# *Hardware Specification*

# *S-7600A*

## *TCP/IP Network Protocol LSI*

*Seiko Instruments USA Inc.*

*Phone +1-909-934-9334*
*Fax +1-909-975-5699*

*2990 West Lomita Boulevard*
*Torrance, California 90505*

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# 1. Introduction

## 1.1. Product Overview

The S-7600A is a LSI that integrates TCP/IP network stack. It offers your devices a quicker and easier connectivity to a network with its on-chip serial interface and a static RAM that operates as a buffer. Implementing this LSI into your system can significantly reduce your software development cost. Also its low operating frequency gives benefits to the power consumption.
The S-7600A also supports a microprocessor interface via the iReady iAPI™ register set, and connection to Physical Transport Layer Interface. iAPI consists of a set of register and operating definitions that allow any micro controller system to interface with the internal modules.

## 1.2. Features

- Industry standard protocols support :
  TCP/IP (Ver. 4.0)
  PPP (STD-51-compliant)
  UDP
- General purpose sockets :
  Configured for two sockets
- MPU interface :
  68k/x80(MOTO/Intel) bus interface or Synchronous serial interface
- Physical Transport Layer Interface :
  Universal Asynchronous Receiver/Transmitter (UART)
- Low clock rate :
  Multiplied four by the bit-rate
- Operating frequency :
  256kHz typical
- Low power consumption :
  Full-transmitting Operating current consumption : 0.9mA typ.
  Non-transmitting Operating current consumption : 150µA typ.
  Standby current consumption : 1.0µA typ.
- Stand-by mode :
  held by RESET signal
- Wide operating voltage range :
  2.4V to 3.6V
- Easier application development :
  portable iAPI™ support

## 1.3. Benefits

- Off-loads MIPS allowing system to operate with low end and low cost processors.
- Consumes minimal power-up to 1/100 of competing solution.

## 1.4. Trademarks

iReady iAPI™ and iAPI™ is a trademark of iReady Corporation. All other products and brand names are trademarks and registered trademarks of their respective companies.

## 1.5. Definitions

- IP Internet Protocol
- PPP Point-to-Point Protocol
- TCP Transmission Control Protocol
- UDP User Datagram Protocol
- API Application Programming Interface

## 1.6. Applicable Documents

- S-7600A Functional Specification
- S-7600A API Application Manual

## 1.7. Cautions

1. DO NOT apply a voltage or current that exceeds the absolute maximum ratings to terminals. If applied, the IC may malfunction or be destroyed.
   The standard values are set with sufficient margins, but use the IC within the recommended operating conditions to optimize device quality.
2. Measures against static electricity
   2.1 When transporting or storing ICs, use conductive containers or metal coated boxes.
   2.2 Check that there is no current leakage in electrical facilities, and be sure to ground them. Also ensure that workbenches and people who handle ICs are grounded.
3. Excessive external noise to the power supply or I/O terminals of CMOS ICs causes latch-up, leading to faults and damage. If latch-up has occurred, immediately turn off the device, eliminate the cause, and turn on the device again.
4. Keep the IC away from mechanical vibration, shock, and sudden changes in temperature. These may cause wires to break.
5. Environment
   5.1 Use and store ICs below the absolute maximum rated temperature.
   5.2 DO NOT use or store ICs where condensation can occur.
   5.3 DO NOT use ICs where they are directly exposed to dust, salt, or acid gas such as $SO_2$. These may cause leaks between element leads and cause corrosion.
   5.4 To store ICs for a long time, DO NOT process them. During storage, DO NOT apply any load to ICs.

## 2. Functional Block Diagram

Figure 2-1 shows a functional block diagram of the S-7600A. There are blocks of the Network Stack and other functions related to it. The S-7600A has the interface for a host MPU and a Physical layer for various data terminal equipment.

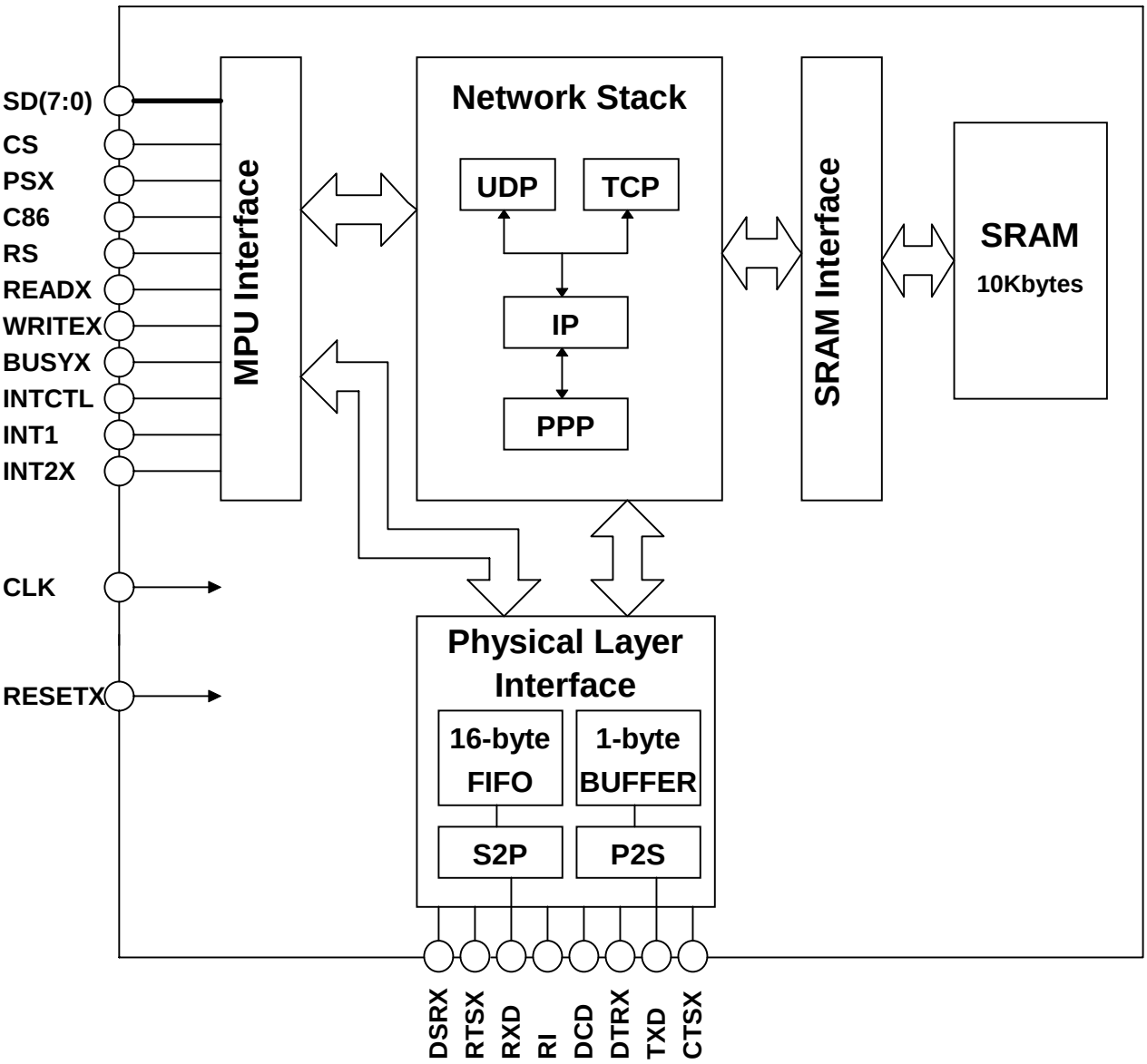



**Figure 2-1 Block diagram**

The transport and network layers contain:

- Two general sockets that provide connectivity between the application layer and the transport layer.
- TCP/UDP module that allows for reliable (retransmission) and unreliable (no retransmission) datagram deliveries.
- IP module that provides connectionless packet delivery.
- PPP module that provides point-to-point connection link between two hosts.

# 3. Terminals

## 3.1. Pin Assignment

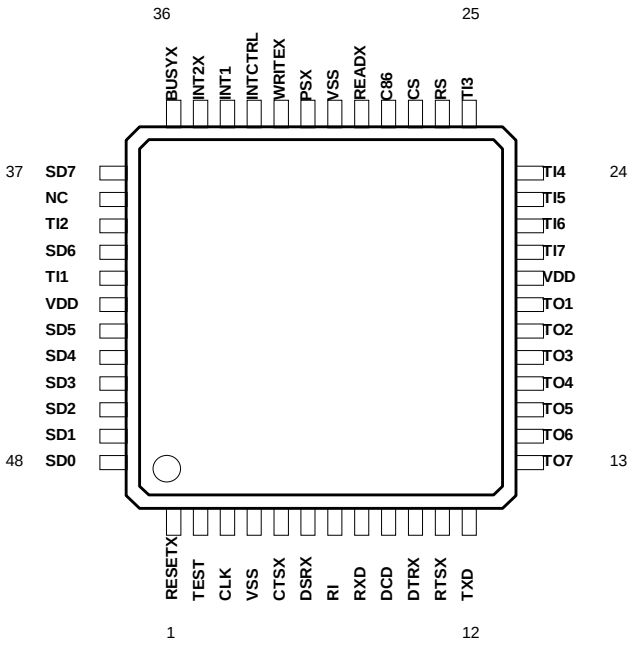



**Figure 3-1Pin Assignment**

Figure 3-1 shows Pin Assignment in Package.

Table 3-1 shows signal names, listed by Pin Number.

| Pin No. | Pin name | Pin No. | Pin name | Pin No. | Pin name | Pin No. | Pin name |
|---|---|---|---|---|---|---|---|
| 1 | RESETX | 13 | TO7 | 25 | TI3 | 37 | SD7 |
| 2 | TEST | 14 | TO6 | 26 | RS | 38 | NC |
| 3 | CLK | 15 | TO5 | 27 | CS | 39 | TI2 |
| 4 | VSS | 16 | TO4 | 28 | C86 | 40 | SD6 |
| 5 | CTSX | 17 | TO3 | 29 | READX | 41 | TI1 |
| 6 | DSRX | 18 | TO2 | 30 | VSS | 42 | VDD |
| 7 | RI | 19 | TO1 | 31 | PSX | 43 | SD5 |
| 8 | RXD | 20 | VDD | 32 | WRITEX | 44 | SD4 |
| 9 | DCD | 21 | TI7 | 33 | INTCTRL | 45 | SD3 |
| 10 | DTRX | 22 | TI6 | 34 | INT1 | 46 | SD2 |
| 11 | RTSX | 23 | TI5 | 35 | INT2X | 47 | SD1 |
| 12 | TXD | 24 | TI4 | 36 | BUSYX | 48 | SD0 |

**Table 3-1 Pin Assignment**

## 3.2. Package Dimensions

S-7600A is housed in a 48-pin QFP package with 0.5mm pin pitch spacing. The package layout is depicted in Figure 3-2.

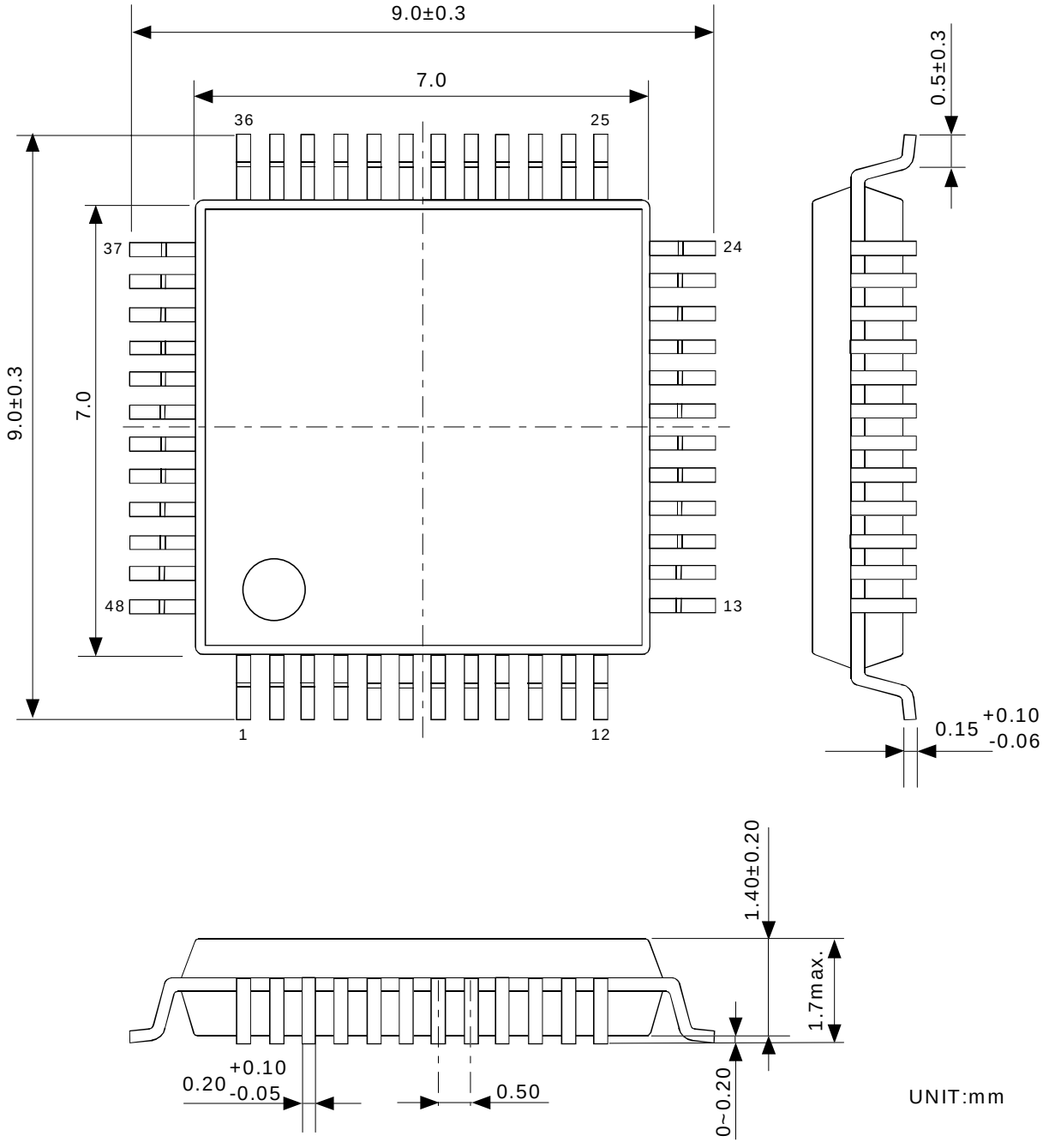



**Figure 3-2 Package Dimensions**

## *3.3. Pin Description*

The pins and signal descriptions are listed by function in Table 3-2.

| Name | I/O | Description | Type |
|---|---|---|---|
| VDD1,VDD2 | - | Positive power supply | |
| VSS1,VSS2 | - | GND potential | |
| RESETX | I | Reset input | A |
| TEST,<br>TI1 to TI7 | I | Test input (pull-down resistor is built in)<br>When normal use, connect to $V_{SS}$ or open | B |
| TO1 to TO7 | O | Test output<br>When normal use, connect to Vss or open | D |
| CLK | I | Clock input | C |
| CTSX | I | Clear to send input | C |
| DSRX | I | Data set ready input | C |
| RI | I | Ring indicator input | C |
| RXD | I | Serial received data input | C |
| DCD | I | Data carrier detect input | C |
| DTRX | O | Data terminal ready output | D |
| RTSX | O | Request to send output | D |
| TXD | O | Serial transmit data output | D |
| RS | I | Register selection input | C |
| CS | I | Chip selection input | C |
| C86 | I | MPU interface mode selection input<br>68k mode : 1<br>x80 mode : 0 | C |
| READX | I | x80 mode : read requirement input<br>68k mode : enable input | C |
| PSX | I | parallel/serial interface selection input | C |
| WRITEX | I | x80 mode : write requirement input<br>68k mode : read/write selection input | C |
| INTCTRL | I | INT1/INT2X drive type(CMOS/OD) selection input | C |
| INT1 | *OT | Interrupt output(active High) from S-7600A chip to MPU | E |
| INT2X | *OT | Interrupt output(active Low) from S-7600A chip to MPU | E |
| BUSYX | O | busy indicator output | D |
| SD7 | *B | x80/68k mode : data bus<br>Serial mode : serial data input | F |
| SD6 | *B | x80/68k mode : data bus<br>Serial mode : serial clock input | F |
| SD5 | *B | x80/68k mode : data bus<br>Serial mode : serial data output | F |
| SD0 to SD4 | *B | Data bus | F |

*OT : Tri-state output
*B : bi-directional

**Table 3-2 Pin Description**

## 3.4. Pin Configuration

Figure 3-3 shows configuration of each pin.

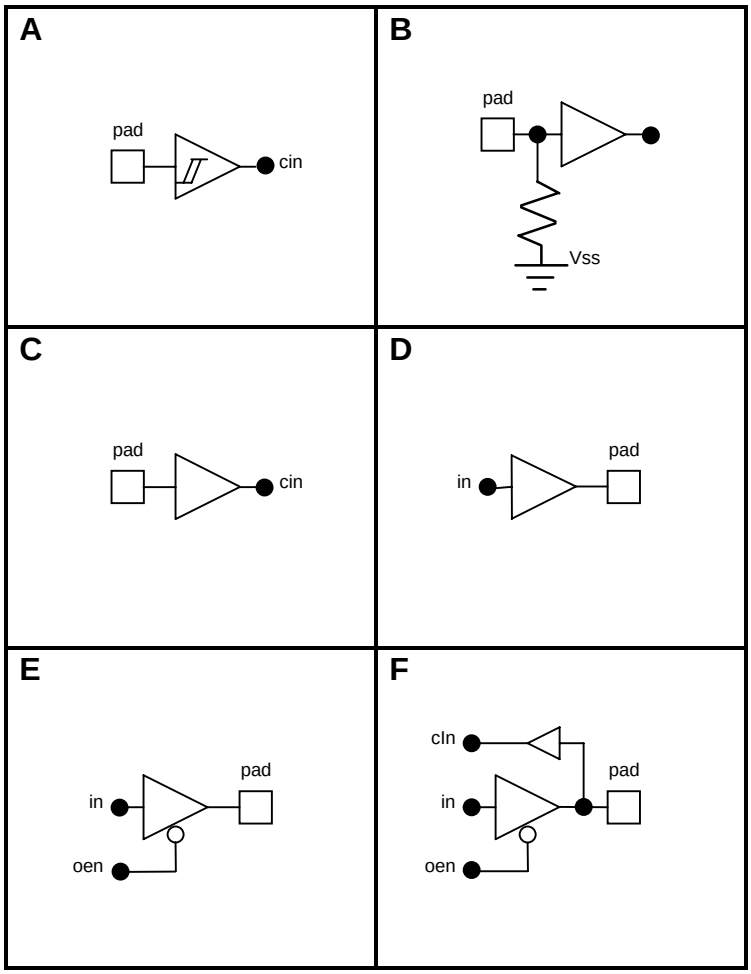



**Figure 3-3 Configuration of Each pin**

# 4. Electrical Characteristics

## 4.1. Absolute Maximum Ratings

| Parameter | Symbol | Conditions | Rating | Unit |
|---|---|---|---|---|
| Storage temperature | $T_{sta}$ | | -40 to +125 | °C |
| Operating temperature | $T_{opr}$ | | -10 to +70 | °C |
| Power supply voltage | $V_{DD}$ | Ta=25°C | -0.3 to +4.0 | V |
| Input voltage | $V_{IN}$ | Ta=25°C | $V_{SS}$-0.3 to $V_{DD}$+0.3 | V |
| Output voltage | $V_{OUT}$ | Ta=25°C | $V_{SS}$ to $V_{DD}$ | V |

**Table 4-1 Absolute Maximum Ratings**

## 4.2. Recommended Operating Conditions

| Parameter | Symbol | Conditions | Min. | Typ. | Max. | Unit | Note |
|---|---|---|---|---|---|---|---|
| Operating Frequency range | $F_{OPR}$ | Ta=-10 to +70°C | - | 0.256 | 5 | MHz | 1 |
| Clock Pulse width | Pw | Ta=-10 to +70°C | 80 | - | - | nS | |
| Operating voltage range | $V_{DD}$ | Ta=-10 to +70°C | 2.4 | - | 3.6 | V | |
| Input voltage | $V_{IN}$ | Ta=-10 to +70°C | 0 | - | $V_{DD}$ | V | |

Note1: The clock is given by the CLK pin and needs to be as four times or more fast as the BAUD rate. (The multiplier is an integer whose tolerance is <2%)

**Table 4-2 Recommended Operating Conditions**

## 4.3. DC Characteristics

Unless otherwise specified: $V_{DD}$=3.0V, $V_{SS}$=0V, Ta=25°C

| Parameter | Symbol | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Low level input voltage | $V_{IL}$ | | 0.2× $V_{DD}$ | - | - | V |
| High level input voltage | $V_{IH}$ | | - | - | 0.8× $V_{DD}$ | V |
| Low level input leakage current | $I_{LL}$ | $V_{IN}$=$V_{SS}$ | -1.0 | - | 1.0 | µA |
| High level input leakage current | $I_{LH}$ | All input terminals without pull-down resister $V_{IN}$=$V_{DD}$ | -1.0 | - | 1.0 | µA |
| High level input current | $I_{IH}$ | All input terminals with pull-down resister $V_{IN}$=$V_{DD}$ | 18 | 70 | 220 | µA |
| Low level output current | $I_{OL}$ | $V_{OL}$=0.4V | 5.0 | - | - | mA |
| High level output current | $I_{OH}$ | $V_{OH}$=2.6V | - | - | -3.5 | mA |
| Schmitt Hysteresis voltage | $V_{WD}$ | | - | 0.46 | - | V |

**Table 4-3 DC Characteristics**

## 4.4. Power Current Consumption

Unless otherwise specified: $V_{DD}$=3.0V, $V_{SS}$=0V, Ta=25°C

| Parameter | Symbol | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Full-transmitting Operating current consumption | $I_{DD1}$ | Ta=-10 to +70°C $F_{OPR}$=256KHz | - | 0.9 | 2.2 | mA |
| Non-transmitting Operating current consumption | $I_{DD2}$ | Ta=-10 to +70°C $F_{OPR}$=256KHz RESETX=$V_{SS}$ | - | 150 | 300 | µA |
| Standby current consumption | Is | Ta=-10 to +70°C | - | 1.0 | 15.0 | µA |

**Table 4-4 Power Current Consumption**

# 5. MPU Interface

## 5.1. Overview

The S-7600A supports two MPU interfaces: parallel and serial. In parallel interface mode, S-7600A can interface with x80 Family MPU or 68k Family MPU.

| PSX | CS | RS | READX | WRITEX | BUSYX | C86 | SD7 | SD6 | SD5 | SD4 to SD0 |
|---|---|---|---|---|---|---|---|---|---|---|
| H: parallel x80 | CS | RS | READX | WRITEX | BUSYX | L | D7 | D6 | D5 | D4 to D0 |
| H: parallel 68k | CS | RS | E | R/WX | BUSYX | H | D7 | D6 | D5 | D4 to D0 |
| L: serial | CS | RS | H or L | R/WX | BUSYX | H or L | SI | SCL | SO | Hi-Z |

**Table 5-1 Interface Selection**

## 5.2. Parallel Interface

Setting **PSX** to “H” select the parallel interface. In parallel interface mode the S-7600A can interface with either x80 Family MPU or 68k Family MPU. The desired MPU mode can be selected by setting the C86 pin to “H” or “L”.

| RS | 68k Family MPU R/WX | x80 Family MPU | | Function |
|---|---|---|---|---|
| | | READX | WRITEX | |
| 1 | 1 | 0 | 1 | Read Register |
| 1 | 0 | 1 | 0 | Write Register |
| 0 | 1 | 0 | 1 | Read Index Register |
| 0 | 0 | 1 | 0 | Write Index Register |

**Table 5-2 Connection Relationship between MPU and Pins**

### *5.2.1. 68k Family MPU Mode*

This mode can be selected by pulling the **C86** input pin “H” and the **PSX** input pin “H”. In this mode, the address and data are muxed into a single 8-bit bus. All cycles start by placing an address on the bus and setting the **RS** pin to “L”. In this mode **WRITEX** signal works as read/write(R/WX) signal and **READX** is the enable(E) signal for 68k Family MPU interface. After the address cycle, the MPU generates a read or writes strobe by setting the **READX** and **WRITEX** pins. The S-7600A MPU interface logic assert a **BUSYX** signal low during data write and read phases. The MPU samples the **BUSYX** signal before starting a new cycle. The CPU can initiate a new cycle if the bit is “H”.

#### *5.2.1.1. Write Cycle Timing*

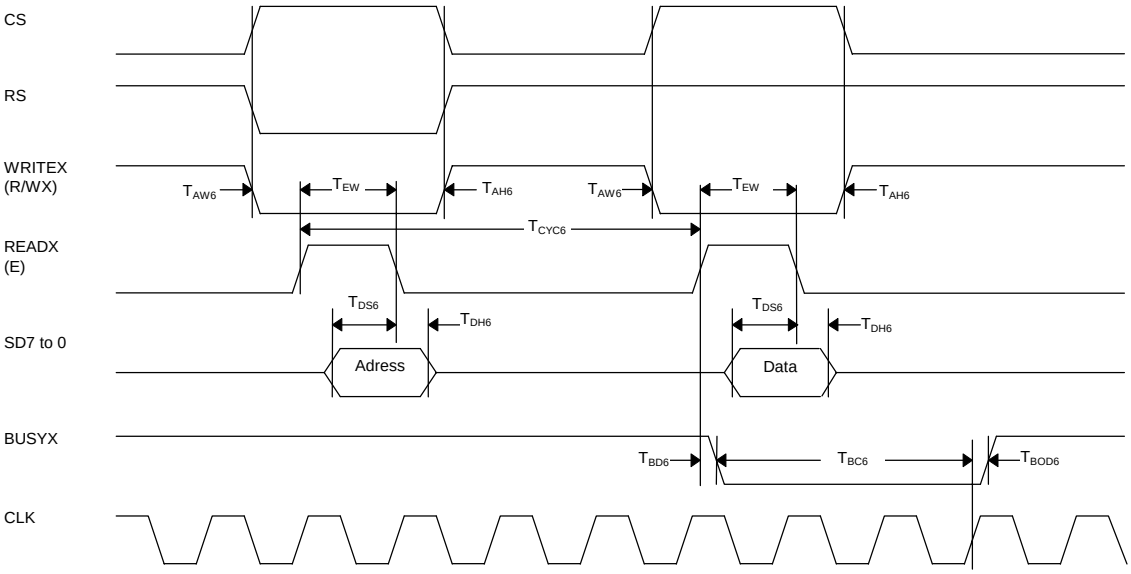



**Figure 5-4 68 Family MPU Write Timing**

| Symbol | Description | Min | Max | Notes |
|---|---|---|---|---|
| $T_{CYC6}$ | System Cycle Time | 100 ns | - | |
| $T_{AH6}$ | Address Hold Time | 20ns | - | |
| $T_{AW6}$ | Address Setup Time | 20ns | - | |
| $T_{DS6}$ | Data Setup time | 20ns | - | |
| $T_{DH6}$ | Data Hold Time | 20 ns | - | |
| $T_{EW}$ | Enable Pulse Width | 40 ns | 1.9CLK | |
| $T_{BD6}$ | BUSYX Delay Time | - | 30ns | CL=80pF |
| $T_{BC6}$ | BUSYX Pulse Width | 2CLK | - | |
| $T_{BOD6}$ | BUSYX Output Disable Time | - | 30ns | CL=80pF |

NOTES:
- CLK is the clock of S-7600A
- Timing is specified of 50% of the signal waveform.
- Rise/fall time(20%,80%) of the input signal is 15nsec or less.

**Table 5-3 68k Family MPU Write Cycle Timing**

### 5.2.1.2. Read Cycle Timing

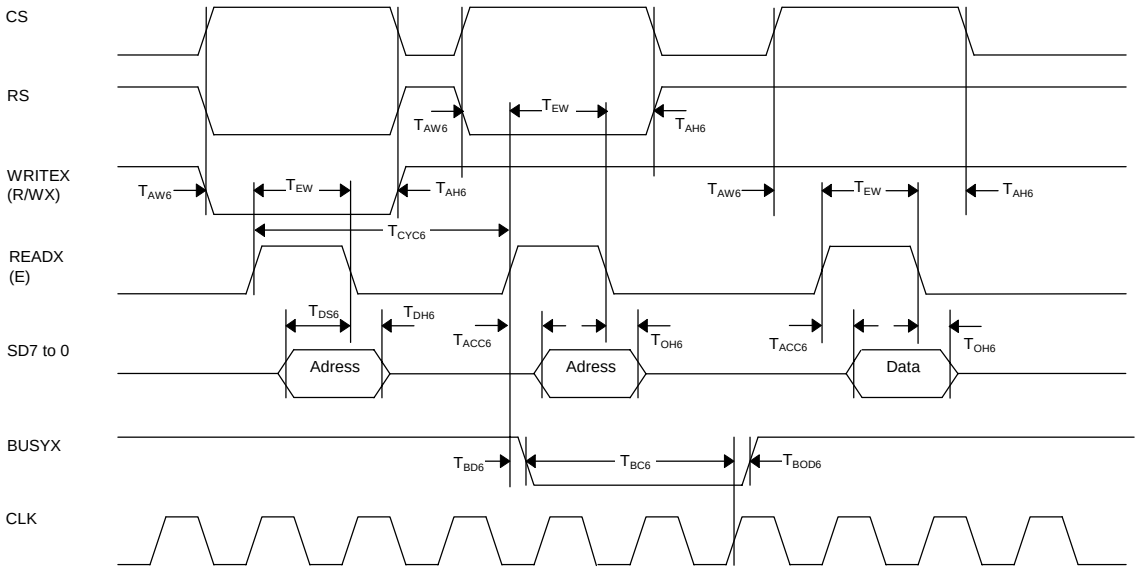



**Figure 5-5 68 Family MPU Read Timing**

| Symbol | Description | Min | Max | Notes |
|---|---|---|---|---|
| $T_{CYC6}$ | System Cycle Time | 100 ns | - | |
| $T_{AH6}$ | Address Hold Time | 20ns | - | |
| $T_{AW6}$ | Address Setup Time | 20ns | - | |
| $T_{DS6}$ | Data Setup time | 20ns | - | |
| $T_{DH6}$ | Data Hold Time | 20 ns | - | |
| $T_{ACC6}$ | Access time | - | 30ns | CL=80pF |
| $T_{OH6}$ | Output Disable Time | 20 ns | - | CL=80pF |
| $T_{EW}$ | Enable Pulse Width | 40 ns | 1.9CLK | |
| $T_{BD6}$ | BUSYX Delay Time | - | 30ns | CL=80pF |
| $T_{BC6}$ | BUSYX Pulse Width | 2CLK | 3CLK | |
| $T_{BOD6}$ | BUSYX Output Disable Time | - | 30ns | CL=80pF |

NOTES:
- CLK is the clock of S-7600A
- Timing is specified of 50% of the signal waveform.
- Rise/fall time(20%,80%) of the input signal is 15nsec or less.

**Table 5-4 68k Family MPU Read Cycle Timing**

### *5.2.2. x80 Family MPU Mode*

This mode is selected by pulling the **C86** input pin "L" and the **PSX** input pin "H". In this mode, the address and data are muxed onto a single 8-bit bus. All cycles start with the address placed on the bus. This address is then latched internally on the rising edge of **WRITEX**. The **RS** pin "L" indicates that the **WRITEX** strobe is for the address phase. In the next phase, data is either written or read by generating **WRITEX** or **READX** strobe. The MPU interface logic will assert the **BUSYX** signal after **READX** or **WRITEX** strobes are de-asserted. The **BUSYX** signal is de-asserted after the S-7600A complete a read or writes operation. The MPU samples the **BUSYX** signal before starting a new cycle. The MPU can initiate a new cycle after the **BUSYX** signal gets de-asserted.

#### *5.2.2.1. Write Cycle Timing*

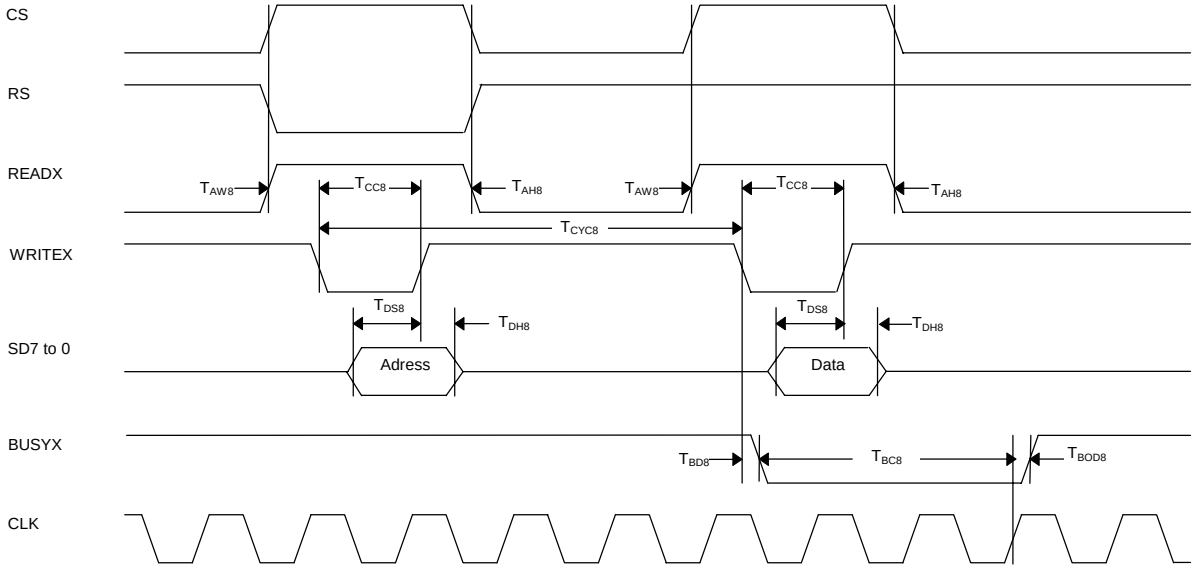



**Figure 5-6 x80 Family MPU Write Cycle Timing**

| Symbol | Description | Min | Max | Notes |
|---|---|---|---|---|
| $T_{CYC8}$ | System Cycle Time | 100 ns | - | |
| $T_{AH8}$ | Address Hold Time | 20ns | - | |
| $T_{AW8}$ | Address Setup Time | 20ns | - | |
| $T_{DS8}$ | Data Setup time | 20ns | - | |
| $T_{DH8}$ | Data Hold Time | 20 ns | - | |
| $T_{CC8}$ | Control Pulse Width | 40 ns | 1.9CLK | |
| $T_{BD8}$ | BUSYX Delay Time | - | 30ns | CL=80pF |
| $T_{BC8}$ | BUSYX Pulse Width | 2CLK | - | |
| $T_{BOD8}$ | BUSYX Output Disable Time | - | 30ns | CL=80pF |

NOTES:
- CLK is the clock of S-7600A
- Timing is specified of 50% of the signal waveform.
- Rise/fall time(20%,80%) of the input signal is 15nsec or less.

**Table 5-5 x80 Family MPU Write Cycle Timing**

### 5.2.2.2. Read Cycle Timing

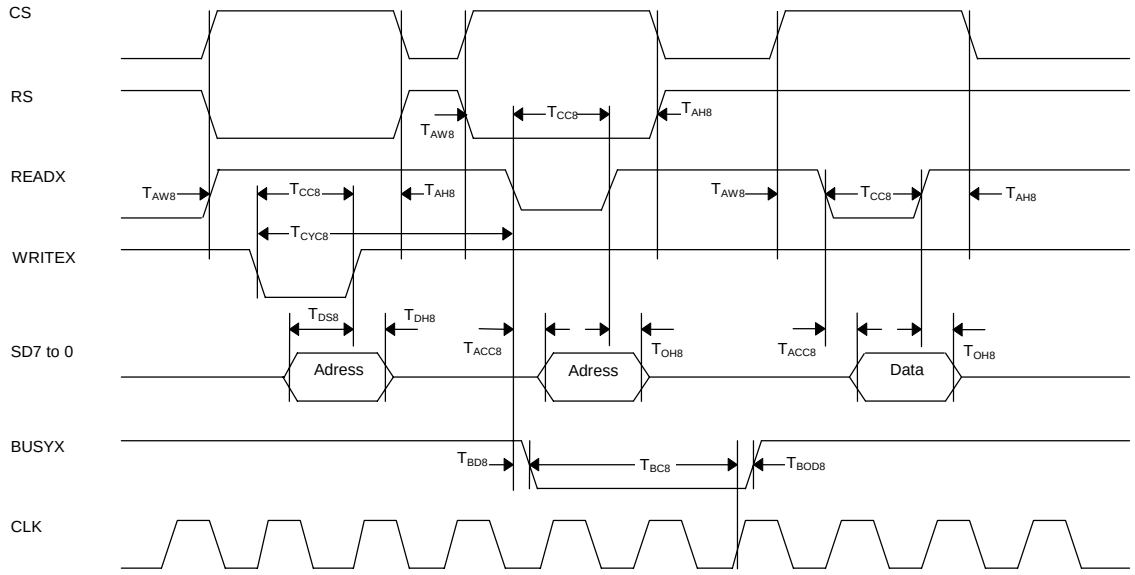



**Figure 5-7 x80 Family MPU Read Cycle Timing**

| Symbol | Description | Min | Max | Notes |
|---|---|---|---|---|
| $T_{CYC8}$ | System Cycle Time | 100 ns | - | |
| $T_{AH8}$ | Address Hold Time | 20ns | - | |
| $T_{AW8}$ | Address Setup Time | 20ns | - | |
| $T_{DS8}$ | Data Setup time | 20ns | - | |
| $T_{DH8}$ | Data Hold Time | 20 ns | - | |
| $T_{ACC8}$ | Access time | - | 30ns | CL=80pF |
| $T_{OH8}$ | Output Disable Time | 20 ns | - | CL=80pF |
| $T_{CC8}$ | Control Pulse Width | 40 ns | 1.9CLK | |
| $T_{BD8}$ | BUSYX Delay Time | - | 30ns | CL=80pF |
| $T_{BC8}$ | BUSYX Pulse Width | 2CLK | - | |
| $T_{BOD8}$ | BUSYX Output Disable Time | - | 30ns | CL=80pF |

NOTES:
- CLK is the clock of S-7600A
- Timing is specified of 50% of the signal waveform.
- Rise/fall time(20%,80%) of the input signal is 15nsec or less.

**Table 5-6 x80 Family MPU Read Cycle Timing**

## 5.3. Serial Interface

This mode is selected by pulling the **PSX** input pin “L”. In this mode Bit 6 of the Data Bus is used as the serial clock and bit 5 and 7 are used as Data Input and Data Output. Bit 0 to 4 are high impedance. By pulling **WRITEX** signal to “H” or “L”, the MPU performs a read or write operation.

### 5.3.1. Write Cycle Timing

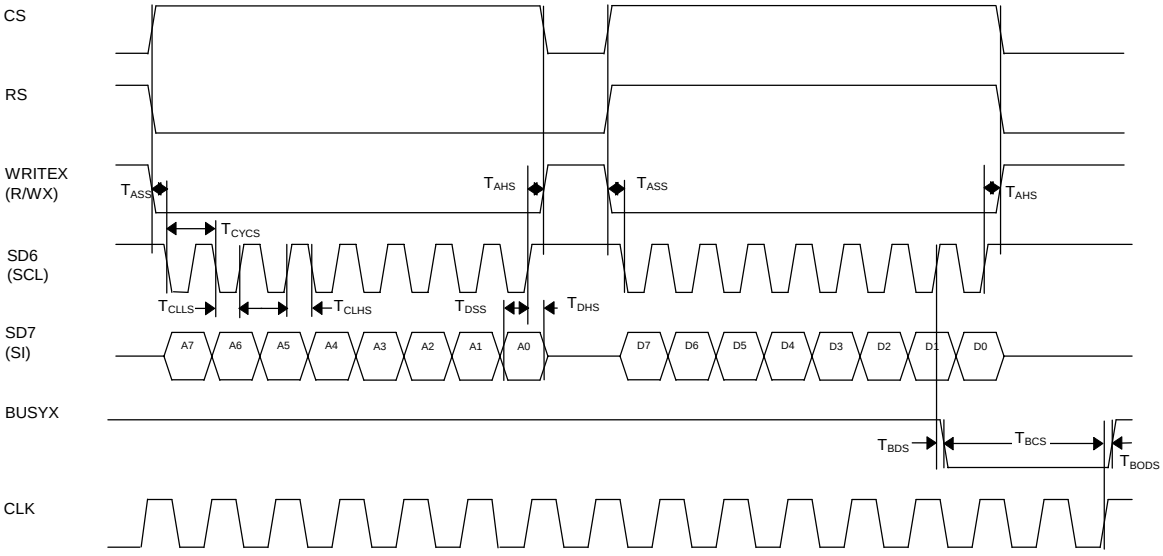



**Figure 5-8 Serial Interface Write Timing**

| Symbol | Description | Min | Max | Notes |
|---|---|---|---|---|
| $T_{CYCS}$ | System Cycle Time | 100 ns | 1.9CLK | |
| $T_{CLLS}$ | Clock L Time | 40ns | - | |
| $T_{CLHS}$ | Clocl H Time | 40 ns | - | |
| $T_{ASS}$ | Address Setup Time | 20ns | - | |
| $T_{AHS}$ | Address Hold Time | 20ns | - | |
| $T_{DSS}$ | Data Setup time | 20ns | - | |
| $T_{DHS}$ | Data Hold Time | 20 ns | - | |
| $T_{BDS}$ | BUSYX Delay Time | - | 30ns | CL=80pF |
| $T_{BCS}$ | BUSYX Pulse Width | 2CLK | - | |
| $T_{BODS}$ | BUSYX Output Disable Time | - | 30ns | CL=80pF |

NOTES:
- CLK is the clock of S-7600A
- Timing is specified of 50% of the signal waveform.
- Rise/fall time(20%,80%) of the input signal is 15nsec or less.

**Table 5-7 Serial Interface Write Cycle Timing**

### 5.3.2. Read Cycle Timing

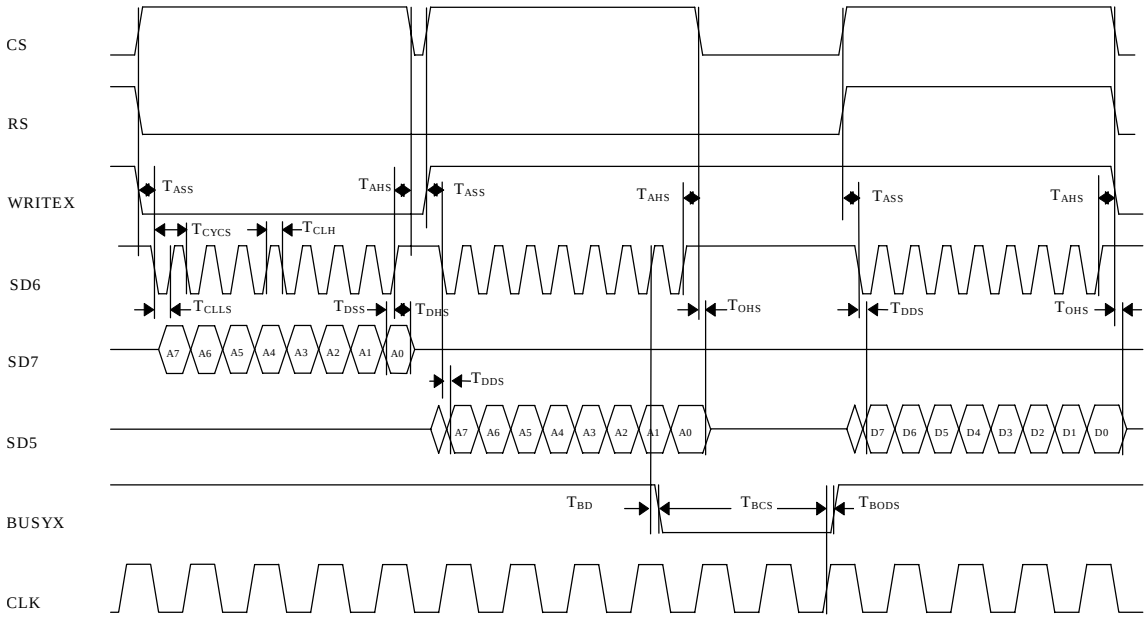



**Figure 5-9 Serial Interface Read Timing**

| Symbol | Description | Min | Max | Notes |
|---|---|---|---|---|
| $T_{CYCS}$ | System Cycle Time | 100 ns | 1.9CLK | |
| $T_{CLLS}$ | Clock L Time | 40ns | - | |
| $T_{CLHS}$ | Clocl H Time | 40 ns | - | |
| $T_{ASS}$ | Address Setup Time | 20ns | - | |
| $T_{AHS}$ | Address Hold Time | 20ns | - | |
| $T_{DSS}$ | Data Setup time | 20ns | - | |
| $T_{DHS}$ | Data Hold Time | 20 ns | - | |
| $T_{DDS}$ | Data delay Time | - | 30ns | CL=80pF |
| $T_{OHS}$ | Output Disable Time | - | 20ns | CL=80pF |
| $T_{BDS}$ | BUSYX Delay Time | - | 30ns | CL=80pF |
| $T_{BCS}$ | BUSYX Pulse Width | 2CLK | - | |
| $T_{BODS}$ | BUSYX Output Disable Time | - | 30ns | CL=80pF |

NOTES:
- CLK is the clock of S-7600A
- Timing is specified of 50% of the signal waveform.
- Rise/fall time(20%,80%) of the input signal is 15nsec or less.

**Table 5-8 Serial Interface Read Cycle Timing**

## 5.4. Interrupt

The interrupt signal outputs an active level while the interrupt flag is set in the interrupt register in the S-7600A's interrupt register. The interrupt signal returns to an inactive level if the flag clears. Show the interrupt timing in the Figure 5-1.
The **INT1** and **INT2X** can be Open Drain or CMOS output depending on the setting of **INTCTL**. The **INT1** and **INT2X** outputs are CMOS if **INTCTL** is "H" otherwise outputs are Open Drain. Table 5-9 defines the interrupt selection.

| Interrupt flag | INTCTL | INT1 | INT2X |
|---|---|---|---|
| Set | H | H | L |
| Set | L | H | L |
| Reset | H | L | H |
| Reset | L | Hi-Z | Hi-Z |

**Table 5-9 Interrupt Selection Table**

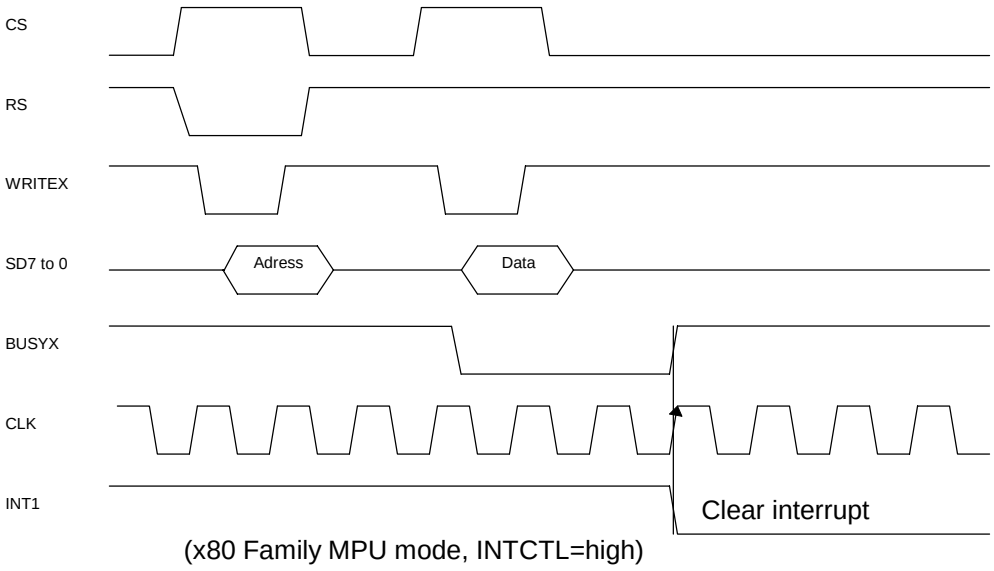



**Figure 5-1 INT1 interrupt timing**

# 6. Memory Requirements

## 6.1. Overview

S-7600A contains too general sockets along with the TCP/UDP/IP and PPP protocols. Their total memory requirement is 10K bytes. This memory is included on the S-7600A chip.

## 6.2. Memory Interface Architecture

The Network Stack feeds all of its memory requests into a single Memory Arbiter inside of the Network Stack core. The arbiter then feeds out one memory request to the SRAM interface. This interface serves to translate the network stack's timing into signal timing required by the SRAM. This architecture is shown in Figure 6-1.

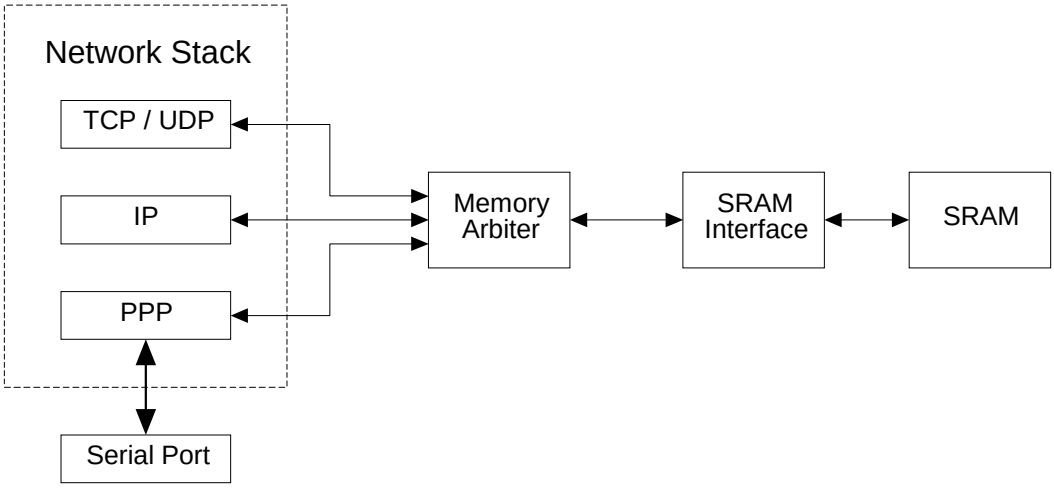



**Figure 6-1 Memory Interface Architecture**

## 6.3. Memory Map

A custom memory map is generated to compact the size of the SRAM required to support S-7600A. S-7600A has two 5K byte memory banks (0 and 1). This mapping is as shown in Table 6-1 and Table 6-2.

**Table 6-1 S-7600A Memory Map (Bank 0)**

| Address | Size | Contents |
|---|---|---|
| 0x0000 - 0x07FF | 2 K | Socket 0 Receive Buffer |
| 0x0800 - 0x0BFF | 1 K | Socket 0 Send Buffer |
| 0x0C00 - 0x0FFF | 1 K | TCP Data Base |
| 0x1000 - 0x13FF | 1 K | IP Buffer |

**Table 6-2 S-7600A Memory Map (Bank 1)**

| Address | Size | Contents |
|---|---|---|
| 0x0000 - 0x07FF | 2 K | Socket 1 Receive Buffer |
| 0x0800 - 0x0BFF | 1 K | Socket 1 Send Buffer |
| 0x0C00 - 0x0FFF | 1 K | PPP Buffer |
| 0x1000 - 0x13FF | 1 K | PAP Buffer |

# 7. S-7600A Register Definitions

## 7.1. Overview

This section covers the S-7600A's API registers. The register are divided into three types: global, direct and indexed.

Global registers occupy the address space from 0x00 to 0x1D and 0x60 to 0x6F. Direct and indexed registers occupy the configuration space from 0x20 to 0x3F. Indexed register require the socket index to be set prior to accessing the registers.

## 7.2. iAPI Register Map

Table 7-1 and Table 7-2 shows the complete iAPI register map for the S-7600A chip. All registers not listed are reserved, and should not be accessed.

**Table 7-1 iAPI Register Map**

| Add | Register | Bit Definitions | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0x00 | Revision | Major Revision Number | | | | Minor Revision Number | | | |
| 0x01 | General_Control | - | - | - | - | - | - | - | SW_RST |
| 0x02 | General_Socket_Location | 0 | 0 | 0 | 0 | 0 | 0 | S1 | S0 |
| 0x04 | Master_Interrupt | - | - | - | - | - | PT_INT | LINK_INT | SOCK_INT |
| 0x08 | Serial_Port_Config | S_DAV | DCD | DSR/HWFC | CTS | RI | DTR | RTS | SCTL |
| 0x09 | Serial_Port_Int | PT_INT | - | - | - | - | - | - | - |
| 0x0A | Serial_Port_Int_Mask | PINT_EN | DSINT_EN | - | - | - | - | - | - |
| 0x0B | Serial_Port_Data | Serial Data Register | | | | | | | |
| 0x0C - 0x0D | BAUD_Rate_Div | BAUD Rate Divider Registers | | | | | | | |
| 0x10 - 0x13 | Our_IP_Address | Our IP Address | | | | | | | |
| 0x1C | Clock_Div_Low | Low Byte for 1 kHz clock divider | | | | | | | |
| 0x1D | Clock_Div_High | High Byte for 1 kHz clock divider | | | | | | | |
| 0x20 | Index | Socket index | | | | | | | |
| 0x21 | TOS* | Type of Service Field | | | | | | | |
| 0x22 | Socket_Config_Status_Low* | TO | Buff_Empty | Buff_Full | Data_Avail/RST | - | Protocol_Type | | |
| 0x23 | Socket_Status_Mid* | URG | RST | Term | ConU | TCP State | | | |
| 0x24 | Socekt_Activate | - | - | - | - | - | - | S1 | S0 |
| 0x26 | Socket_Interrupt | - | - | - | - | - | - | I1 | I0 |
| 0x28 | Socket_Data_Avail | - | - | - | - | - | - | DAV1 | DAV0 |

NOTE: 1)Reserved bits are signified by a dash (-). All reserved bits should be written as “0”.
2)Indexed registers are signified by an asterisk (*).

**Table 7-2 iAPI Register Map (Continued)**

| Add | Register | Bit Definitions | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0x2A | Socket_Interrupt_ Mask_Low* | TO_ En | Buff_ Emp_ En | Buff_F ull | Data_ Avail_ En | - | - | - | - |
| 0x2B | Socket_Interrupt_ Mask_High* | URG_En | RST_ En | Term_ En | ConU En | - | - | - | - |
| 0x2C | Socket_Interrupt_Low* | TO | Buff_ Empty | Buff_ Full | Data_ Avail | - | - | - | - |
| 0x2D | Socket_Interrupt_High* | URG | RST | Term | ConU | - | - | - | - |
| 0x2E | Socket_Data* | Socket 8-bit data | | | | | | | |
| 0x30 | TCP_Data_Send (WO)* | Any write causes data to be sent | | | | | | | |
| 0x30 - 0x31 | Buffer_Out (RO)* | Buffer Out Length | | | | | | | |
| 0x32 - 0x33 | Buffer_In (RO)* | Buffer In Length | | | | | | | |
| 0x34 - 0x35 | Urgent_Data_Pointer* | Urgent Data Offset Pointer, UDP Datagram Size | | | | | | | |
| 0x36 - 0x37 | Their_Port* | Target Port Address | | | | | | | |
| 0x38 - 0x39 | Our_Port* | Our Port Address | | | | | | | |
| 0x3A | Socket_Status_High* | - | - | - | - | - | - | - | Snd _bsy |
| 0x3C - 0x3F | Their_IP_Address* | Target IP Address | | | | | | | |
| 0x60 | PPP_Control_Status | PPP_Int | Con_ Val | Use_ PAP | To_ Dis | PPP_ Int_En | Kick | PPP_ En | PPP_ Up / SRset |
| 0x61 | PPP_Interrupt_Code | Interrupt Code | | | | | | | |
| 0x62 | PPP_Max_Retry | - | | | | PPP Maximum retry | | | |
| 0x64 | PPP_String | Pap user name and password | | | | | | | |

NOTE: 1)Reserved bits are signified by a dash (-). All reserved bits should be written as “0”.
2)Indexed registers are signified by an asterisk (*).

## 7.3. Register Definitions

### 7.3.1. Revision Register (0x00)

(Read-Only, Default 0x2110)

This direct read-only register reports back the design revision. See the design revision form in Table 7-3 and Table 7-4.

**Table 7-3 Revision Register Bit Definitions**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | Major Revision Number | | | | Minor Revision Number | | | |
| Default | 0x21 | | | | 0x10 | | | |

**Table 7-4 Revision Register Description**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 7:4 | Major Revision Number | R | This nibble indicates the major revision number for the S-7600A core. |
| 3:0 | Minor Revision Number | R | This nibble indicates the minor revision number for the S-7600A core. |

### 7.3.2. General Control Register (0x01)

(Read/Write, Default 0x00)

This direct register contains the master software reset. See the register format in Table 7-5 and Table 7-6.See the wave format in figure 9.-2.

**Table 7-5 General Control Register Bit Definitions**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | - | - | - | - | - | - | - | SW_RST |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

NOTE: Reserved bits are signified by a dash (-). All reserved bits should be written as “0”.

**Table 7-6 General Control Register Description**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 0 | SW_RST | R/W | Software Reset.<br>This active high reset returns the S-7600A core to power-on reset settings. It is self-clearing and does not need to be written to “0” for proper operations.<br>0 = Normal operation<br>1 = Soft reset |

### 7.3.3. Generic Socket Location Register (0x02)

(Read-Only)

This register is used to report back the location of general sockets to the software layer. Only bits [1:0] will be set because the S-7600A chip is equipped with two general sockets.

**Table 7-7 Generic Socket Location Register Bit Definitions**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Def.** | S7 | S6 | S5 | S4 | S3 | S2 | S1 | S0 |
| **Value** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

**Table 7-8 Generic Socket Location Register Description**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 7 | S7 | R | Not available |
| 6 | S6 | R | Not available |
| 5 | S5 | R | Not available |
| 4 | S4 | R | Not available |
| 3 | S3 | R | Not available |
| 2 | S2 | R | Not available |
| 1 | S1 | R | General socket 1 available |
| 0 | S0 | R | General socket 0 available |

### 7.3.4. Master Interrupt (0x04)

(Read-Only, Default 0x00)

This direct register indicates the source of the S-7600A interrupt.

**Table 7-9 Master Interrupt Register Bit Definitions**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Def.** | - | - | - | - | - | PT_INT | LINK_INT | SOCK_INT |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

NOTE: Reserved bits are signified by a dash (-). All Reserved bits should be written as “0”.

**Table 7-10 Master Interrupt Register Descriptions (Continued)**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 2 | PT_INT | R | Physical Transport Interrupt<br>The physical transport triggers this interrupt. An application should check the *Serial Port Int* register to determine the actual cause of the interrupt. |
| 1 | LINK_INT | R | Link Layer Interrupt<br>The link layer triggers this interrupt. An application should check the *PPP Interrupt Code* register to determine the actual cause of the interrupt. |
| 0 | SOCK_INT | R | Socket Interrupt<br>One of the sockets that need servicing causes this interrupt. An application should check the *Socket Interrupt* register to determine the actual socket number. |

### *7.3.5. Serial Port Configuration / Status Register (0x08)*

(Read/Write, Default 0X0XX110B)

This register configures the serial port as shown in Table 7-11 and Table 7-12.

**Table 7-11 Conf Status Register Bit Definitions**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Def.** | S_DAV | DCD | DSR/ HWFC | CTS | RI | DTR | RTS | SCTL |
| **Default** | 0 | - | 0 | - | - | 1 | 1 | 0 |

**Table 7-12 Conf Status Register Description**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 7 | S_DAV | R/W | Serial Port Data Available<br>When read, bit indicates that Serial Port data is available.<br>This bit should be written 0. |
| 6 | DCD | R/W | Carrier Detect<br>This bit reflects the current state of the **DCD** bit on the serial port. It is independent of the **SCTL** bit setting.<br>This bit should be written 0. |
| 5 | DSR / HWFC | R/W | Data Send Ready / Hardware Flow Control<br>When read, this bit reflects the current state of the **DSR** bit on the serial port.<br>When this bit written:<br>0 = Hardware Flow control is deactivated<br>1 = Hardware Flow control activated<br>Refer to Chapter 8 for more information about Hardware Flow Control. |
| 4 | CTS | R | Clear To Send<br>This read-only bit reflects the current state of the **CTS** bit on the serial port. It is independent of the **SCTL** bit setting. |
| 3 | RI | R | Ring Indicator<br>This read-only bit reflects the current state of the **RI** bit on the serial port. It is independent of the **SCTL** bit setting. |
| 2 | DTR | R/W | Data Terminal Ready<br>Reading this bit follows the current state of the **DTR** bit on the serial port. The MPU can control the **DTR** by writing to this bit. |
| 1 | RTS | R/W | Request To Send<br>Reading this bit follows the current state of the **RTS** bit on the serial port. The MPU can control the **RTS** by writing to this bit. |
| 0 | SCTL | R/W | Serial Port Control<br>This bit determines who controls the serial port. When this bit is low (default), the MPU controls the port. When the **SCTL** bit is high, the network stack controls the serial port.<br>0 = MPU controls port<br>1 = Hardware controls port |

### 7.3.6. Serial Port Interrupt Register (0x09)

(Read-Only, Default 0X000000B)

This register indicates the state of the serial port interrupt.

**Table 7-13 Serial Port Interrupt Register Bit Definitions**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Def.** | PT_INT | - | - | - | - | - | - | - |
| **Default** | 0 | - | 0 | 0 | 0 | 0 | 0 | 0 |

NOTE: Reserved bits are signified by a dash (-). All Reserved bits should be written as “0”.

**Table 7-14 Serial Port Interrupt Register Description**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 7 | PT_INT | R | Port Transport Interrupt<br>This bit indicates when the serial port interrupt is active. This condition depends on the states of the PINT_EN and DSINT_EN bits in the Serial Port Interrupt Mask Register.<br>When PINT_EN is 1, an interrupt will occur whenever data is available in the serial port input FIFO ("S_DAV" in the Serial Port Configuration/Status Register is 1).<br>When DSINT_EN is 1, an interrupt will be active whenever the CPU can write to the Serial Port Data Register to transmit a byte of data.<br>If both PINT_EN and DSINT_EN are enabled, the interrupt will be active if either condition is met. |

### 7.3.7. Serial Port Interrupt Mask Register (0x0A)

(Read/Write, Default 0x00)

This register enables the serial port interrupts. The default for this register is 0x00 (interrupts disabled).

**Table 7-15 Serial Port Interrupt Mask Register Bit Definitions**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Def.** | PINT_EN | DSINT_EN | - | - | - | - | - | - |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

NOTE: Reserved bits are signified by a dash (-). All Reserved bits should be written as “0”.

**Table 7-16 Serial Port Interrupt Mask Register Description**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 7 | PINT_EN | R/W | Port Interrupt Enable<br>This is the enable for the port interrupt. |
| 6 | DSINT_EN | R/W | Data sent interrupt Enable.<br>This is enable for the data sent interrupt. |

### 7.3.8. Serial Port Data Register (0x0B)

(Read/Write)

This register sends data to and reads data from the serial port UART. The data is valid when the **S_DAV** bit in the *Serial Port Config* register is set. Data can be written to this register when the **PT_INT** bit in the *Serial Port Interrupt* register is set. See the register description in Table 7.-14.

*Note: This register should only be used when the **SCTL** bit in the Serial Port config register is low.*

### 7.3.9. BAUD Rate Divider Registers (0x0C-0x0D)

(Read/Write, Default 0x0000)

These registers set the BAUD rate for the serial port. Calculate the value by using the following formula:

Program Value = [(clk Frequency) / (BAUD Rate)] - 1

Where clk is the clock for the S-7600A core

Example: The clock rate of the S-7600A is 256 KHz and a BAUD rate of 64 Kbps is desired, the programmed value should be:

(256 KHz / 64 k) - 1 = 4 - 1 = 3

*Note: The lowest value that should be programmed into these registers is 0x0003.*

### 7.3.10. Our IP Address Registers (0x10-0x13)

(Read/Write, Default 0x00000000)

These registers store our IP address or the IP address of the local device. The 0x10 register stores the least significant byte and the 0x13 register stores the most significant byte. If the system controller dose not write an IP address, it will be negotiated for during PPP negotiations (floating IP address). When a PPP connection is established (indicated by bit 0, register 60) these registers can be read to query the IP address obtained.

**Table 7-17 Our IP Address Register Bit Definitions (0x10)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | Least significant byte of the local IP address | | | | | | | |
| Default | 0x00 | | | | | | | |

**Table 7-18 Our IP Address Register Bit Definitions (0x11)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | 3rd byte of the local IP address | | | | | | | |
| Default | 0x00 | | | | | | | |

**Table 7-19 Our IP Address Register Bit Definitions (0x12)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | 2nd byte of the local IP address | | | | | | | |
| Default | 0x00 | | | | | | | |

**Table 7-20 Our IP Address Register Bit Definitions (0x13)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | Most significant byte of the local IP address | | | | | | | |
| Default | 0x00 | | | | | | | |

### *7.3.11. Clock Divider Registers (0x1C-0x1D)*

(Read/Write, Default 0x03E7)

These registers program the 1kHz clock generator. This clock is used internally for various S-7600A timing functions. The following equation determines the value programmed into these registers:

(clk Freq/1 kHz) - 1 = Divide Count

Where clk Freq is S-7600A clock frequency. Therefore, for a 1 MHz clock, the divide count equals 1M / 1kHz - 1= 999 = 0x03e7.

### *7.3.12. Index Register (0x20)*

(Read/Write, Default 0x00)

This register must be programmed prior to accessing indexed socket registers. Valid programmed values are 0x00 and 0x01. If the socket number has not changed since the last access, this register not need to be reprogrammed.

**Table 7-21 Index Register Bit Definition**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | Socket Index [7:0] | | | | | | | |
| Default | 0x00 | | | | | | | |

**Table 7-22 Index Register Description**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 7:0 | Socket_Index | R/W | 0x00 : General Socket 0 Selected<br>0x01: General Socket 1 Selected<br>All other values are reserved |

### *7.3.13. Type of Service Register (TOS) (0x21)*

(Read/Write, Default 0x00)

This register configures the TOS field in the IP header for outgoing datagrams. It is an optional setting that defaults to 0x00.

### 7.3.14. Socket Config Status Low Register (0x22)

(Read/Write, Default 0x00)

This register configures the socket.

**Table 7-23 Socket Config Status Low Register Bit Definitions**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Def.** | TO | Buff_ Empty | Buff_Full | Data_ Avail / RST | - | Protocol_Type | | |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | | |

NOTE: Reserved bits are signified by a dash ( - ). All reserved bits should be written as “0”.

**Table 7-24 Socket Config Status Low Register Description**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 7 | TO | R | TCP Timeout<br>This bit indicates that a TCP timeout condition occurred while attempting to establish a TCP connection or while waiting for a TCP packet after the connection was established.<br>0 = Normal Operating Condition<br>1 = Timeout Occurred |
| 6 | Buff_Empty | R | This bit indicates whether or not a socket's outgoing data buffer is empty. The bit sets on an empty condition. It then clears and remains clear as long as there is any data in the socket's outgoing data buffer.<br>0 = Buffer Not Empty<br>1 = Buffer Empty |
| 5 | Buff_Full | R | This bit indicates whether space is available to write data. It also triggers an interrupt when the outgoing data buffer is full, and the **Buff_Full_En** bit in the *Socket Interrupt Mask Low* register (0x2A) is set. The Data Register should not be written to when this bit is a "1".<br>0 = Buffer Space Available<br>1 = No Buffer Space Available |
| 4 | Data_Avail / RST | R/W | Writing this bit resets all socket parameters to default settings. It is self-clearing and dose not need to be written to low for proper operations. Before resetting, ensure that **Snd_Bsy** bit of *Socket Status High* register (0x3A) is 0. When read, this bit indicates that the socket has data available. |
| 2:0 | Protocol_Type | R/W | These bits are used to set the protocol of the socket. All decodes not shown are reserved.<br>010 = TCP Client Mode<br>101 = UDP Mode<br>110 = TCP Server mode |

### 7.3.15. Socket Status Mid Register (0x23)

(Read-Only, Default 0x00)

This read-only register reports other socket status conditions.

**Table 7-25 Socket Status Mid Register Bit Definitions**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | URG | RST | Term | ConU | TCP State | | | |
| Default | 0 | 0 | 0 | 0 | 0x0 | | | |

**Table 7-26 Socket Status Mid Register Description**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 7 | URG | R | This bit indicates the arrival of urgent data. Writing a “1” to the **URG** bit in the *Socket Interrupt* register (bit 7) clears this bit.<br>0 = No urgent data present<br>1 = Urgent data present |
| 6 | RST | R | This bit indicates when the socket receives the RST signal from the TCP peer.<br>0 = No RST received<br>1 = RST received |
| 5 | Term | R | This bit indicates when the socket terminates from the source and triggers an interrupt if the **Term_En** bit is set in the *Socket Interrupt Mask High* register (0x2B). The interrupt mask setting does not effect the reporting of this status bit.<br>0 = Normal Operating Condition<br>1 = Socket terminated from source<br>This bit becomes “1” when the S-7600A receives a TCP segment with the **FIN** flag on. This means that the remote peer has requested to close the TCP connection. |
| 4 | ConU | R | This bit indicates when the socket establishes a connection to a host machine. The bit clears when the connection terminates (by either end).<br>0 = No Connection Established<br>1 = Connection Established |
| 3:0 | TCP State | R | These bits indicate the current TCP state.<br>0 = CLOSED<br>1 = SYN_SENT<br>2 = ESTABLISHED<br>3 = CLOSE_WAIT<br>4 = LAST_ACK<br>5 = FIN_WAIT1<br>6 = FIN_WAIT2<br>7 = CLOSING<br>8 = TIME_WAIT<br>9 = LISTEN<br>a = SYN_RECVD |

### 7.3.16. Socket Activate Register (0x24)

(Read/Write, Default 0x00)

This register is used to activate the sockets and also show the current status of each socket. Setting a bit to “1” activates the corresponding socket. This register defaults to 0x00 upon resets.

**Table 7-27 Socket Activate Register Bit Definitions**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | - | - | - | - | - | - | S1 | S0 |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 7-28 Socket Activate Register Description**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 1 | S1 | R/W | This bit is used to activate general socket 1.<br>0 = General socket 1 inactive<br>1 = General socket 1 active |
| 0 | S0 | R/W | This bit is used to activate general socket 0.<br>0 = General socket 0 inactive<br>1 = General socket 0 active |

### 7.3.17. Socket Interrupt Register (0x26)

(Read-Only, Default 0x00)

This register indicates which socket has interrupts pending. When identification of an interrupting socket occurs, the actual source of the interrupt is determined by examining the specific socket’s interrupt register.

**Table 7-29 Socket Interrupt Register Bit Definitions**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | - | - | - | - | - | - | I1 | I0 |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 7-30 Socket Interrupt Register Description**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 1 | I1 | R | This bit is used to indicate that socket 1 has an interrupt pending.<br>0 = General socket 1 interrupt inactive<br>1 = General socket 1 interrupt active |
| 0 | I0 | R | This bit is used to indicate that socket 0 has an interrupt pending.<br>0 = General socket 0 interrupt inactive<br>1 = General socket 0 interrupt active |

### *7.3.18. Socket Data Available Register (0x28)*

(Read-Only, Default 0x00)

This read-only register indicates which socket has data pending in the input buffer. A “1” in a bit position indicates that the socket has data available. The bit remains set as long as there is data available.

**Table 7-31 Socket Data Avail Register Bit Definitions**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Def.** | - | - | - | - | - | - | DAV1 | DAV0 |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 7-32 Socket Data Avail Register Description**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 1 | DAV1 | R | This bit is used to indicate that socket 1 has data available.<br>0 = General socket 1 has no data available<br>1 = General socket 1 has data available |
| 0 | DAV0 | R | This bit is used to indicate that socket 0 has data available.<br>0 = General socket 0 has data available<br>1 = General socket 0 has data available |

### 7.3.19. Socket Interrupt Mask Low Register (0x2A)

(Read/Write, Default 0x00)

This register reports certain interrupt conditions. Setting a bit enables the corresponding interrupt.

**Table 7-33 Socket Interrupt Mask Low Register Bit Definitions**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Def.** | TO_En | Buff_ Emp_En | Buff_Full_En | Data_Avail_En | - | - | - | - |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

NOTE: Reserved bits are signified by a dash (-). All reserved bits should be written as “0”.

**Table 7-34 Socket Interrupt Mask Low Register Description**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 7 | TO_En | R/W | Writing a “1” enables the Timeout interrupt. |
| 6 | Buff_Empty_En | R/W | Writing a “1” enables the Buffer Empty interrupt. |
| 5 | Buff_Full_En | R/W | Writing a “1” enables the Buffer Full interrupt. |
| 4 | Data_Avail_En | R/W | Writing a “1” enables the Data Available interrupt. |

### 7.3.20. Socket Interrupt Mask High Register (0x2B)

(Read/Write, Default 0x00)

This register enables certain types of interrupt conditions. Setting bits enables their corresponding interrupts.

**Table 7-35 Socket Interrupt Mask High Register Bit Definitions**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Def.** | URG_En | RST_En | Term_En | ConU_En | - | - | - | - |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

NOTE: Reserved bits are signified by a dash (-). All reserved bits should be written as “0”.

**Table 7-36 Socket Interrupt Mask High Register Description**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 7 | URG_En | R/W | Writing a “1” to enable the Urgent Data interrupt. |
| 6 | RST_En | R/W | Writing a “1” to enable the Connection Reset interrupt. |
| 5 | Term_En | R/W | Writing a “1” to enable the Socket Termination interrupt. |
| 4 | ConU_En | R/W | Writing a “1” to enable the Connection Up interrupt. |

### 7.3.21. Socket Interrupt Low Register (0x2C)

(Read/Write, Default 0x00)

This register reports certain interrupt conditions. When an interrupt condition occurs and its enable bit is set, the hardware sets the corresponding bit. Writing a "1" to the bit clears it. Disabling the corresponding enable bit prevents the interrupt from showing.

**Table 7-37 Socket Interrupt Low Register Bit Definitions**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Def.** | TO | Buff_Emplty | Buff_Full | Data_Avail | - | - | - | - |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

NOTE: Reserved bits are signified by a dash (-). All reserved bits should be written as “0”.

**Table 7-38 Socket Interrupt Low Register Description**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 7 | TO | R/W | This interrupt is generated when a timeout condition occurred while trying to establish a connection. Writing a “1” to this bit clears the interrupt. |
| 6 | Buff_Empty | R/W | This interrupt is generated when outgoing buffer is empty. Writing a “1” to this bit clears the interrupt. |
| 5 | Buff_Full | R/W | This interrupt is generated when there is more buffer space available. Writing a “1” to this bit clears the interrupt. |
| 4 | Data_Avail | R/W | This interrupt is generated when data is available from the incoming buffer. Writing a “1” to this bit clears the interrupt. |

### 7.3.22. Socket Interrupt High Register (0x2D)

(Read/Write, Default 0x00)

This register reports certain interrupt conditions. When an interrupt condition occurs and its enable bit is set, the hardware sets the corresponding bit. Writing a "1" to the bit clears it. Disabling the corresponding enable bit prevents the interrupt from showing.

**Table 7-39 Socket Interrupt High Register Bit Definitions**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Def.** | URG | RST | Term | ConU | - | - | - | - |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

NOTE: Reserved bits are signified by a dash (-). All reserved bits should be written as “0”.

**Table 7-40 Socket Interrupt High Register Description**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 7 | URG | R/W | This interrupt is generated when urgent data arrives. The system interface should read the *Urgent Data Pointer* register to see the location of the data. Writing a “1” to this bit clears the interrupt. |
| 6 | RST | R/W | This interrupt is generated when a TCP peer sends the socket RST flag indicating that the current TCP session is not valid. Writing a “1” to this bit clears this interrupt. When this condition occurs, the hardware no longer operates and re-initializing the socket is recommended. |
| 5 | Term | R/W | This interrupt is generated when the socket connection is terminated and a TCP FIN flag is received. Writing a “1” to this bit clears the interrupt. |
| 4 | ConU | R/W | This interrupt is generated when a connection is established. Writing a “1” to this bit clears the interrupt. |

### *7.3.23. Socket Data Register (0x2E)*

(Memory Mapped Read/Write, Default 0x00)

This register is used by a system controller to read incoming data packets and write outgoing data. Data transmissions start for TCP connections only after a write occurs at 0x30.

### *7.3.24. TCP Data Send and Buffer Out Length Registers (0x30 - 0x31)*

(Read/Write, Default 0x03FF)

When read, these registers report the amount of space available in the outgoing buffer. Register 0x30 stores the least significant byte; 0x31 stores the most significant byte. Writing any data to 0x30 causes data transmissions to start on TCP connections.

### *7.3.25. Buffer In Length Registers (0x32-0x33)*

(Read-Only, Default 0x0000)

These read-only registers report the amount of data available in the received data buffer. 0x32 stores the least significant byte; 0x33 stores the most significant byte.

### *7.3.26. Urgent Pointer / UDP Datagram Size Registers (0x34-0x35)*

(Read-Only, Default 0x0000)

These read-only registers report the offset to the start of urgent data (as marked through the TCP header) relative to the incoming data buffer. Register 0x34 stores the least significant byte; 0x35 stores the most significant byte. When a socket is configured as a UDP socket, these registers indicate the size of the current UDP datagram. The least significant byte is stored in 0x34 and the most significant byte is stored in 0x35.

### 7.3.27. Their Port Registers (0x36-0x37)

(Read/Write, Default 0x0000)

These registers specify the destination port for an outgoing data packets. For client mode, this value must be set prior to activating the socket. For TCP server mode, these register are automatically setup on a connection with the peer's port number. Register 0x36 stores the least significant byte and 0x37 stores the most significant byte.

**Table 7-41 Their Port Register Bit Definitions (0x36)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | Least significant byte of the target port number | | | | | | | |
| Default | 0x00 | | | | | | | |

**Table 7-42 Their Port Register Bit Definitions (0x37)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | Most significant byte of the target port number | | | | | | | |
| Default | 0x00 | | | | | | | |

### 7.3.28. Our Port Registers (0x38-0x39)

(Read/Write)

These registers are used it indicate the source port for an outgoing data packet. When setting a TCP client or sending data using UDP, these registers should be set to the proper value. Normally in client applications, the software increments the value of this register. The TCP and UDP server application should set these registers to be the value used by the server applications. Register 0x38 stores the least significant byte; 0x39 stores the most significant byte.

**Table 7-43 Our Port Register Bit Definitions (0x38)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | Least significant byte of the local port number | | | | | | | |
| Default | 0x00 | | | | | | | |

**Table 7-44 Our Port Register Bit Definitions (0x39)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | Most significant byte of the local port number | | | | | | | |
| Default | 0x00 | | | | | | | |

### 7.3.29. Socket Status High Register (0x3A)

(Read-Only, Default 0x00)

This register reports the busy status of the socket.

**Table 7-45 Socket Status High Register Bit Definitions**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | - | - | - | - | - | - | - | Snd_Bsy |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

NOTE: Reserved bits are signified by a dash (-). All reserved bits should be written as "0".

**Table 7-46 Socket Status High Register Description**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 0 | Snd_Bsy | R | This bit indicates that the current socket is busy sending TCP segments. Before the socket is reset, this bit should be 0.<br>0 = Socket not busy<br>1 = Socket busy |

### *7.3.30. Their IP Address Registers (0x3C-0x3F)*

(Read/Write, Default 0x00000000)

These registers indicate the destination IP address for the socket. The value must be set prior to activating the socket. The registers can be written in any order.

**Table 7-47 Their IP Address Register Bit Definitions (0x3C)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | Least significant byte of Destination IP address | | | | | | | |
| Default | 0x00 | | | | | | | |

**Table 7-48 Their IP Address Register Bit Definitions (0x3D)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | 3rd byte of Destination IP address | | | | | | | |
| Default | 0x00 | | | | | | | |

**Table 7-49 Their IP Address Register Bit Definitions (0x3E)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | 2nd byte of Destination IP address | | | | | | | |
| Default | 0x00 | | | | | | | |

**Table 7-50 Their IP Address Register Bit Definitions (0x3F)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | Most significant byte of Destination IP address | | | | | | | |
| Default | 0x00 | | | | | | | |

### 7.3.31. PPP Control and Status Register (0x60)

(Read/Write, Default 0x00)

This register control the PPP layer and reports its status.

**Table 7-51 PPP Control and Status Register Bit Definitions (0x60)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Def.** | PPP_Int | Con_Val | Use_ PAP | TO_Dis | PPP_Int_En | Kick | PPP_En | PPP_Up /SRst |
| **Default** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 7-52 PPP Control Status Register Description**

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 7 | PPP_Int | R/W | PPP Interrupt<br>This bit indicates that the PPP triggered an interrupt condition. Read the PPP interrupt code register to determine the cause. Writing a “1” to this bit position clears the interrupt. |
| 6 | Con_Val | R/W | Connection Valid<br>This bit indicates to the network stack that the underlying connection is up and valid.<br>0 = Connection down<br>1 = Connection up |
| 5 | Use_PAP | R/W | This bit enables PAP authentication within the PPP protocol. If enabled, a PAP request is issued after PAP authentication is negotiated. The PAP string enters through register 0x64.<br>0 = PAP disabled (default)<br>1 = PAP enabled |
| 4 | TO_Dis | R/W | Timeouts Disabled<br>This bit disables the PPP block from timeouts for diagnostic purposes. It should remain enable for normal operations.<br>0 = Timeouts enabled (default)<br>1 = Timeouts disabled |
| 3 | PPP_Int_En | R/W | PPP Interrupt Enable<br>This bit enables the PPP interrupt.<br>0 = PPP Interrupt disabled (default)<br>1 = PPP Interrupt enabled |
| 2 | Kick | W | PPP Kick Start<br>When written to a 1, this bit will start the PPP if it falls into a timeout condition. It clears once the kick operation performs. This bit is self-clearing. |
| 1 | PPP_En | R/W | PPP Enable<br>This bit enables the PPP layer. The bit must be set before any transmissions occur.<br>0 = PPP disabled (default)<br>1 = PPP enabled |

| Bit | Bit Name | Access | Description |
|---|---|---|---|
| 0 | PPP_UP/SRst | R/W | When read, this bit indicates when the PPP layer establishes a connection.<br>0 = PPP Connection down<br>1 = PPP Connection established<br>When written, this bit will reset the PPP engine. It is self-clearing and goes not need to be written low for normal operations.<br>0 = PPP Normal operation<br>1 = PPP Reset |

### *7.3.32. PPP Interrupt Code (0x61)*

(Read-Only, Default 0x00)

This register indicates the interrupt condition that causes the PPP interrupt to trigger.

**Table 7-53 PPP Interrupt Code Register Bit Definitions**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | PPP Interrupt Code | | | | | | | |
| Default | 0 | | | | | | | |

**Table 7-54 PPP Interrupt Error Codes**

| Error Code | Definition |
|---|---|
| 0x00 | Reserved |
| 0x01 | PPP Failed initial LCP negotiations |
| 0x02 | PPP Failed NCP negotiations |
| 0x03 | Unexpected LCP closure |
| 0x04 | Termination Request received |
| 0x05 | PAP Failed negotiations |

### *7.3.33. PPP Max Retry, (0x62)*

(Read/Write, Default 0x0A)

This register configures the maximum retry number. This number is used to determine the maximum number of configuration requests that are sent during the PPP negotiation stage.

**Table 7-55 PPP Max Retry Register**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Def. | - | | | | PPP Maximum Retry | | | |
| Default | 0x0 | | | | 0xA | | | |

NOTE: Reserved bits are signified by a dash (-).

### 7.3.34. PAP String (0x64)

(Write-Only)

This write-only register enters the string for the PAP configuration request packet. Enter the string according to the format shown Table 7-56.

**Table 7-56 PAP String Format**

| Byte | String |
|---|---|
| [0] | Length of username |
| [1] | First byte of username |
| [2] | Second byte of username |
| [n] | Last byte of username (where n is the length of the username string) |
| [n+1] | Length of password |
| [n+2] | First byte of password |
| [n+m+1] | Last byte of password (where m is the length of the password string) |

As an example, if the username string is “joe” and the password is “public”, enter the bytes as shown in Table 7-57.

**Table 7-57 PAP String Example**

| | | |
|---|---|---|
| byte:0 | 0x03 | Length of username string |
| byte:1 | 0x6a | Character “j” |
| byte:2 | 0x6f | Character “o” |
| byte:3 | 0x65 | Character “e” |
| byte:4 | 0x06 | Length of password string |
| byte:5 | 0x70 | Character “p” |
| byte:6 | 0x75 | Character “u” |
| byte:7 | 0x62 | Character “b” |
| byte:8 | 0x6C | Character “l” |
| byte:9 | 0x69 | Character “l” |
| byte:a | 0x63 | Character “c” |

If PAP is used, the **Use_PAP** bit must be set in the *PPP Control and Status* register (0x60) prior to entering the PAP string.

# 8. Serial Port Interface

## 8.1. Overview

The S-7600A chip contains a on-board serial port for physical transports. The data format of the serial port is fixed at 1 start bit (logic "0"), 8 data bits, 1 stop bit (logic "1") and no parity bits. The data bits are sent out, least significant bit first. This data format is shown in Figure 8-1. Also included with the serial port is a 16-bit Receive FIFO and an 8-bit Send Buffer.

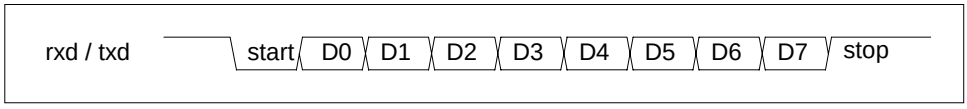



**Figure 8-1 Serial Data Format**

## 8.2. Serial Port Register Map

The following registers are used to communicate with the serial port.

**Table 8-1 Serial Port Register Map**

| Add | Register | Bit Definitions | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0x08 | Serial_Port_Config | S_DAV | DCD | DSR/ HWFC | CTS | RI | DTR | RTS | SCTL |
| 0x09 | Serial_Port_Int | PT_INT | DSINT | - | - | - | - | - | - |
| 0x0A | Serial_Port_Int_ Mask | PINT_ EN | DSINT_ EN | - | - | - | - | - | - |
| 0x0B | Serial_Port_Data | Serial Port Data Register | | | | | | | |
| 0x0C - 0x0D | BAUD_Rate_Div | BAUD Rate Divider Registers | | | | | | | |

### 8.2.1. Hardware Flow Control (RTS/CTS Handshaking)

The Hardware Flow Control is turned off by default. In this mode, data is transmitted independent of the state of the **CTSX** signal. While the MPU is in control of the serial port, it can monitor the state of all the serial port control signals and control when data gets sent or received, either through polling the status bits or interrupts. It can also control the **RTSX** signal by asserting the **RTS** bit in the *Serial Port Config* register. When the S-7600A controls the serial port, data will be sent out as soon as it is available from the PPP layer. When receiving data, the software in the MPU control mode should read the data out of the 16-byte FIFO fast enough to prevent buffer overflow.

Hardware Flow Control can be turned on by writing a "1" to bit 5 (**DSR/HWFC**) of the *Serial Port Config* Register (0x08). With the hardware flow control turned on, full RTS/CTS handshaking is supported. When the serial port detects that **CTS** is de-asserted, it will stop sending data until **CTS** is reasserted. Any byte output at the time **CTS** is de-asserted will complete, but no further bytes will be sent until **CTS** is asserted.

In the other direction, the S-7600A will de-assert **RTS** if the serial port's 16-byte FIFO is half full. This indicates to the machine on the other end of the serial line to stop transmitting data. The **RTS** bit will reassert when the MPU or the S-7600A has read data out of the Receive FIFO and room becomes empty. If the machine communicating with the S-7600A over the serial port does not support RTS/CTS handshaking, the Receive FIFO may overflow and data loss will occur.

### 8.2.2. Serial Port Control

The control of the serial port is turned over to the MPU by default and after any reset condition. In this mode, any data written to the *Serial Port Data* register will be sent out and all data received will be made available to the MPU via this same register. Prior to using the data register, the MPU should set the *BAUD Rate Div* register to the proper setting. An interrupt can be triggered when data is available from the serial port by asserting the **PINT_EN** bit. When this bit is asserted, an interrupt will trigger any time that there is data available to be read from the port. If there is more then one byte in the Receive FIFO, the interrupt will remain active until all bytes are read. An interrupt can also trigger indicating that the outgoing data byte has been sent, by asserting the **DSINT_EN** bit. This interrupt will trigger whenever there is no more data to be sent.

The MPU turns over control to the S-7600A by asserting the **SCTL** bit in the *Serial Port Config* register. When the S-7600A controls the port, the MPU should not access the *Serial Port Data* register. The S-7600A chip will automatically send PPP packets to the serial port and read incoming bytes from the serial port. The serial port interrupts are not valid when the S-7600A controls the port.

# 9. Reset Functions

## 9.1. Overview

The S-7600A has two reset functions which are hardware reset and software reset.

### 9.1.1. Hardware reset function

The S-7600A operates to be synchronous to the **CLK** signal(clock input). When the **RESETX** pin set to low level in one clock period minimum, the S-7600A accept hardware reset input and starts initializing internal circuit at positive edge timing of forth clock. After the **RESETX** pin return to high level, the S-7600A maintains initialized state and turns normal state at positive edge timing of forth clock.

See the Figure 9-1.

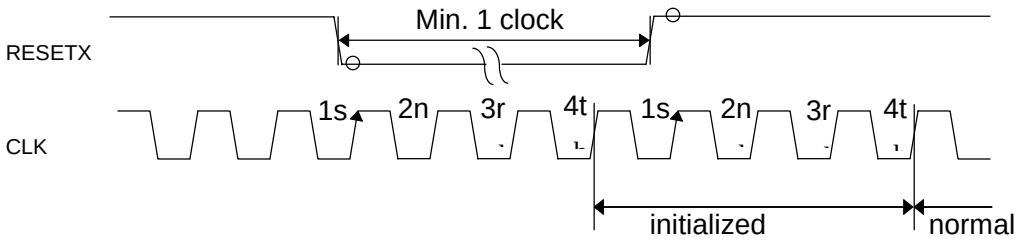
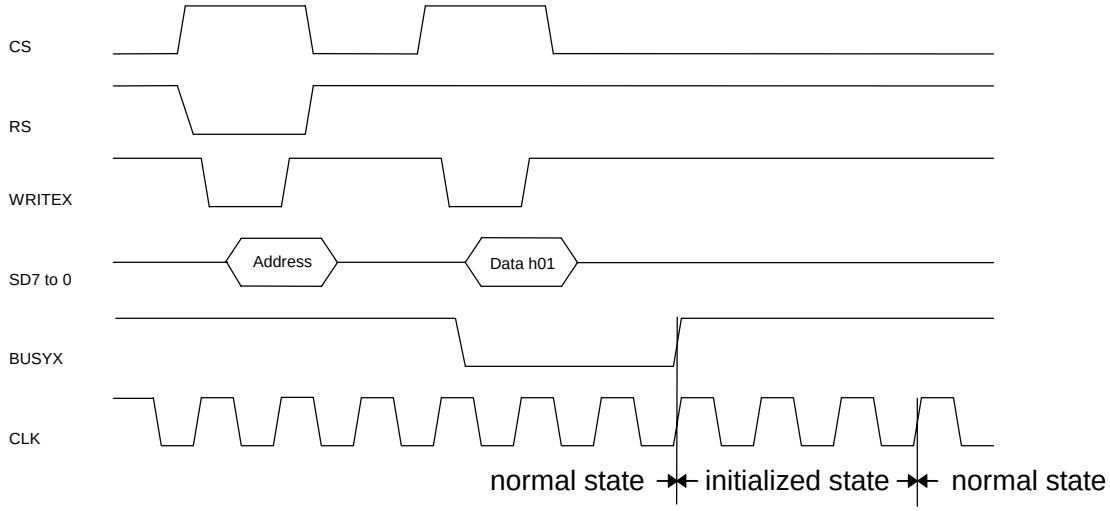



**Figure 9-1 Hardware reset timing**

### 9.1.2. Software reset function

The S-7600A is able to initialize the internal circuit by the *General Control* Register(0x01). Show the reset timing in case of x80 Family MPU mode. See the Figure 9-2.




80 Family MPU mode

**Figure 9-2 Software reset timing**

# 10. Application Examples

## 10.1.1. In case of x80 Family MPU with LCD Controller

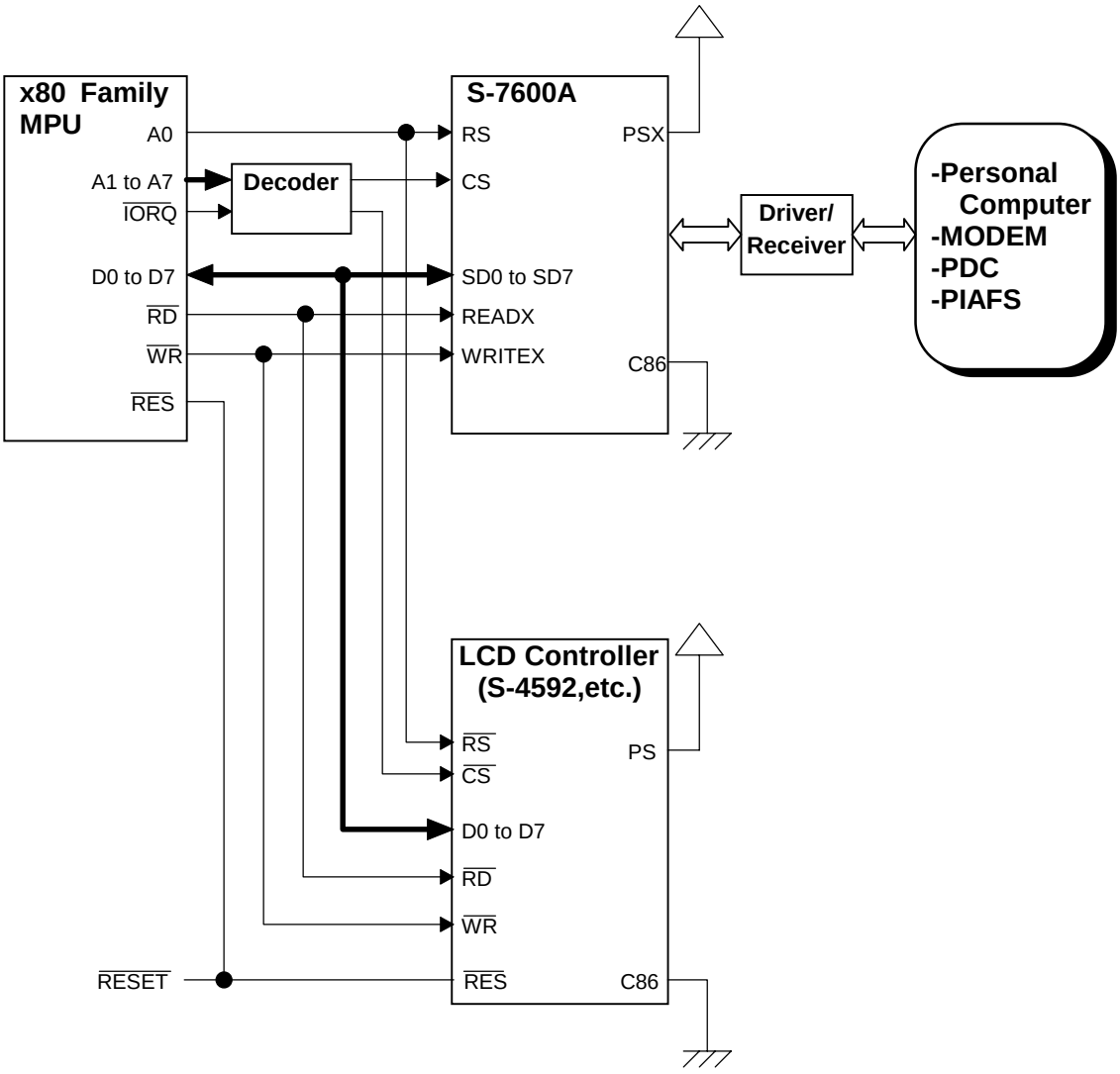



**Figure 10-1 Example for x80 Family MPU**

### 10.1.2. In case of 68k Family MPU with LCD Controller

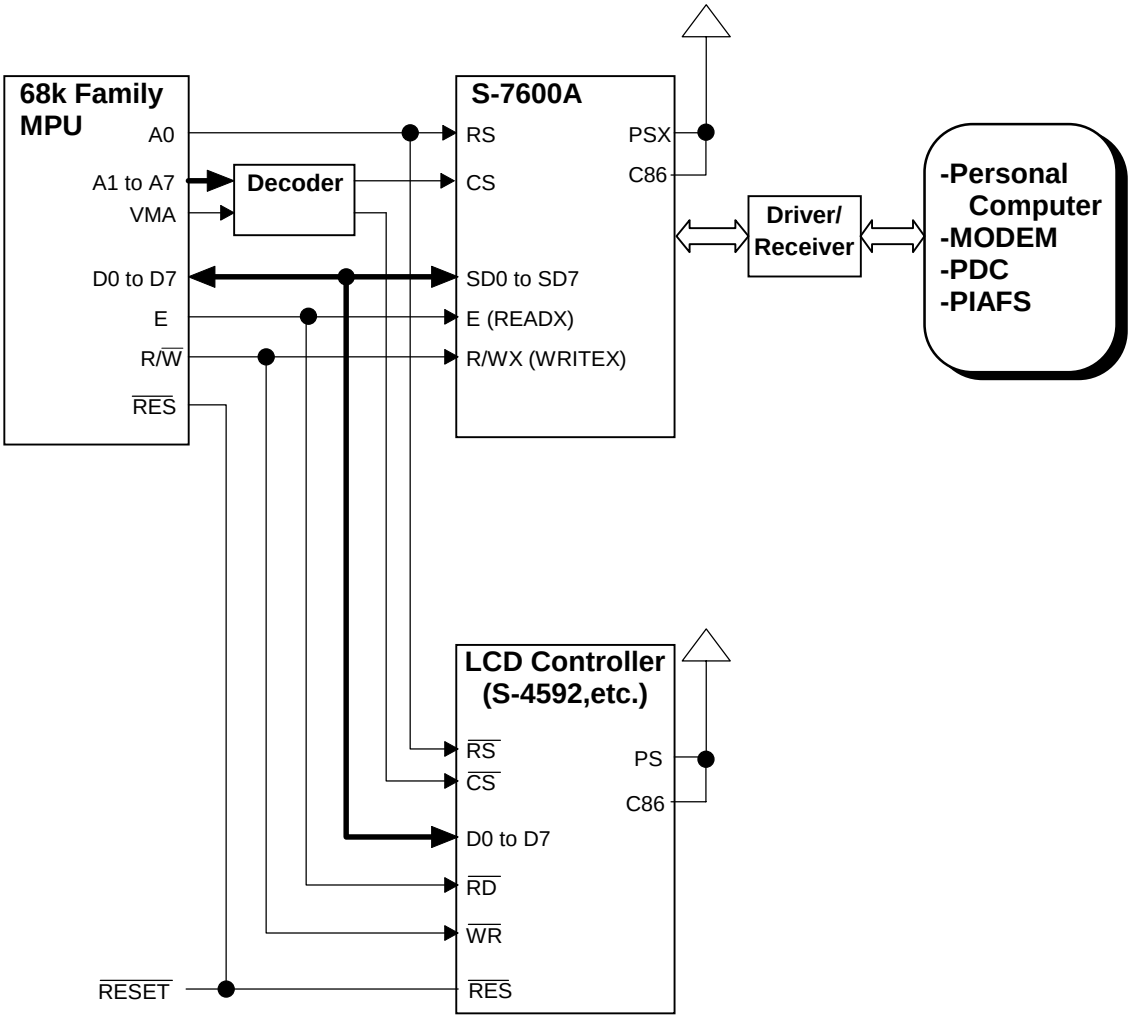



**Figure 10-2 Example for 68k Family MPU**

### 10.1.3. In case of Serial interface with LCD Controller

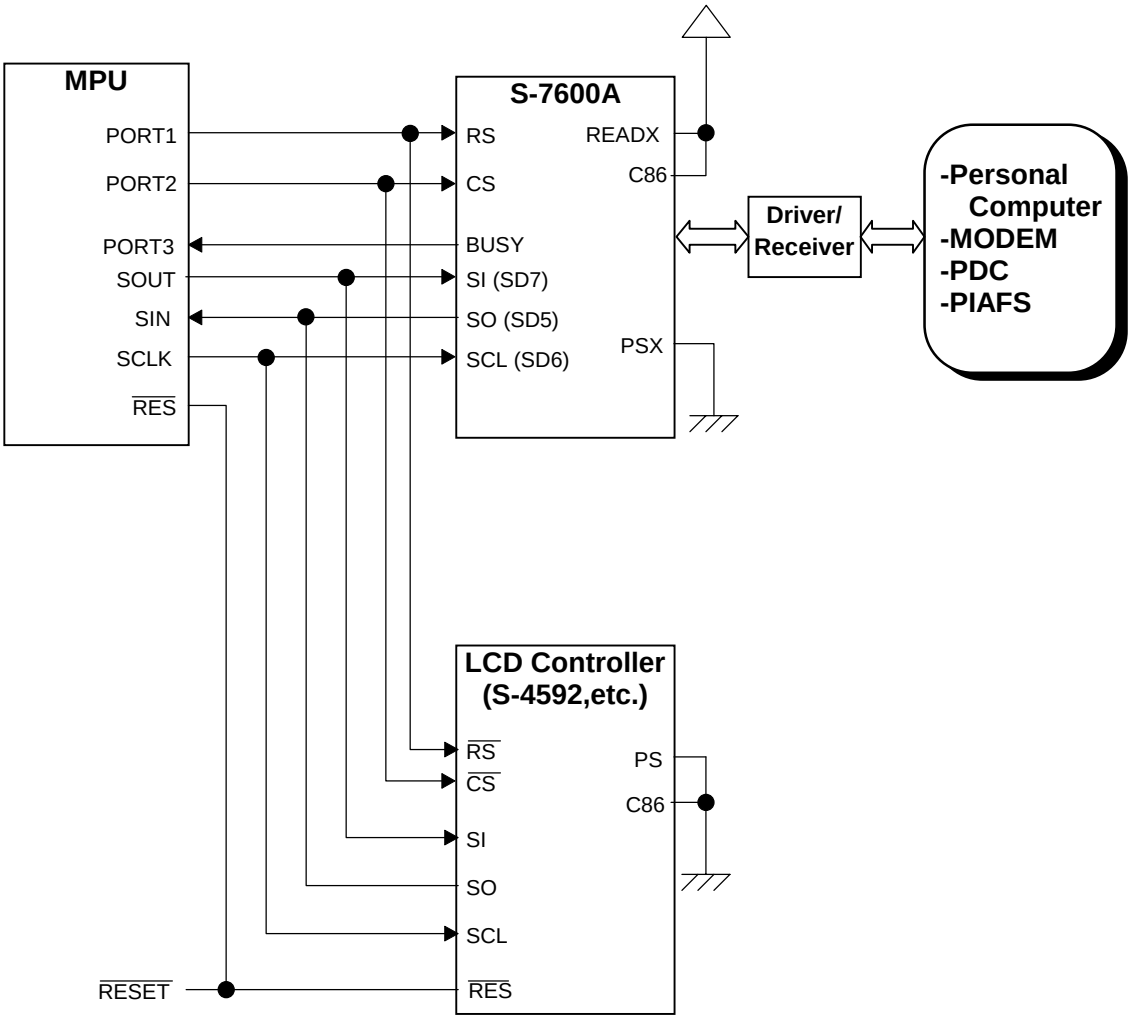



**Figure 10-3 Example for Serial interface**

*Seiko Instruments Inc.*
1-8, Nakase, Mihama-ku, Chiba-shi, Chiba 261, Japan
Components Sales Div.
Telephone : +81-43-211-1196 Facsimile : +81-43-211-8032
E-mail : component@sii.co.jp

*Seiko Instruments USA Inc.*
Electronic Components Div.
2990 W. Lomita Blvd, Torrance, CA 90505, USA
Telephone : +1-909-934-9334 Facsimile : +1-909-975-5699
E-mail : seiko-ecd@salessupport.com
http://www.seiko-usa-ecd.com