RealDigital

Blackboard

Basic Zynq Programmers Reference

Rev B

Author: GA

January 13, 2019

Table of Contents

TABLE OF CONTENTS **1**

INTRODUCTION **4**

GENERIC INTERRUPT CONTROLLER (GIC) [1] **5**

DESCRIPTION 5
OVERVIEW OF REGISTERS* 7
REGISTERS 8
PROGRAMMING EXAMPLES IN C 22

GENERAL PURPOSE I/O (GPIO) MODULE [1] **27**

DESCRIPTION 27
OVERVIEW OF REGISTERS 27
REGISTERS 28
PROGRAMMING EXAMPLES IN C 30

GLOBAL TIMER COUNTER (GTC) MODULE [1] **31**

DESCRIPTION 31
OVERVIEW OF REGISTERS 31
REGISTERS 32

H-BRIDGE MODULE **35**

DESCRIPTION 35
OVERVIEW OF REGISTERS 35
REGISTERS 36
PROGRAMMING EXAMPLE IN C (HB) 37

INEMO INERTIAL MODULE (NAV SENSOR) [3] **38**

DESCRIPTION 38
REGISTERS: ACCELEROMETER AND GYROSCOPE (CS LINE 0) [3] (NAV) 42
REGISTERS: MAGNETOMETER (CS LINE 1) [3] (NAV) 44

I2C CONTROLLER MODULES (I2C0 AND I2C1) [1] **45**

DESCRIPTION 45
OVERVIEW OF REGISTERS 45
REGISTERS 46

LED MODULE **51**

Description 51
Overview of Registers 51
Registers 52
Programming Examples in C 54

LM75BDP NXP Temperature Sensor [4] 55

Description 55
Overview of Registers [4] 55
Reading Temperature Data [4] 55

MIO PIN CONTROL [1] 57

Description 57
Overview of Registers 58
Registers 59
Programming Example in C 60

RGB LED MODULE 61

Description 61
Overview of Registers 61
Registers 64
Programming Examples in C 68

SEVEN SEGMENT DISPLAY MODULE 69

Description 69
Overview of Registers 70
Registers 70
Programming Examples in C 72

SPI CONTROLLER MODULES (SPI0 AND SPI1) [1] 73

Description 73
Overview of Registers 73
Registers 74

SWITCHES AND BUTTONS MODULE 79

Description 79
Overview of Registers 79
Registers 79
Programming Example in C 79

TRIPLE TIMER COUNTER MODULES (TTC0 AND TTC1) [1] 80

DESCRIPTION 80
OVERVIEW OF REGISTERS 80
REGISTERS 82

SYSTEM LEVEL CONTROL REGISTERS (SLCR) [1] 92

DESCRIPTION 92
OVERVIEW OF REGISTERS 92
REGISTERS 92
PROGRAMMING EXAMPLE IN C (SLCR) 93

UART CONTROLLER MODULES (UART0 AND UART1) [1] 94

DESCRIPTION 94
OVERVIEW OF REGISTERS 96
REGISTERS 97

XADC MODULE [5] 107

DESCRIPTION 107
OVERVIEW OF REGISTERS 108
REGISTERS 108
PROGRAMMING EXAMPLE IN C (XADC) 110

REFERENCES 111

REVISION HISTORY 112

Introduction

The Blackboard ships with a Zynq FPGA configuration containing IP blocks that let the processor communicate with all PL-connected devices using memory-mapped registers. This programmer's reference defines the AXI bus addresses and register contents for all IP blocks created in addition to the default Xilinx module's.

A Xilinx Vivado project that defines the IP blocks can be downloaded from the Real Digital website.

Generic Interrupt Controller (GIC) [1]

Description

The ZYNQ chip has a single core ARM processor which has a Generic Interrupt Controller to process interrupts. Before configuring the GIC, you must ensure that you turned off your IRQ interrupts and that your processor has enough privileges to execute interrupts. This can be accomplished by configuring the ARM cortex v-7 A9 current processor status register (cpsr), see figure 1. ARM Cortex A9 has several modes which can be changed by reconfiguring the first four bits within the cpsr register. Bits 7 and 6 are necessary for masking the interrupts. For Blackboard, you need to configure the processor to System mode (0b11111) and set the IRQ and FIQ bits high to disable those interrupts. Examples in C are provided below on how you can enable/disable those interrupts.

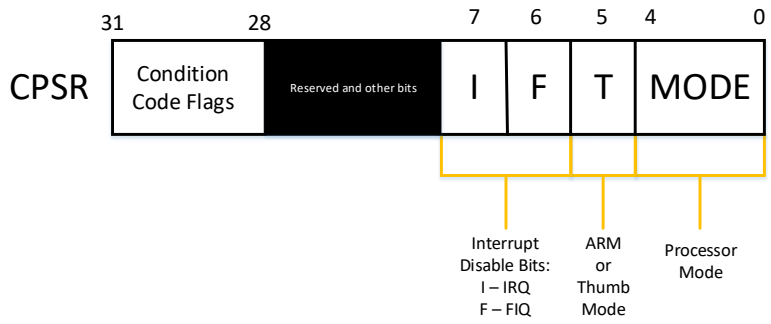

Figure 1. Simple CPSR Register

Configure GIC

Before configuring the GIC, you need to locate which interrupt IDs you need to configure, you can look up IRQ ID values for specific modules within the Zynq chip from Zynq-7000 AP SoC Technical Reference Manual on page 230. [1] Configuring the Generic Interrupt Controller is a multistep process and requires some time to accomplish it. When broken down to doing it line by line, then there is a total of 9 steps you need to go through to set up the GIC for an interrupt:

1. Disable interrupt masks and any CPU's of handling interrupts (ICDIPTR and ICDICER)
2. Disable the distributor using Distributor Control Register (ICDDCR)
3. Set priority levels in the Interrupt Priority Register (ICDIPR)
4. Configure Interrupt Processor Targets Registers (ICDIPTR)
5. Set Interrupt Sensitivity in Interrupt Configuration Register (ICDICFR)
6. Enable Interrupts in the Interrupt Set-Enable Register (ICDISER)
7. Enable all priority levels in the Interrupt Priority Mask Register (ICCPMR)
8. Enable Interrupts in the CPU Interface Control Register (ICCICR)
9. Enable Distributor in the Distributor Control Register (ICDDCR)

When serving the interrupts, you need to determine the interrupt ID or IRQ ID by reading the Interrupt Acknowledge Register (ICCIAR) (Zynq-7000 AP SoC Technical Reference Manual page 1446). Then determine which interrupt it was such as Triple Timer Counter Match Interrupt, GPIO pin 16 interrupt, I2C data transfer completed interrupt, etc. Completely depends on your software program. Finally, clear any outstanding interrupt status bits and you need to notify the GIC that the interrupt has been served, which can be done by acknowledging the interrupt. Write the read value from ICCIAR register and write it to the End of Interrupt Register (ICCEOIR) (Zynq-7000 AP SoC Technical Reference Manual pages 1446 - 1447).

Overview of Registers*

Base Address + Offset (hex)	Register Name	Reset Value (hex)	Access Type	Description
0xF8F00000 + 0x100	ICCICR	0x0	Read/Write	CPU Interface Control Register
0xF8F00000 + 0x104	ICCPMR	0x0	Read/Write	Interrupt Priority Mask Register
0xF8F00000 + 0x10C	ICCIAR	0x3FF	Read/Write	Interrupt Acknowledge Register
0xF8F00000 + 0x110	ICCEOIR	0x0	Read/Write	End of Interrupt Register
0xF8F00000 + 0x1000	ICDDCR	0x0	Read/Write	Distributor Control Register
0xF8F00000 + 0x1100	ICDISER0	0xFFFF	Read/Write	Interrupt Set-Enable Register 0
0xF8F00000 + 0x1104	ICDISER1	0x0	Read/Write	Interrupt Set-Enable Register 1
0xF8F00000 + 0x1108	ICDISER2	0x0	Read/Write	Interrupt Set-Enable Register 2
0xF8F00000 + 0x1180	ICDICER0	0x0	Read/Write	Interrupt Clear-Enable Register 0
0xF8F00000 + 0x1184	ICDICER1	0x0	Read/Write	Interrupt Clear-Enable Register 1
0xF8F00000 + 0x1188	ICDICER2	0x0	Read/Write	Interrupt Clear-Enable Register 2
0xF8F00000 + 0x1418	ICDIPR6	0x0	Read/Write	Interrupt Priority Register 6
0xF8F00000 + 0x1424	ICDIPR9	0x0	Read/Write	Interrupt Priority Register 9
0xF8F00000 + 0x1428	ICDIPR10	0x0	Read/Write	Interrupt Priority Register 10
0xF8F00000 + 0x142C	ICDIPR11	0x0	Read/Write	Interrupt Priority Register 11
0xF8F00000 + 0x1434	ICDIPR13	0x0	Read/Write	Interrupt Priority Register 13
0xF8F00000 + 0x1438	ICDIPR14	0x0	Read/Write	Interrupt Priority Register 14
0xF8F00000 + 0x1444	ICDIPR17	0x0	Read/Write	Interrupt Priority Register 17
0xF8F00000 + 0x1450	ICDIPR20	0x0	Read/Write	Interrupt Priority Register 20
0xF8F00000 + 0x1818	ICDIPTR6	0x0	Read/Write	Interrupt Processor Targets Register 6
0xF8F00000 + 0x1824	ICDIPTR9	0x0	Read/Write	Interrupt Processor Targets Register 9
0xF8F00000 + 0x1828	ICDIPTR10	0x0	Read/Write	Interrupt Processor Targets Register 10
0xF8F00000 + 0x182C	ICDIPTR11	0x0	Read/Write	Interrupt Processor Targets Register 11
0xF8F00000 + 0x1834	ICDIPTR13	0x0	Read/Write	Interrupt Processor Targets Register 13
0xF8F00000 + 0x1838	ICDIPTR14	0x0	Read/Write	Interrupt Processor Targets Register 14
0xF8F00000 + 0x1844	ICDIPTR17	0x0	Read/Write	Interrupt Processor Targets Register 17
0xF8F00000 + 0x1850	ICDIPTR20	0x0	Read/Write	Interrupt Processor Targets Register 20
0xF8F00000 + 0x1C04	ICDICFR1	0x55555555	Read/Write	Interrupt Configuration Register 1
0xF8F00000 + 0x1C08	ICDICFR2	0x55555555	Read/Write	Interrupt Configuration Register 2
0xF8F00000 + 0x1C0C	ICDICFR3	0x55555555	Read/Write	Interrupt Configuration Register 3
0xF8F00000 + 0x1C10	ICDICFR4	0x55555555	Read/Write	Interrupt Configuration Register 4
0xF8F00000 + 0x1C14	ICDICFR5	0x55555555	Read/Write	Interrupt Configuration Register 5

* Note: This overview of registers doesn't include all interrupt ID registers, if other ID's are needed you can find them in Zynq-7000 AP SoC Technical Reference Manual 1432-1511 and 230-231.

Registers

ICCICR (0xF8F00000 + 0x100) (GIC)

Name	Bit Range	Type	Reset	Description
Undefined	31:5	N/A	N/A	Bits 31:5 have no effect.
SBPR	4:4	R/W	0x0	Controls whether the CPU interface uses the Secure or Non-Secure Binary Point Register for preemption. 0: use the Secure Binary Point Register for Secure interrupts and use the Non-Secure Binary Point Register for Non-secure interrupts. 1: use the Secure Binary Point Register for both Secure and Non-secure interrupts.
FIQEn	3:3	R/W	0x0	Controls whether the GIC signals Secure interrupts to a target processor using the FIQ or the IRQ signal. 0: using IRQ, 1: using FIQ. The GIC always signals Non-secure interrupts using IRQ.
AckCtl	2:2	R/W	0x0	Controls whether a Secure read of the ICCIAR, when the highest priority pending interrupt is Non-secure, causes the CPU interface to acknowledge the interrupt.
EnableNS	1:1	R/W	0x0	An alias of the Enable bit in the Non-secure ICCICR. This alias bit means Secure software can enable the signal of Non-secure interrupts.
EnableS	0:0	R/W	0x0	Global enable for the signaling of Secure interrupts by the CPU interfaces to the connected processors.

ICDPMR (0xF8F00000 + 0x104) (GIC)

Name	Bit Range	Type	Reset	Description
Undefined	31:8	N/A	N/A	Bits 31:8 have no effect.
Priority Mask	7:0	R/W	0x0	The priority mask level for the CPU interface. If the priority of an interrupt is higher than the value indicated by this field, the interface signals the interrupt to the processor.

ICCIAR (0xF8F00000 + 0x10C) (GIC)

Name	Bit Range	Type	Reset	Description
Undefined	31:13	N/A	N/A	Bits 31:13 have no effect.
CPU_ID	12:10	R/W	0x0	Identifies the processor that requested the interrupt. Returns the number of the CPU interface that made the request.
IRQ ID or Interrupt ID	9:0	R/W	0x0	The interrupt ID. This read acts as an acknowledge for the interrupt if AckCtl bit is enabled within ICCICR register.

ICCEOIR (0xF8F00000 + 0x110) (GIC)

Name	Bit Range	Type	Reset	Description
Undefined	31:13	N/A	N/A	Bits 31:13 have no effect.
CPU_ID	12:10	R/W	0x0	On completion of the processing of an SGI, this field contains the CPUID value from the corresponding ICCIAR access.
EOIINTID	9:0	R/W	0x0	The ACKINTID value from the corresponding ICCIAR access.

ICDDCR (0xF8F00000 + 0x1000) (GIC)

Name	Bit Range	Type	Reset	Description
Undefined	31:2	N/A	N/A	Bits 31:2 have no effect.
Enable Non-Secure	1:1	R/W	0x0	0 = disables all Non-secure interrupts control bits in the distributor from changing state because of any external stimulus change that occurs on the corresponding SPI or PPI signals 1 = enables the distributor to update register locations for Non-secure interrupts
Enable Secure	0:0	R/W	0x0	0 = disables all Secure interrupt control bits in the distributor from changing state because of any external stimulus change that occurs on the corresponding SPI or PPI signals. 1 = enables the distributor to update register locations for Secure interrupts.

ICDISER0 (0xF8F00000 + 0x1100) (GIC)

Name	Bit Range	Type	Reset	Description
Set Enable 0	31:0	R/W	0x0	Writing 1 to a Set-enable bit enables forwarding of the corresponding interrupt to the CPU interfaces. Bits 31:0 correspond to interrupt IDs 31:0, respectively.

ICDISER1 (0xF8F00000 + 0x1104) (GIC)

Name	Bit Range	Type	Reset	Description
Set Enable 1	31:0	R/W	0x0	Writing 1 to a Set-enable bit enables forwarding of the corresponding interrupt to the CPU interfaces. Bits 31:0 correspond to interrupt IDs 63:32, respectively.

ICDISER2 (0xF8F00000 + 0x1108) (GIC)

Name	Bit Range	Type	Reset	Description
Set Enable 2	31:0	R/W	0x0	Writing 1 to a Set-enable bit enables forwarding of the corresponding interrupt to the CPU interfaces. Bits 31:0 correspond to interrupt IDs 95:64, respectively.

ICDICER0 (0xF8F00000 + 0x1180) (GIC)

Name	Bit Range	Type	Reset	Description
Clear Enable 0	31:0	R/W	0x0	Writing 1 to a Clear-enable bit disables forwarding of the corresponding interrupt to the CPU interfaces. Bits 31:0 correspond to interrupt IDs 31:0, respectively.

ICDICER1 (0xF8F00000 + 0x1184) (GIC)

Name	Bit Range	Type	Reset	Description
Clear Enable 1	31:0	R/W	0x0	Writing 1 to a Clear-enable bit disables forwarding of the corresponding interrupt to the CPU interfaces. Bits 31:0 correspond to interrupt IDs 63:32, respectively.

ICDICER2 (0xF8F00000 + 0x1188) (GIC)

Name	Bit Range	Type	Reset	Description
Clear Enable 2	31:0	R/W	0x0	Writing 1 to a Clear-enable bit disables forwarding of the corresponding interrupt to the CPU interfaces. Bits 31:0 correspond to interrupt IDs 95:64, respectively.

ICDIPR6 (0xF8F00000 + 0x1418) (GIC)

Name	Bit Range	Type	Reset	Description
IRQ ID #27 Priority Level	31:24	R/W	0x0	Interrupt ID#27 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
Undefined	23:0	N/A	0x0	Reserved. Bits 23:0 have no effect.

ICDIPR9 (0xF8F00000 + 0x1424) (GIC)

Name	Bit Range	Type	Reset	Description
IRQ ID #39 Priority Level	31:24	R/W	0x0	Interrupt ID#39 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #38 Priority Level	23:16	R/W	0x0	Interrupt ID#38 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #37 Priority Level	15:8	R/W	0x0	Interrupt ID#37 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #36 Priority Level	7:0	R/W	0x0	Interrupt ID#36 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.

ICDIPR10 (0xF8F00000 + 0x1428) (GIC)

Name	Bit Range	Type	Reset	Description
IRQ ID #43 Priority Level	31:24	R/W	0x0	Interrupt ID#43 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #42 Priority Level	23:16	R/W	0x0	Interrupt ID#42 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #41 Priority Level	15:8	R/W	0x0	Interrupt ID#41 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #40 Priority Level	7:0	R/W	0x0	Interrupt ID#40 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.

ICDIPR11 (0xF8F00000 + 0x142C) (GIC)

Name	Bit Range	Type	Reset	Description
IRQ ID #47 Priority Level	31:24	R/W	0x0	Interrupt ID#47 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #46 Priority Level	23:16	R/W	0x0	Interrupt ID#46 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #45 Priority Level	15:8	R/W	0x0	Interrupt ID#45 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #44 Priority Level	7:0	R/W	0x0	Interrupt ID#44 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.

ICDIPR13 (0xF8F00000 + 0x1434) (GIC)

Name	Bit Range	Type	Reset	Description
IRQ ID #55 Priority Level	31:24	R/W	0x0	Interrupt ID#55 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #54 Priority Level	23:16	R/W	0x0	Interrupt ID#54 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #53 Priority Level	15:8	R/W	0x0	Interrupt ID#53 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #52 Priority Level	7:0	R/W	0x0	Interrupt ID#52 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.

ICDIPR14 (0xF8F00000 + 0x1438) (GIC)

Name	Bit Range	Type	Reset	Description
IRQ ID #59 Priority Level	31:24	R/W	0x0	Interrupt ID#59 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #58 Priority Level	23:16	R/W	0x0	Interrupt ID#58 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #57 Priority Level	15:8	R/W	0x0	Interrupt ID#57 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #56 Priority Level	7:0	R/W	0x0	Interrupt ID#56 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.

ICDIPR17 (0xF8F00000 + 0x1444) (GIC)

Name	Bit Range	Type	Reset	Description
IRQ ID #71 Priority Level	31:24	R/W	0x0	Interrupt ID#71 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #70 Priority Level	23:16	R/W	0x0	Interrupt ID#70 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #69 Priority Level	15:8	R/W	0x0	Interrupt ID#69 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #68 Priority Level	7:0	R/W	0x0	Interrupt ID#68 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.

ICDIPR20 (0xF8F00000 + 0x1450) (GIC)

Name	Bit Range	Type	Reset	Description
IRQ ID #83 Priority Level	31:24	R/W	0x0	Interrupt ID#83 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #82 Priority Level	23:16	R/W	0x0	Interrupt ID#82 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #81 Priority Level	15:8	R/W	0x0	Interrupt ID#81 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.
IRQ ID #80 Priority Level	7:0	R/W	0x0	Interrupt ID#80 Priority Level. Only the upper 5 bits of each 8-bit field are writable; the lower bits are always 0's.

ICDIPTR6 (0xF8F00000 + 0x1818) (GIC)

Name	Bit Range	Type	Reset	Description
Undefined	31:26	N/A	N/A	Bits 31:26 have no effect.
IRQ ID #27	25:24	R	0x1	Targeted CPU(s) for interrupt ID#27 01: CPU 0 targeted 10: CPU 1 targeted
Undefined	23:0	N/A	N/A	Bits 23:18 have no effect.

ICDIPTR9 (0xF8F00000 + 0x1824) (GIC)

Name	Bit Range	Type	Reset	Description
Undefined	31:26	N/A	N/A	Bits 31:26 have no effect.
IRQ ID #39	25:24	R/W	0x0	Targeted CPU(s) for interrupt ID#39 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	23:18	N/A	N/A	Bits 23:18 have no effect.
IRQ ID #38	17:16	R/W	0x0	Targeted CPU(s) for interrupt ID#38 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	15:10	N/A	N/A	Bits 15:10 have no effect.
IRQ ID #37	9:8	R/W	0x0	Targeted CPU(s) for interrupt ID#37 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	7:2	N/A	N/A	Bits 7:2 have no effect.
IRQ ID #36	1:0	R/W	0x0	Targeted CPU(s) for interrupt ID#36 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted

ICDIPTR10 (0xF8F00000 + 0x1828) (GIC)

Name	Bit Range	Type	Reset	Description
Undefined	31:26	N/A	N/A	Bits 31:26 have no effect.
IRQ ID #43	25:24	R/W	0x0	Targeted CPU(s) for interrupt ID#43 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	23:18	N/A	N/A	Bits 23:18 have no effect.
IRQ ID #42	17:16	R/W	0x0	Targeted CPU(s) for interrupt ID#42 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	15:10	N/A	N/A	Bits 15:10 have no effect.
IRQ ID #41	9:8	R/W	0x0	Targeted CPU(s) for interrupt ID#41 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	7:2	N/A	N/A	Bits 7:2 have no effect.
IRQ ID #40	1:0	R/W	0x0	Targeted CPU(s) for interrupt ID#40 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted

ICDIPTR11 (0xF8F00000 + 0x182C) (GIC)

Name	Bit Range	Type	Reset	Description
Undefined	31:26	N/A	N/A	Bits 31:26 have no effect.
IRQ ID #47	25:24	R/W	0x0	Targeted CPU(s) for interrupt ID#47 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	23:18	N/A	N/A	Bits 23:18 have no effect.
IRQ ID #46	17:16	R/W	0x0	Targeted CPU(s) for interrupt ID#46 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	15:10	N/A	N/A	Bits 15:10 have no effect.
IRQ ID #45	9:8	R/W	0x0	Targeted CPU(s) for interrupt ID#45 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	7:2	N/A	N/A	Bits 7:2 have no effect.
IRQ ID #44	1:0	R/W	0x0	Targeted CPU(s) for interrupt ID#44 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted

ICDIPTR13 (0xF8F00000 + 0x1834) (GIC)

Name	Bit Range	Type	Reset	Description
Undefined	31:26	N/A	N/A	Bits 31:26 have no effect.
IRQ ID #55	25:24	R/W	0x0	Targeted CPU(s) for interrupt ID#55 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	23:18	N/A	N/A	Bits 23:18 have no effect.
IRQ ID #54	17:16	R/W	0x0	Targeted CPU(s) for interrupt ID#54 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	15:10	N/A	N/A	Bits 15:10 have no effect.
IRQ ID #53	9:8	R/W	0x0	Targeted CPU(s) for interrupt ID#53 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	7:2	N/A	N/A	Bits 7:2 have no effect.
IRQ ID #52	1:0	R/W	0x0	Targeted CPU(s) for interrupt ID#52 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted

ICDIPTR14 (0xF8F00000 + 0x1838) (GIC)

Name	Bit Range	Type	Reset	Description
Undefined	31:26	N/A	N/A	Bits 31:26 have no effect.
IRQ ID #59	25:24	R/W	0x0	Targeted CPU(s) for interrupt ID#59 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	23:18	N/A	N/A	Bits 23:18 have no effect.
IRQ ID #58	17:16	R/W	0x0	Targeted CPU(s) for interrupt ID#58 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	15:10	N/A	N/A	Bits 15:10 have no effect.
IRQ ID #57	9:8	R/W	0x0	Targeted CPU(s) for interrupt ID#57 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	7:2	N/A	N/A	Bits 7:2 have no effect.
IRQ ID #56	1:0	R/W	0x0	Targeted CPU(s) for interrupt ID#56 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted

ICDIPTR17 (0xF8F00000 + 0x1844) (GIC)

Name	Bit Range	Type	Reset	Description
Undefined	31:26	N/A	N/A	Bits 31:26 have no effect.
IRQ ID #71	25:24	R/W	0x0	Targeted CPU(s) for interrupt ID#71 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	23:18	N/A	N/A	Bits 23:18 have no effect.
IRQ ID #70	17:16	R/W	0x0	Targeted CPU(s) for interrupt ID#70 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	15:10	N/A	N/A	Bits 15:10 have no effect.
IRQ ID #69	9:8	R/W	0x0	Targeted CPU(s) for interrupt ID#69 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	7:2	N/A	N/A	Bits 7:2 have no effect.
IRQ ID #68	1:0	R/W	0x0	Targeted CPU(s) for interrupt ID#68 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted

ICDIPTR20 (0xF8F00000 + 0x1850) (GIC)

Name	Bit Range	Type	Reset	Description
Undefined	31:26	N/A	N/A	Bits 31:26 have no effect.
IRQ ID #83	25:24	R/W	0x0	Targeted CPU(s) for interrupt ID#83 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	23:18	N/A	N/A	Bits 23:18 have no effect.
IRQ ID #82	17:16	R/W	0x0	Targeted CPU(s) for interrupt ID#82 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	15:10	N/A	N/A	Bits 15:10 have no effect.
IRQ ID #81	9:8	R/W	0x0	Targeted CPU(s) for interrupt ID#81 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted
Undefined	7:2	N/A	N/A	Bits 7:2 have no effect.
IRQ ID #80	1:0	R/W	0x0	Targeted CPU(s) for interrupt ID#80 00: no CPU targeted 01: CPU 0 targeted 10: CPU 1 targeted 11: CPU 0 and CPU 1 are both targeted

ICDICFR1 (0xF8F00000 + 0x1C04) (GIC)

Name	Bit Range	Type	Reset	Description
IRQ ID #31	31:30	R/W	0x1	Configuration for interrupt ID#3 (nIRQ) 01: Low-Level Active
IRQ ID #30	29:28	R/W	0x3	Configuration for interrupt ID#30 (CPU Watchdog Timer) 11: Edge Sensitive
IRQ ID #29	27:26	R/W	0x3	Configuration for interrupt ID#29 (CPU Private Timer) 11: Edge Sensitive
IRQ ID #28	25:24	R/W	0x1	Configuration for interrupt ID#28 (nFIQ) 01: Low-Level Active
IRQ ID #27	23:22	R/W	0x3	Configuration for interrupt ID#27 (Global Timer) 11: Edge Sensitive
Undefined	21:0	N/A	N/A	Bits 21:0 have no effect.

ICDICFR2 (0xF8F00000 + 0x1C08) (GIC)

Name	Bit Range	Type	Reset	Description
IRQ ID #47	31:30	R/W	0x1	Configuration for interrupt ID#47 01: high-level active 11: rising-edge The lower bit is read-only and is always 1.
.....	.....	.....	.....	.....
IRQ ID #44	25:24			Configuration for interrupt ID#44 01: high-level active 11: rising-edge The lower bit is read-only and is always 1
IRQ ID #43	23:22			Configuration for interrupt ID#43 01: high-level active 11: rising-edge The lower bit is read-only and is always 1
IRQ ID #42	21:20			Configuration for interrupt ID#42 01: high-level active 11: rising-edge The lower bit is read-only and is always 1
.....	.....	.....	.....	.....
IRQ ID #39	15:14	R/W	0x1	Configuration for interrupt ID#39 01: high-level active 11: rising-edge The lower bit is read-only and is always 1.
.....	.....	.....	.....	.....
IRQ ID #32	1:0	R/W	0x1	Configuration for interrupt ID#32 01: high-level active 11: rising-edge The lower bit is read-only and is always 1.

ICDICFR3 (0xF8F00000 + 0x1C0C) (GIC)

Name	Bit Range	Type	Reset	Description
IRQ ID #63	31:30	R/W	0x1	Configuration for interrupt ID#63 01: high-level active 11: rising-edge The lower bit is read-only and is always 1.
.....	.....	.....	.....	.....
IRQ ID #59	23:22	R/W	0x1	Configuration for interrupt ID#59 01: high-level active 11: rising-edge The lower bit is read-only and is always 1.
IRQ ID #58	21:20	R/W	0x1	Configuration for interrupt ID#58 01: high-level active 11: rising-edge The lower bit is read-only and is always 1.
.....	.....	.....	.....	.....
IRQ ID #52	9:8	R/W	0x1	Configuration for interrupt ID#52 01: high-level active 11: rising-edge The lower bit is read-only and is always 1.
.....	.....	.....	.....	.....
IRQ ID #48	1:0	R/W	0x1	Configuration for interrupt ID#48 01: high-level active 11: rising-edge The lower bit is read-only and is always 1.

ICDICFR4 (0xF8F00000 + 0x1C10) (GIC)

Name	Bit Range	Type	Reset	Description
IRQ ID #79	31:30	R/W	0x1	Configuration for interrupt ID#79 01: high-level active 11: rising-edge The lower bit is read-only and is always 1.
.....	.....	.....	.....	.....
IRQ ID #68	9:8	R/W	0x1	Configuration for interrupt ID#68 01: high-level active 11: rising-edge The lower bit is read-only and is always 1.
.....	.....	.....	.....	.....
IRQ ID #64	1:0	R/W	0x1	Configuration for interrupt ID#64 01: high-level active 11: rising-edge The lower bit is read-only and is always 1.

ICDICFR5 (0xF8F00000 + 0x1C14) (GIC)

Name	Bit Range	Type	Reset	Description
IRQ ID #95	31:30	R/W	0x1	Configuration for interrupt ID#79 01: high-level active 11: rising-edge The lower bit is read-only and is always 1.
.....	.....	.....	.....	.....
IRQ ID #68	9:8	R/W	0x1	Configuration for interrupt ID#68 01: high-level active 11: rising-edge The lower bit is read-only and is always 1.
.....	.....	.....	.....	.....
IRQ ID #82	3:2	R/W	0x1	Configuration for interrupt ID#82 01: high-level active 11: rising-edge The lower bit is read-only and is always 1.
IRQ ID #81	3:2	R/W	0x1	Configuration for interrupt ID#81 01: high-level active 11: rising-edge The lower bit is read-only and is always 1.
IRQ ID #80	1:0	R/W	0x1	Configuration for interrupt ID#80 01: high-level active 11: rising-edge The lower bit is read-only and is always 1.

Programming Examples in C

Disable ARM Cortex v-7 A9 Interrupts (GIC)

```
void disable_ARM_A9_interrupts(){
	uint32_t mode = 0xDF; // System mode [4:0] and IRQ disabled [7]
	uint32_t read_cpsr=0; // used to read previous CPSR value
	uint32_t bit_mask = 0xFF; // used to clear bottom 8 bits
	__asm__ __volatile__("mrs %0, cpsr\n"  : "=r" (read_cpsr) );
	__asm__ __volatile__("msr cpsr,%0\n" : : "r" ((read_cpsr & (~bit_mask))| mode));
	return;
}
```

Enable ARM Cortex v-7 A9 Interrupts (GIC)

```
void enable_ARM_A9_interrupts(){
	uint32_t read_cpsr=0; // used to read previous CPSR value
	uint32_t mode = 0x5F; // System mode [4:0] and IRQ enabled [7]
	uint32_t bit_mask = 0xFF; // used to clear bottom 8 bits
	__asm__ __volatile__("mrs %0, cpsr\n"  : "=r" (read_cpsr) );
	__asm__ __volatile__("msr cpsr,%0\n" : : "r" ((read_cpsr & (~bit_mask))| mode));
	return;
}
```

Configure GIC Example (GIC)

```
void configure_GIC(){
	*((uint32_t*) ICDIPTR_BASEADDR+0x34/4) = 0x00000000; // Step 1
	*((uint32_t*) ICDICER_BASEDDR+0x04/4) = 0x00100000; // Step 1
	*((uint32_t*) ICDDCR_BASEADDR) = 0x0; // Step 2
	*((uint32_t*) ICDIPR_BASEADDR+0x34/4) = 0x000000A0; // Step 3
	*((uint32_t*) ICDIPTR_BASEADDR+0x34/4) = 0x00000001; // Step 4
	*((uint32_t*) ICDICFR_BASEADDR+0xC/4) = 0x55555555; // Step 5
	*((uint32_t *) ICDISER_BASEADDR+0x4/4) = 0x00100000; // Step 6
	*((uint32_t*) ICCPMR_BASEADDR) = 0xFF; // Step 7
	*((uint32_t*) ICCICR_BASEADDR) = 0x3; // Step 8
	*((uint32_t*) ICDDCR_BASEADDR) = 0x1; // Step 9
	return;
}
```

Serving Interrupts (Exception Handler) (GIC)

When an exception happens, processor execution is forced to an address that corresponds to the type of exception. This address is called the exception vector for that exception and a total of 8 exceptions form the vector table that is represented in the following figure.

Offset	Vector tables			
	Hyp[a]	Monitor[b]	Secure	Non-secure
0x00	Not used	Not used	Reset	Not used
0x04	Undefined Instruction, from Hyp mode	Not used	Undefined Instruction	Undefined Instruction
0x08	Hypervisor Call, from Hyp mode	Secure Monitor Call	Supervisor Call	Supervisor Call
0x0C	Prefetch Abort, from Hyp mode	Prefetch Abort	Prefetch Abort	Prefetch Abort
0x10	Data Abort, from Hyp mode	Data Abort	Data Abort	Data Abort
0x14	Hyp Trap, or Hyp mode entry[c]	Not used	Not used	Not used
0x18	IRQ interrupt	IRQ interrupt	IRQ interrupt	IRQ interrupt
0x1C	FIQ interrupt	FIQ interrupt	FIQ interrupt	FIQ interrupt

a. Non-secure state only. Implemented only if the implementation includes the Virtualization Extensions.
b. Secure state only. Implemented only if the implementation includes the Security Extensions.
c. See *Use of offset 0x14 in the Hyp vector table* on page B1-1168.

Figure 2. ARM A9 Vector Table (DDI0406C Arm Architecture Reference Manual ARMv7-A and ARMv7-R edition Section B 1.8 on Page 1166) [2]

Before you're able to set up an exception handler or an interrupt handler to serve the interrupts you've configured, exception vector table initialization needs to take place, which is already done for you in asm_vectors.S file by Xilinx. Lines 68-79 within asm_vectors.S assembly file represent the vector table setup.

```
.globl _vector_table

.section .vectors
_vector_table:
```

```
        B       _boot
        B       Undefined
        B       SVCHandler
        B       PrefetchAbortHandler
        B       DataAbortHandler
        NOP     /* Placeholder for address exception vector*/
        B       IRQHandler
        B       FIQHandler
```

Now, every time an interrupt occurs the processor is forced to branch to IRQHandler function which can be found within asm_vectors.S lines 82-100 (found below). The IRQHandler saves processor register to stack and calls the routine "IRQInterrupt" in line 100.

```
IRQHandler:                             /* IRQ vector handler */

        stmdb  sp!,{r0-r3,r12,lr}       /* state save from compiled code*/
#ifdef __ARM_NEON__
        vpush {d0-d7}
        vpush {d16-d31}
        vmrs r1, FPSCR
        push {r1}
        vmrs r1, FPEXC
        push {r1}
#endif

#ifdef PROFILING
        ldr    r2, =prof_pc
        subs   r3, lr, #0
        str    r3, [r2]
#endif

        bl     IRQInterrupt             /* IRQ vector */
```

Now that the IRQInterrupt function has been called in vectors.c file every time an interrupt happens, it will call a structure `XExc_VectorTable` defined using XExc_VectorTableEntry struct.

```
void IRQInterrupt(void)
{
        XExc_VectorTable[XIL_EXCEPTION_ID_IRQ_INT].Handler(XExc_VectorTable[
                                XIL_EXCEPTION_ID_IRQ_INT].Data);
}

typedef struct {
        Xil_ExceptionHandler Handler;
        void *Data;
} XExc_VectorTableEntry;
```

Where Xil_ExceptionHandler is defined as

```
typedef void (*Xil_ExceptionHandler)(void *data);
```

and where XIL_EXCEPTION_ID_IRQ_INT is defined within `xil_exception.h` header file as

```
#define XIL_EXCEPTION_ID_RESET                  0U
#define XIL_EXCEPTION_ID_UNDEFINED_INT          1U
#define XIL_EXCEPTION_ID_SWI_INT        2U
#define XIL_EXCEPTION_ID_PREFETCH_ABORT_INT     3U
#define XIL_EXCEPTION_ID_DATA_ABORT_INT         4U
#define XIL_EXCEPTION_ID_IRQ_INT        5U
#define XIL_EXCEPTION_ID_FIQ_INT        6U
#define XIL_EXCEPTION_ID_LAST                   6U
```

All of this allows you to set up and interrupt handler named by your own choice. Setting up an interrupt handler can be done using Xilinx's built-in function that also requires `"xil_exception.h"` header to be included in your code:

Xil_ExceptionRegisterHandler(u32 Exception_ID, Xil_ExceptionHandler Handler, void *Data)

Where the function simply connects the name of the handler to the corresponding exception ID in `XExc_VectorTable`. This means that every time an interrupt occurs, **`IRQInterrupt`** is called that, which then calls the Handler corresponding to the Exception_ID. The exception ID can be chosen based on what type of interrupts you want to serve, i.e 0x5 corresponds to IRQ_ID, which is used in this example. The handler associated with this IRQ_ID can be named whatever you would prefer, but for this example it is named as IRQ_Handler. You should have this line of code in your main function to create the interrupt handler:

```
Xil_ExceptionRegisterHandler(5, IRQ_Handler, NULL);
```

You will then need to set up a function named IRQ_Handler to serve the interrupts with the following syntax:

```
void IRQ_Handler(void *data){

	/* Handle the interrupt here */
}
```

IRQ Handler Using MIO Buttons and LED (GIC)

```
void IRQ_Handler(void *data){
	uint32_t GPIO_INT = *((uint32_t*)GPIO_INT_STAT_1);
	uint32_t interrupt_ID = *((uint32_t*)ICCIAR_BASEADDR);
	uint32_t button_press = 0xC0000 & GPIO_INT;
	if (interrupt_ID == 52) { // check if interrupt is from the GPIO
		if (button_press == 0x80000){
			*((uint32_t*) GPIO_MTDATA_OUT_0) = 0xFFF80001;
		}
```

```
            else if (button_press == 0x40000){
                  *((uint32_t*) GPIO_MTDATA_OUT_0) = 0xFFF80002;
            }
      }
      *((uint32_t*)GPIO_INT_STAT_1) = 0xFFFFFF;
      *((uint32_t*)ICCEOIR_BASEADDR) = interrupt_ID;
}
```

General Purpose I/O (GPIO) Module [1]

Description

Xilinx's GPIO module is used to control the GPIO channels of the Zynq chip on Blackboard. For example, MIO pins can be configured as inputs or outputs, which can then be used to read a button press or set a color of an LED. This completely depends on your design, GPIO module also gives you the capability to set up interrupts for those specific pins. Next section, Overview of Registers, doesn't include all registers within the Xilinx's GPIO module, but it brings out some registers that can be used to set up Blackboard's MIO buttons and MIO RGB LED with interrupts. More registers can be found within the Zynq All Programmable SoC Technical Reference Manual.

Overview of Registers

Base Address + Offset (hex)	Register Name	Reset Value (hex)	Access Type	Description
0xE000A000 + 0x04	GPIO_MTDATA_OUT_0	0x0	Read/Write	Maskable Output Data Bank 0
0xE000A000 + 0x204	GPIO_DIRM_0	0x0	Read/Write	Direction Mode for Bank 0
0xE000A000 + 0x208	GPIO_OUTE_0	0x0	Read/Write	Output Enable for Bank 0
0xE000A000 + 0x214	GPIO_INT_DIS_0	0x0	Write Only	Interrupt Disable Bank 0
0xE000A000 + 0x244	GPIO_DIRM_1	0x0	Read/Write	Direction Mode for Bank 1
0xE000A000 + 0x250	GPIO_INT_EN_1	0x0	Write Only	Interrupt Enable Bank 1
0xE000A000 + 0x254	GPIO_INT_DIS_1	0x0	Write Only	Interrupt Disable Bank 1
0xE000A000 + 0x258	GPIO_INT_STAT_1	0x0	Read/Write	Interrupt Status Bank 1
0xE000A000 + 0x25C	GPIO_INT_TYPE_1	0x0	Read/Write	Interrupt Type Bank 1
0xE000A000 + 0x260	GPIO_INT_POL_1	0x0	Read/Write	Interrupt Polarity Bank 1
0xE000A000 + 0x264	GPIO_INT_ANY_1	0x0	Read/Write	Interrupt Sensitivity Bank 1

Registers

GPIO_MTDATA_OUT_0 (0xF8F00000 + 0x04) (GPIO)

Name	Bit Range	Type	Reset	Description
MASK_0_MSW	31:16	W	0x0	On a write, only bits with a corresponding de-asserted mask will change the output value. 0: pin value is updated 1: pin is masked Each bit controls the corresponding pin within MIO pins [31:16].
DATA_0_MSW	15:0	R/W	N/A	On a write, these are the data values for the corresponding GPIO output bits. Each bit controls the corresponding pin within the 16-bit half-bank. Reads return the previous value written to this register or DATA_0[15:0]. Corresponds to MIO pins [31:16]. Reads do not return the value on the GPIO pin

GPIO_DIRM_0 (0xE000A000 + 0x204) (GPIO)

Name	Bit Range	Type	Reset	Description
Direction Mode Bank 0	31:0	R/W	0x0	Direction mode 0: input 1: output Each bit configures the corresponding pin within bank 0, MIO pins [31:0]

GPIO_OUTE_0 (0xE000A000 + 0x208) (GPIO)

Name	Bit Range	Type	Reset	Description
Output Enable Bank 0	31:0	R/W	0x0	Output enables 0: disabled 1: enabled Each bit configures the corresponding pin within bank 0, MIO pins [31:0]

GPIO_INT_DIS_0 (0xE000A000 + 0x214) (GPIO)

Name	Bit Range	Type	Reset	Description
Interrupt Disable Bank 0	31:0	W	0x0	Interrupt disable 0: no change 1: clear interrupt mask (enable interrupt) Each bit configures the corresponding pin within bank 1, MIO pins [31:0]

GPIO_DIRM_1 (0xE000A000 + 0x244) (GPIO)

Name	Bit Range	Type	Reset	Description
Undefined	31:22	N/A	N/A	Bits 31:22 have no effect.
Direction Mode Bank 1	21:0	R/W	0x0	Direction mode 0: input 1: output Each bit configures the corresponding pin within bank 1, MIO pins [53:32]

GPIO_INT_EN_1 (0xE000A000 + 0x250) (GPIO)

Name	Bit Range	Type	Reset	Description
Undefined	31:22	N/A	N/A	Bits 31:22 have no effect.
Interrupt Enable Bank 1	21:0	W	0x0	Interrupt enable 0: no change 1: clear interrupt mask (enable interrupt) Each bit configures the corresponding pin within bank 1, MIO pins [53:32].

GPIO_INT_DIS_1 (0xE000A000 + 0x254) (GPIO)

Name	Bit Range	Type	Reset	Description
Undefined	31:22	N/A	N/A	Bits 31:22 have no effect.
Interrupt Disable Bank 1	21:0	W	0x0	Interrupt disable 0: no change 1: clear interrupt mask (enable interrupt) Each bit configures the corresponding pin within bank 1, MIO pins [53:32].

GPIO_INT_STAT_1 (0xE000A000 + 0x258) (GPIO)

Name	Bit Range	Type	Reset	Description
Undefined	31:22	N/A	N/A	Bits 31:22 have no effect.
Interrupt Status Bank 1	21:0	R/W	0x0	Interrupt status Upon read: 0: no interrupt 1: interrupt event has occurred Upon write: 0: no action 1: clear interrupt status bit Each bit configures the corresponding pin within the 22-bit bank 1 (MIO pins [53:32]).

GPIO_INT_TYPE_1 (0xE000A000 + 0x25C) (GPIO)

Name	Bit Range	Type	Reset	Description
Undefined	31:22	N/A	N/A	Bits 31:22 have no effect.
Interrupt Type Bank 1	21:0	R/W	0x0	Interrupt type 0: level-sensitive 1: edge-sensitive Each bit configures the corresponding pin within the 22-bit bank 1 (MIO pins [53:32]).

GPIO_INT_POL_1 (0xE000A000 + 0x260) (GPIO)

Name	Bit Range	Type	Reset	Description
Undefined	31:22	N/A	N/A	Bits 31:22 have no effect.
Interrupt Polarity Bank 1	21:0	R/W	0x0	Interrupt polarity 0: active low or falling edge 1: active high or rising edge Each bit configures the corresponding pin within the 22-bit bank 1 (MIO pins [53:32]).

GPIO_INT_ANY_1 (0xE000A000 + 0x264) (GPIO)

Name	Bit Range	Type	Reset	Description
Undefined	31:22	N/A	N/A	Bits 31:22 have no effect.
Interrupt Any Sensitive Bank 1	21:0	R/W	0x0	Interrupt edge triggering mode 0: trigger on single edge, using configured interrupt polarity 1: trigger on both edges Each bit configures the corresponding pin within the 22-bit bank 1 (MIO pins [53:32]).

Programming Examples in C

Initialize GPIO Pin (GPIO)

```
void Initialize_IOs(){
	*((uint32_t*) GPIO_DIRM_0) = 0x00070000;
	*((uint32_t*) GPIO_OUTE_0) = 0x00070000;
	*((uint32_t*) GPIO_DIRM_1) = 0x00000000;
	return;
}
```

Initialize GPIO Interrupts (GPIO)

```
void Initialize_GPIO_Interrupts(){
	// Step 1: Disable interrupts
	*((uint32_t*) GPIO_INT_DIS_1) = 0xFFFFFFFF;
	*((uint32_t*) GPIO_INT_DIS_0) = 0xFFFFFFFF;
	// Step 2:
	*((uint32_t*) GPIO_INT_STAT_1) = 0xFFFFFFFF; // Clear Status register
	// Step 3:
	*((uint32_t*) GPIO_INT_TYPE_1) = 0x000000; // Type of interrupt rising edge
	// Step 4:
	*((uint32_t*) GPIO_INT_POL_1) = 0x0C0000; // Polarity of interrupt
	// Step 5:
	*((uint32_t*) GPIO_INT_ANY_1) = 0x000000; // Interrupt any edge sensitivity
	// Step 6:
	*((uint32_t*) GPIO_INT_EN_1) = 0x0C0000; // Enable interrupts in bank 0
	*((uint32_t*) GPIO_MTDATA_OUT_0) = 0xFFF80000; // Set LEDs to 0
}
```

Serving GPIO Interrupts (GPIO)

Example of serving GPIO interrupts can be found here.

Global Timer Counter (GTC) Module [1]

Description

Xilinx's GTC module is deigned to control ARM Cortex-A9 MPCore 64-bit global timer with an auto-increment feature. More information about Cortex-A9 MPCore Global Timer can be found in Xilinx's Zynq SoC All Programmable reference manual under "Global Timer" Section.

Overview of Registers

Base Address + Offset (hex)	Register Name	Reset Value (hex)	Access Type	Description
0xF8F00000 + 0x200	Global Timer Counter Register 0	0x0	Read/Write	Global Timer Counter Register 0
0xF8F00000 + 0x204	Global Timer Counter Register 1	0x0	Read/Write	Global Timer Counter Register 1
0xF8F00000 + 0x208	Global Timer Control Register	0x0	Read/Write	Global Timer Control Register
0xF8F00000 + 0x20C	Global Timer Interrupt Status Register	0x0	Read/Write	Global Timer Interrupt Status Register
0xF8F00000 + 0x210	Comparator Value 0	0x0	Read/Write	Comparator Value Register 0
0xF8F00000 + 0x214	Comparator Value 1	0x0	Read/Write	Comparator Value Register 1
0xF8F00000 + 0x218	Auto Increment Register	0x0	Read/Write	Auto Increment Register

Registers

Global Timer Counter Register 0 (0xF8F00000 + 0x200) (GTC)

Name	Bit Range	Type	Reset	Description
Global Timer Counter Register 0	31:0	R/W	0x0	Lower 32 bits of the 64-bit Global Timer Counter. You must access these registers with 32-bit accesses. To modify the register: 1. Clear the timer enable bit in the Global Timer Control Register 2. Write lower 32-bit Global Timer Counter Register 0 3. Write upper 32-bit Global Timer Counter Register 1 4. Set Timer Enable To read value from the register: 1. Read the upper 32-bit Global Timer Counter Register 1 2. Read the lower 32-bit Global Timer Counter Register 0

Global Timer Counter Register 1 (0xF8F00000 + 0x204) (GTC)

Name	Bit Range	Type	Reset	Description
Global Timer Counter Register 1	31:0	R/W	0x0	Upper 32 bits of the 64-bit Global Timer Counter. You must access these registers with 32-bit accesses. To modify the register: 1. Clear the timer enable bit in the Global Timer Control Register 2. Write lower 32-bit Global Timer Counter Register 0 value 3. Write upper 32-bit Global Timer Counter Register 1 value 4. Set Timer Enable To read value from the register: 1. Read the upper 32-bit Global Timer Counter Register 1 2. Read the lower 32-bit Global Timer Counter Register 0

Global Timer Control Register (0xF8F00000 + 0x208) (GTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Prescaler	15:8	R/W	0x0	The prescaler modifies the clock period for the decrementing event for the Counter Register. The timer interval is calculated using the following equation: (PRESCALER_value+1)*(Load_value+1)*(CPU_3x2x PERIOD). CPU_3x2x frequency is 333.3333MHz (always half of CPU frequency).
Undefined	7:4	N/A	N/A	Bits 7:4 have no effect.
Mode	3:3	R/W	0x0	0 = When the counter reaches the comparator value, sets the event flag. It is the responsibility of software to update the comparator value to get further events. 1 = Each time the counter reaches the comparator value, the comparator register is incremented with the auto-increment register, so that further events can be set periodically without any software updates.
IRQ_Enable	2:2	R/W	0x0	0 = Interrupt disabled 1 = Interrupt enabled Interrupt ID for Global Timer is ID#27
Comparator Enable	1:1	R/W	0x0	Allows comparison between the 64-bit timer counter and the related 64-bit comparator register. 0 = Comparison disabled 1 = Comparison enabled
Timer Enable	0:0	R/W	0x0	0 = Timer Disabled 1 = Timer Enabled

Global Timer Interrupt Status Register (0xF8F00000 + 0x20C) (GTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:1	N/A	N/A	Bits 31:1 have no effect.
Interrupt Flag	0:0	R/W	0x0	The event flag is a sticky bit that is automatically set when the Counter Register reaches the Comparator Register value. If the timer interrupt is enabled, Interrupt ID 27 is set as pending in the Interrupt Distributor after the event flag is set. The event flag is cleared when written to 1.

Comparator Register 0 (0xF8F00000 + 0x210) (GTC)

Name	Bit Range	Type	Reset	Description
Global Timer Counter Register 0	31:0	R/W	0x0	Lower 32 bits of the 64-bit Global Timer Comparator. You must access these registers with 32-bit accesses. To modify the register: 1. Clear the compare enable bit in the timer control register 2. Write the lower 32-bit Global Timer Comparator value 3. Write upper 32-bit Global Timer Comparator value 4. Set comparator enable bit and the interrupt enable bit if necessary.

Comparator Register 1 (0xF8F00000 + 0x214) (GTC)

Name	Bit Range	Type	Reset	Description
Global Timer Counter Register 1	31:0	R/W	0x0	Upper 32 bits of the 64-bit Global Timer Comparator. You must access these registers with 32-bit accesses. To modify the register: 1. Clear the compare enable bit in the timer control register 2. Write the lower 32-bit Global Timer Comparator value 3. Write upper 32-bit Global Timer Comparator value Set comparator enable bit and the interrupt enable bit if necessary. To read value from the register: 1. Read the upper 32-bit Global Timer Counter Register 1 2. Read the lower 32-bit Global Timer Counter Register 0

Auto Increment Register (0xF8F00000 + 0x218) (GTC)

Name	Bit Range	Type	Reset	Description
Auto Increment Value	31:0	R/W	0x0	Auto-increment Register This 32-bit register gives the increment value of the Comparator Register when the Auto-increment bit is set in the Timer Control Register. If the comparator enable and auto-increment bits are set when the global counter reaches the Comparator Register value, the comparator is incremented by the auto-increment value, so that a new event can be set periodically. The global timer is not affected and goes on incrementing.

H-Bridge Module

Description

The H-Bridge IP block is a single-channel controller designed specifically for Digilent's HB5 Pmod. This IP has a total of 5 registers that can be used to control the H bridge. Figure below represents a block diagram of inputs and outputs of how this IP is functioning. An overview and complete description of all the registers within this IP is presented in the following sections. Also, an example of how to program/configure this IP is presented as well.

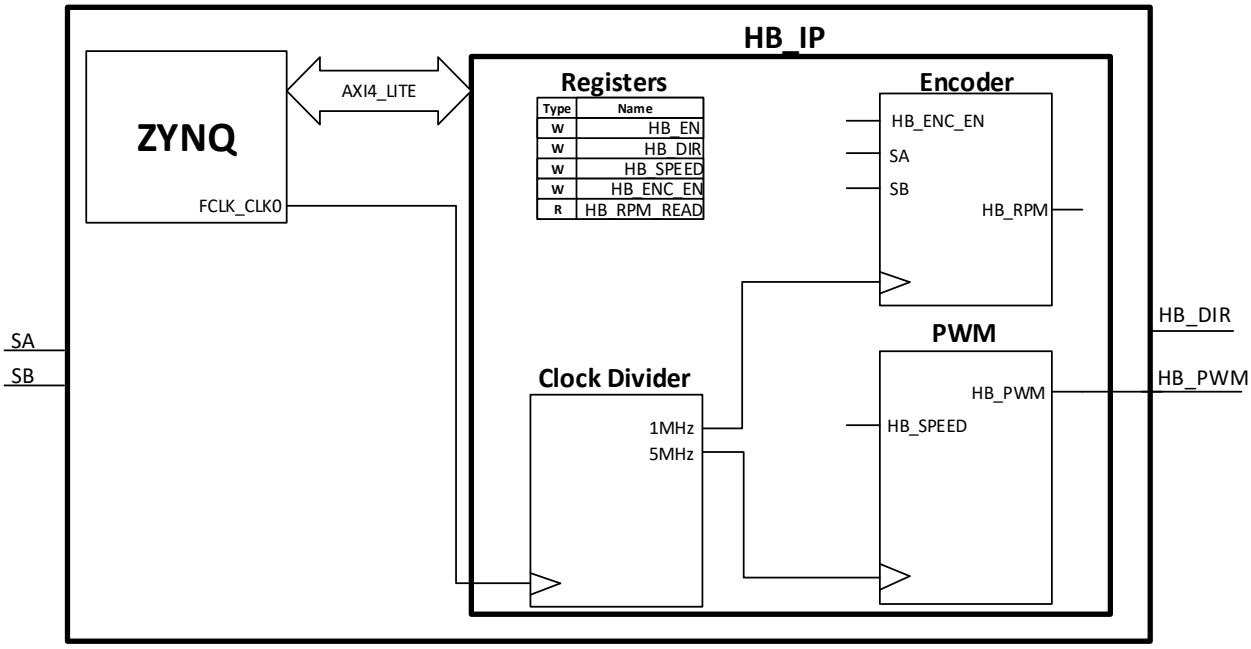

Figure 3 H-Bridge block diagram

Overview of Registers

Base Address + Offset (hex)	Register Name	Reset Value (hex)	Access Type	Description
0x4BB05000 + 0x00	HB_EN	0x0	Read/Write	Enable H-Bridge
0x4BB05000 + 0x04	HB_DIR	0x0	Read/Write	Direction of rotation
0x4BB05000 + 0x08	HB_SPEED	0x0	Read/Write	Adjust the speed of the motor
0x4BB05000 + 0x0C	HB_ENC_EN	0x0	Read/Write	Enable encoder for RPM readings
0x4BB05000 + 0x10	HB_RPM_READ	N/A	Read	Read the raw RPM value

Registers

HB_EN (0x4BB05000 + 0x00) (HB)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:1 have no effect.
H-Bridge Enable	0:0	R/W	0x0	This bit is used to enable the H-Bridge system. Writing a 1 to this bit enables the H-Bridge.

HB_DIR (0x4BB05000 + 0x04) (HB)

Name	Bit Range	Type	Reset	Description
Undefined	31:1	N/A	N/A	Bits 31:1 have no effect.
Motor Rotation Direction	0:0	R/W	0x0	This bit is used choose the direction of rotation. Writing 0 or 1 to this register corresponds to clockwise or counter clockwise rotation of the motor, respectively.

HB_SPEED (0x4BB05000 + 0x08) (HB)

Name	Bit Range	Type	Reset	Description
Undefined	31:10	N/A	N/A	Bits 31:10 have no effect.
Motor Speed Control	9:0	R/W	0x0	Writing into these bits will define the PWM duty cycle of the motor; 0x3FF being the maximum value and 0x000 the lowest value.

HB_ENC_EN (0x4BB05000 + 0x0C) (HB)

Name	Bit Range	Type	Reset	Description
Undefined	31:1	N/A	N/A	Bits 31:1 have no effect.
RPM Enable	0:0	R/W	0x0	This bit is used to choose to enable/start a RPM reading while the HB_EN is high. Writing a 1 to this enables and starts the RPM reading.

HB_RPM_READ (0x4BB05000 + 0x10) (HB)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Read raw RPM value	15:0	R	0x0	This is a read only register; writes have no effect. Reading from this register returns a half word that represents the raw value of RPM. Raw RPM value can be used to calculate the final RPM value using the following equation: $RPM = \left(\frac{1MHz}{RPM_{raw} * GearRatio}\right) * \frac{60s}{min}$

Programming Example in C (HB)

```
#include <stdio.h>
#define H_BRIDGE_BASEADDR 0x4BB05000 //Define HB_IP base address

int final_rpm(int gear_ratio);

int main() {

    // Declare volatile integer i and rpm
    volatile int i = 0;
    volatile int rpm = 0;
    *((uint32_t *) H_BRIDGE_BASEADDR) = 0x01; //Enable HB_IP
    *((uint32_t *) H_BRIDGE_BASEADDR+0x01) = 0x1; // Set DC motor direction bit as 1
    *((uint32_t *) H_BRIDGE_BASEADDR+0x02) = 0x15F; // Set speed value as 0x2FF

    for (i = 0; i < 200000000; i++); // Delay. Wait until motor stabilizes before getting
the RPM value
    rpm = final_rpm(53); // gear ratio is 53:1

    printf("Final RPM value is %d\n", rpm);

    return 1;
}

int final_rpm(int gear_ratio){
       volatile int i = 0;
       volatile int rpm_raw = 0; // Store raw RPM value here
       volatile int rpm = 0; // Store final RPM value here

       /*
        * As soon as the encoder is enabled, a raw RPM value is generated,
        * to read another raw RPM value, the encoder enable bit
        * must be set to 0 and then 1 again
       */
       *((uint32_t *) H_BRIDGE_BASEADDR+0x03) = 0x1; // Enable Encoder to get RPM value

       for (i = 0; i < 10000000; i++); // short delay to wait until value has been
                                       // sto in read register

       rpm_raw = *((uint32_t *) H_BRIDGE_BASEADDR+0x04); // Read the Raw RPM value
       rpm = 60*1000000/(rpm_raw*gear_ratio); // calculate final RPM value
       *((uint32_t *) H_BRIDGE_BASEADDR+0x03) = 0x0; // Disable Encoder

       return rpm;
}
```

iNEMO Inertial Module (NAV Sensor) [3]

Description

SPI Accelerometer, Gyroscope and Magnetometer module is configured to read and write data into 9-axis navigation sensor LSM9DS1. LSM9DS1 is a system-in-package (SiP) featuring a 3-axis digital acceleration sensor, a 3-axis digital angular rate sensor, and a 3-axis digital magnetic sensor. Each axis has 16-bit full scale registers, which enable the user to figure out which direction they are facing, if they are tilted, or what is the magnetic field around them.

You can configure the SiP to have the following features:

- ±2/±4/±8/±16 g linear acceleration full scale
- ±4/±8/±12/±16 gauss magnetic full scale
- ±245/±500/±2000 dps angular rate full scale

The module can be accessed using Xilinx's SPI0 Controller Module.

SPI Read (Accelerometer/Gyroscope) (NAV)

The SPI read is executed with 16 clock pulses (Figure 4). Also, multiple byte read command can be performed by adding 8 clock pulses (Figure 5). "When the CTRL_REG8 (22h) (IF_ADD_INC) bit is '0' the address used to read/write data remains the same for every block. When the CTRL_REG8 (22h) (IF_ADD_INC) bit is '1', the address used to read/write data is increased at every block." [3]

- Bit 0 = 1 (READ bit)
- Bits 1-7 = address AD(6:0). Address of the register.
- Bits 8-15: data DO(7:0) (read mode). Data read from the device (Most significant bit of the byte first)
- Bits 16-…: data DO(…-8). Further data

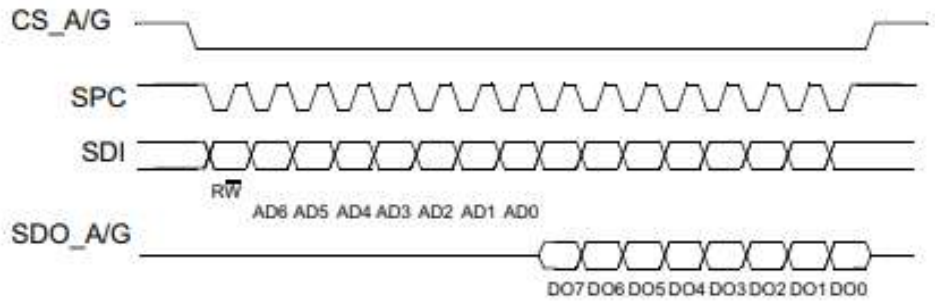

Figure 4. SPI single byte read (Accelerometer/Gyroscope) [3]

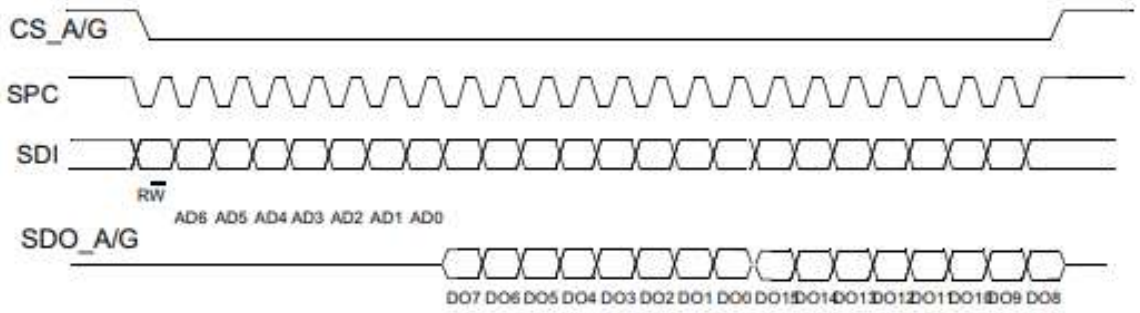

Figure 5. Multiple byte SPI read (2 Bytes, Accelerometer/Gyroscope) [3]

SPI Write (Accelerometer/Gyroscope) (NAV)

The SPI write is executed with 16 clock pulses (Figure 6). Also, multiple byte write command can be performed by adding 8 clock pulses (Figure 7).

- Bit 0 = 0 (WRITE bit)
- Bits 1-7 = address AD(6:0). Address of the register.
- Bits 8-15: data DO(7:0) (write mode). Data written to the sensor registers (Most significant bit of the byte first).
- Bits 16-…: data DO(…-8). Further data.

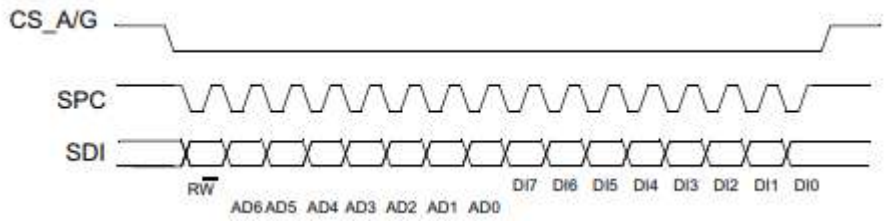

Figure 6. SPI single byte write (Accelerometer/Gyroscope) [3]

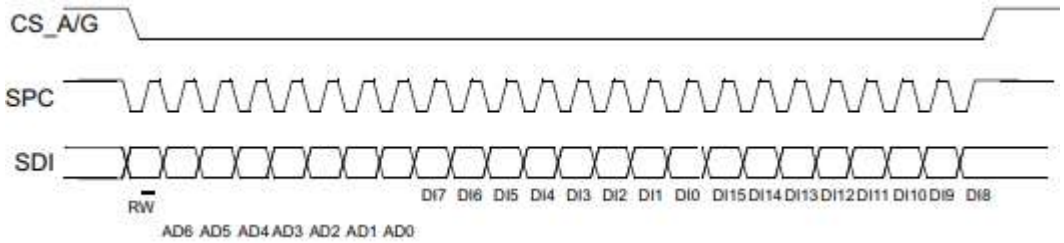

Figure 7. Multiple byte SPI write (2 Bytes, Accelerometer/Gyroscope) [3]

SPI Read (Magnetometer) (NAV)

Like SPI read execution for Accelerometer/Gyroscope, the command for Magnetometer is performed with 16 clock pulses (figure 8). A multiple byte read command can be performed by adding an additional of 8 clock pulses (figure 9).

- Bit 0 = 1 (READ bit)
- Bit 1 = $\overline{MS}$ bit. If this bit = 0, address is not incremented; when bit = 1, address is incremented in multiple byte reads.
- Bits 2-7 = address AD(6:0). Address of the register.
- Bits 8-15: data DO(7:0) (read mode). Data read from the device (Most significant bit of the byte first)
- Bits 16-…: data DO(…-8). Further data

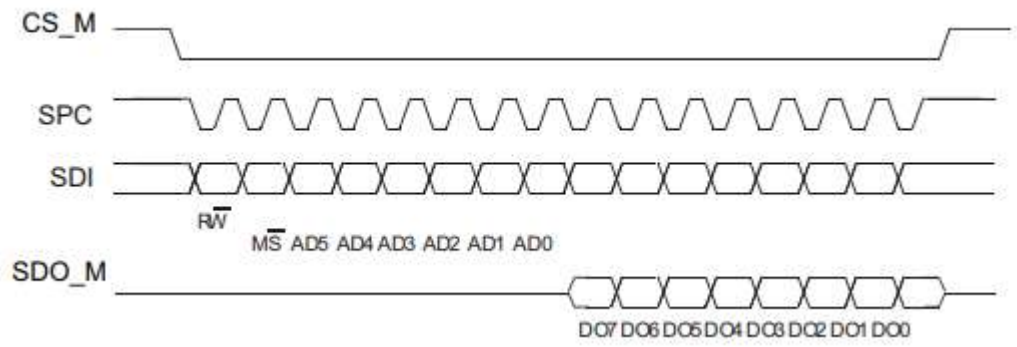

Figure 8. SPI single byte read (Magnetometer) [3]

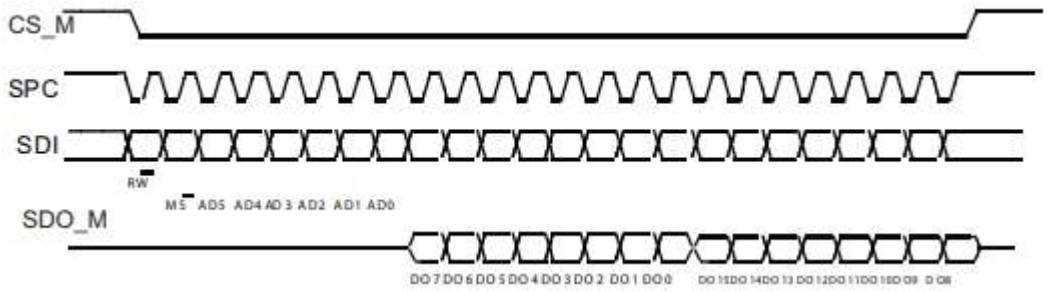

Figure 9. SPI multiple byte read (2 bytes, Magnetometer) [3]

SPI Write (Magnetometer) (NAV)

SPI write command for Magnetometer is performed with 16 clock pulses (figure 10). A multiple byte write execution can be performed by adding an additional of 8 clock pulses (figure 11).

- Bit 0 = 0 (WRITE bit)
- Bit 1 = $\overline{MS}$ bit. If this bit = 0, address is not incremented; when bit = 1, address is incremented in multiple byte reads.
- Bits 2-7 = address AD(6:0). Address of the register.
- Bits 8-15: data DO(7:0) (write mode). Data written to the sensor registers (Most significant bit of the byte first).
- Bits 16-…: data DO(…-8). Further data.

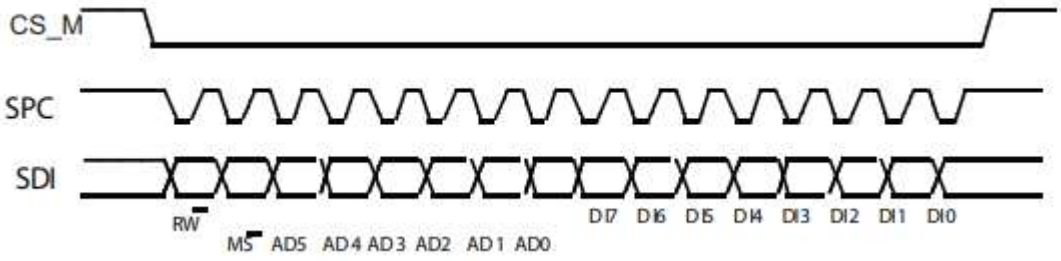

Figure 10. SPI single byte write (Magnetometer) [3]

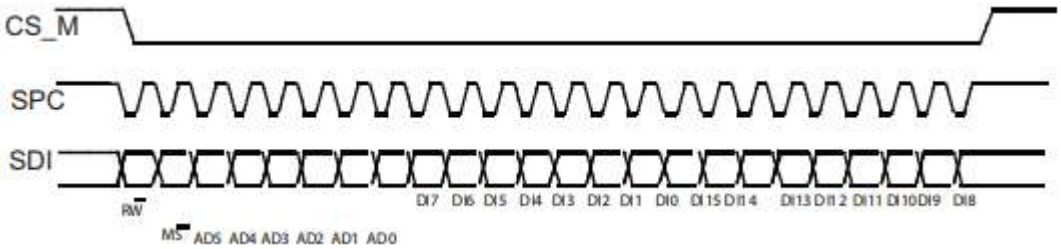

Figure 11. Multiple byte SPI write (2 Bytes, Magnetometer) [3]

Registers: Accelerometer and Gyroscope (CS Line 0) [3] (NAV)

Gyroscope (Angular Rate Sensor) (NAV)

Register Address (hex)	Register Name	Reset Value (hex)	Access Type	Description
0x0F	WHO_AM_I	0x68	Read	Default ID value of Accelerometer and Gyroscope module.
0x10	CTRL_REG1_G	0x10	Read/Write	Activates Accelerometer + Gyroscope when default value of register has been changed based on one of these options: - 245 dps register value: 0x20 (hex) - 500 dps register value: 0x28 (hex) - 2000 dps register value: 0x38 (hex)
0x15	OUT_TEMP_L	Output	Read	Temperature data output register. L and H registers together express a 16-bit word in two's complement. Total temperature value is represented in 12 bits [11:0] where bit 11 is the sign bit. Bits [15:12] are sign extend of bit 11.
0x16	OUT_TEMP_H	Output	Read	
0x18	OUT_X_L_G	Output	Read	Angular rate sensor X-axis angular rate output register. The value is expressed as 16-bit word in two's complement. OUT_X_L_G represents the bottom 8 bits [7:0] of the half word and OUT_X_H_G represent the top 8 bits [15:8] of the half-word.
0x19	OUT_X_H_G	Output	Read	
0x1A	OUT_Y_L_G	Output	Read	Angular rate sensor Y-axis angular rate output register. The value is expressed as 16-bit word in two's complement. OUT_Y_L_G represents the bottom 8 bits [7:0] of the half word and OUT_Y_H_G represent the top 8 bits [15:8] of the half-word.
0x1B	OUT_Y_H_G	Output	Read	
0x1C	OUT_Z_L_G	Output	Read	Angular rate sensor Z-axis angular rate output register. The value is expressed as 16-bit word in two's complement. OUT_Z_L_G represents the bottom 8 bits [7:0] of the half word and OUT_Z_H_G represent the top 8 bits [15:8] of the half-word.
0x1D	OUT_Z_H_G	Output	Read	

Accelerometer (Linear Acceleration Sensor) (NAV)

Register Address (hex)	Register Name	Reset Value (hex)	Access Type	Description
0x20	CTRL_REG6_XL	0x20	Read/Write	Activates only Accelerometer sensor when default value of register has been changed to one of these options: ±2g register value: 0x20 (hex) ±4g register value: 0x30 (hex) ±8g register value: 0x38 (hex) ±16g register value: 0x28 (hex).
0x22	CTRL_REG8	0x04	Read/Write	Control Register 8. By default, register address is automatically incremented during multiple byte access with a serial interface. Write 0x00 into this register to disable register automatic incrementation.
0x28	OUT_X_L_XL	Output	Read	Linear acceleration sensor X-axis output register. The value is expressed as 16-bit word in two's complement. OUT_X_L_XL represents the bottom 8 bits [7:0] of the half word and OUT_X_H_XL represent the top 8 bits [15:8] of the half-word.
0x29	OUT_X_H_XL	Output	Read	
0x2A	OUT_Y_L_XL	Output	Read	Linear acceleration sensor Y-axis output register. The value is expressed as 16-bit word in two's complement. OUT_Y_L_XL represents the bottom 8 bits [7:0] of the half word and OUT_Y_H_XL represent the top 8 bits [15:8] of the half-word.
0x2B	OUT_Y_H_XL	Output	Read	
0x2C	OUT_Z_L_XL	Output	Read	Linear acceleration sensor Z-axis output register. The value is expressed as 16-bit word in two's complement. OUT_Z_L_XL represents the bottom 8 bits [7:0] of the half word and OUT_Z_H_XL represent the top 8 bits [15:8] of the half-word.
0x2D	OUT_Z_H_XL	Output	Read	

Registers: Magnetometer (CS Line 1) [3] (NAV)

Register Address (hex)	Register Name	Reset Value (hex)	Access Type	Description
0x05	OFFSET_X_L_M	0x00	Read/Write	This register is a 16-bit register (L and H together) and represents the X offset used to compensate environmental effects (data as two's complement). This value acts on the magnetic output data value to subtract the environmental offset.
0x06	OFFSET_X_H_M	0x00	Read/Write	
0x07	OFFSET_Y_L_M	0x00	Read/Write	This register is a 16-bit register (L and H together) and represents the Y offset used to compensate environmental effects (data as two's complement). This value acts on the magnetic output data value to subtract the environmental offset.
0x08	OFFSET_Y_H_M	0x00	Read/Write	
0x09	OFFSET_Z_L_M	0x00	Read/Write	This register is a 16-bit register (L and H together) and represents the Z offset used to compensate environmental effects (data as two's complement). This value acts on the magnetic output data value to subtract the environmental offset.
0x0A	OFFSET_Z_H_M	0x00	Read/Write	
0x0F	WHO_AM_I	0x3D	Read	Default ID value of Magnetometer module.
0x21	CTRL_REG2_M	0x00	Read/Write	Magnetometer Full Scale Selection ±4gauss register value: 0x00 (hex) ±8gauss register value: 0x20 (hex) ±12gauss register value: 0x40 (hex) ±16gauss register value: 0x60 (hex).
0x22	CTRL_REG3_M	0x03	Read/Write	Activates Magnetometer with continuous conversion mode when default value of register has been changed to 00 (hex)
0x28	OUT_X_L_M	Output	Read	Magnetic sensor X-axis output register. The value is expressed as 16-bit word in two's complement. OUT_X_L_M represents the bottom 8 bits [7:0] of the half word and OUT_X_H_M represent the top 8 bits [15:8] of the half-word.
0x29	OUT_X_H_M	Output	Read	
0x2A	OUT_Y_L_M	Output	Read	Magnetic sensor Y-axis output register. The value is expressed as 16-bit word in two's complement. OUT_Y_L_M represents the bottom 8 bits [7:0] of the half word and OUT_Y_H_M represent the top 8 bits [15:8] of the half-word.
0x2B	OUT_Y_H_M	Output	Read	
0x2C	OUT_Z_L_M	Output	Read	Magnetic sensor Z-axis output register. The value is expressed as 16-bit word in two's complement. OUT_Z_L_M represents the bottom 8 bits [7:0] of the half word and OUT_Z_H_M represent the top 8 bits [15:8] of the half-word.
0x2D	OUT_Z_H_M	Output	Read	

I2C Controller Modules (I2C0 and I2C1) [1]

Description

Real Digital's Blackboard uses I2C1 Module through MIO pins (20 and 21) that is connected to the on-board Temperature Sensor (LM75BDP by NXP). You can find more information about the registers within NXP Temperature Sensor section. I2C0 is not used by default on the blackboard.hdf file and isn't connected to any peripherals on Blackboard.

Overview of Registers

Base Address + Offset (hex)	Register Name	Reset Value (hex)	Access Type	Description
I2C0: 0xE0004000 + 0x00 I2C1: 0xE0005000 + 0x00	I2C_CTRL	0x00000	Read/Write	I2C Control register
I2C0: 0xE0004000 + 0x04 I2C1: 0xE0005000 + 0x04	I2C_Status	0x00004	Read	Status register
I2C0: 0xE0004000 + 0x08 I2C1: 0xE0005000 + 0x08	I2C_ADDR	0x00000	Read/Write	I2C Address register
I2C0: 0xE0004000 + 0x0C I2C1: 0xE0005000 + 0x0C	I2C_DATA	0x00000	Read/Write	I2C Data register
I2C0: 0xE0004000 + 0x10 I2C1: 0xE0005000 + 0x10	I2C_ISR	0x00000	Read/Write	I2C interrupt status register
I2C0: 0xE0004000 + 0x14 I2C1: 0xE0005000 + 0x14	I2C_T_Size	0x00000	Read/Write	Transfer Size Register
I2C0: 0xE0004000 + 0x18 I2C1: 0xE0005000 + 0x18	I2C_SMPR	0x00000	Read/Write	Slave Monitor Pause Register
I2C0: 0xE0004000 + 0x1C I2C1: 0xE0005000 + 0x1C	I2C_Time_Out	0x00000	Read/Write	Time out register
I2C0: 0xE0004000 + 0x20 I2C1: 0xE0005000 + 0x20	I2C_IMR	0x00000	Read	Interrupt Mask register
I2C0: 0xE0004000 + 0x24 I2C1: 0xE0005000 + 0x24	I2C_IER	0x000FF	Read/Write	Interrupt Enable Register
I2C0: 0xE0004000 + 0x28 I2C1: 0xE0005000 + 0x28	I2C_IDR	0x00001	Read/Write	Interrupt Disable Register

Registers

I2C_CTRL (I2C0: 0xE0004000 + 0x00; I2C1: 0xE0005000 + 0x00) (I2C)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
DIV_A	15:14	R/W	0x1	Divisor for stage A clock divider. Bits = 0 - 3: Divides the input pclk frequency by divisor_a + 1.
DIV_B	13:8	W	0x0	Divisor for stage B clock divider. Bits = 0 - 63: Divides the output frequency from divisor_a by divisor_b + 1.
Undefined	7:7	R/W	0x0	Bits 7:7 have no effect.
Clear FIFO	6:6	R/W	0x0	Bit = 0: No Effect Bit = 1: initializes the FIFO to all zeros and clears the transfer size register except in master receive mode. Automatically gets cleared on the next APB clock after being set.
Slave Monitor Mode	5:5	R/W	0x0	Bit = 0: normal operation. Bit = 1: monitor mode.
Hold Bus	4:4	R/W	0x0	Bit = 0: allow the transfer to terminate as soon as all the data has been transmitted or received. Bit = 1: when no more data is available for transmit or no more data can be received, hold the sclk line low until serviced by the host.
ACKEN	3:3	R/W	0x0	Bit = 0: acknowledge disabled, NACK transmitted. Bit = 1: acknowledge enabled, ACK transmitted.
Addressing Mode	2:2	R/W	0x0	Bit = 0: reserved Bit = 1: normal (7-bit) address
Overall Interface Mode	1:1	R/W	0x0	Bit = 0: slave Bit = 1: master
Direction of Transfer	0:0	R/W	0x0	Bit = 0: master transmitter. Bit = 1: master receiver

I2C_Status (I2C0: 0xE0004000 + 0x04; I2C1: 0xE0005000 + 0x04) (I2C)

Name	Bit Range	Type	Reset	Description
Undefined	31:9	N/A	N/A	Bits 31:9 have no effect.
Bus Active	8:8	R	0x0	Bit = 0: no ongoing transfer Bit = 1: ongoing transfer on the I2C bus
Receiver Overflow	7:7	R	0x0	Bit = 0: No Overflow Bit = 1: This bit is set whenever FIFO is full, and a new byte is received. The new byte is not acknowledged, and contents of the FIFO remains unchanged.
Transmit Data Valid	6:6	R	0x0	SW should not use this to determine data completion, it is the RAW value on the interface. Bit = 0: no bytes to transfer Bit = 1: still a byte of data to be transmitted by the interface.
Receiver Data Valid	5:5	R	0x1	Bit = 0: no new data Bit = 1: valid, new data to be read from the interface.
Undefined	4:4	N/A	N/A	Bits 4:4 have no effect
RX read_write	3:3	R	0x0	Bit = 0: no mode of the transmission received from a master Bit = 1: mode of the transmission received from a master
Undefined	2:0	R	0x0	Bits 2:0 have no effect.

I2C_ADDR (I2C0: 0xE0004000 + 0x08; I2C1: 0xE0005000 + 0x08) (I2C)

Name	Bit Range	Type	Reset	Description
Undefined	31:10	N/A	N/A	Bits 31:10 have no effect.
Address	6:0 (NAM) 9:0 (EAM)	R/W	0x0	Bits = 0 - 1024: Normal addressing mode uses bits[6:0]. Extended addressing mode uses bits[9:0].

I2C_DATA (I2C0: 0xE0004000 + 0x0C; I2C1: 0xE0005000 + 0x0C) (I2C)

Name	Bit Range	Type	Reset	Description
Undefined	31:8	N/A	N/A	Bits 31:8 have no effect.
Data	7:0	R/W	0x0	Bits = 0 -255: When written to, the data register sets data to transmit. When read from, the data register reads the last received byte of data.

I2C_ISR (I2C0: 0xE0004000 + 0x10; I2C1: 0xE0005000 + 0x10) (I2C)

Name	Bit Range	Type	Reset	Description
Undefined	31:10	N/A	N/A	Bits 31:10 have no effect.
Arbitration Lost	9:9	Write to Clear	0x0	Bit = 1: master loses bus ownership during a transfer due to ongoing arbitration
Undefined	8:8	N/A	0x0	Bits 8:8 have no effect.
FIFO Receive Underflow	7:7	Write to Clear	0x0	Bit = 1: host attempts to read from the I2C data register more times than the value of the transfer size register plus one
FIFO Transmit Overflow	6:6	Write to Clear	0x0	Bit = 1: host attempts to write to the I2C data register more times than the FIFO depth
Receive Overflow	5:5	Write to Clear	0x0	Bit = 1: This bit is set whenever FIFO is full, and a new byte is received. The new byte is not acknowledged, and contents of the FIFO remains unchanged.
Monitored Slave Ready	4:4	Write to Clear	0x0	Bit = 1: addressed slave returns ACK.
Transfer Time Out	3:3	Write to Clear		Bit = 1: I2C sclk line is kept low for longer time
Transfer not Acknowledged	2:2	Write to Clear	0x0	Bit = 1: slave responds with a NACK or master terminates the transfer before all data is supplied
More Data	1:1	Write to Clear	0x0	Bit = 1: Data being sent or received.
Transfer Complete	0:0	Write to Clear	0x0	Bit = 1: Transfer is complete

I2C_T_Size (I2C0: 0xE0004000 + 0x14; I2C1: 0xE0005000 + 0x14) (I2C)

Name	Bit Range	Type	Reset	Description
Undefined	31:8	N/A	N/A	Bits 31:8 have no effect.
Transfer Size	7:0	R/W	0x0	Bits = 0 -255: - Master transmitter mode: number of data bytes still not transmitted minus one - Master receiver mode: number of data bytes that are still expected to be received - Slave transmitter mode: number of bytes remaining in the FIFO after the master terminates the transfer - Slave receiver mode: number of valid data bytes in the FIFO

I2C_SMPR (I2C0: 0xE0004000 + 0x18; I2C1: 0xE0005000 + 0x18) (I2C)

Name	Bit Range	Type	Reset	Description
Undefined	31:4	N/A	N/A	Bits 31:4 have no effect.
Pause Interval	3:0	R/W	0x0	Bits = 0 - 7: pause interval

I2C_Time_Out (I2C0: 0xE0004000 + 0x1C; I2C1: 0xE0005000 + 0x1C) (I2C)

Name	Bit Range	Type	Reset	Description
Undefined	31:8	N/A	N/A	Bits 31:8 have no effect.
Time Out	7:0	R/W	0x1F	Bits = 255 - 31: value of time out register

I2C_IMR (I2C0: 0xE0004000 + 0x20; I2C1: 0xE0005000 + 0x20) (I2C)

Name	Bit Range	Type	Reset	Description
Undefined	31:10	N/A	N/A	Bits 31:10 have no effect.
Arbitration Lost	9:9	R	0x0	Bit = 0: unmask this interrupt Bit = 1: Mask this interrupt
Undefined	8:8	N/A	0x0	Bits 8:8 have no effect.
FIFO Receive Underflow	7:7	R	0x0	Bit = 0: unmask this interrupt Bit = 1: Mask this interrupt
FIFO Transmit Overflow	6:6	R	0x0	Bit = 0: unmask this interrupt Bit = 1: Mask this interrupt
Receive Overflow	5:5	R	0x0	Bit = 0: unmask this interrupt Bit = 1: Mask this interrupt
Monitored Slave Ready	4:4	R	0x0	Bit = 0: unmask this interrupt Bit = 1: Mask this interrupt
Transfer Time Out	3:3	R	0x0	Bit = 0: unmask this interrupt Bit = 1: Mask this interrupt
Transfer not Acknowledged	2:2	R	0x0	Bit = 0: unmask this interrupt Bit = 1: Mask this interrupt
More Data	1:1	R	0x0	Bit = 0: unmask this interrupt Bit = 1: Mask this interrupt
Transfer Complete	0:0	R	0x0	Bit = 0: unmask this interrupt Bit = 1: Mask this interrupt

I2C_IER (I2C0: 0xE0004000 + 0x24; I2C1: 0xE0005000 + 0x24) (I2C)

Name	Bit Range	Type	Reset	Description
Undefined	31:10	N/A	N/A	Bits 31:10 have no effect.
Arbitration Lost	9:9	W	0x0	Bit = 1: enable this interrupt
Undefined	8:8	N/A	0x0	Bits 8:8 have no effect.
FIFO Receive Underflow	7:7	W	0x0	Bit = 1: enable this interrupt
FIFO Transmit Overflow	6:6	W	0x0	Bit = 1: enable this interrupt
Receive Overflow	5:5	W	0x0	Bit = 1: enable this interrupt
Monitored Slave Ready	4:4	W	0x0	Bit = 1: enable this interrupt
Transfer Time Out	3:3	W	0x0	Bit = 1: enable this interrupt
Transfer not Acknowledged	2:2	W	0x0	Bit = 1: enable this interrupt
More Data	1:1	W	0x0	Bit = 1: enable this interrupt
Transfer Complete	0:0	W	0x0	Bit = 1: enable this interrupt

I2C_IDR (I2C0: 0xE0004000 + 0x28; I2C1: 0xE0005000 + 0x28) (I2C)

Name	Bit Range	Type	Reset	Description
Undefined	31:10	N/A	N/A	Bits 31:10 have no effect.
Arbitration Lost	9:9	W	0x0	Bit = 1: disable this interrupt
Undefined	8:8	N/A	0x0	Bits 8:8 have no effect.
FIFO Receive Underflow	7:7	W	0x0	Bit = 1: disable this interrupt
FIFO Transmit Overflow	6:6	W	0x0	Bit = 1: disable this interrupt
Receive Overflow	5:5	W	0x0	Bit = 1: disable this interrupt
Monitored Slave Ready	4:4	W	0x0	Bit = 1: disable this interrupt
Transfer Time Out	3:3	W	0x0	Bit = 1: disable this interrupt
Transfer not Acknowledged	2:2	W	0x0	Bit = 1: disable this interrupt
More Data	1:1	W	0x0	Bit = 1: disable this interrupt
Transfer Complete	0:0	W	0x0	Bit = 1: disable this interrupt

LED Module

Description

Blackboard has 4 individual LED's that can be configured to function in two different modes: default mode (DM) and PWM mode (PM). In PWM mode, users have control over both PWM window frequency and duty cycle. This section introduces all registers within the LED IP and how to use these in C coding.

Overview of Registers

Base Address + Offset (hex)	Register Name	Reset Value (hex)	Access Type	Description
0x4BB00000 + 0x00	LED_CTRL	0x0	Read/Write	Enable LEDs. Select between custom and default mode.
0x4BB00000 + 0x04	LED_DFLT	0x0	Read/Write	Turn on/off LEDs if LEDs are enabled and mode bit = 0.
0x4BB00000 + 0x08	LED_CNT_FRQ	0x0	Read/Write	Define PWM counter clock frequency if mode bit = 1 and clock enable bit = 0
0x4BB00000 + 0x0C	LED_1_PWM_DC	0x0	Read/Write	Define the PWM duty cycle for LED 1. Duty Cycle should be less than the period.
0x4BB00000 + 0x10	LED_2_PWM_DC	0x0	Read/Write	Define the PWM duty cycle for LED 2. Duty Cycle should be less than the period.
0x4BB00000 + 0x14	LED_3_PWM_DC	0x0	Read/Write	Define the PWM duty cycle for LED 3. Duty Cycle should be less than the period.
0x4BB00000 + 0x18	LED_4_PWM_DC	0x0	Read/Write	Define the PWM duty cycle for LED 4. Duty Cycle should be less than the period.
0x4BB00000 + 0x1C	LED_1_PWM_PR	0x0	Read/Write	Define the PWM period value for LED 1.
0x4BB00000 + 0x20	LED_2_PWM_PR	0x0	Read/Write	Define the PWM period value for LED 2.
0x4BB00000 + 0x24	LED_3_PWM_PR	0x0	Read/Write	Define the PWM period value for LED 3.
0x4BB00000 + 0x28	LED_4_PWM_PR	0x0	Read/Write	Define the PWM period value for LED 4.

Registers

LED_CTRL (0x4BB00000 + 0x00) (LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:5	N/A	N/A	Bits 31:5 have no effect.
LED mode	4:4	R/W	0x0	This bit is used to determine whether the LEDs operate in default mode (bit=0), or PWM mode (bit=1). This bit also enables or disables the counter clock, Bit=0 means counter clock is disabled and bit=1 means counter clock is enabled.
LED enable	3:0	R/W	0x0	These bits are used to enable the four LEDs. Writing a 1 to any of these bits will enable the corresponding LED.

LED_DFLT (0x4BB00000 + 0x04) (LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:4	N/A	N/A	Bits 31:4 have no effect.
LED control if operating in default mode	3:0	R/W	0x0	Writing a 1 to a specific bit in this register will turn on the corresponding LED if the LED has been enabled in the LED_CTRL register.

LED_CNT_FRQ (0x4BB00000 + 0x08) (LED)

Name	Bit Range	Type	Reset	Description
PWM counter frequency	31:0	R/W	0x0	If PWM mode and Counter Clock have been enabled, these bits are used to define clock dividers division value X using the following equation: $X = \frac{100MHz}{2*(desired\ Frequency)}$. Where X must be an integer value and desi frequency is in Hz. By default, X value is 12, which corresponds to 4.167MHz as counter clock frequency.

LED_1_PWM_DC (0x4BB00000 + 0x0C) (LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
LED 1 PWM Duty Cycle	15:0	R/W	0x0	These bits are used to choose the PWM Duty Cycle value for LED 1.

LED_2_PWM_DC (0x4BB00000 + 0x10) (LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
LED 2 PWM Duty Cycle	15:0	R/W	0x0	These bits are used to choose the PWM Duty Cycle value for LED 2.

LED_3_PWM_DC (0x4BB00000 + 0x14) (LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
LED 3 PWM Duty Cycle	15:0	R/W	0x0	These bits are used to choose the PWM Duty Cycle value for LED 3.

LED_4_PWM_DC (0x4BB00000 + 0x18) (LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
LED 4 PWM Duty Cycle	15:0	R/W	0x0	These bits are used to choose the PWM Duty Cycle value for LED 4.

LED_1_PWM_PR (0x4BB00000 + 0x1C) (LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
LED 1 PWM Period	15:0	R/W	0x0	These bits are used to choose the PWM period value for LED 1.

LED_2_PWM_PR (0x4BB00000 + 0x20) (LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
LED 2 PWM Period	15:0	R/W	0x0	These bits are used to choose the PWM period value for LED 2.

LED_3_PWM_PR (0x4BB00000 + 0x24) (LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
LED 3 PWM Period	15:0	R/W	0x0	These bits are used to choose the PWM period value for LED 3.

LED_4_PWM_PR (0x4BB00000 + 0x28) (LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
LED 4 PWM Period	15:0	R/W	0x0	These bits are used to choose the PWM period value for LED 4.

Programming Examples in C

Default Mode (LED)

```
#include <stdio.h>
#define LED_BASEADDR 0x4BB00000 // Base address of your LED IP

int main() {
    volatile int i = 0;

    *((uint32_t *) LED_BASEADDR) = 0x0F; //Mode = Default, Counter Clock disabled,
                                                    //all LEDs enabled
    *((uint32_t *) LED_BASEADDR+0x01) = 0xA; // Set LED 3 and LED 1 to 1(HIGH)
    for (i = 0; i < 200000000; i++); // Delay.
    *((uint32_t *) LED_BASEADDR+0x01) = 0x0; // Set LEDs back to 0

    return 1;
}
```

PWM Mode (LED)

```
#include <stdio.h>
// Base address of your _LED IP
#define LED_BASEADDR 0x4BB00000


int main() {
    /*
     * Enable all LEDs, set LEDs to normal mode and disable counter clock before
     * defining counter clock frequency
    */
    *((uint32_t *) LED_BASEADDR) = 0x0F;
    *((uint32_t *) LED_BASEADDR+0x02) = 12; // Set counter clock frequency to 4.166MHz
    /* Enable PWM MODE and counter clock after clock has been configured */
    *((uint32_t *) LED_BASEADDR) = 0x01F;

    *((uint32_t *) LED_BASEADDR+0x03) = 50; // Define the PWM duty cycle for LED 0
    *((uint32_t *) LED_BASEADDR+0x04) = 200; // Define the PWM duty cycle for LED 1
    *((uint32_t *) LED_BASEADDR+0x05) = 500; // Define the PWM duty cycle for LED 2
    *((uint32_t *) LED_BASEADDR+0x06) = 700; // Define the PWM duty cycle for LED 3

    *((uint32_t *) LED_BASEADDR+0x07) = 1000; // Define the PWM period for LED 0
    *((uint32_t *) LED_BASEADDR+0x08) = 1000; // Define the PWM period for LED 1
    *((uint32_t *) LED_BASEADDR+0x09) = 1000; // Define the PWM period for LED 2
    *((uint32_t *) LED_BASEADDR+0x0a) = 1000; // Define the PWM period for LED 3

    return 1;
}
```

LM75BDP NXP Temperature Sensor [4]

Description

Real Digital's Blackboard has a temperature sensor, LM75BDP, provided by NXP semiconductors that can be accessed using I2C1 interface on the Zynq chip. The 7-bit address for this chip is 1001000.

Overview of Registers [4]

Register Address or Pointer Value	Register Name	Reset Value (hex)	Access Type	Description
0x00	Temperature Value	0x00	Read	Contains two 8-bit data bytes; to store the measured temp data.
0x01	Configuration	Output	Read/Write	Configuration register: contains a single 8-bit data byte; to set the device operating condition; default = 0
0x02	Hysteresis Register	0x4B00	Read/Write	Hysteresis register: contains two 8-bit data bytes; to store the hysteresis T_{hys} limit; default = 75 °C.
0x03	Threshold Register	0x5000	Read/Write	Overtemperature shutdown threshold register: contains two 8-bit data bytes; to store the overtemperature shutdown $T_{th(ots)}$ limit; default = 80 °C.

Reading Temperature Data [4]

Temperature Register

The temperature register contains two 8-bit data bytes that consists of MSByte (Most Significant Byte) and LSByte (Least Significant Byte). Figure below shows the register overview. There is a total of 11 bits of actual data. D10 – D3 is considered the integer value of temperature data, where D10 is the sign bit (D10 = 0 means positive, D10 = 1 means negative). D2-D0 are the decimal point precision of the temperature value. For example,

- 011 1111 0111 (1015 in decimal) would equal to +126.875C that can be calculated with the following equation: $Temperature = +Temp\ Data * 0.125\ ℃\ or$

  $Temperature = 1015 * 0.125℃ = 126.875℃$
- 110 0100 1001 (-439 in decimal) would equal to =54.875C that can be calculated with the following equation: $Temperature = -(Two's\ Complement\ of\ Temp\ Data) * 0.125\ ℃\ or$

  $Temperature = -439 * 0.125℃ = -54.875℃$

MSByte								LSByte							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	X	X	X	X	X

Figure 12. Temperature Register (MSB and LSB) [4]

Read Temperature I2C Timing Diagram

At power-up, the pointer value is equal to 00 and the temperature register is selected. This means that users can start reading data right away from the Temperature Value register, see figure 13. If the user wants to access other registers a write transaction of the pointer value (register address) must be followed the address bits as the next byte, see figures 14 and 15 for read and write transaction with the pointer byte (register address).

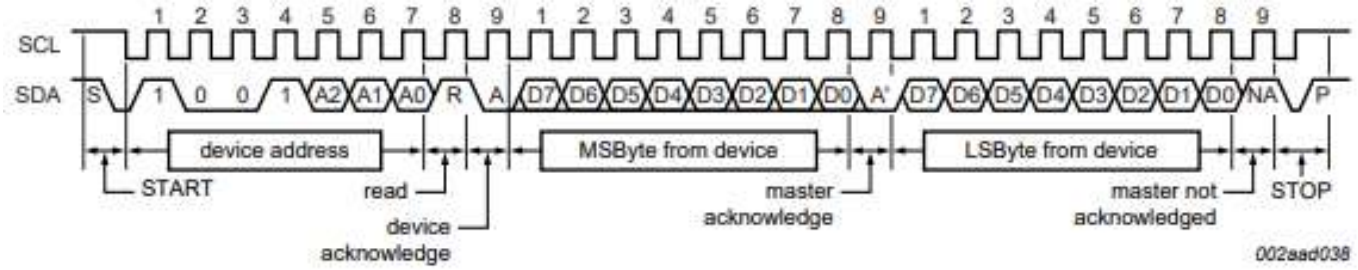

Figure 13. Read Temperature Value register with preset pointer (2-byte data) [4]

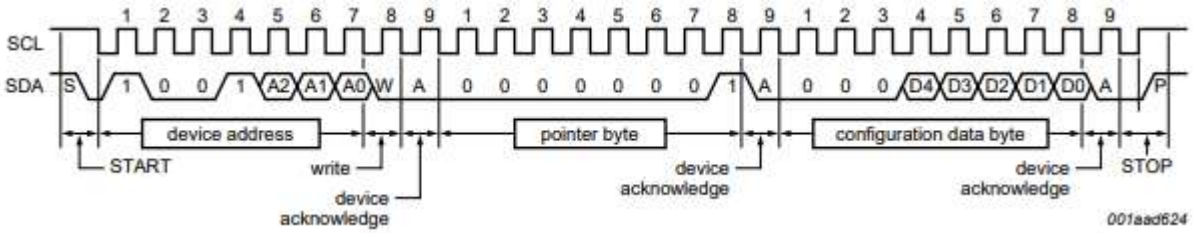

Figure 14. Write Configuration Register (1-Byte Data) [4]

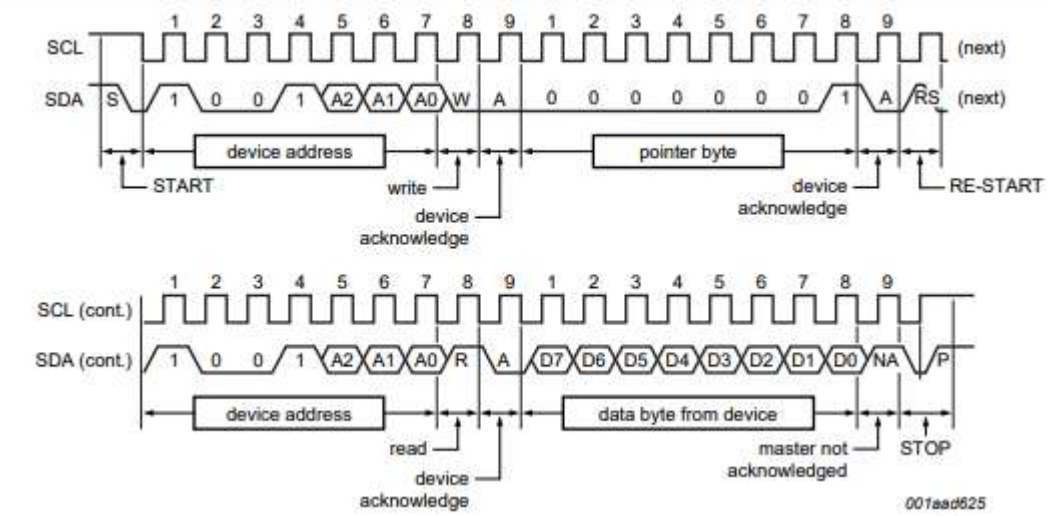

Figure 15. Read configuration register including pointer byte (1-Byte Data) [4]

MIO Pin Control [1]

Description

To initialize the MIO Buttons (BTN4 and BTN5) and MIO RGB LED (LD8), it is important to understand how you can find the corresponding pins that need to be configured. You can find the RGB LED and buttons 4 and 5 from Blackboard's open source board schematics under bank 501, see the figure below. Bank 0 corresponds to MIO pins 31:0 and Bank 1 corresponds to MIO pins 53:32 or 21:0 within the configuration registers. As you can see from the figure, MIO pins 50 and 51 correspond to buttons 4 and 5, respectively; MIO pins 16-18 correspond to the RGB LED colors. All need to be enabled as inputs and correspond to banks 1 and 0.

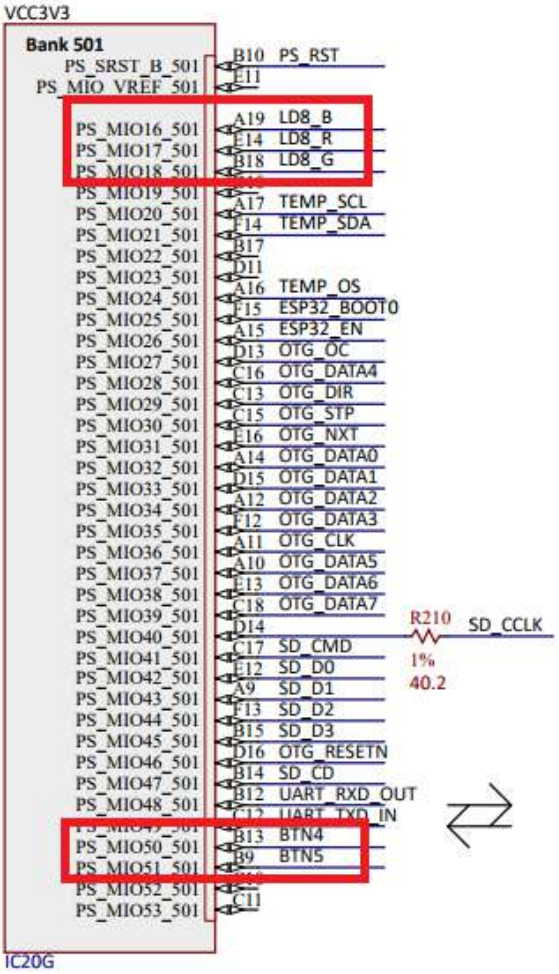

Figure 16. Bank 501; Blackboard's Circuit Schematic

Configuration for MIO pins to GPIO input and output pins is similar and can be done as follows:

1. Select MIO pin as GPIO by setting all bits in Level 0-3 mux selects to 0
2. Disable tristate by setting the tristate enable bit to 0. (if tristate is not disable the GPIO output enable won't have any effect)
3. Choose IO buffer type as LVCMOS33.
4. Select slow CMOS edge as the speed
5. Enable internal pull-up resistor for output pins and disable it for input pins
6. Disable HSTL receiver

Overview of Registers

Base Address + Offset (hex)	Register Name	Reset Value (hex)	Access Type	Description
0xF8000700 + 0x40	MIO_PIN_16	0x1601	Read/Write	MIO Pin 16 control
0xF8000700 + 0x44	MIO_PIN_17	0x1601	Read/Write	MIO Pin 17 control
0xF8000700 + 0x48	MIO_PIN_18	0x1601	Read/Write	MIO Pin 18 control
0xF8000700 + 0xC8	MIO_PIN_50	0x1601	Read/Write	MIO Pin 50 control
0xF8000700 + 0xCC	MIO_PIN_51	0x1601	Read/Write	MIO Pin 51 control

Registers

Configuring MIO pins as GPIO signals can be accomplished using MIO_PIN_XX registers (Zynq-7000 AP SoC Technical Reference Manual page 1633-1685), all of which are identical and work for individual pins. The description of these registers can be found below:

MIO_PIN_XX (0xF8000700 - 0xF80007D4) (MIO)

Name	Bit Range	Type	Reset	Description
Undefined	31:14	N/A	N/A	Bits 31:14 have no effect.
Disable Receiver	13:13	R/W	0x0	Disable HSTL Input Buffer to save power when it is an output-only (IO_Type must be HSTL). 0: enable 1: disable
Pullup	12:12	R/W	0x1	Enables Pullup on IO Buffer pin 0: disable 1: enable
IO_Type	11:9	R/W	0x3	Select the IO Buffer Type. 000: Reserved 001: LVCMOS18 010: LVCMOS25 011: LVCMOS33 100: HSTL 101: Reserved 110: Reserved 111: Reserved
Speed	8:8	R/W	0x0	Select IO Buffer Edge Rate, applicable when IO_Type is LVCMOS18, LVCMOS25 or LVCMOS33. 0: Slow CMOS edge 1: Fast CMOS edge
L3_Select	7:5	R/W	0x0	Level 3 Mux Select 000: GPIO 0 (bank 0), Input/Output others: reserved
L2_Select	4:3	R/W	0x0	Level 2 Mux Select 00: Level 3 Mux 01: SRAM/NOR Chip Select 0, Output 10: NAND Flash Chip Select, Output 11: SDIO 0 Power Control, Output
L1_Select	2:2	R/W	0x0	Level 1 Mux Select 0: Level 2 Mux 1: reserved
L0_Select	1:1	R/W	0x0	Level 0 Mux Select 0: Level 1 Mux 1: Quad SPI 1 chip select, Output
Tristate Enable	0:0	R/W	0x1	Tri-state enable, active high. 0: disable 1: enable

Programming Example in C

```
void Initialize_IO(){
	*((uint32_t *) 0xF8000000+0x8/4) = 0x0000DF0D; // Write unlock code to enable writing
	//into System Level Control Unlock Register
	*((uint32_t*) MIO_PIN_50) = 0x00000600; // BTN4
	*((uint32_t*) MIO_PIN_51) = 0x00000600; // BTN5
	*((uint32_t*) MIO_PIN_16) = 0x00001600; // RGB_LED_B
	*((uint32_t*) MIO_PIN_17) = 0x00001600; // RGB_LED_R
	*((uint32_t*) MIO_PIN_18) = 0x00001600; // RGB_LED_G
	return;
}
```

RGB LED Module

Description

Blackboard has 4 RGB LED's that can be configured to function in two different modes: default mode (DM) and PWM mode (PM). In PWM mode, users have control over both PWM window frequency and duty cycle. This section introduces all registers within the RGB LED IP and how to use these in C coding.

Overview of Registers

Base Address + Offset (hex)	Register Name	Reset Value (hex)	Access Type	Description
0x4BB01000 + 0x00	RGB_LED_CTRL	0x0	Read/Write	Enable LEDs. Select between PWM and default mode.
0x4BB01000 + 0x04	RGB_LED_0_COLOR	0x0	Read/Write	Turn on RGB LED 0 with a specific color if LEDs are enabled and mode bit = 0 or 1.
0x4BB01000 + 0x08	RGB_LED_1_COLOR	0x0	Read/Write	Turn on RGB LED 1 with a specific color if LEDs are enabled and mode bit = 0 or 1.
0x4BB01000 + 0x0C	RGB_LED_2_COLOR	0x0	Read/Write	Turn on RGB LED 2 with a specific color if LEDs are enabled and mode bit = 0 or 1.
0x4BB01000 + 0x10	RGB_LED_3_COLOR	0x0	Read/Write	Turn on RGB LED 3 with a specific color if LEDs are enabled and mode bit = 0 or 1.
0x4BB01000 + 0x14	RGB_LED_0_RED_DC	0x0	Read/Write	Define the PWM duty cycle for RGB LED 0 RED color. Duty Cycle should be less than the period.
0x4BB01000 + 0x18	RGB_LED_0_GRN_DC	0x0	Read/Write	Define the PWM duty cycle for RGB LED 0 GREEN color. Duty Cycle should be less than the period.
0x4BB01000 + 0x1C	RGB_LED_0_BLU_DC	0x0	Read/Write	Define the PWM duty cycle for RGB LED 0 BLUE color. Duty Cycle should be less than the period.

0x4BB01000 + 0x20	RGB_LED_1_RED_DC	0x0	Read/Write	Define the PWM duty cycle for RGB LED 1 RED color. Duty Cycle should be less than the period.
0x4BB01000 + 0x24	RGB_LED_1_GRN_DC	0x0	Read/Write	Define the PWM duty cycle for RGB LED 1 GREEN color. Duty Cycle should be less than the period.
0x4BB01000 + 0x28	RGB_LED_1_BLU_DC	0x0	Read/Write	Define the PWM duty cycle for RGB LED 1 BLUE color. Duty Cycle should be less than the period.
0x4BB01000 + 0x2C	RGB_LED_2_RED_DC	0x0	Read/Write	Define the PWM duty cycle for RGB LED 2 RED color. Duty Cycle should be less than the period.
0x4BB01000 + 0x30	RGB_LED_2_GRN_DC	0x0	Read/Write	Define the PWM duty cycle for RGB LED 2 GREEN color. Duty Cycle should be less than the period.
0x4BB01000 + 0x34	RGB_LED_2_BLU_DC	0x0	Read/Write	Define the PWM duty cycle for RGB LED 2 BLUE color. Duty Cycle should be less than the period.
0x4BB01000 + 0x38	RGB_LED_3_RED_DC	0x0	Read/Write	Define the PWM duty cycle for RGB LED 3 RED color. Duty Cycle should be less than the period.
0x4BB01000 + 0x3C	RGB_LED_3_GRN_DC	0x0	Read/Write	Define the PWM duty cycle for RGB LED 3 GREEN color. Duty Cycle should be less than the period.
0x4BB01000 + 0x40	RGB_LED_3_BLU_DC	0x0	Read/Write	Define the PWM duty cycle for RGB LED 3 BLUE color. Duty Cycle should be less than the period.
0x4BB01000 + 0x44	RGB_LED_0_RED_PR	0x0	Read/Write	Define the PWM period value for RGB LED 0 RED color.
0x4BB01000 + 0x48	RGB_LED_0_GRN_PR	0x0	Read/Write	Define the PWM period value for RGB LED 0 GREEN color.

0x4BB01000 + 0x4C	RGB_LED_0_BLU_PR	0x0	Read/Write	Define the PWM period value for RGB LED 0 BLUE color.
0x4BB01000 + 0x50	RGB_LED_1_RED_PR	0x0	Read/Write	Define the PWM period value for RGB LED 1 RED color.
0x4BB01000 + 0x54	RGB_LED_1_GRN_PR	0x0	Read/Write	Define the PWM period value for RGB LED 1 GREEN color.
0x4BB01000 + 0x58	RGB_LED_1_BLU_PR	0x0	Read/Write	Define the PWM period value for RGB LED 1 BLUE color.
0x4BB01000 + 0x5C	RGB_LED_2_RED_PR	0x0	Read/Write	Define the PWM period value for RGB LED 2 RED color.
0x4BB01000 + 0x60	RGB_LED_2_GRN_PR	0x0	Read/Write	Define the PWM period value for RGB LED 2 GREEN color.
0x4BB01000 + 0x64	RGB_LED_2_BLU_PR	0x0	Read/Write	Define the PWM period value for RGB LED 2 BLUE color.
0x4BB01000 + 0x68	RGB_LED_3_RED_PR	0x0	Read/Write	Define the PWM period value for RGB LED 3 RED color.
0x4BB01000 + 0x6C	RGB_LED_3_GRN_PR	0x0	Read/Write	Define the PWM period value for RGB LED 3 GREEN color.
0x4BB01000 + 0x70	RGB_LED_3_BLU_PR	0x0	Read/Write	Define the PWM period value for RGB LED 3 BLUE color.

Registers

RGB_LED_CTRL (0x4BB01000 + 0x00) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:5	N/A	N/A	Bits 31:5 have no effect.
RGB LED mode	4:4	R/W	0x0	This bit is used to determine whether the RGB LEDs operate in default mode (bit=0), or PWM mode (bit=1).
RGB LED enable	3:0	R/W	0x0	These bits are used to enable the four RGB LEDs. Writing a 1 to any of these bits will enable the corresponding RGB LED.

RGB_LED_0_COLOR (0x4BB01000 + 0x04) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:3	N/A	N/A	Bits 31:3 have no effect.
RGB LED 0 Color Control	2:0	R/W	0x0	Writing a 3-bit value in this register will turn on RGB_LED 0 in the specified color combination if the LED has been enabled in the RGB_LED_CTRL register. This register works even if PWM mode is enabled.

RGB_LED_1_COLOR (0x4BB01000 + 0x08) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:3	N/A	N/A	Bits 31:3 have no effect.
RGB LED 1 Color Control	2:0	R/W	0x0	Writing a 3-bit value in this register will turn on RGB_LED 1 in the specified color combination if the LED has been enabled in the RGB_LED_CTRL register. This register works even if PWM mode is enabled.

RGB_LED_2_COLOR (0x4BB01000 + 0x0C) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:3	N/A	N/A	Bits 31:3 have no effect.
RGB LED 2 Color Control	2:0	R/W	0x0	Writing a 3-bit value in this register will turn on RGB_LED 2 in the specified color combination if the LED has been enabled in the RGB_LED_CTRL register. This register works even if PWM mode is enabled.

RGB_LED_3_COLOR (0x4BB01000 + 0x10) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:3	N/A	N/A	Bits 31:3 have no effect.
RGB LED 3 Color Control	2:0	R/W	0x0	Writing a 3-bit value in this register will turn on RGB_LED 3 in the specified color combination if the LED has been enabled in the RGB_LED_CTRL register. This register works even if PWM mode is enabled.

RGB_LED_0_RED_DC (0x4BB01000 + 0x14) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 0 RED PWM Duty Cycle	15:0	R/W	0x0	These bits are used to choose the PWM Duty Cycle value for RGB LED 0 RED Color.

RGB_LED_0_GRN_DC (0x4BB01000 + 0x18) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 0 GRN PWM Duty Cycle	15:0	R/W	0x0	These bits are used to choose the PWM Duty Cycle value for RGB LED 0 GREEN Color.

RGB_LED_0_BLU_DC (0x4BB01000 + 0x1C) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 0 BLU PWM Duty Cycle	15:0	R/W	0x0	These bits are used to choose the PWM Duty Cycle value for RGB LED 0 BLUE Color.

RGB_LED_1_RED_DC (0x4BB01000 + 0x20) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 1 RED PWM Duty Cycle	15:0	R/W	0x0	These bits are used to choose the PWM Duty Cycle value for RGB LED 1 RED Color.

RGB_LED_1_GRN_DC (0x4BB01000 + 0x24) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 1 GRN PWM Duty Cycle	15:0	R/W	0x0	These bits are used to choose the PWM Duty Cycle value for RGB LED 1 GREEN Color.

RGB_LED_1_BLU_DC (0x4BB01000 + 0x28) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 1 BLU PWM Duty Cycle	15:0	R/W	0x0	These bits are used to choose the PWM Duty Cycle value for RGB LED 1 BLUE Color.

RGB_LED_2_RED_DC (0x4BB01000 + 0x2C) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 2 RED PWM Duty Cycle	15:0	R/W	0x0	These bits are used to choose the PWM Duty Cycle value for RGB LED 2 RED Color.

RGB_LED_2_GRN_DC (0x4BB01000 + 0x30) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 2 GRN PWM Duty Cycle	15:0	R/W	0x0	These bits are used to choose the PWM Duty Cycle value for RGB LED 2 GREEN Color.

RGB_LED_2_BLU_DC (0x4BB01000 + 0x34) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 2 BLU PWM Duty Cycle	15:0	R/W	0x0	These bits are used to choose the PWM Duty Cycle value for RGB LED 2 BLUE Color.

RGB_LED_3_RED_DC (0x4BB01000 + 0x38) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 3 RED PWM Duty Cycle	15:0	R/W	0x0	These bits are used to choose the PWM Duty Cycle value for RGB LED 3 RED Color.

RGB_LED_3_GRN_DC (0x4BB01000 + 0x3C) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 3 GRN PWM Duty Cycle	15:0	R/W	0x0	These bits are used to choose the PWM Duty Cycle value for RGB LED 3 GREEN Color.

RGB_LED_3_BLU_DC (0x4BB01000 + 0x40) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 3 BLU PWM Duty Cycle	15:0	R/W	0x0	These bits are used to choose the PWM Duty Cycle value for RGB LED 3 BLUE Color.

RGB_LED_0_RED_PR (0x4BB01000 + 0x44) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 0 RED PWM Period	15:0	R/W	0x0	These bits are used to choose the PWM period value for RGB LED 0 RED Color.

RGB_LED_0_GRN_PR (0x4BB01000 + 0x48) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 0 GRN PWM Period	15:0	R/W	0x0	These bits are used to choose the PWM period value for RGB LED 0 GREEN Color.

RGB_LED_0_BLU_PR (0x4BB01000 + 0x4C) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 0 BLU PWM Period	15:0	R/W	0x0	These bits are used to choose the PWM period value for RGB LED 1 BLUE Color.

RGB_LED_1_RED_PR (0x4BB01000 + 0x50) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 1 RED PWM Period	15:0	R/W	0x0	These bits are used to choose the PWM period value for RGB LED 1 RED Color.

RGB_LED_1_GRN_PR (0x4BB01000 + 0x54) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 1 GRN PWM Period	15:0	R/W	0x0	These bits are used to choose the PWM period value for RGB LED 1 GREEN Color.

RGB_LED_1_BLU_PR (0x4BB01000 + 0x58) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 1 BLU PWM Period	15:0	R/W	0x0	These bits are used to choose the PWM period value for RGB LED 1 BLUE Color.

RGB_LED_2_RED_PR (0x4BB01000 + 0x5C) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 2 RED PWM Period	15:0	R/W	0x0	These bits are used to choose the PWM period value for RGB LED 2 RED Color.

RGB_LED_2_GRN_PR (0x4BB01000 + 0x60) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 2 GRN PWM Period	15:0	R/W	0x0	These bits are used to choose the PWM period value for RGB LED 2 GREEN Color.

RGB_LED_2_BLU_PR (0x4BB01000 + 0x64) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 2 BLU PWM Period	15:0	R/W	0x0	These bits are used to choose the PWM period value for RGB LED 2 BLUE Color.

RGB_LED_3_RED_PR (0x4BB01000 + 0x68) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 3 RED PWM Period	15:0	R/W	0x0	These bits are used to choose the PWM period value for RGB LED 3 RED Color.

RGB_LED_3_GRN_PR (0x4BB01000 + 0x6C) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 3 GRN PWM Period	15:0	R/W	0x0	These bits are used to choose the PWM period value for RGB LED 3 GREEN Color.

RGB_LED_3_BLU_PR (0x4BB01000 + 0x70) (RGB_LED)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
RGB LED 3 BLU PWM Period	15:0	R/W	0x0	These bits are used to choose the PWM period value for RGB LED 3 BLUE Color.

Programming Examples in C

Default Mode (RGB_LED)

```
#include <stdio.h>
// Base address of your RGB_LED IP
#define RGB_LED_BASEADDR 0x4BB01000

int main() {
    /* Enable all LEDs, set LEDs to default mode */
    *((uint32_t *) RGB_LED_BASEADDR) = 0x0F;
    *((uint32_t *) RGB_LED_BASEADDR+0x01) = 1; // Set RGB_LED_1 color to red
    *((uint32_t *) RGB_LED_BASEADDR+0x02) = 2; // Set RGB_LED_2 color to green
    *((uint32_t *) RGB_LED_BASEADDR+0x03) = 4; // Set RGB_LED_3 color to blue
    *((uint32_t *) RGB_LED_BASEADDR+0x04) = 7; // Set RGB_LED_4 color to white
    return 1;
}
```

PWM Mode (RGB_LED)

```
#include <stdio.h>
// Base address of your RGB_LED IP
#define RGB_LED_BASEADDR 0x4BB01000

int main() {
    /* Enable all LEDs, set LEDs to PWM mode */
       *((uint32_t *) RGB_LED_BASEADDR) = 0x1F;
    //Define PWM duty cycle for RGB LED 3 Color RED
    *((uint32_t *) RGB_LED_BASEADDR+0x0b) = 10;
    //Define PWM duty cycle for RGB LED 3 Color Green
    *((uint32_t *) RGB_LED_BASEADDR+0x0c) = 30;
    // Define PWM duty cycle for RGB LED 3 Color Blue
    *((uint32_t *) RGB_LED_BASEADDR+0x0d) = 60;

    *((uint32_t *) RGB_LED_BASEADDR+23) = 100; //Define PWM period for RGB LED 3 Color RED
    *((uint32_t *) RGB_LED_BASEADDR+24) = 100; //Define PWM period for RGB LED 3 Color Green
    *((uint32_t *) RGB_LED_BASEADDR+25) = 100; //Define PWM period for RGB LED 3 Color Blue

    return 1;
}
```

Seven Segment Display Module

Description

Blackboard has a 4-digit common anode seven-segment display module. As shown in figure below, the first digit of the seven-segment display corresponds to AN1 location and the fourth digit corresponds to AN4 location.

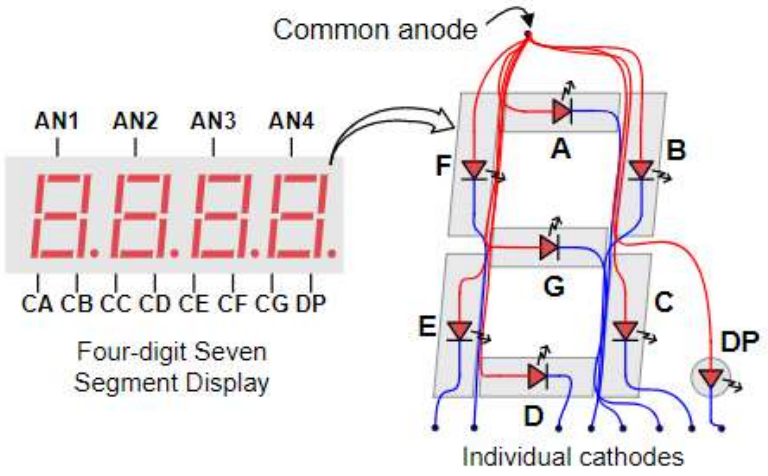

Figure 17. Seven-segment display diagram

This Seven Segment Display IP (SEVEN_SEG_DISP) has two modes: default mode (DM) and custom mode (CM). When operating in default mode, each digit on the 7-segement display can display one hex digit (0-9 and A-F). When operating in custom mode, the user can choose which cathodes are enabled for each digit. As shown in figure 2, bit 0 would correspond to cathode “A,” and bit 6 to cathode “G” in SEVEN_SEG_DIGIT registers.

Overview of Registers

Base Address + Offset (hex)	Register Name	Reset Value (hex)	Access Type	Description
0X4BB03000 + 0x00	SVN_SEG_CTRL	0x0	Read/Write	Enable seven-segment display. Select between custom and default mode.
0X4BB03000 + 0x04	SVN_SEG_DIGIT_1	0x0	Read/Write	Write a value to the first seven segment display digit.
0X4BB03000 + 0x08	SVN_SEG_DIGIT_2	0x0	Read/Write	Write a value to the second seven segment display digit.
0X4BB03000+ 0x0C	SVN_SEG_DIGIT_3	0x0	Read/Write	Write a value to the third seven segment display digit.
0X4BB03000 + 0x10	SVN_SEG_DIGIT_4	0x0	Read/Write	Write a value to the fourth seven segment display digit.
0X4BB03000 + 0x14	SVN_SEG_DP	0x0	Read/Write	Define which decimal points are turned on.

Registers

SVN_SEG_CTRL (0x4BB03000 + 0x00) (SSD)

Name	Bit Range	Type	Reset	Description
Undefined	31:5	N/A	N/A	Bits 31:5 have no effect.
Seven Segment Display Mode	4:4	R/W	0x0	This bit is used to determine whether the seven-segment display operates in default mode (bit = 0), or custom mode (bit = 1).
Undefined	3:1	N/A	N/A	Bits 3:1 have no effect.
Seven Segment Display Enable	0:0	R/W	0x0	This bit is used to enable the seven-segment display. Writing a 1 to this bit enables the display.

SVN_SEG_DIGIT_1 (0x4BB03000 + 0x04) (SSD)

Name	Bit Range	Type	Reset	Description
Undefined	31:4(DM) 31:7(CM)	N/A	N/A	Bits 31:4 have no effect if default mode (DM) is enabled. Bits 31:7 have no effect if custom mode (CM) is enabled.
Seven Segment Display Digit 1 Control	3:0 (DM) 6:0 (CM)	R/W	0x0	These bits are used choose what is being displayed on the first digit based on which mode has been selected.

SVN_SEG_DIGIT_2 (0x4BB03000 + 0x08) (SSD)

Name	Bit Range	Type	Reset	Description
Undefined	31:4(DM) 31:7(CM)	N/A	N/A	Bits 31:4 have no effect if default mode (DM) is enabled. Bits 31:7 have no effect if custom mode (CM) is enabled.
Seven Segment Display Digit 2 Control	3:0 (DM) 6:0 (CM)	R/W	0x0	These bits are used choose what is being displayed on the second digit based on which mode has been selected.

SVN_SEG_DIGIT_3 (0x4BB03000 + 0x0C) (SSD)

Name	Bit Range	Type	Reset	Description
Undefined	31:4(DM) 31:7(CM)	N/A	N/A	Bits 31:4 have no effect if default mode (DM) is enabled. Bits 31:7 have no effect if custom mode (CM) is enabled.
Seven Segment Display Digit 3 Control	3:0 (DM) 6:0 (CM)	R/W	0x0	These bits are used choose what is being displayed on the third digit based on which mode has been selected.

SVN_SEG_DIGIT_4 (0x4BB03000 + 0x10) (SSD)

Name	Bit Range	Type	Reset	Description
Undefined	31:4(DM) 31:7(CM)	N/A	N/A	Bits 31:4 have no effect if default mode (DM) is enabled. Bits 31:7 have no effect if custom mode (CM) is enabled.
Seven Segment Display Digit 4 Control	3:0 (DM) 6:0 (CM)	R/W	0x0	These bits are used choose what is being displayed on the fourth digit based on which mode has been selected.

SVN_SEG_DP (0x4BB03000 + 0x14) (SSD)

Name	Bit Range	Type	Reset	Description
Undefined	31:4	N/A	N/A	Bits 31:4 have no effect.
Seven Segment Display Decimal Point Control	3:0	R/W	0x0	These bits are used to enable decimal points. Bit 0 corresponds to digit 1.

Programming Examples in C

Default Mode (SSD)

```
#include <stdio.h>
#define SSEG 0x4BB03000

int main() {
    *((uint32_t *) SSEG+0x00) = 0x01; //Default mode, enable 7-seg disp
    *((uint32_t *) SSEG+0x01) = 0x0F; //Write F to digit 1
    *((uint32_t *) SSEG+0x02) = 0x05; //Write 5 to digit 2
    *((uint32_t *) SSEG+0x03) = 0x0d; //Write d to digit 3
    *((uint32_t *) SSEG+0x04) = 0x0F; //Write F to digit 4
    *((uint32_t *) SSEG+0x05) = 0x0; // disable all decimal points
}
```

Custom Mode (SSD)

```
#include <stdio.h>
#define SSEG 0x4BB03000

int main() {
    *((uint32_t *) SSEG+0x00) = 0x11; //Custom mode, enable 7-seg disp
    *((uint32_t *) SSEG+0x01) = 0x4E; //Write 0x4E to digit 1, to display r
    *((uint32_t *) SSEG+0x02) = 0x21; //Write 0x21 to digit 2, to display d
    *((uint32_t *) SSEG+0x03) = 0x79; //Write 0x79 to digit 3, to display 1
    *((uint32_t *) SSEG+0x04) = 0x00; //Write 0x00 to digit 4, to display 8
    *((uint32_t *) SSEG+0x05) = 0x2; //Enable second decimal point
}
```

SPI Controller Modules (SPI0 and SPI1) [1]

Description

Real Digital's Blackboard uses SPI0 Module through EMIO (External MIO) pins that is connected to the on-board Navigation Sensor (LSM9DS1 by STMicroelectronics). Since the NAV sensor has three sensors within it (3D Accelerometer, 3D Gyroscope, and 3D Magnetometer) chip select (CS) line 0 is used to initiate data transfer to/from Accelerometer or Gyroscope, and CS line 1 is used to initiate data transfer to/from Magnetometer. You can find more information about the registers within iNEMO Intertial Module section.

Overview of Registers

Base Address + Offset (hex)	Register Name	Reset Value (hex)	Access Type	Description
SPI0: 0xE0006000 + 0x00 SPI1: 0xE0007000 + 0x00	SPI_Config	0x20000	Read/Write	SPI Configuration Register
SPI0: 0xE0006000 + 0x04 SPI1: 0xE0007000 + 0x04	SPI_INT_Status	0x00004	Read/Write	SPI Interrupt Status
SPI0: 0xE0006000 + 0x08 SPI1: 0xE0007000 + 0x08	SPI_INT_EN	0x00000	Read/Write	Interrupt Enable Register
SPI0: 0xE0006000 + 0x0C SPI1: 0xE0007000 + 0x0C	SPI_INT_DIS	0x00000	Read/Write	Interrupt Disable Register
SPI0: 0xE0006000 + 0x10 SPI1: 0xE0007000 + 0x10	SPI_INT_Mask	0x00000	Read	Interrupt Mask Register
SPI0: 0xE0006000 + 0x14 SPI1: 0xE0007000 + 0x14	SPI_Controller_EN	0x00000	Read/Write	SPI Controller Enable
SPI0: 0xE0006000 + 0x18 SPI1: 0xE0007000 + 0x18	SPI_Delay	0x00000	Read/Write	Delay Control
SPI0: 0xE0006000 + 0x1C SPI1: 0xE0007000 + 0x1C	SPI_Transmit	0x00000	Write	Transmit Data.
SPI0: 0xE0006000 + 0x20 SPI1: 0xE0007000 + 0x20	SPI_Receive	0x00000	Read	Receive Data.
SPI0: 0xE0006000 + 0x24 SPI1: 0xE0007000 + 0x24	SPI_Slave_IDL	0x000FF	Read/Write	Slave Idle Count
SPI0: 0xE0006000 + 0x28 SPI1: 0xE0007000 + 0x28	SPI_T_Threshold	0x00001	Read/Write	Transmit FIFO Threshold.
SPI0: 0xE0006000 + 0x2C SPI1: 0xE0007000 + 0x2C	SPI_R_Threshold	0x00001	Read/Write	Receive FIFO Threshold.
SPI0: 0xE0006000 + 0xFC SPI1: 0xE0007000 + 0xFC	SPI_Module_ID	0x90106	Read	Module ID.

Registers

SPI_Config (SPI0: 0xE0006000 + 0x00; SPI1: 0xE0007000 + 0x00) (SPI)

Name	Bit Range	Type	Reset	Description
Undefined	31:18	N/A	N/A	Bits 31:18 have no effect.
ModeFail Generation Enable	17:17	R/W	0x1	Bit = 0: Disable Bit = 1: Enable
Manual Start Command	16:16	W	0x0	Bit = 0: Don't Care Bit = 1: start to transmission of data
Manual Start Enable	15:15	R/W	0x0	Bit = 0: auto mode Bit = 1: enables manual start
Manual CS	14:14	R/W	0x0	Bit = 0: auto mode Bit = 1: manual CS mode
Peripheral Chip Select (CS) Lines	13:10	R/W	0x0	Bits = xxx0 - slave 0 selected Bits = xx01 - slave 1 selected Bits = x011 - slave 2 selected Bits = 0111 - reserved Bits = 1111 - No slave selected
Peripheral Select Decode	9:9	R/W	0x0	Bit = 0: only 1 of 3 selects Bit = 1: allow external 3-to-8 decode
Master Reference Clock Select	8:8	R/W	0x0	Bit = 0: use SPI Reference Clock Bit = 1: not supported
Reserved	7:6	R/W	0x0	Reserved, read as zero, write with 00
Master Mode Baud Rate	5:3	R/W	0x0	Master mode baud rate divisor controls the amount the SPI reference clock is divided inside the SPI block Bits = 000: not supported Bits = 001: divide by 4 Bits = 010: divide by 8 Bits = 011: divide by 16 Bits = 100: divide by 32 Bits = 101: divide by 64 Bits = 110: divide by 128 Bits = 111: divide by 256
Clock Phase	2:2	R/W	0x0	Bit = 0: the SPI clock is active outside the word Bit = 1: the SPI clock is inactive outside the word
Clock Polarity	1:1	R/W	0x0	Bit = 0: the SPI clock is quiescent low Bit = 1: the SPI clock is quiescent high
Mode Select	0:0	R/W	0x0	Bit = 0: the SPI is in slave mode Bit = 1: the SPI is in master mode

SPI_INT_Status (SPI0: 0xE0006000 + 0x04; SPI1: 0xE0007000 + 0x04) (SPI)

Name	Bit Range	Type	Reset	Description
Undefined	31:7	N/A	N/A	Bits 31:7 have no effect.
Transmitter FIFO Underflow	6:6	Write to Clear	0x0	Bit = 0: no underflow has been detected Bit = 1: underflow is detected
Receiver FIFO Full	5:5	Write to Clear	0x0	Bit = 0: FIFO is not full Bit = 1: FIFO is full
Receiver FIFO Not Empty	4:4	Write to Clear	0x0	Bit = 0: FIFO has less than Receiver Threshold entries Bit = 1: FIFO has more than or equal to Threshold entries
Transmitter FIFO Full	3:3	Write to Clear	0x0	Bit = 0: FIFO is not full Bit = 1: FIFO is full
Transmitter FIFO Not Empty	2:2	Write to Clear	0x1	Bit = 0: FIFO has more than or equal to Threshold entries Bit = 1: FIFO has less than Transmitter Threshold entries
ModeFail Interrupt	1:1	Write to Clear	0x0	Indicates the voltage on pin n_ss_in is inconsistent with the SPI mode. Set =1 if n_ss_in is low in master mode (multi-master contention) or n_ss_in goes high during a transmission in slave mode. These conditions will clear the spi_enable bit and disable the SPI. This bit is reset only by a system reset and cleared only when this register is read. Bit = 0: no mode fault has been detected Bit = 1: a mode fault has occurred
Receive Overflow Interrupt	0:0	Write to Clear	0x0	Bit = 0: no overflow occurred Bit = 1: overflow occurred

SPI_INT_EN (SPI0: 0xE0006000 + 0x08; SPI1: 0xE0007000 + 0x08) (SPI)

Name	Bit Range	Type	Reset	Description
Undefined	31:7	N/A	N/A	Bits 31:7 have no effect.
Transmitter FIFO Underflow Interrupt	6:6	W	0x0	Bit = 0: no effect Bit = 1: Enable the Interrupt
Receiver FIFO Full Interrupt	5:5	W	0x0	Bit = 0: no effect Bit = 1: Enable the Interrupt
Receiver FIFO Not Empty Interrupt	4:4	W	0x0	Bit = 0: no effect Bit = 1: Enable the Interrupt
Transmitter FIFO Full Interrupt	3:3	W	0x0	Bit = 0: no effect Bit = 1: Enable the Interrupt
Transmitter FIFO Not Empty Interrupt	2:2	W	0x1	Bit = 0: no effect Bit = 1: Enable the Interrupt
ModeFail Interrupt	1:1	W	0x0	Bit = 0: no effect Bit = 1: Enable the Interrupt
Receive Overflow Interrupt	0:0	W	0x0	Bit = 0: no effect Bit = 1: Enable the Interrupt

SPI_INT_DIS (SPI0: 0xE0006000 + 0x0C; SPI1: 0xE0007000 + 0x0C) (SPI)

Name	Bit Range	Type	Reset	Description
Undefined	31:7	N/A	N/A	Bits 31:7 have no effect.
Transmitter FIFO Underflow Interrupt	6:6	W	0x0	Bit = 0: no effect Bit = 1: Disable the Interrupt
Receiver FIFO Full Interrupt	5:5	W	0x0	Bit = 0: no effect Bit = 1: Disable the Interrupt
Receiver FIFO Not Empty Interrupt	4:4	W	0x0	Bit = 0: no effect Bit = 1: Disable the Interrupt
Transmitter FIFO Full Interrupt	3:3	W	0x0	Bit = 0: no effect Bit = 1: Disable the Interrupt
Transmitter FIFO Not Empty Interrupt	2:2	W	0x0	Bit = 0: no effect Bit = 1: Disable the Interrupt
ModeFail Interrupt	1:1	W	0x0	Bit = 0: no effect Bit = 1: Disable the Interrupt
Receive Overflow Interrupt	0:0	W	0x0	Bit = 0: no effect Bit = 1: Disable the Interrupt

SPI_INT_Mask (SPI0: 0xE0006000 + 0x10; SPI1: 0xE0007000 + 0x10) (SPI)

Name	Bit Range	Type	Reset	Description
Undefined	31:7	N/A	N/A	Bits 31:7 have no effect.
Transmitter FIFO Underflow Interrupt	6:6	R	0x0	Bit = 0: interrupt is enabled Bit = 1: interrupt is disabled
Receiver FIFO Full Interrupt	5:5	R	0x0	Bit = 0: interrupt is enabled Bit = 1: interrupt is disabled
Receiver FIFO Not Empty Interrupt	4:4	R	0x0	Bit = 0: interrupt is enabled Bit = 1: interrupt is disabled
Transmitter FIFO Full Interrupt	3:3	R	0x0	Bit = 0: interrupt is enabled Bit = 1: interrupt is disabled
Transmitter FIFO Not Empty Interrupt	2:2	R	0x0	Bit = 0: interrupt is enabled Bit = 1: interrupt is disabled
ModeFail Interrupt	1:1	R	0x0	Bit = 0: interrupt is enabled Bit = 1: interrupt is disabled
Receive Overflow Interrupt	0:0	R	0x0	Bit = 0: interrupt is enabled Bit = 1: interrupt is disabled

SPI_Controller_EN (SPI0: 0xE0006000 + 0x14; SPI1: 0xE0007000 + 0x14) (SPI)

Name	Bit Range	Type	Reset	Description
Undefined	31:1	N/A	N/A	Bits 31:1 have no effect.
SPI Enable	0:0	R	0x0	Bit = 0: Disable the SPI Bit = 1: Enable the SPI

SPI_Delay (SPI0: 0xE0006000 + 0x18; SPI1: 0xE0007000 + 0x18) (SPI)

Name	Bit Range	Type	Reset	Description
Delay in Master Mode CS Between Words	31:24	R/W	0x0	Delay in SPI Reference Clock or ext_clk cycles for the length that the master mode chip select outputs are de-asserted between words when CPHA=0.
Delay Between Chip Selects De-Activation and Activation	23:16	R/W	0x0	Delay in SPI Reference Clock or ext_clk cycles between one chip select being de-activated and the activation of another
Delay Between Last and First Bit	15:8	R/W	0x0	Delay in SPI Reference Clock or ext_clk cycles between last bit of current word and the first bit of the next word.
Added Delay Before Data Transfer	7:0	R/W	0x0	Added delay in SPI Reference Clock or ext_clk cycles between setting n_ss_out low and first bit transfer.

SPI_Transmit (SPI0: 0xE0006000 + 0x1C; SPI1: 0xE0007000 + 0x1C) (SPI)

Name	Bit Range	Type	Reset	Description
Undefined	31:8	N/A	N/A	Bits 31:8 have no effect.
Data to Transmitter FIFO	7:0	W	0x0	Data to Transmitter FIFO. Writing into this register will initiate data transfer if SPI has been enabled. Size of the FIFO is 128 Bytes deep.

SPI_Receive (SPI0: 0xE0006000 + 0x20; SPI1: 0xE0007000 + 0x20) (SPI)

Name	Bit Range	Type	Reset	Description
Undefined	31:8	N/A	N/A	Bits 31:8 have no effect.
Data from Receiver FIFO	7:0	R	0x0	Data from receiver FIFO. Size of the FIFO is 128 Bytes deep.

SPI_Slave_IDL (SPI0: 0xE0006000 + 0x24; SPI1: 0xE0007000 + 0x24) (SPI)

Name	Bit Range	Type	Reset	Description
Undefined	31:8	N/A	N/A	Bits 31:8 have no effect.
Slave Idle Count	7:0	R/W	0xFF	SPI in slave mode detects a start only when the external SPI master serial clock (sclk_in) is stable (quiescent state) for SPI Reference Clock cycles specified by slave idle count register or when the SPI is deselected.

SPI_T_Threshold (SPI0: 0xE0006000 + 0x28; SPI1: 0xE0007000 + 0x28) (SPI)

Name	Bit Range	Type	Reset	Description
Undefined	31:0	N/A	N/A	Bits 31:8 have no effect.
Transmitter FIFO Threshold Level	7:0	R/W	0x1	Defines the level at which the Transmitter FIFO not full interrupt is generated

SPI_R_Threshold (SPI0: 0xE0006000 + 0x2C; SPI1: 0xE0007000 + 0x28) (SPI)

Name	Bit Range	Type	Reset	Description
Undefined	31:0	N/A	N/A	Bits 31:8 have no effect.
Receiver FIFO Threshold Level	7:0	R/W	0x1	Defines the level at which the Receiver FIFO not full interrupt is generated

SPI_Module_ID (SPI0: 0xE0006000 + 0xFC; SPI1: 0xE0007000 + 0xFC) (SPI)

Name	Bit Range	Type	Reset	Description
Undefined	31:25	N/A	N/A	Bits 31:25 have no effect.
SPI Module ID	24:0	R	0x90106	Module ID number

Switches and Buttons Module

Description

Blackboard has 4 buttons and 8 switches that are non-MIO (Multiplexed Input Output); these can be read through registers from this module.

Overview of Registers

Base Address + Offset (hex)	Register Name	Reset Value (hex)	Access Type	Description
0x4BB02000 + 0x00	SB_SWITCHES	N/A	Read	Read switches
0x4BB02000 + 0x04	SB_BUTTONS	N/A	Read	Read Buttons

Registers

SB_SWITCHES (0x4BB02000 + 0x00) (SB)

Name	Bit Range	Type	Reset	Description
Undefined	31:8	N/A	N/A	Bits 31:8 have no effect.
Switch Values	7:0	R	0x0	Reading from this register will result in the current values of the switches.

SB_BUTTONS (0x4BB02000 + 0x04) (SB)

Name	Bit Range	Type	Reset	Description
Undefined	31:8	N/A	N/A	Bits 31:4 have no effect.
Button Values	3:0	R	0x0	Reading from this register will result in the current values of the buttons.

Programming Example in C

```
#include <stdio.h>
#define SB_BASEADDR 0x4BB02000 // Base address of your Switches and Buttons IP
int main() {
	uint32_t switches, buttons;
	switches = *((uint32_t *) SB_BASEADDR); // read values from switches
	buttons = *((uint32_t *) SB_BASEADDR+0x01); // read values from buttons
	while(1) {
		switches = *((uint32_t *) SB_BASEADDR); // read values from switches
		buttons = *((uint32_t *) SB_BASEADDR+0x01); // read values from buttons
		if(switches == 0x03) {
			printf("Switches 1 and 2 are high\n");
			break;
		}
		else if (buttons == 0x09) {
			printf("Buttons 1 and 4 are high\n");
			break;
		}
	}
    return 1;
}
```

Triple Timer Counter Modules (TTC0 and TTC1) [1]

Description

Blackboard has two Triple Timer Counters (TTC) that are part of the ZYNQ PS. Each TTC has three counters that have an interrupt control system. All the TTC registers introduced within this documentation are taken from Xilinx's documentation.

Overview of Registers

Base Address + Offset (hex)	Register Name	Reset Value (hex)	Access Type	Description
TTC0: 0xF8001000+0x00 TTC1: 0xF8002000+0x00	Clock_Control_1	0x0	Read/Write	Counter 1 Clock Control Register
TTC0: 0xF8001000+0x04 TTC1: 0xF8002000+0x04	Clock_Control_2	0x0	Read/Write	Counter 2 Clock Control Register
TTC0: 0xF8001000+0x08 TTC1: 0xF8002000+0x08	Clock_Control_3	0x0	Read/Write	Counter 3 Clock Control Register
TTC0: 0xF8001000+0x0C TTC1: 0xF8002000+0x0C	Counter_Control_1	0x021	Read/Write	Counter 1 Control Register
TTC0: 0xF8001000+0x10 TTC1: 0xF8002000+0x10	Counter_Control_2	0x021	Read/Write	Counter 2 Control Register
TTC0: 0xF8001000+0x14 TTC1: 0xF8002000+0x14	Counter_Control_3	0x021	Read/Write	Counter 3 Control Register
TTC0: 0xF8001000+0x18 TTC1: 0xF8002000+0x18	Counter_Value_1	0x0	Read	Current Counter 1 Value
TTC0: 0xF8001000+0x1C TTC1: 0xF8002000+0x1C	Counter_Value_2	0x0	Read	Current Counter 2 Value
TTC0: 0xF8001000+0x20 TTC1: 0xF8002000+0x20	Counter_Value_3	0x0	Read	Current Counter 3 Value
TTC0: 0xF8001000+0x24 TTC1: 0xF8002000+0x24	Interval_Value_1	0x0	Read/Write	Counter 1 Interval Value
TTC0: 0xF8001000+0x28 TTC1: 0xF8002000+0x28	Interval_Value_2	0x0	Read/Write	Counter 2 Interval Value
TTC0: 0xF8001000+0x2C TTC1: 0xF8002000+0x2C	Interval_Value_3	0x0	Read/Write	Counter 3 Interval Value
TTC0: 0xF8001000+0x30 TTC1: 0xF8002000+0x30	Match_Value_1 Counter_1	0x0	Read/Write	Match value 1 for counter 1
TTC0: 0xF8001000+0x34 TTC1: 0xF8002000+0x34	Match_Value_1 Counter_2	0x0	Read/Write	Match value 1 for counter 2
TTC0: 0xF8001000+0x38 TTC1: 0xF8002000+0x38	Match_Value_1 Counter_3	0x0	Read/Write	Match value 1 for counter 3
TTC0: 0xF8001000+0x3C TTC1: 0xF8002000+0x3C	Match_Value_2 Counter_1	0x0	Read/Write	Match value 2 for counter 1
TTC0: 0xF8001000+0x40 TTC1: 0xF8002000+0x40	Match_Value_2 Counter_2	0x0	Read/Write	Match value 2 for counter 2
TTC0: 0xF8001000+0x44 TTC1: 0xF8002000+0x44	Match_Value_2 Counter_3	0x0	Read/Write	Match value 2 for counter 3

TTC0: 0xF8001000+0x48 TTC1: 0xF8002000+0x48	Match_Value_3 Counter_1	0x0	Read/Write	Match value 3 for counter 1
TTC0: 0xF8001000+0x4C TTC1: 0xF8002000+0x4C	Match_Value_3 Counter_2	0x0	Read/Write	Match value 2 for counter 1
TTC0: 0xF8001000+0x50 TTC1: 0xF8002000+0x50	Match_Value_3 Counter_3	0x0	Read/Write	Match value 3 for counter 1
TTC0: 0xF8001000+0x54 TTC1: 0xF8002000+0x54	Interrupt Service Routine (ISR) 1	0x0	Clear on Read	Counter 1 Interval, Match, Overflow, and Event interrupts
TTC0: 0xF8001000+0x58 TTC1: 0xF8002000+0x58	Interrupt Service Routine (ISR) 2	0x0	Clear on Read	Counter 2 Interval, Match, Overflow, and Event interrupts
TTC0: 0xF8001000+0x5C TTC1: 0xF8002000+0x5C	Interrupt Service Routine (ISR) 3	0x0	Clear on Read	Counter 3 Interval, Match, Overflow, and Event interrupts
TTC0: 0xF8001000+0x60 TTC1: 0xF8002000+0x60	Interrupt_Enable_1	0x0	Read/Write	Enable Counter 1 Interrupts
TTC0: 0xF8001000+0x64 TTC1: 0xF8002000+0x64	Interrupt_Enable_2	0x0	Read/Write	Enable Counter 2 Interrupts
TTC0: 0xF8001000+0x68 TTC1: 0xF8002000+0x68	Interrupt_Enable_3	0x0	Read/Write	Enable Counter 3 Interrupts
TTC0: 0xF8001000+0x6C TTC1: 0xF8002000+0x6C	Event_Control Timer_1	0x0	Read/Write	Enable, pulse and overflow
TTC0: 0xF8001000+0x70 TTC1: 0xF8002000+0x70	Event_Control Timer_2	0x0	Read/Write	Enable, pulse and overflow
TTC0: 0xF8001000+0x74 TTC1: 0xF8002000+0x74	Event_Control Timer_3	0x0	Read/Write	Enable, pulse and overflow
TTC0: 0xF8001000+0x78 TTC1: 0xF8002000+0x78	Event_Register_1	0x0	Read	Pclk cycle count for event
TTC0: 0xF8001000+0x7C TTC1: 0xF8002000+0x7C	Event_Register_2	0x0	Read	Pclk cycle count for event
TTC0: 0xF8001000+0x80 TTC1: 0xF8002000+0x80	Event_Register_3	0x0	Read	Pclk cycle count for event

Registers

Clock_Control_1 (TTC0: 0xF8001000+0x00; TTC1: 0xF8002000+0x00) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:7	N/A	N/A	Bits 31:7 have no effect.
External Clock Edge	6:6	R/W	0x0	Bit = 1 and external clock is selected: the counter triggers on the negative edge of the external clock input.
Clock Source	5:5	R/W	0x0	Bit = 1: the counter uses external clock input, ext_clk; Bit = 0: clock source is pclk.
Prescale Value	4:1	R/W	0x0	Prescale value (N): if prescale is enabled the count rate is divided by 2^{N+1}
Prescale Enable	0:0	R/W	0x0	Bit = 1: the clock source is prescaled. Bit = 0: prescale is disabled.

Clock_Control_2 (TTC0: 0xF8001000+0x04; TTC1: 0xF8002000+0x04) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:7	N/A	N/A	Bits 31:7 have no effect.
External Clock Edge	6:6	R/W	0x0	Bit = 1 and external clock is selected: the counter triggers on the negative edge of the external clock input.
Clock Source	5:5	R/W	0x0	Bit = 1: the counter uses external clock input, ext_clk; Bit = 0: clock source is pclk.
Prescale Value	4:1	R/W	0x0	Prescale value (N): if prescale is enabled the count rate is divided by 2^{N+1}
Prescale Enable	0:0	R/W	0x0	Bit = 1: the clock source is prescaled. Bit = 0: prescale is disabled.

Clock_Control_3 (TTC0: 0xF8001000+0x08; TTC1: 0xF8002000+0x08) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:7	N/A	N/A	Bits 31:7 have no effect.
External Clock Edge	6:6	R/W	0x0	Bit = 1 and external clock is selected: the counter triggers on the negative edge of the external clock input.
Clock Source	5:5	R/W	0x0	Bit = 1: the counter uses external clock input, ext_clk; Bit = 0: clock source is pclk.
Prescale Value	4:1	R/W	0x0	Prescale value (N): if prescale is enabled the count rate is divided by 2^{N+1}
Prescale Enable	0:0	R/W	0x0	Bit = 1: the clock source is prescaled. Bit = 0: prescale is disabled.

Counter_Control_1 (TTC0: 0xF8001000+0x0C; TTC1: 0xF8002000+0x0C) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:7	N/A	N/A	Bits 31:7 have no effect.
Waveform Polarity	6:6	R/W	0x0	Bit = 1: Waveform output goes from high to low on Match 1 interrupt and returns high on overflow or interval interrupt. Bit = 0: Waveform output goes from low to high on Match 1 interrupt and returns low on overflow or interval interrupt.
Output Waveform Enable	5:5	R/W	0x1	Bit = 1; Output Waveform is Disabled Bit = 0: Output waveform is Enable.
Counter Reset	4:4	R/W	0x0	Bit = 1: counter value is reset, and counting is restarted. When counter is restarted, this reset bit is cleared automatically.
Match Mode	3:3	R/W	0x0	Bit = 1: the timer is in match mode; an interrupt is generated when the count value matches one of the 3 Match Value registers and the corresponding bit is set in the Interrupt Enable 1 register.
Decrement and Increment Mode	2:2	R/W	0x0	Bit = 1: the counter counts down. Bit = 0: the counter counts up.
Interval and Overflow Mode	1:1	R/W	0x0	Bit = 1: Interval Mode, counter generates interrupts at regular intervals. Bit =0: Overflow Mode, timer counts to the maximum value and resets itself afterwards.
Counter Disable	0:0	R/W	0x1	Bit = 1: Counter is stopped, holding its last value until reset, restarted or enabled. Bit = 0: Counter is counting (enabled).

Counter_Control_2 (TTC0: 0xF8001000+0x10; TTC1: 0xF8002000+0x10) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:7	N/A	N/A	Bits 31:7 have no effect.
Waveform Polarity	6:6	R/W	0x0	Bit = 1: Waveform output goes from high to low on Match 2 interrupt and returns high on overflow or interval interrupt. Bit = 0: Waveform output goes from low to high on Match 2 interrupt and returns low on overflow or interval interrupt.
Output Waveform Enable	5:5	R/W	0x1	Bit = 1: Output Waveform is Disabled Bit = 0: Output waveform is Enable.
Counter Reset	4:4	R/W	0x0	Bit = 1: counter value is reset, and counting is restarted. When counter is restarted, this reset bit is cleared automatically.
Match Mode	3:3	R/W	0x0	Bit = 1: the timer is in match mode; an interrupt is generated when the count value matches one of the 3 Match Value registers and the corresponding bit is set in the Interrupt Enable 2 register.
Decrement and Increment Mode	2:2	R/W	0x0	Bit = 1: the counter counts down. Bit = 0: the counter counts up.
Interval and Overflow Mode	1:1	R/W	0x0	Bit = 1: Interval Mode, counter generates interrupts at regular intervals. Bit =0: Overflow Mode, timer counts to the maximum value and resets itself afterwards.
Counter Disable	0:0	R/W	0x1	Bit = 1: Counter is stopped, holding its last value until reset, restarted or enabled. Bit = 0: Counter is counting (enabled).

Counter_Control_3 (TTC0: 0xF8001000+0x14; TTC1: 0xF8002000+0x14) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:7	N/A	N/A	Bits 31:7 have no effect.
Waveform Polarity	6:6	R/W	0x0	Bit = 1: Waveform output goes from high to low on Match 3 interrupt and returns high on overflow or interval interrupt. Bit = 0: Waveform output goes from low to high on Match 3 interrupt and returns low on overflow or interval interrupt.
Output Waveform Enable	5:5	R/W	0x1	Bit = 1; Output Waveform is Disabled Bit = 0: Output waveform is Enable.
Counter Reset	4:4	R/W	0x0	Bit = 1: counter value is reset, and counting is restarted. When counter is restarted, this reset bit is cleared automatically.
Match Mode	3:3	R/W	0x0	Bit = 1: the timer is in match mode; an interrupt is generated when the count value matches one of the 3 Match Value registers and the corresponding bit is set in the Interrupt Enable 3 register.
Decrement and Increment Mode	2:2	R/W	0x0	Bit = 1: the counter counts down. Bit = 0: the counter counts up.
Interval and Overflow Mode	1:1	R/W	0x0	Bit = 1: Interval Mode, counter generates interrupts at regular intervals. Bit =0: Overflow Mode, timer counts to the maximum value and resets itself afterwards.
Counter Disable	0:0	R/W	0x1	Bit = 1: Counter is stopped, holding its last value until reset, restarted or enabled. Bit = 0: Counter is counting (enabled).

Counter_Value_1 (TTC0: 0xF8001000+0x18; TTC1: 0xF8002000+0x18) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Counter Value	15:0	R	0x0	Reading from this address will result in the current value of counter 1.

Counter_Value_2 (TTC0: 0xF8001000+0x1C; TTC1: 0xF8002000+0x1C) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Counter Value	15:0	R	0x0	Reading from this address will result in the current value of counter 2.

Counter_Value_3 (TTC0: 0xF8001000+0x20; TTC1: 0xF8002000+0x20) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Counter Value	15:0	R	0x0	Reading from this address will result in the current value of counter 3.

Interval_Value_1 (TTC0: 0xF8001000+0x24; TTC1: 0xF8002000+0x24) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Interval Value	15:0	R/W	0x0	If interval mode is enabled, this is the maximum value that counter 1 will count up to or down from.

Interval_Value_2 (TTC0: 0xF8001000+0x28; TTC1: 0xF8002000+0x28) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Interval Value	15:0	R/W	0x0	If interval mode is enabled, this is the maximum value that counter 2 will count up to or down from.

Interval_Value_3 (TTC0: 0xF8001000+0x2C; TTC1: 0xF8002000+0x2C) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Interval Value	15:0	R/W	0x0	If interval mode is enabled, this is the maximum value that counter 3 will count up to or down from.

Match_Value_1_Counter_1 (TTC0: 0xF8001000+0x30; TTC1: 0xF8002000+0x30) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Match Value 1	15:0	R/W	0x0	If match mode is enabled and counter 1 has the same value as stored in this register, a match interrupt is generated. Each counter has 3 match registers.

Match_Value_1_Counter_2 (TTC0: 0xF8001000+0x34; TTC1: 0xF8002000+0x34) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Match Value 1	15:0	R/W	0x0	If match mode is enabled and counter 2 has the same value as stored in this register, a match interrupt is generated. Each counter has 3 match registers.

Match_Value_1_Counter_3 (TTC0: 0xF8001000+0x38; TTC1: 0xF8002000+0x38) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Match Value 1	15:0	R/W	0x0	If match mode is enabled and counter 3 has the same value as stored in this register, a match interrupt is generated. Each counter has 3 match registers.

Match_Value_2_Counter_1 (TTC0: 0xF8001000+0x3C; TTC1: 0xF8002000+0x3C) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Match Value 2	15:0	R/W	0x0	If match mode is enabled and counter 1 has the same value as stored in this register, a match interrupt is generated. Each counter has 3 match registers.

Match_Value_2_Counter_2 (TTC0: 0xF8001000+0x40; TTC1: 0xF8002000+0x40) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Match Value 2	15:0	R/W	0x0	If match mode is enabled and counter 2 has the same value as stored in this register, a match interrupt is generated. Each counter has 3 match registers.

Match_Value_2_Counter_3 (TTC0: 0xF8001000+0x44; TTC1: 0xF8002000+0x44) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Match Value 2	15:0	R/W	0x0	If match mode is enabled and counter 3 has the same value as stored in this register, a match interrupt is generated. Each counter has 3 match registers.

Match_Value_3_Counter_1 (TTC0: 0xF8001000+0x48; TTC1: 0xF8002000+0x48) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Match Value 3	15:0	R/W	0x0	If match mode is enabled and counter 1 has the same value as stored in this register, a match interrupt is generated. Each counter has 3 match registers.

Match_Value_3_Counter_2 (TTC0: 0xF8001000+0x4C; TTC1: 0xF8002000+0x4C) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Match Value 3	15:0	R/W	0x0	If match mode is enabled and counter 2 has the same value as stored in this register, a match interrupt is generated. Each counter has 3 match registers.

Match_Value_3_Counter_3 (TTC0: 0xF8001000+0x50; TTC1: 0xF8002000+0x50) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Match Value 3	15:0	R/W	0x0	If match mode is enabled and counter 3 has the same value as stored in this register, a match interrupt is generated. Each counter has 3 match registers.

Interrupt_Service_Routine_1 (TTC0: 0xF8001000+0x54; TTC1: 0xF8002000+0x54) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:6	N/A	N/A	Bits 31:6 have no effect.
Event Timer Overflow Interrupt	5:5	Clear on Read	0x0	Bit = 1: Event Timer 1 overflow has occurred.
Counter Overflow Interrupt	4:4	Clear on Read	0x0	Bit = 1: Counter 1 overflow has occurred.
Match 3 Interrupt	3:3	Clear on Read	0x0	Bit = 1: Counter 1 Match 3 interrupt has occurred
Match 2 Interrupt	2:2	Clear on Read	0x0	Bit = 1: Counter 1 Match 2 interrupt has occurred
Match 1 Interrupt	1:1	Clear on Read	0x0	Bit = 1: Counter 1 Match 1 interrupt has occurred
Interval Interrupt	0:0	Clear on Read	0x0	Bit = 1: Counter 1 Interval interrupt has occurred

Interrupt_Service_Routine_2 (TTC0: 0xF8001000+0x58; TTC1: 0xF8002000+0x58) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:6	N/A	N/A	Bits 31:6 have no effect.
Event Timer Overflow Interrupt	5:5	Clear on Read	0x0	Bit = 1: Event Timer 2 overflow has occurred.
Counter Overflow Interrupt	4:4	Clear on Read	0x0	Bit = 1: Counter 2 overflow has occurred.
Match 3 Interrupt	3:3	Clear on Read	0x0	Bit = 1: Counter 2 Match 3 interrupt has occurred
Match 2 Interrupt	2:2	Clear on Read	0x0	Bit = 1: Counter 2 Match 2 interrupt has occurred
Match 1 Interrupt	1:1	Clear on Read	0x0	Bit = 1: Counter 2 Match 1 interrupt has occurred
Interval Interrupt	0:0	Clear on Read	0x0	Bit = 1: Counter 2 Interval interrupt has occurred

Interrupt_Service_Routine_3 (TTC0: 0xF8001000+0x5C; TTC1: 0xF8002000+0x5C) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:6	N/A	N/A	Bits 31:6 have no effect.
Event Timer Overflow Interrupt	5:5	Clear on Read	0x0	Bit = 1: Event Timer 3 overflow has occurred.
Counter Overflow Interrupt	4:4	Clear on Read	0x0	Bit = 1: Counter 3 overflow has occurred.
Match 3 Interrupt	3:3	Clear on Read	0x0	Bit = 1: Counter 3 Match 3 interrupt has occurred
Match 2 Interrupt	2:2	Clear on Read	0x0	Bit = 1: Counter 3 Match 2 interrupt has occurred
Match 1 Interrupt	1:1	Clear on Read	0x0	Bit = 1: Counter 3 Match 1 interrupt has occurred
Interval Interrupt	0:0	Clear on Read	0x0	Bit = 1: Counter 3 Interval interrupt has occurred

Interrupt_Enable_1 (TTC0: 0xF8001000+0x60; TTC1: 0xF8002000+0x60) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:6	N/A	N/A	Bits 31:6 have no effect.
Event Timer Overflow Interrupt	5:5	R/W	0x0	Bit = 1: Event Timer 1 Interrupt is enabled. Bit = 0: Event Timer 1 Interrupt is disabled.
Counter Overflow Interrupt	4:4	R/W	0x0	Bit = 1: Counter 1 overflow interrupt is enabled. Bit = 0: Counter 1 overflow interrupt is disabled.
Match 3 Interrupt	3:3	R/W	0x0	Bit = 1: Counter 1 Match 3 interrupt is enabled. Bit = 0: Counter 1 Match 3 interrupt is disabled.
Match 2 Interrupt	2:2	R/W	0x0	Bit = 1: Counter 1 Match 2 interrupt is enabled. Bit = 0: Counter 1 Match 2 interrupt is disabled.
Match 1 Interrupt	1:1	R/W	0x0	Bit = 1: Counter 1 Match 1 interrupt is enabled. Bit = 0: Counter 1 Match 1 interrupt is disabled.
Interval Interrupt	0:0	R/W	0x0	Bit = 1: Counter 1 Interval interrupt is enabled. Bit = 0: Counter 1 Interval interrupt is disabled.

Interrupt_Enable_2 (TTC0: 0xF8001000+0x64; TTC1: 0xF8002000+0x64) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:6	N/A	N/A	Bits 31:6 have no effect.
Event Timer Overflow Interrupt	5:5	R/W	0x0	Bit = 1: Event Timer 2 Interrupt is enabled. Bit = 0: Event Timer 2 Interrupt is disabled.
Counter Overflow Interrupt	4:4	R/W	0x0	Bit = 1: Counter 2 overflow interrupt is enabled. Bit = 0: Counter 2 overflow interrupt is disabled.
Match 3 Interrupt	3:3	R/W	0x0	Bit = 1: Counter 2 Match 3 interrupt is enabled. Bit = 0: Counter 2 Match 3 interrupt is disabled.
Match 2 Interrupt	2:2	R/W	0x0	Bit = 1: Counter 2 Match 2 interrupt is enabled. Bit = 0: Counter 2 Match 2 interrupt is disabled.
Match 1 Interrupt	1:1	R/W	0x0	Bit = 1: Counter 2 Match 1 interrupt is enabled. Bit = 0: Counter 2 Match 1 interrupt is disabled.
Interval Interrupt	0:0	R/W	0x0	Bit = 1: Counter 2 Interval interrupt is enabled. Bit = 0: Counter 2 Interval interrupt is disabled.

Interrupt_Enable_3 (TTC0: 0xF8001000+0x68; TTC1: 0xF8002000+0x68) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:6	N/A	N/A	Bits 31:6 have no effect.
Event Timer Overflow Interrupt	5:5	R/W	0x0	Bit = 1: Event Timer 3 Interrupt is enabled. Bit = 0: Event Timer 3 Interrupt is disabled.
Counter Overflow Interrupt	4:4	R/W	0x0	Bit = 1: Counter 3 overflow interrupt is enabled. Bit = 0: Counter 3 overflow interrupt is disabled.
Match 3 Interrupt	3:3	R/W	0x0	Bit = 1: Counter 3 Match 3 interrupt is enabled. Bit = 0: Counter 3 Match 3 interrupt is disabled.
Match 2 Interrupt	2:2	R/W	0x0	Bit = 1: Counter 3 Match 2 interrupt is enabled. Bit = 0: Counter 3 Match 2 interrupt is disabled.
Match 1 Interrupt	1:1	R/W	0x0	Bit = 1: Counter 3 Match 1 interrupt is enabled. Bit = 0: Counter 3 Match 1 interrupt is disabled.
Interval Interrupt	0:0	R/W	0x0	Bit = 1: Counter 3 Interval interrupt is enabled. Bit = 0: Counter 3 Interval interrupt is disabled.

Event_Control_Timer_1 (TTC0: 0xF8001000+0x6C; TTC1: 0xF8002000+0x6C) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:3	N/A	N/A	Bits 31:3 have no effect.
Event Overflow	2:2	R/W	0x0	Bit = 1: Event Timer 1 continues counting on overflow. Bit = 0: Event Timer 1 is disabled and set to zero when an Event Timer 1 register overflow occurs.
Event Low	1:1	R/W	0x0	Bit = 1: Event Timer 1 counts pclk cycles during the low-level duration of ext_clk. Bit = 0: Event Timer 1 counts the high-level duration of ext_clk.
Event Timer Enable	0:0	R/W	0x0	Bit = 1: Enable Event Timer 1. Bit = 0: Disable Event Timer 1.

Event_Control_Timer_2 (TTC0: 0xF8001000+0x70; TTC1: 0xF8002000+0x70) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:3	N/A	N/A	Bits 31:3 have no effect.
Event Overflow	2:2	R/W	0x0	Bit = 1: Event Timer 2 continues counting on overflow. Bit = 0: Event Timer 2 is disabled and set to zero when an Event Timer 2 register overflow occurs.
Event Low	1:1	R/W	0x0	Bit = 1: Event Timer 2 counts pclk cycles during the low-level duration of ext_clk. Bit = 0: Event Timer 2 counts the high-level duration of ext_clk.
Event Timer Enable	0:0	R/W	0x0	Bit = 1: Enable Event Timer 2. Bit = 0: Disable Event Timer 2.

Event_Control_Timer_3 (TTC0: 0xF8001000+0x74; TTC1: 0xF8002000+0x74) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:3	N/A	N/A	Bits 31:3 have no effect.
Event Overflow	2:2	R/W	0x0	Bit = 1: Event Timer 3 continues counting on overflow. Bit = 0: Event Timer 3 is disabled and set to zero when an Event Timer 3 register overflow occurs.
Event Low	1:1	R/W	0x0	Bit = 1: Event Timer 3 counts pclk cycles during the low-level duration of ext_clk. Bit = 0: Event Timer 3 counts the high-level duration of ext_clk.
Event Timer Enable	0:0	R/W	0x0	Bit = 1: Enable Event Timer 3. Bit = 0: Disable Event Timer 3.

Event_Register_1 (TTC0: 0xF8001000+0x78; TTC1: 0xF8002000+0x78) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Event	15:0	R	0x0	Pclk cycle count during the ext_clk high or low pulse.

Event_Register_2 (TTC0: 0xF8001000+0x7C; TTC1: 0xF8002000+0x7C) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Event	15:0	R	0x0	Pclk cycle count during the ext_clk high or low pulse.

Event_Register_3 (TTC0: 0xF8001000+0x80; TTC1: 0xF8002000+0x80) (TTC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Event	15:0	R	0x0	Pclk cycle count during the ext_clk high or low pulse.

System Level Control Registers (SLCR) [1]

Description

Xilinx Zynq All Programmable SoC comes with system level control registers. There are various registers for Xilinx's Zynq SoC modules, such as clock configuration, software reset, and MIO pin configuration registers. MIO pin configuration can be found within in this reference manual under MIO Pin Control. This section focuses on the software reset registers for SPI and I2C modules.

Overview of Registers

Base Address + Offset (hex)	Register Name	Reset Value (hex)	Access Type	Description
0xF8000700 + 0x08	SLCR_UNLOCK	0x0	Write	SLCR Write Protection Unlock
0xF8000700 + 0x21C	SPI_RST_CTRL	0x1601	Read/Write	MIO Pin 17 control
0xF8000700 + 0x224	I2C_RST_CTRL	0x1601	Read/Write	MIO Pin 18 control

Registers

SLCR_UNLOCK (0xF8000000+0x08) (SLCR)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
UNLOCK_KEY	15:0	W	0x0	Write the unlock key, 0xDF0D, to enable writes to the slcr registers. All slcr registers, 0xF800_0000 to 0xF800_0B74, are writeable until locked using the SLCR_LOCK register. A read of this register returns zero.

SPI_RST_CTRL (0xF8000000+0x21C) (SLCR)

Name	Bit Range	Type	Reset	Description
Undefined	31:4	N/A	N/A	Bits 31:4 have no effect.
SPI1_REF_RST	3:3	R/W	0x0	SPIx Reference software reset. On assertion of this reset, the Reference clock portion of the SPIx subsystem will be reset. Bit = 0: No Reset Bit = 1: Reference clock portion of SPIx subsystem held in reset
SPI0_REF_RST	2:2	R/W	0x0	
SPI1_CPU1X_RST	1:1	R/W	0x0	SPIx AMBA software reset. On assertion of this reset, the AMBA clock portion of the SPIx subsystem will be reset. Bit = 0: No Reset Bit = 1: AMBA clock portion of SPIx subsystem held in reset
SPI0_CPU1X_RST	0:0	R/W	0x0	

I2C_RST_CTRL (0xF8000000+0x224) (SLCR)

Name	Bit Range	Type	Reset	Description
Undefined	31:2	N/A	N/A	Bits 31:2 have no effect.
I2C1_CPU1X_RST	1:1	R/W	0x0	I2Cx AMBA software reset. On assertion of this reset, the AMBA clock portion of the I2Cx subsystem will be reset. Bit = 0: No Reset Bit = 1: AMBA clock portion of I2Cx subsystem held in reset
I2C0_CPU1X_RST	0:0	R/W	0x0	

Programming Example in C (SLCR)

SPI Software Reset

```
void SPI_reset(){
	// SPI RESET ************************
	uint32_t register_value;
	*((uint32_t *) MOD_RESET_BASEADDR+0x8/4) = 0x0000DF0D;
	// Assert the reset
	register_value = *((uint32_t *) MOD_RESET_BASEADDR + 0x0000021C/4);
	register_value = register_value | 0xF;
	*((uint32_t *) MOD_RESET_BASEADDR + 0x0000021C/4) = register_value;
	// Release the reset
	register_value = *((uint32_t *) MOD_RESET_BASEADDR + 0x0000021C/4);
	register_value = register_value & ~0xF;
	*((uint32_t *) MOD_RESET_BASEADDR + 0x0000021C/4) = register_value;
	return;
}
```

I2C Software Reset

```
void I2C_reset(){
	//I2C RESET ***************************
	uint32_t register_value;
	/* Unlock the slcr register access lock */
	*((uint32_t *) MOD_RESET_BASEADDR+0x8/4) = 0x0000DF0D;
	/* Assert the reset */
	register_value = *((uint32_t *) MOD_RESET_BASEADDR + 0x00000224/4);
	register_value = register_value | 0x3;
	*((uint32_t *) MOD_RESET_BASEADDR + 0x00000224/4) = register_value;
	/*Release the reset */
	register_value = *((uint32_t *) MOD_RESET_BASEADDR + 0x00000224/4);
	register_value = register_value & ~0x3;
	*((uint32_t *) MOD_RESET_BASEADDR + 0x00000224/4) = register_value;
	return;
}
```

UART Controller Modules (UART0 and UART1) [1]

Description

Xilinx Zynq All Programmable SoC comes with an UART Controller Module that consists UART0 and UART1. Both have pre-assigned connections on the Blackboard by default. UART0 is connected to MIO Pins 14 and 15 (ESP32_TXD0 and ESP32_RXD0, respectively) as shown in figure 6. On the other hand, UART1 is connected to MIO Pins 48 and 49 (UART_TXD_IN and UART_RXD_OUT) as shown in figure 7. The FTDI FT2232H IC automatically connects to a COM port on your computer, which will enable data transfer through UART1 to the console of Xilinx's SDK – this enables the user to send characters through UART.

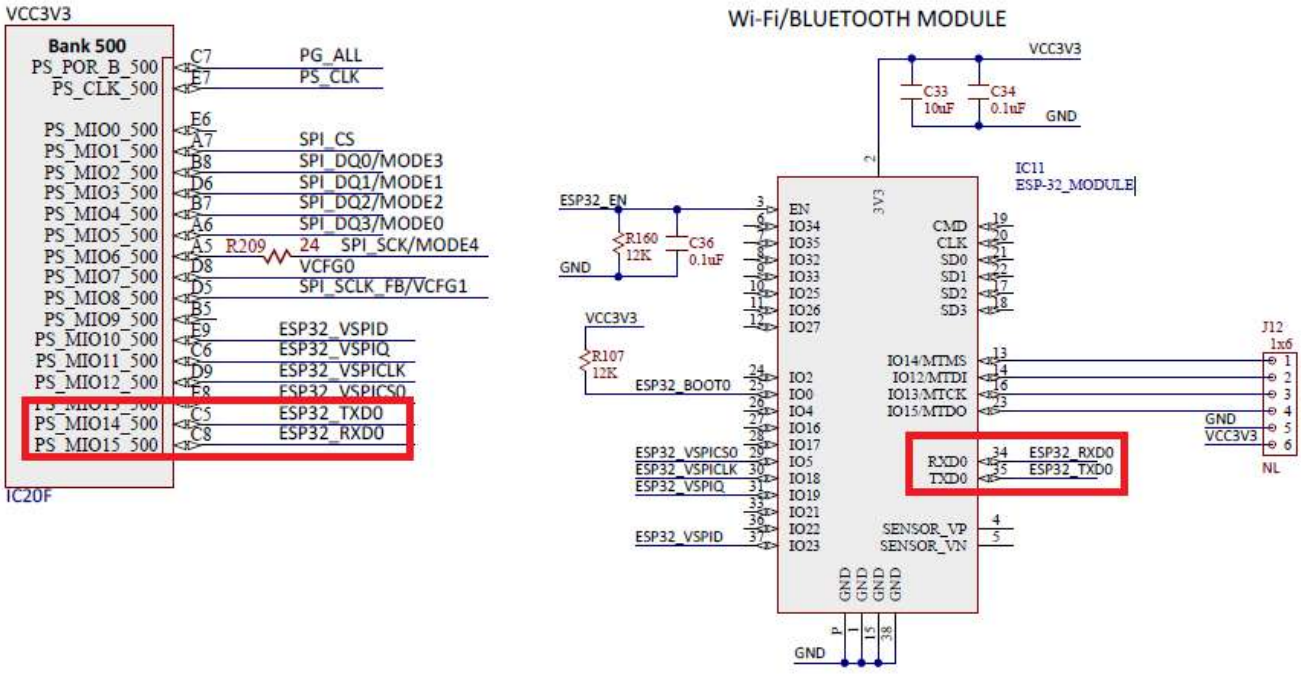

Figure 18 UART0 and ESP32 Wi-Fi/Bluetooth Module

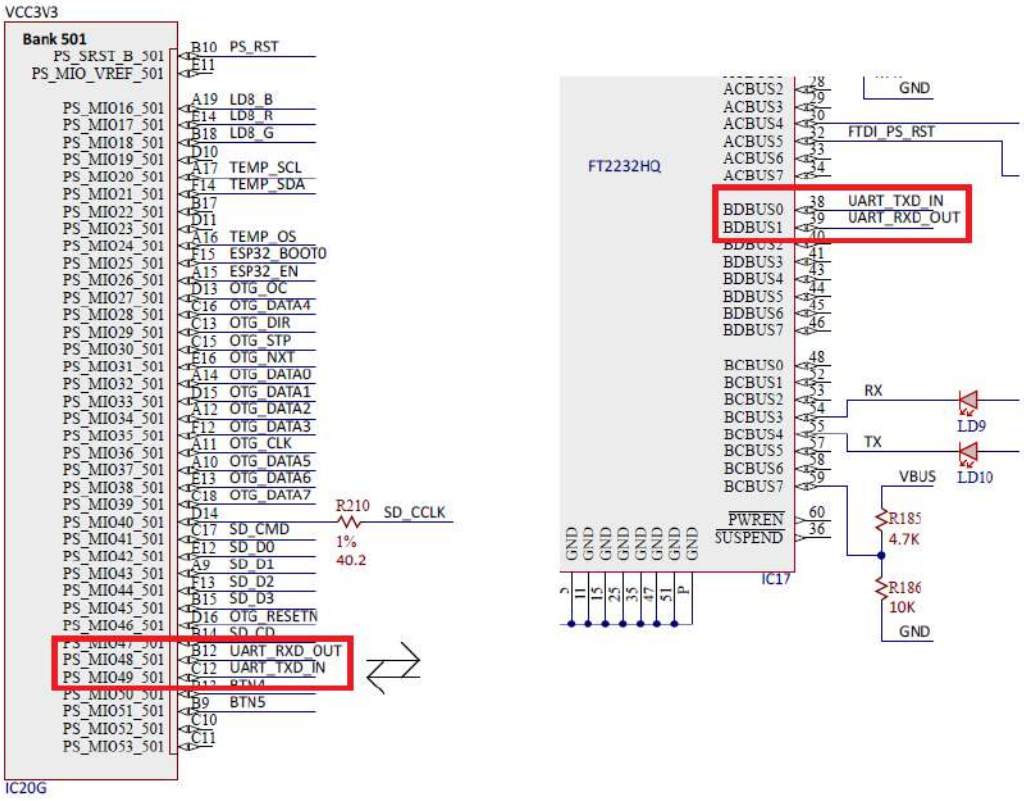

Figure 19 UART1 FTDI IC

Overview of Registers

Base Address + Offset (hex)	Register Name	Reset Value (hex)	Access Type	Description
UART0: 0xE0000000 + 0x00 UART1: 0xE0001000 + 0x00	UART_Control	0x0128	Read/Write	UART Control Register
UART0: 0xE0000000 + 0x04 UART1: 0xE0001000 + 0x04	UART_Mode	0x0	Read/Write	UART Mode Register
UART0: 0xE0000000 + 0x08 UART1: 0xE0001000 + 0x08	UART_INT_EN	0x0	Read/Write	Interrupt Enable Register
UART0: 0xE0000000 + 0x0C UART1: 0xE0001000 + 0x0C	UART_INT_DIS	0x0	Read/Write	Interrupt Disable Register
UART0: 0xE0000000 + 0x10 UART1: 0xE0001000 + 0x10	UART_INT_Mask	0x0	Read	Interrupt Mask Register
UART0: 0xE0000000 + 0x14 UART1: 0xE0001000 + 0x14	UART_ISR	0x0	Write to Clear	Channel Interrupt Status Register
UART0: 0xE0000000 + 0x18 UART1: 0xE0001000 + 0x18	UART_Baud_Gen	0x028B	Read/Write	Baud Rate Generator Register
UART0: 0xE0000000 + 0x1C UART1: 0xE0001000 + 0x1C	UART_RT	0x0	Read/Write	Receiver Timeout Register
UART0: 0xE0000000 + 0x20 UART1: 0xE0001000 + 0x20	UART_RFIFO_LVL	0x20	Read/Write	Receiver FIFO Trigger Level Register
UART0: 0xE0000000 + 0x24 UART1: 0xE0001000 + 0x24	UART_M_Control	0x0	Read/Write	Modem Control Register
UART0: 0xE0000000 + 0x28 UART1: 0xE0001000 + 0x28	UART_M_Status	N/A	Read/Write	Modem Status Register
UART0: 0xE0000000 + 0x2C UART1: 0xE0001000 + 0x2C	UART_C_Status	0x0	Read	Channel Status Register
UART0: 0xE0000000 + 0x30 UART1: 0xE0001000 + 0x30	UART_T_R_FIFO	0x0	Read/Write	Transmit and Receive FIFO
UART0: 0xE0000000 + 0x34 UART1: 0xE0001000 + 0x34	UART_Baud_Div	0x000F	Read/Write	Baud Rate Divider Register
UART0: 0xE0000000 + 0x38 UART1: 0xE0001000 + 0x38	UART_FCD	0x0	Read/Write	Flow Control Delay Register
UART0: 0xE0000000 + 0x44 UART1: 0xE0001000 + 0x44	UART_TFIFO_LVL	0x20	Read/Write	Transmitter FIFO Trigger Level Register

Registers

UART_Control (UART0: 0xE0000000 + 0x00; UART1: 0xE0001000 + 0x00) (UART)

Name	Bit Range	Type	Reset	Description
Undefined	31:9	N/A	N/A	Bits 31:9 have no effect.
Stop Transmitter Break	8:8	R/W	0x1	Bit = 0: no affect Bit = 1: stop transmission of the break after a minimum of one-character length and transmit a high level during 12-bit periods. It can be set regardless of the value of Start Transmitter Break.
Start Transmitter Break	7:7	R/W	0x0	Bit = 0: no affect Bit = 1: start to transmit a break after the characters currently present in the FIFO and the transmit shift register have been transmitted. This bit can only be set if Stop transmitter break is not high.
UART Restart Receiver Timeout Counter	6:6	R/W	0x0	Bit = 1: receiver timeout counter is restarted. This bit will be automatically cleared once the restart has completed.
Transmit Disable	5:5	R/W	0x1	Bit = 0: enable transmitter, if Transmit Enable bit =1 Bit = 1: disable transmitter, regardless of Transmit Enable value
Transmit Enable	4:4	R/W	0x0	Bit = 0: disable transmitter Bit = 1: enable transmitter if the Transmit Disable bit is set to 0.
Receive Disable	3:3	R/W	0x1	Bit = 0: enable, if Receive Enable bit = 1 Bit = 1: disable, regardless of the value of Receive Enable
Receive Enable	2:2	R/W	0x0	Bit = 0: disable Bit = 1: enable When set to one, the receiver logic is enabled, provided the Receive Disable field is set to zero.
Software Reset Tx Data Path	1:1	R/W	0x0	Bit = 0: no affect Bit = 1: transmitter logic is reset, and all pending transmitter data is discarded. This bit will be automatically cleared once the restart has completed.
Software Reset Rx Data Path	0:0	R/W	0x0	Bit = 0: no affect Bit = 1: receiver logic is reset, and all pending receiver data is discarded. This bit will be automatically cleared once the restart has completed.

UART_Mode (UART0: 0xE0000000 + 0x04; UART1: 0xE0001000 + 0x04) (UART)

Name	Bit Range	Type	Reset	Description
Undefined	31:12	N/A	N/A	Bits 31:12 have no effect.
Reserved	11:10	R/W	0x0	Reserved. Do not modify.
Channel Mode	9:8	R/W	0x0	Defines the mode of operation of the UART. Bits = 00: normal Bits = 01: automatic echo Bits = 10: local loopback Bits = 11: remote loopback
Number of Stop Bits	7:6	R/W	0x0	Defines the number of stop bits to detect on receive and to generate on transmit. Bits = 00: 1 stop bit Bits = 01: 1.5 stop bits Bits = 10: 2 stop bits Bits = 11: reserved
Parity Type Select	5:3	R/W	0x0	Defines the expected parity to check on receive and the parity to generate on transmit. Bits = 000: even parity Bits = 001: odd parity Bits = 010: forced to 0 parity (space) Bits = 011: forced to 1 parity (mark) Bits = 1xx: no parity
Character Length Select	2:1	R/W	0x0	Defines the number of bits in each character. Bits = 11: 6 bits Bits = 10: 7 bits Bits = 0x: 8 bits
Clock Source Select	0x0	R/W	0x0	This field defines whether a pre-scalar of 8 is applied to the baud rate generator input clock. Bit = 0: clock source is uart_ref_clk Bit = 1: clock source is uart_ref_clk/8

UART_INT_EN (UART0: 0xE0000000 + 0x08; UART1: 0xE0001000 + 0x08) (UART)

Name	Bit Range	Type	Reset	Description
Undefined	31:13	N/A	N/A	Bits 31:13 have no effect.
Transmitter FIFO Overflow Interrupt	12:12	W	0x0	Bit = 0: no affect Bit = 1: enable (clears mask = 0)
Transmitter FIFO Nearly Full Interrupt	11:11	W	0x0	Bit = 0: no affect Bit = 1: enable (clears mask = 0)
Transmitter FIFO Trigger Interrupt	10:10	W	0x0	Bit = 0: disable Bit =1: enable.
Delta Modem Status Indicator interrupt	9:9	W	0x0	Bit = 0: no affect Bit = 1: enable (clears mask = 0)
Receiver Timeout Error Interrupt	8:8	W	0x0	Bit = 0: no affect Bit = 1: enable (clears mask = 0)
Receiver Parity Error Interrupt	7:7	W	0x0	Bit = 0: disable Bit =1: enable.
Receiver Framing Error Interrupt	6:6	W	0x0	Bit = 0: no affect Bit = 1: enable (clears mask = 0)
Receiver Overflow Error Interrupt	5:5	W	0x0	Bit = 0: no affect Bit = 1: enable (clears mask = 0)
Transmitter FIFO Full Interrupt	4:4	W	0x0	Bit = 0: no affect Bit = 1: enable (clears mask = 0)
Transmitter FIFO Empty Interrupt	3:3	W	0x0	Bit = 0: disable Bit =1: enable.
Receiver FIFO Full Interrupt	2:2	W	0x0	Bit = 0: no affect Bit = 1: enable (clears mask = 0)
Receiver FIFO Empty Interrupt	1:1	W	0x0	Bit = 0: no affect Bit = 1: enable (clears mask = 0)
Receiver FIFO Trigger Interrupt	0:0	W	0x0	Bit = 0: no affect Bit = 1: enable (clears mask = 0)

UART_INT_DIS (UART0: 0xE0000000 + 0x0C; UART1: 0xE0001000 + 0x0C) (UART)

Name	Bit Range	Type	Reset	Description
Undefined	31:13	N/A	N/A	Bits 31:13 have no effect.
Transmitter FIFO Overflow Interrupt	12:12	W	0x0	Bit = 0: no affect Bit = 1: disable (sets mask to 1)
Transmitter FIFO Nearly Full Interrupt	11:11	W	0x0	Bit = 0: no affect Bit = 1: disable (sets mask to 1)
Transmitter FIFO Trigger Interrupt	10:10	W	0x0	Bit = 0: no affect Bit = 1: disable (sets mask to 1)
Delta Modem Status Indicator Interrupt	9:9	W	0x0	Bit = 0: no affect Bit = 1: disable (sets mask to 1)
Receiver Timeout Error Interrupt	8:8	W	0x0	Bit = 0: no affect Bit = 1: disable (sets mask to 1)
Receiver Parity Error Interrupt	7:7	W	0x0	Bit = 0: no affect Bit = 1: disable (sets mask to 1)
Receiver Framing Error Interrupt	6:6	W	0x0	Bit = 0: no affect Bit = 1: disable (sets mask to 1)
Receiver Overflow Error Interrupt	5:5	W	0x0	Bit = 0: no affect Bit = 1: disable (sets mask to 1)
Transmitter FIFO Full Interrupt	4:4	W	0x0	Bit = 0: no affect Bit = 1: disable (sets mask to 1)
Transmitter FIFO Empty Interrupt	3:3	W	0x0	Bit = 0: no affect Bit = 1: disable (sets mask to 1)
Receiver FIFO Full Interrupt	2:2	W	0x0	Bit = 0: no affect Bit = 1: disable (sets mask to 1)
Receiver FIFO Empty Interrupt	1:1	W	0x0	Bit = 0: no affect Bit = 1: disable (sets mask to 1)
Receiver FIFO Trigger Interrupt	0:0	W	0x0	Bit = 0: no affect Bit = 1: disable (sets mask to 1)

UART_INT_Mask (UART0: 0xE0000000 + 0x10; UART1: 0xE0001000 + 0x10) (UART)

Name	Bit Range	Type	Reset	Description
Undefined	31:13	N/A	N/A	Bits 31:13 have no effect.
Transmitter FIFO Overflow Interrupt Mask Status	12:12	R	0x0	Bit = 0: interrupt is disabled Bit = 1: interrupt is enabled
Transmitter FIFO Nearly Full Interrupt Mask Status	11:11	R	0x0	Bit = 0: interrupt is disabled Bit = 1: interrupt is enabled
Transmitter FIFO Trigger Interrupt Mask Status	10:10	R	0x0	Bit = 0: interrupt is disabled Bit = 1: interrupt is enabled
Delta Modem Status Indicator Interrupt Mask Status	9:9	R	0x0	Bit = 0: interrupt is disabled Bit = 1: interrupt is enabled
Receiver Timeout Error Interrupt Mask Status	8:8	R	0x0	Bit = 0: interrupt is disabled Bit = 1: interrupt is enabled
Receiver Parity Error Interrupt Mask Status	7:7	R	0x0	Bit = 0: interrupt is disabled Bit = 1: interrupt is enabled
Receiver Framing Error Interrupt Mask Status	6:6	R	0x0	Bit = 0: interrupt is disabled Bit = 1: interrupt is enabled
Receiver Overflow Error Interrupt Mask Status	5:5	R	0x0	Bit = 0: interrupt is disabled Bit = 1: interrupt is enabled
Transmitter FIFO Full Interrupt Mask Status	4:4	R	0x0	Bit = 0: interrupt is disabled Bit = 1: interrupt is enabled
Transmitter FIFO Empty Interrupt Mask Status	3:3	R	0x0	Bit = 0: interrupt is disabled Bit = 1: interrupt is enabled
Receiver FIFO Full Interrupt Mask Status	2:2	R	0x0	Bit = 0: interrupt is disabled Bit = 1: interrupt is enabled
Receiver FIFO Empty Interrupt Mask Status	1:1	R	0x0	Bit = 0: interrupt is disabled Bit = 1: interrupt is enabled
Receiver FIFO Trigger Interrupt Mask Status	0:0	R	0x0	Bit = 0: interrupt is disabled Bit = 1: interrupt is enabled

UART_ISR (UART0: 0xE0000000 + 0x14; UART1: 0xE0001000 + 0x14) (UART)

Name	Bit Range	Type	Reset	Description
Undefined	31:13	N/A	N/A	Bits 31:13 have no effect.
Transmitter FIFO Overflow Interrupt Mask Status	12:12	Write to Clear	0x0	This event is triggered whenever a new word is pushed into the transmit FIFO when there is not enough room for all the data. This will be set because of any write when the Transmitter FIFO Nearly Full flag in Channel Status Register is already set, or a double byte write when the Transmitter FIFO Nearly Full flag in Channel Status Register is already set. Bit = 0: no interrupt occurred Bit = 1: interrupt occurred
Transmitter FIFO Nearly Full Interrupt Mask Status	11:11	Write to Clear	0x0	This event is triggered whenever a new word is pushed into the transmit FIFO causing the fill level to be such that there is not enough space for a further write of the number of bytes currently specified in the Character Size bits in the Mode register. If this further write were currently attempted, it would cause an overflow. Note that when Character Size is 00, this assumes that a two byte write would be attempted and hence a single byte write is still possible without overflow by driving byte select low for the write. Bit = 0: no interrupt occurred Bit = 1: interrupt occurred
Transmitter FIFO Trigger Interrupt Mask Status	10:10	Write to Clear	0x0	This event is triggered whenever a new word is pushed into the transmit FIFO causing the fill level to become equal to the value defined by Transmit FIFO Trigger Level register. Bit = 0: no interrupt occurred Bit = 1: interrupt occurred
Delta Modem Status Indicator Interrupt Mask Status	9:9	Write to Clear	0x0	This event is triggered whenever the Delta Clear to Send, Delta Data Set Ready, Trailing Edge Ring Indicator, or Delta Data Carrier Detect in the modem status register are being set. Bit = 0: no interrupt occurred Bit = 1: interrupt occurred
Receiver Timeout Error Interrupt Mask Status	8:8	Write to Clear	0x0	This event is triggered whenever the receiver timeout counter has expired due to a long idle condition. Bit = 0: no interrupt occurred Bit = 1: interrupt occurred
Receiver Parity Error Interrupt Mask Status	7:7	Write to Clear	0x0	This event is triggered whenever the received parity bit does not match the expected value. Bit = 0: no interrupt occurred Bit = 1: interrupt occurred
Receiver Framing Error Interrupt Mask Status	6:6	Write to Clear	0x0	This event is triggered whenever the receiver fails to detect a valid stop bit. Bit = 0: no interrupt occurred Bit = 1: interrupt occurred

Receiver Overflow Error Interrupt Mask Status	5:5	Write to Clear	0x0	This event is triggered whenever the contents of the receiver shift register have not yet been transferred to the receiver FIFO and a new start bit is detected. This may be due to the FIFO being full, or due to excessive clock boundary delays. Bit = 0: no interrupt occurred Bit = 1: interrupt occurred
Transmitter FIFO Full Interrupt Mask Status	4:4	Write to Clear	0x0	This event is triggered whenever a new word is inserted into the transmit FIFO causing it to go from a non-full condition to a full condition. Bit = 0: no interrupt occurred Bit = 1: interrupt occurred
Transmitter FIFO Empty Interrupt Mask Status	3:3	Write to Clear	0x0	This event is triggered whenever the final word is removed from the transmit FIFO. Bit = 0: no interrupt occurred Bit = 1: interrupt occurred
Receiver FIFO Full Interrupt Mask Status	2:2	Write to Clear	0x0	This event is triggered whenever a new word is inserted into the receive FIFO causing it to go from a non-full condition to a full condition. Bit = 0: no interrupt occurred Bit = 1: interrupt occurred
Receiver FIFO Empty Interrupt Mask Status	1:1	Write to Clear	0x0	This event is triggered upon exit of the final word from the receive FIFO. Bit = 0: no interrupt occurred Bit = 1: interrupt occurred
Receiver FIFO Trigger Interrupt Mask Status	0:0	Write to Clear	0x0	This event is triggered whenever a new word is inserted into the receive FIFO. Bit = 0: no interrupt occurred Bit = 1: interrupt occurred

UART_Baud_Gen (UART0: 0xE0000000 + 0x18; UART1: 0xE0001000 + 0x18) (UART)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Baud Rate Clock Divisor Value	15:0	R/W	0x28B	Bits = 0: Baud sample is disabled Bits = 1: Clock divisor bypass (baud sample = sel clk) Bits = 2 - 65535: baud sample value

UART_RT (UART0: 0xE0000000 + 0x1C; UART1: 0xE0001000 + 0x1C) (UART)

Name	Bit Range	Type	Reset	Description
Undefined	31:8	N/A	N/A	Bits 31:8 have no effect.
Receiver Timeout Value	7:0	R/W	0x0	Bits = 0: Disables receiver timeout counter Bits = 1 - 255: Receiver timeout in number of baud_sample clocks.

UART_RFIFO_LVL (UART0: 0xE0000000 + 0x20; UART1: 0xE0001000 + 0x20) (UART)

Name	Bit Range	Type	Reset	Description
Undefined	31:6	N/A	N/A	Bits 31:6 have no effect.
Receiver FIFO Trigger Level Value	5:0	R/W	0x20	Bits = 0: Disables receiver FIFO trigger level function Bits = 1 - 63: Trigger set when receiver FIFO fills to the number of bytes specified in this field

UART_M_Control (UART0: 0xE0000000 + 0x24; UART1: 0xE0001000 + 0x24) (UART)

Name	Bit Range	Type	Reset	Description
Undefined	31:6	N/A	N/A	Bits 31:6 have no effect.
Automatic Flow Control Mode	5:5	R/W	0x1	Bit = 0: disable Bit = 1: enable
Undefined	4:2	N/A	0x0	Bits 4:2 have no effect
Request to Send Output Control	1:1	R/W	0x0	This bit is ignored if automatic flow control mode is enabled by Flow Control Mode being high. If FCM is low, the value of this bit is inverted when applied to the EMIOUARTxRTSN output. Bit = 0: EMIOUARTxRTSN output forced to logic 1 Bit = 1: EMIOUARTxRTSN output forced to logic 0
Data Terminal Ready	0:0	R/W	0x0	The value of this bit is inverted when applied to the EMIOUARTxDTRN output. Bit = 0: EMIOUARTxDTRN output forced to logic 1 Bit = 1: EMIOUARTxDTRN output forced to logic 0

UART_M_Status (UART0: 0xE0000000 + 0x28; UART1: 0xE0001000 + 0x28) (UART)

Name	Bit Range	Type	Reset	Description
Undefined	31:9	N/A	N/A	Bits 31:9 have no effect.
Flow Control Mode	8:8	R/W	0x0	Bit = 0: Disabled Bit = 1: Enabled
Data Carrier Detect (DCD) Input Signal from PL Status	7:7	R/W	0x0	Bit = 0: input is high Bit = 1: input is low
Ring Indicator (RI) Input Signal from PL Status	6:6	R/W	0x0	Bit = 0: input is high Bit = 1: input is low
Data Set Ready (DSR) Input Signal from PL Status	5:5	R/W	0x0	Bit = 0: input is high Bit = 1: input is low
Clear to Send (CTS) Input Signal from PL Status	4:4	R/W	0x0	Bit = 0: input is high Bit = 1: input is low
Delta Data Carrier Detect Status	3:3	R/W	0x0	Indicates a change in state of the DCDN input since this bit was last cleared. Bit = 0: No change has occurred Bit = 1: Change has occurred
Trailing Edge Ring Indicator Status	2:2	R/W	0x0	Indicates that the RIN input has change from high to low state since this bit was last cleared. Bit = 0: No trailing edge has occurred Bit = 1: Trailing edge has occurred
Delta Data Set Ready Status	1:1	R/W	0x0	Indicates a change in state of the DSRN input since this bit was last cleared. Bit = 0: No change has occurred Bit = 1: Change has occurred
Delta Clear to Send Status	0:0	R/W	0x0	Indicates a change in state of the CTSN input since this bit was last cleared. Bit = 0: No change has occurred Bit = 1: Change has occurred

UART_C_Status (UART0: 0xE0000000 + 0x2C; UART1: 0xE0001000 + 0x2C) (UART)

Name	Bit Range	Type	Reset	Description
Undefined	31:15	N/A	N/A	Bits 31:15 have no effect.
Transmitter FIFO Nearly Full Continuous Status	14:14	R/W	0x0	This indicates that there is not enough space for the number of bytes currently specified in the Character Length Select bits in the Mode register. If a write were currently attempted, it would cause an overflow. Note that when Character Length Select is 00, this assumes that a two byte write would be attempted and hence a single byte write is still possible without overflow by driving byte selector low for the write. Bit = 0: More than one byte is unused in the Tx FIFO Bit = 1: Only one byte is free in the Tx FIFO
Transmitter FIFO Trigger Continuous Status	13:13	R/W	0x0	Bit = 0: Tx FIFO fill level is less than Transmitter FIFO trigger level Bit = 1: Tx FIFO fill level is greater than or equal to Transmitter FIFO trigger level
Receiver Flow Delay Trigger Continuous Status	12:12	R/W	0x0	Bit = 0: Rx FIFO fill level is less than Flow Control Delay Bit = 1: Rx FIFO fill level is greater than or equal to Flow Control Delay
Transmitter State Machine Active Status	11:11	R/W	0x0	Bit = 0: inactive state Bit = 1: active state
Receiver State Machine Active Status	10:10	R/W	0x0	Bit = 0: inactive state Bit = 1: active state
Undefined	9:5	N/A	0x0	Bits 9:5 have no effect. Do not modify.
Transmitter FIFO Full Continuous Status	4:4	R/W	0x0	Bit = 0: Tx FIFO is not full Bit = 1: Tx FIFO is full
Transmitter FIFO Empty Continuous Status	3:3	R/W	0x0	Bit = 0: Tx FIFO is not empty Bit = 1: Tx FIFO is empty
Receiver FIFO Full Continuous Status	2:2	R/W	0x0	Bit = 0: Rx FIFO is not full Bit = 1: Rx FIFO is full
Receiver FIFO Empty Continuous Status	1:1	R/W	0x0	Bit = 0: Rx FIFO is not empty Bit = 1: Rx FIFO is empty
Receiver FIFO Trigger Continuous Status	0:0	R/W	0x0	Bit = 0: Rx FIFO fill level is less than Receiver FIFO trigger level Bit = 1: Rx FIFO fill level is greater than or equal to Receiver FIFO trigger level

UART_T_R_FIFO (UART0: 0xE0000000 + 0x30; UART1: 0xE0001000 + 0x30) (UART)

Name	Bit Range	Type	Reset	Description
Undefined	31:8	N/A	N/A	Bits 31:8 have no effect.
FIFO	7:0	R/W	0x00	Operates as Tx FIFO and Rx FIFO. Read/Write data from/to the FIFO.

UART_Baud_Div (UART0: 0xE0000000 + 0x34; UART1: 0xE0001000 + 0x34) (UART)

Name	Bit Range	Type	Reset	Description
Undefined	31:8	N/A	N/A	Bits 31:8 have no effect.
Baud Rate Divider Value	7:0	R/W	0xF	Bits = 0 - 3: ignored Bits = 4 - 255: Baud Rate Divider value

UART_FCD (UART0: 0xE0000000 + 0x38; UART1: 0xE0001000 + 0x38) (UART)

Name	Bit Range	Type	Reset	Description
Undefined	31:6	N/A	N/A	Bits 31:6 have no effect.
RxFIFO Trigger Level for Ready to Send (RTS) Output Signal (EMIOUARTxRTSN) De-Assertion	5:0	R/W	0x00	Bits = 0 - 3: Flow delay triggering is disabled, since minimum 4-word hysteresis cannot be satisfied. Bits = 4 to 65535: EMIOUARTxRTSN is driven high when Rx FIFO fill level equals value in this register.

UART_TFIFO_LVL (UART0: 0xE0000000 + 0x44; UART1: 0xE0001000 + 0x44) (UART)

Name	Bit Range	Type	Reset	Description
Undefined	31:6	N/A	N/A	Bits 31:6 have no effect.
Transmitter FIFO Trigger Level	5:0	R/W	0x20	Bits = 0: Disables transmitter FIFO trigger level function Bits = 1 - 63: Trigger set when transmitter FIFO fills to TTRIG bytes

XADC Module [5]

Description

XADC Wizard is an IP provided by Xilinx that can be used via Vivado. “The XADC hard macro data registers are 16 bits in width. The XADC hard macro specification guarantees the first 12 MSB bits accuracy; so only these bits are used for reference.” [5] For more information about the registers refer to the XADC Wizard v3.3 Product Guide. Blackboard is configured by default so that PMODA pins JA2_P/JA2_N and JA3_P/JA3_N are configured for ADC inputs in hardware (see figure below). These pins will represent the VAUX_P_N_13 and VAUX_P_N_9 registers, respectively.

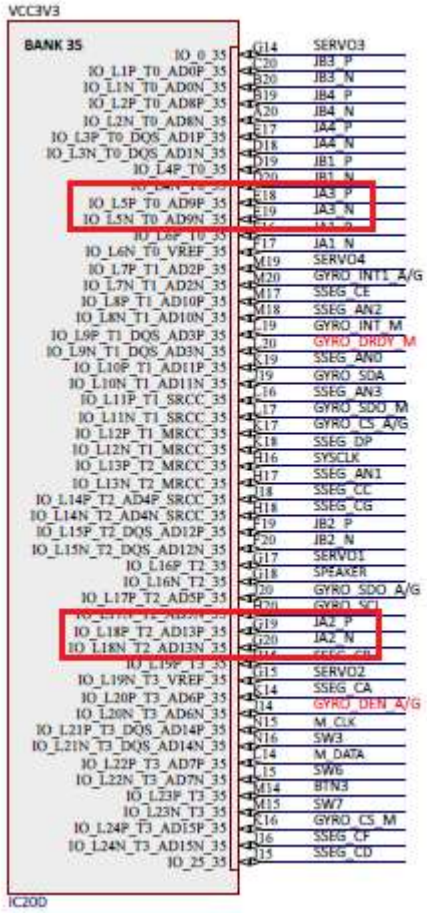

Figure 20 Blackboard Bank 35 pin layout

Overview of Registers

Base Address + Offset (hex)	Register Name	Reset Value (hex)	Access Type	Description
0x4BB06000 + 0x200	XADC_TEMP	N/A	Read	The 12-bit Most Significant Bit (MSB) justified result of on-device temperature measurement is stored in this register.
0x4BB06000 + 0x20C	XADC_Vp_Vn	0x0	Read/ Write	Read: 12-bit MSB justified result of A/D conversion on the dedicated analog input channel is stored in this register Write: Resets XADC hard macro. No specific data required.
0x4BB06000 + 0x264	XADC_VAUX_P_N_9	0x0	Read	The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 9 is stored in this register.
0x4BB06000 + 0x274	XADC_VAUX_P_N_13	0x0	Read	The 12-bit MSB justified result of A/D conversion on the auxiliary analog input 13 is stored in this register.

Registers

XADC_TEMP (0x4BB06000 + 0x200) (XADC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Temp Value	15:0	R	N/A	Reading from this address will return a 12-bit MSB raw on device temperature value. Actual temperature can be calculated using the following formula: $Temp = \frac{ADCdata}{65536 * 0.00198421639} - 273.15\ (°C)$

XADC_Vp_Vn (0x4BB06000 + 0x20C) (XADC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
Vp/Vn Value	15:0	R	0x0	Reading from this address will return a 12-bit MSB raw voltage value. Actual voltage value can be calculated using the following formula: $Voltage = \frac{ADCdata}{65536}\ (V)$

XADC_VAUX_P_N_9 (0x4BB06000 + 0x264) (XADC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
VAUX_P_N_9 Value	15:0	R	0x0	Reading from this address will return a 12-bit MSB raw voltage value. Actual voltage value can be calculated using the following formula: $$Voltage = \frac{ADCdata}{65536}\ (V)$$

XADC_VAUX_P_N_13 (0x4BB06000 + 0x274) (XADC)

Name	Bit Range	Type	Reset	Description
Undefined	31:16	N/A	N/A	Bits 31:16 have no effect.
VAUX_P_N_13 Value	15:0	R	0x0	Reading from this address will return a 12-bit MSB raw voltage value. Actual voltage value can be calculated using the following formula: $$Voltage = \frac{ADCdata}{65536}\ (V)$$

Programming Example in C (XADC)

```
#include <stdio.h>
#include "unistd.h"
#define XADC_DEVICE_ID 0x4BB06000 // baseaddress

//Macro for Raw data value conversion to temperature
#define RawToTemperature(ADC_raw_data)\
            ((((float)(ADC_raw_data)/65536.0f)/0.00198421639f) - 273.15f)

// Macro for Raw voltage data value conversion to voltage
#define RawToVoltage(ADC_raw_data)\
            ((((float)(ADC_raw_data))* (1.0f))/65536.0f)

static int FractToInt(float FloatNumber); // fractions to integers

int main() {
      while(1) {
            float ExtVolData13,ExtVolData9, ExtintegerData;
            //Read the on-chip integererature Data
            uint32_t integerRawData = *((uint32_t *) XADC_DEVICE_ID +0x200/4);
            //convert the data to temp
            ExtintegerData = RawToTemperature(integerRawData);
            //Read the external Vaux7 Data
            uint32_t VolRawData9 = *((uint32_t *) XADC_DEVICE_ID +0x264/4);
            ExtVolData9 = RawToVoltage(VolRawData9); // convert data to voltage
            //Read the external Vaux14 Data
            uint32_t VolRawData13 = *((uint32_t *) XADC_DEVICE_ID +0x274/4);
            ExtVolData13 = RawToVoltage(VolRawData13); // convert data to voltage

            printf("\r\n%0d.%03d [C],%0d.%03d[V],%0d.%03d[V] ",
                        (int)(ExtintegerData), FractToInt(ExtintegerData),
                        (int)(ExtVolData9), FractToInt(ExtVolData9),
                        (int)(ExtVolData13), FractToInt(ExtVolData13));
            usleep(500000); //wait 500ms
      }
      return 0;
}
int FractToInt(float FloatNumber) {
      float integer;
      integer = FloatNumber;
      if (FloatNumber < 0) {
            integer = -(FloatNumber);
      }
      return( ((int)((integer -(float)((int)integer)) * (1000.0f))));
}
```

References

[1] Xilinx, "Xilinx Documentation: Zynq-7000 All Programmable SoC Technical Reference Manual," 6 December 2017. [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf. [Accessed December 2017].

[2] ARM, "ARM® Architecture Reference Manual: ARMv7-A and ARMv7-R edition," 2012. [Online]. Available: http://liris.cnrs.fr/~mmrissa/lib/exe/fetch.php?media=armv7-a-r-manual.pdf. [Accessed 9 February 2018].

[3] STMIcroelectronics, "LSM9DS1; iNEMO inertial module: 3D accelerometer, 3D gyroscope, 3D magnetometer," March 2015. [Online]. Available: http://www.st.com/content/ccc/resource/technical/document/datasheet/1e/3f/2a/d6/25/eb/48/46/DM00103319.pdf/files/DM00103319.pdf/jcr:content/translations/en.DM00103319.pdf. [Accessed February 2018].

[4] NXP Semiconductors, "LM75B: Digital temperature sensor and thermal watchdog," 6 February 2015. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/LM75B.pdf. [Accessed 2 March 2018].

[5] Xilinx, "Xilinx Documentation: XADC Wizard v3.3 LogiCORE IP Product Guide," 5 October 2016. [Online]. Available: https://www.xilinx.com/support/documentation/ip_documentation/xadc_wiz/v3_3/pg091-xadc-wiz.pdf. [Accessed December 2017].

Revision History

Date	Version	Revision
1/13/2019	Rev B	▪ Version Name changed to Rev B
8/13/2018	2.2.0	▪ Logo Update ▪ XADC Register Update ▪ RGB_LED module update (registers updated and PWM example updated) ▪ Fixed Interrupt controller register linking problems ▪ Version changed to align with Hardware Definition File
4/6/2018	2.1.7	▪ Updated table of contents, fixed minor typos in the register description
3/2/2018	2.1.6	▪ Added I2C Interface registers ▪ Changed MIO Buttons C-code example (added write unlock key code) ▪ Added LM75B register descriptions and added citations
2/21/2018	2.1.5	▪ Added SPI Module's registers ▪ Added NAV Sensor Module's Registers ▪ Linked all module register to the "overview of registers"
2/14/2018	2.1.4	▪ Added more registers needed for interrupt configuration ▪ Added UART interface registers
2/12/2018	2.1.3	▪ Added GTC Module's, MIO Configuration's, GPIO Module's sections (overview of register and descriptions of registers, programming examples)
2/9/2018	2.1.2	▪ Added ARM GIC Register Information and Programming examples in C ▪ Fixed minor typos
1/18/2018	2.1.1	▪ Minor wording changes/fixes.
1/17/2018	2.1.0	▪ .hdf file changes: Mic and speaker pins switched in xdc file(fixed now), Xilinx IP updates. ▪ XADC Module updated, fixed register size; ▪ Re-alignment of the header; and ▪ Added Revision History Section.
12/28/2017	2.0	▪ Version changed to 2.0 to align with Hardware Definition File Version; ▪ .hdf file changes: LED module clock enable register removed and combined with MODE bit.
12/18/2017	1.0	▪ Initial Blackboard Programmer's Reference Release