Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

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

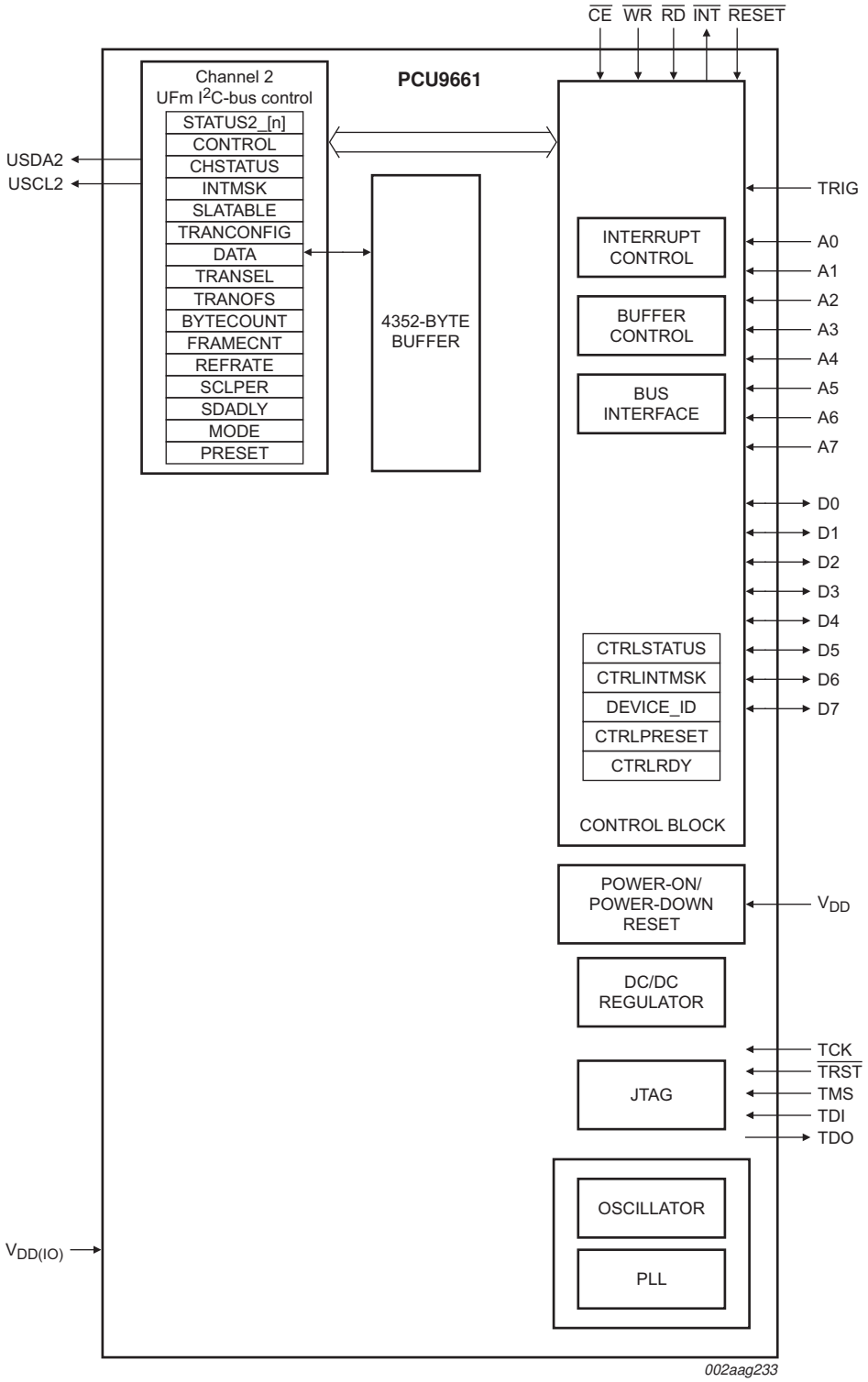


Fig 1. Block diagram

6. Pinning information

6.1 Pinning

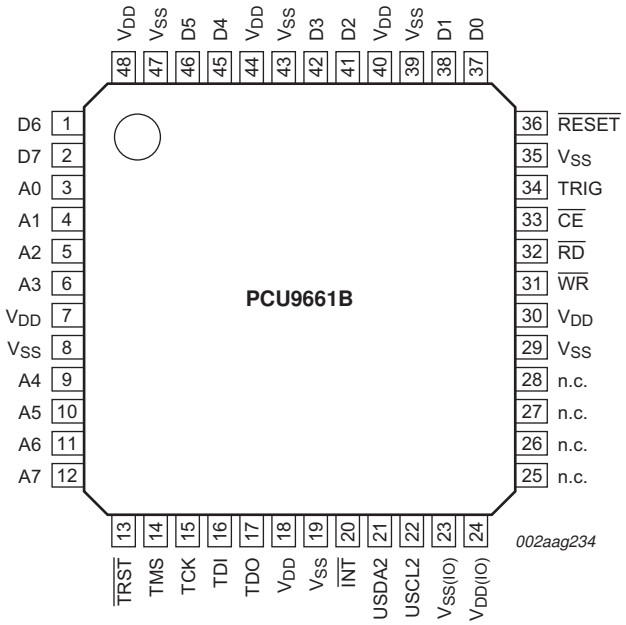


Fig 2. Pin configuration for LQFP48

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

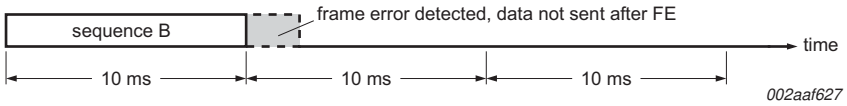
Address: Channel 2 = E1h.

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.

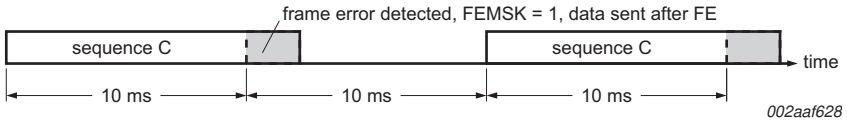
FE - Frame Error bit: This bit indicates that the time required to send the sequence exceeds the refresh rate programmed in the REFRATE register or the time between trigger ticks. Solving frame errors include programming longer refresh rates, speeding up the bus frequency, shortening the amount of bytes 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.

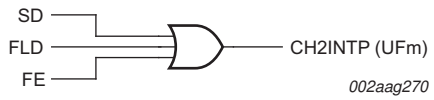


Fig 4. PCU9661 status reporting logic

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

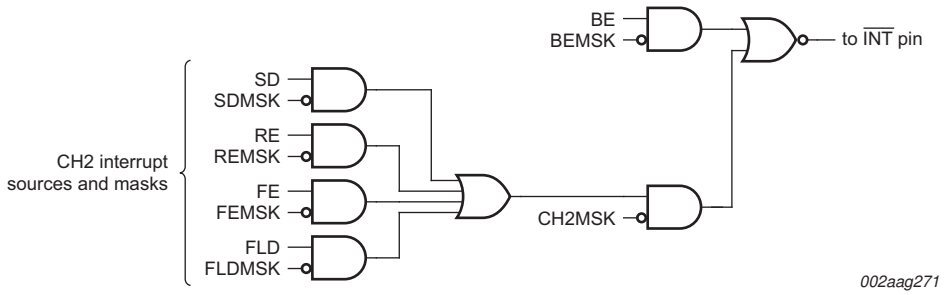


Fig 5. PCU9661 interrupt logic

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.

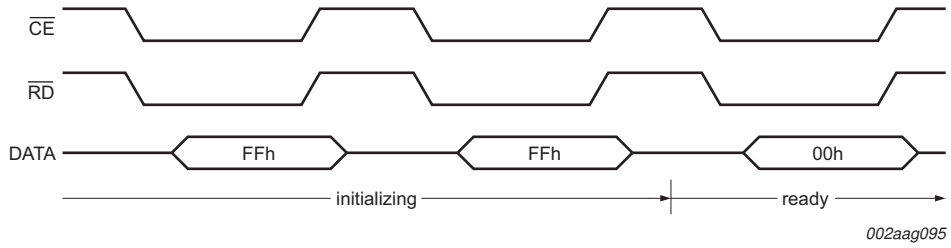


Fig 6. During initialization, $\overline{CE}$ must transition with $\overline{RD}$ at each read operation