Hardware

COLLABORATORS			
	TITLE : Hardware		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		March 14, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Hardware

1.1 Amiga® Hardware Reference Manual: F 8520 Complex Interface Adapters

This appendix contains information about the 8520 Complex Interface Adapter (CIA) chips which handle the serial, parallel, keyboard and other Amiga I/O activities. Each Amiga system contains two 8520 Complex Interface Adapter (CIA) chips. Each chip has 16 general purpose input/output pins, plus a serial shift register, three timers, an output pulse pin and an edge detection input. In the Amiga system various tasks are assigned to the chip's capabilities as follows:

CIAA Address Map

Byte Address	Register Name	Data bits 7	6	5	4	3	2	1	0
BFE001	pra	/FIR1	/FIR0	/RDY	/TK0	/WPRO	/CHNG	/LED	OVL
BFE101	prb	Parallel port							
BFE201	ddra	Direction for port A (BFE001);1=output (set to 0x03)							
BFE301	ddrb	Direction for port B (BFE101);1=output (can be in or out)							
BFE401	talo	CIAA timer A low byte (.715909 Mhz NTSC; .709379 Mhz PAL)							
BFE501	tahi	CIAA timer A high byte							
BFE601	tblo	CIAA timer B low byte (.715909 Mhz NTSC; .709379 Mhz PAL)							
BFE701	tbhi	CIAA timer B high byte							
BFE801	todlo	50/60 Hz event counter bits 7-0 (VSync or line tick)							
BFE901	todmid	50/60 Hz event counter bits 15-8							
BFEA01	todhi	50/60 Hz event counter bits 23-16							
BFEB01		not used							
BFEC01	sdr	CIAA serial data register (connected to keyboard)							
BFED01	icr	CIAA interrupt control register							
BFEE01	cra	CIAA control register A							
BFEF01	crb	CIAA control register B							

Note: CIAA can generate interrupt INT2.

CIAB Address Map

Byte Address	Register Name	Data bits 7	6	5	4	3	2	1	0

```
------------------------------------------------------------------------
BFD000    pra     /DTR  /RTS  /CD   /CTS  /DSR   SEL   POUT  BUSY
BFD100    prb     /MTR  /SEL3 /SEL2 /SEL1 /SEL0 /SIDE  DIR  /STEP
BFD200    ddra    Direction for Port A (BFD000);1 = output (set to 0xFF)
BFD300    ddrb    Direction for Port B (BFD100);1 = output (set to 0xFF)
BFD400    talo    CIAB timer A low byte (.715909 Mhz NTSC; .709379 Mhz PAL)
BFD500    tahi    CIAB timer A high byte
BFD600    tblo    CIAB timer B low byte (.715909 Mhz NTSC; .709379 Mhz PAL)
BFD700    tbhi    CIAB timer B high byte
BFD800    todlo   Horizontal sync event counter bits 7-0
BFD900    todmid  Horizontal sync event counter bits 15-8
BFDA00    todhi   Horizontal sync event counter bits 23-16
BFDB00            not used
BFDC00    sdr     CIAB serial data register (unused)
BFDD00    icr     CIAB interrupt control register
BFDE00    cra     CIAB Control register A
BFDF00    crb     CIAB Control register B

Note:  CIAB can generate INT6.


@{ " Chip Register Map " link F-1}                  @{ " Interrupt Control  ↩
   Register (ICR) " link F-5}
@{ " Register Functional Description " link F-2}    @{ " Control Registers " link  ↩
   F-6}
@{ " Time of Day Clock " link F-3}                  @{ " Port Signal Assignments " ↩
    link F-7}
@{ " Serial Shift Register (SDR) " link F-4}        @{ " Hardware Connection  ↩
   Details " link F-8}
```

1.2 F 8520 Complex Interface Adapters / Chip Register Map

```
                Each 8520 has 16 registers that you may read or write.  Here is  ↩
                   the list
of registers and the access address of each within the memory space
dedicated to the 8520:

                      Register
   RS3  RS2  RS1  RS0  #(hex)  NAME      MEANING
   ------------------------------------------------------------
    0    0    0    0     0     pra
                Peripheral data register A
                   0    0    0    1     1     prb
                Peripheral data register B
                   0    0    1    0     2     ddra      Data
                direction register A
                   0    0    1    1     3     ddrb
                Direction register B
                   0    1    0    0     4     talo
                Timer A
                low register
    0    1    0    1     5     tahi
                Timer A
                high register
    0    1    1    0     6     tblo
```

```
                Timer B
                low register
    0    1    1    1     7     tbhi
                Timer B
                high register
    1    0    0    0     8     todlow
                Event LSB
                   1    0    0    1     9     todmid
                Event 8-15
                   1    0    1    0     A     todhi
                Event MSB
                   1    0    1    1     B              No connect
    1    1    0    0     C     sdr
                Serial data register
                   1    1    0    1     D     icr
                Interrupt control register
                   1    1    1    0     E     cra
                Control register A
                   1    1    1    1     F     crb
                Control register B
                  ------------------------------------------------------------- ↩
```

1.3 F 8520 Complex Interface Adapters / Register Functional Description

```
@{ " I/O Ports (PRA, PRB, DDRA, DDRB) " link F-2-1}
@{ " Handshaking " link F-2-2}
@{ " Interval Timers (Timer A, Timer B) " link F-2-3}
@{ " Input Modes " link F-2-4}
@{ " Bit Names on Read-Register " link F-2-5}
@{ " Bit Names on Write-Register " link F-2-6}
```

1.4 F / Register Functional Description / I/O Ports (PRA, PRB, DDRA, DDRB)

```
                Ports A and B each consist of an 8-bit peripheral data register ( ↩
                   PR) and
an 8-bit data direction register (DDR).  If a bit in the DDR is set to a
1, the corresponding bit position in the PR becomes an output.  If a DDR
bit is set to a 0, the corresponding PR bit is defined as an input.

When you READ a PR register, you read the actual current state of the I/O
pins (PA0-PA7, PB0-PB7, regardless of whether you have set them to be
inputs or outputs.

Ports A and B have passive pull-up devices as well as active pull-ups,
providing both CMOS and TTL compatibility.  Both ports have two TTL load
drive capability.

In addition to their normal I/O operations, ports PB6 and PB7 also provide

                timer
```

output functions.

1.5 F / Register Functional Description / Handshaking

Handshaking occurs on data transfers using the PC output pin and the FLAG input pin. PC will go low on the third cycle after a port B access. This signal can be used to indicate "data ready" at port B or "data accepted" from port B. Handshaking on 16-bit data transfers (using both ports A and B) is possible by always reading or writing port A first. FLAG is a negative edge-sensitive input that can be used for receiving the PC output from another 8520 or as a general- purpose interrupt input. Any negative transition on FLAG will set the FLAG interrupt bit.

REG	NAME	D7	D6	D5	D4	D3	D2	D1	D0
0	PRA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
1	PRB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
2	DDRA	DPA7	DPA6	DPA5	DPA4	DPA3	DPA2	DPA1	DPA0
3	DDRB	DPB7	DPB6	DPB5	DPB4	DPB3	DPB2	DPB1	DPB0

1.6 F / Register Functional Description / Interval Timers (Timer A, B)

Each interval timer consists of a 16-bit read-only timer counter and a 16-bit write-only timer latch. Data written to the timer is latched into the timer latch, while data read from the timer is the present contents of the timer counter.

The latch is also called a prescalar in that it represents the countdown value which must be counted before the timer reaches an underflow (no more counts) condition. This latch (prescalar) value is a divider of the input clocking frequency. The timers can be used independently or linked for extended operations. Various timer operating modes allow generation of long time delays, variable width pulses, pulse trains, and variable frequency waveforms. Utilizing the CNT input, the timers can count external pulses or measure frequency, pulse width, and delay times of external signals.

Each timer has an associated

control register
, providing independent
control over each of the following functions:

@{ " Start/Stop " link F-2-3-1}
@{ " PB on/off " link F-2-3-2}
@{ " Toggle/pulse " link F-2-3-3}
@{ " One-shot/continuous " link F-2-3-4}
@{ " Force load " link F-2-3-5}

1.7 F / / Interval Timers (Timer A, Timer B) / Start/Stop

A control bit (CRx0) allows the timer to be started or stopped by the microprocessor at any time.

1.8 F / / Interval Timers (Timer A, Timer B) / PB on/off

A control bit (CRx1) allows the timer output to appear on a
port B
output line (PB6 for timer A and PB7 for timer B). This function overrides the
DDRB
control bit and forces the appropriate PB line to become an output.

1.9 F / / Interval Timers (Timer A, Timer B) / Toggle/pulse

A control bit (CRx2) selects the output applied to
port B
while the PB
on/off bit is ON. On every timer underflow, the output can either toggle or generate a single positive pulse of one cycle duration. The toggle output is set high whenever the timer is started, and set low by
RES
.

1.10 F / / Interval Timers (Timer A, Timer B) / One-shot/continuous

A control bit (CRx3) selects either timer mode. In one-shot mode, the timer will count down from the latched value to zero, generate an interrupt, reload the latched value, then stop. In continuous mode, the timer will count down from the latched value to zero, generate an interrupt, reload the latched value, and repeat the procedure continuously.

In one-shot mode, a write to timer-high (register 5 for timer A, register 7 for Timer B) will transfer the timer latch to the counter and initiate counting regardless of the start bit.

1.11 F / / Interval Timers (Timer A, Timer B) / Force load

A strobe bit (CRx4) allows the timer latch to be loaded into the timer
counter at any time, whether the timer is running or not.

1.12 F / Register Functional Description / Input Modes

Control bits (CRA5, CRB5-6) allow selection of the clock used to ↩
decrement
the timer.
Timer A
can count 02 clock pulses or external pulses applied
to the CNT pin.
Timer B
can count 02 pulses, external CNT pulses,

timer A
underflow pulses, or
timer A
underflow pulses while the CNT
pin is held high.

The timer latch is loaded into the timer on any timer underflow, on a
force load, or following a write to the high byte of the pre- scalar while
the timer is stopped. If the timer is running, a write to the high byte
will load the timer latch but not the counter.

1.13 F / Register Functional Description / Bit Names on Read-Register

REG	NAME	D7	D6	D5	D4	D3	D2	D1	D0
4	TALO	TAL7	TAL6	TAL5	TAL4	TAL3	TAL2	TAL1	TAL0
5	TAHI	TAH7	TAH6	TAH5	TAH4	TAH3	TAH2	TAH1	TAH0
6	TBLO	TBL7	TBL6	TBL5	TBL4	TBL3	TBL2	TBL1	TBL0
7	TBHI	TBH7	TBH6	TBH5	TBH4	TBH3	TBH2	TBH1	TBH0

1.14 F / Register Functional Description / Bit Names on Write-Register

REG	NAME	D7	D6	D5	D4	D3	D2	D1	D0
4	TALO	PAL7	PAL6	PAL5	PAL4	PAL3	PAL2	PAL1	PAL0
5	TAHI	PAH7	PAH6	PAH5	PAH4	PAH3	PAH2	PAH1	PAH0
6	TBLO	PBL7	PBL6	PBL5	PBL4	PBL3	PBL2	PBL1	PBL0
7	TBHI	PBH7	PBH6	PBH5	PBH4	PBH3	PBH2	PBH1	PBH0

1.15 F 8520 Complex Interface Adapters / Time of Day Clock

TOD consists of a 24-bit binary counter. Positive edge ↩ transitions on
this pin cause the binary counter to increment. The TOD pin has a passive pull-up on it.

A programmable alarm is provided for generating an interrupt at a desired time. The alarm registers are located at the same addresses as the corresponding TOD registers. Access to the alarm is governed by a

control register
bit (CRB7). The alarm is write-only; any read of a TOD address will read time regardless of the state of the ALARM access bit.

A specific sequence of events must be followed for proper setting and reading of TOD. TOD is automatically stopped whenever a write to the register occurs. The clock will not start again until after a write to the
LSB event
register. This assures that TOD will always start at the desired time.

Since a carry from one stage to the next can occur at any time with respect to a read operation, a latching function is included to keep all TOD information constant during a read sequence. All TOD registers latch on a read of
MSB event
and remain latched until after a read of

LSB event
. The TOD clock continues to count when the output registers are latched. If only one register is to be read, there is no carry problem and the register can be read "on the fly" provided that any read of

MSB event
is followed by a read of
LSB Event
to disable the latching.

@{ " Bit Names for Write Time/Alarm or Read Time " link F-3-1}

1.16 F / Time of Day Clock / Bit Names for Write Time/Alarm or Read Time

	REG NAME
8	LSB Event
	E7 E6 E5 E4 E3 E2 E1 E0
9	Event 8-15
	E15 E14 E13 E12 E11 E10 E9 E8
A	

```
                MSB Event
                  E23  E22  E21  E20  E19  E18  E17  E16

        WRITE
        CRB7 = 0
        CRB7 = 1 ALARM
```

1.17 F 8520 Complex Interface Adapters / Serial Shift Register (SDR)

The serial port is a buffered, 8-bit synchronous shift register. A control bit (CRA6) selects input or output mode. In the Amiga system one shift register is used for the keyboard, and the other is unassigned. Note that the RS-232 compatible serial port is controlled by the Paula chip ; see chapter 8 for details.

@{ " Input Mode " link F-4-1}
@{ " Output Mode " link F-4-2}
@{ " Bidirectional Feature " link F-4-3}

1.18 F / Serial Shift Register (SDR) / Input Mode

In input mode, data on the SP pin is shifted into the shift register on the rising edge of the signal applied to the CNT pin. After eight CNT pulses, the data in the shift register is dumped into the serial data register and an interrupt is generated.

1.19 F / Serial Shift Register (SDR) / Output Mode

In the output mode,
Timer A
is used as the baud rate generator. Data is
shifted out on the SP pin at 1/2 the underflow rate of
Timer A
. The
maximum baud rate possible is 02 divided by 4, but the maximum usable baud rate will be determined by line loading and the speed at which the receiver responds to input data.

To begin transmission, you must first set up
Timer A
in continuous mode,
and start the timer. Transmission will start following a write to the serial data register. The clock signal derived from
Timer A
appears as
an output on the CNT pin. The data in the serial data register will be loaded into the shift register, then shifted out to the SP pin when a CNT pulse occurs. Data shifted out becomes valid on the next falling edge of CNT and remains valid until the next falling edge.

After eight CNT pulses, an interrupt is generated to indicate that more data can be sent. If the serial data register was reloaded with new information prior to this interrupt, the new data will automatically be loaded into the shift register and transmission will continue.

If no further data is to be transmitted after the eighth CNT pulse, CNT will return high and SP will remain at the level of the last data bit transmitted.

SDR data is shifted out MSB first. Serial input data should appear in this same format.

1.20 F / Serial Shift Register (SDR) / Bidirectional Feature

The bidirectional capability of the shift register and CNT clock allows many 8520s to be connected to a common serial communications bus on which one 8520 acts as a master, sourcing data and shift clock, while all other 8520 chips act as slaves. Both CNT and SP outputs are open drain to allow such a common bus. Protocol for master/slave selection can be transmitted over the serial bus or via dedicated handshake lines.

REG	NAME	D7	D6	D5	D4	D3	D2	D1	D0
C	SDR	S7	S6	S5	S4	S3	S2	S1	S0

1.21 F 8520 Complex Interface Adapters / Interrupt Control Register (ICR)

There are five sources of interrupts on the 8520:

- Underflow from Timer A (timer counts down past 0)
- Underflow from Timer B
- TOD alarm
- Serial port full/empty
- Flag

A single register provides masking and interrupt information. The interrupt control register consists of a write-only MASK register and a read-only DATA register. Any interrupt will set the corresponding bit in the DATA register. Any interrupt that is enabled by a 1-bit in that position in the MASK will set the IR bit (MSB) of the DATA register and bring the IRQ pin low. In a multichip system, the IR bit can be polled to detect which chip has generated an interrupt request.

When you read the DATA register, its contents are cleared (set to 0), and the
IRQ
line returns to a high state. Since it is cleared on a read, you must assure that your interrupt polling or interrupt service code can preserve and respond to all bits which may have been set in the DATA register at the time it was read. With proper preservation and response, it is easily possible to intermix polled and direct interrupt service methods.

You can set or clear one or more bits of the MASK register without affecting the current state of any of the other bits in the register. This is done by setting the appropriate state of the MSBit, which is called the set/clear bit. In bits 6-0, you yourself form a mask that specifies which of the bits you wish to affect. Then, using bit 7, you specify HOW the bits in corresponding positions in the mask are to be affected.

* If bit 7 is a 1, then any bit 6-0 in your own mask byte which is set to a 1 sets the corresponding bit in the MASK register. Any bit that you have set to a 0 causes the MASK register bit to remain in its current state.

* If bit 7 is a 0, then any bit 6-0 in your own mask byte which is set to a 1 clears the corresponding bit in the MASK register. Again, any 0 bit in your own mask byte causes no change in the contents of the corresponding MASK register bit.

If an interrupt is to occur based on a particular condition, then that corresponding MASK bit must be a 1.

Example: Suppose you want to set the
Timer A
interrupt bit (enable the

Timer A
interrupt), but want to be sure that all other interrupts are cleared. Here is the sequence you can use:

```
        INCLUDE "hardware/cia.i"
        XREF    _ciaa                   ; From amiga.lib
        lea     _ciaa,a0                ; Defined in amiga.lib
        move.b  #%01111110,ciaicr(a0)
```

MSB is 0, means clear any bit whose value is 1 in the rest of the byte

```
        INCLUDE "hardware/cia.i"
        XREF    _ciaa                   ; From amiga.lib
        lea     _ciaa,a0                ; Defined in amiga.lib
        move.b  #%10000001,ciaicr(a0)
```

MSB is 1, means set any bit whose value is 1 in the rest of the byte (do not change any values wherein the written value bit is a zero)

@{ " Read Interrupt Control Register " link F-5-1}
@{ " Write Interrupt Control Mask " link F-5-2}

1.22 F / Interrupt Control Register (ICR) / Read Interrupt Control Register

REG	NAME	D7	D6	D5	D4	D3	D2	D1	D0
D	ICR	IR	0	0	FLG	SP	ALRM	TB	TA

1.23 F / Interrupt Control Register (ICR) / Write Interrupt Control Mask

REG	NAME	D7	D6	D5	D4	D3	D2	D1	D0
D	ICR	S/C	x	x	FLG	SP	ALRM	TB	TA

1.24 F 8520 Complex Interface Adapters / Control Registers

There are two control registers in the 8520, CRA and CRB. CRA is associated with
Timer A
and CRB is associated with
Timer B
. The
format of the registers is as follows:

@{ " Control Register A " link F-6-1}
@{ " Bitmap of Register CRA " link F-6-2}
@{ " Control Register B " link F-6-4}
@{ " Bitmap of Register CRB " link F-6-3}

1.25 F / Control Registers / Control Register A

BIT	NAME	FUNCTION
0	START	1 = start Timer A, 0 = stop Timer A. This bit is automatically reset (= 0) when underflow occurs during one-shot mode.
1	PBON	1 = Timer A output on PB6, 0 = PB6 is normal operation.
2	OUTMODE	1 = toggle, 0 = pulse.
3	RUNMODE	1 = one-shot mode, 0 = continuous mode.
4	LOAD	1 = force load (this is a strobe input, there is no

```
                   data storage;  bit 4 will always read back a zero
                   and writing a 0 has no effect.)
   5
                 INMODE
                  1 = Timer A counts positive CNT transitions,
                0 = Timer A counts 02 pulses.
   6
                 SPMODE
                  1 = Serial port=output (CNT is the source of the shift
                   clock)
                0 = Serial port=input  (external shift clock is
                   required)
   7   UNUSED
```

1.26 F / Control Registers / Bitmap of Register CRA

```
                REG# NAME UNUSED  SPMODE  INMODE  LOAD   RUNMODE  OUTMODE   PBON  ↩
                      START
---- ---- ------  ------  ------  ----   -------  -------   ----    -----
 E   CRA  unused  0=input  0=02  1=force  0=cont. 0=pulse  0=PB6OFF 0=stop
          unused  1=output 1=CNT   load   1=one-  1=toggle 1=PB6ON  1=start
                                 (strobe)   shot

                          |<--------
                 Timer
                 A Variables --------------->|
```

All unused register bits are unaffected by a write and forced to 0 on a read.

1.27 F / Control Registers / Control Register B

```
                   CONTROL REGISTER B:

  BIT  NAME     FUNCTION
  ---  ----     --------
   0
                 START
                   1 = start Timer B, 0 = stop Timer B.
                    This bit is automatically reset (= 0) when
                    underflow occurs during one-shot mode.
   1
                 PBON
                    1 = Timer B output on PB7, 0 = PB7 is normal
                    operation.
   2
                 OUTMODE
                 1 = toggle, 0 = pulse.
   3
                 RUNMODE
```

```
                1 = one-shot mode, 0 = continuous mode.
   4
                LOAD
                   1 = force load (this is a strobe input, there is no
                   data storage;  bit 4 will always read back a
                   zero and writing a 0 has no effect.)
  6,5
                INMODE
                 Bits CRB6 and CRB5 select one of four possible
               input modes for Timer B, as follows:

               CRB6  CRB5   Mode Selected
               ----  ----   ---------------------------------------
                0     0     Timer B counts 02 pulses
                0     1     Timer B counts positive CNT transitions
                1     0     Timer B counts Timer A underflow pulses
                1     1     Timer B counts Timer A underflow pulses
                              while CNT pin is held high.

   7
                ALARM
                   1 = writing to TOD registers sets Alarm
                0 = writing to TOD registers sets TOD clock.
                    Reading TOD registers always reads TOD clock,
                    regardless of the state of the Alarm bit.
```

1.28 F / Control Registers / Bitmap Of Register CRB

```
                REG# NAME ALARM    INMODE     LOAD      RUNMODE  OUTMODE  PBON     ↩
                   START
---- ---- -----    ------     ----      -------  -------  ----    -----
 F   CRB  0=TOD    00=02      1=force   0=cont.  0=pulse  0=PB7OFF 0=stop
          1=Alarm  01=CNT       load    1=one-   1=toggle 1=PB7ON  1=start
                   10=Timer A  (strobe)   shot
                   11=CNT+
                     Timer A

                   |<---------------
                Timer B
                Variables ---------->|
```

All unused register bits are unaffected by a write and forced to 0 on a read.

1.29 F 8520 Complex Interface Adapters / Port Signal Assignments

This part specifies how various signals relate to the available ports of the 8520. This information enables the programmer to relate the port addresses to the outside-world items (or internal control signals) which are to be affected. This part is primarily for the use of the systems

programmer and should generally not be used by applications programmers.
Systems software normally is configured to handle the setting of
particular signals, no matter how the physical connections may change.

> Warning:
> --------
> In a multitasking operating system, many different tasks may be
> competing for the use of the system resources. Applications
> programmers should follow the established rules for resource access
> in order to assure compatibility of their software with the system.

```
CIA-A  Address BFEr01  data bits 7-0  (A12*) (INT2)

PA7..game port 1, pin 6 (fire button*)
PA6..game port 0, pin 6 (fire button*)
PA5.. RDY*      disk ready*
PA4.. TK0*      disk track 00*
PA3.. WPRO*     write protect*
PA2.. CHNG*     disk change*
PA1..LED*       led light (0=bright)
PA0..OVL        memory overlay bit
SP... KDAT      keyboard data
CNT.. KCLK      keyboard clock
PB7..P7         data 7
PB6..P6         data 6
PB5..P5         data 5     Centronics parallel interface
PB4..P4         data 4          data
PB3..P3         data 3
PB2..P2         data 2
PB1..P1         data 1
PB0..P0         data 0
PC... drdy*                Centronics control
F.... ack*


CIA-B  Address BFDr00  data bits 15-8   (A13*) (INT6)

PA7..com line  DTR* , driven output
PA6..com line  RTS* , driven output
PA5..com line carrier detect*
PA4..com line  CTS*
PA3..com line  DSR*
PA2.. SEL       centronics control
PA1.. POUT      paper out ---+
PA0.. BUSY      busy    ---+ |
                           | |
SP... BUSY      commodore -+ |
CNT.. POUT      commodore ---+

PB7.. MTR*      motor
PB6.. SEL3*     select external 3rd drive
PB5.. SEL2*     select external 2nd drive
PB4.. SEL1*     select external 1st drive
PB3.. SEL0*     select internal drive
PB2.. SIDE*     side select*
PB1.. DIR       direction
PB0.. STEP*     step*   (3.0 milliseconds minimum)
```

```
PC...not used
F.... INDEX*    disk index*

     8520_timing.asm
```

1.30 F 8520 Complex Interface Adapters / Hardware Connection Details

The system hardware selects the CIAs when the upper three address bits are 101. Furthermore, CIAA is selected when A12 is low, A13 high; CIAB is selected when A12 is high, A13 low. CIAA communicates on data bits 7-0, CIAB communicates on data bits 15-8.

Address bits A11, A10, A9, and A8 are used to specify which of the 16 internal registers you want to access. This is indicated by "r" in the address. All other bits are don't cares. So, CIAA is selected by the following binary address: 101x xxxx xx01 rrrr xxxx xxx0. CIAB address: 101x xxxx xx10 rrrr xxxx xxx1

With future expansion in mind, we have decided on the following addresses: CIAA = BFEr01; CIAB = BFDr00. Software must use byte accesses to these address, and no other.

@{ " Interface Signals " link F-8-1}

1.31 F / Hardware Connection Details / Interface Signals

@{ " Clock input " link F-8-1-1}
@{ " CS - chip-select input " link F-8-1-2}
@{ " R/W - read/write input " link F-8-1-3}
@{ " RS3-RS0 - address inputs " link F-8-1-4}
@{ " DB7-DB0 - data bus inputs/ outputs " link F-8-1-5}
@{ " IRQ - interrupt request output " link F-8-1-6}
@{ " RES - reset input " link F-8-1-7}

1.32 F / / Interface Signals / Clock input

The 02 clock is a TTL compatible input used for internal device operation and as a timing reference for communicating with the system data bus. On the Amiga, this is connected to the 680x0 "E" clock. The "E" clock runs at 1/10 of the CPU clock. This works out to .715909 Mhz for NTSC or .709379 Mhz for PAL.

1.33 F / / Interface Signals / CS - chip-select input

The CS input controls the activity of the 8520. A low level on CS ↩ while
02 is high causes the device to respond to signals on the
R/W
and address
(
RS
) lines. A high on CS prevents these lines from controlling the ↩ 8520.
The CS line is normally activated (low) at 02 by the appropriate address combination.

1.34 F / / Interface Signals / R/W - read/write input

The R/W signal is normally supplied by the microprocessor and controls the direction of data transfers of the 8520. A high on R/W indicates a read (data transfer out of the 8520), while a low indicates a write (data transfer into the 8520).

1.35 F / / Interface Signals / RS3-RS0 - address inputs

The address inputs select the internal registers as described by ↩ the

register map

1.36 F / / Interface Signals / DB7-DB0 - data bus inputs/outputs

The eight data bus output pins transfer information between the ↩ 8520 and
the system data bus. These pins are high impedance inputs unless
CS
is
low and
R/W
and 02 are high, to read the device. During this read, the data bus output buffers are enabled, driving the data from the selected register onto the system data bus.

1.37 F / / Interface Signals / IRQ - interrupt request output

IRQ is an open drain output normally connected to the processor interrupt input. An external pull-up resistor holds the signal high, allowing multiple IRQ outputs to be connected together. The IRQ output is normally off (high impedance) and is activated low as indicated in the functional description.

1.38 F / / Interface Signals / RES - reset input

A low on the RES pin resets all internal registers. The port pins are set as inputs and port registers to zero (although a read of the ports will return all highs because of passive pull-ups). The timer control registers are set to zero and the timer latches to all ones. All other registers are reset to zero.