# AUSTRALIA TELESCOPE NATIONAL FACILITY PC DISTRIBUTED CLOCK

## HARDWARE DESCRIPTION

29 June 2000.

**CONTENTS**

1.0 General.
2.0 Specifications.
3.0 Register descriptions.
4.0 Programming.
5.0 Programming Examples.
6.0 Engineering Modifications.

## 1.0 General.

The PC based Australia Telescope Distributed Clock (ATDC) MK-VI is the next generation Distributed Clock based on the earlier Jeffrey Crapps[1] stand alone version, the Andrew Hunt[2] Q-Bus version (MK-IV) and the University of Tasmania Physics Department derivative. Key features include:

* Choice of BAT data sourced from internal counters, or from host computer.
* Reduced power consumption.
* Choice of either 10MHz or 5MHz reference frequency input.
* 1mS and 1 Second tick and self interrupt.
* Single AT ISA bus standard supporting both 8 bit and 16 bit transfers.
* ISA bus transfer time typically around 250nS.
* 16 words of on-board configuration memory to allow storage of key parameters such as DUTC, DUT1, Longitude, etc.
* Available in standard 10MHz internal time base (50nS Tick phase resolution) or 50MHz internal time base (10nS Tick phase resolution).
* Reduced cost. Only 1 Xilinx® chip, a VCXO, a few other low cost components and a compact, 2 layer board required.
* Compatible with any IBM® compatible AT PC.

When used with the ATNF pSOS® control program, the clock can be used for the central time keeper for the ATNF.

[1] - Retired, Australia Telescope National Facility, Marsfield, NSW
[2] - Australia Telescope National Facility, Parkes, NSW

# 2.0 Specifications:

## 2.1 General.

The PC AT Distributed Clock transmits a 1mS time frame over balanced TTL lines. The data in the frame is made up of 55 X 16 bit words. Table 2.1 shows the data contents of a frame. Except for the first 4 X 16 bit words, all data is sent from an internal buffer loaded by the host PC during the previous 1mS frame. The first 4 X 16 bit words may be sourced from either internal counters or from the buffer memory.

Each frame is delimited by a sync signal. For a 1mS frame, this consists of 31 “1” data bits followed by a single “0” bit, and a 1 second frame, 47 X “1” data bits and a “0” data bit.

Data is transmitted on a pulse width modulated 1MHz carrier. The rising edge of the pulse is the clock information for all accessories that connect to this signal, while the pulse length indicates a logic 0 or 1.

The clock’s time base is from an internal voltage controlled crystal oscillator. This may be locked to an external 5 MHz or 10MHz reference. An External 1 second tick may also be supplied from a time source such as a GPS receiver to gauge the phase relationship of the ATDC with a external reference.

The clock’s internal time may be tweaked to an accuracy of ±100nS.

The AT Distributed Clock requires a full 16 bit slot on the ISA Bus, and has a length not much longer than the ISA connector.

The Clock Bus (RS-422) is distributed over 100Ω (nominal) twisted pair cable and requires a termination of this amount between the 2 signal lines. Keep common mode voltages to less than ±5V. A ground line is provided on the RS-422 connectors to help with this. Beware of very large voltage transients as a result of lightning strikes.

**The Clock hardware (Xilinx) can only be reset by either power cycling the host PC, or by the Clock Reset switch (a remote switch can be fitted).** The Clock Reset switch is located just above the board’s indicator lights. It therefore follows that the PC’s hardware may be reset or (provided the Clock’s software is non-intrusive at startup) the software rebooted at any time, without affecting the Clock’s fundemental counters. In addition, if the Clock is configured to transmit the BAT directly from the internal BAT counters, then BAT will still be active on the Clock Bus.

| **Word** | **Description** | **MSB ----- LSB** |
|---|---|---|
| 0 | International Atomic Time (IAT) as an | BAT-uS [15:0] |
| 1 | integer count in uS - (BATuS). | BAT-uS [31:16] |
| 2 | | BAT-uS [47:32] |
| 3 | | BAT-uS [63:48] |
| 4 | Universal Coordinated Time (UTC-mS) - | UTC-mS [15:0] |
| 5 | updated every 250mS. | UTC-mS [31:16] |
| 6 | Local Mean Sidereal Time (LMST-mS) - | LMST-mS [15:0] |
| 7 | updated every 250mS. | LMST-mS [31:16] |
| 8 | Modified Julian Day number in UTC. | MJDC [15:1] |
| 9 | Binary Coded Decimal UTC - updated | 10S : S : 1/10S : 1/100S |
| 10 | every 250mS | 10H : H : 10M : M |
| 11 | Binary Coded Decimal LMST - updated | 10S : S : 1/10S : 1/100S |
| 12 | every 250mS. | 10H : H : 10M : M |
| 13 | BCD Australian Eastern Standard Time | 10S : S : 1/10S : 1/100S |
| 14 | (AEST) - updated every 250mS. | 10Hr : Hr : 10Min : Min |
| 15 | BCD Date - updated every 250mS. | 10Mth : Mth : 10D : D |

| | | |
|---|---|---|
| 16 | | 1000Y : 100Y : 10Y : Y |
| 17 | ASCII coded month (eg “Jan”) and Binary | “a” \| “j” |
| 18 | Month Number (BMN) - updated every 250mS. | BMN \| “n” |
| 19 | ASCII coded Day-of-Week (eg “Mon”) and | “o” \| “m” |
| 20 | binary day number (BDN) - updated every 250mS. | BDN \| “n” |
| 21 | BCD Day-number-of-year - updated every 250mS. | 0 : 100D : 10D : D |
| 22 | dUT1 and dUTC[1] | dUT1[15:8] \| dUTC[7:0] |
| 23 | BCD 1 Second Phase Tick and Valid Flag- updated | 100µS: 10µS:µS:1/100µS |
| 24 | every 250mS. [2] | VF(msb) \| 100mS:10mS:1mS |
| 25[3] | Binary 1 Second Phase Tick (nS) and Valid Flag- | 1STP[15:0] |
| 26[3] | updated every 250mS. | VF(msb) \| 1STP[30:16] |
| 27[3] | DUT1 mS (signed integer) | dUT1[15:0] |
| 28 | Unused | 0 |
| ..... | | |
| 55[3] | Status Bits | 0: Not Locked<br>1: Clock not Set<br>2: IER Table Empty<br>3: IER Table nearly Empty.<br>4: BAT Out of Sync. |

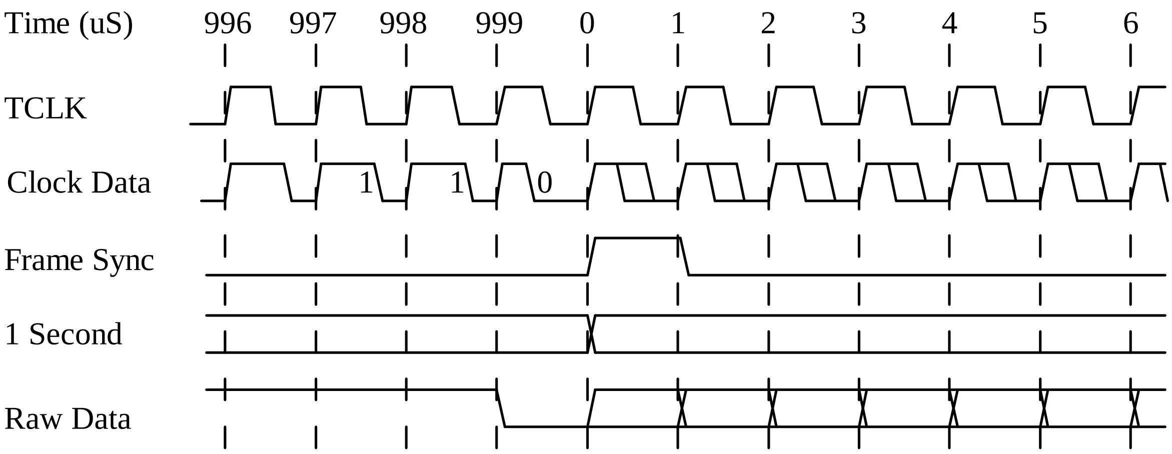
Notes:

1. Use dUT1 in word 27 in lieu of this one.
2. Use binary 1 Second Tick Phase in words 25 and 26 in lieu of this one.
3. New fields over the MK IV ATDC.

## 2.2 Signal Timing.

### 2.2.1 Frame Timing.

The fundamental timing in the Distributed Clock is shown below. Data frames are 1000 bits long. The start of each data frame is marked by one of two SYNC signals sent by the Clock. These are a 1 milli-second SYNC and a 1 second SYNC. The 1 milli-second SYNC consists of at least one “0”, followed by 31 1’s followed by a single “0”. The 1 second SYNC consists of 47 1’s followed by a single “0”. A data word is 17 bits long, of which there are 16 data bits and one STOP (“0”) bit. Thus each frame os made up of 56 WORDS, and 48 SYNC bits. Words 0 to 54 contain data, and word 55 contains a CRC value calculated on all the bits since the start of the current Frame.



FRAME TIMING

### 2.2.2 Timing Signals.

There are 2 BNC output timing signals, 2 BNC input timing signals, and 4 X 10 pin connector output timing signals on the board.

#### 2.2.2.1 Input Signals.

The two input timing signals are the Clock reference signal (either 5MHz or 10MHz) and the external 1 Second Tick. The Clock Reference is the top BNC connector and the External 1 Second Tick is the second BNC connector from the top. Both circuits have an input impedance of 2.4KΩ. If connecting by long lengths of coaxial be sure to terminate correctly.

The Clock Reference signal level must be greater than 200mV p-p and less than 7V p-p, and should have a duty cycle of around 50%. The signal is capacitively coupled to the detection circuit (26LS32).

The External 1 Second Tick is DC coupled to the detection circuit (26LS32) and is set to a threshold of about 1.8V.

On the MKVIb version of the ATDC, there is a 26 way header configured to give 8 balanced inputs and 4 single ended outputs plus 2 Grounds. The 8 balanced lines can be jumper configured for balanced or single ended signals (the latter having a reference voltage of about 1.8V). For ease of connection to the outside world, a ribbon cable from the 26 way PCB connector to a 25 pin D connector can be made. The pinout for such a cable is included in the following table.

| Signal | 26 pin<br>PCB connector | 25 Pin D<br>Connector |
|---|---|---|
| +In 1 | 1 | 1 |
| -In 1 | 2 | 13 |
| +In 2 | 3 | 2 |
| -In 2 | 4 | 14 |
| +In 3 | 5 | 3 |
| -In 3 | 6 | 15 |
| +In 4 | 7 | 4 |
| -In 4 | 8 | 16 |
| +In 5 | 9 | 5 |
| -In 5 | 10 | 17 |
| +In 6 | 11 | 6 |
| -In 6 | 12 | 18 |
| +In 7 | 13 | 7 |
| -In 7 | 14 | 19 |
| +In 8 | 15 | 8 |
| -In 8 | 16 | 20 |
| Out 1 | 17 | 9 |
| Out 2 | 18 | 21 |
| Out 3 | 19 | 10 |
| Out 4 | 20 | 22 |
| Out 5 | 21 | 11 |
| Out 6 | 22 | 23 |
| Out 7 | 23 | 12 |
| Out 8 | 24 | 24 |
| GND | 25,26 | 13 |

### 2.2.2.2 Output Signals.

The two output BNC timing signals are the Clock’s internal 1 Second Tick and the Clocks Internal 1 mS Tick. Both should be capable of driving a 50Ω load. If driving long lengths of coaxial cable, terminate same.

The 4 X 10 pin connector outputs are the balanced TTL timeframe signal. They are designed to be extended to 9 pin female D connectors that can be mounted on appropriate panels in the system, using standard mass terminated insulation displacement connector (IDC) systems. The pinouts are as follows:

| Signal | 10 pin<br>PCB connector | 9 Pin D<br>Connector |
|---|---|---|
| +Clock | 1 | 1 |
| -Clock | 2 | 6 |
| GND | 9,10 | 5 |

When installed into a standard PC, the 9 pin D connectors should sit on unused I/O slot brackets

## 2.3. Interrupts.

IRQ setting:

5, 10, 11, 12, 13. Jumper selection.

Interrupt sources:

1 Millisecond,
Internal 1 Second,
External 1 Second,
Lost Reference (PLL),
Lost External 1 Second, and
Self.

Control:

Bitwise selectable along with a master interrupt enable bit.

## 2.4. Hardware Configuration.

### 2.4.1. Board Address.

The clock occupies 32 X 16 bit address on PC's I/O bus. Base address is set by configuring a 10 bit DIP switch located on the board. Full 16 bit address decoding is done. The default setting is for a base address of 0x2300.

The memory map is as follows:

Memory map.

| Offset | Description. | Section |
|---|---|---|
| 0x0 | Buffer Address Register. | 3.1 |
| 0x2 | Buffer Data Register. | 3.2 |
| 0x4 | General Control and Status Register. | 3.3 |
| 0x6 | Interrupt Control Register. | 3.4 |
| 0x8 | Interrupt Status Register. | 3.5 |
| 0xA | BAT word 0 (0-15) Register. | 3.6 |
| 0xC | BAT word 1 (16-31) Register. | 3.6 |
| 0xE | BAT word 2 (32-47) Register. | 3.6 |
| 0x10 | BAT word 3 (48-63) Register. | 3.6 |
| 0x12 | Phase word 0 (0-15) Register. | 3.7 |
| 0x14 | Phase word 1 (16-23) Register. | 3.7 |
| 0x16 | Up-time word 0 (0-16) Register. | 3.8 |
| 0x18 | Up-time word 1 (16-23) Register. | 3.8 |
| 0x1A | Interrupt Latency Register. | 3.9 |

### 2.4.2. Links.

The board has 9 links of which 5 must be configured. The remaining 4 are for future use. Settings indicated by a * are the default.

2.4.2.1. Reference Frequency Rate.

The rate of the reference frequency may be set to be either 10MHz or 5MHz.

LK1. Reference Frequency Rate.

OPEN - 10MHz.
CLOSED - 5MHz.*

2.4.2.2. IRQ select line.

Set LK2, 3, 4 and 5 to set which IRQ line to use. The links are essentially a binary code os the IRQ number, where Closed is logic Lo, and Open is logic Hi.

| **“IRQ3”** | **“IRQ2”** | **“IRQ1”** | **“IRQ0”** | |
|---|---|---|---|---|
| (LK1) | (LK2) | (LK3) | (LK4) | IRQ |
| C | O | C | O | 5 |
| O | C | O | C | 10 * |
| O | C | O | O | 11 |
| O | O | C | C | 12 |
| O | O | C | O | 13 |

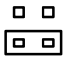
------ All Other selections ---NONE

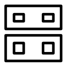
2.4.2.3. Supplementary Tick Phase Inputs. (MK-VIb and later versions only).

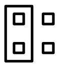
Jumpers JP1 through to JP8 set the input signal mlde on the 8 s Supplementary Tick Phase Inputs. There are 3 possible modes:

1. Single ended TTL with alternate signal lines of each pair OPEN.
2. Single ended TTL with alternate signal lines of each pair GROUNDED
3 Differential TTL.

Supplementary Tick Phase Inputs - Option jumpers JP1-JP8 settings.

 — Single Ended TTL with alternate signal lines of each pair OPEN.

 — Single Ended TTL with alternate signal lines of each pair GROUNDED.

 — Differential TTL.

## 2.4.3 Serial Number PROM.

Each AT Distributed Clock has a serial number PROM. This contains the serial number and other relevant information for each board. Data is held as a preamble followed by a NULL terminated ASCII string (as per the C language). The preamble is any number of 1's, followed by a 0 and then seven other bits (don't cares). The next bit is the start of the data. Each ASCII character is 8 bits and is read from the PROM least significant bit first.

## 2.4.4 Onboard VCO free run bias level setting.

The AT Distributed Clock has a VCO free run bias level circuit. Whenever the External Reference is lost (no incoming signal for at least 1µS), the ATDC is switched to free run mode. In free run mode the phase detector output driver is disabled (tri-stated) and the onboard VCO free run bias level output drivers are enabled. If reasonable synchronization is desired during any External Reference signal loss, adjust the free run bias level to set the VCO’s frequency to match the reference rate.

To adjust the onboard VCO free run bias level, disconnect the External Reference signal from the AT Distributed Clock, and connect it to one of the input channels of a suitable CRO. Connect the other CRO’s input channel to the “RAW_VCO” test pin near the top left hand corner of the PCB and set the CRO to triggeer off the slowest signal (usually the External Reference signal). Adjust the “FREE RUN

ADJ." potentiometer also located at the top left hand corner, until there is minimal drift between the two signals.

## 2.5 Some General Considerations.

### 2.5.1. Board Address.

The Board's base address is set by the 10 bit DIP switch located on the top left corner of the PCB. The value should be set not to conflict with other boards. The default setting is 0x2300. This places it 0x2000 above the default setting of an Event Generator, and allows 2 X BCC interface cards (triple port version).

| Switch*<br>Number | State |
|---|---|
| 0 | On |
| 1 | On |
| 2 | Off |
| 3 | Off |
| 4 | On |
| 5 | On |
| 6 | On |
| 7 | Off |
| 8 | On |
| 9 | On |

*This numbering is the silk screen printing on the PCB.

Summary of other board addresses.

| Card | Address |
|---|---|
| Event Generator | 0x300 |
| BCC Interface (1) | 0x700, 0xb00, 0xf00 |
| BCC interface (2) | 0x1300, 0x1700, 0x1b00 |

# 3.0 Register Description.

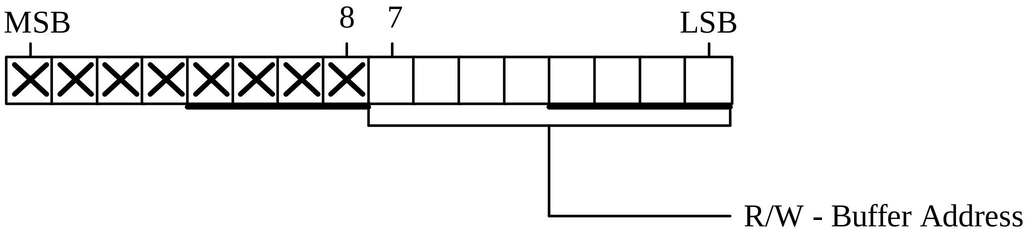
### 3.1. Buffer Address Register.

The Buffer Address Register contains the address of the location in the current buffer where data will be written. Data is written to or read from the buffer through the Buffer Data register. The Buffer Address Register auto-increments its value after each read or write of the Buffer Data. To perform a block transfer, load the Buffer Address Register with the start location in the Buffer, then proceed to do transfers to/from the Buffer Data Register. Care should be taken to exclude any other task from accessing either this register or the Buffer Data register during this time.

Memory map.

| Address Range<br>(hex) | Size<br>(words) | Description. |
|---|---|---|
| 0x00 - 0x3f | 64 | Time Frame data buffer. |
| 0x40 - 0x4f | 16 | Configuration Memory. |
| 0x50 - 0x7f | | Unused (aliased Configuration Memory). |
| 0x80 - 0x8f | 16 | Supplementry Tick Phase counters. |
| 0x90 - 0xbf | | Unused (aliased Supplementry Tick Phase counters). |

BUFFER ADDRESS REGISTER (Offset 0x0)

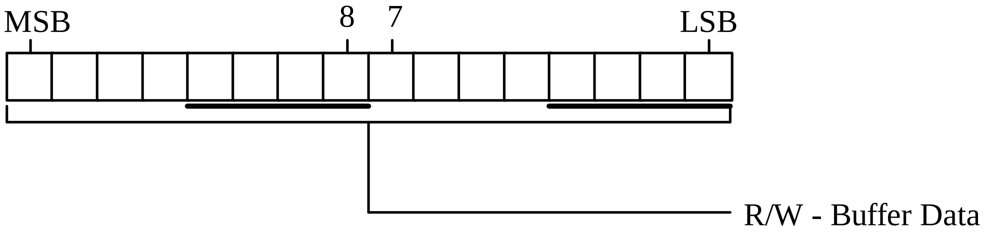


| Mask | Type | Description |
|---|---|---|
| 00ff | R/W | The address to point to in the current bank of the Time Frame data buffer, the Configuration Memory or the Supplementary Tick Phase counters. **Note:** not all versions of the ATDC hardware support the Supplementary Tick Phase counters. |

## 3.2. Buffer Data Register.

The Buffer Data Register is a 16 bit register that is a window into the Data Buffer at the address set in the Buffer Address Register. On completion of either a 16 bit transfer, or the transfer of the most significant byte, the Buffer Address register contents is incremented.

BUFFER DATA REGISTER (Offset 0x2)

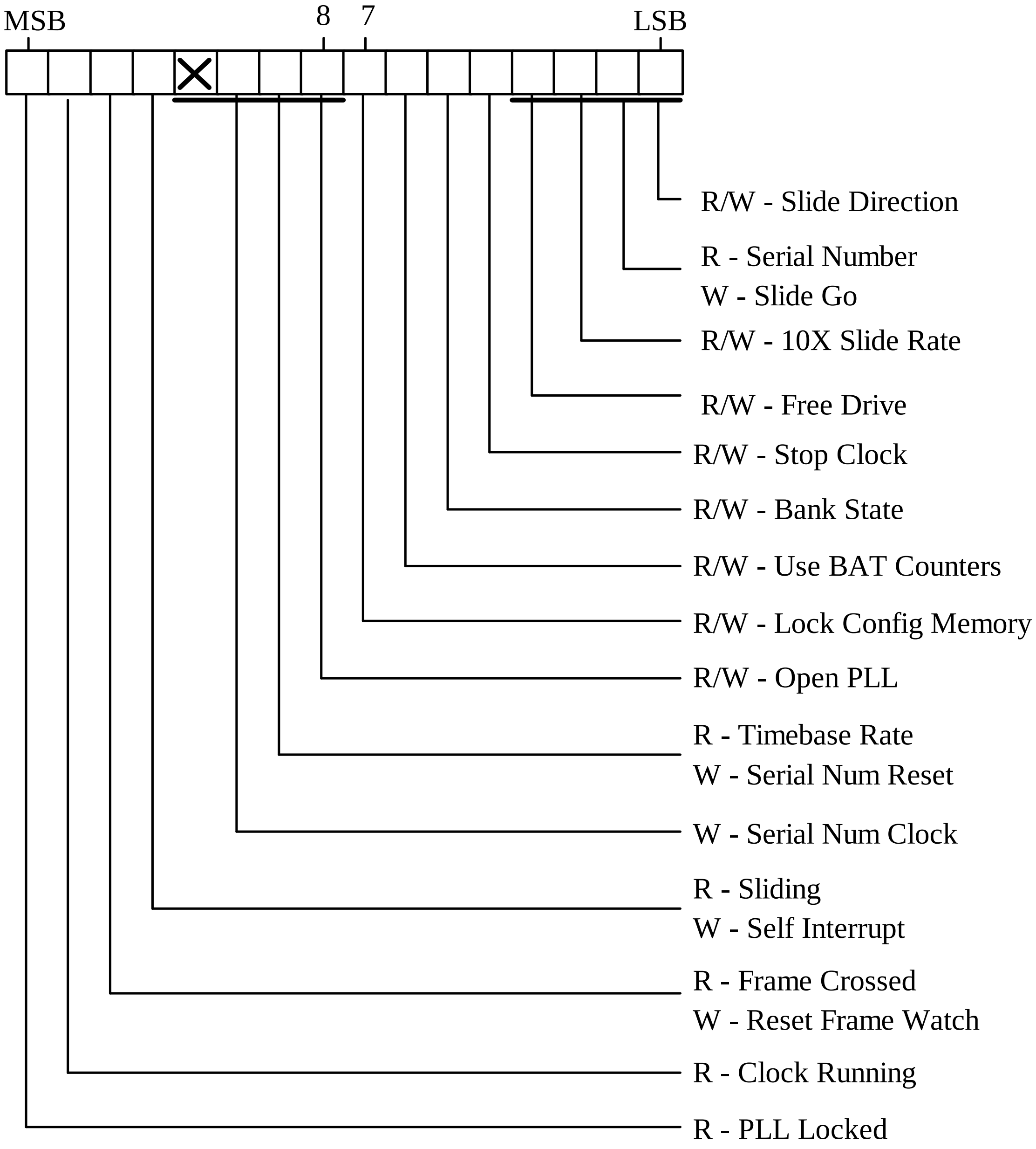


| Mask | Type | Description |
|---|---|---|
| ffff | R/W | 16 bits of data to write to or read from the Buffer at the address set in the Buffer Address Register. |

## 3.3. General Control and Status Register.

The General Control and Status Register contains the various bits that control the operation of the clock or allow monitoring of the clock’s status. When writing to this register, the ISA bus cycle is extended to last at least 1µS, to be compatible with the slower Serial Number PROMs. During normal operation this register would rarely be accessed.

CONTROL AND STATUS REGISTER (Offset 0x4)



| Mask | Type | Description |
|---|---|---|
| 0001 | R/W | Slide Direction. Direction in which to slide the clock. When cleared the Clock to be Advanced, and when set Clock to be Retarded. |
| 0002 | W | Slide Go. Initiates the slide mechanism to advance or retard the clock depending upon the state of the Slide Direction bit. The amount of slide may be set to 200nS or ten times this by the 10X Slide Rate bit. It is not necessary to clear this bit after setting it as the writing process generates the required internal pulse. |
| 0004 | R/W | 10X Slide Rate. When clear, the slide rate is 200nS each time the Slide Go is activated and when set, the slide rate is ten time this. |
| 0008 | R/W | Free Drive. Allows the slide mechanism to run continuously, and in the direction set by the Slide Direction bit. When sliding, the clock effectively runs 10% faster or slower, depending upon the Slide Direction bit. |

0010 R/W Stop Clock. When set the BAT counters will stop on the next 1mS boundary. The entire Frame generation mechanism will continue to run. When cleared, the BAT counters will be ‘armed’ to go at either the next external 1 Second Tick, if a Valid status exists on this line, or the internal 1 second tick. The Clock must be stopped to load new values into the BAT counters.

0020 R/W Bank State. Sets which bank to “look” at. This bit is XORed with the LSB of the Frame counter, so the Bank State toggles on each Frame.

0040 R/W Use BAT Counters. When cleared (default state), the 4 words of BAT that are put out onto the Clock Bus are taken directly from the internal counters. When set, the contents of the data Buffer is used.

0080 R/W Configuration Memory Lock. When set the Configuration Memory is locked. The Configuration Memory is mapped into the top half of the internal memory region; the lower half being occupied by the Frame Buffer memory. When this bit is cleared, the Configuration Memory can be written too. The Configuration Memory is 16 X 16 bit memory available to the user to hold basic configuration data such as DUTC, DUT1, Longitude, etc. This bit may also be used to indicate that the clock has been set.

0100 W Open PLL. Setting this bit will open the Phased Locked oscillator loop. Clearing this bit will close the Phased Locked oscillator loop. The loop is opened by tri-stating the output driver for the VCO control pin and enabling the output drivers for the onboard bias level circuit. See the section on Onboard VCO free run bias level setting.

0200 R Timebase Rate. A LO indicates the ATDC’s internal clock rate (timebase) is 10MHz. A HI indicates the ATDC’s timebase is 50MHz. A 10MHz timebase will give a Tick Phase resolution of 50nS and a 50MHz timebase will give a Tick Phase resolution of 10nS.

W Serial Num Reset. Setting this bit will reset the Serial Number PROM. This should be done before the start of any reading of this PROM. It is not necessary to clear this bit after setting it as the writing process generates the required internal pulse.

0400 W Serial Num Clk. Setting this bit will increment the pointers in the Serial Number PROM. It is not necessary to reset this bit as the writing process generates the required internal pulse.

1000 R Sliding. A Hi indicates that a Slide operation is in progress. Once a slide has been initiated by asserting either a Slide Go or a Free Drive, the software should check this bit and defer any further Slide operations until cleared.

W Self Interrupt. Setting this bit will cause an interrupt to be issued if the Self Interrupt bit and the Master Interrupt Enable bits are set.

2000 R Frame Crossed. A Hi indicates that a new frame occurred since the Frame Watch was last set. Useful when performing random access to various parts of the clock when crossing Frame boundaries is undesirable. If synchronous reads of Frame data are required, consider making use of the 1mS interrupt and the associated Interrupt Service Routine.

W Reset Frame Watch. Setting this bit clears the Frame Watch bit. Useful when performing random access to various parts of the clock when crossing Frame boundaries is undesirable. To use this feature, set this bit, perform the operation as needed, then read this bit and if set, a Frame boundary has been crossed. If synchronous reads of Frame data are required, consider making use of the 1mS interrupt and the associated Interrupt Service Routine.

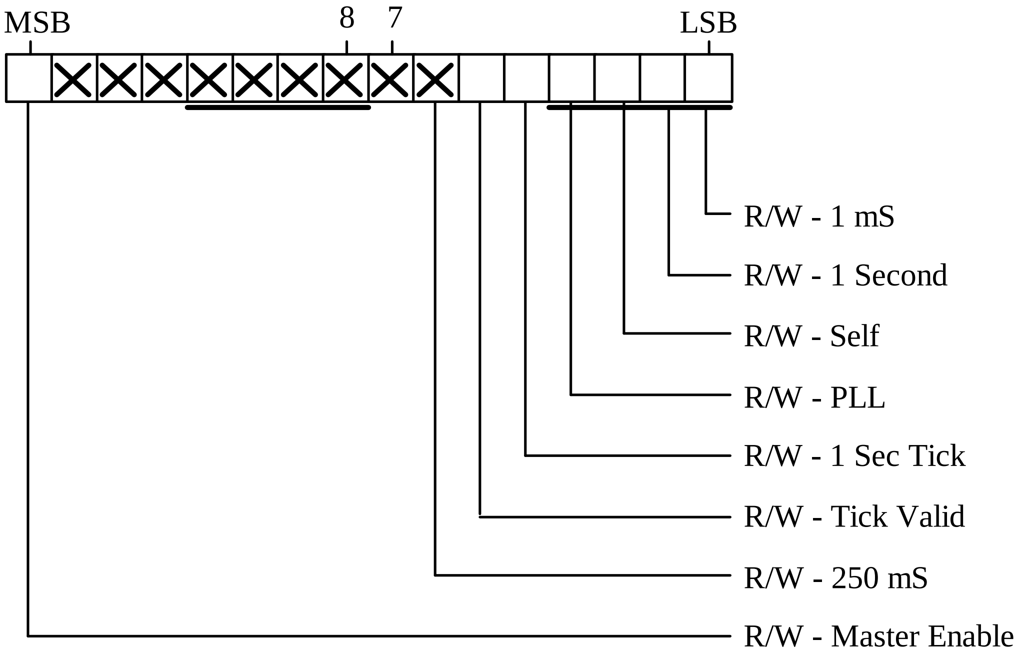
4000 R Clock Running. A Hi indicates the Clock is running. To start the Clock clear the Clock Stop bit, and wait for the next external 1 Second Tick, if there, or else the internal 1 Second Tick.

| | | |
|---|---|---|
| 8000 | R | PLL Locked. A Hi indicated that the PLL is locked to the external reference. |

## 3.4. Interrupt Control Register.

The Interrupt Control Register allows control of the interrupt mechanism. There is one bit to enable or disable each interrupt point available. There is also a Master interrupt control bit which can be used to globally turn on or turn off all interrupts. Each interrupt control point has a corresponding interrupt status point.

INTERRUPT CONTROL  REGISTER (Offset 0x6)

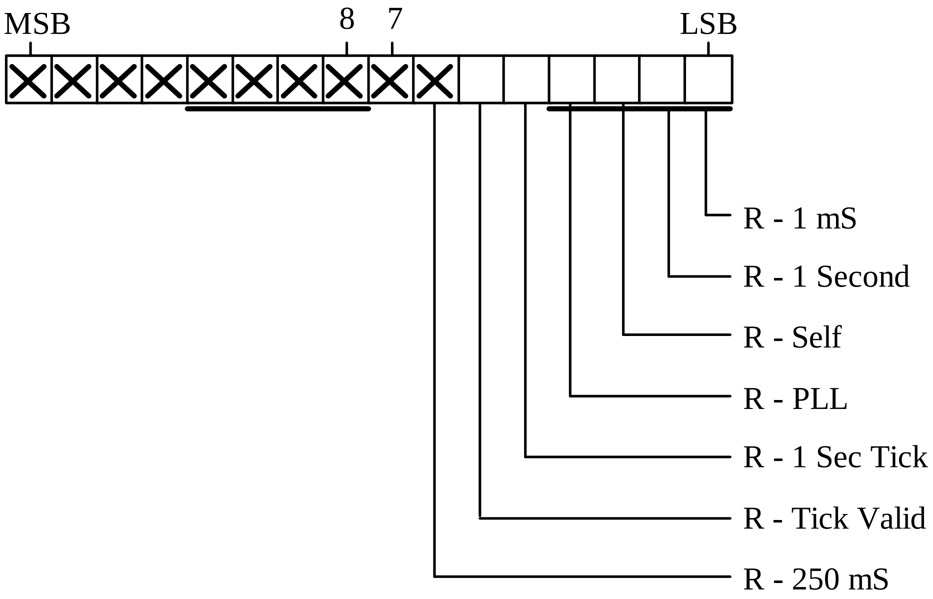


| Mask | Type | Description |
|---|---|---|
| 0001 | R/W | 1 mS. When set, the 1mS frame signal will cause an interrupt. The Master Enable bit must also be set. |
| 0002 | R/W | 1 Second. When set, the internal 1 Second signal will cause an interrupt. The Master Enable bit must also be set. |
| 0004 | R/W | Self. When set, writing a Hi to the Self Interrupt bit in the Control and Status Register will cause an interrupt. The Master Enable bit must also be set. |
| 0008 | R/W | PLL. When set, a change in the PLL state will cause an interrupt. The Master Enable bit must also be set. |
| 0010 | R/W | 1 Sec Tick. When set, the external 1 Second Tick will cause an interrupt. The Master Enable bit must also be set. |
| 0020 | R/W | Tick Valid. When set, a change of state in the Tick Valid status signal will cause an interrupt. The Master Enable bit must also be set. |
| 0040 | R/W | 250 mS. When set, each 250mS mark will cause an interrupt. The Master Enable bit must also be set. |
| 8000 | R/W | Master Enable. When set, both the interrupt mechanism and the IRQ line chosen by the IRQ select links will enabled. |

## 3.5. Interrupt Status Register.

The Interrupt Status Register reflects the current state of the interrupt mechanism. Each bit indicates whether or not it is in an interrupting state. Only those interrupt points set for interrupt in the Interrupt Control Register will cause an interrupt. To know which points caused an interrupt, logically AND the contents of this and the Interrupt Control Register together. Provided both the master Interrupt Enable bit and an appropriate interrupt point is set, upon that condition occurring, the interrupt line feeding the CPU is set HI. It remains high until the Interrupt Status Register is read. The contents of this register is cleared each time it is read; therefore any interrupt service routine should read and locally store the contents of this register and then refer only to the locally stored value.

INTERRUPT STATUS REGISTER (Offset 0x8)

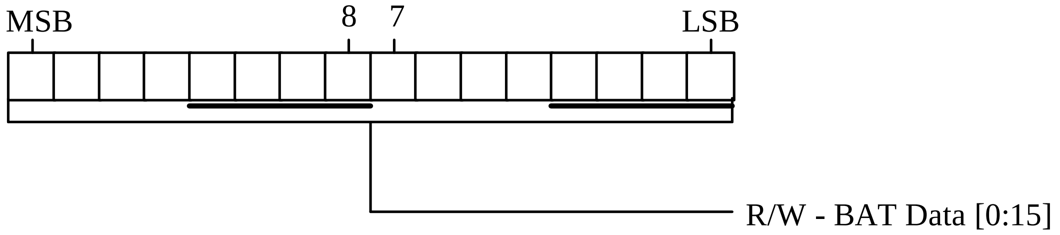


| Mask | Type | Description |
|---|---|---|
| 0001 | R | 1 mS. When set, the 1mS frame signal caused an interrupt. For confirmation logically AND this bit with the corresponding bit in the Interrupt Control Register. |
| 0002 | R | 1 Sec. When set, the 1 Sec signal caused an interrupt. For confirmation logically AND this bit with the corresponding bit in the Interrupt Control Register. |
| 0004 | R | Self. When set, the Self Interrupt signal caused an interrupt. For confirmation logically AND this bit with the corresponding bit in the Interrupt Control Register. |
| 0008 | R | PLL. When set, the PLL signal caused an interrupt. For confirmation logically AND this bit with the corresponding bit in the Interrupt Control Register. |
| 0010 | R | 1 Sec Tick. When set, the external 1 Second Tick signal caused an interrupt. For confirmation logically AND this bit with the corresponding bit in the Interrupt Control Register. |
| 0020 | R | Tick Valid. When set, a change of state on the Tick Valid status signal will cause an interrupt. For confirmation logically AND this bit with the corresponding bit in the Interrupt. |
| 0040 | R | 250 mS. When set, the 250mS mark caused an interrupt. For confirmation logically AND this bit with the corresponding bit in the Interrupt Control Register. |

## 3.6. BAT Registers.

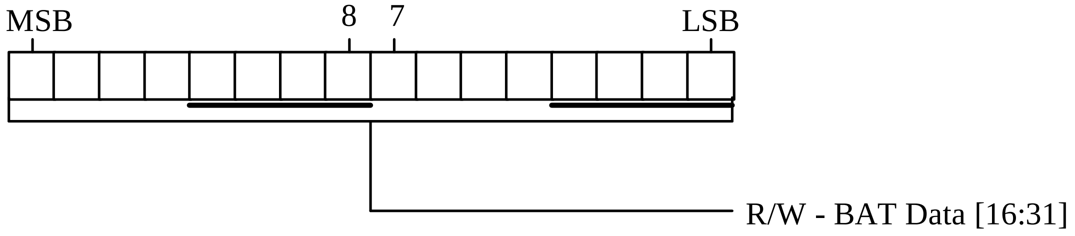
The BAT registers are a set of 4, 16 bit registers, that access the 64 bit Binary Atomic Time counters in the clock. The contents of the counters may be changed by writing the new values to each of these registers**. This can only be done when the Clock has been stopped.** The contents of the counters can be read at any time, but the value returned will only be that at the beginning of the current millisecond time (ie current Frame time). If performing a random read, use the Frame Boundary warning flag available in the General Control and Status Register to indicate if a new Frame occurred during the read operation. When writing to these registers, the ISA bus cycle is extended to last at least 1µS

BAT 0 REGISTER (Offset 0xA)



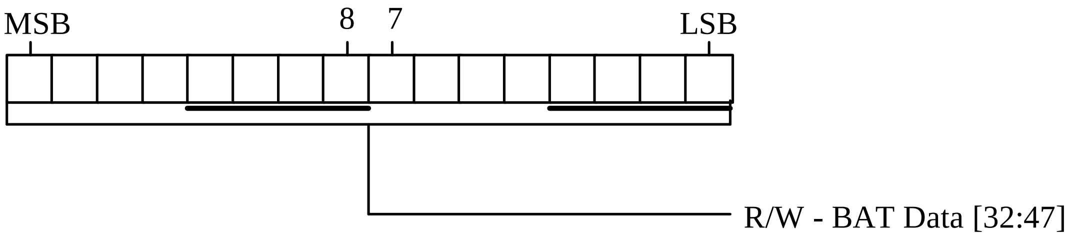
| Mask | Type | Description |
|---|---|---|
| ffff | R/W | BAT data. Bits 0:15. |

BAT 1 REGISTER (Offset 0xC)



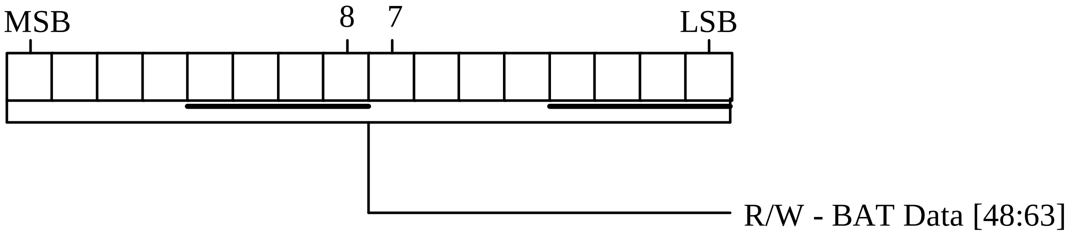
| Mask | Type | Description |
|---|---|---|
| ffff | R/W | BAT data. Bits 16:31. |

BAT 2 REGISTER (Offset 0xE)



| Mask | Type | Description |
|---|---|---|
| ffff | R/W | BAT data. Bits 32:47. |

BAT 3 REGISTER (Offset 0x10)

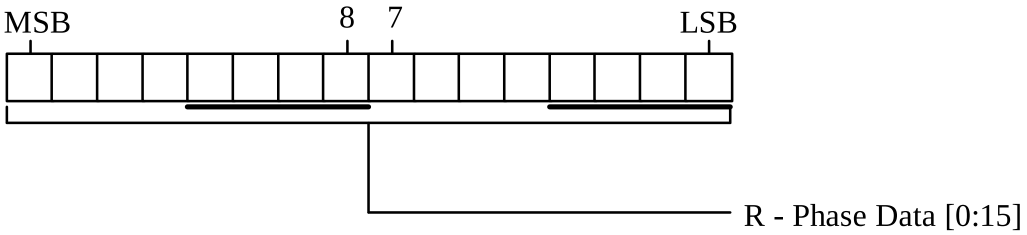


| Mask | Type | Description |
|---|---|---|
| ffff | R/W | BAT data. Bits 48:63. |

## 3.7. Phase Registers.

The Phase Registers contain the binary count of the number of 50nS (10MHz timebase version) or 10nS (50MHz timebase version) ticks that exist between the External 1 Second Tick and the Clock's Internal 1 Second Tick. This value only has meaning if an External 1 Second tick is connected. If the count exceeds 20,000,020 (0x1312d14) or 1,00,000,020 (5f5e114) for the respective timebase versions, then the "Tick Valid" status bit is cleared and remains cleared until an external 1 Second Tick occurs, and if enabled, an appropriate interrupt generated. The counters are cleared on the arrival of the External 1 Second Tick, and the value captured on occurrence of the Internal 1 Second Tick.

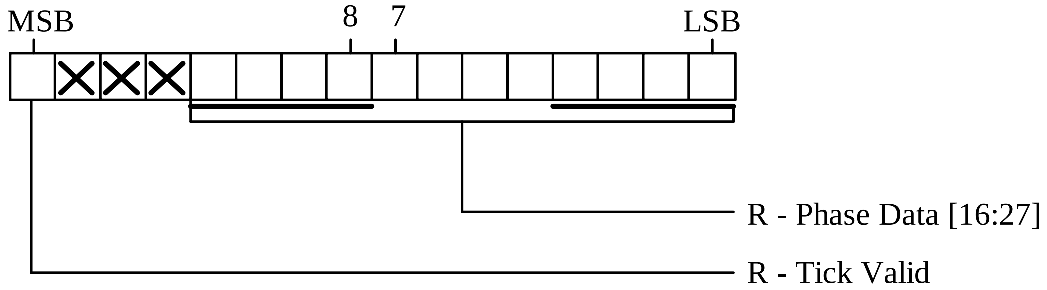
PHASE 0 REGISTER (Offset 0x12)



| Mask | Type | Description |
|---|---|---|

| | | |
|---|---|---|
| ffff | R | Phase data, bits 0:15. |

PHASE 1 REGISTER (Offset 0x14)

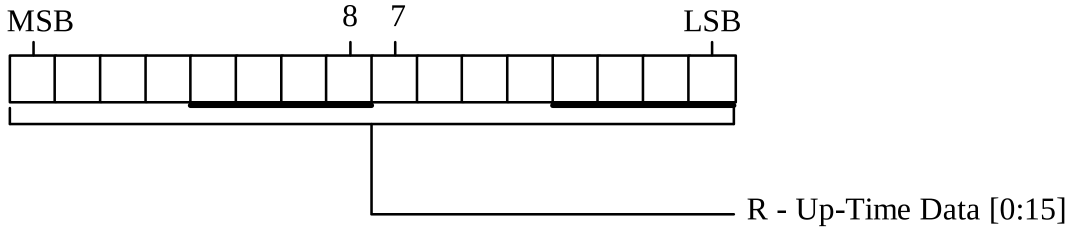


| Mask | Type | Description |
|---|---|---|
| 00ff | R | Phase data, bits 16:27. |
| 8000 | R | Tick Valid. A Hi indicates the Tick Phase is valid. The Tick Phase becomes invalid (Lo) when an external 1 Second Tick has not been received for over 10,000,010 counts of the internal 10MHz clock (or 50,000,010 counts with an internal 50MHz clock). |

## 3.8. Up-time Registers.

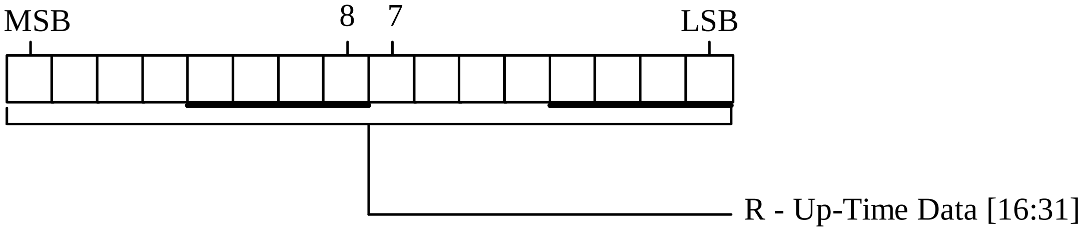
The Up-time registers are a set of 2, 16 bit registers, that keep a count of the number of seconds that the clock has been running. As this is an unsigned integer, the value will roll over after about 136 years.

UP-TIME 0 REGISTER (Offset 0x16)



| Mask | Type | Description |
|---|---|---|
| ffff | R | Phase data, bits 0:15. |

UP-TIME  1 REGISTER (Offset 0x18)

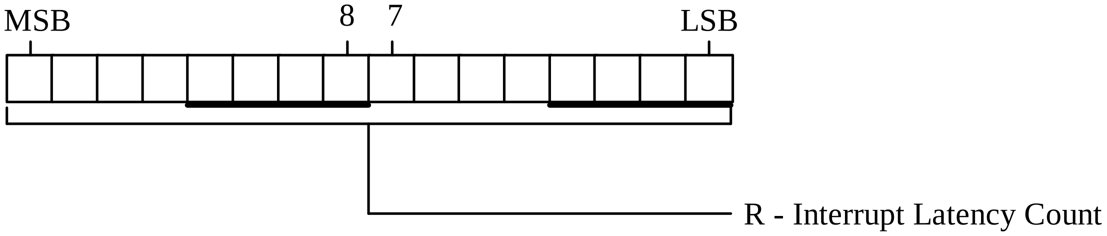


| Mask | Type | Description |
|---|---|---|
| ffff | R | Phase data, bits 16:31. |

### 3.9. Interrupt Latency Register.

The Interrupt Latency register is a 16 bit register, that counts of the number of micro seconds between the time an interrupt signal is generated (IRQx line is asserted) and when the Interrupt Status Register is read. This register allows the controlling program to monitor the time it takes to service the interrupt.

INTERRUPT LATENCY  REGISTER (Offset 0x1A)



| Mask | Type | Description |
|---|---|---|
| ffff | R | Interrupt latency count. This is the number of µS between the IRQx line being asserted and the Interrupt Status Register being read. Counter halts on 0x8000. |

## 4.0 Programming.

### 4.1. General.

This section is still under construction.

## 5.0 Programming Examples.

### 5.1. General.

This section give some examples on how to program and use the clock at a hardware level. It is intended for those people who are writing code for the controlling system.

The examples were written for Borland C®, under MS-DOS®. The following #defines are applicable to the examples.

```
#define IC_REG                              0x6
#define IS_REG                              0x8
```

```
/* General status/control register bits */
#define CSR_SLIDE_DIR                 0x0001
#define CSR_SLIDE_ONCE                0x0002
#define CSR_SERIAL_NUM_MASK           0x0002
#define CSR_SLIDE_10X                 0x0004
#define CSR_SLIDE_CONTINUOUS          0x0008
#define CSR_STOP_CLOCK                0x0010
#define CSR_BANK_STATE                0x0020
#define CSR_BAT_BUFFER_DATA           0x0040
#define CSR_INHIBIT_AUTO_INC          0x0080
#define CSR_SELF_INTERRUPT            0x0100
#define CSR_RESET_SN_PROM             0x0200
#define CSR_INC_SN_DATA               0x0400
#define CSR_SLIDING_MASK              0x1000
#define CSR_FRAME_WATCH_MASK          0x2000
#define CSR_SERIAL_NUMBER             0x8000

/* interrupt control and interrupt status register bits */
#define INTR_1MS                      0x0001
#define INTR_INT_1SEC                 0x0002
#define INTR_SELF                     0x0004
#define INTR_PLL                      0x0008
#define INTR_EXT_1SEC                 0x0010
#define INTR_INVALID_EXT_TICK         0x0020
#define INTR_GLOBAL_ENABLE            0x8000
#define INTR_ALL                      0x007f

#define OCW1P                         0x21
#define OCW2P                         0x20
#define OCW1S                         0xa1
#define OCW2S                         0xa0

#define PIC_DISABLE                   0
#define PIC_ENABLE                    1
```

## 5.2. Interrupt mechanism.

To enable the interrupt mechanism, the user must turn on the interrupts that are required and the Global Interrupt Enable bit. It may be desirable to first turn on the individual interrupt sources, clear the interrupt Status Register, then turn on the Global Interrupt Enable bit. Only when the Global Interrupt Enable bit has been set will the interrupt signal leaving the board work.

```
/* io_port is base address of Clock card */
/* irq is the irq number the board is set to */


void example(unsigned int io_port, int irq)
{
```

```
/* turn on all interrupts except Global Interrupt Enable bit */
    outpw(io_port + IC_REG, INTR_ALL);

/* purge any pending interrupt */
    inpw(io_port + IS_REG);

/* turn on all interrupts bits */
    outpw((*settings).io_port + IC_REG, INTR_GLOBAL_ENABLE | INTR_ALL);

/* Enable Programmable Interrupt Controller */
    SetPic(irq, PIC_ENABLE);

......................................

}
```

```
int SetPic(int irq, int function)
{

    disable(); /* turn off all interrupts during this operation */
    if(function == PIC_ENABLE) {
        if(irq < 8) outp(OCW1P, inp(OCW1P) & ~(1 << irq));
        else {
            outp(OCW1P, inp(OCW1P) & ~0x04);
            outp(OCW1S, inp(OCW1S) & ~(1 << (irq - 8)));
            }
        }
/* function is PIC_DISABLE */
    else {
        if(irq < 8) outp(OCW1P, inp(OCW1P) | (1 << irq));
        else {
            outp(OCW1S, inp(OCW1S) | (1 << (irq - 8)));
            if((inp(OCW1S) & 0xff) == 0xff) outp(OCW1P, inp(OCW1P) | 0x04);
            }
        }

    enable();       /* turn on interrupts */

    return(0);

}
```

## 5.3. Slide mechanism.

The slide mechanism allows the clocks time to be shifted with respect to the external 1 second tick. The shift may be forward or backward, and may be controlled or free running. The controlled mode may be either by 200nS jumps or 2μS jumps. The slide mechanism operates by effectively changing the

speed of the internal clock by exactly 10% for as many 1MHz cycles as required. The operation takes place starting on the second µS after the write operation has been completed. Each 200nS jump requires 1µS to complete, and each 2µS jump requires 10µS to complete. Once the slide has been initiated, the software should check the state of the SLIDING bit in the CSR Register. If TRUE, a slide is in progress and any further slide requests should be deferred.
In this example the routine is given the number of nS that the clock needs to be slid.

```
void slide(long int distance)
{
int direction;   /* direction to slide */
int n;           /* general variable */
int m;           /* general variable */

/* setup direction bit */
        distance >= 0) direction = CSR_SLIDE_DIR;
        else direction = 0;

        distance = distance / 200;   /* get number of 200nS jumps */
        if(distance < 0) distance = -distance;

/* slide clock the required number of 2µ jumps by using the CSR_SLIDE_10X bit */
        n = distance / 10;
        while(n > 0) {
                outpw(io_port + CS_REG, CSR_SLIDE_10X | direction);
                outpw((*settings).io_port + CS_REG, CSR_SLIDE_10X |
                        direction | CSR_SLIDE_ONCE);
                m = 0;
                while((inpw(io_port + CS_REG) & CSR_SLIDING_MASK) != 0)
                        if(++m > 1000) break;
                --n;
                }

/* slide clock the required number of 200nS jumps. */
        n = distance % 10;
        while(n > 0) {
                outpw((*settings).io_port + CS_REG, direction);
                outpw((*settings).io_port + CS_REG, direction | CSR_SLIDE_ONCE);
                m = 0;
                while((inpw(io_port + CS_REG) & CSR_SLIDING_MASK) != 0)
                                if(++m > 1000) break;
                --n;
                }
        return;

}
```

## 5.4. Reading the serial number.

The serial number can be read by recovering the serial data held in the Serial Number PROM. The first operation is to reset the prom by writing a 1 to the prom reset bit. Then repeatedly read the serial number data followed by writing a 1 to the serial number prom clocking bit in the CSR Register.

```
int get_serial_num(int addr, char *string, int bufsize)
{
int n;
int m;
int mask;

    for(n = 0; n < bufsize; ++n) string[n] = 0; /* clear buffer */

    outpw(addr, CSR_RESET_SN_PROM); /* hit PROM reset line */

// wait a short while by reading an i/o port. Each i/o operation takes about 1µS
    for(n = 0; n < 100; ++n) inpw(addr);

    n = 0;
// scan for preamble
    while(1) {
        inpw(addr);
        if((inpw(addr) & CSR_SERIAL_NUM_MASK) == 0) break;
        ++n;
        if(n >= 1000) return(-1);   /* should have got 0 by now */
        outpw(addr, CSR_INC_SN_DATA); /* inc prom addr */
        }
// got preable now pass next 8 bits
    for(n = 0; n < 8; ++n) {
        outpw(addr, CSR_INC_SN_DATA); /* inc prom addr */
        }
// now start loading bits into an ASCII 'C' string
    n = 0;
    while(n < bufsize) {
        string[n] = 0;
        mask = 1;
        for(m = 0; m < 8; ++m) {
            ddelay(1000);
            if((inpw(addr) & CSR_SERIAL_NUM_MASK) != 0)
                            string[n] = string[n] | mask;
            inpw(addr); /* small delay */
            outpw(addr, CSR_INC_SN_DATA); /* inc prom addr */
            mask = mask * 2;
            }
        string[n] = string[n] & 0x7f;
        if(string[n] == 0) return(strlen(string));
        ++n;
        }
    return(-2);
    }
```

### 5.4. Writing Frame data.

The writing of the frame data should occur just after a frame has completed. This occurs every millisecond. An interrupt is provided for this purpose. The clocks buffer is accessed via a single data register, with the address held in a special address register. With the address register set to auto-increment mode (default state), only the start address need be set. Once set, repeated writes may be performed which will load sequential locations with the data.

```
int LoadData(unsigned int base_addr, int *buffer, int length)
{
int n;

/* assume that this has just been called as a result of a 1mS interrupt */
/* and that BAT is being taken from the internal BAT counters so that */
/* the first 4 registers need not be written to */

/* check length of buffer to write */
        if(length > (DATA_BUFFER_LENGTH - 4)) length =  (DATA_BUFFER_LENGTH - 4);

/* set the address register to the start address of 4 */
        outpw(base_addr, 4);
        for(n = 0; n < length; ++n) outpw(base_addr + DATA_REG, buffer[n]);
        return(n);

}
```

## 6.0 Revisions.

### 6.1. 250mS interrupt added.

A 250mS interrupt added for use on later version to negate the 1mS interrupt.

## 7.0 Engineering Modifications.