PCA9541

2-to-1 I²C-bus master selector with interrupt logic and reset

Rev. 04 — 4 January 2006 Product data sheet

1. General description

The PCA9541 is a 2-to-1 I²C-bus master selector designed for high reliability dual master I²C-bus applications where system operation is required, even when one master fails or the controller card is removed for maintenance. The two masters (for example, primary and back-up) are located on separate I²C-buses that connect to the same downstream I²C-bus slave devices. I²C-bus commands are sent by either I²C-bus master and are used to select one master at a time. Either master at any time can gain control of the slave devices if the other master is disabled or removed from the system. The failed master is isolated from the system and will not affect communication between the on-line master and the slave devices on the downstream I²C-bus.

Three versions are offered for different architectures. PCA9541/01 with channel 0 selected at start-up, PCA9541/02 with channel 0 selected after start-up and after STOP condition is detected, and PCA9541/03 with no channel selected after start-up.

The interrupt outputs are used to provide an indication of which master has control of the bus. One interrupt input ($\overline{INT\_IN}$) collects downstream information and propagates it to the 2 upstream I²C-buses ($\overline{INT0}$ and $\overline{INT1}$) if enabled. $\overline{INT0}$ and $\overline{INT1}$ are also used to let the previous bus master know that it is not in control of the bus anymore and to indicate the completion of the bus recovery/initialization sequence. Those interrupts can be disabled and will not generate an interrupt if the masking option is set.

A bus recovery/initialization if enabled sends nine clock pulses, a not acknowledge, and a STOP condition in order to set the downstream I²C-bus devices to an initialized state before actually switching the channel to the selected master.

An interrupt is sent to the upstream channel when the recovery/initialization procedure is completed.

An internal bus sensor senses the downstream I²C-bus traffic and generates an interrupt if a channel switch occurs during a non-idle bus condition. This function is enabled when the PCA9541 recovery/initialization is not used. The interrupt signal informs the master that an external I²C-bus recovery/initialization needs to be performed. It can be disabled and an interrupt will not be generated.

The pass gates of the switches are constructed such that the V_{DD} pin can be used to limit the maximum high voltage, which will be passed by the PCA9541. This allows the use of different bus voltages on each pair, so that 1.8 V, 2.5 V, or 3.3 V devices can communicate with 5 V devices without any additional protection.

The PCA9541 does not isolate the capacitive loading on either side of the device, so the designer must take into account all trace and device capacitances on both sides of the device, and pull-up resistors must be used on all channels.

PHILIPS

External pull-up resistors pull the bus to the desired voltage level for each channel. All I/O pins are 6.0 V tolerant.

An active LOW reset input allows the PCA9541 to be initialized. Pulling the $\overline{\text{RESET}}$ pin LOW resets the I²C-bus state machine and configures the device to its default state as does the internal Power-On Reset (POR) function.

2. Features

- 2-to-1 bidirectional master selector
- I²C-bus interface logic; compatible with SMBus standards
- PCA9541/01 powers up with Channel 0 selected
- PCA9541/02 powers up with Channel 0 selected after STOP condition detected (bus idle) on Channel 0
- PCA9541/03 powers up with no channel selected and either master can take control of the bus
- Active LOW interrupt input
- 2 active LOW interrupt outputs
- Active LOW reset input
- 4 address pins allowing up to 16 devices on the I²C-bus
- Channel selection via I²C-bus
- Bus initialization/recovery function
- Bus traffic sensor
- Low R_{on} switches
- Allows voltage level translation between 1.8 V, 2.5 V, 3.3 V and 5 V buses
- No glitch on power-up
- Supports hot insertion
- Software identical for both masters
- Low standby current
- Operating power supply voltage range of 2.3 V to 5.5 V
- 6.0 V tolerant inputs
- 0 Hz to 400 kHz clock frequency
- ESD protection exceeds 2000 V HBM per JESD22-A114, 200 V MM per JESD22-A115, and 1000 V CDM per JESD22-C101
- Latch-up testing is done to JEDEC Standard JESD78 which exceeds 100 mA
- Packages offered: SO16, TSSOP16, HVQFN16

3. Applications

- High reliability systems with dual masters
- Gatekeeper multiplexer on long single bus
- Bus initialization/recovery for slave devices without hardware reset
- Allows masters without arbitration logic to share resources

4. Ordering information

Table 1: Ordering information

T_{amb} = −40 °C to +85 °C

Type number	Package		
	Name	Description	Version
PCA9541D/01	SO16	plastic small outline package; 16 leads; body width 3.9 mm	SOT109-1
PCA9541PW/01	TSSOP16	plastic thin shrink small outline package; 16 leads; body width 4.4 mm	SOT403-1
PCA9541BS/01	HVQFN16	plastic thermal enhanced very thin quad flat package; no leads; 16 terminals; body 4 × 4 × 0.85 mm	SOT629-1
PCA9541D/02	SO16	plastic small outline package; 16 leads; body width 3.9 mm	SOT109-1
PCA9541PW/02	TSSOP16	plastic thin shrink small outline package; 16 leads; body width 4.4 mm	SOT403-1
PCA9541BS/02	HVQFN16	plastic thermal enhanced very thin quad flat package; no leads; 16 terminals; body 4 × 4 × 0.85 mm	SOT629-1
PCA9541D/03	SO16	plastic small outline package; 16 leads; body width 3.9 mm	SOT109-1
PCA9541PW/03	TSSOP16	plastic thin shrink small outline package; 16 leads; body width 4.4 mm	SOT403-1
PCA9541BS/03	HVQFN16	plastic thermal enhanced very thin quad flat package; no leads; 16 terminals; body 4 × 4 × 0.85 mm	SOT629-1

5. Marking

Table 2: Marking codes

Type number	Topside mark
PCA9541D/01	PCA9541D/01
PCA9541PW/01	9541/01
PCA9541BS/01	41/1
PCA9541D/02	PCA9541D/02
PCA9541PW/02	9541/02
PCA9541BS/02	41/2
PCA9541D/03	PCA9541D/03
PCA9541PW/03	9541/03
PCA9541BS/03	41/3

6. Block diagram

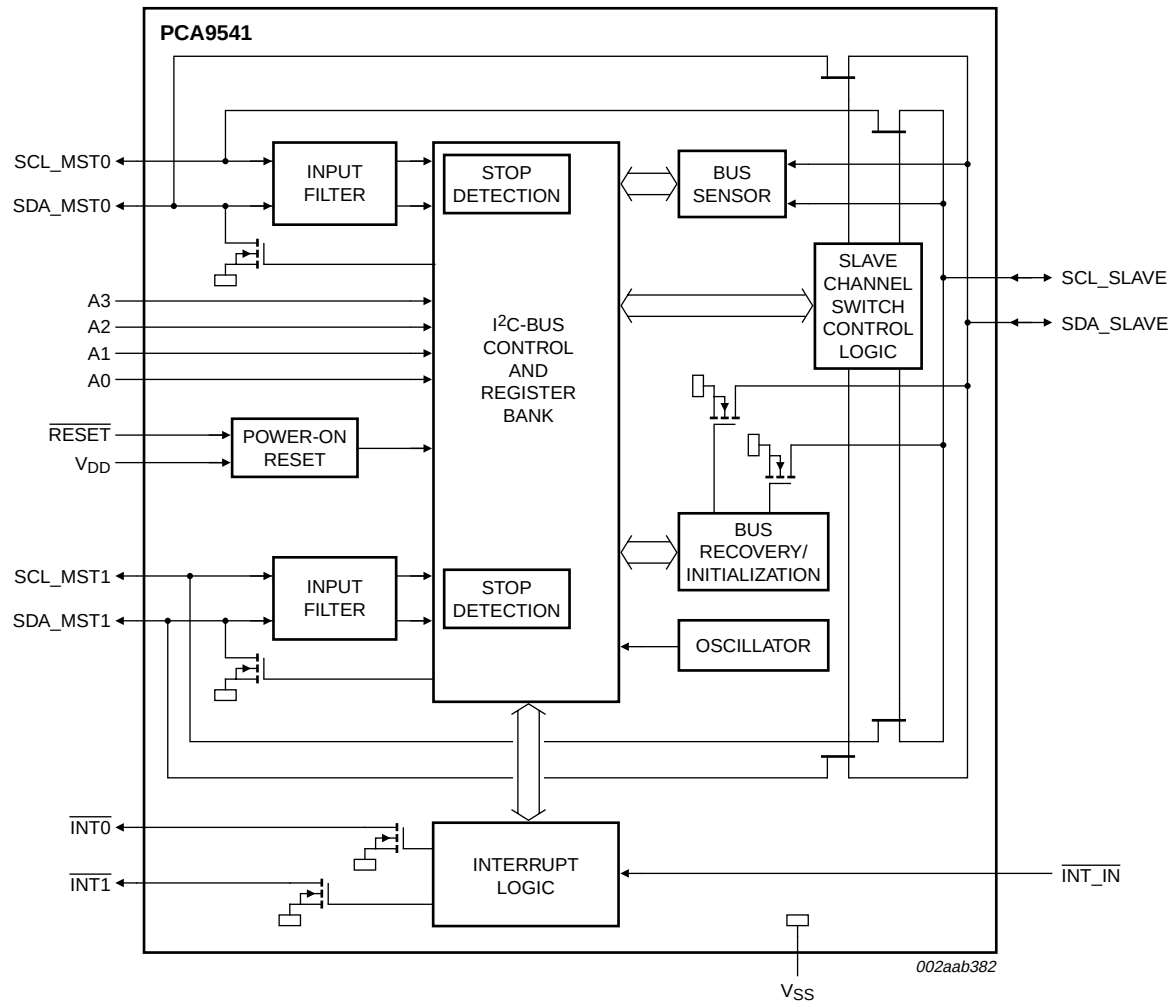


Fig 1. Block diagram of PCA9541

7. Pinning information

7.1 Pinning

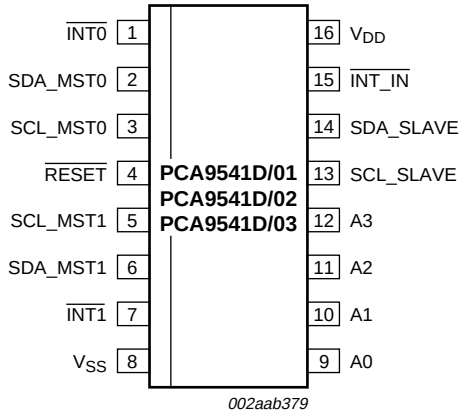


Fig 2. Pin configuration for SO16

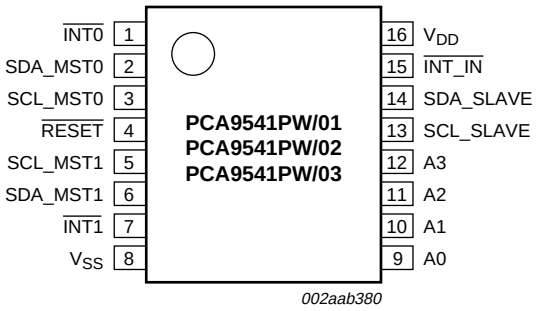
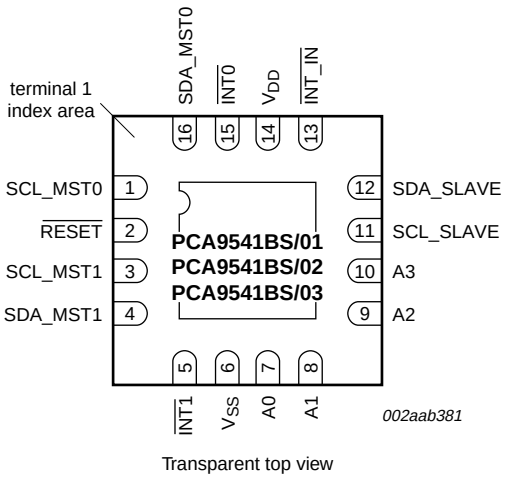


Fig 3. Pin configuration for TSSOP16



Transparent top view

Fig 4. Pin configuration for HVQFN16

7.2 Pin description

Table 3: Pin description

Symbol	Pin		Description
	SO16, TSSOP16	HVQFN16	
$\overline{INT0}$	1	15	active LOW interrupt output 0 (external pull-up required)
SDA_MST0	2	16	serial data master 0 (external pull-up required)
SCL_MST0	3	1	serial clock master 0 (external pull-up required)
$\overline{RESET}$	4	2	active LOW reset input (external pull-up required)
SCL_MST1	5	3	serial clock master 1 (external pull-up required)
SDA_MST1	6	4	serial data master 1 (external pull-up required)
$\overline{INT1}$	7	5	active LOW interrupt output 1 (external pull-up required)
V_{SS}	8	6 [1]	supply ground
A0	9	7	address input 0 (externally held to V_{SS} or V_{DD})
A1	10	8	address input 1 (externally held to V_{SS} or V_{DD})
A2	11	9	address input 2 (externally held to V_{SS} or V_{DD})
A3	12	10	address input 3 (externally held to V_{SS} or V_{DD})
SCL_SLAVE	13	11	serial clock slave (external pull-up required)
SDA_SLAVE	14	12	serial data slave (external pull-up required)
$\overline{INT\_IN}$	15	13	active LOW interrupt input (external pull-up required)
V_{DD}	16	14	supply voltage

[1] HVQFN 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 printed-circuit board in the thermal pad region.

8. Functional description

Refer to Figure 1 “Block diagram of PCA9541”.

8.1 Device address

Following a START condition, the upstream master that wants to control the I²C-bus or make a status check must send the address of the slave it is accessing. The slave address of the PCA9541 is shown in Figure 5. To conserve power, no internal pull-up resistors are incorporated on the hardware selectable pins and they must be pulled HIGH or LOW.

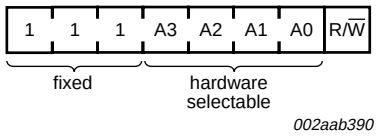


Fig 5. Slave address

The last bit of the slave address defines the operation to be performed. When set to logic 1 a read is selected, while logic 0 selects a write operation.

CAUTION

Reserved I²C-bus addresses must be used with caution since they can interfere with:

- ‘reserved for future use’ I²C-bus addresses (11111XX)
- slave devices that use the 10-bit addressing scheme (11110XX)

8.2 Command Code

Following the successful acknowledgement of the slave address, the bus master will send a byte to the PCA9541, which will be stored in the Command Code register.

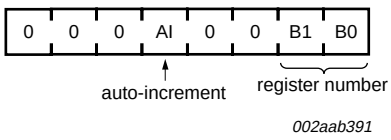


Fig 6. Command Code

The 2 LSBs are used as a pointer to determine which register will be accessed.

If the auto-increment flag is set (AI = 1), the two least significant bits of the Command Code are automatically incremented after a byte has been read or written. This allows the user to program the registers sequentially or to read them sequentially.

- During a read operation, the contents of these bits will roll over to ‘00’ after the last allowed register is accessed (‘10’).

- During a write operation, the PCA9541 will acknowledge bytes sent to the IE and CONTROL registers, but will not acknowledge a byte sent to the Interrupt Status Register since it is a read-only register. The 2 LSBs of the Command Code do not roll over to ‘00’ but stay at ‘10’.

Only the 2 least significant bits are affected by the AI flag.

Unused bits must be programmed with zeros. Any command code (write operation) different from ‘000AI0000’, ‘000AI0001’, and ‘000AI0010’ will not be acknowledged. At power-up, this register defaults to all zeros.

Table 4: Command Code register

B1	B0	Register name	Type	Register function
0	0	IE	R/W	interrupt enable
0	1	CONTROL	R/W	control switch
1	0	ISTAT	R only	interrupt status
1	1	not allowed		

Each system master controls its own set of registers, however they can also read specific bits from the other system master.

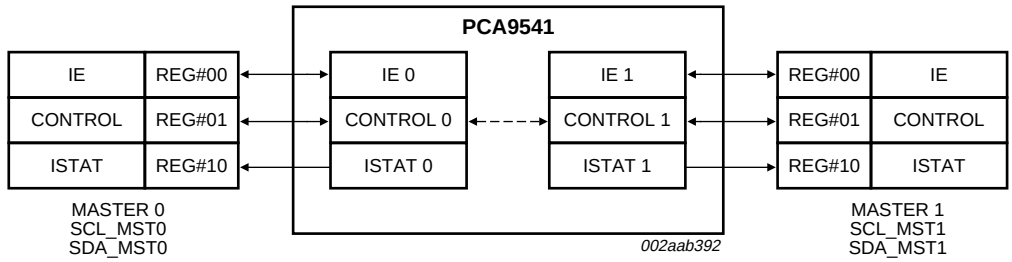


Fig 7. Internal register map

8.3 Interrupt Enable and Control registers description

When a master seeks control of the bus by connecting its I²C-bus channel to the PCA9541 downstream channel, it has to write to the Control Register (Reg#01).

Bits MYBUS and BUSON allow the master to take control of the bus.

The MYBUS and the NMYBUS bits determine which master has control of the bus. Table 9 explains which master gets control of the bus and how. There is no arbitration. Any master can take control of the bus when it wants regardless of whether the other master is using it or not.

The BUSON and the NBUSON bits determine whether the upstream bus is connected or disconnected to/from the downstream bus. Table 10 explains when the upstream bus is connected or disconnected.

Internally, the state machine does the following:

- If the combination of the BUSON and the NBUSON bits causes the upstream to be disconnected from the downstream bus, then that is done. So in this case, the values of the MYBUS and the NMYBUS do not matter.

- If a master was connected to the downstream bus prior to the disconnect, then an interrupt is sent on the respective interrupt output in an attempt to let that master know that it is no longer connected to the downstream bus. This is indicated by setting the BUSLOST bit in the Interrupt Status Register.
- If the combination of the BUSON and the NBUSON bits causes a master to be connected to the downstream bus and if there is no change in the BUSON bits since when the disconnect took effect, then the master requesting the bus is connected to the downstream bus. If it requests a bus initialization sequence, then it is performed.
- If there is no change in the combination of the BUSON and the NBUSON bits and a new master wants the bus, then the downstream bus is disconnected from the old master that was using it and the new master gets control of it. Again, the bus initialization if requested is done. The appropriate interrupt signals are generated.

After a master has sent the bus control request:

1. The previous master is disconnected from the I²C-bus. An interrupt to the previous master is sent through its $\overline{\text{INT}}$ line to let it know that it lost control of the bus. BUSLOST bit in the Interrupt Status Register is set. This interrupt can be masked by setting the BUSLOSTMSK bit to ‘1’.
2. A built-in bus initialization/recovery function can take temporary control of the downstream channel to initialize the bus before making the actual switch to the new bus master. This function is activated by setting the BUSINIT to ‘1’ by the master during the same write sequence as the one programming MYBUS and BUSON bits.

   When activated and whether the bus was previously idle or not:

   a. 9 clock pulses are sent on the SCL_SLAVE.

   b. SDA_SLAVE line is released (HIGH) when the clock pulses are sent to SCL_SLAVE. This is equivalent to sending 8 data bits and a not acknowledge.

   c. Finally a STOP condition is sent to the downstream slave channel.

   This sequence will complete any read transaction which was previously in process and the downstream slave configured as a slave-transmitter should release the SDA line because the PCA9541 did not acknowledge the last byte.
3. When the initialization has been requested and completed, the PCA9541 sends an interrupt to the new master through its $\overline{\text{INT}}$ line and connects the new master to the downstream channel. BUSINIT bit in the Interrupt Status Register is set. **The switch operation occurs after the master asking the bus control has sent a STOP command.** This interrupt can be masked by setting the BUSINITMSK bit to ‘1’.
4. When the bus initialization/recovery function has not been requested (BUSINIT = 0), the PCA9541 connects the new master to the slave downstream channel. **The switch operation occurs after the master asking the bus control has sent a STOP command.** PCA9541 sends an interrupt to the new master through its $\overline{\text{INT}}$ line if the built-in bus sensor function detects a non-idle condition in the downstream slave channel at the switching time. BUSOK bit in the Interrupt Status Register is set. This means that a STOP condition has not been detected in the previous bus communication and that an external bus recovery/initialization must be performed. If an idle condition has been detected at the switching time, no interrupt will be sent. This interrupt can be masked by setting the BUSOKMSK bit to ‘1’.

Interrupt status can be read. See Section 8.4 “Interrupt Status registers” for more information.

The MYTEST and the NMYTEST bits cause the interrupt pins of the respective masters to be activated for a 'functional interrupt test'.

Remark: The regular way to proceed is that a master asks to take the control of the bus by programming MYBUS and BUSON bits based on NMUYBUS and NBUSON values. Nevertheless, the same master can also decide to give up the control of the bus and give it to the other master. This is also done by programming the MYBUS and BUSON bits based on NMYBUS and NBUSON values.

Remark: Any writes either to the Interrupt Enable Register or the Control Register cause the respective register to be updated on the 9th clock cycle, that is, on the rising edge of the acknowledge clock cycle.

Remark: The actual switch from one channel to another or the switching off of both the channels happens on a STOP command that is sent by the master requesting the switch.

8.3.1 Register 0: Interrupt Enable (IE) register (B1:B0 = 00)

This register allows a master to read and/or write (if needed) Mask options for its own channel.

The Interrupt Enable register described below is identical for both the masters. Nevertheless, there are physically 2 internal Interrupt Enable registers, one for each upstream channel. When Master 0 reads/writes in this register, the internal Interrupt Enable Register 0 will be accessed. When Master 1 reads/writes in this register, the internal Interrupt Enable Register 1 will be accessed.

Table 5: Register 0 - Interrupt Enable (IE) register (B1:B0 = 00) bit allocation

7	6	5	4	3	2	1	0
0	0	0	0	BUSLOSTMSK	BUSOKMSK	BUSINITMSK	INTINMSK

Table 6: Register 0 - Interrupt Enable (IE) register bit description

*Legend: * default value*

Bit	Symbol	Access	Value [1]	Description
7:4	-	R only	0*	not used
3	BUSLOSTMSK	R/W	0*	An interrupt on $\overline{INT}$ will be generated after the other master has been disconnected.
			1	An interrupt on $\overline{INT}$ will not be generated after the other master has been disconnected.
2	BUSOKMSK	R/W	0*	After connection is requested and Bus Initialization not requested (BUSINIT = 0), an interrupt on $\overline{INT}$ will be generated when a non-idle situation has been detected on the downstream slave channel by the bus sensor at the switching moment. **Remark:** Channel switching is done automatically after the STOP command.
			1	After connection is requested and Bus Initialization not requested (BUSINIT = 0), an interrupt on $\overline{INT}$ will not be generated when a non-idle situation has been detected on the downstream slave channel by the bus sensor at the switching moment (masked). **Remark:** Channel switching is done automatically after the STOP command.

Table 6: Register 0 - Interrupt Enable (IE) register bit description *…continued*

*Legend: * default value*

Bit	Symbol	Access	Value [1]	Description
1	BUSINITMSK	R/W	0*	After connection is requested and Bus Initialization requested (BUSINIT = 1), an interrupt on $\overline{INT}$ will be generated when the bus initialization is done. **Remark:** Channel switching is done after bus initialization completed.
			1	After connection is requested and Bus Initialization requested (BUSINIT = 1), an interrupt on $\overline{INT}$ will not be generated when the bus initialization is done (masked). **Remark:** Channel switching is done after bus initialization completed.
0	INTINMSK	R/W	0*	Interrupt on $\overline{INT\_IN}$ will generate an interrupt on $\overline{INT}$.
			1	Interrupt on $\overline{INT\_IN}$ will not generate an interrupt on $\overline{INT}$ (masked)

[1] Default values are the same for PCA9541/01, PCA9541/02, PCA9541/03.

8.3.2 Register 1: Control Register (B1:B0 = 01)

The Control Register described below is identical for both the masters. Nevertheless, there are physically 2 internal Control Registers, one for each upstream channel. When master 0 reads/writes in this register, the internal Control Register 0 will be accessed. When master 1 reads/writes in this register, the internal Control Register 1 will be accessed.

Table 7: Register 1 - Control Register (B1:B0 = 01) bit allocation

7	6	5	4	3	2	1	0
NTESTON	TESTON	0	BUSINIT	NBUSON	BUSON	NMYBUS	MYBUS

Table 8: Register 1 - Control Register (B1:B0 = 01) bit description

*Legend: * default value*

Bit	Symbol	Access	Value	Description
7	NTESTON	R/W	0*	A logic level HIGH to the $\overline{INT}$ line of the other channel is sent (interrupt cleared).
			1	A logic level LOW to the $\overline{INT}$ line of the other channel is sent (interrupt generated).
6	TESTON	R/W	0*	A logic level HIGH to the $\overline{INT}$ line is sent (interrupt cleared).
			1	A logic level LOW to the $\overline{INT}$ line is sent (interrupt generated).
5	-	R only	0*	not used
4	BUSINIT	R/W	0*	Bus initialization is not requested.
			1	Bus initialization is requested.
3	NBUSON	R only	see Table 11	NBUSON bit along with BUSON bit decides whether any upstream channel is connected to the downstream channel or not. See Table 10, Table 11, and Table 12.
2	BUSON	R/W	see Table 11	BUSON bit along with the NBUSON bit decides whether any upstream channel is connected to the downstream channel or not. See Table 10, Table 11, and Table 12.
1	NMYBUS	R only	see Table 11	NMYBUS bit along with MYBUS bit decides which upstream channel is connected to the downstream channel. See Table 9, Table 11, and Table 12.
0	MYBUS	R/W	see Table 11	MYBUS bit along with the NMYBUS bit decides which upstream channel is connected to the downstream channel. See Table 9, Table 11, and Table 12.

[1] Default values are the same for PCA9541/01, PCA9541/02, PCA9541/03.

Table 9: MYBUS and NMYBUS truth table

As a master reads its Control Register

NMYBUS [1]	MYBUS [1]	Slave channel
0	0	The master reading this combination has control of the bus.
1	0	The master reading this combination does not have control of the bus.
0	1	The master reading this combination does not have control of the bus.
1	1	The master reading this combination has control of the bus.

[1] MYBUS and NMYBUS is an exclusive-OR type function where:

Equal values ('00' or '11') means that the master reading its Control Register has control of the bus.

Different values ('01' or '10') means that the master reading its Control Register does not have control of the bus.

Table 10: BUSON and NBUSON truth table

NBUSON [1]	BUSON [1]	Slave channel
0	0	off
1	0	on
0	1	on
1	1	off

[1] BUSON and NBUSON is an exclusive-OR type function where:

Equal values ('00' or '11') means that the connection between the upstream and the downstream channels is off.

Different values ('01' or '10') means that the connection between the upstream and the downstream channels is on.

Switch to the new channel is done when the master initiating the switch request sends a STOP command to the PCA9541.

If either master wants to change the connection of the downstream channel, it needs to write to **its Control Register (Reg#01), and then send a STOP command** because an update of the connection to the downstream according to the values in the two internal Control Registers happens only on a STOP command. Writing to one control register followed by a STOP condition on the other master's channel will not cause an update to the downstream connection.

When both masters request a switch to their own channel at the same time, the master who last wrote to its Control Register before the PCA9541 receives a STOP command wins the switching sequence. There is no arbitration performed.

The Auto Increment feature (AI = 1) allows to program the PCA9541 in 4 bytes:

```
Start
111A3A2A1A0 + 0     PCA9541 Address + Write
00010000            Select Reg#00 with AI = 1
Data Reg#00         Interrupt Enable Register data
Data Reg#01         Control Register data
Stop
```

Table 11: Default Control Register values

Type version	Master	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		NTESTON	TESTON	not used	BUSINIT	NBUSON	BUSON	NMYBUS	MYBUS
PCA9541/01	MST_0	0	0	0	0	0	1	0	0
	MST_1	0	0	0	0	1	0	1	0
PCA9541/02 at power-up	MST_0	0	0	0	0	0	0	0	0
	MST_1	0	0	0	0	0	0	1	0
PCA9541/02 after STOP	MST_0	0	0	0	0	0	1	0	0
	MST_1	0	0	0	0	1	0	1	0
PCA9541/03	MST_0	0	0	0	0	0	0	0	0
	MST_1	0	0	0	0	0	0	1	0

Table 12 describes which command needs to be written to the Control Register when a master device wants to take control of the I²C-bus. Byte written to the Control Register is a function of the current I²C-bus control status performed after an initial reading of the Control Register.

Current status of the I²C-bus is determined by the bits MYBUS, NMYBUS, BUSON and NBUSON is one of the following:

- The master reading its Control Register does not have control and the I²C-bus is off.
- The master reading its Control Register does not have control and the I²C-bus is on.
- The master reading its Control Register has control and the I²C-bus is off.
- The master reading its Control Register has control and the I²C-bus is on.

'I²C-bus off' means that upstream and downstream channels are not connected together.

'I²C-bus on' means that upstream and downstream channels are connected together.

Remark: Only the 4 LSBs of the Control Register are described in Table 12 since only those bits control the I²C-bus control. The logic value for the 4 MSBs is specific to the application and are not discussed in the table.

The read sequence is performed by the master as:
S - 111xxxx0 - 000x0001 - Sr - 111xxxx1 - DataRead - P

The write sequence is performed by the master as:
S - 111xxxx0 - 000x0001 - DataWritten - P

Table 12: Bus control sequence

Read Control Register performed by the master							Write Control Register performed by the master					
Byte read [1]	Status		NBUSON	BUSON	NMYBUS	MYBUS	Byte written [1]	Action performed to take mastership	NBUSON [3]	BUSON	NMYBUS [3]	MYBUS
Hex							Hex					
0	bus off	has control	0	0	0	0	4	bus on	x	1	x	0
1	bus off	no control	0	0	0	1	4	bus on, take control	x	1	x	0
2	bus off	no control	0	0	1	0	5	bus on, take control	x	1	x	1
3	bus off	has control	0	0	1	1	5	bus on	x	1	x	1
4	bus on	has control	0	1	0	0	-	no change	no write required			
5	bus on	no control	0	1	0	1	4	take control	x	1	x	0
6	bus on	no control	0	1	1	0	5	take control	x	1	x	1
7	bus on	has control	0	1	1	1	-	no change	no write required			
8	bus on	has control	1	0	0	0	-	no change	no write required			
9	bus on	no control	1	0	0	1	0	take control	x	0	x	0
A	bus on	no control	1	0	1	0	1	take control	x	0	x	1
B	bus on	has control	1	0	1	1	-	no change	no write required			
C	bus off	has control	1	1	0	0	0	bus on	x	0	x	0
D	bus off	no control	1	1	0	1	0	bus on, take control	x	0	x	0
E	bus off	no control	1	1	1	0	1	bus on, take control	x	0	x	1
F	bus off	has control	1	1	1	1	1	bus on	x	0	x	1

[1] Only the 4 LSBs are shown.

[2] x0x0 in binary = 0, 2, 8 or A in hexadecimal
x0x1 in binary = 1, 3, 9 or B in hexadecimal
x1x0 in binary = 4, 6, C or E in hexadecimal
x1x1 in binary = 5, 7, D or F in hexadecimal

[3] x can be either '0' or '1' since those bits are read-only bits.

8.4 Interrupt Status registers

The PCA9541 provides 4 different types of interrupt:

- To indicate to the former I²C-bus master that it is not in control of the bus anymore
- To indicate to the new I²C-bus master that:
  - The bus recovery/initialization has been performed and that the downstream channel connection has been done (built-in bus recovery/initialization active).
  - A ‘bus not well initialized’ condition has been detected by the PCA9541 when the switch has been done (built-in bus recovery/initialization not active). This information can be used by the new master to initiate its own bus recovery/initialization sequence.
- Indicate to both I²C-bus upstream masters that a downstream interrupt has been generated through the $\overline{\text{INT\_IN}}$ pin.
- Functionality wiring test.

8.4.1 Bus control lost interrupt

When an upstream master takes control of the I²C-bus while the other channel was using the downstream channel, an interrupt is generated to the master losing control of the bus ($\overline{\text{INT}}$ line goes LOW to let the master know that it lost the control of the bus) immediately after disconnection from the downstream channel.

By setting the BUSLOSTMSK bit to ‘1’, the interrupt is masked and the upstream master that lost the I²C-bus control does not receive an interrupt ($\overline{\text{INT}}$ line does not go LOW).

8.4.2 Recovery/initialization interrupt

Before switching to a new upstream channel, an automatic bus recovery/initialization can be performed by the PCA9541. This function is requested by setting the BUSINIT bit to ‘1’. When the downstream bus has been initialized, an interrupt to the new master is generated ($\overline{\text{INT}}$ line goes LOW).

By setting the BUSINITMSK bit to ‘1’, the interrupt is masked and the new master does not receive an interrupt ($\overline{\text{INT}}$ line does not go LOW).

When the automatic bus recovery/initialization is not requested, if the built-in bus sensor function (sensing permanently the downstream I²C-bus traffic) detects a non-idle condition (previous bus channel connected to the downstream slave channel, was between a START and STOP condition), then an interrupt to the new master is sent ($\overline{\text{INT}}$ line goes LOW). This interrupt tells the new master that an external bus recovery/initialization must be performed. By setting the BUSOKMSK bit to ‘1’, the interrupt is masked and the new master does not receive an interrupt ($\overline{\text{INT}}$ line does not go LOW).

Remark: In this particular situation, after the switch to the new master is performed, **a read of the Interrupt Status Register is not possible if the switch happened in the middle of a read sequence** because the new master does not have control of the SDA line.

8.4.3 Downstream interrupt

An interrupt can also be generated by a downstream device by asserting the $\overline{INT\_IN}$ pin LOW. When $\overline{INT\_IN}$ is asserted LOW and if both INTINMSK bits are not set to '1' by either master, $\overline{INT0}$ and $\overline{INT1}$ both go LOW.

By setting the INTINMSK bit to '1' by a master and/or the INTINMSK bit to '1' by the other master, the interrupt(s) is (are) masked and the corresponding masked channel(s) does (do) not receive an interrupt ($\overline{INT0}$ and/or $\overline{INT1}$ line does (do) not go LOW).

8.4.4 Functional test interrupt

A master can send an interrupt to itself to test its own $\overline{INT}$ wire or send an interrupt to the other master to test its $\overline{INT}$ line. This is done by:

- setting the TESTON bit to '1' to test its own $\overline{INT}$ line
- setting the NTESTON bit to '1' to test the other master $\overline{INT}$ line

Setting the TESTON and/or NTESTON bits to '0' by a master will clear the interrupt(s).

Remark: Interrupt outputs have an open-drain structure. Interrupt input does not have any internal pull-up resistor and must not be left floating (that is, pulled HIGH to V_{DD} through resistor) in order to avoid any undesired interrupt conditions.

8.4.5 Register 2: Interrupt Status Register (B1:B0 = 10)

The Interrupt Status Register for both the masters is identical and is described below. Nevertheless, there are physically 2 internal Interrupt Registers, one for each upstream channel.

When Master 0 reads this register, the internal Interrupt Register 0 will be accessed.

When Master 1 reads this register, the internal Interrupt Register 1 will be accessed.

Table 13: Register 2 - Interrupt Status register (B1:B0 = 10) bit allocation

7	6	5	4	3	2	1	0
NMYTEST	MYTEST	0	0	BUSLOST	BUSOK	BUSINIT	INTIN

Table 14: Register 0 - Interrupt Enable (IE) register bit description

*Legend: * default value*

Bit	Symbol	Access	Value [1]	Description
7	NMYTEST [2]	R only	0*	no interrupt generated due to NTESTON bit from the other master (NTESTON = 0 from the other master) [3]
			1	interrupt generated due to TESTON bit from the other master (NTESTON = 1 from the other master) [3]
6	MYTEST [2]	R only	0*	no interrupt generated by TESTON bit (TESTON = 0) [3]
			1	interrupt generated by TESTON bit (TESTON = 1) [3]
5	-	R only	0*	not used
4	-	R only	0*	not used
3	BUSLOST [4]	R only	0*	no interrupt generated to the previous master when switching to the new one is initiated
			1	interrupt generated to the previous master when switching to the new one is initiated

Table 14: Register 0 - Interrupt Enable (IE) register bit description *…continued*

*Legend: * default value*

Bit	Symbol	Access	Value [1]	Description
2	BUSOK [4]	R only	0*	no interrupt generated by bus sensor function
			1	interrupt generated by bus sensor function (masked when bus recovery/initialization requested) - Bus was not idle when the switch occurred
1	BUSINIT [4]	R only	0*	no interrupt generated by the bus recovery/initialization function
			1	interrupt generated by the bus recovery/initialization function; recovery/initialization done
0	INTIN [2]	R only	0*	no interrupt on interrupt input ($\overline{INT\_IN}$) [5]
			1	interrupt on interrupt input ($\overline{INT\_IN}$) [5]

[1] Default values are the same for PCA9541/01, PCA9541/02, and PCA9541/03.

[2] Reading the Interrupt Status Register does not clear the MYTEST, NMYTEST or the INTIN bits. They are cleared if:
$\overline{INT\_IN}$ lines goes HIGH for INTIN bit
TESTON bit is cleared for MYTEST bit
NTESTON bit is cleared for NMYTEST bit

[3] Interrupt on a master is cleared after TESTON bit is cleared by the same master or NTESTON bit is cleared by the other master.

[4] BUSINIT, BUSOK and BUSLOST bits in the Interrupt Status Register get cleared after a read of the same register is done. Precisely, the register gets cleared on the second clock pulse during the read operation.

[5] If the interrupt condition remains on $\overline{INT\_IN}$ after the read sequence, another interrupt will be generated (if the interrupt has not been masked).

8.5 Power-on reset

When power is applied to V_{DD}, an internal power-on reset holds the PCA9541 in a reset condition until V_{DD} has reached V_{POR}. At this point, the reset condition is released and the internal registers are initialized to their default states, with:

- PCA9541/01: default Channel 0 (no STOP detect)

  After power-up and/or insertion of the device in the main I²C-bus, the upstream Channel 0 and the downstream slave channel are connected together.

- PCA9541/02: default Channel 0 (STOP detect)

  After power-up and/or insertion of the device in the main I²C-bus, the upstream Channel 0 and the downstream slave channel are connected together after a STOP condition has been detected by the PCA9541/02 on Channel 0.

  – If the bus was not idle, Channel 0 and the downstream slave device will be connected together as soon as a STOP condition occurs at the conclusion of the transmission sequence on Channel 0.

  – If the bus was idle, then Channel 0 is connected to the downstream slave channel after a STOP condition is detected on Channel 0. This I²C-bus command may or may not be addressed to the PCA9541/02.

  – If a switch to Channel 1 (initiated by the master on Channel 1) is requested (before or after the default switch to Channel 0 has been performed), the upstream Channel 1 is connected to the downstream slave channel when the master located in Channel 1 sends the STOP command.

- PCA9541/03: default ‘no channel’ (no STOP detect)

  After power-up and/or insertion of the device in the main I²C-bus, no channel will be connected to the downstream channel. The device is ready to receive a START condition and its address by a master.

  If either register writes to its Control Register, then the connection between the upstream and the downstream channels is determined by the values on the Control Registers.

Thereafter, V_{DD} must be lowered below 0.2 V to reset the device.

8.6 External reset

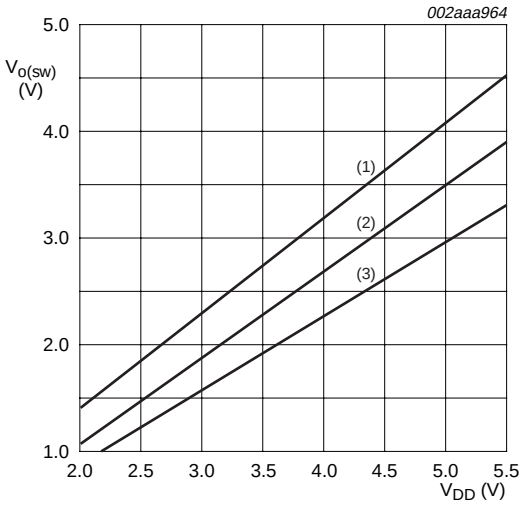
A reset can be accomplished by holding the $\overline{\text{RESET}}$ pin LOW for a minimum of t_{WL}. The PCA9541 registers and I²C-bus state machine will be held in their default states until the $\overline{\text{RESET}}$ input is once again HIGH. This input typically requires a pull-up resistor to V_{DD}.

Default states are:

- I²C-bus upstream Channel 0 connected to the I²C-bus downstream channel for the PCA9541/01
- no I²C-bus upstream channel connected to the I²C-bus downstream channel for the PCA9541/02 with Channel 0 connected to the downstream I²C-bus channel after detection of a STOP on the upstream channel
- no I²C-bus upstream channel connected to the I²C-bus downstream channel for the PCA9541/03.

8.7 Voltage translation

The pass gate transistors of the PCA9541 are constructed such that the V_{DD} voltage can be used to limit the maximum voltage that will be passed from one I²C-bus to another.



(1) maximum
(2) typical
(3) minimum

Fig 8. Pass gate voltage as a function of supply voltage

Figure 8 shows the voltage characteristics of the pass gate transistors (note that the graph was generated using the data specified in Section 12 “Static characteristics” of this data sheet). In order for the PCA9541 to act as a voltage translator, the $V_{o(sw)}$ voltage should be equal to, or lower than the lowest bus voltage. For example, if the main buses were running at 5 V, and the downstream bus was 3.3 V, then $V_{o(sw)}$ should be equal to or below 3.3 V to effectively clamp the downstream bus voltages. Looking at Figure 8, we see that $V_{o(sw)(max)}$ will be at 3.3 V when the PCA9541 supply voltage is 3.5 V or lower so the PCA9541 supply voltage could be set to 3.3 V. Pull-up resistors can then be used to bring the bus voltages to their appropriate levels (see Figure 17).

More Information on voltage translation can be found in Application Note *AN262: PCA954X family of I2C/SMBus multiplexers and switches*.

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 9).

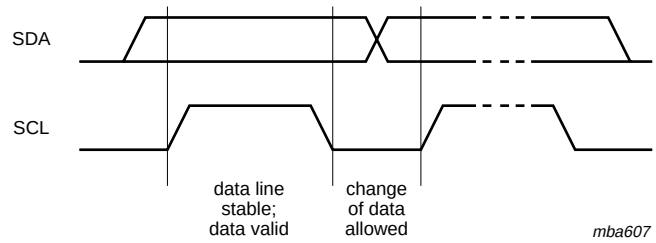


Fig 9. Bit transfer

9.2 START and STOP conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line, while the clock is HIGH is defined as the START condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the STOP condition (P) (see Figure 10).

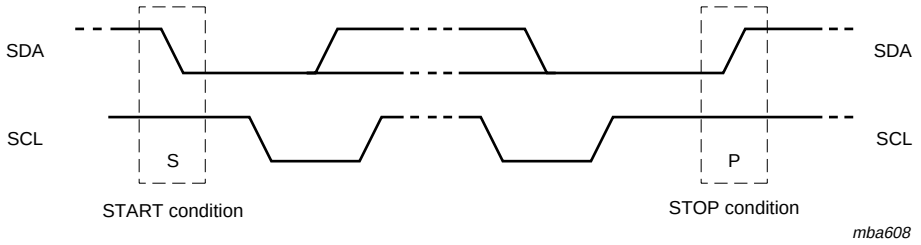


Fig 10. Definition of START and STOP conditions

9.3 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 11).

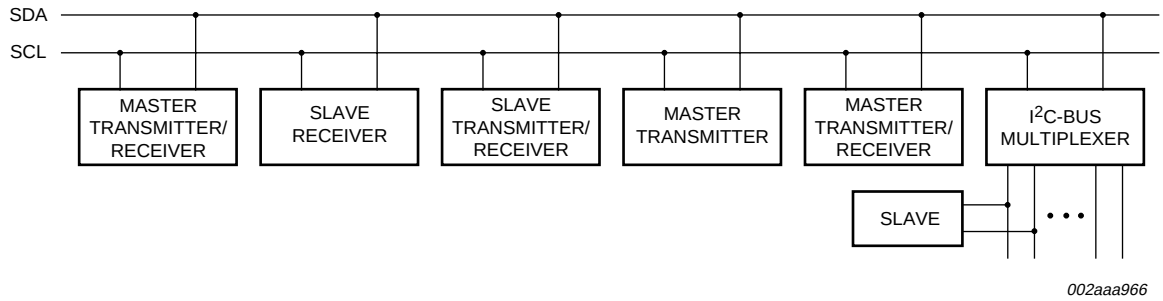


Fig 11. System configuration

9.4 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; setup 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.

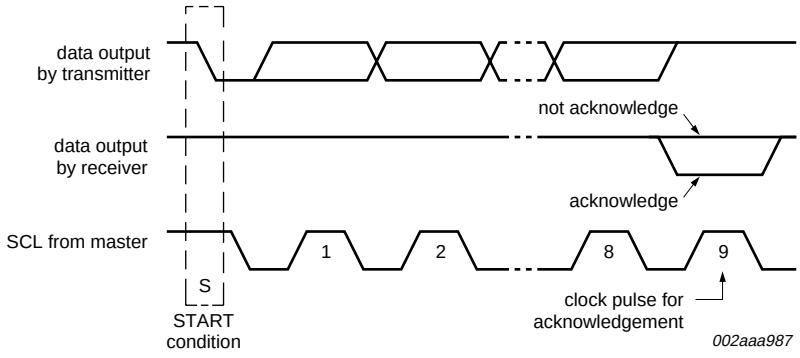


Fig 12. Acknowledgement on the I²C-bus

9.5 Bus transactions

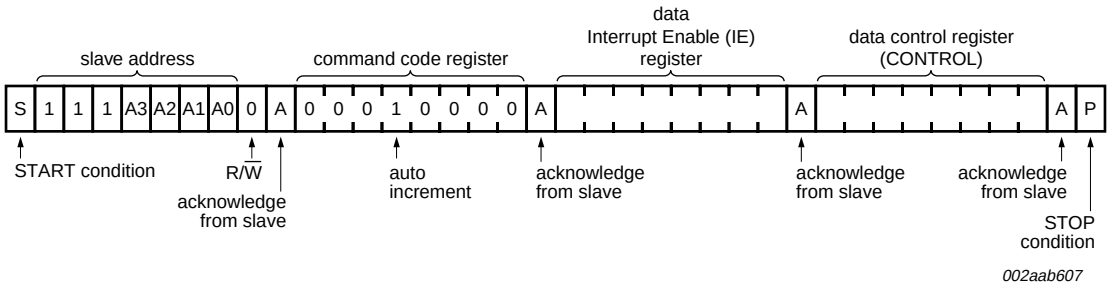
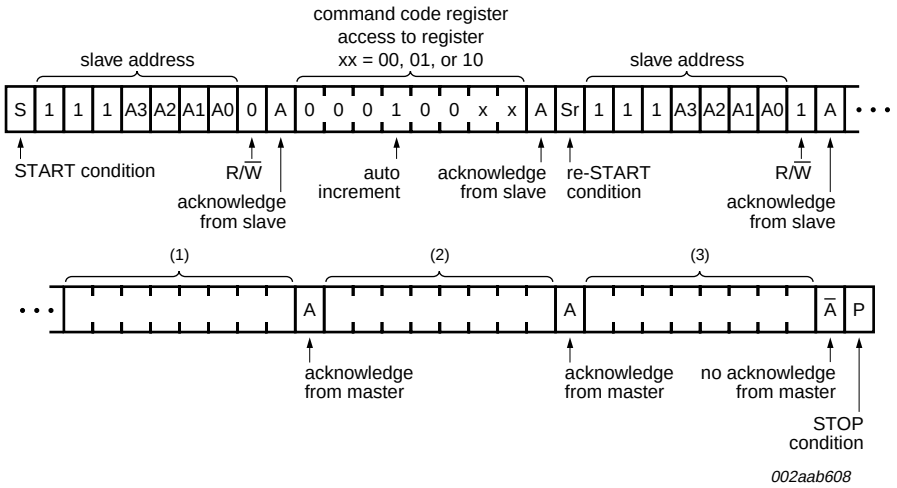


Fig 13. Write to the Interrupt Enable and Control registers using the Auto-Increment (AI) bit

Remark: If a third data byte is sent, it will not be acknowledged by the PCA9541.



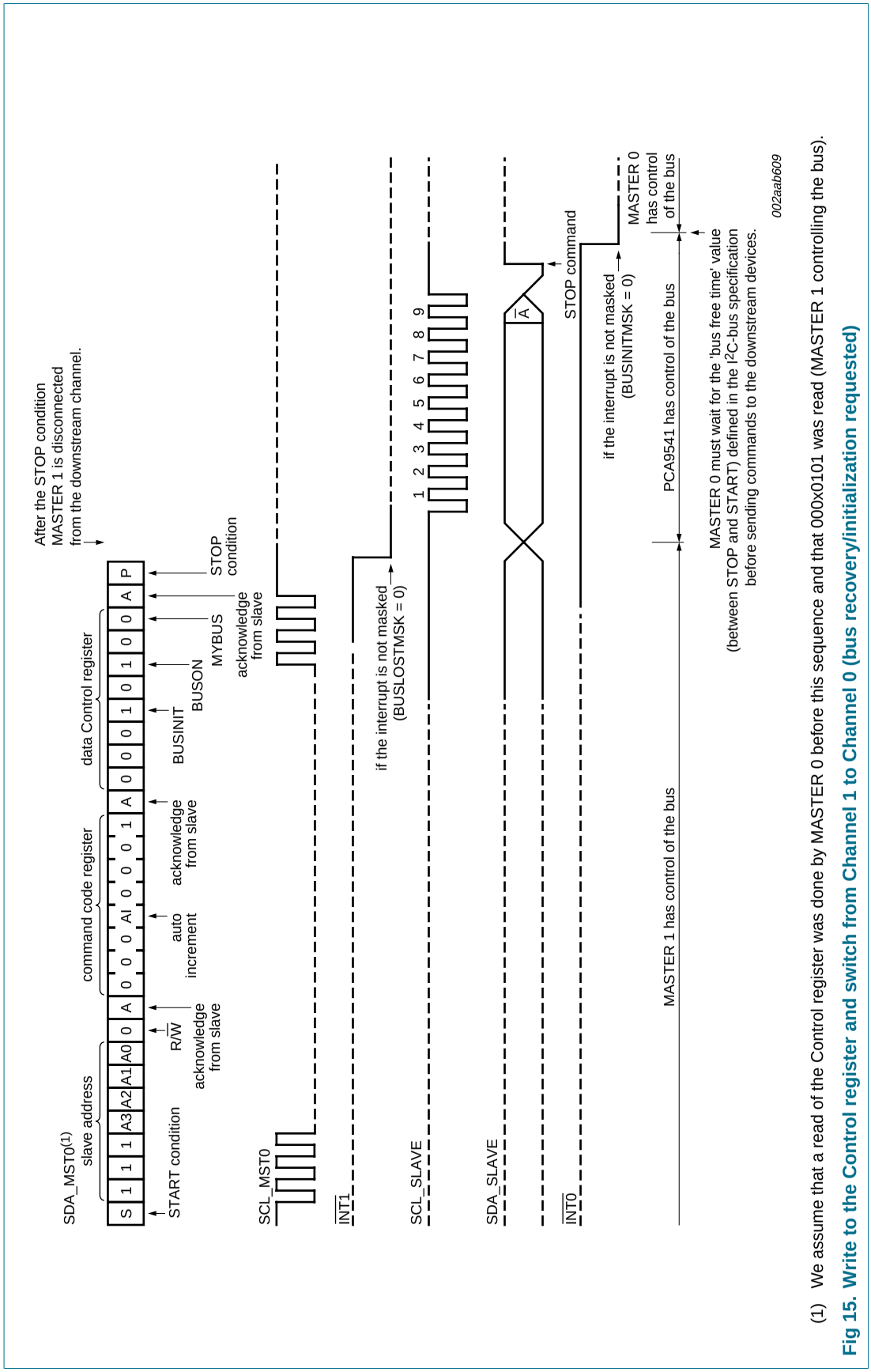
(1) xx = 00: Interrupt Enable register
xx = 01: Control register
xx = 10: INT register

(2) xx = 00: Control register
xx = 01: INT register
xx = 10: Interrupt Enable register

(3) xx = 00: INT register
xx = 01: Interrupt Enable register
xx = 10: Control register

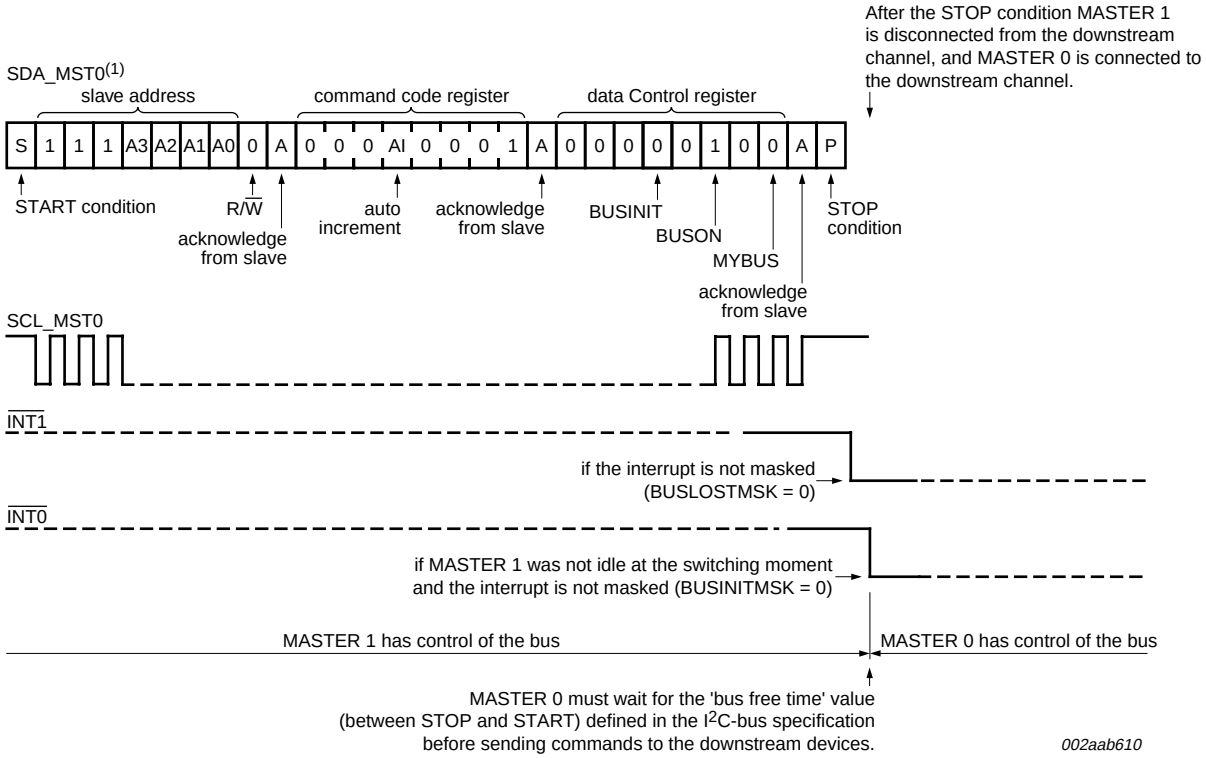
Fig 14. Read the 3 registers using the Auto-Increment (AI) bit

Remark: If a fourth data byte is read, the first register will be accessed.



(1) We assume that a read of the Control register was done by MASTER 0 before this sequence and that 000x0101 was read (MASTER 1 controlling the bus).

Fig 15. Write to the Control register and switch from Channel 1 to Channel 0 (bus recovery/initialization requested)



(1) We assume that a read of the Control register was done by MASTER 0 before this sequence and that 000x0101 was read (MASTER 1 controlling the bus).

Fig 16. Write to the Control register and switch from Channel 1 to Channel 0 (bus recovery/initialization not requested)

10. Application design-in information

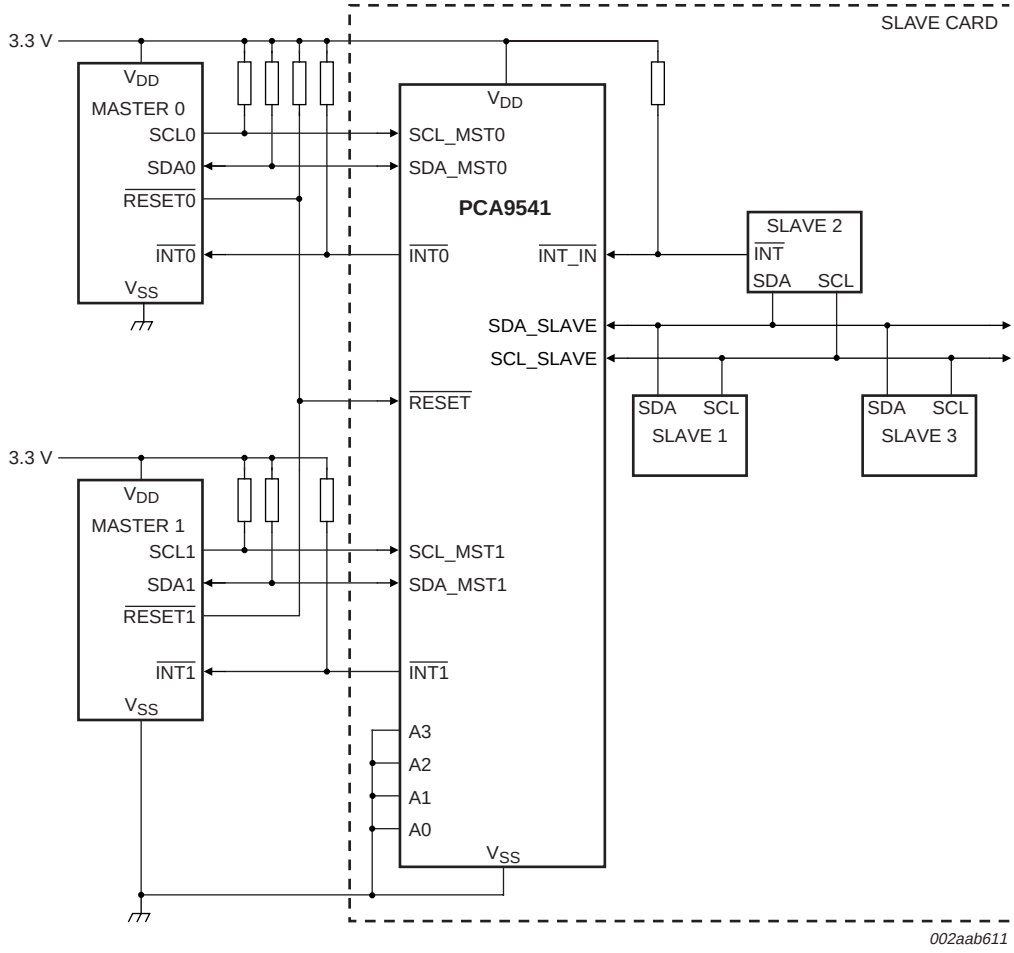


Fig 17. Typical application

10.1 Specific applications

The PCA9541 is a 2-to-1 I²C-bus master selector designed for dual master, high reliability I²C-bus applications, where continuous maintenance and control monitoring is required even if one master fails or its controller card is removed for maintenance. The PCA9541 can also be used in other applications, such as where masters share the same resource but cannot share the same bus, as a gatekeeper multiplexer in long single bus applications or as a bus initialization/recovery device.

10.2 High reliability systems

In a typical multipoint application, shown in Figure 18, the two masters (for example, primary and back-up) are located on separate I²C-buses that connect to multiple downstream I²C-bus slave cards/devices via a PCA9541/01 for non-hot swap applications or the PCA9541/02 for hot swap applications to provide high reliability of the I²C-bus.

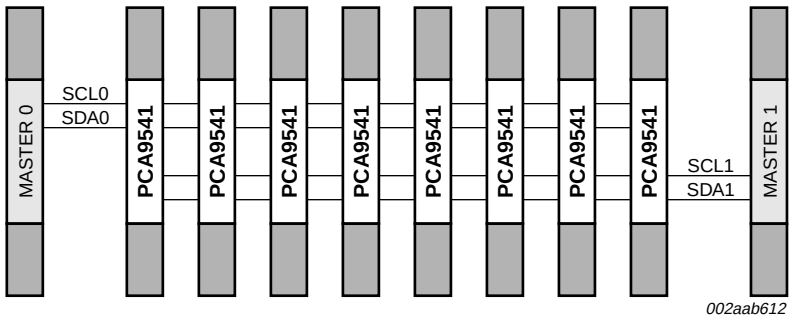


Fig 18. High reliability backplane application

I²C-bus commands are sent via the primary or back-up master and either master can take command of the I²C-bus. Either master at any time can gain control of the slave devices if the other master is disabled or removed from the system. The failed master is isolated from the system and will not affect communication between the on-line master and the slave devices located on the cards.

For even higher reliability in multipoint backplane applications, two dedicated masters can be used for every card as shown in Figure 19.

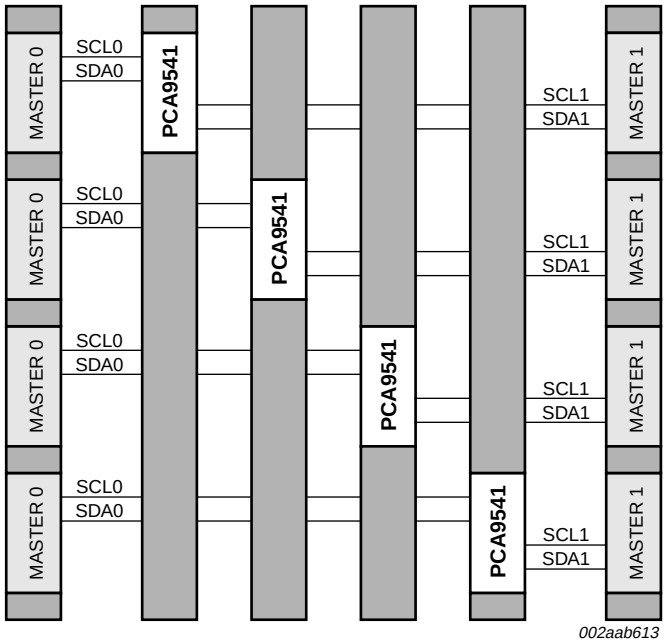


Fig 19. Very high reliability backplane application

10.3 Masters with shared resources

Some masters may not be multi-master capable or some masters may not work well together and continually lock up the bus. The PCA9541 can be used to separate the masters, as shown in Figure 20, but still allow shared access to slave devices, such as Field Replaceable Unit (FRU) EEPROMs or temperature sensors.

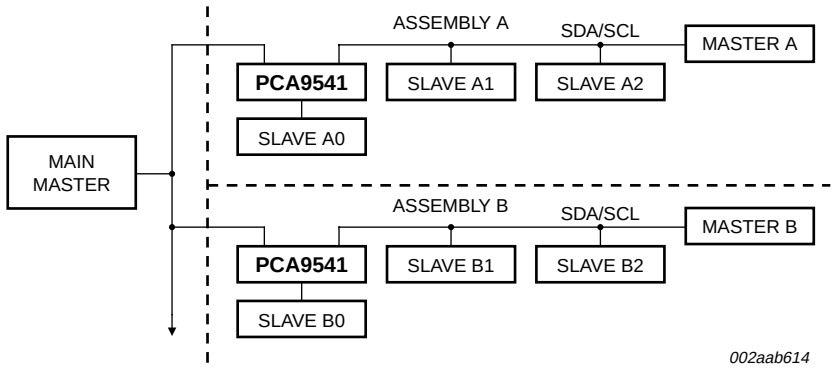


Fig 20. Masters with shared resources application

10.4 Gatekeeper multiplexer

The PCA9541/03 can act as a gatekeeper multiplexer in applications where there are multiple I²C-bus devices with the same fixed address (for example, EEPROMs with address of 'Z' as shown in Figure 21) connected in a multipoint arrangement to the same I²C-bus. Up to 16 hot swappable cards/devices can be multiplexed to the same bus master by using one PCA9541/03 per card/device. Since each PCA9541/03 has its own unique address (for example, 'A', 'B', 'C', and so on), the EEPROMs can be connected to the master, one at a time, by connecting one PCA9541/03 (Master 0 position) while keeping the rest of the cards/devices isolated (off position).

The alternative, shown with dashed lines, is to use a PCA9548 1-to-8 channel switch on the master card and run 8 I²C-bus devices, one to each EEPROM card, to multiplex the master to each card. The number of card pins used is the same in either case, but there are 7 less pairs of SDA/SCL traces on the printed-circuit board if the PCA9541/03 is used.

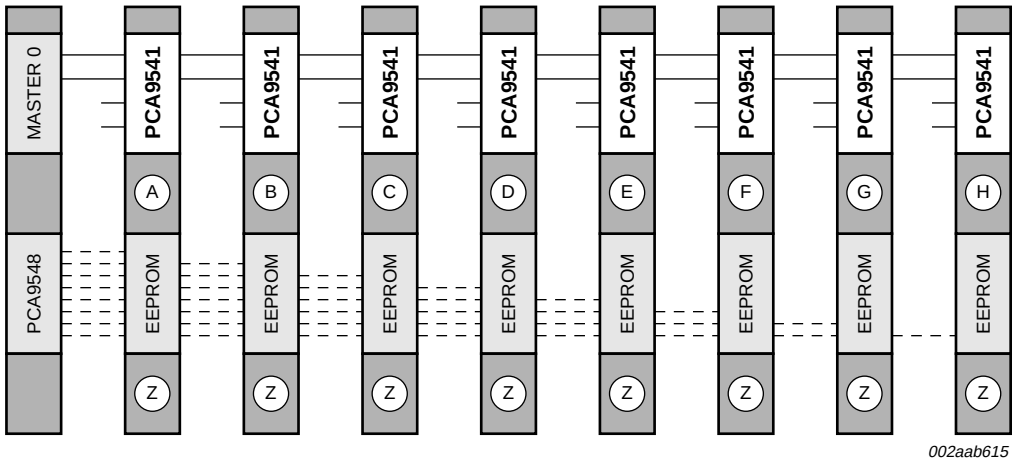


Fig 21. Gatekeeper multiplexer application

10.5 Bus initialization/recovery to initialize slaves without hardware reset

If the I²C-bus is hung, I²C-bus devices without a hardware reset pin (for example, Slave 1 and Slave 2 in Figure 22) can be isolated from the master by the PCA9541/03. The PCA9541/03 disconnects the bus when it is reset via the hardware reset line, restoring the master's control of the rest of the bus (for example, Slave 0). The bus master can then command the PCA9541/03 to send 9 clock pulses/STOP condition to reset the downstream I²C-bus devices before they are reconnected to the master or leave the downstream devices isolated.

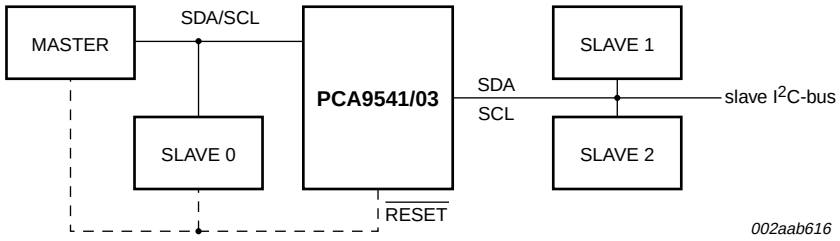


Fig 22. Bus initialization/recovery application

11. Limiting values

Table 15: Limiting values

In accordance with the Absolute Maximum Rating System (IEC 60134). Voltages are referenced to V_{SS} (ground = 0 V). [1]

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DD}	supply voltage		–0.5	+7.0	V
V_I	input voltage		–0.5	+7.0	V
I_I	input current		–20	+20	mA
I_O	output current		–25	+25	mA
I_{DD}	supply current		–100	+100	mA
I_{SS}	ground supply current		–100	+100	mA
P_{tot}	total power dissipation		-	400	mW
T_{stg}	storage temperature		–60	+150	°C
T_{amb}	ambient temperature	operating in free air	–40	+85	°C

[1] The performance capability of a high-performance integrated circuit in conjunction with its thermal environment can create junction temperatures which are detrimental to reliability. The maximum junction temperature of this integrated circuit should not exceed 150 °C.

12. Static characteristics

Table 16: Static characteristics (2.3 V to 3.6 V)

V_{DD} = 2.3 V to 3.6 V; V_{SS} = 0 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified. See Table 17 for V_{DD} = 3.6 V to 5.5 V.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Supply						
V_{DD}	supply voltage		2.3	-	3.6	V
I_{DD}	supply current	Operating mode; V_{DD} = 3.6 V; no load; $V_I = V_{DD}$ or V_{SS}; f_{SCL} = 100 kHz	-	152	200	µA
I_{stb}	standby current	Standby mode; V_{DD} = 3.6 V; no load; $V_I = V_{DD}$ or V_{SS}; f_{SCL} = 0 kHz	-	10	100	µA
V_{POR}	power-on reset voltage	no load; $V_I = V_{DD}$ or V_{SS} [1]	-	1.5	2.1	V
Input SCL_MSTn; input/output SDA_MSTn (upstream and downstream channels)						
V_{IL}	LOW-level input voltage		−0.5	-	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		$0.7V_{DD}$	-	6	V
I_{OL}	LOW-level output current	V_{OL} = 0.4 V	3	-	-	mA
		V_{OL} = 0.6 V	6	-	-	mA
I_L	leakage current	$V_I = V_{DD}$ or V_{SS}	−1	-	+1	µA
C_i	input capacitance	$V_I = V_{SS}$	-	4	5	pF
Select inputs A0 to A3, $\overline{\text{INT\_IN}}$, $\overline{\text{RESET}}$						
V_{IL}	LOW-level input voltage		−0.5	-	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		$0.7V_{DD}$	-	6	V
I_{LI}	input leakage current	$V_I = V_{DD}$ or V_{SS}	−1	-	+1	µA
C_i	input capacitance	$V_I = V_{SS}$	-	2	3	pF
Pass gate						
R_{on}	ON-state resistance	V_{DD} = 3.0 V to 3.6 V; V_O = 0.4 V; I_O = 15 mA	5	14	30	Ω
		V_{DD} = 2.3 V to 2.7 V; V_O = 0.4 V; I_O = 10 mA	7	17	55	Ω
$V_{o(sw)}$	switch output voltage	$V_{i(sw)} = V_{DD}$ = 3.3 V; $I_{o(sw)}$ = −100 µA	-	2.2	-	V
		$V_{i(sw)} = V_{DD}$ = 3.0 V to 3.6 V; $I_{o(sw)}$ = −100 µA	1.6	-	2.8	V
		$V_{i(sw)} = V_{DD}$ = 2.5 V; $I_{o(sw)}$ = −100 µA	-	1.5	-	V
		$V_{i(sw)} = V_{DD}$ = 2.3 V to 2.7 V; $I_{o(sw)}$ = −100 µA	1.1	-	2.0	V
I_L	leakage current	$V_I = V_{DD}$ or V_{SS}	−1	-	+1	µA
$\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ outputs						
I_{OL}	LOW-level output current	V_{OL} = 0.4 V	3	-	-	mA

[1] V_{DD} must be lowered to 0.2 V in order to reset part.

Table 17: Static characteristics (3.6 V to 5.5 V)

V_{DD} = 3.6 V to 5.5 V; V_{SS} = 0 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified. See Table 16 for V_{DD} = 2.3 V to 3.6 V.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Supply						
V_{DD}	supply voltage		3.6	-	5.5	V
I_{DD}	supply current	Operating mode; V_{DD} = 5.5 V; no load; $V_I = V_{DD}$ or V_{SS}; f_{SCL} = 100 kHz	-	349	600	µA
I_{stb}	standby current	Standby mode; V_{DD} = 5.5 V; no load; $V_I = V_{DD}$ or V_{SS}; f_{SCL} = 0 kHz	-	10	200	µA
V_{POR}	power-on reset voltage	no load; $V_I = V_{DD}$ or V_{SS} [1]	-	1.5	2.1	V
Input SCL_MSTn; input/output SDA_MSTn (upstream and downstream channels)						
V_{IL}	LOW-level input voltage		−0.5	-	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		$0.7V_{DD}$	-	6	V
I_{OL}	LOW-level output current	V_{OL} = 0.4 V	3	-	-	mA
		V_{OL} = 0.6 V	6	-	-	mA
I_{IL}	LOW-level input current	$V_I = V_{SS}$	−10	-	+10	µA
I_{IH}	HIGH-level input current	$V_I = V_{DD}$	-	-	100	µA
C_i	input capacitance	$V_I = V_{SS}$	-	4	6	pF
Select inputs A0 to A3, $\overline{INT\_IN}$, $\overline{RESET}$						
V_{IL}	LOW-level input voltage		−0.5	-	$0.3V_{DD}$	V
V_{IH}	HIGH-level input voltage		$0.7V_{DD}$	-	6	V
I_{LI}	input leakage current	$V_I = V_{DD}$ or V_{SS}	−1	-	+50	µA
C_i	input capacitance	$V_I = V_{SS}$	-	2	5	pF
Pass gate						
R_{on}	ON-state resistance	V_{DD} = 4.5 V to 5.5 V; V_O = 0.4 V; I_O = 15 mA	4	12	24	Ω
$V_{o(sw)}$	switch output voltage	$V_{i(sw)} = V_{DD}$ = 5.0 V; $I_{o(sw)}$ = −100 µA	-	3.6	-	V
		$V_{i(sw)} = V_{DD}$ = 4.5 V to 5.5 V; $I_{o(sw)}$ = −100 µA	2.6	-	4.5	V
I_L	leakage current	$V_I = V_{DD}$ or V_{SS}	−1	-	+100	µA
$\overline{INT0}$ and $\overline{INT1}$ outputs						
I_{OL}	LOW-level output current	V_{OL} = 0.4 V	3	-	-	mA

[1] V_{DD} must be lowered to 0.2 V in order to reset part.

13. Dynamic characteristics

Table 18: Dynamic characteristics

Symbol	Parameter	Conditions		Standard-mode I²C-bus		Fast-mode I²C-bus		Unit
				Min	Max	Min	Max	
t_{PD}	propagation delay	(SDA_MSTn to SDA_SLAVE) or (SCL_MSTn to SCL_SLAVE)	[1]	-	0.3	-	0.3	ns
f_{SCL}	SCL clock frequency			0	100	0	400	kHz
$f_{SCL(init/rec)}$	SCL clock frequency (bus initialization/bus recovery)			50	150	50	150	kHz
t_{BUF}	bus free time between a STOP and START condition			4.7	-	1.3	-	µs
$t_{HD;STA}$	hold time (repeated) START condition		[2]	4.0	-	0.6	-	µs
t_{LOW}	LOW period of the SCL clock			4.7	-	1.3	-	µs
t_{HIGH}	HIGH period of the SCL clock			4.0	-	0.6	-	µs
$t_{SU;STA}$	set-up time for a repeated START condition			4.7	-	0.6	-	µs
$t_{SU;STO}$	set-up time for STOP condition			4.0	-	0.6	-	µs
$t_{HD;DAT}$	data hold time			0 [3]	3.45	0 [3]	0.9	µs
$t_{SU;DAT}$	data set-up time			250	-	100	-	ns
t_r	rise time of both SDA and SCL signals			-	1000	$20 + 0.1C_b$ [4]	300	ns
t_f	fall time of both SDA and SCL signals			-	300	$20 + 0.1C_b$ [4]	300	µs
C_b	capacitive load for each bus line			-	400	-	400	pF
t_{SP}	pulse width of spikes that must be suppressed by the input filter			-	50	-	50	ns
$t_{VD;DAT}$	data valid time	HIGH-to-LOW	[5]	-	1	-	1	µs
		LOW-to-HIGH	[5]	-	0.6	-	0.6	µs
$t_{VD;ACK}$	data valid acknowledge time			-	1	-	1	µs
$\overline{INT}$								
$t_{v(INT\_IN-INTn)}$	valid time from pin $\overline{INT\_IN}$ to pin $\overline{INTn}$ signal			-	4	-	4	µs
$t_{d(INT\_IN-INTn)}$	delay time from pin $\overline{INT\_IN}$ to pin $\overline{INTn}$ inactive			-	2	-	2	µs
$t_{w(rej)L}$	LOW-level rejection time	$\overline{INT\_IN}$ input		1	-	1	-	µs
$t_{w(rej)H}$	HIGH-level rejection time	$\overline{INT\_IN}$ input		0.5	-	0.5	-	µs
$\overline{RESET}$								
$t_{w(rst)L}$	LOW-level reset time			4	-	4	-	ns
t_{rst}	reset time	SDA clear		500	-	500	-	ns
$t_{REC;STA}$	recovery time to START condition		[6] [7]	0	-	0	-	ns

[1] Pass gate propagation delay is calculated from the 20 Ω typical R_{on} and the 15 pF load capacitance.

[2] After this period, the first clock pulse is generated.

[3] A device must internally provide a hold time of at least 300 ns for the SDA signal (referred to the $V_{IH(min)}$ of the SCL signal) in order to bridge the undefined region of the falling edge of SCL.

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

[5] Measurements taken with 1 kΩ pull-up resistor and 50 pF load.

[6] Resetting the device while actively communicating on the bus may cause glitches or errant STOP conditions.

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

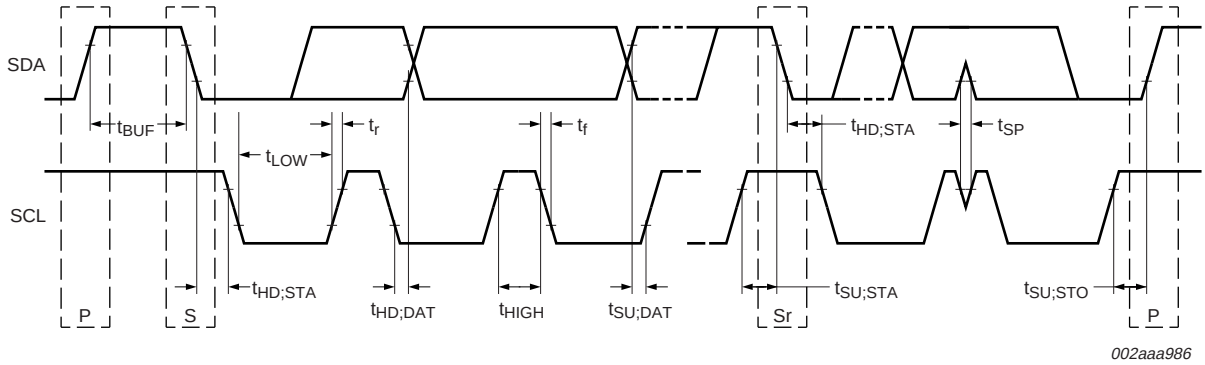
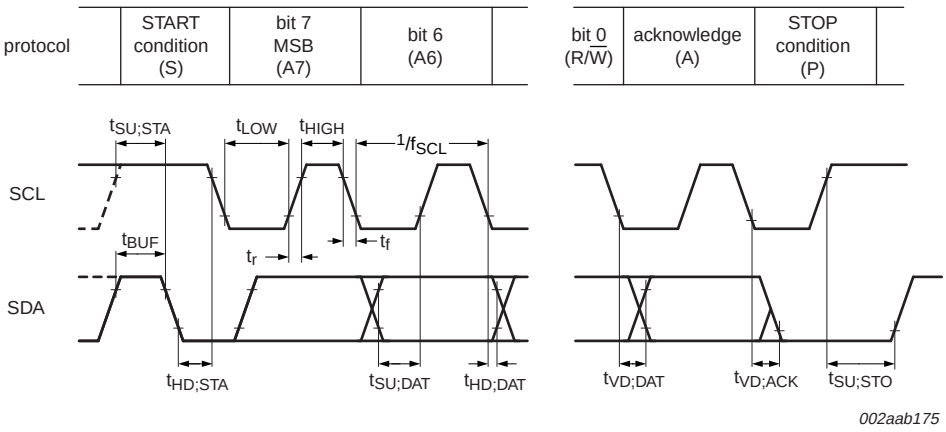


Fig 23. Definition of timing on the I²C-bus



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

Fig 24. I²C-bus timing diagram

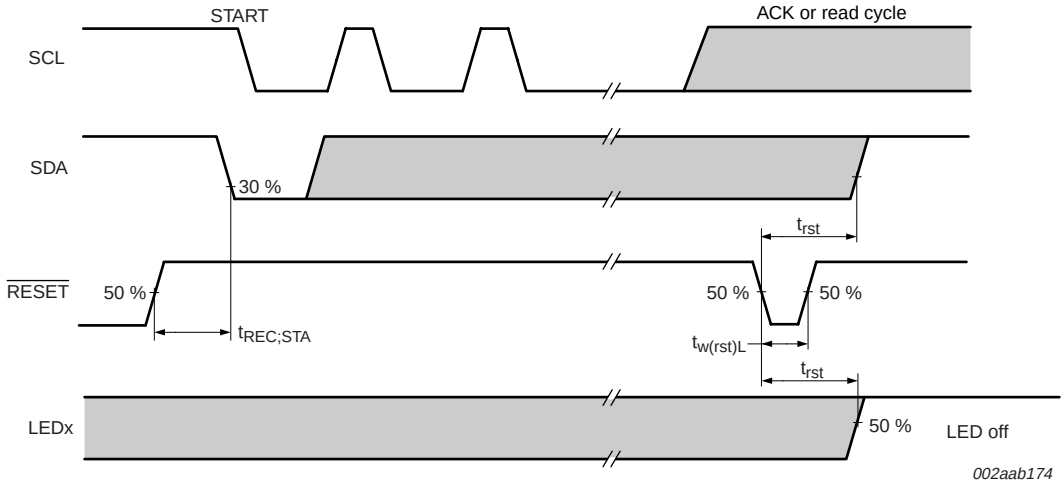
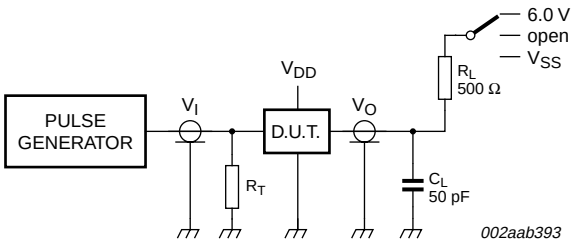


Fig 25. Definition of $\overline{\text{RESET}}$ timing

14. Test information



Definitions test circuit:

R_L = Load resistance.

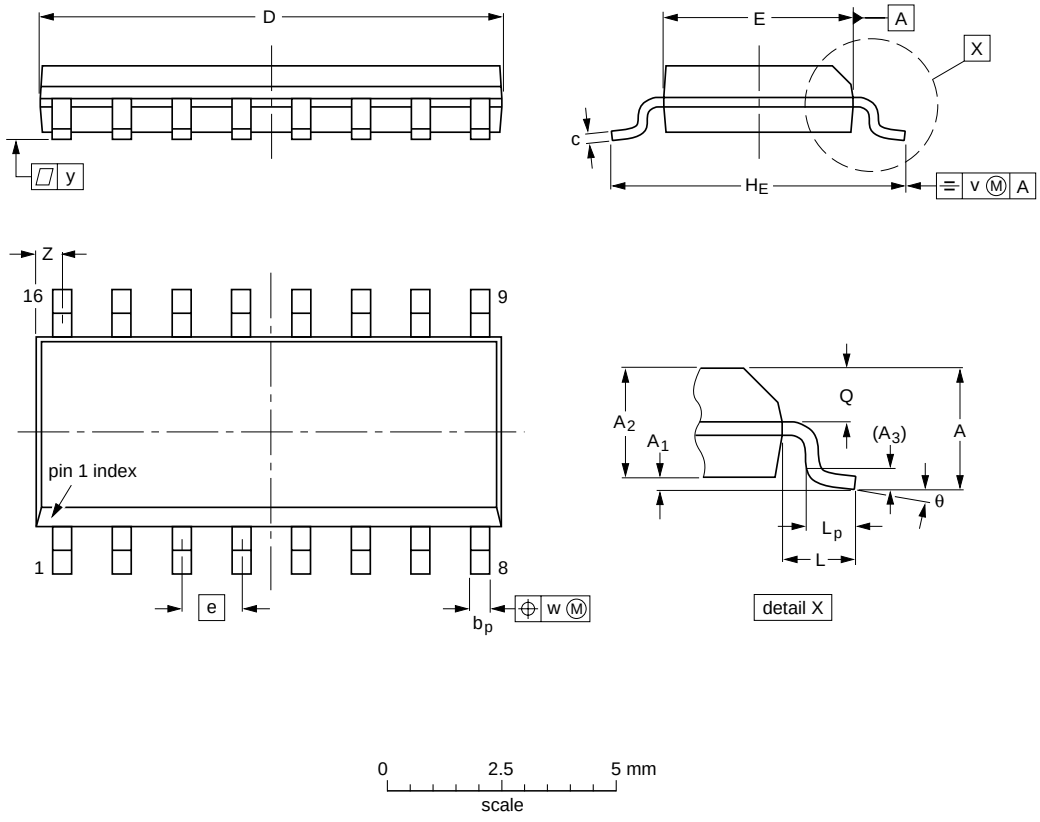
C_L = Load capacitance including jig and probe capacitance.

R_T = Termination resistance should be equal to the output impedance Z_o of the pulse generator.

Fig 26. Test circuitry for switching times

15. Package outline

SO16: plastic small outline package; 16 leads; body width 3.9 mm **SOT109-1**



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	1.75	0.25 0.10	1.45 1.25	0.25	0.49 0.36	0.25 0.19	10.0 9.8	4.0 3.8	1.27	6.2 5.8	1.05	1.0 0.4	0.7 0.6	0.25	0.25	0.1	0.7 0.3	8^o 0^o
inches	0.069	0.010 0.004	0.057 0.049	0.01	0.019 0.014	0.0100 0.0075	0.39 0.38	0.16 0.15	0.05	0.244 0.228	0.041	0.039 0.016	0.028 0.020	0.01	0.01	0.004	0.028 0.012	

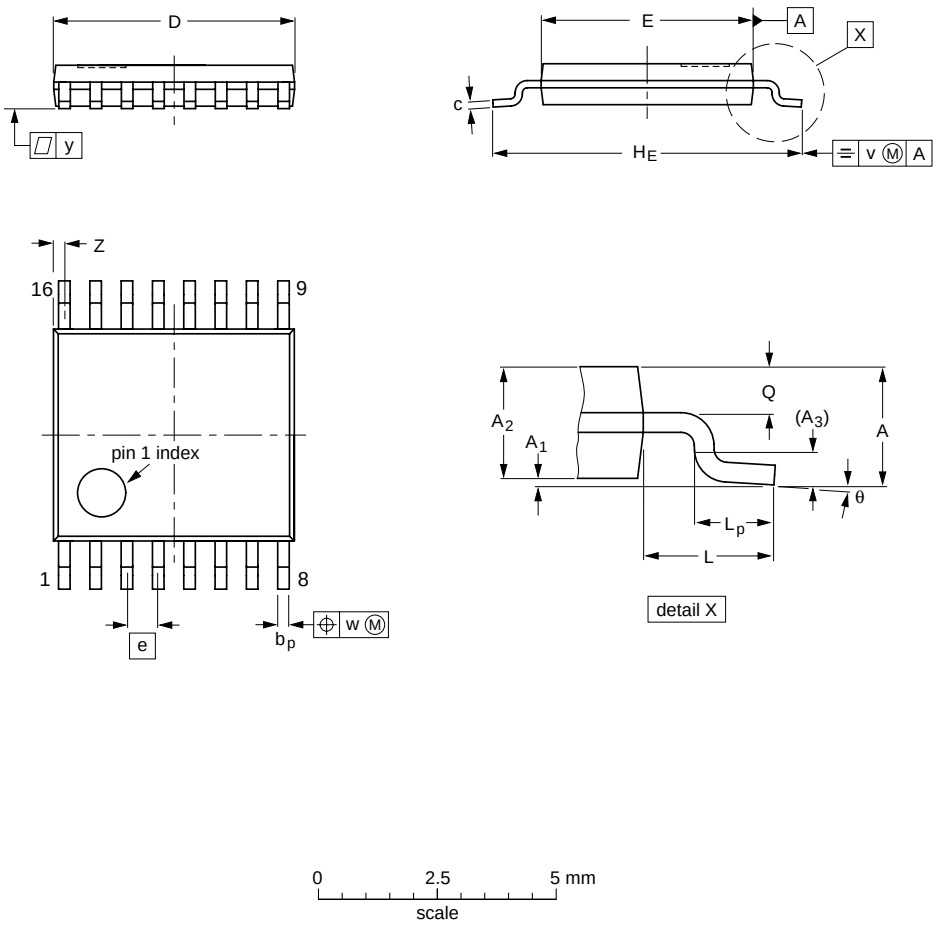
Note

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			
SOT109-1	076E07	MS-012				~~99-12-27~~ 03-02-19

Fig 27. Package outline SOT109-1 (SO16)

TSSOP16: plastic thin shrink small outline package; 16 leads; body width 4.4 mm **SOT403-1**



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	5.1 4.9	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	0.40 0.06	8^o 0^o

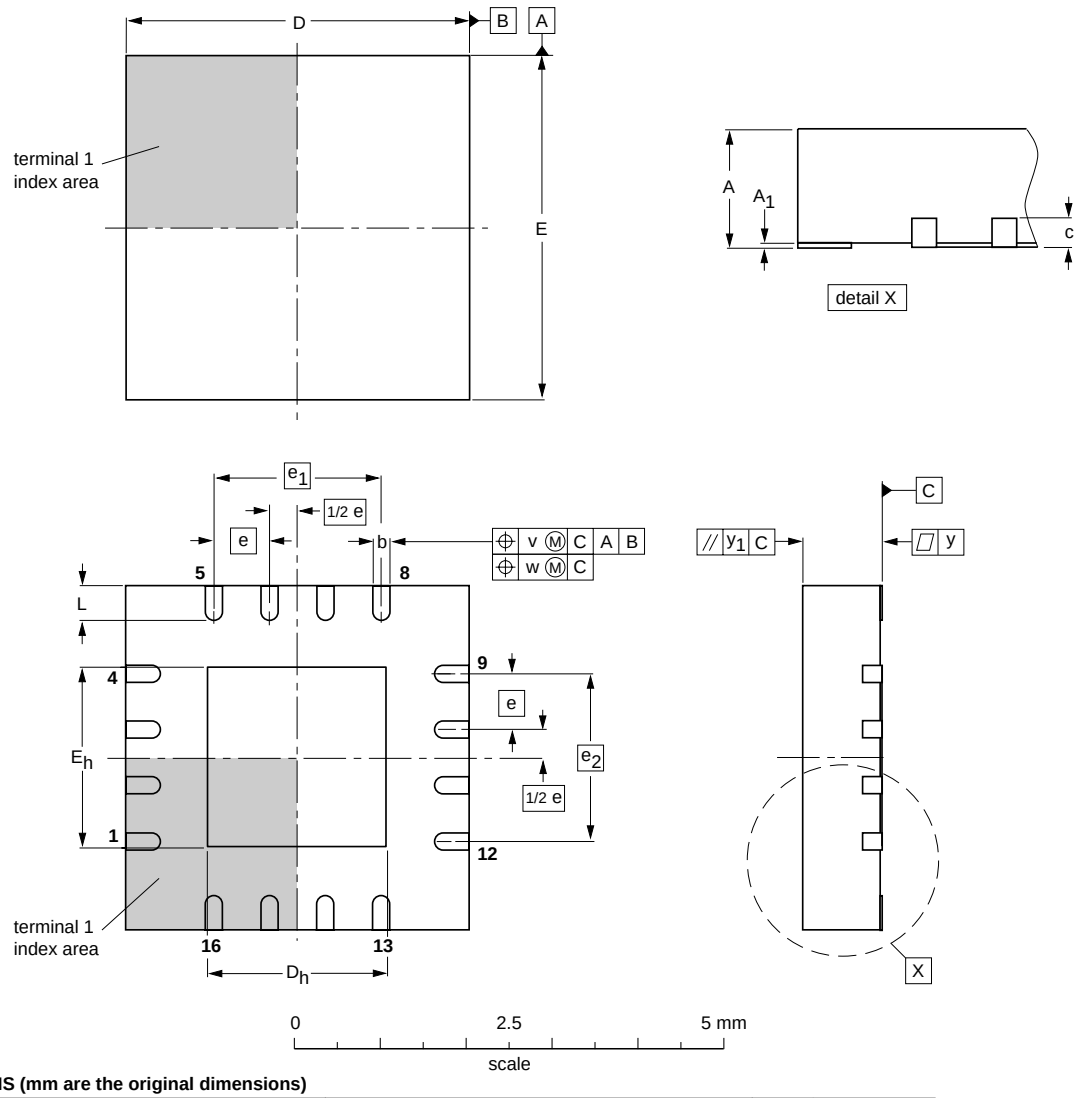
Notes

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			
SOT403-1		MO-153				~~99-12-27~~ 03-02-18

Fig 28. Package outline SOT403-1 (TSSOP16)

HVQFN16: plastic thermal enhanced very thin quad flat package; no leads; 16 terminals; body 4 x 4 x 0.85 mm **SOT629-1**



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	4.1 3.9	2.25 1.95	4.1 3.9	2.25 1.95	0.65	1.95	1.95	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			
SOT629-1	- - -	MO-220	- - -			~~01-08-08~~ 02-10-22

Fig 29. Package outline SOT629-1 (HVQFN16)

16. Soldering

16.1 Introduction to soldering surface mount packages

This text gives a very brief insight to a complex technology. A more in-depth account of soldering ICs can be found in our *Data Handbook IC26; Integrated Circuit Packages* (document order number 9398 652 90011).

There is no soldering method that is ideal for all surface mount IC packages. Wave soldering can still be used for certain surface mount ICs, but it is not suitable for fine pitch SMDs. In these situations reflow soldering is recommended.

16.2 Reflow soldering

Reflow soldering requires solder paste (a suspension of fine solder particles, flux and binding agent) to be applied to the printed-circuit board by screen printing, stencilling or pressure-syringe dispensing before package placement. Driven by legislation and environmental forces the worldwide use of lead-free solder pastes is increasing.

Several methods exist for reflowing; for example, convection or convection/infrared heating in a conveyor type oven. Throughput times (preheating, soldering and cooling) vary between 100 seconds and 200 seconds depending on heating method.

Typical reflow peak temperatures range from 215 °C to 270 °C depending on solder paste material. The top-surface temperature of the packages should preferably be kept:

- below 225 °C (SnPb process) or below 245 °C (Pb-free process)
  - for all BGA, HTSSON..T and SSOP..T packages
  - for packages with a thickness ≥ 2.5 mm
  - for packages with a thickness < 2.5 mm and a volume ≥ 350 mm^3 so called thick/large packages.
- below 240 °C (SnPb process) or below 260 °C (Pb-free process) for packages with a thickness < 2.5 mm and a volume < 350 mm^3 so called small/thin packages.

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

16.3 Wave soldering

Conventional single wave soldering is not recommended for surface mount devices (SMDs) or printed-circuit boards with a high component density, as solder bridging and non-wetting can present major problems.

To overcome these problems the double-wave soldering method was specifically developed.

If wave soldering is used the following conditions must be observed for optimal results:

- Use a double-wave soldering method comprising a turbulent wave with high upward pressure followed by a smooth laminar wave.
- For packages with leads on two sides and a pitch (e):
  - larger than or equal to 1.27 mm, the footprint longitudinal axis is **preferred** to be parallel to the transport direction of the printed-circuit board;

– smaller than 1.27 mm, the footprint longitudinal axis **must** be parallel to the transport direction of the printed-circuit board.

The footprint must incorporate solder thieves at the downstream end.

- For packages with leads on four sides, the footprint must be placed at a 45° angle to the transport direction of the printed-circuit board. The footprint must incorporate solder thieves downstream and at the side corners.

During placement and before soldering, the package must be fixed with a droplet of adhesive. The adhesive can be applied by screen printing, pin transfer or syringe dispensing. The package can be soldered after the adhesive is cured.

Typical dwell time of the leads in the wave ranges from 3 seconds to 4 seconds at 250 °C or 265 °C, depending on solder material applied, SnPb or Pb-free respectively.

A mildly-activated flux will eliminate the need for removal of corrosive residues in most applications.

16.4 Manual soldering

Fix the component by first soldering two diagonally-opposite end leads. Use a low voltage (24 V or less) soldering iron applied to the flat part of the lead. Contact time must be limited to 10 seconds at up to 300 °C.

When using a dedicated tool, all other leads can be soldered in one operation within 2 seconds to 5 seconds between 270 °C and 320 °C.

16.5 Package related soldering information

Table 19: Suitability of surface mount IC packages for wave and reflow soldering methods

Package [1]	Soldering method	
	Wave	Reflow [2]
BGA, HTSSON..T [3], LBGA, LFBGA, SQFP, SSOP..T [3], TFBGA, VFBGA, XSON	not suitable	suitable
DHVQFN, HBCC, HBGA, HLQFP, HSO, HSOP, HSQFP, HSSON, HTQFP, HTSSOP, HVQFN, HVSON, SMS	not suitable [4]	suitable
PLCC [5], SO, SOJ	suitable	suitable
LQFP, QFP, TQFP	not recommended [5] [6]	suitable
SSOP, TSSOP, VSO, VSSOP	not recommended [7]	suitable
CWQCCN..L [8], PMFP [9], WQCCN..L [8]	not suitable	not suitable

[1] For more detailed information on the BGA packages refer to the *(LF)BGA Application Note* (AN01026); order a copy from your Philips Semiconductors sales office.

[2] All surface mount (SMD) packages are moisture sensitive. Depending upon the moisture content, the maximum temperature (with respect to time) and body size of the package, there is a risk that internal or external package cracks may occur due to vaporization of the moisture in them (the so called popcorn effect). For details, refer to the Drypack information in the *Data Handbook IC26; Integrated Circuit Packages; Section: Packing Methods*.

[3] These transparent plastic packages are extremely sensitive to reflow soldering conditions and must on no account be processed through more than one soldering cycle or subjected to infrared reflow soldering with peak temperature exceeding 217 °C ± 10 °C measured in the atmosphere of the reflow oven. The package body peak temperature must be kept as low as possible.

[4] These packages are not suitable for wave soldering. On versions with the heatsink on the bottom side, the solder cannot penetrate between the printed-circuit board and the heatsink. On versions with the heatsink on the top side, the solder might be deposited on the heatsink surface.

[5] If wave soldering is considered, then the package must be placed at a 45° angle to the solder wave direction. The package footprint must incorporate solder thieves downstream and at the side corners.

[6] Wave soldering is suitable for LQFP, QFP and TQFP packages with a pitch (e) larger than 0.8 mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.65 mm.

[7] Wave soldering is suitable for SSOP, TSSOP, VSO and VSSOP packages with a pitch (e) equal to or larger than 0.65 mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.5 mm.

[8] Image sensor packages in principle should not be soldered. They are mounted in sockets or delivered pre-mounted on flex foil. However, the image sensor package can be mounted by the client on a flex foil by using a hot bar soldering process. The appropriate soldering profile can be provided on request.

[9] Hot bar soldering or manual soldering is suitable for PMFP packages.

17. Abbreviations

Table 20: Abbreviations

Acronym	Description
AI	Auto Increment
CDM	Charged Device Model
EEPROM	Electrically Erasable Programmable Read-Only Memory
ESD	ElectroStatic Discharge
FRU	Field Replaceable Unit
HBM	Human Body Model
I²C-bus	Inter Integrated Circuit bus
MM	Machine Model
POR	Power-On Reset
SMBus	System Management Bus

18. Revision history

Table 21: Revision history

Document ID	Release date	Data sheet status	Change notice	Doc. number	Supersedes
PCA9541_4	20060104	Product data sheet	-	-	PCA9541_3
Modifications:	• In data sheet title “I²C” changed to “I²C-bus” • Section 8.1 “Device address”: added Caution statement • “V_{CC}” changed to “V_{DD}” in Table 16 and Table 17 • Table 18 “Dynamic characteristics”: symbol “f_{SCLIR}” changed to “$f_{SCL(init/rec)}$” and parameter description modified • added Table 20 “Abbreviations”				
PCA9541_3	20050713	Product data sheet	-	9397 750 14746	PCA9541_2
PCA9541_2	20041001	Product data sheet	-	9397 750 13629	PCA9541_1
PCA9541_1	20031202	Product data sheet	853-2436 01-A14594	9397 750 12453	-

19. Data sheet status

Level	Data sheet status [1]	Product status [2] [3]	Definition
I	Objective data	Development	This data sheet contains data from the objective specification for product development. Philips Semiconductors reserves the right to change the specification in any manner without notice.
II	Preliminary data	Qualification	This data sheet contains data from the preliminary specification. Supplementary data will be published at a later date. Philips Semiconductors reserves the right to change the specification without notice, in order to improve the design and supply the best possible product.
III	Product data	Production	This data sheet contains data from the product specification. Philips Semiconductors reserves the right to make changes at any time in order to improve the design, manufacturing and supply. Relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN).

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

[2] The product status of the device(s) described in this data sheet may have changed since this data sheet was published. The latest information is available on the Internet at URL http://www.semiconductors.philips.com.

[3] For data sheets describing multiple type numbers, the highest-level product status determines the data sheet status.

20. Definitions

Short-form specification — The data in a short-form specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.

Limiting values definition — Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 60134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.

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

21. Disclaimers

Life support — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status 'Production'), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

22. 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 Koninklijke Philips Electronics N.V.

23. Contact information

For additional information, please visit: **http://www.semiconductors.philips.com**

For sales office addresses, send an email to: **sales.addresses@www.semiconductors.philips.com**

24. Contents

Date of release: 4 January 2006
Document number: PCA9541_4

Published in The Netherlands