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

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

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


## Contact us

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

# PCA9641

## 2-channel I²C-bus master arbiter

Rev. 2.1 — 27 October 2015 Product data sheet

## 1. General description

The PCA9641 is a 2-to-1 I²C master demultiplexer with an arbiter function. It is designed for high reliability dual master I²C-bus applications where correct system operation is required, even when two I²C-bus masters issue their commands at the same time. The arbiter will select a winner and let it work uninterrupted, and the losing master will take control of the I²C-bus after the winner has finished. The arbiter also allows for queued requests where a master requests the downstream bus while the other master has control.

A race condition occurs when two masters try to access the downstream I²C-bus at almost the same time. The PCA9641 intelligently selects one winning master and the losing master gains control of the bus after the winning master gives up the bus or the reserve time has expired.

Multiple transactions can be done without interruption. The time needed for multiple transactions on the downstream bus can be reserved by programming the Reserve Time register. During the reserve time, the downstream bus cannot be lost.

Software reset allows a master to send a reset through the I²C-bus to put the PCA9641's registers into the power-on reset condition.

The Device ID of the PCA9641 can be read by the master and includes manufacturer, device type and revision.

When there is no activity on the downstream I²C-bus over 100 ms, optionally the PCA9641 will disconnect the downstream bus to both masters to avoid a lock-up on the I²C-bus.

The interrupt outputs are used to provide an indication of which master has control of the bus, and which master has lost the downstream bus. One interrupt input ($\overline{\text{INT\_IN}}$) collects downstream information and propagates it to the two upstream I²C-buses ($\overline{\text{INT0}}$ and $\overline{\text{INT1}}$) if enabled. $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ are also used to let the master know if the shared mail box has any new mail or if the outgoing mail has not been read by the other master. Those interrupts can be disabled and will not generate an interrupt if the masking option is set.

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 PCA9641. 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 3.3 V devices without any additional protection.

The PCA9641 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.

External pull-up resistors pull the bus to the desired voltage level for each channel. All I/O pins are 3.6 V tolerant.

An active LOW reset input allows the PCA9641A 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 and benefits

- 2-to-1 bidirectional master selector
- Channel selection via I²C-bus
- I²C-bus interface logic; compatible with SMBus standards
- 2 active LOW interrupt outputs to master controllers
- Active LOW reset input
- Software reset
- Four address pins allowing up to 112 different addresses
- Arbitration active when two masters try to take the downstream I²C-bus at the same time
- The winning master controls the downstream bus until it is done, as long as it is within the reserve time
- Bus time-out after 100 ms on an inactive downstream I²C-bus (optional)
- Readable device ID (manufacturer, device type, and revision)
- Bus initialization/recovery function
- Low $R_{on}$ switches
- Allows voltage level translation between 1.8 V, 2.3 V, 2.5 V, 3.3 V and 3.6 V buses
- No glitch on power-up
- Supports hot insertion
- Software identical for both masters
- Operating power supply voltage range of 2.3 V to 3.6 V
- All I/O pins are 3.6 V tolerant
- Up to 1 MHz clock frequency
- ESD protection exceeds 6000 V HBM per JESD22-A114 and 1000 V CDM per JESD22-C101
- Latch-up testing is done to JEDEC Standard JESD78 which exceeds 100 mA
- Packages offered: 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**

| Type number | Topside marking | Package | | |
|---|---|---|---|---|
| | | Name | Description | Version |
| PCA9641BS | 641 | HVQFN16 | plastic thermal enhanced very thin quad flat package; no leads; 16 terminals; body 3 × 3 × 0.85 mm | SOT758-1 |
| PCA9641PW | PCA9641 | TSSOP16 | plastic thin shrink small outline package; 16 leads; body width 4.4 mm | SOT403-1 |

## 4.1 Ordering options

**Table 2. Ordering options**

| Type number | Orderable part number | Package | Packing method | Minimum order quantity | Temperature |
|---|---|---|---|---|---|
| PCA9641BS | PCA9641BSHP | HVQFN16 | Reel 13” Q2/T3 *Standard mark SMD | 6000 | $T_{amb}$ = −40 °C to +85 °C |
| PCA9641PW | PCA9641PWJ | TSSOP16 | Reel 13” Q1/T1 *Standard mark SMD | 2500 | $T_{amb}$ = −40 °C to +85 °C |

## 5. Block diagram

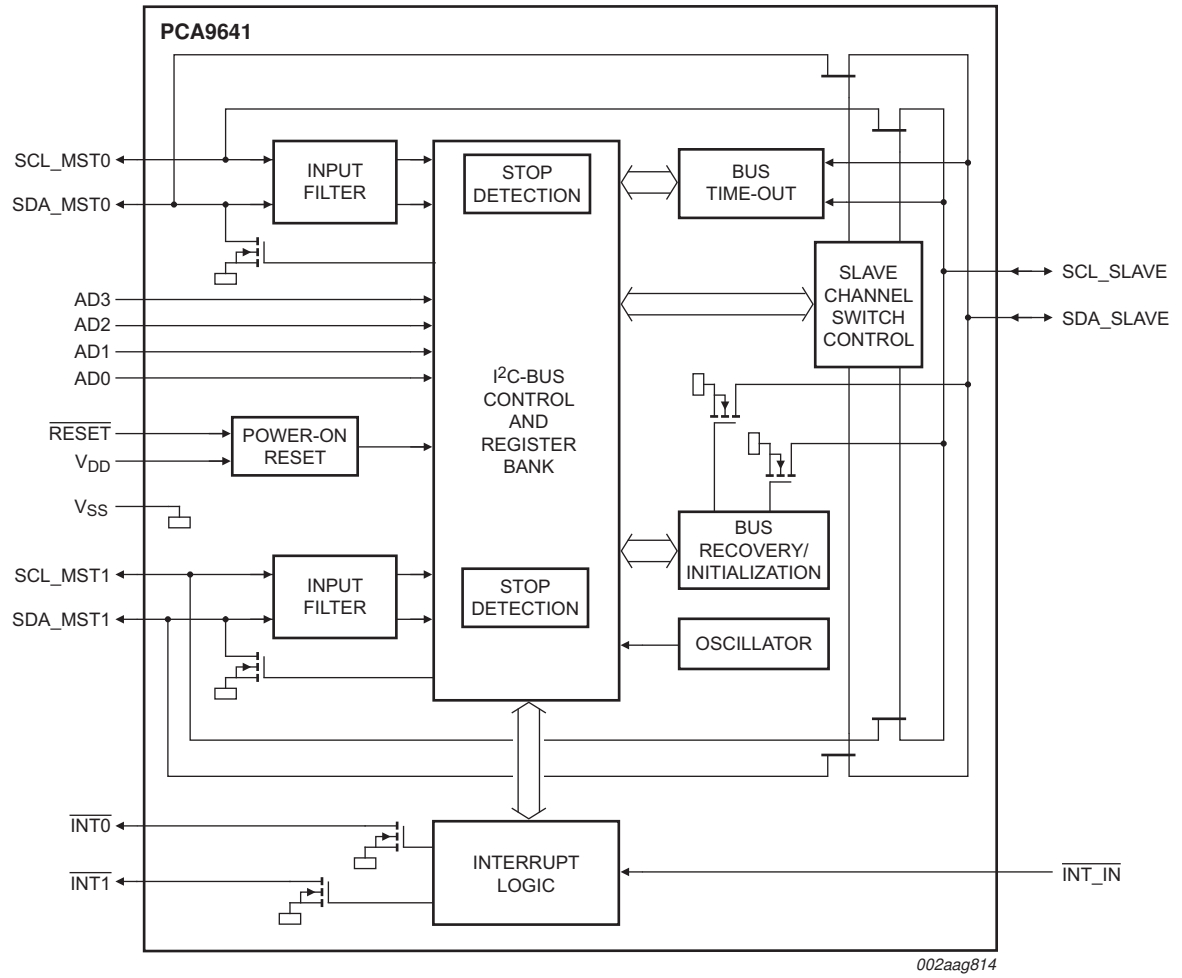



**Fig 1. Block diagram of PCA9641**

# 6. Pinning information

## 6.1 Pinning

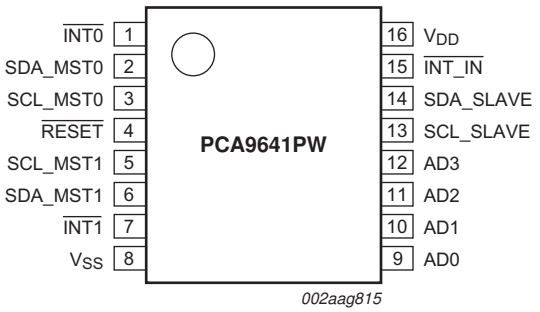



Fig 2. Pin configuration for TSSOP16

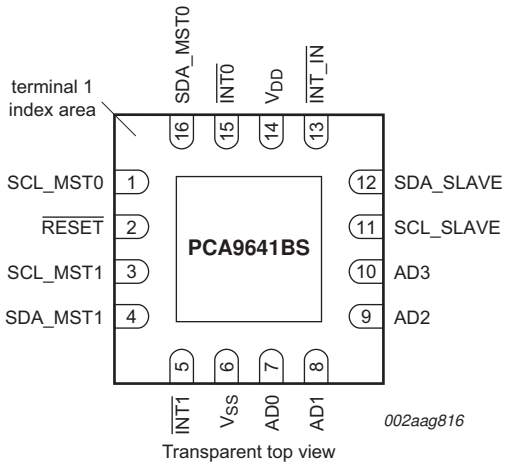



Fig 3. Pin configuration for HVQFN16

## 6.2 Pin description

Table 3. Pin description

| Symbol | Pin | | Description |
|---|---|---|---|
| | 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 |
| AD0 | 9 | 7 | address input 0 (externally held to $V_{SS}$, $V_{DD}$, pull-up to $V_{DD}$ or pull-down to $V_{SS}$) |
| AD1 | 10 | 8 | address input 1 (externally held to $V_{SS}$, $V_{DD}$, pull-up to $V_{DD}$ or pull-down to $V_{SS}$) |
| AD2 | 11 | 9 | address input 2 (externally held to $V_{SS}$, $V_{DD}$, pull-up to $V_{DD}$ or pull-down to $V_{SS}$) |
| AD3 | 12 | 10 | address input 3 (externally held to $V_{SS}$, $V_{DD}$, pull-up to $V_{DD}$ or pull-down to $V_{SS}$) |
| 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] HVQFN16 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.

# 7. Functional description

Refer to Figure 1 “Block diagram of PCA9641”.

## 7.1 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. To conserve power, no internal pull-up resistors are incorporated on the hardware selectable pins and they must be connected to $V_{DD}$, $V_{SS}$, pull-up to $V_{DD}$ or pull-down to $V_{SS}$ directly. PCA9641 can decode 112 addresses, depending on AD3, AD2, AD1 and AD0, and which are found in Table 5 “Address maps”.

At power-up or hardware/software reset, the quinary input pads are sampled and set the slave address of the device internally. To conserve power, once the slave address is determined, the quinary input pads are turned off and will not be sampled until the next time the device is power cycled. Table 4 lists the five possible connections for the quinary input pads along with the external resistor values that must be used.

**Table 4. Quinary input pad connection**

| Pad connection (pins AD3, AD2, AD1, AD0) | Mnemonic | External resistor | |
|---|---|---|---|
| | | Min | Max |
| tie to ground | GND | 0 kΩ | 17.9 kΩ |
| resistor pull-down to ground | PD | 34.8 kΩ | 270 kΩ |
| resistor pull-up to $V_{DD}$ | PU | 31.7 kΩ | 340 kΩ |
| tie to $V_{DD}$ | $V_{DD}$ | 0 kΩ | 22.1 kΩ |

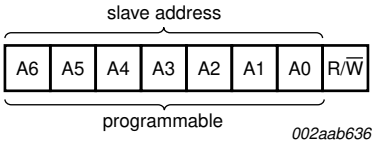



**Fig 4. PCA9641 address**

## 7.2 Address maps

**Table 5. Address maps**

*Do not use any other combination addresses to decode hardware addresses.*

| Pin connectivity | | | | Address of PCA9641 | | | | | | | | Address byte value | | 7-bit hexadecimal address without R/$\overline{W}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AD3 | AD2 | AD1 | AD0 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | R/$\overline{W}$ | Write | Read | |
| $V_{SS}$ | $V_{SS}$ | $V_{SS}$ | $V_{SS}$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 | - | E0h | E1h | 70h |
| $V_{SS}$ | $V_{SS}$ | $V_{SS}$ | $V_{DD}$ | 1 | 1 | 1 | 0 | 0 | 0 | 1 | - | E2h | E3h | 71h |
| $V_{SS}$ | $V_{SS}$ | $V_{DD}$ | $V_{SS}$ | 1 | 1 | 1 | 0 | 0 | 1 | 0 | - | E4h | E5h | 72h |
| $V_{SS}$ | $V_{SS}$ | $V_{DD}$ | $V_{DD}$ | 1 | 1 | 1 | 0 | 0 | 1 | 1 | - | E6h | E7h | 73h |
| $V_{SS}$ | $V_{DD}$ | $V_{SS}$ | $V_{SS}$ | 1 | 1 | 1 | 0 | 1 | 0 | 0 | - | E8h | E9h | 74h |
| $V_{SS}$ | $V_{DD}$ | $V_{SS}$ | $V_{DD}$ | 1 | 1 | 1 | 0 | 1 | 0 | 1 | - | EAh | EBh | 75h |
| $V_{SS}$ | $V_{DD}$ | $V_{DD}$ | $V_{SS}$ | 1 | 1 | 1 | 0 | 1 | 1 | 0 | - | ECh | EDh | 76h |
| $V_{SS}$ | $V_{DD}$ | $V_{DD}$ | $V_{DD}$ | 1 | 1 | 1 | 0 | 1 | 1 | 1 | - | EEh | EFh | 77h |
| $V_{DD}$ | $V_{SS}$ | $V_{SS}$ | PD | 0 | 0 | 0 | 1 | 0 | 0 | 0 | - | 10h | 11h | 08h |
| $V_{DD}$ | $V_{SS}$ | $V_{SS}$ | PU | 0 | 0 | 0 | 1 | 0 | 0 | 1 | - | 12h | 13h | 09h |
| $V_{DD}$ | $V_{SS}$ | $V_{DD}$ | PD | 0 | 0 | 0 | 1 | 0 | 1 | 0 | - | 14h | 15h | 0Ah |
| $V_{DD}$ | $V_{SS}$ | $V_{DD}$ | PU | 0 | 0 | 0 | 1 | 0 | 1 | 1 | - | 16h | 17h | 0Bh |
| $V_{DD}$ | $V_{DD}$ | $V_{SS}$ | PD | 0 | 0 | 0 | 1 | 1 | 0 | 0 | - | 18h | 19h | 0Ch |
| $V_{DD}$ | $V_{DD}$ | $V_{SS}$ | PU | 0 | 0 | 0 | 1 | 1 | 0 | 1 | - | 1Ah | 1Bh | 0Dh |
| $V_{DD}$ | $V_{DD}$ | $V_{DD}$ | PD | 0 | 0 | 0 | 1 | 1 | 1 | 0 | - | 1Ch | 1Dh | 0Eh |
| $V_{DD}$ | $V_{DD}$ | $V_{DD}$ | PU | 0 | 0 | 0 | 1 | 1 | 1 | 1 | - | 1Eh | 1Fh | 0Fh |
| $V_{SS}$ | $V_{SS}$ | PD | $V_{SS}$ | 0 | 0 | 1 | 0 | 0 | 0 | 0 | - | 20h | 21h | 10h |
| $V_{SS}$ | $V_{SS}$ | PD | $V_{DD}$ | 0 | 0 | 1 | 0 | 0 | 0 | 1 | - | 22h | 23h | 11h |
| $V_{SS}$ | $V_{SS}$ | PU | $V_{SS}$ | 0 | 0 | 1 | 0 | 0 | 1 | 0 | - | 24h | 25h | 12h |
| $V_{SS}$ | $V_{SS}$ | PU | $V_{DD}$ | 0 | 0 | 1 | 0 | 0 | 1 | 1 | - | 26h | 27h | 13h |
| $V_{SS}$ | $V_{DD}$ | PD | $V_{SS}$ | 0 | 0 | 1 | 0 | 1 | 0 | 0 | - | 28h | 29h | 14h |
| $V_{SS}$ | $V_{DD}$ | PD | $V_{DD}$ | 0 | 0 | 1 | 0 | 1 | 0 | 1 | - | 2Ah | 2Bh | 15h |
| $V_{SS}$ | $V_{DD}$ | PU | $V_{SS}$ | 0 | 0 | 1 | 0 | 1 | 1 | 0 | - | 2Ch | 2Dh | 16h |
| $V_{SS}$ | $V_{DD}$ | PU | $V_{DD}$ | 0 | 0 | 1 | 0 | 1 | 1 | 1 | - | 2Eh | 2Fh | 17h |
| $V_{DD}$ | $V_{SS}$ | PD | $V_{SS}$ | 0 | 0 | 1 | 1 | 0 | 0 | 0 | - | 30h | 31h | 18h |
| $V_{DD}$ | $V_{SS}$ | PD | $V_{DD}$ | 0 | 0 | 1 | 1 | 0 | 0 | 1 | - | 32h | 33h | 19h |
| $V_{DD}$ | $V_{SS}$ | PU | $V_{SS}$ | 0 | 0 | 1 | 1 | 0 | 1 | 0 | - | 34h | 35h | 1Ah |
| $V_{DD}$ | $V_{SS}$ | PU | $V_{DD}$ | 0 | 0 | 1 | 1 | 0 | 1 | 1 | - | 36h | 37h | 1Bh |
| $V_{DD}$ | $V_{DD}$ | PD | $V_{SS}$ | 0 | 0 | 1 | 1 | 1 | 0 | 0 | - | 38h | 39h | 1Ch |
| $V_{DD}$ | $V_{DD}$ | PD | $V_{DD}$ | 0 | 0 | 1 | 1 | 1 | 0 | 1 | - | 3Ah | 3Bh | 1Dh |
| $V_{DD}$ | $V_{DD}$ | PU | $V_{SS}$ | 0 | 0 | 1 | 1 | 1 | 1 | 0 | - | 3Ch | 3Dh | 1Eh |
| $V_{DD}$ | $V_{DD}$ | PU | $V_{DD}$ | 0 | 0 | 1 | 1 | 1 | 1 | 1 | - | 3Eh | 3Fh | 1Fh |
| $V_{SS}$ | $V_{SS}$ | PD | PD | 0 | 1 | 0 | 0 | 0 | 0 | 0 | - | 40h | 41h | 20h |
| $V_{SS}$ | $V_{SS}$ | PD | PU | 0 | 1 | 0 | 0 | 0 | 0 | 1 | - | 42h | 43h | 21h |
| $V_{SS}$ | $V_{SS}$ | PU | PD | 0 | 1 | 0 | 0 | 0 | 1 | 0 | - | 44h | 45h | 22h |
| $V_{SS}$ | $V_{SS}$ | PU | PU | 0 | 1 | 0 | 0 | 0 | 1 | 1 | - | 46h | 47h | 23h |
| $V_{SS}$ | $V_{DD}$ | PD | PD | 0 | 1 | 0 | 0 | 1 | 0 | 0 | - | 48h | 49h | 24h |
| $V_{SS}$ | $V_{DD}$ | PD | PU | 0 | 1 | 0 | 0 | 1 | 0 | 1 | - | 4Ah | 4Bh | 25h |

**Table 5. Address maps** *…continued*

*Do not use any other combination addresses to decode hardware addresses.*

| Pin connectivity | | | | Address of PCA9641 | | | | | | | | Address byte value | | 7-bit hexadecimal address without R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AD3 | AD2 | AD1 | AD0 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | R/W | Write | Read | |
| $V_{SS}$ | $V_{DD}$ | PU | PD | 0 | 1 | 0 | 0 | 1 | 1 | 0 | - | 4Ch | 4Dh | 26h |
| $V_{SS}$ | $V_{DD}$ | PU | PU | 0 | 1 | 0 | 0 | 1 | 1 | 1 | - | 4Eh | 4Fh | 27h |
| $V_{DD}$ | $V_{SS}$ | PD | PD | 0 | 1 | 0 | 1 | 0 | 0 | 0 | - | 50h | 51h | 28h |
| $V_{DD}$ | $V_{SS}$ | PD | PU | 0 | 1 | 0 | 1 | 0 | 0 | 1 | - | 52h | 53h | 29h |
| $V_{DD}$ | $V_{SS}$ | PU | PD | 0 | 1 | 0 | 1 | 0 | 1 | 0 | - | 54h | 55h | 2Ah |
| $V_{DD}$ | $V_{SS}$ | PU | PU | 0 | 1 | 0 | 1 | 0 | 1 | 1 | - | 56h | 57h | 2Bh |
| $V_{DD}$ | $V_{DD}$ | PD | PD | 0 | 1 | 0 | 1 | 1 | 0 | 0 | - | 58h | 59h | 2Ch |
| $V_{DD}$ | $V_{DD}$ | PD | PU | 0 | 1 | 0 | 1 | 1 | 0 | 1 | - | 5Ah | 5Bh | 2Dh |
| $V_{DD}$ | $V_{DD}$ | PU | PD | 0 | 1 | 0 | 1 | 1 | 1 | 0 | - | 5Ch | 5Dh | 2Eh |
| $V_{DD}$ | $V_{DD}$ | PU | PU | 0 | 1 | 0 | 1 | 1 | 1 | 1 | - | 5Eh | 5Fh | 2Fh |
| $V_{SS}$ | PD | $V_{SS}$ | $V_{SS}$ | 0 | 1 | 1 | 0 | 0 | 0 | 0 | - | 60h | 61h | 30h |
| $V_{SS}$ | PD | $V_{SS}$ | $V_{DD}$ | 0 | 1 | 1 | 0 | 0 | 0 | 1 | - | 62h | 63h | 31h |
| $V_{SS}$ | PD | $V_{DD}$ | $V_{SS}$ | 0 | 1 | 1 | 0 | 0 | 1 | 0 | - | 64h | 65h | 32h |
| $V_{SS}$ | PD | $V_{DD}$ | $V_{DD}$ | 0 | 1 | 1 | 0 | 0 | 1 | 1 | - | 66h | 67h | 33h |
| $V_{SS}$ | PU | $V_{SS}$ | $V_{SS}$ | 0 | 1 | 1 | 0 | 1 | 0 | 0 | - | 68h | 69h | 34h |
| $V_{SS}$ | PU | $V_{SS}$ | $V_{DD}$ | 0 | 1 | 1 | 0 | 1 | 0 | 1 | - | 6Ah | 6Bh | 35h |
| $V_{SS}$ | PU | $V_{DD}$ | $V_{SS}$ | 0 | 1 | 1 | 0 | 1 | 1 | 0 | - | 6Ch | 6Dh | 36h |
| $V_{SS}$ | PU | $V_{DD}$ | $V_{DD}$ | 0 | 1 | 1 | 0 | 1 | 1 | 1 | - | 6Eh | 6Fh | 37h |
| $V_{DD}$ | PD | $V_{SS}$ | $V_{SS}$ | 0 | 1 | 1 | 1 | 0 | 0 | 0 | - | 70h | 71h | 38h |
| $V_{DD}$ | PD | $V_{SS}$ | $V_{DD}$ | 0 | 1 | 1 | 1 | 0 | 0 | 1 | - | 72h | 73h | 39h |
| $V_{DD}$ | PD | $V_{DD}$ | $V_{SS}$ | 0 | 1 | 1 | 1 | 0 | 1 | 0 | - | 74h | 75h | 3Ah |
| $V_{DD}$ | PD | $V_{DD}$ | $V_{DD}$ | 0 | 1 | 1 | 1 | 0 | 1 | 1 | - | 76h | 77h | 3Bh |
| $V_{DD}$ | PU | $V_{SS}$ | $V_{SS}$ | 0 | 1 | 1 | 1 | 1 | 0 | 0 | - | 78h | 79h | 3Ch |
| $V_{DD}$ | PU | $V_{SS}$ | $V_{DD}$ | 0 | 1 | 1 | 1 | 1 | 0 | 1 | - | 7Ah | 7Bh | 3Dh |
| $V_{DD}$ | PU | $V_{DD}$ | $V_{SS}$ | 0 | 1 | 1 | 1 | 1 | 1 | 0 | - | 7Ch | 7Dh | 3Eh |
| $V_{DD}$ | PU | $V_{DD}$ | $V_{DD}$ | 0 | 1 | 1 | 1 | 1 | 1 | 1 | - | 7Eh | 7Fh | 3Fh |
| $V_{SS}$ | PD | $V_{SS}$ | PD | 1 | 0 | 0 | 0 | 0 | 0 | 0 | - | 80h | 81h | 40h |
| $V_{SS}$ | PD | $V_{SS}$ | PU | 1 | 0 | 0 | 0 | 0 | 0 | 1 | - | 82h | 83h | 41h |
| $V_{SS}$ | PD | $V_{DD}$ | PD | 1 | 0 | 0 | 0 | 0 | 1 | 0 | - | 84h | 85h | 42h |
| $V_{SS}$ | PD | $V_{DD}$ | PU | 1 | 0 | 0 | 0 | 0 | 1 | 1 | - | 86h | 87h | 43h |
| $V_{SS}$ | PU | $V_{SS}$ | PD | 1 | 0 | 0 | 0 | 1 | 0 | 0 | - | 88h | 89h | 44h |
| $V_{SS}$ | PU | $V_{SS}$ | PU | 1 | 0 | 0 | 0 | 1 | 0 | 1 | - | 8Ah | 8Bh | 45h |
| $V_{SS}$ | PU | $V_{DD}$ | PD | 1 | 0 | 0 | 0 | 1 | 1 | 0 | - | 8Ch | 8Dh | 46h |
| $V_{SS}$ | PU | $V_{DD}$ | PU | 1 | 0 | 0 | 0 | 1 | 1 | 1 | - | 8Eh | 8Fh | 47h |
| $V_{DD}$ | PD | $V_{SS}$ | PD | 1 | 0 | 0 | 1 | 0 | 0 | 0 | - | 90h | 91h | 48h |
| $V_{DD}$ | PD | $V_{SS}$ | PU | 1 | 0 | 0 | 1 | 0 | 0 | 1 | - | 92h | 93h | 49h |
| $V_{DD}$ | PD | $V_{DD}$ | PD | 1 | 0 | 0 | 1 | 0 | 1 | 0 | - | 94h | 95h | 4Ah |
| $V_{DD}$ | PD | $V_{DD}$ | PU | 1 | 0 | 0 | 1 | 0 | 1 | 1 | - | 96h | 97h | 4Bh |
| $V_{DD}$ | PU | $V_{SS}$ | PD | 1 | 0 | 0 | 1 | 1 | 0 | 0 | - | 98h | 99h | 4Ch |

**Table 5. Address maps** *…continued*

*Do not use any other combination addresses to decode hardware addresses.*

| Pin connectivity | | | | Address of PCA9641 | | | | | | | | Address byte value | | 7-bit hexadecimal address without R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AD3 | AD2 | AD1 | AD0 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | R/$\overline{W}$ | Write | Read | |
| $V_{DD}$ | PU | $V_{SS}$ | PU | 1 | 0 | 0 | 1 | 1 | 0 | 1 | - | 9Ah | 9Bh | 4Dh |
| $V_{DD}$ | PU | $V_{DD}$ | PD | 1 | 0 | 0 | 1 | 1 | 1 | 0 | - | 9Ch | 9Dh | 4Eh |
| $V_{DD}$ | PU | $V_{DD}$ | PU | 1 | 0 | 0 | 1 | 1 | 1 | 1 | - | 9Eh | 9Fh | 4Fh |
| $V_{SS}$ | PD | PD | $V_{SS}$ | 1 | 0 | 1 | 0 | 0 | 0 | 0 | - | A0h | A1h | 50h |
| $V_{SS}$ | PD | PD | $V_{DD}$ | 1 | 0 | 1 | 0 | 0 | 0 | 1 | - | A2h | A3h | 51h |
| $V_{SS}$ | PD | PU | $V_{SS}$ | 1 | 0 | 1 | 0 | 0 | 1 | 0 | - | A4h | A5h | 52h |
| $V_{SS}$ | PD | PU | $V_{DD}$ | 1 | 0 | 1 | 0 | 0 | 1 | 1 | - | A6h | A7h | 53h |
| $V_{SS}$ | PU | PD | $V_{SS}$ | 1 | 0 | 1 | 0 | 1 | 0 | 0 | - | A8h | A9h | 54h |
| $V_{SS}$ | PU | PD | $V_{DD}$ | 1 | 0 | 1 | 0 | 1 | 0 | 1 | - | AAh | ABh | 55h |
| $V_{SS}$ | PU | PU | $V_{SS}$ | 1 | 0 | 1 | 0 | 1 | 1 | 0 | - | ACh | ADh | 56h |
| $V_{SS}$ | PU | PU | $V_{DD}$ | 1 | 0 | 1 | 0 | 1 | 1 | 1 | - | AEh | AFh | 57h |
| $V_{DD}$ | PD | PD | $V_{SS}$ | 1 | 0 | 1 | 1 | 0 | 0 | 0 | - | B0h | B1h | 58h |
| $V_{DD}$ | PD | PD | $V_{DD}$ | 1 | 0 | 1 | 1 | 0 | 0 | 1 | - | B2h | B3h | 59h |
| $V_{DD}$ | PD | PU | $V_{SS}$ | 1 | 0 | 1 | 1 | 0 | 1 | 0 | - | B4h | B5h | 5Ah |
| $V_{DD}$ | PD | PU | $V_{DD}$ | 1 | 0 | 1 | 1 | 0 | 1 | 1 | - | B6h | B7h | 5Bh |
| $V_{DD}$ | PU | PD | $V_{SS}$ | 1 | 0 | 1 | 1 | 1 | 0 | 0 | - | B8h | B9h | 5Ch |
| $V_{DD}$ | PU | PD | $V_{DD}$ | 1 | 0 | 1 | 1 | 1 | 0 | 1 | - | BAh | BBh | 5Dh |
| $V_{DD}$ | PU | PU | $V_{SS}$ | 1 | 0 | 1 | 1 | 1 | 1 | 0 | - | BCh | BDh | 5Eh |
| $V_{DD}$ | PU | PU | $V_{DD}$ | 1 | 0 | 1 | 1 | 1 | 1 | 1 | - | BEh | BFh | 5Fh |
| $V_{SS}$ | PD | PD | PD | 1 | 1 | 0 | 0 | 0 | 0 | 0 | - | C0h | C1h | 60h |
| $V_{SS}$ | PD | PD | PU | 1 | 1 | 0 | 0 | 0 | 0 | 1 | - | C2h | C3h | 61h |
| $V_{SS}$ | PD | PU | PD | 1 | 1 | 0 | 0 | 0 | 1 | 0 | - | C4h | C5h | 62h |
| $V_{SS}$ | PD | PU | PU | 1 | 1 | 0 | 0 | 0 | 1 | 1 | - | C6h | C7h | 63h |
| $V_{SS}$ | PU | PD | PD | 1 | 1 | 0 | 0 | 1 | 0 | 0 | - | C8h | C9h | 64h |
| $V_{SS}$ | PU | PD | PU | 1 | 1 | 0 | 0 | 1 | 0 | 1 | - | CAh | CBh | 65h |
| $V_{SS}$ | PU | PU | PD | 1 | 1 | 0 | 0 | 1 | 1 | 0 | - | CCh | CDh | 66h |
| $V_{SS}$ | PU | PU | PU | 1 | 1 | 0 | 0 | 1 | 1 | 1 | - | CEh | CFh | 67h |
| $V_{DD}$ | PD | PD | PD | 1 | 1 | 0 | 1 | 0 | 0 | 0 | - | D0h | D1h | 68h |
| $V_{DD}$ | PD | PD | PU | 1 | 1 | 0 | 1 | 0 | 0 | 1 | - | D2h | D3h | 69h |
| $V_{DD}$ | PD | PU | PD | 1 | 1 | 0 | 1 | 0 | 1 | 0 | - | D4h | D5h | 6Ah |
| $V_{DD}$ | PD | PU | PU | 1 | 1 | 0 | 1 | 0 | 1 | 1 | - | D6h | D7h | 6Bh |
| $V_{DD}$ | PU | PD | PD | 1 | 1 | 0 | 1 | 1 | 0 | 0 | - | D8h | D9h | 6Ch |
| $V_{DD}$ | PU | PD | PU | 1 | 1 | 0 | 1 | 1 | 0 | 1 | - | DAh | DBh | 6Dh |
| $V_{DD}$ | PU | PU | PD | 1 | 1 | 0 | 1 | 1 | 1 | 0 | - | DCh | DDh | 6Eh |
| $V_{DD}$ | PU | PU | PU | 1 | 1 | 0 | 1 | 1 | 1 | 1 | - | DEh | DFh | 6Fh |

## 7.3 Command Code

Following the successful acknowledgement of the slave address, the bus master will send a byte to the PCA9641, which will be stored in the Command Code register.

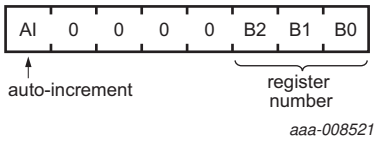



**Fig 5. Command Code**

The 3 LSBs are used as a pointer to determine which register will be accessed. If the auto-increment flag is set (AI = 1), the three 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 000b after the last allowed register is accessed (111b).
- During a write operation, the PCA9641 will acknowledge bytes sent to the CONTR, STATUS, RT, INT_STATUS, INT_MSK, MB_LO and MB_HI registers, but will not acknowledge bytes sent to the ID register since it is a read-only register. The 3 LSBs of the Command Code do not roll over to 000b but stay at 111b.

Only the 3 least significant bits are affected by the AI flag.

Unused bits must be programmed with zeros. Any command code (write operation) different from ‘AI000 0000’, ‘AI000 0001’, ‘AI000 0010’, ‘AI000 0011’, ‘AI000 0100’, ‘AI000 0101’, ‘AI000 0110’ and ‘AI000 111’ will not be acknowledged. At power-up, this register defaults to all zeros.

**Table 6. Command Code register**

| B2 | B1 | B0 | Register name | Type | Register function |
|---|---|---|---|---|---|
| 0 | 0 | 0 | ID | R only | 8-bit device ID |
| 0 | 0 | 1 | CONTR | R/W | control register |
| 0 | 1 | 0 | STATUS | R/W | status register |
| 0 | 1 | 1 | RT | R/W | reserve time |
| 1 | 0 | 0 | INT_STATUS | R/W | interrupt status register |
| 1 | 0 | 1 | INT_MSK | R/W | interrupt mask register |
| 1 | 1 | 0 | MB_LO | R/W | low 8 bits of the mail box |
| 1 | 1 | 1 | MB_HI | R/W | high 8 bits of the mail box |

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

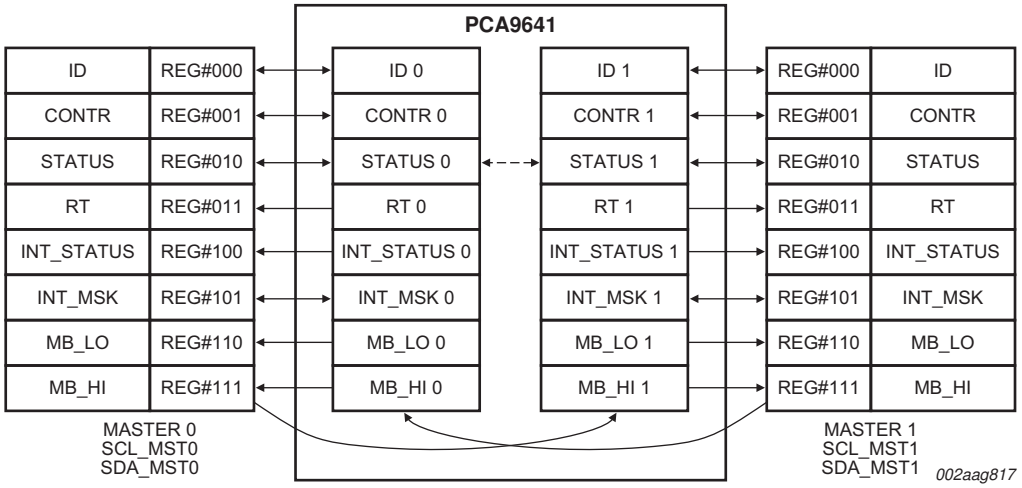



**Fig 6. Internal register map**

## 7.4 Power-on reset

When power (from 0 V) is applied to $V_{DD}$, an internal power-on reset holds the PCA9641 in a reset condition until $V_{DD}$ has reached $V_{POR}$. At that time, the reset condition is released and the PCA9641 registers and I²C-bus/SMBus state machine initialize to their default states. After that, $V_{DD}$ must be lowered to below $V_{POR}$ and back up to the operating voltage for a power-reset cycle.

## 7.5 Reset input ($\overline{\text{RESET}}$)

The $\overline{\text{RESET}}$ input can be asserted to initialize the system while keeping the $V_{DD}$ at its operating level. A reset is accomplished by holding the $\overline{\text{RESET}}$ pin LOW for a minimum of $t_{w(rst)}$. The PCA9641 registers and I²C-bus/SMBus state machine are set to their default state once $\overline{\text{RESET}}$ is LOW (0). When $\overline{\text{RESET}}$ is HIGH (1), normal operation resumes and the I²C downstream bus has no connection to any I²C-bus master.

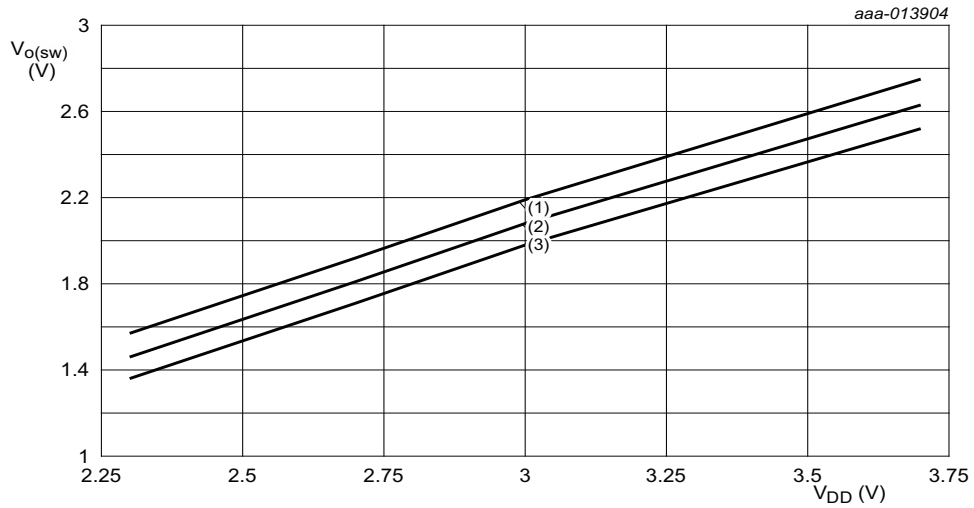
## 7.6 Software reset

When granted or non-granted master sends a software reset (see Section 13 “General call software reset”), PCA9641 will reset all internal registers and:

- If SMBUS_SWRST was enabled before software reset happens, PCA9641 sends SCL LOW for greater than 35 ms to downstream bus following a soft reset.
- If SMBUS_SWRST was disabled before software reset happens, PCA9641 does **not** send SCL LOW to downstream bus following a soft reset.

## 7.7 Voltage translation

The pass gate transistors of the PCA9641 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 7. Pass gate voltage as a function of supply voltage**

Figure 7 shows the voltage characteristics of the pass gate transistors (note that the graph was generated using the data specified in Section 18 “Static characteristics” of this data sheet). In order for the PCA9641 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 3.3 V, and the downstream bus was 2.5 V, then $V_{o(sw)}$ should be equal to or below 2.5 V to effectively clamp the downstream bus voltages. Looking at Figure 7, we see that $V_{o(sw)(max)}$ will be at 2.5 V when the PCA9641 supply voltage is 3.375 V or lower so the PCA9641 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 20).

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

# 8. Register descriptions

## 8.1 Register 0: ID register ([B2:B0] = 000b)

This register is holding the last 8 bits of the Device ID. It is used to distinguish between PCA9541 and PCA9641. When a master reads this register, if the value returned from this register is 38h, it is PCA9641, other than this value it is PCA9541.

**Table 7. ID - Device ID register (pointer address 00h) bit description**

*POR = 38h.*

| Address | Register | Bit | Access | Description |
|---|---|---|---|---|
| 00h | ID | 7:0 | R only | Hard-coded 38h for PCA9641. |

## 8.2 Register 1: Control register ([B2:B0] = 001b)

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

**Table 8. CONTR - Control register (pointer address 01h) bit description**

*POR = 00h.*

*Legend: * default value*

| Bit | Symbol | Access | Value | Description |
|---|---|---|---|---|
| 7 | PRIORITY | R/W | | Master can set this register bit for setting priority of the winner when two masters request the downstream bus at the same time. Table 9 shows how PCA9641 selects the winner when 2 masters set their own PRIORITY bit. |
| | | | 0* | Master can configure the priority bit for the case where two masters request the downstream bus at the same time. See Table 9 for information on how PCA9641 selects the winner. |
| 6 | SMBUS_DIS | R/W | | When PCA9641 detects an SMBus time-out, if this bit is set, PCA9641 will disconnect I²C-bus from master to downstream bus. |
| | | | 0* | Normal operation |
| | | | 1 | Connectivity between master and downstream bus will be disconnected upon detecting an SMBus time-out condition. |

**Table 8. CONTR - Control register (pointer address 01h) bit description** *...continued*

*POR = 00h.*

*Legend: * default value*

| Bit | Symbol | Access | Value | Description |
|---|---|---|---|---|
| 5 | IDLE_TIMER_DIS | R/W | | After RES_TIME is expired, I²C-bus idle for more than 100 ms, PCA9641 will disconnect master from downstream bus and takes away its grant if this register bit is enabled. This IDLE_TIMER_DIS function also applies when there is a grant of a request with zero value on RES_TIME. |
| | | | 0* | Normal operation. |
| | | | 1 | Enable 100 ms idle timer. After reserve timer expires or if reserve timer is disabled, if the downstream bus is idle for more than 100 ms, the connection between master and downstream bus will be disconnected. |
| 4 | SMBUS_SWRST | R/W | | Non-granted or granted master sends a soft reset, if this bit is set, PCA9641 sets clock LOW for 35 ms following reset of all register values to defaults. |
| | | | 0* | Normal operation. |
| | | | 1 | Enable sending SMBus time-out to downstream bus, after receiving a general call soft reset from master. |
| 3 | BUS_INIT | R/W | | Bus initialization for PCA9641 sends one clock out and checks SDA signal. If SDA is HIGH, PCA9641 sends a 'not acknowledge' and a STOP condition. The BUS_INIT function is completed. If SDA is LOW, PCA9641 sends other clock out and checks SDA again. The PCA9641 will send out 9 clocks (maximum), and if SDA is still LOW, PCA9641 determines the bus initialization has failed. |
| | | | 0* | Normal operation. |
| | | | 1 | Start initialization on next bus connect function to downstream bus. |
| 2 | BUS_CONNECT | R/W | | Connectivity between master and downstream bus; the internal switch connects I²C-bus from master to downstream bus only if LOCK_GRANT = 1. |
| | | | 0* | Do not connect I²C-bus from master to downstream bus. |
| | | | 1 | Connect downstream bus; the internal switch is closed only if LOCK_GRANT = 1. |
| 1 | LOCK_GRANT | R only | | This is a status read only register bit. Lock grant status register bit indicates the ownership between reading master and the downstream bus. If this register bit is 1, the reading master has owned the downstream bus. If this register bit is zero, the reading master has not owned the downstream bus. |
| | | | 0* | This master does not have a lock on the downstream bus. |
| | | | 1 | This master has a lock on the downstream bus. |

**Table 8. CONTR - Control register (pointer address 01h) bit description** *...continued*

*POR = 00h.*

*Legend: * default value*

| Bit | Symbol | Access | Value | Description |
|---|---|---|---|---|
| 0 | LOCK_REQ | R/W | | Lock request register bit is for a master requesting the downstream bus when it does not have a lock on downstream bus. When a master has a lock on downstream bus, it can give up the ownership by writing zero to LOCK_REQ register bit. When LOCK_REQ becomes zero, LOCK_GRANT bit becomes zero and the internal switch will be open. |
| | | | 0* | Master is not requesting a lock on the downstream bus or giving up the lock if master had a lock on the downstream bus. |
| | | | 1 | Master is requesting a lock on the downstream bus. |

**Table 9. How PCA9641 selects winner**

| Master 0 priority | Master 1 priority | Last master granted | Result |
|---|---|---|---|
| 0 | 0 | none | Grant is given to Master 0 |
| 0 | 0 | Master 0 | Grant is given to Master 1 |
| 0 | 0 | Master 1 | Grant is given to Master 0 |
| 0 | 1 | n/a | Grant is given to Master 1 |
| 1 | 0 | n/a | Grant is given to Master 0 |
| 1 | 1 | none | Grant is given to Master 1 |
| 1 | 1 | Master 0 | Grant is given to Master 1 |
| 1 | 1 | Master 1 | Grant is given to Master 0 |

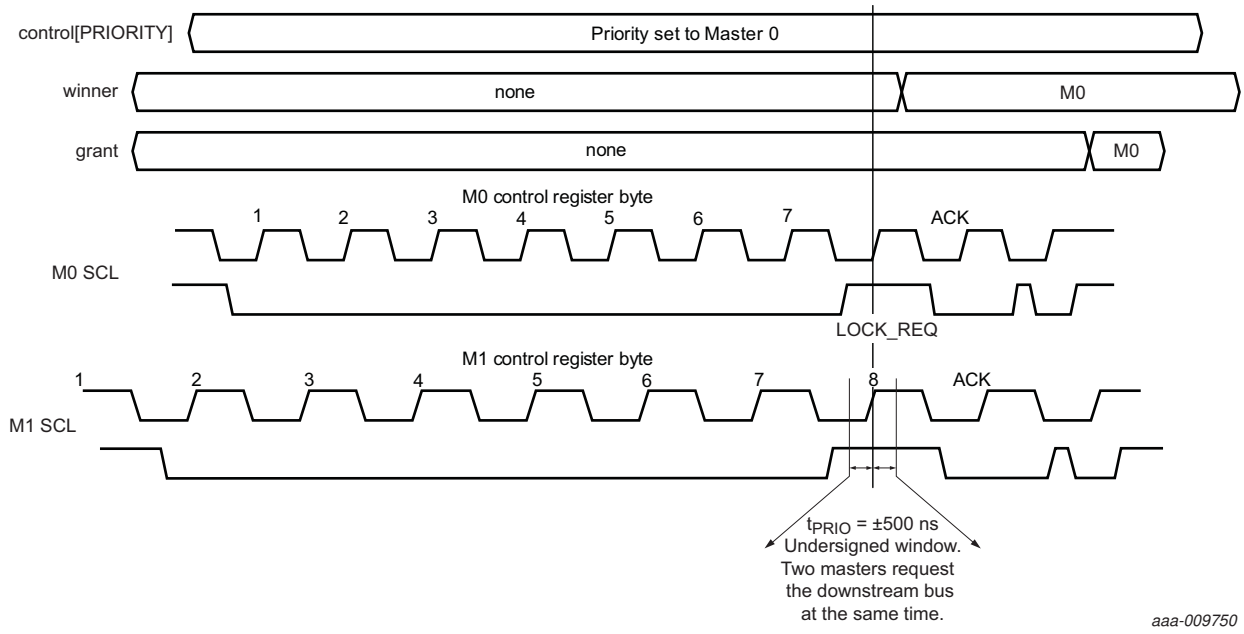



**Fig 8. Two masters request the downstream bus at the same time**

## 8.3 Register 2: Status register ([B2:B0] = 010b)

**Table 10. STATUS - Status register (pointer address 02h) bit description**

*POR = 00h.*

*Legend: * default value*

| Bit | Symbol | Access | Value | Description |
|---|---|---|---|---|
| 7 | SDA_IO | R/W | | SDA becomes I/O pin; master can read or write to this register bit. If master reads this bit, the value is the state of the downstream SDA pin. Zero value means SDA is LOW, and one means SDA pin is HIGH.<br>When master writes '0' to this register bit, the downstream SDA pin will assert LOW.<br>If master writes '1' to this register bit, the downstream SDA pin will be pulled HIGH.<br>**Remark:** SDA becomes I/O pin only when BUS_CONNECT = 0 and LOCK_GRANT = 1. |
| | | | 0* | When read, indicates the SDA pin of the downstream bus is LOW.<br>When written, PCA9641 drives SDA pin of downstream bus LOW. |
| | | | 1 | When read, indicates the SDA pin of the downstream bus is HIGH.<br>When written, PCA9641 drives SDA pin of the downstream bus HIGH. |
| 6 | SCL_IO | R/W | | SCL becomes I/O pin; master can read or write to this register bit. If master reads this bit, the value is the state of the downstream SCL pin. Zero value means SCL is LOW, and one means SCL pin is HIGH.<br>When master writes '0' to this register bit, the downstream SCL pin will assert LOW.<br>If master writes '1' to this register bit, the downstream SCL pin will be pulled HIGH.<br>**Remark:** SCL becomes I/O pin only when BUS_CONNECT = 0 and LOCK_GRANT = 1. |
| | | | 0* | When read, shows the SCL pin of the downstream bus is LOW.<br>When written, PCA9641 drives SCL pin of downstream bus LOW. |
| | | | 1 | When read, shows the SCL pin of the downstream bus is HIGH.<br>When written, PCA9641 drives SCL pin of the downstream bus HIGH. |

**Table 10. STATUS - Status register (pointer address 02h) bit description** *…continued*

*POR = 00h.*

*Legend: * default value*

| Bit | Symbol | Access | Value | Description |
|---|---|---|---|---|
| 5 | TEST_INT | W only | | Test interrupt output pin; a master can send an interrupt to itself by writing '1' to this register bit. Writing '0' to this register bit has no effect. To clear this interrupt, master must write '1' to TEST_INT_INT in Interrupt Status register. |
| | | | 0* | Normal operation |
| | | | 1 | Causes PCA9641 $\overline{\text{INT}}$ pin to go LOW if not masked by TEST_INT_INT in Interrupt Mask register. Allows this master to invoke its Interrupt Service Routine to handle housekeeping tasks. |
| 4 | MBOX_FULL | R only | | This is a read-only status register bit. If this bit is '0', it indicates no data is available in the mail box. If it is '1', it indicates new data is available in the mail box. |
| | | | 0* | No data is available for **this** master. |
| | | | 1 | Mailbox contains data for **this** master from the other master. |
| 3 | MBOX_EMPTY | R only | | This is a read-only status register bit. If this bit is '0', it indicates other master mailbox is full, and this master cannot send more data to other master mailbox. If it is '1', it indicates other master is empty and this master can send data to other master mailbox. |
| | | | 0* | **Other** master mailbox is full; wait until **other** master reads data. |
| | | | 1 | **Other** master mailbox is empty. **Other** master has read previous data and it is permitted to write new data. |
| 2 | BUS_HUNG | R only | | This is a read-only status register bit. If this register bit is '0', it indicates the bus is in normal condition. If this bit is '1', it indicates the bus is hung. The hung bus means SDA signal is LOW and SCL signal does not toggle for more than 500 ms or SCL is LOW for 500 ms. |
| | | | 0* | Normal operation |
| | | | 1 | Downstream bus hung; when SDA signal is LOW and SCL signal does not toggle for more than 500 ms or SCL is LOW for 500 ms. |
| 1 | BUS_INIT_FAIL | R only | | This is a read-only status register bit. If this register bit is '0', it indicates the bus initialization function has passed. The downstream bus is in idle mode (SCL and SDA are HIGH). If this register bit is '1', it indicates the bus initialization function has failed. The SDA signal could be stuck LOW. |
| | | | 0* | Normal operation |
| | | | 1 | Bus initialization has failed. SDA still LOW, the downstream bus cannot recover. |

**Table 10. STATUS - Status register (pointer address 02h) bit description** *…continued*

*POR = 00h.*

*Legend: * default value*

| Bit | Symbol | Access | Value | Description |
|---|---|---|---|---|
| 0 | OTHER_LOCK | R only | | This is a status read-only register bit. Other master lock status indicates the ownership between other master and the downstream bus. If this register bit is '1', the other master has owned the downstream bus. If this register bit is '0', the other master does not own the downstream bus. |
| | | | 0* | The other master does not have a lock on the downstream bus. |
| | | | 1 | The other master has a lock on the downstream bus. |

## 8.4 Register 3: Reserve Time register ([B2:B0] = 011b)

Reserve time is for when a master wants ownership of the downstream bus without interruption. It can reserve from 1 ms to 255 ms ownership of the downstream bus without interruption.

**Table 11. RT - Reserve Time register (pointer address 03h) bit description**

*POR = 00h.*

| Bit | Symbol | Access | Value | Description |
|---|---|---|---|---|
| 7 to 0 | RES_TIME[7:0] | R/W | | Reserve timer. Changes during LOCK_GRANT = 1 will have no effect. |
| | | | 0 | Disable timer or reserve without time limited. |
| | | | 01h | 1 ms |
| | | | : | : |
| | | | FFh | 255 ms |

Reserve time cannot be changed after LOCK_GRANT is one.

If a master requests the downstream bus with 00h in Reserve Time register, this master wants the downstream bus forever or until it gives up the bus by setting LOCK_REQ bit to zero.

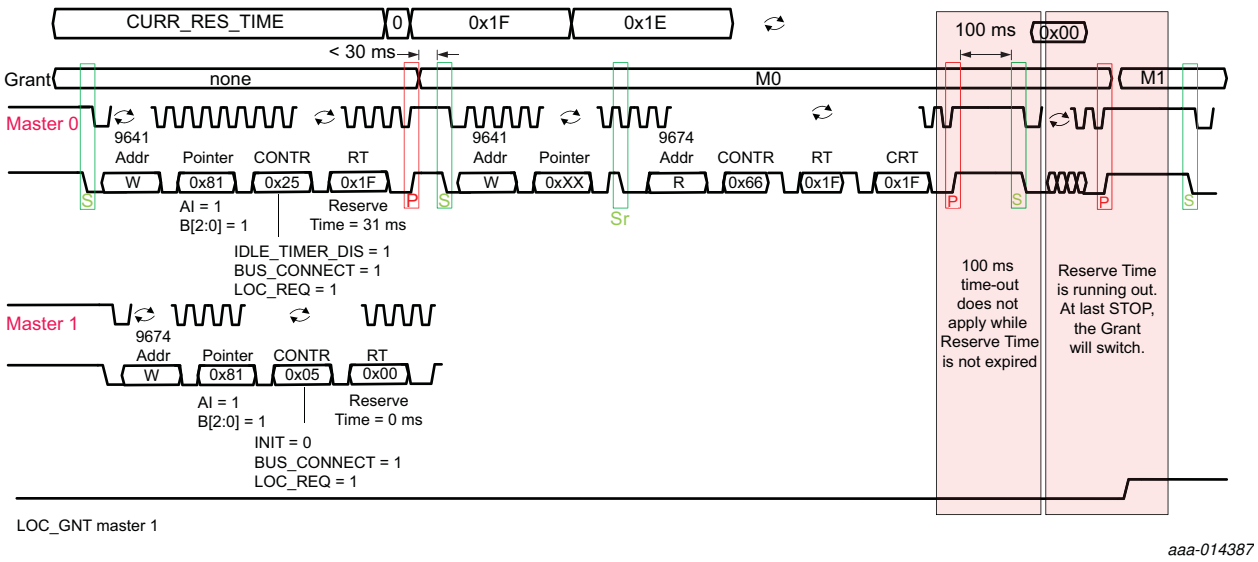



**Fig 9. Request downstream with reserve time**

## 8.5 Register 4: Interrupt Status register ([B2:B0] = 100b)

These interrupt status bits are sticky and will remain set until cleared by writing '1'.

The PCA9641 provides seven different types of interrupt.

**Table 12. INT_STATUS - Interrupt status register (pointer address 04h) bit description**
*POR = 00h.*

| Bit | Symbol | Access | Value | Description |
|---|---|---|---|---|
| 7 | - | | | Reserved. |
| 6 | BUS_HUNG_INT | R only | | Indicates to both masters that SDA signal is LOW and SCL signal does not toggle for more than 500 ms or SCL is LOW for 500 ms. |
| | | | 0 | No interrupt generated; normal operation. |
| | | | 1 | Interrupt generated; downstream bus cannot recover; when SDA signal is LOW and SCL signal does not toggle for more than 500 ms or SCL is LOW for 500 ms, |
| 5 | MBOX_FULL_INT | R/W | | Indicates the mailbox has new mail. |
| | | | 0 | No interrupt generated; mailbox is not full. |
| | | | 1 | Interrupt generated; mailbox full. |
| 4 | MBOX_EMPTY_INT | R/W | | Indicates the sent mail is empty, other master has read the mail. |
| | | | 0 | No interrupt generated; sent mail is not empty. |
| | | | 1 | Interrupt generated; mailbox is empty. |
| 3 | TEST_INT_INT | R/W | | Indicates this master has sent an interrupt to itself. |
| | | | 0 | No interrupt generated; master has not set the TEST_INT bit in STATUS register. |
| | | | 1 | Interrupt generated; master activates its interrupt pin via the TEST_INT bit in STATUS register. |
| 2 | LOCK_GRANT_INT | R/W | | Indicates the master has a lock (ownership) on the downstream bus. |
| | | | 0 | No interrupt generated; this master does not have a lock on the downstream bus. |
| | | | 1 | Interrupt generated; this master has a lock on the downstream bus. |
| 1 | BUS_LOST_INT | R/W | | Indicates the master has involuntarily lost the ownership of the downstream bus. |
| | | | 0 | No interrupt generated; this master is controlling the downstream bus. |
| | | | 1 | Interrupt generated; this master has involuntarily lost the control of the downstream bus. |
| 0 | INT_IN_INT | R/W | | Indicates that there is an interrupt from the downstream bus to both the granted and non-granted masters. |
| | | | 0 | No interrupt on interrupt input pin $\overline{\text{INT\_IN}}$. |
| | | | 1 | Interrupt on interrupt input pin $\overline{\text{INT\_IN}}$. |

## 8.6 Register 5: Interrupt Mask register ([B2:B0] = 101b)

**Table 13. INT_MSK - Interrupt Mask register (pointer address 05h) bit description**
*POR = 7Fh.*

| Bit | Symbol | Access | Value | Description |
|---|---|---|---|---|
| 7 | - | | | Reserved. |
| 6 | BUS_HUNG_MSK | R/W | 0 | Enable output interrupt when BUS_HUNG function is set. |
| | | | 1 | Disable output interrupt when BUS_HUNG function is set. |
| 5 | MBOX_FULL_MSK | R/W | 0 | Enable output interrupt when MBOX_FULL function is set. |
| | | | 1 | Disable output interrupt when MBOX_FULL function is set. |
| 4 | MBOX_EMPTY_MSK | R/W | 0 | Enable output interrupt when MBOX_EMPTY function is set. |
| | | | 1 | Disable output interrupt when MBOX_EMPTY function is set. |
| 3 | TEST_INT_MSK | R/W | 0 | Enable output interrupt when TEST_INT function is set. |
| | | | 1 | Disable output interrupt when TEST_INT function is set. |
| 2 | LOCK_GRANT_MSK | R/W | 0 | Enable output interrupt when LOCK_GRANT function is set. |
| | | | 1 | Disable output interrupt when LOCK_GRANT function is set. |
| 1 | BUS_LOST_MSK | R/W | 0 | Enable output interrupt when BUS_LOST function is set. |
| | | | 1 | Disable output interrupt when BUS_LOST function is set. |
| 0 | INT_IN_MSK | R/W | 0 | Enable output interrupt when INT_IN function is set. |
| | | | 1 | Disable output interrupt when INT_IN function is set. |

## 8.7 Registers 6 and 7: MB registers ([B2:B0] = 110b and 111b)

**Table 14. SMB - Shared Mail Box registers (pointer addresses 06h, 07h) bit description**
*POR = 00h.*

| Address | Bit | Symbol | Access | Description |
|---|---|---|---|---|
| 06h | 7 to 0 | MB_LO[7:0] | R/W | Low 8 bits of the mail box. |
| 07h | 7 to 0 | MB_HI[7:0] | R/W | High 8 bits of the mail box. |

## 8.8 Operating cycle of the downstream bus

### 8.8.1 Request the downstream bus

When a master seeks control of the bus by requesting its I²C-bus channel to the PCA9641 registers, it must write to the Control register (CONTR, 01h) and Reserve Time register (RT, 03h) optional. LOCK_REQ bit and RT[7:0] allow the master to take control of the bus in a period of RES_TIME without interrupting.

While master 0 is working on the downstream bus, master 1 can request the downstream bus by writing to LOCK_REQ bit in CONTR register and RT register. When the downstream bus is free and RES_TIME is expired, master 1 will have control of the downstream bus.

If Reserve Time is set to 0, it will disable the timer counter. That means the master requests the downstream bus forever or until it gives up the bus.

### 8.8.2 Acquire the downstream bus

After the master wrote to LOCK_REQ bit and RT register, it must poll LOCK_GRANT bit in CONTR register or wait for interrupt signal (INTx pin) if LOCK_GRANT_MSK bit is set in INT_MSK register for the ownership of the downstream bus.

When LOCK_GRANT bit is one, this master has full control of the downstream bus.

To start communication with downstream slave devices, master must connect to downstream bus by setting BUS_CONNECT = 1.

### 8.8.3 Give up the downstream bus

The RES_TIME starts countdown after LOCK_GRANT becomes one. When the RES_TIME becomes zero and the I²C-bus is free (SCL_SLAVE and SDA_SLAVE are HIGH) after STOP condition, PCA9641 will clear the LOCK_GRANT bit.

If a master requests the downstream bus with RES_TIME = 0, it must write zero to LOCK_REQ bit to give up its control.

# 9. Arbitration

## 9.1 Rules

1. If a master keeps its request asserted after its grant, the master will indefinitely hold the bus.
   - If the bus goes IDLE for 100 ms, it will be disconnected only if the idle time-out function is enabled and the reserve timer has expired.
2. If a master removes its request, then that master will lose its grant.
   - If the other master is requesting the bus, it will be granted.
   - If no master is requesting the bus, PCA9641 will disconnect from both.
3. If a master sets the reserve timer before its grant, the timer will clear its request when it expires.
   - This timer gives a 1 ms to 255 ms window for locking the bus. When the timer expires, it clears the master's request and follows Step 2.
   - If the bus is idle for 100 ms and the reserve timer has not expired, the grant will **not** be lost.
   - If the master clears its request and the reserve timer has not expired, the grant will be lost.
4. If both masters request the grant at the same time (close), the winner will be determined as follows:
   - The first master to set the request bit in the register wins. START does not matter, and nothing else really matters as the masters might have different clock frequencies, etc. The master might be doing a burst write with an address rollover, making the control register the last byte it writes. However, if the bit is set in the control register first, it wins.
   - The action of the grant is applied when the winning master's transaction is terminated with a STOP. (It is not OK to do a Re-START when requesting the bus; before accessing the downstream slaves, master must issue a STOP.)
   - If both masters request at the exact same time, and logic cannot determine a winner, the control register priority bit determines which master to give the grant to. See Section 8.2.
5. A write to the control register for a REQUEST will always be answered with an ACK.
   - The master must poll the control register or use the interrupts to determine when the grant is awarded.

## 9.2 Disconnect events

The following events cause a master to disconnect condition to occur, assuming the conditions from the previous section are satisfied to allow the grant to be removed and the downstream bus to be disconnected.

1. STOP (ideal, this is the cleanest way).
2. Bus IDLE for 100 ms (not ideal).
3. Writing 0 to LOCK_REQ.

## 10. State machines

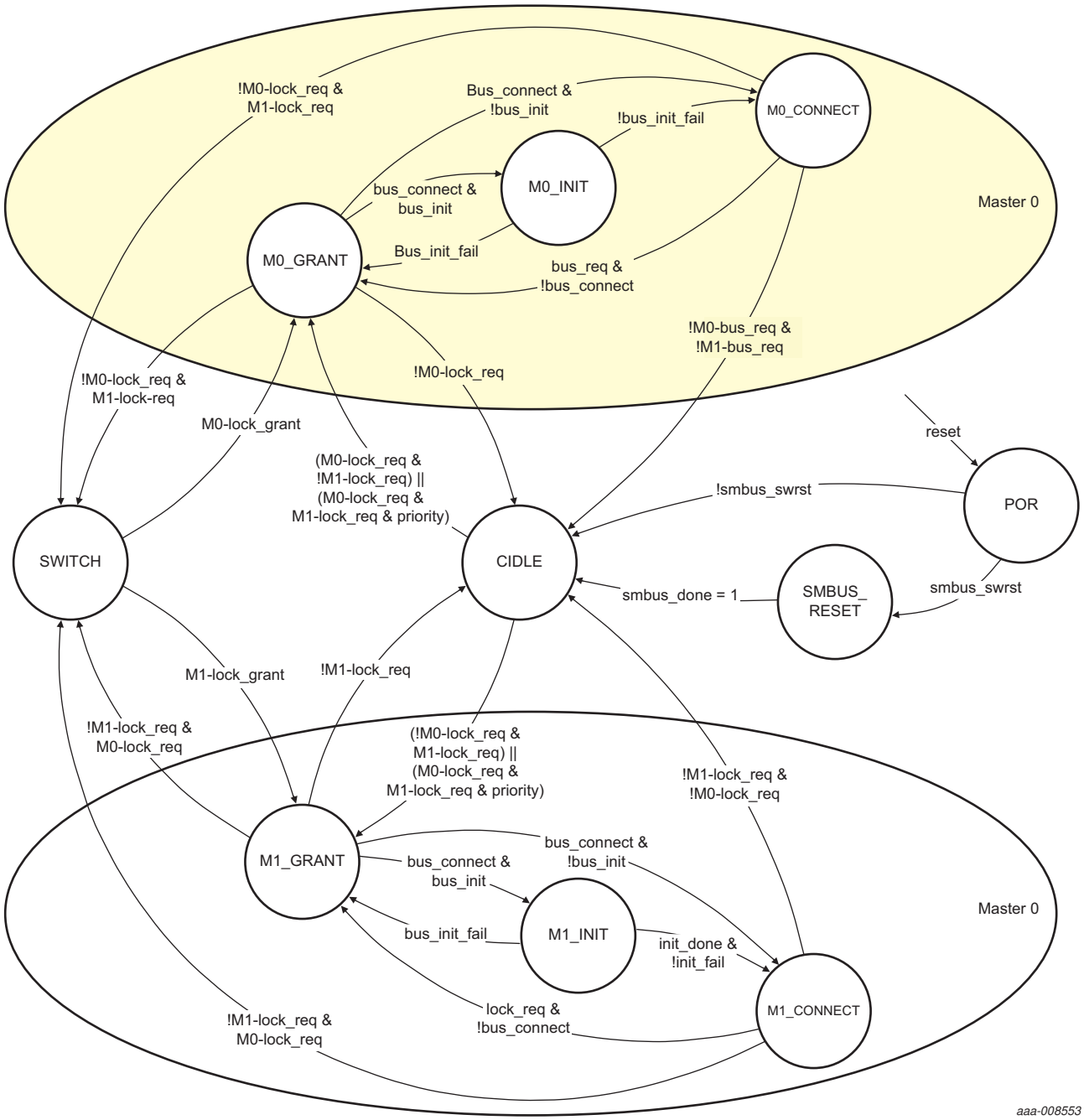



**Fig 10. State machine of downstream bus ownership**

## 11. Request grant examples

In the waveform shown in Figure 11, Master 0 initiated a START first. Master 1 was at a higher clock speed and wrote the request bit first, so Master 1 won the arbitration.

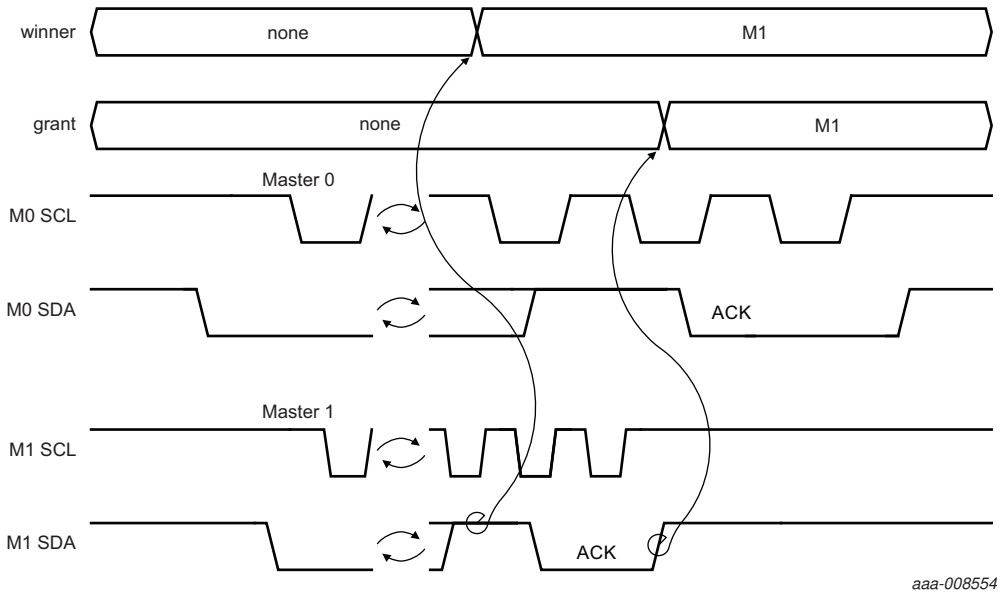



**Fig 11. Request grant example**

The effects of the arbitration do not take effect until the winning master issues a STOP condition. If the winning master were to continue to write to the next register using auto-incrementing addresses, it would delay the grant until the STOP. The master has 'won' the arbitration, though.

Two masters request the bus at the same time (close enough that the logic cannot tell the difference). See Figure 8 for the waveform. In this case the PRIORITY bits are used to determine the winner. The truth table, Table 9, is for winner selection.

## 12. Characteristics of the I²C-bus

The information in this section pertains to both M0 and M1 I²C-bus interfaces.

The I²C-bus interface is used to access the device programmable registers. This interface runs as Fast-mode Plus (Fm+) speeds with a general call software reset. The I²C core is composed of the I²C State Machine, shift register, and the start/stop detection logic.

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.