

Application Note **123**

Versatile/IT1 example design

Document number: ARM DAI 0123A

Issued: April 2004

Copyright ARM Limited 2004

ARM

Application Note 123

Versatile/IT1 example design

Copyright © 2004 ARM Limited. All rights reserved.

Release information

The following changes have been made to this Application Note.

Change history

| Date | Issue | Change |
|----------------|--------------|------------------------------|
| March 30, 2004 | A.01 | First release |
| March 31, 2004 | A.02 | Review comments added |
| June 22, 2004 | A.03 | Second review comments added |

Proprietary notice

ARM, the ARM Powered logo, Thumb and StrongARM are registered trademarks of ARM Limited.

The ARM logo, AMBA, Angel, ARMulator, EmbeddedICE, ModelGen, Multi-ICE, ARM7TDMI, ARM9TDMI, TDMI and STRONG are trademarks of ARM Limited.

All other products, or services, mentioned herein may be trademarks of their respective owners.

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 | 2 |
| 1.1 | Purpose of this application note | 2 |
| 1.2 | Versatile/IT1 overview | 2 |
| 1.3 | Versatile/IT1 example design | 2 |
| 1.4 | Versatile/IT1 considerations with the IDE interface and AHB S bus | 3 |
| 2 | Software Interface | 5 |
| 2.1 | ZBT SSRAM, APB and GPIO registers | 5 |
| 2.2 | Selftest software | 6 |
| 2.3 | Selftest cable harness | 10 |

1 Introduction

1.1 Purpose of this application note

This application note describes an example design implemented in a Versatile/LT-XC2V4000+ Logic Tile to test the Versatile/IT1 Interface Tile. The design is based on the Logic Tile example described in AN119 with the addition of several GPIO interfaces which are used to test the Versatile/IT1 peripherals. While the design is adequate for test purposes it may impose impractical performance overheads on a number of peripherals. A full system design would therefore require additional hardware blocks to realize the full performance of the Versatile/IT1 peripherals.

1.2 Versatile/IT1 overview

The Versatile/IT1 provides a number of peripheral interface drivers which can be driven from a design implemented in a Versatile/LT-XC2V4000+. By default all Versatile/IT1 header X/Y/Z connections are routed from the lower to the upper connectors. This allows the Versatile/IT1 to be placed in a stack above or below the Versatile/LT-XC2V4000+.

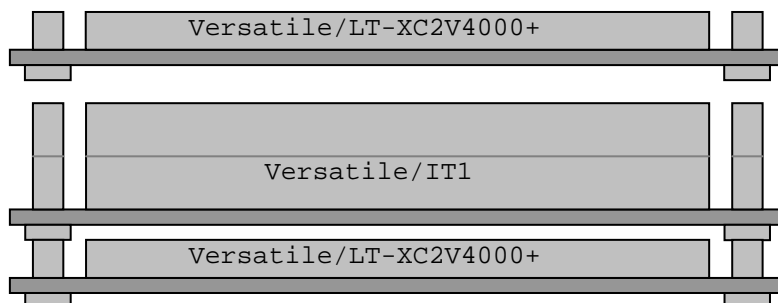


Figure 1-1 Versatile/IT1 board stack

All peripherals can be isolated from header signals using the Peripheral Enable switch S1. This allows a selection of peripherals to be used while freeing up unused header signals.

The Versatile/IT1 also provides a large prototyping area containing 272 user I/O pads and 7 clock related pads. This area may be populated with components or connectors on a 0.1" pitch spacing.

1.3 Versatile/IT1 example design

An example test design is provided for the Versatile/IT1 which connects all of the peripheral interfaces to general purpose I/O ports – GPIO. A total of five 32bit GPIO peripherals are used to implement all peripherals interfaces, although not all bits are used on each GPIO. Figure 1-2 shows the example design, Table 1-2 shows the pin allocation used.

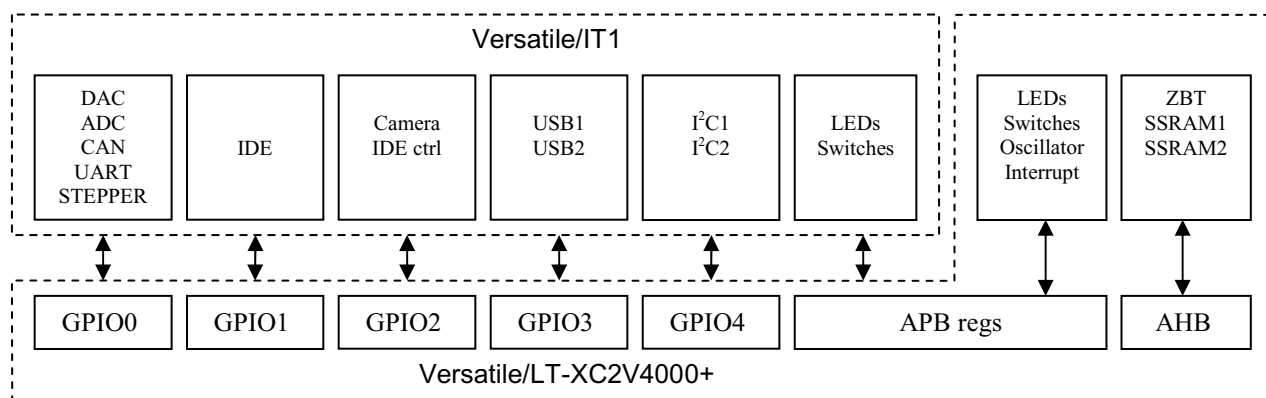


Figure 1-3 Versatile/IT1 example design block diagram

| Generic Tile name | Peripheral Signal | Direction | GPIO | Register Name |
|-------------------|-------------------|-----------|--------------|-----------------|
| Y15:0 | IDE_D[15:0] | I/O | GPIO1[15:0] | IT_IDEDMASK |
| Y16 | IDE_nIOW | O | GPIO1[16] | IT_IDENIOW |
| Y17 | IDE_nIOR | O | GPIO1[17] | IT_IDENIOR |
| Y18 | IDE_DMAACK | O | GPIO1[18] | IT_IDEDMAACK |
| Y[21:19] | IDE_DA[2:0] | O | GPIO1[21:19] | IT_IDEDA[2:0] |
| Y[23:22] | IDE_CS[1:0] | O | GPIO1[23:22] | IT_IDECS[1:0] |
| Y24 | IDE_RST | O | GPIO1[24] | IT_IDERST |
| Y25 | IDE_DMAR | I | GPIO1[25] | IT_IDEDMAR |
| Y26 | IDE_IORDY | I | GPIO1[26] | IT_IDEIORDY |
| Y27 | IDE_IRQR | I | GPIO1[27] | IT_IDEIRQR |
| Y28 | IDE_nIOCS16 | I | GPIO1[28] | IT_IDENIOCS16 |
| Y29 | IDE_PASSDIAG | I | GPIO1[29] | IT_IDEPASSDIAG |
| Y30 | IDE_nACTIVE | I | GPIO1[30] | IT_IDENACTIVE |
| Y[33:31] | IDE_BDIR[2:0] | O | GPIO2[18:16] | IT_IDEBDIR[2:0] |
| Y[36:34] | IDE_nBOE[2:0] | O | GPIO2[21:19] | IT_IDEBBOE[2:0] |
| Y[68:61] | SW[7:0] | I | System Reg | IT_SW[7:0] |
| Y[76:69] | LED[7:0] | O | System Reg | IT_LED[7:0] |
| Z[183:176] | CAM_D[7:0] | O | GPIO2[7:0] | IT_CAMDMASK |
| Z184 | CAM_SCL | O | GPIO2[8] | IT_CAMSCL |
| Z185 | CAM_SDA | O | GPIO2[9] | IT_CAMSDA |
| Z186 | CAM_HSYNC | I | GPIO2[10] | IT_CAMHSYNC |
| Z187 | CAM_VSYNC | I | GPIO2[11] | IT_CAMVSYNC |
| Z188 | CAM_CLK | O | GPIO2[12] | IT_CAMCLK |
| Z189 | CAM_STANDBY | O | GPIO2[13] | IT_CAMSTANDBY |
| Z194 | CAM_nRESET | O | GPIO2[14] | IT_CAMNRESET |
| Z[202:195] | I2C_D[7:0] | I/O | GPIO4[7:0] | IT_I2CDMASK |
| Z203 | I2C_nRST | O | GPIO4[8] | IT_I2CNRST |
| Z[205:204] | I2C_A[1:0] | O | GPIO4[10:9] | IT_I2CA0/1 |
| Z206 | I2C1_nCS | O | GPIO4[11] | IT_I2C1NCS |
| Z207 | I2C2_nCS | O | GPIO4[12] | IT_I2C2NCS |
| Z208 | I2C_nRD | O | GPIO4[13] | IT_I2CNRD |
| Z209 | I2C_nWR | O | GPIO4[14] | IT_I2CNWR |
| Z210 | I2C1_nINT | I | GPIO4[15] | IT_I2C1NINT |
| Z211 | I2C2_nINT | I | GPIO4[16] | IT_I2C2NINT |
| Z212 | USB1_MODE | O | GPIO3[0] | IT_USB1MODE |
| Z213 | USB1_nOE | O | GPIO3[1] | IT_USB1NOE |
| Z214 | USB1_SUSPEND | O | GPIO3[2] | IT_USB1SUSPEND |
| Z215 | USB1_VMO | O | GPIO3[3] | IT_USB1VMO |
| Z216 | USB1_VPO | O | GPIO3[4] | IT_USB1VPO |
| Z217 | USB1_SPEED | O | GPIO3[5] | IT_USB1SPEED |
| Z218 | USB1_RCV | I | GPIO3[6] | IT_USB1RCV |
| Z219 | USB1_VP | I | GPIO3[7] | IT_USB1VP |
| Z220 | USB1_VM | I | GPIO3[8] | IT_USB1VM |
| Z221 | USB2_nOE | O | GPIO3[9] | IT_USB2NOE |
| Z222 | USB2_SUSPEND | O | GPIO3[10] | IT_USB2SUSPEND |
| Z223 | USB2_SPEED | O | GPIO3[11] | IT_USB2SPEED |
| Z224 | USB2_SCL | O | GPIO3[12] | IT_USB2SCL |
| Z225 | USB2_SDA | I/O | GPIO3[13] | IT_USB2SDA |
| Z226 | USB2_nRST | O | GPIO3[14] | IT_USB2NRST |
| Z227 | USB2_nINT | I | GPIO3[15] | IT_USB2NINT |
| Z228 | USB2_VP | I | GPIO3[16] | IT_USB2VP |
| Z229 | USB2_VM | I | GPIO3[17] | IT_USB2VM |
| Z230 | USB2_RCV | I | GPIO3[18] | IT_USB2RCV |
| Z128 | DAC_Clk | O | GPIO0[0] | IT_DACCLK |
| Z129 | DAC_Din | O | GPIO0[1] | IT_DACDI |
| Z130 | DAC_nCS | O | GPIO0[2] | IT_DACNCS |
| Z131 | ADC_Din | O | GPIO0[3] | IT_ADCDI |
| Z132 | ADC_nCS | O | GPIO0[4] | IT_ADCNCS |
| Z133 | ADC_DOut | I | GPIO0[5] | IT_ADCDO |
| Z134 | ADC_Busy | I | GPIO0[6] | IT_ADCBSY |
| Z135 | ADC_Clk | O | GPIO0[7] | IT_ADCCLK |
| Z136 | CAN1_TxD | O | GPIO0[8] | IT_CAN1TX |
| Z137 | CAN1_RxD | I | GPIO0[9] | IT_CAN1RX |
| Z138 | CAN2_TxD | O | GPIO0[10] | IT_CAN2TX |
| Z139 | CAN2_RxD | I | GPIO0[11] | IT_CAN2RX |
| Z140 | UART_TxD | O | GPIO0[12] | IT_UARTTX |
| Z141 | UART_RxD | I | GPIO0[13] | IT_UARTRX |
| Z142 | UART_nRTS | O | GPIO0[14] | IT_UARTRTS |
| Z143 | UART_nDTR | O | GPIO0[15] | IT_UARTDTR |
| Z144 | UART_nCTS | I | GPIO0[16] | IT_UARTCTS |
| Z145 | UART_nDSR | I | GPIO0[17] | IT_UARTDSR |
| Z146 | UART_nDCD | I | GPIO0[18] | IT_UARTDCD |
| Z147 | UART_nRI | I | GPIO0[19] | IT_UARTRI |
| Z152 | STEPPER1_IN[1] | O | GPIO0[20] | IT_STEP1A |
| Z153 | STEPPER1_IN[2] | O | GPIO0[21] | IT_STEP1B |
| Z154 | STEPPER1_IN[3] | O | GPIO0[22] | IT_STEP1C |
| Z155 | STEPPER1_IN[4] | O | GPIO0[23] | IT_STEP1D |
| Z156 | STEPPER2_IN[1] | O | GPIO0[24] | IT_STEP2A |
| Z157 | STEPPER2_IN[2] | O | GPIO0[25] | IT_STEP2B |
| Z158 | STEPPER2_IN[3] | O | GPIO0[26] | IT_STEP2C |
| Z159 | STEPPER2_IN[4] | O | GPIO0[27] | IT_STEP2D |

Table 1-4 GPIO pin allocation

2 Software Interface

The Versatile/IT1 example design is based on the standard example provided with Versatile/LT-XC2V4000+ and described in detail in application note AN119. The example will work in combination with the following sets of boards. The example design requires the Versatile/IT1 to be fitted on top of the Logic Tile and uses IMAGE 0.

1. Integrator/AP + Core Module + Versatile/IM-LT1 + Versatile/LT-XC2V6000 (or 8000) + Versatile/IT1
2. Integrator/AP + Versatile/IM-LT1 + Versatile/LT-XC2V6000 (or 8000) + Versatile/IT1 (using EXP slot)
3. Integrator/CP + Core Module + Versatile/IM-LT1 + Versatile/LT-XC2V6000 (or 8000) + Versatile/IT1
4. Versatile/PB926EJ-S + Versatile/LT-XC2V6000 (or 8000) + Versatile/IT1

2.1 ZBT SSRAM, APB and GPIO registers

This standard example provides two AHB ZBT SSRAM interfaces to 2MB static memory devices shown in Figure 2-1. The example also provided a set of APB control registers for oscillators, LEDs, switches and an interrupt controller. The ZBT interface remains unchanged from the original design while two new registers have been added to the control register list as shown in Table 2-1. The GPIO registers are shown in Figure 2-2

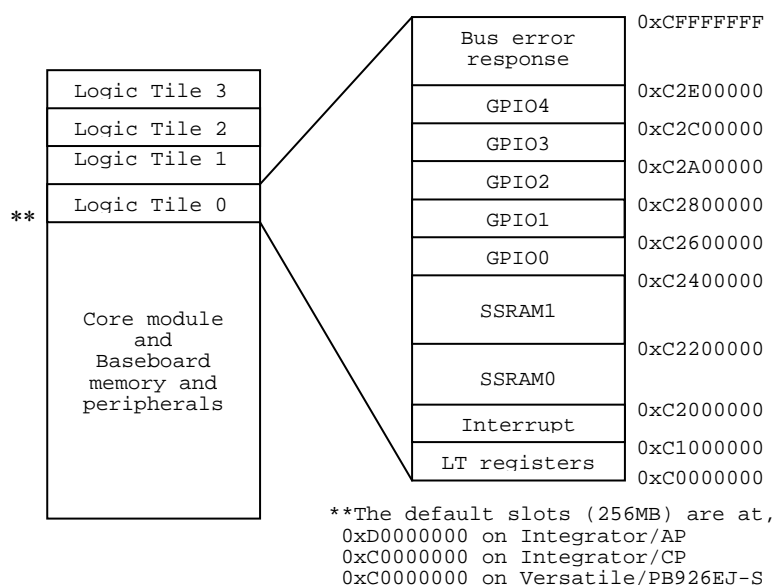


Figure 2-1 Memory Map

| Offset address | Name | Type | Size | Function |
|----------------|-------------|------------|------|---------------------------------|
| 0x00000000 | LT_OSC0 | Read/Write | 19 | Oscillator 0 divisor register |
| 0x00000004 | LT_OSC1 | Read/Write | 19 | Oscillator 1 divisor register |
| 0x00000008 | LT_OSC2 | Read/Write | 19 | Oscillator 2 divisor register |
| 0x0000000C | LT_LOCK | Read/Write | 17 | Oscillator lock register |
| 0x00000010 | LT_LEDS | Read/Write | 4 | User LEDs register,LT LED[3:0] |
| 0x00000014 | IM_LT1_LEDS | Read/Write | 8 | User LEDs register,IM LED[7:0] |
| 0x00000018 | LT_INT | Read/Write | 1 | Push button interrupt register |
| 0x0000001C | LT_SW | Read | 4 | Switches register,LT S1[4:1] |
| 0x00000020 | IM_LT1_SW | Read | 8 | Switches register,IM S3[8:1] |
| 0x00000024 | IT1_LEDS | Read/Write | 8 | User LEDs register,IT1 LED[1:8] |
| 0x00000028 | IT1_SW | Read | 8 | Switches register,IT1 S2[1:8] |

Table 2-1 APB registers

| Offset address | Name | Type | Size | Function |
|----------------|--------------|------------|------|---------------------------------|
| 0x00000000 | IT1_GPIO0SET | Write | 32 | Set output |
| 0x00000004 | IT1_GPIO0CLR | Read/Write | 32 | Clear output |
| 0x00000008 | IT1_GPIO0DIR | Read/Write | 32 | Direction control (0=I/P,1=O/P) |
| 0x0000000C | IT1_GPIO0IN | Read | 32 | Read pin value |

Table 2-1-1 GPIO registers

2.2 Selftest software

Provided with the Versatile/IT1 example design is a simple test program for testing the Versatile/IT1 board as well as providing some example code fragments. The test methods used are shown in Table 2-2.

| Peripheral | Test method | Example code fragments |
|--------------------------|---|--------------------------------------|
| I ² C 1 and 2 | Loopback of I ² C 1 and 2 | Initialize, I2CRead() and I2CWrite() |
| IDE | Loopback of Control and Data signals | GPIO control |
| USB Host and OTG | Loopback of Host and OTG | GPIO control |
| Camera | Loopback of Control and Data signals | GPIO control |
| ADC and DAC | DAC[3:0] drives ADC[3:0] and ADC[7:4] | Read_ADC(), Set_DAC() |
| UART | TX to DCD/CTS, DTR to RX/DSR, RTS to RI | GPIO control |
| CAN 1 and 2 | Loopback of CAN 1 and 2 | GPIO control |
| Stepper 1 and 2 | Drives stepper motors on Stepper 1/2 | Stepper motor sequencing |
| Switch and LEDs | Switch read and displayed on LEDs | APB register control |

Table 2-2 Peripheral test methods

The \SW\ directory contains the source code for the selftest software and an ADS1.2 project file selftest.mcp or RVCT2.1 batch file build.bat.

2.2.1 I²C software interface

The dual I²C interface consists of two Philips PCA9564 Parallel-bus to I²C-bus controllers. The controllers contain configuration registers and control logic to perform master or slave I²C bus control. The I²C controllers connect to GPIO4 with the register assignments shown in Table 2-2-1.

| | | | | | | | | | |
|----------|-------|-------|-----|-----|------|------|--------|------|--------|
| 31:17 | 16 | 15 | 14 | 13 | 12 | 11 | 10:9 | 8 | 7:0 |
| Reserved | nINT2 | nINT1 | nWR | nRD | nCS1 | nCS2 | A[1:0] | nRST | D[7:0] |

Table 2-2-1, GPIO4 - I²C register assignment

The selftest I2CTest(); procedure sets I²C1 as a master controller and I²C2 as a slave controller. Functions I2CWrite() and I2CRead() are used to configure the controllers and read/write data values. The test software provided is derived from Philips application note AN10148. The application note describes in detail how to **Initialize** the controllers (Figure 5) and perform **Master Transmit** (Figure 6) or **Slave Receive** (Figure 8) mode operations. The I²C loopback cable is shown in Figure 2-3-1.

Implementation of the I²C interface through GPIO adds a level of software overhead which reduces the overall performance of the interface. Improvements on the design could be realized by implementing the I²C controllers as memory mapped AHB devices.

IDE software interface

The IDE interface consists of a 40way 0.1" IDC header based on the ATA-3 interface standard. Two buffers interface between the 3V3 Logic Tile fpga signals and the 5V IDE signals. U12 is used as a bidirectional buffer to read/write 16bit data values. U13 is used to interface to the IDE control signals. The IDE buffers connect to GPIO1 with the register assignments shown in Table 2-2-2-1. The buffer directions and enables connect to GPIO2 with the register assignments shown in Table 2-2-2-2.

| | | | | | | | | | | | | | |
|------|-------|-------|---------|------|-------|------|-----|---------|---------|--------|------|------|---------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23:22 | 21:19 | 18 | 17 | 16 | 15:0 |
| Rsvd | nACTV | PDIAG | nIOCS16 | IRQR | IORDY | DMAR | RST | CS[1:0] | DA[2:0] | DMAACK | nIOR | nIOW | D[15:0] |

Table 2-2-2, GPIO1 - IDE register assignment

| | | | |
|----------|-----------|-----------|--------|
| 31:22 | 21:19 | 18:16 | 15:0 |
| Reserved | nBOE[2:0] | BDIR[2:0] | Camera |

Table 2-2-2-1, GPIO2 - IDE buffer control register assignment

The IDE GPIO interface has been implemented to allow easy testing of all IDE signals. The loopback test configures all 16 data signals as inputs and drives them from the control signals which are configured as outputs. The IDE loopback cable is shown in Figure 2-3-2.

Although in principle the GPIO interface provided could be used to implement an IDE controller the performance would be severely limited by the software overhead required. Therefore the IDE controller should be implemented in the Logic Tile fpga using a DMA engine to achieve a good data transfer rate.

2.2.2 USB software interface

The USB interfaces consist of a Philips PDIUSBP11A USB host transceiver and a Philips ISP1301 USB OTG (On-The-Go) transceiver. Both devices act as level translators and buffers from the 3V3 fpga logic signals to the USB bus signals. They do not perform any of the higher level USB control functions required to implement a full USB interface. A full USB Host or OTG interface would have to be implemented in the Logic Tile fpga design with the appropriate software drivers.

| USB2 | | | | | | | | | | USB1 | | | | | | | | | |
|-------|----|----|------|------|-----|-----|-------|---------|-----|------|----|-----|-------|-----|-----|---------|-----|------|--|
| 31:18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Rsvd | VM | VP | nINT | nRST | SDA | SCL | SPEED | SUSPEND | nOE | VM | VP | RCV | SPEED | VPO | VMO | SUSPEND | nOE | MODE | |

Table 2-2-3, GPIO3 - USB register assignment

The two USB controllers connect to GPIO3 with the register assignments shown in Table 2-2-3. The loopback test configures the USB Host (USB1) to drive the USB bus which is read back from the USB OTG (USB2). The USB loopback cable is a standard Host to OTG cable which is shown in Figure 2-3-3.

2.2.3 Camera software interface

The generic digital camera interface is compatible with most 8 bit data cameras for example the OmniVision OV9640 (1.3M pixel). Although an additional interface board may be required to provide power and clock control to the camera module.

| | | | | | | | | | |
|----------|-------|--------|---------|-----|------|------|-----|-----|--------|
| 31:22 | 21:16 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7:0 |
| Reserved | IDE | nRESET | STANDBY | CLK | VSNC | HSNC | SDA | SCL | D[7:0] |

Table 2-2-4, GPIO2 - Camera register assignment

The pin assignments given are for guidance only as all camera signals connect direct to the Logic Tile fpga and may be redefined if required.

The camera loopback test configures all camera control signals and D7 as inputs and drives them from data signals D[6:0] which are configured as outputs. The camera loopback cable is shown in Figure 2-3-4.

2.2.4 ADC, DAC, CAN, UART and Stepper motor interfaces

The ADC, DAC, CAN, UART and Stepper motor interfaces share GPIO3 with the register assignments shown in Table 2-2-5.

| STEPPER 1/2 | | | UART | | | | | | | | CAN 1/2 | | | | ADC | | | | | DAC | | |
|-------------|------------|------------|------|------|------|------|------|------|-----|-----|---------|------|------|------|-----|------|------|-----|-----|-----|-----|-----|
| 31:28 | 27:24 | 23:20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Rsvd | STEP2[3:0] | STEP1[3:0] | nRI | nDCD | nDSR | nCTS | nDTR | nRTS | RXD | TXD | RXD2 | TXD2 | RXD1 | TXD1 | CLK | BUSY | DOUT | nCS | DIN | nCS | DIN | CLK |

Table 2-2-5, GPIO3 - ADC, DAC, CAN, UART and Stepper motor register assignment

2.2.5 ADC and DAC software interface

The ADC interface consists of an ADS7844E, 8 channel 12bit 200KHz converter with a 2.5V reference. The DAC interface consists of an AD5324, 4 channel 12bit 125KHz converter with a 2.5V reference. The input/output range of the converters is 0 to 2.5V giving a 610uV Voltage resolution. To achieve high measurement accuracy it is essential to ensure that appropriate ground connections have been made and that coupling from digital signals are kept to a minimum. Filters on the ADC inputs help reduce input noise above 200KHz.

The selftest software provides two functions Read_ADC() and Set_DAC() to read an ADC input value or set a DAC output value. The loopback test links DAC[3:0] to ADC[7:4] and ADC[3:0]. The test ramps the DAC output levels and measures the resulting ADC levels checking for errors. The ADC/DAC loopback cable is shown in Figure 2-3-5. Although the serial bus interface to the devices is implemented in software this method is adequate given the speed of the converters. However the software overhead may be reduced if required by implementing an SSP controller such as the PL022 SSPC PrimeCell in the Logic Tile fpga.

2.2.6 UART software interface

The UART interface consists of a MAX3243E RS-232 transceiver with 3 output buffers and 5 input buffers. The loopback test links TX to DCD/CTS, DTR to RX/DSR and RTS to RI testing connectivity between these pins. The UART loopback cable is shown in Figure 2-3-6.

The UART interface requires a UART controller to perform the necessary serial data stream and handshake signals. This should be implemented in the Logic Tile fpga using a controller such as the PL011 PrimeCell.

2.2.7 CAN Bus software interface

The CAN interfaces consist of two Philips TJA1050 CAN transceivers. Both devices act as level translators and buffers from the 3V3 fpga logic signals to the CAN bus signals. They do not perform any of the higher level CAN control functions required to implement a full CAN interface. A full CAN bus interface would have to be implemented in the Logic Tile fpga design with the appropriate software drivers.

The CAN loopback test initially drives the CAN bus from CAN1 and is read from CAN2. The direction is then reversed and CAN2 drives the bus which is read from CAN1. The CAN loopback cable is shown in Figure 2-3-7.

2.2.8 Stepper Motor software interface

The Stepper motor interfaces consist of two Allegro UDN2559LB protected quad power drivers. The devices act as level translators and buffers from the 3V3 fpga logic signals to high current/voltage drivers. The buffers are capable of driving lamps or inductive loads such as stepper motors, solenoids or relays. Each buffer output pin is rated at 25V 500mA max, includes a reverse protection diode and is thermally protected against short circuits.

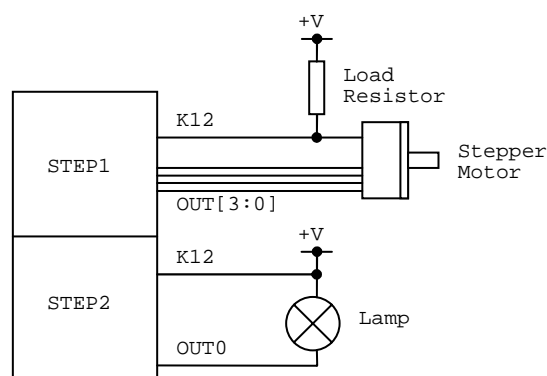


Figure 2-2-8 Example Stepper Motor/Lamp Interface

The Stepper motor test software drives two stepper motors using a conventional step sequence generated in software. An example motor is the Astrosyn Y129 (12V/75R Hybrid size 16/17).

2.3 Selftest cable harness

J1 - 16way IDC

13 -> 14

15 -> 16

Figure 2-3-1, I²C loopback cable

J2 - 16way IDC

13 -> 5 -> 9

14 -> 6 -> 10

15 -> 7 -> 11

16 -> 8 -> 12

Figure 2-3-5, ADC/DAC loopback cable

J6 - 40way IDC

17 -> 23

15 -> 25

13 -> 29

11 -> 35

9 -> 33

7 -> 36

5 -> 37

3 -> 38

4 -> 21

6 -> 27

8 -> 31

10 -> 32

12 -> 34

14 -> 39 -> 18

1 -> 16

Figure 2-3-2, IDE loopback cable

J9 - 10way IDC

5 -> 1 -> 6

7 -> 3 -> 2

8 -> 4

Figure 2-3-6, UART loopback cable

J4 - 10way IDC

1 -> 7

3 -> 9

Figure 2-3-7, CAN loopback cable

J3 -> J4 USB Host to OTG
cable. Mini-B-to-USB Type A
cable assembly.



Figure 2-3-3, USB loopback cable

J7 - 20way IDC

3 -> 2 -> 17

5 -> 4

7 -> 10

9 -> 8

11 -> 14

13 -> 18

15 -> 20

Figure 2-3-4, Camera loopback cable