



SDM35541HR

12-Bit Analog I/O dataModule

User's Manual

BDM-610010060 A



RTD Embedded Technologies, Inc.

103 Innovation Boulevard
State College, PA 16803 USA
Telephone: 814-234-8087
Fax: 814-234-5218

www.rtd.com

sales@rtd.com
techsupport@rtd.com





Revision History

Rev A	Initial Release
-------	-----------------

Advanced Analog I/O, Advanced Digital I/O, aAIO, aDIO, a2DIO, Autonomous SmartCal, "Catch the Express", cpuModule, dspFramework, dspModule, eBuild, expressMate, ExpressPlatform, "MIL Value for COTS prices", multiPort, PlatformBus, and PC/104EZ are trademarks, and "Accessing the Analog World", dataModule, IDAN, HiDAN, HiDANplus, RTD, the RTD logo, and StackNET are registered trademarks of RTD Embedded Technologies, Inc. (formerly Real Time Devices, Inc.). PS/2 is a trademark of International Business Machines Inc. PCI, PCI Express, and PCIe are trademarks of PCI-SIG. PC/104, PC/104-Plus, PCI-104, PCIe/104, PCI/104-Express and 104 are trademarks of the PC/104 Consortium. All other trademarks appearing in this document are the property of their respective owners.

Failure to follow the instructions found in this manual may result in damage to the product described in this manual, or other components of the system. The procedure set forth in this manual shall only be performed by persons qualified to service electronic equipment. Contents and specifications within this manual are given without warranty, and are subject to change without notice. RTD Embedded Technologies, Inc. shall not be liable for errors or omissions in this manual, or for any loss, damage, or injury in connection with the use of this manual.

Copyright © 2026 by RTD Embedded Technologies, Inc. All rights reserved.

Table of Contents

1	Introduction	10
1.1	Product Overview.....	10
1.2	Board Features	10
1.3	Ordering Information.....	11
1.4	Contact Information	11
1.4.1	Sales Support	11
1.4.2	Technical Support	11
2	Specifications	12
2.1	Operating Conditions	12
2.2	Electrical Characteristics	12
3	Board Connection	14
3.1	Board Handling Precautions	14
3.2	Physical Characteristics.....	14
3.3	Connectors and Jumpers.....	15
3.3.1	External I/O Connectors	15
	CN6: External Digital I/O Connector	16
	CN9: External Analog I/O Connector	16
	CN5: SyncBus	17
	CN18: McBSP	17
3.3.2	Bus Connectors	17
	CN1(Top) & CN2(Bottom): PCIe Connector	17
3.3.3	Jumpers	17
	JP1: Local Reset	17
3.4	Steps for Installing	18
4	IDAN Connections	19
4.1	Module Handling Precautions	19
4.2	Physical Characteristics.....	19
4.3	Connectors.....	20
4.3.1	External I/O Connectors	20
4.4	Steps for Installing	23
5	Functional Description	24
5.1	Block Diagram.....	24
5.2	Hardware Description	25
5.2.1	Channel Gain Latch (CGL) and Channel Gain Table (CGT)	25
	Channel Gain Table Mode:	25
	Multi-Channel Operation:	25
5.2.2	Analog-to-Digital Conversion	26
	A/D Converter	26
	A/D FIFO – Sample Buffer	26
	Data Transfer	27
5.2.3	Digital-to-Analog Conversion	27
5.2.4	Timer/Counters	27
5.2.5	Digital I/O	27
5.2.6	High-Speed Digital Inputs	27
5.2.7	SyncBus	28
5.2.8	McBSP Multi-channel Buffered Serial Port	28
5.2.9	Default State of I/O	28
5.3	Analog Connections.....	28
5.3.1	Connecting Analog Input Pins	28

Ground Referenced Single-Ended (GRSE) Input Mode	29
Non-Referenced Single-Ended (NRSE) Input Mode	30
Differential (DIFF) Input Mode	31
5.3.2 Connecting Analog Outputs	31
5.3.3 Connecting the Timer/Counters and Digital I/O	32
6 Register Address Space	33
Register Types	33
Function Block Mapping	33
6.1 BAR0: General Board Control	34
6.1.1 GBC_FMT (Read-Only)	34
6.1.2 GBC_REV (Read-Only)	34
6.1.3 GBC_BRD_RST (Read/Write)	34
6.1.4 GBC_PDP (Read-Only)	34
6.1.5 GBC_BUILD (Read-Only)	34
6.1.6 GBC_SYS_CLK_FREQ (Read Only)	35
6.1.7 GBC_USER_ID (Read Only)	35
6.1.8 GBC_IRQ_STATUS (Read/Clear)	35
6.1.9 GBC_DIRQ_STATUS (Read/Clear)	35
6.1.10 GBC_EOI (Read/Clear)	35
6.1.11 FBn_ID (Read-Only)	35
6.1.12 FBn_Offset (Read-Only)	35
6.1.13 FBn_Offset_DMA (Read-Only)	35
6.2 BAR2: Functional Block Standard DMA	36
6.2.1 FB_DMAM_Action (Read/Write)	36
6.2.2 FB_DMAM_Last_Action (Read/Write)	36
6.2.3 FB_DMAM_Setup (Read/Write)	36
6.2.4 FB_DMAM_Stat_Used (Read/Write)	37
6.2.5 FB_DMAM_Stat_Invalid (Read/Write)	37
6.2.6 FB_DMAM_Stat_Overflow (Read/Write)	37
6.2.7 FB_DMAM_Stat_Underflow (Read/Write)	37
6.2.8 FB_DMAM_Stat_Complete (Read/Write)	37
6.2.9 FB_DMAM_Current_Buffer (Read-Only)	37
6.2.10 FB_DMAM_COUNT (Read-Only)	37
6.2.11 FB_DMAM_RD_FIFO_CNT(Read-Only)	37
6.2.12 FB_DMAM_WR_FIFO_CNT(Read-Only)	37
6.2.13 FB_DMAM_ADDRESSn (Read/Write)	37
6.2.14 FB_DMAM_SIZE n (Read/Write)	37
6.2.15 FB_DMAM_CTRLn (Read/Write)	38
6.2.16 FB_DMAM_STATn (Read/Clear)	38
6.3 BAR2: Functional Block Header Registers	38
6.3.1 FB_ID (Read-Only)	38
6.3.2 FB_DMA_CHANNELS (Read -Only)	38
6.3.3 FB_DMA_BUFFERS (Read-Only)	38
6.3.4 FB_CONTROL_SIZE (Read -Only)	38
6.3.5 FB_CHANNEL_SIZE (Read-Only)	38
6.4 BAR2: SDM355xx Functional Block	39
6.4.1 Function Block Register Map	39
6.4.2 USER_IO (Read/Write)	41
Read Operation	41
Write operation	41
6.4.3 DAC_CLK_ST (Read/Write)	41
Read Operation	41
Write Operation	41
6.4.4 FIFO_STATUS (Read/Write)	41
Read Operation	41
Write operation	42
6.4.5 DAC1_UPDATE (Write)	42
Write operation	42
6.4.6 DAC2_UPDATE (Write)	42
Write operation	42
6.4.7 DAC_UPDATE (Write)	42
Write operation	42
6.4.8 PACER (Read/Write)	42

Read Operation	42
Write Operation	42
6.4.9 TIMER (Read/Write)	42
Read Operation	42
Write Operation	43
6.4.10 INT (Read/Write)	43
Read Operation	43
Write Operation	43
6.4.11 CLEAR_INT (Read/Write)	44
Read Operation	44
Write Operation	44
6.4.12 IT_OVERRUN (Read/Write)	44
Read Operation	44
Write Operation	44
6.4.13 PCLK_CNT (Read/Write)	45
Read Operation	45
Write Operation	45
6.4.14 BURST_CNT (Read/Write)	45
Read Operation	45
Write Operation	45
6.4.15 ADC_SAMPLE (Read/Write)	45
Read Operation	45
Write Operation	45
6.4.16 DACn_SAMPLE (Read/Write)	46
Read Operation	46
Write Operation	46
6.4.17 DELAY_CNT (Read/Write)	46
Read Operation	46
Write Operation	46
6.4.18 ABOUT_CNT (Read/Write)	46
Read Operation	46
Write Operation	46
6.4.19 DAC_CLK (Read/Write)	46
Read Operation	46
Write Operation	47
6.4.20 UTCn (Read/Write)	47
Read Operation	47
Write Operation	47
6.4.21 UTC_CTRL (Write)	47
Write Operation	47
6.4.22 DIO0 (Read/Write)	47
Read Operation	47
Write Operation	47
6.4.23 DIO1 (Read/Write)	48
Read Operation	48
Write Operation	48
6.4.24 DIO_CTRL (Read/Write)	48
Direction Register (DIO_STATUS, bits 1 and 0 = 01):	48
Mask Register (DIO_STATUS, bits 1 and 0 = 10):	48
Compare Register (DIO_STATUS, bits 1 and 0 = 11):	48
6.4.25 DIO_STATUS (Read/Write)	49
Read Operation	49
Write Operation	49
6.4.26 DSP (Read/Write)	49
Read Operation	49
Write Operation	49
6.4.27 Temperature Sensor Registers	50
6.4.28 ALGDIO_MASK (Read/Write)	50
Read Operation	50
Write Operation	50
6.4.29 ALGDIO_DATA (Read/Write)	50
Read Operation	50
Write Operation	50
6.4.30 ALGDIO_DIR (Read/Write)	50
Read Operation	50
Write Operation	50

6.4.31	ALGDIO_INT (Read Only)	51
	Read Operation	51
6.4.32	BOARD_RESET (Write Only)	51
6.4.33	DMA0_SRC (Write Only)	51
6.4.34	DMA1_SRC (Write Only)	51
6.4.35	ADC_CONV (Write Only)	52
6.4.36	BURST_START (Write Only)	52
6.4.37	PACER_START (Write Only)	52
6.4.38	PACER_STOP (Write Only)	53
6.4.39	ACNT_ENABLE (Write Only)	53
6.4.40	PACER_MODE (Write Only)	53
6.4.41	HSDIN_START (Write Only)	53
6.4.42	HSDIN_FIFO_CLR (Write Only)	53
6.4.43	ADC_FIFO_CLR (Write Only)	53
6.4.44	CGT_WRITE (Write Only)	54
	Using the pause bit:	54
	Using the skip bit:	54
6.4.45	CGT_LATCH (Write Only)	54
6.4.46	DT_WRITE (Write Only)	54
6.4.47	CGT_ENABLE (Write Only)	55
6.4.48	DT_ENABLE (Write Only)	55
6.4.49	PAUSE_TABLE (Write Only)	55
6.4.50	CGT_RESET (Write Only)	55
6.4.51	CGT_CLEAR (Write Only)	55
6.4.52	DAC1_RANGE (Write Only)	55
6.4.53	DAC1_SRC (Write Only)	55
6.4.54	DAC1_CYCLE (Write Only)	56
6.4.55	DAC1_RESET (Write Only)	56
6.4.56	DAC1_CLEAR (Write Only)	56
6.4.57	DAC2_RANGE (Write Only)	56
6.4.58	DAC2_SRC (Write Only)	56
6.4.59	DAC2_CYCLE (Write Only)	56
6.4.60	DAC2_RESET (Write Only)	56
6.4.61	DAC1_CLEAR (Write Only)	56
6.4.62	ADC_SCNT_SRC (Write Only)	56
6.4.63	PACER_SELECT (Write Only)	57
6.4.64	SBUS0_SOURCE (Write Only)	57
6.4.65	SBUS0_ENABLE (Write Only)	57
6.4.66	SBUS1_SOURCE (Write Only)	58
6.4.67	SBUS1_ENABLE (Write Only)	58
6.4.68	SBUS2_SOURCE (Write Only)	58
6.4.69	SBUS2_ENABLE (Write Only)	58
6.4.70	ETRG_POLARITY (Write Only)	58
6.4.71	EINT_POLARITY (Write Only)	58
6.4.72	UTC0_CLOCK (Write Only)	59
6.4.73	UTC0_GATE (Write Only)	59
6.4.74	UTC1_CLOCK (Write Only)	59
6.4.75	UTC1_GATE (Write Only)	59
6.4.76	UTC2_CLOCK (Write Only)	60
6.4.77	UTC2_GATE (Write Only)	60
6.4.78	UIO0_SELECT (Write Only)	60
6.4.79	UIO1_SELECT (Write Only)	60
6.4.80	DMA_RSTRQST0 (Write Only)	60
6.4.81	DMA_RSTRQST1 (Write Only)	60
6.4.82	DAC_CLK_START (Write Only)	60
6.4.83	DAC_CLK_STOP (Write Only)	60
6.4.84	PCLK_FREQ (Write Only)	61
6.4.85	BCLK_FREQ (Write Only)	61
6.4.86	DAC_CLK_SOURCE (Write Only)	61
6.4.87	DAC_CLK_MODE (Write Only)	61
6.4.88	MCBSP_AD_CTRL (Write Only)	61
6.4.89	MCBSP_DA_CTRL (Write Only)	61
6.4.90	ADC_FIFO (Read Only)	62
6.4.91	HSDIN_FIFO (Read Only)	62
6.4.92	DAC1_FIFO (Write Only)	63
6.4.93	DAC2_FIFO (Write Only)	63

6.4.94	DMA_CH0_FIFO (READ/WRITE)	63
6.4.95	DMA_CH1_FIFO (READ/WRITE)	64
7	Programming Guide	65
7.1	A/D Conversion	65
7.1.1	Before Starting Conversions: Initializing the Board	65
	Before Starting Conversions (single-channel mode): Programming Channel Gain Latch (CGL)	65
	Before Starting Conversions (multi-channel mode): Programming the Channel-Gain Table (CGT)	65
	16-Bit A/D Table	65
	Channel Select, Gain Select, Input Range and Input Type	65
	Pause bit	65
	D/Ax update bits	66
	Skip bit	66
	8-Bit Digital Table	66
	Setting Up A/D part and Digital part of Channel Gain Table	66
	Using the Channel Gain Table for A/D Conversions	67
	Channel-gain Table and Throughput Rates	67
7.1.2	A/D Conversion Modes	67
	Start A/D Conversion signal	67
	Pacer Clock Start/Stop Trigger Select	68
	Types of Conversions	69
7.1.3	Reading the Converted Data	70
7.1.4	Using the A/D Data Markers	71
7.1.5	Programming the Pacer Clock	71
7.1.6	Programming the Burst Clock	72
7.1.7	Programming the About Counter	72
	Using the About Counter to Create Large Data Arrays	72
7.2	D/A Conversion	73
7.2.1	8K Sample Buffer	75
7.2.2	D/A Cycled or Not Cycled Mode	75
7.2.3	D/A Update Counters	75
7.2.4	D/A Data Markers	75
7.3	Advanced Digital Interrupts	76
	Event Mode	76
	Match Mode	76
	Sampling Digital Lines for Change of State	76
7.4	Timer/Counters	77
7.4.1	The internal Timer Timer/Counters	77
7.4.2	User Timer Timer/Counters	77
7.5	Digital I/O	78
7.5.1	The Digital I/O Chip	79
	Port 0, Bit Programmable Digital I/O	79
	Advanced Digital Interrupts: Mask and Compare Registers	79
	Port 1, Port Programmable Digital I/O	79
	Resetting the Digital Circuitry	80
	Strobing Data into Port 0	80
7.5.2	High-Speed Digital Input	80
7.5.3	Digital Input Data Markers	80
7.6	On-board DSP	80
8	Troubleshooting	81
9	Additional Information	82
9.1	PC/104 Specifications	82
9.2	PCI and PCI Express Specification	82
10	Limited Warranty	83

Table of Figures

Figure 1: Board Dimensions	14
Figure 2: Board Connections	15
Figure 3: Example 104™ Stack	18
Figure 4: IDAN Dimensions	19
Figure 5: Example IDAN System	23
Figure 6: SDM35541HR Block Diagram	24
Figure 7: Channel Gain Table	26
Figure 8: High-Speed Digital Inputs	28
Figure 9: Ground Referenced Single Ended input mode	29
Figure 10: Non-Referenced Single-Ended input mode	30
Figure 11: Differential input mode	31
Figure 12: Digital I/O	32
Figure 13: SyncBus Structure	57
Figure 14: User TC Section	59
Figure 15: A/D Conversion Signal	67
Figure 16: User Timer/Counter	77
Figure 17: Digital I/O Port 0	78
Figure 18: Digital I/O Port 1	79

Table of Tables

Table 1: Ordering Options	11
Table 2: Operating Conditions	12
Table 3: Electrical Characteristics	12
Table 4: Signal Functions	15
Table 5: Digital I/O Connector	16
Table 6: Analog I/O Connector	16
Table 7: SyncBus	17
Table 8: McBSP	17
Table 9: IDAN-SDM35541 Analog I/O 26-pin "D" Connector	20
Table 10: IDAN-SDM35541 Digital I/O 44-pin "D" Connector	21
Table 11: IDAN-SDM35541 SyncBus 9-pin "D" Connector	22
Table 12: IDAN-SDM35541 McBSP 9-pin "D" Connector	22
Table 13: Function Block Mapping	33
Table 14: Base Functional Block	34
Table 15: DMA Registers	36
Table 16: Functional Block Header	38
Table 17: SDM35xx Functional Block	39
Table 18: Pacer Clock Frequency	71
Table 19: Burst Clock Frequency	72
Table 20: DAC Clock Frequency	73
Table 21: DAC Bipolar Binary Value vs. Output Voltage	74
Table 22: DAC Unipolar Binary Value vs. Output Voltage	75

1 Introduction

1.1 Product Overview

The SDM35541 is a high-speed, 12-bit data acquisition module in PCIe/104 format with software-configurable settings and autocalibration. It offers 8 differential or 16 single-ended analog input channels with programmable gain and input ranges. Additionally, it includes two independent analog outputs, multi-board synchronization via SyncBus, and a multi-channel buffered McBSP serial port. A built-in temperature sensor supports thermal monitoring and control.

1.2 Board Features

- High-Speed Analog Inputs
 - 8 Differential or 16 Single-ended channels
 - 12-bit A/D with matched internal Sample and Hold
 - 0.8 μ s conversion time (1.25 MHz throughput)
 - ± 5 , ± 10 , 0 to +10 V Analog Input Ranges
 - Programmable binary gains of 1, 2, 4, 8, 16 & 32
 - 1K entry Channel-Gain Scan Memory with Skip Bit
- High-Speed Data Paths
 - Multi-channel Buffered Serial Port (McBSP)
 - 8K sample buffer on A/D converter
 - 8K sample buffer on each D/A converter channel
- Versatile Triggering
 - Software, Pace Clock, Burst Clock and External Triggers
 - Pre-, Post-, and About- Trigger Modes
 - Random Scan, burst and multi-burst using channel-gain table
- Two fast analog outputs
 - 12-bit resolution
 - 7 μ s full-scale settling time (± 10 V range)
 - ± 5 , +5, ± 10 , & +10V output ranges
 - 5mA output current
- Digital I/O
 - 3 Data-marker Input Bits
 - 1K byte digital input buffer
 - 8-bit-programmable digital I/O lines and an 8-bit programmable port
 - Advanced digital interrupts
 - -12/+24mA output drive currents
- Timer and Counters
 - Twelve 16-bit, 8 MHz timer/counters to support timing and counting functions (3 available for user applications)
- Programmable Interrupt Source
- Auto calibration (SmartCal) for A/D and D/A with onboard DSP
- On-board Temperature sensor
- PCI Express Bus:
 - Provides 2.5 Gbps in each direction
 - Single lane and single Virtual Channel operation
 - Compatible with multi-Virtual Channel chipsets
 - Packetized serial traffic with PCI Express Split Completion protocol
 - Data Link Layer Cyclic Redundancy Check (CRC) generator and checker
 - Automatic Retry of bad packets
 - In-band interrupts and messages
 - Message Signaled Interrupt (MSI) support

1.3 Ordering Information

The SDM35541HR is available with the following options:

Table 1: Ordering Options

Part Number	Description
SDM35541HR	PCIe/104 12-bit Analog I/O Smart dataModule
IDAN-SDM35541HR	PCIe/104 12-bit Analog I/O Smart dataModule in IDAN enclosure

The Intelligent Data Acquisition Node (IDAN®) building block can be used in just about any combination with other IDAN building blocks to create a simple but rugged 104™ stack. This module can also be incorporated in a custom-built RTD HiDAN® or HiDANplus® High Reliability Intelligent Data Acquisition Node. Contact RTD sales for more information on our high reliability systems.

1.4 Contact Information

1.4.1 SALES SUPPORT

For sales inquiries, you can contact RTD Embedded Technologies sales via the following methods:

Phone: 1-814-234-8087 Monday through Friday, 8:00am to 5:00pm (EST).
E-Mail: sales@rtd.com

1.4.2 TECHNICAL SUPPORT

If you are having problems with your system, please try the steps in the [Troubleshooting](#) section of this manual on page 81.

For help with this product, or any other product made by RTD, you can contact RTD Embedded Technologies technical support via the following methods:

Phone: 1-814-234-8087 Monday through Friday, 8:00am to 5:00pm (EST).
E-Mail: techsupport@rtd.com

2 Specifications

2.1 Operating Conditions

Table 2: Operating Conditions

Symbol	Parameter	Test Condition	Min	Max	Unit
V _{cc5}	5V Supply Voltage		4.75	5.25	V
V _{cc3}	3.3V Supply Voltage		n/a	n/a	V
V _{cc12}	12V Supply Voltage		n/a	n/a	V
V _{cc-12}	-12V Supply Voltage		n/a	n/a	V
T _a	Operating Temperature		-40	+85	C
T _s	Storage Temperature		-55	+125	C
RH	Relative Humidity	Non-Condensing	0	90%	%
MTBF	Mean Time Before Failure	Telcordia Issue 2 30°C, Ground benign, controlled		TBD	Hours

2.2 Electrical Characteristics

Table 3: Electrical Characteristics

Symbol	Parameter	Test Condition	Min	Max	Unit
P	Power Consumption	V _{cc5} = 5.0V		TBD	W
I _{cc5}	5V Input Supply Current	Active		TBD	mA
PCIe Bus					
	Differential Output Voltage		0.8	1.2	V
	DC Differential TX Impedance		95.2	116.9	Ω
	Differential Input Voltage		0.175	3.3	V
	DC Differential RX Impedance		92.7	115.8	Ω
	Electrical Idle Detect Threshold		61	173	mV
A/D Circuitry					
	Input Impedance			>10	MΩ
	Gains		1	128	
	Gain Error			0.1%	
	Input Voltage	20V Range Mode		±10	V
		10V Range Mode		±5	V
		Unipolar Mode	0	10	V
	Over Voltage Protection			±15	V
	Common Mode			±10	V
	Channel Scanning Error (Gain=1)	10...500kHz		0%	
		600kHz		0.05%	
		700kHz		0.075%	
		800kHz		0.25%	
		900kHz		0.4%	
		1000kHz		0.45%	
		1250kHz		1%	
	Resolution			12	Bits
	Linearity Error			±1	LSB
	Sampling Rate			1.25	MHz
	FIFO Size X16			8	Kb
	Channel Gain table X24			1	Kb
Digital I/O					
	Output Current			±8	mA
	Input Termination			10	KΩ
	Output Termination			10	Ω

Table 3: Electrical Characteristics

Symbol	Parameter	Test Condition	Min	Max	Unit
D/A Circuitry					
	Resolution			12	Bits
	Relative Accuracy			±1	LSB
	Full-Scale Accuracy			±5	LSB
	Non-linearity			±1	LSB
	Settling Time			5	µs
	Output Current			5	mA
	FIFO Size X16	Per channel		8	Kb

3 Board Connection

3.1 Board Handling Precautions

To prevent damage due to Electrostatic Discharge (ESD), keep your board in its antistatic bag until you are ready to install it into your system. When removing it from the bag, hold the board at the edges, and do not touch the components or connectors. Handle the board in an antistatic environment, and use a grounded workbench for testing and handling of your hardware.

3.2 Physical Characteristics

STEP model is available upon request; contact RTD Tech Support for more information.

- Weight: Approximately 0.12 lbs. (55 g)
- Dimensions: 3.550 in. L x 3.775 in. W (90.17 mm L x 95.89 mm W)

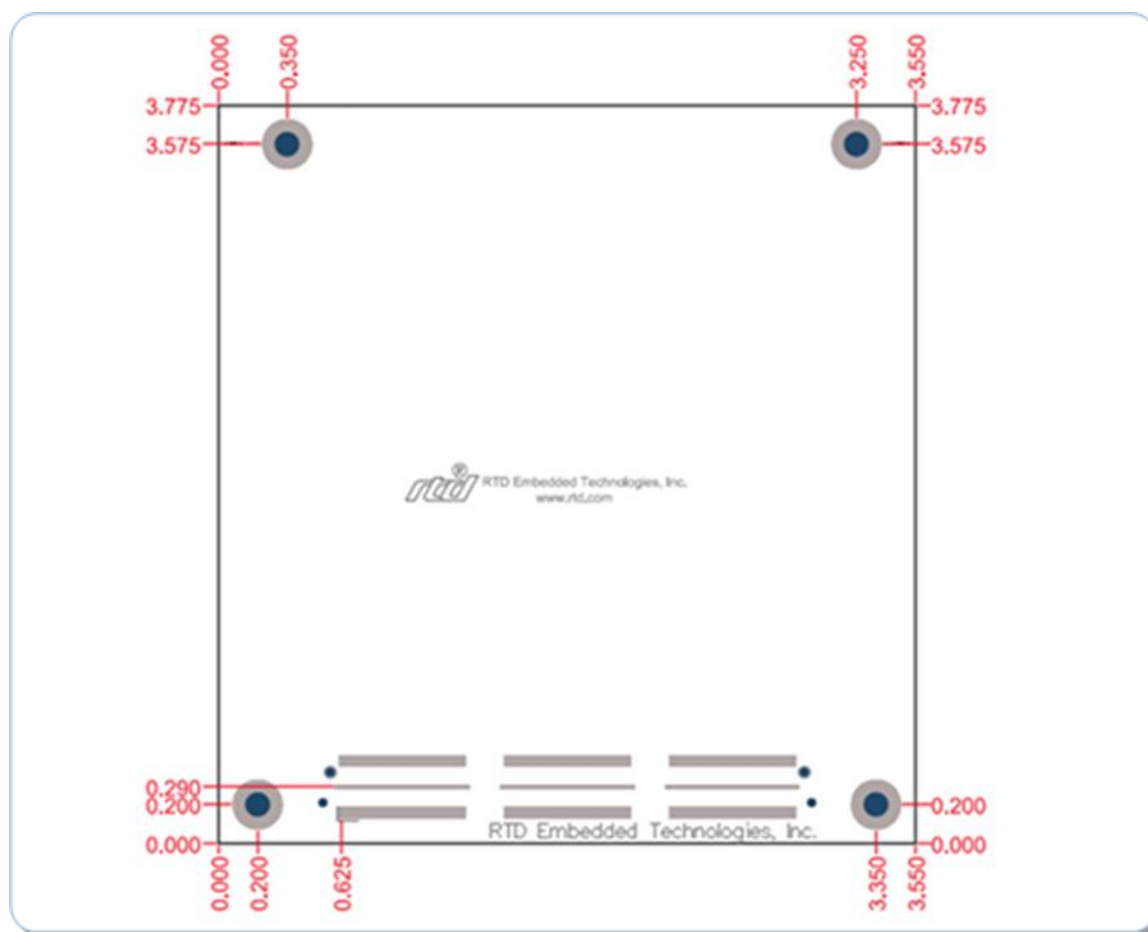


Figure 1: Board Dimensions

3.3 Connectors and Jumpers

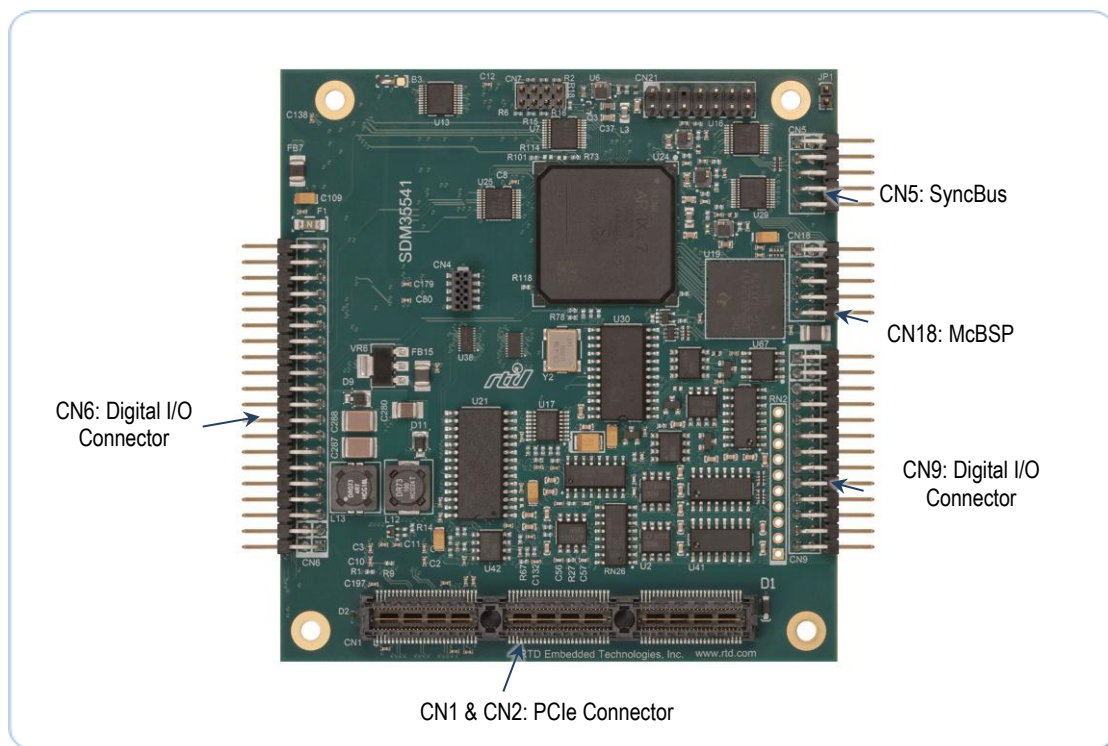


Figure 2: Board Connections

3.3.1 EXTERNAL I/O CONNECTORS

Table 4: Signal Functions

Signal	Function
AINx / AINx +/-AINx -	SE Analog input high sides / DIFF analog input high sides / DIFF analog inputs low sides.
AINSENSE	Reference Signal in Non-ground referenced Single Ended (NRSE) input mode.
AOUT x	Analog outputs.
AGND	Analog Ground
D/A 1 Data Marker 0	D/A 1 0 the output data markers.
D/A 2 Data Marker 0	D/A 2 0 the output data markers.
High Speed Input x / P0.x / A/D DMx	High speed inputs to digital input FIFO / Bit programmable P0 lines from digital I/O Chip.
P1.x / DIG Table x	Port programmable lines from digital I/O Chip. Outputs from digital part of channel gain table.
Trigger Input	External trigger input to trigger A/D pacer clock. (LS TTL)
External Pace Clock Input	External pacer clock to clock A/D. (LS TTL)
Reset	Active low reset output line asserted when the host PC is in hardware reset, or the Board Clear Command is active. (LS TTL)
External Interrupt Input	Programmable rising or falling edge external Interrupt source. (LS TTL)
User Input x	User Input 0 and User Input 1 can be read by the USER_IO. (LS TTL)
User Output x	The source of these buffered lines can be programmed. (LS TTL)
External TC Gate x	External gate signals that go to the software programmable clock source select circuit for the user timer/counters. (LS TTL)

Signal	Function
TC Out x	Buffered outputs from the user timer/counters. (LS TTL)
External TC Clock x	External clock signals that go to the software programmable clock source select circuit for the user timer/counters. (LS TTL)
DGND	Digital Signal Ground
+5V	+5 Volts from the computer power supply to power front end boards. (Max. 2A)

CN6: External Digital I/O Connector

CN6 is an external connector for the majority of Digital I/O connections on the board. The connector contains signals for the digital I/O, data triggers, and timer/counter. The pin assignment is listed below.

Table 5: Digital I/O Connector

P1.7 / DIG TABLE 7	2	1	HIGH SPEED INPUT 7 / P0.7 / A/D DM2
P1.6 / DIG TABLE 6	4	3	HIGH SPEED INPUT 6 / P0.6 / A/D DM1
P1.5 / DIG TABLE 5	6	5	HIGH SPEED INPUT 5 / P0.5 / A/D DM0
P1.4 / DIG TABLE 4	8	7	HIGH SPEED INPUT 4 / P0.4
P1.3 / DIG TABLE 3	10	9	HIGH SPEED INPUT 3 / P0.3
P1.2 / DIG TABLE 2	12	11	HIGH SPEED INPUT 2 / P0.2
P1.1 / DIG TABLE 1	14	13	HIGH SPEED INPUT 1 / P0.1
P1.0 / DIG TABLE 0	16	15	HIGH SPEED INPUT 0 / P0.0
Reserved	18	17	Reserved
D/A 2 DATA MARKER 0	20	19	D/A 1 DATA MARKER 0
DGND	22	21	TRIGGER INPUT
Reserved	24	23	EXTERNAL PACER CLOCK INPUT
DGND	26	25	EXTERNAL INTERRUPT INPUT
USER INPUT 1	28	27	USER INPUT 0
USER OUTPUT 1	30	29	USER OUTPUT 0
DGND	32	31	TC OUT 0
EXTERNAL TC GATE 1	34	33	EXTERNAL TC CLOCK 1
TC OUT 2	36	35	TC OUT 1
EXTERNAL TC GATE 2	38	37	EXTERNAL TC CLOCK 2
DGND	40	39	+5 VOLTS

CN9: External Analog I/O Connector

CN9 is an external connector for the analog I/O connections on the board. The pin assignment is listed below.

Table 6: Analog I/O Connector

Digital I/O 1	2	1	Digital I/O 0
AGND	4	3	+5 VOLTS
AOUT 2	6	5	AGND
AOUT 1	8	7	AGND
AIN 8 / AIN8+	10	9	AIN 16 / AIN8-
AIN 7 / AIN7+	12	11	AIN 15 / AIN7-
AIN 6 / AIN6+	14	13	AIN 14 / AIN6-
AIN 5 / AIN5+	16	15	AIN 13 / AIN5-
AINSENSE	18	17	AGND
AIN 4 / AIN4+	20	19	AIN 12 / AIN4-
AIN 3 / AIN3+	22	21	AIN 11 / AIN3-
AIN 2 / AIN2+	24	23	AIN 10 / AIN2-
AIN 1 / AIN1+	26	25	AIN 9 / AIN1-

CN5: SyncBus

The SyncBus, defined by RTD, is a digital bus that enables synchronous operation with other RTD boards. It uses 5V-compliant TTL signaling without pull-up resistors on the bus lines. For any line to function, a master/driver must be present in the system.

Table 7: SyncBus

SyncBus0	1	2	GND
GND	3	4	GND
SyncBus1	5	6	GND
GND	7	8	GND
SyncBus2	9	10	GND

CN18: McBSP

The McBSP (Multichannel Buffered Serial Port), defined by Texas Instruments, is a serial bus for communication between DSPs and front-end devices, operating at 5V compliance. The DAQ board drives the DR and FRS signals and receives the DX, FSX, CLKX, and CLKR signals, which must be directly connected to the corresponding DSP signals. The CLKS signal, included for TI connector compatibility, is not physically connected on the DAQ board. A straight 10-pin cable is required for this connection.

Table 8: McBSP

NC	1	2	GND
CLKR	3	4	FSR
CLKX	5	6	GND
DR	7	8	FSX
DX	9	10	GND

NOTE: CN14 and CN21 are for Factory use only

3.3.2 BUS CONNECTORS

CN1(Top) & CN2(Bottom): PCIe Connector

The PCIe connector is the connection to the system CPU. The position and pin assignments are compliant with the *PCI/104-Express Specification*. (See PC/104 Specifications on page 82)

The SDM35541 is a "Universal" board, and can connect to either a Type 1 or Type 2 PCIe/104 connector.

3.3.3 JUMPERS

JP1: Local Reset

Installing JP1 causes the local bus to reset. This jumper is not installed for normal operations.

3.4 Steps for Installing

1. Always work at an ESD protected workstation and wear a grounded wrist-strap.
2. Turn off power to the PC/104 system or stack.
3. Select and install stand-offs to properly position the module on the stack.
4. Remove the module from its anti-static bag.
5. Check that pins of the bus connector are properly positioned.
6. Check the stacking order; make sure all of the busses used by the peripheral cards are connected to the cpuModule.
7. Hold the module by its edges and orient it so the bus connector pins line up with the matching connector on the stack.
8. Gently and evenly press the module onto the PC/104 stack.
9. If any boards are to be stacked above this module, install them.
10. Attach any necessary cables to the PC/104 stack.
11. Re-connect the power cord and apply power to the stack.
12. Boot the system and verify that all of the hardware is working properly.

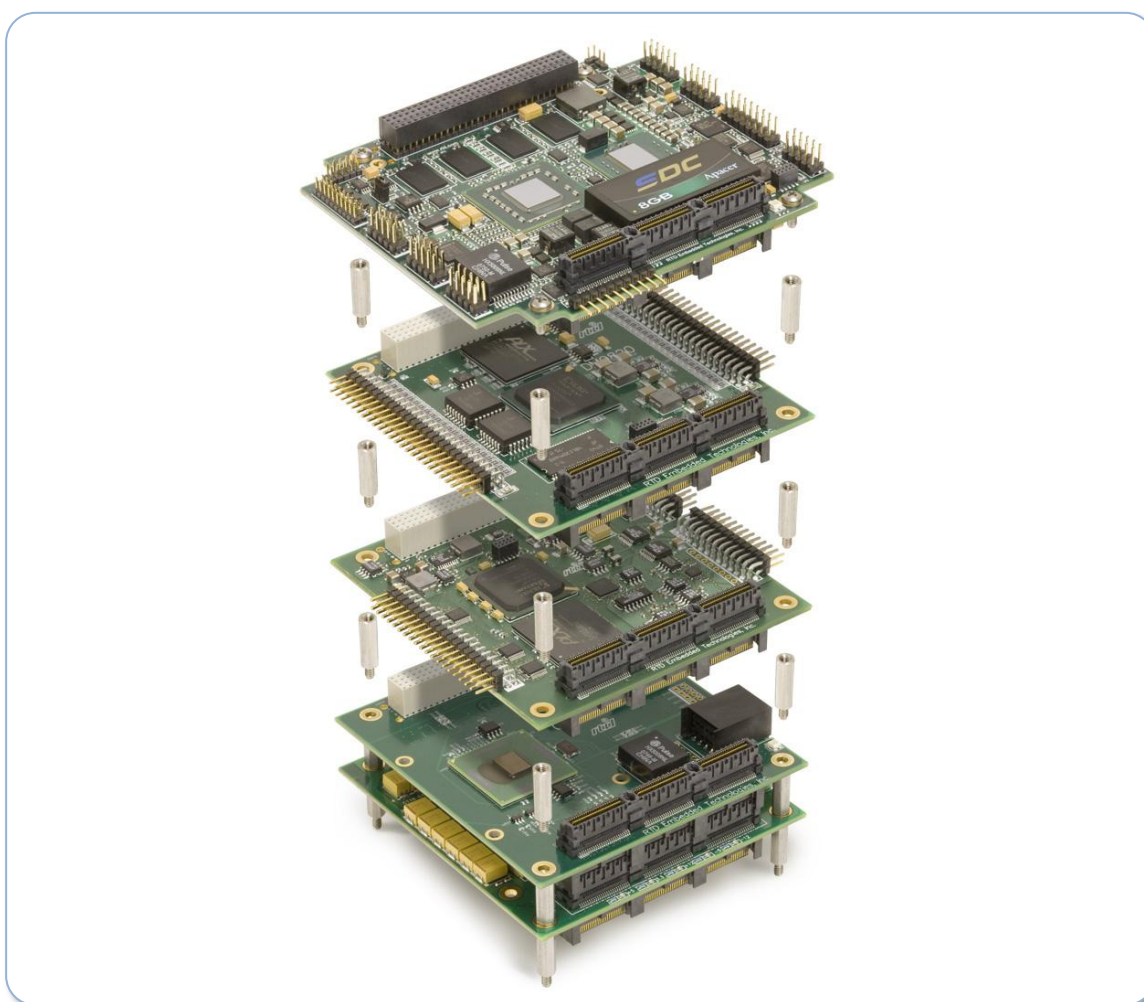


Figure 3: Example 104™ Stack

4 IDAN Connections

4.1 Module Handling Precautions

To prevent damage due to Electrostatic Discharge (ESD), keep your module in its antistatic bag until you are ready to install it into your system. When removing it from the bag, hold the module by the aluminum enclosure, and do not touch the components or connectors. Handle the module in an antistatic environment, and use a grounded workbench for testing and handling of your hardware.

4.2 Physical Characteristics

- Weight: Approximately 0.46 lbs. (0.21 Kg)
- Dimensions:
 - Length: 5.983 inches (151.972 mm)
 - Width: 5.117 inches (129.978 mm)
 - Height: 0.669 inches (16.993 mm)

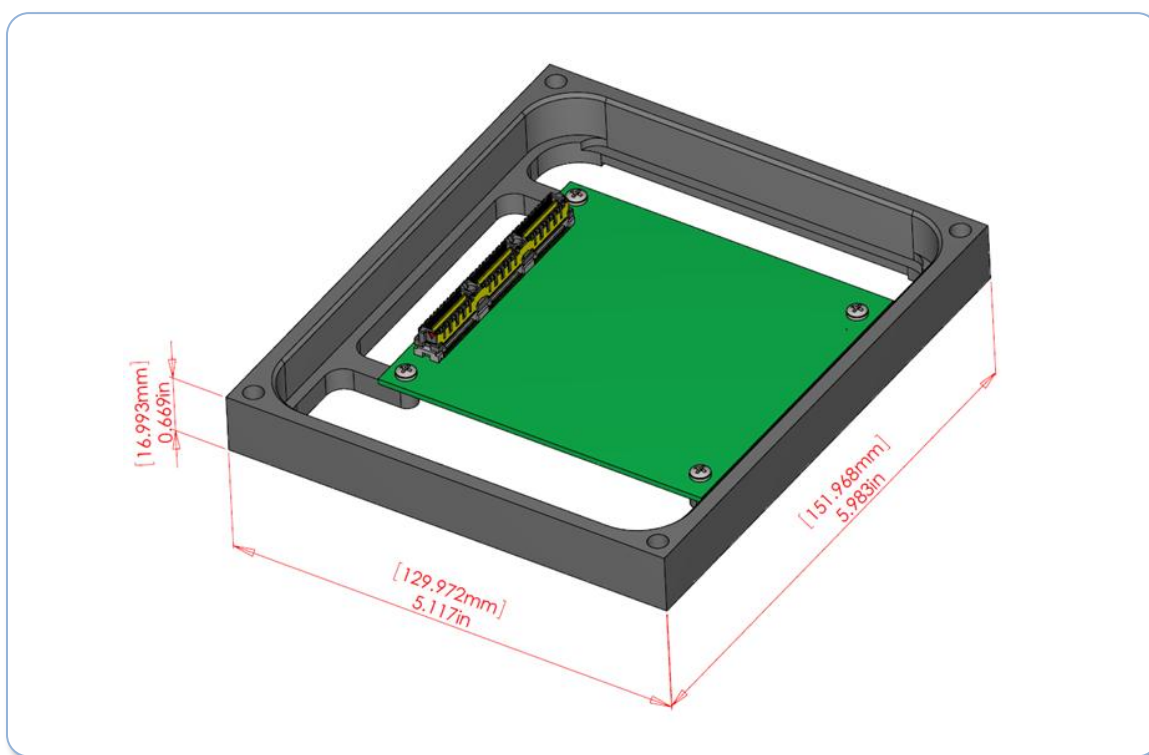


Figure 4: IDAN Dimensions

4.3 Connectors

4.3.1 EXTERNAL I/O CONNECTORS

26 pin "High Density D": female

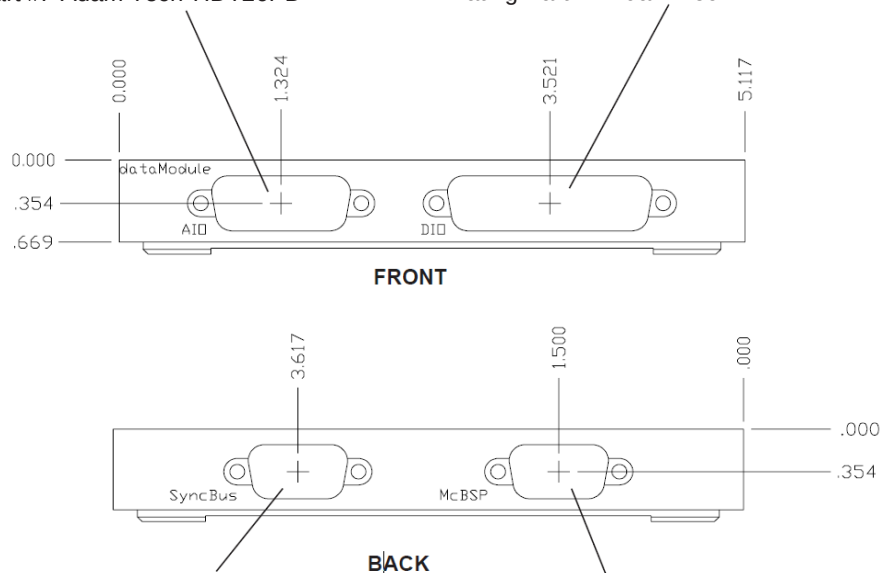
Module Part #: Adam Tech HDT26SD

Mating Part #: Adam Tech HDT26PD

44 pin "High Density D": female

Module Part #: Adam Tech HDT44ST

Mating Part #: Adam Tech HDT44PD



9 pin "D": female

Module Part #: AMP/TYCO 1658610-4

Mating Part #: AMP/TYCO 1658608-4

9 pin "D": male

Module Part #: VALCON DF-9P

Mating Part #: VALCON DF-9S

Note: Drawing is not to scale.

Table 9: IDAN-SDM35541 Analog I/O 26-pin "D" Connector

IDAN Pin#	Signal	SDM35541 CN9 Pin #
1	Digital I/O 0	1
2	Digital I/O 1	2
3	+5 VOLTS	3
4	AGND	4
5	AGND	5
6	AOUT 2	6
7	AGND	7
8	AOUT 1	8
9	AIN 16 / AIN8-	9
10	AIN 8 / AIN8+	10
11	AIN 15 / AIN7-	11
12	AIN 7 / AIN7+	12
13	AIN 14 / AIN6-	13
14	AIN 6 / AIN6+	14
15	AIN 13 / AIN5-	15
16	AIN 5 / AIN5+	16
17	AGND	17
18	AINSENSE	18

IDAN Pin#	Signal	SDM35541 CN9 Pin #
19	AIN 12 / AIN4-	19
20	AIN 4 / AIN4+	20
21	AIN 11 / AIN3-	21
22	AIN 3 / AIN3+	22
23	AIN 10 / AIN2-	23
24	AIN 2 / AIN2+	24
25	AIN 9 / AIN1-	25
26	AIN 1 / AIN1+	26

Table 10: IDAN-SDM35541 Digital I/O 44-pin "D" Connector

IDAN Pin#	Signal	SDM35541 CN6 Pin #
1	DGND	40
2	External TC Clock 2	37
3	External TC Gate 1	34
4	TC Out 0	31
5	User Input 1	28
6	External Interrupt Input	25
7	DGND	22
8	D/A 1 Data Marker 0	19
9	P1.0/DIG Table 0	16
10	High Speed Input 1/P0.1	13
11	P1.6/ DIG Table 6	10
12	High Speed Input 4/P0.4	7
13	P1.6/ DIG Table 6	4
14	High Speed Input 7/P0.7/A/D DM2	1
15	Reserved	-
16	+5V	39
17	TC Out 2	36
18	External TC Clock 1	33
19	User Output 1	30
20	User Input 0	27
21	Reserved	24
22	External Trigger Input	21
23	Reserved	18
24	High Speed Input 0/P0.0	15
25	P1.2/DIG Table 2	12
26	High Speed Input 3/P0.3	9
27	P1.5/DIG Table 5	6
28	High Speed Input 6/P0.6/A/D DM1	3
29	Reserved	-
30	Reserved	-
31	External TC Gate 2	38
32	TC Out 1	35
33	DGND	32
34	User Output 0	29
35	DGND	26

IDAN Pin#	Signal	SDM35541 CN6 Pin #
36	External Pacer Clock Input	23
37	D/A 2 Data Maker 0	20
38	Reserved	17
39	P1.1/DIG Table 1	14
40	High Speed Input 2/P0.2	11
41	P1.4/DIG Table 4	8
42	High Speed Input 5/P0.5/A/D DM0	5
43	P1.7/DIG Table 7	2
44	Reserved	-

Table 11: IDAN-SDM35541 SyncBus 9-pin "D" Connector

IDAN Pin#	Signal	SDM35541 CN5 Pin #
1	SyncBus0	1
2	GND	3
3	SyncBus1	5
4	GND	7
5	SyncBus2	9
6	GND	2
7	GND	4
8	GND	6
9	GND	8

Table 12: IDAN-SDM35541 McBSP 9-pin "D" Connector

IDAN Pin#	Signal	SDM35541 CN18 Pin #
1	NC	1
2	CLKR	3
3	CLKX	5
4	DR	7
5	DX	9
6	GND	2
7	FSR	4
8	GND	6
9	FSX	8

4.4 Steps for Installing

1. Always work at an ESD protected workstation, and wear a grounded wrist-strap.
2. Turn off power to the IDAN system.
3. Remove the module from its anti-static bag.
4. Check that pins of the bus connector are properly positioned.
5. Check the stacking order; make sure all of the busses used by the peripheral cards are connected to the cpuModule.
6. Hold the module by its edges and orient it so the bus connector pins line up with the matching connector on the stack.
7. Gently and evenly press the module onto the IDAN system.
8. If any boards are to be stacked above this module, install them.
9. Finish assembling the IDAN stack by installing screws of an appropriate length.
10. Attach any necessary cables to the IDAN system.
11. Re-connect the power cord and apply power to the stack.
12. Boot the system and verify that all of the hardware is working properly.

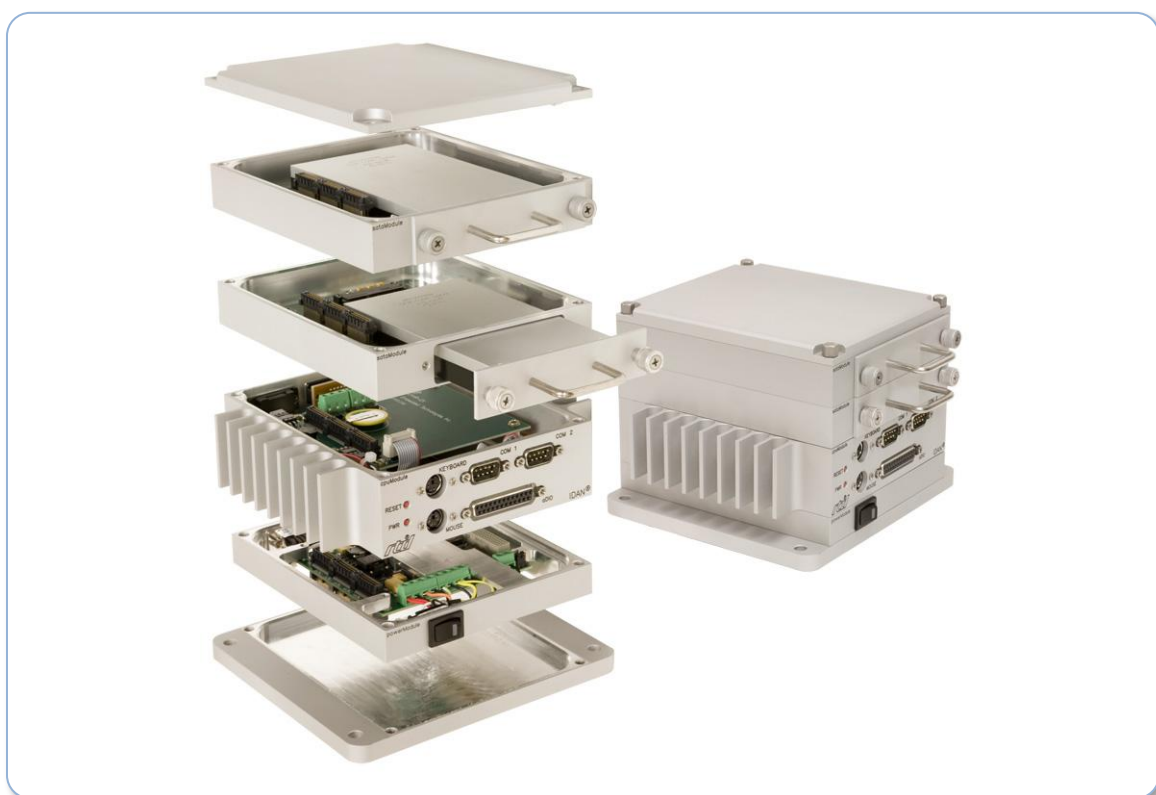


Figure 5: Example IDAN System

5 Functional Description

5.1 Block Diagram

The Figure below shows the functional block diagram of the SDM35541HR. The various parts of the block diagram are discussed in the following sections.

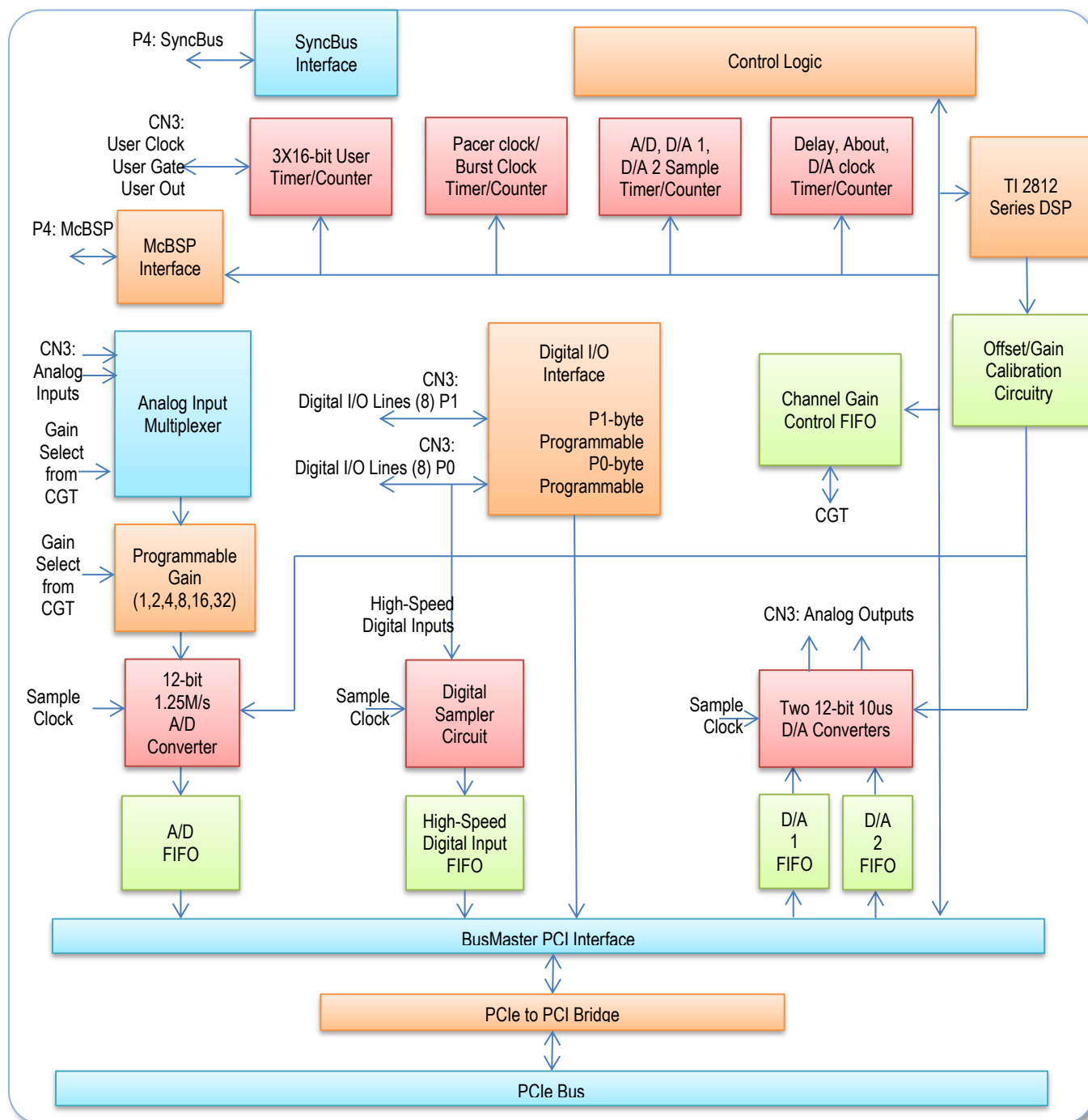


Figure 6: SDM35541HR Block Diagram

5.2 Hardware Description

5.2.1 CHANNEL GAIN LATCH (CGL) AND CHANNEL GAIN TABLE (CGT)

For single-channel operation, configure the Channel Gain Latch mode using the appropriate software command and load the Channel Gain Latch. This mode ensures the highest sampling rate and accuracy and supports the analog trigger function. Select one input channel as the Analog Trigger Input, setting its channel type, number, and gain via software based on the signal source. Monitor the converted data from this channel to detect the analog trigger event. Once detected, you can switch to multi-channel Channel Gain Table mode.

Channel Gain Table Mode:

The Channel Gain Table enables high-speed sampling of channels in any order, with customizable gain for each channel. This 1024 x 24-bit memory supports complex channel-gain scan sequences and digital output control. The digital output control feature allows management of external input expansion boards, such as the TMX32, to increase channel capacity up to 512 channels, with control lines output on Port 1. When not used for expansion, these digital lines can serve other control functions.

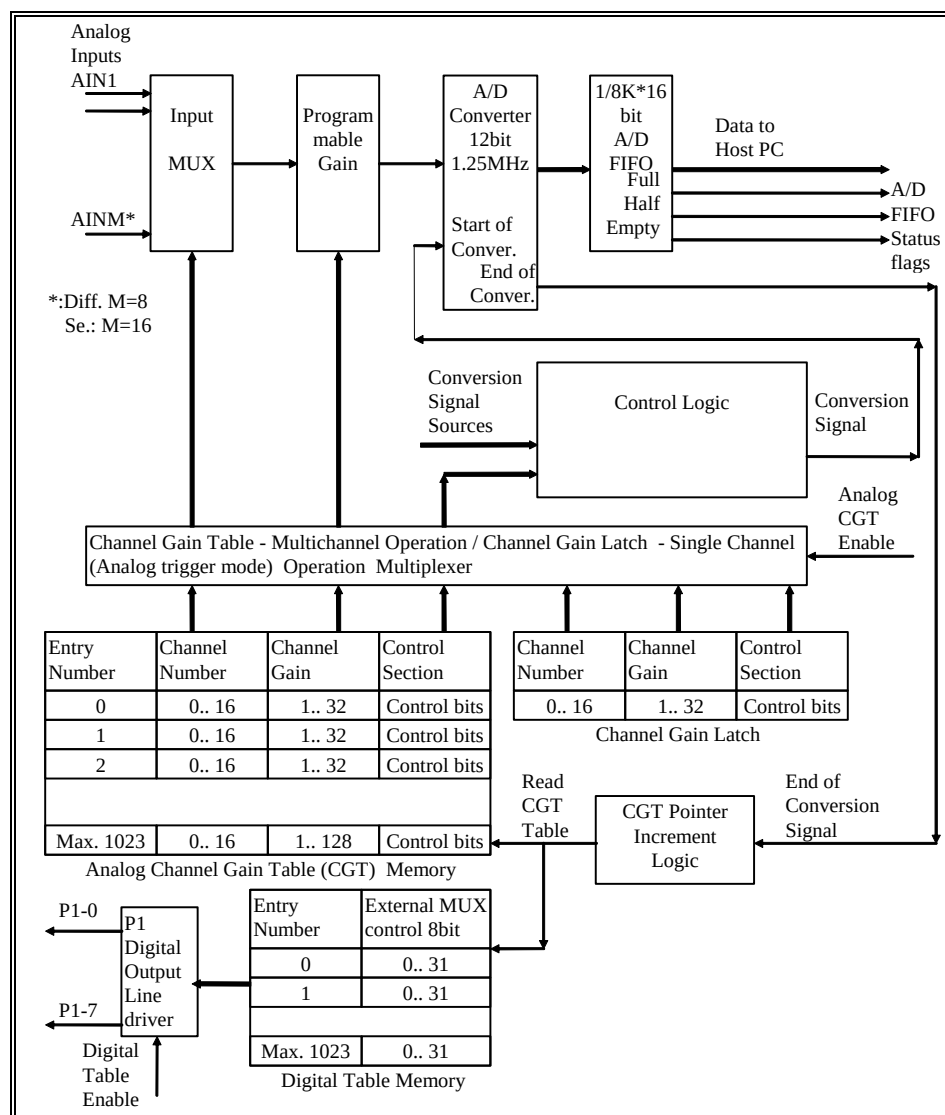
A skip bit in the channel-gain data word allows different sampling rates across channels. When set, this bit triggers an A/D conversion on the selected channel without storing the data in the FIFO.

Multi-Channel Operation:

For multi-channel operation, enable the Channel Gain Table with the appropriate software command. Clear and populate the table with the correct entries using software instructions. After setup, the table's read pointer starts at the first entry. The initial A/D conversion follows this entry. Upon an active Conversion Signal, the A/D Converter generates an End of Conversion Signal, which advances the read pointer to the next entry, stores the converted data in the A/D FIFO, and, if enabled, records High-Speed Digital Input lines in Data Marker Mode to the FIFO. Subsequent conversions follow the next table entries. After the last entry, the read pointer automatically resets to the first entry, or this can be triggered manually with a Reset Channel Gain Table command.

The Channel Gain Table supports independent configuration of channel type (GRSE, NRSE, or DIFF), gain (1 to 32), and input range ($\pm 5V$, $\pm 10V$, or 0–10V). It also enables simultaneous updates of D/A1 and D/A2 with the corresponding input channels, controlled via the table's bit settings.

Figure 7: Channel Gain Table



5.2.2 ANALOG-TO-DIGITAL CONVERSION

The SDM35541 module allows software configuration for each of its up to 16 single-ended or 8 differential analog input channels. Programmable unipolar and bipolar input ranges and gains enable seamless integration with a wide variety of sensors.

A/D Converter

The 12-bit successive approximation A/D converter digitizes dynamic input voltages in 0.8 microseconds, achieving a maximum throughput of 1.25 MHz. This single-chip solution integrates a sample-and-hold amplifier, a 12-bit A/D converter, a 2.5-volt reference, a clock, and a digital interface for complete A/D conversion. Its low-power CMOS logic and high-precision, low-noise design ensure accurate results.

Conversions can be initiated via software commands, a pacer clock, triggers to start or stop sampling, or a sample counter to collect a specific number of samples. The conversion rate can be controlled using either the on-board pacer clock or an external one.

A/D FIFO – Sample Buffer

The 8k-sample First-In, First-Out (FIFO) buffer supports the A/D converter's high throughput by providing flexible storage for converted data. This allows conversions to continue even if the computer cannot read the data as quickly as it is generated, stopping only when the FIFO sends a full flag to halt the converter.

The sample buffer requires no manual addressing; internal mechanisms ensure data is correctly stored and retrieved. All data in the buffer remains intact until the PC completes the transfer. Its asynchronous operation allows data to be written to or read from the buffer at any time and rate. When a transfer occurs, the first data entered into the FIFO is the first to be retrieved.

Data Transfer

Converted data can be transferred to PC memory in three ways: programmed I/O mode, interrupt mode, or using the on-board DMA controller. A specialized interrupt mode, utilizing the REP INS (Repeat Input String) instruction, enables high-speed data transfers. When the FIFO's half-full flag triggers an interrupt, the REP INS instruction executes, rapidly transferring data to PC memory and clearing the sample buffer at the maximum bus speed. The DMA mode provides the fastest burst-mode data transfer as a bus master.

5.2.3 DIGITAL-TO-ANALOG CONVERSION

The digital-to-analog (D/A) circuitry features two independent 12-bit analog output channels, each supporting programmable output ranges of $\pm 5V$, 0 to +5V, $\pm 10V$, or 0 to +10V. Each channel includes a 1,000-sample FIFO buffer to store data prior to output. Users can continuously write data to the buffer to produce a unique, non-repeating waveform or load a fixed dataset to generate a repeating waveform. Data is transferred to the output buffers using memory write instructions. The analog outputs can be updated via software, various clock signals, or triggers, and can operate simultaneously or independently.

5.2.4 TIMER/COUNTERS

The system includes one 8254 programmable 16-bit, 8-MHz interval timer (User TC) and ten internal 16/24-bit timers, offering versatile timing and counting capabilities. The internal timers operate in binary countdown mode.

The 8254 User TC has three independent 16-bit timer/counters, each available for user-defined functions. Each timer/counter features two inputs (CLK and GATE) and one output (timer/counter OUT). The clock and gate input sources for each timer/counter are programmable. You can configure each timer/counter as a binary or BCD down counter by setting the appropriate command word, which also defines the operating mode. The six programmable modes are:

- Mode 0 Event Counter (Interrupt on Terminal Count)
- Mode 1 Hardware-Retriggerable One-Shot
- Mode 2 Rate Generator
- Mode 3 Square Wave Mode
- Mode 4 Software-Triggered Strobe
- Mode 5 Hardware Triggered Strobe (Retriggerable)

5.2.5 DIGITAL I/O

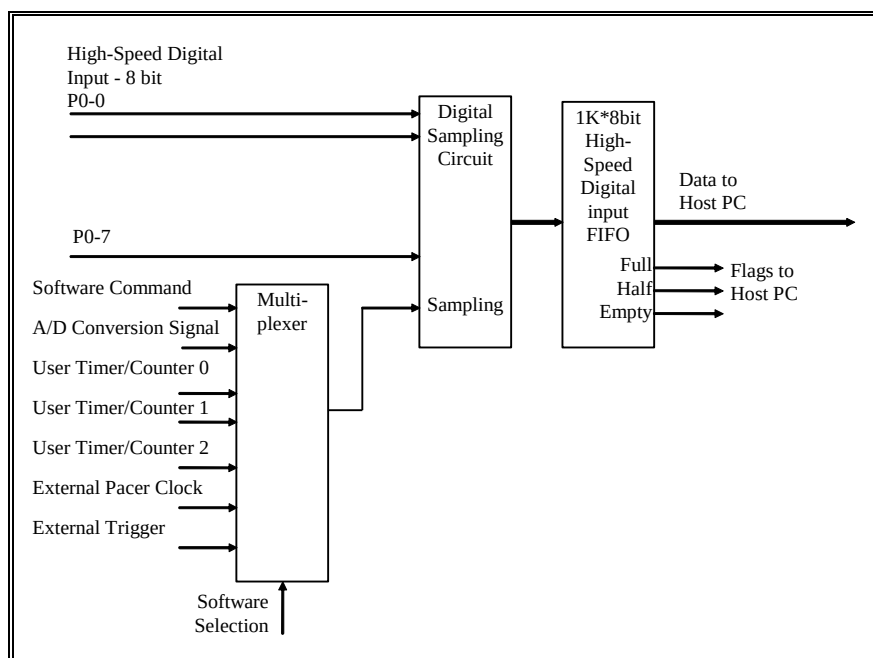
The SDM35541 features 16 buffered TTL/CMOS digital I/O lines, divided into two ports. Port 0 consists of eight independently programmable bits, while Port 1 is an 8-bit programmable port. Port 0 supports two Advanced Digital Interrupt modes: an event interrupt, triggered when any bit changes value, and a match interrupt, triggered when the lines match a programmed value. Both modes allow masking to monitor specific lines. Each line is pulled up with 10k Ω resistors.

5.2.6 HIGH-SPEED DIGITAL INPUTS

The SDM35541 features eight high-speed TTL/CMOS digital input lines, which share the Digital I/O Port 0. These lines are equipped with a 8K sample FIFO buffer for data storage. Each line is pulled up with 10k Ω resistors.

Figure 8, shows the block diagram of High-Speed Digital Input section. The sampling signal can be software selectable.

Figure 8: High-Speed Digital Inputs



Sampled data is automatically stored in the High-Speed Digital Input FIFO buffer. Data can be transferred to PC memory in one of two ways: using programmed I/O mode or interrupt mode, with support from the on-board DMA controller. In interrupt mode, an interrupt can be triggered after a specified amount of data is collected. The User TC1 counter tracks the number of data samples in the FIFO and can also serve as an interrupt source.

5.2.7 SyncBus

The three-line SyncBus assures the possibility of using multiple SDM35541 in one computer synchronously

5.2.8 McBSP MULTI-CHANNEL BUFFERED SERIAL PORT

The McBSP interface (defined by Texas Instruments), with ten lines including grounds, enables the connection of a dspModule (e.g., RTD SPM 6020/6030) to DAQ boards. This provides a direct link between the DSP and the analog world, using the SDM35541 as a front-end board.

5.2.9 DEFAULT STATE OF I/O

At power-up, all I/O lines are set to a tri-state condition. The digital I/O, external trigger, external pacer clock, external interrupt, both external clocks, both external gates, and both user inputs are pulled up to 5V by RTD. These lines remain in this state until the user programs the board to change their state.

5.3 Analog Connections

5.3.1 CONNECTING ANALOG INPUT PINS

The SDM35541 offers versatile input connection options to support various sensors. It allows mixing multiple input modes:

- Ground-Referenced Single-Ended (GRSE)
- Non-Referenced Single-Ended (NRSE)
- Differential (DIFF), available in three configurations:
 - Without ground reference
 - With a dedicated ground
 - With a separate ground reference via a 10 kΩ resistor

The Differential mode with a dedicated ground functions as a single-ended mode but is limited to 8 channels, each with its own dedicated ground pin and ground wire in the cable connecting the board to the signal conditioning card. This mode is particularly useful when signal shielding is critical.

The analog input modes are further explained with text and figures, which include simplified block diagrams of the board's analog input section. The internal logic state variables—NRSEH, ADCDIFFH, INSTGNDH, and AINSENSEH—control the analog input operations, with switches implemented using analog multiplexers.

Ground Referenced Single-Ended (GRSE) Input Mode

The GRSE (Ground-Referenced Single-Ended) mode is recommended for floating signal sources to prevent ground loops. To configure this mode, connect the high side of the input signal to one of the analog input channels (AIN1 through AIN16) and the low side to any ANALOG GND pin on the connector.

Figure 9 illustrates the switch states for this mode:

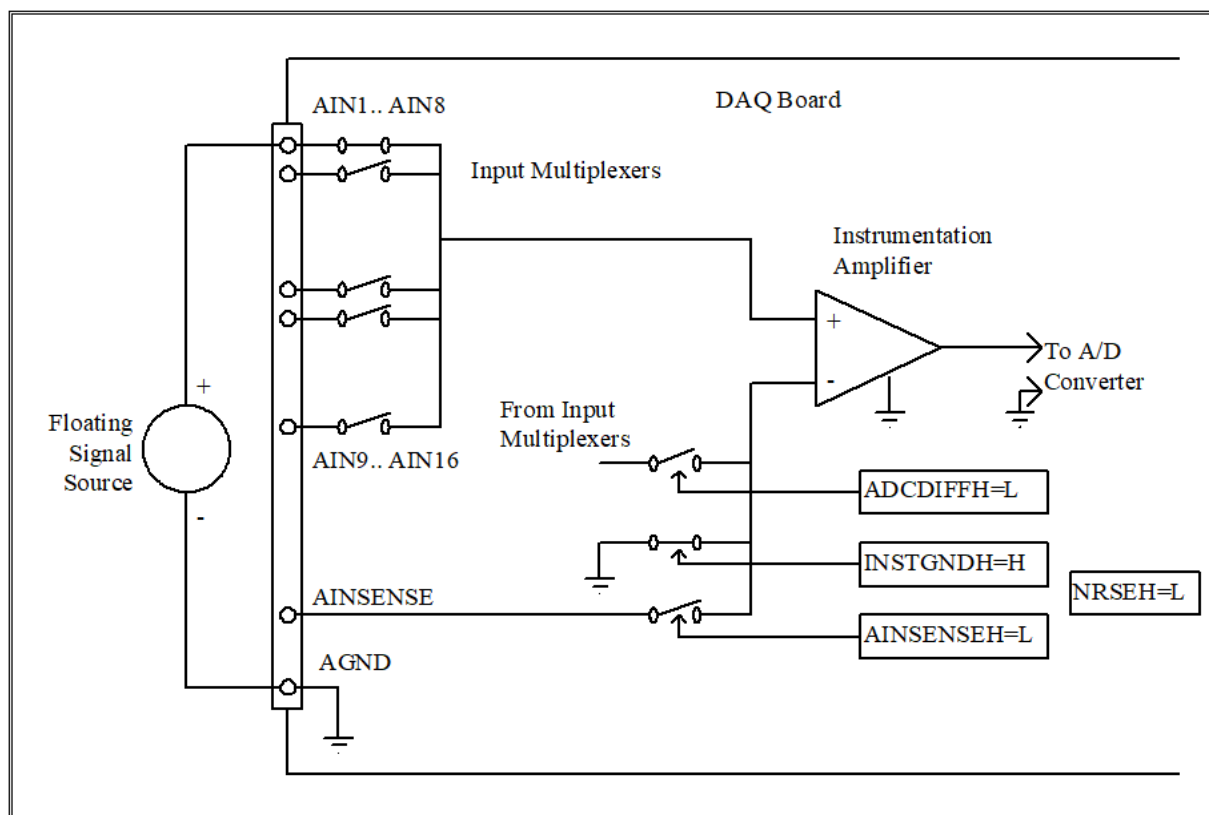
NRSEH bit: Low, indicating this is not NRSE mode.

ADCDIFFH bit: Low, indicating this is not differential mode.

INSTGNDH bit: High, connecting the low side of the Instrumentation Amplifier to Analog Ground (AGND).

AINSENSEH bit: Low, as the Instrumentation Amplifier uses Analog Ground as the reference signal.

Figure 9: Ground Referenced Single Ended input mode



Non-Referenced Single-Ended (NRSE) Input Mode

The NRSE (Non-Referenced Single-Ended) mode is primarily designed for grounded signal sources, as shown in Figure 10, but can also be used with floating sources if an external resistor is added to ground the AINSENSE signal. To configure NRSE mode, connect the high side of the input signal to one of the analog input channels (AIN1 through AIN16) and the low side to the AINSENSE pin on the connector. Figure 10 illustrates the switch states for this mode:

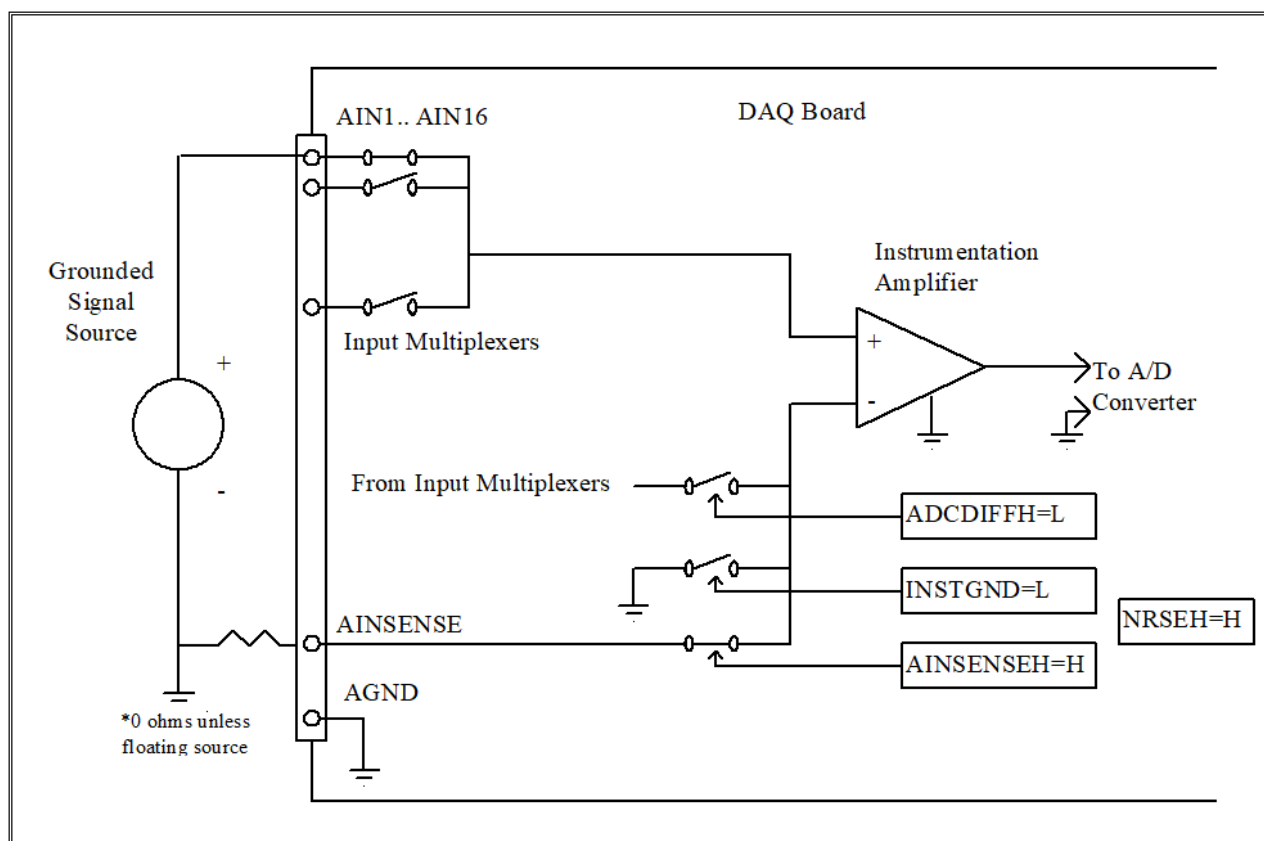
NRSEH bit: High, indicating NRSE mode is active.

ADCDIFFH bit: Low, indicating this is not differential mode.

AINSENSEH bit: High, connecting the low side of the Instrumentation Amplifier to the AINSENSE signal.

INSTGNDH bit: Low, as the Instrumentation Amplifier uses the AINSENSE signal as the reference.

Figure 10: Non-Referenced Single-Ended input mode



Differential (DIFF) Input Mode

The differential input mode supports signal sources with or without a separate ground reference. For sources lacking a separate ground or direct ground connection, a ground reference may be required, which involves adding external components (see Figure 11).

To configure differential mode, connect the high side of the input signal to one of the positive analog input channels (AIN1+ through AIN8+) and the low side to the corresponding negative channel (AIN1- through AIN8-).

Figure 11 shows the switch states for this mode:

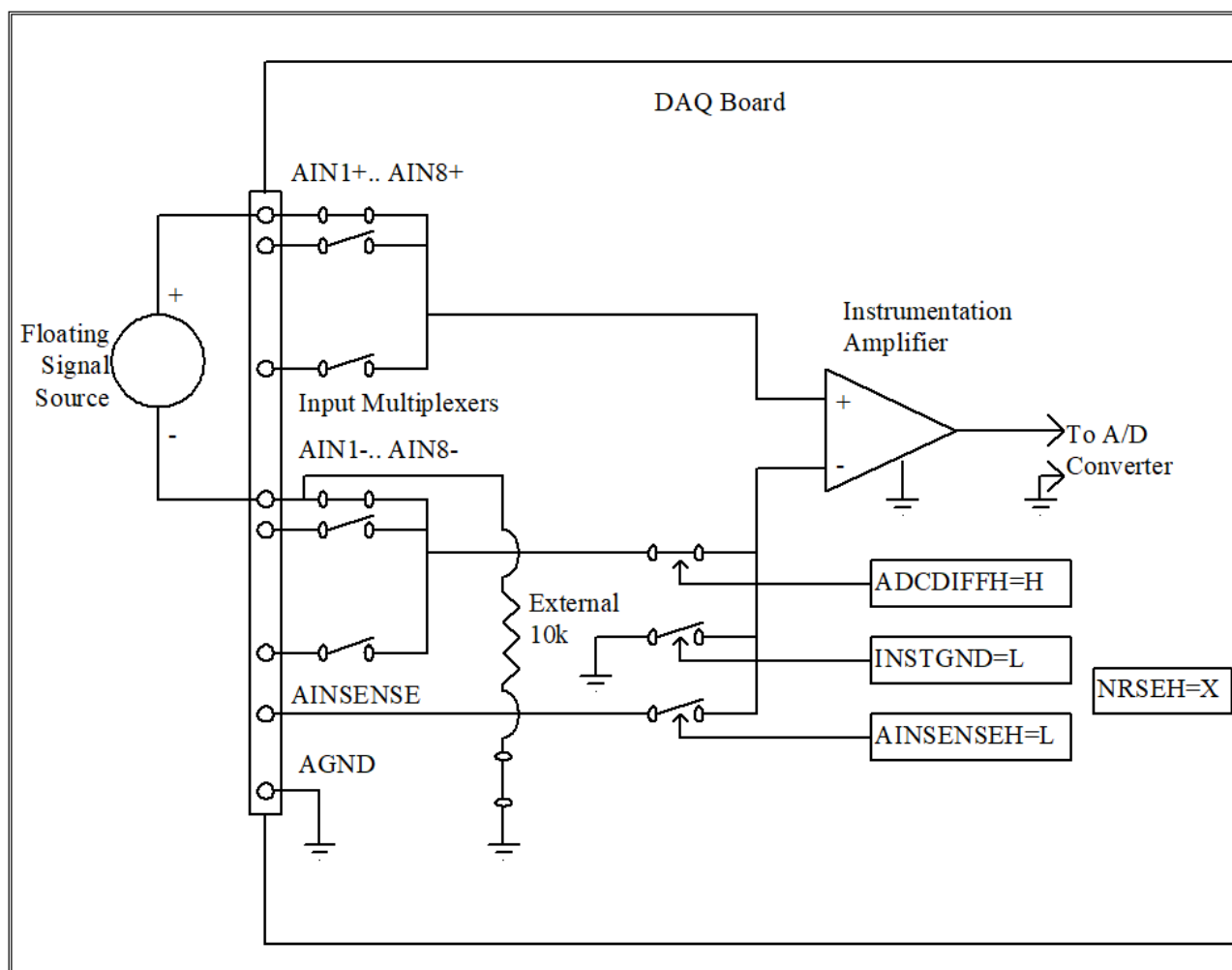
NRSEH bit: Irrelevant in differential mode.

ADCDIFFH bit: High, connecting the low side of the Instrumentation Amplifier to the AIN- signal.

INSTGNDH bit: Low, as the Instrumentation Amplifier uses the AIN- signal as the reference.

AINSENSEH bit: Low, as the AIN- signal serves as the reference for the Instrumentation Amplifier in differential mode.

Figure 11: Differential input mode



5.3.2 CONNECTING ANALOG OUTPUTS

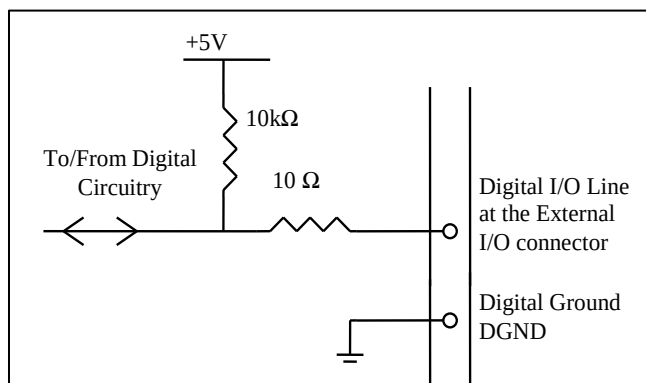
For each D/A output, connect the high side of the receiving device to the AOUT channel and the low side to an ANALOG GND pin.

5.3.3 CONNECTING THE TIMER/COUNTERS AND DIGITAL I/O

For all connections, attach the high side of an external signal source or destination device to the corresponding signal pin on the I/O connector, and connect the low side to any DIGITAL GND pin.

As shown in Figure 12, the termination circuit for digital input/output lines varies. Digital input lines exclude the 10Ω series resistor, while digital output lines exclude the $10k\Omega$ pull-up resistor.

Figure 12: Digital I/O



6 Register Address Space

This product was built as a modular design, which allows each board function to have its own Functional Block (FB). Each functional block is designed to work independently. For this reason, we provide individual DMA channels, interrupts, clocks, and FIFOs to each functional block.

The registers are described by their PCIe Base Address Register (BAR), which is defined in the PCI configuration space for this board. The configuration space is generally handled by the operating system. For more information on how to use the configuration space, consult the *PCI Local Bus Specification, Revision 3.0* from the PCI-SIG.

Register Types

There are several different types of registers that are referred to in this section. A description of each type is below.

- Read/Write Registers: The value that is written to this register can also be read back.
- Maskable Registers: This is a 32 bit register that consists of 16-bit data field in the upper word and a 16-bit mask value in the lower word. For each bit in the data field, it is only written to the register if the corresponding bit in the mask field is '1'.
- Sticky Registers: This is a status read register. When a bit in this register has a value of '1', a '1' needs written to that bit to reset the register to '0'. This is typically used for interrupt status registers.
- Read Only: This register can only be read.



NOTE: Writing to Read-Only registers may have unexpected results.

Function Block Mapping

Below is the Function Block slot mapping for the SDM35541HR.

Table 13: Function Block Mapping

Slot	ID	Description
FB0	0x01021000	SDM35xx ADC/DAC
FB15	0x00000001	Flash Programmer

6.1 BAR0: General Board Control

This section contains the board-level interrupt control, as well as a description of all the Functional Blocks in BAR2. Most of the registers in this section are Read-Only. Although a full implementation requires BAR0 to be 0x400 in length, most implementations will have a shorter length.

Table 14: Base Functional Block

Offset	0x03	0x02	0x01	0x00
0x00	GBC_BRD_RST	GBC_EOI	GBC_REV	GBC_FMT
0x04	GBC_PDP			
0x08	GBC_BUILD			
0x0C	Reserved	GBC_USER_ID	GBC_SYS_CLK_FREQ	
0x10	GBC_IRQ_STATUS			
0x14				
0x18	GBC_DIRQ_STATUS			
0x1C				
0x20	FB0_ID			
0x24	FB0_OFFSET			
0x28	FB0_OFFSET_DMA			
0x2C	reserved			
0x30	FB1_ID			
0x34	FB1_OFFSET			
0x38	FB1_OFFSET_DMA			
0x3C	reserved			
...				
0x20+0x10*n	FBn_ID			
0x24+0x10*n	FBn_OFFSET			
0x28+0x10*n	FBn_OFFSET_DMA			
0x2C+0x10*n	reserved			
...				
0x3F0	FB61_ID			
0x3F4	FB61_OFFSET			
0x3F8	FB61_OFFSET_DMA			
0x3FC	reserved			

6.1.1 GBC_FMT (READ-ONLY)

This register contains the format ID that is used in this board.

- B[7:4] Major Format – Changes to this value indicate changes that are not backward compatible.
 - 0 – Initial Major Format
- B[3:0] Minor Format – Changes to this value indicate additional features that are backward compatible.
 - 1 – Initial Minor Format
 - 2 – Addition of FB_DMAm_CTRLn[Pause] flag and Early DMA Buffer Termination.

6.1.2 GBC_REV (READ-ONLY)

This register contains the PDP revision for this board. A=1, B=2, etc.

6.1.3 GBC_BRD_RST (READ/WRITE)

Write a value of 0xAA to this register to reset the board.

6.1.4 GBC_PDP (READ-ONLY)

This register contains the PDP number for this board in decimal.

6.1.5 GBC_BUILD (READ-ONLY)

Unique 32-bit build number for every FPGA build.

6.1.6 GBC_SYS_CLK_FREQ (READ ONLY)

The measured frequency of the system clock. Units are 10 kHz, i.e. (Frequency in Hertz) = (GBC_SYS_CLK_FREQ * 10 kHz). This is measured once after any reset, including GBC_BRD_RST. This value is not available (will read 0) until 100us after reset.

6.1.7 GBC_USER_ID (READ ONLY)

This value is set by jumpers on the board, and is used to identify individual board when there are multiple in the same system.

6.1.8 GBC_IRQ_STATUS (READ/CLEAR)

This is a 64-bit interrupt status register for non-DMA interrupts. Each bit in this register corresponds to one of the Function Blocks; bit 0 corresponds to FB0 (whose ID and OFFSET are at 0x020), etc. Bits 60 through 63 are reserved. This is a "sticky" register, and the driver clears it by writing a '1' to the appropriate bit. The interrupts signaled by this registered are generally handled by a User Mode Delayed Procedure Call. Any status information required to service the interrupt must be latched in the Function Block. This typically means that there will also be a sticky status register in the Function Block.

The Interrupt Service Routine must be able to clear the interrupt without accessing any registers in the Function Block. This means that if a persistent interrupt condition occurs (i.e. FIFO Empty) the GBC_IRQ_STATUS register will initially be set. The ISR will then clear the bit in GBC_IRQ_STATUS and launch the DPC. In this state the interrupt condition still exists, but the status bit is cleared (because we are servicing it). Depending on implementation, that Function Block may not be able to issue another interrupt while in this state. In this state the full status of the board cannot be determined just from the GBC_IRQ_STATUS register. The Function Block should also maintain a status register that will indicate if the interrupt condition still exists, which will be cleared by the DPC. By doing an End-Of-Interrupt (see Section 6.1.10 GBC_EOI (Read/Clear) below) any "hidden" interrupts will be generated again.

6.1.9 GBC_DIRQ_STATUS (READ/CLEAR)

This is a 64-bit interrupt status register for DMA interrupts. Each bit in this register corresponds to one of the Function Blocks; bit 0 corresponds to FB0 (whose ID and OFFSET are at 0x020), etc. Bits 60 through 63 are reserved. This is a "sticky" register, and the driver clears it by writing a '1' to the appropriate bit. The interrupts signaled by this registered are generally handled by the Kernel Mode ISR. This interrupt is only for the Function Block Standard DMA in Section 6.2 on page 36.

6.1.10 GBC_EOI (READ/CLEAR)

This register is used to acknowledge an interrupt. It is used to safeguard against missing an interrupt. At the end of the ISR, write a 0x01 to this register. If there is another interrupt pending in the status registers, the interrupt line is toggled (Legacy Mode), or another interrupt is sent (MSI Mode).

6.1.11 FBN_ID (READ-ONLY)

This is a 32-bit value that identifies the type of Function Block in slot 'n'. Any changes to a Function Block result in a new Block ID. The mappings for the Functional Block ID can be found on page 33. The ID consists of three fields:

- B[31:24] Type Revision – This is used to indicate the addition of features to the function block. Any checks in the software should be of the form (Type Revision > NN) so that the driver doesn't fail with a newer FPGA. Any changes to the FPGA that would break existing software must change the Type, not the Type Revision.
- B[23:16] Subtype – A Subtype is a variant of the Type with one of the following differences. This allows the driver and library to use the same functions, but have different enumerations.
- B[15:0] Type – The Type of the Function Block is used to define the register map. Any time that the positions of registers change, or registers are added or removed require a new Type number. This is loosely separated into
 - B[15:8] Family – i.e. ADC, DAC, Digital I/O, etc.
 - B[7:0] Member

6.1.12 FBN_OFFSET (READ-ONLY)

This is the offset from the beginning of the Functional Block section (typically the offset in BAR2) that this Functional Block resides in.

6.1.13 FBN_OFFSET_DMA (READ-ONLY)

This is the offset from the beginning of the Functional Block section (typically the offset in BAR2) that the Functional Block DMA Registers resides in.

6.2 BAR2: Functional Block Standard DMA

This section describes a standard DMA engine implementation that may be used by the Functional Blocks. Any Functional Block that uses DMA shall use this implementation so that the kernel mode driver may be used to service it, resulting in better performance.

Table 15: DMA Registers

Offset	0x03	0x02	0x01	0x00
D + 0x00	FB_DMAm_Stat_Underflow	FB_DMAm_Stat_Overflow	FB_DMAm_Setup	FB_DMAm_Action
D + 0x04	FB_DMAm_Current_Buffer	FB_DMAm_Count		
D + 0x08	FB_DMAm_RD_FIFO_CNT		FB_DMAm_WR_FIFO_CNT	
D + 0x0C	FB_DMAm_Last_Action	FB_DMAm_Stat_Complete	FB_DMAm_Stat_Invalid	FB_DMAm_Stat_Used
D + 0x10	FB_DMAm_CTRL0	FB_DMAm_STAT0	Reserved	
D + 0x14	Reserved	FB_DMAm_SIZE0		
D + 0x18	FB_DMAm_ADDRESS0			
D + 0x1C				
D + 0x20	FB_DMAm_CTRL1	FB_DMAm_STAT1	Reserved	
D + 0x24	Reserved	FB_DMAm_SIZE1		
D + 0x28	FB_DMAm_ADDRESS1			
D + 0x2C				
	...			
D + 0x10 + (0x10 * n)	FB_DMAm_CTRLn	FB_DMAm_STATn	Reserved	
D + 0x14 + (0x10 * n)	Reserved	FB_DMAm_SIZEn		
D + 0x18 + (0x10 * n)	FB_DMAm_ADDRESsn			
D + 0x1C + (0x10 * n)				

6.2.1 FB_DMAm_ACTION (READ/WRITE)

After writing to the Action register, the user should poll the Last_Action register (below) until it reads the same value. This shows that the action has been performed by the DMA state machine. This is especially important when entering and exiting the Clear state.

- 0x00 = Clear: Clear the Current Buffer field, the internal offset counters, and the FIFO. DMA is stopped.
- 0x01 = Go: Starts DMA
- 0x02 = Pause: DMA transfers are stopped, but all internal registers maintain their state. During PAUSE you will still receive Stat_Underflow and Stat_Overflow interrupts. After PAUSE, you may transition to GO or CLEAR.
- 0x03 = Halt: Buffer has been filled that has the HALT bit set, or has attempted to use a buffer with the [Valid](#) bit cleared. After HALT, you must transition to CLEAR.

6.2.2 FB_DMAm_LAST_ACTION (READ/WRITE)

The DMA Engine writes the value of FB_DMAm_Action to this register after it has completed this channel. This indicates to the user that the last command has been processed. It specifically aids the transition to Clear. When transitioning to Clear, the user should wait until FB_DMAm_Last_Action indicates that the Clear has been processed before allowing any other Action changes.

The user may also write a value to this register and then poll the register to see when the value changes. This method can be used to detect when the DMA engine services the channel without an Action change.

6.2.3 FB_DMAm_SETUP (READ/WRITE)

- B0: IntEna: Set to '1' to enable the DMA engine to generate interrupts on completion of a buffer.
- B1: ErrIntEna: Set to '1' to enable the DMA engine to generate interrupts on error.
- B2: Direction (RO or R/W): Set to '1' to transfer from the FPGA to the PCI bus. Clear to '0' to transfer from the PCI bus to the FPGA.
- B3: IgnoreUsed: Set to '1' to prevent an error condition when accessing a buffer with the [Used](#) bit set. When "IgnoreUsed" is set, the hardware will still set the Used_Desc bit (below) if the next buffer has the Used bit set. However, it will not generate an interrupt, and will not PAUSE the DMA engine. Examples are continuous output from a DAC, or very large Pre-trigger buffering using system memory.

6.2.4 FB_DMAM_STAT_USED (READ/WRITE)

This register is used to determine the source of a DMA interrupt. The bits are cleared by writing 0x00 to the byte. Stat_Used will be set regardless of having ErrIntEna set to '1', and regardless of the state of the IgnoreUsed bit (above).

- B0: Used_Desc (R/C). Set to '1' by the DMA engine if it attempting to use a descriptor with the [Used](#) bit set.

6.2.5 FB_DMAM_STAT_INVALID (READ/WRITE)

This register is used to determine the source of a DMA interrupt. The bits are cleared by writing 0x00 to the byte. Stat_Invalid will be set regardless of having ErrIntEna set to '1'.

- B0: Invalid_Desc (R/C). Set to '1' by the DMA engine if it attempting to use a descriptor with the [Valid](#) bit cleared.

6.2.6 FB_DMAM_STAT_OVERFLOW (READ/WRITE)

This register is used to determine the source of a DMA interrupt. The bits are cleared by writing 0x00 to the byte. Stat_Overflow will be set regardless of having ErrIntEna set to '1'.

- B0: Overflow (R/C). Set to '1' by the DMA engine if an overflow occurred on the FIFO.

6.2.7 FB_DMAM_STAT_UNDERFLOW (READ/WRITE)

This register is used to determine the source of a DMA interrupt. The bits are cleared by writing 0x00 to the byte. Stat_Underflow will be set regardless of having ErrIntEna set to '1'.

- B0: Underflow (R/C). Set to '1' by the DMA engine if an underflow occurred on the FIFO.

6.2.8 FB_DMAM_STAT_COMPLETE (READ/WRITE)

This register is used to determine the source of a DMA interrupt. The bits are cleared by writing 0x00 to the byte.

- B0: Buffer_Complete (R/C). Set to '1' by the DMA engine when a buffer is filled that has the [Interrupt](#) bit set.

6.2.9 FB_DMAM_CURRENT_BUFFER (READ-ONLY)

This is the ID for the buffer that will be used for the next access. The user may use this to track the progress of the DMA activity.

6.2.10 FB_DMAM_COUNT (READ-ONLY)

This is the offset in the DMA buffer for the next access. The user may use this to track the progress of the DMA activity.

6.2.11 FB_DMAM_RD_FIFO_CNT(READ-ONLY)

- B[9:0] This is the amount of data available in the read FIFO in bytes. Software can use this to determine when the FIFO is empty. A value of 0x3FC indicates that there are 1020 or more bytes available.
- B15: RD_EMPTY- '1' indicates that the read FIFO is empty

6.2.12 FB_DMAM_WR_FIFO_CNT(READ-ONLY)

- B[9:0] This is the amount of space available in the write FIFO in bytes. Software can use this to determine when the FIFO is empty. A value of 0x3FC indicated that there are 1020 or more bytes available.
- B15: WR_FULL- '1' indicates that the read FIFO is full

6.2.13 FB_DMAM_ADDRESSn (READ/WRITE)

This is the 64-bit PCI address for DMA Channel m, buffer n. It must be double-word aligned (i.e. b[1:0] are reserved).

6.2.14 FB_DMAM_SIZEn (READ/WRITE)

This is the size in bytes of the buffer for DMA Channel m, buffer n. It must be an integer number of double-words (i.e. b[1:0] are reserved). The actual size is FB_DMAM_SIZEn + 4 Bytes. The maximum buffer size is 16MB.

6.2.15 FB_DMAM_CTRLN (READ/WRITE)

- B0: Valid: Sets this bit to '1' to indicate that this contains valid information. The DMA engine will set the error bit and halt if it is ready to use this descriptor and it is not valid.
- B1: Halt: Sets this bit to '1' to halt the DMA engine after this buffer is full.
- B2: Loop: Sets this bit to '1' to start back at descriptor 0 after this buffer is full. This has a higher priority than the HALT bit.
- B3: Interrupt: Sets this bit to '1' to generate an interrupt after this buffer is full.
- B4: Pause: Sets this bit to '1' to pause the DMA engine after this buffer is full.

If the last buffer is reached, and the HALT and LOOP bits are both '0', the DMA engine will loop.

If the last buffer is reached, and the HALT and LOOP bits are both '1', the DMA engine will halt and the Current_Buffer will be set to 0.

6.2.16 FB_DMAM_STATN (READ/CLEAR)

- B0: Used (R/C): DMA engine sets to '1' to indicate that it has completely used this descriptor. The user must clear this bit when it is ready to be used again. The DMA engine will set the error bit and PAUSE if it is ready to use this descriptor and the Used bit is set, unless the [IgnoreUsed](#) bit is set. The bits are cleared by writing 0x00 to the byte.
- B1: Terminated (R/C): DMA engine sets to '1' to indicate that this buffer was terminated early at the request of the associated Function Block.

6.3 BAR2: Functional Block Header Registers

Each Functional Block has a standard header as follows.

Table 16: Functional Block Header

Offset	0x03	0x02	0x01	0x00
FB + 0x00	FB_ID			
FB + 0x04	FB_DMA_BUFFERS	FB_DMA_CHANNELS	FB_CONTROL_SIZE	FB_CHANNEL_SIZE

6.3.1 FB_ID (READ-ONLY)

This is a repeat of the ID from the General Board Control section. This can be used to validate that the correct Function Block is being accessed.

6.3.2 FB_DMA_CHANNELS (READ -ONLY)

This register contains the number of DMA Channels in this Function Block.

6.3.3 FB_DMA_BUFFERS (READ-ONLY)

This register contains the number of Buffer Descriptors in each DMA Channel.

6.3.4 FB_CONTROL_SIZE (READ -ONLY)

This register contains the total number of DWORD registers in the control section of the function block. This should be used to calculate the offset for each register in a channel.



NOTE: This register is not implemented on every function block. You must check each function block's register map to see if this register is defined.

6.3.5 FB_CHANNEL_SIZE (READ-ONLY)

This register contains the total number of DWORD registers in each channel of the function block. This should be used to calculate the offset for each register in a channel.



NOTE: This register is not implemented on every function block. You must check each function block's register map to see if this register is defined.

6.4 BAR2: SDM355xx Functional Block

This Function Block is used to mimic the operation of the SDM355xx data modules. The register offset have change between the SDM35541 and SDM35540, since the module no longer feature a PLX bridge and the memory mapping is completely inside the FPGA.

6.4.1 FUNCTION BLOCK REGISTER MAP

Table 17: SDM35xx Functional Block

	Offset	0x03	0x02	0x01	0x00
Header	FB + 0x00	FB_ID			
	FB + 0x04	FB_DMA_BUFFERS	FB_DMA_CHANNELS	Reserved	Reserved
Runtime Registers	FB + 0x08	Reserved			
	FB + 0x0C	Reserved			
	FB + 0x10	USER_IO			
	FB + 0x14	DAC_CLK_ST (START/STOP)			
	FB + 0x18	FIFO_STATUS (FIFO Status / ADC Start)			
	FB + 0x1C	DAC1_UPDATE			
	FB + 0x20	DAC2_UPDATE			
	FB + 0x24	Reserved			
	FB + 0x28	Reserved			
	FB + 0x2C	DAC_UPDATE			
	FB + 0x30	PACER (START/STOP)			
	FB + 0x34	TIMER (Timer Counter Status / HSDIN Sample)			
	FB + 0x38	INT (Interrupt Status / Enable Mask)			
	FB + 0x3C	CLEAR_INT (Interrupt Mask Set/Clear)			
	FB + 0x40	IT_OVERRUN (Interrupt Overrun / Clear Overrun)			
	FB + 0x44	Reserved			
	FB + 0x48	PCLK_CNT			
	FB + 0x4C	BURST_CNT			
	FB + 0x50	ADC_SAMPLE			
	FB + 0x54	DAC1_SAMPLE			
	FB + 0x58	DAC2_SAMPLE			
	FB + 0x5C	DELAY_CNT			
	FB + 0x60	BURST_CNT			
	FB + 0x64	DAC_CLK			
	FB + 0x68	UTC0			
	FB + 0x6C	UTC1			
	FB + 0x70	UTC2			
	FB + 0x74	UTC_CTRL			
	FB + 0x78	DIO0			
	FB + 0x7C	DIO1			
	FB + 0x80	DIO_CTRL (IRQ Status, P0 DIR, P0 MASK, P0 CMP)			
	FB + 0x84	DIO_STATUS			
	FB + 0xB0	DSP (DSP Command)			
	FB + 0xC0	I2C_ADDR			
	FB + 0xC4	I2C_PTR			
	FB + 0xC8	I2C_DATA			
	FB + 0xCC	I2C_GO			
	FB + 0xD0	I2C_READ			
	FB + 0xE0	ALGDIO_MASK			
	FB + 0xE4	ALGDIO_DATA			
	FB + 0xE8	ALGDIO_DIR			
	FB + 0xEC	ALGDIO_INT			

Table 17: SDM35xx Functional Block

	Offset	0x03	0x02	0x01	0x00
Setup Registers	FB + 0x100	BOARD_RESET			
	FB + 0x104	DMA0_SRC			
	FB + 0x108	DMA1_SRC			
	FB + 0x10C	ADC_CONV			
	FB + 0x110	BURST_START			
	FB + 0x114	PACER_START			
	FB + 0x118	PACER_STOP			
	FB + 0x11C	ACNT_ENABLE			
	FB + 0x120	PACER_MODE			
	FB + 0x124	HSDIN_START			
	FB + 0x128	HSDIN_FIFO_CLR			
	FB + 0x12C	ADC_FIFO_CLR			
	FB + 0x130	CGT_WRITE			
	FB + 0x134	CGT_LATCH			
	FB + 0x138	DT_WRITE			
	FB + 0x13C	CGT_ENABLE			
	FB + 0x140	DT_ENABLE			
	FB + 0x144	PAUSE_TABLE			
	FB + 0x148	CGT_RESET			
	FB + 0x14C	CGT_CLEAR			
	FB + 0x150	DAC1_RANGE			
	FB + 0x154	DAC1_SRC			
	FB + 0x158	DAC1_CYCLE			
	FB + 0x15C	DAC1_RESET			
	FB + 0x160	DAC1_CLEAR			
	FB + 0x164	DAC2_RANGE			
	FB + 0x168	DAC2_SRC			
	FB + 0x16C	DAC2_CYCLE			
	FB + 0x170	DAC2_RESET			
	FB + 0x174	DAC2_CLEAR			
	FB + 0x178	ADC_SCNT_SRC			
	FB + 0x17C	Reserved			
	FB + 0x180	PACER_SELECT			
	FB + 0x184	SBUS0_SOURCE			
	FB + 0x188	SBUS0_ENABLE			
	FB + 0x18C	SBUS1_SOURCE			
	FB + 0x190	SBUS1_ENABLE			
	FB + 0x194	Reserved			
	FB + 0x198	SBUS2_SOURCE			
	FB + 0x19C	SBUS2_ENABLE			
	FB + 0x1A0	Reserved			
	FB + 0x1A4	ETRG_POLARITY			
	FB + 0x1A8	EINT_POLARITY			
	FB + 0x1AC	UTC0_CLOCK			
	FB + 0x1B0	UTC0_GATE			
	FB + 0x1B4	UTC1_CLOCK			
	FB + 0x1B8	UTC1_GATE			
	FB + 0x1BC	UTC2_CLOCK			
	FB + 0x1C0	UTC2_GATE			
	FB + 0x1C4	UIO0_SELECT			
	FB + 0x1C8	UIO1_SELECT			
	FB + 0x1CC	DMA_RSTRQST0			
	FB + 0x1D0	DMA_RSTRQST1			
	FB + 0x1D4	DAC_CLK_START			
	FB + 0x1D8	DAC_CLK_STOP			
	FB + 0x1DC	PCLK_FREQ			
	FB + 0x1E0	BCLK_FREQ			
	FB + 0x1E4	DAC_CLK_SOURCE			
	FB + 0x1E8	DAC_CLK_MODE			
	FB + 0x1EC	MCBSP_AD_CTRL			

Table 17: SDM35xx Functional Block

	Offset	0x03	0x02	0x01	0x00
	FB + 0x1F0	MCBSP_DA_CTRL			
	FB + 0x1F4	Reserved			
FIFO	FB + 0x200	HSDIN_FIFO		ADC_FIFO	
	FB + 0x204	DAC2_FIFO		DAC1_FIFO	
	FB + 0x208	Reserved		DMA_CH1_FIFO	DMA_CH0_FIFO
DSP SRAM	FB + 0x300	DSP_SRAM_ADDRESS0			
	FB + 0x304	DSP_SRAM_ADDRESS1			
	FB + 0x308	DSP_SRAM_ADDRESS2			
	FB + 0x30C	DSP_SRAM_ADDRESS3			

6.4.2 USER_IO (READ/WRITE)

Read Operation

A read provides the User Input 0 and User Input 1 bits as below. These digital input lines come from the External I/O connector. The User Input bits are sampled by the read instruction.

31 – 2	1	0
--------	---	---

B0: User Input 0 state
 B1: User Input 1 state
 B[31-8]: Reserved

Write operation

These bits go to the External I/O Connector of the board. If the source of the User Output x is set to the Software Programmable state by the 0x070E and 0x070F Functions, the state of the User Output bits can be programmed by this write operation.

B0: User Input 0 state
 B1: User Input 1 state
 B[31-8]: Reserved

6.4.3 DAC_CLK_ST (READ/WRITE)

Read Operation

Reading this register starts the DAC Clock. The return value has no meaning.

Write Operation

Writing to this register stop the DAC clock. The value doesn't matter

6.4.4 FIFO_STATUS (READ/WRITE)

Read Operation

A read provides the status bits of the FIFO as below.

31 – 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
---------	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

B0: 0 D/A1 FIFO empty; 1 D/A1 FIFO not empty
 B1: 0 D/A1 FIFO not half empty; 1 D/A1 FIFO half empty
 B2: 0 D/A1 FIFO full; 1 D/A1 FIFO not full
 B3: Reserved
 B4: 0 D/A2 FIFO empty; 1 D/A2 FIFO not empty
 B5: 0 D/A2 FIFO not half empty; 1 D/A2 FIFO half empty
 B6: 0 D/A2 FIFO full; 1 D/A2 FIFO not full
 B7: Reserved
 B8: 0 A/D FIFO empty; 1 A/D FIFO not empty

B9:	0 A/D FIFO half full;	1 A/D FIFO not half full
B10:	0 A/D FIFO full;	1 A/D FIFO not full
B11:	Reserved	
B12:	0 HSDI FIFO empty;	1 HSDI FIFO not empty
B13:	0 HSDI FIFO half full;	1 HSDI FIFO not half full
B14:	0 HSDI FIFO full;	1 HSDI FIFO not full
B[31-15]:	Reserved	

Write operation

Any value written to this register generates a Software A/D start command.

6.4.5 DAC1_UPDATE (WRITE)

Write operation

Any value written to this register updates the DAC 1 if the updates source is software updates.

6.4.6 DAC2_UPDATE (WRITE)

Write operation

Any value written to this register updates the DAC 2 if the updates source is software updates.

6.4.7 DAC_UPDATE (WRITE)

Write operation

Any value written to this register simultaneously updates the DAC 1 and DAC 2 if the updates source is software updates.

6.4.8 PACER (READ/WRITE)

Read Operation

Reading this register issues a software start trigger of the Pacer Clock if the start trigger source of the pacer clock is set to a software trigger. The return value has no meaning.

Write Operation

Writing to this register issues a software stop trigger of the Pacer Clock if the stop trigger source pacer clock is set to a software trigger. The value doesn't matter

6.4.9 TIMER (READ/WRITE)

Read Operation

A read provides the status of the gate of the Timer Counter circuits.

31 – 6						5	4	3	2	1	0
--------	--	--	--	--	--	---	---	---	---	---	---

B0:	0 Pacer clock gated;	1 Pacer clock enabled
B1:	0 Burst clock disabled;	1 Burst clock enabled
B2:	0 Pacer clock delayed start trigger over; 1 Pacer clock delayed start trigger in progress	
B3:	0 Pacer clock About trigger completed; 1 Pacer clock About trigger in progress	
B4:	0 Pacer clock can only be start triggered by software pacer start command; 1 Pacer clock can be start triggered	
B5:	0 Analog sampling is not halted 1 Analog sampling is halted by A/D FIFO full. It can be cleared by A/D FIFO clear command	
B[31-6]	Reserved	

Write Operation

Writing to this register issues a software Sample command for 8-bit High-Speed Digital Input lines if the sampling source is the software command. This command means a High-Speed Digital Input FIFO write procedure. The value doesn't matter

6.4.10 INT (READ/WRITE)

The SDM35541 board includes a built-in Priority Interrupt Controller that manages multiple interrupt sources, prioritizing them in order, with 0 being the highest priority. Using the Priority Interrupt Controller is straightforward:

1. **Configure the Interrupt Mask Register:** During software initialization, write to the Interrupt Mask Register (INT) to enable desired interrupt sources by setting their corresponding bits to 1.
2. **Handle Interrupts:** The Priority Interrupt Controller prioritizes and forwards interrupt requests to the PC. When an interrupt occurs, read the Interrupt Status Register (INT) in the Interrupt Service Routine to identify the active source, indicated by a single high bit. Service the interrupt request based on the identified source.
3. **Clear the Interrupt:** Clear the serviced interrupt by writing the appropriate bit pattern to the Interrupt Clear Register (CLEAR_INT), followed by a dummy read from the same address (CLEAR_INT) to execute the clear.

To ensure no new interrupts occurred during servicing, read the Interrupt Overrun Register after clearing the interrupt. A value of zero indicates all interrupts were serviced correctly. A value of one indicates a new interrupt occurred before the previous one was fully serviced. After reading, clear the Interrupt Overrun Register.

Read Operation

A read provides the status flag of the interrupt.

Lowest Priority																Highest Priority															
31 – 16																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B0:	0 Board FIFO Write inactive;															1 Board FIFO Write active															
B1:	0 Reset CGT inactive;															1 Reset CGT active															
B2:	Reserved																														
B3:	0 Pause CGT inactive;															1 Pause CGT active															
B4:	0 About Counter Out inactive;															1 About Counter Out active															
B5:	0 Delay Counter Out inactive;															1 Delay Counter Out active															
B6:	0 A/D Sample Counter inactive;															1 A/D Sample Counter active															
B7:	0 D/A1 Update Counter inactive;															1 D/A1 Update Counter active															
B8:	0 D/A2 Update Counter inactive;															1 D/A2 Update Counter active															
B9:	0 User TC1 Out inactive;															1 User TC1 Out active															
B10:	0 User TC1 Inverted Out inactive;															1 User TC1 Inverted Out active															
B11:	0 User TC2 Out inactive;															1 User TC2 Out active															
B12:	0 Digital Interrupt inactive;															1 Digital Interrupt active															
B13:	0 External Interrupt inactive;															1 External Interrupt active															
B14:	0 External Trigger rising edge inactive;															1 External Trigger rising edge active															
B15:	0 External Trigger falling edge inactive;															1 External Trigger falling edge active															
B[31-16]:	Reserved																														

Write Operation

The interrupt mask register:

B0:	0 Board FIFO Write disabled;	1 Board FIFO Write enabled
B1:	0 Reset CGT disabled;	1 Reset CGT enabled
B2:	Reserved	
B3:	0 Pause CGT disabled;	1 Pause CGT enabled
B4:	0 About Counter Out disabled;	1 About Counter Out enabled
B5:	0 Delay Counter Out disabled;	1 Delay Counter Out enabled
B6:	0 A/D Sample Counter disabled;	1 A/D Sample Counter enabled
B7:	0 D/A1 Update Counter disabled;	1 D/A1 Update Counter enabled
B8:	0 D/A2 Update Counter disabled;	1 D/A2 Update Counter enabled
B9:	0 User TC1 Out disabled;	1 User TC1 Out enabled
B10:	0 User TC1 Inverted Out disabled;	1 User TC1 Inverted Out enabled
B11:	0 User TC2 Out disabled;	1 User TC2 Out enabled
B12:	0 Digital Interrupt disabled;	1 Digital Interrupt enabled
B13:	0 Digital Interrupt disabled;	1 Digital Interrupt enabled
B14:	0 External Trigger rising edge disabled;	1 External Trigger rising edge enabled
B15:	0 External Trigger falling edge disabled;	1 External Trigger falling edge enabled

B[31-16] Reserved

6.4.11 CLEAR_INT (READ/WRITE)

Read Operation

A read clears the interrupt status flags of the selected source set by the clear mask.

Lowest Priority																Highest Priority																
31 – 16																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
B0:	0 Board FIFO Write clear no overrun;																1 Board FIFO Write clear overrun															
B1:	0 Reset CGT clear no overrun;																1 Reset CGT clear overrun															
B2:	Reserved																															
B3:	0 Pause CGT clear no overrun;																1 Pause CGT clear overrun															
B4:	0 About Counter Out no overrun;																1 About Counter Out overrun															
B5:	0 Delay Counter Out no overrun;																1 Delay Counter Out overrun															
B6:	0 A/D Sample Counter clear no overrun;																1 A/D Sample Counter clear overrun															
B7:	0 D/A1 Update Counter clear no overrun;																1 D/A1 Update Counter clear overrun															
B8:	0 D/A2 Update Counter clear no overrun;																1 D/A2 Update Counter clear overrun															
B9:	0 User TC1 Out clear no overrun;																1 User TC1 Out clear overrun															
B10:	0 User TC1 Inverted Out clear no overrun;																1 User TC1 Inverted Out clear overrun															
B11:	0 User TC2 Out clear no overrun;																1 User TC2 Out clear overrun															
B12:	0 Digital Interrupt clear no overrun;																1 Digital Interrupt clear overrun															
B13:	0 External Interrupt clear no overrun;																1 External Interrupt clear overrun															
B14:	0 External Trigger rising edge clear no overrun;																1 External Trigger rising edge clear overrun															
B15:	0 External Trigger falling edge clear no overrun;																1 External Trigger falling edge clear overrun															
BI31-16]	Reserved																															

Write Operation

B0:	0 Board FIFO Write clear disabled;																1 Board FIFO Write clear enabled															
B1:	0 Reset CGT clear disabled;																1 Reset CGT clear enabled															
B2:	Reserved																															
B3:	0 Pause CGT clear disabled;																1 Pause CGT clear enabled															
B4:	0 About Counter Out clear disabled;																1 About Counter Out clear enabled															
B5:	0 Delay Counter Out clear disabled;																1 Delay Counter Out clear enabled															
B6:	0 A/D Sample Counter clear disabled;																1 A/D Sample Counter clear enabled															
B7:	0 D/A1 Update Counter clear disabled;																1 D/A1 Update Counter clear enabled															
B8:	0 D/A2 Update Counter clear disabled;																1 D/A2 Update Counter clear enabled															
B9:	0 User TC1 Out clear disabled;																1 User TC1 Out clear enabled															
B10:	0 User TC1 Inverted Out clear disabled;																1 User TC1 Inverted Out clear enabled															
B11:	0 User TC2 Out clear disabled;																1 User TC2 Out clear enabled															
B12:	0 Digital Interrupt clear disabled;																1 Digital Interrupt clear enabled															
B13:	0 External Interrupt clear disabled;																1 External Interrupt clear enabled															
B14:	0 External Trigger rising edge clear disabled;																1 External Trigger rising edge clear enabled															
B15:	0 External Trigger falling edge clear disabled;																1 External Trigger falling edge clear enabled															
B[31-16]	Reserved																															

6.4.12 IT_OVERRUN (READ/WRITE)

Read Operation

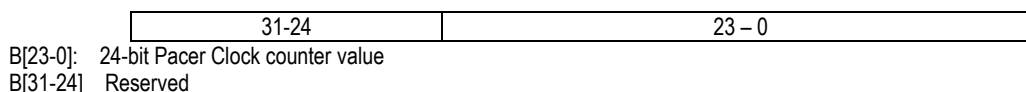
A read provides the Interrupt Overrun Register. If the interrupts serviced in time all bits are zeros. If a new interrupt request comes before the previous has been serviced and the request is cleared, the appropriate overrun bit goes into high.

Write Operation

A write clears all bits of the Interrupt Overrun Register.

6.4.13 PCLK_CNT (READ/WRITE)

The Pacer Clock Counter is a 24-bit down counter implemented in the FPGA. It operates using a 20 MHz or 8 MHz clock signal, with 8 MHz as the default frequency. This frequency can be adjusted by writing to PCLK_FREQ Register. The output, known as the Pacer Clock signal, remains high during counting and switches to low when the counter reaches zero.



Read Operation

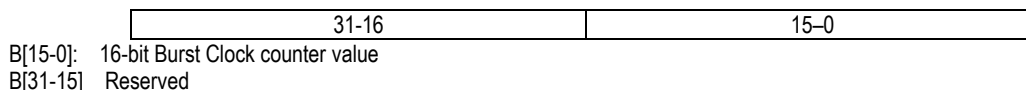
Current value of the counter.

Write Operation

Loads new counter value. The counter will start as soon as the write is completed.

6.4.14 BURST_CNT (READ/WRITE)

The Burst Clock Counter is a 16-bit down counter implemented in the FPGA. It operates using a 20 MHz or 8 MHz clock signal, with 8 MHz as the default frequency. This frequency can be adjusted by writing to BCLK_FREQ Register. The output, known as the Pacer Clock signal, remains high during counting and switches to low when the counter reaches zero. The 16-bit counter ensures a minimum Burst Clock frequency of 122 Hz.



Read Operation

Current value of the counter.

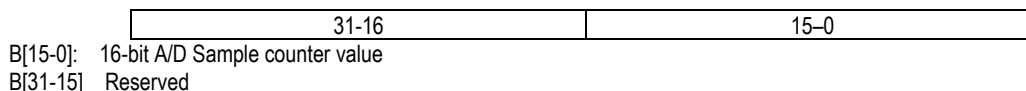
Write Operation

Loads new counter value. The counter will start as soon as the write is completed.

6.4.15 ADC_SAMPLE (READ/WRITE)

The A/D Sample Counter is a 16-bit down counter implemented in the FPGA. Its clock signal can be configured by writing to the ADC_SCNT_SRC address. The output, known as the A/D Sample Counter signal, remains high during counting, except when the counter reaches zero, at which point it switches to low. This high-to-low transition can trigger an interrupt, as the signal can serve as an interrupt source. When the counter is loaded, an immediate interrupt is generated, which can be prevented by disabling interrupts during the loading process. If a value n is written to the Sample Counter, the counter will reach zero and generate an interrupt after n+1 events.

The 16-bit wide A/D Sample Counter assures a maximum A/D sample count of 65536.



Read Operation

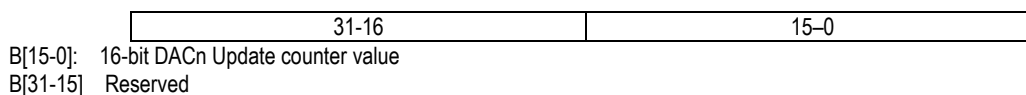
Current value of the counter.

Write Operation

Loads new counter value. The counter will start as soon as the write is completed.

6.4.16 DACn_SAMPLE (READ/WRITE)

The DACn Update Counter is a 16-bit down counter implemented in the FPGA. It is driven by the DACn update signal as its clock input. While counting, the counter's output signal remains high and transitions to low when the count reaches zero. With its 16-bit width, the counter can track up to 65,536 DACn update events.



Read Operation

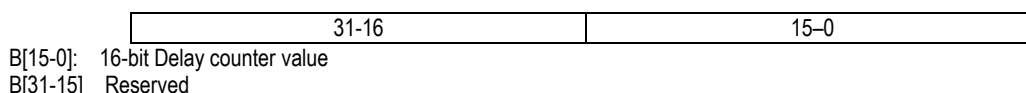
Current value of the counter.

Write Operation

Loads new counter value. The counter will start as soon as the write is completed

6.4.17 DELAY_CNT (READ/WRITE)

The Delay Counter Counter is a 16-bit down counter implemented in the FPGA. It uses a clock signal with the same frequency as the Pacer Clock. While the counter is active, the Pacer Clock is disabled. With its 16-bit width, the Delay Counter can delay the Start Pacer Clock signal by up to 65,535 Pacer Clock periods.



Read Operation

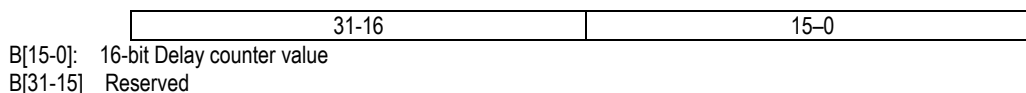
Current value of the counter.

Write Operation

Loads new counter value. The counter will start as soon as the write is completed.

6.4.18 ABOUT_CNT (READ/WRITE)

The About Counter enables delayed stopping of the Pacer Clock. When the sampling clock is set to the Pacer Clock, the About Counter specifies how many additional samples to collect after the stop trigger. It is a 16-bit down counter implemented in the FPGA, allowing up to 65,535 delayed samples. Loading the counter triggers an interrupt, which should be ignored in software as it is expected behavior.



Read Operation

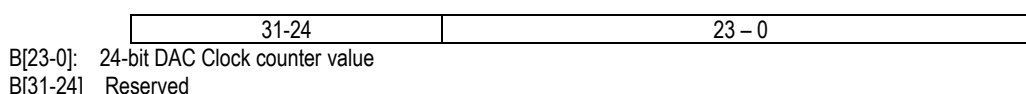
Current value of the counter.

Write Operation

Loads new counter value. The counter will start as soon as the write is completed.

6.4.19 DAC_CLK (READ/WRITE)

The DAC Clock Counter is a 24-bit down counter implemented in the FPGA. It operates using an 8 or 20 MHz clock signal. While counting, its output—the DAC Clock signal—remains high and transitions to low when the counter reaches zero. This DAC Clock signal can serve as the update trigger for the D/A converters.



Read Operation

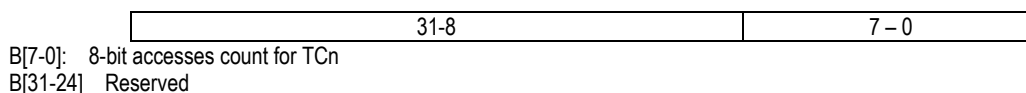
Current value of the counter.

Write Operation

Loads new counter value. The counter will start as soon as the write is completed.

6.4.20 UTCn (READ/WRITE)

The SDM35541 DAQ Board has an 8254 Timer Counter chip for the user. The clock sources and gates can be programmed.



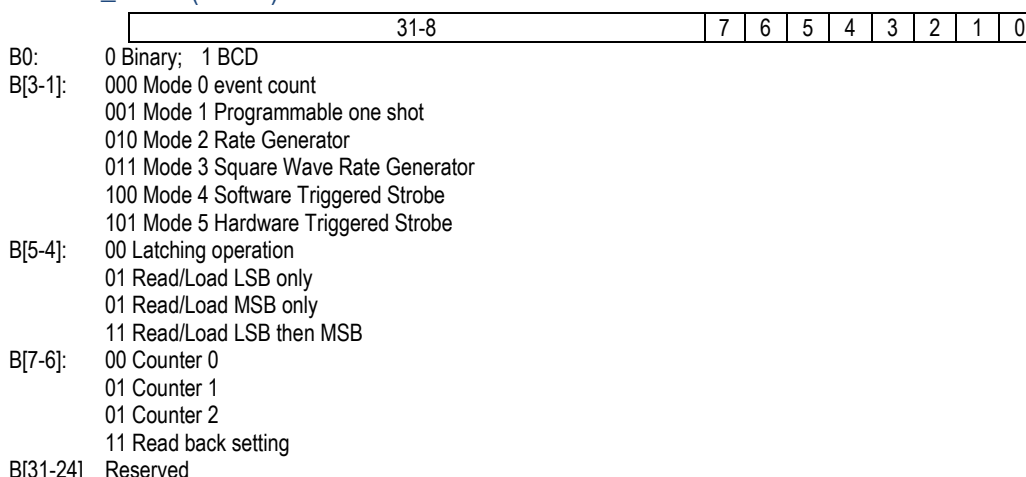
Read Operation

Two 8-bit accesses will read the count from TCn (LSB followed by MSB)

Write Operation

Two 8-bit accesses will write the count in TCn (LSB followed by MSB)

6.4.21 UTC_CTRL (WRITE)



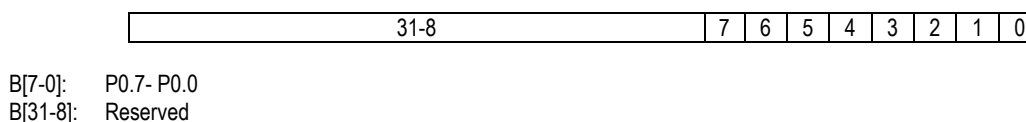
Write Operation

Accesses the timer/counter's control register to directly control the three 16-bit counters, 0, 1, and 2.

6.4.22 DIO0 (READ/WRITE)

Port 0 is an 8-bit programmable digital I/O interface that transfers data between the board and external devices. Each bit can be configured as either input or output by writing to the Direction Register (DIO_CTRL).

Note: Any reset of the digital circuitry—such as a chip clear or system reset—automatically sets all bits to input mode and clears the output registers.



Read Operation

Input mode: Reading retrieves the current input value
Output mode: Reading returns the last value sent

Write Operation

Input mode: Has no effect on DIO
Output mode: Updates the output line with the new value

6.4.23 DIO1 (READ/WRITE)

Port 1 is an 8-bit digital I/O interface used to exchange data between the board and external devices. Its operation depends on the configured direction:

Note: Any digital circuitry reset—such as a chip clear or system reset—automatically sets all Port 1 lines to input mode and clears their output registers.

31-8	7	6	5	4	3	2	1	0
------	---	---	---	---	---	---	---	---

B[7-0]: P1.7- P1.0

B[31-8]: Reserved

Read Operation

Input mode: Reading retrieves the current input value

Output mode: Reading returns the last value sent

Write Operation

Input mode: Has no effect on DIO

Output mode: Updates the output line with the new value

6.4.24 DIO_CTRL (READ/WRITE)

Read and write operations at DIO_STATUS behave differently depending on the values of bits 0 and 1:

- When bits 1:0 = 00:
 - Read: Clears the digital IRQ status flag.
 - Write: Clears the digital chip.
- When bits 1:0 ≠ 00:
 - The operation targets one of Port 0's three control registers.
 - **Read:** Returns the contents of the selected register.
 - **Write:** Programs the selected register.

Direction Register (DIO_STATUS, bits 1 and 0 = 01):

This register programs the direction, of each bit at Port 0, when DIO_STATUS = xxxxxx01 binary.

31-8	7	6	5	4	3	2	1	0
------	---	---	---	---	---	---	---	---

B[7-0]: 0 Input; 1 Output

B[31-8]: Reserved

Mask Register (DIO_STATUS, bits 1 and 0 = 10):

This register programs the mask, of each bit at Port 0, when DIO_STATUS = xxxxxx10 binary.

31-8	7	6	5	4	3	2	1	0
------	---	---	---	---	---	---	---	---

B[7-0]: 0 Bit enabled; 1 Masked

B[31-8]: Reserved

Compare Register (DIO_STATUS, bits 1 and 0 = 11):

This register is used in Advanced Digital Interrupt modes and operates differently depending on the selected mode:

Match Mode: An interrupt is triggered when the bit pattern at Port 0 matches a predefined value. This register holds the bit pattern to be matched. Individual bits can be masked to exclude them from the comparison. Important: Before writing to this register, ensure that bit 3 at address DIO_STATUS is set to 1 to enable match mode.

Event Mode: An interrupt occurs when any bit at Port 0 changes state. The bit pattern that caused the interrupt is latched into this register and can be read. Bits can be masked using the Mask Register to ignore changes on specific lines.

6.4.25 DIO_STATUS (READ/WRITE)

		31-8				7	6	5	4	3	2	1	0
B[1-0]:	Port 0 Control Register Select (see register DIO_CTRL)												
	00 Clear Digital IRQ Status Flag (read); Reset Digital I/O (write)												
	01 P0 direction register												
	10 P0 mask register												
	11 P0 compare register												
B2:	0 P1 direction Input;											1 Output	
B3:	0 Digital IRQ Event Mode;											1 Match Mode	
B4:	0 Digital IRQ Disabled;											1 Enabled	
B5:	Digital Sample Clock Select												
	0 8MHz clock												
	1 Programmable clock												
B6:	0 No digital interrupt											1 Digital interrupt (READ only)	
B7:	0 No strobe											1 Strobe (READ only)	
B[31-8]	Reserved												

Read Operation

- B5-0: Show the current states of these bits.
- Bit 6: Indicates whether a digital interrupt has occurred. A high value means an interrupt has taken place.
- Bit 7: Indicates whether a strobe event has occurred. A high value means a strobe has been issued.

Write Operation

- B5-0: Change states of these bits.
- B7-6: Do not modify the value if these bits

6.4.26 DSP (READ/WRITE)

Read Operation

This register is written to by the DSP after it has run the command routine. The host is responsible for reading this register to see what error code if any has been generated.

31-16	15	14	13	14	13	12	11	10	9	8	7-0
B[31-16]:	Reserved										
B[15-8]:	0 Command successful;		1 No Auto-Calibration in Flash								
B[7-0]:	Reserved										

Write Operation

This register is written to by the host to perform a variety of board functions. A non-maskable interrupt is generated to the DSP which in turn reads the register and performs the function.

31-16	7-0
B[31-8]:	Reserved
B[7-0]:	0x01 Auto-Calibrate
	0x02 Internal Flash Download
	0x03 Reserved
	0x04 Reserved
	0x05 Erase Internal Flash
	0x06 Reserved
	0x07 Attention DSP (is DSP alive)
	0x08 Load Factory Default calibration values
	0x09 Reserved
	0x0A Get Auto-Calibration Code Version
	0x0B Get Boot loader Code Version

6.4.27 TEMPERATURE SENSOR REGISTERS

The SDM35541HR has an board temperature sensor. The following registers are used to read the temperature (I2C_ADDR, I2C_PTR, I2C_DATA, I2C_GO, I2C_READ). The OS's API call will set the registers with values needed to the return the boards temperature.

6.4.28 ALGDIO_MASK (READ/WRITE)

31-6	5	4	3	2	1	0
------	---	---	---	---	---	---

B0:	0 Data output pin 1 of CN9 is masked;	1 Data output pin 1 of CN9is unmasked
B1:	0 Data output pin 2 of CN9 is masked;	1 Data output pin 2 of CN9is unmasked
B2:	0 Positive edge interrupt pin 1 of CN9 is masked;	1 Positive edge interrupt pin 1 of CN9 is unmasked
B3:	0 Positive edge interrupt pin 2 of CN9 is masked;	1 Positive edge interrupt pin 2 of CN9 is unmasked
B4:	0 Negative edge interrupt pin 1 of CN9 is masked;	1 Negative edge interrupt pin 1 of CN9 is unmasked
B5:	0 Negative edge interrupt pin 2 of CN9 is masked;	1 Negative edge interrupt pin 2 of CN9 is unmasked
B[31-6]	Reserved	

Read Operation

Returns the state of the mask for interrupts on the input and digital outputs.

Write Operation

Sets the mask for interrupts on the input and digital outputs. Default is all bits masked.

6.4.29 ALGDIO_DATA (READ/WRITE)

31-2	1	0
------	---	---

B0:	Data value pin 1 CN9
B1:	Data value pin 2 CN9
B[31-2]	Reserved

Read Operation

Return the actual signal present at the CN9 pin.

Write Operation

The written data reflects the value sent to the output buffer, not necessarily what appears at the physical pin. If the direction bit is not set to output, reading back the value may not match what was written.

6.4.30 ALGDIO_DIR (READ/WRITE)

Read /Write operation (32bit, 2 bits are used)

31-2	1	0
------	---	---

B0:	0 Input pin 1 CN9	1 Output pin 1 CN9
B1:	0 Input pin 2 CN9	1 Output pin 2 CN9
B[31-2]	Reserved	

Read Operation

Returns the state of the direction of the DIO on CN9.

Write Operation

Set the direction of the DIO pins on CN9. Default is input.

6.4.31 ALGDIO_INT (READ ONLY)

31-6	5	4	3	2	1-0
------	---	---	---	---	-----

B1-B0:	Reserved	
B2:	0 Positive edge interrupt pin 1 of CN9 is inactive;	1 Positive edge interrupt pin 1 of CN9 is active
B3:	0 Positive edge interrupt pin 2 of CN9 is inactive;	1 Positive edge interrupt pin 2 of CN9 is active
B4:	0 Negative edge interrupt pin 1 of CN9 is inactive;	1 Negative edge interrupt pin 1 of CN9 is active
B5:	0 Negative edge interrupt pin 2 of CN9 is inactive;	1 Negative edge interrupt pin 2 of CN9 is active
B31-B6	Reserved	

Read Operation

Reading this register provides the interrupt status for pins configured as inputs. It returns either masked or unmasked status, depending on configuration, and automatically clears the register after the read.

6.4.32 BOARD_RESET (WRITE ONLY)

Writing a dummy value to this address means a Software Reset. A software reset restores all internal logic variables of the board to their default power-up states.

6.4.33 DMA0_SRC (WRITE ONLY)

Writing to DMA0_SRC will select the Source Signal for DMA0 according to the list below.

0x00 = Request disable
 0x01 = A/D Sample Counter *
 0x02 = D/A1 Sample Counter *
 0x03 = D/A2 Sample Counter *
 0x04 = User TC 1 *
 0x08 = A/D FIFO half full
 0x09 = D/A1 FIFO half Empty
 0x0A = D/A2 FIFO half Empty

The selected source controls the DMA request signal. The signals, marked by the symbol *, set a request flip-flop only. These setups need the Reset DMA0 Request Machine command.

The FIFO flags control the DMA request signal directly, so they do not need the Reset command.

6.4.34 DMA1_SRC (WRITE ONLY)

Writing to DMA1_SRC will select the Source Signal for DMA1 according to the list below.

0x00 = Request disable
 0x01 = A/D Sample Counter *
 0x02 = D/A1 Sample Counter *
 0x03 = D/A2 Sample Counter *
 0x04 = User TC 1 *
 0x08 = A/D FIFO half full
 0x09 = D/A1 FIFO half Empty
 0x0A = D/A2 FIFO half Empty

The selected source controls the DMA request signal. The signals marked by the symbol * set a request flip-flop only. These setups need the Reset DMA1 Request Machine command.

The FIFO flags control the DMA request signal directly, so they do not need the Reset command.

6.4.35 ADC_CONV (WRITE ONLY)

The A/D conversion Signal can be selected by writing these values to ADC_CONV:

0x0 = Software A/D Start (value unimportant)
 0x1 = Pacer Clock (Ext. or Int.)
 0x2 = Burst Clock
 0x3 = Digital Interrupt
 0x4 = D/A1 Data Marker 1
 0x5 = D/A2 Data Marker 1
 0x6 = SyncBus0
 0x7 = SyncBus1
 0x8 = SyncBus2

The A/D Conversion signal select Function is used to choose the A/D Sampling signal. The Data Markers (0x4 and 0x5) are updated simultaneously with the appropriate DAC output. The conversion is started at the rising edge of the data marker. The data marker must be held in low state until you want to start a conversion. After the conversion the appropriate data marker must be cleared.

6.4.36 BURST_START (WRITE ONLY)

If you want to use the burst clock as conversion signal source the start trigger must be set by writing this address. The stop trigger of the burst clock is generated automatically, because the stop signal basically is the CGT reset signal that occurs at the end of the whole CGT cycle.

0x0 = Software A/D Start (value unimportant)
 0x1 = Pacer Clock
 0x2 = External Trigger
 0x3 = Digital Interrupt
 0x4 = SyncBus 0
 0x5 = SyncBus 1
 0x6 = SyncBus 2

6.4.37 PACER_START (WRITE ONLY)

If you want to use the Pacer Clock you must specify the start and stop conditions. The Pacer Clock Start Trigger Function selects the start signal of the Pacer Clock:

0x0 = Software Pacer Start (a dummy read from PACER)
 0x1 = External trigger
 0x2 = Digital interrupt
 0x3 = User TC 2 out
 0x4 = SyncBus 0
 0x5 = SyncBus 1
 0x6 = SyncBus 2
 0x7 = Reserved
 0x8 = Delayed Software Pacer Start (a dummy read from PACER)
 0x9 = Delayed external trigger
 0xA = Delayed digital interrupt
 0xB = Delayed User TC 2 out
 0xC = Delayed SyncBus 0
 0xD = Delayed SyncBus 1
 0xE = Delayed SyncBus 2
 0xF = External Trigger Gated controlled

The following start trigger sources, 0x8 through 0xF provide delayed triggering. When the start trigger is issued, the A/D Delay Counter, counts down and conversions are started when the A/D Delay Counter reaches 0. The A/D Delay Counter counts at the pacer clock rate. When using the External Trigger Gated control (0xF) the pacer clock runs as long as the External Trigger Input line is held high or low, depending on the trigger polarity. This mode does not use a stop trigger.

6.4.38 PACER_STOP (WRITE ONLY)

The Pacer Clock Stop Trigger Function selects the stop signal of the Pacer Clock:

0x0 = Software Pacer Stop (a dummy write to PACER)
 0x1 = External Trigger
 0x2 = Digital Interrupt
 0x3 = About Counter
 0x4 = User TC2 out
 0x5 = SyncBus 0
 0x6 = SyncBus 1
 0x7 = SyncBus 2
 0x8 = About Software Pacer Stop (a dummy write to PACER)
 0x9 = About External Trigger
 0xA = About Digital Interrupt
 0xB = Reserved
 0xC = About User TC2 out
 0xD = About SyncBus 0
 0xE = About SyncBus 1
 0xF = About SyncBus 2

Stop trigger sources 0x8 through 0xF provide about triggering, where data is acquired from the time the start trigger is received, and continues for a specified number of samples *after* the stop trigger is received. The number of samples taken after the stop trigger is received is set by the About Counter which counts down and is clocked by the writing signal of A/D FIFO. This method assures the desired number of samples will be sampled (count reaches 0).

6.4.39 ACNT_ENABLE (WRITE ONLY)

If the Pacer Clock is the source of A/D Conversion signal and the Pacer Clock Stop comes from the About Counter counting the samples in the A/D FIFO you can extend the counting capability highest number than 10 bit defined 1024 samples. Writing this address you can enable or disable the stop function:

0 = Stop enabled
 1 = Stop disabled

6.4.40 PACER_MODE (WRITE ONLY)

When set to single cycle, a trigger will initiate one conversion cycle and then stop, regardless of whether the trigger line is pulsed more than once; when set to repeat, a new cycle will start each time a trigger is received, and the current cycle has been completed. Triggers received while a cycle is in progress will be ignored. Writing this address you can select single cycle or repeat mode:

0x0 = Single Cycle Mode - new cycle can be possible after a Software Pacer Start command
 0x1 = Trigger Repeat Mode - Pacer can be started by the selected Pacer Start Trigger

6.4.41 HSDIN_START (WRITE ONLY)

The sampling signal of High-Speed Digital Input can be selected by writing this address. If you select the A/D conversion signal, the 8-bit digital input lines are simultaneously sampled with the analog signals.

0x0 = Software (dummy write TIMER)
 0x1 = A/D Conversion Signal
 0x2 = User TC out
 0x3 = User TC out 1
 0x4 = User TC out 2
 0x5 = External Pacer Clock
 0x6 = External Trigger

6.4.42 HSDIN_FIFO_CLR (WRITE ONLY)

Writing a dummy data to this address clears the High-Speed Digital Input FIFO.

6.4.43 ADC_FIFO_CLR (WRITE ONLY)

Writing a dummy data to this address clears the A/D FIFO.

6.4.44 CGT_WRITE (WRITE ONLY)

In the case of multi-channel operation the Channel Gain Table must be used. Before writing the channel gain table entries write a dummy data to 14Ch to clear the table. The structure of the entries in the table can be seen below:

31 – 16			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B3-0:	0x0	1 st analog input channel																
	0x1	2 nd analog input channel																
																		
	0xF	16 th analog input channel																
B6-4:	0x0	Gain 1																
	0x1	Gain 2																
	0x2	Gain 4																
	0x3	Gain 8																
	0x4	Gain 16																
	0x5	Gain 32																
	0x6	Gain 64																
B7:	0	AGND referenced SE input; 1 AINSENSE referenced SE input																
B9-8:	0x0	+/-5V input range																
	0x1	+/-10V input range																
	0x2	0 – 10V input range																
	0x3	reserved																
B10:	0	Single Ended; 1 Differential																
B11:	0	Pause bit disabled; 1 Enabled																
B12:	0	DAC1 update disabled; 1 Enabled																
B13:	0	DAC2 update disabled; 1 Enabled																
B14:		Skip bit disabled; 1 Enabled																
B15:		Reserved																
B31-B16		Reserved																

Using the pause bit: The pause bit of the channel-gain word is set to 1 if you want to stop at an entry in the table and wait for the next trigger to resume conversions. In burst mode, the pause bit is ignored.

Using the skip bit: The skip bit of the channel-gain word is set to 1 if you want to skip an entry in the table. This feature allows you to sample multiple channels at different rates on each channel. For example, if you want to sample channel 1 once each second and channel 4 once every 3 seconds, you can set the skip bit on channel 4. With the skip bit set on the four table entries; these entries will be ignored, and no A/D conversion will be performed. This saves memory and eliminates the need to throw away unwanted data.

6.4.45 CGT_LATCH (WRITE ONLY)

In the case of single-channel operation the Channel Gain Latch must be used. The data structure is the same as in the Channel Gain Table but there are no meaning of the skip bit, DACx Update and the Pause Bit. These bits must be zero.

6.4.46 DT_WRITE (WRITE ONLY)

The Digital Table is part of the Channel Gain Table, and can be used to control external devices. Using this function you can fill the 8-bit wide Digital Table. Reading of the Digital Table is simultaneous with reading the Channel Gain Table.

31–16																15–0															
-------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

B7-B0: P1.7-P1.0 Eight bit Digital output table bits
B31-B8 Reserved

The Digital Output Table bits use the same lines as the Digital I/O Chip Port 1 I/O port. In the case of usage the P1 digital I/O lines the Digital Table bits cannot be used. The Enable Digital Table Function (DT_ENABLE) can be used to select between the Digital I/O P1 port and the Digital Output Table bits. The digital portion of the channel-gain table provides 8-bits to control devices such as external expansion boards. For example, if you have connected one of your input channels on the SDM35541 to RTD's TMX32 input expansion board, you can use the bottom 5 bits in this byte to control the TMX32 board channel selection. To load digital information into this portion of the channel, use this function. This information will be output on the Port 1 lines when you run through the table. The format shown above is for controlling the TMX32's channel selection (32 single-ended or 16 differential). The first load operation will be in the first entry slot of the table (lining up with the first entry in the A/D table), and each load thereafter fills the

next position in the channel-gain table. Note that when you are using the digital table, all 8-bits are used and controlled by the table, regardless of the number of bits you may actually need for your digital control application.

6.4.47 CGT_ENABLE (WRITE ONLY)

Writing to this address you can select the Channel Gain Latch or the Channel Gain Table controlled operation.

0x0 = CGT disabled; Channel Gain Latch enabled
0x1 = CGT enabled Channel Gain Latch disabled

6.4.48 DT_ENABLE (WRITE ONLY)

Writing to this address you can select the P1 port of Digital I/O chip or the output of the Digital Table on the pin 32, 46 of External I/O connector.

0x0 = Digital Table disabled; Digital I/O P1 port enabled
0x1 = Digital Table enabled; Digital I/O P1 port disabled

6.4.49 PAUSE_TABLE (WRITE ONLY)

The pause bit of the Channel Gain Table is set to 1 if you want to stop at an entry in the table and wait for the next trigger to resume conversions. In burst mode, the pause bit is ignored. Writing this address this mode can be enabled:

0x0 = Table Pause enabled
0x1 = Table Pause disabled

6.4.50 CGT_RESET (WRITE ONLY)

Writing a dummy data to this address sets the read pointer of the Channel Gain Table to the beginning of the Table. The write pointer of the Table does not change.

6.4.51 CGT_CLEAR (WRITE ONLY)

Writing a dummy data to this address sets the read **and** the write pointer of the Channel Gain Table to the beginning of the Table.

6.4.52 DAC1_RANGE (WRITE ONLY)

Writing this address sets the voltage output range and polarity for DAC1:

0x0 = 0 – 5V range
0x1 = 0 – 10V range
0x2 = ± 5 V range
0x3 = ± 10 V range

6.4.53 DAC1_SRC (WRITE ONLY)

Writing this address selects the update source for DAC1.

0x0 = Software D/A1 Update (a dummy write to DAC1_UPDATE)
0x1 = CGT controlled D/A1 Update
0x2 = D/A Clock (source is output of D/A clock counter)
0x3 = External pacer clock
0x4 = SyncBus 0
0x5 = SyncBus 1
0x6 = SyncBus 2

The CGT Controlled Update assures simultaneous D/A update with the A/D conversion

6.4.54 DAC1_CYCLE (WRITE ONLY)

This bit enables the cycle mode for D/A1 converter. By writing 0x01, DAC1 will continuously repeat the data that is stored in the DAC1 FIFO. This is useful for waveform generation.

0x0 = not cycle
0x1 = cycle

6.4.55 DAC1_RESET (WRITE ONLY)

Writing a dummy data to this address sets the read pointer of the DAC1 FIFO to the beginning of the FIFO. The write pointer of the FIFO does not change.

6.4.56 DAC1_CLEAR (WRITE ONLY)

Writing a dummy data to this address sets the read **and** the write pointer of the DAC1 FIFO to the beginning of the FIFO.

6.4.57 DAC2_RANGE (WRITE ONLY)

Writing this address sets the voltage output range and polarity for DAC2:

0x0 = 0 – 5V range
0x1 = 0 – 10V range
0x2 = $\pm 5V$ range
0x3 = $\pm 10V$ range

6.4.58 DAC2_SRC (WRITE ONLY)

Writing this address selects the update source for DAC2.

0x0 = Software D/A2 Update (a dummy write to DAC2_UPDATE)
0x1 = CGT controlled D/A2 Update
0x2 = D/A Clock (source is output of D/A clock counter)
0x3 = External pacer clock
0x4 = SyncBus 0
0x5 = SyncBus 1
0x6 = SyncBus 2

The CGT Controlled Update assures simultaneous D/A update with the A/D conversion.

6.4.59 DAC2_CYCLE (WRITE ONLY)

This bits enables the cycle mode for DAC2 converter. By writing 0x01, DAC2 will continuously repeat the data that is stored in the DAC2 FIFO. This is useful for waveform generation.

0x0 = not cycle
0x1 = cycle

6.4.60 DAC2_RESET (WRITE ONLY)

Writing a dummy data to this address sets the read pointer of the DAC2 FIFO to the beginning of the FIFO. The write pointer of the FIFO does not change.

6.4.61 DAC1_CLEAR (WRITE ONLY)

Writing a dummy data to this address sets the read **and** the write pointer of the DAC2 FIFO to the beginning of the FIFO.

6.4.62 ADC_SCNT_SRC (WRITE ONLY)

Writing this address the A/D Sample Counter Clock can be selected:

0x0 = Reset Channel Gain Table
0x1 = A/D FIFO write

If you want to count all of the sampled analog data select the A/D FIFO write argument. If you want to count the CGT periods select the Reset Channel Gain Table argument.

6.4.63 PACER_SELECT (WRITE ONLY)

Selects the internal Pacer Clock, which is the output of internal Pacer Clock generator or an external Pacer Clock routed onto the board through External I/O connector:

0x0 = External Pacer Clock
0x1 = Internal Pacer Clock

The maximum Pacer Clock rate supported by the board is 1.25 MHz.

6.4.64 SBUS0_SOURCE (WRITE ONLY)

This function selects the source of the SyncBus 0 signal:

0x0 = Software A/D Start
0x1 = Pacer Clock
0x2 = Burst Clock
0x3 = Digital Interrupt
0x4 = External Trigger
0x5 = Software Simultaneous D/A1 and D/A2 Update
0x6 = D/A Clock
0x7 = User TC2 out

The SyncBus is a 3-line synchronization purpose bus to synchronize the operation of multiple SDM35541 DAQ Boards. The source of signals can be the same and can be on the other SDM35541 boards.

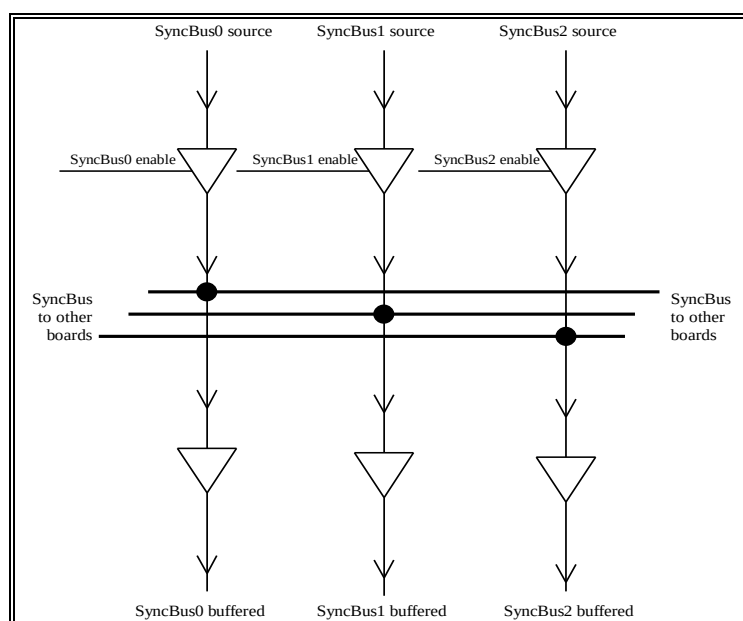
6.4.65 SBUS0_ENABLE (WRITE ONLY)

This function enables the SyncBus 0 buffer

0x0 = disable
0x1 = enable

NOTE: When connecting SyncBus signals together ensure that each signal has only one driver.

Figure 13: SyncBus Structure



6.4.66 SBUS1_SOURCE (WRITE ONLY)

This function selects the source of the SyncBus 1 signal:

0x0 = Software A/D Start
0x1 = Pacer Clock
0x2 = Burst Clock
0x3 = Digital Interrupt
0x4 = External Trigger
0x5 = Software Simultaneous D/A1 and D/A2 Update
0x6 = D/A Clock
0x7 = User TC2 out

The SyncBus is a 3-line synchronization purpose bus to synchronize the operation of multiple SDM35541 DAQ Boards. The source of signals can be the same and can be on the other SDM35541 boards.

6.4.67 SBUS1_ENABLE (WRITE ONLY)

This function enables the SyncBus 1 buffer

0x0 = disable
0x1 = enable

NOTE: When connecting SyncBus signals together ensure that each signal has only one driver.

6.4.68 SBUS2_SOURCE (WRITE ONLY)

This function selects the source of the SyncBus 2 signal:

0x0 = Software A/D Start
0x1 = Pacer Clock
0x2 = Burst Clock
0x3 = Digital Interrupt
0x4 = External Trigger
0x5 = Software Simultaneous D/A1 and D/A2 Update
0x6 = D/A Clock
0x7 = User TC2 out

The SyncBus is a 3-line synchronization purpose bus to synchronize the operation of multiply SDM35541 DAQ Boards. The source of signals can be the same and can be on the other SDM35541 boards.

6.4.69 SBUS2_ENABLE (WRITE ONLY)

This function enables the SyncBus 1 buffer

0x0 = disable
0x1 = enable

NOTE: When connecting SyncBus signals together ensure that each signal has only one driver.

6.4.70 ETRG_POLARITY (WRITE ONLY)

This function selects the active polarity of External Trigger signal from the I/O connector

0x0 = positive edge
0x1 = negative edge

6.4.71 EINT_POLARITY (WRITE ONLY)

This function selects the active polarity of the External Interrupt signal:

0x0 = positive edge
0x1 = negative edge

The External Interrupt signal comes from the External I/O connector. The External Interrupt may be a source of built-in priority Interrupt Controller.

6.4.72 UTC0_CLOCK (WRITE ONLY)

This function selects the source of the User TC 0 clock signal:

0x0 = 8MHz
 0x1 = Ext. TC Clock 1
 0x2 = Ext. TC Clock 2
 0x3 = Ext. Pacer Clock

6.4.73 UTC0_GATE (WRITE ONLY)

This function selects the source of the User TC 0 gate signal:

0x0 = Not gated (free running)
 0x1 = Gated (logic high or low)
 0x2 = Ext. TC Gate 1
 0x3 = Ext. TC Gate 2

6.4.74 UTC1_CLOCK (WRITE ONLY)

This function selects the source of the User TC 1 clock signal:

0x0 = 8MHz
 0x1 = Ext. TC Clock 1
 0x2 = Ext. TC Clock 2
 0x3 = Ext. Pacer Clock
 0x4 = User Timer/Counter 0 out
 0x5 = High-Speed Digital Input Sampling signal

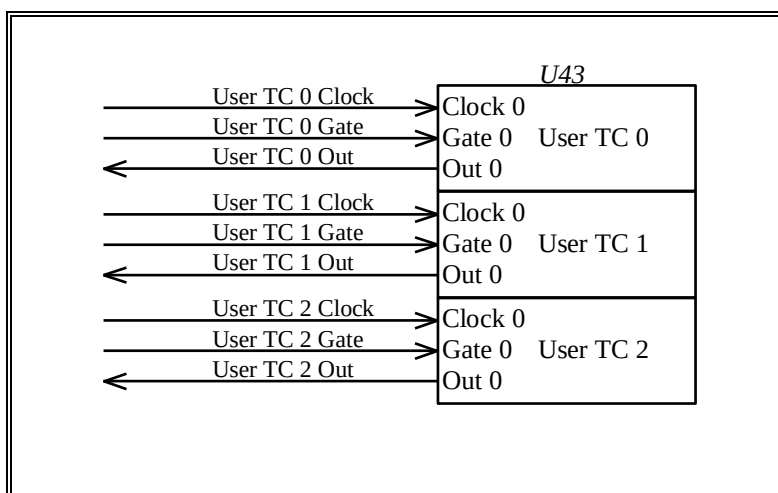
You can cascade timer TC 1 using 0x4. You can use User TC 1 as a Sample Counter for the High Speed Digital Input FIFO.

6.4.75 UTC1_GATE (WRITE ONLY)

This function selects the source of the User TC 1 gate signal:

0x0 = Not gated (free running)
 0x1 = Gated (logic high or low)
 0x2 = Ext. TC Gate 1
 0x3 = Ext. TC Gate 2
 0x4 = User Timer/Counter 0 out

Figure 14: User TC Section



6.4.76 UTC2_CLOCK (WRITE ONLY)

This function selects the source of the User TC 2 clock signal:

0x0 = 8MHz
 0x1 = Ext. TC Clock 1
 0x2 = Ext. TC Clock 2
 0x3 = Ext. Pacer Clock
 0x4 = User Timer/Counter 1 out

You can cascade timer TC 1 using 0x4.

6.4.77 UTC2_GATE (WRITE ONLY)

This function selects the source of the User TC 2 gate signal:

0x0 = Not gated (free running)
 0x1 = Gated (logic high or low)
 0x2 = Ext. TC Gate 1
 0x3 = Ext. TC Gate 2
 0x4 = User Timer/Counter 1 out

6.4.78 UIO0_SELECT (WRITE ONLY)

The selected sources are buffered and connected to the External I/O connector. The source of the *User Out 0* can be programmed using this function:

0x0 = A/D Conversion Signal
 0x1 = D/A1 Update
 0x2 = D/A2 Update
 0x3 = Software Programmable (see register USER_IO)

6.4.79 UIO1_SELECT (Write Only)

The selected sources are buffered and connected to the External I/O connector. The source of the *User Out 1* can be programmed using this function:

0x0 = A/D Conversion Signal
 0x1 = D/A1 Update
 0x2 = D/A2 Update
 0x3 = Software Programmable (see register USER_IO)

6.4.80 DMA_RSTQST0 (WRITE ONLY)

The *Reset DMA0 Request machine* command resets the DMA0 Request. This command can be activated by writing a dummy value. This command has effect only in Demand Mode. It is needed only with DMA request sources signed by * in DMA0_SRC.

6.4.81 DMA_RSTQST1 (WRITE ONLY)

The *Reset DMA1 Request machine* command resets the DMA1 Request. This command can be activated by writing a dummy value. This command has effect only in Demand Mode. It is needed only with DMA request sources signed by * in DMA1_SRC.

6.4.82 DAC_CLK_START (WRITE ONLY)

0x0 = Software Pacer Start (Read PACER)
 0x1 = External trigger
 0x2 = Digital interrupt
 0x3 = User TC 2 out
 0x4 = SyncBus 0
 0x5 = SyncBus 1
 0x6 = SyncBus 2
 0x7 = Software D/A clock start (Read DAC_CLK_ST)

6.4.83 DAC_CLK_STOP (WRITE ONLY)

0x0 = Software Pacer Stop (Write PACER)
 0x1 = External Trigger

0x2 = Digital Interrupt
0x3 = User TC2 out
0x4 = SyncBus 0
0x5 = SyncBus 1
0x6 = SyncBus 2
0x7 = Software D/A clock stop (Write DAC_CLK_ST)
0x8 = D/A1 update counter
0x9 = D/A2 update counter

6.4.84 PCLK_FREQ (WRITE ONLY)

0x0 = 8MHz
0x1 = 20MHz

6.4.85 BCLK_FREQ (WRITE ONLY)

0x0 = 8MHz
0x1 = 20MHz

6.4.86 DAC_CLK_SOURCE (WRITE ONLY)

0x0 = 8MHz
0x1 = 20MHz

6.4.87 DAC_CLK_MODE (WRITE ONLY)

0x0 = D/A clock is running free
0x1 = D/A clock started or stopped by functions DAC_CLK_START or DAC_CLK_STOP

6.4.88 MCBSP_AD_CTRL (WRITE ONLY)

This Function enables the automatic sending of the A/D FIFO data to the connected DSP via the McBSP serial connection:

0x0 = A/D FIFO data to DSP is disabled
0x1 = A/D FIFO data to DSP is enabled

This means that this data cannot be read by the host via the PCI bus.

6.4.89 MCBSP_DA_CTRL (WRITE ONLY)

This Function enables a connected DSP to write to the D/A1 and D/A2 FIFO via the McBSP serial connection:

0x0 = D/A1 and D/A2 FIFO data from DSP is disabled
0x1 = D/A1 and D/A2 FIFO data from DSP is enabled

6.4.90 ADC_FIFO (READ ONLY)

A read provides the 12-bit A/D converted data as shown below. Bit 15 is the sign bit extension. This sign bit extension gives the opportunity to read the converted data as two's complement number in either unipolar or bipolar mode. The bottom three bits are the samples of the buffered version of the External I/O connector Port 0 Digital I/O port P0-5, P0-6, P0-7 lines which can be used as independent Data Markers. The sampling is simultaneous with this read instruction.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

B0: P0-5
 B1: P0-6
 B2: P0-7
 B3: A/D bit 1
 B4: A/D bit 2
 B5: A/D bit 3
 B6: A/D bit 4
 B7: A/D bit 5
 B8: A/D bit 6
 B9: A/D bit 7
 B10: A/D bit 8
 B11: A/D bit 9
 B12: A/D bit 10
 B13: A/D bit 11
 B14: A/D bit 12
 B15: Sign bit

6.4.91 HSDIN_FIFO (READ ONLY)

A read provides the 8-bit High Speed Digital Input Data bits which are programmable source sampled. The High Speed Digital Input lines are commonly used with the Digital I/O, bit programmable P0 port and can be used as independent Data Markers. The upper byte is undefined.

15-8	7	6	5	4	3	2	1	0
------	---	---	---	---	---	---	---	---

B0: P0-0
 B1: P0-1
 B2: P0-2
 B3: P0-3
 B4: P0-4
 B5: P0-5
 B6: P0-6
 B7: P0-7
 B15-B8: Undefined

6.4.92 DAC1_FIFO (WRITE ONLY)

A write programs the D/A1 FIFO in the format shown below, as two's complement data. A write also sets the D/A1 data markers. The buffered version of D/A1 data marker 0 is connected to the I/O connector. It can be used as a source for the A/D Sample signal. This register can also be written to via the McBSP connection. In this mode the D/A selection bit (bit 2) controls the data direction to the D/A1 or D/A2 FIFO.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

- B0: D/A1 digital Output Data Marker 0
- B1: D/A1 digital Output Data Marker 1
- B2: 0 D/A1 receives McBSP Data; 1 D/A2 receives McBSP Data (McBSP mode only)
- B3: D/A1 bit 1 (unless using McBSP, then see bit 2)
- B4: D/A1 bit 2 (unless using McBSP, then see bit 2)
- B5: D/A1 bit 3 (unless using McBSP, then see bit 2)
- B6: D/A1 bit 4 (unless using McBSP, then see bit 2)
- B7: D/A1 bit 5 (unless using McBSP, then see bit 2)
- B8: D/A1 bit 6 (unless using McBSP, then see bit 2)
- B9: D/A1 bit 7 (unless using McBSP, then see bit 2)
- B10: D/A1 bit 8 (unless using McBSP, then see bit 2)
- B11: D/A1 bit 9 (unless using McBSP, then see bit 2)
- B12: D/A1 bit 10 (unless using McBSP, then see bit 2)
- B13: D/A1 bit 11 (unless using McBSP, then see bit 2)
- B14: D/A1 bit 12 (unless using McBSP, then see bit 2)
- B15: D/A1 sign bit

6.4.93 DAC2_FIFO (WRITE ONLY)

A write programs the D/A2 FIFO in the format shown below, as two's complement data. A write also sets the D/A2 data markers. The buffered version of D/A2 data marker 0 is connected to the I/O connector. It can be used as a source for the A/D Sample signal. This register can also be written to via the McBSP connection. In this mode the D/A selection bit (bit 2) controls the data direction to the D/A1 or D/A2 FIFO.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

- B0: D/A2 digital Output Data Marker 0
- B1: D/A2 digital Output Data Marker 1
- B2: 0 D/A1 receives McBSP Data; 1 D/A2 receives McBSP Data (McBSP mode only)
- B3: D/A2 bit 1 (unless using McBSP, then see bit 2)
- B4: D/A2 bit 2 (unless using McBSP, then see bit 2)
- B5: D/A2 bit 3 (unless using McBSP, then see bit 2)
- B6: D/A2 bit 4 (unless using McBSP, then see bit 2)
- B7: D/A2 bit 5 (unless using McBSP, then see bit 2)
- B8: D/A2 bit 6 (unless using McBSP, then see bit 2)
- B9: D/A2 bit 7 (unless using McBSP, then see bit 2)
- B10: D/A2 bit 8 (unless using McBSP, then see bit 2)
- B11: D/A2 bit 9 (unless using McBSP, then see bit 2)
- B12: D/A2 bit 10 (unless using McBSP, then see bit 2)
- B13: D/A2 bit 11 (unless using McBSP, then see bit 2)
- B14: D/A2 bit 12 (unless using McBSP, then see bit 2)
- B15: D/A2 sign bit

6.4.94 DMA_CH0_FIFO (READ/WRITE)

Selects the FIFO source for DMA Channel 0. This register is used in combination with DMA0_SRC, to setup the DMA channel.

- 0x0 = Disabled
- 0x1 = ADC_FIFO
- 0x2 = DAC1_FIFO
- 0x3 = DAC2_FIFO
- 0x3 = HSDIN_FIFO

6.4.95 DMA_CH1_FIFO (READ/WRITE)

Selects the FIFO source for DMA Channel 1. This register is used in combination with DMA1_SRC, to setup the DMA channel.

0x0 = Disabled
0x1 = ADC_FIFO
0x2 = DAC1_FIFO
0x3 = DAC2_FIFO
0x3 = HSDIN_FIFO

7 Programming Guide

7.1 A/D Conversion

This chapter shows you how to program your SDM35541 to perform A/D conversions and read the results. Included in this discussion are instructions on setting up the Channel Gain Table (CGT), the on-board clocks and sample counter, and various conversions and triggering modes. The following paragraphs walk you through the programming steps for performing A/D conversions. Detailed information about the conversion modes and triggering is presented in this section. You can follow these steps in the example programs included with the board.

7.1.1 BEFORE STARTING CONVERSIONS: INITIALIZING THE BOARD

Regardless of the conversion mode you wish to set up, you should always start your program with a board initialization sequence. This sequence should include:

- Clear Board Command
- Clear IRQ command
- Clear Channel Gain Table command
- Clear A/D FIFO command
- Clear D/A FIFOs commands
- Clear Digital I/O chip

This initialization procedure clears all board registers, empties the Channel Gain Table, resets the digital I/O chip and empties the A/D and D/A FIFOs.

Before Starting Conversions (single-channel mode): Programming Channel Gain Latch (CGL)

Setting up these things can be done using the Channel Gain Latch (single-channel mode) or using the Channel Gain Table (multi-channel mode) The CGL can be filled up by CGT_LATCH. The Channel Gain Latch has very similar structure to the Channel Gain Table, so all operations are explained in the next sections of CGT.

Before Starting Conversions (multi-channel mode): Programming the Channel-Gain Table (CGT)

The Channel Gain Table can be programmed with 8K 24-bit entries in tabular format. Sixteen bits contain the A/D channel-gain data (A/D Table), and 8-bits contain digital control data (Digital Table) to support complex channel-gain sequences. To load a new Channel Gain Table, first: clear the Channel Gain Table by CGT_CLEAR. To add entries to an existing table, simply write to the A/D Table (and Digital Table if used) as described in the following paragraphs. Note that writing beyond the end of the table is ignored.

16-Bit A/D Table

The A/D portion of the Channel Gain Table with the channel, gain, input range, input type, pause and skip bit information is programmed into the channel-gain scan memory using the CGT_WRITE. If you have cleared the existing table, the first word written will be placed in the first entry of the table, the second word will be placed in the second entry, and so on. If you are adding to an existing table, the new data written will be added at the end.

Channel Select, Gain Select, Input Range and Input Type

The channel number, gain value, input range and input type are entered in the table using bits 0 through 10. Each of these parameters can be set independently for every entry in the table. This allows you to set up a complex array of sampling sequences mixing channels, gains, input ranges and input types. Care must be taken in selecting the proper input type. The board is capable of 16 single-ended inputs or 8 differential inputs. You can select combinations of single-ended and differential, but each differential channel actually uses 2 single-ended channels. If you select channel 1 to be a differential channel, you must connect your signal to AIN1+ and AIN1-. Channel 8 now is not available as a single-ended channel. In the case of single ended mode, you can choose the Ground Referenced Single Ended (GRSE mode) or the Non Referenced Single Ended Mode (NRSE).

Pause bit

Bit 11 is used as a pause bit. If this bit is set to a "1" and the Pause function is enabled by PAUSE_TABLE, the A/D conversions will stop at this entry in the table and resume on the next Start Trigger. This is useful if you have 2 different sequences loaded in the table. You can enable and disable this bit's function by PAUSE_TABLE. In the case of single channel mode, when the CGL is used this function is meaningless. NOTE: This bit is ignored in the Burst sampling modes.

D/Ax update bits

Bit 12, 13 is used for simultaneous update of the D/Ax converter with the sampling of the appropriate analog input channel. When these bits are in high state a D/A update signal is generated at the sampling time of the analog input.

Skip bit

If bit 14 of the data loaded is set to 1, then the skip bit is enabled and this entry in the channel-gain table will be skipped, meaning an A/D conversion will be performed but the data is not written into the A/D FIFO. This feature provides an easy way to sample multiple channels at different rates without saving unwanted data. A simple example illustrates this bit's function.

In this example, we want to sample channel 1 once each second and channel 4 once every three seconds. First, we must program 6 entries into the channel-gain table. The channel 4 entries with the skip bit set will be skipped when A/D conversions are performed. The table will continue to cycle until a stop trigger is received.

Next, we will set the pacer clock to run at 2 Hz (0.5 seconds). This allows us to sample each channel once per second, the maximum sampling rate required by one of the channels (pacer clock rate = number of different channels sampled x fastest sample rate). The first clock pulse starts an A/D conversion according to the parameters set in the first entry of the channel-gain table, and each successive clock pulse incrementally steps through the table entries. The first clock pulse starts a sample on channel 1. The next pulse looks at the second entry in the channel-gain table and sees that the skip bit is set to 1. No A/D data is stored. The third pulse starts a sample on channel 1 again, the fourth pulse skips the next entry, and the fifth pulse takes our third reading on channel 1. On the sixth pulse, the skip bit is disabled and channel 4 is sampled. Then the sequence starts over again. Samples are not stored when they are not wanted; saving memory and eliminating the need to throw away unwanted data.

8-Bit Digital Table

The digital portion of the channel-gain table can be programmed with digital control information using the Write Digital Table DT_WRITE. If you have cleared the existing table by the CGT clear CGT_CLEAR, the first byte written will be placed in the first entry of the table, the second byte will be placed in the second entry, and so on. If you are adding to an existing table, the new data written will be added at the end. The first entry made into the Digital Table lines up with the first entry made into the A/D Table, the second entry made into the Digital Table lines up with the second entry made into the A/D Table, and so on. Make sure that, if you add to an existing table and did not program the Digital Table portion when you made your A/D Table entries previously, you fill those entries with digital data first before entering the desired added data. Since the first digital entry you make always lines up with the first A/D entry made, failure to do this will cause the A/D and digital control data to be misaligned in the table. You cannot turn the digital control lines off for part of a conversion sequence and then turn them on for the remainder of the sequence. Note that the digital data programmed here is sent out on the Port 1 digital I/O lines whenever this portion of the table is enabled by DT_ENABLE.

These lines can be used to control input expansion boards such as the TMX32 analog input expansion board at the same speed as the A/D conversions are performed with no software overhead.

NOTE: If you only need to use the A/D part of the table, you do not have to program the Digital Table. However if you only want to use the Digital part of the table you must program the A/D part of the table.

Setting Up A/D part and Digital part of Channel Gain Table

Let's look at how the Channel Gain Table is set up for a simple example using both the A/D and Digital Tables. In this example, we have a TMX32 expansion board connected to channel 1 on the SDM35541. Load the channel-gain sequence into the A/D Table (CGT_WRITE):

```
Entry 1      0000 0000 0000 0000 gain = 1, channel number = 1
Entry 2      0000 0000 0010 0000 gain = 4, channel number = 1
Entry 3      0100 0000 0000 0000 skip sample
Entry 4      0000 0000 0010 0000 gain = 4, channel number = 1
Entry 5      0000 0000 0000 0000 gain = 1, channel number = 1
Entry 6      0000 0000 0010 0000 gain = 4, channel number = 1
```

Load the digital data into the Digital Table by DT_WRITE. The first digital word loaded lines up with the first A/D Table entry, and so on:

```
Entry 1 0000 0000 0000 0000 gain=1 SDM35541 channel=1 0000 0000 TMX32 channel=1
Entry 2 0000 0000 0010 0000 gain=4 SDM35541 channel=1 0000 0011 TMX32 channel=4
Entry 3 0000 1000 0000 0000 skip sample                      0000 0000 TMX32 channel=1 (skip)
Entry 4 0000 0000 0010 0000 gain=4 SDM35541 channel=1 0000 0011 TMX32 channel=4
```

```
Entry 5 0000 0000 0000 0000 gain=1 SDM35541 channel=1 0000 0000 TMX32 channel=1
Entry 6 0000 0000 0010 0000 gain=4 SDM35541 channel=1 0000 0011 TMX32 channel=4
```

Using the Channel Gain Table for A/D Conversions

After the Channel Gain Table is programmed, it must be enabled in order to be used for A/D conversions by CGT_ENABLE. The Digital Table can be enabled by DT_ENABLE when the digital control data is stored. You cannot use the Digital Table without enabling the Channel Gain Table. When the Digital Table is enabled, the 8-bit data is sent out on the Port 1 digital I/O lines.

When you are using the channel-gain table to take samples, it is strongly recommended that you do not enable, disable, and then re-enable the table while performing a sequence of conversions. This causes skipping of an entry in the table. In this case you should issue a reset table command by CGT_RESET.

Channel-gain Table and Throughput Rates

When using the Channel Gain Table, you should group your entries to maximize the throughput of your module. Low-level input signals and varying gains are likely to drop the throughput rate because low level inputs must drive out high level input residual signals. To maximize throughput:

- Keep channels configured for a certain range grouped together, even if they are out of sequence.
- Use external signal conditioning if you are performing high speed scanning of low level signals. This increases throughput and reduces noise.
- If you have room in the channel-gain table, you can make an entry twice to make sure that sufficient settling time has been allowed and an accurate reading has been taken. Set the skip bit for the first entry so that it is ignored.
- For best results, do not use the channel-gain table when measuring steady-state signals. Use the single convert mode to step through the channels.

7.1.2 A/D CONVERSION MODES

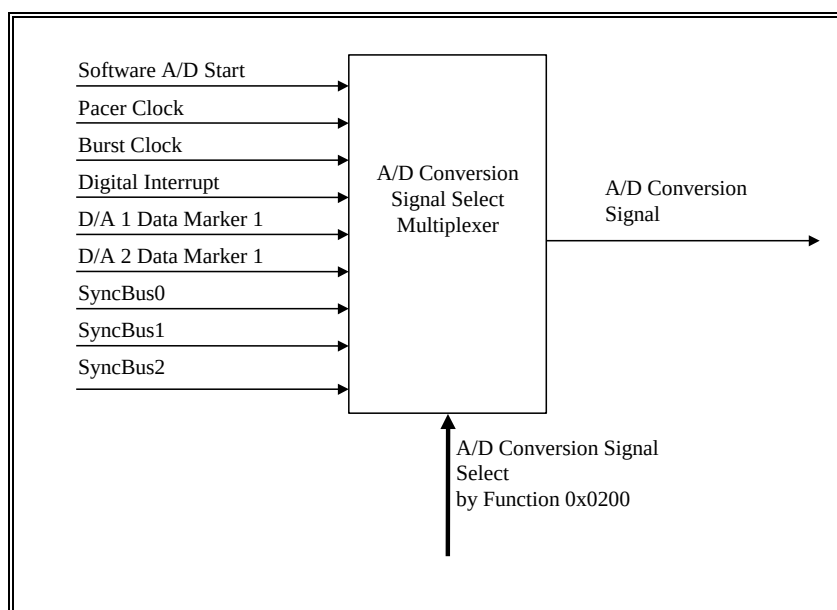
To support a wide range of sampling requirements, the SDM35541 provides several conversion modes with a selection of trigger sources to start and stop a sequence of conversions. Understanding how these modes and sources can be configured to work together is the key to understanding the A/D conversion capabilities of your module.

The following paragraphs describe the conversion and trigger modes.

Start A/D Conversion signal

Using the ADC_CONV one of nine modes can be selected as A/D conversion signal as can be seen on Figure 15 below.

Figure 15: A/D Conversion Signal



- *Software A/D Start* (by writing FIFO_STATUS to initiate a Start Convert)
- *Pacer Clock* (internal TC)
- *Burst Clock* (internal TC)
- *Digital Interrupt* generated by the Advanced Digital Interrupt circuit
- *D/A 1 Data Marker 1** for simultaneous A/D conversion with D/A update
- *D/A 2 Data Marker 1** for simultaneous A/D conversion with D/A update
- *SyncBus* signals (three lines)

Pacer Clock Start/Stop Trigger Select

The Pacer Clock start trigger can be set by the PACER_START. The Pacer Clock stop trigger can be set by the PACER_STOP. This function can be used to turn the pacer clock (internal or external) on and off. Through these different combinations of start and stop triggers, the SDM35541 supports pre-trigger, post-trigger, and about-trigger modes with various trigger sources.

The Pacer Clock start trigger sources are:

- *Software Pacer Start* When selected, a read at PACER will start the Pacer Clock.
- *External trigger* When selected, a positive- or negative-going edge (depending on the setting of the trigger polarity, ETRG_POLARITY) on the external TRIGGER INPUT line, will start the pacer clock. The pulse duration should be at least 100 nanoseconds.
- *Digital interrupt* When selected, a digital interrupt -generated by Advanced Digital I/O chip - will start the Pacer Clock.
- *User TC 2 out* When selected, a pulse on the User Timer/Counter 2 output line (Counter 2's count reaches 0) will start the pacer clock.
- *SyncBus 0..2* When selected a positive edge on the SyncBus 0 line will start the pacer clock.

The following start trigger sources provide delayed triggering. When the trigger is issued, the A/D delay counter, counts down and conversions are started when the A/D delay counter reaches 0. The A/D delay counter counts at the pacer clock rate.

- *Delayed Software Pacer Start*. When selected, a read at PACER will start the delay counter.
- *Delayed external trigger*. When selected, a positive- or negative-going edge (depending on the setting of the trigger polarity, bit 11 in the Control Register) on the external TRIGGER INPUT line, will start the delay counter. The pulse duration should be at least 100 nanoseconds.
- *Delayed digital interrupt*. When selected, a digital interrupt will start the delay counter.
- *Delayed User TC Counter 2 output*. When selected, a pulse on the Counter 2 output line (Counter 2's count reaches 0) will start the delay counter.
- *Delayed SyncBus 0..2* When selected, a pulse on the SyncBus0 will start the delay counter.
- *External Trigger Gated mode*. When selected, the pacer clock runs when the external TRIGGER INPUT line is held high. When this line goes low, conversions stop. This trigger mode does not use a stop trigger. If the trigger polarity bit is set for negative, the pacer clock runs when this line is low and stops when it is taken high.

The Pacer Clock stop trigger sources are:

- *Software Pacer Stop trigger*. When selected, a write at PACER will stop the Pacer Clock.
- *External trigger*. When selected, a positive- or negative-going edge (depending on the setting of the trigger polarity, setting up by EINT_POLARITY) on the external TRIGGER INPUT line, will stop the Pacer Clock. The pulse duration should be at least 100 nanoseconds.
- *Digital interrupt*. When selected, a digital interrupt will stop the pacer clock.
- *About Counter*. When selected, the Pacer Clock stops when the About Counter's count reaches 0. About Counter counts samples which are written into the A/D FIFO
- *User TC2 out* When selected, the Pacer Clock stops when the User TC 2 counter's count reaches 0.
- *SyncBus0.. 2 signals* when selected, the Pacer Clock stops when there is a rising edge on the SyncBus line.

The following stop trigger sources provide ABOUT triggering, where data is acquired from the time the start trigger is received, and continues for a specified number of samples after the stop trigger. The number of samples to acquire after the stop trigger is programmed in the About Counter. About Counter counts samples which are written into the A/D FIFO

- *About Software Pacer Stop trigger*. When selected, a Software Pacer Stop trigger starts the About counter, and sampling continues until the About Counter's count reaches 0.
- *About external trigger*. When selected, an external trigger starts the About counter, and sampling continues until the sample counter's count reaches 0.
- *About digital interrupt*. When selected, a digital interrupt starts the About Counter, and sampling continues until the About Counter's count reaches 0.

- *About User TC Counter 2 output.* When selected, a pulse on the User Timer Counter 2 output line (Counter 2's count reaches 0) starts the About Counter, and sampling continues until the About Counter's count reaches 0.
- *About SyncBus0.. 2.* When selected, a rising edge on SyncBus0..2 starts the About Counter, and sampling continues until the About Counter's count reaches 0.

Note that the external trigger (TRIGGER INPUT) can be set to occur on a positive-going edge or a negative-going edge, depending on the setting up the EINT_POLARITY.

Burst Clock Start Trigger to trigger burst sample

The following paragraph describes the operation when the A/D conversion start signal is selected as Burst Clock. Burst clock is an output of Timer Counter. The clock signal of the Burst Clock Timer Counter is 8MHz or 20MHz. The gate signal of this TC is used to start and stop the Burst Clock. The start signal can be select from the following list, and the stop is derived from the empty signal of Channel Gain Table. The Burst Clock operation belongs to the Multichannel - Channel Gain Table operation.

The start triggers can be set by BURST_START:

- *Software A/D start* (by writing FIFO_STATUS)
- *Pacer Clock* (internal or external)
- *external TRIGGER INPUT*
- *Digital Interrupt*
- *SyncBus0.. 2*

Single Cycle Mode, Trigger Repeat mode

Using the Pacer Start Mode select PACER_MODE the Single Cycle mode or the Trigger Repeat Mode can be selected. This function controls the conversion sequence when using a trigger to start the Pacer Clock. When the Function argument is low, the first pulse on the selected Pacer Clock Start Trigger source will start the pacer clock. After the stop trigger has ended the conversion cycle, the triggering circuit is disarmed and must be rearmed before another start trigger can be recognized. To rearm this trigger circuit, you must issue a Software A/D Start command (Write FIFO_STATUS).

When Function argument is high, the conversion sequence is repeated each time a selected Pacer Clock Start Trigger is received.

Pacer Clock Source

The Pacer Clock can be generated from an internal source or an external source using the PACER_SELECT.

Types of Conversions

Single Conversion

In this mode, a single specified channel is sampled whenever the Software A/D Start Command is occurred. The active channel is the one specified in the Channel Gain Latch. This is the easiest of all conversions. It can be used in a wide variety of applications, such as sample every time a key is pressed on the keyboard, sample with each iteration of a loop, or watch the system clock and sample every five seconds.

Multiple Conversions

In this mode, conversions are continuously performed at the Pacer Clock rate, or other selected A/D Conversion Signal rate. The pacer clock can be internal or external. The maximum rate supported by the board is 1.25MHz. The Pacer Clock can be turned on and off using any of the start and stop triggering modes using the PACER_START, and PACER_STOP. If you use the internal pacer clock, you must program it to run at the desired rate.

This mode is ideal for filling arrays, acquiring data for a specified period of time, and taking a specified number of samples.

Random Channel Scan

In this mode, the Channel Gain Table is incrementally scanned through, with each selected A/D Conversion Signal pulse starting a conversion at the channel and gain specified in the current table entry. Before starting a conversion sequence Channel Gain Table, you need to load the table with the desired data. Then make sure that the Channel Gain Table is enabled by the CGT_ENABLE. This enables the A/D portion of the Channel Gain Table. If you are using the Digital Table as well, you must also enable this using the DT_ENABLE. Each rising edge of selected A/D Conversion Signal starts a conversion using the current Channel Gain data and then increments to the next position in the table. When the last entry is reached, the next pulse starts the table over again.

Programmable Burst

In this mode, a single trigger initiates a scan of the entire Channel-gain table. Before starting a burst of the channel-gain table, you need to load the table with the desired data. Then make sure that the channel-gain table is enabled. If you are using the Digital Table as well, you must also enable it.

Burst is used when you want one sample from a specified number of channels for each trigger. The burst trigger, which is a trigger or pacer clock, triggers the burst and the burst clock initiates each conversion. At high speeds, the burst mode emulates simultaneous sampling of multiple input channels. For time critical simultaneous sampling applications, a simultaneous sample-and-hold board can be used (SS8 eight-channel boards are available from RTD).

Programmable Multiscan

This mode - when the A/D Conversion Start Signal is the Burst Clock - lets you scan the Channel Gain Table after a Burst Clock Start Signal. When the Channel Gain Table is empty the Burst Clock is stopped, and waiting for a new Start Signal.

As you can see, SDM35541 is designed to support a wide range of conversion requirements. You can set the clocks, triggers, and channel and gain to a number of configurations to perform simple or very complex acquisition schemes where multiple bursts are taken at timed intervals. Remember that the key to configuring the board for your application is to understand what signals can actually control conversions and what signals serve as triggers. The discussions presented in this section and the example programs on the disk should help you to understand how to configure the board.

Starting an A/D Conversion

Depending on your conversion and trigger settings, the Software A/D Start command (Write FIFO_STATUS) has different functions. In any mode that uses the Software A/D Start command, this command will do the appropriate action. In any mode that does not use the Software A/D Start command as the trigger; you will still need to do a write the FIFO_STATUS to arm (enable) the triggering circuitry. An example of this would be, if you set the Pacer Clock Start Trigger as external trigger, write the FIFO_STATUS is required to arm the external trigger circuitry. After you have set all the trigger and conversion registers to the proper values, the last command will need to be Software A/D Start. Any external triggers received before this command will be ignored. It is also a good practice to clear the A/D FIFO just prior to triggering the measurement or arming the trigger. Study the example programs to see this sequence.

Conversion Status Monitoring

The A/D conversion status can be monitored through the A/D FIFO empty flag in the FIFO status word read at FIFO_STATUS. Typically, you will want to monitor the Empty flag (active low) for a transition from low to high. This tells you that a conversion is complete and data has been placed in the sample buffer.

Halting Conversions

In single convert modes, a single conversion is performed and the module waits for another Software A/D Start command. In multi-convert modes, conversions are halted by one of two methods: when a stop trigger has been issued to stop the pacer clock, or when the FIFO is full. The Pacer Clock Shut Down Flag, bit 4 of the status word (PACER), is set when the sample buffer is full, disabling the A/D converter. Even if you've removed data from the sample buffer since the buffer filled up and the FIFO full flag is no longer set, the Pacer Clock Shut Down Flag will confirm that at some point in your conversion sequence, the sample buffer filled and conversions were halted. At this point a clear A/D FIFO command must be issued and a Software A/D Start convert (write at FIFO_STATUS) to rearm the trigger circuitry.

7.1.3 READING THE CONVERTED DATA

Each 12-bit conversion is stored in a 16-bit word in the sample buffer, in the A/D FIFO. The buffer can store 1024 samples. This section explains how to read the data stored in the sample buffer.

The sample buffer - A/D FIFO contains the converted data and 3-bit data marker (if used) in a 16-bit word.

The 12-bit A/D data + sign bit are left justified in a 16-bit word, with the least significant three bits reserved for the data marker. Because of this, the A/D data read must be scaled to obtain a valid A/D reading. The data marker portion should be masked out of the final A/D result. Shifting the word three bits to the right will eliminate the data marker from the data word. If you are using the data marker, then you should preserve these bits someplace in your program.

The output code format is always two's complement. This is true for both bipolar and unipolar signals since the sign bit is added above the 12-bit conversion data. For bipolar conversions, the sign bit will follow the MSB of the 12-bit data. If this bit is a "0", the reading is a positive value. If this bit is a "1", the reading is a negative value. When the

input is a unipolar range, the coding is the same except that the sign bit is always a "0" indicating a positive value. The data should always be read from the A/D FIFO as a signed integer.

Voltage values for each bit will vary depending on input range and gain. For example, if the input is set for ± 5 volts and the gain = 1, the formula for calculating voltage is as follows:

$$\begin{aligned}\text{Voltage} &= ((\text{input range} / \text{Gain}) / 4096) \times \text{Conversion Data} \\ \text{Voltage} &= ((10 / 1) / 4096) \times \text{Conversion Data} \\ \text{Voltage} &= 2.44 \text{ mV} \times \text{Conversion Data}\end{aligned}$$

Remember that when you change the gain, you are increasing the resolution of the bit value but you are decreasing the input range. In the above example if we change the gain to 4, each bit will now be equal to $610 \mu\text{V}$ but our input range is decreased from 10 volts to 2.5 volts. The formula would look like this:

$$\begin{aligned}\text{Voltage} &= ((\text{input range} / \text{Gain}) / 4096) \times \text{Conversion Data} \\ \text{Voltage} &= ((10 / 4) / 4096) \times \text{Conversion Data} \\ \text{Voltage} &= 610 \mu\text{V} \times \text{Conversion Data}\end{aligned}$$

If we now change the input range to ± 10 volts and the gain = 1, the formula would be:

$$\begin{aligned}\text{Voltage} &= ((\text{input range} / \text{Gain}) / 4096) \times \text{Conversion Data} \\ \text{Voltage} &= ((20 / 1) / 4096) \times \text{Conversion Data} \\ \text{Voltage} &= 4.88 \text{ mV} \times \text{Conversion Data}\end{aligned}$$

7.1.4 USING THE A/D DATA MARKERS

For certain applications where you may want to store digital information with the analog data at the same rate the analog data is being acquired, the bottom three bits of the converted data are available for this feature. For example, you may want to tag the acquired data with a marker so that you know when the data was sampled. Three lines are available at I/O connector to send the data marker settings to the sample buffer along with the 12-bit A/D converted data. These lines are P0.5, P0.6 and P0.7.

7.1.5 PROGRAMMING THE PACER CLOCK

The SDM35541 has a 24-bit count down on-board pacer clock with 8MHz/20MHz clock signal. When you want to use the pacer clock for continuous A/D conversions, you must select the Pacer Clock as A/D Conversion Signal and program the clock rate.

The pacer clock is accessed for programming at PCLK_CNT. To find the value you must load into the clock to produce the desired rate, you first have to calculate the value of Divider for the 24-bit clock. The formulas for making this calculation are as follows:

$$\begin{aligned}\text{Pacer Clock frequency} &= 8 (20) \text{ MHz} / (\text{Divider} + 1) \\ \text{Divider} &= (8 (20) \text{ MHz} / \text{Pacer Clock Frequency}) - 1\end{aligned}$$

The Pacer Clock frequency range is 1.14 MHz.. 0.47Hz defined by the 8MHz clock frequency, the 24bit wide counter and the 1.25MHz maximum Sampling frequency.

Table 18: Pacer Clock Frequency

Pacer Clock	Sampling Cycle Time	Pacer Clock	Sampling Cycle Time	Divider (decimal)
Primary Clock – 8 MHz See PCLK_FREQ		Primary Clock – 20 MHz See PCLK_FREQ		
1.14MHz	877ns	-	-	6
1MHz	1 μs	-	-	7
888.89kHz	1.125 μs	-	-	8
800kHz	1.25 μs	-	-	9
500kHz	2 μs	1.25MHz	800ns	15
470.588kHz	2.15 μs	1.176MHz	850ns	16
..	..	..	..	..
100kHz	10 μs	250kHz	4 μs	79
..	..	..	..	..
50kHz	20 μs	125kHz	8 μs	159
10kHz	100 μs	25kHz	40 μs	799

Pacer Clock	Sampling Cycle Time	Pacer Clock	Sampling Cycle Time	Divider (decimal)
Primary Clock – 8 MHz See PCLK_FREQ		Primary Clock – 20 MHz See PCLK_FREQ		
1kHz	1ms	2.5kHz	0.4ms	7999
100Hz	10ms	250Hz	4ms	79999
10Hz	100ms	25Hz	40ms	799999
1Hz	1s	2.5Hz	0.4s	7999999
477mHz	2.09s	1.19Hz	0.84s	16777216

Writing the Divider into the PCLK_CNT the Pacer Clock works immediately according to this value. Writing process clears the Counter - generates a Pacer Clock pulse, and loads the Divider value to the Counter.

7.1.6 PROGRAMMING THE BURST CLOCK

The SDM35541 has a 16-bit count down on-board Burst Clock timer with 8 / 20MHz clock signal. When you want to use the Burst Clock for performing A/D conversions in the burst mode, you must program the clock rate by writing the BURST_CNT. To find the Divider value you must load into the Burst Clock Counter to produce the desired rate, make the following calculation:

The Burst Clock Frequency Range is 1.14MHz.. 122Hz – 8MHz primary clock, and 1.25MHz.. 305Hz – 20MHz primary clock

$$\text{Burst Clock Frequency} = 8 / 20 \text{ MHz} / (\text{Divider} + 1)$$

$$\text{Divider} = (8 / 20 \text{ MHz} / \text{Pacer Clock Frequency}) - 1$$

Table 19: Burst Clock Frequency

Burst Clock	Sampling Cycle Time	Burst Clock	Sampling Cycle Time	Divider (decimal)
Primary Clock – 8 MHz See BCLK_FREQ		Primary Clock – 20 MHz See BCLK_FREQ		
1.14MHz	877ns	-	-	6
1MHz	1µs	-	-	7
888.89kHz	1.125µs	-	-	8
800kHz	1.25µs	-	-	9
500kHz	2µs	1.25MHz	800ns	15
100kHz	10µs	250kHz	4µs	79
50kHz	20µs	125kHz	8µs	159
10kHz	100µs	25kHz	40µs	799
122Hz	8.1ms	305.17Hz	3.2ms	65535

For example to set the burst clock frequency at 100 kHz, this equation becomes:

$$\text{Divider} = (8 \text{ MHz} / \text{Pacer Clock Frequency}) - 1 = 8\text{MHz}/100\text{kHz} - 1 = 79$$

After you determine the divider value that will result in the desired clock frequency, write it into the BURST_CNT. Writing the Divider into the BURST_CNT the Burst Clock works immediately according to this value. Writing process clears the Counter - generates a Burst Clock pulse, and loads the Divider value to the Counter.

7.1.7 PROGRAMMING THE ABOUT COUNTER

The About Counter lets you program the SDM35541 to take a certain number of samples and then halt conversions. (Select A/D Conversion Signal to Pacer Clock, selects the Pacer Clock Stop Trigger to About Counter) The number of samples minus one to be taken is loaded into the 16-bit About Counter at BURST_CNT.

Note that once the counter is properly loaded and starts, any subsequent countdowns of this count will be accurate.

After you determine the desired number of samples, load the number minus 1 to the About Counter register.

Note: Make sure all registers are set and board and FIFO's are cleared before enabling interrupts.

Using the About Counter to Create Large Data Arrays

The 16-bit About Counter allows you to take up to 65,535 samples before the count reaches 0 and sampling is halted. Suppose, however, you want to take 100,000 samples and stop. The SDM35541 provides a Function, About Counter

Stop Enable 0x0204 which allows you to use the About counter to take more than 65,535 samples in a conversion sequence.

The About Counter stop enable bit can be set to 1 to allow the sample counter to continuously cycle through the loaded count until the stop enable bit is set to 0, which then causes the sample counter to stop at the end of the current cycle. Let's look back at our example where we want to take 100,000 readings. First, we must divide 100,000 by a whole number that gives a result of less than 65,535. In our example, we can divide as follows:

$$\text{Sample Counter Count} = 100,000 / 2 = 50,000$$

To use the sample counter to take 100,000 samples, we will load a value of 50,000 into the counter and cycle the counter two times. After the value is loaded, make sure that the Stop Bit is set to 1 so that the sample counter will cycle. Then, set up the sample counter so that it generates an interrupt when the count reaches 0. Initialize the sample counter as described in the preceding section and start the conversion sequence. When the sample counter interrupt occurs telling you that the count has reached 0 and the cycle is starting again, set the Stop Bit to 0 to stop the sample counter after the second cycle is completed. The result: the sample counter runs through the count two times and 100,000 samples are taken.

7.2 D/A Conversion

Two independent 12-bit analog output channels are included on the SDM35541. The analog outputs are generated by two 12-bit D/A converters with independent software programmable output ranges. Each D/A channel have 8K Sample D/A FIFO. The analog output signals are accompanied by two digital data markers, DM0, and DM1. The DM0 bits are buffered and wired to the External I/O Connector. The DM1 bits may be the Start Conversion Signal of A/D converter. The digital data markers are updated simultaneously with the analog output signal.

D/A1 data is written to DAC1_FIFO and D/A2 data is written to DAC2_FIFO. The data are written into the D/A FIFOs, and the Update signals read the FIFOs, and update the D/A converters.

The DAC1_RANGE (DAC2_RANGE for D/A2) sets the voltage output range and polarity for D/A1. The output ranges are ± 5 , ± 10 , 0 to +5, or 0 to +10 volts.

The Function DAC1_SRC selects the update source for D/A1:

- *Software D/A1 Update.* Write a dummy data to DAC1_UPDATE. (DAC2_UPDATE for D/A2)
- *CGT Controlled D/A1 Update* If the D12 (D13 for D/A2) bit of CGT is 1 the D/A1 is updated simultaneously with the A/D sampled analog input
- *DAC Clock* The 24bit D/A clock inside the control logic.

The SDM35541 has a 16-bit count down on-board DAC Clock timer with 8 / 20MHz clock signal. When you want to use the DAC Clock for performing D/A conversions in the burst mode, you must program the clock rate by writing the DAC_CLK. To find the Divider value you must load into the DAC Clock Counter to produce the desired rate, make the following calculation:

The DAC Clock Frequency Range is 200 kHz... 0.47Hz – 8MHz primary clock, and 200 kHz... 1.19Hz – 20MHz primary clock.

$$\text{Burst Clock Frequency} = 8 / 20 \text{ MHz} / (\text{Divider} + 1)$$

$$\text{Divider} = (8 / 20 \text{ MHz} / \text{Pacer Clock Frequency}) - 1$$

Table 20: DAC Clock Frequency

DAC Clock	Sampling Cycle Time	DAC Clock	Sampling Cycle Time	Divider (decimal)
Primary Clock – 8 MHz DAC_CLK_SOURCE		Primary Clock – 20 MHz DAC_CLK_SOURCE		
200kHz	5 μ s	-	-	39
195.1kHz	5.125 μ s	-	-	40
80kHz	12.5 μ s	200kHz	5 μ s	99
79.208kHz	12.625 μ s	198.02kHz	5.05 μ s	100
10kHz	100 μ s	25kHz	40 μ s	79
8kHz	125 μ s	20kHz	50 μ s	999

DAC Clock	Sampling Cycle Time	DAC Clock	Sampling Cycle Time	Divider (decimal)
Primary Clock – 8 MHz DAC_CLK_SOURCE		Primary Clock – 20 MHz DAC_CLK_SOURCE		15916777215
0.47Hz	2.127s	1.19Hz	0.838s	

For example to set the DAC clock frequency at 100 kHz, this equation becomes:

$$\text{Divider} = (8 \text{ MHz} / \text{DAC Clock Frequency}) - 1 = 8\text{MHz}/100\text{kHz} - 1 = 79$$

After you determine the divider value that will result in the desired clock frequency, write it into DAC_CLK. Writing the Divider into DAC_CLK the DAC Clock works immediately according to this value. Writing process clears the Counter - generates a DAC Clock pulse, and loads the Divider value to the Counter. Note, that the DAC clock needs a start command.

- *External Pacer Clock* The rising edge of External Pacer Clock at the external I/O Connector updates the D/A1. The minimum pulse width is 100ns.
- *SyncBus0... 2* The rising edge of SyncBus signals updates the D/A1. The source of SyncBus signals may be on the same board or on another SDM35541 board.

The DAC1_CYCLE (DAC2_CYCLE for D/A2) selects the cycled or not cycled mode for D/A1. In the case of cycled mode, emptying the D/A1 FIFO the Update pointer of the FIFO is set to the beginning of the data array in the FIFO. This mode can be used for generating periodic signals without any processor intervention. This means that setting this bit to a 1; the D/A1 will continuously repeat the data that is stored in the D/A1 FIFO. This is useful for waveform generation. The not cycled mode is the normal operation mode.

The Function DAC1_RESET (DAC2_RESET for D/A2) resets the D/A1 FIFO. This Function sets the update pointer of the D/A1 FIFO to the beginning of the data array in the FIFO.

The Function DAC1_CLEAR (DAC2_CLEAR for D/A2) clears the D/A1 FIFO. This Function sets the update and write pointer of the D/A1 FIFO to the beginning of. This means that the FIFO is ready to fill with new data.

The following tables list the key digital codes and corresponding output voltages for the D/A converters.

Table 21: DAC Bipolar Binary Value vs. Output Voltage

Bipolar D/A Bit Weight	Ideal Output Voltage(mV)	
	-5 to +5 Volts	-10 to +10 Volts
2047	+4997.56	+9995.12
1024	+2500.00	+5000.00
512	+1250.00	+2500.00
256	+625.00	+1250.00
128	+312.50	+625.00
64	+156.25	+312.50
32	+78.13	+156.25
16	+39.06	+78.13
8	+19.53	+39.06
4	+9.77	+19.53
2	+4.88	+9.77
1	+2.44	+4.88
0	0	0.00
-1	-2.44	-4.88
-2	-4.88	-9.77
-4	-9.77	-19.53
-8	-19.53	-39.06
-16	-39.06	-78.13
-32	-78.13	-156.25
-64	-156.25	-312.50
-128	-312.50	-625.00
-256	-625.00	-1250.00
-512	-1250.00	-2500.00
-1024	-2500.00	-5000.00
-2048	-5000.00	-10000.00

Table 22: DAC Unipolar Binary Value vs. Output Voltage

Unipolar D/A Bit Weight	Ideal Output Voltage(mV)	
	-5 to +5 Volts	-10 to +10 Volts
4095	+4997.56	+9995.12
2047	+2500.00	+5000.00
1024	+1250.00	+2500.00
512	+625.00	+1250.00
256	+312.50	+625.00
128	+156.25	+312.50
64	+78.13	+156.25
32	+39.06	+78.13
16	+19.53	+39.06
8	+9.77	+19.53
4	+4.88	+9.77
2	+2.44	+4.88
1	+1.22	+2.44
0	0	0.00

7.2.1 8K SAMPLE BUFFER

Each D/A channel have a 8k sample buffers for storing data to be sent to the D/A converter (D/A FIFO). This means that you can fill the buffer with data and set up the D/A to output this data automatically. This is very useful for outputting high speed data or generating waveforms with precise timing requirements. By setting the cycled mode, you can fill the buffer with one cycle of a wave, start the D/A update clock and the buffer will continue to repeat until the clock is stopped. Combining this feature with the variety of update sources, you can build a flexible waveform generator.

If you are trying to generate a non-repetitive waveform, you can combine the sample buffer capability with the D/A Update Counter. To utilize this feature of the SDM35541 properly, you should load the buffer with data, program the D/A Update Counter for half the buffer size (512 samples), and use the Update Counter to generate an interrupt. When an interrupt is received, you should reload the buffer with 512 new samples. By continuing this cycle, you can generate a non-repetitive waveform at high speeds.

Status of the FIFO buffers can be monitored at FIFO_STATUS. Any samples that are written to the FIFO after it is full will be ignored. You can write up to 1024 samples to the buffer before it is full. Each update pulse (either software or from one of the clocks) will remove a sample from the buffer and send it out the D/A. Each update - read after the FIFO buffer is empty will be ignored, and the output of the D/A remains in the last updated state.

At power-up or reset, the D/A outputs are set to 0 volts. Before loading data into the sample buffer it is best to clear the buffer by DAC1_CLEAR or DAC2_CLEAR. When you issue the "Clear D/A FIFO" command, all data in the buffer is erased. If you issue the "Reset DAC FIFO" command, the data in the buffer is not erased, however the address pointer is set back to the beginning of the buffer: This is useful when you are generating waveforms and stop the updating in the middle of a cycle.

7.2.2 D/A CYCLED OR NOT CYCLED MODE

The cycle bit is used to make the buffer data repeat. Under normal operation, without the cycled mode set, data is written into the buffer and the update clock reads data out of the buffer. When the buffer is empty, the output of the D/A remains unchanged. If you set the cycle bit high, the data in the buffer will repeat. If you load a data set into the buffer, when the update clock reaches the end of the data it will automatically wrap around to the beginning and start over. This is useful for generating waveforms.

7.2.3 D/A UPDATE COUNTERS

The D/A1 and D/A2 16-bit wide Update Counters, are useful when using clocks to output data to the D/As. The counters can be accessed at DAC1_SAMPLE, and DAC2_SAMPLE addresses. These counters will count update pulses sent to the D/A's and can be polled to read the current count or can be used to generate interrupts when the count reaches 0. These counters can be loaded to any starting value and count down. When the count reaches 0 it will automatically be reloaded with the original starting value.

7.2.4 D/A DATA MARKERS

The D/A Data Markers are used to send out digital pulses synchronized to the D/A analog output. Since each D/A FIFO buffer is 16-bits wide and the D/A only uses 12 bits, there are bits left for Data Markers. Two of these bit

locations can be filled with data and this data is sent out on the appropriate pins synchronized to the D/A analog output. This is useful for sending out a trigger pulse each time a waveform crosses zero or to send out pulses to trigger A/D conversions at the proper time in the D/A waveform. Each D/A channel have 2 Data Marker bits. The DMO outputs can be accessed at the external I/O connector.

7.3 Advanced Digital Interrupts

The bit programmable digital I/O circuitry supports two Advanced Digital Interrupt modes, event mode or match mode. These modes are used to monitor digital input lines (P0..7) for state changes. The mode is selected at DIO_STATUS, B3 and enabled at DIO_STATUS, B4.

Event Mode

When enabled, this mode samples the Port 0 input lines at a specified clock rate (using the 8 MHz system clock or a programmable clock in User TC Counter 1), looking for a change in state in any one of the eight bits. When a change of state occurs, an interrupt is generated, and the input pattern is latched into the Compare Register. You can read the contents of this register at DIO_CTRL (DIO_STATUS bits 1 and 0 set to 11) to see which bit caused the interrupt to occur. Bits can be masked, and their state changes ignored by programming the Mask Register with the mask at DIO_CTRL (DIO_STATUS bits 1 and 0 set to 10).

Match Mode

When enabled, this mode samples the Port 0 input lines at a specified clock rate (using the 8 MHz system clock or a programmable clock in User TC Counter 1) and compares all input states to the value programmed in the Compare Register at DIO_CTRL (DIO_STATUS bits 1 and 0 set to 11). When the states of all of the lines match the value in the Compare Register, an interrupt is generated. Bits can be masked, and their states ignored by programming the Mask Register with the mask at DIO_CTRL (DIO_STATUS bits 1 and 0 set to 10).

Sampling Digital Lines for Change of State

In the Advanced Digital Interrupt modes, the digital lines are sampled at a rate set by the 8 MHz system clock or the clock programmed in User TC Counter 1. With each clock pulse, the digital circuitry looks at the state of the next Port 0 bits. To provide noise rejection and prevent erroneous interrupt generation because of noise spikes on the digital lines, a change in the state of any bit must be seen for two edges of a clock pulse to be recognized by the circuit.

7.4 Timer/Counters

The Timer/Counter section contains internal TCs in the Control EPLD and 8254 programmable interval timers for User Timer Counter.

7.4.1 THE INTERNAL TIMER/TIMER/COUNTERS

The internal Timer Counters works similar to the 8254 in rate mode. The SDM35541 has 8 internal timer/counter:

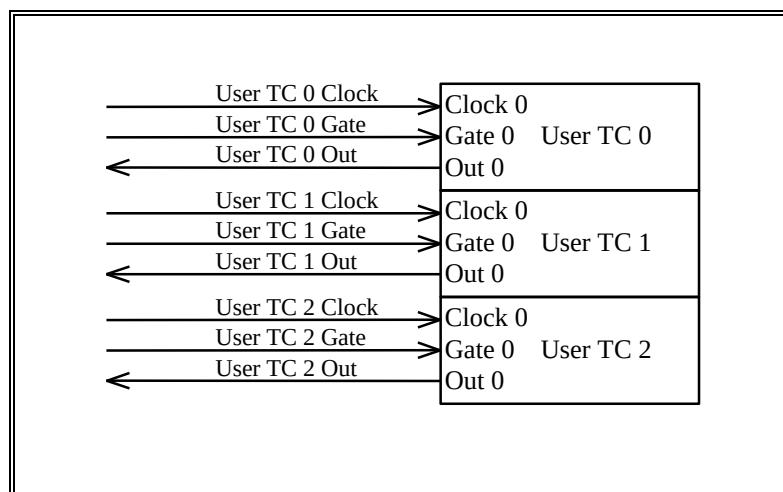
1. Pacer Clock - 24bit - Clock Signal is 8/20 MHz
2. Burst Clock - 16bit - Clock Signal is 8MHz
3. A/D Sample counter - 16-bit - Clock signal can be programmed
4. D/A1 Update counter - 16-bit - Clock signal is D/A1 update
5. D/A2 Update counter - 16-bit - Clock signal is D/A2 update
6. Delay Counter - 16-bit - Clock signal can be programmed
7. About Counter - 16-bit - Clock signal can be programmed
8. D/A Clock - 24-bit - Clock Signal is 8MHz

7.4.2 USER TIMER/TIMER/COUNTERS

The 8254 is the User TC. All three counters on this chip are available for user functions. For details on the programming modes of the 8254, see the data sheet in Appendix.

Each timer/counter has two inputs, CLK in and GATE in, and one output, timer/counter OUT. They can be programmed as binary or BCD down counters by writing the appropriate data to the command word, as described in the I/O map discussion in BAR2: SDM355xx Functional Block. See Figure 16. The sources of the user TC clocks and gates can be programmed by User Timer address area is the UTC0 ... UTC_CTRL. It is important, that the registers of the Timer Counter can be accessed by byte wide instructions. The 16-bit wide word must be created from the bytes.

Figure 16: User Timer/Counter



The timers can be programmed to operate in one of six modes, depending on your application. The following paragraphs briefly describe each mode.

Mode 0, Event Counter (Interrupt on Terminal Count)

This mode is typically used for event counting. While the timer/counter counts down, the output is low, and when the count is complete, it goes high. The output stays high until a new Mode 0 control word is written to the timer/counter.

Mode 1, Hardware-Retriggerable One-Shot

The output is initially high and goes low on the clock pulse following a trigger to begin the one-shot pulse. The output remains low until the count reaches 0, and then goes high and remains high until the clock pulse after the next trigger.

Mode 2, Rate Generator

This mode functions like a divide-by-N counter and is typically used to generate a real-time clock interrupt. The output is initially high, and when the count decrement to 1, the output goes low for one clock pulse. The output then

goes high again, the timer/counter reloads the initial count, and the process is repeated. This sequence continues indefinitely.

Mode 3, Square Wave Mode

Similar to Mode 2 except for the duty cycle output, this mode is typically used for baud rate generation. The output is initially high, and when the count decrements to one-half its initial count, the output goes low, for the remainder of the count. The timer/counter reloads and the output goes high again. This process repeats indefinitely.

Mode 4, Software-Triggered Strobe

The output is initially high. When the initial count expires, the output goes low for one clock pulse and then goes high again. Counting is “triggered” by writing the initial count.

Mode 5, Hardware Triggered Strobe (Retriggerable)

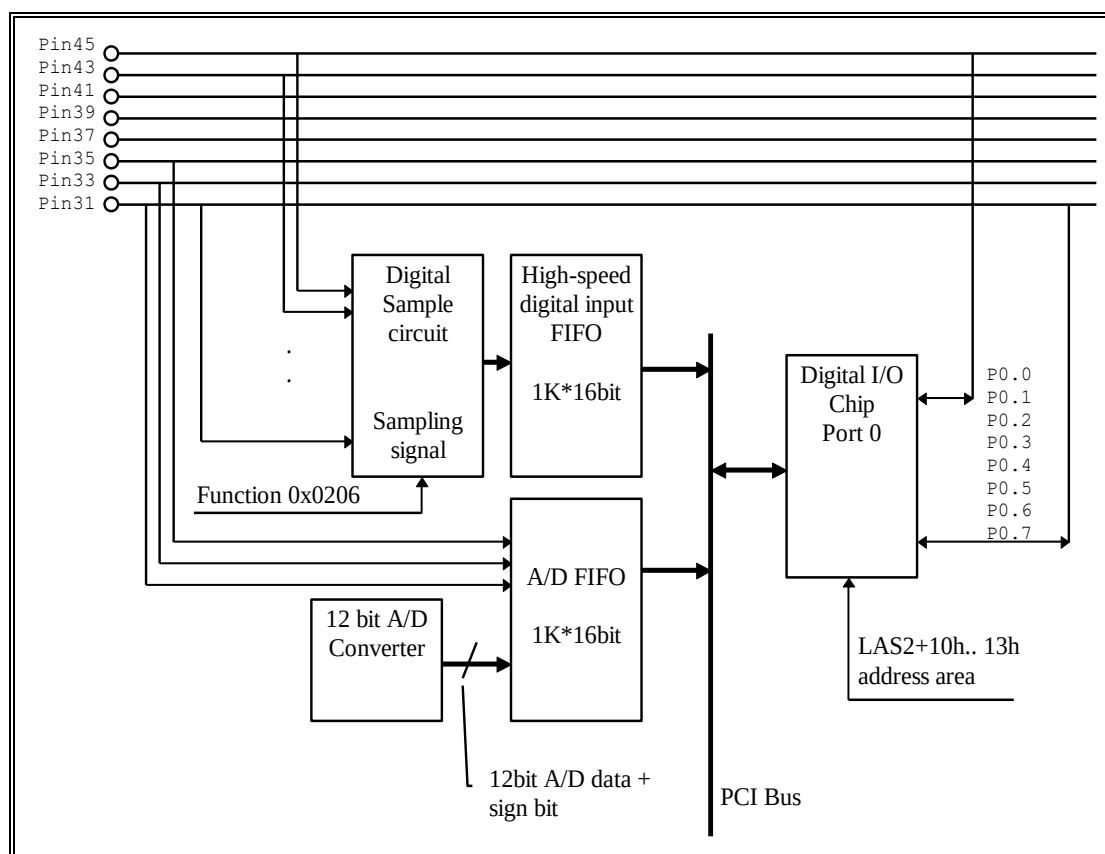
The output is initially high. Counting is triggered by the rising edge of the gate input. When the initial count has expired, the output goes low for one clock pulse and then goes high again.

7.5 Digital I/O

The SDM35541 has several digital lines to receive and transmit digital data from, or to the external digital world. This chapter describes only the 31...46 pins of External I/O connector. These 16 Digital Input/Output lines are multifunction and assure a flexible connection with the digital world.

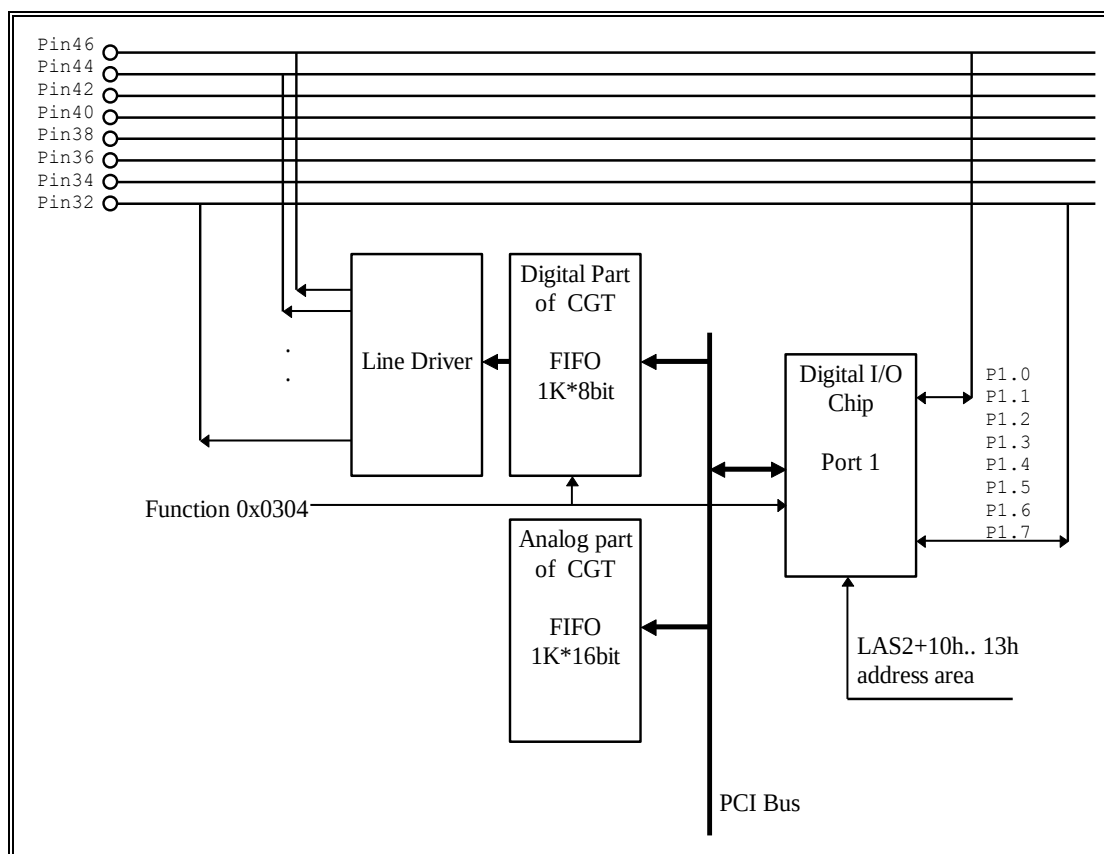
The connections of odd numbered pins are shown in the Figure 17. These lines are monitored by the high-speed digital input circuitry, and the least significant three bits of the A/D FIFO as data markers, therefore these lines can be read or driven by the Port 0 of Digital I/O Chip.

Figure 17: Digital I/O Port 0



The connections of even numbered pins are shown in the Figure 18. These lines can be driven by the digital part of the CGT; therefore, these lines can be read or driven by the Port 1 of Digital I/O Chip.

Figure 18: Digital I/O Port 1



7.5.1 THE DIGITAL I/O CHIP

The SDM35541 has 16 buffered TTL/CMOS digital I/O lines available for digital control applications. These lines are grouped in two 8-bit ports. The sixteen bits in Port 0 can be independently programmed as input or output. Port 1 can be programmed as 8-bit input or output ports. These lines are grouped in digital I/O chip with sixteen lines. The Digital I/O chip is addressed at DIO0... DIO_STATUS.

All digital inputs are pulled up to +5V by 10kΩ resistors. All digital outputs are terminated by series 10Ω resistors.

Port 0, Bit Programmable Digital I/O

The eight Port 0 digital lines are individually set for input or output by writing to the Direction Register at DIO_CTRL. The input lines are read and the output lines are written at DIO0.

Advanced Digital Interrupts: Mask and Compare Registers

The Port 0 bits support two Advanced Digital Interrupt modes. An interrupt can be generated when the data read at the port matches the value loaded into the Compare Register. This is called a match interrupt. Or an interrupt can be generated whenever any bit changes state. This is an event interrupt. For either interrupt, bits can be masked by setting the corresponding bit in the Mask Register high. In a digital interrupt mode, this masks out selected bits when monitoring the bit pattern for a match or event. In normal operation where the Advanced Digital Interrupt mode is not activated, the Mask Register can be used to preserve a bit's state, regardless of the digital data written to Port 0.

When using event interrupts, you can determine which bit caused an event interrupt to occur by reading the contents latched into the Compare Register.

Port 1, Port Programmable Digital I/O

The direction of the eight-bit Port 1 digital lines is programmed at DIO_STATUS, bit 2. These lines are configured as all inputs or all outputs, with their states read and written at DIO1.

Resetting the Digital Circuitry

When a digital chip clear (DIO_STATUS bits 1 and 0 = 00 followed by a write to DIO_CTRL), Software Reset of the board (BOARD_RESET), all of the digital I/O lines are set up as inputs.

Strobing Data into Port 0

When not in an Advanced Digital Interrupt mode, external data can be strobed into Port 0 by connecting a trigger pulse through the External Pacer Clock pin at the External I/O Connector. This data can be read from the Compare Register at DIO_CTRL.

7.5.2 HIGH-SPEED DIGITAL INPUT

As you can see in the Figure 17, the Pin 31...45 digital pins of external I/O connector can be sampled by high-speed digital input circuitry. The sampling signal can be selected by the HSDIN_START. Samples are written automatically into the high-speed digital input FIFO. The status bits of FIFO can be monitored at the address FIFO_STATUS.

If you want to get an interrupt at a required number of samples in the FIFO, use User Timer Counter 1 as high-speed digital input sample counter. Select the high-speed digital input sampling signal as clock of user TC1 by the UTC1_CLOCK.

All digital inputs are pulled up to +5V by 10kΩ resistors.

7.5.3 DIGITAL INPUT DATA MARKERS

As you can see in the Figure 17, the digital pin 31, 33, 35 can be sampled nearly simultaneously with the analog input signal by the A/D FIFO. The delay time between the analog input sampling and the digital input sampling is app. 800 ns. If you want to sample the digital lines with the analog lines really simultaneously, use the high-speed digital input with A/D Conversion Signal as sampling signal of high-speed digital input.

All digital inputs are pulled up to +5V by 10kΩ resistors.

7.6 On-board DSP

The SDM35541 dataModule has an on-board DSP which help the host CPU during following processes:

- Auto-calibration

The DSP is seamless, and the user doesn't need to program it to use auto-calibration.

8 Troubleshooting

If you are having problems with your system, please try the following initial steps:

- **Simplify the System** – Remove modules one at a time from your system to see if there is a specific module that is causing a problem. Perform your troubleshooting with the least number of modules in the system possible.
- **Swap Components** – Try replacing parts in the system one at a time with similar parts to determine if a part is faulty or if a type of part is configured incorrectly.

If problems persist, or you have questions about configuring this product, contact RTD Embedded Technologies via the following methods:

Phone: +1-814-234-8087
E-Mail: techsupport@rtd.com

Be sure to check the RTD web site (<http://www.rtd.com>) frequently for product updates, including newer versions of the board manual and application software.

9 Additional Information

9.1 PC/104 Specifications

A copy of the latest PC/104 specifications can be found on the webpage for the PC/104 Embedded Consortium:

www.pc104.org

9.2 PCI and PCI Express Specification

A copy of the latest PCI and PCI Express specifications can be found on the webpage for the PCI Special Interest Group:

www.pcisig.com

10 Limited Warranty

RTD Embedded Technologies, Inc. warrants the hardware and software products it manufactures and produces to be free from defects in materials and workmanship for one year following the date of shipment from RTD Embedded Technologies, Inc. This warranty is limited to the original purchaser of product and is not transferable.

During the one year warranty period, RTD Embedded Technologies will repair or replace, at its option, any defective products or parts at no additional charge, provided that the product is returned, shipping prepaid, to RTD Embedded Technologies. All replaced parts and products become the property of RTD Embedded Technologies. Before returning any product for repair, customers are required to contact the factory for a Return Material Authorization (RMA) number.

This limited warranty does not extend to any products which have been damaged as a result of accident, misuse, abuse (such as: use of incorrect input voltages, improper or insufficient ventilation, failure to follow the operating instructions that are provided by RTD Embedded Technologies, "acts of God" or other contingencies beyond the control of RTD Embedded Technologies), or as a result of service or modification by anyone other than RTD Embedded Technologies. Except as expressly set forth above, no other warranties are expressed or implied, including, but not limited to, any implied warranties of merchantability and fitness for a particular purpose, and RTD Embedded Technologies expressly disclaims all warranties not stated herein. All implied warranties, including implied warranties for merchantability and fitness for a particular purpose, are limited to the duration of this warranty. In the event the product is not free from defects as warranted above, the purchaser's sole remedy shall be repair or replacement as provided above. Under no circumstances will RTD Embedded Technologies be liable to the purchaser or any user for any damages, including any incidental or consequential damages, expenses, lost profits, lost savings, or other damages arising out of the use or inability to use the product.

Some states do not allow the exclusion or limitation of incidental or consequential damages for consumer products, and some states do not allow limitations on how long an implied warranty lasts, so the above limitations or exclusions may not apply to you.

This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

RTD Embedded Technologies, Inc.

103 Innovation Boulevard
State College, PA 16803 USA
Telephone: 814-234-8087
Fax: 814-234-5218

www.rtd.com

sales@rtd.com
techsupport@rtd.com

