Aniotek Inc.

UNIFIED FIELDBUS CONTROLLER
UFC100-L2
BASIC MODE

USERôS MANUAL

COPYRIGHT NOTICE

For further information contact:

Aniotek Inc.
10 April Drive
Dayton, NJ 08810
+1 732 274 2648
info@aniotek.com

Revision History

Rev. no.	Reason for change
1.0	Released UFC100-L2, the second source from another foundry, technical changes are documented in 1.5 Changes between UFC100-L1 and UFC100-L2. The Version number read from register 0x00 has changed ï see Table 2: Basic mode registers and 2.2.3 Reset, version

TABLE OF CONTENTS

1 INTRODUCTION

The UFC100-L2 (Unified Fieldbus Controller) is a peripheral that that can be used in a Fieldbus Device or Host to provide a complete solution for implementing Fieldbus equipment. The UFC100-L2 includes all of time-critical functions in the hardware. It implements part of Physical and Data Link Layers for the Foundation Fieldbus H1 and Profibus-PA. This document describes the mode of operation that is compatible with existing Fieldbus controllers. It shows the pin signals, the internal registers that can be accessed by a software program, electrical specifications and package dimensions. It also includes procedures for software device drivers and hardware test.

1.1 Overview

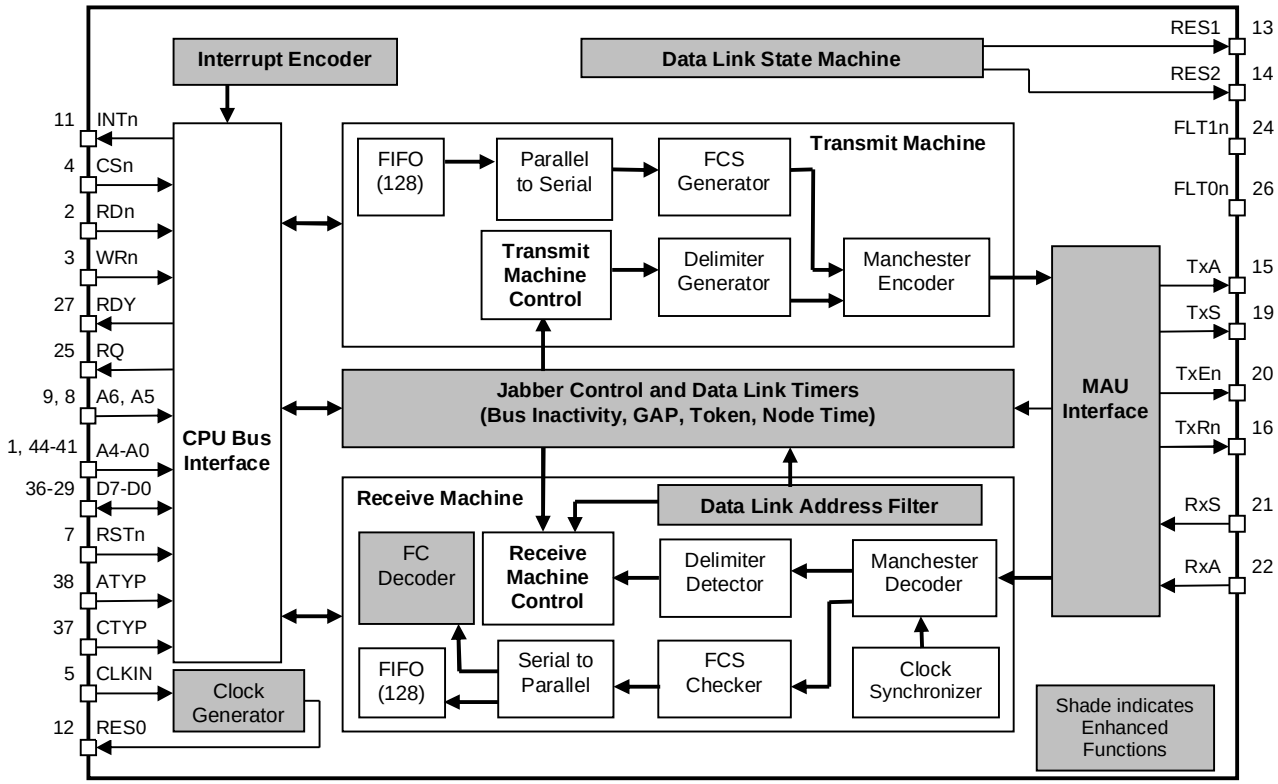

Figure 1: UFC100-L2 Block Diagram

1.2 Features

It is:

- Compliant to IEC 61158-2 Physical layer at 31.25 Kbit/s,
- Compliant to IEC 61158-4 Data Link layer,
- RoHS certified 44 pin LQFP package,
- Operating voltage 2.7 to 3.6 V,
- Low current consumption suitable for Field devices,
- Flexible 8-bit CPU bus interface suitable for all types of processors,
- 128 byte Transmit and Receive FIFO to reduce the number of the interrupts to the CPU.

1.3 Applications

The UFC100-L2 can be used for:

- FF-H1 Fieldbus Device,
- PROFIBUS-PA Fieldbus Device,
- H1 Host Interface,
- HSE Linking Device.

1.4 Ordering Information

UFC100-L2 Unified Fieldbus Controller in LQFP package

Order number of UFC100-L2: IFL-KK-02091

1.5 Changes between UFC100-L1 and UFC100-L2

The UFC100-L2 has function, package and pins same as UFC100-L1, except in L2 the following pins are not 5 V tolerant.

Pin	Signal
7	RESETn
21	RxS
22	RxA
26	FLT0n
37	CTYP
38	ATYP

The UFC100-L2 draws lower current for its operation ï see Current Consumption.

1.6 PIN Description

The following conventions are used.

Name If the name ends in ónô then that signal is active low.

Type It specifies the type of input or output.

P	Power
ICH	Input ï CMOS with hysteresis
O	Output ï always active
BCH	Input / Output CMOS with hysteresis

Reset value For output signals, it specifies the value when external (hardware) reset is applied.

H	High
L	Low
TS	Tristate

Table 1: UFC100-L2 Pin out

Pin no.	Name	Type	Reset value	Description
1	A4	ICH		CPU Address bus
2	RDn/EDSn	ICH		Read Strobe (Intel mode), E or DSn (Freescale mode)
3	WRn/RWn	ICH		Write Strobe (Intel mode), RWn ï Read or Write select (Freescale mode)
4	CSn	ICH		Chip Select ï active state enables Read or Write access.
5	CLKIN	ICH		Clock input ï the frequency has to be one of 1, 2, 4 or 8 Mhz.
6	Vss	P		Power negative side
7	RESETn	ICH		Reset ï active (low).
8	A5	ICH		This pin can be connected to CPU address A5, or to Vdd or Vss.
9	A6	ICH		This pin can be connected to CPU address A6, or to Vdd or Vss.
10	MS2	ICH		Reserved for test use; connect it to Vss.
11	INTn	O	H	Interrupt request to the CPU
12	RES0	O	L	Reserved
13	RES1	O	L	Reserved
14	RES2	O	L	Reserved
15	TxA	O	L	It is used as TxA output, which is high while transmitter is active.
16	TxRn	O	H	Low pulse of 8 µs duration whenever TxS changes.
17	Vdd	P		Power plus side
18	Vss	P		Power negative side
19	TxS	O	L	Transmit signal to medium attachment unit
20	TxEn	O	H	Transmit control to medium attachment unit
21	RxS	ICH		Receive Signal from medium attachment unit
22	RxA	ICH		Receive activity (carrier detect) from medium attachment unit
23	TST0	ICH		Test input; connect to Vss in normal operation.
24	FLT1n	ICH		Not used ï this pin can be connected either Vss or Vdd.
25	RQ	O	L	DMA request output, one pulse per byte to be transferred
26	FLT0n	ICH		Not used ï this pin can be connected either Vss or Vdd.
27	RDY DACKn	O	H H	ATYP high: low value indicates that the data is ready or accepted. ATYP low: high value indicates that the data is ready or accepted.
28	Vss	P		Power negative side
29	D0	BCH	TS	CPU Data bus
30	D1	BCH	TS	CPU Data bus
31	D2	BCH	TS	CPU Data bus
32	D3	BCH	TS	CPU Data bus
33	D4	BCH	TS	CPU Data bus
34	D5	BCH	TS	CPU Data bus
35	D6	BCH	TS	CPU Data bus
36	D7	BCH	TS	CPU Data bus
37	CTYP	ICH		Type of CPU ï Low: Intel, High: Freescale
38	ATYP	ICH		Type of bus access ï Low: RDY output, High: DACKn output
39	Vdd	P		Power plus side
40	Vss	P		Power negative side
41	A0	ICH		CPU Address bus
42	A1	ICH		CPU Address bus
43	A2	ICH		CPU Address bus
44	A3	ICH		CPU Address bus

2 BASIC MODE OPERATION

2.1 Operation

2.1.1 Transmit Machine

It has 128 byte Transmit FIFO. For most of the frame types, the CPU can write the entire frame to this memory. The automatic FCS can be disabled for test purpose. The Transmission starts only when the gap from the immediately prior reception or transmission is more than or equal to the programmed minimum Gap. Transmission starts by sending a programmed number of Preamble bytes followed by the Start Delimiter. The transmission ends by sending any remaining bytes of the frame, followed by two FCS bytes, followed by End Delimiter. DMA can be used to transfer bytes to be transmitted from the memory to the FIFO.

If there is a transmission error or if the CPU aborts the transmission; it ends by sending the current byte, followed by wrong (inverted) FCS followed by End Delimiter; and the CPU or DMA attempt to write to the FIFO is ignored until the CPU resets transmission error flags. If the error or abort is detected while transmitting Preamble, then it ends after sending the full byte of Preamble. If the error or abort is detected while transmitting Start Delimiter, then it ends after sending Start Delimiter followed by End Delimiter. If the error or abort by the CPU occurs after the transmitter has read the last byte from the FIFO ï only possible for FIFO overflow error or length mismatch error, then the error flag is set, but transmission is not aborted. The error flags are set when Transmitter has finished transmitting the last bit, unless it is FIFO overflow error and it is detected before TRON is set to ó1ô

2.1.2 Receive Machine

The received frame is stored in the 128 byte Receive FIFO. The signal polarity of the received signal is automatically corrected. For most frames, an interrupt is generated only after the entire frame has been stored in this memory. The CPU can cancel the current reception. The current reception is also cancelled if any error is detected, but the Receive FIFO is not cleared. The receiver does not write any more bytes to this FIFO. The error flag is set when the error is detected, but further errors are not registered. DMA can be used to transfer received bytes from the FIFO to the memory.

2.1.3 Data Link Timers

The Node Time counter keeps the value of Node time with a resolution of 1/4 ms or 1/32 ms depending upon another setting.

Watch-time counter is used to monitor Maximum-Response-Delay, Immediate-Response-Recovery-Delay and Token-Recovery-Delay. Whenever there is no receive or transmit activity, it counts once every byte-time. Whenever there is receive activity or there is active transmission, this counter is reset to zero. There is a filter on bus activity, so that noise does not reset this counter.

Gap counter is used to provide the minimum gap between two frames ï it runs whenever there is no receive or transmit activity.

Jabber Counter is used to check the length of transmit frame or receive frame ï it runs whenever there is transmit activity and jabber is enabled or there is receive activity and jabber is enabled.

Token Counter is used as Remaining Token Duration timer. It is always loaded from the PT frame. It can be reloaded by the CPU any time. The internal clock has to be active to load Token counter. Token timer is always enabled. It counts down whenever it is non-zero.

2.1.4 MAU Interface

It converts the internal transmit signals to TxE and TxS. The Transmit driver can be setup for Transmit Enable or ADD mode. The input RxA and RxS signals are converted to internal receive signals. The loopback modes can be setup for testing. Physical layer parameters such as Preamble extension and minimum Gap can be setup.

2.1.5 CPU Bus Interface

The UFC100-L2 can be connected to any synchronous or asynchronous bus. The type of CPU can be Intel or Freescale. The selection is done by two input pins. The data interface is 8-bit wide; the address bus is 5-bit wide.

2.2 Basic mode registers

2.2.1 Register list

Table 2: Basic mode registers

Address HEX	Name	Access Read/ Write	Reset value	Description
00	Reset, Version	R/W	0x30	Software reset**, UFC100-L2 Version**
01	Mode	R/W	0x00	Selection of operating modes
02	Control	R/W	0x00	Control functions
03	Status	R/-	0x82	Shows status
04	Interrupt status	R/-	0x00	Shows reason of interrupts
05	Error status	R/-	0x00	Shows reason of communication errors
06	Interrupt mask	R/W	0xFE	Mask for interrupts
07	Error mask	R/W	0xFB	Mast for error interrupts
08	Tx length (LOW)	R/W	0x00	Length of transmitted frame (Lower byte)
09	Tx length (HIGH)	R/W	0x00	Length of transmitted frame (Higher 2 bits)
0A	Reserved	-/-	--(1)	Not used
0B	FIFO control	R/W	0x00	Control register of FIFO
0C	FIFO status	R/-	0x11	Shows status of FIFO
0D	DATA	R/W	0x00	Transmit/Receive data to/from FIFO
0E	Reserved	-/-	--(1)	Not used
0F	Reserved	-/-	--(1)	Not used
10	Clock mode	R/W	0x00	DL mode, Timer enables
11	Timer status	R	0x00	Node-timer status
12	Node time (LOW)	R/W	0x00	Timer to hold DL NODE time (Lower byte)
13	Node time (HIGH)	R/W	0x00	Timer to hold DL NODE time (Higher byte)
14	Gap time	-/W	0xFF	Value to generate minimum inter-PDU delay
15	Reserved	-/-	--(3)	Not used
16	Watch time (LOW)	-/W	0xFF	Value to detect no-activity of bus (Lower byte)
17	Watch time (HIGH)	-/W	0xFF	Value to detect no-activity of bus (Higher byte)
18	Token counter (LOW)	R/W	0x00	Remaining token holding time (Lower byte)
19	Token counter (HIGH)	R/W	0x00	Remaining token holding time (Higher byte)
1A	Timer control	R/W	0x00	Control DL timers
1B	Reserved	-/-	--(1)	Not used
1C	Reserved	-/-	--(1)	Not used
1D	Reserved	-/-	--(1)	Not used
1E	Reserved	-/-	--(1)	Not used
1F	Reserved	-/-	--(1)	Not used
(1): Unused registers read as 0x00.				

2.2.2 Performance Improvement

Even with existing software, UFC100-L2 reduces the number of the interrupts to the CPU and thus provides performance improvement. There is filter on bus activity to make it less sensitive to noise.

2.2.3 Reset, version

Address: 0x00
Reset value: 0x30

Table 3: UFC Reset, Version register

	Reset, Version
Bit no.	**Name**
7	RST
6	Not used
5 ï 0	Ver[5..0]

RST **Soft Reset**

Writing ó1ô to this field applies reset to all parts of the UFC100-L2. It may take up to 4 cycles of input clock to complete the reset operation. If the CPU does not use RDY signal, then it should check ARDY field in Status register ï see 2.2.6, before starting the next Read or Write cycle from the CPU. This field is always read as ó0ô

Ver[5..0] **UFC Version**

This read only field shows the UFC version number. Its value is 0b010000.

2.2.4 Mode

Address: 0x01
Reset value: 0x00

This registers sets up clock and other modes. The internal clock is enabled only after the CPU writes any value to Mode register even if the value to be written is 0x00, so that the internal clock does not start at a higher than the desired frequency.

Table 4: UFC Mode register

	Mode
Bit no.	**Name**
7, 6	CLOCK
5	LB
4	FDP
3,2	PRE
1	TFCS
0	TMD

CLOCK **Clock divider**

This field is used to select divide factor of prescaler to generate internal clock of 500 kHz frequency.

Table 5: Clock divider

Input clock	Divider	CLOCK[7,6]
1 MHz	2	00
2 MHz	4	01
4 MHz	8	10
8 MHz	16	11

LB **Loop back**

If this field is set to ó1ô then internal loopback from the transmitter to the receiver is enabled. In this loopback mode, the transmitted signal is fed to the receive circuit and the signals from the external MAU are ignored. TxS and TxEn stay inactive in the internal loopback.

FDP **Full duplex**

If this field is set to ó0ô then the receiver is disabled while transmitting, else receiver is always enabled. It is necessary to set FDP to ó1ô to receive the transmitted frame in the internal loopback mode. The value of this field does not affect the current consumption.

PRE **Preamble extension**

It specifies extension of preamble bytes for each transmission.

```
PRE    Preamble count
00     1 byte,
01     2 bytes,
10     3 bytes,
11     4 bytes.
```

TFCS **Through pass of FCS**

If this field is set to ó1ô then the two bytes of the FCS are not appended during transmission and the last two bytes of the received frame are stored in the Receive FIFO. The FCS is always checked during reception and if incorrect then the FCSE field in the Error status register (see 2.2.8) is set to ó1ô.

TMD **Transmission mode**

This field controls transmission mode.

```
0      Enable mode,
1      ADD mode.
```

2.2.5 Control

Address: 0x02
Reset value: 0x00

Table 6: UFC Control register

	Control
Bit no.	**Name**
7	DMA
6	RE
5	RCL
4	Not used
3	JIE
2	TRON
1	Not used
0	IE

DMA **DMA enable**

If this field is set to ó1ô then the data transfer between the FIFO and the memory is done using DMA and RQ output is enabled. The TRON field controls the direction of the data transfer. If TRON is ó1ô, then data is transferred from the memory to the Transmit FIFO, else the data is transferred from the Receive FIFO to the memory.

For transmission, the DMA transfer starts when ôTRONô and ôDMAô both are set to ó1ô. The Transmit length register is decremented by ó1ô for every byte written to the Transmit FIFO. The DMA transfer continues until either the Transmit length register becomes zero or the Transmit FIFO becomes full. If the Transmit FIFO becomes full then DMA starts again when there is space in that FIFO and continues until the Transmit length register becomes zero. When the Transmit length register becomes zero, the ôDMAô field is reset to ó0ô. ôDMAô field is also reset to ó0ô when transmission is aborted.

For reception, if the DMA is enabled then all data is transferred from the Receive FIFO to the memory before disabling DMA. If a reception error occurs then the reception is cancelled but DMA is not disabled and the error flag is set only after the Receive FIFO becomes empty.

RE **Receiver enable**

If this field is set to ó1ô then the Receiver is enabled, else the UFC100-L2 ignores the receive activity. If ôREô is reset to ó0ô while a reception is active, then it cancels that reception and resets the Receiver.

RCL **Receive cancel**

If this field is set to ó1ô then the current reception is cancelled. It also clears error status and the receiver FIFO even if the entire frame had been received in the FIFO. If the CPU sets RCL to ó1ô while Receiver is inactive, then the attempt to write to this field is ignored. The read back of this field always returns ó0ô.

JIE **Jabber detection enable**

If this field is set to ó1ô then jabber detection function is enabled. Jabber timer starts when RxA signal becomes active. If this signal stays active for 4096 bit time (131 ms at 31.25kbps), jabber timer detects this and sets JI field in the Error status register ï see 2.2.8.

TRON **Transmitter ON**

Writing ó1ô to this field starts a transmission, which ends after sending the End Delimiter. This field is reset to ó0ô by the UFC100-L2 at the end of the transmission. If the CPU resets this field to ó0ô during an active transmission, then the transmission is aborted by sending the current byte, followed by wrong (inverted) FCS followed by End Delimiter and the CPU attempt to write to the FIFO is ignored until the CPU resets transmission error status or TED status field. If the CPU initiates the abort and no other error occurs then the completion is indicated by TED status.

IE **Interrupt enable**

If this field is set to ó1ô then all unmasked interrupts are used to activate INTn hardware signal.

2.2.6 Status

Address: 0x03
Reset value: 0x82 or 0x83

Table 7: UFC Status register

	Status
Bit no.	**Name**
7	ARDY
6 ï 3	Not used
2	RFRY
1	TFRY
0	CD

ARDY **Access ready**

If this field is ó1ô, then the CPU can access the internal registers. If RDY hardware signal is used to control the access cycle, then it is not necessary to check this status field.

RFRY **Receive FIFO ready**

If this field is ó1ô, then it indicates that there is at least one byte stored in the Receive FIFO.

TFRY **Transmit FIFO ready**

If this field is ó1ô, then it indicates that there is space for at least one more byte in the Transmit FIFO.

CD **Carrier detect**

If this field reflects the status of RxA input ï ó1ô indicates active carrier. In loopback, it shows the status of internal RxA.

2.2.7 Interrupt status

Address: 0x04
Reset value: 0x00

This register shows the reason(s) for the interrupt. A ó1ô in a field indicates that the condition for that interrupt is active. If an interrupt status is ó1ô and the corresponding condition is not masked (2.2.9) and IE field of Control register (see 2.2.5) is ó1ô, then INTn hardware signal becomes low. Bits 1 through 6 of this register are automatically cleared when this register is read. Bit 7 of this register (ERS) is cleared when Error status register is read.

Table 8: UFC Interrupt status register

	Interrupt status
Bit no.	**Name**
7	ERS
6	RTI
5	TED
4	RED
3	TFI
2	RFI
1	LCD
0	Not used

ERS **Error condition**
This field is inclusive OR of all unmasked fields in the Error status register.

RTI **Watch-timer interrupt**
This field is set to ó1ô when the Watch time counter reaches the value set in Watch time register. This field is not set to ó1ô if this field is masked in the Interrupt mask register. This interrupt can be used to find out if the bus is inactive for more than the set duration.

TED **Transmission end**
This field is set to ó1ô when a frame transmission is successfully terminated or completed. If there is any error in the transmission then this field is not set, but the corresponding error status is set to ó1ô.

RED **Reception end**
This field is set to ó1ô when a complete frame is received without any error and RE field of Control register is ó1ô. If the frame is received with error then this field is not set, but the corresponding error status is set to ó1ô.

TFI **Transmit FIFO indication**
This field is set to ó1ô when the byte count in the Transmit FIFO decreases to reach the threshold value specified by TCTRL field of FIFO control register (see 2.2.12). The õTFIô status becomes ó1ô only if the last byte of the current transmission has not been read out of the FIFO. õTFIô is reset to ó0ô when Transmitter has read all bytes of the frame from the FIFO and before it has transmitted FCS and End Delimiter. It is also reset when the CPU resets Transmit FIFO using CTF command (see 2.2.12).

RFI **Receive FIFO indication**
This field is set to ó1ô when the byte count in the Receive FIFO increases to reach the threshold value specified by RCTRL field of FIFO control register (see 2.2.12).

LCD **Loss of carrier detect**
This field is set to ó1ô when the RxA signal changes from HIGH to LOW and if this condition is not masked. This field is not set if this condition is masked or if the internal loopback is enabled.

2.2.8 Error status

Address: 0x05
Reset value: 0x00

This register shows the reason(s) for the error interrupt. A ó1ô in a field indicates that the condition for that interrupt is active. An inclusive OR of all unmasked conditions (see 0) is shown in the ERS field of the Interrupt status register. The error status bits are automatically cleared when this register is read.

If at least one of FFER, RFR and NEPT fields is ó1ô, UFC cancels reception of the frame causing the error, in the same manner as setting of RCL field of Control register to ó1ô, except that the Receive FIFO is not cleared. The error flag is set when the error is detected, but further errors are not registered and the Receiver does not write any more bytes to the Receive FIFO.

Table 9: UFC Error status register

	Error status
Bit no.	**Name**
7	FFER
6	JI
5	TLM
4	CNS
3	FCSE
2	Not used
1	RFR
0	NEPT

FFER **FIFO error**
This field is ó1ô when any error field in FIFO (RFOR, RFUR, TFOR or TFUR fields in FIFO Status register) is ó1ô

JI **Jabber**
This field is set to ó1ô when the jabber timer reaches to 4096 bit time.

TLM **Transmit length mismatch**
This field is set to ó1ô when the CPU tries to write more bytes than the length of the frame set in the Transmit frame length register ï see 0.

CNS **Carrier not seen**
This field is set to ó1ô when UFC finishes transmission of Start Delimiter and RxA is not active at that moment. This implies problem in driver/receiver circuit of the attached MAU.

FCSE **FCS error**
This field is set to ó1ô when FCS in the received frame is not equal to the calculated value.

RFR **Receive framing error**
This field is set to ó1ô when reception is enabled and any of the following error is detected during the reception:

- detection of N+ or N- code which is not a part of Start or End Delimiter, or
- End Delimiter is not on the byte boundary, or
- RxA is negated before End Delimiter is detected, or
- Start Delimiter is not detected and the receive activity ends after being active for more than 24 bit duration.

NOTE receive activity of 24 bit or smaller duration is considered as noise and ignored.

NEPT **FIFO is not ready for a new frame**
This error flag is set to ó1ô when the FC byte of the next frame is received and either the Receive FIFO is not empty or receiver interrupt and error flags for the previous reception have not been reset to ó0ô. This new frame is lost.

2.2.9 Interrupt mask

Address: 0x06
Reset value: 0xFE

This register controls interrupts specified in Interrupt status register in the same order of interrupt reasons. When a field of this register is set to ó1ô, then the corresponding interrupt in Interrupt status register is masked (does not affect the INTn signal). When the CPU needs to receive any interrupt, corresponding field of this register should be cleared to ó0ô prior to waiting for that interrupt. The ERS field of this register masks all error interrupts.

Table 10: UFC Interrupt mask register

	Interrupt mask
Bit no.	**Name**
7	ERS
6	RTI
5	TED
4	RED
3	TFI
2	RFI
1	LCD
0	Not used

2.2.10 Error mask

Address: 0x07
Reset value: 0xFB

This register controls error interrupts specified in Error status register in the same order of error reasons. When a field of this register is set to ó1ô then the corresponding error in Error status register is masked (does not affect the ERS bit in the Interrupt status register). When the CPU needs to receive any error interrupt, corresponding fields of this register should be cleared to ó0ôprior to waiting for the error interrupt.

Table 11: UFC Error mask register

	Error mask
Bit no.	**Name**
7	FFER
6	JI
5	TLM
4	CNS
3	FCSE
2	Not used
1	RFR
0	NEPT

2.2.11 Transmit frame length

Address: 0x08, 0x09
Reset value: 0x0000

This 10-bit register stores the length of the frame as number of bytes to be transferred from memory to the Transmit FIFO. This count does not include Preamble, Start Delimiter and End Delimiter. The FCS is not included in this count, unless transmission of FCS is disabled by TFCS. It is necessary to write a non-zero value to this register before transferring data to the Transmit FIFO.

This register decrements by one when the CPU writes one byte into Transmit FIFO. If the CPU tries to write more bytes than the value of Transmit length register then the extra bytes are not written to the Transmit FIFO. If the CPU does so after TRON has been set to ó1ô then the transmission is aborted and õTLMôis set to ó1ô If the CPU does so before TRON has been set to ó1ô then the transmission is aborted when the CPU tries to set TRON to ó1ôand õTLMôis set to ó1ôat that time.

The CPU can read and write this register. When the CPU reads the lower byte of this register, the current value of the register is not latched. If a transmission is active, then the reading of this register may not return correct value. The address 0x08 accesses the lower 8-bits of the frame length. The upper 6-bits at address 0x09 are read as zeros.

2.2.12 FIFO control Register

Address: 0x0B
Reset value: 0x00

Table 12: UFC FIFO control register

	FIFO control
Bit no.	**Name**
7	Not used
6	CRF
5, 4	RCTRL
3	Not used
2	CTF
1, 0	TCTRL

CRF **Clear Receive FIFO**

Writing ó1ô to this field clears the Receive FIFO, its error status and the FIFO becomes empty. This field is automatically reset to ó0ô by the UFC100-L2.

RCTRL **Receive FIFO threshold**

The value of this field sets the FIFO threshold that is used to set RFI field ï see 2.2.7.

```
RCTRL   Threshold
00       8 bytes,
01      16 bytes,
10      24 bytes,
11      32 bytes.
```

CTF **Clear Transmit FIFO**

Writing ó1ô to this field clears the Transmit FIFO, its error status and the FIFO becomes empty. It also resets Transmit length register to zero. This field is automatically reset to ó0ô by the UFC100-L2.

TCTRL **Transmit FIFO threshold**

The value of this field sets the FIFO threshold that is used to set TFI field ï see 2.2.7. The threshold value depends upon TRON.

```
RCTRL   Threshold
        TRON = '0'    TRON = '1'
00       4 bytes,      8 bytes,
01       8 bytes,     16 bytes,
10      16 bytes,     32 bytes,
11      24 bytes,     64 bytes.
```

Some of the software programs try to fill the Transmit FIFO before turning on the transmission by setting TRON to ó1ô The FIFO threshold is lower during this time, so that there is less delay in turning on the transmission.

2.2.13 FIFO status

Address: 0x0C
Reset value: 0x11

This read only register shows the full, empty and error status of the FIFOs.

Table 13: UFC FIFO status register

	FIFO status
Bit no.	**Name**
7	RFOR
6	RFUR
5	RFF
4	RFE
3	TFOR
2	TFUR
1	TFF
0	TFE

RFOR **Receive FIFO overrun**
This field is set to ó1ô when Receive FIFO is full and another byte is received in the current reception. This field is reset to ó0ô when the CPU writes ó1ô to CRF in FIFO control register.

RFUR **Receive FIFO underrun**
This field is set to ó1ô when Receive FIFO is empty and the CPU or DMA tries to read another byte from the Receive FIFO. This field is reset to ó0ô when the CPU writes ó1ô to CRF in FIFO control register.

RFF **Receive FIFO full**
This field is ó1ô when Receive FIFO is full and ó0ô whenever there is at least one byte space in the FIFO.

RFE **Receive FIFO empty**
This field is ó1ô when Receive FIFO is empty and ó0ô whenever there is at least one byte in the FIFO.

TFOR **Transmit FIFO overrun**
This field is set to ó1ô when the Transmit FIFO is full and the CPU or DMA tries to write another byte to Transmit FIFO. This field is reset to ó0ô when the CPU writes ó1ô to CTF in FIFO control register.

TFUR **Transmit FIFO underrun**
This field is set to ó1ô when the Transmit FIFO is empty and the UFC100-L2 tries to read another byte from the Transmit FIFO to transmit the frame, or if the transmission is started and Transmit length register is zero. This field is reset to ó0ô when the CPU writes ó1ô to CTF in FIFO control register.

TFF **Transmit FIFO full**
This field is ó1ô when the Transmit FIFO is full and ó0ô whenever there is at least one byte space in the FIFO.

TFE **Transmit FIFO empty**
This field is ó1ô when the Transmit FIFO is empty and ó0ô whenever there is at least one byte in the FIFO.

2.2.14 FIFO data

Address: 0x0D
Reset value: undefined, reads as 0x00

Write to this register appends one byte to Transmit FIFO. This FIFO is 8-bit wide and 128 bytes deep.

Read from this register removes one byte, if available, from Receive FIFO. This FIFO is 8-bit wide and 128 bytes deep.

The two FIFOs are separate, but share one address in the CPU address space.

2.2.15 Clock mode

Address: 0x10
Reset value: 0x00

This register can be written and read.

Table 14: Clock mode register

	Clock mode
Bit no.	**Name**
7 ï 3	Not used
2	NDE
1	GPE
0	Not used

NDE **Node-timer enable**
If this value is set to ó1ô, then the internal Node-time counter uses 32 kHz clock. Else, it uses 4 kHz clock.

GPE **GAP-time enable**
This sets the unit of the GAP-time register. If this value is set to ó1ô, then the unit is eight (8) bit time. Else, then the unit is four (4) bit time.

2.2.16 Timer status

Address: 0x11
Reset value: 0x00

This read only register shows the Node-timer status.

Table 15: UFC Timer status register

	Timer status
Bit no.	**Name**
7 ï 5	Not used
4	NTOF
3 ï 0	Not used

NTOF **Node-timer overflow**
When Node-timer rolls over from the maximum value to zero, an internal overflow bit is set to ó1ô. Its value is latched to this register at the same time when sixteen (16) bit Node-time register is latched by setting LTN bit of Timer control register ï see 2.2.21. This bit is automatically cleared to ó0ôwhen this register is read.

NOTE Multiple overflows cannot be detected by this register.

2.2.17 Node time

Address: 0x12, 0x13
Reset value 0x0000

This is a holding register that is used to read from and write to the internal Node-timer. Node-timer is an internal 16-bit up counter which is reset to zeros by Reset. After Reset, it starts to count up and after reaching the maximum count, it rolls over to zero.

The CPU has to write the lowest address byte (least significant value) first and then the highest address byte (most significant value) to the Node-time register. If the CPU writes to the high address byte first then that write is ignored. If the CPU writes low address byte and does not write high address byte, then holding register continues to wait. The value in the holding register is written to the internal Node-timer immediately after the CPU has written the high address byte after the low address byte.

Read of the internal Node-timer requires that the CPU first snap its value into Node-time register by writing ó1ôto LTN field of the Timer control register (see 2.2.21). If snap command is received after the CPU has written the low address byte, but before it has written the high address byte to the Node-time register, then the snap command is ignored.

The counting frequency of this register depends upon the setting of óNDEô in Clock mode register - 2.2.15.

2.2.18 GAP time

Address: 0x14
Reset value 0xFF

This register stores the time to guarantee minimum gap between successive frames on the bus. It is used only when transmitting a frame. The internal GAP timer starts whenever the bus becomes inactive and stops when it becomes equal to the value set in GAP time register. When transmission is started by setting TRON bit of Control register, the transmission is started only if the internal GAP timer has stopped. The unit of this register depends upon the setting of óGPEô in Clock mode register - 2.2.15. The CPU can read back this register. The internal GAP timer itself cannot be read.

2.2.19 Watch time

Address: 0x16, 0x17
Reset value 0xFFFF

These registers hold threshold to detect ñno activityò on the bus. UFC100-L2 has an internal Watch-time counter that is a monotonously increasing counter and is reset by detecting Start Delimiter or End Delimiter in the RxS signal. When the content of Watch-time counter reaches the value set in Watch time register and if RTI is not masked, RTI interrupt status is set to ó1ô. The Watch-time counter is reset whenever the bus is active. This counter is also reset by setting CT bit of Timer control register (see 2.2.21). The unit of this register is eight (8) bit time.

The address 0x16 accesses the lower 8-bits of the Watch time threshold.

The CPU can read back these registers. The internal Watch-time counter itself cannot be read.

2.2.20 Token counter

Address: 0x18, 0x19
Reset value 0x0000

These registers are used to read the value from or write the value to the internal Token-time counter. The internal counter is a 16-bit down counter. It is reset to zero at the start of UFC100-L2 operation. Whenever it has non-zero value, it counts down once every byte time ï transmission duration of one byte. It stops count down at zero value.

The content of the internal Token-time counter is latched to these registers when LTR bit of Timer control register (see 2.2.21) is set.

The content of these registers are loaded to the internal Token-time counter when any data is written to higher byte register ($19). This implies lower byte ($18) should be written first. It is also loaded from the received PT frame.

2.2.21 Timer control

Address: 0x1A
Reset value 0x00

This write-only register is used to control various internal timers and counters. Read of this register returns 0x00.

Table 16: UFC Timer control register

	Timer control
Bit no.	**Name**
7	Not used
6	LTN
5	LTR
4 ï 2	Not used
1	CT
0	Not used

LTN Latch Node-timer

Writing ó1ô to this field snaps the value of the internal Node-time counter into Node-time register ï address 0x12, 0x13.

LTR **Latch Token-timer**

Writing ó1ô to this field snaps the value of the internal Token-time counter into Token counter register ï address 0x18, 0x19.

CT **Clear Watch-timer**

Writing ó1ô to this field clears the internal Watch-time counter to zero.

3 EXTERNAL INTERFACES

The UFC100-L2 operates at 2.7 ï 3.6 volts. All inputs and outputs require that the voltage does not exceed power supply voltage.

The external interface signals are divided into following groups:

Clock input,
CPU Bus,
MAU, and
Others.

3.1 Clock Input

The UFC100-L2 can work at clock rate of 1, 2, 4 or 8 MHz.

3.2 CPU Bus Interface

The UFC100-L2 has 8-bit wide data bus and 5 bit address bus interface. It can be connected to most of the CPU types without any óglueô logic. It always indicates the completion of the access on óRDY / DACKnô output pin.

1. If the CPU uses multiplexed address and data lines, then the external circuit has to use a latch to store the address.
2. It always indicates the completion of the access on óRDY / DACKnô output pin. A CPU does not have to use óRDY / DACKnô signal, if it can be programmed with wait states. The worst case cycle time is four (4) periods of CLKIN input.
3. If the CPU uses a bus that runs at a higher clock rate than CLKIN input of UFC100-L2 and if the CPU cannot use óRDY / DACKnô signal and if the CPU cannot insert enough wait states, then it has to use software to poll a status bit (ARDY) inside UFC100-L2. This bit indicates that it is ready for next cycle.
4. The inputs A6 and A5 can be connected to ó0ô. These addresses are not used.
5. If the microcontroller does not support DMA, then do not connect RQ output.
6. If the microcontroller supports DMA or DMA controller is available, then use RQ. This output has one pulse for each byte to be transferred. The active polarity of this pulse is high.
7. The reset input can be connected to either an output port of the microcontroller or its reset input.

Table 17: CPU Bus type connections

CTYP (pin 37)	ATYP (pin 38)	CPY type	Description
0	0	Intel	Pin 2 is RDn ï Read Strobe, active low. Pin 3 is WRn ï Write Strobe, active low. Pin 27 is RDY ï a high indicates the cycle can be completed.
0	1	Intel	Pin 2 is RDn ï Read Strobe, active low. Pin 3 is WRn ï Write Strobe, active low. Pin 27 is DACKn ï a low indicates the cycle can be completed.
1	0	Freescale	Pin 2 is E ï a high indicates start of the active part of the cycle. Pin 3 is RWn ï a high indicates Read cycle. Pin 27 is RDY ï a high indicates the cycle can be completed.
1	1	Freescale	Pin 2 is DSn ï a low indicates start of the active part of the cycle. Pin 3 is RWn ï a high indicates Read cycle. Pin 27 is DACKn ï a low indicates the cycle can be completed.

3.2.1 Renesas CPU with RDY

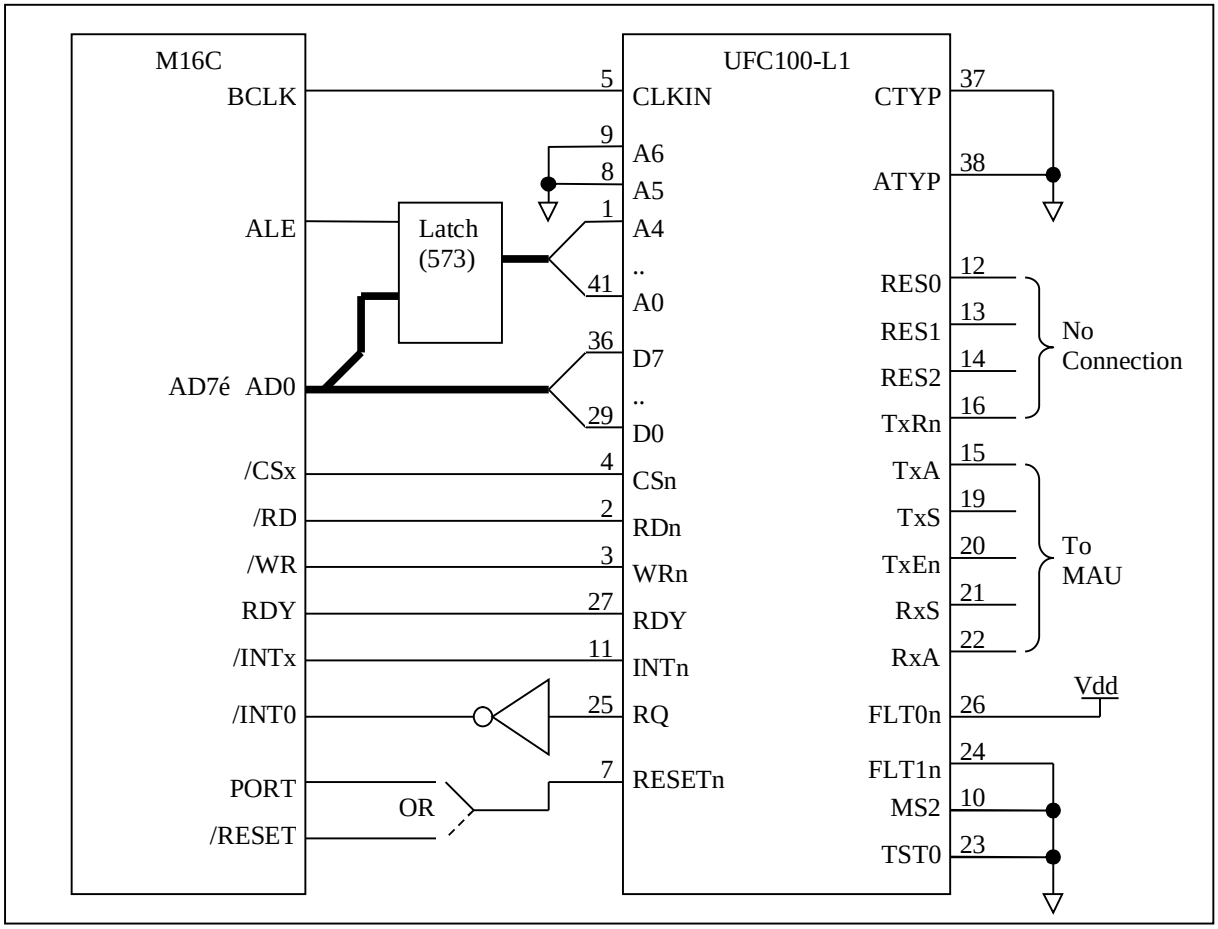

Figure 2: Interface to Renesas M16 using Multiplexed Bus

This type of interface requires separate Read and Write control signals. The UFC100-L2 provides RDY signal, which is normally high. It may become low at the start of the Read or Write cycle to indicate that the CPU has to wait. It becomes high when the CPU can complete the cycle. The Figure 2 shows interface to Renesas microcontroller. This example is meant for existing designs.

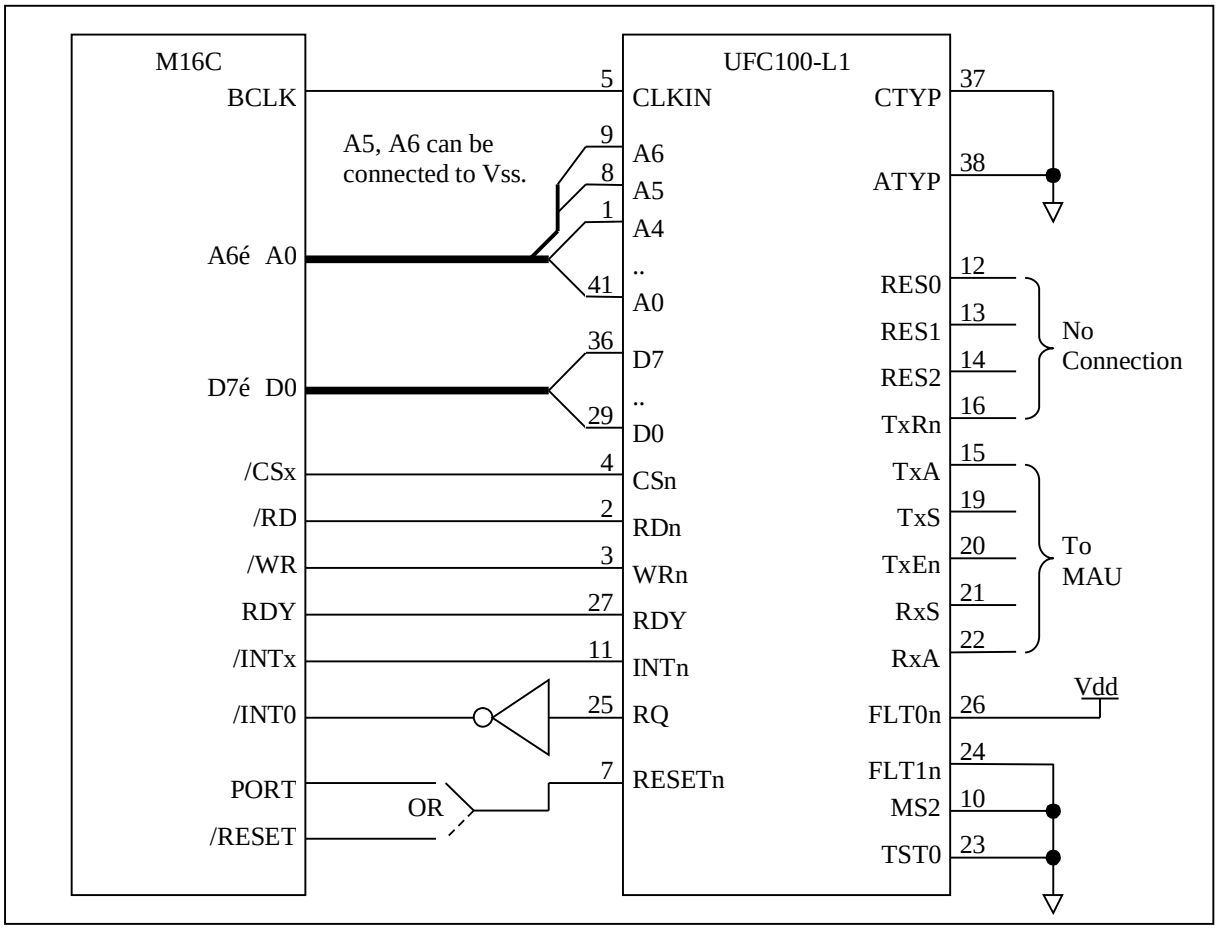

Figure 3: Interface to Renesas M16 using Non-multiplexed Bus

The newer design can use the non-multiplexed bus as shown in the Figure 3. The /INT0 input is used for DMA request.

3.2.2 Intel X86 Type CPU with /READY

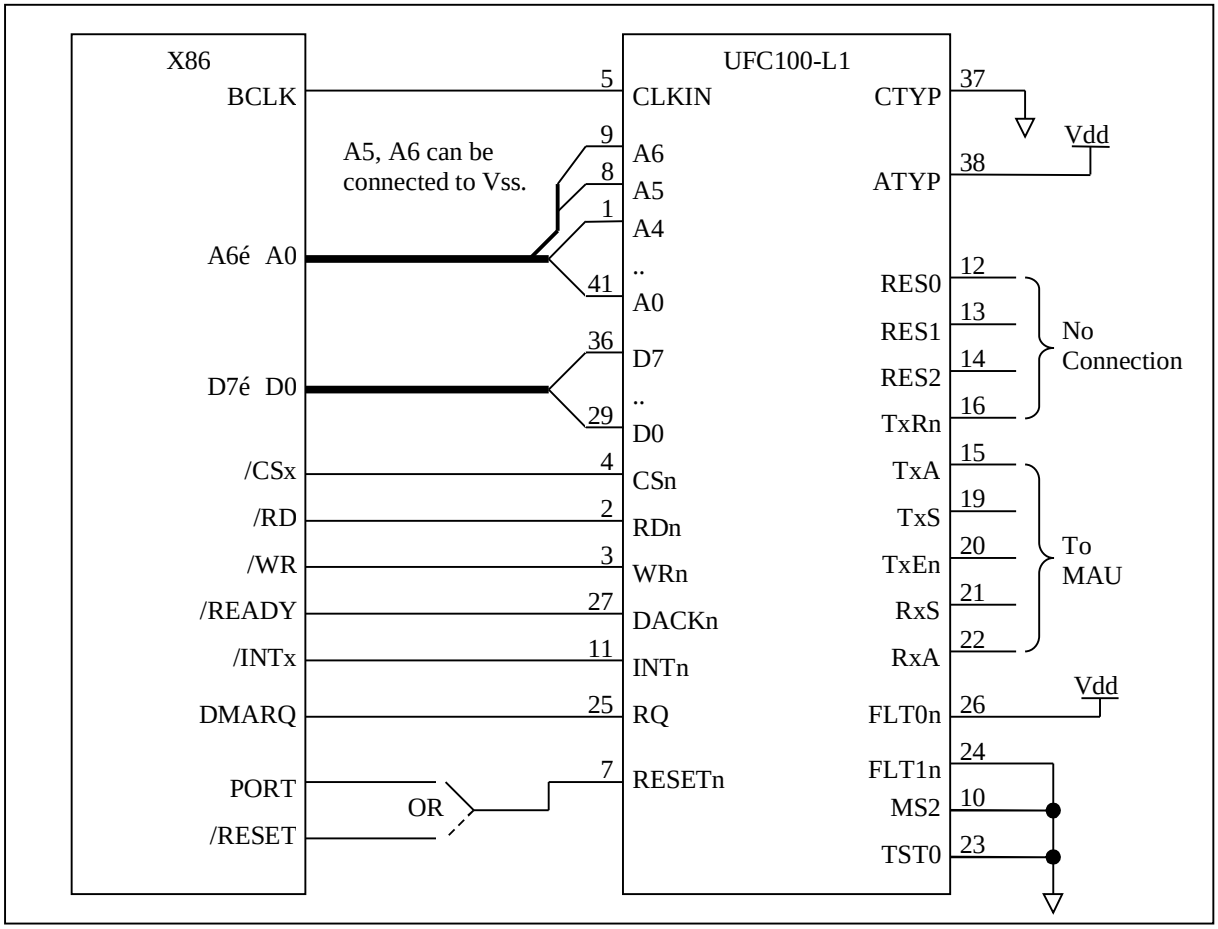

Figure 4: Interface to Intel Type CPU with /READY

This type of interface requires separate Read and Write control signals. It provides a DACKn signal, which is normally high. It becomes low towards the end of a Read or Write cycle to indicate to the CPU that it can complete the cycle. It becomes high when the CPU completes the cycle. The Figure 4 shows interface to such CPU. This example is meant for new designs, because the polarity of the output at pin 27 is reversed.

3.2.3 Freescale Type CPU with /DTACK, Existing Design

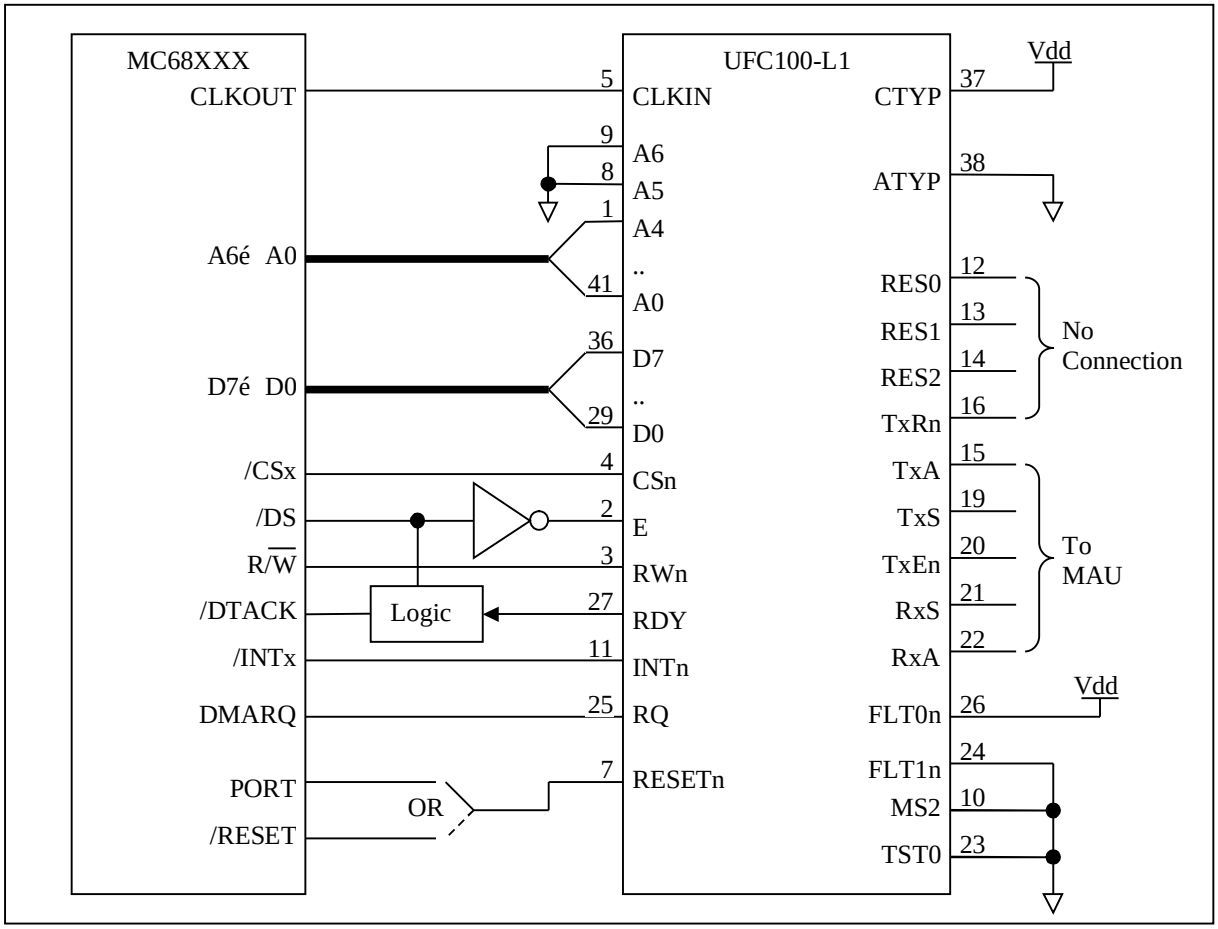

Figure 5: Interface to Freescale MC683XX, MC68C16 ï existing designs

This type of interface requires a common data strobe and another signal for Read, Write control. It provides a RDY signal, which is normally high. It may become low at the start of the Read or Write cycle to indicate that the CPU has to wait. It becomes high when the CPU can complete the cycle. The RDY output requires some amount of logic to convert it to /DTACK signal. The Figure 5 is meant for existing designs, because it requires inverter and logic for two control signals.

3.2.4 Freescale Type CPU with /DTACK, New Design

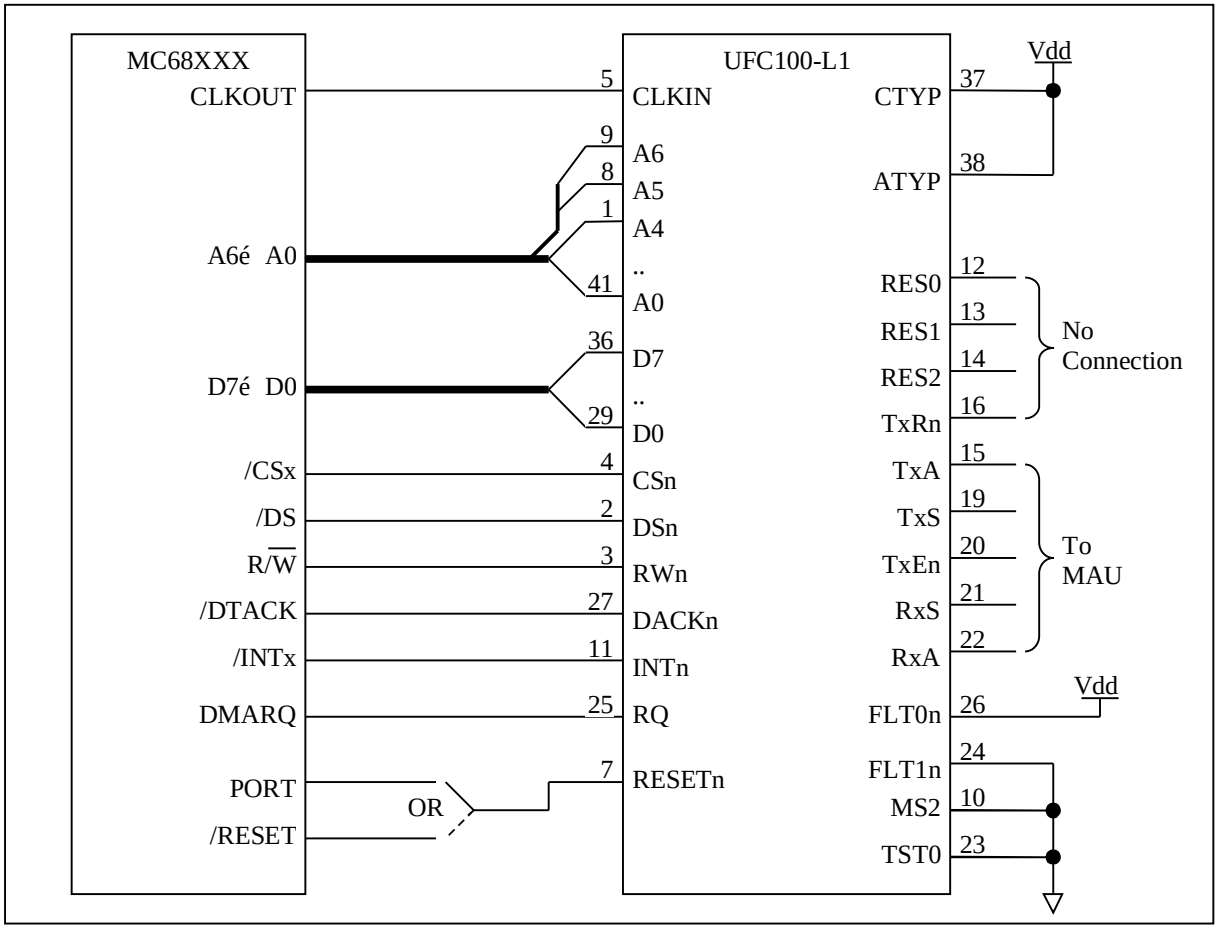

Figure 6: Interface to Freescale MC683XX, MC68C16 ï new designs

This type of interface requires a common data strobe and another signal for Read, Write control. The UFC100-L2 provides a DACKn signal, which is normally high. It becomes low towards the end of a Read or Write cycle to indicate to the CPU that it can complete the cycle. It becomes high when the CPU completes the cycle.

3.2.5 Power PC

The PowerPC has a bus interface which is similar to Intel type CPU. It has /OE and /WE signals which can be connected to RDn and WRn inputs of the UFC100-L2. It can be programmed for various number of wait states.

3.3 MAU Interface

This interface consists of four signals ï TxEn, TxS, TxRn and TxA for Transmitter and two signals ï RxA and RxS for Receiver. These MAU signals are specified in 3.3.1 and 3.3.2.

3.3.1 Transmitter Interface

The two signals ï TxEn and TxS can be setup for Enable or ADD mode as shown in the Table 18.

Table 18: MAU Transmit Modes

Transmit Mode	TxEn	TxS	Operation
Enable Low	1	0	Transmitter Inactive
	1	1	Transmitter Inactive
	0	0	Transmit ó0ĉ
	0	1	Transmit ó1ĉ
ADD	1	0	Transmitter Inactive
	0	1	Transmitter Inactive
	0	0	Transmit ó0ĉ
	1	1	Transmit ó1ĉ

The TxA signal is always high during active transmission for all transmit modes. It can be used to enable the transmit buffer in the MAU that drives voltage or to change the bias current in the MAU that modulates the current. It can also be used to drive an LED to show active transmission. This signal stays high for 4 µs after the end of the transmission, so that the MAU can drive the transmission line to inactive state. TxRn signal is high during inactive transmission. It changes from high to low every time TxS has a transition. It remains low for 8 µs for all transitions, except at the start and the end of the transmission. The start and end low duration is 4 µs.

3.3.2 Receiver Interface

The two signals ï RxA and RxS are shown in the Table 19.

Table 19: MAU Receive Mode

Receive Mode	RxA	RxS	Indication
Level high	0	1	Receive Inactive
	0	0	Receive Inactive
	1	0	Receive ó0ĉ
	1	1	Receive ó1ĉ

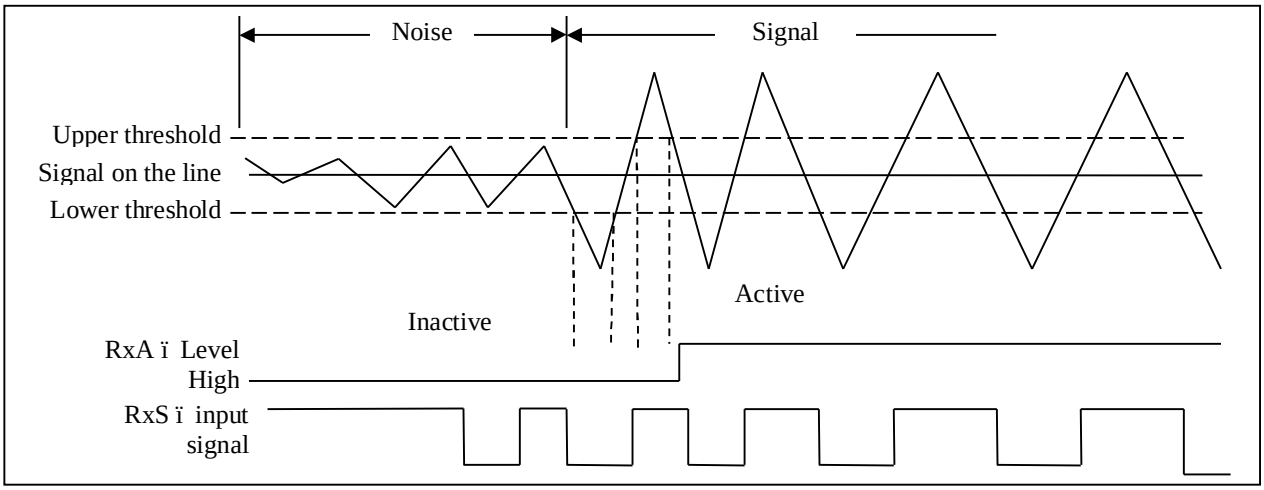

Figure 7: Carrier Detect (RxA) and Signal (RxS) inputs

RxS signal should be generated by comparing the analog received signal with zero and thus it can become active due to noise as shown in the Figure 7. The external MAU should generate active RxA signal only after detecting sufficient activity beyond the noise threshold and for sufficient time, so that noise above threshold can also be filtered. After the noise filter, RxA signal should be continuous level as long as the signal level and duration is considered to be sufficient to infer a valid signal. Receiver does not filter RxA signal. The RxS is considered valid only when RxA is active.

3.4 Other Interfaces

3.4.1 Reset and Interrupt Signals

Table 20: Reset and Interrupt Signals

Signal	Description
RESETn	A low value at this input resets all of the UFC100-L2.
INTn	This output becomes active when it needs to interrupt the CPU. The active polarity is low. This signal stays active as long as any one of the interrupt sources in the UFC100-L2 is active.

The reset signal should be applied as soon as possible after the power is applied. The clock can become active after reset signal is applied. But, the reset should not be removed until after the clock input CLKIN has become active. The VDD power must be at least at the required minimum operating value when reset is removed.

4 ELECTRICAL AND TEMPERATURE SPECIFICATIONS

4.1 Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Units
V_{DD}	Supply Voltage	ï 0.3	4.0	V
VI	DC Input Voltage	ï 0.3	V_{DD} + 0.3	V
VO	DC Output Voltage	ï 0.3	V_{DD} + 0.3	V
T_J	Junction Temperature	ï 40	125	^{0}C

4.2 Operating Conditions

Symbol	Parameter and condition	Min	Typical	Max	Units	Note
V_{DD}	Supply voltage	2.7		3.6	V	
	DC input voltage	0		V_{DD}	V	
VIL	Input low voltage ï ICH type	0		0.3*V_{DD}	V	
VIH	Input high voltage ï ICH type	0.7*V_{DD}		V_{DD}	V	
VH	Input hysteresis		0.4		V	
IL	Input leakage current			5	µA	
VOL	Output low voltage @ IOL = 0.1 mA			0.2	V	
VOL	Output low Voltage RES0, RES1, RES2 @ IOL = 8 mA D7 ï D0, RDY @ IOL = 4 mA All other outputs @ IOL = 2 mA			0.5	V	
VOH	Output high voltage @ ION = ï 0.1 mA	V_{DD} ï 0.2				
VOH	Output high Voltage RES0, RES1, RES2 @ IOL = 8 mA D7 ï D0, RDY @ IOL = 4 mA All other outputs @ IOL = 2 mA	V_{DD} ï 0.8			V	
VO	Voltage applied to tristate output	0		V_{DD}	V	
	Tristate output leakage current			5	µA	
Topr	Operating Temperature	ï 40		85	^{0}C	
I_{DD}	Operating Current consumption @ 3 V @ CLKIN frequency = 2 MHz @ CLKIN frequency = 4 MHz Operating Current consumption @ 3.3 V @ CLKIN frequency = 8 MHz All inputs connected to CMOS outputs, All outputs driving CMOS inputs.		 0.25 0.30 0.45		mA	

4.2.1 Current Consumption

The typical operating current for most applications is less than 1 mA. The current consumption is shown in the Figure 8 and Figure 9 below. It was measured while the device was continuously transmitting and receiving in full duplex using DMA, with continuous Read and Write access to the device.

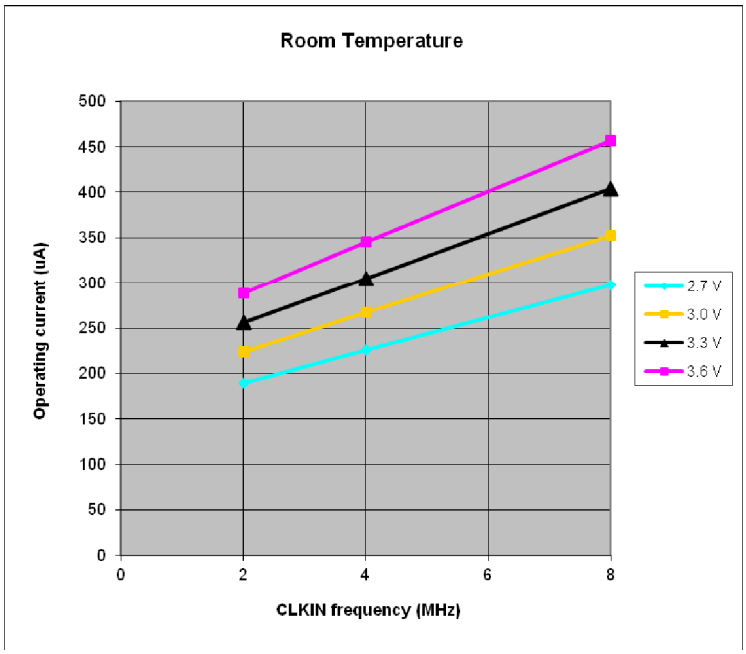

Figure 8: Typical operating current vs. frequency and voltage

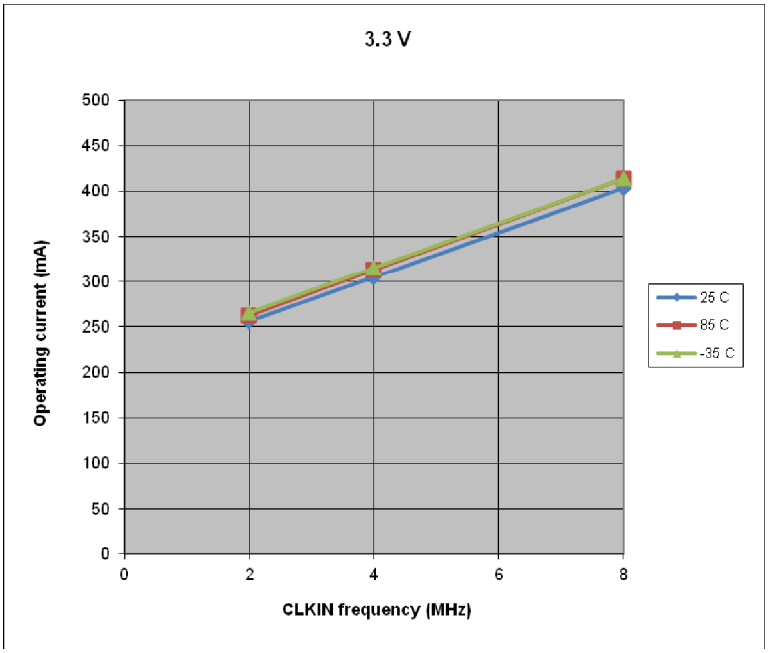

Figure 9: Typical operating current vs. frequency and temperature

The operating current at high temperature is slightly higher than current at room temperature. The operating current at low temperature is about the same as the current at room temperature.

4.3 Clock Input Timings

The timings are shown in the Table 21 and the Figure 10.

Table 21: Clock Timings

Name	Description	Min	Max
T_{CLK}	Clock input period @ V_{DD} = 2.7 to 3.6 V	125 ns	1000 ns
t_{CPWH}	Clock input pulse width high	10 ns	
t_{CPWL}	Clock input pulse width low	10 ns	

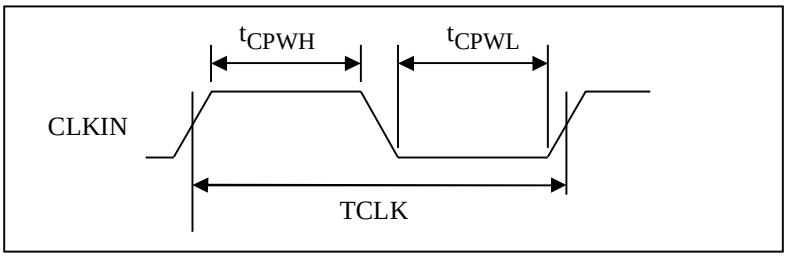

Figure 10: Clock Timings

4.4 CPU Bus Access Timings

The timing diagrams are shown for Read and Write access for all types of CPU. The CPU interface is asynchronous to CLKIN. However, internally the bus access is synchronized to CLKIN, for all Write accesses and for those Read accesses (FIFO, interrupt status, error status) that cause a Write. The successive accesses that cause Write have to be at least 4 clocks apart. This delay is indicated by the delay in RDY (DACKn) output. If the CPU can use RDY (DACKn) signal then it can issue successive access without delay. If the CPU cannot use RDY (DACKn) signal and if the successive bus access cannot be delayed then the CPU has to check completion of the prior access by reading a status register.

If Freescale type CPU with óEôsignal is connected, then replace óDSnôsignal in the timing diagrams by óEôsignal with inverted polarity.

The timings are shown in Figure 11, Figure 12, Table 22, Figure 13 Figure 14 and Table 23. The timing values for output signals are specified for a 50 pf load. All times are in ns unless specified otherwise.

4.4.1 Intel Type CPU

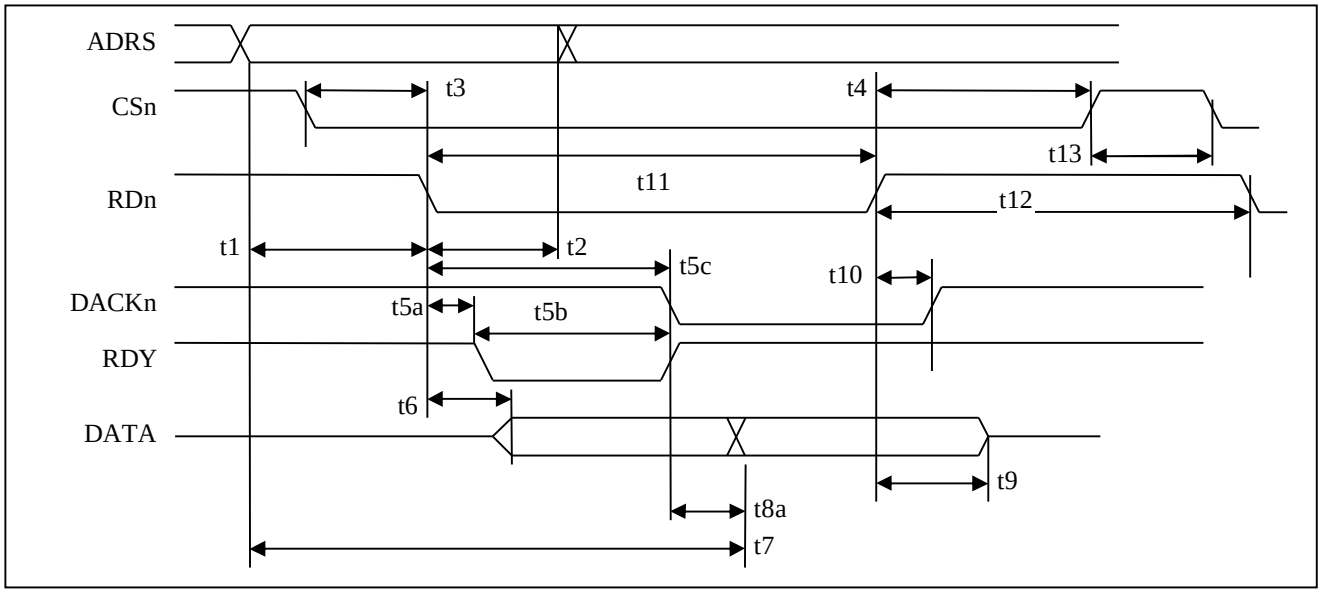

Figure 11: Intel Bus Read Cycle Timing Diagram

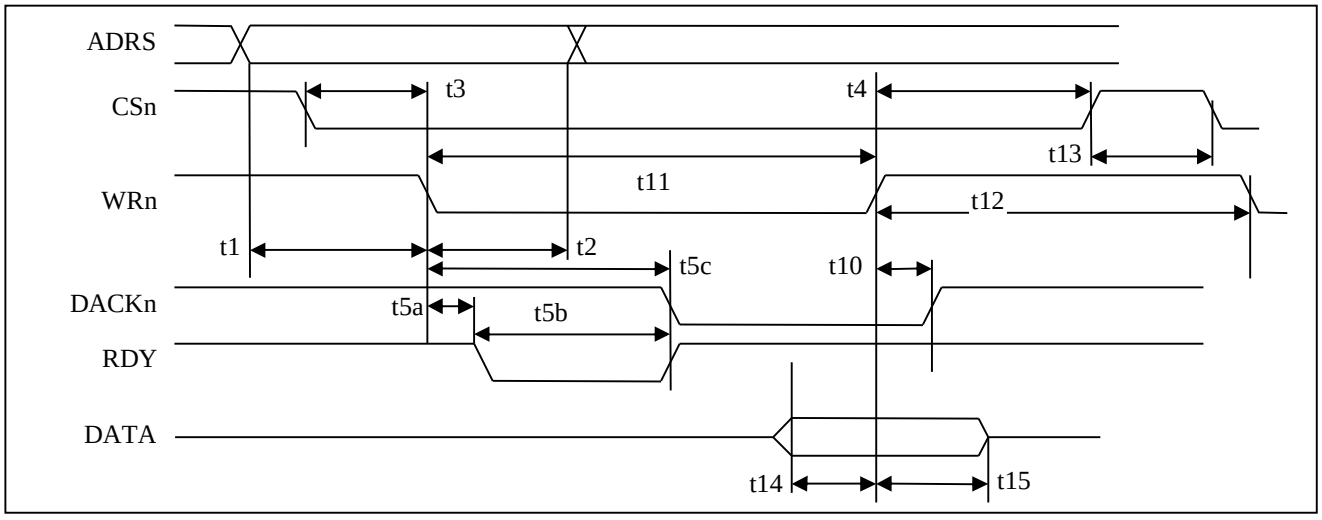

Figure 12: Intel Bus Write Cycle Timing Diagram

Table 22: Bus Timings for Intel type CPU

Num	Description	Min	Max	Notes
t1	Valid Address to RDn, WRn assertion (setup time)	10		
t2	RDn, WRn assertion to invalid Address (hold time)	10		
t3	CSn assertion to RDn, WRn assertion setup time	0		
t4	RDn, WRn negation to CSn negation (hold time)	0		
t5a	RDn, WRn assertion to RDY negation delay		20	3
t5b	RDY negation duration	0	4*TCLK 3*TCLK	1 2
t5c	RDn, WRn assertion to DACKn assertion delay	5	4*TCLK 3*TCLK	1 2
t5d	End of previous cycle to DACKn (RDY) assertion delay		4*TCLK + 30 3*TCLK + 30	3 2
t6	RDn assertion to active Data output delay		20	
t7	Valid address to valid data output access delay		30	
t8a	RDY (DACKn) to valid Data delay		20	3
t8b	Positive edge of CLKIN to valid Data delay		30	3
t9	RDn negation to tristate Data, invalid Data delay	3	15	
t10	RDn, WRn negation to DACKn negation delay	0	15	
t11	RDn, WRn assertion duration	10		
t12	RDn, WRn inactive time	10		
t13	CSn inactive time	10		
t14	Valid Data to WRn negation (setup time)		5	
t15	WRn negation to invalid Data (hold time)	5		

Notes:

1. This delay depends upon the immediately prior cycle. If there is sufficient gap between the two successive cycles, then RDY is not negated; DACKn assertion delay is 20 ns. The Write cycles and some of the Read cycles require that the end of the following cycle be spaced by 4*TCLK.
2. DACKn (RDY) can be asserted one clock earlier, if configured to do so by software. It requires that the delay from DACKn (RDY) assertion to RDn, WRn negation be at least one clock.
3. These delays apply only if RDY is negated. Otherwise, t7 applies. Use t8a only if the CPU cannot use CLKIN to synchronize the Read data input.

4.4.2 Freescale Type CPU

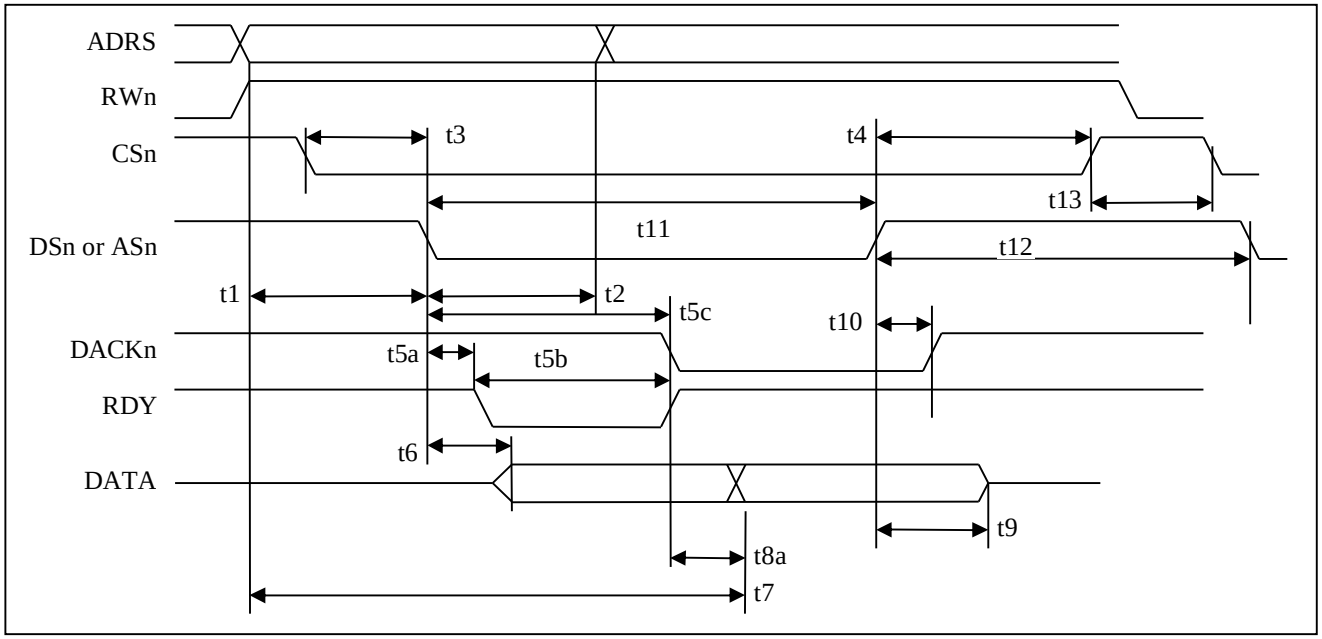

Figure 13: Freescale Bus Read Cycle Timing Diagram

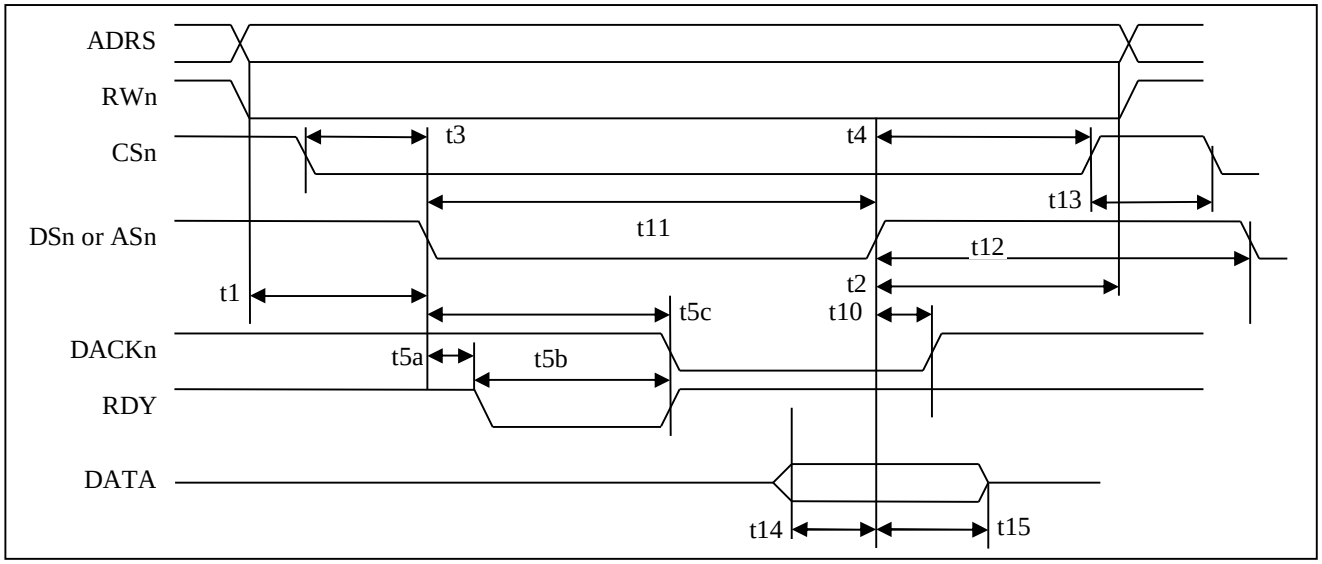

Figure 14: Freescale Bus Write Cycle Timing Diagram

Table 23: Bus Timings for Freescale type CPU

Num	Description	Min	Max	Notes
t1	Valid Address to DSn assertion (setup time)	10		
t2	DSn assertion to invalid Address (hold time)	10		
t3	CSn assertion to DSn assertion setup time	0		
t4	DSn negation to CSn negation (hold time)	0		
t5a	DSn assertion to RDY negation delay		20	3
t5b	RDY negation duration	0	4*TCLK 3*TCLK	1 2
t5c	DSn assertion to DACKn assertion delay	5	4*TCLK 3*TCLK	1 2
t5d	End of previous cycle to DACKn (RDY) assertion delay		4*TCLK + 30 3*TCLK + 30	3 2
t6	DSn assertion to active Data output delay		20	
t7	Valid address to valid data output access delay		30	
t8a	RDY (DACKn) to valid Data delay		20	3
t8b	Positive edge of CLKIN to valid Data delay		30	3
t9	DSn negation to tristate Data, invalid Data delay	3	15	
t10	DSn negation to DACKn (RDY) negation delay	0	15	
t11	DSn assertion duration	10		
t12	DSn inactive time	10		
t13	CSn inactive time	10		
t14	Valid Data to WRn negation (setup time)		5	
t15	WRn negation to invalid Data (hold time)	5		

Notes:

1. This delay depends upon the immediately prior cycle. If there is sufficient gap between the two successive cycles, then RDY is not negated; DACKn assertion delay is 20 ns. The Write cycles and some of the Read cycles require that the end of the following cycle be spaced by 4*TCLK.
2. DACKn (RDY) can be asserted one clock earlier, if configured to do so by software. It requires that the delay from DACKn (RDY) assertion to RDn, WRn negation be at least one clock.
3. These delays apply only if RDY is negated. Otherwise, t7 applies. Use t8a only if the CPU cannot use CLKIN to synchronize the Read data input.

4.4.3 DMA Request Timings

Table 24: DMA Request Timings

Num	Description	Min	Max
t_{RQD}	DMA transfer to inactive RQ delay	$2*T_{CLK}$	$4*T_{CLK}$
t_{RQP}	Pause between two successive RQ pulses	$4*T_{CLK}$	

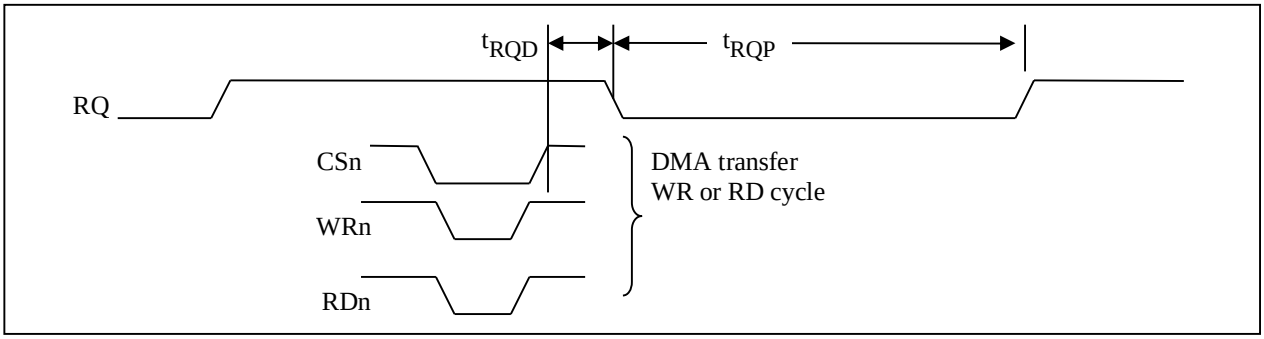

Figure 15: DMA Request Timings

4.5 MAU Interface Timings

The timings are shown in the Table 25 and the Figure 16 ï Figure 18. The timings for output signals are specified for a 50 pf load.

4.5.1 MAU Interface Timings

Table 25: MAU Interface Timings

Num	Description	Min	Nom	Max	Notes
t_{TTR}	Transmit signal output transition time			20 ns	
t_{TSKW}	Skew between transitions in TxEn and TxS outputs			10 ns	
t_{TBIT}	Transmit bit average period		32 µs		1
t_{TXAD}	End of transmission of End Delimiter to TxEn, TxA delay		4 µs		1
t_{RBIT}	Receive signal input average period	30 µs		34 µs	
t_{RTR}	Receive signal input transition time			200 ns	
t_{RJTR}	Receive signal zero crossing jitter			± 2 µs	
t_{LOCK}	Receive lock time			110 µs	2
t_{RxAH}	Receive activity signal hold time	0			

Notes:

1. This time depends upon the correct setting of internal clock. The average bit period has the same tolerance as the clock input tolerance. The bit to bit period jitter will be a small fraction (< 1/32) of the clock input period jitter.
2. The Clock Synchronizer locks to the clock in RxS in maximum of four mid-bit transitions. Therefore, RxA signal has to be active at least 3.5 bit time before the end of Preamble.

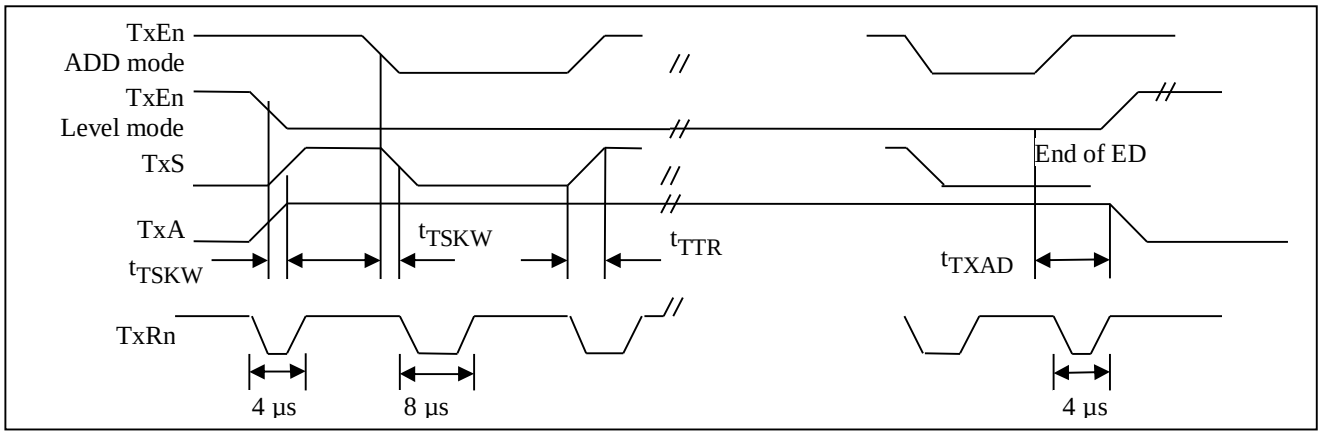

Figure 16: MAU Interface Transmit Signal Timings

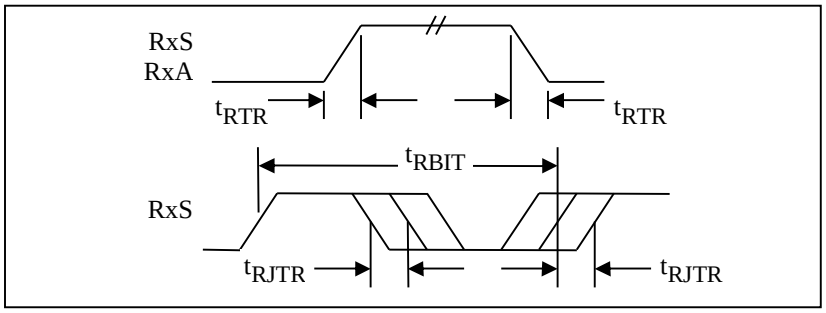

Figure 17: MAU Interface Receive Signal Timings

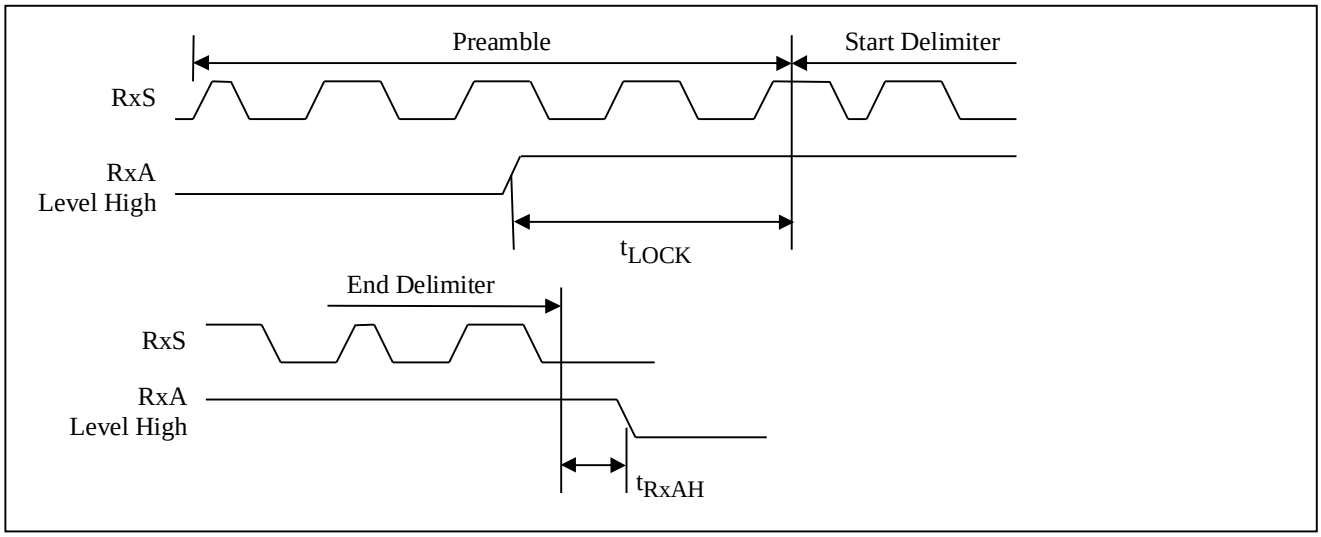

Figure 18: MAU Interface Receiver Timings

4.6 Other Timings

4.6.1 Reset Timings

Table 26: Reset Timings

Num	Description	Min	Max	Notes
t_{RST}	Reset active duration after the active clock	$3*T_{CLK}$		1
t_{RDLY}	Delay from inactive reset input to active chip select	$4*T_{CLK}$		

Notes:

1. RESETn input has to stay active while the power supply voltage is below minimum value.

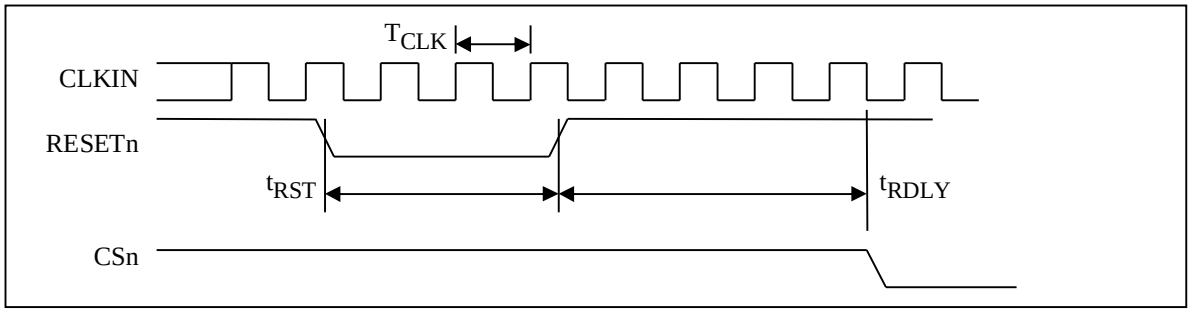

Figure 19: Reset Signal Timings

4.6.2 Interrupt Timings

Table 27: Interrupt Timings

Num	Description	Min	Max	Notes
t_{TXINTD}	Active INTn output to end of transmission delay	7 µs	10 µs	1
$t_{RXINTD1}$	End delimiter in RxS to active INTn output delay	6 µs	14 µs	2
$t_{RXINTD2}$	RQ end to active INTn output delay	1 µs	6 µs	3

Notes:

1. INTn becomes active before TxEn becomes inactive. The CPU can try to start the next transmission before TxEn for the current transmission becomes inactive and the next transmission is queued properly.

2. If end delimiter is not detected and RxA becomes inactive, then this delay is from the end of activity. If DMA is enabled, then the interrupt becomes active only after the receive FIFO becomes empty. The last received byte is available for transfer at the start of end delimiter. Therefore, in DMA mode, the FIFO is likely to become empty before the end delimiter is detected.
3. If DMA is enabled and the receive FIFO does not become empty until after the end delimiter is detected, then INTn becomes active this delay after last RQ.

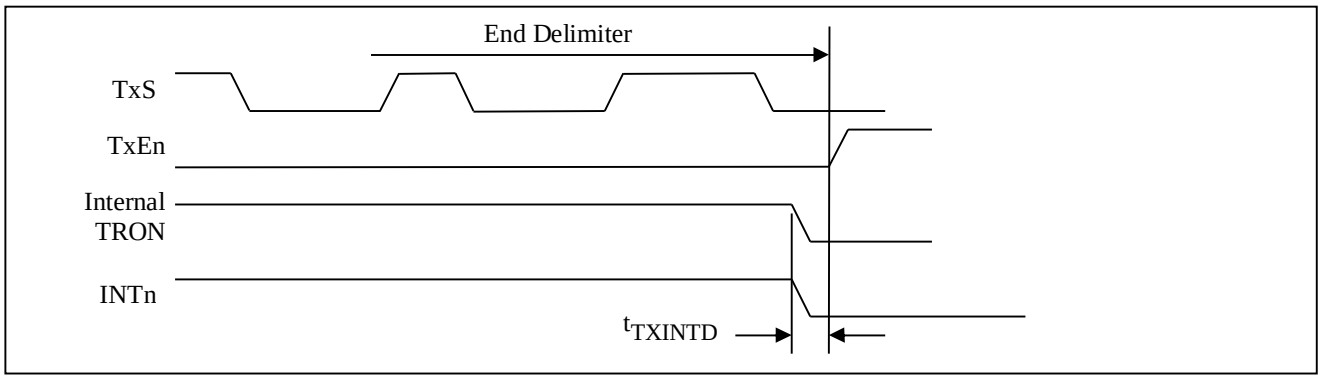

Figure 20: TED Interrupt Timings

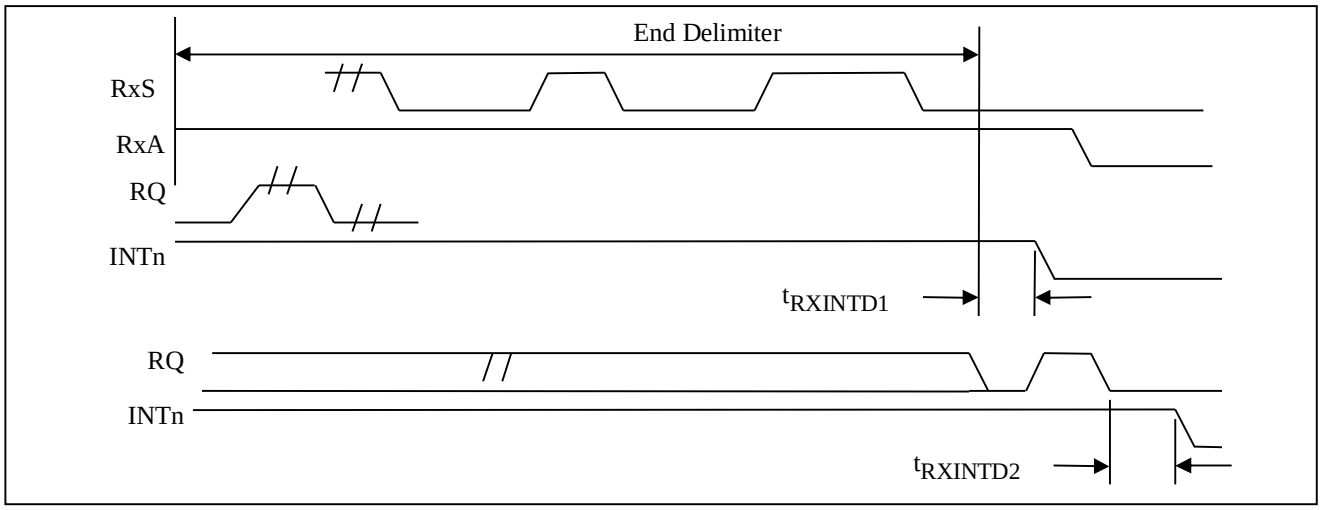

Figure 21: RED Interrupt Timings

5 PACKAGE

UFC100-L2 is available in 44 pin RoHS certified LQFP package. The dimensions are shown in Figure 22.

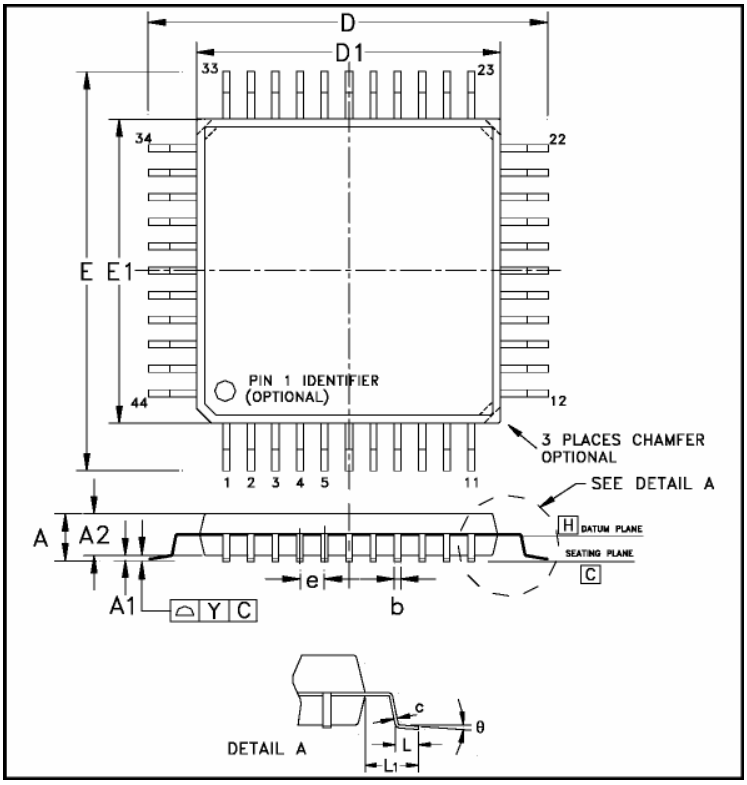

Figure 22: LQFP package dimensions

Table 28: LQFP package dimensions

Symbol	MILLIMETER			INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A			1.60			0.063
A1	0.05		0.15	0.002		0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.22	0.30	0.38	0.009	0.012	0.015
c	0.09		0.20	0.004		0.008
D		12.00 BSC.			0.472 BSC.	
E		12.00 BSC.			0.472 BSC.	
e		0.80 BSC.			0.031 BSC.	
D1		10.00 BSC			0.393 BSC.	
E1		10.00 BSC			0.393 BSC.	
L	0.45	0.60	0.75	0.018	0.024	0.030
L1		1.00 REF			0.039 REF	
Y		0.10			0.004	
ď	0^o		7^o	0^o		7^o

Symbol	Parameter	Value	Units
Rth j-a	Thermal resistance from junction to ambient with no airflow	39.4	^{0} C/W
MSL	Moisture sensitivity level	3	
Reflow temperature	Maximum reflow soldering temperature	260	^{0} C

6 TEST PROCEDURES

The procedures to test the hardware are described. These procedures can also be used in device driver with some changes to suit the product software.

6.1 Register access

If the hardware interface to the CPU uses RDY signal, then the registers can be accessed as memory locations. If the RDY is not used, but Read and Write cycles are at least 4 cycles of the CLKIN input to the UFC100-L2, the registers can be accessed as memory locations. Else, the software should poll ARDY from Status register; and it should Read or Write only if ARDY is ó1ô

6.2 Initialization

The UFC100-L2 requires following settings before it can be used for transmission or reception.

1. Write 0x80 to Reset register (0x00).
2. Write to Mode register (0x01). The value depends upon the frequency of the clock input. The internal clock has to be 500 kHz ï see CLOCK in 2.2.4. Set other fields as following:

   LB = 0, FDP = 1 (for test purpose), PRE = 1 (for test purpose), TFCS = 0, TMD = 0 (assuming using MAU with Enable mode).
3. Write to Interrupt mask (0x06) and Error mask (0x07) registers to enable interrupt, if required.
4. Write to FIFO control register (0x0B) according to software need. For transmission, it is better to set high threshold. For reception, it is better to set low threshold, so that the incoming frame FC and address can be checked. After checking these two bytes in the received frame, it is better to set high threshold to reduce the interrupts. For hardware test, use any threshold setting.
5. Write to GAP time register (0x14). The value depends upon the Data Link layer configuration. For test purpose, set it to 0x10 (8 bytes).
6. Write to Watch time register (0x16, 0x17). The value depends upon the Data Link layer configuration. For test purpose, set it to 0x20 (32 bytes).
7. Clear Watch-timer by writing 0x02 to Timer control register (0x1A).

6.3 Frame transmission

The procedure for transmitting one frame is described. It should be used for testing the hardware including UFC100-L2 and external MAU. The product software may use slightly different procedure e.g. it may use Half-duplex. These procedures do not use interrupt.

6.3.1 Using non-DMA Write to the Transmit FIFO

1. Setup a buffer in the memory to store data bytes to be transmitted. Keep the data pattern such that any shorts in the data bus can be detected. For example 256 bytes with value from 0x00 to 0xFF can be stored.
2. Write to Mode register if necessary to update LB, FDP, PRE and TFCS. Note that this register may have been set to non-zero value during initialization.
3. Clear the FIFO status register by writing ó1ô to CTF in FIFO control register (0x0B).
4. Write the number of bytes to be transmitted in Transmit frame length register (0x08, 0x09).
5. Write few bytes in the FIFO data register (0x0D), so that delay in software loop does not cause underrun.
6. Write 0x04 to Control register (0x02) (TRON is set to ó1ô). Note that the Receiver is disabled.
7. Poll TFRY in Status register (0x03). As long as this is ó1ô another byte can be written to the FIFO data. Write all of the bytes in the frame to be transmitted. The number of bytes to be written in step 5 and this step has to be equal to the value written in the step 4. Otherwise, underrun or TLM error occurs.

8. Poll ERS and TED in the Interrupt status register (0x04). If TED is ó1ô then it is the end of successful transmission. If ERS is ó1ô then there was an error.
9. If there was an error then first read FIFO status register (0x0C) and then Error status register (0x05) to find the reason for the error. Clear the FIFO status register by writing ó1ô to CTF in FIFO control register (0x0B).

You can use an oscilloscope to check that the frame was transmitted. If RxA input was ó1ô before starting the transmission then the UFC waits until RxA becomes ó0ô before starting the transmission. Since the MAU itself receives the transmitted signal, the RxA input to the UFC100-L2 should be ó1ô during the transmission after a delay. If RxA remain ó0ô during the transmission then it is considered an error, CNS in Error status register is set to ó1ô and the transmission is aborted.

You can deliberately cause error such as underrun to test error handling.

6.3.2 Using DMA Write to the Transmit FIFO

1. Setup a buffer in the memory to store data bytes to be transmitted. Keep the data pattern such that any shorts in the data bus can be detected. For example 256 bytes with value from 0x00 to 0xFF can be stored.
2. Write to Mode register if necessary to update LB, FDP, PRE and TFCS. Note that this register may have been set to non-zero value during initialization.
3. Clear the FIFO status register by writing ó1ô to CTF in FIFO control register (0x0B).
4. Write the number of bytes to be transmitted in Transmit frame length register (0x08, 0x09).
5. Setup DMA controller to Write the number of bytes equal to the value written in step 4.
6. Write 0x84 to Control register (0x02) (DMA and TRON are set to ó1ô). Note that the Receiver is disabled.
7. Poll ERS and TED in the Interrupt status register (0x04). If TED is ó1ô then it is the end of successful transmission. If ERS is ó1ô then there was an error.
8. If there was an error then first read FIFO status register (0x0C) and then Error status register (0x05) to find the reason for the error. Clear the FIFO status register by writing ó1ô to CTF in FIFO control register (0x0B).

The DMA transfer starts as soon as TRON is set to ó1ô

6.4 Frame reception

It is necessary to send a frame to the device under test. This can be done either by an external source or by transmitting the frame and receiving it in the same device. The internal loopback is not used so that external MAU can be tested. The procedure for transmitting and receiving one frame is described. It should be used for testing the hardware including UFC100-L2 and external MAU. The product software may use slightly different procedure e.g. it may use Half-duplex. These procedures do not use interrupt.

6.4.1 Using non-DMA access to the FIFO

1. Setup a buffer in the memory to store data bytes to be transmitted and another buffer to store the received data bytes.
2. Set FDP in Mode register (0x01) to ó1ô to enable Full-duplex mode. Note that this register may have been set to non-zero value during initialization.
3. Clear the FIFO status register by writing ó1ô to CRF and CTF in FIFO control register (0x0B).
4. Write the number of bytes to be transmitted in Transmit frame length register (0x08, 0x09).
5. Write few bytes in the FIFO data register (0x0D), so that delay in software loop does not cause underrun.
6. Write 0x44 to Control register (0x02) (RE and TRON are set to ó1ô). Note that the Receiver is enabled.
7. Poll TFRY and RFRY in Status register (0x03). As long as TFRY is ó1ô another byte can be written to the FIFO data. If RFRY is ó1ô then the received data byte can be read from the FIFO data and stored in the memory buffer. Write all of the bytes in the frame to be transmitted. The number of bytes to be written in step 5 and this step has to be equal to the value written in the step 4. Otherwise, underrun or TLM error occurs.
8. Poll Interrupt status register and Status register. If RFRY is ó1ô then read the byte from the FIFO data. If RED is ó1ô then it is the end of successful reception and stop polling. If ERS is ó1ô then it indicates an error.

9. If there was an error then first read FIFO status register (0x0C) and then Error status register (0x05) to find the reason for the error. Clear the FIFO status register by writing ó1ô to CRF and CTF in FIFO control register (0x0B).
10. Compare the received data bytes with transmitted bytes to verify that the correct frame was received.

6.4.2 Using DMA access to the FIFO

The DMA access can only be used for transmission. The reception has to be done by polling RFRY.

1. Setup a buffer in the memory to store data bytes to be transmitted and another buffer to store the received data bytes.
2. Set FDP in Mode register (0x01) to ó1ô to enable Full-duplex mode. Note that this register may have been set to non-zero value during initialization.
3. Clear the FIFO status register by writing ó1ô to CRF and CTF in FIFO control register (0x0B).
4. Write the number of bytes to be transmitted in Transmit frame length register (0x08, 0x09).
5. Setup DMA controller to write the number of bytes equal to the value written in step 4.
6. Write 0xC4 to Control register (0x02) (DMA, RE and TRON are set to ó1ô). Note that the Receiver is enabled.
7. Poll Interrupt status register and Status register. If RFRY is ó1ô then read the byte from the FIFO data. If TED is ó1ô then it is the end of successful transmission. If RED is ó1ô then it is the end of successful reception and stop polling. If ERS is ó1ô then it indicates an error.
8. If there was an error then first read FIFO status register (0x0C) and then Error status register (0x05) to find the reason for the error. Clear the FIFO status register by writing ó1ô to CRF and CTF in FIFO control register (0x0B).
9. Compare the received data bytes with transmitted bytes to verify that the correct frame was received.

6.5 Watch timer

This procedure is used to test Watch-timer and also check that the INTn signal is working.

1. Transmit a frame using the procedure in 6.3.
2. After transmission setup Watch-timer interrupt by writing 0xBF to Interrupt mask register (0x06).
3. Set IE in Control register (0x02) to ó1ô.
4. Wait for a duration which is longer than 256 x (value in Watch-time register) microseconds.
5. Check that INTn becomes low.
6. Poll Interrupt register and check that RTI becomes ó1ô.

Note that when Interrupt register is read, RTI will be reset to ó0ô and INTn will become high.

If this procedure ï transmission of a frame and waiting for Watch-timer interrupt, is put in a loop then you can use oscilloscope to monitor INTn. Also check that the gap between two transmissions is equal to 128 x (value in GAP time register) µs.

6.6 Token timer

The Token timer does not affect the external hardware and thus it is not required to test the hardware. Its operation can be tested by using the procedure described below.

1. Write a value larger than the transmission frame length in byte to Token counter (0x18, 0x19). Since it is 16-bit value a much larger value can be written. Note that the least significant byte (0x18) has to be written first.
2. Transmit a frame using the procedure in 6.3.
3. After transmission write ó1ô to LTR in Timer control register (0x1A).
4. Read Token counter (0x18, 0x19) and check that its value is approximately equal to (initial value ï number of byte transmitted ï 5). The subtraction of õ5ô is for Preamble (1 byte), Start Delimiter, FCS (2 bytes) and End Delimiter.