Application Note 226

Using the Cortex-M0 on the Microcontroller Prototyping System

Document number: ARM DAI0226B
Issued: May 2010

ARM

Application Note 226
Using the Cortex-M0 on the Microcontroller Prototyping System

Release information

The following changes have been made to this Application Note:

Change history

Date	Issue	Change
August 2009	A	First release
May 2010	B	Second release

Version controlled by Domino.Doc DS158-GENC-009698 3.0

References

	Document	Issuer
[1]	User Manual for HMALC-AS3-52	Gleichmann Industries
[2]	HPE_Desk-Basic Online Help	Gleichmann Industries
[3]	AN227 Using the Microcontroller Prototyping System with the example reference design	ARM Ltd.
[4]	MPS QuickStart Guide	ARM Ltd.
[5]	CH7303 HDTV / DVI Transmitter (CH7303) Data Sheet	Chrontel
[6]	PrimeCell® Synchronous Serial Port (PL022) Technical Reference Manual	ARM Ltd.
[7]	PrimeCell® UART (PL011) Technical Reference Manual	ARM Ltd.

Proprietary notice

Confidentiality status

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

Feedback on this Application Note

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

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

General suggestions for additions and improvements are also welcome.

ARM web address

http://www.arm.com

Table of Contents

1 Introduction

1.1 Purpose of this application note

This Application Note covers the operation of the HMALC-AS3 Hpe®_module with the Hpe®_midiv2 FPGA development system from Gleichmann Electronics Research. It describes the contents of the CPU FPGA on the HMALC-AS3, including the clock structure and peripherals local to the CPU.

After reading this Application Note the user should be able to use the CPU FPGA with the example reference design described in [3] or with their own DUT FPGA design.

1.2 Overview of the hardware platform

This application note is designed to work on the Microcontroller Prototyping System (as shown in Figure 1) fitted with the ARM Hpe®_module (as shown in Figure 2).

For further details on this system please see [1].

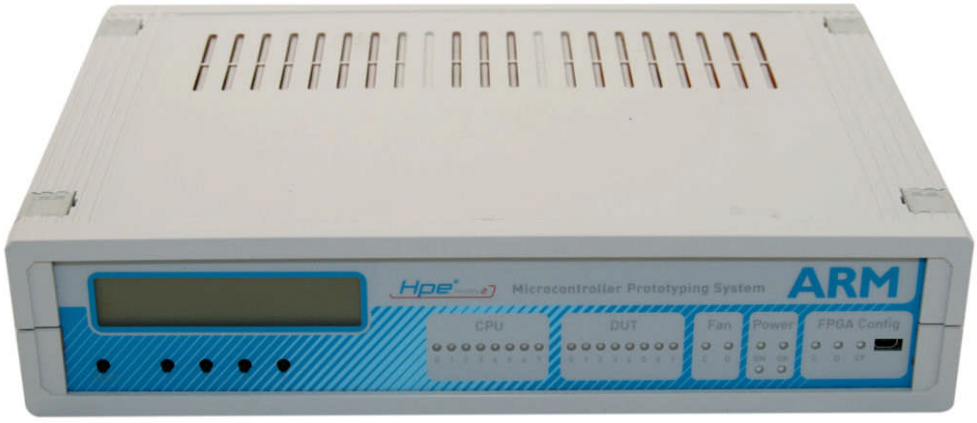

Figure 1: Microcontroller Prototyping System

Figure 2: ARM Hpe®_module HMALC-AS3

1.3 Getting started

The system comes pre-configured with an example design installed on the customer FPGA, described in [3]. The CPU FPGA is pre configured with the ARM Cortex-M3 processor, and BootMonitor software is loaded into the system flash memory.

Refer to the MPS QuickStart Guide [4] for details of setting up and using the MPS, including how to download an alternative ARM processor image to the CPU FPGA.

2 About the Processor Implementation

The FPGA images supplied with this Application Note contains an implementation of the ARM Cortex-M0 r0p0 processor, plus peripherals and bus infrastructure which are described in section 2.2. The ARM Cortex-M0 processor implements the ARMv6-M architecture.

The two FPGA images are identical except for the implemented debug interface:

- `fpga_processor_cortex-m0_serial_wire_encrypted.pof` : Serial Wire
- `fpga_processor_cortex-m0_jtag_encrypted.pof` : JTAG

2.1 Fixed Configuration

The ARM Cortex-M0 r0p0 processor includes a number of configuration options that may be set when the device is synthesized. Table 1 lists the options chosen for the FPGA implementation that accompanies this Application Note. Table 2 lists the system configuration choices that are relevant to this FPGA image.

The “Configuration Name” is the Verilog parameter name used to configure the processor and is included for reference for processor licencees.

Core Configuration Option	Configuration Name	Value	Comments
Architectural Clock Gating	ACG	0	Architectural Clock Gates **not** present.
Big Endian	BE	0	Processor is Little Endian (**LE**) only.
Number of Breakpoints	BKPT	4	4 Breakpoint Units.
Debug	DBG	1	Debug features are included.
Debugger Interface	JTAGnSW	0 1	Serial Wire debug interface JTAG debug interface Note: Two FPGA images are supplied to support the two options.
Number of Interrupts	NUMIRQ	32	32 external IRQs.
Register Reset	RAR	0	Only essential registers are reset.
System Timer	SYST	1	SysTick timer is included.
Small Multiplier	SMUL	0	Fast single-cycle multiplier is implemented.
Wake Up Interrupt Controller	WIC	0	The WIC is **not** implemented in the FPGA
Number of Watchpoints	WPT	2	2 Watchpoint comparators.

Table 1: Cortex-M0 Processor Configuration

System Configuration Option	Comments
Power Management Unit	Sleep modes will not offer any power saving because there is no PMU implemented in the FPGA.
System Timer Reference Clock	The SysTick timer is provided with a 100kHz reference clock. The appropriate 10ms calibration value is also provided.
Multi Processor Communications	The processor TXEV and RXEV pins are exported to the DUT.

Table 2: Cortex-M0 System Configuration

2.2 Software Programmable Configuration

Some aspects of the processor and system may be dynamically configured by software using the CPU FPGA System Registers (see section 5.4). Table 3 shows the software configurable features in this FPGA image.

Configuration Register	Comments
SYS_CPUCFG	The Cortex-M0 IRQLATENCY[7:0] configuration pins are driven by SYS_CPUCFG bits [7:0]. Other bits in the SYS_CPUCFG register are *Reserved*. See section 4.4 for details about IRQLATENCY.
SYS_BASE	The Cortex-M0 DAP baseaddr[31:0] configuration pins are driven by the SYS_BASE register.

Table 3: Software Programmable Configuration

3 MPS Architecture

The Gleichmann Microcontroller Prototyping System includes two FPGAs on which an AHB-Lite (AMBA 2.0) system is implemented:

- CPU FPGA
- DUT FPGA

Figure 3 shows a high-level block view of the MPS. This Application Note describes the CPU FPGA. Refer to [3] for details of the DUT FPGA, including how to customize and rebuild the DUT FPGA.

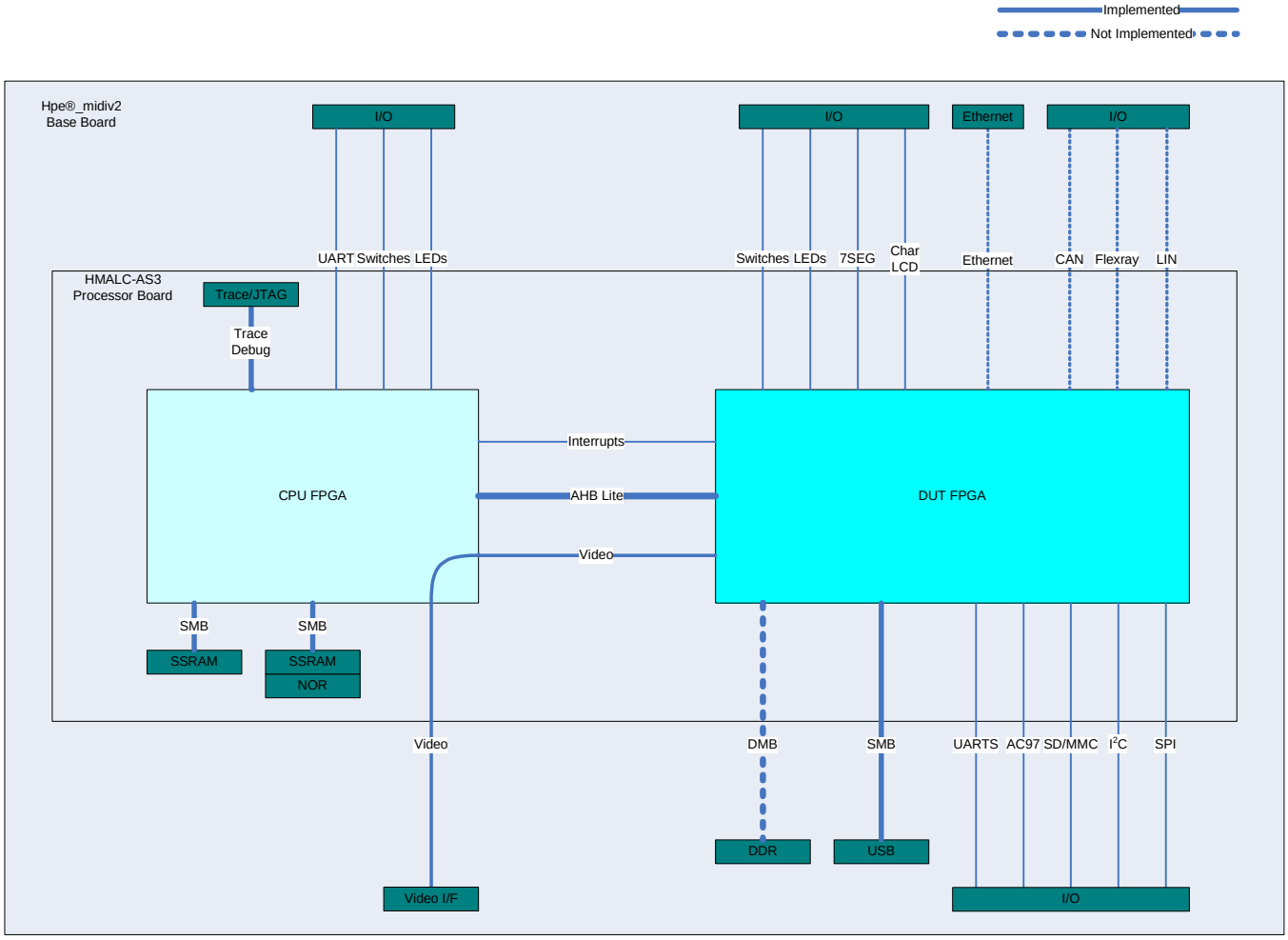

Figure 3: Block diagram of the Microcontroller Prototyping System

4 CPU FPGA Architecture

The CPU FPGA contains:

- ARM Cortex-M0 processor with
  - Serial Wire *OR* JTAG Debug Access Port (DAP)
- AHB Memory Controllers that interface to
  - 64k FPGA RAM (Internal “No Bus Latency” RAM)
  - 8MB SSRAM (Zero Bus Turnaround SSRAM)
  - 64MB NOR FLASH
- AHB Master Interface to the DUT FPGA
- AHB to APB bridge
  - APB Registers
    - Configuration of local components
    - Interfaces to LEDs and Switches
  - APB PrimeCell Components
    - PL011 UART (UART 3)
    - PL022 SSP (interface to TouchScreen controller)
  - APB Components
    - DS702 I^2C (interface to DVI Transmitter)

4.1 Bus Architecture

The CPU FPGA implements an AHB bus infrastructure to give the processor access to the local FLASH and SSRAM memory, and to the Customer DUT FPGA. An APB bus is used to connect local PrimeCell APB peripherals. Figure 4 shows the full AHB and APB system.

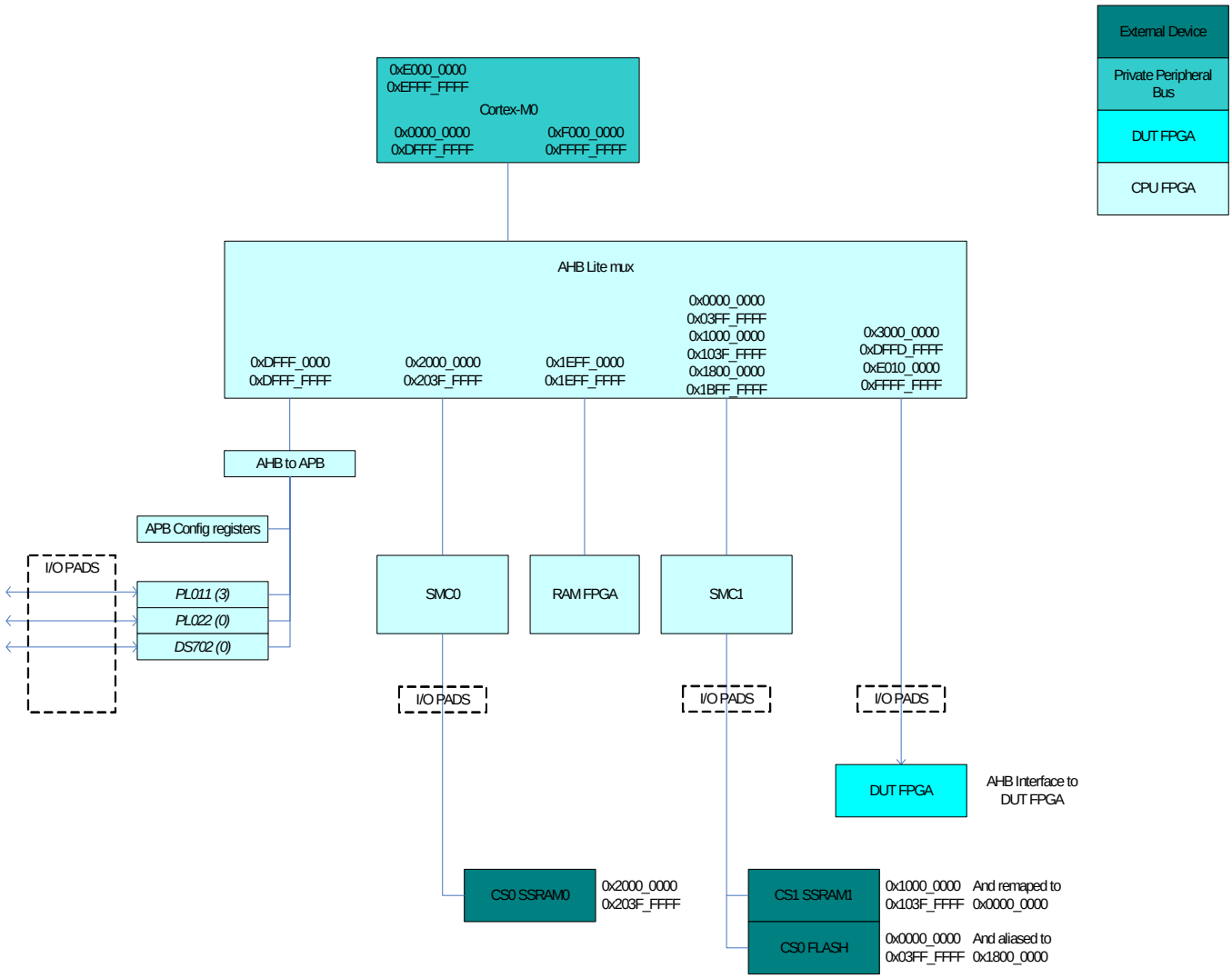

Figure 4: CPU FPGA Bus Architecture

4.2 Benchmarking Information

Table 4 describes the cycle performance of the CPU FPGA components shown in Figure 4. This information should be considered when benchmarking the performance of software running on the MPS. Please refer to section 4.4 if you are also benchmarking interrupt performance.

Component	Wait States	Note
AHB-Lite Mux	0	Implemented as combinatorial logic, and therefore do not introduce any cycle delays even when switching between segments.
SSRAM0	0	The two SSRAM blocks use ZBT RAMs to provide zero wait state access.
SSRAM1	0 (best case)	SSRAM1 shares physical pins with FLASH, so wait states may be incurred when accessing the FLASH and SSRAM1 in sequence. For benchmarking purposes, it is advisable to avoid using both the FLASH and SSRAM1 simultaneously. For best performance, program code from FLASH can be copied to SSRAM1 for execution.
FLASH	3 (default)	The FLASH memory interface inserts wait states according to the value programmed in SYS_WSCFG and the clock frequency selected by SYS_CLKCFG, see section 5.4 for further details.
RAM FPGA	0	The RAM FPGA block uses FPGA “No Bus Latency RAM” internally to provide zero wait state access.
AHB to APB Bridge	1	The AHB-APB bridge adds 1 wait state for accesses to the local peripherals and System Registers.

Table 4: AHB Component Cycle Performance

4.3 Clocks and Resets

The example reference system application note [3] details the clock and reset structure of the overall MPS system. Figure 5 shows the clocks used within the CPU FPGA.

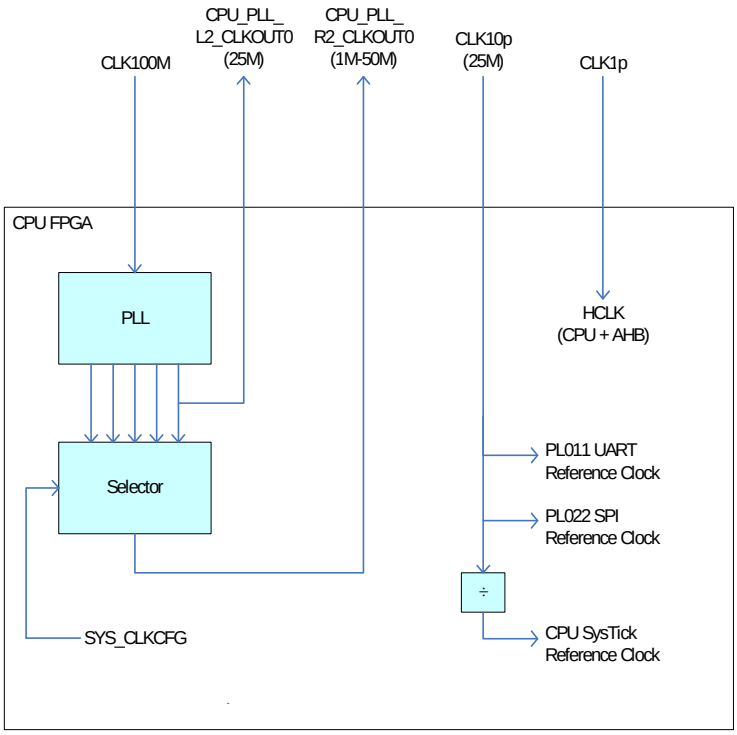

Figure 5: CPU FPGA Clocks

The CPU FPGA uses the following clocks:

- CLK100M: 100MHz reference from the oscillator (to drive the PLL)
- CLK1p: HCLK from the clock factory (AHB subsystem)
- CLK10p: 25MHz reference from the clock factory (PrimeCell reference clock for SPI and UART, and divided down to generate 100kHz SysTick external clock reference)

The CPU FPGA generates the following clocks:

- CPU_PLL_R2_CLKOUT0: for use as HCLK when re-distributed by the Clock Factory
- CPU_PLL_L2_CLKOUT0: 25MHz clock to Clock Factory

The CPU FPGA uses one internal PLL to generate a range of fixed clock frequencies from the 100MHz reference clock. A software controllable block allows code running on the processor, or a debugger, to change the HCLK frequency of the system by switching between these PLL generated clocks. See section 5.4.7 for details of how to program the HCLK frequency.

For normal operation, the Clock Factory must be configured to route the CPU FPGA programmable clock back to the CPU and DUT FPGAs as HCLK on signal CLK1p.

If you need to operate the MPS at an HCLK frequency not supported by the clock switcher above, a suitable HCLK must be generated within the DUT FPGA and the Clock Factory configured to route it to CLK1p.

Since the flash wait state configuration register adjusts automatically according to the clock frequency selection, in such case (using DUT for system clock generation), you may still want to program the clock configuration register and the flash configuration register to adjust the flash memory wait state. At reset (power-on-reset as well as system reset), the flash memory defaults to 3 wait states per access. You must ensure that the selected wait state configuration is acceptable for the HCLK frequency you are driving from the DUT.

4.3.1 Resets

The CPU FPGA is reset by the USER_RESET# signal. The PLL lock status output is also factored in to ensure the FPGA does not leave reset before the PLL has stabilized.

The CPU FPGA drives the AHB HRESETn signal to the DUT FPGA to create a synchronous reset with respect to HCLK. The DUT FPGA can use this to resynchronise resets to all other local clock domains as required.

A Reset may be generated by:

- Pushing the Reset Button
- Writing 1 to the ARM Cortex-M architected reset request bit, AIRCR.SYSRESETREQ. This may be done by code executing on the processor or by an external debugger. Note that the reset is not guaranteed to take place immediately after the write.
- The DUT FPGA asserts signal WDOGRES to the CPU FPGA. This could be used to implement a programmable watchdog timer within the DUT. See section 6 for details of the signal connections between the CPU and DUT.

4.4 Interrupts

The ARM Cortex-M0 processor can guarantee deterministic instruction execution and jitter free interrupt behavior for Zero-Wait-State memory systems when the IRQLATENCY input is correctly configured. If you are developing a system that requires deterministic execution or jitter free interrupt behavior, use the information in this chapter to remove or minimize wait states.

The Cortex-M0 processor includes an input bus IRQLATENCY to control the *minimum* number of clock cycles the processor will take to service an interrupt request. On the MPS platform, the SYS_CPUCFG register allows the user to program the value of IRQLATENCY.

The optimal value for IRQLATENCY is system dependant. For zero jitter interrupt behavior in a zero wait state system, IRQLATENCY should be tied to the decimal value 13. For non zero wait state systems, this value should be increased by the worst case number of wait states. For example, when the maximum wait state is N, the IRQLATENCY should be set to 13+N to get zero jitter behavior.

If you do not require zero-jitter interrupt behavior, IRQLATENCY should be tied to 0, and the Cortex-M0 processor will service interrupts as quickly as possible. By default the IRQLATENCY value is set to 0.

Note: The ARM Cortex-M0 processor does not include clock-domain-crossing synchronizers on its interrupt inputs. In the MPS, interrupts that are generated outside of the processor clock domain are double-flop synchronized into the processor clock domain before being presented to the processor NVIC. This incurs a delay of two HCLK cycles before an interrupt is recognized. Within the CPU FPGA, this affects the UART and SPI interrupts. Refer to your DUT documentation to determine which other interrupt sources are also double-flop synchronized.

5 Programmer's model

This chapter describes the MPS Programmer's Model in terms of the CPU FPGA:

- Interrupt Architecture
- Memory Map

Parts of the memory map and interrupt allocation depend on the system in the DUT FPGA. Refer to [3] for details of the example DUT reference design.

5.1 Interrupt architecture

The ARM Cortex-M processor family include a Nested Vectored Interrupt Controller (NVIC) which is integrated into the processor.

Figure 6 shows the mapping of external interrupts to the NVIC. The top eight interrupts are reserved for use within the CPU FPGA as shown. The remaining interrupts and NMI are available to the DUT FPGA. The allocation of these interrupts is dependent on the DUT FPGA.

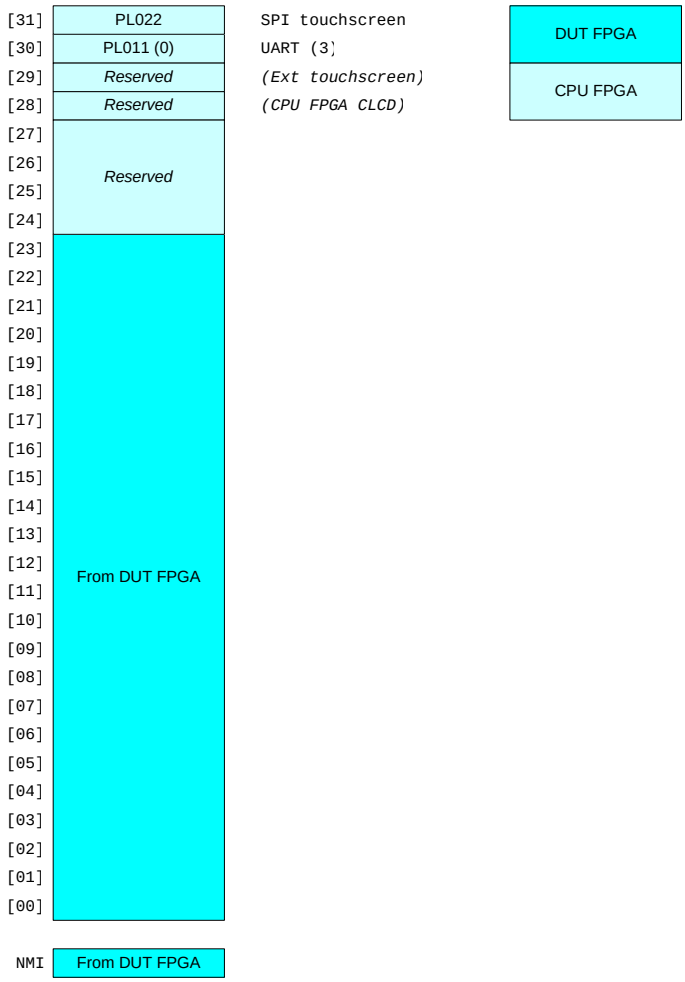

Figure 6: Interrupt Allocation Table

5.2 Memory map

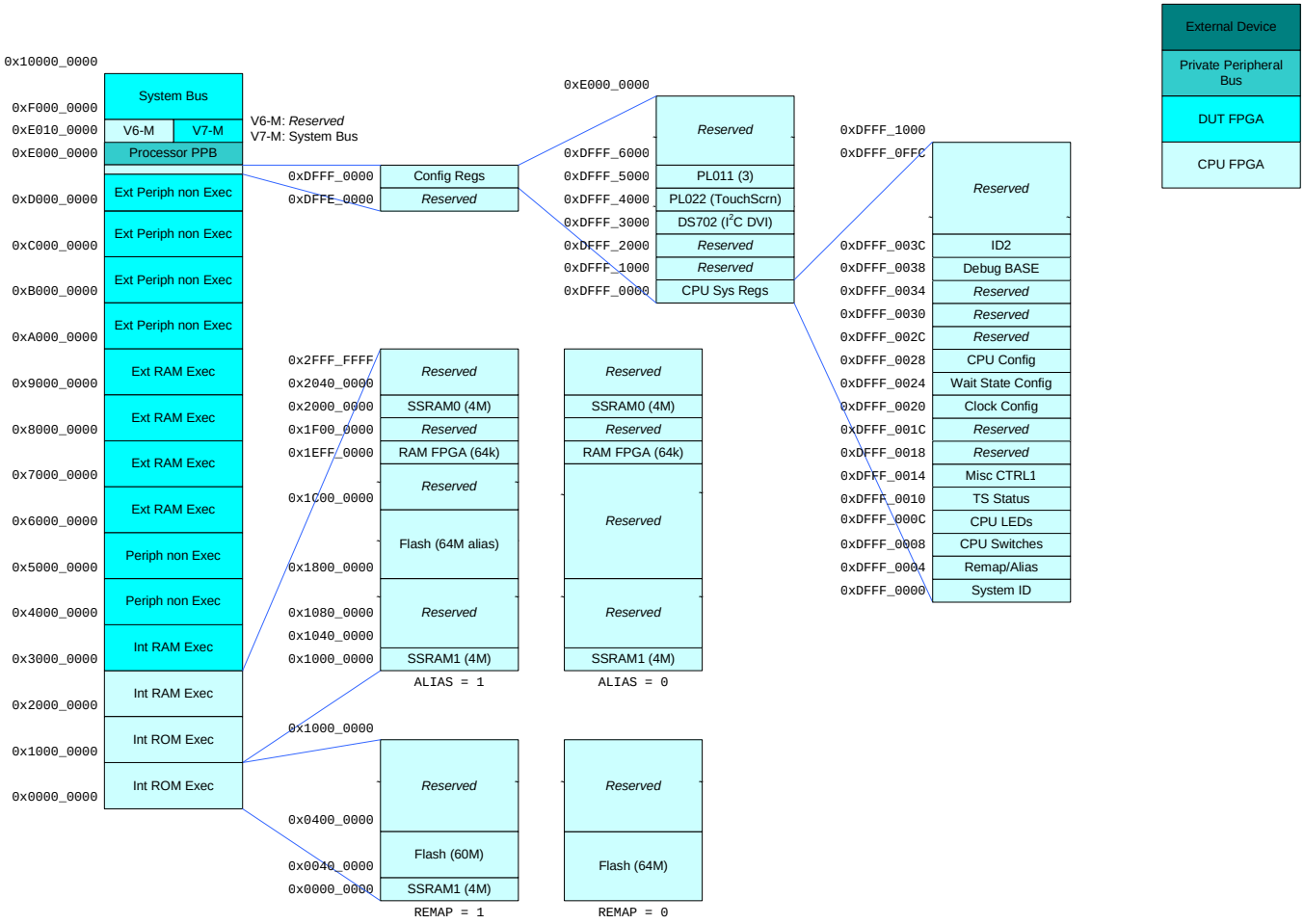

Figure 7: CPU FPGA memory map

Figure 7 shows the memory map of the CPU FPGA. SSRAM0 provides 4MB of RAM in the architected (ARMv6-M and ARMv7-M) SRAM memory region. SSRAM1 provides an additional 4MB of RAM within the architected CODE memory region. SSRAM1 can be used to provide zero wait state code access in fast systems, and to allow code development without the need to reprogram the Flash memory. The REMAP and ALIAS control bits are available to software via the SYS_MEMCFG register.

5.3 CPU FPGA Peripherals

The CPU FPGA includes the local peripherals listed in Table 5. Details of the PrimeCell UART (PL011) and SSP (PL022) can be found in [7] and [6]. Details of the DS072 and the CPU System Registers can be found in section 5.4.

Address	Peripheral	Usage
0xDFFF6000 – 0xDFFFFFFF	*Reserved*	
0xDFFF5000 – 0xDFFF5FFF	PL011	RS232 interface UART3, used by BootMonitor
0xDFFF4000 – 0xDFFF4FFF	PL022	SSP interface to touchscreen controller on Hpe_midiv2
0xDFFF3000 – 0xDFFF3FFF	DS072	I^2C interface to DVI Transmitter on Hpe_midiv2
0xDFFF1000 – 0xDFFF2FFF	*Reserved*	
0xDFFF0000 – 0xDFFF0FFF	CPU System Registers	LEDs, Switches and local configuration controls

Table 5: CPU FPGA Peripherals

5.4 CPU FPGA System Registers

The CPU System Registers are based at address 0xDFFF0000. Table 6 lists the registers. Full descriptions can be found in the following sections.

Address	Register	Description
0xDFFF0000	SYS_ID	ID Registers
0xDFFF0004	SYS_MEMCFG	Memory Configuration (including REMAP and ALIAS controls)
0xDFFF0008	SYS_SW	CPU DIP Switches
0xDFFF000C	SYS_LED	CPU LEDs
0xDFFF0010	SYS_TS	TouchScreen Status
0xDFFF0014	SYS_CTRL1	Miscellaneous Configuration
0xDFFF0018	*Reserved*	
0xDFFF001C	*Reserved*	
0xDFFF0020	SYS_CLKCFG	Clock Configuration
0xDFFF0024	SYS_WSCFG	Wait State Configuration
0xDFFF0028	SYS_CPUCFG	CPU Configuration
0xDFFF002C	*Reserved*	
0xDFFF0030	*Reserved*	
0xDFFF0034	*Reserved*	
0xDFFF0038	SYS_BASE	Debug Access Port CoreSight Component Pointer Address
0xDFFF003C	SYS_ID2	Secondary Identification Register
0xDFFF0040 – 0xDFFF0FFC	*Reserved*	

Table 6: CPU FPGA System Registers

5.4.1 SYS_ID (0xDFFF0000)

The System Identification register returns a value specific to the CPU FPGA image.

Name	Bits	Access	Reset	Note
REV	31:28	RO	'h1	Board Revision B
BOARD	27:16	RO	'h023	HBI Board number
VARIANT	15:12	RO	'h0	Build Variant of board
ARCH	11:8	RO	'h4	Bus Architecture (4 AHB, 5 AXI)
BUILD	7:0	RO	'hxx	FPGA build

5.4.2 SYS_MEMCFG (0xDFFF0004)

The Memory Configuration register is reset only by power-on-reset – soft resets such as debug or system reset do not alter the state of this register. The default value allows the MPS to boot from Flash memory. Refer to section 5.1 for details of the function of the REMAP and ALIAS bits.

Name	Bits	Access	Power On Reset	Note
Reserved	31:2			
ALIAS	1	RW	'b1	Alias FLASH. 1 is Aliased on 0 Aliased off
REMAP	0	RW	'b0	Remap SSRAM. 1 is Remap on 0 Remap off

5.4.3 SYS_SW (0xDFFF0008)

The Switch register returns the value of the eight switches (arranged as two groups of four) labeled "P1" and "P2" on the HMALC-AS3 board [1].

Name	Bits	Access	Reset	Note
Reserved	31:8			
USER_SWITCH	7:0	RO	'h--	Value depends on switch settings

5.4.4 SYS_LED (0xDFFF000C)

The LED register controls the eight processor LEDs on the HMALC-AS3 board [1], unless overridden by SYS_CTRL1.

Name	Bits	Access	Reset	Note
Reserved	31:8			
LED	7:0	RW	'h00	Write 1 to light the corresponding LED. Reads return the last value written to the register.

5.4.5 SYS_TS (0xDFFF0010)

The TouchScreen status register shows the busy and interrupt status from a touchscreen device on the Hpe_midiv2 baseboard.

Name	Bits	Access	Reset	Note
Reserved	31:2			
TS_INT	1	RO	'b-	External Interrupt from Touchscreen
TS_BUSY	0	RO	'b-	External Busy signal from Touchscreen

5.4.6 SYS_CTRL1 (0xDFFF0014)

The Miscellaneous control register allows the function of the eight processor LEDs that are normally controlled by the SYS_LED register to be overridden with the status of various processor outputs. The output signals are pulse-stretched to ensure that single-cycle pulses are visible as flashes on the LEDs.

Name	Bits	Access	Reset	Note
Reserved	31:1			
LED_FUNC	0	RW	'b0	Write 0 to drive LEDs from SYS_LED register. Write 1 to drive LEDs from CPU status signals: 7: SLEEPING 6: SLEEPDEEP 5: HALTED 4: LOCKUP 3: SYSRESETREQ 2: TXEV 1: WAKEUP 0: DBGRESTARTED

5.4.7 SYS_CLKCFG (0xDFFF0020)

The Clock Configuration register allows the clock speed of the processor and its AHB subsystem to be modified easily by software for benchmarking purposes. Attempting to write a reserved value to this register will result in a valid clock value being selected. Note that when 12MHz setting is used, the duty cycle of the output clock can either be 40% or 60%.

Name	Bits	Access	Reset	Note
Reserved	31:4			
CLOCKCFG	3:0	RW	'hD	0xF, 0xE – Reserved 0xD – 50MHz 0xC – 40MHz 0xB – 30MHz 0xA – 25MHz 0x9 – 20MHz 0x8 – 15MHz 0x7 – 12MHz 0x6 – 10MHz 0x5 – 8MHz 0x4 – 6MHz 0x3 – 4MHz 0x2 – 2MHz 0x1 – 1MHz 0x0 – Reserved

5.4.8 SYS_WSCFG (0xDFFF0024)

The Wait State configuration register controls the number of wait-states inserted by the memory controller when accessing Flash memory. The default value of three wait states is required for the default operating frequency of 50MHz.

The register contains hardware logic to determine if the write data is valid for the current operating frequency and automatically forces the write data to a valid value if necessary.

If the clock frequency setting is updated to a higher frequency, WSCFG is automatically updated to a valid value if necessary.

Table 7 shows the valid wait state configurations when running at the different clock frequencies supported by SYS_CLKCFG.

Name	Bits	Access	Reset	Note
Reserved	31:2			
WSCFG	1:0	RW	'h3	0x3 – 3 wait states on read, 3 wait states on write 0x2 – 2 wait states on read, 2 wait states on write 0x1 – 1 wait state on read, 1 wait state on write 0x0 – 0 wait state on read, 1 wait state on write

Frequency (MHz)	3 Wait States	2 Wait States	1 Wait State	0 Wait State
40, 50	Yes	-	-	-
30	Yes	Yes	-	-
15, 20, 25	Yes	Yes	Yes	-
1, 2, 4, 6, 8, 10, 12	Yes	Yes	Yes	Yes

Table 7: Valid Wait State Configurations

5.4.9 SYS_CPUCFG (0xDFFF0028)

The CPU Configuration register is used to control various processor specific features. Refer to section 2 to determine if and how this register is used by the processor FPGA.

Name	Bits	Access	Reset	Note
	31:8	--	--	Processor Specific – Refer to section 2

5.4.10 SYS_BASE (0xDFFF0038)

The SYS_BASE register drives the value that an external (Serial Wire or JTAG) debugger sees when connecting to the Debug Access Port and reading its BASE register. Refer to section 2 to determine if this feature is supported by the processor FPGA.

BASEADDRESS is reset by power-on-reset to the ARMv6-M and ARMv7-M architected value for the processor ROM table. Soft resets such as debug or system reset do not alter the state of this register.

This register may be updated by software to point to a system level ROM table in the DUT FPGA, which in turn points to the architected processor ROM table. The system level ROM table(s) may provide identification of the user's customized processor system and any additional CoreSight compliant peripherals within it. This register allows the user to test debug tools connectivity with such customized systems.

Name	Bits	Access	Reset	Note
BASEADDRESS	31:0	*See Note*	'hE00FF003	This register will be RO if the processor FPGA does not support reprogramming of the DAP BASE register value.

5.4.11 SYS_ID2 (0xDFFF003C)

The SYS_ID2 register may contain additional CPU FPGA build information. The format of that information is not specified.

Name	Bits	Access	Reset	Note
ID_2	31:0	RO	'hx	Reserved for ARM internal use.

5.5 DS072 I^{2}C Interface to DVI Transmitter

The DS072 I^{2}C peripheral is based at address 0xDFFF3000 and is used to interface to the DVI transmitter [5] on the baseboard. The DS072 implements a simple register interface only – the I^{2}C protocol must be generated in software using a "bit-banging" technique. Table 8 lists the registers. Full descriptions can be found in the following sections.

Address	Register	Description
0xDFFF3000	SB_CONTROL	Status Register
0xDFFF3000	SB_CONTROLS	Output Set Register (Note – Same address as SB_CONTROL)
0xDFFF3004	SB_CONTROLC	Output Clear Register

Table 8: DS072 I^{2}C Registers

5.5.1 SB_CONTROL (0xDFFF3000)

The SB_CONTROL register returns the value of the serial data (SDA) and serial clock (SCL) pins when read.

Name	Bits	Access	Reset	Note
Reserved	31:2			
SB_SDA	1	RO	'b0	Level of SDA signal
SB_SCL	0	RO	'b0	Level of SCL signal

5.5.2 SB_CONTROLS (0xDFFF3000)

The SB_CONTROLS (SET) register allows the SDA and SCL pins to be pulled high by pullup resistors on the board by writing 'b1 to the corresponding bit.

Name	Bits	Access	Reset	Note
Reserved	31:2			
SB_nSDAOUTEN	1	W	'b0	Sets SDA line when 1
SB_SCLOUT	0	W	'b0	Sets SCL line when 1

5.5.3 SB_CONTROLC (0xDFFF3004)

The SB_CONTROLC (CLEAR) register allows the SDA and SCL pins to be driven low by writing 'b1 to the corresponding bit.

Name	Bits	Access	Reset	Note
Reserved	31:2			
SB_nSDAOUTEN	1	W	'b0	Clears SDA line when 1
SB_SCLOUT	0	W	'b0	Clears SCL line when 1

5.6 PL022 SPI Interface

The PL022 PrimeCell peripheral is based at address 0xDFFF4000 and is used to interface to the touchscreen on the baseboard.

5.7 PL011 UART Interface

The PL011 PrimeCell peripheral is based at address 0xDFFF5000 and provides the UART3 RS-232 serial interface. This interface may be used by the BootMonitor software, see [3] for more details.

6 Signal assignments

6.1 Resets

FPGA Signal	Direction [Width]	Note
USER_RESETn	input	User reset

6.2 Clocks from Clock Factory

FPGA Signal	Direction [Width]	Note
CPU_CLK1	input	AHB and Processor clock input, HCLK.
CPU_CLK5	input	Not used.
CPU_CLK10	input	Peripheral reference clock input, 25MHz.
CPU_CLK15	input	Not used.
CPU_CLK100M	input	100MHz reference clock.

6.3 Clocks to Clock Factory

FPGA Signal	Direction [Width]	Note
CPU_PLL_L2_CLKOUT0	output	Peripheral reference clock from FPGA PLL. 25MHz
CPU_PLL_R2_CLKOUT0	output	AHB and Processor clock from FPGA PLL. 1-50MHz
CPU_PLL_B1_CLKOUT3	output	Not used by clock factory. 60MHz.
CPU_PLL_T1_CLKOUT3	output	Not used by clock factory. 40MHz.

6.4 AHB-Lite Interface to DUT

The CPU FPGA implements a 32bit AHB-Lite master interface to the DUT FPGA.

FPGA Signal	Direction [Width]	Note
FPGA_IC[31:0]	output [31:0]	HWDATA
FPGA_IC[32]	output	HWRITE
FPGA_IC[35:33]	output	HBURST[2:0]
FPGA_IC[36]	output	HMASTLOCK
FPGA_IC[40:37]	output [3:0]	HPROT
FPGA_IC[43:41]	output [2:0]	HSIZE
FPGA_IC[45:44]	output [1:0]	HTRANS
FPGA_IC[46]	output	*Reserved for HSEL, tied to 1'b1*
FPGA_IC[78:47]	output [31:0]	HADDR
FPGA_IC[110:79]	input [31:0]	HRDATA
FPGA_IC[111]	input	HRESP
FPGA_IC[112]	input	HREADY
FPGA_IC[113]	output	HRESETn

6.5 Processor Miscellaneous Signals

The following CPU signals are exported to the DUT FPGA.

FPGA Signal	Direction [Width]	Note
FPGA_IC[125:114]	input [11:0]	INT[11:0]
L14_DUTOUT_DN[11:0]	input [23:12]	INT[23:12]
L14_DUTOUT_DN[12]	input	NMI
L14_CPUOUT_DN[0]	output	SLEEPING
L14_CPUOUT_DN[1]	output	SLEEPDEEP
L14_CPUOUT_DN[2]	output	HALTED

L14_CPUOUT_DN[3]	output	LOCKUP
L14_CPUOUT_DN[4]	output	TXEV
L14_CPUOUT_DN[5]	input	RXEV - Tie low if not used by DUT.
L14_CPUOUT_DN[6]	input	WDOGRES – Watchdog Reset Request, tie low if not used by DUT.
L14_CPUOUT_DN[12:7]	N/C	
L14_CPUOUT_DP[9:0]	N/C	
L14_CPUOUT_CLK	N/C	

6.6 Touch Screen Interface

The TouchScreen SPI interface is driven by PrimeCell PL022 [6]. Additional signals are visible via the TouchScreen Status Register, see section 5.4.5.

FPGA Signal	Direction [Width]	Note
TOUCH_SPI_BUSY	input	Connects to TS_BUSY
TOUCH_SPI_CS#	output	Driven by TS_FSSOUT
TOUCH_SPI_DCLK	output	Driven by TS_CLK
TOUCH_SPI_DIN	input	Connects to TS_DIN
TOUCH_SPI_DOUT	output	Driven by TS_DOUT
TOUCH_SPI_IRQ#	input	Connects to TS_INTn

6.7 LCD, VGA/DVI

The CPU FPGA de-multiplexes signals from the DUT FPGA and passes them on to the LCD connector and DVI transmitter. Refer to [3] for details of how to drive these interfaces from the DUT FPGA.

The DVI Transmitter [5] is controlled using an I^2C interface as described in section 5.5. Figure 8 shows the electrical connections for this interface.

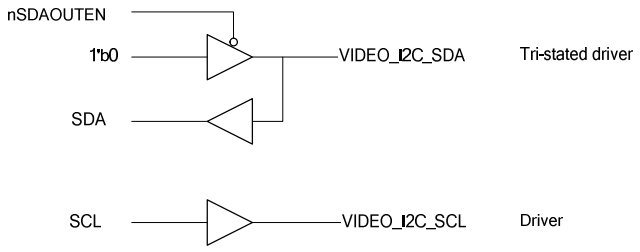

Figure 8: Video I^2C Connections

6.8 Human Interface (Switches and LEDs)

FPGA Signal	Direction [Width]	Note
CPU_DSW	input [7:0]	Read via System register SYS_SW.
CPU_LED	output [7:0]	Set via System Register SYS_LED.

6.9 JTAG and Trace

The Processor FPGA can support JTAG / Serial Wire debug interfaces and a Trace interface. Refer to section 2 to determine the options supported by the processor.

The JTAG/Serial Wire debug interface is available on two connectors; the rear-panel debug connector and the internal Mictor connector. The debug interface signals are routed to the rear-panel connector if a debugger cable is detected, otherwise they are routed to the Mictor connector.

6.9.1 Rear-Panel JTAG Connector (X14 - "Debug")

FPGA Signal	Direction [Width]	Note
FTSH_GNDDET	bi-dir	Connector Detect. Weak pull-up on FPGA which is pulled low by the connector to indicate a connection.
FTSH_TMS	bi-dir	Input to SWDIOTMS on processor. Also used as Data Out for Serial Wire Debug.
FTSH_TCK	bi-dir	JTAG Clock to processor.
FTSH_TDO	bi-dir	JTAG Data Out from processor.
FTSH_TDI	bi-dir	JTAG Data In to processor.
FTSH_TRST	bi-dir	JTAG Reset. This is an active low signal.

6.9.2 Mictor Connector (X14)

The Mictor connector allows a Trace probe to be connected to processors that support ETM.

FPGA Signal	Direction [Width]	Note
INTCPU_TDI	input	JTAG Data In to processor.
INTCPU_TDO	bi-dir	JTAG Data Out from processor.
INTCPU_TCK	input	JTAG Clock to processor.
INTCPU_TMS	input	Input to TMS on processor. This is not connected to the Serial Wire Debug Data Out.
INTCPU_TRSTn	input	JTAG TAP Reset. This is an active low signal.
INTCPU_SRSTn	input	Factored into CPU reset.
INTCPU_RTCK	output	Unused, Tied to '0'.
INTCPU_DBGRQ	input	Not connected.
INTCPU_DBGACK	output	Unused, Tied to '0'.
MICTOR_PIPESTAT0	output	TRACEDATA[0] from the processor.
MICTOR_PIPESTAT1	output	Tied to '0'.
MICTOR_PIPESTAT2	output	Tied to '1'.
MICTOR_EXTTRIG	output	Tied to '0'.
MICTOR_TRACEPKT	output[15:0]	{TRACEDATA[15:1], 0} from the processor.
MICTOR_TRACESYNC	output	Unused, Tied to '0'.
MICTOR_TRACECLK	output	Connects to processor TRACECLK port.

6.10 RS232 connection (X4)

The CPU FPGA includes one UART, implemented using PrimeCell PL011 [7].

FPGA Signal	Direction [Width]	Note
RS1_RXD_LVTTL	input	
RS1_TXD_LVTTL	output	