THESIS

DESIGN OF A UNIVERSAL TEST PLATFORM FOR
RADIATION TESTING OF DIGITAL COMPONENTS

by

Duane E. Amsler Jr.

September 1996

Thesis Advisor: Douglas Fouts

Approved for public release; distribution is unlimited.
DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.
<table>
<thead>
<tr>
<th>REPORT DOCUMENTATION PAGE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AGENCY USE ONLY (Leave blank)</strong></td>
<td><strong>REPORT DATE</strong></td>
</tr>
<tr>
<td></td>
<td>September 1996</td>
</tr>
<tr>
<td><strong>REPORT TYPE AND DATES COVERED</strong></td>
<td><strong>FUNDING NUMBERS</strong></td>
</tr>
<tr>
<td></td>
<td>Master’s Thesis</td>
</tr>
<tr>
<td><strong>DESIGN OF A UNIVERSAL TEST PLATFORM FOR RADIATION TESTING OF DIGITAL COMPONENTS</strong></td>
<td></td>
</tr>
<tr>
<td><strong>AUTHOR(S)</strong></td>
<td>Duane E. Amsler Jr., CPT USA</td>
</tr>
<tr>
<td><strong>PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</strong></td>
<td><strong>PERFORMING ORGANIZATION REPORT NUMBER</strong></td>
</tr>
<tr>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td>Monterey CA 93943-5000</td>
<td></td>
</tr>
<tr>
<td><strong>SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</strong></td>
<td><strong>SPONSORING/MONITORING AGENCY REPORT NUMBER</strong></td>
</tr>
<tr>
<td>Naval Research Laboratory, Washington, D.C.</td>
<td></td>
</tr>
<tr>
<td><strong>SUPPLEMENTARY NOTES</strong></td>
<td>The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.</td>
</tr>
<tr>
<td><strong>DISTRIBUTION/AVAILABILITY STATEMENT</strong></td>
<td>Further distribution only as directed by Superintendent, Code 043, Naval Postgraduate School, Monterey, CA 93940, or higher DoD Authority.</td>
</tr>
<tr>
<td><strong>ABSTRACT (maximum 200 words)</strong></td>
<td>In this research, programmable, microcontroller-based test hardware was designed, constructed, debugged, and programmed. The wire-wrapped board will be used to test two custom static random access memory (SRAM) chips, as well as other custom chips designed at the Naval Postgraduate School. Components for the test hardware were selected to allow prototyping with standard parts that can later be replaced with radiation hardened parts as budgets permit. Control of the test hardware is via a RS-232 serial interface, which allows remote control programming and monitoring of the test hardware and device being tested.</td>
</tr>
<tr>
<td><strong>SUBJECT TERMS</strong></td>
<td>Radiation Testing, 8051, Programmable Test Device, LT GaAs, Radiation Hardened, RAD Hard</td>
</tr>
<tr>
<td><strong>SECURITY CLASSIFICATION OF REPORT</strong></td>
<td><strong>SECURITY CLASSIFICATION OF THIS PAGE</strong></td>
</tr>
<tr>
<td>SECRET</td>
<td>Unclassified</td>
</tr>
<tr>
<td><strong>SECURITY CLASSIFICATION OF ABSTRACT</strong></td>
<td><strong>LIMITATION OF ABSTRACT</strong></td>
</tr>
<tr>
<td>Unclassified</td>
<td></td>
</tr>
<tr>
<td><strong>NUMBER OF PAGES</strong></td>
<td>194</td>
</tr>
<tr>
<td><strong>PRICE CODE</strong></td>
<td></td>
</tr>
</tbody>
</table>

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 299-18 298-102
DESIGN OF A UNIVERSAL TEST PLATFORM FOR RADIATION TESTING OF DIGITAL COMPONENTS

Duane E. Amsler Jr.
Captain, United States Army
B.S., Clarkson University, 1986

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

September 1996

Author: 

Duane E. Amsler Jr.

Approved by: 

Douglas J. Fouts, Thesis Advisor

Todd R. Weatherford, Second Reader

Herschel H. Loomis, Jr., Chairman,
Department of Electrical and Computer Engineering
ABSTRACT

In this research, programmable, microcontroller-based test hardware was designed, constructed, debugged, and programmed. The wire-wrapped board will be used to test two custom static random access memory (SRAM) chips, as well as other custom chips designed at the Naval Postgraduate School. Components for the test hardware were selected to allow prototyping with standard parts that can later be replaced with radiation hardened parts as budgets permit. Control of the test hardware is via a RS-232 serial interface, which allows remote control programming and monitoring of the test hardware and device being tested.
# TABLE OF CONTENTS

I. INTRODUCTION ............................................................................................................. 1
   A. OVERVIEW .............................................................................................................. 1
   B. THESIS ORGANIZATION ....................................................................................... 1

II. THE LOW TEMPERATURE GALLIUM ARSENIDE RESEARCH PROGRAM .......... 3
   A. OVERVIEW ............................................................................................................ 3
   B. VITESSE 256 X 4 SRAM ..................................................................................... 4
   C. MOTOROLA 256 X 16 SRAM ............................................................................. 4

III. COMPONENT SUMMARY .......................................................................................... 7
   A. OVERVIEW ............................................................................................................ 7
   B. SUMMARY OF VLSI COMPONENTS ...................................................................... 8
      1. UT69RH051 Microcontroller (i8051) ................................................................. 8
      2. MAX233 ............................................................................................................. 8
      3. HS-82C55ARH (82C55) ..................................................................................... 8
      4. UT28F64 (AM2764A) ..................................................................................... 9
      5. UT67164 (61C64) ............................................................................................ 9
   C. SUMMARY OF MSI COMPONENTS ...................................................................... 9
      1. UT54ACS08 (74HC08) .................................................................................... 9
      2. UT54ACS138 (74HC138) ................................................................................ 9
      3. UT54ACS245 (74HC245) ................................................................................ 9
      4. HCS573MS (74HC573) .................................................................................. 10
      5. HCS574MS (74HC574) .................................................................................. 10
   D. DIGITAL COMPONENT ELECTRICAL REQUIREMENTS ...................................... 10

E. DISCRETE COMPONENTS .......................................................................................... 11
   1. Crystal Oscillator ............................................................................................... 11
   2. Capacitors .......................................................................................................... 11
      a. 10 µF .............................................................................................................. 11
      b. 1 µF ............................................................................................................. 11
      c. .1 µF .......................................................................................................... 12
   3. Resistors .............................................................................................................. 12

F. CONNECTORS ........................................................................................................... 12
   1. 40 Pin Header .................................................................................................... 12
   2. 25-pin sub-D .................................................................................................... 12

IV. DESIGN ANALYSIS .................................................................................................. 13
   A. OVERVIEW ............................................................................................................ 13
   B. SERIAL INTERFACE ........................................................................................... 13
      1. MAX232 ......................................................................................................... 13
      2. MAX233 ......................................................................................................... 14
      3. Serial connector .............................................................................................. 15
      4. Overall Serial Subsystem .............................................................................. 15
   C. TIMING GENERATION ......................................................................................... 16
   D. MEMORY INTERFACE ......................................................................................... 17
      1. Overview .......................................................................................................... 17
      2. XX573 ............................................................................................................ 18
      3. XX138 ............................................................................................................ 18
LIST OF FIGURES

Figure 1: Vitesse 256 X 4 SRAM ................................................................. 4
Figure 2: Motorola 256 X 16 SRAM .............................................................. 6
Figure 3: MAX232 .................................................................................. 14
Figure 4: MAX233 .................................................................................. 14
Figure 5: RS-232 Connector (minimal) ....................................................... 15
Figure 6: Complete Serial Subsystem ....................................................... 16
Figure 7: Timing Subsystem .................................................................... 17
Figure 8: Memory Interface .................................................................... 20
Figure 9: Memory Subsystem .................................................................. 22
Figure 10: 8255 Interface ....................................................................... 24
Figure 11: Byte Latch Circuitry ................................................................. 25
Figure 12: Reset Circuit .......................................................................... 26
Figure 13: Complete Circuit Diagram ...................................................... 32
Figure 14: 8255 Control Word ................................................................. 34
LIST OF TABLES

Table 1: Summary Of Component Pin Currents.................................................................11
Table 2: Memory Map........................................................................................................18
Table 3: Peripheral Memory Map....................................................................................22
Table 4: Connector 1 ........................................................................................................28
Table 5: Connector 2 ........................................................................................................28
Table 6: Connector 3 ........................................................................................................30
Table 7: Connector 4 ........................................................................................................30
Table 8: Memory Map........................................................................................................33
Table 9: Transceiver Enables .........................................................................................35
I. INTRODUCTION

A. OVERVIEW

One of the major constraints in designing for the space environment is the ability of parts to withstand radiation. Radiation hardened (RAD hard) parts are typically reengineered designs of commercially available parts, which are adapted to a RAD hard fabrication process. Traditional designs are reworked to assure immunity to such radiation effects as latchup and single-event upsets (SEUs). The additional engineering and radiation hardened process significantly increases cost and time to market. Ongoing research at the Naval Postgraduate School is developing a process that should allow off-the-shelf gallium arsenide (GaAs) integrated circuits (IC) designs to be fabricated on specially prepared wafers that contain buried, low-temperature grown, GaAs buffer layers. The intent of the program is to run these specially prepared wafers through commercial GaAs wafer processing lines to create radiation-hardened ICs, without additional IC re-engineering and special IC processing steps.

The goal of this research is to create test hardware capable of testing ICs created using low-temperature (LT) GaAs wafers for functionality, as well as immunity to single-event upsets in a radiation environment. The original requirement was to test a Vitesse 256 X 4 static random access memory (SRAM), but additional requirements to test a Motorola 256 x 16 SRAM and a custom SEU detection IC were added. The test hardware is programmable through a RS-232 serial port and can be configured to read and write any desired data and address patterns for devices with up to 80 I/O pins.

Current work in this area is sparse. Oregon State University uses a basic 8051 design for teaching microprocessor basics and have developed very good monitor, which is used as part of this design. A previous thesis by Chris Mooney[1] address interfacing these two SRAMs to a satellite test board. The Mooney thesis design is only for these specific ICs and cannot be used independent of the satellite. This thesis creates a design that is capable of testing these two SRAMs, as well as a multitude of ICs, in a radiated environment.

B. THESIS ORGANIZATION

The goal of this thesis is to document the design, construction, testing, and programming of hardware that is capable of testing ICs in a radiation environment. To this end, the following organization will be followed. Chapter II will present an overview of the LT GaAs program, as
well as a brief overview of radiation effects. Chapter III discusses the various components selected and their basic operation. Chapter IV will provide a detailed discussion of the design and operation of the test hardware. Chapter V discusses the software involved to run the hardware, as well as the code for specific tests. Chapter VI presents conclusions, as well as potential future improvements and enhancements.
II. THE LOW TEMPERATURE GALLIUM ARSENIDE RESEARCH PROGRAM

A. OVERVIEW

One of the most significant costs in developing systems to operate in the space environment is that of radiation hardening electronic components. These RAD hard ICs are specially engineered to a set of design rules that apply to a specific fabrication process. As a result of this additional engineering, special fabrication processes, and relatively low demand, RAD hard devices are typically 100 times more expensive that their commercial counterparts.

Some of the radiation effects of digital ICs include device degradation, latchup, and single event upsets. Degradation is caused by both the depositing of charge within the device and actual crystal lattice damage. Both will cause the device specifications to change and may eventually result in total failure. Latchup is a condition where a temporary short from power to ground is caused by the interaction between a transistor, a charged particle, and parasitic circuitry on the IC. If this does not destroy the device or power supply, the circuit may be powered down and restarted. A single-event upset is a condition where a particle passes through a transistor or memory element and causes the associated bit to flip, producing a data error. After this temporary data upset, the device will continue to function normally.[2]

To mitigate the effects of radiation, a number of technologies have been developed for complementary metal oxide silicon (CMOS) ICs which require the reengineering of commercial parts, as mentioned above. For high-speed applications gallium arsenide (GaAs) circuits have proven to be superior to emitter coupled logic (ECL) and bipolar complementary metal oxide silicon (BiCMOS) in resistance to degradation and latchup. However GaAs SEU rates are four to seven times greater than most Department of Defense (DoD) requirements. A great deal of research has been conducted at the Naval Postgraduate School and elsewhere to find a method of improving the SEU characteristics of GaAs. One such research effort is called Low Temperature Gallium Arsenide (LT GaAs), where a buffer layer of gallium arsenide is grown at low temperatures, around 200° C, as opposed to the 600°C that is typically used for gallium arsenide processes. It has been shown that this low-temperature buffer layer eliminates SEU effects in GaAs ICs. If successful, this process will allow specially prepared wafers to be run through
traditional GaAs processing steps with commercially developed mask sets. This will significantly reduce the cost of developing high-speed, low-power circuits for space-based applications.[3]

To this end, some LT GaAs wafers were run through commercial processes at both Vitesse and Motorola. Vitesse produced a 256 X 4 SRAM and Motorola produced a 256 X 16 SRAM, both with existing commercial mask sets. These two memories need to be tested for both functionality and resistance to single-event upsets. To this end, the design of a board capable of performing such tests is examined in this thesis. The following two sections detail the electrical characteristics and pin-outs of the LT GaAs chips to be tested.

B. VITESSE 256 X 4 SRAM

The Vitesse SRAM was produced with their commercial mask set and was packaged in a 28-pin dual flat pack. This SRAM operates in the 10 ns range and has separate data inputs and outputs. The overall interface includes 8 address, 4 data in, 4 data out, read, write, and two chip select pins, one active high and one active low. Vitesse no longer produces this SRAM as an off-the-shelf part and a third party package was needed to package the die, which is the reason for the somewhat unconventional pin-out of the package. Figure 1 shows the package pin out diagram of the Vitesse SRAM.

![Vitesse 256 X 4 SRAM](image)

**Figure 1: Vitesse 256 X 4 SRAM**

C. MOTOROLA 256 X 16 SRAM

This SRAM mask set is part of a library for digital signal processing (DSP) IC development and does not have off-chip drivers, therefore requiring additional support circuitry. Input and output voltage levels are 0 and 0.9 to 2.0 volts. Thus, voltage reduction is required for
the inputs and amplification for the outputs. The voltage reduction can be accomplished using two diodes with a resistor, and the amplification can be performed with high-speed comparators. This interface circuitry is being developed by another student at the Naval Postgraduate School.

The interface of the chip, which drives the test setup on this board, consists of 8 address, 16 data in, 16 data out, write enable, and clock lines. There are also multiple power and ground connections for the clock buffers, output drivers, and memory elements. The manufacture's data sheet states that the address and write enable lines should be changed during clock low. On a clock high, the address is decoded and data is written if the write line is enabled, or read if the write line is low.

This chip is mounted in a 68-pin leaded chip carrier, again a third party package as this is not an off-the-shelf SRAM at Motorola. Figure 2 shows the pin outs of the 68-pin package.
Figure 2: Motorola 256 X 16 SRAM
III. COMPONENT SUMMARY

A. OVERVIEW

The intent of this project was to design and build test hardware that would allow functional and radiation testing of a Vitesse 256 X 4 SRAM fabricated with the LT GaAs process. The test fixture is required to be portable and capable of interfacing to a laptop computer via a RS-232 serial port. The original considerations for designs were as follows:

- RS-232 controlled logic
- Microprocessor Based Design
- Microcontroller Based Design

Using a single, RS-232 I/O port design would have required a number of registers to control the experiment. For the Vitesse chip, which is a 256x4 SRAM, two 8-bit registers would have been needed to control chip inputs (8 bits for address, 4 bits for data, and 4 bits for control signals CS1, CS2, Write Enable, and Output Enable), and a 4-bit register would have been required for reading data out. This design, although relatively simple, is very limited and only allows for testing of this specific chip. Furthermore, addressing all the registers would require additional hardware and operation would be very slow. To test a more complex chip, other circuitry must be added, and the software correspondingly changed. Ultimately, this design methodology was scrapped due to its limited utility.

The microprocessor-based design was considered, along with the microcontroller-based design, and it was determined that a microcontroller was more flexible and capable due to the built-in features of most microcontrollers, which usually include serial support, on-board ROM, and built-in I/O.

Once it was determined that a microcontroller-based design was to be used, the question of which microcontroller arose. Since one of the design considerations was to have a test fixture that could be used in a radiation environment, the choice was narrowed down to 3 or 4 options. Ultimately, the United Technologies UT69RH051 was chosen for its compatibility with the 8051 family of microcontrollers. The 8051 family has a wealth of development tools available, which greatly aided in the development of this system.
B. SUMMARY OF VLSI COMPONENTS

Based on the choice of the UT69RH051 as the controller for the design, the following components are required. Data sheets for the following components are available in Appendix C. The first part number is the RAD hard version and the number in parenthesis is the commercial version for each IC.

1. UT69RH051 Microcontroller (i8051)

The UT69RH051 is a RAD Hard version of the Intel i8051 series of microcontrollers. This is a 40-pin device consisting of four 8-bit ports (ports 0-3), with additional pins for clock (XTAL1 and XTAL2), address latch enable (ALE), program store enable (PSEN*), external access enable (EA*), and reset (RST). Port 0 is a multiplexed port providing both the lower byte of the address and the data byte. The ALE line is asserted when the low order address byte is on port 0. The PSEN* serves as the read signal for the program memory space, while the EA* is an input indicating whether to use internal ROM (not available on the UT69RH051) or an external ROM. The XTAL1 and XTAL2 lines are used to provide timing control. The XTAL1 input can be used alone if an oscillator is used to generate timing, otherwise both XTAL lines are used with a simple crystal and capacitor combination. Port 1 may be used for special purpose I/O. However, it is used here as a basic 8-bit I/O port. Port 2 provides the high-order address byte. Port 3 provides for general I/O control to include the read (RD*) and write (WR*) signals to memory and the serial transmit (TXD) and receive (RXD) lines to the RS-232 interface.

2. MAX233

The MAX233 is a multi-channel RS-232 driver/receiver that converts serial data between 0 and +5 V (TTL) and ±10 V (RS-232). This chip has the added benefit of not requiring any additional power supply beyond the +5 V on which the chip and board operate. The only pins of interest are Tin, Tout, Rin, and Rout, which are the transmitter and receiver inputs and outputs. Other interface pins are tied together per the data sheet.

3. HS-82C55ARH (82C55)

The HS-8255ARH is a general-purpose, programmable, I/O device with 24 I/O pins organized as ports A, B, and C. Ports A and B are 8-bit ports and can each be programmed as either byte input or byte output. Port C is divided into two 4-bit I/O groups, each of which can be programmed as input or output. The port C lines can be configured to be individually set or reset.
in output mode, which aids in the simulation of control signals. For interfacing, chip select (CS*), write (WR*), read (RD*), two address lines, and 8 data lines (D0-D7) are provided. A RESET pin is also provided to allow the 8255 to be started in a known program state. Lines A0 and A1 specify ports A, B, C, or control-word access.

4. **UT28F64 (AM2764A)**

The UT28F64 is a RAD hard 8Kx8 PROM manufactured by United Technologies. This PROM will operate on 5 V and has an access time of 35ns. The interface pins are 13 address lines, 8 data lines, 3 control lines, power and ground. The three control lines are output enable (OE*), which is essentially a read line, chip select (CE*), and a programming line (PE*), which is asserted during programming and held high during operation. The 2764 has one additional pin, Vpp (pin 1) which is used during programming and is tied high for operation.

5. **UT67164 (61C64)**

This is an 8Kx8 RAD hard SRAM with a 55 ns access time. The UT67164 is also SEU hardened but is contained in a 600 mil package, as opposed the standard 300 mil used for standard SRAM's, as well as the 61C64 used on the prototype board. Interface pins are identical to the PROM with the omission of the program lines.

**C. SUMMARY OF MSI COMPONENTS**

1. **UT54ACS08 (74HC08)**

The UT54ACS08 is a RAD hard version of a quad 2 input AND gate.

2. **UT54ACS138 (74HC138)**

The UT54ACS138 is a RAD hard version of a 3-to-8 line decoder with three enable inputs, one active high, and 2 active low. When the proper input conditions are met, a single output line will go low indicating that it is selected, all other outputs remain high.

3. **UT54ACS245 (74HC245)**

The UT54AC245 is a tri-state octal bus transceiver. It is RAD hard and has an 8-bit A bus, an 8-bit B bus, an enable line, and a direction control line. As with the other devices, if the enable line is inactive (high), both buses are in the high impedance state. If the enable is asserted, data is transmitted from the A to the B side if the direction input is high, otherwise data is sent from the B to the A side.
4. HCS573MS (74HC573)

The HS573MS is an octal latch with tri-state outputs and is RAD hard. The 573 has a broadside configuration where all inputs are on one side and all outputs are on the other. The interface pins consist of 8 input bits and 8 output bits, as well as an output enable (OE*) and a latch enable (LE). When output enable is not asserted (high), the outputs are in the high impedance state. When latch enable is high, the latch is transparent. When the latch enable is asserted, the input data at that moment is held on the outputs until the latch enable goes high again.

5. HCS574MS (74HC574)

The HCS574MS is a RAD hard broadside octal D flip-flop with tri-state outputs. It has 8 input and 8 output pins, as well as an output enable (OE*) and a clock (CP) pin. On a low-to-high transition of the clock line, data on the inputs is stored and available on the outputs until the next low-to-high transition (assuming outputs are enabled).

D. DIGITAL COMPONENT ELECTRICAL REQUIREMENTS

The majority of electrical components are available in both CMOS and TTL compatible versions. For driving external components, HCT (high speed CMOS supporting TTL I/O levels) components are used, otherwise HC (high speed CMOS) components are used. Port 0 of the 8051 is capable of driving 4 TTL loads or a virtually unlimited number of CMOS loads (as the typical CMOS load draws an order of magnitude less current than TTL loads). The remainder of the 8051 ports will sink 3.5mA or source 0.3 mA, which is more than sufficient to drive 10 CMOS loads. The bus transceivers at the outputs of the XX245 will sink and source 12mA, which is sufficient to drive virtually any device to be tested. The outputs of the 8255 will sink and source 2mA, which is sufficient to drive most control lines. Further interface limitations will be discussed in Chapter V. A summary of the output currents of the devices used are shown in Table 1.
<table>
<thead>
<tr>
<th>Component</th>
<th>$I_{OL} (\text{mA})$</th>
<th>$I_{OL} (\text{mA})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>8051 Port 0</td>
<td>-7.0</td>
<td>7.0</td>
</tr>
<tr>
<td>8051 Port 1,2,3</td>
<td>-0.3</td>
<td>3.5</td>
</tr>
<tr>
<td>MAX232/233</td>
<td>-1.0</td>
<td>3.2</td>
</tr>
<tr>
<td>HS-82C55ARH</td>
<td>-2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>UT28F64</td>
<td>-0.4</td>
<td>4.0</td>
</tr>
<tr>
<td>UT67164</td>
<td>-0.4</td>
<td>4.0</td>
</tr>
<tr>
<td>UT54ACS08</td>
<td>-8.0</td>
<td>8.0</td>
</tr>
<tr>
<td>UT54ACS10</td>
<td>-8.0</td>
<td>8.0</td>
</tr>
<tr>
<td>UT54ACS138</td>
<td>-8.0</td>
<td>8.0</td>
</tr>
<tr>
<td>UT54ACS139</td>
<td>-8.0</td>
<td>8.0</td>
</tr>
<tr>
<td>UT54ACS245</td>
<td>-12.0</td>
<td>12.0</td>
</tr>
<tr>
<td>HC573MS</td>
<td>-7.2</td>
<td>7.2</td>
</tr>
<tr>
<td>HC574MS</td>
<td>-7.2</td>
<td>7.2</td>
</tr>
</tbody>
</table>

Table 1: Summary Of Component Pin Currents

E. DISCRETE COMPONENTS

1. Crystal Oscillator

A crystal oscillator with a frequency of 7.3728 MHz was used to provide timing input to the microcontroller. This seemingly odd frequency aids in ensuring the baud rates on the serial port fall within standard ranges such as 4800, 9600, etc. The microcontroller will run on any frequency from 2 - 20 MHz.

2. Capacitors

   a. 10 $\mu F$

   A 10 $\mu F$ capacitor is used in conjunction with a resistor to generate a time delay for the reset line of 8051 and 8255 on power-up. This value is recommended in the data sheets.

   b. 1$\mu F$

   Four, 1 $\mu F$ capacitors are used with the MAX232 to provide ±10 V DC using the charge pump circuitry built into the device. These will not be required on the production board as the MAX233 will be used, which does not require external capacitors.
c. 1μF

These capacitors are used for decoupling, that is providing a momentary source of current when ICs switch, providing voltage stability.

3. Resistors

A 8.2KΩ resistor is used in conjunction with a capacitor to provide a time delay for the reset line of both the 8051 and the 8255 on power-up. This ensures the board starts in a known state, and is recommended on the data sheets.

F. CONNECTORS

1. 40 Pin Header

Four, 40-pin headers are used to provide an interface with the device to be tested. The pins are set up in a 2 X 20 grid with 100 mil spacing.

2. 25-pin sub-D

A 25-pin, sub-D, male connector is used to interface the board with a standard serial RS-232 port of a laptop computer.
IV. DESIGN ANALYSIS

A. OVERVIEW

A detailed description of the universal test platform's design and operation is provided in this section. The goal is to provide a stand-alone document that fully explains the wiring and operation of each section of the design.

B. SERIAL INTERFACE

The first task was to create an interface with the outside world using a RS-232 serial interface. The actual serial connection itself was rather easy in that this basic function was built into the microcontroller. The key part of this interface was to convert the TTL-level serial signals to the RS-232 levels. This could have been accomplished using one of the two major families of RS-232 drivers. The first is the 1488/1489 line driver/receiver combination. Both 148X ICs require ±12 V, which is not required elsewhere in this design. This choice also requires two IC packages, whereas the MAX232/233 family requires only one.

As this is a relatively simple serial connection, with only a transmit line (TXD) and a receive line (RXD), only one line driver and receiver are required. Maxim makes a family of line drivers and receivers on the same chip with the added benefit of requiring only a +5 V supply, instead of the ±12 V that is typically required of line drivers. This family of chips uses a technique called dual charge-pump voltage conversion, which uses capacitors to generate the ±10 V required for the RS-232 line drivers. The two devices used in this research are the MAX232, which use external capacitors, and the MAX233, which has internal capacitors and therefore requires less board space. The MAX232 was used on the wire wrapped board, primarily because of its availability early in the design process. For the production board, the MAX233 is used.

1. MAX232

The MAX232 requires four external 1 µF capacitors, as well as power and ground for support circuitry. The functional portion of the MAX232 contains two line drivers and two line receivers, of which one each is used. The wiring diagram for the MAX232 is shown in Figure 3.
2. MAX233

The MAX233 is an improvement upon the MAX232 in that it requires only +5 V and ground, with no external capacitor requirement. It also contains two line drivers and two line receivers and is wired as shown in Figure 4.

Figure 3: MAX232

Figure 4: MAX233
3. Serial connector

The output of the line driver and the input of the line receiver are wired to a standard, male, 25-pin, sub-D connector, as shown below in Figure 5. The line driver output is connected to pin 3 of the connector (received data for DTE in the RS-232 standard), and the line receiver is connected to pin 2 of the connector (transmitted data). Pin 7 (signal ground) is connected to ground.

![Diagram of RS-232 Connector](image)

**Figure 5: RS-232 Connector (minimal)**

4. Overall Serial Subsystem

The input of the line driver is connected to the serial transmit (TXD) line of the microcontroller and the line receiver output is connected to the serial receive (RXD) line of the microcontroller. The MAX232X receive data in (Rin) and transmit data out (Tout) are wired in accordance with the RS-232 standard, as shown above. The complete communications subsystem is shown in Figure 6.
C. TIMING GENERATION

The timing generation for this system can use one of two methods, depending upon what the desired timing source is. The two choices are to use a crystal or a R/C oscillator. The difference is accuracy versus complexity and both have been used in developing the prototype board. The R/C oscillator was dismissed because it is not stable enough for serial-line baud-rate generation. When the R/C oscillator is used, the output is wired to pin 19 (XTAL1) of the microcontroller.

When a crystal is used, two additional capacitors are required, providing parallel resonance for the fundamental response mode of the crystal. When this configuration is used, the crystal is wired to pins 18 and 19 (XTAL2 and XTAL1) of the microcontroller and both pins are
grounded through capacitors, as shown in Figure 7. This method was chosen because of its simplicity and the relative tolerance crystals and capacitors show to radiation effects.

![Figure 7: Timing Subsystem](image)

D. MEMORY INTERFACE

1. Overview

The memory subsystem is also relatively straightforward, the most difficult part of which is latching the low order address-byte so it is available while the data-byte is on the output port. The memory space is divided into eight, 8K blocks within the 64K memory space. The basic memory layout is shown in Table 2.
<table>
<thead>
<tr>
<th>MEMORY LOCATIONS</th>
<th>USE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000H-1FFFH</td>
<td>ROM SPACE</td>
</tr>
<tr>
<td>2000H-3FFFH</td>
<td>RAM 1 SPACE</td>
</tr>
<tr>
<td>4000-5FFFH</td>
<td>RAM 2 SPACE (PRODUCTION BOARD)</td>
</tr>
<tr>
<td>6000-7FFFH</td>
<td>BOARD CONTROL SPACE</td>
</tr>
<tr>
<td>8000-9FFFH</td>
<td>TEST SPACE 1</td>
</tr>
<tr>
<td>A000-BFFFH</td>
<td>TEST SPACE 2</td>
</tr>
<tr>
<td>C000-DFFFH</td>
<td>TEST SPACE 3</td>
</tr>
<tr>
<td>E000-FFFFH</td>
<td>TEST SPACE 4</td>
</tr>
</tbody>
</table>

Table 2: Memory Map

2. **XX573**

Port 0 of the 8051 is multiplexed and used for both the low-order address byte and the data byte to be written or read. Therefore, the address byte must be stored and held for use during the memory access cycle. This is accomplished easily with either a XX373 or XX573 latch. The XX573 was chosen for its broadside configuration, where all input pins are on one side of the package and all output pins are on the opposite side of the package. This makes it easier for both wire wrapping and printed circuit board production, where it is desirable to keep trace and wire lengths to a minimum.

The support pins on this chip are straightforward: power, ground, output enable (OE*), and latch enable (LE*). The output enable was wired low, allowing the address to always be available after latching. The latch enable line is wired to the address latch enable (ALE*) line of the microcontroller, an obvious choice.

3. **XX138**

The XX138 is used to partition the 64K address space into eight, 8K blocks. The support pins are: power, ground, and 3 enable lines. In this design, all enable lines are asserted, allowing the selected partition to be active as long as the upper three address bits indicate that partition.
4. XX08

The XX08 is used to combine the program store enable (PSEN*) and read (RD*) lines. This is a recommendation in many application notes [4,5] and allows the RAM to be used as both program and data memory. This facilitates the downloading of code into the RAM, which can then be run as a program. This eliminates the need to program a PROM every time a change is made in a program and allows for easy code modification with a simple monitor program. The complete memory interface is shown in Figure 8.
Figure 8: Memory Interface
E. MEMORY

1. PROM (UT28F64)

The PROM interface is very straightforward. The data lines are wired directly to port 0 of the microcontroller. The low-order byte of the address is latched by the XX573, therefore address lines 0-7 are wired to the latch. The 5 high-order address bits are wired to the microcontroller address lines on port 2. The remaining pins to be interfaced are the chip select (CS*), read (RD*), program (PGM), and program voltage (Vpp) lines. The PROM is designed to occupy the first 8K partition of the memory space and is therefore wired to the Y0 output of the XX138. The read line could be wired to either the read (RD*) line of the microcontroller or to the output of the XX08. Here, the output of the XX08 was chosen to minimize the lines wired directly to the microcontroller. The program and program voltage lines are wired high, as this design does not provide for programming the PROM. The PROM interface is shown in Figure 9.

2. SRAM (UT67164)

The SRAM interface is very similar to that of the PROM. The four support pins are the read (RD*), write (WR*), chip select 1 (CS1*), and chip select2 (CS2) lines. The write line is wired directly to the microcontroller and the read line is wired to the XX08 to allow the SRAM to act as both program and data memory space. The chip select 1 line is wired to the Y1 output to configure the SRAM as the second 8K partition. The chip select 2 line is not needed and is, therefore, wired high to allow only the chip select 1 line to activate the SRAM. The address and data lines are wired the same as for the PROM. Only one 8K SRAM is implemented on the prototype board. The basic interface is shown in Figure 9.
F. PERIPHERAL INTERFACE

This section provides for additional I/O to interface with the device to be tested. Table 3 shows the memory map of the peripheral interface control space.

<table>
<thead>
<tr>
<th>MEMORY LOCATION</th>
<th>USE</th>
</tr>
</thead>
<tbody>
<tr>
<td>6000-6003H</td>
<td>8255 #1</td>
</tr>
<tr>
<td>6007-6007H</td>
<td>8255 #2</td>
</tr>
<tr>
<td>6008-600BH</td>
<td>8255 #3</td>
</tr>
<tr>
<td>600C-600FH</td>
<td>8255 #4</td>
</tr>
<tr>
<td>6010H</td>
<td>LATCHED BYTE 0</td>
</tr>
<tr>
<td>6014H</td>
<td>LATCHED BYTE 1</td>
</tr>
<tr>
<td>6018H</td>
<td>LATCHED BYTE 2</td>
</tr>
<tr>
<td>601CH</td>
<td>LATCHED BYTE 3</td>
</tr>
</tbody>
</table>

Table 3: Peripheral Memory Map
1. **XX138**

The XX138 is used to further subdivide the lower 32 bits of memory partition 3 (6000H-7FFFH) into 4-bit units. This is done to keep all the control board control functions within partition 3, allowing a 32K memory space for the device under test. The control inputs to this device are three address lines and three enable lines. The address lines are wired to address bits 2 to 4 of the XX573 latch, creating eight, 4-bit blocks. These blocks repeat every 32 addresses within memory partition 3 (6000H-7FFFH). The additional circuitry required to fully decode this address space is not implemented in this design.

2. **8255**

The 8255 interface methodology is similar to that of a simple RAM. There is an 8-bit data bus and a 2-bit address bus along with read (RD*), write (WR*), chip select (CS*), and reset (RESET) lines. The data byte is wired to the microcontroller data bus, while the two address lines are wired to the lowest 2 bits on the memory-interface XX573 latch. The read, write and reset lines are all wired directly to their counterparts on the microcontroller. The remaining line to be interfaced is the chip select, which is controlled by the output of the XX138 that selects which 8255 is to be addressed. The first four outputs of the XX138 (Y0-Y3) select one of the four 8255s on the production board, or one of the two 8522s on the wire-wrapped board. The 8255s I/O interface has a total of 24 bits, which is arranged as three 8-bit ports. The address pins(A0,A1) select between ports A, B, C, and the control word. The three I/O ports are wired to the 40-pin connectors, allowing for control of the device to be tested. The 8255 interface is show in Figure 10.
Figure 10: 8255 Interface
G. LATCHING / READING 32 BITS

To allow this device to test other ICs with data buses wider than 8 bits, a technique had to be developed to allow an eight-bit microcontroller to read multiple bytes. The method used here latches up to 32 bits of data, then reads the values a byte at a time from the control space.

1. XX574

The purpose of the XX574 used here is two fold, it allows for latching the data as well as acting as a tri-state buffer. The XX574 flip-flop was chosen because storage is desired on transitions, not simply on levels, as with the XX573 latch. The control lines of the XX574 consist of an output enable (OE*) and a clock input (CLK). When the CLK signal transitions from low to high, the data on the inputs is stored in the flip flops. The logic to control the clock line is discussed below. When a control space read is initiated, the output enable is activated and the stored data appears on the data bus.

2. XX08

The XX08 provides the input clock to the flip-flops when a specific set of circumstances is present, mainly when A15 is high and the read line transitions from low to high.

![Diagram of Byte Latch Circuitry]

Figure 11: Byte Latch Circuitry

25
3. Word latch

To read 32 bits, all data bits of the device under test are stored at one time and are then read a byte at a time as part of the control space. A read from the test data space (upper 32K) will cause 32 bits of data to be clocked into the XX574s (16 bits on the prototype board). These bytes can then be read as part of the control space at addresses 6010H, 6014H, 6018H, and 601CH, as determined by the XX138 described in the peripheral interface section.

H. RESET

The reset circuitry is relatively straightforward, a switch with a capacitor and resistor to provide an appropriate delay, as specified in the applications notes. A diagram of this circuit is shown in Figure 12.

![Reset Circuit Diagram]

**Figure 12: Reset Circuit**
I. EXTERNAL CONNECTORS

1. RS-232

The connection to the outside world is through a 25-pin, sub-D, male connector, which is compatible with most standard computer serial interfaces. There are only three pins used, thus the protocol is fairly simple. Pin 7 is used for signal ground and pins 2 and 3 are used for transmit data and receive data, respectively.

2. Test Interface

The external connectors are 40-pin dip headers which are broken down by functionality. Connector 1 is the microcontroller bus interface. Connectors 2 and 3 each contain 16 bits of programmable I/O and a 16-bit latch capability, as well as 4 control bits. Connector 4 provides 32 bits of programmable I/O and some additional bit control lines.

a. connector 1

Connector 1 contains all the necessary lines to connect a device with an address space of up to 32Kb and an 8-bit data word. Also provided are 8 bits that can be programmed as an input or output byte. The specific signals include 16 address bits, 8 data bits, 8 I/O bits, and 4 chip selects, along with read, write, power and ground. Table 4 shows the specific pin arrangements. This connector is fully implemented on the prototype board. All signals use TTL levels and will drive TTL loads.

b. connector 2

Connector 2 contains two bytes that may be configured as either input or output, as well as two bytes capable of being latched by the control board. Also provided is a group of 4 bits that can be individually set and cleared, or serve as a 4-bit I/O port. The details are shown in Table 5 and are fully implemented on the prototype board.

c. connector 3

Connector 3 is designed to be implemented on the production board and is not on the prototype board. These signals are similar to what is provided on connector 2 and makes it possible to test a 32-bit memory device. The details are shown in Table 6.
<table>
<thead>
<tr>
<th>PIN</th>
<th>FUNCTION</th>
<th>PIN</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DATA 0</td>
<td>21</td>
<td>ADDRESS 0</td>
</tr>
<tr>
<td>2</td>
<td>DATA 1</td>
<td>22</td>
<td>ADDRESS 1</td>
</tr>
<tr>
<td>3</td>
<td>DATA 2</td>
<td>23</td>
<td>ADDRESS 2</td>
</tr>
<tr>
<td>4</td>
<td>DATA 3</td>
<td>24</td>
<td>ADDRESS 3</td>
</tr>
<tr>
<td>5</td>
<td>DATA 4</td>
<td>25</td>
<td>ADDRESS 4</td>
</tr>
<tr>
<td>6</td>
<td>DATA 5</td>
<td>26</td>
<td>ADDRESS 5</td>
</tr>
<tr>
<td>7</td>
<td>DATA 6</td>
<td>27</td>
<td>ADDRESS 6</td>
</tr>
<tr>
<td>8</td>
<td>DATA 7</td>
<td>28</td>
<td>ADDRESS 7</td>
</tr>
<tr>
<td>9</td>
<td>POWER (+5V DC)</td>
<td>29</td>
<td>GROUND</td>
</tr>
<tr>
<td>10</td>
<td>I/O 0 (FROM 1)</td>
<td>30</td>
<td>ADDRESS 8</td>
</tr>
<tr>
<td>11</td>
<td>I/O 1</td>
<td>31</td>
<td>ADDRESS 9</td>
</tr>
<tr>
<td>12</td>
<td>I/O 2</td>
<td>32</td>
<td>ADDRESS 10</td>
</tr>
<tr>
<td>13</td>
<td>I/O 3</td>
<td>33</td>
<td>ADDRESS 11</td>
</tr>
<tr>
<td>14</td>
<td>I/O 4</td>
<td>34</td>
<td>ADDRESS 12</td>
</tr>
<tr>
<td>15</td>
<td>I/O 5</td>
<td>35</td>
<td>ADDRESS 13</td>
</tr>
<tr>
<td>16</td>
<td>I/O 6</td>
<td>36</td>
<td>ADDRESS 14</td>
</tr>
<tr>
<td>17</td>
<td>I/O 7</td>
<td>37</td>
<td>ADDRESS 15</td>
</tr>
<tr>
<td>18</td>
<td>CS A000-FFFFh</td>
<td>38</td>
<td>READ*</td>
</tr>
<tr>
<td>19</td>
<td>CS C000-DFFFh</td>
<td>39</td>
<td>WRITE*</td>
</tr>
<tr>
<td>20</td>
<td>CS E000-FFFFh</td>
<td>40</td>
<td>CS 8000-9FFFh</td>
</tr>
</tbody>
</table>

**Table 4: Connector 1**

<table>
<thead>
<tr>
<th>PIN</th>
<th>FUNCTION</th>
<th>PIN</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8255 #1 PORT</td>
<td>21</td>
<td>T/S DATA IN 0</td>
</tr>
<tr>
<td>2</td>
<td>I/O 0</td>
<td>22</td>
<td>T/S DATA IN 1</td>
</tr>
<tr>
<td>3</td>
<td>I/O 1</td>
<td>23</td>
<td>T/S DATA IN 2</td>
</tr>
<tr>
<td>4</td>
<td>I/O 2</td>
<td>24</td>
<td>T/S DATA IN 3</td>
</tr>
<tr>
<td>5</td>
<td>I/O 3</td>
<td>25</td>
<td>T/S DATA IN 4</td>
</tr>
<tr>
<td>6</td>
<td>I/O 4</td>
<td>26</td>
<td>T/S DATA IN 5</td>
</tr>
<tr>
<td>7</td>
<td>I/O 5</td>
<td>27</td>
<td>T/S DATA IN 6</td>
</tr>
<tr>
<td>8</td>
<td>I/O 6</td>
<td>28</td>
<td>T/S DATA IN 7</td>
</tr>
<tr>
<td>9</td>
<td>8255 #1 PORT</td>
<td>29</td>
<td>T/S DATA IN 8</td>
</tr>
<tr>
<td>10</td>
<td>I/O 1</td>
<td>30</td>
<td>T/S DATA IN 9</td>
</tr>
<tr>
<td>11</td>
<td>I/O 2</td>
<td>31</td>
<td>T/S DATA IN 10</td>
</tr>
<tr>
<td>12</td>
<td>I/O 3</td>
<td>32</td>
<td>T/S DATA IN 11</td>
</tr>
<tr>
<td>13</td>
<td>I/O 4</td>
<td>33</td>
<td>T/S DATA IN 12</td>
</tr>
<tr>
<td>14</td>
<td>I/O 5</td>
<td>34</td>
<td>T/S DATA IN 13</td>
</tr>
<tr>
<td>15</td>
<td>I/O 6</td>
<td>35</td>
<td>T/S DATA IN 14</td>
</tr>
<tr>
<td>16</td>
<td>I/O 7</td>
<td>36</td>
<td>T/S DATA IN 15</td>
</tr>
<tr>
<td>17</td>
<td>8255 #1 PORT</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>I/O 4</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>I/O 5</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>I/O 7</td>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>

**Table 5: Connector 2**
d. connector 4

Connector 4 is designed to be implemented on the production board and is not on the prototype board. These signals provide additional I/O lines that may be used to drive a 32-bit address bus on larger devices being tested. Details are shown in Table 7.

J. TIMING ANALYSIS

The overall timing of components for this design is straightforward, as all timing is synchronous. Analyzing the memory timing constraints of the 8051 shows that time from a low to data valid must be within the limits of Equation 4.1.

\[ t_{RLDV} = 5t_{CCL} - 165\text{ns} \]  \hspace{1cm} \text{Equation 4.1}

Here \( t_{CCL} \) is the clock period low transition to low transition. Operating the 8051 at a maximum clock rate of 20MHz, \( t_{CCL} \) is 50ns and \( t_{RLDV} \) is 85 ns. For the read and write pulse widths, Equations 4.2 and 4.3 apply.

\[ t_{RLRH} = 6t_{CCL} - 100\text{ns} \]  \hspace{1cm} \text{Equation 4.2}

\[ t_{WLWH} = 6t_{CCL} - 100\text{ns} \]  \hspace{1cm} \text{Equation 4.3}

The pulse widths for the read and write signals are 200 ns when the 8051 is operating at 20MHz. When operation is slowed down to 7.3728, the \( t_{CCL} \) drops to 136 ns, yielding a \( t_{RLDV} \) of 513ns and a read/write pulse width of 716 ns. The RAD hard memory components have access times of 35 ns or less, allowing operation of this design at the 8051 maximum data rate.

The 8255 requires a read pulse width of 250 ns and has a 200 ns maximum requirement for data valid from read asserted. On writes, the 8255 requires only 100 ns, which falls well within the maximum speed of this design. Taking these 8255 read and write constraints into consideration, if only output functions are being utilized, a 20MHz clock may be used. However, if input from the 8255 is desired, the speed must be slowed down to 17 MHz.
<table>
<thead>
<tr>
<th>PIN</th>
<th>FUNCTION</th>
<th>PIN</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8255 #2 PORT A I/O 0</td>
<td>21</td>
<td>T/S DATA IN 17</td>
</tr>
<tr>
<td>2</td>
<td>I/O 1</td>
<td>22</td>
<td>T/S DATA IN 18</td>
</tr>
<tr>
<td>3</td>
<td>I/O 2</td>
<td>23</td>
<td>T/S DATA IN 19</td>
</tr>
<tr>
<td>4</td>
<td>I/O 3</td>
<td>24</td>
<td>T/S DATA IN 20</td>
</tr>
<tr>
<td>5</td>
<td>I/O 4</td>
<td>25</td>
<td>T/S DATA IN 21</td>
</tr>
<tr>
<td>6</td>
<td>I/O 5</td>
<td>26</td>
<td>T/S DATA IN 22</td>
</tr>
<tr>
<td>7</td>
<td>I/O 6</td>
<td>27</td>
<td>T/S DATA IN 23</td>
</tr>
<tr>
<td>8</td>
<td>I/O 7</td>
<td>28</td>
<td>T/S DATA IN 24</td>
</tr>
<tr>
<td>9</td>
<td>8255 #2 PORT B I/O 0</td>
<td>29</td>
<td>T/S DATA IN 25</td>
</tr>
<tr>
<td>10</td>
<td>I/O 1</td>
<td>30</td>
<td>T/S DATA IN 26</td>
</tr>
<tr>
<td>11</td>
<td>I/O 2</td>
<td>31</td>
<td>T/S DATA IN 27</td>
</tr>
<tr>
<td>12</td>
<td>I/O 3</td>
<td>32</td>
<td>T/S DATA IN 28</td>
</tr>
<tr>
<td>13</td>
<td>I/O 4</td>
<td>33</td>
<td>T/S DATA IN 29</td>
</tr>
<tr>
<td>14</td>
<td>I/O 5</td>
<td>34</td>
<td>T/S DATA IN 30</td>
</tr>
<tr>
<td>15</td>
<td>I/O 6</td>
<td>35</td>
<td>T/S DATA IN 31</td>
</tr>
<tr>
<td>16</td>
<td>I/O 7</td>
<td>36</td>
<td>T/S DATA IN 32</td>
</tr>
<tr>
<td>17</td>
<td>8255 #2 PORT C I/O 4</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>I/O 5</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>I/O 6</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>I/O 7</td>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Connector 3

<table>
<thead>
<tr>
<th>PIN</th>
<th>FUNCTION</th>
<th>PIN</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8255 #3 PORT A I/O 0</td>
<td>21</td>
<td>8255 #4 PORT A I/O 0</td>
</tr>
<tr>
<td>2</td>
<td>I/O 1</td>
<td>22</td>
<td>I/O 1</td>
</tr>
<tr>
<td>3</td>
<td>I/O 2</td>
<td>23</td>
<td>I/O 2</td>
</tr>
<tr>
<td>4</td>
<td>I/O 3</td>
<td>24</td>
<td>I/O 3</td>
</tr>
<tr>
<td>5</td>
<td>I/O 4</td>
<td>25</td>
<td>I/O 4</td>
</tr>
<tr>
<td>6</td>
<td>I/O 5</td>
<td>26</td>
<td>I/O 5</td>
</tr>
<tr>
<td>7</td>
<td>I/O 6</td>
<td>27</td>
<td>I/O 6</td>
</tr>
<tr>
<td>8</td>
<td>I/O 7</td>
<td>28</td>
<td>I/O 7</td>
</tr>
<tr>
<td>9</td>
<td>8255 #3 PORT B I/O 0</td>
<td>29</td>
<td>8255 #4 PORT B I/O 0</td>
</tr>
<tr>
<td>10</td>
<td>I/O 1</td>
<td>30</td>
<td>I/O 1</td>
</tr>
<tr>
<td>11</td>
<td>I/O 2</td>
<td>31</td>
<td>I/O 2</td>
</tr>
<tr>
<td>12</td>
<td>I/O 3</td>
<td>32</td>
<td>I/O 3</td>
</tr>
<tr>
<td>13</td>
<td>I/O 4</td>
<td>33</td>
<td>I/O 4</td>
</tr>
<tr>
<td>14</td>
<td>I/O 5</td>
<td>34</td>
<td>I/O 5</td>
</tr>
<tr>
<td>15</td>
<td>I/O 6</td>
<td>35</td>
<td>I/O 6</td>
</tr>
<tr>
<td>16</td>
<td>I/O 7</td>
<td>36</td>
<td>I/O 7</td>
</tr>
<tr>
<td>17</td>
<td>8255 #3 PORT C I/O 4</td>
<td>37</td>
<td>8255 #4 PORT C I/O 4</td>
</tr>
<tr>
<td>18</td>
<td>I/O 5</td>
<td>38</td>
<td>I/O 5</td>
</tr>
<tr>
<td>19</td>
<td>I/O 6</td>
<td>39</td>
<td>I/O 6</td>
</tr>
<tr>
<td>20</td>
<td>I/O 7</td>
<td>40</td>
<td>I/O 7</td>
</tr>
</tbody>
</table>

Table 7: Connector 4
K. COMPLETE DESIGN DIAGRAM

Figure 13 shows the complete layout of this board design. Decoupling capacitors are omitted, as well as the specific connector layout, which is available elsewhere in this document.
V. PROGRAMMING

A. OVERVIEW

In order to use the test hardware to test IC devices, a test program must be written, the board configured, and a test fixture created. This chapter will deal with all of these requirements, as well as describing some of the specific tests that the hardware will run.

B. BOARD CONFIGURATION

This process consists of determining which parts of the board are to be utilized and then writing a configuration routine to set up the appropriate parameters. This initialization may be run separately or incorporated as part of the actual test program. The simplest initialization program is actually no program at all. When the board is initialized, the 8255 ports are configured as inputs and the 8051 bus is fully accessible. Any simple memory device with 8 data bits and up to 32K of address space can be connected directly to the microcontroller bus (available on connector 1) with no additional setup requirements.

The 8255 will operate in one of three basic modes, byte input, byte output, and bit-set mode. As mentioned before, for byte input mode, the data simply needs to be read from the appropriate address, as shown below in Table 8.

<table>
<thead>
<tr>
<th>MEMORY LOCATION</th>
<th>USE</th>
<th>MEMORY LOCATION</th>
<th>USE</th>
</tr>
</thead>
<tbody>
<tr>
<td>6000H</td>
<td>PORT A 8255 #1</td>
<td>600AH</td>
<td>PORT C 8255 #3</td>
</tr>
<tr>
<td>6001H</td>
<td>PORT B 8255 #1</td>
<td>600BH</td>
<td>8255 #3 CONTROL WORD</td>
</tr>
<tr>
<td>6002H</td>
<td>PORT C 8255 #1</td>
<td>600CH</td>
<td>PORT A 8255 #4</td>
</tr>
<tr>
<td>6003H</td>
<td>8255 #1 CONTROL WORD</td>
<td>600DH</td>
<td>PORT B 8255 #4</td>
</tr>
<tr>
<td>6004H</td>
<td>PORT A 8255 #2</td>
<td>600EH</td>
<td>PORT C 8255 #4</td>
</tr>
<tr>
<td>6005H</td>
<td>PORT B 8255 #2</td>
<td>600FH</td>
<td>8255 #4 CONTROL WORD</td>
</tr>
<tr>
<td>6006H</td>
<td>PORT C 8255 #2</td>
<td>6010H</td>
<td>LATCHED BYTE 0</td>
</tr>
<tr>
<td>6007H</td>
<td>8255 #2 CONTROL WORD</td>
<td>6014H</td>
<td>LATCHED BYTE 1</td>
</tr>
<tr>
<td>6008H</td>
<td>PORT A 8255 #3</td>
<td>6018H</td>
<td>LATCHED BYTE 2</td>
</tr>
<tr>
<td>6009H</td>
<td>PORT B 8255 #3</td>
<td>601CH</td>
<td>LATCHED BYTE 3</td>
</tr>
</tbody>
</table>

Table 8: Memory Map
To set up the I/O mode, the appropriate byte must be written to the 8255 control word. The most common I/O type, mode 0, performs basic byte input and output. The layout of the control word is shown in Figure 14. To set up port A as output, bit 7 must be set, bits 5 and 6 need to be 00 for Mode 0, and bit 4 needs to be 0 for output. Hence, to set up the #1 8255 port A for output mode, write 1000xxxxxB (here x is don’t care, and B is binary) to address 6003H. Then, it is simply a matter of writing the data byte that is to be put on port A to address 6000H. Port C is broken into an upper half and a lower half and each half can be independently set for input or output. To set up the lower half of port C for output, control bit 7 must be set, bit 2 should be cleared, and bit 0 should be cleared. Hence, a control word of 1xxxx0x0B should be written to the control register. Of course, it should be noted that if ports A and C are to both be used, the control words must be combined yielding a control word of 1000x0x0B.

Figure 14: 8255 Control Word
C. ENABLING TRANCEIVERS

Once the modes of the 8255 ports are configured, the bus transceivers must be enabled and configured for the proper directions. Table 9 shows which bits of the upper C port control which transceiver functions.

<table>
<thead>
<tr>
<th>Port C bit</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 7</td>
<td>Port A direction (0 = input, 1 = output)</td>
</tr>
<tr>
<td>bit 6</td>
<td>Port A enable (0 = enabled)</td>
</tr>
<tr>
<td>bit 5</td>
<td>Port B direction (0 = input, 1=output)</td>
</tr>
<tr>
<td>bit 4</td>
<td>Port B enable (0 = enabled)</td>
</tr>
</tbody>
</table>

Table 9: Transceiver Enables

To set up Port A as output, bit 7 must be set and bit 6 must be cleared, yielding a control word of 10xxxxxx. To set up Port B as input, bits 4 and 5 must be cleared. The combined control word for output on Port A and input on Port B is 1000xxxxB. It should be noted the lower half of Port C is available as a 4-bit input or output port.

D. PROGRAMMING

In order to actually perform tests, a program must be written. This program is written in 8051 assembly language and must be debugged, compiled, and loaded as with any assembly language program. The assembler used for this work is a shareware version called A51. Once the program is written, it must be loaded onto the test board by either RAM download or PROM program. The actual ‘operating system’ of this board is a monitor program called PAULMON, which was developed at Oregon State University. On top of this monitor is a front-end that will run any of the test routines written as part of this research, as well as provide instruction on how to download additional programs. The monitor occupies address space from 0000H-0FFFFH in its basic form and occupies through 1FFFH if disassembler and debugger options are used. The monitor provides functions for serial I/O as well as memory editing, program downloading, and memory dumping. A listing of built-in functions is shown in Appendix A.

The easiest way to load a program is to download it via the RS-232 serial port and then run it from the monitor. For this method, address 2000H starts the RAM area. A complete description of the process is detailed below in the Vitesse section.
E. TEST DEVICE INTERFACE

Once the board is set up and the program is functioning, the device to be tested must be interfaced with the board. For each device to be tested, a separate test fixture must be built to interface with the test hardware connectors. For most applications, two, 40-conductor ribbon cables will carry the signals to the test fixture of the device under tested. Typically, only a socket and two 40-pin headers are required. This method was chosen for maximum flexibility, as there are so many package types and many different power configurations per package.

F. VITESSE

The setup for the Vitesse SRAM is very simple. It must be connected to the microcontroller bus, therefore requiring only connector 1. The layout of the IC is 8 address lines, 4 data in lines, 4 data out lines, and the usual read, write, and chip-select lines. Only the active-low enable line is used, therefore the active-high chip select line is tied high. The address lines are wired to the address 0-7 lines on connector 1, along with the read, write, and one of the chip-select lines (the 8000H line was used here). The only remaining lines are the data lines, of which it was decided to wire the inputs to data 0-3 and the outputs to data 4-7.

The program is relatively straight forward, simply write a fixed value to addresses 8000H-80FFH and then read the values back. The only small glitch in this procedure is that the data read must be ANDed with F0H before a comparison is made. The assembly language follows:

; Written by Duane Amsler as part of masters thesis

; the following program writes a value of 55H to memory locations 8000H-80FFH and then
; reads back the data in those locations. The read data is then masked with F0H to disregard
; the lower 4 bits, as this program is designed to test a 4-bit memory device

; initial assignments to allow output of memory locations and ASCII data

.equ  Cout, 0x0030 ;Send Acc to serial port

.equ  pHex16, 0x0036 ;Print Hex value of DPTR

36
.ORG H'2000

; start this program within the RAM space at 2000H

mov dptr,#h'3000
; set up test address to start at 8000H
mov a,#h'55
; use a value of 55H (01010101B) as a test pattern
loopd1: movx @dptr,a
; move accumulator to memory at data pointer (dptr)
inc dpt
; increment the data pointer
mov r7,dpl
; move the lower half of the data pointer to register 7
cjne r7,#0,loopd1
; if the data pointer lower byte is not 0, continue
ret
; return

.ORGH'2100

; start this program within the RAM space at 2100H

read: mov dptr,#h'8000
; start reading data from address 8000H
loopr1: movx a,@dptr
; load the data at data pointer to the accumulator
anl a,#h'F0
; mask out the lower 4 bits
cjne a,#h'05,error
; if the value is x5, continue, else jump to error
inc dpt
; increment the data pointer
mov r7,dpl
; move the lower data pointer byte to register 7
cjne r7,#0,loopr1
; if lower data pointer is 00 exit, else continue
ret
; return to the calling program

error: lcall cout
; if error, call cout (will print the accumulator value,
; which will not be x5H if error was called)
mov a,'#'
; load accumulator with ASCII value for space
lcall cout
; output the space in the accumulator
lcall phex16
; output the value of the data pointer (the location where
; the error occurred
mov a,#h'55
; move 55 into the accumulator
movx @dptr,a
; store 55 in the location where the error occurred
ajmp loopr1
; jump back to the check loop

The above program was tested in RAM at location 3000H. The first routine was run from
the monitor and then values written were verified by a memory dump. After the correct data was
verified, a memory location was edited to a different value and the second routine was run. The
value was correctly detected, corrected, and output to the laptop via the serial port.
VI. CONCLUSIONS

A. DESIGN CONCLUSIONS

The overall design of the test hardware was fairly straightforward, although designing to test up to a 32-bit device with an 8-bit microcontroller was rather challenging. The basic memory layout is widely used in applications notes and required some modification. The latching and shifting circuitry is original and may not be the most optimal solution, which will allow for future scrutiny and possible optimization of this work. It may be possible to program the 8255s to latch and hold. However, this will require additional configuration lines, which may be implemented as DIP switches, outputs from the 8051, or 8255 outputs.

B. BOARD FABRICATION

The board design is laid out in the CADENCE design tool CONCEPT, and can be converted into a GERBER file for fabrication. This will provide a reliable board that is much more elegant than the prototype board that was wire-wrapped.

C. COMPONENT COST

The RAD hard parts are very expensive and will require a fair amount of funding to purchase. The United Technologies VLSI parts range from $1500-$1900, while their commercial counterparts range from $3-$20. The United Technologies MSI RAD hard logic devices are all $166, as opposed to the commercial MSI component costs of less than a dollar.

D. PROGRAMMING & TESTING

The initial attempt to test a Vitesse SRAM caused the microcontroller bus to fail. This may be the result of heavy loading of the bus by the part, or a faulty part. The next test will be to isolate the SRAM from the microcontroller bus and perform the tests under manual control of the 8255s. This will require the control lines be simulated with the bit set and clear capabilities of port C of the 8255. This work is ongoing.

E. FOLLOW-ON WORK

There are a few areas that may allow significant follow-on work. The most obvious is actually testing the LT GaAs experimental chips, as well as this board, in a radiation environment.
1. **Produce Professional PC Board**

To improve upon system reliability and robustness, a commercially-produced PC board should be considered to both increase capabilities to 32 bits, as well as provide an aesthetically pleasing package for transport to radiation test sites.

2. **Use RAD Hard components**

The intent of this design was to create a unit capable of testing ICs in a radiation environment. In its current configuration (with commercial parts), lead bricks must be set up around this board to shield it from radiation. This method will work, but is far from the optimal solution.

3. **ECC on Serial Link**

The serial link currently uses a MAX233, which has no RAD hard counterparts. The 1488/1489 family also has no RAD hard counterpart. Because of this, some sort of error correcting code should be used on the RS-232 serial link. This will ensure proper data is obtained on tests. Something simple, such as a parity bit, possibly with check summing and retransmission, could be used. Further research is warranted.

4. **Use of a C Compiler**

To aid in the development of test programs, a high-level language could be used. Although assembly language is still useful, the use of C will aid in increasing program complexity and program development speed.

5. **Creating a Script Language**

Instead of another high-level language such as C, a scripting language could be developed to create an intuitive method for programming tests.

**F. SUMMARY**

This overall design was a valuable learning tool and should provide some functionality and utility in the future. This research should serve as a good starting point for improved designs, as well as an example of good, general-purpose test hardware.
LIST OF REFERENCES


APPENDIX A. PAULMON DOCUMENTATION

The following is the documentation file for PAULMON, and although it is a bit rough, it does contain a wealth of valuable information.

Introduction:

The PAULMON debugger is my attempt to make a user-friendly 8051 debugger, with enough on-line information that it should be unnecessary to read this doc file. PAULMON is targeted for use by the microprocessor design course at Oregon State, but may be used by anyone (who can figure it out) for projects ranging from research to commercial products. PAULMON is free and may not be distributed for profit whatsoever.

Since I don't expect Prof's or TA's at OSU to make students aware of this documentation nor to provide it nor do I expect students to read much of it, I wrote PAULMON to be very simple and to provide lots of on-line clues about that it can do and how to go about it. I hope that you find PAULMON to be useful and easy to use. Good Luck.

-Paul Stoffregen
(paul@cce.orst.edu)

DISCLAIMER: This is free software. As far as warranty is concerned, you get exactly what you pay for! I have tried to make this code as good as possible during the four weeks I worked on it, but nobody is perfect and portions (the single step in particular) were never well tested. USE AT YOUR OWN RISK. The assembly source is provided in case there's something you don't like.

ADDITIONAL DISCLAIMER: This doc file has lots of typos and other errors, and I really don't care. PAULMON was written to be easy enough that this file ought to be unnecessary, but people ask for it nonetheless, usually before they even try to use the thing.

What You Will Need to use it:

PAULMON is 8051 assembly code which is intended to be burned into a 2764 EPROM, though a pair of 2732's could be used or a bigger rom can be used with the rest being empty or filled with other code. The EPROM with PAULMON should be addressed so that it is read from 0000 to 1FFF with the 8051's EA pin wired to make it read all code from external memory. PAULMON uses the built in UART in the 8051 to communicate with the user. Typically, a PC computer is used with a terminal program, an 8051 assembler, and a text editor to form a simple, low cost 8051 development system with PAULMON. A serial line receiver and driver should be used (the MAX232 is a good choice, IMHO) to interface the 8051 to the PC's serial port. Only TxD, RxD and ground are used (no handshaking) and PAULMON adapts to use whatever baud rate the computer is using (if it can with the crystal you select, see below)

PAULMON is intended to be used with RAM as well, and the default location for the beginning of the RAM is 2000 (hex), right after the EPROM, though the RAM can be used anywhere in the
range of 0000-FFFF. The read enable signal to the RAM should be the logical OR of the RD and
PSEN signals, so that read attempts to external code memory or program memory spaces will read
from the RAM. (use an AND gate to do the logical OR of these signals, since they are active low!) Obviously the write enable of the RAM should be connected to the WR pin of the 8051.

Having a RAM connected in this way will allow the download command in PAULMON to write
your program into the RAM (writing into the external data memory space). Then you can run your
program, since read attempts from the external program memory space will read from the RAM
chip.

How to get is set up:

Design and build your 8051 board. All that is really required is the 8051, an EPROM, a latch
(74xx373), some sort of address decoding to enable the EPROM for memory access between
0000-1FFF, and a line receiver to convert the high voltage RS232 to a TTL (or CMOS)
compatible signal (or else you'll toast the 8051 before it even has a chance).

To really use PAULMON, a RAM is required as well as the AND gate to allow both program and
data read cycles to read the RAM memory, and a reset button to easily get back to PAULMON
when your program crashes.

With just the minimal setup, set the computer's baud rate to something slow (like 1200 bps) and
power up the board. Press Enter (Return) and hopefully you'll see a screenful of text from
PAULMON. PAULMON does not send line feed characters, so the terminal emulator software
must be configured to translate CR -> CR/LF. (PAULMON ignores LF characters is receives.) If
the entire message ends up on one line, then the terminal is not translating CR -> CR/LF. After it
works, you can try increasing the baud rate and COLD-BOOTING (you must turn the power off,
taking the reset line high will not make PAULMON look for the new baud rate... or change the
bytes where it stores the old baud rate... see the code if you're interested) If the minimal system
shows no signs of life, it's time to check the wiring, usually starting by making sure you didn't
swap the TxD and RxD lines.

The Automatic Baud Rate detection:

This code was borrowed from MDP/51 by Kei-Yong Khoo. It is run immediately after a system
reset. It waits for a <RETURN> character, and uses it to calculate the timer #1 reload value.
Some modifications have been made to Khoo's code. It requires only one character. It also stores
the reload value in four memory locations in internal ram (78H, 79H, 7AH, and 7BH). These
four locations are unlikely to be changed during a user program's execution or while the debugger
is running. When another reset occurs (without removing the power) the program looks at those
four locations. If all four agree, then it uses that reload value and does not require another
keypress. It is interesting to note that occasionally, with crystal values which produce exact reload values (such an 7.3728 MHz), the baud rate detection routine may not correctly calculate the
reload value. Garbage will get printed all over the screen. If this happens, just switch off the
power and try again. The advantage of crystals such as the 7.3728 MHz is that they allow
transmission at speeds of 9600 and 19200 baud! It is highly recommended that you use the
highest possible baud rate with this debugger, as is tends to print quite a bit of text to the screen.

On-line Help:
By typing "?" at the main menu, a help screen summarizing the available commands is printed. Online help is also available regarding the single step run feature. This help is accessed by typing "?" just after using the 'R' command. While in the single step mode, a summary of commands is also available, again by typing "?".

The <ESC> key:

The <ESC> key is supported extensively. It will abort all commands from any prompt. It will stop the list and hex dump commands in the middle of their printing. It will also interrupt the printing of text to the screen! This is useful at slow baud rates, since a full screen of text can take quite a while to print at 300 baud.

The Download Program command (type 'D')

This allows you to send the object code from the assembler to the external ram. The object file must be a standard Intel Hex Format file, such as the .OBJ file created by the Psuedo-Assembler, by Psuedo-Corp. The file must be sent as an ASCII transfer. The protocol such as XMODEM is used. Pressing the <ESC> key at any time will abort the transfer. Please note that most communications programs use the <ESC> key to abort their transfer. In this case, the first <ESC> will halt the terminal, pressing it again will abort the receive at the 8051/31. Unlike some other debuggers, PAULMON will recognize the <ESC> key anywhere in the middle of the incoming data, not just at the beginning of a line.

The Run Program command (type 'R')

The run command allows you to execute your program. Two types of run are supported, Normal and Single-Step. The single step mode is explained later, as it is fairly complex. During a normal run, the quivilient of an LCALL to your code is given. During the execution of your program, the debugger obviously has no control of the system, unless of course your program calls one of the subroutines offered by the debugger in the jump table at location 0030H. After specifying which run mode you need, the location of your program is prompted, with the current memory pointer value as the default choice. As is the case at all prompts, pressing the <ESC> key will abort the run command. It is interesting to note that the run command leaves timer #1 in auto-baud rate generation mode. If serial communication is desired at the same baud rate as that used for the debugger, timer #1 need not be given a new reload value. It is recommended that the character input and output routines from the debugger be used via the jump table.

The New Memory Location command (type 'N')

The debugger operates with a pointer to the data memory with which you are working. This pointer is used by the list and hex dump command. It is also the default run location. The pointer is incremented as memory is viewed or modified. Just type 'N' to change it.

The List command (type 'L')

This debugger gives you the ability to list your program's code directly from the computer's memory. All the 8051/31 mnemonicics are supports, as well as the names of the special function registers. Bit addressable locations are displayed using the standard syntax (e.g. PWS.2 or 20.5), but individual bit location names are not supported (e.g. SCON.0 will print in place of RI).

Obviously, the original labels used in the source code cannot be printed, instead the memory locations are displayed. Other special Intel assembly formats, such as $ and CALL are not
supported. However, the list command can provide a reassuring look at the program directly from the memory.

The Hex Dump command (type 'H')

By typing 'H', the next 256 bytes of ram are dumped to the screen in hex and ascii. The <ESC> key may be pressed to abort the printout.

The Edit command (type 'E')

This command allows you to change the values of memory locations in the external ram. Each location's old value is shown. If <ESC> is pressed, the current location's value is not changed.

The Jump Table:

Despite the use of the word "jump", the user must LCALL to these locations! The individual locations contain jumps to the subroutines, which all terminate with a RET. The table provides the user with a memory location to call to that WILL NOT CHANGE if the debugger is reassembled. The routines available are:

- 0030: Cout - Sends the byte in Acc to the serial port.
- 0032: Cin - Waits for a character from the serial port, returned in Acc.
- 0034: pHex - Prints the two digit hex value in Acc to the serial port.
- 0036: pHex16 - Prints the four digit hex value in DPTR to the serial port.
- 0038: pString - Prints the string in code memory pointed to by DPTR to the serial port.

  The string must terminate with 00H or a high bit set

- 003A: gHex - Gets a two digit hex value from the serial port, returned in Acc
- 003C: gHex16 - Gets a four digit hex value from the serial port, returned in DPTR
- 003E: Esc - Checks to see if the <ESC> key is waiting in SBUF. Clears the buffer if it is, and returns with the carry set. Otherwise, leaves SBUF untouched, and returns with C=0.
- 0040: Upper - Converts character in Acc to uppercase if it is lowercase
- 0042: Init - Automatic baud rate detection.

The memory location can be placed directly in your code, or an EQU can be used to make your code more readable. For example:

Program::EQU gHex16, 003AH ; this make the code nice

MOV DPTR, #StrLoc ; load DPTR
LCALL gHex16 ; print the DPTR
MOV A,#13
LCALL 0030H ; print a <RET>
LCALL 0038H ; print the string
RETLoc: .DB "This is my String.", 0

Most of these routines leave the registers unchanged, however, it is a good idea to consult the source code just to be sure... In particular, the phex routine DESTROYS the contents of Acc, so beware. (this has caused some people some grief, who had assumed the phex would leave Acc unchanged. If you want it unchanged, the original .ASM file is provided for you to modify) The Single-Step Run:

[This part was never written, and the single step run code is somewhat buggy, primarily due to a lack of available beta testers... so docs were never written, but PAULMON ought to give you enough clues to figure it out if you try.]
APPENDIX B. 8051 PROGRAMMERS REFERENCE
PROGRAMMER'S GUIDE AND INSTRUCTION SET

Memory Organization

Program Memory
The 80C51 has separate address spaces for program and data memory. The Program memory can be up to 64k bytes long. The lower 4k can reside on-chip. Figure 1 shows a map of the 80C51 program memory.

The 80C51 can address up to 64k bytes of data memory to the chip. The MOVX instruction is used to access the external data memory.

The 80C51 has 128 bytes of on-chip RAM, plus a number of Special Function Registers (SFRs). The lower 128 bytes of RAM can be accessed either by direct addressing (MOV data addr) or by indirect addressing (MOV @Ri). Figure 2 shows the Data Memory organization.

Direct and Indirect Address Area
The 128 bytes of RAM which can be accessed by both direct and indirect addressing can be divided into three segments as listed below and shown in Figure 3.

1. Register Banks 0-3: Locations 0 through 1FH (32 bytes). The device after reset defaults to register bank 0. To use the other register banks, the user must select them in software. Each register bank contains eight 1-byte registers 0 through 7. Reset initializes the stack pointer to location 07H, and it is incremented once to start from location 06H, which is the first register (R0) of the second register bank. Thus, in order to use more than one register bank, the SP should be initialized to a different location of the RAM where it is not used for data storage (i.e., the higher part of the RAM).

2. Bit Addressable Area: 16 bytes have been assigned for this segment, 20H-2FH.

Each one of the 128 bits of this segment can be directly addressed (0-7FH). The bits can be referred to in two ways, both of which are acceptable by most assemblers. One way is to refer to their address (i.e., 0-7FH). The other way is with reference to bytes 20H to 2FH. Thus, bits 0-7 can also be referred to as bits 20.0-20.7, and bits 8-FH are the same as 21.0-21.7, and so on. Each of the 16 bytes in this segment can also be addressed as a byte.

3. Scratch Pad Area: 30H through 7FH are available to the user as data RAM. However, if the stack pointer has been initialized to this area, enough bytes should be left aside to prevent SP data destruction.

Figure 2 shows the different segments of the on-chip RAM.

![Figure 1. 80C51 Program Memory](image)
Figure 2. 80C51 Data Memory

Figure 3. 128 Bytes of RAM Direct and Indirect Addressable
# 80C51 Family

80C51 family programmer’s guide and instruction set

## Table 1. 80C51 Special Function Registers

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>DESCRIPTION</th>
<th>DIRECT ADDRESS</th>
<th>BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION MSB</th>
<th>LSB</th>
<th>RESET VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC*</td>
<td>Accumulator</td>
<td>E0H</td>
<td>E7 E6 E5 E4 E3 E2 E1 E0</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>B*</td>
<td>B register</td>
<td>F0H</td>
<td>F7 F6 F5 F4 F3 F2 F1 F0</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>DPTR</td>
<td>Data pointer (2 bytes)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DPH</td>
<td>Data pointer high</td>
<td>83H</td>
<td></td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>DPL</td>
<td>Data pointer low</td>
<td>82H</td>
<td></td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>IE*</td>
<td>Interrupt enable</td>
<td>A8H</td>
<td>EA - - - ES ET1 EX1 EX0</td>
<td></td>
<td>00000000B</td>
</tr>
<tr>
<td>IP*</td>
<td>Interrupt priority</td>
<td>B8H</td>
<td>- - - - PS PT1 PX1 PT0 PX0</td>
<td></td>
<td>xx000000B</td>
</tr>
<tr>
<td>P0*</td>
<td>Port 0</td>
<td>80H</td>
<td>AD7 AD6 AD5 AD4 AD3 AD2 AD1 AD0</td>
<td></td>
<td>FFH</td>
</tr>
<tr>
<td>P1*</td>
<td>Port 1</td>
<td>90H</td>
<td>97 96 95 94 93 92 91 90</td>
<td></td>
<td>FFH</td>
</tr>
<tr>
<td>P2*</td>
<td>Port 2</td>
<td>A0H</td>
<td>A7 A6 A5 A4 A3 A2 A1 A0</td>
<td></td>
<td>FFH</td>
</tr>
<tr>
<td>P3*</td>
<td>Port 3</td>
<td>B0H</td>
<td>RD WR T1 T0 INTT INT0 TxD RxD</td>
<td></td>
<td>FFH</td>
</tr>
<tr>
<td>PCON†</td>
<td>Power control</td>
<td>87H</td>
<td>SMOD - - - GF1 GF0 PD IDL</td>
<td></td>
<td>00000000B</td>
</tr>
<tr>
<td>PCON†</td>
<td></td>
<td></td>
<td>D7 D6 D5 D4 D3 D2 D1 D0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSW*</td>
<td>Program status word</td>
<td>D0H</td>
<td>CY AC F0 RS1 RS0 OV - - P</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>SBUF</td>
<td>Serial data buffer</td>
<td>99H</td>
<td></td>
<td></td>
<td>xo000000B</td>
</tr>
<tr>
<td>SCON*</td>
<td>Serial controller</td>
<td>98H</td>
<td>SM0 SM1 SM2 REN TB8 RB8 TI RI</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>SP</td>
<td>Stack pointer</td>
<td>81H</td>
<td></td>
<td></td>
<td>07H</td>
</tr>
<tr>
<td>TCON*</td>
<td>Timer control</td>
<td>88H</td>
<td>TF1 TR1 TF0 TR0 IE1 IT1 IE0 IT0</td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>TH0</td>
<td>Timer high 0</td>
<td>8CH</td>
<td></td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>TH1</td>
<td>Timer high 1</td>
<td>8DH</td>
<td></td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>TL0</td>
<td>Timer low 0</td>
<td>8AH</td>
<td></td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>TL1</td>
<td>Timer low 1</td>
<td>8BH</td>
<td></td>
<td></td>
<td>00H</td>
</tr>
<tr>
<td>TMOD</td>
<td>Timer mode</td>
<td>89H</td>
<td>GATE C/T M1 M0 GATE C/T M1 M0</td>
<td></td>
<td>00H</td>
</tr>
</tbody>
</table>

**NOTES:**

* Bit addressable

1. Bits GF1, GF0, PD, and IDL of the PCON register are not implemented on the NMOS 8051/8031.
Figure 4. SFR Memory Map
Those SFRs that have their bits assigned for various functions are listed in this section. A brief description of each bit is provided for quick reference. For more detailed information refer to the Architecture Chapter of this book.

**PSW: PROGRAM STATUS WORD. BIT ADDRESSABLE.**

<table>
<thead>
<tr>
<th>CY</th>
<th>AC</th>
<th>F0</th>
<th>RS1</th>
<th>RS0</th>
<th>OV</th>
<th>–</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>CY</td>
<td>PSW.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC</td>
<td>PSW.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F0</td>
<td>PSW.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RS1</td>
<td>PSW.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RS0</td>
<td>PSW.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OV</td>
<td>PSW.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>–</td>
<td>PSW.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>PSW.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:**

1. The value presented by RS0 and RS1 selects the corresponding register bank.

<table>
<thead>
<tr>
<th>RS1</th>
<th>RS0</th>
<th>REGISTER BANK</th>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>00H-07H</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>08H-0FH</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>10H-17H</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>18H-1FH</td>
</tr>
</tbody>
</table>

**PCON: POWER CONTROL REGISTER. NOT BIT ADDRESSABLE.**

<table>
<thead>
<tr>
<th>SMOD</th>
<th>–</th>
<th>–</th>
<th>–</th>
<th>–</th>
<th>GF1</th>
<th>GF0</th>
<th>PD</th>
<th>IDL</th>
</tr>
</thead>
</table>

**SMOD**

- Double baud rate bit. If Timer 1 is used to generate baud rate and SMOD = 1, the baud rate is doubled when the Serial Port is used in modes 1, 2, or 3.
- Not implemented, reserved for future use.*
- Not implemented reserved for future use.*
- Not implemented reserved for future use.*

**GF1**

General purpose flag bit.

**GF0**

General purpose flag bit.

**PD**

Power Down Bit. Setting this bit activates Power Down operation in the 80C51. (Available only in CMOS.)

**IDL**

Idle mode bit. Setting this bit activates Idle Mode operation in the 80C51. (Available only in CMOS.)

If 1s are written to PD and IDL at the same time, PD takes precedence.

* User software should not write 1s to reserved bits. These bits may be used in future 8051 products to invoke new features.
INTERRUPTS:
To use any of the interrupts in the 80C51 Family, the following three steps must be taken.
1. Set the EA (enable all) bit in the IE register to 1.
2. Set the corresponding individual interrupt enable bit in the IE register to 1.
3. Begin the interrupt service routine at the corresponding Vector Address of that interrupt. See Table below.

<table>
<thead>
<tr>
<th>INTERRUPT SOURCE</th>
<th>VECTOR ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE0</td>
<td>0003H</td>
</tr>
<tr>
<td>TF0</td>
<td>000BH</td>
</tr>
<tr>
<td>IE1</td>
<td>0013H</td>
</tr>
<tr>
<td>TF1</td>
<td>001BH</td>
</tr>
<tr>
<td>RI &amp; TI</td>
<td>0023H</td>
</tr>
</tbody>
</table>

In addition, for external interrupts, pins INT0 and INT1 (P3.2 and P3.3) must be set to 1, and depending on whether the interrupt is to be level or transition activated, bits IT0 or IT1 in the TCON register may need to be set to 1.
ITx = 0 level activated
ITx = 1 transition activated

IE: INTERRUPT ENABLE REGISTER. BIT ADDRESSABLE.
If the bit is 0, the corresponding interrupt is disabled. If the bit is 1, the corresponding interrupt is enabled.

<table>
<thead>
<tr>
<th>EA</th>
<th>ES</th>
<th>ET1</th>
<th>EX1</th>
<th>ET0</th>
<th>EX0</th>
</tr>
</thead>
<tbody>
<tr>
<td>EA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>—</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ES</td>
<td>IE.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ET1</td>
<td>IE.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EX1</td>
<td>IE.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ET0</td>
<td>IE.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EX0</td>
<td>IE.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

EA: Disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.

ES: Enable or disable the serial port interrupt.
ET1: Enable or disable the Timer 1 overflow interrupt.
EX1: Enable or disable External Interrupt 1.
ET0: Enable or disable the Timer 0 overflow interrupt.
EX0: Enable or disable External Interrupt 0.

* User software should not write 1s to reserved bits. These bits may be used in future 80C51 products to invoke new features.
ASSIGNING HIGHER PRIORITY TO ONE OR MORE INTERRUPTS:
In order to assign higher priority to an interrupt the corresponding bit in the IP register must be set to 1.
Remember that while an interrupt service is in progress, it cannot be interrupted by a lower or same level interrupt.

PRIORITY WITHIN LEVEL:
Priority within level is only to resolve simultaneous requests of the same priority level.
From high to low, interrupt sources are listed below:
IE0
TF0
IE1
TF1
RI or TI

IP: INTERRUPT PRIORITY REGISTER. BIT ADDRESSABLE.
If the bit is 0, the corresponding interrupt has a lower priority and if the bit is 1 the corresponding interrupt has a higher priority.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>PS</th>
<th>PT1</th>
<th>PX1</th>
<th>PT0</th>
<th>PX0</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td></td>
<td></td>
<td>IP.7</td>
<td>Not implemented, reserved for future use.*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-</td>
<td></td>
<td></td>
<td>IP.6</td>
<td>Not implemented, reserved for future use.*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-</td>
<td></td>
<td></td>
<td>IP.5</td>
<td>Not implemented, reserved for future use.*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PS</td>
<td></td>
<td></td>
<td>IP.4</td>
<td>Defines the Serial Port interrupt priority level.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT1</td>
<td></td>
<td></td>
<td>IP.3</td>
<td>Defines the Timer 1 interrupt priority level.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PX1</td>
<td></td>
<td></td>
<td>IP.2</td>
<td>Defines External Interrupt 1 priority level.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT0</td>
<td></td>
<td></td>
<td>IP.1</td>
<td>Defines the Timer 0 interrupt priority level.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PX0</td>
<td></td>
<td></td>
<td>IP.0</td>
<td>Defines the External Interrupt 0 priority level.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* User software should not write 1s to reserved bits. These bits may be used in future 80C51 products to invoke new features.
TCON: TIMER/COUNTER CONTROL REGISTER. BIT ADDRESSABLE.

<table>
<thead>
<tr>
<th>TF1</th>
<th>TR1</th>
<th>TF0</th>
<th>TR0</th>
<th>IE1</th>
<th>IT1</th>
<th>IE0</th>
<th>IT0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF1</td>
<td>TCON.7</td>
<td>Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TR1</td>
<td>TCON.6</td>
<td>Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1 ON/OFF.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TF0</td>
<td>TCON.5</td>
<td>Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TR0</td>
<td>TCON.4</td>
<td>Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IE1</td>
<td>TCON.3</td>
<td>External Interrupt 1 edge flag. Set by hardware when External Interrupt edge is detected. Cleared by hardware when interrupt is processed.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IT1</td>
<td>TCON.2</td>
<td>Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IE0</td>
<td>TCON.1</td>
<td>External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IT0</td>
<td>TCON.0</td>
<td>Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TMOD: TIMER/COUNTER MODE CONTROL REGISTER. NOT BIT ADDRESSABLE.

<table>
<thead>
<tr>
<th>GATE</th>
<th>C/T</th>
<th>M1</th>
<th>M0</th>
<th>GATE</th>
<th>C/T</th>
<th>M1</th>
<th>M0</th>
</tr>
</thead>
</table>

Timer 1

Timer 0

GATE
When TRx (in TCON) is set and GATE = 1, TIMER/COUNTERx will run only while INTx pin is high (hardware control).
When GATE = 0, TIMER/COUNTERx will run only while TRx = 1 (software control).

C/T
Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).

M1
Mode selector bit. (NOTE 1)

M0
Mode selector bit. (NOTE 1)

NOTE 1:

<table>
<thead>
<tr>
<th>M1</th>
<th>M0</th>
<th>Operating Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>13-bit Timer (8048 compatible)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>16-bit Timer/Counter</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>8-bit Auto-Reload Timer/Counter</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standart Timer 0 control bits. TH0 is an8-bit Timer and is controlled by Timer 1 control bits.</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>(Timer 1) Timer/Counter 1 stopped.</td>
</tr>
</tbody>
</table>

March 1995
**TIMER SET-UP**

Tables 2 through 5 give some values for TMOD which can be used to set up Timer 0 in different modes.

It is assumed that only one timer is being used at a time. If it is desired to run Timers 0 and 1 simultaneously, in any mode, the value in TMOD for Timer 0 must be ORed with the value shown for Timer 1 (Tables 5 and 6).

For example, if it is desired to run Timer 0 in mode 1 GATE (external control), and Timer 1 in mode 2 COUNTER, then the value that must be loaded into TMOD is 69H (09H from Table 2 ORed with 60H from Table 5).

Moreover, it is assumed that the user, at this point, is not ready to turn the timers on and will do that at a different point in the program by setting bit TRx (in TCON) to 1.

**TIMER/COUNTER 0**

**Table 2. As a Timer:**

<table>
<thead>
<tr>
<th>MODE</th>
<th>TIMER 0 FUNCTION</th>
<th>TMOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>13-bit Timer</td>
<td>INTERNAL CONTROL (NOTE 1)</td>
</tr>
<tr>
<td>1</td>
<td>16-bit Timer</td>
<td>01H</td>
</tr>
<tr>
<td>2</td>
<td>8-bit Auto-Reload</td>
<td>02H</td>
</tr>
<tr>
<td>3</td>
<td>Two 8-bit Timers</td>
<td>03H</td>
</tr>
</tbody>
</table>

**Table 3. As a Counter:**

<table>
<thead>
<tr>
<th>MODE</th>
<th>COUNTER 0 FUNCTION</th>
<th>TMOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>13-bit Timer</td>
<td>INTERNAL CONTROL (NOTE 1)</td>
</tr>
<tr>
<td>1</td>
<td>16-bit Timer</td>
<td>05H</td>
</tr>
<tr>
<td>2</td>
<td>8-bit Auto-Reload</td>
<td>06H</td>
</tr>
<tr>
<td>3</td>
<td>One 8-bit Counter</td>
<td>07H</td>
</tr>
</tbody>
</table>

**NOTES:**
1. The timer is turned ON/OFF by setting/clearing bit TR0 in the software.
2. The Timer is turned ON/OFF by the 1-to-0 transition on INT0 (P3.2) when TR0 = 1 (hardware control).
### TIMER/COUNTER 1

#### Table 4. As a Timer:

<table>
<thead>
<tr>
<th>MODE</th>
<th>TIMER 1 FUNCTION</th>
<th>TMOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>13-bit Timer</td>
<td>00H, 80H</td>
</tr>
<tr>
<td>1</td>
<td>16-bit Timer</td>
<td>10H, 90H</td>
</tr>
<tr>
<td>2</td>
<td>8-bit Auto-Reload</td>
<td>20H, A0H</td>
</tr>
<tr>
<td>3</td>
<td>Does not run</td>
<td>30H, B0H</td>
</tr>
</tbody>
</table>

#### Table 5. As a Counter:

<table>
<thead>
<tr>
<th>MODE</th>
<th>COUNTER 1 FUNCTION</th>
<th>TMOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>13-bit Timer</td>
<td>40H, C0H</td>
</tr>
<tr>
<td>1</td>
<td>16-bit Timer</td>
<td>50H, D0H</td>
</tr>
<tr>
<td>2</td>
<td>8-bit Auto-Reload</td>
<td>60H, E0H</td>
</tr>
<tr>
<td>3</td>
<td>Not available</td>
<td>–</td>
</tr>
</tbody>
</table>

**NOTES:**
1. The timer is turned ON/OFF by setting/clearing bit TR1 in the software.
2. The Timer is turned ON/OFF by the 1-to-0 transition on INT1 (P3.2) when TR1 = 1 (hardware control).
SCON: SERIAL PORT CONTROL REGISTER. BIT ADDRESSABLE.

<table>
<thead>
<tr>
<th>SM0</th>
<th>SM1</th>
<th>SM2</th>
<th>REN</th>
<th>TB8</th>
<th>RB8</th>
<th>Ti</th>
<th>RI</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM0</td>
<td>SCON.7</td>
<td>Serial Port mode specifier. (NOTE 1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SM1</td>
<td>SCON.6</td>
<td>Serial Port mode specifier. (NOTE 1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SM2</td>
<td>SCON.5</td>
<td>Enables the multiprocessor communication feature in modes 2 &amp; 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0. (See Table 6.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REN</td>
<td>SCON.4</td>
<td>Set/Cleared by software to Enable/Disable reception</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TB8</td>
<td>SCON.3</td>
<td>The 9th bit that will be transmitted in modes 2 &amp; 3. Set/Cleared by software</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RB8</td>
<td>SCON.2</td>
<td>In modes 2 &amp; 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ti</td>
<td>SCON.1</td>
<td>Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes. Must be cleared by software</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RI</td>
<td>SCON.0</td>
<td>Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes (except see SM2). Must be cleared by software</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTE 1:

<table>
<thead>
<tr>
<th>SM0</th>
<th>SM1</th>
<th>Mode</th>
<th>Description</th>
<th>Baud Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Shift Register</td>
<td>$FOSC/12$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>8-bit UART</td>
<td>Variable</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>9-bit UART</td>
<td>$FOSC/64$ or $FOSC/32$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>9-bit UART</td>
<td>Variable</td>
</tr>
</tbody>
</table>

SERIAL PORT SET-UP:

Table 6.

<table>
<thead>
<tr>
<th>MODE</th>
<th>SCON</th>
<th>SM2 VARIATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10H</td>
<td>Single Processor Environment (SM2 = 0)</td>
</tr>
<tr>
<td>1</td>
<td>50H</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>90H</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>D0H</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>NA</td>
<td>Multiprocessor Environment (SM2 = 1)</td>
</tr>
<tr>
<td>1</td>
<td>70H</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>80H</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>F0H</td>
<td></td>
</tr>
</tbody>
</table>

GENERATING BAUD RATES

Serial Port in Mode 0:

Mode 0 has a fixed baud rate which is 1/12 of the oscillator frequency. To run the serial port in this mode none of the Timer/Counters need to be set up. Only the SCON register needs to be defined.

$$\text{Baud Rate} = \frac{\text{Osc Freq}}{12}$$

Serial Port in Mode 1:

Mode 1 has a variable baud rate. The baud rate is generated by Timer 1.
USING TIMER/COUNTER 1 TO GENERATE BAUD RATES:

For this purpose, Timer 1 is used in mode 2 (Auto-Reload). Refer to Timer Setup section of this chapter.

\[
\text{Baud Rate} = \frac{K \text{ Osc Freq}}{32 \frac{256}{(\text{TH1})}}
\]

If SMOD = 0, then K = 1.
If SMOD = 1, then K = 2 (SMOD is in the PCON register).

Most of the time the user knows the baud rate and needs to know the reload value for TH1.

\[
\text{TH1} = 256 \frac{384}{\text{Osc Freq}} \frac{384}{\text{baud rate}}
\]

TH1 must be an integer value. Rounding off TH1 to the nearest integer may not produce the desired baud rate. In this case, the user may have to choose another crystal frequency.

Since the PCON register is not bit addressable, one way to set the bit is logical OR'ing the PCON register (i.e., ORL PCON,#80H). The address of PCON is 87H.

SERIAL PORT IN MODE 2:

The baud rate is fixed in this mode and is 1/32 or 1/64 of the oscillator frequency, depending on the value of the SMOD bit in the PCON register.

In this mode none of the Timers are used and the clock comes from the internal phase 2 clock.

SMOD = 1, Baud Rate = 1/32 Osc Freq.
SMOD = 0, Baud Rate = 1/64 Osc Freq.

To set the SMOD bit: ORL PCON,#80H. The address of PCON is 87H.

SERIAL PORT IN MODE 3:

The baud rate in mode 3 is variable and sets up exactly the same as in mode 1.
## 80C51 FAMILY INSTRUCTION SET

### Table 7. 80C51 Instruction Set Summary

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Flag</th>
<th>Instruction</th>
<th>Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>ADDC</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SUBB</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>MUL</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DIV</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DA</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RRC</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RLC</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SETB C</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(1) Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

### Notes on instruction set and addressing modes:

- **Rn**: Register R7-R0 of the currently selected Register Bank.
- **direct**: 8-bit internal data location's address. This could be an Internal Data RAM location (0-127) or a SFR [i.e., I/O port, control register, status register, etc. (128-255)].
- **@Ri**: 8-bit internal data RAM location (0-255) addressed indirectly through register R1 or R0.
- **#data**: 8-bit constant included in the instruction.
- **#data 16**: 16-bit constant included in the instruction.
- **addr 16**: 16-bit destination address. Used by LCALL and LJMP. A branch can be anywhere within the 64k-byte Program Memory address space.
- **addr 11**: 11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2k-byte page of program memory as the first byte of the following instruction.
- **rel**: Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is –128 to +127 bytes relative to first byte of the following instruction.
- **bit**: Direct Addressed bit in Internal Data RAM or Special Function Register.

### ARITHMETIC OPERATIONS

<table>
<thead>
<tr>
<th>MNEMONIC</th>
<th>DESCRIPTION</th>
<th>BYTE</th>
<th>OSCILLATOR PERIOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>A,Rn</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>ADD</td>
<td>A,direct</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>ADD</td>
<td>A,@Ri</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>ADD</td>
<td>A,#data</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>ADDC</td>
<td>A,Rn</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>ADDC</td>
<td>A,direct</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>ADDC</td>
<td>A,@Ri</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>ADDC</td>
<td>A,#data</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>SUBB</td>
<td>A,Rn</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>SUBB</td>
<td>A,direct</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>SUBB</td>
<td>A,@Ri</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>SUBB</td>
<td>A,#data</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>INC</td>
<td>A</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>INC</td>
<td>Rn</td>
<td>1</td>
<td>12</td>
</tr>
</tbody>
</table>

All mnemonics copyrighted © Intel Corporation 1980
### Table 7. 80C51 Instruction Set Summary (Continued)

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Description</th>
<th>Byte</th>
<th>Oscillator Period</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ARITHMETIC OPERATIONS</strong> (Continued)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INC</td>
<td>direct</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>INC @Ri</td>
<td>Increment indirect RAM</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>DEC</td>
<td>A</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>DEC Rn</td>
<td>Decrement Register</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>DEC direct</td>
<td>Decrement direct byte</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>DEC @Ri</td>
<td>Decrement indirect RAM</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>INC DPTR</td>
<td>Increment Data Pointer</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>MUL</td>
<td>AB</td>
<td>1</td>
<td>48</td>
</tr>
<tr>
<td>DIV</td>
<td>AB</td>
<td>1</td>
<td>48</td>
</tr>
<tr>
<td>DA</td>
<td>A</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td><strong>LOGICAL OPERATIONS</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANL</td>
<td>A,Rn</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>ANL A,direct</td>
<td>AND direct byte to Accumulator</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>ANL A,@Ri</td>
<td>AND indirect RAM to Accumulator</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>ANL A,#data</td>
<td>AND immediate data to Accumulator</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>ANL direct,A</td>
<td>AND Accumulator to direct byte</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>ANL direct,#data</td>
<td>AND immediate data to direct byte</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>ORL</td>
<td>A,Rn</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>ORL A,direct</td>
<td>OR direct byte to Accumulator</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>ORL A,@Ri</td>
<td>OR indirect RAM to Accumulator</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>ORL A,#data</td>
<td>OR immediate data to Accumulator</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>ORL direct,A</td>
<td>OR Accumulator to direct byte</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>ORL direct,#data</td>
<td>OR immediate data to direct byte</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>XRL</td>
<td>A,Rn</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>XRL A,direct</td>
<td>Exclusive-OR direct byte to Accumulator</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>XRL A,@Ri</td>
<td>Exclusive-OR indirect RAM to Accumulator</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>XRL A,#data</td>
<td>Exclusive-OR immediate data to Accumulator</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>XRL direct,A</td>
<td>Exclusive-OR Accumulator to direct byte</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>XRL direct,#data</td>
<td>Exclusive-OR immediate data to direct byte</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>CLR</td>
<td>A</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>CPL</td>
<td>A</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>RL</td>
<td>A</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>RLC</td>
<td>A</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>RR</td>
<td>A</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>RRC</td>
<td>A</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>SWAP</td>
<td>A</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td><strong>DATA TRANSFER</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOV</td>
<td>A,Rn</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>MOV A,direct</td>
<td>Move direct byte to Accumulator</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>MOV A,@Ri</td>
<td>Move indirect RAM to Accumulator</td>
<td>1</td>
<td>12</td>
</tr>
</tbody>
</table>

All mnemonics copyrighted © Intel Corporation 1980
### Table 7. 80C51 Instruction Set Summary (Continued)

<table>
<thead>
<tr>
<th>MNEMONIC</th>
<th>DESCRIPTION</th>
<th>BYTE</th>
<th>OSCILLATOR PERIOD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DATA TRANSFER (Continued)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOV A,#data</td>
<td>Move immediate data to Accumulator</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>MOV Rn,A</td>
<td>Move Accumulator to register</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>MOV Rn,direct</td>
<td>Move direct byte to register</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>MOV RN,#data</td>
<td>Move immediate data to register</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>MOV direct,A</td>
<td>Move Accumulator to direct byte</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>MOV direct,Rn</td>
<td>Move register to direct byte</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>MOV direct,direct</td>
<td>Move direct byte to direct</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>MOV direct,@Ri</td>
<td>Move indirect RAM to direct byte</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>MOV direct,#data</td>
<td>Move immediate data to direct byte</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>MOV @Ri,A</td>
<td>Move Accumulator to indirect RAM</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>MOV @Ri,direct</td>
<td>Move direct byte to indirect RAM</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>MOV @Ri,#data</td>
<td>Move immediate data to indirect RAM</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>MOV DPTR,#data16</td>
<td>Load Data Pointer with a 16-bit constant</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>MOVC A,#data</td>
<td>Move Code byte relative to DPTR to ACC</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>MOVC A,#data16</td>
<td>Move Code byte relative to PC to ACC</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>MOVX A,#data</td>
<td>Move external RAM (8-bit addr) to ACC</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>MOVX A,#data16</td>
<td>Move external RAM (16-bit addr) to ACC</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>MOVX A,#Ri,A</td>
<td>Move ACC to external RAM (8-bit addr)</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>MOVX @DPTR,A</td>
<td>Move ACC to external RAM (16-bit addr)</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>PUSH direct</td>
<td>Push direct byte onto stack</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>POP direct</td>
<td>Pop direct byte from stack</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>XCH A,Rn</td>
<td>Exchange register with Accumulator</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>XCH A,direct</td>
<td>Exchange direct byte with Accumulator</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>XCH A,#Ri</td>
<td>Exchange indirect RAM with Accumulator</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>XCHD A,#Ri</td>
<td>Exchange low-order digit indirect RAM with ACC</td>
<td>1</td>
<td>12</td>
</tr>
</tbody>
</table>

### BOOLEAN VARIABLE MANIPULATION

<table>
<thead>
<tr>
<th>MNEMONIC</th>
<th>DESCRIPTION</th>
<th>BYTE</th>
<th>OSCILLATOR PERIOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLR C</td>
<td>Clear carry</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>CLR bit</td>
<td>Clear direct bit</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>SETB C</td>
<td>Set carry</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>SETB bit</td>
<td>Set direct bit</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>CPL C</td>
<td>Complement carry</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>CPL bit</td>
<td>Complement direct bit</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>ANL C,bit</td>
<td>AND direct bit to carry</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>ANL C,bit</td>
<td>AND complement of direct bit to carry</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>ORL C,bit</td>
<td>OR direct bit to carry</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>ORL C,bit</td>
<td>OR complement of direct bit to carry</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>MOV C,bit</td>
<td>Move direct bit to carry</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>MOV bit,C</td>
<td>Move carry to direct bit</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>JC rel</td>
<td>Jump if carry is set</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>JNC rel</td>
<td>Jump if carry not set</td>
<td>2</td>
<td>24</td>
</tr>
</tbody>
</table>

All mnemonics copyrighted © Intel Corporation 1980
### Table 7. 80C51 Instruction Set Summary (Continued)

<table>
<thead>
<tr>
<th>MNEMONIC</th>
<th>DESCRIPTION</th>
<th>BYTE</th>
<th>OSCILLATOR PERIOD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BOOLEAN VARIABLE MANIPULATION (Continued)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JB</td>
<td>rel Jump if direct bit is set</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>JNB</td>
<td>rel Jump if direct bit is not set</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>JBC</td>
<td>bit,rel Jump if direct bit is set and clear bit</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td><strong>PROGRAM BRANCHING</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACAALL</td>
<td>addr11 Absolute subroutine call</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>LCALL</td>
<td>addr16 Long subroutine call</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>RET</td>
<td>Return from subroutine</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>RETI</td>
<td>Return from interrupt</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>AJMP</td>
<td>addr11 Absolute jump</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>LJMP</td>
<td>addr16 Long jump</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>SJMP</td>
<td>rel Short jump (relative addr)</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>JMP</td>
<td>@A+DPTR Jump indirect relative to the DPTR</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>JZ</td>
<td>rel Jump if Accumulator is zero</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>JNZ</td>
<td>rel Jump if Accumulator is not zero</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>CJNE</td>
<td>A,direct,rel Compare direct byte to $A_{CC}$ and jump if not equal</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>CJNE</td>
<td>A,#data,rel Compare immediate to $A_{CC}$ and jump if not equal</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>CJNE</td>
<td>RN,#data,rel Compare immediate to register and jump if not equal</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>CJNE</td>
<td>@Ri,#data,rel Compare immediate to indirect and jump if not equal</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>DJNZ</td>
<td>Rn,rel Decrement register and jump if not zero</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>DJNZ</td>
<td>direct,rel Decrement direct byte and jump if not zero</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>NOP</td>
<td>No operation</td>
<td>1</td>
<td>12</td>
</tr>
</tbody>
</table>

All mnemonics copyrighted © Intel Corporation 1980

March 1995
INSTRUCTION DEFINITIONS

ACALL  addr11

Function: Absolute Call

Description: ACALL unconditionally calls a subroutine located at the indicated address. The instruction increments the PC twice to obtain the address of the following instruction, then pushes the 16-bit result onto the stack (low-order byte first) and increments the Stack Pointer twice. The destination address is obtained by successively concatenating the five high-order bits of the incremented PC, opcode bits 7-5, and the second byte of the instruction. The subroutine called must therefore start within the same 2k block of the program memory as the first byte of the instruction following ACALL. No flags are affected.

Example: Initially SP equals 07H. The label "SUBRTN" is at program memory location 0345 H. After executing the instruction,

ACALL  SUBRTN

at location 0123H, SP will contain 09H, internal RAM locations 08H and 09H will contain 25H and 01H, respectively, and the PC will contain 0345H.

Bytes: 2
Cycles: 2

Encoding: 

| a10 a9 a8 | 0 0 0 1 | a7 a6 a5 a4 | a3 a2 a1 a0 |

Operation:

ACALL
(PC) ← (PC) + 2
(SP) ← (SP) + 1
(SP) ← (PC7-0)
(SP) ← (SP) + 1
(SP) ← (PC15-8)
(PC10-6) ← page address
### ADD A, <src-byte>

**Function:** Add

**Description:** ADD adds the byte variable indicated to the Accumulator, leaving the result in the Accumulator. The carry and auxiliary-carry flags are set, respectively, if there is a carry-out from bit 7 or bit 3, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occurred.

OV is set if there is a carry-out of bit 6 but not out of bit 7, or a carry-out of bit 7 but not bit 6; otherwise OV is cleared. When adding signed integers, OV indicates a negative number produced as the sum of two positive operands, or a positive sum from two negative operands.

Four source operand addressing modes are allowed: register, direct, register-indirect, or immediate.

**Example:**
The Accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B). The instruction, `ADD A,R0`
will leave 6DH (01101101B) in the Accumulator with the AC flag cleared and both the Carry flag and OV set to 1.

### ADD A, Rn

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Encoding:**
```
0 0 1 0 1 r r r
```

**Operation:**
```
ADD (A) ← (A) + (Rn)
```

### ADD A, direct

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

**Encoding:**
```
0 0 1 0 0 1 0 1
```

**Operation:**
```
ADD (A) ← (A) + (direct)
```

### ADD A, @Ri

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Encoding:**
```
0 0 1 0 0 1 1 i
```

**Operation:**
```
ADD (A) ← (A) + ([Ri])
```

### ADD A, #data

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

**Encoding:**
```
0 0 1 0 0 1 0 0
```

**Operation:**
```
ADD (A) ← (A) + #data
```
ADDC A,<src-byte>

Function: Add with Carry
Description: ADDC simultaneously adds the byte variable indicated, the carry flag and the Accumulator contents, leaving the result in the Accumulator. The carry and auxiliary-carry flags are set, respectively, if there is a carry-out from bit 7 or bit 3, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occurred.
OV is set if there is a carry-out of bit 6 but not out of bit 7, or a carry-out of bit 7 but not out of bit 6; otherwise OV is cleared. When adding signed integers, OV indicates a negative number produced as the sum of two positive operands, or a positive sum from two negative operands.
Four source operand addressing modes are allowed: register, direct, register-indirect, or immediate.
Example: The Accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B) with the carry flag set. The instruction,
  ADDC A,R0
will leave 6EH (01101110B) in the Accumulator with AC cleared and both the Carry flag and OV set to 1.

ADDC A,Rn
Bytes: 1
Cycles: 1
Encoding:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>1</th>
<th>r</th>
<th>r</th>
</tr>
</thead>
</table>
Operation:
ADDC
(A) ← (A) + (C) + (Rn)

ADDC A,direct
Bytes: 2
Cycles: 1
Encoding:

|    |    |    | 0 | 1 | 0 | direct address

Operation:
ADDC
(A) ← (A) + (C) + (direct)

ADDC A,@Ri
Bytes: 1
Cycles: 1
Encoding:

|    |    |    | 0 | 1 | 1 | i |

Operation:
ADDC
(A) ← (A) + (C) + ((Ri))

ADDC A,#data
Bytes: 2
Cycles: 1
Encoding:

|    |    |    | 0 | 1 | 0 | 0 | immediate data

Operation:
ADDC
(A) ← (A) + (C) + #data
AJMP  

Function: Absolute Jump

Description: AJMP transfers program execution to the indicated address, which is formed at run-time by concatenating the high-order five bits of the PC (after incrementing the PC twice), opcode bits 7-5, and the second byte of the instruction. The destination must therefore be within the same 2k block of program memory as the first byte of the instruction following AJMP.

Example: The label “JMPADR” is at program memory location 0123H. The instruction, AJMP  JMPADR is at location 0345H and will load the PC with 0123H.

Bytes: 2
Cycles: 2

Encoding:

<table>
<thead>
<tr>
<th>a10 a9 a8</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>a7 a6 a5 a4</td>
<td>a3 a2 a1 a0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Operation:

AJMP

(\(PC\) \(\rightarrow\) (PC) + 2

(\(PC_{10:0}\) \(\leftarrow\) page address

ANL  \(<\text{dest-byte}>,<\text{src-byte}>\>

Function: Logical-AND for byte variables

Description: ANL performs the bitwise logical-AND operation between the variables indicated and stores the results in the destination variable. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the Accumulator or immediate data.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins.

Example: If the Accumulator holds 0C3H (11000011B) and register 0 holds 55H (01010101B) then the instruction, ANL  A,R0 will leave 41H (01000001B) in the Accumulator.

When the destination is a directly addressed byte, this instruction will clear combinations of bits in any RAM location or hardware register. The mask byte determining the pattern of bits to be cleared would either be a constant contained in the instruction or a value computed in the Accumulator at run-time. The instruction, ANL  P1,#01110011B will clear bits 7, 3, and 2 of output port 1.
ANL A,Rn
Bytes: 1
Cycles: 1
Encoding: 0 1 0 1 1 r r r
Operation: ANL (A) ← (A) ∧ (Rn)

ANL A,direct
Bytes: 2
Cycles: 1
Encoding: 0 1 0 1 0 1 0 1
direct address
Operation: ANL (A) ← (A) ∧ (direct)

ANL A,Ri
Bytes: 1
Cycles: 1
Encoding: 0 1 0 1 0 1 1 i
Operation: ANL (A) ← (A) ∧ (Ri)

ANL A,#data
Bytes: 2
Cycles: 1
Encoding: 0 1 0 1 0 1 0 0
immediate data
Operation: ANL (A) ← (A) ∧ #data

ANL direct,A
Bytes: 2
Cycles: 1
Encoding: 0 1 0 1 0 0 1 0
direct address
Operation: ANL (A) ← (direct) ∧ (A)

ANL direct,#data
Bytes: 3
Cycles: 2
Encoding: 0 1 0 1 0 0 1 1
direct address immediate data
Operation: ANL (direct) ← (direct) ∧ #data
ANL  C,<src-bit>

Function: Logical-AND for bit variables

Description: If the Boolean value of the source bit is a logical 0 then clear the carry flag; otherwise leave the carry flag in its current state. A slash (\slash) preceding the operand in the assembly language indicates that the logical complement of the addressed bit is used as the source value, but the source bit itself is not affected. No other flags are affected.

Only direct addressing is allowed for the source operand.

Example: Set the carry flag if, and only if, P1.0 = 1, ACC.7 = 1, and OV = 0:
MOV  C,P1.0 ;LOAD CARRY WITH INPUT PIN STATE
ANL  C,ACC.7;AND CARRY WITH ACCUM. BIT 7
ANL  C,/OV ;AND WITH INVERSE OF OVERFLOW FLAG

ANL  C,bit

Bytes: 2
Cycles: 2

Encoding: 1 0 0 0 0 0 1 0
Operation: ANL
(C) ← (C) ∧ (bit)

ANL  C,/bit

Bytes: 2
Cycles: 2

Encoding: 1 0 1 1 0 0 0 0
Operation: ANL
(C) ← (C) ∧ 1 (bit)
**CJNE <dest-byte>,<src-byte>,rel**

**Function:** Compare and Jump if Not Equal

**Description:** CJNE compares the magnitudes of the first two operands, and branches if their values are not equal. The branch destination is computed by adding the signed relative-displacement in the last instruction byte to the PC, after incrementing the PC to the start of the next instruction. The carry flag is set if the unsigned integer value of <dest-byte> is less than the unsigned integer value of <src-byte>; otherwise, the carry is cleared. Neither operand is affected.

The first two operands allow four addressing mode combinations: the Accumulator may be compared with any directly addressed byte or immediate data, and any indirect RAM location or working register can be compared with an immediate constant.

**Example:** The Accumulator contains 34H. Register 7 contains 56H. The first instruction in the sequence,

```
CJNE R7,#60H,NOT_EQ
; NOT_EQ JC REQ_LOW ; R7 = 60H.
; .... ; IF R7 < 60H.
```

sets the carry flag and branches to the instruction at label NOT_EQ. By testing the carry flag, this instruction determines whether R7 is greater or less than 60H.

If the data being presented to Port 1 is also 34H, then the instruction,

```
WAIT: CJNE A,P1,WAIT
```

clears the carry flag and continues with the next instruction in sequence, since the Accumulator does equal the data read from P1. (If some other value was being input on P1, the program will loop at this point until the P1 data changes to 34H.)

**CJNE A,direct,rel**

**Bytes:** 3

**Cycles:** 2

**Encoding:**

| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | direct address | rel. address |

**Operation:**

\[
\begin{align*}
(PC) & \leftarrow (PC) + 3 \\
\text{IF } (A) & < (\text{direct}) \\
\text{THEN} & \\
\quad & (PC) \leftarrow (PC) + \text{relative offset} \\
\text{IF } (A) & < (\text{direct}) \\
\text{THEN}. & \\
\quad & (C) \leftarrow 1 \\
\text{ELSE} & \\
\quad & (C) \leftarrow 0
\end{align*}
\]
CJNE A,#data,rel
Bytes: 3
Cycles: 2
Encoding: 10110100 immediate data rel. address
Operation: (PC) ← (PC) + 3
   IF (A) <> data
   THEN (PC) ← (PC) + relative offset
   IF (A) < data
   THEN (C) ← 1
   ELSE (C) ← 0

CJNE Rn,#data,rel
Bytes: 3
Cycles: 2
Encoding: 1011rrrr immediate data rel. address
Operation: (PC) ← (PC) + 3
   IF (Rn) <> data
   THEN (PC) ← (PC) + relative offset
   IF (Rn) < data
   THEN (C) ← 1
   ELSE (C) ← 0

CJNE @Ri,#data,rel
Bytes: 3
Cycles: 2
Encoding: 1011011i immediate data rel. address
Operation: (PC) ← (PC) + 3
   IF ((Ri)) <> data
   THEN (PC) ← (PC) + relative offset
   IF ((Ri)) < data
   THEN (C) ← 1
   ELSE (C) ← 0
CLR A

Function: Clear Accumulator
Description: The Accumulator is cleared (all bits reset to zero). No flags are affected.
Example: The Accumulator contains 5CH (01011100B). The instruction,
CLR A
will leave the Accumulator set to 00H (00000000B).
Bytes: 1
Cycles: 1
Encoding: 1 1 1 0 0 1 0 0
Operation: CLR (A) ← 0

CLR bit

Function: Clear bit
Description: The indicated bit is cleared (reset to zero). No other flags are affected. CLR can operate on the carry flag or any directly addressable bit.
Example: Port 1 has previously been written with 5DH (01011101B). The instruction,
CLR P1.2
will leave the port set to 59H (01011001B).

CLR C

Bytes: 1
Cycles: 1
Encoding: 1 1 0 0 0 0 1 1
Operation: CLR (C) ← 0

CLR bit

Bytes: 2
Cycles: 1
Encoding: 1 1 0 0 0 0 1 0 bit address
Operation: CLR (bit) ← 0
CPL A

Function: Complement Accumulator
Description: Each bit of the Accumulator is logically complemented (one's complement). Bits which previously contained a one are changed to a zero and vice-versa. No flags are affected.
Example: The Accumulator contains 5CH (01011100B). The instruction,
        CPL A
        will leave the Accumulator set to 0A3H (10100011B).
Bytes: 1
Cycles: 1
Encoding: 1 1 1 1 0 1 0 0
Operation: CPL
           (A) ← ↑ (A)

CPL bit

Function: Complement bit
Description: The bit variable specified is complemented. A bit which had been a one is changed to zero and vice-versa. No other flags are affected. CLR can operate on the carry or any directly addressable bit.
Note: When this instruction is used to modify an output pin, the value used as the original data will be read from the output data latch, not the input pin.
Example: Port 1 has previously been written with 5DH (01011101B). The instruction sequence,
        CPL P1.1
        CPL P1.2
        will leave the port set to 5BH (01011011B).

CPL C

Bytes: 1
Cycles: 1
Encoding: 1 0 1 1 0 0 1 1
Operation: CPL
           (C) ← ↑ (C)

CPL bit

Bytes: 2
Cycles: 1
Encoding: 1 0 1 1 0 0 1 0
          bit address
Operation: CPL
           (bit) ← ↑ (bit)
Function: Decimal-adjust Accumulator for Addition

Description: DA A adjusts the eight-bit value in the Accumulator resulting from the earlier addition of two variable (each in packed-BCD format), producing two four-bit digits. Any ADD or ADDC instruction may have been used to perform the addition.

If Accumulator bits 3-0 are greater than nine (xxx1010-xxx1111), or if the AC flag is one, six is added to the Accumulator, producing the proper BCD digit in the low-order nibble. This internal addition would set the carry flag if a carry-out of the low-order four-bit field propagated through all high-order bits, but it would not clear the carry flag otherwise.

If the carry flag is now set, or if the four high-order bits now exceed nine (1010xxx-111xxx), these high-order bits are incremented by six, producing the proper BCD digit in the high-order nibble. Again, this would set the carry flag if there was a carry-out of the high-order bits, but wouldn’t clear the carry. The carry flag thus indicates if the sum of the original two BCD variables is greater than 100, allowing multiple precision decimal addition. OV is not affected.

All of this occurs during the one instruction cycle. Essentially, this instruction performs the decimal conversion by adding 00H, 06H, 60H, or 66H to the Accumulator, depending on initial Accumulator and PSW conditions.

Note: DA A cannot simply convert a hexadecimal number in the Accumulator to BCD notation, nor does DA A apply to decimal subtraction.

Example: The Accumulator holds the value 56H (01010110B) representing the packed BCD digits of the decimal number 56. Register 3 contains the value 67H (01100111B) representing the packed BCD digits of the decimal number 67. The carry flag is set. The instruction sequence, ADDC A,R3
DA A will first perform a standard two's-complement binary addition, resulting in the value 0BEH (10111110B) in the Accumulator. The carry and auxiliary carry flags will be cleared.

The Decimal Adjust instruction will then alter the Accumulator to the value 24H (00100100B), indicating the packed BCD digits of the decimal number 24, the low-order two digits of the decimal sum of 56, 67, and the carry-in. The carry flag will be set by the Decimal Adjust instruction, indicating that a decimal overflow occurred. The true sum 56, 67, and 1 is 124.

BCD variables can be incremented or decremented by adding 01H or 99H. If the Accumulator initially holds 30H (representing the digits of 30 decimal), the the instruction sequence, ADD A,#99H
DA A will leave the carry set and 29H in the Accumulator, since 30 + 99 = 129. The low-order byte of the sum can be interpreted to mean 30 – 1 = 29.

Bytes: 1
Cycles: 1

Encoding: 1 1 0 1 0 1 0 0

Operation: DA
–contents of Accumulator are BCD
IF [(A3:0) > 9] V [(AC) = 1]
THEN (A3:0) ← (A3:0) + 6
AND
IF [(A7:4) > 9] V [(C) = 1]
THEN (A7:4) ← (A7:4) + 6
## DEC byte

**Function:** Decrement

**Description:** The variable indicated is decremented by 1. An original value of 00H will underflow to 0FFH. No flags are affected. Four operand addressing modes are allowed: accumulator, register, direct, or register-indirect.

*Note:* When this instruction is used to modify an output port, the value used as the original data will be read from the output data latch, not the input pin.

**Example:** Register 0 contains 7FH (01111111B). Internal RAM locations 7EH and 7FH contain 00H and 40H, respectively. The instruction sequence,

```
DEC @R0
DEC R0
DEC @R0
```

will leave register 0 set to 7EH and internal RAM locations 7EH and 7FH set to 0FFH and 3FH.

### DEC A

<table>
<thead>
<tr>
<th>Bytes:</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycles:</td>
<td>1</td>
</tr>
<tr>
<td>Encoding:</td>
<td>0 0 0 1 0 1 0 0</td>
</tr>
<tr>
<td>Operation:</td>
<td>DEC (A) ← (A) − 1</td>
</tr>
</tbody>
</table>

### DEC Rn

<table>
<thead>
<tr>
<th>Bytes:</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycles:</td>
<td>1</td>
</tr>
<tr>
<td>Encoding:</td>
<td>0 0 0 1 1 r r r</td>
</tr>
<tr>
<td>Operation:</td>
<td>DEC (Rn) ← (Rn) − 1</td>
</tr>
</tbody>
</table>

### DEC direct

<table>
<thead>
<tr>
<th>Bytes:</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycles:</td>
<td>1</td>
</tr>
<tr>
<td>Encoding:</td>
<td>0 0 0 1 0 1 0 1</td>
</tr>
<tr>
<td>Operation:</td>
<td>DEC (direct) ← (direct) − 1</td>
</tr>
</tbody>
</table>

### DEC @Ri

<table>
<thead>
<tr>
<th>Bytes:</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycles:</td>
<td>1</td>
</tr>
<tr>
<td>Encoding:</td>
<td>0 0 0 1 0 1 1 i</td>
</tr>
<tr>
<td>Operation:</td>
<td>DEC ((Ri)) ← ((Ri)) − 1</td>
</tr>
</tbody>
</table>
DIV AB

Function: Divide

Description: DIV AB divides the unsigned eight-bit integer in the Accumulator by the unsigned eight-bit integer in register B.

The Accumulator receives the integer part of the quotient; register B receives the integer remainder. The carry and OV flags will be cleared.

Exception: if B had originally contained 00H, the values returned in the Accumulator and B-register will be undefined and the overflow flag will be set. The carry flag is cleared in any case.

Example: The Accumulator contains 251 (0FBH or 11111011B) and B contains 18 (12H or 00010010B). The instruction,
DIV AB
will leave 13 in the Accumulator (0DH or 00001101B) and the value 17 (11H or 00010001B) in B, since 251 = (13 x 18) + 17. Carry and OV will both be cleared.

Bytes: 1
Cycles: 4

Encoding: 1 0 0 0 0 1 0 0

Operation: DIV
(A)15-8 ← (A)/(B)
(B)7-0
**DJNZ <byte>,<rel-addr>**

**Function:** Decrement and Jump if Not Zero

**Description:** DJNZ decrements the location indicated by 1, and branches to the address indicated by the second operand if the resulting value is not zero. An original value of 00H will underflow to 0FFH. No flags are affected. The branch destination would be computed by adding the signed relative-displacement value in the last instruction byte to the PC, after incrementing the PC to the first byte of the following instruction.

The location decremented may be a register or directly addressed byte.

**Note:** When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins.

**Example:** Internal RAM locations 40H, 50H, and 60H contain the values 01H, 70H, and 15H, respectively. The instruction sequence,

```
DJNZ 40H, LABEL_1
DJNZ 50H, LABEL_2
DJNZ 60H, LABEL_3
```

will cause a jump to the instruction at LABEL_2 with the values 00H, 6FH, and 15H in the three RAM locations. The first jump was not taken because the result was zero.

This instruction provides a simple way of executing a program loop a given number of times, or for adding a moderate time delay (from 2 to 512 machine cycles) with a single instruction. The instruction sequence,

```
MOV R2,#8
TOGGLE: CPL P1.7
DJNZ R2,TOGGLE
```

will toggle P1.7 eight times, causing four output pulses to appear at bit 7 of output Port 1. Each pulse will last three machine cycles, two for DJNZ and one to alter the pin.

**DJNZ Rn,rel**

**Bytes:** 2

**Cycles:** 2

**Encoding:**

```
1 1 0 1 1 r r r rel. address
```

**Operation:**

```
DJNZ
(PC) ← (PC) + 2
(Rn) ← (Rn) - 1
IF (Rn) > 0 or (Rn) < 0
THEN
   (PC) ← (PC) + rel
```

**DJNZ direct,rel**

**Bytes:** 3

**Cycles:** 2

**Encoding:**

```
1 1 0 1 0 1 0 1 direct data rel. address
```

**Operation:**

```
DJNZ
(PC) ← (PC) + 2
(direct) ← (direct) - 1
IF (direct) > 0 or (direct) < 0
THEN
   (PC) ← (PC) + rel
```
INC <byte>

Function: Increment

Description: INC increments the indicated variable by 1. An original value of 0FFH will overflow to 00H. No flags are affected. Three addressing modes are allowed: register, direct, or register-indirect.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins.

Example: Register 0 contains 7EH (0111110B). Internal RAM locations 7EH and 7FH contain 0FFH and 40H, respectively. The instruction sequence,

INC @R0
INC R0
INC @R0

will leave register 0 set to 7FH and internal RAM locations 7EH and 7FH holding (respectively) 00H and 41H.

INC A

Bytes: 1
Cycles: 1

Encoding: 0 0 0 0 0 1 0 0
Operation: INC (A) ← (A) + 1

INC Rn

Bytes: 1
Cycles: 1

Encoding: 0 0 0 0 1 r r r
Operation: INC (Rn) ← (Rn) + 1

INC direct

Bytes: 2
Cycles: 1

Encoding: 0 0 0 0 0 1 0 1  direct address
Operation: INC (direct) ← (direct) + 1

INC @Ri

Bytes: 1
Cycles: 1

Encoding: 0 0 0 0 0 1 1 i
Operation: INC ((Ri)) ← ((Ri)) + 1
INC DPTR

Function: Increment Data Pointer

Description: Increment the 16-bit data pointer by 1. A 16-bit increment (modulo $2^{16}$) is performed; an overflow of the low-order byte of the data pointer (DPL) from 0FFH to 00H will increment the high-order byte (DPH). No flags are affected.

This is the only 16-bit register which can be incremented.

Example: Registers DPH and DPL contain 12H and 0FEH, respectively. The instruction sequence,

```
INC DPTR
INC DPTR
INC DPTR
```

will change DPH and DPL to 13H and 01H.

Bytes: 1

Cycles: 2

Encoding: 1 0 1 0 0 0 1 1

Operation: INC (DPTR) ← (DPTR) + 1

---

JB bit,rel

Function: Jump if Bit set

Description: If the indicated bit is a one, jump to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. The bit tested is not modified. No flags are affected.

Example: The data present at input port 1 is 11001010B. The Accumulator holds 56 (01010110B). The instruction sequence,

```
JB P1.2, LABEL1
JB ACC.2, LABEL2
```

will cause program execution to branch to the instruction at label LABEL2.

Bytes: 3

Cycles: 2

Encoding: 0 0 1 0 0 0 0 0

Operation: JB (PC) ← (PC) + 3

IF (bit) = 1

THEN

```
(PC) ← (PC) + rel
```
JBC  bit,rel

**Function:** Jump if Bit is set and Clear bit

**Description:** If the indicated bit is a one, branch to the address indicated; otherwise proceed with the next instruction. *The bit will not be cleared if it is already a zero.* The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. No flags are affected.

**Note:** When this instruction is used to test an output pin, the value used as the original data will read from the output data latch, not the input pin.

**Example:** The Accumulator holds 56H (01010110B). The instruction sequence,

```
JBC ACC.3, LABEL1
JBC ACC.2, LABEL2
```

will cause program execution to continue at the instruction identified by the LABEL2, with the Accumulator modified to 52H (01010010B).

**Bytes:** 3

**Cycles:** 2

**Encoding:**

```
0 0 0 1 0 0 0
```

**Operation:**

```
JBC
(PC) ← (PC) + 3
IF (bit) = 1
THEN
(bit) ← 0
(PC) ← (PC) + rel
```

JC  rel

**Function:** Jump if Carry is set

**Description:** If the carry flag is set, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. No flags are affected.

**Example:** The carry flag is cleared. The instruction sequence,

```
JC LABEL1
CPL C
JC LABEL2
```

will set the carry and cause program execution to continue at the instruction identified by the label LABEL2.

**Bytes:** 2

**Cycles:** 2

**Encoding:**

```
0 1 0 0 0 0 0 0
```

**Operation:**

```
JC
(PC) ← (PC) + 2
IF (C) = 1
THEN
(PC) ← (PC) + rel
```
JMP  @A+DPTR

Function: Jump indirect

Description: Add the eight-bit unsigned contents of the Accumulator with the sixteen-bit data pointer, and load the resulting sum to the program counter. This will be the address for subsequent instruction fetches. Sixteen-bit addition is performed (modulo $2^{16}$): a carry-out from the low-order eight bits propagates through the higher-order bits. Neither the Accumulator nor the Data Pointer is altered. No flags are affected.

Example: An even number from 0 to 6 is in the Accumulator. The following sequence of instructions will branch to one of four AJMP instructions in a jump table starting at JMP_TBL:

```
  MOV DPTR,#JMP_TBL
  JMP @A+DPTR
  JMP_TBL: AJMP LABEL0
            AJMP LABEL1
            AJMP LABEL2
            AJMP LABEL3
```

If the Accumulator equals 04H when starting this sequence, execution will jump to label LABEL2. Remember that AJMP is a two-byte instruction, so the jump instructions start at every other address.

Bytes: 1
Cycles: 2
Encoding: 01110011
Operation: JMP (PC) ← (A) + (DPTR)

JNB  bit,rel

Function: Jump if Bit Not set

Description: If the indicated bit is a zero, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. The bit tested is not modified. No flags are affected.

Example: The data present at input port 1 is 11001010B. The Accumulator holds 56H (01010110B). The instruction sequence,

```
  JNB P1.3, LABEL1
  JNB ACC.3, LABEL2
```

will cause program execution to continue at the instruction at label LABEL2.

Bytes: 3
Cycles: 2
Encoding: 00110000 bit address rel. address
Operation: JNB (PC) ← (PC) + 3
IF (bit) = 0
THEN
(PC) ← (PC) + rel
JNC rel

Function: Jump if Carry Not set

Description: If the carry flag is a zero, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice to point to the next instruction. The carry flag is not modified.

Example: The carry flag is set. The instruction sequence,
JNC LABEL1
CPL C
JNC LABEL2

will clear the carry and cause program execution to continue at the instruction identified by the label LABEL2.

Bytes: 2
Cycles: 2

Encoding:

| 0 | 1 | 0 | 1 | 0 | 0 | 0 | rel. address |

Operation:

JNC
(PC) ← (PC) + 2
IF (C) = 0
THEN
(PC) ← (PC) + rel

JNZ rel

Function: Jump if Accumulator Not Zero

Description: If any bit of the Accumulator is a one, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The Accumulator is not modified. No flags are affected.

Example: The Accumulator originally holds 00H. The instruction sequence,
JNZ LABEL1
INC A
JNZ LABEL2

will set the Accumulator to 01H and continue at label LABEL2.

Bytes: 2
Cycles: 2

Encoding:

| 0 | 1 | 1 | 1 | 0 | 0 | 0 | rel. address |

Operation:

JNZ
(PC) ← (PC) + 2
IF A ≠ 0
THEN (PC) ← (PC) + rel
80C51 Family

80C51 family programmer’s guide and instruction set

JZ rel

Function: Jump if Accumulator Zero

Description: If all bits of the Accumulator are zero, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The Accumulator is not modified. No flags are affected.

Example: The Accumulator originally holds 01H. The instruction sequence,

```
JZ LABEL1
DEC A
JZ LABEL2
```

will change the Accumulator to 00H and cause program execution to continue at the instruction identified by the label LABEL2.

Bytes: 2
Cycles: 2
Encoding: `0 1 1 0 0 0 0 0` rel. address
Operation: JZ

(\text{PC}) \leftarrow (\text{PC}) + 2
\text{IF } A = 0
\text{THEN } (\text{PC}) \leftarrow (\text{PC}) + \text{rel}

LCALL addr16

Function: Long Call

Description: LCALL calls a subroutine located at the indicated address. The instruction adds three to the program counter to generate the address of the next instruction and then pushes the 16-bit result onto the stack (low byte first), incrementing the Stack Pointer by two. The high-order and low-order bytes of the PC are then loaded, respectively, with the second and third bytes of the LCALL instruction. Program execution continues with the instruction at this address. The subroutine may therefore begin anywhere in the full 64K-byte program memory address space. No flags are affected.

Example: Initially the Stack Pointer equals 07H. The label “SUBRTN” is assigned to program memory location 1234H. After executing the instruction,

```
LCALL SUBRTN
```

at location 0123H, the Stack Pointer will contain 09H, internal RAM locations 08H and 09H will contain 26H and 01H, and the PC will contain 1235H.

Bytes: 3
Cycles: 2
Encoding: `0 0 0 1 0 0 1 0` addr15-addr8 addr7-addr0
Operation: LCALL

(\text{PC}) \leftarrow (\text{PC}) + 3
(\text{SP}) \leftarrow (\text{SP}) + 1
((\text{SP})) \leftarrow (\text{PC}_{7:0})
(\text{SP}) \leftarrow (\text{SP}) + 1
((\text{SP})) \leftarrow (\text{PC}_{15:8})
(\text{PC}) \leftarrow \text{addr}_{15:0}
LJMP  addr16  (Implemented in 87C751 and 87C752 for in-circuit emulation only.)

Function: Long Jump
Description: LJMP causes an unconditional branch to the indicated address, by loading the high-order and low-order bytes of the PC (respectively) with the second and third instruction bytes. The destination may therefore be anywhere in the full 64k program memory address space. No flags are affected.
Example: The label "JMPADR" is assigned to the instruction at program memory location 1234H. The instruction, LJMP  JMPADR
at location 0123H will load the program counter with 1234H.
Bytes: 3
Cycles: 2
Encoding:   0 0 0 0 0 0 1 0   addr15-addr8   addr7-addr0
Operation: LJMP
           (PC) ← addr15-0

MOV  <dest-byte>,<src-byte>

Function: Move byte variable
Description: The byte variable indicated by the second operand is copied into the location specified by the first operand. The source byte is not affected. No other register or flag is affected.
This is by far the most flexible operation. Fifteen combinations of source and destination addressing modes are allowed.
Example: Internal RAM location 30H holds 40H. The value of RAM location 40H is 10H. The data present at input port 1 is 11001010B (0CAH). The instruction sequence,
MOV  R0,#30H ;R0 ← 30H
MOV  A,@R0   ;A ← 40H
MOV  R1,A    ;R1 ← 40H
MOV  B,@R1   ;B ← 10H
MOV  @R1,P1  ;RAM (40H) ← 0CAH
MOV  P2,P1   ;P2 ← #0CAH
leaves the value 30H in register 0, 40H in both the Accumulator and register 1, 10H in register B, and 0CAH (11001010B) both in RAM location 40H and output on port 2.

MOV  A,Rn
Bytes: 1
Cycles: 1
Encoding:   1 1 1 0 1 r r r
Operation: MOV
           (A) ← (Rn)
*MOV A,direct
Bytes: 2
Cycles: 1
Encoding: 1 1 1 0 0 1 0 1
direct address
Operation: MOV (A) ← (direct)

MOV A,@Ri
Bytes: 1
Cycles: 1
Encoding: 1 1 1 0 0 1 1 i
Operation: MOV (A) ← ((Ri))

MOV A,#data
Bytes: 2
Cycles: 1
Encoding: 0 1 1 1 0 1 0 0
immediate data
Operation: MOV (A) ← #data

MOV Rn,A
Bytes: 1
Cycles: 1
Encoding: 1 1 1 1 1 1 1 1
Operation: MOV (Rn) ← (A)

MOV Rn,direct
Bytes: 2
Cycles: 2
Encoding: 1 0 1 0 1 1 1 1
direct address
Operation: MOV (Rn) ← (direct)

MOV Rn,#data
Bytes: 2
Cycles: 1
Encoding: 0 1 1 1 1 1 1 r r r
immediate data
Operation: MOV (Rn) ← #data

*MOV A,ACC is not a valid instruction.
MOV direct,A
Bytes: 2
Cycles: 1
Encoding: 1111 0101
Operation: MOV (direct) ← (A)

MOV direct,Rn
Bytes: 2
Cycles: 2
Encoding: 1000 1rrr
direct address
Operation: MOV (direct) ← (Rn)

MOV direct,direct
Bytes: 3
Cycles: 2
Encoding: 1000 0101
dir. addr. (src) dir. addr. (dest)
Operation: MOV (direct) ← (direct)

MOV direct,@Ri
Bytes: 2
Cycles: 2
Encoding: 1000 011i
direct address
Operation: MOV (direct) ← (@(Ri))

MOV direct,#data
Bytes: 3
Cycles: 2
Encoding: 0111 0101
direct address immediate data
Operation: MOV (direct) ← #data

MOV @Ri,A
Bytes: 1
Cycles: 1
Encoding: 1111 011i
Operation: MOV (@(Ri)) ← (A)
MOV @Ri,direct

Bytes: 2
Cycles: 2

Encoding: 1 0 1 0 0 1 1 i
Operation: MOV ((Ri)) ← (direct)

MOV @Ri,#data

Bytes: 2
Cycles: 1

Encoding: 0 1 1 1 0 1 1 i
Operation: MOV ((Ri)) ← #data

MOV <dest-bit>,<src-bit>

Function: Move bit data
Description: The Boolean variable indicated by the second operand is copied into the location specified by the first operand. One of the operands must be the carry flag; the other may be any directly addressable bit. No other register or flag is affected.
Example: The carry flag is originally set. The data present at input Port 3 is 11000101B. The data previously written to output Port 1 is 35H (00110101B). The instruction sequence,

<table>
<thead>
<tr>
<th>MOV</th>
<th>P1.3,C</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOV</td>
<td>C,P3.3</td>
</tr>
<tr>
<td>MOV</td>
<td>P1.2,C</td>
</tr>
</tbody>
</table>

will leave the carry cleared and change Port 1 to 39H (00111001B).

MOV C,bit

Bytes: 2
Cycles: 1

Encoding: 1 0 1 0 0 0 1 0
Operation: MOV (C) ← (bit)

MOV bit,C

Bytes: 2
Cycles: 2

Encoding: 1 0 0 1 0 0 1 0
Operation: MOV (bit) ← (C)
MOV DPTR,#data16

Function: Load Data Pointer with a 16-bit constant

Description: The Data Pointer is loaded with the 16-bit constant indicated. The 16-bit constant is loaded into the second and third bytes of the instruction. The second byte (DPH) is the high-order byte, while the third byte (DPL) holds the low-order byte. No flags are affected.

This is the only instruction which moves 16 bits of data at once.

Example: The instruction,
MOV DPTR,#1234H

will load the value 1234H into the Data Pointer: DPH will hold 12H and DPL will hold 34H.

Bytes: 3
Cycles: 2

Encoding: 1 0 0 1 0 0 0 0

Operation: MOV
(DPTR) ← (#data16,0)
DPL ← #data16,3 DPL ← #data7,0

MOVC A,@A+<base-reg>

Function: Move Code byte

Description: The MOVC instructions load the Accumulator with a code byte, or constant from program memory. The address of the byte fetched is the sum of the original unsigned eight-bit Accumulator contents and the contents of a sixteen-bit base register, which may be either the Data Pointer or the PC. In the latter case, the PC is incremented to the address of the following instruction before being added with the Accumulator; otherwise the base register is not altered. Sixteen-bit addition is performed so a carry-out from the low-order eight bits may propagate through higher-order bits. No flags are affected.

Example: A value between 0 and 3 is in the Accumulator. The following instructions will translate the value in the Accumulator to one of four values defined by the DB (define byte) directive:

REL PC:
INC A
MOVC A,@A+PC
RET
DB 66H
DB 77H
DB 88H
DB 99H

If the subroutine is called with the Accumulator equal to 01H, it will return with 77H in the Accumulator. The INC A before the MOVC instruction is needed to "get around" the RET instruction above the table. If several bytes of code separated the MOVC from the table, the corresponding number would be added to the Accumulator instead.

MOVC A,@A+DPTR

Bytes: 1
Cycles: 2

Encoding: 1 0 0 1 0 0 1 1

Operation: MOVC
(A) ← ((A) + (DPTR))
MOVC A, @A+PC

Bytes: 1
Cycles: 2
Encoding: 1 0 0 0 0 0 1 1
Operation:
(MC)
( )

MOVC <dest-byte>, <src-byte>  (Not implemented in the 8XC752 or 8XC752)

Function: Move External
Description: The MOVC instructions transfer data between the Accumulator and a byte of external data memory, hence the "X" appended to MOVC. There are two types of instructions, differing in whether they provide an eight-bit or sixteen-bit indirect address to the external data RAM.

In the first type, the contents of R0 or R1 in the current register bank provide an eight-bit address multiplexed with data on P0. Eight bits are sufficient for external I/O expansion decoding or for a relatively small RAM array. For somewhat larger arrays, port pins can be used to output higher-order address bits. These pins would be controlled by an output instruction preceding the MOVC.

In the second type of MOVC instruction, The Data Pointer generates a sixteen-bit address. P2 outputs the high-order address bits (the contents of DPH) while P0 multiplexes the low-order eight bits (DPL) with data. The P2 Special Function Register retains its previous contents while the P2 output buffers are emitting the contents of DPH. This form is faster and more efficient when accessing very large data arrays (up to 64k bytes), since no additional instructions are needed to set up the output ports.

It is possible in some situations to mix the two MOVC types. A large RAM array with its high-order address lines driven by P2 can be addressed via the Data Pointer, or with code to output high-order address bits to P2 followed by a MOVC instruction using R0 or R1.

Example: An external 256 byte RAM using multiplexed address/data lines is connected to the 8051 Port 0. Port 3 provides control lines for the external RAM. Ports 1 and 2 are used for normal I/O. Registers 0 and 1 contain 12H and 34H. Location 34H of the external RAM holds the value 56H. The instruction sequence,

MOVC A, @R1
MOVC @R0, A

copies the value 56H into both the Accumulator and external RAM location 12H.
MO VX @RI,A
Bytes: 1
Cycles: 2
Encoding: 1 1 1 1 0 0 1 i
Operation: MO VX
((RI)) ← (A)

MO VX @DPTR,A
Bytes: 1
Cycles: 2
Encoding: 1 1 1 1 0 0 0 0
Operation: MO VX
((DPTR)) ← (A)

MUL AB
Function: Multiply
Description: MUL AB multiplies the unsigned eight-bit integers in the Accumulator and register B. The low-order byte of the sixteen-bit product is left in the Accumulator, and the high-order byte in B. If the product is greater than 255 (0FFH) the overflow flag is set; otherwise it is cleared. The carry flag is always cleared.
Example: Originally the Accumulator holds the value 80 (50H). Register B holds the value 160 (0A0H). The instruction,
MUL AB
will give the product 12,800 (3200H), so B is changed to 32H (00110010B) and the Accumulator is cleared. The overflow flag is set, carry is cleared.
Bytes: 1
Cycles: 4
Encoding: 1 0 1 0 0 1 0 0
Operation: MUL
(A)7-0 ← (A) x (B)
(B)15-8
NOP

Function: No Operation

Description: Execution continues at the following instruction. Other than the PC, no registers or flags are affected.

Example: It is desired to produce a low-going output pulse on bit 7 of Port 2 lasting exactly 5 cycles. A simple
SETB/CLR sequence would generate a one-cycle pulse, so four additional cycles must be inserted. This
may be done (assuming are enabled) with the instruction sequence,

```
CLR    P2.7
NOP
NOP
NOP
NOP
SETB P2.7
```

Bytes: 1
Cycles: 1

Encoding: 0 0 0 0 0 0 0 0

Operation: NOP
(\( \text{PC} \leftarrow (\text{PC}) + 1 \))

ORL <dest-byte>,<src-byte>

Function: Logical-OR for byte variables

Description: ORL performs the bitwise logical-OR operation between the indicated variables, storing the results in the
destination byte. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the Accumulator, the
source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct
address, the source can be the Accumulator or immediate data.

Note: When this instruction is used to modify an output port, the value used as the original port data will
be read from the output data latch, not the input pins.

Example: If the Accumulator holds 0C3H (11000011B) and R0 holds 55H (01010101B) then the instruction,

```
ORL A,R0
```

will leave the Accumulator holding the value 0D7H (11010111B).

When the destination is a directly addressed byte, the instruction can set combinations of bits in any RAM
location or hardware register. The pattern of bits to be set is determined by a mask byte, which may be
either a constant data value in the instruction or a variable computed in the Accumulator at run-time. The
instruction,

```
ORL P1,#00110010B
```

will set bits 5, 4, and 1 of output Port 1.

ORL A,Rn

Bytes: 1
Cycles: 1

Encoding: 0 1 0 0 1 r r r

Operation: ORL
(A) ← (A) ⊕ (Rn)
ORL A, direct
Bytes: 2
Cycles: 1
Encoding: 0100 0101
Operation: ORL
(A) ← (A) ∨ (direct)

ORL A, @Ri
Bytes: 1
Cycles: 1
Encoding: 0100 011i
Operation: ORL
(A) ← (A) ∨ ((Ri))

ORL A, #data
Bytes: 2
Cycles: 1
Encoding: 0100 0100
Operation: ORL
(A) ← (A) ∨ #data

ORL direct, A
Bytes: 2
Cycles: 1
Encoding: 0100 0010
Operation: ORL
(direct) ← (direct) ∨ (A)

ORL direct, #data
Bytes: 3
Cycles: 2
Encoding: 0100 0011
Operation: ORL
(direct) ← (direct) ∨ #data
ORL $\text{C},<$src-bit$>

Function: Logical-OR for bit variables

Description: Set the carry flag if the Boolean value is a logical 1; leave the carry in its current state otherwise. A slash (') preceding the operand in the assembly language indicates that the logical complement of the addressed bit is used as the source value, but the source bit itself is not affected. No other flags are affected.

Example: Set the carry flag if and only if P1.0 = 1, ACC.7 = 1, or OV = 0:

ORL C,P1.0 ;LOAD CARRY WITH INPUT PIN P10
ORL C,ACC.7 ;OR CARRY WITH THE ACC. BIT 7
ORL C,OV ;OR CARRY WITH THE INVERSE OF OV.

ORL C,bit

Bytes: 2
Cycles: 2

Encoding: \begin{tabular}{c c c c}
0 & 1 & 1 & 1 \hline
0 & 0 & 1 & 0
\end{tabular} [bit address]

Operation: ORL
(C) \leftarrow (C) \lor (\text{bit})

ORL C,bit

Bytes: 2
Cycles: 2

Encoding: \begin{tabular}{c c c c c c c c c c c c c c c}
1 & 0 & 1 & 0 \hline
0 & 0 & 0 & 0
\end{tabular} [bit address]

Operation: ORL
(C) \leftarrow (C) \lor (\text{bit})
### POP direct

**Function:** Pop from stack

**Description:** The contents of the internal RAM location addressed by the Stack Pointer is read, and the Stack Pointer is decremented by one. The value read is then transferred to the directly addressed byte indicated. No flags are affected.

**Example:** The Stack Pointer originally contains the value 32H, and internal RAM locations 30H through 32H contain the values 20H, 23H, and 01H, respectively. The instruction sequence,

```
POP DPH
POP DPL
```

will leave the Stack Pointer equal to the value 30H and the Data Pointer set to 0123H. At this point the instruction,

```
POP SP
```

will leave the Stack Pointer set to 20H. Note that in this special case the Stack Pointer was decremented to 2FH before being loaded with the value popped (20H).

**Bytes:** 2

**Cycles:** 2

**Encoding:**

```
1 1 0 1 0 0 0 0
```

direct address

**Operation:**

```
POP (direct) ← ((SP))
(SP) ← (SP) − 1
```

### PUSH direct

**Function:** Push onto stack

**Description:** The Stack Pointer is incremented by one. The contents of the indicated variable is then copied into the internal RAM location addressed by the Stack Pointer. Otherwise no flags are affected.

**Example:** On entering an interrupt routine the Stack Pointer contains 09H. The Data Pointer holds the value 0123H. The instruction sequence,

```
PUSH DPL
PUSH DPH
```

will leave the Stack Pointer set to 0BH and store 23H and 01H in internal RAM locations 0AH and 0BH, respectively.

**Bytes:** 2

**Cycles:** 2

**Encoding:**

```
1 1 0 0 0 0 0 0
```

direct address

**Operation:**

```
PUSH (SP) ← (SP) + 1
((SP)) ← (direct)
```
RET

<table>
<thead>
<tr>
<th>Function:</th>
<th>Return from subroutine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>RET pops the high- and low-order bytes of the PC successively from the stack, decrementing the Stack Pointer by two. Program execution continues at the resulting address, generally the instruction immediately following an ACALL or LCALL. No flags are affected.</td>
</tr>
<tr>
<td>Example:</td>
<td>The Stack Pointer originally contains the value 0BH. Internal RAM locations 0AH and 0BH contain the values 23H and 01H, respectively. The instruction, RET will leave the Stack Pointer equal to the value 09H. Program execution will continue at location 0123H.</td>
</tr>
<tr>
<td>Bytes:</td>
<td>1</td>
</tr>
<tr>
<td>Cycles:</td>
<td>2</td>
</tr>
<tr>
<td>Encoding:</td>
<td>0 0 1 0 0 0 1 0</td>
</tr>
<tr>
<td>Operation:</td>
<td>RET (PC15-8) ← ((SP)) (SP) ← (SP) − 1 (PC7-0) ← ((SP)) (SP) ← (SP) − 1</td>
</tr>
</tbody>
</table>

RETI

<table>
<thead>
<tr>
<th>Function:</th>
<th>Return from interrupt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>RETI pops the high- and low-order bytes of the PC successively from the stack, and restores the interrupt logic to accept additional interrupts at the same priority level as the one just processed. The Stack Pointer is left decremented by two. No other registers are affected; the PSW is not automatically restored to its pre-interrupt status. Program execution continues at the resulting address, which is generally the instruction immediately after the point at which the interrupt request was detected. If a lower- or same-level interrupt has been pending when the RETI instruction is executed, that one instruction will be executed before the pending interrupt is processed.</td>
</tr>
<tr>
<td>Example:</td>
<td>The Stack Pointer originally contains the value 0BH. An interrupt was detected during the instruction ending at location 0122H. Internal RAM locations 0AH and 0BH contain the values 23H and 01H, respectively. The instruction, RETI will leave the Stack Pointer equal to 09H and return program execution to location 0123H.</td>
</tr>
<tr>
<td>Bytes:</td>
<td>1</td>
</tr>
<tr>
<td>Cycles:</td>
<td>2</td>
</tr>
<tr>
<td>Encoding:</td>
<td>0 0 1 1 0 0 1 0</td>
</tr>
<tr>
<td>Operation:</td>
<td>RETI (PC15-8) ← ((SP)) (SP) ← (SP) − 1 (PC7-0) ← ((SP)) (SP) ← (SP) − 1</td>
</tr>
</tbody>
</table>
RL A

Function: Rotate Accumulator Left
Description: The eight bits in the Accumulator are rotated one bit to the left. Bit 7 is rotated into the bit 0 position. No flags are affected.
Example: The Accumulator holds the value 0C5H (11000101B). The instruction,
        RL A
leaves the Accumulator holding the value 8BH (10001011B) with the carry unaffected.
Bytes: 1
Cycles: 1
Encoding: 0 0 1 0 0 0 1 1
Operation: RL
           (A_{n+1}) \leftarrow (A_n), n = 0 \to 6
           (A0) \leftarrow (A7)

RLC A

Function: Rotate Accumulator Left through the Carry flag
Description: The eight bits in the Accumulator and the carry flag are together rotated one bit to the left. Bit 7 moves into the carry flag; the original state of the carry flag moves into the bit 0 position. No other flags are affected.
Example: The Accumulator holds the value 0C5H (11000101B), and the carry is zero. The instruction,
        RLC A
leaves the Accumulator holding the value 8AH (10001010B) with the carry set.
Bytes: 1
Cycles: 1
Encoding: 0 0 1 1 0 0 1 1
Operation: RLC
           (A_{n+1}) \leftarrow (A_n), n = 0 \to 6
           (A0) \leftarrow (C)
           (C) \leftarrow (A7)
RR  A

Function:  Rotate Accumulator Right

Description:  The eight bits in the Accumulator are rotated one bit to the right. Bit 0 is rotated into the bit 7 position. No flags are affected.

Example:  The Accumulator holds the value 0C5H (11000101B). The instruction,

         RR  A

leaves the Accumulator holding the value 0E2H (11100010B) with the carry unaffected.

Bytes:  1

Cycles:  1

Encoding:  0 0 0/0 0 1 1

Operation:  RR

         (A_n) \leftarrow (A_{n+1}), n = 0 \ldots 6

         (A7) \leftarrow (A0)

RRC  A  

Function:  Rotate Accumulator Right through the Carry flag

Description:  The eight bits in the Accumulator and the carry flag are together rotated one bit to the right. Bit 0 moves into the carry flag; the original state of the carry flag moves into the bit 7 position. No other flags are affected.

Example:  The Accumulator holds the value 0C5H (11000101B), and the carry is zero. The instruction,

         RRC  A

leaves the Accumulator holding the value 62 (01100010B) with the carry set.

Bytes:  1

Cycles:  1

Encoding:  0 0 0/1 0 0/1 1

Operation:  RRC

         (A_n) \leftarrow (A_{n+1}), n = 0 \ldots 6

         (A7) \leftarrow (C)

         (C) \leftarrow (A0)
### SETB <bit>

<table>
<thead>
<tr>
<th>Function:</th>
<th>Set Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>SETB sets the indicated bit to one. SETB can operate on the carry flag or any directly addressable bit. No other flags are affected.</td>
</tr>
<tr>
<td>Example:</td>
<td>The carry flag is cleared. Output Port 1 has been written with the value 34H (00110100B). The instructions, SETB C SETB P1.0 will leave the carry flag set to 1 and change the data output on Port 1 to 35H (00110101B).</td>
</tr>
</tbody>
</table>

### SETB C

| Bytes: | 1 |
| Cycles: | 1 |
| Encoding: | 11010011 |
| Operation: | SETB (C) ← 1 |

### SETB bit

| Bytes: | 2 |
| Cycles: | 1 |
| Encoding: | 11010010 | bit address |
| Operation: | SETB (bit) ← 1 |

### SJMP rel

<table>
<thead>
<tr>
<th>Function:</th>
<th>Short Jump</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>Program control branches unconditionally to the address indicated. The branch destination is computed by adding the signed displacement in the second instruction byte to the PC, after incrementing the PC twice. Therefore, the range of destinations allowed is from 128 bytes preceding this instruction to 127 bytes following it.</td>
</tr>
<tr>
<td>Example:</td>
<td>The label &quot;RELADR&quot; is assigned to an instruction at program memory location 0123H. The instruction, SJMP RELADR will assemble into location 0100H. After the instruction is executed, the PC will contain the value 0123H. <em>(Note: Under the above conditions the instruction following SJMP will be at 102H. Therefore, the displacement byte of the instruction will be the relative offset (0123H-0102H) = 21H. Put another way, an SJMP with a displacement of 0FEH would be a one-instruction infinite loop.)</em></td>
</tr>
<tr>
<td>Bytes:</td>
<td>2</td>
</tr>
<tr>
<td>Cycles:</td>
<td>2</td>
</tr>
<tr>
<td>Encoding:</td>
<td>10000000</td>
</tr>
<tr>
<td>Operation:</td>
<td>SJMP (PC) ← (PC) + 2 (PC) ← (PC) + rel</td>
</tr>
</tbody>
</table>
SUBB A, <src-byte>

Function: Subtract with borrow

Description: SUBB subtracts the indicated variable and the carry flag together from the Accumulator, leaving the result in the Accumulator. SUBB sets the carry (borrow) flag if a borrow is needed for bit 7, and clears C otherwise. (If C was set before executing a SUBB instruction, this indicates that a borrow was needed for the previous step in a multiple precision subtraction, so the carry is subtracted from the Accumulator along with the source operand.) AC is set if a borrow is needed for bit 3, and cleared otherwise. OV is set if a borrow is needed into bit 6, but not into bit 7, or into bit 7, but not bit 6.

When subtracting signed integers OV indicates a negative number produced when a negative value is subtracted from a positive value, or a positive result when a positive number is subtracted from a negative number.

The source operand allows four addressing modes: register, direct, register-indirect, or immediate.

Example: The Accumulator holds 0C9H (11001001B), register 2 holds 54H (01010100B), and the carry flag is set. The instruction,

```
SUBB A,R2
```

will leave the value 74H (01110100B) in the Accumulator, with the carry flag and AC cleared but OV set.

Notice that 0C9H minus 54H is 75H The difference between this and the above result is due to the carry (borrow) flag being set before the operation. If the state of the carry is not known before starting a single or multiple-precision subtraction, it should be explicitly cleared by a CLR C instruction.

SUBB A,Rn

Bytes: 1

Cycles: 1

Encoding: 1 0 0 1 1 r r r

Operation: SUBB (A) ← (A) − (C) − (Rn)

SUBB A,direct

Bytes: 2

Cycles: 1

Encoding: 1 0 0 1 0 1 0 1  (direct address)

Operation: SUBB (A) ← (A) − (C) − (direct)

SUBB A,@Ri

Bytes: 1

Cycles: 1

Encoding: 1 0 0 1 0 1 1 i

Operation: SUBB (A) ← (A) − (C) − (Ri)

SUBB A,#data

Bytes: 2

Cycles: 1

Encoding: 1 0 0 1 0 1 0 0  (immediate data)

Operation: SUBB (A) ← (A) − (C) − (#data)
### SWAP A

**Function:** Swap nibbles within the Accumulator

**Description:** SWAP A interchanges the low- and high-order nibbles (four-bit fields) of the Accumulator (bits 3-0 and bits 7-4). The operation can also be thought of as a four-bit rotate instruction. No flags are affected.

**Example:** The Accumulator holds the value 0C5H (11000101B). The instruction,

```
SWAP A
```

leaves the Accumulator holding the value 5CH (01011100B).

**Bytes:** 1  
**Cycles:** 1 

<table>
<thead>
<tr>
<th>Encoding:</th>
<th>1 1 0 0</th>
<th>0 1 0 0</th>
</tr>
</thead>
</table>

**Operation:**

\[
(A_{3:0}) \leftrightarrow (A_{7:4})
\]

### XCH A,<byte>

**Function:** Exchange Accumulator with byte variable

**Description:** XCH loads the Accumulator with the contents of the indicated variable, at the same time writing the original Accumulator contents to the indicated variable. The source/destination operand can use register, direct, or register-indirect addressing.

**Example:** R0 contains the address 20H. The Accumulator holds the value 3FH (0011111B). Internal RAM location 20H holds the value 75H (01110101B). The instruction,

```
XCH A, @R0
```

will leave the RAM location 20H holding the values 3FH (0011111B) and 75H (01110101B) in the Accumulator.

### XCH A,Rn

**Bytes:** 1 
**Cycles:** 1 

<table>
<thead>
<tr>
<th>Encoding:</th>
<th>1 1 0 0</th>
<th>1 r r r</th>
</tr>
</thead>
</table>

**Operation:**

\[
XCH (A) \leftrightarrow (R_n)
\]

### XCH A, direct

**Bytes:** 2 
**Cycles:** 1 

<table>
<thead>
<tr>
<th>Encoding:</th>
<th>1 1 0 0</th>
<th>0 1 0 1</th>
</tr>
</thead>
</table>

**Operation:**

\[
XCH (A) \leftrightarrow \text{(direct)}
\]

### XCH A, @ Ri

**Bytes:** 1 
**Cycles:** 1 

<table>
<thead>
<tr>
<th>Encoding:</th>
<th>1 1 0 0</th>
<th>0 1 1 i</th>
</tr>
</thead>
</table>

**Operation:**

\[
XCH (A) \leftrightarrow \text{((R_i))}
\]
XCHD  A, @Ri

Function: Exchange Digit

Description: XCHD exchanges the low-order nibble of the Accumulator (bits 3-0), generally representing a hexadecimal or BCD digit, with that of the internal RAM location indirectly addressed by the specified register. The high-order nibbles (bits 7-4) of each register are not affected. No flags are affected.

Example: R0 contains the address 20H. The Accumulator holds the value 36H (00110110B). Internal RAM location 20H holds the value 75H (01110101B). The instruction,
XCHD  A, @R0
will leave RAM location 20H holding the value 76H (01110110B) and 35H (00110101B) in the Accumulator.

Bytes: 1

Cycles: 1

Encoding: 1 1 0 1 0 1 1

Operation: XCHD
(A3-0) ↔ (Ri3-0)

XRL  <dest-byte>,<src-byte>

Function: Logical Exclusive-OR for byte variables

Description: XRL performs the bitwise logical Exclusive-OR operation between the indicated variables, storing the results in the destination. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the Accumulator or immediate data.

(Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins.)

Example: If the Accumulator holds 0C3H (11000011B) and register 0 holds 0AH (10101010B) then the instruction,
XRL  A, R0
will leave the Accumulator holding the value 69H (01101001B).

When the destination is a directly addressed byte, this instruction can complement combinations of bits in any RAM location or hardware register. The pattern of bits to be complemented is then determined by a mask byte, either a constant contained in the instruction or a variable computed in the Accumulator at run-time. The instruction,
XRL  P1,#00110001B
will complement bits 5, 4, and 0 of output Port 1.
XRL A, Rn
Bytes: 1
Cycles: 1
Encoding: 0 1 1 0 1 r r r
Operation: XRL
(A) ← (A) xor (Rn)

XRL A, direct
Bytes: 2
Cycles: 1
Encoding: 0 1 1 0 0 1 0 1
direct address
Operation: XRL
(A) ← (A) xor (direct)

XRL A, @Ri
Bytes: 1
Cycles: 1
Encoding: 0 1 1 0 0 1 1 i
Operation: XRL
(A) ← (A) xor (Ri)

XRL A, #data
Bytes: 2
Cycles: 1
Encoding: 0 1 1 0 0 1 0 0
immediate data
Operation: XRL
(A) ← (A) xor #data

XRL direct, A
Bytes: 2
Cycles: 1
Encoding: 0 1 1 0 0 0 1 0
direct address
Operation: XRL
(direct) ← (direct) xor (A)

XRL direct, #data
Bytes: 3
Cycles: 2
Encoding: 0 1 1 0 0 0 1 1
direct address immediate data
Operation: XRL
(direct) ← (direct) xor #data
APPENDIX C. DATASHEETS

The following Pages include manufacturer data sheets from the following companies:

- **UTMC**
  - UT69RH051
  - UT28F64
  - UT67164

- **MAXIM**
  - MAX 232
  - MAX 233

- **Harris**
  - HS-82C55ARH
FEATURES

- Three 16-bit timer/counters
  - High speed output
  - Compare/capture
  - Pulse width modulator
  - Watchdog timer capabilities
- 256 bytes of on-chip data RAM
- 32 programmable I/O lines
- 7 interruptable sources
- Programmable serial channel with:
  - Framing error detection
  - Automatic address recognition
- TTL and CMOS compatible logic levels
- 64K external data and program memory space
- MCS*-51 fully compatible instruction set

- Flexible clock operation
  - 1Hz to 20MHz with external clock
  - 2MHz to 20MHz using internal oscillator with external crystal
- Radiation-hardened process and design; total dose irradiation testing MIL-STD-883 Method 1019
  - Total dose: 1.0E6 rads (Si)
  - Single event upset: <25.6E-6 errors/device-day
  - Latchup immune
- Post-radiation AC/DC performance characteristics guaranteed to MIL-STD-883 Method 1019 testing at 1.0E6 rads (Si)
- Built on low-power, 1.2μ CMOS process
- Packaging options:
  - 40-pin DIP
  - 44-lead flatpack

Figure 1. UT69RH051 MicroController Block Diagram
1.0 INTRODUCTION

The UT69RH051 is a radiation-tolerant 8-bit microcontroller that is pin equivalent to the Intel 8XC51E microcontroller. The UT69RH051's static design allows operation from 1Hz to 20MHz. This product brief will describe hardware and software interfaces to the UT69RH051.

2.0 SIGNAL DESCRIPTION

VDD: +5V Supply voltage

VSS: Circuit Ground

Port 0 (P0.0 - P0.7): Port 0 is an 8-bit port. Its pins are used as the low-order multiplexed address and data bus during accesses to external program and data memory. Port 0 pins use strong internal pullups when emitting 1's, and are TTL compatible.

Port 1 (P1.0 - P1.7): Port 1 is an 8-bit bidirectional I/O port with internal pullups. The output buffers can drive TTL loads. When the Port 1 pins have 1's written to them, they are pulled high by the internal pullups and can be used as inputs in this state. As inputs, any pins that are externally pulled low will source current because of the pullups. In addition, Port 1 pins have the alternate uses shown in Table 1.

Port 2 (P2.0 - P2.7): Port 2 is an 8-bit port. Its pins are used as the high-order address bus during accesses to external Program Memory and during accesses to external Data Memory that uses 16-bit addresses (i.e., MOVX@DPTR). It uses strong internal pullups when emitting 1's in this mode. During operations that do not require a 16-bit address, Port 2 emits the contents of the P2 Special Function Registers (SFR). The pins have internal pullups and can drive TTL loads.

Port 3 (P3.0 - P3.7): Port 3 is an 8-bit bidirectional I/O port with internal pullups. The output buffers can drive TTL loads. When the Port 3 pins have 1's written to them, they are pulled high by the internal pullups and can be used as inputs in this state. As inputs, any pins that are externally pulled low will source current because of the pullups. In addition, Port 3 pins have the alternate uses shown in Table 2.

### Table 1. Port 1 Alternate Functions

<table>
<thead>
<tr>
<th>Port Pin</th>
<th>Alternate Name</th>
<th>Alternate Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1.0</td>
<td>T2</td>
<td>External clock input to Timer/Counter 2</td>
</tr>
<tr>
<td>P1.1</td>
<td>T2EX</td>
<td>Timer/Counter 2 Capture/Reload trigger and direction control</td>
</tr>
<tr>
<td>P1.2</td>
<td>ECI</td>
<td>External count input to PCA</td>
</tr>
<tr>
<td>P1.3</td>
<td>CEX0</td>
<td>External I/O for PCA capture/compare Module 0</td>
</tr>
<tr>
<td>P1.4</td>
<td>CEX1</td>
<td>External I/O for PCA capture/compare Module 1</td>
</tr>
<tr>
<td>P1.5</td>
<td>CEX2</td>
<td>External I/O for PCA capture/compare Module 2</td>
</tr>
<tr>
<td>P1.6</td>
<td>CEX3</td>
<td>External I/O for PCA capture/compare Module 3</td>
</tr>
<tr>
<td>P1.7</td>
<td>CEX4</td>
<td>External I/O for PCA capture/compare Module 4</td>
</tr>
</tbody>
</table>

### Table 2. Port 3 Alternate Functions

<table>
<thead>
<tr>
<th>Port Pin</th>
<th>Alternate Name</th>
<th>Alternate Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>P3.0</td>
<td>RXD</td>
<td>Serial port input</td>
</tr>
<tr>
<td>P3.1</td>
<td>TXD</td>
<td>Serial port output</td>
</tr>
<tr>
<td>P3.2</td>
<td>INT0</td>
<td>External interrupt 0</td>
</tr>
<tr>
<td>P3.3</td>
<td>INT1</td>
<td>External interrupt 1</td>
</tr>
<tr>
<td>P3.4</td>
<td>T0</td>
<td>External clock input for Timer 0</td>
</tr>
<tr>
<td>P3.5</td>
<td>T1</td>
<td>External clock input for Timer 1</td>
</tr>
<tr>
<td>P3.6</td>
<td>WR</td>
<td>External Data Memory write strobe</td>
</tr>
<tr>
<td>P3.7</td>
<td>RD</td>
<td>External Data Memory read strobe</td>
</tr>
</tbody>
</table>

RST: Reset Input. A high on this input for one oscillator period while the oscillator is running resets the device. All ports and SFRs reset to their default conditions. Internal data memory is undefined after reset. Program execution begins within 12 oscillator periods (one machine cycle) after the RST signal is brought low. RST contains an internal pulldown resistor to allow implementing power-up reset with only an external capacitor.
ALE: Address Latch Enable. The ALE output is a pulse for latching the low byte of the address during accesses to external memory. In normal operation the ALE pulse is output every sixth oscillator cycle and may be used for external timing or clocking. However, during each access to external Data Memory (MOVX instruction), one ALE pulse is skipped.

PSEN: Program Store Enable. This active low signal is the address strobe to the external program memory. PSEN is activated every sixth oscillator cycle except that two PSEN activations are skipped during external data memory accesses.

EA: External Access Enable. This pin should be strapped to VSS (Ground) for the UT69RH051.

XTAL1: Input to the inverting oscillator amplifier.

XTAL2: Output from the inverting oscillator amplifier.

2.1 Hardware/Software Interface

2.1.1 Memory
The UT69RH051 has a separate address space for Program and Data Memory. Internally the UT69RH051 contains 256 bytes of Data Memory. It can address up to 64Kbytes of external Data Memory and 64Kbytes of external Program Memory.

2.1.1.1 Program Memory
There is no internal program memory in the UT69RH051. All program memory is accessed as external through ports P0 and P2. The EA pin must be tied to VSS (ground) to enable access to external locations $0000_{H}$ through $7FF_{F}$.

2.1.1.2 Data Memory
The UT69RH051 implements 256 bytes of internal data RAM. The upper 128 bytes of this RAM occupy a parallel address space to the SFRs. The CPU determines if the internal access to an address above $7F_{H}$ is to the upper 128 bytes of RAM or to the SFR space by the addressing mode of the instruction. If indirect addressing is used, the access is to the SFR space. If indirect addressing is used, the access is to the internal RAM. Stack operations are indirectly addressed so the upper portion of RAM can be used as stack space. Figure 3 shows the organization of the internal Data Memory.

The first 32 bytes are reserved for four register banks of eight bytes each. The processor uses one of the four banks as its working registers depending on the RS1 and RS0 bits in the PSW SFR. At reset, bank 0 is selected. If four registre banks are not required, use the unused banks as general purpose scratch pad memory. The next 16 bytes (128 bits) are individually bit addressable. The remaining bytes are byte addressable and can be used as general purpose scratch pad memory. For addresses 0 to 7F_{H}, use either direct or indirect addressing. For addresses larger than 7F_{H}, use only indirect addressing.

In addition to the internal Data Memory, the processor can access 64 Kbytes of external Data Memory. The MOVX instruction accesses external Data Memory.

2.1.2 Special Function Registers
Table 3 contains the SFR memory map. Unoccupied addresses are not implemented on the device. Read accesses to these addresses will return unknown values and write accesses will have no effect.

---

Figure 2. UT69RH051 Pin Connections

<table>
<thead>
<tr>
<th>Port</th>
<th>Pin</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>
<th>10</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>
<th>21</th>
</tr>
</thead>
<tbody>
<tr>
<td>T2</td>
<td>P1.0</td>
<td>40</td>
<td>39</td>
<td>38</td>
<td>37</td>
<td>36</td>
<td>35</td>
<td>34</td>
<td>33</td>
<td>32</td>
<td>31</td>
<td>30</td>
<td>29</td>
<td>28</td>
<td>27</td>
<td>26</td>
<td>25</td>
<td>24</td>
<td>23</td>
<td>22</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>T2EX</td>
<td>P1.1</td>
<td></td>
<td>2</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>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EC1</td>
<td>P1.2</td>
<td>1</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>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EX0</td>
<td>P1.3</td>
<td>3</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>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EX1</td>
<td>P1.4</td>
<td>5</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>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EX2</td>
<td>P1.5</td>
<td>7</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>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EX3</td>
<td>P1.6</td>
<td>9</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>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EX4</td>
<td>P1.7</td>
<td>11</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>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RXD</td>
<td>P1.8</td>
<td>13</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>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TXD</td>
<td>P1.9</td>
<td>15</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>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INT0</td>
<td>P1.10</td>
<td>17</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>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INT1</td>
<td>P1.11</td>
<td>19</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>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T0</td>
<td>P1.12</td>
<td>21</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>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1</td>
<td>P1.13</td>
<td>23</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>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WR</td>
<td>P1.14</td>
<td>25</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>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RD</td>
<td>P1.15</td>
<td>27</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>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2. UT69RH051 Pin Connections

VSS = 20

P2.0 = 21
2.1.3 Reset

The reset input is the RST pin. To reset, hold the RST pin high for a minimum of 24 oscillator period while the oscillator is running. The CPU generates an internal reset from the external signal. The ports pins are driven to the reset state as soon as a valid high is detected on the RST pin.

While RST is high, PSEN, ALE, and the port pins are pulled weakly high. All SFRs are reset to their reset values as shown in table 3. The internal Data Memory content is indeterminate.

The processor will begin operation one machine cycle after the RST line is brought low. A memory access occurs immediately after the RST line is brought low, but the data is not brought into the processor. The memory access repeats on the next machine cycle and actual processing begins at that time.

2.1.4 Instruction Set

The instruction set for the UT69RH051 is compatible to the Intel MCS-51 instruction set used on the 8XC51FC.
<table>
<thead>
<tr>
<th></th>
<th>F8</th>
<th>F0</th>
<th>E8</th>
<th>E0</th>
<th>D8</th>
<th>D0</th>
<th>C8</th>
<th>C0</th>
<th>B8</th>
<th>B0</th>
<th>A8</th>
<th>A0</th>
<th>98</th>
<th>90</th>
<th>88</th>
<th>80</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CH</td>
<td>B</td>
<td>CL</td>
<td>ACC</td>
<td>CCON</td>
<td>PSW</td>
<td>T2CON</td>
<td>T2MOD</td>
<td>00000000</td>
<td>IP</td>
<td>P3</td>
<td>IE</td>
<td>P2</td>
<td>SCON</td>
<td>P1</td>
<td>TCON</td>
<td>P0</td>
</tr>
<tr>
<td></td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CCAP0H</td>
<td>CCAP1H</td>
<td>CCAP0L</td>
<td>CCAP1L</td>
<td>CCAP2L</td>
<td>CCAP2H</td>
<td>CCAP3L</td>
<td>CCAP4L</td>
<td>00000000</td>
<td>SADEN</td>
<td>SADDR</td>
<td>SBUF</td>
<td>00000000</td>
<td>SCON</td>
<td>TCON</td>
<td>00000000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CCAP3H</td>
<td>CCAP4H</td>
<td>CCAP4L</td>
<td>CCAP3L</td>
<td>CCAP2L</td>
<td>CCAP1L</td>
<td>CCAP0L</td>
<td>CCAP0H</td>
<td>X00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
</tr>
<tr>
<td></td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td>X00000000</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FF</td>
<td>F7</td>
<td>EF</td>
<td>E7</td>
<td>DF</td>
<td>D7</td>
<td>CF</td>
<td>C7</td>
<td>BF</td>
<td>B7</td>
<td>AF</td>
<td>A7</td>
<td>9F</td>
<td>97</td>
<td>8F</td>
<td>87</td>
<td>PCON</td>
</tr>
<tr>
<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></td>
<td>00XX00XX</td>
</tr>
</tbody>
</table>

Notes:
1. Values shown are the reset values of the registers.
2. X = undefined.
3.0 RADIATION HARDNESS

The UT69RH051 incorporates special design and layout features which allow operation in high-level radiation environments. UTMC has developed special low-temperature processing techniques designed to enhance the total-dose radiation hardness of both the gate oxide and the field oxide while maintaining the circuit density and reliability. For transient radiation hardness and latchup immunity, UTMC builds all radiation-hardened products on epitaxial wafers using an advanced twin-tub CMOS process. In addition, UTMC pays special attention to power and ground distribution during the design phase, minimizing dose-rate upset caused by rail collapse.

RADIATION HARDNESS DESIGN SPECIFICATIONS

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>CONDITION</th>
<th>MINIMUM</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Dose</td>
<td>+25°C per MIL-STD-883 Method 1019</td>
<td>1.0E6</td>
<td>rads(Si)</td>
</tr>
<tr>
<td>Dose Rate Upset</td>
<td>≤ 4μs pulselength</td>
<td>1.0E8</td>
<td>rads(Si)/sec</td>
</tr>
<tr>
<td>Dose Rate Survival</td>
<td>20ns pulselength</td>
<td>1.0E10</td>
<td>rads(Si)/sec</td>
</tr>
<tr>
<td>LET Threshold</td>
<td>-55°C to +125°C</td>
<td>36</td>
<td>MeV·cm²/mg</td>
</tr>
<tr>
<td>Neutron Fluence</td>
<td>1MeV equivalent</td>
<td>1.0E14</td>
<td>n/cm²</td>
</tr>
</tbody>
</table>

Note:
1. The UT69RH051 will not latchup during radiation exposure under recommended operating conditions.

4.0 ABSOLUTE MAXIMUM RATINGS

(Referenced to VSS)

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>LIMITS</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDD</td>
<td>DC Supply Voltage</td>
<td>-0.5 to 7.0</td>
<td>V</td>
</tr>
<tr>
<td>VIO</td>
<td>Voltage on Any Pin</td>
<td>-0.5 to VDD+3V</td>
<td>V</td>
</tr>
<tr>
<td>TSTG</td>
<td>Storage Temperature</td>
<td>-65 to +150</td>
<td>°C</td>
</tr>
<tr>
<td>PD</td>
<td>Maximum Power Dissipation</td>
<td>750</td>
<td>mW</td>
</tr>
<tr>
<td>TJ</td>
<td>Maximum Junction Temperature</td>
<td>175</td>
<td>°C</td>
</tr>
<tr>
<td>ΘJC</td>
<td>Thermal Resistance, Junction-to-Case ²</td>
<td>10</td>
<td>°C/W</td>
</tr>
<tr>
<td>Ii</td>
<td>DC Input Current</td>
<td>± 10</td>
<td>mA</td>
</tr>
</tbody>
</table>

Notes:
1. Stresses outside the listed absolute maximum ratings may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at these or any other conditions beyond limits indicated in the operational sections of this specification is not recommended. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.
2. Test per MIL-STD-883, Method 1012.
### 6.0 DC ELECTRICAL CHARACTERISTICS (Pre/Post-Radiation)*

**$V_{DD} = 5.0\text{V} \pm 10\%$; $TA = -55^\circ\text{C} < T_C < +125^\circ\text{C}$**

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>CONDITION</th>
<th>MINIMUM</th>
<th>MAXIMUM</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{IL}$</td>
<td>Low-level Input Voltage</td>
<td></td>
<td>-0.5</td>
<td>.8</td>
<td>V</td>
</tr>
<tr>
<td>$V_{IH}$</td>
<td>High-level Input Voltage (except XTAL2, RST, EA)</td>
<td></td>
<td>2.0</td>
<td>$V_{DD}+0.3$</td>
<td>V</td>
</tr>
<tr>
<td>$V_{IHI}$</td>
<td>High-level Input Voltage (XTAL, RST)</td>
<td></td>
<td>3.85</td>
<td>$V_{DD}+0.3$</td>
<td>V</td>
</tr>
<tr>
<td>$V_{OL}$</td>
<td>Low-level Output Voltage$^1$ (Ports 1, 2 and 3)</td>
<td>$I_{OL} = 100\mu\text{A}$</td>
<td>0.3</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$I_{OL} = 1.6\text{mA}$</td>
<td>0.45</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$I_{OL} = 3.5\text{mA}$</td>
<td>1.0</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>$V_{OL1}$</td>
<td>Low-level Output Voltage$^1$ (Port 0, ALE/PROG, PSEN)</td>
<td>$I_{OL} = 200\mu\text{A}$</td>
<td>0.3</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$I_{OL} = 3.2\text{mA}$</td>
<td>0.45</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$I_{OL} = 7.0\text{mA}$</td>
<td>1.0</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>$V_{OH}$</td>
<td>High-level Output Voltage (Ports 1, 2, and 3 ALE/PROG and PSEN)</td>
<td>$I_{OH} = -10\mu\text{A}$</td>
<td>4.2</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$I_{OH} = -30\mu\text{A}$</td>
<td>3.8</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$I_{OH} = -60\mu\text{A}$</td>
<td>3.0</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>$V_{OHI}$</td>
<td>High-level Output Voltage (Port 0 in External Bus Mode)</td>
<td>$I_{OH} = -200\mu\text{A}$</td>
<td>4.2</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$I_{OH} = -3.2\text{mA}$</td>
<td>3.8</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$I_{OH} = -7.0\text{mA}$</td>
<td>3.0</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>$I_{IL}$</td>
<td>Logical 0 Input Current (Ports 1, 2, and 3)</td>
<td>$V_{IN} = 0.45\text{V}$</td>
<td>-50</td>
<td></td>
<td>$\mu\text{A}$</td>
</tr>
<tr>
<td>$I_{LI}$</td>
<td>Input Leakage Current (Port 0)</td>
<td>$V_{IN} = V_{IL}$ or $V_{IH}$</td>
<td>± 10</td>
<td></td>
<td>$\mu\text{A}$</td>
</tr>
<tr>
<td>$I_{TL}$</td>
<td>Logical 1 to 0 Transition Current (Ports 1, 2, and 3)</td>
<td>$V_{IN} = 2\text{V}$</td>
<td>-650</td>
<td></td>
<td>$\mu\text{A}$</td>
</tr>
<tr>
<td>$C_{IO}$</td>
<td>Pin Capacitance</td>
<td>@ 1MHz, 25°C</td>
<td>10</td>
<td></td>
<td>pF</td>
</tr>
<tr>
<td>$I_{CC}$</td>
<td>Power Supply Current: (Running at 16MHz)</td>
<td>Note 2</td>
<td>52</td>
<td></td>
<td>mA</td>
</tr>
</tbody>
</table>

**Notes:**


1. Under steady state (non-transient) conditions, $I_{OL}$ must be limited externally as follows:
   - Maximum $I_{OL}$ per port pin: 10mA
   - Maximum $I_{OL}$ per 8-bit port:
     - Port 0: 26mA
     - Ports 1, 2, & 3: 15mA
     - Maximum total $I_{OL}$ for all output pins: 71mA

   If $I_{OL}$ exceeds the test condition, $V_{OL}$ may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. See figures 4, 5, and 6 for test conditions.
Figure 4. \( I_{DD} \) Test Condition, Active Mode
All other pins disconnected

\[ t_{CLCH} = t_{CHCL} = 5\text{ns} \]

Figure 5. Clock Signal Waveform for \( I_{CC} \) Tests in Active and Idle Modes
\[ t_{CLCH} = t_{CHCL} = 5\text{ns} \]
### 7.0 AC CHARACTERISTICS READ CYCLE (Post-Radiation)*

*(V<sub>DD</sub> = 5.0V ± 10%; -55°C < T<sub>C</sub> < +125°C)*

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>Parameter</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>tCLCL</td>
<td>Clock Period</td>
<td>50</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>f&lt;sub&gt;CLCL&lt;/sub&gt;</td>
<td>Oscillator Frequency</td>
<td></td>
<td>16</td>
<td>MHz</td>
</tr>
<tr>
<td>t&lt;sub&gt;LHLL&lt;/sub&gt;</td>
<td>ALE Pulse Width</td>
<td>2&lt;sup&gt;tCLCL-40&lt;/sup&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;AVLL&lt;/sub&gt;</td>
<td>Address Valid to ALE Low</td>
<td>t&lt;sub&gt;CLCL-40&lt;/sub&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;LLAX&lt;/sub&gt;</td>
<td>Address Hold after ALE Low</td>
<td>t&lt;sub&gt;CLCL-30&lt;/sub&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;LLIV&lt;/sub&gt;</td>
<td>ALE Low to Valid Instruction In</td>
<td>4&lt;sup&gt;tCLCL-100&lt;/sup&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;LLPL&lt;/sub&gt;</td>
<td>ALE Low to PSEN Low</td>
<td>t&lt;sub&gt;CLCL-30&lt;/sub&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;PLPH&lt;/sub&gt;</td>
<td>PSEN Pulse Width</td>
<td>3&lt;sup&gt;tCLCL-45&lt;/sup&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;PLIV&lt;/sub&gt;</td>
<td>PSEN Low to Valid Instruction In</td>
<td>3&lt;sup&gt;tCLCL-105&lt;/sup&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;PXIX&lt;/sub&gt;</td>
<td>Input Instruction Hold after PSEN</td>
<td>0</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;PXIZ&lt;/sub&gt;</td>
<td>Input Instruction Float After PSEN</td>
<td>t&lt;sub&gt;CLCL-25&lt;/sub&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;AVIV&lt;/sub&gt;</td>
<td>Address to Valid Instruction In</td>
<td>5&lt;sup&gt;tCLCL-105&lt;/sup&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;PLAZ&lt;/sub&gt;</td>
<td>PSEN Low to Address Float</td>
<td>10</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;RLRH&lt;/sub&gt;</td>
<td>RD Pulse Width</td>
<td>6&lt;sup&gt;tCLCL-100&lt;/sup&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;WLWH&lt;/sub&gt;</td>
<td>WR Pulse Width</td>
<td>6&lt;sup&gt;tCLCL-100&lt;/sup&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;RLDV&lt;/sub&gt;</td>
<td>RD Low to Valid Data In</td>
<td>5&lt;sup&gt;tCLCL-165&lt;/sup&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;RHDX&lt;/sub&gt;</td>
<td>Data Hold After RD</td>
<td>0</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;RHDZ&lt;/sub&gt;</td>
<td>Data Float After RD</td>
<td>2&lt;sup&gt;tCLCL-60&lt;/sup&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;LLDV&lt;/sub&gt;</td>
<td>ALE Low Valid Data In</td>
<td>8&lt;sup&gt;tCLCL-150&lt;/sup&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;AVDV&lt;/sub&gt;</td>
<td>Address to Valid Data In</td>
<td>9&lt;sup&gt;tCLCL-165&lt;/sup&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;LLWL&lt;/sub&gt;</td>
<td>ALE Low to RD or WR Low</td>
<td>3&lt;sup&gt;tCLCL-50&lt;/sup&gt;</td>
<td>3&lt;sup&gt;tCLCL+50&lt;/sup&gt;</td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;AVWL&lt;/sub&gt;</td>
<td>Address Valid to WR Low</td>
<td>4&lt;sup&gt;tCLCL-130&lt;/sup&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;QVWX&lt;/sub&gt;</td>
<td>Data Valid Before WR</td>
<td>t&lt;sub&gt;CLCL-50&lt;/sub&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;WHQX&lt;/sub&gt;</td>
<td>Data Hold After WR</td>
<td>t&lt;sub&gt;CLCL-50&lt;/sub&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;QVWH&lt;/sub&gt;</td>
<td>Data Valid to WR High</td>
<td>7&lt;sup&gt;tCLCL-150&lt;/sup&gt;</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;RLAZ&lt;/sub&gt;</td>
<td>RD Low to Address Float</td>
<td>0</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t&lt;sub&gt;WHILH&lt;/sub&gt;</td>
<td>RD or WR High to ALE High</td>
<td>t&lt;sub&gt;CLCL-40&lt;/sub&gt;</td>
<td>t&lt;sub&gt;CLCL+40&lt;/sub&gt;</td>
<td>ns</td>
</tr>
</tbody>
</table>

**Note:**
Figure 6. External Program Memory Read Timing Waveforms

Figure 7. External Data Memory Read Cycle Waveforms

Figure 8. External Data Memory Write Cycle Waveforms
8.0 SERIAL PORT TIMING CHARACTERISTICS

\( V_{DD} = 5.0V \pm 10\% \); -55°C < \( T_{C} < +125°C \)

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>MINIMUM</th>
<th>MAXIMUM</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>t_{XLXL}</td>
<td>Serial Port Clock Period</td>
<td>12 ( t_{CLCL} -10 ) ns</td>
<td>12 ( t_{CLCL} +10 ) ns</td>
<td>ns</td>
</tr>
<tr>
<td>t_{QVXH}</td>
<td>Output Data Setup to Clock Rising Edge</td>
<td>10 ( t_{CLCL} -133 ) ns</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t_{XHQX}</td>
<td>Output Data Hold after Clock Rising Edge</td>
<td>2 ( t_{CLCL} -70 ) ns</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t_{XHDX}</td>
<td>Input Data Hold after Clock Rising Edge</td>
<td>0 ns</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t_{XHDV}</td>
<td>Clock Rising Edge to Input Data Valid</td>
<td>10 ( t_{CLCL} -133 ) ns</td>
<td></td>
<td>ns</td>
</tr>
</tbody>
</table>

Figure 9. Serial Port Timing Waveforms

9.0 EXTERNAL CLOCK DRIVE TIMING CHARACTERISTICS

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>MINIMUM</th>
<th>MAXIMUM</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/( t_{CLCL} )</td>
<td>Oscillator Frequency</td>
<td>16 MHz</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t_{CHCX}</td>
<td>High Time</td>
<td>20 ns</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t_{CLCX}</td>
<td>Low Time</td>
<td>20 ns</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t_{CLCH}</td>
<td>Rise Time</td>
<td>20 ns</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>t_{CHCL}</td>
<td>Fall Time</td>
<td>20 ns</td>
<td></td>
<td>ns</td>
</tr>
</tbody>
</table>

Figure 10. External Clock Drive Timing Waveforms
10.0 PACKAGING

**TOP VIEW**

**SIDE VIEW**

**END VIEW**

Notes:
1. All package finishes are per MIL-1-38535.

Figure 11. 40-pin Side-Brazed DIP
TBD

Figure 12. 44-Lead Flatpack
APPENDIX A
Difference Between Intel 8XC51FC and UTMC69RH051

There are a few areas in which the UT69RH051 differs from the 8XC51FC. These differences will be covered in this section. In this discussion, 8XC51FC will be used generically to refer to all speed grades of the Intel 8XC51FC family, including the 20MHz 8XC51FC-1.

1.0 RESET
The UT69RH051 requires the RST input to be held high for at least 24 oscillator periods to guarantee the reset is completed in the chip. Also, the port pins are reset asynchronously as soon as the RST pin is pulled high. On the UT69RH051 all portions of the chip are reset synchronously when the RST pin is high during a rising edge of the input clock. When coming out of reset, the 8XC51FC takes 1 to 2 machine cycles to begin driving ALE and PSEN immediately after the RST is removed but the access during the first machine cycle after reset is ignored by the processor. The second cycle will repeat the access and processing will begin.

2.0 POWER SAVING MODES OF OPERATION

2.1 Idle Mode
Idle mode and the corresponding control bit in the PCON SFR have not been implemented in the UT69RH051. Setting the idle control bit will have no effect.

2.2 Power Down Mode
Power down mode and the corresponding control bit in the PCON register have not been implemented in the UT69RH051. Setting the power down control bit will have no effect. Also, the Power Off Flag in the PCON has not been implemented.

3.0 ON CIRCUIT EMULATION
The On Circuit Emulation mode of operation in the 8XC51FC has not been implemented in the UT69RH051.

4.0 OPERATING CONDITIONS
The operating voltage range for the 8XC51FC is 5V ± 20%. The operating temperature range is 0° to 70°C. On the UT69RH051, the operating voltage range is 5V ± 10%. The operating temperature range is -55° to +125°C.
APPENDIX B
Impact of External Program ROM

The 8051 family of microcontrollers, including the 8XC51FC, use ports 0 and 2 to access external memory. In implementations with external program memory, these two ports are dedicated to the program ROM interface and cannot be used as Input/Output ports. The UT69RH051 uses external program ROM, so ports 0 and 2 will not be available for I/O.
FEATURES

- Programmable, read-only, asynchronous, radiation-hardened, 8K x 8 memory
  - Supported by industry standard programmers
- 35ns maximum address access time (-55 °C to +125 °C)
- TTL/CMOS compatible input and output levels
- Three-state data bus
- Low operating and standby current
  - Operating: 140mA maximum @28.5MHz
  - Standby: 1mA maximum (post-rad)
  - Derating: 5mA/MHz
- Radiation-hardened process and design; total dose irradiation testing to MIL-STD-883, Method 1019
  - Total dose: 1.0E6 rads(Si)
  - Single event effects:
    - Latchup threshold 80 MeV-cm²/mg (min)
    - Dose rate upset: 1.0E8 rads(Si)/sec
    - Dose rate survival: 1.0E10 rads(Si)/sec
    - Neutron fluence: 1.0E14 n/cm²
- QML Q & V compliant part (check factory for availability)
  - AC and DC testing at factory
- Packaging options:
  - 28-pin 100-mil center DIP (0.600 x 1.4)
  - 28-lead 50-mil center flatpack (0.700 x 0.75)
  - 28-lead 50-mil center flatpack (0.490 x 0.74)
- \( V_{DD} \): 5.0 volts \( \pm 10\% \)

PRODUCT DESCRIPTION

The UT28F64 amorphous silicon anti-fuse PROM is a high performance, asynchronous, radiation-hardened, 8K x 8 programmable memory device. The UT28F64 PROM features fully asynchronous operation requiring no external clocks or timing strobes. An advanced radiation-hardened twin-well CMOS process technology is used to implement the UT28F64. The combination of radiation-hardness, fast access time, and low power consumption make the UT28F64 ideal for high speed systems designed for operation in radiation environments.

![PROM Block Diagram](image)

Figure 1. PROM Block Diagram
DEVICE OPERATION

The UT28F64 has three control inputs: Chip Enable (CE), Program Enable (PE), and Output Enable (OE); thirteen address inputs, A(12:0); and eight bidirectional data lines, DQ(7:0). CE is the device enable input that controls chip selection, active, and standby modes. Asserting CE causes IDD to rise to its active value and decodes the fifteen address inputs to select one of 8,192 words in the memory. PE controls program and read operations. During a read cycle, OE must be asserted to enable the outputs.

PIN CONFIGURATION

<table>
<thead>
<tr>
<th>NC</th>
<th>1</th>
<th>28</th>
<th>VCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>A12</td>
<td>2</td>
<td>27</td>
<td>PE</td>
</tr>
<tr>
<td>A7</td>
<td>3</td>
<td>26</td>
<td>NC</td>
</tr>
<tr>
<td>A6</td>
<td>4</td>
<td>25</td>
<td>A8</td>
</tr>
<tr>
<td>A5</td>
<td>5</td>
<td>24</td>
<td>A9</td>
</tr>
<tr>
<td>A4</td>
<td>6</td>
<td>23</td>
<td>A11</td>
</tr>
<tr>
<td>A3</td>
<td>7</td>
<td>22</td>
<td>OE</td>
</tr>
<tr>
<td>A2</td>
<td>8</td>
<td>21</td>
<td>A10</td>
</tr>
<tr>
<td>A1</td>
<td>9</td>
<td>20</td>
<td>CE</td>
</tr>
<tr>
<td>A0</td>
<td>10</td>
<td>19</td>
<td>DQ7</td>
</tr>
<tr>
<td>DQ9</td>
<td>11</td>
<td>18</td>
<td>DQ6</td>
</tr>
<tr>
<td>DQ1</td>
<td>12</td>
<td>17</td>
<td>DQ5</td>
</tr>
<tr>
<td>DQ2</td>
<td>13</td>
<td>16</td>
<td>DQ4</td>
</tr>
<tr>
<td>GND</td>
<td>14</td>
<td>15</td>
<td>DQ3</td>
</tr>
</tbody>
</table>

PIN NAMES

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A(12:0)</td>
<td>Address</td>
<td></td>
</tr>
<tr>
<td>CE</td>
<td>Chip Enable</td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>Output Enable</td>
<td></td>
</tr>
<tr>
<td>PE</td>
<td>Program Enable</td>
<td></td>
</tr>
<tr>
<td>DQ(7:0)</td>
<td>Data Input/Data Output</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Device Operation Truth Table

<table>
<thead>
<tr>
<th>OE</th>
<th>PE</th>
<th>CE</th>
<th>I/O MODE</th>
<th>MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Three-state</td>
<td>Standby</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Data Out</td>
<td>Read</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Data In</td>
<td>Program</td>
</tr>
</tbody>
</table>

Notes:
1. "X" is defined as a “don’t care” condition.
2. Device active; outputs disabled.

ABSOLUTE MAXIMUM RATINGS

(Referenced to VSS)

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>LIMITS</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDD</td>
<td>DC supply voltage</td>
<td>-0.3 to 7.0</td>
<td>V</td>
</tr>
<tr>
<td>VIL</td>
<td>Voltage on any pin</td>
<td>-0.5 to (VDD + 0.5)</td>
<td>V</td>
</tr>
<tr>
<td>TSTG</td>
<td>Storage temperature</td>
<td>-65 to +150</td>
<td>°C</td>
</tr>
<tr>
<td>PD</td>
<td>Maximum power dissipation</td>
<td>1.5</td>
<td>W</td>
</tr>
<tr>
<td>TJ</td>
<td>Maximum junction temperature</td>
<td>+175</td>
<td>°C</td>
</tr>
<tr>
<td>θJC</td>
<td>Thermal resistance, junction-to-case</td>
<td>10</td>
<td>°C/W</td>
</tr>
<tr>
<td>IIL</td>
<td>DC input current</td>
<td>±10</td>
<td>mA</td>
</tr>
</tbody>
</table>

Notes:
1. Stresses outside the listed absolute maximum ratings may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at these or any other conditions beyond limits indicated in the operational sections of this specification is not recommended. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.
2. Test per MIL-STD-883, Method 1012.
### RECOMMENDED OPERATING CONDITIONS

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>LIMITS</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{DD}$</td>
<td>Positive supply voltage</td>
<td>4.5 to 5.5</td>
<td>V</td>
</tr>
<tr>
<td>$T_C$</td>
<td>Case temperature range</td>
<td>-55 to +125</td>
<td>°C</td>
</tr>
<tr>
<td>$V_{IN}$</td>
<td>DC input voltage</td>
<td>0 to $V_{DD}$</td>
<td>V</td>
</tr>
</tbody>
</table>

### DC ELECTRICAL CHARACTERISTICS (Pre/Post-Radiation)*

$(V_{DD} = 5.0V \pm 10\%; -55^\circ C < T_C < +125^\circ C)$

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>CONDITION</th>
<th>MINIMUM</th>
<th>MAXIMUM</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{IH}$</td>
<td>High-level input voltage</td>
<td>(CMOS)</td>
<td>3.5</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>$V_{IH}$</td>
<td>High-level input voltage</td>
<td>(TTL)</td>
<td>2.2</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>$V_{IL}$</td>
<td>Low-level input voltage</td>
<td>(CMOS)</td>
<td>1.5</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>$V_{IL}$</td>
<td>Low-level input voltage</td>
<td>(TTL)</td>
<td>0.8</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>$V_{OL}$</td>
<td>Low-level output voltage</td>
<td>$I_{OL} = 200\mu A, V_{DD} = 4.5V$ (CMOS)</td>
<td>$V_{SS} + 0.05$</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>$V_{OL}$</td>
<td>Low-level output voltage</td>
<td>$I_{OL} = 4.0mA, V_{DD} = 4.5V$ (TTL)</td>
<td>0.4</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>$V_{OH}$</td>
<td>High-level output voltage</td>
<td>$I_{OH} = -200\mu A, V_{DD} = 4.5V$ (CMOS)</td>
<td>$V_{DD} - 0.05$</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>$V_{OH}$</td>
<td>High-level output voltage</td>
<td>$I_{OH} = -400\mu A, V_{DD} = 4.5V$ (TTL)</td>
<td>2.4</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>$C_{IN}$</td>
<td>Input capacitance</td>
<td>$f = 1MHz &amp; 0V, V_{DD} = 4.5V$</td>
<td>15</td>
<td></td>
<td>pF</td>
</tr>
<tr>
<td>$C_{IO}$</td>
<td>Bidirectional I/O capacitance</td>
<td>$f = 1MHz &amp; 0V, V_{DD} = 4.5V$</td>
<td>20</td>
<td></td>
<td>pF</td>
</tr>
<tr>
<td>$I_{IN}$</td>
<td>Input leakage current</td>
<td>$V_{IN} = V_{DD}$ and $V_{SS}$</td>
<td>-10</td>
<td>10</td>
<td>μA</td>
</tr>
<tr>
<td>$I_{OZ}$</td>
<td>Three-state output leakage current</td>
<td>$V_{O} = V_{DD}$ and $V_{SS}$ $V_{DD} = 5.5V$ $OE = 5.5V$</td>
<td>-10</td>
<td>10</td>
<td>μA</td>
</tr>
<tr>
<td>$I_{OS}$</td>
<td>Short-circuit output current</td>
<td>$V_{DD} = 5.5V, V_{O} = V_{DD}$ $V_{DD} = 5.5V, V_{O} = 0V$</td>
<td>-90</td>
<td>90</td>
<td>mA</td>
</tr>
<tr>
<td>$I_{DD(OP)}$</td>
<td>Supply current operating</td>
<td>CMOS inputs (i.e., $I_{OUT} = 0$) $V_{DD} = 5.5V$</td>
<td>140</td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>$I_{DD(SB)}$</td>
<td>Supply current standby</td>
<td>CMOS inputs (i.e., $I_{OUT} = 0$) $CE = V_{DD} - 0.5, V_{DD} = 5.5V$</td>
<td>100</td>
<td></td>
<td>μA</td>
</tr>
<tr>
<td>$I_{DD(SB)}$</td>
<td>Supply current standby</td>
<td>CMOS inputs (i.e., $I_{OUT} = 0$) $CE = V_{DD} - 0.5, V_{DD} = 5.5V$</td>
<td>1</td>
<td></td>
<td>mA</td>
</tr>
</tbody>
</table>

### Notes:

* Post-radiation performance guaranteed at 25°C per MIL-STD-883 Method 1019 at 1.0E6 rads(Si).

1. Measured only for initial qualification, and after process or design changes that could affect input/output capacitance.
2. Supplied as a design limit but not guaranteed or tested.
3. Not more than one output may be shorted at a time for maximum duration of one second.
4. Guaranteed by device characterization.
READ CYCLE

A combination of \( \overline{FE} \) greater than \( V_{IH} \text{(min)} \), and \( \overline{CE} \) less than \( V_{IL} \text{(max)} \) defines a read cycle. Read access time is measured from the latter of device enable, output enable, or valid address to valid data output.

An address access read is initiated by a change in address inputs while the chip is enabled with \( \overline{OE} \) asserted and \( \overline{FE} \) deasserted. Valid data appears on data output, \( DQ(7:0) \), after the specified \( t_{AVQV} \) is satisfied. Outputs remain active throughout the entire cycle. As long as device enable and output enable are active, the address inputs may change at a rate equal to the minimum read cycle time.

The chip enable-controlled access is initiated by \( \overline{CE} \) going active while \( \overline{OE} \) remains asserted, \( \overline{FE} \) remains deasserted, and the addresses remain stable for the entire cycle. After the specified \( t_{ELQV} \) is satisfied, the eight-bit word addressed by \( A(12:0) \) is accessed and appears at the data outputs \( DQ(7:0) \).

Output enable-controlled access is initiated by \( \overline{OE} \) going active while \( \overline{CE} \) is asserted, \( \overline{FE} \) is deasserted, and the addresses are stable. Read access time is \( t_{GLQV} \) unless \( t_{AVQV} \) or \( t_{ELQV} \) have not been satisfied.

AC CHARACTERISTICS READ CYCLE (Post-Radiation)*

\( V_{DD} = 5.0 \text{V} \pm 10\%; -55^\circ \text{C} < T_C < +125^\circ \text{C} \)

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>28F64-35</th>
<th>28F64-45</th>
<th>UN11</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MIN</td>
<td>MAX</td>
<td>MIN</td>
</tr>
<tr>
<td>( t_{AVAV} )</td>
<td>Read cycle time</td>
<td>35</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>( t_{AVQV} )</td>
<td>Read access time</td>
<td>35</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>( t_{AXQX} )</td>
<td>Output hold time</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>( t_{GLQX} )</td>
<td>( \overline{OE} )-controlled output enable time</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( t_{GLQV} )</td>
<td>( \overline{OE} )-controlled output enable time</td>
<td>15</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>( t_{GHQZ} )</td>
<td>( \overline{OE} )-controlled output three-state time</td>
<td>15</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>( t_{ELQX} )</td>
<td>( \overline{CE} )-controlled output enable time</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( t_{ELQV} )</td>
<td>( \overline{CE} )-controlled access time</td>
<td>35</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>( t_{EHQZ} )</td>
<td>( \overline{CE} )-controlled output three-state time</td>
<td>15</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

Notes:
* Post-radiation performance guaranteed at 25°C per MIL-STD-883 Method 1019 at 1.0E6 rads(Si).
RADIATION HARDNESS

The UT28F64 PROM incorporates special design and layout features which allow operation in high-level radiation environments. UTMC has developed special low-temperature processing techniques designed to enhance the total-dose radiation hardness of both the gate oxide and the field oxide while maintaining the circuit density and reliability. For transient radiation hardness and latchup immunity, UTMC builds all radiation-hardened products on epitaxial wafers using an advanced twin-tub CMOS process. In addition, UTMC pays special attention to power and ground distribution during the design phase, minimizing dose-rate upset caused by rail collapse.

RADIATION HARDNESS DESIGN SPECIFICATIONS

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>CONDITION</th>
<th>MINIMUM</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Dose</td>
<td>+25°C per MIL-STD-883 Method 1019</td>
<td>1.0E6</td>
<td>rads(Si)</td>
</tr>
<tr>
<td>Dose Rate Upset</td>
<td>≤ 4μs pulsewidth</td>
<td>1.0E8</td>
<td>rads(Si)/sec</td>
</tr>
<tr>
<td>Dose Rate Survival</td>
<td>20ns pulsewidth</td>
<td>1.0E10</td>
<td>rads(Si)/sec</td>
</tr>
<tr>
<td>LET Threshold</td>
<td>-55°C to +125°C</td>
<td>80</td>
<td>MeV·cm²/mg</td>
</tr>
<tr>
<td>Neutron Fluence</td>
<td>1MeV equivalent</td>
<td>1.0E14</td>
<td>n/cm²</td>
</tr>
</tbody>
</table>

Notes:
1. The PROM will not latchup during radiation exposure under recommended operating conditions.
Notes:
1. Seal ring to be electrically isolated.
2. All exposed metalized areas to be plated per MIL-I-38535.
3. Ceramic to be opaque.

Figure 3. 28-Pin 100-mil Center DIP (0.600 x 1.4)
Notes:
1. Seal ring to be electrically isolated.
2. All exposed metalized areas to be plated per MIL-I-38535.
3. Ceramic to be opaque.

Figure 4. 28-Lead 50-mil Center Flatpack (0.700 x 0.75)
Notes:
1. All exposed metalized areas are gold plated over electroplated nickel per MIL-M-38510.
2. The lid is electrically connected to Vss.
3. Lead finishes are in accordance with MIL-I-38535.
   ▲ Lead position and coplanarity are not measured.
   ▲ ID mark symbol is vendor option.
   ▲ With solder, increase maximum by 0.003.

Figure 5. 28-Lead 50-mil Center Flatpack (0.490 x 0.74)
ORDERING INFORMATION

64K PROM: Prototypes, Reduced High-Reliability, & Class S

UT 28F64

Total Dose:
(H) = 1E6 rads(Si) per MIL-STD-883, Method 1019
(R) = 1E5 rads(Si) per MIL-STD-883, Method 1019
(S) = 1E5 rads(Si) verified with an X-ray monitor
( ) = Total dose characteristics neither tested nor guaranteed

Lead Finish: 1
(A) = Hot solder dipped
(C) = Gold
(X) = Factory Option (gold or solder) 2

Screening:
(S) = Class S per MIL-STD-883, Method 5004 only (QCI, per Method 5005, may be purchased separately) 3
(V) = See note 4
(Q) = See note 4
(C) = Reduced High-Reliability flow 5
(P) = Prototype flow 6

Package Type:
(P) = 28-lead ceramic side-brazed DIP
(W) = 28-lead ceramic top-brazed flatpack (0.050" lead pitch; 0.700" x 0.750" body)
(X) = 28-lead ceramic bottom-brazed flatpack (0.050" lead pitch; 0.490" x 0.740" body)

Access Time:
(35) = 35ns access time
(45) = 45ns access time

Device Type Modifier:
(T) = TTL-compatible I/O levels
(C) = CMOS-compatible I/O levels

Notes:
1. Lead finish (A, C, or X) must be specified.
2. If an "X" is specified, part marking will match the lead finish and will be either "A" (solder) or "C" (gold).
3. Total dose must be specified. (Not available without radiation hardening.)
4. This device will be offered as a MIL-1-38535 QML Q, QML V, or SMD device. Anticipated availability is 3Q94. Please contact UTMC for the correct part number and ordering information.
5. Reduced High-Reliability flow per UTMC Manufacturing Flows Technical Description. Devices have 48 hours of burn-in and are tested at -55°C, room temperature, and 125°C. Radiation characteristics are neither tested nor guaranteed and may not be specified.
6. Prototype flow per UTMC Manufacturing Flows Technical Description. Devices have prototype assembly and are tested at 25°C only. Radiation characteristics are neither tested nor guaranteed and may not be specified. Lead finish is at UTMC's option and an "X" must be specified when ordering.
Military-standard Products
UT67164 Radiation-Hardened 8K x 8 SRAM -- SEU Hard
Data Sheet

February 1992

FEATUERS

☐ 55ns maximum address access time, single-event upset less than 1.0E-10 errors/bit-day
(-55°C to +125°C)

☐ Asynchronous operation for compatibility with industry-standard 8K x 8 SRAM

☐ TTL-compatible input and output levels

☐ Three-state bidirectional data bus

☐ Low operating and standby current

☐ Full military operating temperature range, -55°C to +125°C, screened to specific test methods listed in Table I MIL-STD-883 Method 5004 for Class S or Class B

☐ Radiation-hardened process and design; total dose irradiation testing to MIL-STD-883 Method 1019
- Total-dose: 1.0E6 rads (Si)
- Dose rate upset: 1.0E9 rads (Si)/sec
- Dose rate survival: 1.0E12 rads (Si)/sec
- Single-event upset: <1.0E-10 errors/bit-day

☐ Industry standard (JEDEC) 64K SRAM pinout

☐ Packaging options:
- 28-pin 100-mil center DIP (.600 x 1.2)
- 28-pin 50-mil center flatpack (.700 x .75)

☐ 5-volt operation

☐ Post-radiation AC/DC performance characteristics guaranteed by MIL-STD-883 Method 1019 testing at 1.0E6 rads(Si)

INTRODUCTION

The UT67164 SRAM is a high performance, asynchronous, radiation-hardened, 8K x 8 random access memory conforming to industry-standard fit, form, and function. The UT67164 SRAM features fully static operation requiring no external clocks or timing strobes. UTMC designed and implemented the UT67164 SRAM using an advanced radiation-hardened twin-well CMOS process. Advanced CMOS processing along with a device enable/disable function result in a high performance, power-saving SRAM. The combination of radiation-hardness, fast access time, and low power consumption make UT67164 ideal for high-speed systems designed for operation in radiation environments.

PIN NAMES

<table>
<thead>
<tr>
<th>A(12:0)</th>
<th>Address</th>
<th>W</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQ(7:0)</td>
<td>Data Input/Output</td>
<td>G</td>
<td>Output Enable</td>
</tr>
<tr>
<td>E1</td>
<td>Enable 1</td>
<td>VDD</td>
<td>Power</td>
</tr>
<tr>
<td>E2</td>
<td>Enable 2</td>
<td>VSS</td>
<td>Ground</td>
</tr>
</tbody>
</table>

Figure 1. SRAM Block Diagram

Figure 2. SRAM Pinout
DEVICE OPERATION

The UT67164 has four control inputs called Enable 1 (E1), Enable 2 (E2), Write Enable (W), and Output Enable (G); thirteen address inputs, A(12:0); and eight bidirectional data lines, DQ(7:0). E1 and E2 are device enable inputs that control device selection, active, and standby modes. Asserting both E1 and E2 enables the device, causes IDD to rise to its active value, and decodes the thirteen address inputs to select one of 8,192 words in the memory. W controls read and write operations. During a read cycle, G must be asserted to enable the outputs.

Figure 3c shows Read Cycle 3, the Output Enable-controlled Access. For this cycle, E1 and E2 are asserted, W is deasserted, and the addresses are stable before G is enabled. Read access time is tAVQV unless tAVQV or tETQV have not been satisfied.

WRITE CYCLE

A combination of W less than VIL(max), E1 less than VIL(max), and E2 greater than VIL(min) defines a write cycle. The state of G is a "don't care" for a write cycle. The outputs are placed in the high-impedance state when either G is greater than VIL(min), or when W is less than VIL(max).

Write Cycle 1, the Write Enable-controlled Access shown in figure 4a, is defined by a write terminated by W going high, with E1 and E2 still active. The write pulse width is defined by tWLWH when the write is initiated by W, and by tETWH when the write is initiated by the latter of E1 or E2. Unless the outputs have been previously placed in the high-impedance state by G, the user must wait tWLOZ before applying data to the eight bidirectional pins DQ(7:0) to avoid bus contention.

Write Cycle 2, the Chip Enable-controlled Access shown in figure 4b, is defined by a write terminated by the latter of E1 or E2 going inactive. The write pulse width is defined by tWLRF when the write is initiated by W, and by tETRF when the write is initiated by the latter of E1 or E2 going active. For the W initiated write, unless the outputs have been previously placed in the high-impedance state by G, the user must wait tWLOZ before applying data to the eight bidirectional pins DQ(7:0) to avoid bus contention.

RADIATION HARDNESS

The UT67164 SRAM incorporates special design and layout features which allow operation in high-level radiation environments. UTMC has developed special low-temperature processing techniques designed to enhance the total-dose radiation hardness of both the gate oxide and the field oxide while maintaining the circuit density and reliability. For transient radiation hardness and latchup immunity, UTMC builds all radiation-hardened products on epitaxial wafers using an advanced twin-tub CMOS process. In addition, UTMC pays special attention to power and ground distribution during the design phase, minimizing dose-rate upset caused by rail collapse.
### Table 2. Radiation Hardness Design Specifications

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Dose</td>
<td>1.0E6 rads(Si)</td>
</tr>
<tr>
<td>Dose Rate Upset</td>
<td>1.0E9 rads(Si)/20ns pulse</td>
</tr>
<tr>
<td>Dose Rate Survival</td>
<td>1.0E12 rads(Si)/20ns pulse</td>
</tr>
<tr>
<td>Single-Event Upset</td>
<td>1.0E-10 errors/bit-day</td>
</tr>
<tr>
<td>Neutron Fluence</td>
<td>3.0E14 n/cm²</td>
</tr>
</tbody>
</table>

**Notes:**
1. The SRAM will not latch up during radiation exposure under recommended operating conditions.
2. 90% Adam's worst case spectrum (-55°C to +125°C).

### Table 3. SEU versus Temperature

![Graph showing SEU versus Temperature](image)

### Absolute Maximum Ratings
(Referenced to VSS)

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>LIMITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDD</td>
<td>DC supply voltage</td>
<td>-0.5 to 7.0</td>
</tr>
<tr>
<td>VIO</td>
<td>Voltage on any pin</td>
<td>-0.5 to VDD + 0.5</td>
</tr>
<tr>
<td>TSTG</td>
<td>Storage temperature</td>
<td>-65 to +150°C</td>
</tr>
<tr>
<td>PD</td>
<td>Maximum power dissipation</td>
<td>1.0 W</td>
</tr>
<tr>
<td>TJ</td>
<td>Maximum junction temperature</td>
<td>+150°C</td>
</tr>
<tr>
<td>TCJ</td>
<td>Thermal resistance, junction-to-case</td>
<td>10°C/W</td>
</tr>
<tr>
<td>ILU</td>
<td>Latchup immunity (see figure 6b)</td>
<td>+/- 150 mA</td>
</tr>
<tr>
<td>I1</td>
<td>DC input current</td>
<td>+/- 10 mA</td>
</tr>
</tbody>
</table>

**Notes:**
1. Stresses outside the listed absolute maximum ratings may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at these or any other conditions beyond limits indicated in the operational sections of this specification is not recommended. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.
2. Test per MIL-STD-883, Method 1012.

### Recommended Operating Conditions

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>LIMITS</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDD</td>
<td>Positive supply voltage</td>
<td>4.5 to 5.5</td>
<td>V</td>
</tr>
<tr>
<td>TC</td>
<td>Case temperature range</td>
<td>-55 to +125</td>
<td>°C</td>
</tr>
<tr>
<td>VIN</td>
<td>DC input voltage</td>
<td>0 to VDD</td>
<td>V</td>
</tr>
</tbody>
</table>
DC ELECTRICAL CHARACTERISTICS (Post-Radiation)*
(VDD = 5.0V +/-10%; -55°C < Tc < +125°C)

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>CONDITION</th>
<th>MINIMUM</th>
<th>MAXIMUM</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIL</td>
<td>Low-level input voltage</td>
<td></td>
<td>0.8</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>VIL</td>
<td>High-level input voltage</td>
<td></td>
<td>2.2</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>VIL</td>
<td>Low-level output voltage</td>
<td>IOL = +/- 4.0 mA, VDD = 4.5V</td>
<td>0.4</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>VOH</td>
<td>High-level output voltage</td>
<td>IOH = +/- 4.0mA, VDD = 4.5V</td>
<td>2.4</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>CIN</td>
<td>Input capacitance 1</td>
<td>F = 1MHz @ 0V, VDD = 4.5V</td>
<td>15</td>
<td></td>
<td>pF</td>
</tr>
<tr>
<td>CO</td>
<td>Bidirectional I/O capacitance</td>
<td>F = 1MHz @ 0V, VDD = 4.5V</td>
<td>20</td>
<td></td>
<td>pF</td>
</tr>
<tr>
<td>IN</td>
<td>Input leakage current</td>
<td>V IN = V DD and V SS</td>
<td>-10</td>
<td>10</td>
<td>μA</td>
</tr>
<tr>
<td>IOZ</td>
<td>Three-state output leakage current</td>
<td>V O = V DD and V SS</td>
<td>-10</td>
<td>10</td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td>TTL outputs</td>
<td>V DD = 5.5V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IOS</td>
<td>Short-circuit output current 2</td>
<td>V DD = 5.5V, V O = V DD</td>
<td>-90</td>
<td>90</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>V DD = 5.5V, V O = 0V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDD(OP)</td>
<td>Supply current operating @1MHz</td>
<td>CMOS inputs (i.e., IOUT = 0)</td>
<td>40</td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>IDD(SB)</td>
<td>pre-rad</td>
<td>V DD = 5.5V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Supply current standby</td>
<td>CMOS inputs (i.e., IOUT = 0)</td>
<td>200</td>
<td></td>
<td>μA</td>
</tr>
<tr>
<td></td>
<td>E1 = V DD - 0.5 V DD = 5.5 V</td>
<td>E2 = V SS + 0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>post-rad</td>
<td>CMOS inputs (i.e., IOUT = 0)</td>
<td>3</td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td>@ f = 0 Hz</td>
<td>CS1 =negated V DD = 5.5 V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CS2 =negated</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes:
1. Measured only for initial qualification, and after process or design changes that could affect input/output capacitance.
2. Supplied as a design limit but not guaranteed or tested.
3. Not more than one output may be shorted at a time for maximum duration of one second.
* Post-radiation performance guaranteed at 25°C per MIL-STD-883 Method 1019 at 1.0E6 rads(Si).

AC CHARACTERISTICS READ CYCLE (Post-Radiation)*
(VDD = 5.0V +/-10%; -55°C < Tc < +125°C)

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>67164-85</th>
<th>67164-70</th>
<th>67164-55</th>
<th>UIT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MIN</td>
<td>MAX</td>
<td>MIN</td>
<td>MAX</td>
</tr>
<tr>
<td>tvAV</td>
<td>Read cycle time</td>
<td>85</td>
<td>70</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tvAVQ</td>
<td>Read access time</td>
<td>85</td>
<td>70</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tvAQQ</td>
<td>Output hold time</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tvGQ</td>
<td>G-controlled output enable time</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>tvGQ</td>
<td>G-controlled output enable time (Read Cycle 3)</td>
<td>30</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>tvHQ</td>
<td>G-controlled output three-state time</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>tvEQ</td>
<td>E-controlled output enable time</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>tvEQ</td>
<td>E-controlled access time</td>
<td>85</td>
<td>70</td>
<td>55</td>
<td>55</td>
</tr>
<tr>
<td>tvEQ</td>
<td>E-controlled output three-state time</td>
<td>25</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

Notes:
1. The ET (enable true) notation refers to the rising edge of E2 or the falling edge of ET, whichever comes last. SEU immunity does not affect the read parameters.
2. The EF (enable false) notation refers to the falling edge of E2 or the rising edge of E1, whichever comes first. SEU immunity does not affect the read parameters.
3. Three-state is defined as a 500mV change from steady-state output voltage.
* Post-radiation performance guaranteed at 25°C per MIL-STD-883 Method 1019 at 1.0E6 rads(Si).
Figure 3a. SRAM Read Cycle 1: Address Access

Figure 3b. SRAM Read Cycle 2: Chip Enable Access

Figure 3c. SRAM Read Cycle 3: Output Enable Access
AC CHARACTERISTICS WRITE CYCLE (Post-Radiation)*
(VDD = 5.0V +/-10%; -55°C < Tc < +125°C)

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>67144-85</th>
<th>67164-70</th>
<th>67164-55</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>tAVAV</td>
<td>Write cycle time</td>
<td>85</td>
<td>70</td>
<td>55</td>
<td>ns</td>
</tr>
<tr>
<td>tETWH</td>
<td>Device enable to end of write</td>
<td>65</td>
<td>60</td>
<td>50</td>
<td>ns</td>
</tr>
<tr>
<td>tAVET</td>
<td>Address setup time for write (Et or E2 - controlled)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>ns</td>
</tr>
<tr>
<td>tAVWL</td>
<td>Address setup time for write (W - controlled)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>ns</td>
</tr>
<tr>
<td>tWLWH</td>
<td>Write pulse width</td>
<td>50</td>
<td>35</td>
<td>35</td>
<td>ns</td>
</tr>
<tr>
<td>tWHAX</td>
<td>Address hold time for write (W - controlled)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>ns</td>
</tr>
<tr>
<td>tEFAX</td>
<td>Address hold time for device enable (Et or E2 - controlled)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>ns</td>
</tr>
<tr>
<td>tWLOQZ</td>
<td>W-controlled three-state time</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>ns</td>
</tr>
<tr>
<td>tWHQX</td>
<td>W-controlled output enable time</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>ns</td>
</tr>
<tr>
<td>tETEF</td>
<td>Device enable pulse width (Et or E2 - controlled)</td>
<td>65</td>
<td>60</td>
<td>50</td>
<td>ns</td>
</tr>
<tr>
<td>tDVWH</td>
<td>Data setup time</td>
<td>50</td>
<td>35</td>
<td>35</td>
<td>ns</td>
</tr>
<tr>
<td>tWHDX</td>
<td>Data hold time</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>ns</td>
</tr>
<tr>
<td>tWLEF</td>
<td>Device enable controlled write pulse width</td>
<td>65</td>
<td>60</td>
<td>50</td>
<td>ns</td>
</tr>
<tr>
<td>tDVEF</td>
<td>Data setup time</td>
<td>50</td>
<td>35</td>
<td>35</td>
<td>ns</td>
</tr>
<tr>
<td>tEFDX</td>
<td>Data hold time</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>ns</td>
</tr>
</tbody>
</table>

* Post-radiation performance guaranteed at 25°C per MIL-STD-883 Method 1019 at 1.0E6 rads(Si).

A(12:0)

- E2
- Et
- W

Q(7:0)

D(7:0)

Assumptions:
1) $\bar{G} \leq V_{IL} (\text{min})$. If $\bar{G} \geq V_{IH} (\text{min})$ then Q(7:0) will be in three-state for the entire cycle.

Figure 4a. SRAM Write Cycle 1: W-Controlled Access
Assumptions & Notes:
1) $O \leq V_H$ (max). If $O \geq V_H$ (min) then $Q(7:0)$ will be in three-state for the entire cycle.
2) Either $E1/E2$ scenario above can occur.
3) If $E1$ or $E2$ is asserted simultaneously with or after the $W$ low transition, the outputs will remain in a high-impedance state.

$\text{t}_{\text{WLEF}}$, $\text{t}_{\text{DVEF}}$, $\text{t}_{\text{EFAX}}$

Figure 4b. SRAM Write Cycle 2: Enable-Controlled Access
### DATA RETENTION CHARACTERISTICS (Post-Radiation)*

\( (T_C = 25^\circ C) \)

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PARAMETER</th>
<th>MINIMUM</th>
<th>MAXIMUM VDD @ 2.0V</th>
<th>MAXIMUM VDD @ 3.0V</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDR</td>
<td>VDD for data retention</td>
<td>2.0</td>
<td>--</td>
<td>--</td>
<td>V</td>
</tr>
<tr>
<td>IDD</td>
<td>Data retention current (^1)</td>
<td>--</td>
<td>75</td>
<td>90</td>
<td>µA</td>
</tr>
<tr>
<td>tEFR</td>
<td>Chip deselect to data retention time (^1)</td>
<td>0</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>tR</td>
<td>Operation recovery time (^1)</td>
<td>( t_{AVAV} )</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
</tbody>
</table>

Note:\(^\):
1. \( V_{LC} = 0.2V \)
2. \( V_{HC} = V_{DD} - 0.2V \)
3. \( E1 \geq V_{HC}, E2 > V_{HC} \)

* Post-radiation performance guaranteed at 25°C per MIL-STD-883 Method 1019 at 1.0E6 rads(Si).

---

**Figure 5. Low \( V_{DD} \) Data Retention Waveform**

---

\( DATA\ RETENTION\ MODE \)

\( V_{DD} \)

\( V_{DR} \geq 2 \text{ V} \)

\( 4.5\text{ V} \)

\( t_{EFR} \)

\( V_{IH} \)

\( t_{R} \)

\( V_{IH} \)
LATCHUP TEST CONFIGURATION

Figure 6b shows the latchup test. $V_{DD}$ holds at +5.5 VDC and $V_{SS}$ holds at ground. The device test is at 125°C. Each type of I/O alternately receives a positive and then negative 150 mA pulse of 500 ms duration. The current is monitored after the pulse for latchup condition. To prevent burnout, the supply current is limited to 400 mA.

The SRAM has latchup immunity in excess of +150 mA for 500 ms.
Notes:
1. Seal ring to be electrically isolated.
2. All exposed metalized areas to be plated per MIL-M-38510.
3. Ceramic to be opaque.

Figure 7a. 28-pin Ceramic Flatpack

Notes:
1. Seal ring to be electrically isolated.
2. All exposed metalized areas to be plated per MIL-M-38510.
3. Ceramic to be opaque.

Figure 7b. 28-pin Ceramic DIP Package
ORDERING INFORMATION

To order the UT67164 SRAM, use the following part number guide:

UT67164 ** * *

(R) Radiation-Hardened 1.0E5 rads(Si)
(H) Radiation-Hardened 1.0E6 rads(Si)
(B) MIL-STD-883
(C) Modified high-reliability flow, -55°C to 125°C
(P) Prototype
(radiation performance not guaranteed)
(S) Screened to Class S per UTM Co's standard flow
(contact factory for availability)
(P) 28-pin Ceramic Side-Brazed DIP
(CDIP2-T28, MIL-STD-1835)
(W) 28-pin Ceramic Flatpack .050 center
(85) 85ns Access Time
(70) 70ns Access Time
(55) 55ns Access Time
General Description
The MAX220-MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, and in particular, for those applications where ±12V is not available. These parts are particularly useful in battery-powered systems since their low-power shutdown mode reduces power dissipation to less than 50mW. The MAX225, MAX235, and MAX245-MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications
- Portable Computers
- Low-Power Modems
- Interface Translation
- Battery-Powered RS-232 Systems
- Multi-Drop RS-232 Networks

Features
- Superior to Bipolar
- Operate from Single +5V Power Supply (+5V and ±12V—MAX231 and MAX239)
- Low-Power Receive Mode in Shutdown (MAX223/MAX242)
- Meet All EIA/TIA-232E and V.28 Specifications
- Multiple Drivers and Receivers
- 3-State Driver and Receiver Outputs
- Open-Line Detection (MAX243)

Ordering Information

<table>
<thead>
<tr>
<th>PART</th>
<th>TEMP. RANGE</th>
<th>PIN-PACKAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX220</td>
<td>0°C to +70°C</td>
<td>16-Pin SO</td>
</tr>
<tr>
<td>MAX230</td>
<td>0°C to +70°C</td>
<td>16-Wide SO</td>
</tr>
<tr>
<td>MAX225</td>
<td>0°C to +70°C</td>
<td>16-Wide SO</td>
</tr>
<tr>
<td>MAX235</td>
<td>0°C to +70°C</td>
<td>16-Wide SO</td>
</tr>
<tr>
<td>MAX227</td>
<td>0°C to +70°C</td>
<td>16-Wide SO</td>
</tr>
<tr>
<td>MAX241</td>
<td>0°C to +70°C</td>
<td>16-Wide SO</td>
</tr>
<tr>
<td>MAX242</td>
<td>0°C to +70°C</td>
<td>16-Wide SO</td>
</tr>
</tbody>
</table>

Ordering information continued on last page.

Selection Table

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Power Supply</th>
<th>No. of RS-232 Drivers</th>
<th>No. of Cap. Values</th>
<th>No.</th>
<th>Act. +2V</th>
<th>Data Rate</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX220</td>
<td>±5V</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX230</td>
<td>±5V</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX225</td>
<td>±5V</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX235</td>
<td>±5V</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX227</td>
<td>±5V</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX241</td>
<td>±5V</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX242</td>
<td>±5V</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MAXIM Integrated Products

Call toll free 1-800-998-8800 for free samples or literature.
### ELECTRICAL CHARACTERISTICS

**MAX220/222/232A/233A/242/243**

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>CONDITIONS</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN Input Threshold Low</td>
<td>1.4</td>
<td>0.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EN Input Threshold High</td>
<td>2.0</td>
<td>1.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating Supply Voltage</td>
<td>4.5</td>
<td>4.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vcc Supply Current (SHDN = 1)</td>
<td>4.5</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shutdown Supply Current</td>
<td>10</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHDN Input Leakage Current</td>
<td>5</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHDN Threshold Low</td>
<td>1.5</td>
<td>0.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHDN Threshold High</td>
<td>1.5</td>
<td>0.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transition Slew Rate</td>
<td>500</td>
<td>36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transmitter Propagation Delay (TLL to RS-232) (Normal Operation)</td>
<td>4.5</td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receiver Propagation Delay (RC-232 to TLL) (Normal Operation)</td>
<td>4.5</td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receiver-TX Output Enable Time, Figure 3</td>
<td>4.5</td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receiver-TX Output Disable Time, Figure 3</td>
<td>4.5</td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receiver-TX Output Enable Time (SHDN goes High), Figure 4</td>
<td>4.5</td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transmitter-TX Output Disable Time (SHDN goes Low), Figure 4</td>
<td>4.5</td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receiver-TX Propagation Delay Difference (Normal Operation)</td>
<td>4.5</td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receiver-TX Propagation Delay Difference (Normal Operation)</td>
<td>4.5</td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** MAX243 RO4 is guaranteed to be low when RO4 is 0V or in RO4H.
+5V-Powered, Multi-Channel RS-232 Drivers/Receivers

**ELECTRICAL CHARACTERISTICS — MAX223/MAX230-MAX241 (continued)**

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>CONDITIONS</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS-232 Input Threshold Low</td>
<td>TA = +25°C, VCC = 5V</td>
<td>0.6</td>
<td>1.2</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>RS-232 Input Threshold High</td>
<td>TA = +70°C, VCC = 5V</td>
<td>0.6</td>
<td>1.5</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>RS-232 Input Hysteresis</td>
<td></td>
<td>0.2</td>
<td>0.5</td>
<td>1.0</td>
<td>V</td>
</tr>
<tr>
<td>RS-232 Input Resistance</td>
<td>TA = +25°C, VCC = 5V</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>KΩ</td>
</tr>
<tr>
<td>TTLCMOS Output Voltage Low</td>
<td></td>
<td>0.4</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>TTLCMOS Output Leakage Current</td>
<td>IOUT = 1.6mA (MAX223-233). OUT = 3.2mA</td>
<td>3.5</td>
<td></td>
<td></td>
<td>μA</td>
</tr>
<tr>
<td>Receiver Output Enable Time</td>
<td>Normal operation</td>
<td>600</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Receiver Output Disable Time</td>
<td>Normal operation</td>
<td>900</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Propagation Delay</td>
<td>RS-232 IN to TTLCMOS</td>
<td>0.5</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td>OUT, CL = 10pF</td>
<td>4</td>
<td>10</td>
<td></td>
<td>pS</td>
</tr>
<tr>
<td>Transition Region Slope Rate</td>
<td>MAX223/MAX223/MAX233/MAX241</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>TA = +35°C, VCC = 5V, RL = 4kΩ to 7kΩ, CL = 50pF to 2500pF, measured from ±3V to ±3V vs. +5V to +5V</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAX231/MAX232/MAX233</td>
<td>4</td>
<td></td>
<td></td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>TA = +35°C, VCC = 5V, RL = 4kΩ to 7kΩ, CL = 50pF to 2500pF, measured from ±3V to ±3V or ±3V to ±3V</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transmitter Output Resistance</td>
<td>VCC = 5V, Vout = ±1V, Yout = ±2V</td>
<td>300</td>
<td></td>
<td></td>
<td>Ohm</td>
</tr>
<tr>
<td>Transmitter Output Short-Circuit Current</td>
<td></td>
<td>±10</td>
<td></td>
<td></td>
<td>mA</td>
</tr>
</tbody>
</table>
# +5V-Powered, Multi-Channel RS-232 Drivers/ Receivers

## ELECTRICAL CHARACTERISTICS—MAX225/MAX244-MAX249

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>CONDITIONS</th>
<th>MAX225</th>
<th>MAX244-MAX249</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage</td>
<td></td>
<td>4.75</td>
<td>5.25</td>
</tr>
<tr>
<td>Current</td>
<td></td>
<td>4.5</td>
<td>5.5</td>
</tr>
</tbody>
</table>

### POWER SUPPLY AND CONTROL LOGIC

- **Power Supply Voltage:** No Load
  - No Load: MAX225 (10 mA), MAX244-MAX249 (11 mA)
- **Shutdown/Enable Current:**
  - MAX225: 1.3 mA
  - MAX244-MAX249: 1.25 mA
- **Leakage Current:** 50 μA

### AC CHARACTERISTICS

- **MDT / Propagation Delay Difference:**
  - MAX225: 150 ns
  - MAX244-MAX249: 100 ns

### Performance Notes:
- The 3000 minimum specified value is for MAX225/MAX244-MAX249. The actual minimum value may vary depending on the specific application and operating conditions.

---

## +5V-Powered, Multi-Channel RS-232 Drivers/ Receivers

## ELECTRICAL CHARACTERISTICS—MAX225/MAX244-MAX249 (continued)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>CONDITIONS</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage</td>
<td></td>
<td>3.0</td>
<td>5.0</td>
<td>5.5</td>
<td>V</td>
</tr>
<tr>
<td>Current</td>
<td></td>
<td>4.0</td>
<td>10</td>
<td></td>
<td>mA</td>
</tr>
</tbody>
</table>

### Power Supply and Control Logic

- **Power Supply Voltage:**
  - No Load: MAX225 (10 mA), MAX244-MAX249 (11 mA)
- **Shutdown/Enable Current:**
  - MAX225: 1.3 mA
  - MAX244-MAX249: 1.25 mA
- **Leakage Current:** 50 μA

### AC Characteristics

- **MDT / Propagation Delay Difference:**
  - MAX225: 150 ns
  - MAX244-MAX249: 100 ns

### Performance Notes:
- The 3000 minimum specified value is for MAX225/MAX244-MAX249. The actual minimum value may vary depending on the specific application and operating conditions.
# MAX220-MAX249

## Table 1a. MAX225 Control Pin Configurations

<table>
<thead>
<tr>
<th>ENA</th>
<th>ENR</th>
<th>OPERATION STATUS</th>
<th>TRANSMITTERS</th>
<th>RECEIVERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Normal Operation</td>
<td>All Active</td>
<td>All Active</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Normal Operation</td>
<td>All Active</td>
<td>All 3-State</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Shutdown</td>
<td>All 3-State</td>
<td>All Low Power, Receiver Mode</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Shutdown</td>
<td>All 3-State</td>
<td>All 3-State</td>
</tr>
</tbody>
</table>

## Table 1b. MAX245 Control Pin Configurations

<table>
<thead>
<tr>
<th>ENA</th>
<th>ENR</th>
<th>OPERATION STATUS</th>
<th>TRANSMITTERS</th>
<th>RECEIVERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Normal Operation</td>
<td>All Active</td>
<td>All Active</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Normal Operation</td>
<td>All Active</td>
<td>All 3-State</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Shutdown</td>
<td>All 3-State</td>
<td>All Low Power, Receiver Mode</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Shutdown</td>
<td>All 3-State</td>
<td>All 3-State</td>
</tr>
</tbody>
</table>

## Table 1c. MAX246 Control Pin Configurations

<table>
<thead>
<tr>
<th>ENA</th>
<th>ENR</th>
<th>OPERATION STATUS</th>
<th>TRANSMITTERS</th>
<th>RECEIVERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Normal Operation</td>
<td>All Active</td>
<td>All Active</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Normal Operation</td>
<td>All Active</td>
<td>All 3-State</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Shutdown</td>
<td>All 3-State</td>
<td>All Low Power, Receiver Mode</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Shutdown</td>
<td>All 3-State</td>
<td>All 3-State</td>
</tr>
</tbody>
</table>

## Table 1d. MAX247/248/249 Control Pin Configurations

<table>
<thead>
<tr>
<th>ENRA</th>
<th>ENRB</th>
<th>OPERATION STATUS</th>
<th>TRANSMITTERS</th>
<th>RECEIVERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Normal Operation</td>
<td>All Active</td>
<td>All Active</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Normal Operation</td>
<td>All Active</td>
<td>All 3-State</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Normal Operation</td>
<td>All 3-State</td>
<td>All Low Power, Receiver Mode</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Normal Operation</td>
<td>All 3-State</td>
<td>All 3-State</td>
</tr>
</tbody>
</table>
Detailed Description

The MAX220/MAX249 contain four sections: dual charge-pump DC-DC voltage converters, RS-232 drivers, and receiver and transmitter enable control inputs.

Dual Charge-Pump Voltage Converter

The MAX220/MAX249 have two internal charge-pumps that convert +5V to +10V (unregulated) for RS-232 driver operation. The first converter uses capacitor C1 to double the +5V input to +10V on C3 at the V+ output. The second converter uses capacitor C2 to invert +10V to -5V on C4 at the V- output.

A small amount of power may be drawn from the +10V (V+) and -10V (V-) outputs to power external circuitry (see Typical Operating Characteristics), except on the MAX225 and MAX245/MAX247, where these pins are not available. V+ and V- are not regulated, so the output voltage drops with increasing load current. Do not load V+ and V- to a point that violates the minimum +5V voltage for the driver output voltage when sourcing current from V+ and V- to external circuitry.

When using the shutdown feature in the MAX222, MAX225, MAX235, MAX236, MAX240, MAX241 and MAX245-MAX249 avoid using V+ and V- to power external circuitry. When these pins are shut down, V+ falls to 0V, and V- falls to -5V. For applications where a +10V external supply is applied to the V+ pin (instead of using the internal charge-pump to generate +10V), the C1 capacitor must not be installed and the STDBY pin must be tied to VCC. This is because V+ is internally connected to VCC in shutdown mode.

RS-232 Drivers

The typical driver output voltage swing is ±5V when loaded with a nominal 5kΩ RS-232 receiver and VCC = +5V. Output swing is guaranteed to meet the EIA/TIA-232 and V.28 specifications, that calls for ±5V minimum driver output levels under worst-case conditions. These include: ±3kΩ load, VCC = +5V, ±5V, and 10Mbps operating temperature. Unloaded driver output voltage ranges from ±5V (V+0.5V) to ±5V (V+0.5V).

Input thresholds are both TTL and CMOS compatible. The inputs of unused drivers can be left unconnected since 400kΩ input resistors to VCC are built in. The pull-up resistors force the outputs of unused drivers low because all drivers invert. The internal input pull-up resistor typically source 10μA, except in shutdown, where the pull-ups are disabled. Driver outputs turn off and enter a high-impedance state where leakage current is typically micromilliamps (maximum 25μA)—when in shutdown, in three-state mode, or when device power is removed. Outputs can be driven to ±5V. The power-supply current typically drops to 8μA in shutdown mode.

The MAX220 has a receiver 3-state control line, and the MAX225, MAX235, MAX236, MAX240, and MAX241 have both a receiver 3-state control line and a low-power shutdown control. The receiver shutdown is specified in the data sheet. Normally, the MAX243's receiver (1.5V threshold) is used for this line (T0 or RD), while the negative shutdown threshold receiver is connected to the 3-state control (DHR, D1H, CTS, RTS, etc.).

Other members of the MAX243 family implement the optional cable fault protection as specified by EIA/TIA-232 specifications. This means a receiver output goes high whenever its input is driven negative, left floating, or shorted to ground. The high output tells the serial communication circuit to stop sending data. To avoid this, the control logic must either be driven or connected with pull-up to an appropriate positive voltage level.

RS-232 Receivers

EIA/TIA-232 and V.28 specifications define a voltage level greater than 3V as a logic 0, so all receivers invert. Input thresholds are set at 0.8V and 2.4V, so receivers respond to TTL level inputs as well as EIA/TIA-232 and V.28 level inputs. The receiver inputs withstand an input overvoltage up to ±2.5V and provide input terminating resistors with a nominal 3kΩ value. The receivers implement Type 1 signalization of the fault conditions of V.28 and EIA/TIA-232.

The receiver input hysteresis is typically 0.5V with a guaranteed minimum of 0.2V. The receiver accepts data from a transmissions with slow-moving input signals, even with moderate amounts of noise and ringing. The receiver propagation delay is typically 0.25ns and is independent of input swing direction.

Low-Power Receive Mode

The low-power receive mode feature of the MAX223, MAX225, MAX235, MAX236, MAX240, and MAX241 puts the IC into a low-power receive mode with a reduced power mode when the chip is in shutdown. Under these conditions, the propagation delay is increased to about 2.5μs for a high-to-low transition.

When in shutdown, the receiver acts as a CMOS inverter with 2.5μs rise time. The MAX243 and MAX247 also have a receiver 3-state enable input (EN) that allows receiver output control independent of STDBY. With both of these devices, STDBY disables the receiver outputs.

MAX222 and MAX242

The MAX222 and MAX242 provide five transmitters and two receivers, where the MAX245 provides ten receivers and eight transmit lines. Both devices have variable transmitter levels, and transmitter gain and variable transmitters can be added to the basic circuit and the device is capable of delivering a logic high on any transmit line.

Applications Information

The MAX222 and MAX242 feature transmitters and receiver enable control inputs.

The receivers have three modes of operation: full-speed reception (RS-232), half-speed reception (DCE devices), and low-speed reception (parallel and serial devices). The low-speed receiver (1.5V threshold) is used for this line (T0 or RD), while the negative shutdown threshold receiver is connected to the 3-state control (DHR, D1H, CTS, RTS, etc.).

The transmitter enables a receive or three-state mode. The transmitter outputs control the shutdown mode. The device also supports Shutdown mode when all transmitters are disabled. The transmitter enable circuitry contains a low-power receive mode when in shutdown.

Table 1A-10 defines the control states. The MAX244 has an enable control pin and is not included in these tables.

The MAX246 has ten receivers and eight drivers with current capability, each controlling one side of the device. A logic high at the A-side control input (ENA) causes the four A-side receivers and drivers to go into a three-state mode. Similarly, the B-side control input (ENB) causes the four B-side receivers and drivers to go into a three-state mode. All eight receivers and drivers can be driven to a three-state mode, and ENB (or ENA) must be active to drive signals on either A-side or B-side.

The MAX248 provides nine receivers and eight drivers with transmitter control pins. The ENR and ENB receiver enables each control four receiver outputs. The ENR and ENB transmitter enable inputs each control four transmitter outputs. The nine receivers and eight drivers are driven by the ENR and ENB inputs, which are also active-low.

The MAX248 provides eight transmitters and eight drivers with transmitter control pins. These enable inputs each drive four transmitter outputs. The MAX249 provides nine transmitters and eight drivers with transmitter control pins. These enable inputs each drive four transmitter outputs. The MAX249 provides a three-state control pin. The ENR and ENB receiver enable inputs each control four receiver outputs. The ENR and ENB transmitter enable inputs each control four transmitter outputs. The MAX249 provides a three-state control pin. The ENR and ENB receiver enable inputs each control four receiver outputs. The ENR and ENB transmitter enable inputs each control four transmitter outputs.

The MAX249 provides a three-state control pin. The ENR and ENB receiver enable inputs each control four receiver outputs. The ENR and ENB transmitter enable inputs each control four transmitter outputs. The device can be driven to a three-state mode with either A-side or B-side control inputs active.
+5V-Powered, Multi-Channel RS-232 Drivers/ Receivers

**Figure 5** MAX220/232/232A Pin Configuration and Typical Operating Circuit

**Figure 6** MAX220/232/232A Pin Configuration and Typical Operating Circuit

MAXIM

**CAPACITANCE (pF)**

<table>
<thead>
<tr>
<th>DEVICE</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX220</td>
<td>4.7</td>
<td>4.7</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX232</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>MAX232A</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>
+5V-Powered, Multi-Channel RS-232 Drivers/ Receivers

MAX220-MA249

Figure 14. MAX227 Pin Configuration and Typical Operating Circuit

Figure 15. MAX228 Pin Configuration and Typical Operating Circuit
Figure 18: MAX241, MAX223 Pin Configuration and Typical Operating Circuit
+5V-Powered, Multi-Channel RS-232 Drivers/Receivers

**MAX220–MAX249**

**MAX241 Functional Description**
10 Receivers Always Active
5 A-Side Receivers
5 B-Side Receivers
8 Transmitters
4 A-Side Transmitters
4 B-Side Transmitters
No Control Pins

Figure 20: MAX244 Pin Configuration and Typical Operating Circuit

**MAX245 Functional Description**
10 Receivers
5 A-Side Receivers (8 A, always active)
5 B-Side Receivers (8 B, always active)
8 Transmitters
4 A-Side Transmitters
4 B-Side Transmitters
2 Control Pins
1 Receiver Enable (REN)
1 Transmitter Enable (ENT)

Figure 21: MAX245 Pin Configuration and Typical Operating Circuit
+5V-Powered, Multi-Channel RS-232 Drivers/Receivers

MAX220-MAX249

MAX246 Functional Description
10 Receivers
5 A-Side Receivers (RAS always active)
5 B-Side Receivers (RAS always active)
8 Transmitters
4 A-Side Transmitters
4 B-Side Transmitters
2 Control Pins
Enable A-Side (ENA)
Enable B-Side (ENB)

Figure 22. MAX246 Pin Configuration and Typical Operating Circuit

MAX247 Functional Description
10 Receivers
5 A-Side Receivers (RAS always active)
5 B-Side Receivers (RAS always active)
8 Transmitters
4 A-Side Transmitters
4 B-Side Transmitters
2 Power Pins
Enable Receiver A-Side (ENA)
Enable Receiver B-Side (ENB)
Enable Transmitter A-Side (ENTA)
Enable Transmitter B-Side (ENTB)
+5V-Powered, Multi-Channel RS-232 Drivers/Receivers

MAX220–MAX249

MAX248 Functional Description
8 Receivers
4 A-Side Receivers
4 B-Side Receivers
8 Transmitters
4 A-Side Transmitters
4 B-Side Transmitters
4 Control Pins
Enable Receiver A-Side (ENRA)
Enable Receiver B-Side (ENRB)
Enable Transmitter A-Side (ENTA)
Enable Transmitter B-Side (ENTB)

Figure 24 MAX248 Pin Configuration and Typical Operating Circuit

MAX249 Functional Description
10 Receivers
5 A-Side Receivers
5 B-Side Receivers
6 Transmitters
3 A-Side Transmitters
3 B-Side Transmitters
4 Control Pins
Enable Receiver A-Side (ENRA)
Enable Receiver B-Side (ENRB)
Enable Transmitter A-Side (ENTA)
Enable Transmitter B-Side (ENTB)

Figure 25 MAX249 Pin Configuration and Typical Operating Circuit
HS-82C55ARH

Radiation Hardened
CMOS Programmable Peripheral Interface

September 1995

---

Features
- Radiation Hardened
  - Total Dose >10^5 RAD (Si)
  - Transient Upset <10^6 RAD (Si)/s
  - Latch Up Free EPI-CMOS
- Low Power Consumption
  - IDDSB = 20μA
- Pin Compatible with NMOS 8255A and the Harris 82C55A
- High Speed, No "Wait State" Operation with 5MHz HS-80C86RH
- 24 Programmable I/O Pins
- Bus-Hold Circuitry on All I/O Ports Eliminates Pull-Up Resistors
- Direct Bit Set/Reset Capability
- Enhanced Control Word Read Capability
- Hardened Field, Self-Aligned, Junction Isolated CMOS Process
- Single 5V Supply
- 2.0mA Drive Capability on All I/O Port Outputs
- Military Temperature Range: -55°C to +125°C

Description
The Harris HS-82C55ARH is a high performance, radiation hardened CMOS version of the industry standard 8255A and is manufactured using a hardened field, self-aligned silicongate CMOS process. It is a general purpose programmable I/O device which may be used with many different microprocessors. There are 24 I/O pins which are organized into two 8-bit and two 4-bit ports. Each port may be programmed to function as either an input or an output. Additionally, one of the 8-bit ports may be programmed for bi-directional operation, and the two 4-bit ports can be programmed to provide handshaking capabilities. The high performance, radiation hardness, and industry standard configuration of the HS-82C55ARH make it compatible with the HS-80C86RH radiation hardened microprocessor.

Static CMOS circuit design insures low operating power. Bus hold circuitry eliminates the need for pull-up resistors. The Harris hardened field CMOS process results in performance equal to or greater than existing radiation resistant products at a fraction of the power.

Ordering Information

<table>
<thead>
<tr>
<th>PART NUMBER</th>
<th>TEMPERATURE</th>
<th>PACKAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HS1-82C55ARH-Q</td>
<td>-55°C to +125°C</td>
<td>40 Lead SBIDP</td>
</tr>
<tr>
<td>HS1-82C55ARH-8</td>
<td>-55°C to +125°C</td>
<td>40 Lead SBIDP</td>
</tr>
<tr>
<td>HS1-82C55ARH/Sample</td>
<td>+25°C</td>
<td>40 Lead SBIDP</td>
</tr>
</tbody>
</table>

---

Pinout
40 LEAD CERAMIC DUAL-IN-LINE METAL SEAL PACKAGE (SBIDIP)
MIL-STD-1835 CDIP2-740

TABLE VIEW

<table>
<thead>
<tr>
<th>PIN</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>D7</td>
<td>D0</td>
</tr>
<tr>
<td>D8</td>
<td>D1</td>
</tr>
<tr>
<td>D9</td>
<td>D2</td>
</tr>
<tr>
<td>D10</td>
<td>D3</td>
</tr>
<tr>
<td>D11</td>
<td>D4</td>
</tr>
<tr>
<td>D12</td>
<td>D5</td>
</tr>
<tr>
<td>D13</td>
<td>D6</td>
</tr>
<tr>
<td>D14</td>
<td>D7</td>
</tr>
<tr>
<td>D15</td>
<td>VDD</td>
</tr>
<tr>
<td>A0</td>
<td>Port Address</td>
</tr>
<tr>
<td>A1</td>
<td>Port Address</td>
</tr>
<tr>
<td>PA0</td>
<td>Port A (Bit)</td>
</tr>
<tr>
<td>PA1</td>
<td>Port A (Bit)</td>
</tr>
<tr>
<td>PB0</td>
<td>Port B (Bit)</td>
</tr>
<tr>
<td>PB1</td>
<td>Port C (Bit)</td>
</tr>
<tr>
<td>PB2</td>
<td>VDD</td>
</tr>
<tr>
<td>PB3</td>
<td>-5 volts</td>
</tr>
<tr>
<td>GND</td>
<td>0 volts</td>
</tr>
</tbody>
</table>

---

CAUTION: These devices are sensitive to electrostatic discharge. Users should follow proper I.C. Handling Procedures
Copyright © Harris Corporation 1995

Spec Number 518060
File Number 3191.1
**Pin Description**

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>PIN NUMBERS</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA0-7</td>
<td>1-4, 37-40</td>
<td>I/O</td>
<td>Port A: General purpose I/O Port. Data direction and mode is determined by the contents of the Control Word.</td>
</tr>
<tr>
<td>PB0-7</td>
<td>18-25</td>
<td>I/O</td>
<td>Port B: General purpose I/O port. See Port A.</td>
</tr>
<tr>
<td>PC0-3</td>
<td>14-17</td>
<td>I/O</td>
<td>Port C (Lower): Combination I/O port and control port associated with Port B. See Port A.</td>
</tr>
<tr>
<td>PC4-7</td>
<td>10-13</td>
<td>I/O</td>
<td>Port C (Upper): Combination I/O Port and control port associated with Port A. See Port A.</td>
</tr>
<tr>
<td>D0-7</td>
<td>27-34</td>
<td>I/O</td>
<td>Bidirectional Data Bus: Three-State data bus enabled as an input when CS and WR are low and as an output when CS and RD are low.</td>
</tr>
<tr>
<td>VDD</td>
<td>26</td>
<td>I</td>
<td>VDD: The +5V power supply pin. A 0.1μF capacitor between pins 26 and 7 is recommended for decoupling.</td>
</tr>
<tr>
<td>GND</td>
<td>7</td>
<td>I</td>
<td>Ground.</td>
</tr>
<tr>
<td>CS</td>
<td>6</td>
<td>I</td>
<td>Chip Select: A “low” on this input pin enables the communication between the HS-82C55ARH and the CPU.</td>
</tr>
<tr>
<td>RD</td>
<td>5</td>
<td>I</td>
<td>Read: A “low” on this input pin enables the HS-82C55ARH to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to “read from” the HS-82C55ARH.</td>
</tr>
<tr>
<td>WR</td>
<td>36</td>
<td>I</td>
<td>Write: A “low” on this input pin enables the CPU to write data or control words into the HS-82C55ARH.</td>
</tr>
<tr>
<td>A0 and A1</td>
<td>8, 9</td>
<td>I</td>
<td>Port Select 0 and Port Select 1: These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the Least Significant Bits of the address bus (A0 and A1).</td>
</tr>
<tr>
<td>Reset</td>
<td>35</td>
<td>I</td>
<td>Reset: A “high” on this input clears the control register and all ports (A, B, C) are set to the input mode. “Bus hold” devices internal to the HS-82C55ARH will hold the I/O port inputs to a logic “1” state with a maximum hold current of 40μA.</td>
</tr>
</tbody>
</table>

**Functional Diagram**

[Diagram of the HS-82C55ARH with labels for POWER SUPPLIES, DATA BUS BUFFER, READWRITE CONTROL LOGIC, GROUP A CONTROL, GROUP B CONTROL, GROUP A PORT (8), GROUP B PORT C UPPER (4), GROUP B PORT C LOWER (4), GROUP B PORT B (8)].
### Specifications HS-82C55ARH

#### Absolute Maximum Ratings

- **Supply Voltage**................. +7.0V
- **Input, Output or I/O Voltage**........... ±VDD±0.3V to VDD±0.3V
- **Storage Temperature Range**........... -65°C to +125°C
- **Junction Temperature**.............. +150°C
- **Lead Temperature (Soldering 10s)**...... +300°C
- **ESD Classification**............... Class 1

#### Reliability Information

- **Thermal Resistance**......
  - **θJA**........... 40°C/W (BFM Pin)
  - **θJC**........... 6°C/W (SMD Pin)
- **Maximum Package Power Dissipation** at +125°C Ambient
  - **SBIDP Package**...... 1.25W
- **If device power exceeds package dissipation capability, provide heat sinking or derate linearly at the following rate:**
  - **SBIDP Package**...... 25.0mW/°C

**CAUTION**: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

### Operating Conditions

- **Operating Voltage Range**........... +4.5V to +5.5V
- **Input Low Voltage**............... 0V to +0.8V
- **Operating Temperature Range**........... -55°C to +125°C
- **Input High Voltage**.............. VDD-1.5V to VDD

#### TABLE 1. DC ELECTRICAL PERFORMANCE CHARACTERISTICS

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>SYMBOL</th>
<th>CONDITIONS</th>
<th>GROUP A SUBGROUP</th>
<th>TEMPERATURE</th>
<th>LIMITS</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTL Output High Voltage</td>
<td>VOH1</td>
<td>VDD = 4.5V, IO = -2.5mA, V = 0V, 4.5V</td>
<td>1, 2, 3</td>
<td>55°C to 25°C, 125°C</td>
<td>3.0</td>
<td>V</td>
</tr>
<tr>
<td>CMOS Output High Voltage</td>
<td>VOH2</td>
<td>VDD = 4.5V, IO = -100μA, VIN = 0V, 4.5V</td>
<td>1, 2, 3</td>
<td>55°C to 25°C, 125°C</td>
<td>VDD-0.4</td>
<td>V</td>
</tr>
<tr>
<td>Output Low Voltage</td>
<td>VOL</td>
<td>VDD = 4.5V, IO = 2.5mA, V = 0V, 4.5V</td>
<td>1, 2, 3</td>
<td>55°C to 25°C, 125°C</td>
<td>-</td>
<td>0.4</td>
</tr>
<tr>
<td>Input Leakage Current</td>
<td>III or IIH</td>
<td>VDD = 5.5V, VIN = 0V, 5.5V</td>
<td>1, 2, 3</td>
<td>55°C to 25°C, 125°C</td>
<td>-1.0</td>
<td>0.4</td>
</tr>
<tr>
<td>Output Leakage Current</td>
<td>IQZL or IQZH</td>
<td>VDD = 5.5V, VIN = 0V, 5.5V</td>
<td>1, 2, 3</td>
<td>55°C to 25°C, 125°C</td>
<td>-10</td>
<td>10</td>
</tr>
<tr>
<td>Input Current Bus Hold High</td>
<td>IBHH</td>
<td>VDD = 4.5V or 5.5V, VIN = 3.0V (See Note 1)</td>
<td>1, 2, 3</td>
<td>55°C, 25°C, 125°C</td>
<td>-800</td>
<td>-60</td>
</tr>
<tr>
<td>Input Current Bus Hold Low</td>
<td>IBHL</td>
<td>VDD = 4.5V or 5.5V, VIN = 1.0V (See Note 2)</td>
<td>1, 2, 3</td>
<td>55°C to 25°C, 125°C</td>
<td>60</td>
<td>800</td>
</tr>
<tr>
<td>Standby Power Supply Current</td>
<td>IDDSB</td>
<td>VDD = 5.5V, IO = 0mA, VIN = GND or VDD</td>
<td>1, 2, 3</td>
<td>55°C, 25°C, 125°C</td>
<td>-</td>
<td>20</td>
</tr>
<tr>
<td>Darlington Drive Voltage</td>
<td>VDAR</td>
<td>VDD = 4.5V, IO = -2.0mA, VIN = GND or VDD</td>
<td>1, 2, 3</td>
<td>55°C, 25°C, 125°C</td>
<td>3.9</td>
<td>-</td>
</tr>
<tr>
<td>Functional Tests</td>
<td>FT</td>
<td>VDD = 4.5V and 5.5V, VIN = GND or VDD, f = 1MHz</td>
<td>7, 8, 8B</td>
<td>55°C to 25°C, 125°C</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Noise Immunity Functional Test</td>
<td>FN</td>
<td>VDD = 5.5V, VIN = GND or VDD, VDD = 1.5V and VDD = 4.5V, VIN = 0.8V or VDD</td>
<td>7, 8, 8B</td>
<td>55°C, 25°C, 125°C</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**NOTES:**
1. IBHH should be measured after raising VIN and then lowering to 3.0V.
2. IBHL should be measured after lowering VIN to VSS and then raising to 0.8V.
3. No internal current limiting exists on the Port Outputs. A resistor must be added externally to limit the current.
4. For VIH (VDD = 5.5V) and VIL (VDD = 4.5V) each of the following groups is tested separately with all other inputs using VIH = 2.6V, VIL = 0.4V. PA, PB, PC, Control Pins (Pins 5, 6, 8, 9, 35, 36).

Spec Number 518060
# Specifications HS-82C55ARH

## TABLE 2. AC ELECTRICAL PERFORMANCE CHARACTERISTICS $T_a = -55^\circ\text{C} \text{ to } +125^\circ\text{C}$

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>SYMBOL</th>
<th>CONDITIONS</th>
<th>SUB-GROUPS</th>
<th>TEMPERATURE</th>
<th>LIMITS</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>READ</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address Stable Before $\text{RD}$</td>
<td>TAVRL</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>0</td>
<td>ns</td>
</tr>
<tr>
<td>Address Stable After $\text{RD}$</td>
<td>TRHAX</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>0</td>
<td>ns</td>
</tr>
<tr>
<td>$\text{RD}$ Pulse Width</td>
<td>TRLRH</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>250</td>
<td>ns</td>
</tr>
<tr>
<td>Data Valid From $\text{RD}$</td>
<td>TRLDV</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>-</td>
<td>200</td>
</tr>
<tr>
<td>Data Float After $\text{RD}$</td>
<td>TRHDX</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>10</td>
<td>ns</td>
</tr>
<tr>
<td>Time Between $\text{RD}$s and/or $\text{WR}s$</td>
<td>TRWHRWL</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>300</td>
<td>ns</td>
</tr>
<tr>
<td><strong>WRITE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address Stable Before $\text{WR}$</td>
<td>TAVWL</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>0</td>
<td>ns</td>
</tr>
<tr>
<td>Address Stable After $\text{WR}$</td>
<td>TWHAX</td>
<td>VDD = 4.5, 5.5V, Ports A and B</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>20</td>
<td>ns</td>
</tr>
<tr>
<td>$\text{WR}$ Pulse Width</td>
<td>TWLWH</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>100</td>
<td>ns</td>
</tr>
<tr>
<td>Data Valid to $\text{WR}$ High</td>
<td>TDVWH</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>100</td>
<td>ns</td>
</tr>
<tr>
<td>Data Valid After $\text{WR}$ High</td>
<td>TWHDX</td>
<td>VDD = 4.5, 5.5V, Ports A and B</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>30</td>
<td>ns</td>
</tr>
<tr>
<td>Other Timings</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{WR} = 1$ to Output</td>
<td>TVHPV</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>-</td>
<td>350</td>
</tr>
<tr>
<td>Peripheral Data Before $\text{RD}$</td>
<td>TPVRL</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>0</td>
<td>ns</td>
</tr>
<tr>
<td>Peripheral Data After $\text{RD}$</td>
<td>TRHPX</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>0</td>
<td>ns</td>
</tr>
<tr>
<td>$\text{ACK}$ Pulse Width</td>
<td>TKLKH</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>200</td>
<td>ns</td>
</tr>
<tr>
<td>$\text{STB}$ Pulse Width</td>
<td>TSLSH</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>100</td>
<td>ns</td>
</tr>
<tr>
<td>Peripheral Data Before $\text{STB}$ High</td>
<td>TPVSH</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>20</td>
<td>ns</td>
</tr>
<tr>
<td>Peripheral Data After $\text{STB}$ High</td>
<td>TSHPX</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>50</td>
<td>ns</td>
</tr>
<tr>
<td>$\text{ACK} = 0$ to Output</td>
<td>TKLPV</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>-</td>
<td>175</td>
</tr>
<tr>
<td>$\text{ACK} = 1$ to output Float</td>
<td>TKHPZ</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>$-55^\circ\text{C}, +25^\circ\text{C}, +125^\circ\text{C}$</td>
<td>10</td>
<td>ns</td>
</tr>
</tbody>
</table>
**Specifications HS-82C55ARH**

**TABLE 2. AC ELECTRICAL PERFORMANCE CHARACTERISTICS** \( T_A = -55^\circ C \) to \(+125^\circ C\) (Continued)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>SYMBOL</th>
<th>CONDITIONS</th>
<th>SUB-</th>
<th>TEMPERATURE</th>
<th>LIMITS</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>WR = 1 to OBF = 0</td>
<td>TWHOL</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>-55°C, +25°C, +125°C</td>
<td>-</td>
<td>150</td>
</tr>
<tr>
<td>ACK = 0 to OBF = 1</td>
<td>TKLOH</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>-55°C, +25°C, +125°C</td>
<td>-</td>
<td>150</td>
</tr>
<tr>
<td>STB = 0 to IBF = 1</td>
<td>TSLIH</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>-55°C, +25°C, +125°C</td>
<td>-</td>
<td>150</td>
</tr>
<tr>
<td>RD = 1 to IBF = 0</td>
<td>TRHIL</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>-55°C, +25°C, +125°C</td>
<td>-</td>
<td>150</td>
</tr>
<tr>
<td>RD = 0 to INTR = 1</td>
<td>TRNLN</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>-55°C, +25°C, +125°C</td>
<td>-</td>
<td>200</td>
</tr>
<tr>
<td>STB = 1 l INTR = 1</td>
<td>TSHNH</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>-55°C, +25°C, +125°C</td>
<td>-</td>
<td>150</td>
</tr>
<tr>
<td>ACK = 1 to INTR = 1</td>
<td>TKHNH</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>-55°C, +25°C, +125°C</td>
<td>-</td>
<td>150</td>
</tr>
<tr>
<td>WR = 0 to INTR = 0</td>
<td>TWLNL</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>-55°C, +25°C, +125°C</td>
<td>-</td>
<td>200</td>
</tr>
<tr>
<td>RESET Pulse Width</td>
<td>TRSHRSL</td>
<td>VDD = 4.5, 5.5V</td>
<td>9, 10, 11</td>
<td>-55°C, +25°C, +125°C</td>
<td>500</td>
<td>-</td>
</tr>
</tbody>
</table>

**NOTES:**
1. AC's tested at worst case VDD, guaranteed over full operating range.
2. Period of initial RESET pulse after power-on must be at least 50µs. Subsequent RESET pulses may be 500ns minimum.

**TABLE 3. ELECTRICAL PERFORMANCE CHARACTERISTICS**

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>SYMBOL</th>
<th>CONDITIONS</th>
<th>TEMPERATURE</th>
<th>LIMITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Capacitance</td>
<td>CIN</td>
<td>VDD = Open, f = 1MHz, All measurements referenced to device ground</td>
<td>( T_A = +25^\circ C )</td>
<td>-</td>
</tr>
<tr>
<td>I/O Capacitance</td>
<td>CI/O</td>
<td>VDD = Open, f = 1MHz, All measurements referenced to device ground</td>
<td>( T_A = +25^\circ C )</td>
<td>-</td>
</tr>
<tr>
<td>Data Float After RD</td>
<td>TRHDX</td>
<td>VDD = 4.5V and 5.5V</td>
<td>-55°C &lt; ( T_A &lt; +125^\circ C )</td>
<td>-</td>
</tr>
<tr>
<td>ACK = 1 to Output Float</td>
<td>TKHPZ</td>
<td>VDD = 4.5V and 5.5V</td>
<td>-55°C &lt; ( T_A &lt; +125^\circ C )</td>
<td>-</td>
</tr>
</tbody>
</table>

**NOTE:** The parameters listed in Table 3 are controlled via design or process parameters and are not directly tested. These parameters are characterized upon initial design release and upon design changes which would affect these characteristics.

**TABLE 4. POST 100K RAD ELECTRICAL PERFORMANCE CHARACTERISTICS**

See +25°C limits in Table 1 and Table 2 for Post RAD limits (Subgroups 1, 7, 9)
# Specifications HS-82C55ARH

## TABLE 5. BURN-IN DELTA PARAMETERS (+26°C)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>SYMBOL</th>
<th>DELTA LIMITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static Current</td>
<td>IDSSB</td>
<td>±10μA</td>
</tr>
<tr>
<td>Input Leakage Current</td>
<td>III, IH</td>
<td>±200nA</td>
</tr>
<tr>
<td>Output Leakage Current</td>
<td>IOZL, IOZH</td>
<td>±2μA</td>
</tr>
<tr>
<td>Low Level Output Voltage</td>
<td>VOL</td>
<td>±90mV</td>
</tr>
<tr>
<td>TTL Output High Voltage</td>
<td>VOH1</td>
<td>±600mV</td>
</tr>
<tr>
<td>CMOS Output High Voltage</td>
<td>VOH2</td>
<td>±150mV</td>
</tr>
</tbody>
</table>

## TABLE 6. APPLICABLE SUBGROUPS

<table>
<thead>
<tr>
<th>CONFORMANCE GROUP</th>
<th>MIL-STD-883 METHOD</th>
<th>GROUP A SUBGROUPS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TESTED FOR -Q</td>
<td>RECORDED FOR -Q</td>
</tr>
<tr>
<td>Initial Test</td>
<td>100% 5004</td>
<td>1, 7, 9</td>
</tr>
<tr>
<td>Interim Test</td>
<td>100% 5004</td>
<td>1, 7, 9, Δ</td>
</tr>
<tr>
<td>PDA</td>
<td>100% 5004</td>
<td>1, 7, Δ</td>
</tr>
<tr>
<td>Final Test</td>
<td>100% 5004</td>
<td>2, 3, 8A, 8B, 10, 11</td>
</tr>
<tr>
<td>Group A (Note 1)</td>
<td>Sample 5005</td>
<td>1, 2, 3, 7, 8A, 8B, 9, 10, 11</td>
</tr>
<tr>
<td>Subgroup B5</td>
<td>Sample 5005</td>
<td>1, 2, 3, 7, 8A, 8B, 9, 10, 11, Δ</td>
</tr>
<tr>
<td>Subgroup B6</td>
<td>Sample 5005</td>
<td>1, 7, 9</td>
</tr>
<tr>
<td>Group C</td>
<td>Sample 5005</td>
<td>N/A</td>
</tr>
<tr>
<td>Group D</td>
<td>Sample 5005</td>
<td>1, 7, 9</td>
</tr>
<tr>
<td>Group E, Subgroup 2</td>
<td>Sample 5005</td>
<td>1, 7, 9</td>
</tr>
</tbody>
</table>

**NOTES:**
1. Alternate Group A testing in accordance with MIL-STD-883 method 5005 may be exercised.
2. Table 5 parameters only.
Harris Space Level Product Flow -Q

Wafer Lot Acceptance (All Lots) Method 5007
  (Includes SEM)
GAMMA Radiation Verification (Each Wafer) Method 1019,
  2 Samples/Wafer, 0 Rejects
100% Die Attach
100% Nondestructive Bond Pull, Method 2023
Sample - Wire Bond Pull Monitor, Method 2011
Sample - Die Shear Monitor, Method 2019 or 2027
100% Internal Visual Inspection, Method 2010, Condition A
CSI and/or GSI PreCap (Note 6)
100% Temperature Cycle, Method 1010, Condition C,
  10 Cycles
100% Constant Acceleration, Method 2001, Condition per
  Method 5004
100% PIND, Method 2020, Condition A
100% External Visual
100% Serialization
100% Initial Electrical Test (T0)
100% Static Burn-In 1, Condition A or B, 72 Hours Min,
  +125°C Min, Method 1015

100% Interim Electrical Test 1 (T1)
100% Delta Calculation (T0-T1)
100% PDA 1, Method 5004 (Note 1)
100% Dynamic Burn-In, Condition D, 240 Hours, +125°C or
  Equivalent, Method 1015
100% Interim Electrical Test 2(T2)
100% Delta Calculation (T0-T2)
100% PDA 2, Method 5004 (Note 1)
100% Final Electrical Test
100% Fine/Gross Leak, Method 1014
100% Radiographic (X-Ray), Method 2012 (Note 2)
100% External Visual, Method 2009
Sample - Group A, Method 5005 (Note 3)
Sample - Group B, Method 5005 (Note 4)
Sample - Group D, Method 5005 (Notes 4 and 5)
100% Data Package Generation (Note 7)
CSI and/or GSI Final (Note 6)

NOTES:
1. Failures from subgroup 1, 7 and deltas are used for calculating PDA. The maximum allowable PDA = 5% with no more than 3% of the failures from subgroup 7.
2. Radiographic (X-Ray) inspection may be performed at any point after serialization as allowed by Method 5004.
3. Alternate Group A testing may be performed as allowed by MIL-STD-883, Method 5005.
4. Group B and D inspections are optional and will not be performed unless required by the P.O. When required, the P.O. should include separate line items for Group B Test, Group B Samples, Group D Test and Group D Samples.
5. Group D Generic Data, as defined by MIL-I-38535, is optional and will not be supplied unless required by the P.O. When required, the P.O. should include a separate line item for Group D Generic Data. Generic data is not guaranteed to be available and is therefore not available in all cases.
6. CSI and/or GSI inspections are optional and will not be performed unless required by the P.O. When required, the P.O. should include separate line items for CSI PreCap inspection, CSI final inspection, GSI PreCap inspection, and/or GSI final inspection.
7. Data Package Contents:
   • Cover Sheet (Harris Name and/or Logo, P.O. Number, Customer Part Number, Lot Date Code, Harris Part Number, Lot Number, Quantity).
   • Wafer Lot Acceptance Report (Method 5007). Includes reproductions of SEM photos with percent of step coverage.
   • GAMMA Radiation Report. Contains Cover page, disposition, Rad Dose, Lot Number, Test Package used, Specification Numbers, Test equipment, etc. Radiation Read and Record data on file at Harris.
   • X-Ray report and film. Includes penetrometer measurements.
   • Screening, Electrical, and Group A attributes (Screening attributes begin after package seal).
   • Lot Serial Number Sheet (Good units serial number and lot number).
   • Variables Data (All Delta operations). Data is identified by serial number. Data header includes lot number and date of test.
   • Group B and D attributes and/or Generic data is included when required by the P.O.
   • The Certificate of Conformance is a part of the shipping invoice and is not part of the Data Book. The Certificate of Conformance is signed by an authorized Quality Representative.
HS-82C55ARH

Harris Space Level Product Flow -8

GAMMA Radiation Verification (Each Wafer) Method 1019,
2 Samples/Wafer, 0 Rejects
100% Die Attach
Periodic Wire Bond Pull Monitor, Method 2011
Periodic Die Shear Monitor, Method 2019 or 2027
100% Internal Visual Inspection, Method 2010, Condition B
CSI and/or GSI PreCap (Note 5)
100% Temperature Cycle, Method 1010, Condition C,
10 Cycles
100% Constant Acceleration, Method 2001, Condition per
Method 5004
100% External Visual
100% Initial Electrical Test
100% Dynamic Burn-In, Condition D, 160 Hours, +125°C or
Equivalent, Method 1015
100% Interim Electrical Test
100% PDA, Method 5004 (Note 1)
100% Final Electrical Test
100% Fine/Gross Leak, Method 1014
100% External Visual, Method 2009
Sample - Group A, Method 5005 (Note 2)
Sample - Group B, Method 5005 (Note 3)
Sample - Group C, Method 5005 (Notes 3 and 4)
Sample - Group D, Method 5005 (Notes 3 and 4)
100% Data Package Generation (Note 6)
CSI and/or GSI Final (Note 5)

NOTES:
1. Failures from subgroup 1, 7 are used for calculating PDA. The maximum allowable PDA = 5%. 2. Alternate Group A testing may be performed as allowed by MIL-STD-883, Method 5005.
3. Group B, C and D inspections are optional and will not be performed unless required by the P.O. When required, the P.O. should include separate line items for Group B Test, Group C Test, Group C Samples, Group D Test and Group D Samples.
4. Group C and/or Group D Generic Data, as defined by MIL-I-38535, is optional and will not be supplied unless required by the P.O. When required, the P.O. should include a separate line item for Group C Generic Data and/or Group D Generic Data. Generic data is not guaranteed to be available and is therefore not available in all cases.
5. CSI and/or GSI inspections are optional and will not be performed unless required by the P.O. When required, the P.O. should include separate line items for CSI PreCap inspection, CSI final inspection, GSI PreCap inspection, and/or GSI final inspection.
6. Data Package Contents:
• Cover Sheet (Harris Name and/or Logo, P.O. Number, Customer Part Number, Lot Date Code, Harris Part Number, Lot Number, Quantity).
• GAMMA Radiation Report. Contains Cover page, disposition, Rad Dose, Lot Number, Test Package used, Specification Numbers, Test equipment, etc. Radiation Read and Record data on file at Harris.
• Screening, Electrical, and Group A attributes (Screening attributes begin after package seal).
• Group B, C and D attributes and/or Generic data is included when required by the P.O.
• The Certificate of Conformance is a part of the shipping invoice and is not part of the Data Book. The Certificate of Conformance is signed by an authorized Quality Representative.

AC Test Circuit

FROM OUTPUT UNDER TEST

R1
V1

TEST POINT

R2
C1*

* Includes stray and jig capacitance

TEST CONDITIONS DEFINITION TABLE

<table>
<thead>
<tr>
<th>V1</th>
<th>R1</th>
<th>R2</th>
<th>C1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.7V</td>
<td>523Ω</td>
<td>Open</td>
<td>150pF</td>
</tr>
</tbody>
</table>

AC Testing Input, Output Waveforms

INPUT
0.4V

2.8V

1.5V

1.5V

NOTE: AC Testing: All parameters tested as per test circuits. Input rise and fall times are driven at 1V/ns.
**Burn-In Circuits**

**PROGRAMMABLE PERIPHERAL INTERFACE**

**STATIC CONFIGURATION**

**NOTES:**
1. VDD = 6.0V ± 0.5%
2. IDD < 500μA
3. $T_A$ Min = $+125^\circ$C

**PROGRAMMABLE PERIPHERAL INTERFACE**

**DYNAMIC CONFIGURATION**

**NOTES:**
1. VDD = 6.0V ± 5% for Burn-In
2. VDD = 5.0V ± 5% for Life Test
3. All resistors are 10KΩ ± 5%
4. $-0.3V \leq V_{IL} \leq 0.8V$
5. VDD - 1.0V ≤ $V_{IH}$ ≤ VDD
6. IDD < 5mA
7. $F_0$ = 10KHz, 50% Duty cycle
8. $F_1$ = $F_0/2$; $F_2$ = $F_1/2$; $F_3$ = $F_2/2$; $F_4$ = $F_3/2$; ... $F_7$ = $F_6/2$
9. $T_A$ Min = $+125^\circ$C
Irradiation Circuit

CMOS PROGRAMMABLE PERIPHERAL INTERFACE

+5.5V

NOTE:
1. VDD = 5.5V
Functional Description

The HS-82C55ARH is a programmable peripheral interface designed to allow microcomputer systems to control and interface with all types of peripheral devices. It has the ability to generate and respond to all asynchronous handshaking signals necessary to transfer data to and from peripheral devices, and it can also interrupt the processor when a peripheral needs servicing. These capabilities allow the HS-82C55ARH to be used in an unlimited number of applications including EXTERNAL SYSTEM CONTROL, ASYNCHRONOUS DATA TRANSFER, and SYSTEMS MONITORING.

Data Bus Buffer

This tri-state bidirectional 8-bit buffer is used to interface the HS-82C55ARH to the system data bus (see Figure 8). Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfer of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU writes a control word to the HS-82C55ARH. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the HS-82C55ARH.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group - Port A and Port C upper (C7 - C4)
Control Group - Port B and Port C lower (C3 - C0).

Ports A, B, C

The HS-82C55ARH contains three 8-bit ports (A, B and C). All can be configured to a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the HS-82C55ARH.

Port A

One 8-bit data output latch/buffer and one 8-bit data input latch. Both "pull-up" and "pull-down" bus hold devices are present on Port A. See Figure 9A.

Port B

One 8-bit data input/output latch/buffer and one 8-bit data input buffer. See Figure 9B.

Port C

One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and can be used for the control signal outputs and status signal inputs in conjunction with Ports A and B. See Figure 9B.

Operational Description

Control Word

The data direction and mode of Ports A, B and C are determined by the contents of the Control Word. See Figure 11. The Control Word can be both written and read as shown in Table 1 and 2. During write operations, the function of the Control Word being written is determined by data bit D7. If D7 is low, the data on D0 - D3 will set or reset one of the bits of Port C. See Figure 12. During read operations, the

Spec Number 518060
Control Word will always be in the format illustrated in Figure 11 with Bit D7 high to indicate Control Word Mode Information.

**TABLE 1.**

<table>
<thead>
<tr>
<th>A1</th>
<th>A0</th>
<th>RD</th>
<th>WR</th>
<th>CS</th>
<th>INPUT OPERATION (READ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Port A - Data Bus</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Port B - Data Bus</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Port C - Data Bus</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Control Word - Data Bus</td>
</tr>
</tbody>
</table>

**TABLE 2.**

<table>
<thead>
<tr>
<th>A1</th>
<th>A0</th>
<th>RD</th>
<th>WR</th>
<th>CS</th>
<th>OUTPUT OPERATION (WRITE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Data Bus - Port A</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Data Bus - Port B</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Data Bus - Port C</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Data Bus - Control Word</td>
</tr>
</tbody>
</table>

**TABLE 3.**

<table>
<thead>
<tr>
<th>A1</th>
<th>A0</th>
<th>RD</th>
<th>WR</th>
<th>CS</th>
<th>DISABLE FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>Data Bus - 3-State</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Data Bus - 3-State</td>
</tr>
</tbody>
</table>

**FIGURE 11. MODE SET CONTROL WORD FORMAT**

**Mode Selection**

There are three basic modes of operation that can be selected by the system software:

Mode 0 - Basic Input/Output
Mode 1 - Strobed Input/Output
Mode 2 - Bidirectional Bus

When the RESET input goes "high", all ports will be set to the input mode with all 24 port lines held at the logic "one" level by internal bus hold devices. After reset, the HS-82C55ARH can remain in the input mode with no additional initialization required. This eliminates the need for pullup or pulldown resistors in all CMOS designs. During the execution of the system program, any of the other modes may be selected using a single output instruction. This allows a single HS-82C55ARH to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status register, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance: Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results. Group A could be programmed in Mode 1 to monitor a keyboard or tape recorder on an interrupt-driven basis.
HS-82C55ARH

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the HS-82C55ARH has taken into account things such as efficient PCB layout, control signal definition vs. PCB layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Interrupt Control Functions

When the HS-82C55ARH is programmed to operate in Mode 1 or Mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from Port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the Bit Set/Reset function of Port C.

This function allows the programmer to enable or disable a CPU interrupt by a specific I/O device without affecting any other device in the interrupt structure.

INTE Flip-Flop Definition:

(BIT-SET) - INTE is SET - Interrupt enable.

(BIT-RESET) - INTE is RESET - Interrupt disable.

NOTE: All mask flip-flops are automatically reset during mode selection and device Reset.

Operating Modes

Mode 0 (Basic Input/Output)

This functional configuration provides simple input and output operations for each of the three ports. No handshaking is required, data is simply written to or read from a specific port.

Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports
- Any port can be input or output
- Outputs are latched
- Inputs are not latched
- 16 different Input/Output configurations possible

---

**FIGURE 12. BIT SET/RESET CONTROL WORD FORMAT**

**Single Bit/Set/Reset Feature**

Any of the eight bits of Port C can be Set or Reset using a single OUTPUT instruction. See Figure 12. This feature reduces software requirements in control-based applications.

---

**FIGURE 13. MODE 0 (BASIC INPUT)**

**FIGURE 14. MODE 0 (BASIC OUTPUT)**
### Mode 0 Port Definition

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>GROUP A</th>
<th>NO.</th>
<th>GROUP B</th>
</tr>
</thead>
<tbody>
<tr>
<td>D4</td>
<td>D3</td>
<td>D1</td>
<td>D0</td>
<td>PORT A</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Output</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Output</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Output</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Output</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Output</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Output</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Output</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Output</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Input</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Input</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Input</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Input</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Input</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Input</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Input</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Input</td>
</tr>
</tbody>
</table>

### Mode 0 Configurations

#### CONTROL WORD #0

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### CONTROL WORD #1

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

#### CONTROL WORD #2

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

#### CONTROL WORD #3

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
**Mode 0 Configurations (Continued)**

**CONTROL WORD #12**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**CONTROL WORD #13**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**CONTROL WORD #14**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**CONTROL WORD #15**

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

---

**Operating Modes**

**Mode 1 (Strobed Input/Output)**

This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In Mode 1, Port A and Port B use the lines on Port C to generate or accept these "handshaking" signals.

**Mode 1 Basic Functional Definitions:**

- **Two Groups (Group A and Group B)**
- Each group contains one 8-bit port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit port.

**Input Control Signal Definition**

**STB (Strobe Input)**

A "low" on this input loads data into the input latch.

**IBF (Input Buffer Full F/F)**

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgment. IBF is set by STB input being low and is reset by the rising edge of the RD input.

---

**INTR (Interrupt Request)**

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the rising edge of STB and reset by the falling edge of RD. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

**INTA**

Controlled by Bit Set/Reset of PC4.

**INTB**

Controlled by Bit Set/Reset of PC2.

---

Figures 15. Mode 1 Input
Output Control Signal Definition

OBF (Output Buffer Full F/F)
The OBF output will go "low" to indicate that the CPU has written data out to the specified port. This does not mean valid data is sent out of the port at this time since OBF can go true before data is available. Data is guaranteed valid at the rising edge of OBF. See Note 1. The OBF F/F will be set by the rising edge of the WR input and reset by ACK input being low.

ACK (Acknowledge Input)
A "low" on this input informs the HS-82C55ARH that the data from Port A or Port B is ready to be accepted. In essence, a response from the peripheral device indicating that it is ready to accept data. See Note 1.

INTR (Interrupt Request)
A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set by the rising edge of ACK and reset by the falling edge of WR.

MODE 1 (PORT A) CONTROL WORD

1 0 1 0 1 0

PC4, 5
1 = INPUT
0 = OUTPUT

MODE 1 (PORT B) CONTROL WORD

1 1 1 1 1 0

PC6, 7
1 = INPUT
0 = OUTPUT

FIGURE 17. MODE 1 OUTPUT

INTE A
Controlled by Bit Set/Reset of PC6.

INTE B
Controlled by Bit Set/Reset of PC2.

FIGURE 18. MODE 1 (STROBED OUTPUT)

FIGURE 19. COMBINATIONS OF MODE 1

NOTE:
1. To strobe data into the peripheral device, the user must operate the strobe line in a hand shaking mode. The user needs to send OBF to the peripheral device, generate an ACK from the peripheral device and then latch data into the peripheral device on the rising edge of OBF.

Combinations of Mode 1: Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.
Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O)

The functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline similar to MODE 1. Interrupt generation and enable/disable functions are also available.

Mode 2 Basic Functional Definitions:

- Used in Group A only.
- One 8-bit, bidirectional bus port (Port A) and a 5-bit control port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bidirectional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request)

A high on this output can be used to interrupt the CPU for both input or output operations. INTR will be set either by the rising edge of ACK (INTE1 = 1) or the rising edge of STB (INTE2 = 1). INTR will be reset by the falling edge of WR (if previously set by the rising edge of ACK), the falling edge of RD (if previously set by the rising edge of STB), or the falling edge of WR when immediately following a low RD pulse or the falling edge of RD when immediately following a low WR pulse (if previously set by the rising edges of both ACK and STB).

Output Operations

OBF (Output Buffer Full)

The OBF output will go "low" to indicate that the CPU has written data out to Port A.

ACK (Acknowledge)

A "low" on this input enables the tri-state output buffer of Port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 (The INTE Flip-Flop Associated with OBF)

Controlled by Bit Set/Reset of PC6.

Input Operations

STB (Strobe Input)

A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF)

Controlled by Bit Set/Reset of PC4.
### MODE DEFINITION SUMMARY

<table>
<thead>
<tr>
<th>Mode 0</th>
<th>Mode 1</th>
<th>Mode 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IN</strong></td>
<td><strong>OUT</strong></td>
<td><strong>IN</strong></td>
</tr>
<tr>
<td>PA0</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>AP1</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PA2</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PA3</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PA4</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PA5</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PA6</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PA7</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PB0</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PB1</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PB2</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PB3</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PB4</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PB5</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PB6</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PB7</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PC0</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PC1</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PC2</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PC3</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PC4</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PC5</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PC6</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>PC7</td>
<td>In</td>
<td>Out</td>
</tr>
</tbody>
</table>

### Special Mode Combination Considerations

There are several combinations of modes possible. For any combination, some or all of Port C lines are used for control or status. The remaining bits are either inputs or outputs as defined by a "Set Mode" command.

During a read of Port C, the state of all the Port C lines, except the ACK and STB lines, will be placed on the data bus. In place of the ACK and STB line states, flag status will appear on the data bus in the PC2, PC4, and PC6 bit positions as illustrated by Figure 25.

Through a "Write Port C" command, only the Port C pins programmed as outputs in a Mode 0 group can be written. No other pins can be affected by a "Write Port C" command, nor can the interrupt enable flags be accessed. To write to any Port C output programmed as an output in a Mode 1 group or to change an interrupt enable flag, the "Set/Reset Port C Bit" command must be used.

With a "Set/Reset Port C Bit" command, any Port C line programmed as an output (including IBF and OBF) can be written, or an interrupt enable flag can be either set or reset. Port C lines programmed as inputs, including ACK and STB lines, associated with Port C are not affected by a "Set/Reset Port C Bit" command. Writing to the corresponding Port C bit positions of the ACK and STB lines with the "Set/Reset Port C Bit" command will affect the Group A and Group B interrupt enable flags, as illustrated in Figure 25.

### INPUT CONFIGURATION

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O</td>
<td>I/O</td>
<td>IBFA</td>
<td>INTEA</td>
<td>INTRA</td>
<td>INTEB</td>
<td>IBFB</td>
<td>INTRB</td>
</tr>
</tbody>
</table>

GROUP A

### OUTPUT CONFIGURATION

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBFA</td>
<td>INTEA</td>
<td>I/O</td>
<td>I/O</td>
<td>INTRA</td>
<td>INTEB</td>
<td>OBF B</td>
<td>INTRB</td>
</tr>
</tbody>
</table>

GROUP A

GROUP B

### FIGURE 23. MODE 1 STATUS WORD FORMAT

<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBFA</td>
<td>INTE1</td>
<td>IBFA</td>
<td>INTE2</td>
<td>INTRA</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

GROUP A

GROUP B

NOTE: (Defined by Mode 0 or Mode 1 Selection)

### FIGURE 24. MODE 2 STATUS WORD FORMAT

Spec Number 518060

20
Current Drive Capability
Any output on Port A, B or C can sink or source 2.5mA. This feature allows the 82C55A to directly drive Darlington type drivers and high-voltage displays that require such sink or source current.

Reading Port C Status (Figures 23 and 24)
In Mode 0, Port C transfers data to or from the peripheral device. When the 82C55A is programmed to function in Modes 1 or 2, Port C generates or accepts “hand shaking” signals with the peripheral device. Reading the contents of Port C allows the programmer to test or verify the “status” of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

<table>
<thead>
<tr>
<th>INTERRUPT ENABLE FLAG</th>
<th>POSITION</th>
<th>ALTERNATE PORT C PIN SIGNAL (MODE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTE B</td>
<td>PC2</td>
<td>ACKB (Output Mode 1) or STBB (Input Mode 1)</td>
</tr>
<tr>
<td>INTE A2</td>
<td>PC4</td>
<td>STBA (Input Mode 1 or Mode 2)</td>
</tr>
<tr>
<td>INTE A1</td>
<td>PC6</td>
<td>ACKA (Output Mode 1 or Mode 2)</td>
</tr>
</tbody>
</table>

FIGURE 25. INTERRUPT ENABLE FLAGS IN MODES 1 AND 2
Metallization Topology

DIE DIMENSIONS:
3420µm x 4350µm x 485µm ± 25µm

METALLIZATION:
Type: Al/Si
Thickness: 11kÅ ± 2kÅ

GLASSIVATION:
Type: SiO2
Thickness: 8kÅ ± 1kÅ

WORST CASE CURRENT DENSITY:
7.7 x 10^3 A/cm²

Metallization Mask Layout

All Harris Semiconductor products are manufactured, assembled and tested under ISO9000 quality systems certification.
INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center ...................................................... 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library .............................................................................. 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Chairman, Code EC .............................................................................. 1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

4. Prof. Douglas J. Fouts, Code EC/Fs ....................................................... 2
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

5. Prof. Todd R. Weatherford, Code EC/Wt ............................................. 1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

6. CPT Duane E. Amsler Jr. ................................................................. 2
   80 Blessing Rd.
   Slingerlands, NY 12159