GPC® 552

General Purpose Controller 84C552 Philips

TECHNICAL MANUAL

CPU 80C552, 87C552, 80C562, 87C562 with 22.1184 or 29.4912 MHz;
1 serial line configurable in RS 232, RS 422-485 or Current Loop;
supports IDLE MODE or POWER-DOWN MODE; 44 I/O TTL lines; 16
bits Timer-Counter with 4 Capture registers and 3 compare registers; 6
Set-Reset outputs associated to comparator T2, plus 2 Toggle outputs;
Watch Dog; Dip Switch; Buzzer; 1 diagnose LED; 1 activity LED; 32K
EPROM, 32K RAM/EEPROM, 32 RAM/EEPROM or EPROM; optional
serial EEPROM from 512 to 2048 bytes; optional Lithium backed RTC
plus 256 bytes RAM; eight 10 bits A/D converter lines; FC BUS; 3 standard 20 pins Abaco® I/O connectors; 1 standard 20 pins Abaco® A/D connector; Power supply from 220 Vac or low voltage;
very low power required; card for DIN 46277-1 and 3 rail.
IMPORTANT

Although all the information contained herein have been carefully verified, grifo® assumes no responsibility 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
  - CLOCK DEVICE
  - POWER SUPPLY
  - SERIAL COMMUNICATION
  - MEMORY DEVICES
  - PERIPHERAL DEVICES
  - CONTROL LOGIC
  - RESET CONTACT

## TECHNICAL FEATURES
- 9
  - GENERAL FEATURES
  - PHYSICAL FEATURES
  - ELECTRIC FEATURES

## INSTALLATION
- 11
  - CONNECTIONS
  - CN3 - CONNECTOR FOR POWER SUPPLY
  - CN1 - CONNECTOR FOR PPI 82C55 PORT B, PWM, GENERAL OUTPUT
  - CN2 - CONNECTOR FOR PPI 82C55 PORT A AND C
  - CN4 - CONNECTOR FOR CPU PORT 5, A/D CONVERTER INPUTS
  - CN5 - CONNECTOR FOR I2C BUS AND RS 232 SECOND SERIAL PORT
  - CN8 - CONNECTOR FOR EXTERNAL BACK UP BATTERY
  - CN6 - CONNECTOR FOR MICROPROCESSOR I/O
  - CN7 - CONNECTOR FOR SERIAL COMMUNICATION
  - LEDS
  - I/O CONNECTION
  - RESET KEY
  - TEST POINT
  - ON BOARD INPUT
  - POWER SUPPLY SELECTION
  - ANALOG INPUT TYPE SELECTION
  - TRIMMERS AND CALIBRATION
  - LOCAL USER INTERFACES
  - JUMPERS
    - 2 PINS JUMPER
    - 3 PINS JUMPER
    - 4 PINS JUMPER
    - 5 PINS JUMPER
FIGURE INDEX

FIGURE 1: BLOCK DIAGRAM ................................................................. 5
FIGURE 2: COMPONENTS MAP ............................................................. 8
FIGURE 3: CN3 - POWER SUPPLY CONNECTOR ................................. 11
FIGURE 4: CN1 - CONNECTOR FOR PPI 82C55 PORT B, PWM, GENERAL OUTPUT .................................................. 12
FIGURE 5: CN2 - I/O CONNECTOR FOR PPI 82C55 PORT A AND C .......................................................... 13
FIGURE 6: CN 4- CONNECTOR FOR CPU PORT 5, A/D CONVERTER INPUTS .............................................................. 14
FIGURE 7: A/D CONVERTER INPUTS DIAGRAM .................................. 15
FIGURE 8: CN5 - SCREW TERMINAL CONNECTOR FOR THE I2C BUS AND THE SECOND SERIAL PORT ........................................... 16
FIGURE 9: CN8 - CONNECTOR FOR EXTERNAL BACK UP BATTERY ......................................................... 17
FIGURE 10: CN6 - CONNECTOR FOR MICROPROCESSOR I/O .................. 18
FIGURE 11: I/O CONNECTION DIAGRAM ............................................ 19
FIGURE 12: CN7-CONNECTOR FOR SERIAL COMMUNICATION .................. 20
FIGURE 13: SERIAL COMMUNICATION DIAGRAM .................................. 21
FIGURE 14: LEDS TABLE .................................................................. 22
FIGURE 15: LEDS, CONNECTORS, DIP SWITCH, ETC. LOCATION .......... 23
FIGURE 16: JUMPERS SUMMARIZING TABLE ......................................... 27
FIGURE 17: 2 PINS JUMPERS TABLE .................................................. 28
FIGURE 18: JUMPERS LOCATION ........................................................ 29
FIGURE 19: 3 PINS JUMPERS TABLE .................................................. 30
FIGURE 20: 4 PINS JUMPER TABLE ..................................................... 31
FIGURE 21: 5 PINS JUMPER TABLE ..................................................... 31
FIGURE 22: MEMORY SELECTION TABLE ............................................ 33
FIGURE 23: I/O ADDRESSES TABLE .................................................... 36
FIGURE 24: MODE 1 MEMORY CONFIGURATION ................................... 37
FIGURE 25: MODE 2 MEMORY CONFIGURATION ................................... 38
FIGURE 26: MODE 3 MEMORY CONFIGURATION ................................... 39
FIGURE 27: MODE 4 MEMORY CONFIGURATION ................................... 40
FIGURE 28: CARD PHOTO ................................................................ 41
FIGURE 29: AVAILABLE CONNECTIONS DIAGRAM ............................. 51
FIGURE A1: SERIAL COMMUNICATION JUMPERS LOCATION ................. A-1
FIGURE A2: MEMORY SELECTION JUMPERS LOCATION ......................... A-2
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® 552 card release 180796 and later. The validity of the bring informations is subordinate to the number of the card release. The user must always verify the correct 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 in the top right corner on the component side).
**GENERAL FEATURES**

GPC® 552 is a new powerfull control card, featuring ultra low power consumption, capable of operating in stand alone mode or as an intelligent peripheral in a wider remote control or acquisition network, with an excellent ratio price/performance. The GPC® 552 module is secured in a plastic mount for connection to standard DIN 46277-1 and 46277-3 omega rails, thereby dispensing with the need of a rack and allowing a less costly mounting direct to the electrical control panel. The card can solve many of the standard industrial problems as it is but, when necessary, its power and its application fields can be increased through the communication lines. The GPC® 552 is supplied with a series of standard connectors, all Abaco® specification, allowing immediate use of the many I/O modules available.

A 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.

- card size: **100x195** mm for DIN 46277-1 and DIN 46277-3 omega rails;
- CPU **80C552, 87C552, 80C562, 87C562**, with of **22.1184** or **29.4912** MHz clock, **8051** INTEL code compatible;
- sockets for **32K EPROM, 32K RAM or EEPROM, 32K RAM, EEPROM or EPROM**;
- option of serial **EEPROM** from **512** to **2048** Bytes;
- option of **RTC** with **256** Bytes of **RAM**, backed with **lithium** battery;
- maximum **1024** bytes of serial **EEPROM** when **RTC** is mounted;
- **8 lines 10 bits** A/D converter, with range **+2.49V** or **+5V** and conversion time **50µs** at **12MHz**;
- standard connector for **I2C bus**;
- **44 TTL I/O** lines, configurable via software;
- **2 independent 8 bits** PWM output;
- software readable octal **dip switch**;
- on board **BUZZER** circuitry;
- **Watch-Dog** configurable from software and deselectable;
- **16 bit timer counter** with 4 capture and 3 compare registers;
- **6 set reset** output linked to comparator **T2**, plus **2 toggle** output;
- Standard **16 bit timer counter** register;
- **RS232** or **RS422-485** or **Current Loop** serial line;
- **1 RS 232** software serial line;
- sundry diagnostics and activity indicator **LEDs**, with software control;
- **3 standard ABACO® 20 way I/O** connectors;
- **1 standard ABACO® 20 way A/D** connector;
- facility of operation in **Idle Mode** or **Power down Mode**;
- built in **mains** power circuit or **low voltage** power supply;
- wide range of development software including: **Monitor, Debugger, Assembler, BASIC Interpreter, BASIC Compiler, FORTH, C, PASCAL**, etc.
CPU

The GPC® 552 can use all version of microprocessors namely 80C552, 87C552, 80C562, 87C562. These 8 bit microprocessors are code compatible with the 8051 INTEL and so they have an extended instruction set, fast execution time, easy use of all kind of memory and an efficient interrupt management.

The most important features of the described microprocessors, are:

80C552:  
- 8k bytes EPROM, 256 bytes RAM;  
- 6 independent 8 bits I/O ports;  
- 2 standard 16 bits Timer/Counters;  
- 16 bits Timer/Counters with Capture and Compare function;  
- 2 priority level for interrupts;  
- 8 lines 10 bits A/D converter;  
- 2 independent 8 bits PWM outputs;  
- 1 synchronous/asynchronous serial line;  
- 1 I2C bus line;  
- Watch Dog Timer;  
- Idle mode or Power down mode;

For further informations, please refer to specific documentation of the manufacturing company.

CLOCK DEVICE

On GPC® 552 there are two separate circuits with crystal to generate the clock signal for the microprocessor (22.1184 or 29.4912 MHz) and the clock signal for IC 25 Real Time Clock (32.768 Hz). The choice of using two circuits and two separated crystals, has the advantage to change the microprocessor working speed (when best performances are required) without additional changes.

POWER SUPPLY

One of the most important features of GPC® 552 is its on board power supply which is capable to generate the only +5Vdc supply voltage needed. The card can be powered in four different ways: mains power supply requiring 230 Vac, linear power supply requiring 6÷10 Vac (+12 Vdc), switching power supply requiring 8÷26 Vac or no power supply requiring +5 Vdc (for more informations please refer to the paragraph "POWER SUPPLY VOLTAGES") and the power can be provided by a standard connector easy to install. The board takes advantage of componentistic and circuital choices intended to reduce the consumption, including the feature to work in power down and idle mode. The power supply type must be specified at the moment of the order because the User cannot change it.
SERIAL COMMUNICATION

Serial communication based on the microprocessor inside SIO 0 is completely software settable both for protocol and baud rate.
Setting these parameters is possible by programming the SIO 0 registers internal to the 80C552 CPU, for further informations please refer to the manufacturer documentation or see the appendix B of this manual.
SIO 0 can be buffered as RS232, Current Loop or RS422-485 both Full Duplex or Half Duplex, the selection of this protocol happens by acting on a set of jumpers.
80C552 CPU includes also a second serial device called SIO 1 used to communicate with devices supporting the FC bus protocol.
For further informations about SIO 1 please refer to the manufacturer documentation or see the appendix B of this manual.

MEMORY DEVICES

On the card can be mounted 98 K and 256 bytes of memory divided with a maximum of 32K EPROM, 32K RAM/EEPROM/EPROM, 32K EEPROM/RAM/EPROM, 256 bytes of serial RAM+RTC and 2K of serial EEPROM.
The GPC® 552 memory configuration must be chosen considering the application to realize or the specific requirements of the User. Normally the card is provided with 32K byte of RAM and all different configuration must be specified from the User, at the moment of the order. By selecting the backed RAM module or the EEPROM module, there is the possibility to keep data also when power supply is missed; in this way the card is always able to maintain parameters, logged data, system status and configuration, etc. without using expensive external UPS. In addition to that, the IC 25 RAM module, can be provided with on board lithium battery and with Real Time Clock which manages time (hours, minutes, seconds) and date (day, month, year, day of the week).
The addressing of memory devices is controlled by a specific on board circuit, that provides to allocate the devices in the microprocessor address space. For further informations about memory configuration, sockets description and jumpers connection, please refer to chapter "HARDWARE DESCRIPTION", "PERIPHERAL SOFTWARE DESCRIPTION" and to the paragraph "MEMORY SELECTION".
FIGURE 1: BLOCK DIAGRAM
PERIPHERIAL DEVICES

GPC® 552 is the right card to solve many control problems in automation fields, in fact it is provided of some peripheral components that facilitate the connection and the management of external system like probes, switches, relays, motor controllers, etc. These peripherals are:

- **SIO 0**: it is a microprocessor peripheral device that manages serial communication with any other system provided of RS 232, RS 422, RS 485 or current loop serial line. By software the User can set baud rate, length of character, stop bit number, parity and handshake through a simple programmation of internal microprocessor registers.

- **SIO 1**: it is a microprocessor peripheral device that manages the FC bus physical protocol.

- **A/D converter**: it is a microprocessor peripheral device that converts 8 different analog signals with 10 bits of resolution. By software the User selects the channel to convert, starts the conversion and controls the end of conversion, through programmation of internal microprocessor registers.

- **PWM**: it is a microprocessor peripheral device that generates two separate PWM signals. By software the User can sets frequency and duty cycle of each signal, programming three 8 bits internal registers.

- **WATCH-DOG**: it is a microprocessor peripheral device that can reset the card at programmable time intervals, if not retriggered. By software the User can set the watch dog time (from 2 ms to 500 ms), enable or disable the watch dog device and retrigger the circuit to prevent card reset.

- **Board Configuration**: in order to make the board and the User program easily configurable an 8 ported dip switch has been installed. This dip switch has a double purpose: to select the RUN DEBUG mode and to configure the control software. The status of part of the switches is readable by the User program, this allows the User to manage many situations by an only program, without the need to employ other input lines (typical applications are: selecting the language, setting software parameters, determining operational modes etc.). There is also an activity LED that the User can manage to visualize the system status.

- **Real Time Clock**: it is an optional device that can be provided with the backed RAM module to be mounted on IC25. It can manage hours, minutes, secindes, day of month, month, year and day of week in complete autonomy.

- **serial EEPROM**: it is inispensable to mount the EEPROM module (IC 24) when the User needs to keep informations even when power supply is absent in an extremly secure way without taking away bytes to the backed RAM. Size of this module can vary from 512 to 2048 bytes, as it is optional the User must specify it explicitly in the order.

- **BUZZER**: a specific circuitry to drive a capacitive buzzer is installed on the GPC® 552 board. It can be software enabled/disabled and can be used to generate any kind of audio feedback, acoustic alarm, etc.

- **PPI 82C55**: it manages 24 TTL I/O lines divided in three 8 bit parallel ports. The lines direction is software settable at byte level. The PPI 82C55 is completely driven by software programming of the four registers allocated in the addressing space of the microprocessor.
For further information about peripheral device please refer to appendix B of this manual or to the technical documentation of the manufacturing company.

**CONTROL LOGIC**

The addresses of all peripheral device's registers and of memory devices on GPC® 552 are assigned from a specific control logic that allocates all these devices in the microprocessor addressing space. For further information please refer to paragraph "I/O ADDRESSES" of this manual.

**RESET CONTACT**

P1 reset contact of the GPC® 552 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 status.
FIGURE 2: COMPONENTS MAP
TECHNICAL FEATURES

GENERAL FEATURES

Devices
- 44 programmable TTL Input/Output lines
- 3 Timer Counters
- 1 bidirectional RS232, RS422-485 or current loop serial line
- 1 FC bus line
- 1 bidirectional RS232 software serial line
- 1 Watch-Dog
- 8 A/D converter lines
- 1 local key for reset
- 1 Real Time Clock
- 1 Buzzer
- 2 LEDs
- 1 Dip switch with 8 dips
- 2 PWM lines

Memory
- IC 15: 32K x 8 EPROM
- IC 13: from 8K x 8 to 32K x 8 RAM; EEPROM
- IC 12: from 8K x 8 to 32K x 8 RAM; EEPROM; EPROM
- IC 24: serial EEPROM from 256 bytes to 2048 bytes
- IC 25: 256 bytes of serial RAM+RTC

CPU
- PHILIPS 80C552 22.1184 or 29.4912 MHz

A/D conversion resolution
- 10 bits

A/D conversion time
- 27.126 µs at 22.1184 MHz
- 20.345 µs at 29.4912 MHz

PHYSICAL FEATURES

Size
- 100 x 195 mm

Weight
- 570 g.

Connectors
- CN1: 20 pins, male, vertical, low profile connector
- CN2: 20 pins, male, vertical, low profile connector
- CN3: 2 pins, quick release, screw terminal
- CN4: 20 pins, male, vertical, low profile connector
- CN5: 6 pins screw terminal connector
- CN6: 20 pins, male, vertical, low profile connector
- CN7: 9 pins female D connector
- CN8: 2 pins, male, vertical, low profile connector

Temperature range
- 10 ÷ 40 °C

Relative humidity
- 20% ÷ 90% (without condense)
## ELECTRIC FEATURES

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuse F2</td>
<td>100 mA; 250 V, fast type</td>
</tr>
<tr>
<td>Power supply tension</td>
<td></td>
</tr>
<tr>
<td>230 Vac; 50 Hz (mains power supply)</td>
<td></td>
</tr>
<tr>
<td>8÷26 Vac</td>
<td>(switching power supply)</td>
</tr>
<tr>
<td>6÷10 Vac</td>
<td>(linear power supply) *</td>
</tr>
<tr>
<td>+5 Vdc</td>
<td>(no power supply)</td>
</tr>
<tr>
<td>Consumption on 5 Vdc</td>
<td>100÷140 mA</td>
</tr>
<tr>
<td>Current supplied on +5 Vdc</td>
<td>450÷500 mA (mains) *</td>
</tr>
<tr>
<td>for external loads</td>
<td>260÷300 mA (switching) *</td>
</tr>
<tr>
<td></td>
<td>860÷900 mA (linear) *</td>
</tr>
<tr>
<td>External Back-Up battery</td>
<td>3.6÷5 Vdc 15 µA back-up current</td>
</tr>
<tr>
<td>Analog inputs voltage</td>
<td>0÷2.49 Vdc or 0÷5 Vdc</td>
</tr>
<tr>
<td>Analog inputs current</td>
<td>0÷20 mA</td>
</tr>
</tbody>
</table>

* These datas have been measured at 20 centigrad degrees of ambient temperature (for further informations please see the paragraph "POWER SUPPLY VOLTAGE").
INSTALLATION

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

CONNECTIONS

The GPC® 552 module has eight connectors that can be linked to other devices or directly to the field, according to system requirements. In this paragraph there are connectors pin out, a brief signals description (including the signal direction), connectors location (see figure 15) and some electrical diagrams that show the on board circuit of each connector.

CN3 - CONNECTOR FOR POWER SUPPLY

CN3 is a 2 pins, quick release, screw terminal connector where are available the two power supply signals for the card. CN3 must always be used for power supply, independently from the selected tension range.

![Figure 3: CN3 - Power supply connector]

Signals description:

230 Vac / +5 Vdc / 6÷10 Vac / 8÷26 Vac =

- I - Lines for 230 Vac mains power supply.
- I - Lines for +5 Vdc power supply.
- I - Lines for 6÷10 Vac linear power supply.
- I - Lines for 8÷26 Vac switching power supply.

GND = - Ground signal.
CN1 - CONNECTOR FOR PPI 82C55 PORT B, PWM, GENERAL OUTPUT

CN1 is a 20 pins, male, vertical, low profile connector with 2.54 mm pitch. On CN1 connector are available PPI 82C55 port B (equal to 8 I/O digital lines), 2 PWM outputs and 4 general TTL outputs; all CN1 signals follow TTL standard.


Signals description:

| PPI PB.n | = | I/O  | - Digital line n of PPI 82C55 port B  |
| PWM0     | = | O    | - PWM line n. 0 from CPU  |
| PWM1     | = | O    | - PWM line n.1 from CPU  |
| OUT n    | = | O    | - General digital output n.  |
| Vdc      | = |      | - Line connected to +5 Vdc power supply  |
| GND      | = |      | - Digital ground signal  |
| N.C.     | = |      | - Not connected  |
CN2 - CONNECTOR FOR PPI 82C55 PORT A AND C

CN2 is a 20 pins, male, vertical, low profile connector with 2.54 mm pitch. On CN2 connector are available PPI 82C55 port A and C equal to 16 I/O digital lines; all these signals follow TTL standard.

**Figure 5: CN2 - I/O Connector for PPI 82C55 Port A and C**

Signals description:

- **PPI PA.n** = I/O - Digital line n of PPI 82C55 port A
- **PPI PC.n** = I/O - Digital line n of PPI 82C55 port C
- **Vdc** = Line connected to +5 Vdc power supply
- **GND** = Digital ground signal
- **N.C.** = Not connected
CN4 - CONNECTOR FOR CPU PORT 5, A/D CONVERTER INPUTS

CN4 is a 20 pins, male, vertical, low profile connector with 2.54 mm pitch. On CN4 connector is available microprocessor port 5; this port can be connected either to digital TTL inputs or to analog signals for A/D converter section, in fact it has a double functionality. Between CN4 and port 5 lines there is a low frequency filter as described in figure 7.

**Figure 6: CN 4- Connector for CPU Port 5, A/D converter inputs**

<table>
<thead>
<tr>
<th>Vcc</th>
<th></th>
<th></th>
<th>1</th>
<th>2</th>
<th></th>
<th></th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>GND</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>
<td></td>
</tr>
<tr>
<td>AGND</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>
<td></td>
</tr>
<tr>
<td>AGND</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>
<td></td>
</tr>
<tr>
<td>AGND</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>
<td></td>
</tr>
<tr>
<td>AGND</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>
<td></td>
</tr>
</tbody>
</table>

Signals description:

- **STADC** = I - Digital line for hardware start to A/D conversion
- **P5.n/ADCn** = I - Digital line n or A/D channels n of CPU port 5
- **Vdc** = - Line connected to +5 Vdc power supply
- **GND** = - Digital ground signal
- **AGND** = - Analog ground signal
**Figure 7: A/D Converter Inputs Diagram**

- Analog VRef.
- VProt.
- CPU 80C552
- CN4
- Analog GND
- FILTER

- Connections and labels for each pin as shown in the diagram.
CN5 - CONNECTOR FOR I²C BUS AND RS 232 SECOND SERIAL PORT

CN5 is a 6 pins screw terminal connector
CN5 allows to communicate to any other device supporting the standard I²C bus, in addition it allows the connection to the signals of the software second serial port.

**Figure 8: CN5 - Screw terminal connector for the I²C bus and the second serial port**

Signals description:

<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tx2 - Transmittion signal of the software RS 232 second serial port.</td>
</tr>
<tr>
<td>2</td>
<td>Rx2 - Reception signal of the software RS 232 second serial port.</td>
</tr>
<tr>
<td>3</td>
<td>SDA - DATA signal of I²C bus.</td>
</tr>
<tr>
<td>4</td>
<td>SCL - CLOCK signal of I²C bus.</td>
</tr>
<tr>
<td>5</td>
<td>GND - Ground signal.</td>
</tr>
<tr>
<td>6</td>
<td>+5 Vdc - +5 Vdc power supply.</td>
</tr>
</tbody>
</table>
CN8 - CONNECTOR FOR EXTERNAL BACK UP BATTERY

CN8 is a 2 pins, male, vertical, low profile connector with 2.54 mm pitch. CN8 allows the connection of the external back up battery, which grants the integrity of on-board RAMs data and the correct working of the real time clock also when power supply is missing (for further informations please refer to the paragraph "BACK UP").

![Diagram of CN8 connector](image)

**Figure 9: CN8 - Connector for external back up battery**

Signals description:

- **+Vbat**: Positive terminal of external back up battery
- **GND**: Negative terminal of external back up battery
CN6 - CONNECTOR FOR MICROPROCESSOR I/O

CN6 is a 20 pins, male, vertical, low profile connector with 2.54 pitch. On CN4 are available microprocessor ports 1, 3 and 4, equal to 16 I/O digital lines; all these signals follow TTL standard.

Signals description:

- **P1.n** = I/O - Digital line n of microprocessor port 1
- **P3.n** = I/O - Digital line n of microprocessor port 3
- **P4.n** = I/O - Digital line n of microprocessor port 4
- **Vdc** = - Lines connected to +5Vdc power supply
- **GND** = - Digital ground signal
- **N.C.** = - Not connected

**Figure 10: CN6 - Connector for microprocessor I/O**
**Figure 11: I/O Connection Diagram**

- **PPI 82C55**
  - Port A
  - Port C
  - Port B
  - CN2
  - Pin 1–8
  - Pin 9–16

- **General Output**
  - 4 lines
  - Pin 1–8
  - Pin 9–12
  - Pin 15–16

- **80C552**
  - Pin 4–5
  - Port 3.2
  - Port 3.4
  - Port 4
  - Port 1.0–5
  - CN1
  - CN6
  - Pin 9
  - Pin 10
  - Pin 1–8
  - Pin 11–16
  - +5 Vdc
  - 2 lines
  - 8 lines
  - 6 lines

---

**GPC® 552**  Rel. 5.20  Page 19
CN7 - CONNECTOR FOR SERIAL COMMUNICATION

CN7 is a 9 pins female D connector where are available RS 232, RS 422, RS 485 and Current loop communication lines. The RS 232 signals are pined out to be compatible with P.C. connectors, while the remaining signals are placed in order to reduce interference and electrical noise.

Figura 12: CN7-Connector for serial communication

Signals description:

<table>
<thead>
<tr>
<th>Signal</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>RX-</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX+</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX-</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX+</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>RxD RS 232</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TxD RS 232</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX- Curr. Loop</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RX+ Curr. Loop</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX- Curr. Loop</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX+ Curr. Loop</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GND</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- RX- = Receive Data Negative for RS 422-485
- RX+ = Receive Data Positive for RS 422-485
- TX- = Transmit Data Negative for RS 422-485
- TX+ = Transmit Data Positive for RS 422-485
- RxD RS 232 = Receive Data: for RS 232
- TxD RS 232 = Transmit Data for RS 232
- RX- Curr. Loop = Receive Data Negative for Current Loop
- RX+ Curr. Loop = Receive Data Positive for Current Loop
- TX- Curr. Loop = Transmit Data Negative for Current Loop
- TX+ Curr. Loop = Transmit Data Positive for Current Loop
- GND = Digital ground signal
Figure 13: Serial communication diagram
LEDS

On GPC® 552 there are two LEDs that show some of the card status information, as described in the following table:

<table>
<thead>
<tr>
<th>LEDs</th>
<th>COLOUR</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD1</td>
<td>Red</td>
<td>Power supply indicator: it shows the presence of +5Vdc tension</td>
</tr>
<tr>
<td>LD2</td>
<td>Green</td>
<td>Activity LED driven by software</td>
</tr>
</tbody>
</table>

**Figure 14: LEDs table**

The main function of these LEDs is to inform the User about card status, with a simple visual indication and in addition to this, LEDs make easier the debug and test operations of the complete system.

To recognize the LEDs location on the card, please refer to figure 15.
FIGURE 15: LEDs, CONNECTORS, DIP SWITCH, ETC. LOCATION
I/O CONNECTION

To prevent possible connecting problems between GPC® 552 and the external systems, the User has to read carefully the previous paragraph information and he must follow these instructions:

- For RS 232, RS 422, RS 485 and current loop signals the User must follow the standard rules of each one of these protocols;
- For all TTL signals the User must follow the rules of this electric standard. The connected digital signal must be always reported to card digital ground. For TTL signals, the 0V level corresponds to logic state 0, while 5V level corresponds to logic state 1.
- Inputs of A/D section must be connected to low impedance signal sources compliant to the range 0±2.49 Vdc, 0±+5 Vdc, or 0±20 mA according to the chosen configuration. For further informations please refer to the paragraph "A/D CONVERTER".

RESET KEY

P1 reset key of GPC® 552 allows the activation of /RESET card signal. Pressing P1 key, the card restarts execution of the program saved in EPROM and all the on board peripheral devices are reset at the same time. P1 key is commonly used to exit from endless loop, especially during debug phase. To recognize reset key location on GPC® 552, please refer to figure 15.

TEST POINT

The board is provided with a test point called TP1, which allows, using a galvanically coupled multimeter, to measure the reference voltage Vref=2.4900 (default) or Vref=5.0000 V (if requested). The test point is made of two contacts reporting these signals:

- pin 1 -> Vref
- pin 2 -> GND

To easily locate the test point on the board, please refer to figure 15, while for further informations about the Vref signal please refer to the paragraphs "ANALOG INPUT TYPE SELECTION" and "TRIMMERS AND CALIBRATION";

ON BOARD INPUT

GPC® 552 card is provided of one 8 ways dip switch (DSW1), that can be read by software, normally used for system configuration (operating mode selection, card number programmation inside a network system, firmware configuration, etc.). Reading the dip switch register by software, the User obtain a negated 8 bits combination, in fact "ON" position corresponds to 0 logic state and "OFF" position corresponds to 1 logic state. The DSW1 register is allocated in the microprocessor addressing space by the control logic as described in the paragraph "I/O ADDRESSES". To recognize DSW1 location on GPC® 552, please refer to figure 15.
POWER SUPPLY SELECTION

GPC® 552 is supplied with an efficient power supply circuit that makes the card connectable to any standard industrial source like mains, transformer, battery, solar cell, etc. The available power supply types are:

- Mains power supply: in this configuration the board is supplied by the mains 230 Vac that must be provided on pins 1 and 2 of CN3.
- No power supply: in this configuration the board is supplied by a +5 Vdc that must be supplied on pins 1 and 2 of CN3.
- Linear power supply: in this configuration the board is supplied by a 6÷10 Vac (or by a corresponding direct current, for example 12 Vdc) that must be supplied on pins 1 and 2 of CN3.
- Switching power supply: in this configuration the board is supplied by a 8÷26 Vac (or by a corresponding direct current) that must be supplied on pins 1 and 2 of CN3.

An efficient protection circuitry is always present on the board to protect it against overvoltages or noise. We would remark that the selection of the power supply type must be made in the order; in fact a different power supply section implies a different hardware configuration that must be performed by specialized grifo® technician.

ANALOG INPUT TYPE SELECTION

The GPC® 552 board may have voltage and/or current analog inputs, as described in the previous paragraphs and chapters. The selection of the input type is made in the order, the full range of the analog inputs can be +2.490 V or +5 V. The voltage/current selection is performed by mounting a conversion module based on simple voltage drop resistances. The channel - resistance correspondence is:

\[
\begin{align*}
\text{R9} &\rightarrow \text{channel 0} \\
\text{R12} &\rightarrow \text{channel 1} \\
\text{R14} &\rightarrow \text{channel 2} \\
\text{R16} &\rightarrow \text{channel 3} \\
\text{R18} &\rightarrow \text{channel 4} \\
\text{R20} &\rightarrow \text{channel 5} \\
\text{R22} &\rightarrow \text{channel 6} \\
\text{R24} &\rightarrow \text{channel 7}
\end{align*}
\]

In the (default) case that the module is not installed the corresponding channel accepts a voltage input in the range 0÷2.49 V (default) or 0÷5 V (to be specified in the order); otherwise the input is a current. The value of the resistance on the conversion module is:

\[
R = +2.49 \text{ V} / I_{\text{max}} \quad \text{or} \quad R = +5 \text{ V} / I_{\text{max}}
\]

The module is usually based on 124 Ω or 248 Ω precision resistances, allowing 4÷20 mA or 0÷20 mA input currents. To easily locate the module please refer to figure 15.
TRIMMERS AND CALIBRATION

On GPC® 552 a trimmer is used to set the voltage reference for the optional A/D conversion section. The trimmer is labelled RV1, to easily locate please refer to figure 15.

The GPC® 552 is subjected to a careful test that verifies and calibrates all the card sections. The calibration is done in laboratory at +20 °C following this steps:

- The A/D voltage reference (Vref) is calibrated through RV1 trimmer by using a 5 digits tester to a value of +2,4900 Vdc.

- The correspondence between the analog input signal and the A/D combination is verified. The verification is done with a reference signal on the A/D input and testing that the A/D combination and the theoretical combination differ at maximum of the A/D section errors sum.

- The trimmer is blocked with paint.

The analog interfaces use high precision components that are selected during mounting phase to avoid complicate and long calibration procedures. After the calibration, all the on board trimmer are blocked with paint to warrant calibration also in presence of mechanics stresses (vibrations, movements, etc.). The user must not modify the card calibration, but if thermic drifts, time drifts, etc require it, he must follow the previous described procedure.

LOCAL USER INTERFACES

With CN2 (standard I/O Abaco® connector) the GPC® 552 can be connected to some of the numerous grifo® boards modules that have the same pin_out. It is really important the capability of directly connect a series of boards such as QTP 24P, KDL x24, KDF 224, etc. that are useful to solve local user interfaces problems. These boards already have all the resources (alphanumeric displays, matrix keyboards, LEDs etc) necessary to solve the common man-machine communication problems at a short distance from GPC® 552. For hardware installation these locals user-interfaces requires only a 20 ways flat cable (with power supply too) while for software the programmer can use the relative procedure contained in all the GPC® 552 software tools. These procedures normally are drivers software added to the language and they use directly its console instructions (for example INPUT and PRINT for BASIC, PRINTF and SCANF for C etc.), so for the user is very simple to write on displays and to get data from keyboards.

For further informations please refer to the paragraph "EXTERNAL DEVICES" and to the documentation of the software packages.
JUMPERS

On GPC® 552 there are 22 jumpers for card configuration. Connecting these jumpers, the User can define for example the memory type and size, the peripheral devices functionality, the serial communication interface and so on. To easily locate the jumpers please refer to figure 18. Here below is the jumpers list, location and function:

<table>
<thead>
<tr>
<th>JUMPERS</th>
<th>PIN N.</th>
<th>USE</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1</td>
<td>2</td>
<td>Double Selects IC 15 size between 8, 16, 32 KBytes</td>
</tr>
<tr>
<td>J2</td>
<td>2</td>
<td>Triple Selects IC 15 size between 16, 24, 32 KBytes</td>
</tr>
<tr>
<td>J3</td>
<td>3</td>
<td>Configures IC12 for 8 or 32 KBytes RAM/EEPROM/EPROM</td>
</tr>
<tr>
<td>J4</td>
<td>3</td>
<td>Configures IC13 for 8 or 32 KBytes RAM/EEPROM</td>
</tr>
<tr>
<td>J5</td>
<td>2</td>
<td>Enables/Disables optional microprocessor internal ROM</td>
</tr>
<tr>
<td>J6</td>
<td>2 Double</td>
<td>Manages PULL-UP resistors of SCL and SDA lines</td>
</tr>
<tr>
<td>J7</td>
<td>2</td>
<td>Connects to Vdc pin 5 of CN 5</td>
</tr>
<tr>
<td>J8, J9</td>
<td>2</td>
<td>They connect termination and force circuit to RS 422-485 serial line.</td>
</tr>
<tr>
<td>J10</td>
<td>4</td>
<td>Selects the signal to connect to the CPU INT0 pin</td>
</tr>
<tr>
<td>J11</td>
<td>2</td>
<td>WATCH-DOG hardware enable.</td>
</tr>
<tr>
<td>J13</td>
<td>5</td>
<td>Selects direction and operating mode for RS 422-485</td>
</tr>
<tr>
<td>J14</td>
<td>3 Triple</td>
<td>Selects memory configuration between BASIC and ASSEMBLER mode</td>
</tr>
<tr>
<td>J15</td>
<td>2</td>
<td>Configures back up circuitry for RAM and RTC</td>
</tr>
<tr>
<td>J16</td>
<td>3</td>
<td>Selects the signal to connect to the CPU T0 pin</td>
</tr>
<tr>
<td>J17</td>
<td>3</td>
<td>Selects size and memory device type for IC 12</td>
</tr>
<tr>
<td>J18</td>
<td>3</td>
<td>Selects memory device type for IC 12</td>
</tr>
<tr>
<td>J19, J20</td>
<td>2</td>
<td>Connect the primary serial port RS 232 driver to CN7</td>
</tr>
<tr>
<td>J21</td>
<td>2</td>
<td>Sets the primary serial port in RS232 mode</td>
</tr>
<tr>
<td>J22</td>
<td>3</td>
<td>Enables the use of special memory mapping for future uses</td>
</tr>
<tr>
<td>J23</td>
<td>3</td>
<td>Manages the supply of the resistor networks used by the CPU PORT 4 and 1 and the 82C55 PORT A, B e C</td>
</tr>
</tbody>
</table>

**FIGURE 16: JUMPERS SUMMARIZING TABLE**
## 2 PINS JUMPER TABLE

<table>
<thead>
<tr>
<th>JUMPERS</th>
<th>CONNECTION</th>
<th>FUNCTION</th>
<th>DEF.</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1</td>
<td>position 1</td>
<td>It sets size of IC 15 to 32 KBytes</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>position 2</td>
<td>It sets size of IC 15 to 16 KBytes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>position 3</td>
<td>It sets size of IC 15 to 8 KBytes</td>
<td></td>
</tr>
<tr>
<td>J2</td>
<td>position 1</td>
<td>It sets size of IC 13 to 32 KBytes</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>position 2</td>
<td>It sets size of IC 13 to 16 KBytes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>position 3</td>
<td>It sets size of IC 13 to 24 KBytes</td>
<td></td>
</tr>
<tr>
<td>J5</td>
<td>non connesso</td>
<td>Enables microprocessor internal ROM (cut the line on solder side)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>connected</td>
<td>Enables external ROM</td>
<td>*</td>
</tr>
<tr>
<td>J6</td>
<td>position 1</td>
<td>Pull-up resistor disconnected from SCL signal</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>position 2</td>
<td>Pull-up resistor connected to SCL signal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>position 3</td>
<td>Pull-up resistor disconnected from SDA signal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>position 2</td>
<td>Pull-up resistor connected to SDA signal</td>
<td>*</td>
</tr>
<tr>
<td>J7</td>
<td>not connected</td>
<td>Pin 5 of CN 5 open</td>
<td></td>
</tr>
<tr>
<td></td>
<td>connected</td>
<td>Pin 5 of CN 5 connected to Vdc</td>
<td>*</td>
</tr>
<tr>
<td>J8, J9</td>
<td>not connected</td>
<td>They disconnect the termination and force circuit from the RS 422-485 serial line</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>connected</td>
<td>They connect the termination and force circuit to the RS 422-485 serial line</td>
<td></td>
</tr>
<tr>
<td>J11</td>
<td>non connesso</td>
<td>Watch-dog hardware disabilitation</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>connected</td>
<td>Watch-dog always enabled</td>
<td></td>
</tr>
<tr>
<td>J15</td>
<td>not connected</td>
<td>RAM on IC 13 backed up only by an eventual external battery. RAM+RTC on IC 25 backed up only by on board battery BT1</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>connected</td>
<td>RAM on IC 13 backed up by an eventual external battery an by the optional on board battery BT1. RAM+RTC on IC 25 backed up by on board battery BT1 and by an eventual external battery.</td>
<td></td>
</tr>
<tr>
<td>J19, J20</td>
<td>not connected</td>
<td>They disconnect the primary RS 232 serial driver from CN7</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>connected</td>
<td>They sconnect the primary RS 232 serial driver to CN7</td>
<td></td>
</tr>
<tr>
<td>J21</td>
<td>not connected</td>
<td>Disables primary RS232 serial line</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>connected</td>
<td>Enables primary RS232 serial line</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 17: 2 Pins Jumper Table*

The * denotes the default connection for the board.
FIGURE 18: JUMPERS LOCATION
## 3 PINS JUMPER

<table>
<thead>
<tr>
<th>JUMPER</th>
<th>CONNECTION</th>
<th>USE</th>
<th>DEF.</th>
</tr>
</thead>
<tbody>
<tr>
<td>J3</td>
<td>position 1-2</td>
<td>Configures IC12 for 32 KBytes RAM/EEPROM/EPROM</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>position 2-3</td>
<td>Configures IC12 for 8KBytes RAM/EEPROM/EPROM</td>
<td></td>
</tr>
<tr>
<td>J4</td>
<td>position 1-2</td>
<td>Configures IC13 for 32 KBytes RAM/EEPROM</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>position 2-3</td>
<td>Configures IC13 for 8 KBytes RAM/EEPROM</td>
<td></td>
</tr>
<tr>
<td>J14 (1,2,3)</td>
<td>BASIC position</td>
<td>Configures the board for BASIC software tools</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>ASM position</td>
<td>Configures the board for ASSEMBLER software tools</td>
<td></td>
</tr>
<tr>
<td>J16</td>
<td>position 1-2</td>
<td>Connects CPU T0 signal to Tx2 signal of the software RS 232 serial line (second serial)</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>position 2-3</td>
<td>Connects CPU T0 signal to pin 10 of CN 6</td>
<td></td>
</tr>
<tr>
<td>J17</td>
<td>position 1-2</td>
<td>Configures IC12 for EPROM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>position 2-3</td>
<td>Configures IC12 for 32 K RAM/EEPROM</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>Not connected</td>
<td>Configures IC12 for 8 K RAM/EEPROM</td>
<td></td>
</tr>
<tr>
<td>J18</td>
<td>position 1-2</td>
<td>Configures IC12 for EPROM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>position 2-3</td>
<td>Configures IC12 for RAM/EEPROM</td>
<td>*</td>
</tr>
<tr>
<td>J22</td>
<td>position 1-2</td>
<td>Memory configurations for future use</td>
<td></td>
</tr>
<tr>
<td></td>
<td>position 2-3</td>
<td>Keep the memory configurations already defined</td>
<td>*</td>
</tr>
<tr>
<td>J23</td>
<td>position 1-2</td>
<td>Keeps pull up connected on the CPU PORT 4 and 1 and the 82C55 PORT A, B and C signals</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>position 2-3</td>
<td>Software management of pull up or pull down on the CPU PORT 4 and 1 and the 82C55 PORT A, B and C signals</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 19: 3 PINS JUMPERS table**

The "*" 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.
4 PINS JUMPER

<table>
<thead>
<tr>
<th>JUMPERS</th>
<th>CONNECTION</th>
<th>USE</th>
<th>DEF.</th>
</tr>
</thead>
<tbody>
<tr>
<td>J10</td>
<td>position 1-2</td>
<td>Connects the CPU INT0 signal to the Rx2 of the software RS 232 serial line (second serial)</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>position 2-3</td>
<td>Connects the CPU INT0 signal to pin 9 of CN6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>position 2-4</td>
<td>Connects the CPU INT0 signal to pin 7 of IC25 (Real Time Clock Interrupt)</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 20: 4 pins jumper table**

The "*" 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.

5 PINS JUMPER

<table>
<thead>
<tr>
<th>JUMPERS</th>
<th>CONNECTION</th>
<th>USE</th>
<th>DEF.</th>
</tr>
</thead>
<tbody>
<tr>
<td>J13</td>
<td>position 1-2 and 3-4</td>
<td>Selects serial communication in RS 485 (2 wires half duplex)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>position 2-3 and 4-5</td>
<td>Selects serial communication in RS 422-485 (4 wires half or full duplex)</td>
<td>*</td>
</tr>
</tbody>
</table>

**Figure 21: 5 pins jumper table**

The "*" 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.

**NOTE**

Here follow more detailed instructions on how to configure the board.

**PULL UP / DOWN TYPE AND SUPPLY SELECTION**

As shown in figure 19, jumper J 23 may be connected in two positions:

- Position 1-2: The pull up on the CPU PORT 4 and 1 and the 82C55 PORT A, B and C is always connected. This means that whenever a power on or a reset occur the outputs will be set to a logic state "1" with consequent activation of the NPN devices there connected. Be careful.

- Position 2-3: Software management of the pull up on the CPU PORT 4 and 1 and the 82C55 PORT A, B and C. The logic state "0" on these outputs is assured whenever a power on or a reset occur, it is so avoided the activation of the NPN devices there connected. Next the User will be able to set the initial status of the outputs and to connect the pull up by acting on the EPUL register setting D0=1 (please refer to the paragraph "PULL UP/DOWN SELECTION ON I/O SIGNALS").
SERIAL COMMUNICATION SELECTION

The serial line A is available on connector CN3A and can be buffered in RS 232, RS 422, RS 485 or current loop electric standard. By hardware can be selected which one of these electric standard is used, through jumpers connection. By software the serial line can be programmed to operate with the desired physical communication protocol acting on the CPU internal registers. In the following paragraphs there are all the informations on serial communication configurations; please note that jumpers which are not mentioned below do not affect the serial communication whatever their configuration is.

- RS 232 SERIAL LINE
  MAX 232 serial driver must be installed on IC23, while on IC19, IC20, IC21, IC22 no driver must be installed. Jumpers J19, J20 and J21 must be connected, jumpers J8, J9, J13 are not important.

- CURRENT LOOP SERIAL LINE
  HCPL 4100 must be installed n IC21, HCPL4200 must be installed on IC22, while on IC19 and IC20 no driver must be installed. Jumpers J19, J20 and J21 must be not connected, jumpers J8, J9, J13 are not important.

- RS 485 SERIAL LINE
  SN75176 serial driver must be installed on IC20, while no driver must be installed on IC21 and IC22. Jumper J13 must be connected in position 1-2 and 3-4, jumpers J19, J29, J21 must be not connected. With DIR signal, the user can select by software the line direction, pins 1 and 2 of CN7 become transmission or reception lines, according the status of DIR signal (0=low= reception, 1=high=transmission). This kind of serial communication can be used for multi-point connections, in addition it is possible to listen to own transmission, so the user is allowed to verify the success of transmission. In fact, any conflict on the line can be recognized by testing the received character after each transmission.

- RS 422 SERIAL LINE
  SN75176 serial drivers must be installed on IC20 and IC19 while no driver must be installed on IC21 and IC22. Jumper J13 must be connected in position 2-3 and 4-5, jumpers J19, J29, J21 must be not connected. DIR signal can be kept always high (active transmittre) for point-to-point connections, while for multi-point connections the transmitter must be activated only before the transmission (DIR =1=high=transmitter activated).

If using the RS 422-485 serial line, it is possible to connect the terminating and forcing circuit on the line by using J8 and J9. This circuit must be always connected in case of point-to-point connections, while in case of multi-point connections it must be connected only in the farthest boards, that is on the edges of the communication line.
MEMORY SELECTION

On GPC® 552 can be mounted 98K and 256 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 CONFIGURATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>RAM/EEPROM</td>
<td>8K Bytes</td>
<td>J18 in position 2-3; J17 not connected; J3 in position 2-3</td>
</tr>
<tr>
<td></td>
<td>RAM/EEPROM</td>
<td>32K Bytes</td>
<td>J18 in position 2-3; J17 in position 2-3; J3 in position 1-2</td>
</tr>
<tr>
<td></td>
<td>EPROM</td>
<td>8K Bytes</td>
<td>J18 in position 1-2; J17 in position 1-2; J3 not connected</td>
</tr>
<tr>
<td></td>
<td>EPROM</td>
<td>32K Bytes</td>
<td>J18 in position 1-2; J17 in position 1-2; J3 in position 1-2</td>
</tr>
<tr>
<td>13</td>
<td>RAM/EEPROM</td>
<td>8K Bytes</td>
<td>J4 in position 2-3</td>
</tr>
<tr>
<td></td>
<td>RAM/EEPROM</td>
<td>32K Bytes</td>
<td>J4 in position 1-2</td>
</tr>
<tr>
<td>15</td>
<td>EPROM</td>
<td>32K Bytes</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>EEPROM</td>
<td>512± 2048 Bytes</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>RAM+RTC</td>
<td>256 Bytes</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 22: Memory selection table**

The sockets described above follow the JEDEC standard, so the mounted memory devices must have JEDEC pin outs; exceptions are the serial EEPROM on IC24 and the serial RAM+RTC on IC25 devices that must be specified at the moment of the order and can be mounted only by grifo® technician. For further information about the memory devices please refer to the manufacturer documentation. We would want to remark that in case the IC25 component is mounted, the maximum size of IC24 is 1024 bytes.

RAM modules on IC12 and IC13 can be backed on request.

BACK UP

On GPC® 552 can be mounted a lithium battery that keeps data on IC 25 RAM components and keeps the Real Time Clock counting, also when power supply is missed. This lithium battery and RAM+RTC on IC25 components are available on the board only if specified in the card order.

- If the User needs to back up RAM on IC13 an external battery must be connected on CN8. Acting on jumper J15 can be backed also the eventual RAM+RTC on IC25, saving battery BT1.

- The eventual on board battery BT1 can be used to back up also RAM on IC13 but only for short time intervals, like, for example, the time needed to replace the external battery.

No other working condition has been prevented and so cannot be performed.
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. Generally all software packages available for the mounted microprocessor, or for the 51 family, can be used. For example:

**MDP**: monitor debugger program able to load and debug each HEX Intel files for 51 microprocessor family. It is provided of the standard commands available on in circuit emulator and requires only an external P.C. connected through a serial line.

**FORTH**: complete software development tools to program the card with FORTH high level language. It needs a P.C. for User interface and it is really interesting for its fast execution and small size, of the generated code.

**DESIGN 51**: in circuit emulator for 8051, 80C51, 8031, 80C31, 8032, 80C32, 8751, 80C451, 80C552, 80C562, 80C652, 87C751, 87C752. It is a powerful hardware and software tools that includes a Symbolic Debugger and a Cross Assembler, with a low price.

**MCS BASIC 552**: complete development tools for MCS BASIC (interpreted BASIC language for industrial application). It needs a P.C. for console and program saving operations, while the debug, test and program operations are performed on the card. Special instructions which manage the on board peripheral devices have been added.

**BXC51**: cross compiler for source files written in MCS BASIC 552. It allows a considerable speed increment of the starting MCS BASIC program and it is completely executed on external P.C.

**OS552**: powerful macro cross assembler composed by communication program, editor, macro assembler, symbolic linker and a source remote debugger.

**µ-BASIC-51**: floating point BASIC cross compiler for 51 family, that works in together with OS552.

**MICRO/ASM-51**: macro cross assembler available for MS-DOS operating system in absolute and relocatable version. In this relocatable version is supplied with a linker and a library manager.

**MICRO/C-51**: integer cross compiler for source files in standard ANSI C. It produces a source assembly file compatible with MICRO/ASM 51 or with Intel macro relocatable assembler MCS 51.

**MICRO/SLD-51**: source level debugger and simulator. It is executed on P.C. without any additional hardware and it allows loading of HEX and SYMBOLIC files, breakpoint setting, instruction execution in trace mode, registers and memory dump, etc.

**HI-TECH C**: cross compiler for C source program. It is a powerful software tool that includes editor, C compiler, assembler, optimizer, linker, library, and remote symbolic debugger, in one easy to use integrated development environment.

**XPAS51**: cross compiler for PASCAL source program, executable on P.C. with MS-DOS operating system.
HARDWARE

INTRODUCTION

In this chapter are reported all information about card use, related to hardware features of GPC® 552. For example the registers addresses, the memory allocation and peripheral devices software management are described below.

ADDRESSES

The card 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, in two separate manners. The 80C552 microprocessor addresses 64K bytes of code memory and 64K bytes of data memory and the control logic provides on board memory and peripheral devices allocation inside these addresses space. Control logic sets size, type and addresses of memory device through jumpers J1, J2, J3, J4 and J14, while it sets I/O addresses always in the upper 8 bytes of microprocessor memory. Summarizing the control logic allocates:

- Up to 32K bytes of EPROM on IC 15
- Up to 32K bytes of RAM, EEPROM on IC 13 (without last 8 bytes used for I/O)
- Up to 32K bytes of RAM, EEPROM, EPROM on IC 12 (without last 8 bytes used for I/O)
- PPI 82C55
- DSW1 dip switch
- EPUL signal (pull up/down management on TTL signals)
- DIR signal (direction in RS 422-485 communication)
- Buzzer
- Activity LED (and EMU signal for future use)
- 4 general output
- Status reading of previous five devices

The addresses of all these devices are described in the following two paragraphs and can't be set with different value.

Other devices such EEPROM on IC24 and RAM+RTC on IC25 are always managed by the control logic but don't occupy room in the addressing space because they use a synchronous serial communication based on the CPU I/O lines.
I/O ADDRESSES

I/O addresses are located in the last 8 bytes (6 used, 2 reserved for future use) of the 64K bytes microprocessor space, to avoid conflict problems. Next table shows addresses, meanings and direction of peripheral device registers (only the external ones to microprocessor):

<table>
<thead>
<tr>
<th>DEVICE</th>
<th>REG.</th>
<th>IND.</th>
<th>R/W</th>
<th>USE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPI 82C55</td>
<td>PDA</td>
<td>FFF8H</td>
<td>R/W</td>
<td>Port A data register</td>
</tr>
<tr>
<td></td>
<td>PDB</td>
<td>FFF9H</td>
<td>R/W</td>
<td>Port B data register</td>
</tr>
<tr>
<td></td>
<td>PDC</td>
<td>FFFAH</td>
<td>R/W</td>
<td>Port C data register</td>
</tr>
<tr>
<td></td>
<td>CNT</td>
<td>FFFBH</td>
<td>R/W</td>
<td>Status and command register</td>
</tr>
<tr>
<td>ACT. LED</td>
<td>LD2</td>
<td>FFFCH</td>
<td>R/W</td>
<td>Register for activity LED management</td>
</tr>
<tr>
<td>BUZZER</td>
<td>BUZ</td>
<td>FFFCH</td>
<td>R/W</td>
<td>Register for activity buzzer management</td>
</tr>
<tr>
<td>PULL UP / DOWN</td>
<td>EPUL</td>
<td>FFFCH</td>
<td>R/W</td>
<td>Pull up/down type and supply selection.</td>
</tr>
<tr>
<td>SUPPLY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRIVER RS</td>
<td>DIR</td>
<td>FFFCH</td>
<td>R/W</td>
<td>Register for RS 422-485 driver selection</td>
</tr>
<tr>
<td>422-485 DIRECT.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GENERAL OUTPUT</td>
<td>OUTG</td>
<td>FFFCH</td>
<td>R/W</td>
<td>Register for 4 general outputs management</td>
</tr>
<tr>
<td>DSW1</td>
<td>DSW1</td>
<td>FFFDH</td>
<td>R</td>
<td>Register to obtain DSW1 configuration</td>
</tr>
</tbody>
</table>

**Figure 23: I/O addresses table**

For further information about register meanings, please refer to next paragraph called "PERIPHERAL DEVICES SOFTWARE DESCRIPTION".

MEMORY ADDRESSES

On the GPC® 552 four different memory configurations can be selected. The configuration must be selected both according to used software tools and User requests and/or application features. For example, some of this typical configurations are:

- BAS 552 -> Memory configuration 4; IC 13=32K RAM; IC 12=8 or 32K EEPROM
- MDP -> Memory configuration 2; IC 13=32K RAM; IC 12=32K RAM
- HI TECH C -> Memory configuration 1; IC 13=32K RAM; IC 12=32K RAM

The following figures describe available memory configuration, with the relative jumpers setting.
MEMORY CONFIGURATION 1

**CODE and DATA AREA**

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFFFH</td>
<td>ON BOARD I/O</td>
</tr>
<tr>
<td>FFF8H</td>
<td>NOT USED</td>
</tr>
<tr>
<td>FFF7H</td>
<td>NOT USED</td>
</tr>
</tbody>
</table>

- **IC12**
  - RAM
  - EPROM
  - EEPROM

- **IC13**
  - RAM
  - EEPROM

- **IC15**
  - EPROM

**Figure 24: Mode 1 memory configuration**

**Jumpers:** J1 in position 3; J2 in position 3; J14(1,2,3) in position ASM

Used by software tools like: HI TECH C; DDS C; etc.
MEMORY CONFIGURATION 2

**CODE and DATA AREA**

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFFFH</td>
<td>ON BOARD I/O</td>
<td>FFFFH</td>
</tr>
<tr>
<td>FFF8H</td>
<td>NOT USED</td>
<td>FFFFH</td>
</tr>
<tr>
<td>FFF7H</td>
<td>NOT USED</td>
<td>FFFEH</td>
</tr>
<tr>
<td></td>
<td>DSW1</td>
<td>FFFDH</td>
</tr>
<tr>
<td></td>
<td>EPUL, DIR, BUZZER, LD2, 4 OUTPUT</td>
<td>FFFCH</td>
</tr>
<tr>
<td>8000H</td>
<td>82C55 CONTROL Register</td>
<td>FFFBH</td>
</tr>
<tr>
<td>7FFFH</td>
<td>82C55 PORT C Register</td>
<td>FFFAH</td>
</tr>
<tr>
<td></td>
<td>82C55 PORT B Register</td>
<td>FFF9H</td>
</tr>
<tr>
<td></td>
<td>82C55 PORT A Register</td>
<td>FFF8H</td>
</tr>
<tr>
<td>4000H</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3FFFH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IC12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RAM EPROM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EEPROM</td>
<td></td>
</tr>
<tr>
<td>0000H</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IC13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RAM EPROM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EEPROM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IC15</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EPROM</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 25: Mode 2 Memory Configuration**

**Jumpers:** J1 in position 2; J2 in position 2; J14(1,2,3) in position ASM

Used by software tools like: MDP; HI TECH C; DDS C; etc.
MEMORY CONFIGURATION 3

CODE and DATA AREA

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFFFFFFH</td>
<td>ON BOARD I/O</td>
</tr>
<tr>
<td>FFF8H</td>
<td>NOT USED</td>
</tr>
<tr>
<td>FFF7H</td>
<td>NOT USED</td>
</tr>
<tr>
<td>8000H</td>
<td>IC13</td>
</tr>
<tr>
<td>7FFFH</td>
<td>RAM EEPROM</td>
</tr>
<tr>
<td>0000H</td>
<td>IC15</td>
</tr>
<tr>
<td></td>
<td>EPROM</td>
</tr>
<tr>
<td>7FFFH</td>
<td>DSW1</td>
</tr>
<tr>
<td></td>
<td>EPUL, DIR, BUZZER, LD2, 4 OUTPUT</td>
</tr>
<tr>
<td>8000H</td>
<td>82C55 CONTROL Register</td>
</tr>
<tr>
<td>7FFFH</td>
<td>82C55 PORT C Register</td>
</tr>
<tr>
<td>0000H</td>
<td>82C55 PORT A Register</td>
</tr>
<tr>
<td></td>
<td>82C55 PORT B Register</td>
</tr>
</tbody>
</table>

**Figure 26: Mode 3 Memory Configuration**

**Jumpers:** J1 in position 1; J2 in position 1; J14(1,2,3) in position ASM

Used by software tools like: HI TECH C; DDS C; etc.
MEMORY CONFIGURATION 4

**Figure 27: Mode 4 Memory Configuration**

- **Code Area**
  - ON BOARD I/O
  - IC12
    - RAM
    - EPROM
    - EEPROM
- **Data Area**
  - NOT USED
  - DSW1
  - EPUL, DIR, BUZZER, LD2, 4 OUTPUT
  - 82C55 CONTROL Register
  - 82C55 PORT C Register
  - 82C55 PORT B Register
  - 82C55 PORT A Register

**Jumpers:** J1 in position 1; J2 in position 1; J14(1,2,3) in position BASIC

Used by software tools like: MCS BASIC323; BXC 51; HI TECH C; DDS C; etc.
Figure 28: Card photo
PERIPHERAL DEVICES SOFTWARE DESCRIPTION

In the previous paragraphs are described the external registers addresses, while in this one is described a specific description of registers meaning and function (please refer to figure 23, in fact in this documentation are used the registers name defined in that figure). For microprocessor internal peripheral devices, not described in this paragraph, or for further information, please refer to "APPENDIX B" of this manual or to manufacturing company documentation.

BUZZER

Buzzer is activated by performing a "write operation" with bit D2=1 at the address of register BUZ; vice versa buzzer is disabled by performing the same operation with bit D2=0. The remaining 7 bits of register BUZ must be defined according to previous setting, for avoiding modification on the other devices managed with BUZ register address (we would remark that registers called EPUL, LD2, BUZ, DIR, OUTG share the same address). As well, the previous state of buzzer can be obtained performing a "read operation" of the same register BUZ and masking D2 bit. BUZ register is reset (all bits to 0) after Reset or power on, maintaining disabled the buzzer circuit.

GENERAL OUTPUT

The management of GPC® 552 4 general outputs is performed through register OUTG. The most significant four bits of OUTG register are used to set the output with the following correspondence:

- bit D7 -> Generic output 7 on CN1 pin 10
- bit D6 -> Generic output 6 on CN1 pin 9
- bit D5 -> Generic output 5 on CN1 pin 12
- bit D4 -> Generic output 4 on CN1 pin 11

Performing a "write operation" with bit = 1, the general output is set to high level and vice versa. The remaining 4 bits of register OUTG must be defined according to previous setting, for avoiding modification on the other devices managed with OUTG register address (we would remark that registers called EPUL, LD2, BUZ, DIR, OUTG share the same address). As well, the previous state of general outputs can be obtained performing a "read operation" of the same register OUTG and masking D4÷D7 bits. OUTG register is reset (all bits to 0) after Reset or power on, maintaining to low level all the outputs.

ACTIVITY LED

Activity LED is enabled by performing a "write operation" with bit D3=1 at the address of register LD2; vice versa LED is disabled by performing the same operation with bit D3=0. The remaining 7 bits of register LD2 must be defined according to previous setting, for avoiding modification on the other devices managed with LD2 register address (we would remark that registers called EPUL, LD2, BUZ, DIR, OUTG share the same address). As well, the previous state of activity LED can be obtained performing a "read operation" of the same register LD2 and masking D3 bit. LD2 register is reset (all bits to 0) after Reset or power on, maintaining disabled the activity LED.
DIP SWITCH

The on board DSW1 dip switch state can be obtained by software, through a simple "read operation" at the DSW1 register address. The correspondence between register bits and dip switch is as follows:

- bit D7 -> DSW1.8
- bit D6 -> DSW1.7
- bit D5 -> DSW1.6
- bit D4 -> DSW1.5
- bit D3 -> DSW1.4
- bit D2 -> DSW1.3
- bit D1 -> DSW1.2
- bit D0 -> DSW1.1

Reading DSW1 register by software, the User obtains a negated 8 bits combination, in fact "ON" position corresponds to 0 logic state and "OFF" position corresponds to 1 logic state.

RS 422-485 COMMUNICATION DIRECTION

Bit D1 of DIR register sets the status of DIR signal described in paragraph "SERIAL COMMUNICATION SELECTION". Transmission is enabled by performing a "write operation" with bit D1=1 at the address of register DIR; vice versa reception is enabled by performing the same operation with bit D1=0. The remaining 7 bits of register DIR must be defined according to previous setting, for avoiding modification on the other devices managed with DIR register address (we would remark that registers called EPUL, LD2, BUZ, DIR, OUTG share the same address). As well, the previous state of RS 422-485 direction can be obtained performing a "read operation" of the same register DIR and masking D1 bit.
DIR register is reset (all bits to 0) after Reset or power on, maintaining enabled the reception and disabled the transmission.

PULL UP/DOWN SELECTION ON I/O SIGNALS

The selection of pull up / down on I/O signals is performed by an output operation, using D0, to the address of register EPUL (J23 must be in position 2-3).
More specifically:

- D0 = 0 --> PULL DOWN activated
- D0 = 1 --> PULL UP activated

The remaining 7 bits of register EPUL must be defined according to previous setting, for avoiding modification on the other devices managed with DIR register address (we would remark that registers called EPUL, LD2, BUZ, DIR, OUTG share the same address). As well, the previous state of pull up/down selection can be obtained performing a "read operation" of the same register EPUL and masking bit D0. EPUL register is reset (all bits to 0) after Reset or power on, maintaining enabled the reception and disabled the transmission.
SERIAL EEPROM

For software management of serial EEPROM module of IC 24, please refer to specific documentation or to demo programs supplied with the card. No other information is provided by this manual because the use of this component requires a deep knowledge of its management techniques, and, however, the User can take advantage of the high level routines provided with the programming tool. The first 30 bytes of serial EEPROM are reserved for software tools use, so they can't be read or written by User program.

The electric connection is:

- DATA signal (SDA) --> CPU pin 27 (P3.3)
- CLOCK signal (SCL) --> CPU pin 29 (P3.5)

We also would remark that, due to our implementation of the management circuitry of EEPROM module, signals A2, A1, A0 of slave address are respectively set to 1, 0, 0.

BACKED RAM + SERIAL RTC

For software management of serial RAM+RTC module of IC 25, please refer to specific documentation or to demo programs supplied with the card. No other information is provided by this manual because the use of this component requires a deep knowledge of its management techniques, and, however, the User can take advantage of the high level routines provided with the programming tool. We would remark that the maximum size of EEPROM must be 1024 bytes if this RAM+RTC component is installed on IC24.

The electric connection is:

- DATA signal (SDA) --> CPU pin 27 (P3.3)
- CLOCK signal (SCL) --> CPU pin 29 (P3.5)

We also would remark that, due to our implementation of the management circuitry of RAM+RTC module, signal A0 of slave address is set to 0.

PPI 82C55

This external peripheral device is managed through 4 registers: one status register (CNT) and three data registers (PDA, PDB, PDC). The data registers are available both for read operation (to obtain signal status) and for write operation (to set signal status) with the correspondence described in figure 23. The PPI 82C55 can work in three different modes:

- MODE 0 = it provides two 8 bits bidirectional ports (A, B) and two 4 bits bidirectional ports (C LOW, C HIGH); output signals are latched and input signals are not latched; no handshake signals are provided.

- MODE 1 = it provides two 12 bits ports (A+C LOW and B+C HIGH) where ports A and B are used as 8 I/O lines and port C are used as 4 handshake lines. Both inputs and outputs are latched.

- MODE 2 = it provides a 13 bits port (A+C3÷7) where port A is used as 8 I/O lines and the 5 bits of port C are used as handshake, and a 11 bits port (B+C0÷2) where port B is used as 8 I/O lines and the 3 bits of port C are used as handshakes. Both inputs and outputs are latched.
The device is programmed writing an 8 bits word in the status register CNT, with the following bit meaning:

\[
\text{CNT} = \text{SF} \quad \text{M1} \quad \text{M2} \quad \text{A} \quad \text{CH} \quad \text{M3} \quad \text{B} \quad \text{CL}
\]

where
- \( \text{SF} \) = mode Set Flag: if actived (1) the device is enabled for standard I/O operation
- \( \text{M1} \) M2 = mode selection:
  - 0 0 = mode 0
  - 0 1 = mode 1
  - 1 X = mode 2
- A = port A direction: 1=input; 0=output
- CH = port C HIGH direction: 1=input; 0=output
- M3 = mode selection: 1=mode 1; 0=mode 0
- B = port B direction: 1=input; 0=output
- CL = port C LOW direction: 1=input; 0=output

After Reset or power on PPI 82C55 is programmed in mode 0 with all three ports in input; in this way any connection of PPI 82C55 signals can be used without conflict problems.

**PWM**

Please refer to proper technical documentation described in "APPENDIX B".

**SIO**

Please refer to proper technical documentation described in "APPENDIX B".

**A/D CONVERTER**

Please refer to proper technical documentation described in "APPENDIX B".

**TIMER COUNTER**

Please refer to proper technical documentation described in "APPENDIX B".

**INTERNAL WATCH DOG**

Please refer to proper technical documentation described in "APPENDIX B".
EXTERNAL DEVICES FOR GPC® 552

GPC® 552 can be connected to a wide range of grifo® cards and to many system of other companies. Hereunder these cards are listed, for further information please call grifo®.

QTP 24 - QTP 24P
Quick Terminal Panel 24 keys with Parallel interface
Intelligent user panel equipped with Fluorescent or LCD display, LEDs backlit, 20x2 or 20x4 characters; RS 232, RS 422, RS 485 or current loop serial line; serial E2 for set up and message. Possibility of re-naming keys, LEDs and panel name by inserting label with new name into the proper slot; 24 Keys and 16 LEDs with blinking attribute and buzzer manageable by software; built in power supply; RTC option, reader of magnetic badge and relay. The QTP 24P is low cost no intelligent (passive) version. It is directly driven from 16 TTL I/O lines; high level languages supported.

QTP G26
Quick Terminal Panel - LCD Graphic, 26 keys
Intelligent user panel equipped with graphic LCD display 240x128 pixel, CFC backlit; optocoupled RS 232 line and additional RS 232, RS 422, RS 485 or current loop serial line. Independent optional CAN line controller; serial E2 for set up; RTC and RAM Lithium backed; primary graphic objects; possibility of re-naming keys, LEDs and panel name by inserting label with new name into the proper slot; 26 Keys and 16 LEDs with blinking attribute and buzzer manageable by software; built in power supply; reader of magnetic badge, smart-card and relay option.

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.

RBO 08   TBO 08
Relays or Transistor BLOCK Output
Interface for Abaco® standard I/O 20 pins connector; 8 displayed Relays 3 A with MOV or 8 optocoupled Transistors 3 A open collectors; screw terminal; Connection for DIN C Type and Ω rails.

RBO 16
Relays BLOCK Output
Interface for Abaco® standard I/O 20 pins connector; 16 displayed Relays 3 A with MOV; screw terminal; Connection for DIN C Type and Ω rails.

XBI R4  XBI T4
miXed BLOCK Input-Output
Interface for Abaco® standard I/O 20 pins connector; 4 Relays 3 A with MOV or 4 optocoupled Transistors 3 A open collectors; 4 input lines optocoupled; I/O lines dispayed; screw terminal; Connection for DIN C Type and Ω rails.
**FBC xxx**
Flat BLOCK Contact
This interconnection system “wires to board” allows the connection to many types of flat cable connectors to a terminal for external connections.
Other interfacing for most popular connectors such as D, mini DIN, ACCESS.bus™, and so on, are available. Connection for DIN C Type and Ω rails.

**IPC 51**
Intelligent Peripheral Controller
This Intelligent peripheral card acquires 8 temperature sensors PT 100 type or thermo couple J, K, S, T type; BUS interfacing or through RS 232, RS 422-485 or Current Loop line; 16 Bits + sign A/D section; 5 or 8 conversion per second; 0,1 °C resolution.

**UAR 24**
Universal Analog Regulator
This Intelligent peripheral card acquires 2 temperature sensors PT 100 type or 2 thermo couple J, K, S, T type; 4 3 A relays output; 2 D/A outputs 12 bits 0÷10 Vdc each; BUS interfacing or through RS 232, RS 422-485 or Current Loop line; 16 Bits + sign A/D section

**QTP 22**
Quick Terminal Panel 22 keys
Intelligent user panel equipped with alphanumeric LCD or fluorescent display (40x1, 40x2 or 40x4 characters); RS 232, RS 422-485 or Current Loop serial lines; serial EEPROM for set-up and messages; Possibility of re-naming the 22 keys and name panel by inserting label with new name into the proper slot; 22 LEDs with blinking attribute and Buzzer manageable by software; built-in 24 Vac power supply; RTC option, reader of magnetic badge and relays.

**QTP 24**
Quick Terminal Panel 24 keys
Intelligent user panel equipped with Fluorescent 20x2 or LCD display, LEDs backlit, 20x2 or 20x4 characters; RS 232, RS 422-485 or Current Loop serial lines; serial EEPROM for set-up and messages; Possibility of re-naming keys, LEDs and panel name by inserting label with new name into the proper slot; 24 keys and 16 LEDs with blinking attribute and Buzzer manageable by software; built-in power supply; RTC option, reader of magnetic badge and relays.

**QTP G26**
Quick Terminal Panel LCD Graphic
Intelligent user panel equipped with graphic LCD display 240x120 pixels, LEDs backlit; 1RS 232 line, additional RS 232, RS 422-485 or Current Loop lines; serial EEPROM for set-up; 256K EPROM, FLASH and EEPROM; RTC and 128K RAM; primary graphic object; Possibility of re-naming keys, LEDs and panel name by inserting label with new name into the proper slot; 16 keys and 16 LEDs, Buzzer, built-in power supply.
**OBI 01 - OBI 02**
Opto BLOCK Input NPN-PNP
Interface between 16 NPN, PNP optocoupled and displayed input lines, screw terminal, and Abaco® standard I/O 20 pins connector; power supply section; connection for DIN C type and Ω rails.

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

**TBO 01**
Transistor BLOCK Output
Interface between ABACO® standard I/O 20 pins connector and 16 transistor output lines; 45 Vdc 3 A open collector; screw terminal; optocoupled and displayed signals; connection for DIN C type and Ω rails.

**RBO 01**
Relays BLOCK Output
Interface between ABACO® standard I/O 20 pins connector and 8 relay output lines; 5 or 10 A outputs; N.O. or N.C. contact; screw terminal; displayed signals; connection for DIN C type and Ω rails.

**XBI 01**
miXed BLOCK Input-Output
Interface for ABACO® standard I/O 20 pins connector; 8 transistor output lines 45 Vdc 3 A; 8 input lines; screw terminal; optocoupled and displayed signals; connection for DIN C type and Ω rails.

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

**DEB 01**
Didactis 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.

**KDI LT - KDI L32 - KDI L33 - KDI FF - KDI F32 - KDI F33**
Keyboard Display Interface 32 Key
Interface for AAbaco® standard I/O 20 pins connector; 32 keys matrix keyboard (short key for 32 types, long keys for 33 types and external keyboard for LT, FF types); 8 LEDs; buzzer; fluorescentor LCD alphanumeric display.

**CBT 420**
Current Block Transmitter 4÷20mA
Interface between 4 input lines 0÷5, 0÷10 Vdc and 4 current output channels 4÷20 mA; signals on screw terminal; 14 bit resolution; quick connection for DIN C type and Ω rails.
**KDL 224 - KDL 424**

Keyboard Display LCD
Interface for **AAbaco®** standard I/O 20 pins connector; 24 keys matrix keyboard (compatible pin out with 3x4 and 4x4 telephone keyboards); LCD alphanumeric display with 20x2 or 20x4 characters.

**KDF 224**

Keyboard Display FUTABA
Interface for **AAbaco®** standard I/O 20 pins connector; 24 keys matrix keyboard (compatible pin out with 3x4 and 4x4 telephone keyboards); fluorescent alphanumeric display with 20x2 or 20x4 characters.

**IAC 01**

Interface Adapter Centronics
Interface between **AAbaco®** standard I/O 20 pins connector and D 25 pins connector with Centronics standard pin out.
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® 552.

Data book MAXIM: New Releases Data Book - Volume 4
Data book HEWLETT PACKARD: Optoelectronic Designer's Catalog
Data book NEC: Memory Products
Data book NEC: Microprocessors and Peripherals - Volume 3

Data book TEXAS INSTRUMENTS: The TTL Data Book - SN54/74 Families
Data book TEXAS INSTRUMENTS: Linear Circuits Data Book - Volumi 1 e 3
Data book TEXAS INSTRUMENTS: RS-422 and RS-485 Interface Circuits

Data book XICOR: Data Book
Data book PHILIPS: 80C51 - Based 8-Bit Microcontrollers
Data book PHILIPS: IC12 - IC Bus

Data book NATIONAL SEMICONDUCTOR: Linear Databook - Volume 1

For further informations and upgrades please refer to specific internet web pages of the manufacturing companies.
Figure 29: Available connections diagram

- **Power Supply**: 230Vac (standard), 50-60 Hz
- **Optional**: +5V or +12 Vdc, 6-10 Vdc, 8-26 Vdc (switching)
- **Digital TTL Input/Output**: to XBI-01, OBI-01, RBO-08 etc...
- **Opto Coupled**: TRANS.
- **Relay**: Coupled
- **Direct Connection**: TO QTP 24P
- **10 Bit Analog Input**: Voltage +2-400V opt. +5Vdc or Current 0-20mA +4-20mA (or 8 digital input)
- **Current to Voltage Converter**: with 8 A-V modules
- **External Lithium Battery**: 3.6 V for RAM Back up
- **1 RS 232 OR RS 422, 485, Current Loop**
- **1 Software Serial Line**: RS-232 only
- **1 Software Serial Line**: PC-like or Macintosh
- **PLC**: QTP 24 etc.
- **PC-like or Macintosh**: PLC
- **1C BUS**: RS-232, RS-422, RS-485, Current Loop
- **1C Line 2C BUS**: RS-232, RS-422, RS-485, Current Loop
- **Buzzer**: Direct Connection TO QTP 24P
- **Figure 29: Available Connections Diagram**
APPENDIX A: JUMPERS LOCATION

Figure A1: Serial Communication Jumpers Location
Figure A2: Memory Selection Jumpers Location
Appendix B: On Board Device Description

Philips Semiconductors 80C552/83C552

Single-chip 8-bit microcontroller with 10-bit A/D, capture/compare timer, high-speed outputs, PWM

Features:
- 80C51 central processing unit
- 8k × 8 ROM expandable externally to 64k bytes
- ROM code protection
- An additional 16-bit timer/counter coupled to four capture registers and three compare registers
- Two standard 16-bit timer/counters
- 256 × 8 RAM, expandable externally to 64k bytes
- Capable of producing eight synchronized, timed outputs
- A 10-bit ADC with eight multiplexed analog inputs
- Two 8-bit resolution, pulse width modulation outputs
- Five 8-bit I/O ports plus one 8-bit input port shared with analog inputs
- PC-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART compatible with the standard 80C51
- On-chip watchdog timer
- Three speed ranges:
  - 3.5 to 16MHz
  - 3.5 to 24MHz (ROM, ROMless only)
  - 3.5 to 30MHz (ROM, ROMless only)
- Three operating ambient temperature ranges:
  - P83C552xBx: 0°C to +70°C
  - P83C552xFx: −40°C to +85°C (XTAL frequency max. 24 MHz)
  - P83C552xHx: −40°C to +125°C (XTAL frequency max. 16 MHz)

The 80C552/83C552 (hereafter generically referred to as 8XC552) Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 8XC552 has the same instruction set as the 80C51. Three versions of the derivative exist:
- 83C552—8k bytes mask programmable ROM
- 80C552—ROMless version of the 83C552
- 87C552—8k bytes EPROM (described in a separate chapter)

The 8XC552 contains a non-volatile 8k × 8 read-only program memory (83C552), a volatile 256 × 8 read/write data memory, five 8-bit I/O ports, one 8-bit input port, two 16-bit timer/event counters (identical to the timers of the 80C51), an additional 16-bit timer coupled to capture and compare latches, a 15-source, two-priority-level, nested interrupt structure, an 8-input ADC, a dual DAC pulse width modulated interface, two serial interfaces (UART and I2C-bus), a “watchdog” timer and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XC552 can be expanded using standard TTL compatible memories and logic.

In addition, the 8XC552 has two software selectable modes of power reduction—idle mode and power-down mode. The idle mode freezes the CPU while allowing the RAM, timers, serial ports, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 45 two-byte, and 17 three-byte. With a 16MHz (24MHz) crystal, 58% of the instructions are executed in 0.75µs (0.5µs) and 40% in 1.5µs (1µs). Multiply and divide instructions require 3µs (2µs).

1998 Aug 13
The 8XC552 is a stand-alone high-performance microcontroller designed for use in real-time applications such as instrumentation, industrial control, and automotive control applications, such as engine management and transmission control. The device provides, in addition to the 80C51 standard functions, a number of dedicated hardware functions for these applications.

The 8XC552 single-chip 8-bit microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 8XC552 uses the powerful instruction set of the 80C51. Additional special function registers are incorporated to control the on-chip peripherals. These functions are available even if the device is in active mode and do not exhibit any of the drawbacks of an 80C51.

The 8XC552 contains a nonvolatile 8k x 8-bit-only program memory, a 256 x 8-bit data memory, the 80C51 I/O ports and on-chip 8-bit input, port 0x68 timer/event counters (identical to the 80C51), an additional 16-bit timer, and programmable capture/compare registers. Many system features require extra capability, and the 8XC552 can be expanded using standard CMOS and discrete components.

The 8XC552 has software-selectable modes of reduced activity for further power reduction—Idle and Power-down. In Idle mode, the CPU is reset and Timer T2 is stopped. When the CPU is re-enabled, Timer T2 recommences counting. The Power-down mode saves the RAM contents but frees the oscillator, allowing all other chip functions to become inactive.

**8XC552 OVERVIEW**

The 8XC552 contains a nonvolatile 8k x 8-bit-only program memory, a 256 x 8-bit data memory, the 80C51 I/O ports and on-chip 8-bit input, port 0x68 timer/event counters (identical to the 80C51), an additional 16-bit timer, and programmable capture/compare registers. Many system features require extra capability, and the 8XC552 can be expanded using standard CMOS and discrete components.

The 8XC552 has software-selectable modes of reduced activity for further power reduction—Idle and Power-down. In Idle mode, the CPU is reset and Timer T2 is stopped. When the CPU is re-enabled, Timer T2 recommences counting. The Power-down mode saves the RAM contents but frees the oscillator, allowing all other chip functions to become inactive.

The 8XC552 has been derived from the 8XC552 with the following changes:

- The 8XC552 retains the 8-kbyte program memory, 256 bytes RAM, 80C51 I/O ports, and on-chip 8-bit input, port 0x68 timer/event counters (identical to the 80C51), and an additional 16-bit timer coupled to capture/compare registers, a fifteen-source, two priority-level, nested interrupt structure, an 8-bit ADC, a dual D/C pulse-width modulation interface, two serial interfaces (UART and I²C bus), a "watchdog" timer, and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XC552 can be expanded using standard CMOS and discrete components.

The 8XC552 has software-selectable modes of reduced activity for further power reduction—Idle and Power-down. In Idle mode, the CPU is reset and Timer T2 is stopped. When the CPU is re-enabled, Timer T2 recommences counting. The Power-down mode saves the RAM contents but frees the oscillator, allowing all other chip functions to become inactive.

The 8XC552 has been derived from the 8XC552 with the following changes:

- The SIO (I²C) interface has been omitted.
- The output of port pins P0.6 and P0.7 has been changed from 8-bit to 8-bit.
- The 8XC552 has been derived from the 8XC552 with the following changes:
  - The resolution of the AD converter is increased from 10 bits to 8 bits.
  - The time of an AD conversion has decreased from 50 machine cycles to 24 machine cycles.

In all other functions, pinning and packaging are unchanged.

This chapter of the user's guide can be used with the 8XC552 by omitting or changing the following sections:

- Omit the description of SIO (I²C)
- Omit the SIO configuration interface: 83C552, 80C552, 87C552, and 8XC552 are not implemented. The I²C-related flag is present in SFR I²C (address 0EEH) and P1.7 is program-implemented. These two flags are undefined after the initial power-up sequence. The interrupt vector for the pin is held low, all instruction fetches are then performed using the standard configuration. After reset, the 8-bit P1.6 and P1.7 have alternative functions.

The 8XC552 uses a powerful instruction set (see Figure 1) comprising on-chip peripheral hardware. Other registers include on-chip peripheral hardware. Other registers include on-chip peripheral hardware. Other registers include on-chip peripheral hardware. Other registers include on-chip peripheral hardware. Other registers include on-chip peripheral hardware.
80C51 Family Derivatives

Table 1. 8XC552 Special Function Registers (Continued)

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>DESCRIPTION</th>
<th>DIRECT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION</th>
<th>LSB</th>
<th>RESET VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWMP#</td>
<td>PWM prescaler</td>
<td>ES#</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>PWMM#</td>
<td>PWM register 1</td>
<td>ICH</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>PWMO#</td>
<td>PWM register 0</td>
<td>ICH</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>RBF#</td>
<td>Reset frag register</td>
<td>EP#</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>SP#</td>
<td>Stack pointer</td>
<td>8H</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>SIAD#</td>
<td>Serial 0 address</td>
<td>DA#</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>S1AD#</td>
<td>Serial 1 address</td>
<td>DA#</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>S0BUF#</td>
<td>Serial 0 data buffer</td>
<td>9BH</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>S0CON#</td>
<td>Capture low 0</td>
<td>DF#</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>S1DRA#</td>
<td>Capture low 1</td>
<td>DF#</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>S1DB#</td>
<td>Capture low 2</td>
<td>DF#</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>S1DR#</td>
<td>Capture low 3</td>
<td>DF#</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>S1CON#</td>
<td>Capture high 0</td>
<td>DF#</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>S2CON#</td>
<td>Capture high 1</td>
<td>DF#</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>S3CON#</td>
<td>Capture high 2</td>
<td>DF#</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>S4CON#</td>
<td>Capture high 3</td>
<td>DF#</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>T1#</td>
<td>Timer 1 high</td>
<td>DF#</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>T2#</td>
<td>Timer 2 high</td>
<td>DF#</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>T3#</td>
<td>Timer 3 high</td>
<td>DF#</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>T0#</td>
<td>Timer 0 high</td>
<td>DF#</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>T1O#</td>
<td>Timer 1 overflow</td>
<td>DF#</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>T2O#</td>
<td>Timer 2 overflow</td>
<td>DF#</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>T3O#</td>
<td>Timer 3 overflow</td>
<td>DF#</td>
<td></td>
<td>00H</td>
</tr>
</tbody>
</table>

Legend:
- # SFRs are bit addressable.
- * SFRs are modified from or added to the 80C51 SFRs.
- ** SFRs are modified from or added to the 80C51 SFRs.

Please refer to the full document for further details.
Timer T2: Capture and Compare Logic

Timer T2 is connected to four 16-bit capture registers and three 16-bit compare registers. A capture register may be used to capture the contents of Timer T2 when a transition occurs on its corresponding input pin. A compare register may be used to set, reset, or toggle port 4 output pins at certain pre-programmed time intervals.

The combination of Timer T2 and the capture and compare logic is very powerful in applications involving rotating machinery, automotive injection systems, etc. Timer T2 and the capture and compare logic are shown in Figure 4.

Capture Logic: The four 16-bit capture registers that Timer T2 is connected to are: CT0, CT1, CT2, and CT3. These registers are loaded with the contents of Timer T2, and an interrupt is requested upon receipt of the input CT0I, CT1I, CT2I, or CT3I of Timer T2. These input signals are latched and sampled during S1P1 of the cycle following the cycle in which a match occurred. If the capture facility is not required, these inputs can be regarded as additional external interrupt inputs.

Using the capture control register CTCON (see Figure 5), these inputs may capture on a rising edge, a falling edge, or on either a rising or falling edge. The inputs are sampled during S1P1 of each cycle. When a selected edge is detected, the contents of Timer T2 are captured at the end of the cycle.

Compare Logic: Each time Timer T2 is incremented, the contents of the three 16-bit compare registers CM0, CM1, and CM2 are compared with the new counter value of Timer T2. When a match is found, the corresponding interrupt flag in TM2IR is set at the end of the following cycle. When a match occurs, the controller sets bits 0-5 of port 4 (the corresponding bits of the set enable register STE are at logic 1). If a match occurs, the controller stops the increment timer interrupt service routine.

When a match with CM0 occurs, the controller sets bits 0-7 of port 4 (the corresponding bits of the reset enable register RETE are at logic 1). If a match occurs, the controller stops the increment timer interrupt service routine.

When a match with CM1 occurs, the controller sets bits 0-7 of port 4 (the corresponding bits of the reset enable register RETE are at logic 1). If a match occurs, the controller stops the increment timer interrupt service routine.

When a match with CM2 occurs, the controller sets bits 0-7 of port 4 (the corresponding bits of the reset enable register RETE are at logic 1). If a match occurs, the controller stops the increment timer interrupt service routine.

Two additional flags are set after the last operation, and it is these flags that are toggled.

Thus, if the current operation is "set," the next operation will be "reset," and vice versa. It is possible that the "reset" operation occurs in a three-byte extension on which gives a maximum cycle time of approximately 3.68 hours.

Capture Logic: The four 16-bit capture registers that Timer T2 is connected to are: CT0, CT1, CT2, and CT3. These registers are loaded with the contents of Timer T2, and an interrupt is requested upon receipt of the input CT0I, CT1I, CT2I, or CT3I of Timer T2. These input signals are latched and sampled during S1P1 of the cycle following the cycle in which a match occurred. If the capture facility is not required, these inputs can be regarded as additional external interrupt inputs.

Compare Logic: Each time Timer T2 is incremented, the contents of the three 16-bit compare registers CM0, CM1, and CM2 are compared with the new counter value of Timer T2. When a match is found, the corresponding interrupt flag in TM2IR is set at the end of the following cycle. When a match occurs, the controller sets bits 0-5 of port 4 (the corresponding bits of the set enable register STE are at logic 1). If a match occurs, the controller stops the increment timer interrupt service routine.

When a match with CM0 occurs, the controller sets bits 0-7 of port 4 (the corresponding bits of the reset enable register RETE are at logic 1). If a match occurs, the controller stops the increment timer interrupt service routine.

When a match with CM1 occurs, the controller sets bits 0-7 of port 4 (the corresponding bits of the reset enable register RETE are at logic 1). If a match occurs, the controller stops the increment timer interrupt service routine.

When a match with CM2 occurs, the controller sets bits 0-7 of port 4 (the corresponding bits of the reset enable register RETE are at logic 1). If a match occurs, the controller stops the increment timer interrupt service routine.

Two additional flags are set after the last operation, and it is these flags that are toggled.
interrupt requests are recognized during the following cycle. If these
flags are not polled during S2; CMI1 and CMI2 are scanned during S3 and S4. A match
will be recognized by the interrupt logic (or by polling the flags) two
cycles after the match take place.

The 16-bit overflow flag (T2OV) and the byte overflow flag (T2BO)
are set during S6 of the cycle in which the overflow occurs. These
flags are recognized by the interrupt logic during the next cycle.

Special function register IP1 (Figure 8) is used to determine the
Timer T2 interrupt priority. Setting a bit high gives the function a
high priority, and setting a bit low gives the function a low priority.
The functions controlled by the various bits of this IP1 register are shown in Figure 8.

| BIT SYMBOL CAPTURE/INTERRUPT ON: |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| IP1.7 PT2 Timer T2 overflow interrupt(s) priority level |
| IP1.6 PCM2 Timer T2 comparator 2 interrupt priority level |
| IP1.4 PCM0 Timer T2 comparator 0 interrupt priority level |
| IP1.3 PCT3 Timer T2 capture register 3 interrupt priority level |
| IP1.2 PCT2 Timer T2 capture register 2 interrupt priority level |
| IP1.1 PCT1 Timer T2 capture register 1 interrupt priority level |
| IP1.0 PCT0 Timer T2 capture register 0 interrupt priority level |

Interrupt Flag Register (TM2IR)

In addition to Timer T2 and the standard timers, a watchdog timer is
also incorporated on the 8XC552. The purpose of a watchdog timer is to reset the
microcontroller if a microprocessor state (possibly caused by electrical noise or RFI) within an
acceptable period of time. An analogy is the "dead man's handle" in railway
locomotives. When enabled, the watchdog circuitry will generate a
system reset if the user program fails to reload the watchdog timer
within a specified length of time known as the "watchdog interval." To operate the watchdog timer,
the programmer must first set the watchdog timer to the length chosen by the user.

Timer T3, The Watchdog Timer

In addition to Timer T2 and the standard timers, a watchdog timer is
also incorporated on the 8XC552. The purpose of a watchdog timer is to reset the
microcontroller if a microprocessor state (possibly caused by electrical noise or RFI) within an
acceptable period of time. An analogy is the "dead man's handle" in railway
locomotives. When enabled, the watchdog circuitry will generate a
system reset if the user program fails to reload the watchdog timer
within a specified length of time known as the "watchdog interval." To operate the watchdog timer,
the programmer must first set the watchdog timer to the length chosen by the user.

The prescaler is set with a signal whose frequency is 1/12 the
oscillator frequency (1MHz with a 12MHz oscillator). The 8-bit timer is incremented every "t" seconds,
where:

\[ t = \frac{1}{12 \times \frac{1}{f_{osc}}} \] (= 1.5ms at f_{osc} = 16MHz; = 1ms at f_{osc} = 24MHz)

If the 8-bit timer overflows, a short internal reset pulse is generated
operation by software.

Timer T3, The Watchdog Timer

In addition to Timer T2 and the standard timers, a watchdog timer is
also incorporated on the 8XC552. The purpose of a watchdog timer is to reset the
microcontroller if a microprocessor state (possibly caused by electrical noise or RFI) within an
acceptable period of time. An analogy is the "dead man's handle" in railway
locomotives. When enabled, the watchdog circuitry will generate a
system reset if the user program fails to reload the watchdog timer
within a specified length of time known as the "watchdog interval." To operate the watchdog timer,
the programmer must first set the watchdog timer to the length chosen by the user.

The prescaler is set with a signal whose frequency is 1/12 the
oscillator frequency (1MHz with a 12MHz oscillator). The 8-bit timer is incremented every "t" seconds,
where:

\[ t = \frac{1}{12 \times \frac{1}{f_{osc}}} \] (= 1.5ms at f_{osc} = 16MHz; = 1ms at f_{osc} = 24MHz)

If the 8-bit timer overflows, a short internal reset pulse is generated
operation by software.

Timer T3, The Watchdog Timer

In addition to Timer T2 and the standard timers, a watchdog timer is
also incorporated on the 8XC552. The purpose of a watchdog timer is to reset the
microcontroller if a microprocessor state (possibly caused by electrical noise or RFI) within an
acceptable period of time. An analogy is the "dead man's handle" in railway
locomotives. When enabled, the watchdog circuitry will generate a
system reset if the user program fails to reload the watchdog timer
within a specified length of time known as the "watchdog interval." To operate the watchdog timer,
the programmer must first set the watchdog timer to the length chosen by the user.

The prescaler is set with a signal whose frequency is 1/12 the
oscillator frequency (1MHz with a 12MHz oscillator). The 8-bit timer is incremented every "t" seconds,
where:

\[ t = \frac{1}{12 \times \frac{1}{f_{osc}}} \] (= 1.5ms at f_{osc} = 16MHz; = 1ms at f_{osc} = 24MHz)

If the 8-bit timer overflows, a short internal reset pulse is generated
operation by software.

Timer T3, The Watchdog Timer

In addition to Timer T2 and the standard timers, a watchdog timer is
also incorporated on the 8XC552. The purpose of a watchdog timer is to reset the
microcontroller if a microprocessor state (possibly caused by electrical noise or RFI) within an
acceptable period of time. An analogy is the "dead man's handle" in railway
locomotives. When enabled, the watchdog circuitry will generate a
system reset if the user program fails to reload the watchdog timer
within a specified length of time known as the "watchdog interval." To operate the watchdog timer,
the programmer must first set the watchdog timer to the length chosen by the user.

The prescaler is set with a signal whose frequency is 1/12 the
oscillator frequency (1MHz with a 12MHz oscillator). The 8-bit timer is incremented every "t" seconds,
where:

\[ t = \frac{1}{12 \times \frac{1}{f_{osc}}} \] (= 1.5ms at f_{osc} = 16MHz; = 1ms at f_{osc} = 24MHz)

If the 8-bit timer overflows, a short internal reset pulse is generated
operation by software.

Timer T3, The Watchdog Timer

In addition to Timer T2 and the standard timers, a watchdog timer is
also incorporated on the 8XC552. The purpose of a watchdog timer is to reset the
microcontroller if a microprocessor state (possibly caused by electrical noise or RFI) within an
acceptable period of time. An analogy is the "dead man's handle" in railway
locomotives. When enabled, the watchdog circuitry will generate a
system reset if the user program fails to reload the watchdog timer
within a specified length of time known as the "watchdog interval." To operate the watchdog timer,
the programmer must first set the watchdog timer to the length chosen by the user.

The prescaler is set with a signal whose frequency is 1/12 the
oscillator frequency (1MHz with a 12MHz oscillator). The 8-bit timer is incremented every "t" seconds,
where:

\[ t = \frac{1}{12 \times \frac{1}{f_{osc}}} \] (= 1.5ms at f_{osc} = 16MHz; = 1ms at f_{osc} = 24MHz)

If the 8-bit timer overflows, a short internal reset pulse is generated
operation by software.

Timer T3, The Watchdog Timer

In addition to Timer T2 and the standard timers, a watchdog timer is
also incorporated on the 8XC552. The purpose of a watchdog timer is to reset the
microcontroller if a microprocessor state (possibly caused by electrical noise or RFI) within an
acceptable period of time. An analogy is the "dead man's handle" in railway
locomotives. When enabled, the watchdog circuitry will generate a
system reset if the user program fails to reload the watchdog timer
within a specified length of time known as the "watchdog interval." To operate the watchdog timer,
the programmer must first set the watchdog timer to the length chosen by the user.

The prescaler is set with a signal whose frequency is 1/12 the
oscillator frequency (1MHz with a 12MHz oscillator). The 8-bit timer is incremented every "t" seconds,
where:

\[ t = \frac{1}{12 \times \frac{1}{f_{osc}}} \] (= 1.5ms at f_{osc} = 16MHz; = 1ms at f_{osc} = 24MHz)

If the 8-bit timer overflows, a short internal reset pulse is generated
operation by software.

Timer T3, The Watchdog Timer

In addition to Timer T2 and the standard timers, a watchdog timer is
also incorporated on the 8XC552. The purpose of a watchdog timer is to reset the
microcontroller if a microprocessor state (possibly caused by electrical noise or RFI) within an
acceptable period of time. An analogy is the "dead man's handle" in railway
locomotives. When enabled, the watchdog circuitry will generate a
system reset if the user program fails to reload the watchdog timer
within a specified length of time known as the "watchdog interval." To operate the watchdog timer,
the programmer must first set the watchdog timer to the length chosen by the user.

The prescaler is set with a signal whose frequency is 1/12 the
oscillator frequency (1MHz with a 12MHz oscillator). The 8-bit timer is incremented every "t" seconds,
where:

\[ t = \frac{1}{12 \times \frac{1}{f_{osc}}} \] (= 1.5ms at f_{osc} = 16MHz; = 1ms at f_{osc} = 24MHz)

If the 8-bit timer overflows, a short internal reset pulse is generated
operation by software.
The CPU interface to the I 2 C logic via the following four special function registers: SCON (I/O control register), SSTA (I/O data register), S1DAT (slave address register). The I 2 C logic interfaces to the external PC bus via its port 1 pins P1.6/SCL (serial clock line) and P1.7/SDA (serial data line).

A typical I 2 C bus configuration is shown in Figure 10, and Figure 11 shows how a data transfer is accomplished on the bus. Depending on the state of the direction bit (R/W), two types of data transfers are possible on this I 2 C bus:

1. Data transfer from a master transmitter to a slave receiver: The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave responds with an acknowledge bit after each received byte.

2. Data transfer from a slave transmitter to a master receiver: The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. Next follows the data bytes transmitted by the slave to the master. The master returns an acknowledge bit after all received bytes, other than the last byte. At the end of the last received byte, a "not acknowledged" is returned.

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition or with a repeated START condition is not allowed, the master waits until the bus is free before the master mode is entered so that a possible slave action is not interrupted. If bus arbitration is not in the master mode, the I 2 C interface is implemented, and the following text describes the individual blocks.

**Interior Filters and Output Stage**

The input levels have I 2 C compatible input levels. If the input voltage is less than 1.5V, the input logic level is interpreted as 0; if the input voltage is greater than 3.0V, the input logic level is interpreted as 1. Input signals are synchronized with the internal clock (fosc/12).

**Watchdog Timer**

The watchdog timer is reloaded in two stages in order to prevent erroneous software from locking the watchdog. First PCON.4 (WLE) must be set. The T3 (WLE) must be loaded. When T3 is loaded, PCON.4 (WLE) is automatically reset. T3 cannot be loaded if PCON.4 (WLE) is reset. Reload code may be put in a subroutine as it is called infrequently. Since Timer 3 is an up-counter, a reload value of 00H gives the minimum watchdog interval (2ms with a 12MHz oscillator), and a reload value of 0FFH gives the minimum watchdog interval (2ms with a 12MHz oscillator).

In this mode, the watchdog circuitry remains active. When watchdog operation is implemented, the power-down mode cannot be used since both states are contradictory. Thus, when watchdog operation is enabled by setting the EW bit, the input data cannot be used. If the watchdog input data is changed or the input signal is corrupted, an attempt to set the watchdog data will have no effect. PCON.1 will remain at logic 0.

During the early stages of software development debugging, the watchdog may be disabled by tying the EW pin high. At a later stage, EW may be tied low to facilitate debugging.

**Operating Examples**

The following example shows how a watchdog operation might be handled in a user program:

```
1. CALL WATCHDOG
2. ; watchdog operation starts
3. ; watchdog operation disabled
4. ; watchdog operation ends
```

The next example shows how a watchdog operation might be handled in a user program:

```
1. CALL WATCHDOG
2. ; watchdog operation starts
3. ; watchdog operation disabled
4. ; watchdog operation ends
5. ; watchdog operation disabled
```

### 80C51 Family Derivatives

#### 8XC552/562 overview

The CPU interface to the I 2 C logic via the following four special function registers: SCON (I/O control register), SSTA (I/O data register), S1DAT (slave address register). The I 2 C logic interfaces to the external PC bus via its port 1 pins P1.6/SCL (serial clock line) and P1.7/SDA (serial data line).

A typical I 2 C bus configuration is shown in Figure 10, and Figure 11 shows how a data transfer is accomplished on the bus. Depending on the state of the direction bit (R/W), two types of data transfers are possible on this I 2 C bus:

1. Data transfer from a master transmitter to a slave receiver: The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave responds with an acknowledge bit after each received byte.

2. Data transfer from a slave transmitter to a master receiver: The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. Next follows the data bytes transmitted by the slave to the master. The master returns an acknowledge bit after all received bytes, other than the last byte. At the end of the last received byte, a "not acknowledged" is returned.

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition or with a repeated START condition is not allowed, the master waits until the bus is free before the master mode is entered so that a possible slave action is not interrupted. If bus arbitration is not in the master mode, the I 2 C interface is implemented, and the following text describes the individual blocks.

**Interior Filters and Output Stage**

The input levels have I 2 C compatible input levels. If the input voltage is less than 1.5V, the input logic level is interpreted as 0; if the input voltage is greater than 3.0V, the input logic level is interpreted as 1. Input signals are synchronized with the internal clock (fosc/12).

**Watchdog Timer**

The watchdog timer is reloaded in two stages in order to prevent erroneous software from locking the watchdog. First PCON.4 (WLE) must be set. The T3 (WLE) must be loaded. When T3 is loaded, PCON.4 (WLE) is automatically reset. T3 cannot be loaded if PCON.4 (WLE) is reset. Reload code may be put in a subroutine as it is called infrequently. Since Timer 3 is an up-counter, a reload value of 00H gives the minimum watchdog interval (2ms with a 12MHz oscillator), and a reload value of 0FFH gives the minimum watchdog interval (2ms with a 12MHz oscillator).

In this mode, the watchdog circuitry remains active. When watchdog operation is implemented, the power-down mode cannot be used since both states are contradictory. Thus, when watchdog operation is enabled by setting the EW bit, the input data cannot be used. If the watchdog input data is changed or the input signal is corrupted, an attempt to set the watchdog data will have no effect. PCON.1 will remain at logic 0.

During the early stages of software development debugging, the watchdog may be disabled by tying the EW pin high. At a later stage, EW may be tied low to facilitate debugging.

**Operating Examples**

The following example shows how a watchdog operation might be handled in a user program:

```
1. CALL WATCHDOG
2. ; watchdog operation starts
3. ; watchdog operation disabled
4. ; watchdog operation ends
```

The next example shows how a watchdog operation might be handled in a user program:

```
1. CALL WATCHDOG
2. ; watchdog operation starts
3. ; watchdog operation disabled
4. ; watchdog operation ends
5. ; watchdog operation disabled
```

### 80C51 Family Derivatives

#### 8XC552/562 overview

The CPU interface to the I 2 C logic via the following four special function registers: SCON (I/O control register), SSTA (I/O data register), S1DAT (slave address register). The I 2 C logic interfaces to the external PC bus via its port 1 pins P1.6/SCL (serial clock line) and P1.7/SDA (serial data line).

A typical I 2 C bus configuration is shown in Figure 10, and Figure 11 shows how a data transfer is accomplished on the bus. Depending on the state of the direction bit (R/W), two types of data transfers are possible on this I 2 C bus:

1. Data transfer from a master transmitter to a slave receiver: The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave responds with an acknowledge bit after each received byte.

2. Data transfer from a slave transmitter to a master receiver: The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. Next follows the data bytes transmitted by the slave to the master. The master returns an acknowledge bit after all received bytes, other than the last byte. At the end of the last received byte, a "not acknowledged" is returned.

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition or with a repeated START condition is not allowed, the master waits until the bus is free before the master mode is entered so that a possible slave action is not interrupted. If bus arbitration is not in the master mode, the I 2 C interface is implemented, and the following text describes the individual blocks.

**Interior Filters and Output Stage**

The input levels have I 2 C compatible input levels. If the input voltage is less than 1.5V, the input logic level is interpreted as 0; if the input voltage is greater than 3.0V, the input logic level is interpreted as 1. Input signals are synchronized with the internal clock (fosc/12).

**Watchdog Timer**

The watchdog timer is reloaded in two stages in order to prevent erroneous software from locking the watchdog. First PCON.4 (WLE) must be set. The T3 (WLE) must be loaded. When T3 is loaded, PCON.4 (WLE) is automatically reset. T3 cannot be loaded if PCON.4 (WLE) is reset. Reload code may be put in a subroutine as it is called infrequently. Since Timer 3 is an up-counter, a reload value of 00H gives the minimum watchdog interval (2ms with a 12MHz oscillator), and a reload value of 0FFH gives the minimum watchdog interval (2ms with a 12MHz oscillator).

In this mode, the watchdog circuitry remains active. When watchdog operation is implemented, the power-down mode cannot be used since both states are contradictory. Thus, when watchdog operation is enabled by setting the EW bit, the input data cannot be used. If the watchdog input data is changed or the input signal is corrupted, an attempt to set the watchdog data will have no effect. PCON.1 will remain at logic 0.

During the early stages of software development debugging, the watchdog may be disabled by tying the EW pin high. At a later stage, EW may be tied low to facilitate debugging.

**Operating Examples**

The following example shows how a watchdog operation might be handled in a user program:

```
1. CALL WATCHDOG
2. ; watchdog operation starts
3. ; watchdog operation disabled
4. ; watchdog operation ends
```

The next example shows how a watchdog operation might be handled in a user program:

```
1. CALL WATCHDOG
2. ; watchdog operation starts
3. ; watchdog operation disabled
4. ; watchdog operation ends
5. ; watchdog operation disabled
```
Other Device with 
I2C Interface

Figure 10. Typical I2C Bus Configuration

Start
Condition

Acknowledgment
Signal from Receiver

Slave Address
R/W
Direction

Repeated
Start
Condition

Stop
Condition

Repeated

Figure 11. Data Transfer on the I2C Bus

Figure 12. I2C Bus Serial Interface Block Diagram
The most significant bit corresponds to the first bit received from the I2C bus after a start condition. A logic 1 in S1ADR corresponds to a high level on the I2C bus, and a logic 0 corresponds to a low level on the bus.

The Data Register, S1DAT, contains a byte of serial data to be transmitted or just received. A logic 1 in S1DAT corresponds to a high level on the I2C bus, and a logic 0 corresponds to a low level on the bus.

SIO1 hardware. The contents of this register are irrelevant when SIO1 is in a slave mode. The seven least significant bits of the status register are always zero. If the status code is used as a vector to service routines, then the routines are displaced by eight address locations. Eight bytes of code are sufficient for most of the service routines (see the software example in this section).

The Four SIO1 Special Function Registers: S1ADR (DBH), S1DAT, S1CON, and S1STA) are described individually in the following sections.

The synchronization logic will synchronize the serial clock generator with the clock pulses on the SCL line from another device. If two or more master devices generate clock pulses, the “mark” duration is determined by the device that generates the shortest “marks,” and the “space” duration is determined by the device that generates the longest “spaces.” Figure 14 shows the synchronization procedure.

A slave may stretch the space duration to slow down the bus. If the bus is busy when a slave device reads the SCL line, the slave device stretches the space duration to prevent other devices from attempting to communicate on the bus. A master device can also extend the space duration to slow down the bus if it is busy when a slave device reads the SCL line. When a master device extends the space duration, the acknowledge bit is ignored.

The address decoder takes all of the internal status bits and compresses them into a 5-bit code. This code is unique for each I2C bus status. The 5-bit code may be used to generate vector addresses for fast processing of the various service routines. Each service routine processes a particular status bus. There are 32 possible status bus modes of S1DAT are used. The 5-bit status code is latched into the five most significant bits of the status register when the serial interrupt flag is set (by hardware) and remains visible until the interrupt flag is cleared by software. If the code is used as a vector to service routines, then the routines are displaced by eight address locations. Eight bytes of code are sufficient for most of the service routines (see the software example in this section).

The Four SIO1 Special Function Registers: The microcontroller interfaces to S1DAT via four special function registers. These four SFRs (S1ADR, S1DAT, S1CON, and S1STA) are described individually in the following sections.

The address register, S1ADR, is used to control the I2C bus. The CPU can read from and write to the S1ADR register to access the I2C bus. The S1ADR register contains the slave address and the data to be transmitted or received. S1ADR is a read/write register.

The data register, S1DAT, contains a byte of serial data to be transmitted or received. S1DAT is a read/write register.

The control register, S1CON, contains control bits to select the operation mode of S1DAT. S1CON is a read/write register.

The status register, S1STA, contains status flags to indicate the current state of S1DAT. S1STA is a read-only register.

The clock pulses for serial byte handling are generated by the shift pulse generator. The shift pulse generator provides the SCL clock pulses when S1DAT is in the master transmitter or master receiver mode. The shift pulse generator also generates the SDA clock pulses for serial data transfer.

The synchronization logic will synchronize the serial clock generator with the clock pulses on the SCL line from another device. If two or more master devices generate clock pulses, the “mark” duration is determined by the device that generates the shortest “marks,” and the “space” duration is determined by the device that generates the longest “spaces.” Figure 14 shows the synchronization procedure.
80C51 Family Derivatives

8XC552/562 overview

The Status Register, S1STA: In this bit only special functions are global. The three least significant bits are always zero. The five most significant bits contain the status code. There are 26 possible status codes. When S1STA contains F4H, no relevant data information is available and no serial interrupt is generated. All other S1STA values correspond to defined S1STAT states. When each of these states is entered, a serial interrupt is requested (SI = "1"). A valid status code is present in S1STA on a machine cycle after SI is set by hardware and it is still present one machine cycle after SI has been raised by the microcontroller.

More information on S1 Operatung Modes: The four operating modes are:
- Master Transmitter
- Master Receiver
- Slave Receiver
- Slave Transmitter

Data transfers in each mode of operation are shown in Figures 17-19. These figures contain the following abbreviations:

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI</td>
<td>Start condition</td>
</tr>
<tr>
<td>SLA</td>
<td>Slave address</td>
</tr>
<tr>
<td>R</td>
<td>Read bit (high level at SDA)</td>
</tr>
<tr>
<td>W</td>
<td>Write bit (low level at SDA)</td>
</tr>
<tr>
<td>A</td>
<td>Acknowledge bit (low level at SDA)</td>
</tr>
<tr>
<td>Data</td>
<td>Data byte (high level at SDA)</td>
</tr>
<tr>
<td>Pol</td>
<td>Polarity</td>
</tr>
</tbody>
</table>

In Figures 17-19, circles are used to indicate when the serial interrupt flag is set. The numbers in the circles show the status code held in the S1STA register. At these points, a service routine must be executed to continue the serial transfer. These service routines are not critical since the serial transfer is suspended until the serial interrupt flag is cleared by software.

When a serial interrupt routine is entered, the status code in S1STA is used to branch into the appropriate service routine. For each status code, the required software action and details of the following serial transfer are given in Table 3.7.

Master Transmitter Mode: In this master transmitter mode, a number of data bytes are transmitted to a slave receiver (see Figure 17). Before the master transmitter mode can be entered, SIO1 must be initialized as follows:

<table>
<thead>
<tr>
<th>SI</th>
<th>SLA</th>
<th>R</th>
<th>W</th>
<th>A</th>
<th>Data</th>
<th>Pol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CPL CR1, and CR2 define the serial bit rate. SI must be set to logic 1 for master SIO. If the AA bit is reset, SI will not acknowledge its own slave address or the general call address in the event of another device becoming master of the bus. In other words, if AA is reset, SIO0 cannot enter a slave mode. STO, STA, and SI must be reset.

The master transmitter mode may now be entered by setting the STA bit using the SETB instruction. The SIO flag will reset the FC bus and generate a start condition as soon as the bus becomes free. When a START condition is transmitted, the serial interrupt flag (SI) is set, and the status code in the S1STAT register will be 01H. This status code must be used to initialize a serial interrupt service routine that loads S1STA with the slave address and the data bit (S1DAT). The SI bit in S1CON must therefore be set before the serial transfer can continue.

When the slave address and the direction bit have been transmitted and an acknowledge has been received, the serial interrupt flag (SI) is set again, and a number of status codes in S1STA are possible. There are 10H, 20H, or 38H if the slave mode was enabled (AA = 1). The appropriate action to be taken for each of these status codes is detailed in Table 3. A repeated start condition (state 10H) SIO1 may switch to the master receiver mode by loading S1FOH with SLA(0).
Figure 16. Shift-in and Shift-out Timing

Table 2. Serial Clock Rates

<table>
<thead>
<tr>
<th>CR2</th>
<th>CR1</th>
<th>CR0</th>
<th>BIT FREQUENCY (MHz) AT fOSC</th>
<th>fSclk DIVIDED BY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.25 + 125</td>
<td>3.96</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.5 + 25</td>
<td>3.96</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.5 + 125.5</td>
<td>3.96</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.5 + 62.5</td>
<td>3.96</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.625 + 50</td>
<td>3.96</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.625 + 100</td>
<td>3.96</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.625 + 50</td>
<td>3.96</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.625 + 100</td>
<td>3.96</td>
</tr>
</tbody>
</table>

Figure 17. Format and States in the Master Transmitter Mode
Figure 18. Format and States in the Master Receiver Mode

Figure 19. Format and States in the Slave Receiver Mode
Master Transmitter Mode:
In the master transmitter mode, a number of data bytes are received from a slave transmitter (see Figure 16). There are 3 main states: 1) If the slave mode was enabled (AA = 1), the appropriate action is taken. 2) If the AA bit is reset during a transfer, SIO1 will respond to the slave address of the master transmitter mode by loading SLA+W with SLA+W. 3) Any number of data bytes and their associated acknowledge bits can be transmitted. The upper 7 bits of the slave address are the address to which SIO1 will respond to.

Slave Receiver Mode:
In the slave receiver mode, a number of data bytes are received from a master transmitter (see Figure 15). To initiate the slave receiver mode, S1ADR and S1CON must be loaded as follows:

The upper 7 bits are the address to which SIO1 will respond to.

Table 3. Master Transmitter Mode

<table>
<thead>
<tr>
<th>STATUS CODE (S1STA)</th>
<th>STATUS OF THE FC BUS AND S1STA (B8H)</th>
<th>APPLICATION SOFTWARE RESPONSE</th>
<th>NEXT ACTION TAKEN BY SIO1 HARDWARE</th>
</tr>
</thead>
<tbody>
<tr>
<td>08H</td>
<td>SLA+W has been transmitted</td>
<td>SLA+W (W) will be transmitted; ACK bit will be received</td>
<td></td>
</tr>
<tr>
<td>19H</td>
<td>Repeated START condition has been transmitted</td>
<td>Repeated START condition will be transmitted; STOP condition will be transmitted; STO flag will be reset</td>
<td></td>
</tr>
<tr>
<td>18H</td>
<td>SLA+W/R has been transmitted</td>
<td>SLA+W/R will be transmitted; STO will be switched to MSTR/REC mode</td>
<td></td>
</tr>
<tr>
<td>28H</td>
<td>SLA+R has been transmitted</td>
<td>SLA+R will be transmitted; STO flag will be reset</td>
<td></td>
</tr>
<tr>
<td>38H</td>
<td>Data byte has been transmitted</td>
<td>Data byte will be transmitted; ACK bit will be received</td>
<td></td>
</tr>
</tbody>
</table>

When S1ADR and S1CON have been initialized, SIO1 waits until it is addressed by its own slave address (see Figure 16). If the AA bit is reset during a transfer, SIO1 will return a not acknowledge (logic 1) to SDA after the next received data byte. While AA is reset, SIO1 does not respond to its own slave address or a general call address.

If the AA bit is reset during a transfer, SIO1 will return a not acknowledge (logic 1) to SDA after the next received data byte. While AA is reset, SIO1 does not respond to its own slave address or a general call address. However, the FC bus is still monitored and address recognition may be resumed at any time by setting AA.

This means that the AA bit may be used to temporarily isolate SIO1 from the FC bus.
### Table 4. Master Receiver Mode

<table>
<thead>
<tr>
<th>STATUS CODE</th>
<th>STATUS OF THE PC BUS AND SO1 HARDWARE</th>
<th>APPLICATION SOFTWARE RESPONSE</th>
<th>NEXT ACTION TAKEN BY SO1 HARDWARE</th>
</tr>
</thead>
<tbody>
<tr>
<td>08H</td>
<td>A START condition has been transmitted</td>
<td>No SIADT action or no S1STA action</td>
<td>Data byte will be received and NOT ACK will be returned</td>
</tr>
<tr>
<td>09H</td>
<td>A repeated START condition has been transmitted</td>
<td>No SIADT action or no S1STA action</td>
<td>Data byte will be received and NOT ACK will be returned</td>
</tr>
<tr>
<td>28H</td>
<td>SLA+R has been transmitted; ACK bit will be received</td>
<td>Data byte will be received; ACK bit will be returned</td>
<td></td>
</tr>
<tr>
<td>29H</td>
<td>SLA+R has been transmitted; ACK bit will be received</td>
<td>Data byte will be received; ACK bit will be returned</td>
<td></td>
</tr>
<tr>
<td>38H</td>
<td>A START condition will be transmitted</td>
<td>Data byte will be received and ACK will be returned</td>
<td></td>
</tr>
<tr>
<td>39H</td>
<td>A START condition will be transmitted</td>
<td>Data byte will be received and ACK will be returned</td>
<td></td>
</tr>
<tr>
<td>48H</td>
<td>Data byte has been received; ACK bit will be received</td>
<td>DATA byte will be received; ACK will be returned</td>
<td></td>
</tr>
<tr>
<td>49H</td>
<td>Data byte has been received; ACK bit will be received</td>
<td>DATA byte will be received; ACK will be returned</td>
<td></td>
</tr>
</tbody>
</table>

### Table 5. Slave Receiver Mode

<table>
<thead>
<tr>
<th>STATUS CODE</th>
<th>STATUS OF THE PC BUS AND SO1 HARDWARE</th>
<th>APPLICATION SOFTWARE RESPONSE</th>
<th>NEXT ACTION TAKEN BY SO1 HARDWARE</th>
</tr>
</thead>
<tbody>
<tr>
<td>68H</td>
<td>Over SLA+W has been received; ACK has been returned</td>
<td>No SIADT action or no S1STA action</td>
<td>Data byte will be received and NOT ACK will be returned</td>
</tr>
<tr>
<td>69H</td>
<td>Arbitration lost in SLA+W/R or not addressed SLV mode</td>
<td>No SIADT action or no S1STA action</td>
<td>Data byte will be received and NOT ACK will be returned</td>
</tr>
<tr>
<td>78H</td>
<td>General call address has been received; ACK has been returned</td>
<td>No SIADT action or no S1STA action</td>
<td>Data byte will be received and NOT ACK will be returned</td>
</tr>
<tr>
<td>88H</td>
<td>Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free.</td>
<td>No SIADT action or no S1STA action</td>
<td>Data byte will be received and NOT ACK will be returned</td>
</tr>
</tbody>
</table>
Table 5. Slave Receiver Mode

<table>
<thead>
<tr>
<th>STATUS IN CODE (SN STA)</th>
<th>STATUS OF THE FC BUS AND SO1 HARDWARE</th>
<th>TO/TOP FROMSO1 DAT</th>
<th>TO/TOP FROMSO1 CON</th>
<th>NEXT ACTION TAKEN BY SO1 HARDWARE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR</td>
<td>Open S1ADR has been recognized</td>
<td>No S1DAT action or no S1STO action</td>
<td>No S1DAT action or no S1STO action</td>
<td>Switched to not addressed SLV mode. Over SLV will be recognized. General call address will be recognized if S1ADR.0 = logic 1. Switched to not addressed SLV mode. Over SLV will be recognized. General call address will be recognized if S1ADR.0 = logic 1.</td>
</tr>
<tr>
<td>ARH</td>
<td>Arbitration lost in SLV/REC or SLV/TRAN</td>
<td>Load data byte/y</td>
<td>Load data byte/y</td>
<td>Last data byte will be transmitted and ACK will be received. Data byte will be transmitted. ACK will be received.</td>
</tr>
<tr>
<td>BH</td>
<td>Data byte in S1DAT has been transmitted, NOT ACK has been received</td>
<td>No S1DAT action or no S1STO action</td>
<td>No S1DAT action or no S1STO action</td>
<td>Switched to not addressed SLV mode. Over SLV will be recognized. General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.</td>
</tr>
<tr>
<td>CS</td>
<td>Lost data byte in S1DAT has been transmitted</td>
<td>No S1DAT action or no S1STO action</td>
<td>No S1DAT action or no S1STO action</td>
<td>Switched to not addressed SLV mode. Over SLV will be recognized. General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.</td>
</tr>
</tbody>
</table>

Table 6. Slave Transmitter Mode

<table>
<thead>
<tr>
<th>STATUS IN CODE (SN STA)</th>
<th>STATUS OF THE FC BUS AND SO1 HARDWARE</th>
<th>TO/TOP FROMSO1 DAT</th>
<th>TO/TOP FROMSO1 CON</th>
<th>NEXT ACTION TAKEN BY SO1 HARDWARE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR</td>
<td>Open S1ADR has been recognized</td>
<td>No S1DAT action or no S1STO action</td>
<td>No S1DAT action or no S1STO action</td>
<td>Switched to not addressed SLV mode. Over SLV will be recognized. General call address will be recognized if S1ADR.0 = logic 1. Switched to not addressed SLV mode. Over SLV will be recognized. General call address will be recognized if S1ADR.0 = logic 1.</td>
</tr>
<tr>
<td>ARH</td>
<td>Arbitration lost in SLV/REC or SLV/TRAN</td>
<td>Load data byte/y</td>
<td>Load data byte/y</td>
<td>Last data byte will be transmitted and ACK will be received. Data byte will be transmitted. ACK will be received.</td>
</tr>
<tr>
<td>BH</td>
<td>Data byte in S1DAT has been transmitted, NOT ACK has been received</td>
<td>No S1DAT action or no S1STO action</td>
<td>No S1DAT action or no S1STO action</td>
<td>Switched to not addressed SLV mode. Over SLV will be recognized. General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.</td>
</tr>
<tr>
<td>CS</td>
<td>Lost data byte in S1DAT has been transmitted</td>
<td>No S1DAT action or no S1STO action</td>
<td>No S1DAT action or no S1STO action</td>
<td>Switched to not addressed SLV mode. Over SLV will be recognized. General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.</td>
</tr>
</tbody>
</table>

Table 7. Slave Transmitter Mode

<table>
<thead>
<tr>
<th>STATUS IN CODE (SN STA)</th>
<th>STATUS OF THE FC BUS AND SO1 HARDWARE</th>
<th>TO/TOP FROMSO1 DAT</th>
<th>TO/TOP FROMSO1 CON</th>
<th>NEXT ACTION TAKEN BY SO1 HARDWARE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR</td>
<td>Open S1ADR has been recognized</td>
<td>No S1DAT action or no S1STO action</td>
<td>No S1DAT action or no S1STO action</td>
<td>Switched to not addressed SLV mode. Over SLV will be recognized. General call address will be recognized if S1ADR.0 = logic 1. Switched to not addressed SLV mode. Over SLV will be recognized. General call address will be recognized if S1ADR.0 = logic 1.</td>
</tr>
<tr>
<td>ARH</td>
<td>Arbitration lost in SLV/REC or SLV/TRAN</td>
<td>Load data byte/y</td>
<td>Load data byte/y</td>
<td>Last data byte will be transmitted and ACK will be received. Data byte will be transmitted. ACK will be received.</td>
</tr>
<tr>
<td>BH</td>
<td>Data byte in S1DAT has been transmitted, NOT ACK has been received</td>
<td>No S1DAT action or no S1STO action</td>
<td>No S1DAT action or no S1STO action</td>
<td>Switched to not addressed SLV mode. Over SLV will be recognized. General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.</td>
</tr>
<tr>
<td>CS</td>
<td>Lost data byte in S1DAT has been transmitted</td>
<td>No S1DAT action or no S1STO action</td>
<td>No S1DAT action or no S1STO action</td>
<td>Switched to not addressed SLV mode. Over SLV will be recognized. General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.</td>
</tr>
</tbody>
</table>

Table 8. Slave Transmitter Mode

<table>
<thead>
<tr>
<th>STATUS IN CODE (SN STA)</th>
<th>STATUS OF THE FC BUS AND SO1 HARDWARE</th>
<th>TO/TOP FROMSO1 DAT</th>
<th>TO/TOP FROMSO1 CON</th>
<th>NEXT ACTION TAKEN BY SO1 HARDWARE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR</td>
<td>Open S1ADR has been recognized</td>
<td>No S1DAT action or no S1STO action</td>
<td>No S1DAT action or no S1STO action</td>
<td>Switched to not addressed SLV mode. Over SLV will be recognized. General call address will be recognized if S1ADR.0 = logic 1. Switched to not addressed SLV mode. Over SLV will be recognized. General call address will be recognized if S1ADR.0 = logic 1.</td>
</tr>
<tr>
<td>ARH</td>
<td>Arbitration lost in SLV/REC or SLV/TRAN</td>
<td>Load data byte/y</td>
<td>Load data byte/y</td>
<td>Last data byte will be transmitted and ACK will be received. Data byte will be transmitted. ACK will be received.</td>
</tr>
<tr>
<td>BH</td>
<td>Data byte in S1DAT has been transmitted, NOT ACK has been received</td>
<td>No S1DAT action or no S1STO action</td>
<td>No S1DAT action or no S1STO action</td>
<td>Switched to not addressed SLV mode. Over SLV will be recognized. General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.</td>
</tr>
<tr>
<td>CS</td>
<td>Lost data byte in S1DAT has been transmitted</td>
<td>No S1DAT action or no S1STO action</td>
<td>No S1DAT action or no S1STO action</td>
<td>Switched to not addressed SLV mode. Over SLV will be recognized. General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.</td>
</tr>
</tbody>
</table>
Table 7. Miscellaneous States

<table>
<thead>
<tr>
<th>STATUS CODE</th>
<th>STATUS OF APPLICATION SOFTWARE RESPONSE</th>
<th>TO PROM/ST DAT</th>
<th>TO S1CON</th>
<th>NEXT ACTION TAKEN BY SIO1 HARDWARE</th>
</tr>
</thead>
<tbody>
<tr>
<td>F8H</td>
<td>No relevant state information available; SI = 0</td>
<td>No SIO1 action</td>
<td>No SIO1 action</td>
<td>Wait or proceed current transfer</td>
</tr>
<tr>
<td>00H</td>
<td>Bus error during MST or selected slave modes, due to an illegal START or STOP condition. State 00H can also occur when interference causes SIO1 to enter an undefined state.</td>
<td>No SIO1 action</td>
<td>0 1 0 X</td>
<td>Only the internal hardware is affected in the MST or addressed SLV modes. In all cases, the bus is released and SIO1 is switched to the next addressed SLV mode. STO is reset.</td>
</tr>
</tbody>
</table>

Figure 21. Simultaneous Repeated START Conditions from 2 Masters

Figure 22. Forced Access to a Busy I2C Bus

Figure 23. Recovering from a Bus Obstruction Caused by a Low Level on SDA
Interrupts

The 8XC552 has fifteen interrupt sources, each of which can be assigned one of two priority levels, as shown in Figure 27. The interrupt sources common to both the 80C51 and the 8XC552 are the external interrupts (IT0 and IT1), the timer 0 and timer 1 interrupts (T0 and T1), and the serial interrupt (RI or TI). In the 80C51, the standard serial interrupt is called SIO0. Since the subsystems which create these interrupts are identical on both parts, their functionality is likewise identical. The only differences are the locations of the enable and priority register configurations and the priority structure. This is detailed below along with the specifics of the interrupts unique to the 8XC552.

The eight Timer T2 interrupts are generated by flags CTI0-CTI13, CMI0-CMI2, and by the logical OR of flags T2OV and T2BO. Flags CTIO-CTI13 are set when an input signal to the Timer T2 matches a count value in the Timer T2 Compare registers C1, C2, C3, and C4. When an 8-bit or 16-bit overflow occurs, flags T2OV and T2BO are set, respectively. These same flags are not cleared by hardware and must be reset by software to avoid recurring interrupts.

The ADC interrupt is generated by the ADC flag in the ADC control register (ADCON). This flag is set when an ADC conversion result is ready to be read. ADCI is not cleared by hardware and must be reset by software to avoid recurring interrupts.

The SIO1 (Interrupt 0) is generated by the SI flag in the SCI control register (SCICON). This flag is set when S1STA is loaded with a valid status code.

The AOC flag may be disabled by software. It cannot be set by software. All other flags that generate interrupts may be set or cleared by software, and the effect of setting or resetting the flags by hardware. Thus, interrupts may be generated by software and pending interrupts can be canceled by software.

Interrupt Enable Registers: Each interrupt source can be individually enabled or disabled by setting or clearing a bit in the interrupt enable special function registers IEN0 and IEN1. All interrupt sources can also be globally enabled or disabled by setting or clearing bit EA in IEN0. The interrupt enable registers are described in Figures 38 and 39.

Interrupt Priority Structure: Each interrupt source can be assigned one of two priority levels. Interrupt priority levels are defined by the interrupt priority special function registers IP0 and IP1. IP0 and IP1 are described in Figure 38 and 39.

Interrupt priority levels are as follows:
- 0 — low priority
- 1 — high priority

A low priority interrupt may be interrupted by a high priority interrupt. A high priority interrupt cannot be interrupted by any other interrupt source. If two requests of different priority occur simultaneously, the high priority request is serviced. If requests of the same priority are received simultaneously, an internal polling sequence determines which request is serviced. Thus, within each priority level, there is a second priority structure determined by the polling sequence. This second priority structure is shown in Table 9.

The above Priority Within Level structure is only used when there are simultaneous requests of the same priority level.

Interrupt Handling: The interrupt sources are sampled at SOS of every machine cycle. The samples are polled during the following machine cycle. If one of the flags was in a set condition at SOS of the previous machine cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine, provided this hardware-generated LCALL is not blocked by any of the following conditions:

1. An interrupt of higher or equal priority level is already in progress.
2. The current machine cycle is not the final cycle in the execution of the instruction in progress (this interrupt request will be serviced until the instruction in progress is completed).
3. The instruction in progress is RETI or any access to the interrupt priority or interrupt enable registers. This interrupt will be serviced after RETI or after a read or write to IP0, IP1, IE0, or IE1 until at least one other instruction has been subsequently executed.

The polling cycle is repeated with every machine cycle, and the values polled are the values present at SOS of the previous machine cycle. Note that if an interrupt flag is active but is not being responded to because of one of the above conditions, and if the flag is inactive when the blocking condition is removed, then the blocked interrupt will not be serviced. Thus, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is necessary.

The processor acknowledgment an interrupt request by executing a hardware-generated LCALL to the appropriate service routine. In some cases it also clears the flag which generated the interrupt, and in others it does not. It clears the Timer 0, Timer 1, and external interrupt flags. An external interrupt flag (ISO or INT1) is cleared only if it was transition-activated. If all other interrupt flags are not cleared by hardware and must be cleared by the software. The LCALL pushes the contents of the program counter on the stack (but it does not save the PSW) and reloads the PC with an address that depends on the source of the interrupt being vectored to as shown in Table 9.

Execution proceeds from the vector address until the RETI instruction is encountered. The RETI instruction clears the “priority level active” flag. If an interrupt was acknowledged, it then pops the top two bytes from the stack and reloads the program counter. Execution of the interrupted program continues from where it was interrupted.
**80C51 Family Derivatives 8XC552/562 Overview**

**Figure 28. Interrupt Enable Register (IEN0)**

<table>
<thead>
<tr>
<th>BIT</th>
<th>SYMBOL</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEN0.7</td>
<td>EA</td>
<td>Global enable/disable control</td>
</tr>
<tr>
<td>IEN0.6</td>
<td>EAD</td>
<td>Enable ADC interrupt</td>
</tr>
<tr>
<td>IEN0.5</td>
<td>ES1</td>
<td>Enable SIO1 (I2C) interrupt</td>
</tr>
<tr>
<td>IEN0.4</td>
<td>ES0</td>
<td>Enable SIO0 (UART) interrupt</td>
</tr>
<tr>
<td>IEN0.3</td>
<td>ET1</td>
<td>Enable Timer 1 interrupt</td>
</tr>
<tr>
<td>IEN0.2</td>
<td>EX1</td>
<td>Enable External interrupt 1</td>
</tr>
<tr>
<td>IEN0.1</td>
<td>ET0</td>
<td>Enable Timer 0 interrupt</td>
</tr>
<tr>
<td>IEN0.0</td>
<td>EX0</td>
<td>Enable External interrupt 0</td>
</tr>
</tbody>
</table>

In all cases, if the enable bit is 0, then the interrupt is disabled, and if the enable bit is 1, then the interrupt is enabled.

**Figure 29. Interrupt Enable Register (IEN1)**

<table>
<thead>
<tr>
<th>BIT</th>
<th>SYMBOL</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEN1.7</td>
<td>ET2</td>
<td>Enable Timer 2 overflow interrupt(s)</td>
</tr>
<tr>
<td>IEN1.6</td>
<td>ECM2</td>
<td>Enable T2 Comparator 2 interrupt</td>
</tr>
<tr>
<td>IEN1.5</td>
<td>ECM1</td>
<td>Enable T2 Comparator 1 interrupt</td>
</tr>
<tr>
<td>IEN1.4</td>
<td>ECM0</td>
<td>Enable T2 Comparator 0 interrupt</td>
</tr>
<tr>
<td>IEN1.3</td>
<td>ECT3</td>
<td>Enable T2 Capture register 3 interrupt</td>
</tr>
<tr>
<td>IEN1.2</td>
<td>ECT2</td>
<td>Enable T2 Capture register 2 interrupt</td>
</tr>
<tr>
<td>IEN1.1</td>
<td>ECT1</td>
<td>Enable T2 Capture register 1 interrupt</td>
</tr>
<tr>
<td>IEN1.0</td>
<td>ECT0</td>
<td>Enable T2 Capture register 0 interrupt</td>
</tr>
</tbody>
</table>

**Table 8. Interrupt Priority Structure**

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>NAME</th>
<th>PRIORITY WITHIN LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>External interrupt 0</td>
<td>X0</td>
<td>(highest)</td>
</tr>
<tr>
<td>SIO1 (I2C)</td>
<td>S1</td>
<td></td>
</tr>
<tr>
<td>ADC completion</td>
<td>ADC</td>
<td></td>
</tr>
<tr>
<td>Timer 0 overflow</td>
<td>T0</td>
<td>(highest)</td>
</tr>
<tr>
<td>Timer 1 overflow</td>
<td>T1</td>
<td></td>
</tr>
<tr>
<td>Timer 1</td>
<td>CT1</td>
<td></td>
</tr>
<tr>
<td>Timer 1 capture 0</td>
<td>CT0</td>
<td></td>
</tr>
<tr>
<td>Timer 1 capture 1</td>
<td>CT1</td>
<td></td>
</tr>
<tr>
<td>Timer 1 capture 2</td>
<td>CT2</td>
<td></td>
</tr>
<tr>
<td>Timer 1 capture 3</td>
<td>CT3</td>
<td></td>
</tr>
<tr>
<td>Timer 2 overflow</td>
<td>T2</td>
<td></td>
</tr>
</tbody>
</table>

**Table 9. Interrupt Vector Addresses**

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>NAME</th>
<th>VECTOR ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>External interrupt 0</td>
<td>X0</td>
<td>0003H</td>
</tr>
<tr>
<td>Timer 0 overflow</td>
<td>T0</td>
<td>0000H</td>
</tr>
<tr>
<td>Timer 1 overflow</td>
<td>T1</td>
<td>0001H</td>
</tr>
<tr>
<td>Timer 1</td>
<td>CT1</td>
<td></td>
</tr>
<tr>
<td>Timer 1 capture 0</td>
<td>CT0</td>
<td></td>
</tr>
<tr>
<td>Timer 1 capture 1</td>
<td>CT1</td>
<td></td>
</tr>
<tr>
<td>Timer 1 capture 2</td>
<td>CT2</td>
<td></td>
</tr>
<tr>
<td>Timer 1 capture 3</td>
<td>CT3</td>
<td></td>
</tr>
<tr>
<td>ADC completion</td>
<td>ADC</td>
<td></td>
</tr>
<tr>
<td>Timer 0 capture 0</td>
<td>CT0</td>
<td></td>
</tr>
<tr>
<td>Timer 0 capture 1</td>
<td>CT1</td>
<td></td>
</tr>
<tr>
<td>Timer 0 capture 2</td>
<td>CT2</td>
<td></td>
</tr>
<tr>
<td>Timer 0 capture 3</td>
<td>CT3</td>
<td></td>
</tr>
<tr>
<td>Timer 2</td>
<td>T2</td>
<td></td>
</tr>
</tbody>
</table>

**80C51 Family Derivatives 8XC552/562 Overview**
PWM0 and PWM1. The pulse-width-ratio is in the range of 0 to 1.

### I/O Port Structure

The 8XC552 has six 8-bit ports. Each port consists of a latch (special function register P0 to P5), an output driver (port 0 to 4 only), Ports 0 is the same as in the 8051, with the exception of the additional functions of port 1. The parallel I/O function of port 4 is equal to that of ports 1, 2, and 3. Port 5 may be used as an input port only.

Figure 32 shows the bit latch and I/O buffer functional diagrams of the unique 8XC552 ports. A bit latch corresponds to a bit in a port's SFR and is represented as a D type flip-flop. A "read latch" signal from the CPU places the D input of the flip-flop in the internal bus. A "read pin" signal from the CPU places the actual port pin level on the internal bus. Some instructions that read a port read the actual port pin levels, and other instructions read the latch (SFR) contents.

#### Port 1 Operation

Port 1 operates the same as does in the 8051 with the exception of port lines P1.6 and P1.7, which may be selected as the SCL and SDA lines of serial port SIO1 (I2C). Because the I2C bus may be active while the device is disconnected from Vcc, these pins are provided with open drain diodes. Therefore pins P1.6 and P1.7 do not have internal pull-ups.

#### Port 5 Operation

Port 5 may be used to input an analog signal to the ADC. Unused ADC inputs may be used to input digital inputs. These inputs have an inherent hysteresis to prevent the input log from drawing excessive current from the power lines when driven by analog signals. Channel to channel crosstalk (CS) should be taken into consideration when both analog inputs and digital signals are simultaneously input to Port 5 (see B. C. characteristics in data sheet).

Port 5 is not bidirectional and may not be configured as an output port. All six ports are multifunctional, and their alternate functions are listed in Table 10. A more detailed description of these features can be found in the relevant parts of this section.

### Pulse-Width Modulated Outputs

The 8XC552 contains two pulse width modulated output channels (see Figure 33). These channels generate pulses of programmable length and intensity. The repetition frequency is defined by an 8-bit prescaler PWMP, which supplies the clock for the counter. The prescaler and counter are common to both PWM0 and PWM1. The MSB of the 8-bit counter modulus is PWM0/1.0-7} Low/high ratio of PWMn

When a compare register (PWM0 or PWM1) is loaded with a new value, the associated output is updated immediately. It does not have to wait until the end of the current counter period. Both PWM0 and PWM1 output pins are driven by push-pull drivers. These drivers are not used for any other purposes.

#### Prescaler Frequency Control Register PWMP

PWMP = PWMP - 7

#### Analog-to-Digital Converter

The analog input circuitry consists of an 8-bit input analog multiplexer and a 10-bit, straight binary, successive approximation ADC. The analog input voltage and all analog supplies are connected via separate input pins. The converter takes 50 machine cycles, i.e., 375 µs at an oscillator frequency of 16MHz, 25 µs at an oscillator frequency of 33MHz. Input voltage swing is from 0V to +5V. Because the internal DAC employs a ratiometric potentiometer, there are no discontinuities in the converter characteristic. Figure 34 shows a functional diagram of the analog input circuitry.
### Table 10. Input/Output Ports

<table>
<thead>
<tr>
<th>PORT PIN</th>
<th>ALTERNATE FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0.0</td>
<td>A0I</td>
</tr>
<tr>
<td>P0.1</td>
<td>A0I</td>
</tr>
<tr>
<td>P0.2</td>
<td>A0I</td>
</tr>
<tr>
<td>P0.3</td>
<td>A0I</td>
</tr>
<tr>
<td>P0.4</td>
<td>A0I</td>
</tr>
<tr>
<td>P0.5</td>
<td>A0I</td>
</tr>
<tr>
<td>P0.6</td>
<td>A0I</td>
</tr>
<tr>
<td>P0.7</td>
<td>A0I</td>
</tr>
<tr>
<td></td>
<td>Multiplexed lower order address/data bus used during external memory accesses</td>
</tr>
<tr>
<td>P1.0</td>
<td>CT0I</td>
</tr>
<tr>
<td>P1.1</td>
<td>CT1I</td>
</tr>
<tr>
<td>P1.2</td>
<td>CT2I</td>
</tr>
<tr>
<td>P1.3</td>
<td>CT3I</td>
</tr>
<tr>
<td>P1.4</td>
<td>T2I</td>
</tr>
<tr>
<td>P1.5</td>
<td>RT2</td>
</tr>
<tr>
<td>P1.6</td>
<td>SCL</td>
</tr>
<tr>
<td>P1.7</td>
<td>SDA</td>
</tr>
<tr>
<td></td>
<td>Capture timer input signals for timer T2</td>
</tr>
<tr>
<td>P2.0</td>
<td>A8I</td>
</tr>
<tr>
<td>P2.1</td>
<td>A9I</td>
</tr>
<tr>
<td>P2.2</td>
<td>A10I</td>
</tr>
<tr>
<td>P2.3</td>
<td>A11I</td>
</tr>
<tr>
<td>P2.4</td>
<td>A12I</td>
</tr>
<tr>
<td>P2.5</td>
<td>A13I</td>
</tr>
<tr>
<td>P2.6</td>
<td>A14I</td>
</tr>
<tr>
<td>P2.7</td>
<td>A15I</td>
</tr>
<tr>
<td></td>
<td>High order address/data used during external memory accesses</td>
</tr>
<tr>
<td>P3.0</td>
<td>RXD</td>
</tr>
<tr>
<td>P3.1</td>
<td>TXD</td>
</tr>
<tr>
<td>P3.2</td>
<td>T0I</td>
</tr>
<tr>
<td>P3.3</td>
<td>T1I</td>
</tr>
<tr>
<td>P3.4</td>
<td>T1I</td>
</tr>
<tr>
<td>P3.5</td>
<td>T2I</td>
</tr>
<tr>
<td>P3.6</td>
<td>T2I</td>
</tr>
<tr>
<td>P3.7</td>
<td>T3I</td>
</tr>
<tr>
<td></td>
<td>Serial port UART</td>
</tr>
<tr>
<td>P4.0</td>
<td>CMSPSH</td>
</tr>
<tr>
<td>P4.1</td>
<td>CMSPSR</td>
</tr>
<tr>
<td>P4.2</td>
<td>CMSPSH</td>
</tr>
<tr>
<td>P4.3</td>
<td>CMSPSR</td>
</tr>
<tr>
<td>P4.4</td>
<td>CMSPSH</td>
</tr>
<tr>
<td>P4.5</td>
<td>CMSPSR</td>
</tr>
<tr>
<td>P4.6</td>
<td>CMSPSH</td>
</tr>
<tr>
<td>P4.7</td>
<td>CMSPSR</td>
</tr>
<tr>
<td></td>
<td>Timer T2 compare and set/reset outputs on a match with timer T2</td>
</tr>
<tr>
<td>P5.0</td>
<td>ADC0</td>
</tr>
<tr>
<td>P5.1</td>
<td>ADC1</td>
</tr>
<tr>
<td>P5.2</td>
<td>ADC2</td>
</tr>
<tr>
<td>P5.3</td>
<td>ADC3</td>
</tr>
<tr>
<td>P5.4</td>
<td>ADC4</td>
</tr>
<tr>
<td>P5.5</td>
<td>ADC5</td>
</tr>
<tr>
<td>P5.6</td>
<td>ADC6</td>
</tr>
<tr>
<td>P5.7</td>
<td>ADC7</td>
</tr>
<tr>
<td></td>
<td>Eight analogue ADC inputs</td>
</tr>
</tbody>
</table>

---

**Figure 33. Functional Diagram of Pulse Width Modulated Outputs**

**Figure 34. Functional Diagram of Analog Input Circuitry**
The voltage slew rate must be less than 10V/μs in order to prevent an undefined result.

If the input voltage is greater than VDAC, then the bit remains set; otherwise it is cleared. The conversion continues bit by bit until all ten bits have been tested, at which stage the result of the conversion is held in the successive approximation register. Figure 36 shows a conversion flow chart. The bit pointer identifies the bit under test. The conversion takes four machine cycles per bit.

The upper 8 bits of the result are held in special function registers ADCI and ADCON.6 (ADC.0). The user may ignore the two least significant bits in ADCON and use the ADC as an 8-bit converter (8 upper bits in ADCH). In any event, the total actual conversion time is 50 machine cycles for the 8XC552 or 24 machine cycles for the 8XC562. ADCS can be set by software only or by either hardware or software.

Control bits ADCON.0, ADCON.1, and ADCON.2 are used to control an analog multiplexer which selects one of eight analog channels (ADCON.2 as above or by applying a rising edge to external pin STADC). When a conversion is started by setting either ADCON.3 (ADCS) or ADCON.5 = 1, and a conversion may be started by setting ADCON.3 (as above or by applying a rising edge to external pin STADC). When a conversion is started by setting ADCON.3 (as above or by applying a rising edge to external pin STADC), the conversion commences at the beginning of the next cycle. When a conversion is initiated by software, the conversion starts at the beginning of the machine cycle which follows the instruction that sets ADCS. ADCS is actually implemented with two flip-flops: a command flip-flop which is affected by set operations, and a status flag flip-flop which is accessed during read operations.

During the next eight machine cycles, the voltage at the previously selected pin is sampled, and the input voltage may be stable in order to obtain an useful sample. In any event, the input voltage slew rate must be less than 10V/μs in order to prevent an undefined result.

The successive approximation control logic checks the most significant bit and then all other bits in the successive approximation register (10 0000 0000B). If the input voltage is greater than 14DAC, then the bit is set; otherwise it is cleared.

The successive approximation control logic now sets the next most significant bit (1 110000 0000B or 01 0000 0000B, depending on the previous result) and VDAC is compared to Vin again. If the input voltage is greater than 14DAC, then the bit being tested remains set; otherwise it is cleared.

During the next eight machine cycles, the voltage at the previously selected pin is sampled, and the input voltage may be stable in order to obtain an useful sample. In any event, the input voltage slew rate must be less than 10V/μs in order to prevent an undefined result.

The successive approximation control logic checks the most significant bit and then all other bits in the successive approximation register (10 0000 0000B). If the input voltage is greater than 14DAC, then the bit is set; otherwise it is cleared.

The successive approximation control logic now sets the next most significant bit (1 110000 0000B or 01 0000 0000B, depending on the previous result) and VDAC is compared to Vin again. If the input voltage is greater than 14DAC, then the bit being tested remains set; otherwise it is cleared.

The successive approximation control logic checks the most significant bit and then all other bits in the successive approximation register (10 0000 0000B). If the input voltage is greater than 14DAC, then the bit is set; otherwise it is cleared.

The successive approximation control logic now sets the next most significant bit (1 110000 0000B or 01 0000 0000B, depending on the previous result) and VDAC is compared to Vin again. If the input voltage is greater than 14DAC, then the bit being tested remains set; otherwise it is cleared.

The successive approximation control logic checks the most significant bit and then all other bits in the successive approximation register (10 0000 0000B). If the input voltage is greater than 14DAC, then the bit is set; otherwise it is cleared.

The successive approximation control logic now sets the next most significant bit (1 110000 0000B or 01 0000 0000B, depending on the previous result) and VDAC is compared to Vin again. If the input voltage is greater than 14DAC, then the bit being tested remains set; otherwise it is cleared.

The successive approximation control logic checks the most significant bit and then all other bits in the successive approximation register (10 0000 0000B). If the input voltage is greater than 14DAC, then the bit is set; otherwise it is cleared.

The successive approximation control logic now sets the next most significant bit (1 110000 0000B or 01 0000 0000B, depending on the previous result) and VDAC is compared to Vin again. If the input voltage is greater than 14DAC, then the bit being tested remains set; otherwise it is cleared.

The successive approximation control logic checks the most significant bit and then all other bits in the successive approximation register (10 0000 0000B). If the input voltage is greater than 14DAC, then the bit is set; otherwise it is cleared.

The successive approximation control logic now sets the next most significant bit (1 110000 0000B or 01 0000 0000B, depending on the previous result) and VDAC is compared to Vin again. If the input voltage is greater than 14DAC, then the bit being tested remains set; otherwise it is cleared.

The successive approximation control logic checks the most significant bit and then all other bits in the successive approximation register (10 0000 0000B). If the input voltage is greater than 14DAC, then the bit is set; otherwise it is cleared.

The successive approximation control logic now sets the next most significant bit (1 110000 0000B or 01 0000 0000B, depending on the previous result) and VDAC is compared to Vin again. If the input voltage is greater than 14DAC, then the bit being tested remains set; otherwise it is cleared.

The successive approximation control logic checks the most significant bit and then all other bits in the successive approximation register (10 0000 0000B). If the input voltage is greater than 14DAC, then the bit is set; otherwise it is cleared.

The successive approximation control logic now sets the next most significant bit (1 110000 0000B or 01 0000 0000B, depending on the previous result) and VDAC is compared to Vin again. If the input voltage is greater than 14DAC, then the bit being tested remains set; otherwise it is cleared.

The successive approximation control logic checks the most significant bit and then all other bits in the successive approximation register (10 0000 0000B). If the input voltage is greater than 14DAC, then the bit is set; otherwise it is cleared.
ADC Resolution and Analog Supply: Figure 38 shows how the ADC is realized. The ADC has its own supplies (AVDD and AVSS) and is connected to a resistance-ladder with 1023 equally spaced taps, separated by a resistance of R. The first tap is located 0.5 x R above Vref–, and the last tap is located 1.5 x R below Vref+. This structure ensures that the DAC is monotonic and results in a symmetrical quantization error as shown in Figure 40.

For input voltages between Vref– and (Vref– + 1/2 LSB), the 10-bit result of an A/D conversion will be 00 0000 0000B = 000H. For input voltages between (Vref+) – 3/2 LSB and Vref+, the result of a conversion will be 11 1111 1111B = 3FFH. AVref+ and AVref– may be between AVDD + 0.2V and AVSS – 0.2V. AVref+ should be positive with respect to AVref–, and the input voltage (Vin) should be between AVref+ and AVref–. If the analog input voltage range is from 2V to 4V, then 10-bit resolution can be obtained over this range if AVref+ = 4V and AVref– = 2V.

The result can always be calculated from the following formula:

\[ \text{Result} = \frac{V_{\text{IN}} - \text{AVref}}{\text{AVref}} \times 1024 \]

Power Reduction Modes:

The 8XC552 has two reduced power modes of operation: the idle mode and the power-down mode. These modes are entered by setting bits in the PCON special function register. When the 8XC552 enters the idle mode, the following functions are disabled:

- CPU (halted)
- Timer T2 (halted and reset)
- PWM0, PWM1 (reset; outputs are high)
- ADC (conversion aborted if in progress).

In idle mode, the following functions remain active:

- Timer 0
- Timer 1
- Timer T3
- SIO0, SIO1
- External interrupts

When the 8XC552 enters the power-down mode, the oscillator is stopped. The power-down mode is entered by setting the PD bit in the PCON register. The PD bit can only be set if the EW input is tied HIGH.

NOTE:

Because the analog to digital converter has a sampled-data comparator, the input looks capacitive to a source. When a conversion is initiated, switch Sm closes for 8tcy (8 \( \mu \)s @ 12MHz crystal frequency) during which time capacitance Cs + Cc is charged. It should be noted that the sampling causes the analog input to present a varying load to an analog source.
Power-Down Mode: The instruction that sets PCON.1 will be the last instruction executed in the normal operating mode before the power-down mode is entered. In the power-down mode, the on-chip oscillator is stopped. This freezes all functions, only the on-chip RAM and special function registers are held. The port pins output the contents of their respective special function registers. A hardware reset is the only way to terminate the power-down mode. Reset redefines all the special function registers, but does not change the on-chip RAM.

In the power-down mode, VDD and AVDD must not be reduced below the power-down mode threshold. The reset that terminates the power-down mode also freezes the oscillator. The reset should not be activated before VDD and AVDD are restored to their normal operating level, and must be held active long enough to allow the oscillator to start and stabilize (more than 10ms).

The status of the external pins during power-down is shown in Table 11. If the power-down mode is entered while the 8XC552 is executing out of external program memory, the data that is held in the P2 special function register is restored to port 2. If a port latch contains a "1", the port pin is held HIGH during the power-down mode by the strong pull-up translation.

Power-Control Register (PCON): The Idle and power-down modes are initiated by writing to bits in PCON. PCON is not bit addressable. See Figure 41.

Memory Organization
The memory organization of the 8XC552 is the same as in the 80C51, with the exception that the 8XC552 has 9k bytes of ROM, 256 bytes of RAM, and additional SFRs. Addressing modes are the same in the 8XC552 and the 80C51. Details of the differences are given in the following paragraphs.

In the 80C552, the last 4k of the 51k program memory address space is filled by internal ROM. By tying the MCLR pin high, the processor fetches instructions from internal program ROM. Bus expansion for accessing program memory from 9k upwards is automatic since internal instruction fetches occur automatically when the program counter exceeds 8191. If the MCLR is tied to VSS, all program memory fetches are from external memory. The execution speed of the 8XC552 is the same regardless of whether fetches are from external or internal program memory. If all storage is on-chip, the byte location 8191 should be left vacant to prevent an undefined fetch from external program memory address 8192.

Certain locations in program memory are reserved for specific functions. Locations 0000H to 0004H are reserved for initialization. Locations 0004H to 000FH are reserved for the I/O space interrupt request service routines.

Functionally, the internal data memory is the most flexible of the address spaces. The internal data memory space is subdivided into a 256-byte internal RAM address space and a 128-byte special function register (SFR) address space, as shown in Figure 42.

The internal RAM address space is 0 to 255. For 8-bit register banks, the on-chip data RAM occupies locations 0 to 31. 128 bytes of the internal data RAM have addresses accessible by direct addressing. The stack can reside in 128 bytes of internal data RAM at locations 20H to 27H. These are the registers located anywhere in the internal data RAM address space by loading the 8-bit stack pointer. The stack depth may be 256 bytes maximum.

The SFR address space is 128 to 255. All registers except the program counter and the four 8-bit register banks reside in this address space. Memory-mapping the SFRs all from 255 to 0 serves as an easy exit to internal RAM, and as such, they can be operated on by most instructions. The SFRs are listed in Figure 43 and, when the 80C51 is addressing the SFR space, they appear in Figure 44-46.

The memory address space in the 80C51 and 8XC552 are summarized in Figure 46. The special function bit addresses are summarized in Figure 47.

---

**Table 11. External Pin Status During Idle and Power-Down Modes**

<table>
<thead>
<tr>
<th>MODE</th>
<th>MEMORY</th>
<th>ALLE</th>
<th>PORT 0</th>
<th>PORT 1</th>
<th>PORT 2</th>
<th>PORT 3</th>
<th>PORT 4</th>
<th>PIMMA/PIMMF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle (1)</td>
<td>Internal 1 1  Port data Port data Port data Port data Port data HIGH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Idle (2)</td>
<td>External 1 1 Floating Port data Port data Port data Port data Port data HIGH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power-down</td>
<td>External 0 0 Port data Port data Port data Port data Port data Port data HIGH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Figure 41. Power Control Register (PCON)**

**Figure 42. Internal Data Memory Address Space**

**Figure 43. Special Function Registers**
Figure 45. Bit and Byte Addressing Overview of Internal Data Memory

Figure 46. RAM Bit Addresses

Figure 47. Special Function Register Bit Addresses
APPENDIX C: ALPHABETICAL INDEX

A
A/D  2, 6, 9, 14, 26, 45
ABACO® 2, 26
ANALOG INPUT TYPE SELECTION 25

B
BACK UP 17, 33
BIBLIOGRAPHY 50
BUZZER 2, 6, 9, 42

C
CLOCK 2
CLOCK DEVICE 3
CONNECTIONS 11
CONNECTORS 9
   CN1 12
   CN2 13
   CN3 11
   CN4 14
   CN5 16
   CN6 18
   CN7 20
   CN8 17
CONTROL LOGIC 7
CPU 3, 9
CPU PORT 14
CURRENT LOOP 2, 4, 20, 24, 32

D
DIP SWITCH 6, 9, 43

E
EEPROM 2, 4, 6, 9
ELECTRIC FEATURES 10
EPROM 2, 4, 9
EXTERNAL BACK-UP BATTERY 10
EXTERNAL DEVICES 46

F
FUSE F2 10

G
GENERAL FEATURES 2, 9
GENERAL OUTPUT 12, 42
I
I/O 2, 18
I/O ADDRESSES 36
I/O CONNECTION 24
FC 2, 9, 16
IDLE MODE 2
INSTALLATION 11
J
JUMPERS 27
  2 PINS JUMPERS 28
  3 PINS JUMPERS 30
  4 PINS JUMPER 31
  5 PINS JUMPER 31
L
LED 9, 22, 42
LOCAL USER INTERFACES 26
M
MEMORY 9
MEMORY DEVICES 4
MEMORY SELECTION 33
O
ON BOARD INPUT 24
P
P1 7
POWER DOWN MODE 2
POWER SUPPLY 3, 11
POWER SUPPLY SELECTION 25
POWER SUPPLY TENSION 10
PPI 82C55 6, 12, 13, 44
PULL UP/DOWN 31, 43
PWM 2, 9, 12, 45
R
RAM 2, 4, 33, 44
REAL TIME CLOCK 9
REGISTERS
  BUZ 42
  DSW1 43
  EPUL 43
  LD2 42
  OUTG 42
RELATIVE HUMIDITY 9
RESET 7, 24
RS 232 2, 4, 9, 16, 20, 24, 32
RS 422 24, 32
RS 422-485  2, 4, 9, 20, 43  
RS 485      24, 32  
RTC        2, 4, 33, 44  
S  
SERIAL COMMUNICATION  20, 32  
SERIAL EEPROM       44  
SIO          45  
SIO 0        6  
SIO 1        6  
SIZE         2, 9  
SOCKETS      2  
SOFTWARE     2, 34  
T  
TECHNICAL FEATURES  9  
TEMPERATURE RANGE  9  
TEST POINT      24  
TIMER COUNTER   45  
TRIMMERS AND CALIBRATION  26  
V  
VREF         24, 26  
W  
WATCH DOG     2, 6, 9, 45  
WEIGHT       9