PCU9661

Parallel bus to 1 channel UFm I²C-bus controller

Rev. 1 — 12 September 2011 Product data sheet

1. General description

The PCU9661 is an advanced single master mode I²C-bus controller. It is a fourth generation bus controller designed for data intensive I²C-bus data transfers. It has a transmit only transfer rate of up to 5 Mbits/s using the new Ultra Fast-mode (UFm) bus with push-pull topology. The serial channel has a generous 4352 byte data buffer which makes the PCU9661 the ideal companion to any CPU that needs to transmit and receive large amounts of serial data with minimal interruptions.

The PCU9661 is an 8-bit parallel-bus to I²C-bus protocol converter. It can be configured to communicate with up to 64 slaves in one serial sequence with no intervention from the CPU. The controller also has a sequence loop control feature that allows it to automatically retransmit a stored sequence.

Its onboard oscillator and PLL allow the controller to generate the clocks for the I²C-bus and for the interval timer used in sequence looping. This feature greatly reduces CPU overhead when data refresh is required in fault tolerant applications.

An external trigger input allows data synchronization with external events. The trigger signal controls the rate at which a stored sequence is re-transmitted over the I²C-bus.

Error reporting is handled at the transaction level, channel level, and controller level. A simple interrupt tree and interrupt masks allow further customization of interrupt management.

The controller parallel bus interface runs at 3.3 V and the I²C-bus I/Os logic levels are referenced to a dedicated $V_{DD(IO)}$ input pin with a range of 3.0 V to 5.5 V.

2. Features and benefits

- Parallel-bus to I²C-bus protocol converter and interface
- 5 Mbit/s unidirectional data transfer Ultra Fast-mode (UFm) channel (push-pull driver)
- Internal oscillator trimmed to 1 % accuracy reduces external components
- 4352-byte UFm channel buffer
- Three levels of reset: software channel reset, global software reset on parallel bus, global hardware $\overline{\text{RESET}}$ pin
- Communicates with up to 64 slaves in one serial sequence
- Sequence looping with interval timer
- JTAG port available for boundary scan testing during board manufacturing process
- Trigger input synchronizes serial communication exactly with external events
- Maskable interrupts
- Operating supply voltage: 3.0 V to 3.6 V (device and host interface)

- I²C-bus I/O supply voltage: 3.0 V to 5.5 V
- Latch-up testing is done to JEDEC Standard JESD78 which exceeds 100 mA
- ESD protection exceeds 8000 V HBM per JESD22-A114 and 1000 V CDM per JESD22-C101
- Packages offered: LQFP48

3. Applications

- Add UFm I²C-bus port to controllers/processors that do not have one
- Add additional UFm I²C-bus ports to controllers/processors that need multiple I²C-bus ports
- Converts 8 bits of parallel data to serial data stream to prevent having to run a large number of traces across the entire printed-circuit board
- Entertainment systems
- LED matrix control
- Data intensive I²C-bus transfers

4. Ordering information

Table 1. Ordering information

Type number	Topside mark	Package		
		Name	Description	Version
PCU9661B	PCU9661	LQFP48	plastic low profile quad flat package; 48 leads; body 7 × 7 × 1.4 mm	SOT313-2

5. Block diagram

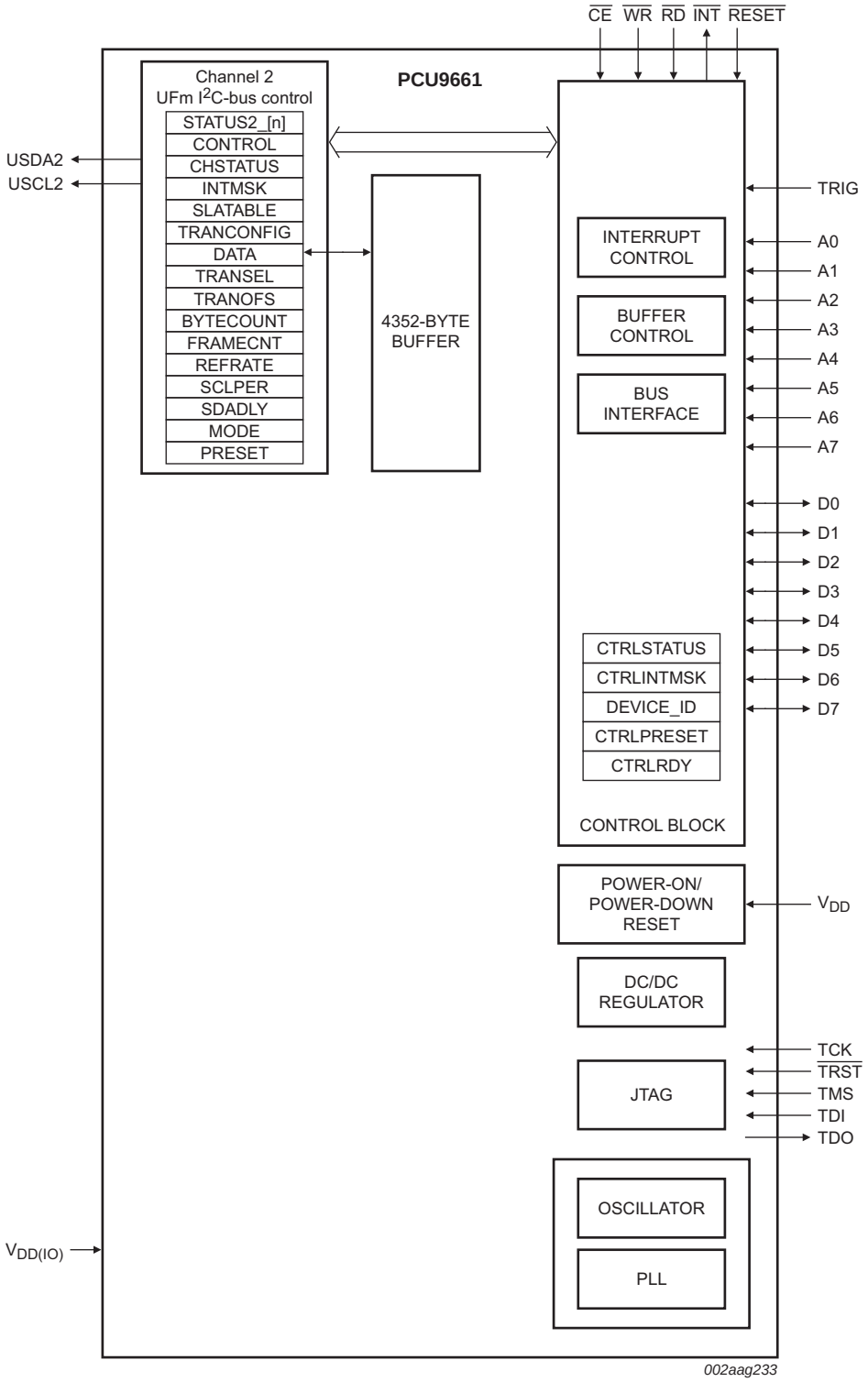


Fig 1. Block diagram

6. Pinning information

6.1 Pinning

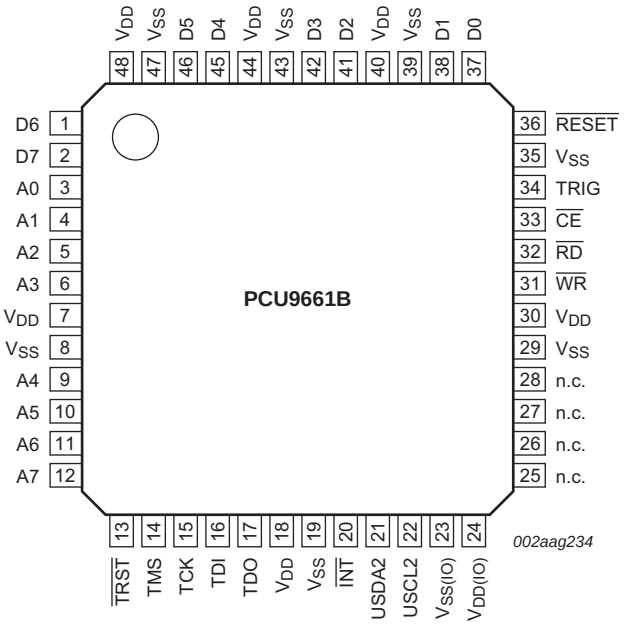


Fig 2. Pin configuration for LQFP48

6.2 Pin description

Table 2. Pin description

Symbol	Pin	Type	Description
A0	3	I	**Address inputs:** selects the bus controller's internal registers and ports for read/write operations. Address is registered when $\overline{CE}$ is LOW and whether $\overline{WR}$ or $\overline{RD}$ transitions LOW. A0 is the least significant bit.
A1	4	I	
A2	5	I	
A3	6	I	
A4	9	I	
A5	10	I	
A6	11	I	
A7	12	I	
D0	37	I/O	**Data bus:** bidirectional 3-state data bus used to transfer commands, data and status between the bus controller and the host. D0 is the least significant bit. Data is registered on the rising edge of $\overline{WR}$ when $\overline{CE}$ is LOW.
D1	38	I/O	
D2	41	I/O	
D3	42	I/O	
D4	45	I/O	
D5	46	I/O	
D6	1	I/O	
D7	2	I/O	

Table 2. Pin description *...continued*

Symbol	Pin	Type	Description
$\overline{TRST}$	13	I	**JTAG test reset input.** For normal operation, hold LOW (V_{SS}).
TMS	14	I	**JTAG test mode select input.** For normal operation, hold HIGH (V_{DD}).
TCK	15	I	**JTAG test clock input.** For normal operation, hold HIGH (V_{DD}).
TDI	16	I	**JTAG test data in input.** For normal operation, hold HIGH (V_{DD}).
TDO	17	O	**JTAG test data out output.** For normal operation, do not connect (n.c.).
$\overline{INT}$	20	O	**Interrupt request:** Active LOW, open-drain, output. This pin requires a pull-up device.
USDA2	21	O	**Channel 2 Ultra Fast-mode I²C-bus serial data output.** Push-pull drive. No pull-up device is needed.
USCL2	22	O	**Channel 2 Ultra Fast-mode I²C-bus serial clock output.** Push-pull drive. No pull-up device is needed.
$\overline{WR}$	31	I	**Write strobe:** When LOW and $\overline{CE}$ is also LOW, the content of the data bus is loaded into the addressed register. Data are latched on the rising edge of $\overline{WR}$. $\overline{CE}$ may remain LOW or transition with $\overline{WR}$.
$\overline{RD}$	32	I	**Read strobe:** When LOW and $\overline{CE}$ is also LOW, causes the contents of the addressed register to be presented on the data bus. The read cycle begins on the falling edge of $\overline{RD}$. Data lines are driven when $\overline{RD}$ and $\overline{CE}$ are LOW. $\overline{CE}$ may transition with $\overline{RD}$.
$\overline{CE}$	33	I	**Chip Enable:** Active LOW input signal. When LOW, data transfers between the host and the bus controller are enabled on D0 to D7 as controlled by the $\overline{WR}$, $\overline{RD}$ and A0 to A7 inputs. When HIGH, places the D0 to D7 lines in the 3-state condition. During the initialization period, $\overline{CE}$ must transition with $\overline{RD}$ until controller is ready.
TRIG	34	I	**Trigger input:** provides the trigger to start a new frame.
$\overline{RESET}$	36	I	**Reset:** Active LOW input. A LOW level resets the device to the power-on state. Internally pulled HIGH through weak pull-up current.
$V_{DD(IO)}$	24	power	**I/O power supply:** 3.0 V to 5.5 V. Power supply reference for I²C-bus pins. Sets the voltage reference point for V_{IL}/V_{IH} and the output drive rail for the UFm channel.
$V_{SS(IO)}$	23	power	**I/O supply ground.** Can be tied to V_{SS}.
V_{DD}	7, 18, 30, 40, 44, 48	power	**Power supply:** 3.0 V to 3.6 V. All V_{DD} pins must be tied together externally.
V_{SS}	8, 19, 29, 35, 39, 43, 47	power	**Supply ground.** All V_{SS} pins must be tied together externally.
n.c.	25, 26, 27, 28	-	not connected

7. Functional description

7.1 General

The PCU9661 acts as an interface device between standard high-speed parallel buses and the serial I²C-bus. On the I²C-bus, it acts as a master. Data transfer between the I²C-bus and the parallel-bus host is carried out on a buffered basis, using either an interrupt or polled handshake.

7.2 Internal oscillator and PLL

The PCU9661 contains an internal 12.0 MHz oscillator and 156 MHz PLL which are used for all internal and I²C-bus timing. The oscillator and PLL require up to $t_{init(po)}$ to start up and lock after power-up. The oscillator is not shut down if the serial bus is disabled.

7.3 Buffer description

Remark: In the following section a ‘transaction’ is defined as a contiguous set of commands and/or data sent/received to/from a single slave. A ‘sequence’ is a set of transactions stored in the buffer.

The PCU9661 serial channel has a 4352-byte data buffer (see Section 7.3.2 “Buffer size”) that allows several transactions to be executed before an interrupt is generated. This allows the host to request several transactions (up to maximum buffer size on each channel) in a single sequence and lets the PCU9661 perform it without the intervention of the host each time a requested transaction is performed. The host can then perform other tasks while the PCU9661 executes the requested sequences.

By following a simple procedure, the I²C-bus controller can store several I²C-bus transactions directed to different slaves addresses on any of the channels. Let us consider the scenario where the host has done the initialization (mode, masks, and other configuration) and writes data into the buffer.

The host starts by programming the buffer configuration registers TRANCONFIG (number of slaves and bytes per slave) and then the SLATABLE (slave addresses). Then the host programs the TRANSEL (Transaction Data Buffer Selection) and the TRANOFS (byte offset selection) to 00h to set the memory pointers to the beginning of the buffer (the default value is 00h after a power-on or $\overline{\text{RESET}}$). Next, the host transfers the data into DATA until the entire sequence is loaded.

Care should be taken so as to not overflow the buffer with excessive read/write commands. In the event of an overflow, represented by the BE bit in the CTRLSTATUS register, will be set to logic 1. The $\overline{\text{INT}}$ pin will be set LOW if the BEMSK bit in the CTRLINTMSK register is logic 0. To recover the channel, a channel reset is required. All configuration and data needs to be checked by the host and resent to the I²C-bus controller. (See Section 7.3.2 “Buffer size”.)

After sending all the commands and data it wanted to the I²C-bus controller, the host writes to the CONTROL register to begin data transmission on the serial channel. The transactions will be sent on the I²C-bus in the order in which the slave addresses are listed in the SLATABLE, separated by a RESTART condition. The last transaction in the sequence will end with a STOP condition.

7.3.1 Buffer management assumptions

- Repeated STARTs will be sent between two consecutive transactions.
- After the last operation on a channel is completed, a STOP will be sent.

7.3.2 Buffer size

The PCU9661 serial channel has a 4352-byte buffer assigned to it. The contents of the buffers should only be modified during channel idle states.

The buffer size represents the memory allocated for the data block only. The slave address table and configuration bytes are contained in other locations and do not need to be included in the required buffer size calculation.

For example, to calculate the size of the memory needed to write 26 bytes to 10 slaves:

10 slaves × 26 bytes/slave = 260 bytes for the write transactions

A total of 260 bytes of buffer space is required to complete the sequence.

Remark: Note that the bytes required to store the 10 slave addresses are not included in the calculation since they are stored in the SLATABLE register.

7.4 Error reporting and handling

In case of any transaction error conditions, the device will load the transaction error status in the STATUS2_[n], generate an interrupt, if unmasked, by pulling down the $\overline{\text{INT}}$ pin and update the CHSTATUS and CTRLSTATUS registers. The status for the individual SLA addresses will be stored in the STATUS2_[n] registers.

7.5 Registers

The PCU9661 contains several registers that are used to configure the operation of the device, status reporting, and to send data. The device also contains global registers for chip level control and status reporting.

The STATUS2_[n] registers are channel-level direct access registers. The DATA, SLATABLE, TRANCONFIG, and BYTECOUNT registers are auto-increment registers.

The memory access pointer to the DATA registers can be programmed using the TRANSEL and TRANOFS registers. See Section 7.5.1.2 “CONTROL — Control register”, for information on the pointer reset bits BPTRRST and AIPTRRST.

Table 3. PCU9661 register address map - direct register access

7	6	5	4	3	2	1	0	Register name	Access	Write access while CH active	Description	Default	Size (bytes)
Channel status registers													
0	0	[3F:00]						reserved	R	no	reserved	00h	64
0	1	[3F:00]						reserved	R	no	reserved	00h	64
1	0	channel 2 transaction number (hex)						STATUS2_[n]	R	no	individual transaction status (direct address) ([7:2] = 0 in UFm)	00h	64
Channel 2 (UFm) registers													
1	1	1	0	0	0	0	0	CONTROL	R/W	yes[1]	channel 2 control ([7] = 1)	00h	1
				0	0	0	1	CHSTATUS	R	no	channel 2 status ([5:1] = 0 in UFm)	00h	1
				0	0	1	0	INTMSK	R/W	yes	channel 2 interrupt mask ([5:1] = don't care)	00h	1
				0	0	1	1	SLATABLE	R/W	no	channel 2 slave address table (auto-increment)	00h	64
				0	1	0	0	TRANCONFIG	R/W	yes, for TRANCOUNT[2]	channel 2 transaction configuration (auto-increment)	00h	65
				0	1	0	1	DATA	R/W	yes	channel 2 data (auto-increment)	00h	bufsize[3]
				0	1	1	0	TRANSEL	R/W	yes	channel 2 transaction data buffer select	00h	1
				0	1	1	1	TRANOFS	R/W	yes	channel 2 transaction data buffer byte offset	00h	1
				1	0	0	0	BYTECOUNT	R	no	channel 2 transmitted byte count (auto-increment)	00h	64
				1	0	0	1	FRAMECNT	R/W	no	channel 2 frame count	01h	1
				1	0	1	0	REFRATE	R/W	no	channel 2 frame refresh rate	00h	1
				1	0	1	1	SCLPER	R/W	no	channel 2 clock period	20h	1
				1	1	0	0	SDADLY	R/W	no	channel 2 SDA delay	08h	1
				1	1	0	1	MODE[4]	R/W	no	channel 2 mode	83h	1
				1	1	1	0	-	-	no	reserved	00h	1
				1	1	1	1	PRESET	R/W	yes	channel 2 parallel reset	00h	1

Table 3. PCU9661 register address map - direct register access *…continued*

7	6	5	4	3	2	1	0	Register name	Access	Write access while CH active	Description	Default	Size (bytes)
Global registers													
1	1	1	1	0	0	0	0	CTRLSTATUS	R	yes	controller status	00h	1
				0	0	0	1	CTRLINTMSK	R/W	yes	master interrupt mask	00h	1
				0	0	1	0	-	R	no	reserved	08h	
				0	0	1	1	-	R	no	reserved	00h	
				0	1	0	0	-	R	no	reserved	00h	
				0	1	0	1	-	R	no	reserved	00h	
				0	1	1	0	DEVICE_ID	R	no	device ID	E1h	
				0	1	1	1	CTRLPRESET	R/W	yes	master parallel reset	00h	1
				1	1	1	1	CTRLRDY[5]	R	no	controller ready register	FFh	1

[1] Except TP and TE. Changing polarity of TP while TE is active will cause a false trigger.

[2] The transaction count (TRANCONFIG[0]) can be written to during the idle period between sequences.

[3] Refer to Section 7.3.2 “Buffer size” for channel memory allocation.

[4] Unused bits in the UFm register set will return 0b when read and writes will be ignored.

[5] Controller ready = FFh immediately after POR or after a hardware reset or global reset. It will clear (00h) once the initialization routine is done.

7.5.1 Channel registers

7.5.1.1 STATUS2_[n] — Transaction status registers

STATUS2_[n] is an 8-bit × 64 read-only registers that provide status information for a given transaction. Only the 2 lower bits are used; the top bits will always read 0. The controller will auto-clear the STATUS2_[n] registers at each START of a sequence when FRAMECNT = 1 and only at the first START when FRAMECNT ≠ 1.

Each register byte can be accessed by direct addressing so that the host can choose to read the status on one or more individual transactions without having to read all 64 status bytes.

Table 4. STATUS2_[n] - Transaction status code register bit description

Bit	Symbol	Description
7:2	ST[7:2]	always reads 0000 00
1	TA	Transaction active. When 1, the transaction is currently active on the serial bus. No interrupt is requested.
0	TR	Transaction ready. When 1, a transaction is loaded in the buffer and waiting to be executed. No interrupt is requested.

Remark: When STATUS2_[n] = 00h, no interrupt is requested and the transaction is in the Done/Idle state.

During program execution, the TR and TA bits behave as follows:

Example, we are to transfer 3 transactions in a sequence. All initialization is completed (loading of SLA, TRANCONFIG, DATA) and device is ready for serial transfer.

Before the STA bit is set, the STATUS2_[n] register will contain:

STATUS2_[0] = 0

STATUS2_[1] = 0

STATUS2_[2] = 0

STATUS2_[3] = 0

:

After STA is set:

STATUS2_[0] = 2

STATUS2_[1] = 1

STATUS2_[2] = 1

STATUS2_[3] = 0

:

Since there is no timing requirement in setting the STA bit after the initialization, the device will update the first status when the STA bit is set and will always go from 0 to 2 (Idle to Transaction active).

7.5.1.2 CONTROL — Control register

CONTROL is an 8-bit register. The STO bit is affected by the bus controller hardware: it is cleared when a STOP condition is present on the I²C-bus.

Table 5. CONTROL - Control register bit description

Address: Channel 2 = E0h.
*Legend: * reset value*

Bit	Symbol	Access	Value	Description
7	STOSEQ	R/W		Stop sequence bit.
			1	When the STOSEQ bit is set while the channel is active, a STOP condition will be transmitted immediately following the end of the current sequence being transferred on the I²C-bus. No further buffered transactions will be carried out and the channel will return to the idle state. Normal error reporting will occur up until the last bit. When a STOP condition is detected on the bus, the hardware clears the STOSEQ flag.
			0*	When STOSEQ is reset, no action will be taken.
6	STA	R/W		The START flag.
			1	The STA bit is set to begin a sequence. The STA bit may be set only at a valid idle state. The controller will reset the bit under the following conditions: • A sequence is done and FRAMECNT = 1. • A sequence loop is done and FRAMECNT > 1. • The STOSEQ bit is set, FRAMECNT = 0, and the current sequence is done. • The STOSEQ bit is set, FRAMECNT > 1, and the current sequence is done. • The STO bit is set and the current byte transaction is done. This bit cannot be set if the CHEN bit is 0.
			0*	When the STA bit is reset, no START condition will be generated.
5	STO	R/W		The STOP flag.
			1	When the STO bit is set while the channel is active, a STOP condition will be transmitted immediately following the current data or slave address byte being transferred on the I²C-bus. No further buffered transactions will be carried out and the channel will return to the idle state. Normal error reporting will occur up until the last bit. When a STOP condition is detected on the bus, the hardware clears the STO flag.
			0*	When the STO bit is reset, no action will be taken.
4	TP	R/W		Trigger polarity bit. Cannot be changed while channel is active.
			1	Trigger will be detected on a falling edge.
			0*	Trigger will be detected on a rising edge.

Table 5. CONTROL - Control register bit description *…continued*

Address: Channel 2 = E0h.
*Legend: * reset value*

Bit	Symbol	Access	Value	Description
3	TE	R/W		Trigger Enable (TE) bit controls the trigger input used for frame refresh. TE cannot be changed while channel is active. When the trigger input is enabled, the trigger will override the contents of the FRAMECNT register and will start triggering when STA bit is set. Thereafter, when a trigger tick is detected, the controller will issue a START command and the stored sequence will be transferred on the serial bus.
			1	When TE = 1, the sequence is controlled by the Trigger input.
			0*	When TE = 0, the trigger inputs are ignored.
2	BPTRRST	W	1	Resets auto increment pointers for BYTECOUNT. Reads back as 0.
1	AIPTRRST	W	1	Resets auto increment pointers for SLATABLE and TRANCONFIG. The DATA register auto-increment pointer will be set to the value that corresponds to TRANSEL and TRANOFS registers. Reads back as 0. **Remark:** To reset the data pointer, write 00h to TRANSEL.
0	-	W	0	Reserved. User must write 0 to this bit.

Remark: Due to a small latency between setting the STA bit and the ability to detect a trigger pulse, if the STA bit is set simultaneously to an incoming trigger pulse, the pulse will be ignored and the controller will wait for the next trigger to send the START.

If the STO or STOSEQ bit are set at anytime while the STA bit is 0, then no action will be taken and the write to these bits is ignored.

Remark: STO has priority over STOSEQ.

Table 6. CONTROL register bits STA, STO, STOSEQ operation/behavior

Channel state (initialization steps)	Next write action by host					Results
	FRAMECNT	TE	STA	STO	STOSEQ	
Idle (reset, TRANCONFIG, SLATABLE, DATA, STA = 0)	1	0	0	X	X	No action.
	1	0	1	X	X	START transmitted on serial bus followed by sequence stored in buffer.
Active (reset, load TRANCONFIG, SLATABLE, DATA, STA = 1	1	0	X	0	X	No change; cannot write STA while active.
	1	0	X	1	X	When the STO bit is set: 1. A STOP is sent after the end of ACK cycle of the current byte and BYTECNT is updated. The SD bits will be set.
REFRATE Loop idle (reset, load TRANCONFIG, SLATABLE, DATA STA = 1)[1]	≠ 1	0	0	X	X	No action.
	≠ 1	0	X	0	1	Channel will go immediately to the inactive state and SD and FLD bits will be set.[2]
	≠ 1	0	X	1	X	Channel will go immediately to the inactive state and SD and FLD bits will be set.[2]
REFRATE Loop active (reset, load, TRANCONFIG, SLATABLE, DATA, STA = 1)	≠ 1	0	X	0	0	No action.
	≠ 1	0	X	0	1	STOP at end of current frame. The SD and FLD bits will be set.
	≠ 1	0	X	1	X	When the STO bit is set: 1. A STOP is sent after the end of ACK cycle of the current byte and BYTECNT is updated. The SD and FLD bits will be set.
Trigger Loop Idle (reset, load TRANCONFIG, SLATABLE, DATA, STA = 1)	X	1	0	X	X	No action.
	X	1	X	0	1	STOP at end of current frame. The SD and FLD bits will be set.
	X	1	X	1	X	When the STO bit is set: 1. A STOP is sent after the end of ACK cycle of the current byte and the BYTECNT is updated. The SD and FLD bits will be set.
Trigger Loop active (reset, load TRANCONFIG, SLATABLE, DATA, STA = 1)	X	1	X	0	0	No action.
	X	1	X	0	1	Channel will go immediately to the inactive state and SD and FLD bits will be set.[2]
	X	1	X	1	X	Channel will go immediately to the inactive state and SD and FLD bits will be set.[2]

[1] Loop Idle is defined as the time elapsed from a STOP to the START of the next sequence while STA = 1.

[2] Channel Active is defined by the CTRLSTATUS[5:3] bits.

7.5.1.3 CHSTATUS — Channel status register

CHSTATUS is an 8-bit read-only register that provides status information for the serial channel. All these status drive the $\overline{INT}$ pin active LOW. To clear the channel interrupt request, you must read the CHSTATUS register. The BE interrupt is cleared by reading the CTRLSTATUS register.

After the CHSTATUS register is cleared, only new errors or status updates will cause the CHSTATUS bits to be set.

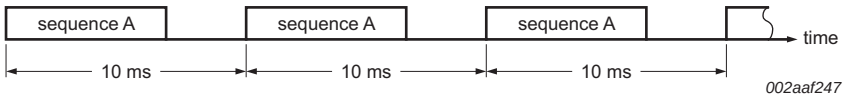
Table 7. CHSTATUS - Channel and buffer status codes register bit description

Address: Channel 2 = E1h.

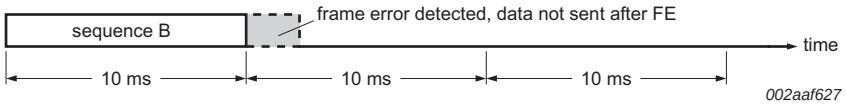
Bit	Symbol	Description
7	SD	Sequence Done. The sequence loaded in the buffer was sent and STOP issued on the serial bus.
6	FLD	Frame Loop Done. The FRAMECNT value has been reached. A STOP has been issued on the bus.
5:1	-	Reserved.
0	FE	Frame Error detected. The time required to send the sequence exceeds refresh rate programmed to the REFRATE register or the time between trigger ticks.

[1] Bits [5:1] always read as logic 0.

FE - Frame Error bit: This bit indicates that the time required to send the sequence exceeds the refresh rate programmed in the REFRATE register or the time between trigger ticks. Solving frame errors include programming longer refresh rates, speeding up the bus frequency, shortening the amount of bytes sent/received in the sequence, or increasing the time between trigger ticks. If the frame error is masked by the FEMSK, the device will continue to transmit transactions until the end of the sequence without re-starting the sequence even if new triggers are detected. The total number of sequences transmitted will be the number stored in the FRAMECNT register. Once a complete sequence is transmitted, a new sequence will initiate when a subsequent trigger appears. The FE flag will be held HIGH and sequences will still be transmitted unless CHSTATUS is read. If the frame error is unmasked, the sequence will be aborted at the next logical stopping point, a STOP transmitted and an interrupt will be generated. The FE bit is set after the STOP is detected on the bus.

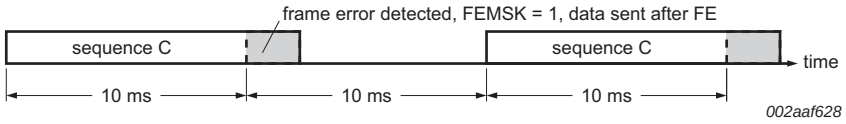


a. Sequence fully executed within the period programmed in REFRATE register



This condition causes a frame error and the FE bit to be set.

b. Sequence exceeds period programmed in REFRATE register, FEMSK = 0



c. Sequence exceeds period programmed in REFRATE register, FEMSK = 1

Fig 3. Frame Error detection

7.5.1.4 INTMSK — Interrupt mask register

Through the INTMSK register, there is the option to manage which states generate an interrupt, allowing more control from the host on the transaction. The interrupt mask applies to all transactions on the channel. A bit set to 1 indicates that the mask is active. The INTMSK register default is all interrupts are un-masked (00h).

Table 8. INTMSK - Interrupt mask register bit description

Address: Channel 0 = C2h; Channel 1 = D2h; Channel 2 = E2h.

Bit	Symbol	Description
7	SDMSK	Sequence Done Mask. The end of sequence interrupt will not be generated.
6	FLDMSK	Frame loop done mask. A frame loop done interrupt will not be generated. The controller will enter the idle state.
5:1	-	reserved
0	FEMSK	Frame Error Mask. A frame error interrupt will not be generated. **Remark:** Use caution and good judgement when using this mask. Unexpected/erratic behavior may result in the slave devices.

7.5.1.5 SLATABLE — Slave address table register

SLATABLE is an 8-bit × 64 register set that makes up a table that stores the slave address for each transaction in the sequence. The table is loaded by using an auto-increment pointer that is not user-accessible. To reset the pointer, the AIPTRRST bit must be set in the CONTROL register. The slave addresses in the SLATABLE register are stored with a zero-based (N – 1) index. The first slave address occupies the 00h position.

Remark: Slave address entries greater than the transaction count are not part of the sequence. TRANCONFIG[0] contains the transaction count that will be included in the sequence.

Table 9. SLATABLE - Slave address table register bit description

Address: Channel 2 = E3h.

Bit	Symbol	Description
7:1	SLATABLE[7:1]	Slave address.
0	SLATABLE[0]	When 1, a read transaction will be ignored and a write transaction is requested. When 0, a write transaction is requested.

Table 10. Example of SLATABLE registers

Transaction	Slave address
00h	10h
01h	12h
02h	28h
03h	40h
04h	14h
:	:
3Fh	36h

7.5.1.6 TRANCONFIG — Transaction configuration register

The TRANCONFIG register is an 8-bit × 65 register set that makes up a table that contains the number of transactions that will be executed in a sequence and the number of data bytes involved in the transaction.

The first byte of the register is the Transaction Count register. The remaining 64 registers are the Transaction Length registers.

Table 11. TRANCONFIG, byte 0 - Transaction configuration register bit description

Address: Channel 2 = E4h.

Bit	Symbol	Description
7:0		Number of transactions in the sequence. Maximum is 40h.

Table 12. TRANCONFIG, byte 1 to 40h - Transaction configuration register bit description

Bit	Symbol	Description
7:0		Number of bytes per transaction in the sequence. Maximum is FFh.

Table 13. Example of TRANCONFIG register loaded

Register	Value	Description
Transaction count	10h	16 transactions = 16 slave addresses in the SLATABLE
Transaction length 00h	0Ah	10 byte transaction
Transaction length 01h	12h	18 byte transaction
Transaction length 02h	28h	40 byte transaction
Transaction length 03h	40h	64 byte transaction
:	:	:
Transaction length 3Fh	12h	18 byte transaction

Remark: Even if the Transaction length (TRANCONFIG[1:40h]) and the SLATABLE([0:3Fh]) are fully initialized, only the specified number of transactions in the Transaction count (TRANCONFIG[0]) will be part of the sequence.

If the Transaction count is 0, then there will be no activity on the serial bus if the STA bit is set. In addition, there will be no interrupts generated or status updated. The controller will simply reset the CONTROL.STA bit without performing any transactions.

If the Transaction length is 0, a write transaction will send the slave address plus write bit (SLA+W) on the serial bus with no data bytes.

7.5.1.7 DATA — I²C-bus Data register

DATA is an 8-bit read/write, auto-increment register. It is the interface port to the channel buffer. When accessing the buffer, the host writes a byte of serial data to be transmitted at this location. The host can read from the DATA at any time and can only write to this 8-bit register while the channel is idle.

The host can read or write data up to the amount of memory space allotted to the channel. The location at which the data is accessed is stored in the TRANSEL and TRANOFS register (both default at 00h).

To return to the data location pointed by the contents of the TRANSEL and TRANOFS register after read or write access to the DATA register, set the AIPTRRST (auto-increment pointer reset) bit in the control register.

To return to the first DATA register location in the buffer set the TRANSEL to 00h.

Table 14. DATA - Data register bit description

Address: Channel 2 = E5h.

Bit	Symbol	Description
7:0	D[7:0]	Eight bits to be transmitted. A logic 1 in DATA corresponds to a HIGH level on the I²C-bus. A logic 0 corresponds to a LOW level on the bus.

7.5.1.8 TRANSEL — Transaction data buffer select register

The TRANSEL register is used to select the pointer to a specific transaction in the DATA buffer. This allows the user to update the data of a specific slave without having to re-write the entire data buffer. The value of this register is the slave address position in the SLATABLE register. The TRANSEL register is zero-based (N – 1) register.

For example, if a change to the 22nd slave address data is required, the host would set the TRANSEL register to 15h. This register can be used in conjunction with the TRANSOFS register to access a specific byte in the data buffer. The host would then proceed to write the new data to the DATA register. The auto-increment feature continues to operate from this new position in the DATA register.

Setting TRANSEL to an uninitialized TRANCONFIG entry may cause a request to read/write data outside the data buffer. If this occurs, the BE bit in the CTRLSTATUS register will be set to a logic 1. Write data will be ignored and read data will be invalid.

When a new transaction is selected by programming the TRANSEL registers, the TRANSOFS register will automatically be reset to 00h.

Remark: When updating the data buffer, if the number of bytes to be updated or read exceeds the number of bytes that were specified in the TRANCONFIG register, the auto-increment will go over the transaction boundary into the next transaction stored in the buffer.

Remark: To reset the DATA pointer, write 00h to the TRANSEL register.

Table 15. TRANSEL - Transaction data buffer select register bit description

Address: Channel 2 = E6h.

Bit	Symbol	Description
7	-	Reserved.
6	-	Reserved.
5:0	TRANSEL[5:0]	Slave address position in the SLATABLE. The maximum number is 3Fh.

7.5.1.9 TRANOFS — Transaction data buffer byte select register

In conjunction with the TRANSEL register, the TRANOFS register is used to select the pointer to a specific byte in a transaction in the data buffer. This allows the user to read or re-write a specific data byte of a specific slave without having to read/re-write the entire data buffer. The TRANOFS register is zero-based (N − 1), so the maximum bytes this register will point to is 256.

For example, if the tenth byte in the 40th slave address data is required, the host would set the TRANSEL register to 27h and the TRANSOFS register to 09h. The host would then proceed with a read to the DATA register.

Setting TRANOFS to a byte offset outside of the data buffer will cause the BE bit in the CTRLSTATUS register will be set to a logic 1. Write data will be ignored and read data will be invalid.

Remark: The number of bytes to be updated or read should not exceed the number of bytes that were specified in the TRANCONFIG register. Doing so will cause the auto-increment to go over the transaction boundary into the next transaction stored in the buffer.

Table 16. TRANOFS - Transaction data buffer byte select register bit description
Address: Channel 2 = E7h.

Bit	Symbol	Description
7:0	TRANOFS[7:0]	Byte index for the specified transaction buffer in TRANSEL.

7.5.1.10 BYTECOUNT — Transmitted and received byte count register

The BYTECOUNT register stores the number of bytes that have been sent. The count is continuously updated, therefore the BYTECOUNT is a real time reporting of transmitted and received bytes. This is a read-only register. The BYTECOUNT includes only the bytes that have been ACKed in a write transaction. The BYTECOUNT register is cleared at the START of every sequence.

Table 17. BYTECOUNT, byte 0 - Transaction configuration register bit description
Address: Channel 2 = E8h.

Bit	Symbol	Description
7:0	BYTECOUNT[7:0]	Number of bytes sent per transaction in the sequence. Maximum is FFh.

7.5.1.11 FRAMECNT — Frame count register

Table 18. FRAMECNT - Frame count register bit description
Address: Channel 2 = E9h.

Bit	Symbol	Description
7:0	FRAMECNT[7:0]	Bit 7 to bit 0 indicate the number of times buffered commands are to be re-transmitted. Default is 01h.

This register is a read/write register. The contents of this register holds the programmed value by the host and is not a real-time count of frames sent on the serial bus.

If the FRAMECNT is 00h, the sequence stored in the buffer will loop continuously. A STOP will be sent at the end of each sequence.

If the FRAMECNT is 01h, it is defined as the default state and the sequence stored in the buffer will be sent once and a STOP will be sent at the end of the sequence.

If the FRAMECNT is greater than 01h, the sequence stored in the buffer will loop FRAMECNT times and a STOP will be sent at the end of each sequence.

Remark: The FRAMECNT can only be set to loop on the sequence stored in the buffer.

7.5.1.12 REFRATE — Refresh rate register

The REFRATE register defines the time period between each sequence start when REFRATE looping is enabled (FRAMECNT ≠ 1, and TE = 0).

The refresh period defined by REFRATE should always be programmed to be greater than the time it takes for the sequence to be transferred on the I²C-bus. If the REFRATE values is too small, the frame error (FE) bit will be set and an interrupt will be requested.

Table 19. REFRATE - Refresh rate register bit description

Address: Channel 2 = EAh.

Bit	Symbol	Description
7:0	REFRATE[7:0]	Bit 7 to bit 0 indicate the sequence refresh period. The resolution is 100 μs. The default value is 00h, the timer is disabled, and the sequences will be sent back-to-back if the FRAMECNT is = 0 or FRAMECNT is > 1.

Remark: If the FRAMECNT is 1, then the refresh rate function will be disabled.

7.5.1.13 SCLPER, SDADLY — Clock rate registers

The clock rate register for the Ultra Fast-mode channel is controlled by the SCLPER and SDADLY registers. They define the data rate for the serial bus of the PCU9661. The actual frequency on the serial bus is determined by t_{HIGH} (time where SCL is HIGH), t_{LOW} (time where SCL is LOW), t_r (rise time), and t_f (fall time) values. Writing illegal values into SCLPER registers will cause the part to operate at the maximum channel frequency.

For UFm mode, the clock is a fixed 50 % duty cycle defined by the SCLPER and t_r and t_f are system/application dependent.

Table 20. SCLPER - Clock Period register bit description (Ultra Fast mode)

Address: Channel 2 = EBh.

Bit	Symbol	Description
7:0	L[7:0]	Eight bits defining the clock period (Ultra Fast mode). Default 32 (20h).

Table 21. SDADLY - SDA delay register bit description (Ultra Fast mode)

Address: Channel 2 = ECh.

Bit	Symbol	Description
7:6	H[7:6]	Reserved. Read only read back zero.
5:0	H[5:0]	Six bits defining the SDA delay (Ultra Fast mode). Default: 8 (08h).

Calculating clock settings for Ultra Fast mode (UFm):

The clock period is defined as follows (50 % duty cycle):

$$SCLPER_{(min)} = \frac{1}{T_{PLL} \times freq} \quad (1)$$

The data will be delayed with respect to the falling edge of the clock as follows:

$$SDADLY_{(max)} = \frac{SCLPER}{4} \quad (2)$$

Table 22. Sample clock period and allowable data delay

Frequency	SCLPER[1]	SDADLY[2]
5.0 MHz	32	2 to 8
4.0 MHz	39	2 to 9
3.0 MHz	53	2 to 13
2.0 MHz	79	2 to 19
1.0 MHz	158	2 to 39

[1] The minimum allowable value that can be stored in SCLPER is 32.

[2] The minimum allowable value that can be stored in SDADLY is 2.

The PCU9661 will force a 50 % duty cycle by shifting the contents of the SCLPER register right by 1.

When the user writes the SCLPER register, the SDADLY will be loaded automatically with a value ¼ the value of SCLPER (SCLPER register value right shifted twice). The user can then overwrite the SDADLY register if desired.

The order in which the registers should be written is first the SCLPER, then the SDADLY register to adjust the delay.

The maximum value for SDADLY is the preferred value to be loaded.

7.5.1.14 MODE — I²C-bus mode register

MODE is a read/write register. It contains the control bits that select the bus recovery options, and the correct timing parameters. Timing parameters involved with AC[1:0] are t_{BUF}, $t_{HD;STA}$, $t_{SU;STA}$, $t_{SU;STO}$, t_{HIGH}, t_{LOW}. The auto recovery and bus recovery bits are contained in this register. They control the bus recovery sequence.

Table 23. MODE - I²C-bus mode register bit description

Address: Channel 2 = EDh.

Bit	Symbol	Description
7	CHEN	Channel Enable bit. R/W. 0: Channel is disabled, SCL and SDA high-impedance, USDA and USCL driven HIGH. All registers are accessible for setup and configuration, however a sequence cannot be started if the CHEN bit is 0 (STA cannot be set). 1 (default): Channel is enabled.
6:2	-	Reserved.
UFm Channel 2		
1:0	AC[1:0]	I²C-bus mode selection to ensure proper timing parameters (see Table 33). AC[1:0] = 00: Reserved. AC[1:0] = 01: Reserved. AC[1:0] = 10: Reserved. AC[1:0] = 11 (default): Ultra Fast-mode AC parameters selected. Read-only bits.

Remark: CHEN bit value must be changed only when the I²C-bus is idle.

Remark: The AC[1:0] are read-only bits. The UFm channel AC parameters are controlled internally.

7.5.1.15 PRESET — I²C-bus channel parallel software reset register

Table 24. PRESET - I²C-bus channel parallel software reset register bit description

Address: Channel 2 = EFh.

Bit	Symbol	Description
7:0	PRESET[7:0]	Read/Write register used during an I²C-bus channel parallel reset command.

PRESET is an 8-bit write-only register. Programming the PRESET register allows the user to reset the channel under software control. The software reset is achieved by writing two consecutive bytes to this register. The first byte must be A5h while the second byte must be 5Ah. The writes must be consecutive and the values must match A5h and 5Ah. If this sequence is not followed as described, the reset is aborted.

The PRESET resets state-machines, registers, and buffer pointers to the default values, zeroes the TRANCONFIG, SLATABLE, BYTECOUNT, and DATA arrays of the respective channel and will not reset the entire chip. The parallel bus remains active while a software reset is active. The user can read the PRESET register to determine when the reset has completed, PRESET returns all 1s when the reset is active and all 0s when complete.

7.5.2 Global registers

7.5.2.1 CTRLSTATUS — Controller status register

The CTRLSTATUS register reports the status of the controller, including the interrupts generated by the parallel bus. There are three status bits. When CTRLSTATUS contains 00h, it indicates the idle state and therefore no serial interrupts are requested. The content of this register is continuously updated during the operation of the controller.

Bit 2 indicates the serial channel has an interrupt request pending. To clear the serial channel interrupt request, you must read the CHSTATUS register. Bit 5 indicates if the serial channel is currently active or if it is in the idle state.

Table 25. CTRLSTATUS - Interrupt status register bit description

Address: F0h.

Bit	Symbol	Description
7	BE	Buffer Error. A buffer error such as overflow has been detected.
6	-	Reserved.
5	CH2ACT	Channel 2 is active.
4	-	Reserved.
3	-	Reserved.
2	CH2INTP	Channel 2 interrupt pending.
1	-	Reserved.
0	-	Reserved.

Remark: A global reset will reset all channels and configuration settings.

BE - Buffer Error bit: This bit indicates that a buffer error has been detected. For example, a buffer overflow due to the host programming too many bytes will set this bit. A software or hardware reset is necessary to recover from a buffer error.

The buffer error may occur when a data location is being read or written to that has not previously been configured by the TRANCONFIG register. The buffer error can occur on a parallel data write or read beyond the buffer capacity, or setting the TRANSEL and TRANOFS pointers beyond the buffer boundary.

When the DATA register is loaded with data that goes beyond the capacity of the buffer, the bytes that go over the buffer size will be ignored and a Buffer Error (BE) will be generated.

Special case: The BE interrupt is cleared by reading the CTRLSTATUS register. All other interrupts are cleared by reading the respective CHSTATUS register.

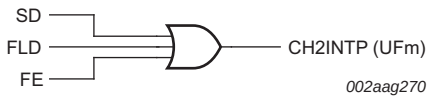


Fig 4. PCU9661 status reporting logic

See Table 7 for channel status.

7.5.2.2 CTRLINTMSK — Control Interrupt mask register

The CTRLINTMSK masks all interrupts generated by the masked channel. This allows the host MCU to complete other operations before servicing the interrupt without being interrupted by the same channel.

Table 26. CTRLINTMSK - Control interrupt mask register bit description

Address: F1h.

Bit	Symbol	Description
7	BEMSK	Buffer Error Mask. A buffer error interrupt will not be generated. **Remark:** Use caution and good judgement when using this mask. Unexpected/erratic behavior may result in the slave devices.
6:3	-	reserved
2	CH2MSK	When this bit is set to 1, all interrupts for the channel will be masked and the $\overline{INT}$ pin will not be pulled LOW.
1:0	-	reserved

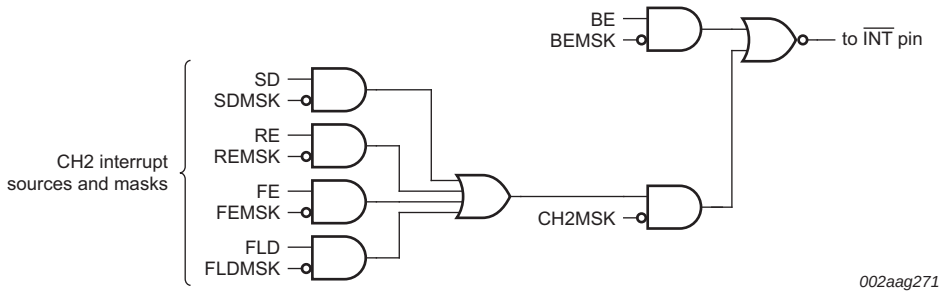


Fig 5. PCU9661 interrupt logic

See Table 8 for interrupt mask.

7.5.2.3 DEVICE_ID — Device ID

The DEVICE_ID register stores the bus controller part number so it can be identified on the parallel bus.

Table 27. DEVICE_ID - Device ID register bit description

Address: F6h.

Bit	Symbol	Description
7	U/A	Selects PCU or PCA device. 1 = PCU96xx 0 = PCA96xx
6:0	BCD	BCD (Binary Coded Decimal) code of the ending 2 digits for ID. Range is 00h to 79h. The code for the PCU9661 is E1h.

7.5.2.4 CTRLPRESET — Parallel software reset register

Table 28. CTRLPRESET - Parallel software reset register bit description

Address: F7h.

Bit	Symbol	Description
7:0	CTRLPRESET[7:0]	Write-only register used during a device parallel reset command.

CTRLPRESET is an 8-bit write-only register. Programming the CTRLPRESET register allows the user to reset the PCU9661 under software control. The software reset is achieved by writing two consecutive bytes to this register. The first byte must be A5h while the second byte must be 5Ah. The writes must be consecutive and the values must match A5h and 5Ah. If this sequence is not followed as described, the reset is aborted.

7.5.2.5 CTRLRDY — Controller ready register

Table 29. CTRLRDY - Controller ready register bit description

Address: FFh.

Bit	Symbol	Description
7:0	CTRLRDY[7:0]	Read-only register indicates the internal state of the controller. FFh indicates the controller is initializing, 00h indicates controller is in normal operating mode.

CTRLRDY (address FFh) is an 8-bit read-only register. It indicates the internal state of the controller. When the register is FFh, the controller is in the initialization state. The initialization state will be entered at power-up, after a hardware reset, or after a global software reset.

The oscillator and the PLL will be initialized only after a Power-On Reset (POR), a hardware reset, or a global software reset (CTRLPRESET).

When the register is 00h, the controller is in the normal operating mode.

Access while the controller is initializing requires $\overline{CE}$ pin follow the $\overline{RD}$ pin transitions to update the state of the controller that is read back. After controller is ready, the $\overline{CE}$ pin can be held LOW while $\overline{RD}$ and $\overline{WR}$ pins transition. See Figure 6, Figure 7 and Figure 8.

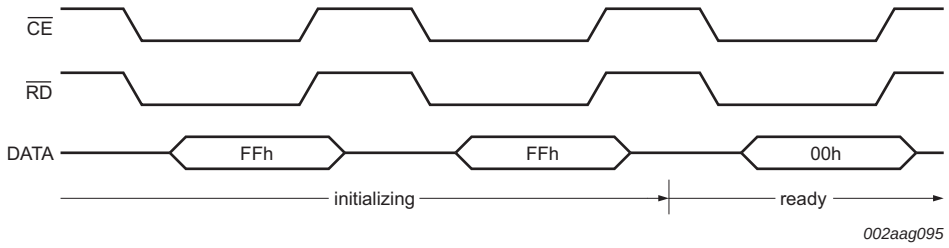


Fig 6. During initialization, $\overline{CE}$ must transition with $\overline{RD}$ at each read operation

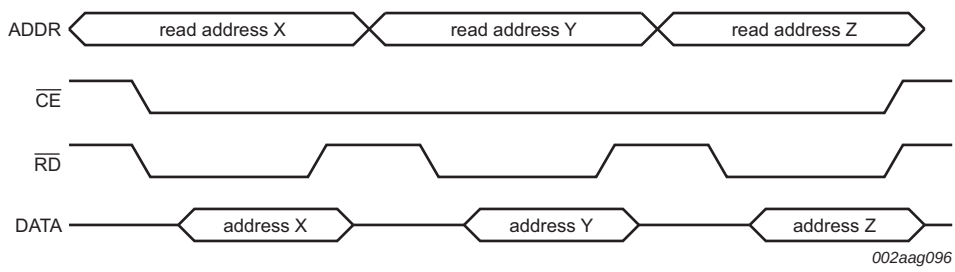


Fig 7. During normal operation, $\overline{CE}$ may remain LOW while $\overline{RD}$ transitions during multiple reads

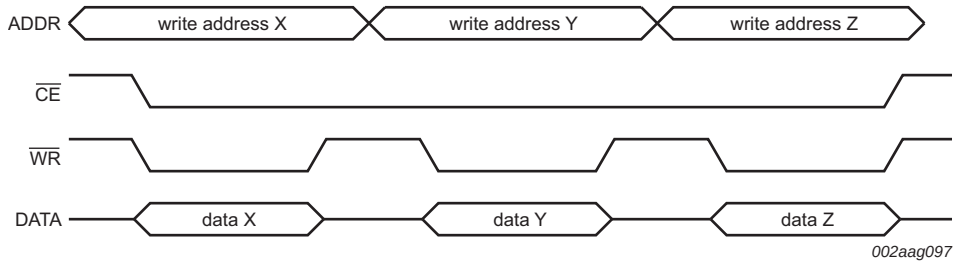


Fig 8. During normal operation, $\overline{CE}$ may remain LOW while $\overline{WR}$ transitions during multiple writes

8. PCU9661 operation

The PCU9661 is designed to efficiently transmit large amounts of data on a single master bus. There are three major components that compose the architecture of the I2C-bus controller that interact with each other to provide a high throughput and a high level of automation when it conducts transactions:

- Slave address table: specifies the address of the slaves on the bus and the direction (write only).
- Transaction configuration: specifies the size of the transaction.
- Data buffer: contains the data to be transmitted from the slave.

These three components are integrated in the PCU9661 to build a sequence. A sequence is a set of write transactions and the minimum sequence size is one write transaction. Several transactions can be stored in one sequence and be executed without the intervention of the host controller (CPU) through loop control and using the built-in refresh rate timers.

The PCU9661 executes transactions in the order they were loaded into the buffer without interrupting the host. Once the end of a sequence is reached, the Sequence Done (SD) bit will be asserted in the CHSTATUS register and the controller will request an interrupt, if SDMSK = 0. At this point, the host can reload the buffer with a new sequence or resend the one that is currently loaded in the buffer.

When a sequence is in progress, no interrupts are generated unless there is an error when a transaction is conducted. The host will only receive an interrupt when the sequence is done. The PCU9661 will behave as a Master Transmitter regardless of the direction bits specified in the SLATABLE. The host has the ability to update the data buffer while the controller carries on the remaining transactions in the sequence.

8.1 Sequence execution

Sequences can only be of one type:

- Write transactions, where the PCU9661 will behave as a Master Transmitter

Data transfers in one direction are shown in Figure 9. This figure contains the following abbreviations:

S — START condition

SLA — 7-bit slave address

W — Write bit (LOW level at SDA)

$\overline{A}$ — Not acknowledge bit (HIGH level at SDA)

Data — 8-bit data byte

P — STOP condition

In Figure 9, circles are used to indicate when a bit is set in the CHSTATUS register. A channel interrupt is not requested when CHSTATUS = 00h and the $\overline{INT}$ pin is not asserted when the interrupt is masked (see Section 7.5.2.2).

For a successful sequence execution, all three components mentioned above must exist in the memory and must be correctly set up. There are not safeguards against programming incorrect transaction sizes, data buffer lengths, or direction bits. If the transaction length is set to 00h, then only the slave address with direction bit will be transmitted.

Once the host has configured the serial port and programmed the TRANCONFIG (number of slaves and bytes per slave), the SLATABLE (slave addresses), TRANSEL (transaction data buffer selection) and the TRANOFS (byte offset selection) and loaded the serial data into the DATA buffer, the sequence is ready to be transmitted.

To send the sequence, the host will set the STA bit in the CONTROL register and the controller will immediately send a START on the serial bus. Then, the transactions will be carried out in the order they appear in the SLATABLE, each being separated by a ReSTART command.

If the interrupts are unmasked, the serial transfer will be conducted without generating interrupts in between transactions. Once all transactions are successfully completed, the controller will generate a STOP, the Sequence Done bit (SD) will be set in the CHSTATUS and an interrupt will be generated.

Once all transactions are completed, the controller will generate a STOP and the Sequence Done bit (SD) will be set in the CHSTATUS and an interrupt will be generated.

If the host wants to poll the PCU9661, it can mask all registers including the SD bit and read the CTRLSTATUS, CHSTATUS, STATUS2_[n], and/or the CONTROL registers to determine the state of the controller.

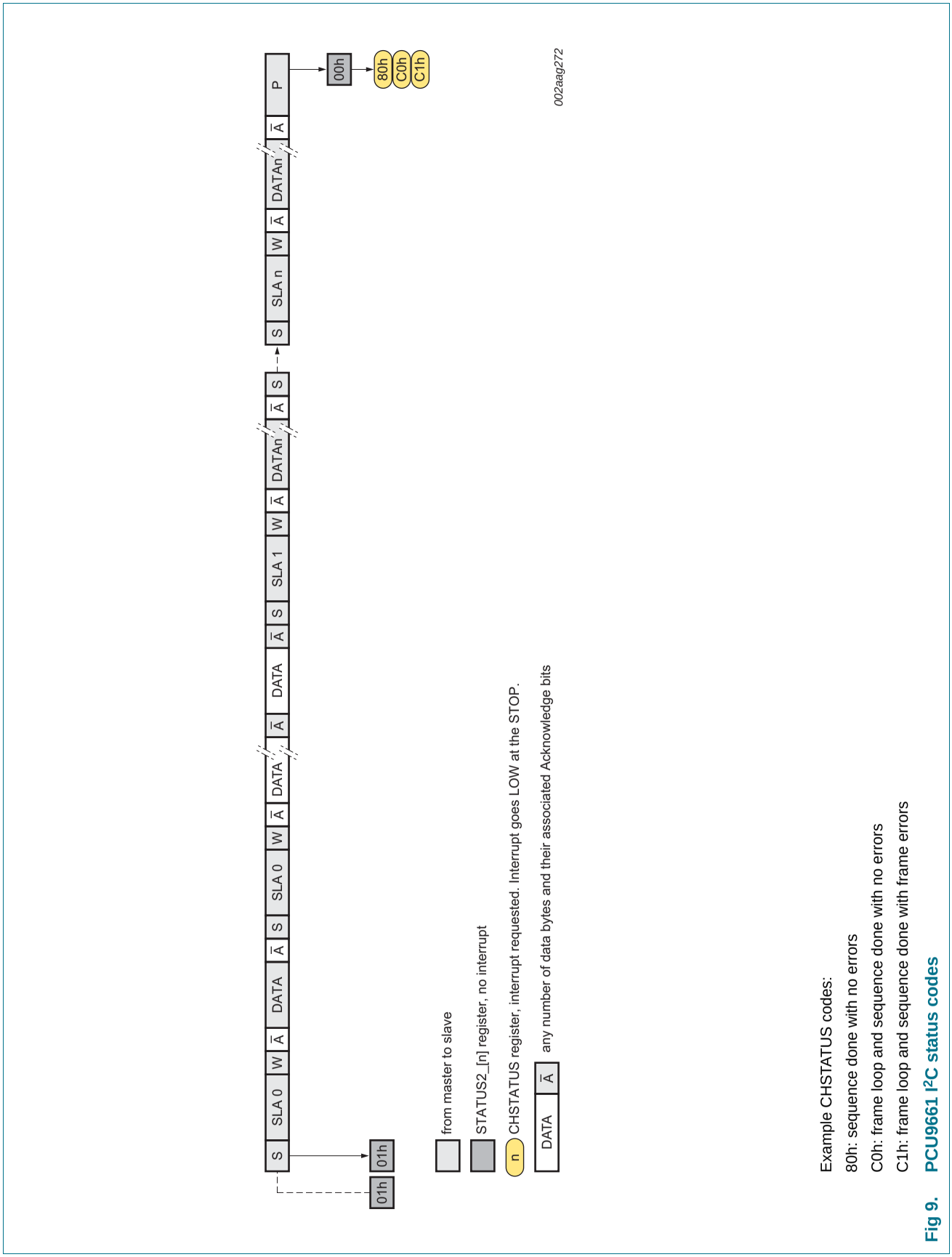
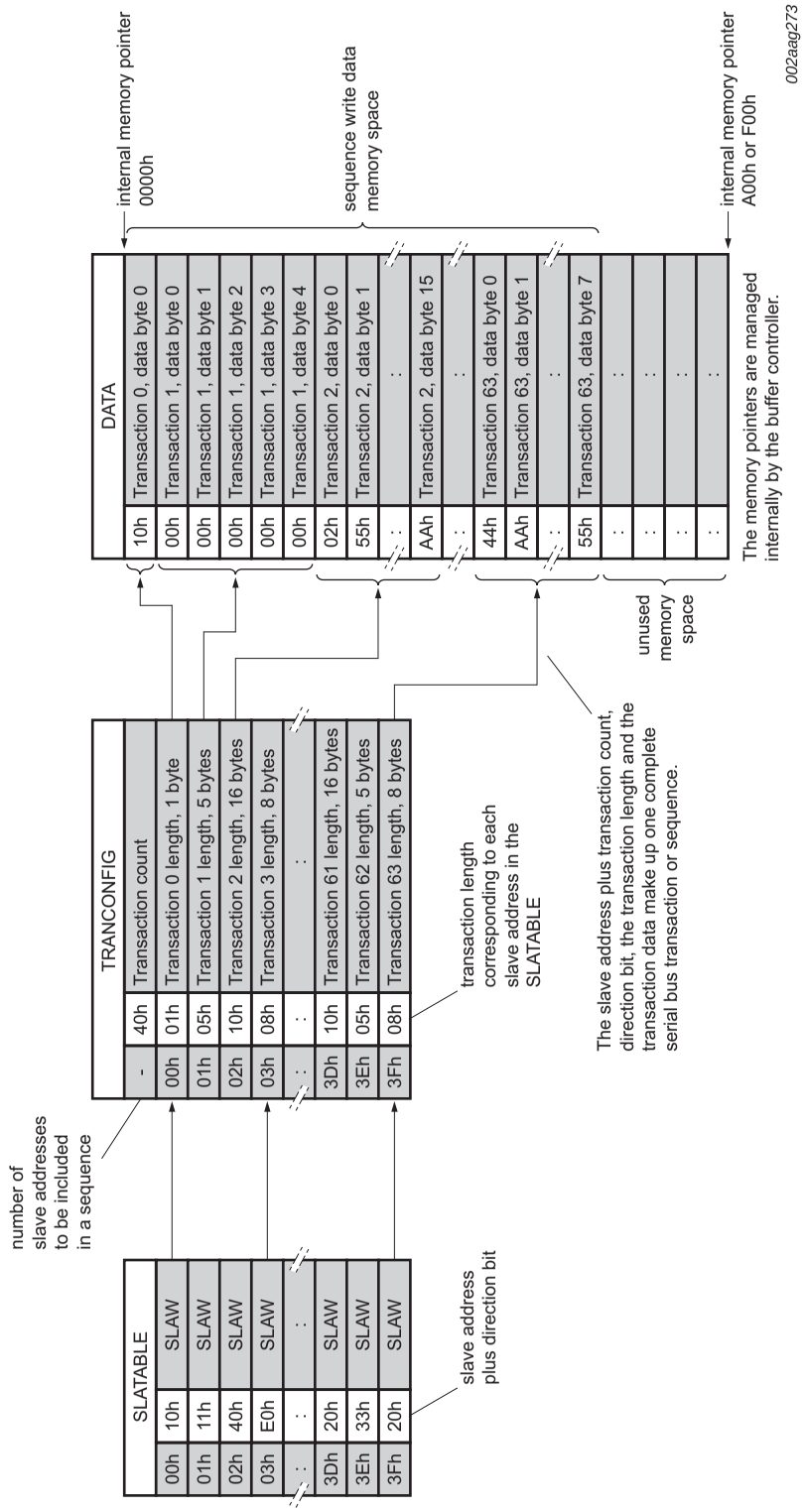


Fig 9. PCU9661 I²C status codes



Status and configuration registers are not shown.

Shaded areas are comments/indexes that are not user-accessible.

Fig 10. PCU9661 sequence block diagram; sample sequence loaded

8.2 Stopping a sequence

If the host needs to stop the execution of a sequence, it should set the STO bit in the CONTROL register. For write transactions, the host will issue a STOP after the acknowledge cycle of the current byte being transferred on the serial bus. No interrupts will be generated and all the status registers will be up to date. The Sequence Done bit (SD) will be set to indicate to the host that the STOP condition was completed and the bus is idle. The Sequence Done and the Frame Loop Done will be set if the channel is in Loop mode (FRAMECNT $\neq$ 1) and a STO or STOSEQ bit is set.

If the host issues a STOP (by setting the STO) in the middle of a sequence followed by a START (by setting the STA), then the controller will re-send the sequence from the beginning, not from the point where the sequence was last stopped.

8.3 Looping a sequence

A sequence can be set to automatically loop several times using the FRAMECNT and one of the following:

- The REFRATE register. The REFRATE register contains the value of the refresh rate which is timing required between the START of two sequences. The refresh rate is derived from the internal clock of the bus controller. If the REFRATE is programmed to 00h, the sequences will be looped back-to-back.
- Trigger enable (TE) bit. When TE is set, the refresh rate is controlled by the external trigger input and the contents of the REFRATE registers is ignored. There is no maximum timing requirement for the trigger interval.

The FRAMECNT register sets the number of times the sequence will be repeated. A frame is defined as a sequence associated with its respective refresh rate. As described above, the frame refresh rate is determined by the REFRATE register or an external trigger source.

During looping, there is no host intervention required and all status and error reporting remains active. The SD (Sequence Done) bit can be masked to avoid getting interrupted each time a frame is completed while the other error reporting bits remain unmasked. In this manner, normal transactions can run without host intervention and errors will be reported at the STOP of the current byte where the error occurred.

Once the FRAMECNT values is reached, the FLD bit in the CHSTATUS register is set and no further transactions will be executed and the channel will go to the idle state. The FLD interrupt can be masked with the FLDMSK bit in the CTRLINTMSK register. The host can poll the CTRLSTATUS register to check if the channel is active (looping) or if it is idle.

For indefinite or long term looping the host can do the following:

1. A sequence can be set to loop indefinitely by setting the FRAMECNT register to 00h. Each frame will be sent out following the REFRATE settings or the Trigger input if the TE bit is set. To end the Loop mode, the host sets the STO or STOSEQ bits in the CONTROL register.
2. A frame will be sent out continuously and back-to-back if FRAMECNT and REFRATE are set to 00h. To end the Loop mode, the hose sets the STO or STOSEQ bits in the CONTROL register.

8.3.1 Looping with REFRATE control

When using the REFRATE register (TE bit is 0) the refresh rate timing is controlled internally. Once the STA bit is set, the START command will be immediately sent on the serial bus followed by the sequence. Thereafter, the controller will issue a START command followed by the stored sequence every time the REFRATE value is reached. It is important to program enough time in the REFRATE to allow a complete sequence to reach the Sequence Done state. If the refresh rate is not long enough, the Frame Error (FE) bit will be set and an interrupt will be generated. The FE bit is maskable, however, masking the FE bit may yield undesired results on the serial interface. If the FE bit is masked, the Loop mode will continue to operate and the FE flag will remain set. To exit the Loop mode, the STO or the STOSEQ bit should be set.

8.3.2 Looping with Trigger control

The PCU9661 has one trigger input. The trigger enable (TE) bit in the CONTROL register is used to control the use of external triggering. Once enabled, the trigger will override the contents of the REFRATE register, and will start triggering when the STA bit is set. Therefore, a significant time delay can occur between setting the STA bit and the detection of a trigger. When a trigger edge is detected, the controller will issue a START command and the stored sequence will be transferred on the serial bus. The trigger will control the timing of the frame, therefore, enough time should be allowed by the trigger to allow the sequence to reach the Sequence Done state.

If a trigger edge is detected while a sequence is actively being transmitted on the bus, the Frame Error (FE) bit will be set and an interrupt will be generated. The FE bit is maskable, however, masking the FE bit may yield undesired results on the serial interface. If the FE bit is masked, the Loop mode will continue to operate and the FE flag will remain set. The polarity of the trigger edge detect is controlled by the TP bit in the CONTROL register. To exit the Trigger mode, the STO or the STOSEQ bit should be set.

8.4 Power-on reset

When power is applied to V_{DD}, an internal Power-On Reset holds the PCU9661 in a reset condition until V_{DD} has reached V_{POR}. At this point, the reset condition is released and the PCU9661 goes to the power-up initialization phase where the following operations are performed:

1. The oscillator and PLL will be re-initialized.
2. Internal register initialization is performed.
3. The memory space will be zeroed out.

The complete power-up initialization phase takes t_{rst} to be performed. During this time, writes to the PCU9661 through the parallel port are ignored. However, the parallel port can be read. This allows the device connected to the parallel port of the PCU9661 to poll the CTRLRDY register.

8.5 Global reset

Reset of the PCU9661 to its default state can be performed in 2 different ways:

- By holding the $\overline{\text{RESET}}$ pin LOW for a minimum of $t_{w(rst)}$.
- By using the Parallel Software Reset sequence as described in Figure 11. The host must write to the CTRLPRESET register of the target channel in two successive parallel bus writes to the bus controller. The first byte is A5h and the second byte is 5Ah.

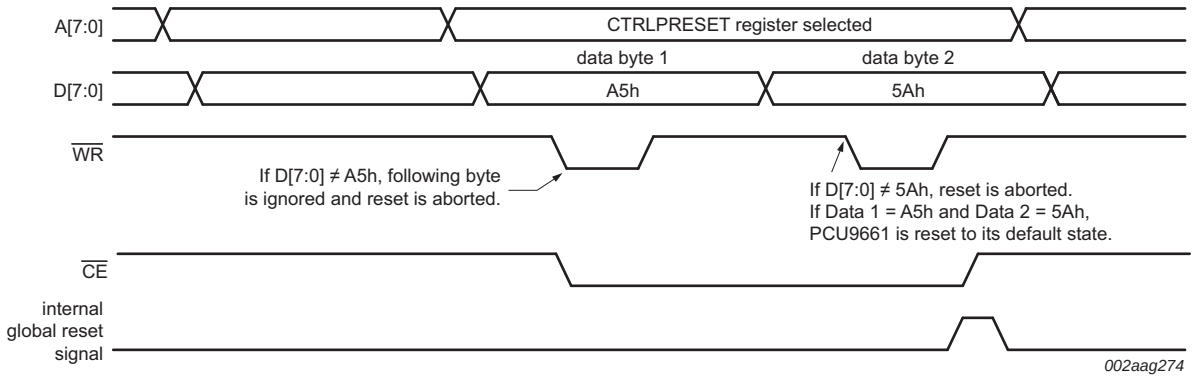


Fig 11. Parallel Software Reset sequence

The $\overline{\text{RESET}}$ hardware pin and the global software reset function behave the same as the power-on reset. A complete power-up initialization phase will be performed as defined in Section 8.4. The $\overline{\text{RESET}}$ pin has an internal pull-up resistor (through a series diode) to guarantee proper operation of the device. This pin should not be left floating and should always be driven.

8.6 Channel reset

In addition to the above chip reset options, each channel can be individually reset by programming the PRESET register for that channel as described in Figure 12. The channel will reset to its default power-up state. The host must write to the PRESET register of the target channel in two successive parallel bus writes to the bus controller. The first byte is A5h and the second byte is 5Ah.

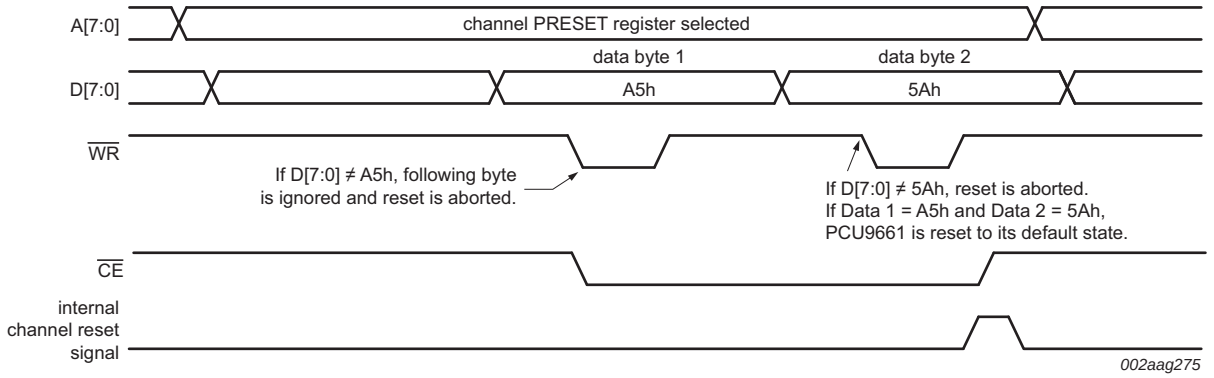
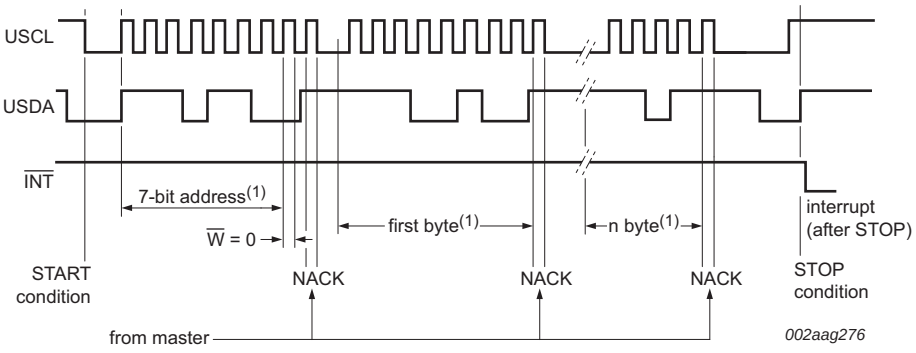


Fig 12. I²C-bus Channel Parallel Software Reset sequence

8.7 I²C-bus timing diagram

Figure 13 illustrates the typical timing diagram for the PCU9661.



PCU9661 writes data to slave.

(1) 7-bit address + $\overline{W}$ = 0 byte and number of bytes sent = value programmed in Transaction length register in TRANCONFIG register.

Fig 13. Bus timing diagram; write transactions

9. Characteristics of the I²C-bus — Ultra Fast-mode (UFm)

The PCU9661 UFm bus is a 2-wire push-pull serial bus that operates from 50 kHz to 5 MHz transmitting data in one direction. The UFm protocol is based on the I²C-bus protocol that consists of a START, slave address, command bit, 9th clock, and a STOP bit. The command bit is a 'write' only, and the data bit on the 9th clock is driven HIGH, ignoring the ACK cycle due to the unidirectional nature of the bus. The 2-wire pull-pull drivers consists of a UFm clock (USCL) and data (USDA), requiring external series resistors to allow proper line termination. The UFm bus is designed to be used in high performance single master multi-drop applications.

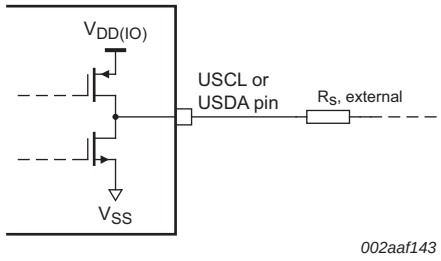


Fig 14. Simplified schematic of USCL, USDA outputs

The external resistors are chosen based upon the characteristic impedance of the UFm bus. For example, if the characteristic input impedance of the line is 175 Ω, a series resistance of 175 Ω can be used. Since the output resistance of the driver is approximately 50 Ω, the value of the series resistance used would then be 125 Ω. The final value of the resistance also depends upon the electrical length of the bus and the signal settling time required to meet the UFm timing characteristics. Larger values result in longer time for the signal to settle to its final valid value. Lower values can result in overshoot and ringing on the bus. Careful consideration must be made in designing the I²C-bus routing and selecting the series resistance.

9.1 Bit transfer

One data bit is transferred during each clock pulse. The data on the USDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals (see Figure 15).

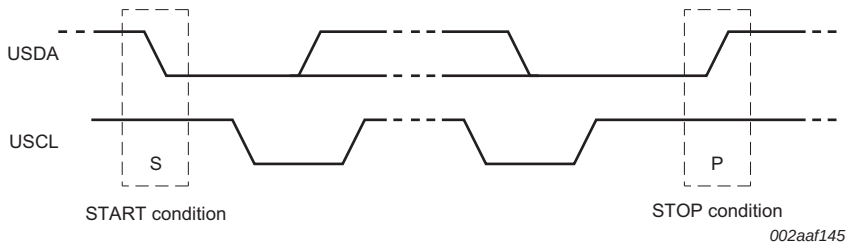


Fig 15. UFm I²C-bus bit transfer

9.2 START and STOP conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line while the clock is HIGH is defined as the START condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the STOP condition (P) (see Figure 16).

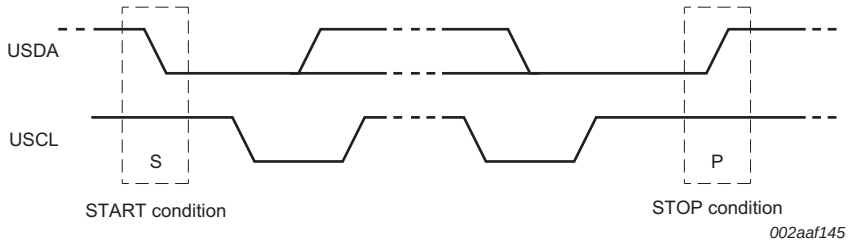


Fig 16. Definition of START and STOP conditions for UFm I²C-bus

9.3 Acknowledge (9th clock)

The UFm bus functions as a transmitter only (unidirectional). The number of data bytes transferred between the START and the STOP conditions from transmitter to receiver is not limited. Each byte of eight bits of real data is followed by a dummy bit which is a HIGH level put on the bus by the transmitter, which also generates an associated clock pulse. Since the UFm bus is unidirectional, a slave receiver shall not generate an acknowledge pulse. The slave USCL and USDA pins are input only.

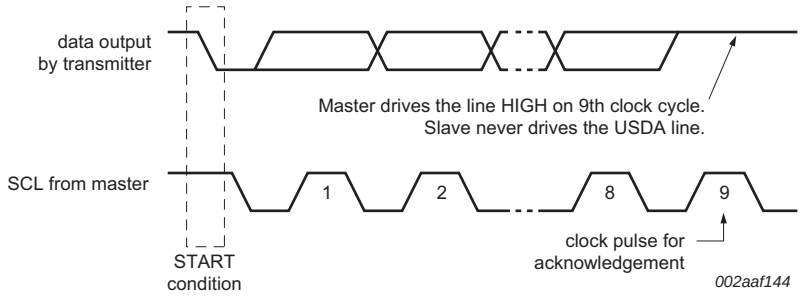


Fig 17. Acknowledgement on the UFm I²C-bus

10. JTAG port

The PCU9661 has a JTAG IEEE 1149.1 compliant port. All signals (TDI, TMS, TCK, $\overline{\text{TRST}}$ and TDO) are accessible. Only EXTEST functions are enabled, for example to conduct board-level continuity tests. Device debug/emulation functionality such as INTEST commands are not supported. The JTAG port is used for boundary scan testing (i.e., opens/shorts) during PCB manufacturing.

The following EXTEST JTAG instructions are supported:

- BYPASS
- EXTEST
- IDCODE
- SAMPLE
- PRELOAD
- CLAMP
- HIGHZ

If the JTAG boundary scan is not being used, then the JTAG pins **must** be held in the following states:

- TDI, TCK, TMS: V_{DD}
- $\overline{\text{TRST}}$: V_{SS}

11. Application design-in information

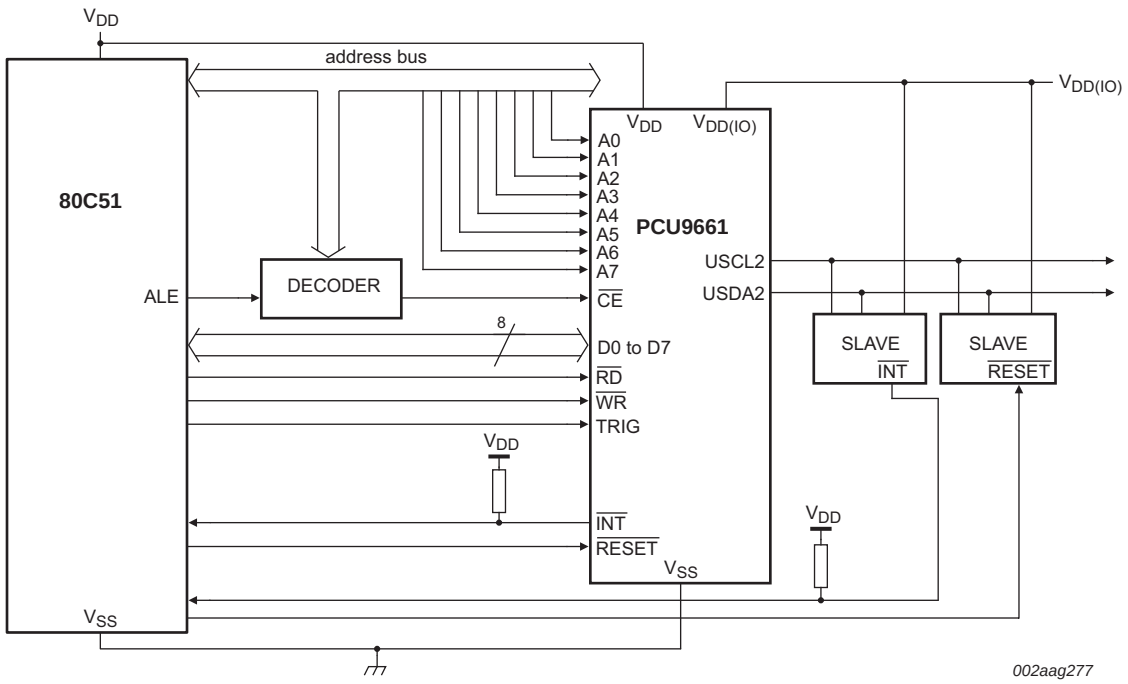


Fig 18. Application diagram using the 80C51

11.1 Specific applications

The PCU9661 is a parallel bus to UFm I²C-bus controller that is designed to allow 'smart' devices to interface with UFm I²C-bus components, where the 'smart' device does not have an integrated UFm I²C-bus port and the designer does not want to 'bit-bang' the UFm I²C-bus port. The PCU9661 can also be used to add more UFm I²C-bus ports to 'smart' devices, provide a higher frequency serial bus, and avoid running multiple traces across the printed-circuit board.

11.2 Add UFm I²C-bus port

As shown in Figure 19, the PCU9661 converts 8-bits of parallel data into a single master capable I²C-bus port for microcontrollers, microprocessors, custom ASICs, DSPs, etc., that need to interface with UFm I²C-bus components.

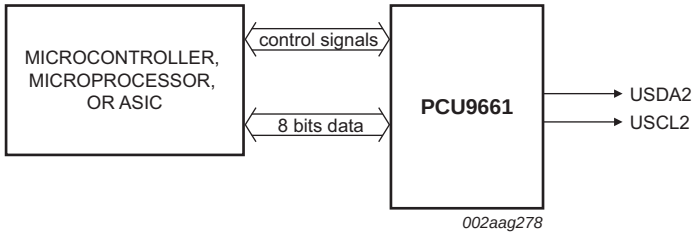


Fig 19. Adding UFm I²C-bus port application

11.3 Add additional UFm I²C-bus ports

The PCU9661 can be used to convert 8-bit parallel data into additional single master capable I²C-bus port as shown in Figure 20. It is used if the microcontroller, microprocessor, custom ASIC, DSP, etc., already have an I²C-bus port but need one or more additional UFm I²C-bus ports to interface with UFm I²C-bus components.

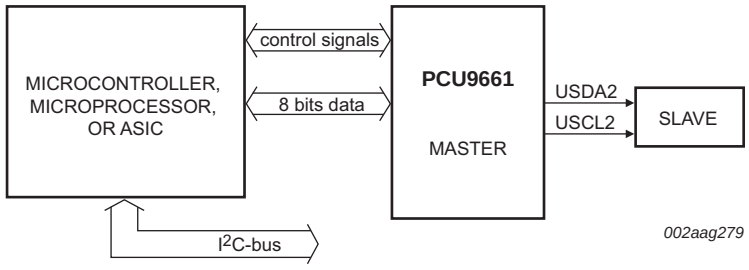


Fig 20. Adding additional UFm I²C-bus ports application

12. Limiting values

Table 30. Limiting values

In accordance with the Absolute Maximum Rating System (IEC 60134).

Symbol	Parameter	Conditions		Min	Max	Unit
V_{DD}	supply voltage			–0.3	+4.6	V
$V_{DD(IO)}$	input/output supply voltage	power supply reference for I²C-bus I/O pins		–0.3	+7.0	V
V_I	input voltage	parallel bus interface		–0.3	+4.6	V
		I²C-bus pins	[1]	–0.3	+7.0	V
I_I	input current	any input		–10	+10	mA
I_O	output current	any output		–10	+10	mA
I_{OSH}	HIGH-level short-circuit output current	I/O D0 to D7		-	106	mA
I_{OSL}	LOW-level short-circuit output current	I/O D0 to D7		-	110	mA
P_{tot}	total power dissipation			-	300	mW
P/out	power dissipation per output			-	50	mW
T_{stg}	storage temperature			–65	+150	°C
T_{amb}	ambient temperature	operating		–40	+85	°C

[1] 5.5 V steady state voltage tolerance on inputs and outputs is valid only when the supply voltage is present. 4.6 V steady state voltage tolerance on inputs and outputs when no supply voltage is present.

13. Static characteristics

Table 31. Static characteristics

V_{DD} = 3.0 V to 3.6 V; T_{amb} = –40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Supply						
V_{DD}	supply voltage	monotonic supply during power-up and power-down with a ramp time (t_{ramp}): 5 µs < t_r < 20 ns (5 % $V_{DD(min)}$ to 95 % $V_{DD(min)}$)	3.0	-	3.6	V
$V_{DD(PLL)}$	PLL supply voltage	power supply for PLL bias circuit	3.0	-	3.6	V
$V_{DD(IO)}$	input/output supply voltage	power supply reference for I²C-bus I/O pins	3.0	-	5.5	V
I_{DD}	supply current	operating mode; no load	-	15	25	mA
$I_{DD(IO)}$	input/output supply current	$V_{DD(IO)}$ = 5.5 V; V_{DD} = 3.6 V; I/O not switching	-	-	1	mA
V_{POR}	power-on reset voltage	LOW to HIGH	-	2.75	-	V
		HIGH to LOW	-	2.60	-	V

Table 31. Static characteristics *...continued*

V_{DD} = 3.0 V to 3.6 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
Inputs $\overline{WR}$, $\overline{RD}$, A0 to A7, $\overline{CE}$, TRIG							
V_{IL}	LOW-level input voltage			0	-	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		[1]	$0.7V_{DD}$	-	3.6	V
V_{hys}	hysteresis voltage			$0.1V_{DD}$	-	-	V
I_L	leakage current	input; V_I = 0 V or 3.6 V		−1	-	+1	μA
C_i	input capacitance	$V_I = V_{SS}$ or V_{DD}		-	2.0	4.5	pF
Input $\overline{RESET}$							
V_{IL}	LOW-level input voltage			0	-	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		[1]	$0.7V_{DD}$	-	3.6	V
V_{hys}	hysteresis voltage			$0.1V_{DD}$	-	-	V
I_L	leakage current	input; V_I = 0 V or 3.6 V		−1	-	+75	μA
C_i	input capacitance	$V_I = V_{SS}$ or V_{DD}		-	2.0	4.5	pF
Inputs/outputs D0 to D7							
V_{IL}	LOW-level input voltage			0	-	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage			$0.7V_{DD}$	-	3.6	V
I_{OH}	HIGH-level output current	$V_{OH} = V_{DD(IO)} - 0.4$ V		3.2	-	-	mA
I_{OL}	LOW-level output current	V_{OL} = 0.4 V		2.0	-	-	mA
I_L	leakage current	input; V_I = 0 V or 5.5 V		−1	-	+1	μA
C_{io}	input/output capacitance	$V_I = V_{SS}$ or V_{DD}		-	2.8	5	pF
USDA2 and USCL2							
I_{OL}	LOW-level output current	V_{OL} = 0.4 V		5	-	-	mA
I_{OH}	HIGH-level output current	$V_{OH} = V_{DD(IO)} - 0.4$ V		4.8	-	-	mA
C_{io}	input/output capacitance	$V_I = V_{SS}$ or $V_{DD(IO)}$		-	5.6	7	pF
R_{ON}	ON resistance			-	50	-	Ω
I_L	leakage current	V_{DD} = 3.6 V		−1	-	+1	μA
		V_{DD} = 5.5 V		−10	-	+10	μA
Output $\overline{INT}$							
I_{OL}	LOW-level output current	V_{OL} = 0.4 V		6.0	-	-	mA
I_L	leakage current	V_O = 0 V or 3.6 V		−1	-	+75	μA
C_o	output capacitance	$V_I = V_{SS}$ or V_{DD}		-	3.8	5.5	pF

[1] 5.5 V steady state voltage tolerance on inputs and outputs is valid only when the supply voltage is present. 4.6 V steady state voltage tolerance on inputs and outputs when no supply voltage is present.

14. Dynamic characteristics

Table 32. Dynamic characteristics (3.3 volt)[1][2][3]

V_{DD} = 3.3 V ± 0.3 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
Initialization timing							
$t_{init(po)}$	power-on initialization time	$V_{DD} \geq 3.0$ V		-	-	650	µs
t_{init}	initialization time	channel initialization time from Channel Software Reset		-	-	70	µs
		controller initialization time from POR, $\overline{RESET}$, or Global Software Reset inactive		-	-	650	µs
$\overline{RESET}$ timing							
$t_{w(rst)}$	reset pulse width			4	-	-	µs
t_{rst}	reset time		[4][5]	1.5	-	-	µs
$\overline{INT}$ timing							
$t_{as(int)}$	interrupt assert time			-	-	500	ns
$t_{das(int)}$	interrupt de-assert time			-	-	100	ns
TRIG timing							
$t_{w(trig)}$	trigger pulse width	HIGH or LOW		100	-	-	ns
Bus timing (see Figure 21 and Figure 23)							
$t_{su(A)}$	address set-up time	to $\overline{RD}$, $\overline{WR}$ LOW		0	-	-	ns
$t_{h(A)}$	address hold time	from $\overline{RD}$, $\overline{WR}$ LOW		14	-	-	ns
$t_{su(CE\_N)}$	$\overline{CE}$ set-up time	to $\overline{RD}$, $\overline{WR}$ LOW		0	-	-	ns
$t_{h(CE\_N)}$	$\overline{CE}$ hold time	from $\overline{RD}$, $\overline{WR}$ LOW		0	-	-	ns
$t_{w(RDL)}$	$\overline{RD}$ LOW pulse width			40	-	-	ns
$t_{w(WRL)}$	$\overline{WR}$ LOW pulse width			40	-	-	ns
$t_{d(DV)}$	data valid delay time	after $\overline{RD}$ and $\overline{CE}$ LOW		-	-	45	ns
$t_{d(QZ)}$	data output float delay time	after $\overline{RD}$ or $\overline{CE}$ HIGH		-	-	7	ns
$t_{su(Q)}$	data output set-up time	before $\overline{WR}$ HIGH		5	-	-	ns
$t_{h(Q)}$	data output hold time	after $\overline{WR}$ HIGH		2	-	-	ns
$t_{w(RDH)}$	$\overline{RD}$ HIGH pulse width			40	-	-	ns
$t_{w(WRH)}$	$\overline{WR}$ HIGH pulse width			40	-	-	ns

[1] Parameters are valid over specified temperature and voltage range.

[2] All voltage measurements are referenced to ground (V_{SS}). For testing, all inputs swing between 0 V and 3.0 V with a transition time of 5 ns maximum. All time measurements are referenced at input voltages of 1.5 V and output voltages shown in Figure 21 and Figure 23.

[3] Test conditions for outputs: C_L = 50 pF; R_L = 500 Ω, except open-drain outputs.
Test conditions for open-drain outputs: C_L = 50 pF; R_L = 1 kΩ pull-up to V_{DD}.

[4] Resetting the device while actively communicating on the bus may cause glitches or an errant STOP condition.

[5] Upon reset, the full delay will be the sum of t_{rst} and the RC time constant of the SDA and SCL bus.

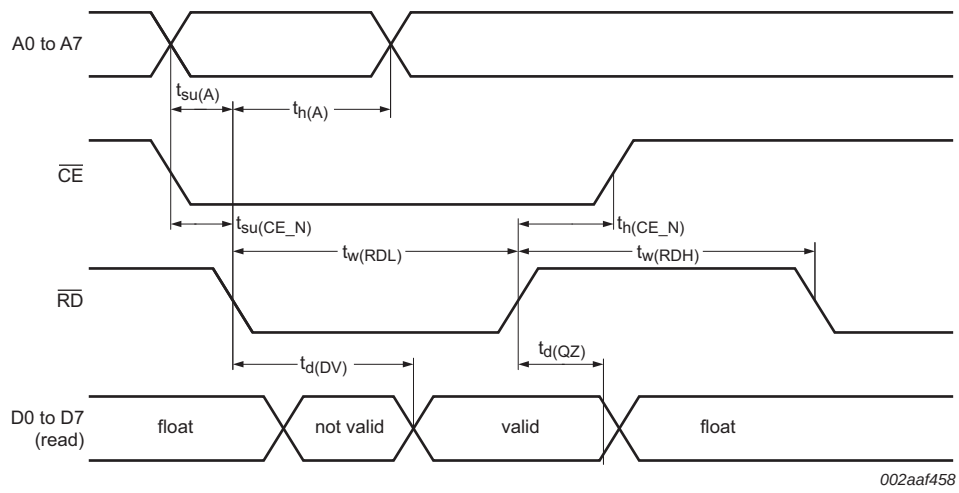


Fig 21. Bus timing (read cycle)

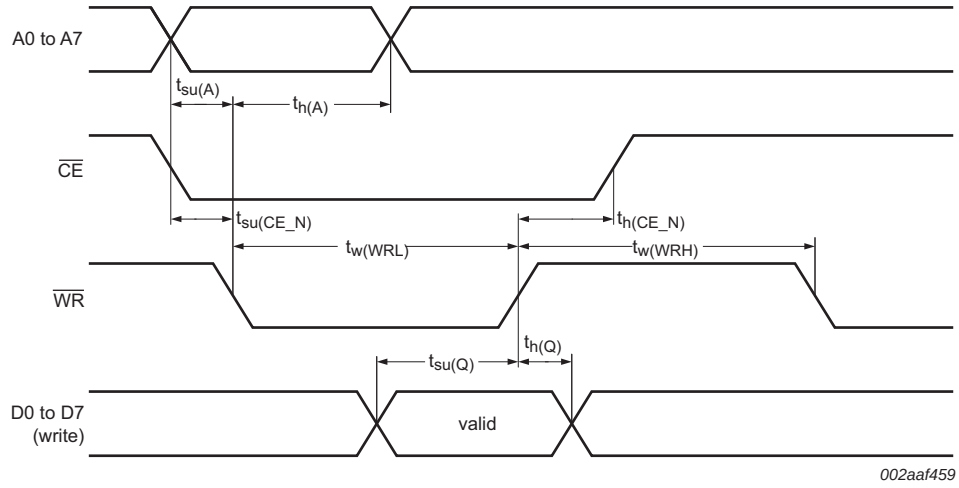
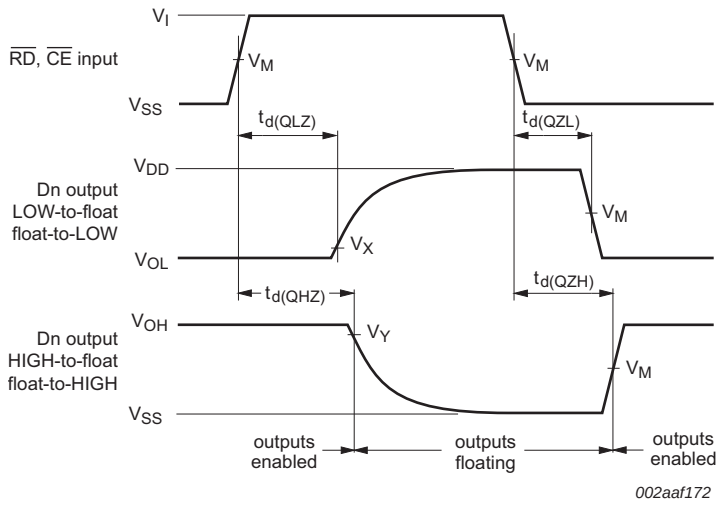


Fig 22. Parallel bus timing (write cycle)



V_M = 1.5 V

$V_X = V_{OL}$ + 0.2 V

$V_Y = V_{OH}$ − 0.2 V

V_{OL} and V_{OH} are typical output voltage drops that occur with the output load.

Fig 23. Data timing

Table 33. UFm I²C-bus frequency and timing specifications

All the timing limits are valid within the operating supply voltage and ambient temperature range; V_{DD} = 2.5 V ± 0.2 V and 3.3 V ± 0.3 V; T_{amb} = −40 °C to +85 °C; and refer to V_{IL} and V_{IH} with an input voltage of V_{SS} to V_{DD}.

Symbol	Parameter	Conditions	Ultra Fast-mode I²C-bus		Unit
			Min	Max	
f_{USCL}	USCL clock frequency		0	5000	kHz
t_{BUF}	bus free time between a STOP and START condition		0.08	-	µs
$t_{HD;STA}$	hold time (repeated) START condition		0.05	-	µs
$t_{SU;STA}$	set-up time for a repeated START condition		0.05	-	µs
$t_{SU;STO}$	set-up time for STOP condition		0.05	-	µs
$t_{HD;DAT}$	data hold time		10	-	ns
$t_{VD;DAT}$	data valid time	[1]	10	-	ns
$t_{SU;DAT}$	data set-up time		30	-	ns
t_{LOW}	LOW period of the USCL clock		0.05	-	µs
t_{HIGH}	HIGH period of the USCL clock		0.05	-	µs
t_f	fall time of both USDA and USCL signals		-[2]	50	ns
t_r	rise time of both USDA and USCL signals		-[2]	50	ns

[1] $t_{VD;DAT}$ = minimum time for SDA data out to be valid following SCL LOW.

[2] Typical rise/fall times for UFm signals is 25 ns measured from the 20 % level to the 80 % (rise time) or from the 80 % level to the 20 % level (fall time).

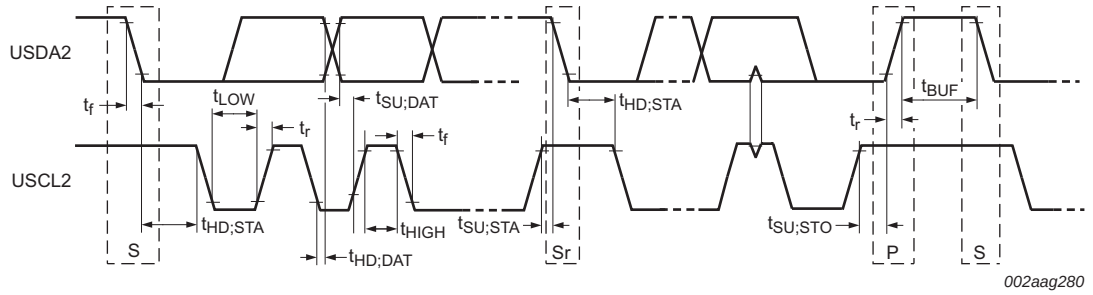
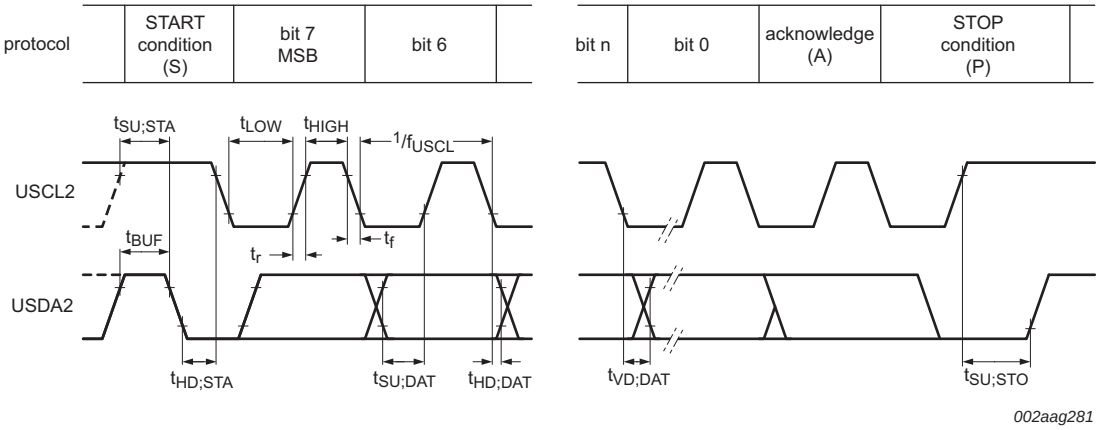


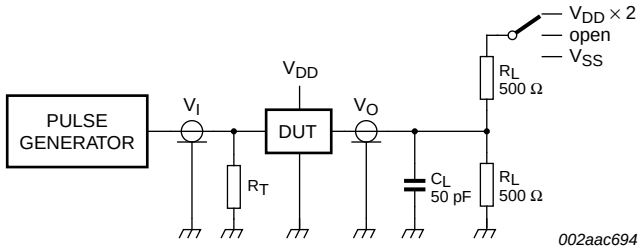
Fig 24. Definition of timing on the UFm I²C-bus



Rise and fall times refer to V_{IL} and V_{IH}.

Fig 25. UFm I²C-bus timing diagram

15. Test information



Test data are given in Table 34.

R_L = load resistance.

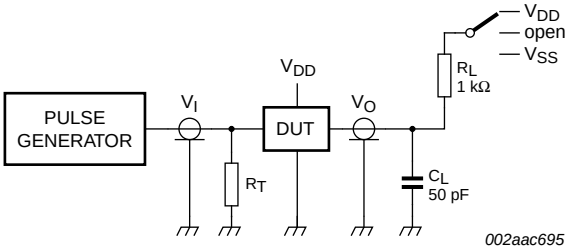
C_L = load capacitance includes jig and probe capacitance.

R_T = termination resistance should be equal to the output impedance Z_O of the pulse generators.

Fig 26. Test circuitry for switching times

Table 34. Test data

Test	Conditions	Load		S1
		C_L	R_L	
$t_{d(DV)}$, $t_{d(QZ)}$	Dn outputs active LOW	50 pF	500 Ω	$V_{DD} \times 2$
	Dn outputs active HIGH	50 pF	500 Ω	open



Test data are given in Table 35.

R_L = load resistance.

C_L = load capacitance includes jig and probe capacitance.

R_T = termination resistance should be equal to the output impedance Z_O of the pulse generators.

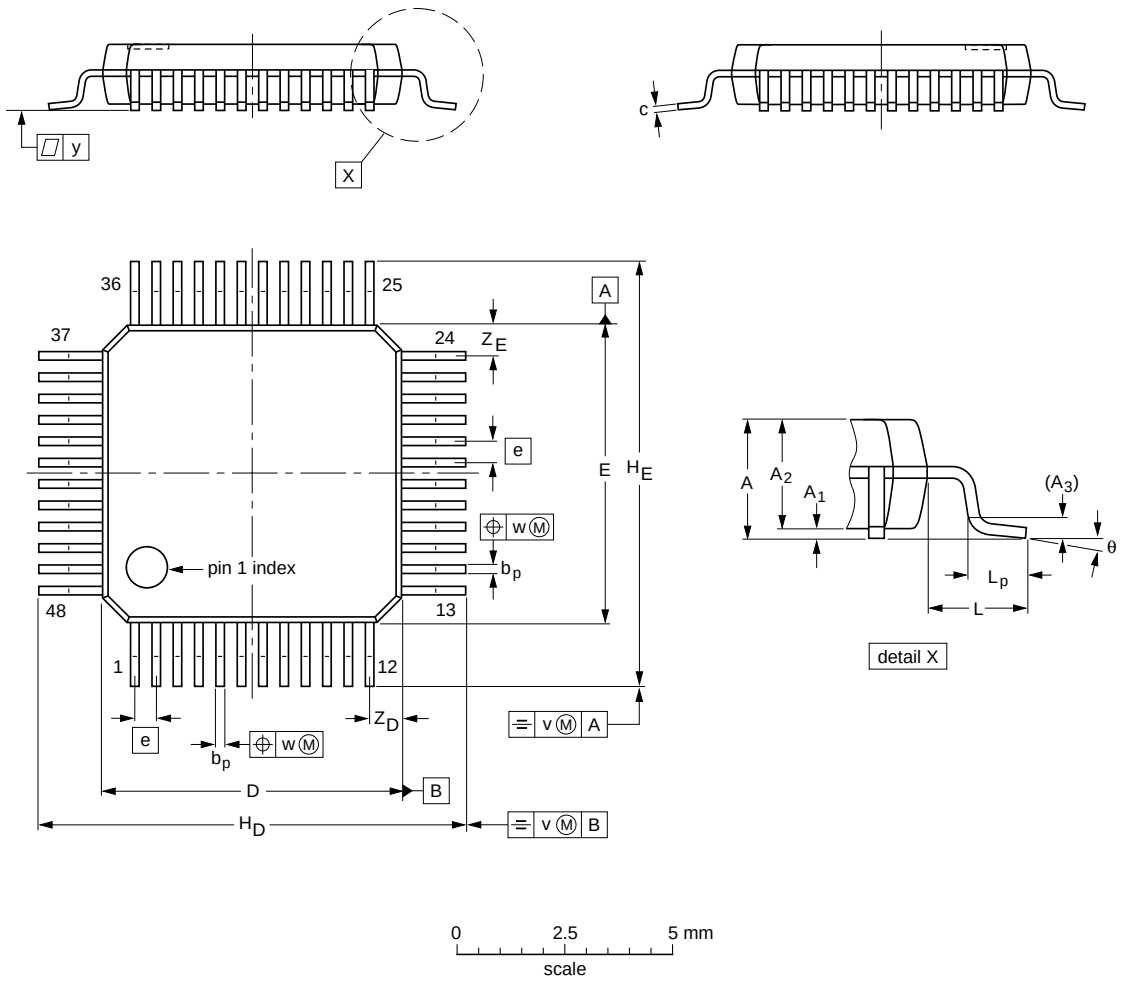
Fig 27. Test circuitry for open-drain switching times

Table 35. Test data

Test	Load		S1
	C_L	R_L	
$t_{as(int)}$	50 pF	1 kΩ	V_{DD}
$t_{das(int)}$	50 pF	1 kΩ	V_{DD}

16. Package outline

LQFP48: plastic low profile quad flat package; 48 leads; body 7 x 7 x 1.4 mm **SOT313-2**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_D	H_E	L	L_p	v	w	y	$Z_D^{(1)}$	$Z_E^{(1)}$	θ
mm	1.6	0.20 0.05	1.45 1.35	0.25	0.27 0.17	0.18 0.12	7.1 6.9	7.1 6.9	0.5	9.15 8.85	9.15 8.85	1	0.75 0.45	0.2	0.12	0.1	0.95 0.55	0.95 0.55	7^o 0^o

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT313-2	136E05	MS-026				~~00-01-19~~ 03-02-25

Fig 28. Package outline SOT313-2 (LQFP48)

17. Handling information

All input and output pins are protected against ElectroStatic Discharge (ESD) under normal handling. When handling ensure that the appropriate precautions are taken as described in *JESD625-A* or equivalent standards.

18. Soldering of SMD packages

This text provides a very brief insight into a complex technology. A more in-depth account of soldering ICs can be found in Application Note *AN10365 "Surface mount reflow soldering description"*.

18.1 Introduction to soldering

Soldering is one of the most common methods through which packages are attached to Printed Circuit Boards (PCBs), to form electrical circuits. The soldered joint provides both the mechanical and the electrical connection. There is no single soldering method that is ideal for all IC packages. Wave soldering is often preferred when through-hole and Surface Mount Devices (SMDs) are mixed on one printed wiring board; however, it is not suitable for fine pitch SMDs. Reflow soldering is ideal for the small pitches and high densities that come with increased miniaturization.

18.2 Wave and reflow soldering

Wave soldering is a joining technology in which the joints are made by solder coming from a standing wave of liquid solder. The wave soldering process is suitable for the following:

- Through-hole components
- Leaded or leadless SMDs, which are glued to the surface of the printed circuit board

Not all SMDs can be wave soldered. Packages with solder balls, and some leadless packages which have solder lands underneath the body, cannot be wave soldered. Also, leaded SMDs with leads having a pitch smaller than ~0.6 mm cannot be wave soldered, due to an increased probability of bridging.

The reflow soldering process involves applying solder paste to a board, followed by component placement and exposure to a temperature profile. Leaded packages, packages with solder balls, and leadless packages are all reflow solderable.

Key characteristics in both wave and reflow soldering are:

- Board specifications, including the board finish, solder masks and vias
- Package footprints, including solder thieves and orientation
- The moisture sensitivity level of the packages
- Package placement
- Inspection and repair
- Lead-free soldering versus SnPb soldering

18.3 Wave soldering

Key characteristics in wave soldering are:

- Process issues, such as application of adhesive and flux, clinching of leads, board transport, the solder wave parameters, and the time during which components are exposed to the wave
- Solder bath specifications, including temperature and impurities

18.4 Reflow soldering

Key characteristics in reflow soldering are:

- Lead-free versus SnPb soldering; note that a lead-free reflow process usually leads to higher minimum peak temperatures (see Figure 29) than a SnPb process, thus reducing the process window
- Solder paste printing issues including smearing, release, and adjusting the process window for a mix of large and small components on one board
- Reflow temperature profile; this profile includes preheat, reflow (in which the board is heated to the peak temperature) and cooling down. It is imperative that the peak temperature is high enough for the solder to make reliable solder joints (a solder paste characteristic). In addition, the peak temperature must be low enough that the packages and/or boards are not damaged. The peak temperature of the package depends on package thickness and volume and is classified in accordance with Table 36 and 37

Table 36. SnPb eutectic process (from J-STD-020C)

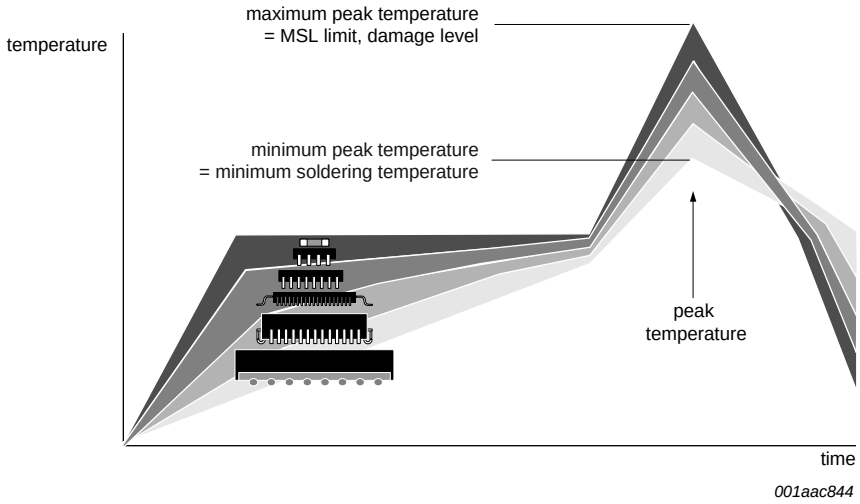
Package thickness (mm)	Package reflow temperature (°C)	
	Volume (mm³)	
	< 350	≥ 350
< 2.5	235	220
≥ 2.5	220	220

Table 37. Lead-free process (from J-STD-020C)

Package thickness (mm)	Package reflow temperature (°C)		
	Volume (mm³)		
	< 350	350 to 2000	> 2000
< 1.6	260	260	260
1.6 to 2.5	260	250	245
> 2.5	250	245	245

Moisture sensitivity precautions, as indicated on the packing, must be respected at all times.

Studies have shown that small packages reach higher temperatures during reflow soldering, see Figure 29.



MSL: Moisture Sensitivity Level

Fig 29. Temperature profiles for large and small components

For further information on temperature profiles, refer to Application Note *AN10365 "Surface mount reflow soldering description"*.

19. Abbreviations

Table 38. Abbreviations

Acronym	Description
ASIC	Application Specific Integrated Circuit
CDM	Charged-Device Model
CPU	Central Processing Unit
DSP	Digital Signal Processor
ESD	ElectroStatic Discharge
Fm+	Fast-mode Plus
HBM	Human Body Model
I²C-bus	Inter-Integrated Circuit bus
I/O	Input/Output
LED	Light Emitting Diode
MCU	MicroController Unit
PCB	Printed-Circuit Board
PLL	Phase-Locked Loop
SMBus	System Management Bus
UFm	Ultra Fast-mode

20. Revision history

Table 39. Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
PCU9661 v.1	20110912	Product data sheet	-	-

21. Legal information

21.1 Data sheet status

Document status[1][2]	Product status[3]	Definition
Objective [short] data sheet	Development	This document contains data from the objective specification for product development.
Preliminary [short] data sheet	Qualification	This document contains data from the preliminary specification.
Product [short] data sheet	Production	This document contains the product specification.

[1] Please consult the most recently issued document before initiating or completing a design.

[2] The term 'short data sheet' is explained in section "Definitions".

[3] The product status of device(s) described in this document may have changed since this document was published and may differ in case of multiple devices. The latest product status information is available on the Internet at URL http://www.nxp.com.

21.2 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

Short data sheet — A short data sheet is an extract from a full data sheet with the same product type number(s) and title. A short data sheet is intended for quick reference only and should not be relied upon to contain detailed and full information. For detailed and full information see the relevant full data sheet, which is available on request via the local NXP Semiconductors sales office. In case of any inconsistency or conflict with the short data sheet, the full data sheet shall prevail.

Product specification — The information and data provided in a Product data sheet shall define the specification of the product as agreed between NXP Semiconductors and its customer, unless NXP Semiconductors and customer have explicitly agreed otherwise in writing. In no event however, shall an agreement be valid in which the NXP Semiconductors product is deemed to offer functions and qualities beyond those described in the Product data sheet.

21.3 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from national authorities.

Non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

21.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

I²C-bus — logo is a trademark of NXP B.V.

22. Contact information

For more information, please visit: **http://www.nxp.com**

For sales office addresses, please send an email to: **salesaddresses@nxp.com**

23. Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 12 September 2011
Document identifier: PCU9661