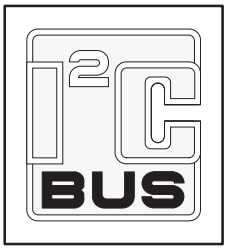
DATA SHEET



PCA9564

Parallel bus to I^{2}C-bus controller

Product data sheet
Supersedes data of 2004 Jun 25

2006 Sep 01

Philips Semiconductors

PHILIPS

Parallel bus to I²C-bus controller PCA9564

FEATURES

- Parallel-bus to I²C-bus protocol converter and interface
- Both master and slave functions
- Multi-master capability
- Internal oscillator reduces external components
- Operating supply voltage 2.3 V to 3.6 V
- 5 V tolerant I/Os
- Standard and fast mode I²C capable and compatible with SMBus
- ESD protection exceeds 2000 V HEM 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 exceed 100 mA.
- Packages offered: DIP20, SO20, TSSOP20, HVQFN20

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
- Higher frequency, lower voltage migration path for the PCF8584
- Converts 8 bits of parallel data to serial data stream to prevent having to run a large number of traces across the entire PC board

DESCRIPTION

The PCA9564 is an integrated circuit designed in CMOS technology that serves as an interface between most standard parallel-bus microcontrollers/microprocessors and the serial I²C-bus and allows the parallel bus system to communicate bi-directionally with the I²C-bus. The PCA9564 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-wise basis using interrupt or polled handshake. The PCA9564 controls all the I²C-bus specific sequences, protocol, arbitration and timing with no external timing element required.

The PCA9564 is similar to the PCF8584 but operates at lower voltages and higher I²C frequencies. Other enhancements requested by design engineers have also been incorporated.

Characteristic	PCA9564	PCF8584	Comments
Voltage range	2.3–3.6 V	4.5–5.5 V	PCA9564 is 5 V tolerant
Maximum master mode I²C frequency	360 kHz	90 kHz	Faster I²C interface
Maximum slave mode I²C frequency	400 kHz	100 kHz	Faster I²C interface
Clock source	Internal	External	Less expensive and more flexible with internal oscillator
Parallel interface	Fast 50 MHz	Slow	Compatible with faster processors

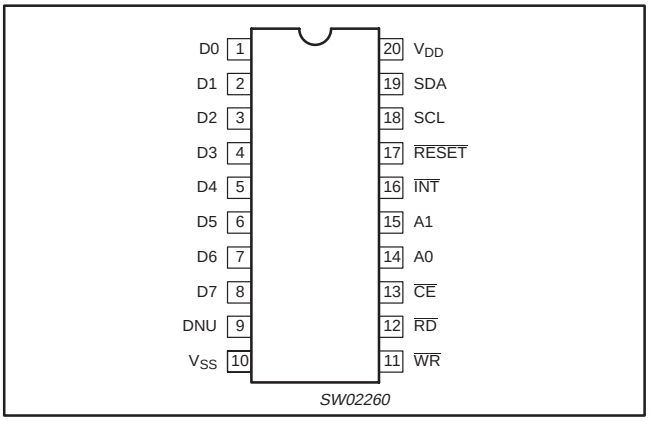
While the PCF8584 supported most parallel-bus microcontrollers/microprocessors including the Intel 8049/8051, Motorola 6800/68000 and the Zilog Z80, the PCA9564 has been designed to be very similar to the Philips standard 80C51 microcontroller I²C hardware so the devices are not code compatible. Additionally, the PCA9564 does not support the bus monitor “Snoop” mode nor the long distance mode and is not footprint compatible with the PCF8584.

ORDERING INFORMATION

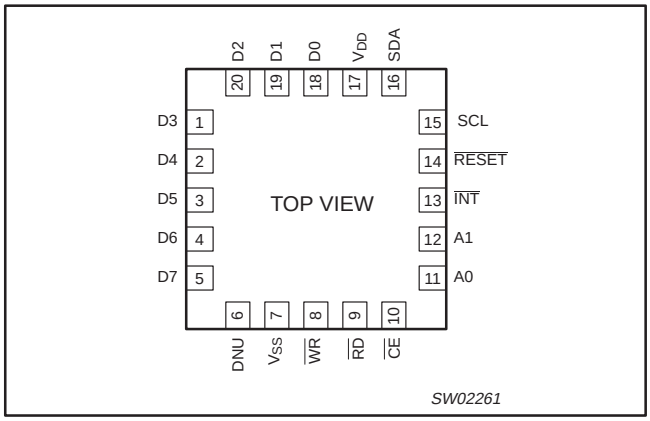
PACKAGES	TEMPERATURE RANGE	ORDER CODE	TOPSIDE MARK	DRAWING NUMBER
20-Pin Plastic DIP	–40 °C to +85 °C	PCA9564N	PCA9564N	SOT146-1
20-Pin Plastic SO	–40 °C to +85 °C	PCA9564D	PCA9564D	SOT163-1
20-Pin Plastic TSSOP	–40 °C to +85 °C	PCA9564PW	PCA9564	SOT360-1
20-Pin Plastic HVQFN	–40 °C to +85 °C	PCA9564BS	9564	SOT662-1
whole wafer	–40 °C to +85 °C	PCA9564U	n/a	n/a

Standard packing quantities and other packaging data are available at www.standardics.philips.com/packaging.

PIN CONFIGURATION — DIP, SO, TSSOP



PIN CONFIGURATION — HVQFN



PIN DESCRIPTION

PIN NUMBER		SYMBOL	PIN TYPE	NAME AND FUNCTION
DIP, SO, TSSOP	HVQFN			
1, 2, 3, 4, 5, 6, 7, 8	1, 2, 3, 4, 5, 18, 19, 20	D0–D7	I/O	**Data Bus:** Bi-directional 3-State data bus used to transfer commands, data and status between the controller and the CPU. D0 is the least significant bit.
9	6	DNU		Do not use: must be left floating (pulled LOW internally)
10	7[1]	V_{SS}	Pwr	Ground
11	8	$\overline{WR}$	I	**Write Strobe:** When LOW and $\overline{CE}$ is also LOW, the contents of the data bus is loaded into the addressed register. The transfer occurs on the rising edge of the signal.
12	9	$\overline{RD}$	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 $\overline{RD}$.
13	10	$\overline{CE}$	I	**Chip Enable:** Active-LOW input signal. When LOW, data transfers between the CPU and the controller are enabled on D0–D7 as controlled by the $\overline{WR}$, $\overline{RD}$ and A0–A1 inputs. When HIGH, places the D0–D7 lines in the 3-State condition.
14, 15	11, 12	A0, A1	I	**Address Inputs:** Selects the controller internal registers and ports for read/write operations.
16	13	$\overline{INT}$	O	**Interrupt Request:** Active-LOW, open-drain, output. This pin requires a pull-up device.
17	14	$\overline{RESET}$	I	**Reset:** A LOW level clears internal registers resets the I²C state machine.
18	15	SCL	I/O	I²C-bus serial clock input/output (open-drain).
19	16	SDA	I/O	I²C-bus serial data input/output (open-drain).
20	17	V_{DD}	Pwr	**Power Supply:** 2.3 to 3.6 V

NOTES:

1. HVQFN package die supply ground is connected to both V_{SS} pin and exposed center pad. 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.

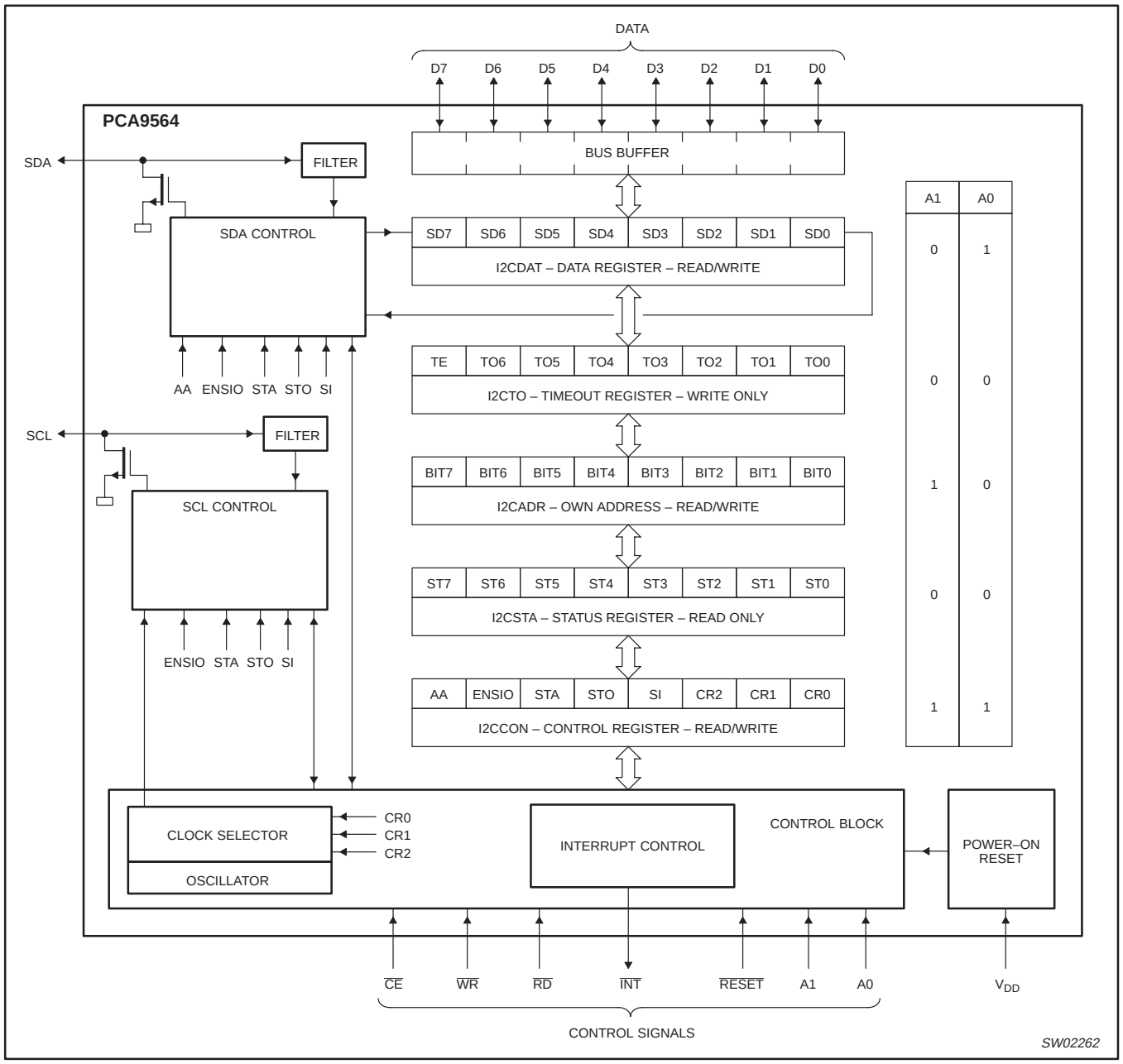


Figure 1. Block diagram

FUNCTIONAL DESCRIPTION

General

The PCA9564 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 master or slave. Bidirectional data transfer between the I2C-bus and the parallel-bus microcontroller is carried out on a byte-wise basis, using either an interrupt or polled handshake.

Internal Oscillator

The PCA9564 contains an internal 9 MHz oscillator which is used for all I2C timing. The oscillator requires up to 500 µs to start-up after ENSIO bit is set to “1”.

Registers

The PCA9564 contains four registers which are used to configure the operation of the device as well as to send and receive serial data.

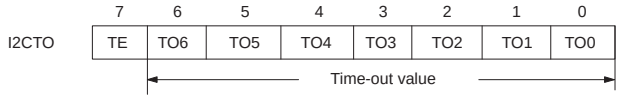
The registers are selected by setting pins A0 and A1 to the appropriate logic levels before a read or write operation is executed.

CAUTION: Do not write to I2C registers while the I2C-bus is busy and the SIO is in master or addressed slave mode.

REGISTER NAME	REGISTER FUNCTION	A1	A0	READ/ WRITE	DEFAULT
I2CSTA	Status	0	0	R	F8h
I2CTO	Time-out	0	0	W	FFh
I2CDAT	Data	0	1	R/W	00h
I2CADR	Own address	1	0	R/W	00h
I2CCON	Control	1	1	R/W	00h

The Time-out Register, I2CTO: The time-out register is used to determine the maximum time that SCL is allowed to be LOW before the I2C state machine is reset.

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

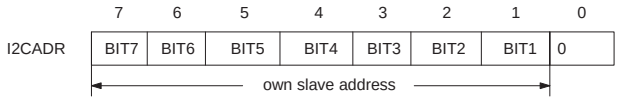


The most significant bit of I2CTO (TE) is used as a time-out enable/disable. A “1” will enable the time-out function. The time-out period = (I2CTO[6:0] + 1) × 113.7 µs. The time-out value may vary some and is an approximate value.

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

1. When the SIO, in the master mode, wants to send a START condition and the SCL line is held LOW by some other device. The SIO waits a time period equivalent to the time-out value for the SCL to be released. In case it is not released, the SIO concludes that there is a bus error, loads 90H 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 an external reset in order to reset the SIO.
2. 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 SIO concludes there is a bus error and behaves in the manner described above.
3. In case of a forced access to the I2C-bus. (See more details on page 15.)

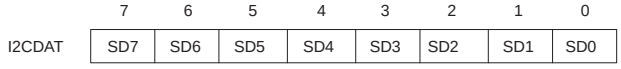
The Address Register, I2CADR: I2CADR is not affected by the SIO hardware. The contents of this register are irrelevant when SIO is in a master mode. In the slave modes, the seven most significant bits must be loaded with the microcontroller’s own slave address.



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. The least significant bit is not used but should be programmed with a ‘0’.

The Data Register, I2CDAT: I2CDAT 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 it is not in the process of shifting a byte. This occurs when SIO 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 SIO generates an interrupt, the I2CDAT registers contain the data byte that was just transferred on the I2C-bus.

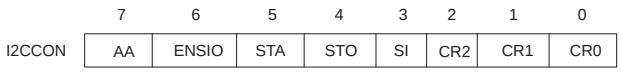
NOTE: The I2CDAT register will capture the serial address as data when addressed via the serial bus. Also, the data register will continue to capture data from the serial bus during 38H so the I2CDAT register will need to be reloaded when the bus becomes free.



- SD7 - SD0:

Eight bits to be transmitted or just received. A logic 1 in I2CDAT corresponds to a HIGH level on the I2C-bus, and a logic 0 corresponds to a LOW level on the bus.

The Control Register, I2CCON: The microcontroller can read from and write to this 8-bit register. Two bits are affected by the SIO 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 clears the SI bit and causes the Serial Interrupt line to be de–asserted and the next clock pulse on the SCL line to be generated. 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.



- ENSIO, THE SIO 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, SIO is in the “not addressed” slave state.

ENSIO = “1”: When ENSIO is “1”, SIO is enabled.

After the ENSIO bit is set, it takes 500 µs for the internal oscillator to start up, therefore, the PCA9564 will enter either the master or the slave mode after this time. ENSIO should not be used to temporarily

release the PCA9564 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 in the following text).

In the following text, it is assumed that ENSIO = “1”.

- STA, THE START FLAG

STA = “1”: When the STA bit is set to enter a master mode, the SIO 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 SIO 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 SIO is already in a master mode and one or more bytes are transmitted or received, SIO transmits a repeated START condition. STA may be set at any time. STA may also be set when SIO is an addressed slave.

STA = “0”: When the STA bit is reset, no START condition or repeated START condition will be generated.

- STO, THE STOP FLAG

STO = “1”: When the STO bit is set while SIO is in a master mode, a STOP condition is transmitted to the I2C-bus. When the STOP condition is detected on the bus, the SIO hardware clears the STO flag.

If the STA and STO bits are both set, then a STOP condition is transmitted to the I2C-bus if SIO is in a master mode. SIO then transmits a START condition.

STO = “0”: When the STO bit is reset, no STOP condition will be generated.

- SI, THE SERIAL INTERRUPT FLAG

SI = “1”: When the SI flag is set, then, if the ENSIO bit is also set, a serial interrupt is requested. SI is set by hardware when one of 24 of the 25 possible SIO 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 must be reset by writing “0” to the SI bit. 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.

- AA, THE ASSERT ACKNOWLEDGE FLAG

AA = “1”: If the AA flag is set, an acknowledge (LOW level to SDA) will be returned during the acknowledge clock pulse on the SCL line when:

– The “own slave address” has been received
– A data byte has been received while SIO is in the master receiver mode
– A data byte has been received while SIO is in the addressed slave receiver mode

AA = “0”: if the AA flag is reset, a not acknowledge (HIGH level to SDA) will be returned during the acknowledge clock pulse on SCL when:

– A data byte has been received while SIO is in the master receiver mode
– A data byte has been received while SIO is in the addressed slave receiver mode
– “Own slave address” has been received

When SIO is in the addressed slave transmitter mode, state C8H will be entered after the last serial is transmitted (see Figure 5). When SI is cleared, 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 SIO 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, SIO can be temporarily released from the I2C-bus while the bus status is monitored. While SIO 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.

- THE CLOCK RATE BITS, CR2, CR1, AND CR0

Three bits determine the serial clock frequency when SIO is in master mode. The various serial rates are shown in Table 1.

The clock frequencies only take the HIGH and LOW times into consideration. The rise and fall time will cause the actual measured frequency to be lower than expected.

The frequencies shown in Table 1 are unimportant when SIO is in a slave mode. In the slave modes, SIO will automatically synchronize with any clock frequency up to 400 kHz.

Table 1. Serial Clock Rates

CR2	CR1	CR0	SERIAL CLOCK FREQUENCY (kHz)
0	0	0	330
0	0	1	288
0	1	0	217
0	1	1	146
1	0	0	88[1]
1	0	1	59
1	1	0	44
1	1	1	36

NOTE:

1. The clock frequency values are approximate and may vary with temperature, supply voltage, process, and SCL output loading. If normal mode I2C parameters must be strictly followed (SCL < 100kHz), it is recommended not to use CR[2:0] = 100 (SCL = 88kHz) since the clock frequency might be slightly higher than 100 kHz under certain temperature, voltage, and process conditions and use CR[2:0] = 101 (SCL = 59 kHz) instead.

The Status Register, I2CSTA: I2CSTA is an 8-bit read-only register. The three least significant bits are always zero. The five most significant bits contain the status code. There are 25 possible status codes. When I2CSTA contains F8H, no relevant state information is available and no serial interrupt is requested. All other I2CSTA values correspond to defined SIO states. When each of these states is entered, a serial interrupt is requested (SI = “1”).

More Information on SIO Operating Modes

The four operating modes are:
– Master Transmitter
– Master Receiver
– Slave Receiver
– Slave Transmitter

Data transfers in each mode of operation are shown in Figures 2–5. These figures contain the following abbreviations:

Abbreviation	Explanation
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)
$\overline{A}$	Not acknowledge bit (HIGH level at SDA)
Data	8-bit data byte
P	Stop condition

In Figures 2-5, 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. 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 Tables 2-6.

Master Transmitter Mode: In the master transmitter mode, a number of data bytes are transmitted to a slave receiver (see Figure 2). Before the master transmitter mode can be entered, I2CCON must be initialized as follows:

	7	6	5	4	3	2	1	0
I2CCON	AA	ENSIO	STA	STO	SI	CR2	CR1	CR0
	X	1	0	0	0	← bit rate →		

ENSIO must be set to logic 1 to enable SIO. If the AA bit is reset, SIO 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, SIO cannot enter a slave mode. STA, STO, and SI must be reset.

The master transmitter mode may now be entered by setting the STA bit. The SIO logic will now 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, 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). The SI bit in I2CCON must then be reset before the serial transfer can continue.

When the slave address and the direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in I2CSTA are possible. There are 18H, 20H, or 38H for the master mode and also 68H, or B0H if the slave mode was enabled (AA = logic 1). The appropriate action to be taken for each of these status codes is detailed in Table 2. After a repeated start condition (state 10H). SIO may switch to the master receiver mode by loading I2CDAT with SLA+R).

Note that a master should never transmit its own slave address.

Master Receiver Mode: In the master receiver mode, a number of data bytes are received from a slave transmitter (see Figure 3). The transfer is initialized as in the master transmitter mode. When the start condition has been transmitted, the interrupt service routine must load I2CDAT with the 7-bit slave address and the data direction bit (SLA+R). The SI bit in I2CCON must then be cleared before the serial transfer can continue.

When the slave address and the data direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in I2CSTA are possible. These are 40H, 48H, or 38H for the master mode and also 68H, or B0H if the slave mode was enabled (AA = logic 1). The appropriate action to be taken for each of these status codes is detailed in Table 3. ENSIO is not affected by the serial transfer and are not referred to in Table 3. After a repeated start condition (state 10H), SIO may switch to the master transmitter mode by loading I2CDAT with SLA+W.

Note that a master should not transmit its own slave address.

Slave Receiver Mode: In the slave receiver mode, a number of data bytes are received from a master transmitter (see Figure 4). To initiate the slave receiver mode, I2CADR and I2CCON must be loaded as follows:

	7	6	5	4	3	2	1	0
I2CADR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	0
	← own slave address →							

The upper 7 bits are the address to which SIO will respond when addressed by a master.

	7	6	5	4	3	2	1	0
I2CCON	AA	ENSIO	STA	STO	SI	CR2	CR1	CR0
	1	1	0	0	0	X	X	X

ENSIO must be set to logic 1 to enable SIO. The AA bit must be set to enable SIO to acknowledge its own slave address, STA, STO, and SI must be reset.

When I2CADR and I2CCON have been initialized, SIO waits until it is addressed by its own slave address followed by the data direction bit which must be “0” (W) for SIO to operate in the slave receiver mode. After its own slave address and the W bit have been received, the serial interrupt flag (I) is set and a valid status code can be read from I2CSTA. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status codes is detailed in Table 4. The slave receiver mode may also be entered if arbitration is lost while SIO is in the master mode (see status 68H).

If the AA bit is reset during a transfer, SIO will return a not acknowledge (logic 1) to SDA after the next received data byte. While AA is reset, SIO 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.

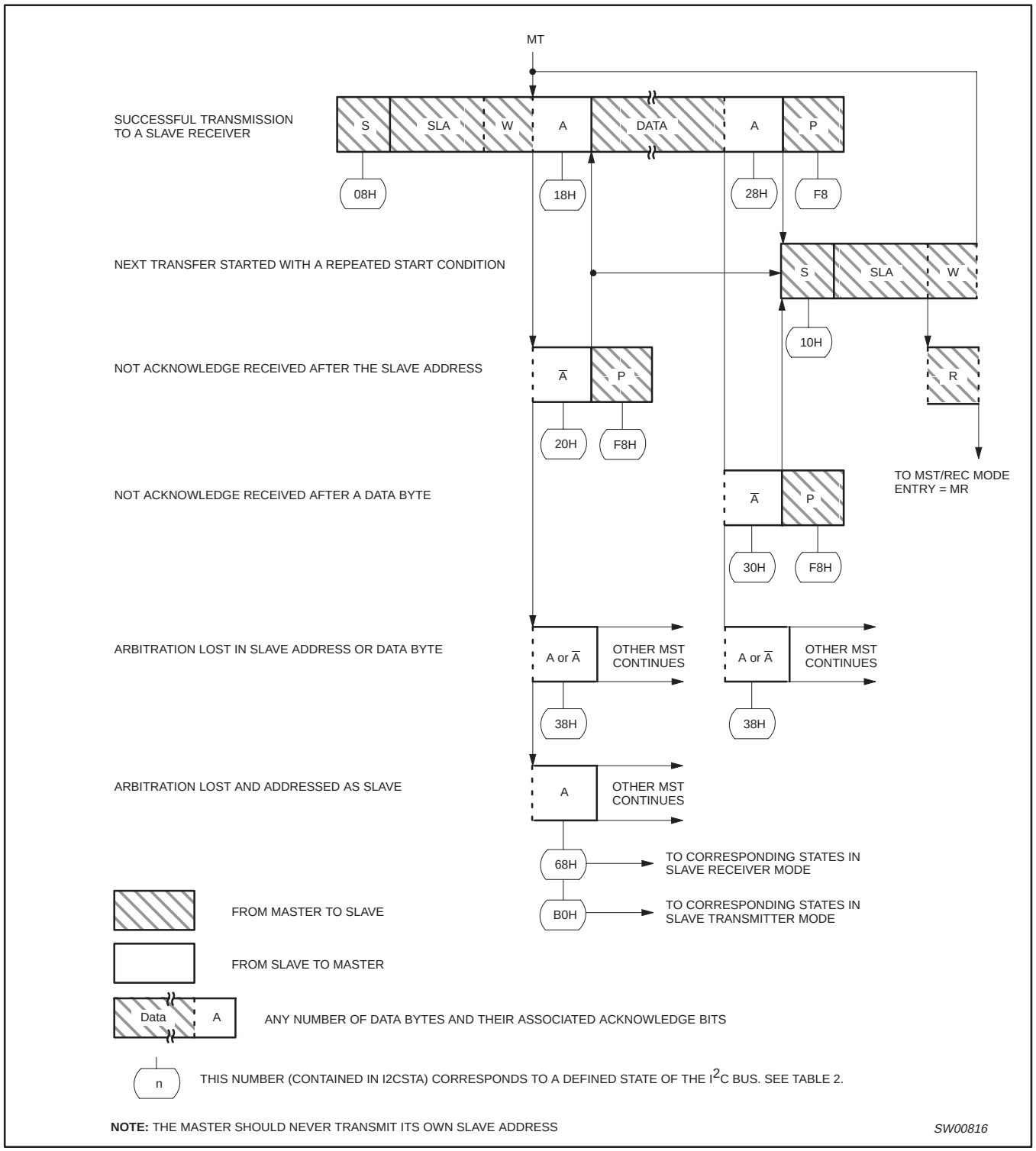


Figure 2. Format and states in the master transmitter mode

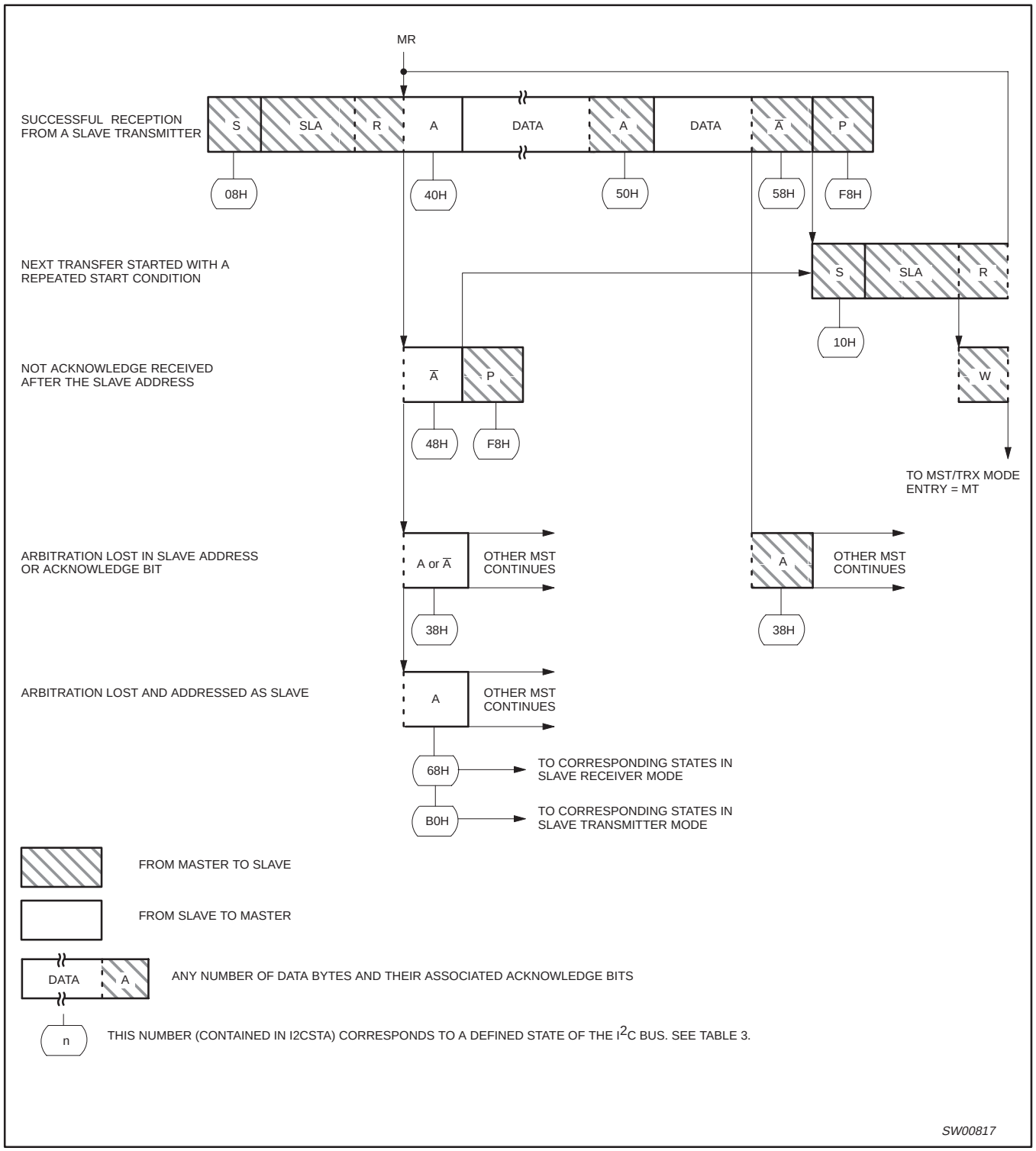


Figure 3. Format and states in the master receiver mode

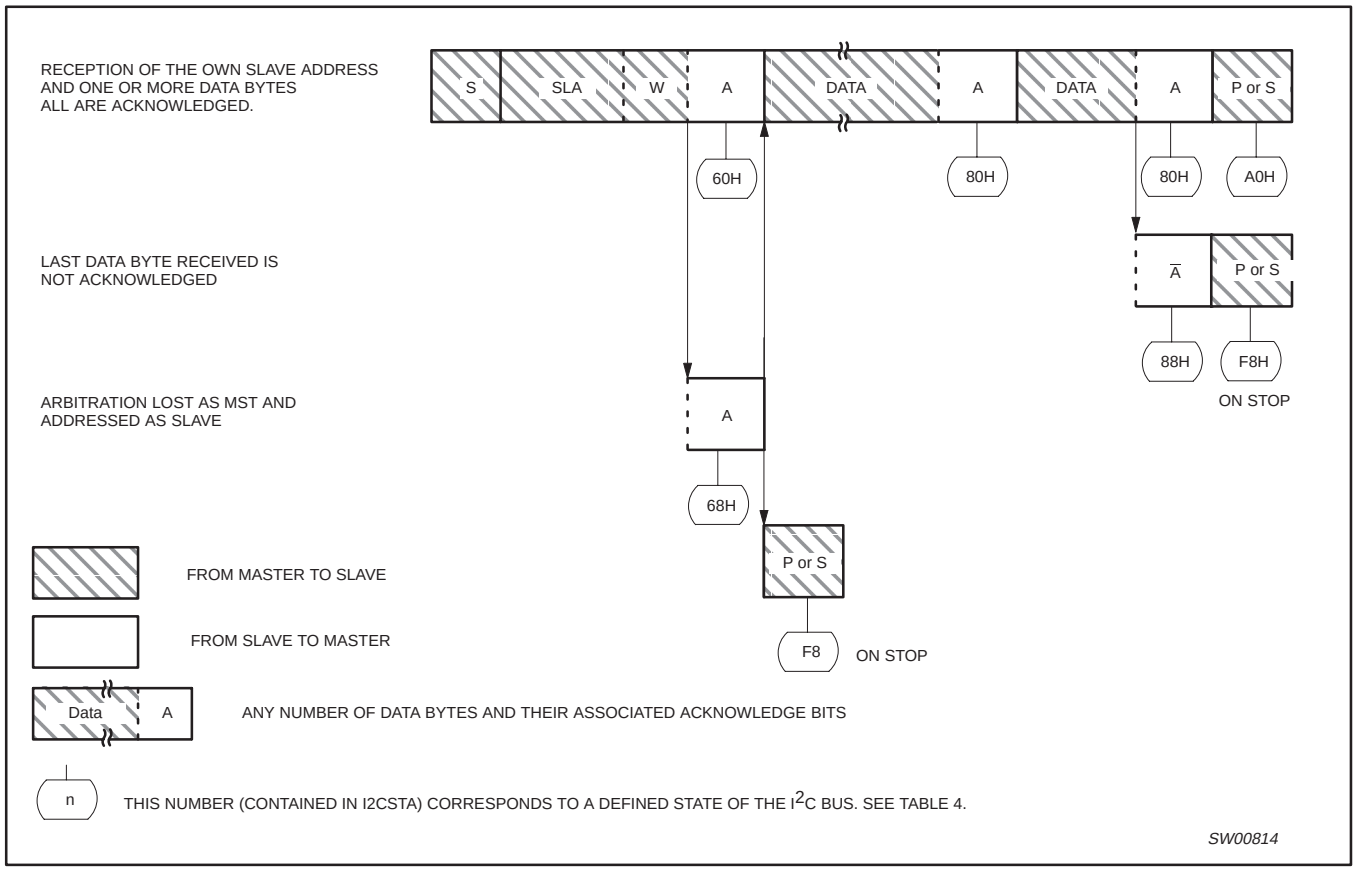


Figure 4. Format and states in the slave receiver mode

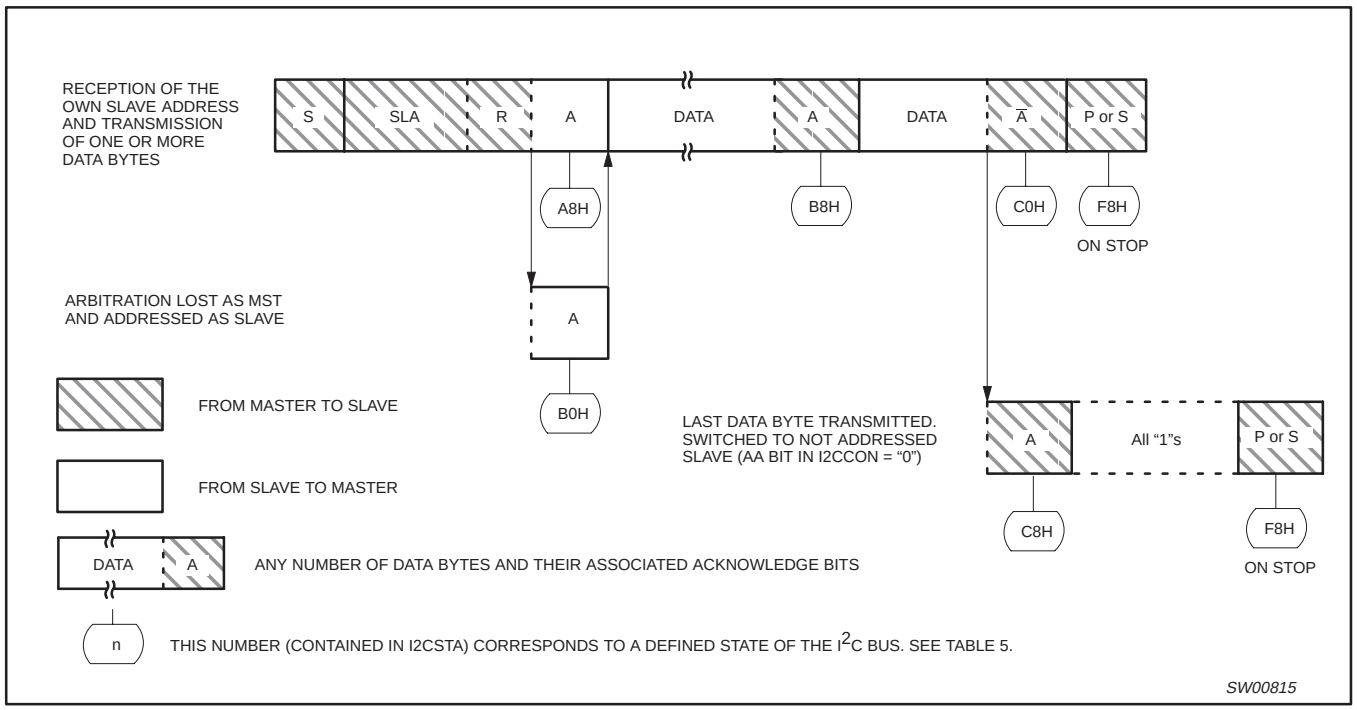


Figure 5. Format and states of the slave transmitter mode

Table 2. Master Transmitter Mode

STATUS CODE (I2CSTA)	STATUS OF THE I²C BUS AND SIO HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO HARDWARE
		TO/FROM I2CDAT	TO I2CCON				
			STA	STO	SI	AA	
08H	A START condition has been transmitted	Load SLA+W	X	X	0	X	SLA+W will be transmitted; ACK bit will be received
10H	A repeated START condition has been transmitted	Load SLA+W or	X	X	0	X	As above
		Load SLA+R	X	X	0	X	SLA+R will be transmitted; SIO will be switched to MST/REC mode
18H	SLA+W has been transmitted; ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		no I2CDAT action or	1	0	0	X	Repeated START will be transmitted;
		no I2CDAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no I2CDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
20H	SLA+W has been transmitted; NOT ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		no I2CDAT action or	1	0	0	X	Repeated START will be transmitted;
		no I2CDAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no I2CDAT action	1	1	0	X	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	Data byte will be transmitted; ACK bit will be received
		no I2CDAT action or	1	0	0	X	Repeated START will be transmitted;
		no I2CDAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no I2CDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
30H	Data byte in I2CDAT has been transmitted; NOT ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		no I2CDAT action or	1	0	0	X	Repeated START will be transmitted;
		no I2CDAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no I2CDAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
38H	Arbitration lost in SLA+$\overline{W}$ or Data bytes	No I2CDAT action or	0	0	0	X	I²C-bus will be released; not addressed slave will be entered
		No I2CDAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free (STOP or SCL and SDA high)

Table 3. Master Receiver Mode

STATUS CODE (I2CSTA)	STATUS OF THE I2C BUS AND SIO HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO HARDWARE
		TO/FROM I2CDAT	TO I2CCON				
			STA	STO	SI	AA	
08H	A START condition has been transmitted	Load SLA+R	X	X	0	X	SLA+R will be transmitted; ACK bit will be received
10H	A repeated START condition has been transmitted	Load SLA+R or	X	X	0	X	As above
		Load SLA+W	X	X	0	X	SLA+W will be transmitted; SIO will be switched to MST/TRX mode
38H	Arbitration lost in NOT ACK bit	No I2CDAT action or	0	0	0	X	I2C-bus will be released; SIO will enter a slave mode
		No I2CDAT action	1	0	0	X	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	Data byte will be received; NOT ACK bit will be returned
		no I2CDAT action	0	0	0	1	Data byte will be received; ACK bit will be returned
48H	SLA+R has been transmitted; NOT ACK has been received	No I2CDAT action or	1	0	0	X	Repeated START condition will be transmitted
		no I2CDAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no I2CDAT action	1	1	0	X	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	Data byte will be received; NOT ACK bit will be returned
		read data byte	0	0	0	1	Data byte will be received; ACK bit will be returned
58H	Data byte has been received; NOT ACK has been returned	Read data byte or	1	0	0	X	Repeated START condition will be transmitted
		read data byte or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		read data byte	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
38H	Arbitration lost in SLA+R	No I2CDAT action or	0	0	0	X	I2C-bus will be released; not addressed slave will be entered
		No I2CDAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free

Table 4. Slave Receiver Mode

STATUS CODE (I2CSTA)	STATUS OF THE I²C BUS AND SIO HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO HARDWARE
		TO/FROM I2CDAT	TO I2CCON				
			STA	STO	SI	AA	
60H	Own SLA+W has been received; ACK has been returned	No I2CDAT action or	X	X	0	0	Data byte will be received and NOT ACK will be returned
		no I2CDAT action	X	X	0	1	Data byte will be received and ACK will be returned
68H	Arbitration lost in SLA+R/$\overline{W}$ as master; Own SLA+W has been received, ACK returned	No I2CDAT action or	X	X	0	0	Data byte will be received and NOT ACK will be returned
		no I2CDAT action	X	X	0	1	Data byte will be received and ACK will be returned
80H	Previously addressed with own SLV address; DATA has been received; ACK has been returned	Read data byte or	X	X	0	0	Data byte will be received and NOT ACK will be returned
		read data byte	X	X	0	1	Data byte will be received and ACK will be returned
88H	Previously addressed with own SLA; DATA byte has been received; NOT ACK has been returned	Read data byte or	0	X	0	0	Switched to not addressed SLV mode; no recognition of own SLA
		read data byte or	0	X	0	1	Switched to not addressed SLV mode; Own SLA will be recognized
		read data byte or	1	X	0	0	Switched to not addressed SLV mode; no recognition of own SLA. A START condition will be transmitted when the bus becomes free
		read data byte	1	X	0	1	Switched to not addressed SLV mode; Own SLA will be recognized. 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 SLV/REC	No I2CDAT action or	0	X	0	0	Switched to not addressed SLV mode; no recognition of own SLA
		No I2CDAT action or	0	X	0	1	Switched to not addressed SLV mode; Own SLA will be recognized
		No I2CDAT action or	1	X	0	0	Switched to not addressed SLV mode; no recognition of own SLA. A START condition will be transmitted when the bus becomes free
		No I2CDAT action	1	X	0	1	Switched to not addressed SLV mode; Own SLA will be recognized. A START condition will be transmitted when the bus becomes free.

Table 5. Slave Transmitter Mode

STATUS CODE (I2CSTA)	STATUS OF THE I²C BUS AND SIO HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO HARDWARE
		TO/FROM I2CDAT	TO I2CCON				
			STA	STO	SI	AA	
A8H	Own SLA+R has been received; ACK has been returned	Load data byte or	X	X	0	0	Last data byte will be transmitted and ACK bit will be received
		load data byte	X	X	0	1	Data byte will be transmitted; ACK will be received
B0H	Arbitration lost in SLA+R/$\overline{W}$ as master; Own SLA+R has been received, ACK has been returned	Load data byte or	X	X	0	0	Last data byte will be transmitted and ACK bit will be received
		load data byte	X	X	0	1	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	Last data byte will be transmitted and ACK bit will be received
		load data byte	X	X	0	1	Data byte will be transmitted; ACK bit will be received
C0H	Data byte in I2CDAT has been transmitted; NOT ACK has been received	No I2CDAT action or	0	X	0	0	Switched to not addressed SLV mode; no recognition of own SLA
		no I2CDAT action or	0	X	0	1	Switched to not addressed SLV mode; Own SLA will be recognized
		no I2CDAT action or	1	X	0	0	Switched to not addressed SLV mode; no recognition of own SLA. A START condition will be transmitted when the bus becomes free
		no I2CDAT action	1	X	0	1	Switched to not addressed SLV mode; Own SLA will be recognized. 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	Switched to not addressed SLV mode; no recognition of own SLA
		no I2CDAT action or	0	X	0	1	Switched to not addressed SLV mode; Own SLA will be recognized
		no I2CDAT action or	1	X	0	0	Switched to not addressed SLV mode; no recognition of own SLA. A START condition will be transmitted when the bus becomes free
		no I2CDAT action	1	X	0	1	Switched to not addressed SLV mode; Own SLA will be recognized. A START condition will be transmitted when the bus becomes free.

Table 6. Miscellaneous States

STATUS CODE (I2CSTA)	STATUS OF THE I²C BUS AND SIO HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO HARDWARE
		TO/FROM I2CDAT	TO I2CCON				
			STA	STO	SI	AA	
F8H	On reset or STOP	No I2CDAT action	1	X	0	X	Go into master mode; send START
		No I2CDAT action	0	X	0	0	No recognition of own SLA
		No I2CDAT action	0	X	0	1	Will recognize own SLA
70H	Bus error SDA stuck LOW						Reset SIO (Requires reset to return to state F8H)
90H	Bus error SCL stuck LOW						Reset SIO (Requires reset to return to state F8H)
00H	Bus error during master or slave mode, due to illegal START or STOP condition						Reset SIO (Requires reset to return to state F8H)

Slave Transmitter Mode: In the slave transmitter mode, a number of data bytes are transmitted to a master receiver (see Figure 5). Data transfer is initialized as in the slave receiver mode. When I2CADR and I2CCON have been initialized, SIO waits until it is addressed by its own slave address followed by the data direction bit which must be "1" (R) for SIO 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 and a valid status code can be read from I2CSTA. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status codes is detailed in Table 5. The slave transmitter mode may also be entered if arbitration is lost while SIO is in the master mode (see state B0H).

If the AA bit is reset during a transfer, SIO will transmit the last byte of the transfer and enter state C8H. SIO 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 1s as serial data. While AA is reset, SIO 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.

Miscellaneous States: There are four I2CSTA codes that do not correspond to a defined SIO hardware state (see Table 6). These are discussed below.

I2CSTA = F8H:
This status code indicates that no relevant information is available because the serial interrupt flag, SI, is not yet set. This occurs on a STOP condition and when SIO is not involved in a serial transfer.

I2CSTA = 00H:
This status code indicates that a bus error has occurred during an SIO 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 SIO signals. When a bus error occurs, SI is set. To recover from a bus error, the microcontroller must send an external reset signal to reset the SIO.

I2CSTA = 70H:
This status code indicates that the SDA line is stuck LOW when the SIO, in master mode, is trying to send a START condition.

I2CSTA = 90H:
This status code indicates that the SCL line is stuck LOW.

Some Special Cases: The SIO hardware has facilities to handle the following special cases that may occur during a serial transfer:

- 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 6). Until this occurs, arbitration is not lost by either master since they were both transmitting the same data.

If the SIO hardware 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.

- 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 Figures 2 and 3).

NOTE: In order to exit state 38H, a Timeout, 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.

- 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 '64 concludes that no other master is using the bus and sends a START condition.

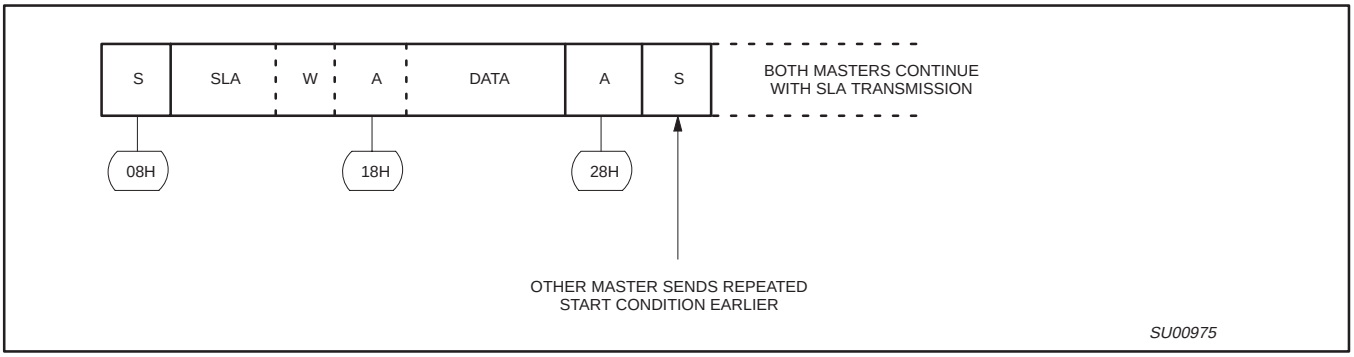


Figure 6. Simultaneous repeated START conditions from 2 masters

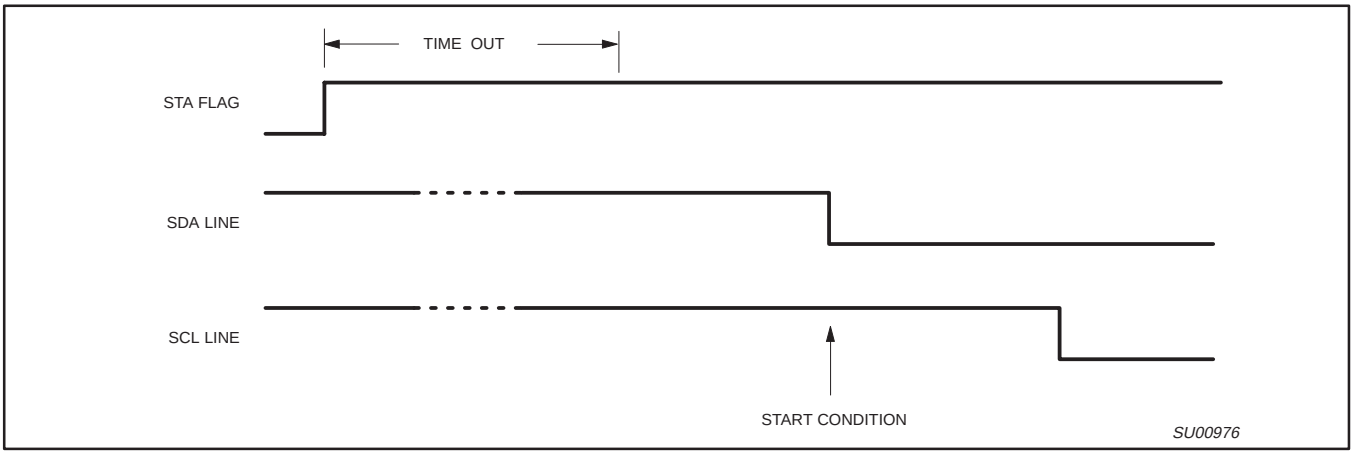


Figure 7. Forced access to a busy I2C-bus

- 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 SIO hardware 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 '64 concludes that this is a bus error and behaves in a manner described on page 5 under “Time-out Register”.

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 8). The SIO hardware 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 SIO, state 08H is entered and the serial transfer continues. If the SDA line is not released by the slave pulling it LOW, then the SIO 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 an external reset signal in order to reset the SIO.

If a forced bus access occurs or a repeated START condition is transmitted while SDA is obstructed (pulled LOW), the SIO hardware 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.

- 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 SIO hardware 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, SIO 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 6. The microcontroller must send an external reset signal to reset the SIO.

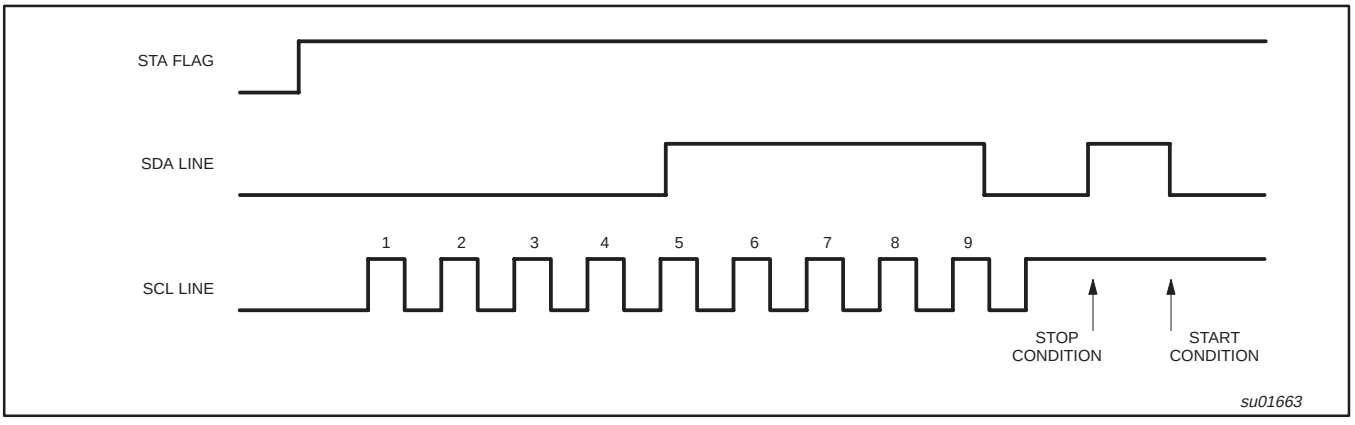


Figure 8. Recovering from a bus obstruction caused by a LOW level on SDA

I2C-BUS TIMING DIAGRAMS

The diagrams (Figures 9 to 12) illustrate typical timing diagrams for the PCA9564 in master/slave functions.

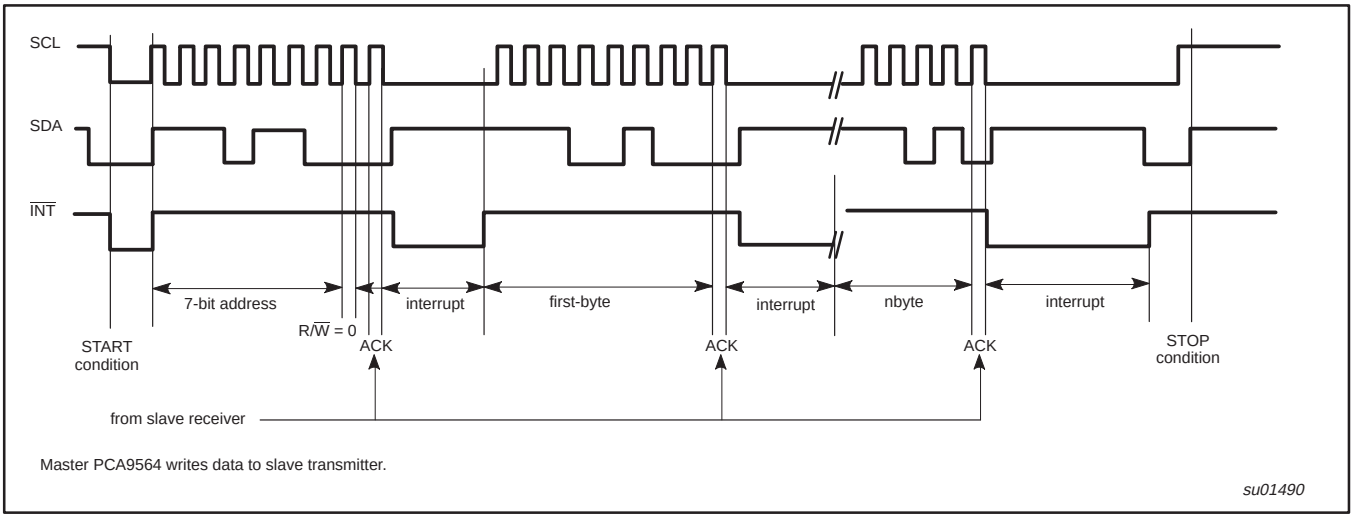


Figure 9. Bus timing diagram; master transmitter mode

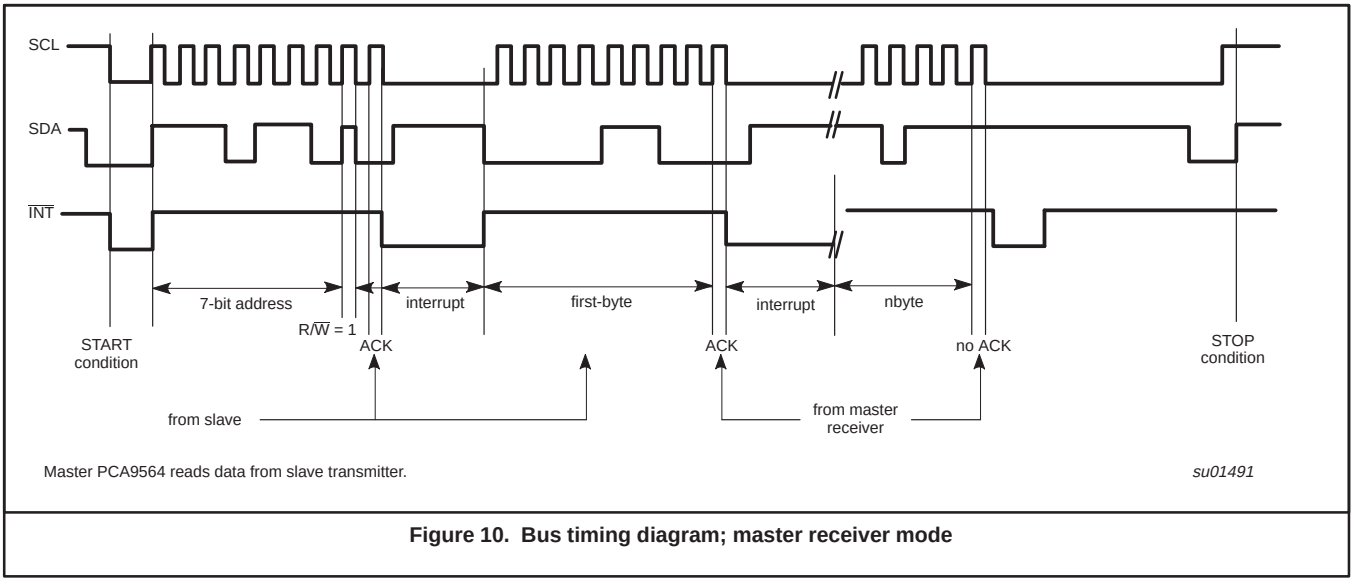


Figure 10. Bus timing diagram; master receiver mode

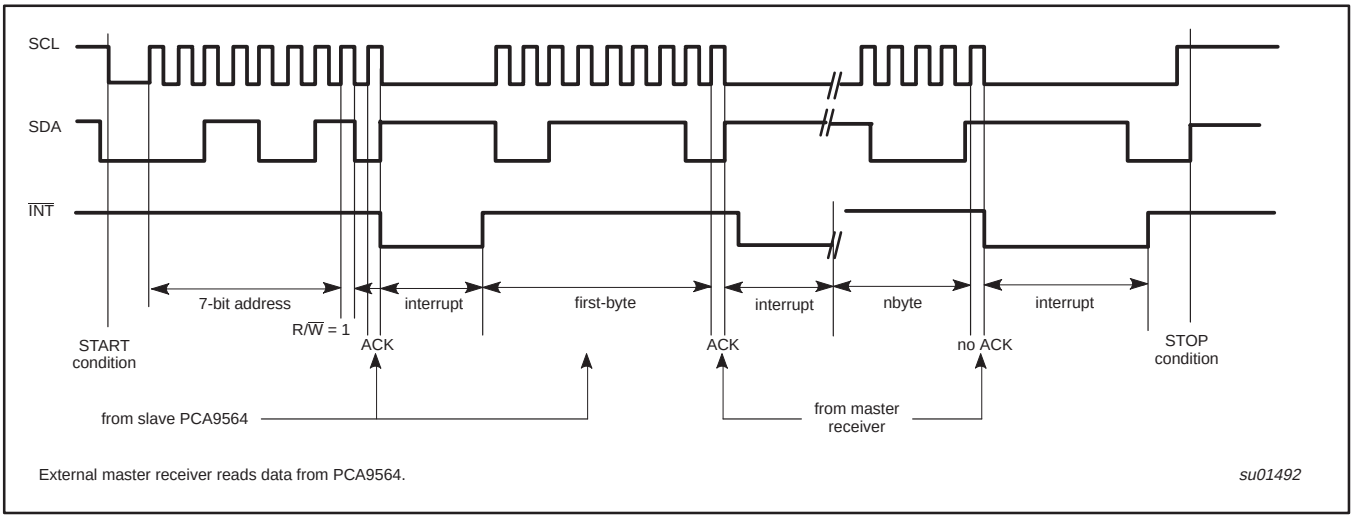


Figure 11. Bus timing diagram; slave transmitter mode

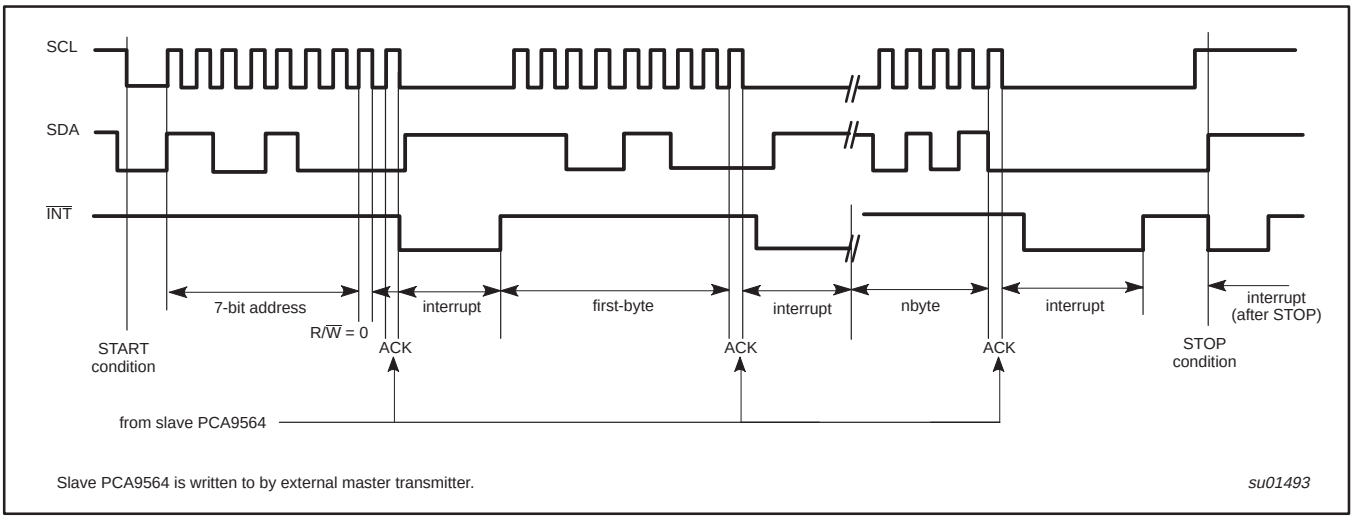


Figure 12. Bus timing diagram; slave receiver mode

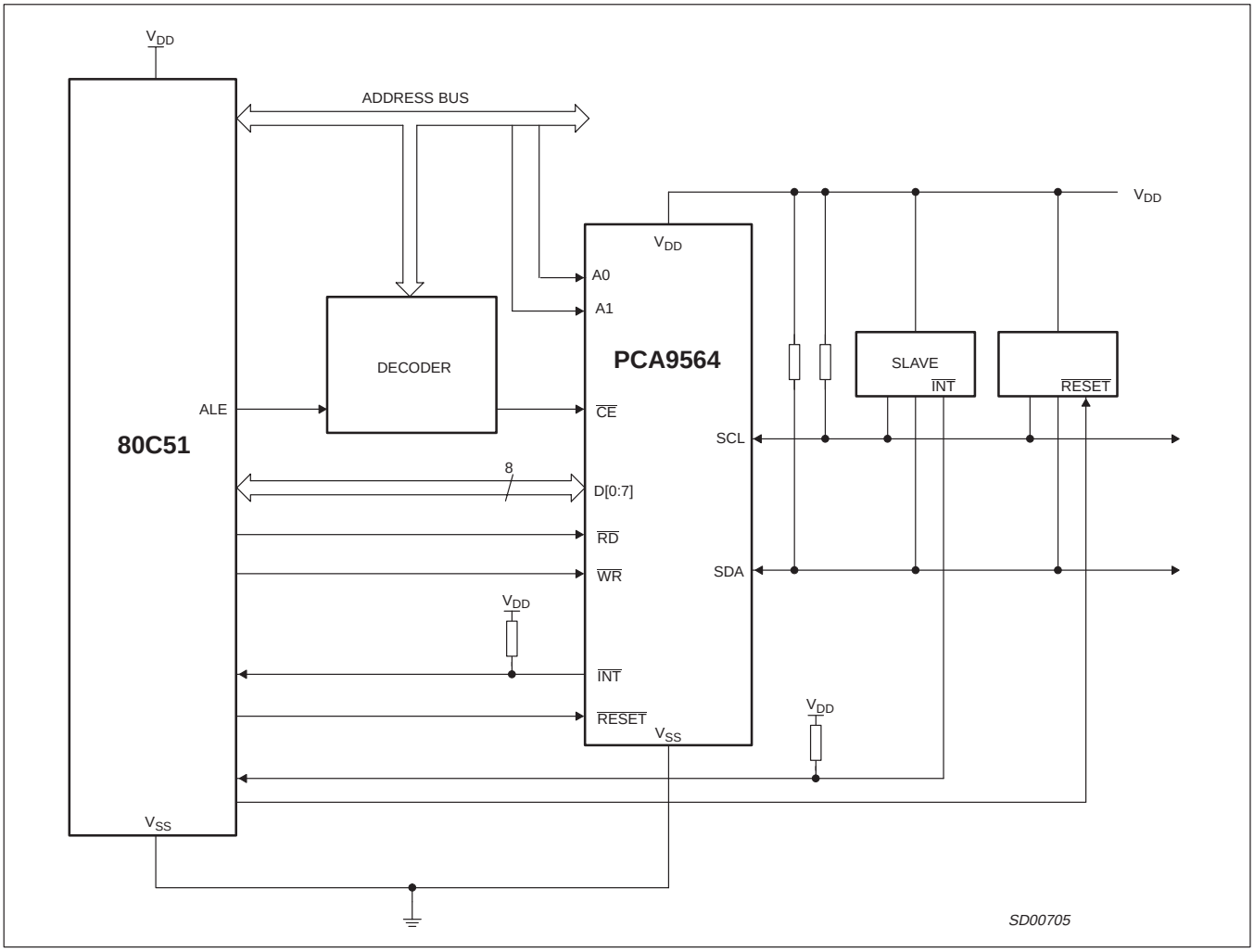


Figure 13. Application diagram using the 80C51

SPECIFIC APPLICATIONS

The PCA9564 is a parallel bus to I²C bus controller that is designed to allow “smart” devices to interface with I²C or SMBus components, where the “smart” device does not have an integrated I²C port and the designer does not want to “bit-bang” the I²C port. The PCA9564 can also be used to add more I²C 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 PC board.

ADD I²C-BUS PORT

As shown in Figure 14, the PCA9564 converts 8-bits of parallel data into a multiple master capable I²C port for microcontrollers, microprocessors, custom ASICs, DSPs, etc., that need to interface with I²C or SMBus components.

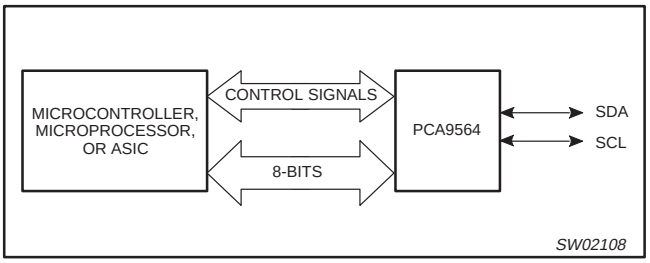


Figure 14. Adding I²C-bus Port Application

ADD ADDITIONAL I²C-BUS PORTS

The PCA9564 can be used to convert 8-bit parallel data into additional multiple master capable I²C port as shown in Figure 15. It is used if the microcontroller, microprocessor, custom ASIC, DSP, etc., already have an I²C port but need one or more additional I²C ports to interface with more I²C 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).

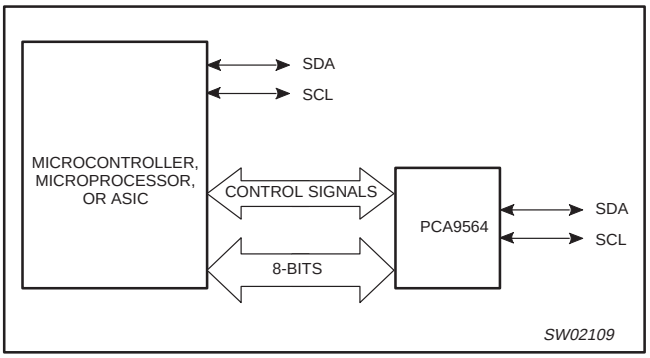


Figure 15. Adding Additional I²C-bus Ports Application

PCA8584 MIGRATION PATH

The PCA9564 does the same type of parallel to serial conversion as the PCF8584. Although not footprint or code compatible, the PCA9564 provides improvements such as:

1. Operating at 3.3 V and 2.5 V voltage nodes with 5 V tolerant I/Os
2. Allows interface with I²C or SMBus components at speeds up to 400 kHz.
3. Built-in oscillator provides a cost effective solution since the external clock input is no longer required.
4. Parallel data can be exchanged at speeds up to 50 MHz allowing the use of faster processors.

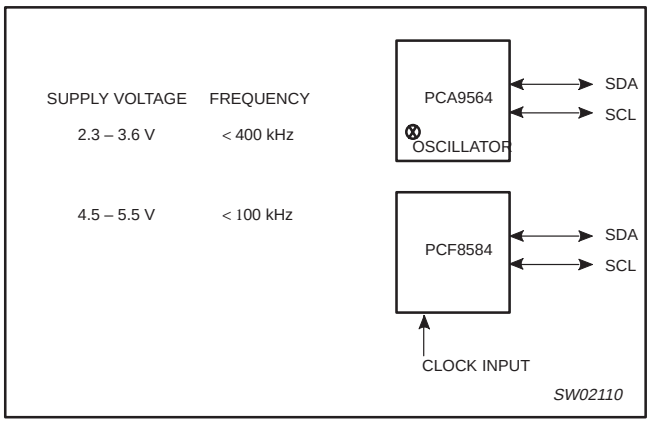


Figure 16. PCF8584 Migration Path

CONVERT 8 BITS OF PARALLEL DATA INTO I²C-BUS SERIAL DATA STREAM

Functioning as a slave transmitter, the PCA9564 can convert 8-bit parallel data into a two-wire I²C data stream as is shown in Figure 17. This would prevent having to run 8 traces across the entire width of the PC board.

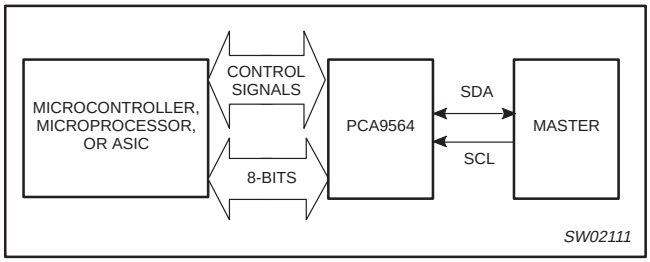


Figure 17. Converting Parallel to Serial Data Application

ABSOLUTE MAXIMUM RATINGS

In accordance with the Absolute Maximum Rating System (IEC 134)

SYMBOL	PARAMETER	CONDITIONS	MIN	MAX	UNIT
V_{DD}	Supply voltage		–0.3	4.6	V
V_I	Voltage range (any input)		–0.8	6.0[1]	V
I_I	DC input current (any input)		–10	10	mA
I_O	DC output current (any output)		–10	10	mA
P_{tot}	Total power dissipation		—	300	mW
P_O	Power dissipation per output		—	50	mW
T_{amb}	Operating ambient temperature		–40	+85	°C
T_{stg}	Storage temperature		–65	+150	°C

NOTE:

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.

HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However, to be totally safe, it is desirable to take precautions appropriate to handling MOS devices. Advice can be found in Data Handbook IC24 under "*Handling MOS devices*".

DC CHARACTERISTICS

V_{DD} = 2.3 V to 3.6 V; T_{amb} = –40 to +85 °C; unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNIT
Supplies						
V_{DD}	Supply voltage		2.3	—	3.6	V
I_{DD}	Supply current	standby	—	0.1	3.0	µA
		operating – no load	—	—	6.0	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		2.0	—	5.5[1]	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}	—	1.7	3	pF
Inputs/outputs D0 to D7						
V_{IL}	LOW-level input voltage		0	—	0.8	V
V_{IH}	HIGH-level input voltage		2.0	—	5.5[1]	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}	—	2.4	4	pF
SDA and SCL						
V_{IL}	LOW-level input voltage		0	—	0.3 V_{DD}	V
V_{IH}	HIGH-level input voltage		0.7 V_{DD}	—	5.5[1]	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	
I_{OL}	LOW-level output current	V_{OL} = 0.4 V	5.0	8.5	—	mA
C_{IO}	Input/output capacitance	$V_I = V_{SS}$ or V_{DD}	—	2.5	4	pF
Outputs $\overline{INT}$						
I_{OL}	LOW-level output current	V_{OL} = 0.4 V	3.0	—	—	mA
I_L	Leakage current	V_O = 0 or 3.6 V	–1	—	1	µA
C_O	Output capacitance	$V_I = V_{SS}$ or V_{DD}	—	2.1	4	pF

NOTE:

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.

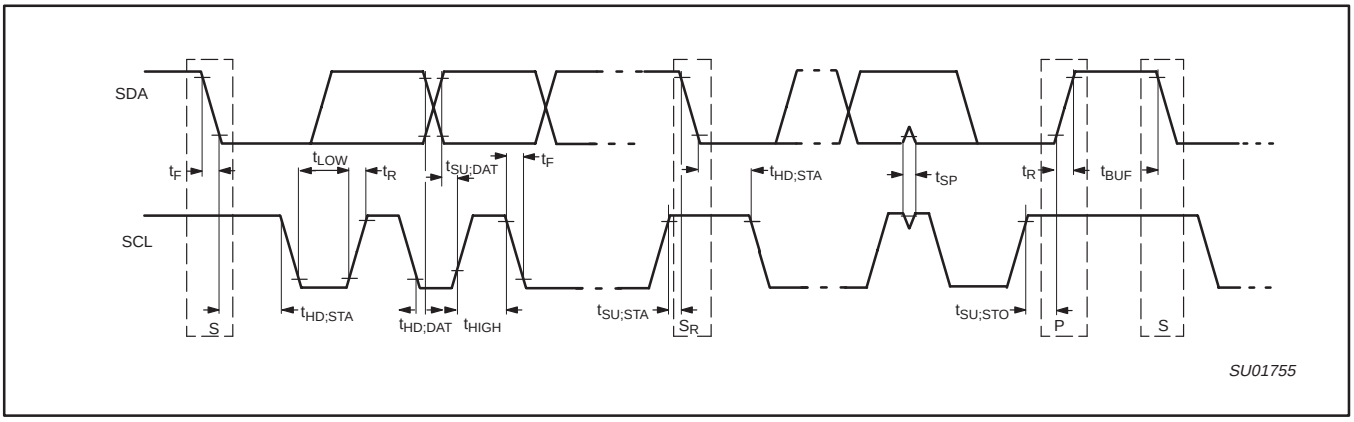


Figure 18. Definition of timing

I2C-BUS 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 to +85 °C; and refer to V_{IL} and V_{IH} with an input voltage of V_{SS} to V_{DD}.

SYMBOL	PARAMETER	STANDARD-MODE I2C-BUS		FAST-MODE I2C-BUS		UNITS
		MIN	MAX	MIN	MAX	
f_{SCL}	Operating frequency	0	100	0	400	kHz
t_{BUF}	Bus free time between STOP and START conditions	4.7	—	1.3	—	µs
$t_{HD;STA}$	Hold time after (repeated) START condition	4.0	—	0.6	—	µs
$t_{SU;STA}$	Repeated START condition setup time	4.7	—	0.6	—	µs
$t_{SU;STO}$	Setup time for STOP condition	4.0	—	0.6	—	µs
$t_{HD;DAT}$	Data in hold time	0	—	0	—	ns
$t_{VD;ACK}$	Valid time for ACK condition	—	0.6	—	0.6	µs
$t_{VD;DAT(L)}$	Data out valid time LOW	—	0.6	—	0.6	µs
$t_{VD;DAT(H)}$	Data out valid time HIGH	—	0.6	—	0.6	µs
$t_{SU;DAT}$	Data setup time	250	—	100	—	ns
t_{LOW}	Clock LOW period	4.7	—	1.3	—	µs
t_{HIGH}	Clock HIGH period	4.0	—	0.6	—	µs
t_F	Clock/Data fall time	—	0.3	—	0.3	µs
t_R	Clock/Data rise time	—	1	—	0.3	µs
t_{SP}	Pulse width of spikes that must be suppressed by the input filters	—	50	—	50	ns

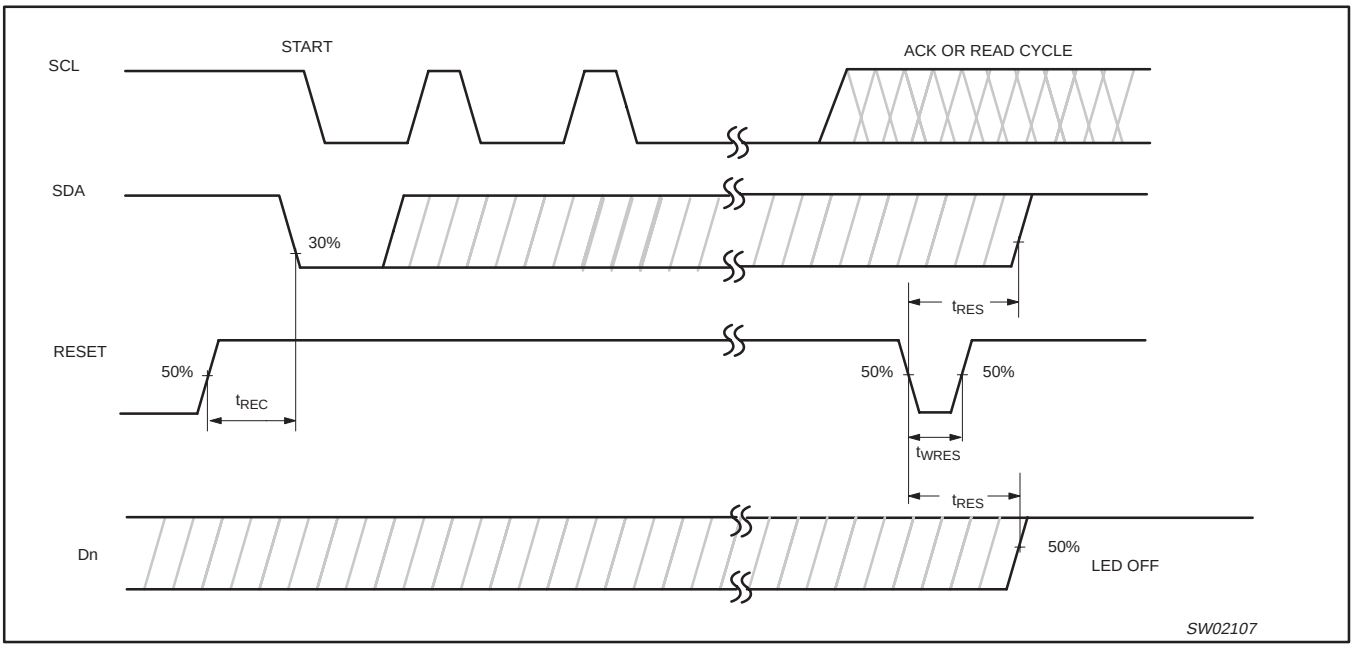


Figure 19. Reset timing

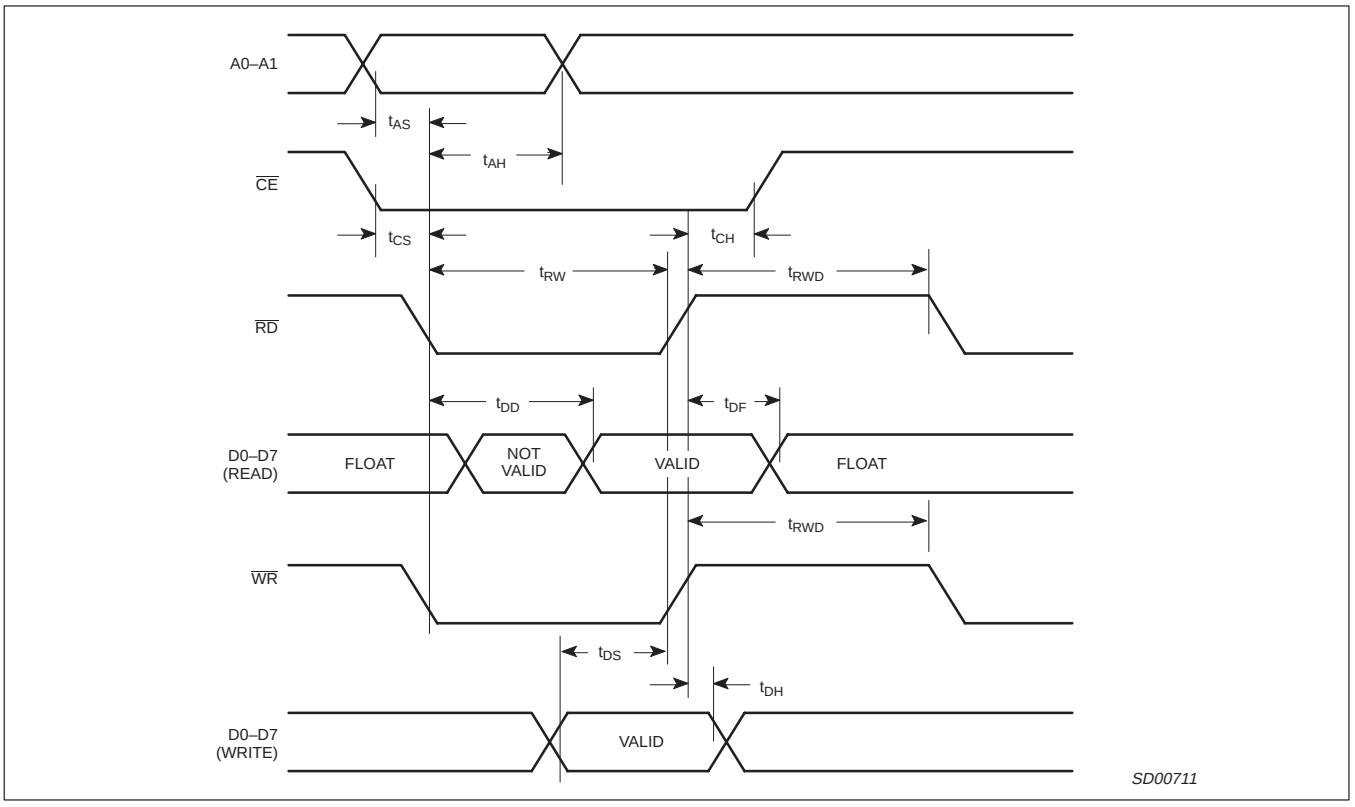


Figure 20. Bus timing

AC 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 page 25 for 2.5 V.)

SYMBOL	PARAMETER	LIMITS		UNIT
		Min	Max	
Reset Timing (See Figure 19)				
t_{WRES}	Reset pulse width	10	—	ns
t_{RES}[4,5]	Time to reset	250	—	ns
t_{REC}	Reset recovery time	0	—	ns
Bus Timing (See Figure 20, 21)				
t_{AS}	A0–A1 setup time to $\overline{RD}$, $\overline{WR}$ LOW	0	—	ns
t_{AH}	A0–A1 hold time from $\overline{RD}$, $\overline{WR}$ LOW	7	—	ns
t_{CS}	$\overline{CE}$ setup time to $\overline{RD}$, $\overline{WR}$ LOW	0	—	ns
t_{CH}	$\overline{CE}$ Hold time from $\overline{RD}$, $\overline{WR}$ LOW	0	—	ns
t_{RW}	$\overline{WR}$, $\overline{RD}$ pulse width (Low time)	7	—	ns
t_{DD}	Data valid after $\overline{RD}$ and $\overline{CE}$ LOW	—	17	ns
t_{DF}	Data bus floating after $\overline{RD}$ or $\overline{CE}$ HIGH	—	17	ns
t_{DS}	Data bus setup time before $\overline{WR}$ or $\overline{CE}$ HIGH (write cycle)	7	—	ns
t_{DH}	Data hold time after $\overline{WR}$ HIGH	0	—	ns
t_{RWD}	High time between read and/or write cycles	12	—	ns

NOTES:
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 Figures 20–21.
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 pullup to V_{DD}.
4. Resetting the device while actively communicating on the bus may cause glitches or an errant STOP condition.
5. Upon reset, the full delay will be the sum of t_{RES} and the RC time constant of the SDA and SCL bus.

AC CHARACTERISTICS (2.5 VOLT) [1, 2, 3]

V_{CC} = 2.5 V ± 0.2 V, T_{amb} = –40 to +85 °C, unless otherwise specified. (See page 24 for 3.3 V.)

SYMBOL	PARAMETER	LIMITS		UNIT
		Min	Max	
Reset Timing (See Figure 19)				
t_{WRES}	Reset pulse width	10	—	ns
t_{RES}[4,5]	Time to reset	250	—	ns
t_{REC}	Reset recovery time	0	—	ns
Bus Timing (See Figure 20, 21)				
t_{AS}	A0–A1 setup time to $\overline{RD}$, $\overline{WR}$ LOW	0	—	ns
t_{AH}	A0–A hold time from $\overline{RD}$, $\overline{WR}$ LOW	9	—	ns
t_{CS}	$\overline{CE}$ setup time to $\overline{RD}$, $\overline{WR}$ LOW	0	—	ns
t_{CH}	$\overline{CE}$ Hold time from $\overline{RD}$, $\overline{WR}$ LOW	0	—	ns
t_{RW}	$\overline{WR}$, $\overline{RD}$ pulse width (low time)	9	—	ns
t_{DD}	Data valid after $\overline{RD}$ and $\overline{CE}$ LOW	—	22	ns
t_{DF}	Data bus floating after $\overline{RD}$ or $\overline{CE}$ HIGH	—	17	ns
t_{DS}	Data bus setup time before $\overline{WR}$ or $\overline{CE}$ HIGH (write cycle)	8	—	ns
t_{DH}	Data hold time after $\overline{WR}$ HIGH	0	—	ns
t_{RWD}	High time between read and/or write cycles	12	—	ns

NOTES:
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 Figures 20–21.
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 pullup to V_{DD}.
4. Resetting the device while actively communicating on the bus may cause glitches or an errant STOP condition.
5. Upon reset, the full delay will be the sum of t_{RES} and the RC time constant of the SDA and SCL bus.

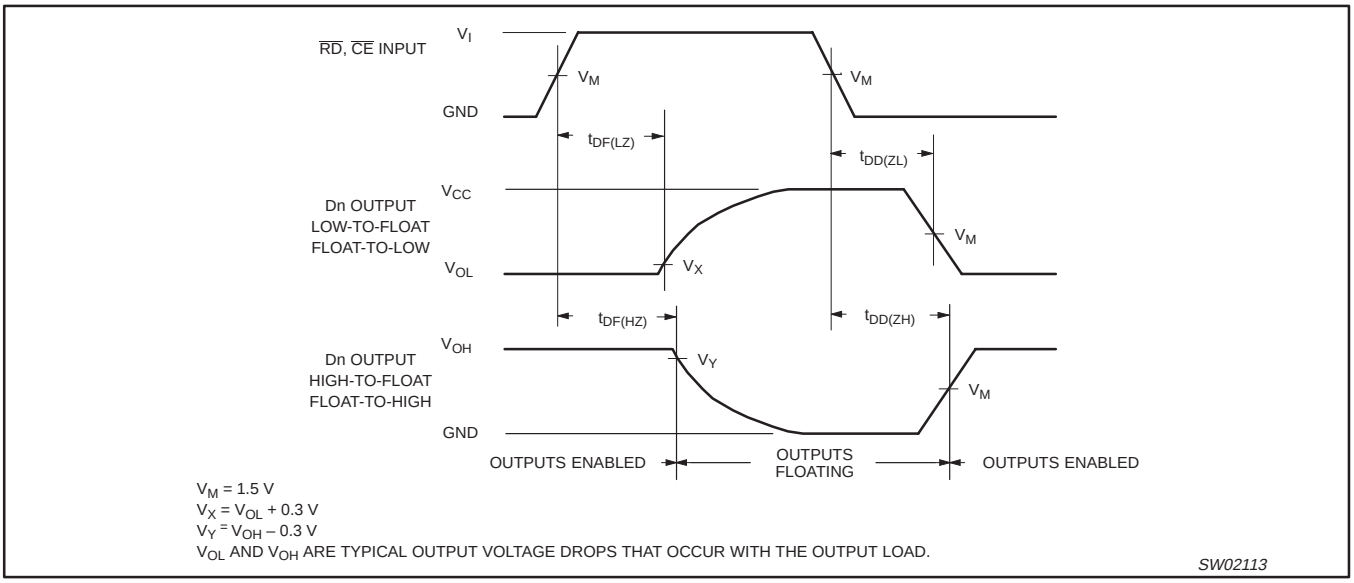


Figure 21. t_{DD} and t_{DF} times

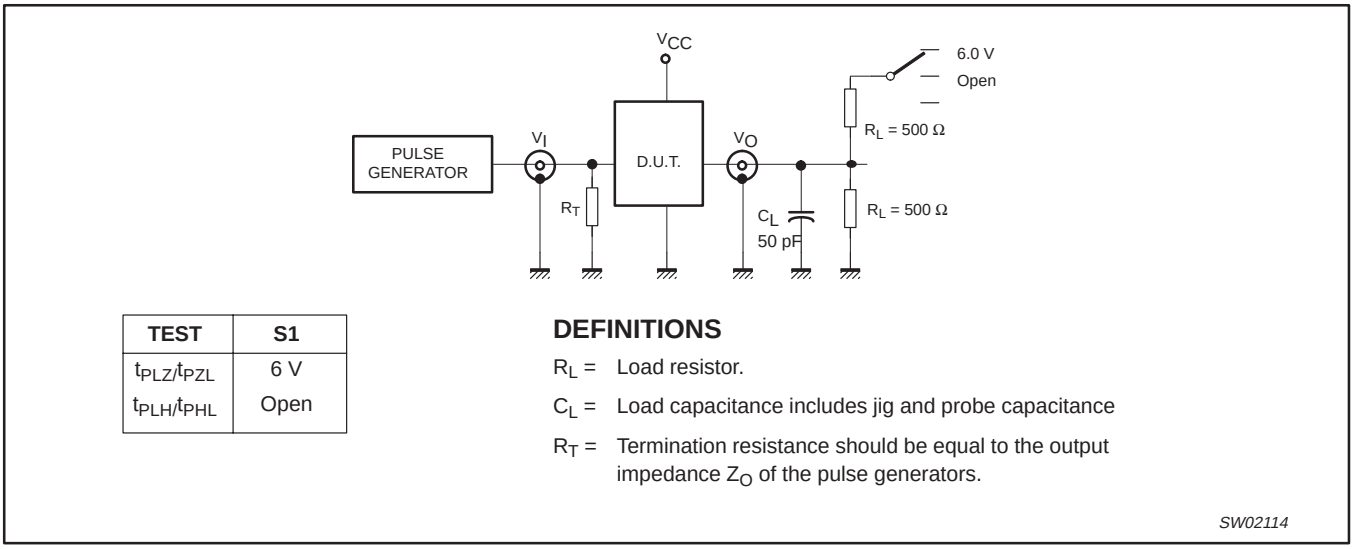
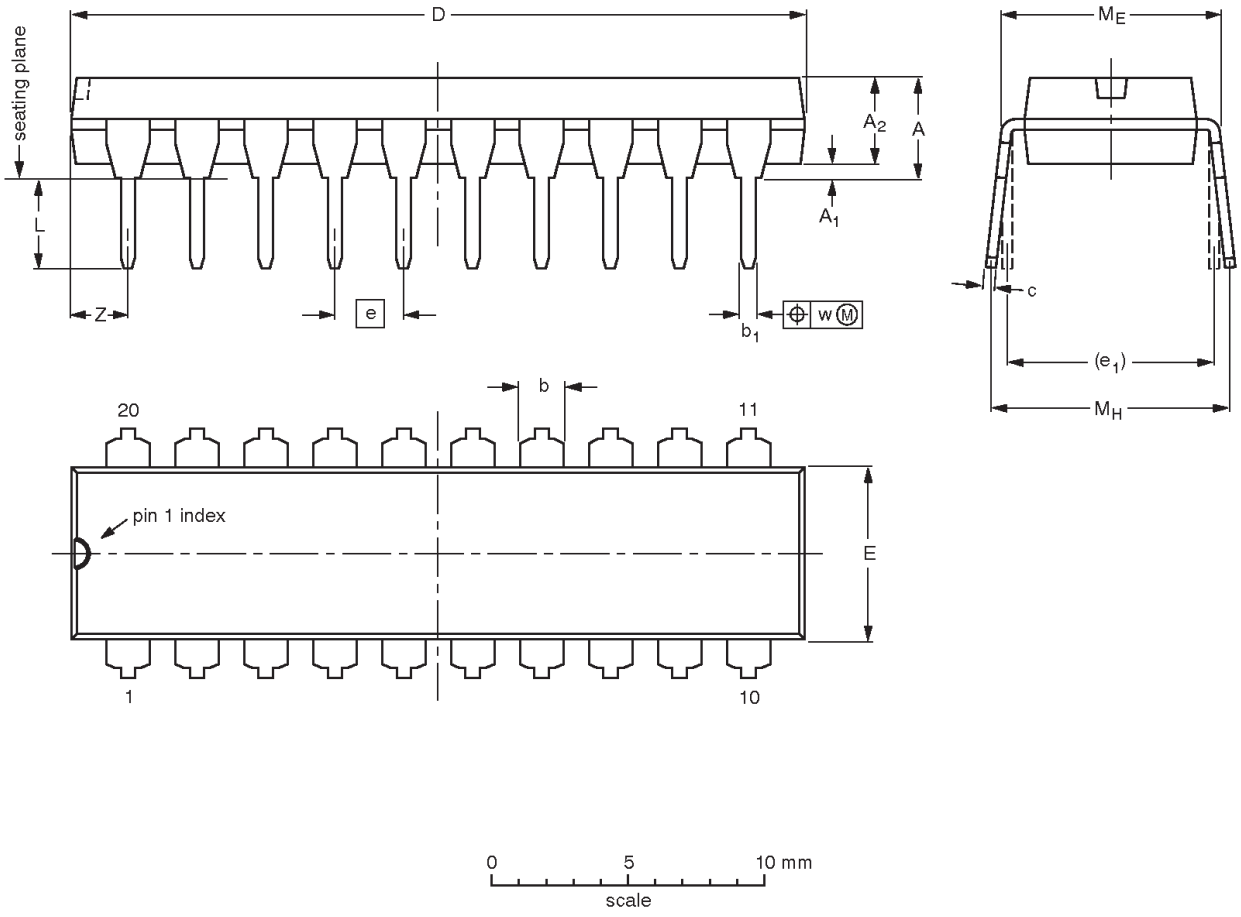


Figure 22. Test circuitry for switching times

DIP20: plastic dual in-line package; 20 leads (300 mil) **SOT146-1**



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

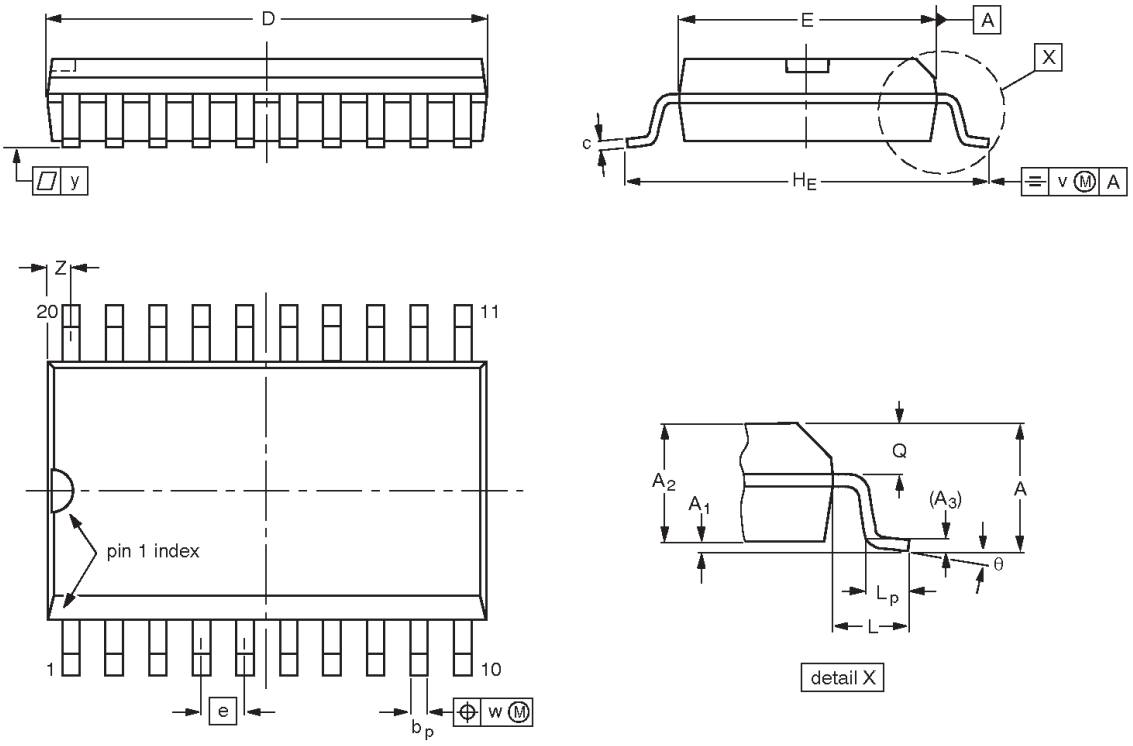
UNIT	A max.	A_1 min.	A_2 max.	b	b_1	c	D (1)	E (1)	e	e_1	L	M_E	M_H	w	Z (1) max.
mm	4.2	0.51	3.2	1.73 1.30	0.53 0.38	0.36 0.23	26.92 26.54	6.40 6.22	2.54	7.62	3.60 3.05	8.25 7.80	10.0 8.3	0.254	2
inches	0.17	0.02	0.13	0.068 0.051	0.021 0.015	0.014 0.009	1.060 1.045	0.25 0.24	0.1	0.3	0.14 0.12	0.32 0.31	0.39 0.33	0.01	0.078

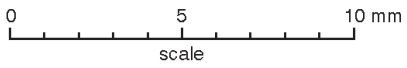
Note

1. Plastic or metal protrusions of 0.25 mm (0.01 inch) maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT146-1		MS-001	SC-603			~~99-12-27~~ 03-02-13

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





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

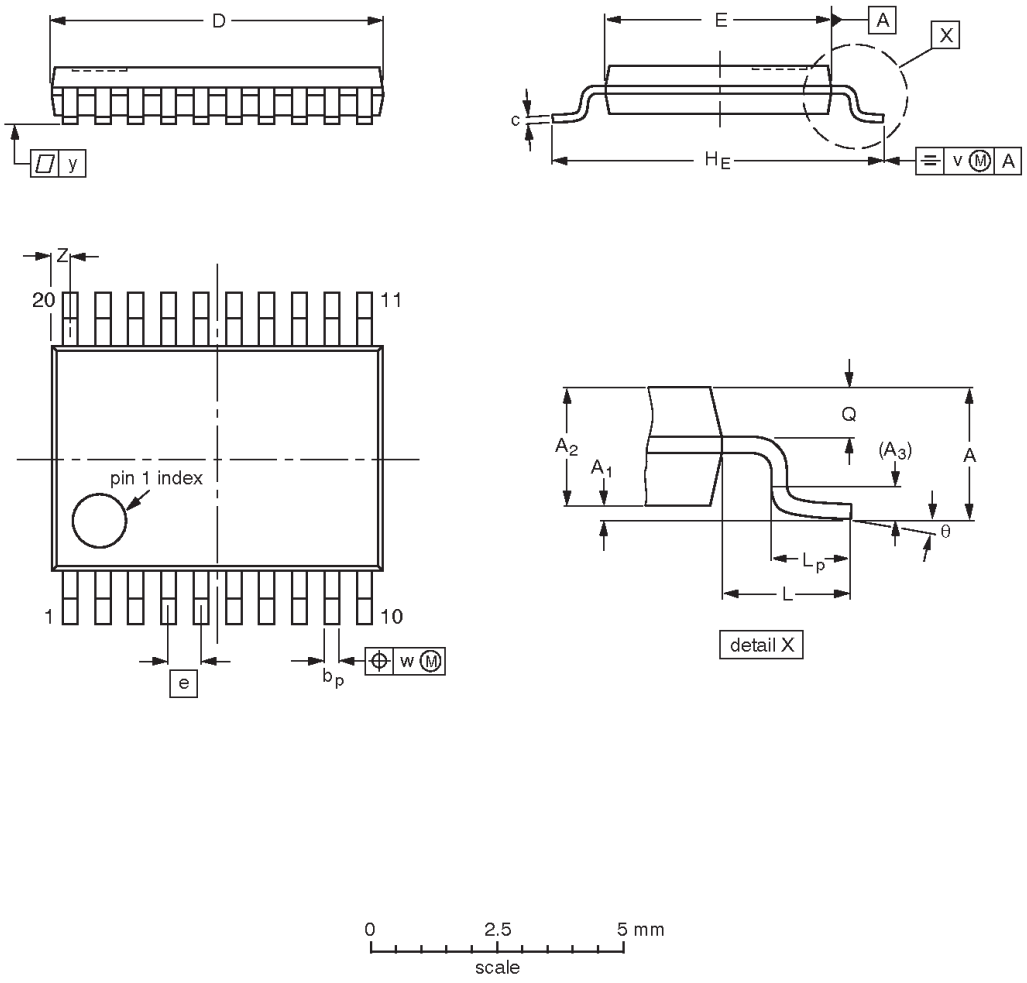
UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_E	L	L_p	Q	v	w	y	$Z^{(1)}$	θ
mm	2.65	0.3 0.1	2.45 2.25	0.25	0.49 0.36	0.32 0.23	13.0 12.6	7.6 7.4	1.27	10.65 10.00	1.4	1.1 0.4	1.1 1.0	0.25	0.25	0.1	0.9 0.4	8° 0°
inches	0.1	0.012 0.004	0.096 0.089	0.01	0.019 0.014	0.013 0.009	0.51 0.49	0.30 0.29	0.05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

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



DIMENSIONS (mm are the original dimensions)

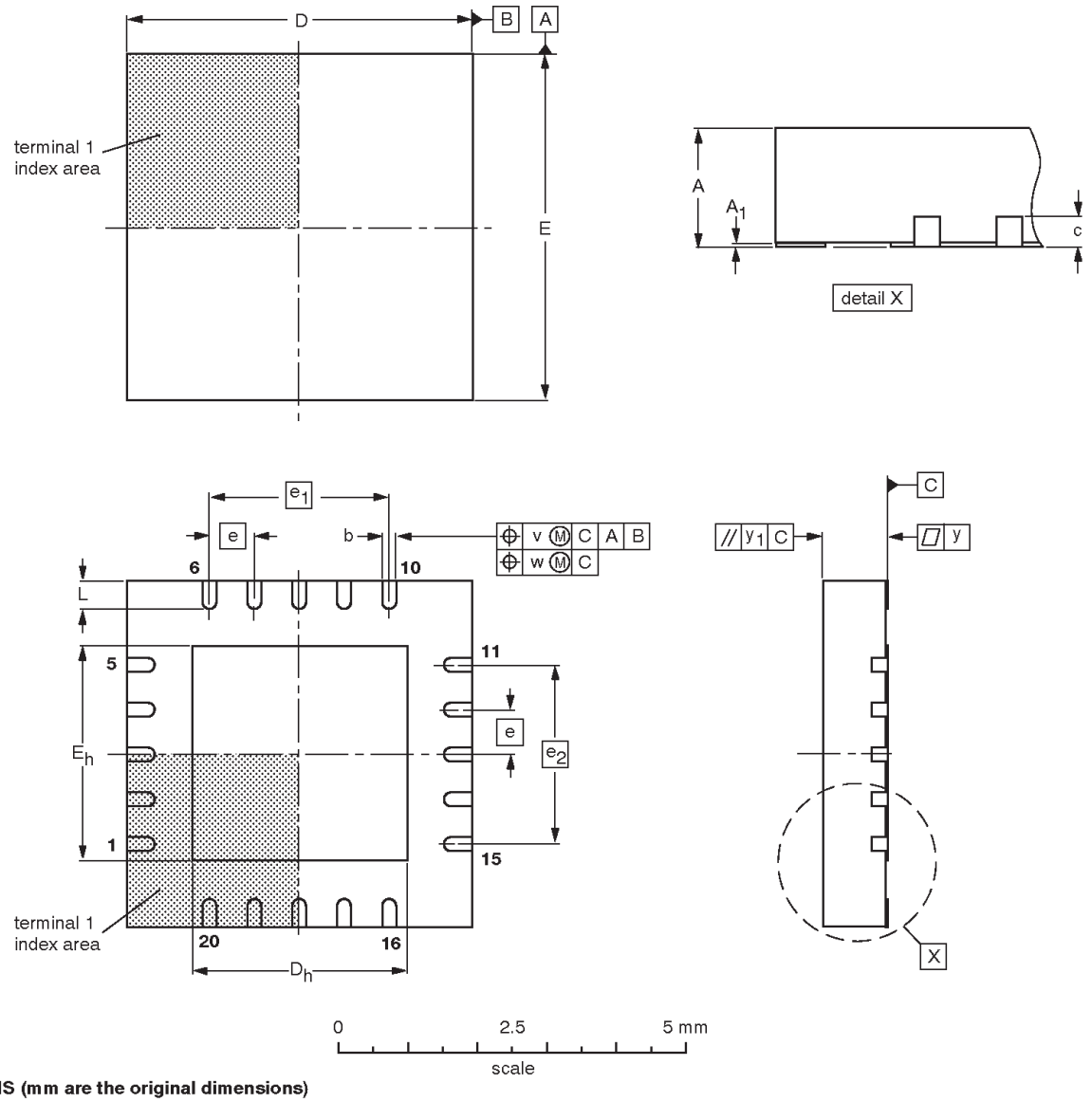
UNIT	A max.	A_1	A_2	A_3	b_p	c	D (1)	E (2)	e	H_E	L	L_p	Q	v	w	y	Z (1)	θ
mm	1.1	0.15 0.05	0.95 0.80	0.25	0.30 0.19	0.2 0.1	6.6 6.4	4.5 4.3	0.65	6.6 6.2	1	0.75 0.50	0.4 0.3	0.2	0.13	0.1	0.5 0.2	8^o 0^o

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

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

REVISION HISTORY

Rev	Date	Description
_4	20060901	**Product data sheet. Supersedes data of 2004 Jun 25 (9397 750 13272).** ● Ordering information table on page 2: added whole wafer package option (PCA9564U). ● Pin description table on page 3: added table note 1 and its reference at HVQFN pin 7 (V_{SS}). ● Section “The Control Register, I2CCON” on page 5: 3rd sentence re-written.
_3	20040625	**Product data sheet (9397 750 13272). Supersedes data of 2003 Apr 02 (9397 750 11353).**
_2	20030402	**Product data (9397 750 11353). ECN 853-2419 29715 Dated 24 March 2003. Supersedes Objective data of 2003 Feb 26 (9397 750 11153).**
_1	20030226	**Objective data (9397 750 11153).**

Legal Information

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 data sheet was published and may differ in case of multiple devices. The latest product status information is available on the Internet at URL http://www.semiconductors.philips.com.

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. Philips 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 Philips Semiconductors sales office. In case of any inconsistency or conflict with the short data sheet, the full data sheet shall prevail.

Disclaimers

General — Information in this document is believed to be accurate and reliable. However, Philips 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.

Right to make changes — Philips 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 — Philips Semiconductors products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of a Philips Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. Philips Semiconductors accepts no liability for inclusion and/or use of Philips 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. Philips Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

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

Terms and conditions of sale — Philips Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at **http://www.semiconductors.philips.com/profile/terms**, including those pertaining to warranty, intellectual property rights infringement and limitation of liability, unless explicitly otherwise agreed to in writing by Philips Semiconductors. In case of any inconsistency or conflict between information in this document and such terms and conditions, the latter will prevail.

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.

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.

Contact information

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

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

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

Date of release: 20060901
Document identifier: PCA9564_4