      1          ;1st time access drv 1
;
;THE FOLLOWING ARE ADDED IN FOR GRAPHIC BIOS
;
DOTPOSY: DS      1          ;DOT POS. Y
DOTPOSX: DS      2          ;DOT POS. X
DOTBYTE: DS      2          ;LOCATION OF DOT IN VRAM
ESC5:    DS      1          ;ESC5=YX FUNCTION
FLENGTH  EQU     $-FBEGIN
;------- THE FOLLOWING 20 BYTES MUST FOLLOW ABOVE ---
NEWTBL:  DS      10
OLDTBL:  DS      10
FLENG2   EQU     $-FLENGTH
;
         DS      20                 ;temp stack for
STACK2   EQU     $                  ;bank change
         END
;-----------------------------------------------
```