Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

TMC4331A DATASHEET

TMC4331A Document Revision 1.01 • 2016-NOV-25

The S-ramp and sixPoint™ ramp motion controller for stepper motors is optimized for high velocities, allowing on-the-fly changes. TMC4331A offers SPI and Step/Dir interfaces.

Figure 1: Sample Image TMC4331A

**Marking details are explained on page 172.*

Features

- SPI Interfaces for µC with easy-to-use protocol.
- SPI Interfaces for SPI motor stepper drivers.
- Integrated ChopSync™ and dcStep™ support.
- Internal ramp generator generating S-shaped ramps or sixPoint™ ramps supporting on-the-fly changes.
- Controlled PWM output.
- Reference switch handling.
- Hardware and virtual stop switches.
- Extensive Support of TMC stepper motor drivers.
- Electronic gearing support.

Applications

- Textile, sewing machines
- CCTV, security
- Printers, scanners
- ATM, cash recycler
- Office automation
- POS
- Factory automation
- Lab automation
- Pumps and valves
- Heliostat controllers
- CNC machines
- Robotics

Block Diagram: TMC4331A Interfaces & Features

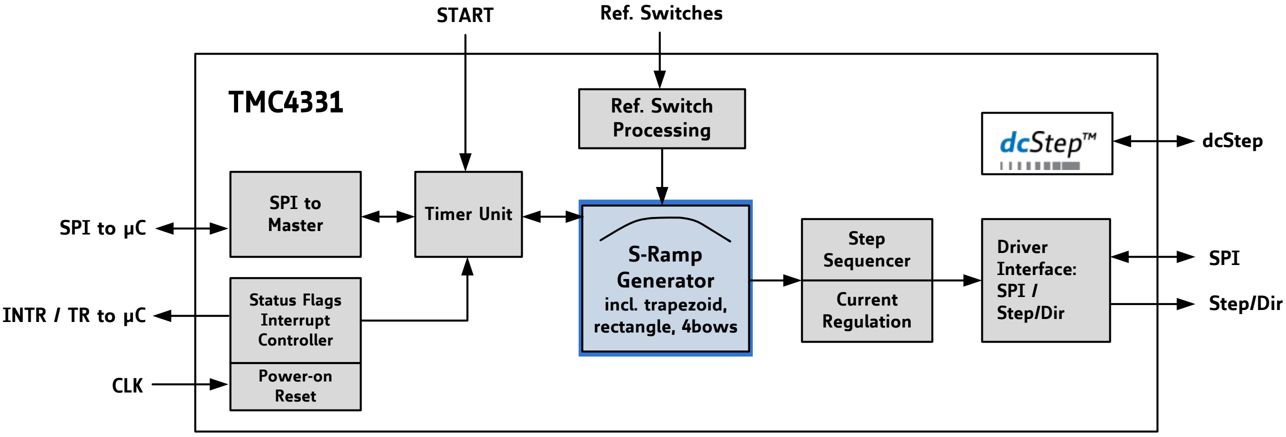

Figure 2: Block Diagram

Read entire documentation; especially the Supplemental Directives in chapter 18 (page 173).

Functional Scope of TMC4331A

TMC4331A is a miniaturized high-performance motion controller for stepper motor drivers, particularly designed for fast and jerk-limited motion profile applications with a wide range of ramp profiles. The S-shaped or sixPoint™ velocity profile, and open-loop features offer many configuration options to suit the user's specifications, as presented below:

S-Shaped Velocity Profile

S-shaped ramp profiles are jerk-free. Seven ramp segments form the S-shaped ramp that can be optimally adapted to suit the user's requirements. High torque with high velocities can be reached by calibrating the bows of the ramp, as explained in this user manual.

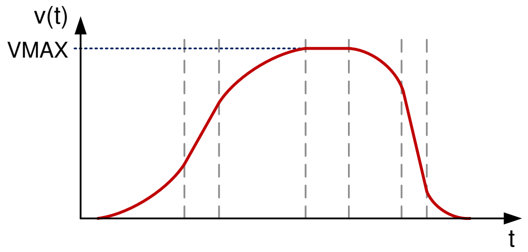

Figure 3: S-shaped Velocity Profile

i More information on ramp configurations and other velocity profiles, e.g. sixPoint™ ramps, are provided in chapter 6 (Page 24).

Reference Switch Support

A typical hardware setup for open-loop operation with enhanced modifications, by use of external stop switches with the TMC26x motor stepper driver is shown below. Home switches with different configurations are also supported.

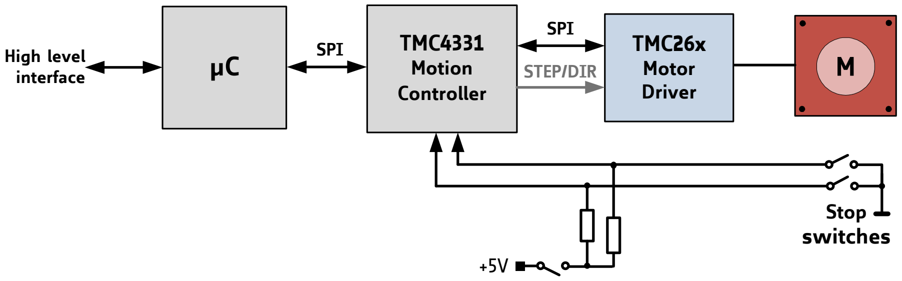

Figure 4: Open-Loop Hardware Set-up with TMC26x supporting External Stop Switches

Open-loop Operation with dcStep™ Feature

A typical hardware setup for dcStep operation with a TMC2130 stepper motor driver is shown in the diagram below. This feature is also available for TMC26x stepper motor drivers.

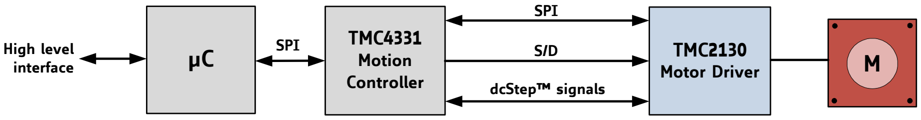

Figure 5: Hardware Set-up for Open-loop Operation with TMC2130

Order Codes

Order code	Description	Size
TMC4331A-LA	*Motion controller with dcStep features, QFN32*	*4 x 4 mm^2*

Table 1: TMC4331A Order Codes

TABLE OF CONTENTS

Read entire documentation; especially the "Supplemental Directives" on page 173.

MAIN MANUAL

1. Pinning and Design-In Process Information

In this chapter you are provided with a list of all pin names and a functional description of each.

1.1. Pin Assignment: Top View

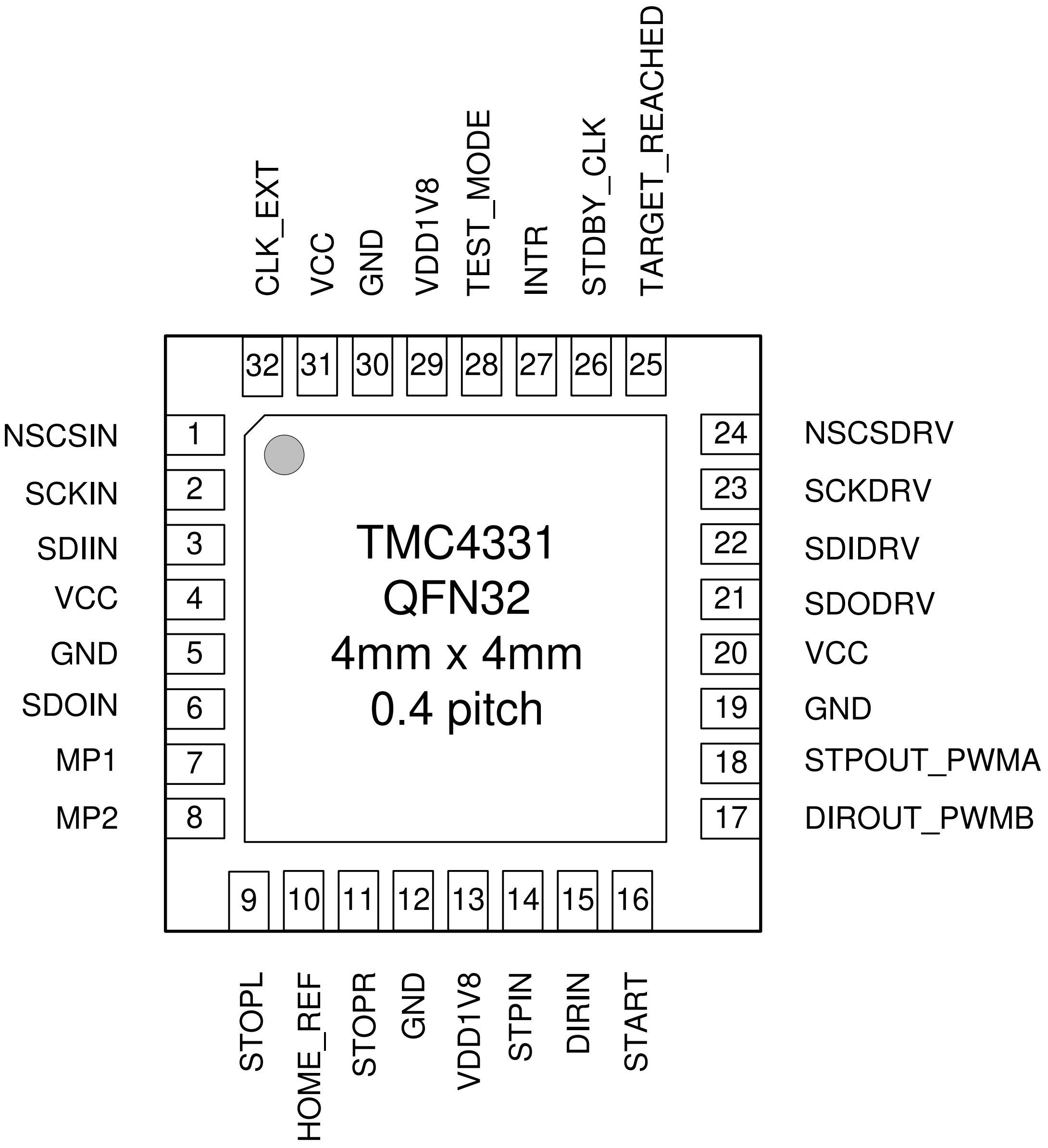

Figure 6: Package Outline: Pin Assignments Top View

1.2. Pin Description

Pin Names and Descriptions

Pin	Number	Type	Function
Supply Pins			
GND	5, 12, 19, 30	GND	Digital ground pin for IOs and digital circuitry.
VCC	4, 20, 31	VCC	Digital power supply for IOs and digital circuitry (3.3V… 5V).
VDD1V8	13, 29	VDD	Connection of internal generated core voltage of 1.8V.
CLK_EXT	32	I	Clock input to provide a clock with the frequency fCLK for all internal operations.
TEST_MODE	28	I	Test mode input. Tie to low for normal operation.
Interface Pins for µC			
NSCSIN	1	I	Low active chip selects input of SPI interface to µC.
SCKIN	2	I	Serial clock for SPI interface to µC.
SDIIN	3	I	Serial data input of SPI interface to µC.
SDOIN	6	O	Serial data output of SPI interface to µC (Z if NSCSIN=1).
INTR	27	O	Interrupt output, programmable PD/PU for wired-and/or.
TARGET_REACHED	25	O	Target reached output, programmable PD/PU for wired-and/or.
Reference Pins			
STOPL	9	I (PD)	Left stop switch. External signal to stop a ramp. If not connected, internal pull-down resistor is active.
HOME_REF	10	I (PD)	Home reference signal input. External signal for reference search. If not connected, internal pull-down resistor is active.
STOPR	11	I (PD)	Right stop switch. External signal to stop a ramp. If not connected, internal pull-down resistor is active.
STPIN	14	I (PD)	Step input for external step control. If not connected, internal pull-down resistor is active.
DIRIN	15	I (PD)	Direction input for external step control. If not connected, internal pull-down resistor is active.
START	16	IO	Start signal input/output.
S/D Output Pins			
STPOUT PWMA DACA	18	O	Step output. First PWM signal (Sine). First DAC output signal (Sine).
DIROUT PWMB DACB	17	O	Direction output. Second PWM signal (Cosine). Second DAC output signal (Cosine).

•→ *Continued on next page!*

Pin Names and Descriptions			
Pin	**Number**	**Type**	**Function**
Interface Pins for Stepper Motor Drivers			
NSCSDRV PWMB	24	O	Low active chip selects output of SPI interface to motor driver. Second PWM signal (Cosine) to connect with PHB (TMC23x/24x).
SCKDRV MDBN	23	O	Serial clock output of SPI interface to motor driver. MDBN output signal for MDBN pin of TMC23x/24x.
SDODRV PWMA	21	O	Serial data output of SPI interface to motor driver. First PWM signal (Sine) to connect with PHA (TMC23x/24x).
SDIDRV ERR	22	I (PD)	Serial data input of SPI interface to motor driver. Error input signal to ERR pin of TMC23x/24x. If not connected, internal pull-down resistor is active.
MP1	7	I (PD)	DC_IN as external dcStep input control signal. If not connected, internal pull-down resistor is active.
MP2	8	IO	DCSTEP_ENABLE as dcStep output control signal. SPE_OUT as output signal, connect to SPE pin of TMC23x/24x.
STDBY_CLK	26	O	StandBy signal or internal CLK output or ChopSync output.

Table 2: Pin Names and Descriptions

1.3. System Overview

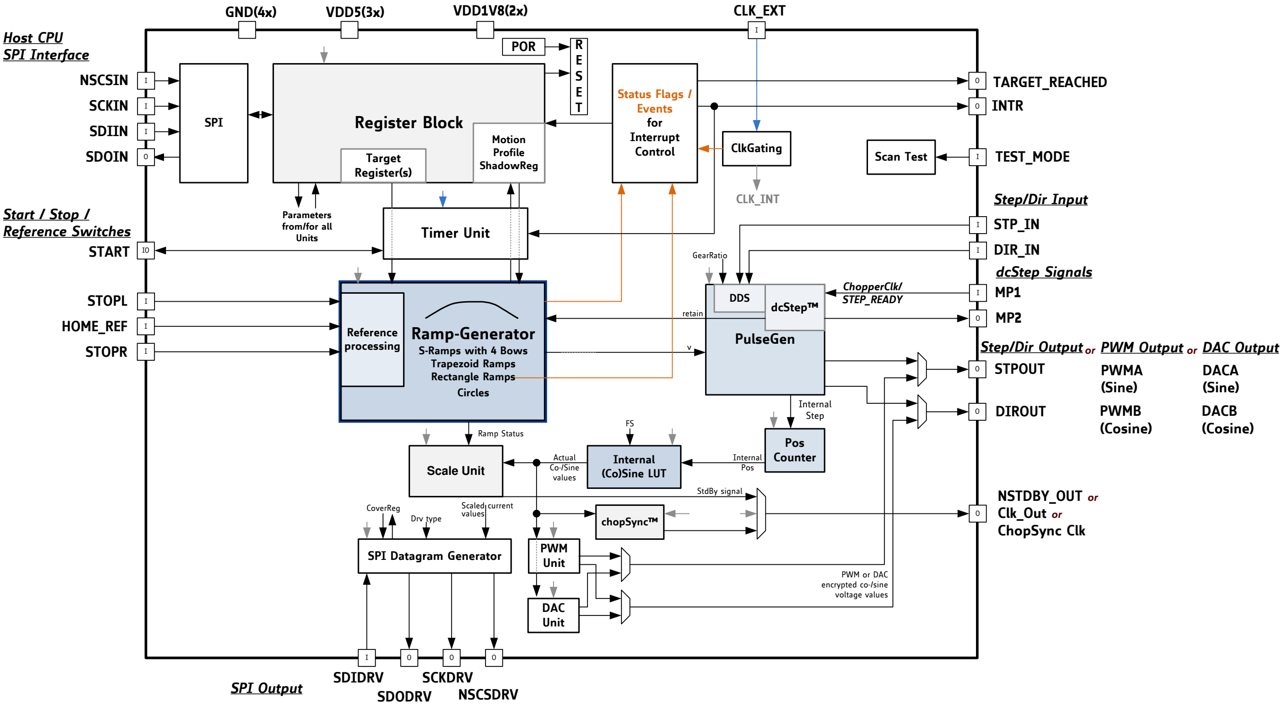

Figure 7: System Overview

Read entire documentation; especially the "Supplemental Directives" on page 173.

2. Application Circuits

In this chapter application circuit examples are provided that show how external components can be connected.

2.1. TMC4331A Standard Connection: VCC=3.3V

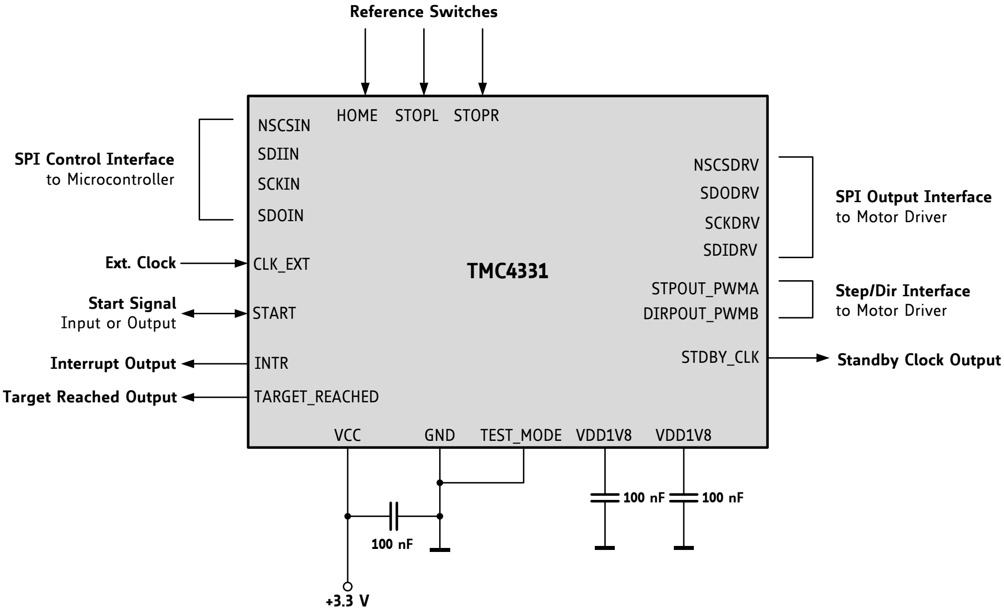

Figure 8: TMC4331A Connection: VCC=3.3V

2.2. TMC4331A with TMC26x Stepper Connection

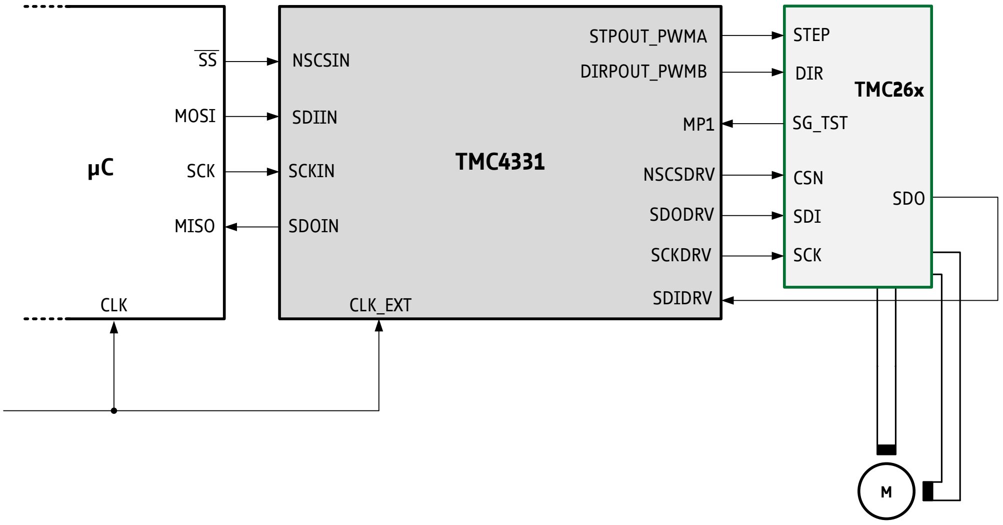

Figure 9: TMC4331A with TMC26x Stepper Driver in SPI Mode or S/D Mode

2.3. TMC4331A with TMC248 Stepper Driver

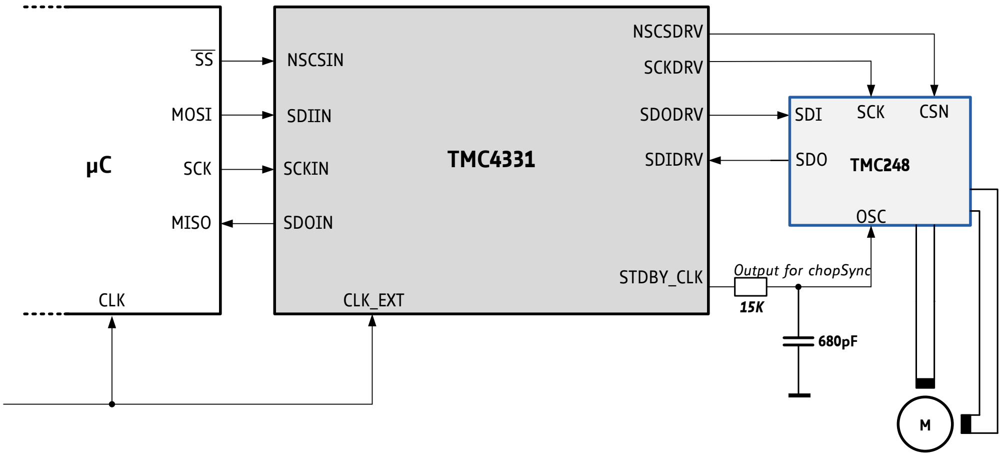

Figure 10: TMC4331A with TMC248 Stepper Driver in SPI Mode

2.4. TMC4331A with TMC2130 Stepper Driver

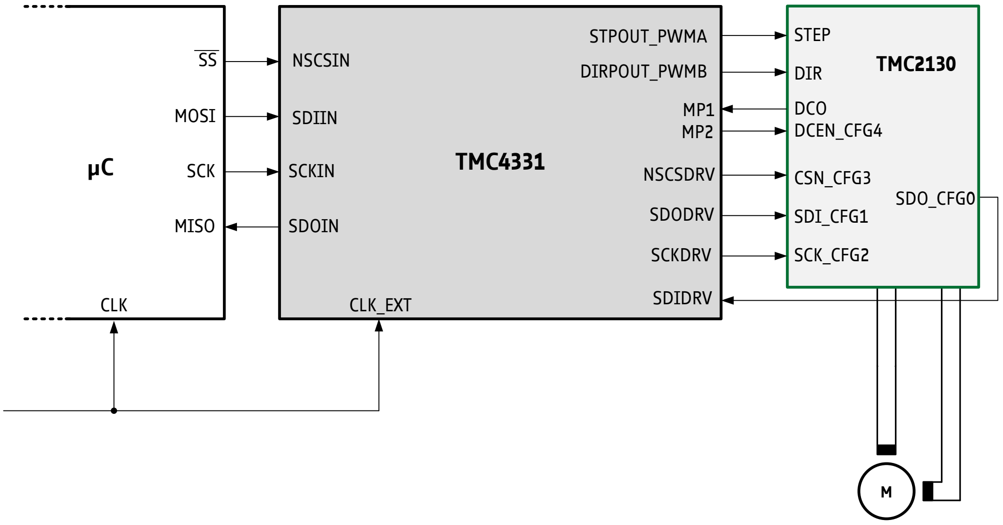

Figure 11: TMC4331A with TMC2130 Stepper Driver in SPI Mode or S/D Mode

3. SPI Interfacing

TMC4331A uses 40-bit SPI datagrams for communication with a microcontroller. The bit-serial interface is synchronous to a bus clock. For every bit sent from the bus master to the bus slave, another bit is sent simultaneously from the slave to the master. In the following chapter information is provided about the SPI control interface, SPI datagram structure and SPI transaction process.

SPI Input Control Interface Pins		
Pin Name	**Type**	**Remarks**
NSCSIN	Input	Chip Select of SPI-µC interface (low active)
SCKIN	Input	Serial clock of SPI-µC interface
SDIIN	Input	Serial data input of SPI-µC interface
SDOIN	Output	Serial data output of SPI-µC interface

Table 3: SPI Input Control Interface Pins

3.1. SPI Datagram Structure

- Microcontrollers that are equipped with hardware SPI are typically able to communicate using integer multiples of 8 bit.
- The NSCSIN line of the TMC4331A has to stay active (low) for the complete duration of the datagram transmission.
- Each datagram that is sent to TMC4331A is composed of an address byte followed by four data bytes. This allows direct 32-bit data word communication with the register set of TMC4331A. Each register is accessed via 32 data bits; even if it uses less than 32 data bits.

i Each register is specified by a one-byte address:

- *For read access the most significant bit of the address byte is 0.*
- *For write access the most significant bit of the address byte is 1.*

NOTE:

→ *Some registers are write only registers. Most registers can be read also; and there are also some read only registers.*

<table>
<tr><th colspan="40">TMC4331A SPI Datagram Structure</th></tr>
<tr><td colspan="13"><b>MSB (transmitted first)</b></td><td colspan="14"><b>40 bits</b></td><td colspan="13"><b>LSB (transmitted last)</b></td></tr>
<tr><td colspan="13"><b>39</b></td><td colspan="14"><b>...</b></td><td colspan="13"><b>0</b></td></tr>
<tr><td colspan="8">→ 8-bit address
← 8-bit SPI status</td><td colspan="32">← → 32-bit data</td></tr>
<tr><td colspan="8">39 ... 32</td><td colspan="32">31 ... 0</td></tr>
<tr><td colspan="8">→ to TMC4331:
RW + 7-bit address
← from TMC4331:
8-bit SPI status</td><td colspan="8">8-bit data</td><td colspan="8">8-bit data</td><td colspan="8">8-bit data</td><td colspan="8">8-bit data</td></tr>
<tr><td colspan="8">39 / 38 ... 32</td><td colspan="8">31 ... 24</td><td colspan="8">23 ... 16</td><td colspan="8">15 ... 8</td><td colspan="8">7 ... 0</td></tr>
<tr><td>W</td><td colspan="7">38...32</td><td colspan="4">31...28</td><td colspan="4">27...24</td><td colspan="4">23...20</td><td colspan="4">19...16</td><td colspan="4">15...12</td><td colspan="4">11...8</td><td colspan="4">7...4</td><td colspan="4">3...0</td></tr>
<tr><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

Figure 12: TMC4331A SPI Datagram Structure

Read/Write Selection Principles and Process

Read and write selection is controlled by the MSB of the address byte (bit 39 of the SPI datagram). This bit is 0 for read access and 1 for write access. Consequently, the bit named W is a WRITE_notREAD control bit.

The active high write bit is the MSB of the address byte.
Consequently, 0x80 must be added to the address for a write access.

The SPI interface always delivers data back to the master, independent of the Write bit W.

Difference between Read and Write Access	
If …	**Then …**
The previous access was a read access.	The data transferred back is the data read from the address which was transmitted with the previous datagram.
The previous access was a write access	The data read back mirrors the previously received write data.

Figure 13: Difference between Read and Write Access

Conclusion:
Consequently, the difference between a read and a write access is that the read access does not transfer data to the addressed register but it transfers the address only; and its 32 data bits are dummies.

NOTE:

→ *Please note that the following read delivers back data read from the address transmitted in the preceding read cycle. The data is latched immediately after the read request.*

AREAS OF SPECIAL CONCERN

Use of Dummy Write Data

! **A read access request datagram uses dummy write data.**
Read data is transferred back to the master with the subsequent read or write access.

i Reading multiple registers can be done in a pipelined fashion. Data that is delivered is latched immediately after the initiated data transfer.

Read and Write Access Examples

For read access to register *XACTUAL* with the address 0x21, the address byte must be set to 0x21 in the access preceding the read access.
For write access to register *VACTUAL*, the address byte must be set to 0x80 + 0x22 = 0xA2. For read access, the data bit can have any value, e.g., 0.

Read and Write Access Examples		
Action	**Data sent to TMC**	**Data received from TMC**
read *XACTUAL*	→ 0x2100000000	← 0xSS[1] & unused data
read *XACTUAL*	→ 0x2100000000	← 0xSS & *XACTUAL*
write *VACTUAL*:= 0x00ABCDEF	→ 0xA200ABCDEF	← 0xSS & *XACTUAL*
write *VACTUAL*:= 0x00123456	→ 0xA200123456	← 0xSS00ABCDEF

Table 4: Read and Write Access Examples

[1] *SS is a placeholder for the status bits SPI_STATUS.*

Data Alignment

All data is right-aligned. Some registers represent unsigned (positive) values; others represent integer values (signed) as two's complement numbers.
Some registers consist of switches that are represented as bits or bit vectors.

SPI Transaction Process

The SPI transaction process is as follows:

- The slave is enabled for SPI transaction by a transition to low level on the chip select input NSCSIN.
- Bit transfer is synchronous to the bus clock SCKIN, with the slave latching the data from SDIIN on the rising edge of SCKIN and driving data to SDOIN following the falling edge.
- The most significant bit is sent first.

i A minimum of 40 SCKIN clock cycles is required for a bus transaction with TMC4331A.

AREAS OF SPECIAL CONCERN !

System Behavior Specifics

Take the following aspects into consideration:

- **Whenever data is read from or written to the TMC4331A,** the first eight bits that are delivered back contain the SPI status *SPI_STATUS* that consists of eight user-selected event bits. The selection of these bits are explained in chapter 5.2. (Page 22).
- **If less than 40 clock cycles are transmitted,** the transfer is not valid; even for read access. However, sending only eight clock cycles can be useful to obtain the SPI status because it sends the status information back first.
- **If more than 40 clocks cycles are transmitted,** the additional bits shifted into SDIIN are shifted out on SDOIN after a 40-clock delay through an internal shift register. This can be used for daisy chaining multiple chips.
- **NSCSIN must be low during the whole bus transaction**. When NSCSIN goes high, the contents of the internal shift register are latched into the internal control register and recognized as a command from the master to the slave. If more than 40 bits are sent, only the last 40 bits received - *before the rising edge of NSCSIN* - are recognized as the command.

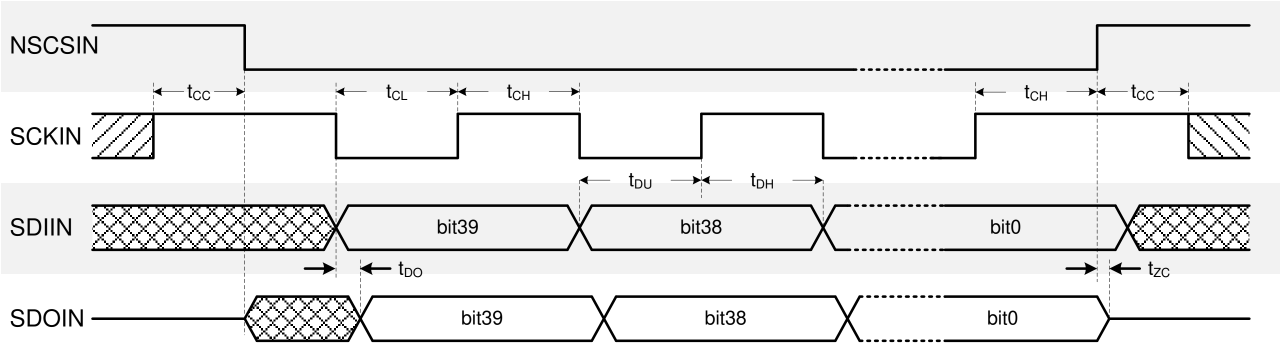

Figure 14: SPI Timing Datagram

3.1.1. SPI Timing Description

The SPI interface is synchronized to the internal system clock, which limits SPI bus clock SCKIN to a quarter of the system clock frequency. The signal processing of SPI inputs is supported with internal Schmitt Trigger, but not with RC elements.

NOTE:

→ *In order to avoid glitches at the inputs of the SPI interface between µC and TMC4331A, external RC elements have to be provided.*

Figure 14 shows the timing parameters of an SPI bus transaction, and the table below specifies the parameter values.

SPI Interface Timing						
SPI Interface Timing	**AC Characteristics:**		**External clock period: t_{CLK}**			
Parameter	**Symbol**	**Conditions**	**Min**	**Type**	**Max**	**Unit**
SCKIN valid before or after change of NSCSIN	t_{CC}		10			ns
NSCSIN high time	t_{CSH}	Min. time is for synchronous CLK with SCKIN high one t_{CH} before SCSIN high only.	t_{CLK}	$>2 \cdot t_{CLK}+10$		ns
SCKIN low time	t_{CL}	Min. time is for synchronous CLK only.	t_{CLK}	$>t_{CLK}+10$		ns
SCKIN high time	t_{CH}	Min. time is for synchronous CLK only.	t_{CLK}	$>t_{CLK}+10$		ns
SCKIN frequency using external clock (Example: f_{CLK} = 16 MHz)	f_{SCK}	Assumes synchronous CLK.			f_{CLK} / 4 (4)	MHz
SDIIN setup time before rising edge of SCKIN	t_{DU}		10			ns
SDIIN hold time after rising edge of SCKIN	t_{DH}		10			ns
Data out valid time after falling SCKIN clock edge	t_{DO}	No capacitive load on SDOIN.			$t_{FILT}+5$	ns

Table 5: SPI Interface Timing

i $t_{CLK} = 1 / f_{CLK}$

4. Input Filtering

Input signals can be noisy due to long cables and circuit paths. To prevent jamming, every input pin provides a Schmitt trigger. Additionally, several signals are passed through a digital filter. Particular input pins are separated into three filtering groups. Each group can be programmed individually according to its filter characteristics. In this chapter informed on the digital filtering feature of TMC4331A is provided; and how to separately set up the digital filter for input pins.

Input Filtering Groups		
Pin Names	**Type**	**Remarks**
STPIN DIRIN	Inputs	Step/Dir interface inputs.
STOPL HOME_REF STOPR	Inputs	Reference input pins.
START	Input	START input pin.

Table 6: Input Filtering Groups (Assigned Pins)

Register Names			
Register Names	**Register Address**		**Remarks**
INPUT_FILT_CONF	0x03	RW	Filter configuration for all four input groups.

Table 7: Input Filtering (Assigned Register)

Input Filter Assignment

Every filtering group can be configured separately with regard to input sample rate and digital filter length.

The following groups exist:

- Step/Dir input pins.
- Reference input pins.
- Start input pin.

Input Sample Rate (SR)

Input sample rate = f_{CLK} $1/2^{SR}$ where:

^{SR} (extended with a particular name extension) is in [0… 7].

i This means that the next input value is considered after 2^{SR} clock cycles.

Sample Rate Configuration

Sample Rate Configuration	
SR Value	**Sample Rate**
0	f_{CLK}
1	$f_{CLK}/2$
2	$f_{CLK}/4$
3	$f_{CLK}/8$
4	$f_{CLK}/16$
5	$f_{CLK}/32$
6	$f_{CLK}/64$
7	$f_{CLK}/128$

Table 8: Sample Rate Configuration

Digital Filter Length (*FILT_L*)

i The filter length *FILT_L* can be set within the range [0… 7].

i The filter length *FILT_L* specifies the number of sampled bits that must have the same voltage level to set a new input bit voltage level.

Digital Filter Length Configuration Table

Configuration of Digital Filter Length	
***FILT_L* value**	**Filter Length**
0	No filtering.
1	2 equal bits.
2	3 equal bits.
3	4 equal bits.
4	5 equal bits.
5	6 equal bits.
6	7 equal bits.
7	8 equal bits.

Table 9: Configuration of Digital Filter Length

4.1. Input Filtering Examples

The following three examples depict input pin filtering of three different input filtering groups.

i After passing Schmitt trigger, voltage levels are compared to internal signals, which are processed by the motion controller.

i The sample points are depicted as green dashed lines.

Example 1: Reference Input Pins

In this example every second clock cycle is sampled. Two sampled input bits must be equal to receive a valid input voltage.

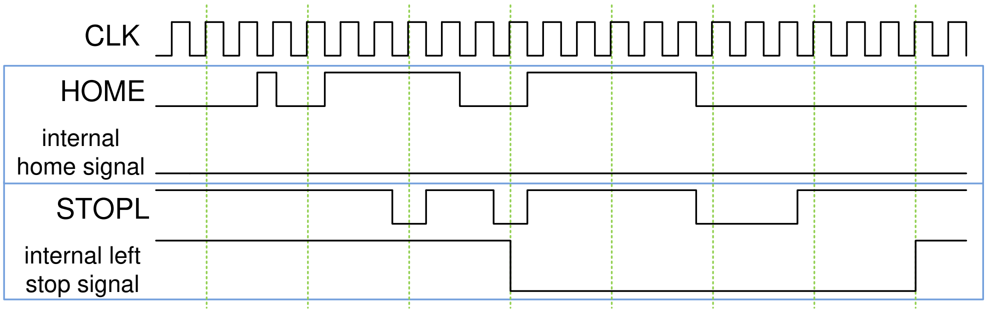

Figure 15: Reference Input Pins: SR_REF = 1, FILT_L_REF = 1

Example 2: START Input Pin

This example shows the START input pattern at every fourth clock cycle:

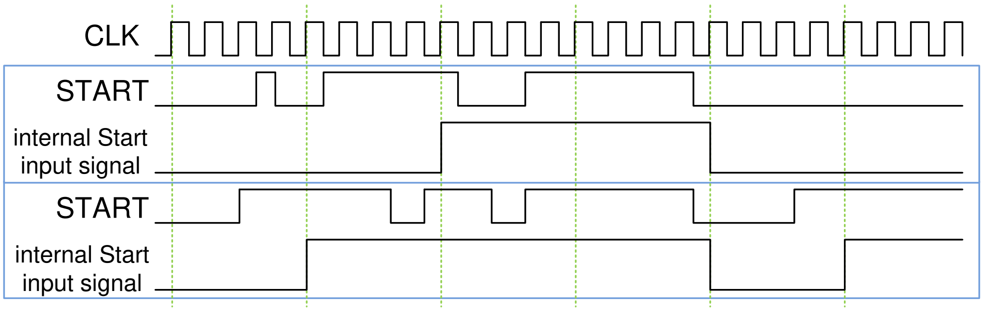

Figure 16: START Input Pin: SR_S = 2, FILT_L_S = 0

Example 3: S/D Input Pins

This example shows every clock cycle bit. Eight sampled input bits must be equal to receive a valid input voltage.

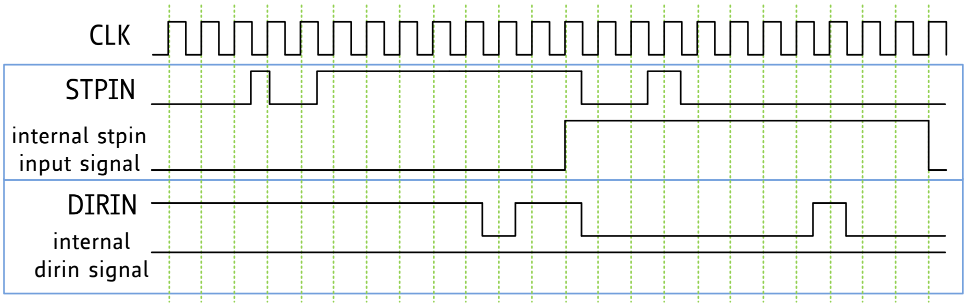

Figure 17: S/D Input Pins: SR_SD_IN = 0, FILT_L_SD_IN = 7

5. Status Flags and Events

TMC4331A provides a range of over 20 status flags and status events in order to obtain short information on the internal status or motor driver status. These flags and events can be read out from dedicated registers. In the following chapter, you are informed about the generation of interrupts based on status events. Status events can also be assigned to the first eight SPI status bits, which are sent within each SPI datagram.

Pin Names: Status Events		
Pin Names	**Type**	**Remarks**
INTR	Output	Interrupt output to indicate status events.

Table 10: Pins Names: Status Events

Register Names: Status Flags and Events			
Register Name	**Register Address**		**Remarks**
GENERAL_CONF	0X00	RW	Bits: 15, 29, 30.
STATUS_FLAGS	0X0F	R	Status flags of TMC4331A and the connected TMC motor driver chip.
EVENTS	0X0E	R+CW	Events triggered by altered TMC4331A status bits.
SPI_STATUS_SELECTION	0X0B	RW	Selection of 8 out of 32 events for SPI status bits.
EVENT_CLEAR_CONF	0X0C	RW	Exceptions for cleared event bits.
INTR_CONF	0X0D	RW	Selection of events for INTR output.

Table 11:Register Names: Status Flags and Events

5.1. Status Event Description

Status events are based on status bits. If the status bits change, related events are triggered from inactive to active level. Resetting events back to inactive must be done manually.

Association of Status Bits

Status bits and status events are associated in different ways:

- Status flags reflect the as-is-condition, whereas status events indicate that the dedicated information has changed since the last read request of the *EVENTS* register. Several status events are associated with one status bit.
- Some status events show the status transition of one or more status bits out of a status bit group. The motor driver flags, e.g., trigger only one motor driver event *MOTOR_EV* in case one of the selected motor driver status flags becomes active.
- In case a flag consists of more than one bit, the number of associated events that can be triggered corresponds to the valid combinations. The *VEL_STATE* flag, e.g., has two bit but three associated velocity state events (b'00/b'01/b'10). Such an event is triggered if the associated combination switches from inactive to active.

NOTE:

→ *Some events have no equivalence in the* STATUS_FLAGS *register 0x0F (e.g.,* COVER_DONE *which indicates new data from the motor driver chip).*

Automatic Clearance of EVENTS

The *EVENTS* register 0x0E is automatically cleared after reading the register; subsequent to an SPI datagram request. Events are important for interrupt generation and SPI status monitoring.

NOTE:

→ *It is recommended to clear EVENTS register 0x0E by read request before regular operation.*

AREAS OF SPECIAL CONCERN

!

How to Avoid Lack of Information

Recognition of a status event can fail; in case it is triggered right before or during *EVENTS* register 0x0E becomes cleared.

In order to prevent events from being cleared, assign *EVENT_CLEAR_CONF* register 0x0C according to the particular event in the *EVENTS* register:

Action:
➢ Set related *EVENT_CLEAR_CONF* register bit position to 1.

Result:
The related event is not cleared when *EVENTS* register is read out.

In order to clear these events, do the following, if necessary:

Action:

➢ Set related *EVENTS* register 0x0E bit position to 1.

Result:
The related event is cleared by writing to the *EVENTS* register.

5.2. SPI Status Bit Transfer

Up to eight events can be selected for permanent SPI status report. Consequently, these events are always transferred at the most significant transfer bits within each TMC4331A SPI response.

Assign an Event to a Status Bit

In order to select an event for the SPI status bits, assign the *SPI_STATUS_SELECTION* register 0x0B according to the particular event in the EVENTS register:

Action:

- Set the related *SPI_STATUS_SELECTION* register bit position to 1.

Result:

The related event is transferred with every SPI datagram response as *SPI_STATUS*.

NOTE:

→ *The bit positions are sorted according to the event bit positions in the EVENTS register 0x0E. In case more than eight events are selected, the first eight bits (starting from index 0 = LSB) are forwarded as SPI_STATUS.*

5.3. Generation of Interrupts

Similar to *EVENT_CLEAR_CONF* register and *SPI_STATUS_SELECTION* register, events can be selected for forwarding via INTR output. The selected events are ORed to one signal which means that INTR output switches active as soon as one of the selected events triggers.

Generate Interrupts

In order to select an event for the INTR output pin, assign the *INTR_CONF* register 0x0D according to the particular event in the EVENTS register:

Action:

- Set the related *INTR_CONF* register bit position to 1.

Result:

The related event is forwarded at the INTR output. If more than one event is requested, INTR becomes active as soon as one of the selected events is active.

INTR Output Polarity

Per default, the INTR output is low active.

In order to change the INTR polarity to high active, do the following:

Action:

- Set *intr_pol* =1 (*GENERAL_CONF* register 0x00).

Result:

INTR is high active.

5.4. Connection of Multiple INTR Pins

INTR pin can be configured for a shared interrupt signal line of several TMC4331A interrupt signals to the microcontroller.

Connecting several Interrupt Pins

In order to make use of a Wired-Or or Wired-And behavior, the below described actions must be taken:

Action:

- **Step 1:** Set *intr_tr_pu_pd_en* = 1 (*GENERAL_CONF* register 0x00).

OPTION 1: WIRED-OR

Action:

- **Step 2:** Set *intr_as_wired_and* = 0 (*GENERAL_CONF* register 0x00).

Result:
The INTR pin works efficiently as Wired-Or (default configuration).

i In case INTR pin is inactive, the pin drive has a weak inactive polarity output. If one of the connected pins is activated, the whole line is set to active polarity.

OPTION 2: WIRED-AND

Action:

- **Step 2:** Set *intr_as_wired_and* = 1 of the *GENERAL_CONF* register 0x00.

Result:
In case no interrupt is active, the INTR pin has a strong inactive polarity output. During the active state, the pin drive has a weak active polarity output. Consequently, the whole signal line is activated in case all pins are forwarding the active polarity.

6. Ramp Configurations for different Motion Profiles

Step generation is one of the main tasks of a stepper motor motion controller. The internal ramp generator of TMC4331A provides several step generation configurations with different motion profiles. They can be configured in combination with the velocity or positioning mode.

Pin Names: Ramp Generator		
Pin Names	**Type**	**Remarks**
STPOUT_PWMA	Output	Step output signal.
DIROUT_PWMB	Output	Direction output signal.

Table 12: Pin Names: Ramp Generator

Register Names: Ramp Generator			
Register Name	**Register Address**		**Remarks**
GENERAL_CONF	0x00	RW	Ramp generator affecting bits 5:0.
STP_LENGTH_ADD	0x10	RW	Additional step length in clock cycles; 16 bits.
DIR_SETUP_TIME			Additional time in clock cycles when no steps will occur after a direction change; 16 bits.
RAMPMODE	0x20	RW	Requested motion profile and operation mode; 3 bits.
XACTUAL	0x21	RW	Current internal microstep position; signed; 32 bits.
VACTUAL	0x22	R	Current step velocity; 24 bits; signed; no decimals.
AACTUAL	0x23	R	Current step acceleration; 24 bits; signed; no decimals.
VMAX	0x24	RW	Maximum permitted or target velocity; signed; 32 bits= 24+8 (24 bits integer part, 8 bits decimal places).
VSTART	0x25	RW	Velocity at ramp start; unsigned; 31 bits=23+8.
VSTOP	0x26	RW	Velocity at ramp end; unsigned; 31 bits=23+8.
VBREAK	0x27	RW	At this velocity value, the aceleration/deceleration will change during trapezoidal ramps; unsigned; 31 bits=23+8.
AMAX	0x28	RW	Maximum permitted or target acceleration; unsigned; 24 bits=22+2 (22 bits integer part, 2 bits decimal places).
DMAX	0x29	RW	Maximum permitted or target deceleration; unsigned; 24 bits=22+2.
ASTART	0x2A	RW	Acceleration at ramp start or below VBREAK; unsigned; 24 bits=22+2.
DFINAL	0x2B	RW	Deceleration at ramp end or below VBREAK; unsigned; 24 bits=22+2.
BOW1	0x2D	RW	First bow value of a complete velocity ramp; unsigned; 24 bits=24+0 (24 bits integer part, no decimal places).
BOW2	0x2E	RW	Second bow value of a complete velocity ramp; unsigned; 24bits=24+0.
BOW3	0x2F	RW	Third bow value of a complete velocity ramp; unsigned; 24 bits=24+0.
BOW4	0x30	RW	Fourth bow value of a complete velocity ramp; unsigned; 24 bits=24+0.
CLK_FREQ	0x31	RW	External clock frequency f_{CLK}; unsigned; 25 bits.
XTARGET	0x37	RW	Target position; signed; 32 bits.

Table 13: Register Names: Ramp Generator

6.1. Step/Dir Output Configuration

This section focuses on the description of the Step/Dir output configuration.

6.1.1. Step/Dir Output Configuration Steps

Step/Dir output signals can be configured for the driver circuit.

If step signals must be longer than one clock cycle, do as follows:

Action:

➢ Set proper *STP_LENGTH_ADD* register 0x10 (bit 15:0).

Result:

The resulting step length is equal to *STP_LENGTH_ADD*+1 clock cycles. This is how the step length is assigned within a range of up to 1-up-to-2^{16} clock cycles.

Action:

➢ Set proper *DIR_SETUP_TIME* register 0x10 (bit 31:16).

Result:

The delay period between DIROUT and STPOUT voltage level transitions last *DIR_SETUP_TIME* clock cycles. No steps are sent via STPOUT for *DIR_SETUP_TIME* clock cycles after a level change at DIROUT.

PRINCIPLE:

DIROUT does not change the level:

- During active step pulse signal
- For (*STP_LENGTH_ADD*+1) clock cycles after the step signal returns to inactive level

6.1.2. STPOUT: Changing Polarity

STPOUT characteristics can be set differently, as follows:

Per default, the step output is high active because a rising edge at STPOUT indicates a step.

In order to change the polarity, do as follows:

Action:

➢ Set *step_inactive_pol* =1 (bit3 of *GENERAL_CONF* register 0x00).

Result:

Each falling edge indicates a step.

How to prompt Level Change with every Step

In order to prompt a step at every level change, do as follows:

Action:

➢ Set *toggle_step* =1 (bit4 of *GENERAL_CONF* register 0x00).

Result:

Every level change indicates a step.

DIROUT: Changing the Polarity

Per default, voltage level 1 at DIROUT indicates a negative step direction. DIROUT characteristics can be set differently, as shown below.

In order to change polarity, do as follows:

Action:

➢ Set *pol_dir_out* =0 (bit5 of *GENERAL_CONF* register 0x00).

Result:

A high voltage level at DIROUT indicates a positive step direction.

NOTE:

→ *DIROUT is based on the internal µStep position MSCNT and is therefore based on the internal SinLUT, see 10.2. , page 83 .*