The Embedded I/O Company

TEWS TECHNOLOGIES

TPMC550

8/4 Channels of Isolated 12 bit D/A

Version 1.1

User Manual

Issue 1.1.9

August 2014

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

TPMC550-10R

8 channels of isolated 12 bit D/A, front panel I/O

TPMC550-11R

4 channels of isolated 12 bit D/A, front panel I/O

TPMC550-20R

8 channels of isolated 12 bit D/A, P14 I/O

TPMC550-21R

4 channels of isolated 12 bit D/A, P14 I/O

This document contains information, which is proprietary to TEWS TECHNOLOGIES GmbH. Any reproduction without written permission is forbidden.

TEWS TECHNOLOGIES GmbH has made any effort to ensure that this manual is accurate and complete. However TEWS TECHNOLOGIES GmbH reserves the right to change the product described in this document at any time without notice.

TEWS TECHNOLOGIES GmbH is not liable for any damage arising out of the application or use of the device described herein.

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	First Issue	May 1999
1.1	Additions to Technical Specifications	March 2000
1.2	General Revision	January 2003
1.3	DVRx bit Descriptions corrected in DAC Status Register	July 2003
1.4	New address TEWS LLC	September 2006
1.5	New Hardware Version	October 2007
1.1.6	New notation of User Manual Issue	November 2008
1.1.7	PCI 9030 Header correction	February 2009
1.1.8	Remove Calibration EEPROM Lock Register from Register Space Address Map	February 2011
1.1.9	General Revision	August 2014

Table of Contents

List of Figures

List of Tables

1 Product Description

The TPMC550 is a standard single-width 32 bit PMC module providing isolated 12 bit analog outputs.

The number of D/A channels and the I/O connector depends on the module option:

TPMC550-10R:	8 channels of isolated 12 bit D/A, front panel I/O
TPMC550-11R:	4 channels of isolated 12 bit D/A, front panel I/O
TPMC550-20R:	8 channels of isolated 12 bit D/A, P14 I/O
TPMC550-21R:	4 channels of isolated 12 bit D/A, P14 I/O

The settling time to 0.012% is 10µs maximum. It supports immediate and simultaneous load.

A sequencer mode allows periodically update of D/A output channels. The sequence timer range extends from 100 µs to 6.5 s.

The programmable output voltage range is +/- 10V or 0 - 10V selectable by jumper configuration for a group of 4 channels.

An on board DC/DC converter powers the isolated DAC and the output buffers.

Each TPMC550 is factory calibrated. The calibration information is stored in an EEPROM space unique to each PMC module.

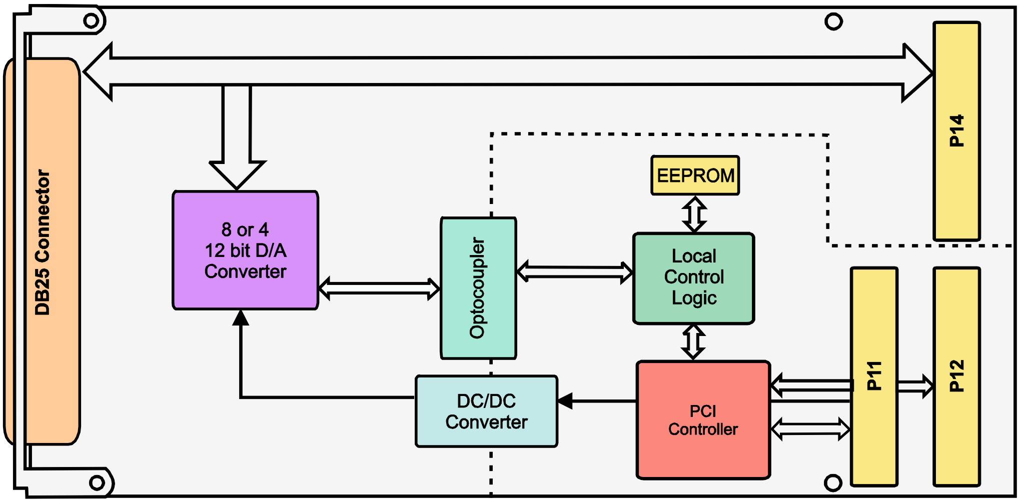


Figure 1-1 : Block Diagram

2 Technical Specification

Logic Interface	
Mechanical Interface	PCI Mezzanine Card (PMC) Interface Single Size
Electrical Interface	PCI Rev. 2.2 compliant 33 MHz / 32 bit PCI 3.3V and 5V PCI Signaling Voltage
On Board Devices	
PCI Target Chip	PCI9030 (PLX Technology)
D/A Converter	DAC7615E (Burr-Brown / Texas Instruments)
D/A Conversion	
Bit Resolution	12 bit
Output Voltage Range	0…10V or +/- 10V, selectable for a group of 4 channels
Output Load per Channel	4mA, 1000pF
Output Settling Time	10V steps: 5µs typical, 10µs maximum
I/O Interface	
Number of Channels	TPMC550-x0R: 8 channels TPMC550-x1R: 4 channels
I/O Connector	TPMC550-1xR: DB25 female connector (front panel) (Harting part# 09.66.352.661.6 or compatible) TPMC550-2xR: PMC P14 I/O (64 pin Mezzanine Connector)
Physical Data	
Power Requirements	425mA typical @ +5V DC with maximum load 110mA typical @ +3,3V DC
Temperature Range	Operating -40 °C to +85 °C Storage -40°C to +125°C
MTBF	TIP550-10R: 484000 h TIP550-11R: 530000 h TIP550-20R: 450000 h TIP550-21R: 483000 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	77 g

Table 2-1 : Technical Specification

3 Local Space Addressing

3.1 PCI9030 Local Space Configuration

The local on board addressable regions are accessed from the PCI side by addressing the PCI9030 local spaces.

PCI9030 Local Space	PCI9030 PCI Base Address (Offset in PCI Configuration Space)	PCI Space Mapping	Size (Byte)	Port Width (Bit)	Endian Mode	Description
0	2 (0x18)	I/O	32	16	BIG	Register & Sequencer RAM Space
1	3 (0x1C)	MEM	32	8	BIG	Calibration Data Space

Table 3-1 : PCI9030 Local Space Configuration

3.2 Register Space

The Register Space includes the DAC and Sequencer Registers and the Sequencer Data RAM.

PCI Base Address: PCI9030 PCI Base Address 2 (Offset 0x18 in PCI Configuration Space).

Offset to PCI Base Address	Symbol	Description	Size (Bit)	Access
0x00	DAC_CTRL	DAC Control Register	16	R/W
0x02	DAC_DATA	DAC Data Register	16	R/W
0x04	DAC_STAT	DAC Status Register	16	R
0x06	DAC_CONV	DAC Convert Register	16	R/W
0x08	SEQ_CTRL	Sequencer Control Register	16	R/W
0x0A	SEQ_STAT	Sequencer Status Register	16	R/C
0x0C	SEQ_TIME	Sequencer Timer Register	16	R/W
0x0E	-	reserved	-	-
0x10	SEQ_DATA_CH1	Sequencer Data Channel 1 Register	16	W
0x12	SEQ_DATA_CH2	Sequencer Data Channel 2 Register	16	W
0x14	SEQ_DATA_CH3	Sequencer Data Channel 3 Register	16	W
0x16	SEQ_DATA_CH4	Sequencer Data Channel 4 Register	16	W
0x18	SEQ_DATA_CH5	Sequencer Data Channel 5 Register	16	W
0x1A	SEQ_DATA_CH6	Sequencer Data Channel 6 Register	16	W
0x1C	SEQ_DATA_CH7	Sequencer Data Channel 7 Register	16	W
0x1E	SEQ_DATA_CH8	Sequencer Data Channel 8 Register	16	W

Table 3-2 : Register Space Address Map

PROM_LOCK is for factory use only. Do not write to this register.

3.2.1 DAC Control Register DAC_CTRL (Offset 0x00)

The DAC_CTRL register can be used to reset the DAC outputs.

Bit	Symbol	Description	Access	Reset Value
15 : 1		Reserved (always read as 0)	-	
0	DRST	DAC Reset Bit (used to hold all DAC outputs in a reset state) 0 = DAC outputs of all channels are in operating mode 1 = DAC outputs of all channels are forced to a reset state (0v output voltage)	R/W	0

Table 3-3 : DAC_CTRL Register (Offset 0x00)

DAC analog outputs of all channels are also held in a reset state for the time of a PCI reset or Local reset.

Setting the DRST bit in the DAC_CTRL register will set the DAC analog outputs to 0V but will not completely initialize the DAC chips.

See chapter “Functional Description” for details on DAC chip initialization.

3.2.2 DAC Data Register DAC_DATA (Offset 0x02)

The DAC_DATA register stores the 12 bit data value for the next D/A conversion. Only bits 15:4 (12 bit) are used for the D/A conversion. Bits 3:0 are ignored for the D/A conversion.

16 bit scaled data values are used to program the DAC_DATA register.

Bit	Symbol	Description	Access	Reset Value
15 : 4		12 bit digital value for D/A conversion	R/W	0x000
3 : 0		Ignored for D/A Conversion (always read as 0)	R/W	0x0

Table 3-4 : DAC_DATA Register (Offset 0x02)

3.2.2.1 DAC Data Coding

Data Value	Analog Output Voltage	
Voltage Range 0 … 10V		
0x0000	Zero	0V
0x7FF0	Midscale -1LSB	4.99756V
0x8000	Midscale	5V
0xFFF0	Full scale -1LSB	9.99756V
Voltage Range -10V … +10V		
0x8000	- Full scale	-10V
0xFFF0	Midscale -1LSB	-4.88mV
0x0000	Midscale	0V
0x7FF0	Full scale -1LSB	9.99512V

Table 3-5 : DAC Data Coding

3.2.3 DAC Status Register DAC_STAT (Offset 0x04)

The DAC_STAT register indicates DAC status conditions.

Bit	Symbol	Description	Access	Reset Value
15 : 4		Reserved (always read as 0)	-	
3	NRCH	Number of D/A channels (bit indicates the number of available D/A channels) 0 = 4 D/A channels are available (TPMC550-x1) 1 = 8 D/A channels are available (TPMC550-x0)	R	x
2	DVR2	DAC Voltage Range for D/A channels 5 to 8 (bit indicates the selected output voltage range) 0 = D/A channels 5 to 8 are set to 0…10V output voltage range 1 = D/A channels 5 to 8 are set to +/-10V output voltage range This bit is only valid if 8 D/A channels are available.	R	x
1	DVR1	DAC Voltage Range for D/A channels 1 to 4 (bit indicates the selected output voltage range) 0 = D/A channels 1 to 4 are set to 0…10V output voltage range 1 = D/A channels 1 to 4 are set to +/-10V output voltage range	R	x
0	DBSY	DAC Busy (bit indicates an active D/A conversion) 0 = No D/A conversion in progress 1 = D/A conversion in progress The DBSY bit must be read as '0' before a conversion is started by writing to the DAC_CONV register.	R	0

Table 3-6 : DAC_STAT Register (Offset 0x04)

3.2.4 DAC Convert Register DAC_CONV (Offset 0x06)

The DAC_CONV register is used to start a D/A conversion. The D/A channel and the conversion mode are setup in the same write access that starts the D/A conversion. The conversion uses the data value stored in the DAC_DATA register.

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th><th>Access</th><th>Reset Value</th></tr>
<tr><td>15 : 5</td><td></td><td>Reserved (always read as 0)</td><td>-</td><td></td></tr>
<tr><td>4</td><td>DLDC</td><td>DAC Load Command (bit is used to start a simultaneous load for all D/A channel outputs)
0 = Transparent or Latched Mode (loads DAC channel register) (load mode depends on DLDM bit)
1 = Simultaneous Load (all D/A channel analog outputs are updated according to the loaded DAC channel registers)
For DLDC = 1 (Simultaneous Load) the bits 3:0 should be written 0b0000.</td><td>R/W</td><td>0</td></tr>
<tr><td>3</td><td>DLDM</td><td>DAC Load Mode (bit selects the load mode for the D/A conversion)
0 = Transparent mode (loads selected DAC channel register and D/A channel analog output immediately)
1 = Latched mode (loads DAC channel register only) (the selected D/A channel analog output is not updated)
Use the latched mode to load the DAC internal D/A channel registers for a following simultaneous load command.</td><td>R/W</td><td>0</td></tr>
<tr><td>2

1

0</td><td>DCH2

DCH1

DCH0</td><td>DAC Channel selection for D/A conversion
<table>
<tr><th>DCH2</th><th>DCH1</th><th>DCH0</th><th>Selected DAC Channel</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>DAC Channel 1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>DAC Channel 2</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>DAC Channel 3</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>DAC Channel 4</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>DAC Channel 5</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>DAC Channel 6</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>DAC Channel 7</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>DAC Channel 8</td></tr>
</table></td><td>R/W</td><td>000</td></tr>
</table>

Table 3-7 : DAC_CONV Register (Offset 0x06)

The DBSY bit in the DAC_STAT register must be read as '0' before a conversion is started by writing to the DAC_CONV register.

3.2.5 Sequencer Control Register SEQ_CTRL (Offset 0x08)

The SEQ_CTRL register is used to setup channels and modes for the sequencer mode.

Bit	Symbol	Description	Access	Reset Value
15	CH8E	The CHxE bits are used to enable or disable DAC channel x for the next sequence. 0 = DAC channel x is disabled	R/W	0
14	CH7E			
13	CH6E			

12	CH5E	1 = DAC channel x is enabled		
11	CH4E			
10	CH3E			
9	CH2E			
8	CH1E			
7 : 4		Reserved (always read as 0)	-	
3	INTE	Sequencer Interrupt Enable 0 = Interrupts disabled 1 = Interrupts enabled The only interrupt source is the SDAT status bit in the SEQ_STAT register. Interrupt are requested at the LINT1 input of the PCI9030 Target Chip.	R/W	0
2	SLMD	Sequencer Load Mode 0 = Transparent mode (D/A analog outputs of enabled DAC channels are updated as soon as possible within the sequence) 1 = Latched mode (all D/A analog outputs of all DAC channels are updated simultaneously at the end of the sequence)	R/W	0
1	SRMD	Sequencer Runaround Mode 0 = Continuous mode (the next sequence starts immediately after the actual sequence is done) 1 = Timer mode (the next sequence starts when the sequencer timer has expired)	R(W	0
0	SEQE	Sequencer Enable 0 = Sequencer disabled (an actual sequence in progress will complete) 1 = Sequencer enabled (starts the sequencer mode)	R/W	0

Table 3-8 : SEQ_CTRL Register (Offset 0x08)

3.2.6 Sequencer Status Register SEQ_STAT (Offset 0x0A)

The SEQ_STAT register indicates sequencer conditions.

Bit	Symbol	Description	Access	Reset Value
15 : 2		Reserved (always read as 0)	-	
1	SUFL	Sequencer Data Underflow (bit is used to signalize a data underflow condition for the sequencer data RAM) 1 = Sequencer Data Underflow (sequencer is ready for the next sequence but the user has not yet confirmed new data in sequencer data RAM) The bit is cleared by writing a '1'.	R/C	0
0	SDAT	Sequencer Data Request/Acknowledge (bit is used to signalize data request for the sequencer data RAM) 1 = Sequencer Data Request (sequencer is requesting new data in the sequencer data RAM) During sequencer mode this status bit must be cleared by the user after the sequencer data RAM has been updated with data for the next sequence. The bit is cleared by writing a '1'.	R/C	0

Table 3-9 : SEQ_STAT Register (Offset 0x0A)

Bit 0 of the SEQ_STAT register can assert an interrupt request (PCI9030 LINT1) if interrupts are enabled in the SEQ_CTRL register. The bit must be cleared for interrupt acknowledge.

3.2.7 Sequencer Timer Register SEQ_TIME (Offset 0x0C)

The SEQ_TIME register is used to set up the time for the sequencer timer mode.

Bit	Symbol	Description	Access	Reset Value
15:0		Sequencer Timer Value The time step size is 100µs steps. So the time can be set to a value between 0s and 6.5535s in 100µs steps.	R/W	0x0000

Table 3-10: SEQ_TIMER Register (Offset 0x0C)

The sequencer timer is loaded with the value in the sequencer timer register when a sequence is started. In sequencer timer mode the start of the next sequence is delayed until the sequencer timer expires.

The time base for the sequencer timer is derived from the PCI system clock. A 33MHz PCI system clock is assumed.

3.2.8 Sequencer Data RAM (Offset 0x10 – 0x1E)

The sequencer data RAM is an 8 x 16 bit write only RAM space that stores the conversion data for all 8 DAC channels in sequencer mode.

Each 16 bit word stores the conversion data for a D/A channel. Data structure and data coding is the same as for the DAC_DATA register. Only the upper 12 bits (15:4) of each 16 bit RAM word are used for the D/A conversion.

Offset to PCI Base Address	Function
0x10	Conversion Data for DAC channel 1
0x12	Conversion Data for DAC channel 2
0x14	Conversion Data for DAC channel 3
0x16	Conversion Data for DAC channel 4
0x18	Conversion Data for DAC channel 5
0x1A	Conversion Data for DAC channel 6
0x1C	Conversion Data for DAC channel 7
0x1E	Conversion Data for DAC channel 8

Table 3-11: Sequencer Data RAM

3.3 Calibration Data Space

The DAC Calibration Data is mapped to the PCI9030 Local Address Space 1.

PCI Base Address: PCI9030 PCI Base Address 3 (Offset 0x1C in PCI Configuration Space).

The DAC Calibration Data Space stores the factory calibration data (offset and gain error) for each DAC channel.

See the Programming Hints chapter for data correction formulas.

The calibration data is read only.

Offset to PCI Base Address	DAC Correction Data	Value
Calibration Data for 0…+10V Output Voltage Range		
0x00	DAC Channel 1 Offset Error	Board dependent
0x01	DAC Channel 2 Offset Error	Board dependent
0x02	DAC Channel 3 Offset Error	Board dependent
0x03	DAC Channel 4 Offset Error	Board dependent
0x04	DAC Channel 5 Offset Error	Board dependent
0x05	DAC Channel 6 Offset Error	Board dependent
0x06	DAC Channel 7 Offset Error	Board dependent
0x07	DAC Channel 8 Offset Error	Board dependent
0x08	DAC Channel 1 Gain Error	Board dependent
0x09	DAC Channel 2 Gain Error	Board dependent
0x0A	DAC Channel 3 Gain Error	Board dependent
0x0B	DAC Channel 4 Gain Error	Board dependent
0x0C	DAC Channel 5 Gain Error	Board dependent
0x0D	DAC Channel 6 Gain Error	Board dependent
0x0E	DAC Channel 7 Gain Error	Board dependent
0x0F	DAC Channel 8 Gain Error	Board dependent
Calibration Data for +/-10V Output Voltage Range		
0x10	DAC Channel 1 Offset Error	Board dependent
0x11	DAC Channel 2 Offset Error	Board dependent
0x12	DAC Channel 3 Offset Error	Board dependent
0x13	DAC Channel 4 Offset Error	Board dependent
0x14	DAC Channel 5 Offset Error	Board dependent
0x15	DAC Channel 6 Offset Error	Board dependent
0x16	DAC Channel 7 Offset Error	Board dependent
0x17	DAC Channel 8 Offset Error	Board dependent
0x18	DAC Channel 1 Gain Error	Board dependent
0x19	DAC Channel 2 Gain Error	Board dependent

Offset to PCI Base Address	DAC Correction Data	Value
0x1A	DAC Channel 3 Gain Error	Board dependent
0x1B	DAC Channel 4 Gain Error	Board dependent
0x1C	DAC Channel 5 Gain Error	Board dependent
0x1D	DAC Channel 6 Gain Error	Board dependent
0x1E	DAC Channel 7 Gain Error	Board dependent
0x1F	DAC Channel 8 Gain Error	Board dependent

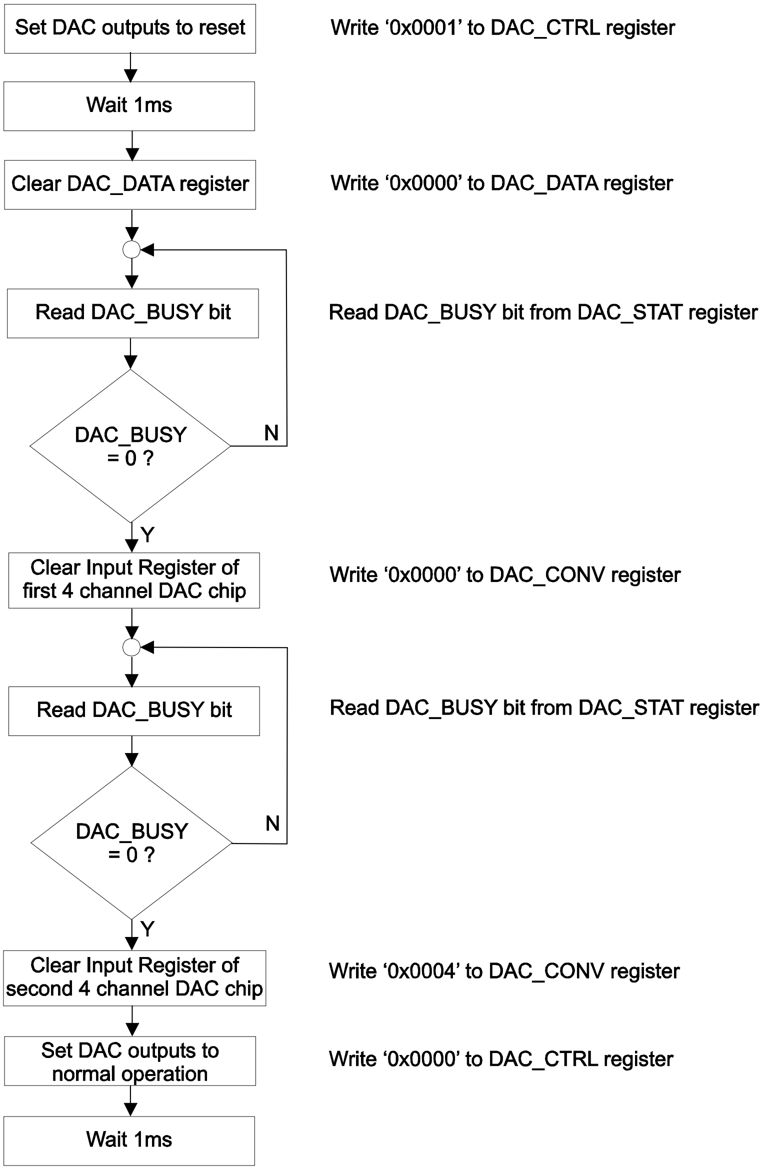
Table 3-12: Calibration Data Space Address Map

4 Functional Description

4.1 DAC Initialization

The following DAC initialization procedure must be used once after power on reset and every time DAC software reset is desired.

If this DAC initialization procedure is not used, programming one D/A output channel could result in undefined output voltages on other D/A output channels due to random initial data in the DAC chips serial input shift register.



Set DAC outputs to reset
Write '0x0001' to DAC_CTRL register
Wait 1ms
Clear DAC_DATA register
Write '0x0000' to DAC_DATA register
Read DAC_BUSY bit
Read DAC_BUSY bit from DAC_STAT register
DAC_BUSY = 0 ?
N
Y
Clear Input Register of first 4 channel DAC chip
Write '0x0000' to DAC_CONV register
Read DAC_BUSY bit
Read DAC_BUSY bit from DAC_STAT register
DAC_BUSY = 0 ?
N
Y
Clear Input Register of second 4 channel DAC chip
Write '0x0004' to DAC_CONV register
Set DAC outputs to normal operation
Write '0x0000' to DAC_CTRL register
Wait 1ms

Figure 4-1 : DAC Initialization

4.2 Programming Modes

4.2.1 Conventional Mode

In conventional mode D/A conversions are controlled by the following registers: DAC_DATA, DAC_STAT, and DAC_CONV.

The data for the next D/A conversion must be written to the DAC_DATA register.

The D/A conversion is started by writing to the DAC_CONV register.

The DAC_BUSY bit in the DAC_STAT register must be read as '0' before starting a D/A conversion.

The output channel and the mode for the D/A conversion is setup in the DAC_CONV register in the same write access that starts the D/A conversion.

New data can be written to the DAC_DATA register immediately after starting a D/A conversion by writing to the DAC_CONV register.

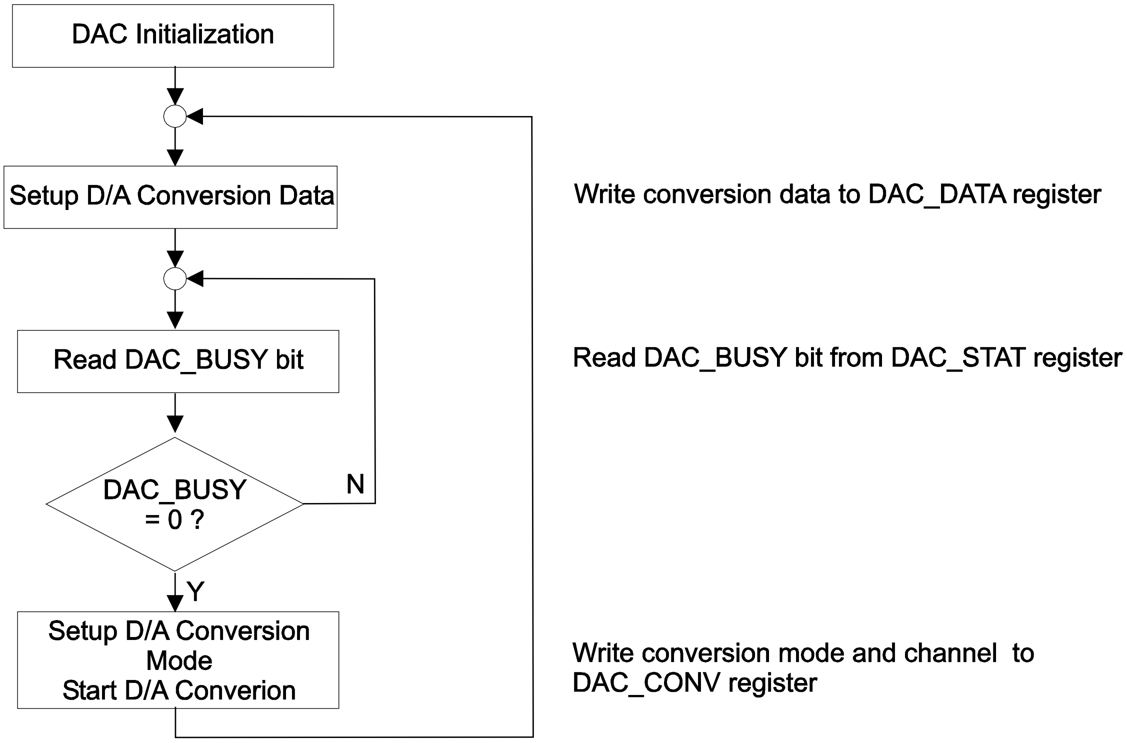

DAC Initialization
Setup D/A Conversion Data
Write conversion data to DAC_DATA register
Read DAC_BUSY bit
Read DAC_BUSY bit from DAC_STAT register
DAC_BUSY = 0 ?
N
Y
Setup D/A Conversion Mode Start D/A Converion
Write conversion mode and channel to DAC_CONV register

Figure 4-2 : Flowchart Conventional Mode

4.2.2 Sequencer Mode

In sequencer mode D/A conversions are controlled by the registers SEQ_CTRL, SEQ_STAT, SEQ_TIME and the Sequencer Data RAM.

The conversion data for the D/A channels must be written to the sequencer data RAM.

The sequencer timer must be programmed in the SEQ_TIME register.

The sequencer must be configured and enabled for the (next) sequence by programming the SEQ_CTRL register.

After that, the sequencer will start the (next) sequence.

The user must now wait until the sequencer sets the SEQ_DAT bit in the SEQ_STAT register.
If the SEQ_DAT bit is read as '1' the D/A channel conversion data for the next sequence must be written to the sequencer data RAM. Even if the conversion data should be the same for the next sequence, the data must be rewritten.
The sequencer data RAM update must be confirmed to the sequencer by clearing the SEQ_DAT bit in the SEQ_STAT register.

If the sequencer is ready to start a next runaround and a new data update has not been confirmed yet, the sequencer sets the SUFL bit in the SEQ_STAT register to inform that a data underflow has occurred. The sequencer will then start the sequence using the data and mode from the previous runaround.

Data under-runs occur if the configured sequence takes less time than the time needed for updating the sequencer data RAM and confirming the updated data.

The time duration of a sequence depends on the number of enabled D/A channels.

Sequence Time is appr. #_of_enabled_D/A_channels x 4.625µs.

The sequence time can be extended by using the sequencer timer.

The sequencer mode is not intended for updating D/A channels at a fast conversion rate. It is intended for updating D/A channels at a fix conversion rate.

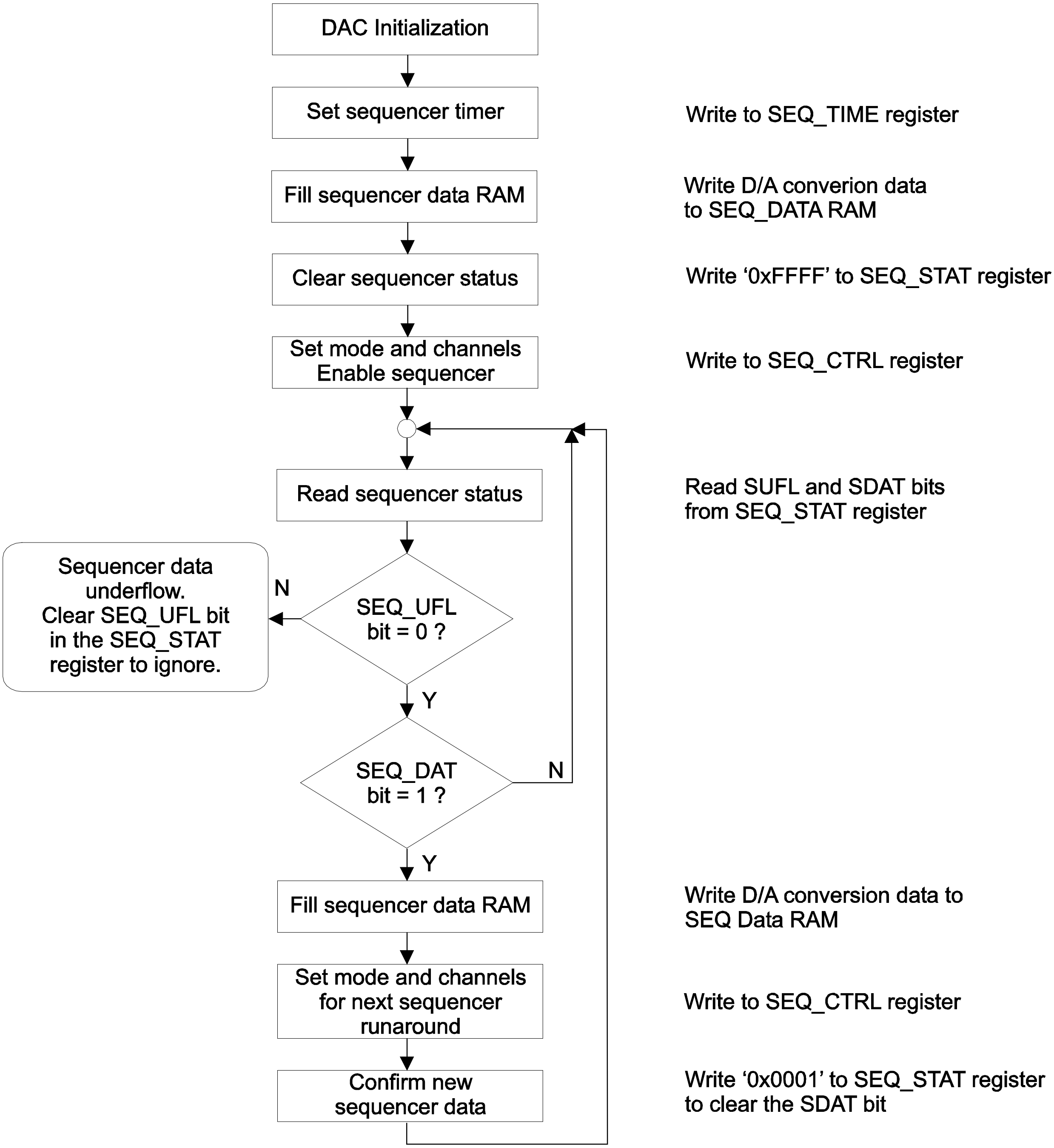

DAC Initialization
Set sequencer timer
Write to SEQ_TIME register
Fill sequencer data RAM
Write D/A converion data to SEQ_DATA RAM
Clear sequencer status
Write ‘0xFFFF’ to SEQ_STAT register
Set mode and channels Enable sequencer
Write to SEQ_CTRL register
Read sequencer status
Read SUFL and SDAT bits from SEQ_STAT register
Sequencer data underflow. Clear SEQ_UFL bit in the SEQ_STAT register to ignore.
N
SEQ_UFL bit = 0 ?
Y
SEQ_DAT bit = 1 ?
N
Y
Fill sequencer data RAM
Write D/A conversion data to SEQ Data RAM
Set mode and channels for next sequencer runaround
Write to SEQ_CTRL register
Confirm new sequencer data
Write ‘0x0001’ to SEQ_STAT register to clear the SDAT bit

Figure 4-3 : Flowchart Sequencer Mode

4.3 Load Modes

The DAC load mode applies to both programming modes (conventional mode and sequencer mode).

In conventional mode the DAC load mode is selected in the DAC_CONV register.

In sequencer mode the DAC load mode is selected in the SEQ_CTRL register.

4.3.1 Transparent Mode

In transparent mode the D/A conversion loads the DAC internal channel register and immediately updates the analog D/A output.

4.3.2 Latched Mode

In latched mode the D/A conversion loads only the DAC internal channel register without updating the analog D/A output. For conventional mode a separate simultaneous load command updates the analog D/A outputs all at once according to the loaded values in the DAC internal channel registers. In sequencer mode this simultaneous load command is generated automatically by the sequencer.

4.4 Runaround Modes

The runaround mode applies to the sequencer mode only. The runaround mode is selected in the SEQ_CTRL register.

4.4.1 Continuous Mode

In continuous mode the next sequence is immediately started after a sequence is done.

If the DAC load mode is set to transparent mode an analog D/A output is updated immediately when the DAC internal channel register is loaded within the sequence.

If the DAC load mode is set to latched mode all analog D/A outputs are updated simultaneously at the end of the sequence (immediately after all DAC internal channel registers are loaded for the sequence). The time between two analog D/A output updates depends on the number of enabled D/A channels.

4.4.2 Timer Mode

In timer mode, at the end of a sequence the sequencer waits until the sequencer timer expires before the next sequence is started (and new data in the sequencer RAM is expected to be confirmed). The sequencer timer value is set in the SEQ_TIME register. The time value can be set in steps of 100µs. A time value of 0 sets the sequencer to a continuous mode.

If the DAC load mode is set to transparent mode an analog D/A output is updated immediately when the DAC internal channel register is loaded within the sequence. At the end of the sequence the sequencer waits for the timer to expire before the next sequence is started.

If the DAC load mode is set to latched mode all analog D/A outputs are updated simultaneously when the timer expires (if time value > sequence time) or immediately after the DAC internal channel registers are loaded for the sequence (if time value < sequence time).

The time between two D/A channel analog output updates is independent from the number of enabled DAC channels (if time value > sequence time) or depends on the number of enabled D/A channels (if time value < sequence time).

5 PCI9030 Target Chip

5.1 PCI Configuration Registers (PCR)

5.1.1 PCI9030 Header

PCI CFG Register Address	Write ‘0’ to all unused (Reserved) bits				PCI writeable	Initial Values (Hex Values)
	31 24	23 16	15 8	7 0		
0x00	Device ID		Vendor ID		N	9050 10B5
0x04	Status		Command		Y	0280 0000
0x08	Class Code			Revision ID	N	118000 00
0x0C	BIST	Header Type	PCI Latency Timer	Cache Line Size	Y[7:0]	00 00 00 00
0x10	PCI Base Address 0 for MEM Mapped Config. Registers				Y	FFFFFF80
0x14	PCI Base Address 1 for I/O Mapped Config. Registers				Y	FFFFFF81
0x18	PCI Base Address 2 for Local Address Space 0				Y	FFFFFFE1
0x1C	PCI Base Address 3 for Local Address Space 1				Y	FFFFFFE0
0x20	PCI Base Address 4 for Local Address Space 2				Y	00000000
0x24	PCI Base Address 5 for Local Address Space 3				Y	00000000
0x28	PCI CardBus Information Structure Pointer				N	00000000
0x2C	Subsystem ID		Subsystem Vendor ID		N	0226 1498
0x30	PCI Base Address for Local Expansion ROM				Y	00000000
0x34	Reserved			New Cap. Ptr.	N	000000 40
0x38	Reserved				N	00000000
0x3C	Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	Y[7:0]	00 00 01 00
0x40	PM Cap.		PM Nxt Cap.	PM Cap. ID	N	4801 48 01
0x44	PM Data	PM CSR EXT	PM CSR		Y	00 00 0000
0x48	Reserved	HS CSR	HS Nxt Cap.	HS Cap. ID	Y[23:16]	00 00 4C 06
0x4C	VPD Address		VPD Nxt Cap.	VPD Cap. ID	Y[31:16]	0000 00 03
0x50	VPD Data				Y	00000000

Table 5-1 : PCI9030 Header

5.1.2 PCI Base Address Initialization

PCI Base Address Initialization is scope of the PCI host software.

PCI9030 PCI Base Address Initialization:

1. Write 0xFFFF_FFFF to the PCI9030 PCI Base Address Register.
2. Read back the PCI9030 PCI Base Address Register.
3. For PCI Base Address Registers 0:5, check bit 0 for PCI Address Space.

   Bit 0 = '0' requires PCI Memory Space mapping

   Bit 0 = '1' requires PCI I/O Space mapping

   For the PCI Expansion ROM Base Address Register, check bit 0 for usage.

   Bit 0 = ‘0’: Expansion ROM not used

   Bit 0 = ‘1’: Expansion ROM used
4. For PCI I/O Space mapping, starting at bit location 2, the first bit set determines the size of the required PCI I/O Space size.

   For PCI Memory Space mapping, starting at bit location 4, the first bit set to '1' determines the size of the required PCI Memory Space size.

   For PCI Expansion ROM mapping, starting at bit location 11, the first bit set to '1' determines the required PCI Expansion ROM size.

   For example, if bit 5 of a PCI Base Address Register is detected as the first bit set to ‘1’, the PCI9030 is requesting a 32 byte space (address bits 4:0 are not part of base address decoding).
5. Determine the base address and write the base address to the PCI9030 PCI Base Address Register. For PCI Memory Space mapping the mapped address region must comply with the definition of bits 3:1 of the PCI9030 PCI Base Address Register.

After programming the PCI9030 PCI Base Address Registers, the software must enable the PCI9030 for PCI I/O and/or PCI Memory Space access in the PCI9030 PCI Command Register (Offset 0x04). To enable PCI I/O Space access to the PCI9030, set bit 0 to '1'. To enable PCI Memory Space access to the PCI9030, set bit 1 to '1'.

Offset in Config.	Description	Usage
0x10	PCI9030 LCR’s MEM	Used
0x14	PCI9030 LCR’s I/O	Used
0x18	PCI9030 Local Space 0	Used
0x1C	PCI9030 Local Space 1	Used
0x30	Expansion ROM	Not used

Table 5-2 : PCI9030 PCI Base Address Usage

5.2 Local Configuration Registers

After reset, the PCI9030 Local Configuration Registers are loaded from the on board serial configuration EEPROM.

The PCI base address for the PCI9030 Local Configuration Registers is PCI9030 PCI Base Address 0 (PCI Memory Space) (Offset 0x10 in the PCI9030 PCI Configuration Register Space) or PCI9030 PCI Base Address 1 (PCI I/O Space) (Offset 0x14 in the PCI9030 PCI Configuration Register Space).

Do not change hardware dependent bit settings in the PCI9030 Local Configuration Registers.

Offset from PCI Base Address	Register	Value	Description
0x00	Local Address Space 0 Range	0x0FFF_FFE1	
0x04	Local Address Space 1 Range	0x0FFF_FFE0	
0x08	Local Address Space 2 Range	0x0000_0000	
0x0C	Local Address Space 3 Range	0x0000_0000	
0x10	Local Exp. ROM Range	0x0000_0000	
0x14	Local Re-map Register Space 0	0x0000_0001	
0x18	Local Re-map Register Space 1	0x0000_0101	
0x1C	Local Re-map Register Space 2	0x0000_0000	
0x20	Local Re-map Register Space 3	0x0000_0000	
0x24	Local Re-map Register ROM	0x0000_0000	
0x28	Local Address Space 0 Descriptor	0x5141_2080	
0x2C	Local Address Space 1 Descriptor	0x5505_6280	
0x30	Local Address Space 2 Descriptor	0x0000_0000	
0x34	Local Address Space 3 Descriptor	0x0000_0000	
0x38	Local Exp. ROM Descriptor	0x0000_0000	
0x3C	Chip Select 0 Base Address	0x0000_0009	
0x40	Chip Select 1 Base Address	0x0000_0019	
0x44	Chip Select 2 Base Address	0x0000_0111	
0x48	Chip Select 3 Base Address	0x0000_0000	
0x4C	Interrupt Control/Status	0x0041	
0x4E	EEPROM Write Protect Boundary	0x0030	
0x50	Miscellaneous Control Register	0x0078_0000	
0x54	General Purpose I/O Control	0x0000_0240	
0x70	Hidden1 Power Management data select	0x0000_0000	
0x74	Hidden 2 Power Management data scale	0x0000_0000	

Table 5-3 : PCI9030 Local Configuration Register

5.3 Configuration EEPROM

After power-on or PCI reset, the PCI9030 loads initial configuration register data from the on board configuration EEPROM.

The configuration EEPROM contains the following configuration data:

- Address 0x00 to 0x27 : PCI9030 PCI Configuration Register Values
- Address 0x28 to 0x87 : PCI9030 Local Configuration Register Values
- Address 0x88 to 0xFF : Reserved

See the PCI9030 Manual for more information.

Address	**Offset**							
	0x00	0x02	0x04	0x06	0x08	0x0A	0x0C	0x0E
0x00	0x9050	0x10B5	0x0280	0x0000	0x1180	0x0000	0x0226	0x1498
0x10	0x0000	0x0040	0x0000	0x0100	0x4801	0x4801	0x0000	0x0000
0x20	0x0000	0x4C06	0x0000	0x0003	0x0FFF	0xFFE1	0x0FFF	0xFFE0
0x30	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0001
0x40	0x0000	0x0101	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
0x50	0x5141	0x2080	0x5505	0x6280	0x0000	0x0000	0x0000	0x0000
0x60	0x0000	0x0000	0x0000	0x0009	0x0000	0x0019	0x0000	0x0111
0x70	0x0000	0x0000	0x0030	0x0041	0x0078	0x0000	0x0000	0x0240
0x80	0x0000	0x0000	0x0000	0x0000	0xFFFF	0xFFFF	0xFFFF	0xFFFF
0x90	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
0xA0	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
0xB0	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
0xC0	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
0xD0	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
0xE0	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
0xF0	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF

Table 5-4 : Configuration EEPROM TPMC550-xx

5.4 Software Reset

The PCI9030 Local Reset Output LRESETo# is used to reset the on board local logic.

The PCI9030 local reset is active during PCI reset or if the PCI Adapter Software Reset bit is set in the PCI9030 local configuration register CNTRL (offset 0x50).

CNTRL[30] PCI Adapter Software Reset:

Value of ‘1’ resets the PCI9030 and issues a reset to the Local Bus (LRESETo# asserted). The PCI9030 remains in this reset condition until the PCI Host clears this bit. The contents of the PCI9030 PCI and Local Configuration Registers are not reset. The PCI9030 PCI Interface is not reset.

6 Configuration Hints

6.1 Big / Little Endian

- PCI – Bus (Little Endian)

Byte 0	AD[7..0]
Byte 1	AD[15..8]
Byte 2	AD[23..16]
Byte 3	AD[31..24]

- Every Local Address Space (0...3) and the Expansion ROM Space can be programmed to operate in Big or Little Endian Mode.

Big Endian		**Little Endian**	
32 Bit		**32 Bit**	
Byte 0	D[31..24]	Byte 0	D[7..0]
Byte 1	D[23..16]	Byte 1	D[15..8]
Byte 2	D[15..8]	Byte 2	D[23..16]
Byte 3	D[7..0]	Byte 3	D[31..24]
16 Bit upper lane		**16 Bit**	
Byte 0	D[31..24]	Byte 0	D[7..0]
Byte 1	D[23..16]	Byte 1	D[15..8]
16 Bit lower lane			
Byte 0	D[15..8]		
Byte 1	D[7..0]		
8 Bit upper lane		**8 Bit**	
Byte 0	D[31..24]	Byte 0	D[7..0]
8 Bit lower lane			
Byte 0	D[7..0]		

Table 6-1 : Local Bus Little/Big Endian

Standard use of the TPMC550:

Local Address Space 0	16 bit bus in Big Endian Mode
Local Address Space 1	8 bit bus in Big Endian Mode
Local Address Space 2	not used
Local Address Space 3	not used
Expansion ROM Space	not used

To change the Endian Mode use the Local Configuration Registers for the corresponding Space. Bit 24 of the according register sets the mode. A value of 1 indicates Big Endian and a value of 0 indicates Little Endian.

Use the PCI Base Address 0 + Offset or PCI Base Address 1 + Offset:

Short cut	Offset	Name
LAS0BRD	0x28	Local Address Space 0 Bus Region Description Register
LAS1BRD	0x2C	Local Address Space 0 Bus Region Description Register
LAS2BRD	0x30	Local Address Space 0 Bus Region Description Register
LAS3BRD	0x34	Local Address Space 0 Bus Region Description Register
EROMBRD	0x38	Expansion ROM Bus Region Description Register

You could also use the PCI - Base Address 1 I/O Mapped Configuration Registers.

7 Programming Hints

7.1 DAC Initialization

See the chapter “Functional Description”.

7.2 DAC Data Correction

These are two errors which affect the DC accuracy of the DAC.

The first error is the zero error (offset). For the DAC this is the data value that is required to produce a zero voltage output signal. This error is corrected by subtracting the offset from the DAC data value.

The second error is the gain error. The gain error is the difference between the ideal gain and the actual gain of the DAC. It is corrected by multiplying the DAC data value with a correction factor.

The correction values are obtained during factory calibration and are stored in an on board EEPROM as 2-complement byte wide values in the range from -128 to +127. For higher accuracy they are scaled to ¼ LSB.

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

7.2.1 DAC Value Correction for 0V…10V Output Voltage Range

The basic formula for correcting the DAC output value in unipolar mode is:

$$\textbf{Data} = \textbf{Value} * (1 - \textbf{GAIN}_{\textbf{corr}} / 16384) - \textbf{OFFSET}_{\textbf{corr}} * 4$$

Data is the corrected digital value that should be programmed to the data register. *Value* is the ideal digital value for the desired output voltage. *GAIN_{corr}* and *OFFSET_{corr}* are the correction factors from the on board EEPROM. *GAIN_{corr}* and *OFFSET_{corr}* are stored separately for each of the possible D/A channels.

7.2.2 DAC Value Correction for +/-10V Output Voltage Range

The basic formula for correcting DAC output value in bipolar mode is:

$$\textbf{Data} = \textbf{Value} * (1 - \textbf{GAIN}_{\textbf{corr}} / 8192) - \textbf{OFFSET}_{\textbf{corr}} * 4$$

Data is the corrected digital value that should be programmed to the data register. *Value* is the ideal digital value for the desired output voltage. *GAIN_{corr}* and *OFFSET_{corr}* are the correction factors from the on board EEPROM. *GAIN_{corr}* and *OFFSET_{corr}* are stored separately for each of the possible D/A channels.

The *GAIN_{corr}* and *OFFSET_{corr}* values can be read in the Calibration Data Space.

8 Installation

8.1 Jumper Configuration

Jumper	Function	Configuration	Option
J2	Output Voltage Range D/A channels 1 to 4	1-2 closed	0…10V
		2-3 closed	+/-10V
J1 (TPMC550-x0 only)	Output Voltage Range D/A channels 5 to 8	1-2 closed	0…10V
		2-3 closed	+/-10V

Figure 8-1 : Jumper Configuration

8.1.1 Jumper Location

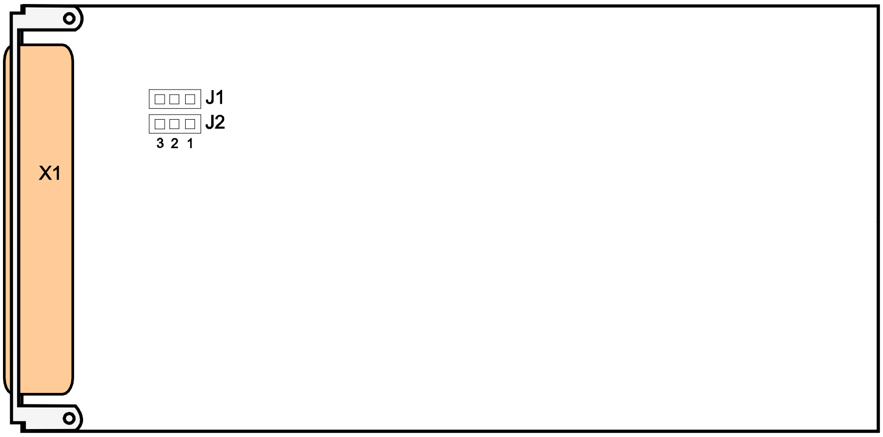


Figure 8-2 : Jumper Location

9 I/O Connector Pin Assignment

9.1 Front Panel I/O Connector (DB25 female)

The front panel I/O connector applies for TPMC550-1xR board options only.

Pin	Signal	Function
1	DAC_OUT1	D/A output channel 1
2	DAC_OUT2	D/A output channel 2
3	DAC_OUT3	D/A output channel 3
4	DAC_OUT4	D/A output channel 4
5	DAC_OUT5	D/A output channel 5 (TPMC550-10R only)
6	DAC_OUT6	D/A output channel 6 (TPMC550-10R only)
7	DAC_OUT7	D/A output channel 7 (TPMC550-10R only)
8	DAC_OUT8	D/A output channel 8 (TPMC550-10R only)
9		
10		
11		
12		
13		
14	AGND	Signal ground for D/A output channels
15	AGND	Signal ground for D/A output channels
16	AGND	Signal ground for D/A output channels
17	AGND	Signal ground for D/A output channels
18	AGND	Signal ground for D/A output channels
19	AGND	Signal ground for D/A output channels
20	AGND	Signal ground for D/A output channels
21	AGND	Signal ground for D/A output channels
22		
23		
24		
25		

Table 9-1 : Front Panel I/O Connector (DB25 female)

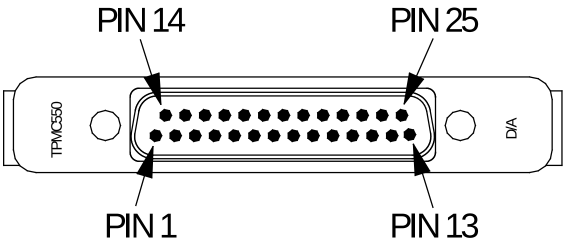


Figure 9-1 : Assembly Drawing

9.2 Mezzanine Card P14 Connector

The P14 mezzanine back I/O connector applies for TPMC550-2xR board options only.

Pin	Signal	Level
1	DAC_OUT1	D/A output channel 1
2	AGND	Signal ground for D/A output channels
3	DAC_OUT2	D/A output channel 2
4	AGND	Signal ground for D/A output channels
5	DAC_OUT3	D/A output channel 3
6	AGND	Signal ground for D/A output channels
7	DAC_OUT4	D/A output channel 4
8	AGND	Signal ground for D/A output channels
9	DAC_OUT5	D/A output channel 5 (TPMC550-20R only)
10	AGND	Signal ground for D/A output channels
11	DAC_OUT6	D/A output channel 6 (TPMC550-20R only)
12	AGND	Signal ground for D/A output channels
13	DAC_OUT7	D/A output channel 7 (TPMC550-20R only)
14	AGND	Signal ground for D/A output channels
15	DAC_OUT8	D/A output channel 8 (TPMC550-20R only)
16	AGND	Signal ground for D/A output channels
17		
…		
64		

Table 9-2 : Mezzanine Card P14 Connector