AH1601

Device Configuration Application Hint

Rev. 01.01 — 28 July 2016 **Application Hint**

Document information

Info	Content
Keywords	SPI, static configuration, generic loader format, CGU
Abstract	This document gives a brief overview of how to configure the SJA1105. This includes the clocking and loading the configuration.

Revision history

Rev	Date	Description
0.01	20160315	Initial draft
0.02	20160322	Fixed CFG_PAD_MII0_xxx name
0.03	20160421	Fixed CFG_PAD_xxx addresses in RGMII example
1.00	20160622	Release
1.01	20160728	Fixed Table 13

Contact information

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

1. Introduction

1.1 Scope and purpose of the document

The main aim of this document is to provide an overview and understanding of the resources required by software designers to get started with the SJA1105 device and configure the device to bring it to operation.

This includes programming the clock generation unit (CGU) and loading a static configuration.

2. Clock Generation Unit (CGU)

Loading the static configuration into the device alone is not sufficient to send or receive traffic. Additionally, the clocking of the xMII interfaces has to be set up through the CGU.

In order to get the switch functional for handling traffic and application development, the following steps are necessary.

The CGU (cf. Fig 1) is organized as a matrix, which can forward input clocks (e.g. from a physical clock input such as RXC or a PLL) to various clock sinks. Clock sinks can have multiple purposes. In Fig 1, they appear colored to indicate in which mode they are relevant. Some sinks are used to clock internal parts of the device (e.g. the MII logic, which is shown in violet), while other sinks drive clock pins (e.g. TXC).

Clock sinks and clock sources (except the IO pins) can be configured over a control register. For sinks, this allows to select the source (CLKSRC field) and for sources, it allows to control advanced functionality such as the settings for the integer dividers (IDIVs), which are explained later.

This is a list of all registers and addresses in the CGU.

Table 1. CGU register overview

Offset	Name	Access	Reset Value
100007h	PLL_0_S	R	00000000h
100009h	PLL_1_S	R	00000000h
10000ah	PLL_1_C	R/W	0A000003h
10000bh	IDIV_0_C	R/W	0A000000h
10000ch	IDIV_1_C	R/W	0A000000h
10000dh	IDIV_2_C	R/W	0A000000h
10000eh	IDIV_3_C	R/W	0A000000h
10000fh	IDIV_4_C	R/W	0A000000h
100013h	MII_TX_CLK_0	R/W	11000000h
100014h	MII_RX_CLK_0	R/W	11000000h
100015h	RMII_REF_CLK_0	R/W	0E000000h
100016h	RGMII_TX_CLK_0	R/W	11000000h
-	RGMII_RX_CLK_0[1]	-	-
100018h	EXT_TX_CLK_0	R/W	11000000h
100019h	EXT_RX_CLK_0	R/W	11000000h

[1] In MRA2 silicon version, the register RGMII_RX_CLK_x is reserved. This RGMII sink is always hardwired to RXC and no setup is needed. In MRA1 silicon version, this register does exist and needs to be programmed.

Offset	Name	Access	Reset Value
10001Ah	MII_TX_CLK_1	R/W	12000000h
10001Bh	MII_RX_CLK_1	R/W	12000000h
10001Ch	RMII_REF_CLK_1	R/W	0E000000h
10001Dh	RGMII_TX_CLK_1	R/W	12000000h
-	RGMII_RX_CLK_1	-	-
10001Fh	EXT_TX_CLK_1	R/W	12000000h
100020h	EXT_RX_CLK_1	R/W	12000000h
100021h	MII_TX_CLK_2	R/W	13000000h
100022h	MII_RX_CLK_2	R/W	13000000h
100023h	RMII_REF_CLK_2	R/W	0E000000h
100024h	RGMII_TX_CLK_2	R/W	13000000h
-	RGMII_RX_CLK_2	-	-
100026h	EXT_TX_CLK_2	R/W	13000000h
100027h	EXT_RX_CLK_2	R/W	13000000h
100028h	MII_TX_CLK_3	R/W	14000000h
100029h	MII_RX_CLK_3	R/W	14000000h
10002Ah	RMII_REF_CLK_3	R/W	0E000000h
10002Bh	RGMII_TX_CLK_3	R/W	14000000h
-	RGMII_RX_CLK_3	-	-
10002Dh	EXT_TX_CLK_3	R/W	14000000h
10002Eh	EXT_RX_CLK_3	R/W	14000000h
10002Fh	MII_TX_CLK_4	R/W	15000000h
100030h	MII_RX_CLK_4	R/W	15000000h
100031h	RMII_REF_CLK_4	R/W	0E000000h
100032h	RGMII_TX_CLK_4	R/W	15000000h
-	RGMII_RX_CLK_4	-	-
100034h	EXT_TX_CLK_4	R/W	15000000h
100035h	EXT_RX_CLK_4	R/W	15000000h

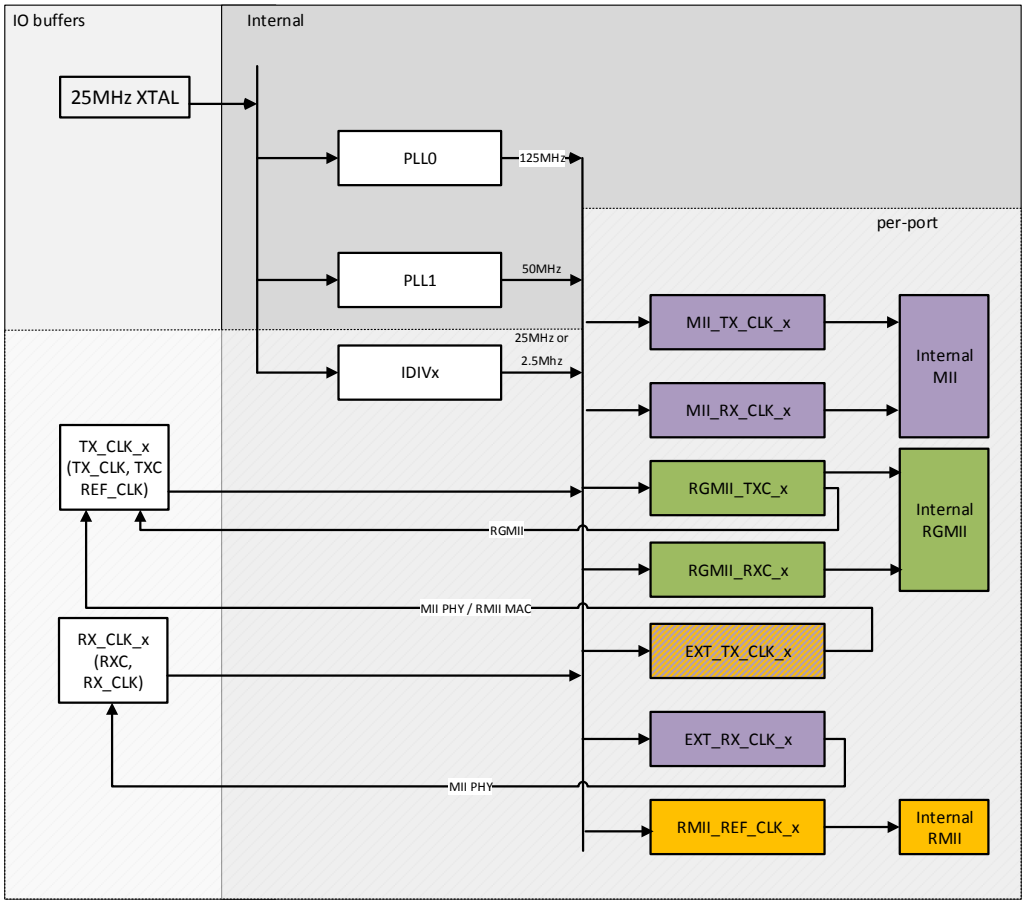

Fig 1. Generic CGU Overview

2.1 Clock Sources

The CGU consists of the following sources (once per SJA1105). The value in brackets indicate the clock selection value, which is used to select the source. Boldfaced names correspond to selectable sources in the CGU.

- OSC
  A fixed 25 Mhz XTAL oscillator. This cannot be directly selected as a source.
- **PLL0 (0Bh)**
  A fixed 125 Mhz PLL which is used to drive the digital core clock.
- **PLL1 (0Eh)**
  A fixed 50 Mhz PLL.

Next to these clock sources, the switch has the following per port sources and sinks. The value in brackets indicate the clock selection value, which is used to select the source. For instance 2h*x means a value of 2h for port 0 and a value of 4h for port 1 and so on.

- IO pad (TXC, REF_CLK, TX_CLK) referred to as **TX_CLK_x (02h*x)**
  This can operate either as an output (RGMII, RMII-MAC, MII PHY) or as input (RMII-PHY, MII-MAC).

- IO pad (RXC, RX_CLK) referred to as **RX_CLK_x (02h*x + 01h)**
  This can operate either as an output (RGMII, RMII-MAC, MII PHY) or as input (RMII-PHY, MII-MAC).
- **IDIVx (11h+x)**
  An integer divider (by 1 to 10). It is used to feed through 25 Mhz or generate 2.5 Mhz.

2.2 Clock Sinks

- **MII_TX_CLK_x**
  A clock sink, that is connected with the internal MII logic and drives the transmitting side of the MII logic of port x.
- **MII_RX_CLK_x**
  A clock sink, that is connected with the internal MII logic and drives the receiving side of the MII logic of port x.
- **RGMII_TXC_x**
  A clock sink, that is connected with the internal RGMII logic and drives the transmitting side of the RGMII logic of port x.
- RGMII_RXC_x *(hardwired)*
  A clock sink, that is connected with the internal RGMII logic and drives the transmitting side of the RGMII logic of port x.
- **EXT_TX_CLK_x**
  A clock sink, that is connected to the TXC/REF_CLK/TX_CLK IO pad and can, depending on the mode (MAC/PHY), drive the pad.
- **EXT_RX_CLK_x**
  A clock sink that is connected to the RXC/TX_CLK IO pad and can, depending on the mode (MAC/PHY) drive, the pad.
- **RMII_REF_CLK_x**
  A clock sink, that is connected with the internal RMII logic of port x. In RMII-MAC mode, the EXT_TX_CLK_x will drive the REF_CLK pin.

Each clock sink register (except RGMII_RXC_x) has a corresponding register to select the clock source. The following table specifies the register layout.

Table 2. Clock sink registers 1 to 30 (addr. 100013h to 100035h)

Bits	Symbol	Access	Description
28:24	CLKSRC[2]	R/W	Internal clock source selection
23:12	reserved	R/W	0h
11	AUTOBLOCK	R/W	Blocking the output when the source is changed to prevent glitches. Write 1 when changing CLKSRC.
10:1	reserved	R/W	0h
0	PD	R/W	Power down

[2] Not all CLKSRCs can be selected for all clock sinks. The user manual provides a list of allowed selections.

2.3 Bringing a Port in MII-MAC Mode

It must be ensured that the xMII configuration block is setup correctly to bring the given port into MII MAC mode. The following Figure illustrates the clocking scheme for the CGU only.

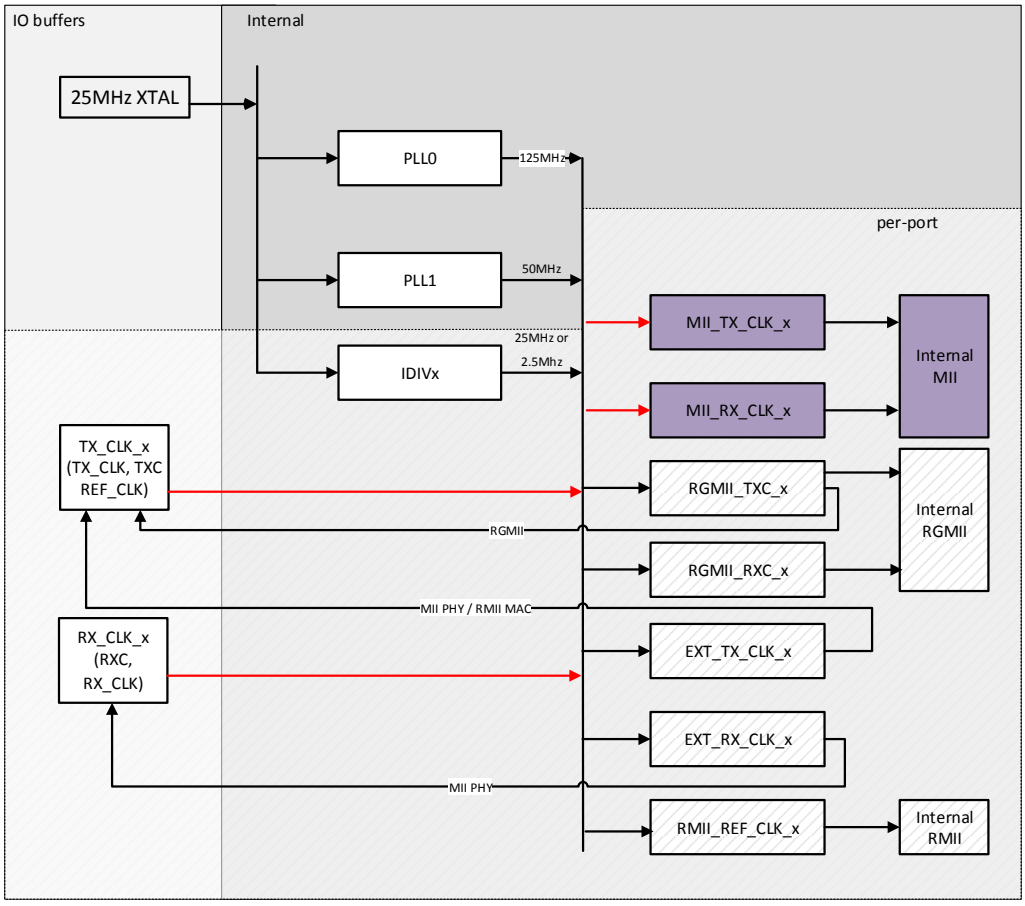

Fig 2. CGU in MII-MAC configuration

2.3.1 Sources

No configuration required. To reduce the power consumption the IDIVx can be disabled (PD=1).

2.3.2 Sinks

Table 3. Source to sink mapping

Sink	Source	CLKSRC
MII_TX_CLK_x	TX_CLK_x	02h*x
MII_RX_CLK_x	RX_CLK_x	02h*x+1h

Table 4. Register Settings for 100 Mbps MII-MAC

Entity	Register	Address	Value	Description
General	-			
Port 0	IDIV0	0x0010000B	0x0A000001	Disable IDIV0
	MII_TX_CLK_0	0x00100013	0x00000800	Setting CLKSRC of MII_TX_CLK_0 to TX_CLK_0
	MII_RX_CLK_0	0x00100014	0x01000800	Setting CLKSRC of MII_RX_CLK_0 to RX_CLK_0
1	IDIV1	0x0010000C	0x0A000001	Disable IDIV1
	MII_TX_CLK_1	0x0010001A	0x02000800	Setting CLKSRC of MII_TX_CLK_1 to TX_CLK_1
	MII_RX_CLK_1	0x0010001B	0x03000800	Setting CLKSRC of MII_RX_CLK_1 to RX_CLK_1
2	IDIV2	0x0010000D	0x0A000001	Disable IDIV2
	MII_TX_CLK_2	0x00100021	0x04000800	Setting CLKSRC of MII_TX_CLK_2 to TX_CLK_2
	MII_RX_CLK_2	0x00100022	0x05000800	Setting CLKSRC of MII_RX_CLK_2 to RX_CLK_2
3	IDIV3	0x0010000E	0x0A000001	Disable IDIV3
	MII_TX_CLK_3	0x00100028	0x06000800	Setting CLKSRC of MII_TX_CLK_3 to TX_CLK_3
	MII_RX_CLK_3	0x00100029	0x07000800	Setting CLKSRC of MII_RX_CLK_3 to RX_CLK_3
4	IDIV4	0x0010000F	0x0A000001	Disable IDIV4
	MII_TX_CLK_0	0x0010002F	0x06000800	Setting CLKSRC of MII_TX_CLK_4 to TX_CLK_4
	MII_RX_CLK_0	0x00100030	0x07000800	Setting CLKSRC of MII_RX_CLK_4 to RX_CLK_4

2.4 Bringing a Port in MII-PHY Mode

It must be ensured that the xMII configuration block is setup correctly to bring the given port into MII PHY mode. The following Figure illustrates the clocking scheme for the CGU only.

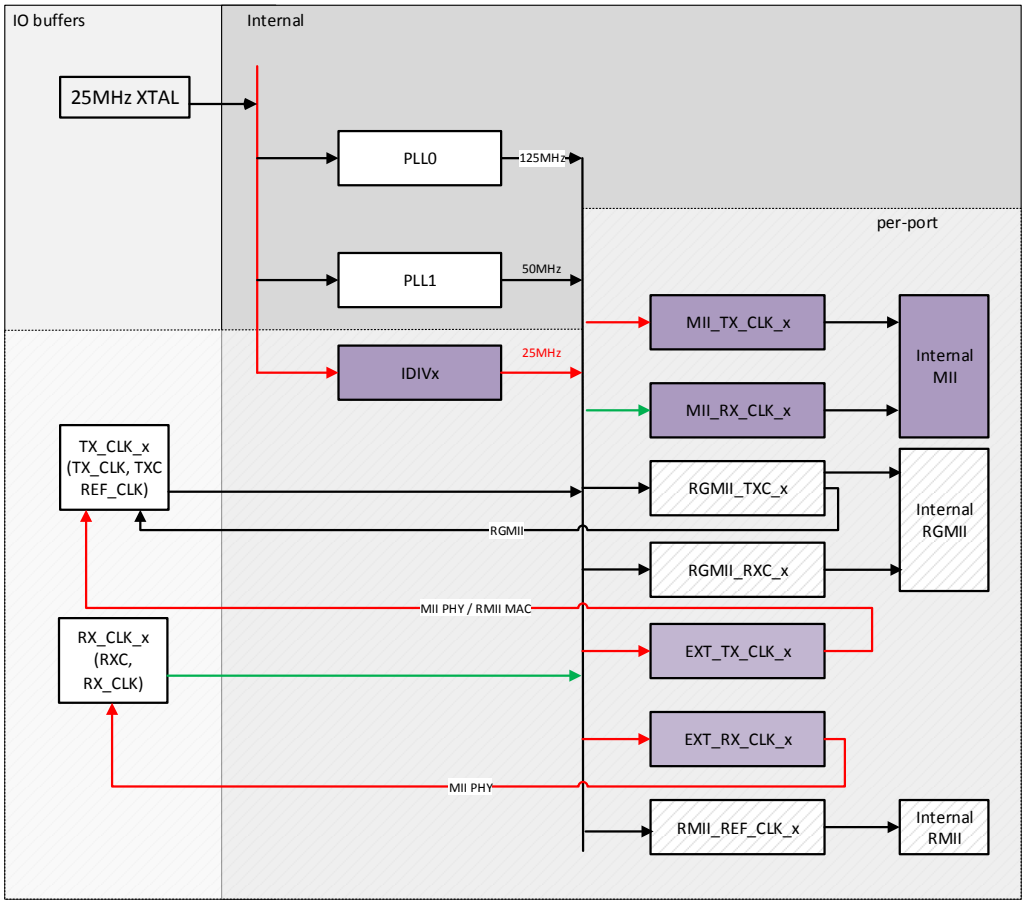

Fig 3. CGU in MII-PHY configuration

2.4.1 Sources

No configuration required

2.4.2 Sinks

Table 5. Source to sink mapping

Sink	Source	CLKSRC
MII_TX_CLK_x	IDIVx	11h+x
MII_RX_CLK_x	RX_CLK_x	02h*x+1h
EXT_TX_CLK_x	IDIVx	11h+x
EXT_RX_CLK_x	IDIVx	11h+x

Table 6. Register Settings for 100 Mbps MII-PHY

Entity	Register	Address	Value	Description
General	-			
Port 0	IDIV0	0x0010000B	0x0A000000	Enable IDIV0 and divide by 1.
	MII_TX_CLK_0	0x00100013	0x11000800	CLKSRC of MII_TX_CLK_0 to IDIV0
	MII_RX_CLK_0	0x00100014	0x01000800	Setting CLKSRC of MII_RX_CLK_0 to RX_CLK_0
	EXT_TX_CLK_0	0x00100018	0x11000800	Setting CLKSRC of EXT_TX_CLK_0 to IDIV0
	EXT_RX_CLK_0	0x00100019	0x11000800	Setting CLKSRC of EXT_TX_CLK_0 to IDIV0
Port 1	IDIV1	0x0010000C	0x0A000000	Enable IDIV1 and divide by 1.
	MII_TX_CLK_1	0x0010001A	0x12000800	CLKSRC of MII_TX_CLK_1 to IDIV1
	MII_RX_CLK_1	0x0010001B	0x03000800	Setting CLKSRC of MII_RX_CLK_1 to RX_CLK_1
	EXT_TX_CLK_1	0x0010001F	0x12000800	Setting CLKSRC of EXT_TX_CLK_1 to IDIV1
	EXT_RX_CLK_1	0x00100020	0x12000800	Setting CLKSRC of EXT_TX_CLK_1 to IDIV1
Port 2	IDIV2	0x0010000D	0x0A000000	Enable IDIV2 and divide by 1.
	MII_TX_CLK_2	0x00100021	0x13000800	CLKSRC of MII_TX_CLK_2 to IDIV2
	MII_RX_CLK_2	0x00100022	0x05000800	Setting CLKSRC of MII_RX_CLK_2 to RX_CLK_2
	EXT_TX_CLK_2	0x00100026	0x13000800	Setting CLKSRC of EXT_TX_CLK_2 to IDIV2
	EXT_RX_CLK_2	0x00100027	0x13000800	Setting CLKSRC of EXT_TX_CLK_2 to IDIV2
Port 3	IDIV3	0x0010000E	0x0A000000	Enable IDIV3 and divide by 1.
	MII_TX_CLK_3	0x00100028	0x14000800	CLKSRC of MII_TX_CLK_3 to IDIV3
	MII_RX_CLK_3	0x00100029	0x07000800	Setting CLKSRC of MII_RX_CLK_3 to RX_CLK_3
	EXT_TX_CLK_3	0x0010002D	0x14000800	Setting CLKSRC of EXT_TX_CLK_3 to IDIV3
	EXT_RX_CLK_3	0x0010002E	0x14000800	Setting CLKSRC of EXT_TX_CLK_3 to IDIV3
Port 4	IDIV4	0x0010000F	0x0A000000	Enable IDIV4 and divide by 1.
	MII_TX_CLK_4	0x0010002F	0x15000800	CLKSRC of MII_TX_CLK_4 to IDIV4
	MII_RX_CLK_4	0x00100030	0x09000800	Setting CLKSRC of MII_RX_CLK_4 to RX_CLK_4
	EXT_TX_CLK_4	0x00100034	0x15000800	Setting CLKSRC of EXT_TX_CLK_4 to IDIV4

Entity	Register	Address	Value	Description
	EXT_RX_CLK_4	0x00100035	0x15000800	Setting CLKSRC of EXT_TX_CLK_4 to IDIV4

2.5 Bringing a Port in RMII-MAC Mode

It must be ensured that the xMII configuration block is setup correctly to bring the given port into MII MAC mode. The following Figure illustrates the clocking scheme for the CGU only.

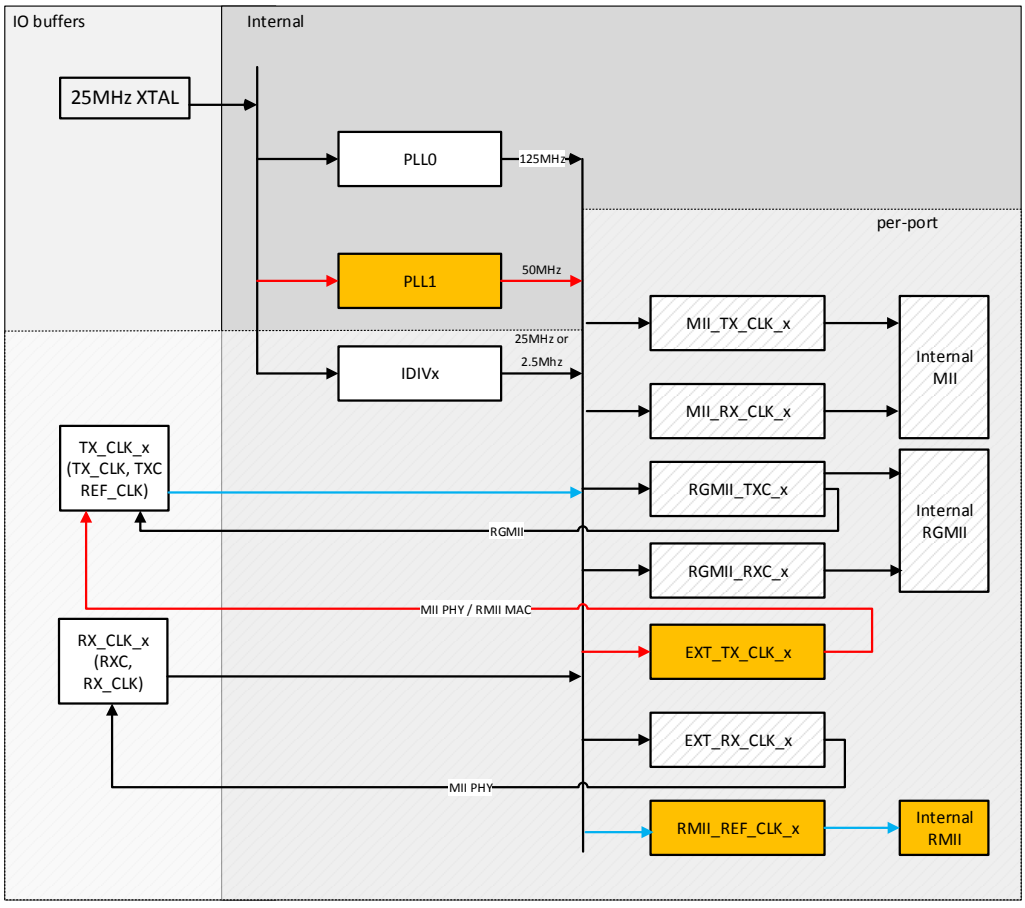

Fig 4. CGU in RMII-MAC configuration

2.5.1 Sources

PLL1 must be enabled and output 50 Mhz. This is done by writing first 0x0A010941 to the **PLL_1_C** register and then deasserting power down (PD) 0x0A010940.

2.5.2 Sinks

Table 7. Source to sink mapping

Sink	Source	CLKSRC
RMII_REF_CLK_x	TX_CLK_x	02h*x

Sink	Source	CLKSRC
EXT_TX_CLK_x	PLL1	0Eh

Table 8. Register Settings for 100 Mbps RMII-MAC

Entity	Register	Address	Value	Description
General	PLL1	0x0010000A	0x0A010941	Step1: PLL1 setup for 50MHz
		0x0010000A	0x0A010940	Step2: Enable PLL1
Port 0	IDIV0	0x0010000B	0x0A000001	Disable IDIV0
	RMII_REF_CLK_0	0x00100015	0x00000800	Setting CLKSRC of RMII_REF_CLK_0 to TX_CLK_0
	EXT_TX_CLK_0	0x00100018	0x0E000800	Setting CLKSRC of EXT_TX_CLK_0 to PLL1
Port 1	IDIV1	0x0010000C	0x0A000001	Disable IDIV1
	RMII_REF_CLK_1	0x0010001C	0x02000800	Setting CLKSRC of RMII_REF_CLK_1 to TX_CLK_1
	EXT_TX_CLK_1	0x0010001F	0x0E000800	Setting CLKSRC of EXT_TX_CLK_1 to PLL1
Port 2	IDIV2	0x0010000D	0x0A000001	Disable IDIV2
	RMII_REF_CLK_2	0x00100023	0x04000800	Setting CLKSRC of RMII_REF_CLK_2 to TX_CLK_2
	EXT_TX_CLK_2	0x00100026	0x0E000800	Setting CLKSRC of EXT_TX_CLK_2 to PLL1
Port 3	IDIV3	0x0010000E	0x0A000001	Disable IDIV3
	RMII_REF_CLK_3	0x0010002A	0x06000800	Setting CLKSRC of RMII_REF_CLK_3 to TX_CLK_3
	EXT_TX_CLK_3	0x0010002D	0x0E000800	Setting CLKSRC of EXT_TX_CLK_3 to PLL1
Port 4	IDIV4	0x0010000F	0x0A000000	Disable IDIV4
	RMII_REF_CLK_4	0x00100031	0x08000800	Setting CLKSRC of RMII_REF_CLK_4 to TX_CLK_4
	EXT_TX_CLK_4	0x00100034	0x0E000800	Setting CLKSRC of EXT_TX_CLK_4 to PLL1

2.6 Bringing a Port in RMII-PHY Mode

It must be ensured that the xMII configuration block is setup correctly to bring the given port into MII MAC mode. The following Figure illustrates the clocking scheme for the CGU only.

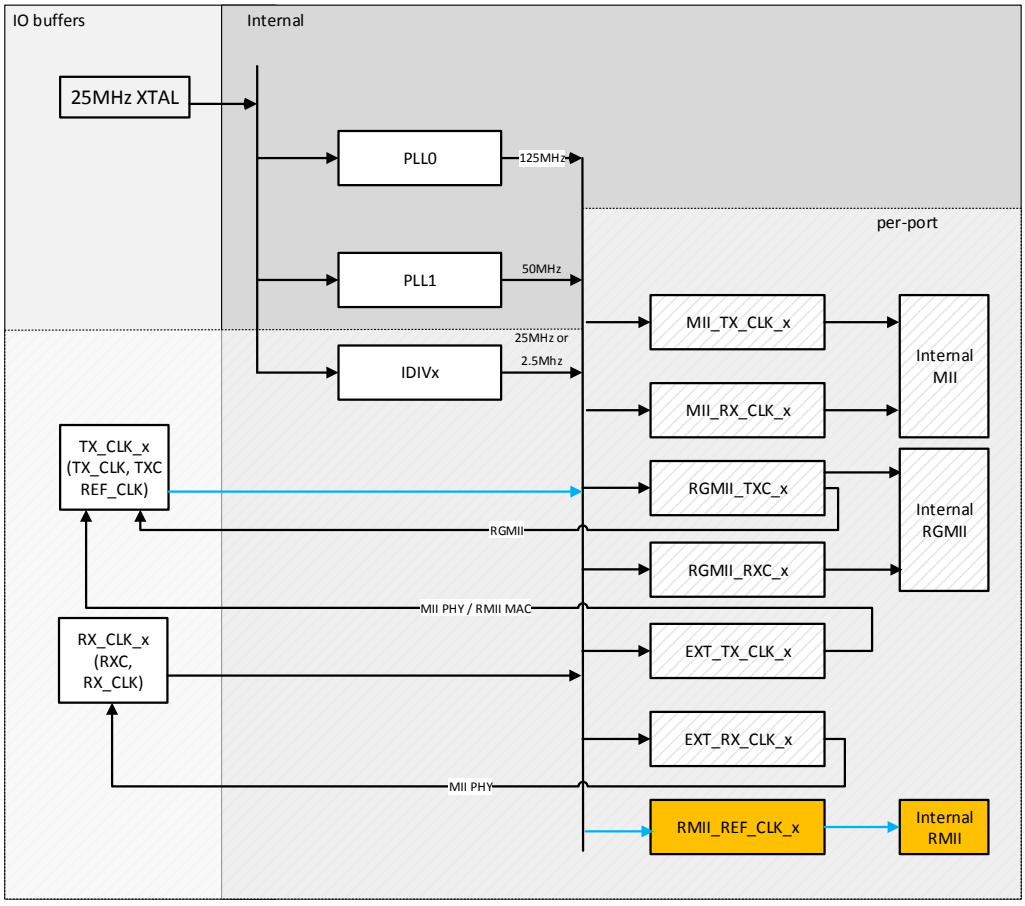

Fig 5. CGU in MII-MAC configuration

2.6.1 Sources

No configuration required. To reduce the power consumption, the IDIVx can be disabled (PD=1).

2.6.2 Sinks

Table 9. Source to sink mapping

Sink	Source	CLKSRC
RMII_REF_CLK_x	TX_CLK_x	02h*x

Table 10. Register Settings for 100 Mbps RMII-PHY

Entity	Register	Address	Value	Description
General	-			
Port 0	IDIV0	0x0010000B	0x0A000001	Disable IDIV0
	RMII_REF_CLK_0	0x00100015	0x00000800	Setting CLKSRC of RMII_REF_CLK_0 to TX_CLK_0
Port 1	IDIV1	0x0010000C	0x0A000001	Disable IDIV1
	RMII_REF_CLK_1	0x0010001C	0x02000800	Setting CLKSRC of RMII_REF_CLK_1

Entity	Register	Address	Value	Description
				to TX_CLK_1
Port 2	IDIV2	0x0010000D	0x0A000001	Disable IDIV2
	RMII_REF_CLK_2	0x00100023	0x04000800	Setting CLKSRC of RMII_REF_CLK_2 to TX_CLK_2
Port 3	IDIV3	0x0010000E	0x0A000001	Disable IDIV3
	RMII_REF_CLK_3	0x0010002A	0x06000800	Setting CLKSRC of RMII_REF_CLK_3 to TX_CLK_3
Port 4	IDIV4	0x0010000F	0x0A000000	Disable IDIV4
	RMII_REF_CLK_4	0x00100031	0x08000800	Setting CLKSRC of RMII_REF_CLK_4 to TX_CLK_4

2.7 Bringing a Port in RGMII Mode

It must be ensured that the xMII configuration block is setup correctly to bring the given port into MII MAC mode. The following Figure illustrates the clocking scheme for the CGU only. For RGMII, the pad speed has to be adjusted in addition. The relevant register writes are included in the sink configuration.

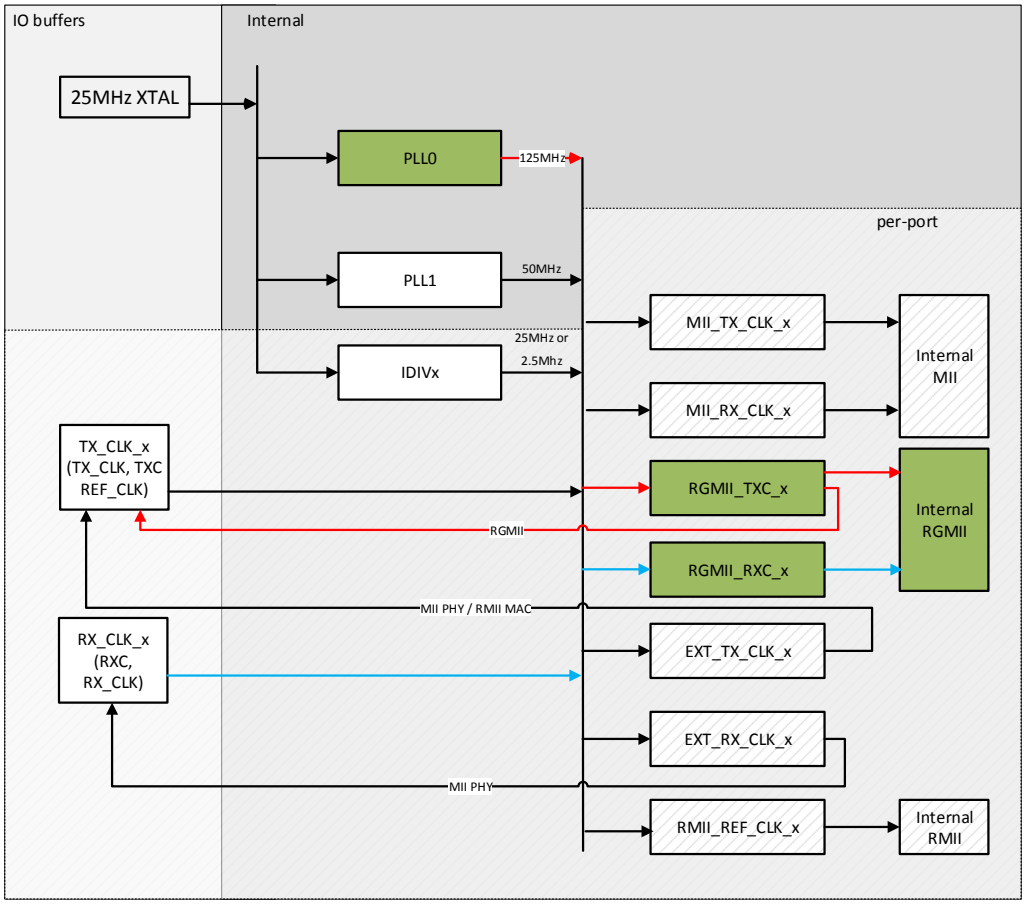

Fig 6. CGU in MII-MAC configuration

2.7.1 Sources

No configuration required. To reduce the power consumption, the IDIVx can be disabled (PD=1).

2.7.2 Sinks

Table 11. Source to sink mapping

Sink	Source	CLKSRC
RGMII_RXC_x	*hardwired*[3]	-
RGMII_TXC_x	PLL0	0Bh

Table 12. Register Settings for 1Gbps RGMII

Entity	Register	Address	Value	Description
General	-			
Port 0	IDIV0	0x0010000B	0x0A000001	Disable IDIV0
	RGMII_TXC_0	0x00100016	0x0B000800	Setting CLKSRC of RGMII_TXC_0 to PLL0
	CFG_PAD_MII0_TX	0x00100800	0x1A1A1A1A	Setting output speed to high speed
Port 1	IDIV1	0x0010000C	0x0A000001	Disable IDIV1
	RGMII_TXC_1	0x0010001D	0x0B000800	Setting CLKSRC of RGMII_TXC_1 to PLL0
	CFG_PAD_MII1_TX	0x00100802	0x1A1A1A1A	Setting output speed to high speed
Port 2	IDIV2	0x0010000D	0x0A000001	Disable IDIV2
	RGMII_TXC_2	0x00100024	0x0B000800	Setting CLKSRC of RGMII_TXC_2 to PLL0
	CFG_PAD_MII2_TX	0x00100804	0x1A1A1A1A	Setting output speed to high speed
Port 3	IDIV3	0x0010000E	0x0A000001	Disable IDIV3
	RGMII_TXC_3	0x0010002B	0x0B000800	Setting CLKSRC of RGMII_TXC_3 to PLL0
	CFG_PAD_MII3_TX	0x00100806	0x1A1A1A1A	
Port 4	IDIV4	0x0010000F	0x0A000000	Disable IDIV4
	RGMII_TXC_4	0x00100032	0x0B000800	Setting CLKSRC of RGMII_TXC_4 to PLL0
	CFG_PAD_MII3_TX	0x00100808	0x1A1A1A1A	Setting output speed to high speed

[3] In MRA1 silicon, setting up RGMII_RXC_x is required.

3. Static Configuration

Configuration information for the switch core needs to be loaded at startup, using the generic loader format. The configuration area starts at address 0x20000. The entire configuration area is write-only. A read access to any address in this area will return arbitrary data. The configuration data is divided into a number of blocks as per Table 1 below.

Table 13. Configuration Blocks

Block Name	Block ID	Loading Mandatory
L2 Address Lookup table	05h	No
L2 Policing table	06h	Yes, atleast one entry
VLAN Lookup table	07h	Yes, at least the default untagging VLAN
L2 Forwarding table	08h	Yes
MAC Configuration table	09h	Yes
L2 Address Lookup parameters	0Dh	No
L2 Forwarding Parameters	0Eh	Yes
AVB Parameters	10h	No
General Parameters	11h	Yes
Retagging table	12h	No
xMII Mode Parameters	4Eh	Yes

Data is loaded into the configuration area as a continuous stream of 32-bit data. The load operation is initiated by writing the device ID (0x9C00000C) to relative address 0 (i.e. relative to the start address, 0x20000). The format for subsequent write operations is illustrated in Figure 1. The configuration data blocks, listed in Table 1, are loaded in turn.

In the context of the following description, 32-bit values are referred to as double-words. The first double-word after the device ID contains the block ID of the first block. The second double-word contains length of the first data block to be loaded (i.e the number of data double-words, excluding the checksum). This is followed by the CRC checksum and the data.

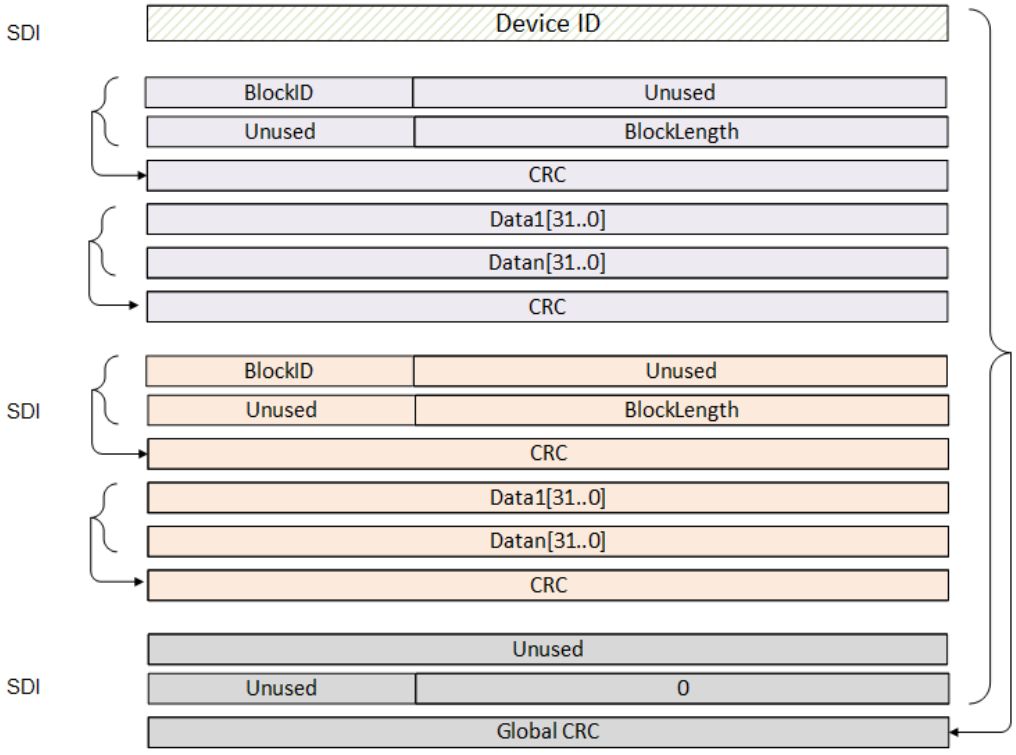

Fig 7. Generic Loader Format

The data blocks can be loaded in any order. For the VLAN Lookup table, for example, the first 8 bits of the first double-word would contain the block id (0x07). The last 24 bits of the second double-word constitutes the BlockLength. It is defined as the number of following double-words in this block, excluding the checksums.
A block length of 0 signals the end of the configuration file and a global CRC is expected to follow. Fields labeled ‘unused’ (e.g bits 0 to 26 in the VLAN Lookup table) are not interpreted by the IP and may be set to any value. However, the values assigned to ‘unused’ fields must be reflected in the checksum.

Checksums are calculated as CRC-32 Ethernet checksums with the lower bytes of each double-word included first in the CRC calculation. See IEEE 802.3-2012, clause 3 for details on how CRC checksums are calculated for Ethernet frames.

The previous paragraphs only describe how the configuration stream is formatted, please refer to Section 4.2 for downloading the static configuration to the device.

3.1 xMII Mode parameters

As discussed, setting up the MDI consists of setting up the CGU and configuring the xMII Mode parameters in the static configuration. Port 0 to port 4 provide MII, RMII, and RGMII options. In PHY mode, the port behaves like a PHY on the Ethernet interface and it is used when the partner on the Ethernet interface is a MAC. In MAC mode the port behaves like a MAC on the Ethernet interface and it is used when the partner on the Ethernet interface is a PHY.

3.2 Hex File Structure

In the NXP switch configuration flow the static configuration is generated as an Intel HEX file. This file contains a classical data to address mapping. A typical record (lline) in the hex file has the following structure:

:040000000E03009F4C

1. Start code (“**:**”): The start of the record is marked by an ASCII colon.
2. Byte Count (**04**): Start code is followed by a 1 byte (two hex digits) of byte count.
3. Address (**0000**): The Byte count is followed by a 2 byte (4 hex digits) address which signifies the offset within the configuration file.
4. Record type (00): The record type is a 1 byte (2 hex digits) in size. This represents the type of data. The record type can have the following possible values:
    a. 00 → Data
    b. 01 → End of file
    c. 02 → Extended segment address
    d. 03 → Start Segment Address
    e. 04 → Extended Linear Address
    f. 05 → Start Linear Address
5. Data (**0E03009F**): The record type is followed by the data. In this case the data is interpreted as 0x9F00030E.
6. Checksum (**4C**): The last byte of the record specifies the checksum. The checksum is the 2s complement of the sum of the bytes of the record. For example 2sComplement(**04+ 00 + 00 + 00+ 0E + 03 + 00 + 9F**) = **4C**

For example let us consider a typical Hex file generated by the configuration generation scripts

The first line of the hex file would be- **:040000000E03009F4C** which decodes to

Table 14. Hex first record dissected

Data	Field Name	Description
:	Start of code	Signifies the start of the Hex file
04	Byte count	4bytes or uint32_t
0000	Address	Offset within the configuration file.
00	Record type	Data
0E03009F	Data	0x9F00030E
4C	Checksum	2’s complement of the sum of the bytes of the record.

Table 15. Hex last record dissected

Data	Field Name	Description
:	Start of code	Signifies the start of the Hex file
00	Byte count	No data
0000	Address	Offset within the configuration file.
01	Record type	End of file
FF	Checksum	2's complement of the sum of the bytes of the record.

4. SPI Framing

The static configuration is loaded over SPI using the SPI framing format used for read/write operations as described in the user manual. The framing format is given in the following figure.

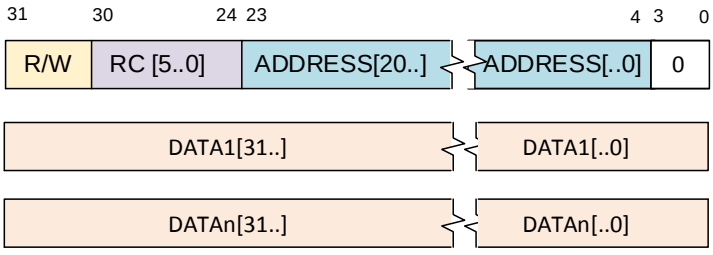

Fig 8. SPI Framing Format

1. Bit 31 (R/W) of the first quadlet (32 bit word) specifies a read (0) or a write (1) action.
2. Bits 30-25 specify the read count (RC) in case of a SPI read. This specifies the number of quadlets (32-bit) to be read **up to 64 quadlets**. For a write access this can be set to zero.
3. Bits 24-4 specify the address (ADDRESS) to be accessed.
4. Bits 3-0 should be set to zero.

The SJA1105 mirrors the control part of the SPI framing back onto the MISO (SDIO) pin.

4.1 SPI Write

Data to be written is received on the SDI input of the switch and is treated as a sequence of quadlets with each quadlet starting with SPI cycle 1.

In order to download the hex configuration the hex file has to be parsed and the SPI write action has to be performed for each record in the file.

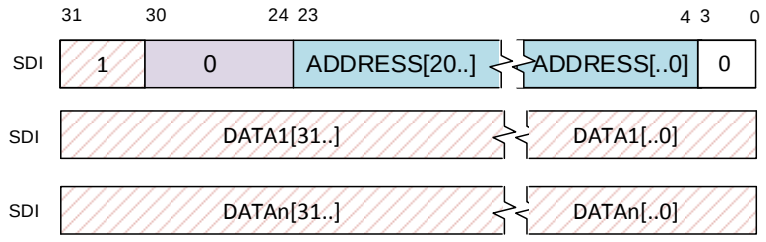

Fig 9. SPI Write Example

4.2 SPI Read

Data (containing the number of 32-bit quadlets to be read and address to read from) is received on the SDI input of the switch and is treated as a sequence of 32-bit quadlets. Data corresponding to the address received is output on SDO pin of the switch.

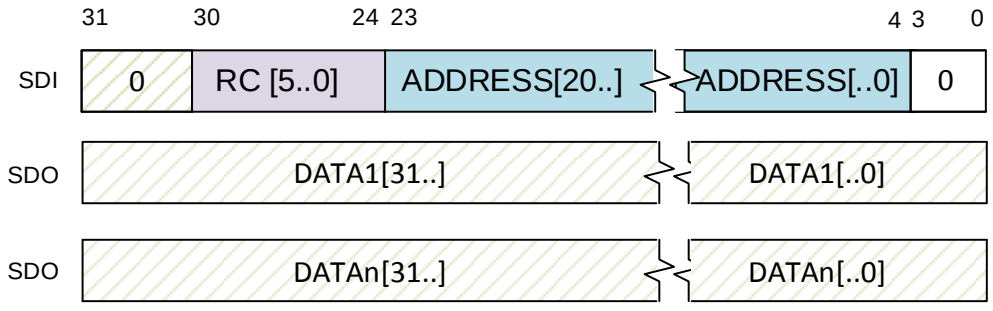

Fig 10. SPI Example: Reading

4.3 Loading a Configuration to the Device

To download the static configuration, a raw configuration has to be compiled first, which conforms to the generic loader format as explained above. This configuration can then be written into the switch by one (or multiple) SPI write transactions. The maximum SPI burst possible in the SJA1105(T)EL consists of 64 32-bit transactions. Therefore, the configuration has to be split into multiple SPI transfers. A typical loading process is shown in Fig 11. Here, the configuration data is split into two SPI transactions. The cut does not need to be aligned with the generic payload format and can happen at any 32-bit boundary. The process starts with a write transaction to address 0x20000. The first double-word must be the device id (which must match with the device id of the silicon revision[4]). After this, the device is unlocked for configuration and the configuration stream is expected. Subsequent write transaction **must** be issued to an address different then 0x20000. For instance the address can be incremented as indicated by the figure. The configuration is completed once the last block with block id zero is received.

Prior to writing the last block (BlockID 0) it is recommended (but not required) to wait until the L2BUSY flag is deasserted. This is to ensure that the L2 forwarding table is fully initialized before the switch is released out of configuration. Otherwise frames are

[4] MRA1 and MRA2 silicon version have different device IDs and a different configuration stream format.

dropped during the remaining time in which the L2 forwarding table is initialized and the flag L2BUSYFDS is found set.

After loading a configuration, the “Initial device configuration flag register” (address 0x1) should be checked to verify that the configuration was accepted. Specifically, it indicates whether a false device ID was used or if a local or global CRC error was found. An incomplete configuration sequence can be restarted by writing the device ID to address 0x20000. After a successful configuration, a new configuration can only be loaded after reset.

Note that the device does not check the configuration for logical consistency. For instance, it is not checked, if the configuration of queues, partition size, or priority remapping makes sense. Only the CRC of the global loader format is checked for correctness.

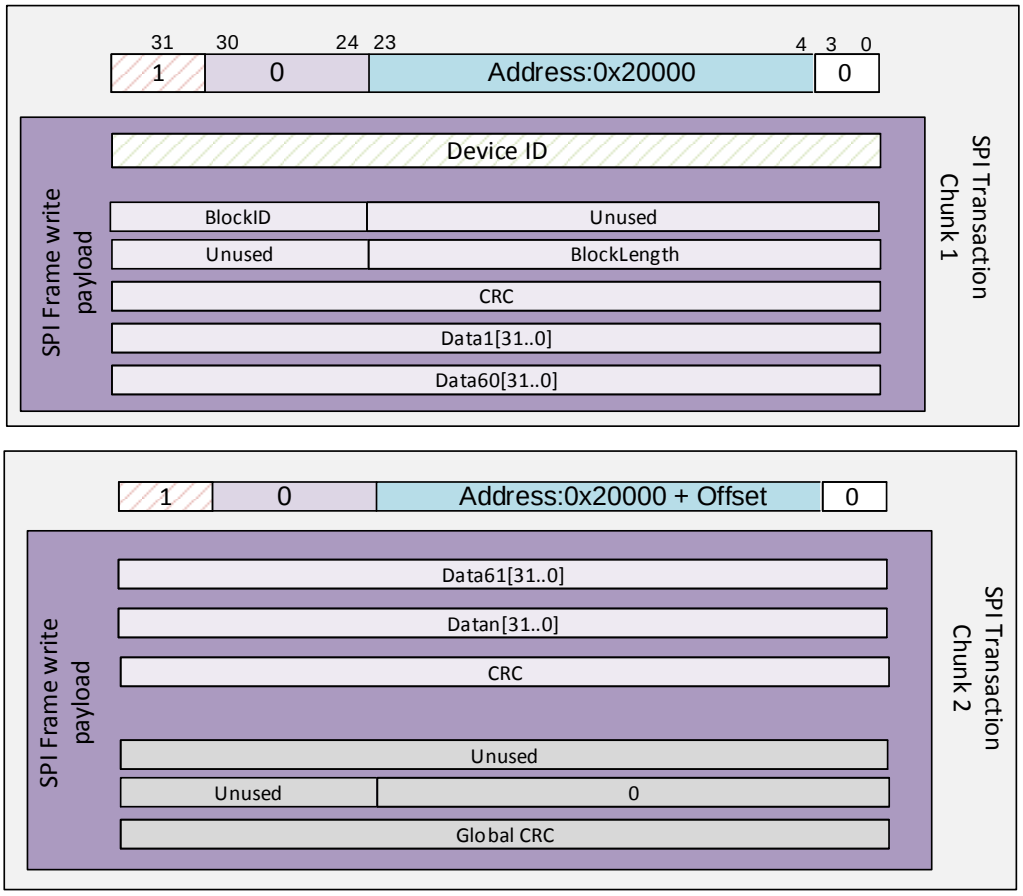

Fig 11. SPI Download Configuration

5. Legal information

5.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

5.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

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

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

5.3 Licenses

Purchase of NXP <xxx> components
<License statement text>

5.4 Patents

Notice is herewith given that the subject device uses one or more of the following patents and that each of these patents may have corresponding patents in other jurisdictions.

<Patent ID> — owned by <Company name>

5.5 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

<Name> — is a trademark of NXP B.V.

6. Index

7. List of figures

8. List of tables

9. Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 28 July 2016
Document identifier: AH1601