PCA9665; PCA9665A

Fm+ parallel bus to I²C-bus controller

Rev. 4 — 29 September 2011 Product data sheet

1. General description

The PCA9665/PCA9665A serves as an interface between most standard parallel-bus microcontrollers/microprocessors and the serial I²C-bus and allows the parallel bus system to communicate bidirectionally with the I²C-bus. The PCA9665/PCA9665A can operate as a master or a slave and can be a transmitter or receiver. Communication with the I²C-bus is carried out on a Byte or Buffered mode using interrupt or polled handshake. The PCA9665/PCA9665A controls all the I²C-bus specific sequences, protocol, arbitration and timing with no external timing element required.

The PCA9665 and PCA9665A have the same footprint as the PCA9564 with additional features:

- 1 MHz transmission speeds
- Up to 25 mA drive capability on SCL/SDA
- 68-byte buffer
- I²C-bus General Call
- Software reset on the parallel bus

2. Features and benefits

- Parallel-bus to I²C-bus protocol converter and interface
- Both master and slave functions
- Multi-master capability
- Internal oscillator trimmed to 15 % accuracy reduces external components
- 1 Mbit/s and up to 25 mA SCL/SDA I_{OL} (Fast-mode Plus (Fm+)) capability
- I²C-bus General Call capability
- Software reset on parallel bus
- 68-byte data buffer
- Operating supply voltage: 2.3 V to 3.6 V
- 5 V tolerant I/Os
- Standard-mode and Fast-mode I²C-bus capable and compatible with SMBus
- PCA9665A ‘glitch-free’ restart is suitable for use with buffer drivers
- ESD protection exceeds 2000 V HBM per JESD22-A114 and 1000 V CDM per JESD22-C101
- Latch-up testing is done to JEDEC Standard JESD78 which exceeds 100 mA
- Packages offered:
  - PCA9665: SO20, TSSOP20, HVQFN20
  - PCA9665A: TSSOP20

3. Applications

- Add I²C-bus port to controllers/processors that do not have one
- Add additional 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

4. Ordering information

Table 1. Ordering information

T_{amb} = −40 °C to +85 °C.

Type number	Topside mark	Package		
		Name	Description	Version
PCA9665BS	9665	HVQFN20	plastic thermal enhanced very thin quad flat package; no leads; 20 terminals; body 5 × 5 × 0.85 mm	SOT662-1
PCA9665D	PCA9665D	SO20	plastic small outline package; 20 leads; body width 7.5 mm	SOT163-1
PCA9665PW	PCA9665	TSSOP20	plastic thin shrink small outline package; 20 leads; body width 4.4 mm	SOT360-1
PCA9665APW	CA9665A	TSSOP20	plastic thin shrink small outline package; 20 leads; body width 4.4 mm	SOT360-1

5. Block diagram

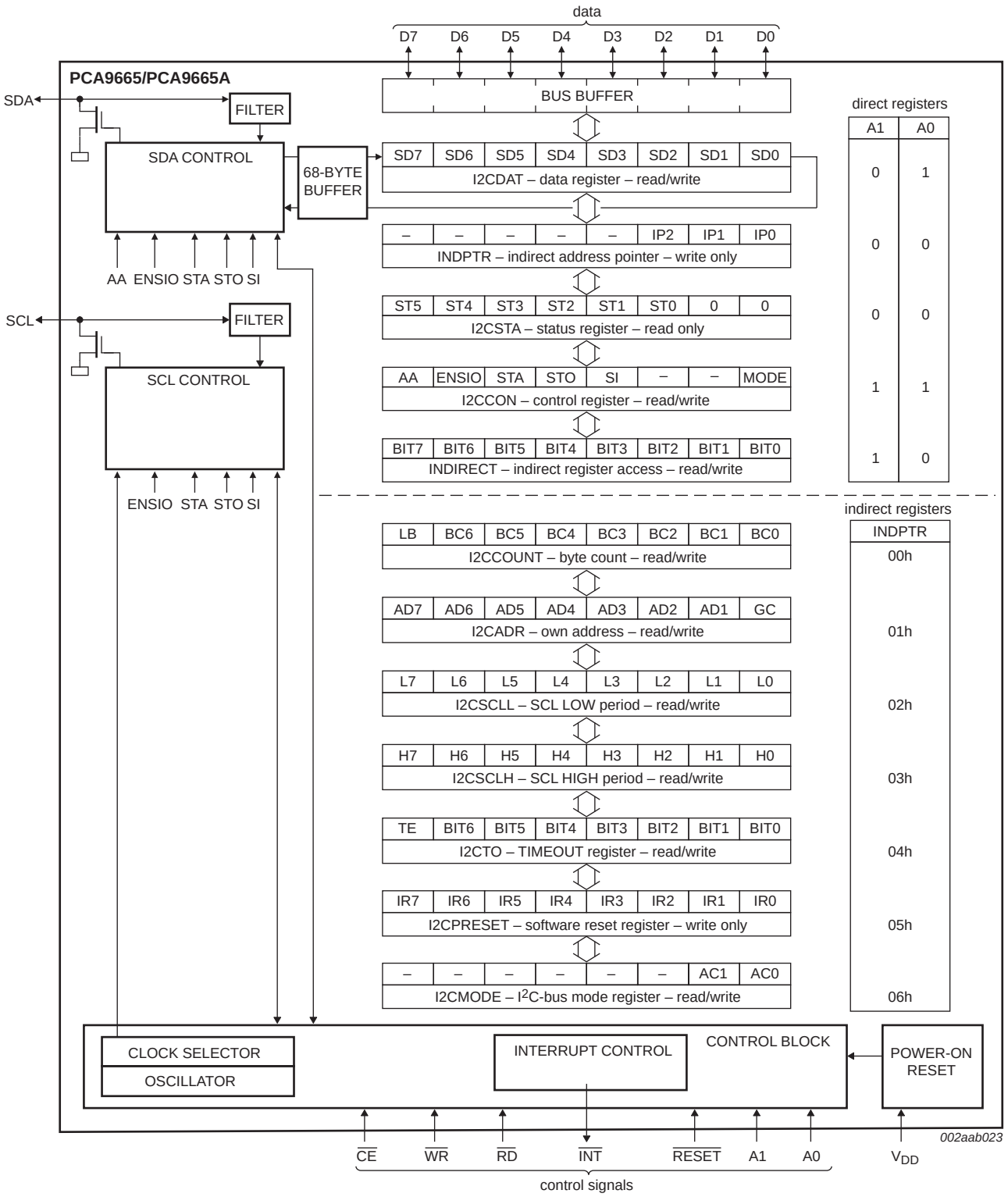


Fig 1. Block diagram of PCA9665/PCA9665A

6. Pinning information

6.1 Pinning

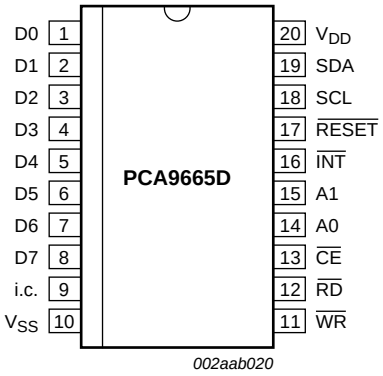


Fig 2. Pin configuration for SO20

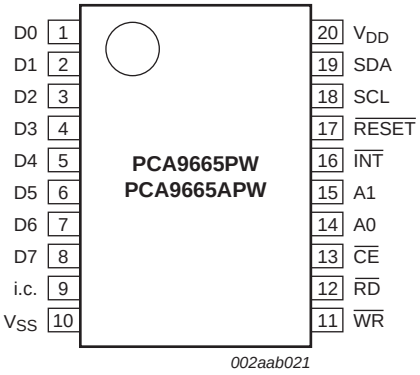
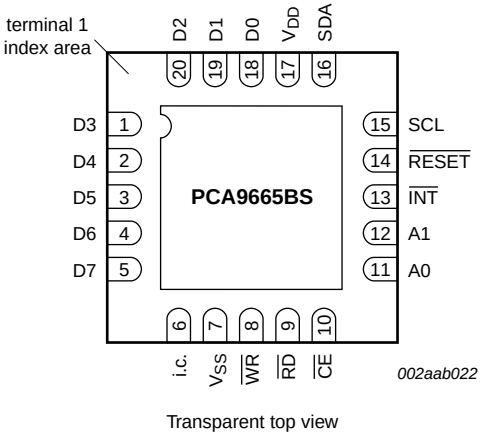


Fig 3. Pin configuration for TSSOP20



Transparent top view

Fig 4. Pin configuration for HVQFN20

6.2 Pin description

Table 2. Pin description

Symbol	Pin		Type	Description
	SO20, TSSOP20	HVQFN20		
D0	1	18	I/O	**Data bus:** Bidirectional 3-state data bus used to transfer commands, data and status between the bus controller and the CPU. D0 is the least significant bit.
D1	2	19	I/O	
D2	3	20	I/O	
D3	4	1	I/O	
D4	5	2	I/O	
D5	6	3	I/O	
D6	7	4	I/O	
D7	8	5	I/O	
i.c.	9	6	-	**internally connected:** must be left floating (pulled LOW internally)
V_{SS}	10	7[1]	power	**Supply ground**
$\overline{WR}$	11	8	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 either $\overline{WR}$ or $\overline{CE}$.
$\overline{RD}$	12	9	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 $\overline{RD}$.
$\overline{CE}$	13	10	I	**Chip Enable:** Active LOW input signal. When LOW, data transfers between the CPU and the bus controller are enabled on D0 to D7 as controlled by the $\overline{WR}$, $\overline{RD}$ and A0 to A1 inputs. When HIGH, places the D0 to D7 lines in the 3-state condition. Data are written into the addressed register on rising edge of either $\overline{CE}$ or $\overline{WR}$.
A0	14	11	I	**Address inputs:** Selects the bus controller's internal registers and ports for read/write operations.
A1	15	12	I	
$\overline{INT}$	16	13	O	**Interrupt request:** Active LOW, open-drain, output. This pin requires a pull-up device.
$\overline{RESET}$	17	14	I	**Reset:** Active LOW input. A LOW level clears internal registers and resets the I²C-bus state machine.
SCL	18	15	I/O	I²C-bus serial clock input/output (open-drain). This pin requires a pull-up device.
SDA	19	16	I/O	I²C-bus serial data input/output (open-drain). This pin requires a pull-up device.
V_{DD}	20	17	power	**Power supply:** 2.3 V to 3.6 V

[1] HVQFN20 package die supply ground is connected to both the V_{SS} pin and the exposed center pad. The V_{SS} pin must be connected to supply ground for proper device operation. For enhanced thermal, electrical, and board-level performance, the exposed pad needs to be soldered to the board using a corresponding thermal pad on the board, and for proper heat conduction through the board thermal vias need to be incorporated in the PCB in the thermal pad region.

7. Functional description

7.1 General

The PCA9665/PCA9665A acts as an interface device between standard high-speed parallel buses and the serial I²C-bus. On the I²C-bus, it can act either as a master or slave. Bidirectional data transfer between the I²C-bus and the parallel-bus microcontroller is carried out on a byte or buffered basis, using either an interrupt or polled handshake.

7.2 Internal oscillator

The PCA9665/PCA9665A contains an internal oscillator which is used for all I²C-bus timing. Typical oscillator frequency is 28.5 MHz for the PCA9665 and 32 MHz for the PCA9665A. The oscillator requires up to 550 μs to start-up after ENSIO bit is set to '1'.

7.3 Registers

The PCA9665/PCA9665A contains eleven registers which are used to configure the operation of the device as well as to send and receive serial data. There are four registers that can be accessed directly and seven registers that are accessed indirectly by setting a register pointer.

The four direct registers are selected by setting pins A0 and A1 to the appropriate logic levels before a read or write operation is executed on the parallel bus.

The seven indirect registers require that the INDPTR (indirect register pointer, one of the four direct registers described above) is initially loaded with the address of the register in the indirect address space before a read or write is performed to the INDIRECT data field.

For example, in order to write to the indirectly addressed I2CSCLL register, the INDPTR register should be loaded with 02h by performing a write to the direct INDPTR register (A1 = 0, A0 = 0). Then the I2CSCLL register can be programmed by writing to the INDIRECT data field (A1 = 1, A0 = 0) in the direct address space. Register mapping is described in Table 3, Table 4 and Figure 5.

Remark: Do not write to any I²C-bus registers while the I²C-bus is busy and the PCA9665/PCA9665A is in master or addressed slave mode.

Table 3. Direct register selection by setting A0 and A1

Register name	Register function	A1	A0	Read/Write	Default
I2CSTA	status	0	0	R	F8h
INDPTR	indirect register pointer	0	0	W	00h
I2CDAT	data	0	1	R/W	00h
I2CCON	control	1	1	R/W	00h[1]
INDIRECT	indirect data field access	1	0	R/W	00h

[1] See Section 8.10 "Power-on reset" for more detail.

Table 4. Indirect register selection by setting A1 = 1 and A0 = 0

Register name	Register function	INDPTR	Read/Write	Default
I2CCOUNT	byte count	00h	R/W	01h
I2CADR	own address	01h	R/W	E0h
I2CSCLL	SCL LOW period	02h	R/W	9Dh
I2CSCLH	SCL HIGH period	03h	R/W	86h
I2CTO	time-out	04h	R/W	FFh
I2CPRESET	parallel software reset	05h	W	00h
I2CMODE	I²C-bus mode	06h	R/W	00h

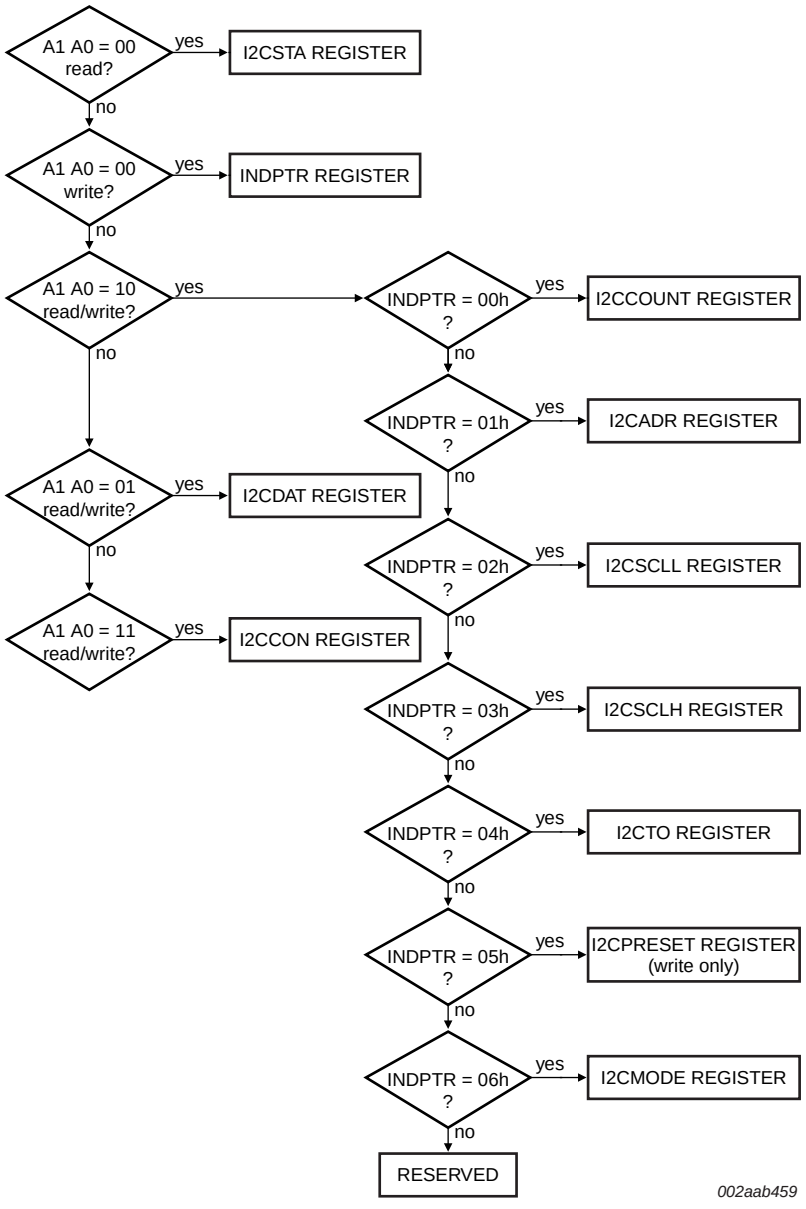


Fig 5. Register mapping flowchart

7.3.1 Direct registers

7.3.1.1 The Status register, I2CSTA (A1 = 0, A0 = 0)

I2CSTA is an 8-bit read-only register. The two least significant bits are always zero. The six most significant bits contain the status code. There are 30 possible status codes. When I2CSTA contains F8h, it indicates the idle state and therefore no serial interrupt is requested. All other I2CSTA values correspond to defined states. When each of these states is entered, a serial interrupt is requested (SI = 1 and $\overline{\text{INT}}$ asserted LOW).

Remark: Data in I2CSTA is valid only when a serial interrupt occurs (SI = 1 and $\overline{\text{INT}}$ asserted LOW). Reading the register when SI = 0 and $\overline{\text{INT}}$ is HIGH may cause wrong values to be read.

Table 5. I2CSTA - Status register (A1 = 0, A0 = 0) bit allocation

7	6	5	4	3	2	1	0
ST5	ST4	ST3	ST2	ST1	ST0	0	0

Table 6. I2CSTA - Status register (A1 = 0, A0 = 0) bit description

Bit	Symbol	Description
7:2	ST[5:0]	status code corresponding to the different I²C-bus states
1:0	-	always at zero

7.3.1.2 The Indirect Pointer register, INDPTR (A1 = 0, A0 = 0)

Table 7. INDPTR - Indirect Register Pointer (A1 = 0, A0 = 0) bit allocation

7	6	5	4	3	2	1	0
-	-	-	-	-	IP2	IP1	IP0

Table 8. INDPTR - Indirect Pointer register (A1 = 0, A0 = 0) bit description

Bit	Symbol	Description
7:3	-	reserved; must be written with zeroes
2:0	IP2 to IP0	address of the indirect register

INDPTR is an 8-bit write-only register. It contains a pointer to a register in the indirect address space (IP[2:0]). The value in the register will determine what indirect register will be accessed when the INDIRECT register is read or written, as defined in Table 4.

7.3.1.3 The I²C-bus Data register, I2CDAT (A1 = 0, A0 = 1)

I2CDAT is an 8-bit read/write register. It contains a byte of serial data to be transmitted or a byte which has just been received. In master mode, this includes the slave address that the master wants to send out on the I²C-bus, with the most significant bit of the slave address in the SD7 bit position and the Read/Write bit in the SD0 bit position. The CPU can read from and write to this 8-bit register while the PCA9665/PCA9665A is not in the process of shifting a byte. This occurs when PCA9665/PCA9665A is in a defined state and the serial interrupt flag is set. Data in I2CDAT remains stable as long as SI is set. Whenever the PCA9665/PCA9665A generates an interrupt, the I2CDAT register contains the data byte that was just transferred on the I²C-bus.

In Byte mode, the CPU can read or write a single byte at a time. In Buffered mode, the CPU can read or write up to 68 bytes at a time. See Section 8.1 “Configuration modes” for more detail.

Remark: The I2CDAT register will capture the serial address as data when addressed via the serial bus.

Remark: In Byte mode only, the data register will capture data from the serial bus during 38h (arbitration lost in slave address + R/$\overline{W}$ or data bytes causing this data in I2CDAT to be changed), so the I2CDAT register will need to be reloaded when the bus becomes free.

In Buffered mode, the data is not written in the data register when arbitration is lost, which keeps the buffer intact.

Table 9. I2CDAT - Data register (A1 = 0, A0 = 1) bit allocation

7	6	5	4	3	2	1	0
SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0

Table 10. I2CDAT - Data register (A1 = 0, A0 = 1) bit description

Bit	Symbol	Description
7:0	SD[7:0]	Eight bits to be transmitted or just received. A logic 1 in I2CDAT corresponds to a HIGH level on the I²C-bus. A logic 0 corresponds to a LOW level on the bus.

7.3.1.4 The Control register, I2CCON (A1 = 1, A0 = 1)

I2CCON is an 8-bit read/write register. Two bits are affected by the bus controller hardware: the SI bit is set when a serial interrupt is requested, and the STO bit is cleared when a STOP condition is present on the I²C-bus. A Write to the I2CCON register via the parallel interface automatically clears the SI bit, which causes the Serial Interrupt line to be de-asserted and the next clock pulse on the SCL line to be generated.

Remark: Since none of the registers should be written to via the parallel interface once the Serial Interrupt line has been de-asserted, all the other registers that need to be modified should be written to before the content of the I2CCON register is modified.

Table 11. I2CCON - Control register (A1 = 1, A0 = 1) bit allocation

7	6	5	4	3	2	1	0
AA	ENSIO	STA	STO	SI	-	-	MODE

Table 12. I2CCON - Control register (A1 = 1, A0 = 1) bit description

Bit	Symbol	Description
7	AA	The Assert Acknowledge flag. AA = 1: If the AA flag is set, an acknowledge (LOW level on SDA) will be returned during the acknowledge clock pulse on the SCL line when: • 'Own slave address' has been received (as defined in I2CADR register). • A data byte has been received while the bus controller is in the Master Receiver mode. • A data byte has been received while the bus controller is in the addressed Slave Receiver mode. AA = 0: if the AA flag is reset, a not acknowledge (HIGH level on SDA) will be returned during the acknowledge clock pulse on SCL when: • 'Own slave address' has been received (as defined in I2CADR register). • A data byte has been received while the PCA9665/PCA9665A is in the Master Receiver mode. • A data byte has been received while the PCA9665/PCA9665A is in the addressed Slave Receiver mode. When the bus controller is in the addressed Slave Transmitter mode, state C8h will be entered after the last data byte is transmitted and an ACK is received from the Master Receiver (see Figure 9 and Figure 13). When SI is cleared, the PCA9665/PCA9665A enters the not addressed Slave Receiver mode, and the SDA line remains at a HIGH level. In state C8h, the AA flag can be set again for future address recognition. When the PCA9665/PCA9665A is in the not addressed slave mode, its own slave address is ignored. Consequently, no acknowledge is returned, and a serial interrupt is not requested. Thus, the bus controller can be temporarily released from the I²C-bus while the bus status is monitored. While the bus controller is released from the bus, START and STOP conditions are detected, and serial data is shifted in. Address recognition can be resumed at any time by setting the AA flag.
6	ENSIO	The bus controller enable bit. ENSIO = 0: When ENSIO is '0', the SDA and SCL outputs are in a high-impedance state. SDA and SCL input signals are ignored, the PCA9665/PCA9665A is in the 'not addressed' slave state. Internal oscillator is off. ENSIO = 1: When ENSIO is '1', the PCA9665/PCA9665A is enabled. After the ENSIO bit is set to '1', it takes 550 μs enable time for the internal oscillator to start up and the serial interface to initialize. The PCA9665/PCA9665A will enter either the master or the slave mode after this time. ENSIO should not be used to temporarily release the PCA9665/PCA9665A from the I²C-bus since, when ENSIO is reset, the I²C-bus status is lost. The AA flag should be used instead (see description of the AA flag above). In the following text, it is assumed that ENSIO = '1' for Normal mode operation. For power-up behavior, please refer to Section 8.10 "Power-on reset".

Table 12. I2CCON - Control register (A1 = 1, A0 = 1) bit description *...continued*

Bit	Symbol	Description
5	STA	The START flag. STA = 1: When the STA bit is set to enter a master mode, the bus controller hardware checks the status of the I²C-bus and generates a START condition if the bus is free. If the bus is not free, then the bus controller waits for a STOP condition (which will free the bus) and generates a START condition after the minimum buffer time (t_{BUF}) has elapsed. If STA is set while the bus controller is already in a master mode and one or more bytes are transmitted or received, the bus controller transmits a repeated START condition. STA may be set at any time. STA may also be set when the bus controller is an addressed slave. A START condition will then be generated after a STOP condition and the minimum buffer time (t_{BUF}) has elapsed. STA = 0: When the STA bit is reset, no START condition or repeated START condition will be generated.
4	STO	The STOP flag. STO = 1: When the STO bit is set while the bus controller is in a master mode, a STOP condition is transmitted on the I²C-bus. When a STOP condition is detected on the bus, the hardware clears the STO flag. If the STA and STO bits are both set and the PCA9665/PCA9665A is in master mode, then a STOP condition is transmitted on the I²C-bus. The bus controller then transmits a START condition after the minimum buffer time (t_{BUF}) has elapsed. STO = 0 : When the STO bit is reset, no STOP condition will be generated.
3	SI	The Serial Interrupt flag. SI = 1: When the SI flag is set, and, if the ENSIO bit is also set, a serial interrupt is requested. SI is set by hardware when one of 29 of the 30 possible states of the bus controller states is entered. The only state that does not cause SI to be set is state F8h, which indicates that no relevant state information is available. While SI is set, the LOW period of the serial clock on the SCL line is stretched, and the serial transfer is suspended. A HIGH level on the SCL line is unaffected by the serial interrupt flag. SI is automatically cleared when the I2CCON register is written. The SI bit cannot be set by the user. SI = 0: When the SI flag is reset, no serial interrupt is requested, and there is no stretching of the serial clock on the SCL line.
2:1	-	Reserved. When I2CCON is read, zeroes are read. Must be written with zeroes.
0	MODE	The Mode flag. MODE = 0; Byte mode. See Section 8.1.1 “Byte mode” for more detail. MODE = 1; buffered mode. See Section 8.1.2 “Buffered mode” for more detail.

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

7.3.1.5 The indirect data field access register, INDIRECT (A1 = 1, A0 = 0)

The registers in the indirect address space can be accessed using the INDIRECT data field. Before writing or reading such a register, the INDPTR register should be written with the address of the indirect register that needs to be accessed. Once the INDPTR register contains the appropriate value, reads and writes to the INDIRECT data field will actually read and write the selected indirect register.

7.3.2 Indirect registers

7.3.2.1 The Byte Count register, I2CCOUNT (indirect address 00h)

The I2CCOUNT register is an 8-bit read/write register. It contains the number of bytes that have been stored in Master/Slave Buffered Receiver mode, and the number of bytes to be sent in Master/Slave Buffered Transmitter mode. Bit 7 is the last byte control bit and applies to the Master/Slave Buffered Receiver mode only. The data in the I2CCOUNT register is relevant only in Buffered mode (MODE = 1) and should not be used (read or written) in Byte mode (MODE = 0).

Table 13. I2CCOUNT - Byte Count register (indirect address 00h) bit allocation

7	6	5	4	3	2	1	0
LB	BC6	BC5	BC4	BC3	BC2	BC1	BC0

Table 14. I2CCOUNT - Byte Count register (indirect address 00h) bit description

Bit	Symbol	Description
7	LB	Last Byte control bit. Master/Slave Buffered Receiver mode only. LB = 1: PCA9665/PCA9665A does not acknowledge the last received byte. LB = 0: PCA9665/PCA9665A acknowledges the last received byte. A future bus transaction must complete the read sequence by not acknowledging the last byte.
6:0	BC[6:0]	Number of bytes to be read or written (up to 68 bytes). If BC[6:0] is equal to 0 or greater than 68 (44h), no bytes will be read or written and an interrupt is immediately generated after writing to the I2CCON register (in Buffered mode only).

7.3.2.2 The Own Address register, I2CADR (indirect address 01h)

I2CADR is an 8-bit read/write register. It is not affected by the bus controller hardware. The content of this register is unused when the controller is in a master mode. A master should never transmit its own slave address. In the slave modes, the seven most significant bits must be loaded with the microcontroller's own slave address and the least significant bit determines if the General Call address will be recognized or not.

Remark: AD[7:1] must be different from the General Call address (000 0000) for proper device operation.

Remark: The I2CADR default value is E0h.

Table 15. I2CADR - Address register (indirect address 01h) bit allocation

7	6	5	4	3	2	1	0
AD7	AD6	AD5	AD4	AD3	AD2	AD1	GC

Table 16. I2CADR - Address register (indirect address 01h) bit description

Bit	Symbol	Description
7:1	AD[7:1]	Own slave address. The most significant bit corresponds to the first bit received from the I²C-bus after a START condition. A logic 1 in I2CADR corresponds to a HIGH level on the I²C-bus, and a logic 0 corresponds to a LOW level on the bus.
0	GC	General Call. GC = 1: General Call address (00h) is recognized. GC = 0: General Call address (00h) is ignored.

7.3.2.3 The Clock Rate registers, I2CSCLL and I2CSCLH (indirect addresses 02h and 03h)

I2CSCLL and I2CSCLH are 8-bit read/write registers. They define the data rate for the PCA9665/PCA9665A when used as a bus master. The actual frequency is determined by t_{HIGH} (time where SCL is HIGH), t_{LOW} (time where SCL is LOW), t_r (rise time), t_f (fall time) and t_d (delay time) values.

t_{HIGH} and t_{LOW} are calculated based on the values that are programmed into I2CSCLH and I2CSCLL registers and the internal oscillator frequency. t_r and t_f are system/application dependent. t_d for the PCA9665 is approximately 175 ns and the PCA9665A is approximately 300 ns.

$$f_{SCL} = \frac{1}{T_{osc}(I2CSCLL + I2CSCLH) + t_r + t_f + t_d} \quad (1)$$

with T_{osc} = internal oscillator period = 35 ns ± 5 ns for the PCA9665 and 33 ns ± 5 ns for the PCA9665A

The delay time 't_d' is the sum of the time between the oscillator edge of the SCLL terminal count until the SCL is up to $0.3V_{DD}$ and the oscillator edge of the SCLH terminal count until the SCL is down to $0.7V_{DD}$.

Remark: The I2CMODE register needs to be programmed before programming the I2CSCLL and I2CSCLH registers in order to know which I²C-bus mode is selected. See Section 7.3.2.6 "The I²C-bus mode register, I2CMODE (indirect address 06h)" for more detail.

Standard-mode is the default selected mode at power-up or after reset.

Table 17. I2CSCLL - Clock Rate Low register (indirect address 02h) bit allocation

7	6	5	4	3	2	1	0
L7	L6	L5	L4	L3	L2	L1	L0

Table 18. I2CSCLL - Clock Rate Low register (indirect address 02h) bit description

Bit	Symbol	Description
7:0	L[7:0]	Eight bits defining the LOW state of SCL.

Table 19. I2CSCLH - Clock Rate High register (indirect address 03h) bit allocation

7	6	5	4	3	2	1	0
H7	H6	H5	H4	H3	H2	H1	H0

Table 20. I2CSCLH - Clock Rate High register (indirect address 03h) bit description

Bit	Symbol	Description
7:0	H[7:0]	Eight bits defining the HIGH state of SCL.

7.3.2.4 The Time-out register, I2CTO (indirect address 04h)

I2CTO is an 8-bit read/write register. It is used to determine the maximum time that SCL is allowed to be in a LOW logic state before the I²C-bus state machine is reset or the PCA9665/PCA9665A initiates a forced action on the I²C-bus.

When the I²C-bus interface is operating, I2CTO is loaded in the time-out counter at every LOW SCL transition.

Table 21. I2CTO - Time-out register (indirect register 04h) bit allocation

7	6	5	4	3	2	1	0
TE	TO6	TO5	TO4	TO3	TO2	TO1	TO0

Table 22. I2CTO - Time-out register (indirect register 04h) bit description

Bit	Symbol	Description
7	TE	Time-out enable/disable TE = 1: Time-out function enabled TE = 0: Time-out function disabled
6:0	TO[6:0]	Time-out value. The time-out value may vary some, and is an approximate value. PCA9665 typical time-out period = (I2CTO[6:0] + 1) × 143 μs. PCA9665A typical time-out period = (I2CTO[6:0] + 1) × 134 μs.

The Time-out register can be used in the following cases:

- When the bus controller, in the master mode, wants to send a START condition and the SCL line is held LOW by some other device. Then the bus controller waits a time period equivalent to the time-out value for the SCL to be released. In case it is not released, the bus controller concludes that there is a bus error, loads 78h in the I2CSTA register, generates an interrupt signal and releases the SCL and SDA lines. After the microcontroller reads the status register, it needs to send a reset in order to reset the bus controller.
- In the master mode, the time-out feature starts every time the SCL goes LOW. If SCL stays LOW for a time period equal to or greater than the time-out value, the bus controller concludes there is a bus error and behaves in the manner described above. When the I²C-bus interface is operating, I2CTO is loaded in the time-out counter at every SCL transition. See Section 8.11 “Reset” for more information.
- In case of a forced access to the I²C-bus. (See more details in Section 8.9.3 “Forced access to the I²C-bus”.)

7.3.2.5 The Parallel Software Reset register, I2CPRESET (indirect address 05h)

I2CPRESET is an 8-bit write-only register. Programming the I2CPRESET register allows the user to reset the PCA9665/PCA9665A 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.

7.3.2.6 The I²C-bus mode register, I2CMODE (indirect address 06h)

I2CMODE is an 8-bit read/write register. It contains the control bits that select the correct timing parameters when the device is used in master mode (AC[1:0]). Timing parameters involved with AC[1:0] are t_{BUF}, $t_{HD;STA}$, $t_{SU;STA}$, $t_{SU;STO}$, t_{HIGH}, t_{LOW}.

Table 23. I2CMODE - I²C-bus Mode register (indirect address 06h) bit allocation

7	6	5	4	3	2	1	0
-	-	-	-	-	-	AC1	AC0

Table 24. I2CMODE - I²C-bus Mode register (indirect address 06h) bit description

Bit	Symbol	Description
7:2	-	Reserved. When I2CMODE is read, zeroes are read. Must be written with zeroes.
1:0	AC[1:0]	I²C-bus mode selection to ensure proper timing parameters (see Table 25 and Table 51). AC[1:0] = 00: Standard-mode AC parameters selected. AC[1:0] = 01: Fast-mode AC parameters selected. AC[1:0] = 10: Fast-mode Plus AC parameters selected. AC[1:0] = 11: Turbo mode. In this mode, the user is not limited to a maximum frequency of 1 MHz.

Remark: Change from an I²C-bus mode to a slower one (Fast-mode to Standard-mode, for example) will cause the HIGH and LOW timings of SCL to be violated. It is then required to program the I2CSCLL and I2CSCLH registers with values in accordance with the selected mode.

Table 25. I²C-bus mode selection example[1]

I2CSCLL (hexadecimal)	I2CSCLH (hexadecimal)	I²C-bus frequency (kHz)		AC[1:0]	Mode
		PCA9665[2]	PCA9665A[3]		
9D	86	98.0	103.3	00	Standard
2C	14	371.1	371.4	01	Fast
11	09	836.8	788.6	10	Fast-mode Plus
0E	05	1015	932.8	11	Turbo mode

[1] I2CSCLL and I2CSCLH values in the table also represents the minimum values that can be used for the corresponding I²C-bus mode. Use of lower values will cause the minimum values to be loaded.

[2] Using the formula $f_{SCL} = \frac{1}{T_{osc}(I2CSCLL + I2CSCLH) + t_r + t_f + t_d}$ with T_{osc} at 30 ns (minimum pulse width), t_d = 175 ns, and t_r and t_f at maximum data sheet values for the mode.

[3] Using the formula $f_{SCL} = \frac{1}{T_{osc}(I2CSCLL + I2CSCLH) + t_r + t_f + t_d}$ with T_{osc} at 28 ns (minimum pulse width), t_d = 300 ns, and t_r and t_f at maximum data sheet values for the mode.

8. PCA9665/PCA9665A modes

8.1 Configuration modes

Byte mode and Buffered mode are selected using the MODE bit in I2CCON register:

MODE = 0: Byte mode

MODE = 1: Buffered mode

8.1.1 Byte mode

The Byte mode allows communication on a single command basis. Only one specific command is executed at a time and the Status Register is updated once this single command has been performed. A command can be a START, a STOP, a Byte Write, a Byte Read, and so on.

8.1.2 Buffered mode

The Buffered mode allows several instructions to be executed before an Interrupt is generated and before the I2CSTA register is updated. This allows the microcontroller to request a sequence, up to 68 bytes in a single transmission and lets the PCA9665/PCA9665A perform it without having to access the Status Register and the Control Register each time a single command is performed. The microcontroller can then perform other tasks while the PCA9665/PCA9665A performs the requested sequence.

The number of bytes that needs to be sent from the internal buffer (Transmitter mode) or received into the internal buffer (Receiver mode) is defined in the indirectly addressed I2CCOUNT Register (BC[6:0]). Up to 68 bytes can be sent or received.

8.2 Operating modes

The four operating modes are:

- Master Transmitter
- Master Receiver
- Slave Receiver
- Slave Transmitter

Each mode can be used on a byte basis (Byte mode) or in an up to 68-byte buffer basis (Buffered mode).

Data transfers in each mode of operation are shown in Figure 6 through Figure 9. These figures contain the following abbreviations:

S — START condition

SLA — 7-bit slave address

R — Read bit (HIGH level at SDA)

W — Write bit (LOW level at SDA)

A — Acknowledge bit (LOW level at SDA)

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

Data — 8-bit data byte

P — STOP condition

In Figure 6, Figure 7, Figure 8, Figure 9, Figure 10, Figure 11, Figure 12 and Figure 13, circles are used to indicate when the serial interrupt flag is set. A serial interrupt is not generated when I2CSTA = F8h. This happens on a STOP condition or when an external reset is generated (at power-up, when $\overline{\text{RESET}}$ pin is going LOW or during a software reset on the parallel bus). The numbers in the circles show the status code held in the I2CSTA register. At these points, a service routine must be executed to continue or complete the serial transfer. These service routines are not critical since the serial transfer is suspended until the serial interrupt flag is cleared by software.

When a serial interrupt routine is entered, the status code in I2CSTA is used to branch to the appropriate service routine. For each status code, the required software action and details of the following serial transfer are given in Table 27, Table 28, Table 31, Table 32, Table 35, Table 36, Table 40, and Table 41.

8.3 Byte mode

8.3.1 Master Transmitter Byte mode

In the Master Transmitter Byte mode, a number of data bytes are transmitted to a slave receiver (see Figure 6). Before the Master Transmitter Byte mode can be entered, I2CCON must be initialized as shown in Table 26.

Table 26. I2CCON initialization (Byte mode)

Bit	7	6	5	4	3	2	1	0
Symbol	AA	ENSIO	STA	STO	SI	reserved	reserved	MODE
Value	X	1	0	0	0	X	X	0

ENSIO must be set to logic 1 to enable the PCA9665/PCA9665A. If the AA bit is reset, the PCA9665/PCA9665A will not acknowledge its own slave address in the event of another device becoming master of the bus. (In other words, if AA is reset, PCA9665/PCA9665A cannot enter a slave mode.) STA, STO, and SI must be reset. Once ENSIO has been set to 1, it takes about 550 μs for the oscillator to start up.

The Master Transmitter Byte mode may now be entered by setting the STA bit. The I²C-bus state machine will first test the I²C-bus and generate a START condition as soon as the bus becomes free. When a START condition is transmitted, the serial interrupt flag (SI) is set, the Interrupt line ($\overline{\text{INT}}$) goes LOW and the status code in the status register (I2CSTA) will be 08h. This status code must be used to vector to an interrupt service routine that loads I2CDAT with the slave address and the data direction bit (SLA+W). A write to I2CCON resets the SI bit, clears the Interrupt ($\overline{\text{INT}}$ goes HIGH) and allows the serial transfer to continue.

When the slave address with the direction bit have been transmitted, the Serial Interrupt flag (SI) is set again, the Interrupt line ($\overline{\text{INT}}$) goes LOW again and I2CSTA is loaded with the following possible codes:

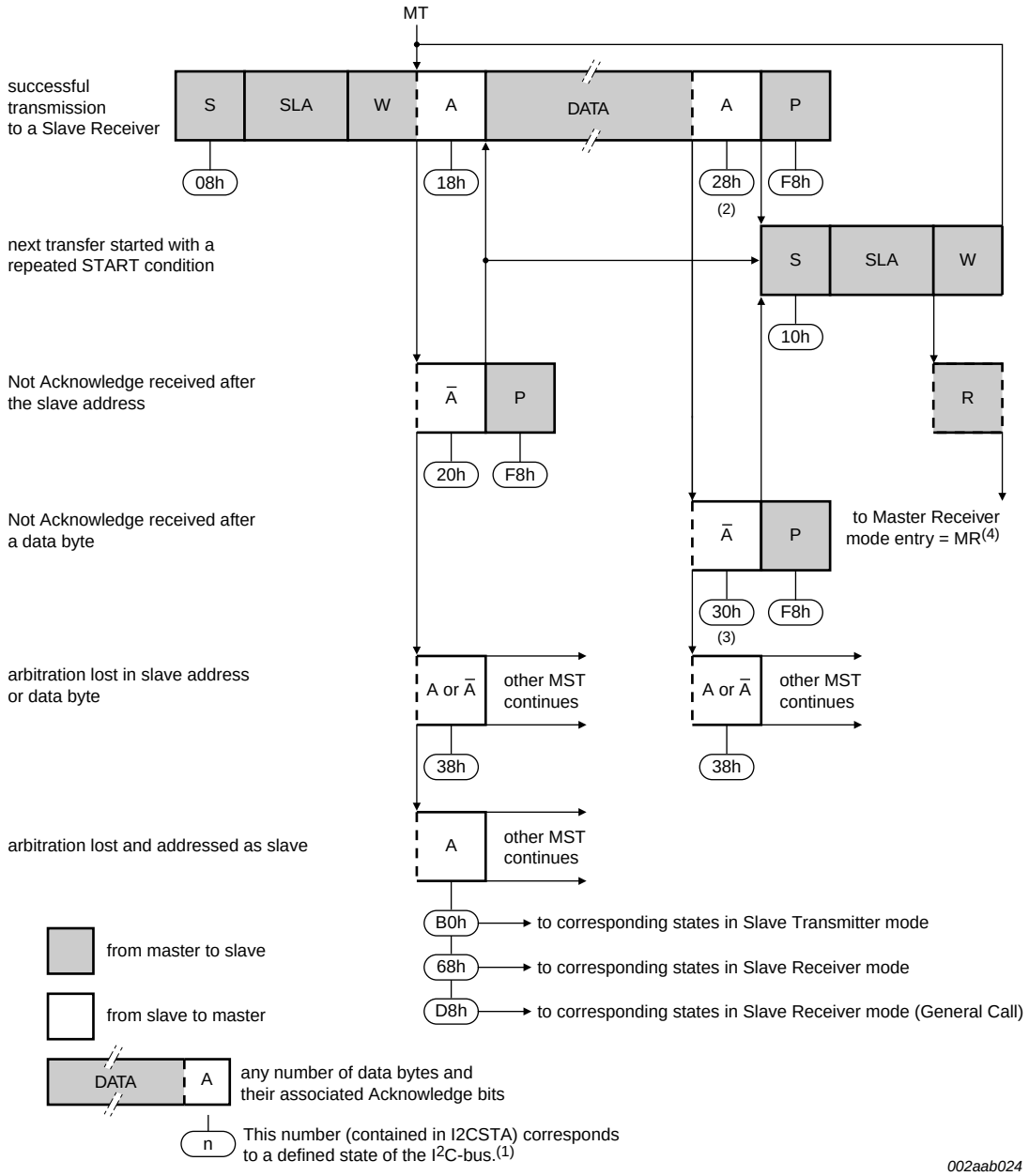
- 18h if an acknowledgment bit (ACK) has been received
- 20h if an no acknowledgment bit (NACK) has been received
- 38h if the PCA9665/PCA9665A lost the arbitration

- B0h if the PCA9665/PCA9665A lost the arbitration and is addressed as a slave transmitter (slave mode enabled with AA = 1)
- 68h if the PCA9665/PCA9665A lost the arbitration and is addressed as a slave receiver (slave mode enabled with AA = 1)
- D8h if the PCA9665/PCA9665A lost the arbitration and is addressed as a slave receiver during a General Call sequence (slave mode enabled with AA = 1 and General Call address enabled with GC = 1 in I2CADR register)

The appropriate action to be taken for each of these status codes is detailed in Table 27. ENSIO is not affected by the serial transfer and is not referred to in Table 27.

After a repeated START condition (state 10h), the PCA9665/PCA9665A may switch to the Master Receiver mode by loading I2CDAT with SLA+R.

Remark: A master should not transmit its own slave address.



(1) See Table 27

(2) Defined state when a single byte is sent and an ACK is received.

(3) Defined state when a single byte is sent and a NACK is received.

(4) Master Receiver Byte mode is entered when MODE = 0. Master Receiver Buffered mode is entered when MODE = 1.

Fig 6. Format and states in the Master Transmitter Byte mode (MODE = 0)

Table 27. Master Transmitter Byte mode (MODE = 0)

Status code (I2CSTA)	Status of the I²C-bus and the PCA9665/65A	Application software response						Next action taken by the PCA9665/PCA9665A
		To/from I2CDAT	To I2CCON					
			STA	STO	SI	AA	MODE	
08h	A START condition has been transmitted	Load SLA+W	X	X	0	X	0	SLA+W will be transmitted; ACK/NACK will be received
10h	A repeated START condition has been transmitted	Load SLA+W or	X	X	0	X	0	SLA+W will be transmitted; ACK/NACK will be received
		Load SLA+R	X	X	0	X	0	SLA+R will be transmitted; PCA9665/PCA9665A will be switched to Master Receiver Byte mode
18h	SLA+W has been transmitted; ACK has been received	Load data byte or	0	0	0	X	0	Data byte will be transmitted; ACK/NACK will be received
		no I2CDAT action or	1	0	0	X	0	Repeated START will be transmitted;
		no I2CDAT action or	0	1	0	X	0	STOP condition will be transmitted; STO flag will be reset
		no I2CDAT action	1	1	0	X	0	STOP condition followed by a START condition will be transmitted; STO flag will be reset
20h	SLA+W has been transmitted; NACK has been received	Load data byte or	0	0	0	X	0	Data byte will be transmitted; ACK/NACK will be received
		no I2CDAT action or	1	0	0	X	0	Repeated START will be transmitted;
		no I2CDAT action or	0	1	0	X	0	STOP condition will be transmitted; STO flag will be reset
		no I2CDAT action	1	1	0	X	0	STOP condition followed by a START condition will be transmitted; STO flag will be reset
28h	Data byte in I2CDAT has been transmitted; ACK has been received	Load data byte or	0	0	0	X	0	Data byte will be transmitted; ACK/NACK will be received
		no I2CDAT action or	1	0	0	X	0	Repeated START will be transmitted;
		no I2CDAT action or	0	1	0	X	0	STOP condition will be transmitted; STO flag will be reset
		no I2CDAT action	1	1	0	X	0	STOP condition followed by a START condition will be transmitted; STO flag will be reset

Table 27. Master Transmitter Byte mode (MODE = 0) *…continued*

Status code (I2CSTA)	Status of the I²C-bus and the PCA9665/65A	Application software response						Next action taken by the PCA9665/PCA9665A
		To/from I2CDAT	To I2CCON					
			STA	STO	SI	AA	MODE	
30h	Data byte in I2CDAT has been transmitted; NACK has been received	Load data byte or	0	0	0	X	0	Data byte will be transmitted; ACK/NACK will be received
		no I2CDAT action or	1	0	0	X	0	Repeated START will be transmitted;
		no I2CDAT action or	0	1	0	X	0	STOP condition will be transmitted; STO flag will be reset
		no I2CDAT action	1	1	0	X	0	STOP condition followed by a START condition will be transmitted; STO flag will be reset
38h	Arbitration lost in SLA+W or Data bytes	No I2CDAT action or	0	0	0	0	0	I²C-bus will be released; PCA9665/PCA9665A will enter Slave mode.
		No I2CDAT action or	0	0	0	1	0	I²C-bus will be released; PCA9665/PCA9665A will enter the Slave mode.
		No I2CDAT action	1	0	0	X	0	A START condition will be transmitted when the bus becomes free

8.3.2 Master Receiver Byte mode

In the Master Receiver Byte mode, a number of data bytes are received from a slave transmitter one byte at a time (see Figure 7). The transfer is initialized as in the Master Transmitter Byte mode.

The Master Receiver Byte mode may now be entered by setting the STA bit. The I²C-bus state machine will first test the I²C-bus and generate a START condition as soon as the bus becomes free. When a START condition is transmitted, the Serial Interrupt flag (SI) is set, the Interrupt line ($\overline{INT}$) goes LOW and the status code in the status register (I2CSTA) will be 08h. This status code must be used to vector to an interrupt service routine that loads I2CDAT with the slave address and the data direction bit (SLA+R). A write to I2CCON resets the SI bit, clears the Interrupt ($\overline{INT}$ goes HIGH) and allows the serial transfer to continue.

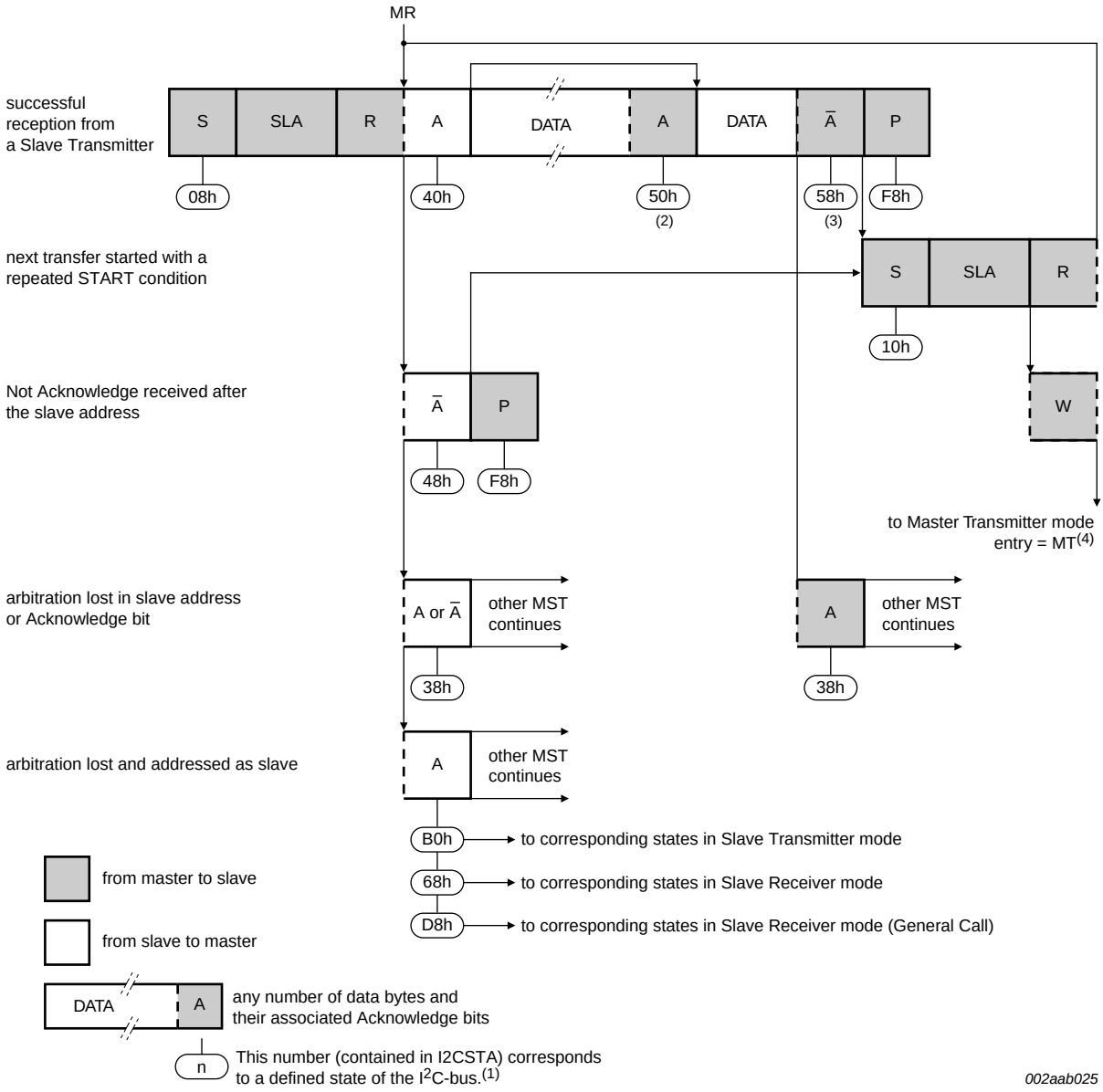
When the slave address and the data direction bit have been transmitted, the serial interrupt flag (SI) is set again, the Interrupt line ($\overline{INT}$) goes LOW again and I2CSTA is loaded with the following possible codes:

- 40h if an acknowledgment bit (ACK) has been received for the slave address with direction bit
- 48h if a no acknowledgment bit (NACK) has been received for the slave address with direction bit
- 38h if the PCA9665/PCA9665A lost the arbitration
- B0h if the PCA9665/PCA9665A lost the arbitration and is addressed as a slave transmitter (slave mode enabled with AA = 1)
- 68h if the PCA9665/PCA9665A lost the arbitration and is addressed as a slave receiver (slave mode enabled with AA = 1)
- D8h if the PCA9665/PCA9665A lost the arbitration and is addressed as a slave receiver during a General Call sequence (slave mode enabled with AA = 1 and General Call address enabled with GC = 1 in I2CADR register).

The appropriate action to be taken for each of these status codes is detailed in Table 28. ENSIO is not affected by the serial transfer and is not referred to in Table 28.

After a repeated START condition (state 10h), the PCA9665/PCA9665A may switch to the Master Transmitter mode by loading I2CDAT with SLA+W.

Remark: A master should not transmit its own slave address.



(1) See Table 28.

(2) Defined state when a single byte is received and an ACK is sent (AA = 1).

(3) Defined state when a single byte is received and a NACK is sent (AA = 0).

(4) Master Transmitter Byte mode is entered when MODE = 0. Master Transmitter Buffered mode is entered when MODE = 1.

Fig 7. Format and states in the Master Receiver Byte mode (MODE = 0)

Table 28. Master Receiver Byte mode (MODE = 0)

Status code (I2CSTA)	Status of the I²C-bus and the PCA9665/65A	Application software response						Next action taken by the PCA9665/PCA9665A
		To/from I2CDAT	To I2CCON					
			STA	STO	SI	AA	MODE	
08h	A START condition has been transmitted	Load SLA+R	X	X	0	X	0	SLA+R will be transmitted; ACK/NACK bit will be received
10h	A repeated START condition has been transmitted	Load SLA+R or	X	X	0	X	0	SLA+R will be transmitted; ACK/NACK bit will be received
		Load SLA+W	X	X	0	X	0	SLA+W will be transmitted; PCA9665/PCA9665A will be switched to Master Transmitter Byte mode
38h	Arbitration lost in NACK bit	No I2CDAT action or	0	0	0	X	0	I²C-bus will be released; PCA9665/PCA9665A will enter a slave mode
		no I2CDAT action	1	0	0	X	0	A START condition will be transmitted when the bus becomes free
40h	SLA+R has been transmitted; ACK has been received	No I2CDAT action or	0	0	0	0	0	Data byte will be received; NACK bit will be returned
		no I2CDAT action	0	0	0	1	0	Data byte will be received; ACK bit will be returned
48h	SLA+R has been transmitted; NACK has been received	No I2CDAT action or	1	0	0	X	0	Repeated START condition will be transmitted
		no I2CDAT action or	0	1	0	X	0	STOP condition will be transmitted; STO flag will be reset
		no I2CDAT action	1	1	0	X	0	STOP condition followed by a START condition will be transmitted; STO flag will be reset
50h	Data byte has been received; ACK has been returned	Read data byte or	0	0	0	0	0	Data byte will be received; NACK bit will be returned
		read data byte	0	0	0	1	0	Data byte will be received; ACK bit will be returned
58h	Data byte has been received; NACK has been returned	Read data byte or	1	0	0	X	0	Repeated START condition will be transmitted
		read data byte or	0	1	0	X	0	STOP condition will be transmitted; STO flag will be reset
		read data byte	1	1	0	X	0	STOP condition followed by a START condition will be transmitted; STO flag will be reset

8.3.3 Slave Receiver Byte mode

In the Slave Receiver Byte mode, a number of data bytes are received from a master transmitter one byte at a time (see Figure 8). To initiate the Slave Receiver mode, I2CADR and I2CCON must be loaded as shown in Table 29 and Table 30.

Table 29. I2CADR initialization

Bit	7	6	5	4	3	2	1	0
Symbol	AD7	AD6	AD5	AD4	AD3	AD2	AD1	GC
Value	own slave address							X

The upper 7 bits are the I²C-bus address to which PCA9665/PCA9665A will respond when addressed by a master. GC is the control bit that allows the PCA9665/PCA9665A to respond or not to the General Call address (00h).

When programmed to logic 1, the PCA9665/PCA9665A will acknowledge the General Call address.

When programmed to logic 0, the PCA9665/PCA9665A will not acknowledge the General Call address.

Table 30. I2CCON initialization

Bit	7	6	5	4	3	2	1	0
Symbol	AA	ENSIO	STA	STO	SI	-	-	MODE
Value	1	1	0	0	0	X	X	0

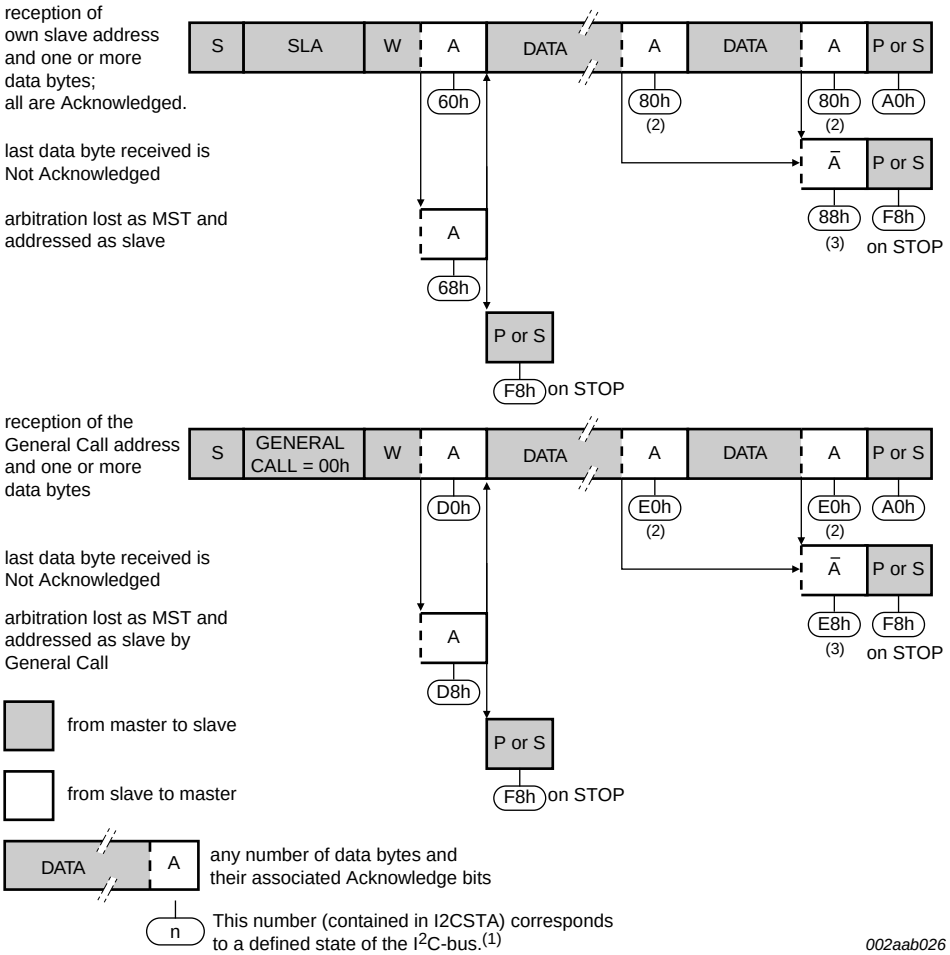
ENSIO must be set to logic 1 to enable the I²C-bus interface. The AA bit must be set to enable PCA9665/PCA9665A to acknowledge its own slave address, STA, STO, and SI must be reset.

When I2CADR and I2CCON have been initialized, the PCA9665/PCA9665A waits until it is addressed by its own slave address followed by the data direction bit which must be ‘0’ (W) to operate in the Slave Receiver mode. After its own slave address and the W bit have been received, the Serial Interrupt flag (SI) is set, the Interrupt line ($\overline{INT}$) goes LOW, and I2CSTA is loaded with 60h. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken is detailed in Table 31.

The Slave Receiver Buffered mode may also be entered when:

- The arbitration is lost while the PCA9665/PCA9665A is in the master mode. See status 68h and D8h.
- The General Call Address (00h) has been received (General Call address enabled with GC = 1). See status D0h.

If the AA bit is reset during a transfer, the PCA9665/PCA9665A will return a not acknowledge (logic 1) on SDA after the next received data byte. While AA is reset, the I²C-bus state machine does not respond to its own slave address. However, the I²C-bus is still monitored and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate PCA9665/PCA9665A from the I²C-bus.



(1) See Table 31.

(2) Defined state when a single byte is received and an ACK is sent (AA = 1).

(3) Defined state when a single byte is received and a NACK is sent (AA = 0).

Fig 8. Format and states in the Slave Receiver Byte mode (MODE = 0)

Table 31. Slave Receiver Byte mode (MODE = 0)

Status code (I2CSTA)	Status of the I²C-bus and the PCA9665/65A	Application software response						Next action taken by the PCA9665/PCA9665A
		To/from I2CDAT	To I2CCON					
			STA	STO	SI	AA	MODE	
60h	Own SLA+W has been received; ACK has been returned	No I2CDAT action or	X	X	0	0	0	Data byte will be received and NACK will be returned
		no I2CDAT action	X	X	0	1	0	Data byte will be received and ACK will be returned
68h	Arbitration lost in SLA+R/W as master; Own SLA+W has been received, ACK has been returned	No I2CDAT action or	X	X	0	0	0	Data byte will be received and NACK will be returned
		no I2CDAT action	X	X	0	1	0	Data byte will be received and ACK will be returned
D0h	General Call address (00h) has been received; ACK has been returned.	No I2CDAT action or	X	X	0	0	0	Data byte will be received and NACK will be returned.
		no I2CDAT action	X	X	0	1	0	Data byte will be received and ACK will be returned.
D8h	Arbitration lost in SLA = R/W as master; General Call address has been received; ACK bit has been returned.	No I2CDAT action or	X	X	0	0	0	Data byte will be received and NACK will be returned.
		no I2CDAT action	X	X	0	1	0	Data byte will be received and ACK will be returned.
80h	Previously addressed with own slave address; DATA has been received; ACK has been returned	Read data byte or	X	X	0	0	0	Data byte will be received and NACK will be returned
		read data byte	X	X	0	1	0	Data byte will be received and ACK will be returned
88h	Previously addressed with own slave address; DATA byte has been received; NACK has been returned	Read data byte or	0	X	0	0	0	Switched to not addressed slave mode; no recognition of own SLA or General Call address
		read data byte or	0	X	0	1	0	Switched to not addressed slave mode; Own slave address will be recognized; General Call address will be recognized if GC = 1.
		read data byte or	1	X	0	0	0	Switched to not addressed slave mode; no recognition of own slave address or General Call address. A START condition will be transmitted when the bus becomes free
		read data byte	1	X	0	1	0	Switched to not addressed slave mode; Own slave address will be recognized; General Call will be recognized if GC = 1. A START condition will be transmitted when the bus becomes free.

Table 31. Slave Receiver Byte mode (MODE = 0) *…continued*

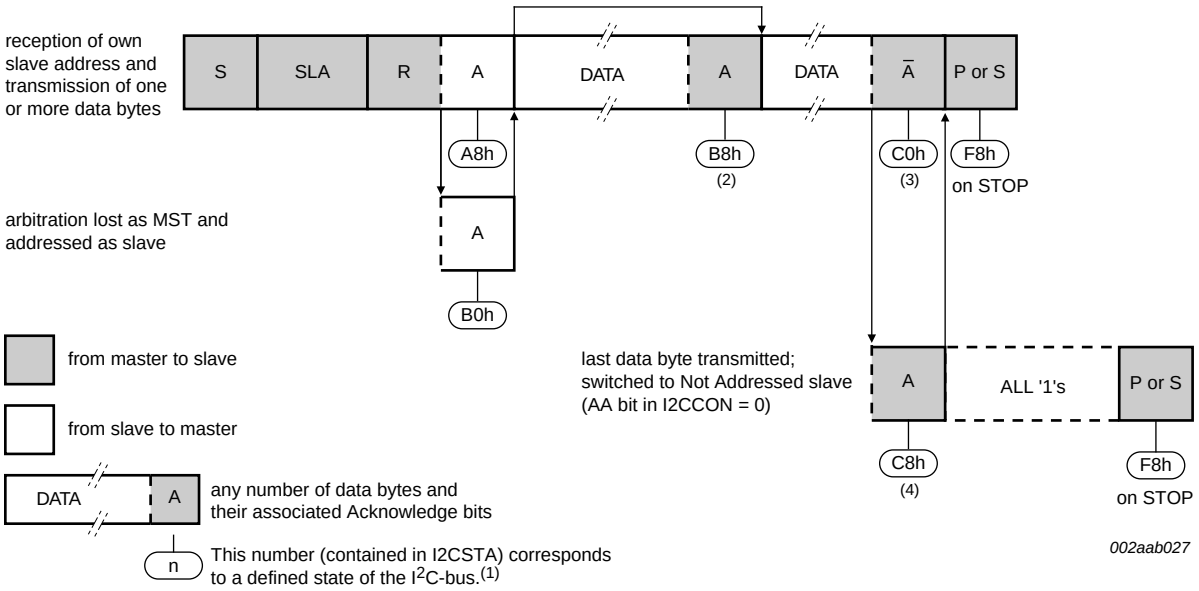
Status code (I2CSTA)	Status of the I²C-bus and the PCA9665/65A	Application software response						Next action taken by the PCA9665/PCA9665A
		To/from I2CDAT	To I2CCON					
			STA	STO	SI	AA	MODE	
E0h	Previously addressed with General Call; Data has been received; ACK has been returned	Read data byte or	X	X	0	0	0	Data byte will be received and NACK will be returned.
		read data byte	X	X	0	1	0	Data byte will be received and ACK will be returned.
E8h	Previously addressed with General Call; Data has been received; NACK has been returned	Read data byte or	0	X	0	0	0	Switched to not addressed slave mode; no recognition of own slave address or General Call address.
		read data byte or	0	X	0	1	0	Switched to not addressed slave mode; own slave address will be recognized; General Call address will be recognized if GC = 1.
		read data byte or	1	0	0	0	0	Switched to not addressed slave mode; no recognition of own slave address or General Call address. A START condition will be transmitted when the bus becomes free.
		read data byte	1	0	0	1	0	Switched to not addressed slave mode; own slave address will be recognized; General Call address will be recognized if GC = 1. A START condition will be transmitted when the bus becomes free.
A0h	A STOP condition or repeated START condition has been received while still addressed as Slave Receiver	No I2CDAT action or	0	X	0	0	0	Switched to not addressed slave mode; no recognition of own slave address or General Call address.
		No I2CDAT action or	0	X	0	1	0	Switched to not addressed slave mode; Own slave address will be recognized; General Call will be recognized if GC = 1.
		No I2CDAT action or	1	X	0	0	0	Switched to not addressed slave mode; no recognition of own slave address or General Call. A START condition will be transmitted when the bus becomes free
		No I2CDAT action	1	X	0	1	0	Switched to not addressed slave mode; Own slave address will be recognized; General Call will be recognized if GC = 1. A START condition will be transmitted when the bus becomes free.

8.3.4 Slave Transmitter Byte mode

In the Slave Transmitter Byte mode, a number of data bytes are transmitted to a master receiver one byte at a time (see Figure 9). Data transfer is initialized as in the Slave Receiver Byte mode. When I2CADR and I2CCON have been initialized, the PCA9665/PCA9665A waits until it is addressed by its own slave address followed by the data direction bit which must be '1' (R) for the PCA9665/PCA9665A to operate in the Slave Transmitter mode. After its own slave address and the R bit have been received, the Serial Interrupt flag (SI) is set, the Interrupt line ($\overline{INT}$) goes LOW and I2CSTA is loaded with A8h. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken is detailed in Table 32.

The Slave Transmitter Byte mode may also be entered if arbitration is lost while the PCA9665/PCA9665A is in the master mode. See state B0h and appropriate actions in Table 32.

If the AA bit is reset during a transfer, the PCA9665/PCA9665A will transmit the last byte of the transfer and enter state C8h. The PCA9665/PCA9665A is switched to the not addressed slave mode and will ignore the master receiver if it continues the transfer. Thus the master receiver receives all '1's as serial data. While AA is reset, the PCA9665/PCA9665A does not respond to its own slave address. However, the I²C-bus is still monitored, and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate SIO from the I²C-bus.



(1) See Table 31.

(2) Defined state when a single byte is transmitted and an ACK is received.

(3) Defined state when a single byte is transmitted and a NACK is received.

(4) Defined state when a single byte is transmitted and the PCA9665/PCA9665A goes to the non-addressed mode (AA = 0) and an ACK is received.

Fig 9. Format and states in the Slave Transmitter Byte mode (MODE = 0)

Table 32. Slave Transmitter Byte mode (MODE = 0)

Status code (I2CSTA)	Status of the I²C-bus and the PCA9665/65A	Application software response						Next action taken by PCA9665/PCA9665A
		To/from I2CDAT	To I2CCON					
			STA	STO	SI	AA	MODE	
A8h	Own SLA+R has been received; ACK has been returned	Load data byte or	X	X	0	0	0	Last data byte will be transmitted and ACK/NACK bit will be received
		load data byte	X	X	0	1	0	Data byte will be transmitted; ACK/NACK will be received
B0h	Arbitration lost in SLA+R/W as master; Own SLA+R has been received, ACK has been returned	Load data byte or	X	X	0	0	0	Last data byte will be transmitted and ACK/NACK bit will be received
		load data byte	X	X	0	1	0	Data byte will be transmitted; ACK bit will be received
B8h	Data byte in I2CDAT has been transmitted; ACK has been received	Load data byte or	X	X	0	0	0	Last data byte will be transmitted and ACK/NACK bit will be received
		load data byte	X	X	0	1	0	Data byte will be transmitted; ACK/NACK bit will be received
C0h	Data byte in I2CDAT has been transmitted; NACK has been received	No I2CDAT action or	0	X	0	0	0	Switched to not addressed slave mode; no recognition of own slave address. General Call address recognized if GC = 1.
		no I2CDAT action or	0	X	0	1	0	Switched to slave mode; Own slave address will be recognized. General Call address recognized if GC = 1.
		no I2CDAT action or	1	X	0	0	0	Switched to not addressed slave mode; no recognition of own slave address. General Call address recognized if GC = 1. A START condition will be transmitted when the bus becomes free
		no I2CDAT action	1	X	0	1	0	Switched to slave mode; Own slave address will be recognized. General Call address recognized if GC = 1. A START condition will be transmitted when the bus becomes free.
C8h	Last data byte in I2CDAT has been transmitted (AA = 0); ACK has been received	No I2CDAT action or	0	X	0	0	0	Switched to not addressed slave mode; no recognition of own slave address. General Call address recognized if GC = 1.
		no I2CDAT action or	0	X	0	1	0	Switched to slave mode; Own slave address will be recognized. General Call address recognized if GC = 1.
		no I2CDAT action or	1	X	0	0	0	Switched to not addressed slave mode; no recognition of own slave address. General Call address recognized if GC = 1. A START condition will be transmitted when the bus becomes free
		no I2CDAT action	1	X	0	1	0	Switched to slave mode; Own slave address will be recognized. General Call address recognized if GC = 1. A START condition will be transmitted when the bus becomes free.

8.4 Buffered mode

8.4.1 Master Transmitter Buffered mode

In the Master Transmitter Buffered mode, a number of data bytes are transmitted to a slave receiver several bytes at a time (see Figure 10). Before the Master Transmitter Buffered mode can be entered, I2CCON must be initialized as shown in Table 33.

Table 33. I2CCON initialization (Buffered mode)

Bit	7	6	5	4	3	2	1	0
Symbol	AA	ENSIO	STA	STO	SI	reserved	reserved	MODE
Value	X	1	0	0	0	X	X	1

Table 34. I2CCOUNT programming

Bit	7	6	5	4	3	2	1	0
Symbol	LB	BC6	BC5	BC4	BC3	BC2	BC1	BC0
Value	X	number of bytes received in a single sequence (1 byte to 68 bytes)						

ENSIO must be set to logic 1 to enable the PCA9665/PCA9665A. If the AA bit is reset, the PCA9665/PCA9665A will not acknowledge its own slave address in the event of another device becoming master of the bus (in other words, if AA is reset, the PCA9665/PCA9665A cannot enter a slave mode). STA, STO, and SI must be reset. Once ENSIO has been set to logic 1, it takes about 550 μs for the oscillator to start up.

The Master Transmitter Buffered mode may now be entered by setting the STA bit. The I²C-bus state machine will first test the I²C-bus and generate a START condition as soon as the bus becomes free. When a START condition is transmitted, the Serial Interrupt flag (SI) is set, the Interrupt line ($\overline{INT}$) goes LOW and the status code in the status register (I2CSTA) will be 08h. This status code must be used to vector to an interrupt service routine that loads I2CDAT with the slave address and the data direction bit (SLA+W) followed by the number of data bytes to be sent. The byte count register (I2CCOUNT) has been previously programmed with the number of bytes that need to be sent in a single sequence (BC[6:0]) as shown in Table 34. LB bit is only used for the Receiver Buffered modes and can be programmed to either logic 0 or logic 1. The total number of bytes loaded in I2CDAT (slave address with direction bit plus data bytes) must be equal to the value programmed in I2CCOUNT. A write to I2CCON resets the SI bit, clears the Interrupt ($\overline{INT}$ goes HIGH) and allows the serial transfer to continue.

When the slave address with the direction bit and part of or all the following bytes have been transmitted, the Serial Interrupt flag (SI) is set again, the Interrupt line ($\overline{INT}$) goes LOW again and I2CSTA is loaded with the following possible codes:

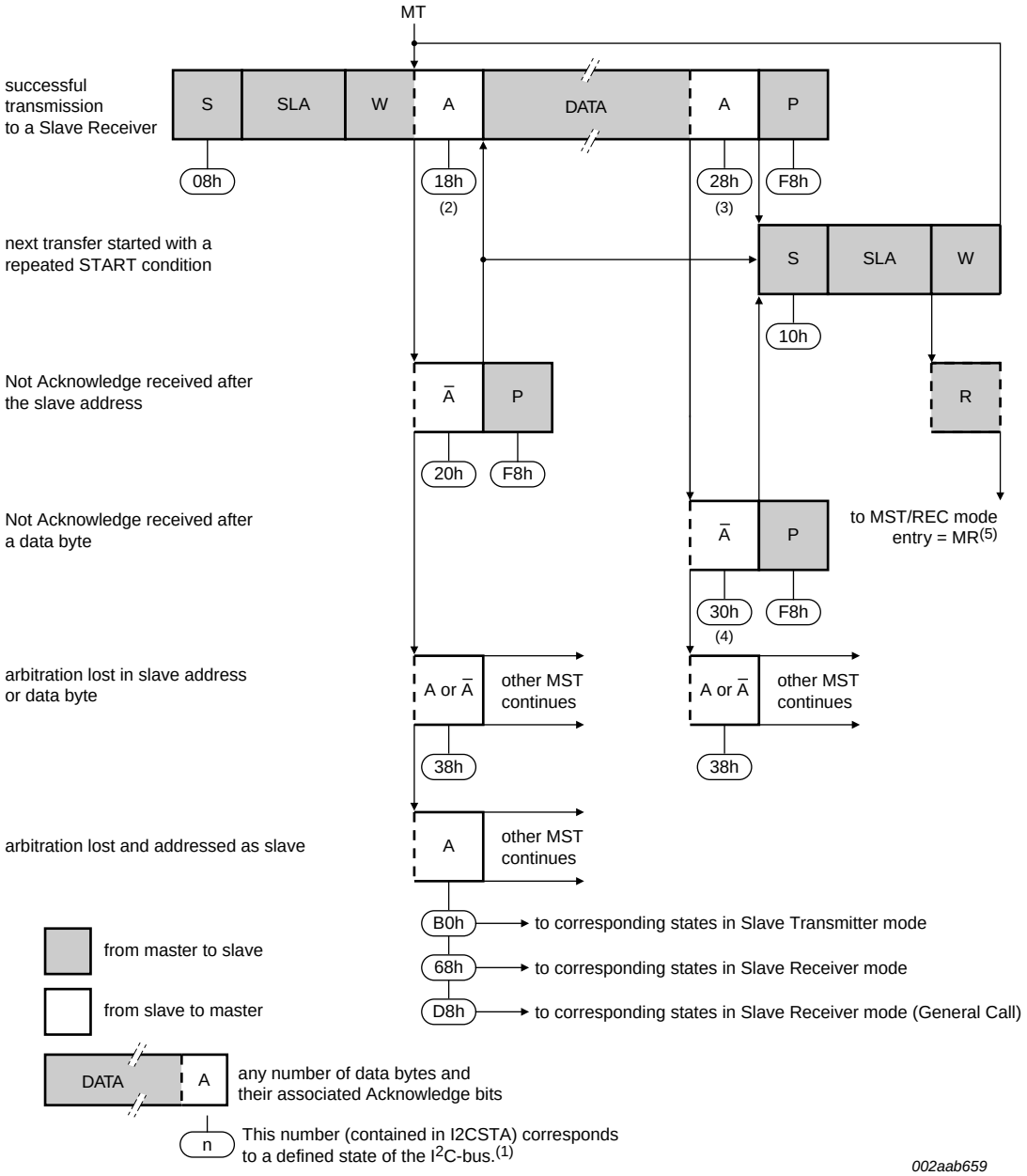
- 18h if an acknowledgment bit (ACK) has been received for the slave address with direction bit (happens only if I2CCOUNT = 1; no data bytes have been sent).
- 20h if a no acknowledgment bit (NACK) has been received for the slave address with direction bit (no data bytes have been sent).
- 28h if the slave address with direction bit and all the data bytes have been transmitted and an acknowledgement bit has been received for each of them (number of bytes sent is equal to value in I2CCOUNT).

- 30h if the slave address with direction bit has been successfully sent and no acknowledgement (NACK) has been received while transmitting the data bytes (number of total bytes sent is lower than or equal to value in I2CCOUNT).
- 38h if the PCA9665/PCA9665A lost the arbitration when sending the slave address with the direction bit or when sending data bytes.
- B0h if the PCA9665/PCA9665A lost the arbitration and is addressed as a slave transmitter (slave mode enabled with AA = 1).
- 68h if the PCA9665/PCA9665A lost the arbitration and is addressed as a slave receiver (slave mode enabled with AA = 1).
- D8h if the PCA9665/PCA9665A lost the arbitration and is addressed as a slave receiver during a General Call sequence (slave mode enabled with AA = 1 and General Call address enabled with GC = 1 in I2CADR register).

The appropriate action to be taken for each of these status codes is detailed in Table 35. ENSIO is not affected by the serial transfer and is not referred to in Table 35.

After a repeated START condition (state 10h), the PCA9665/PCA9665A may switch to the Master Receiver mode by loading I2CDAT with SLA+R).

Remark: A master should not transmit its own slave address.



(1) See Table 35

(2) Serial interrupt that occurs when BC[6:0] = 01. No serial interrupt if BC[6:0] > 01.

(3) Defined state when the number of bytes sent is equal to the value in I2CCOUNT register and an ACK has been received for all the bytes sent.

(4) Defined state when a NACK received while number of bytes sent is lower than or equal to value in I2CCOUNT register.

(5) Master Receiver Byte mode is entered when MODE = 0. Master Receiver Buffered mode is entered when MODE = 1.

Remark: The master should never transmit its own slave address.

Fig 10. Format and states in the Master Transmitter Buffered mode (MODE = 1)

Table 35. Master Transmitter Buffered mode (MODE = 1)

Status code (I2CSTA)	Status of the I²C-bus and the PCA9665/65A	Application software response								Next action taken by the PCA9665/PCA9665A
		To/from I2CDAT	To I2CCOUNT		To I2CCON					
			LB	BC[6:0]	STA	STO	SI	AA	MODE	
08h	A START condition has been transmitted	Load SLA+W and the data bytes	X	Total number of bytes to be transmitted (= SLA+W + number of data bytes)	X	X	0	X	1	SLA+W will be transmitted. If ACK bit received, data bytes will be transmitted until all of them have been sent and an ACK has been received for each of them or until a NACK bit is received.
10h	A repeated START condition has been transmitted	Load SLA+W and the data bytes or	X	Total number of bytes to be transmitted (= SLA+W + number of data bytes)	X	X	0	X	1	SLA+W will be transmitted. If ACK bit received, data bytes will be transmitted until all of them have been sent and an ACK has been received for each of them or until a NACK bit is received.
		Load SLA+R	X	Total number of bytes to be received	X	X	0	X	1	SLA+R will be transmitted. PCA9665/PCA9665A will be switched to Master Receiver Buffered mode.
18h	SLA+W has been transmitted; ACK has been received	Load the data bytes or	X	Total number of data bytes to be transmitted	0	0	0	X	1	Up to BC[6:0] data bytes will be transmitted (until all of them have been sent and an ACK has been received for each of them or until a NACK bit is received).
		no I2CDAT action or	X	X	1	0	0	X	1	Repeated START will be transmitted.
		no I2CDAT action or	X	X	0	1	0	X	1	STOP condition will be transmitted. STO flag will be reset.
		no I2CDAT action	X	X	1	1	0	X	1	STOP condition followed by a START condition will be transmitted. STO flag will be reset.
20h	SLA+W has been transmitted; NACK has been received	Load the data bytes or	0	Total number of data bytes to be transmitted	0	0	0	X	1	Up to BC[6:0] data bytes will be transmitted (until all of them have been sent and an ACK has been received for each of them or until a NACK bit is received).
		no I2CDAT action or	1	X	1	0	0	X	1	Repeated START will be transmitted.
		no I2CDAT action or	0	X	0	1	0	X	1	STOP condition will be transmitted;. STO flag will be reset.
		no I2CDAT action	1	X	1	1	0	X	1	STOP condition followed by a START condition will be transmitted. STO flag will be reset.

Table 35. Master Transmitter Buffered mode (MODE = 1) *…continued*

Status code (I2CSTA)	Status of the I²C-bus and the PCA9665/65A	Application software response								Next action taken by the PCA9665/PCA9665A
		To/from I2CDAT	To I2CCOUNT		To I2CCON					
			LB	BC[6:0]	STA	STO	SI	AA	MODE	
28h	BC[6:0] bytes in I2CDAT have been transmitted; ACK has been received for all of them	Load the data bytes or	X	Total number of data bytes to be transmitted	0	0	0	X	1	Up to BC[6:0] data bytes will be transmitted (until all of them have been sent and an ACK has been received for each of them or until a NACK bit is received).
		no I2CDAT action or	X	X	1	0	0	X	1	Repeated START will be transmitted.
		no I2CDAT action or	X	X	0	1	0	X	1	STOP condition will be transmitted. STO flag will be reset.
		no I2CDAT action	X	X	1	1	0	X	1	TOP condition followed by a START condition will be transmitted. STO flag will be reset.
30h	Up to BC[6:0] bytes in I2CDAT have been transmitted; NACK has been received for the last byte	Load the data bytes or	X	Total number of data bytes to be transmitted	0	0	0	X	1	Up to BC[6:0] data bytes will be transmitted (until all of them have been sent and an ACK has been received for each of them or until a NACK bit is received).
		no I2CDAT action or	X	X	1	0	0	X	1	Repeated START will be transmitted.
		no I2CDAT action or	X	X	0	1	0	X	1	STOP condition will be transmitted. STO flag will be reset.
		no I2CDAT action	X	X	1	1	0	X	1	STOP condition followed by a START condition will be transmitted. STO flag will be reset.
38h	Arbitration lost in SLA+W or Data bytes	No I2CDAT action or	X	X	0	0	0	0	1	I²C-bus will be released; PCA9665/PCA9665A will enter the not addressed slave mode.
		No I2CDAT action or	X	X	0	0	0	1	1	I²C-bus will be released; PCA9665/PCA9665A will enter the slave mode.
		No I2CDAT action	X	X	1	0	0	X	1	A START condition will be transmitted when the bus becomes free.

8.4.2 Master Receiver Buffered mode

In the Master Receiver Buffered mode, a number of data bytes are received from a slave transmitter several bytes at a time (see Figure 11). The transfer is initialized as in the Master Transmitter Byte mode.

The Master Receiver Buffered mode may now be entered by setting the STA bit. The I²C-bus state machine will first test the I²C-bus and generate a START condition as soon as the bus becomes free. When a START condition is transmitted, the Serial Interrupt flag (SI) is set, the Interrupt line ($\overline{INT}$) goes LOW and the status code in the status register (I2CSTA) will be 08h. This status code must be used to vector to an interrupt service routine that loads I2CDAT with the slave address and the data direction bit (SLA+R). The byte count register (I2CCOUNT) needs to be programmed with the number of bytes that need to be received in a single sequence (BC[6:0]). LB bit is programmed with logic 0 if the last received byte needs to be acknowledged (read operation still ongoing) or with logic 1 if the last received byte needs to be not acknowledged (read operation ends so the PCA9665/PCA9665A can issue a STOP or Re-START condition). A write to I2CCON resets the SI bit, clears the Interrupt ($\overline{INT}$ goes HIGH) and allows the serial transfer to continue.

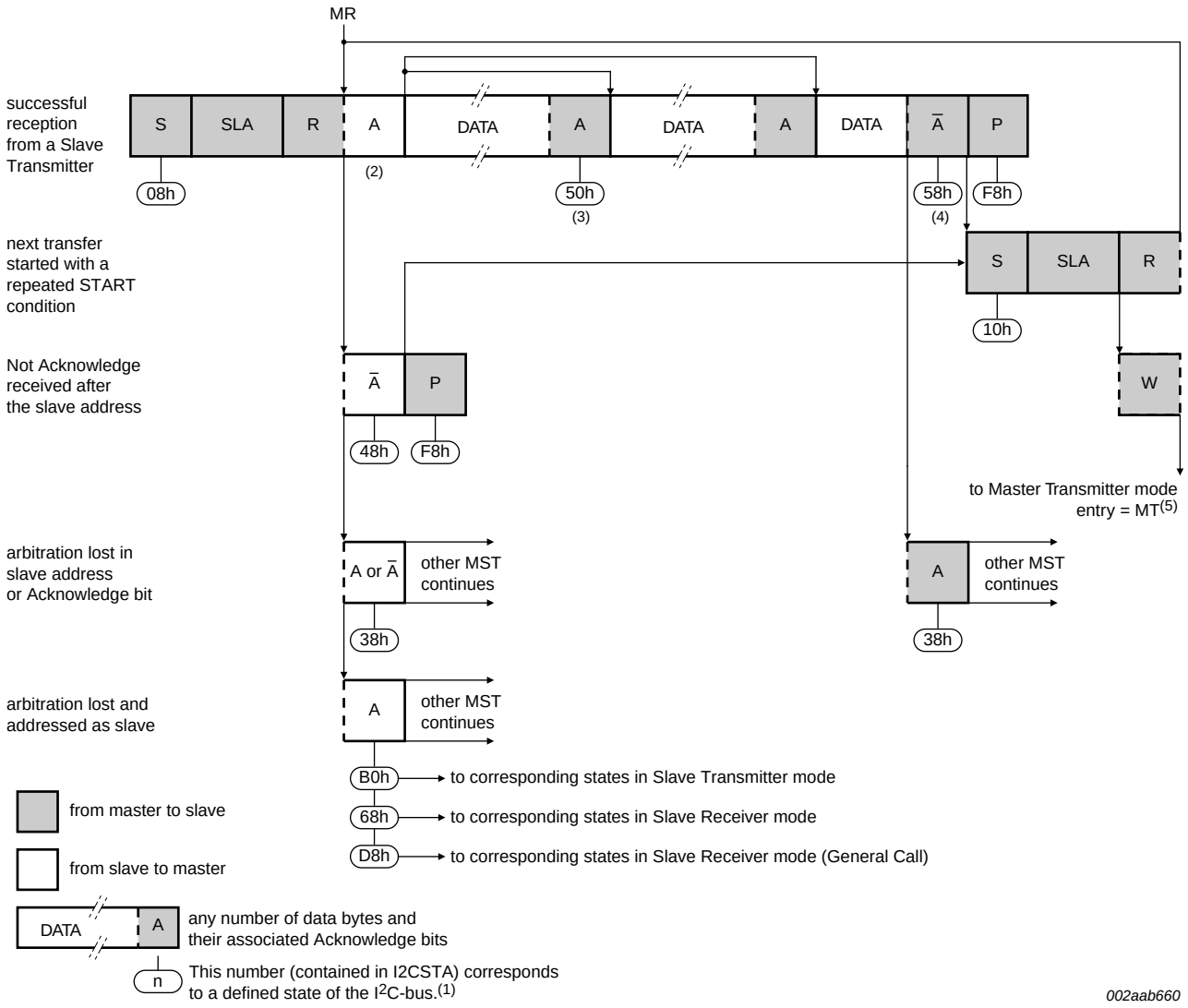
When the slave address and the data direction bit have been transmitted and all the data bytes have been received, the Serial Interrupt flag (SI) is set again, the Interrupt line ($\overline{INT}$) goes LOW again and I2CSTA is loaded with the following possible codes:

- 48h if a no acknowledgment bit (NACK) has been received for the slave address with direction bit
- 50h when all the bytes have been received and an acknowledgement bit (ACK) has been returned for all the bytes
- 58h when all the bytes have been received and an acknowledgement bit (ACK) has been returned for all the bytes except the last one
- 38h if the PCA9665/PCA9665A lost the arbitration
- B0h if the PCA9665/PCA9665A lost the arbitration and is addressed as a slave transmitter (slave mode enabled with AA = 1)
- 68h if the PCA9665/PCA9665A lost the arbitration and is addressed as a slave receiver (slave mode enabled with AA = 1)
- D8h if the PCA9665/PCA9665A lost the arbitration and is addressed as a slave receiver during a General Call sequence (slave mode enabled with AA = 1 and General Call address enabled with GC = 1 in I2CADR register).

The appropriate action to be taken for each of these status codes is detailed in Table 36. ENSIO is not affected by the serial transfer and is not referred to in Table 36.

After a repeated START condition (state 10h), the PCA9665 may switch to the Master Transmitter mode by loading I2CDAT with SLA+W.

Remark: A master should not transmit its own slave address.



(1) See Table 28.

(2) No serial interrupt.

(3) Defined state when LB = 0 and the number of bytes received is equal to the value in I2CCOUNT register.

(4) Defined state when LB = 1 and the number of bytes received is equal to the value in I2CCOUNT register.

(5) Master Transmitter Byte mode is entered with MODE = 0. Master Transmitter Buffered mode is entered when MODE = 1.

Fig 11. Format and states in the Master Receiver Buffered mode (MODE = 1)

Table 36. Master Receiver Buffered mode (MODE = 1)

Status code (I2CSTA)	Status of the I²C-bus and the PCA9665/65A	Application software response								Next action taken by the PCA9665/PCA9665A
		To/from I2CDAT	To/from I2CCOUNT		To I2CCON					
			LB	BC[6:0]	STA	STO	SI	AA	MODE	
08h	A START condition has been transmitted	Load SLA+R	0	Total number of bytes to be received	X	X	0	X	1	SLA+R will be transmitted. If ACK bit received, BC[6:0] data bytes will be received, ACK bit will be returned for all of them.
			1	Total number of bytes to be received	X	X	0	X	1	SLA+R will be transmitted. If ACK bit received, BC[6:0] data bytes will be received, ACK bit will be returned for all of them, except for the last one where NACK bit will be returned.
10h	A repeated START condition has been transmitted	Load SLA+R or	0	Total number of bytes to be received	X	X	0	X	1	SLA+R will be transmitted. If ACK bit received, BC[6:0] data bytes will be received, ACK bit will be returned for all of them.
			1	Total number of bytes to be received	X	X	0	X	1	SLA+R will be transmitted. If ACK bit received, BC[6:0] data bytes will be received, ACK bit will be returned for all of them, except for the last one where NACK bit will be returned.
		Load SLA+W and the data bytes	X	Total number of bytes to be transmitted (= SLA+W + number of data bytes)	X	X	0	X	1	SLA+W will be transmitted; PCA9665/PCA9665A will be switched to Master Transmitter Buffered mode.
38h	Arbitration lost in NACK bit	No I2CDAT action or	X	X	0	0	0	X	1	I²C-bus will be released; PCA9665/PCA9665A will enter slave mode.
		No I2CDAT action	X	X	1	0	0	X	1	A START condition will be transmitted when the bus becomes free.
48h	SLA+R has been transmitted; NACK has been received	No I2CDAT action or	X	X	1	0	0	X	1	Repeated START condition will be transmitted.
		No I2CDAT action or	X	X	0	1	0	X	1	STOP condition will be transmitted; STO flag will be reset.
		No I2CDAT action	X	X	1	1	0	X	1	STOP condition followed by a START condition will be transmitted; STO flag will be reset.

Table 36. Master Receiver Buffered mode (MODE = 1) *…continued*

Status code (I2CSTA)	Status of the I²C-bus and the PCA9665/65A	Application software response							Next action taken by the PCA9665/PCA9665A
		To/from I2CDAT	To/from I2CCOUNT		To I2CCON				
			LB	BC[6:0]	STA	STO	SI	AA	MODE	
50h	BC[6:0] data bytes have been received; ACK has been returned for all the bytes	Read data bytes or	0	Total number of bytes to be received	0	0	0	X	1	BC[6:0] data bytes will be received, ACK bit will be returned for all of them
		Read data bytes or	1	Total number of bytes to be received	0	0	0	X	1	BC[6:0] data bytes will be received, ACK bit will be returned for all of them, except for the last one where NACK bit will be returned
58h	BC[6:0] data bytes have been received; ACK has been returned for all the bytes, except for the last one where NACK bit has been returned	Read data bytes or	X	X	1	0	0	X	1	Repeated START condition will be transmitted
		Read data bytes or	X	X	0	1	0	X	1	STOP condition will be transmitted; STO flag will be reset.
		Read data bytes	X	X	1	1	0	X	1	STOP condition followed by a START condition will be transmitted; STO flag will be reset.

8.4.3 Slave Receiver Buffered mode

In the Slave Receiver Buffered mode, a number of data bytes are received from a master transmitter several bytes at a time (see Figure 12). To initiate the Slave Receiver Byte mode, I2CADR and I2CCON must be loaded as shown in Table 37 and Table 38.

Table 37. I2CADR initialization

Bit	7	6	5	4	3	2	1	0
Symbol	AD7	AD6	AD5	AD4	AD3	AD2	AD1	GC
Value	own slave address							X

The upper 7 bits are the I²C-bus address to which PCA9665/PCA9665A will respond when addressed by a master. GC is the control bit that allows the PCA9665/PCA9665A to respond or not to the General Call address (00h).

When programmed to logic 1, the PCA9665/PCA9665A will acknowledge the General Call address.

When programmed to logic 0, the PCA9665/PCA9665A will not acknowledge the General Call address.

Table 38. I2CCON initialization

Bit	7	6	5	4	3	2	1	0
Symbol	AA	ENSIO	STA	STO	SI	-	-	MODE
Value	1	1	0	0	0	X	X	1

Table 39. I2CCOUNT programming

Bit	7	6	5	4	3	2	1	0
Symbol	LB	BC6	BC5	BC4	BC3	BC2	BC1	BC0
Value	X	number of bytes received in a single sequence (1 byte to 68 bytes)						

ENSIO must be set to logic 1 to enable the I²C-bus interface. The AA bit must be set to enable the PCA9665/PCA9665A to acknowledge its own slave address; STA, STO, and SI must be reset.

When I2CADR and I2CCON have been initialized, the PCA9665/PCA9665A waits until it is addressed by its own slave address followed by the data direction bit which must be ‘0’ (W) to operate in the Slave Receiver mode. After its own slave address and the W bit have been received, the Serial Interrupt flag (SI) is set, the Interrupt line ($\overline{\text{INT}}$) goes LOW and I2CSTA is loaded with 60h. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken is detailed in Table 40.

The Slave Receiver Buffered mode may also be entered when:

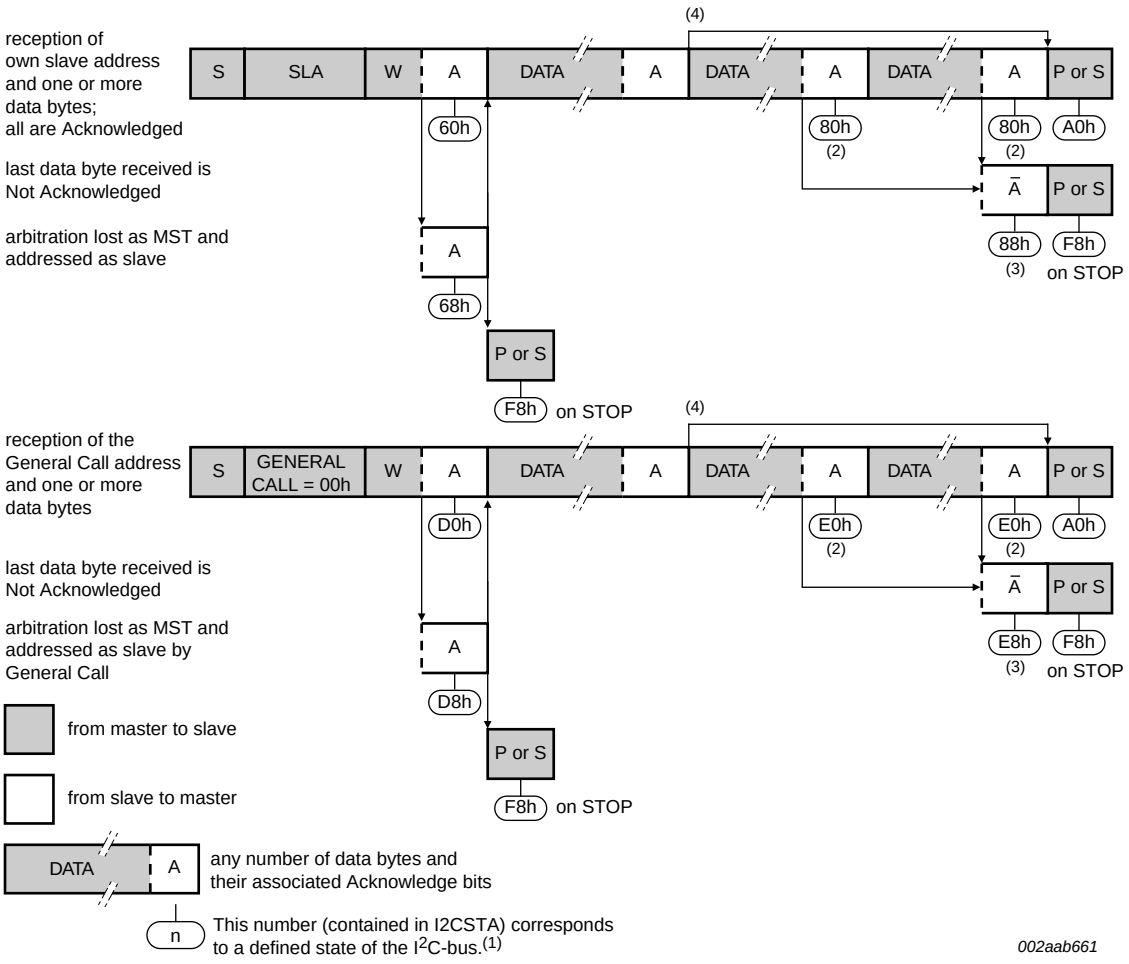
- The arbitration is lost while the PCA9665/PCA9665A is in the master mode. See status 68h and D8h.
- The General Call Address (00h) has been received (General Call address enabled with GC = 1). See status D0h.

Appropriate actions to be taken from these status codes are also detailed in Table 40.

The byte count register (I2CCOUNT) is programmed with the number of bytes that need to be sent in a single sequence (BC[6:0]) as shown in Table 39.

If the LB bit is reset (logic 0), the PCA9665/PCA9665A will return an acknowledge for all the bytes that will be received. The maximum number of bytes that are received in a single sequence is defined by BC[6:0] in I2CCOUNT register as shown in Table 39.

If the LB bit is set (logic 1) during a transfer, the PCA9665/PCA9665A will return a not acknowledge (logic 1) on SDA after receiving the last byte. If the AA bit is reset, the I²C-bus state machine does not respond to its own slave address. However, the I²C-bus is still monitored and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate the PCA9665/PCA9665A from the I²C-bus.



(1) See Table 40.

(2) Defined state when the number of bytes received is equal to the value in I2CCOUNT register and LB = 0.

(3) Defined state when the number of bytes received is equal to the value in I2CCOUNT register and LB = 1.

(4) Number of bytes received is lower than I2CCOUNT.

Fig 12. Format and states in the Slave Receiver Buffered mode (MODE = 1)

Table 40. Slave Receiver Buffered mode (MODE = 1)

Status code (I2CSTA)	Status of the I²C-bus and the PCA9665/65A	Application software response								Next action taken by the PCA9665/PCA9665A
		To/from I2CDAT	To/from I2CCOUNT		To I2CCON					
			LB	BC[6:0]	STA	STO	SI	AA	MODE	
60h	Own SLA+W has been received; ACK has been returned	No I2CDAT action or	0	Total number of bytes to be received	X	X	0	X	1	Up to BC[6:0] data bytes will be received, ACK bit will be returned for all of them.
		No I2CDAT action	1	Total number of bytes to be received	X	X	0	X	1	Up to BC[6:0] data bytes will be received, ACK bit will be returned for all of them, except for the last one where NACK bit will be returned (unless master transmitter sends a STOP or Repeated START condition before).
68h	Arbitration lost in SLA+R/W as master; Own SLA+W has been received; ACK has been returned	No I2CDAT action or	0	Total number of bytes to be received	X	X	0	X	1	Up to BC[6:0] data bytes will be received, ACK bit will be returned for all of them.
		No I2CDAT action	1	Total number of bytes to be received	X	X	0	X	1	Up to BC[6:0] data bytes will be received, ACK bit will be returned for all of them, except for the last one where NACK bit will be returned (unless master transmitter sends a STOP or Repeated START condition before).
D0h	General Call address (00h) has been received; ACK has been returned.	No I2CDAT action or	0	Total number of bytes to be received	X	X	0	X	1	Up to BC[6:0] data bytes will be received, ACK bit will be returned for all of them.
		No I2CDAT action	1	Total number of bytes to be received	X	X	0	X	1	Up to BC[6:0] data bytes will be received, ACK bit will be returned for all of them, except for the last one where NACK bit will be returned (unless master transmitter sends a STOP or Repeated START condition before).
D8h	Arbitration lost in SLA = R/W as master; General Call address has been received; ACK bit has been returned.	No I2CDAT action or	0	Total number of bytes to be received	X	X	0	X	1	Up to BC[6:0] data bytes will be received, ACK bit will be returned for all of them.
		No I2CDAT action	1	Total number of bytes to be received	X	X	0	X	1	Up to BC[6:0] data bytes will be received, ACK bit will be returned for all of them, except for the last one where NACK bit will be returned (unless master transmitter sends a STOP or Repeated START condition before).

Table 40. Slave Receiver Buffered mode (MODE = 1) *…continued*

Status code (I2CSTA)	Status of the I²C-bus and the PCA9665/65A	Application software response							Next action taken by the PCA9665/PCA9665A
		To/from I2CDAT	To/from I2CCOUNT		To I2CCON				
			LB	BC[6:0]	STA	STO	SI	AA	MODE	
80h	Previously addressed with own slave address; BC[6:0] data bytes have been received; ACK has been returned for all the bytes	Read data bytes or	0	Total number of bytes to be received	X	X	0	X	1	Up to BC[6:0] data bytes will be received, ACK bit will be returned for all of them.
		Read data bytes	1	Total number of bytes to be received	X	X	0	X	1	Up to BC[6:0] data bytes will be received, ACK bit will be returned for all of them, except for the last one where NACK bit will be returned (unless master transmitter sends a STOP or Repeated START condition before).
88h	Previously addressed with own slave address; BC[6:0] data bytes have been received; ACK has been returned for all the bytes, except for the last one where NACK bit has been returned	Read data bytes or	X	X	0	X	0	0	1	Switched to not addressed slave mode; No recognition of own slave address; General Call address will be recognized if GC = 1.
		Read data bytes or	X	X	0	X	0	1	1	Switched to not addressed slave mode; Own slave address will be recognized; General Call address will be recognized if GC = 1.
		Read data bytes or	X	X	1	X	0	0	1	Switched to not addressed slave mode; No recognition of own slave address; General Call address will be recognized if GC = 1; A START condition will be transmitted when the bus becomes free.
		Read data bytes	X	X	1	X	0	1	1	Switched to not addressed slave mode; Own slave address will be recognized; General Call address will be recognized if GC = 1; A START condition will be transmitted when the bus becomes free.
E0h	Previously addressed with General Call; BC[6:0] data bytes have been received; ACK has been returned for all the bytes	Read data bytes or	0	Total number of bytes to be received	X	X	0	X	1	BC[6:0] data bytes will be received, ACK bit will be returned for all of them.
		Read data bytes	1	Total number of bytes to be received	X	X	0	X	1	BC[6:0] data bytes will be received, ACK bit will be returned for all of them, except for the last one where NACK bit will be returned (unless master transmitter sends a STOP or Repeated START condition before).

Table 40. Slave Receiver Buffered mode (MODE = 1) *…continued*

Status code (I2CSTA)	Status of the I²C-bus and the PCA9665/65A	Application software response							Next action taken by the PCA9665/PCA9665A
		To/from I2CDAT	To/from I2CCOUNT		To I2CCON				
			LB	BC[6:0]	STA	STO	SI	AA	MODE	
E8h	Previously addressed with General Call; BC[6:0] data bytes have been received; ACK has been returned for all the bytes, except for the last one where NACK bit has been returned	Read data bytes or	X	X	0	X	0	0	1	Switched to not addressed slave mode; No recognition of own slave address; General Call address will be recognized if GC = 1
		Read data bytes or	X	X	0	X	0	1	1	Switched to not addressed slave mode; Own slave address will be recognized; General Call address will be recognized if GC = 1
		Read data bytes or	X	X	1	X	0	0	1	Switched to not addressed slave mode; No recognition of own slave address; General Call address will be recognized if GC = 1; A START condition will be transmitted when the bus becomes free.
		Read data bytes	X	X	1	X	0	1	1	Switched to not addressed slave mode; Own slave address will be recognized; General Call address will be recognized if GC = 1; A START condition will be transmitted when the bus becomes free.
A0h	A STOP condition or repeated START condition has been received while still addressed as slave receiver	No I2CDAT action or	X	X	0	X	0	0	1	Switched to not addressed slave mode; No recognition of own slave address; General Call address will be recognized if GC = 1
		No I2CDAT action or	X	X	0	X	0	1	1	Switched to not addressed slave mode; Own slave address will be recognized; General Call address will be recognized if GC = 1
		No I2CDAT action or	X	X	1	X	0	0	1	Switched to not addressed slave mode; No recognition of own slave address; General Call address will be recognized if GC = 1; A START condition will be transmitted when the bus becomes free.
		No I2CDAT action	X	X	1	X	0	1	1	Switched to not addressed slave mode; Own slave address will be recognized; General Call address will be recognized if GC = 1; A START condition will be transmitted when the bus becomes free.

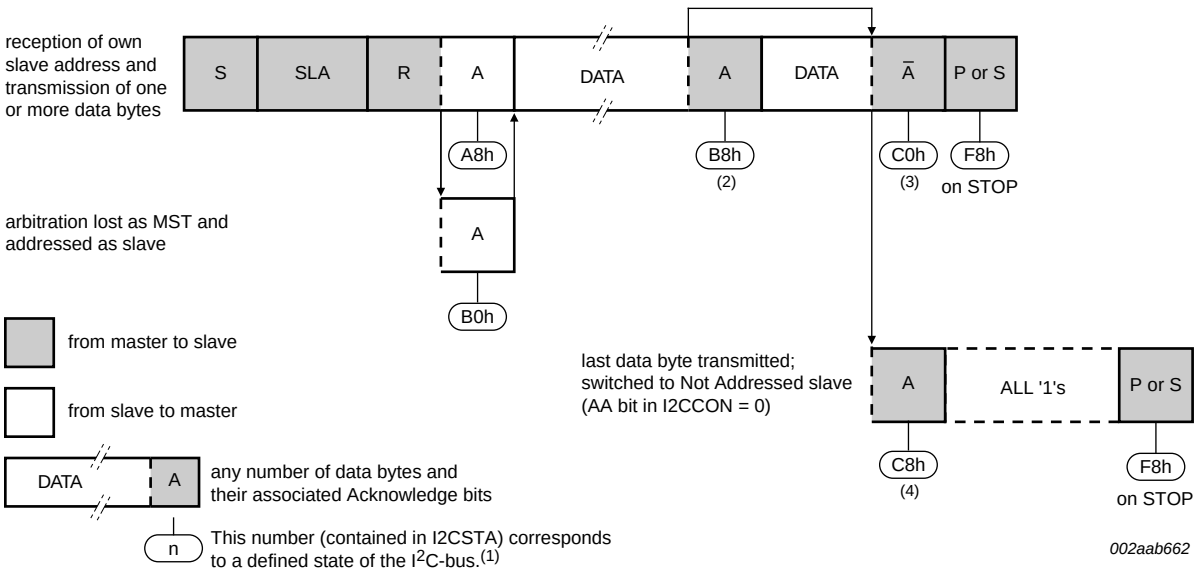
8.4.4 Slave Transmitter Buffered mode

In the Slave Transmitter Buffered mode, a number of data bytes are transmitted to a master receiver several bytes at a time (see Figure 13). Data transfer is initialized as in the Slave Receiver Buffered mode. When I2CADR and I2CCON have been initialized, the PCA9665/PCA9665A waits until it is addressed by its own slave address followed by the data direction bit which must be '1' (R) for the PCA9665/PCA9665A to operate in the Slave Transmitter mode. After its own slave address and the R bit have been received, the Serial Interrupt flag (SI) is set, the Interrupt line ($\overline{INT}$) goes LOW and I2CSTA is loaded with A8h. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken is detailed in Table 41.

The Slave Transmitter Buffered mode may also be entered if arbitration is lost while the PCA9665/PCA9665A is in the master mode. See state B0h and appropriate actions in Table 41.

The byte count register (I2CCOUNT) is programmed with the number of bytes that need to be sent in a single sequence (BC[6:0]) as shown in Table 39. LB bit is only used for the Receiver Buffered modes and can be programmed to either logic 0 or logic 1.

If the AA bit is reset during a transfer, the PCA9665/PCA9665A will transmit all the bytes of the transfer (values defined by BC[6:0]) and enter state C8h. The PCA9665/PCA9665A is switched to the not addressed slave mode and will ignore the master receiver if it continues the transfer. Thus the master receiver receives all '1's as serial data. While AA is reset, the PCA9665/PCA9665A does not respond to its own slave address. However, the I²C-bus is still monitored, and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate the PCA9665/PCA9665A from the I²C-bus.



(1) See Table 31.

(2) Defined state when the number of bytes sent is equal to the value in I2CCOUNT register.

(3) Defined state when a NACK is received. The number of bytes transmitted is lower than or equal to the value in the I2CCOUNT register.

(4) Defined state after the last byte has been transmitted and the PCA9665/PCA9665A goes to the non-addressed mode (AA = 0) and an ACK is received. The number of bytes that are transmitted is equal to the value in I2CCOUNT register.

Fig 13. Format and states in the Slave Transmitter Buffered mode (MODE = 1)

Table 41. Slave Transmitter Buffered mode (MODE = 1)

Status code (I2CSTA)	Status of the I²C-bus and the PCA9665/65A	Application software response								Next action taken by the PCA9665/PCA9665A
		To/from I2CDAT	To/from I2CCOUNT		To I2CCON					
			LB	BC[6:0]	STA	STO	SI	AA	MODE	
A8h	Own SLA+R has been received; ACK has been returned	Load data bytes or	X	Total number of data bytes to be transmitted	X	X	0	0	1	Up to BC[6:0] bytes will be transmitted. PCA9665/PCA9665A switches to the not addressed mode after BC[6:0] bytes have been transmitted.
		Load data bytes	X	Total number of data bytes to be transmitted	X	X	0	1	1	Up to BC[6:0] bytes will be transmitted.
B0h	Arbitration lost in SLA+R/W as master; Own SLA+R has been received, ACK has been returned	Load data bytes or	X	Total number of data bytes to be transmitted	X	X	0	0	1	Up to BC[6:0] bytes will be transmitted. PCA9665/PCA9665A switches to the not addressed mode after BC[6:0] bytes have been transmitted
		Load data bytes	X	Total number of data bytes to be transmitted	X	X	0	1	1	Up to BC[6:0] bytes will be transmitted.
B8h	BC[6:0] bytes in I2CDAT have been transmitted; ACK has been received	Load data bytes or	X	Total number of data bytes to be transmitted	X	X	0	0	1	Up to BC[6:0] bytes will be transmitted. PCA9665/PCA9665A switches to the not addressed mode after BC[6:0] bytes have been transmitted
		Load data bytes	X	Total number of data bytes to be transmitted	X	X	0	1	1	Up to BC[6:0] bytes will be transmitted.
C0h	Up to BC[6:0] bytes in I2CDAT have been transmitted; NACK has been received	No I2CDAT action or	X	X	0	X	0	0	1	Switched to not addressed slave mode; No recognition of own slave address; General Call address recognized if GC = 1
		No I2CDAT action or	X	X	0	X	0	1	1	Switched to slave mode; Own slave address will be recognized; General Call address recognized if GC = 1
		No I2CDAT action or	X	X	1	X	0	0	1	Switched to not addressed slave mode; No recognition of own slave address; General Call address will be recognized if GC = 1; A START condition will be transmitted when the bus becomes free
		No I2CDAT action	X	X	1	X	0	1	1	Switched to slave mode; Own slave address will be recognized; General Call address will be recognized if GC = 1; A START condition will be transmitted when the bus becomes free

Table 41. Slave Transmitter Buffered mode (MODE = 1) *…continued*

Status code (I2CSTA)	Status of the I²C-bus and the PCA9665/65A	Application software response								Next action taken by the PCA9665/PCA9665A
		To/from I2CDAT	To/from I2CCOUNT		To I2CCON					
			LB	BC[6:0]	STA	STO	SI	AA	MODE	
C8h	BC[6:0] bytes in I2CDAT have been transmitted (AA = 0); ACK has been received	No I2CDAT action or	X	X	0	X	0	0	1	Switched to not addressed slave mode; No recognition of own slave address; General Call address recognized if GC = 1.
		No I2CDAT action or	X	X	0	X	0	1	1	Switched to slave mode; Own slave address will be recognized; General Call address recognized if GC = 1.
		No I2CDAT action or	X	X	1	X	0	0	1	Switched to not addressed slave mode; No recognition of own slave address; General Call address will be recognized if GC = 1; A START condition will be transmitted when the bus becomes free.
		No I2CDAT action	X	X	1	X	0	1	1	Switched to slave mode; Own slave address will be recognized; General Call address will be recognized if GC = 1; A START condition will be transmitted when the bus becomes free.

8.5 Buffered mode examples

8.5.1 Buffered Master Transmitter mode of operation

1. Program the I2CCOUNT register with the number of bytes that need to be sent to the I²C-bus (BC[6:0] has a value from 01h to 44h). LB bit is used for Receiver mode only and can be set to 0 or 1.
2. Load the data bytes in I2CDAT buffer. The different bytes to be sent will be stored in the PCA9665/PCA9665A buffer. There is no protection against writing over a buffer's boundary. If more than 68 bytes are written to the buffer, the data at address 00h will be overwritten. The number of bytes that needs to be loaded in I2CDAT is equal to BC[6:0] in the I2CCOUNT register. The number of data bytes sent is equal to BC[6:0], therefore, if the number of data bytes loaded is greater than BC[6:0], the additional data will not be sent. If the number of data bytes written to the buffer is less than BC[6:0], the PCA9665/PCA9665A will still send out BC[6:0] data bytes.
3. Program I2CCON register to initiate the Master Transmitter Buffered sequence. In Master mode, if STA = 1, a START command is sent. An interrupt will be asserted and the SI bit is set in the I2CCON register after the START has been sent. The I2CSTA register contains the status of the transmission. MODE bit must be set to '1' each time a write to the I2CCON register is performed.
4. After reading the I2CSTA status register, the I2CCON is programmed with STA = 0. That clears the previous Interrupt. If a START command has been previously sent, the first byte loaded into the buffer and sent to the I²C-bus is interpreted as the I²C-bus address + R/$\overline{W}$ operation. In transmitter mode, R/$\overline{W}$ = 0 and the following bytes that are sent to the I²C-bus are interpreted as data bytes.
5. When the sequence has been executed, an Interrupt is asserted and the SI bit is set in the I2CCON register. The I2CSTA register contains the status of the transmission and the I2CCOUNT register contains the number of bytes that have been sent to the I²C-bus as described in Table 42.
6. More sequence (program I2CCOUNT register, load data bytes in I2CDAT buffer, write the I2CCON register to send the data to the I²C-bus, read the I2CSTA register when the sequence has been executed) can be performed as long as a STOP or Repeated START command has not been sent. Master Transmitter Buffered mode ends when the I2CCOUNT register is programmed with STO = 1.

8.5.2 Buffered Master Receiver mode of operation

1. Program the I2CCOUNT register with the number of bytes that need to be read from a slave device in the I²C-bus (BC[6:0] has a value from 01h to 44h). LB bit is used in Receiver mode to let the PCA9665/PCA9665A know if the last byte received must be acknowledged or not.

   LB = 0: Last received byte is acknowledged and another sequence can be executed.

   LB = 1: Last received byte is not acknowledged. The last sequence before sending a STOP or Repeated START must be executed with LB = 1.
2. Load the I²C-bus address + R/$\overline{W}$ = 1 in I2CDAT buffer.

3. Program I2CCON register to initiate the Master Receiver Buffered sequence. In Master mode, if STA = 1, a START command is sent. An interrupt will be asserted and the SI bit is set in the I2CCON register after the START has been sent. The I2CSTA register contains the status of the transmission. MODE bit must be set to '1' each time a write to the I2CCON register is performed.
4. After reading the I2CSTA status register, the I2CCON is programmed with STA = 0. That clears the previous Interrupt. If a START command has been previously sent, the I^2C-bus address + R/$\overline{W}$ = 1 byte that has been loaded into the buffer is sent to the I^2C-bus, the PCA9665/PCA9665A then becomes a master receiver device and starts receiving data from the addressed slave device.

   Remark: The PCA9665/PCA9665A is already a master receiver device if a buffered sequence has been previously executed.
5. When the sequence has been executed, an Interrupt is asserted and the SI bit is set in the I2CCON register. The I2CSTA register contains the status of the transmission and the I2CCOUNT register contains the number of bytes that have been received. I2CDAT buffer contains all the data that has been received and can be read by the microcontroller.
6. More sequences (program the I2CCOUNT register, write to the I2CCON register, read the I2CSTA register when sequence has been executed, read the I2CDAT buffer) can be performed as long as a STOP or a Repeated START command has not been sent. To be able to end the reception, the last buffered sequence must be performed with LB = 1. Master Receiver Buffered mode ends when the I2CCOUNT register is programmed with STO = 1.

8.5.3 Buffered Slave Transmitter mode

1. An interrupt is asserted and the SI bit is set in the I2CCON register when the PCA9665/PCA9665A's own slave address has been detected on the I^2C-bus (AA = 1, own slave address defined in the I2CADR register). In Slave Transmitter mode, R/$\overline{W}$ = 1.
2. Program the I2CCOUNT register with the number of bytes that need to be sent to the I^2C-bus (BC[6:0] has a value from 01h to 44h). LB bit is used for Receiver Buffered mode only.
3. Load the data bytes in I2CDAT buffer. The different bytes to be sent will be stored in the PCA9665/PCA9665A buffer. There is no protection against writing over a buffer's boundary. If more than 68 bytes are written to the buffer, the data at address 00h will be overwritten. The number of bytes that needs to be loaded in I2CDAT is equal to BC[6:0] in the I2CCOUNT register. The number of data bytes sent is equal to BC[6:0], therefore, if the number of data bytes loaded is greater than BC[6:0], the additional data will not be sent. If the number of data bytes written to the buffer is less than BC[6:0], the PCA9665/PCA9665A will still send out BC[6:0] data bytes.
4. The I2CCON is programmed to clear the previous Interrupt. The bytes loaded into the buffer are sent to the I^2C-bus. MODE bits must be set to '1' each time a write to the I2CCON register is performed.
5. When the sequence has been executed (BC[6:0] bytes sent or the master sent a NACK), an Interrupt is asserted and the SI bit is set in the I2CCON register. The I2CSTA register contains the status of the transmission and the I2CCOUNT register contains the number of bytes that have been sent to the I^2C-bus.

6. More sequences (program I2CCOUNT register, load data bytes in I2CDAT buffer, write the I2CCON register to send the data to the I²C-bus, read the I2CSTA register when sequence has been executed) can be performed as long as the master acknowledges the bytes sent by the PCA9665/PCA9665A and AA = 1. Slave Transmitter Buffered mode ends when the I²C-bus master does not acknowledge a byte or when the PCA9665/PCA9665A goes to Non-addressed Slave mode.

8.5.4 Buffered Slave Receiver mode

1. An interrupt is asserted and the SI bit is set in the I2CCON register when the PCA9665/PCA9665A's own slave address has been detected in the I²C-bus (AA = 1, own slave address defined in the I2CADR register). In Slave Receiver mode, R/$\overline{W}$ = 0.
2. Program the I2CCOUNT register with the number of bytes that needs to be read from a master device in the I²C-bus (BC[6:0] has a value from 01h to 44h). LB bit is used in Receiver mode to let the PCA9665/PCA9665A know if the last byte received must be acknowledged or not.

   LB = 0: Last received byte is acknowledged and another sequence can be executed.

   LB = 1: Last received byte is not acknowledged.
3. The I2CCON is programmed to clear the previous Interrupt. The PCA9665/PCA9665A receives data from the I²C-bus master. MODE bit must be set to '1' each time a write to the I2CCON register is performed.
4. When the sequence has been executed (BC[6:0] bytes have been received or the master sent a STOP or Repeated START command), an Interrupt is asserted and the SI bit is set in the I2CCON register. The I2CSTA register contains the status of the transmission and the I2CCOUNT register contains the number of bytes that have been received. I2CDAT buffer contains all the data that has been received and can be read by the microcontroller.
5. More sequence (program the I2CCOUNT register, write to the I2CCON register, read the I2CDAT buffer) can be performed as long as a STOP or a Repeated START command has not been sent by the I²C-bus master. Slave Receiver Buffered mode ends when the I²C-bus master sends a STOP or Repeated START command, or when the PCA9665/PCA9665A does not acknowledge the received bytes any more.

8.5.5 Example: Read 128 bytes in two 64-byte sequences of an EEPROM (I²C-bus address = A0h for write operations and A1h for read operations) starting at Location 08h

1. Program I2CCOUNT = 02h (2 bytes to be sent): I²C-bus slave address and memory allocation.
2. Write A0h (I²C-bus slave address and write command) and 08h (Location) into the I2CDAT register.
3. Program I2CCON with STA = 1, STO = SI = 0, MODE = 1.
   - the PCA9665/PCA9665A sends a START command
   - the PCA9665/PCA9665A sends an interrupt, sets SI = 1 and updates I2CSTA register
   - I2CSTA reads 08h

4. Program I2CCON with STA = STO = SI = 0, MODE = 1.
   - I²C-bus slave address A0h, then EEPROM sub address 08h is sent on the bus
   - the SCL line is held LOW by the PCA9665/PCA9665A after the 2 bytes have been sent
   - the PCA9665/PCA9665A sends an Interrupt, sets SI = 1 and updates I2CSTA register
   - I2CSTA reads 28h
5. Program I2CCOUNT = 40h (64 bytes to read and Last byte acknowledged).
6. Load I2CDAT with A1h (I²C-bus slave address and Read command).
7. Program I2CCON with STA = 1, SI = 0, MODE = 1.
   - the PCA9665/PCA9665A sends a ReSTART command
   - an interrupt is asserted and the I2CSTA register is updated
   - the I2CSTA register reads 10h
8. Program I2CCON with STA = STO = SI = 0, MODE = 1.
   - address A1h is sent followed by a read of 64 data bytes
   - the last data byte is acknowledged
   - the SCL line is held LOW by the PCA9665/PCA9665A after the data is read
   - the PCA9665/PCA9665A sends an interrupt and updates I2CSTA register
   - I2CSTA reads 50h
9. The microcontroller reads the 64 data bytes from the PCA9665/PCA9665A.
10. Program I2CCOUNT = C0h (64 bytes and Last byte is not acknowledged).
11. Program I2CCON with STA = STO = SI = 0, MODE = 1.
12. The PCA9665/PCA9665A reads 64 bytes and does not acknowledge the last byte.
    - the PCA9665/PCA9665A sends an Interrupt and updates I2CSTA register
    - the I2CSTA reads 58h
    - the SCL line is held LOW by the PCA9665/PCA9665A
    - the slave should release the SDA line
13. The microcontroller reads the 64 bytes from the PCA9665/PCA9665A.
14. Program I2CCON with SI = STA = 0, ST0 = 1, MODE = X.
    - the PCA9665/PCA9665A sends a STOP condition
    - no interrupt is generated by the PCA9665/PCA9665A
    - the I2CSTA register contains F8h

8.6 I2CCOUNT register

When a write to the I2CCOUNT register is requested, the buffer pointer is reset and points at the first byte. Loading of the data in the I2CDAT buffer then starts at the first byte.

Once an operation has been performed (SI = 1 and an interrupt is generated), the I2CCOUNT register contains the number of bytes that have been received (Receiver mode) or the number of bytes that have been sent (Transmitter mode). See Table 42 for more information.

In Buffered Transmitter mode, the first byte that is sent to the I2C-bus is always the first byte that has been loaded in the I2CDAT buffer.

In Buffered Receiver mode, when an interrupt is generated and SI is set to 1 (after a STOP command or a buffer full condition), the buffer pointer is reset and points at the first received data byte. Reading the I2CCOUNT register then indicates the number of bytes that have been sent or received (BC[6:0]). Reading of the data from I2CDAT buffer can then be initiated starting with the first received byte.

Table 42. I2CCOUNT register value based on the performed operation

Operation performed	I2CCOUNT register value
Master Transmitter Buffered mode	
After START condition	don't care
After Slave Address Sent + ACK bit received and interrupt received	1
After Slave Address Sent + NACK bit received	1
After Slave Address Sent + 'n' data bytes sent, ACK bit received, both address and 'n' data	n + 1
After Slave Address Sent + 'n' data bytes sent, last byte	n + 1
After STOP	don't care
After losing arbitration in Slave Address + W and addressed as slave	0
After losing arbitration in slave address + W and not addressed as slave	0
After losing arbitration in data at n^{th} byte	n (if there was no interrupt after slave address was sent) n – 1 (if there was an interrupt after slave address was sent)
Master Receiver Buffered mode	
After START condition	don't care
After Slave Address Sent + ACK bit received	don't care (because no interrupt received here)
After Slave Address Sent + NACK bit received	1
After Slave Address Sent + 'n' data bytes received, ACK bit received for address and ACK bit returned for 'n' data bytes	n
After Slave Address Sent + 'n' data bytes received, NACK bit returned for the last byte	n
After STOP	don't care
After losing arbitration in Slave Address + R bit and addressed as slave	0
After losing arbitration in slave address + R and not addressed as slave	0
After losing arbitration in ACK of n^{th} byte	n
Slave Receiver Buffered mode (regular slave mode and General Call response	
After Slave Address + W and ACK bit returned for slave address (both in regular mode and when PCA9665/PCA9665A loses arbitration and is addressed as slave)	0
After receiving 'n' bytes, ACK bit returned for the 'n' bytes	n
After receiving 'n' bytes, NACK bit returned for the last byte	n

Table 42. I2CCOUNT register value based on the performed operation *…continued*

Operation performed	I2CCOUNT register value
Slave Transmitter Buffered mode	
After Slave Address + R and ACK bit returned for slave address (both in regular mode and when PCA9665/PCA9665A loses arbitration and is addressed as slave)	0
After ‘n’ data bytes transmitted and ACK bit received for ‘n’ bytes	n
After ‘n’ data bytes transmitted and NACK bit received for the last byte	n

Remark: Request to send or receive a number of bytes equal to 0 or higher than 68 (BC[6:0] = 000 0000 or BC[6:0] > 100 0100) will cause no data to be transferred and an interrupt to be generated after writing to the I2CCON register. I2CSTA status register is loaded with FCh that indicates that an invalid value was requested to be loaded in I2CCOUNT.

8.7 Acknowledge management (I²C-bus addresses and data) in Byte and Buffered modes

Data acknowledge/not acknowledge management can be controlled on a byte basis (Byte mode) or on a sequence basis (Buffered mode). The PCA9665/PCA9665A can be programmed to respond (ACK) or not (NACK) to two different I²C-bus addresses. Table 43 shows how this is performed based on the different control bits (AA, GC, LB and MODE) and the different modes.

Table 43. Own slave address, General Call address, and Data acknowledge management

AA	GC	LB	MODE	Address	Data received[1]
Master mode: the PCA9665/65A generates a START command and controls the I²C-bus					
0	X	X	0	not applicable	data (each byte) = NACK
1	X	X	0	not applicable	data (each byte) = ACK
X	X	0	1	not applicable	all the bytes (BC[6:0] bytes) = ACK
X	X	1	1	not applicable	all the bytes except the last one (BC[6:0] bytes − 1) = ACK; last byte = NACK
Slave mode: I²C-bus message starting with the PCA9665/65A’s Own Slave address					
0	X	X	0	Own address = NACK	data (each byte) = NACK
1	X	X	0	Own address = ACK	data (each byte) = ACK
0	X	0	1	Own address = NACK	all the bytes (≤ BC[6:0] bytes) = ACK
0	X	1	1	Own address = NACK	all the bytes except the last one (BC[6:0] bytes - 1) = ACK; last byte = NACK[2]
1	X	0	1	Own address = ACK	all the bytes (≤ BC[6:0] bytes) = ACK
1	X	1	1	Own address = ACK	all the bytes except the last one (BC[6:0] bytes - 1) = ACK; last byte = NACK[2]

Table 43. Own slave address, General Call address, and Data acknowledge management

AA	GC	LB	MODE	Address	Data received[1]
Slave mode: I²C-bus message starting with the General Call address					
X	0	X	0	GC address = NACK	data (each byte) = NACK
0	1	X	0	GC address = ACK	data (each byte) = NACK
1	1	X	0	GC address = ACK	data (each byte) = ACK
X	0	X	1	GC address = NACK	data (each byte) = NACK
X	1	0	1	GC address = ACK	all the bytes (≤ BC[6:0] bytes) = ACK
X	1	1	1	GC address = ACK	all the bytes except the last one (BC[6:0] bytes - 1) = ACK; last byte = NACK[2]

[1] Assumption is that Data Received follows the address (as defined in column “Address”); valid for slave mode only.

[2] Unless the master sends a STOP command before.

Table 44. Unbuffered Mode (MODE = 0)

Control bits	LB = x	
AA = 0	Master Transmitter mode • address/data are transmitted on a byte basis	Master Receiver mode • address is transmitted and data are received on a byte basis • NACK returned after one byte received
	Slave Transmitter mode • NACK returned after own slave address received • switch to not addressed slave mode any time during an I²C-bus sequence	Slave Receiver mode • NACK returned after own slave address received • NACK returned after one byte received
AA = 1	Master Transmitter mode • address/data are transmitted on a byte basis	Master Receiver mode • data are received on a byte basis • ACK returned after one byte received
	Slave Transmitter mode • ACK returned after own slave address received • always addressed during an I²C-bus sequence	Slave Receiver mode • ACK returned after own slave address received • ACK returned after one byte received

Table 45. Buffered Mode (MODE = 1)

Control bits	LB = 0		LB = 1	
AA = 0	Master Transmitter mode • address/data are transmitted on a multiple byte basis = BC[6:0] value	Master Receiver mode • address is transmitted and data are received on a multiple byte basis = BC[6:0] value • ACK returned after the last byte of a buffered sequence received (after bytes received = BC[6:0] value)	Master Transmitter mode • address/data are transmitted on a multiple byte basis = BC[6:0] value	Master Receiver mode • address is transmitted and data are received on a multiple byte basis = BC[6:0] value • NACK returned after the last byte of a buffered sequence received (after bytes received = BC[6:0] value)
	Slave Transmitter mode • NACK returned after own slave address received • in addressed mode, data are transmitted on a multiple byte basis = BC[6:0] value • in addressed mode, switch to non addressed mode after the last byte of a buffered sequence is transmitted (after bytes sent = BC[6:0] value)	Slave Receiver mode • NACK returned after own slave address received • in addressed mode, data are received on a multiple byte basis = BC[6:0] value • in addressed mode, ACK returned after the last byte of a buffered sequence received (after bytes received = BC[6:0] value) • in addressed mode, switch to non-addressed mode after the last byte of a buffered sequence is received (after bytes received = BC[6:0] value)	Slave Transmitter mode • NACK returned after own slave address received • in addressed mode, data are transmitted on a multiple byte basis = BC[6:0] value • in addressed mode, switch to non addressed mode after the last byte of a buffered sequence is transmitted (after bytes sent = BC[6:0] value)	Slave Receiver mode • NACK returned after own slave address received • in addressed mode, data are received on a multiple byte basis = BC[6:0] value • in addressed mode, NACK returned after the last byte of a buffered sequence received (after bytes received = BC[6:0] value) • in addressed mode, switch to non-addressed mode after the last byte of a buffered sequence is received (after bytes received = BC[6:0] value)

Table 45. Buffered Mode (MODE = 1) *…continued*

Control bits	LB = 0		LB = 1	
AA = 1	Master Transmitter mode • address/data are transmitted on a multiple byte basis = BC[6:0] value	Master Receiver mode • address is transmitted and data are received on a multiple byte basis = BC[6:0] value • ACK returned after the last byte of a buffered sequence received (after bytes received = BC[6:0] value)	Master Transmitter mode • address/data are transmitted on a multiple byte basis = BC[6:0] value	Master Receiver mode • address is transmitted and data are received on a multiple byte basis = BC[6:0] value • NACK returned after the last byte of a buffered sequence received (after bytes received = BC[6:0] value)
	Slave Transmitter mode • ACK returned after own slave address received • in addressed mode, data are transmitted on a multiple byte basis = BC[6:0] value • always addressed during a buffered sequence	Slave Receiver mode • ACK returned after own slave address received • in addressed mode, data are received on a multiple byte basis = BC[6:0] value • in addressed mode, ACK returned after the last byte of a buffered sequence received (after bytes received = BC[6:0] value)	Slave Transmitter mode • ACK returned after own slave address received • in addressed mode, data are transmitted on a multiple byte basis = BC[6:0] value • always addressed during a buffered sequence	Slave Receiver mode • ACK returned after own slave address received • in addressed mode, data are received on a multiple byte basis = BC[6:0] value • in addressed mode, NACK returned after the last byte of a buffered sequence received (after bytes received = BC[6:0] value)

8.8 Miscellaneous states

There are four I2CSTA codes that do not correspond to a defined PCA9665/PCA9665A state (see Table 46). These are discussed in Section 8.8.1 through Section 8.8.4.

Table 46. Miscellaneous states

Status code (I2CSTA)	Status of the I²C-bus and the PCA9665/65A	Application software response						Next action taken by PCA9665/PCA9665A
		To/from I2CDAT	To I2CCON					
			STA	STO	SI	AA	MODE	
F8h	On hardware or software reset or STOP	No I2CDAT action	1	X	0	X	X	Go into master mode; send START
		No I2CDAT action	0	X	0	0	X	No recognition of own slave address. General Call address will be recognized if GC = 1.
		No I2CDAT action	0	X	0	1	X	Will recognize own slave address. General Call address will be recognized if GC = 1.
70h	Bus error SDA stuck LOW	No I2CDAT action	No I2CCON action					Hardware or software reset of the PCA9665/PCA9665A (requires reset to return to state F8h)
78h	Bus error SCL stuck LOW	No I2CDAT action	No I2CCON action					Hardware or software reset of the PCA9665/PCA9665A (requires reset to return to state F8h)
FCh	Illegal value in I2CCOUNT	No I2CDAT action	No I2CCON action					Program a valid value in I2CCOUNT: BC[6:0] between 1 and 68.
00h	Bus error during master or slave mode, due to illegal START or STOP condition	No I2CDAT action	No I2CCON action					Hardware or software reset of the PCA9665/PCA9665A (requires reset to return to state F8h)

8.8.1 I2CSTA = F8h

This status code indicates that the PCA9665/PCA9665A is in an idle state and that no relevant information is available because the serial interrupt flag, SI, is not yet set. This occurs on a STOP condition or during a hardware or software reset event and when the PCA9665/PCA9665A is not involved in a serial transfer.

8.8.2 I2CSTA = 00h

This status code indicates that a bus error has occurred during a serial transfer. A bus error is caused when a START or STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions are during the serial transfer of an address byte, a data byte, or an acknowledge bit. A bus error may also be caused when external interference disturbs the internal PCA9665/PCA9665A signals. When a bus error occurs, SI is set. To recover from a bus error, the microcontroller must send an external hardware or software reset signal to reset the PCA9665/PCA9665A.

8.8.3 I2CSTA = 70h

This status code indicates that the SDA line is stuck LOW when the PCA9665/PCA9665A, in master mode, is trying to send a START condition.

8.8.4 I2CSTA = 78h

This status code indicates that the SCL line is stuck LOW.

8.9 Some special cases

The PCA9665/PCA9665A has facilities to handle the following special cases that may occur during a serial transfer.

8.9.1 Simultaneous repeated START conditions from two masters

A repeated START condition may be generated in the Master Transmitter or Master Receiver modes. A special case occurs if another master simultaneously generates a repeated START condition (see Figure 14). Until this occurs, arbitration is not lost by either master since they were both transmitting the same data.

If the PCA9665/PCA9665A detects a repeated START condition on the I²C-bus before generating a repeated START condition itself, it will use the repeated START as its own and continue with the sending of the slave address.

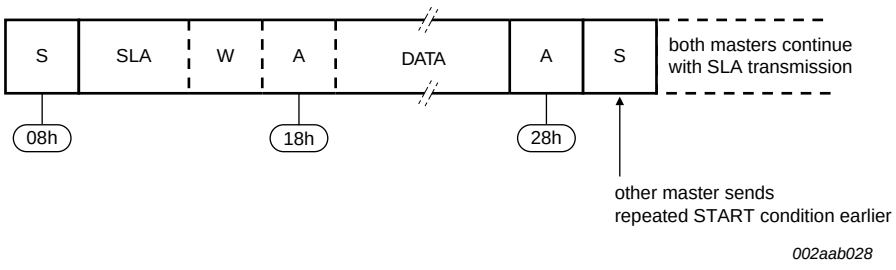


Fig 14. Simultaneous repeated START conditions from 2 masters

8.9.2 Data transfer after loss of arbitration

Arbitration may be lost in the Master Transmitter and Master Receiver modes. Loss of arbitration is indicated by the following states in I2CSTA; 38h, 68h, and B0h (see Figure 6, Figure 10, Figure 7, and Figure 11).

Remark: In order to exit state 38h, a Time-out, Reset, or external STOP are required.

If the STA flag in I2CCON is set by the routines which service these states, then, if the bus is free again, a START condition (state 08h) is transmitted without intervention by the CPU, and a retry of the total serial transfer can commence.

8.9.3 Forced access to the I²C-bus

In some applications, it may be possible for an uncontrolled source to cause a bus hang-up. In such situations, the problem may be caused by interference, temporary interruption of the bus or a temporary short-circuit between SDA and SCL.

If an uncontrolled source generates a superfluous START or masks a STOP condition, then the I²C-bus stays busy indefinitely. If the STA flag is set and bus access is not obtained within a reasonable amount of time, then a forced access to the I²C-bus is possible. If the I²C-bus stays idle for a time period equal to the time-out period, then the PCA9665/PCA9665A concludes that no other master is using the bus and sends a START condition.

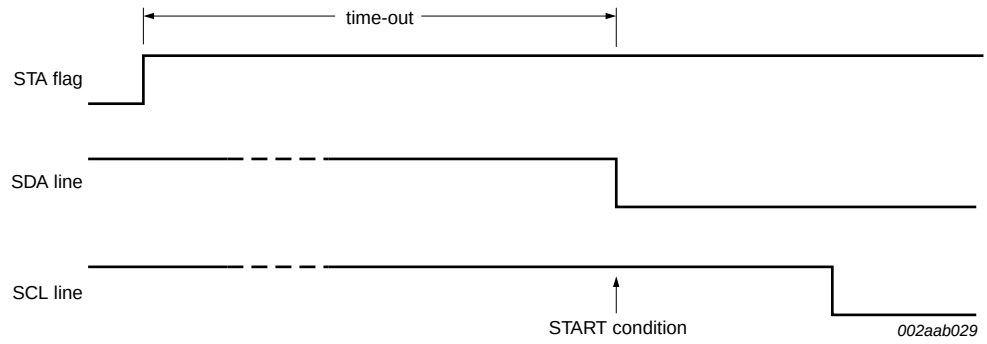


Fig 15. Forced access to a busy I²C-bus

8.9.4 I²C-bus obstructed by a LOW level on SCL or SDA

An I²C-bus hang-up occurs if SDA or SCL is pulled LOW by an uncontrolled source. If the SCL line is obstructed (pulled LOW) by a device on the bus, no further serial transfer is possible, and the PCA9665/PCA9665A cannot resolve this type of problem. When this occurs, the problem must be resolved by the device that is pulling the SCL bus line LOW.

When the SCL line stays LOW for a period equal to the time-out value, the PCA9665/PCA9665A concludes that this is a bus error and behaves in a manner described in Section 7.3.2.4 “The Time-out register, I2CTO (indirect address 04h)”.

If the SDA line is obstructed by another device on the bus (e.g., a slave device out of bit synchronization), the problem can be solved by transmitting additional clock pulses on the SCL line (see Figure 16). The PCA9665/PCA9665A sends out nine clock pulses followed by the STOP condition. If the SDA line is released by the slave pulling it LOW, a normal START condition is transmitted by the PCA9665/PCA9665A, state 08h is entered and the serial transfer continues. If the SDA line is not released by the slave pulling it LOW, then the PCA9665/PCA9665A concludes that there is a bus error, loads 70h in I2CSTA, generates an interrupt signal, and releases the SCL and SDA lines. After the microcontroller reads the status register, it needs to send a reset signal (hardware through the $\overline{\text{RESET}}$ pin, or software through the parallel port) in order to reset the PCA9665/PCA9665A. See Section 8.11 “Reset” for more information.

If a forced bus access occurs or a repeated START condition is transmitted while SDA is obstructed (pulled LOW), the PCA9665/PCA9665A performs the same action as described above. In each case, state 08h is entered after a successful START condition is transmitted and normal serial transfer continues. Note that the CPU is not involved in solving these bus hang-up problems.

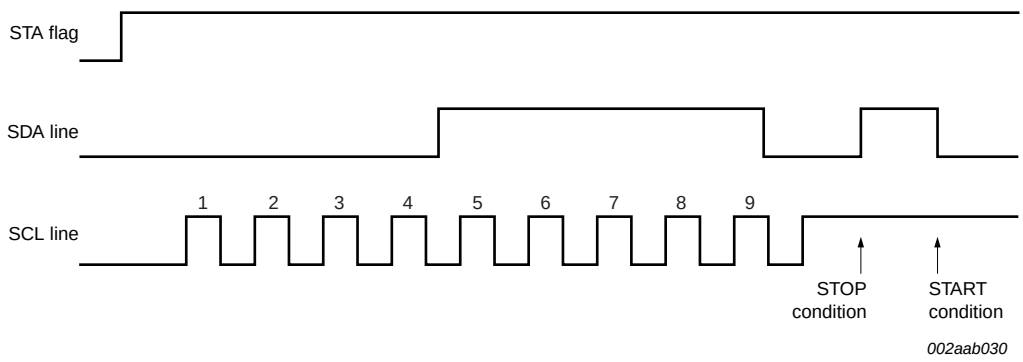


Fig 16. Recovering from a bus obstruction caused by a LOW level on SDA

8.9.5 Bus error

A bus error occurs when a START or STOP condition is present at an illegal position in the format frame. Examples of illegal positions are during the serial transfer of an address byte, a data or an acknowledge bit.

The PCA9665/PCA9665A only reacts to a bus error when it is involved in a serial transfer either as a master or an addressed slave. When a bus error is detected, PCA9665/PCA9665A releases the SDA and SCL lines, sets the interrupt flag, and loads the status register with 00h. This status code may be used to vector to a service routine which either attempts the aborted serial transfer again or simply recovers from the error condition as shown in Table 46 “Miscellaneous states”. The microcontroller must send an external hardware or software reset signal to reset the PCA9665/PCA9665A.

8.10 Power-on reset

When power is applied to V_{DD}, an internal Power-On Reset holds the PCA9665/PCA9665A in a reset condition until V_{DD} has reached V_{POR}. At this point, the reset condition is released and the PCA9665/PCA9665A goes to the power-up initialization phase where the following operations are performed:

1. ENSIO bit is set to 1 to enable the internal oscillator.
2. Internal register initialization is performed.
3. ENSIO bit is set to 0 to disable the internal oscillator and go to the non-addressed low power mode.

The complete power-up initialization phase takes 550 μs to be performed. During this time, write to the PCA9665/PCA9665A through the parallel port is not permitted. However, the parallel port can be read. This allows the device connected to the parallel port of the PCA9665/PCA9665A to poll the I2CCON register and read the ENSIO state bit. When ENSIO bit is equal to 1, this means that the power-up initialization is in progress. When ENSIO is set to 0, this means that the power-up initialization is done and that the PCA9665/PCA9665A is initialized and ready to be used.

8.11 Reset

Reset of the PCA9665/PCA9665A 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 17.

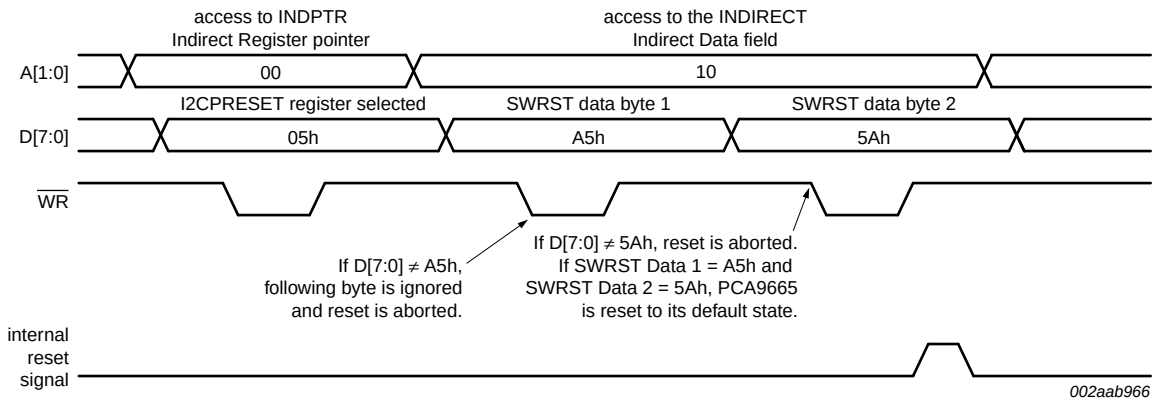


Fig 17. Parallel Software Reset sequence

The $\overline{\text{RESET}}$ hardware pin and software reset function only resets the internal registers and control logic, and does not re-initialize the internal oscillator because the oscillator initialization is performed only on power-up. If the device hangs up and does not respond to a normal $\overline{\text{RESET}}$ input or software reset command, the only way to recover is by powering down and then powering the device back up.

A simple way to implement this circuit without actually having to de-power the entire system is by using a dual gate buffer such as the 74LVC2G125 to control the V_{DD} of PCA9665/PCA9665A as shown in Figure 18. Now, instead of powering the V_{DD} of the PCA9665/PCA9665A directly from the supply rail, it is powered by the output of the 74LVC2G125 with its input connected to the supply rail. Ganging up the two buffers provides twice the drive and minimizes the voltage drop. The 74LVC2G125 enable pins ($1\overline{OE}$, $2\overline{OE}$) are now used to power cycle and recover the PCA9665.

A 100 pF capacitor is used for filtering the supply of PCA9665/PCA9665A and averaging the dynamic current (typically, maximum peak current is 24 mA). Do not size the capacitor too large as the larger capacitor could discharge during power-down, and possibly damage the output of the buffer.

The enable pins are pulled down to ground by a 10 kΩ resistor. During normal operation, the enable pins are held LOW and the buffer is turned on, powering the PCA9665/PCA9665A. An external signal (either from a controller or processor) controls the 74LVC2G125 enable pins to switch on or switch off the supply voltage of the PCA9665/PCA9665A. A HIGH logic level places the buffer in a high-impedance state and turns off the supply to the PCA9665/PCA9665A, which discharges through the 100 pF capacitor. When the enable pins are once again pulled LOW, the PCA9665 powers up and re-initializes to an operation state.

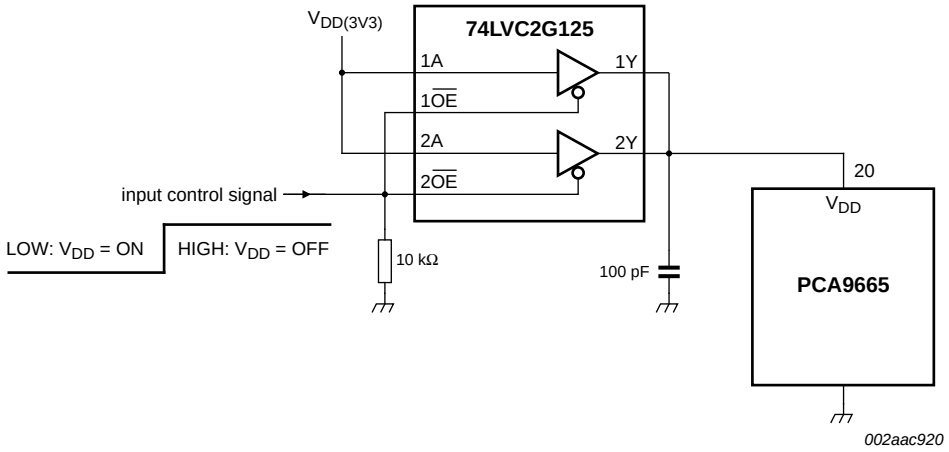
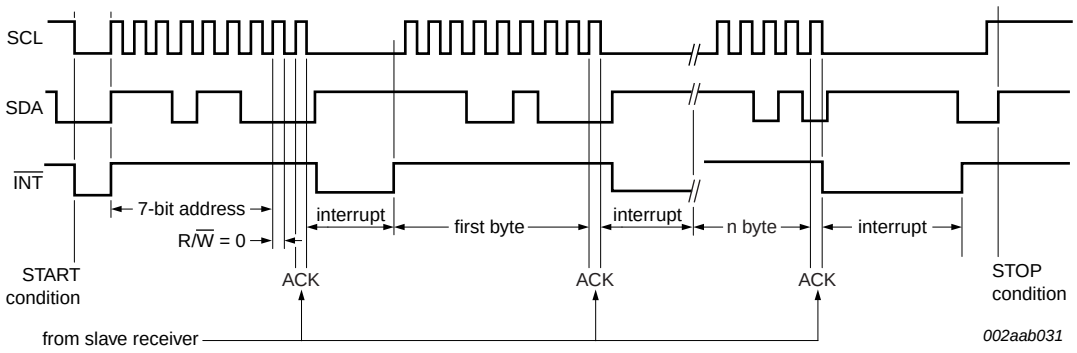


Fig 18. Schematic to power-on/power-off PCA9665/PCA9665A

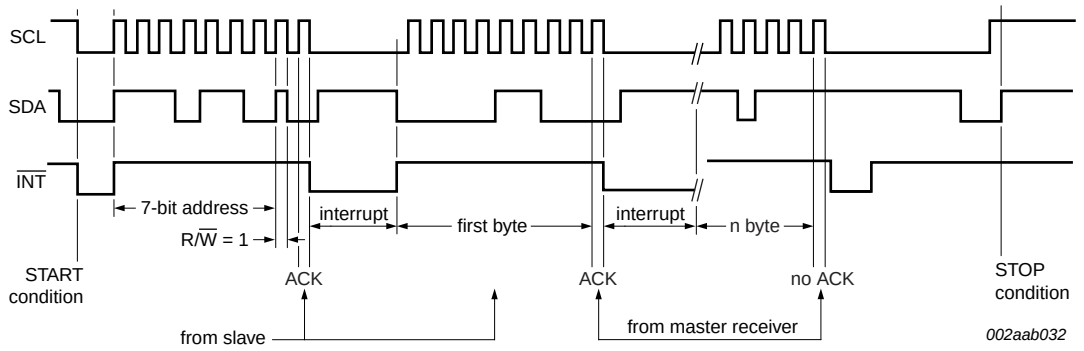
8.12 I²C-bus timing diagrams, Unbuffered mode

The diagrams (Figure 19 through Figure 22) illustrate typical timing diagrams for the PCA9665/PCA9665A in master/slave functions.



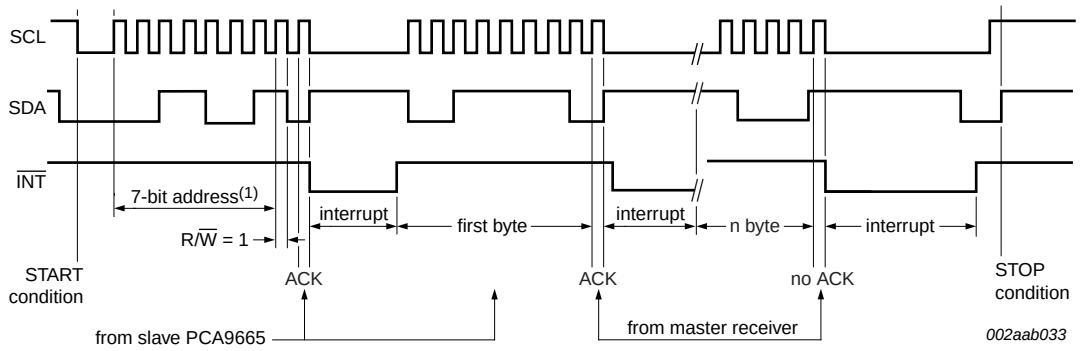
Master PCA9665/PCA9665A writes data to slave transmitter.

Fig 19. Bus timing diagram; Unbuffered Master Transmitter mode



Master PCA9665/PCA9665A reads data from slave transmitter.

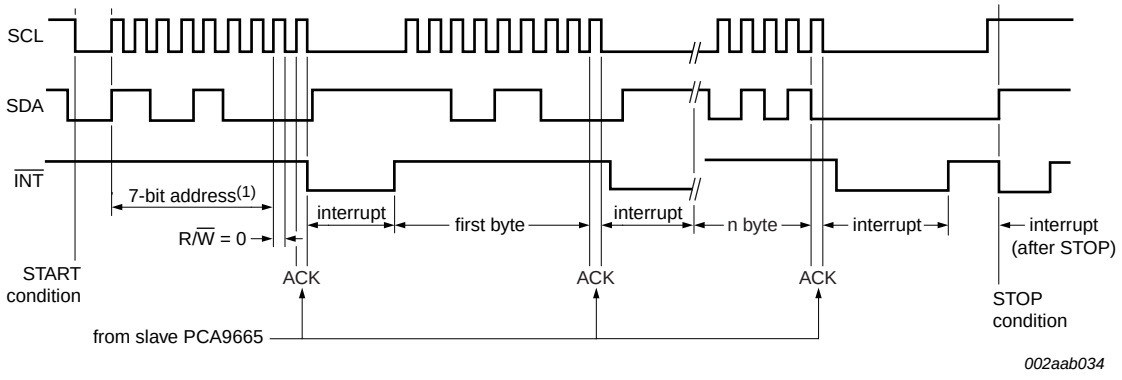
Fig 20. Bus timing diagram; Unbuffered Master Receiver mode



External master receiver reads data from PCA9665/PCA9665A.

(1) As defined in I2CADR register.

Fig 21. Bus timing diagram; Unbuffered Slave Transmitter mode



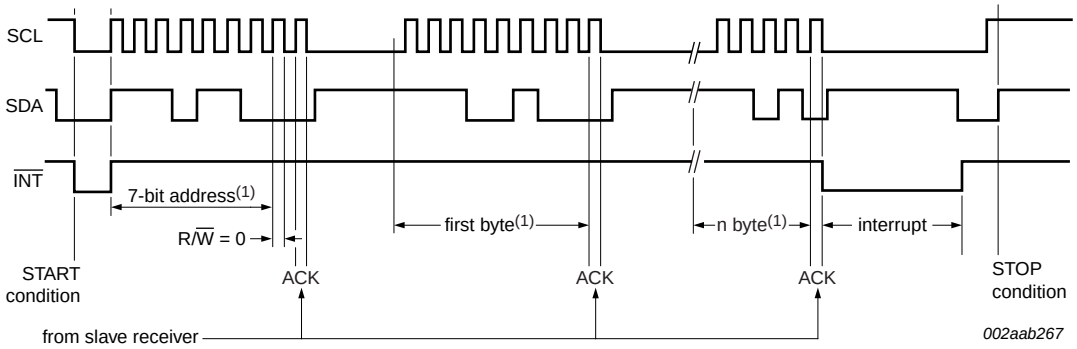
Slave PCA9665/PCA9665A is written to by external master transmitter.

(1) As defined in I2CADR register.

Fig 22. Bus timing diagram; Unbuffered Slave Receiver mode

8.13 I²C-bus timing diagrams, Buffered mode

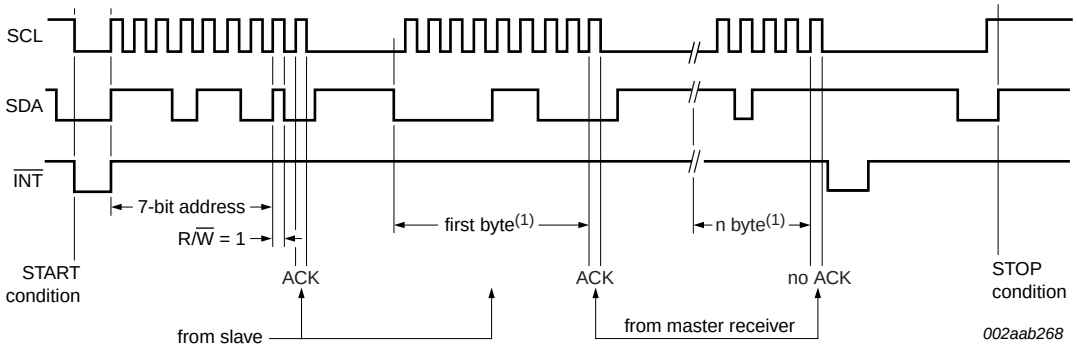
The diagrams (Figure 23 through Figure 26) illustrate typical timing diagrams for the PCA9665/PCA9665A in master/slave functions.



Master PCA9665/PCA9665A writes data to slave transmitter.

(1) 7-bit address + R/$\overline{W}$ = 0 byte and number of bytes sent = value programmed in I2CCOUNT register (BC[6:0] ≤ 68).

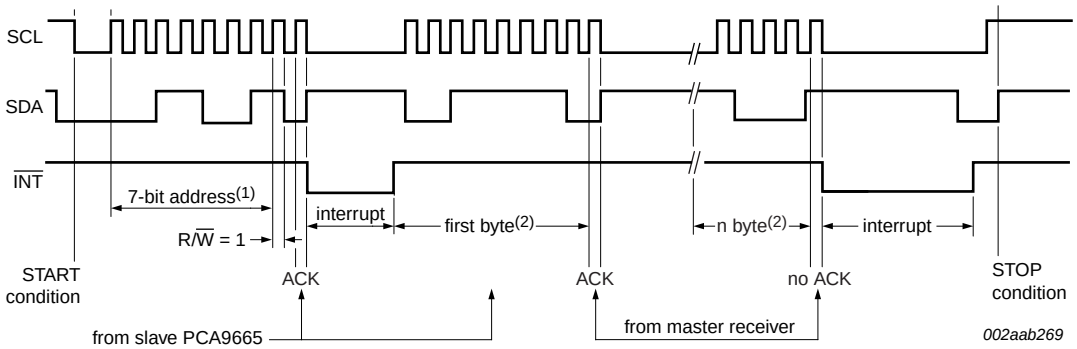
Fig 23. Bus timing diagram; Buffered Master Transmitter mode



Master PCA9665/PCA9665A reads data from slave transmitter.

(1) Number of bytes received = value programmed in I2CCOUNT register (BC[6:0] ≤ 68).

Fig 24. Bus timing diagram; Buffered Master Receiver mode

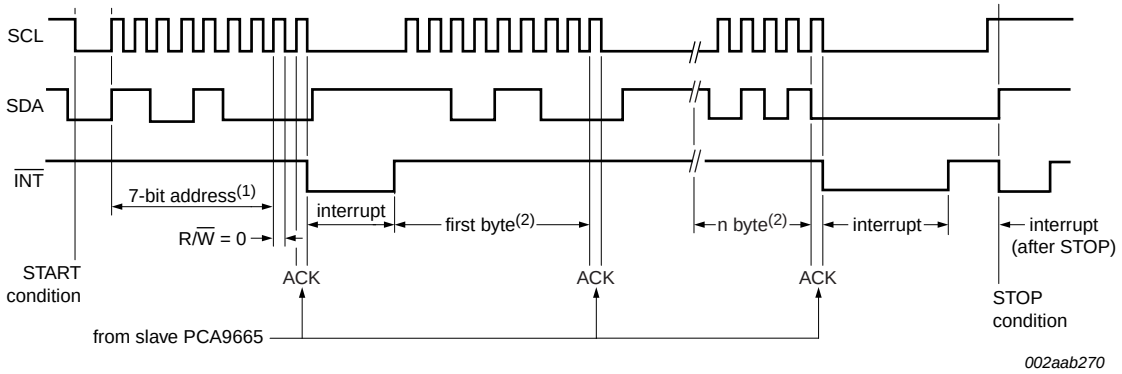


External master receiver reads data from PCA9665/PCA9665A.

(1) As defined in I2CADR register.

(2) Number of bytes received = value programmed in I2CCOUNT register (BC[6:0] ≤ 68).

Fig 25. Bus timing diagram; Buffered Slave Transmitter mode



Slave PCA9665/PCA9665A is written to by external master transmitter.

(1) As defined in I2CADR register.

(2) Number of bytes received = value programmed in I2CCOUNT register (BC[6:0] ≤ 68).

Fig 26. Bus timing diagram; Buffered Slave Receiver mode

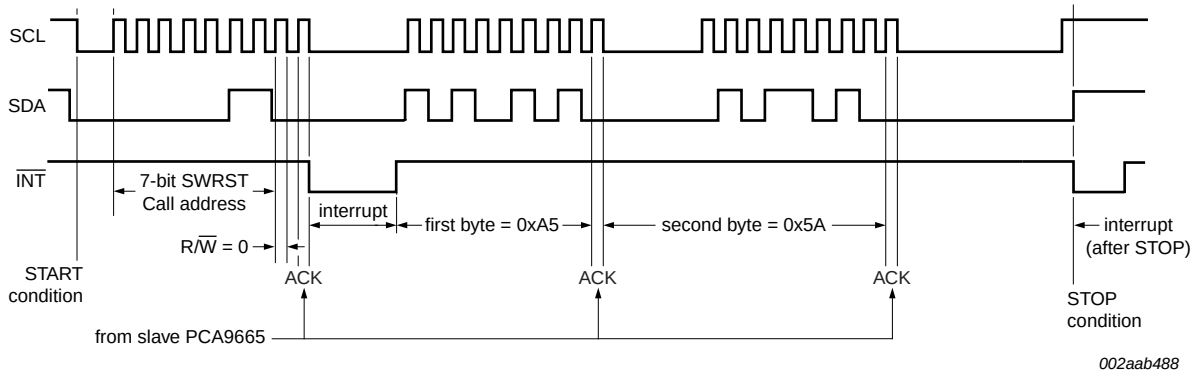


Fig 27. Bus timing diagram; Software Reset Call

9. Characteristics of the I2C-bus

The I2C-bus is for 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device. Data transfer may be initiated only when the bus is not busy.

9.1 Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA 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 28).

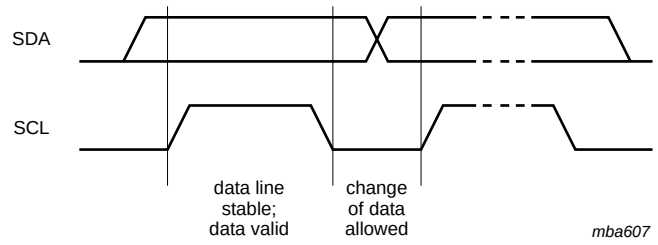


Fig 28. Bit transfer

9.1.1 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 29).

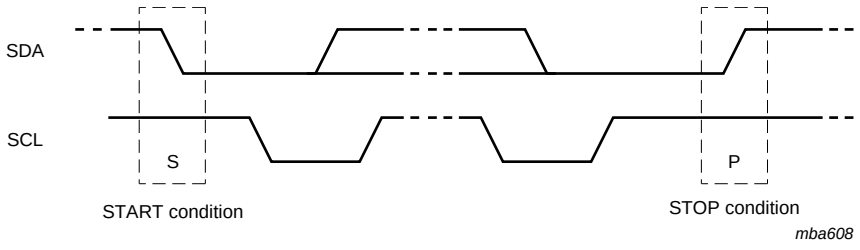


Fig 29. Definition of START and STOP conditions

9.2 System configuration

A device generating a message is a ‘transmitter’; a device receiving is the ‘receiver’. The device that controls the message is the ‘master’ and the devices which are controlled by the master are the ‘slaves’ (see Figure 30).

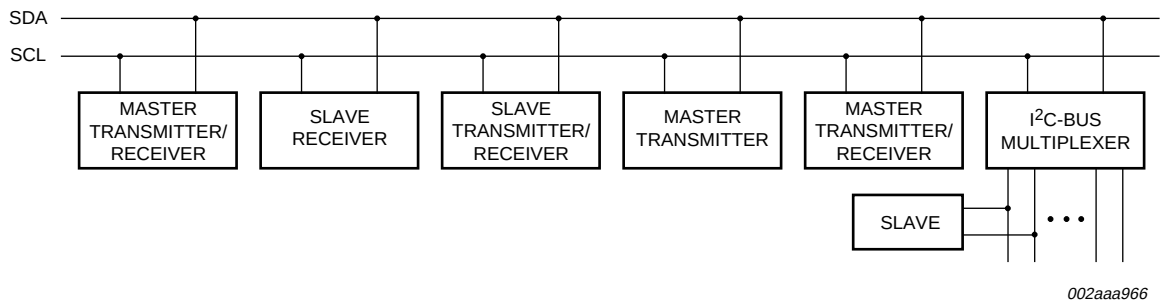


Fig 30. System configuration

9.3 Acknowledge

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 is followed by one acknowledge bit. The acknowledge bit is a HIGH level put on the bus by the transmitter, whereas the master generates an extra acknowledge related clock pulse.

A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse; set-up and hold times must be taken into account.

A master receiver must signal an end of data to the transmitter by not generating an acknowledge on the last byte that has been clocked out of the slave. In this event, the transmitter must leave the data line HIGH to enable the master to generate a STOP condition.

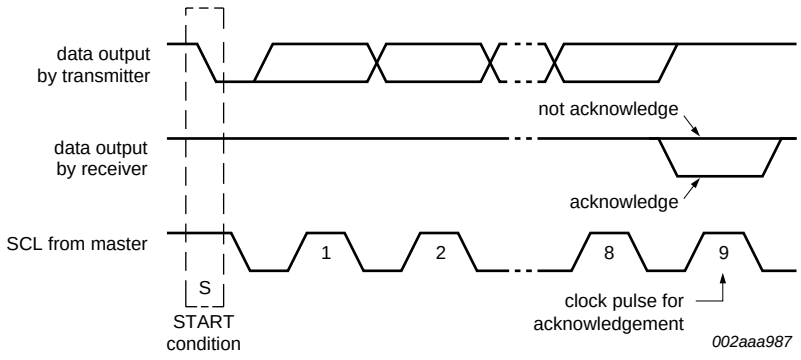


Fig 31. Acknowledgement on the I²C-bus

10. Application design-in information

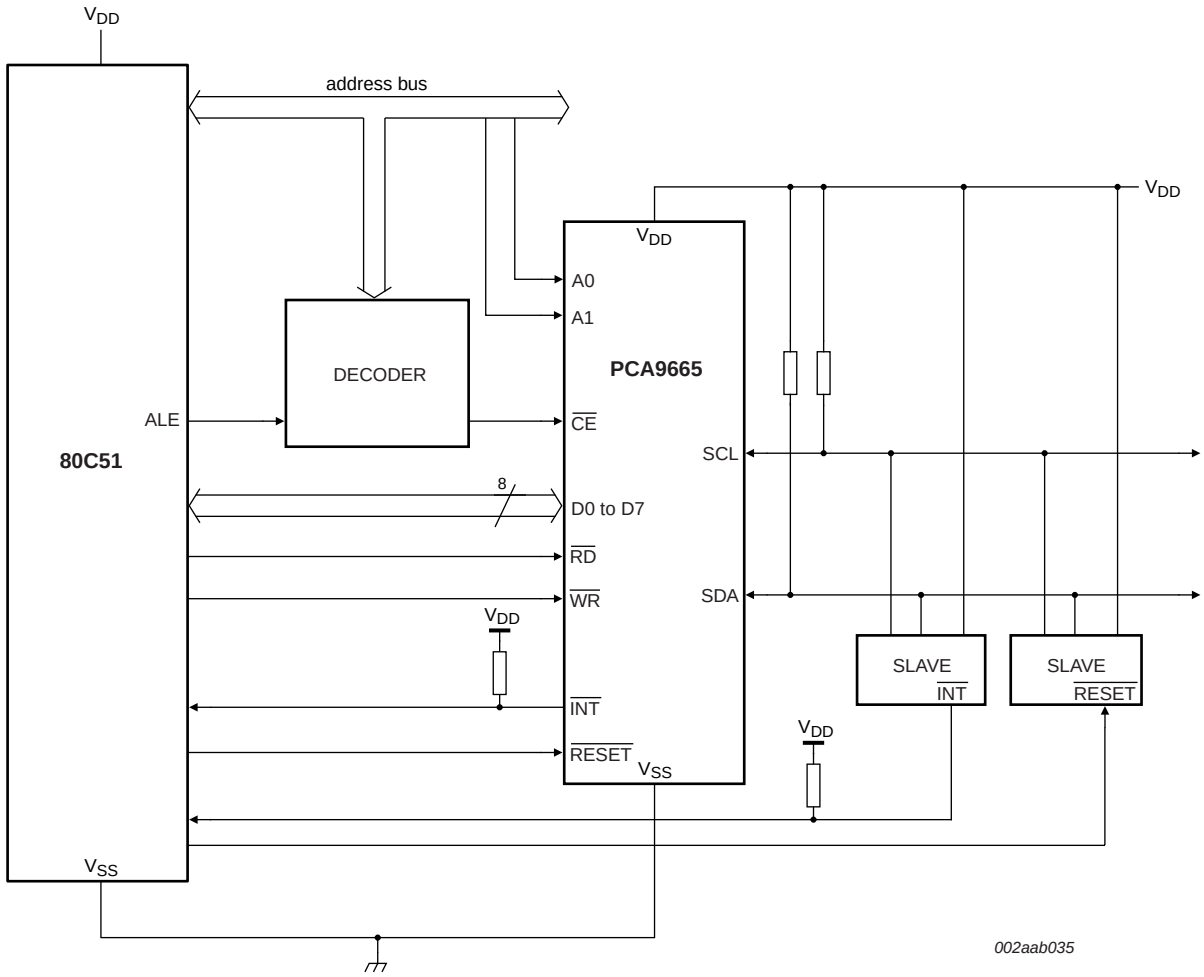


Fig 32. Application diagram using the 80C51

10.1 Specific applications

The PCA9665/PCA9665A is a parallel bus to I²C-bus controller that is designed to allow 'smart' devices to interface with I²C-bus or SMBus components, where the 'smart' device does not have an integrated I²C-bus port and the designer does not want to 'bit-bang' the I²C-bus port. The PCA9665/PCA9665A can also be used to add more I²C-bus ports to 'smart' devices, provide a higher frequency, lower voltage migration path for the PCF8584 and convert 8 bits of parallel data to a serial bus to avoid running multiple traces across the printed-circuit board.

10.2 Add I²C-bus port

As shown in Figure 33, the PCA9665/PCA9665A converts 8-bits of parallel data into a multiple master capable I²C-bus port for microcontrollers, microprocessors, custom ASICs, DSPs, etc., that need to interface with I²C-bus or SMBus components.

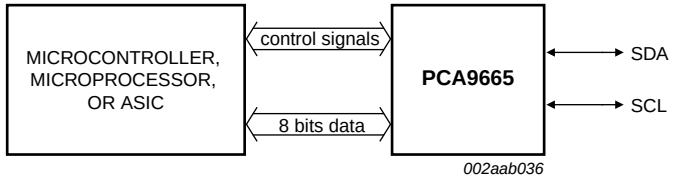


Fig 33. Adding I²C-bus port application

10.2.1 Add I²C-bus port with 'hot swap bus buffers'

The PCA9665A should be used when incremental offset hot swap buffers like the PCA9511A are included in the system such as shown in Figure 34 since the PCA9665 is susceptible to producing a LOW-going spike on a HIGH SDA line during a repeated START as shown in Figure 35 spike labeled 'A'.

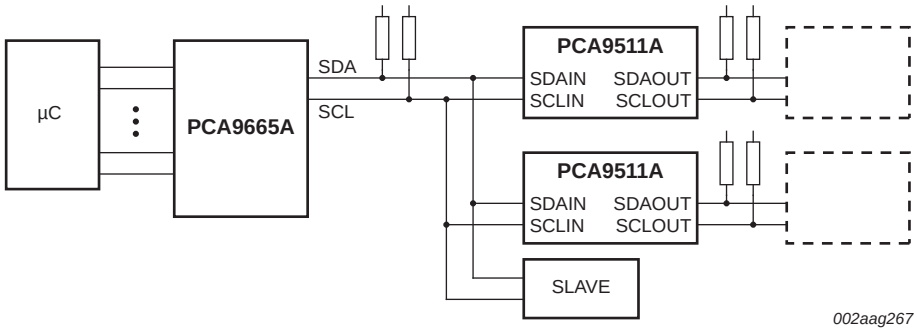


Fig 34. Adding I²C-bus port with 'hot swap bus buffers'

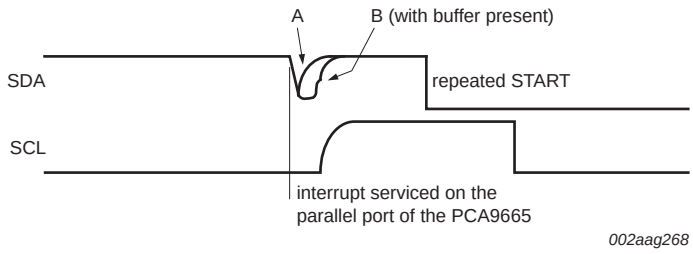


Fig 35. Stretching duration of spike

The spike varies in depth and width, and will return HIGH greater than a data set-up time ($t_{SU;DAT}$) before the SCL rises, so the spike is not a violation of the I²C-bus specification.

An incremental offset hot swap buffer (e.g., PCA9511A) however may interact with the spike in such a way as to stretch the duration of the spike (see Figure 35 spike labeled 'B') and possibly violate the data set-up time. It is therefore recommended to use the PCA9665A in these types of applications since the PCA9665A eliminates the spike on

repeated START conditions. The differences between PCA9665 and PCA9665A are timing related. Section 7.3.2.3 “The Clock Rate registers, I2CSCLL and I2CSCLH (indirect addresses 02h and 03h)” and Section 7.3.2.6 “The I²C-bus mode register, I2CMODE (indirect address 06h)” outline these timing differences. There are no functional or operational differences between the two bus controllers so a PCA9665A can be a drop-in replacement for the PCA9665 with only a small sacrifice in the serial data rate.

Figure 36 shows the glitch-free SDA output of the PCA9665A during a repeated START.

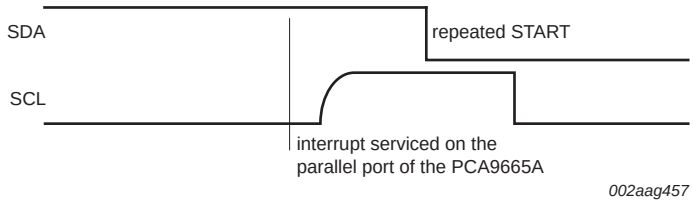


Fig 36. Glitch-free repeated START on PCA9665A

10.3 Add additional I²C-bus ports

The PCA9665/PCA9665A can be used to convert 8-bit parallel data into additional multiple master capable I²C-bus port as shown in Figure 37. It is used if the microcontroller, microprocessor, custom ASIC, DSP, etc., already have an I²C-bus port but need one or more additional I²C-bus ports to interface with more I²C-bus or SMBus components or components that cannot be located on the same bus (e.g., 100 kHz and 400 kHz slaves on different buses so that each bus can operate at its maximum potential).

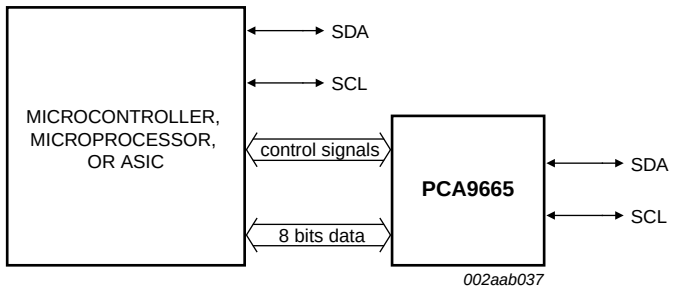


Fig 37. Adding additional I²C-bus ports application

10.4 Convert 8 bits of parallel data into I²C-bus serial data stream

Functioning as a slave transmitter, the PCA9665/PCA9665A can convert 8-bit parallel data into a two-wire I²C-bus data stream as is shown in Figure 38. This would prevent having to run 8 traces across the entire width of the printed-circuit board.

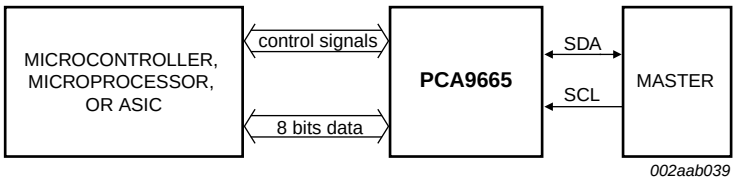


Fig 38. Converting parallel to serial data application

11. Limiting values

Table 47. 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_I	input voltage	any input	[1]	–0.8	+6.0	V
I_I	input current	any input		–10	+10	mA
I_O	output current	any output		–10	+10	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
T_j	junction temperature			-	125	°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.

12. Static characteristics

Table 48. Static characteristics

V_{DD} = 2.3 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			2.3	-	3.6	V
I_{DD}	supply current	standby mode		-	0.1	3.0	mA
		operating mode; no load		-	-	8.0	mA
V_{POR}	power-on reset voltage			-	1.8	2.2	V
Inputs $\overline{WR}$, $\overline{RD}$, A0, A1, $\overline{CE}$, $\overline{RESET}$							
V_{IL}	LOW-level input voltage			0	-	0.8	V
V_{IH}	HIGH-level input voltage		[1]	2.0	-	5.5	V
I_L	leakage current	input; V_I = 0 V or 5.5 V		−1	-	+1	μA
C_i	input capacitance	$V_I = V_{SS}$ or V_{DD}		-	2.0	3	pF
Inputs/outputs D0 to D7							
V_{IL}	LOW-level input voltage			0	-	0.8	V
V_{IH}	HIGH-level input voltage		[1]	2.0	-	5.5	V
I_{OH}	HIGH-level output current	$V_{OH} = V_{DD} - 0.4$ V		−4.0	−7.0	-	mA
I_{OL}	LOW-level output current	V_{OL} = 0.4 V		4.0	8.0	-	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	4	pF
SDA and SCL							
V_{IL}	LOW-level input voltage			0	-	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		[1]	$0.7V_{DD}$	-	5.5	V
I_L	leakage current	input/output; V_I = 0 V or 3.6 V		−1	-	+1	μA
		input/output; V_I = 5.5 V		−1	-	+10	μA
I_{OL}	LOW-level output current	V_{OL} = 0.4 V		20	-	-	mA
C_{io}	input/output capacitance	$V_I = V_{SS}$ or V_{DD}		-	5.6	7	pF
Outputs $\overline{INT}$							
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	-	+1	μA
C_o	output capacitance	$V_I = V_{SS}$ or V_{DD}		-	3.8	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.

13. Dynamic characteristics

Table 49. Dynamic characteristics (3.3 volt)[1][2][3]

V_{CC} = 3.3 V ± 0.3 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified. (See Table 50 on page 75 for 2.5 V)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Initialization timing						
$t_{init(po)}$	power-on initialization time		-	-	550	µs
Serial interface initialization timing						
$t_{init(sintf)}$	serial interface initialization time[4]	from ENSIO bit HIGH	-	-	550	µs
RESET timing (see Figure 39)						
$t_{w(rst)}$	reset pulse width		10	-	-	ns
t_{rst}	reset time	[5][6]	250	-	-	ns
$t_{rec(rst)}$	reset recovery time		0	-	-	ns
INT timing (see Figure 40)						
$t_{as(int)}$	interrupt assert time		-	-	500	ns
$t_{das(int)}$	interrupt de-assert time		-	-	20	ns
Bus timing (see Figure 41 and Figure 43)						
$t_{su(A)}$	address set-up time	to $\overline{RD}$, $\overline{WR}$ LOW	0	-	-	ns
$t_{h(A)}$	address hold time	from $\overline{RD}$, $\overline{WR}$ LOW	13	-	-	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		20	-	-	ns
$t_{w(WRL)}$	$\overline{WR}$ LOW pulse width		20	-	-	ns
$t_{d(DV)}$	data valid delay time	after $\overline{RD}$ and $\overline{CE}$ LOW	-	-	17	ns
$t_{d(QZ)}$	data output float delay time	after $\overline{RD}$ or $\overline{CE}$ HIGH	-	-	17	ns
$t_{su(Q)}$	data output set-up time	before $\overline{WR}$ or $\overline{CE}$ HIGH (write cycle)	12	-	-	ns
$t_{h(Q)}$	data output hold time	after $\overline{WR}$ HIGH	0	-	-	ns
$t_{w(RDH)}$	$\overline{RD}$ HIGH pulse width		18	-	-	ns
$t_{w(WRH)}$	$\overline{WR}$ HIGH pulse width		18	-	-	ns

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

[2] All voltage measurements are referenced to ground (GND). 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 41 and Figure 43.

[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] Initialization time for the serial interface after ENSIO bit goes HIGH in a write operation to the control register.

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

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

Table 50. Dynamic characteristics (2.5 volt)[1][2][3]

V_{CC} = 2.5 V ± 0.2 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified. (See Table 49 on page 74 for 3.3 V)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Initialization timing						
$t_{init(po)}$	power-on initialization time		-	-	550	µs
Serial interface initialization timing						
$t_{init(sintf)}$	serial interface initialization time[4]	from ENSIO bit HIGH	-	-	550	µs
RESET timing (see Figure 39)						
$t_{w(rst)}$	reset pulse width		10	-	-	ns
t_{rst}	reset time	[5][6]	250	-	-	ns
$t_{rec(rst)}$	reset recovery time		0	-	-	ns
INT timing (see Figure 40)						
$t_{as(int)}$	interrupt assert time		-	-	550	ns
$t_{das(int)}$	interrupt de-assert time		-	-	20	ns
Bus timing (see Figure 41 and Figure 43)						
$t_{su(A)}$	address set-up time	to $\overline{RD}$, $\overline{WR}$ LOW	0	-	-	ns
$t_{h(A)}$	address hold time	from $\overline{RD}$, $\overline{WR}$ LOW	13	-	-	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		20	-	-	ns
$t_{w(WRL)}$	$\overline{WR}$ LOW pulse width		20	-	-	ns
$t_{d(DV)}$	data valid delay time	after $\overline{RD}$ and $\overline{CE}$ LOW	-	-	22	ns
$t_{d(QZ)}$	data output float delay time	after $\overline{RD}$ or $\overline{CE}$ HIGH	-	-	17	ns
$t_{su(Q)}$	data output set-up time	before $\overline{WR}$ or $\overline{CE}$ HIGH (write cycle)	12	-	-	ns
$t_{h(Q)}$	data output hold time	after $\overline{WR}$ HIGH	0	-	-	ns
$t_{w(RDH)}$	$\overline{RD}$ HIGH pulse width		18	-	-	ns
$t_{w(WRH)}$	$\overline{WR}$ HIGH pulse width		18	-	-	ns

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

[2] All voltage measurements are referenced to ground (GND). 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 41 and Figure 43.

[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] Initialization time for the serial interface after ENSIO bit goes HIGH in a write operation to the control register.

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

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

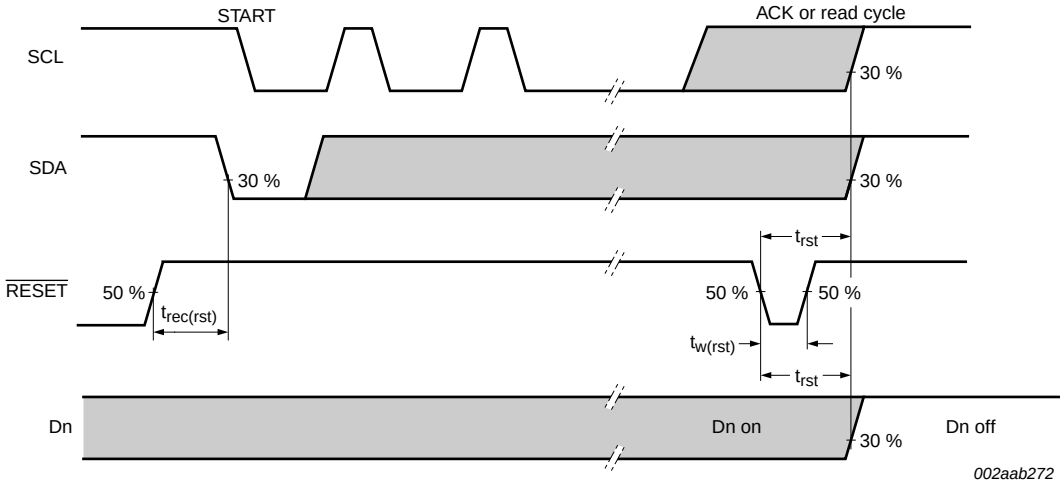


Fig 39. Reset timing

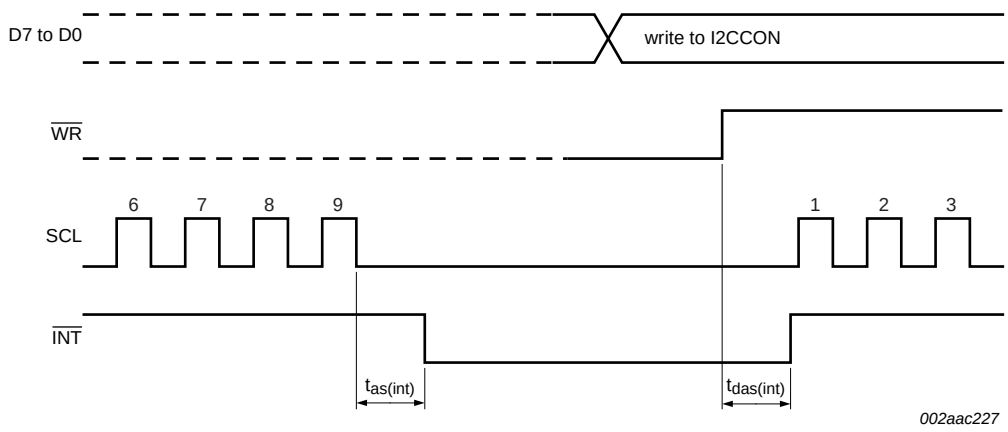


Fig 40. Interrupt timing

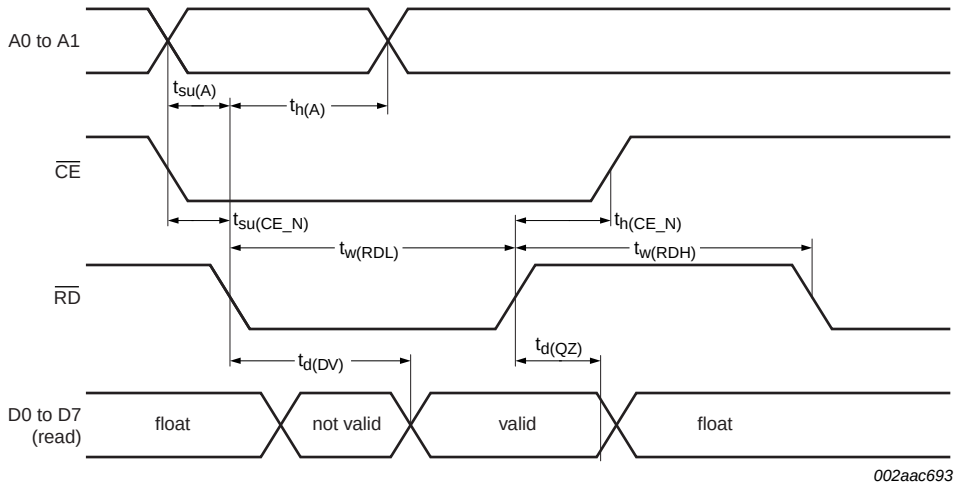


Fig 41. Bus timing (read cycle)

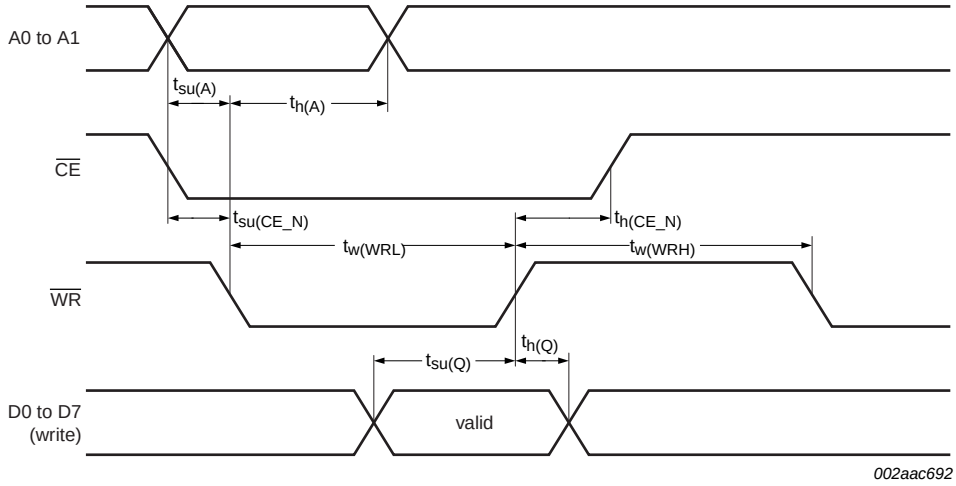
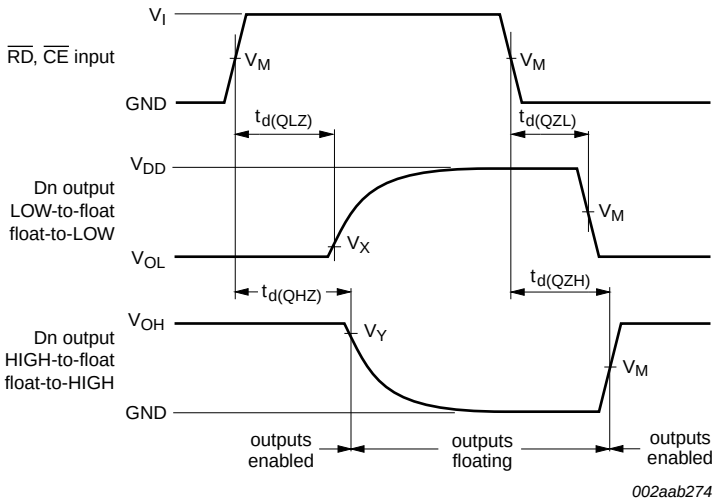


Fig 42. Parallel bus timing (write cycle)



V_M = 1.5 V

$V_X = V_{OL}$ + 0.3 V

$V_Y = V_{OH}$ − 0.3 V

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

Fig 43. Data timing

Table 51. 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	Standard-mode I²C-bus		Fast-mode I²C-bus		Fast-mode Plus I²C-bus		Unit
			Min	Max	Min	Max	Min	Max	
f_{SCL}	SCL clock frequency	[1]	0	100[2]	0	400[2]	0	1000[2]	kHz
t_{BUF}	bus free time between a STOP and START condition		4.7	-	1.3	-	0.5	-	µs
$t_{HD;STA}$	hold time (repeated) START condition		4.0	-	0.6	-	0.26	-	µs
$t_{SU;STA}$	set-up time for a repeated START condition		4.7	-	0.6	-	0.26	-	µs
$t_{SU;STO}$	set-up time for STOP condition		4.0	-	0.6	-	0.26	-	µs
$t_{HD;DAT}$	data hold time		0	-	0	-	0	-	ns
$t_{VD;ACK}$	data valid acknowledge time	[3]	0.05	3.45	0.05	0.9	0.05	0.45	µs
$t_{VD;DAT}$	data valid time	[4]	50	-	50	-	50	-	ns
$t_{SU;DAT}$	data set-up time		250	-	100	-	50	-	ns
t_{LOW}	LOW period of the SCL clock		4.7	-	1.3	-	0.5	-	µs
t_{HIGH}	HIGH period of the SCL clock		4.0	-	0.6	-	0.26	-	µs
t_f	fall time of both SDA and SCL signals	[6][7]	-	300	$20 + 0.1C_b$[5]	300	-	120	ns
t_r	rise time of both SDA and SCL signals		-	1000	$20 + 0.1C_b$[5]	300	-	120	ns
t_{SP}	pulse width of spikes that must be suppressed by the input filter	[8]	-	50	-	50	-	50	ns

[1] Minimum SCL clock frequency is limited by the bus time-out feature, which resets the serial bus interface if either SDA or SCL is held LOW for a minimum of 25 ms. Disable bus time-out feature for DC operation.

[2] The f_{SCL} maximum is derived from the sum of the pulse width HIGH minimum, the pulse width LOW minimum, the t_f maximum and the t_r maximum for each node.

[3] $t_{VD;ACK}$ = time for Acknowledgement signal from SCL LOW to SDA (out) LOW.

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

[5] C_b = total capacitance of one bus line in pF.

[6] A master device must internally provide a hold time of at least 300 ns for the SDA signal (refer to the V_{IL} of the SCL signal) in order to bridge the undefined region SCL's falling edge.

[7] The maximum t_f for the SDA and SCL bus lines is specified at 300 ns. The maximum fall time for the SDA output stage t_f is specified at 250 ns. This allows series protection resistors to be connected between the SDA and the SCL pins and the SDA/SCL bus lines without exceeding the maximum specified t_f.

[8] Input filters on the SDA and SCL inputs suppress noise spikes less than 50 ns.

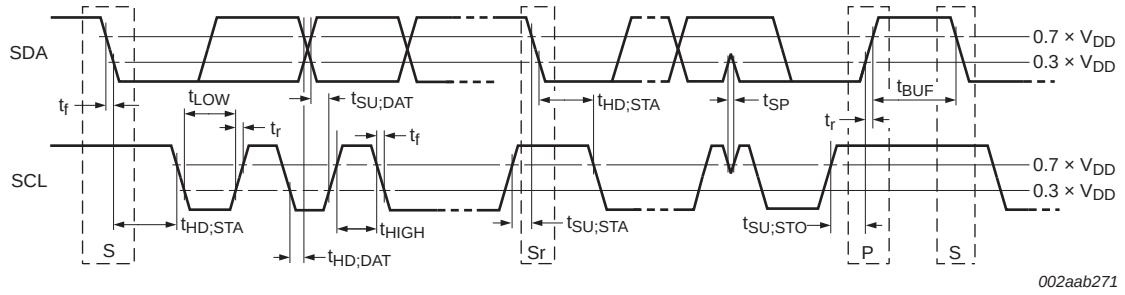
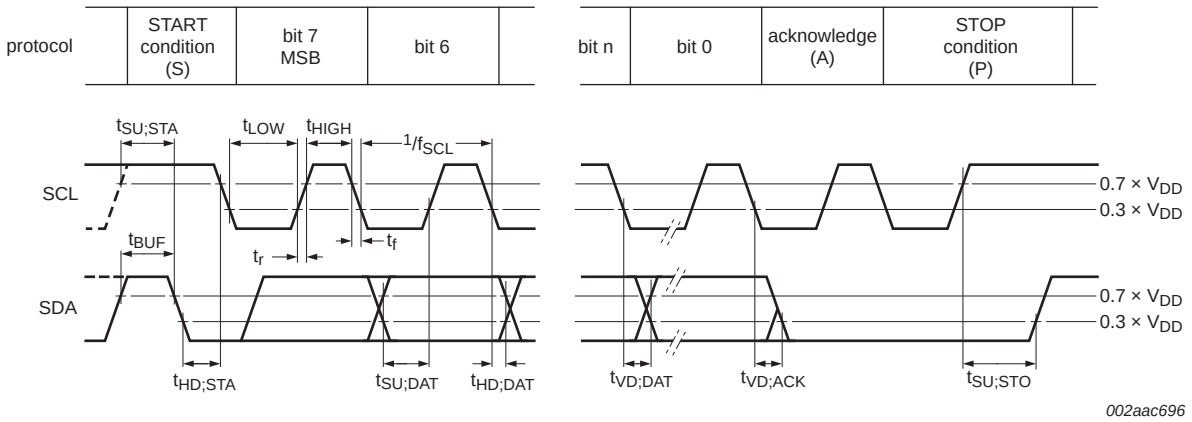


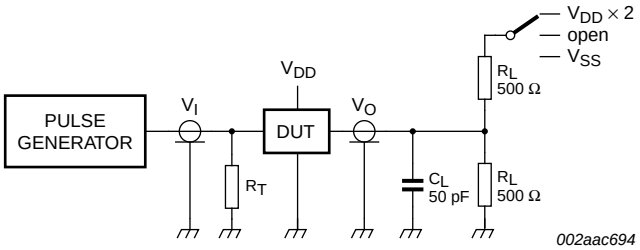
Fig 44. Definition of timing on the I²C-bus



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

Fig 45. I²C-bus timing diagram

14. Test information



Test data are given in Table 52.

R_L = load resistance.

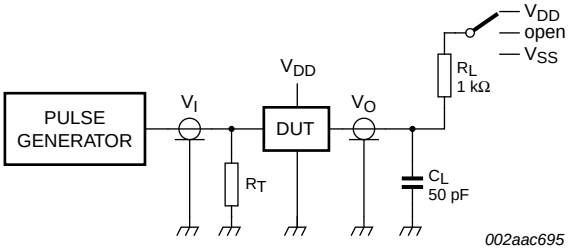
C_L = load capacitance includes jig and probe capacitance.

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

Fig 46. Test circuitry for switching times

Table 52. Test data

Test	Load		S1
	C_L	R_L	
$t_{d(DV)}$	50 pF	500 Ω	$V_{DD} \times 2$
$t_{d(QZ)}$	50 pF	500 Ω	open



Test data are given in Table 53.

R_L = load resistance. R_L for SDA and SCL > 1 kΩ (3 mA or less current).

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.

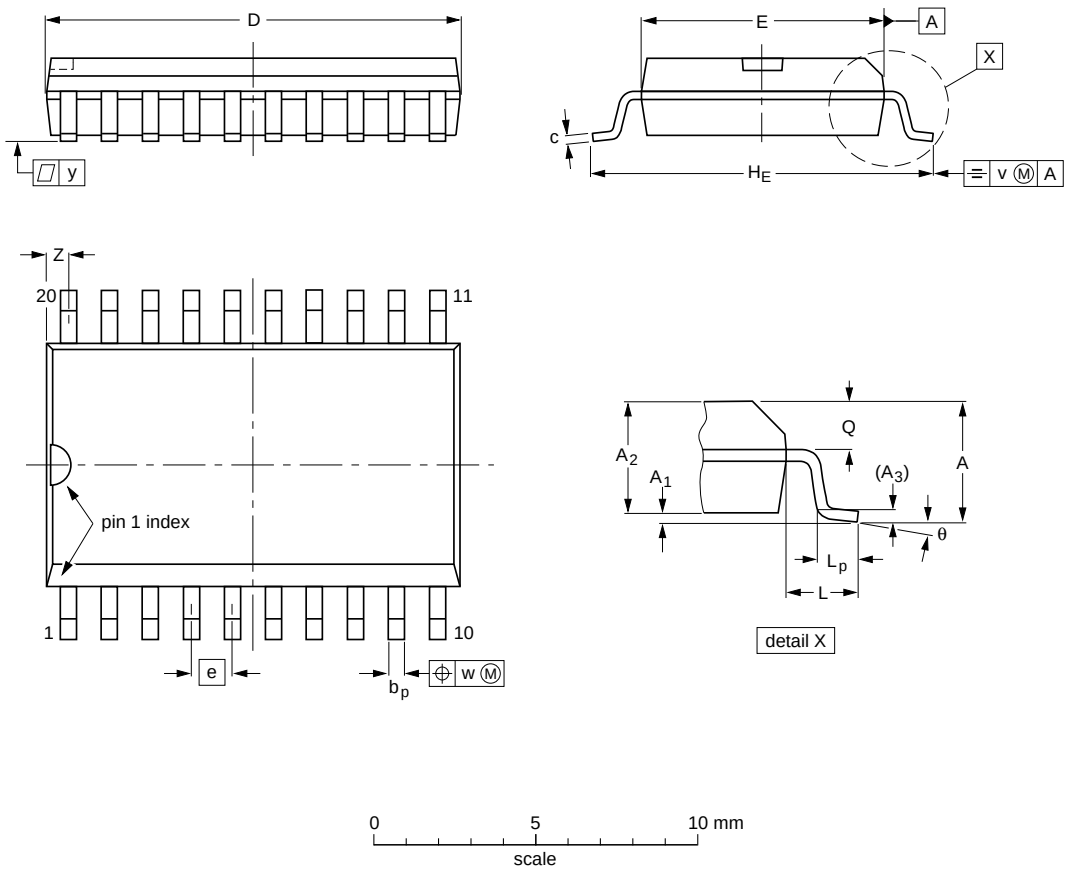
Fig 47. Test circuitry for open-drain switching times

Table 53. Test data

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

15. Package outline

SO20: plastic small outline package; 20 leads; body width 7.5 mm **SOT163-1**



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	D (1)	E (1)	e	H_E	L	L_p	Q	v	w	y	Z (1)	θ
mm	2.65	0.3 0.1	2.45 2.25	0.25	0.49 0.36	0.32 0.23	13.0 12.6	7.6 7.4	1.27	10.65 10.00	1.4	1.1 0.4	1.1 1.0	0.25	0.25	0.1	0.9 0.4	8^o 0^o
inches	0.1	0.012 0.004	0.096 0.089	0.01	0.019 0.014	0.013 0.009	0.51 0.49	0.30 0.29	0.05	0.419 0.394	0.055	0.043 0.016	0.043 0.039	0.01	0.01	0.004	0.035 0.016	

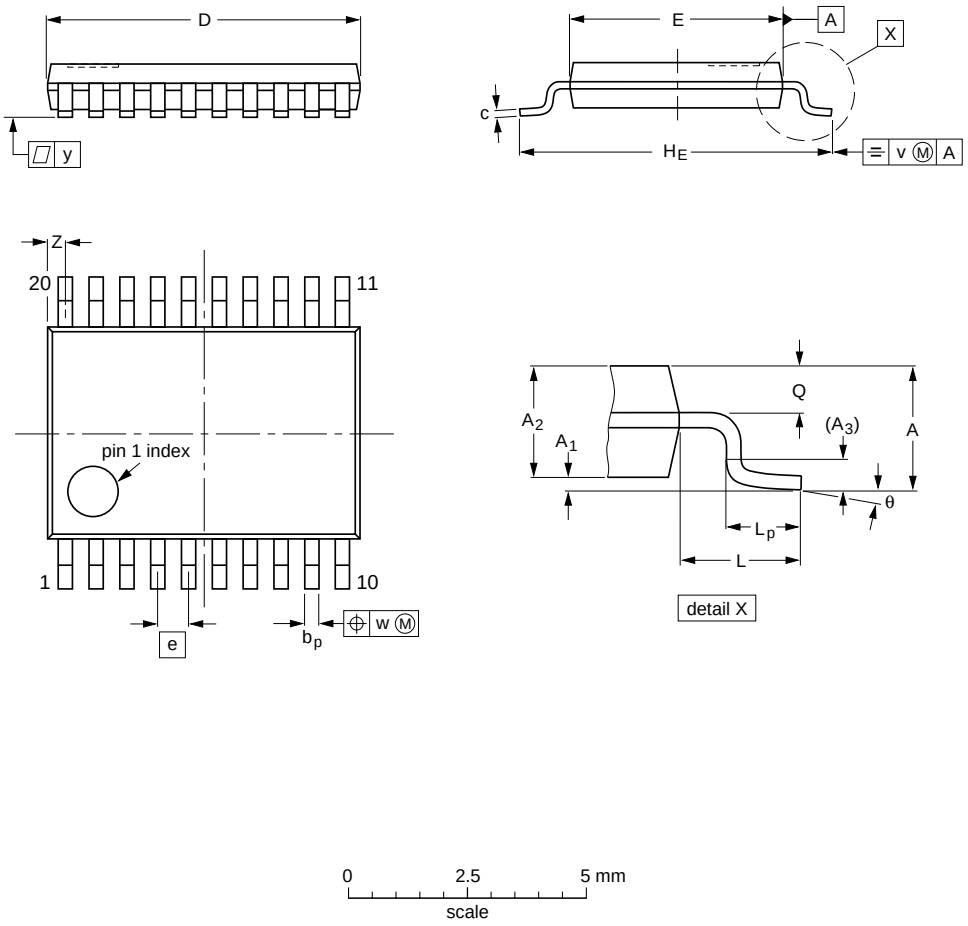
Note

1. Plastic or metal protrusions of 0.15 mm (0.006 inch) maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT163-1	075E04	MS-013				~~99-12-27~~ 03-02-19

Fig 48. Package outline SOT163-1 (SO20)

TSSOP20: plastic thin shrink small outline package; 20 leads; body width 4.4 mm **SOT360-1**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	D (1)	E (2)	e	H_E	L	L_p	Q	v	w	y	Z (1)	θ
mm	1.1	0.15 0.05	0.95 0.80	0.25	0.30 0.19	0.2 0.1	6.6 6.4	4.5 4.3	0.65	6.6 6.2	1	0.75 0.50	0.4 0.3	0.2	0.13	0.1	0.5 0.2	8^o 0^o

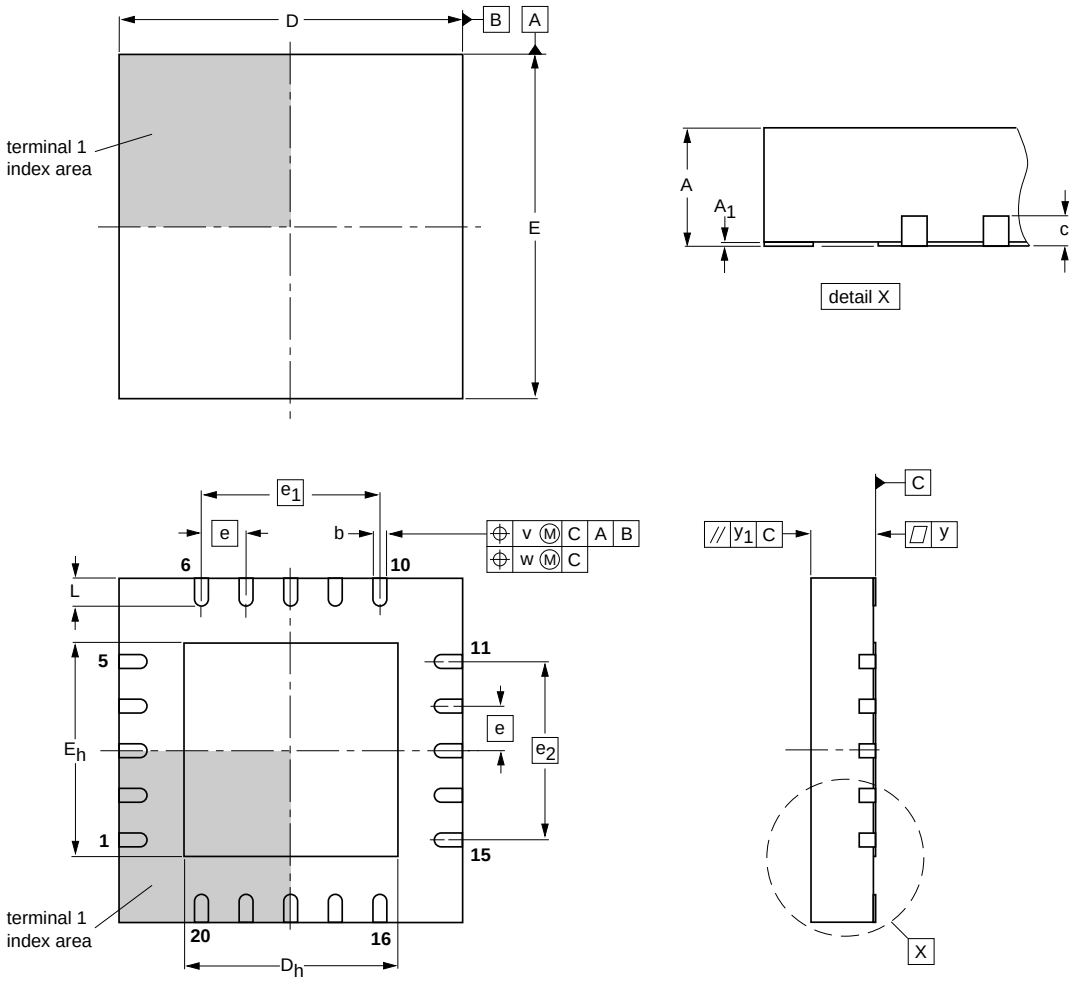
Notes

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.
2. Plastic interlead protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT360-1		MO-153				~~99-12-27~~ 03-02-19

Fig 49. Package outline SOT360-1 (TSSOP20)

HVQFN20: plastic thermal enhanced very thin quad flat package; no leads; 20 terminals; body 5 x 5 x 0.85 mm **SOT662-1**



DIMENSIONS (mm are the original dimensions)

UNIT	$A^{(1)}$ max.	A_1	b	c	$D^{(1)}$	D_h	$E^{(1)}$	E_h	e	e_1	e_2	L	v	w	y	y_1
mm	1	0.05 0.00	0.38 0.23	0.2	5.1 4.9	3.25 2.95	5.1 4.9	3.25 2.95	0.65	2.6	2.6	0.75 0.50	0.1	0.05	0.05	0.1

Note

1. Plastic or metal protrusions of 0.075 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT662-1	- - -	MO-220	- - -			~~01-08-08~~ 02-10-22

Fig 50. Package outline SOT662-1 (HVQFN20)

16. 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.

17. 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"*.

17.1 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.

17.2 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

17.3 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

17.4 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 51) 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 54 and 55

Table 54. SnPb eutectic process (from J-STD-020C)

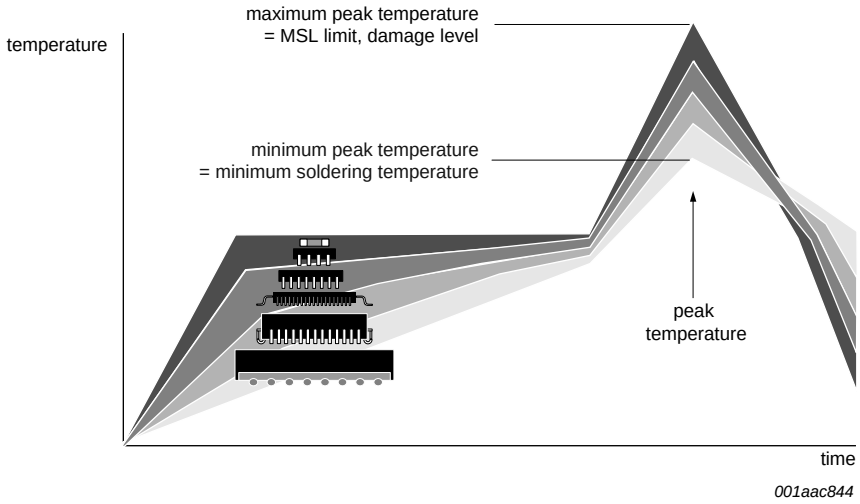
Package thickness (mm)	Package reflow temperature (°C)	
	Volume (mm³)	
	< 350	≥ 350
< 2.5	235	220
≥ 2.5	220	220

Table 55. 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 51.



MSL: Moisture Sensitivity Level

Fig 51. Temperature profiles for large and small components

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

18. Abbreviations

Table 56. Abbreviations

Acronym	Description
ASIC	Application Specific Integrated Circuit
CDM	Charged Device Model
CPU	Central Processing Unit
DSP	Digital Signal Processing
DUT	Device Under Test
ESD	ElectroStatic Discharge
HBM	Human Body Model
I²C-bus	Inter-Integrated Circuit bus
I/O	Input/Output
PCB	Printed-Circuit Board
RC	Resistor-Capacitor network
SMBus	System Management Bus

19. Revision history

Table 57. Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
PCA9665_PCA9665A v.4	20110929	Product data sheet	-	PCA9665 v.3
Modifications:	• Added type number PCA9665A • Section 2 “Features and benefits”: – added new 12th bullet item – (new) 13th bullet item: deleted phrase “200 V MM per JESD22-A115” – (new) 15th bullet item, first sub-bullet: deleted “DIP20” • Table 1 “Ordering information”: – deleted type number “PCA9665N”, DIP20 package – added type number “PCA9665APW” • Section 6.1 “Pinning”: removed (old) Figure 4, “Pin configuration for DIP20” • Table 2 “Pin description”: removed “DIP20” from column heading • Section 7.2 “Internal oscillator”: – first sentence: deleted “28.5 MHz” – added (new) second sentence • Section 7.3.2.3 “The Clock Rate registers, I2CSCLL and I2CSCLH (indirect addresses 02h and 03h)”: – first paragraph, third sentence: appended “t_d (delay time)” – second paragraph: added second sentence – second paragraph: third sentence re-written – Equation 1 modified – added new third paragraph • Table 22 “I2CTO - Time-out register (indirect register 04h) bit description”: description of bits TO[6:0] updated • Table 25 “I²C-bus mode selection example[1]”: – I²C-bus frequency (kHz) values for PCA9665 updated – added column for PCA9665A – added value for PCA9665 Turbo mode – Table note [2] re-written – added (new) Table note [3] • Table 47 “Limiting values”: added “T_j, junction temperature” limit • Table 51 “I²C-bus frequency and timing specifications”: added (new) Table note [2] • Added Section 10.2.1 “Add I²C-bus port with ‘hot swap bus buffers’” • Deleted (old) Figure 49, “Package outline SOT146-1 (DIP20)” • Deleted (old) Section 18, “Soldering of through-hole mount packages”			
PCA9665 v.3	20080812	Product data sheet	-	PCA9665 v.2
PCA9665 v.2	20061207	Product data sheet	-	PCA9665 v.1
PCA9665 v.1	20060807	Objective data sheet	-	-

20. Legal information

20.1 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.

20.2 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.

20.3 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 competent 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.

20.4 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.

21. Contact information

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

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

22. Contents

continued >>

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

Date of release: 29 September 2011
Document identifier: PCA9665_PCA9665A

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

NXP:

PCA9665BS,118 PCA9665D,112 PCA9665D,118 PCA9665PW,112 PCA9665PW,118 PCA9665APW,118

LifeElectronics
Живое партнерство

ООО «ЛайфЭлектроникс» **“LifeElectronics” LLC**

ИНН 7805602321 КПП 780501001 Р/С 40702810122510004610 ФАКБ "АБСОЛЮТ БАНК" (ЗАО) в г.Санкт-Петербурге К/С 30101810900000000703 БИК 044030703

Компания «Life Electronics» занимается поставками электронных компонентов импортного и отечественного производства от производителей и со складов крупных дистрибьюторов Европы, Америки и Азии.

С конца 2013 года компания активно расширяет линейку поставок компонентов по направлению коаксиальный кабель, кварцевые генераторы и конденсаторы (керамические, пленочные, электролитические), за счёт заключения дистрибьюторских договоров

Мы предлагаем:

- *Конкурентоспособные цены и скидки постоянным клиентам.*
- *Специальные условия для постоянных клиентов.*
- *Подбор аналогов.*
- *Поставку компонентов в любых объемах, удовлетворяющих вашим потребностям.*

- *Приемлемые сроки поставки, возможна ускоренная поставка.*
- *Доставку товара в любую точку России и стран СНГ.*
- *Комплексную поставку.*
- *Работу по проектам и поставку образцов.*
- *Формирование склада под заказчика.*

- *Сертификаты соответствия на поставляемую продукцию (по желанию клиента).*
- *Тестирование поставляемой продукции.*
- *Поставку компонентов, требующих военную и космическую приемку.*
- *Входной контроль качества.*
- *Наличие сертификата ISO.*

В составе нашей компании организован Конструкторский отдел, призванный помогать разработчикам, и инженерам.

Конструкторский отдел помогает осуществить:

- *Регистрацию проекта у производителя компонентов.*
- *Техническую поддержку проекта.*
- *Защиту от снятия компонента с производства.*
- *Оценку стоимости проекта по компонентам.*
- *Изготовление тестовой платы монтаж и пусконаладочные работы.*

Тел: +7 (812) 336 43 04 (многоканальный)
Email: org@lifeelectronics.ru

www.lifeelectronics.ru