Designing LPC2000 Targets to Use with Development Tools

By Dr. Kurt Boehringer, Rev. 03-2004-06-23

The LPC2000 ARM7 microcontroller family has the advantage of highly sophisticated integrated debugging features. In conjunction with good development tools, the efficient development and easy test and debugging of quality applications is possible.

The internal debugging features are available in two levels:

Level 1: Run control features via the JTAG pins

Level 2: Additional trace information about the executed code via trace port pins

To make use of these on-chip features, some pins are used for development purposes and so cannot be used as I/O or function pins. To make the right decision as to which features are needed to support the development tools before designing the target, the possibilities and advantages of the two debugging levels are described in this paper. It is also shown how to design the connectors for the development tools.

Level 1

The LPC2000 has the ARM Embedded ICE Macro Cell implemented which can be accessed by an external tool via the JTAG pins (originally designed for boundary scan). This Embedded ICE Macro Cell includes two complex hardware breakpoints for detection of execution and data access events.

The debug facilities incorporated are the download of applications, the definition of execution breakpoints and data access watch points, the start and stop of the application and the examination and modification of registers or data.

The standardized interface required to connect an external development tool is shown in Table 1, with the pin assignment of an alternate 14-pin connector is shown in Table 2. Both connectors are 2-row strips with a 2.54 mm pin pitch.

Pin	Signal	Pin	Signal
1	VCC 3.3V	**2**	VCC 3.3V
3	nTRST	**4**	Ground
5	TDI (to LPC2100)	**6**	Ground
7	TMS	**8**	Ground
9	TCK	**10**	Ground
11	RTCK	**12**	Ground
13	TDO (from LPC2100)	**14**	Ground
15	nRST	**16**	Ground
17	no connection	**18**	Ground
19	no connection	**20**	Ground

Table 1 Embedded ICE debug connector 20 pin

Pin	Signal	Pin	Signal
1	VCC 3.3V	**2**	Ground
3	nTRST	**4**	Ground
5	TDI (to LPC2100)	**6**	Ground
7	TMS	**8**	Ground
9	TCK	**10**	Ground
11	TDO (from LPC2100)	**12**	NRST
13	VCC 3.3V	**14**	Ground

Table 2 Embedded ICE debug connector 14 pin

Due to the low JTAG pin count, the communication with the Embedded ICE Macro Cell has to be done serially. The performance of this communication and thus the efficiency of the development depends on the speed of the external development tool used. To make development more convenient, the elementary commands in the Embedded ICE Macro Cell have to be expanded with a powerful debugger and user interface.

With the right development tool - even with level 1 - the most important development activities can be performed:

1. Downloading the application in RAM or FLASH
2. Executing the application totally or partly by high level language stepping
3. Stopping the application with execution breakpoints or data watch points

© Hitex Development Tools

4. Examining and changing variables, registers and memory locations
5. Examining and modifying SFRs in "plain text"
6. Automated unit test in the target and system test with test protocol
7. FLASH programming of the final system

To design-in the Embedded ICE debug connector , some considerations have to be made. The LPC2104, -05 and -06 have two ways of using the JTAG pins. In primary mode, the RTCK, TRST, TCK, TDI, TDO and TMS pins are used and replace the P0.26, P0.27, P0.28, P0.29, P0.30 and P0.31 I/O port pins. These I/O pins are lost and cannot be used by the application. To enter the primary mode, the RTCK pin must be high when nRST is released. If open, RTCK is pulled up by an internal resistor or can be driven high by the development tool.

It is also possible to use the second set of JTAG pins. They replace the P0.17 to P0.21 port pins, which can also have the alternate functions CAPT1.2, CAPT1.3, MAT1.2 and MAT1.3 respectively. Using this set of JTAG pins these I/O or alternate functions cannot be used by the application. This set of JTAG pins has to be activated by the software programming of the registers in the Pin Connect Block.

The LPC2114, -19, -24 and -29 include the primary JTAG pins only.

Level 2

The level 2 advantage of the LPC2000 controllers is the implemented Embedded Trace Macro Cell (ETM). Besides the normal development test and debugging features of level 1, a reconstruction of executed code of the application can be performed (i.e. trace recording). For this purpose, additional 10 microcontroller pins are needed. The connector's pin assignment for a development tool supporting trace is shown in Table 3. The connector is a 38-pin, high-speed Amp Mictor connector. To activate the trace port, the TRACESYNC pin has to be pulled up during release of RST.

Pin	Signal	Pin	Signal
1	no connection	2	no connection
3	no connection	4	no connection
5	Ground	6	TRACECLK
7	no connection	8	no connection
9	nRST	10	EXTTRG
11	TDO (from LPC2100)	12	VCC 3.3V
13	RTCK	14	VCC 3.3V
15	TCK	16	Ground
17	TMS	18	Ground
19	TDI (to LPC2100)	20	Ground

Pin	Signal	Pin	Signal
21	nTRST	22	Ground
23	Ground	24	TRACEPKT[3]
25	Ground	26	TRACEPKT[2]
27	Ground	28	TRACEPKT[1]
29	Ground	30	TRACEPKT[0]
31	Ground	32	TRACESYNC
33	Ground	34	PIPESTAT[2]
35	Ground	36	PIPESTAT[1]
37	Ground	38	PIPESTAT[0]

Table 3 Pin assignment of the ETM Trace debug connector

The ETM is optimized to bring out of the microcontroller as much information as possible. Via the trace port, the ETM logic broadcasts the instruction flow information, including the addresses of executed branches and the contents of the execution pipeline. With this raw information, the development tool has the task of reconstructing instructions flow. Please note that self-modifying code cannot be traced in this way.

The trace recording can be controlled and selected by a trigger unit that includes address comparators as well as counters and sequencers.

A professional development tool with trace features has to perform the reconstruction of the trace as a complete raw trace, instruction trace or as a high level language line trace. A link from the current trace line to the corresponding source file is very helpful for examining the application's behaviour. If the development tool adds time stamps to each ETM message, the time measurement of the application's behaviour is also possible.

If you have any questions about the design of the LPC2000 debug connectors or selecting the right development tools, please don't hesitate to contact ARM@hitex.de.