The Embedded I/O Company

TEWS TECHNOLOGIES

TPMC530

16/8-Channel Isolated Simultaneous Sampling AD/DA

Version 1.1

User Manual

Issue 1.1.1

December 2017

TEWS TECHNOLOGIES GmbH
Am Bahnhof 7 25469 Halstenbek, Germany
ne: +49 (0) 4101 4058 0 Fax: +49 (0) 4101 4058 19
e-mail: info@tews.com www.tews.com

TPMC530-10R

16 Channels of 16 bit A/D, 8 Channels of 16 bit D/A, with front panel I/O

TPMC530-20R

8 Channels of 16 bit A/D, 4 Channels of 16 bit D/A, with front panel I/O

Style Conventions

Hexadecimal characters are specified with prefix 0x, i.e. 0x029E (that means hexadecimal value 029E).

For signals on hardware products, an 'Active Low' is represented by the signal name with # following, i.e. IP_RESET#.

Access terms are described as:

W	Write Only
R	Read Only
R/W	Read/Write
R/C	Read/Clear
R/S	Read/Set

Issue	Description	Date
1.0.0	Initial Issue	July 2014
1.0.1	Minor Clarifications	July 2014
1.0.2	Correction of Connector Description	August 2014
1.0.3	Added TPMC530-20R	June 2015
1.1.0	• Added "Internal Synchronization" to the ADC & DAC Configuration Register • Added "DAC Suspend on Frame End" to the DAC Configuration Register • Added "DAC Frame Interrupt" to the Interrupt Mask and Status Register • Added DAC Data Readback Register • Added Firmware Version Register • Added DMA "Bulk" and "Low Latency" Modes to the DMA Control / Status Register • Added FIFO Fill Level Register • Updated “6.1.1 DMA Controller” with block-mode support • Updated "6.1.2 DMA Descriptor" with new "EARLY" and "FRAME" control and status bits • Added “6.1.3 DMA FIFO Modes" • Updated "6.1.4 DMA Errors" with "EARLY" and "FRAME" control and status bits • Removed Write protection from Correction Value ROM	June 2017
1.1.1	Correction of Figure 7-1: Front Panel I/O Connector Pin Numbering	December 2017

Table of Contents

List of Figures

List of Tables

1 Product Description

The TPMC530 is a standard single width 32 bit PMC with faceplate I/O. The TPMC530-10R provides 16 channels of isolated 16 bit simultaneous sampling analog input and 8 channels of isolated 16 bit simultaneous update analog output. The TPMC530-20R provides 8 ADC channels and 4 DAC channels. All signals are accessible through a HD50 SCSI-2 type front I/O connector.

The ADC offers true differential inputs with software selectable ±5 V and ±10 V bipolar input voltage ranges (one setting for all channels). The sampling rate is up to 200 ksps and the ADC offers an oversampling capability with digital filter.

The DAC offers software selectable 0-5 V, 0-10 V, ±5 V and ±10 V output voltage ranges (one setting for all channels). The conversion time is typ. 10 µs and the DAC outputs are capable to drive a load of 2 kΩ, with a capacitance up to 4000 pF.

Both ADC and DAC offer programmable sample clocks. External synchronization is possible with a Trigger/Sync RS485 I/O.

Each ADC / DAC provides a 1024 sample input / output FIFO with programmable trigger levels. Data transfer on the PCI bus is handled by TPMC530 initiated scatter-gather DMA cycles with minimum host/CPU intervention.

Each TPMC530 is factory calibrated. The calibration can be automatically applied to data written to and read from the TPMC530.

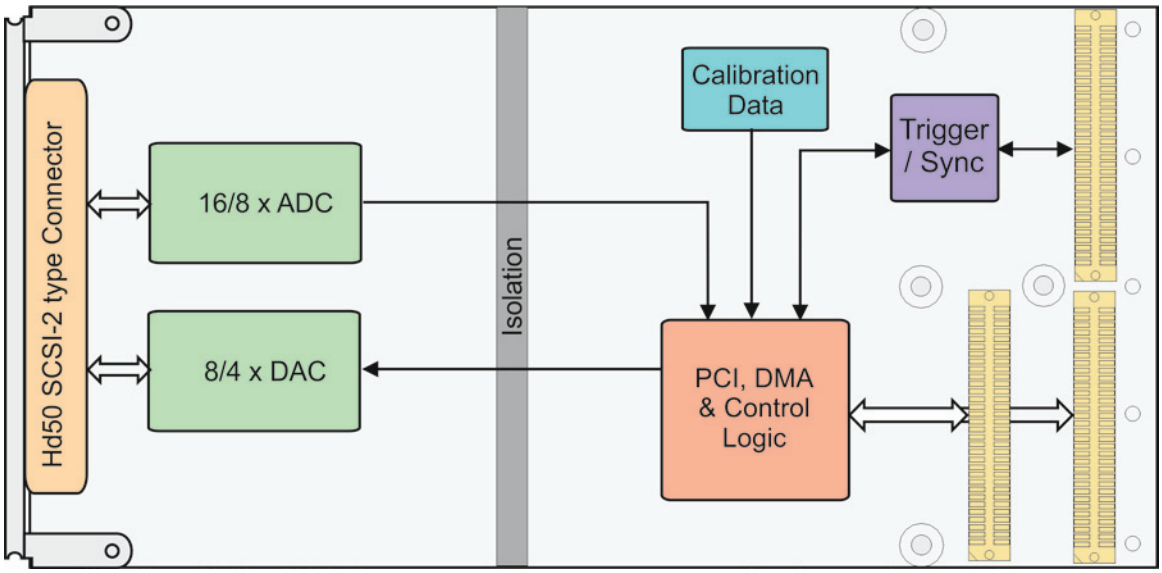


Figure 1-1 : Block Diagram

2 Technical Specification

PMC Interface	
Mechanical Interface	PCI Mezzanine Card (PMC) Interface ANSI/VITA 20-2001 (R2005) Single Size
Electrical Interface	PCI Rev. 3.0 compatible 33 MHz / 32 bit PCI 3.3 V and 5 V PCI Signaling Voltage
On Board Devices	
PCI Target Chip	Spartan-6 FPGA with PCI core
ADC	AD7609
DAC	AD5754R
Analog Input	
Resolution	16 Bit
Input Type	True bipolar differential input
Input Voltage Ranges	Software selectable: ±5 V, ±10 V Maximum ground related voltage on ADC input pins
Full Scale Range	Software Selectable: ±10 V, ±20 V
Sample Rate	200 ksps
Input Impedance	1 MΩ
Protection	Overvoltage protection up to ±16.5 V, 7 kV ESD rating
DNL/INL (typical)	±0.75 / ±3 LSB
FIFO Size	1024 samples
Analog Output	
Resolution	16 Bit
Output Type	Single-ended, software selectable: unipolar / bipolar
Output Voltage Ranges	Software selectable: ±5 V, ±10 V, +5 V, +10 V
Settling Time	10 µs
Output Load	Max. 2 kΩ | 4000 pF per channel
Protection	20 mA current limit, thermal shutdown option, 3.5 kV ESD rating
DNL/INL (typical)	±1 / ±16 LSB
FIFO Size	1024 samples
I/O Interface	
Number of Channels	16 (-10R) or 8 (-20R) simultaneous sampling ADC channels 8 (-10R) or 4 (-20R) simultaneous update DAC channels
I/O Connector	Front I/O: HD50 SCSI-2 type connector (e.g. AMP# 787395-5) PMC P14 I/O (64 pin Mezzanine Connector)

Physical Data		
Power Requirements	330 mA typical @ +5V DC	
Temperature Range	Operating Storage	-40 °C to +85 °C -40 °C to +85 °C
MTBF	544.000 h MTBF values shown are based on calculation according to MIL-HDBK-217F and MIL-HDBK-217F Notice 2; Environment: G_B 20°C. The MTBF calculation is based on component FIT rates provided by the component suppliers. If FIT rates are not available, MIL-HDBK-217F and MIL-HDBK-217F Notice 2 formulas are used for FIT rate calculation.	
Humidity	5 – 95 % non-condensing	
Weight	78 g	

Table 2-1 : Technical Specification

3 Address Map

3.1 PCI Configuration Space

PCI CFG Register Address	Write '0' to all unused (Reserved) bits								Initial Values (Hex Values)
	31	24	23	16	15	8	7	0	
0x00	Device ID				Vendor ID				0212 1498
0x04	Status				Command				0280 0000
0x08	Class Code						Revision ID		118000 00
0x0C	not supported		Header Type		PCI Latency Timer		not supported		00 00 00 00
0x10	Base Address Register 0 (BAR0)								FFFFFF00
0x14	Base Address Register 1 (BAR1)								FFFFFE00
0x18	not supported								00000000
0x1C	not supported								00000000
0x20	not supported								00000000
0x24	not supported								00000000
0x28	PCI CardBus Information Structure Pointer								00000000
0x2C	Subsystem ID				Subsystem Vendor ID				s.b. 1498
0x30	not supported								00000000
0x34	Reserved						New Cap. Ptr.		000000 00
0x38	Reserved								00000000
0x3C	Max_Lat		Min_Gnt		Interrupt Pin		Interrupt Line		00 00 01 00
0x40-0xFF	Reserved								00000000

Table 3-1 : PCI Configuration Space Header

Device-ID:	0x0212	TPMC530
Vendor-ID:	0x1498	TEWS TECHNOLOGIES
Subsystem-ID:	0x000A	-10R
	0x0014	-20R
Subsystem Vendor-ID:	0x1498	TEWS TECHNOLOGIES

3.2 PCI Memory Space

The TPMC530 maps its internal registers into two address spaces within the PCI memory space, using two Base Address Registers.

Base Address Register (BAR)	PCI Space Mapping	Size (Byte)	Port Width (Bit)	Endian Mode	Description
0	MEM	256	32	Little	Register Space
1	MEM	512	32	Little	Correction Value ROM

Table 3-2 : Memory Address Spaces

3.2.1 Register Space

The Register Space is accessible via the PCI Base Address Register 0 (BAR0).

Offset	Description	Size (Bit)
ADC Registers		
0x00	ADC Data Register 1 & 2	32
…	…	32
0x1C	ADC Data Register 15 & 16	32
0x20	ADC Configuration Register	32
0x24	ADC Control Register	32
0x28	ADC Conversion Start Register	32
0x2C	ADC Status Register	32
0x30	ADC Sample Clock	32
0x34 … 0x3C	Reserved	-
DAC Registers		
0x40	DAC Data Register 1 & 2	32
…	…	32
0x4C	DAC Data Register 7 & 8	32
0x50	DAC Configuration Register	32
0x54	DAC Control Register	32
0x58	DAC Load Register	32
0x5C	DAC Status Register	32
0x60	DAC Sample Clock	32
0x64 … 0x6C	Reserved	-
0x70	DAC Data Readback Register 1 & 2	32
…	…	32
0x7C	DAC Data Readback Register 7 & 8	32
General Registers		
0x80	Interrupt Mask Register	32

Offset	Description	Size (Bit)
0x84	Interrupt Status Register	32
0x88	External Trigger & Sync Register	32
0x8C	General Purpose Interval Timer 1 Control Register	32
0x90	General Purpose Interval Timer 1 Preload Register	32
0x94	General Purpose Interval Timer 1 Register	32
0x98	General Purpose Interval Timer 2 Control Register	32
0x9C	General Purpose Interval Timer 2 Preload Register	32
0xA0	General Purpose Interval Timer 2 Register	32
0xA4	Correction Control/Status Register	32
0xA8 … 0xBC	Reserved	-
DMA Registers		
0xC0	ADC DMA Control / Status	32
0xC4	ADC DMA Next Descriptor Pointer Register	32
0xC8	ADC DMA Current Descriptor Pointer Register	32
0xCC	DAC DMA Control / Status	32
0xD0	DAC DMA Next Descriptor Pointer Register	32
0xD4	DAC DMA Current Descriptor Pointer Register	32
0xD8	FIFO Fill Level Register	32
0xDC … 0xF8	Reserved	-
0xFC	Firmware Version Register	32

Table 3-3 : Register Space Address Map

3.2.2 Correction Value ROM

The Correction Value ROM is accessible via the PCI Base Address Register 1 (BAR1).

The correction values are determined during the factory acceptance test and correct module specific deviations.

There is an offset correction value and a gain correction value for each A/D and D/A channel at each voltage range. To achieve a higher accuracy, they are scaled to ¼ LSB. See the “Functional Description” chapter for the data correction formula.

The correction values are loaded from a serial EEPROM after power-up or PCI reset and are available approx. 5 ms after PCI reset. The correction values are stored consecutively in 16 bit values, in order of the A/D and D/A channels, starting with ADC channel 1.

If a system wide correction is necessary, the values in the Correction Value ROM can be overwritten. Write to the ADC or DAC Configuration Register to take over the new correction values to the internal correction.

Offset	Description	Voltage Range	Size (Bit)
0x000	ADC Channel 1 $Offset_{CORR}$	±5 V	16
0x002	ADC Channel 1 $Gain_{CORR}$	±5 V	16
…	…	…	
0x03C	ADC Channel 16 $Offset_{CORR}$	±5 V	16
0x03E	ADC Channel 16 $Gain_{CORR}$	±5 V	16
0x040	ADC Channel 1 $Offset_{CORR}$	±10 V	16
0x042	ADC Channel 1 $Gain_{CORR}$	±10 V	16
…	…	…	
0x07C	ADC Channel 16 $Offset_{CORR}$	±10 V	16
0x07E	ADC Channel 16 $Gain_{CORR}$	±10 V	16
0x080-0x0FE	Reserved	-	-
0x100	DAC Channel 1 $Offset_{CORR}$	±5 V	16
0x102	DAC Channel 1 $Gain_{CORR}$	±5 V	16
…	…	…	
0x11C	DAC Channel 8 $Offset_{CORR}$	±5 V	16
0x11E	DAC Channel 8 $Gain_{CORR}$	±5 V	16
0x120	DAC Channel 1 $Offset_{CORR}$	±10 V	16
0x122	DAC Channel 1 $Gain_{CORR}$	±10 V	16
…	…	…	
0x13C	DAC Channel 8 $Offset_{CORR}$	±10 V	16
0x13E	DAC Channel 8 $Gain_{CORR}$	±10 V	16
0x140	DAC Channel 1 $Offset_{CORR}$	+5 V	16
0x142	DAC Channel 1 $Gain_{CORR}$	+5 V	16
…	…	…	
0x15C	DAC Channel 8 $Offset_{CORR}$	+5 V	16
0x15E	DAC Channel 8 $Gain_{CORR}$	+5 V	16
0x160	DAC Channel 1 $Offset_{CORR}$	+10 V	16
0x162	DAC Channel 1 $Gain_{CORR}$	+10 V	16
…	…	…	
0x17C	DAC Channel 8 $Offset_{CORR}$	+10 V	16
0x17E	DAC Channel 8 $Gain_{CORR}$	+10 V	16

Table 3-4 : Correction Value Space Address Map

For the TPMC530-20R the correction values for ADC9 – ADC16 and DAC5 – DAC8 are reserved.

4 Register Description

4.1 ADC Registers

4.1.1 ADC Data Register

The ADC Data Registers always hold the latest data, independent of mode.

To accelerate data access, the ADC Data Registers are holding two ADC values. The upper register word holds the ADC channel with the higher number; the lower word holds the ADC channel with the lower number, i.e. ADC 2 & ADC 1.

Bit	Symbol	Description	Access	Reset Value
31:16	DATA	ADC data (i.e. ADC channel 2)	R	0x0000
15:0	DATA	ADC data (i.e. ADC channel 1)	R	0x0000

Table 4-1 : ADC Data Register

Due to the ADC's true differential inputs, the ADC output coding significantly differs compared to a single ended input.

Analogue to a single ended input, where the range setting directly describes the ground related output voltage range, the ADC range setting describes the range of ground related voltages that can be tied to the ADC differential inputs. This results in an extended input voltage range, since the ADC measures the voltage between the differential inputs VIN- and VIN+.

An Example: The ADC voltage range is ±10 V, so the allowed (single ended, ground related) voltage on each ADC input pin is ±10 V. When we examine the two largest differential voltages, we get following results:

VIN-	VIN+	ADC Input Value
-10 V	+10 V	+20 V
+10 V	-10 V	-20 V

Table 4-2 : ADC Data Coding Example

The example shows that the range of differential ADC input values is -20 V to +20 V, which results to a full scale range of 40 V for the ±10 V ADC Input Range setting. Similar, the full scale range for the ±5 V ADC Input Range setting is 20 V.

The data coding is two's complement.

Description	±10 V	±5 V	Digital Code
Full Scale Range	40 V	20 V	
Least Significant Bit	610.35 µV	305.18 µV	
Full Scale (pos.)	19.99939 V	9.999695 V	0x7FFF
FSR - 1 LSB	19.99878 V	9.99939 V	0x7FFE
Midscale + 1 LSB	610.35 µV	305.18 µV	0x0001
Midscale	0 V	0 V	0x0000
Midscale – 1 LSB	-610.35 µV	-305.18 µV	0xFFFF
-FSR + 1 LSB	-19.99939 V	-9.999695 V	0x8001
Full Scale (neg.)	-20 V	-10 V	0x8000

Table 4-3 : ADC Data Coding, Bipolar Input Range

Registers for ADCs that are not installed (ADC9 – ADC16 for TPMC530-20R) may contain invalid data and should be ignored.

4.1.2 ADC Configuration Register

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th><th>Access</th><th>Reset Value</th></tr>
<tr><td>31:10</td><td>-</td><td>Reserved
Set '0' for writes, undefined for reads.</td><td>-</td><td>-</td></tr>
<tr><td>9:7</td><td>OS</td><td>Oversampling Mode
If oversampling is active, the ADC takes multiple samples and averages them. This improves the Signal-to-Noise ratio. The ADC BUSY high time is extended until all samples are taken.

<table><tr><th>OS</th><th>Oversampling Ratio</th></tr><tr><td>000</td><td>No OS</td></tr><tr><td>001</td><td>2</td></tr><tr><td>010</td><td>4</td></tr><tr><td>011</td><td>8</td></tr><tr><td>100</td><td>16</td></tr><tr><td>101</td><td>32</td></tr><tr><td>110</td><td>64</td></tr><tr><td>111</td><td>Not valid</td></tr></table>
When changing the Oversampling Mode, a dummy conversion is required to set the ADC to the new Oversampling Ratio.</td><td>R/W</td><td>000</td></tr>
<tr><td>6:4</td><td>ADC DMA ENA</td><td>DMA Enable
When set to a value other than “000”, only a part of the available ADC channels is transferred to the FIFOs and used for DMA. The lower channel numbers are transferred, the higher channel numbers are omitted, dependent of the DMA ENA setting.
When set to “000”, the FIFO is disabled.

<table><tr><th>DMA ENA</th><th>Enabled ADC Channels</th></tr><tr><td>000</td><td>No channel (no DMA)</td></tr><tr><td>100</td><td>All 16 channels</td></tr><tr><td>101</td><td>Channels 1 – 12</td></tr><tr><td>110</td><td>Channels 1 – 8</td></tr><tr><td>111</td><td>Channels 1 – 4</td></tr></table>
Data from channels not enabled for DMA can still be read in the ADC Data Registers.</td><td>R/W</td><td>000</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th><th>Access</th><th>Reset Value</th></tr>
<tr><td>3:2</td><td>ADC SAMP</td><td>Sample Mode
<table><tr><th>SAMP</th><th>Sample Mode</th></tr><tr><td>00</td><td>Manual</td></tr><tr><td>01</td><td>Sample Clock</td></tr><tr><td>10</td><td>External Trigger</td></tr><tr><td>11</td><td>Internal Synchronization</td></tr></table>If set to “Manual”, a conversion is started with a write to the “Conversion Start Register”.
If set to “Sample Clock”, the ADC Sample Clock is used to start a conversion.
If set to “External Trigger”, the external Trigger input is used to start a conversion.
If set to “Internal Synchronization”, the DAC trigger is used to start a conversion. Do not set the ADC and DAC sample mode to “Internal Synchronization” at the same time as this would result in a lock-up.</td><td>R/W</td><td>00</td></tr>
<tr><td>1:0</td><td>ADC RANGE</td><td>ADC Input Range
This setting describes the allowed input voltage on the VIN± pins. See also chapter “ADC Data Register”.
<table><tr><th>RANGE</th><th>Input Voltage Range</th></tr><tr><td>00</td><td>±5V (10V p-p)</td></tr><tr><td>01</td><td>±10V (20V p-p)</td></tr><tr><td>10</td><td>reserved</td></tr><tr><td>11</td><td>reserved</td></tr></table></td><td>R/W</td><td>00</td></tr>
</table>

Table 4-4 : ADC Configuration Register

Allow a settling time of about 100µs when the ADC input range is changed.

4.1.3 ADC Control Register

Bit	Symbol	Description	Access	Reset Value
31:1	-	Reserved Set '0' for writes, undefined for reads.	-	-
0	RES	ADC Reset Setting this bit to ‘1’ resets the ADCs. The ADCs should be reset once after power-up. This bit is self-clearing.	W	0

Table 4-5 : ADC Control Register

4.1.4 ADC Conversion Start Register

The Conversion Start Register can be used to manually start a conversion of the ADCs.

Bit	Symbol	Description	Access	Reset Value
31:1	-	Reserved Set '0' for writes, undefined for reads.	-	-
0	ADC CONV	Write '1' to start a conversion. This bit is self-clearing.	W	0

Table 4-6 : ADC Conversion Start Register

4.1.5 ADC Status Register

Bit	Symbol	Description	Access	Reset Value
31:2	-	Reserved Set '0' for writes, undefined for reads.	-	-
1	FIFO OF	FIFO overflow The FIFO is full. Sampling will be stopped (Sample Mode is reset to Manual by setting ADC DMA ENA to "000" and ADC SAMP to "00") The DMA controller stays active and can be used to read out the remaining FIFO data.	R	0
0	ADC BUSY	ADC Busy. Indicates that a data conversion is in progress. If "Automatic sampling" is OFF, this bit is set by writing to the ADC Conversion Start Register. This bit must be read as '0' before the conversion data is read from the ADC Data Register. If "Automatic sampling" is ON, this bit still shows an ongoing conversion, but the conversion data can always be read.	R	0

Table 4-7 : ADC Status Register

4.1.6 ADC Sample Clock Register

When the “ADC SAMP” bits are set to “01”, the ADC Sample Clock enables automatic sampling in programmable intervals.

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th><th>Access</th><th>Reset Value</th></tr>
<tr><td>31:30</td><td>SC BASE</td><td>Sample Clock Time Base
<table><tr><th>SC BASE</th><th>Sample Clock Time Base</th></tr><tr><td>00</td><td>100 ns</td></tr><tr><td>01</td><td>1 µs</td></tr><tr><td>10</td><td>1 ms</td></tr><tr><td>11</td><td>reserved</td></tr></table></td><td>R/W</td><td>00</td></tr>
<tr><td>29:0</td><td>SC ADC</td><td>ADC Sample Clock
These bits set the starting value for the sample clock timer. The timer runs with the time base set by SC BASE.</td><td>R/W</td><td>0x0000 0000</td></tr>
</table>

Table 4-8 : ADC Sample Clock

The ADC Sample Clock is programmable from 0 to 1073741823 steps. Actual step is SC ADC x SC BASE. The ADC Sample Clock is derived from an on board oscillator.

Whenever the timer expires, a conversion of all ADCs is triggered.

Setting the ADC Sample Clock to values smaller than the ADC conversion time (~5µs) may result in undesired effects. Sample Clock triggers are suppressed as long ADC_BUSY is set.

If the ADC Sample Clock Register is set to ‘0’, the “Continuous Sampling Mode” is selected. A new conversion starts directly after the previous conversion has finished. With this mode the data registers are continually updated without the need to start a conversion. This is some kind of “don’t care mode”; the ADC data is constantly updated without CPU intervention. When read, a sample is not older than 5µs, but this mode does not guarantee equidistant sampling.

4.2 DAC Registers

4.2.1 DAC Data Register

To accelerate data access, the DAC Data Registers holds two DAC values. The upper register word holds the DAC channel with the higher number; the lower word holds the DAC channel with the lower number, i.e. DAC 2 & DAC 1.

Bit	Symbol	Description	Access	Reset Value
31:16	DATA	DAC data (i.e. DAC channel 2)	W	0x0000
15:0	DATA	DAC data (i.e. DAC channel 1)	W	0x0000

Table 4-9 : DAC Data Register

For bipolar output ranges, the data coding is two's complement.

Description	±10 V	±5 V	Digital Code
Full Scale Range	±10 V	±5 V	
Least Significant Bit	305.18 µV	152.59 µV	
Full Scale (pos.)	9.999695 V	4.999847 V	0x7FFF
FSR - 1 LSB	9.99939 V	4.999695 V	0x7FFE
Midscale + 1 LSB	305.18 µV	152.59 µV	0x0001
Midscale	0 V	0 V	0x0000
Midscale – 1 LSB	-305.18 µV	-152.59 µV	0xFFFF
-FSR + 1 LSB	-9.999695 V	-4.999847 V	0x8001
Full Scale (neg.)	-10 V	-5 V	0x8000

Table 4-10: DAC Data Coding, Bipolar Output Range

For unipolar output ranges, the data coding is straight binary.

Description	+10 V	+5 V	Digital Code
Full Scale Range	+10 V	+5 V	
Least Significant Bit	152.59 µV	76.295 µV	
Full Scale (pos.)	9.999847 V	4.999924 V	0xFFFF
FSR - 1 LSB	9.999695 V	4.999847 V	0xFFFE
Midscale + 1 LSB	5.000153 V	2.500076 V	0x8001
Midscale	5 V	2.5 V	0x8000
Midscale – 1 LSB	4.999847 V	2.499924 V	0x7FFF
-FSR + 1 LSB	152.59 µV	76.295 µV	0x0001
Full Scale (neg.)	0 V	0 V	0x0000

Table 4-11: DAC Data Coding, Unipolar Output Range

Registers for DACs that are not installed (DAC5 – DAC8 for TPMC530-20R) are still writeable, but take no effect.

4.2.2 DAC Configuration Register

A write to a DAC Configuration Register starts the DAC configuration (i.e. the DAC configuration register setting is transferred to the DAC internal configuration registers).

The DACs must be configured before they can be used. The DACs can only be used when the PU-bit is set.

When powered on, the DAC outputs are clamped to 0 V via a low impedance path until the first DAC load.

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th><th>Access</th><th>Reset Value</th></tr>
<tr><td>31:12</td><td>-</td><td>Reserved
Set '0' for writes, undefined for reads.</td><td>-</td><td>-</td></tr>
<tr><td>11</td><td>DAC SUSP ENA</td><td>DAC Suspend on Frame End Enable
When set, the DAC loads will be suspended when the Frame End bit is received while reading the FIFO and the FIFO is then empty. The FIFO will not underrun in this case. DAC loads will be suspended until new data is available in the FIFO. When new data is available, the DAC automatically continues with DAC loads using the Sample Mode set in DAC SAMP.
This bit will not prevent from FIFO underruns when the Frame End bit is not received while reading the FIFO.</td><td>R/W</td><td>0</td></tr>
<tr><td>10</td><td>TSD ENA</td><td>TSD Enable
Set to enable the DAC thermal shutdown feature.
Cleared to disable the thermal shutdown feature (default).
The DAC incorporates a thermal shutdown feature that automatically shuts down the device if the max device temperature is exceeded.</td><td>R/W</td><td>0</td></tr>
<tr><td>9</td><td>CLR SEL</td><td>DAC Clear Select
Selects the D/A Channel default values in case the DAC is cleared using the Clear Register.

<table><tr><th>CLR SEL</th><th>Unipolar Output Range</th><th>Bipolar Output Range</th></tr><tr><td>0</td><td>0V</td><td>0V</td></tr><tr><td>1</td><td>Mid-scale</td><td>Negative Full-scale</td></tr></table></td><td>R/W</td><td>0</td></tr>
<tr><td>8</td><td>PU</td><td>DAC Power-Up.
When set, this bit places the DACs in normal operating mode.
When cleared, this bit places the DACs in power-down mode (default).</td><td>R/W</td><td>0</td></tr>
<tr><td>7</td><td>ASR</td><td>Automatic Status Read
Set to enable automatic status reads of the DAC internal status register. Clear to disable automatic status reads.</td><td>R/W</td><td>0</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th><th>Access</th><th>Reset Value</th></tr>
<tr><td>6:4</td><td>DAC DMA ENA</td><td>DMA Enable
When set to a value other than “000”, only a part of the available DAC channels is used for DMA and transferred to the FIFOs. The lower channel numbers are transferred, the higher channel numbers are omitted, dependent of the DMA ENA setting.
When set to “000”, the FIFO is disabled.

<table><tr><th>DMA ENA</th><th>Enabled DAC Channels</th></tr><tr><td>000</td><td>No channel (no DMA)</td></tr><tr><td>100</td><td>All 8 channels</td></tr><tr><td>101</td><td>Channels 1 – 6</td></tr><tr><td>110</td><td>Channels 1 – 4</td></tr><tr><td>111</td><td>Channels 1 – 2</td></tr></table>
Channels not enabled for DMA get their data from the DAC Data Registers.</td><td>R/W</td><td>000</td></tr>
<tr><td>3:2</td><td>DAC SAMP</td><td>Sample Mode

<table><tr><th>SAMP</th><th>Sample Mode</th></tr><tr><td>00</td><td>Manual</td></tr><tr><td>01</td><td>Sample Clock</td></tr><tr><td>10</td><td>External Trigger</td></tr><tr><td>11</td><td>Internal Synchronization</td></tr></table>
If set to “Manual”, a conversion is started with a write to the “Conversion Start Register”.
If set to “Sample Clock”, the DAC Sample Clock is used to start a conversion.
If set to “External Trigger”, the external Trigger input is used to start a conversion.
If set to “Internal Synchronization”, the ADC trigger is used to start a conversion. Do not set the ADC and DAC sample mode to “Internal Synchronization” at the same time as this would result in a lock-up.</td><td>R/W</td><td>00</td></tr>
<tr><td>1:0</td><td>DAC RANGE</td><td>DAC Output Range

<table><tr><th>RANGE</th><th>Output Voltage Range</th></tr><tr><td>00</td><td>±5V</td></tr><tr><td>01</td><td>±10V</td></tr><tr><td>10</td><td>+5V</td></tr><tr><td>11</td><td>+10V</td></tr></table></td><td>R/W</td><td>00</td></tr>
</table>

Table 4-12: DAC Configuration Register

4.2.3 DAC Control Register

Bit	Symbol	Description	Access	Reset Value
31:2	-	Reserved Set '0' for writes, undefined for reads.	-	-
1	RDSTA	Read DAC Status Register When set, a request for reading the DAC status is logged and the status valid bit (SVAL) in the DAC Status Register is cleared. When the DAC status read is done, the DAC status register is updated and the status valid bit is set again. Reading the DAC status takes about 2.5 µs, but status read may be delayed by ongoing processes. This bit clears immediately.	R/S	0
0	CLR	When set, the DAC clear signal is asserted. Could be used to set all DAC channels (internal D/A Channel registers and analog outputs) to a value specified by the DACs CLR SEL setting (see DAC Configuration Register).	R/W	0

Table 4-13: DAC Control Register

4.2.4 DAC Load Register

The Load Register is used to load (=update) the analog outputs of the DACs.

Bit	Symbol	Description	Access	Reset Value
31:1	-	Reserved Set '0' for writes, undefined for reads.	-	-
0	LOAD	DAC Analog Output Update Request. When set, a request is logged to update the DAC analog outputs. When the request has been logged, the DAC analog outputs are updated when all available DAC Channel data has been transferred to the DAC internal data registers. The bit is automatically cleared when the DAC analog outputs are actually updated.	R/W	0

Table 4-14: DAC Load Register

4.2.5 DAC Status Register

The DAC devices provide an internal status register, which is reflected in the DAC Status Register. The DAC Status Registers are only updated when a status register read is ordered by writing a '1' to the RDSTA bit, when the ASR option is active or after a DAC Configuration Register write.

Some DAC status bits are available on a per channel basis, some are available for groups if four DAC channels. Refer to the status bit descriptions for details.

Bit	Symbol	Description	Access	Reset Value
31:27	-	Reserved Set '0' for writes, undefined for reads.	-	-
26	FIFO RDY	FIFO Ready To avoid instant FIFO underflows after DMA is enabled, any DAC updates are inhibited until the FIFO is filled with at least 4 DWORDs. Until this bit reads as '1', any DAC loads are discarded.	R	0
25	FIFO UF	FIFO Underflow The FIFO is empty. DAC update will be stopped (Sample Mode is reset to Manual by setting DAC DMA ENA to "000" and DAC SAMP to "00")). The DMA controller stays active.	R	0
24	SVAL	Status is valid This bit indicates that the other register bits are showing the result of a DAC status read (no random data). This bit is automatically set after the first DAC status read (e.g. during DAC configuration). The bit is cleared upon a DAC status read request via the DAC control register. The bit is then automatically set again, when the requested DAC status read is done.	R	0
23	DAC B SET	DAC B Settle. See DAC A SET description. This bit is for DAC channels 5-8.	R	0
22	DAC A SET	DAC A Settle Indicates DAC analog output settling time. '1' when DAC analog outputs are settling, '0' when DAC analog outputs are stable. This is no physical representation of any kind, just an internal timer that expires 10µs after an update of the DAC analog outputs. This bit is for DAC channels 1-4.	R	0
21	DAC B BUSY	DAC B Busy. See DAC A BUSY description. This bit is for DAC channels 5-8.	R	0
20	DAC A BUSY	DAC A Busy Set when a transfer to the DAC is in progress (DAC configuration, DAC channel data or status read). Clear when DAC control logic is in Idle state. This bit is for DAC channels 1-4.	R	0
19	DAC B TSD	DAC B Thermal Shutdown Alert This bit is for DAC channels 5-8. In the event of an over-temperature situation, this bit is set.	R	0
18	DAC A TSD	DAC A Thermal Shutdown Alert In the event of an over-temperature situation, this bit is set. This bit is for DAC channels 1-4.	R	0

Bit	Symbol	Description	Access	Reset Value
17	DAC B PUREF	DAC B Reference Power-Up This bit is for DAC channels 5-8. When set, this bit indicates that the DAC internal reference is powered-up. Since the DACs are operating with the internal reference, this bit should always be set for any status read from the DACs.	R	0
16	DAC A PUREF	DAC A Reference Power-Up When set, this bit indicates that the DAC internal reference is powered-up. Since the DACs are operating with the internal reference, this bit should always be set for any status read from the DACs. This bit is for DAC channels 1-4.	R	0
15	PU8	DAC Channel 8 Power-Up. See PU1 description.	R	0
14	PU7	DAC Channel 7 Power-Up. See PU1 description.	R	0
13	PU6	DAC Channel 6 Power-Up. See PU1 description.	R	0
12	PU5	DAC Channel 5 Power-Up. See PU1 description.	R	0
11	PU4	DAC Channel 4 Power-Up. See PU1 description.	R	0
10	PU3	DAC Channel 3 Power-Up. See PU1 description.	R	0
9	PU2	DAC Channel 2 Power-Up. See PU1 description.	R	0
8	PU1	DAC Channel 1 Power-Up ‘0’ when powered down, ‘1’ when powered up. On detection of an over-current condition, the DAC channel will power down automatically. PU1 will be cleared to reflect this.	R	0
7	OC8	DAC Channel 8 Over-current Alert. See OC1 description.	R	0
6	OC7	DAC Channel 7 Over-current Alert. See OC1 description.	R	0
5	OC6	DAC Channel 6 Over-current Alert. See OC1 description.	R	0
4	OC5	DAC Channel 5 Over-current Alert. See OC1 description.	R	0
3	OC4	DAC Channel 4 Over-current Alert. See OC1 description.	R	0
2	OC3	DAC Channel 3 Over-current Alert. See OC1 description.	R	0
1	OC2	DAC Channel 2 Over-current Alert. See OC1 description.	R	0
0	OC1	DAC Channel 1 Over-current Alert In the event of an over-current situation on DAC channel 1, this bit is set.	R	0

Table 4-15: DAC Status Register

4.2.6 DAC Sample Clock Register

When the “DAC SAMP” bits are set to “01”, the DAC Sample Clock enables automatic sampling in programmable intervals.

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th><th>Access</th><th>Reset Value</th></tr>
<tr><td>31:30</td><td>SC BASE</td><td>Sample Clock Time Base
<table><tr><th>SC BASE</th><th>Sample Clock Time Base</th></tr><tr><td>00</td><td>100 ns</td></tr><tr><td>01</td><td>1 µs</td></tr><tr><td>10</td><td>1 ms</td></tr><tr><td>11</td><td>reserved</td></tr></table></td><td>R/W</td><td>00</td></tr>
<tr><td>29:0</td><td>SC DAC</td><td>DAC Sample Clock
These bits set the starting value for the sample clock timer. The timer runs with the time base set by SC BASE.</td><td>R/W</td><td>0x0000 0000</td></tr>
</table>

Table 4-16: DAC Sample Clock Register

The DAC Sample Clock is programmable from 0 to 1073741823 steps. Actual step is SC DAC x SC BASE. The DAC Sample Clock is derived from an on board oscillator.

Whenever the timer expires, a conversion of both DACs is triggered.

Setting the DAC Sample Clock to values smaller than the DAC settling time (~10µs) may result in undesired effects. Sample Clock triggers are suppressed as long DAC_SET is set.

If the DAC Sample Clock Register is set to ‘0’, the “Continuous Sampling Mode” is selected. A new conversion starts directly after the previous conversion has finished. With this mode the DACs are continually updated from the data registers without the need to start a conversion. If a data register was not updated, the old value will be used.

4.2.7 DAC Data Readback Register

The DAC Data Readback Registers allow to read back the last value that was loaded into the DACs.

Bit	Symbol	Description	Access	Reset Value
31:16	DATA	DAC data (i.e. DAC channel 2)	W	0x0000
15:0	DATA	DAC data (i.e. DAC channel 1)	W	0x0000

Table 4-17 : DAC Data Readback Register

4.3 General Registers

4.3.1 Interrupt Mask Register

Bit	Symbol	Description	Access	Reset Value
31	IRQ ACK CONF	Interrupt Acknowledge Configuration 0 = Interrupts are acknowledged by writing '1' to the appropriate bit in the Interrupt Status Register 1 = Interrupts are cleared when the Interrupt Status Register is read	R/W	0
30:14	-	Reserved Set '0' for writes, undefined for reads.	-	-
13	DAC FRAME IRQ	Enable DAC Frame Interrupt 0 = DAC Frame IRQ disabled 1 = DAC Frame IRQ enabled An interrupt is asserted when the last datum of a descriptor with the FRAME bit set is loaded to the DAC.	R/W	0
12	DAC DMA ABORT IRQ	Enable DAC DMA Abort Received Interrupt 0 = DAC DMA Abort Received IRQ disabled 1 = DAC DMA Abort Received IRQ enabled An interrupt is asserted when the DMA controller received an PCI bus abort.	R/W	0
11	DAC DMA DONE IRQ	Enable DAC DMA Descriptor Complete Interrupt 0 = DAC DMA Descriptor Complete IRQ disabled 1 = DAC DMA Descriptor Complete IRQ enabled An interrupt is asserted when the DMA controller completes a descriptor and the IRQ flag is set in the descriptor.	R/W	0
10	ADC DMA ABORT IRQ	Enable ADC DMA Abort Received Interrupt 0 = ADC DMA Abort Received IRQ disabled 1 = ADC DMA Abort Received IRQ enabled An interrupt is asserted when the DMA controller received an PCI bus abort.	R/W	0
9	ADC DMA DONE IRQ	Enable ADC DMA Descriptor Complete Interrupt 0 = ADC DMA Descriptor Complete IRQ disabled 1 = ADC DMA Descriptor Complete IRQ enabled An interrupt is asserted when the DMA controller completes a descriptor and the IRQ flag is set in the descriptor.	R/W	0
8	TIME 2 IRQ	Enable Interval Timer 2 Interrupt 0 = Interval Timer 2 IRQ disabled 1 = Interval Timer 2 IRQ enabled An interrupt is asserted when the interval timer reaches zero	R/W	0
7	TIME 1 IRQ	Enable Interval Timer 1 Interrupt 0 = Interval Timer 1 IRQ disabled 1 = Interval Timer 1 IRQ enabled An interrupt is asserted when the interval timer reaches zero	R/W	0

Bit	Symbol	Description	Access	Reset Value
6	TRIG DA IRQ	Enable DAC Trigger Interrupt 0 = Trigger IRQ disabled 1 = Trigger IRQ enabled An interrupt will be generated when an external Trigger was detected. For pending interrupts and interrupt acknowledge see the Interrupt Status Register.	R/W	0
5	DAC FIFO ERR IRQ	Enable DAC FIFO Error Interrupt 0 = DAC FIFO Error IRQ disabled 1 = DAC FIFO Error IRQ enabled An interrupt is asserted when an DAC FIFO underflow is detected	R/W	0
4	DAC IRQ AL	Enable DAC Alert Interrupt 0 = DAC Alert IRQ disabled 1 = DAC Alert IRQ enabled Set when the DAC status is read and any of the over-current bits or the thermal alert bit is set and the interrupt is enabled. For pending interrupts and interrupt acknowledge see the Interrupt Status Register.	R/W	0
3	DAC IRQ SET	Enable after DAC Settle Interrupt 0 = IRQ after DAC Settle disabled 1 = IRQ after DAC Settle enabled An interrupt will be generated after the DAC has settled (DAC SET changes from 1 to 0). For pending interrupts and interrupt acknowledge see the Interrupt Status Register.	R/W	0
2	TRIG AD IRQ	Enable ADC Trigger Interrupt 0 = Trigger IRQ disabled 1 = Trigger IRQ enabled An interrupt will be generated when an external Trigger was detected. For pending interrupts and interrupt acknowledge see the Interrupt Status Register.	R/W	0
1	ADC FIFO ERR IRQ	Enable ADC FIFO Error Interrupt 0 = ADC FIFO Error IRQ disabled 1 = ADC FIFO Error IRQ enabled An interrupt is asserted when an ADC FIFO overflow is detected	R/W	0
0	ADC IRQ C	IRQ after Conversion Enable 0 = IRQ after Conversion disabled 1 = IRQ after Conversion enabled An interrupt will be generated after a conversion is finished and data is available (ADC_BUSY changes from 1 to 0). For pending interrupts and interrupt acknowledge see the Interrupt Status Register.	R/W	0

Table 4-18: Interrupt Mask Register

4.3.2 Interrupt Status Register

There are two ways to acknowledge interrupts:

1) Acknowledge selected Interrupts by writing a '1' to the appropriate interrupt status bit. This method is used when the IRQ ACK CONF bit in the Interrupt Mask Register is set to '0' (default).

2) Acknowledge all interrupts by reading this register. This method is used when the IRQ ACK CONF bit in the Interrupt Mask Register is set to '1'.

For an interrupt status bit to be set, the interrupt must be enabled prior to the interrupt event. An interrupt is asserted if at least one bit is set in the Interrupt Status Register.

Bit	Symbol	Description	Access	Reset Value
31:14	-	Reserved Set '0' for writes, undefined for reads.	-	-
13	DAC FRAME IRQ	DAC Frame Interrupt Status	R/C	0
12	DAC DMA ABORT IRQ	DAC DMA Abort Received Interrupt Status	R/C	0
11	DAC DMA DONE IRQ	DAC DMA Descriptor Complete Interrupt Status	R/C	0
10	ADC DMA ABORT IRQ	ADC DMA Abort Received Interrupt Status	R/C	0
9	ADC DMA DONE IRQ	ADC DMA Descriptor Complete Interrupt Status	R/C	0
8	TIME 2 IRQ	Interval Timer 2 Interrupt Status	R/C	0
7	TIME 1 IRQ	Interval Timer 1 Interrupt Status	R/C	0
6	TRIG DA IRQ	DAC Trigger Interrupt Status	R/C	0
5	DAC FIFO ERR IRQ	DAC FIFO Error Interrupt Status	R/C	0
4	DAC IRQ AL	DAC Alert Interrupt Status	R/C	0
3	DAC IRQ SET	Interrupt after DAC Settle Status	R/C	0
2	TRIG AD IRQ	ADC Trigger Interrupt Status	R/C	0
1	ADC FIFO ERR IRQ	ADC FIFO Error Interrupt Status	R/C	0

Bit	Symbol	Description	Access	Reset Value
0	ADC IRQ C	Interrupt after ADC Conversion Status	R/C	0

Table 4-19: Interrupt Status Register

4.3.3 External Trigger & Sync Register

Bit	Symbol	Description	Access	Reset Value
31:22	-	Reserved Set '0' for writes, undefined for reads.	-	-
21	DAC TERM ENA	Enable RS-485 Termination When this board is at the end of the synchronization line, the RS-485 termination should be enabled.	R/W	0
20	DAC SYNC OUT	Sync Output When SYNC ENA is set to “11” the SYNC output line can be used as general purpose output. In this case this bit sets output state of the SYNC output line.	R/W	0
19:18	DAC SNYC ENA	Enable External Sync If enabled, a synchronization impulse will be put out on the SNYC output line. SNYC ENA / Synchronization Source 00 – Sync disabled 01 – Load Register 10 – DAC sample clock 11 – General purpose output If Sync is disabled, the RS-485 driver will be in a high impedance state.	R/W	0
17	DAC TRIG POL	Trigger Polarity 0 = Rising Edge / Positive Pulse 1 = Falling Edge / Negative Pulse	R/W	0
16	DAC TRIG IN	Trigger Input This bit reflects the actual input state of the Trigger line at all times.	R	0
15:6	-	Reserved Set '0' for writes, undefined for reads.	-	-
5	ADC TERM ENA	Enable RS-485 Termination When this board is the end of the synchronization line, the RS-485 termination should be enabled.	R/W	0
4	ADC SYNC OUT	Sync Output When SYNC ENA is set to “11” the SYNC output line can be used as general purpose output. In this case this bit sets output state of the SYNC output line.	R/W	0

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th><th>Access</th><th>Reset Value</th></tr>
<tr><td>3:2</td><td>ADC SNYC ENA</td><td>Enable External Sync
If enabled, a synchronization impulse will be put out on the SNYC output line.

<table><tr><th>SNYC ENA</th><th>Synchronization Source</th></tr><tr><td>00</td><td>Sync disabled</td></tr><tr><td>01</td><td>Conversion Register</td></tr><tr><td>10</td><td>ADC sample clock</td></tr><tr><td>11</td><td>General purpose output</td></tr></table>
If Sync is disabled, the driver will be in a high impedance state.</td><td>R/W</td><td>0</td></tr>
<tr><td>1</td><td>ADC TRIG POL</td><td>Trigger Polarity
0 = Rising Edge / Positive Pulse
1 = Falling Edge / Negative Pulse</td><td>R/W</td><td>0</td></tr>
<tr><td>0</td><td>ADC TRIG IN</td><td>Trigger Input
This bit reflects the actual input state of the Trigger line at all times.</td><td>R</td><td>0</td></tr>
</table>

Table 4-20: Trigger & Sync Register

To accommodate the run-time of the synchronization impulse and transceiver delays, enable both the external sync and trigger modes and connect the trigger input with the sync output lines. This way the conversion signal will pass through the external transceiver and the conversion is really synchronous with other synchronized boards.

4.3.4 General Purpose Interval Timer Control Register

Bit	Symbol	Description	Access	Reset Value
31:1	-	Reserved Set '0' for writes, undefined for reads.	-	-
0	TIME EN	Enable Interval Timer 0 = disable Interval Timer 1 = enable Interval Timer	R/W	0

Table 4-21: General Purpose Interval Timer Control Register

The interval timer is a simple preloadable down-counter. A timer interrupt can be enabled in the Interrupt Mask Register. When the counter is enabled, it loads the value of TIME PRE and starts counting down. When the counter reaches 0, the interrupt is asserted. The counter wraps around to TIME PRE and continues to count until disabled.

The interval timer can be used as a reference timer in closed loop applications or as a timeout when using DMA with external ADC / DAC triggering.

4.3.5 General Purpose Interval Timer Preload Register

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th><th>Access</th><th>Reset Value</th></tr>
<tr><td>31:30</td><td>TIME BASE</td><td>Interval Timer Time Base
<table><tr><th>SC BASE</th><th>Sample Clock Time Base</th></tr><tr><td>00</td><td>100 ns</td></tr><tr><td>01</td><td>1 µs</td></tr><tr><td>10</td><td>1 ms</td></tr><tr><td>11</td><td>reserved</td></tr></table></td><td>R/W</td><td>00</td></tr>
<tr><td>29:0</td><td>TIME PRE</td><td>Interval Timer preload value.</td><td>R/W</td><td>0</td></tr>
</table>

Table 4-22: General Purpose Interval Timer Control Register

The Interval Timer is programmable from 0 to 1073741823 steps. Actual step is TIME PRE x TIME BASE. The Interval Timer Clock is derived from an on board oscillator.

4.3.6 General Purpose Interval Timer Register

Bit	Symbol	Description	Access	Reset Value
31:30	-	Reserved Set '0' for writes, undefined for reads.	-	-
29:0	TIME	Interval Timer value. This register holds the actual timer counter value	R/W	0

Table 4-23: General Purpose Interval Timer Register

4.3.7 Correction Control/Status Register

Bit	Symbol	Description	Access	Reset Value
31:18	-	Reserved Set '0' for writes, undefined for reads.	R	0
17	EEBSY	Read-Only Activity Status of the on board Correction Value EEPROM. 0: Correction Value EEPROM is not busy 1: Correction Value EEPROM is busy After power-up or PCI reset, the content of the Correction Value EEPROM is automatically copied to the Correction Value Data ROM. During this process, the EEBSY is set. The EEBSY bit is also set when the EELCK bit is cleared (from being set) while data is written to – or read from – the EEPROM. Software should once check that the EEBSY bit is '0' before reading data from the Correction Value Data Space.	R	0
16	EELCK	Correction Value EEPROM Lock Bit This bit must be set to unlock the access to the Correction Value EEPROM. When this bit is cleared (set from '1' to'0'), an automatic EEPROM update procedure starts. The content of the Correction Data ROM is stored in the on board Correction Value EEPROM, and is Immediately read back to the Correction Value Data ROM. The EEBSY bit is set during this procedure. Before setting the EELCK bit, software should check that the EEBSY bit is clear.	R/W	0
15:3	-	Reserved Set '0' for writes, undefined for reads.	-	0
1	COR RDY	Correction Ready This bit is set to '1' when the correction values are loaded and the internal correction is available.	R	0
0	COR ENA	Correction Enable 0: Internal Correction is disabled 1: Internal Correction is enabled	R/W	0

Table 4-24: Correction Control/Status Register

4.3.8 Firmware Version Register

Bit	Symbol	Description	Access	Reset Value
31:0	BUILD	Major & minor version, revision and build number of the FPGA firmware. Value depends on module version	R	-

Table 4-25: Firmware Version Register

4.4 DMA Registers

There are two DMA channels, one for the ADCs, and another for the DACs. Each channel has its own set of the three DMA registers.

4.4.1 DMA Control / Status Register

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th><th>Access</th><th>Reset Value</th></tr>
<tr><td>31:16</td><td>DMA COUNT</td><td>DWORDs served
Number of DWORDs that are written to the data buffer of the current descriptor</td><td>R</td><td>0</td></tr>
<tr><td>15:11</td><td>-</td><td>Reserved
Set '0' for writes, undefined for reads.</td><td>-</td><td>-</td></tr>
<tr><td>10</td><td>DMA ERR</td><td>DMA Error
A master abort occurred. Check the descriptor settings and reset the DMA channel to continue.</td><td>R</td><td>0</td></tr>
<tr><td>9</td><td>DMA DONE</td><td>Set to 1 when the end of the linked descriptor list is reached (EOL bit is set) or DMA is disabled</td><td>R</td><td>0</td></tr>
<tr><td>8</td><td>DMA IDLE</td><td>Idle
The DMA controller is in the idle state. No descriptor is loaded and the FIFOs are held inactive.</td><td>R</td><td>1</td></tr>
<tr><td>7:4</td><td>DMA TRSH</td><td>FIFO Threshold
Number of FIFO entries (in DWORDS) before requesting the PCI bus. Adjustable in steps of 32 entries, but with 16 entries as lowest threshold.
<table><tr><th>DMA TRSH</th><th>FIFO Threshold</th></tr><tr><td>0000</td><td>16</td></tr><tr><td>0001</td><td>32</td></tr><tr><td>0010</td><td>64</td></tr><tr><td>…</td><td>…</td></tr><tr><td>1110</td><td>448</td></tr><tr><td>1111</td><td>480</td></tr></table>Depending on if this is a read- (ADC) or write- (DAC) channel, the PCI bus is requested when the FIFO level rises above (read) or falls below (write) the FIFO threshold.</td><td>R/W</td><td>0000</td></tr>
<tr><td>3</td><td>DMA MODE</td><td>DMA FIFO Mode
0 = “Bulk” mode
1 = “Low-Latency” mode
See below for further details.</td><td>R/W</td><td>0</td></tr>
<tr><td>2</td><td>DMA ENDIAN</td><td>DMA endian mode
Set to 0 when the data is little endian, set to 1 when the data is big endian</td><td>R/W</td><td>0</td></tr>
<tr><td>1</td><td>DMA RESET</td><td>Reset DMA
Reset the DMA controller. When set to 1, the DMA controller stops, the IDLE bit is asserted and the FIFOs are disabled.
This bit is self-clearing</td><td>R/W</td><td>0</td></tr>
</table>

Bit	Symbol	Description	Access	Reset Value
0	DMA EN	Enable DMA When set to 1, the DMA controller loads the first descriptor and starts to do DMA transfers. When set to 0, further DMA transfers are suspended. When enabled again, the DMA controller continues from the point it was disabled.	R/W	0

Table 4-26: DMA Control / Status Register

The DMA Controller provides two FIFO modes: a “Bulk” and a “Low Latency” mode. The DMA FIFO Modes determine how the FIFO Threshold are used and can be used to control the PCI bus occupancy.

Both modes work differently for ADC and DAC.

DMA FIFO Mode	ADC	DAC
Bulk	The FIFO is filled up to the programmed DMA TRSH, then the DMA engine reads the FIFO until it is empty	When the FIFO level falls below DMA TRSH, the DMA engine fills the FIFO until it is full
Low Latency	The DMA engine reads the FIFO as soon as data is available	The DMA engine fills the FIFO as soon as the FIFO level falls below DMA TRSH. The FIFO is filled until DMA TRSH is reached again

Table 4-27: DMA FIFO Mode

Also refer to chapter “DMA FIFO Modes”.

4.4.2 DMA First Descriptor Pointer Register

Bit	Symbol	Description	Access	Reset Value
31:0	DMA POINT FIRST	Pointer to the first descriptor in the linked list.	R/W	0

Table 4-28: DMA First Descriptor Pointer Register

4.4.3 DMA Current Descriptor Pointer Register

Bit	Symbol	Description	Access	Reset Value
31:0	DMA POINT CUR	Pointer to the current descriptor in the linked list. In case of an error (a master abort was received because of an invalid descriptor or data buffer address) this register contains the address of the descriptor that caused the abort. Validate both the descriptor and the data buffer address before the DMA engine is restarted. When the descriptor list is expanded, but the DMA already reached the hitherto existing end of list, the DMA engine can be “unhold” with a write to this register. The DMA engine will then refetch the current descriptor und continues to process the descriptor list.	R/S	0

Table 4-29: DMA Current Descriptor Pointer Register

4.4.4 FIFO Fill Level Register

The FIFO Fill Level Register allows to read the current FIFO fill level. These values should be treated as approximation, not as exact values.

Bit	Symbol	Description	Access	Reset Value
31:25	-	Reserved Set '0' for writes, undefined for reads.	-	-
24:0	DAC FIFO CNT	This value shows the current DAC FIFO fill level in DWORDs	R	0
15:9	-	Reserved Set '0' for writes, undefined for reads.	-	-
8:0	ADC FIFO CNT	This value shows the current ADC FIFO fill level in DWORDs	R	0

Table 4-30: FIFO Fill Level Register

5 Configuration Hints

5.1 I/O Electrical Interface

5.1.1 ADC

All analog inputs are directly connected to ADC pins.

Protection	7 kV ESD rating on analog input channels ±16.5 V overvoltage protection
Input Impedance	1 MΩ
Input Capacitance	5 pF
Common-Mode Input Range	±5 V

Table 5-1 : ADC Electrical Interface

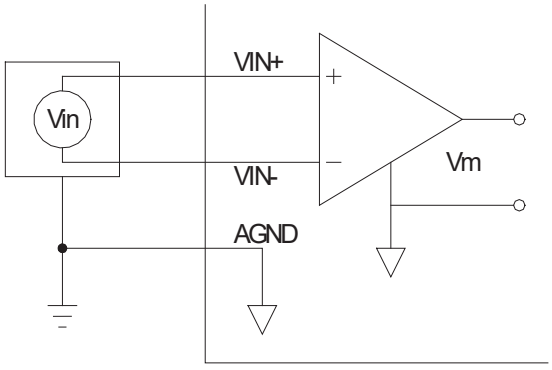
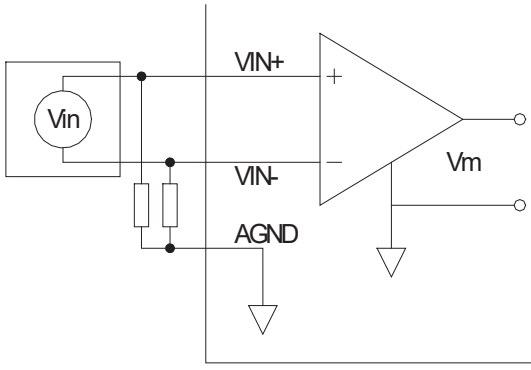
Differential with Ground Reference	Differential without Ground Reference
	

Table 5-2 : ADC Input Schemes

If signals without a ground reference should be connected, connect VIN+ and VIN- to AGND with a resistor to prevent the signal source to float out of the ADC's common-mode range. In most cases the VIN- connection suffices.

Unused ADC channels should be connected to AGND.

5.1.2 DAC

All analog outputs are directly connected to DAC pins.

DC Output Impedance	0.5 Ω
Load	2 kΩ
Capacitive Load	4000 pF

Table 5-3 : DAC Electrical Interface

5.1.3 DAC Overcurrent Protection

The TPMC530 uses the “Automatic Channel Power-Down” feature of the AD5754R. In case of an overload condition, the DAC channel is powered down and its output is clamped to ground with a resistance of ~4 kΩ. This condition is indicated by the OCx bit set to ‘1’ and the PUx set to ‘0’. The channel can be powered up again by setting the PUx bit back to ‘1’ after the overload condition is relieved.

6 Functional Description

6.1 DMA

Since the TPMC530 can produce quite large data volumes when running at full speed, bulk data transfers should be handled with PCI DMA transfers. The TPMC530's DMA controller uses the scatter/gather method, with a linked list of descriptors. The data is buffered with FIFOs so that delays in bus accesses don't affect ADC or DAC operation. When possible, the DMA controller uses bursts of 16 DWORDs to reduce the bus load.

The TPMC530 offers two DMA channels. These are dedicated, one channel to the ADCs, one channel to the DACs. Both channels are similar in structure and function, except that the ADC DMA channel reads from the TPMC530 and writes to the PCI memory space; the DAC DMA channel reads data from the PCI memory space and writes to the TPMC530.

6.1.1 DMA Controller

The DMA Controller stores and fetches data in and from small buffers in the host RAM that are linked together with a list of descriptors. Each descriptor contains some data fields with the buffer size, control and status bits and the location of the next descriptor in the list.

The CPU must prepare this linked list of descriptors. It may be done by dynamically allocating and linking descriptors and buffers as needed during runtime or by static predefined memory structures e.g. ring-chained-lists (the 'last' descriptor points back to the first descriptor). A mix of predefined descriptor lists but dynamically handled data buffers may also be an appropriate solution. This strategy depends on the specific application. The DMA controller provides multiple control mechanisms supporting all of these combinations in an efficient way.

The descriptors and data buffers can be stored in separate memory spaces within the 32-bit address range allowing full scatter/gather methods. Each descriptor contains a "next descriptor address" field to implement the linked list. Because the DMA controller cannot distinguish between valid and invalid addresses, an "End of List" mechanism is provided to prevent the DMA controller from branching to invalid memory locations.

The CPU will write the address of the first descriptor of each linked list to the "DMA First Descriptor Pointer Register" during the DMA initialization procedure. The DMA controller starts processing the descriptor list by fetching the first descriptor from this address.

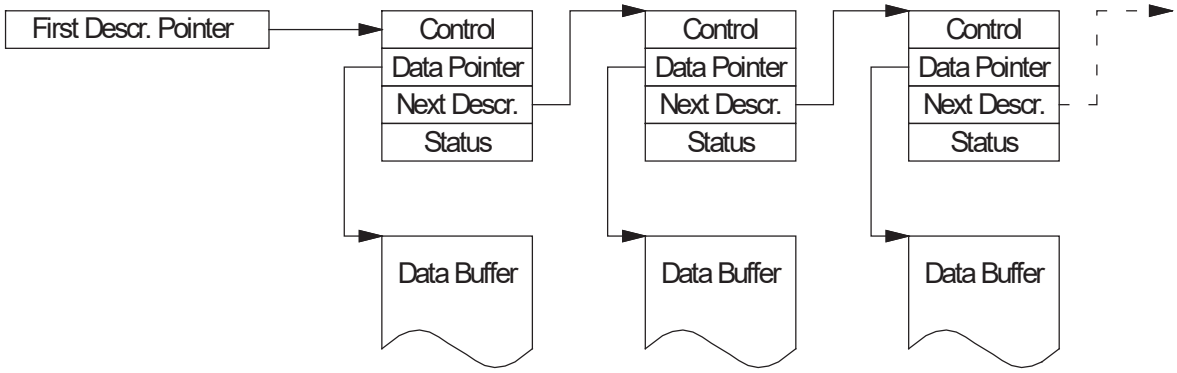


Figure 6-1 : Linked Descriptor List

A typical DMA application flow looks like this:

- Allocate and initialize memory
- Build the linked list of DMA descriptors
- Initialize the “DMA First Descriptor Pointer Register” Register with the address of the first descriptor in the linked list
- Enable DMA

The DMA engine loads a descriptor and starts to read/write from/to the specified memory address. After a descriptor is served, the DMA engine checks if the descriptor is the last descriptor in the linked list. If the end is reached, the DMA engine stops. Otherwise it loads the next descriptor. In both cases an interrupt can be asserted. Before the DMA engine continues, it writes a status word into the processed descriptor.

The last descriptor in the list can be a “null descriptor” where only the “End Of List”-bit is set. A new list can be appended to the existing list by removing the “EOL”-bit and setting a valid “Next Descriptor Pointer” address, which would be the first descriptor of the new list. If the DMA engine is stopped because it already reached the end of the descriptor list, it can be “unhold“ with a write to the DMA Current Descriptor Pointer Register, otherwise it will just continue with the appended list. When the DMA engine is “unholded”, it refetches the current descriptor and serves it as usual.

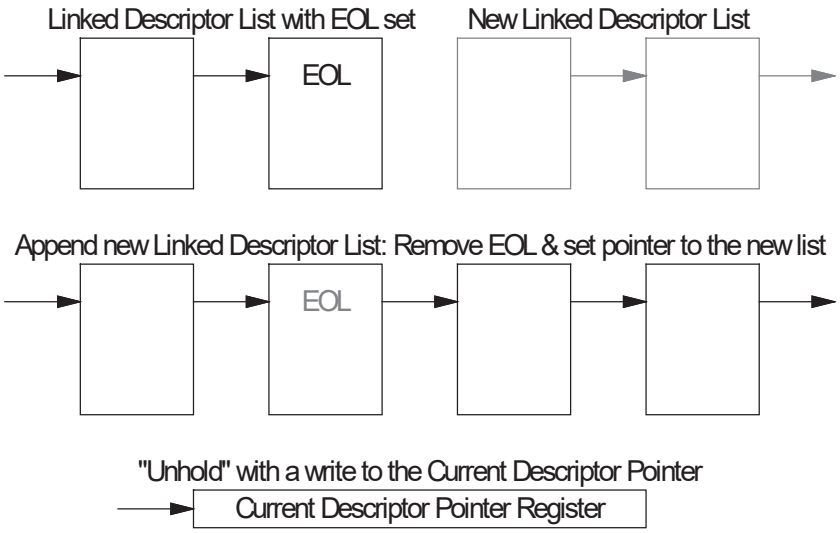


Figure 6-2 : Linked Descriptor List Expansion

A typical descriptor list expansion flow looks like this:

- Allocate and initialize additional memory
- Build the additional linked list of DMA descriptors
- Append the additional descriptors to the list by modifying the previous last descriptor to point to the new list of descriptors
- “Unhold” the DMA engine with a write to the DMA Current Descriptor Pointer Register

Simple block-wise DMA transfers can be implemented by using a single descriptor with SIZE and EOL-bit set. When the descriptor is served, the DMA engine is stopped. The descriptor can be reused by simply “unholding” the DMA engine.

6.1.2 DMA Descriptor

Each descriptor consists of four consecutive 32-bit little endian data fields that are located in the host RAM and that must be 32-bit (DWORD)-aligned. The first three data fields are written by the CPU and read by the DMA controller when branching from a previous descriptor. The descriptor provides information about the next descriptor in the linked list, the attached data buffer address and size and some control bits.

The fourth data field is written by the DMA controller indicating that operation on this descriptor is finished.

DWORD	31:16	15:3	2	1	0
0	SIZE	0	FRAME	IRQ	EOL
1	Data Buffer Base Address Pointer				
2	Next Descriptor Pointer				
3	USED	0		EARLY	DONE

Table 6-1 : DMA Descriptor

SIZE: Size of the DMA target data buffer in DWORDs (255 kB max).

FRAME: For DAC descriptors this bit can be used to assert an interrupt when the last data sample is loaded into the DAC. If the FIFO is empty, further DAC loads can be suspended until new data is available (i.e. via “unholding” the DMA engine) by enabling DAC Suspend on Frame End (DAC SUSP ENA) in the DAC Configuration Register. For descriptors with SIZE = 0 this bit takes no effect.

IRQ: If ‘1’, an interrupt is asserted when the descriptor is served. The interrupt must be enabled in the “Interrupt Mask Register”.

EOL: End of List. If ‘1’, this is the last descriptor in the list. When the descriptor is served, the DONE bit will be set in the “DMA Control / Status Register”.

USED: Number of DWORDs in the data buffer that contains valid data.

EARLY: The descriptor is finished because the ADC turned sampling off. This can be due to an overflow error or because sampling was turned off in the Configuration Register.

DONE: Data buffer size reached, descriptor served

6.1.3 DMA FIFO Modes

The DMA Controller provides two FIFO modes: a “Bulk” and a “Low Latency” mode. The DMA FIFO Modes determine how the FIFO Threshold are used and can be used to control the PCI bus occupancy.

Both modes work differently for ADC and DAC.

ADC

When the “Bulk” mode is selected, the FIFO will be filled by the ADC. When the DMA Threshold is reached, the FIFO is read by the DMA Controller until it is empty. Setting the DMA Threshold to a higher value would result in less frequent but more extended PCI bus requests.

When the “Low Latency” mode is selected, the DMA Controller reads the FIFO as soon as data is available. This results in more frequent but shorter PCI bus requests.

DAC

When the “Bulk” mode is selected, the DMA Controller initially fills the FIFO until it is full. When the FIFO level falls below the DMA Threshold, the FIFO is filled again until it is full. Setting the DMA Threshold to a lower value would result in less frequent but more extended PCI bus requests.

When the “Low Latency” mode is selected, the DMA Controller initially fills the FIFO until the DMA Threshold is passed over. When the FIFO level falls below the DMA Threshold, the FIFO is filled again until the DMA Threshold is passed over. This results in more frequent but shorter PCI bus requests. Setting the DMA Threshold controls the amount of data that is buffered in the FIFO.

6.1.4 DMA Errors

Running DMA bears two errors sources: Invalid addresses or FIFO errors.

Invalid addresses

When the DMA engine accesses invalid descriptor or data buffer addresses, PCI aborts are raised. PCI aborts are fatal for the DMA engine. In this case the DMA Current Descriptor Pointer Register holds the address of the descriptor that caused the abort. Both the descriptor and the data buffer address should be validated before the DMA engine is restarted.

FIFO errors

Two FIFO errors can occur when the DMA controller runs: a FIFO overrun on the ADC DMA channel, and a FIFO underrun on the DAC DMA channel. These errors can result from excessive PCI bus loading, so that the DMA controller is not able to read or write as fast as required and will always indicate a data loss. If enabled, an interrupt will be asserted by the ADC or DAC to inform the host of the data loss.

FIFO errors inhibit the DMA engine. If the DMA engine starts again, it will continue from the point it was stopped.

In case of a FIFO overrun on the ADC DMA channel, the ADC tries to submit a status bit over the FIFO. When the DMA controller reads this status bit, it finishes the current active descriptor by setting the DONE and the EARLY bit. The SIZE field then indicates the amount of valid data in the descriptor. This provides a way to gather all data that was valid before the FIFO overrun occurred.

FIFO underruns on the DAC DMA channel can be inhibited when the descriptor’s FRAME bit is set and the “Suspend on Frame End” option is enabled in the DAC Configuration Register. In this case, the DAC loads will be suspended when the Frame End bit is received while reading the FIFO and the FIFO is then empty. The FIFO will not underrun in this case. DAC loads will be suspended until new data is available in the FIFO, after which the DAC will automatically continue with DAC loads.

This can be used when no continuous DAC output is required, but only a data burst every now and then. To start a new data burst just append a new descriptor and unhold the DMA controller.

6.1.5 DMA Endianness

Per default, the data is assumed to be little endian. When DMA ENDIAN is set to 1, the DMA controller swaps the bytes within an ADC / DAC data word from little to big endian, but maintains the order of the data words.

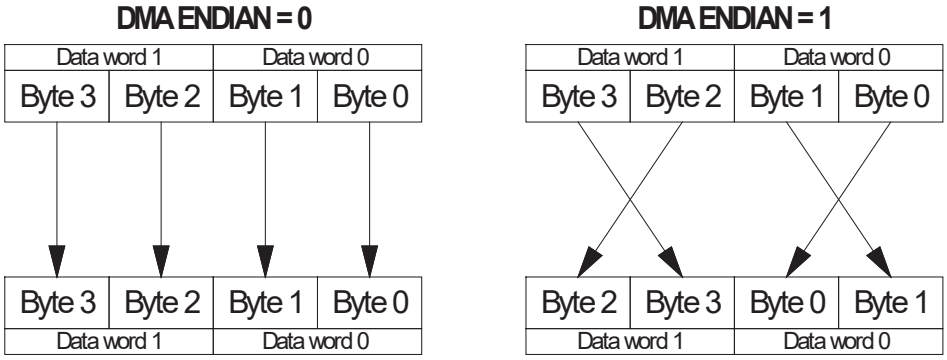


Figure 6-3 : Endian Byte Swaps

6.1.6 DMA Data Buffer Structure

The ADC & DAC data is stored in consecutive “samples”. A sample contains all channels that are enabled for DMA.

The following examples show the DMA Data Buffer Structure with a DMA ENDIAN setting according to the system architecture and for 16-bit word accesses.

Example ADC (all channels enabled):

Base Address	ADC1	ADC2	ADC3	ADC4	ADC5	ADC6	ADC7	ADC8	
Base Address +10	ADC9	ADC10	ADC11	ADC12	ADC13	ADC14	ADC15	ADC16	1st Sample
Base Address +20	ADC1	ADC2	ADC3	ADC4	ADC5	ADC6	ADC7	ADC8	
Base Address +30	ADC9	ADC10	ADC11	ADC12	ADC13	ADC14	ADC15	ADC16	2nd Sample
...	...								...

Example DAC (all channels enabled):

Base Address	DAC1	DAC2	DAC3	DAC4	DAC5	DAC6	DAC7	DAC8	1st Sample
Base Address +10	DAC1	DAC2	DAC3	DAC4	DAC5	DAC6	DAC7	DAC8	2nd Sample
...	...								...

If not all channels are enabled for DMA, the sample is accordingly smaller.

Example ADC (only channels 1-8 enabled):

Base Address	ADC1	ADC2	ADC3	ADC4	ADC5	ADC6	ADC7	ADC8	1st Sample
Base Address +10	ADC1	ADC2	ADC3	ADC4	ADC5	ADC6	ADC7	ADC8	2nd Sample
...	...								...

6.2 Correction

There are two errors that affect the DC accuracy of the ADCs and DACs.

Offset Error:

The Offset Error is the deviation from an ideal zero. For ADCs it is the data value when converting with the input connected to its own ground in single-ended mode, or with shorted inputs in differential mode. For DACs it is the difference between the ideal and actual DAC output with zero code as digital input. This error is corrected by subtracting the known error from the datum.

Gain Error:

The Gain Error is the deviation in slope of the ideal ADC or DAC transfer characteristic. This error is corrected by multiplying the datum with the correction factor.

The TPMC530 provides offset and gain correction values for each channel and voltage range.

The correction values are obtained during factory calibration and are stored in an on board EEPROM as 2-complement 2-byte-wide values in the range from -32768 to +32767. To achieve a higher accuracy, they are scaled to ¼ LSB.

6.2.1 Internal Correction

The TPMC530 provides an internal correction feature. The ADC and DAC offset and gain errors can be corrected with correction values loaded from the Correction Value ROM. The correction is applied to data written to and read from the TPMC530 when the Correction Enable bit is set in the Correction Control/Status Register. This is done by the TPMC530 hardware, and causes no further off-module calculation.

Due to rounding errors caused by the internal fixed point calculations used for the correction an additional error of up to 1 LSB is added to the ADC & DAC data. This error can be avoided when the correction is done off-module using the correction values provided by the ADC/DAC Correction Value ROM and the appropriate floating point or scaled integer arithmetic.

6.2.2 Off-Module Correction

The correction can be done off-module in software, using the following correction formula:

$$Data = Value \cdot \left(1 - \frac{Gain_{corr}}{262144}\right) - \frac{Offset_{corr}}{4}$$

Figure 6-4 : Correction Formula

Value is the uncorrected datum (for AD channels this is the ADC reading, for DA channels this is the desired output value).

Data is the corrected result (for AD channels this is the ADC input value, for DA channels this is the value that has to be written to the DAC to achieve the desired output value).

$Gain_{corr}$ and $Offset_{corr}$ are the correction values stored in the Calibration Value ROM.

Floating point arithmetic or scaled integer arithmetic must be used to avoid rounding errors in computing above formula.

Due to inherent device deviation, the extremes of the full scale range may not be fully reachable, even after calibration.

6.3 Trigger and Sync

The Trigger input and Sync output can be used to synchronize TPMC530 modules. One TPMC530 would be the master that has its Sync output enabled; the other modules would be the slaves.

The Trigger input and Sync output can also be used as a simple interrupt capable general purpose input and output.

The signal standard for the Trigger and Sync is RS-485. The transceiver provides a switchable termination; the module that is at the end of a synchronization chain should enable the termination. A trigger impulse must be at least 25 ns long to be detected. The Sync output pulse duration is 100 ns.

To accommodate the run-time of the synchronization impulse and transceiver delays, enable both the external sync and trigger modes and connect the trigger input with the sync output lines. This way the conversion signal will pass through the external transceiver and the conversion is really synchronous with other synchronized boards.

6.4 Continuous Sampling Mode

When the sample timing is not crucial but recent data must always be available, a “continuous sampling mode” can be selected by setting “Sample Mode” to “Automatic” while the Sample Clock is null (or smaller than 5µs). In this mode the ADCs/DACs are continuously converting. ADCs will write the conversion data into the ADC Data Registers, the DACs will use the values from the DAC Data Registers, using the old values if they are not updated.

This mode allows to always having recent data available without the need of loading the CPU with starting and controlling the conversions.

7 Pin Assignment – I/O Connector

7.1 Front Panel I/O Connector

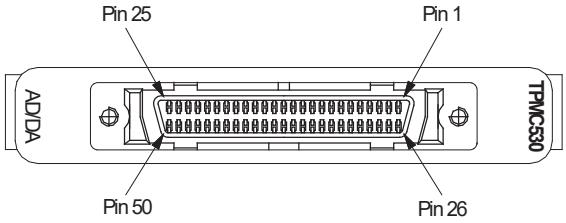


Figure 7-1 : Front Panel I/O Connector Pin Numbering

The TPMC530 front panel I/O connector is a HD50 SCSI-2 type female connector (e.g. AMP# 787395-5).

Pin	Signal	Pin	Signal
1	AGND	26	AGND
2	VIN 1+	27	VIN 1-
3	VIN 2+	28	VIN 2-
4	VIN 3+	29	VIN 3-
5	VIN 4+	30	VIN 4-
6	VIN 5+	31	VIN 5-
7	VIN 6+	32	VIN 6-
8	VIN 7+	33	VIN 7-
9	VIN 8+	34	VIN 8-
10	VIN 9+	35	VIN 9-
11	VIN 10+	36	VIN 10-
12	VIN 11+	37	VIN 11-
13	VIN 12+	38	VIN 12-
14	VIN 13+	39	VIN 13-
15	VIN 14+	40	VIN 14-
16	VIN 15+	41	VIN 15-
17	VIN 16+	42	VIN 16-
18	AGND	43	AGND
19	VOUT 1	44	VOUT 2
20	VOUT 3	45	VOUT 4
21	VOUT 5	46	VOUT 6
22	VOUT 7	47	VOUT 8
23	AGND	48	AGND
24	AGND	49	AGND
25	AGND	50	AGND

Table 7-1 : Pin Assignment Front Panel I/O Connector

7.2 P14 Back I/O Connector

Pin	Signal	Pin	Signal
1	GND	33	-
2	GND	34	-
3	ADC Trigger+ (in)	35	-
4	ADC Trigger- (in)	36	-
5	ADC Sync+ (out)	37	-
6	ADC Sync- (out)	38	-
7	GND	39	-
8	GND	40	-
9	DAC Trigger+ (in)	41	-
10	DAC Trigger- (in)	42	-
11	DAC Sync+ (out)	43	-
12	DAC Sync- (out)	44	-
13	GND	45	-
14	GND	46	-
15	-	47	-
16	-	48	-
17	-	49	-
18	-	50	-
19	-	51	-
20	-	52	-
21	-	53	-
22	-	54	-
23	-	55	-
24	-	56	-
25	-	57	-
26	-	58	-
27	-	59	-
28	-	60	-
29	-	61	-
30	-	62	-
31	-	63	-
32	-	64	-

Table 7-2 : Pin Assignment Front Panel I/O Connector