SMT229

Ethernet TRAM User's Guide

Overview

This document covers the usage of the Sundance SMT229 Ethernet TRAM, both from an external hardware view, and from a programmer's view in sufficient detail to allow bespoke driver software to be developed if necessary.

SUNDANCE will can supply Packet Driver (CPD) or Helios® device driver software with this TRAM. In addition, extensive programming support for standalone TCP/IP under 3L Parallel C and Inmos ANSI C is available. CPD can be supplied for most common Transputer programming environments, including Inmos ANSI C, OCCAM Toolset, 3L Parallel C and Helios C. Software is documented separately.

Intended audience

By their very nature, TRAMs such as the SMT229 requires some detailed knowledge of the Transputer and the TRAM standard in order to configure a working system. This level of expertise is assumed.

On the whole, Ethernet provides a relatively simple plug-and-play interface at the hardware level. Most of the complexity is in the network layers that sit above the hardware. The techniques and pitfalls associated with Ethernet networking are beyond the scope of this document, so some familiarity with this subject is also assumed.

The CPD library software provides a straightforward interface to the SMT229 similar to the Clarkson Packet Driver interface developed for the IBM PC. Although a great deal of additional work is required to interwork with say, OSI or TCP/IP, CPD does provide an efficient, full-featured interface to the hardware. The Helios device driver software implements the standard device driver interface to Ethernet under Helios, which can either be used with the Helios TCP/IP and NFS implementations, or directly with custom user applications.

The Ethernet implementation provided by the SMT229 is a high-specification interface, with sufficient on-board RAM to implement a full protocol stack locally. The programmer's information in this manual is intended to be read in conjunction with the appropriate Transputer, Am79C900 Ethernet etc. documentation if direct programming of the hardware is contemplated.

Description

Physical

The SMT229 Ethernet TRAM conforms to the dimensions and profile of the size 2 TRAM layout defined in Inmos Technical Note 29, with the exception that vertical stacking is not possible in position 1 due to interference from the Ethernet transceiver connector (AUI) and associated components.

Dimensions	
Length:	92.96mm
Width:	53.33 mm
Height:	10.50 mm (above PCB, including A UI connector)
Electrical	
Power supply:	
Voltage:	5.0V dc +/- 5%
	12.0V dc (optional, for transceiver)
Power dissipation	
20MHz processor:	2.4W
25MHz processor:	2.5W

Ethernet AUI (Optional)

5V sub-miniature transceiver (MAU)

12V dc-dc converter, for driving 12V MAUs

Device complement

T400/T425, 4Mbyte DRAM

AMD ILACC (Am79C900), with 32 bit DMA support

Xicor 24C01 64 byte NVRAM for on-board Ethernet address

Ethernet installation

The commissioning of an Ethernet network is beyond the scope of this manual, but the SMT229 presents no special additional network installation problems.

GSL sub-miniature transceiver

A common solution to the packaging of thin-wire Ethernet interfaces is to integrate the transceiver with the interface card. This works well in layouts such as the IBM PC, where the card edge can be very close to the network T-piece. Where the interface card is a TRAM, it is unlikely that the edge of the board can be close enough to connect directly in this way. Solutions that involve running the actual network cable over the TRAM motherboard to a T connection inside a chassis provoked screams of horror from our network management specialists, so we have not provided an on-board transceiver!

A solution involving a separate transceiver is the alternative, although TRAMs cannot directly supply the required 12V power. Also, typical thin-wire

transceivers are too bulky for a neat installation within the same enclosure as the TRAM motherboard.

Figure 1 SMT229 AUI Cable

Figure 2 SMT229 Connector Pinouts

We have solved this problem, by implementing a compact, 5V-supply, thin-wire transceiver. It can be connected to the SMT229 directly using twisted-pair ribbon and IDC connectors with no external power supply. Maximum recommended drop cable length is 50cm. The sub-miniature transceiver is light and compact enough to hang on the back of the network BNC connector itself. In an IBM PC chassis for example, we can supply a "hanger" to mount the transceiver in an empty slot.

When using the GSL transceiver, a jumper should be fitted between pins 3 and 4 of the external AUI power connector to supply +5V to the main connector.

Third party transceiver/DC-DC converter

If the network is not thin-wire, or if for some other reason the GSL transceiver is not to be used, the AUI connector on the SMT229 has the correct pin layout to convert to a standard 15 pin AUI connector using twisted-pair ribbon cable and IDC connectors (see Figure 1).

In this configuration, the third party transceiver will expect 12V power through the connector. If 12V is not available elsewhere on the TRAM motherboard host, we can factory-install a small DC-DC converter piggyback board that will supply 12V to the AUI connector. It can supply up to 1.5W to a standard transceiver.

Third party transceiver/external 12v source

If a third party transceiver is used, and 12V is available elsewhere on the TRAM host, it can be wired to pins 1 and 4 of the external AUI power connector as shown in figure 2. This will then provide the 12V supply through the AUI connector. The external 12V supply is assumed to have a common ground reference with the TRAM +5V supply.

TRAM installation

The SMT229 is a standard, size 2 TRAM with subsystem control. It is compatible with most TRAM motherboards such as the Inmos B008. It responds to the LinkSpeed and Reset/Analyse pins as specified in Inmos Technical Note 29. Extended LinkSpeed options are available as detailed in Table 1.

A standard size 1 TRAM can be stacked vertically in position 2 on this TRAM. In situations where this TRAM will be the boot device, we recommend the HPT21 Flash EEPROM TRAM or the HPT20 PROM TRAM in this position. The extended feature pins in position 2 will allow the SMT229 to be booted whilst presenting all links to the motherboard as uncommitted once the boot image is transferred.

Programming model

The SMT229 is available with comprehensive drivers for most common languages and environments, including Inmos ANSI C, OCCAM Toolset, 3L Parallel C and Helios C. Whilst these should solve most problems without recourse to device-level programming, the following provides additional low-level information should the need arise.

Figure 3 Transputer Memory Map

Transputer memory map

The two 79C900 registers are 16 bits wide. Hence the data port and register address ports are mapped into the bottom 16 bits of the 32 bit words at addresses 4000000016 and 4000000416 respectively. Example program fragments, showing 79C900 register access in C and OCCAM are shown in Figures 4 and 5.

Figure 4 79C900 Register Access Example (C)

To provide a unique, non-volatile Ethernet address for each SMT229, it is equipped with a Xicor 24C01 1024 bit NVRAM. The data line from the 24C01 can be read/written from/to address 2000000016, bit 0. The write line driver is an inverting, wired-or configuration. Hence, write data should be inverted, and address 2000000016 should be written with 0x00 to allow the read 'high' state to be interpreted correctly. The clock line can be controlled by writing to 2000000416, bit 0. Refer to the Xicor 24C01 data sheet for information on manipulating the NVRAM to read the data. The Ethernet address for the part is supplied in bytes 0-5 of the NVRAM. Bytes 6-31 are reserved. Bytes 32-63 are unused. Example program fragments, showing 24C01 clock and data access in C and OCCAM are shown in Figures 6 and 7.

The TRAM Error/Reset/Analyse registers are mapped as detailed in Inmos Technical Note 29. The parity register is not used.

FFFFFFFF16	Reserved for RAM
8040000016 - 803FFFFF16	4092kbytes external RAM
8000100016	4kbytes internal RAM
7FFFFFFF16	Reserved

4000000816	
4000000816	
4000000416	79C900 Reg.Addr.Ptr. (LS. 16 bits)
4000000016	79C900 Data Port (LS. 16 bits)
3FFFFFFF16	Reserved
2000000816	
2000000416	Write: bit 0 - 24C01 data (inverted)
	Read: bit 0 - 24C01 data
2000000016	Write: bit 0 - 24C01 clock
1FFFFFFF16	Reserved
0000000816	
0000000416	Write: bit 0 = SubSystem Analyse
0000000016	Write: bit 0 = SubSystem Reset
	Read: bit 0 = SubSystem NotError

```
INT INLINE FUNCTION ILACC.Read.Reg(VAL INT csr)
    [2]Int ILACC:
    PLACE ILACC AT -#40000000:
    INT ILACC.data IS ILACC[0]:
    INT ILACC.rap  IS ILACC[1]:
    VALOF
      ILACC.rap := csr
      RESULT( ILACC.data /\ #FFFF)
:
INLINE PROC ILACC.Write.Reg(VAL INT csr,data)
    [2]INT ILACC:
    PLACE ILACC AT -4#0000000:
    INT ILACC.data IS ILACC[0]:
    INT ILACC.rap  IS ILACC[1]:
    SEQ
      ILACC.rap,ILACC.data := csr,data
:
```

Figure 5 79C900 Register Access Example (Occam)

```
#define X24C01_DATA  ((int *)0x20000000)
#define X24C01_CLOCK ((int *)0x4000004)
int X24C01_Read_Data(void)
{
      return (*X24C01_DATA & 1) == 0;
}
void X24C01_Write_Data(int flag)
{
    *X24C01_DATA = (flag) ? 0 : 1;
}
void X24C01_Write_Clock(int flag)
{
    *X24C01_CLOCK = (flag) ? 1 : 0;
}
```

Figure 6 Xicor 24C01 Register Access Example

```
#define ILACC_DATA ((unsigned *)0x40000000)
#define ILACC_RAP  ((unsigned *)0x4000004)
unsigned ILACC_Read_Reg(unsigned csr)
{
      *ILACC_RAP = csr;
      return (*ILACC_DATA) & 0xFFFF;
}
void ILACC_Write_Reg(unsigned csr, unsigned value)
{
    *ILACC_RAP  = csr;
    *ILACC_DATA = value;
}
```

```
BOOL INLINE FUNCTION x24C01.Read.Data()
    [2]INT X24C01:
    PLACE X24C01 AT -#20000000:
    INT   X24C01.Data IS X24C01[0]:
    VALOF
      SKIP
      RESULT (X24C01.Data /\ #1) = 0
:
INLINE PROC X24C01.Write.Data(BOOL flag)
    [2]INT X24C01:
 PLACE X24C01 AT -#20000000:
    INT   X24C01.Data IS X24C01[0]:
    SEQ
      IF
        (flag)
          X24C01.Data := 0
        TRUE
          X24C01.Data := 1
:
INLINE PROC X24C01.Write.Clock(BOOL flag)
    [2]INT X24C01:
 PLACE X24C01 AT -#20000000:
    INT   X24C01.Clock IS X24C01[1]:
    SEQ
      IF
        (flag)
          X24C01.Clock := 1
        TRUE
          X24C01.Clock := 0
:
```

Figure 7 Xicor 24C01 Register Access Example (Occam)

ILACC memory map

The Ethernet interface processor fitted is an AMD 79C900. It contains a 32 bit master mode DMA engine and extensive microcode. The device is normally operated by setting up transmit and receive ring buffers in memory, and patching up the list pointers as interrupts signal packets arriving or as packets need to be sent. The address map visible to the 79C900 is shown

in Figure 8. To simplify DMA address programming, most addresses will have the same representation on the Transputer and 79C900.

Figure 8 79C900 Memory Map

However, addresses from 8000000016 to 80000FFF16 represent internal RAM to the Transputer, but external RAM to the 79C900. If the 79C900 is set to perform DMA to/from this address range, it will not access internal memory the Transputer sees mapped in these addresses.

Interrupts

The 79C900 INTR and RINTRlines are ORed to the Event handshake on the Transputer. A 79C900 interrupt appears as a single word on the Event channel. Interrupt acknowledge is achieved by writing a 1 to the appropriate bit in CSR0.

The Transputer event line is effectively edge-triggered in this application, so it is possible for overlapping transmit and receive interrupts to mask each other. To avoid missing interrupts, we recommend polling CSR0 and CSR4 and processing any pending conditions immediately before waiting on the Event channel. Once a read of CSR0/CSR4 shows no outstanding interrupt conditions, it is safe to wait on the Event channel because any subsequent interrupts must cause a significant transition on the EventReq line.

SMT229-specifics

The hardware implementation of the Ethernet TRAM using the 79C900 has certain consequences for register-level programming of the device. These are as follows:

Bus Control

The SMT229 uses the 80x86 bus-control and byte-ordering modes, so CSR4 ACON should be 00.

Address strobe

The ALE/AS strobe is active low, so CSR3 ACON should be 0.

DMA burst length

The SMT229's memory bandwidth is far higher than the Ethernet line data rate, so the 79C900 DMA FIFO will fill/empty very quickly once a DMA is established. The Transputer DRAM refresh support will not be compromised by any conceivable DMA transfer length, so for best performance CSR4 DMAPLUS should be 1.

Transmit mode

The SMT229 AUI connection is transformer-coupled, so CSR15 TSEL should be 0.

Port Select

The SMT229 is designed to use the internal SIA only, so CSR15 PORTSEL should be 0.

FFFFFFFF16

Reserved for RAM

0040000016

003FFFFF16

4096kbytes RAM

000000016

Issue date : 23 May 1992

Reference : 702/ENET/SUM/01

Version : 1

Trademarks

3L is a registered trademark of 3L Ltd.

OCCAM, Inmos and IMS are trademarks of Inmos Ltd.

Helios is a registered trademark of Perehelion Software Ltd.

Copyright