AN4
QCA7000 SPI / UART Protocol

I2SE GmbH: Christian Aurich, Stefan Wahren

February 7, 2017

Contents

1 Revisions

4	February 7, 2017	additionally describe UART interface to QCA7000
3	April 1, 2016	add SPI data width, add diagrams, add OSI model, fix initial setup fix interrupt handling, avoid misunderstandings
2	April 21, 2015	added SPI CLK speed, changed EOF fields of transmit frames to length 2
1	September 30, 2014	initial release

2 Introduction

This application note aims to describe the QCA7000 PLC Chip SPI / UART interface. The documentation was derived from the original Qualcomm Atheros Linux driver for the QCA7000, which was initially published at https://github.com/IoE/qca7000. All information within this document can be derived from this code.
The basis for this document is the status of the driver at the time this document was released initially. It's focus is on layers 1 and 2 of the OSI model.

OSI layer	**Protocol**	**Software component**
7 - Application	Application	Application
6 - Presentation		
5 - Session		
4 - Transport	TCP / UDP	TCP / UDP stack
3 - Network	IP v4 / v6	IP stack
2 - Data link	Ethernet	Ethernet over SPI / UART protocol driver
1 - Physical	SPI / UART	SPI / UART driver

I2SE GmbH does not give any free support for driver development. Please contact us if you want to know conditions of our paid support.
Note: During development we strongly suggest to have a logic analyzer with the following features:

- at least 5 channels
- at least 24 MHz sampling rate
- SPI / UART decoding

3 SPI Physical Layer

The QCA7000 acts as a SPI slave and uses Mode 3: CPOL=1, CPHA=1.
SPI data width is 8 bit. The SPI CLK period should not be less than 83.3 ns.
The SPI should be used in burst mode, meaning that the chip select is held low during a complete SPI message.
Note: The SPI lines between Host CPU and QCA7000 should be kept as short as possible.

4 UART Physical Layer

The QCA7000 has a 4 wire UART interface (TX, RX, CTS, RTS) where the hardware flow control is optional. Several UART settings are possible and are configurable with tools that are available under NDA from Qualcomm. I2SE chose the settings in Table 4 as a default for all UART interfaced modules.

Setting	Value
Baud Rate	115200
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	None

Table 4: Default UART Settings

5 SPI Command Structure

The general function of the SPI protocol is to first send a two-byte command and then send or receive the associated data. The following interpretation is the same like on the SPI: most significant bit first.

BIT	15	14	13-0
MEANING	Read/Write	Internal/External	Register

- Read/Write: read = 1 / write = 0
- Internal/External: Internal = 1 / External = 0, internal is used to read or write SPI registers of the local QCA7000, external is used to read or write Ethernet data.
- Register:
  - if Internal/External = 1: the register to read from or write to as defined in the list of QCA7000 SPI registers
  - if Internal/External = 0: the complete register must be zero

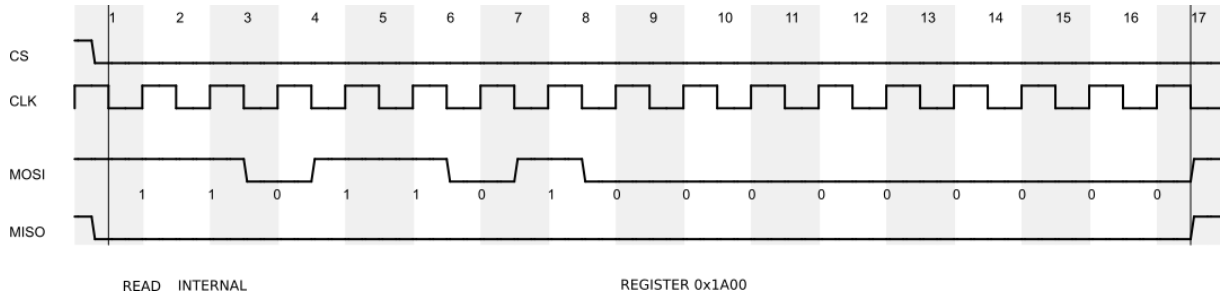

Figure 1: Example - Read SPI_REG_SIGNATURE 1/2

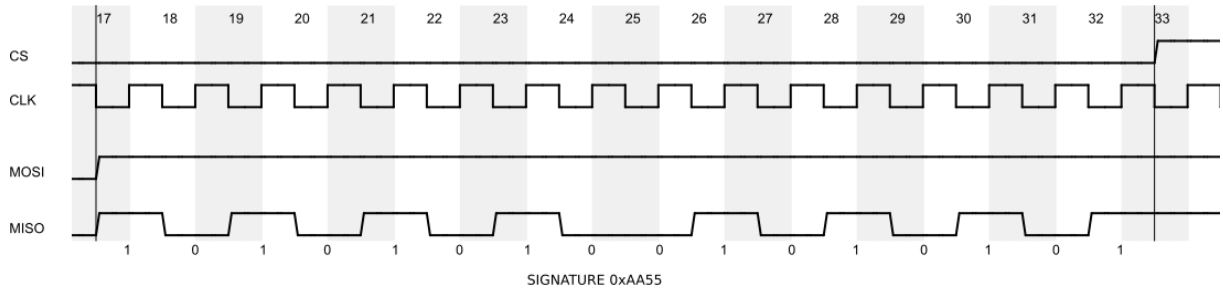

Figure 2: Example - Read SPI_REG_SIGNATURE 2/2

5.1 QCA7000 SPI registers

REG Name	Register Address	Access	Value width
SPI_REG_BFR_SIZE	0x0100	W	16 bit
SPI_REG_WRBUF_SPC_AVA	0x0200	R	16 bit
SPI_REG_RDBUF_BYTE_AVA	0x0300	R	16 bit
SPI_REG_SPI_CONFIG	0x0400	R/W	16 bit
SPI_REG_INTR_CAUSE	0x0C00	R/W	16 bit
SPI_REG_INTR_ENABLE	0x0D00	R/W	16 bit
SPI_REG_SIGNATURE	0x1A00	R	16 bit

NOTE: The column access is derived from the actual function of the original driver. It may be that some registers can also be accessed otherwise.

5.1.1 SPI_REG_BFR_SIZE

This register has to be set before reading or writing external data. This register needs to contain the byte count of the data to be accessed over SPI.

5.1.2 SPI_REG_WRBUF_SPC_AVA

This register can be read to determine how much space is available in the write buffer. This contains the byte count of data that can be written via write external command.
This register should have the value 0x0C5B (3163 bytes) after a reset, as long as no data was sent before via external write command.
Reserving more data with SPI_REG_BFR_SIZE than available via SPI_REG_WRBUF_SPC_AVA isn't allowed.

5.1.3 SPI_REG_RDBUF_BYTE_AVA

This register can be read to determine how much bytes are available in the read buffer. This contains the byte count of data that can be read via read external command.
Requesting more data with SPI_REG_BFR_SIZE than available via SPI_REG_RDBUF_BYTE_AVA isn't allowed.

5.1.4 SPI_REG_SPI_CONFIG

This register contains ,,settings" of the SPI interface.
Meaning of bits:

```
QCASPI_SLAVE_RESET_BIT    (1 << 6)
```

enabling this bit causes the QCA7000 to reset
All the other bits should be kept. So this register should be used in a ,,read, modify, write" cycle.

5.1.5 SPI_REG_INTR_CAUSE

This register can be queried to find out the reason of an hardware interrupt on the dedicated SPI interrupt line.
Meaning of bits (same as SPI_REG_INTR_ENABLE):

```
SPI_INT_CPU_ON          (1 << 6)
SPI_INT_WRBUF_ERR       (1 << 2)
SPI_INT_RDBUF_ERR       (1 << 1)
SPI_INT_PKT_AVLBL       (1 << 0)
```

This register can also be written to to confirm the SPI interrupt.

5.1.6 SPI_REG_INTR_ENABLE

This register contains the enable flags for different SPI interrupts.
Meaning of bits (same as SPI_REG_INTR_CAUSE):

```
SPI_INT_CPU_ON          (1 << 6)
SPI_INT_WRBUF_ERR       (1 << 2)
SPI_INT_RDBUF_ERR       (1 << 1)
SPI_INT_PKT_AVLBL       (1 << 0)
```

5.1.7 SPI_REG_SIGNATURE

This register can be used to ensure the correct function of the SPI interface and that the host handles the endianess correctly. It always contains 0xAA55.

6 Framing for Ethernet Packets

All Ethernet packets must framed as described in this chapter. Additional for the SPI interface these frames must be encapsulated in external read / write operations.

6.1 Transmit Frame

The SPI / UART transmit framing is used for data packets sent from the host processor to the QCA7000.

Offset	Length	Symbol	Description
0x0000	4	SOF	Start Of Frame. Must be 0xAAAAAAAA.
0x0004	2	FL	The Ethernet frame length in little endian format. The frame starts at offset 0x0008 and includes all fields up to but excluding EOF. The minimum is 60. The maximum is 1518 if VLAN is omitted and 1522 if not.
0x0006	2	RSVD	Must be 0x0000. Reserved to ensure 4-byte frame alignment.
0x0008	60-1522	ETH	Frame Body (Layer 2 Ethernet frame without FCS), contains Destination Address, Source Address, (VLAN Tag), Ethertype and the actual data
0x004A to 0x5F8	2	EOF	End Of Frame. Must be 0x5555. This field starts at offset 0x0008 plus FL.

Note: In case the frame body is smaller than 60 bytes it needs to be padded with zero.

6.2 SPI Receive Frame

The SPI receive framing is used for data packets sent from the QCA7000 to the host processor.

Offset	Length	Symbol	Description
0x0000	4	LEN	Hardware generated packet length.
0x0004	4	SOF	Start Of Frame. Must be 0xAAAAAAAA.
0x0008	2	FL	The Ethernet frame length in little endian format. The frame starts at offset 0x000C and includes all fields up to but excluding EOF. The minimum is 60. The maximum is 1518 if VLAN is omitted and 1522 if not.
0x000A	2	RSVD	Must be 0x0000. Reserved to ensure 4-byte frame alignment.
0x000C	60-1522	ETH	Frame Body (Layer 2 Ethernet frame without FCS), contains Destination MAC Address, Source MAC Address, optional VLAN Tag, Ethertype and the actual data
0x0008 plus FL (0x004A to 0x5F8)	2	EOF	End Of Frame. Must be 0x5555.

6.3 UART Receive Frame

The UART receive framing is used for data packets sent from the QCA7000 to the host processor.

Offset	Length	Symbol	Description
0x0000	4	SOF	Start Of Frame. Must be 0xAAAAAAAA.
0x0004	2	FL	The Ethernet frame length in little endian format. The frame starts at offset 0x000C and includes all fields up to but excluding EOF. The minimum is 60. The maximum is 1518 if VLAN is omitted and 1522 if not.
0x0006	2	RSVD	Must be 0x0000. Reserved to ensure 4-byte frame alignment.
0x0008	60-1522	ETH	Frame Body (Layer 2 Ethernet frame without FCS), contains Destination MAC Address, Source MAC Address, optional VLAN Tag, Ethertype and the actual data
0x0004 plus FL (0x004A to 0x5F8)	2	EOF	End Of Frame. Must be 0x5555.

7 Recommended Usage of the SPI

7.1 Initial Setup after each QCA7000 reset

1. read SPI_REG_SIGNATURE and ignore the response
2. read SPI_REG_SIGNATURE again and make sure it matches the defined value of 0xAA55
3. set the SPI_REG_INTR_ENABLE and enable the following interrupts: SPI_INT_CPU_ON, SPI_INT_PKT_AVLBL, SPI_INT_RDBUF_ERR, SPI_INT_WRBUF_ERR

7.2 Handling SPI Interrupts

The QCA7000 has an interrupt signal to raise a flag if it has something to tell to the host CPU. If the host CPU captures a raising edge on the interrupt signal (or if it was missed and a static high level is observed) all interrupts should be disabled by writing zero into the SPI_REG_INTR_ENABLE register first. After that the SPI_REG_INTR_CAUSE register can be read. In this register the QCA7000 tells why the interrupt was raised.
Once the content of the interrupt cause register was received the interrupt cause should confirmed by writing the exact same content back. This resets all interrupt causes - so make sure to react on all of them correctly. Resetting of all interrupt causes results in the SPI interrupt signal transitioning back to low. Finally re-enable all desired interrupts in the SPI_REG_INTR_ENABLE register.

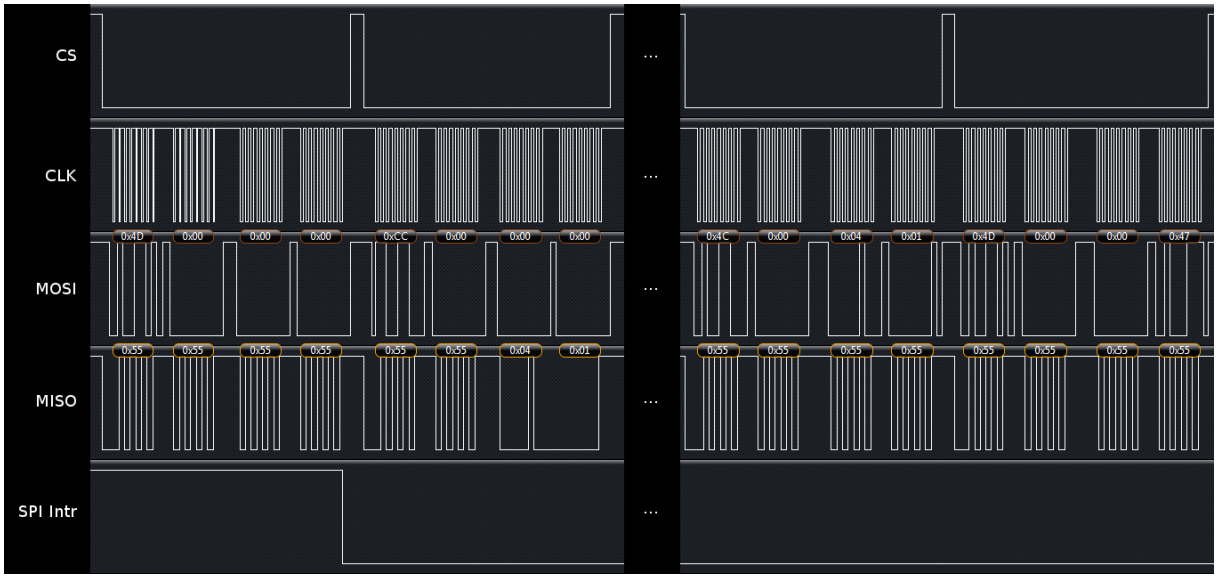

Figure 3: Example - Interrupt handling

Reaction for interrupt causes:

Interrupt cause	Recommended reaction
SPI_INT_CPU_ON	the QCA7000 just performed a startup, see ,,initial setup”
SPI_INT_WRBUF_ERR	perform a QCA7000 restart via QCASPI_SLAVE_RESET_BIT
SPI_INT_RDBUF_ERR	perform a QCA7000 restart via QCASPI_SLAVE_RESET_BIT
SPI_INT_PKT_AVLBL	perform an external read

7.3 External Read

To read the data from the QCA7000 via external read the following steps should be performed:

1. read register SPI_REG_RDBUF_BYTE_AVA, this gives the length of the available data
2. write register SPI_REG_BFR_SIZE, this sets the length of data to be read via ,,external read”
3. start an external read and receive as much data as set in SPI_REG_BFR_SIZE before

Note: The SPI chip select must be kept low between the external read command and data receiving.

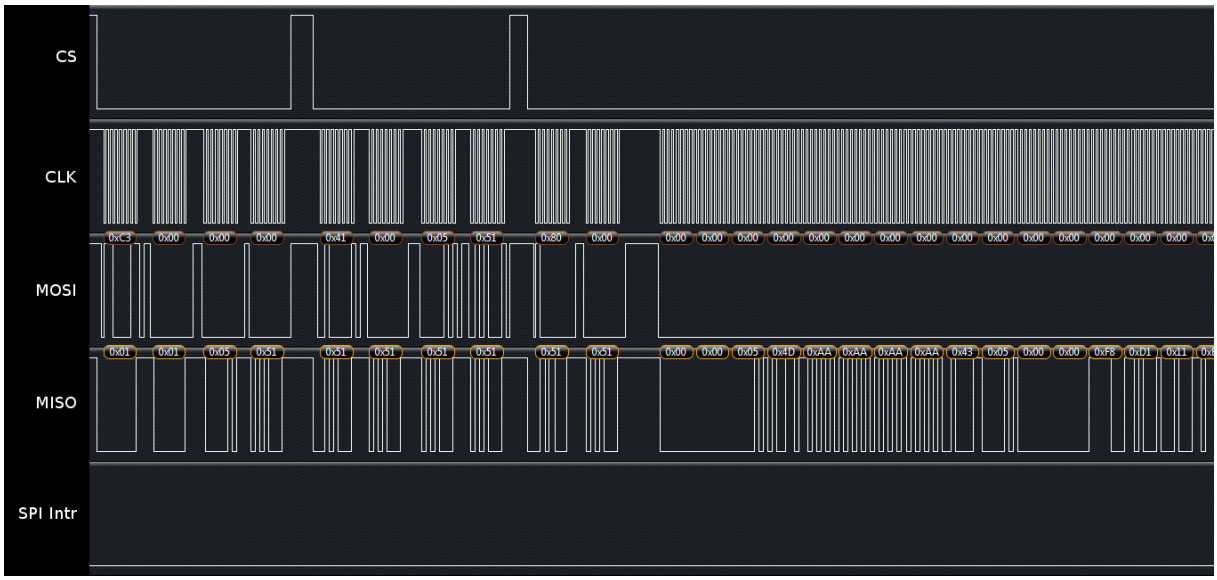

Figure 4: Example - External read

7.4 External Write

To write the data to the QCA7000 via external write following steps should be performed:

1. calculate value for SPI_REG_BFR_SIZE (value = SOF + FL + RSVD + ETH + EOF)
2. read register SPI_REG_WRBUF_SPC_AVA to find out how much space is available in the QCA7000
3. write register SPI_REG_BFR_SIZE to set the length of data to be sent via the next ,,external write”
4. start an external write and send as much data as set in SPI_REG_BFR_SIZE before.

Note: The SPI chip select must be kept low between the external write command and data writing.

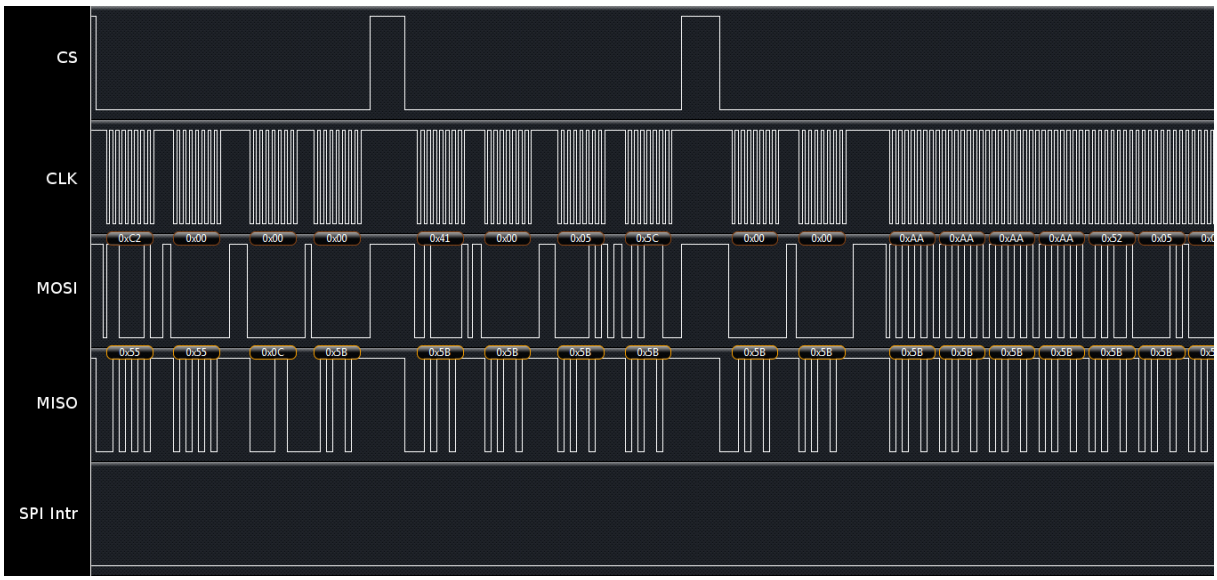

Figure 5: Example - External write

8 Troubleshooting

If your SPI communication is not working you should try the following:

- keep the SPI signal length as short as possible, preferably under 10 cm
- if the response of the QCA7000 for the signature register seems to be incorrect you should try to add series termination resistors (between 10 and 20 Ohm) in the SPI clock and data lines

9 Appendix

1. UART/SPI drivers for Qualcomm Atheros QCA7000 serial-to-powerline bridge chip. This version is specific to Linux 2.6.35 and Freescale iMX28 CPU. (separate as an4_qca7000_driver_30_09_2014.zip)