Local Time Management System

MS13196

Draft B 28 July 1998

Features

- Compliance with the CCSDS CUC Elapsed Time code standard format
- Compliance with the Serial Time Distribution Protocol Extension of the ESA 4-255 Data Bus
- Time resolution from 2^{-19} to 2^{-22} of a second
- Serial and parallel time synchronisation message reception or stand alone operation
- Parallel microprocessor interface directly compatible with ERC32 and MA31750
- Time stamp and alarm clock facilities
- Pulse and waveform generators, combinable to form a programmable frequency generator
- Stopwatch counting the number of events within a specified time interval or the time span for a specified number of events

Ordering Information

MS13196.10K	Commercial CQFP
MS13196.20K	Military CQFP
MS13196.30K	Space SCC9000 CQFP

-55˚C - 125˚C

- Extension interface for implementation of additional time facilities and error checkers
- Crystal oscillator interface driving an internal or external DPLL to achieve high resolution and free-wheeling capability
- Error detection and management with recovery capability
- ESA SCC part number 9544/007

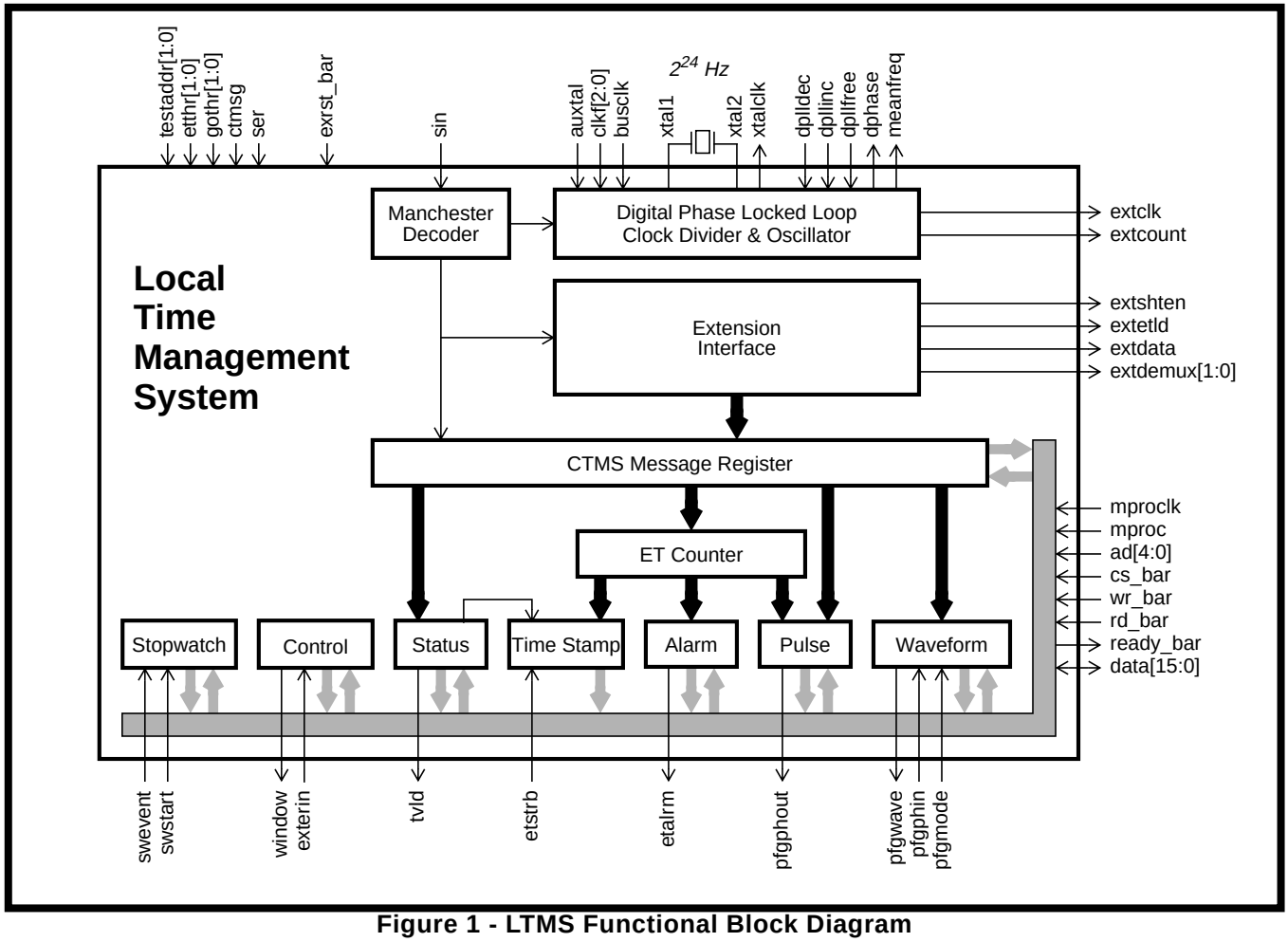


Figure 1 - LTMS Functional Block Diagram

LTMS

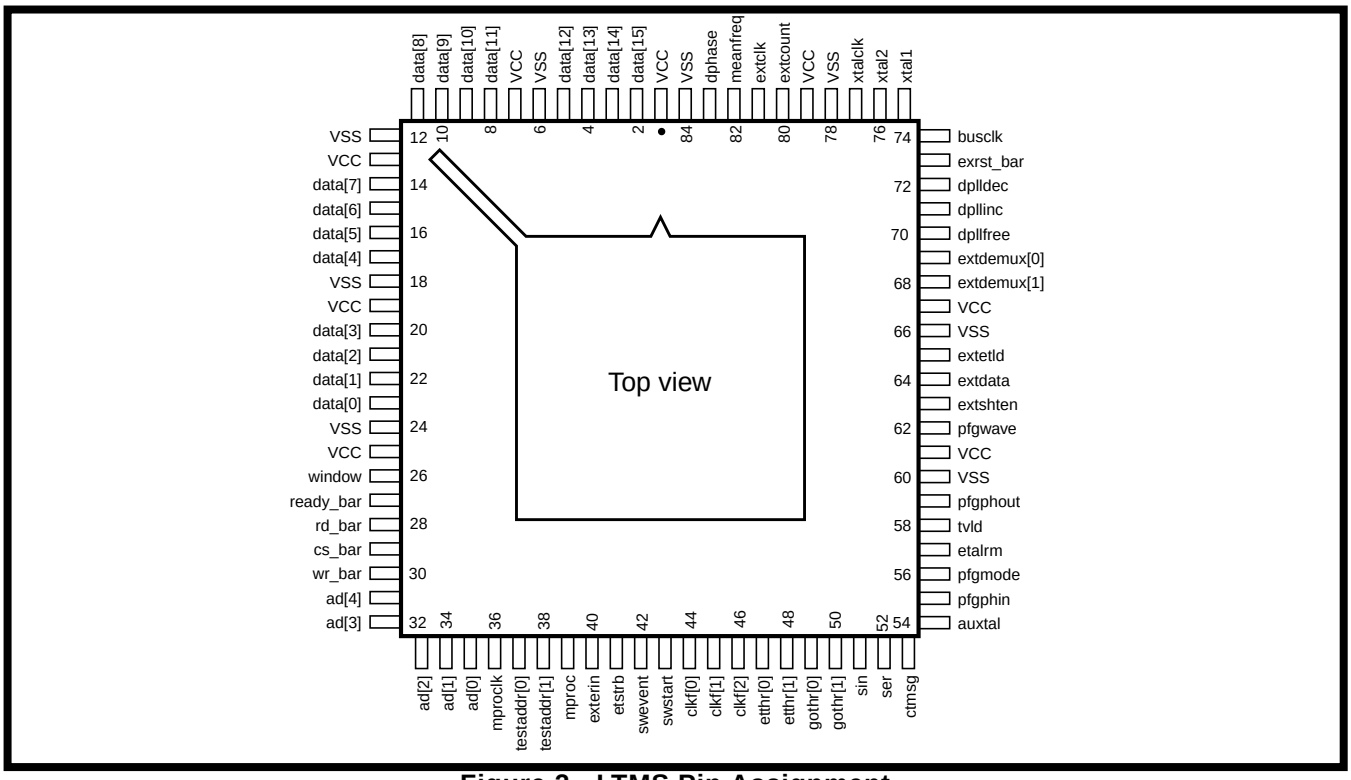


Figure 2 - LTMS Pin Assignment

Pin #	Name	Type
1	VCC	Power Supply
2	data[15]	CMOS-IO
3	data[14]	CMOS-IO
4	data[13]	CMOS-IO
5	data[12]	CMOS-IO
6	V_{SS}	Power Supply
7	V_{CC}	Power Supply
8	data[11]	CMOS-IO
9	data[10]	CMOS-IO
10	data[9]	CMOS-IO
11	data[8]	CMOS-IO
12	V_{SS}	Power Supply
13	V_{CC}	Power Supply
14	data[7]	CMOS-IO
15	data[6]	CMOS-IO
16	data[5]	CMOS-IO
17	data[4]	CMOS-IO
18	V_{SS}	Power Supply
19	V_{CC}	Power Supply
20	data[3]	CMOS-IO
21	data[2]	CMOS-IO

Pin #	Name	Type
22	data[1]	CMOS-IO
23	data[0]	CMOS-IO
24	V_{SS}	Power Supply
25	V_{CC}	Power Supply
26	window	CMOS-O
27	ready_bar	CMOS-O
28	rd_bar	CMOS-I
29	cs_bar	CMOS-I
30	wr_bar	CMOS-I
31	ad[4]	CMOS-I
32	ad[3]	CMOS-I
33	ad[2]	CMOS-I
34	ad[1]	CMOS-I
35	ad[0]	CMOS-I
36	mproclk	CMOS-I
37	testaddr[0]	CMOS-I
38	testaddr[1]	CMOS-I
39	mproc	CMOS-I
40	exterin	CMOS-I
41	etstrb	CMOS-I
42	swevent	CMOS-I

Pin #	Name	Type
43	swstart	CMOS-I
44	clkf[0]	CMOS-I
45	clkf[1]	CMOS-I
46	clkf[2]	CMOS-I
47	etthr[0]	CMOS-I
48	etthr[1]	CMOS-I
49	gothr[0]	CMOS-I
50	gothr[1]	CMOS-I
51	sin	CMOS-I
52	ser	CMOS-I
53	ctmsg	CMOS-I
54	auxtal	CMOS-I
55	pfgphin	CMOS-I
56	pfgmode	CMOS-I
57	etalrm	CMOS-O
58	tvld	CMOS-O
59	pfgphout	CMOS-O
60	V_{SS}	Power Supply
61	V_{CC}	Power Supply
62	pfgwave	CMOS-O
63	extshten	CMOS-O
64	extdata	CMOS-O
65	extetld	CMOS-O
66	V_{SS}	Power Supply
67	V_{CC}	Power Supply
68	extdemux[1]	CMOS-O
69	extdemux[0]	CMOS-O
70	dpllfree	CMOS-I
71	dpllinc	CMOS-I
72	dplldec	CMOS-I
73	exrst_bar	CMOS-I
74	busclk	CMOS-I
75	xtal1	CMOS-I
76	xtal2	CMOS-O
77	xtalclk	CMOS-O
78	V_{SS}	Power Supply
79	V_{CC}	Power Supply
80	extcount	CMOS-O
81	extclk	CMOS-O
82	meanfreq	CMOS-O
83	dphase	CMOS-O
84	V_{SS}	Power Supply

Conventions

An octet comprises 8 bits and a word comprises 16 bits, with the most significant bit to the left. The least significant bit is to the right and has the index 0. Note the difference w.r.t. reference number 3.

For all drawings and diagrams, the most significant bit, octet or word is always to the left.

Integer amounts of seconds are denoted *T*, *T+1*, etc.

When an *x* occurs in a table, the corresponding signal or pin value is ignored.

Abbreviations

CCSDS	Consultative Committee for Space Data Systems
CDMU	Central Data Management Unit
CQFP	Ceramic Quad Flat Package
CTMS	Central Time Management System
CUC	CCSDS Unsegmented Code
DPLL	Digital Phase Locked Loop
ET	Elapsed Time
LSB	Least Significant Bit
LSW	Least Significant Word
LTMS	Local Time Management System
MSB	Most Significant Bit
MSW	Most Significant Word
OBDH	On Board Data Handling
SOS	Silicon On Sapphire

References

1 Time Code Standards, Consultative Committee for Space Data Systems, CCSDS 301.0-B-2, Blue Book, Issue 2, April 1990

2 4-255 Data Bus Modem Specification, European Space Agency, ESA PSS-04-257

3 4-255 Data Bus Protocol Extensions, European Space Agency, ESA PSS-04-256

LTMS

Description

The Local Time Management System (LTMS) is the key element of a decentralised time distribution scheme. Local copies of the centralised Elapsed Time (ET) reference of a spacecraft are maintained by LTMS devices located close to the users. The central ET reference is assumed to be managed by a Central Time Management System (CTMS).

1.1 Overview

The central ET reference on a spacecraft is assumed to be managed by a CTMS. Coherence between the central and the local ET references is maintained by means of synchronisation information distributed from the CTMS. The LTMS performs regular synchronisations with respect to the central ET reference using this information. The local ET reference is implemented in the LTMS ET Counter. The information provided by the CTMS to the LTMS comprises messages with time codes, synchronisation markers and phase references:

- Once every second, a CTMS message including a time code is broadcast from the CTMS to all the LTMS devices in the system. These messages can be acquired through a serial or a parallel microprocessor interface. The serial message is Manchester encoded, which is either self-synchronised or synchronised to a communication bus clock.

- Each CTMS message is followed by a synchronisation marker, which acts as a mark to establish the instant at which the delivered time code is applied (cf. figure 1.1). The synchronisation instants occur at one second intervals and coincide with the increments of the units of seconds bit in the ET counter. No sub-second information needs therefore to be included in messages sent by the CTMS.

- The CTMS provides the LTMS with phase references to which the clock driving the ET counter can be adjusted.

- The clock driving the LTMS ET counter does not need to originate from the CTMS and may thus be generated locally. It may be provided from external circuitry and should then be consistent with the desired ET resolution.

The LTMS checks the format and contents of the received CTMS messages and the timing of the synchronisation marker. If they are found to be correct, a re-synchronisation of the local ET counter is performed at the synchronisation instant when required. The synchronisation consists of loading the received time code into the coarse part of the ET counter and resetting the fine part to zero.

By setting the initialisation flag in the CTMS message, it is possible to temporarily disable the synchronisation checks and force a CTMS message to be used for initialising the LTMS ET counter.

The LTMS user is permanently informed about the time validity by a dedicated output. For more details, the user has access to a status register via the parallel microprocessor interface. The LTMS also has a register dedicated to the status of the rest of the time distribution chain.

If the phase references occur at regular intervals but at a too low rate with respect to the required time resolution, or if they occur irregularly, or if free-wheeling capability is required in case of phase reference transmission failure, the LTMS makes use of a Digital Phase Locked Loop (DPLL). It can be either the internal LTMS DPLL or an external one.

The DPLL provides an internal 2^{22} Hz clock, phase adjusted to the incoming phase references, that drives the ET counter. This internal 2^{22} Hz clock is derived from an external 2^{24} Hz clock. It can be either a clock signal or a crystal oscillator. Phase adjustment is done by modifying the internal 2^{22} Hz clock period by adding or subtracting a 2^{24} Hz clock period to lock on to the phase references.

If the DPLL cannot lock onto the incoming phase references it starts free-wheeling. It then generates a

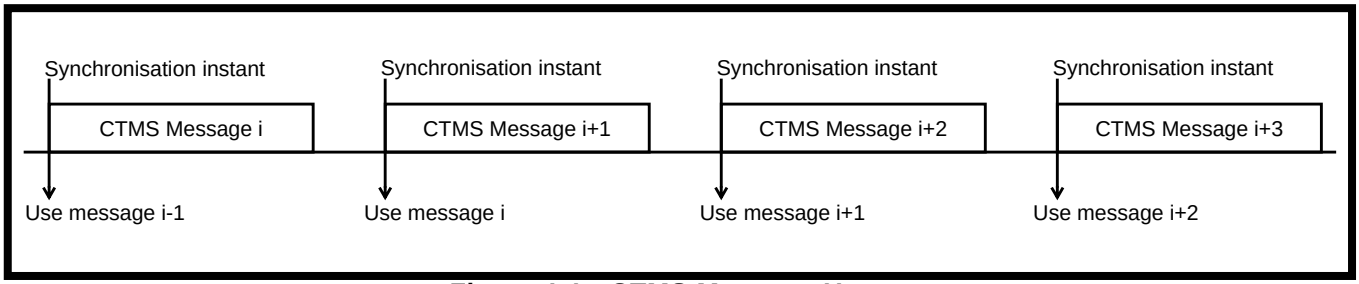


Figure 1.1 - CTMS Message Usage

2^{22} Hz clock that is merely a division of the external 2^{24} Hz clock without performing any phase corrections. Meanwhile the DPLL tries to lock onto the incoming phase references again.

In a stand alone mode, the LTMS operates without a central ET reference. It then maintains its own ET reference and no synchronisation is required. It is however possible to initialise the LTMS from time to time via the parallel microprocessor interface. This provides the capability to use the LTMS even when no CTMS is available in a system.

Figure 1.2 depicts a typical decentralised time distribution system with the LTMS acquiring the CTMS message serially.

The LTMS supports several time facilities accessible via the parallel microprocessor interface and dedicated pins. The time facilities can be controlled and observed via dedicated inputs and outputs, as well as via registers.

The Time Stamp facility allows the LTMS user to time-tag data. An external event causes the current ET counter value, plus the status of the LTMS and the rest of the time-distribution chain, to be stored in dedicated facility registers.

The Alarm Clock facility allows the LTMS user to generate an alarm or interrupt at a predetermined time instant.

The Stopwatch facility allows the LTMS user to measure time intervals or the number of events within a predetermined time interval.

The Pulse Generator facility allows the CTMS to generate locally pulses at a maximum 1 Hz rate.

The Waveform Generator facility allows the LTMS user or the CTMS to generate a periodic waveform. The waveform frequency and shape are programmable via the parallel microprocessor interface or the CTMS message respectively.

The pulse- and waveform generator facilities can be combined to form a programmable frequency generator that generates a periodic waveform in phase with the central ET reference.

The Extension Interface allows the LTMS user to implement external time facilities with a minimum of external hardware.

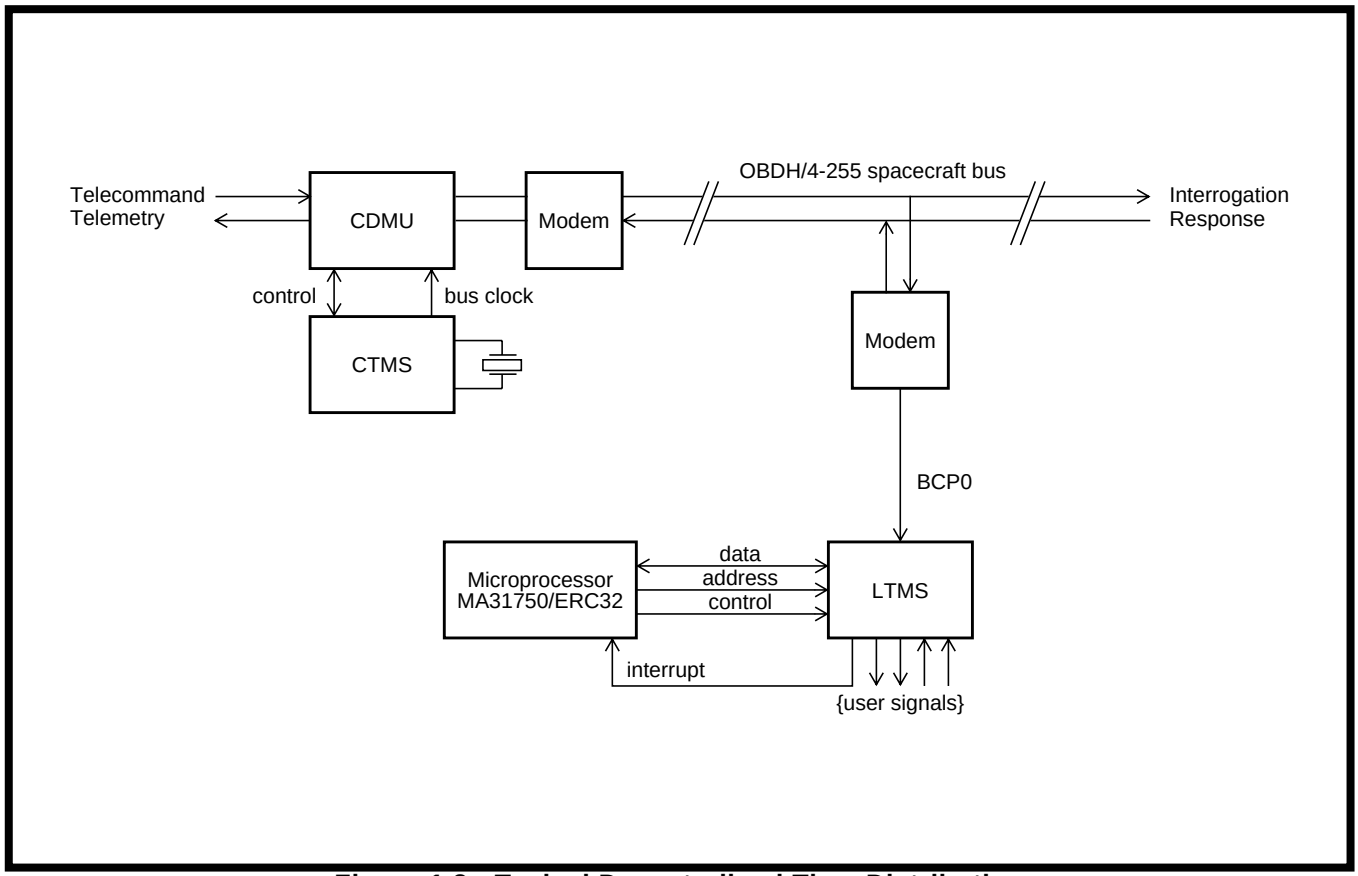


Figure 1.2 - Typical Decentralised Time Distribution

1.2 Elapsed Time Format

All time information sent to and provided by the LTMS is compliant with the CCSDS CUC format. The corresponding P-field the CTMS message time information is constant for the LTMS application, and is consequently not required to be transmitted. The corresponding T-field of time information comprises 4 coarse time octets (cf. figure 1.3).

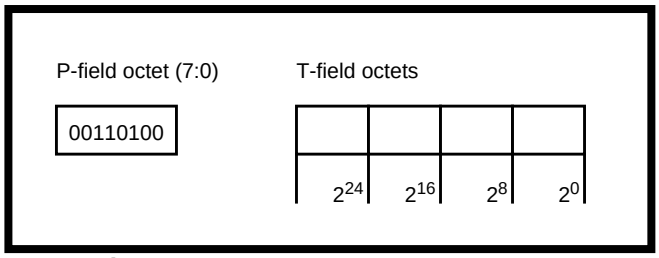


Figure 1.3 - CCSDS CUC ET Format

The LTMS ET counter is the local ET reference consisting of the ET Fine Counter (22 bits with weights less than 1 second) and the ET Coarse Counter (32 bits with weights greater or equal to 1 second and with a wrap-around period of 136 years).

The LTMS ET counter resolution depends on the operation selection. As a result, the least significant bit of the ET counter can have the weight 2^{-22}, 2^{-21}, 2^{-20} or 2^{-19}. For a bit with a defined weight, the position within an octet is constant and independent of the selected LTMS operation.

The LTMS stopwatch facility resolution depends on the mode selected. The weight of the least significant bit can be 2^{-24}, 2^{-22}, 2^{-21}, 2^{-20} or 2^{-19}. The most significant bit has the weight 2^{31}.

Some LTMS registers have bits that carry no information. When such registers are read, a zero is read for those bits. This applies to bits with weights smaller than the selected resolution, e.g. in the ET counter or in the stopwatch facility (cf. figure 1.4).

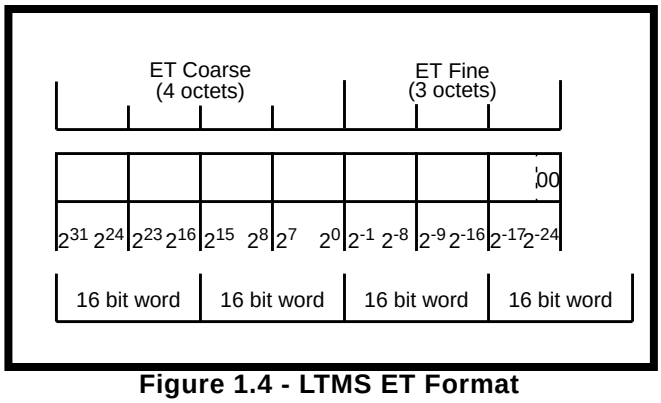


Figure 1.4 - LTMS ET Format

1.3 Operational Modes

Three basic operation modes are defined for the LTMS: Serial, Parallel and Stand Alone operation. For each mode, two sub-modes exist, depending on the use of an auxiliary 2^{24} Hz clock or not.

In all operational modes, the LTMS requires four kinds of information: the CTMS message, the synchronisation marker, the phase references and the ET counter clock. The information format differs according to the selected operation mode.

Table 1.1 shows the values of the configuration pins, the phase reference frequency, the operating frequency and the associated ET resolution.

1.4 Serial Operation

In serial operation the CTMS messages are provided serially. The phase references are either carried in the serial information (SERAUX) or are received on a dedicated input to drive directly the ET counter (SERBUS). If the phase reference frequency is too low compared to the expected time resolution, the LTMS should be used in SERAUX operation with an auxiliary clock to maintain its local ET counter between two successive synchronisations.

In SERAUX operation (cf. figure 1.5), the four kinds of information are provided to the LTMS as follows:

CTMS message:	Manchester coded on the **sin** input.
Synchronisation marker:	Falling **sin** edge before the first bit of the Manchester coded CTMS message.
Phase references:	Falling and rising **sin** edges on the Manchester coded input, occurring at an irregular frequency since dependent on the transmitted data stream.
ET counter clock:	From DPLL using external 2^{24} Hz clock on **xtal1**.

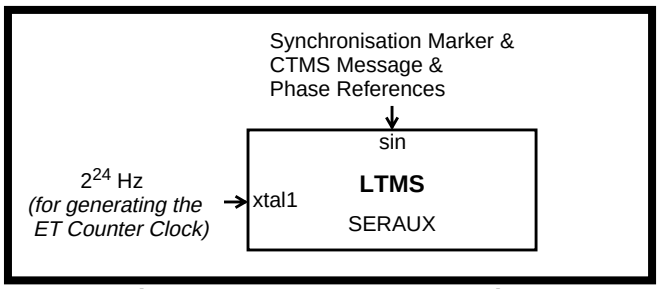


Figure 1.5 - SERAUX Operation

In SERBUS operation (cf. figure 1.6), the four kinds of information are provided to the LTMS as follows:

CTMS message: Manchester coded on the **sin** input.

Synchronisation marker: Falling **sin** edge before the first bit of the Manchester coded CTMS message.

Phase references: Rising **busclk** edges.

ET counter clock: Rising **busclk** edges.

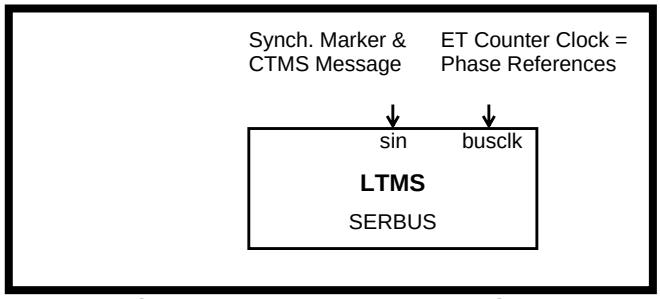


Figure 1.6 - SERBUS Operation

1.5 Parallel Operation

In parallel operation the CTMS messages are provided via the microprocessor interface. The LTMS uses the phase references from a dedicated input (**busclk**) to drive the ET counter (PARBUS).

If the phase reference frequency is too low compared to the expected time resolution, the LTMS should be used in PARAUX operation with an auxiliary clock to maintain the local ET counter between two successive synchronisations.

In PARAUX operation (cf. figure 1.7), the four kinds of information are provided to the LTMS as follows:

CTMS message: Via **data[15:0]** on parallel microprocessor interface.

Synchronisation marker: First rising **busclk** edge while **sin**=1.

Phase references: Rising **busclk** edges.

ET counter clock: From DPLL using external 2^{24} Hz clock on **xtal1**.

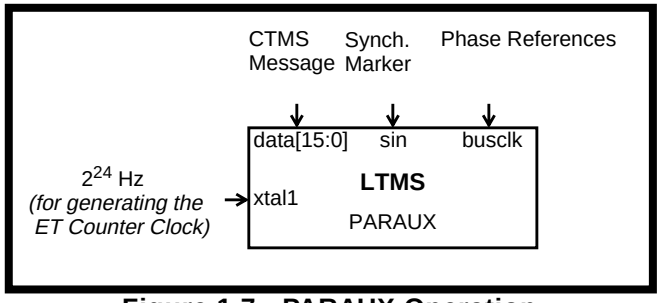


Figure 1.7 - PARAUX Operation

In PARBUS operation (cf. figure 1.8), the four kinds of information are provided to the LTMS as follows:

CTMS message: Via **data[15:0]** on parallel microprocessor interface.

Synchronisation marker: First rising **busclk** edge while **sin**=1.

Phase references: Rising **busclk** edges.

ET counter clock: Rising **busclk** edges.

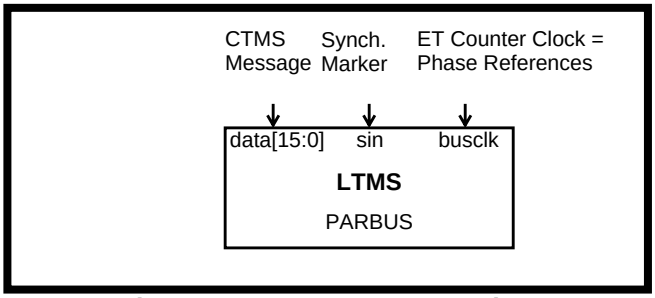


Figure 1.8 - PARBUS Operation

1.6 Stand Alone Operation

In stand alone operation, the LTMS maintains its own time reference and synchronisations are not required. However, they can be performed using the same protocol as used in parallel operation, except that they do not have to occur regularly.

The difference between parallel and stand alone operation is that timeouts that would lead to synchronisation errors in parallel operation are not flagged as errors in stand alone operation.

In SALAUX operation (cf. figure 1.9), the four kinds of information are provided to the LTMS as follows:

CTMS message: Optional, via **data[15:0]** on parallel microprocessor interface.

Synchronisation marker: Optional, first rising **busclk** edge while **sin**=1.

Phase references: Optional, rising **busclk** edges.

ET counter clock: From DPLL using external 2^{24} Hz clock on **xtal1**.

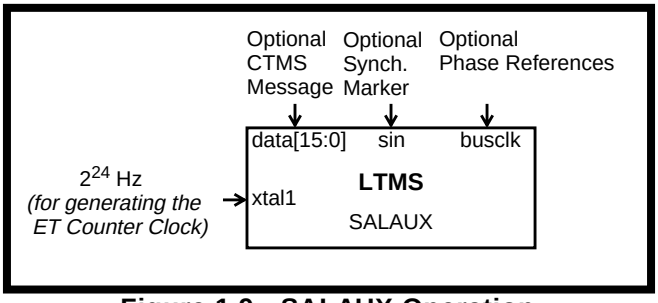


Figure 1.9 - SALAUX Operation

In SALBUS operation (cf. figure 1.10), the four kinds of information are provided to the LTMS as follows:

CTMS message: Optional, via **data[15:0]** on parallel microprocessor interface.

Synchronisation marker: Optional, first rising **busclk** edge while **sin**=1.

Phase references: Rising **busclk** edges.

ET counter clock: Rising **busclk** edges.

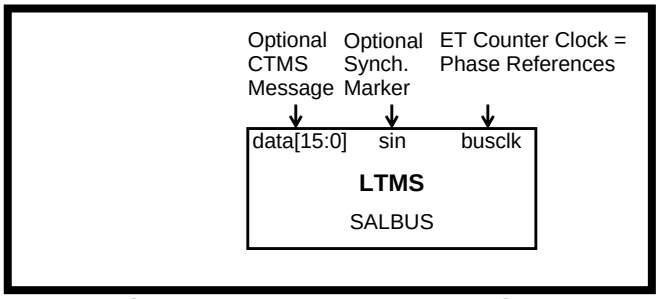


Figure 1.10 - SALBUS Operation

1.7 CTMS Message Reception

1.7.1 CTMS Message Format

The CTMS message format accepted by the LTMS is compliant with the Serial Time Distribution Protocol Extension defined for the ESA 4-255 Data Bus. The format (cf. figure 1.11) is mode independent.

The CTMS Status Field provides the LTMS with information about the CTMS message contents, the configuration of the rest of the time chain, and mission specific flags. The contents of the CTMS status field are defined in table 1.2.

The CTMS Coarse Time Field provides the LTMS with time synchronisation information. Since LTMS synchronisations are performed on integer amounts of seconds, it is not necessary to provide the LTMS with any sub-second time information.

Mode		ctmsg	ser	auxtal	clkf[2:0]	Phase Reference	Frequency	ET Resolution
SERIAL	**SERAUX**	1	1	1	x11	2^{22}/32 Hz	2^{24} Hz	2^{-22}
					x10	2^{21}/32 Hz	2^{24} Hz	2^{-22}
					x01	2^{20}/32 Hz	2^{24} Hz	2^{-22}
					x00	2^{19}/32 Hz	2^{24} Hz	2^{-22}
	SERBUS	1	1	0	x11	2^{22} Hz	2^{22} Hz	2^{-22}
					x10	2^{21} Hz	2^{21} Hz	2^{-21}
					x01	2^{20} Hz	2^{20} Hz	2^{-20}
					x00	2^{19} Hz	2^{19} Hz	2^{-19}
PARALLEL	**PARAUX**	1	0	1	011	2^{22} Hz	2^{24} Hz	2^{-22}
					010	2^{21} Hz	2^{24} Hz	2^{-22}
					001	2^{20} Hz	2^{24} Hz	2^{-22}
					000	2^{19} Hz	2^{24} Hz	2^{-22}
					111	2^{22}/1024 Hz	2^{24} Hz	2^{-22}
					110	2^{21}/1024 Hz	2^{24} Hz	2^{-22}
					101	2^{20}/1024 Hz	2^{24} Hz	2^{-22}
					100	2^{19}/1024 Hz	2^{24} Hz	2^{-22}
	PARBUS	1	0	0	x11	2^{22} Hz	2^{22} Hz	2^{-22}
					x10	2^{21} Hz	2^{21} Hz	2^{-21}
					x01	2^{20} Hz	2^{20} Hz	2^{-20}
					x00	2^{19} Hz	2^{19} Hz	2^{-19}
STANDALONE	**SALAUX**	0	x	1	x11	2^{22} Hz	2^{24} Hz	2^{-22}
					x10	2^{21} Hz	2^{24} Hz	2^{-22}
					x01	2^{20} Hz	2^{24} Hz	2^{-22}
					x00	2^{19} Hz	2^{24} Hz	2^{-22}
	SALBUS	0	x	0	x11	2^{22} Hz	2^{22} Hz	2^{-22}
					x10	2^{21} Hz	2^{21} Hz	2^{-21}
					x01	2^{20} Hz	2^{20} Hz	2^{-20}
					x00	2^{19} Hz	2^{19} Hz	2^{-19}

Table 1.1 - LTMS Configuration and ET Resolution

The CTMS Pulse Field is only transmitted in CTMS messages for which the Pulse Flag in the CTMS status field is set. This field is used by the pulse generator facility.

The CTMS Waveform Field is only transmitted in CTMS messages for which the Waveform Flag in the CTMS status field is set. This field is used by the waveform generator facility.

1.7.2 Serial Protocol

The serial protocol implemented for the CTMS message reception is compliant with the Serial Time Distribution Protocol Extension defined for the ESA 4-255 Data Bus (cf. figure 1.12). It is however possible to use the LTMS in systems not implementing the aforementioned data bus.

The LTMS receives every second a synchronisation marker and a subsequent CTMS message. Each octet is followed by a parity bit. The number of ones in each octet plus its parity bit has to be even. Bits used to fill up octets are also taken into account. The most significant bit of each octet is transmitted first.

After the CTMS message, an optional Mission Parameter Field can be sent to the LTMS. The mission parameter field also includes a parity bit after each octet.

The CTMS message, or the mission parameter field, is followed by a continuous stream of zeros, called the Inter-message Symbols, filling up the gap until the next synchronisation marker. The inter- message symbols comply by definition to even parity. It is possible that an incomplete octet of inter-message symbols is transmitted just before a synchronisation marker. The boundary between the optional mission parameter field and the inter-message symbols can be arbitrary with respect to LTMS operation.

The resulting data stream is received via the **sin** input and has to be Manchester encoded. The bits of the CTMS message, the mission parameter field, the parity, the inter-message symbols and the synchronisation marker are represented by the patterns shown in figure 1.13. The first bit sent after a synchronisation marker in serial operation has to be zero, else no synchronisation marker is detected and a Message Error occurs. On the detection of a synchronisation marker, the previously acquired

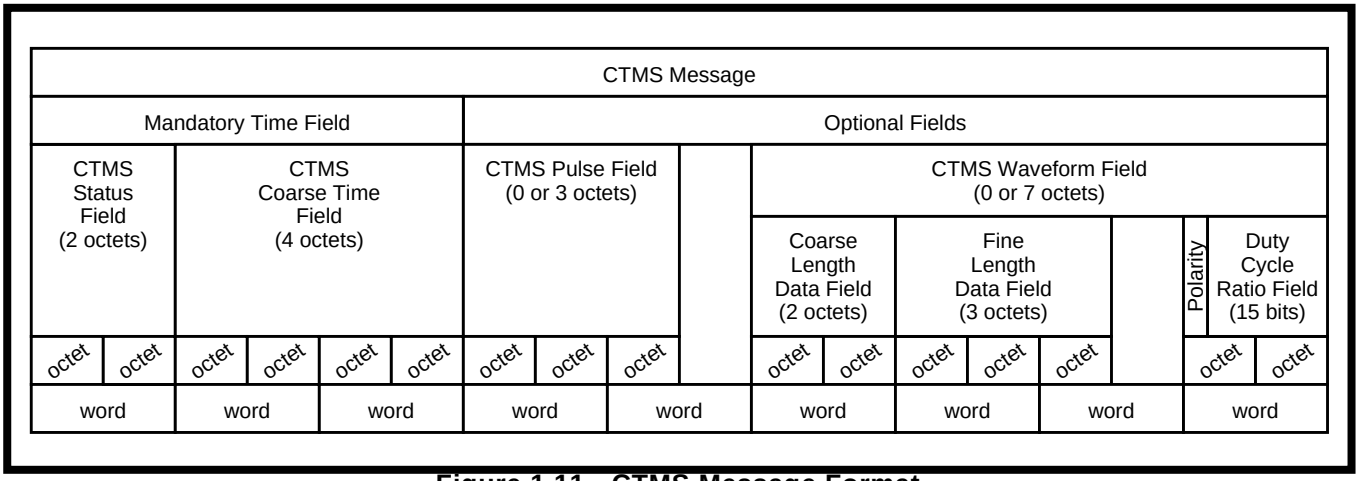


Figure 1.11 - CTMS Message Format

Bit	Name	Description
15	Zero Bit	Bit 15 always has to be zero.
14	Pulse Flag	Indicates that a CTMS Pulse Field is available in the CTMS message.
13	Waveform Flag	Indicates that a CTMS Waveform Field is available in the CTMS message.
12	Initialisation Flag	When set, an LTMS synchronisation will occur overriding timing checks, possibly inducing a time discontinuity.
11-8	Time Chain Flags	These bits can be used to transmit time chain related information to the user.
7-0	Mission Specific Flags	These bits can be used to transmit mission specific information to the user.

Table 1.2 - CTMS Message: CTMS Status Field

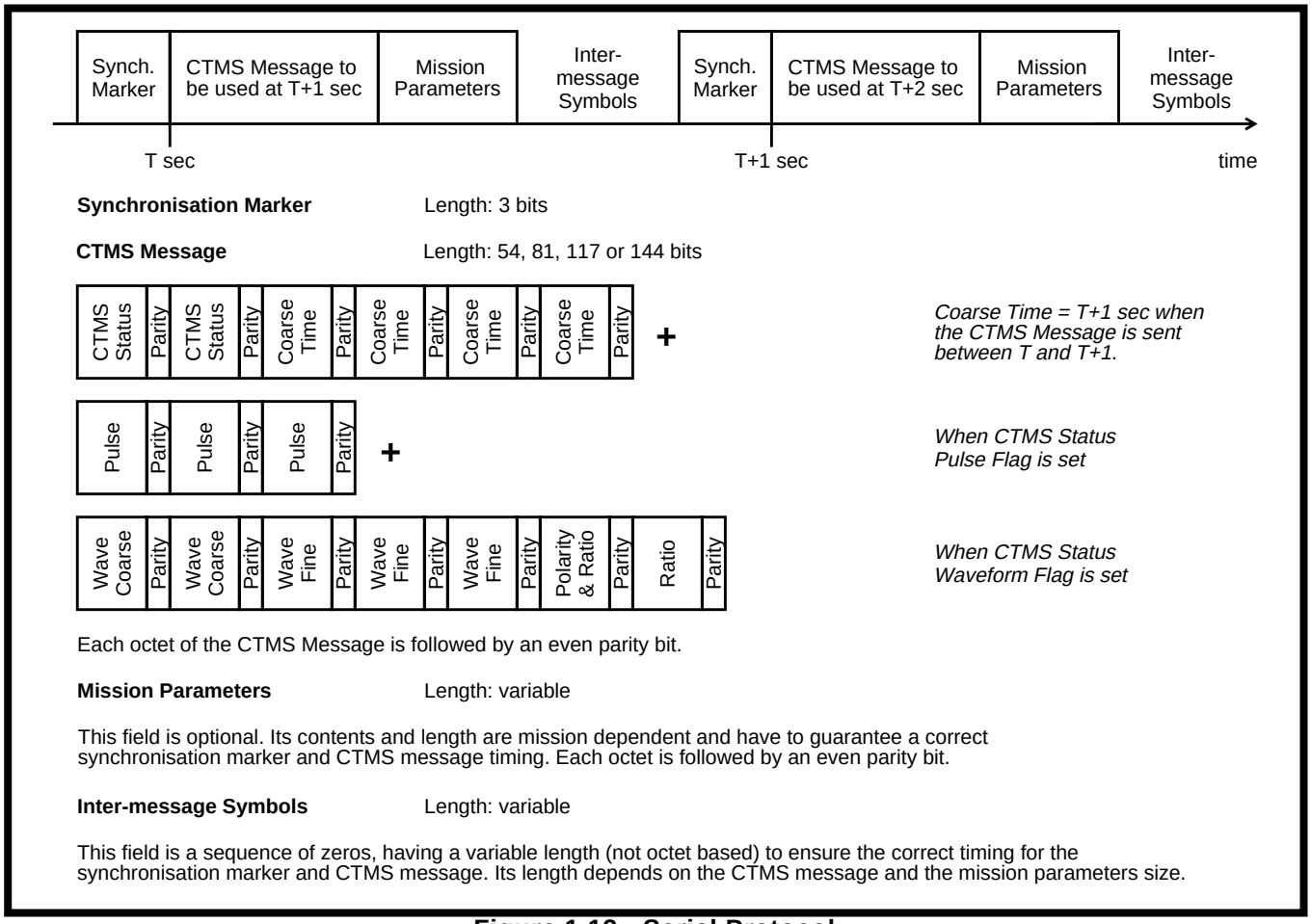


Figure 1.12 - Serial Protocol

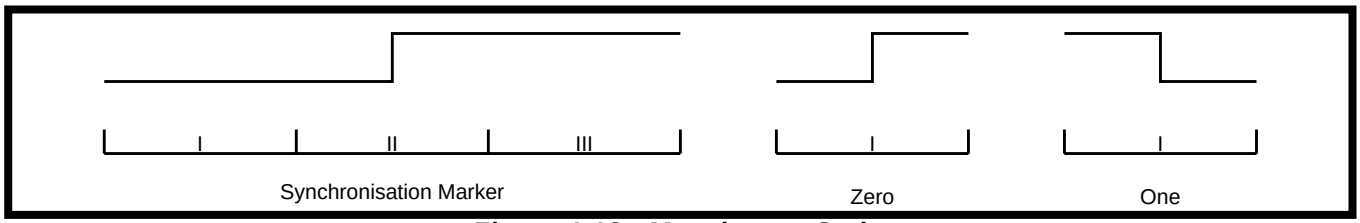


Figure 1.13 - Manchester Code

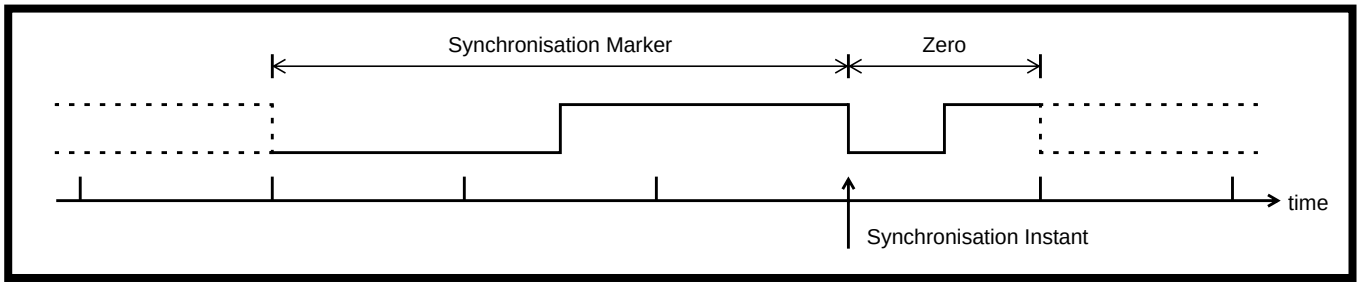


Figure 1.14 - Synchronisation Instant

CTMS message is used for LTMS synchronisation. The LTMS uses the falling edge at the end of the synchronisation marker as the reference point for LTMS synchronisation. This is the Synchronisation Instant at which the ET counter should become equal to an integer number of seconds (cf. figure 1.14).

In SERAUX operation, the rising and falling edges of the **sin** input are used as phase references.

In SERBUS operation, the rising edges of the **busclk** input are used as phase references. The **sin** input is sampled on the rising edges of the **busclk** input.

1.7.3 Parallel and Stand Alone Protocol

In parallel and stand alone operation, CTMS messages are written to the LTMS using the parallel microprocessor interface. The writing has to be performed in the following order:

1. CTMS Status and Coarse Time Fields
2. CTMS Pulse Field (optional)
3. CTMS Waveform Field (optional)

This sequence allows the LTMS to know whether the last two fields are present in a CTMS message and which word is therefore considered to be the last (i.e. address 3, 5 or 9). The optional fields will only be used by the LTMS if the corresponding flag is set in the CTMS status field. Figure 1.15 depicts the CTMS message words and the moment of their occurrence.

If the pulse flag or waveform flag in the CTMS status field are zero, the corresponding fields can still be written (dummy words). The LTMS will neither use these fields nor make them available via the extension interface. These fields cannot be read back because not being part of the CTMS message.

On the occurrence of a synchronisation marker on the **sin** input, the previously acquired CTMS message is used for LTMS synchronisation, provided that all conditions are met (cf. section 1.8).

In case of errors during the CTMS message transmission, a new attempt is possible by writing a new sequence. The new CTMS message supersedes the previous one. Only the last message before a synchronisation marker is checked for message errors. If the LTMS does not receive a valid CTMS message before the occurrence of a synchronisation marker, it flags this as a message error in the LTMS status register.

In parallel and stand alone operation, the **window** output indicates when the LTMS expects a new CTMS message. A CTMS message has to be provided while **window**=1, giving the LTMS enough time to check the CTMS message, and to shift it out via the extension interface, before it expects the next synchronisation marker.

If the **window** output goes low between the detection of the first (1) and the last (3, 5 or 9) address while writing a CTMS message, it is considered a window violation. This violation is flagged with the LTMS status register *windout* flag.

If a CTMS message is written to the LTMS while **window**=0, the risk exists that the LTMS will not have completely shifted out the CTMS message on the extension interface before the **extetld** signal will qualify the information. This could result in loading an erroneous value into an external ET counter.

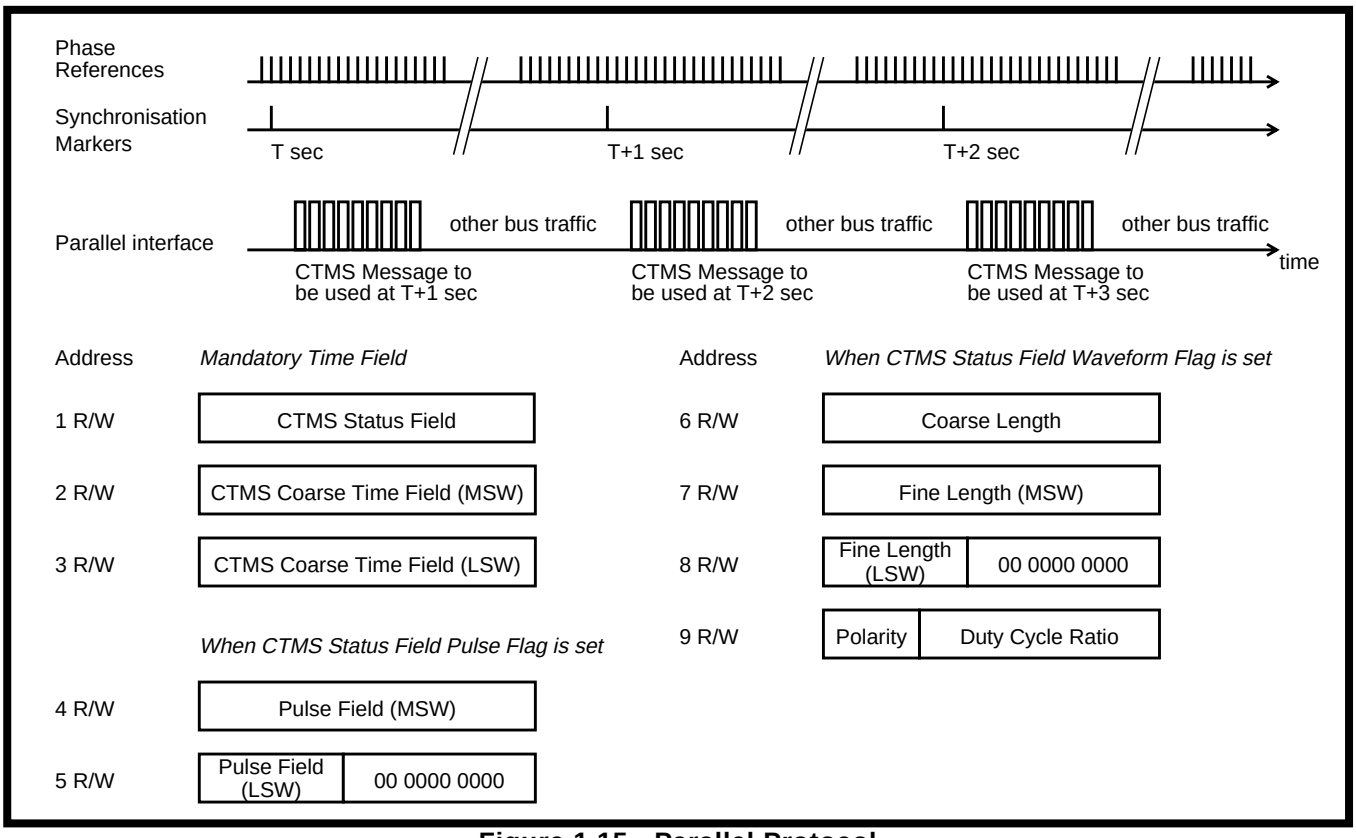


Figure 1.15 - Parallel Protocol

1.7.4 CTMS Message Register

In serial operation, the CTMS message information is decoded and stripped of its parity bits in the Manchester Decoder and is loaded octet-wise into the CTMS message register. Incorrect partial updates can occur if message errors are detected. The CTMS message register may only be read when **window**=1, to avoid accessing the register while its contents are updated, resulting in reading parts of two consecutive CTMS messages. In case this happens, a window violation is flagged with the LTMS status register *windout* flag.

In parallel and stand alone operation, only complete CTMS messages are loaded into the CTMS message register. Until a new and complete CTMS message has been written, the information from the previous complete CTMS message is available in the CTMS message register. Hence, one has to wait after writing the last address of the CTMS message until it has been processed by the LTMS before reading the CTMS message register. A violation of this rule results in possibly reading incorrect message information. The LTMS uses the **window** output to inform when it expects the next CTMS message. Each time the **window** output becomes one, the CTMS message register should be updated.

In stand alone operation, CTMS messages can be sent and LTMS synchronisations can be performed independently of the **window** output. The timing for sending CTMS messages and synchronisation markers is dependent on whether an external ET counter is making use of the information on the extension interface or not (cf. section 1.7.6).

In all operational modes, the CTMS message register can be read via the parallel microprocessor interface at addresses 1 to 9. It always has to be read as a whole and in the sequence defined hereafter. Address 1 has to be read first. Reading address 9 is recognised as the end of the read operation of the CTMS message register. No particular read sequence is required for the addresses in-between. If the **window** output has a 1 to 0 transition between the detection of a read of the address 1 and that of address 9, a window violation is flagged with the LTMS status register *windout* flag.

The **window** output is high during approximately 900 ms each second, giving ample margins to both the user and the LTMS under nominal conditions. The **window** information is also available via the LTMS control register *windowbit* flag.

In all operation modes it is possible to perform a full read back of valid CTMS messages sent to the LTMS. When no errors occur, the CTMS Message Register contains all fields available in the last valid CTMS message. Message errors can temporarily invalidate the values in the CTMS message register.

1.7.5 CTMS Message Errors

A LTMS Synchronisation can only occur after the reception of a valid CTMS message. Format violations during the acquisition of a CTMS message are called Message Errors. A valid CTMS message is received when none of the following message errors occur:

In serial operation:

- parity error in the CTMS message, the mission parameters or the inter-message symbols;
- Manchester error, excluding the synchronisation marker but including errors in bit 15 of the CTMS message status field;
- a synchronisation marker occurrence before the detection of the end of a CTMS message.

In parallel operation:

- a synchronisation marker occurrence before a complete CTMS message has been detected after the previous synchronisation marker;
- bit 15 of the CTMS message status field is not equal to zero.

In stand alone operation:

- a synchronisation marker occurrence before a complete CTMS message has been written;
- bit 15 in the CTMS message status field is not equal to zero.

At the occurrence of a synchronisation marker after a message error, the LTMS:

- disables the current synchronisation process;
- increments the Error Counter in the LTMS status register (cf. section 1.8.6);
- sets error code 01 in the LTMS status register;
- waits for the next CTMS message.

A special error case is the Synchronisation Marker Timeout (cf. section 1.8.4). This error results in immediate LTMS action, instead of waiting until the next synchronisation marker. At the occurrence of a synchronisation maker timeout, the LTMS:

- increments the error counter, if not saturated;
- sets error code 01 in the LTMS status register;
- sets the LTMS status register *synchto* flag;
- waits for the next CTMS message.

Synchronisation markers that lead to message errors are not considered for LTMS synchronisation purposes. In serial operation, each synchronisation marker is regarded as the beginning of a new message. In parallel and stand alone operation, there always has to be a CTMS message between two synchronisation markers.

After a synchronisation timeout error has occurred, a new CTMS message is expected. If not acquired, the subsequent synchronisation marker will result in another message error. Note that this is always the case in serial operation, because a new CTMS message is started with a synchronisation marker.

1.7.6 Extension Interface

The Extension Interface allows to build and maintain external copies of the internal LTMS ET counter. This external counter can be used as an additional ET reference to be used by external time facilities customised to the users needs, or to perform additional error checks. The extension interface shifts out only valid CTMS messages in serial format. The format of the CTMS messages shifted out via the extension interface does not depend on the LTMS operation selection.

In parallel and stand alone operation, the shift process is initiated only when the update of the CTMS message register contents has been completed. Correct operation of the extension interface is only guaranteed when the CTMS message is written while **window**=1.

In parallel and stand alone operation, CTMS messages for which bit 15 in the CTMS status field is not equal to zero (considered being message errors) are not shifted out on the extension interface. In case of message errors or writing of a new CTMS message, the ongoing transmission is immediately aborted and re-initialised so that a new CTMS message can be transferred when acquired.

In serial operation, the CTMS message is immediately available from the Manchester decoder, but with the parity bit stripped. The extension interface also delivers the optional mission parameters and the inter-message symbols.

The CTMS message information is available on the **extdata** output on the rising edges of the external ET counter clock output (**extcount**) when the external shift enable (**extshten**) is set (cf. figure 1.16). The **extdemux** outputs inform about the nature of the data shifted on the **extdata** output (CTMS status field, coarse time field, pulse or waveform fields, mission parameters or inter-message symbols) and can be used for controlling an external demultiplexer (cf. figure 1.17 and table 1.3).

Figure 1.16 shows the extension interface protocol. In serial operation the same basic format applies with two differences. The **extshten** output is not necessarily asserted constantly between the beginning and the end of the data transfer. Consequently, values on **extdata** that are available at rising **extcount** edges are only valid while **extshten**=1. The values of **extdemux** also differ, as seen in figure 1.17.

When **extetld**=1 at a rising **extcount** edge, the CTMS message information that was shifted via the extension interface can be used to perform synchronisations of an external ET counter. Because of internal delays in the LTMS, the ET synchronisation happens at an integer second plus one ET counter clock tick. Hence, the least

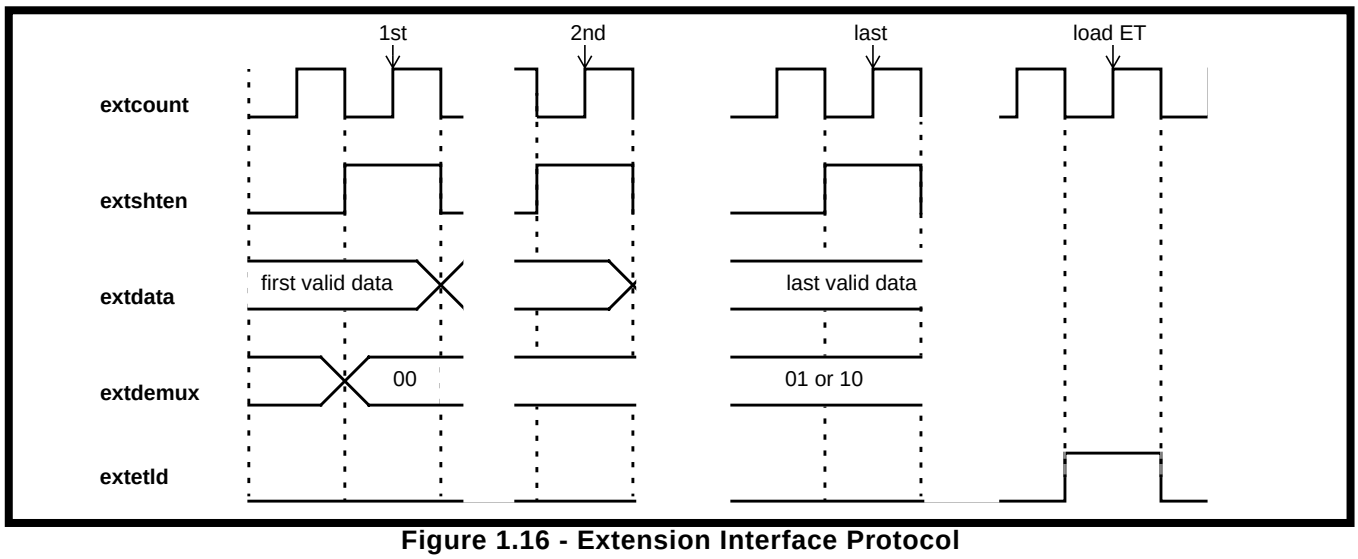


Figure 1.16 - Extension Interface Protocol

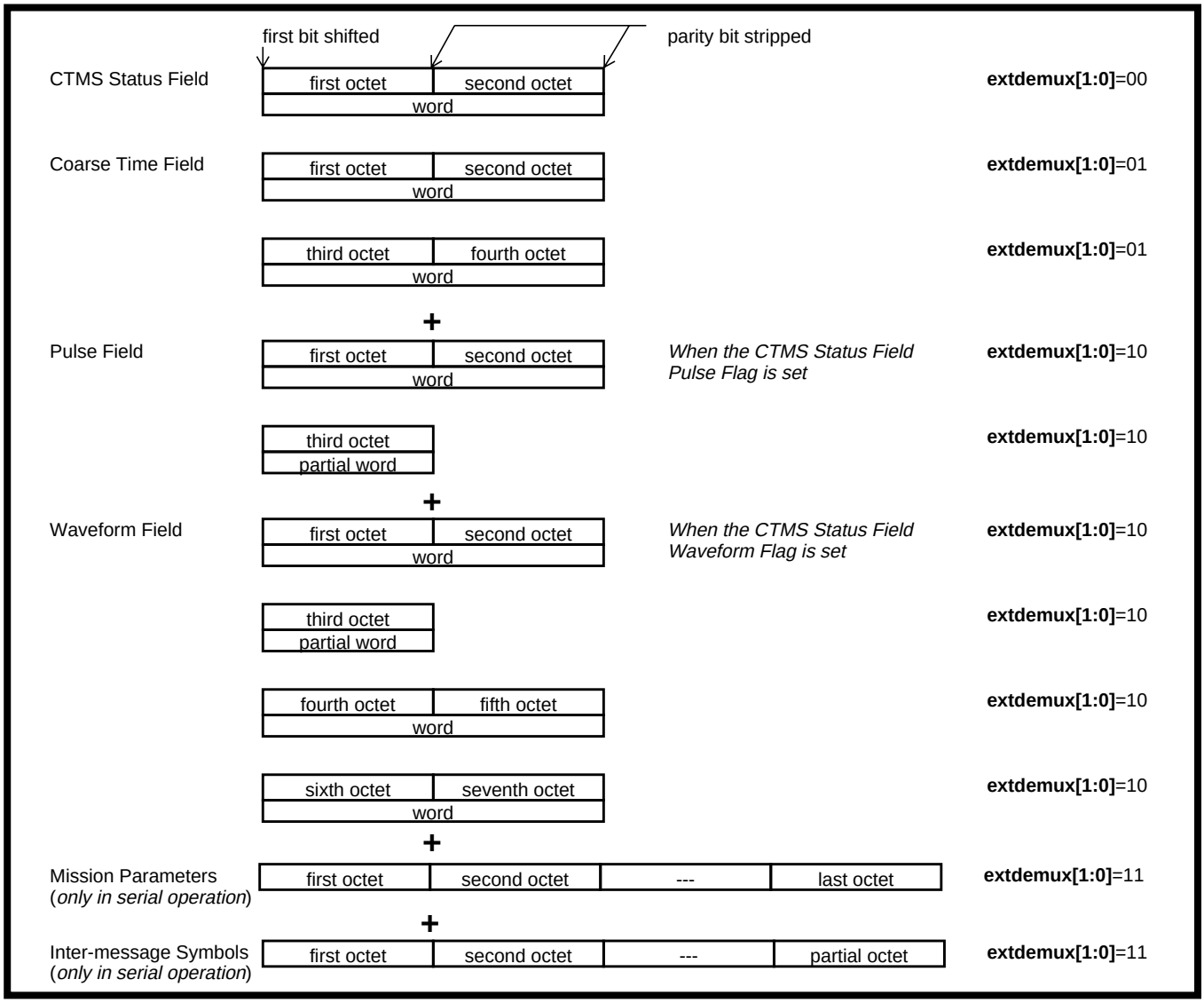


Figure 1.17 - Extension Interface Output Protocol

extdemux[1:0]	data currently shifted on the extdata output
00	CTMS Message - Status
01	CTMS Message - ET Coarse
10	CTMS Message - Pulse, Waveform Length, Polarity, Duty Cycle Ratio
11	Mission Parameters or Inter-message Symbols (serial operation only)

Table 1.3 - Extension Interface Demultiplexer

significant bit of the external ET counter corresponding to the LTMS resolution (cf. table 1.1) should be preset to 1. All other external ET counter fine time bits should be cleared to 0.

The **exterin** input can be used to flag errors detected by checks performed outside the LTMS. Asserting **exterin** will assert the LTMS status register *exterror* flag and hence disable the LTMS synchronisation. If no external checks are performed, the **exterin** input should be tied to zero. Acquisition of consecutive CTMS messages is not disabled while **exterin** or the *exterror* flag are set.

As the user also has the possibility to report results of external checks via the LTMS control register *exterinbit*, the *exterror* flag will be cleared as soon as both the **exterin** input and the *exterinbit* flag are cleared as a result of a CTMS message received that contained no errors.

1.8 LTMS Synchronisation

1.8.1 LTMS Synchronisation Conditions

LTMS Synchronisations are executed on the detection of synchronisation markers, only if all the conditions in one of the following sets are met.

After an LTMS reset:
- the first valid CTMS message has been acquired after a reset.

During normal LTMS operation:
- a valid CTMS message has been acquired;
- $ET_{coarse\text{-}message} = ET_{coarse\text{-}counter} + 1$ second (cf. section 1.8.2);
- the synchronisation marker comes within the ET threshold window (cf. section 1.8.3);
- the LTMS status register *exterror* flag is zero.

During error accumulation:
- a valid CTMS message has been acquired;
- the Error Counter has reached its Go-threshold: a build up of errors has happened during the acquisition and synchronisation of previous CTMS messages (cf. section 1.8.6);
- the LTMS status register *exterror* flag is zero.

At ET counter initialisation:
- a valid CTMS message has been acquired;
- the CTMS message *initialisation* flag is set.

At ET counter synchronisation in stand alone mode:
- the LTMS is in stand alone operation;
- a valid CTMS message has been acquired;
- the LTMS status register *exterror* flag is zero.

If one set of conditions is met at the synchronisation instant, an LTMS synchronisation is performed. It consists of all of the following actions:
- $ET_{coarse\text{-}message}$ is loaded in $ET_{coarse\text{-}counter}$;
- the lowest resolution bit of the $ET_{fine\text{-}counter}$ is set and all other bits of the counter are cleared;
- the CTMS message flags are loaded in the CTMS status register;
- the error counter is decremented (cf. section 1.8.6);
- the LTMS status register *wasfree* flag is cleared;
- the Error Code in the LTMS status register is set to 00 or 11 (cf. table 1.10).

1.8.2 ET Coarse Timeout

As soon as a complete and valid CTMS message has been received, the LTMS compares the ET coarse information in the CTMS message with the current ET coarse value of the ET counter. The value in the CTMS message should be the value in the counter incremented by one. If this is not the case there is an ET coarse timeout.

1.8.3 ET Threshold Timeout and Monotony

The ET Threshold selection defines the least significant bit of the ET guaranteed to be monotonous, provided the threshold limit is met.

In serial and parallel operation, the LTMS synchronisation has to occur within an ET threshold window centred around the moment on which the ET counter fine time would become zero again: i.e. the next expected synchronisation instant. Different widths for the ET threshold window can be selected using the **etthr** inputs (cf. table 1.4). In this table, the column with typical time values assumes an ET counter clock period of approximately 0.25 µs.

Since LTMS synchronisation can happen anywhere in the ET threshold window, time discontinuities are possible in the ET counter. However, from a certain bit position upwards, the time count will be monotonous. The part of the ET counter for which time is monotonous is called the monotonous ET counter. The least significant position of the monotonous ET counter can be found in table 1.4. This monotony is guaranteed in serial and parallel operation as long as no forced LTMS synchronisation occur due to error accumulation or ET counter initialisation (cf. section 1.8.6).

In stand alone operation, time monotony is not guaranteed and time discontinuities are possible at the LTMS synchronisation instants. It is possible to select an ET threshold value even though a timeout is not considered to be an error. The timeouts are flagged in the LTMS status register to indicate that the local ET reference underwent a discontinuity, but have no further impact on the LTMS.

1.8.4 Synchronisation Marker Timeout

In serial and parallel operation, the first synchronisation marker after external reset should occur within two seconds after the rising **exrst_bar** edge. From then on, new synchronisation markers should occur one second after a previous one.

In serial or parallel operation, the LTMS only detects a synchronisation timeout when a synchronisation marker does not appear within 1.5 to 2 seconds after an external reset or the previous synchronisation marker. The actual values and the uncertainty are a result of the fact that the timeout detection is using the ET counter. The timeouts are flagged with the LTMS status register *synchto* flag. The flag is reset by the next correctly timed synchronisation marker.

In stand alone operation, no synchronisation rate is imposed and not time out checks are performed.

1.8.5 Synchronisation Errors

Timeouts at the moment the LTMS synchronisation should occur or flagging of an external error (cf. section 1.7.6) are called Synchronisation Errors, being defined as:

In serial and parallel operation:

- an ET coarse timeout occurs while neither the *initialisation* flag in the CTMS message was set, nor the go-threshold was reached;
- an ET threshold timeout occurs while neither the *initialisation* flag in the CTMS message was set, nor the go-threshold was reached;
- the LTMS control register *exterror* flag is set while the CTMS message *initialisation* flag was not set.

In stand alone operation:

- the LTMS control register *exterror* flag is set while the CTMS message *initialisation* flag was not set.

At the occurrence of a synchronisation error, the LTMS immediately:

- disables the current synchronisation process;
- increments the error counter;
- sets error code 10 in the LTMS status register;
- updates the LTMS status register *etcto* flag in case the ET threshold is exceeded;
- loads the CTMS message status flags into the CTMS status register;
- waits for the next CTMS message.

In stand alone operation, only external errors are considered to be synchronisation errors. The ET threshold timeout and ET coarse timeout conditions are flagged in the LTMS status register to provide the user with time discontinuity warnings, but do not impede the LTMS synchronisation.

1.8.6 Error Counter

The Error Counter (i.e. *errcount* flags in the LTMS status register) keeps track of message error and synchronisation error build-up. Each time an LTMS synchronisation is performed while the error counter is not zero, the error counter is decremented. It is incremented once per message with message errors and on synchronisation errors. When several errors are detected within a CTMS message, the error counter is incremented only once. A synchronisation marker following a CTMS message with a message error is not considered for LTMS synchronisation. It will not generate an error, nor will it increment the error counter.

etthr[1:0]	ET Resolution	Least significant monotonous bit	ET counter clock periods	Typical time
00	2^{-22}	2^{-20}	±4	±1 µs
01	2^{-22}	2^{-16}	±64	±15.2 µs
10	2^{-22}	2^{-12}	±1024	±244 µs
11	2^{-22}	2^{-8}	±16384	±3.9 ms
00	2^{-21}	2^{-19}	±4	±1.9 µs
01	2^{-21}	2^{-15}	±64	±30.5 µs
10	2^{-21}	2^{-11}	±1024	±488 µs
11	2^{-21}	2^{-7}	±16384	±7.8 ms
00	2^{-20}	2^{-18}	±4	±3.8 µs
01	2^{-20}	2^{-14}	±64	±61 µs
10	2^{-20}	2^{-10}	±1024	±0.97 ms
11	2^{-20}	2^{-6}	±16384	±15.6 ms
00	2^{-19}	2^{-17}	±4	±7.6 µs
01	2^{-19}	2^{-13}	±64	±122 µs
10	2^{-19}	2^{-9}	±1024	±1.95 ms
11	2^{-19}	2^{-5}	±16384	±31.2 ms

Table 1.4 - ET Threshold Values and Monotony

In case of errors in successive CTMS messages, the error counter is incremented, saturating at a maximum value, called the Go-threshold. The go-threshold can be programmed to 0, 3, 7 and Infinite via the **gothr** inputs (cf. table 1.5). In serial and parallel operation, when the go-threshold is reached, a forced synchronisation will occur if no message errors and no external errors were detected. If the go-threshold is set to 0, no matter what the outcome of the ET threshold check is, updates are always performed on a synchronisation marker as long as no message error is detected and the *exterror* flag is zero. If the go-threshold is set to 0, the error counter invariably stays at zero.

gothr[1:0]	Go-threshold
00	0
01	3
10	7
11	Infinite

Table 1.5 - Go-threshold values

When the go-threshold is set to 3 (respectively 7), LTMS synchronisations are performed, regardless of any synchronisation errors, after an accumulation of 3 (7) errors in the error counter, i.e. the ET counter is loaded with the fourth (eighth) errorless CTMS message, when the *exterror* flag is zero. Reaching the go-threshold can happen either due to 3 (7) consecutive errors but also following a series of increases and decreases of the error counter eventually resulting in the error counter reaching its maximum value.

The go-threshold Infinity setting means that LTMS synchronisations are not performed as long as the ET threshold is exceeded while the *initialisation* flag of the CTMS message preceding the synchronisation marker is zero. If the ET threshold is exceeded, the LTMS synchronisation is performed only if the *initialisation* flag is set. For this go-threshold selection, the error counter saturates at 7. The selection has no impact in stand alone operation.

1.8.7 Time Validity

A summary of the status and error information in the LTMS status register is provided on the **tvld** output and in the LTMS control register *tvldbit* flag, which are set when the LTMS status register holds zeros on bits 3 to 7 and 9 to 10. The values of the other status flags have no impact.

To assess the time validity and the overall results of the internal LTMS error checks, the user can look at the **tvld** output or at the LTMS control register *tvldbit* flag. The user can also read the LTMS status register for further error analysis. The CTMS status register provides additional time validity information through its time chain flags.

1.9 Phase Synchronisation

1.9.1 DPLL and Phase References

If high resolution has to be obtained, or when a free-wheeling capability is required in case of phase reference irregularities, the LTMS can be used with an external 2^{24} Hz clock or crystal (cf. figure 1.18). An internal Digital Phase Locked Loop (DPLL) uses this clock to generate a 2^{22} Hz clock for the ET counter. The phase of the ET counter clock is adjusted with respect to the incoming phase references (cf. table 1.1). In case the internal DPLL drift and jitter criteria are insufficient, an external DPLL can be used instead. The interface between the LTMS and an external DPLL is shown in figure 1.18.

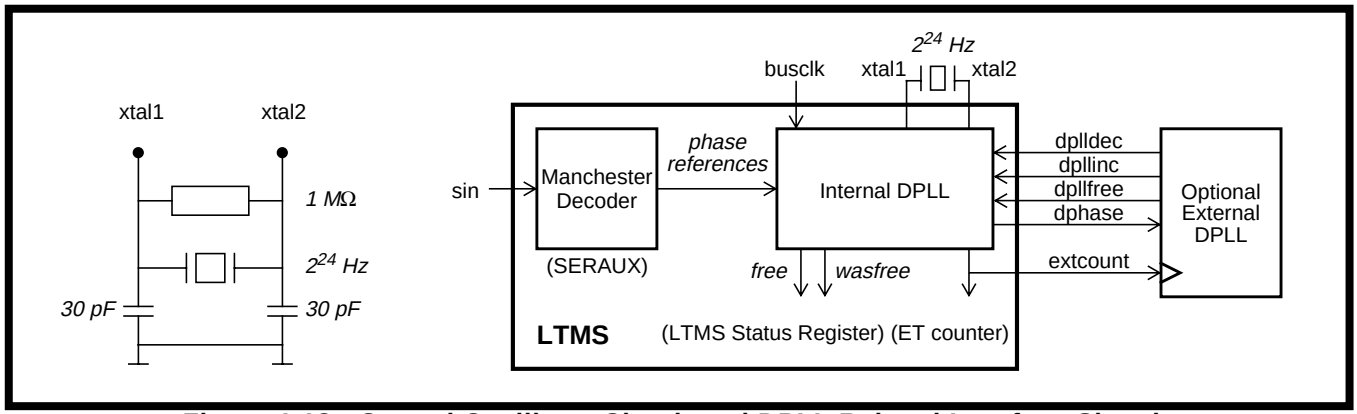


Figure 1.18 - Crystal Oscillator Circuit and DPLL Related Interface Signals

In case of phase reference errors, the internal DPLL starts free-wheeling and warns that drift may have occurred via the LTMS status register *free* and *wasfree* flags.

A phase reference error is defined as:

- a phase reference is missing, or
- the phase reference drift and/or jitter is too large (cf. section 1.9.5).

If the bus clock can be directly used by the LTMS without phase adjustment, the LTMS DPLL and its flags are forced to 0 and no auxiliary 2^{24} Hz clock signal or crystal oscillator is required. This is achieved by using the LTMS in SERBUS, PARBUS or SALBUS operation.

1.9.2 DPLL Choice

The LTMS provides an internal DPLL, but can also be operated in connection with an external DPLL in SERAUX, PARAUX and SALAUX operation. The values of the **dplldec** and **dpllinc** inputs define the DPLL choice. The inputs are sampled after a rising **exrst_bar** edge, either on the first rising **busclk** edge when **auxtal**=0, or on the first rising **xtal1** edge when **auxtal**=1 (cf. table 1.6).

1.9.3 Internal DPLL

The operation sequence below applies when the LTMS uses its internal DPLL. The sequence starts as soon as the LTMS has been reset or a phase reference error has been detected:

- The internal DPLL starts free-wheeling. The *free* and *wasfree* flags are set. Free-wheeling does not prevent CTMS message acquisition.
- Once free-wheeling, the internal DPLL immediately attempts to lock on to the phase references. No phase correction is applied to the LTMS ET counter clock. Valid CTMS messages are used for LTMS synchronisation. Due to clock drifts, the rate of rejected messages may increase when free-wheeling.
- If the internal DPLL has succeeded to lock onto three consecutive phase references, the DPLL locks its output on the phase references (again). The *free* flag is immediately cleared.
- The *wasfree* flag is cleared at the next LTMS synchronisation.

The regenerated internal phase reference is compared to the incoming phase references. The internal DPLL is able to add/subtract one 2^{24} Hz clock period per phase reference period to/from the regenerated internal phase reference. At the level of a phase reference period, contribution of drift and jitter effects can not be differentiated; only the maximum edge deviation, i.e. maximum drift plus maximum jitter is relevant.

1.9.4 External DPLL Interface

The use of an internal DPLL imposes constraints on parameters as drift and jitter on phase references and crystal frequency accuracy (cf. section 1.9.5). If the constraints are too tight for use with an internal DPLL an external DPLL can be used. This in turn means that the user becomes responsible for the synchronisation of the LTMS to the external phase references. The optional external DPLL uses then the LTMS DPLL as a slave.

The external DPLL should provide phase correction commands to the LTMS in order to minimise the phase error between external phase references and the ET counter phase as much as possible. These commands are provided via **dplldec** and **dpllinc**. In order to perform phase adjustment, the external DPLL should take into account the following signals:

- **dphase**: provides each phase reference detected by the LTMS to the user;
- **extetld**: informs the user when an LTMS synchronisation is being performed;
- **extcount**: clocks.

The **dpllfree** input controls the internal LTMS flags *free* and *wasfree*. The *free* flag is high as long as

dplldec	dpllinc	External DPLL available	extcount period (expressed in xtal1 periods)	Comments
0	0	yes	4	no correction
0	1	yes	3	speed up
1	0	yes	5	slow down
1	1	no	3, 4 or 5	LTMS DPLL only

Table 1.6 - The use of dplldec and dpllinc

dpllfree=1. The *wasfree* flag is set as soon as the **dpllfree** input is detected to be high and is cleared at the next LTMS synchronisation when **dpllfree**=0. When **dplldec** and **dpllinc** are both equal to 1 during reset (cf. table 3.5), the LTMS uses its internal DPLL. In the three other cases, the external DPLL is selected. In the latter case **dplldec** and **dpllinc** are used for adapting the period of **extcount,** which is the clock driving the LTMS ET counter (cf. table 1.6).

Detailed information on the generation of phase correction commands can be found in Appendix B.

1.9.5 Crystal Requirements

The requirements for the optional crystal used with the LTMS depend on the operational mode and which DPLL is used. These definitions are used:

F_{xtal1}	ideal 2^{24} Hz crystal frequency	(Hz)
T_{xtal}	ideal 2^{24} Hz crystal period	(s)
N	number of ideal crystal periods in an ideal phase reference period	
j	jitter on the Phase Reference	(ns)
j_n	normalised jitter: $\lvert j/T_{xtal1} \rvert$	
Δ_x	deviation on crystal frequency	(ppm)
Δ_ϕ	drift in phase reference frequency	(ppm)

For the internal DPLL to be able to lock, the following relationship must be respected:

$$\Delta_x\, 10^{-6} < \min\left(\frac{(1-\Delta_\Phi 10^{-6})(1-2j_n) - N\Delta_\Phi 10^{-6}}{N+2j_n(1-\Delta_\Phi 10^{-6})}, \frac{(1+\Delta_\Phi 10^{-6})(1-2j_n) - N\Delta_\Phi 10^{-6}}{N-2j_n(1+\Delta_\Phi 10^{-6})}\right)$$

The LTMS is designed to operate at a maximum crystal frequency of 18.63 MHz (11.11% faster than 2^{24} Hz). This means an absolute upper limit for Δ_x of 111111 ppm. The following absolute maximum values for Δ_x, Δ_ϕ and j_n can be derived:

Δ_x max 10^{-6}	=	$\min(\frac{1}{N}, 111111)$	for j_n, $\Delta_\phi = 0$
j_n max	=	0.5	for Δ_x, $\Delta_\phi = 0$
Δ_ϕ max 10^{-6}	=	$\frac{1}{N+1}$	for Δ_x, $j_n = 0$

Table 1.5 lists maximum values for Δ_x, Δ_ϕ, j_n and N. For stand alone operation, the listed values are only correct if no phase references are missing. If e.g. half of the nominal phase frequency rate is applied to the LTMS instead of the nominal phase frequency rate, the parameter N must be doubled. For serial operation the N value corresponding to the nominal phase reference frequency is multiplied by 3 because up to 2 phase references may be missing.

If an external DPLL is used, the crystal requirements depend on the characteristics of that DPLL. The calculations involve additional parameters, such as:

- the clock frequency of the external DPLL and its accuracy;
- uncertainties due to synchronisation of the **dplldec** and **dpllinc** inputs with the clock driving the LTMS et counter;
- the maximum allowed deviation of **dphase** versus ET counter phase (maximum number of corrections per phase reference, to be performed on **extcount**).

Operation	clkf[2:0]	j_n max	Δ_x max	Δ_ϕ max	Phase Reference	N
SERAUX	x11	0.5	2604	2597	2^{22}/32 Hz	384
	x10	0.5	1302	1300	2^{21}/32 Hz	768
	x01	0.5	651	651	2^{20}/32 Hz	1536
	x00	0.5	326	325	2^{19}/32 Hz	3072
PARAUX	011	0.5	111111	200000	2^{22} Hz	4
	010	0.5	111111	111111	2^{21} Hz	8
	001	0.5	62500	58824	2^{20} Hz	16
	000	0.5	31250	30303	2^{19} Hz	32
	111	0.5	244	244	2^{22}/1024 Hz	4096
	110	0.5	122	122	2^{21}/1024 Hz	8192
	101	0.5	61	61	2^{20}/1024 Hz	16384
	100	0.5	31	31	2^{19}/1024 Hz	32768
SALAUX	x11	0.5	111111	200000	2^{22} Hz	4
	x10	0.5	111111	111111	2^{21} Hz	8
	x01	0.5	62500	58824	2^{20} Hz	16
	x00	0.5	31250	30303	2^{19} Hz	32

Table 1.7 - Crystal Requirements for Internal DPLL

1.10 Time Facilities

The LTMS contains the following time facilities, accessible via the parallel microprocessor interface and dedicated pins:

- Time Stamp facility;
- Alarm Clock facility;
- Stopwatch facility;
- Pulse Generator facility;
- Waveform Generator facility;
- LTMS Status Register;
- CTMS Status Register;
- LTMS Control Register.

Except for the pulse generator and the waveform generator, these time facilities are under the LTMS user control. The pulse generator is always controlled by the CTMS, while the waveform generator can be either controlled from the CTMS or by the LTMS user depending on the configuration.

1.10.1 The Time Stamp Facility

The Time Stamp facility allows the LTMS user to time tag events. On the occurrence of an event that should be time stamped, the application should generate a pulse on the **etstrb** input. As soon as a rising **etstrb** edge is detected, the current value of the ET counter is latched in the time stamp facility together with the LTMS status information and the 8 most significant bits of the CTMS status register, informing about the validity of the time stamp. Instead of using the **etstrb** input, it is also possible to initiate a time stamp via a rising edge on the LTMS control register *etstrbbit* flag.

After a time stamp, the five words of the time information stored in the time stamp facility can be read using the parallel microprocessor interface, using addresses 16 to 20 (cf. figure 1.19). Address 20 is used to read the least significant bits of the ET fine counter and the 8 most significant bits of the CTMS status register.

Because the LTMS status information available through the time stamp facility is the LTMS status information that was available at the moment the time stamp operation was performed, the LTMS status fields in the time stamp facility are not necessarily identical to those currently in the LTMS status register. A similar remark applies for the 8 most significant bits of the CTMS status register.

At a time stamp event, further time stamps are disabled until address 20 has been read via the parallel microprocessor interface. Time stamp requests during this period are not served and the LTMS status register *timestamp* flag is set. This means that address 20 has to be read last. Addresses 16 to 19 can be read in any order.

Reading the time stamp facility immediately after having read address 20 should be avoided since this includes the risk to read parts of two successive time stamps. Before reading the time stamp facility again, one has to wait until after the next rising **etstrb** edge or *etstrbbit* flag transition.

There are at least two bit positions in the time stamp facility that are permanently zero. The number of bits having an invariable zero value increases if the LTMS is using an internal clock slower than 2^{22} Hz. The LTMS automatically forces the appropriate number of register positions to zero, in accordance to its configuration and the ET counter resolution (cf. table 1.1).

1.10.2 The Alarm Clock Facility

The Alarm Clock facility allows the LTMS user to generate an alarm or interrupt at a predefined time instant. The LTMS user sets the predefined alarm time by writing to the alarm register using the parallel microprocessor interface.

The **etalrm** output is set at the moment the ET counter value becomes equal to or higher than the alarm time. Bits in the alarm time with a smaller weight than the current ET counter resolution are not

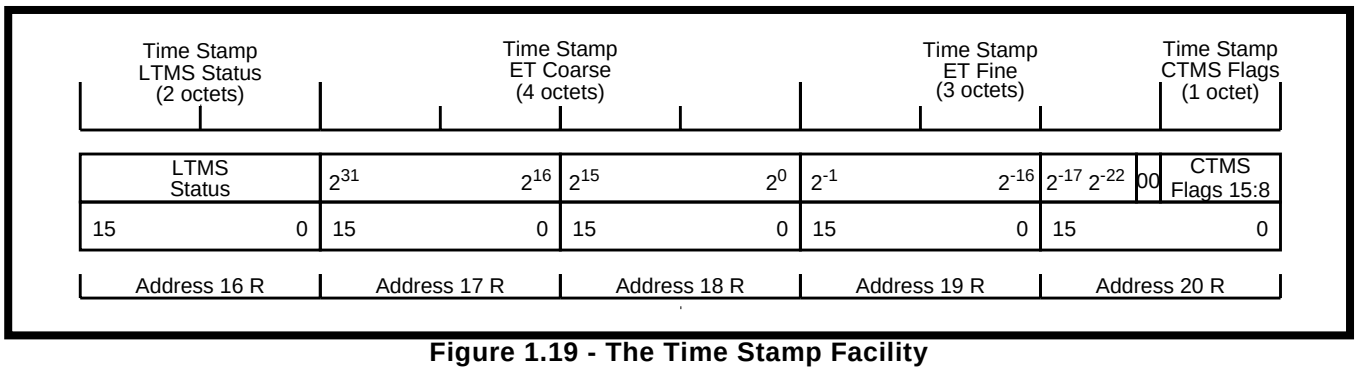


Figure 1.19 - The Time Stamp Facility

used for this comparison. The resulting rising **etalrm** edge informs the application of the event. The information on the **etalrm** output is also available via the LTMS control register *etalrmbit* flag.

The alarm register can be read via the parallel microprocessor interface using addresses 12 to 15 (cf. figure 1.20). Reading the alarm register has no impact on the comparison process. If the LTMS user writes non-zero values on bits with a weight smaller than the ET counter resolution, these values will be discarded. When reading address 15, zeros will be read on the corresponding bit positions.

The following scheme is to be used to avoid erroneous alarms while loading the predefined time in the alarm register. When address 12, 13 or 14 is written, the **etalrm** output is cleared and the comparison is disabled. The comparison is enabled again when address 15 is written. Hence it is not necessary to update all the information in the alarm clock facility to define new alarm times.

When a time already past is written to the alarm register, the **etalrm** output and the LTMS status register *alarm* flag are both set provided that the alarm clock facility is enabled.

1.10.3 The Pulse Generator Facility

The Pulse Generator facility allows the CTMS to generate pulses on the **pfgphout** output, at a maximum rate of 1 Hz (i.e. one pulse per 1 second interval). This facility is always under the control of the CTMS via the CTMS message pulse field. In case a CTMS message pulse field is available, a pulse is generated when an equality is detected between the monotonous ET counter fine time and the respective bits available in the CTMS message.

The absence of a pulse field in the CTMS message is not equivalent to a pulse field that is all-zero. In the first case no pulse is generated, while in the second case a pulse is generated at T+1 seconds if no error is detected.

The pulse generator generates a pulse on the **pfgphout** output, only if <u>all</u> of the following conditions are met (cf. figure 1.22):

- a valid CTMS message including a pulse field has been received;
- a LTMS synchronisation using that CTMS message has been performed, synchronising the ET counter to T+1 seconds;
- The fine time value of the monotonous ET counter (cf. section 1.8.3) and the relevant bits of the pulse field received in that CTMS message become equal before the ET fine counter wraps around (i.e. at T+2) or before the next LTMS synchronisation.

The pulse occurs during the first period of the clock driving the ET counter, during which the above equality exists. If the specified CTMS message pulse field was zero, the pulse comes at the LTMS synchronisation instant. The pulse length (L in figure 1.22) is equal to one period of the clock used to drive the ET counter (cf. table 1.1). If the value of the pulse field is specified to be zero, the possibility could exist that two consecutive pulses to be generated are stuck together. The LTMS however delays the second of the two pulses (the one generated because of a zero pulse value) for one period of the clock driving the ET counter. This way concatenated pulses are eliminated.

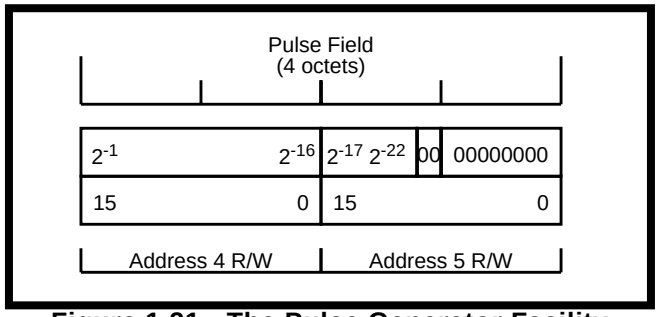


Figure 1.21 - The Pulse Generator Facility

The last values for the pulse generator parameters made available to the LTMS (i.e. not necessarily the values currently being used) can be read via the parallel microprocessor interface using addresses

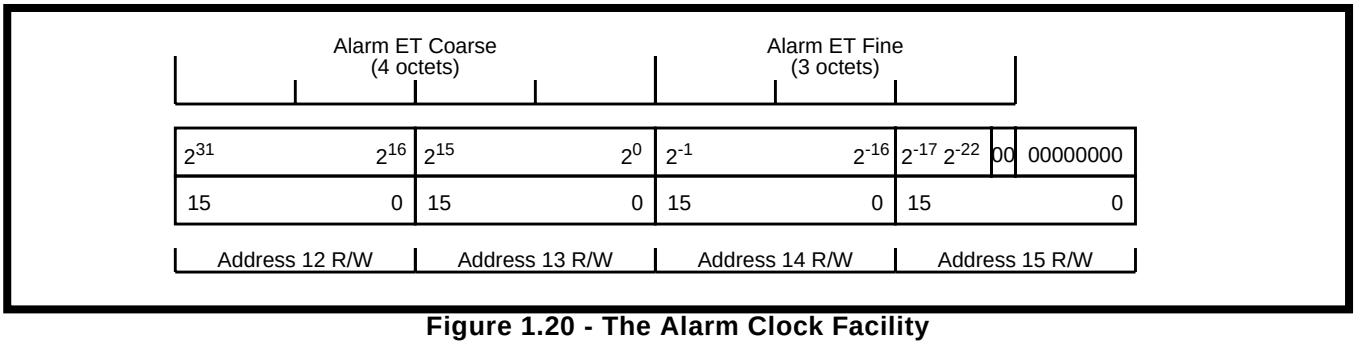


Figure 1.20 - The Alarm Clock Facility

4 and 5 (cf. figure 1.21). This should be done according to the **window** related limitations described in section 1.7.4.

The LTMS informs about the status of the pulse generator facility via the *phase* flag in the LTMS status register. The flag is normally updated at each LTMS synchronisation. However, if the fine time field of the monotonous ET counter wraps around before the next LTMS synchronisation occurs, the *phase* flag is updated at the first wrap around occurrence (at T+2). The *phase* flag then remains unchanged at the next LTMS synchronisation (at t > T+2). No pulse is generated between the wrap around instant (T+2) and the next synchronisation.

Assuming to be at T+2 while an LTMS synchronisation occurred at T+1, the *phase* flag is set if no pulse was generated on the **pfgphout** output between $T+1 \leq t < T+2$ while the CTMS message transmitted between T and T+1 included a pulse field. The *phase* flag is cleared at T+2 if no CTMS message pulse field was available (i.e. no pulse had to be generated between $T+1 \leq t < T+2$) or if a pulse has been successfully generated between $T+1 \leq t < T+2$.

1.10.4 The Waveform Generator Facility

The Waveform Generator facility provides a periodic waveform to the user. It can be controlled either from the CTMS (**pfgmode**=0) or by the LTMS user himself (**pfgmode**=1). The frequency and waveform shape can be selected by programming three parameters: the Waveform Length, the Polarity, and the Duty Cycle Ratio (cf. table 1.8).

If the CTMS generates the parameters, the waveform generator is controlled by parameters included in CTMS messages having the *waveform* flag set. If no new parameters have to be provided, CTMS messages with the *waveform* flag cleared should be transmitted. A set of parameters is passed to the waveform generator at LTMS synchronisations only. If a new CTMS message is attempted before an LTMS synchronisation, it will over write the previous one. Only valid CTMS messages are used by the waveform generator.

If the waveform generator is controlled by the LTMS user, the user writes the waveform generator parameters via the parallel microprocessor interface using addresses 28 to 31. The addresses have to be written in ascending order. A new set of parameters is considered to be available when the facility detects the last write operation (address 31).

In addition to the three parameters, the waveform generator facility requires at least one rising **pfgphin** edge (cf. figure 1.23) or an assertion of the LTMS control register *pfgphinbit* flag to trigger the waveform generation. This rising edge is called the Start Phase, as it allows to control the phase of the waveform. Once the generation is initiated, subsequent start phases restart the waveform generation (i.e. performing phase corrections).

The parameters most recently transmitted to the LTMS can be read using either addresses 6 to 9 (if **pfgmode**=0) or addresses 28 to 31 (if **pfgmode**=1) (cf. figure 1.24). Reading the parameters has no influence on the waveform generation.

Assuming that the waveform generator facility is stopped (either after a stop command or an external reset) and that a new set of parameters is received without the stop command, it discards the new parameter set if one or more of the following conditions is fulfilled:

- the duty cycle ratio is equal to 1;
- the waveform length is too short;
- the waveform generator facility is still disabled following an external reset as described in section 1.12.4.

If none of these conditions are true the new set of parameters is valid and used for waveform

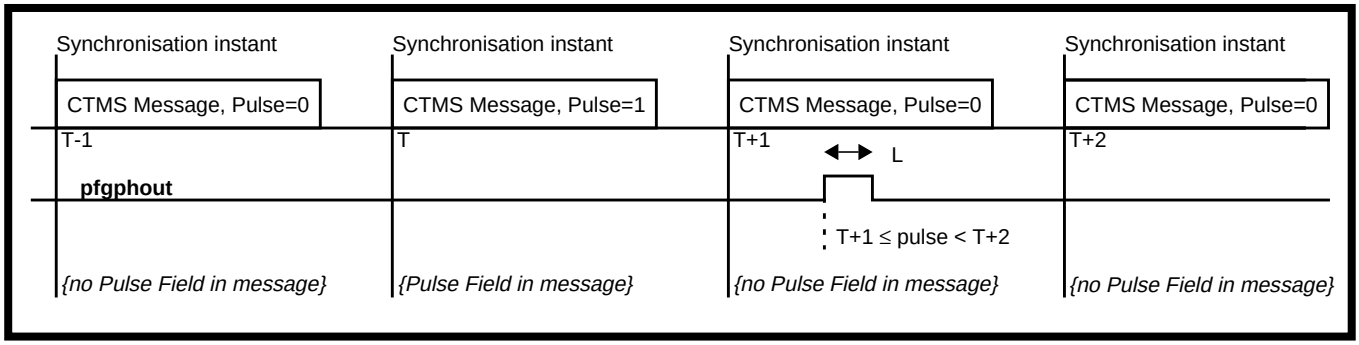


Figure 1.22 - Pulse Generator Facility Functionality

generation on the **pfgwave** output. The new polarity value is placed on the **pfgwave** output synchronously with the clock driving the ET counter. The waveform generator facility is put on hold: no waveform is generated on the **pfgwave** output yet. New sets of parameters, which do not include the stop command, are discarded when the waveform generator is on hold.

The facility only goes from hold to active at the detection of the first rising **pfgphin** edge. As soon as it is active, the facility generates the waveform specified by the new parameters. Rising edges on the **pfgphin** input, when the waveform generator facility is active, are used to immediately phase adjust the waveform. The waveform generation restarts, as it did after the **pfgphin** rising edge that made the facility go from hold to active.

If the waveform generator facility is active, it is generating a waveform. If a new set of parameters becomes available, the facility checks if these parameters include the stop command. If no stop command is detected, the facility discards this new set of parameters and continues to generate a waveform according to its previous valid parameter set without visible disruption on the **pfgwave** output. If the set includes a stop command, the facility stops the waveform generation and adjusts the polarity on the **pfgwave** output according to the polarity bit available with the command. Stop commands received while the facility is stopped, adjust the **pfgwave** polarity each time according to the polarity sent with the command.

The start phase can also be created by a 0 to 1 transition of the *pfgphinbit* flag in the LTMS control register. It is also possible to obtain a copy of the **pfgwave** output via the LTMS control register *pfgwavebit* flag. The status of the waveform generator facility is available via the *waven* flag in the LTMS status register. The flag is set when the facility is active. It is cleared as soon as the facility is on hold or is stopped.

If **pfgmode**=1, waveform parameters available in the CTMS message are not used by the waveform generator, but can however be read back using address 6 to 9 and are shifted out on the extension interface. If **pfgmode**=0, addresses 28 to 31 can be written and read but are not used by the LTMS.

Waveform Length	40 bits	The waveform length is the duration during which the waveform is active during one waveform period (cf. figure 1.23). The waveform length is divided in a coarse (16 bits) and a fine part (24 bits). The format is as for the ET with the same bit weight boundaries. Consequently, it is possible to specify the waveform length between 2^{-22} and 2^{16} seconds. The LTMS remains stopped if a length below the ET resolution is specified (cf. table 1.1). The bits of the waveform length parameter having a weight lower than the least significant bit of the ET counter are ignored by the waveform generator facility.
Polarity	1 bit	When zero, the **pfgwave** output is 1 during the active part of the waveform. When one, the **pfgwave** output is 0 during the active part of the waveform.
Duty Cycle Ratio	15 bits	These bits define the ratio between the waveform period and the length. In accordance to this definition, the ratio values 0 and 1 are meaningless. The value 0 is instead used to stop the generation of a waveform and is referred as the Stop command.

Table 1.8 - Waveform Generator Parameters

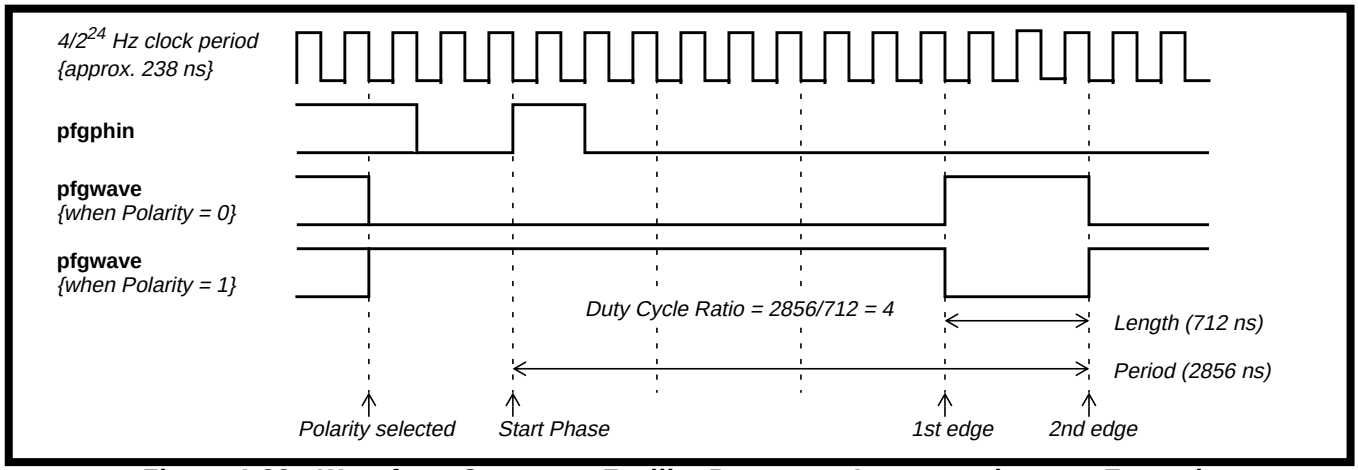


Figure 1.23 - Waveform Generator Facility Parameter Interpretation - an Example

1.10.5 The Programmable Frequency Generator

The pulse and waveform generators can be combined to form a Programmable Frequency Generator, by externally connecting the **pfgphout** and **pfgphin** pins, and it can be controlled from the CTMS. The generated periodic waveform is phase locked to the time reference and is also phase, frequency, and shape controlled by parameters supplied in a CTMS message (cf. figure 1.25).

1.10.6 The Stopwatch Facility

The Stopwatch facility allows to accurately measure relative time periods without being disturbed by the LTMS synchronisation process. It can either measure the time during which a predefined number of events occur (Time Measurement) or count the number of events occurring during a predefined time interval (Event Measurement). To do this, the stopwatch facility uses two counters (cf. figure 1.26):

- **The Time counter** is a free-running non-wrapping binary counter driven by the 2^{24} Hz clock if **auxtal**=1 (the resolution is 2^{-24}), or the bus clock if **auxtal**=0 (the resolution is 2^{-22}, 2^{-21}, 2^{-20} or 2^{-19} depending on the **busclk** frequency). Any bit with a weight below the current resolution has the value zero.
- **The Event counter** is a non-wrapping binary counter, counting the number of rising **swevent** edges or 0 to 1 transitions of the LTMS control register *sweventbit* flag.

Time measurement is selected by writing the number of events (addresses 24 to 26, in said order) in the event counter. The time counter is reset to 0 and the stopwatch is disabled when a write operation to any of these addresses is performed.

In this measurement mode, the event counter decrements while the time counter increments. As soon as the number of user specified events is reached, the time counter is stopped and the LTMS status register *stopwatch* flag is set. The time counter value can then be read. A special case is to specify a single event, which allows to measure a time interval.

After writing address 26, the stopwatch facility waits for a rising **swstart** edge (if *swselbit*=1 in LTMS control register) or a 0 to 1 transition on the *swstartbit* (if *swselbit*=0) to enable the measurement.

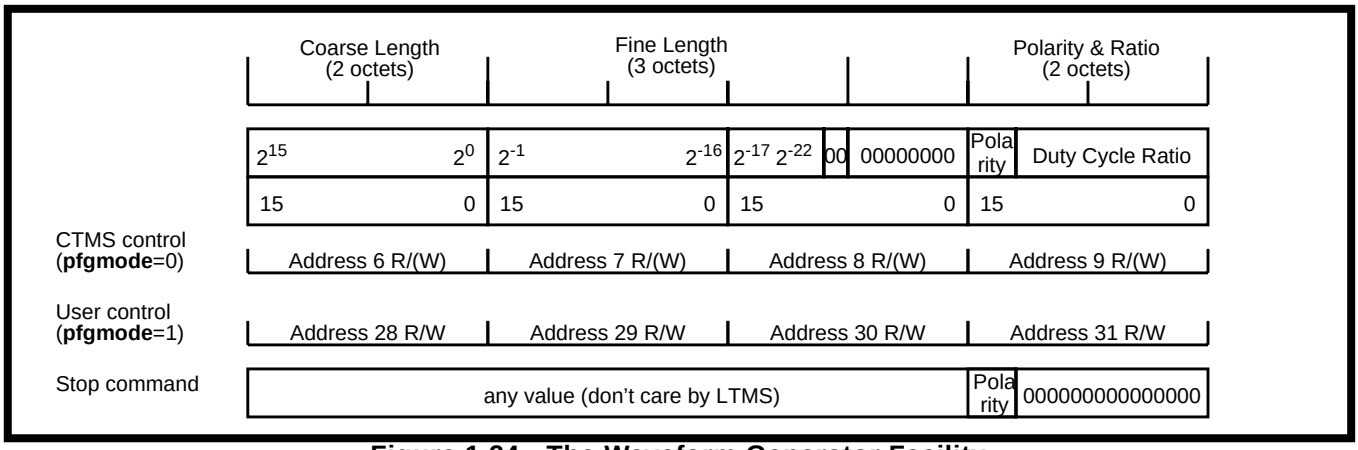


Figure 1.24 - The Waveform Generator Facility

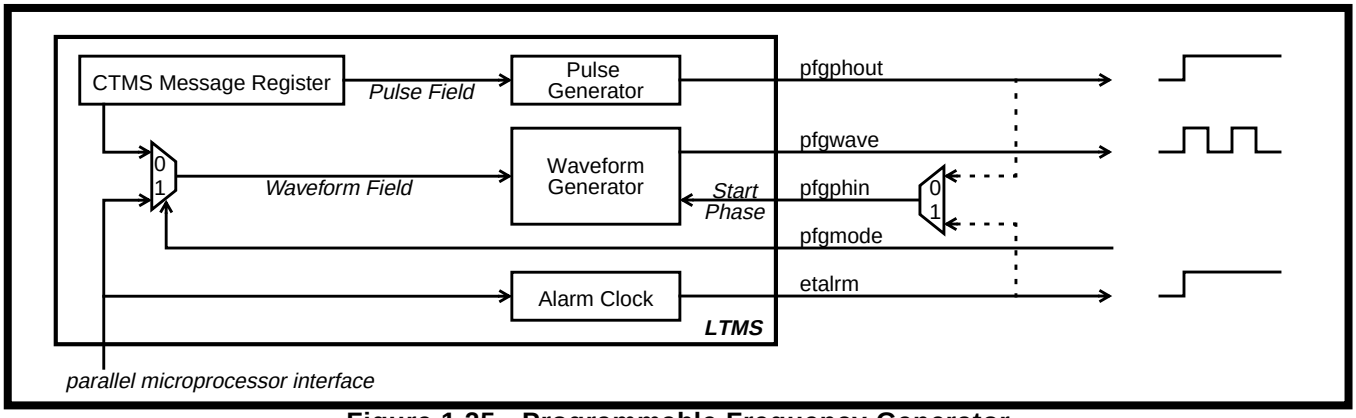


Figure 1.25 - Programmable Frequency Generator

Event measurement is selected by writing the time duration (addresses 21 to 23, in said order) in the time counter. The event counter is reset to 0 and the stopwatch is disabled when a write operation to any of these addresses is done.

In this measurement mode, the time counter decrements, while the event counter increments. When the time counter reaches zero, the event counter is stopped. The LTMS user can then read the value of the event counter to know how many events occurred during the time interval.

After writing address 23 the stopwatch facility waits for a rising **swstart** edge (if *swselbit*=1) or a 0 to 1 transition on the *swstartbit* (if *swselbit*=0) to enable the measurement.

Any type of measurement is enabled by a rising **swstart** edge, when the LTMS control register *swselbit* flag equals zero. When the *swselbit* flag is one, a 0 to 1 transition of the LTMS control register *swstartbit* flag enables the measurement. The measurement continues as long as the **swstart** input (or the *swstartbit* flag) is one. Temporarily halting the measurement is possible by putting the **swstart** input (or the *swstartbit* flag) to zero. A new rising edge continues the measurement.

Reading the stopwatch facility registers is only allowed when a measurement is completed (LTMS status register *stopwatch* flag is set) or halted. The measurement is completed when one of the following conditions is true:

- the specified duration is elapsed when in event measurement mode;
- the specified number of events has been reached when in time measurement mode;
- there is a stopwatch counter saturation.

When the measurement is completed, the LTMS status register *stopwatch* flag is immediately set. A counter saturation can be detected by reading the stopwatch facility counter that was preset for this measurement. If it is not zero, a saturation occurred. Remaining saturation cases can be detected by reading the saturated value of the stopwatch facility counter that was not preset.

1.10.7 The LTMS Status Register

The LTMS Status register (address 10) provides information about the status of the LTMS facilities and hence the validity of the current time information. The flags available in the LTMS status register are explained in tables 1.7 and 1.8. A summary of the time validity information in the LTMS status register is available via the **tvld** output and the LTMS control register *tvldbit* flag (cf. section 1.10.9).

1.10.8 The CTMS Status Register

The CTMS Status register (address 11) holds information transmitted to the LTMS in CTMS messages. CTMS status field bits 14 to 0 of a valid CTMS message are copied to the CTMS status register. Bit 15 is used by the LTMS as a toggle bit signalling register updates. Table 1.9 shows what information is available in the CTMS status register.

Bits 15 to 8 of the CTMS status register are copied to the time stamp facility at a time stamp occurrence.

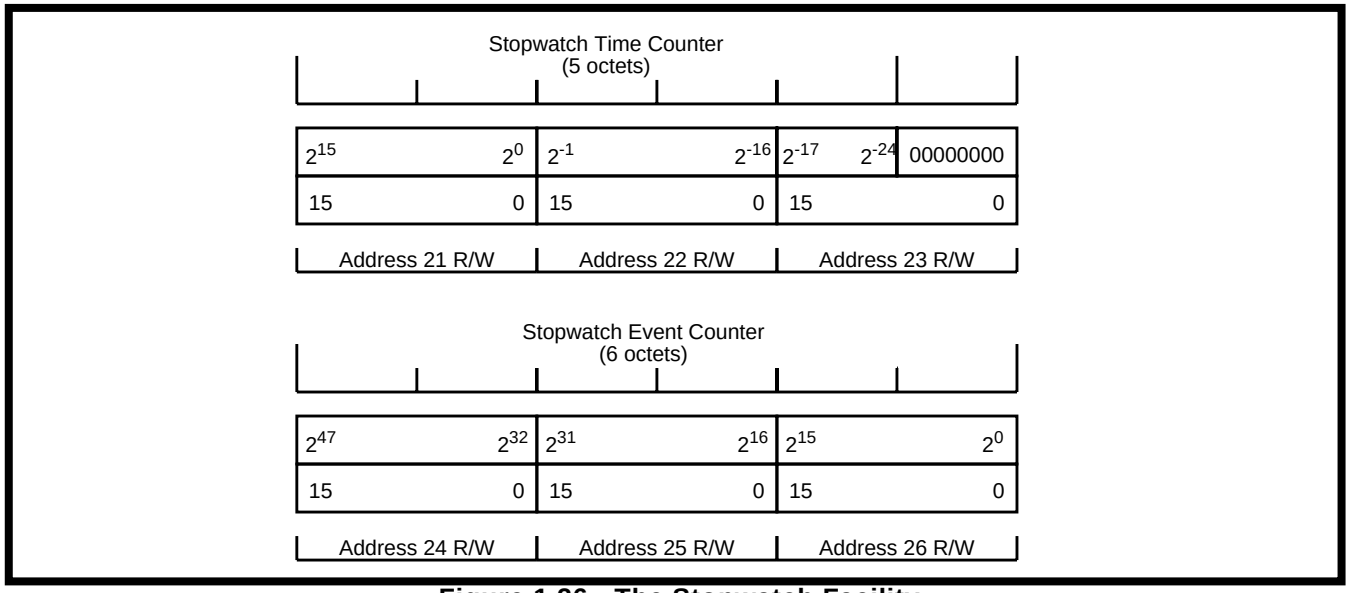


Figure 1.26 - The Stopwatch Facility

Bit	Name	Description
15	*stopwatch*	This flag is set when the stopwatch has completed the requested measurement. I.e. when the event counter (when time measurement) or the time counter (when event measurement) reaches 0. The *stopwatch* flag is also set when the event counter (when event measurement) or the time counter (when time measurement) saturates. The flag is cleared when the user starts to write to the stopwatch facility.
14	*waven*	This flag is set as soon as the waveform generation has started on the **pfgwave** output, i.e. the waveform generator facility is active. The *waven* flag is reset as soon as the waveform generator facility is stopped or put on hold.
13	*phase*	This flag concerns the pulse generator facility. The *phase* flag is updated at LTMS synchronisation instant or at the first wrap around instant of the fine time of the monotonous ET counter following the last LTMS synchronisation. In case the *phase* flag is updated at the wrap around instant, the *phase* flag value simply remains unchanged at the next LTMS synchronisation at t =T+2. Assuming to be at T+2 while an LTMS synchronisation occurred at T+1, the *phase* flag is set to 1 if no pulse was generated on **pfgphout** between T+1 = t < T+2 while the CTMS message transmitted between T and T+1 included a pulse field. The *phase* flag is cleared at T+2 if no CTMS message pulse field was available or if a pulse has been successfully generated between $T+1 \leq t < T+2$.
12	*alarm*	Writing an alarm time greater than the current ET de-asserts the *alarm* flag, until the ET becomes equal or greater than the set alarm time. This flag is set after an external reset. In serial and parallel operation it is cleared on the second LTMS synchronisation. In stand alone mode it is cleared at the moment address 15 is written. From the moment it is cleared, the flag is set when a time already past is loaded in the alarm clock facility, in all operations.
11	*timestamp*	This flag is set when an event is detected on the **etstrb** input but cannot be served since the time stamp facility is busy. It is cleared only when the next event is served correctly.
10	*synchto*	This flag is set when a synchronisation marker timeout is detected in serial and parallel operation. It is cleared at the detection of the next correctly timed synchronisation marker reception. In stand alone operation, the *synchto* flag is always zero.
9	*etcto*	This flag is set when the ET threshold is exceeded at the moment an LTMS synchronisation should be performed (cf. sections 1.8.2 and 1.8.3). Whether the LTMS synchronisation is performed or not depends on conditions explained in section 1.8.1. The *etcto* flag is reset at a following LTMS synchronisation where the synchronisation marker is inside the ET threshold window.
8	*windout*	In parallel and stand alone operation, this flag is set when a window violation is encountered while writing to the CTMS message register (cf. section 1.7.3). The *windout* flag is cleared as soon as address 1 of a new CTMS message is written while **window**=1. Read accesses have no impact on the *windout* flag in parallel operation. In serial operation, the *windout* flag is set as soon as the window violations described in section 1.7.4 are encountered. The *windout* flag is reset the next time address 1 of the CTMS message is read while **window**=1.
7	*exterror*	The *exterror* flag is set when the **exterin** input and/or LTMS control register *exterinbit* flag are set. It means that an external check has detected an error in the CTMS message. No regular LTMS synchronisation is performed as long as the *exterror* flag is set (cf. section 1.8.5). However, the assertion of the *exterror* flag does not disable the acquisition of new CTMS messages. The *exterror* flag is cleared when both the **exterin** input and the LTMS control register *exterinbit* flag become/are low.
6	*wasfree*	This flag is set when the DPLL has been free-wheeling at least once since the last LTMS synchronisation, warning that the local time reference has possibly drifted (cf. sections 1.9.3 and 1.9.4). The *wasfree* flag is reset on the next LTMS synchronisation.

Table 1.9 - The LTMS Status Register

Bit	Name	Description
5	*free*	This flag is set when the DPLL is free-wheeling, and is cleared when in lock again (cf. sections 1.9.3 and 1.9.4).
4-3	*errcode[1:0*]	These flags hold information about how the last received CTMS message has been handled by the LTMS (cf. table 1.10, and sections 1.7.5, 1.8.1 to 1.8.6).
2-0	*errcount[2:0]*	This is the Error Counter that records the build-up of message and synchronisation errors detected in CTMS messages (cf. section 1.8.6).

Table 1.9 - The LTMS Status Register

Code	Name	Description
00	No error	LTMS synchronisation has been performed. No message error, synchronisation error or warning was detected (cf. sections 1.7.5, 1.8.1 and 1.8.5).
01	Message Error	A message error or a synchronisation marker timeout has been detected. No LTMS synchronisation has been or will be performed.
10	Synchronisation Error	In serial and parallel operation, a valid CTMS message has been received, but a synchronisation error occurred. No LTMS synchronisation has been performed. In stand alone operation, a valid CTMS message has been received, but a synchronisation error or warning was detected. In case of an error, no LTMS synchronisation occurred. In case of a warning, the LTMS synchronisation has been performed but a time discontinuity may have occurred.
11	Initialisation or Go-Synchronisation	This code warns that a time discontinuity occurred. A valid CTMS message has been received. At the moment the LTMS synchronisation occurred, the synchronisation marker that resulted in the LTMS synchronisation did not occur inside the ET threshold window; or the coarse time field of the CTMS message was not equal to the ET coarse counter + 1. However, the initialisation flag was set or, in serial and parallel operation only, the go-threshold was reached while the LTMS status register *exterror* flag was 0.

Table 1.10 - Error Codes in LTMS Status Register

Bit	Name	Description
15	*Toggle*	Bit 15 is toggled each time new CTMS status information is loaded into the CTMS Status register.
14	*Pulse Flag*	Pulse Flag of the last valid CTMS message preceding the latest synchronisation marker.
13	*Waveform Flag*	Waveform Flag of the last valid CTMS message preceding the latest synchronisation marker.
12	*Initialisation Flag*	Initialisation Flag of the last valid CTMS message preceding the synchronisation marker.
11-8	*Time Chain Flags*	Time chain flags of the last valid CTMS message preceding the synchronisation marker.
7-0	*Mission Specific Flags*	Mission specific flags of the last valid CTMS Message preceding the synchronisation marker.

Table 1.11 - The CTMS Status Register

1.10.9 The LTMS Control Register

Most of the information in the LTMS Control register (address 27) is already available via dedicated pins. The LTMS control register provides a means to control and observe the LTMS time facilities through the parallel microprocessor interface. Using the LTMS control register reduces the number of external LTMS time facility control lines and address decoding logic used in addition to the parallel microprocessor interface; due to bus access delays the accuracy of the LTMS time facilities will however decrease. It is possible to use the LTMS without using the LTMS control register at all.

Some of the LTMS control register bits are generated by the user, some by the LTMS and some by both. Reading the LTMS control register has no effect on its contents. When the LTMS control register is written, the read-only bits are not affected.

Reading the LTMS control register provides the values currently used by the LTMS. This means that it is possible that the bit value immediately read after a write access (for writeable bits only) differs from the written value. Reading after the time necessary for the LTMS to recognise and initiate the relevant action results in reading the just written value.

The functionality of the LTMS control register is summarised in table 1.10.

Bit	Name	Mode	Control	Definition
15-10	not used	R	LTMS	Zero values are provided during read.
9	*tvldbit*	R	LTMS	The *tvldbit* flag is a copy of the **tvld** output (cf. section 1.8.7).
8	*swstartbit*	R/W	LTMS and user	This flag allows the LTMS user to control the stopwatch facility operations from the LTMS control register when *swselbit* is one. In this case, the stopwatch can be started or continued with *swstartbit* one; it is halted by clearing *swstartbit* to zero. When *swselbit*=0, the stopwatch operation is under control of the **swstart** input. The LTMS copies the **swstart** value to the *swstartbit* flag. Reading *swstartbit* can thus be used for monitoring **swstart**.
7	*swselbit*	R/W	User	The *swselbit* flag allows to select if the stopwatch facility operations (i.e. start/continue or hold/stop the measurement) are controlled via the **swstart** input (if *swselbit* = 0, being default after reset) or via the *swstartbit* flag (if *swselbit* = 1). User requests for updating the *swselbit* flag (and changing the control mode) are only accepted if, at the same time, the *swstartbit* flag is specified to be zero. If *swstartbit* is one, the *swselbit* remains unchanged.
6	*sweventbit*	R/W	User	A 0 to 1 transition on the *sweventbit* flag is equivalent to generating a pulse on the **swevent** input. Pulses generated on the **swevent** input are not reflected onto the *sweventbit* flag.
5	*exterinbit*	R/W	User	The *exterinbit* flag is fully under the control of the user. When *exterinbit*=1 indicates that external checks on the current CTMS message have detected errors. Asserting the *exterinbit* flag has the same effect as raising the **exterin** input but both operate independently. Either one being set impedes synchronisation. (cf. section 1.8.5). The **exterin** input is not reflected on the *exterinbit*, but can be monitored via the LTMS Status register).
4	*pfgwavebit*	R	LTMS	The *pfgwavebit* flag is a copy of the **pfgwave** output.

Table 1.12 - The LTMS Control Register

Bit	Name	Mode	Control	Definition
3	*pfgphinbit*	R/W	LTMS and user	The *pfgphinbit* flag is set when: • a one is written to this bit position • a rising edge is detected on the **pfgphin** input. Assuming that the *pfgphinbit* flag is initially zero, raising the flag has the same effect as rising the **pfgphin** input (cf. section 1.10.4). When the *pfgphinbit* flag is high, there is a difference. In particular, the next rising edges detected on the **pfgphin** input are not disabled: the related phase information is passed to the waveform generator facility whatever the value of the *pfgphinbit* flag is. The *pfgphinbit* flag hence has not to be cleared at each rising edge of the **pfgphin** input in case that the waveform generator facility is controlled by the **pfgphin** input. Attempting to assert the *pfgphinbit* flag while it is already one does not generate further phase signals to the waveform generator facility: the *pfgphinbit* flag has to be cleared first.
2	*etstrbbit*	R/W	LTMS and user	The *etstrbbit* flag is set when: • a one is written to this bit position • a rising edge is detected on the **etstrb** input. Asserting *etstrbbit* has the same effect as raising the **etstrb** input (cf. section 1.10.1). The *etstrbbit* flag is cleared by the LTMS at the end of a read access to address 20 of the time stamp facility (i.e. when the time stamp facility is ready for a new time stamp cycle).
1	*etalrmbit*	R	LTMS	The *etalrmbit* flag is a copy of the **etalrm** output.
0	*windowbit*	R	LTMS	The *windowbit* flag is a copy of the **window** output.

Table 1.12 - The LTMS Control Register

1.11 Parallel Microprocessor Interface

All LTMS registers are accessible via the Parallel Microprocessor Interface. The register address mapping is defined in table 1.13.

The timing of the data, address and control lines of the microprocessor interface is directly compatible with the MA31750 microprocessor when **mproc**=1. When **mproc**=0, it is directly compatible with the ERC32 microprocessor. Both of these interface types need the control input signals Chip Select (**cs_bar**)**,** Read (**rd_bar**) and Write (**wr_bar**) to be synchronous with the Microprocessor Clock (**mproclk**). These control signals are sampled by the LTMS on the falling edge of the **mproclk** input when **mproc**=0 and on the rising edge of the **mproclk** input when **mproc**=1.

When **auxtal**=0, the minimum frequency of the **mproclk** input should be twice the frequency of the **busclk** input. When **auxtal**=1, the minimum frequency of the **mproclk** input is unlimited. The maximum frequency of the **mproclk** input should be less or equal to 2^{24} Hz (i.e. the frequency of the **xtal1** input).

When **mproc**=1, the **ready_bar** output is asserted when a read or write cycle is completed, otherwise it is permanently deasserted.

Figure 1.27 shows a schematic of the ERC32 chip set (IU, FPU and MEC) interfaced with the LTMS. The glue logic consists of an octal D-flip-flop with clock enable (54AC377) to latch the addresses and two octal bus transceivers for the 16 bit data bus (54HSC/T245).

Figure 1.28 shows how the LTMS should be interfaced with the MA31750 microprocessor including the glue logic necessary to create the **cs_bar** input and to buffer the **data** bus.

LTMS

Word description		Address	Read	Write	Comments
Not Used		0	R	-	All zero when read.
CTMS MESSAGE	CTMS Status Field	1	R	W	Writable only in parallel and stand alone operation. Read back of a CTMS message is only possible after the end of the message is detected by the LTMS.
	CTMS Coarse Time Field MSW	2	R	W	
	CTMS Coarse Time Field LSW	3	R	W	
	Pulse Field MSW	4	R	W	
	Pulse Field LSW	5	R	W	
	Waveform Coarse Length Field	6	R	W	
	Waveform Fine Length Field MSW	7	R	W	
	Waveform Fine Length Field LSW	8	R	W	
	Polarity & Duty Cycle Ratio	9	R	W	
LTMS Status Register		10	R	-	
CTMS Status Register		11	R	-	
ALARM	Alarm - ET Coarse MSW	12	R	W	
	Alarm - ET Coarse LSW	13	R	W	
	Alarm - ET Fine MSW	14	R	W	
	Alarm - ET Fine LSW	15	R	W	
TIMESTAMP	Time Stamp - LTMS Status	16	R	-	
	Time Stamp - ET Coarse MSW	17	R	-	
	Time Stamp - ET Coarse LSW	18	R	-	
	Time Stamp - ET Fine MSW	19	R	-	
	Time Stamp - ET Fine LSBs & CTMS Status 8 MSBs	20	R	-	
STOPWATCH	Stopwatch - Coarse time	21	R	W	A write access enables event measurement.
	Stopwatch - Fine MSW	22	R	W	
	Stopwatch - Fine LSB	23	R	W	
	Stopwatch - Events MSW	24	R	W	A write access enables time measurement.
	Stopwatch - Events	25	R	W	
	Stopwatch - Events LSW	26	R	W	
LTMS Control Register		27	R	W	Only bits 2, 3 and 5 to 8 are writable.
WFAVEM	Waveform Coarse Length	28	R	W	Only internally used when **pfgmode**=1. If **pfgmode**=0, these address locations can be used as storage registers.
	Waveform Fine Length MSW	29	R	W	
	Waveform Fine Length LSW	30	R	W	
	Polarity & Duty Cycle Ratio	31	R	W	

Table 1.13 - LTMS Register Address Map

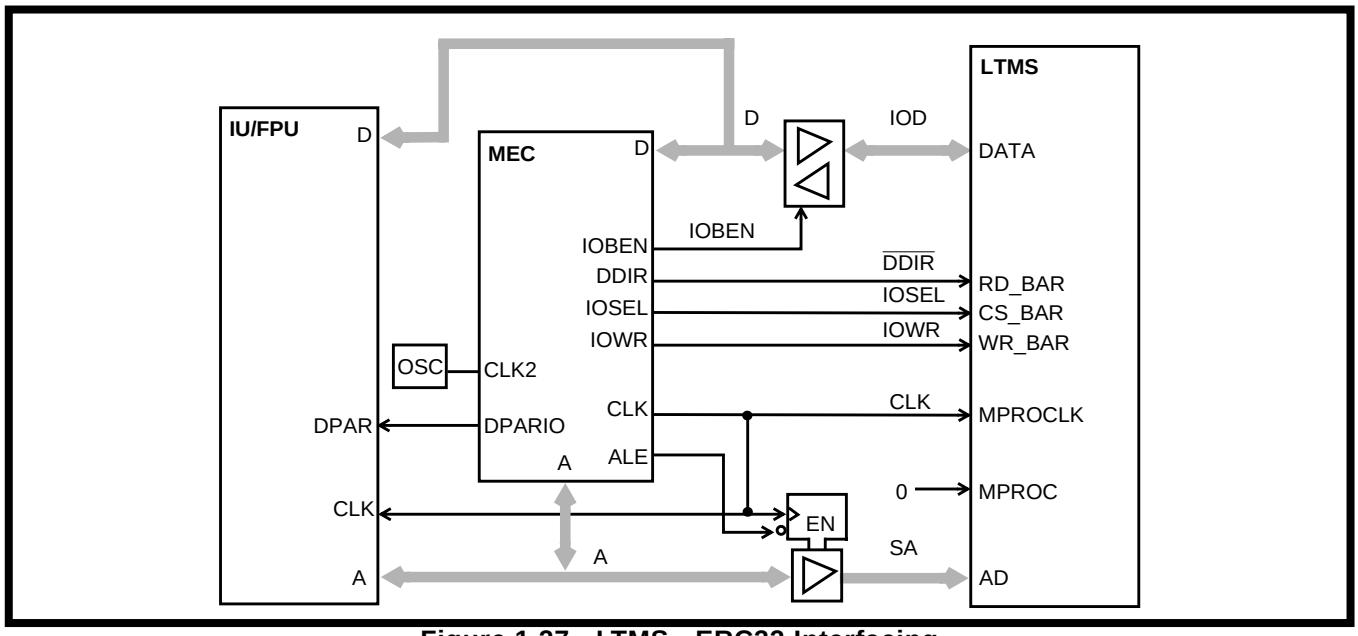


Figure 1.27 - LTMS - ERC32 Interfacing

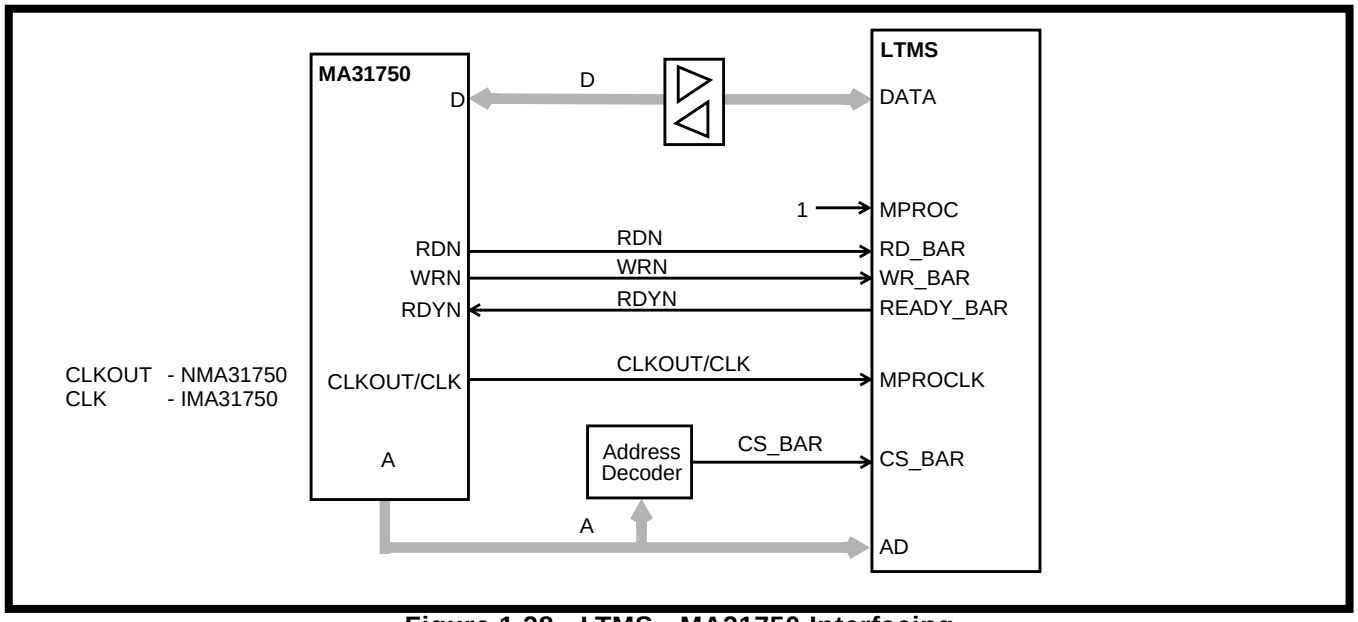


Figure 1.28 - LTMS - MA31750 Interfacing

1.11.1 ERC32 Write Cycle

Figure 1.29 shows an example of two complete write cycles and the start of a third one. The first write cycle is the shortest possible one, while the second write cycle is extended with two extra **mproclk** periods by keeping the **cs_bar** and **wr_bar** inputs asserted for two more **mproclk** periods. This allows slow devices to communicate with the LTMS.

The **ad** input is always latched internally in the LTMS at the same time after the start of a Write cycle. After the LTMS samples **data**, either the **wr_bar** or the **cs_bar** input should become deasserted before the next rising **mproclk** edge, otherwise **data** will be re-sampled at that rising **mproclk** edge.

The LTMS detects the end of a write cycle and allows the start of a new write/read cycle at the falling **mproclk** edge after the **cs_bar** input became deasserted, during which the value of the **wr_bar** input is not sampled with the **mproclk** input. This means that it is not necessary to deassert **wr_bar** between subsequent write cycles. This is shown in figure 1.29 between the second and third write cycle. Note that the T5s constraint is relevant only when the information on the **data** bus becomes invalid.

In figure 1.29, the write moment is indicated. This is the **mproclk** edge to which timing is referred when using the LTMS control register to control the time facilities.

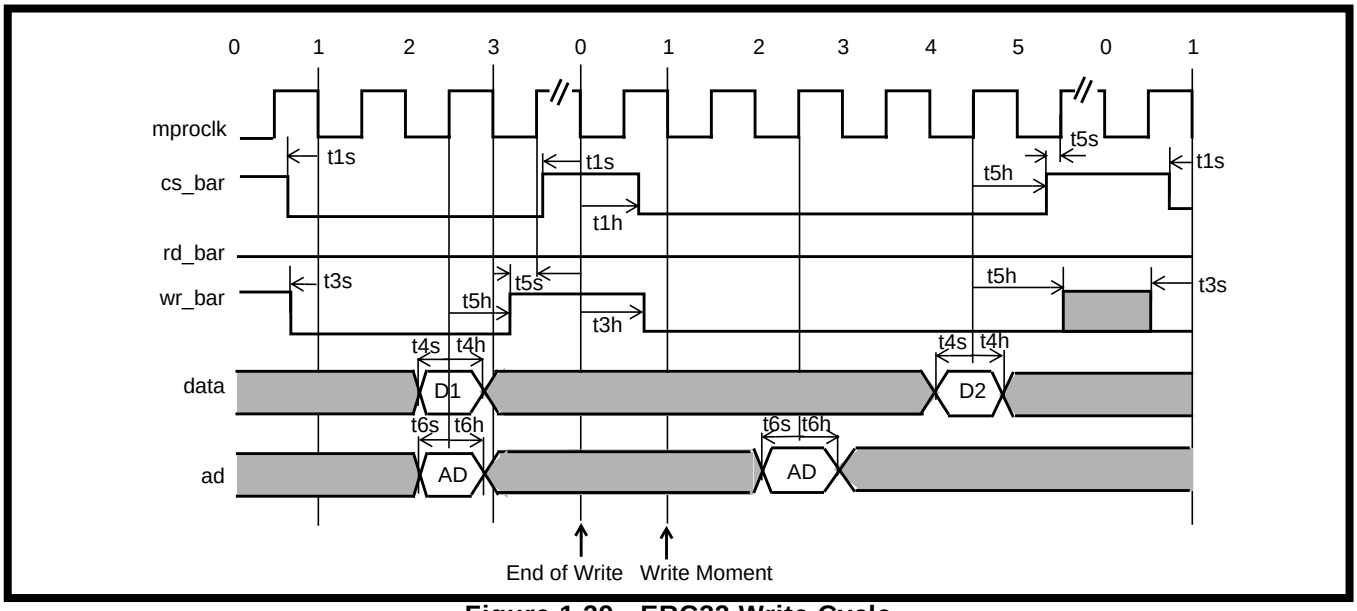


Figure 1.29 - ERC32 Write Cycle

1.11.2 ERC32 Read Cycle

Figure 1.30 shows an example of two subsequent read cycles. The first read cycle is the shortest possible cycle while the second read cycle is extended with one extra **mproclk** period by keeping the **cs_bar** and **rd_bar** inputs asserted for one additional **mproclk** period. This allows slow devices to communicate with the LTMS.

The LTMS detects the end of a read cycle and allows the start of a new write/read cycle at the falling edge of the **mproclk** input when either the **cs_bar** or the **rd_bar** input is deasserted. This means that between subsequent read cycles either the **rd_bar** or the **cs_bar** input can remain asserted, but not both.

In contrast to the write cycle, the **ad** input bus is not latched and should remain unchanged during the read cycle as shown in figure 1.30.

If the **rd_bar**, **wr_bar** and **cs_bar** inputs are all sampled asserted at the falling edge of the **mproclk** input, an illegal cycle is detected and the following actions are performed:

- the **data** bus is immediately set to a high-impedance state;
- the current cycle is interrupted;
- the LTMS waits until the **cs_bar** input is sampled deasserted on the falling **mproclk** edge.

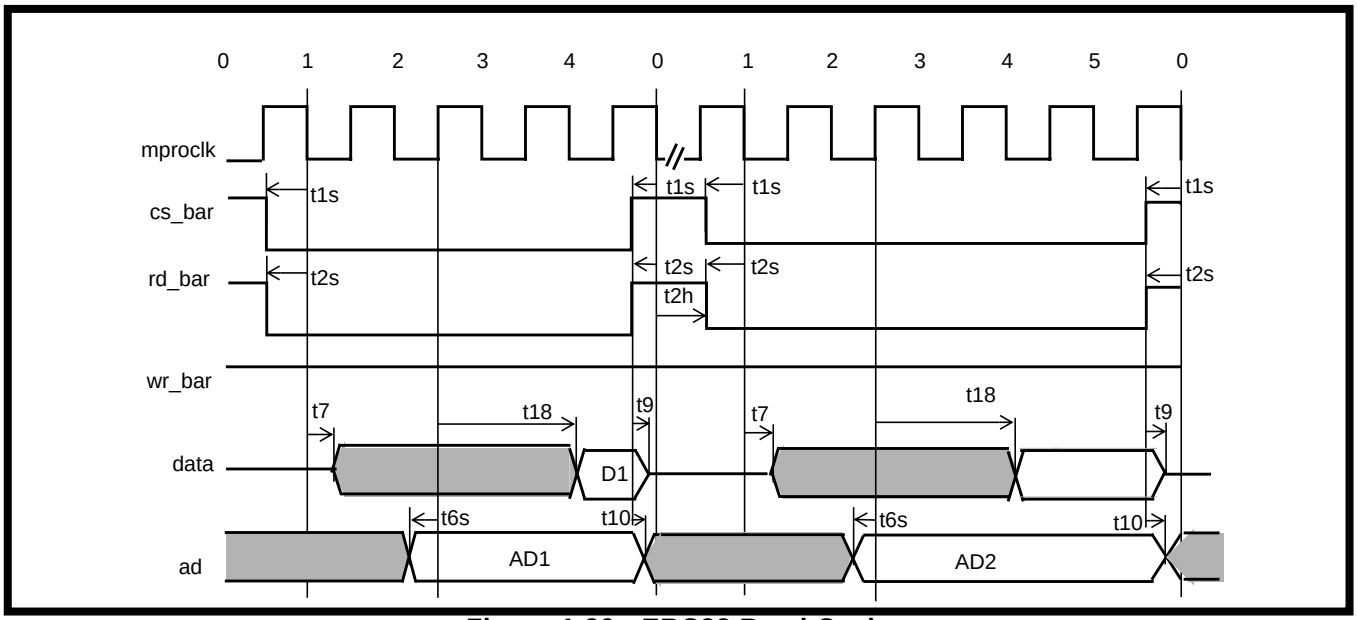


Figure 1.30 - ERC32 Read Cycle

1.11.3 MA31750 Write Cycle

Figure 1.31 shows an example of two complete write cycles and the start of a third one. The first write cycle is the shortest possible cycle while the second write cycle is extended with two extra **mproclk** periods by keeping the **cs_bar** and **wr_bar** inputs asserted for two additional **mproclk** periods. This allows slow devices to communicate with the LTMS.

The **ad** input is always latched internally in the LTMS at the same time after the start of a write cycle.

After the LTMS samples valid **data,** the **wr_bar** input should become high before the next rising **mproclk** edge, unless data remains valid, meaning that the LTMS will continue to sample **data** until the **wr_bar** input becomes deasserted.

The LTMS detects the end of a write cycle and allows the start of a new write/read cycle at the rising **mproclk** edge after the **wr_bar** input became deasserted independently of the value of the **cs_bar** input. It is therefore possible to keep the **cs_bar** input asserted during consecutive write cycles. This is also true for consecutive read and write cycles in any order. The **ready_bar** input always remains asserted until the end of the write cycle is detected.

Since only an asserted **wr_bar** input value ends the write cycle the **ready_bar** output will remain asserted when the **cs_bar** input becomes asserted while the **wr_bar** input remains deasserted. The **data** however will not be sampled any more from the moment that the **cs_bar** input becomes asserted.

In figure 1.31, the write moment is indicated. This is the **mproclk** edge to which timing is referred when using the LTMS control register to control the time facilities.

1.11.4 MA31750 Read Cycle

Figure 1.32 shows an example of two subsequent read cycles. The first read cycle is the shortest possible cycle while the second one is extended with one extra **mproclk** period by keeping the **cs_bar** and **rd_bar** inputs asserted for one additional **mproclk** period. This allows slow devices to communicate with the LTMS. The LTMS detects the end of a read cycle and allows the start of a new write or read cycle on the rising **mproclk** edge after **rd_bar** input becomes deasserted independently of the value of the **cs_bar** input. It is therefore possible to keep the **cs_bar** input asserted during consecutive read cycles. This is also true for consecutive read and write cycles in any order.

The **ready_bar** output always remains asserted until the end of the read cycle is detected. Since only an asserted **rd_bar** input value ends the read cycle the **ready_bar** output will remain asserted when the **cs_bar** input becomes deasserted while the **rd_bar** input remains asserted. The **data** however will immediately be set to a high impedance state.

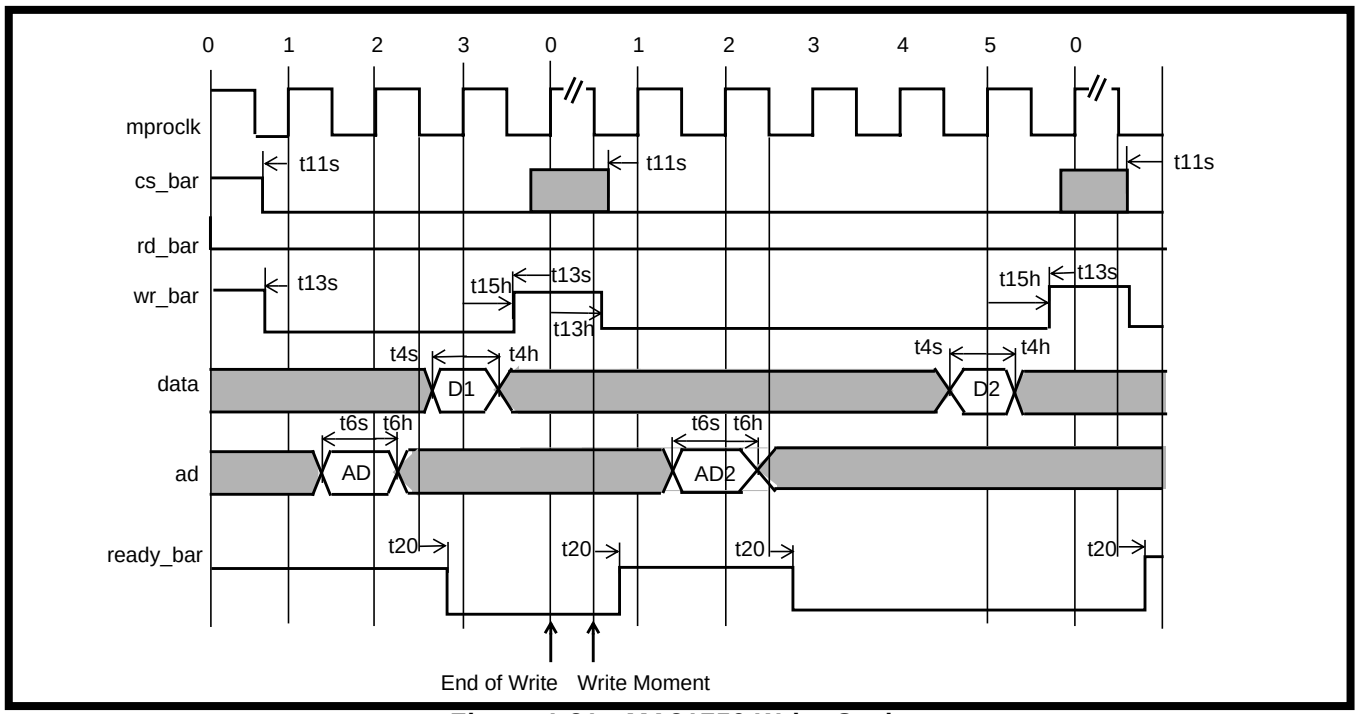


Figure 1.31 - MA31750 Write Cycle

In contrast with the write cycle, the **ad** input bus is not latched and should remain unchanged during the read cycle, as shown in figure 1.32.

If the **rd_bar**, **wr_bar** and **cs_bar** inputs are all asserted at the rising **mproclk** edge, an illegal cycle is detected and the following actions are performed:

- the **data** bus is immediately set to a high-impedance state;
- the current cycle is interrupted;
- the LTMS waits until the **rd_bar** and **wr_bar** inputs are sampled deasserted on the rising **mproclk** edge.

1.11.5 Minimum Duration Between Cycles

For read and write operations that control certain actions of the LTMS, minimum delays have to be taken into account when accessing the same register or facility twice subsequently. No restrictions exist for access to different registers or facilities.

The minimum delay between:

- 2 subsequent write cycles to the LTMS control register (address 27),
- 2 subsequent write cycles to the highest address of the alarm facility (address 15),
- 2 subsequent read cycles of the highest address of the time stamp facility (address 20),
- 2 subsequent write cycles to the same address in the CTMS message register (addresses 1 to 9),
- 2 subsequent read cycles to the same address in the CTMS message register (addresses 1 to 9),
- 2 subsequent write cycles to the time counter of the stopwatch facility (addresses 21 to 23),
- 2 subsequent write cycles to the event counter of the stopwatch facility (addresses 24 to 26),

should not be less than two **busclk** periods when **auxtal**=0.

When **auxtal**=1, the minimum delay should not be less than two **xtal1** periods between:

- 2 subsequent write cycles to the highest address of the alarm register (address 15),
- 2 subsequent write cycles to the same address of the CTMS message register (addresses 1 to 9),
- 2 subsequent read cycles to the same address in the CTMS message register (addresses 1 to 9),

No read operation is allowed to the CTMS message register between the write of the last CTMS message field (addresses 3, 5, or 9) and 2.5 **busclk** periods later if **auxtal**=0. If **auxtal**=1, this interval is equal to 11 **xtal1** periods.

Read operations to the CTMS message register before writing the highest field of the CTMS message, result in reading values corresponding to previously fully written CTMS message, and not to the CTMS message being currently written.

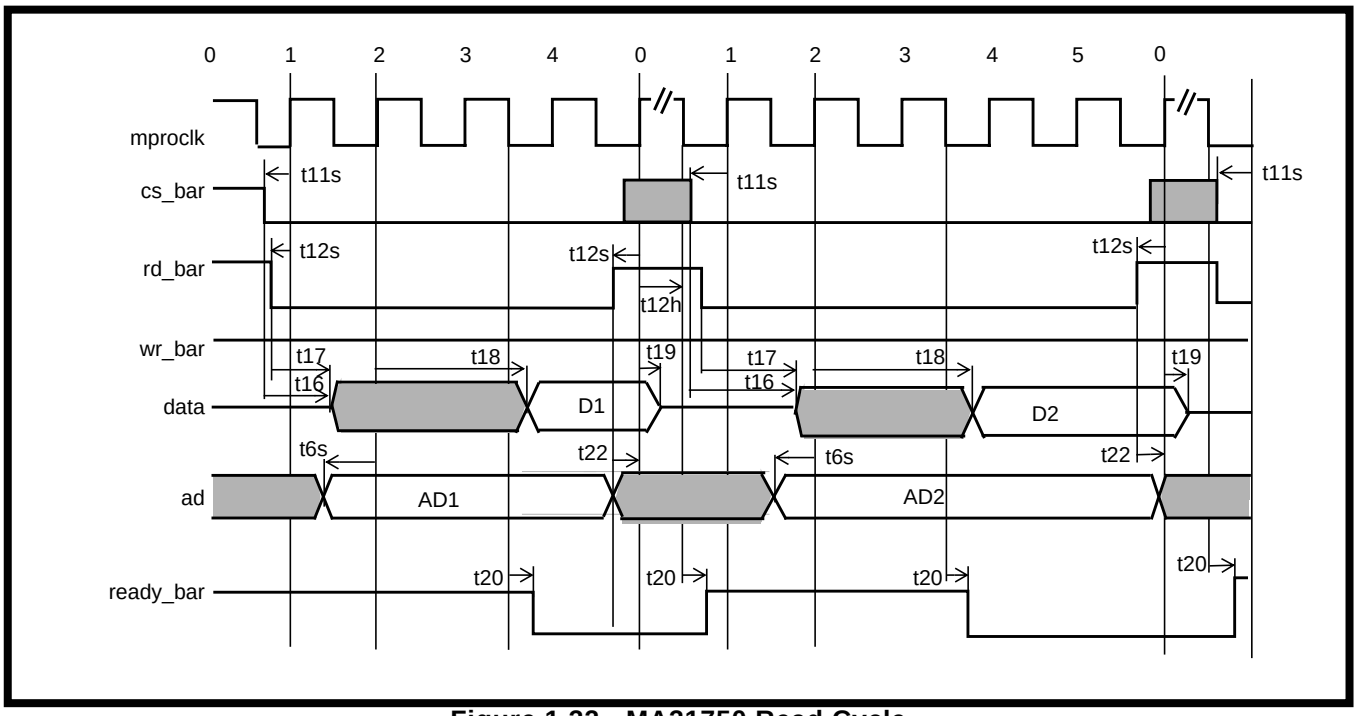


Figure 1.32 - MA31750 Read Cycle

1.12 LTMS Reset

1.12.1 State After Reset

The **data** bus of the parallel microprocessor interface is held in a high impedance state during and after external reset. The **ready_bar**, **extcount** and **extclk** outputs are asserted after external reset. **extclk** is silent (equal to asserted) during reset. If **auxtal**=0, the **extcount** output is asserted during reset. If **auxtal**=1, the **extcount** output is deasserted during reset. The **pfgphout**, **etalrm**, **pfgwave**, **tvld**, **extshten, extdata**, **extetld**, **pfgwave** and **pfgphout** outputs are deasserted after external reset. The **extdemux** outputs are 11 after external reset in serial operation. In parallel or stand alone operation their values after reset are 00.

The value after reset of the **dphase** and **meanfreq** outputs depends on the value selected for the **auxtal** input. When **auxtal**=1, **dphase** and **meanfreq** are both deasserted. When **auxtal**=0, both outputs change each time the **busclk** input changes, even during external reset.

The value of the **window** output after external reset is dependent on the operation mode. In serial operation, **window** is deasserted. In parallel or stand alone operation, it is asserted.

The **etalrm** output and LTMS control register *etalrmbit* flag are disabled until the first predefined alarm time code is written. In serial and parallel operation the alarm clock is disabled until the second LTMS synchronisation.

The value of the LTMS ET counter after reset is all ones. The alarm register, waveform generator counters, CTMS status register, stopwatch facility counters, CTMS message register and bits 15 to 1 of the LTMS control register contain zeros after reset. Bit 0 of the LTMS control register, the *windowbit* flag, then is set to a value dependent on the operation mode selection, similar to the **window** output. The initial value of *windowbit* flag is 1 if the LTMS is configured for parallel or stand alone operation. If the LTMS is in serial operation, the initial value is 0. The stopwatch counters are disabled until a value has been written in either one of them. Initial values of the LTMS status register after reset are as shown in figure 1.33.

The initial value of the *free* and *wasfree* flags is equal to the value of the **auxtal** input. The initial value of the *etcto* flag is equal to the value of the **ctmsg** input.

The time stamp facility contains zeros after reset. It is immediately ready to serve the first event on the **etstrb** input or the *etstrbbit* flag.

The *swselbit* in the LTMS control register is cleared at reset.

1.12.2 Reset Constraints

The values of the **dplldec** and **dpllinc** inputs have to be fixed before and during reset (cf. table 3.5 for T_{dph}, the hold time after reset). If the value of the **dplldec** or the **dpllinc** pin is changed for the purpose of selecting the internal or external DPLL, the LTMS has to be reset thereafter.

The LTMS has to be reset after any of the following inputs are changed: **etthr[1:0]**, **mproc**, **gothr[1:0]**, **ctmsg**, **ser**, **auxtal**, **testaddr[1:0]** and **pfgmode**.

1.12.3 Time Synchronisation and Validity After Reset

The influence of the first few CTMS messages after a reset on the validity of the time information in the LTMS is depicted in figure 1.34. The message transmitted during the reset is incomplete and is not used by the LTMS for synchronisation. The first complete message is used for initialising the LTMS ET counter. Only after the reception of the second complete message is the LTMS able to check the ET coarse time out and the ET threshold time out during the subsequent LTMS synchronisation.

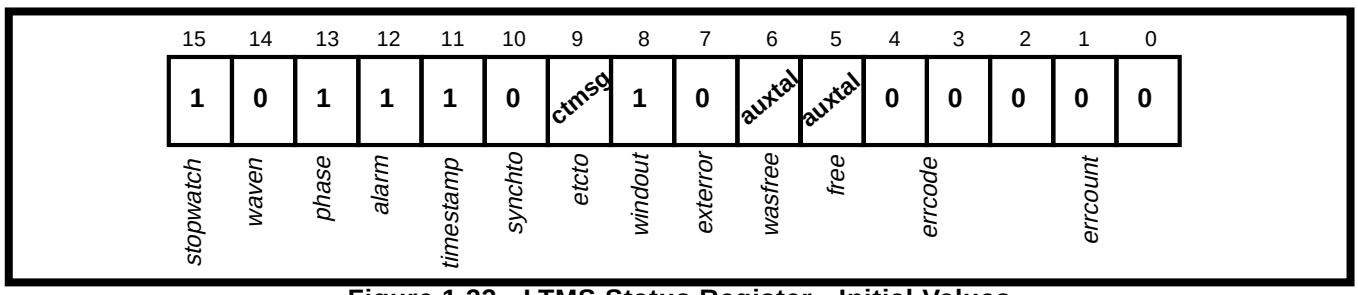


Figure 1.33 - LTMS Status Register - Initial Values

In serial and parallel operations, the time valid (cf. section 1.8.7) generation is enabled at the second LTMS synchronisation after an external reset, i.e. after the first LTMS synchronisation during which an ET threshold check can be performed. Before that moment the **tvld** output and the LTMS control register *tvldbit* flag are kept at 0.

In stand alone operation, the time valid generation is immediately enabled after an external reset.

1.12.4 Time Facilities Availability After Reset

In serial and parallel operation, when the first valid CTMS message has been received, the message is used for LTMS synchronisation. The the alarm, waveform generator and pulse generator outputs can only be asserted after the second LTMS synchronisation. These outputs stay deasserted until the second LTMS synchronisation.

In serial and parallel operation, the *alarm* flag, which is set to 1 by an external reset, is cleared on the second LTMS synchronisation (cf. figure 1.34). From then on the alarm clock facility is enabled.

In stand alone operation the *alarm* flag is cleared the first time an alarm time is written into the alarm register after an external reset. From then on the alarm clock facility is enabled (cf. figure 1.34).

The waveform generator facility is enabled as soon as the first set of required waveform generator parameters has been provided.

Changes on the **pfgwave** output and the LTMS control register *pfgwavebit* flag are disabled until, at least, the second LTMS Synchronisation in serial and parallel operation to avoid generation of erroneous waveforms. The waveform generator facility is however able to acquire its parameters possibly available in the CTMS messages preceding the second LTMS synchronisation.

In stand alone operation, the waveform generator facility is disabled until the reception of the first set of waveform parameters after reset. An external reset stops the waveform generator facility. It remains stopped as long as its output is disabled (cf. figure 1.34).

The value of the *phase* flag is 1 after external reset. It can only change on and after the second LTMS synchronisation. On the second LTMS synchronisation after an external reset, the *phase* flag gets the value of the *pulse* flag in the CTMS message, loaded at the first LTMS synchronisation.

Pulses on the **pfgphout** output are disabled until at least the second LTMS synchronisation in serial and parallel operations to avoid generation of erroneous pulses (cf. figure 1.34). The pulse generator facility is however able to acquire its parameters possibly available in the CTMS messages preceding the second LTMS synchronisation. This means that it is possible to generate a pulse on the **pfgphout** output from t=T+2 on in serial and parallel operation. The *phase* flag in the status register is set to 1.

In stand alone operation the pulse generator facility is disabled until the reception of the first pulse generator parameters (cf. figure 1.34).

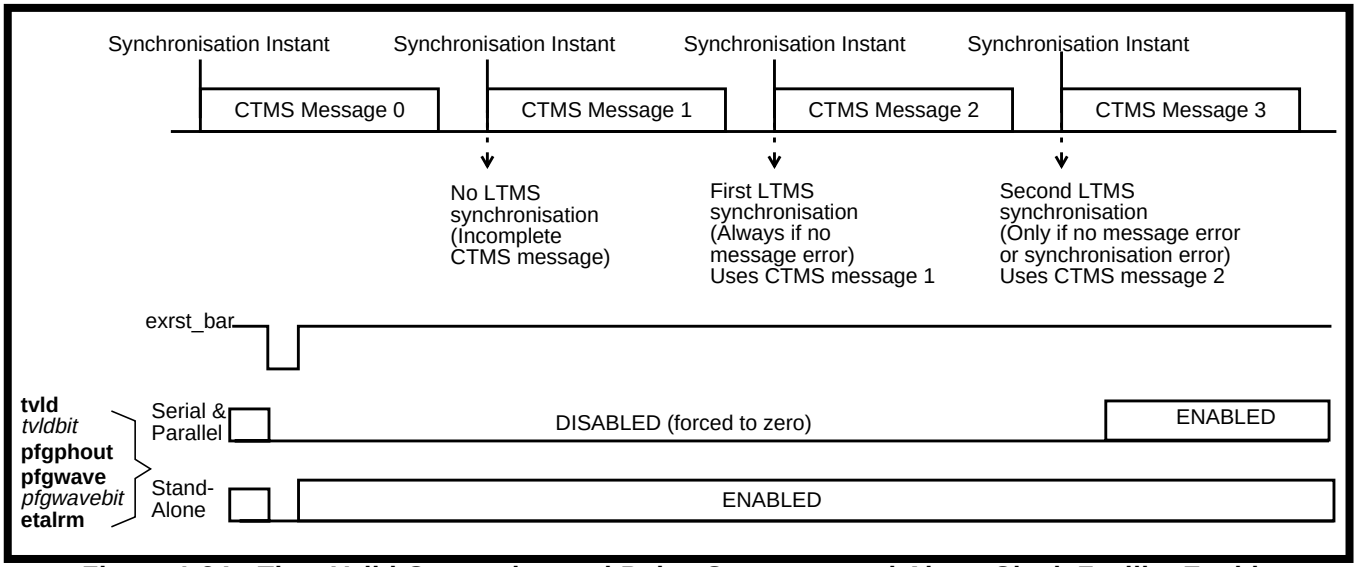


Figure 1.34 - Time Valid Generation and Pulse Generator and Alarm Clock Facility Enable

1.13 LTMS Time Accuracy

Any differences between the ideal time events should take place (T_{ref}) and the actual time they are effectively handled by the LTMS place (T_{LTMS}) are called Accuracy Errors. The following equality applies: $T_{ref} = T_{LTMS} + AccErr + T_{LTMS\text{-}output\text{-}delay.}$

The values are calculated hereafter without taking into account the output delays ($T_{LTMS\text{-}output\text{-}delay}$) of the signals (cf. table 3.5), which might be considered when high accuracy is required. In case actions are started using the LTMS control register, all timing in this section has to be referred to the write moment in figures 1.29 and 1.31.

These following definitions are used in the rest of this document:

- T_x Crystal oscillator period (**xtal1**)
- T_m Microprocessor clock period (**mproclk**)
- T_4 Ideal 2^{22} Hz period, $T_4 \sim 4 * T_x$
- T_b Bus clock period (a multiple of T_4)
- T_{16} Ideal 2^{24} Hz period
- T General variable, expressed in multiples of T_4
- bd Difference between the number of bits of the ET counter and the monotonous ET counter.
- k Phase reference period to period of clock driving ET counter period ratio (minimum 2).
- k_{max} k value when $T_b \in [(k_{max}\text{-}1)*T_m, k_{max}*T_m]$

1.13.1 Internal Time Accuracy

For the LTMS ET counter, the internal time accuracy is the phase difference between the incoming external phase references and the LTMS ET counter phase. The time accuracy is dependent on the time validity as specified in table 1.14.

1.13.2 External Time Accuracy

Any external ET counter is clocked with the same **extcount** signal as the internal LTMS ET counter, having identical accuracy.

1.13.3 Time Stamp Facility Accuracy

For the time stamp facility, T_{ref} is the asynchronous strobe time and T_{LTMS} the time read via the parallel microprocessor interface after the time stamp has been performed (cf. table 1.15). The overall time stamp accuracy is a combination of the facility specific accuracy and the internal time accuracy.

1.13.4 Alarm Clock Facility Accuracy

For the alarm clock facility, T_{ref} is the time at which the alarm should occur as a function of the ET counter. T_{LTMS} is the actual time the alarm occurs. The accuracy error of the alarm clock facility is zero.

tvld	auxtal	Phase reference frequency	Minimum	Maximum
1	1	2^{22} Hz	$-T_x$	$+3T_x$
1	1	$\neq 2^{22}$ Hz	$-T_x$	$+2T_x$
1	0	all	0	0
0	1	all	$-(2^{bd}\text{-}1)4T_x$	$(2^{bd}+1)4T_x$
0	0	all	$-(2^{bd}\text{-}1)T_b$	$(2^{bd}+1)T_b$

Table 1.14 - Internal Time Accuracy

tvld	auxtal	Phase reference frequency	Minimum	Maximum
1	1	$= 2^{22}$ Hz	$-T_4$	$+2T_x$
1	1	$\neq 2^{22}$ Hz	$(4k\text{-}3)T_x\text{-}kT_4$	$(4k\text{-}2)T_x\text{-}(k\text{-}1)T_4$
1	0	all	$-3T_m$	$(k_{max}\text{-}2)T_m$
0	1	$= 2^{22}$ Hz	$-2^{bd}T_4$	$2T_x\text{-}2^{bd}T_4$
0	1	$\neq 2^{22}$ Hz	$(4k\text{-}3)T_x\text{-}(k+2^{bd}\text{-}1)T_4$	$(4k\text{-}2)T_x\text{-}(1\text{-}k+2^{bd})T_4$
0	0	all	$-3T_m\text{-}(2^{bd}\text{-}1)T_b$	$(k_{max}\text{-}2)T_m+2^{bd}T_b$

Table 1.15 - Time Stamp Facility Specific Accuracy

The overall LTMS alarm clock facility accuracy is hence equal to the internal time accuracy.

1.13.5 Pulse Generator Facility Accuracy

For the pulse generator facility, T_{ref} is the time at which the pulse should be generated as a function of the ET counter. T_{LTMS} is the actual time the pulse is generated. The pulse generator facility specific accuracy error is zero except for a pulse that is delayed because it would otherwise be stuck together with the previous one. The facility specific accuracy error for this pulse is one clock period of the local ET counter (i.e. 3, 4 or 5 T_x when **auxtal**=1, T_b when **auxtal**=0).

The overall LTMS pulse generator accuracy is a combination of this facility specific accuracy and the internal time accuracy. Keep in mind that the pulse generator uses the monotonous ET counter.

1.13.6 Waveform Generator Facility Accuracy

The waveform generator facility adjusts its phase to the rising edges of the **pfgphin** input. Ideally the phases of the waveform generated on the **pfgphout** output should match those on the **pfgphin** input. The ET counter and the **pfgphin** input are however not phase-related, hence the resulting accuracy is less than the phase matching of the ET counter and the external phase references. The accuracy of the waveform generator facility is only related to the accuracy of the phase of the clock driving the ET counter (and not to the ET counter value). It differs for the first and the second edge of the generated Waveform (cf. figure 1.23 and table 1.16). On the average, it takes about 6 **xtal1** periods in all cases before phase adjustment of the waveform is performed.

1.13.7 Programmable Frequency Generator Accuracy

The combination of the pulse and waveform generator facilities requires the **pfgphout** output to be connected to the **pfgphin** input (cf. figure 1.25). The accuracy error in this case is the delay from ET counter fine time equality to the moment of the phase update of the generated waveform. The same accuracy holds for both edges of the waveform (cf. table 1.17). The overall accuracy of the programmable frequency generator is a combination of its specific accuracy and the internal time accuracy.

1.13.8 Stopwatch Facility Accuracy

Due to synchronisations of stopwatch facility inputs, it also has an accuracy error for both measurement modes. In the event measurement mode, the accuracy error is not a time, but an amount of events (cf. table 1.18).

tvld	auxtal	Minimum	Maximum
1	1	$-1T_x$	$6T_x$
1/0	0	0	$1T_b$
0	1	Accuracy Error of very first second edge: minimum 0, maximum $4T_x$	

Table 1.16 - Waveform Generator Facility Accuracy

auxtal	Minimum	Maximum
1	0	T_4
0	T_b	

Table 1.17 - Programmable Frequency Generator Specific Accuracy

auxtal	Event measurement		Time measurement	
	Minimum	Maximum	Minimum	Maximum
0	0 events	2 events	$-T_b$	T_b
1	0 events	3 events	$-T_x$	T_x

Table 1.18 - Stopwatch Facility Accuracy

Interface and Signal Description

The LTMS has 35 inputs, 17 outputs and 16 bidirectionals, all CMOS signals. Together with 16 power supply pins, it gives a total pin count of 84. Names ending on *_bar* denote active low signals.

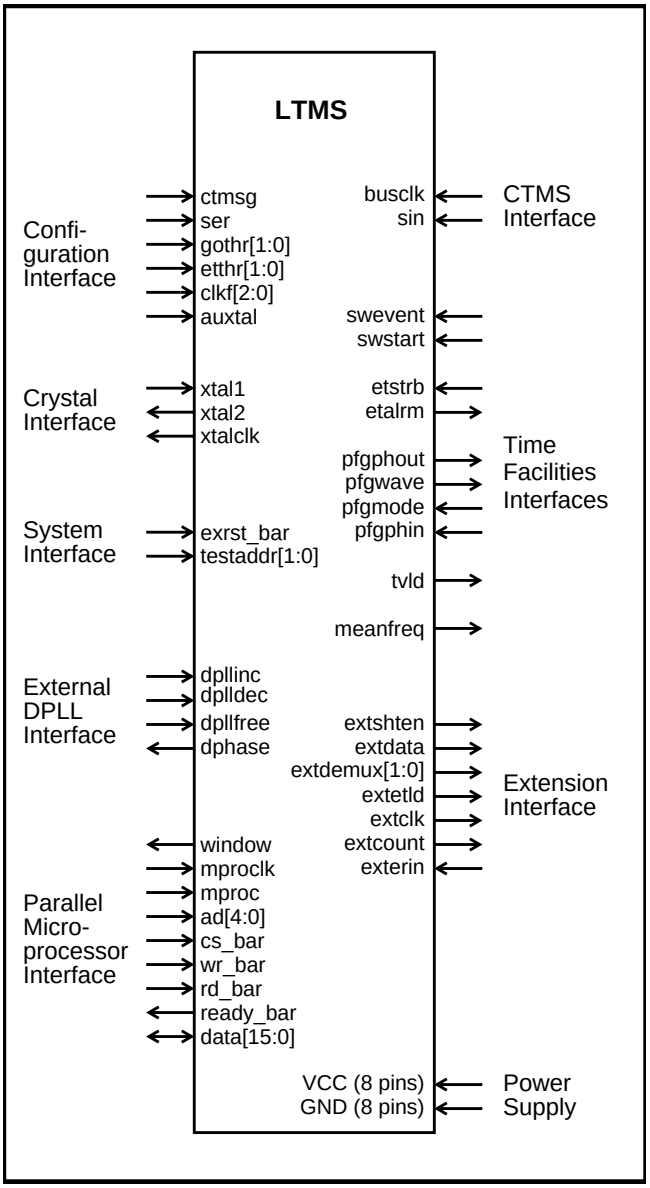


Figure 2.1 - Signal overview

2.1 Configuration Interface

ctmsg *input* CTMS Message Selection

Use the LTMS in connection with a CTMS which sends CTMS messages at regular intervals in serial and parallel operation (**ctmsg**=1); or use it in stand alone operation (**ctmsg**=0) with optional messages.

ser *input* Serial or Parallel Mode

When **ctmsg**=1, the LTMS is in serial operation if **ser**=1 or in parallel operation if **ser**=0. The **ser** input is not used when **ctmsg**=0, but should be tied to a defined logical level.

gothr[1:0] *input* Go-threshold Selectors

Inputs to select the go-threshold (cf. table 1.5). The go-threshold is the maximum amount of accumulated errors that have to occur before an LTMS synchronisation is forced, in serial and parallel operations. Go-threshold selection has no impact in stand alone operation.

etthr[1:0] *input* ET Threshold Selectors

Inputs selecting the ET threshold (cf. table 1.4).

clkf[2:0] *input* Bus Clock Frequency Selectors

Bus clock frequency selection (cf. table 1.1).

auxtal *input* Auxiliary Clock Selection

Input to select the use of the LTMS with an auxiliary 2^{24} Hz clock signal (**auxtal**=1) or directly with the bus clock (**auxtal**=0). If **auxtal**=0, the DPLL is disabled.

2.3 Crystal Interface

xtal1 *input* Crystal Input

Input to which a 2^{24} Hz crystal or external clock should be connected.

xtal2 *output* Crystal Output

Output to which the 2^{24} Hz crystal should be connected (cf. figure 1.18).

xtalclk *output* Crystal clock output

The 2^{24} Hz clock generated by the crystal oscillator or the external 2^{24} Hz clock, is available on this output for external use.

2.2 System Interface

exrst_bar *input* External Reset

Active low external reset, used to initialise the LTMS.

testaddr[1:0] *input* Test Address Bus

These inputs are used for test purposes. They both have to be tied to logical 0 in normal operation.

2.4 External DPLL Interface

dpllinc *input* DPLL Increment

At LTMS reset, this input allows the selection of an external DPLL or not. If an external DPLL is selected, the **dpllinc** input is used to increase the frequency of the LTMS DPLL output. (cf. table 1.6).

dplldec *input* DPLL Decrement

At LTMS reset, this input allows the selection of an external DPLL or not. If an external DPLL is selected, the **dplldec** input is used to reduce the frequency of the LTMS DPLL output (cf. table 1.6)

dpllfree *input* DPLL Free-wheeling

If an external DPLL is used, this input is used to control the *free* and *wasfree* flags of the LTMS status registers. If **dpllfree**=1, the *free* and *wasfree* flags will both be set. If **dpllfree** becomes 0, only the *free* flag will be cleared.

dphase *output* DPLL Phase Reference

Each phase reference edge received by the LTMS is provided on this output.

2.5 Parallel Microprocessor Interface

window *output* Window

In parallel and stand alone operation, the CTMS messages should be sent using the parallel microprocessor interface while **window**=1. In serial operation, **window** is set as soon as a new CTMS message has been acquired, allowing the user to read the CTMS message register; when a new CTMS message starts to be acquired, **window** is cleared.

mproclk *input* Microprocessor Clock

This clock has to be available in all operational modes. When **auxtal**=0 the frequency of the microprocessor clock has to be at least twice the bus clock frequency. When **auxtal**=1, the frequency of the microprocessor clock has to be at most equal to the **xtal1** frequency (2^{24} Hz).

mproc *input* Microprocessor Selection

When **mproc**=1 the MA31750 mode is selected. When **mproc**=0 the ERC32 mode is selected.

ad[4:0] *input* Address Bus

Address; **ad[4]** is the most significant bit; **ad[0]** is the least significant.

cs_bar *input* Chip Select

Chip Select, active low.

wr_bar *input* Write Strobe

Write strobe, active low.

rd_bar *input* Read Strobe

Read Strobe, active low.

ready_bar *output* Ready

When **mproc**=1, **ready_bar** is an active low output for inserting wait states (default is deasserted). When **mproc**=0, **ready_bar** is deasserted.

data[15:0] *in/out* Data Bus

Bidirectional data; **data[15]** is the most significant; **data[0]** is the least significant.

2.6 CTMS Interface

busclk *input* Bus Clock

The rising **busclk** edges are used as phase references in stand alone operation (**ctmsg**=0), parallel operation (**ctmsg**=1, **ser**=0) and in SERBUS operation (**ctmsg**=1, **ser**=1, **auxtal**=0). This input should be strapped to logical 0 in SERAUX operation. In SERBUS, PARBUS and SALBUS mode, the duty cycle of the clock signal on **busclk** should be between 40% and 60%.

sin *input* Serial Input

If **ctmsg**=1 and **ser**=1 (serial operation) the **sin** input carries the Manchester coded CTMS message. If **ctmsg**=0 (stand alone operation) or **ser**=0 (parallel operation), the first rising **busclk** edge while **sin**=1 is the LTMS synchronisation marker.

2.7 Time Facilities Interfaces

Some of the following inputs and outputs can also be controlled and observed via the LTMS registers.

swevent *input* Stopwatch Event

Rising edges on this input are stop watch events by definition.

swstart *input* Stopwatch Start/Hold

If the *swselbit* flag in the LTMS control register is zero (default value after start-up), the stopwatch time and event counters are enabled, as long as **swstart**=1. When **swstart** is cleared to 0 the stopwatch counters are frozen. When **swstart** is set to 1, the stopwatch counters are enabled. If *swselbit*=1, the **swstart** input should be held to a logical value.

etalrm *output* Elapsed Time Alarm

The **etalrm** output is set when the time in the ET counter reaches the user defined alarm time. When a new alarm time is loaded, the **etalrm** output is cleared.

etstrb *input* Elapsed Time Strobe

Input used to latch the current ET counter and status flags values in the time stamp facility.

pfgphout *output* Pulse Generator Output

If a pulse field is available in the CTMS message, a pulse is generated on **pfgphout** when an equality is detected between the pulse information and the ET counter value or as soon as the ET counter has overrun the pulse information.

pfgwave *output* Waveform Generator Output

The waveform generated by the waveform generator is available on this output.

pfgphin *input* Waveform Generator Input

The waveform generator facility starts at the detection of the first rising edge on the **pfgphin** input, following the reception of a new parameter set. The edge is called the start phase. Subsequent rising **pfgphin** edges are used to immediately correct the waveform phase.

pfgmode *input* Waveform Generator Mode

If **pfgmode**=1, the user controls the waveform generator. If **pfgmode**=0, the waveform generator is controlled by a CTMS.

tvld *output* Time Valid

This output provides a summary of all the status and error information in the LTMS status register. If **tvld**=1, there are no errors indicated in the status flags (bit positions 3 to 7 and 9 to 10). If **tvld**=0, at least one error has occurred.

meanfreq *output* Mean Frequency

The internal DPLL provides a regenerated phase reference signal with a mean frequency similar to the one of the phase references.

2.8 Extension Interface

extshten *output* External Shift Enable

When **extshten**=1, data are shifted out on the **extdata** output.

extdata *output* External Data

The received CTMS messages are shifted out on this output. In serial operation, the mission parameters and inter-message symbols are also shifted out. Parity bits are stripped by the LTMS and are not shifted out on the **extdata** output.

extetld *output* External ET Load

This output informs when the ET coarse time code transmitted via the extension interface, has to be loaded in an external ET counter, on a rising edge of the **extcount** output.

extdemux[1:0] *output* External Demultiplexer

These outputs inform about the nature of the data being shifted out on the **extdata** output (cf. table 1.3)

extclk *output* External DPLL clock

External 2^{24} Hz clock output if an auxiliary clock (**auxtal**=1) is used. If **auxtal**=0, the signal on the **busclk** input is available on this pin.

extcount *output* External ET Counter clock

This is the clock driving the ET counter. If **auxtal**=1, the regenerated 2^{22} Hz (mean) clock (i.e. the LTMS DPLL output) is available.

If **auxtal**=0, **extcount** provides a copy of **busclk**. This clock should be used to drive any external ET counter, and to sample data shifted out on **extdata** while **extshten**=1.

exterin *input* External Error

This input informs the LTMS about the result of additional external error checks. If an external error is detected, **exterin** should be asserted to 1. Otherwise it is should be deasserted to 0.

Electrical and Mechanical Data

The LTMS has been designed for the full military temperature ranges in the MITEL SOS5 1.25 μm CMOS/SOS process. The die has an area of 121 mm^2 and contains 108k transistors.

The LTMS is a fully static design.

3.1 Absolute Maximum Ratings

Storage temperature	-65 ˚C to +150 ˚C
Operating ambient temperature	-55 ˚C to +125 ˚C
Supply voltage, V_{DD}	0 V to 7 V
Input voltage	V_{SS} - 0.5 V to V_{DD} + 0.5 V
DC input current (any pin)	10 mA
Power dissipation	300 mW_{DC}
Output dissipation	50 mW_{DC}/output
ESD minimum critical path failure voltage	500 V
Total dose tolerance	> 100 kRad (Si)
Latch-up	Immune

Table 3.1 - Absolute Maximum Ratings

3.2 DC Specifications

Parameter	Symbol	Condition	Min	Max	Unit
I_{DD} quicscent current	I_{DDQ}	V_{DD} = 5.5V, V_{SS} = 0V, I_O = 0mA		TBD	mA
I_{DD} operating	I_{DDop}	**xtal1** @ 2^{24} Hz		TBD	mA
Input leakage current, High level	I_{LH}	V_{DD} = 5.5V, V_{SS} = 0V, V_{IN} = 5.5V		10	μA
Input leakage current, Low level	I_{IL}	V_{DD} = 5.5V, V_{SS} = 0V, V_{IN} = 0V		10	μA
Input Voltage, High level	V_{IH}	V_{DD} = 4.5V, V_{SS} = 0V	3.2		V
Input Voltage, Low level	V_{IL}	V_{DD} = 4.5V, V_{SS} = 0V		1.3	V
Output Voltage, High level, (all outputs except **xtal2**)	V_{OHI}	V_{DD} = 4.5V, V_{SS} = 0V, I_O = -4mA	4.0		V
Output Voltage, Low level, (all outputs except **xtal2**)	V_{OLI}	V_{DD} = 4.5V, V_{SS} = 0V, I_O = 4mA		0.4	V
Output Voltage, High Level, (**xtal2**)	V_{OH2}	V_{DD} = 4.5V, V_{SS} = 0V, I_O = -0.5mA	4.0		V
Output Voltage, Low level, (**xtal2**)	V_{OL2}	V_{DD} = 4.5V, V_{SS} = 0V, I_O = 0.5mA		0.4	V
Tri-state current, High level, (only for bidirectional pins)	I_{OZH}	V_{DD} = 5.5V, V_{SS} = 0 V, V_O = 5.5V		100	μA
Tri-state current, Low level, (only for bidirectional pins)	I_{OZL}	V_{DD} = 5.5V, V_{SS} = 0V, V_O = 0V		-100	μA

Table 3.2 - DC Specifications

3.3 AC Specifications

All timing values are valid for the full operational temperature range, radiation range and power supply range, with 100pF output load. Minimum values are 0.4 times the Typical values. Maximum values are 2.7 times the Typical values. The definitions at the beginning of section 1.13 apply.

Symbol	Parameter	Remark	Max.	Units
t4s	Setup **data** to **mproclk** ↑	m	6	ns
t4h	Hold **data** to **mproclk**↑	m	3	ns
t6s	Setup **ad** to **mproclk** ↑	m	0	ns
t6h	Hold **ad** to **mproclk** ↑	m	11	ns
t18	Data valid after **mproclk** ↑	an	72	ns
if **mproc** = 0				
t1s	Setup **cs_bar** to **mproclk** ↓	m	1	ns
t1h	Hold **cs_bar** to **mproclk** ↓	m	9	ns
t2s	Setup **rd_bar** to **mproclk** ↓	m	2	ns
t2h	Hold **rd_bar** to **mproclk** ↓	m	8	ns
t3s	Setup **wr_bar** to **mproclk** ↓	m	1	ns
t3h	Hold **wr_bar** to **mproclk**↓	m	9	ns
t5s	Setup **cs_bar/wr_bar** to **mproclk** ↑	m	13	ns
t5h	Hold **cs_bar/wr_bar** to **mproclk** ↑	m	7	ns
t7	Delay enable **data** after **mproclk** ↓	m	58	ns
t9	Data valid after deassertion of **rd_bar** or **cs_bar**	m	59	ns
t10	Hold **ad** after deassertion of **rd_bar** or **cs_bar**	m	19 (2)	ns
if **mproc** = 1				
t11s	Setup **cs_bar** to **mproclk** ↑	m	9	ns
t12s	Setup **rd_bar** to **mproclk** ↑	m	9	ns
t12h	Hold **rd_bar** to **mproclk** ↑	m	12	ns
t13s	Setup **wr_bar** to **mproclk** ↑	m	9	ns
t13h	Hold **wr_bar** ↓ to **mproclk** ↑	m	3	ns
t15h	Hold **wr_bar** ↑ to **mproclk** ↑	m	7	ns
t16	Delay enable **data** after assertion of **cs_bar**	m	50	ns
t17	Delay enable **data** after assertion of **rd_bar**	m	50	ns
t19	Data valid after deassertion of **rd_bar**	m	59	ns
t20	**mproclk** ↓ to **ready_bar**	m	36 (1)	ns
t22	Hold **ad** after deassertion of **rd_bar**	m	19 (2)	ns

Table 3.3 - Parallel Microprocessor Interface Timing

(1) If the load on the **ready_bar** output is 50pF, the delay is 28 ns (an).
(2) When the hold time is violated the **data** output may change value before a high impedance state is reached'.
Convention: an = Postlayout analysis, de = derived from auxtal=0, m = measured

Symbol	Parameter	Remark	Max	Units
	xtal1 frequency	2^{24} Hz nominal	18.63	MHz
	busclk frequency		18.63/4	MHz
	mproclk frequency		18.63	MHz

Table 3.4 - Clock Frequencies

Symbol	Parameter	Remark	Max	Units
Tm	**mproclk** ↓ (internal write moment) to **etalrm** ↓	m	77	ns
if **auxtal** = 1				
T_{dpll}	DPLL delay (cf. section 1.9.4)	an	65	ns
Tdph	**dplldec** and **dpllinc** hold after **exrst_bar** ↑	an	Tx + 32	ns
Txx	**xtal1** to **xtalclk**	m	26	ns
Txdpr	**xtal1** ↑ to **meanfreq** ↓↑ (via extclk ↑)	m	80	ns
Txdp	**xtal1** ↑ to **dphase** ↓↑ (via extclk ↑)	m	80	ns
Txwd	**xtal1** ↑ to **window** ↓↑ (via extcount ↑)	m	102	ns
Txal	**xtal1** ↑ to **etalrm**↑ (via extcount ↑)	de	105	ns
Txv	**xtal1** ↑ to **tvld** ↓↑ (via extcount ↑)	de	104	ns
Txph	**xtal1** ↑ to **pfgphout** ↓↑ (via extcount ↑)	de	104	ns
Txwv	**xtal1** ↑ to **pfgwave** ↓↑ (via extcount ↑)	de	104	ns
Txxc	**xtal1** ↑ to **extcount** ↓↑	m	94	ns
Txsh	**xtal1** ↑ to **extshten** ↓↑ (via extcount ↓)	m	118	ns
Txdt	**xtal1** ↑ to **extdata** ↓↑ (via extcount ↓)	m	118	ns
Txdmx	**xtal1** ↑ to **extdemux** ↓↑ (via extcount ↓)	m	118	ns
Txxck	**xtal1** ↑ to **extclk** ↓↑	m	64	ns
Txldf	**xtal1** ↑ to **extetld** ↓ (via extclk ↑)	an	81	ns
Txldr	**xtal1** ↑ to **extetld** ↑ (via extcount ↓)	an	111	ns
Txwait	**exrst_bar** ↑ to application of stimuli to LTMS	an	Tx +Tm	
Tx_{SAext}	Write of last message word to LTMS synchronisation, stand alone operation, external ET counter used.	an	655Tx + 43	ns
$Tx_{SAnoext}$	Write of last message word to LTMS synchronisation, stand alone operation, no external ET counter used.	an	10Tx + 43	ns

Table 3.5 - Output Delays

Convention: an = Postlayout analysis, de = derived from auxtal=0, m = measured

Symbol	Parameter	Remark	Max	Units
if **auxtal** = 0				
Tbdpr	**busclk** to **meanfreq**	m	42	ns
Tbdp	**busclk** to **dphase**	m	42	ns
Tbwd	**busclk** ↑ to **window** ↓↑ (via extcount ↑)	m	74	ns
Tbal	**busclk** ↑ to **etalrm** ↑ (via extcount ↑)	m	75	ns
Tbv	**busclk** ↑ to **tvld** ↓↑ (via extcount ↑)	m	74	ns
Tbph	**busclk** ↑ to **pfgphout** ↓↑ (via extcount ↑)	m	74	ns
Tbwv	**busclk** ↑ to **pfgwave** ↓↑ (via extcount ↑)	m	74	ns
Tbxc	**busclk** to **extcount**	m	62	ns
Tbsh	**busclk** ↓ to **extshten** ↓↑ (via extcount ↓)	m	87	ns
Tbdt	**busclk** ↓ to **extdata** ↓↑ (via extcount ↓)	m	87	ns
Tbdmx	**busclk** ↓ to **extdemux** ↓↑ (via extcount↓)	m	87	ns
Tbxck	**busclk** to **extclk**	m	50	ns
Tbldf	**busclk** ↓ to **extetld** ↓ (via extclk↓)	m	96	ns
Tbldr	**busclk** ↓ to **extetld** ↑ (via extcount ↓)	m	96	ns
Tbwait	**exrst_bar** ↑ to application of stimuli to LTMS	an	Tb +Tm	
Tb_{SAext}	Write of last message word to LTMS synchronisation, stand alone operation, external ET counter used.	an	131Tb + 43	ns
$Tb_{SAnoext}$	Write of last message word to LTMS synchronisation, stand alone operation, no external ET counter used.	an	2Tb + 43	ns

Table 3.5 - Output Delays, continued

Convention: an = Postlayout analysis, de = derived from auxtal=0, m = measured

Signal	auxtal=1	auxtal=0
exrst_bar	T_x+T_m (sampled by **xtal1**)	T_b+T_m (sampled by **busclk**)
dpllfree	$5T_x$ (sampled by **xtal1**)	Not Applicable
busclk	T_x (sampled by **xtal1**)	Not Applicable
sin	T_x (sampled by **xtal1**)	T_b (sampled by **busclk**)
etstrb	T_x (sampled by **xtal1**)	T_m (sampled by **mproclk**)
pfgphin	T_x (sampled by **xtal1**)	T_b (sampled by **busclk**)
swevent	T_x (sampled by **xtal1**)	T_b (sampled by **busclk**)
swstart	T_x (sampled by **xtal1**)	T_b (sampled by **busclk**)
exterin	$5T_x$ (sampled by **extcount**)	T_b (sampled by **busclk**)

Table 3.6 - Minimum Pulse Durations

If no external ET counter is used with the LTMS, there has to be a time of at least $T_{SAnoext}$ between the write moment of the last word of the CTMS message and the synchronisation marker (cf. section 1.7.3). If an external ET counter is used, this time becomes T_{saext}.

Reading $13T_X$ (when **auxtal**=1, else $4T_b$ when **auxtal**=0) the LTMS control register after a write moment to the same register, always results in reading the previously written values.

For the stopwatch facility, the time between a transition from 0 to 1 on the *swselbit* flag until a transition from 0 to 1 on the *swstartbit* flag occurs must meet a minimum time constraint.

These transitions may at earliest occur T_b after the write moment of address 23 when **auxtal**=0, or address 26 when **auxtal**=1 at earliest $2T_X$ after the write moment (cf. sections 1.10.6 and 1.13 for definitions).

Symbol	Parameter	Remark	Max.	Units
	Write moment of LTMS control register with a new *exterinbit* flag value to a subsequent read of the same flag with the new value	an	0	ns
if **auxtal** = 1				
	Write moment of last word of past time into alarm cock to - read of the status register *etalrm* flag - read of the LTMS control register *etalrmbit* flag	an	15Tx + 3Tm + 43	ns
	Write moment of LTMS control register write to internal usage of the *exterinbit* value.	an	11Tx + 64	ns
	Change of **exterin** to internal usage	an	11Tx+26	ns
	Write moment of LTMS control register write with a rising edge on the *etstrbbit* flag to setting the *timestamp* flag in the status register when time stamp facility busy	an	4Tx+58	ns
	Change on **swstart**, until being readable in the *swstartbit* flag	an	4Tx+50	ns
	Write moment of LTMS control register write with a rising edge on the *swstartbit* flag to internal use of the value	an	3Tx+43	ns
	Write moment of LTMS control register write with a new *swstartbit* flag value to that value being readable	an	4Tx+93	ns
	Write moment of LTMS control register write to the internal usage of the *swselbit* flag value	an	Tx+43	ns
	Write moment of LTMS control register write with a rising edge on the *sweventbit* flag to its usage in the stopwatch	an	3Tx+43	ns
	Change on **pfgphin**, until being readable in the *pfgphinbit* flag	an	5Tx+18	ns
	Write moment of LTMS control register write with a rising edge on the *pfgphinbit* flag to internal use of the value	an	9Tx+64	ns
	Write moment of LTMS control register write with new value of *pfgphinbit* flag to that value being readable	an	5Tx+58	ns
	Change on **etstrb**, until being readable in the *etstrbbit* flag	an	3Tx	

Table 3.7 -Synchronisation Delays

(1) Changes on **exterin** are not visible via the *exterinbit* flag.

Convention: an = Postlayout analysis, de = derived from auxtal=0, m = measure

Symbol	Parameter	Remark	Max.	Units
if **auxtal** = 0				
	Write moment of last word of past time to - read of the status register *etalrm* flag - read of the LTMS control register *etalrmbit* flag	an	3Tm + 3Tb+43	ns
	Write moment of LTMS control register write to internal usage of the *exterinbit* value.	an	3Tb+43	ns
	Change of **exterin** to internal usage (1)	an	3Tb+5	ns
	Write moment of LTMS control register write with a rising edge on the *etstrbbit* flag to setting the *timestamp* flag in the status register when time stamp facility busy	an	4Tm+ 58	ns
	Change on **swstart**, until being readable in the *swstartbit* flag	an	2.5Tm+Tb+ 18	ns
	Write moment of LTMS control register write with a rising edge on *swstartbit* to internal use of the value	an	2Tb+43	ns
	Write moment of LTMS control register write with a new *swstartbit* flag value to that value being readable	an	2.5Tm+ Tb+ 58	ns
	Write moment of LTMS control register write to the internal usage of the *swselbit* flag value	an	Tb+43	ns
	Write moment of LTMS control register write with a rising edge on the *sweventbit* flag to its usage in the stopwatch	an	2Tb+43	ns
	Change on **pfgphin**, until being readable in the *pfgphinbit* flag	an	2.5Tm+ 2Tb+18	ns
	Write moment of LTMS control register write with a rising edge on the *pfgphinbit* flag to internal use of the value	an	2Tb+72	ns
	Write moment of LTMS control register with new value of pfgphinbit flag to that value being readable	an	2.5Tm+ 2Tb+58	ns
	Change on **etstrb**, until being readable in the *etstrbbit* flag	an	3Tm	

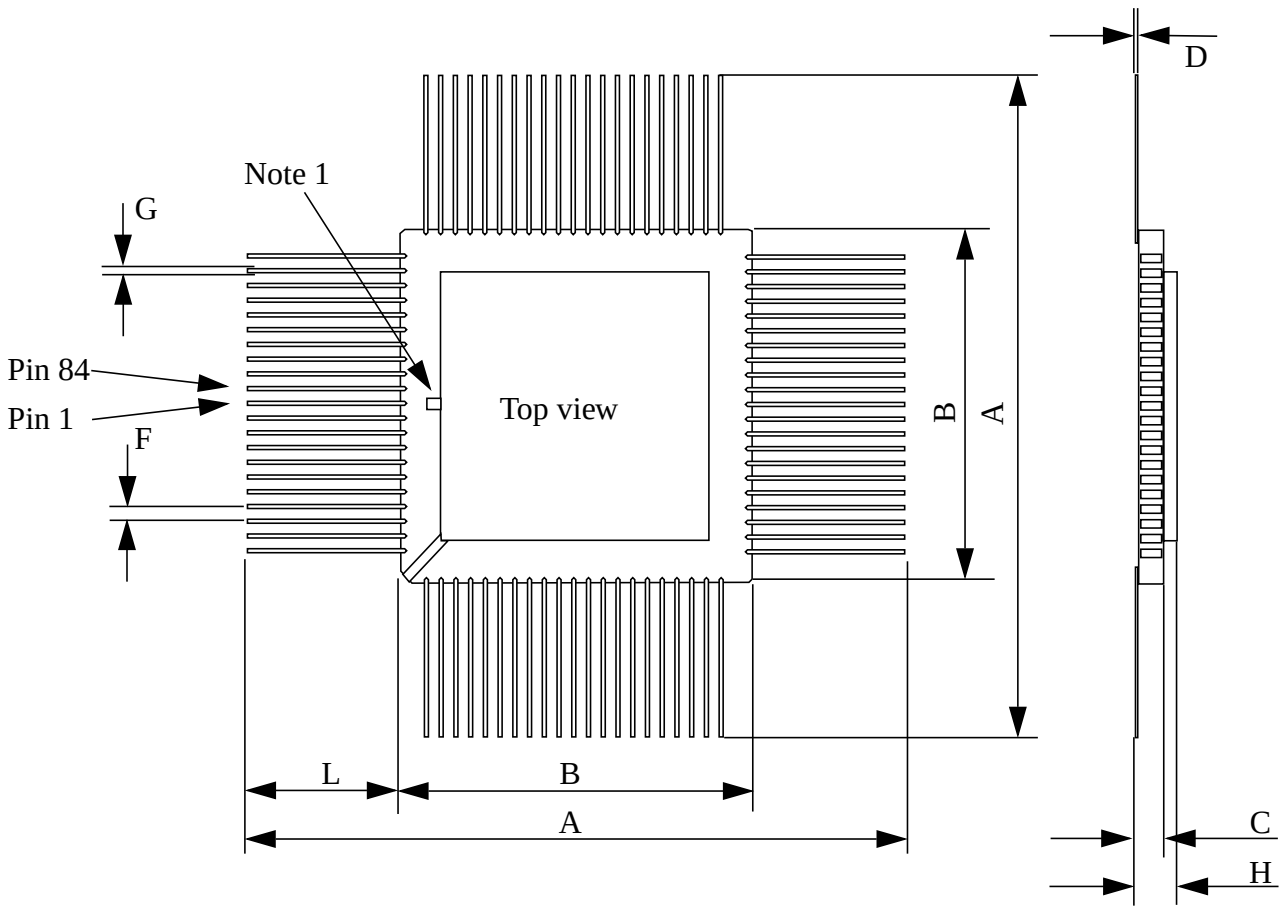
Table 3.7 -Synchronisation Delays, continued

(1) Changes on **exterin** are not visible via the *exterinbit* flag.

Convention: an = Postlayout analysis, de = derived from auxtal=0, m = measure

3.4 Package Dimensions

The LTMS is available in an 84 pin Ceramic Quad Flat Package with 40 mil pin pitch.



Symbol	Millimeters		Notes
	Min	Max	
A	42.64	44.25	
B	23.14	23.75	
C	1.47	1.83	2, 5
D	0.10	0.20	
F	0.94	1.10	3, 4
G	0.17	0.33	2
H	1.80	2.31	
L	9.75	10.25	2

Note 1: Pin no. 1 location
Note 2: All leads
Note 3: 80 spaces for flat package
Note 4: The true position pin spacing is x mm between center lines. Each pin centerline is located within +/- 0.1 mm of its true longitudinal position relative to pin 1 and pin 84.
Note 5: The dimension are measured at the point of exit of the lead from the body.

Figure 3.1 - Dimensions for 40 mil pin pitch Ceramic Quad Flat Pack

Appendix A: Typical Configurations

This appendix describes some typical LTMS applications in the different operational modes. The descriptions also highlight ways to connect the LTMS to various types of data handling systems.

OBDH/4-255 Data Bus Application

One possible way to use the LTMS is to combine it with the ESA OnBoard Data Handling (OBDH) subsystem. The most efficient use of the LTMS is in serial operation (cf. figure A.3).

The system time reference is located in the OBDH Central Data Management Unit (CDMU). The CDMU manages all the traffic on the OBDH 4-255 data bus. Local copies of the CTMS time reference are maintained in the experiment equipment by their LTMS. To keep the different LTMSs synchronised in the system, the CTMS modulates the Broadcast Pulse Bit 0 (BCP0) available in each 32 bit Interrogation to carry the CTMS message. The format of the 32 bit interrogation is shown in figures A.1 and A.2.

The BCP are decoded by all the terminals connected to the OBDH 4-255 data bus. As the CDMU generates a continuous interrogation word flow, BCP0 can be considered as a virtual serial line with a data throughput 32 times slower than the one of the interrogation bus line. To guarantee the distribution of the phase references accurately, the OBDH 4-255 data bus clock is generated by the CTMS. This is totally transparent to the rest of the OBDH traffic. While in the past, the BCP were used to transfer pulses only, this new time distribution scheme uses the BCP more efficiently. The BCP0 carries a synchronisation marker, the CTMS message and phase references. The LTMS is connected to the OBDH 4-255 data bus through a modem. The modem provides a dedicated bus clock output and dedicated broadcasting pulse outputs.

In the frame of the OBDH 4-255 data bus, the selection between SERAUX and SERBUS operation depends on the free-wheeling capability and whether the bus clock has a clocking rate high enough to provide and maintain the local time with the required resolution.

If the free-wheeling capability is not required and if the bus clock rate is high enough (2^{19} Hz), the SERBUS operation is preferred. No auxiliary 2^{24} Hz clock or crystal oscillator is required and the LTMS simply needs access to the bus clock and to the BCP0 output (modulating the CTMS messages).

If free-wheeling capability is required or if the bus clock rate is not high enough, the LTMS is used in SERAUX operation. To generate the auxiliary 2^{24} Hz clock which is required in SERAUX operation, a crystal oscillator is connected to the LTMS if no 2^{24} Hz signal is already available in the experiment equipment.

In both serial modes, the CTMS modulates the virtual BCP0 line to broadcast messages to the LTMSs each second. The synchronisation process is fully supported and completely transparent to the user hosting the LTMS. It is possible to use the LTMS in parallel mode with the OBDH 4-255 data bus, but it is a less efficient approach.

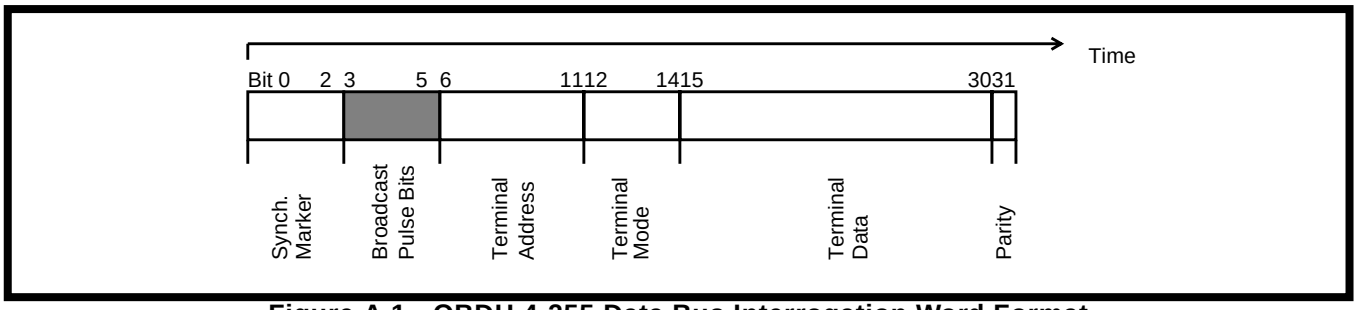


Figure A.1 - OBDH 4-255 Data Bus Interrogation Word Format

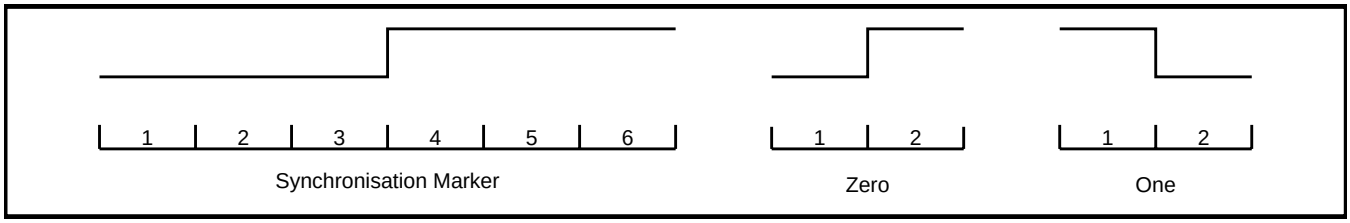


Figure A.2 - OBDH 4-255 Data Bus Interrogations Required For Each Manchester Symbol

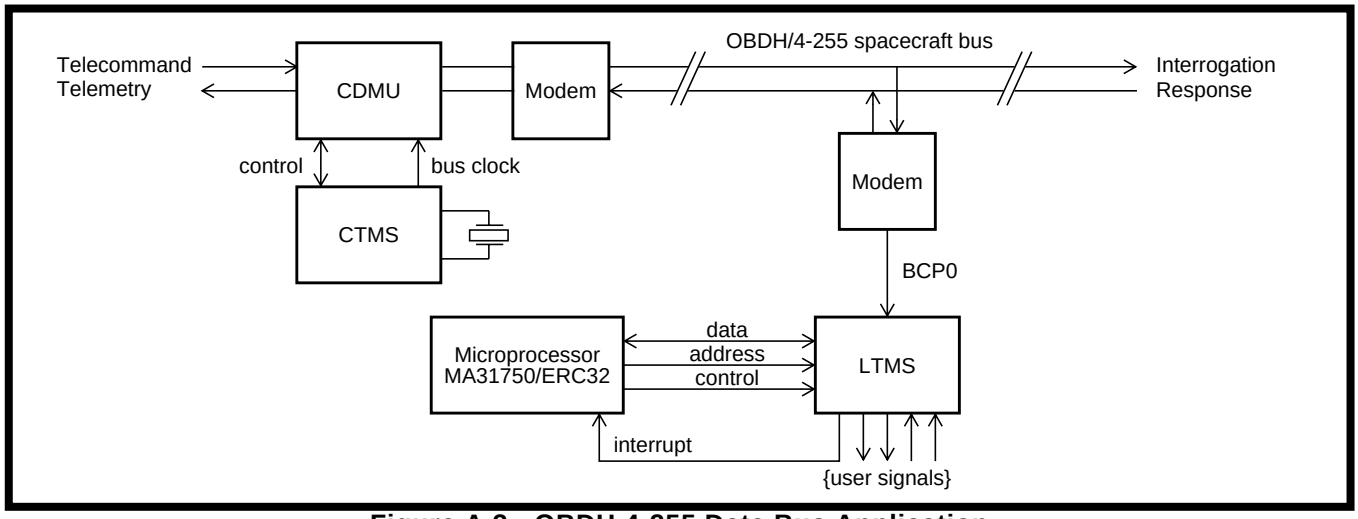


Figure A.3 - OBDH 4-255 Data Bus Application

Time Bus Application

SERAUX operation is suitable for systems where it is not possible to use the existing data bus for distributing the time information. Reasons may be: delays, jitter, traffic load, etc. In such cases, the time distribution has to be performed via an additional bus dedicated to time only. With the LTMS in SERAUX operation, this extra bus is reduced to a single physical line or a twisted shielded pair to reduce electromagnetic coupling problems.

In the time bus application (cf. figure A.4), the LTMS is used in SERAUX operation in order to minimise the additional hardware required. Thanks to the Manchester encoding, one line carries all the required synchronisation information: CTMS message, phase references and synchronisation marker. An auxiliary clock is required locally as the Manchester code used on the time bus does not provide a regular clock signal. The data bus clock cannot be used as this bus is totally independent.

GPS Information Distribution Application

In serial operation, the LTMS can be used to distribute information from a Global Positioning System (GPS) receiver to the various equipment with a minimum of harness as described above. Assuming that the GPS receiver is located next to the CTMS, the GPS information can be fed as mission parameters in the LTMS serial protocol. At the LTMS side, the GPS data are routed to the LTMS extension interface; some external logic is required to store and build a GPS counter and facilities.

The LTMS ET reference and its pulse generator can be used to generate the GPS synchronisation marker indicating when the latest received GPS data becomes valid. To maintain two time references across the system may sound luxurious. However, the ET reference is an autonomous time reference. That autonomy makes that it is not affected by external failures (e.g. GPS signal loss) and can be considered as a robust time scale for operation sequencing. With the LTMS, the same ET time scale is maintained across the whole system while any other synchronisation task are possible as the above local GPS reference maintenance.

Finally, the ET reference uses an unsegmented binary code format which is fully machine oriented. The use and distribution of the GPS information across a system has consequently to be considered as an add-on, a complement to the LTMS ET reference in order to keep that system synchronised with its environment.

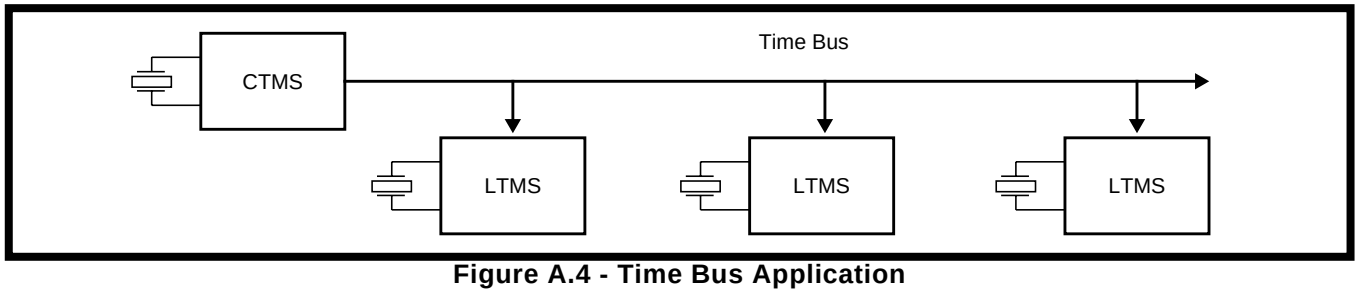


Figure A.4 - Time Bus Application

MIL-1553B Bus Application

Even if the MIL-1553B is asynchronous, the LTMS allows to recover a high accurate time information in the different MIL-1553B terminals (cf. figure A.5). As permanent traffic is not achievable with the MIL-1553B bus and as most of the MIL-1553B terminal chip sets are microprocessor oriented, the LTMS is used in PARAUX operation. PARBUS operation is not possible because the bus has a half duplex nature and a bus clock is not permanently available.

The CTMS has to share the bus with the other services and is preferably located in the MIL-1553B Bus Controller (BC) terminal. The BC is in charge of managing the traffic on the bus, minimising the latency problems. The transmission of the CTMS message is not too difficult to achieve as only one slot for a CTMS message with maximum nine data words has to be guaranteed once every second.

More crucial is the transmission of the phase reference pulses. They have to be sent at accurate instants, because the overall time distribution chain accuracy directly depends on the phase reference accuracy. The phase references are broadcasted via a Mode Command Synchronise signal which has the highest priority. The start of other transfers is disabled a few slots in advance to guarantee that the bus is free when required.

As the emission of the phase references may substantially disturb the rest of the bus traffic, the LTMS should be used with **clkf[2]** equal to one, in order to significantly reduce the required number of phase references per second. There is a price to pay: the local crystal oscillator needs a better stability.

For the synchronisation marker, two possibilities exist. Either this information is linked with the phase reference (by appending a data word to the Mode Command Synchronise for example) or a short message is sent in order to assert **sin** until the reception of the next phase reference. The latter of the two solutions is the preferred one.

At the Remote Terminal side (RT) where the LTMSs are located, one of the 32 sub-addresses is dedicated to the CTMS message as a mailbox. The support of the user microprocessor is required to empty the time mailbox once every second. The user routes the mailbox contents, i.e. the CTMS messages, to the CTMS message register of the LTMS.

The reception of the phase references generates pulses on a dedicated output of the MIL-1553B chip set, connected to the LTMS **busclk** input. For the synchronisation marker, either some logic or a limited involvement of the user microprocessor is required to detect the reception of the relevant MIL-1553 message and to handle the LTMS **sin** input accordingly.

In parallel operation, the LTMS requires some support from the user microprocessor. This support is limited to the transfer of a maximum of two short data packets (CTMS message and synchronisation marker) each second.

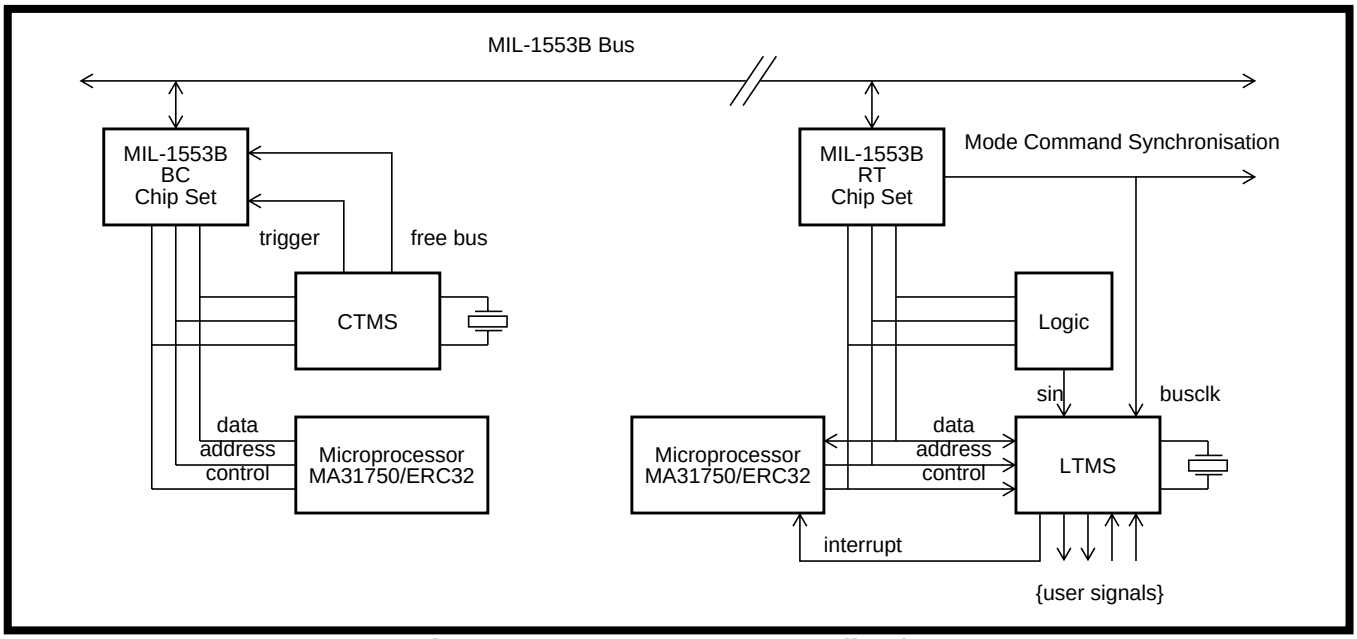


Figure A.5 - MIL-1553B Bus Application

Appendix B: External DPLL Usage

The use of an internal DPLL imposes constraints on parameters as drift and jitter on phase references and crystal frequency accuracy (cf. section 1.9.5). If the constraints are too tight for use with an internal DPLL, an external DPLL must be used. This in turn means that the user becomes responsible for the synchronisation of the LTMS.

The external DPLL should provide phase correction commands to the LTMS in order to minimise the phase error between external phase references and the ET counter phase as much as possible. These commands are provided via **dplldec** and **dpllinc**. In order to be able to perform phase-adjustment properly, the external DPLL must take into account the following inputs:

- **dphase**: provides each phase reference detected by the LTMS to the user.
- **extetld**: informs the user when an LTMS synchronisation is being performed.
- **extclk**, **extcount**: clocks

The phase corrections to be provided by the external DPLL are twofold:

- **Phase corrections because an LTMS synchronisation took place:**

  Since an external DPLL is used, the phase relationship between external asynchronous synchronisation marker and actual LTMS synchronisation moment is not fixed. The phase difference between both events is variable and lies between 0 and 6 **xtal1** periods. The phase correction provided by the DPLL should compensate the uncertainty of the LTMS synchronisation moment.

  If the user can live with the inaccuracy introduced by the LTMS synchronisation, this type of phase correction is not needed. In order to compensate the LTMS synchronisation inaccuracy, the user must measure the phase difference between **dphase** and the moment an LTMS synchronisation is performed (assertion of **extetld**). Once this difference is known the external DPLL can calculate the number of phase corrections to be performed.
- **Phase corrections because of a discrepancy between external phase references and the ET counter phase:**

  This discrepancy occurs due to jitter and drift on phase references or due to crystal frequency inaccuracy. The phase correction provided by the DPLL should compensate these effects. Phase corrections to be performed can be calculated by measuring the distance between two subsequent **dphase** occurrences (normally a window in which the next phase reference is expected is used). This measurement allows the user to determine whether or not a phase reference is valid (free-wheeling capability) and also the number of phase corrections to be performed. If the external DPLL is designed such that all phase corrections have been terminated by the LTMS at the moment a new phase reference can occur then the LTMS synchronisation, inaccuracy can be limited to maximum 3 **xtal1** periods.

Only one type of phase correction per phase reference should be performed; phase corrections due to LTMS synchronisation take precedence over the normal phase corrections.

Practicalities and recommendations

The following recommendations should be followed:

- Time differences should be measured in terms of **xtal1** periods.
- Phase corrections are performed at a rate of maximum one **xtal1** period per **extcount** period
- Only one type of phase correction per phase reference should be performed; this should be achieved by not starting to perform normal phase corrections before one is sure that no LTMS synchronisation event happened. The latest occurrence of LTMS synchronisation is the ideal occurrence + 6 (or 3) **xtal1** periods (cf. figure B.1).
- **dphase** and **extcount** are most easily to be sampled by the **extclk** rising edge.
- **extetld** is a non-spikefree output signal that only can be sampled properly by **extcount**. In order to measure phase differences, first a clocked version of **extetld** should be made (extetld_clocked). This clocked version must then be synchronised with the rising edge of **extclk** after which a synchronous rising edge detection can be performed that stands for the LTMS synchronisation moment. The comparison between occurrence of this rising edge detection versus the ideal occurrence gives the number of phase corrections to be performed.

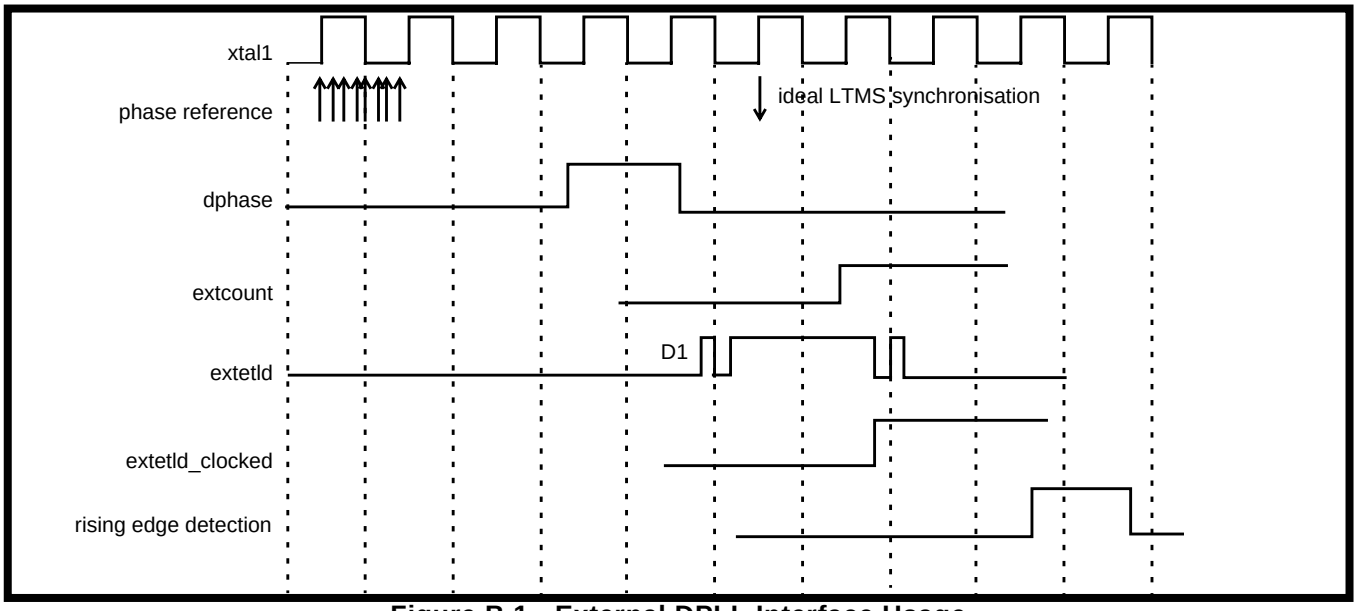


Figure B.1 - External DPLL Interface Usage

The timing diagram in figure B.1 illustrates the principle. The rising edge detection (one synchronisation stage in the figure) gives an indication of when the LTMS synchronisation happened. **dphase** gives an indication where the LTMS synchronisation should have happened. Combining the information results in the calculation of the number of corrections.

The phase references arrows in the figure indicate the occurrence of one asynchronous phase reference edge. The ideal occurrence of the LTMS synchronisation (rising edge of **extcount**) has to be related to this edge.

- **dplldec**/**dpllinc** must be synchronous with either rising or falling edge of **extcount**. After the occurrence of an **extcount** edge, **dplldec** and **dpllinc** must be stable after at most T_{maxF} expressed in the following equality:

  $T_{maxF} = \alpha.T_x - T_{dpll}$

  where α=2 when rising **extcount** edges are used and α=3 when falling edges are used. Δ_{xta}l is the crystal tolerance, expressed in ppm.

- A correction command (by means of assertion of **dplldec**/**dpllinc**) is executed by the LTMS at the second falling edge of **extcount** following the assertion of **dplldec**/**dpllinc**.

Point of contact

MITEL Semiconductor AB
Bruttovägen 1
Box 520
SE-17526 Järfälla, Sweden
+46 8580 24500