GPC® 184
General Purpose Controller Z80195

TECHNICAL MANUAL
GPC® 184
General Purpose Controller Z80195

TECHNICAL MANUAL

Intelligent Module Abaco® BLOCK, Serie 4, size 100x50 mm. Optional container, format DIN 46277-1 and 3Ω rails compliant. CPU Z80195, with 22,1184 MHz crystal and clock (Z80, Z180 compatible code). Configuration jumper for RUN/DEBUG mode selection and activity LED. Up to 512K of EPROM or FLASH, up to 512K of RAM, serial EEPROM up to 8K. FGDOS uses memory exceeding 64K as RAM/ROM disk. The User can reprogram the on board FLASH memory with his/her application program. Real Time Clock capable to generate INTerrupt. Back Up for RAM and RTC, through on board LITHIUM battery or external battery, with charge status acquired by software. 2 channels of 16 bits Programmable Reload Timer, 4 channels of 8 bits Timer Counters that can be cascaded by software obtaining a counter up to 32 bits. 2 asynchronous serial lines, one in RS232 and one settable as RS232, RS422, RS485 or Current Loop, plus a synchronous or asynchronous TTL serial line. All the three lines are settable by software with separate Baud Rate generators from 50 to 115200 Baud. Double Watch Dog circuitry, hardware and/or software manageable. 26 pins expansion connector with Abaco® I/O BUS standard pin out. 18 digital I/O lines on a standard 20 pins connector. DMA circuit that manages high speed data transfer between memory and peripherals. Numerous interrupt sources, including an efficient Power Failure circuitry. Single 5Vdc power supply with many different power saving modes. On board logic protected against transients by TransZorb™. Wide range of development software available such as Remote Symbolic Debugger; Macro Assembler; GET 80: C compilers (HI TECH C 80, DDS MICRO C 85); PASCAL compilers (PASCAL 80); FGDOS 184; etc.
IMPORTANT

Although all the information contained herein have been carefully verified, grifo® assumes no responsability for errors that might appear in this document, or for damage to things or persons resulting from technical errors, omission and improper use of this manual and of the related software and hardware. grifo® reserves the right to change the contents and form of this document, as well as the features and specification of its products at any time, without prior notice, to obtain always the best product.

For specific informations on the components mounted on the card, please refer to the Data Book of the builder or second sources.

SYMBOLS DESCRIPTION

In the manual could appear the following symbols:

⚠️  Attention: Generic danger

⚠️⚡  Attention: High voltage

Trade Marks

GPC®, grifo®: are trade marks of grifo®.

Other Product and Company names listed, are trade marks of their respective companies.
GENERAL INDEX

INTRODUCTION ................................................................................................................... ..... 1
CARD VERSION ..................................................................................................................... 1
GENERAL FEATURES ........................................................................................................... 2
  CPU .................................................................................................................................... 3
  ABACO® I/O BUS ............................................................................................................. 3
  CLOCK ............................................................................................................................ 4
MEMORY DEVICES .............................................................................................................. 4
  SERIAL COMMUNICATION ............................................................................................ 4
BOARD CONFIGURATION .................................................................................................... 6
  REAL TIME CLOCK ......................................................................................................... 6
WATCH DOGS ..................................................................................................................... 6
TIMERS COUNTERS .......................................................................................................... 6
I/O LINES ........................................................................................................................... 7
CONTROL LOGIC ............................................................................................................... 7
TECHNICAL FEATURES ...................................................................................................... 8
  GENERAL FEATURES ...................................................................................................... 8
  PHYSICAL FEATURES ................................................................................................... 8
  ELECTRIC FEATURES .................................................................................................. 9
INSTALLATION .................................................................................................................... 10
CONNECTIONS .................................................................................................................. 10
  CN2 - BACK UP EXTERNAL BATTERY CONNECTOR .................................................. 10
  CN1 - ABACO® I/O BUS CONNECTOR ....................................................................... 11
  CN3A - SERIAL LINE A CONNECTOR ......................................................................... 12
  CN3B - SERIAL LINE B CONNECTOR ......................................................................... 14
  CN5A - SERIAL LINE C AND AUXILIARY SIGNALS CONNECTOR ......................... 20
  CN5 - PIA, IEEE, CTC I/O CONNECTOR .................................................................. 22
I/O CONNECTION ............................................................................................................... 24
DIGITAL I/O INTERFACES .............................................................................................. 24
VISUAL SIGNALATIONS ................................................................................................... 25
POWER SUPPLY ............................................................................................................... 25
RESET CONTACT ............................................................................................................. 25
JUMPERS ......................................................................................................................... 26
  2 PINS JUMPERS ......................................................................................................... 28
  5 PINS JUMPERS ......................................................................................................... 28
  3 PINS JUMPERS ......................................................................................................... 29
BACK UP .......................................................................................................................... 29
SOLDER JUMPERS .......................................................................................................... 29
RESET AND WATCH DOG .............................................................................................. 30
INTERRUPTS .................................................................................................................... 30
MEMORY SELECTION ....................................................................................................... 31
SERIAL COMMUNICATION SELECTION ...................................................................... 32
POWER FAILURE ............................................................................................................. 35
SOFTWARE ................................................................................................................................. 36
ADDRESSES AND MAPS ............................................................................................................. 39
INTRODUCTION .......................................................................................................................... 39
ADDRESSES ............................................................................................................................... 39
ABACO® I/O BUS ADDRESSES .................................................................................................. 39
MEMORY ADDRESSES ............................................................................................................... 40
I/O ADDRESSES ......................................................................................................................... 41
PERIPHERAL DEVICES SOFTWARE DESCRIPTION .............................................................. 45
REAL TIME CLOCK ................................................................................................................... 45
EXTERNAL WATCH DOG ........................................................................................................... 47
CONFIGURATION JUMPER ......................................................................................................... 47
SERIAL EEPROM ....................................................................................................................... 47
BATTERY STATUS ..................................................................................................................... 48
ACTIVITY LED .......................................................................................................................... 48
CPU INTERNAL PERIPHERALS ............................................................................................... 48
EXTERNAL CARDS ..................................................................................................................... 49
BIBLIOGRAPHY .......................................................................................................................... 52
APPENDIX A: CARD MECHANICAL MOUNTING ................................................................. A-1
APPENDIX B: ON BOARD DEVICES DESCRIPTION ............................................................ B-1
APPENDIX C: ELECTRIC DIAGRAMS .................................................................................. C-1
APPENDIX D: ALPHABETICAL INDEX ................................................................................... D-1
INTRODUCTION

The use of these devices has turned - IN EXCLUSIVE WAY - to specialized personnel.

The purpose of this handbook is to give the necessary information to the cognizant and sure use of the products. They are the result of a continual and systematic elaboration of data and technical tests saved and validated from the manufacturer, related to the inside modes of certainty and quality of the information.

The reported data are destined- IN EXCLUSIVE WAY- to specialized users, that can interact with the devices in safety conditions for the persons, for the machine and for the enviroment, impersonating an elementary diagnostic of breakdowns and of malfunction conditions by performing simple functional verify operations, in the height respect of the actual safety and health norms.

The informations for the installation, the assemblage, the dismantlement, the handling, the adjustment, the reparation and the contingent accessories, devices etc. installation are destined - and then executable - always and in exclusive way from specialized warned and educated personnel, or directly from the TECHNICAL AUTHORIZED ASSISTANCE, in the height respect of the manufacturer recommendations and the actual safety and health norms.

The devices can't be used outside a box. The user must always insert the cards in a container that respect the actual safety normative. The protection of this container is not threshold to the only atmospheric agents, but specially to mechanic, electric, magnetic, etc. ones.

To be on good terms with the products, is necessary guarantee legibility and conservation of the manual, also for future references. In case of deterioration or more easily for technical updates, consult the AUTHORIZED TECHNICAL ASSISTANCE directly.

To prevent problems during card utilization, it is a good practice to read carefully all the informations of this manual. After this reading, the user can use the general index and the alphabetical index, respectly at the begining and at the end of the manual, to find information in a faster and more easy way.

CARD VERSION

The present handbook is reported to the GPC® 184 card release 310700 and later. The validity of the bring informations is subordinate to this card version number. The user must always verify the right correspondence among the two denotations. On the card the release number is present in more points both board printed diagram (serigraph) and printed circuit (for example between the CPU and the memory devices on the component side).
GENERAL FEATURES

GPC® 184 board is a powerful control low cost module capable of operating in stand alone mode or as an intelligent peripheral, and/or remoted, in a wider telecontrol or acquisition network. It is part of the CPU Serie 4, in BLOCK format, as low as 100x50 mm size.

The GPC® 184 module can be secured in a plastic mount for connection to Omega rails DIN 46277-1 and DIN 46277-3, thereby dispensing with the need of a rack and allowing a less costly mounting direct to the electrical control panel. Thanks to this small size, the GPC® 184 put into the same plastic rails that contains the peripheral I/O, i.e ZBx xxx, forms an unique BLOCK element.

The GPC® 184 can also be mounted as a macro CPU module on a peripheral card of the end user, in Piggy Back (stack through) mode.

The powerful rom based FGDOS operating system makes easy to take advantage of the on board resources and allows reduced development time thanks to its support to high level languages like C, PASCAL, etc. FGDOS also lets the User see as RAM/ROM disks the memory devices, making possible to use them immediately and easily by the high level languages file system instructions.

FGDOS can manage PCMCIA RAM cards, LCD or fluorescent displays, a matrix keyboard and a parallel printer through proper interfaces or well developed device, as MCI 64, KDx x24, QTP 24P and QTP 16P obtaining a significant cost reduction. FGDOS is also capable to program a FLASH memory directly on the board.

The features that characterizes the GPC® 184 is the presence of three independent asynchronous serial lines, provided of different electric interfaces that easily solve typical application of: protocol conversion, data logger from serial devices, serial lines concentrator or sorter and so on.

For getting a quick prototype, cards such as SPA 03 and SPA 04 on which it is possible to mount the GPC® 184 in Piggy Back mode, are used. The presence on board of the ABACO® I/O BUS connector, allows to drive directly I/O cards as: ZBR xxx, ZBT xxx, etc. and through ABB 03, ABB 05 it is possible to manage even all the peripheral cards available on Abaco® BUS.

- Intelligent Module Abaco® BLOCK, Serie 4, size 100x50 mm
- Optional container, format DIN 46277-1 and 3 Ω rails compliant
- CPU Z80195, with 22,1184 MHz crystal and clock (Z80, Z180 compatible code)
- Configuration jumper for RUN/DEBUG mode selection and activity LED
- Up to 512K of EPROM or FLASH, up to 512K of RAM, serial EEPROM up to 8K.
- FGDOS uses memory exceeding 64K as RAM/ROM disk. The User can reprogram the on board FLASH memory with his/her application program.
- Real Time Clock capable to generate INTerrupt
- Back Up for RAM and RTC, through on board LITHIUM battery or external battery, with charge status acquired by software.
- 2 channels of 16 bits Programmable Reload Timer, 4 channels of 8 bits Timer Counters that can be cascaded by software obtaining a counter up to 32 bits.
- 2 asynchronous serial lines, one in RS232 and one settable as RS232, RS422, RS485 or Current Loop, plus a synchronous or asynchronous TTL serial line. All the three lines are settable by software with separate Baud Rate generators from 50 to 115200 Baud.
- Double Watch Dog circuitry, hardware and/or software manageable
- 26 pins expansion connector with Abaco® I/O BUS standard pin out.
- 18 digital I/O lines on a standard 20 pins connector
- DMA circuit that manages high speed data transfer between memory and peripherals
- Numerous interrupt sources, including an efficient Power Failure circuitry
- Single 5Vdc power supply with many different power saving modes
- On board logic protected against transients by TransZorb™
- Wide range of development software available such as **Remote Symbolic Debugger**; **Macro Assembler**; **GET 80**; **C compilers** (HI TECH C 80, DDS MICRO C 85); **PASCAL compilers** (PASCAL 80); **FGDOS 184**; etc.

Here follows a description of the board's functional blocks, with an indication of the operations performed by each one. To easily locate these blocks and verify their connections please refer to figure 1.

**CPU**

**GPC® 184** board is designed to employ the **Z80195** CPU manufactured by **ZILOG**. This 8 bits CPU is code compatible with Z80 and Z180, so it features an extended instructions set (170), fast execution time, high speed in data manipulation and efficient vectorized interrupts management. Remarkable is the presence of these peripherals inside the CPU:

- Two 16 bits timers, provided with programmable prescaler (PRT);
- Four 8 bits counters timers (CTC);
- Two asynchronous serial lines capable to manage handshake signals (ASCI);
- One enhanced asynchronous serial line (EMSCC) that supports SDLC protocol;
- Two DMA channels for high speed data transfers (DMAC);
- One bidirectional Centronics interface (IEEE) that shares two 8 bits parallel I/O ports (PIA);
- Memory management unit (MMU);
- One synchronous serial line (CSI/O);
- Internal watch dog timer;
- Interrupt controller;
- 1 MByte of addressable memory;
- Wait states generator to access slow external devices;
- Idle and Stop modes plus many other functionalities, to reduce power consumption;

For further informations about this component please refer to the manufacturer documentation, or see appendix B of this manual. Please remember that the previous list describes the general microprocessor features and some of them can be not supported by the card.

**ABACO® I/O BUS**

One of the most important features of **GPC® 184** is its possibility to be interfaced to industrial **ABACO® I/O BUS**. Thanks to its standard **ABACO® I/O BUS** connector, the card can be connected to some of the numerous **grifo®** boards, both intelligent and not. For example the user can directly use cards for analog signals acquisition, cards for analog signals generation, cards for digital I/O signals management, cards with timers and counters, cards for temperature controls, etc. Also custom boards designed to satisfy specific needs of the end user.

Using **ABB 03** or **ABB 05** mother boards it is possible manage all the BUS **ABACO®** single EURO cards. So **GPC® 184** becomes the right component for each industrial automation system, in fact **ABACO® I/O BUS** makes the card easily expandable with the best price/performance ratio.

The chapter "EXTERNAL DEVICES" shows a short description of these expansion cards.
CLOCK

GPC® 184 is provided with a circuitry that generates the CPU clock frequency (22,1184 MHz); this frequency is used also to generate the frequencies needed to the other sections of the board (Timer, serial lines, internal watch dog, etc.). If the User needs to run slow applications and to reduce the power consumption of the card, the clock frequency can be even divided by software. We would remark that in default condition (after a reset or power on) the CPU clock frequency is always half of the crystal oscillation frequency. However the card is completely tested with the 22.1184 MHz clock frequency and the software development tools provided by grifo® always set this working frequency.

MEMORY DEVICES

On the card can be monted 1032K bytes of memory divided with a maximum of 512K EPROM or FLASH EPROM, 512K static RAM and 8K bytes of EEPROM. The GPC® 184 memory configuration must be chosen considering the application to realize or the specific requirements of the user. Normally the card is equipped with 128K byte of static RAM plus 512 bytes of EEPROM and all different configurations must be specified from the user, at the moment of the order. With the on board back up circuit there is the possibility to keep data, also when power supply is failed; in this way the card is always able to maintain parameters, logged data, system status and configuration, etc. without using expensive external UPS. The back up circuit is supplied by a on board lithium battery and/or an external battery to be connected to a specific connector. The addressing of memory devices is controlled by a specific control logic, that provides to allocate the devices in the microprocessor address space, this control logic automatically manages the different addressing mode and it satisfies the requests of each GPC® 184 software tools. For further information about memory configuration, sockets description and jumpers connection, please refer to "ADDRESSES AND MAPS" and "PERIPHERAL DEVICES SOFTWARE DESCRIPTION" chapters and to "MEMORY SELECTION" paragraph.

SERIAL COMMUNICATION

Serial communication is completely software settable for physical protocol (baud rate, stop bits number, character length, parity and handshakes management) on all the three asynchronous serial lines as well as for the synchronous serial line (bits rate and clock source). These settings are performed programming the microprocessor internal registers of the ASCI, EMSCC and CSI/O sections as described on the manufacturer documentation or in appendix B of this manual. Please remind that the CSI/O synchronous line shares some signals of the ASCI 0 asynchronous line, so they can't be used contemporaneously. By hardware it is possible to select, through some on board jumpers and drivers installation, the electric communication protocol. In detail, one line (A) is always buffered as RS 232, one line(C) is not buffered and it is at TTL level while the third line (B) can be buffered in four different electrical protocols: RS 232, current loop, RS 485 or RS 422; in this last cases also directionality and line activation is programmable. For further information about serial communication please refer to chapters "CONNECTIONS" and "SERIAL COMMUNICATION SELECTION".
Figure 1: Block diagram
BOARD CONFIGURATION

Jumper J4 has been introduced expressly to make the board, and in particular the application program, configurable. The possibility to read by software the status of this jumper gives the user the ability to manage many different conditions by an unique program, without having to employ other input signals (typical applications are: language choice, program parameters definition, operational mode selection, etc.). Some software tools developed for the GPC® 184 board use jumper J4 to select between the RUN and DEBUG operation modalities, as described in the manuals of the tools themselves. Moreover the card has a status LED, software driven, that can be used to show the current configuration, as described in the specific chapter.

REAL TIME CLOCK

GPC® 184 board is provided with a complete Real Time Clock device capable to manage hours, minutes, seconds, day of month, month, year and day of week in stand alone mode. The component is supplied by the back up circuitry to warrant data integrity in every working condition and is completely software programmable acting on 16 registers addressable in the CPU I/O addressing space, by a dedicated control logic. The RTC section can generate interrupts at a software programmable rate, for diverting the CPU from its normal tasks or awakening it from one of its low consumption working modes.

WATCH DOGS

GPC® 184 board is provided with a double watch dog circuit that, if used, allows to exit from endless loop or abnormal conditions not estimated by application program. This circuit is made by two astable sections, one with fixed 1.4 sec intervent time and one with software programmable time, is completely software managed (by accessing registers addressed in the CPU I/O addressing space) and gives the board an extreme degree of safety.

TIMERS COUNTERS

The on board CTC section includes four 8 bits timers counters. Each channel can be set with a different prescaler and with a different operating mode (timer with external trigger, counter of rising edge or falling edge, etc.), including possible interrupt generation. CTC is completely driven by software programming five registers allocated in microprocessor I/O addressing space, by a specific control logic, and it is connected to the external electronics by seven digital signals that are multiplexed with digital I/O lines. Always by software the CTC channel can be cascaded obtaining timer counters with 16, 24 or 32 bits. Moreover the CPU includes the PRT section with two 16 bits timers that can be used to manage all the short temporizations. These timers are software programmable through 7 registers addressed in CPU I/O space.
I/O LINES

The Z80195 CPU incorporates a PIA section and a IEEE bidirectional controller providing in all 24 digital TTL I/O lines. Six of these lines are used for on board resource management (activity LED, battery status, serial EEPROM, etc.) while the remaining 18 are connected to an I/O ABACO® standard connector. The lines direction for 15 of the 18 user available lines is software settable at bit level. These I/O lines add the possibility to connect several devices (for example user interface) even when the handshaking is completely software driven. A specific control logic allows the complete programmations of the internal PIA and IEEE by using ten registers mapped in the CPU I/O space.

CONTROL LOGIC

A specific control logic is responsible of mapping the registers of the on board devices and the memory devices.

The logic allocates these devices in the CPU addressing space, for further informations please refer to the paragraphs "I/O ADDRESS" and "MEMORY ADDRESS"

*FIGURE 2: COMPONENTS MAPS (SOLDER SIDE AND COMPONENTS SIDE)*
TECHNICAL FEATURES

GENERAL FEATURES

Devices:
- Two 16 bit timers (PRT)
- Four 8 bit timers counters (CTC)
- 2 DMA channels (DMAC)
- 1 RS 232 serial line (ASCI1=A)
- 1 RS 232, RS 422, RS 485, current loop serial line (EMSCC=B)
- 1 TTL serial line (ASCI0=C)
- 1 synchronous serial line (CSI/O)
- 18 TTL programmable Input/Output lines (PIA, IEEE)
- 1 reset contact
- 1 activity LED
- 2 astable watch dog
- 1 real time clock
- 1 configuration jumper
- 1 ABACO® I/O BUS interface
- 1 power failure circuit
- 1 back up circuit

Memory:
- IC 2: EPROM from 128K x 8 to 512K x 8
- FLASH EPROM from 128K x 8 to 512K x 8
- IC 5: SRAM from 128K x 8 to 512K x 8
- IC 7: Serial EEPROM from 256 to 8K bytes

Memories acces time: 70 nsec

CPU:
- ZILOG Z80195

Crystal (clock) frequency: 22.1184 (22.1184) MHz

External watch dog reset time: from 940 msec to 2060 msec (typical 1420 msec)

PHYSICAL FEATURES

Size (W x H x D):
- 100 x 50 x 25 mm (without container)
- 110 x 60 x 60 mm (with DIN rails container)

Weight:
- 65 g (without container)
- 120 g (with DIN rails container)

Connectors:
- CN1: 26 pins, male, vertical, low profile connector
- CN2: 2 pins, male, vertical, low profile connector
- CN3A: 6 pins, plug, female
- CN3B: 6 pins, plug, female
- CN5: 20 pins, male, vertical, strip connector
- CN5A: 4+4 pins, male, vertical, strip connector
**Temperature range:** from 0 to 50 Centigrad degrees

**Relative humidity:** 20% up to 90% (without condens)

**ELECTRIC FEATURES**

**Power Supply:** +5 Vdc ±5%

**Consumption on 5 Vdc:**
- 120 mA (default configuration)
- 140 mA (full and maximum configuration)

**On board back up battery:** 3.0 Vdc; 180 mAh

**External back up battery:** 3.6÷5 Vdc

**Back up current:**
- 2.5 µA (on board battery)
- 4.5 µA (external 3.6 V battery)

**RS 422, RS 485 line termination:**
- Line termination resistor= 120 Ω
- Positive pull up resistor= 3.3 KΩ
- Negative pull down resistor= 3.3 KΩ

**Power failure intervent threshold:** 52 mV before reset activation

*Figure 3: Card photo*
INSTALLATION

In this chapter there are the information for a right installation and correct use of the card. The user can find the location and functions of each connectors, jumpers and some explanatory diagrams.

CONNECTIONS

The GPC®184 module has 6 connectors that can be linkeded to other devices or directly to the field, according to system requirements. In this paragraph there are connectors pin out, a short signals description (including the signals direction) and connectors location (see figure 9).

CN2 - BACK UP EXTERNAL BATTERY CONNECTOR

CN2 is a 2 pins, vertical, male connector with 2,54mm pitch. Through CN2 the user can connect an external battery for RAM and RTC back up when the power supply is switched off (for further information please refer to chapter "BACK UP").

Signals description:

+Vbat = 1 - Positive pin of external back up battery.
GND = - Negative pin of external back up battery.
CN1 - ABACO® I/O BUS CONNECTOR

CN1 is a 26 pins, male, vertical, low profile connector with 2.54 mm pitch. Through CN1 the card can be connected to some of the numerous grifo® boards, both intelligent and not. All this connector signals are at TTL level.

**Signals description:**

- **A0-A7** = O - Address BUS.
- **D0-D7** = I/O - Data BUS.
- **/INT BUS** = I - Interrupt request (open collector type).
- **/NMI BUS** = I - Non maskable interrupt (open collector type).
- **/IORQ** = O - Input output request.
- **/RD** = O - Read cycle status.
- **/WR** = O - Write cycle status.
- **/RESET** = O - Reset.
- **+5 Vdc** = I - +5 Vdc power supply.
- **GND** = - Ground signal.
- **N.C.** = - Not connected.

**Figure 5 CN1 - ABACO® I/O BUS CONNECTOR**
CN3A - SERIAL LINE A CONNECTOR

CN3A is a 6 pins, female PLUG connector. On CN3A are available the buffered signals for RS 232 serial line A that is physically connected to the ASCII 1 serial line of the CPU. Placing of the signal has been designed to reduce interference and electrical noise and to simplify connections with other systems, while the electric protocol follows the CCITT normative.

![Diagram of CN3A Serial Line A Connector]

**Signals description:**

- **RXA RS 232** = 1 - Serial line A = ASCII1 RS 232 Receive Data.
- **TXA RS 232** = 0 - Serial line A = ASCII1 RS 232 Transmit Data.
- **CTSA RS 232** = 1 - Serial line A RS 232 Clear To Send, connected to IEEE NINIT signal.
- **RTSA RS 232** = 0 - Serial line A = ASCII1 (*1) RS 232 Request To Send.
- **+5 Vdc** = 0 - +5 Vdc power supply.
- **GND** = - Ground signal

*1: The output handshake signal RTSA is not software manageable so it is kept continuously enabled = +10 Vdc. If this configuration is incompatible with the system to be connected, perform the connection without this signal.
FIGURE 7: SERIAL COMMUNICATION DIAGRAM

Z80 195

RS 232
DRIVERS
RS 422
RS 485
CURRENT LOOP

SERIAL LINE A
SERIAL LINE A
SERIAL LINE B

ASCII
EMSCC

RS 232
DRIVERS

CLOCKED SERIAL LINE
SERIAL LINE C

CN5A
CN3A
CN3B

ASCII 0
CSI/O

GPC 184  Rel. 5.20
Page 13
CN3B - SERIAL LINE B CONNECTOR

CN3B is a 6 pins, female PLUG connector. On CN3B are available the buffered signals for RS 232, RS 422, RS 485, current loop serial line B that is physically connected to the EMSCC serial line of the CPU. Placing of the signal has been designed to reduce interference and electrical noise and to simplify connections with other systems, while the electric protocol follows the CCITT normative.

Signals description:

**RXB RS 232** = I - Serial line B=EMSCC RS 232 Receive Data.
**TXB RS 232** = O - Serial line B=EMSCC RS 232 Transmit Data.
**CTSB RS 232** = I - Serial line B=EMSCC RS 232 Clear To Send.
**RTSB RS 232** = O - Serial line B=EMSCC RS 232 Request To Send.
**RXB- RS 422** = I - Receive Data Negative: Serial line B=EMSCC negative signal for RS 422 serial differential receive.
**RXB+ RS 422** = I - Receive Data Positive: Serial line B=EMSCC positive signal for RS 422 serial differential receive.
**TXB- RS 422** = O - Transmit Data Negative: Serial line B=EMSCC negative signal for RS 422 serial differential transmit.
**TXB+ RS 422** = O - Transmit Data Positive: Serial line B=EMSCC positive signal for RS 422 serial differential transmit.
RXTXB- RS 485 = I/O - Receive Transmit Data Negative: Serial line B=EMSCC negative signal for RS 485 serial differential receive and transmit.

RXTXB+ RS 485 = I/O - Receive Transmit Data Positive: Serial line B=EMSCC positive signal for RS 485 serial differential receive and transmit.

RXB- C.L. = I - Receive Data Negative: Serial line B=EMSCC negative signal for Current Loop serial bipolar receive.

RXB+ C.L. = I - Receive Data Positive: Serial line B=EMSCC positive signal for Current Loop serial bipolar receive.

TXB- C.L. = O - Transmit Data Negative: Serial line B=EMSCC negative signal for Current Loop serial bipolar transmit.

TXB+ C.L. = O - Transmit Data Positive: Serial line B=EMSCC positive signal for Current Loop serial bipolar transmit.

+5 Vdc = O - +5 Vdc or ground signal.

GND = - Ground signal.

**Figure 9: LEDs, Connectors, Memories, etc. Location**
**Figure 10: RS 232 Point to Point Connection Example**

- CN3A,B GPC® 184
- External System

1. RXA RS232, RXB RS232 → TX
2. TXA RS232, TXB RS232 → RX
3. RTSA RS232, RTSB RS232 → CTS
4. CTSA RS232, CTSB RS232 → RTS
5. RXARS232, RXB RS232
6. GND → GND

**Figure 11: RS 422 Point to Point Connection Example**

- CN3B GPC® 184
- External System

1. RXB- RS422 → TX-
2. TXB- RS422 → RX-
3. TXB+ RS422 → RX+
4. RXB+ RS422 → TX+
5. GND → GND

**Figure 12: RS 485 Point to Point Connection Example**

- CN3B GPC® 184
- External System

1. RXTXB- RS485 → TX-,RX-
2. RXTXB+ RS485 → TX+,RX+
3. GND → GND
Figure 13: RS 485 network connection example

Please remark that in a RS 485 network two forcing resistors must be connected across the net and two termination resistors (120 $\Omega$) must be placed at its extremas, respectively near the Master unit and the Slave unit at the greatest distance from the Master.

Forcing and terminating circuitry is installed on GPC® 184 board. It can be enabled or disabled through specific jumpers, as explained later.

For further informations please refer to TEXAS INSTRUMENTS Data-Book, "RS 422 and RS 485 Interface Circuits", the introduction about RS 422-485.
FIGURE 14: 4 WIRES CURRENT LOOP POINT TO POINT CONNECTION EXAMPLE

FIGURE 15: 2 WIRES CURRENT LOOP POINT TO POINT CONNECTION EXAMPLE
Possible current loop connections are two: 2 wires and 4 wires. These connections are shown in figures 14÷16 where it is possible to see the voltage for $V_{CL}$ and the resistances for current limitation ($R$). The supply voltage varies in compliance with the number of connected devices and voltage drop on the connection cable.

The choice of the values for these components must be done considering that:
- circulation of a 20 mA current must be guaranteed;
- potential drop on each transmitter is about 2.35 V with a 20 mA current;
- potential drop on each receiver is about 2.52 V with a 20 mA current;
- in case of shortcircuit each transmitter must dissipate at most 125 mW;
- in case of shortcircuit each receiver must dissipate at most 90 mW.

For further info please refer to HEWLETT-PACKARD Data Book, (HCPL 4100 and 4200 devices).
CN5A - SERIAL LINE C AND AUXILIARY SIGNALS CONNECTOR

CN5A is 2.54 mm pitch, 4+4 pins, male, vertical, strip connector. On CN5A are available the TTL signal of serial line C (that is physically connected to ASCI 0 serial line of the CPU), 3 signals of the CSI/O synchronous serial communication, one /INT1 interrupt signal and the power supply. All signals are at TTL level. The CSI/O line shares some signals of the ASCI 0 line, so they can’t be used contemporaneously.

**Signals description:**

- **CKS** = I/O - Synchronous serial line clock signal.
- **RXS** = I - Synchronous serial line receive signal.
- **TXS** = O - Synchronous serial line transmit signal.
- **/INT1** = I - Interrupt request /INT1.
- **CKA0** = I/O - Serial line B=ASCI 0 baud rate generator clock signal.
- **RXC TTL** = I - Serial line B=ASCI 0 TTL Receive Data.
- **TXC TTL** = O - Serial line B=ASCI 0 TTL Transmit Data.
- **CTSC TTL** = I - Serial line B=ASCI 0 TTL Clear To Send.
- **RTSC TTL** = O - Serial line B=ASCI 0 TTL Request To Send.
- **+5 Vdc** = O - +5 Vdc power supply.
- **GND** = - Ground signal.

**Figure 17: CN5 - Serial line C and auxiliary signals connector**

**Figure 18: TTL point to point connection example**
**Figure 19: CN5A Signals Connection Diagram**

- **Z80195**
  - PIN 1: /INT1
  - PIN 3: CSI/O
  - PIN 4: ASCI 0
  - PIN 5: +5 Vdc
  - PIN 6: TXC TTL, RXS, RTSC TTL, TXS, CTSC TTL, RXS, CKA0, CKS

- **CN5A**
  - PIN 1: CKS, CTA0
  - PIN 2: CTSC TTL, RXS
  - PIN 3: RTSC TTL, TXS
  - PIN 4: RXC TTL
  - PIN 5: CTA1
  - PIN 6: TXC TTL

The diagram illustrates the connections between the Z80195 and CN5A components.
CN5 - PIA, IEEE, CTC I/O CONNECTOR

CN5 is a 20 pins, male, vertical, low profile connector with 2.54 mm pitch. On CN5 are available the 18 digital I/O lines of CPU internal PIA and IEEE, that can be connected to the external world. Some pins of this connector have a double functionality in fact, by software, the CTC internal section can be multiplexed with the I/O signals. All these signals follow TTL standard.

Signals description:

- **PIA1n** = I/O - CPU internal PIA port 1 nth digital signal.
- **PIA2n** = I/O - CPU internal PIA port 2 nth digital signal.
- **NAUTOFD** = I - CPU internal IEEE digital signal.
- **SELECT** = O - CPU internal IEEE digital signal.
- **PERROR** = O - CPU internal IEEE digital signal.
- **CLK Tn** = I - Clock Trigger signal of CTC nth timer counter.
- **ZC Tn** = O - Zero Count signal of CTC nth timer counter.
- **+5 Vdc** = O - +5 Vdc power supply.
- **GND** = - Ground signal.
FIGURE 21: PIA, IEEE DIGITAL I/O LINES DIAGRAM
I/O CONNECTION

To prevent possible connecting problems between GPC® 184 and the external systems, the user has to read carefully the information of the previous paragraphs and he must follow these instructions:

- For RS 232, RS 422, RS 485 or current loop communication signals the user must follow the standard rules of these protocols.

- For all TTL signals the user must follow the rules of this electric standard. The connected digital signals must be always referred to card digital ground and if an electric insulation is necessary, then an opto coupled interface must be connected. For TTL signals, the 0 Vdc level corresponds to logic state "0", while +5 Vdc level corresponds to logic state "1".

DIGITAL I/O INTERFACES

Through CN5 (I/O ABACO® standard connector) the GPC® 184 card can be connected to all of the numerous grifo® boards featuring the same standard pin out. Installation of these interfaces is very easy; in fact only a 20 pins flat cable (code FLT.20+20) is required, while the software management of these interfaces is as easy; in fact most of the software packages available for GPC® 184 card are provided with the necessary procedures. These latter ones are "software driver" that are added to the programming language and allow to use directly high level instructions with all their power. Remarkable modules are:

- QTP 16P, QTP 24P, KDL x24, KDF 224, DEB 01, etc. that solve all the local operator interfacing problems. These modules are provided with all the resources needed to obtain a high level human machine interface (they feature alphanumeric displays, matrix keyboard and visualization LEDs) at a short distance from GPC® 184 card. The available software drivers allow to manage the operator interface resources directly through the high level instructions for console management.

- IAC 01, DEB 01: it is an interface for CENTRONICS parallel printer that can be connected with a standard printer cable. The printer is managed by software through the high level instructions of the selected programming language (LPRINT for BASIC, PRINTF for C, etc.).

- MCI 64: it is a large mass memory support that can directly manage the PCMCIA memory cards RAM, FLASH, ROM, etc.) in their available sizes. About software the developed drivers provide a complete file system interfacing allowing to access the informations stored in the memory card directly through the high level file management instructions.

- RBO xx, TBO xx, XBI xx, OBI xx: these are buffer interfaces for I/O TTL signals. With these modules the the TTL input signals are converted in NPN or PNP optoisolated inputs and the TTL output signals are converted in relays or transistor optoisolated outputs. Some of the above mentioned interfaces can be connected directly to CN5.

For more information, please refer to "EXTERNAL CARDS" chapter and the software tools documentation.
VISUAL SIGNALATIONS

GPC® 184 board is provided of the following visual signalation:

<table>
<thead>
<tr>
<th>LED</th>
<th>COLOUR</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD2</td>
<td>Green</td>
<td>Activity LED, driven by software.</td>
</tr>
</tbody>
</table>

**FIGURE 22: VISUAL SIGNALATIONS TABLE**

The main purpose of this visualization is to provide an immediate indication of the board status or the correct execution of the application program. To easily locate the LED on the board see figure 9 while for a description about its management, please refer to paragraph "ACTIVITY LED".

POWER SUPPLY

The card must be powered only with:

+5 Vdc: that supplies all the on board logic; it must be in the range +5 Vdc ± 5% and must be provided through the pin 25 (GND) and pin 26 (+5Vdc) of CN1 connector.

On the card the power supply signals are available on all the connectors but when the best power lay out is required we suggest to supply power through CN1 and to get it from the other connectors. This explain the direction reported for +5 Vdc signal on each connector of the card. The power supply circuit generates all the necessary voltages for the card and it is designed for reducing the consumption (the microprocessor power down, idle and clock reduction modes are available) and for increasing the electrical noise immunity. In fact, as low as 120 mA of consumption for the normal working mode, allow the User to supply the board by batteries, solar panels, small power supplies, etc. An intersting power failure circuitry capable to detect the imminent power black out is installed on the board, so it can start a software intervent by generating an interrupt.

Please remind that on board there is a protection circuit against voltage peaks by TransZorb™ and that it is a good practise to mantain galvanically isolated from +5 Vdc all the others power voltages of the developed system.

RESET CONTACT

P1 reset contact of the GPC® 184 board allows the user to reset the board and restarting it in a general clearing condition. The main purpose of this contact is to come out of infinite loop conditions, useful especially during debug or to grant a particular initial stat. Please see figure 9 for an easy localization of this contact.
JUMPERS

On GPC® 184 there are 11 jumpers for card configuration, 8 of them are solder jumpers. Connecting these jumpers, the user can define for example the memory type and size, the peripheral devices functionality and so on. Below there is the jumpers list, location and function.

<table>
<thead>
<tr>
<th>JUMPERS</th>
<th>N. PIN</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1</td>
<td>3</td>
<td>It selects SRAM size for IC5.</td>
</tr>
<tr>
<td>J4</td>
<td>2</td>
<td>It selects the connection for RUN/DEBUG user input.</td>
</tr>
<tr>
<td>J6</td>
<td>5</td>
<td>It selects memory type and size for IC2.</td>
</tr>
<tr>
<td>JS1, JS2</td>
<td>2</td>
<td>They connect the termination and forcing circuit to the RS 422, RS 485 serial line B.</td>
</tr>
<tr>
<td>JS3</td>
<td>3</td>
<td>It selects the type of the connection for pin 1 of CN3A.</td>
</tr>
<tr>
<td>JS4</td>
<td>3</td>
<td>It selects the type of the connection for pin 1 of CN3B.</td>
</tr>
<tr>
<td>JS10</td>
<td>2</td>
<td>It activates the external watch dog circuit.</td>
</tr>
<tr>
<td>JS14</td>
<td>2</td>
<td>It connects the on board battery BT1 to the back up circuit.</td>
</tr>
<tr>
<td>JS15</td>
<td>3</td>
<td>It selects direction and activation mode for serial line B in RS 422, RS 485.</td>
</tr>
<tr>
<td>JS19</td>
<td>3</td>
<td>It connects the power failure circuit to microprocessor interrupts.</td>
</tr>
</tbody>
</table>

**Figure 23: Jumpers summarizing table**

The following tables describe all the right connections of GPC® 184 jumpers with their relative functions. To recognize these valid connections, please refer to the board printed diagram (serigraph) or to figure 2 of this manual, where the pins numeration is listed; for recognizing jumpers location, please refer to figures 24, 25.

The "*" used in the following tables, denotes the default connection, or on the other hand the connection set up at the end of testing phase, that is the configuration the user receives.
Figure 24: Jumpers location (Component side)

Figure 25: Jumpers location (Solder side)
2 PINS JUMPERS

<table>
<thead>
<tr>
<th>JUMPERS</th>
<th>CONNECTION</th>
<th>USE</th>
<th>DEF.</th>
</tr>
</thead>
<tbody>
<tr>
<td>J4</td>
<td>not connected</td>
<td>It sets user input at logic level 1 (RUN mode).</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>connected</td>
<td>It sets user input at logic level 0 (DEBUG mode).</td>
<td></td>
</tr>
<tr>
<td>JS1, JS2</td>
<td>not connected</td>
<td>The forcing and terminating circuit is not connected to RS 422 receive or RS 485 serial line B.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>connected</td>
<td>The forcing and terminating circuit is connected to RS 422 receive or RS 485 serial line B.</td>
<td></td>
</tr>
<tr>
<td>JS10</td>
<td>not connected</td>
<td>It disables the external watch dog circuit.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>connected</td>
<td>It enables the external watch dog circuit.</td>
<td></td>
</tr>
<tr>
<td>JS14</td>
<td>not connected</td>
<td>The on board battery BT1 is not connected to the back up circuit.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>connected</td>
<td>The on board battery BT1 is connected to the back up circuit.</td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 26: 2 PINS JUMPERS TABLE**

5 PINS JUMPERS

<table>
<thead>
<tr>
<th>JUMPERS</th>
<th>CONNECTION</th>
<th>USE</th>
<th>DEF.</th>
</tr>
</thead>
<tbody>
<tr>
<td>J6</td>
<td>position 1-2 and 3-4</td>
<td>It configures IC2 socket for EPROM device.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>position 2-3 and 4-5</td>
<td>It configures IC2 socket for FLASH EPROM device.</td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 27: 5 PINS JUMPERS TABLE**
3 PINS JUMPERS

<table>
<thead>
<tr>
<th>JUMPERS</th>
<th>CONNECTION</th>
<th>USE</th>
<th>DEF.</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1</td>
<td>position 1-2</td>
<td>It configures IC5 socket for 128K Bytes SRAM.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>position 2-3</td>
<td>It configures IC5 socket for 512K Bytes SRAM.</td>
<td></td>
</tr>
<tr>
<td>JS3</td>
<td>position 1-2</td>
<td>It connects pin 1 of CN3A to GND.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>position 2-3</td>
<td>It connects pin 1 of CN3A to +5 Vdc.</td>
<td></td>
</tr>
<tr>
<td>JS4</td>
<td>position 1-2</td>
<td>It connects pin 1 of CN3B to GND.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>position 2-3</td>
<td>It connects pin 1 of CN3B to +5 Vdc.</td>
<td></td>
</tr>
<tr>
<td>JS15</td>
<td>Not connected</td>
<td>It configures serial line B for RS 232 electric standard.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>position 1-2</td>
<td>It configures serial line B for RS 485 electric standard (2 wires).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>position 2-3</td>
<td>It configures serial line B for RS 422 electric standard (4 wires).</td>
<td></td>
</tr>
<tr>
<td>JS19</td>
<td>Not connected</td>
<td>It does not connect the power failure circuitry.</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>position 1-2</td>
<td>It connects the power failure circuitry to the CPU /NMI signal.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>position 2-3</td>
<td>It connects the power failure circuitry to the CPU /INT1 signal.</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 28: 3 pins jumpers table**

**BACK UP**

**GPC® 184** has an on board lithium battery BT1 for the back up of SRAM and RTC content when power supply is switched off. Jumper JS14 connects physically the battery so it can be disconnected to save its duration whenever back up is not needed. By CN2 connector it is possible to connect an external battery: configuration of jumper JS14 does not affect the working of this battery and it can replace BT1 completely.

Please refer to paragraph "ELECTRIC FEATURES" to choose the type of external battery, see figure 9 to easily locate the on board battery and read paragraph "BATTERY STATUS" to acquire the back up battery charge status.

**SOLDER JUMPERS**

The solder jumpers called **JSxx** are connected by a thin copper connection on the solder side. So, if one of their configurations has to be changed, the user must first cut this connection using a sharpened cutter, then make the new connection using a low power soldering tool and some non corrosive tin.
RESET AND WATCH DOG

On GPC® 184 there are two separated watch dog circuits that are really efficient and easy to use. The most important features of the external watch dog circuit are:
- astable functionality;
- intervention time of about 1420 msec;
- hardware enable;
- software retrigger;

In astable mode when intervention time is elapsed the circuit becomes active, it stays active till the end of reset time (about 200 msec) and after it is deactivated. Jumpers JS10 connects the watch dog circuit to reset circuit so when it is connected, the watch dog is enabled and viceversa. The external watch dog retrigger is described in the honorimous paragraph.

The most important features of the CPU internal watch dog circuit are:
- astable functionality;
- intervention time programmable by software;
- software enable;
- software retrigger;

and its management is completely described in appendix B.

In response to a /RESET signal activation and successive deactivation the board restarts the execution of the program stored at address 0000H on IC2 (EPROM or FLASH EPROM) starting from a global reset status of all the on board peripheral devices.

Please remark that /RESET signal is connected also to pin 20 of CN1 connector and that on GPC® 184 are available other reset sources as the power good circuit and the contact P1. The two pins of P1 can be connected to a normally open contact (i.e. a push button) and when the contact is closed (shortcut of the two pins) the reset circuit is enabled.

INTERRUPTS

One of the most important GPC® 184 features is the powerful interrupts management. Here is a short description of how the board's hardware interrupt signals can be managed; a more complete description of the hardware interrupts can be found in the microprocessor data sheets or in appendix B of this manual.

- ABACO® I/O BUS -> It generates an /NMI interrupt, by the /NMI BUS signal of CN1 connector.
  It generates an /INT0 non vectored interrupt, by the /INT BUS signal of CN1 connector.
- Real Time Clock -> It generates an /INT2 vectored interrupt.
- Auxiliary signals -> It generates an /INT1 vectored interrupt, by the homonymous CN5A signal.
- Power failure -> It generates an /NMI interrupt or an /INT1 non vectored interrupt according to jumper JS19 connection.
- CPU inside devices -> They generate a vectored interrupt. Possible sources of internal interrupt requests are: PRT 0, PRT 1, DMA 0, DMA 1, CSI/O, ASCI 0, ASCI 1, CTC, EMSCC, etc.
The board features a chained priority structure that manages the case of contemporary interrupts. The addresses of the response procedures for vectored interrupts can be software programmed by the user acting on microprocessor inside registers. So the user program has always the possibility to react promptly to every external event, deciding also the priority of interrupts. In detail the internal peripheral devices have a programmable priority that can be changed by software.

MEMORY SELECTION

On GPC® 184 can be mounted up to 1032K bytes of memory divided in several configurations, as described in the following table:

<table>
<thead>
<tr>
<th>IC</th>
<th>DEVICE</th>
<th>SIZE</th>
<th>JUMPERS POSITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>EPROM</td>
<td>128K Byte</td>
<td>J6 in position 1-2 and 3-4</td>
</tr>
<tr>
<td></td>
<td>EPROM</td>
<td>256K Byte</td>
<td>J6 in position 1-2 and 3-4</td>
</tr>
<tr>
<td></td>
<td>EPROM</td>
<td>512K Byte</td>
<td>J6 in position 1-2 and 3-4</td>
</tr>
<tr>
<td></td>
<td>FLASH EPROM</td>
<td>128K Byte</td>
<td>J6 in position 2-3 and 4-5</td>
</tr>
<tr>
<td></td>
<td>FLASH EPROM</td>
<td>512K Byte</td>
<td>J6 in position 2-3 and 4-5</td>
</tr>
<tr>
<td>5</td>
<td>SRAM</td>
<td>128K Byte</td>
<td>J1 in position 1-2</td>
</tr>
<tr>
<td></td>
<td>SRAM</td>
<td>512K Byte</td>
<td>J1 in position 2-3</td>
</tr>
<tr>
<td>7</td>
<td>EEPROM</td>
<td>256÷8K Byte</td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 29: MEMORY SELECTION TABLE**

The sockets IC2 and IC5 follow the JEDEC standard, so the mounted memory devices must have JEDEC pin outs. IC 7 is a serial memory device that must be specified at the moment of the order and it can be mounted only by grifo® technicians. To determine the signature name of the memory devices that can be mounted, please refer to the manufacturer documentation.

Normally GPC® 184 is supplied in its default configuration with 128K SRAM on IC5 and 512 bytes serial EEPROM on IC7; each different configurations can be defined in the order or self mounted by the user. Below are reported the order codes of the possible memory options:

.512K -> 512K Bytes SRAM
.EE08 -> 1K Bytes serial EEPROM
.EE16 -> 2K Bytes serial EEPROM
.EE64 -> 8K Bytes serial EEPROM

For further informations about memory options and their cost please contact grifo®, while to easily locate the memory devices on the board please refer to figure 9.
SERIAL COMMUNICATION SELECTION

The communication serial line A is always buffered in RS 232, the serial line B can be buffered in RS 232, RS 422, RS 485 or current loop electric standard and finally the serial line C is not buffered (TTL level). By software the serial lines can be programmed to operate with 7, 8, 9 bits per character, parity, between 1 and 2 stop bits at standard or no standard baud rates, through some CPU internal registers setting.

The selection of serial line B electric standards is performed by hardware, through jumpers connection (as described in the previous tables). Some components necessary for RS 422, RS 485 and current loop communication are not mounted and not tested on the default configuration card, so the first not standard configuration must always be executed by grifo® technicians; then the user can change himself the configuration, following the below description:

- SERIAL LINE B=EMSCC CONFIGURED IN RS 232 (default configuration)

<table>
<thead>
<tr>
<th>Component</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC9</td>
<td>MAX 202 driver</td>
</tr>
<tr>
<td>IC10</td>
<td>no component</td>
</tr>
<tr>
<td>IC11</td>
<td>no component</td>
</tr>
<tr>
<td>IC12</td>
<td>no component</td>
</tr>
<tr>
<td>IC13</td>
<td>no component</td>
</tr>
</tbody>
</table>

  JS15 = not connected
  JS1, JS2 = not connected

- SERIAL LINE B=EMSCC CONFIGURED IN CURRENT LOOP (.CLOOP option)

<table>
<thead>
<tr>
<th>Component</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC9</td>
<td>no component</td>
</tr>
<tr>
<td>IC10</td>
<td>no component</td>
</tr>
<tr>
<td>IC11</td>
<td>HP 4200 driver</td>
</tr>
<tr>
<td>IC12</td>
<td>no component</td>
</tr>
<tr>
<td>IC13</td>
<td>HP 4100 driver</td>
</tr>
</tbody>
</table>

  JS15 = not connected
  JS1, JS2 = not connected

  The current loop serial line is a passive line, so during connection the user must provide an external power supply, as described in figures 14÷16. The current loop interface allows either point to point or network connection with 4 or 2 wires.

- SERIAL LINE B=EMSCC CONFIGURED IN RS 422 (.RS422 option)

<table>
<thead>
<tr>
<th>Component</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC9</td>
<td>no component</td>
</tr>
<tr>
<td>IC10</td>
<td>SN 75176 driver</td>
</tr>
<tr>
<td>IC11</td>
<td>no component</td>
</tr>
<tr>
<td>IC12</td>
<td>SN 75176 driver</td>
</tr>
<tr>
<td>IC13</td>
<td>no component</td>
</tr>
</tbody>
</table>

  JS15 = position 2-3
  JS1, JS2 = (*)

  With /RTSB=/RTS signal (managed by software with EMSCC registers) the user enables or disables the transmitter driver:

  /RTSB=/RTS = low level = 0 logic state -> transmitter driver enabled
  /RTSB=/RTS = high level = 1 logic state -> transmitter driver disabled

  allowing either point to point (driver can be maintained always enabled) or network (driver is enabled only when the unit can hold the line) connection.
Serial B = EMSCC in RS 232
Serial B = EMSCC in current loop
Serial B = EMSCC in RS 422
Serial B = EMSCC in RS 485

**Figure 30: Serial communication drivers location**
- SERIAL LINE B=EMSCC CONFIGURED IN RS 485 (.RS485 option)

<table>
<thead>
<tr>
<th>IC</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC9</td>
<td>no component</td>
</tr>
<tr>
<td>J5</td>
<td>position 1-2</td>
</tr>
<tr>
<td>IC10</td>
<td>no component</td>
</tr>
<tr>
<td>JS1, JS2</td>
<td>(*)</td>
</tr>
<tr>
<td>IC11</td>
<td>no component</td>
</tr>
<tr>
<td>IC12</td>
<td>SN 75176 driver</td>
</tr>
<tr>
<td>IC13</td>
<td>no component</td>
</tr>
</tbody>
</table>

With /RTSB=/RTS signal (managed by software with EMSCC registers) the user defines the RS 485 line direction:

- /RTSB=/RTS = low level = 0 logic state -> RS 485 line transmitting
- /RTSB=/RTS = high level = 1 logic state -> RS 485 line receiving

allowing network connection in a master multi slave system and multi master system. With RS 485 communication line, on CN3B the pins 4 and 5 have the double function of reception and transmission signals. All the transmitted characters are at the same time received when the user select RS 485 on GPC® 184; in this way the line conflicts can be immediately recognized by simply testing the received character after each transmission.

(*) With jumper JS1 and JS2 the RS 422 receiving line or the RS 485 line can be terminated and forced with a suitable resistors circuit. The line termination must be added only at the beginning and at the end of the physical line, connecting both the jumpers. Normally these jumper must be connected in point to point network, or on the farther cards in multipoints network.

After reset or power on phase, the /RTSB=/RTS signal is forced to high level that maintain the RS 485 driver receiving and that disables the RS 422 transmitter driver; this condition eliminates any conflict on the communication line.

- SERIAL LINE A=ASCI 1

The output handshake signal RTSA RS 232 is not software manageable so it is kept continuously active = +10 V. If this condition is incompatible with the system to be connected, perform the connection without this signal.

The input handshake signal CTSA RS 232 can be acquired by reading the status of NINIT signal, connected to microprocessor internal IEEE peripheral device, with the following correspondence:

- CTSA RS 232 enabled (+10 V) -> NINIT = 0 logic state
- CTSA RS 232 disabled (-10 V) -> NINIT = 1 logic state

- SERIAL LINE C=ASCI 0

The serial line C is available at TTL level on CN5A connector, with 5 proper signals: TXC TTL, RXC TTL for serial communication; CTSC TTL, RTSC TTL for handshaking and CKA0 for timing signal. This serial line can be easily buffered with any industrial electric standard by using the MSI 01 card.

The serial C handshake and timing signal are multiplied with the CSI/O signals RXS, TXS and CKS, so the serial line C can't be used contemporaneously with the syncronous serial line. The selection between ASCI 0 or CSI/O functionality for these pins is performed by software through the setting of dedicated internal registers.

For further information about serial communication, please refer to connection examples described in figures 10÷16 and 18.
POWER FAILURE

In addition to the CPU controlled power management section, GPC® 184 card also features an efficient power failure circuit. Through jumper JS19 this latter can be connected to the micropocessor /NMI or /INT1 interrupt signal. The task of this circuit is to keep under control power supply voltage and activate on output to request a CPU action when this voltage reaches a value lower than a threshold of 52 mV above the reset intervient. Please remark that the time interval between power failure activation and reset activation changes according to the type of supply being used; it is however about 100 µsec, long enough only to execute a fast response routine (for example to save a flag in the backed SRAM). Typical use of power failure is to inform the board about the imminent power supply black out, so the CPU can save appropriate informations.
SOFTWARE

A wide selection of software development tools can be obtained, allowing use of the module as a system for its own development, both in assembler and in other high level languages; in this way the user can easily develop all the requested application programs in a very short time.

Please remind that all the software development tools provided by grifo® always set the highest clock frequency (22 MHz) to obtain the best performances and they are always accompanied by examples programs that shows how to use each sections of GPC® 184. Generally all software packages available for the mounted microprocessor, or for the Z80 and Z180 family, can be used, i.e.:

GET 80
It is a complete program with Editor, Communication driver, and Mass Memory management for all Z80 family cards. This program, developed by grifo®, allows to operate in the best conditions when GDOS, FGDOS or xGDOS MCI software tools are used; GET 80 is supplied when one of these tools is ordered and it can be personalized with name and general data of the customer. A useful list of pull down menus and the possibility of using mouse, make program use very comfortable.

GET 80 program can be executed both on MS-DOS system and on MACINTOSH computers too, through SOFT-PC program. It is supplied on MS-DOS 3”1/2 floppy disk, or CD, with the documentation on GDOS 80 manual.

GDOS 184
It is a complete development Tool for GPC® 184 card. It is supplied together with GET 80 program to allow an easy and immediate use of this powerful development system. GDOS is divided in two different structures: the first one works on PC maintaining serial communication with the second one. The second structure is on EPROM, it works on board of the card and it is an efficient operating system that executes many low level functions and, at the same time, it allows high level language use. The combination of the said structures results in a complete machine, in fact the card uses PC resources (like floppy disk, hard disk, printer, keyboard, monitor, etc.) as its own peripheral devices.

This resulting “virtual machine” performs operation in a transparent way for the user, so this latter can operate with the same modality of standard PC languages. It is really interesting the compatibility of GDOS with all CP/M program and languages; so, if the user has experience, knowledge or developed applications with CP/M, he can use immediately GDOS, without any changes.

Moreover, GDOS can manage all memory devices exceeding 64K Bytes as RAM disk and ROM disk. The on board RAM devices can directly be used performing data read and write operations with the confortable file formats.

This software tools is supplied on EPROM, floppy disk and CD with some examples, utilities and the operating system documentation.

FGDOS 184
It is really similar to GDOS, but it can program and erase the on board FLASH EPROM with the application program developed from the User. In this way the external EPROM programmer is not necessary to store definitely the program and it is possible to modify or to add code directly on the installed machine, through a portable PC.

This software tools is supplied on FLASH EPROM, floppy disk and CD with some examples, utilities and the operating system documentation.
xGDOS MCI 184
It is a version of GDOS or FGDOS software tools, capable of PCMCIA Memory Card management.
Using MCI 64 card, the GDOS operating system manages memory cards as RAM disk or ROM disk.
All applications with data acquisition and data logging can be realized with high level languages that
manage data on files, with a fast development time and without any software complication.
This software tool is supplied on EPROM or FLASH EPROM, floppy disk and CD with some
examples, utilities and the operating system documentation.

PASCAL 80
It is an efficient and complete PASCAL Compiler for Z80 family cards, with features similar to
Release 3.0 of Borland Turbo PASCAL. It must work together with any GDOS version and it can
exceed the 64K memory limits of Z80 family microprocessors through OVERLAY modality. More
than one application program can be saved in RAM and/or ROM disks and subsequently executed.
The terminal emulation of GET 80 program support the typical full screen PASCAL Editor,
including the attributes management.
This program is supplied as ROM DISK file in GDOS EPROM or FLASH EPROM and on MS-DOS
floppy disk with some example and manual.

HI TECH C 80
Professional C Cross Compiler of Hi-Tech Software Inc. This Compiler is terribly fast and it
generates a small quantity of code. This result is due to advanced techniques in optimizing the
generated code based on Artificial Intelligence techniques which allow to get a very compact and
very fast code. The package includes: IDE, Compiler, Code optimizer, Assembler, Linker, Remote Debugger and so on. This tool is Full ANSI/ISO Standard and Full Library Source Code. Once the porting of the Remote-Debugger module is done, this tool allows the user the software debugging directly on the hardware that he is experimenting. This type of specilization of the Remote Debugger is available from now ant it is supplied with all grifo® CPU cards. This software package is on 3” 1/2 diskettes under MS-DOS format along with user manual. This version supports these following CPUs: Z80, Z180, 84C011, 84C11, 84C013, 80C13, 84C015, 84C15, 64180, NCS800, Z181, Z182 and all the derivate ones.

DDS MICRO C 85
Low cost cross compiler for C source program. It is a powerful software tool that includes editor, C
compiler (integer), assembler, optimizer, linker, library, and remote debugger, in one easy to use
integrated development environment. There are also included the library sources and many utilities
programs.
It is supplied on floppy disks that includes also necessary documentations and examples.

MICRO IDE
It is an Integrated Development Environment for micro controller system application development.
It integrates essential components of firmware developing including: multi file editor, integrations
with toolkit including command line compilers, assemblers, linkers, project manager, many tools
(terminal program, calculator, ASCII chart, etc.), download capability to target microcontroller boards. MICRO IDE is the suitable integrations for DDS MICRO C 85 software tools.
It is supplied on floppy disks that includes also necessary documentations and examples.
NOICE
It is a PC hosted debugger consists of a target specific DOS program, NOICExxx.EXE, and a target resident monitor program. The two programs communicate via RS 232. NOICE includes: source level debug; a disassembler; a file viewer; memory display and editing; a virtually unlimited number of breakpoints; hardware free single step; definition of symbols; the ability to record and play back files of commands; on line help.
It is supplied on EPROM, floppy disk and printed user manual.

RSD 184
This software tools is a Remote Symbolic Debugger with two operating mode. The first one is a monitor debugger modality with software emulation on P.C.; the second is a remote monitor debugger modality that execute code directly on the card. Through serial communication the user can: download an HEX file and associated symbol table, debug code in symbolic mode, execute code in step to step mode or in real time mode, set breakpoint, dump and modify memory and registers, etc.
RSD software tool supports both Z80 and Z180 instruction sets. Really interesting is the program execution management, in fact many hardware and software breakpoint are supported. RSD can be used together with assembler tools, like ZASM 80, and C Compiler CC 80.
It is supplied on EPROM and on MS-DOS floppy disk with technical manual.

ZASM 80
It is a macro cross assember that operates on any PC with MS-DOS operating system. It supports both Z80 and Z180 instruction sets. The generated code can be debugged on PC, through software simulation, or directly on target card, through remote modality, using RSD software tools. ZASM 80 is compatible with C Compiler CC 80 of which it assemble the compilation result.
It is supplied on MS-DOS floppy disk with technical manual.

CC 80
It is a complete C Compiler with ANSI/ISO standard, provided of floating point procedure, that can generate code for Z80 and Z180 family microprocessors. It works together with cross assembler ZASM 80 and Symbolic Debugger RSD.
It is supplied on MS-DOS floppy disk with technical manual.
ADDRESSES AND MAPS

INTRODUCTION

In this chapter are reported all information about card use, related to hardware and software features. For example the registers addresses, the memory and peripheral devices allocation are described below.

ADDRESSES

The on board devices addresses are managed from a control logic, realized with CMOS gates. This control logic allocates memory and peripheral devices with very low power consumption and simple software management.

The control logic has been designed to control the memory and the I/O peripherals addresses in a separate manner. The Z180 microprocessor directly addresses 64K bytes of memory and 256 I/O registers and the control logic provides on board memory and peripheral devices allocation inside these addresses spaces. The maps management is completely driven by software through the MMU circuit programmation: the used memory can be selected and divided in some size definable segments. About I/O maps the control logic avoids every conflicts problems between CPU internal and external peripherals.

Summarizing the control logic allocates:

- **ABACO® I/O BUS**
  - Up to 512K bytes of EPROM or FLASH EPROM on IC 2
  - Up to 512K of RAM on IC 5
  - Up to 8K of serial EEPROM on IC7
  - Configuration jumper J4
  - Real Time Clock
  - External watch dog circuit
  - Activity LED
  - All the microprocessor internal devices (ASCI, ESMCC, PIA, IEEE, CTC, PRT, MMU, DMAC, CSI/O, etc.)

The addresses of all these devices are described in the following paragraphs and can't be set with different values. If some different specific maps are required, please contact directly grifo®.

**ABACO® I/O BUS ADDRESSES**

The **GPC® 184** control logic defines **ABACO® I/O BUS** addresses and only these addresses must be used to manages correctly the BUS. As described in following "I/O ADDRESSES" table, only the addresses 80H÷D7H and F8H÷FFH are available for **ABACO® I/O BUS**. Any I/O operations at each one of these addresses enables the /IORQ signal and the other control signals of CN1 connector.
MEMORY ADDRESSES

The maximum 1032K bytes of memory, are allocated on the board as below described:

- Up to 512K bytes of EPROM or FLASH EPROM allocated in memory space.
- Up to 512K bytes of SRAM allocated in memory space.
- Up to 8K bytes of serial EEPROM allocated in I/O space.

GPC® 184 can directly manage no more than 64K bytes of memory that is the microprocessor logic addressable space. On the board this logic space can be divided in three separated segments: each ones of these segment have software programmable dimension and start address. The CPU internal MMU circuit divides the logical space directly managed by the microprocessor into these 3 segments and it allocates them in the physical memory devices space. The MMU circuit is software programmable with I/O operations to three specific registers in a fast and comfortable manner. So MMU allows software managements of a physical memory space very larger than microprocessor memory space.

The following figure describe available memory configurations; for further informations on MMU use and segments meaning (Common Area 0, Common Area 1 and Bank Area), please refer to appendix B, while for memory devices location and configuration refer to figures 9 and 29.

After power on or reset phase, the MMU circuit allocates all the logical 64K space at the beginning of the physical space, therefore the card starts execution of code saved at address 0000H of EPROM or FLASH EPROM on IC2.

The memory size and type configurations must be selected both according to used software tools and user requests and/or application features. The card configuration for the selected memory device types and sizes on IC2 and IC5 sockets, is performed with some comfortable jumpers, as described in "MEMORY SELECTION" chapter.

Some software tools, i.e. GDOS and FGDOS, self manage the MMU circuit to use all the available memories at high level witout user intervention.
I/O ADDRESSES

The on board control logic manages the allocation of all the peripheral devices registers in the microprocessor I/O space, that normally is 256 bytes long. Next tables shows names, addresses, meanings and directions of peripheral device registers (including the internal microprocessor ones). For a detailed description of the registers function, please refer to next chapter "PERIPHERAL DEVICES SOFTWARE DESCRIPTION".

**Figure 31: Memory Allocation**
<table>
<thead>
<tr>
<th>DEVICES</th>
<th>REG.</th>
<th>ADDRESS</th>
<th>R/W</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCI</td>
<td>CNTLA0</td>
<td>00H</td>
<td>R/W</td>
<td>ASCI 0 control register A</td>
</tr>
<tr>
<td></td>
<td>CNTLA1</td>
<td>01H</td>
<td>R/W</td>
<td>ASCI 1 control register A</td>
</tr>
<tr>
<td></td>
<td>CNTLB0</td>
<td>02H</td>
<td>R/W</td>
<td>ASCI 0 control register B</td>
</tr>
<tr>
<td></td>
<td>CNTLB1</td>
<td>03H</td>
<td>R/W</td>
<td>ASCI 1 control register B</td>
</tr>
<tr>
<td></td>
<td>STAT0</td>
<td>04H</td>
<td>R/W</td>
<td>ASCI 0 status register</td>
</tr>
<tr>
<td></td>
<td>STAT1</td>
<td>05H</td>
<td>R/W</td>
<td>ASCI 1 status register</td>
</tr>
<tr>
<td></td>
<td>TDR0</td>
<td>06H</td>
<td>R/W</td>
<td>ASCI 0 TX data register</td>
</tr>
<tr>
<td></td>
<td>TDR1</td>
<td>07H</td>
<td>R/W</td>
<td>ASCI 1 TX data register</td>
</tr>
<tr>
<td></td>
<td>RDR0</td>
<td>08H</td>
<td>R/W</td>
<td>ASCI 0 RX data register</td>
</tr>
<tr>
<td></td>
<td>RDR1</td>
<td>09H</td>
<td>R/W</td>
<td>ASCI 1 RX data register</td>
</tr>
<tr>
<td></td>
<td>ASEX0T</td>
<td>12H</td>
<td>R/W</td>
<td>ASCI 0 extension control register</td>
</tr>
<tr>
<td></td>
<td>ASEX1T</td>
<td>13H</td>
<td>R/W</td>
<td>ASCI 1 extension control register</td>
</tr>
<tr>
<td></td>
<td>ASTC0L</td>
<td>1AH</td>
<td>R/W</td>
<td>ASCI 0 time constant low register</td>
</tr>
<tr>
<td></td>
<td>ASTC0H</td>
<td>1BH</td>
<td>R/W</td>
<td>ASCI 0 time constant high register</td>
</tr>
<tr>
<td></td>
<td>ASTC1L</td>
<td>1CH</td>
<td>R/W</td>
<td>ASCI 1 time constant low register</td>
</tr>
<tr>
<td></td>
<td>ASTC1H</td>
<td>1DH</td>
<td>R/W</td>
<td>ASCI 1 time constant high register</td>
</tr>
<tr>
<td>CSI/O</td>
<td>CNTR</td>
<td>0AH</td>
<td>R/W</td>
<td>CSI/O control register</td>
</tr>
<tr>
<td></td>
<td>TRDR</td>
<td>0BH</td>
<td>R/W</td>
<td>CSI/O transmit/receive data register</td>
</tr>
<tr>
<td>PRT</td>
<td>TMDR0L</td>
<td>0CH</td>
<td>R/W</td>
<td>Timer 0 data low register</td>
</tr>
<tr>
<td></td>
<td>TMDR0H</td>
<td>0DH</td>
<td>R/W</td>
<td>Timer 0 data high register</td>
</tr>
<tr>
<td></td>
<td>RLD0L</td>
<td>0EH</td>
<td>R/W</td>
<td>Timer 0 reload low register</td>
</tr>
<tr>
<td></td>
<td>RLD0H</td>
<td>0FH</td>
<td>R/W</td>
<td>Timer 0 reload high register</td>
</tr>
<tr>
<td></td>
<td>TCR</td>
<td>10H</td>
<td>R/W</td>
<td>Timer control register</td>
</tr>
<tr>
<td></td>
<td>TMDR1L</td>
<td>14H</td>
<td>R/W</td>
<td>Timer 1 data low register</td>
</tr>
<tr>
<td></td>
<td>TMDR1H</td>
<td>15H</td>
<td>R/W</td>
<td>Timer 1 data high register</td>
</tr>
<tr>
<td></td>
<td>RLDR1L</td>
<td>16H</td>
<td>R/W</td>
<td>Timer 1 reload low register</td>
</tr>
<tr>
<td></td>
<td>RLDR1H</td>
<td>17H</td>
<td>R/W</td>
<td>Timer 1 reload high register</td>
</tr>
<tr>
<td>CPU</td>
<td>FRC</td>
<td>18H</td>
<td>R/W</td>
<td>Free running counter</td>
</tr>
<tr>
<td></td>
<td>CCR</td>
<td>1FH</td>
<td>R/W</td>
<td>CPU control register</td>
</tr>
<tr>
<td></td>
<td>RCR</td>
<td>36H</td>
<td>R/W</td>
<td>Refresh control register</td>
</tr>
<tr>
<td></td>
<td>OMCR</td>
<td>3EH</td>
<td>R/W</td>
<td>Operation mode control register</td>
</tr>
<tr>
<td></td>
<td>ICR</td>
<td>3FH</td>
<td>R/W</td>
<td>I/O control register</td>
</tr>
<tr>
<td></td>
<td>WSG</td>
<td>D8H</td>
<td>R/W</td>
<td>WAIT state chip select register</td>
</tr>
<tr>
<td></td>
<td>SCR</td>
<td>EDH</td>
<td>R/W</td>
<td>System configuration register</td>
</tr>
<tr>
<td></td>
<td>RAMUBR</td>
<td>EAH</td>
<td>R/W</td>
<td>RAM upper boundary register</td>
</tr>
<tr>
<td></td>
<td>RAMLB</td>
<td>EBH</td>
<td>R/W</td>
<td>RAM lower boundary register</td>
</tr>
<tr>
<td></td>
<td>ROMBR</td>
<td>ECH</td>
<td>R/W</td>
<td>ROM address boundary register</td>
</tr>
</tbody>
</table>

**Figure 32: I/O addresses table (1st part)**
<table>
<thead>
<tr>
<th>DEVICES</th>
<th>REG.</th>
<th>ADDRESS</th>
<th>R/W</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMAC</td>
<td>SAR0L</td>
<td>20H</td>
<td>R/W</td>
<td>DMA 0 source address low register</td>
</tr>
<tr>
<td></td>
<td>SAR0H</td>
<td>21H</td>
<td>R/W</td>
<td>DMA 0 source address high register</td>
</tr>
<tr>
<td></td>
<td>SAR0B</td>
<td>22H</td>
<td>R/W</td>
<td>DMA 0 source address boundary register</td>
</tr>
<tr>
<td></td>
<td>DAR0L</td>
<td>23H</td>
<td>R/W</td>
<td>DMA 0 destination address low register</td>
</tr>
<tr>
<td></td>
<td>DAR0H</td>
<td>24H</td>
<td>R/W</td>
<td>DMA 0 destination address high register</td>
</tr>
<tr>
<td></td>
<td>DAR0B</td>
<td>25H</td>
<td>R/W</td>
<td>DMA 0 destination address boundary register</td>
</tr>
<tr>
<td></td>
<td>BCR0L</td>
<td>26H</td>
<td>R/W</td>
<td>DMA 0 byte count low register</td>
</tr>
<tr>
<td></td>
<td>BCR0H</td>
<td>27H</td>
<td>R/W</td>
<td>DMA 0 byte count high register</td>
</tr>
<tr>
<td></td>
<td>MAR1L</td>
<td>28H</td>
<td>R/W</td>
<td>DMA 1 memory address low register</td>
</tr>
<tr>
<td></td>
<td>MAR1H</td>
<td>29H</td>
<td>R/W</td>
<td>DMA 1 memory address high register</td>
</tr>
<tr>
<td></td>
<td>MAR1B</td>
<td>2AH</td>
<td>R/W</td>
<td>DMA 1 memory address boundary register</td>
</tr>
<tr>
<td></td>
<td>IAR1L</td>
<td>2BH</td>
<td>R/W</td>
<td>DMA 1 I/O address low register</td>
</tr>
<tr>
<td></td>
<td>IAR1H</td>
<td>2CH</td>
<td>R/W</td>
<td>DMA 1 I/O address high register</td>
</tr>
<tr>
<td></td>
<td>IAR1B</td>
<td>2DH</td>
<td>R/W</td>
<td>DMA 1 I/O address boundary register</td>
</tr>
<tr>
<td></td>
<td>BCR1L</td>
<td>2EH</td>
<td>R/W</td>
<td>DMA 1 byte count low register</td>
</tr>
<tr>
<td></td>
<td>BCR1H</td>
<td>2FH</td>
<td>R/W</td>
<td>DMA 1 byte count high register</td>
</tr>
<tr>
<td></td>
<td>DSTAT</td>
<td>30H</td>
<td>R/W</td>
<td>DMA status register</td>
</tr>
<tr>
<td></td>
<td>DMODE</td>
<td>31H</td>
<td>R/W</td>
<td>DMA mode register</td>
</tr>
<tr>
<td></td>
<td>DCNTL</td>
<td>32H</td>
<td>R/W</td>
<td>DMA WAIT control register</td>
</tr>
<tr>
<td>INTERRUPT</td>
<td>IL</td>
<td>33H</td>
<td>R/W</td>
<td>Interrupt table low address register</td>
</tr>
<tr>
<td></td>
<td>ITC</td>
<td>34H</td>
<td>R/W</td>
<td>INT TRAP control register</td>
</tr>
<tr>
<td></td>
<td>IER</td>
<td>DFH</td>
<td>R/W</td>
<td>Interrupt edge control register</td>
</tr>
<tr>
<td>MMU</td>
<td>CBR</td>
<td>38H</td>
<td>R/W</td>
<td>Common base register</td>
</tr>
<tr>
<td></td>
<td>BBR</td>
<td>39H</td>
<td>R/W</td>
<td>Bank Base register</td>
</tr>
<tr>
<td></td>
<td>CBAR</td>
<td>3AH</td>
<td>R/W</td>
<td>Common Bank area register</td>
</tr>
<tr>
<td>REAL</td>
<td>SEC1</td>
<td>40H</td>
<td>R/W</td>
<td>Data register for seconds units</td>
</tr>
<tr>
<td>TIME</td>
<td>SEC10</td>
<td>41H</td>
<td>R/W</td>
<td>Data register for seconds decines</td>
</tr>
<tr>
<td>CLOCK</td>
<td>MIN1</td>
<td>42H</td>
<td>R/W</td>
<td>Data register for minutes units</td>
</tr>
<tr>
<td></td>
<td>MIN10</td>
<td>43H</td>
<td>R/W</td>
<td>Data register for minutes decines</td>
</tr>
<tr>
<td></td>
<td>HOU1</td>
<td>44H</td>
<td>R/W</td>
<td>Data register for hours units</td>
</tr>
<tr>
<td></td>
<td>HOU10</td>
<td>45H</td>
<td>R/W</td>
<td>Data register for hours decines and AM/PM</td>
</tr>
<tr>
<td></td>
<td>DAY1</td>
<td>46H</td>
<td>R/W</td>
<td>Data register for day units</td>
</tr>
<tr>
<td></td>
<td>DAY10</td>
<td>47H</td>
<td>R/W</td>
<td>Data register for day decines</td>
</tr>
<tr>
<td></td>
<td>MON1</td>
<td>48H</td>
<td>R/W</td>
<td>Data register for month units</td>
</tr>
<tr>
<td></td>
<td>MON10</td>
<td>49H</td>
<td>R/W</td>
<td>Data register for month decines</td>
</tr>
<tr>
<td></td>
<td>YEA1</td>
<td>4AH</td>
<td>R/W</td>
<td>Data register for year units</td>
</tr>
<tr>
<td></td>
<td>YEA10</td>
<td>4BH</td>
<td>R/W</td>
<td>Data register for year decines</td>
</tr>
<tr>
<td></td>
<td>WEE</td>
<td>4CH</td>
<td>R/W</td>
<td>Data register for week day</td>
</tr>
<tr>
<td></td>
<td>REGD</td>
<td>4DH</td>
<td>R/W</td>
<td>Control register D</td>
</tr>
<tr>
<td></td>
<td>REGE</td>
<td>4EH</td>
<td>R/W</td>
<td>Control register E</td>
</tr>
<tr>
<td></td>
<td>REGF</td>
<td>4FH</td>
<td>R/W</td>
<td>Control register F</td>
</tr>
</tbody>
</table>

**Figure 33: I/O Addresses Table (2nd Part)**
On GPC® 184 there are other devices as activity LED, serial EEPROM, back up circuit that haven't specific management registers. These devices are driven through I/O lines directly connected to CPU internal IEEE interface, as described in the following chapter.

#### FIGURE 34: I/O ADDRESSES TABLE (3RD PART)

<table>
<thead>
<tr>
<th>DEVICES</th>
<th>REG.</th>
<th>ADDRESS</th>
<th>R/W</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXT. WDOG</td>
<td>RWD</td>
<td>40H:7FH</td>
<td>R/W</td>
<td>External watch dog retriger register</td>
</tr>
<tr>
<td>JUMPER J4</td>
<td>RUNDEB</td>
<td>40H:7FH</td>
<td>R</td>
<td>Configuration jumper acquisition register</td>
</tr>
<tr>
<td>ABACO® I/O BUS</td>
<td>I/OBUS</td>
<td>80H:D7H F8H:FFH</td>
<td>R/W</td>
<td>ABACO® I/O BUS addresses</td>
</tr>
<tr>
<td>IEEEEE</td>
<td>PARM</td>
<td>D9H</td>
<td>R/W</td>
<td>Parallel mode register</td>
</tr>
<tr>
<td></td>
<td>PARC</td>
<td>DAH</td>
<td>R/W</td>
<td>Parallel control register</td>
</tr>
<tr>
<td></td>
<td>PARC2</td>
<td>DBH</td>
<td>W</td>
<td>Parallel control register 2</td>
</tr>
<tr>
<td></td>
<td>PART</td>
<td>DCH</td>
<td>R/W</td>
<td>Parallel time constant register</td>
</tr>
<tr>
<td></td>
<td>PARV</td>
<td>DDH</td>
<td>R/W</td>
<td>Parallel vector register</td>
</tr>
<tr>
<td>PIA</td>
<td>PIA1CTC</td>
<td>DEH</td>
<td>R/W</td>
<td>PIA 1 CTC pin select register</td>
</tr>
<tr>
<td></td>
<td>PIA1DIR</td>
<td>E0H</td>
<td>R/W</td>
<td>PIA 1 data direction register</td>
</tr>
<tr>
<td></td>
<td>PIA1D</td>
<td>E1H</td>
<td>R/W</td>
<td>PIA 1 data register</td>
</tr>
<tr>
<td></td>
<td>PIA2DIR</td>
<td>E2H</td>
<td>R/W</td>
<td>PIA 2 data direction register</td>
</tr>
<tr>
<td></td>
<td>PIA2D</td>
<td>E3H</td>
<td>R/W</td>
<td>PIA 2 data register</td>
</tr>
<tr>
<td></td>
<td>PIA2DA</td>
<td>EEH</td>
<td>R/W</td>
<td>PIA 2 data alternate address register</td>
</tr>
<tr>
<td>CTC</td>
<td>CTC0</td>
<td>E4H</td>
<td>R/W</td>
<td>CTC 0 control register</td>
</tr>
<tr>
<td></td>
<td>CTC1</td>
<td>E5H</td>
<td>R/W</td>
<td>CTC 1 control register</td>
</tr>
<tr>
<td></td>
<td>CTC2</td>
<td>E6H</td>
<td>R/W</td>
<td>CTC 2 control register</td>
</tr>
<tr>
<td></td>
<td>CTC3</td>
<td>E7H</td>
<td>R/W</td>
<td>CTC 3 control register</td>
</tr>
<tr>
<td>EMSCC</td>
<td>EMSCCC</td>
<td>E8H</td>
<td>R/W</td>
<td>EMSCC control register</td>
</tr>
<tr>
<td></td>
<td>EMSCCD</td>
<td>E9H</td>
<td>R/W</td>
<td>EMSCC data register</td>
</tr>
<tr>
<td>INT. WDOG</td>
<td>WDTMR</td>
<td>F0H</td>
<td>R/W</td>
<td>Internal watch dog master register</td>
</tr>
<tr>
<td></td>
<td>WDTCSR</td>
<td>F1H</td>
<td>W</td>
<td>Internal watch dog command register</td>
</tr>
</tbody>
</table>
PERIPHERAL DEVICES SOFTWARE DESCRIPTION

In the previous paragraphs are described the external registers addresses, while in this one there is a specific description of registers meaning and function (please refer to I/O addresses tables, for the registers names and addresses values). For a more detailed description of the devices, please refer to manufacturing company documentation while for microprocessor internal peripheral devices, not described in this paragraph, refer to appendix B. In the following paragraphs the D7÷D0 and .0÷.7 indications denotes the eight bits of the combination used in I/O operations.

REAL TIME CLOCK

This peripheral is allocated in 16 consecutives I/O addresses, 3 of which correspond to control and status registers while the remaining 13 are for datas. Data registers are used both for read operations (of the current time and date) and write operations (to initialize the time and date) just like the status registers which are used in write operations (to program the operative mode) and in read operations (to acquire the RTC status). Here follows a list of the RTC data registers' meanings:

- SEC1 - Units of seconds - 4 least significant bits of SEC1.3÷SEC1.0
- SEC10 - Decines of seconds - 3 least significant bits of SEC10.2÷SEC10.0
- MIN1 - Units of minutes - 4 least significant bits of MIN1.3÷MIN1.0
- MIN10 - Decines of minutes - 3 least significant bits of MIN10.2÷MIN10.0
- HOU1 - Units of hours - 4 least significant bits of HOU1.3÷HOU1.0
- DAY1 - Units of day number - 4 least significant bits of DAY1.3÷DAY1.0
- DAY10 - Decines of day number - 2 least significant bits of DAY10.1÷DAY10.0
- MON1 - Units of month - 4 least significant bits of MON1.3÷MON1.0
- MON10 - Decines of month - 1 least significant bit of MON10.0
- YEA1 - Units of year - 4 least significant bits of YEA1.3÷YEA1.0
- YEA10 - Decines of year - 4 least significant bits of YEA10.3÷YEA10.0
- WEE - Day of the week - 3 least significant bits of WEE.2÷WEE.0

The third bit of HOU10.2 indicates AM/PM

For this last register the three least significant bits mean:

<table>
<thead>
<tr>
<th>WEE.2</th>
<th>WEE.1</th>
<th>WEE.0</th>
<th>Day of the week</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Sunday</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Monday</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Tuesday</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Wednesday</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Thursday</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Friday</td>
</tr>
</tbody>
</table>
| 1     | 1     | 0     | Saturday

The meaning of the three control registers is:
bit 7 6 5 4 3 2 1 0
REG D = NU NU NU NU 30S IF B H
where:
NU = Not used.
30S = If high (1) it allows a 30 seconds correction of the time. Once set the RTC seconds are reset and the minutes increased, if the previous seconds was equal or greater than 30.
IF = It manages the RTC interrupt status. When read it shows the current interrupt status (1 active and vice versa), when reset to 0 it disables the RTC interrupt signal if the interrupt mode is selected.
B = Indicates whether R/W operations can be performed on the registers:
   1 -> operations are not permitted and vice versa.
H = If high (1) it stores the written time and date.

bit 7 6 5 4 3 2 1 0
REG E = NU NU NU NU T1 T0 I M
where:
NU = Not used.
T1 T0 = Determine the duration of the internal counters interrupt cycle.
   0 0 -> 1/64 second
   0 1 -> 1 second
   1 0 -> 1 minute
   1 1 -> 1 hour
I = It defines the interrupt operating mode:
   1 -> it selects interrupt mode: when the selected duration elapses the interrupt is enabled and then disabled only with a reset of bit IF of control register D;
   0 -> it selects the standard mode: when the selected duration elapses the interrupt is enabled and then disabled only after 7.8 msec.
M = It mask the interrupt status:
   1 -> interrupt masked: the RTC interrupt signal is always disabled;
   0 -> interrupt not masked: the RTC interrupt signal reflects interrupt status.

bit 7 6 5 4 3 2 1 0
REG F = NU NU NU NU T 24/12 S R
where:
NU = Not used.
T = It determines from which internal counter to take the counting signal:
   1 -> main counter (fast counter for test);
   0 -> 15th counter.
24/12 = It determines the hours counting mode:
   1 -> 0-23;
   0 -> 1-12 with AM/PM.
S = If high (1) it stops the clock time counting until the next enabling (0).
R = If high (1) it resets all the internal counters.

Please remind that RTC registers have the same allocation address of register RUNDEB and RWD, so each RTC registers access imply also external watch dog retrigger.
After a reset or power on none of the RTC registers is setted, so their previous value are mantained.
If back up circuit is connected, the data registers are update even when the card is not powered.
EXTERNAL WATCH DOG

Retrigger operation of GPC® 184 external watch dog circuit is performed with a simple read operation at the addresses of register RWD. This register shares the same addresses of other on board peripherals, but no conflict are generated in fact retrigger operation is an input operation and the read data has no meaning. To avoid watch dog activation it is necessary to retrigger its circuit at regular time periods and the duration of these periods must be smaller than intervent time. If retrigger doesn't happen as before described and JS10 is connected, when intervention time is elapsed, the card is reset. The default intervention time is about 1420 msec but it can considerably change from 940 msec to 2060 msec.

After a reset or power on the external watch dog time is cleared.

CONFIGURATION JUMPER

The J4 configuration jumper installed on the GPC® 184 board can be acquired simply by performing a read operation from RUNDEB registers and masking bit D7. The value is in complemented logic, this means that:

J4 connected -> RUNDEB.7 = 0 value
J4 not connected -> RUNDEB.7 = 1 value

This register shares the same addresses of other on board peripherals, but no conflict are generated in fact jumper acquisition is an input operation and the D0÷D6 data has no meaning. This jumper switches between the RUN (not connected) or the DEBUG (connected) mode, a feature used by some grifo® software tools. Other frequently applications for this jumper are: working conditions setting, selection of some on board firmware parameters, language selection, etc.

To easily locate the jumper on the board please refer to figure 24.

SERIAL EEPROM

For software management of serial EEPROM module installed on IC7, please refer to specific documentation or to demo programs supplied with the card. The user must realize a serial communication with I²C bus standard protocol, through three I/O microprocessor pins of IEEE section. The only necessary information is the electric connection:

<table>
<thead>
<tr>
<th>Signal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSTROBE</td>
<td>DATA input signal (SDA)</td>
</tr>
<tr>
<td>NFAULT</td>
<td>DATA output signal (SDA)</td>
</tr>
<tr>
<td>BUSY</td>
<td>CLOCK signal (SCL)</td>
</tr>
</tbody>
</table>

Known the serial EEPROM management circuitry hardware implementation, please remind that signals A0,A1,A2 of this device's slave address are all set to logic 0. Bit logic status 0 corresponds to low logic status (=0V) of the corresponding signal, while bit logic status 1 corresponds to high logic status (=5V). The first 32 bytes (0÷31) of serial EEPROM are reserved so the user should avoid to modify their value.

For further informations about IEEE signals management please refer to proper technical documentation of appendix B of this manual.
BATTERY STATUS

Status of on board back up circuit of GPC® 184 can be acquired by software, performing a simple acquisition of IEEE NSELECTIN signal, that has the following meaning:

\[
\begin{align*}
\text{NSELECTIN} = 0 & \quad \Rightarrow \quad \text{battery discharged} \quad (\leq 2,265 \text{ V}) \\
\text{NSELECTIN} = 1 & \quad \Rightarrow \quad \text{battery charged} \quad (> 2,265 \text{ V})
\end{align*}
\]

As described in paragraph BACK UP, both the on board battery and the external battery are connected to back up circuit, so the described signal reports the status of both these possible batteries. When both the batteries are not connected (CN2 and JS14 not connected) the battery status can randomly change, but this is a don’t care condition.

For further informations about on board battery and back up circuitry please refer to proper previous paragraphs.

ACTIVITY LED

On board control logic allows the management of an activity LED, called LD2, through the IEEE NACK signal, as below described:

\[
\begin{align*}
\text{NACK} = 0 & \quad \Rightarrow \quad \text{LD2 turned on} \\
\text{NACK} = 1 & \quad \Rightarrow \quad \text{LD2 turned off}
\end{align*}
\]

Signal NACK is set to logic level 1 when a reset or a power on occour, so after one of such events LED is turned off.

CPU INTERNAL PERIPHERALS

The descriptions of the registers that manages the CPU internal peripheral devices (ASCI, CSI/O, PRT, DMAC, INTERRUPT, MMU, CTC, EMSCC, PIA, IEEE, etc) is available in the appendix B.

To correctly use the 80195 microprocessor on GPC® 184 board, the following internal registers setting is suggested:

\[
\begin{align*}
\text{OMCR} & = 80H \\
\text{SCR} & = 28H \\
\text{WDTMR} & = 73H \\
\text{WDTCR} & = B1H \\
\text{DCNTL} & = 70H \\
\text{WSG} & = 00H \\
\text{CCR} & = 80H \\
\text{ICR} & = 00H \\
\text{RCR} & = 00H \\
\text{DSTAT} & = 00H \\
\text{DMODE} & = 00H \\
\text{TCR} & = 00H \\
\text{ITC} & = 00H \\
\text{IER} & = 50H \\
\text{CNTR} & = 07H
\end{align*}
\]

Please use figures 32÷34 for register addresses and appendix B for the meaning of the described values. Whenever this informations are still insufficient, please refer to specific documentation of the manufacturing company.
EXTERNAL CARDS

GPC® 184 can be connected to a wide range of block modules and operator interface system produced by grifo®, or to many system of other companies. The on board resources can be expanded with a simple connection to the numerous peripheral grifo® boards, both intelligent and not, thanks to its standard connectors. Even single EURO cards with BUS ABACO® can be connected, by using the proper mother boards. Hereunder some of these cards are briefly described; ask the detailed information directly to grifo®, if required.

QTP G28
Quick Terminal Panel - LCD Graphic, 28 keys
Intelligent user panel equipped with graphic LCD display 240x128 pixels, CFC backlit; optocoupled RS 232 line and additional RS 232, RS 422, RS 485 or current loop serial line; CAN line controller; E² for set up; RTC and RAM lithium backed; primary graphic object; possibility of renaming keys, LEDs and panel name by inserting label with new name into the proper slot; 28 keys and 16 LEDs with blinking attribute and buzzer manageable by software; built in power supply; reader of magnetic badge and relay option.

ADC 812
Analog to Digital Converter, 12 bits, multi range
DAS (Data Acquisition System) multi range 8 channels 12 bit A/D conversion lines; track and hold; 6µs conversion time; range ±10, ±5, +10, +5Vdc or 0÷20, 4÷20mA; analog inputs connections through quick terminal screw connectors; ABACO® I/O BUS interface; direct mounting for DIN 247277-1 and 3 rails; 4 type dimension.

DAC 212
Digital to Analog Converter 12 bits, multi range
Digital to Analog converter; multi range 2 channels 12 bits ± 10, +10 Vdc output; analog outputs connections through quick terminal screw connectors; ABACO® I/O BUS interface; direct mounting for DIN 247277-1 and 3 rails; 4 type dimension.

CAN 14
Control Area Network, 1 channel, galvanically insulated
UART CAN SJA1000; 1 serial channels galvanically insulated; ABACO® I/O BUS interface; 4 type dimension; support of CAN 2.0B protocol; transfer rate up to 1M bit/sec; direct mounting for DIN 247277-1 and 3 rails.

ETI 324
Encoder Timer I/O, 3 counters, 24 I/O
Three timers counters driven by 82C54; bidirectional optocoupled encoder input; direction identifier; phases multiplier; 24 digital lines driven by 82C55 on two standard I/O ABACO® connectors; ABACO® I/O BUS interface; direct mounting for DIN 247277-1 and 3 rails; 4 type dimension.

ABB 05
ABACO® Block BUS 5 slots
5 slots ABACO® mother board with power supply. Double power supply built in; 5Vdc 2,5A section for powering the on board logic; second section at 24Vdc 400mA galvanically coupled, for the optocoupled input lines. Auxiliary connector for ABACO® I/O BUS. Connection for DIN Ω rails.
ABB 03
ABACO® Block BUS 3 slots
3 slots ABACO® mother board; 4 TE pitch connectors; ABACO® I/O BUS connector; screw terminal for power supply; connection for DIN C type and Ω rails.

ZBT xxx
Zipped BLOCK Transistors xy Input + yz Output
Peripheral cards family having xy optocoupled inputs and yz 3A open collector transistor outputs; plastic container for Ω rails mounting; double power supply, galvanically coupled, for the optocoupled input lines and for the logic plus external card. I/O lines displayed by LEDs; transistors outputs equipped with protection against inductive loads; I/O connections available on easy quick terminal connectors; interface to ABACO® I/O BUS. The following models are available: xxx=324 -> 32 In and 24 Out; xxx=246 -> 24 In and 16 Out; xxx=168 -> 16 In and 8 Out; xxx=84 -> 8 In and 4 Out.

IBC 01
Interface Block Communication
Conversion card for serial communication, 2 RS 232 lines; 1 RS 422 or RS 485 line; 1 optical fibre line; selectable DTE/DCE interface; quick connection for DIN 46277-1 and 3 rails.

FBC xxx
Flat Block Contactxxx pins
This interconnection system "wire to board" allows the connection to many type of flat cable connectors to terminal for external connections. Connection for DIN Ω rails.

MCI 64
Memory Cards Interfaces 64 MBytes
Interfacing card for managing 68 pins PCMCIA memory cards, it is directly driven from any ABACO® I/O standard connector; High level languages GDOS supported.

OBI N8 - OBI P8
Opto BLOCK Input NPN-PNP
Interface between 8 NPN, PNP optocoupled and displayed input lines, with screw terminal and ABACO® standard I/O 20 pins connector; power supply section; connection for DIN Ω rails.

TBO 01 - TBO 08
Transistor BLOCK Output
Interface for ABACO® standard I/O 20 pins connector; 16 or 8 transistor output lines 45 Vdc 3 A open collector; screw terminal; optocoupled and displayed lines; connection for DIN 247277-1 and 3 rails.

MSI 01
Multi Serial Interface
Interface for TTL serial line and a buffered RS 232, RS 422, RS 485 or current loop serial line. TTL signals on screw terminal connector and buffered signals on standard plug connector.

QTP 16P
Quick Terminal Panel 16 key with parallel interface
User panel equipped with Fluorescent or LCD display, LEDs backlit, 20x2 or 20x4 characters and keyboard with 16 keys. It is directly driven from 16 TTL I/O lines and it has power supply options.
**Figure 35: Available Connections Diagram**

- **Hardware Serial Line**: RS-232, RS-422, RS-485, Current Loop
- **18 Digital TTL I/O Lines**
  - Direct to XBI 01, OBI 01, RBO 08, etc.
  - Opto-Relay
- **Memory, Shift Registers, Peripherals, or any synchronous serial interface (CSI/O, SPI, Microwire) devices**
- **TTL Serial Line**: GPC R/T94
- **PC-like or Macintosh**: QTP xxP
- **Digital I/O Interfaces**: 18 Digital TTL I/O Lines direct to XBI 01, OBI 01, RBO 08, etc...
- **Opto-Relay**: TRANSISTOR COUPLED
- **ABACO® I/O BUS**: ABACO® BUS
- **ZBX series**: ZBx series
- **ABB 03 or ABB 05, etc.**
- **External Lithium Battery**: 3.6V
- **ABACO® BUS**: External Lithium Battery 3.6V
DEB 01
Didactic Experimental Board
Supporting card for 16 TTL I/O lines use. It includes: 16 keys, 16 LEDs, 4 digits, 16 keys matrix keyboard, Centronics printer interface, LCD display and fluorescent display interface, GPC® 68 I/O connector, field connection with screw terminal.

IPC 52
Intelligent Peripheral Controller, 24 analogic input
This intelligent peripheral card acquires 24 independent analogic input lines: 8 PT 100 or PT 1000 sensors, 8 J,K,S,T termocouples, 8 analog input ±2Vdc or 4÷20mA; 16 bits + sign A/D section; 0.1 °C resolution; 32K RAM for local data logging; buzzer; 16 TTL I/O lines; 5 or 8 conversion per second; facility of networking up to 127 IPC 52 cards using serial line. BUS interfacing or through RS 232, RS 422, RS 485 or current loop line. Only 5Vdc power supply.

JMS 34
Jumbo Multifunction Support for Axis control
Generic peripheral axis control card. 3 optocoupled acquisition channels, with 16 bits bidirectional counter, for incremental encoder. 4 12bits ±10Vdc D/A channels. 8 Opto-in; 8 NPN Opto-output 40Vdc 500 mA. All I/O lines displayed with LEDs.

BIBLIOGRAPHY
In this chapter there is a complete list of technical books, where the user can find all the necessary documentations on the components mounted on GPC® 184.

Data book Manuale TEXAS INSTRUMENTS: The TTL Data Book - SN54/74 Families
Data book Manuale TEXAS INSTRUMENTS: RS-422 and RS-485 Interface Circuits
Data book NEC: Memory Products
Data book AMD: Flash Memory Products
Data book TOSHIBA: Mos Memory Products
Data book XICOR: Data Book
Data book HEWLETT PACKARD: Optoelectronics Designer's Catalog
Data book MAXIM: New Releases Data Book - Volume IV
Data book MAXIM: New Releases Data Book - Volume V
Data sheets SEIKO EPSON: RTC-62421 Real Time Clock module

For further information and upgrades please refer also to specific internet web pages of the manufacturing companies.
The GPC® 184 can be physically mounted in two different manners. The first is the piggy back mounting (stack trough mode) that use the three connectors CN1, CN5 and CN5A for the connection to a user developed board. This connectors lead out of 7 mm on solder side and the user board must have proper female strip connectors (2.54 mm pitch) where the card can be plugged in, obtaining a single system.

The second mode expect a mounting inside a proper plastic container for a direct mounting on DIN 247277-1 and 3 Ω rails (order code BLOCK.100.50); if the card is used with some other peripheral cards (i.e. ZBR xxx or ZBT xxx), a single longer container can be used obtaining a single module. The described long plastic container code is 414487 type RS/100 by Weidmuller and it can be ordered to grifo® as .EXT-WMIII options, where III is the required length. By selecting this mounting the electric connection between GPC® 184 and other peripheral cards is performed with a flat cable that must be really short, as the FLT.26+26 I/O for ABACO® I/O BUS signals.

In the following figures are described the module dimensions with the connector positions and some images that illustrate the connection modes.
FIGURE A2: PIGGY BACK MOUNTING

FIGURE A3: WEIDMULLER RAIL MOUNTING
APPENDIX B: ON BOARD DEVICES DESCRIPTION

PRELIMINARY PRODUCT SPECIFICATION

Z80185/Z80195
SMART PERIPHERAL CONTROLLERS

FEATURES

- Enhanced Z8S180 MPU
- Four Z80 CTC Channels
- One Channel ESCC™ Controller
- Two 8-Bit Parallel I/O Ports
- Bidirectional Centronics Interface (IEEE 1284)
- Low-EMI Option

GENERAL DESCRIPTION

The Z80185 and Z80195 are smart peripheral controller devices designed for general data communications applications, and architected specifically to accommodate all input and output (I/O) requirements for serial and parallel connectivity. Combining a high-performance CPU core with a variety of system and I/O resources, the Z80185/195 are useful in a broad range of applications. The Z80195 is the ROMless version of the device.

The Z80185 and Z80195 feature an enhanced Z8S180 microprocessor linked with one enhanced channel of the Z85230 ESCC™ serial communications controller, and 25 bits of parallel I/O, allowing software code compatibility with existing software code.

Seventeen lines can be configured as bidirectional Centronics (IEEE 1284) controllers. When configured as a 1284 controller, an I/O line can operate in either the host or peripheral role in compatible, nibble, byte or ECP mode. In addition, the Z80185 includes 32 Kbytes of on-chip ROM.

These devices are well-suited for external modems using a parallel interface, protocol translators, and cost-effective WAN adapters. The Z80185/195 is ideal for handling all laser printer I/O, as well as the main processor in cost-effective printer applications.

Notes:
- All signals with a preceding front slash, "/", are active Low.
- Power connections follow conventional descriptions below.

<table>
<thead>
<tr>
<th>Connection</th>
<th>Circuit</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>VCC</td>
<td>VDD</td>
</tr>
<tr>
<td>Ground</td>
<td>GND</td>
<td>VSS</td>
</tr>
</tbody>
</table>

Figure 1. Z80185/195 Functional Block Diagram

TIMING DIAGRAMS (Continued)
Z80185 MPU FUNCTIONAL DESCRIPTION

The Z80185 includes a Zilog Z8S180 MPU (Static Z80180 MPU). This allows software code compatibility with existing Z80/Z180 software code. The following is an overview of the major functional units of the Z80185.

The MPU portion of the Z80185 is the Z8S180 core with added features and modifications. The single-channel EMSCC of the Z80185 is compatible with the Z85233 EMSCC and features additional enhancements for LocalTalk and the demultiplexing of the /DTR//REQ and /WT//REQ lines.

Architecture

The Z80185 combines a high performance CPU core with a variety of system and I/O resources useful in a broad range of applications. The CPU core consists of four functional blocks:

1. **Clock Generator**
   - This logic generates the system clock from either an external crystal or clock input. The external clock is divided by two, or one if programmed, and is provided to both internal and external devices.

2. **Bus State Controller**
   - This logic performs all of the status and bus control activity associated with both the CPU and some on-chip peripherals. This includes wait state timing, reset cycles, DRAM refresh, and DMA bus exchanges.

3. **interrupt Controller**
   - This logic monitors and prioritizes the variety of internal and external interrupts and traps to provide the correct responses from the CPU. To maintain compatibility with the Z80 CPU, three different interrupt modes are supported.

4. **Memory Management Unit**
   - The MMU allows the user to "map" the memory used by the CPU (logically only 64 Kbytes) into the 1 Mbyte addressing range supported by the Z80185. The organization of the MMU object code maintains compatibility with the Z80 CPU while offering access to an extended memory space. This is accomplished by using an effective "common area-banked area" scheme.

5. **Central Processing Unit**
   - The CPU is microcoded to provide a core that is object-code compatible with the Z80 CPU. It also provides a superset of the Z80 instruction set, including 8-bit multiply. This core has been modified to allow many of the instructions to execute in fewer clock cycles.

6. **DMA Controller**
   - The DMA controller provides high-speed transfers between memory and I/O devices. Transfer operations supported are memory-to-memory, memory to or from I/O, and I/O-to-I/O. Transfer modes supported are request, burst, and cycle steal. DMA transfers can access the full 1 Mbyte addressing range with a block length up to 64 Kbytes, and can cross over the 64 Kbyte boundaries.

Figure 21. Z8S180 MPU Block Diagram
DMA Controller

The two DMA channels of the Z80185 can transfer data to or from the EMSCC channel, the parallel interface, the async ports, or an external device. The I/O device encoding in SAR18-16 and DAR18-16 of the existing Z80180 is modified as shown in Table 1.

Table 1. SAR18-16 and DAR18-16 I/O Device Encoding

<table>
<thead>
<tr>
<th>SM1-0</th>
<th>SAR18-16</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>11 0</td>
<td>0 0 0</td>
<td>ext (TOUT/DREQ)</td>
</tr>
<tr>
<td>11 0</td>
<td>0 0 1</td>
<td>ASCI0 Rx</td>
</tr>
<tr>
<td>11 0</td>
<td>0 1 0</td>
<td>ASCI1 Rx</td>
</tr>
<tr>
<td>11 0</td>
<td>0 1 1</td>
<td>EMSCC Rx</td>
</tr>
<tr>
<td>11 0</td>
<td>1 0 0</td>
<td>Reserved, do not program.</td>
</tr>
<tr>
<td>11 1</td>
<td>0 0 0</td>
<td>PIA27-20 in</td>
</tr>
<tr>
<td>11 1</td>
<td>0 0 1</td>
<td>PIA27-20 out</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DM1-0</th>
<th>DAR18-16</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>11 0</td>
<td>0 0 0</td>
<td>ext (TOUT/DREQ)</td>
</tr>
<tr>
<td>11 0</td>
<td>0 0 1</td>
<td>ASCI0 Tx</td>
</tr>
<tr>
<td>11 0</td>
<td>0 1 0</td>
<td>ASCI1 Tx</td>
</tr>
<tr>
<td>11 0</td>
<td>0 1 1</td>
<td>EMSCC Tx</td>
</tr>
<tr>
<td>11 0</td>
<td>1 0 0</td>
<td>Reserved, do not program.</td>
</tr>
<tr>
<td>11 1</td>
<td>0 0 0</td>
<td>PIA27-20 in</td>
</tr>
<tr>
<td>11 1</td>
<td>0 0 1</td>
<td>PIA27-20 out</td>
</tr>
</tbody>
</table>

Asynchronous Serial Communications Interface (ASCI)

The ASCI logic provides two individual full-duplex UARTs. Each channel includes a programmable baud rate generator and modem control signals. The ASCI channel can also support an asynchronous bus communications format. Each channel has up to three modem control signals and one clock signal can be pinned out, while ASCI1 has a data-only interface.

The receiver includes a 4-byte FIFO, plus a shift register as shown in Figure 22.

Error Condition Handling

When the receiver places a data character in the Receive FIFO, it also places any associated error conditions in the error FIFO. The outputs of the error FIFO go to the set inputs of the software-accessible error latches. Writing a 0 to the CNTLA register is the only way to clear these latches. In other words, when a receive error bit reaches the top of the FIFO, it sets an error latch. If the FIFO has more data and the software reads the next byte out of the FIFO, the error latch remains set, until the software writes a 0 to the EFR bit. The error bits are cumulative, so if additional errors are in the FIFO, they will set any unset error latches as they reach the top.

Overrun Error

An overrun occurs if the receive FIFO is full when the receiver has just assembled a byte in the shift register and is ready to transfer it to the FIFO. If this occurs, a break error bit associated with the previous byte in the FIFO is set. The latest data byte is not transferred on the shift register to the FIFO in this case, and is lost. Once an overrun occurs, the receiver does not place any further data in the FIFO until the last good byte received has been transferred to the FIFO so that the Overrun latch is set, and software then clears the Overrun latch. Assembly of bytes continues in the shift register, but this data is ignored until the byte with the overrun error reaches the top of the FIFO and is cleared with a write of 0 to the EFR bit.

Error Condition Handling

When the receiver places a data character in the Receive FIFO, it also places any associated error conditions in the error FIFO. The outputs of the error FIFO go to the set inputs of the software-accessible error latches. Writing a 0 to the CNTLA register is the only way to clear these latches. In other words, when a receive error bit reaches the top of the FIFO, it sets an error latch. If the FIFO has more data and the software reads the next byte out of the FIFO, the error latch remains set, until the software writes a 0 to the EFR bit. The error bits are cumulative, so if additional errors are in the FIFO, they will set any unset error latches as they reach the top.

Overrun Error

An overrun occurs if the receive FIFO is full when the receiver has just assembled a byte in the shift register and is ready to transfer it to the FIFO. If this occurs, a break error bit associated with the previous byte in the FIFO is set. The latest data byte is not transferred on the shift register to the FIFO in this case, and is lost. Once an overrun occurs, the receiver does not place any further data in the FIFO until the last good byte received has been transferred to the FIFO so that the Overrun latch is set, and software then clears the Overrun latch. Assembly of bytes continues in the shift register, but this data is ignored until the byte with the overrun error reaches the top of the FIFO and is cleared with a write of 0 to the EFR bit.

Break Detect

A break is defined as a framing error with the data equal to all zeros. When a break occurs, the all-zero byte with its associated error bits are transferred to the FIFO, if it is not full. If the data is full, an overrun is generated, but the break framing error and data, are not transferred to the FIFO. Any time a break is detected, the receiver will not receive any more data until the data pin returns to a high state. If the channel is set in multiprocessor mode and the EMI bit of the CNTLA register is set to 1, then breaks are ignored unless the MP bit in the transmission is 1. Note: The two conditions listed above could cause a break condition to be missed if the FIFO is full and the break occurs, or if the MP bit on the transmission is not 1 with the conditions specified above.

Parity and Framing Errors

Parity and Framing Errors do not affect subsequent receiver operation.
Baud Rate Generator

The Baud Rate Generator (BRG) has two modes. The first is the same as in the Z80180. The second is a 16-bit down counter that divides the processor clock by the value in a 16-bit time constant register, and is identical to the EMSCC BRG. This allows a common baud rate of up to 512 Kbps to be selected. The BRG can also be disabled in favor of an external clock on the CKA pin.

The Receiver and Transmitter will subsequently divide the output of the BRG (or the signal from the CKA pin) by 1, 16 or 64, under the control of the DR bit in the CNTLB register, and the X1 bit in the ASCI Extension Control Register. To compute baud rate, use the following formulas.

If $ss2,1,0 = 111$, baud rate $= f_{\text{CKA}} / \text{Clock mode}$

else if BRG mode baud rate $= f_{\text{PHI}} / (2 \times (TC+2) \times \text{Clock mode})$

else baud rate $= f_{\text{PHI}} / ((10 + 20*PS) \times 2^{ss} \times \text{Clock mode})$

Where:

BRG mode is bit 3 of the ASEXT register

PS is bit 5 of the CNTLB register

TC is the 16-bit value in the ASCII Time Constant registers

The TC value for a given baud rate is:

$TC = (f_{\text{PHI}} / (2 \times \text{baud rate} \times \text{Clock mode})) - 2$

Clock mode depends on bit 4 in ASEXT and bit 3 in CNTLB:

$X1 \text{ DR Clock Mode}$

<table>
<thead>
<tr>
<th>X1</th>
<th>DR</th>
<th>Clock Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>64</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Reserved, do not use.</td>
</tr>
</tbody>
</table>

$2^{ss}$ depends on the three LS bits of the CNTLB register:

<table>
<thead>
<tr>
<th>ss2</th>
<th>ss1</th>
<th>ss0</th>
<th>$2^{ss}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>32</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>64</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>External Clock from CKA0 (see above)</td>
</tr>
</tbody>
</table>

The ASCIs require a 50 percent duty cycle when CKA is used as an input. Minimum High and Low times on CKA are typical of most CMOS devices.

RDFR is set, and if enabled an Rx Interrupt or DMA Request is generated, when the receiver transfers a character from the Rx Shift Register to the Rx FIFO. The FIFO merely provides margin against overruns. When there's more than one character in the FIFO, and software or a DMA channel reads a character, a RDRF flag is set. Once this bit is cleared, the FIFO is then immediately set again. For example, if a receive interrupt service routine doesn't read all the characters in the Rx FIFO, the RDRF flag remains asserted.

The Rx DMA request is disabled when any of the error flags PE or FE or OVRN are set, so that software can identify with which character the problem is associated.

If Bit 7, RDRF Interrupt Inhibit, is set to 1 (see Figures 32 and 33), the ASCI does not request a Receive interrupt when its RDRF flag is 1. Set this bit when programming a DMA channel that needs to receive data from an ASCI. The other causes for an ASCI Receive interrupt (PE, FE, OVRN, and for ASC2, CDR) continue to request Rx interrupt if the RIE bit is 1. (The RxDMA request is inhibited if PE or FE or OVRN are set, so that software can tell where an error occurred.) When this bit is 0, it is after a reset. RDRF will cause an ASCI interrupt if RIE is 1.

Programmable Reload Timer (PRT)

This logic consists of two separate channels, each containing a 16-bit counter (timer) and count reload register. The time base for the counters is derived from the system clock (divided by 20) before reaching the counter. PRT channel 1 provides an optional output to allow for waveform generation.

The T_out output of PRT1 is available on a multiplexed pin.

Clocked Serial I/O (CSIO)

These pins are multiplexed with the RTS, CTS, and clock pins for ASC0. Note: It is possible to use both ASC0 and the CSIO at the same time. If bit 4 of the System Configuration Register is set to 1, the CSK clock signal will internally drive the clock for ASC0 instead of the system clock.

The pins for this function are multiplexed with the RTS, CTS, and clock pins for ASC0. Note: It is possible to use both ASC0 and the CSIO at the same time. If bit 4 of the System Configuration Register is set to 1, the CSK clock signal will internally drive the clock for ASC0 instead of the system clock.
Zilog S MART P ERIPHERAL C ONTROLERS

Programmable Power-Down Modes

The Z8S180 is designed to save power. Two low-power, programmable power-down modes have been added: STANDBY mode and IDLE mode. The STANDBY/IDLE mode is selected by multiplexing D6 and D3 of the CPU Control Register (CCR, I/O Address = 1FH).

To enter STANDBY mode:

1. Set D6 and D3 to 1 and 0, respectively.
2. Set the I/O STOP bit (D5 of ICR, I/O Address = 3FH) to 1.
3. Execute the SLEEP instruction.

When the device is in STANDBY mode, it behaves similar to the SYSTEM STOP mode as it exists on the Z80180, except that the STANDBY mode stops the external oscillator, internal clocks and reduces power consumption to 50 μA (typical).

Since the clock oscillator has been stopped, a restart of the oscillator requires a period of time for stabilization. An 18-bit counter has been added in the Z8S180 to allow for oscillator stabilization. When the part receives an external IRQ or BUSREQ during STANDBY mode, the oscillator is restarted and the timer counts down 217 counts before acknowledgment is sent to the interrupt source. The recovery source needs to remain asserted for the duration of the 217 count, otherwise standby will be reentered.

The Z8S180 can exit STANDBY mode by asserting input /NMI. The STANDBY mode may also exit by asserting /INT0, /INT1 or /INT2, depending on the conditions specified in the following paragraphs.

If exit conditions are met, the internal counter provides time for the crystal oscillator to stabilize, before the internal clocking and the system clock output within the Z8S180 are resumed.

STANDBY Mode Exit with BUS REQUEST

Optional, if the BREXT bit (D5 of CPU Control Register) is set to 1, the Z8S180 exits STANDBY mode when the BUSREQ input is asserted; the crystal oscillator is then restarted. An internal counter automatically provides time for the oscillator to stabilize, before the internal clocking and the system clock output of the Z8S180 are resumed.

If exit conditions are met, the internal counter provides time for the crystal oscillator to stabilize, before the internal clocking and the system clock output within the Z8S180 are resumed.

Notes:

1. D6 and IDLE mode are only offered in the Z8S180. Note that the minimum recovery time can be achieved if INTERRUPT is used as the Recovery Source.

Add-On Features

There are five different power-down modes: SLEEP and SYSTEM STOP are inherited from the Z80180. In SLEEP Mode, the CPU is in a stopped state while the on-chip I/Os are still operating. In IDLE mode, the on-chip I/Os are in a stopped state while the CPU is running. In SYSTEM STOP mode, both the CPU and the on-chip I/Os are in the stopped state to reduce current consumption. The Z8S180 has added two additional power-down modes, STANDBY and IDLE, to reduce current consumption even further. The differences in these power-down modes are summarized in Table 2.

Notes:

1. IDLE and STANDBY modes are only offered in the Z8S180. Note that the minimum recovery time can be achieved if INTERRUPT is used as the Recovery Source.

17 counts before oscillator stabilization.
IDLE Mode

IDLE mode is another power-down mode offered by the Z8S180. To enter IDLE mode:

1. Set D6 and D3 to 0 and 1, respectively.
2. Set the I/O STOP bit (D5 of IOR, I/O Address = 3FH) to 1.
3. Execute the SLEEP instruction.

When the part is in IDLE mode, the clock oscillator is kept oscillating, but the clock to the rest of the internal circuit, including the CLKOUT, is stopped completely. IDLE mode is entered in a similar way as STANDBY mode, in other words, RESET, BUS REQUEST or EXTERNAL INTERRUPTS, except that the 217 bit wake-up timer is bypassed; all control signals are asserted eight clock cycles after the exit conditions are gathered.

Standby-Quick Recovery Mode

STANDBY-QUICK RECOVERY mode is an option offered in STANDBY mode to reduce the clock recovery time in STANDBY mode from 217 clock cycles (6.5 ms at 20 MHz) to 26 clock cycles (3.2 ms at 20 MHz). This feature can only be used when providing an oscillator as clock source.

To enter STANDBY-QUICK RECOVERY mode:

1. Set D6 and D3 to 1 and 1, respectively.
2. Set the I/O STOP bit (D5 of ICR, I/O Address = 3FH) to 1.
3. Execute the SLEEP instruction.

When the part is in STANDBY-QUICK RECOVERY mode, the operation is identical to STANDBY mode except when exit conditions are gathered, in other words, RESET, BUS REQUEST or EXTERNAL INTERRUPTS. The clock and other control signals are recovered sooner than the STANDBY mode.

Note: If STANDBY-QUICK RECOVERY is enabled, the user must make sure stable oscillation is obtained within 64 clock cycles.
get Figure 24. ASCI Control Register B (Ch. 0)

<table>
<thead>
<tr>
<th>SS, 2, 1, 0</th>
<th>PS = 0 (Divide Ratio = 10)</th>
<th>PS = 1 (Divide Ratio = 30)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>General Divide Ratio</td>
<td>DR = 0 (x16)</td>
</tr>
<tr>
<td>000</td>
<td>0 + 160</td>
<td>0 + 480</td>
</tr>
<tr>
<td>001</td>
<td>0 + 320</td>
<td>0 + 960</td>
</tr>
<tr>
<td>010</td>
<td>0 + 640</td>
<td>0 + 1920</td>
</tr>
<tr>
<td>011</td>
<td>0 + 1280</td>
<td>0 + 3840</td>
</tr>
<tr>
<td>100</td>
<td>0 + 2560</td>
<td>0 + 7680</td>
</tr>
<tr>
<td>101</td>
<td>0 + 5120</td>
<td>0 + 30720</td>
</tr>
<tr>
<td>110</td>
<td>0 + 10240</td>
<td>0 + 122880</td>
</tr>
<tr>
<td>111</td>
<td>External Clock (Frequency &lt; 0)</td>
<td></td>
</tr>
</tbody>
</table>

---

get Figure 25. ASCI Control Register B (Ch. 1)

<table>
<thead>
<tr>
<th>SS, 2, 1, 0</th>
<th>PS = 0 (Divide Ratio = 10)</th>
<th>PS = 1 (Divide Ratio = 30)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>General Divide Ratio</td>
<td>DR = 0 (x16)</td>
</tr>
<tr>
<td>000</td>
<td>0 + 160</td>
<td>0 + 480</td>
</tr>
<tr>
<td>001</td>
<td>0 + 320</td>
<td>0 + 960</td>
</tr>
<tr>
<td>010</td>
<td>0 + 640</td>
<td>0 + 1920</td>
</tr>
<tr>
<td>011</td>
<td>0 + 1280</td>
<td>0 + 3840</td>
</tr>
<tr>
<td>100</td>
<td>0 + 2560</td>
<td>0 + 7680</td>
</tr>
<tr>
<td>101</td>
<td>0 + 5120</td>
<td>0 + 30720</td>
</tr>
<tr>
<td>110</td>
<td>0 + 10240</td>
<td>0 + 122880</td>
</tr>
<tr>
<td>111</td>
<td>External Clock (Frequency &lt; 0)</td>
<td></td>
</tr>
</tbody>
</table>

---

*Note: The * † symbol indicates that /CTS depends on the condition of /CTS pin.*
Upon Reset

RDRF
OVRN /DCD0 TDRE TIE
0
R
0
R
0
R/W R/W
††
R
0
R/WR/W

S TAT 0
Transmit Interrupt Enable
Transmit Data Register Empty
Data Carrier Detect
Receive Interrupt Enable
Framing Error
Parity Error
Over Run Error
Receive Data Register Full

Figure 26. ASCI Status Register (Ch. 0)

Upon Reset

RDRF
OVRN /DCD0 TDRE TIE
0
R
0
R
0
R/W R/W
††
R
0
R/WR/W

S TAT 1
Transmit Interrupt Enable
Transmit Data Register Empty
Reserved
Receive Interrupt Enable
Framing Error
Parity Error
Over Run Error
Receive Data Register Full

Figure 27. ASCI Status Register (Ch. 1)

ASCI CHANNELS CONTROL REGISTERS (Continued)

Figure 28. ASCI Transmit Data Register (Ch. 0)

Figure 29. ASCI Transmit Data Register (Ch. 1)

Figure 30. ASCI Receive Data Register (Ch. 0)

Figure 31. ASCI Receive Data Register (Ch. 1)

Figure 32. ASCI0 Extension Control Register (I/O Address 12)

Figure 33. ASCI1 Extension Control Register (I/O Address 13)
ACSI TIME CONSTANT REGISTERS

New Z8S180 Registers

<table>
<thead>
<tr>
<th>Register</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCI0 Time Constant Low</td>
<td>1Ah</td>
</tr>
<tr>
<td>ASCI1 Time Constant Low</td>
<td>1Ch</td>
</tr>
<tr>
<td>ASCI0 Time Constant High</td>
<td>1Bh</td>
</tr>
<tr>
<td>ASCI1 Time Constant High</td>
<td>1Dh</td>
</tr>
</tbody>
</table>

CSI/O REGISTERS

<table>
<thead>
<tr>
<th>Bit</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>EF</td>
</tr>
<tr>
<td>1</td>
<td>BE</td>
</tr>
<tr>
<td>2</td>
<td>RE</td>
</tr>
<tr>
<td>3</td>
<td>TE</td>
</tr>
<tr>
<td>4-7</td>
<td>SS2, SS1, SS0</td>
</tr>
</tbody>
</table>

- **CNTR**
  - **EF**: Enable Transmit
  - **BE**: Enable Receive
  - **RE**: Read Enable
  - **TE**: Transmit Enable
  - **SS2, SS1, SS0**: Speed Select

- **R/W**: Read/Write
- **0AH**: Upon Reset

<table>
<thead>
<tr>
<th>Baud Rate</th>
<th>Speed Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>000 Ø + 20</td>
<td></td>
</tr>
<tr>
<td>001 Ø + 40</td>
<td></td>
</tr>
<tr>
<td>010 Ø + 80</td>
<td></td>
</tr>
<tr>
<td>011 Ø + 100</td>
<td></td>
</tr>
</tbody>
</table>

- **TRDR**: Read/Write, Addr 0BH
  - Read - Received Data
  - Write - Transmit Data

- **RLDR0L**: Read/Write, Addr 0EH
  - When Read, read Data Register L before reading Data Register H.

- **RLDR0H**: Read/Write, Addr 0FH
  - When Read, read Data Register L before reading Data Register H.

- **RLDR1L**: Read/Write, Addr 16H
  - When Read, read Data Register L before reading Data Register H.

- **RLDR1H**: Read/Write, Addr 17H
  - When Read, read Data Register L before reading Data Register H.
**Timer Control Register**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCR</td>
<td>Timer Down Count Enable 1,0</td>
</tr>
<tr>
<td>TOC1</td>
<td>Timer Output Control 1,0</td>
</tr>
<tr>
<td>TIE0</td>
<td>Timer Interrupt Enable 1,0</td>
</tr>
</tbody>
</table>

**FREE RUNNING COUNTER**

**CPU Control Register**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>ext (TOUT/DREQ)</td>
</tr>
<tr>
<td>ASCI0 Rx</td>
<td>ASCI1 Rx</td>
</tr>
<tr>
<td>ESCC Rx</td>
<td>PIA27-20 IN</td>
</tr>
</tbody>
</table>

**DMA Registers**

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAR0L</td>
<td>SAR0H</td>
</tr>
<tr>
<td>DAR0L</td>
<td>DAR0H</td>
</tr>
</tbody>
</table>

**DMA 0 Source Address Registers**

<table>
<thead>
<tr>
<th>Source</th>
<th>SAR0B</th>
</tr>
</thead>
<tbody>
<tr>
<td>ext (TOUT/DREQ)</td>
<td>1100</td>
</tr>
<tr>
<td>ASCI0 Rx</td>
<td>1100</td>
</tr>
<tr>
<td>ASCI1 Rx</td>
<td>1101</td>
</tr>
<tr>
<td>ESCC Rx</td>
<td>1111</td>
</tr>
<tr>
<td>PIA27-20 IN</td>
<td>1111</td>
</tr>
</tbody>
</table>

**DMA 0 Destination Address Registers**

<table>
<thead>
<tr>
<th>Destination</th>
<th>DAR0B</th>
</tr>
</thead>
<tbody>
<tr>
<td>ext (TOUT/DREQ)</td>
<td>1100</td>
</tr>
<tr>
<td>ASCI0 Tx</td>
<td>1100</td>
</tr>
<tr>
<td>ASCI1 Tx</td>
<td>1101</td>
</tr>
<tr>
<td>ESCC Tx</td>
<td>1111</td>
</tr>
<tr>
<td>PIA27-20 OUT</td>
<td>1111</td>
</tr>
</tbody>
</table>

---

*Note: See Figure 87 for full description.*
DMA Register Description

Bit 7. This bit should be set to 1 only when both DMA channels are set to take their requests from the same device. If this bit is 1 (it resets to 0), the channel end output of DMA channel 0 sets a flip-flop, so that thereafter the device's request is visible to channel 1, but not visible to channel 0. The channel end output of channel 1 clears the FF, so that thereafter, the device's request is visible to channel 0, but not visible to channel 1.

Bit 6. When both DMA channels are programmed to take their requests from the same device, this bit (FF mentioned in the previous paragraph) controls which channel the device's request is presented to: 0 = DMA 0, 1 = channel 1. When bit 7 is 1, this bit is automatically toggled by the channel end output of the channels, as described above.

Bits 5-4. Reserved and should be programmed as 0.

Bits 3. This bit controls the direction and use of the TOUT/DREQ pin. When it's 0, TOUT/DREQ is the DREQ input; when it's 1, TOUT/DREQ is an output that can carry the TOUT signal from PRT1, if PRT1 is so programmed.

Bits 2-0. With "DIM1", bit 1 of DCNTL, these bits control which request is presented to DMA channel 1, as follows:

0 000 ext TOUT/DREQ
0 001 ASCI0 Rx
0 010 ASCI1 Rx
0 011 ESCC1
0 10X Reserved, do not program.
0 111 PIA27-20 in
1 000 ext TOUT/DREQ
1 001 ASCI0 Tx
1 010 ASCI1 Tx
1 011 ESCC out
1 10X Reserved, do not program.
1 111 PIA27-20 out
DMA REGISTERS (Continued)

![DMA Mode Registers Diagram](image)

**Figure 55. DMA Mode Registers**

- **DMA Modes**
  - **DM1, 0**
    - 00: M - DMA1
    - 01: M - DMA0
    - 10: M - DMA0 Fixed
    - 11: I/O - DMA0 Fixed

- **Memory MODE Select**
  - Ch 0 Source Mode 1, 0
  - Ch 0 Destination Mode 1, 0

- **Address**
  - Bit 1: 0
  - Bit 0: R/W

**Figure 56. DMA/WAIT Control Register**

- **DCNTL**
  - **DMS0**
    - 00: M - I/O
    - 01: M - I/O
    - 10: I/O - M
    - 11: I/O - M

- **DM1, 0**
  - 00: M - I/O
  - 01: M - I/O
  - 10: I/O - M
  - 11: I/O - M

- **MWR1**
  - **IWI0**
    - 00: Select, i = 1, 0
    - 01: I/O Wait Insertion
    - 10: Memory Wait Insertion

- **DM1, 0**
  - 00: M - I/O
  - 01: M - I/O
  - 10: I/O - M
  - 11: I/O - M

- **MAR1**
  - **MAR1-1**
    - 00: M - I/O
    - 01: M - I/O
    - 10: I/O - M
    - 11: I/O - M

**Note:**
- If using the Wait-State Generators provided in register D8, the MWR1-0 bit should be set to 00.
### System Control Registers

**Interrupt Vector Low Register**

- **Address:** 33H
- **Bits:**
  - IL7
  - IL6
  - IL5
  - IL4
  - IL3
  - IL2
  - IL1
- **Reset Values:**
  - IL7: 0
  - IL6: 0
  - IL5: 0
  - IL4: 0
  - IL3: 0
  - IL2: 0
  - IL1: 0
- **Read/Write:** Read/Write

**INT/Trap Control Register**

- **Address:** 34H
- **Bits:**
  - TRAP
  - UFO
  - ITE2
  - ITE1
  - ITE0
- **Reset Values:**
  - TRAP: 0
  - UFO: 0
  - ITE2: 1
  - ITE1: 1
  - ITE0: 0
- **Read/Write:** Read/Write

**Refresh Control Register**

- **Address:** 36H
- **Bits:**
  - CYC1
  - CYC0
  - CYC1
  - CYC0
- **Reset Values:**
  - CYC1: 1
  - CYC0: 1
- **Read/Write:** Read/Write

**MMU Registers**

**MMU Common Base Register**

- **Address:** 38H
- **Bits:**
  - CB7
  - CB6
  - CB5
  - CB4
  - CB3
  - CB2
  - CB1
  - CB0
- **Reset Values:**
  - CB7: 0
  - CB6: 0
  - CB5: 0
  - CB4: 0
  - CB3: 0
  - CB2: 0
  - CB1: 0
  - CB0: 0
- **Read/Write:** Read/Write

**MMU Bank Base Register**

- **Address:** 39H
- **Bits:**
  - BB7
  - BB6
  - BB5
  - BB4
  - BB3
  - BB2
  - BB1
  - BB0
- **Reset Values:**
  - BB7: 0
  - BB6: 0
  - BB5: 0
  - BB4: 0
  - BB3: 0
  - BB2: 0
  - BB1: 0
  - BB0: 0
- **Read/Write:** Read/Write

**MMU Bank Area Register**

- **Address:** 3AH
- **Bits:**
  - BA3
  - BA2
  - BA1
  - BA0
- **Reset Values:**
  - BA3: 1
  - BA2: 1
  - BA1: 1
  - BA0: 0
- **Read/Write:** Read/Write

**MMU Common Area Register**

- **Address:** 3BH
- **Bits:**
  - CB3
  - CB4
  - CB5
- **Reset Values:**
  - CB3: 0
  - CB4: 0
  - CB5: 0
- **Read/Write:** Read/Write

**Figure 57. Interrupt Vector Low Register**

**Figure 58. INT/Trap Control Register**

**Figure 59. Refresh Control Register**

**Figure 60. MMU Common Base Register**

**Figure 61. MMU Bank Base Register**

**Figure 62. MMU Common/Bank Area Register**
CPU CONTROL REGISTER

The CPU Control Register allows the programmer to select options that directly affect the CPU performance as well as controlling the STANDBY operating mode of the chip. The CPU Control Register (CCR) allows the programmer to change the divide-by-two internal clock to divide-by-one.

In addition, applications where EMI noise is a problem, the Z8S180 can reduce the output drivers on selected groups of pins to 33 percent of normal pad driver capability which minimizes the EMI noise generated by the part (Figure 65).

System Control Registers

CPU Control Register (CCR) Addr 1FH

- Clock Divide
  0 = XTAL/2
  1 = XTAL/1

- Standby/Idle Enable
  00 = No Standby
  01 = Idle After Sleep
  10 = Standby After Sleep
  11 = Standby After Sleep
  64 Cycle Exit (Quick Recovery)

- BREXT
  0 = Ignore BUSREQ
  1 = Standby/Idle Exit on BUSREQ

- LNAD/DA
  0 = Standard Drive
  1 = 33% Drive On CPU
- LNCPUCTL
  0 = Standard Drive
  1 = 33% Drive On CPU
- LNPHI
  0 = Standard Drive
  1 = 33% Drive On EXT PH Clock

I/O Control Register

Figure 63. Operation Mode Control Register

Figure 64. I/O Control Register

Figure 65. CPU Control Register
Bit 7. Clock Divide Select. Bit 7 of the CCR allows the programmer to set the internal clock to divide the external clock by two if the bit is 0 and divide-by-one if the bit is 1. Upon reset, this bit is set to 0 and the part is in divide-by-two mode. Since the on-board oscillator is not guaranteed to operate above 20 MHz, an external source must be used to achieve the maximum 33 MHz operation of the device, such as an external clock at 66 MHz with 50 percent duty cycle.

If an external oscillator is used in divide-by-one mode, the minimum pulse width requirement must be satisfied.

Bits 6 and 3. STANDBY/IDLE Enable. These two bits are used for enabling/disabling the IDLE and STANDBY mode.

Setting D6, D3 to 0 and 1, respectively, enables the IDLE mode. In the IDLE mode, the clock oscillator is kept oscillating but the clock to the rest of the internal circuitry, including the CLKOUT, is stopped. The Z8S180 enters IDLE mode after fetching the second opcode of a SLEEP instruction, if the I/O STOP bit is set.

Setting D6, D3 to 1 and 0, respectively, enables the STANDBY mode. In the STANDBY mode, the clock oscillator is stopped completely. The Z8S180 enters STANDBY after fetching the second opcode of a SLEEP instruction, if the I/O STOP bit is set.

Setting D6, D3 to 1 and 1, respectively, enables the STANDBY–QUIK RECOVERY mode. In this mode, its operations are identical to STANDBY except that the clock recovery is reduced to 64 clock cycles after the exit conditions are gathered. Similarly, in STANDBY mode, the Z8S180 enters IDLE mode after fetching the second opcode of a SLEEP instruction, if the I/O STOP bit is set.

Bit 5. BREXT. This bit controls the ability of the Z8S180 to honor a bus request during STANDBY mode. If this bit is set to 1 and the part is in STANDBY mode, a BUSREQ is honored after the clock stabilization timer is timed out.

Bit 4. LNPHI. This bit controls the drive capability on the PHI Clock output. If this bit is set to 1, the PHI Clock output is reduced to 33 percent of its drive capability.

Bit 12. LMOV. This bit controls the drive capability of certain external I/O pins on the Z8S180. When this bit is set to 0, the output drive capability of the following pins is reduced to 33 percent of the original drive capability:

- RTS0/TXS
- TXA0
- CKA1
- TXA1
- CKA0
- TOUT

Bit 11. LNCPUCTL. This bit controls the drive capability of the CPU Control pins. When this bit is set to 1, the output drive capability of the following pins is reduced to 33 percent of the original drive capability:

- BUSACK
- /IORQ
- /RD
- /RFSH
- /WR
- /HALT
- /M1
- /MREQ
- /WAIT

Bit 10. LNA/DHATA. This bit controls the drive capability of the Address/Data bus output drivers. If this bit is set to 1, the output drive capability of the Address and Data bus output is reduced to 33 percent of its original drive capability.

Significant features of the EMSCC include:

- Hardware and software compatible with Zilog’s SCC/ESCC
- 4-Byte Transmit FIFO
- 8-Byte Receive FIFO
- Programmable FIFO Interrupt Levels Provide Flexible Interrupt Response
- Improved SDLC Frame Status FIFO
- New Programmable Features Added with Write Register 7

In addition, the following features have been added to the EMSCC channel in the Z80185:

- Programmable LocalTalk feature
- Non-Multiplexed /DTR Pin
- Internal Connection of DMA Request and /WAIT Signals
- EMSCC Programmable Clock
  - Programmed to be Equal to System Clock Divided by One or Two
  - Programmed by System Configuration Register

Note: The EMSCC programmable clock must be programmed to divide-by-two mode when operating above the following condition: PHI > 20 MHz at 5.0V.
The Z80185 features a one-channel EMSCC that uses two I/O addresses:

- EMSCC Channel A Control I/O Address %E8
- Data I/O Address %E9

Divide-by-two should be programmed when operating the Z80185 beyond 20 MHz, 5V.

**Initialization.** The system program first issues a series of commands to initialize the basic mode of operation. This is followed by other commands to qualify conditions within the selected mode. For example, in the Asynchronous mode, character length, clock rate, number of stop bits, and even or odd parity should be set first. Then the interrupt mode is set, and finally, the receiver and transmitter are enabled.

**Write Registers.** The EMSCC contains 16 write registers (17 counting the transmit buffer) in each channel. These write registers are programmed separately to configure the functional "personality" of the channels. A new register, WR7', was added to the EMSCC and may be written to if WR15, D0 is set. Figure 50 shows the format of each write register.

**Read Registers.** The EMSCC contains ten read registers (11 counting the receive buffer) in each channel. Four of these may be read to obtain status information (RR0, RR1, RR10, and RR15). Two registers (RR12 and RR13) are read to learn the baud rate generator time constant. RR2 contains the unmodified interrupt vector (channel A) or the vector modified by status information (channel B). RR3 contains the Interrupt Pending (IP) bits (channel A only). RR6 and RR7 contain the information in the SDLG Frame Status FIFO, but is only read when WR15 D2 is set. If WR7' D6 is set, Write Registers WR3, WR4, WR5, WR7', and WR10 can be read as RR9, RR4, RR5, and RR14, respectively. Figure 51 shows the format of each read register.

With the Z80185, the EMSCC channel’s DTR, Tx and Rx DMA Request and WAIT outputs are not subject to multiplexing and are routed separately to the CPU and pins.

In other words,
1. the DTR pin is not multiplexed and always follows WR5 bit 7;
2. if WR1 bits 7-6 are 10, and the processor reads the RDR when the RxFIFO is empty, or writes the TDR when the TxFIFO is full, the processor is "waited" until a character arrives or has been sent out;
3. WR1 bit 5 has no effect;
4. WR14 bit 2 should be kept 0;
5. WR1 bits 7-6 should not be programmed as 11.

**EMSCC**

**Transmit Logic**
- Transmit FIFO
- Data Encoding & CRC Generation
- Transmit MUX
- Transmit/Receive Clock Multiplexer
- Digital Phase-Locked Loop
- Crystal Oscillator
- Amplifier

**Receive Logic**
- Receive MUX
- CRC Checker, Data Decode & Synchronization Detection
- Receiver Status FIFO
- Receiver Data FIFO
- SDLC Frame Status FIFO

**Modem/Control Logic**
- Modem/Control Logic
- Baud Rate Generator
- Digital Phase-Locked Loop
- Crystal Oscillator
- Amplifier

**Interrupt Control Logic**
- CPU DMA Bus Interface
- Interrupt Control

**Channel A**
- TX/RX
- Channel A Register

**Figure 66. EMSCC Block Diagram**
A LocalTalk feature has been added in one EMSCC of the Z80185, operating as follows:

If a certain set of register bits are set, RTS acts as a LocalTalk Driver Enable output that operates as shown in Figure 50. All of the following bits and fields must be programmed exactly as shown to enable this mode:

- WR4.3-2 = 00: sync modes
- WR4.5-4 = 10: SDLC
- WR5.1 = 0: no RTS
- WR10.3 = 1: auto RTS deactivation
- WR6.4 = 1: Send Break

When the first five conditions above are set (as for LocalTalk operation), the WR6.4 bit is used as a Select LocalTalk Driver enable bit, rather than the Send Break command bit used in async mode.

Setting these register bits in this manner configures the EMSCC Transmitter to send three Flags before a frame, negating RTS during the first to create a coding violation, when software writes the first character of a frame to the TDR and TxRIFO. This mode also makes the Transmitter ensure at least 16 bits of idle time between a closing Flag and the end of frame interval. The RTS output is driven active for one bit time at the start of the first of the three flags, then inactive for four bit times, then active again for the duration of the opening flags, the frame, and closing flag plus 16 bit times thereafter.

There is one other difference in EMSCC operation when this new mode is enabled. The setting of the TxIP bit, that normally occurs after the last bit of the CRC is sent, is delayed until the 16-bit idle is sent and RTS is negated.

Figure 67. EMS CC Transmitter Flag Commands

Figure 68. Write Register Bit Functions
Figure 69. Write Register Bit Functions (Continued)
Figure 71. Write Register Bit Functions (Continued)

Figure 72. Read Register Bit Functions
The Centronics P1284 Controller can operate in either the Host or Peripheral role in Compatibility mode (host to printer), Nibble or Byte mode (printer to host), and ECP mode (bidirectional). It provides no hardware support for the EPP mode, although it may be possible to implement this mode by software.

Nine control signals have dedicated hardware pins, and have a 12 mA drive (P1284 Level 2) capability as does the 8-bit data port PIA27-20. Note: Signal names listed below are those for the original Compatible mode. The names shown in parentheses represent the same signal, but in a more recent mode. The Z80185 does not include hardware support for the P1284 EPP mode.

The following signals are outputs in a Peripheral mode, inputs in a Host mode:

- Busy (PtrBusy, PeriphAck)
- nAck (PtrClk, PeriphClk)
- nPError (AckDataReq, nAckReverse)
- nFault (nDataAvail, nPeriphRequest)
- Select (Xflag)

The following signals are inputs in a Peripheral mode, outputs in a Host mode:

- nStrobe (HostClk)
- nAutoFd (HostBusy, HostAck)
- nSelectIn (P1284Active)
- nInit (nReverseRequest)

Note that, because the Host/Peripheral mode is fully controlled by software, a Z80185-based product can operate as a Host in one system, or as a Peripheral in another, without any change to the hardware. A Z80185-based product could even act as a Host at one time and a Peripheral at another time within the same system, if there is a mechanism to control such alternate use.

In general, the interface architecture automates operations that are seen as performance-critical, while leaving less frequent operations to software control. To achieve top performance, software should assign a DMA channel to the current direction of dataflow.

Note: The IEEE 1284 Interface should be used with the /IOC bit (bit D5) in the OMCR set to 0. The setting of this bit primarily affects RLE expansion in peripheral ECP forward and host ECP reverse modes.
Bidirectional Centronics Registers

Reading the Parallel Controls (PARC) register allows software to sense the state of the input signals per the current mode, plus two or three status flags:

<table>
<thead>
<tr>
<th>Busy</th>
<th>PError</th>
<th>Select</th>
<th>nFault</th>
<th>nAck</th>
<th>IIKop</th>
<th>DREQ</th>
<th>Idle</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 74a. Reading PARC in a Host Mode**

(I/O Address %DA)

The controller sets IIKop (Illegal Operation) when it detects an error in the protocol, for example, if it’s in Peripheral mode and it detects that the host has driven P1284A drive (nSelect) Low at a time that mandates an immediate Abort, that is, outside one of the “windows” in which this event indicates an organized disengagement. If “status interrupts” are enabled, such an interrupt is always requested when IIKop is set. Writing PARM with NewMode=1 clears IIKop.

DREQ is the Request presented to the DMA channels, which may or may not be programmed to service this request. If not, an interrupt can be enabled when DREQ is set.

**Figure 74b. Reading PARC in a Peripheral Mode**

(I/O Address %DA)

Writing to PARC allows the software to set and clear the output signals per the current mode:

<table>
<thead>
<tr>
<th>nAutoFd</th>
<th>nStrobe</th>
<th>nSelctIn</th>
<th>nInit</th>
<th>nAck</th>
<th>nFault</th>
<th>Busy</th>
<th>PError</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 75a. Writing to PARC in a Host Mode**

(I/O Address %DA)

NewMode = 1 reinitializes the state machine to the initial state for the mode called out by MODE. Never change MODE without writing a 1 in this bit.

IdleIE = 1 enables interrupts when the controller sets the Idle flag. When software uses a DMA channel to provide data to the P1284 controller, it can be expected that the channel will do so in a timely manner, and thus, that an Idle condition signifies that the channel has finished transferring the block. (Software can also enable an interrupt from the DMA channel, but on the transmit side, such interrupts are not well-synchronized to events on the P1284 controller.) Conversely, if software provides data, Idle may not be grounds for an interrupt.

StatIE = 1 enables “status” interrupts that are described separately for each mode.

DREQIE = 1 enables interrupts when the controller sets DREQ, except that in those modes that set DREQ when they are entered, such setting doesn’t request an interrupt.

**Table 3. Bidirectional Centronics Mode Selection**

<table>
<thead>
<tr>
<th>MODE</th>
<th>Non-P1284 mode</th>
<th>Peripheral Compatible/Negotiation mode</th>
<th>Peripheral Nibble mode</th>
<th>Peripheral Byte mode</th>
<th>Peripheral ECP Reverse mode</th>
<th>Peripheral Inactive mode</th>
<th>Peripheral ECP Forward mode with software RLE handling</th>
<th>Peripheral ECP Forward mode with hardware RLE expansion</th>
<th>Host Negotiation mode</th>
<th>Host Compatible mode</th>
<th>Host Nibble mode</th>
<th>Host Byte mode</th>
<th>Host ECP Forward mode</th>
<th>Host ECP Reverse mode with software RLE handling</th>
<th>Host ECP Reverse mode with hardware RLE expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0010</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0100</td>
<td>Peripheral ECP Reverse mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0101</td>
<td>Peripheral Inactive mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0110</td>
<td>Peripheral ECP Forward mode with software RLE handling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0111</td>
<td>Peripheral ECP Forward mode with hardware RLE expansion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>Host Negotiation mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1001</td>
<td>Host Compatible mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td>Host Nibble mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1011</td>
<td>Host Byte mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1100</td>
<td>Host ECP Forward mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1101</td>
<td>Host Reserved mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td>Host ECP Reverse mode with software RLE handling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1111</td>
<td>Host ECP Reverse mode with hardware RLE expansion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A second output register has been added for PIA27-20. Writing to either the Z80181-compatible PIA 2 Data Register (address E3) or the new Alternate PIA 2 Data Register (address E6) write to the Output Holding Register (OHR). When the PIA27-20 pins are outputs, the outputs of the OHR are the inputs to the second register, which is called the I/O register (IOR), these outputs drive the PIA27-20 pins. When the pins are inputs, they are the inputs to the IOR, which can be read from the PIA 2 Data Register (address E3).

In non-P1284 mode, Host Negotiation mode, Reserved Modes, and in Peripheral Compatible/Negotiation mode when the host drives nSelectIn (P1284Active) High to select negotiation, the direction of the PIA27-20 pins are controlled by the PIA 2 Data Direction register, as on the Z80181. Also in these modes the IOR is loaded on every PHI clock, so that operation is virtually identical to the Z80181. In other modes the controller controls the direction of PIA27-20 and when the IOR is loaded.

A Time Constant Register PART must be loaded by software with the smallest number of PHI clocks that equals or exceeds the “critical time” for the mode selected in PARM. The critical time is 750 ns for Host Compatible mode, 500 ns for most other modes, and the time necessary to indicate DMA completion in Host ECP Forward and Peripheral ECP Reverse modes.

Reading PART yields the status of the IP and IUS bits, which are described in the Bidirectional Centronics Interface section:

<table>
<thead>
<tr>
<th>IUS</th>
<th>IP</th>
<th>number of PHI clocks in critical time</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 78. PART Write (I/O Address %DC)**

The Vector Register PARV must be loaded by software with the interrupt vector to be used for interrupts from this controller.

<table>
<thead>
<tr>
<th>Interrupt Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
</tr>
</tbody>
</table>

**Figure 80. PARV (I/O Address %DD)**
Interrupts

As in other Zilog peripherals, the controller includes an interrupt pending bit (IP), and an interrupt under service bit (IUS). The controller is part of an on-chip interrupt acknowledge daisy-chain that extends from the E1 pin, through the EMCCC, CTC, and this controller in a programmable priority order, and from the lowest-priority of these devices to the E0 pin. The interrupt request from the controller is logically OR'd with INT0 and other on-chip interrupt requests to the processor.

The controller sets its IP bit whenever any of the conditions occurs:
1. PARM4 is 1, and the controller sets the DREQ bit. This does not include when the controller forces the DREQ bit to 1, when software first places the controller in Peripheral Nibble, Peripheral Byte, Peripheral ECP Reverse, Host Compatible, or Host ECP Forward mode.
2. PARM5 is 1, and a mode-dependent “status interrupt” condition occurs. The following sections describe the status interrupt conditions (if any) for each mode.
3. PARM6 is 1, and the controller sets the Idle bit, except when the controller forces the Idle bit to 1, when software first places the controller in Peripheral Nibble, Peripheral Byte, Peripheral ECP Reverse, Host Compatible, or Host ECP Forward mode. The following sections describe when idle is set in each mode.

Once IP is set, it remains set until software writes a 1 to PART6.

The controller will begin requesting an interrupt of the processor whenever IP is set, its IEI signal from the on-chip daisy-chain is High, and its IUS bit is 0. Once it starts requesting an interrupt, the controller will continue to do so until IEO goes Low in an interrupt acknowledge cycle, or IP is 0, or IUS is 1.

The controller drives its IEO output High, if its IEI input is High, and its IP and IUS bits are both 0. A Z80 interrupt acknowledge cycle is signalled by IM1 going Low, followed by IRO going Low. The controller, and all other devices in the daisy-chain, freeze the transfer of their IP bits to the OE outputs while IM1 is Low, which prevents new events from affecting the daisy-chain. By the time IRO goes Low, one and only one device will have its IEI pin High, and its IEO pin Low. —This device responds to the interrupt by providing an interrupt vector, and setting its IUS bit. This controller also clears its IP bit when it responds to an interrupt acknowledge cycle.

Non-P1284 Mode

The Z80185 defaults to this mode after a Reset, and this mode is compatible with the use of PA27-20 on the Z80181. The directions of PA27-20 can be controlled individually by writing to register E2, as on the Z80181. The state of outputs PA27-20 can be set by writing to register E3, and the state of eight pins can be sensed by reading register E3. The Busy, nAck, PError, nFault, and Select pins are tri-stated in this mode, while nStrobe, nAutoFd, nSelect, and nStrobe are inputs. There are no status interrupts in this mode.

Peripheral Inactive Mode

This mode operates identically to Non-P1284 mode as described above, except that the Busy, nAck, PError, nFault, and Select pins are outputs that can be controlled via the PARC and PARC2 registers, and status interrupts can occur in response to any event on nAutoFd, nStrobe, nSelect, or nStrobe. This mode differs from Peripheral Compatibility/Negotiation mode with nSelect (P1284 Active) High, only in that the controller will not operate in Compatibility mode if nSelect goes Low.
In this mode, software should monitor for the condition P1284Active(nSelectIn) High, and nAutoFd Low simulta-
neously. If software detects this state, it should initiate a Negotiation process. Software should read the value on PIA27-30 and set PErr, nFault, Xflag, and nAck as appropriate for the data value. As long as P1284active(nSelectIn) remains High in this mode, software is in complete control of the controller. After the host has driven nSrobe Low and then High again for an acceptable value, software should reprogram the MODE field to the appropri-
ate one of the following Peripheral modes.

Status interrupts in this mode include rising and falling edges on P1284active(nSelectIn) and ninit, and rising and falling edges on HostBusy(nAutoFd) and HostClk(nStrobe) while P1284Active(nSelectIn) is High.

Host Nibble Mode

1. If, during Host Negotiation mode, software has placed the value 00 or 04 on the data lines, and received a positive response on Xflag (Select) and a Low on nDataAvail(nFault) at a rising edge of PtrClk(nAck), then after optionally programming a DMA channel to store data, it should set this mode.

2. For each byte in this mode, the controller drives HostBusy (nAutoFd) Low and waits until DREQ is cleared, indicating that the CPU or DMA has taken any previous data, and the peripheral has driven PrpClk(nAck) Low. At this point it samples the other four status lines from the peripheral into the less-significant four bits of the Input/ Output Register as follows:

<table>
<thead>
<tr>
<th>Signal</th>
<th>First Data Bit</th>
<th>Second Data Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Busy</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>PError</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>nSelectIn</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>nFault</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

3. If nDataAvail(nFault) is High at a rising edge of nAck in this mode, indicating that the peripheral has no more data, the controller sets Idle and waits for software to program it back to Host Negotiation mode. Software can then select the next mode (reference IEEE P1284 specification).

If host software is programmed not to select all the data that a peripheral has available, it should first disable the DMA channel, if one is in use, then wait for DREQ to be 1 and PrpClk(nAck) to be High. If nDataAvail(nFault) is Low at this point, the controller will have already driven HostBusy(nAutoFd) Low to solicite the next byte. Software should then program the controller back to Host Negotiation mode, read the IOR to get the current byte, and take the next byte from the peripheral under software control. After the peripheral drives nAck High after the second nibble, software can drive P1284Active(nSelectIn) Low to tell the peripheral to leave Nibble mode.

There are no status interrupts in Host Nibble mode.

Peripheral Nibble Mode

1. Software shouldn’t set this mode until there is reverse data available to send. In other words, it should implement the P1284 “reverse idle mode” via software in Peripheral Compatibility/Negotiation mode. After software has driven nDataAvail(nFault), Ack/DataReq(PError), and Xflag (Select) all Low to signify that data is available, the driven PrpClk(nAck) High after 500 ns, and if requested programmed a DMA channel to pro-
vide data to send, when it sees HostBusy(nAutoFd) Low to request data, software should set this mode. Setting this mode sets DREQ and Idle, and these settings do not request an interrupt. The PIA27-30 pins remain configured for data input but are not used. Instead, four of the five control outputs are driven with the LS and MS four bits of the Input/Output Register, as shown in Table 2, while PrpClk(nAck) serves as a handshake/clock output. On entering this mode the hardware begins routing bits 3-0 of the IOR to these lines.

2. If software, or a DMA channel, writes a byte to the Holding Register while Idle is set, the controller immediately transfers the byte to the IOR and clears Idle, and regates DREQ only momentarily to request another byte from software or the DMA channel. After data has been valid on the four control outputs for 500 ns (as controlled by the P1284Active(nSelectIn) Low), the controller drives HostBusy(nAutoFd) Low to indicate readiness for a byte from the peripheral. Then it waits for the host to drive the HostBusy(nAutoFd) line back to High, switches the four control lines to bits 7-4 of the IOR, and begins waiting for the host to drive HostBusy (nAutoFd) back to Low. When bits 7-4 have been valid for 500 ns, the host has driven HostBusy (nAutoFd) Low, the controller drives PrpClk(nAck) Low again and begins waiting for the host to drive HostBusy (nAutoFd) High. When HostBusy (nAutoFd) High has been driven High, the controller returns the four control outputs to the state set by software in PARC. At this point, if software or a DMA channel has not yet sent another byte to the Output Holding Register (thus clearing DREQ), the controller sets Idle and waits for software to do so. If software or a DMA channel has written another byte to the OHIR, the controller transfers the byte to the IOR, sets DREQ, and clears Idle if it has been set. When the control outputs have been valid for 500 ns, the controller drives PrpClk(nAck) to High. Then it waits for the host to drive HostBusy(nAutoFd) back to Low, at which time it switches the four control lines back to bits 3-0 of the IOR and returns to the event sequence at the start of this paragraph.

If there is no more data to send, when the controller sets Idle, software should modify PARC to make nDataAvail(nFault) and Ack/DataReq(PError) High, and then change the mode to Peripheral Compatible/Negotiation mode. Then (after 500 ns), software should set PrpClk(nAck) back to High in PARC and enter Reverse Idle state.

Status interrupts in Peripheral Nibble mode include rising and falling edges on P1284Active(nSelectIn) and nInit. The controller sets the IllOp (Illegal Operation) bit if P1284Active(nSelectIn) goes Low in this mode, before it drives nAck High for the status states on the four control lines, or after the host drives HostBusy Low thereafter, in which case software should immediately enter Peripheral Compatibility/Negotiation mode. If P1284Active goes Low, but IllOp stays 0, indicating that the host negated P1284Active in a legitimate manner, software should enter Peripheral Inactive mode for the duration of the “return to Compatibility mode”, and then enter Peripheral Compatibility/Negotiation mode.

Host Byte Mode

1. When in Host Negotiation mode the software has pre-
sented the value hex 01 or 05 on PIA27-20, it has been acknowledged by the peripheral, and the peripheral has driven DataAvail(nFault) and Ack/DataReq(PError) to Low to indicate data availability and then driven PrpClk(nAck) back to High, software should set this mode. This sets PIA27-20 as inputs regardless of the contents of register E3, and clears the Idle flag. The controller then waits 500 ns (as controlled by the PART register) before proceeding.

2. For each byte, the controller drives HostBusy(nAutoFd) Low to indicate readiness for a byte from the peripheral. Then it waits for PrpClk(nAck) to go Low, at which time it captures the state of PIA27-20 into the Input/Output Register, sets the DREQ bit to request software, or the DMA channel to take the byte, and drives HostBusy(nAutoFd) High and HostClk(nStrobe) Low. When software, or the DMA channel, has taken the byte (thus clearing DREQ) and the peripheral has driven PrpClk(nAck) back High, and at least 500 ns after driving HostClk(nStrobe) Low, the controller drives HostClk(nStrobe) back to High, and samples DataAvail(nFault). If it is still Low, the controller returns to the event sequence at the start of this paragraph, otherwise it sets the Idle flag.

Table 4. Nibble Mode Bit Assignments

<table>
<thead>
<tr>
<th>Signal</th>
<th>First Data Bit</th>
<th>Second Data Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Busy</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>PError</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>nSelectIn</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>nFault</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>
In response to Idle, software should enter Host Negotiation mode. Thereafter, it can set HostBusy (nAutoFd) Low, to enter Reverse Idle state, or enter Host Compatible mode (reference IEEE P1284 specification), or conduct a new negotiation.

If software is programmed not to accept all the data that a peripheral has available in this mode, it should first disable the DMA channel, if one is in use, and then wait for DREQ to be 1 and nAck to be 0. Then it should reprogram the controller back to Host Negotiation mode, read the last byte from the IOR, drive HostClk (nStrobe) back to High, and then drive P1284Active (nSelectIn) Low to instruct the peripheral to leave Byte mode.

There are no status interrupts in Host Byte mode.

Peripheral Byte Mode

1. Software should not set this mode until there is reverse data available to send — that is, it should implement the P1284 “reverse idle mode” via software in Peripheral Compatibility/Negotiation mode. The exact sequencing among PtrClk (nAck), nDataAvail (nFault), and AckDataReq (nError) differs according to whether this mode is entered directly from Negotiation or from reverse Idle phase, and is controlled by software. But in either case, before software sets this mode, it should set nDataAvail (nFault) and AckDataReq (nError) to Low, then after 500 ns, set PtrClk (nAck) to High. When it detects that the host has driven HostBusy (nAutoFd) Low to request data, software should set this mode, which sets the DREQ and Idle flags.

2. In this mode, as long as P1284Active (nSelectIn) remains High, the controller drivers PIA27-20 as outputs, regardless of the contents of register E2. When software, or a DMA channel, writes the first byte to the Output Holding Register, the controller immediately transfers the byte to the Input/Output Register, clears Idle but negates DREQ only momentarily, to request another byte from software, or the DMA channel.

3. After each byte is transferred to the IOR, the controller waits 500 ns data setup time (as controlled by the PART register) before driving PtrClk (nAck) Low, and thereafter waits for the host to drive HostBusy (nAutoFd) High.

When this occurs, if software, or the DMA channel, has not written more data to the Output Holding Register, that is, nDREQ is still set, the controller uses the IOR and waits for software or the DMA channel to do so. If software, or the DMA channel, then writes data to the Output Holding Register, the controller clears DREQ and Idle. When there is data in the OHR and DREQ is 0, this guarantee is that it is appropriate to keep nDataAvail (nFault) and AckDataReq (nError) Low to indicate that more data is available, and the controller drives PtrClk (nAck) back to High. The controller then waits for a rising edge on HostClk (nStrobe), and then for the host to drive HostBusy (nAutoFd) Low, at which time it transfers the byte from the OHR to the Output Register, sets DREQ, and then returns to the event sequence at the start of this paragraph.

While this mode is in effect, software should monitor the interface for two conditions:

Case 1: Idle set and no more data to send.

Case 2: P1284Active (nSelectIn) Low.

In Case #1, the software should write to register E3 to keep PIA27-20 outputs momentarily, and then set the mode back to Peripheral Compatibility, so that the interface is fully under software control, set nDataAvail (nFault) and AckDataReq (nError) High to signify no more data, wait 500 ns, and set PtrClk (nAck) back to High. When HostBusy goes back to Low, the software should set PIA27-20 back to inputs.

In Case #2, if a falling edge on P1284Active (nSelectIn) occurs any time other than between a rising edge on HostClk (nStrobe), and the next falling edge on HostBusy (nAutoFd), the controller immediately transfers the byte to the IOR, clears Idle, and negates DREQ only momentarily, to request another byte.

In this mode, the alternate address for the Output Holding Register allows software to send a “channel address” or an RLE count value. Such bytes are typically written by software rather than a DMA channel. Writing to the alternate address loads the OHR and clears DREQ, like writing to the primary address, but clears a ninth bit that is set when software, or a DMA channel, writes to the primary address. A similar ninth bit is associated with the Input/Output Register, from which it drives the HostAck (nAutoFd) line.

Status interrupts in Peripheral Byte Mode include rising and falling edges on nPeriphRequest (nFault).

Host ECP Forward Mode

1. After a negotiation for ECP mode, "host" software should remain in Negotiation mode so that it has complete control of the interface, until one of two situations occurs. If software has data to send, it should optionally program the DMA channel to provide the data, and then set this mode. Alternatively, if software has no data to send and it detects that nPeriphRequest (nFault) has gone Low, indicating the peripheral is requesting reverse transfer, it should set PIA27-20 as inputs, wait 500 ns, drive nReverseRequest (nInit) to Low to indicate a reverse transfer, and then set HostECP Reverse mode. In other words, software should handle all aspects of ECP mode, other than active data transfer sequences.

2. Setting this mode configures PIA27-20 as outputs regardless of the contents of register E2. On entry to this mode, the controller sets Idle and DREQ to request a byte from software or a DMA channel, but these settings do not cause an interrupt request.

3. If software, or a DMA channel, writes data to the Output Holding Register while the Input/Output Register is empty, the controller immediately transfers the byte to the IOR, clears Idle, and negates DREQ only momentarily, to request another byte.

4. In this mode, the alternate address for the Output Holding Register allows software to send a "channel address" or an RLE count value. Such bytes are typically written by software rather than a DMA channel. Writing to the alternate address loads the OHR and clears DREQ, like writing to the primary address, but clears a ninth bit that is set when software, or a DMA channel, writes to the primary address. A similar ninth bit is associated with the Input/Output Register, from which it drives the HostAck (nAutoFd) line.

5. As each nine bits arrive in the IOR and thus out onto PIA27-20 and HostAck (nAutoFd), the controller waits one PHI clock and then drives HostClk (nStrobe) to Low. It then waits for the peripheral to drive PIA27-20 (Busy) to High, after which it drives HostClk (nStrobe) back to High. Then it waits for the peripheral to drive PIA27-20 (Busy) back to Low. When this has happened, if software or a DMA channel has written a new byte to the Output Holding Register, and thus cleared DREQ, the controller transfers the byte to the IOR, sets DREQ again, and returns to the event sequence at the start of this paragraph. Otherwise, it returns to the event sequence at the start of paragraph B.3. If software, or a DMA channel, does not provide a new byte for the time indicated in the PART register, the controller sets the Idle flag.

6. While this mode is in effect, software should monitor for the condition "Idle and no more data left to send", and do nPeriphRequest (nFault) Low. Host software has complete freedom as to whether to honor the peripheral’s reverse request on nFault while it has data to send. When there is no more data, software can set Host Negotiation mode to have full control of the interface, and if requested can drive P1284Active (nSelectIn) to Low to terminate ECP mode, or can set Host ECP Reverse mode, wait 500 ns, and drive nReverseRequest (nInit) to Low.

Status interrupts in Host ECP Forward mode include rising and falling edges on nPeriphRequest (nFault).

Zilog P R E L I M I N A R Y

Z80185 BIDIRECTIONAL CENTRONICS P1284 CONTROLLER

(Continued)
Peripheral ECP Forward Modes

1. After a negotiation for ECP mode, peripheral software should remain in Compatibility Negotiation mode with P1284Active (nSelectIn) High, so that it has complete control of the interface, though when it detects the host drive HostAck (nAutoFd) Low for the second time, it should then set nAckReverse (nError) High. If software has data to send, it should drive Peripheral Request (nFault) Low at the same time, and optionally program a DMA channel to provide the data. Whether or not it has data to send, software should then set one of the two ECP Forward modes.

2. In these modes, the controller configures PIA27-20 as inputs regardless of the contents of register E2. On entry to one of these modes, the controller clears the idle bit, if it had been set.

3. For each byte, the controller waits for the host to drive HostClk (nStrobe) to Low. When HostClk (nStrobe) is Low and software, or the DMA channel, has taken any previous byte and thus cleared DREQ, operation diverges into four cases depending on the state of HostAck (nAutoFd), the mode, the MSBit of the data, and the state of an internal 7-bit Run-Length Encoding (RLE) counter.

   a. If HostAck (nAutoFd) is High, indicating that this byte is neither an RLE value nor a Channel Address, the controller captures the data from PIA27-20 into the Input/Output Register, sets DREQ to request software, or the DMA channel, to provide the data, and sets Peripheral ECP Reverse mode.

   b. If HostAck (nAutoFd) is Low and the MS bit of the byte is zero (PIA27 is Low), the byte is an RLE repeat count. If the mode is "hardware RLE expansion," the controller transfers (the seven LSBs of) it to the RLE counter, leaves DREQ cleared, and drives PeripheralAck (Busy) High.

4. If HostAck (nAutoFd) is Low and the MS bit of the byte is zero (PIA27 is Low), the byte is an RLE repeat count. If the mode is "hardware RLE expansion," the controller transfers (the seven LSBs of) it to the RLE counter, leaves DREQ cleared, and drives PeripheralAck (Busy) High.

5. Thereafter, the controller waits for the host to drive HostClk (nStrobe) back to High, at which time it drives PeripheralAck (Busy) back to Low, and returns to the event sequence at the start of paragraph #3.

   a. If HostAck (nAutoFd) is Low, and PIA27 is High, the byte is a "channel address." In this case, or when PIA27 is Low and the mode is "software RLE handling," the controller captures the data from PIA27-20 into the Input/Output Register, leaves DREQ cleared to keep a DMA channel from storing the byte, and sets the idles bit, which it does not otherwise set while in this mode. Software should respond to this condition by reading the byte from the PIA 2 data register E3. Software can then do whatever else is needed to handle the situation, and then set BusyHigh. Thereafter the controller clears idle, waits (if necessary) for the host to drive HostClk (nStrobe) back to High, and then drives PeripheralAck (Busy) back to Low and returns to the event sequence at the start of paragraph #3.

   b. If P1284Active (nSelectIn) goes Low, the controller sets nAckReverse (nError) Low, optionally program a DMA channel to provide the data, and sets Peripheral ECP Forward mode.

6. If PeripheralAck (Busy) is Low, and the MS bit of the byte is zero (PIA27 is High), the byte is a "channel address." In this case, or when the LS bit is zero, but the mode is "software RLE handling," the controller captures the data from PIA27-20 in the IOR, leaves DREQ cleared, to keep a DMA channel from storing the byte, and sets idle, which it does not otherwise set in this mode. Software should respond to this condition by reading the byte from the PIA 2 data register E3, reprogramming a DMA channel, if necessary, and doing whatever else is needed to handle the channel address, and finally setting HostAck (nAutoFd) High. Thereafter the controller clears idle, waits for the peripheral to drive PeripheralAck (nAck) back to High, and then drives PeripheralAck (nAutoFd) back to Low, and returns to the start of the event sequence in paragraph #2 above.

7. If data has become available to be sent while this mode is in effect and software elects to send it, it should drive nAckReverse (nFault) Low to alert the host of this fact. Also software should monitor the controller for either of two conditions:

   a. If the host drives nReverseRequest (nInit) Low in response to a PeripheralRequest (nFault) Low, software should drive nAckReverse (nError) Low, optionally program a DMA channel to provide the data, and sets Peripheral ECP Reverse mode.

   b. If P1284Active (SelectIn) goes Low, the controller sets the idles bit in PARC, (thus occurs between the time the host drives HostClk (nStrobe) Low, and when the controller subsequently drives PeripheralAck (Busy) back to Low, in which case software should immediately enter Peripheral Compatibility Negotiation mode. If P1284Active goes Low, but iBlox stays zero, indicating a "legal" termination, software should enter Peripheral Inactive mode and sequence the nAckReverse (nError), PeripheralAck (Busy), PeripheralClk (nAck), PeripheralRequest (nFault), and Xflag (Select) lines to leave ECP mode.

Host ECP Reverse Modes

1. In these modes the controller configures PIA27-20 as inputs, regardless of the contents of register E2. On entry to one of these modes, the controller clears the idle bit, if it had been set.

2. For each byte, the controller waits for the host to drive PeripheralClk (nAck) Low. When PeripheralClk (nAck) is Low and software, or the DMA channel, has taken any previous byte and thus cleared DREQ, operation diverges into four cases depending on the state of PeripheralAck (Busy), the mode, the MS bit of the data, and the state of an internal 7-bit RLE counter.

   a. If PeripheralAck (Busy) is High, indicating that this byte is neither an RLE value nor a Channel Address, the controller captures the data from PIA27-20 in the IOR, sets DREQ to notify software, or the DMA channel to take the byte, and drives HostAck (nAutoFd) High. If the RLE counter is zero, the controller waits (if necessary) for the peripheral to drive PeripheralClk (nAck) back to High, after which it drives HostAck (nAutoFd) back to Low and returns to the event sequence at the start of paragraph #3.

   b. If PeripheralAck (Busy) is Low, and the MS bit of the byte is zero (PIA27 is Low), the byte is an RLE repeat count. If the mode is "hardware RLE expansion," the controller transfers (the seven LSBs of) it to the RLE counter, leaves DREQ cleared, and drives PeripheralAck (Busy) High.

3. If PeripheralAck (Busy) is Low, and the MS bit of the byte is zero (PIA27 is Low), the byte is an RLE repeat count. If the mode is "hardware RLE expansion," the controller transfers (the seven LSBs of) it to the RLE counter, leaves DREQ cleared, and drives PeripheralAck (Busy) High. Thereafter the controller waits for the peripheral to drive PeripheralClk (nAck) back to High, at which time it drives PeripheralAck (nAutoFd) back to Low and returns to the event sequence at the start of paragraph #2.

4. If PeripheralAck (Busy) is Low, and the MS bit of the byte is zero (PIA27 is Low), the byte is an RLE repeat count. If the mode is "hardware RLE expansion," the controller transfers (the seven LSBs of) it to the RLE counter, leaves DREQ cleared, and drives PeripheralAck (Busy) High. Thereafter the controller waits for the peripheral to drive PeripheralClk (nAck) back to High, at which time it drives PeripheralClk (nAutoFd) back to Low and returns to the event sequence at the start of paragraph #2.

5. If data has become available to be sent while this mode is in effect and software elects to send it, it should drive nReverseRequest (nInit) High to set, Host Negotiation mode to take full control of the interface, wait for PeripheralClk (nAck) High, and then set PIA27-20 as outputs.

6. Status interrupts in Host ECP Reverse mode include rising and falling edges on PeripheralRequest (nFault). PeripheralRequest carries a valid "reverse data available" indication during Reverse ECP mode. If so, enable status interrupts during this mode; if not, disable them.
Peripheral ECP Reverse Mode

1. In this mode, as long as nReverseRequest (nInit) is Low, and P1284Active (nSelectIn) is High, the controller drives the contents of the Input/Output Register onto PIA27-20, regardless of the contents of the E2 register. On entry to this mode, the controller sets Idle, and sets DREQ to request data from software, or a DMA channel.

2. If software, or a DMA channel, writes data to the Output Holding Register while the Input/Output Register is empty, the controller immediately transfers the byte to the IOR, clears Idle, and negates DREQ only momentarily, to request another byte.

3. In this mode, an alternate address for the Output Holding Register allows software to send a "channel address" or an RLE count value. Such bytes are not typically written by a DMA channel. Writing to this alternate address loads the OHR and clears DREQ, the same as writing to the primary address, but clears a ninth bit set when software, or a DMA channel, writes to the primary address. A similar ninth bit is associated with the IOR, and drives the PeriphAck (Busy) line in this mode.

4. As each nine bits arrive in the IOR, and thus out onto PIA27-20 and PeriphAck (Busy), the controller waits one PHI clock, and then drives PeriphClk (nAck) Low. It then waits for the host to drive HostAck (nAutoFd) High, after which it drives PeriphClk (nAck) back to High. The controller then waits for the host to drive HostAck (nAutoFd) back to Low. When this has happened, if software, or the DMA channel, has written a new byte to the Output Holding Register, and thus cleared DREQ, the controller transfers the byte to the IOR, sets DREQ again, and returns to the start of the event sequence in this paragraph. Otherwise, it returns to the event sequence at the start of paragraph #2. If software, or the DMA channel, doesn't provide new data within the time indicated by the PART register, the controller sets the Idle bit.

5. While this mode is in effect, software should monitor whether the host drives nReverseRequest (nInit) High. If it detects this, it should set the mode back to Peripheral ECP Forward, wait 500 ns and then drive nAckReverse (PError) back to High, before proceeding as described for Peripheral ECP Forward mode above.

6. Status interrupts in Peripheral ECP Reverse mode include rising and falling edges on P1284Active (nSelectIn) and nReverseRequest (nInit). Since there are no "legal terminations" during the time this mode is set, the controller sets B80p for any falling edge on P1284Active (nSelectIn) in this mode.

System Configuration Register

This register controls a number of device-level features on the Z80185 and includes the following control bits:

- Daisy-Chain Configuration
- ROM Emulator Mode (REME)
- 0 = Data Bus in Normal Mode
  1 = Data Bus in ROM Emulator Mode
- 0 = ESCC CLK is PHI
  1 = ESCC CLK is PHI/2
- 0 = /RTS0, /CTS0, CKA0
  1 = TxS, RxS, CKS
- Disable /ROMCS
  0 = /ROMCS is Enabled
  1 = /ROMCS is Disabled
- Daisy-Chain Configuration
- Daisy-Chip Configuration
- Decode High I/O
  0 = A15-8 not decoded for "non-180" registers
  1 = A15-8 must be 00 to access "non-180" regs.
Zilog P R E L I M I N A R Y

Z80185 PIA AND MISCELLANEOUS REGISTERS (Continued)

System Configuration Register (Continued)

Bit 7. Decode High I/O. If this bit is 0, as it is after a Reset, A15-8 are not decoded for the registers for which A7-6 are 1, that is, the registers for the EMSCC, CTCs, I/O Ports, Bidirectional Centronics Controller. If this bit is 1, A15-8 must all be zero to access these registers, as for the other registers in the Z80185. When set to 0, this bit is compatible with the Z80181 and Z80182, and allows shorter, and more basic I/O instructions to be used to access these registers. Alternately, when set to 1, this bit allows more extensive off-chip I/O.

Bit 6. Daisy-Chain Configuration Bit 2. This bit is described with bits 1-0 below.

Bit 5. Disable /ROMCS. When this bit is 1, /ROMCS is forced to High, regardless of the status of the address decode logic. This bit Resets to 0 so that /ROMCS is enabled.

Bit 4. When this bit is 0, the /RTS0/TXS, /CTS0/RXS, and CKA0/CKS pins have the /RTS0, /CTS0 and CKA0 functions, respectively. When this bit is 1, the pins have the TXS, RXS, and CKS functions, and the CSO0 facility can be used. When this bit is 1, if ASC0 is used, the "CTS auto-enable" function must not be enabled. The multiplexing of CKAO is important only with respect to output — the same external clock could be used for both ASC0 and the CS0.

Bit 3. When this bit is 0, the PCLK clock of the EMSCC is the same as the processor's PHI clock. When this bit is 1, this clock is PHI/2. Set this bit if the PHI clock is too fast for the EMSCC.

Bit 2. ROM Emulator Mode Enable. When this bit is 1, read data from on-chip sources is driven onto the D7-D0 pins, as shown in Table 6. This bit Resets to 0.

Bits 1-0. These bits, plus bit 6, determine the routing of the on-chip interrupt daisy-chain, and thus the relative interrupt priority of the on-chip interrupt sources on the daisy-chain as shown in Table 5.

Table 5. Interrupt Daisy-Chain Routing

<table>
<thead>
<tr>
<th>b6</th>
<th>b1</th>
<th>b0</th>
<th>Daisy Chain Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>IEC pin -&gt; EMSCC -&gt; CTC -&gt; Bidirectional Centronics Controller -&gt; IEO pin</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>IEC pin -&gt; EMSCC -&gt; Bidirectional Centronics Controller -&gt; CTC -&gt; IEO pin</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>X</td>
<td>IEC pin -&gt; Bidirectional Centronics Controller -&gt; EMSCC -&gt; CTC -&gt; IEO pin</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>IEC pin -&gt; CTC -&gt; EMSCC -&gt; Bidirectional Centronics Controller -&gt; IEO pin</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>IEC pin -&gt; CTC -&gt; Bidirectional Centronics Controller -&gt; EMSCC -&gt; IEO pin</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>X</td>
<td>IEC pin -&gt; Bidirectional Centronics Controller -&gt; CTC -&gt; EMSCC -&gt; IEO pin</td>
</tr>
</tbody>
</table>

Note:
On-chip ROM should be fast enough to support no-wait-state operation at the maximum specified clock rate, but this field is included as a "hedge" against difficulties in this area, as well as to provide timing compatibility in unusual circumstances.

Bits 3-2. This field controls how many wait states are inserted for accesses to on-chip ROM, and is encoded like bits 7-6. Note: On-chip ROM should be fast enough to support no-wait-state operation at the maximum specified clock rate, but this field is included as a "hedge" against difficulties in this area, as well as to provide timing compatibility in unusual circumstances.

Bits 1-0. This field controls how many wait states are inserted for accesses to external memory in which neither RAMCS nor ROMCS is asserted, and is encoded the same as bits 7-6.

All fields in this register Reset to 11. The 4-wait-state feature is included to allow the use of commodity DRAMs with a clock rate at, or near, the maximum.

Figure 86. WSG Chip Select Register (I/O Address %D8)

Bits 7-6. This field controls how many wait states are inserted for accesses to external memory in which RAMCS is asserted: 00 = none, 01 = 1, 10 = 2, 11 = 4 wait states.

Bits 5-4. This field controls how many wait states are inserted for accesses to external memory in which ROMCS is asserted, and is encoded like bits 7-6.
Interrupt Edge Register

**Interrupt Edge Register (I/O Address %DF)**

<table>
<thead>
<tr>
<th>Bit 7-6</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>IO/CKA0 is IO/CKA0</td>
</tr>
<tr>
<td>1</td>
<td>IO/CKA0 is CKA0</td>
</tr>
</tbody>
</table>

**Drive Select**
- 0: Select normal drive
- 1: Select low noise (33%) drive capabilities

**INT1 Sense/Unlatch**
- 0: /INT1 is low
- 1: /INT1 is high
- **Out:** unlatch edge detection

**INT2 Sense/Unlatch**
- 0: /INT2 is low
- 1: /INT2 is high
- **Out:** unlatch edge detection

**INT1 Mode Select**
- 0X: Normal level detect
- 10: Falling (Neg) edge det.
- 11: Rising (Pos) edge det.

**INT2 Mode Select**
- 0X: Normal level detect
- 10: Falling (Neg) edge det.
- 11: Rising (Pos) edge det.

**Bits 5-4**
- These bits control the interrupt capture logic for the external /INT1 pin. When these bits are 0X, the /INT1 pin is level sensitive and Low active. When these bits are 10, negative edge detection is enabled. Any falling edge will latch an active Low on the internal /INT1 to the processor. This interrupt must be cleared by writing a 1 to bit 2 of this register. Programming these bits to 11 enables rising edge interrupts to be latched. The latch must be cleared in the same fashion as for a falling edge.

**Bit 3**
- Software can read this register to sense the state of the /INT2 pin. Writing a 1 to this bit clears the edge detection logic for /INT2.

**Bit 2**
- Software can read this register to sense the state of the /INT1 pin. Writing a 1 to this bit clears the edge detection logic for /INT1.

**Bit 1**
- This bit selects low noise or normal drive for the parallel ports, bidirectional Centronics controller pins, Chip Select pins, and EMSCC pins as follows:

- **PIA 10-13**
  - RTS
  - nFault
  - **PIA 14-16/ZCT0 0-2**
    - DTR
    - nInit
  - **PIA 27-20**
    - TXD
    - nsSelect
  - **ROMCS**
    - TRXCS
    - nStrobe
  - **RAMCS**
    - BUSY
    - nError
  - **I/OCS**
    - nAck
    - Select
  - **IEO**
    - nAutoFd

A 1 in this bit selects the low noise option, which is a 33 percent reduction in drive capability. A 0 selects normal drive, and is the default after power-up. Additionally, refer to CPU Register (CCR) for a list of the pins that are programmable for low drive, via the CCR register.

**Bit 0**
- If this bit is 1, the IO/CKA0/CKA1 pin has the CKA1 function. The pin is always connected to the DCD input of ASC0, so if this pin is 1, and ASC0 is used, it should not be programmed to use DCD as a receive auto-enable.

---

**Individual Pin Selection Between PIA1 and CTCs**

The assignment of the choice between PIA1 and CTC I/Os is controlled by the PIA1/CTC Pin Select Register (Figure 79).

**Bits 7**
- Reserved, and should be programmed as 0.

**Bits 6-4**
- These bits control the pin selection between PIA1 and CTC I/Os. For each of these bits that is 0, the pin is driven with the state of the corresponding bit of the PIA1 Data register, while for each of these bits that is 1, the associated pin is driven with the indicated CTC output. These bits Reset to 0.

**Bit 3**
- The assignment of the choice between PIA1 and CTC I/Os is controlled by the PIA1/CTC Pin Select Register (Figure 79).

**Bits 2**
- These bits control whether the CLK/TRG inputs of CTCs 3-1 are taken from PIA1-3, respectively, or from the ZC/TO outputs of CTC2-0, respectively. These bits do not have any affect on the operating mode of the CTCs.

**Bit 0**
- This bit is reserved and should be programmed as 0. CTC0's CLK/TRG0 input is always connected to the PIA10 pin.
Z80185 PIA AND MISCELLANEOUS REGISTERS (Continued)

CTC Control Registers

Channel Control Byte

This byte is used to set the operating modes and parameters. Bit D0 must be a 1 to indicate that this is a Control Byte (Figure 82).

The Channel Control Byte register has the following fields:

- **Bit D7**: Interrupt Enable. This bit enables the interrupt logic so that an internal INT is generated at zero count. Intermittent interrupts are programmed in either mode, and may be enabled or disabled at any time.
- **Bit D6**: Mode Bit. This bit, along with bit 3, is used to select either Timer mode or Counter mode (Table 8).
- **Bit D5**: Prescaler Factor. This bit selects the prescaler factor for use in the timer mode. Either divide-by-16 or divide-by-256 is available.
- **Bit D4**: Clock/Trigger Edge Selector. This bit selects the active edge of the CLK/TRG input pulses.
- **Bit D3**: Mode Bit. This bit, along with bit 6, selects either Timer mode or Counter mode (Table 8).
- **Bit D2**: Time Constant. This bit indicates that the next byte programmed is time constant data for the downcounter.
- **Bit D1**: Software Reset. Writing a 1 to this bit indicates a software reset operation, which stops counting activities until another time constant word is written.

<table>
<thead>
<tr>
<th>Table 8. CTC Operation Modes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCW6</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

Table 8. CTC Operation Modes

| CCW6 | CCW3 | Operation                        |
| 0    | 0    | (Auto Start) Timer mode. The prescaler is clocked by PHI, and the counter is clocked by the prescaler. Counting is enabled when the timer constant is loaded. |
| 0    | 1    | Timer with CLK/TRG Trigger. The prescaler is clocked by PHI, and the counter is clocked by the prescaler. Timing starts when the transition specified by D4 is detected on the PIA pin, or for CTC3-1, the ZC/TO output of CTC3-0, respectively. |
| 1    | 0    | Classic Counter mode. The counter is clocked by the PIA pin, or for CTC3-1 the ZC/TO output of CTC3-0, respectively. |
| 1    | 1    | Long Counter mode. The prescaler is clocked by the PIA pin, or for CTC3-1 the ZC/TO output of CTC3-0, respectively, and the counter is clocked by the prescaler. |

Figure 89. CTC Channel Control Word
Z80185 PIA AND MISCELLANEOUS REGISTERS (Continued)

CTC Control Registers (Continued)

Time Constant

Before a channel can start counting, it must receive a time constant. The time constant value may be anywhere between 1 and 256, with 0 indicating a count of 256 (Figure 90).

Interrupt Vector

If one or more of the CTC channels have interrupt enabled, then the Interrupt Vector Word should be programmed. Only the five most significant bits of this word are used, bit D0 must be 0. Bits D2-D1 are automatically modified by the CTC channels after responding with an interrupt vector (Figure 91).

Watch-Dog Timer

The Z80185's Watch-Dog Timer (WDT) facility is identical to Zilog's Z84C15 WDT with the following exceptions:

1. The HALT mode field of the WDT Master Register is not used. Power control is handled as on the Z8S180.

Watch-Dog Control Registers

Two registers control WDT operations. These are WDT Master Register (WDTMR; I/O Address F0h) and the WDT Command Register (WDTCR; I/O Address F1h). WDT logic has a "double key" structure to prevent accidental disabling of the WDT.

Enabling the WDT. The WDT is enabled by reset, and setting the WDT Enable Bit (WDTMR7) to 1.

Disabling the WDT. The WDT is disabled by clearing WDT Enable bit (WDTMR7) to 0 followed by writing "B1h" to the WDT Command Register (WDTCR; I/O Address F1h).

Clearing the WDT. The WDT can be cleared by writing "4Eh" into the WDTCR.

Watch-Dog Timer Master Register (WDTMR; I/O Address F0h). This register controls the activities of the Watch-Dog Timer.

Bit D7, Watch-Dog Timer Enable (WDTE). The WDT can be enabled by setting this bit to 1. To disable WDT, write 0 to this bit, followed by writing "B1h" to the WDT Command Register. Upon Power-On Reset, this bit is set to 1 and the WDT is enabled.

Bit D6-D5, WDT Periodic Field (WDTP). This 2-bit field determines the desired time period. Upon Power-On reset, this field sets to "11".

00 - Period is (Tcc * 2^16)
01 - Period is (Tcc * 2^18)
10 - Period is (Tcc * 2^20)
11 - Period is (Tcc * 2^22)

Bit D4. If this bit is 1 and the WDT times out, the Z80185 drives the Reset pin Low to reset external logic. If this bit is 0, a WDT timer only resets the Z80185 internally.

Bit D3-D0. Reserved. These three bits are reserved and should always be programmed as 0011. Reading these bits returns 0011.
Parallel Ports

The Z80185 has two 8-bit bidirectional ports. Each bit is individually programmable for input or output. Each port includes two registers: the Port Direction Control Register and the Port Data Register. The second port also includes an Alternate Address that is used with the Bidirectional Centronics feature.

### PIA 1 Data Direction Register

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 94. PIA 1 Data Direction Register (I/O Address %E0)**

The data direction register determines which of the PIA 16-10 pins are inputs and outputs. When a bit is set to 1, the corresponding bit in the PIA 1 Data Register is an input. If the bit is 0, then the corresponding pin is an output. These bits must be set appropriately if these pins are used for CTC inputs and outputs.

### PIA 1 Data Register

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 95. PIA 1 Data Register (I/O Address %E1)**

When the processor writes to the PIA 1 Data Register, the data is stored in the internal buffer. Any bits that are output are then driven on to the pins. In certain modes of the Bidirectional Centronics Controller, an intermediate register called the Output Holding Register is activated, and the transfer of data from the OHR to the pins is under the control of the controller.

When the processor reads the PIA 1 Data Register, the data on the external pins is returned. In certain modes of the Bidirectional Centronics Controller, reading from this address shows the data stored in the port register from PIA 27-20 under the control of the controller.

### PIA 2 Data Direction Register

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 96. PIA 2 Data Direction Register (I/O Address %E2)**

Reading and writing this register is exactly the same as reading and writing address E3 as described above, except that in certain modes of the Bidirectional Centronics Controller, writing to this address sets a “ninth bit” in the opposite sense from writing address E3, and this drives one of the control outputs with the opposite polarity.
APPENDIX C: ELECTRIC DIAGRAMS

In this appendix are available some electric diagrams of the most frequently used GPC® 184 interfaces. All these interface can be yourself produced and some of them are standard grifo® cards and, if required, they can be directly ordered.

**FIGURE C1: PPI EXPANSION ELECTRIC DIAGRAM**
FIGURE C2: SPA 03 ELECTRIC DIAGRAM
Figure C3: QTP 16P electric diagram
**FIGURE C4: QTP 24P ELECTRIC DIAGRAM (1 OF 2)**
FIGURE C5: QTP 24P ELECTRIC DIAGRAM (2 OF 2)
**Figure C6: ABACO® I/O BUS INPUT OUTPUT ELECTRIC DIAGRAM**

Date: 28/04/1999

Rel. 1.2
**Title:** BUS interface  
**Date:** 16/11/98  
**Page:** 1 of 1

**Figure C7:** BUS INTERFACE ELECTRIC DIAGRAM
FIGURE C8: IAC 01 ELECTRIC DIAGRAM
APPENDIX D: ALPHABETICAL INDEX

Simbols

82c55  C-1

A

ABACO® I/O BUS  3, 11, 30, 39, 44, C-6
Acces time  8
Activity LED  25, 48
Addresses  39
ASCI  4, 8, 12, 20, 34, 42
Assistance  1

B

Back up  9, 10, 29, 48
Battery  9, 10, 28, 29, 48
Bibliography  52
Block diagram  5
Board configuration  6, 44, 47

C

Clock  4, 8
Compiler  37
Components maps  7
Connections  24, 51
Connectors  8, 15
   CN1  11
   CN2  10
   CN3A  12
   CN3B  14
   CN5  22
   CN5A  20
Consumption  9, 25
Control logic  7, 39
Counter  6, 22
CPU  3, 8, 42, 48, B-1
Crystal  4, 8
CSI/O  4, 8, 20, 34, 42
CTC  6, 8, 22, 44
Current loop  4, 14, 18, 32, 33

D

Debugger  38
Default configuration  26, 31
Devices  8, 30, 39, B-1
Dimension  A-1
DIN Ω rails  8, A-1
DMAC  8, 43

E
EEPROM  4, 8, 31, 40, 47
Electric diagrams  C-1
EMSCC  4, 8, 14, 32, 44
EPROM  4, 8, 31, 40
Expansion  3, 49, C-1
External cards  49

F
Features
  electric  9
  general  2, 8
  physical  8
FLASH EPROM  4, 8, 31, 40

G
GDOS  36, 40
General features  2
GET 80  36

H
Handshake  12, 32, 34
Humidity  9

I
I/O addresses  41
I/O lines  7, 22, 24, C-1
IEEE  7, 8, 22, 34, 44, 48
Installation  10
Instructions set  3
Interfaces  24, C-1
Interrupt  20, 30
Interrupts  43

J
JEDEC  31
Jumpers  26
  2 pins  28
  3 pins  29
  5 pins  28
location 27
solder 29

L
LED 15, 25, 48

M
Maintenance 1
Maps 39
Memory 4, 8, 15, 31, 39
MMU 40, 43
Mounting A-1

N
Network 17, 19, 32

O
Operating system 36
Operator interfaces 24, C-3, C-4
Options 31, 32, A-1

P
Peripherals 3, 8
Photo 9
PIA 7, 8, 22, 44
Piggy back A-1
Power failure 9, 29, 30, 35
Power supply 9, 11, 25, 35
Printer 24, C-8
Protection 25
PRT 6, 8, 42

R
Real Time Clock 6, 30, 43, 45
Registers setting 48
Registers 41
Release 1
Reset 30, 34
Reset contact 8, 15, 25, 30
RS 232 4, 12, 14, 16, 32, 33
RS 422 4, 9, 14, 16, 32, 33
RS 485 4, 9, 14, 16, 33, 34
RUN/DEBUG mode 6, 28, 47
S
Serial communication  4, 13, 21, 32
Serial drivers  33
Serial line A  4, 12, 32, 34
Serial line B  4, 14, 32
Serial line C  4, 20, 34
Size  8, A-1
Software  36
SRAM  4, 8, 31, 40

T
Temperature  9
Termination and forcing circuit  9, 17, 34
Timer  6, 22

V
Version  1

W
Warranty  1
Watch dog  6, 8, 28, 30, 44, 47
Weight  8