ARM710T Header Card

(KPI-0032A)

User Guide

ARM

ARM DUI 0114A

ARM710T Header Card User Guide

Release information

Change history

Date	Issue	Change
27th April 1999	A	Release

Proprietary notice

Document confidentiality status

This document is Open Access. This document has no restriction on distribution.

Product status

The information in this document is Final (information on a developed product).

ARM web address

`http://www.arm.com`

Preface

This preface introduces the ARM710T Header Card and its reference documentation. It contains the following sections:

About this document

This document is the ARM710T Header Card User Guide.

Intended audience

This document has been written for experienced hardware and software engineers who wish to use an ARM710T header card, with their ARM development board, for code development and evaluation.

Organization

This document is organized into the following chapters:

Chapter 1 *Overview of the ARM710T Header Card*

Read this chapter for an introduction to the ARM710T Header Card.

Chapter 2 *Setting up your System*

Read this chapter for a description of how to set up the ARM development board to work with the ARM710T Header Card.

Chapter 3 *Configuring the ARM710T Header Card*

Read this chapter for a description of how to configure the ARM710T Header Card.

Chapter 4 *Circuit Descriptions*

Read this chapter for a description of the circuit board of the ARM710T Header Card.

Typographical conventions

The following typographical conventions are used in this document:

bold Highlights ARM processor signal names within text, and interface elements such as menu names. May also be used for emphasis in descriptive lists where appropriate.

italic Highlights special terminology, cross-references and citations.

`typewriter` Denotes text that may be entered at the keyboard, such as commands, file names and program names, and source code.

`typewriter` Denotes a permitted abbreviation for a command or option. The underlined text may be entered instead of the full command or option name.

`typewriter italic`
Denotes arguments to commands or functions where the argument is to be replaced by a specific value.

`typewriter bold`
Denotes language keywords when used outside example code.

Further reading

This section lists publications by ARM Limited.

ARM publications

For additional information, refer to the following:

- *ARM Target Development System User Guide* (ARM DUI 0061*)*.
- *ARM Multi-ICE User Guide* (ARM DUI 0048).
- *ARM Multi-ICE Installation* Guide (ARM DSI 0005).
- *ARM Software Development Toolkit User Guide* (ARM DUI 0040).
- *ARM710T Datasheet* (ARM DDI 0086).

Feedback

ARM Limited welcomes feedback both on the ARM710T header card, and on the documentation.

Feedback on this document

If you have any comments on this document, please send email to `errata@arm.com` giving:

- the document title
- the document number
- the page number(s) to which your comments refer
- a concise explanation of your comments.

General suggestions for additions and improvements are also welcome.

Feedback on the ARM710T header card

If you have any comments or suggestions about this product, please contact your supplier giving:

- the product name
- a concise explanation of your comments.

Contents
ARM710T Header Card User Guide

Chapter 1
Overview of the ARM710T Header Card

This chapter introduces the ARM710T header card and contains the following sections:

1.1 Introduction to the ARM710T header card

The ARM710T header card (part number KPI-0032A) is a processor daughter board for the ARM development board (HBI-0011B). This guide describes how to set up your ARM development board and ARM710T header card.

The header card and development board combination provides a suitable platform for code development and evaluation of the ARM710T processor.

Together with the *ARM Software Development Toolkit*, the user can download, execute and debug code. This can be with either the Multi-ICE debugging system, available separately from ARM, or the Angel debug monitor.

1.1.1 System requirements

To use the ARM710T header card you will need the following:

- ARM development board (HBI-0011B)
- ARM debugger, such as:
  - Multi-ICE (recommended)
  - Angel.
- ARM Software Development Toolkit.

1.1.2 Features of the ARM710T header card

The main features of the ARM710T header card are:

- selectable core frequency up to 40MHz
- logic analyzer connectors for every signal of the ARM710T
- Multi-ICE connector for debugging using the EmbeddedICE macrocell.

1.2 Board layout

Figure 1-1 shows the layout of the main components of the ARM710T header card.

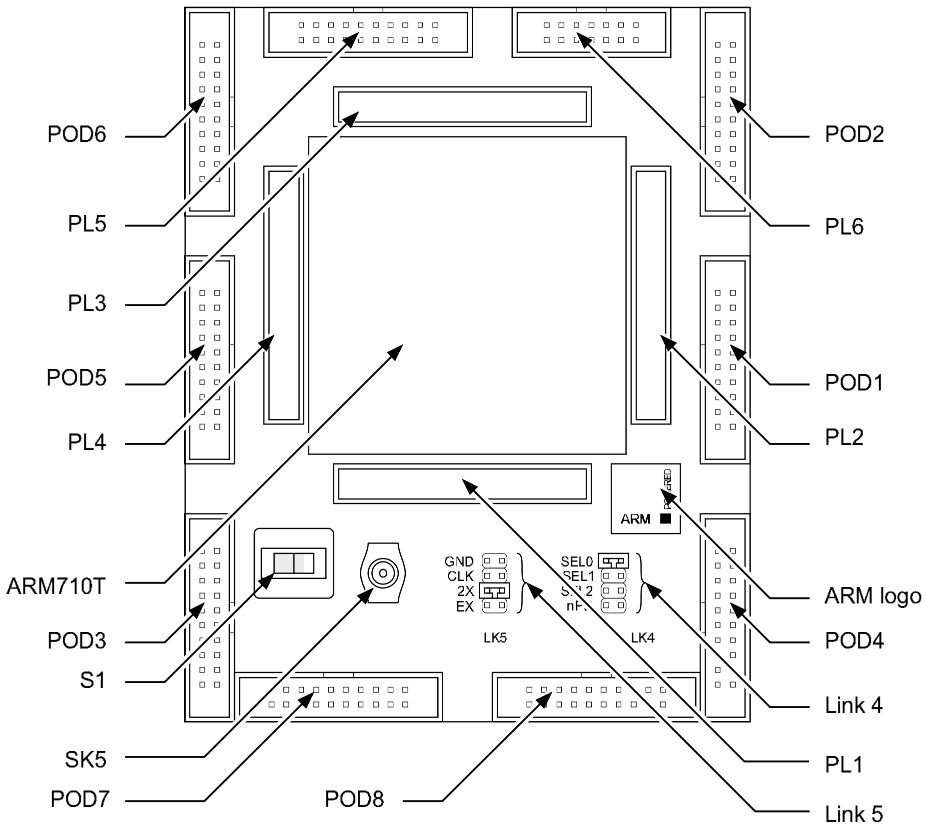

Figure 1-1 Board layout

1.2.1 Schematic diagrams

The complete set of schematic diagrams for the ARM710T header card is supplied in pdf format on the accompanying disk.

Chapter 2
Setting up your System

This chapter describes how to set up your ARM development board to work with the ARM710T header card and gives details of connecting debuggers. It contains the following sections:

2.1 Setting up your ARM development board

The ARM710T header card plugs into the top left hand corner of the development board, with the ARM logo on the silk-screen positioned as shown in Figure 2-1, so that the header is flush with the development board. The correct way to mount the header card is shown in Figure 2-1.

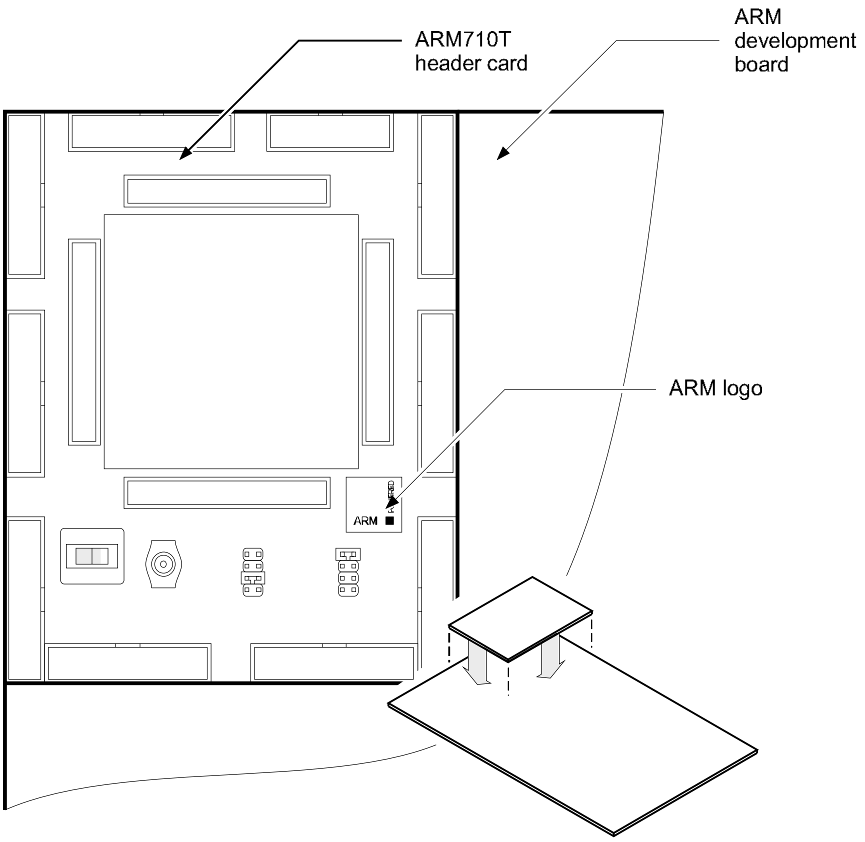

Figure 2-1 Mounting the header card

The ARM710T can be debugged using one of the following:

- Multi-ICE debugging system (available separately)
- EmbeddedICE debugging system (available separately)
- Angel debug monitor (supplied with the ARM development board).

——— **Note** ———

EmbeddedICE can be used to debug an ARM710T but does not support debugging with the caches on.

2.1.1 Setting the bus clock frequency

The ARM710T header card requires the bus clock frequency, **BCLK**, of the board to be set to 4, 8, 16 or 20MHz. This is set by changing the FREQ SELECT switch, S1, on the ARM development board, shown in Figure 2-2.

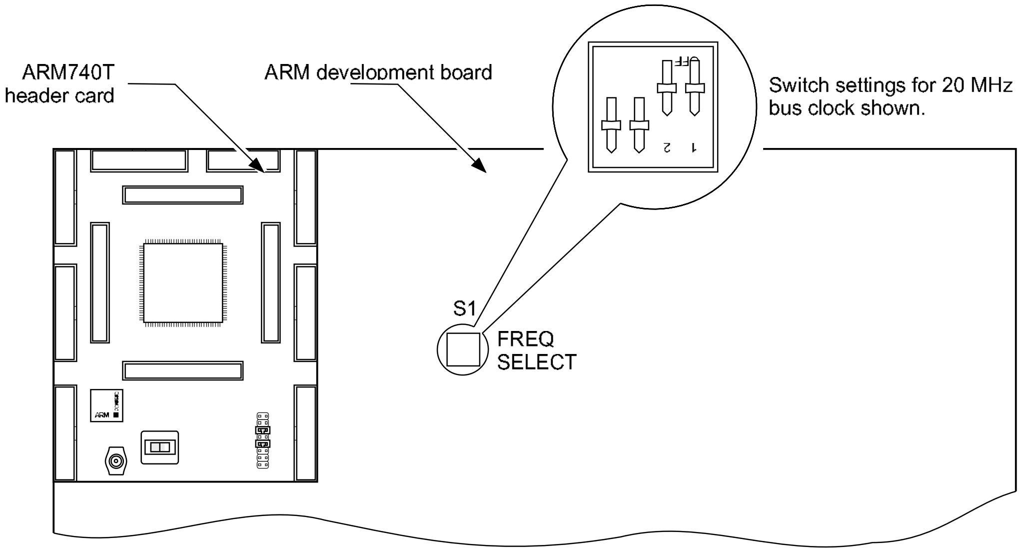

Figure 2-2 Position of switch S1

Table 2-1 shows the switch settings for the allowable bus clock frequencies.

Table 2-1 Bus clock settings

Position 4	Position 3	Position 2	Position 1	BCLK (MHz)
On	On	On	On	4
On	On	On	Off	8
On	On	Off	On	16
On	On	Off	Off	20

Note

Do not select any combination other than shown in the table.

2.2 Debugging using Angel

Angel is a program that enables rapid development and debugging of applications running on ARM-based hardware. Angel runs on the ARM development board or a development version of the product hardware alongside your application. It communicates with a debugger that can handle the Angel communications protocol, such as the *ARM Debugger for Windows*, or *armsd*. A serial/parallel or ethernet connection to the host debugging system is required.

The binary image for the Angel is supplied with the ARM development board. To download, execute and debug code the host needs to be running the *ARM Debugger for Windows* or *armsd* program supplied as part of the *ARM Software Development Toolkit*, available separately from ARM.

——— **Note** ———

If debugging with Angel is required, jumper links on the ARM development board may need to be changed. Please refer to the manual *ARM Target Development System User Guide* for details of how to establish a debug link between the board and the host system.

———————

2.3 Debugging using Multi-ICE

The Multi-ICE debugging system provides a nonintrusive debugging system with fast download and is available separately from ARM.

The ARM710T header card provides Multi-ICE connector, PL5, which connects via a 20-way ribbon cable to the Multi-ICE hardware. The Multi-ICE server and debugger software should be run on the host computer. Please refer to the *Multi-ICE Installation Guide* for software installation instructions and the *Multi-ICE User Guide* for use of the Multi-ICE software.

The debugging system should be set up as shown in Figure 2-3.

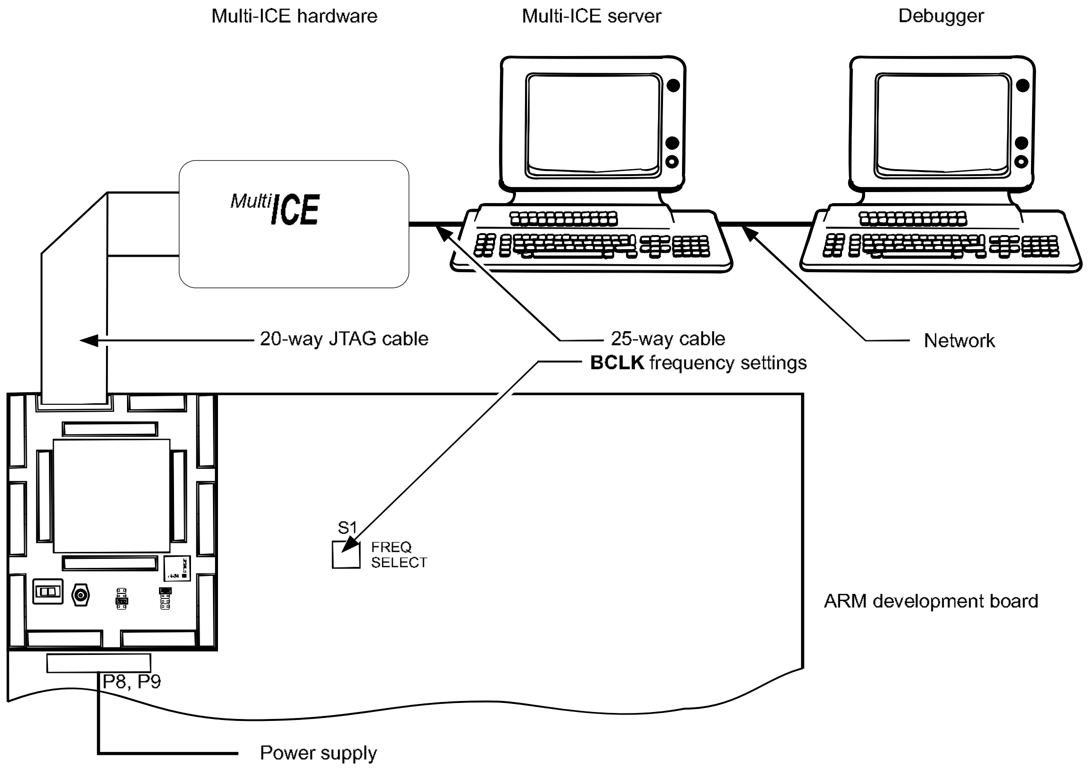

Figure 2-3 Multi-ICE debugging system

Note

Debugger and Multi-ICE server can be the same machine or two networked machines.

2.4 Debugging using EmbeddedICE

Debugging using EmbeddedICE is not recommended because EmbeddedICE does not support debugging with the caches on.

Chapter 3
Configuring the ARM710T Header Card

This chapter describes the board link and switch settings that configure the ARM710T header card. It contains the following sections:

3.1 Setting the core clock frequency

On the ARM710T header card, a phase-locked loop performs the core clock generation. The frequency and source of the core clock is determined by the settings of links LK4 and LK5. The positions of LK4 and LK5 are shown in Figure 3-1.

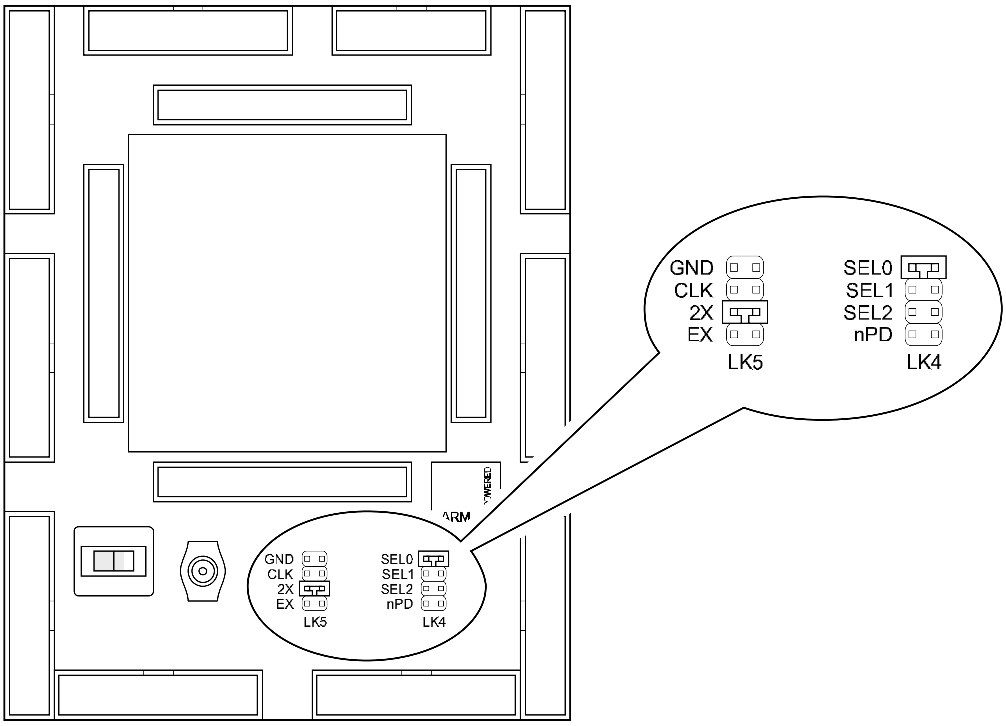

Figure 3-1 Position of links LK4 and LK5

The appropriate settings for the eight jumper links of LK4 and LK5 for a range of core clock frequencies are shown in Table 3-1 on page 3-3.

Table 3-1 Core clock frequency settings

Clock frequency (MHz)	LK5				LK4			
	GND	CLK	2X	EX	SEL0	SEL1	SEL2	nPD
4	OUT	IN	OUT	OUT	IN	IN	IN	OUT
8	OUT	OUT	IN	OUT	IN	IN	IN	OUT
16	OUT	OUT	IN	OUT	OUT	IN	IN	OUT
20	OUT	IN	OUT	OUT	OUT	OUT	IN	OUT
25	OUT	IN	OUT	OUT	IN	IN	OUT	OUT
32	OUT	OUT	IN	OUT	IN	OUT	IN	OUT
33	OUT	IN	OUT	OUT	OUT	IN	OUT	OUT
40	OUT	OUT	IN	OUT	OUT	OUT	IN	OUT
50	OUT	OUT	IN	OUT	IN	IN	OUT	OUT
66	OUT	OUT	IN	OUT	OUT	IN	OUT	OUT
80	OUT	OUT	IN	OUT	IN	OUT	OUT	OUT
100	OUT	OUT	IN	OUT	OUT	OUT	OUT	OUT
EXTCLK	OUT	OUT	OUT	IN	X	X	X	OUT
0 (stopped)	IN	OUT	OUT	OUT	X	X	X	IN

Operation is guaranteed only for those table values which generate frequencies up to 40MHz.

Note

Do not select any combination other than shown in the table.

3.2 Surface mount links

The ARM710T header card has three surface mount links, LK1, LK2, and LK3, which should not be moved for normal operation. Link LK1 sets the global bus enable, **GBE**, to HIGH (A-C position, default) or LOW (B-C position). Links LK2 and LK3 are used to control which grant and request signals (from the arbiter on the ARM development board) are used by the processor. Figure 3-2 shows the default position of links LK1, LK2 and LK3.

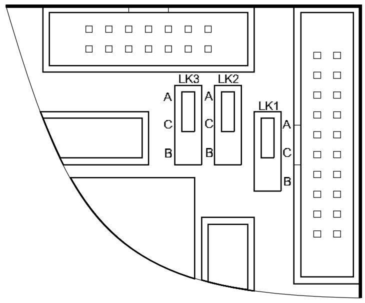

Figure 3-2 Default position of links LK1, LK2, and LK3

3.3 Using the external clock input

A 50-ohm mini-coax connector, SK5, is provided to allow an external clock source to drive core clock, **FCLK**, of the ARM710T. A 47-ohm resistor, R20, provides an approximate 50-ohm termination. The EXT link of LK5 should be fitted before an external clock source is connected to SK5. Figure 3-3 shows the position of SK5.

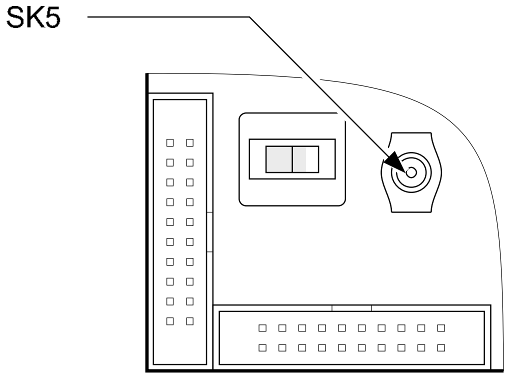

Figure 3-3 Position of SK5

3.4 Setting the bus clocking mode

Switch S1 is used to select between asynchronous mode (STD) and Fastbus mode (FB). The position of S1 is shown in Figure 1-1 on page 1-3.

Note

Do not change the setting of S1 while the processor is running.

Three bus clocking modes are supported by the processor. These are:

- Asynchronous mode - the bus clock frequency is separate to the core clock frequency.
- Synchronous mode - the bus clock frequency is separate to the core clock frequency, but the two are phase related. The core clock frequency is a multiple of the bus clock frequency.
- Fastbus mode - the core clock and the bus clock use the same clock. In this case no setting-up is required for the header clock frequency.

Note

Synchronous clocking mode is not supported on this header card.

For more details of clocking modes see the *ARM710T Datasheet*.

3.5 Setting the endianness

The ARM710T header card can be configured for either little-endian or big-endian operation. The factory setting is for little-endian operation.

3.5.1 Little-endian memory system

To configure the ARM710T header card for little-endian operation:

1. Remove BIGEND link (LK4) on the ARM development board.
2. Reset the ARM710T. It automatically assumes a little-endian memory system, so no software configuration is required.

3.5.2 Big-endian memory system

To configure the ARM710T header card for big-endian operation:

1. Fit BIGEND link (LK4) on the ARM development board.
2. Reset the ARM710T. It automatically assumes a little-endian memory system.
3. Change the ARM710T to BIGEND mode by setting bit 7 of coprocessor 15 register 1, see Example 3-1.

Note

Any byte accesses before this bit is set will be little-endian so will not access the expected data.

Example 3-1 Setting the ARM710T into BIGEND mode

```
MRC p15,0,r0,c1,0 ; read coprocessor 15 register 1
ORR r0,r0,#0x80   ; set bit 7
MCR p15,0,r0,c1,0 ; write coprocessor 15 register 1
```


Chapter 4
Circuit Descriptions

This chapter describes the operation of the ARM710T header card. It contains the following sections:

4.1 The header card circuit board

The header card is a 4-layer board and consists of:

- the ARM710T processor
- a *phase-locked loop* (PLL) clock generation chip
- external clock input
- logic analyzer, Multi-ICE, and EmbeddedICE connectors.

4.1.1 Processor in PGA

The ARM710T processor is supplied in a 299-pin *pin grid array* (PGA) package.

A number of inputs are tied to default values through resistors. These are listed in Table 4-1.

Table 4-1 Values of tied inputs

ARM710T signal names	Value
nFIQ	HIGH
nIRQ	HIGH
CPA	HIGH
CPB	HIGH
DSEL	LOW
SDOUTBS	LOW
BREAKPT	LOW
DBGRQ	LOW
EXTERN[1:0]	LOW
CPDBE	LOW
DBGEN	HIGH
nTRST	HIGH
TCK	HIGH
TMS	HIGH
TDI	HIGH

4.1.2 Clock generation

A PLL chip (U1) generates the high-speed clock, **FCLK**, which is used to clock the core of ARM710T. The PLL chip has two programmable outputs, **CLK** and **2X** which is double the frequency of **CLK**.

The links, LK4 and LK5, can be used to set the frequency of the core clock, **FCLK**, as shown in Table 2-1 on page 2-3.

The four links, GND, CLK, 2X, and EX, of LK5 allow the selection of clock source. Only one link of LK5 should be fitted at any one time.

4.1.3 External clock input

A 50-ohm mini-coaxial connector socket is fitted on the board to allow a clock input to be supplied from an external clock generator. The header card clock input is terminated in a 47-ohm resistor, R25, on the board. The EX link of LK5 should be fitted before applying an external clock source. R25 can be changed, if required, to allow proper termination of a clock frequency source of other than 50-ohms output impedance.

4.2 Connectors

The ARM710T header card has four 60-way sockets (SK1-4) mounted on the underneath. These correspond to the four 60-way plug connectors (PL1-PL4) on the development card, which allow the header card to be fitted to the development board.

4.2.1 Logic analyzer connectors

Six 20-way box headers, POD1-6, are provided to allow connection of Hewlett Packard 20-pin (HP 01650-63203) pods suitable for use with an HP1650B-series logic analyzer and thus trace the ARM710T activity. These connectors can also be used for expansion purposes and give access to coprocessor bus, CPD[31:0]. The pinout of connectors POD1 to 6 is given in Figure 4-1.

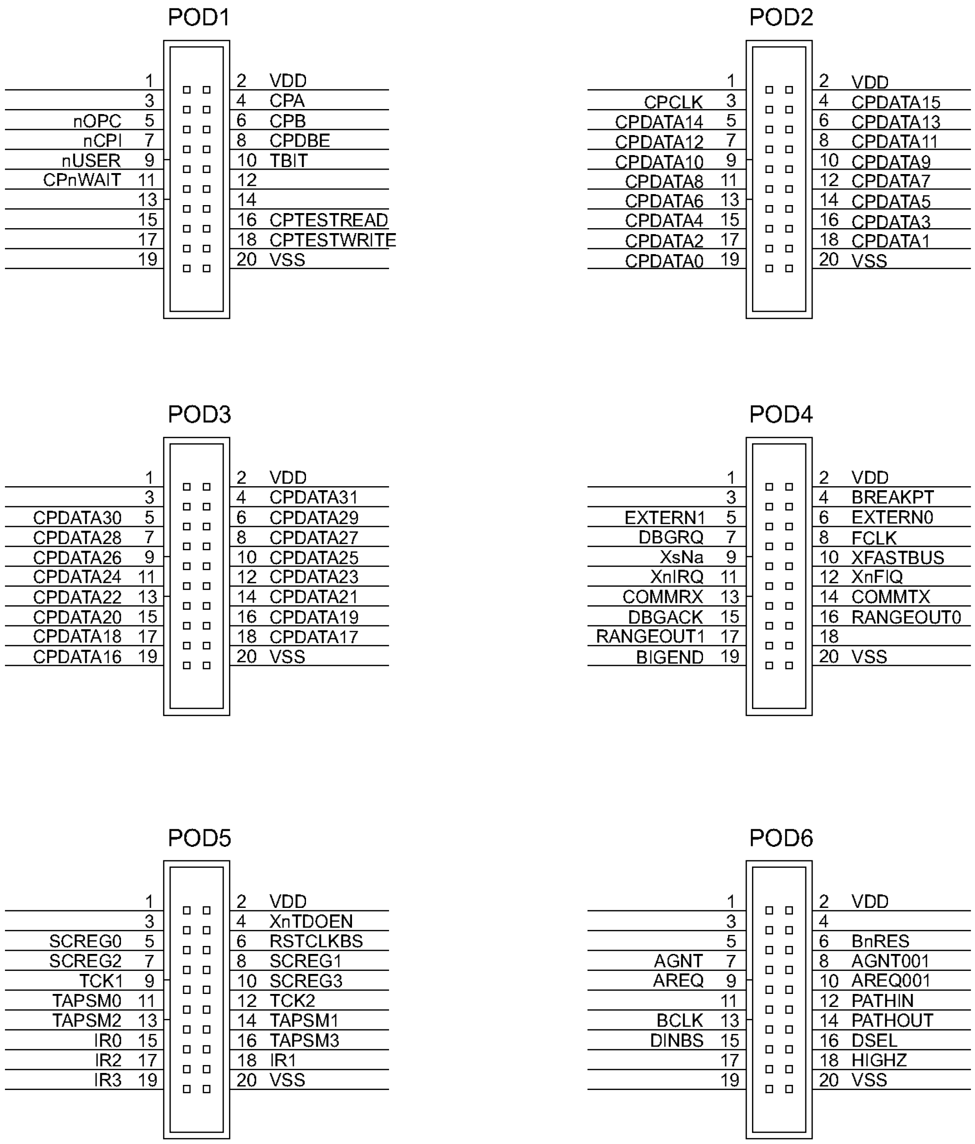

Figure 4-1 Boxed header pinouts

4.2.2 Multi-ICE connector

A 20-way connector, PL5, situated at the top of the header card, allows debugging of ARM710T using Multi-ICE. The pinout of PL5 is shown in Figure 4-2.

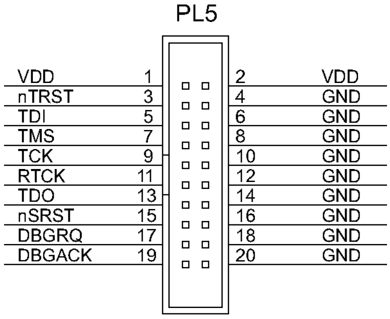

Figure 4-2 Pinout of Multi-ICE connector PL5

The Multi-ICE connector has the standard five JTAG signals, **nTRST**, **TDI**, **TMS**, **TCK** and **TDO**, which are interspersed with ground pins to reduce noise.

The signal **RTCK**, returned **TCK**, is connected to **TCK** on the header card, to allow Multi-ICE to make use of the *adaptive clock timing* option.

The signal **nSRST** can be used by the Multi-ICE unit to reset the ARM710T and ARM development board.

The **DBGRQ** and **DBGACK** signals are not used by Multi-ICE.

4.2.3 EmbeddedICE connector

A 14-way connector, PL6, situated at the top of the header card, allows debugging of ARM710T using EmbeddedICE. However, this method of debugging the ARM710T is not recommended, as EmbeddedICE does not support cache processors, so will not operate correctly if the cache of ARM710T is turned on.

For details of the pinout of PL6, see the schematic diagrams supplied in pdf format on the accompanying disk.

Index

The items in this index are listed in alphabetic order. The references given are to page numbers.

P

S