INTELLIGENT MASTER-SLAVE I/O SYSTEM WITH COUNTER/TIMER

ADAX, Inc.
737 Dwight Way
Berkeley, CA 94710

TABLE OF CONTENTS

APPENDICES

1. Master-Slave I/O System Overview

The Master-Slave I/O system (MSIO) is a subsystem for the combined facilities of parallel, serial, and real-time counter timer devices. The system has been developed for highly efficient programmed I/O or DMA access through an I/O co-processor connected to the MSIO. The MSIO system consists of one, two, or more boards connected over a ribbon cable for the addition of serial ports if desired. The "master" board contains three parallel ports, three counter-timers, and the necessary logic to connect the optional 4-64 serial USART devices located on the "slave" boards.

These devices are memory mapped and may be addressed on any 1024 byte boundary by the selection of the address decoding switches, 4D and 7D, on the master board. The switch at 4D decodes address lines A23-A16, while the switch at 7D decodes address lines A15-A10. The memory space is divided as follows:

Offset from base decode address:

Hex	Decimal	
0-1ff	0-511	64 USART devices, 8 bytes per USART
200	512	Receive Interrupt Register
204	516	Transmit Interrupt Register
210-216	528-532	8254 Counter/Timer
218-23E	536-574	(reserved for future use)
240-27E	576-638	68230 Parallel Port

The above devices respond to byte-wide memory read or write operations. Specific details will be given in other sections concerning the initialization and operation of each unit.

2. 68230 Parallel Port and 8254 Counter/Timer Programming

The memory map of the MSIO has been designed for optimal programming by 16-bit processors (such as the MC68000) as well as 8-bit processors. To improve throughput all devices are located on even address locations (i.e. $FEA000,$FEA002, etc.). The 68230 parallel port provides a versatile double buffered interface. Many modes are available under program control including: bit I/O, unidirectional 8-bit and 16-bit, and bidirectional 8-bit and 16-bit. Four handshake lines may be selected in conjunction with the above operation modes. The read/write registers of the 68230 are memory-mapped on even addresses. This arrangement facilitates the use of 16-bit word transfer operations (using the "movep.w" instruction for instance). A complete functional description of the 68230 is given in Appendix A.

Programming the 8254 is a relatively simple matter. Again the read/write registers are arranged with increments of two (i.e. offsets: 00,02,04,06 from the 8254 base address). Otherwise the 8254 operates precisely as shown in the data sheets given in APPENDIX B. Special MSIO hardware features enhance the 8254 operation. By providing additional software triggering for interrupt control, many possibilities are available for sophisticated applications in real-time control, scientific data gathering, or operating system timing. These will be discussed in Section 5 with examples given in APPENDIX D.

3. 68661(2661) USART Programming

The four read/write resisters of each 68661(2661) are located at offsets 00,02,04,06 from the base address of a given USART. This arrangement facilitates multiple byte transfers through software command and faster addressing for processors without address bit zero. The first USART is located at offset zero from the address selected by the DIP switches 4B and 4C.

There are two fully independent interrupt scanning circuits in the MSIO which provide information on pending interrupts to the CPU. These circuits indicate which USART requires interrupt service (either for transmit or receive operations). When an interrupt is pending, the circuits may be interrogated to determine which port to service next by reading a resister on the MASTER board. The contents of these byte-wide resisters are a flag bit (bit 7) which when set means an interrupt is pending and USART number (bits 6-0) which represents a number in the range [0,63]. The second quantity when multiplied by eight gives the offset of the USART from the "base" address of the MSIO system. Hence, the USART needing service may be directly accessed without polling or searching. When bit 7 of the interrupt resister is zero (clear), no interrupts from any USART are pending. The MSIO scanning circuits provide round-robin interrupt service to all USARTs without polling or the software overhead of programmable priority encoders. Furthermore, multiple interrupts may be rapidly dispatched within a single entry to an interrupt handling routine.

4. Jumper Selectable Options

A variety of user selectable options are available for the MSIO. They adjust the many features of the MSIO to a particular situation. For instance: selection of vectored interrupt or counter/timer method. The options will be discussed in the order of MASTER then SLAVE options.

MASTER JUMPERS:

1) P1 Connector - Interrupt Bus Connections

2) 4B and 4C Headers - Scanner Port Max Count Header

3) P2 Connector - Counter/Timer Options Wire Wrap Area

The P1 connector provides a path to the bus for the RECEIVE and TRANSMIT interrupts from the scanners. By selecting the appropriate pins the receive and transmit interrupts may be driven onto any of the vectored interrupt lines by open-collector devices. A common practice would be to allocate a higher priority interrupt for receive and a relatively low priority interrupt to the transmit. The following table gives the P1 pin descriptions

P1 CONNECTOR: (left to right)

pin1	RECV interrupt (open collector)
pin2	XMIT interrupt (open collector)
pin3	VI0 (vectored interrupt zero)
pin4	VI1
pin5	VI2
pin6	VI3
pin7	VI4
pin8	VI5
pin9	NMI (non-maskable interrupt)
pin10	(reserved for future use)

The dip headers located at chip positions 4B and 4C establish the scan limits. While it is not necessary to understand the schematic in order to program these units, it is rather important that they be wired correctly for a given configuration. Headers 4B and 4C determine the maximum number of USARTs in the current MSIO system. The system comes with the capacity for 16 terminals. To implement more than 16, consult APPENDIX F.

Finally the P2 connector is the last jumper area on the MASTER and perhaps the most interesting in its possibilities. The 8254 counter/timer is a versatile and powerful device containing three independent counters with a maximum clock rate of 10 MHZ. In addition the MSIO supports this device with inputs suitable for conditioning (or enabling) outputs of the 8254.

For example, one support circuit will enable interrupts only after the 8254 has been accessed through software given that a "bus slave clear" has occurred. Yet another will latch a momentary pulse and then hold an interrupt until the 8254 has again been accessed by software. These support circuits enhance the counting and timing features of the already rich family of methods offered by the 8254. Many other possibilities are available and need only a little examination to discover their utility. Very complex data gathering and timing applications may be devised using these resources. Furthermore, many operating systems require timing and "alarm-clock" services. These are easily accomplished using the wire-wrap area of the P2 to connect open-collector drivers to the interrupt bus lines.

P2 CONNECTOR: (counted left to right, then down)

pin1	2 MHZ from bus
pin2	enable strobe from support circuit ENAOUT0
pin3	enable input of support circuit ENAIN0
pin4	open-collector buffer INPUT0
pin5	open-collector buffer OUTPUT0
pin6	open-collector buffer INPUT1
pin7	open-collector buffer OUTPUT1
pin8	8254 CLK0 input
pin9	8254 GATE0 input
pin10	8254 OUT0 output
pin11	8254 CLK1 input
pin12	8254 GATE1 input
pin13	8254 OUT1 output
pin14	8254 CLK2 input
pin15	8254 GATE2 input
pin16	8254 OUT2 output
pin17	+5v through 1k pull-up
pin18	enable output of support circuit ENAOUT1
pin19	GROUND
pin20	enable input from support circuit ENAIN1

PARALLEL PORT INTERFACE LINES AND HEADER CONNECTIONS

There are three parallel ports on the I/O MASTER board controlled by software command as input, output, or bi-directional. Two of the ports, port A and port B, may be configured as a single 16 bit port (see the data sheet for details). Port A and port B are available through the P6 connector. Interleaved ground lines separate each signal line on all three ports; A, B, and C.

SIGNAL CONNECTIONS:

PORT A LINE	P6 HEADER LINE
PA7	25
PA6	27
PA5	29
PA4	31
PA3	33
PA2	35
PA1	37
PA0	39
HC1	23
HC2	21

PORT B LINE	P6 HEADER LINE
PB7	1
PB6	3
PB5	5
PB4	7
PB3	9
PB2	11
PB1	13
PB0	15
HC3	19
HC4	17

PORT C LINE	P7 HEADER LINE
PC7	1
PC6	3
PC5	5
PC4	7
PC3	9
PC2	11
PC1	13
PC0	15

5. RS-232C Interface

At the top of the SLAVE board are eight ten-pin headers which provide RS-232C interface in either DTE or DCE mode. In DTE mode the connector pins are wired to accept a modem or similar device. In DCE mode the connector provides appropriate signals for a terminal or similar device. There is an asymmetry regarding these two modes such that the DCD (data carrier detect) line is only driven when modems are in service. A pull-up has been provided to assert this line when left open in the event that a terminal is connected to the given port. The following table shows the correspondence of the two modes, DTE vs DCE, and the required cable implementation.

TEN PIN RS-232 CABLE INTERFACE

HEADER	DCE-DB25	DTE-DB25
1	8	8
2	7	7
3	(unused)	
4	(unused)	
5	20	6
6	2	3
7	4	5
8	3	2
9	6	20
10	5	4

As a general rule, terminals may be connected using a DCE cable (as defined above) and modems will require a DTE cable. In the event that the wrong cable is tried, no damage should result under normal circumstances as the RS-232C interface should withstand such errors.

6. Detailed Circuit Description

a. Board Address Decoding Logic

The MSIO devices operate in memory-mapped mode such that when accessed by the CPU they respond as if an ordinary memory device. The memory locations assigned to the MSIO are determined by the address selection switches located at chip positions 4D and 7D. The switch at 4D determines the address comparison for A23-A16. Similarly, the switch at 7D compares A15-A10. Address lines A9 and A8 need not be specified on switch 7D.

Once a match is made for address lines A23-A10, other logic selects the correct device to be referenced (parallel port, counter/timer, USART, interrupt register).

For systems running CPUs with bus clock rates above 4Mhz, a wait state may be necessary. The "wait-state" generator may be selected by cutting the trace at jumper location P3 and installing a wire from the center position to the lower header position. Units may be ordered from the factory with this option installed.

b. USART Operations

The USARTs are located on 8-byte boundaries starting at the location determined by the address selection switches 4D and 7D. There are four addressable ports in each USART: DATA, STATUS, MODE, and COMMAND. These are referenced by offsets of 0, 2, 4, and 6 from the base address of the particular USART. For example, to address the command port of the second USART, one must take the base address and then add the offset of 14 (that is 8 + 6). Suppose the MSIO system is located at $FEA200 hex. This address would be $FEA000 + 8 + 6 or $FEA00E hex.

Another important matter is the implementation of the interrupts. Header P1 permits the connection of receive or transmit interrupts to the S100 vectored interrupt lines. These are open-collector outputs and may be used in any suitable fashion. Often these are used as follows: assign a high priority interrupt for receive operations and a lower priority interrupt for transmit.

Chip positions 4B and 4C are dip headers which define the maximum number of USARTs to be used in the system. This reflects the number of "SLAVE" boards attached to the master. Examples are given in Appendix E for 1, 2, 4, and 8 SLAVEs for a total of up to 64 USARTs. This is a rather large number of terminal, but certain data communication applications warrant that many devices.

c. Receive and Transmit Interrupt Registers

These registers greatly enhance interrupt processing and allow the CPU to handle many pending requests during a single entry to the input/output drivers and handlers. The registers contain a bit which indicates a pending interrupt from a particular USART. The number of the USART needing service is given directly by the port. The driver/handler may immediately re-read the register after handling the first request to determine the next USART needing service. If no USART requires attention then the "active-interrupt" bit will be set to zero. The byte value when read from the interrupt register is a binary value in the range [0,63]. Bit 7 qualifies the value the port and indicates an active interrupt when set to one. So, for example, if the register shows a value of hex $09, then no interrupt is pending. But, for instance, a value of $83 indicates an active interrupt on the fourth port (port 0 is the first port).

d. 8254 Counter/Timer Operations

Addressing the 8254 follows essentially the same method as given for the USARTs. The ports of the 8254 are located at even addresses spaced by 2. The offset of the 8254 device is $210 hex (or 522 decimal) from the base address selected by switches 4D and 7D.

Many possible arrangements can be considered for the 8254 and four examples are given in Appendix D arising in real-time and general purpose computing. Others may be tried as all inputs and outputs of the 8254 are available on the P2 connector on the MASTER board. By using the open-collector devices available on P2, interrupts may be driven to the S100 bus for operating system timing or other such purposes.

e. 68230 Parallel Ports

Operations with the 68230 are decoded by the board select address comparators and then device select logic. Device select logic enables the 68230 when offset $240-$27E are referenced from the origin of the MSIO memory base as determined by switches 4B and 4C. The 68230 logic requires long-cycles for some setup operations and therefore a single wait-state may be observed when referencing the parallel port. This is done transparently without any special jumpering required. The 68230 generates a DTACK* when finished which in turn removes the "not READY" being asserted on pin 72 of the S100 bus.

APPENDIX A - 68230 PARALLEL PORT SPECIFICATIONS

APPENDIX B - 68661/2661 USART Specifications

APPENDIX C - 8254 Counter/Timer Specifications

APPENDIX D - Programming Examples

APPENDIX

PROGRAMMING EXAMPLES IN C

```
 /* set and put a character under program control
  * using port 0 of MSIO; initialized by call to initmsio.
  * note: type definitions are given as follows.
  *    UTINY is unsigned char.
  *    FAST is register.
  *    VOID is long, but implies that nothing is returned.
  */
#include <std.h>

typedef struct {
    UTINY u_data;
    UTINY u_null1;
    UTINY u_stat;
    UTINY u_null3;
    UTINY u_mode;
    UTINY u_null5;
    UTINY u_cmd;
    UTINY u_null7;
    } USART;

#define C_MODE1     0x4e
#define C_MODE2     0x3e
#define C_BASE      0x14
#define C_DTR       0x22
#define C_RIENA     0x04
#define C_TIENA     0x01
#define C_ERST      0x10

/*  status bit difinitions
 */
#define S_RRDY      0x02
#define S_TRDY      0x01
#define S_TEMT      0x04
#define S_CAR       0x80
#define S_E_FR      0x20
#define S_E_OV      0x10
#define S_E_PAR     0x08
#define S_ERR       (S_E_FR|S_E_OV|S_E_PAR)
```

```
VOID initmsio(port)
    int port;
    {
    FAST USART *pv = 0xfea000;  /* base address of MSIO */
    FAST UTINY c, stat;

    pv = pv + port; /* point to port of interest with pv */
    stat = pv->u_cmd; /* read and discard 2661 cmd register
                      /*   to start init process of 2661. */
    pv->u_mode = 0x4e; /* set mode register 1 */
    pv->u_mode = 0x3e; /* set mode register 2 for 9600 baud */
    pv->u_cmd  = 0x37; /* start recv and xmit operation */
    }

UTINY getch()
    {
    FAST USART *pv = 0xfea000;
    FAST UTINY c, stat;

    while (!((stat = pv->u_stat) & S_RRDY))
        ;
    c = pv->u_data & 0x7f;
    if (c == '\r')
        {
        putch('\n');
        return (c);
        }
    putch(c);
    return (c);
    }

/*  put a char
 */
VOID putch(c)
    UTINY c;
    {
    FAST USART *pv = 0xfea000;
    FAST UTINY stat;

    while (!((stat = pv->u_stat) & S_TRDY))
        ;
    pv->u_data = c;
    if (c == '\n')
        putch('\r');
    while (!((stat = pv->u_stat) & S_TRDY))
        ;
    }
```

```
VOID init8254()
    {
    int i,c;
    unsigned char *c8254ct0 = 0xfea210;  /* address of 8254 when base
                                            /* of MSIO is 0xfea000.    */
    unsigned char *c8254mod = 0xfea216;

    /* initialize 8254 counter/timer for MODE 2 operation
    /*   with MSIO support hardware to latch interrupts until
    /*   8254 is subsequently accessed by the CPU.
    */

    *c8254mod = 0x34; /* set for 8254 MODE 2 */
    *c8254ct0 = 0xa8; /* lsb of count */
    *c8254ct0 = 0x61; /* msb of 16 bit count */
                      /* input clock is 2 MHZ; count is 25000 */
                      /*   resulting in 80 HZ interrupt rate. */
    }
```

APPENDIX E - Hardware Support Options for 8254, and Sample Configurations

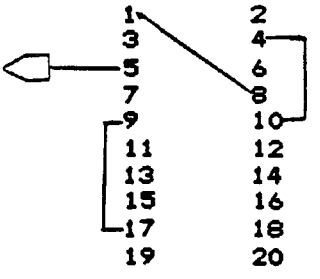
APPENDIX

8254 HARDWARE OPTIONS AND EXAMPLES

This section gives a number of common applications of the 8254 and associated MSIO support circuitry. Figure 1 displays the pin configuration of the P2 connector and the schematic of the support circuits. Interrupts may be driven to the bus by a wire-wrap (or jumper) between P2 and the appropriate pin on P1. In each example below, we will give a connection pattern for jumpers on P2 and describe the resulting output from the 8254 or selected MSIO support circuit.

1. Mode 0. Interrupt driven to S-100 Buss

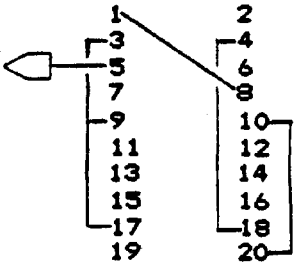
P2 Connector



(enable GATE0, 2MHZ -> CLK0,
OUT0 -> INPUT0,
OUTPUT0 -> S-100 interrupt)

2. Mode 0. Interrupt gated by 8254sthx then driven to bus.

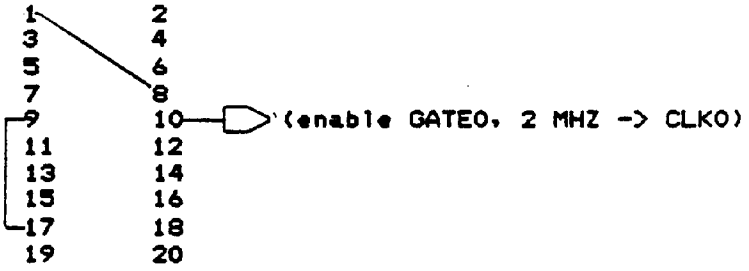
P2 Connector



(enable GATE0, 2 MHZ -> CLK0,
VCC -> ENAIN0,
OUT0 -> ENAIN1,
ENAOUT1 -> INPUT0,
OUTPUT0 -> S-100 interrupt)

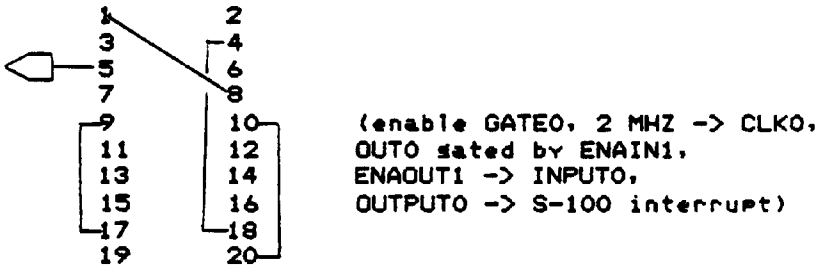
3. Mode 2. Real time clock output from P2 pin 10.

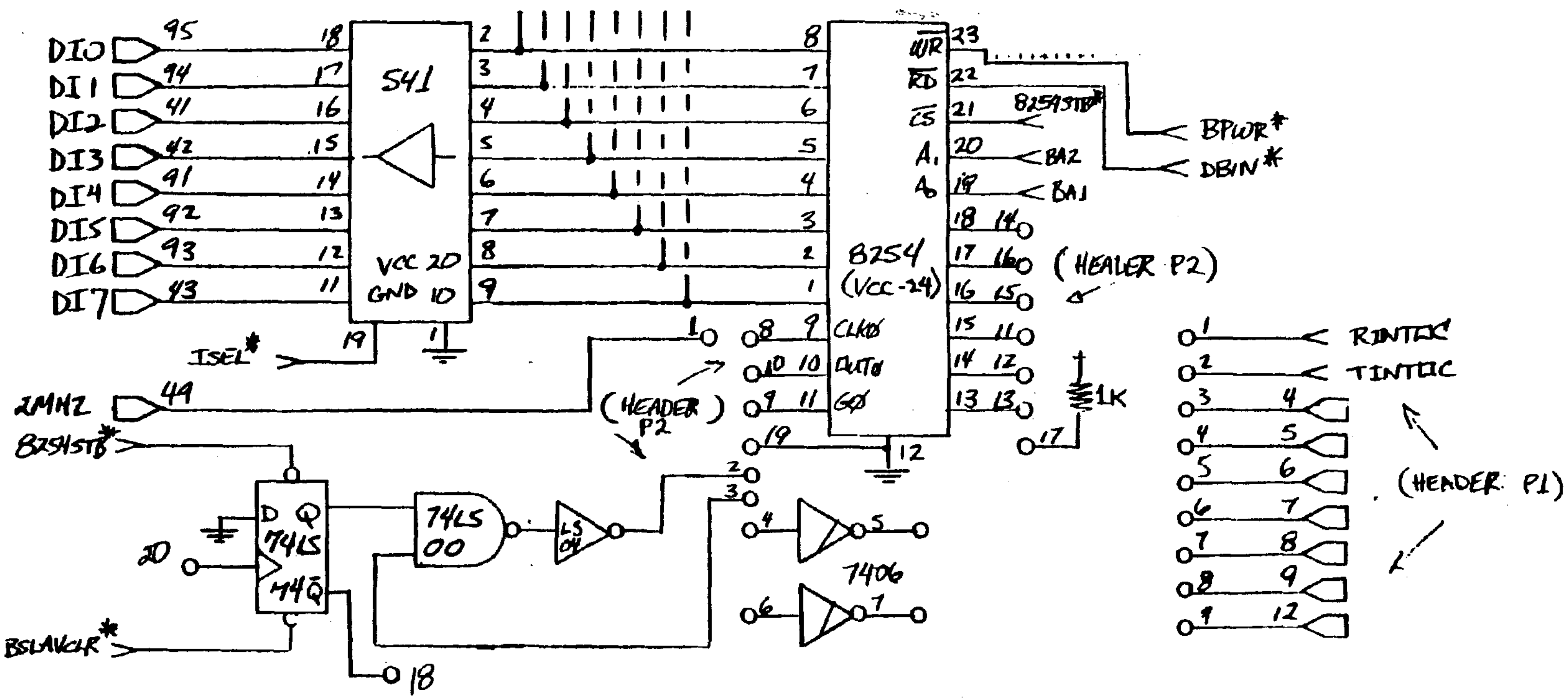
P2 Connector



4. Mode 2. Interrupt latched/ software control.

P2 Connector





DI0
DI1
DI2
DI3
DI4
DI5
DI6
DI7
541
VCC 20
GND 10
ISEL*
2MHZ
8254
(VCC-24)
WR
RD
CS
A1
A0
CLK0
OUT0
G0
BPWR*
DBIN*
BA2
BA1
8254STB*
(HEADER P2)
1K
74LS 74
74LS 00
LS 04
7406
BSLAVCLR*
RINTCK
TINTCK
(HEADER P1)

APPENDIX F - Master Board Scanner Limit Headers 4B and 4C

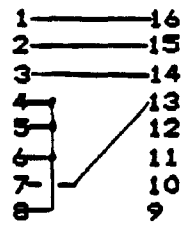
APPENDIX

MASTER BOARD SCAN LIMIT HEADERS 4B AND 4C

This section gives the wiring patterns for the headers of the scanners. These headers determine the maximum number of SLAVE boards associated with a MASTER controlling board. The headers specify one of 4 possible configurations: 1, 2, 4, or 8 SLAVE boards.

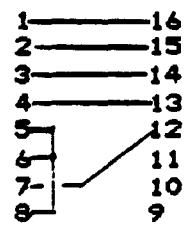
1. 1 SLAVE BOARD

HEADER 4B and 4C



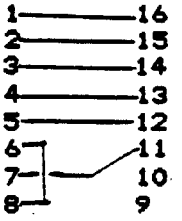
2. 2 SLAVE BOARDS

HEADER 4B and 4C



3. 4 SLAVE BOARDS

HEADER 4B and 4C



4. 8 SLAVE BOARDS

HEADER 4B and 4C