

AS3955

NFC Forum Compliant Dynamic Tag

General Description

AS3955 NFC Dynamic Tag IC is the ultimate solution to easily add NFC functionality to electronic devices. Thanks to a high sensitivity ISO14443A frontend and high integrated resonance capacitor, AS3955 offers standalone NFC passive tag functionality in a small footprint. Fast system integration and high speed data transfer are guaranteed by the available SPI and I²C interfaces and by the optimized protocols (Tunneling Mode and Extended Mode), allowing bidirectional communication between the device microcontroller and an external NFC compliant device or ISO14443A reader device (PCD).

AS3955 is able to operate fully powered by the RF field, without any external supply. This, combined with an advanced energy harvesting feature, greatly increases battery life time or even allows battery-less designs.

AS3955 is used with an appropriate antenna coil connected to the terminals LC1 and LC2, and behaves as a standard passive ISO 14443A tag (PICC). After the anti-collision protocol stage, based on configuration, AS3955 can operate as a standalone NFC Forum Type 2 Tag or, when tunneling mode is activated, as a bridge between the PCD and the microcontroller, e.g. to emulate a custom or standard ISO14443A Level 3 or Level 4 PICC or a NFC Forum tag. A configurable wake-up signal notifies the microcontroller on ongoing RF activities, in order to minimize overall power consumption.

AS3955 includes an embedded EEPROM memory that can be accessed from the PCD through the RF link or from the microcontroller through the slave SPI or I²C interfaces. Part or all memory can be protected by a 32-bit password, or permanently locked.

AS3955 supports ISO 14443A up to Level 4 and is designed according to EMVCo requirements, to enable the emulation of contactless smart cards or NFC Forum compliant Type 2 or Type 4 Tags.

AS3955 is designed for high reliability, and can operate in a wide power supply range from 1.65V to 5.5V, in a wide temperature range from -40°C to 125°C. EEPROM memory reaches automotive grade quality with endurance of 100000 cycles and data retention of 10 years at 125°C.

Ordering Information and Content Guide appear at end of datasheet.

Key Benefits & Features

The benefits and features of AS3955, NFC Forum Compliant Dynamic Tag are listed below:

Figure 1:
Added Value of Using AS3955

| Benefits | Features |
|---|--|
| <ul style="list-style-type: none"> NFC Forum compliance for full interoperability | <ul style="list-style-type: none"> Type 2 Tag standalone functionality Type 2 or 4 Tag emulation with external MCU ISO 14443A Level 3 or 4 card emulation with external MCU |
| <ul style="list-style-type: none"> NFC Forum compliance ISO 14443A compliance up to Level 4 | <ul style="list-style-type: none"> Operating frequency at 13.56 MHz Bit rate at 106 kbps 7-byte UID |
| <ul style="list-style-type: none"> Choice of memory size based on application | <ul style="list-style-type: none"> 4 kbit EEPROM (472 bytes of user data) |
| <ul style="list-style-type: none"> Allows zero-power standby | <ul style="list-style-type: none"> Configurable passive wake-up interrupt |
| <ul style="list-style-type: none"> Enables long battery life time, or battery-less designs | <ul style="list-style-type: none"> Energy harvesting to supply up to 5mA @ 4.5V (regulated) |
| <ul style="list-style-type: none"> Allows fast antenna prototyping (ISO antenna classes 1 to 6) | <ul style="list-style-type: none"> 45 pF integrated resonant capacitor |
| <ul style="list-style-type: none"> Design flexibility, easy integration. Fits requirements for various embedded applications | <ul style="list-style-type: none"> 3/4-wire SPI slave interface up to 5 Mbps I²C slave interface up to 1 Mbps |
| <ul style="list-style-type: none"> Design flexibility, easy integration | <ul style="list-style-type: none"> Programmable I²C address |
| <ul style="list-style-type: none"> Fits supply requirements for various applications, including industrial | <ul style="list-style-type: none"> Wide interface supply range (1.65V to 5.5V) |
| <ul style="list-style-type: none"> Support for multiple applications, and storage of sensitive data | <ul style="list-style-type: none"> 32-bit password memory protection |
| <ul style="list-style-type: none"> High performance and robust data communication, allows custom protocols to be implemented | <ul style="list-style-type: none"> Tunneling and Extended modes for MCU communication |
| <ul style="list-style-type: none"> Consistent NFC behavior of battery supplied devices in e.g. pairing applications | <ul style="list-style-type: none"> Silent mode (MCU power status detection), configurable between 1.42V and 3.65V |
| <ul style="list-style-type: none"> Possibility to disable RF communication | <ul style="list-style-type: none"> Configurable Chip Kill mode |
| <ul style="list-style-type: none"> Small outline, compatibility to common inlay and card manufacturing lines, surface-mount assembly | <ul style="list-style-type: none"> Sawn wafer WL-CSP package (2.125mm x 1.42 mm) @ 0.4mm pitch thick 600µm, thin 316µm typ 10-pin MLPD 3mm x3mm x0.9mm package |

Applications

AS3955 is suited to a wide range of applications, including

- NFC connection handover (Bluetooth™, Bluetooth Low Energy, Wi-Fi pairing)
- Equipment setup, service and configuration
- Firmware upgrades
- Activity and status logging
- Wireless authentication (e.g. access control to buildings and equipment)

Typical Markets:

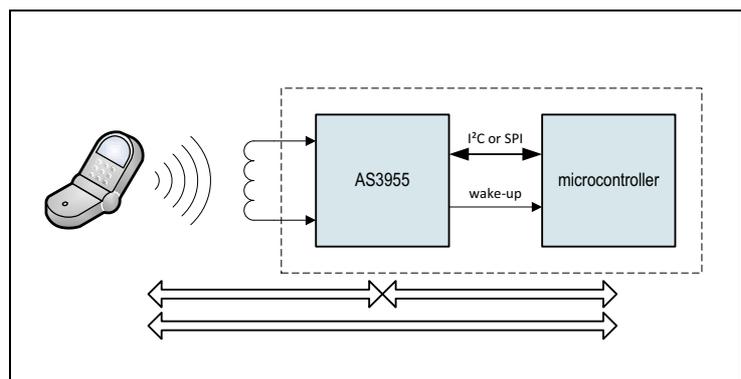
- Consumer electronics, wearables and smart clothing
- Home appliances
- Automotive
- Industrial equipment and building automation
- Remote sensing
- Gaming
- EMV payment cards

A typical system diagram is depicted in [Figure 2](#).

At the presence of a 13.56 MHz field generated by a NFC device, AS3955 powers up, notifies the microcontroller through a wake-up signal and handles the tag activation sequence. Depending on configuration, several operations are then possible:

- AS3955 exchanges with the NFC device NDEF data stored in the internal EEPROM
- The microcontroller exchanges with the NFC device NDEF data stored in external memory
- The microcontroller exchanges data with AS3955. This operation can be performed concurrently with communication over the RF link, or also in absence of RF power, in presence of an external supply.

Figure 2:
Typical System Diagram



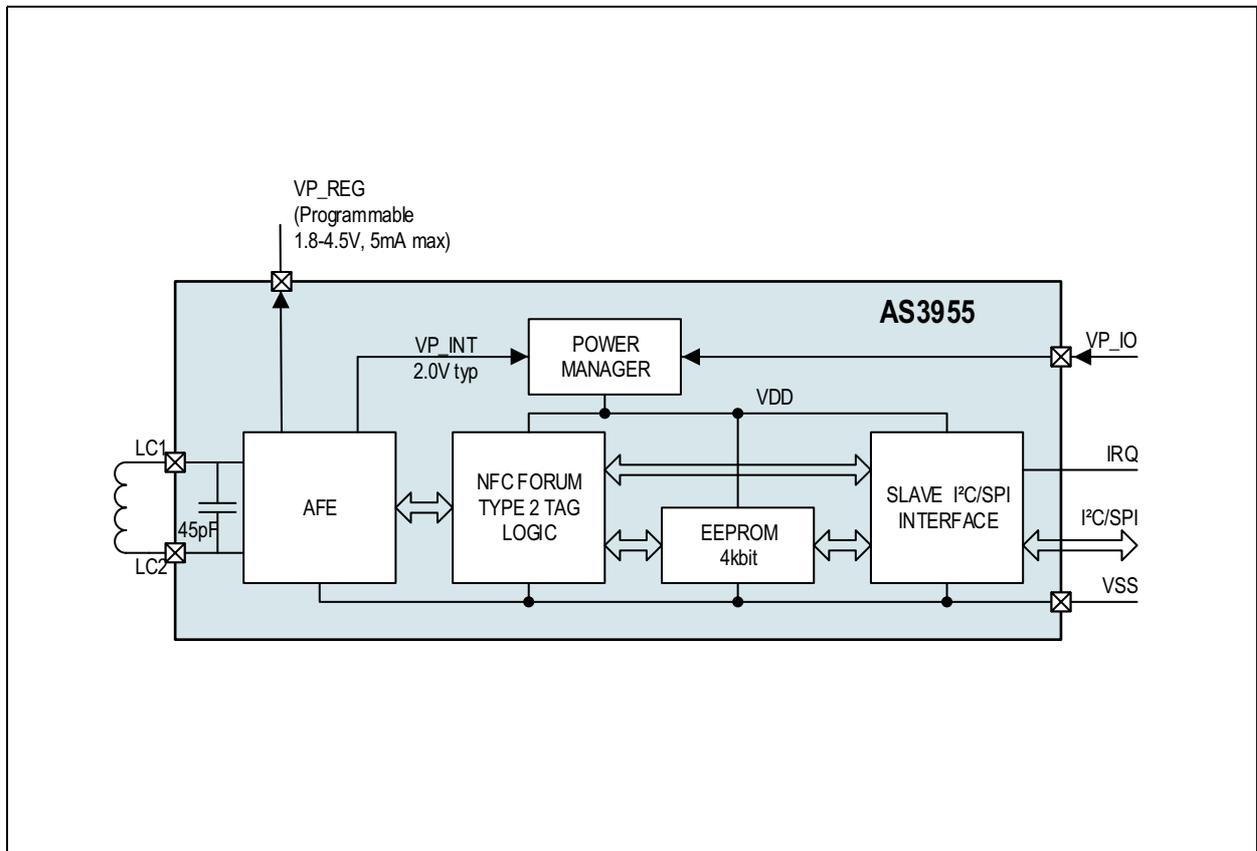
This built-in flexibility makes it ideal for a number of applications requiring non-volatile memory to be accessed when the system is not powered, e.g.:

- personalization data is programmed by the NFC device (even in case SPI / I²C is not powered) and it is later read by microcontroller through SPI / I²C interface
- Log data is stored periodically by the microcontroller and can then be read by the NFC device even when the microcontroller is not powered
- A NDEF message is regularly modified by the microcontroller (e.g. Bluetooth pin code, or Wi-Fi key, or dynamic URL) and it is later read by a NFC device.

Block Diagram

The functional blocks of this device are shown below:

Figure 3:
AS3955 Block Diagram



AS3955 is composed of NFC-A Analog Front End (AFE), NFC Type 2 Tag Logic, EEPROM, SPI / I²C Interface and Power Supply Manager Block (Power Manager).

The AFE is connected to an external tag coil which forms, together with integrated resonant capacitor, a LC tank resonating with the external electromagnetic field frequency of 13.56 MHz. The AFE has built-in rectifier and regulators. The output of the internal regulator (VP_INT) is used to supply the AFE and also the Logic and EEPROM (through Power Supply Manager). A regulator output VP_REG is available on a pin to supply external circuitry by harvesting energy from the RF field.

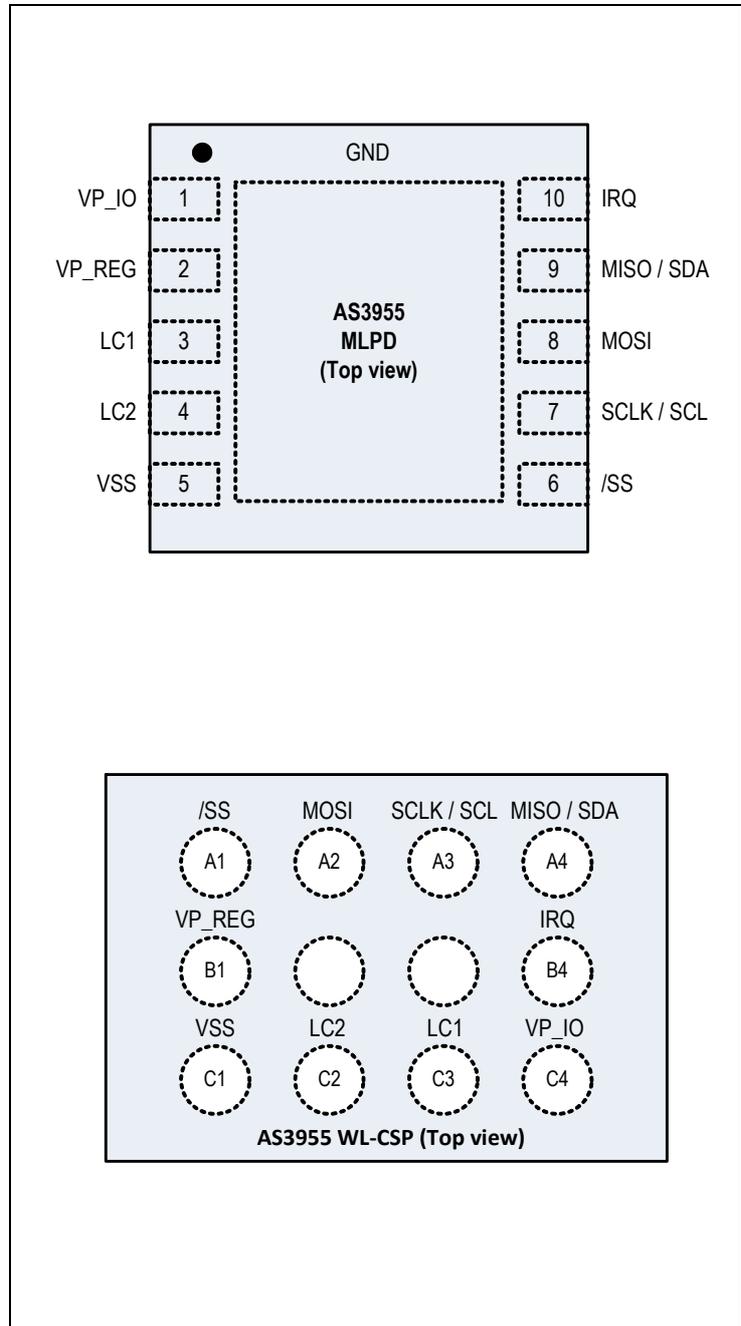
The Power Manager is controlling the power supply of Logic and EEPROM. The two blocks can be supplied either from VP_INT or from VP_IO (SPI / I²C power supply) depending on the power mode of the chip. AS3955 offers a power mode where VP_IO supply is switched to VP_INT whenever the RF field is present. VP_IO is typically used when some activity is started over the SPI / I²C and the VP_INT is too low to be used as a power supply.

The Logic is responsible for handling the anti-collision sequence, when acting as NFC Type 2 Tag, and other data transfer. The interface logic contains also a 32-byte buffer for block transmission between NFC device and AS3955.

The EEPROM is used to store the UID, configuration and control bits, and user data which can be accessed also via the SPI / I²C.

Pin Assignment

Figure 4:
AS3955 Pin Diagram



Pin Description

Figure 5:
Pin Description

| 10-pin MLPD | 10-pin WL-CSP | Die | Pin Name | Pin Type | Description |
|-------------|---------------|-----|------------|---------------------------|---|
| NA | NA | 1 | meas | Analog I/O | Analog test pin ⁽¹⁾ |
| 1 | C4 | 2 | VP_IO | Supply Pad | Positive supply of the interface / IC |
| 2 | B1 | 3 | VP_REG | Analog Output | Regulator output |
| 3 | C3 | 4 | LC1 | Analog I/O | Connection to tag coil |
| 4 | C2 | 5 | LC2 | | Connection to tag coil |
| 5 | C1 | 6 | VSS | Supply Pad | Ground, die substrate potential |
| 6 | A1 | 9 | /SS | Digital Input | SPI enable (active low) / I ² C interface enable |
| 7 | A3 | 10 | SCLK / SCL | | SPI / I ² C clock |
| 8 | A2 | 11 | MOSI | | SPI data input |
| 9 | A4 | 12 | MISO / SDA | Digital Output / Tristate | SPI data output / I ² C data line |
| 10 | B4 | 13 | IRQ | Digital Output | Interrupt request output (active high) |

Note(s):

1. Pin *meas* is not bonded in MLPD package. It is only used during wafer sort test.

Absolute Maximum Ratings

Stresses beyond those listed under [Absolute Maximum Ratings](#) may cause permanent damage to the device. These are stress ratings only. Functional operation of the device at these or any other conditions beyond those indicated under [Operating Conditions](#) is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Figure 6:
Absolute Maximum Ratings

| Symbol | Parameter | Min | Max | Unit | Comments |
|--------------------------------|--|------|-----|------|--|
| Electrical Parameters | | | | | |
| VDD | DC supply voltage | -0.5 | 6.5 | V | |
| V_{in} | Input pin voltage except LC1 and LC2 | -0.5 | 6.5 | V | |
| | Input pin voltage pins LC1 and LC2 | -0.5 | 6.5 | V | |
| | Peak current induced on pins LC1 and LC2 | | 100 | mA | |
| I_{scr} | Input current (latch-up immunity) | -100 | 100 | mA | Norm: Jedec 78 |
| Electrostatic Discharge | | | | | |
| ESD _{HBM} | Electrostatic discharge HBM | ±2 | | kV | Norm: MIL 883 E method 3015 (Human Body Model) |
| ESD _{CDM} | ESD – machine and charged device models | ±500 | | V | |

| Symbol | Parameter | Min | Max | Unit | Comments |
|--|--|-----|-----|--------|---|
| Temperature Ranges and Storage Conditions | | | | | |
| T_{strg} | Storage temperature | -55 | 150 | °C | |
| T_{body} | Package body temperature | | 260 | °C | Norm: IPC/JEDEC J-STD-020. The reflow peak soldering temperature (body temperature) is specified according IPC/JEDEC J-STD-020 "Moisture/Reflow Sensitivity Classification for Non-hermetic Solid State Surface Mount Devices." The lead finish for Pb-free leaded packages is matte tin (100% Sn). |
| RH_{NC} | Relative humidity non-condensing | 5 | 85 | % | |
| MSL | Moisture sensitivity level for MLPD package | 3 | | | Represents a max. floor life time of 168h |
| | Moisture sensitivity level for WL-CSP and thin WL-CSP | 1 | | | Represents an unlimited floor life time |
| t_{STRG_DOF} | Storage time for DOF/dies or wafers on foil | | 3 | months | Refers to indicated date of packing |
| T_{STRG_DOF} | Storage temperature for DOF/dies or wafers on foil | 17 | 28 | °C | |
| RH_{OPEN_DOF} | Relative humidity for DOF/dies or wafers on foil in open package | | 15 | % | Opened package |
| RH_{UNOPEN_DOF} | Relative humidity for DOF/dies or wafers on foil in sealed package | 40 | 60 | % | Sealed bag |

Electrical Characteristics

All limits are guaranteed. The parameters with min and max values are guaranteed with production tests or SQC (Statistical Quality Control) methods.

Operating Conditions

All in this specification defined tolerances for external components need to be assured over the whole operation conditions range and also over lifetime.

Figure 7:
Electrical Characteristics

| Symbol | Parameter | Min | Max | Unit | Notes |
|--------------|----------------------|------|-----|------|--------------------------------------|
| I_{lim} | Limiter current | | 30 | mA | |
| V_{VP_IO} | SPI power supply | 1.65 | 5.5 | V | When logic powered from RF interface |
| V_{VP_IO} | SPI power supply | 1.65 | 5.5 | V | When logic powered from VP_IO |
| T_{junc} | Junction temperature | -40 | 125 | °C | |

DC/AC Characteristics for Digital Inputs and Outputs

Figure 8:
CMOS Inputs

| Symbol | Parameter | Min | Typ | Max | Unit | Note |
|------------|--------------------------|--------------------|-----|--------------------|------|--------|
| V_{IH} | High level input voltage | $0.7 * V_{DD_IO}$ | | | V | |
| V_{IL} | Low level input voltage | | | $0.3 * V_{DD_IO}$ | V | |
| I_{LEAK} | Input leakage current | | | 10 | µA | @125°C |

Note(s):

- Valid for input pins /SS, MOSI and SCLK

Figure 9:
CMOS Outputs

| Symbol | Parameter | Min | Typ | Max | Unit | Note |
|--------|-------------------------------|--------------------------|-----|--------------------------|------|---|
| VOH | High level output voltage | 0.85 * VDD _{IO} | | | V | I _{source} =1mA VP _{IO} = 5V |
| VOL | Low level output voltage | | | 0.15 * VDD _{IO} | V | |
| RO | Output resistance | | 200 | 400 | Ω | |
| RPD | Pull-down resistance pad MISO | | 10 | | kΩ | See note (1) |

Note(s):Valid for output pins MISO and IRQ

1. Pull down can be enabled while MISO output is in tristate. The activation is controlled by register setting.

Electrical Specifications

Figure 10:
Electrical Specifications

| Symbol | Parameter | Min | Typ | Max | Unit | Note |
|---------------------|--|---------|------------|-----|--------|---|
| I _{SB_SPI} | Stand by consumption on VP _{IO} | | 65 | 100 | nA | @ 25°C (RF field not applied) |
| V _{LIM} | Limiter voltage | | 5.2 | 5.5 | V | I _{LC} =30mA (DC) ⁽²⁾ |
| I _S | Supply current | | 350 | | μA | See note (3) |
| V _{HF_PON} | HF_PON threshold (rising VREG) | | 1.6 2.3 | | V | See note (4) |
| V _{POR_HY} | HF_PON hysteresis | | 0.8 | | V | |
| V _{MOD} | Modulator ON voltage drop | | 1.2 2.6 | | V | I _{LC} =1mA ⁽²⁾ I _{LC} =30mA ⁽²⁾ |
| C _{RES} | Resonance capacitor | | 45 | | pF | |
| EE _{EN} | EEPROM endurance | 100 000 | | | cycles | @ 125°C ⁽⁵⁾ |
| EE _{RET} | EEPROM retention | 10 | | | years | @ 125°C ⁽⁶⁾ |

Note(s):

1. VP_{IO}=1.8V, Temperature 25°C unless noted otherwise.

2. I_{LC} is the current flowing through LC1 and LC2 pins

3. Internal supply current measured over VP_{IO} pin, by forcing internal digital supply to 2.0V, and applying 13.56 MHz alternative pulses with amplitude 3.0Vpp to LC1 and LC2.

4. 1.6V is set in [Power Mode 2](#) only.

5. See [Figure 11](#).

6. See Figure 12.

Figure 11:
EEPROM Endurance Over Temperature

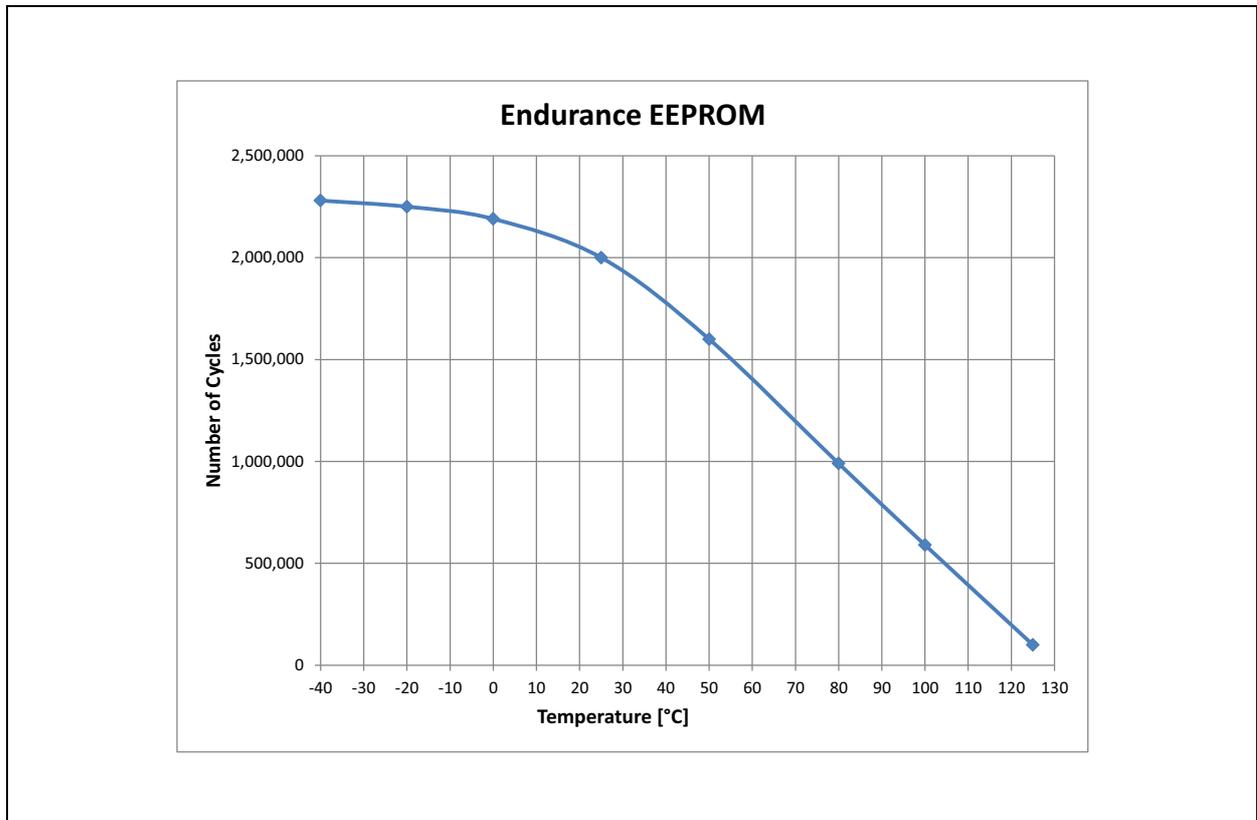
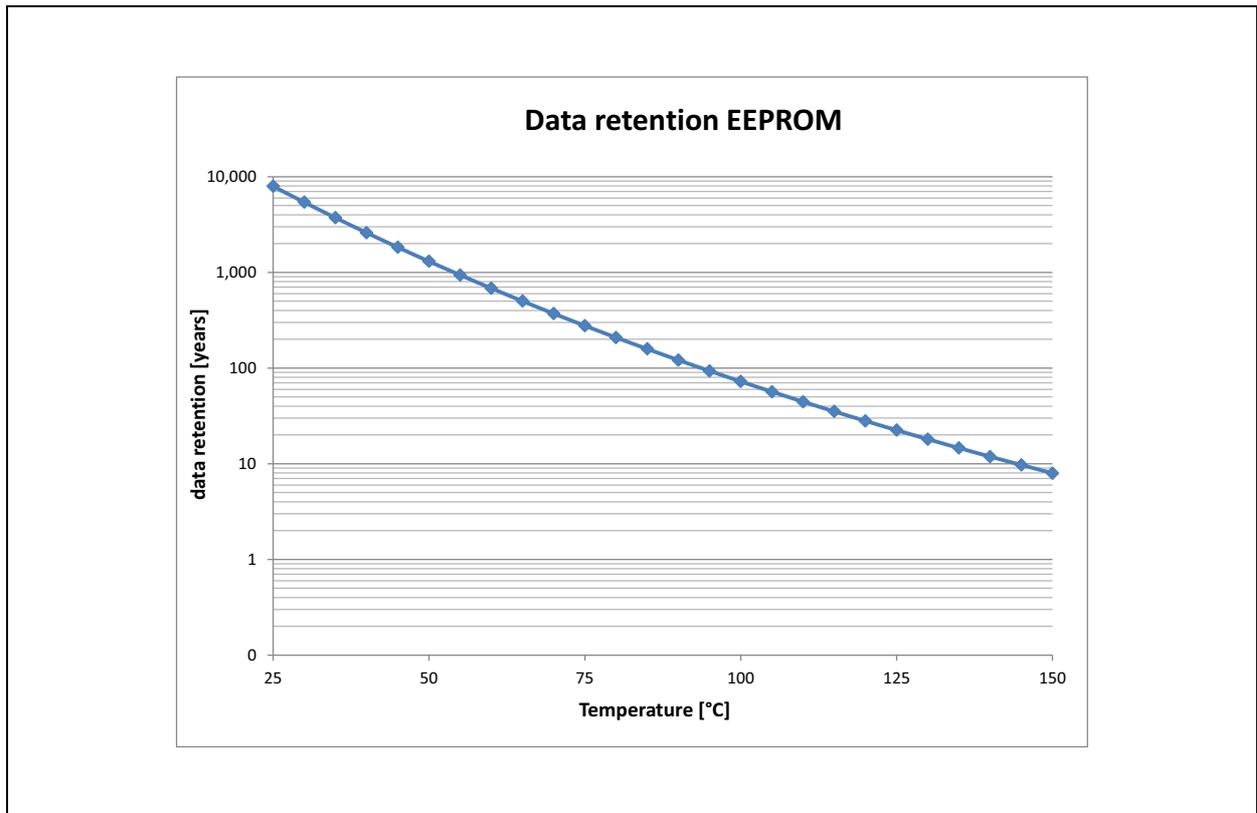


Figure 12:
EEPROM Data Retention Over Temperature



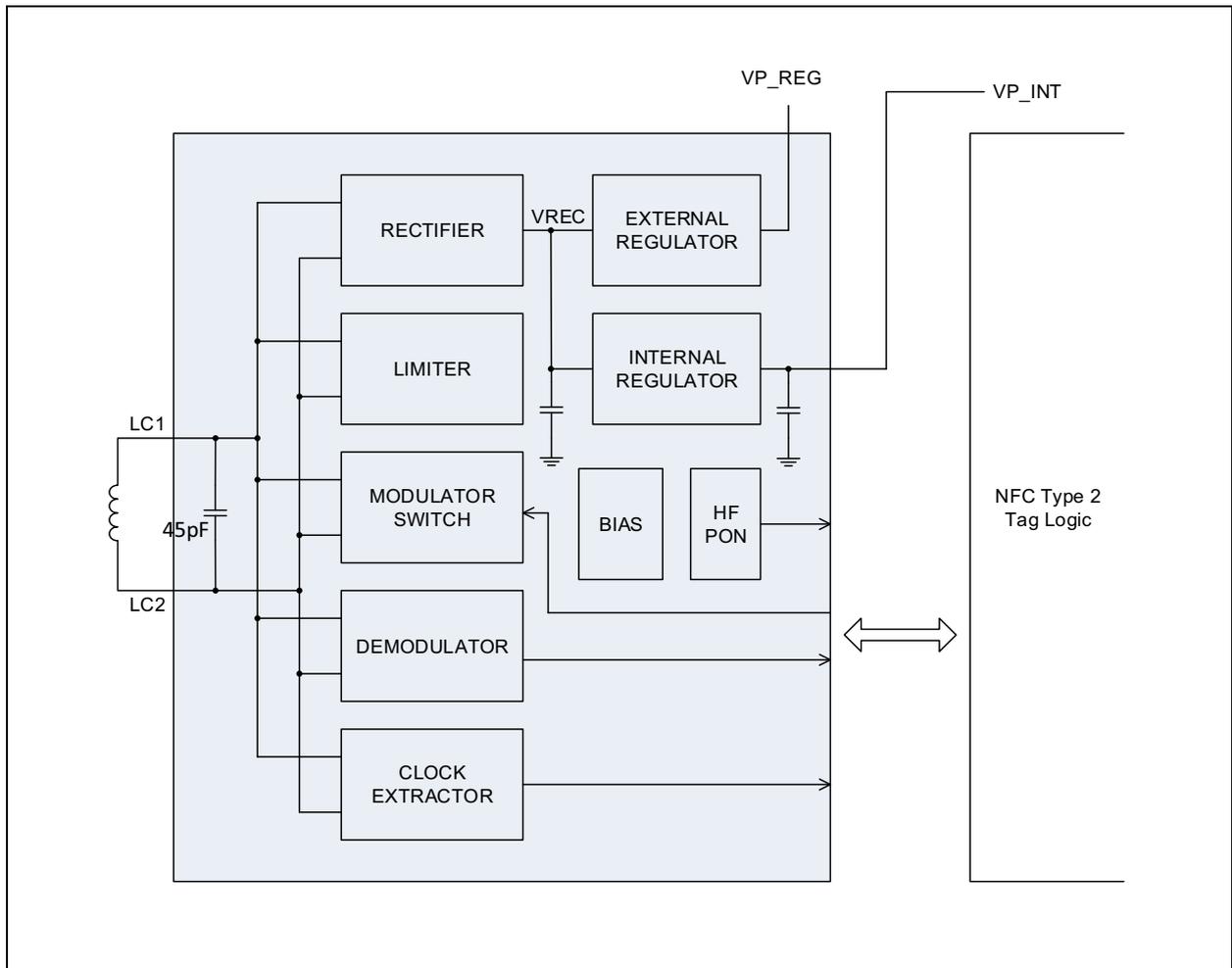
Detailed Description

Analog Frontend (AFE)

The AFE is connected to external tag coil, which together with the integrated resonant capacitor forms an LC tank resonating at the external electromagnetic field frequency (13.56 MHz).

Figure 13 depicts the main AFE building blocks.

Figure 13:
PICC AFE Block Diagram



Rectifier extracts DC power supply from the AC voltage induced on coil terminals.

Limiter limits the maximum voltage on coil terminals to protect AFE from destruction. At voltages that exceed limiter voltage it starts to absorb current (acts as some sort of shunt regulator).

Modulator Switch is used for communication the NFC tag to a NFC device. When switched on it will draw current from coil terminals. This mechanism is called load modulation. Variation of current in the modulator switch (ON and OFF state) is seen as modulation by the NFC device.

Demodulator is used for communication NFC device to NFC tag. It detects AM modulation of the NFC device magnetic field. The demodulator is designed to accept modulation according to NFC-A specifications ([[NFC Analog](#)] [[NFC Digital](#)]).

Clock Extractor extracts a digital clock signal from the PCD carrier field frequency which is used as clock signal by logic blocks.

HF_PON enables operation of the AFE and the logic when the supply voltage is sufficiently high. A buffer capacitor and HF_PON hysteresis guarantee that there is no reset during NFC device modulation.

Internal Regulator provides regulated voltage VP_INT to the AFE and in most cases also to EEPROM and logic blocks. Typical regulated voltage VP_INT is 2.0V. A buffer capacitor is also integrated.

External Regulator provides regulated voltage on external pin VP_REG where it can be used to supply some external circuitry. The regulated voltage and output resistance can be adjusted using EEPROM settings. Appropriate external buffer capacitor is needed in case VP_REG is used in the application. Current which may be provided depends on reader field strength, antenna size and Q factor, but it is limited to maximum 5mA.

Bias provides bias currents and reference voltages to AFE analog blocks.

Power Management

AS3955 power management comprises of four different modes to fit requirements of different applications. AS3955 supports two power sources, whose activation depends on the selected power mode.

Power Mode 0

In this power mode, the Power Manager is controlling the supply of the PICC Logic, EEPROM and SPI / I²C Interface (VDD). Its inputs are VP_INT (rectified and regulated supply extracted from RF field) and the VP_IO (supplied by external battery).

In standby mode, when AS3955 is not in the RF field (the condition is that rectified supply voltage is below HF_PON threshold) and the SPI / I²C is not active (/SS is high), the VDD supply is disconnected. The only current consumption in this state is leakage on VP_IO, mainly due to level shifters and SPI / I²C pins.

When AS3955 is placed in a RF field, VDD is connected to VP_INT. This happens once the VP_INT level is above the HF_PON threshold.

VP_IO is connected to VDD only when AS3955 is not in the RF field (rectified supply voltage is below HF_PON threshold) and the SPI / I²C interface is activated by pulling /SS signal low. The switch to VP_IO is controlled by /SS signal. The deactivation is delayed by 0.7ms minimum, so that the switch shall stay closed in case of shorter times between successive SPI / I²C activations. The switch is also closed during EEPROM writes activated over SPI / I²C.

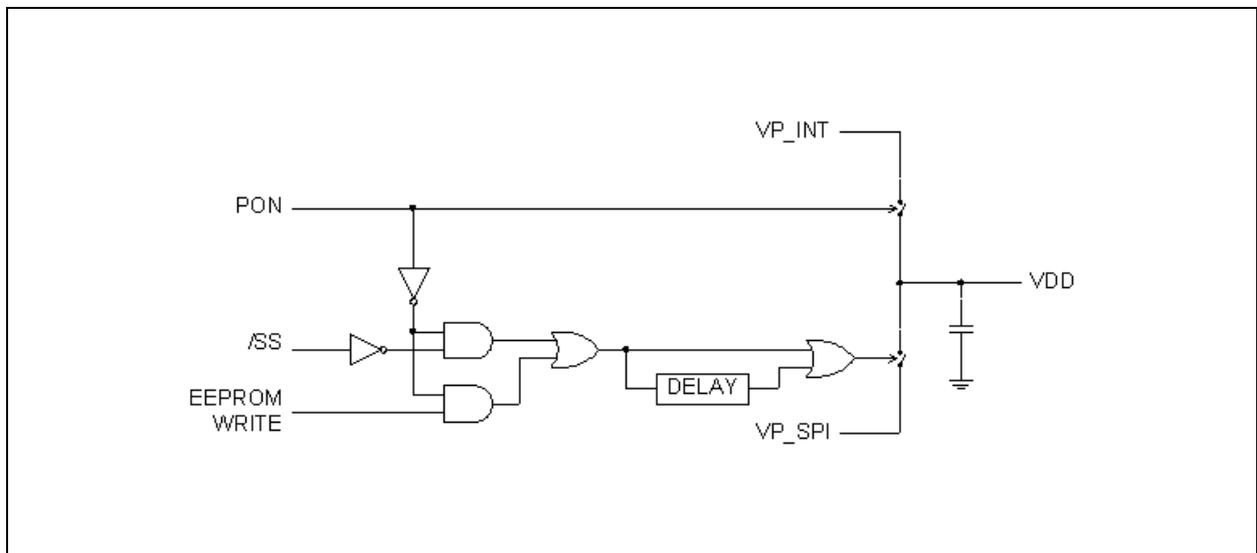
At activation of the switch, the time between the falling edge of the /SS signal and rising edge of SCLK shall be at least 300µs to allow charging of internal VDD buffer capacitor, expiration of POR signal and to perform a complete IC initialization. Please note that the only SPI / I²C operations, which are allowed in this mode, are read and write of EEPROM and registers.

If the RF field is lost during operation and the external system (MCU) is supplied over battery and /SS is low then power manager will automatically connect the VDD to VP_IO.

To enable low power mode where tag consumes less than 1uA at room temperature following conditions must be met:

- SPI interface configured
- All SPI interface input lines (including /SS) must be set to high
- All SPI output lines must be open

Figure 14:
Power Manager Concept



Power Mode 1

AS3955 is fully supplied by RF field. AS3955 checks if Extended or Tunneling mode are enabled. In this case, VP_REG supplies the system (SPI / I²C pads, pull-ups, MCU), otherwise energy harvesting is turned off and VP_REG is set in tristate.

Such power mode can be used in battery-less systems where the system is fully powered by the RF field. In such configuration, the VP_REG output pin for energy harvesting and VP_IO input supply shall be externally connected. Battery, even when present, will not be involved.

If this power mode is enabled and neither tunneling nor extended mode are enabled, *rreg* value from [IC Configuration Register 1](#) will be forced to zero. This will disable energy harvesting.

Power Mode 2

In this case, the external supply is used to provide power to digital blocks, EEPROM, SPI / I²C pads and MCU. External supply VP_REG is not used. Since this mode can be enabled only after initialization of the chip, the /SS line must be either permanently set to low or pulled low for short time (400µs) to complete the initialization.

This mode is specifically designed to operate with AS392x products (**ams'** Active Boost). In this mode, the HF_PON threshold of the chip will be decreased so that it will operate with external voltage on LC pin in the range of 2.7-3.6Vpp.

If this mode is enabled, AS3955 will not be turned off as long as there is an external supply present.

Power Mode 3

In this case the external supply shall be used to provide power to digital blocks, EEPROM, SPI / I²C pads and MCU. External supply VP_REG is not used. Since this mode can be enabled only after initialization of the chip, the /SS line must be either permanently set to low or pulled low for short time to complete the initialization.

In this mode, the HF_PON threshold of the chip is set so that it will operate with external voltage on LC pin in the range of 4.1-5Vpp.

If this mode is enabled, AS3955 will not be turned off as long as there is an external power present.

Interface Arbitration

Concurrent access to AS3955 internal EEPROM from RF or SPI / I²C requires arbitration, to resolve conflicts or undesired behaviour.

AS3955 implements two arbitration modes, which can be set in [Configuration Byte IC_CFG0](#).

Arbitration Mode 0

AS3955 arbitrates EEPROM write accesses according to first-come-first-serve principle.

- In case no write access is currently ongoing, both RF and SPI / I²C interfaces are allowed to write into EEPROM.
- In case a write request comes over RF, while SPI / I²C is writing, AS3955 will return a NAK.
- In case a write request comes over SPI / I²C while RF is writing, AS3955 will trigger a `I_err_acc` interrupt (see [Figure 92](#)).

Arbitration Mode 1

AS3955 gives always priority to RF accesses. In this mode, AS3955 behaves over RF as a pure contactless tag.

- In case SPI / I²C is performing a EEPROM write while the RF field is turned on, the write operation is interrupted to allow the initialization of the RF communication
- In case the RF field is already on and SPI / I²C performs a write to EEPROM and a READ or WRITE command is received via RF, the write operation of SPI / I²C is interrupted so that the RF operation can be performed

In both cases, a `I_err_acc` interrupt (see [Figure 92](#)) will be triggered.

Please note that the interruption of an EEPROM write may result in an undefined or “weak” state for the cell being programmed, and a second successful write attempt is suggested.

Energy Harvesting

AS3955 has energy harvesting capability. The regulated voltage output pin for energy harvesting is VP_REG. The energy harvesting is enabled only in Power Mode 0 and 1. [Figure 16](#) shows settings of the regulated voltage output. The output can be set between 1.8V and 4.5V in 100mV step. The output voltage and resistance can be set by [Configuration Byte IC_CFG1](#). The energy harvesting can be disabled by setting the output resistance register to 0 as shown in [Figure 15](#).

Figure 15:
Output Resistance Settings

| rreg<1:0> | Output Resistance | Comment |
|-----------|-------------------|--------------------------------------|
| 00b | X | Disabled – output pin is in tristate |
| 01b | 100Ω | |
| 10b | 50Ω | |
| 11b | 25Ω | |

[Figure 15](#) shows settings of the regulated voltage output. The output can be set between 1.8V and 4.5V in 100mV step.

Figure 16:
Regulated Voltage Output Settings

| vreg<4:0> | Output Voltage [V] |
|-----------|--------------------|
| 00000 | 1.9 |
| 00001 | 2.0 |
| 00010 | 2.1 |
| 00011 | 2.2 |
| 00100 | 2.3 |
| 00101 | 2.4 |
| 00110 | 2.5 |
| 00111 | 2.6 |
| 01000 | 2.7 |
| 01001 | 2.8 |
| 01010 | 2.9 |
| 01011 | 3.0 |
| 01100 | 3.1 |

| vreg<4:0> | Output Voltage [V] |
|-----------|--------------------|
| 01101 | 3.2 |
| 01110 | 3.3 |
| 01111 | 3.4 |
| 10000 | 3.5 |
| 10001 | 3.6 |
| 10010 | 3.7 |
| 10011 | 3.8 |
| 10100 | 3.9 |
| 10101 | 4.0 |
| 10110 | 4.1 |
| 10111 | 4.2 |
| 11000 | 4.3 |
| 11001 | 4.4 |
| 11010 | 4.5 |
| 11011 | 1.8 |
| 11100 | 1.8 |
| 11101 | 1.8 |
| 11110 | 1.8 |
| 11111 | 1.8 |

Abs. accuracy is $\pm 115\text{mV}$ at 1.9V linearly increasing to $\pm 225\text{mV}$, step size is $100\text{mV} \pm 20\text{mV}$.

Silent Mode

The Silent mode enables detection of the power status of a circuit whose supply (Vdd) is connected to VP_IO pin. If this mode is enabled and the voltage measured on pin VP_IO is below the configured threshold value, the RF part of AS3955 will be disabled, and the IC will not be responsive to incoming commands. Silent mode settings can be performed by using [Configuration Byte IC_CFG0](#).

This feature overcomes a potentially inconsistent behavior in a battery powered system, where a passive NFC tag can always communicate with a NFC device, also in case the battery is not sufficiently charged to supply the rest of the system. A typical example is when the NFC tag is used for Bluetooth pairing: AS3955 would trigger a pairing procedure only in case the system is fully operational by monitoring the supply voltage.

Memory Protection

AS3955 internal memory can be protected from unauthorized access by enabling password authentication. A 32-bit password can be set to protect the full user memory, or part of it, to allow the creation of a public data and a private data area. Password protection can be applied for read and write accesses.

Password authentication is performed through a standard WRITE command to the Authentication Password block. A maximum of 7 negative attempts are permitted before the chip is locked. Once authenticated, the user can modify the password.

Password protection applies to RF communication only.

Further information on how to handle password authentication can be found in [Authentication Password](#), [Configuration Byte AUTH_CFG](#), [Configuration Byte AUTH_CNT](#) and [Configuration Byte AUTH_LIM](#).

Passive Wake-Up

AS3955 is able to operate NFC tag operations standalone and fully powered by the RF field. The connected MCU can remain in standby/sleep mode as long as its intervention is not required by the application, in order to save power. AS3955 can be configured to notify the MCU through a wake-up interrupt.

A number of triggering events can be selected, e.g.:

- Power up
- SELECTED state entered
- Reception of SLP_REQ command
- NFC device has updated memory content

For a complete interrupt source list, please refer to the section [Interrupt Registers](#).

Chip Kill

Some applications require that the RF link is active only under certain conditions, e.g. during device configuration only in a controlled environment like a production facility.

AS3955 can be configured by the MCU in order to restrict the NFC device access to the system. By setting the [Configuration Byte CHIP_KILL](#) in EEPROM, the MCU can disable access to SPI / I²C from the RF link (i.e. Tunneling and Extended mode are permanently disabled), or even disable RF communication completely. In the latter case, AS3955 will not respond to incoming RF commands.

This configuration can be modified only by the MCU through SPI / I²C interfaces.

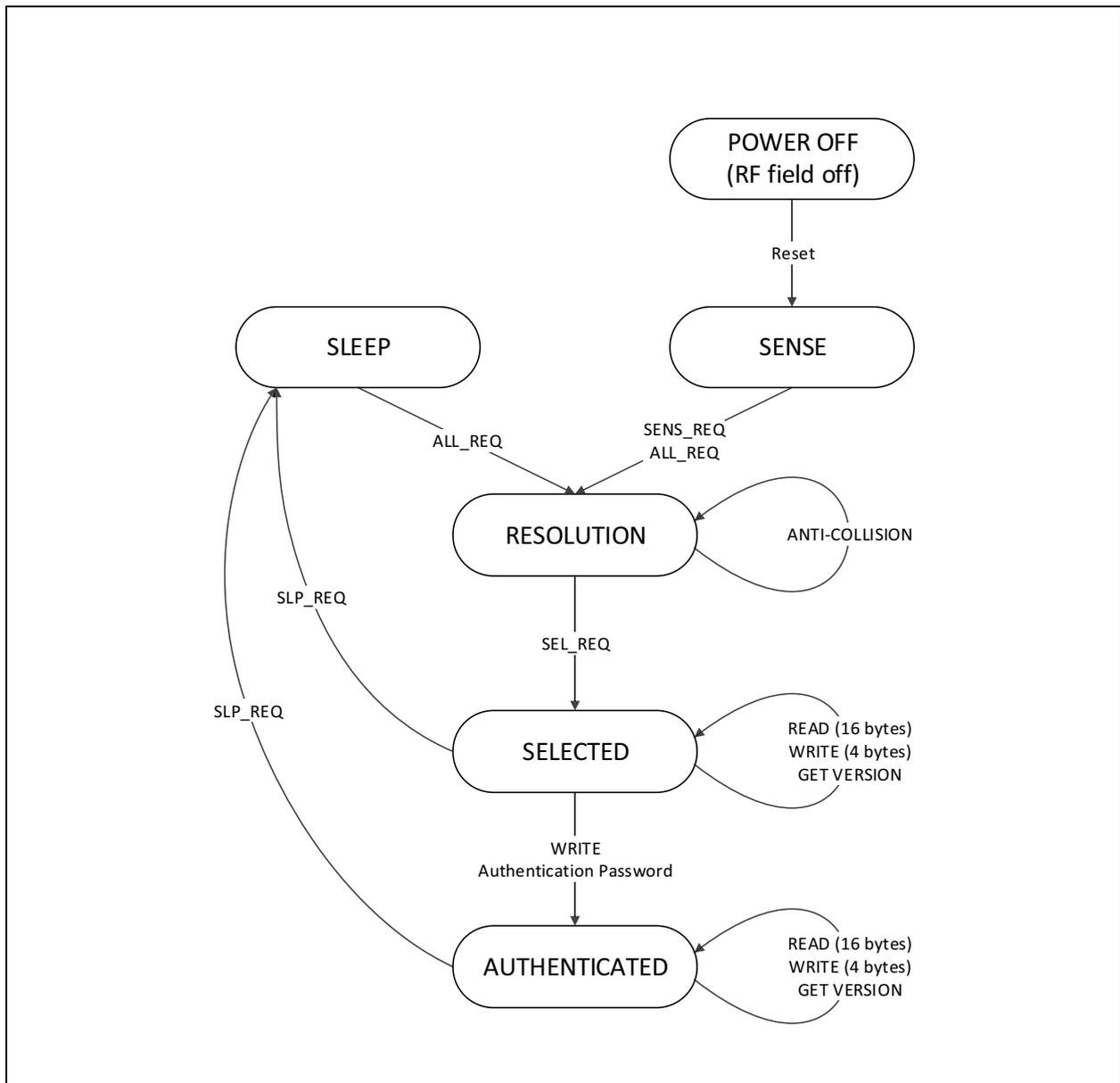
NFC Tag Functionality

Communication Principle

AS3955 autonomously executes complete NFC-A anti-collision communication sequence, during which the 7-byte UID is used ([\[NFC Analog\]](#) [\[NFC Digital\]](#)). After anti-collision, the NFC tag is brought into SELECTED state where read and write commands can be processed. The NFC tag will accept only read and write command issued to the address space actually available in AS3955 EEPROM. Any attempt to access an address outside the internal memory address space will be rejected. This default behavior of the NFC tag can be modified by enabling Tunneling or Extended mode.

A simplified AS3955 state diagram is shown in [Figure 17](#).

Figure 17:
AS3955 State Diagram



SENSE State

After a power-on reset (POR), AS3955 switches to the SENSE state. This state is exited when a SENS_REQ or an ALL_REQ command is received from the NFC device. Any other data received while in this state is interpreted as an error and AS3955 remains in SENSE state.

When in SELECTED or AUTHENTICATED state, a correctly executed SLP_REQ command will modify the default waiting state from SENSE to SLEEP state. SLEEP state can be exited when an ALL_REQ command is received.

SLEEP State

Together with SENSE state, SLEEP state is the other waiting state for AS3955. SLEEP state can be entered upon reception of a SLP_REQ command. The distinction between SENSE and SLEEP state is made necessary to discriminate selected and not yet selected tags.

AS3955 can only exit this state upon reception of an ALL_REQ command. Any other command received in this state is interpreted as an error and AS3955 state remains unchanged.

RESOLUTION State

In RESOLUTION state, the NFC device is resolving the tag UID. Since AS3955 has a double size UID, the RESOLUTION state actually comprises of two sub-states, where the anti-collision procedure is carried out in Cascade Level 1 and 2. Please refer to [ISO18092] for further information.

SELECTED State

All memory operations are operated in SELECTED state.

Upon reception of a SLP_REQ command, SELECTED state is exited and AS3955 transits to SLEEP state. Any other command received when the device is in this state is interpreted as an error. Depending on its previous state, AS3955 returns to either SENSE or SLEEP state.

Upon reception of a SECTOR SELECT command, AS3955 returns a NAK and transits to SENSE or SLEEP state, depending on its previous state.

AS3955 transits to the AUTHENTICATED state after successful password verification, using a standard WRITE command to a dedicated memory address (see [Authentication Password](#)). The number of permitted failed authentications is set to 7, after which AS3955 transits to LOCKED sub-state (not shown in the picture). When LOCKED state is entered, only the MCU can bring AS3955 back to SENSE state by resetting the authentication counter ([Configuration Byte AUTH_CNT](#)) back to 0 and issue a [Set Default](#), or [Go To Sense](#), or [Go To Sleep](#) command.

When in LOCKED sub-state, all memory operations are only allowed in the memory area not password protected, as defined by the configuration byte [Configuration Byte AUTH_LIM](#).

Upon reception of a SLP_REQ command, SELECTED state is exited and AS3955 transits to SLEEP state.

Any other command received when the device is in this state is interpreted as an error and, depending on its previous state, AS3955 returns to either SENSE or SLEEP state.

AUTHENTICATED State

In this state, all operations on memory blocks, which are configured as password verification protected, can be performed.

Upon reception of a SECTOR SELECT command, AS3955 returns a NAK and transits to SENSE or SLEEP state, depending on its previous state.

Upon reception of a SLP_REQ command, AUTHENTICATED state is exited and AS3955 transits to SLEEP state.

Any other command received when the device is in this state is interpreted as an error and, depending on its previous state, AS3955 returns to either SENSE state or SLEEP state.

NFC Forum Type 2 Tag Support

NFC Forum NFC-A commands ALL_REQ, SENS_REQ, SDD_REQ, SEL_REQ, SLP_REQ are required for anti-collision. Commands READ and WRITE are used for internal memory access. If NFC device issues a SECTOR SELECT command, AS3955 shall always reply with NAK.

Figure 18:
NFC-A vs ISO14443 Terminology

| NFC-A Term | ISO14443 Term |
|-----------------------------|---------------|
| States | |
| SENSE | IDLE |
| SLEEP | HALT |
| RESOLUTION | READY |
| SELECTED | ACTIVE |
| Commands / Responses | |
| SENS_REQ | REQA |
| ALL_REQ | WUPA |
| SENS_RES | ATQA |
| SSD_REQ | AC |
| SEL_REQ | SELECT |
| SLP_REQ | HLTA |

UID Coding

Anti-collision procedure is based on the Unique Identification Number (UID). AS3955 supports double size UID (7 bytes). First three bytes of the UID are hardwired inputs to the PICC Logic (uid<23:0>). The last 4 bytes of the UID are stored in EEPROM UID block.

First UID Byte (uid0)

The first byte of UID is Manufacturer ID according to [ISO7816-6]. It is coded on bits uid<7:0>. **ams** IC Manufacturer ID is 3Fh.

Second UID Byte (uid1)

The second byte of UID (uid<15:8>) is reserved for **ams**' chip type (IC Type). Every **ams**' RFID tag IC has its own chip type assigned. AS3955 IC type is 14h.

Third UID Byte (uid2)

The third byte of UID (uid<23:16>) is set to 00h.

Figure 19:
Coding of First Three UID Bytes

| UID Byte | Value (Hex) |
|----------|-------------|
| uid0 | 3F |
| uid1 | 14 |
| uid2 | 00 |

Last Four UID Bytes (uid3-uid6)

The last 4 bytes of UID are read from EEPROM (UID block) and pre-programmed during IC production. Those 4 bytes are unique, and cannot be modified.

Figure 20:
Last Four UID Bytes

| UID Byte | UID Block Bits |
|----------|----------------|
| uid3 | b7-b0 |
| uid4 | b15-b8 |
| uid5 | b23-b16 |
| uid6 | b31-b24 |

Coding of SENS_RES, SEL_REQ, ACK and NACK

Several bits in certain responses are defined as don't-care in the NFC-A standard [NFC Digital], some others are defined by optional choices in standard protocol. This chapter defines how these bits are actually set in AS3955.

SENS_RES Response

SENS_RES is a response on ALL_REQ and SENS_REQ commands. The SENS_RES is defined by Configuration Bytes SENS_R1 and SENS_R2 stored in EEPROM.

Figure 21:
Coding of SENS_RES Response

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|--------|-----|-----|-----|-----|-----|----|----|--------|----|----|----|----|----|----|----|
| SENSR2 | | | | | | | | SENSR1 | | | | | | | |

SEL_RES Response, Cascade Level 1 and 2

SEL_RES is the response to SEL_REQ command. Since AS3955 UID is double sized, SEL_RES responses for Cascade Level 1 and Cascade Level 2 are defined. SEL_RES on Cascade Level 1 and 2 is defined with Configuration Byte SELR except for cascade bit 3. The response on Cascade Level 2 is also configured by *selr_b6_inv* bit which, when set, inverts cascade bit 6 in SEL_RES response on Cascade Level 2 (see IC_CFG2).

Figure 22:
SEL_RES CL1 Coding

| b8 MSB | b7 | b6 | b5 | b4 | b3 | b2 | b1 LSB | Description |
|--------------|----|----|----|----|----|--------------|--------|-----------------------------------|
| sel_res<7:3> | | | | | 1 | sel_res<1:0> | | Cascade bit set: UID not complete |

Figure 23:
SEL_RES CL2 Coding

| b8 MSB | b7 | b6 | b5 | b4 | b3 | b2 | b1 LSB | Description |
|--------------|----|----------------|--------------|----|----|--------------|--------|----------------------|
| sel_res<7:6> | | sel_res<5> | sel_res<4:3> | | 0 | sel_res<1:0> | | selr_b6_inv set to 0 |
| | | NOT sel_res<5> | | | | | | selr_b6_inv set to 1 |

Note(s):

- According to [ISO14443-3], all bits except b3 are "don't care" for Cascade Level 1, and all bits except b6 and b3 are "don't care" for Cascade Level 2. Bit b6 in CL2 indicates whether the tag is compliant to [ISO14443] or not (resp. b6=1 and b6=0). In case of applications requiring EMVCo compliance, bit b6 in Cascade Level 1 shall be set as bit b6 in Cascade Level 2 ([EMVCO-1]).

ACK Response

The ACK response of AS3955 is a 4-bit value Ah.

NACK Response

The AS3955 uses all four combinations of NAK values. The usage of various NAK values is explained in section [Error Handling](#).

Access to UID, SENS_RES and SEL_REQ During Anti-Collision

UID block, [SENSR1](#), [SENSR2](#) and [SELR](#) bytes are stored in a buffer. The purpose of storing these data into the buffer is faster access to the data and UID verification during the anti-collision procedure. Buffer access over SPI / I²C is locked until NFC tag enters SELECTED state.

Get Version Command

In addition to standard NFC tag commands, AS3955 supports a custom Get Version command. This command consists of 8 bits and shall be transmitted only in standalone and Extended mode when the tag is SELECTED state. The command code and the tag response are shown resp. in [Figure 24](#) and [Figure 25](#).

Figure 24:
Coding of Get Version Command

| b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 |
|----|----|----|----|----|----|----|----|
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

Figure 25:
Response to Get Version Command

| Byte No. | Description | | Comment | |
|----------|-----------------------|------------|------------------|--------------|
| 0 | Fixed Header | 00h | | |
| 1 | Vendor Id | 3Fh | ams AG | |
| 2 | Product Type | 14h | AS395x | |
| 3 | Product Subtype | 01h | AS3955 | |
| 4 | Major Product Version | 02h | 2 ⁽³⁾ | |
| 5 | Minor Product Version | 02h | 2 ⁽³⁾ | |
| 6 | Storage Size | 15h | 17h | See note (1) |
| 7 | Supported Features | 01h or 02h | | See note (2) |

Note(s):Notes:

- The most significant 7 bits of the storage size byte are interpreted as an unsigned integer value n . As a result, it codes the total available user memory size as 2^n . If the least significant bit is 0b, the user memory size is exactly 2^n . If the least significant bit is 1b, the user memory size is between 2^n and 2^{n+1} .
The user memory is the memory available for user data (Capability Container and lock bits are excluded)
- The figure below is maintained consistently across the whole AS395x family, where Get Version command is supported.
- For all ordering codes beginning AS3955A the major product version is 02h and the minor product version is 02h.

Figure 26:
Get Version Response, Byte 7

| Get Version | |
|-------------|--|
| Byte No. 7 | Description |
| b7 | 00000b: ISO14443A Level 3 supported, max baudrate 106 kbps, Tunneling and Extended mode present, Password protection available, energy harvesting available Others: RFU |
| b6 | |
| b5 | |
| b4 | |
| b3 | |
| b2 | 000b: No wired interface available 001b: SPI slave, passive wake-up available 010b: I ² C slave, passive wake-up available |
| b1 | |
| b0 | |

Memory Organization

AS3955 contains an embedded EEPROM module which can be accessed via RF or SPI / I²C interface. EEPROM contains 4096 bits (512 bytes) organized in 128 blocks of 4 bytes each. Blocks in EEPROM are numbered from 00h to 7Fh. Bits in a block are numbered from 0 to 31.

Most of the EEPROM is NFC Type 2 Tag user data area (472 bytes). The position of the dynamic lock bits is fixed at the end of NFC Type 2 Tag user data area (blocks 7Ah and 7Bh). The configuration bits which define AS3955 operating options are stored in blocks 01h, 7Ch, 7Dh, 7Eh, 7Fh. Housekeeping information is stored in block 00h. 4kbit EEPROM organization is shown in [Figure 27](#).

4kbit EEPROM Organization

Numbers of dynamic lock bits:

- Data area size in bytes:
 $4 * (127 - 3) - 7 - 17 = 472$ bytes
- Number of dynamic lock bits:
 $(472 - 48) / 8 = 53$ bits (53 bits)

Figure 27:
4kbit EEPROM Organization

| Byte Number in Block | | | | | | |
|----------------------|------------|------------|----------|----------|------------------|--------|
| Block | 0 | 1 | 2 | 3 | Description | Access |
| 00h | UID0 | UID1 | UID2 | UID3 | UID / Internal | RO |
| 01h | FAB_CFG0 | FAB_CFG1 | FAB_CFG2 | FAB_CFG3 | Fabrication data | RO |
| 02h | Internal 8 | Internal 9 | Lock 0 | Lock 1 | Internal / Lock | OTP |
| 03h | CC 0 | CC 1 | CC 2 | CC 3 | CC | OTP |
| 04h | Data 0 | Data 1 | Data 2 | Data 3 | Data | RW |
| 05h | Data 4 | Data 5 | Data 6 | Data 7 | Data | RW |
| 06h | Data 8 | Data 9 | Data 10 | Data 11 | Data | RW |
| 07h : : 79h | | | | | Data | RW |

| Byte Number in Block | | | | | | |
|----------------------|-----------|----------|-----------------------|------------|-------------------------|--------|
| Block | 0 | 1 | 2 | 3 | Description | Access |
| 7Ah | Lock 2 | Lock 3 | Lock 4 | Lock 5 | Lock | OTP |
| 7Bh | Lock 6 | Lock 7 | Lock 8 ⁽¹⁾ | Reserved 0 | Lock / Reserved | OTP |
| 7Ch | RFP0 | RFP1 | RFP2 | RFP3 | Authentication password | RW |
| 7Dh | CHIP_KILL | AUTH_CNT | AUTH_LIM | AUTH_CFG | Authentication settings | RW |
| 7Eh | SENSR1 | SENSR2 | SELR | IC_CFG0 | Config. block 0 | RW |
| 7Fh | IC_CFG1 | IC_CFG2 | MIRQ_0 | MIRQ_1 | Config. block 1 | RW |

Note(s):

1. Bits that are not used should be set to 0.

Access properties:

RO: Read only, writing to this word is not possible

RW: Reading and writing to this word is possible

OTP: One time programmable. A bit of this word once set to 1 cannot be set back to 0.

UID Bytes

The UID block contains four LSB bytes of the 7-byte UID which is used during anti-collision and selection process. Every IC is programmed by a unique number during fabrication process at **ams**. See [UID Coding](#) about details on UID.

This block stores some IC manufacturer data which is programmed and locked during fabrication process at **ams**.

UID is treated as IC internal configuration, and is permanently locked during IC production.

Fabrication Data

Fabrication data are used to set internal configuration and trimming values. They are treated as IC internal configuration.

Fabrication Data FAB_CFG0

Figure 28:
Fabrication Data FAB_CFG0

| Conf. Bit | Name | Default | Function | Note |
|-----------|------|---------|------------------------|------|
| b7 | | 0 | Internal configuration | |
| b6 | | 0 | | |
| b5 | | 0 | | |
| b4 | | 0 | | |
| b3 | | 0 | | |
| b2 | | 0 | | |
| b1 | | 0 | | |
| b0 | | 0 | | |

Note(s):

1. This byte can be set only during production.

Fabrication Data FAB_CFG1

Figure 29:
Fabrication Data FAB_CFG1

| Conf. Bit | Name | Default | Function | Note |
|-----------|---------|---------|---|------|
| b7 | fdel<3> | 0 | PCD to PICC frame delay time compensation; frame compensation defined as $fdel * 1 / fc$ | |
| b6 | fdel<2> | 0 | | |
| b5 | fdel<1> | 0 | | |
| b4 | fdel<0> | 0 | | |
| b3 | osct<1> | 0 | Oscillator trimming bits | |
| b2 | osct<0> | 0 | | |
| b1 | decc<1> | 0 | Decoder compensation register | |
| b0 | decc<0> | 0 | | |

Note(s):

1. This byte can be set only during production.

The *fdel* bits define frame delay time (FDT) adjustment and represent a time compensation in number of clocks of carrier frequency. The *osct* bits are trimming bits for the internal oscillator.

Fabrication Data FAB_CFG2

Figure 30:
Fabrication Data FAB_CFG2

| Conf. Bit | Name | Default | Function | Note |
|-----------|-----------|---------|--|------|
| b7 | test_mode | 0 | | |
| b6 | mod_r | 0 | 1: decreased modulator switch resistance | |
| b5 | miso_pd2 | 0 | 1: pull down on MISO, when \SS is low and MISO is not driven by the AS3955 | |
| b4 | miso_pd1 | 0 | 1: pull down on MISO when \SS is high | |
| b3 | rfu | 0 | | |
| b2 | rfu | 0 | | |
| b1 | rfu | 0 | | |
| b0 | rfu | 0 | | |

Note(s):

1. This byte can be set only during production.

Fabrication Data FAB_CFG3

Figure 31:
Fabrication Data FAB_CFG3

| Conf. Bit | Name | Default | Function | Note |
|-----------|------------|---------|-----------------------------|------|
| b7 | uid_crc<7> | | CRC value calculated on UID | |
| b6 | uid_crc<6> | | | |
| b5 | uid_crc<5> | | | |
| b4 | uid_crc<4> | | | |
| b3 | uid_crc<3> | | | |
| b2 | uid_crc<2> | | | |
| b1 | uid_crc<1> | | | |
| b0 | uid_crc<0> | | | |

Note(s):

1. This byte can be set only during production.

Reserved Bytes

The reserved bytes belong to reserved memory areas.

OTP Blocks

Write and Read Lock blocks are OTP (One Time Programmable). This means that once they are set to 1, they cannot be set back to 0. Since setting OTP bits is an irreversible operation, it is strongly recommended to perform this operation in controlled environment.

Lock Bits

The bits of byte 2 and 3 of block 02h represent the field-programmable read-only locking mechanism called “static lock bytes”. They are called static because their position in memory is fixed.

When data memory is larger than 16 blocks (64 bytes), also “dynamic lock bytes” are required. They are located starting at address 7Ah (4kbit version). Block lock granularity is 1 block per bit for the first 16 blocks, 2 blocks per bit for the remaining blocks.

Lock bits are OTP, i.e. setting bits to 1b is an irreversible operation. Bits at 0b can be set to 1b through a WRITE operation, the result being a bit-wise OR with the current value.

Lock bits apply only to RF interface, as SPI / I²C interface has unlimited access to user data area.

- Lock 0 byte locks 8 blocks starting from address 00h where lock bit 0 locks block on address 00h
- Lock 1 byte locks 8 blocks starting from address 08h where lock bit 0 locks block on address 08h
- Lock 2 byte locks 16 blocks starting from address 10h where lock bit 0 locks block on address 10h and 11h
- Lock 3 byte locks 16 blocks starting from address 20h where lock bit 0 locks block on address 20h and 21h
- Lock 4 byte locks 16 blocks starting from address 30h where lock bit 0 locks block on address 30h and 31h
- Lock 5 byte locks 16 blocks starting from address 40h where lock bit 0 locks block on address 40h and 41h
- Lock 6 byte locks 16 blocks starting from address 50h where lock bit 0 locks block on address 50h and 51h
- Lock 7 byte locks 16 blocks starting from address 60h where lock bit 0 locks block on address 60h and 61h
- Lock 8 byte locks 16 blocks starting from address 70h where lock bit 0 locks block on address 70h and 71h

Figure 32:
Example of Lock Bits

| Lock 0 byte | | | | | | | | Lock 1 byte | | | | | | | |
|-------------|-----|-----|-----|-----|-----|-----|-----|-------------|-----|-----|-----|-----|-----|-----|-----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| locks block | | | | | | | | locks block | | | | | | | |
| 07h | 06h | 05h | 04h | 03h | 02h | 01h | 00h | 0Fh | 0Eh | 0Dh | 0Ch | 0Bh | 0Ah | 09h | 08h |

Capability Container

Block 03h in AS3955 EEPROM contains the Capability Container (CC), pre-programmed during IC production according to NFC Forum Type 2 Tag specifications [T2T]. CC bits are OTP, i.e. setting bits to 1b is an irreversible operation. Bits at 0b can be set to 1b through a WRITE operation, the result being a bit-wise OR with the current value.

Figure 33 shows the Capability Container content at delivery.

Figure 33:
CC Content at Delivery (4kbit Version)

| Block address | Byte number in block | | | |
|---------------|----------------------|-----|-----|-----|
| | 0 | 1 | 2 | 3 |
| 03h | E1h | 10h | 3Bh | 00h |

Configuration Bytes

The Configuration bytes are used to define AS3955 operating options. AS3955 is delivered by **ams** with default settings.

Authentication Password

The Authentication password block (bytes RFP0, RFP1, RFP2, and RFP3) contains the 32-bit password. This password is used for authentication over RF side. The NFC tag is set into AUTHENTICATED state when a write command is issued via the RF to write password address with the data that has same content as the data stored in Authentication password block. If the NFC tries to authenticate with a wrong password, AS3955 shall not respond and returns into SENSE / SLEEP state and the value of the [AUTH_CNT](#) is decreased.

The password can be overwritten via RF only in AUTHENTICATED state and can always be set via SPI / I²C. Configuration register [AUTH_CFG](#) defines access rights controlled by the password. The RF password can't be read via RF.

The authentication for read or write is required only to the memory portion defined by [AUTH_LIM](#). The chip is no longer in AUTHENTICATED state when the tag leaves SELECTED state.

An attempt to read the password block will return zeroes. The authentication does not override permissions set by the lock bits. Authentication also does not restrict access over SPI / I²C in any way.

Authentication can be configured using configuration bits in [AUTH_CFG](#) and [AUTH_LIM](#) bytes.

Configuration Byte CHIP_KILL

Figure 34:
Configuration Byte CHIP_KILL

| Conf. Bit | Name | Default | Function | Note |
|-----------|-------------|---------|---|------|
| b7 | chip_kill_2 | 0 | 1: Tunneling and Extended mode are disabled | |
| b6 | chip_kill_1 | 0 | 1: RF communication part is disabled | |
| b5 | rfu | 0 | | |
| b4 | rfu | 0 | | |
| b3 | rfu | 0 | | |
| b2 | rfu | 0 | | |
| b1 | rfu | 0 | | |
| b0 | rfu | 0 | | |

Note(s):

1. This byte can always be accessed for read and write via the SPI / I²C and can't be accessed for read and write from the RF side.

By setting **CHIP_KILL** byte the RF communication part of the chip or Tunneling and Extended mode will be disabled permanently. At its initial state, the **CHIP_KILL** byte is set to value 00h. The value of this byte can be changed via SPI / I²C. If bit b6 is set to value 1, the RF part of the chip shall be disabled and AS3955 will not respond to any command received from a NFC device. By setting bit b7 to 1, Tunneling and Extended mode will be disabled.

Configuration Byte AUTH_CNT

Figure 35:
Configuration Byte AUTH_CNT

| Conf. Bit | Name | Default | Function | Note |
|-----------|--------------|---------|--------------------------|------|
| b7 | rfu | 0 | | |
| b6 | auth_cnt2<2> | 1 | Authentication counter 2 | |
| b5 | auth_cnt2<1> | 1 | | |
| b4 | auth_cnt2<0> | 1 | | |
| b3 | rfu | 0 | | |
| b2 | auth_cnt1<2> | 1 | Authentication counter 1 | |
| b1 | auth_cnt1<1> | 1 | | |
| b0 | auth_cnt1<0> | 1 | | |

Note(s):

1. This byte can always be accessed for read and write via the SPI / I²C and can't be accessed for read and write from the RF side.

This byte indicates the number of allowed unsuccessful authentication attempts over RF available before disabling the chip. The byte consists of two separate counters where the second counter is a copy of the first counter.

These counters are updated at each failed authentication. At each successful authentication, counters are reset to 7 and, at each unsuccessful authentication attempt, counters are decreased by one. If the value of the counters reaches 0, the chip will be permanently locked and cannot be authenticated any longer over the RF field. The chip will also be locked in case the two counter values don't match. The lock can always be cleared via SPI / I²C interface.

Configuration Byte AUTH_LIM

Figure 36:
Configuration Byte AUTH_LIM

| Conf. Bit | Name | Default | Function | Note |
|-----------|-------------|---------|---|------|
| b7 | auth_lim<7> | 1 | AUTH_LIM defines the block address above which password verification is required. | |
| b6 | auth_lim<6> | 1 | | |
| b5 | auth_lim<5> | 1 | | |
| b4 | auth_lim<4> | 1 | | |
| b3 | auth_lim<3> | 1 | | |
| b2 | auth_lim<2> | 1 | | |
| b1 | auth_lim<1> | 1 | | |
| b0 | auth_lim<0> | 1 | | |

Note(s):

1. This byte can always be accessed for read and write via SPI / I²C and can be accessed for read and write from the RF side if chip is in AUTHENTICATED state and configuration bit *auth_set* is 1.

AUTH_LIM defines the block address above which password verification is required. Valid address range for the AUTH_LIM byte is from 00h to FFh. If AUTH_LIM is set to a block address higher than the last block of the EEPROM address space, the password protection is effectively disabled. Addresses above the limit are protected for read / write depending on *auth_r* and *auth_w* values. If no bits are set, password protection is disabled.

If the NFC device tries to access protected blocks without authenticating first, then:

- If only protected blocks are accessed, AS3955 will not respond
- If protected and unprotected blocks are accessed¹, AS3955 will return actual stored values only for the unprotected portion, and zeroes for the protected portion.

1. This can occur, for instance, with a READ command crossing the border between protected and unprotected memory.

Configuration Byte AUTH_CFG
Figure 37:
Configuration Byte AUTH_CFG

| Conf. Bit | Name | Default | Function | Note |
|-----------|-----------|---------|--|------|
| b7 | rfu | 0 | | |
| b6 | rfu | 0 | | |
| b5 | rfu | 0 | | |
| b4 | rfu | 0 | | |
| b3 | rfu | 0 | | |
| b2 | rfu | 0 | | |
| b1 | auth_w<1> | 0 | Authentication is required for writing | |
| b0 | auth_r<0> | 0 | Authentication is required for reading | |

Note(s):

1. This byte can always be accessed for read and write via SPI / I²C and can be accessed for read and write from the RF side if chip is in AUTHENTICATED state and configuration bit *auth_set* is 1.

Bits *auth_w* and *auth_r* define for which operations the authentication is needed. If a lock bit is set for a certain block, then write cannot be performed even if IC is in AUTHENTICATED state. This means that lock bits overrule authentication.

Configuration Byte SENS1

Figure 38:
Configuration Byte SENS1

| Conf. Bit | Name | Default | Function | Note |
|-----------|--------------|---------|--------------------------------------|------|
| b7 | sens_res<15> | 0 | SENS_RES response byte 2 on SENS_REQ | |
| b6 | sens_res<14> | 0 | | |
| b5 | sens_res<13> | 0 | | |
| b4 | sens_res<12> | 0 | | |
| b3 | sens_res<11> | 0 | | |
| b2 | sens_res<10> | 0 | | |
| b1 | sens_res<9> | 0 | | |
| b0 | sens_res<8> | 0 | | |

Note(s):

1. This byte can always be accessed for read and write via SPI / I²C and can be accessed for read and write from the RF side if *rfcfg_en* is set to 1.

Configuration Byte SENS2

Figure 39:
Configuration Byte SENS2

| Conf. Bit | Name | Default | Function | Note |
|-----------|-------------|---------|--------------------------------------|------|
| b7 | sens_res<7> | 0 | SENS_RES response byte 1 on SENS_REQ | |
| b6 | sens_res<6> | 1 | | |
| b5 | sens_res<5> | 0 | | |
| b4 | sens_res<4> | 0 | | |
| b3 | sens_res<3> | 0 | | |
| b2 | sens_res<2> | 1 | | |
| b1 | sens_res<1> | 0 | | |
| b0 | sens_res<0> | 0 | | |

Note(s):

1. This byte can always be accessed for read and write via SPI / I²C and can be accessed for read and write from the RF side if *rfcfg_en* is set to 1.

Configuration Byte SELR

Figure 40:
Configuration Byte SELR

| Conf. Bit | Name | Default | Function | Note |
|-----------|------------|---------|---------------------------------------|--|
| b7 | sel_res<7> | 0 | SEL_RES response on Cascade Level 1/2 | |
| b6 | sel_res<6> | 0 | | |
| b5 | sel_res<5> | 0 | | |
| b4 | sel_res<4> | 0 | | |
| b3 | sel_res<3> | 0 | | |
| b2 | sel_res<2> | 0 | | This bit is not used, as cascade bit 3 in SEL_RES CL1 is fixed to 1, and to 0 in SEL_RES CL2 |
| b1 | sel_res<1> | 0 | | |
| b0 | sel_res<0> | 0 | | |

Note(s):

1. This byte can always be accessed for read and write via SPI / I²C and can be accessed for read and write from the RF side if *rfcfg_en* is set to 1.

Configuration Byte IC_CFG0

Figure 41:
Configuration Byte IC_CFG0

| Conf. Bit | Name | Default | Function | Note |
|-----------|--------------|---------|---|------|
| b7 | slnt_mod | 0 | 1: Enable silent mode | |
| b6 | slnt_vl<2> | 0 | Silent mode voltage level (see Silent Mode) | |
| b5 | slnt_vl<1> | 0 | | |
| b4 | slnt_vl<0> | 0 | | |
| b3 | arbit_mod | 0 | 1: RF has priority access to EEPROM over SPI / I ² C | |
| b2 | i2c_addr3<2> | 0 | I ² C slave address | |
| b1 | i2c_addr2<1> | 0 | | |
| b0 | i2c_addr1<0> | 0 | | |

Note(s):

1. This byte can always be accessed for read and write via SPI / I²C and can be accessed for read and write from the RF side if *rfcfg_en* is set to 1. Bit *arbit_mod* can be modified after initialization over SPI / I²C.

Bit *slnt_mod* enables Silent mode. In this mode, the supply pin VP_IO is being observed. If voltage is below the level defined in *slnt_vl<2:0>*, then Silent mode is activated. This means that RF part of the IC is turned off and stops being responsive to incoming RF commands.

Voltage threshold settings on VP_IO are shown in [Figure 42](#).

Figure 42:
Silent Mode Threshold Voltage Levels

| <i>slnt_vl<2:0></i> | Voltage Threshold | Abs. Accuracy |
|---------------------------|-------------------|--------------------------------|
| 000b | 1.42V | ±15mV |
| 001b | 1.62V | Linearly increasing over range |
| 010b | 1.82V | |
| 011b | 2.23V | |
| 100b | 2.53V | |
| 101b | 2.74V | |
| 110b | 3.04V | |
| 111b | 3.65V | ±25mV |

The voltage level of the supply on VP_IO is measured when tag enters RF field. The selected voltage level threshold can be properly measured only if the RF field is strong enough to provide sufficient supply voltage level. When small antennas are used, it is advisable to set lower threshold.

Bit *arbit_mod* defines arbitration mode during mutual EEPROM access via SPI / I²C and RF side. If *arbit_mod* is set to 0, then the EEPROM access follows the first-come-first-serve principle. If *arbit_mod* is set to 1, then the RF part will always have higher priority over SPI / I²C. For further details, please refer to section [Interface Arbitration](#).

The *i2c_addr* bits represent lower three bits of the I²C address. The upper four bits of the I²C address that represent a group shall be set to 1010b.

Configuration Byte IC_CFG1**Figure 43:**
Configuration Byte IC_CFG1

| Conf. Bit | Name | Default | Function | Note |
|-----------|-----------|---------|--|------|
| b7 | en_rx_crc | 0 | 1: CRC stored in the buffer in the tunneling mode | |
| b6 | vreg<4> | 0 | Voltage level for voltage regulator VP_REG (see Figure 16) | |
| b5 | vreg<3> | 0 | | |
| b4 | vreg<2> | 0 | | |
| b3 | vreg<1> | 0 | | |
| b2 | vreg<0> | 0 | | |
| b1 | rreg<1> | 0 | Output resistance value for voltage regulator VP_REG (see Figure 15) | |
| b0 | rreg<0> | 0 | | |

Note(s):

1. This byte can always be accessed for read and write via SPI / I²C and can be accessed for read and write from the RF side if *rfcfg_en* is set to 1.

If bit *en_rx_crc* is set to 1 then the CRC shall be part of the message in the buffer. This implies that maximum message effective length is reduced to 30 bytes. CRC check is performed regardless of the value of the *en_rx_crc* bit.

Configuration Byte IC_CFG2

Figure 44:
Configuration Byte IC_CFG2

| Conf. Bit | Name | Default | Function | | Note |
|-----------|-------------------|---------|---|----------------------------------|------|
| b7 | rfcfg_en | 1 | 1: Enables personalization / configuration over RF | | |
| b6 | tun_mod | 0 | 1: Enables Tunneling Mode | | |
| b5 | ext_mod | 0 | 1: Enables Extended Mode | | |
| b4 | nak_on_crc_parity | 0 | 1: Defines error handling and response | | |
| b3 | auth_set | 0 | 1: Configuration of the authentication settings is enabled from RF side | | |
| b2 | selr_b6_inv | 0 | 1: Inverts bit 6 in SEL_RES response on Cascade Level 2 | | |
| b1 | powm<1> | 0 | 00: Power Mode 0 | 10: Power Mode 2 | |
| b0 | powm<0> | 0 | 01: Power Mode 1 | 11: Power Mode 3 | |

Note(s):

1. This byte can always be accessed for read and write via SPI / I²C and can be accessed for read and write from the RF side if *rfcfg_en* is set to 1. Bits *tun_mod* and *ext_mod* represents a default value stored in a volatile memory which can be modified after initialization over SPI / I²C.

Bit *rfcfg_en* enables the personalization process during production at customer facilities. When this bit is set to 0, the modification of the last two blocks is not possible anymore over the RF field.

If *auth_set* is set then changing of authenticated setting (authentication limits, read/write permission) is enabled over the RF after successful authentication. Password can always be changed via the RF side if tag is in authenticated state regardless of the value of read/write bits.

Bit *nak_on_crc_parity* configures error handling mechanism as described in [Error Handling](#).

Bit *selr_b6_inv* configures [SEL_RES](#) response on Cascade Level 2. If *selr_b6_inv* is set to 1, bit b6 in [SEL_RES](#) response on Cascade Level 2 will be inverted, otherwise it will be set as configured in [SELR](#) byte.

Bits *powm<1:0>* are setting power modes as defined in [Power Management](#).

Configuration Bytes MIRQ_0 and MIRQ_1

These two bytes define the default value of the volatile memory for [Mask Interrupt Register 0](#) and [Mask Interrupt Register 1](#) registers. These bytes can always be accessed for read and write via the SPI / I²C and can be accessed for read and write from the RF side if *rfcfg_en* is set to 1.

AS3955 Communication Modes

AS3955 supports three different modes. The basic communication mode is a standalone mode where AS3955 can behave as a standalone NFC tag without MCU intervention. The other two modes (Tunneling and Extended mode) represent modification of the communication in SELECTED state. The anti-collision process is the same for all three modes.

It is possible to change the mode of operation at any time.

In case Tunneling and Extended mode are both enabled, Tunneling mode has priority over Extended mode.

Standalone NFC Type 2 Tag Mode

If neither of the two modes are enabled (Tunneling and Extended mode) in [IC Configuration Register 2](#), the tag is in standalone mode. In this mode, all RF incoming commands address the internal EEPROM. In this mode, it is always possible to connect the MCU since all other functionalities of AS3955 are not limited by the communication modes. The main purpose of this mode is to use the AS3955 as a standalone chip or a chip in combination with MCU where the MCU is used for managing AS3955 configuration and memory content.

Tunneling Mode

Tunneling mode enables transparent data transfer between NFC device and MCU. In this mode, the internal EEPROM cannot be accessed via RF and any type of data received will be forwarded to MCU when the tag is in SELECTED state. An error during the reception will trigger a corresponding interrupt. In this mode the MCU shall take care for the correct response. For this purpose, the MCU may issue an ACK or a different type NAK response using implemented commands.

By enabling this mode, the MCU can emulate NFC type 2 tags, NFC type 4 tags, ISO14443A Level 3 cards, ISO14443A Level 4 cards, and also implement higher level protocols such as [\[PHDC\]](#).

Tunneling mode can be configured by setting [Configuration Byte IC_CFG2](#) in EEPROM or the corresponding register ([IC Configuration Register 2](#)).

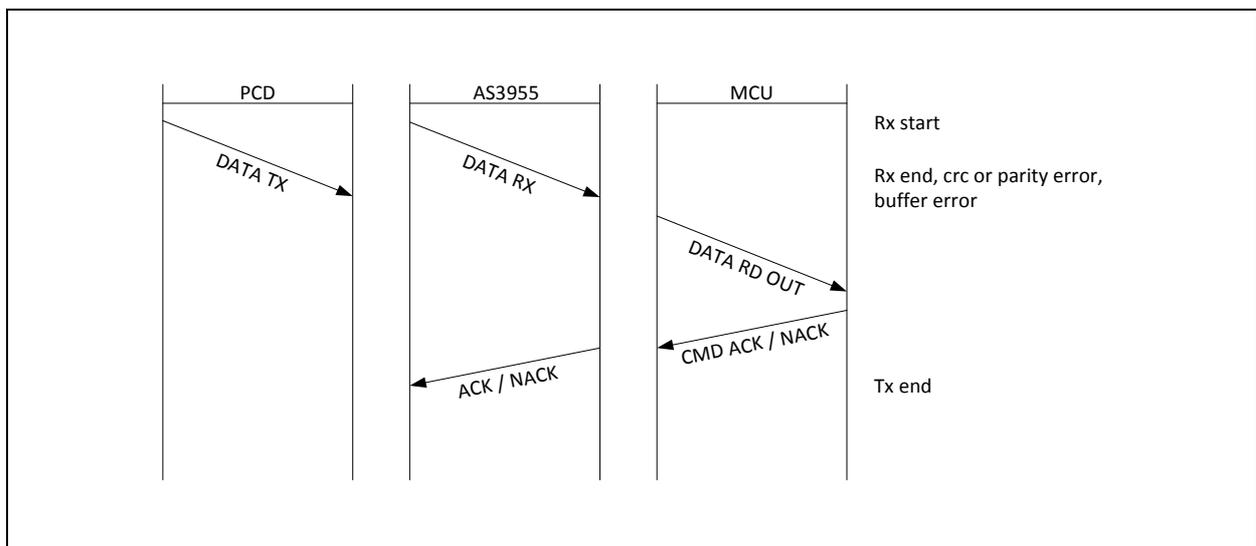
This mode also allows access to the internal EEPROM via the SPI / I²C during RF communication.

The basic assumption of Tunneling mode is that MCU is responsible for generating the response and that AS3955 takes care of the response synchronization over RF. For this reason, AS3955 has a possibility to transmit ACK and NACK by issuing a direct command (see section [Direct Commands](#)). The MCU can also transmit data by writing the data into the buffer and issuing a Transmit command. The implementation of Tunneling mode also requires the handling of the SLP_REQ command since all the received data in SELECTED state are stored in the buffer and will not be processed by AS3955. This also implies that MCU has to take care of correct transition of the tag into SLEEP or SENSE state by using one of the three available commands [Go To Sleep](#), [Go To Sense](#) and [Go To Sense / Sleep](#).

Data Transaction in Tunneling Mode

Figure 45 shows an example of how communication in Tunneling mode should be implemented.

Figure 45:
Read First, then Acknowledge



ACK and NACK responses can be replaced by data, in case of an incoming READ command.

Relevant Registers, Interrupts and Commands

Registers:

- [Buffer Status Register 1](#) and [Buffer Status Register 2](#) (data length and error type)

Interrupts:

- Rx start (*l_rxs* bit in [Interrupt Register 1](#))
- Rx end, Tx end (resp. *l_rxe* and *l_txe* bits in [Interrupt Register 0](#))
- CRC, parity and framing interrupt (resp. *l_crc_err*, *l_par_err* and *l_frm_err* bits in [Interrupt Register 1](#))
- Buffer error (*l_bf_err* in [Interrupt Register 1](#))
- SPI / I²C buffer access error (*l_acc_err* in [Interrupt Register 1](#))

Commands:

- Transmit ACK
- Transmit NACK 0-5
- Transmit Buffer
- Go To Sense
- Go To Sleep
- Go To Sense / Sleep

Extended Mode

Extended mode enables communication between the NFC device and MCU by employing standard NFC Tag 2 Type READ and WRITE commands. The purpose of this communication mode is to provide a simple data transfer mechanism between a NFC device and a MCU while guaranteeing correct timing and synchronization. This is achieved by implementing a robust handshake mechanism.

This mode uses a part of the memory address space that is out of range of the internal physical memory. The communication between a NFC device and a MCU can be performed by using WRITE/READ commands on address FCh – FFh, independently of the EEPROM configuration size. Data with CRC or any other error will not be forwarded to the MCU. In case of successful reception of data, AS3955 will automatically reply with ACK.

Error handling in Extended mode is defined in [Error Handling](#).

Data received from RF side are kept available until the MCU reads the data. The implemented asynchronous transmission protocol arbitrates on overlapping memory accesses (producer-consumer principle) and complies with timing constraints of both RF and SPI / I²C protocols regardless of the MCU performance.

Extended mode can be configured by setting [Configuration Byte IC_CFG2](#) in EEPROM or the corresponding register ([IC Configuration Register 2](#)).

Extended mode uses an address space above the address space of the internal EEPROM. It is then possible for a NFC device to perform accesses to AS3955 internal memory and data transfer to/from the MCU, without switching modes. This feature allows the NFC device e.g. to request to the MCU a switch to Tunneling mode with a simple WRITE command.

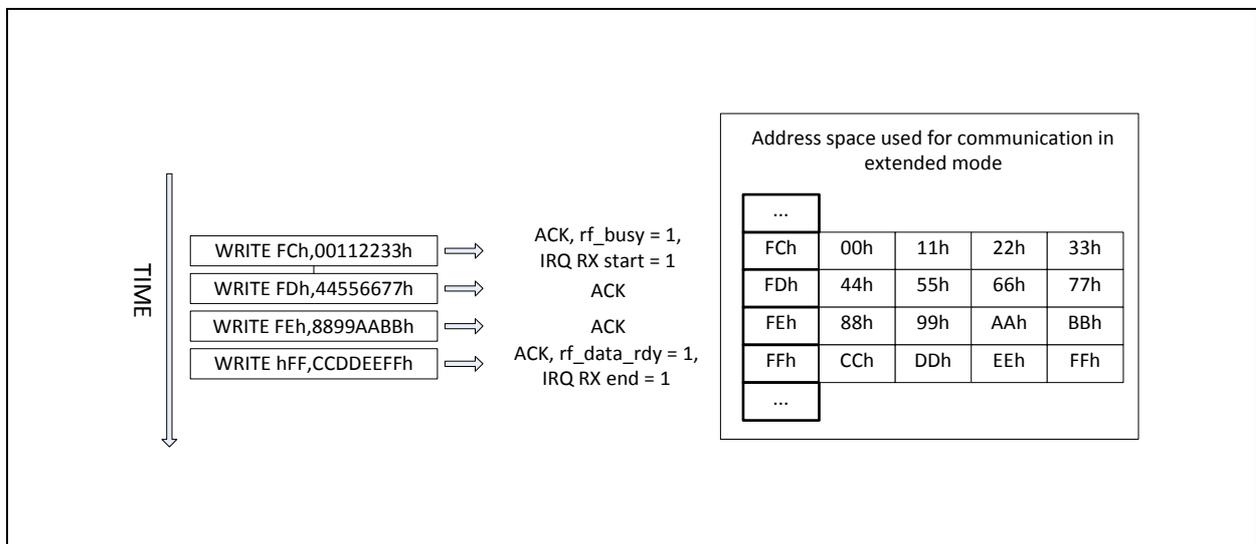
The extended mode communication employs a built-in buffer for communication. Access to buffer from RF and SPI / I²C is mutually exclusive. AS3955 ensures that buffer content shall be kept as long as the AS3955 is powered (even in case RF field is not present) and as long the tag is in SELECTED state. In case the RF field is switched off and then on again, the buffer content will be reset.

NFC Device to MCU Data Flow Protocol

For the data transmission from NFC device to the MCU employing the Extended mode, a NFC WRITE shall be used. Each data transfer from NFC device is comprised of four WRITE commands starting from address FCh and ending at address FFh. The protocol implemented on a NFC device is expected to always start the data transmission at address FCh, which signals the beginning of the communication, and end at address FFh.

Figure 46 depicts regular implementation of the Extended mode using the WRITE commands.

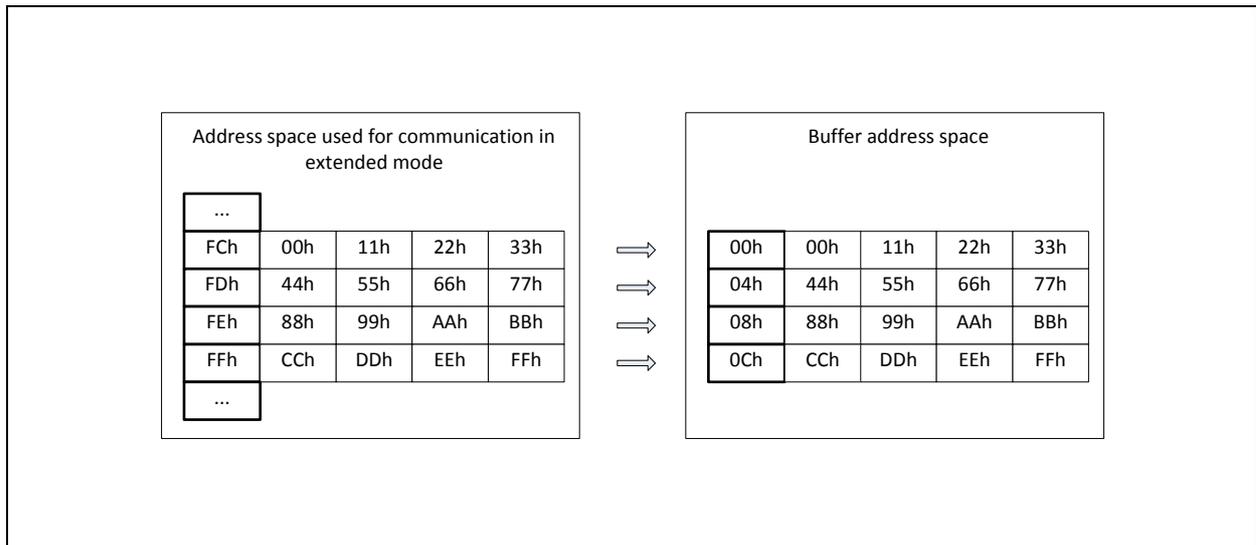
Figure 46:
Address Space Employed for Communication



Assuming that the internal buffer is empty, the NFC device may start with data transmission by sending a WRITE command on address FCh. When the first block is written, the *rf_busy* flag is set and *Rx_start* interrupt is triggered. At this point, the buffer cannot be accessed over SPI / I²C until the entire write message is received which is assumed to be complete when NFC device sends a write command to block FFh. At this point, an Rx end interrupt is triggered. This implies that a minimum two messages must be received from a NFC device in order to successfully complete a message. If a WRITE command is received on address FFh before a WRITE command to address FCh, AS3955 will assume that an error has occurred and will respond as described in [Error Handling in Extended Mode](#) .

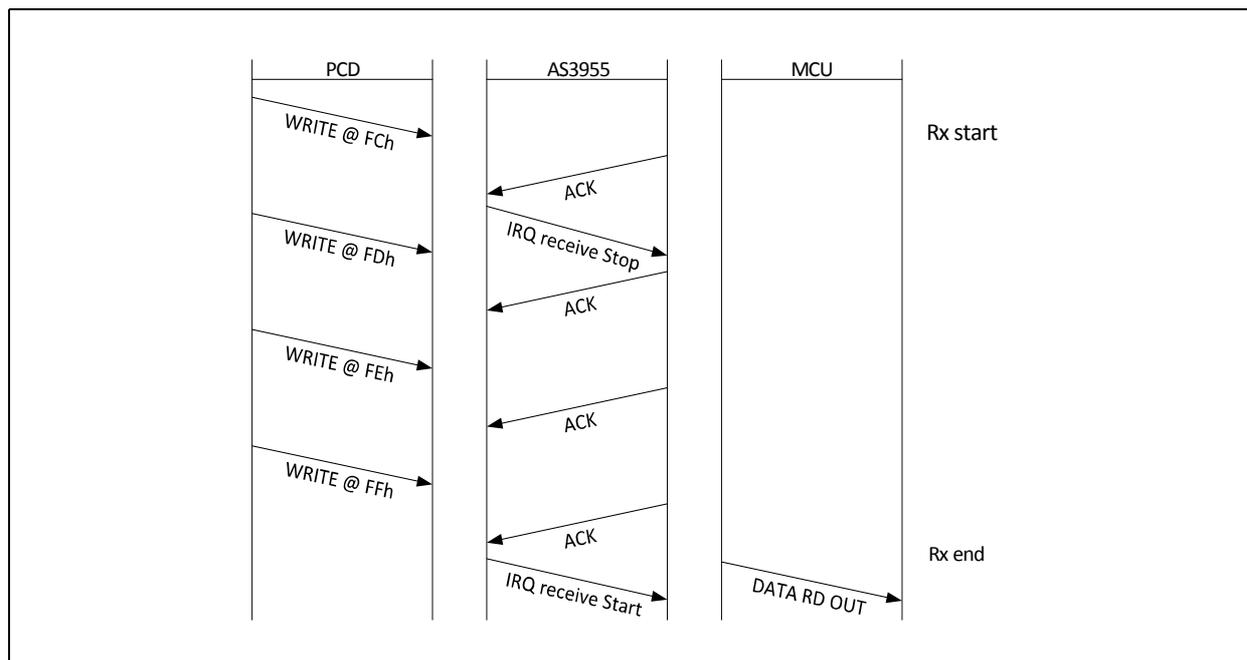
When the reader writes into the last block FFh, *rf_data_rdy* flag is set and the reader cannot change the buffer content until the MCU reads the content of the block and clears the *rf_data_rdy* flag. Any additional reception of WRITE commands from NFC device, prior to MCU reading out the buffer content, shall result in a response as defined in [Error Handling](#). The blocks from internal memory address space are directly mapped into buffer space as shown on [Figure 47](#).

Figure 47:
Mapping of Data Received into Buffer



When MCU reads out the data from the buffer, data will be sent to MCU in the same order as they were stored in the buffer starting from address 00h. When MCU has read all buffer content, it shall issue a [Clear Buffer](#) command to clear the flag *rf_data_rdy*. At this point, a new data message can be received.

Figure 48:
Data Reception in Extended Mode



MCU to NFC Device Data Flow Protocol

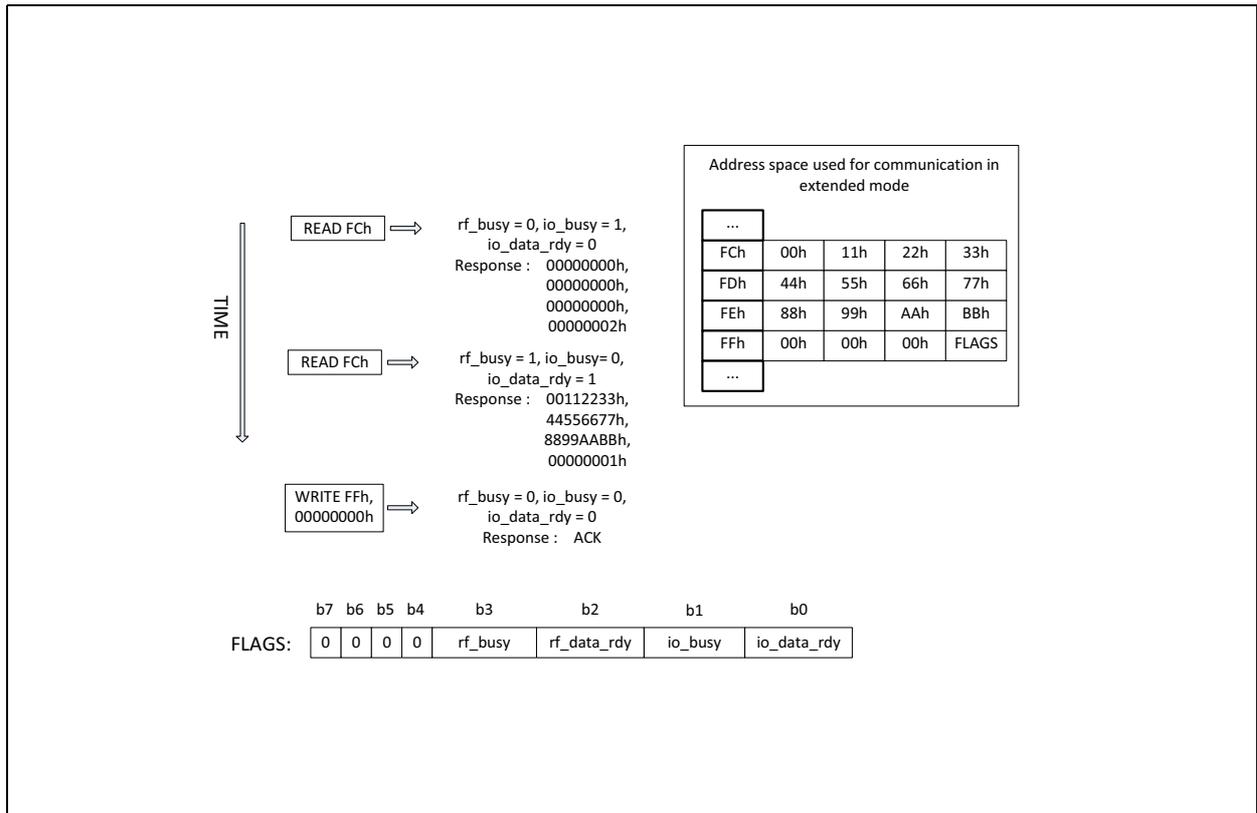
NFC devices can receive data from a MCU through AS3955 by using standard READ commands.

Prior to writing data into the buffer, the MCU shall issue a [Clear Buffer](#) command to AS3955 to ensure data are correctly mapped into the buffer. To start data transmission, the MCU shall issue then a [Transmit Buffer](#) command. In this way, the flag *io_data_rdy* is set and data can be transmitted to the reader. The transmission of the data from the tag to the NFC device is done by issuing a READ command on block FCh. The fourth block being read during the read command contains the status flags of the ongoing communication.

NFC devices can receive data from a MCU through AS3955 by using standard READ commands.

Prior to writing data into the buffer, it is advisable that to issue a [Clear Buffer](#) command to AS3955 to ensure data are correctly mapped into the buffer. MCU can trigger data transmission by writing three words of data into the buffer at addresses FCh-FEh, and issuing a [Transmit Buffer](#) command. This will also set *io_data_rdy* flag. The transmission of the data from the tag to the read is done by issuing a read command on word 0xFC. The NFC device can then retrieve the data from the buffer by sending a READ command to address FCh. The fourth word contains the status flags as shown in [Figure 49](#).

Figure 49:
Data Transmission from MCU to NFC Device



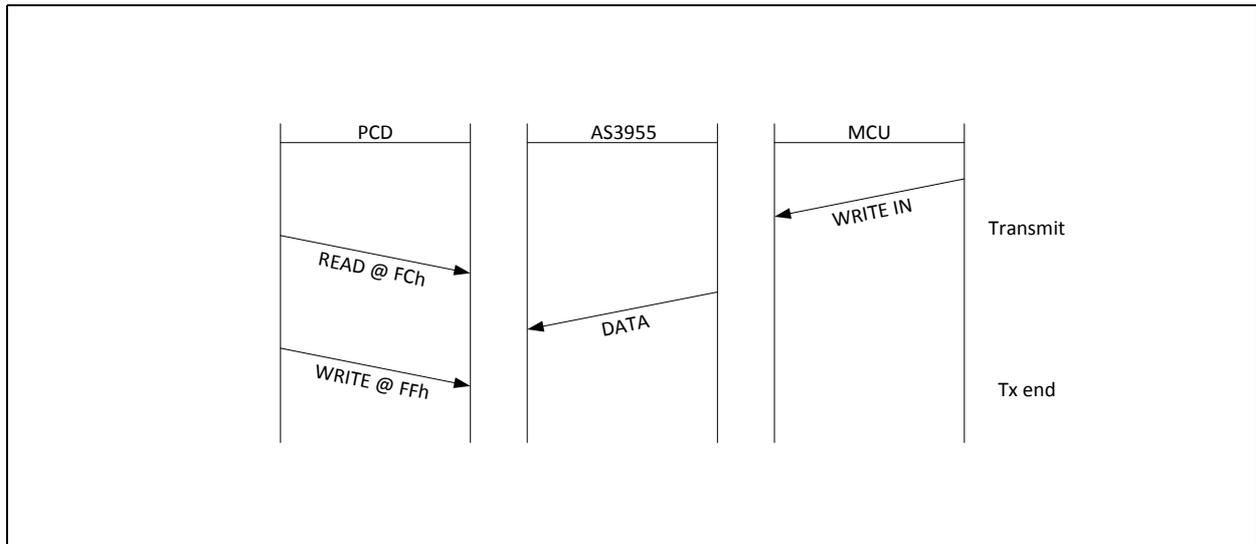
When the NFC device has successfully read all data from the buffer, it shall clear the buffer and prepare for further data transfer by issuing a WRITE command to address FFh. This will trigger a *Tx end* interrupt.

If a READ command is received on address FCh prior to the **Transmit Buffer** command, AS3955 will return zero data, *io_busy* set to 1 and *io_data_rdy* set to 0. A NFC device can then poll AS3955 by continuously sending READ commands at address FCh and waiting until the last word equals 01h.

A READ command can be issued only to address FCh. Issuing a READ command to any other address than FCh while in Extended mode shall be treated as an error.

In Extended mode, it is assumed that AS3955 will always receive 16 bytes and transmit 12 bytes of data, 3 empty bytes and 1 byte for FLAGS. If the message size differs in any direction, then the MCU or NFC device are responsible for proper error management.

Figure 50:
Data Transmission in Extended Mode



If MCU decides to update the buffer before the NFC device issues a WRITE command to address FFh, a *I_acc_err* interrupt will be triggered, signaling that buffer cannot be accessed (see [Interrupt Register 1](#)).

Relevant Registers, Interrupts and Commands

Relevant Registers:

- [Buffer Status Register 1](#) (status flags: *rf_busy*, *rf_data_rdy*, *io_data_rdy*)

Relevant Interrupts:

- *Rx start* (*I_rxs* bit in [Interrupt Register 1](#))
- *Rx end*, *Tx end* (resp. *I_rxe* and *I_txe* bits in [Interrupt Register 0](#))

Commands:

- [Transmit Buffer](#)
- [Clear Buffer](#)

Implementation Recommendations

While in Extended mode, the NFC device shall not issue a WRITE command to the address space of the internal EEPROM until the buffer is empty. The NFC device can issue a READ command to check internal status flag (io_busy = 0), before writing to EEPROM.

It is also not allowed to interleave data communication in Extended mode (e.g. writing to addresses FCh-FFh) and issue a WRITE command to internal EEPROM. If case this happens, writing to internal EEPROM would be successful, but the buffer content would be lost.

Reading status flags is always allowed.

Error Handling

Figure 53 shows how different type of errors in Standalone and Extended communication modes are handled and what is the resulting state of the tag.

Figure 53:
Error Handling

| Command | Condition | AS3955 nak_on_crc_parity=0 | AS3955 nak_on_crc_parity=1 (empty fields have same reply as nak_on_crc_parity=0) |
|----------|---|---|--|
| Any | Bit Coding Error | No Response, SLEEP | |
| Any | Incomplete Frame | No Response, SLEEP | |
| T2T Read | Parity Error | No Response, SLEEP | NAK_1, SLEEP |
| T2T Read | CRC Error | No Response, SLEEP | NAK_1, SLEEP |
| T2T Read | 1 Byte Frame (no CRC) | No Response, SLEEP | |
| T2T Read | Empty frame (only CRC bytes) | No Response, SLEEP | |
| T2T Read | Missing address | No Response, SLEEP | |
| T2T Read | Too long frame | No Response, SLEEP | |
| T2T Read | Memory fully locked | Send Block Content | |
| T2T Read | Memory partially locked | Send Block Content | |
| T2T Read | Memory fully protected | NAK_4, SLEEP | |
| T2T Read | Memory partially protected | Send unprotected memory as is, replace protected memory with zero bytes | |
| T2T Read | Memory address range fully not existent | NAK_0, SLEEP | |

| Command | Condition | AS3955 nak_on_crc_parity=0 | AS3955 nak_on_crc_parity=1 (empty fields have same reply as nak_on_crc_parity=0) |
|---------------------|---|---|--|
| T2T Read | Memory address range partially not existent | Send existing memory appended with zero bytes | |
| T2T Read | EEPROM access collision | NAK_5, SLEEP | |
| T2T Read | No Error | Send Block Content | |
| T2T Write | Parity Error | No Response, SLEEP | NAK_1, SLEEP |
| T2T Write | CRC Error | No Response, SLEEP | NAK_1, SLEEP |
| T2T Write | 1 Byte Frame (no CRC) | No Response, SLEEP | |
| T2T Write | 2 Byte Frame with correct CRC | No Response, SLEEP | |
| T2T Write | Missing address and data | No Response, SLEEP | |
| T2T Write | Missing data | No Response, SLEEP | |
| T2T Write | Incomplete data | No Response, SLEEP | |
| T2T Write | Too long frame | No Response, SLEEP | |
| T2T Write | Memory fully locked | NAK_0, SLEEP | |
| T2T Write | Memory partially locked | cannot occur | |
| T2T Write | Memory fully protected | NAK_4, SLEEP | |
| T2T Write | Memory partially protected | cannot occur | |
| T2T Write | Memory address range fully not existent | NAK_0, SLEEP | |
| T2T Write | Memory address range partially not existent | cannot occur | |
| T2T Write | EEPROM access collision | NAK_5, SLEEP | |
| T2T Write | No Error | ACK, SELECTED | |
| T2T Sector Select 1 | Parity Error | No Response, SLEEP | NAK_1, SLEEP |
| T2T Sector Select 1 | CRC Error | No Response, SLEEP | NAK_1, SLEEP |
| T2T Sector Select 1 | 1 Byte Frame (no CRC) | No Response, SLEEP | |
| T2T Sector Select 1 | Empty frame (only CRC bytes) | No Response, SLEEP | |

| Command | Condition | AS3955 nak_on_crc_parity=0 | AS3955 nak_on_crc_parity=1 (empty fields have same reply as nak_on_crc_parity=0) |
|----------------------|--|-------------------------------|--|
| T2T Sector Select 1 | Missing second byte | No Response, SLEEP | |
| T2T Sector Select 1 | Incorrect second byte (not FFh) | No Response, SLEEP | |
| T2T Sector Select 1 | Too long frame | No Response, SLEEP | |
| T2T Sector Select 1 | Only 1 sector available | NAK_0, SLEEP | |
| T2T Sector Select 2 | Parity Error | No Response, SLEEP | NAK_1, SLEEP |
| T2T Sector Select 2 | CRC Error | No Response, SLEEP | NAK_1, SLEEP |
| T2T Sector Select 2 | 1 Byte Frame | No Response, SLEEP | |
| T2T Sector Select 2 | Empty frame (only CRC bytes) | No Response, SLEEP | |
| T2T Sector Select 2 | Missing sector number | No Response, SLEEP | |
| T2T Sector Select 2 | Missing RFU bytes | No Response, SLEEP | |
| T2T Sector Select 2 | Too few RFU bytes | No Response, SLEEP | |
| T2T Sector Select 2 | Too long frame | No Response, SLEEP | |
| T2T Sector Select 2 | Selected sector not existent (cannot happen for AS3955) | No Response, SLEEP | |
| T2T Sector Select 2 | No Error (cannot occur for AS3955) | No Response, SLEEP | |
| Unknown Command Code | Parity Error | No Response, SLEEP | NAK_1, SLEEP |
| Unknown Command Code | CRC Error | No Response, SLEEP | NAK_1, SLEEP |

| Command | Condition | AS3955 nak_on_crc_parity=0 | AS3955 nak_on_crc_parity=1 (empty fields have same reply as nak_on_crc_parity=0) |
|----------------------------|-----------|-------------------------------|--|
| Unknown Command Code | No Error | No Response, SLEEP | |

Figure 54:
Error Handling in Extended Mode

| Command | Condition | Error Type |
|---------------------------------------|--|-------------|
| Write buffer (T2T Write) | Start with FD, FE or FF | NAK0, SLEEP |
| Write buffer (T2T Write) | Write when io_data_rdy=1 or rf_data_rdy=1 | NAK0, SLEEP |
| Clear buffer (T2T Write on block FFh) | Write none zero data | NAK0, SLEEP |
| Read buffer (T2T read on block FCh) | Read on FD, FE, FF | NAK0, SLEEP |
| Read buffer (T2T read on block FCh) | Buffer access collision | NAK5, SLEEP |
| Write buffer (T2T Write) | Buffer access collision | NAK5, SLEEP |
| Clear buffer (T2T Write on block FFh) | Buffer access collision | NAK5, SLEEP |

Wired Interfaces

AS3955 host interface can be configured to be either SPI or I²C at production ([Fabrication Data FAB_CFG0](#)). In both cases, the /SS signal is also used to control the IC power state. By pulling the /SS low, the chip interface or the chip itself is enabled / powered (see [Power Management](#)). Note that interrupting SPI / I²C operations or issuing incomplete command sequence from the MCU may result in corrupted data content. For more information on EEPROM and buffer data reading and writing see [EEPROM Read and Write](#) and [Data Buffer](#) sections.

SPI / I²C Access Modes

Figure 55:
Access Modes

| Mode | MODE Byte (com. bits) | | | | | | | | MODE Related Data | |
|----------------|-----------------------|----|----|----|---------|----|----|----|--------------------|-----------------------|
| | MODE | | | | Trailer | | | | | |
| | M2 | M1 | M0 | C4 | C3 | C2 | C1 | C0 | | |
| Register Write | 0 | 0 | 0 | A4 | A3 | A2 | A1 | A0 | Data byte(s) | |
| Register Read | 0 | 0 | 1 | A4 | A3 | A2 | A1 | A0 | Data byte(s) | |
| EEPROM Write | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | Block Address byte | 4 bytes of block data |
| EEPROM Read | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | Block Address byte | N*4 bytes |
| Buffer Load | 1 | 0 | 0 | x | x | x | x | x | Data byte(s) | |
| Buffer Read | 1 | 0 | 1 | x | x | x | x | x | Data byte(s) | |
| COMMAND Mode | 1 | 1 | C5 | C4 | C3 | C2 | C1 | C0 | | |

SPI Interface

Communication between AS3955 and microcontroller can be done via a four-wire Serial Peripheral Interface (SPI) and an additional interrupt signal. AS3955 acts an SPI slave device, and it can request MCU attention by sending an interrupt (pin IRQ). The SCLK frequency can be between 100 kHz and 5 MHz.

Figure 56:
SPI and Interrupt Signals

| Name | Signal | Signal Level | Description |
|------|------------------------------|--------------|------------------------------------|
| /SS | Digital Input with pull up | CMOS | SPI enable (active low) |
| MOSI | Digital Input | CMOS | Serial data input |
| MISO | Digital Output with tristate | CMOS | Serial data output |
| SCLK | Digital Input | CMOS | Clock for serial communication |
| IRQ | Digital Output | CMOS | Interrupt Output pin (active high) |

SPI interface is in reset mode when signal /SS is high, and it is enabled when /SS is low. It is recommended to keep signal /SS high whenever the SPI interface is not used. MOSI is sampled at the falling edge of SCLK. All communication is done in 8-bit blocks (bytes). First three bits of first byte transmitted after /SS high to low transition define SPI operation mode. MSB bit is always transmitted first (valid for address and data). Read and Write modes support address auto incrementing, which means that in case some additional data bytes may be sent (read), they are written to (read from) addresses incremented by 1 after the address and first data byte.

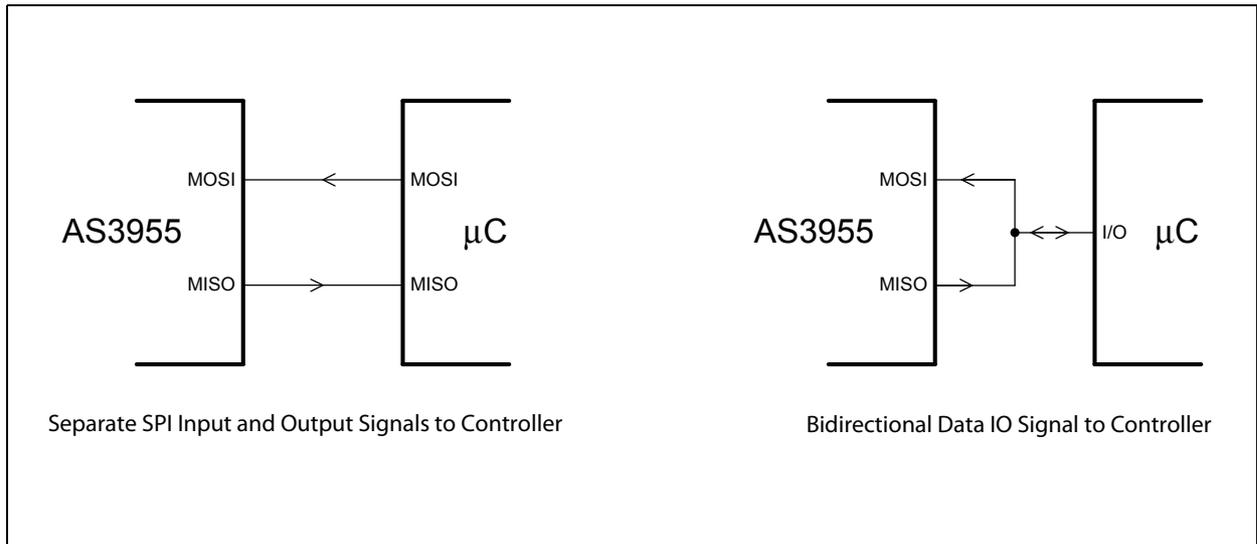
SPI interface supports the following modes:

- internal registers read and write
- EEPROM read and write
- buffer read and write
- direct commands

Note that in case when logic and EEPROM are supplied from the VP_IO, the only SPI operations permitted are reading and writing of EEPROM and registers (see also [Power Management](#)).

MISO output is usually in tristate and it is only driven when output data is available. MOSI and MISO can then be externally shorted to create a bidirectional signal. When MISO output is in tristate, it is possible to switch on a 10 kΩ pull down by activating option bits *miso_pd1* and *miso_pd2* in [IO Configuration Register](#).

Figure 57:
IO Signals to Controller



Writing Data to Addressable Registers (Register Write Mode)

Following figures show waveforms of writing a single byte and writing multiple bytes with auto-incrementing address. After the SPI operation mode bits, the address of starting register to be written is provided. Then one or more data bytes are transferred from the SPI, always MSB first. The data byte is written in register on falling edge of its last clock. In case the communication is terminated by putting /SS high before a packet of 8 bits composing one byte is sent, writing of this register is not performed. In case the register on the defined address does not exist or it is a read-only register, no write is performed.

Figure 58:
SPI Communication: Writing a Single Register

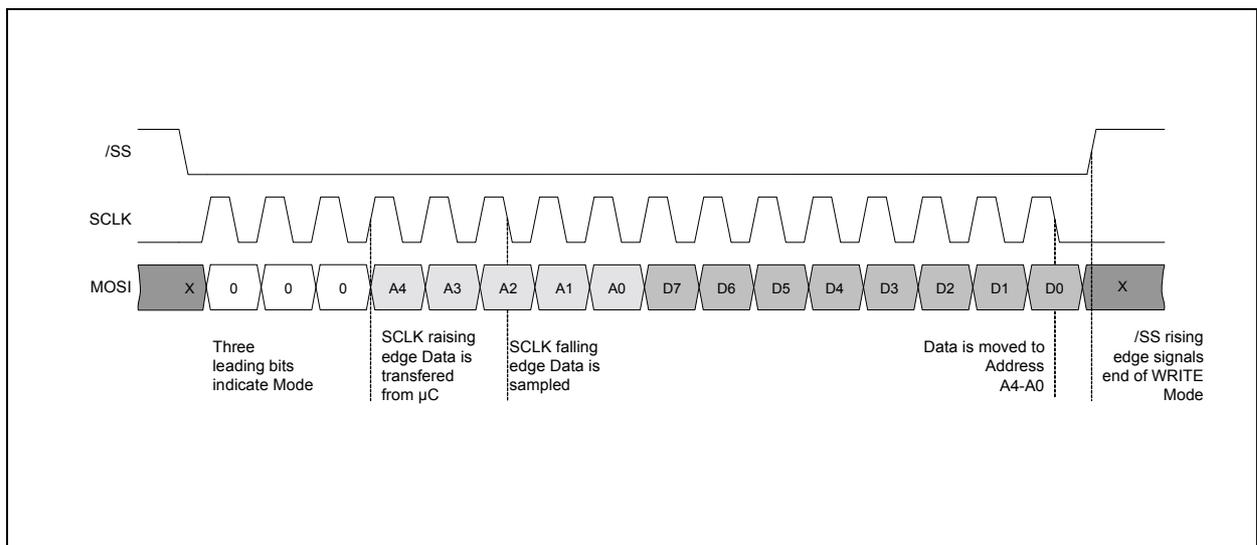
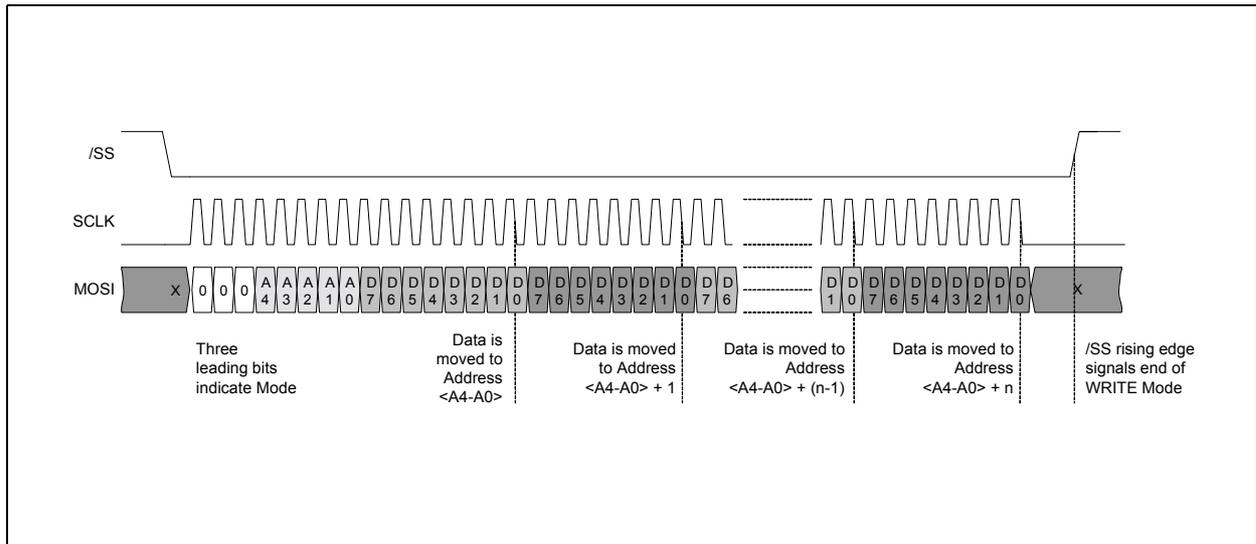


Figure 59:
SPI Communication: Writing Register Data with Auto-Incrementing Address



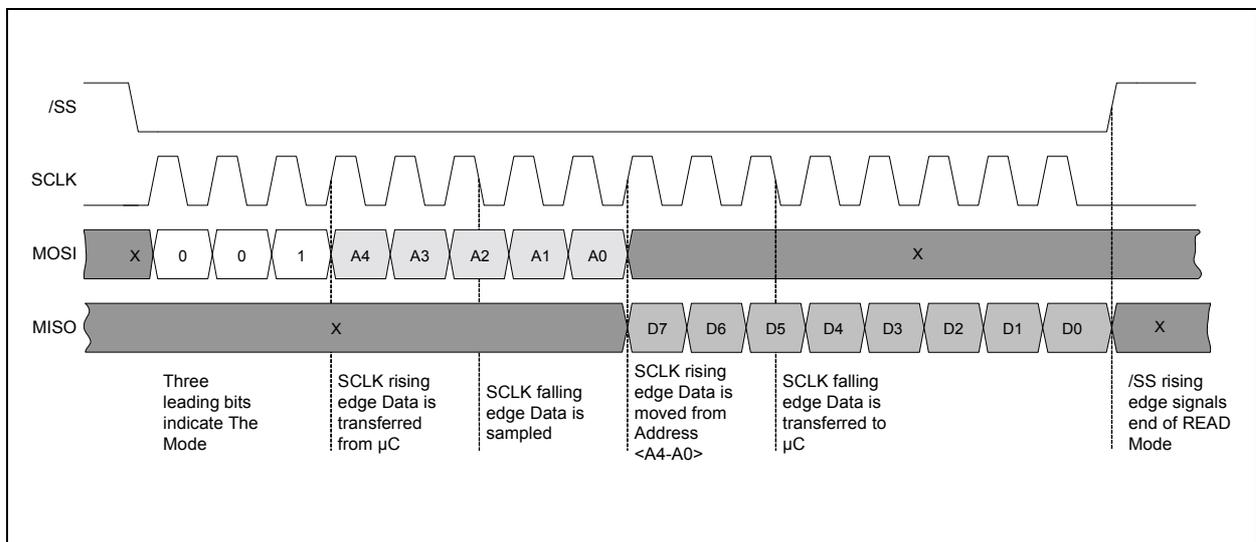
Reading Data from Addressable Registers (Register Read Mode)

After the SPI operation mode bits, the register address to be read shall be provided, MSB first. Then one or more data bytes are transferred to MISO output, always MSB first. As in case of the write mode, also the read mode supports auto-incrementing addressing.

MOSI is sampled at the falling SCLK edge (as shown in the following diagrams); data to be read from AS3955 internal register is driven to MISO pin on rising edge of SCLK and is sampled by the master at the falling SCLK edge.

In case the register on defined address does not exist, all 0 data is sent to MISO. In the following figure an example of reading of a single byte is given.

Figure 60:
SPI Communication: Reading a Single Register



Writing and Reading of EEPROM Through SPI

EEPROM data can be read and written also through SPI interface. Due to possible conflict with the RFID interface additional arbitration bit can be set (IC_CFG0) in order to set the priority access to the EEPROM access. If EEPROM write operation is terminated due to higher priority of the RF an I_acc_err (Interrupt Register 1) IRQ is sent. The description is referring to following case when an EEPROM write operation is triggered via SPI / I²C and bit arbit_mod (EEPROM, IC_CFG0, bit3) is set to 1 which means that RF has priority over SPI when accessing EEPROM. In this case EEPROM via SPI / I²C write will be terminated and IRQ I_acc_err is triggered if tags enters RF field and EEPROM is needed for RF initialization, EPROM read or write is issued via RF.

Word Address Byte

Both EEPROM modes (Read and Write) use Word Address byte to define the address of the EEPROM word which is accessed. Seven MSB bits of the Address Byte are used to define the address; while the last bit is don't care (it is used to synchronize EEPROM access).

Figure 61:
EEPROM Block Address Byte

| | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|----------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| EEPROM Block Address | WA6 | WA5 | WA4 | WA3 | WA2 | WA1 | WA0 | x |

EEPROM Write

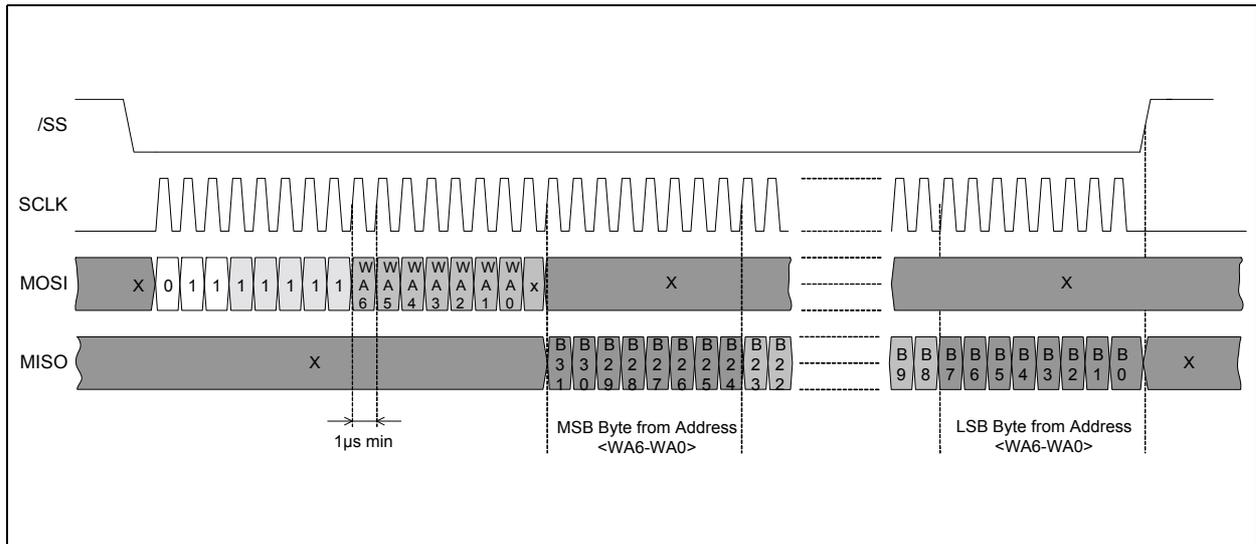
In order to program an EEPROM block, six bytes shall be sent (mode byte, block address byte and 4 bytes of data, all of them MSB first). Actual EEPROM programming is started with rising /SS edge signal which terminates the EEPROM Write command.

EEPROM Read

In order to read data from EEPROM, first a mode byte is sent, followed by the block address byte (MSB first). Then one or more blocks of data with address auto-incrementing (packets of 4 bytes) are transferred to MISO output, also MSB first. MOSI is sampled at the SCLK falling edge; data to be read from AS3955 EEPROM is driven to MISO pin on SCLK rising edge and is sampled by the master at the SCLK falling edge. In case the block on the defined address does not exist, all 0 data is sent to MISO.

Please note that SCLK frequency should not exceed 1MHz during EEPROM Read (limited by EEPROM read access time).

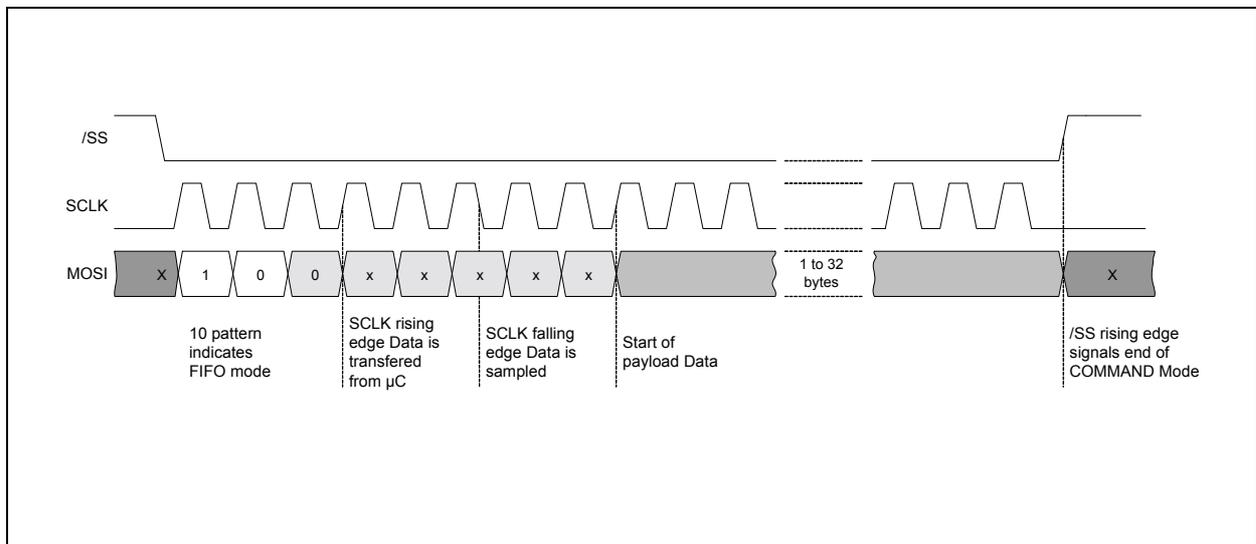
Figure 62:
Reading an EEPROM Block over SPI



Loading Transmission Data into Buffer

Loading the transmitting data into the buffer is similar to writing data into an addressable registers. Difference is that in case of loading more bytes all bytes go to the buffer. The command mode code 100b indicates buffer write operation. A bit stream of 32 bytes of data can be transferred. The following figure shows how to load the transmission data into the buffer.

Figure 63:
Loading Data into Buffer over SPI



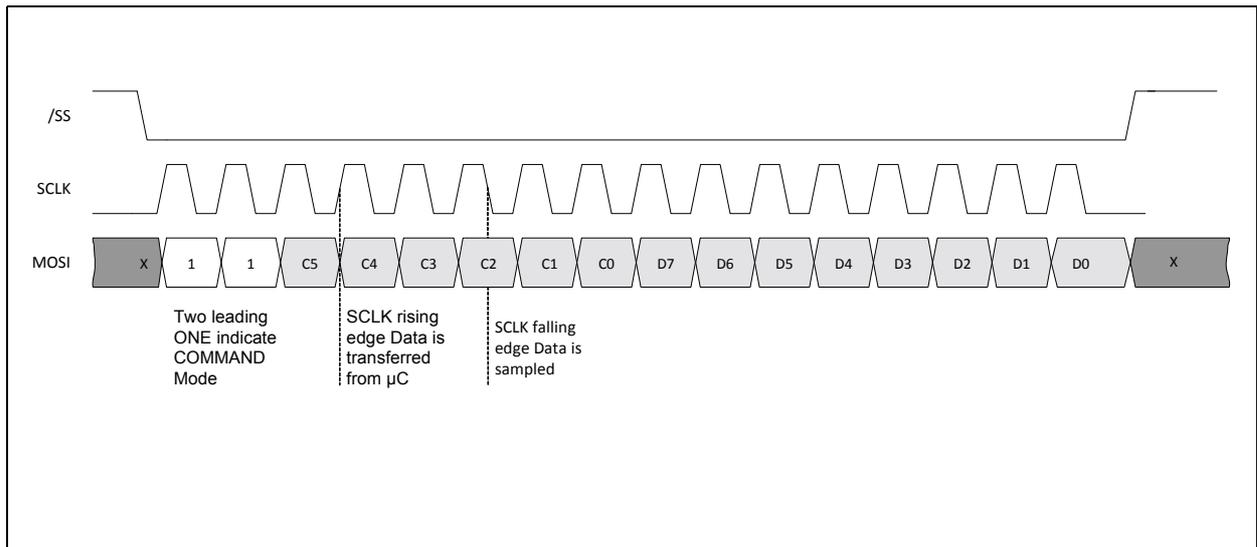
Reading Received Data from Buffer

Reading received data from the buffer is similar to reading data from an addressable registers. Difference is that, in case of reading more bytes, they all come from the buffer. The command mode code 101b indicates buffer operations. In case the command is terminated by putting /SS high before a packet of 8 bits composing one byte is read that particular byte is considered read.

Direct Command Mode

Direct Command Mode is comprised of one command byte followed by argument byte. SPI operation mode bits 11b indicate Direct Command Mode. The following six bits define command code, sent MSB first. Last two bits in argument byte indicate success of the direct command. Value 01h of the argument byte indicates that command was accepted, while value 02h indicates rejected due to internal access priorities. The argument byte does not provide information on timing of the command execution.

Figure 64:
Sending a Direct Command Over SPI



SPI Timing

Figure 65:
SPI Timing

| Symbol | Parameter | Min | Typ | Max | Unit | Note |
|---|----------------------------------|-----|-----|-----|------|------------------|
| General timing (VDD=VDD_IO=VDD_D= 3.3V, Temperature 25°C) | | | | | | |
| T _{SCLK} | SCLK period | 200 | | | ns | (1) |
| T _{SCLKL} | SCLK low | 80 | | | ns | |
| T _{SCLKH} | SCLK high | 80 | | | ns | |
| T _{SSH} | SPI reset (/SS high) | 50 | | | ns | |
| T _{NCSL} | /SS falling to SCLK rising | 25 | | | ns | first SCLK pulse |
| T _{NCSH} | SCLK falling to /SS rising | 80 | | | ns | last SCLK pulse |
| T _{DIS} | Data in setup time | 10 | | | ns | |
| T _{DIH} | Data in hold time | 10 | | | ns | |
| Read timing (VDD=VDD_IO=VDD_D= 3.3V, Temperature 25°C, Cloud ≤ 50pF) | | | | | | |
| T _{DOD} | Data out delay | | 20 | | ns | |
| T _{DOHZ} | Data out to high impedance delay | | 20 | | ns | |

Note(s):

1. T_{SCLK}=T_{SCLKL}+T_{SCLKH}, during EEPROM read the SCLK period has to be increased to 1μs (this limitation is imposed by EEPROM read access time)

Figure 66:
SPI General Timing

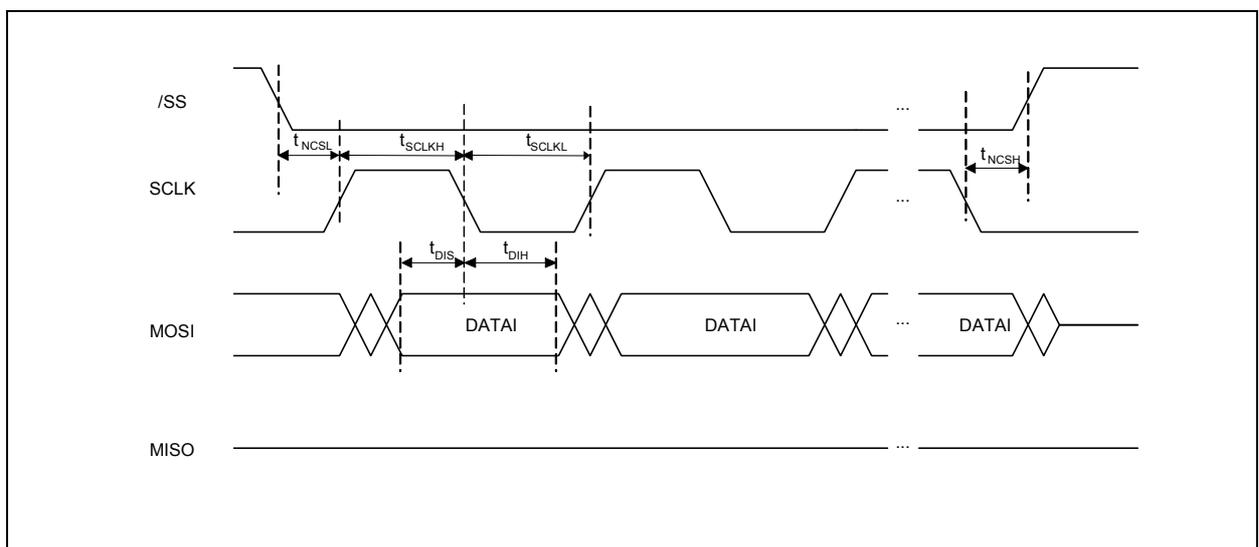
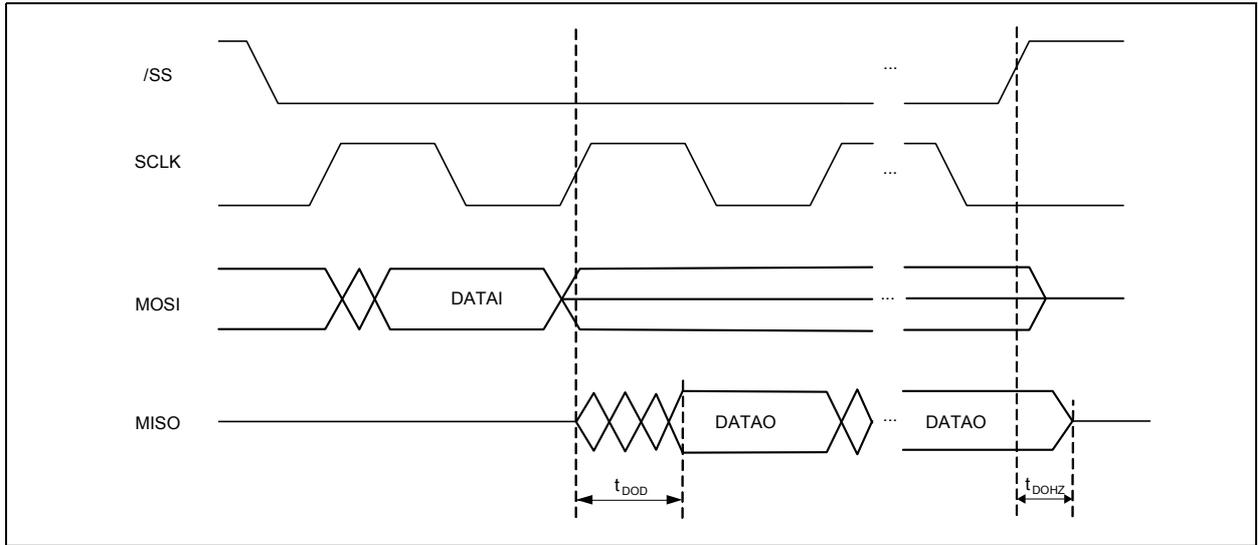


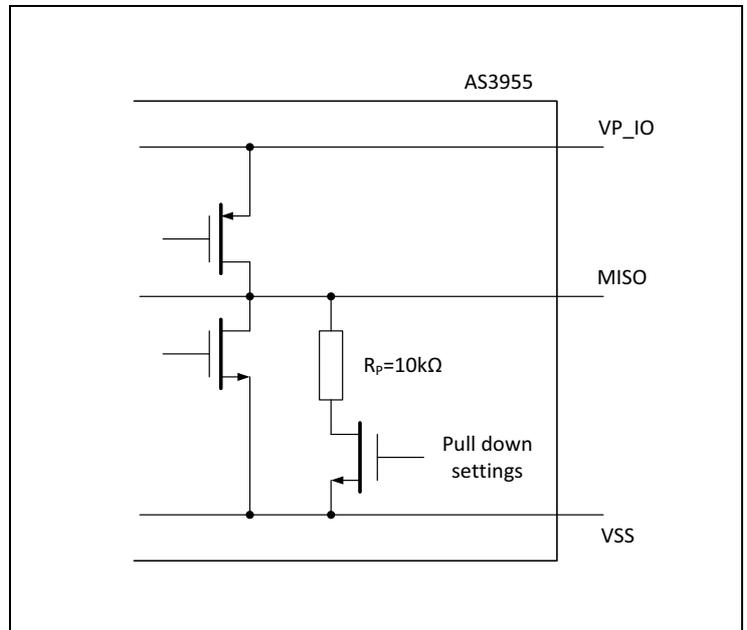
Figure 67:
SPI Read Timing



SPI Electrical Connection

A pull-down resistor can be connected by setting miso_pd1 and miso_pd2 bits in [IO Configuration Register](#).

Figure 68:
SPI Electrical Connection



I²C Interface

Communication between AS3955 and microcontroller can be done via a I²C interface and an additional interrupt signal. AS3955 acts an I²C slave device, and supports single master, multiple slave configurations. AS3955 I²C supports following modes: Standard-mode, Fast-mode, Fast-mode Plus.

Figure 69:
I²C and Interrupt Signals

| Name | Signal | Signal Level | Description |
|------|----------------------------|--------------|---|
| /SS | Digital Input with pull up | CMOS | Should be set to high during I ² C communication |
| SDA | Digital Output | CMOS | Serial Data output |
| SCL | Digital Input | CMOS | Clock for serial communication |
| IRQ | Digital Output | CMOS | Interrupt Output pin (active high) |

During I²C communication, the signal /SS should be set to low. By setting the /SS to low, I²C interface is enabled. It is recommended to keep signal /SS high whenever the I²C interface is not used.

I²C interface supports the following modes:

- Internal registers read and write
- EEPROM read and write
- Buffer read and write
- Direct commands

Please note that the only I²C operations allowed, when logic and EEPROM are supplied from VP_IO, are EEPROM and registers reading and writing (see also [Power Management](#)).

I²C Slave Address

The I²C slave address is composed of 7 bits where the first 4 MSBs are fixed and the last 3 LSBs are programmable.

Figure 70:
I²C Slave Address

| | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|--------------------------------|----|----|----|----|----|----|----|
| I ² C Slave Address | 1 | 0 | 1 | 0 | A2 | A1 | A0 |

The value of bits A2:A0 can be set in the Configuration Byte [IC_CFG0](#).

Writing Data to Addressable Registers (Register Write Mode)

Following figures show cases of writing a single byte and writing multiple bytes with auto-incrementing address. After the I²C slave address, the initial register address follows. Then one or more data bytes are transferred from the I²C, MSB first. The data byte is written in register on falling edge of its last clock.

Figure 71:
Writing a Single Register over I²C

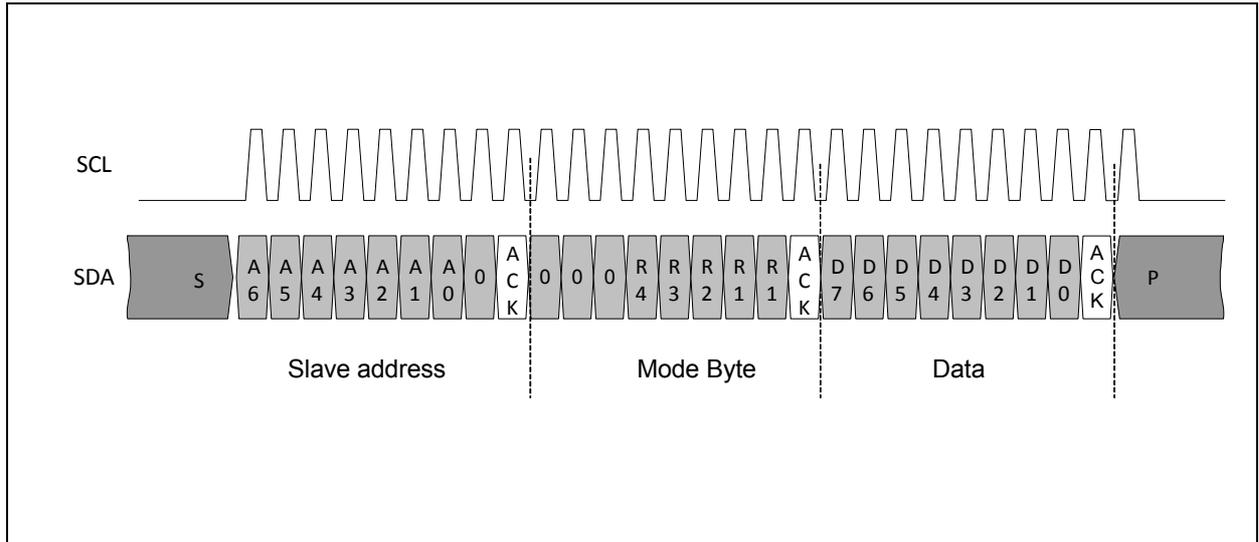
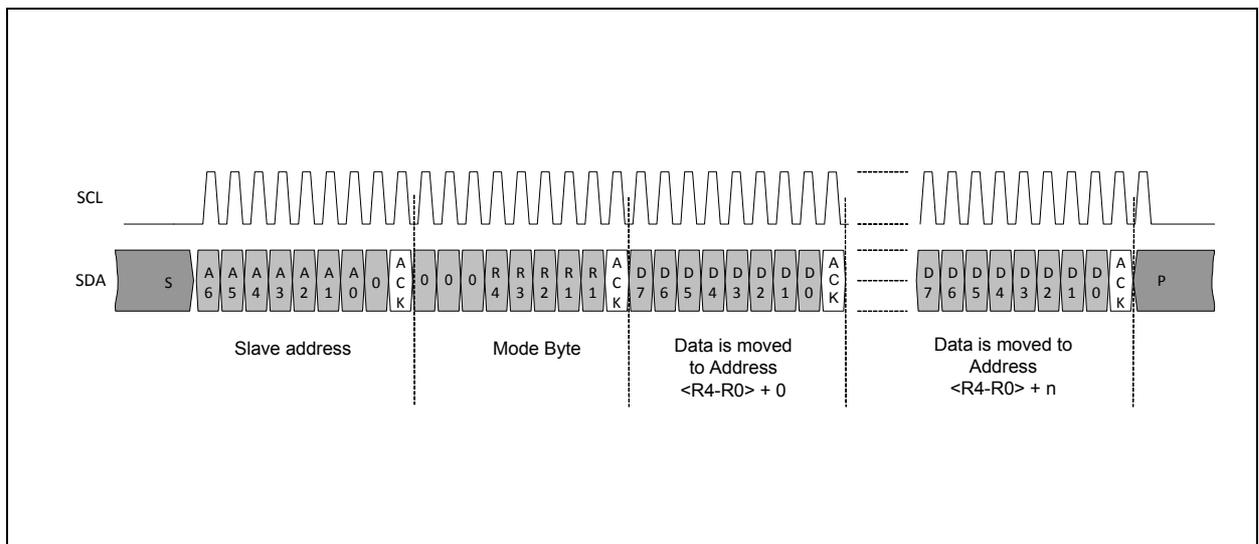


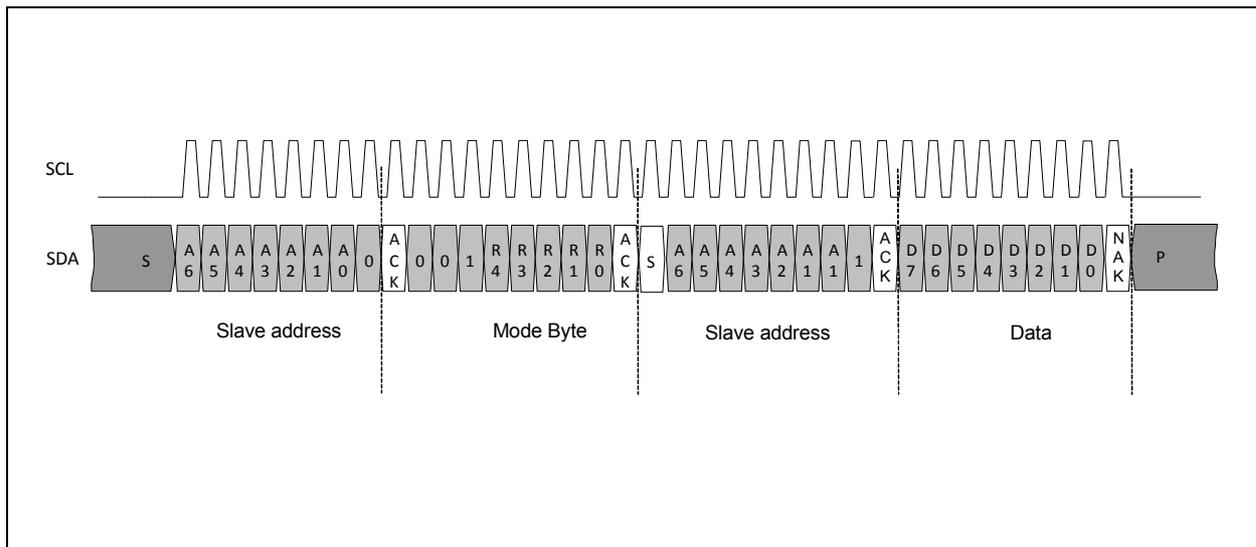
Figure 72:
Writing Register Data with Auto-Incrementing Address over I²C



Reading Data from Addressable Registers (Register Read Mode)

After the I²C slave address, the address of register to be read shall be provided, MSB first. Then one or more data bytes are transferred to SDA output, also MSB first. As in case of the write mode, also read mode supports auto-incrementing address. In case the register at the defined address does not exist, all 0 data is sent to SDA. In the following figure, an example for reading of single byte is given.

Figure 73:
Reading a Single Register over I²C



Writing and Reading EEPROM through I²C

EEPROM data can be read and written through I²C interface. Due to possible conflict with RFID interface, additional arbitration bit can be set (`IC_CFG0`) in order to set the priority access to the EEPROM access. If EEPROM write operation is terminated due to higher priority of the RF an `I_spi` ([Interrupt Register 1](#)), an IRQ is sent.

Block Address Byte

Both EEPROM Read and Write use Block Address byte to define the EEPROM block address to be accessed. Seven MSB bits of the Address Byte are used to define the address; while the last bit is don't-care (it is used to synchronize EEPROM access).

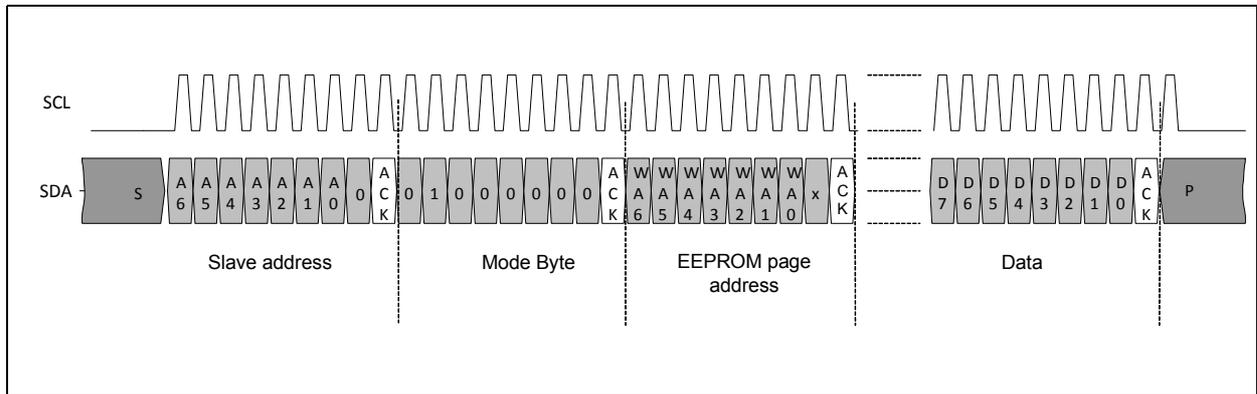
Figure 74:
EEPROM Block Address Byte over I²C

| | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|----------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| EEPROM Block Address | WA6 | WA5 | WA4 | WA3 | WA2 | WA1 | WA0 | x |

EEPROM Write

In order to program an EEPROM block, seven bytes shall be sent (slave address, mode byte, block address byte and 4 bytes of data, all of them MSB first). Actual EEPROM programming is started with rising edge of ACK of the last byte.

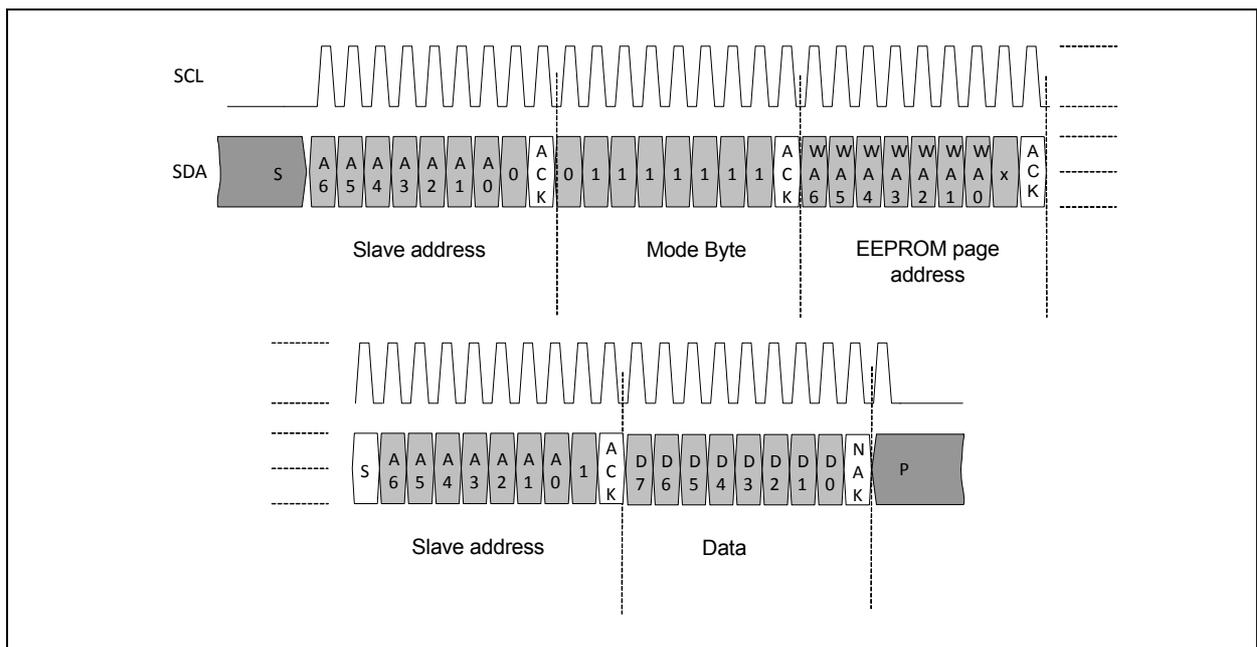
Figure 75:
Writing Register Data with Auto-Incrementing Address over I²C



EEPROM Read

In order to read data from EEPROM, first a slave address and mode byte is sent, followed by the block address byte (MSB first). Then one or more blocks of data with address auto-incrementing (packets of 4 bytes) are transferred via SDA line, also MSB first. SDA is sampled at the SCL falling edge. In case the block on defined address does not exist, all 0 data is sent via SDA line.

Figure 76:
Reading a EEPROM Block over I²C

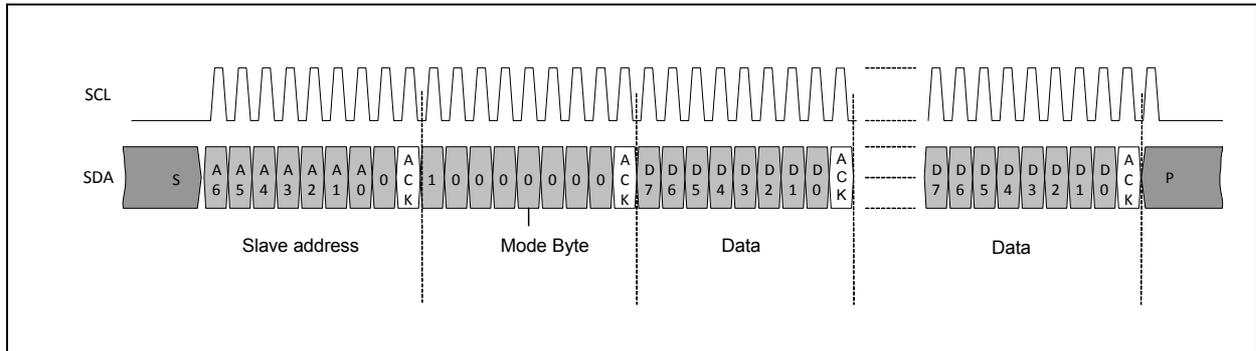


Loading Transmission Data into Buffer

Loading the transmitting data into the buffer is similar to writing data into an addressable register. Difference is that, in case of loading more bytes, all bytes go to the buffer. The command mode code 10b indicates buffer operations.

The following figure shows how to load the transmitting data into the buffer.

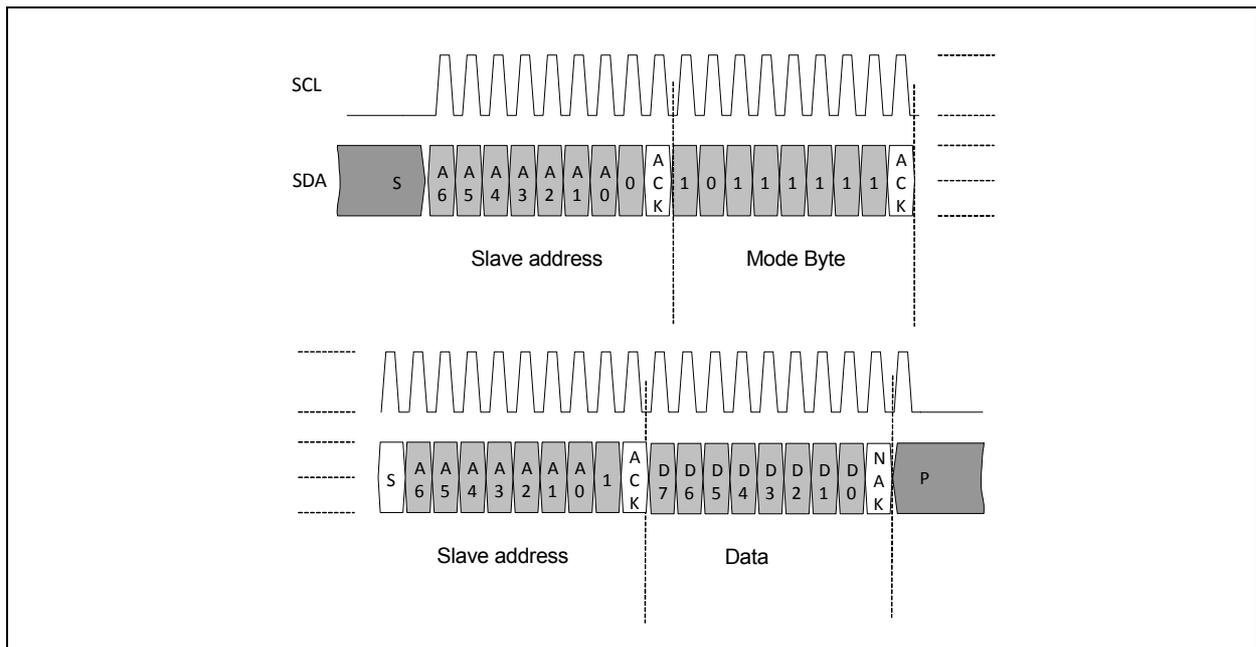
Figure 77:
Loading Data into Buffer over I²C



Reading Received Data from Buffer

Reading received data from the buffer is similar to reading data from an addressable register. Difference is that, in case of reading more bytes, they all come from the buffer. The command mode code 10b indicates buffer operations. In case of reading the received data from the buffer, all bits <C5 – C0> are set to 1.

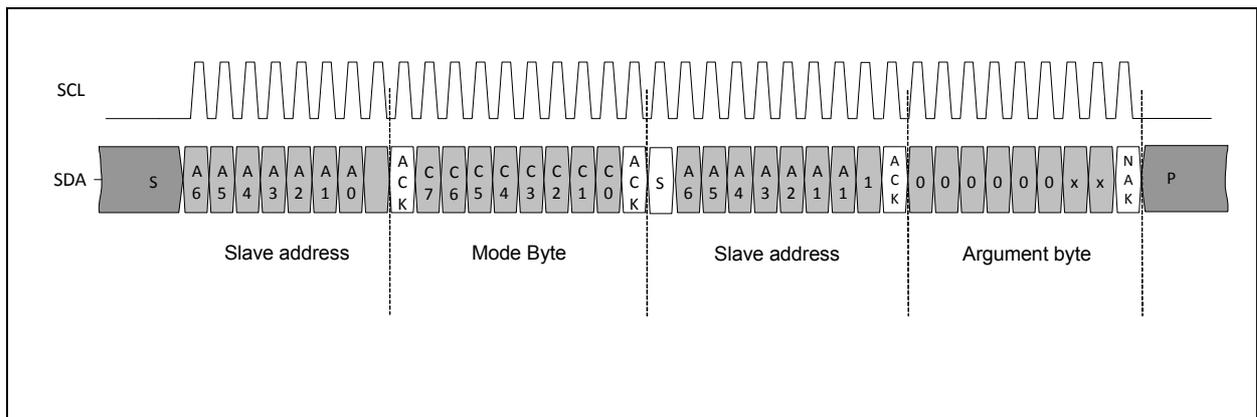
Figure 78:
Reading Data from Buffer over I²C



Direct Command Mode

Direct Command Mode is comprised of one command byte and argument byte. The two msb's of command code 11b indicate Direct Command Mode. The following six bits define command code, sent MSB first. Last two bits in argument byte indicate success of the direct command. Value 01h of the argument byte indicates that command was accepted, while 02h indicates it was rejected due to internal access priorities. The argument byte does not provide information on timing of the command execution.

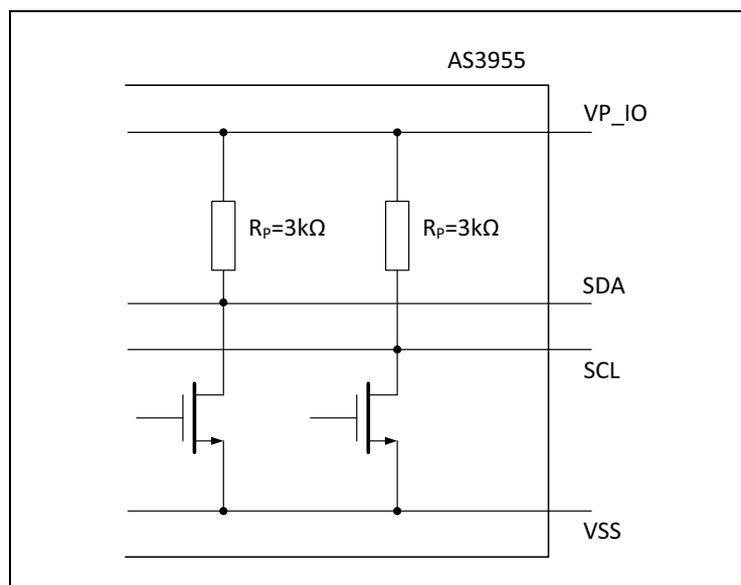
Figure 79:
Sending a Direct Command over I²C



I²C Electrical Connection

The AS3955 has two built-in 3kΩ pull-up resistors on SDA and SCL lines as shown on Figure 80 and used only in I²C mode. The SPI pull-down bits in IO Configuration Register must be set 00h.

Figure 80:
I²C Interface with Built-In Pull-Up Resistors



Interrupt Interface Description

There are two interrupt registers implemented in AS3955 ([Interrupt Register 0](#) and [Interrupt Register 1](#)).

When an interrupt condition is met, the source of interrupt bit is set in the interrupt register and the IRQ pin goes to high.

The microcontroller shall read the interrupt registers to distinguish between different interrupt sources. After an interrupt register is read, its content is reset to 0. IRQ pin goes to low after the interrupt bit(s), which caused its transition to high, has been read. Please note that there may be multiple interrupt register bits set in case the controller does not immediately read the Interrupt Registers after the IRQ signal was set and another event causing interrupt occurred.

The process of reading interrupt registers is composed of two phases. In the first phase, interrupts are pushed into internal buffer just before reading the first bit of the register. In the second phase the bits stored in the buffer are read out. The interrupts stored in each internal register are cleared and considered being read out at the rising edge of SCLK of the first interrupt bit for each interrupt register.

Buffer Interrupts and Buffer Status Register

The AS3955 contains a 32 byte buffer. In case of transmission the Control logic shifts data which was previously loaded by the external controller to the Framing Block and further to the Transmitter. During reception, the demodulated data is stored in the buffer and the external controller can download received data.

Transmit and receive capability of the AS3955 is limited by the buffer size. The maximum message size that can be received or transmitted is 32 bytes. The mutual access to the buffer from NFC block and SPI / I²C block is not allowed. At the beginning of each reception the buffer is cleared automatically. Before the data is loaded into the buffer via SPI / I²C for transmission the MCU must take care of clearing the buffer.

Five interrupts that can be triggered during read or write of the buffer:

- When buffer overflow or underflow is detected an `I_bf_err` is triggered. The reason for the interrupt can be distinguished by reading the [Buffer Status Register 1](#).
- Buffer overflow IRQ is not blocked, in case if more than 32 bytes are received from reader or written via SPI / I²C into buffer the buffer underflow IRQ is produced.
- When data is loaded via data RF an `I_rxs` and `I_rxe` are triggered
- When all data is transmitted an `I_txe` is triggered
- When buffer is already busy due to another operation in progress and MCU tries to access the internal buffer an interrupt `I_acc_err` shall be triggered.

After data is received, the microcontroller shall check the amount of bytes actually received. This information is available in the [Buffer Status Register 2](#), which displays number of bytes in the buffer not yet read out. [Buffer Status Register 1](#) additionally contains a flag indicating buffer overflow.

EEPROM Read and Write

The EEPROM can be accessed from SPI / I²C interface or over RF field issuing NFC read and write commands. The EEPROM access is controlled by the internal EEPROM controller. Since the mutual access to the EEPROM is not possible AS3955 needs to arbitrate between MCU and NFC device in cases of mutual access. How this is handled it is defined according to `arbit_mod` bit. Typical EEPROM programming time is 8.3 ms and should not exceed 9.5 ms. Please note that word data is sent MSB first which is opposite to the RFID EEPROM programming where LSB is sent first.

Three interrupts can be triggered during EEPROM read or write:

- When EEPROM programming is executed successfully an `I_io_eewr` interrupt is triggered.
- When EEPROM programming is terminated due to higher priority of the RF part, writing to a non-existent address or writing is not allowed an `I_eeac_err` interrupt is triggered.
- When EEPROM is busy and can't be accessed an `I_acc_err` interrupt is triggered

Note that EEPROM does not contain any kind of anti-tearing mechanism that would allow detection of corrupted data

The valid range for the Block Address byte depends on the EEPROM size. For 4kbit EEPROM, the valid range for the Block Address byte is from `000_0000b` to `111_1111b` (EEPROM blocks from 00h to 7Fh).

Data Buffer

Data in buffer is organized in bytes; each byte is terminated by a parity bit. Data bits in a byte are numbered from `b1` to `b8` where `b1` is LSB bit, LSB is sent first.

Data sent over SPI / I²C is also organized in bytes, bits in a byte are marked `D0` to `D7`, where `D0` is LSB bit, MSB is sent first. During reception, the framing engine checks the parity bit and removes it from data frame, and only data bytes are put into buffer. During transmission, the process is reversed, only data bytes are put into buffer, while the framing engine adds the parity bits.

The data bits `b1` to `b8` are mapped to buffer data bits `D0` to `D7`, which means that the order of receiving/transmitting bits in a byte is reversed (the bytes are sent LSB first while the SPI / I²C bytes are sent MSB first).

Direct Commands

Figure 81:
List of Direct Commands

| Command Byte Value <hex> | Command | Comments |
|--------------------------|----------------------------|---|
| C2 | Set Default | Set AS3955 to default state |
| C4 | Clear Buffer | Clears buffer and its error flags |
| C6 | Restart Transceiver | Restarts transceiver communication logics |
| C7 | Disable/Enable Transceiver | Toggles disable transceiver bit |
| C8 | Transmit Buffer | Starts a transmit sequence of the buffer content |
| C9 | Transmit ACK | Transmits NFC ACK reply |
| CA | Transmit NACK 0 | Transmits NFC NAK reply with code 0h |
| CB | Transmit NACK 1 | Transmits NFC NAK reply with code 1h |
| CD | Transmit NACK 4 | Transmits NFC NAK reply with code 4h |
| CC | Transmit NACK 5 | Transmits NFC NAK reply with code 5h |
| D0 | Go To Sleep | Puts a tag in SLEEP state |
| D1 | Go To Sense | Puts a tag in SENSE state |
| D2 | Go To Sense / Sleep | Puts a tag in SENSE or SLEEP state depending on the internal AS3955 state |

Set Default

This direct command puts the AS3955 in the same state as power-up initialization except for I²C bit in register IC Status Display Register that defines the interface and IO Configuration Register, which are not cleared.

Clear Buffer

Clears buffer, buffer overflow and underflow bits.

Restart Transceiver

Resets protocol logic to its initial state except for the tag state. This interrupts all receptions and transmissions that are being processed at the time.

Disable/Enable Transceiver

This direct command toggles internal states which enables or disables the reception or transmission of the data. This command should be used when MCU wants to disable the interruption of EEPROM write/read over SPI/I²C by the received RF wire/read command. The RF part can also be enabled by issuing the Restart RX/TX. This command shall not change the tag state.

Transmit Buffer

This direct command trigger the transmission of data stored in the buffer over the RF link. Before issuing a Transmit command, the MCU shall send a Clear Buffer command, and write into the buffer the data to be transmitted by sending a Buffer Load command.

Execution of this direct command is only enabled when the AS3955 antenna coil is in a PCD field (VP_INT is above HF_PON threshold) and AS3955 Extended or Tunneling modes are enabled.

Transmit ACK

Transmit 4-bit ACK response.

Transmit NACK 0-5

Transmit 4-bit NACK response with different NACK codes.

Go To Sleep

Puts tag in SLEEP state. Execution of this direct command is only enabled when AS3955 antenna coil is in a NFC device field (VP_INT is above HF_PON threshold).

Go To Sense

Puts tag in SENSE state. Execution of this direct command is only enabled when the AS3955 antenna coil is in a NFC device field (VP_INT is above HF_PON threshold).

Go To Sense / Sleep

Puts tag in SLEEP state depending on the internal state of the tag. Execution of this direct command is only enabled when AS3955 antenna coil is in a NFC device field (VP_INT is above HF_PON threshold).

Register Description

The 6-bit register addresses below are defined in hexadecimal notation. The possible address range is from 00h to 3Fh.

There are two types of registers implemented in AS3955: configuration registers and display registers. The configuration registers are used to configure AS3955. They can be written and read through SPI / I²C (RW). The display registers are read-only (RO); they contain information about AS3955 internal state, which can be accessed through SPI / I²C.

Registers Overview

Figure 82:
List of the SPI / I²C Internal Registers

| Address [hex] | Content | Comment | Type |
|---------------|----------------------------------|---|-------|
| 00 | IO Configuration Register | | RW |
| 01 | IC Configuration Register 0 | Default: IC_CFG0 | RO/RW |
| 02 | IC Configuration Register 1 | Default: IC_CFG1 | RO |
| 03 | IC Configuration Register 2 | Default: IC_CFG2+ battery supply enable | RO/RW |
| 04 | RFID Status Display Register | | RO |
| 05 | IC Status Display Register | | RO |
| 08 | Mask Interrupt Register 0 | Default: MIRQ_0 | RW |
| 09 | Mask Interrupt Register 1 | Default: MIRQ_1 | RW |
| 0A | Interrupt Register 0 | | RO |
| 0B | Interrupt Register 1 | | RO |
| 0C | Buffer Status Register 2 | | RO |
| 0D | Buffer Status Register 1 | | RO |
| 0E | Last NFC Address Access Register | | RO |
| 1E | Version Control – Major Revision | | RO |
| 1F | Version Control – Minor Revision | | RO |

IO Configuration Register

Figure 83:
IO Configuration Register

| Address 00h: IO Configuration | | | | Type: RW |
|-------------------------------|----------|---------|--|----------|
| Bit | Name | Default | Function | Comments |
| 7 | miso_pd2 | 0 | 1: pull down on MISO, when \SS is low and MISO is not driven by the AS3955 | |
| 6 | miso_pd1 | 0 | 1: pull down on MISO when \SS is high | |
| 5 | rfu | 0 | | |
| 4 | rfu | 0 | | |
| 3 | rfu | 0 | | |
| 2 | rfu | 0 | | |
| 1 | rfu | 0 | | |
| 0 | rfu | 0 | | |

Note(s):

1. Default value is loaded from EEPROM configuration block bits ([IC_CFG2](#)).

IC Configuration Registers

Figure 84:
IC Configuration Register 0

| Address 01h: IC Configuration 0 | | | | |
|---------------------------------|--------------|----------|---|------|
| Bit | Name | Default | Function | Type |
| 7 | slnt_mod | See note | 1: Enable silent mode | RO |
| 6 | slnt_vl<2> | | Silent mode voltage level (see Silent Mode) | RO |
| 5 | slnt_vl<1> | | | |
| 4 | slnt_vl<0> | | | |
| 3 | arbit_mod | | 1: RF has priority access to EEPROM over SPI / I ² C | RW |
| 2 | i2c_addr3<2> | | I ² C slave address | RO |
| 1 | i2c_addr2<1> | | | |
| 0 | i2c_addr1<0> | | | |

Note(s):

1. Default value is loaded from EEPROM configuration block bits ([IC_CFG0](#)).

Figure 85:
IC Configuration Register 1

| Address 02h: IC Configuration 1 | | | | |
|---------------------------------|-----------|----------|---|------|
| Bit | Name | Default | Function | Type |
| 7 | en_rx_crc | See note | 1: CRC stored in the buffer in the Tunneling mode | RO |
| 6 | vreg<4> | | Voltage level for voltage regulator VP_REG (see Energy Harvesting) | RO |
| 5 | vreg<3> | | | |
| 4 | vreg<2> | | | |
| 3 | vreg<1> | | | |
| 2 | vreg<0> | | | |
| 1 | rreg<1> | | Output resistance value for voltage regulator VP_REG (see Energy Harvesting) | RO |
| 0 | rreg<0> | | | |

Note(s):

1. Default value is loaded from EEPROM configuration block bits ([IC_CFG1](#)).

Figure 86:
IC Configuration Register 2

| Address 03h: IC Configuration 2 | | | | |
|---------------------------------|-------------------|----------|---|------|
| Bit | Name | Default | Function | Type |
| 7 | rfcfg_en | See note | See Configuration Byte IC_CFG2 for full description | RO |
| 6 | tun_mod | | | RW |
| 5 | ext_mod | | | RW |
| 4 | nak_on_crc_parity | 0 | | RO |
| 3 | auth_set | See note | | RO |
| 2 | selr_b6_inv | 0 | | RO |
| 1 | powm<1> | See note | | RW |
| 0 | powm<0> | | | RW |

Note(s):

1. Default value is loaded from EEPROM configuration block bits ([IC_CFG2](#)).

Status Display Registers

Figure 87:
RFID Status Display Register

| Address 04h: RFID Status Display | | | |
|----------------------------------|----------------|---|------|
| Bit | Name | Function | Type |
| 7 | hf_pon | 1: PICC AFE is active | RO |
| 6 | state<3> | 0000: POWER OFF 0001: SENSE 0011: RESOLUTION 0010: RESOLUTION_L2 0110: SELECTED 0111: SECTOR_2 1111: SECTORX_2 1110: SELECTEDX 1010: SENSEX_L2 1011: SENSEX 1001: SLEEP | RO |
| 5 | state<2> | | |
| 4 | state<1> | | |
| 3 | state<0> | | |
| 2 | state_notvalid | 1: State update – indicates that the state is not valid | RO |
| 1 | rfu | | RO |
| 0 | rfu | | RO |

Figure 88:
IC Status Display Register

| Address 05h: IC Status Display | | | | |
|--------------------------------|-------------|--------------|---|------|
| Bit | Name | Default | Function | Type |
| 7 | ee2k | See note (1) | | RO |
| 6 | i2c | | | RO |
| 5 | chipkill_2 | 0 | Signal depends on the CHIP_KILL value at initialization | RO |
| 4 | chipkill_1 | 0 | Signal depends on the CHIP_KILL value at initialization | RO |
| 3 | auth_locked | 0 | Signal depends on the AUTH_CNT value | RO |
| 2 | rfu | 0 | | RO |
| 1 | rfu | 0 | | RO |
| 0 | rfu | 0 | | RO |

Note(s):

1. Default is set during production

Interrupt Registers

Figure 89:
Mask Interrupt Register 0

| Address 08h: Mask Interrupt 0 ⁽¹⁾ | | | | |
|--|----------|--------------|-------------------|------|
| Bit | Name | Default | Function | Type |
| 7 | M_pu | See note (2) | Mask I_pu IRQ | RW |
| 6 | M_wu_a | | Mask I_wu_a IRQ | RW |
| 5 | M_slp | | Mask I_slp IRQ | RW |
| 4 | M_eew_rf | | Mask I_eew_rf IRQ | RW |
| 3 | M_eer_rf | | Mask I_eer_rf IRQ | RW |
| 2 | M_rxe | | Mask I_rxe IRQ | RW |
| 1 | M_txe | | Mask I_txe IRQ | RW |
| 0 | M_xrf | | Mask I_xrf IRQ | RW |

Note(s):

1. The mask bits only mask the triggering of the physical interrupt line. The interrupt bits in the Interrupt register still get set when the interrupt is masked.
2. Default values are loaded from EEPROM configuration block bits ([MIRQ_0](#)).

Figure 90:
Mask Interrupt Register 1

| Address 09h: Mask Interrupt 1 ⁽¹⁾ | | | | |
|--|------------|--------------|---------------------|------|
| Bit | Name | Default | Function | Type |
| 7 | M_rxs | See note (2) | Mask I_rxs IRQ | RW |
| 6 | M_frm_err | | Mask I_frm_err IRQ | RW |
| 5 | M_par_err | | Mask I_par_err IRQ | RW |
| 4 | M_crc_err | | Mask I_crc_err IRQ | RW |
| 3 | M_bf_err | | Mask I_bf_err IRQ | RW |
| 2 | M_io_eewr | | Mask I_io_eewr IRQ | RW |
| 1 | M_eaac_err | | Mask I_eaac_err IRQ | RW |
| 0 | M_acc_err | | Mask I_acc_err IRQ | RW |

Note(s):

1. The mask bits only mask the triggering of the physical interrupt line. The interrupt bits in the Interrupt register still get set when the interrupt is masked.
2. Default values are loaded from EEPROM configuration block bits ([MIRQ_1](#)).

Figure 91:
Interrupt Register 0

| Address 0Ah: Interrupt Register 0 | | | |
|-----------------------------------|----------|--|------|
| Bit | Name | Function | Type |
| 7 | I_pu | Power-up IRQ. Interrupt is triggered at each power up (RF or battery), when chip is already powered and RF field appears and after Go To Sleep , Go To Sense , Go To Sense / Sleep and Set Default command | RO |
| 6 | I_wu_a | Wake-up IRQ at entry in SELECTED state | RO |
| 5 | I_slp | IRQ due to reception of SLP_REQ command | RO |
| 4 | I_eew_rf | PCD has updated the content of the data area | RO |
| 3 | I_eer_rf | PCD has read the content of the data area | RO |
| 2 | I_rxe | IRQ due to End of Receive. Applicable when receive frame is put in buffer | RO |
| 1 | I_txe | IRQ due to End of Transmission. Applicable when data from buffer is sent | RO |
| 0 | I_xrf | Exit RF field IRQ | RO |

Note(s):

1. Power-up and [Set Default](#) command set the content of this register to 0. After Interrupt Register has been read, its content is set again to 0.

Figure 92:
Interrupt Register 1

| Address 0Bh: Interrupt Register 1 | | | |
|-----------------------------------|-----------|---|------|
| Bit | Name | Function | Type |
| 7 | I_rxs | IRQ due to Start of Receive. Applicable when receive frame is put in buffer | RO |
| 6 | I_frm_err | Lower layer error (broken byte, wrong bit coding sequence) / In case of error the receive data is still put in buffer, error IRQ is additionally sent | RO |
| 5 | I_par_err | Parity error / In case of error the receive data is still put in buffer, error IRQ is additionally sent | RO |
| 4 | I_crc_err | CRC error / In case of CRC error the receive data is still put in buffer, error IRQ is additionally sent | RO |
| 3 | I_bf_err | Buffer error (overflow/underflow). See Buffer Status Register 2 | RO |

| Address 0Bh: Interrupt Register 1 | | | |
|-----------------------------------|------------|---|------|
| Bit | Name | Function | Type |
| 2 | l_io_eewr | IRQ due to successful termination of EEPROM programming. In case EEPROM write command was sent through SPI / I ² C | RO |
| 1 | l_eaac_err | IRQ due to EEPROM write on write protected block or write to non-existing address | RO |
| 0 | l_acc_err | IRQ due to interruption of SPI / I ² C operation on buffer, EEPROM or registers due to access control | RO |

Note(s):

1. Power-up and [Set Default](#) command reset the content of this register to 0. After Interrupt Register has been read, its content is set again to 0.

Buffer Registers

Figure 93:
Buffer Status Register 2

| Address 0Ch: Buffer Status Register 2 | | | |
|---------------------------------------|-----------------|--|------|
| Bit | Name | Function | Type |
| 7 | buf_len_invalid | Buffer content is being changed – data length not valid | RO |
| 6 | rfu | | RO |
| 5 | buf_len<5> | Number of data bytes (binary coded) in the buffer not yet read out. Valid range is from 000000b to 100000b – 000000b means that there are no data bytes to be read out | RO |
| 4 | buf_len<4> | | |
| 3 | buf_len<3> | | |
| 2 | buf_len<2> | | |
| 1 | buf_len<1> | | |
| 0 | buf_len<0> | | |

Note(s):

1. Power-up and commands [Set Default](#) and [Clear Buffer](#) reset the content of this register to 0.

Figure 94:
Buffer Status Register 1

| Address 0Dh: Buffer Status Register 1 | | | |
|---------------------------------------|-------------|---|------|
| Bit | Name | Function | Type |
| 7 | rfu | | RO |
| 6 | rfu | | RO |
| 5 | rf_busy | Extended mode buffer status flags (see Extended Mode); <i>io_busy</i> flag is omitted since in this it is always zero when this byte is read out | RO |
| 4 | rf_data_rdy | | RO |
| 3 | io_data_rdy | | RO |
| 2 | rfu | | RO |
| 1 | buf_unf | Buffer underflow. Set when read more bytes than actual content of buffer | RO |
| 0 | buf_ovr | Buffer overflow. Set when written more bytes than actual content of buffer | RO |

Note(s):

1. Power-up and commands [Set Default](#) and [Clear Buffer](#) reset the content of this register to 0.

NFC Last Address Register

Figure 95:
Last NFC Address Access Register

| Address 0Eh: Last NFC Address Access | | | |
|--------------------------------------|--------------|---|------|
| Bit | Name | Function | Type |
| 7 | last_addr<7> | Contains last address accessed by reader. Updated on internal EEPROM access over RF. The address is 8 bit long and contains also access attempts that are higher than the EEPROM address space. Value FFh indicates that the value is being updated and is not valid. | RO |
| 6 | last_addr<6> | | |
| 5 | last_addr<5> | | |
| 4 | last_addr<4> | | |
| 3 | last_addr<3> | | |
| 2 | last_addr<2> | | |
| 1 | last_addr<1> | | |
| 0 | last_addr<0> | | |

Note(s):

1. Power-up and commands [Set Default](#) and [Clear Buffer](#) reset the content of this register to 0.

Version Control Register

Figure 96:
Version Control – Major Revision

| Address 1Eh: Version Control – Major Revision | | | | |
|---|------|---------|------------------------|------|
| Bit | Name | Default | Function | Type |
| 7 | maj7 | 0 | Major version revision | RO |
| 6 | maj6 | 0 | | |
| 5 | maj5 | 0 | | |
| 4 | maj4 | 0 | | |
| 3 | maj3 | 0 | | |
| 2 | maj2 | 0 | | |
| 1 | maj1 | 1 | | |
| 0 | maj0 | 0 | | |

Figure 97:
Version Control – Minor Revision

| Address 1Fh: Version Control – Minor Revision | | | | |
|---|------|---------|------------------------|------|
| Bit | Name | Default | Function | Type |
| 7 | min7 | 0 | Minor version revision | RO |
| 6 | min6 | 0 | | |
| 5 | min5 | 0 | | |
| 4 | min4 | 0 | | |
| 3 | min3 | 0 | | |
| 2 | min2 | 0 | | |
| 1 | min1 | 1 | | |
| 0 | min0 | 0 | | |

Note(s):

1. Default values are loaded from EEPROM configuration block bits ([Configuration Bytes MIRQ_0 and MIRQ_1](#))

Application Information

This section describes some general use cases of AS3955 in combination with a microcontroller. The examples are shown for explanatory purposes of specific AS3955 features.

Writing a NDEF Message into AS3955 Memory

AS3955 is an NFC Forum Type 2 Tag Platform and as such its memory has to have the layout defined in NFC Forum Type 2 Tag Operation Specification. In a Type 2 Tag platform data is stored in memory in the form of TLV blocks. A TLV block has three fields: T field – tag field, L field – length field and V field – value field.

- T field – indicates the type of the TLV block and is one byte long.
- L field – gives the size in bytes of the value field. It's 1 or 3 bytes long depending on the size of the value field. When the size of the value field is between 0 and 254 bytes, the L field is one byte long and has a value between 00h and FEh. For sizes of the value field between 255 and 65535 bytes, the L field is 3 bytes long. First byte is FFh indicating that the size will be provided by the following two bytes that can have values between 00FFh and FFEh.
- V field – holds the bytes of the data carried by the TLV block. When the L field is zero or not present, the V field is omitted.

For NDEF messages the TLV block has a T field value of 03h and the message is stored inside the V field. The NDEF Message TLV should always be present in a T2T platform and at a minimum it is an empty NDEF Message TLV, which is defined as an NDEF Message TLV with L field equal 00h and no V field. For the NDEF message format refer to NFC Data Exchange Format Technical Specification.

Two more TLV types can exist in the memory, these are Lock Control TLV and Memory Control TLV. When they are present inside the memory, the NDEF Message TLV should be placed after them.

Another structure that has to be present in the memory of T2T platform is the Capability Container (CC). CC contains NFC Forum management data. It is comprised of 4 bytes and always resides at block number 03h of the memory.

The AS3955 comes with programmed CC (refer to Capability Container in the section Memory Organization of this datasheet), no Lock Control TLV and no Memory Control TLV. The NDEF Message TLV then can be placed already at the beginning of the data area of the memory - block 04h.

Following is an example of an NDEF message with one Well Known URL record, which carries the www.ams.com URL:

- In the NDEF message format, the URL above will be represented by the bytes:
[D1 01 08 55 01 61 6D 73 2E 63 6F 6D]h
- The NDEF Message TLV will have the content: [03 0C D1 01 08 55 01 61 6D 73 2E 63 6F 6D]h; first byte is 03h – T field value, indicating this is an NDEF Message TLV, second byte is 0Ch saying the V field is 12 bytes long, the following 12 bytes are the V field holding the NDEF message.

The NDEF message TLV given above will fully occupy the first 3 blocks of the data area and the first two bytes of the 4th block of the data area (each memory block is 4 bytes long).

Reading/Writing Through NFC Device

NFC devices use the T2T WRITE command to write data into a T2T platform. The T2T WRITE command consists of 6 bytes, 1st byte is the command code – A2h, 2nd byte is the block number where data is to be written and the remaining 4 bytes are the data content. When the execution of the command is successful, the tag will return an ACK response – A0h. In case of failure, a NACK code will be returned (see Error Handling section of this datasheet for NACK codes).

The T2T READ command consists of 2 bytes, first byte is the command code and is equal to 30h, second byte is the first block number from which data will be returned. The response, when successful, will return 4 blocks of data, each block is 4 bytes long, so 16 bytes of data. In case of error, one byte NACK response will be returned (see error handling section of AS3955 datasheet for NACK codes).

AS3955 has to be configured in Extended or Normal mode in order for an NFC device to write/read NDEF data from the EEPROM of the tag.

An NFC device has to send 4 write commands to the tag to write the NDEF Message TLV bytes [03 0C D1 01 08 55 01 61 6D 73 2E 63 6F 6D]hex defined in the section above into blocks 04h-07h:

1. [A2 04 03 0C D1 01]hex
2. [A2 05 08 55 01 61]hex
3. [A2 06 6D 73 2E 63]hex
4. [A2 07 6F 6D 00 00]hex

The same NDEF Message TLV bytes can be obtained with a single READ command, as the TLV occupies only 4 blocks and the READ command returns 16 bytes. In case of longer NDEF messages of course several READs are necessary until the complete message has been read out.

Reading/Writing Through SPI / I²C

An MCU can write the NDEF Message TLV bytes [03 0C D1 01 08 55 01 61 6D 73 2E 63 6F 6D]hex into blocks 04h-07h by sending the following 4 bytes sequences through SPI or I²C (as described in the Wired Interfaces section of this datasheet):

Writing:

1. [40 08 03 0C D1 01]hex
2. [40 0A 08 55 01 61]hex
3. [40 0C 6D 73 2E 63]hex
4. [40 0E 6F 6D 00 00]hex

To read the same data bytes written in blocks 04h – 07h, the MCU should send the bytes sequences below (2 clock cycles for MODE and address and 4 additional clock cycles to read the data sent from AS3955, refer to the Wired Interfaces section of this datasheet):

1. [7F 08 00 00 00 00]hex
2. [7F 0A 00 00 00 00]hex
3. [7F 0C 00 00 00 00]hex
4. [7F 0E 00 00 00 00]hex

Using Extended Mode to Switch AS3955 into Tunneling Mode

An NFC device can force the AS3955 in Tunneling mode by sending a command via Extended to the MCU to switch the AS3955 operating mode from Extended to Tunneling (refer to Extended mode section of this datasheet for implementation details). If the MCU should emulate a T4T after the switch, AS3955 should be sent to SENSE state by the MCU with the direct command "Go To Sense" (command code D1h) and the NFC device should do a new anti-collision round for proper T4T activation.

Using Tunneling Mode to Emulate a NFC Type 4 Tag

For T4T emulation the Tunneling mode has to be enabled by setting the tun_mod bit of IC_CFG2 to 1 and the SELR to 0x20. With this configuration, once AS3955 has passed anti-collision and entered SELECTED state, all frames coming from the NFC interface will be pushed to the BUFFER. That means also the SENS_REQ, ALL_REQ, SSD_REQ, SEL_REQ and SLP_REQ will not be executed by the AS3955, so MCU must send the tag to SLEEP state on SLP_REQ (with direct command “Go To Sleep”) and to SENSE state (with direct command “Go To Sense”) on any of the other commands listed above.

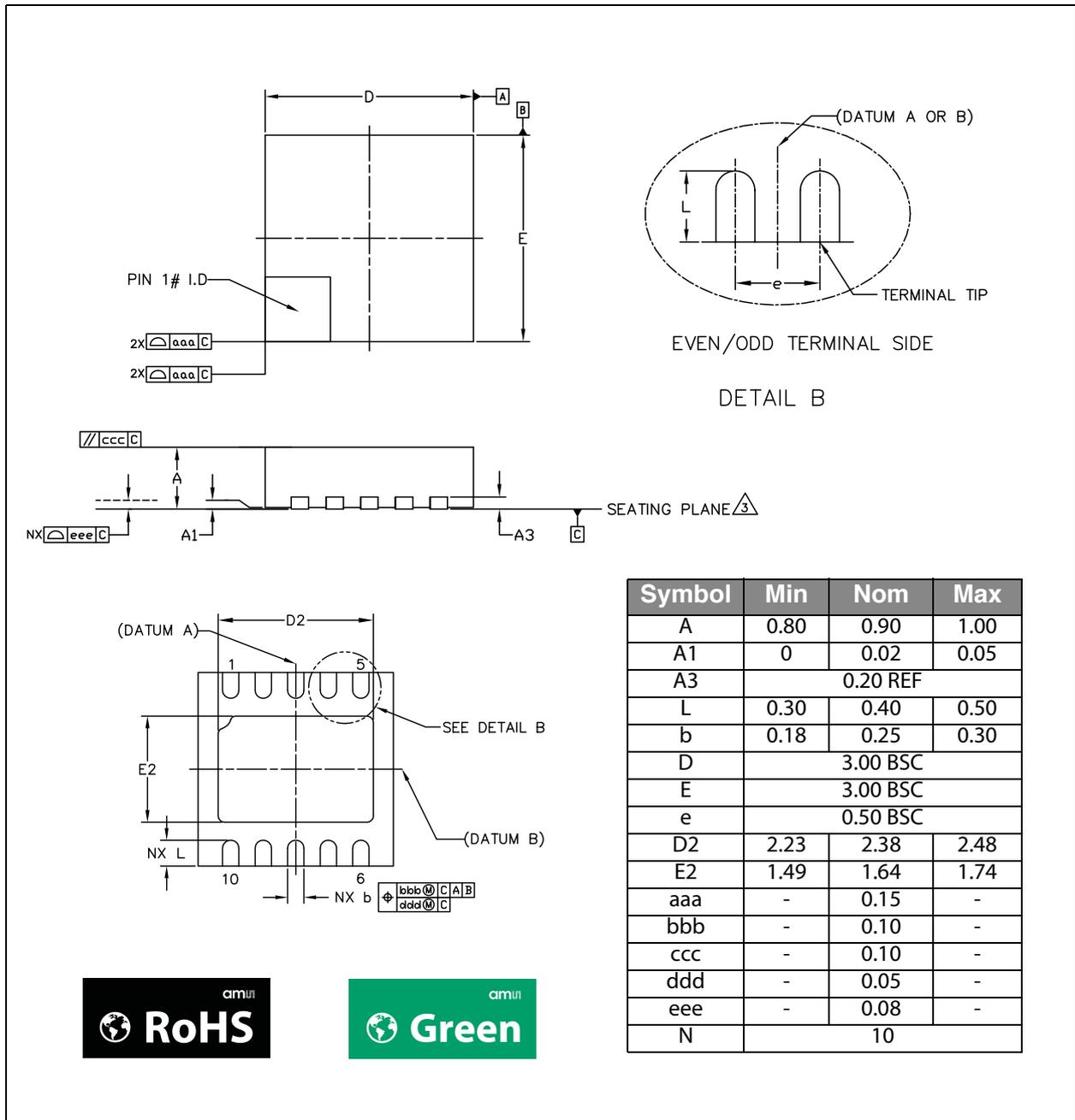
The ISO14443A-4 command RATS should also be handled by the MCU. RATS is sent by an NFC device to activate ISO14443A-4, on top of which is T4T platform is implemented.

References

- [EMVCO-1] EMVCo Type Approval Contactless Terminal Level 1 – PCD Digital Test Bench & Test Cases, version 2.3.1a, January 2013
- [ISO14443-3] ISO/IEC 14443-3:2011(E), Identification cards — Contactless integrated circuit cards — Proximity cards — Part 3: Initialization and anticollision
- [ISO14443-4] ISO/IEC 14443-4:2008(E), Identification cards — Contactless integrated circuit(s) cards — Proximity cards — Part 4: Transmission protocol
- [ISO18092] ISO/IEC 18092:2013 — Information technology — Telecommunications and information exchange between systems — Near Field Communication — Interface and Protocol (NFCIP-1)
- [ISO7816-3] ISO/IEC 7816-3:2006, Identification cards — Integrated circuit cards — Part 3: Cards with contacts — Electrical interface and transmission protocols
- [ISO7816-4] ISO/IEC 7816-4:2005, Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange
- [ISO7816-6] ISO/IEC 7816-6:2004, Identification cards — Integrated circuit cards — Part 6: Interindustry data elements for interchange
- [NFC Analog] NFC Analog Specification — NFC Forum, 11.07.2011, Version 1.0
- [NFC Digital] NFC Digital Protocol — NFC Forum, 17.11.2010, Version 1.0
- [NDEF] NFC Data Exchange Format (NDEF), Technical Specification — NFC Forum, 24.07.2006, Version 1.0
- [PHDC] Personal Health Device Communication), Technical Specification — NFC Forum, 27.02.2013, Version 1.0
- [T2T] Tag 2 Type Operation, Technical Specification — NFC Forum, 31.05.2011, Version 1.1
- [T4T] Tag 4 Type Operation, Technical Specification — NFC Forum, 28.06.2011, Version 2.0

Package Drawings & Markings

Figure 98:
Package Outline Drawings MLPD



Note(s):

1. Dimensioning and tolerancing conform to ASME Y14.5M-1994.
2. All dimensions are in millimeters. Angles are in degrees.
3. Coplanarity applies to the exposed heat slug as well as the terminal.
4. Radius on terminal is optional.
5. N is the total number of terminals.

Figure 99:
Package Marking MLPD (I²C)

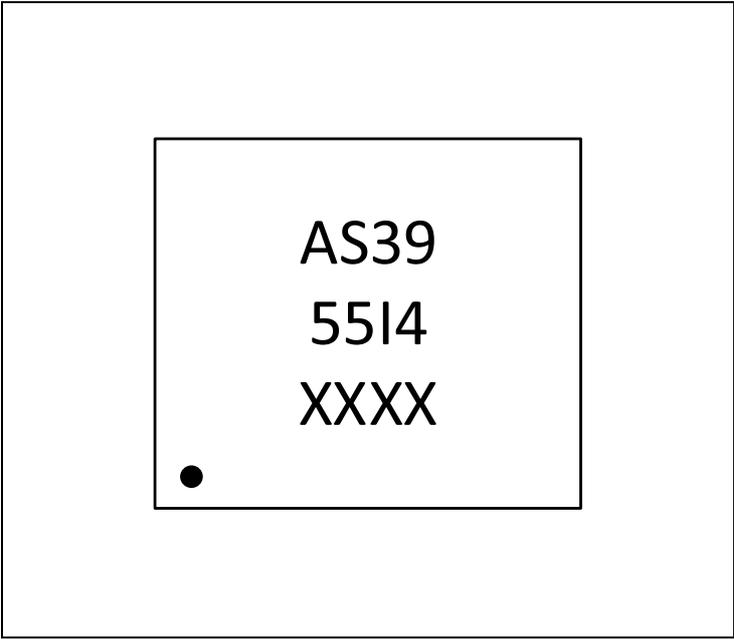


Figure 100:
Package Marking MLPD (SPI)

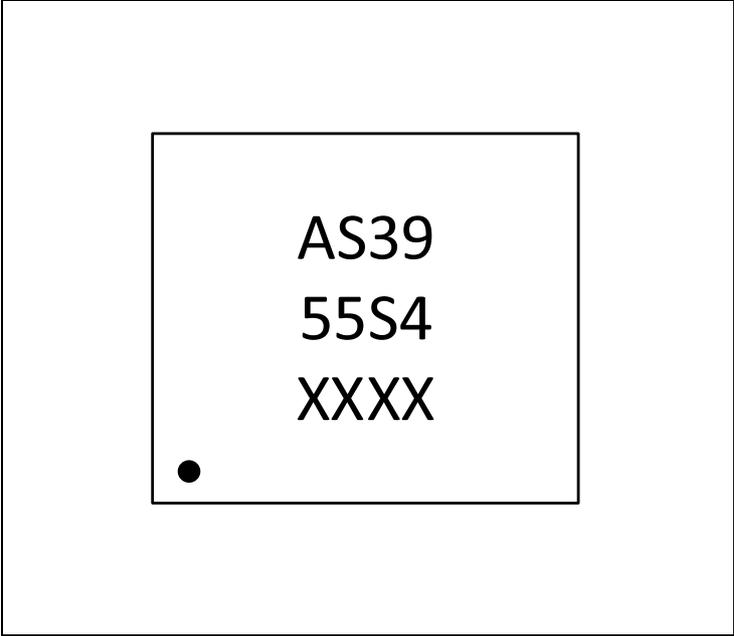
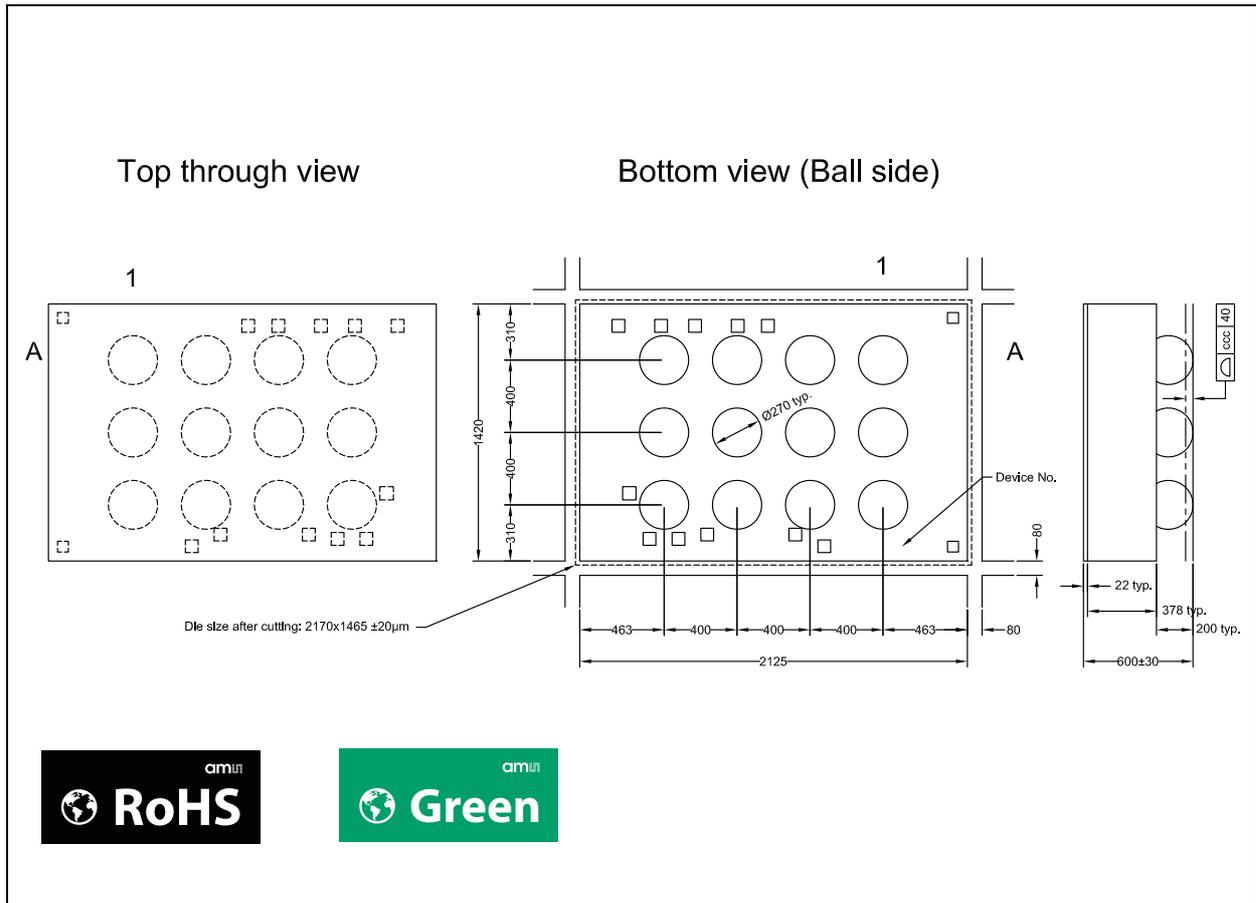


Figure 101:
Package Code MLPD

| |
|-----------|
| XXXX |
| Tracecode |

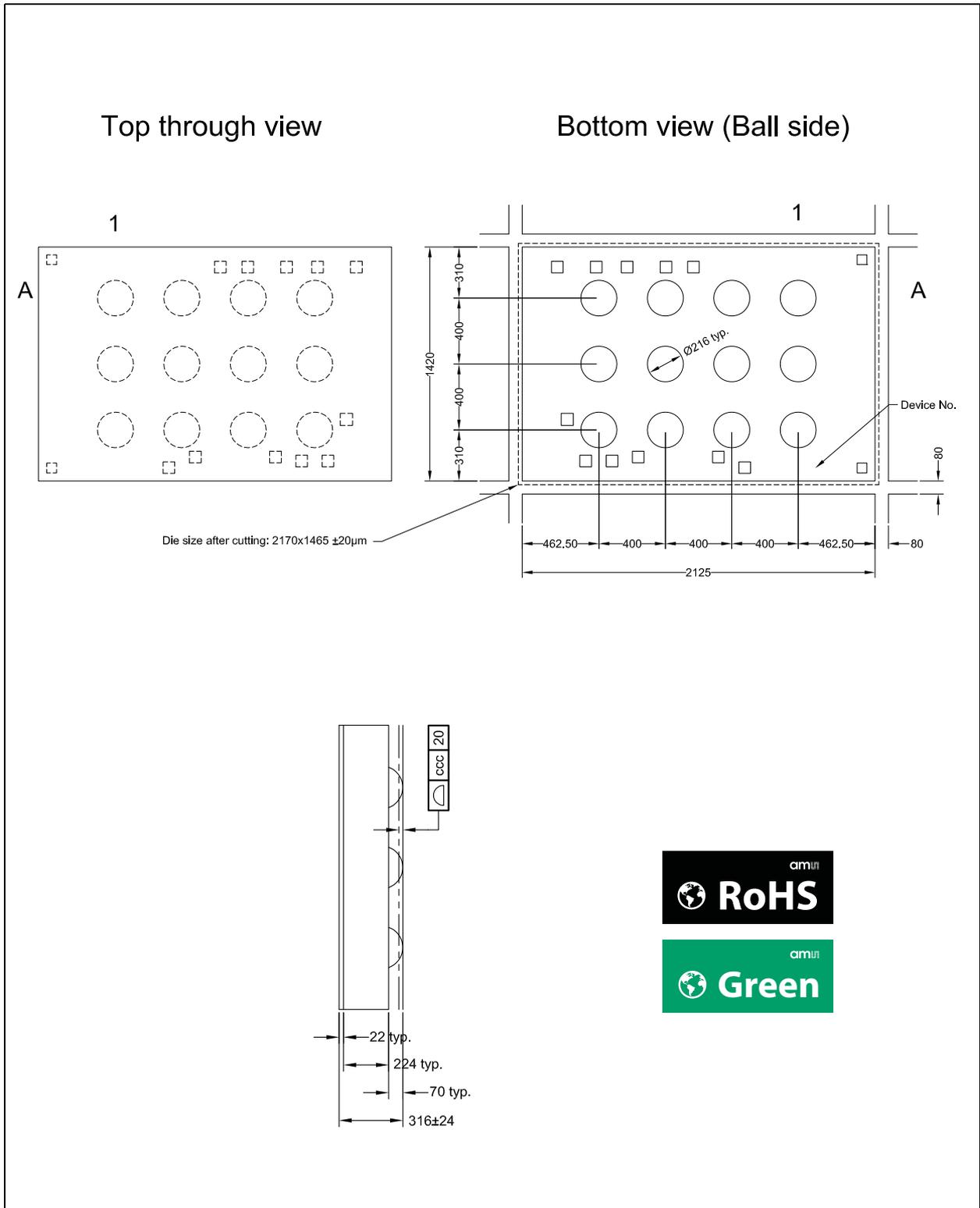
Figure 102:
Package Outline Drawings WL-CSP



Note(s):

1. Pin 1 = A1.
2. ccc Coplanarity.
3. All dimensions are in μm .
4. This applies to AWLT ordering codes.

Figure 103:
Package Outline Drawings Thin WL-CSP



Note(s):

1. Pin1 = A1.
2. ccc coplanarity.
3. All dimensions are in μm.
4. This applies to ATWT ordering codes.

Figure 104:
Package Marking WL-CSP

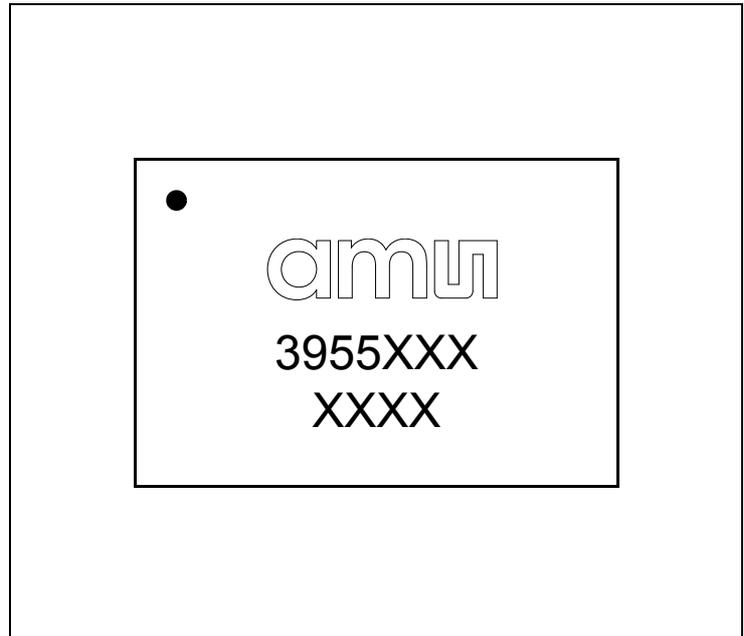


Figure 105:
Package Code WL-CSP

| |
|-----------|
| XXXX |
| Tracecode |

Ordering & Contact Information

Figure 106:
Ordering Information

| Ordering Code | Package | Marking | Delivery Form | Configuration |
|-----------------|--------------|----------|---------------|---------------------------------|
| AS3955A-ATDM-S4 | MLPD | AS3955S4 | Mini Reel | SPI - EEPROM 4kbit |
| AS3955A-ATDT-S4 | MLPD | AS3955S4 | Tape & Reel | SPI - EEPROM 4kbit |
| AS3955A-ATDM-I4 | MLPD | AS3955I4 | Mini Reel | I ² C - EEPROM 4kbit |
| AS3955A-ATDT-I4 | MLPD | AS3955I4 | Tape & Reel | I ² C - EEPROM 4kbit |
| AS3955A-AWLT-S4 | WL-CSP | 3955_S4 | Tape & Reel | SPI - EEPROM 4kbit |
| AS3955A-AWLT-I4 | WL-CSP | 3955_I4 | Tape & Reel | I ² C - EEPROM 4kbit |
| AS3955A-ASWF-S4 | Sorted wafer | NA | Wafer Box | SPI - EEPROM 4kbit |
| AS3955A-ASWF-I4 | Sorted wafer | NA | Wafer Box | I ² C - EEPROM 4kbit |
| AS3955A-ATWT-S4 | Thin WL-CSP | 3955_S4 | Tape & Reel | SPI - EEPROM 4kbit |
| AS3955A-ATWT-I4 | Thin WL-CSP | 3955_I4 | Tape & Reel | I ² C - EEPROM 4kbit |

Note(s):

1. Will be available on request.

Buy our products or get free samples online at:

www.ams.com/ICdirect

Technical Support is available at:

www.ams.com/Technical-Support

Provide feedback about this document at:

www.ams.com/Document-Feedback

For further information and requests, e-mail us at:

ams_sales@ams.com

For sales offices, distributors and representatives, please visit:

www.ams.com/contact

Headquarters

ams AG
Tobelbader Strasse 30
8141 Premstaetten
Austria, Europe

Tel: +43 (0) 3136 500 0

Website: www.ams.com

RoHS Compliant & ams Green Statement

RoHS: The term RoHS compliant means that ams AG products fully comply with current RoHS directives. Our semiconductor products do not contain any chemicals for all 6 substance categories, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, RoHS compliant products are suitable for use in specified lead-free processes.

ams Green (RoHS compliant and no Sb/Br): ams Green defines that in addition to RoHS compliance, our products are free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material).

Important Information: The information provided in this statement represents ams AG knowledge and belief as of the date that it is provided. ams AG bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. ams AG has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. ams AG and ams AG suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

Copyrights & Disclaimer

Copyright ams AG, Tobelbader Strasse 30, 8141 Premstaetten, Austria-Europe. Trademarks Registered. All rights reserved. The material herein may not be reproduced, adapted, merged, translated, stored, or used without the prior written consent of the copyright owner.

Devices sold by ams AG are covered by the warranty and patent indemnification provisions appearing in its General Terms of Trade. ams AG makes no warranty, express, statutory, implied, or by description regarding the information set forth herein. ams AG reserves the right to change specifications and prices at any time and without notice. Therefore, prior to designing this product into a system, it is necessary to check with ams AG for current information. This product is intended for use in commercial applications. Applications requiring extended temperature range, unusual environmental requirements, or high reliability applications, such as military, medical life-support or life-sustaining equipment are specifically not recommended without additional processing by ams AG for each application. This product is provided by ams AG "AS IS" and any express or implied warranties, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose are disclaimed.

ams AG shall not be liable to recipient or any third party for any damages, including but not limited to personal injury, property damage, loss of profits, loss of use, interruption of business or indirect, special, incidental or consequential damages, of any kind, in connection with or arising out of the furnishing, performance or use of the technical data herein. No obligation or liability to recipient or any third party shall arise or flow out of ams AG rendering of technical or other services.

Document Status

| Document Status | Product Status | Definition |
|--------------------------|-----------------|--|
| Product Preview | Pre-Development | Information in this datasheet is based on product ideas in the planning phase of development. All specifications are design goals without any warranty and are subject to change without notice |
| Preliminary Datasheet | Pre-Production | Information in this datasheet is based on products in the design, validation or qualification phase of development. The performance and parameters shown in this document are preliminary without any warranty and are subject to change without notice |
| Datasheet | Production | Information in this datasheet is based on products in ramp-up to full production or full production which conform to specifications in accordance with the terms of ams AG standard warranty as given in the General Terms of Trade |
| Datasheet (discontinued) | Discontinued | Information in this datasheet is based on products which conform to specifications in accordance with the terms of ams AG standard warranty as given in the General Terms of Trade, but these products have been superseded and should not be used for new designs |

Revision Information

| Changes from 1-04 (2016-Feb-03) to current revision 1-07 (2016-Nov-17) | Page |
|---|------|
| 1-04 (2016-Feb-03) to 1-05 (2016-Oct-24) | |
| Updated text under General Description | 1 |
| Updated Figure 1 | 2 |
| Updated Figure 3 | 4 |
| Updated Figure 13 | 12 |
| Updated Figure 16 and added text under it | 17 |
| Updated Memory Organization section by removing 2kbit EEPROM Organization | 28 |
| Updated Figure 106 | 96 |
| 1-05 (2016-Oct-24) to 1-06 (2016-Oct-31) | |
| Updated Figure 102 | 95 |
| 1-06 (2016-Oct-31) to 1-07 (2016-Nov-17) | |
| Updated Figure 6 | 8 |
| Updated Figure 25 | 29 |
| Updated figures 96 and 97 | 87 |
| Updated Figure 104 | 97 |
| Updated Figure 106 | 98 |

Note(s):

1. Page and figure numbers for the previous version may differ from page and figure numbers in the current revision.
2. Correction of typographical errors is not explicitly mentioned.

Content Guide

| | |
|-----------|---|
| 1 | General Description |
| 2 | Key Benefits & Features |
| 3 | Applications |
| 4 | Block Diagram |
| 6 | Pin Assignment |
| 7 | Pin Description |
| 8 | Absolute Maximum Ratings |
| 10 | Electrical Characteristics |
| 10 | Operating Conditions |
| 10 | DC/AC Characteristics for Digital Inputs and Outputs |
| 11 | Electrical Specifications |
| 14 | Detailed Description |
| 14 | Analog Frontend (AFE) |
| 15 | Power Management |
| 15 | Power Mode 0 |
| 17 | Power Mode 1 |
| 17 | Power Mode 2 |
| 17 | Power Mode 3 |
| 18 | Interface Arbitration |
| 18 | Arbitration Mode 0 |
| 18 | Arbitration Mode 1 |
| 19 | Energy Harvesting |
| 21 | Silent Mode |
| 21 | Memory Protection |
| 21 | Passive Wake-Up |
| 22 | Chip Kill |
| 22 | NFC Tag Functionality |
| 22 | Communication Principle |
| 25 | NFC Forum Type 2 Tag Support |
| 26 | UID Coding |
| 27 | Coding of SENS_RES, SEL_REQ, ACK and NACK |
| 27 | SENS_RES Response |
| 27 | SEL_RES Response, Cascade Level 1 and 2 |
| 28 | ACK Response |
| 28 | NACK Response |
| 28 | Access to UID, SENS_RES and SEL_REQ During Anti-Collision |
| 28 | Get Version Command |
| 30 | Memory Organization |
| 30 | 4kbit EEPROM Organization |
| 31 | UID Bytes |
| 32 | Fabrication Data |
| 35 | Capability Container |
| 36 | Configuration Bytes |
| 46 | AS3955 Communication Modes |
| 46 | Standalone NFC Type 2 Tag Mode |
| 46 | Tunneling Mode |
| 48 | Extended Mode |
| 56 | Error Handling |
| 60 | Wired Interfaces |

| | |
|------------|--|
| 60 | SPI / I ² C Access Modes |
| 61 | SPI Interface |
| 69 | I ² C Interface |
| 75 | Interrupt Interface Description |
| 75 | Buffer Interrupts and Buffer Status Register |
| 76 | EEPROM Read and Write |
| 76 | Data Buffer |
| 77 | Direct Commands |
| 79 | Register Description |
| 79 | Registers Overview |
| 80 | IO Configuration Register |
| 80 | IC Configuration Registers |
| 82 | Status Display Registers |
| 83 | Interrupt Registers |
| 85 | Buffer Registers |
| 86 | NFC Last Address Register |
| 87 | Version Control Register |
| 88 | Application Information |
| 88 | Writing a NDEF Message into AS3955 Memory |
| 89 | Reading/Writing Through NFC Device |
| 90 | Reading/Writing Through SPI / I ² C |
| 90 | Using Extended Mode to Switch AS3955 into Tunneling Mode |
| 91 | Using Tunneling Mode to Emulate a NFC Type 4 Tag |
| 92 | References |
| 93 | Package Drawings & Markings |
| 98 | Ordering & Contact Information |
| 99 | RoHS Compliant & ams Green Statement |
| 100 | Copyrights & Disclaimer |
| 101 | Document Status |
| 102 | Revision Information |