AH1704

SJA1105PQRS Application Hints

Rev. 1.3 — 23 July 2020

Application hints

Document information

Info	Content
Keywords	SJA1105PQRS, PCB, Schematic, Layout, Configuration
Abstract	This document gives guidelines for the application of the SJA1105PQRS automotive Ethernet switch.

Revision history

Rev	Date	Description
0.1	24.04.2017	Initial Version
0.2	15.08.2017	Added recommended system solutions Added TAS Added SGMII Added more CGU configurations
1.0	29.09.2017	Minor cleanups
1.1	07.11.2017	Moved PLL configurations from User Manual to AppHint Added examples for VLAN, Policing, Priority Remapping and Mirroring Added statement for RMII-MAC CLK Hysteresis Minor improvements
1.2	08.03.2018	Added hints for migrating from SJA1105(T) RGMII clock delay clarified, fixed RGMII IDIV_x setup Added frame latency measurements clarified frame handling and address learning algorithm Minor improvements
1.3	23.07.2020	expanded how to recover from TDL hangup (PD bit) delay line timing table for all speeds and angles pin behavior during reset added use case “communication hub for uC’s on the same PCB” and how to support sleep mode of directly connected µC explanation for packet drops added throughput measurements and pitfalls related to it. cascading switches HW + SW aspects sending management frames explained in depth remove use case SJA1105PQRS + TJA1102 with RMII and two crystals PTP explanation clarified and example configuration added for 1-step and 2-ste How to use temp sensor CGU configuration More examples

Contact information

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

1. Introduction

1.1 SJA1105PQRS switch family

The SJA1105P/Q/R/S is a 5-port automotive Ethernet switch supporting IEEE Audio Video Bridging (AVB) and Time-Sensitive Networking (TSN) standards. Each of the five ports can be individually configured to operate at 10 Mbit/s, 100 Mbit/s or 1000 Mbit/s. This arrangement provides the flexibility to connect a mix of PHY devices such as the TJA1101 and TJA1102 PHYs from NXP Semiconductors and other commercially available Fast Ethernet and Gigabit Ethernet PHYs. The high-speed interface makes it easy to cascade multiple SJA1105P/Q/R/S and SJA1105(T) for scalability. It can be used in various automotive scenarios such as gateway applications, body domain controllers or for interconnecting multiple ECUs in a daisy chain.

The device comes in two pinout variants. In the variant SJA1105P/Q all five ports can be configured for operation in MII/RMIII/RGMII mode. The second variant (SJA1105R/S) provides a SGMII PHY interface in place of one MII/RMII/RGMII.

The device also comes in two functional variants. The SJA1105Q/S implements Time-Triggered Ethernet and TSN on top of the SJA1105P/R functionality.

Table 1. Switch variants

	5-Port Automotive GBit Switch with AVB	additionally: Time-Triggered Ethernet and TSN functions
Port4 MII/RMII/RGMII	SJA1105P	SJA1105Q
Port4 SGMII	SJA1105R	SJA1105S

This document covers all four variants.

1.2 Other documents

The documentation set for the SJA1105PQRS family consists of three separate documents, each serving different purposes:

- Data Sheet

  The data sheet focuses on typical hardware topics like pinning, electrical behavior, signal timing, wave forms, current consumption, etc.

- User Manual (UM11040)

  The user manual lists the format and the semantics of the registers and the general concepts implemented in the device.

- Application Hints (AH1704)

  Since the UM does not cover use cases and typical applications, or directions how to avoid pitfalls, this info is contained in this application hint.

© NXP Semiconductors N.V. 2020. All rights reserved.

2. Schematic and Layout

2.1 Power Supply

This chapter describes the power design for the SJA1105PQRS. The required voltage domains depend on the used xMII interface types and the logic levels of the host interface used. The device knows the following voltage interfaces:

- Core voltage (VDD_CORE):
  1.2 V core voltage for digital core supply
- Individual xMII interface supply voltage for 5 ports (VDDIO_xMII):
  1.8 V, 2.5 V, 3.3 V supported (depends on xMII type, consult datasheet)
- Host interface supply used for JTAG, SPI and PTP_CLK (VDDIO_HOST)
  1.8 V, 2.5 V, 3.3 V supported. Choice depends on the voltage level of the used microcontroller.
- Clock out supply (VDDIO_CLO)
  1.8 V, 2.5 V, 3.3 V supported. Choice depends on peer device.
- Auxiliary supply for oscillator and PLL (VDDA_OSC, VDDA_PLL)
  1.2 V
- SGMII supply
  1.2V core supply (SGMII_VDD_PHY)
  2.5V analog supply for SGMII PHY (SGMII_VDDA_PHY)

2.1.1 Voltages

Table 2. Voltage requirement overview

Symbol	Description	Min [V]	Typ [V]	Max [V]
VDD_CORE	1.2 V core voltage	1.14	1.20	1.30
VDDA_OSC	1.2 V oscillator supply	1.10	1.20	1.30
VDDA_PLL	1.2 V PLL supply	1.10	1.20	1.30
VDDIO_MIIx	MII mode	3.00 2.30 1.65	3.30 2.50 1.80	3.60 2.70 1.95
	RMII mode	3.00 2.30	3.30 2.50	3.60 2.70
	RGMII mode	3.00 2.30 1.65	3.30 2.50 1.80	3.60 2.70 1.95
VDDA_SGMII	Analog SGMII supply voltage	2.3	2.5	2.7
VDDD_SGMII	Digital SGMII supply voltage	1.14	1.2	1.3
VDDIO_HOST	Host interface supply voltage.			

© NXP Semiconductors N.V. 2020. All rights reserved.

Symbol	Description	Min [V]	Typ [V]	Max [V]
	1.8 V operation	1.65	1.8	1.95
	2.5 V operation	2.3	2.5	2.7
	3.3 V operation	3.0	3.3	3.6
VDDIO_CLO	Voltage supply depends on CLK_OUT pin connection to other chip (i.e. TJA1102 or SJA1105PQRS)			
	1.8 V operation	1.65	1.8	1.95
	2.5 V operation	2.3	2.5	2.7
	3.3 V operation	3.0	3.3	3.6

2.2 Filtering

This section describes the power input topologies by input voltages, usages and power filter circuit design.

As a 330Ohm@100MHz ferrite, the BLM18P331 type is recommended.

2.2.1 Core and Auxiliary Supply

The power input topology for VDD_CORE, VDDA_OSC and VDDA_PLL is illustrated in Fig 1. A ferrite bead is recommended for each power rail.

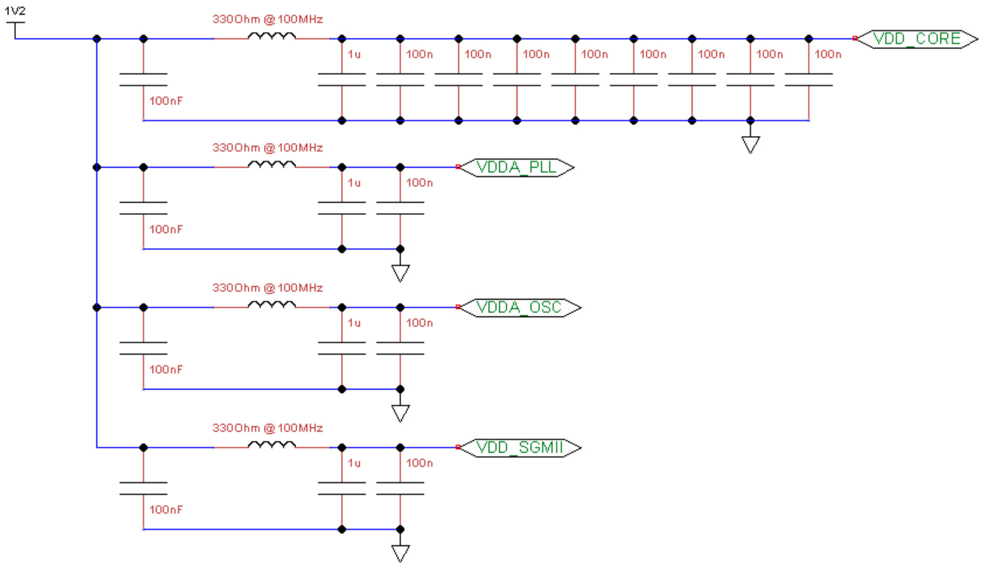


Fig 1. **1.2 V Power Rail Topology with Filtering**

2.2.2 MII/RMII/RGMII/SGMII Supply

The power input topology and filter for VDDIO_MIIx is illustrated in Fig 2. Each xMII-port should be powered with 1.8V, 2.5V or 3.3V according to its mode configuration.

A PI type filter with one ferrite bead and two capacitors shall be used for each VDDIO_MIIx power filtering.

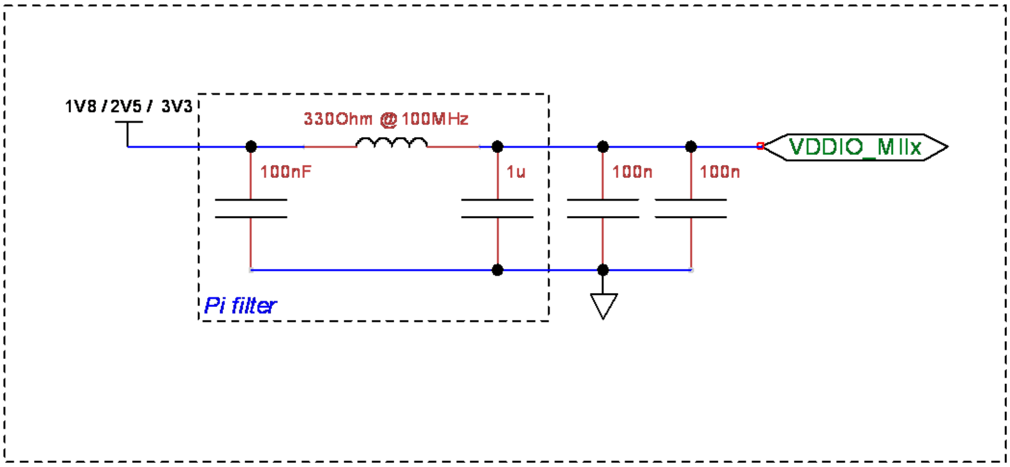


Fig 2. **Low-pass power filter for VDDIO_MIIx**

The SGMII port on SJA1105R/S has two supply domains: 1.2V and 2.5V for analog IO. The same PI type filter shall be used for both supplies (Fig 2 and Fig 3).

Note that the analog voltages on the TXP and TXN pins are not directly driven by the 2.5V analog IO supply. The differential voltage can be controlled through register settings. See Section 6.2 for details.

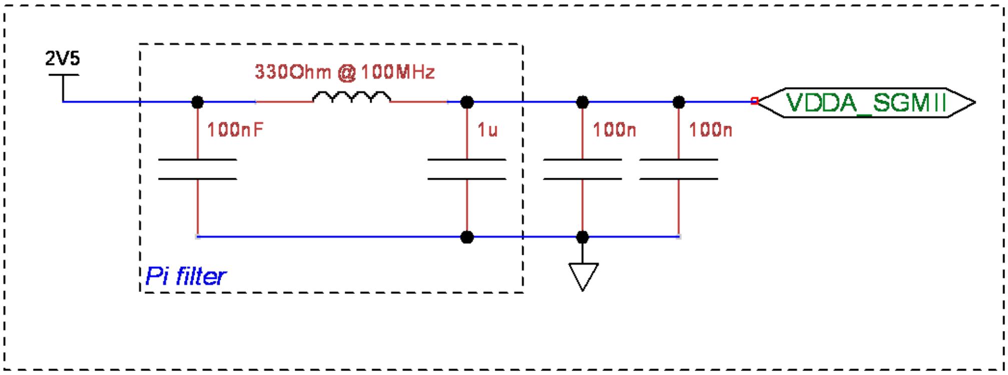


Fig 3. **Low-pass power filter for VDDA_SGMII**

2.2.3 Unused Ports

If a port remains unused, the VDDIO_MIIx supply can be connected directly to ground in order to save power. Optionally the power supply for individual ports can by dynamically turned off while the core supply remains active. However, VDDIO_MIIx must not be floating.

For unused data pins, consult Section 2.5.1.3. Please note that unused ports need to be disabled in software.

2.2.4 HOST and CLK_OUT Supply

The power input topology and filter for VDDIO_HOST and VDDIO_CLO are illustrated in Fig 4. Both VDDIO_HOST and VDDIO_CLO can accept a 1.8 V, 2.5 V or 3.3 V supply. The SPI, JTAG and PTP_CLK interfaces are supplied via VDDIO_HOST. The 25 MHz clock output, CLK_OUT, is supplied from VDDIO_CLO. Independent supply pins and various voltage options provide flexibility for different interface voltage compatibility.

It is important that the domain which supplies VDDIO_HOST, also supplies the attached microprocessor’s SPI interface. This guarantees that the voltage on SDO and SDI pins is derived from VDD_HOST. In any case, the SPI input pins (SDI, CS) must be never asserted before VDDIO_HOST is supplied to the switch.

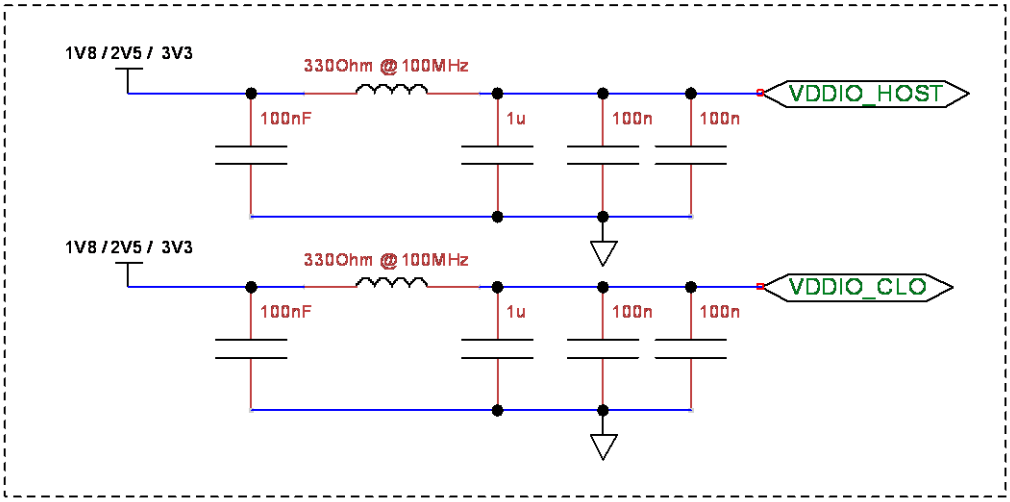


Fig 4. **Low-pass power filter for VDDIO_HOST and VDDIO_CLO**

2.2.5 Layout Recommendations

During decoupling capacitor placement design, attention must be paid to the parasite inductance and parasite resistance which can affect the resonant frequency of the decoupling capacitor and consequently impact the decoupling performance. To minimize the parasite inductance and resistance:

- Reduce the trace length between the power and ground pin to the decoupling capacitor.
- Increase the trace width.

- Place small capacitors as close as possible to the chip pins because small capacitors with higher resonant frequency are more likely to be affected by the parasite inductance and resistance.

2.2.6 Power consumption

The power consumption depends on the mode of operation and the traffic which is processed by the device. Its maximum power consumption will not exceed 800 mW.

2.3 Clocking

This section describes the clock requirements and design suggestion for the SJA1105.

Table 3. Clock Pins and purpose

Symbol	Pin	Description
OSC_IN OSC_OUT	K2 L1	25 MHz crystal/oscillator input Connect a crystal between OSC_IN and OSC_OUT pins or a clock source to OSC_IN pin. When a crystal is used, use load caps matching the crystal specification. When an external oscillator is used, leave OSC_OUT open. The maximum OSC_IN input voltage is 1.2 V.
CLK_OUT	H1	25 MHz clock output. The clock can be used as the reference clock for PHYs. Its voltage level is defined by VDDIO_CLO. By default, this pin is active - to improve EMC the CLK_OUT pin can be disabled by software.
PTP_CLK	N4	PTP clock output or CASC_SYNC in/out. Can be used as a timer input to a microcontroller or to synchronize the internal free running timestamp counter. By default this pin is disabled.

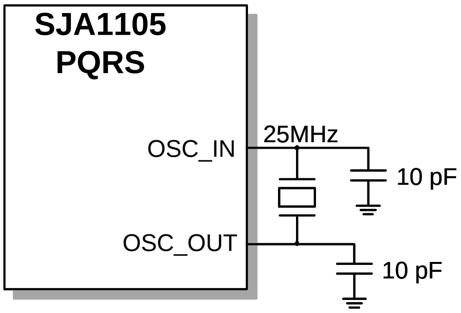
2.3.1 Reference Clock Design Requirements

SJA1105PQRS supports both crystal input (oscillator mode) and external clock input (clock mode) as reference.

2.3.1.1 Crystal Mode

The basic principle for crystal and load capacitor selection is to ensure a stable oscillation frequency within the tolerance range of 25 MHz ± 50 ppm (for RMII and RGMII) or 25 MHz ± 100 ppm if all ports operate in MII mode. Two 10 pF ceramic capacitors are recommended for use with the default configuration. The capacitance value may vary with crystal specification and board-level test results.

When crystal is used for the clock source, place the crystal as close as possible to the SJA1105PQRS for minimizing EMI impact.



Note: 10pF indicative only. Actual value depends on crystal

Fig 5. **Oscillator in Crystal / Master mode**

Table 4. Recommended Crystal Parameters

Symbol	Parameter	Min	Typ	Max	Unit
Fc	Crystal fundamental frequency	-	25	-	MHz
Ft	Frequency tolerance (MII-only)	-100		+100	ppm
	Frequency tolerance (RMII, RGMII, SGMII)	-50		+50	ppm
Cl	Load Capacitance		10		pF

2.3.1.2 Crystal Layout

The following rules should be followed for optimal performance.

- Place the crystal as close as possible to the OSC_IN and OSC_OUT pins to minimize the trace length. Short traces help to reduce the parasitic inductance, stray capacitance and noise coupling.
- Avoid routing any other signal trace directly underneath the crystal or the OSC_IN and OSC_OUT traces, especially running parallel with the clock traces. Keep all other signal traces at least 200 mil away from the OSC_IN and OSC_OUT traces.

- Do not use the crystal output to drive any other devices. The crystal oscillator is designed to drive the crystal in a direct feedback loop. Any extra branching or loading may prevent the oscillator from working properly.
- It is recommended to have a ground ring on the signal surrounding the area formed by the OSC_IN, OSC_OUT pins, and the footprint of the crystal.
- Avoid ground islands and stitch ground traces to the ground plane below.

2.3.2 Clock Mode

The SJA1105PQRS supports AC-coupled (without external voltage bias) external 25 MHz clock input through the OSC_IN pin. When an external clock is injected through the OSC_IN pin, leave the OSC_OUT pin open. A 100 pF capacitor is required for such applications. Note that the input voltage of OSC_IN must not exceed 1.2V peak-to-peak (VDDA_OSC). If a digital clock source is used which exceeds 1.2V peak-to-peak, a resistor voltage divider is recommended.

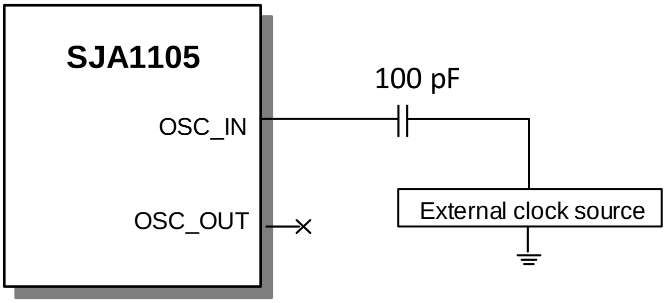


Fig 6. **Oscillator in External / Slave mode**

The recommended external single-ended 25 MHz clock source parameters are listed in the following Table.

Table 5. Recommended external clock parameters

Symbol	Parameter	Min	Typ	Max	Unit
Fext	External clock frequency	–	25	–	MHz
Ft	Frequency tolerance				
	only MII	-100	–	+100	PPM
	RMII / RGMII / SGMII[1]	-50	–	+50	PPM
Vi(OSC_IN)	Input voltage (RMS) on pin OSC_IN	0.2		1.2	V

[1] SJA1105R/S only

2.3.3 Cascaded clocking scheme

In a cascaded switch setup, multiple SJA1105PQRS devices are connected in a daisy chain. It is recommended, but not strictly required, that the processor is connected to the first or the last switch of the chain.

If multiple SJA1105PQRS are cascaded, a single crystal oscillator can be used. The clock signal is daisy chained through the switches over the CLK_OUT pin into the OSC_IN of the following switch. However, care must be taken to not exceed the OSC_IN voltage specification of 1.2V. Assuming VDDIO_CLO supply is already on the minimal possible voltage level of 1.8V, the RMS of the CLK_OUT signal might exceed 1.2V, depending on the waveform. Therefore, a voltage divider is required to bring the CLK_OUT voltage down (Fig 7).

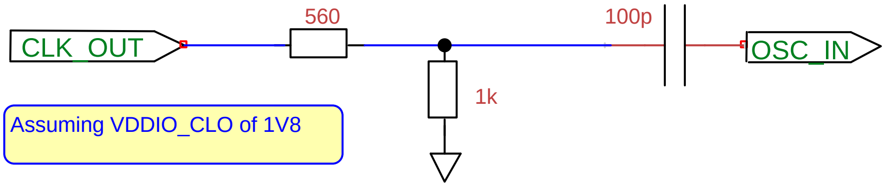


Fig 7. **Clocking in cascaded switches**

2.3.4 Clock output (CLK_OUT)

The 25 MHz clock output, CLK_OUT, is supplied from VDDIO_CLO. This clock can be used as reference clocks for external device. For example, TJA1102 100BASE-T1 Ethernet transceiver can be supplied with this clock signal. Its voltage level is defined by VDDIO_CLO power supply, could be 1.8V, 2.5V or 3.3V, depending on the voltage requirement of the receiving device (TJA1102 requires 3.3V). If this 25MHz clock output is used to drive two or more external chips, a clock buffer is suggested. In any case stubs must be avoided. The CLK_OUT trace should be impedance controlled at 50 Ohms.

By default, this pin is enabled and needs to be disabled in software (CFG_PAD_MISC register) if not required. The CLK_OUT pin is directly driven from the internal oscillator and will immediate start once the oscillator has stabilized. The CLK_OUT operation will not be affected by a device reset (RST_N or software reset).

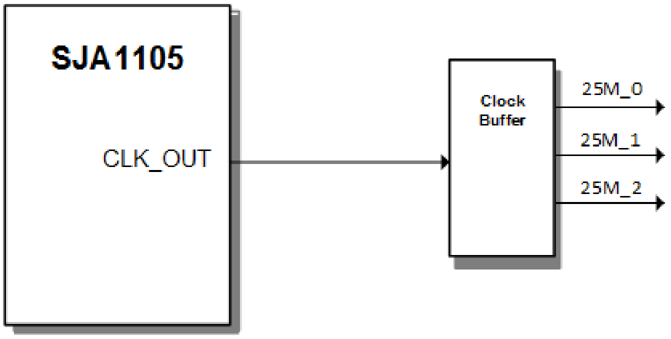


Fig 8. **Multi-point clock output topology**

Table 6. Recommended external clock parameters

Symbol	Parameter	Min	Typ	Max	Unit
Fout	Frequency		25		MHz
	1. Output frequency stability depends on the crystal circuit oscillation frequency or input clock frequency stability. 2. If CLK_OUT is not used, it can be left unconnected. 3. If it is used as a chip reference clock, pay attention to the input requirement, especially the jitter.				

Layout recommendations:

- Route the clock signals where the traces do not intersect each other.
- Ensure that a minimum number of vias are used on clock trace. Vias change the trace impedance, and thus cause reflections. This can result in EMI or jitter issues. Therefore, route clock traces on one layer as far as possible. If clock signals cannot be routed on one layer, ensure that they stay one layer from either the power or ground plane to minimize trace impedance changes.
- Ensue that a solid ground plane is on the layer adjacent to the clock trace routing layer and no discontinuities are created by vias.
- Do not use 90 degree angles when routing clock traces. Use smoothly curved traces.

Note: if CLK_OUT is unconnected, VDDIO_CLO may be connected to GND to save power.

2.4 Reset and Power-on Sequence Design

2.4.1 Reset

Table 7. Reset pin properties

Symbol	Pin	Description
RST_N	P3	Chip reset input (active-LOW)

RST_N is the logical reset input to the SJA1105PQRS. A typical reset input is illustrated in Fig 9. A 10 k Ohm resistor pull-up to VDDIO_HOST is needed. RST_N voltage thresholds are according to VDDIO_HOST, which is either 1.8V, 2.5V or 3.3V, depending on the application.

Note: For the SJA1105PQRS it is **not** required to assert the TRST_N pin together with the RST_N pin.

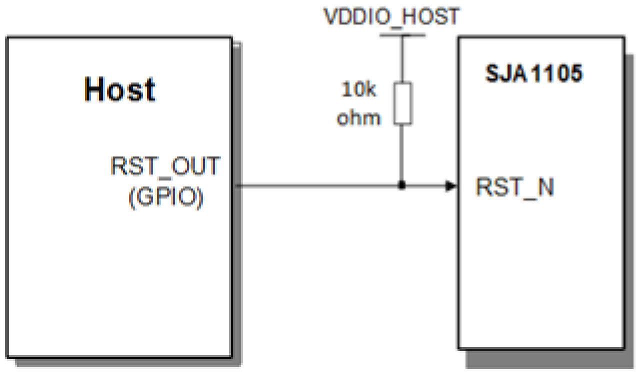


Fig 9. **Reset input**

2.4.2 Pin behavior during Reset

All MIIx pins (that is: TX_CLK/REF_CLK/TXC, TX_EN/TX_CTL, TX_ER, TXD[3:0], RX_CLK/RXC, RX_ER, RX_DV/CRS_DV/RX_CTL, RXD[3:0]) go into a High-Z state when Reset is asserted.

However, other pins are pulled to their respective VCCIO or to GND by internal weak resistors, even during reset. This fact needs to be considered if the signal connected to such a pin is used for pin strapping of a component connected to the switch.

Table 8. Pin behavior during reset

Pin name	Pin behavior in reset
TRST_N	Pull-up
TDI	Pull-up
TCK	Pull-up
TMS	Pull-up
TDO	High-Z
SDI	Pull-up
SDO	High-Z
SCK	Pull-down
SS_N	Pull-up
PTP_CLK	High-Z
CLK_OUT	(always active, outputs clock signal)

2.4.3 Pin behavior when supply is missing

In a functional safety context, the behavior of a ethernet port pin needs to be known when one or more supply voltages are failing.

2.4.3.1 RGMII pins

For the RGMII I/O pins four different supply states can be considered. A voltage being "Low" means power on this rail is missing. The resulting state of the pins is defined in the following table:

Table 9. RGMII Pin behavior under power-fail

VDD_CORE	VDDIO_MIIx	xMII pin state
Low	Low	Clamped by a diode caused by the absence of I/O supply voltage VDDIO_MIIx
High	Low	Clamped by a diode caused by the absence of I/O supply voltage VDDIO_MIIx
Low	High	Tristate
High	High	Normal operation (tristate until configured by the host)

2.4.3.2 SGMII pins

The SGMII block has two supplies, next to the core supply, and therefore eight different 'static' supply states can be considered. Note that the SGMII core supply VDD_SGMII must be derived from the core supply VDD_CORE.

Table 10. SGMII Pin behavior under power-fail

VDD_CORE	VDD_SGMII	VDDA_SGMII	xMII pin state
Low	Low	Low	Clamped by a diode caused by the absence of analog supply voltage VDDA_SGMII
High	Low	Low	Clamped by a diode caused by the absence of analog supply voltage VDDA_SGMII
Low	High	Low	Clamped by a diode caused by the absence of analog supply voltage VDDA_SGMII
High	High	Low	Clamped by a diode caused by the absence of analog supply voltage VDDA_SGMII
Low	Low	High	Undefined
High	Low	High	Undefined
Low	High	High	Undefined
High	High	High	Normal Operation

Unfortunately it is not defined what the state of the pins SGMII_RXP, SGMII_RXN, SGMII_TXP, SGMII_TXN is when the SGMII analog supply VDDA_SGMII is high but one of the other two supplies is not.

2.4.4 Reset and Power-On Sequence

The general guidelines for power-up sequences are illustrated in Fig 10. The power-up sequences follow the rules:

- Definition of "power-up": the supply voltage reaches 90 % of its nominal voltage.
- Definition of "rise time": the duration for the supply voltage to rise from 10% to 90%.

No special sequencing of the involved 1.2V/2.5V/3.3V/1.8V rails is required.

- The RST_N signal must be kept low for at least 5 us after all power supplies and reference clock signals become stable.

Note: during RST_N, the CLK_OUT pin will continue to output a stable 25 MHz clock.

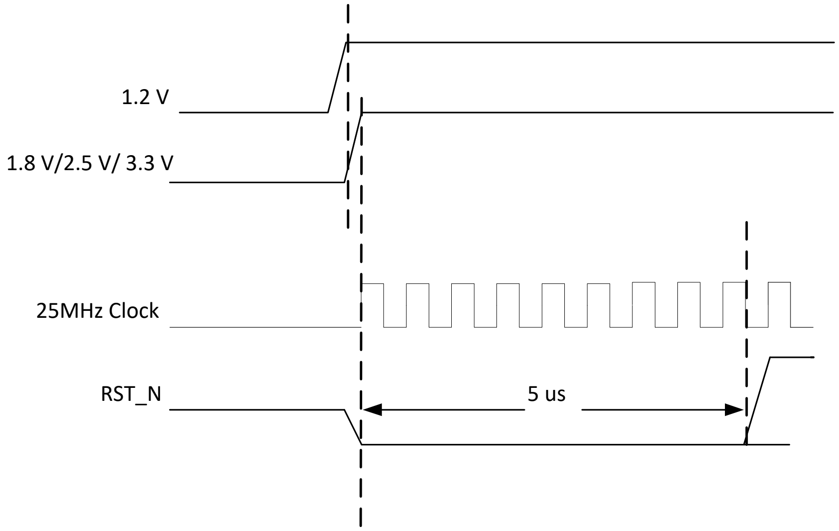


Fig 10. **Reset timing**

2.5 Interface Design

2.5.1 Ethernet Interface

The SJA1105PQRS xMII interfaces can be configured to support a wide variety of PHYs and host controllers. Each port can be configured for connecting a PHY (MAC-Mode for a MAC-to-PHY communication) or another MAC (PHY-Mode for MAC-to-MAC communication).

2.5.1.1 MII/RMII/RGMII layout guideline

1. Single-ended 50 Ohm impedance
2. Matched trace length: use same lengths for the signals in a group, e.g. RX data and RX clk, and especially for differential pairs.
3. Buried traces for clock and data signal is recommended to reduce radiated emission.

2.5.1.2 SGMII layout guideline

The traces should be routed as 100 Ohm differential, controlled impedance transmission lines. Pairs must be length matched to reduce EMI and Jitter. It is recommended to route the traces over a single ground plane. Traces should avoid vias and layer changes.

External AC-coupling of 100nF in RX and TX pairs is required. For SGMII it does not really matter where the AC coupling caps are physically located, either close to the transmitter or close to the receiver. However, to gain a small signal integrity benefit caps should be placed within approx. 4cm of the perimeter of the SJA1105RS package (considering there is <1cm routing inside the SJA1105RS package laminate). Note that for EMI reasons it is important that the two caps for TXP and TXN are placed next to each other, both at the same distance from the SJA1105RS.

For applications that use SGMII to interface to an SFP module (see 6.2.3.2), designers should be aware that according to the "Small Form-factor Pluggable (SFP) Transceiver MultiSource Agreement (MSA)"[2], section B2 point 7 and 9, the AC coupling capacitors must be implemented inside the SFP module. If they are implemented in the SFP module, they are NOT required on the SJA1105RS host board.

It is required that the 1.2V VDD_SGMII is supplied from the **same voltage source** as the VDD_CORE.

The option to swap the SGMII interface polarity (TX_POL_INV and RX_POL_INV in DIGITAL_CONTROL_2) helps avoid crossings when routing a MAC-to-MAC configuration, e.g. in a cascaded scenario:

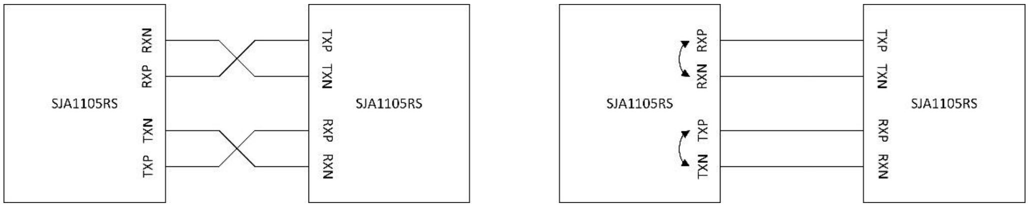


Fig 11. **Swapping SGMII interface polarity**

2.5.1.3 Unused xMII Interface

Hardware Recommendations

An unused xMII interface shall have defined, non-floating inputs. In particular, all RX inputs shall be connected to GND. Alternatively internal pullups/pulldowns (see ACU registers) can be used to define a level on the pins. These internal resistors might need to be explicitly enabled by software.

Outputs (TX pins) can be left open.

In order to reduce power consumption, unused ports can be powered-off by connecting VDDIO_xMII to GND permanently. For temporarily disabling and re-enabling a port, this is not recommended. For the details see ch. 6.3.

If the SGMII port is permanently not used, connect all SGMII power pins (VDD_SGMII, VDDA_SGMII) to GND. SGMII_RREF is an output, which can be left open. SGMII's RX pins may be left open, because of the internal termination.

Software Recommendations

Note that unused ports need to be disabled in software. Not doing so wastes switch frame memory resources and power.

Permanently unused ports need to be configured disabled for ingress and egress (INGRESS, EGRESS in MAC Configuration table) and "*disabled"* must be selected in the xMII Mode Configuration Table.

[2] http://www.schelto.com/SFP/SFP%20MSA.pdf

2.5.1.4 RMII and MII-lite

In RMII and MII-lite connections, some input and output pins remain unconnected. In RMII the pins TXD[3:2] and RXD[3:2] are switched to high-impedance. SJA1105PQRS internal pull-up/-down resistor settings can be used to force a state on these pins (see ACU registers). This is not needed for operation of the SJA1105PQRS but might be required for connected devices.

In MII setting all interface pins must be connected. If unused, RX_ER must be connected to GND or the internal pullup/down must be configured.

Note: RX_ER must never be open

2.5.2 SPI Interface

The SJA1105PQRS must be connected via SPI to a host controller for initial configuration and adjustments of clock, and for general management. It is a full-duplex synchronous channel supporting a four-wire interface (SCK, SDI, SDO and SS_N). The SJA1105PQRS is always in slave mode. The host can control/configure the SJA1105PQRS via register (address) based access or via a configuration stream. In order to ensure support for a wide range of microcontrollers, the SPI interface can operate at a supply voltage of 3.3 V, 2.5 V or 1.8 V, depending on VDDIO_HOST power supply. The SPI pins are listed in Table 11.

Table 11. SPI pins

Symbol	Pin	Description
SCK	P5	SPI serial clock, up to 25MHz
SDI	N5	SPI data in. This input signal is used to transfer data serially into the SJA1105PQRS
SDO	P4	SPI data out. This output signal is used to transfer data serially out of the SJA1105PQRS
SS_N	N6	SPI chip select, active low

The maximum SPI clock frequency acceptable by SJA1105PQRS is 25MHz. SPI interface should be routed with care. Fig 12 shows the typical SPI connection between Host and SJA1105PQRS.

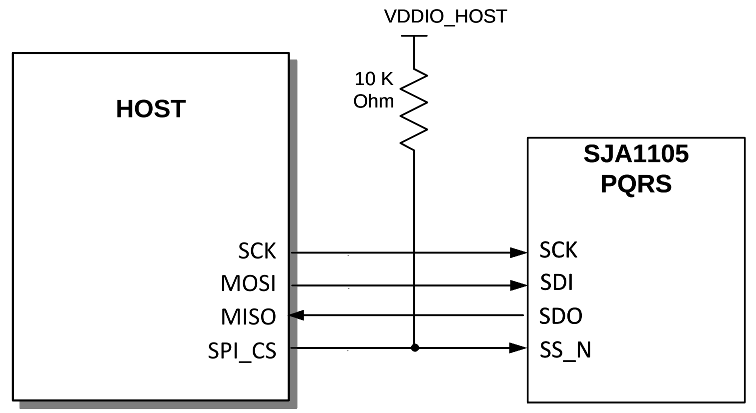


Fig 12. **SPI Interface**[3]

SPI design guidelines:

1. Single-ended 50 Ohm impedance controlled trace.
2. Overall length match between the group (SCK, SDI, SDO and SS_N) is recommended

2.5.3 JTAG Interface

The device supports IEEE1149.1 compliant JTAG interface for TAP controller access and boundary scan.

Use of JTAG is optional. The device does not need any programming or JTAG configuration.

3. Recommended System Solutions

This section provides typical system solutions for TJA1101 (single channel PHY for100BASE-T1), TJA1102(S) (dual channel PHY for 100BASE-T1) and SJA1105PQRS for optimized interoperability, as well as a typical connection to a management host's MAC for a PHY-less interface.

3.1 Interoperability with TJA1101 and TJA1102(S)

TJA1101 and TJA1102(S) are identical regarding their xMII interface. Without loss of generality, this chapter is meant for all devices, despite only TJA1102 is mentioned. Exceptions are explicitly stated.

3.1.1 MII Mode with XTAL

In this mode, each device has its dedicated crystal. In this mode the TJA1102 will not output a clock on CLK_IN_OUT. The switch must be operated with VDDIO_MIIx of 3.3V.

[3] External 10k Ohms resistor is optional, SJA1105PQRS has an internal pull-up.

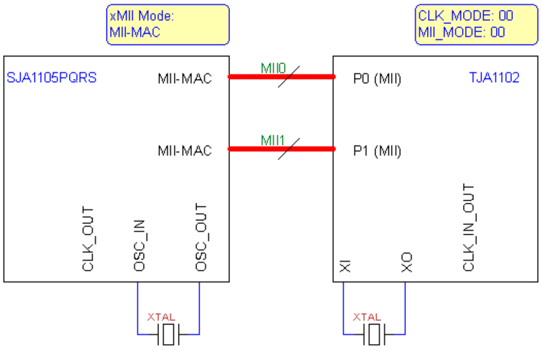


Fig 13. **MII with crystals at SJA1105PQRS and TJA1102**

3.1.2 MII Mode with digital clock (CLK_OUT)

In this mode, the SJA1105PQRS has its dedicated crystal, and emits 25 MHz on CLK_OUT. The TJA1102 will operate the CLK_IN_OUT pin as input. In case multiple TJA1102s are used, a clock buffer is recommended. Xi and Xo of the TJA1102 can be left open.

The switch must be operated with VDDIO_CLO of 3.3V and VDDIO_MIIx of 3.3V of the interfaces connected to the TJA1102.

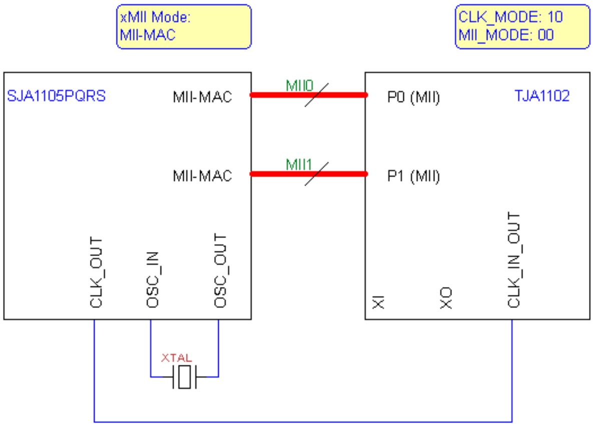


Fig 14. **MII with crystal at SJA1105PQRS and digital clock supply to TJA1102**[4]

[4] Please note, that if the CGU is incorrectly configured for MII-PHY mode to use the IDIV generated internal clock and NOT the PHY's clock signals (refer to Fig 19), this will work nevertheless most of the time, as both PHY and MAC use the same clock signal. However, due to the clock skew influenced by factors like temperature, setup and hold time, specifications are violated, which results in sporadic faults, which are hard to isolate.

Please note, that the TJA1101 external clock mode cannot be configured by pinstrapping. Instead, it must be configured via SMI before sending traffic over the PHY.

3.1.3 RMII Mode with digital clock (REF_CLK)

In this mode, the TJA1102 will be clocked over the REF_CLK pins of the RMII interfaces. Each port of the TJA1102 will be clocked with its own digital clock from the REF_CLK pin of the associated interface. The pins Xi and Xo of the TJA1102 can be left floating.

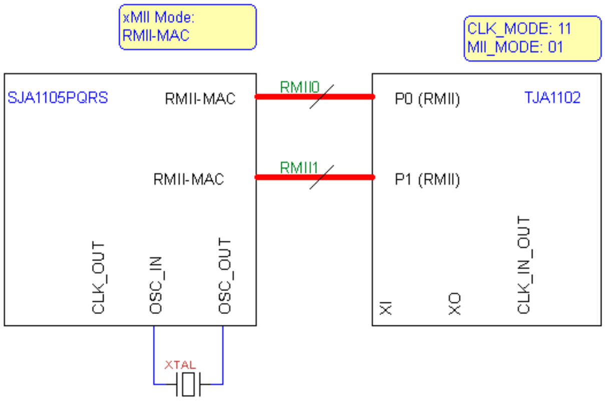


Fig 15. **RMII with crystal at SJA1105PQRS and digital clock supply to TJA1102**

Please note, that the TJA1101 external clock mode cannot be configured by pinstrapping. Instead, it must be configured via SMI before sending traffic over the PHY.

3.2 PHY-less connection to a microcontroller's MAC

In applications, where multiple microcontrollers on a PCB have to be connected, the SJA1105PQRS can be used as a high bandwidth communication hub. A rich set of configuration options *for each port* allows to connect a cluster of up to 5 microcontrollers, or 4 microcontrollers, if the cluster also wants to communicate with the rest of the world (Fig 16). Cascading the switch allows even more local connections.

Depending on the MAC configuration options of the onboard µCs, communication speed between µC and switch can be up to 1Gbps (using RGMII or SGMII) or 100Mbps (RMII or MII).

From a microcontroller's point of view, the switch must look like a PHY connected to the xMII interface of the microcontroller's MAC. Due to the switch's capability to set *each port* into PHY mode, it can pretend to be a PHY without any external logic, without any restrictions regarding link speed. This makes the SJA1105PQRS an ideal solution for this type of application.

For SGMII or RGMII there is no special setting necessary -just connect the RX with TX. For MII and RMII the switch's port must be set to PHY mode. Refer to ch. 6.1.5 for MII-PHY mode and ch. 6.1.8 - 6.1.9 for RMII-PHY mode.

© NXP Semiconductors N.V. 2020. All rights reserved.

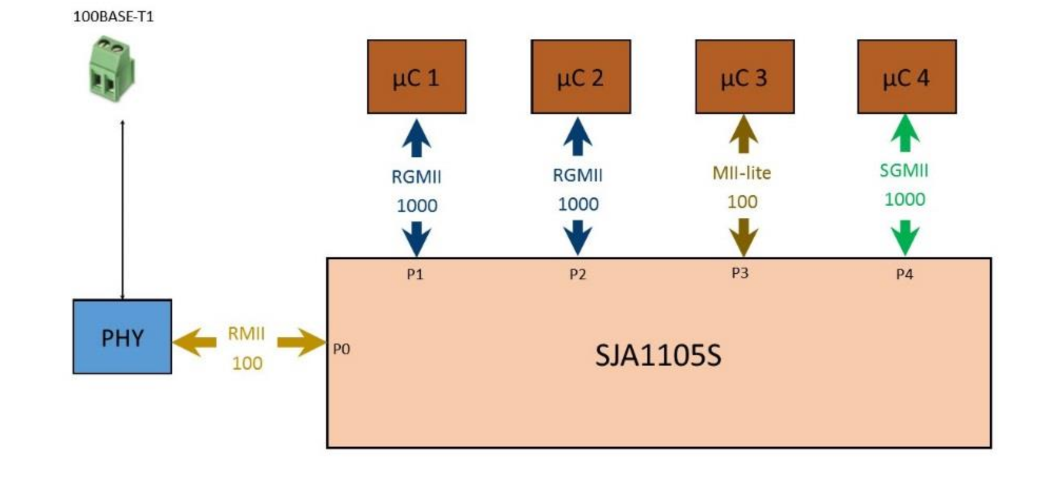


Fig 16. **Multiple microcontroller hub example**

For hints how to deal with architectures where a connected µC may go into low power sleep mode while the switch keeps working or the switch may sleep while the µC stays on, see ch. 6.3.

3.3 RMII Interoperability Remarks

If a port of the SJA1105PQRS is operating in RMII-PHY mode, it expects a true MAC peer device as per RMII 1.2 specification (The switch fully emulates /J/, /K/ carrier sense encoding).

This means: A SJA1105PQRS in RMII-PHY mode does NOT communicate with a TJA1101/TJA1102 PHY, despite it is configured with MII-MODE=01 (RMII-mode). This is because the TJA1101/TJA1102 is also operating in RMII-PHY mode.

For connecting a PHY in RMII mode, the SJA1105PQRS must be configured for MAC mode operation.

4. Infrastructure Modules

4.1 RCU

The RCU implements the time sequencing of internal reset signals. The reset control register gives the opportunity to start the reset sequence at a certain stage. All reset actions belonging to this and stages afterwards will be executed, while stages before are skipped. The bit order in the reset control register exposes the staging sequence with lower bits being the early stages, and higher bits the later stages.

Table 12. Reset control register (addr 100440h)

Bit	Symbol	Description
31:9	reserved	
8	SWITCH_RST	Main reset for all functional modules

Bit	Symbol	Description
7	CFG_RST	Resets all chip-configuration
6	reserved	
5	CAR_RST	Resets the clock and reset control logic
4	OTP_RST	Initiates an OTP read-cycle to read out the product configuration settings
3	WARM_RST	Perform a warm reset
2	COLD_RST	Perform a cold reset
1	POR_RST	Perform a power-on reset
0	reserved	

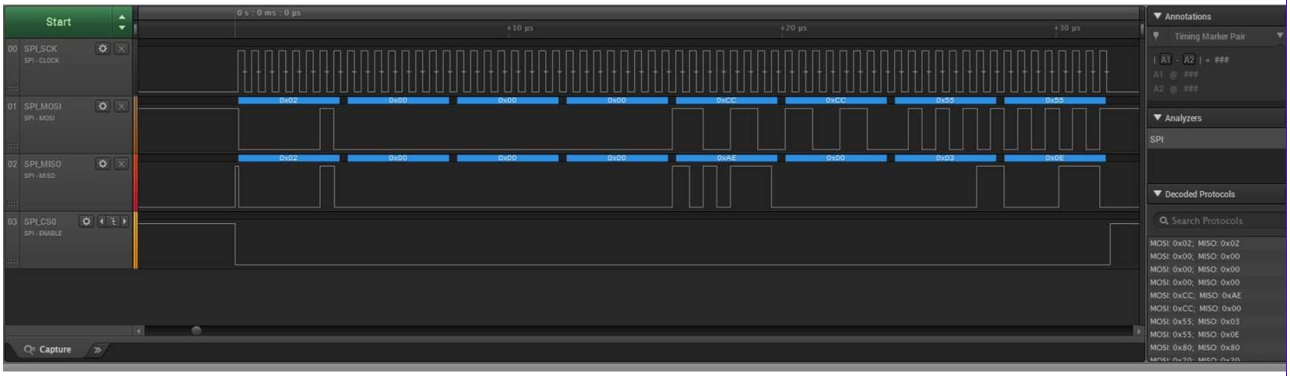
Practical use might have the following reset stages:

POR_RST:	Trigger power-on reset – same effect as power-off and power-on
CFG_RST:	purge the downloaded configuration and expect a new one.
SWITCH_RST:	reset only switch logic, CGU and ACU registers are not reset

5. SPI communication protocol

5.1 SPI waveform

Complementary to the information in the UM to the SPI data format, the following scope plots show how a successful SPI transaction looks like.



```
MOSI: 0x02;  MISO: 0x02
MOSI: 0x00;  MISO: 0x00
MOSI: 0x00;  MISO: 0x00
MOSI: 0x00;  MISO: 0x00
MOSI: 0xCC;  MISO: 0xAE
MOSI: 0xCC;  MISO: 0x00
MOSI: 0x55;  MISO: 0x03
MOSI: 0x55;  MISO: 0x0E
MOSI: 0x80;  MISO: 0x80
MOSI: 0x20;  MISO: 0x20
```

Fig 17. **SPI register read example**

Fig 17 shows the waveforms of a single-word read operation for register 00h (DEVICEID register). The return value in this case is AE00030Eh. The 2nd data word sent by the host (CCCC5555h) has been randomly chosen and is not used in the SJA1105PQRS. However, in case the SJA1105PQRS is still in reset or the main clock is missing, the SPI interface mirrors the input data to the output. The unusual send pattern can help to detect the switch status.

© NXP Semiconductors N.V. 2020. All rights reserved.

5.2 SPI write operation length

UM11040 up to version 1.2 claimed there is a maximum number of 64*32 bits allowed for a single write operation.

This has been fixed in later versions, as there is virtually no limit in the number of words per write operation.

6. Interface Configuration

This chapter provides an overview and understanding of the resources required by software designers to get started with the SJA1105PQRS device and configure the device to bring it to operation.

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

6.1 MII/RMII/RGMII Configuration

Configuring an MII/RMII/RGMII Ethernet port comprises three aspects.

- Setting the **xMII Mode Parameters Table** (PHY_MAC and xMII_MODE) in the static configuration. This configures the mode of the Interface (MII, RMII, RGMII).
- The port speed in the **MAC Configuration Table** (SPEED) must be set. Note that the SPEED parameter can also be set over the Dynamic Control Interface at run-time. This is useful in port setups which determine the link speed by autonegotiation.
- Programming the **Clock Generation Unit** (CGU). The CGU links the clock sources and sinks so that the clocking scheme matches the speed, MAC/PHY and xMII mode. It must be made sure that all three parameters are consistent, otherwise no or unreliable communication is established and frame drops are observed. For RGMII, the tunable delay line (TDL) can optionally be used for on-chip delay (see 6.1.13).

To summarize, the following entities need to be configured in the sequence shown below:

- MAC Configuration Table (SPEED)
- xMII Mode Parameters Table
- Optional (RGMII only): TDL configuration
- CGU configuration

Setting MAC Configuration Table and xMII Mode Parameters is straight forward and is not explained in details. However, configuration of the CGU can be complex and is therefore discussed in depth.

The CGU (cf. Fig 18) 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 18, 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. The configuration sequence of clock sinks and clock sources is arbitrary. 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 13. CGU register overview

Offset	Name	Access	Reset Value
100005h	RFRQ	R/W	00000000h
100006h	XO66M_0_C	R	0000002Ch
100007h	PLL_0_S	R	00000000h
100008h	PLL_0_C	R	0A040040h
100009h	PLL_1_S	R	00000000h
10000Ah	PLL_1_C	R/W	0A000083h
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	11000000h
100016h	RGMII_TX_CLK_0	R/W	11000000h
100017h	EXT_TX_CLK_0	R/W	11000000h
100018h	EXT_RX_CLK_0	R/W	11000000h
100019h	MII_TX_CLK_1	R/W	12000000h
10001Ah	MII_RX_CLK_1	R/W	12000000h
10001Bh	RMII_REF_CLK_1	R/W	12000000h
10001Ch	RGMII_TX_CLK_1	R/W	12000000h
10001Dh	EXT_TX_CLK_1	R/W	12000000h
10001Eh	EXT_RX_CLK_1	R/W	12000000h
10001Fh	MII_TX_CLK_2	R/W	13000000h
100020h	MII_RX_CLK_2	R/W	13000000h
100021h	RMII_REF_CLK_2	R/W	13000000h
100022h	RGMII_TX_CLK_2	R/W	13000000h
100023h	EXT_TX_CLK_2	R/W	13000000h
100024h	EXT_RX_CLK_2	R/W	13000000h
100025h	MII_TX_CLK_3	R/W	14000000h
100026h	MII_RX_CLK_3	R/W	14000000h
100027h	RMII_REF_CLK_3	R/W	14000000h
100028h	RGMII_TX_CLK_3	R/W	14000000h
100029h	EXT_TX_CLK_3	R/W	14000000h

Offset	Name	Access	Reset Value
10002Ah	EXT_RX_CLK_3	R/W	14000000h
10002Bh	MII_TX_CLK_4	R/W	15000000h
10002Ch	MII_RX_CLK_4	R/W	15000000h
10002Dh	RMII_REF_CLK_4	R/W	15000000h
10002Eh	RGMII_TX_CLK_4	R/W	15000000h
10002Fh	EXT_TX_CLK_4	R/W	15000000h
100030h	EXT_RX_CLK_4	R/W	15000000h

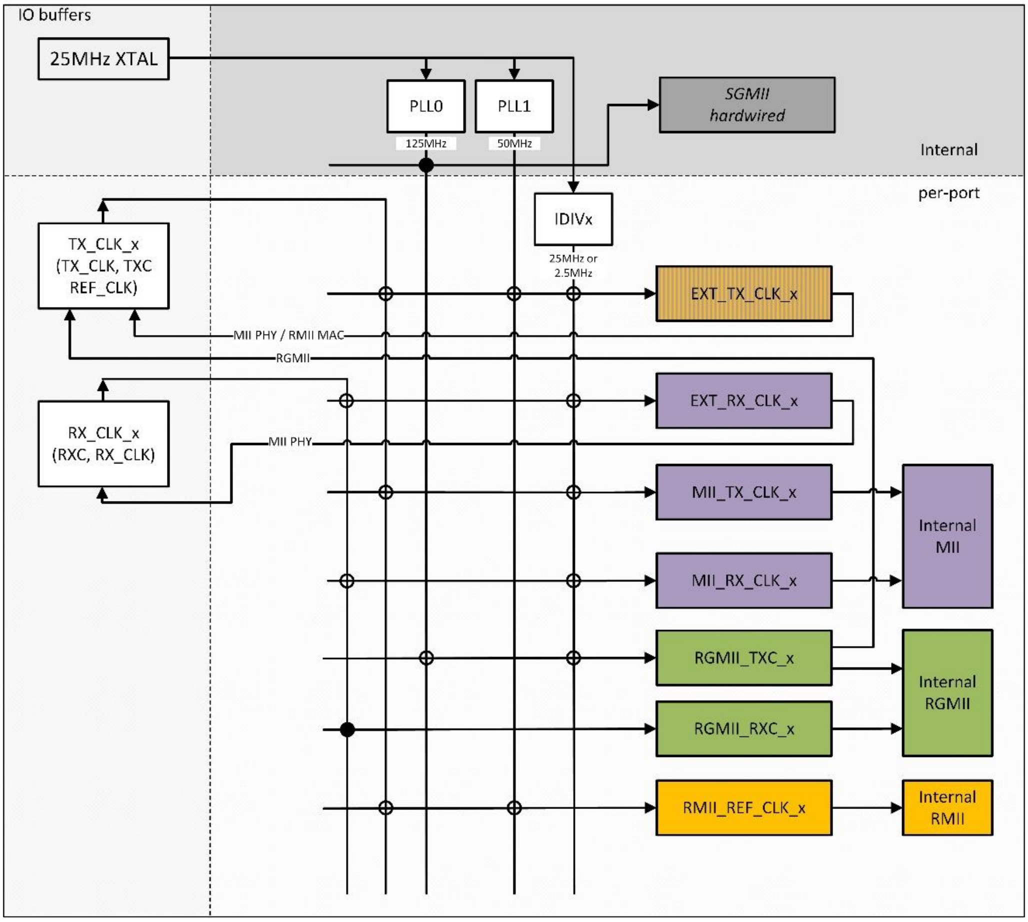


Fig 18. **Generic CGU Overview**

6.1.1 Selectable Clock Sources

The CGU consists of the following sources (once per SJA1105PQRS). Boldfaced names correspond to selectable sources in the CGU. The 5bit hex value in brackets is the value for CLKSRC in the clock sink registers to select this particular source (Table 15).

- OSC
  A fixed 25 MHz XTAL oscillator. This cannot be directly selected as a source.
- **PLL0 (0Bh =** 0°**, 0Ch =** 120°**, 0Dh =** 240°**)**
  A fixed 125 MHz PLL with three phase outputs (0°, 120°, 240°) which is used to drive the digital core clock and the SGMII interface.

  The additional phases (120°, 240°), can be enabled with P23EN, see UM11040. This allows ports running on the same clock frequency but with different phases, e.g. for better EMC behavior.

  Note that 120° and 240° phase outputs are disabled after reset.
- **PLL1 (0Eh =** 0°**, 0Fh =** 120°**, 10h =** 240°**)**
  A fixed 50 MHz PLL with three phase outputs (0°, 120°, 240°).

  The additional phases (120°, 240°), can be enabled with P23EN, see UM11040. This allows ports running on the same clock frequency but with different phases, e.g. for better EMC behavior.

  Note that PLL1 is powered-down after reset.

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 0h for port 0 and a value of 2h 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).
- **IDIV_x (11h+x)**
  An integer divider (by 1 and 10). It is used to feed through 25 MHz or generate 2.5 MHz.

6.1.2 PLL Frequency Selection

PLL1 is normally set to output 50 MHz. For this it must be operated in Integer mode, with PSEL=1h, MSEL=1h and NSEL=0h. If custom settings are selected, F_{CCO} must guaranteed to be between 156 and 320 MHz.

Table 14. PLL Operating Modes
'X' is "don't care"

Operating Mode	FBSET	PD	BYPASS	DIRECT
Integer mode	1	0	0	0
Non-integer mode	0	0	0	0
Direct mode	X	0	0	1
Power down mode	X	1	0	X
Bypass mode	X	X	1	0
Direct bypass mode	X	X	1	1

Integer Mode

In this mode the post-divider is enabled and the feedback divider is set to run on the PLL output clock, which gives the following frequency relations:

$$F_{clkout} = M \times \frac{F_{clkin}}{N}$$

$$F_{cco} = 2 \times P \times F_{clkout} = 2 \times P \times M \times \frac{F_{clkin}}{N}$$

Non-integer Mode

In this mode the post-divider is enabled and the feedback divider is set to run directly on the CCO clock, which gives the following frequency relations:

$$F_{cco} = M \times \frac{F_{clkin}}{N}$$

$$F_{clkout} = \frac{F_{cco}}{2 \times P} = \frac{M}{2 \times P} \times \frac{F_{clkin}}{N}$$

Direct Mode

In this mode the post-divider is disabled and the CCO clock is sent directly to the output, leading to the following frequency equation:

$$F_{clkout} = F_{cco} = M \times \frac{F_{clkin}}{N}$$

Power down mode

While in power down mode, the lock output will be low, to indicate that the PLL is not in lock. When the power down mode is terminated by making PD low, the PLL will resume its normal operation, and will make the lock signal high once it has regained lock on the input clock.

Bypass mode

If FBSEL is set to ‘1’, the feedback clock output will be a divided down version of the PLL’s output clock.

$$F_{clkout} = \frac{F_{clkin}}{2 \times P}$$

$$F_{clkfb} = \frac{F_{clkout}}{M} = \frac{F_{clkin}}{2 \times P \times M}$$

$$F_{clkref} = \frac{F_{clkin}}{N}$$

Direct bypass mode

In this mode, the analog part is placed in power down, the post-divider is disabled and the input clock is sent directly to the output.

6.1.3 Configurable Clock Sinks

Each clock sink is configured via the associated register.

- **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 receiving 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 drive the pad depending on the mode (MAC/PHY).
- **EXT_RX_CLK_x**
  A clock sink that is connected to the RXC/RX_CLK IO pad and can drive the paddepending on the mode (MAC/PHY).
- **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 (except RGMII_RXC_x which is always hardwired) has a corresponding register to select the clock source. The following table specifies the register layout.

Table 15. Clock sink registers 1 to 30 (addr. 100013h to 100030h)

Bits	Symbol	Access	Description
28:24	CLKSRC[5]	R/W	Internal clock source selection
23:12	reserved	R/W	0h

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

Bits	Symbol	Access	Description
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

6.1.4 MII-MAC Mode 100 / 10 Mbps

It must be ensured that the xMII configuration block is setup correctly to bring the given port into MII MAC mode. Fig 19 illustrates the clocking scheme within the CGU. In this mode both clocks are supplied by the external PHY over the TX_CLK and RX_CLK pins to the switch.

Note that for 10 Mbps operation, only the SPEED setting of the MAC Configuration Table needs to be changed. The CGU setup for 10 Mbps and 100 Mbps are identical.

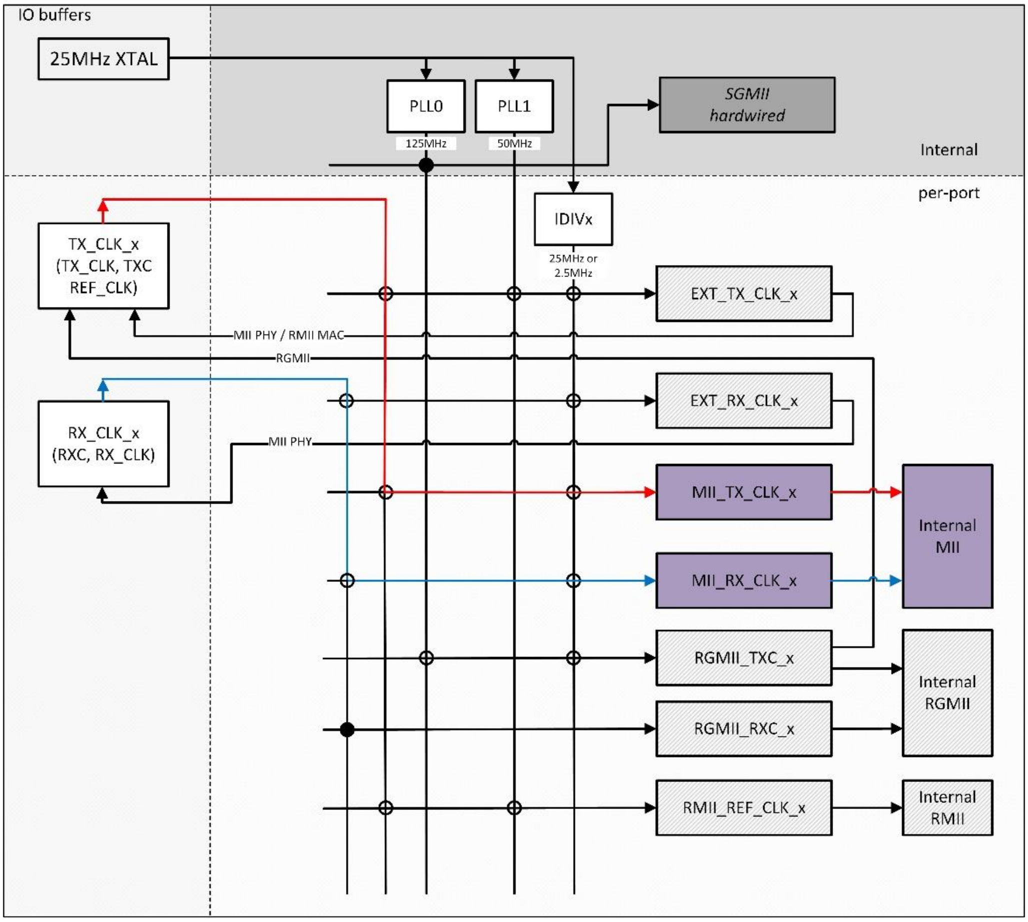


Fig 19. **CGU in MII-MAC configuration**

6.1.4.1 Sources

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

6.1.4.2 Sinks

Table 16. 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 17. 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	0x00100019	0x02000800	Setting CLKSRC of MII_TX_CLK_1 to TX_CLK_1
	MII_RX_CLK_1	0x0010001A	0x03000800	Setting CLKSRC of MII_RX_CLK_1 to RX_CLK_1
2	IDIV2	0x0010000D	0x0A000001	Disable IDIV2
	MII_TX_CLK_2	0x0010001F	0x04000800	Setting CLKSRC of MII_TX_CLK_2 to TX_CLK_2
	MII_RX_CLK_2	0x00100020	0x05000800	Setting CLKSRC of MII_RX_CLK_2 to RX_CLK_2
3	IDIV3	0x0010000E	0x0A000001	Disable IDIV3
	MII_TX_CLK_3	0x00100025	0x06000800	Setting CLKSRC of MII_TX_CLK_3 to TX_CLK_3
	MII_RX_CLK_3	0x00100026	0x07000800	Setting CLKSRC of MII_RX_CLK_3 to RX_CLK_3
4	IDIV4	0x0010000F	0x0A000001	Disable IDIV4
	MII_TX_CLK_4	0x0010002B	0x08000800	Setting CLKSRC of MII_TX_CLK_4 to TX_CLK_4
	MII_RX_CLK_4	0x0010002C	0x09000800	Setting CLKSRC of MII_RX_CLK_4 to RX_CLK_4

6.1.5 MII-PHY Mode 100 Mbps

It must be ensured that the xMII configuration block is setup correctly to bring the given port into MII PHY mode. Fig 20 illustrates the clocking scheme within the CGU. In MII-PHY mode (or revMII), the switch generates TX_CLK and RX_CLK. Note that TXD pins are outputs and RXD are inputs (consult datasheet).

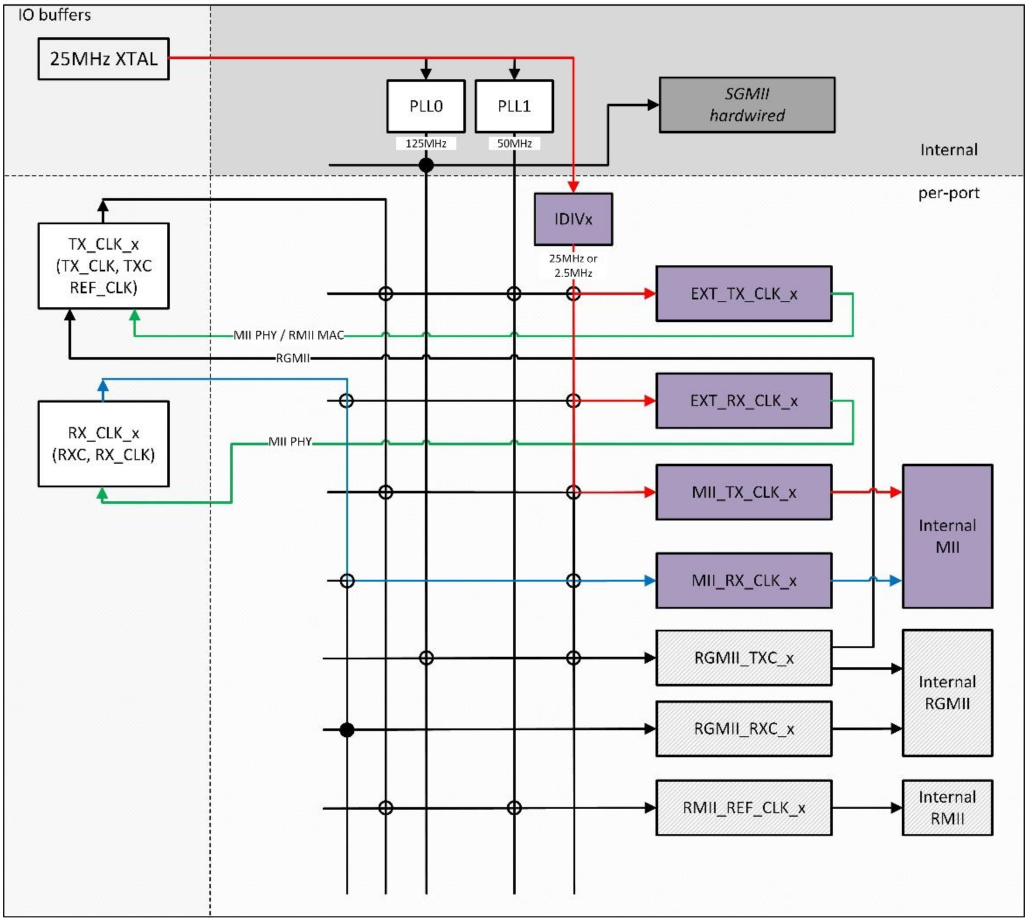


Fig 20. **CGU in MII-PHY configuration**

6.1.5.1 Sources

No configuration required

6.1.5.2 Sinks

Table 18. 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

Sink	Source	CLKSRC
EXT_RX_CLK_x	IDIVx	11h+x

Table 19. 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	0x00100017	0x11000800	Setting CLKSRC of EXT_TX_CLK_0 to IDIV0
	EXT_RX_CLK_0	0x00100018	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	0x00100019	0x12000800	CLKSRC of MII_TX_CLK_1 to IDIV1
	MII_RX_CLK_1	0x0010001A	0x03000800	Setting CLKSRC of MII_RX_CLK_1 to RX_CLK_1
	EXT_TX_CLK_1	0x0010001D	0x12000800	Setting CLKSRC of EXT_TX_CLK_1 to IDIV1
	EXT_RX_CLK_1	0x0010001E	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	0x0010001F	0x13000800	CLKSRC of MII_TX_CLK_2 to IDIV2
	MII_RX_CLK_2	0x00100020	0x05000800	Setting CLKSRC of MII_RX_CLK_2 to RX_CLK_2
	EXT_TX_CLK_2	0x00100023	0x13000800	Setting CLKSRC of EXT_TX_CLK_2 to IDIV2
	EXT_RX_CLK_2	0x00100024	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	0x00100025	0x14000800	CLKSRC of MII_TX_CLK_3 to IDIV3
	MII_RX_CLK_3	0x00100026	0x07000800	Setting CLKSRC of MII_RX_CLK_3 to RX_CLK_3
	EXT_TX_CLK_3	0x00100029	0x14000800	Setting CLKSRC of EXT_TX_CLK_3 to IDIV3

Entity	Register	Address	Value	Description
	EXT_RX_CLK_3	0x0010002A	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	0x0010002B	0x15000800	CLKSRC of MII_TX_CLK_4 to IDIV4
	MII_RX_CLK_4	0x0010002C	0x09000800	Setting CLKSRC of MII_RX_CLK_4 to RX_CLK_4
	EXT_TX_CLK_4	0x0010002F	0x15000800	Setting CLKSRC of EXT_TX_CLK_4 to IDIV4
	EXT_RX_CLK_4	0x00100030	0x15000800	Setting CLKSRC of EXT_TX_CLK_4 to IDIV4

6.1.6 MII-PHY Mode 10 Mbps

It must be ensured that the xMII configuration block is setup correctly to bring the given port into MII PHY mode. In MII-PHY mode (or revMII), the switch generates TX_CLK and RX_CLK. Note that TXD pins are outputs and RXD are inputs (consult datasheet).

6.1.6.1 Sources

No configuration required

6.1.6.2 Sinks

Table 20. 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 21. Register Settings for 10 Mbps MII-PHY

Entity	Register	Address	Value	Description
General	-			
Port 0	IDIV0	0x0010000B	0x0A000824	Enable IDIV0 and divide by 10.
	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	0x00100017	0x11000800	Setting CLKSRC of EXT_TX_CLK_0 to IDIV0
	EXT_RX_CLK_0	0x00100018	0x11000800	Setting CLKSRC of EXT_TX_CLK_0 to IDIV0
Port 1	IDIV1	0x0010000C	0x0A000824	Enable IDIV1 and divide by 10.

Entity	Register	Address	Value	Description
	MII_TX_CLK_1	0x00100019	0x12000800	CLKSRC of MII_TX_CLK_1 to IDIV1
	MII_RX_CLK_1	0x0010001A	0x03000800	Setting CLKSRC of MII_RX_CLK_1 to RX_CLK_1
	EXT_TX_CLK_1	0x0010001D	0x12000800	Setting CLKSRC of EXT_TX_CLK_1 to IDIV1
	EXT_RX_CLK_1	0x0010001E	0x12000800	Setting CLKSRC of EXT_TX_CLK_1 to IDIV1
Port 2	IDIV2	0x0010000D	0x0A000824	Enable IDIV2 and divide by 10.
	MII_TX_CLK_2	0x0010001F	0x13000800	CLKSRC of MII_TX_CLK_2 to IDIV2
	MII_RX_CLK_2	0x00100020	0x05000800	Setting CLKSRC of MII_RX_CLK_2 to RX_CLK_2
	EXT_TX_CLK_2	0x00100023	0x13000800	Setting CLKSRC of EXT_TX_CLK_2 to IDIV2
	EXT_RX_CLK_2	0x00100024	0x13000800	Setting CLKSRC of EXT_TX_CLK_2 to IDIV2
Port 3	IDIV3	0x0010000E	0x0A000824	Enable IDIV3 and divide by 10.
	MII_TX_CLK_3	0x00100025	0x14000800	CLKSRC of MII_TX_CLK_3 to IDIV3
	MII_RX_CLK_3	0x00100026	0x07000800	Setting CLKSRC of MII_RX_CLK_3 to RX_CLK_3
	EXT_TX_CLK_3	0x00100029	0x14000800	Setting CLKSRC of EXT_TX_CLK_3 to IDIV3
	EXT_RX_CLK_3	0x0010002A	0x14000800	Setting CLKSRC of EXT_TX_CLK_3 to IDIV3
Port 4	IDIV4	0x0010000F	0x0A000824	Enable IDIV4 and divide by 10.
	MII_TX_CLK_4	0x0010002B	0x15000800	CLKSRC of MII_TX_CLK_4 to IDIV4
	MII_RX_CLK_4	0x0010002C	0x09000800	Setting CLKSRC of MII_RX_CLK_4 to RX_CLK_4
	EXT_TX_CLK_4	0x0010002F	0x15000800	Setting CLKSRC of EXT_TX_CLK_4 to IDIV4
	EXT_RX_CLK_4	0x00100030	0x15000800	Setting CLKSRC of EXT_TX_CLK_4 to IDIV4

6.1.7 RMII-MAC Mode 100 / 10 Mbps

If the SJA1105PQRS is operated in RMII-MAC mode, it operates as a MAC according to Fig. 1 in the RMII spec and does not output CRS information.

It must be ensured that the xMII configuration block is setup correctly to bring the given port into RMII MAC mode. Fig 21 illustrates the clocking scheme within the CGU.

© NXP Semiconductors N.V. 2020. All rights reserved.

Note that for 10 Mbps operation, only the SPEED setting of the MAC Configuration Table needs to be changed. The CGU setup for 10 Mbps and 100 Mbps are identical.

For long REF_CLK traces, e.g. if the interface is fed over a connector, it is recommended to set CFG_PAD_MIIx_TX.CLK_IH. This enables hysteresis for the REF_CLK pin in RMII-MAC mode. This improves clocking despite REF_CLK being an output.

In RMII mode the IO cell is used as an output to drive REF_CLK and at the same time the signal is read back from the REF_CLK trace. The internal clock signal as used to sample the data lines always is always derived from the trace. Long traces will show ringing and reflection. Thus, hysteresis will improve the clock quality.

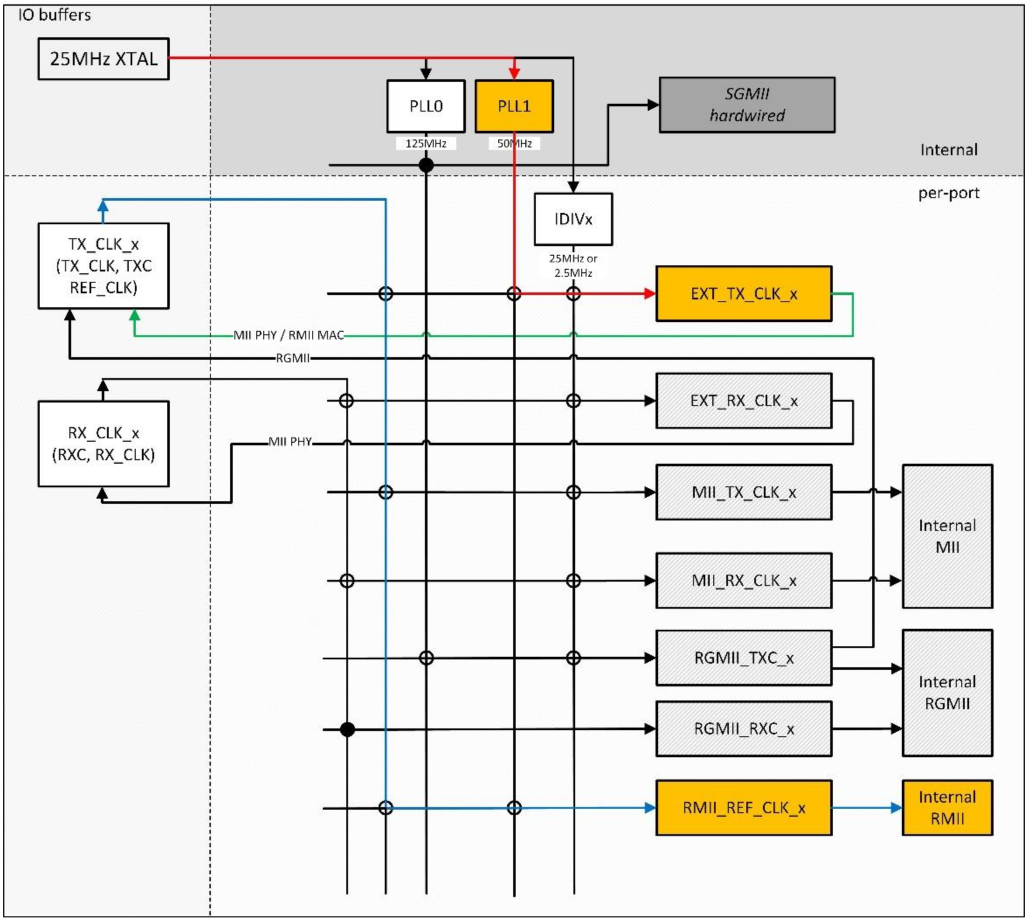


Fig 21. **CGU in RMII-MAC configuration**

Fig 22 shows the wiring of the SJA1105PQRS with a PHY. The PHY needs no crystal, as it receives its clock via REF_CLK from the MAC.

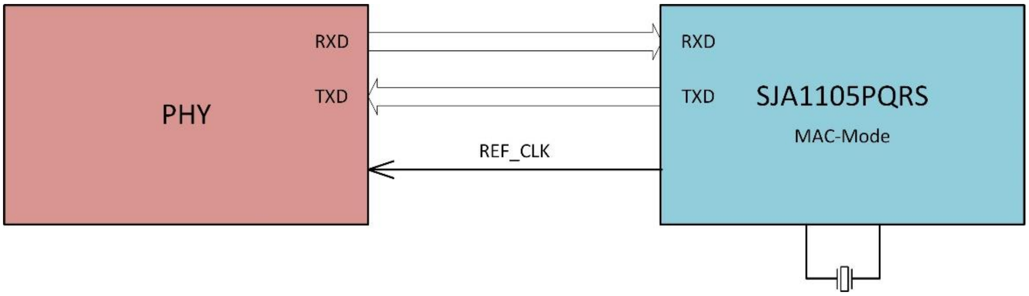


Fig 22. **RMII with SJA1105PQRS in MAC mode**

6.1.7.1 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.

6.1.7.2 Sinks

Table 22. Source to sink mapping

Sink	Source	CLKSRC
RMII_REF_CLK_x	TX_CLK_x	02h*x
EXT_TX_CLK_x	PLL1	0Eh (alternatively 0Fh or 10h)

Table 23. 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	0x00100017	0x0E000800	Setting CLKSRC of EXT_TX_CLK_0 to PLL1
Port 1	IDIV1	0x0010000C	0x0A000001	Disable IDIV1
	RMII_REF_CLK_1	0x0010001B	0x02000800	Setting CLKSRC of RMII_REF_CLK_1 to TX_CLK_1
	EXT_TX_CLK_1	0x0010001D	0x0E000800	Setting CLKSRC of EXT_TX_CLK_1 to PLL1
Port 2	IDIV2	0x0010000D	0x0A000001	Disable IDIV2
	RMII_REF_CLK_2	0x00100021	0x04000800	Setting CLKSRC of RMII_REF_CLK_2 to TX_CLK_2
	EXT_TX_CLK_2	0x00100023	0x0E000800	Setting CLKSRC of EXT_TX_CLK_2 to PLL1
Port 3	IDIV3	0x0010000E	0x0A000001	Disable IDIV3

Entity	Register	Address	Value	Description
	RMII_REF_CLK_3	0x00100027	0x06000800	Setting CLKSRC of RMII_REF_CLK_3 to TX_CLK_3
	EXT_TX_CLK_3	0x00100029	0x0E000800	Setting CLKSRC of EXT_TX_CLK_3 to PLL1
Port 4	IDIV4	0x0010000F	0x0A000000	Disable IDIV4
	RMII_REF_CLK_4	0x0010002D	0x08000800	Setting CLKSRC of RMII_REF_CLK_4 to TX_CLK_4
	EXT_TX_CLK_4	0x0010002F	0x0E000800	Setting CLKSRC of EXT_TX_CLK_4 to PLL1

6.1.7.3 Configuration example

```
#Setting clocks and system for RMII-MAC operation on port 3 and port 4
### 50 MHz on PLL1
platform.spi_write( 0x0010000A, 0x0A010941 )
platform.spi_write( 0x0010000A, 0x0A010940 )
for i in [ 3, 4 ]:
    # disable the port's IDIVi
    platform.spi_write( 0x0010000B+i,   (0x0001 | (0x0a << 24)))
    # set RMII_REF_CLK_x to TX_CLK_x
    platform.spi_write( 0x00100015+i*6, (0x0800 | (i*2) << 24))
    # set EXT_TX_CLK_x to PLL1
    platform.spi_write( 0x00100017+i*6, (0x0800 | (0xe << 24)))
```

Fig 23. **RMII-MAC configuration example**

6.1.8 RMII PHY Mode 100 / 10 Mbps

If the SJA1105PQRS is operated in RMII-PHY mode, it operates as a PHY according to Fig.1 in the RMII spec and outputs CRS information as additional JK groups.

It must be ensured that the xMII configuration block is setup correctly to bring the given port into RMII-PHY mode. Fig 24 illustrates the clocking scheme within the CGU.

For 10 Mbps operation, only the SPEED setting of the MAC Configuration Table needs to be changed. The CGU setup for 10 Mbps and 100 Mbps are identical.

Note that the RMII-PHY fully emulates a PHY (see RMII 1.2 specification). This means the switch will transmit /J/ and /K/ symbols prior to the Ethernet preamble. This mode typically requires a real MAC as a peer device.

© NXP Semiconductors N.V. 2020. All rights reserved.

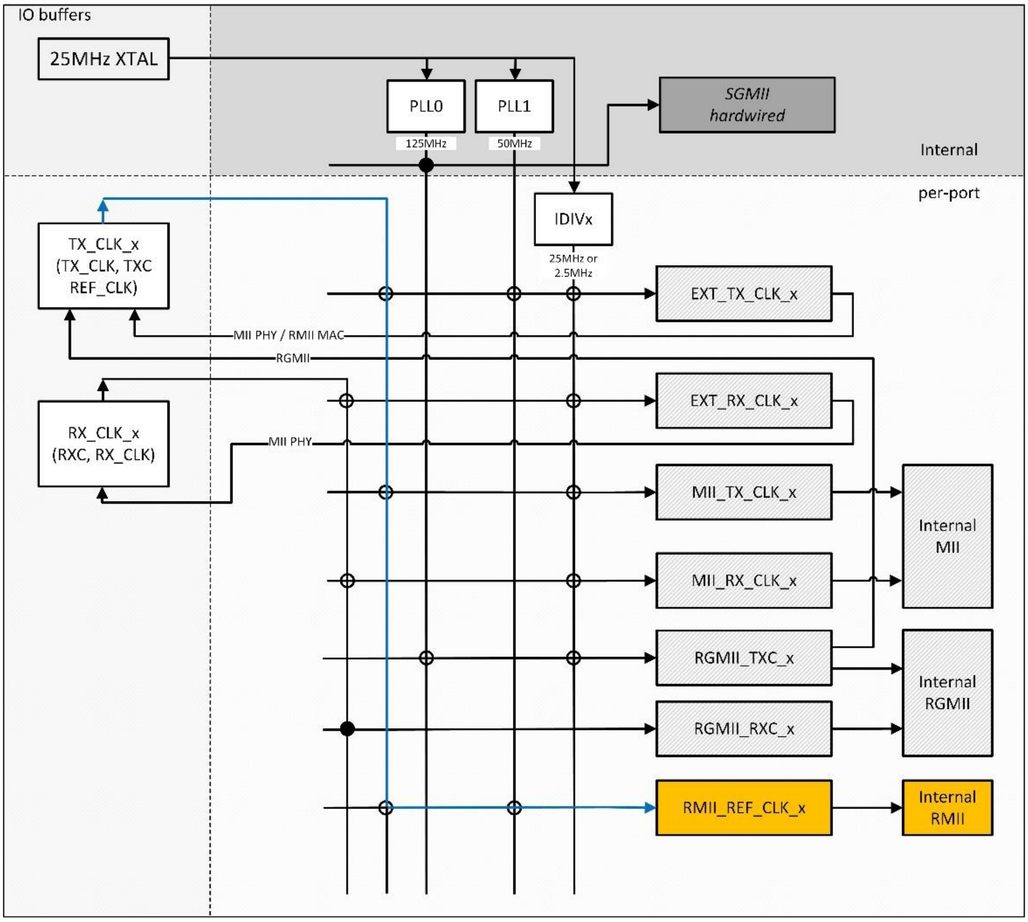


Fig 24. **CGU in RMII-PHY configuration**

Fig 25 shows how the SJA1105PQRS can act as a PHY when connected to a MAC, e.g. to a microcontroller. REF_CLK comes from the SOC, and is connected to the SJA1105PQRS’s REF_CLK pin.

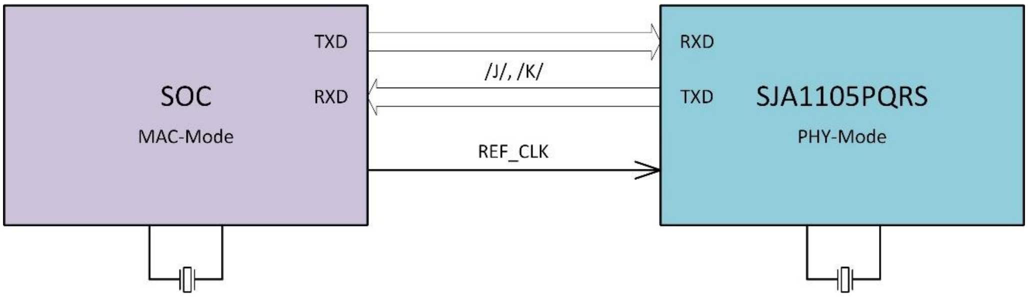


Fig 25. **RMII with SJA1105PQRS in PHY mode**

6.1.8.1 Sources

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

6.1.8.2 Sinks

Table 24. Source to sink mapping

Sink	Source	CLKSRC
RMII_REF_CLK_x	TX_CLK_x	02h*x

Table 25. 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	0x0010001B	0x02000800	Setting CLKSRC of RMII_REF_CLK_1 to TX_CLK_1
Port 2	IDIV2	0x0010000D	0x0A000001	Disable IDIV2
	RMII_REF_CLK_2	0x00100021	0x04000800	Setting CLKSRC of RMII_REF_CLK_2 to TX_CLK_2
Port 3	IDIV3	0x0010000E	0x0A000001	Disable IDIV3
	RMII_REF_CLK_3	0x00100027	0x06000800	Setting CLKSRC of RMII_REF_CLK_3 to TX_CLK_3
Port 4	IDIV4	0x0010000F	0x0A000000	Disable IDIV4
	RMII_REF_CLK_4	0x0010002D	0x08000800	Setting CLKSRC of RMII_REF_CLK_4 to TX_CLK_4

6.1.9 RMII-PHY Mode with a MAC without REF_CLK

If the SJA1105PQRS is operated in RMII-PHY mode, it operates as a PHY according to Fig. 1 in the RMII spec and outputs CRS information as additional JK groups.

There are use cases, where the SJA1105PQRS must work in PHY mode, because it is directly connected to a MAC, but this MAC cannot provide the REF_CLK, as a MAC would normally do.

In the SJA1105PQRS implementation, the REF_CLK direction is linked to the MAC/PHY mode. In that sense, in RMII-MAC mode, REF_CLK is output and in RMII-PHY mode, it is input. This means, in the above mentioned scenario, the SJA1105PQRS cannot provide REF_CLK either.

There are two solutions to this conflict: add an additional clock source for REF_CLK or let the SJA1105PQRS provide the REF_CLK, despite being used in PHY mode.

6.1.9.1 RMII-PHY Mode and external oscillator

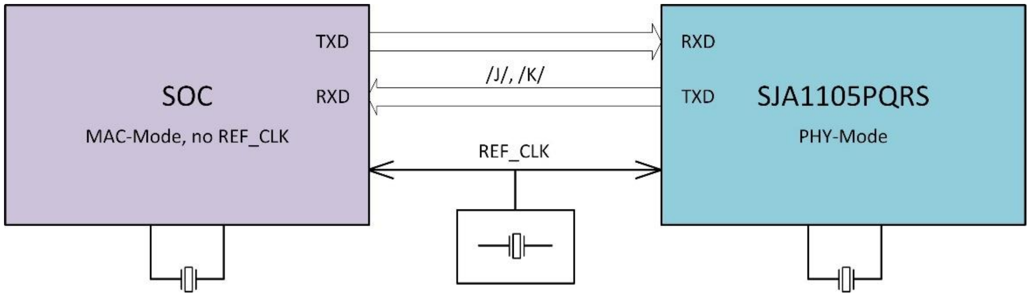


Fig 26. **RMII with SJA1105PQRS in PHY mode and additional oscillator for REF_CLK**

This is a straight-forward solution using the configuration settings mentioned before. However, the additional oscillator adds to the HW cost.

6.1.9.2 SJA1105PQRS used as a PHY and providing REF_CLK

To avoid the additional external oscillator for REF_CLK(Fig 26), the SJA1105PQRS can provide the REF_CLK, despite acting as a PHY.

The setup is not intuitive, because despite the switch is used as a PHY, it nevertheless must be *configured in MAC* mode (Fig 27), exactly as described in ch. 6.1.7.

This is a more detailed view into the topic:

- The switch’s REF_CLK must output a 50MHz clock. Therefore, the CGU must be configured as shown in Fig 21. This is the same configuration used in RMII-MAC mode, and the settings in Table 23 can be used without change.
- The REF_CLK pin direction must be set to output. REF_CLK direction is coupled to the port’s PHY/MAC setting (static config stream). MAC means REF_CLK is an output, and PHY means input.
- The PHY/MAC setting has only limited influence on the data transfer pins in RMII mode: all pins keep their semantics and direction with the following exceptions:
  - Direction of REF_CLK changes - refer to Fig 7 of the SJA1105PQRS data sheet.
  - In PHY mode the switch would send /J/ and /K/ symbols on TXD as a carrier sense (CRS) prior to the preamble. This is observed as zero nibbles at the receiving device (MAC of the SOC). However, from a receiver perspective, this is not mandatory, so a MAC compatible with the RMII 1.2 spec does not require these symbols on its receive lane.
  - In MAC mode the SJA1105PQRS would not send /J/ and /K/, but since spec conforming MACs don’t require them, this does not pose a problem.

  Besides these topics, there is no other difference between MAC and PHY mode for RMII in the SJA1105PQRS. Since the topics have no impact in the scenario at hand, it is possible to use the SJA1105PQRS in MAC mode although it acts as a PHY. The desired side effect is, that it provides REF_CLK, which is needed for connecting to a SOC with a MAC which cannot provide REF_CLK.

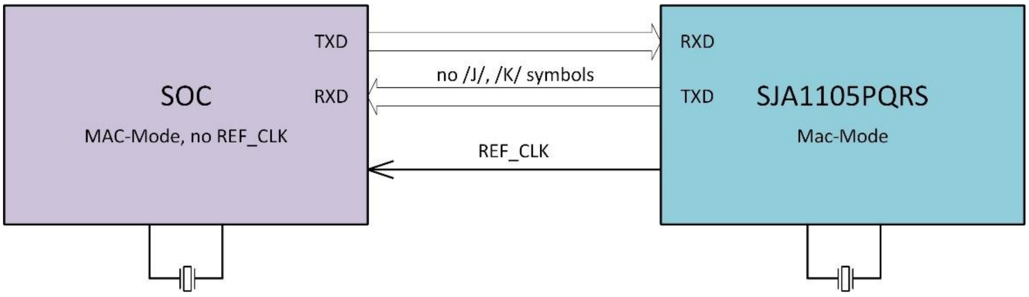


Fig 27. **RMII with SJA1105PQRS and a SOC without REF_CLK in MAC mode**

6.1.10 GMII Mode 1 Gbps

It must be ensured that the xMII configuration block is setup correctly to bring the given port into RGMII mode. Fig 28 illustrates the clocking scheme within the CGU. For RGMII, the pad speed has to be adjusted in addition. The relevant register writes are included in the sink configuration.

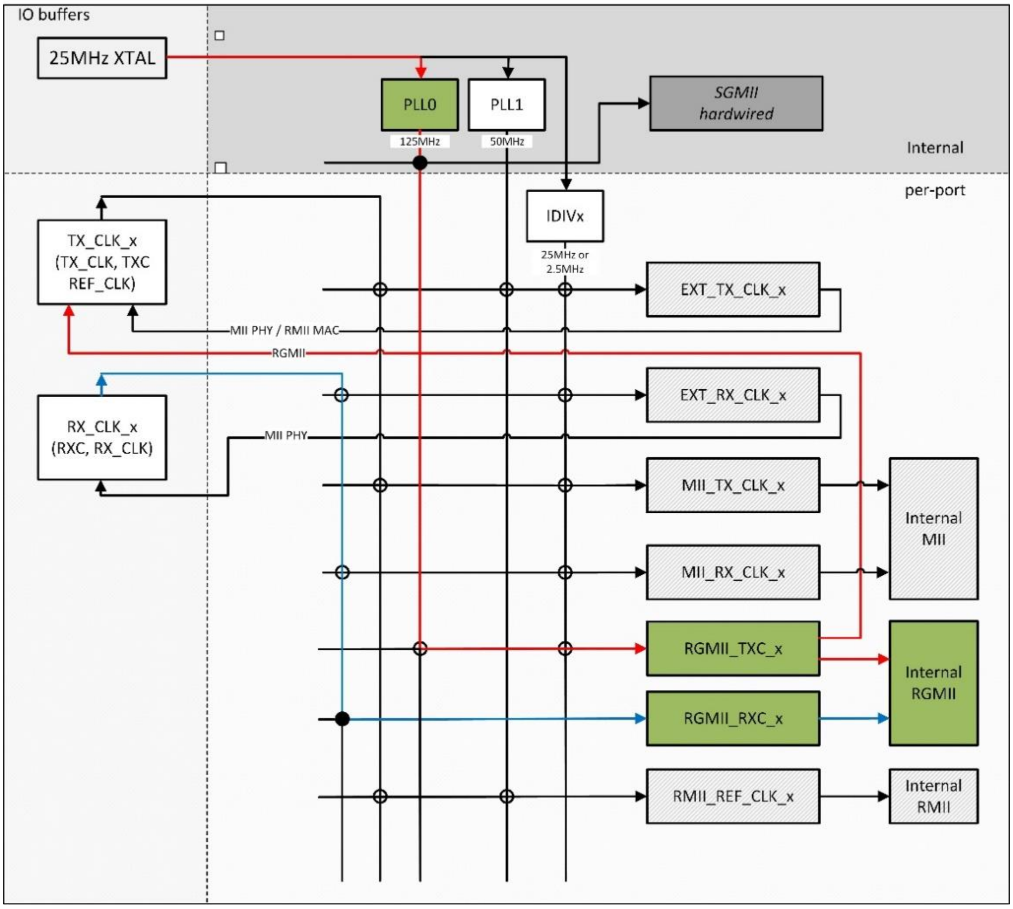


Fig 28. **CGU in RGMII 1Gbps configuration**

6.1.10.1 Sources

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

6.1.10.2 Sinks

Table 26. Source to sink mapping

Sink	Source	CLKSRC
RGMII_RXC_x	*hardwired*	-
RGMII_TXC_x	PLL0	0Bh (alternatively 0Ch or 0Dh)

Table 27. 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		*See slew rate section*
Port 1	IDIV1	0x0010000C	0x0A000001	Disable IDIV1
	RGMII_TXC_1	0x0010001C	0x0B000800	Setting CLKSRC of RGMII_TXC_1 to PLL0
	CFG_PAD_MII1_TX	0x00100802		*See slew rate section*
Port 2	IDIV2	0x0010000D	0x0A000001	Disable IDIV2
	RGMII_TXC_2	0x00100022	0x0B000800	Setting CLKSRC of RGMII_TXC_2 to PLL0
	CFG_PAD_MII2_TX	0x00100804		*See slew rate section*
Port 3	IDIV3	0x0010000E	0x0A000001	Disable IDIV3
	RGMII_TXC_3	0x00100028	0x0B000800	Setting CLKSRC of RGMII_TXC_3 to PLL0
	CFG_PAD_MII3_TX	0x00100806		*See slew rate section*
Port 4	IDIV4	0x0010000F	0x0A000001	Disable IDIV4
	RGMII_TXC_4	0x0010002E	0x0B000800	Setting CLKSRC of RGMII_TXC_4 to PLL0
	CFG_PAD_MII4_TX	0x00100808		*See slew rate section*

6.1.11 RGMII Mode 100 Mbps

It must be ensured that the xMII configuration block is setup correctly to bring the given port into RGMII mode for 100Mbps. For RGMII, the pad speed has to be adjusted in addition. The relevant register writes are included in the sink configuration.

6.1.11.1 Sources

The IDIV_x must be enabled and set to 1 for an output of 25 MHz.

© NXP Semiconductors N.V. 2020. All rights reserved.

6.1.11.2 Sinks

Table 28. Source to sink mapping

Sink	Source	CLKSRC
RGMII_RXC_x	*hardwired*	-
RGMII_TXC_0	IDIV0	11h
RGMII_TXC_1	IDIV1	12h
RGMII_TXC_2	IDIV2	13h
RGMII_TXC_3	IDIV3	14h
RGMII_TXC_4	IDIV4	15h

Table 29. Register Settings for 100Mbps RGMII

Entity	Register	Address	Value	Description
General	-			
Port 0	IDIV0	0x0010000B	0x0A000000	IDIV0 divide by 1
	RGMII_TXC_0	0x00100016	0x11000800	Setting CLKSRC of RGMII_TXC_0 to IDIV0
	CFG_PAD_MII0_TX	0x00100800		*See slew rate section*
Port 1	IDIV1	0x0010000C	0x0A000000	IDIV1 divide by 1
	RGMII_TXC_1	0x0010001C	0x12000800	Setting CLKSRC of RGMII_TXC_1 to IDIV1
	CFG_PAD_MII1_TX	0x00100802		*See slew rate section*
Port 2	IDIV2	0x0010000D	0x0A000000	IDIV2 divide by 1
	RGMII_TXC_2	0x00100022	0x13000800	Setting CLKSRC of RGMII_TXC_2 to IDIV2
	CFG_PAD_MII2_TX	0x00100804		*See slew rate section*
Port 3	IDIV3	0x0010000E	0x0A000000	IDIV3 divide by 1
	RGMII_TXC_3	0x00100028	0x14000800	Setting CLKSRC of RGMII_TXC_3 to IDIV3
	CFG_PAD_MII3_TX	0x00100806		*See slew rate section*
Port 4	IDIV4	0x0010000F	0x0A000000	IDIV4 divide by 1
	RGMII_TXC_4	0x0010002E	0x15000800	Setting CLKSRC of RGMII_TXC_4 to IDIV4
	CFG_PAD_MII4_TX	0x00100808		*See slew rate section*

6.1.12 RGMII Mode 10 Mbps

It must be ensured that the xMII configuration block is setup correctly to bring the given port into RGMII 10 Mbps mode. For RGMII, the pad speed has to be adjusted in addition. The relevant register writes are included in the sink configuration.

6.1.12.1 Sources

The IDIV_x must be enabled and set to 10 for an output of 2.5 MHz.

6.1.12.2 Sinks

Table 30. Source to sink mapping

Sink	Source	CLKSRC
RGMII_RXC_x	***hardwired***	**-**
RGMII_TXC_0	IDIV0	11h
RGMII_TXC_1	IDIV1	12h
RGMII_TXC_2	IDIV2	13h
RGMII_TXC_3	IDIV3	14h
RGMII_TXC_4	IDIV4	15h

Table 31. Register Settings for 10 Mbps RGMII

Entity	Register	Address	Value	Description
General	-			
Port 0	IDIV0	0x0010000B	0x0A000824	IDIV0 divide by 10
	RGMII_TXC_0	0x00100016	0x11000800	Setting CLKSRC of RGMII_TXC_0 to IDIV0
	CFG_PAD_MII0_TX	0x00100800		*See slew rate section*
Port 1	IDIV1	0x0010000C	0x0A000824	IDIV1 divide by 10
	RGMII_TXC_1	0x0010001C	0x12000800	Setting CLKSRC of RGMII_TXC_1 to IDIV1
	CFG_PAD_MII1_TX	0x00100802		*See slew rate section*
Port 2	IDIV2	0x0010000D	0x0A000824	IDIV2 divide by 10
	RGMII_TXC_2	0x00100022	0x13000800	Setting CLKSRC of RGMII_TXC_2 to IDIV2
	CFG_PAD_MII2_TX	0x00100804		*See slew rate section*
Port 3	IDIV3	0x0010000E	0x0A000824	IDIV3 divide by 10
	RGMII_TXC_3	0x00100028	0x14000800	Setting CLKSRC of RGMII_TXC_3 to IDIV3
	CFG_PAD_MII3_TX	0x00100806		*See slew rate section*
Port 4	IDIV4	0x0010000F	0x0A000824	IDIV4 divide by 10
	RGMII_TXC_4	0x0010002E	0x15000800	Setting CLKSRC of RGMII_TXC_4 to IDIV4
	CFG_PAD_MII4_TX	0x00100808		*See slew rate section*

6.1.13 RGMII clock delay

6.1.13.1 RGMII clock delay concept

The RGMII specification requires that clock and data change at the same time when leaving the sending side. This is shown by example for the timing of the TX direction in Fig 29. The top half shows the timing relation at the sender side, and the bottom half on the receiver side.

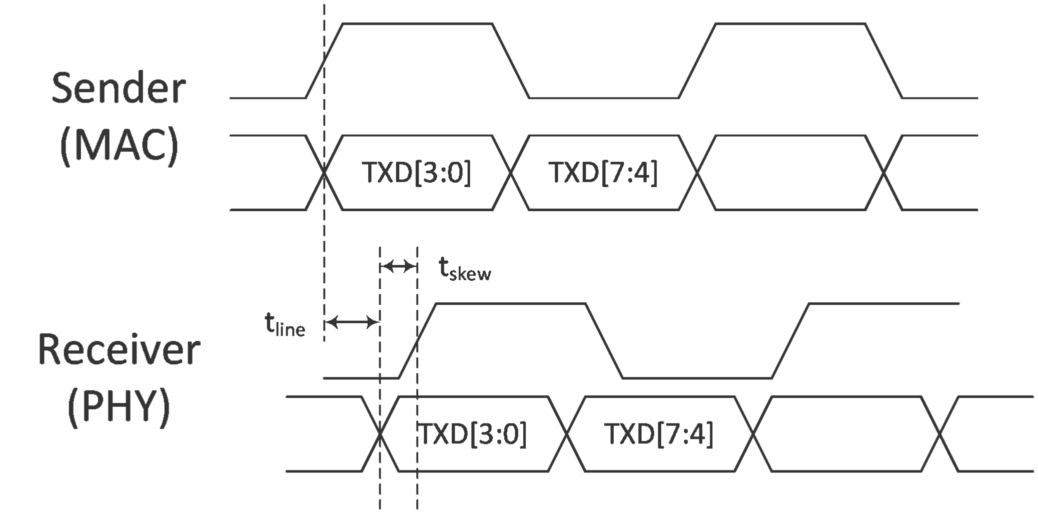


Fig 29. **RGMII timing delay**

Due to delay caused by the wires between MAC and PHY, there is an unavoidable, common delay for clock and data lines: t_{line}, which depends on the trace length. All PCB traces for the TX path must be length matched, so t_{line} is equal.

The RGMII specification requires an additional delay for the clock line t_{skew}. This is to meet setup time requirements on the receiving side, which means, that data and control lines have stabilized before the clock edge causes the data to be latched. Stabilization needs some time. t_{skew} is in the range of 1..2.6ns, which is substantial at a clock cycle time of 8ns@125MHz. Both clock edges are used (DDR).

There are a lot of options where and how to implement the delay. The easiest is to just make the clock line PCB trace a little longer than the data lines. However, most of the time board space is scarce for meanders, and an integrated delay line device might be an alternative. Trace length and delay line component are shown as options B and E in Fig 30. Many MACs and PHYs also allow to configure delays. This is indicated by A, C, D, F.

Please note, that exactly one delay option is needed for TX_CLK (A, B, or C) and another one for RX_CLK (D, E, F).

The SJA1105PQRS implements configurable RGMII delay with tunable delay lines TDL.

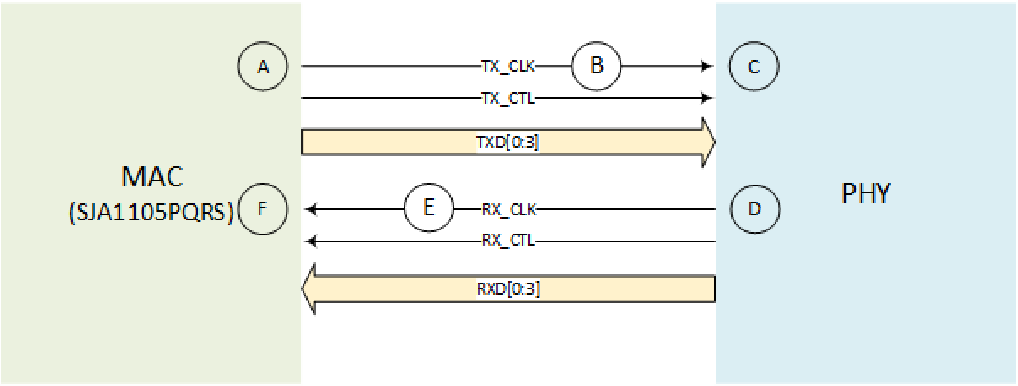


Fig 30. **Options to implement RGMII clock delay**

6.1.13.2 RGMII TDL Configuration

The tunable delay lines are programmed through registers CFG_PAD_MIIx_ID (addr. 100810h - 100814h).

The switch has a configurable delay in TXC as well as RXC for each MII/RMII/RGMII port. After reset, the delay is disabled (BYPASS). The TDL is enabled by bringing it out of BYPASS:

```
txc_delay = 0x8
rxc_delay = 0x8
spi_write(0x100810, (txc_delay << 2) | (rxc_delay << 10))
```

Fig 31. **TDL Configuration Code**

The delay is configured in degree phase of the clock cycle. The absolute delay time is computed as follows:

$$\text{delay[ns]} = \text{cycle\_time[ns]} * \left((73.8° + \text{delay\_value} \times 0.9°) \; / \; 360° \right)$$

The default delay_value (or delay_tune as named in UM11040) is 0x8, which corresponds to 1.8ns (@125 MHz) (see Fig 33)

For a delay_value range of 0..31, the absolute delay time can be set between 1.64..2.26ns (@125MHz). Please note, that some delay_values are not recommended, since the delay line is only specified for an angle range of 76°..96°.

For the correspondence between a cycle angle and the delay time see Fig 32

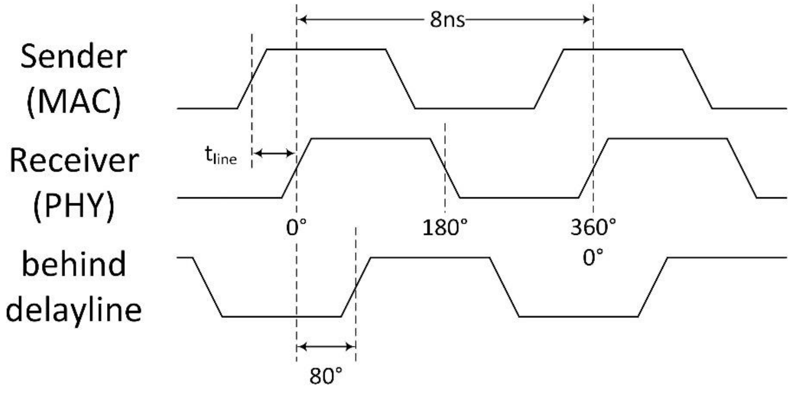
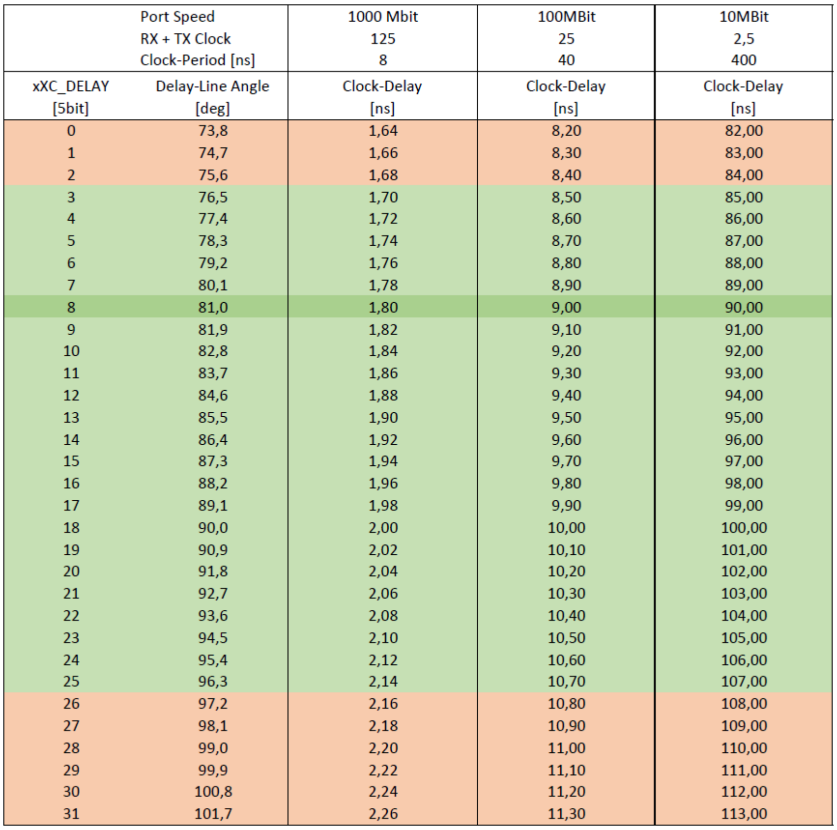


Fig 32. **Delay line setting in degree**

For RGMII-100 and RGMII-10 the configured delay scales up with the clock cycle time, as it is specified in terms of an angle.



	Port Speed	1000 Mbit	100MBit	10MBit
	RX + TX Clock	125	25	2,5
	Clock-Period [ns]	8	40	400
xXC_DELAY	Delay-Line Angle	Clock-Delay	Clock-Delay	Clock-Delay
[5bit]	[deg]	[ns]	[ns]	[ns]
0	73,8	1,64	8,20	82,00
1	74,7	1,66	8,30	83,00
2	75,6	1,68	8,40	84,00
3	76,5	1,70	8,50	85,00
4	77,4	1,72	8,60	86,00
5	78,3	1,74	8,70	87,00
6	79,2	1,76	8,80	88,00
7	80,1	1,78	8,90	89,00
8	81,0	1,80	9,00	90,00
9	81,9	1,82	9,10	91,00
10	82,8	1,84	9,20	92,00
11	83,7	1,86	9,30	93,00
12	84,6	1,88	9,40	94,00
13	85,5	1,90	9,50	95,00
14	86,4	1,92	9,60	96,00
15	87,3	1,94	9,70	97,00
16	88,2	1,96	9,80	98,00
17	89,1	1,98	9,90	99,00
18	90,0	2,00	10,00	100,00
19	90,9	2,02	10,10	101,00
20	91,8	2,04	10,20	102,00
21	92,7	2,06	10,30	103,00
22	93,6	2,08	10,40	104,00
23	94,5	2,10	10,50	105,00
24	95,4	2,12	10,60	106,00
25	96,3	2,14	10,70	107,00
26	97,2	2,16	10,80	108,00
27	98,1	2,18	10,90	109,00
28	99,0	2,20	11,00	110,00
29	99,9	2,22	11,10	111,00
30	100,8	2,24	11,20	112,00
31	101,7	2,26	11,30	113,00

Fig 33. **RGMII delay line settings**

6.1.13.3 Frequency Transitions

The TDL assumes a **steady state** clock input. For a RX TDL this is the clock from the port's RXC pin, and for a TX TDL it is the clock from the CGU.

No rapid frequency transitions must occur at the TDL's input. A rapid frequency transition is a change where the silent time is **less than** 2µs, see Fig 34. In case a rapid frequency change occurs (e.g. from 25 MHz to 125 MHz) the output frequency of the TDL is undefined and remains so until recovered as described below. If a silent time of >2µs is detected by the TDL, the new frequency is always recognized correctly and no further action is required.

In case the frequency is expected to change rapidly (t_{silent} ≤ 2µs), disable TDL during the frequency change phase by temporarily asserting BYPASS, or (for a RX TDL) ensure through external means that the RXC clock is temporarily stopped as described below to recover.

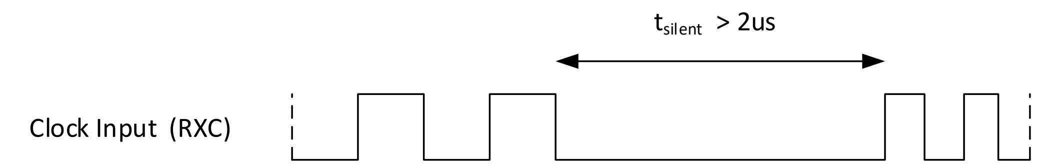


Fig 34. **TDL input clock transition phase**

Frequency transitions often occur during auto negotiation or link-startup (e.g. if 1000Base-T PHYs are used). Some clock changes cannot be anticipated or controlled, e.g. during auto negotiation or link-startup, and are only reported **after** they have occurred. For recovering the TDL to work properly with the new frequency, one of the following actions are recommended:

- temporarily pause the TDL's **input clock for at least 2µs** as shown in the Figure above**.** The TDL can cope with frequency transitions that have a defined transition time (TDL's input is not toggling) of more than 2µs.
- power cycle the TDL: set the PD bit and clear it after >2µsec. Temporarily entering bypass mode for >2µsec does not always recover a TDL.

6.1.13.4 Further hints

Make sure, the TDLs are out of power down mode when in use. Power down control bits (PD bits) are also located in the CFG_PAD_MIIx_ID register. After reset, power down is enabled.

6.1.14 Slew Rate Settings

The IO cells in the SJA1105PQRS allow to select the slew rate of the MII/RMII/RGMII digital outputs. This can be selected via the registers CFG_PAD_MIIx_TX and CFG_PAD_MIIx_RX. It is advised to set the slew rate settings to the minimum possible setting to achieve best EMC performance.

The slew rate can be set individually for the following pin groups.

- TXD3, TXD2
- TXD1, TXD0
- TX_EN, TX_ER

© NXP Semiconductors N.V. 2020. All rights reserved.

- TX_CLK
- RX_CLK (only in MII-PHY mode)

Table 32. Supported Slew Rates

Slew Rate	Voltage [V]	Maximum Frequency [MHz]	Support Modes
High-speed mode	3.3	125	RGMII
	2.5	125	RGMII
	1.8	125	RGMII
Fast-speed mode	3.3	125	MII, RMII, RGMII
	2.5	125	MII, RMII, RGMII
	1.8	125	MII, RGMII
Medium-speed mode	3.3	125	MII, RMII, RGMII
	2.5	125	MII, RMII, RGMII
	1.8	50	MII
Low-speed mode	3.3	50	MII, RMII
	2.5	50	MII, RMII
	1.8	25	MII

Table 33. Recommended Slew Rates

Mode	Voltage [V]	Recommended Slew Rate
RGMII	3.3	Medium-speed mode
	2.5	Medium-speed mode
	1.8	Fast-speed mode
RMII	3.3	Low-speed mode
	2.5	Low-speed mode
	1.8	*Not supported*
MII	3.3	Low-speed mode
	2.5	Low-speed mode
	1.8	Low-speed mode

Note for RMII-MAC mode: For long REF_CLK traces, e.g. if the interface is fed over a connector, it is recommended to set CFG_PAD_MIIx_TX.CLK_IH. See Section 6.1.7 for an explanation.

6.2 SGMII

For the SJA1105R/S devices port 4 is hardwired to an internal SGMII PHY. Other PHY interface options are not possible on port 4 of these devices. In order to use the port, the xMII Mode Parameters must be set to SGMII. No special CGU setup is required as the digital clock is always supplied automatically to the SGMII PHY. However, the SPEED in the MAC Configuration table must match the speed in which the SGMII PHY operates.

The standard PHY registers are available for the SGMII PHY at address range 1f0000h..1f000fh. The SGMII PHY can be entirely configured either with the static config stream or entirely via the dynamic interface, or a mix of both.

6.2.1 Initialization

After reset, the SGMII PHY starts-up with inverted polarity on the TX pair. This might have to be changed – depending how the SGMII pins are wired (Fig 11) - by configuring TX_POL_INV and RX_POL_INV properly (register DIGITAL_CONTROL_2 address 0x1f80E1).

TX_POL_INV must be set to 1 and RX_POL_INV must be set to 0.

6.2.2 Disabled Auto-Negotiation

To disable auto negotiation, the AUTONEG_ENABLE flag in the BASIC_CONTROL register (address 0x1f0000) must be deasserted and SPEED_SELECT must be set (e.g. to 1 Gbps mode)

6.2.3 Auto-Negotiation

6.2.3.1 SGMII Auto-Negotiation

The device supports SGMII Auto-Negotiation as specified in SGMII. The Auto-Negotiation process is as per 1000Base-X Clause 37 with SGMII specific changes described in the SGMII standard. In particular, the contents of the Config_Reg register and hence the exchanged capabilities differentiate from 1000Base-X Clause 37.

This configuration is used e.g. for cascading switches (Fig 36) or for connecting to a host (Fig 35).

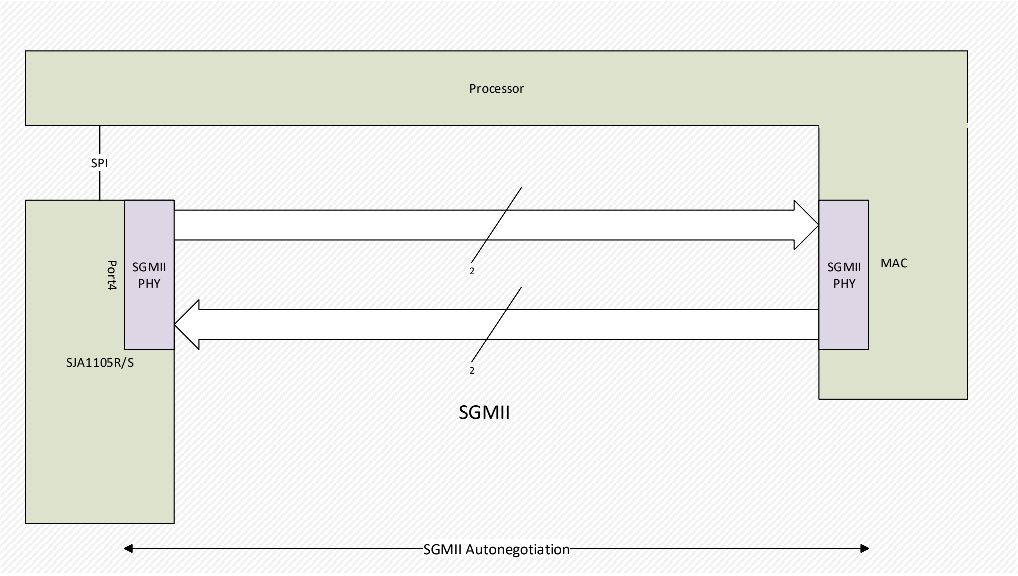


Fig 35. **MAC-Host link with SGMII**

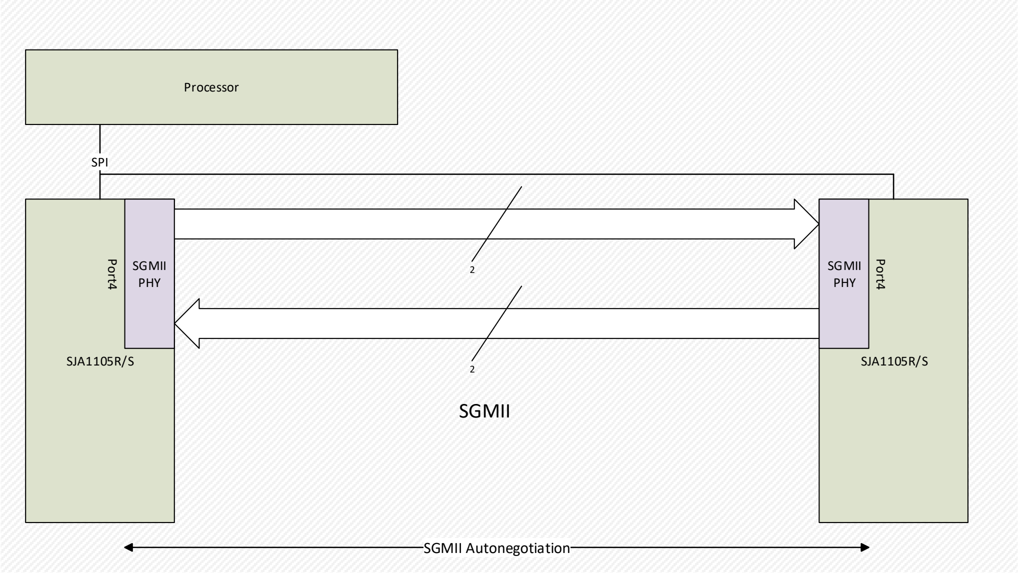


Fig 36. **Cascaded switches with SGMII**

For automotive applications, it is not recommended to use auto-negotiation in these scenarios, as the link capabilities are known at design time, and can be configured consistently via the management interface for both ends.

There are use-cases in which Auto-Negotiation is preferred, for instance if an SFP module is used.

6.2.3.2 SFP Module

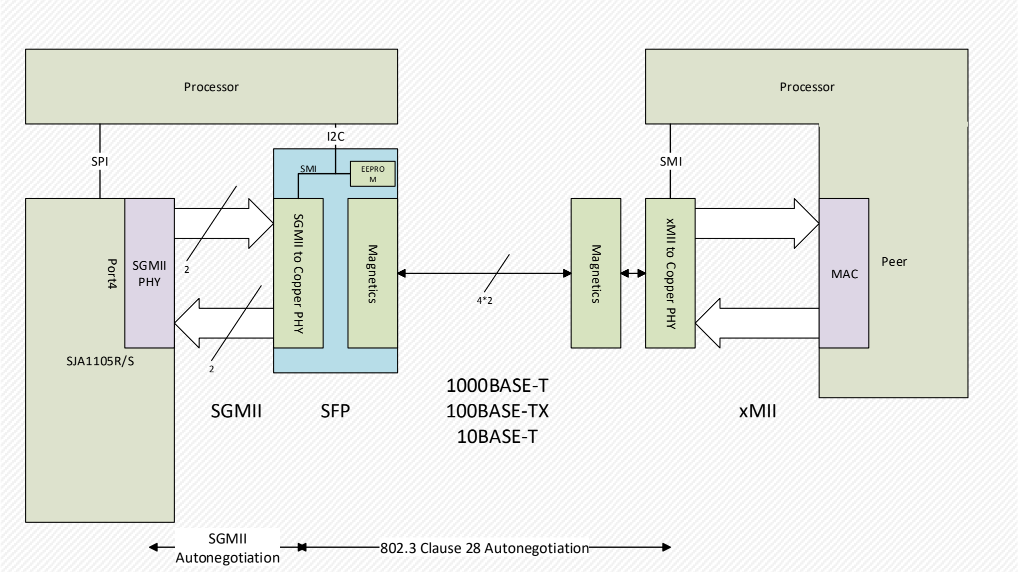


Fig 37. **SGMII connected to modern SFP-Module for copper**

Using a SFP to add a 1000BASE-T capable copper interface via SGMII makes auto-negotiation on SGMII mandatory, because IEEE 802.3ab (1000BASE-T standard) requires auto-negotiation. There are two different auto-negotiation processes involved in this scenario: one for the copper link parameters using IEEE 802.3-2012 Clause 28 (fast link pulses), and another one between the MAC and the SGMII-2-Copper PHY in the SFP module.

To the SGMII PHY it seems like it is driving a fiber transceiver or a direct SGMII connection, and auto-negotiation acknowledgements are coming from the link partner on the copper line, while they are actually coming from the SFP's PHY.

Advertisements from the SGMII-PHY are received by the SFP-PHY, translated to copper advertisements, which are negotiated with the remote link partner. Negotiation acknowledges to the SGMII-PHY are delayed until copper link parameters are finalized. This is usually done without user interaction. Please note, that some SFP vendors require auto-negotiation on the SGMII link according to 1000BASE-X, some require auto-negotiation with SGMII standards, and some allow configuring the standard in the SFP-PHY via the I2C interface.

6.2.3.3 PHY-MAC Roles

If auto-negotiation is used, PHY- and MAC-roles must be assigned manually to the link partners. For the SJA1105RS this is done with the PHY_MODE bit in AUTONEG_CONTROL.

In SGMII Auto-Negotiation the capabilities are transferred unidirectionally from SGMII-PHY to SGMII-MAC.

The SGMII PHY in SFP modules is typically configured in PHY mode to propagate the result of the Auto-Negotiated Copper capabilities to the MAC.

6.2.3.4 1000Base-X Clause 37 Auto-Negotiation

Not supported

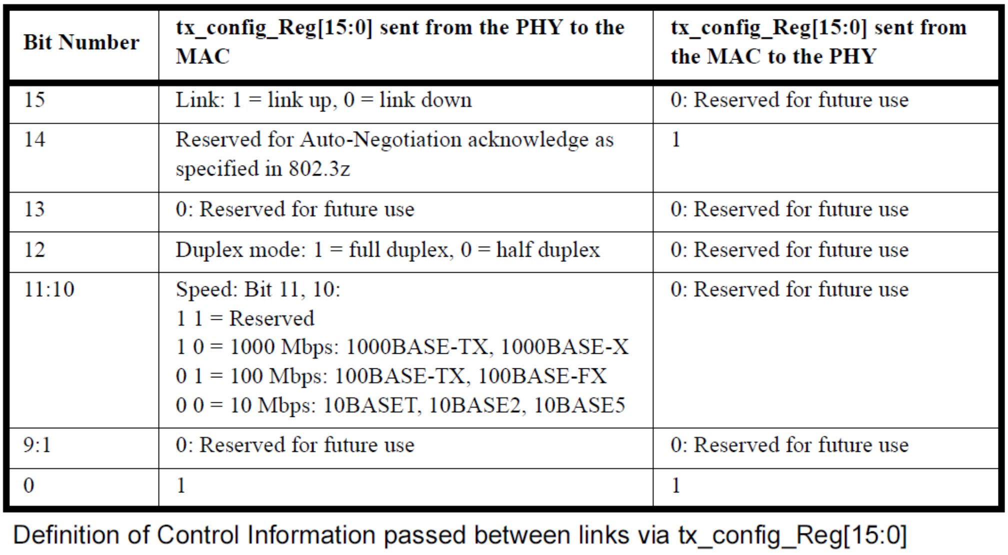
6.2.3.5 Capabilities

SGMII Auto-Negotiation can determine the following capabilities:

- Speed (10, 100 and 1000 Mbps)
- Duplex setting
- Link status

Note that the SJA1105RS's internal MACs only supports *full duplex* operation. The SGMII PHY can be configured for the half-duplex operation which is unsupported by the switch fabric and results in undefined system behavior.

The following table lists the 16bit capability as defined by the SGMII specification.



Bit Number	tx_config_Reg[15:0] sent from the PHY to the MAC	tx_config_Reg[15:0] sent from the MAC to the PHY
15	Link: 1 = link up, 0 = link down	0: Reserved for future use
14	Reserved for Auto-Negotiation acknowledge as specified in 802.3z	1
13	0: Reserved for future use	0: Reserved for future use
12	Duplex mode: 1 = full duplex, 0 = half duplex	0: Reserved for future use
11:10	Speed: Bit 11, 10: 1 1 = Reserved 1 0 = 1000 Mbps: 1000BASE-TX, 1000BASE-X 0 1 = 100 Mbps: 100BASE-TX, 100BASE-FX 0 0 = 10 Mbps: 10BASET, 10BASE2, 10BASE5	0: Reserved for future use
9:1	0: Reserved for future use	0: Reserved for future use
0	1	1

Definition of Control Information passed between links via tx_config_Reg[15:0]

Fig 38. **Config_Reg SGMII**

Depending on the PHY_MODE_CONTROL bit (DIGITAL_CONTROL_1 register) different sources for the fields in Config_Reg are used as described in the following table.

Table 34. **Advertised Capabilities**

PHY_MODE_CONTROL	Config_Reg	Source of data
0	Link (Bit 15)	SGMII_LINK See register AUTONEG_CONTROL (1f8001h)
	Duplex (Bit 12)	FULL_DUPLEX, See register AUTONEG_ADV (1f0004h)
	Speed (Bit 11:10)	SPEED_SELECT See register BASIC_CONTROL
1	Link (Bit 15)	1 if xMII Mode = SGMII (see xMII Mode Parameter Table)
	Duplex (Bit 12)	1
	Speed (Bit 11:10)	Speed as set in the MAC Configuration Table

6.2.4 Loopback mode

Loopback mode is a feature of the built-in SGMII PHY.

The xMII ports (Port0..3 or Port4 on the SJA1105PQ) do not feature any loopback mode.

For the SGMII PHY there are two loopback mode bits in the configuration registers:

- LOOPBACK in BASIC_CONTROL (1f0000h) for feeding TX traffic from the MAC back to RX

- REMOTE_LOOPBACK in DIGITAL_CONTROL_1 (1f8000h) for feeding RX traffic from the peer back to TX

Fig 39 shows the meaning of these modes: The LOOPBACK bit controls switch (2), and the REMOTE_LOOPBACK bit controls switch (1). Switching to loopback mode is exclusive, so normal traffic is no longer forwarded. So if LOOPBACK is enabled, received traffic from the SGMII lines is not forwarded to the MAC.

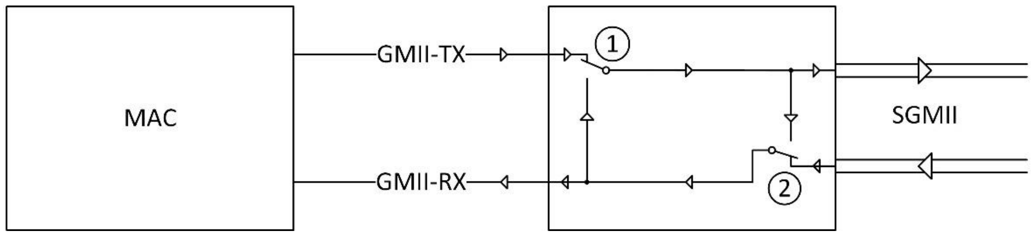


Fig 39. **LOOPBACK configuration options for the SGMII port**

6.2.5 SGMII Level Setting

For SGMII pre-emphasis and de-emphasis are not required and need to be disabled (TX_BOOST, RX_EQ_VAL).

The TX output amplitude can be changed via two distinct parameters:

- LEVEL: TX_LVL parameter in LEVEL_CONTROL register
- ATTENUATE: TX_ATTN parameter in TX_ATTN_CONTROL register.

TX_LVL has a range from 0h-1Fh. 0 being the minimum amplitude, 1Fh being the maximum amplitude. TX_ATTN has a range from 0h-7h. 0h resulting in the maximum amplitude, 7h in the minimum amplitude.

The SGMII specification specifies $|V_{od}|$ to be between 150 mV and 400 mV. This results in 300 mVpp and 800mVpp peak-to-peak voltage.

6.2.5.1 SGMII output level calculation

The actual SGMII differential peak-to-peak output amplitude is given by:

$$Vpp = attn(TX\_ATTN) * boost(TX\_BOOST) * 1.24 * \frac{\left(48 + \frac{level(TX\_LVL)}{2}\right)}{63.5}\ [Vpp]$$

For SGMII, boost is disabled, so boost(TX_BOOST) = 1.
The effective attenuation factor is shown in Table 35:

Table 35. SGMII attenuation coding

TX_ATTN value	Factor (fraction)	Factor (decimal)
b000	16/16	1.0000
b001	14/16	0.8750
b010	12/16	0.7500
b011	10/16	0.6250
b100	9/16	0.5625

TX_ATTN value	Factor (fraction)	Factor (decimal)
b101	8/16	0.5000
b11x	reserved	reserved

6.2.5.2 Maximum Output amplitude

The maximum output amplitude is set by: TX_LVL = 1Fh, TX_ATTN = 0h.

This results in an amplitude of 1055mV. VDDA_SGMII current consumption is approximately 15.7mA for this setting.

This amplitude value is from qualification under lab conditions. The theoretical amplitude according to the formulas above is 950mV.

Note: Setting the maximum Tx differential amplitude to greater 1Vpp results in overdrive (applying 1.2V) to the thin-gate output transistors. This is not recommended.

6.2.5.3 Minimum Output Amplitude

Minimum output amplitude is set by: TX_LVL = 0h, TX_ATTN = 7h.

This results in a measured amplitude of about 406mV. VDDA_SGMII current consumption is 13.5mA. These measurements are given for orientation only.

6.2.6 Recommended Values

The slew rate of the differential signal is controlled through the TX_EDGE_CONTROL register (address 0x1f8033). It is recommended to leave this value to default setting (fast edge rate).

Lowering the TX_LVL affects the current consumption. The difference from max level 1Fh to min level 0 is about 3.5mW in power consumption.

TX_ATTN also affects power consumption. The maximum is reached at settings 2h and 3h, the minimum is reached at settings 0h and 7h. There is a difference of 1.3mA (depending on TX_LVL setting), which is 3.25mW assuming VDDA_SGMII is 2.5V.

Lowering the amplitude lowers the common mode noise which helps reducing EMC radiation. Therefore, it is not recommended to set the output level to the maximum. Instead an amplitude of 650mVpp is recommended.

Common mode level and thus EMC radiation marginally drops when TX_ATTN is increased. As an alternative to the recommended settings given in the table, the following settings for TX_LVL and TX_ATTN can be used, but current consumption is not optimal for these settings.

TX_LVL = 1Fh, TX_ATTN = 0x4, Vod = 657mVpp, 16.8 mA

TX_LVL = 10h, TX_ATTN = 0x2, Vod = 683mVpp, 15.6 mA

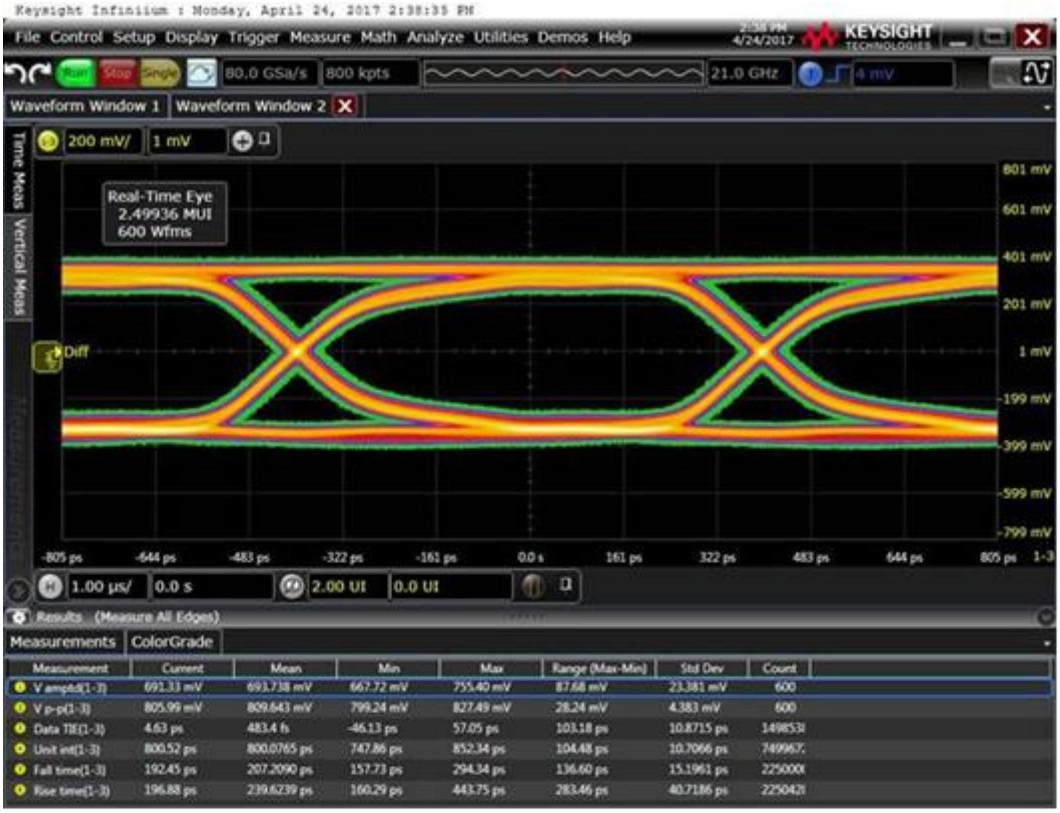
The following Table 36 shows the recommended settings. This results in 13.8mA current consumption for VDDA_SGMII.

Table 36. SGMII Level Settings

Register	Address	Field	Value	Description
TX_ATTN_CONTROL	0x1f8031	TX_ATTN	1h	~650mVpp TX signal
TX_BOOST_CONTROL	0x1f8030	TX_BOOST	0h	Disable boost

Register	Address	Field	Value	Description
LEVEL_CONTROL	0x1f8090	TX_LVL	0h	Lowest level
LEVEL_CONTROL	0x1f8090	LOS_LVL	09h	50mVpp threshold
RX_EQ_CONTROL	0x1f8051	RX_EQ_VAL	0h	Disable equalizer
TX_EDGERATE_CONTROL	0x1f8033	TX_EDGERATE	0h	Fast edge rate

The recommended settings will result in the following Eye diagram.



Measurement	Current	Mean	Min	Max	Range (Max-Min)	Std Dev	Count
V amptd(1-3)	691.33 mV	693.738 mV	667.72 mV	755.40 mV	87.68 mV	23.381 mV	600
V p-p(1-3)	805.99 mV	809.643 mV	799.24 mV	827.49 mV	28.24 mV	4.383 mV	600
Data TIE(1-3)	4.63 ps	483.4 fs	-46.13 ps	57.05 ps	103.18 ps	10.8715 ps	1498531
Unit int(1-3)	800.52 ps	800.0765 ps	747.86 ps	852.34 ps	104.48 ps	10.7066 ps	749967.
Fall time(1-3)	192.45 ps	207.2090 ps	157.73 ps	294.34 ps	136.60 ps	15.1961 ps	2250001
Rise time(1-3)	196.88 ps	239.6239 ps	160.29 ps	443.75 ps	283.46 ps	40.7186 ps	2250421

Fig 40. **SGMII Eye Diagram – Recommended Settings**

6.3 Low power sleep operation

As mentioned in ch. 3.2 the SJA1105PQRS allows to directly connect to a microcontroller. There are scenarios, where one side needs to go into low power sleep, while the other side must work normally. There are pitfalls regarding parasitic power paths, ability to wake up or precautions to not affect the operations of the non-sleeping side.

6.3.1 Sleeping microcontroller

In case the connected µC goes to sleep (“Sleeping Peer”) the switch’s port must be disabled temporarily, so no traffic is sent to the peer. Furthermore, the xMII pins to the peer must output a LOW level, so the µC is not accidentally powered via the data lines.

Undriven lines from the µC to the switch must be pulled to GND via weak pull downs, e.g. the configurable internal pull downs. They are not needed in normal operation.

Disabling a port temporarily is non-trivial, because the switch has no explicit port enable bit. Though disabling a port is possible via xMII_MODE[0] = 11b, this setting can only be done once after reset in the static configuration. Downloading a new static configuration stream means to reset the switch, which also affects other communication over the switch, and this is not acceptable.

Shutting off VDD_IO_MIIx is also not an option, because there is no implicit Power-on-Reset mechanism for the port section of the HW, so the behavior after re-applying power on VDD_IO_MIIx is not defined.

To support sleep mode of a µC connected to the switch, use the following steps:

1. Disable EGRESS (Bit 32 of the Mac Configuration Table). This inhibits new packets being transferred to the port’s egress queues. This is not only relevant for broadcasts and multicasts, but also for unicasts as long as the address of the µC is still contained in the L2 Address Lookup Table. The only traffic still queued is management traffic, which is under control of the management host. As the management host can be considered as the instance, which is also in control of the process for going to sleep, there are other ways to inhibit management traffic in the first place.
2. Disable INGRESS. After that, no traffic received on this port is forwarded to the switch fabric.
3. Wait for the egress queues and ingress queues being drained. Wait time depends on the fill level of the queues.
4. Disable the xMII clock(s) for this port. Depending on the mode in which the port is operated, this may involve switching off the clock generated by the Sleeping Peer. For clocks generated in the switch, do the following register settings:

Table 37. Disable switch’s output clock to support Sleeping Peer

Port Mode	How to disable switch output clock
RGMII	RGMII_TX_CLK_x[PD] = 1
RMII (PHY-Mode)	RMII_REF_CLK[PD] = 1

MII (PHY-Mode)	MII_TX_CLK_x[PD] = 1, MII_RX_CLK_x[PD] = 1,
	EXT_TX_CLK_x[PD] = 1, EXT_RX_CLK_x[PD] = 1

   Check the Sleeping Peer’s MAC configuration on how to switch off the xMII clocks on their side. This may be a non-issue, since as soon as the peer goes to sleep, it may terminate any output anyway.

5. Enable the weak pulldown resistors on xMII input pins. Since the Sleeping Peer no longer drives these lines, they may catch noise and cause increased power consumption on the switch’s input pins. The weak pulldown stabilizes the lines.

Putting the port into operation again when the Sleeping Peer awakes requires reverting the steps.

If RGMII is used, make sure to enable the RX TDL (if necessary) only after the RX_CLK from the peer is up and stable, and the TX TDL (if necessary) after TX_CLK has been re-enabled. Refer to ch. 6.1.13. for the details.

6.3.2 Sleeping switch

In some other scenarios the switch may need to sleep while the connected µC continues to be powered up.

The SJA1105PQRS does not support any low power mode, so power to the switch has to be cut by some external load switch or a PMIC. All power rails supplying the switch should be switched off.

Again, there is the problem of parasitic powering via the xMII lines. This has to be solved on the far side of the port links, e.g. on the µC. This includes the TX clock generated by the µC. If a port is connected to a PHY, this PHY should be sent to sleep as well or powered down.

The SJA1105PQRS does not create the 25MHz on CLK_OUT when switched off. Therefore, devices depending on that must either also be sent to sleep, or must use a different clock source.

The switch's reset line is supposed to be under control of the µC or a power supervisor, so the switch can come up after sleep just like during normal power-up.

Since the switch loses any memory during power-off, statistic data should be saved before powerdown, and the static and dynamic configuration must be written again to the switch after wakeup.

7. Switch Core Configuration

7.1 Static L2 Configuration Entries (TCAM)

Static L2 configuration entries are VLAN/MAC-to-port mappings which are configured by the user and not dynamically learned by the switch. These entries can be programmed through the static configuration or programmed through the programming interface at runtime.

The L2 Address Lookup table is used to make forwarding decisions during packet handling based on stored unicast as well as multicast forwarding rules. Fig 41 shows the L2 Lookup Table operation.

The L2 Address Lookup logic operates only on a single (VLANID, MAC)-pair. In case of double tagged frame (Service-Tag/S-Tag with type 88a8h and Consumer-Tag/C-Tag with 8100h), the outer VLANID (S-Tag) is always used for searches in the TCAM. In case of a single outer tagged (C-tag) or single inner tagged (S-tag) frame there is only one VLANID.

The flag IOTAG is used to indicate if the VLANID belongs to an inner (C-tag) or outer (S-tag) tag. If the frame is a double tagged frame (S-Tag, C-tag) or a single outer tagged (S-Tag), IOTAG will be set to 0. For frames with are untagged or have a single inner tag (C-tag), IOTAG will be set to 1.

For each frame, the TCAM is searched with the following key: IOTAG & VLANID & MACADDR.

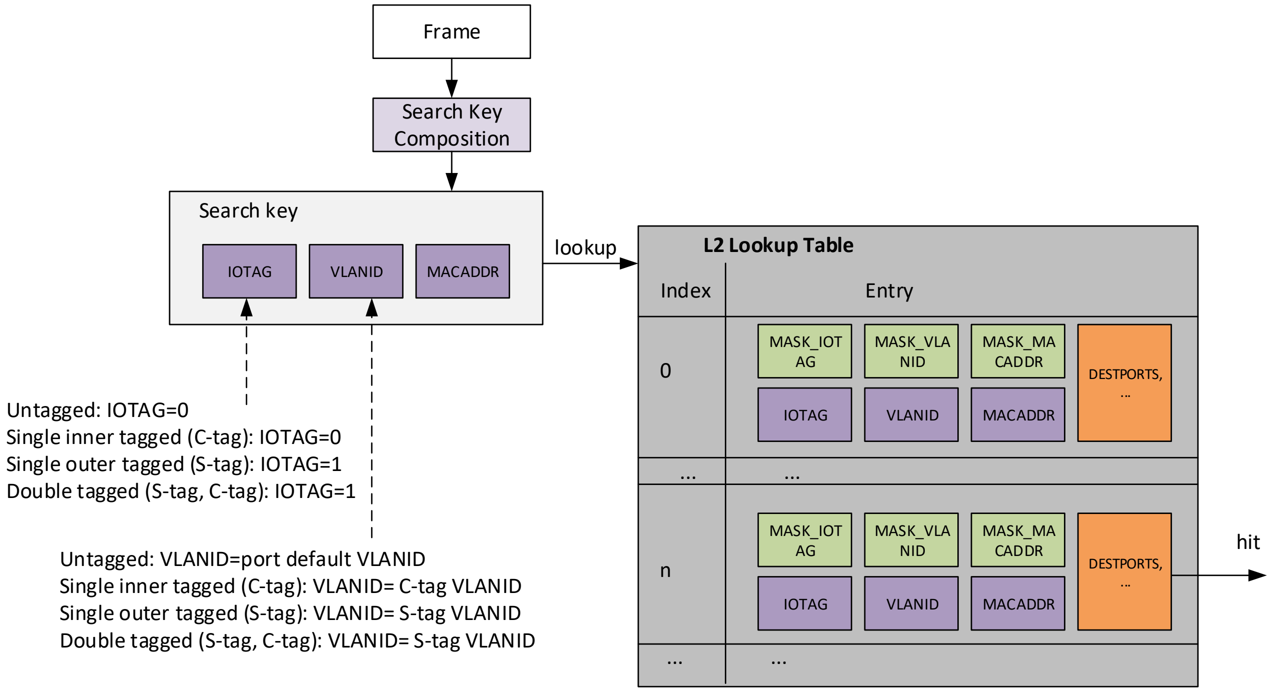

Frame
Search Key Composition
Search key
IOTAG
VLANID
MACADDR
lookup
L2 Lookup Table
Index
Entry
0
MASK_IOTAG
MASK_VLANID
MASK_MACADDR
DESTPORTS, ...
IOTAG
VLANID
MACADDR
...
...
n
hit
...
...
Untagged: IOTAG=0
Single inner tagged (C-tag): IOTAG=0
Single outer tagged (S-tag): IOTAG=1
Double tagged (S-tag, C-tag): IOTAG=1
Untagged: VLANID=port default VLANID
Single inner tagged (C-tag): VLANID= C-tag VLANID
Single outer tagged (S-tag): VLANID= S-tag VLANID
Double tagged (S-tag, C-tag): VLANID= S-tag VLANID

Fig 41. **L2 Lookup Table operation**

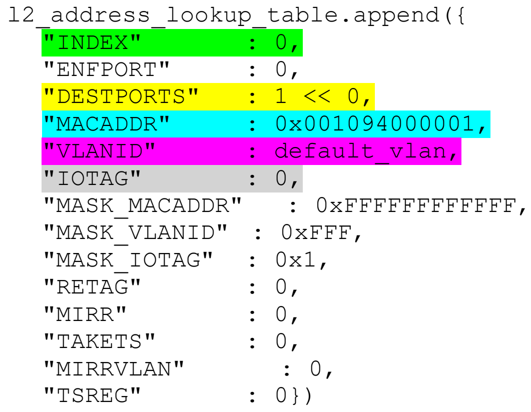
The following shows an example from the SJA1105PQRS tools package, provided alongside with the documentation.

The critical parameters for a simple entry are INDEX, DESTPORTS, MACADDR and IOTAG/VLANID[6]. Index selects to which of the 1024 available slots, the entry will be programmed to. DESTPORTS selects the ports that the frame is forwarded to. MACADDR and VLANID indicate the MAC address and VID to which the entry applies. IOTAG selects if the entry matches against an outer tagged / double tagged frame (IOTAG = 1) or the entry matches against a single tagged / untagged frame (IOTAG=0). For TCAM search, only the most outer tag is used. For a double tagged frame only the outer tag's VID is used (S-tag) for TCAM searching. For a single outer tag, the outer tag's VID is used for TCAM searching. For single tag and untagged frames the C-tag's or default VID is used.

It is not possible to craft a rule in which the inner tag of a double tagged frame is used for forwarding decisions.

Note that if multiple entries match (MASKs are used) the entry with the lowest index is used.

[6] Assuming shared learning is disabled (SHARED_LEARN is deasserted).



```
l2_address_lookup_table.append({
   "INDEX"          : 0,
   "ENFPORT"        : 0,
   "DESTPORTS"      : 1 << 0,
   "MACADDR"        : 0x001094000001,
   "VLANID"         : default_vlan,
   "IOTAG"          : 0,
   "MASK_MACADDR"    : 0xFFFFFFFFFFFF,
   "MASK_VLANID"  : 0xFFF,
   "MASK_IOTAG"   : 0x1,
   "RETAG"          : 0,
   "MIRR"           : 0,
   "TAKETS"         : 0,
   "MIRRVLAN"         : 0,
   "TSREG"          : 0})
```

Fig 42. **Static L2 Entry**

7.1.1 IVL and SVL

Independent Vlan Learning (IVL) and Shared Vlan Learning (SVL) can be implemented with the SHARED_LEARN bit in L2 Lookup Parameters Table: with SHARED_LEARN=1 new MAC addresses are entered with MASK_VLANID = 0, which means that the VLANID is not considered during L2 lookups.

7.1.2 Mirroring

The MIRR setting of the TCAM entry is used to enable flow mirroring.

If the **destination or source** MAC address of a received frame produces a match in the TCAM and this flag is found asserted in any (or both) of the matched TCAM entries, the frame is mirrored to MIRR_PORT (if MIRR_PORT contains a valid port number).

If the flag is found asserted for a *source* MAC address match, then the mirrored frame is handled according to the ingress mirroring rules (it is tagged with the INGMIRRVID associated with the port on which the frame is received).

If the flag is found asserted for a *destination* MAC address match, then the mirrored frame is handled according to the egress mirroring rules (it is tagged with the EGRMIRRVID from the General Parameters Table). If the flag is found asserted for both the destination and source MAC addresses, then the flag MIRRCIE resolves if the frame is handled as an ingress or an egress mirrored frame. See also the configuration example in ch. 7.6.1.

7.1.3 Retagging

The RETAG flag in the TCAM controls flow-based retagging. Retagging is able to change the VLAN of a frame (VLAN hopping).

If a destination or source MAC address of a received frame produces a match in the TCAM and this flag is found asserted in any of the matched TCAM entries, then the frame is retagged using the MIRRVLAN as the new VLAN ID.

Retagging means creating copies of tagged Ethernet frames, which will receive a new VLAN ID and will then be routed as if they were received on the original source port. If the flag is found asserted for both the destination and source MAC addresses, then the flag from the source MAC address match has precedence and the frame is retagged using the MIRRVLAN VLAN ID from the TCAM entry associated with the source mac address.

In case the original frame is to be dropped and only the retagged frame is to be forwarded, DESTPORTS should be set to zero. If DESTPORTS is not set to zero, two frames will be transmitted: the original frame and the retagged frame.

7.1.4 Trapping

The TAKETS flag controls TCAM-based trapping (automatic forwarding to the HOST port) and timestamping behavior.

The TCAM can be used for two purposes: first, to filter frames, taking an ingress timestamp and sending them both to the HOST port. Second, to inject frames from the HOST and send them to one or multiple ports, capturing a timestamp.

It must be noted that the MAC_FLT mechanism in the General Configuration Table offers a similar feature. The MAC_FLT is sitting directly at the ingress stage, very early in the pipeline flow. The TCAM is located after the ingress filter, policers and other protection mechanisms. Frames that are dropped and do not cause a TCAM search operation will not be trapped or filtered.

If a destination MAC address received on the Host Port produces a match in the TCAM memory, and both TAKETS and LOCKEDS flags are asserted, then an egress timestamp is captured for each frame transmitted to a destination port as defined by DESTPORTS. The egress timestamp is stored in one of the two per-port egress timestamp registers. .i.e. PTPEGR_TS = 2*PORT+TSREG, where PORT is the destination port for which a timestamp is captured.

If a received frame produces a match in the TCAM memory and this flag and LOCKEDS are both found asserted and the frame is received on any port other than Host Port or Cascaded Port, the respective frame is forwarded to the Host Port followed by the Meta-data follow-up frame that contains the ingress timestamp of the received frame.

7.2 Dynamic Reconfiguration

7.2.1 Reading a L2 Lookup Table Entry

Reading an entry of the L2 Address Lookup table can be done by this:

- Write a dummy entry to the “L2 Address Lookup table reconfiguration register 1 to 5”, with all 0, except the key to search for, that is e.g. the index, or the MAC address.
- Write to the “L2 Address Lookup table reconfiguration register 0” with VALID=1, RDWRSET=read, HOSTCMD=read
- The requested entry can now be read from the “L2 Address Lookup table reconfiguration register 1 to 5”.

Please note, that the “L2 Address Lookup table reconfiguration registers” exactly map to the L2 Address Lookup table in Table 16 of the UM. The first 6 unused bits in Table 16 are the same 6 reserved bits in the reconfiguration register 1 at address 24h.

```
def dump_single_l2_lut(idx):
  entry = [0, 0, 0, 0, 0]
  platform.spi_write(0x24, (idx & 0x3ff) << 6)
  # HOSTCMD = read; VALID = 1, RDWRSET = read
  platform.spi_write(0x29, (2 << 23) | ( 1 << 31) | ( 0 << 30))
  validEntry = platform.spi_read(0x29)
  if ((validEntry & (1 << 27)) == (1 << 27)):
    readData = [platform.spi_read(0x24+i) for I in range(5)]

    LOCKEDS     = ((validEntry  >> 28) & 1)

    INDEX       =  ((readData[0] >> ( 6 % 32)) & 0x3FF)
    ENFPORT     =  ((readData[0] >> (16 % 32)) &     1)
    DESTPORT    =  ((readData[0] >> (17 % 32)) & 0x01f)
    MACADDR     = (((readData[0] >> 22) & 0x000003ff) <<  0) + \
                  (((readData[1] >>  0) & 0xffffffff) << 10) + \
                  (((readData[2] >>  0) & 0x0000003f) << 42)
    VLANID      =  ((readData[2] >> (70 % 32)) & 0xfff)
    IOTAG       =  ((readData[2] >> (82 % 32)) &    1)


    entry = (str(INDEX), '{0:05b}'.format(DESTPORT), mac_to_str(MACADDR),
str(VLANID), str(ENFPORT), str(IOTAG), str(LOCKEDS))

    print(entry)
```

Fig 43. **Output a L2 address lookup table entry**

7.2.2 Writing a L2 Lookup Table Entry

Writing to the L2 Address Lookup table is possible by using a certain table index. If there is already an entry it will be overwritten.

The following steps are necessary:

- Check the “L2 Address Lookup table reconfiguration register 0” until the VALID bit is 0.
- Write the whole entry you intend to write to the “L2 Address Lookup table reconfiguration register 1 to 5”. The INDEX part selects the entry, into which the new content will end up. VALIDENT determines if the new content will be valid or invalid. Invalid entries will be used for address learning.
- Write to the “L2 Address Lookup table reconfiguration register 0” with VALID=1, RDWRSET=write (1), HOSTCMD=write (b011).
- Poll the “L2 Address Lookup table reconfiguration register 0” until the VALID bit has been cleared and the operation is complete.

7.3 Timestamping and AVB

The timestamping infrastructure is closely related to the frame filtering facility. In order to understand how timestamps work, filtering or trapping must be understood, see the following chapters.

7.3.1 Frame Trapping

Frame trapping or frame filtering is the mechanism of identifying special frames (gPTP or 802.1X EAP). Such frames are typically identified by a special multicast address range. Two mechanisms can be used to trap frames.

© NXP Semiconductors N.V. 2020. All rights reserved.

1. **The ingress MAC filter**
   The General Parameters Table allows to configure two such filters: MAC_FLTRES[0], MAC_FLT[0] and MAC_FLTRES[1], MAC_FLT[1].
   If parts of the destination address of an ingressing frame match the MAC filter, it will be forwarded to the host port. Additionally, if the SEND_META bit associated to the firing MAC filter is set, an ingress timestamp will be generated and sent in a following META frame (see user manual).
   Note: Even if the port is disabled for ingress (INGRESS bit is cleared in MAC Configuration table), a match with this filter will forward the frame to the host port.
2. **TCAM matching**
   If a frame matches a TCAM (L2 Lookup Table) entry and the entry’s TAKETS bit is set, the frame will be forwarded to the host port. Additionally, also an ingress or egress timestamp is generated. Consult the user manual - section Flow Trapping - on how to configure the TCAM for trapping.

Both mechanisms can be used independently from each other to handle trapping and timestamping.

7.3.1.1 Disabling trapping

A MGMT filter is disabled by setting MAC_FLT to all zeros and MAC_FLTRES to all ones.

```
"MAC_FLTRES[0]" : 0xFFFFFFFFFFFF,
"MAC_FLT[0]"    : 0x000000000000,
```

If no TCAM rules with the TAKETS bit are set, no frames are trapped to the host.

7.3.2 Sending management frames

Sending a special frame from the host via a particular port can be seen as the opposite of trapping. Management protocols, like RSTP rely on this mechanism.

A management frame from the host has normally a special destination address which is not included in the L2 address lookup table. So the switch does not know what to do with that frame. There may be a trapping rule for that destination address – but that is also not what is needed, because then the frame would be sent back to the host. Furthermore: it might be necessary to send the frame to a port, which is blocked for standard traffic, e.g. because it is a redundant link temporarily disabled. Management traffic nevertheless must be sent nevertheless.

To handle these problems, there are 4 additional entries for the L2 address lookup table. These entries have the same syntax as standard L2 address lookup table entries, but they are invalidated once they are used. There is no way to use the static config stream to set these entries, as they are by nature dynamic entries. The mechanism to write them is the same, as standard L2 address lookup table entries, except MGMTROUTE must be set in L2 address lookup table reconfiguration register.

So the basic algorithm on the host for sending out a management frame is:

1. Create the frame you want to send
2. Find a free entry in the L2 management lookup table (there may be multiple management frame send processes active)

3. Write this entry (index) with the destination address and the destination port(s) of the frame to be sent. As in the L2 address lookup table the destination port entry is a bit-set to allow egressing the matching frame via multiple ports. The mechanism to write to the L2 management lookup table is the same as for the L2 address lookup table entry.
4. Send the frame towards the switch

As soon as the frame has been egressed, the entry in the lookup table is invalidated and can be reused for other management frames.

7.3.3 1-step and 2-step SYNC frames

In order to implement synchronization protocols such as gPTP, timestamps can be captured at ingress and egress events. For 1-step PTP SYNC frames the switch will automatically add the residence time to the frame. For other protocols, such handling has to be done in software on the host processor using the timestamps provided by the switch HW.

The timestamping infrastructure is closely related to the frame filtering facility. In order to understand how timestamps work, filtering or trapping must be understood, see the following chapters.

The SJA1105PQRS updates the Transparent Clock field for the 1588 frames regardless if the frame is trapped to the host (by MGMT filters) or not. The Transparent Clock field is updated as follow:

1. If IGNORE2STF flag is set, the switch HW updates the Transparent Clock for Sync, Delay_Req, Pdelay_Req, Pdelay_Resp frames, regardless if the twoStepFlag in the frames is set or not.
2. If IGNORE2STF flag is not set, the switch HW updates the Transparent Clock for Sync and Delay_Req frames only if twoStepFlag in the frames is not set (only for 1-step frames).

See chapter 7.6.6 for configuration examples.

7.3.4 Reconstruction for META frames

The SJA1105PQRS has a 64 bit (8 byte) internal free running counter which is used to capture timestamps on egress and ingress events. Depending on the CORRCLK4TS bit, this is either

- PTPTSCLK (8ns clock intervals derived from the system clock) or
- PTPCLKVAL, which is a rate corrected counter, whose speed is controlled PTPCLKRATE. For more info refer to ch. 0.

When the SJA1105PQRS is configured to capture an ingress timestamp, it produces a META frame which carries the lower 4 bytes of the timestamp information.

In order to process the timestamp and compute the residence time, a full 8 byte timestamp has to be reconstructed from the 4 byte. Do not directly use the 4 byte timestamp from the META frame, except for the full 64 bit timestamp reconstruction.

For this the free running counter is captured some time (but sufficiently fast) after the META frame has been generated. The upper 4 bytes of the free running counter and the 4 byte of the META frame must be stitched together.

Care must be taken to compensate for overrun effects.

The following pseudocode snippet shows the concept. A type-safe version can be found in the SJA1105PQRS driver package.

```
timeStampL = META frame timestamp;
timeStamp = 8 byte counter snapshot:
timeStampLMask = (1 << 32) – 1U;
timeStampLRecent = timeStamp & timeStampLMask;
if (timeStampLRecent < timeStampL)
  {
    /* Overrun */
    timeStamp = timestamp – (1 << 32);
  }
timeStamp &= ~timeStampLMask;  /* set lower bits to 0 */
timeStamp |= timeStampL & timeStampLMask;
```

Care must be taken to compensate for overrun effects.

7.3.5 Ingress Timestamp towards the Host Interface

In some applications, the host controller implements endpoint as well as bridge functionality. A typical application would be a gPTP bridge/endpoint combination. Here, the bridge part of the microcontroller firmware generates a Sync frame which needs to be forwarded to the endpoint part of the firmware, together with an ingress timestamp. For this a management route must be setup via the management lookup table and the host port must be reachable to itself via the REACH_PORT field in the L2 forwarding table. This will then enable the switch to trap frames which come from the host itself. The following figure shows an example for this:

a) A frame arrives at port 1, based on its destination MAC address it is decided that the frame needs to be trapped and forwarded to the host port.

b) The original frame and the META frame (green), which carries the ingress timestamp that belongs to port 1 arrive at the host port.

c) The bridge portion of the microcontroller firmware processes the frame and sets up a management route towards the all ports excluding port 1. This explicitly includes the host port. The frame is forwarded to all ports. The frame forwarded towards the host port is trapped.

d) A new META frame (orange) is generated and carries the time when the frame was received by the switch on the host port (port 0). Also the egress timestamp on port 0 can be read to compute the residence time from Ingress port 0 to Egress port 0.

The egress timestamp register of the host port records a 3byte timestamp when the SOF of the META frame generated for the trapped frame is seen by the reference plane for timestamping and hence a correction is required for the recorded timestamp.

$$EgressTimestampCorrection = \frac{(FrameSize[byte] + IFG[byte] + Preamble[byte]) * 8\ bit/byte}{speed[bit/s]}$$

$$EgressTimestampCorrection = \frac{(FrameSize + 20) * 8}{speed}$$

The register PTPEGR_TSn contains the time in units of 8ns, hence the correction for the actual register is

$$\mathrm{PTPEGR}_{\mathrm{TSn}_{\mathrm{corrected}}} = \mathrm{PTPEGR}_{\mathrm{TSn}} - \mathit{EgressTimestampCorrection} * 125000000$$

- *FrameSize is the L2 size of the trapped frame in Bytes.*
- *IFG, Inter-frame Gap (12Bytes) The IFG can be extended beyond 12 bytes using the MAC configuration table.*
- *Preamble is the Ethernet Preamble (8Bytes).*
- *Speed is the link speed (bits per second).*
- *PTPEGR_TSn_corrected is the corrected timestamp register value.*
- *PTPEGR_TSn is the egress timestamp on host port n as read over SPI.*
- *EgressTimestampCorrection is the offset in seconds that need to be corrected.*

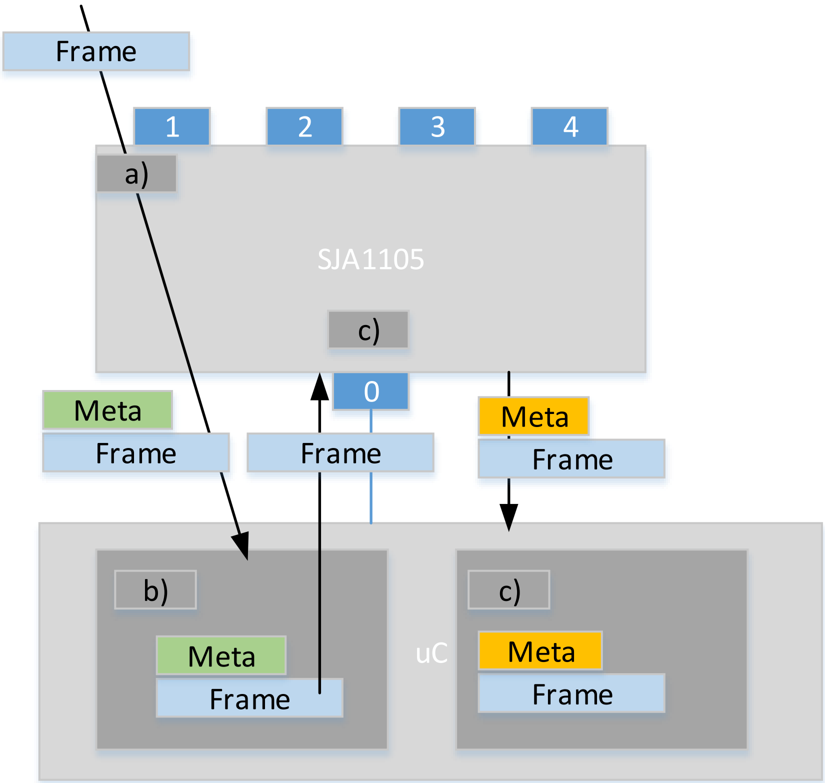


Fig 44. **Example for generating an Ingress Timestamp at the HOST port**

7.3.6 Credit Based Shaper Parameters

This section explains the Credit Based Shaper (CBS) configuration. Throughout the section, it is assumed that the configuration is set to L2CBS=0 (AVB Parameters Table). In this mode, the shapers are 802.1Qav compliant. The other mode is not discussed in this document.

The CBS is controlled using four parameters: sendSlope, idleSlope, hiCredit and loCredit. These parameters translate to the registers. The parameters idleSlope and sendSlope control the data rate of the traffic class. Additionally, the parameters hiCredit and loCredit can be used to saturate the credit level. If they are not required, they can be set to the minimum and maximum values.

The following figure shows an example how the data rate for AVB Class B is controlled. The shaper admits frames only if the credit level is higher or equal to zero. The credit level is replenished and consumed according to the following rules:

The credit is reset to zero if the associated queue is empty.

If the queue is not able to transmit (i.e. due to higher priority interference or because the credit level is negative) the credit is increased with a rate of idleSlope.

If frames of the queue are transmitted, the credit level is decreased at a rate of sendSlope.

In AVB, the idleSlope directly translates to the reserved L1 data rate. In this case an idleSlope of 1 Mbit/s will reserve 1 Mbit/s bandwidth. The sendSlope is portTransmitRate – idleSlope.

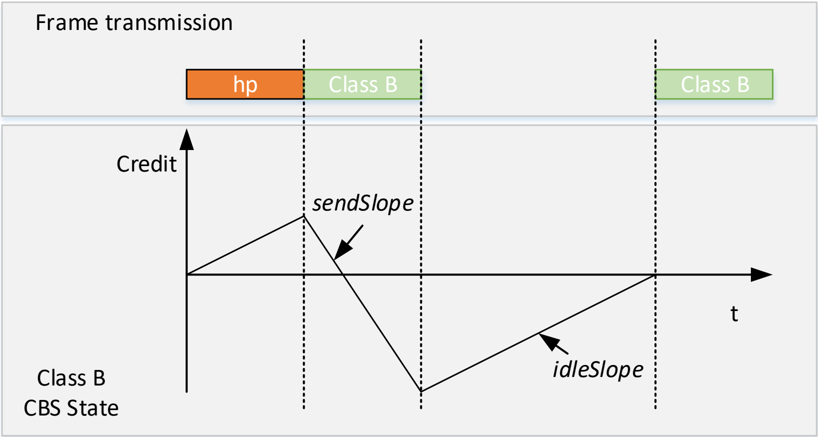


Fig 45. **Example CBS Operation**

The sendSlope and idleSlope are expected by the device in units of Byte/s.

As an example, consider an Ethernet AVB Frame of 80 bytes generated every 125 µs. This translates to a frame rate of 8000 Frames/s.

With added Ethernet overheads of 20 octets- *IFG (12 octets), Preamble/SFD (8 octets)*, the AVB Data Rate works out to 6.4 Mbit/s or 0.8 MB/s

$$actualBandwidth\left[\frac{bit}{s}\right] = \frac{20 \cdot 8\ bit + 640\ bit}{0.000125\ s} = 6400000\frac{bit}{s}$$

$$portTransmitRate = 100\frac{Mbit}{s} = 100000000\frac{bit}{s}$$

The following computation is performed according to the 802.1Qav specification:

$$sendSlope = portTransmitRate - actualBandwidth = 93600000\frac{bit}{s} = 11700000\frac{byte}{s}$$

$$idleSlope = actualBandwidth = 6400000\frac{bit}{s} = 800000\frac{byte}{s}$$

The parameters CREDIT_HI and CREDIT_LO are used to saturate the credit at an upper and a lower threshold. These parameters control the burst behavior of the shaped stream and are typically application dependent. A larger value of CREDIT_HI implies that if the shaper was stalled for a long time, the shaper can emit large bursts in order to catch up to the average rate. Similarly a small value of CREDIT_HI would limit the maximum burst size.

A small value of CREDIT_LO reduces the worst-case latency by allowing to send a frame earlier then it is normally dictated because the replenishment happens faster.

Note: IEEE 802.1Qav does not specify CREDIT_LO and CREDIT_HI. For nominal IEEE 802.1Qav operation both values should be set to the maximum allowed value of 0x3FFFFFFF.

7.3.7 Policing

Policing is used to enforce ingress rules regarding utilization of bandwidth, frame size, frame rate, buffer memory usage, etc., and to drop incoming frames violating any of those rules.

The following table lists the relevant fields in the L2 Policing Table block which needs to be configured for handling QoS. These are relevant for:

- Instantiating policing blocks
- Limiting ingress rate and burst size
- Limiting maximum ingress frame length
- Defining memory partition used for ingress buffering

The table entries correspond to a flow of traffic at a port based on the PCP value of tagged traffic or the default PCP associated with untagged traffic. The L2 policing table is structured in the following way (see Fig 46): the first 8 (i=0 to 7) entries of the table correspond to port 0, PCP=0 to 7. The next 8 entries (i=8 to 15) correspond to port 1, PCP=0 to 7 and so on. Broadcast traffic at a port is treated as a separate flow and there is one table entry per port (i=40 to 44 for Ports 0 to 4 respectively). If no corresponding entry for broadcast traffic is defined, then the default PCP for such traffic is treated as 0 and the policing block used for such traffic at a port corresponds to that of non-broadcast traffic with PCP=0.

Note that the policing unit implements a two-stage scheme. At first, an ingress frame is mapped to an entry in the policing table. In the second step the policing table determines the policer type and its parameters through the SHARINDX field.

For the first step, the port number and priority (PCP) value of the frame is used. A frame which does not carry a tag, will be prioritized with port default priority value (see MAC Configuration Table of the associated port).

The entry that is used is determined by:

IDX = port * 5 + PCP (for non-broadcast frames)

IDX= 40 + port (for broadcast frames)

The policer parameters SMAX, RATE, MAXLEN and PARTITION are determined in the second step, by evaluating the SHARINDX of the policer table entry at IDX:

If SHARIND is equal to IDX, the parameters are taken from this entry. If SHARIND is different, the parameters are taken from the entry at entry SHARIND. This allows to link multiple ports and prios to the same policer parameters. The concept is visualized in the following Figure.

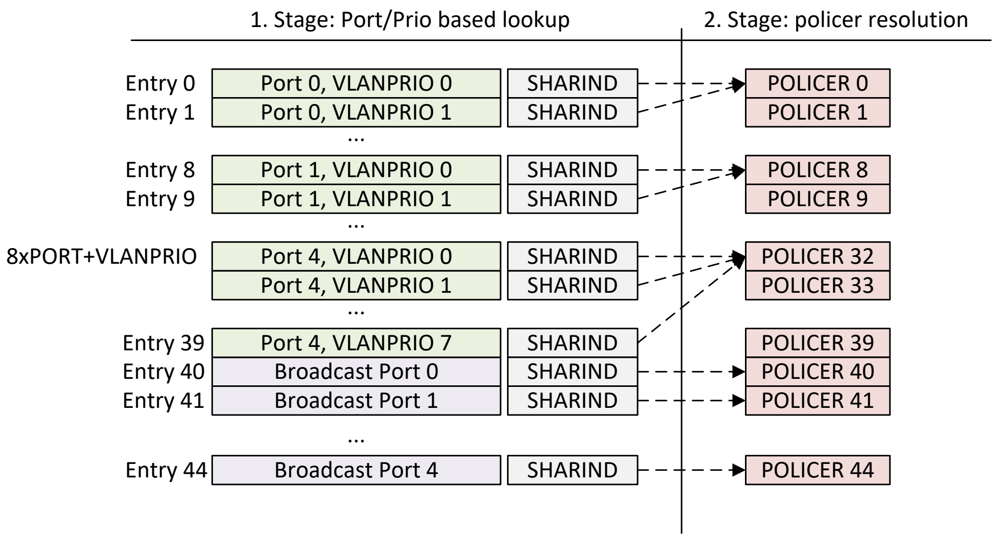


Fig 46. **Port / priority to policer mapping**

7.3.8 Priority Regeneration and Priority to Queue Mapping

The switch allows to remap the priority value (PCP) of ingress frames to a different PCP value to be used for transmission. This is typically referred to as priority regeneration.

The SJA1105PQRS uses the same mechanism to assign the frame to a certain priority queue at the egress stage.

Both settings are controlled through the L2 Forwarding Table (Fig 47). This table contains two sub-tables with different semantics, syntax and usage.

Similar to the policing table, the priority reassignment is implemented in two stages:

1. Per-port priority remapping, priority regeneration
2. Queue assignment

Each phase uses its own sub-table.

The first sub-table is used for per-port based remapping of the ingress priority values to egress priority values. It contains one entry per port (5 in total). For instance, the value of VLAN_PMAP in entry 0 defines the mapping of either a received or per-port assigned ingress priority value p_i to an egress priority value p_o for frames received on port 0 by assigning p_o = VLAN_PMAP[p_i]. This means that p_o will be used as the PCP (Priority Code Point) value on all egress ports forwarding the frame with a VLAN tag included (obtained by the TAG_PORT parameter in the VLAN configuration, see Table 12).

Furthermore, the first sub-table contains other per-port settings for broadcast domain, reachability and flooding domain.

The second sub-table is used for a per-egress priority-based mapping of logical priority values to physical priority queues of the different ports. It contains one entry for each priority (8 in total). For the previously obtained egress priority value p_o, the resulting mapping to priority queues on each port is obtained by assigning q_i = VLAN_PMAP[i], where q_i is the priority queue used for p_o on port i.

Both sub-tables are adjacent, and use a similar entry layout. The 2nd sub-table starts with index 5. Some elements in the 2nd sub-table entries are reserved.

For example, to map priority value p_o = 4 to priority queue 1 on port 0 and to priority queue 2 on port 3, the value of VLAN_PMAP for entry 5 + 4 = 9 must be set to 1 for index 0 and to 2 for index 3.

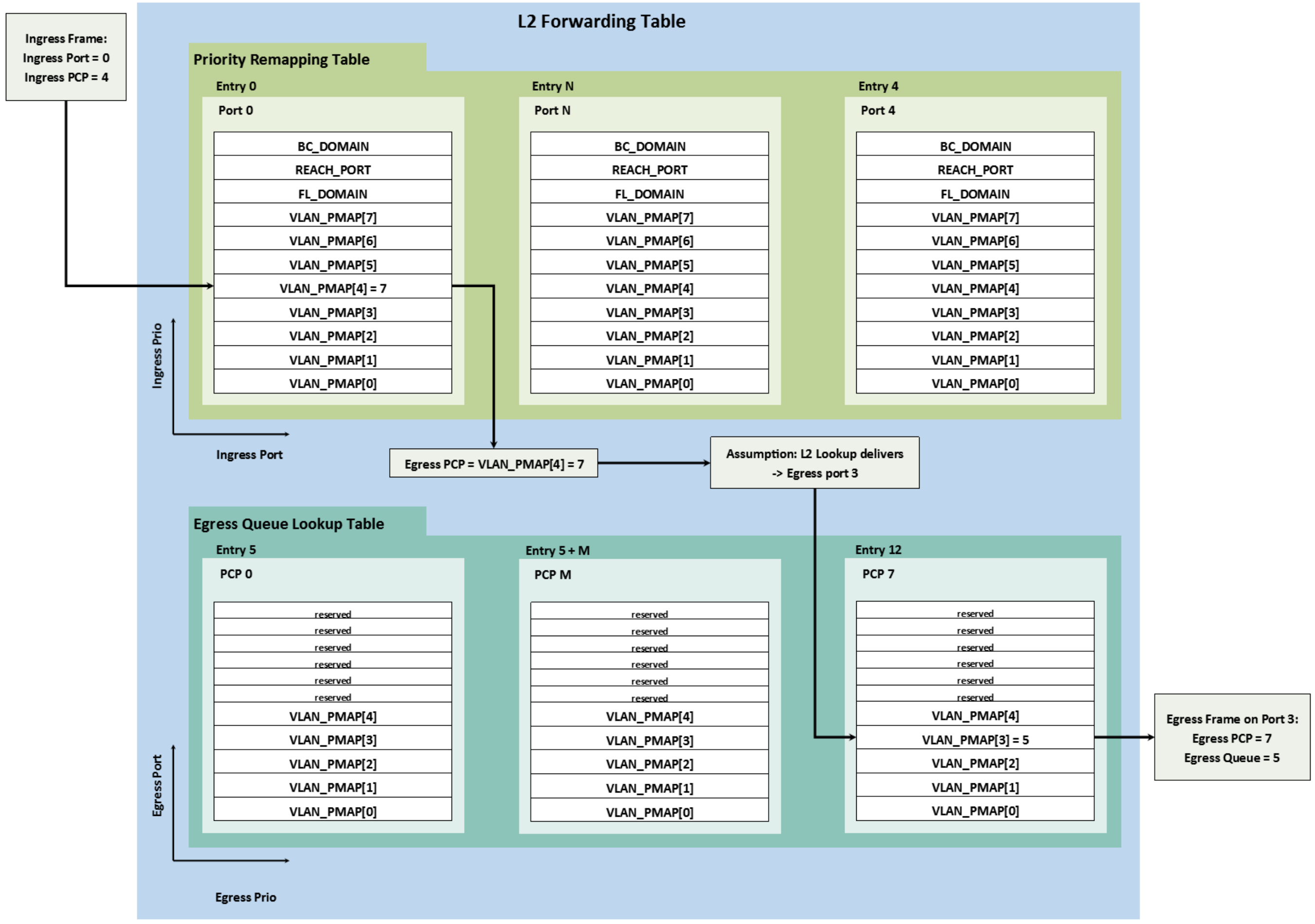


Fig 47. **Ingress to Egress PCP resolution and queue selection**

7.3.9 Timestamping in Cascaded Switch designs

For automatic residence time calculation, a simple difference operation between the full 64bit egress- and ingress timestamp is used. If ingress and egress takes place on different physical switches in a multi switch design, the timestamp clocks must run synchronously. See 7.6.7 on how this can be achieved.

7.4 TSN – Time Sensitive Networking

SJA1105Q and SJA1105S only

TSN extends IEEE 802.1Q Ethernet in general and AVB in particular to improve determinism in the network. The SJA1105PQRS supports the Time Aware Shaper and Per Stream Policing.

© NXP Semiconductors N.V. 2020. All rights reserved.

7.4.1 Time Aware Shaper (TAS) – 802.1Qbv

The TAS is a traffic shaping and transmission control algorithm, which can be applied on egress queues to achieve highly predictable traffic behavior. Combined with time synchronization methods, it allows very low latency data transmissions.

7.4.1.1 Introduction TAS

The TAS algorithm is defined in IEEE 802.1Qbv. It is an amendment to IEEE 802.1Q and extends the standard queuing and forwarding rules with a TDMA based egress transmission selection. In contrast to previous attempts to establish TDMA in automotive in-vehicle networking like FlexRay, the scheduling is not performed on message level but on traffic class level. This abstraction reduces the configuration complexity and thus enables an easy and incremental entrance into the benefits of TDMA methods.

TAS works by allowing or gating traffic transmission of egress queues based on a predefined schedule (see Fig 48). This schedule is defined in a so called “Gate control list” for each port. It is a sorted list defining the state of each gate at specific points in time. The state of a gate can be “open” or “closed”. A gate is associated with each separate egress queue.

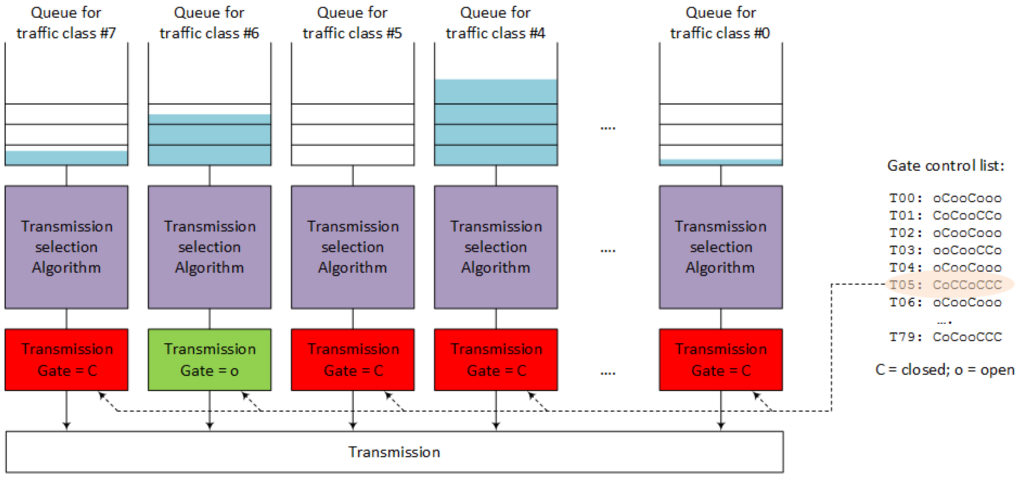

Queue for traffic class #7
Queue for traffic class #6
Queue for traffic class #5
Queue for traffic class #4
Queue for traffic class #0
....
Transmission selection Algorithm
Transmission selection Algorithm
Transmission selection Algorithm
Transmission selection Algorithm
Transmission selection Algorithm
Transmission Gate = C
Transmission Gate = o
Transmission Gate = C
Transmission Gate = C
Transmission Gate = C
Transmission
Gate control list:
T00: oCooCooo
T01: CoCooCCo
T02: oCooCooo
T03: ooCooCCo
T04: oCooCooo
T05: CoCCoCCC
T06: oCooCooo
....
T79: CoCooCCC
C = closed; o = open

Fig 48. **Transmission selection in 802.1Qbv using TAS**

The TAS can be used in two ways: Asynchronously for traffic shaping only (as an alternative to CBS) or among multiple time synchronized switches and endpoints. The second case requires multiple switches in the network to be synchronized using gPTP as in IEEE802.1AS and have their PTP clock synchronized to the grand master. This allows to coordinate the schedules of multiple network elements so that critical frames are guaranteed to be forwarded with minimal waiting time.

7.4.1.2 Static Configuration of TAS Schedule Table

The switch maintains the gate control lists in one central schedule table (see “Schedule table” in the user manual). For a TAS configuration, only three fields are relevant in one entry of the schedule table.

Table 38. Fields to be configured in an entry of the Schedule Table for TAS operation

Field	Description
DESTPORTS	Defines the ports (1 bit per each port) that the respective trigger event applies to. Bits at lower bit positions are assigned to ports with lower port numbers. Example: 01001b → Port 0 and 3 will have the gate status updated, other ports unchanged
RESMEDIA	Defines the status of the gate (0: open, 1: closed) for each queue. Bits at lower bit positions are assigned to queues with lower queue index. Bit 8 is an enable flag. Changes to the gates only apply if this flag is set. Example: 100111111b → Enable flag set, Queue 7 and 6 open, rest closed
DELTA	Time difference to previous entry in multiples of 200 ns (25 system clock cycles). It is not allowed to configure entries with DELTA=0. Thus, the range is 200 ns to 52.439 ms. Example: 5000 → 1 ms

An example configuration could thus look the following:

Table 39. Example configuration of Schedule Table

Delta	RESMEDIA	DESTPORTS	Explanation
5000	101111111b	11111b	After 1 ms, only allow Queue 7 to transmit on all ports
500	110111111b	11111b	After 1.1 ms, only allow Queue 6 to transmit on all ports
500	111000000b	11111b	After 1.2 ms, only allow Queue 0-5 to transmit on all ports As this is the last entry, the schedule will start again

The cycle time of the schedule is 1.2 ms (sum of all deltas). It will reserve 8.3 % of the bandwidth to Queue 7 and 6, respectively. The rest of the bandwidth is share by the rest of the traffic classes. The scenario is depicted as a function of time in Fig 49.

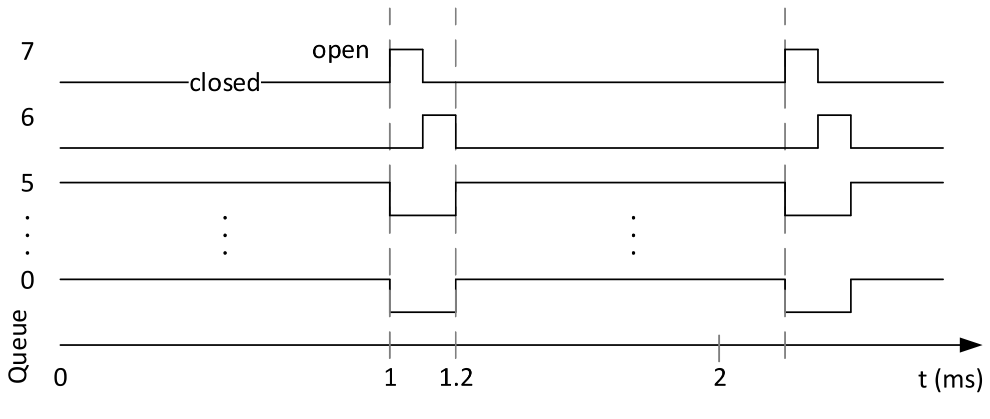


Fig 49. Gate states over time in the example configuration

This configuration is simplified, e.g. it does not define guard bands (times before change in gate configuration to avoid a transmission is still ongoing while the gate configuration is changed).

7.4.1.3 Static Configuration of TAS Parameters

On top of the schedule table itself, certain configuration tables (Schedule Entry Points, Schedule Parameters, Schedule Entry Points Parameters) must be configured, and a few more optional settings can be made. The following tables summarize the possible settings. In many cases, this is a fixed value for TSN. Fields not explicitly mentioned are 'don't care'.

Table 40. Fields to be configured in the Schedule Entry Points Parameters for TAS operation

Field	Value	Description
ACTSUBSCH	0	Needs to be set to 0 as there is only a single active sub-schedule in TSN
CLKSRC	1 or 3	Selects the internal clock to be used for the TAS 1: Standalone mode, using the PTPTSCLK 3: Host controlled clock (PTPCLK) Clock mode 1 is easier to use, as it requires no host interaction and can work just off the static configuration. However, it can work only of the free running clock, i.e. it is only applicable in scenarios where TAS is used for asynchronous traffic shaping. Clock mode 3 needs to be used if the TAS is to be synchronized to the gPTP grand master. In this case, the schedule needs to be manually started (see 7.4.1.4).

With a single sub-schedule used in TSN configuration, only the first field in the Schedule Parameters needs to be configured.

Table 41. Fields to be configured in the Schedule Parameters for TAS operation

Field	Value	Description
SUBSCHEIND[0]	0-1023	Needs to be set to the number of used entries in the Schedule Table minus 1. Example: The Schedule Table contains 10 entries → Set to 9

The Schedule Entry Points Table has to be configured with a single entry, containing values as specified below.

Table 42. Fields to be configured in an entry of the Schedule Entry Points Table for TAS operation

Field	Value	Description
ADDRESS	0	Must be 0 for TSN operation
DELTA	1 - 3FFFFh	Delayed start of the schedule in multiples of 200 ns. Can be used to offset the schedule with respect to other switches. In most use-cases, simply configuring 1, the minimum value, is sufficient
SUBSCHINDX	0	Must be 0 for TSN operation, only one sub-schedule used

7.4.1.4 Rate Ratio Correction and Synchronization

If the PTPSRC is used as a clock source, the rate correction is also applied to the scheduling engine. This guarantees that the schedule stays in sync with the global system time.

The schedule engine is internally clocked with the 125 MHz digital clock source derived from the crystal oscillator. If the PTPCLK is used as a clock source, a rate correction factor (PTPCLKRATE) is applied to speedup or slow down the scheduling engine. The rate is set by the software PTP/AVB stack by computing a correction factor.

The parameter PTPCLKCORP controls how the rate correction is applied to the scheduling engine.

In the ideal scenario when PTPCLKRATE is exactly one, one DELTA time is exactly 25 system clock cycles at 125 MHz. If PTPCLKRATE is larger or smaller, up to 5 clock cycles are added or subtracted from the nominal 25 cycles. Hence a DELTA is on average 200ns wall clock time, sometimes slightly larger/smaller depending on the PTPCLKRATE factor.

Additional cycles are added/removed every PTPCLKCORP time. To synchronize this with the schedule duration, it is recommended to set PTPCLKCORP to the cycle length.

7.4.1.5 Starting/Stopping TAS Scheduling

If CLKSRC 3 is selected in the Schedule Entry Points Parameters, the TAS schedule needs to be started at run-time.

For that, PTPSCHTM can be used to define the PTP clock value, at which the schedule should start. In addition, PTPSTRTSCH needs to be set together with VALID in PTP control register 0.

The simplest way of starting the schedule is this:

1. Set PTPCLKCORP to the scheduling duration. Each PTPCLKCORP period, the scheduler will incorporate the rate ratio correction into the schedule.
2. Set PTPSTRTSCH to a small value, e.g. 1 (equivalent of 8 ns)
3. Reset the PTP clock and set the flag to start the schedule (RESPTP + PTPSTRTSCH + VALID set at the same time)

A realistic use-case is to synchronize the schedule to a PTP master clock.

1. Synchronize the PTPCLKVAL with the PTP reference
   a. Select the PTP clock for timestamping (CORRCLK4TS+VALID)
   b. Trap PTP Sync and Followup frames to obtain the offset from the PTP timestamping clock (PTPCLKVAL) to the PTP master
   c. Use PTPCLKADD and PTPCLKSUB commands to correct the PTPCLK
   d. Details of the PTP / gPTP protocols are omitted here. It is assumed that the PTPCLK value is corrected after this step.
2. Configure PTPCLKCORP
   a. It is recommended to set PTPCLKCORP to the TAS cycle time.

3. Further adaption of the PTPCLK value needs to be executed to keep the scheduler in sync with other nodes. Adapt the PTPCLKVAL time by modulating PTPCLKRATE. **Note that direct changes of PTPCLKVAL or PTPCLKADD and PTPCLKSUB will not change the phase of the schedule**. E.g. adding 8ns via PTPCLKADD will **not** advance the schedule by 8 ns. The only mechanism to absolute phase of the scheduler as well as the PTPCLKVAL is through the PTPCLKRATE mechanism.

Do not use PTPCLKADD/SUB after the schedule is started.

7.4.1.6 802.1Qbv Pre-Standard Conformance

The SJA1105PQRS devices supports 802.1Qbv pre-standard and is in some aspects different to the final 802.1Qbv-2015 spec.

Credit Based Shaper

The Credit Based Shaper implementation is unaware of gate OPEN/CLOSE times. That is the value of idleSlope is conformant to 802.1Qav. Clause 8.6.8.2 is not implemented.

Gate Control List

In 802.1Qbv, a gate control list is associated with each port. This is an ordered list which contains gate operations. Each entry of this list has a parameter GateState which dictates the gate state for each queue and the TimeInterval after which the scheduler will proceed to the next list item (Clause 8.6.8.4).

The schedule cycle duration is configured through the parameter OperCycleTime (Clause 8.6.9.1.1). Each OperCycleTime, the Gate Control List is reexecuted, regardless of the sum of the TimeInterval values.

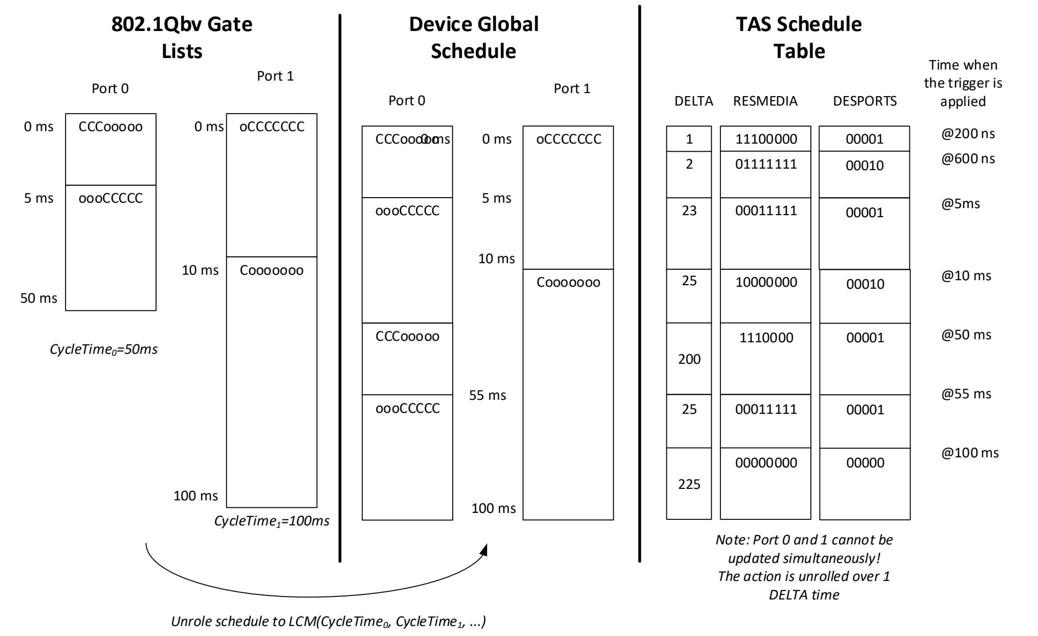
In the SJA1105PQRS, there is a single schedule control table (See Section 7.4.1.2) which controls the state of all ports. Hence the schedule duration for each port is identical and the sum of all DELTAs of the table.

The 802.1Qbv behavior can be emulated by unrolling the per-gate schedule into a single big per-device schedule. This is shown in the following example.

In the example, we start with two schedules. One applied to port 0 and another to port 1. Note that both have difference cycle times (50ms and 100ms). The first step is to unroll both schedules into a schedule with a common cycle time (100ms in this example). In the third step, both schedules are merged into a single TAS schedule. Not that there are situations which require you to add dummy DELTA cycles. For instance, at the beginning of the schedule an update of Port 0 (to RESPORT=11100000) and to port 1 (to RESPORT=01111111) is required. However, these setting cannot be done in a single step and need to be unrolled over two triggers.

Note that LCM in the following example is the least common multiple of all involved CycleTimes.

The queueMaxSDU parameter is not supported. Depending on the use-case the policing mechanism can emulate a similar behavior.



DELTA	RESMEDIA	DESPORTS	Time when the trigger is applied
1	11100000	00001	@200 ns
2	01111111	00010	@600 ns
23	00011111	00001	@5ms
25	10000000	00010	@10 ms
200	1110000	00001	@50 ms
25	00011111	00001	@55 ms
225	00000000	00000	@100 ms

Fig 50. **TAS Schedule Computation from 802.1Qbv Gate List**

Dynamic Schedule Configuration

The 802.1Qbv contains a mechanism that manages dynamic updates of the gate control list (List Config state machine). In the SJA1105PQRS, the schedule is fixed with the static configuration. The schedule cannot be changed at runtime.

7.4.2 Per Stream Policing

Stream Policing in the TSN context according to IEEE802.1Qci ensures that frames for critical traffic follow the traffic load plans regarding e.g. frame length and frame rate in case of rate constrained traffic. Frames violating the rules are dropped.

In contrast to L2 policing, which is available in all SJA1105PQRS variants, Stream Policing only deals with critical traffic, and is only available in SJA1105QS variants.

The VL Lookup table is used for filtering critical traffic based on criterions like destination MAC address, VLANID, VLANPRIO and the ingress port.

The VL Lookup table index of the matching entry is used for accessing the VL policing table, which contains the policing type (rate-constrained or time-triggered) and the type specific necessary parameters.

For rate-constrained (RC) critical traffic, the frame length is checked against a MAXLEN parameter, and the reception time is checked against a time window defined by the BAG parameter (minimal time between successive packets belonging to this stream) and the JITTER parameter. Both checks ensure, that the bandwidth consumed by the incoming frames of the stream is at or below the planned rate. For the meaning of BAG and JITTER see Fig 51:

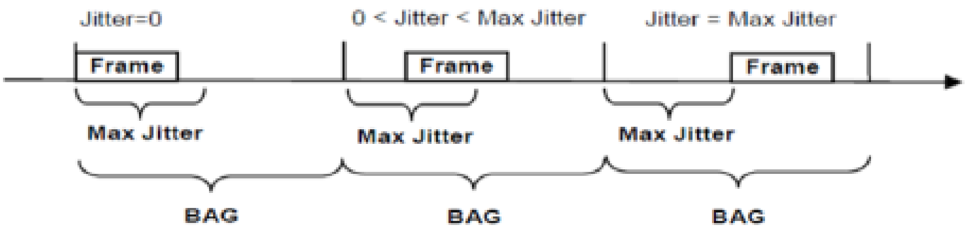


Fig 51. **Definition of BAG and JITTER for stream rate policing**

Please note, that streams with critical traffic consist of a sequence of equally spaced packets. The jitter setting allows to specify a time tolerance in the distance between adjacent packets, which might have been introduced by the communication path between the original sender and the current point. For rate-constrained traffic, stream policing protects against too low a distance between packets, while Fig 51 shows scenarios with packets being too late. This is no contradiction, as this is one visible effect of a single packet being delayed, which does not trigger a timing error. However, the other visible effect is, that the distance to following un-delayed packet will be shorter. This then might trigger a timing violation.

For time-triggered (TT) critical traffic, policing is done by a time window check for each received frame based on the schedule tables.

For each traffic stream handled by a VL Lookup table entry, there is a set of policing error counters for LENGTH, TIMING and UNRELEASED policing errors. The counter set is accessed using the VL Lookup table index, see Virtual Link Status Register in the UM. Error counter sets without valid VL Lookup table entry return random values. Error counters stay on the maximum count when full.

Once the critical frames have passed the policing stage, a forwarding decision is done based on an entry in the VL Forwarding table, which is also referenced by the VL Lookup table index. The DESTPORTS vector determines the egress port(s), PRIORITY in which priority queue of the output port(s) the frame is placed, and TYPE, if the output is dispatched immediately (rate-controlled) or is triggered by a schedule (time-triggered).

If VLLUPFORMAT (General Parameters Table) is asserted, critical traffic is no longer identified by the tuple (destination MAC, VLANID, VLANPRIO, ingress port), but by a match between the 32 MSBs of the destination MAC with the VIMARKER/VIMASK configuration in the General Configuration Table. For this addressing scheme the 16 LSBs of the destination MAC are used as a VL identifier which is the search key for a matching entry in the VL Lookup table.

The VL handling mechanism can also be used to deal with non-critical traffic. An entry in the VL Lookup table may be marked with ISCRITCAL being de-asserted. For any matching frames, no policing tests are performed, and only best-effort handling is carried out.

Please note, that the VL Lookup table must be sorted. For more detail, see UM11040.

7.5 Security

In this section, special security use-cases are discussed. The purpose of this section is to point the reader to the relevant configuration options to control a security use-case. The subsections provide a pointer to the relevant registers or tables and/or explain how the parameters are used to control security.

7.5.1 Port Reachability

The port reachability restricts port admission to a destination port. The setting is controlled via the REACH_PORT vector of the L2 Forwarding Table. This allows to physically separate ports from any kind of cross-communication.

Note that communication relations can also be restricted unidirectionally. This is done by declaring port A reachable from port B but not the other way around.

Further restrictions can be made by configuring how frames with an unknown DA are forwarded (FL_DOMAIN) or how broadcast frames are forwarded (BC_DOMAIN).

See Section 7.3.8 on detailed information how to configure the L2 Forwarding Table.

7.5.2 VLAN Configuration

7.5.2.1 Single tag and general VLAN behavior

The VLAN settings are controlled by the VLAN Lookup Table. For VLANs a whitelisting strategy is implemented and unconfigured VLANs are implicitly dropped. If no VLAN table is loaded at all, VID 0 is automatically configured (see user manual).

The VLAN Lookup Table allows to control VLAN based reachability (VLAN_BC) and membership (VMEMB_PORT) rules.

The membership vector controls which ports are allowed to receive a frame of the associated VID. The broadcast vector controls the reachability and lists the ports accessible for egress.

Note that port mirroring overrules such settings. In production, mirroring should be disabled by setting MIRR_PORT to an invalid port value.

7.5.2.2 Double tag detection and blocking

Switches which do not detect double tagged frames are prone to VLAN hopping (DoS) attacks: In VLAN hopping a frame with two tags is transmitted by a malicious device. Switches that do not recognize this behavior treat this frame as a single tagged frame, and (depending on the configuration) strip the outer tag, leaving a tag which potentially violates membership rules.

The SJA1105PQRS devices detect and prevent such attacks.

Frames with two VLAN tags (QinQ) are identified by the TPID (0x88A8) and TPID2 (0x8100) parameters found in the General Parameters Table.

The switch evaluates the first two tags and checks them for consistency. Illegally tagged frames (Two S-tags, C-tag followed by S-tag, two S-tags) are dropped.

Additional per-port settings in the MAC Configuration Table control if double tagged, single outer tagged, single inner tagged or untagged frames are admitted on a port. These parameters are DRPDTAG, DRPSOTAG, DRPSITAG, DRPUNTAG.

© NXP Semiconductors N.V. 2020. All rights reserved.

Note that during the actual VLAN membership check (VLAN Lookup), the SJA1105PQRS does not differentiate if the VID originates from an outer (S-tag) or inner (C-tag). The configuration of a VID implicitly enables outer as well as inner tagged frames, assuming the port is configured to receive such frames.

7.5.3 Quality of Service (QoS)

7.5.3.1 Priority Handling and Regeneration

The switch allows to detect flows based on 802.1Q PCP (priority code point) values. Based on the PCP value, a stream can be mapped to a queue (L2 Forwarding Table). Along with the mapping a new PCP value can be assigned for transmission (Priority Regeneration).

See Section 7.3.8 on detailed information how to configure the L2 Forwarding Table.

7.5.3.2 Policing and Broadcast Storm Prevention

The switch allows to meter flows based on a priority/port combination. The ingress rate of broadcast frames can be controlled (broadcast storm prevention).

See QoS – Policing for configuration.

7.5.3.3 Egress Shaping (Credit Based Shaper)

The switch has up to 16 credit based shaper blocks. They can be freely assigned to any queue/port combination.

If configured, the shaper will reduce the egress burst size which improves buffer utilization across your network topology.

See Timestamping -AVB for configuration details.

7.5.4 Address Learning

7.5.4.1 Dynamic Address Learning

Dynamic address learning can be enabled/disabled per port via the DYN_LEARN flag (MAC Configuration Table).

In typical applications, some MAC addresses (e.g. multicast and unicast) are pre-programmed or dynamically programmed by a processor while others are dynamically learned by the switch on the fly.

Statically programmed MAC addresses are recommended to be programmed to low indices (INDEX, L2 Lookup Parameter). This guarantees that they are preferred in the lookup process. If multiple rules match, the one with the lowest index is selected.

Memory can be allocated explicitly for dynamic learning, this allows to split static and dynamic addresses physically in the TCAM memory. START_DYNSPC (L2 Lookup Parameter Table) is used as a separator and marks the index of the first entry used to store a dynamically learned address. It is strictly advised to keep the number of static entries and the boundary indicated by START_DYNSPC consistent.

If the TCAM is full, a new address is to be learned and OWR_DYN is asserted, the switch will start overwriting old dynamic entries (but never static entries) incrementally starting from START_DYNSPC and uses a round-robin policy to find the eviction candidate.

7.5.4.2 Dynamic Address Learning Limitation

The number of TCAM space available for dynamic learning can be easily restricted on a per-port basis.

For this the MAXADDRP[x] settings in the L2 Lookup Parameters Table are used. They define the maximum number of MAC addresses, which can be learned for each port. If the switch has learned enough addresses to exhaust a port's available slots, addresses are not learned from that point on. Note, that in order to make use of this, DYN_LEARN must be set.

If the maximum number of entries for dynamic learning has been reached, frames cannot be learned due to an out of memory condition. If DRPNOLEARN is set, the frame with the new address will be dropped. If DRPNOLEARN is not set, the frame will be forwarded according to the FL_DOMAIN (L2 Forwarding Table) setting. This is sometimes called “flooding”.

DRPNOLEARN is only effective when DYN_LEARN is set: If DYN_LEARN is de-asserted, dynamic address learning based on received frames is disabled, but the frames still get forwarded regardless of the DRPNOLEARN setting . This is the basic flooding mechanism which is always used when there is no TCAM entry for a destination address.

Table 43. Frame learning and frame dropping behavior

DYN_LEARN	DRPNOLEARN	per-port space exhausted (MAXADDR[x])	configured dynamic table space exhausted	OWR_DYN	new address entered in table	Frame with unknown address dropped [1]	old entries overwritten	Description
0	X	X	X	X	0	0	0	only static entries
1	0	X	X	0	1	0	0	standard, problem free address learning use case
1	0	X	0	1	1	0	0	
1	1	0	0	X	1	0	0	
1	0	X	1	1	1	0	1	configured dynamic space exhausted, old entry overwritten, not dropped
1	1	1	X	0	1	1	0	
1	1	0	1	0	1	1	0	
1	1	1	0	1	1	1	0	
1	1	X	1	1	1	1	1	

Remarks:

(1) If an unknown frame is not dropped it is flooded to the FL_DOMAIN

(2) OWR_DYN is only effective when configured dynamic address space is exceeded.

(3) Declaring ∑(MAXADDRP[x]) larger than the overall available dynamic space is no error, but may trigger the valid corner case, where per-port space is still available, but total dynamic space is used up.

(4) Frames are dropped if DROPNOLEARN is enabled AND if either total dynamic space or per-port slots are exhausted.

7.5.4.3 Single Shot Learning

Single shot learning ties source MAC addresses to a single port. This mode is a dynamic learning strategy which is stricter than normal dynamic learning. Once an address is learnt, it is not possible to re-learn the same address on a different port until the entry ages out due to the ageing mechanism. The aging counter is updated when frames are received on the port on which they are initially learned but not when a frame arrived on an unexpected port.

This mode is selected when LEARN_ONCE (L2 Lookup Parameters) is asserted.

Note that if LEARN_ONCE is set, all other entries in the L2 Lookup Table are implicitly port enforced, regardless of their ENFPORT setting.

7.5.5 MAC Address Blacklisting

MAC address blacklisting is used to exclude a known set of addresses.

In order to blacklist a destination or source MAC address, create a static L2 entry and deassert all DESTPORTS bits and assert ENFPORT.

```
l2_address_lookup_table.append({
  "INDEX"        : 0,
  "ENFPORT"      : 1,
  "DESTPORTS"    : 0,
  "MACADDR"      : 0x001094000001,
  "VLANID"       : default_vlan,
  "IOTAG"        : 0,
  "MASK_MACADDR" : 0xFFFFFFFFFFFF,
  "MASK_VLANID"  : 0xFFF,
  "MASK_IOTAG"   : 0x1,
  "RETAG"        : 0,
  "MIRR"         : 0,
  "TAKETS"       : 0,
  "MIRRVLAN"     : 0,
  "TSREG"        : 0})
```

Fig 52. **Blacklisting a MAC address**

7.5.6 MAC Address Whitelisting

In MAC address whitelisting only known MAC address are forwarded. There are two use-cases associated with this feature: destination address whitelisting and source address whitelisting. In destination address whitelisting frames with unknown DA stations are dropped. In source address whitelisting, frames with unknown SA address or invalid source address / ingress port combinations are dropped and not learned.

Destination address whitelisting is enabled by configuring the flooding domain of a port. The flooding domain (FL_DOMAIN, L2 Forwarding Table) configures how frames with an unknown destination address are forwarded.

Source address whitelisting is implemented by two means: binding known source addresses to a source port and dropping frames with unknown source addresses.

A source MAC address is tied to a source port via the ENFPORT setting (L2 Lookup Table). To understand this setting, the operation of the lookup process must be explained. For each frame, the L2 Address Lookup Table is searched twice. On a first run, it is searched by the source address for learning and for the ENFPORT check. On the second run the destination address is searched for obtaining the forwarding vector. If the mechanism finds that a source address matches an entry, the entry’s ENFPORT is set, and the ingress port does not match the entry’s DESTPORTS, the frame is dropped.

To ensure that an unknown source address is not learnt but dropped, the MAXADDRP[x] (L2 Lookup Parameters Table) is used in conjunction with the DRPNOLEARN flag. If DRPNOLEARN is set, frames which cannot be learned (due to out of TCAM memory conditions) are dropped. In order to trigger this, MAXADDRP[x] must be set to zero for the respective ports.

Note that deasserting DYN_LEARN (MAC Configuration Table) **does not** cause a frame to be dropped if DRPNOLEARN is set.

```
l2_address_lookup_table.append({
  "INDEX"          : 0,
  "ENFPORT"        : 1,
  "DESTPORTS"      : 1 << 4,
  "MACADDR"        : 0x001094000001,
  "VLANID"         : default_vlan,
  "IOTAG"          : 0,
  "MASK_MACADDR"   : 0xFFFFFFFFFFFF,
  "MASK_VLANID"    : 0xFFF,
  "MASK_IOTAG"     : 0x1,
  "RETAG"          : 0,
  "MIRR"           : 0,
  "TAKETS"         : 0,
  "MIRRVLAN"       : 0,
  "TSREG"          : 0})
```

Fig 53. **Enforce Port Example**

7.5.7 802.1X Port-Based Authentication

The SJA1105PQRS supports the functionality to implement 802.1X in software. In order to implement 802.1X, certain port states need to be implemented. After power-up a port is unauthenticated and waits for EAP frames. Normal communication (ingress and egress) is forbidden and also MAC address learning must be prohibited.

© NXP Semiconductors N.V. 2020. All rights reserved.

Software-wise, this is implemented through the INGRESS, EGRESS and DYN_LEARN flag of the MAC Configuration table. These flags can be controlled during run-time through the dynamic control interface (address 0x4b to 0x53).

If a port is disabled for INGRESS, it will still accept MGMT frames. If a frame is received which qualifies the MAC_FLT (General Parameters Table), it is forwarded to the HOST port for processing. The host implements the EAP authentication.

Similarly, if a port is disabled for EGRESS it is still possible to send MGMT frames, given that the MGMT table (see Management L2 Address Lookup table in the user manual) is configured.

If a device authenticated successfully, INGRESS and EGRESS can be asserted to allow normal operation.

Make sure, that all PHYs connected to the ports notify the host processor for link-flaps and disconnects. Such events should typically trigger the software to set the port to unauthenticated.

7.6 Configuration Examples

7.6.1 Mirroring

Frame mirroring configures the switch to forward copies of traffic received or sent on a particular port to a designated mirroring port.

There are three methods to enable mirroring and select the frames to be mirrored:

- Port based: any frame ingressing and/or egressing a port

  Controlled with ING_MIRR and EGR_MIRR of the MAC configuration table. (see below)

- VLAN-based: any frame belonging to a certain VLAN

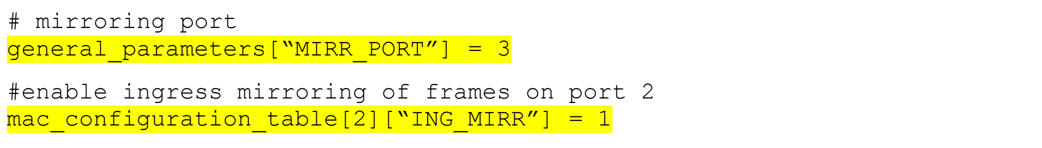
  Controlled with VING_MIRR and VEGR_MIRR of the VLAN's entry in the VLAN Lookup Table.

- Address based: any frame matching a L2 address table entry (TCAM)

  Controlled by the MIRR bit in the L2 Address Lookup table

Mirroring can be enabled for either ingressing or egressing frames on each of the ports individually. The exact port that receives the mirrored frames (mirror port) is configured via the General Parameters table, and this field needs to contain a valid port index to enable mirroring on the switch.

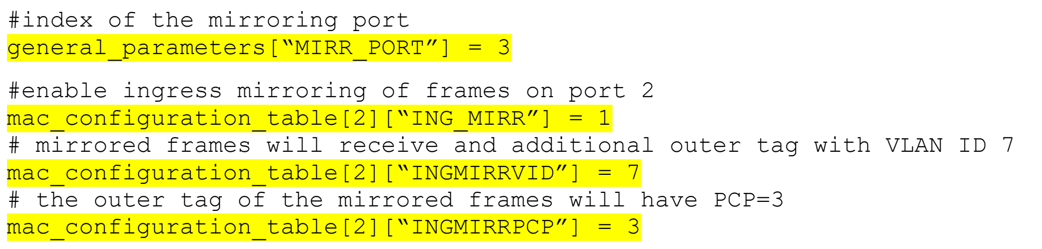
It is possible to separate the mirrored traffic from the original flow by attaching a outer-tagged VLAN tag to the mirrored frames. This can be done for both ingress (configured via the MAC Configuration table) and egress (configured via the General Parameters table) mirroring.



```
# mirroring port
general_parameters[“MIRR_PORT”] = 3

#enable ingress mirroring of frames on port 2
mac_configuration_table[2][“ING_MIRR”] = 1
```

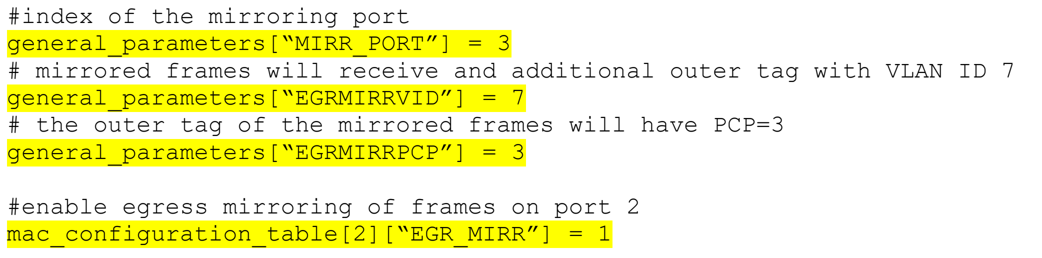
Fig 54. **Ingress mirroring of frames received on port 2 to port 3 without tagging**



```
#index of the mirroring port
general_parameters[“MIRR_PORT”] = 3

#enable ingress mirroring of frames on port 2
mac_configuration_table[2][“ING_MIRR”] = 1
# mirrored frames will receive and additional outer tag with VLAN ID 7
mac_configuration_table[2][“INGMIRRVID”] = 7
# the outer tag of the mirrored frames will have PCP=3
mac_configuration_table[2][“INGMIRRPCP”] = 3
```

Fig 55. **Ingress mirroring of frames received on port 2 to port 3 with additional tagging of mirrored frames**



```
#index of the mirroring port
general_parameters[“MIRR_PORT”] = 3
# mirrored frames will receive and additional outer tag with VLAN ID 7
general_parameters[“EGRMIRRVID”] = 7
# the outer tag of the mirrored frames will have PCP=3
general_parameters[“EGRMIRRPCP”] = 3

#enable egress mirroring of frames on port 2
mac_configuration_table[2][“EGR_MIRR”] = 1
```

Fig 56. **Egress mirroring of frames received on port 2 to port 3 with additional tagging of mirrored frames**

7.6.2 VLANs

VLANs allow to separate traffic flows within the device. In order to be able to receive frames of a certain VLAN, the ports of the switch need to be configured members of that VLAN. In order to send the frames tagged with a particular VLAN ID, the port needs to be a member of its broadcast domain. The VLAN Lookup table contains separate entries for each VLAN ID, and needs to be populated for the switch to work with particular VLAN IDs.

An entry is specific for one VLAN ID and configures its member ports, the broadcast domain and whether or not the frames remain tagged with this VLAN ID on egress.

```
vlan_lookup_table.append({
    "VING_MIRR"         : 0,
    "VEGR_MIRR"         : 0,
    # all ports can receive frames
    "VMEMB_PORT"        : 0x1F,
    # all ports can send frames
    "VLAN_BC"           : 0x1F,
    # frames are sent untagged on all ports
    "TAG_PORT"          : 0x00,
    # configuration for VLAN ID 17
    "VLANID"            : 17})
```

Fig 57. **VLAN entry configuration example 1**

```
vlan_lookup_table.append({
    "VING_MIRR"         : 0,
    "VEGR_MIRR"         : 0,
    # port 4 cannot receive frames
    "VMEMB_PORT"        : 0x0F,
    # port 4 cannot send frames
    "VLAN_BC"           : 0x0F,
    # frames are sent tagged only on ports  0 and 2
    "TAG_PORT"          : 0x05,
    # configuration for VLAN ID 55
    "VLANID"            : 55})
```

Fig 58. **VLAN entry configuration example 2**

Please note, that in contrast to what the names of the port bitset setup entries might suggest, the implemented semantic is slightly different:

- VMEMB_PORT determines, if any frames belonging to that VLAN may be *ingressed* if received on that port.
- VLAN_BC determines, if a frame (unicast, multicast and broadcast) belonging to that VLAN may be *egressed* on that port.

7.6.3 Ingress Policing

Ingress policing of the frames on the ports allows to restrict the incoming traffic on a port with respect to the burst size and frame length. Different policing rules can be configured for frames received on each of the ports and with each of the priorities. Ingress frames which violate the restrictions placed by these rules are dropped.

The L2 policing table contains 45 entries. The first 40 are used for per-port per-priority policing, and the latter 5 define the policing rules for the broadcast traffic. By default, the entry with SHARINDX=0 is used if no rules were defined for the specific port and priority combination.

```
for port in range(NO_ETH_PORTS): # for each of the ports
    for prio in range(NO_PRIORITIES): # for each of the priorities
        l2_policing_table.append({
            # individual policing block for each priority on each port
            "SHARINDX"  : port * NO_PRIORITIES + prio,
            # Maximum burst of 10 maximum sized, double tagged frames
            "SMAX"      : 10 * 1526,
            # Unit: [15.625 kbps]
            "RATE"      : 6400,  # bandwidth is credited at 100 Mbps
            # maximum accepted frame length is 1526 B
            "MAXLEN"    : 1526,
            "PARTITION" : prio
        })
```

Fig 59. **Ingress policing rule example 1**

```
for port in range(NO_ETH_PORTS): # for each of the ports
    for prio in range(NO_PRIORITIES): # for each of the priorities
        l2_policing_table.append({
            # same policing block for all ports and priorities
            "SHARINDX"  : 0,
            # Maximum burst of 15 frames of size 512 B
            "SMAX"      : 15 * 512,
            # Unit: [15.625 kbps]
            "RATE"      : 64000, # bandwidth is credited at 1 Gbps
            # maximum accepted frame length is 512 B
            "MAXLEN"    : 512,
            "PARTITION" : prio
        })

for port in range(NO_ETH_PORTS):
    l2_policing_table.append({
        "SHARINDX"  : 40 + port,
        # Maximum burst as 10 maximum sized, double tagged frames
        "SMAX"      : 10 * 1526,
        "RATE"      : 6400,  # bandwidth is credited at 100 Mbps
        "MAXLEN"    : 1526,
        "PARTITION" : 0
    })
```

Fig 60. **Ingress policing rule with broadcast policing example 2**

7.6.4 Priority Remapping

The switch supports manipulation of the PCP field in tagged frames between ingress and egress on its ports. Priority remapping is used to modify the tag of the frames coming on a particular port and with a particular priority with a different priority on egress. Please also refer to Section 7.3.8.

L2 Forwarding table contains separate entries for each port of the switch, which allows to provide unique remapping for all ports if necessary. The index of VLAN_PMAP corresponds to the ingress PCP for which the egress PCP is defined.

```
for i in range(NO_ETH_PORTS):
    reachable_ports = 0x1F & ~(1 << i)
    broadcast_domain = 0x1F & ~(1 << i)
    default_route    = 0x1F & ~(1 << i)

    l2_forwarding_table.append({
        "FL_DOMAIN"    : default_route,
        "BC_DOMAIN"    : broadcast_domain,
        "REACH_PORT"   : reachable_ports,
        "VLAN_PMAP[0]" : 0
        "VLAN_PMAP[1]" : 1,
        "VLAN_PMAP[2]" : 2,
        "VLAN_PMAP[3]" : 3,
        "VLAN_PMAP[4]" : 4, # ingress frames with PCP=3 will retain PCP=3 on
egress
        "VLAN_PMAP[5]" : 5,
        "VLAN_PMAP[6]" : 6,
        "VLAN_PMAP[7]" : 7})
```

Fig 61. **L2 Forwarding table configuration with no remapping**

```
for i in range(NO_ETH_PORTS):
    reachable_ports = 0x1F & ~(1 << i)
    broadcast_domain = 0x1F & ~(1 << i)
    default_route    = 0x1F & ~(1 << i)

    l2_forwarding_table.append({
        "FL_DOMAIN"    : default_route,
        "BC_DOMAIN"    : broadcast_domain,
        "REACH_PORT"   : reachable_ports,
        "VLAN_PMAP[0]" : 7,
        "VLAN_PMAP[1]" : 6,
        "VLAN_PMAP[2]" : 5,
#ingress frames with PCP=3 will have   PCP=4 on egress
        "VLAN_PMAP[3]" : 4,
        "VLAN_PMAP[4]" : 3,
        "VLAN_PMAP[5]" : 2,
        "VLAN_PMAP[6]" : 1,
        "VLAN_PMAP[7]" : 0})
```

Fig 62. **L2 Forwarding table configuration with reverse mapping of PCP values**

7.6.5 Per-Stream Policing

The following section shows an Stream Policing configuration using the virtual link mechanism.

```
VLLUPFORMAT = 0
VIMARKER/VIMASK = don’t care
```

Fig 63. **General Parameter Table**

```
MAC of 00:10:94:00:00:01,
VLANID=5
VLANPRIO=2
```

```
frame length of 128 Bytes
payload length = 106 Bytes
sent every 10ms
```

Fig 64. **Example Stream**

Required forwarding properties of the switch: The stream is to be received on port 1 and forwarded to port 3. We want rate-constrained policing on ingress, and rate-constrained egress. Frame should be egressed on Prio4. Use memory partition 0 for storing frames until egressing.

The following entries are made:

```
VL Lookup Table:
   VLANPRIOR = 1
   PORT = 1
   VLANID = 5
   MACADDR = 00:10:94:00:00:01
   ISCRITICAL = 1
   DESTPORTS = 0 # don't care
VL Policing Table:
   JITTER = 200 # 2ms
   BAG = 100 # 10ms
   SHARINDX = 0
   MAXLEN = 128
   TTRC = 0
VL Forwarding Table:
   DESTPORTS = (1 << 3)
   PARTITION = 0
   PRIORITY = 4
   TYPE = 0
VL Forwarding Properties Table:
   PARTSPEC(0) = 24 # 12*128=1536 - one full frame
   PARTSPEC(1)..PARTSPEC(7) = 0
   DEBUGEN=1
```

Fig 65. **Example configuration settings for stream based policing**

7.6.6 PTP

For the theory behind the examples in this chapter see ch. 7.3.

Assumptions: The SYNC frame is presumed to have 01-80-c2-00-00-0e as DEST_MAC, and be conformant to

Table 44. Static configuration of 1-step and 2-step SYNC frame handling

Field	1-step	2-step
HOST_PORT	Don't care	Port to mgmt. processor
L2 lookup table	01-80-c2-00-00-0e must NOT be included !!	01-80-c2-00-00-0e must NOT be included !!
MAC_FLT[]		01-80-c2-00-00-0e

Field	1-step	2-step
MAC_FLTRES[]	Must not trap 01-80-c2-00-00-0e	ff-ff-ff-00-00-ff
SEND_META[]		1
INCL_SRCPT[]		1
DESTMETA	-	e.g. 02-60-37-de-ca-de
SRCMETA	-	e.g. 02-60-37-c0-ff-ee
IGNORE2STF	1	0

DESTMETA and SRCMETA can be freely chosen as long as there is no interference with L2 lookup table entries or MAC addresses with special semantics. They are an agreement with the ethernet input filter of the management processor. The example addresses are from the locally administered MAC address range (02-xx-xx-xx-xx-xx)

It is common practice to leave CORRCLK4TS to 0 to use the free running PTPCLK for timestamping. If the rate error of the free running clock is known and must be included into the transit time calculation, this can be performed in SW with the 2-step method.

7.6.6.1 1-Step setup

There are no further steps necessary besides loading a proper static configuration stream, which meets the properties in Table 44.

The IGNORE2STF overrides the indication in the SYNC frame, which might request to use the 2-step method.

When flooding a SYNC frame, the SJA1105PQRS increments on each egressed frame the correction field by the individual transit time this frame has experienced. This might be different for each egressed frame, e.g. due to the egress queue situation.

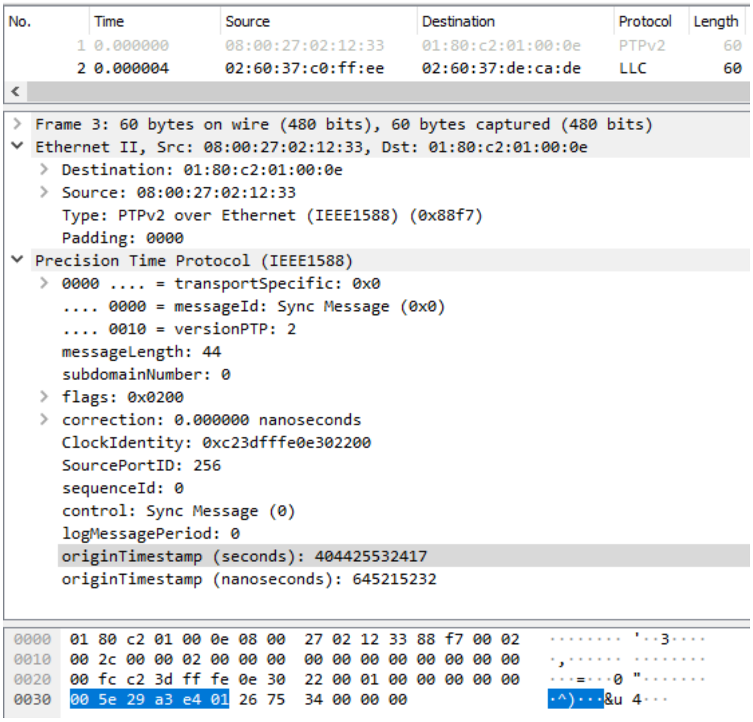
A SYNC frame traversing from one 100Mbit port to another 100Mbit port without any additional traffic automatically receives an increment of about 13700ns. This value matches roughly with the measurements presented in ch. 9.1.

7.6.6.2 2-Step setup

Again, the entire configuration for a 2-step setup can be placed in the static config stream according to Table 44.

For each SYNC frame two frames are egressed to the HOST_PORT: the trapped sync frame with SWITCH_ID and PORT embedded into the destination address, and the META frame (see Fig. 13 of UM11040).

A dissected SYNC frame and META frame, which have been created with the configuration used in this example are shown in Fig 66 and Fig 67:

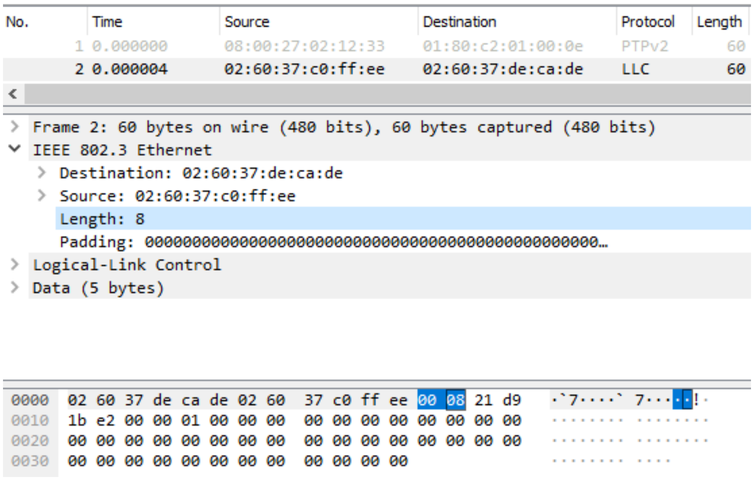


```
No.  Time        Source              Destination         Protocol  Length
   1 0.000000    08:00:27:02:12:33   01:80:c2:01:00:0e   PTPv2         60
   2 0.000004    02:60:37:c0:ff:ee   02:60:37:de:ca:de   LLC           60

> Frame 3: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
v Ethernet II, Src: 08:00:27:02:12:33, Dst: 01:80:c2:01:00:0e
   > Destination: 01:80:c2:01:00:0e
   > Source: 08:00:27:02:12:33
     Type: PTPv2 over Ethernet (IEEE1588) (0x88f7)
     Padding: 0000
v Precision Time Protocol (IEEE1588)
   > 0000 .... = transportSpecific: 0x0
     .... 0000 = messageId: Sync Message (0x0)
     .... 0010 = versionPTP: 2
     messageLength: 44
     subdomainNumber: 0
   > flags: 0x0200
   > correction: 0.000000 nanoseconds
     ClockIdentity: 0xc23dfffe0e302200
     SourcePortID: 256
     sequenceId: 0
     control: Sync Message (0)
     logMessagePeriod: 0
     originTimestamp (seconds): 404425532417
     originTimestamp (nanoseconds): 645215232

0000  01 80 c2 01 00 0e 08 00  27 02 12 33 88 f7 00 02   ········ '··3····
0010  00 2c 00 00 02 00 00 00  00 00 00 00 00 00 00 00   ·,······ ········
0020  00 fc c2 3d ff fe 0e 30  22 00 01 00 00 00 00 00   ···=···0 "·······
0030  00 5e 29 a3 e4 01 26 75  34 00 00 00               ·^)···&u 4···
```

Fig 66. **Dissected SYNC frame on the HOST_PORT**



```
No.  Time        Source              Destination         Protocol  Length
   1 0.000000    08:00:27:02:12:33   01:80:c2:01:00:0e   PTPv2         60
   2 0.000004    02:60:37:c0:ff:ee   02:60:37:de:ca:de   LLC           60

> Frame 2: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
v IEEE 802.3 Ethernet
   > Destination: 02:60:37:de:ca:de
   > Source: 02:60:37:c0:ff:ee
     Length: 8
     Padding: 00000000000000000000000000000000000000000000000000…
> Logical-Link Control
> Data (5 bytes)

0000  02 60 37 de ca de 02 60  37 c0 ff ee 00 08 21 d9   ·`7····` 7·····!·
0010  1b e2 00 00 01 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0030  00 00 00 00 00 00 00 00  00 00 00 00               ········ ····
```

Fig 67. **Dissected FOLLOW_UP frame on the HOST_PORT**

7.6.6.3 PTP output clock

The rate-adjusted timestamp clock in PTPCLKVAL can be brought to the PTP_CLK pin for supervision and other purposes. There is a divider between PTPCLKVAL and

© NXP Semiconductors N.V. 2020. All rights reserved.

PTP_CLK pin, so the frequency is adjustable. In case the output is 1Hz this is often called a PPS or 1PPS output.

These are the necessary steps for a 20Hz (50ms) clock output. The start of the PTP_CLK output can be accurately timed by the PTPPINST compare register, and is set to 200ms into the future.

```
# Register addresses
PTPCONTROL_REG = 0x18
PTPPINDUR_REG = 0x17
PTPPINST_U_REG = 0x16
PTPPINST_L_REG = 0x15
PTPTSCLK_U_REG = 0x1d
PTPTSCLK_L_REG = 0x1c

platform.spi_write(PTPPINDUR_REG, 50*125000)      # 8ns * (50 * 125000) = 50ms

current_ts = platform.spi_read(PTPTSCLK_L_REG) + \
            (platform.spi_read(PTPTSCLK_U_REG) << 32)
# start toggling output 200ms from the current_ts fetch:
start_ts = current_ts + 200*125000                # 8ns * 125000 * 200 = 200ms

platform.spi_write(PTPPINST_U_REG, start_ts >> 32)
platform.spi_write(PTPPINST_L_REG, start_ts & 0xffffffff)

platform.spi_write(PTPCONTROL_REG, (1<<31|1<<28)) # VALID, STARTPTPCP,PTPCLKADD
```

Fig 68. **PTP_CLK toggle output**

7.6.7 Cascading Switches

SJA1105PQRS switches can be connected to add more ports. There are peculiarities which need special attention.

For reasons explained below the SJA1105PQRS allows only a linear topology of cascaded switches. Tree type topologies are not supported if a management processor is needed. However, for applications not needing the management-processor, the topology of the switch cluster is not limited.

7.6.7.1 Ethernet links for cascading

For the communication link between two SJA1105PQRS, a Gbit link is recommended to create a channel with maximum capacity. This reduces the bottleneck for traffic between ports on different switches.

SGMII or RGMII are the available MII interfaces with Gbit speed. SGMII is available on the SJA1105RS variants. RGMII requires more wires between connected ports than SGMII.

7.6.7.2 Clocks and Timestamp counter synchronization

In a cascaded switch design the PTP framework requires that the timestamp counters in all switches must run synchronously (see section 7.3.9). This means, that clock increments happen at the same time (no drift), and there is no offset between the timestamp counters.

The SJA1105PQRS features a free running and a rate corrected timestamp counter.

For this, two provisions must be made:

- Drift: Only one switch uses a crystal oscillator (“Clock master”), the other switches’ clock input must come from the clock master’s CLK_OUT pin, as described in ch. 2.3.4. With this, the timestamp clock counters advance at the same rate.
- Any offset between the timestamp clocks on the switches must be removed. This is done with the *cascading clock sync mechanism*. This must not be confused with the clock synchronization mechanism between ethernet connected nodes which is needed e.g. for traffic shaping in Time-Triggered Ethernet, and which can be performed using the PTP protocol.

  The cascading clock sync mechanism allows to determine the timestamp offset between the switches and remove it. Since the timestamp counters are run from the same crystal, offset elimination generally has to be done only once.

To support the cascading clock synchronization there is a dedicated HW sync line. It connects the PTP_CLK[7] pins of all switches. One of the switches is selected to be the “Sync Master” switch. PTP_CLK is an output on the sync master and an input on the sync slaves. The role is defined by the CAS_MASTER bit in the AVB parameter table of the static configuration and cannot be changed during runtime. The sync line is activated by writing a ‘1’ to the CASSYNC bit (bit 26 of PTP control register 1) on the sync master, see Fig 69.

```
# set line high
platform.spi_write(0x18, (1<<31) | (1<<26))    # VALID + CASSYNC
. . . .
# set line low
platform.spi_write(0x18, (1<<31) )             # VALID
```

Fig 69. **CASSYNC control**

The CASSYNC bit works like a data bit of a GPIO output register. It is NOT self-clearing.

The rising edge triggers a simultaneous timestamp capture on the sync master and sync slaves. These timestamps are available in the PTPSYNCTS register. They are used for offset calculation and for adjusting the clocks to the same value. Offset correction is assisted by hardware to avoid race conditions.

7.6.7.3 Trapping in a cascading setup

Management traffic is identified by a match to the MAC_FLT[] settings (see 7.3.1.). INCL_SRCPT[] indicates, if the SWITCH_ID and the ingress port is embedded into the trapped frame’s DST MAC address, before it is forwarded to the HOST_PORT. Usually, this port connects directly to the µC for management.

In a cascaded setup as in the example in Fig 70, a management frame trapped in Switch-3 does not reach the host directly, but must travel through Switch-2 and Switch-1 to reach the host. During the ingress process in Switch-2, this frame is trapped a 2nd time, since MAC_FLT[] and INCL_SRCPT[] settings are usually identical on all switches. To avoid a 2nd modification of the DST MAC (and lose the original valid SWITCH_ID and ingress port information) DST MAC modification can be inhibited for the port in CASC_PORT.

[7] PTP_CLK is a dual use pin. Besides the CAS_SYNC functionality, it can also be configured to output a clock derived from the timestamp clock, see ch. 7.6.7.2. The clock output must be switched off for CAS_SYNC to work.

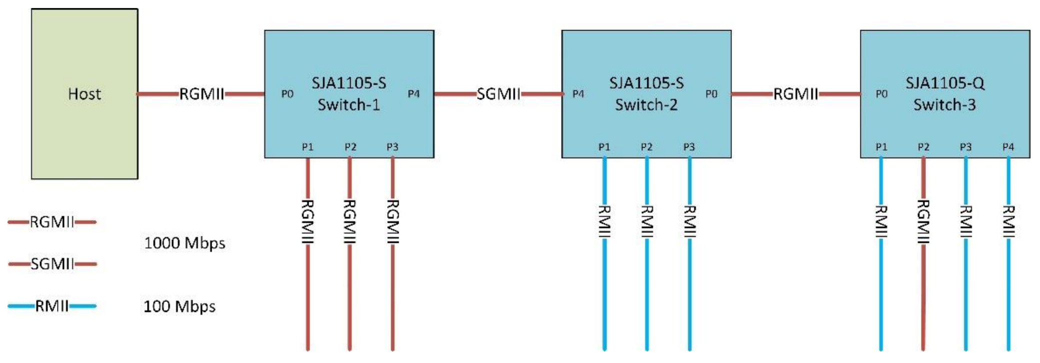


Fig 70. **Cascading example**

CASC_PORT contains is a port-number, not a port bit-set (vector), so there may be only one port on each switch to be used for receiving management traffic. This means, that only linear topologies (string of switches) are supported, and that the host must be connected to either the first or the last switch in the row.

7.6.7.4 Sending management frames

Sending management traffic from the host via a particular port can be done with the Management L2 Address Lookup table, as described in ch 7.3.2.

This also holds true for a cascaded configuration, except a management entry must be written to each switch, which is passed by the frame on its way to its egress port(s).

Using the example in Fig 70: for a frame with DST MAC 01-80-c2-00-00-0e, which should be egressed to:

- Switch-1 P3,
- Switch-2 P1, P3
- Switch-3 P2, P3, and P4

the following settings are required in their respective L2 management lookup entry:

Table 45. L2 management table example for cascaded switches

Config item	Switch-1	Switch-2	Switch-3
MACADDR	01-80-c2-00-00-0e	01-80-c2-00-00-0e	01-80-c2-00-00-0e
DESTPORTS	0x18	0x0b	0x1c
MGMTVALID	1	1	1

These entries must be set before the frame is sent by the host. They have to be written again for the next frame, because they are automatic invalidated after being used.

7.6.7.5 Static config stream

The following table shows how certain items should be set for the use case of two cascading switches as shown in Fig 70 :

Table 46. Static config items for cascaded switches

Config item	Switch-1	Switch-2	Switch-3	Remark
SWITCH_ID	0	1	2	

Config item	Switch-1	Switch-2	Switch-3	Remark
HOST_PORT	0	4	0	Link in direction of the host
CASC_PORT	4	0	6	No need for CASC_PORT on Switch-3
CAS_MASTER	1	0	0	Switch-1 is SyncMaster

The following static config items should be configured consistently on all switches:

- VLAN table
- L2 lookup table.

7.6.8 Temperature sensor

The SJA1105PQRS features a temperature sensor and a comparator against a configurable threshold. Direct temperature measurements are not possible, but the actual temperature can be determined, by modifying the threshold.

The following example code snippet tests the temperature against an increasing threshold. A more resource aware method would be a bisection algorithm.

```
############################################################
# temp sensor relevant constants
TS_CONFIG = 0x100A00
TS_STATUS = 0x100A01

TEMP_LUT = {
  0b000000: -999.9,   0b000001: -45.7,    0b000010: -41.7,    0b000011: -37.5,
  0b000100: -33,      0b000101: -28.4,    0b000110: -23.5,    0b000111: -18.3,
  0b001000: -11.4,    0b001001: -6.1,     0b001010: -2.1,     0b001011: +2.1,
  0b001100: +6.5,     0b001101: +11.0,    0b001110: +15.7,    0b001111: +20.6,
  0b010000: +25.6,    0b010001: +30.9,    0b010010: +36.4,    0b010011: +42.0,
  0b010100: +46.1,    0b010101: +50.2,    0b010110: +54.5,    0b010111: +58.8,
  0b011000: +63.3,    0b011001: +67.9,    0b011010: +72.6,    0b011011: +77.4,
  0b011100: +82.4,    0b011101: +87.5,    0b011110: +92.8,    0b011111: +98.2,
  0b100000: +102.5,   0b100001: +106.9,   0b100010: +111.4,   0b100011: +116.0,
  0b100100: +120.7,   0b100101: +125.5,   0b100110: +130.5,   0b100111: +135.5 }

# perform a reset
platform.reset()

# power up the temp sensor
platform.spi_write(TS_CONFIG, platform.spi_read(TS_CONFIG) & ~(1 << 6))

oldstatus = 1
# loop with threshold going upwards
for threshold in range(40):
    # set temp threshold
    platform.spi_write(TS_CONFIG, (platform.spi_read(TS_CONFIG) & ~0x3f) | (threshold &
0x3f))
    # read back status
    status = platform.spi_read(TS_STATUS)
    if status == 0:
        # first readback with actual temp over threshold
        flag = True
        print("chip temp is between %5.1f and %5.1f" % (TEMP_LUT[threshold-1],
TEMP_LUT[threshold]), flush=True)
        exit(0)

Setting clocks and system for RMII-MAC operation on port 3 and port 4
### 50 MHz on PLL1
platform.spi_write( 0x0010000A, 0x0A010941 )
platform.spi_write( 0x0010000A, 0x0A010940 )
for i in [ 3, 4 ]:
    # disable the port's IDIVi
    platform.spi_write( 0x0010000B+i,   (0x0001 | (0x0a << 24)))
    # set RMII_REF_CLK_x to TX_CLK_x
    platform.spi_write( 0x00100015+i*6, (0x0800 | (i*2) << 24))
    # set EXT_TX_CLK_x to PLL1
    platform.spi_write( 0x00100017+i*6, (0x0800 | (0xe << 24)))
```

Fig 71. **Code snippet for temperature measurement**

8. Safety

It is strongly recommended to poll the RAMPARERR register (address 0Ch and 0Dh). Bit flips in all SRAMs will be detected and reported over the RAM parity error indication (RAMPARERR). If any of these flags are found set during operation, the host must reset the switch.

The switch will stop forwarding frames if a memory parity error is detected. That means When a parity error occurs in the frame memory, the current frame is transmitted with TX_ER asserted. When a parity error occurs in any other memory all the frames are transmitted with TX_ER asserted, until reset is applied.

9. Measurements

This chapter contains empiric performance measurements beyond the specification data contained in the SJA1105P/Q/R/S product data sheet. It is for orientation purposes only.

9.1 Frame Latency

This is defined as the time needed for a frame to traverse the switch or a switch fabric in a cascaded setup. The measured numbers are best case values, as there was no other traffic. Congestion and queueing effects will worsen the values to some degree.

Two types of measurements are given here, using the edges of the RX_DV/CRS_DV signal taken from the port receiving the frame, and the TX_EN signal from the port on the transmitting end. Fig 72 provides the definition of the measurement points on the timing diagram.

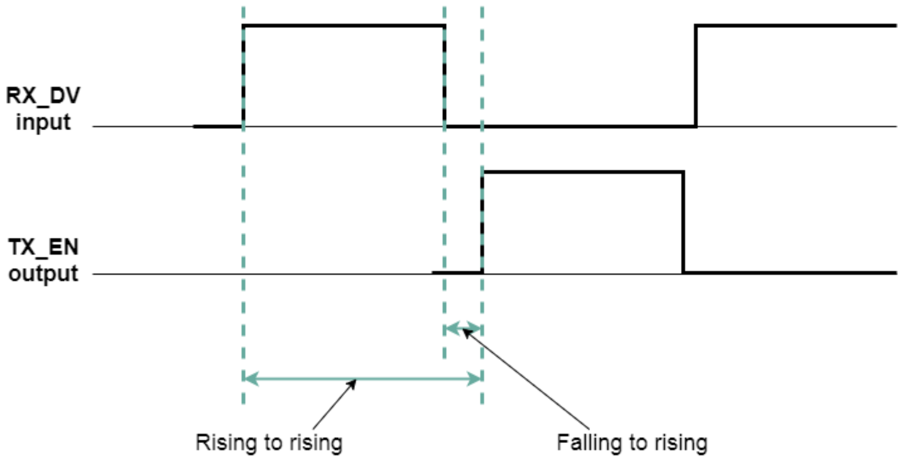


Fig 72. **Latency measurement point definitions**

For a single switch setup the measurements were performed for the three xMII modes: MII, RMII and RGMII, varying the link speed and the frame size.

Table 47. Frame latency in a single switch setup

	Rising to rising		Falling to rising	
	64B	**1518B**	**64B**	**1518B**
RGMII				
10 Mbps	70,6 µs	1,23 ms	18,7 µs	18,7 µs
100 Mbps	14,34 µs	130,6 µs	8,65 µs	8,65 µs
1 Gbps	2,89 µs	14,69 µs	2,5 µs	2,5 µs
RMII				
10 Mbps	76,41 µs	1,24 ms	17,32 µs	17,48 µs
100 Mbps	14,03 µs	132,2 µs	8,51 µs	8,51 µs
MII				

© NXP Semiconductors N.V. 2020. All rights reserved.

	Rising to rising		Falling to rising	
	64B	**1518B**	**64B**	**1518B**
10 Mbps	74,5 µs	1,23 ms	17,45 µs	17,45 µs
100 Mbps	14,24 µs	130,53 µs	8,51 µs	8,51 µs

For measuring latency on SGMII, a cascaded setup has been used with two SJA1105S connected with SGMII, and frames entering and leaving the switches via RMII. As with the single switch setup, latency was measured with the RMII handshake signals on port A1 and B0 in Fig 73.

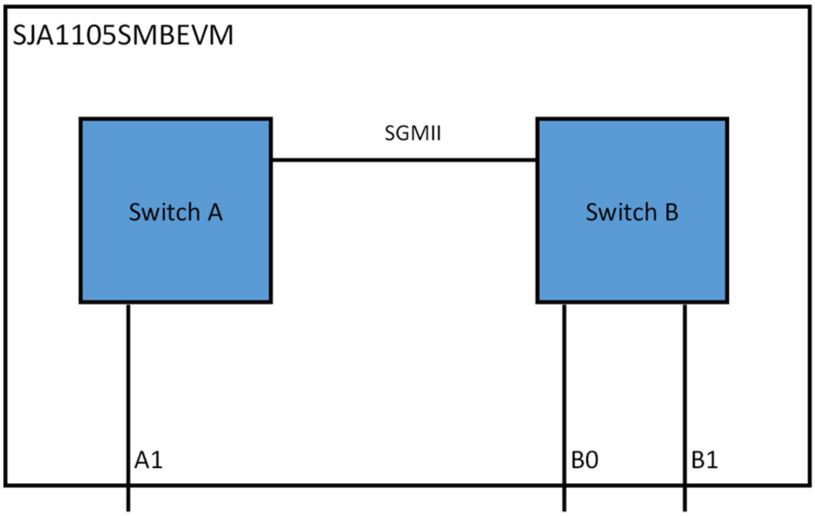


Fig 73. **Dual switch setup for SGMII latency measurements**

Table 48. Frame latency in a dual switch setup

	Rising to rising		Falling to rising	
	64B	**1518B**	**64B**	**1518B**
RMII+SGMII				
100/1000 Mbps	17.3µs	145,36µs	11.6µs	23.3µs
SGMII isolated	1.6µs	7.3µs	1.6µs	11.5µs

Isolated SGMII latency may be calculated by subtracting the RMII-to-RMII latencies from the combined RMII+SGMII latencies, and dividing by 2 for the average SGMII egress or ingress latency.

9.2 Packet drop caused by clock tolerance

During stress tests with link loads near 100% sometimes packet drops can be observed. This is a fundamental problem caused by different clock speeds, still perfectly within the specification. Consider the following setup:

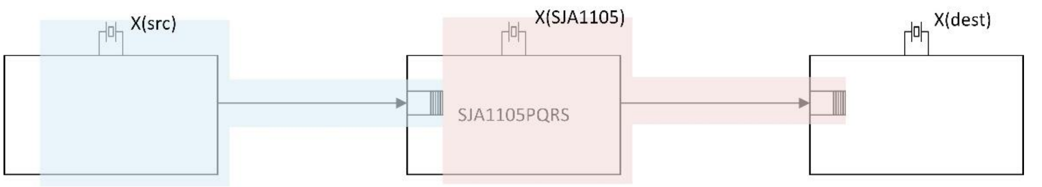


Fig 74. **Setup for explaining the effect of clock tolerances**

The SJA1105PQRS receives data at 100% link speed from a src node. The switch's receive process is controlled by the sending peer's clock X(src) – blue clock realm. Let's presume that in the SJA1105PQRS all received data has to be forwarded to the same egress port. The egress port has the same nominal speed as the ingress port[8]. The data rate on the egress port is controlled by the switch's clock X(SJA1105) – red clock realm.

Both crystals are within the specification of ±100ppm (±50ppm for RGMII). Let's also assume X(src) runs faster than X(SJA1105) within the allowed tolerance. This results in the switch receiving data faster than it can transmit – despite both links are working at 100%. For some time, the switch's buffering capability can camouflage this situation. But if the 100% load persists and the available buffers are filled up, packets inevitably get dropped.

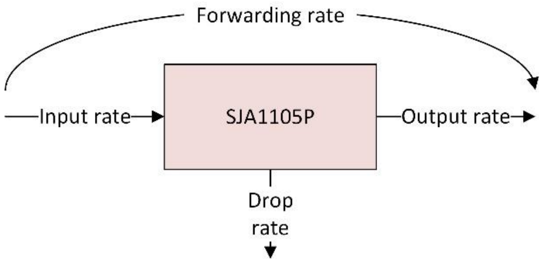
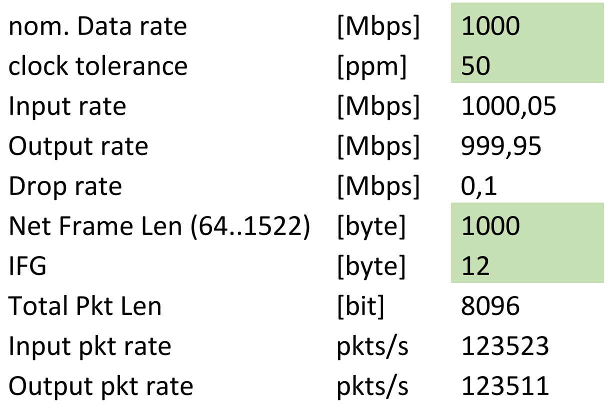


Fig 75. **Definitions of rates**

In a steady state (buffers full, so no longer effective), the drop rate (pkts/s) depends on the actual clock difference, the overall frame length and the interframe gap:

Example:



nom. Data rate	[Mbps]	1000
clock tolerance	[ppm]	50
Input rate	[Mbps]	1000,05
Output rate	[Mbps]	999,95
Drop rate	[Mbps]	0,1
Net Frame Len (64..1522)	[byte]	1000
IFG	[byte]	12
Total Pkt Len	[bit]	8096
Input pkt rate	pkts/s	123523
Output pkt rate	pkts/s	123511

[8] The speed-step-situation explained in 9.3.1 is also a source of packet drops, but has a different reason, and is not considered here.

drop rate	pkts/s	12

9.3 Throughput measurements

One key performance parameter for switches is throughput. Measurement data streams are made up of a sequence of ethernet packets with various size and various (temporal) distance, which in the end results in various source bandwidths of the data streams. These streams are piped through the switch, and the lost packets – if there are any - are counted.

The SJA1105PQRS has a “backplane bandwidth” which is significantly higher than the sum of the maximum port bandwidth of all 5 ports, so from the bandwidth perspective, there should not be any packet drops. However, there are some pitfalls in the measurement setup, which may be the reason for packet drops, and which may lead to false conclusions regarding the switch’s performance.

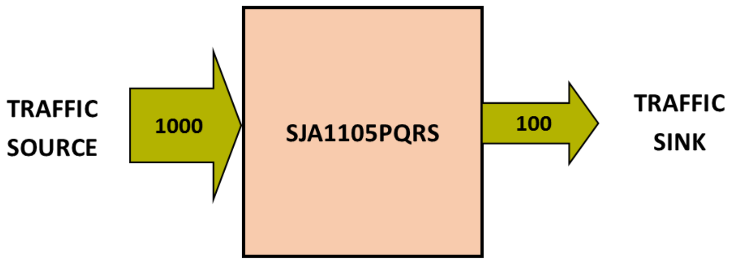


Fig 76. **Throughput measurement example**

9.3.1 Speed step configuration

Each link has a maximum data capacity, basically determined by the link speed. It should be clear, that you cannot expect 200Mbps throughput over a 100Mbps link. This sounds trivial, but once you have traffic merging and splitting, things get quickly complicated.

The simple setup in Fig 76 shows that measurement takes place at the source and the sink, so measurement always covers the whole path between source and sink. In this case the ingress and egress links of the switch are included.

In a branched network of links and switches the overall link capacity from source to sink is determined by the “thinnest link” regarding available bandwidth. This presumes, that there is no other traffic in the system during measurements. If there is, this traffic impacts the available capacity on a certain part of the overall link.

Link setups with a link speed configuration similar to Fig 76 have a speed step. Therefore, the source must be throttled to less than the smallest link speed (here: 100Mbps) to avoid systematic packet drops within the switch due to egress queue congestion or switch memory exhaustion.

9.3.2 Bursts

However, despite the correct source data, there may still be packet drops, due to the way traffic generation tools calculate the nominal output data rate: Ideally, the packets are sent out with an equidistant interpacket gap, and the gap depends on the (uniform) packet length, the required data rate and the actual link speed. Equally spaced packets give the switch enough time to egress the received packet to the “slow” port, before the next packet arrives.

HW supported, high performance tools, like Spirent Automotive C50[9] can generate equidistant packets very well, so the actual link usage (in % of the link capacity) stays the same, even for time intervals of one frame and one gap.

Tools like iperf[10] or Ostinato[11], which are basically running on standard PC hardware, are susceptible to performance jitter due to resource competition, like for example scheduling, system load, memory, interrupts, etc. This disturbs more or less often the timely sending of the packets, so the gaps must then be compensated by bursts of packets to catch up.

Fig 77 shows a iperf experiment over a length of 10sec: the number of received packets within a 1ms wide window are shown as a single dot. Most of the time, there are 8 or 9 packets per 1-ms, which can be seen in two horizontal lines. Refer to a magnification on the upper right hand side. But in some 1ms windows, there are up to 80 packets and down to 0 packets. In other words, they send out bursts of packets at link speed without any or too small a gap.

The iperf client (packet source) has been running on a well-powered PC (i7, lots of memory, no significant other computational load), and the packets have been received and counted with the Spirent (HW assisted). When using a microcontroller as an iperf client, bursting may be even worse.

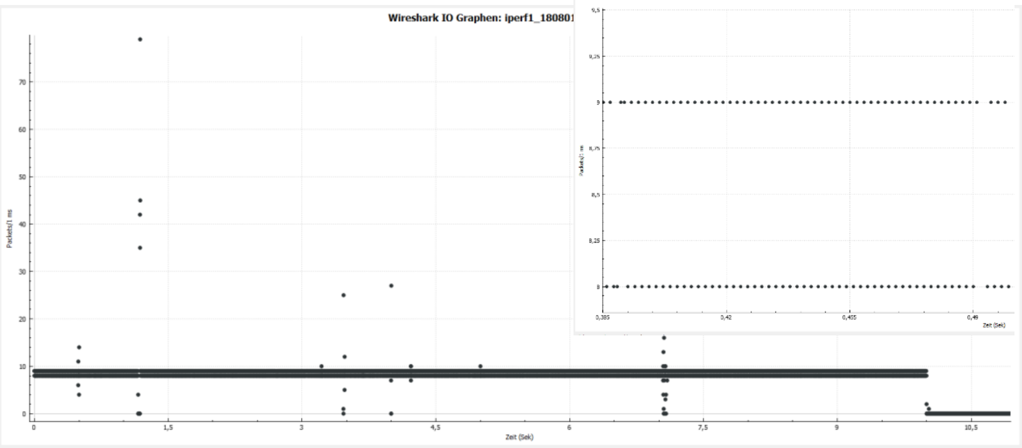


Fig 77. **Packet-to-packet distance histogram of an iperf generated UDP stream[12]**

As counting is normally done in software (and not in hardware as in the Spirent tool), the iperf server also influences measurements, because if the iperf server process is blocked

[9] https://www.spirent.com/-/media/Datasheets/Automotive/DS_Spirent_Automotive_C50.pdf

[10] https://iperf.fr/

[11] https://ostinato.org/

[12] This type of diagrams can be created with wireshark, having loaded the captured packets from the iperf source side, and calling “statistics -> IO graph”, with “interval” set to 1ms.

for too long, packets may be dropped in the IP stack, though the device under test will be blamed.

In a speed step setup like in Fig 76, packet preservation mainly depends on the switch’s buffer capability: Once the buffer is full and the packets still come in faster than they leave the switch, packet drops are unavoidable. This is not a matter of the switch’s performance, but a problem with the measurement setup.

Packet drops are influenced by two switch resources: frame memory and egress queue size. For big packets, frame memory exhaustion is the main reason for packet drops, while for small packets, it is rather the number of available egress queue slots. The available resources for a certain iperf stream depend on the resources actually allocated to the stream’s PCP and the memory and egress queue slots assigned to the egress port used by that stream (L2 Forwarding table’s PART_SPC for each PCP and MAC configuration table’s BASE and TOP for each PCP).

9.3.3 iperf measurement with TCP and UDP

TCP is the default protocol to be used for iperf measurements. TCP is designed to deal with lossy links and remedies link problems with concepts like resending lost packets and throttling the source. Therefore, you will not directly see packet losses with TCP, but just a “slower” overall link. The real problems are covered up, but the underlying root causes are still the same. In the example in Fig 76 this would be for example a “measured” overall link speed between source and sink of only 30Mbps.

The recommended iperf call parameters for TCP are:

```
Server side:      iperf -s -i 1 -e
Client side:      iperf -c <serv-IP> -i 1 -b <bitrate> -e
bitrate:          nominal data rate for throttled sending on client
                  side, e.g. 80M for 80Mbps

Use -e for enhanced output, showing the details of the iperf session
```

Fig 78. **Recommended iperf call parameters for TCP**

For UDP, use the following calls:

```
Server side:      iperf -u -s -i 1
Client side:      iperf -u -c <serv-IP> -i 1 -b <bitrate> -l <payload>
bitrate:          nominal data rate for throttled sending on client
                  side, e.g. 80M for 80Mbps
payload:          UDP payload size. Should be <1500B to prevent
                  fragmentation. Add 46Byte for UDP, IP, L2 header.
```

Fig 79. **Recommended iperf call parameters for UDP**

10. Issues and Workarounds

10.1 RUNT Frames

On the SJA1105PQRS a short, invalid Ethernet frame between 1-5 bytes of Ethernet L2 (L1 overhead should be received correct – if the error occurs in the L1 overhead the issue is not triggered) length can in rare cases lead to a frame memory leak.

The situation occurs under the following conditions. At least 3 ports have received a valid nominal L2 Frame at the same time and the reception of the frames finishes in the same clock cycle. On one of these ports which run at 1Gbps speed a RUNT frame (either RX_ER asserted or RX_DV deasserted) on the first 5 bytes of L1 payload (following the SOF) is received. The Switch will in rare situations overwrite the previously received healthy frame and forwards it as an illegal RUNT frame. The memory associated with the healthy frame is not released and more memory can be released than was allocated to store the RUNT frame. In case this condition is triggered too frequently and the switch has leaked all available memory it will indicate a memory parity error and stop forwarding frames.

The cause for the issue is a RUNT frame between 1 and 5 bytes received in special conditions on a **Gigabit speed port only.** Such an unusual RUNT frame is typically only caused by an aborted transmission (either RX_ER asserted or RX_DV deasserted). This happens if a link is lost or the PHY is prematurely powered off. These situations must be avoided. If RGMII is used in a MAC-MAC connection without PHYs such a scenario does typically not occur. **The issue is never triggered at 10Mbps or 100Mbps speed**.

If a RUNT frame is received from any port of the Switch or the MII error counter (N_MIIERR) or the runt error counter (N_RUNT) indicates a value larger than a threshold value, which can be defined based on the frequency the counters are read, it is advised to reset the switch.

10.2 JTAG Boundary Scan

All devices with date code 1743 and onwards have fully operational JTAG Boundary Scan.

If an xMII port is unused and its VDDIO_MIIx supply is connected to ground (as in section 2.5.1.3), the unused IO pins may generate random data during boundary scan. Data from unused pins should be masked, so they don’t lead to fail messages. Note that this is not the case if VDDIO_MIIx supply is present.

11. Migration from SJA1105(T)

11.1 Assumptions

Typical Configuration

- Static L2 Lookup and VLAN configuration
- gPTP operation
- Credit-based shaping
- 100BASE-T1 PHYs over MII/RMII

Summary:

- Few changes necessary (minimal or covered by provided tooling)
- Efforts for adjustment: ~1 week
- New optional features allow further improvements (e.g. more counters)

11.2 Static Configuration

Minimal changes, updated configuration generator provided by NXP

- L2 lookup configuration is extended due to TCAM
- VLAN Lookup table is unchanged
- Credit-based shaping can be configured statically now, dynamic option still exists
- xMII Mode Parameters unchanged

11.3 Programming Interface

Core

- Changed register addresses
- gPTP: option to sync cascaded switches
- gPTP: egress timestamp update information and timestamp are split into two registers
- Changed layout of the L2 Lookup Table in reconfiguration

Clock Generation Unit

- Configuration unchanged for MII/RMII
- Simplified configuration for RGMII
- Additional Configuration for SGMII if required

Auxiliary configuration Unit

- 95 % unchanged
- Additional registers for internal delay lines (only for RGMII)

12. Legal information

12.1 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.

12.2 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. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

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 and its suppliers accept 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.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

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.

12.3 Licenses

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

12.4 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>

12.5 Trademarks

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

<Name> — is a trademark of NXP Semiconductors N.V.

13. List of figures

Fig 1. 1.2 V Power Rail Topology with Filtering 5
Fig 2. Low-pass power filter for VDDIO_MIIx..............6
Fig 3. Low-pass power filter for VDDA_SGMII............6
Fig 4. Low-pass power filter for VDDIO_HOST and VDDIO_CLO 7
Fig 5. Oscillator in Crystal / Master mode 9
Fig 6. Oscillator in External / Slave mode 10
Fig 7. Clocking in cascaded switches 11
Fig 8. Multi-point clock output topology.....................11
Fig 9. Reset input...13
Fig 10. Reset timing 15
Fig 11. Swapping SGMII interface polarity..................16
Fig 12. SPI Interface 18
Fig 13. MII with crystals at SJA1105PQRS and TJA1102...19
Fig 14. MII with crystal at SJA1105PQRS and digital clock supply to TJA1102 19
Fig 15. RMII with crystal at SJA1105PQRS and digital clock supply to TJA1102 20
Fig 16. Multiple microcontroller hub example.............21
Fig 17. SPI register read example..............................22
Fig 18. Generic CGU Overview 25
Fig 19. CGU in MII-MAC configuration........................29
Fig 20. CGU in MII-PHY configuration 31
Fig 21. CGU in RMII-MAC configuration 35
Fig 22. RMII with SJA1105PQRS in MAC mode.........36
Fig 23. RMII-MAC configuration example 37
Fig 24. CGU in RMII-PHY configuration......................38
Fig 25. RMII with SJA1105PQRS in PHY mode 38
Fig 26. RMII with SJA1105PQRS in PHY mode and additional oscillator for REF_CLK 40
Fig 27. RMII with SJA1105PQRS and a SOC without REF_CLK in MAC mode 41
Fig 28. CGU in RGMII 1Gbps configuration................41
Fig 29. RGMII timing delay 45
Fig 30. Options to implement RGMII clock delay 46
Fig 31. TDL Configuration Code 46
Fig 32. Delay line setting in degree............................47
Fig 33. RGMII delay line settings 47
Fig 34. TDL input clock transition phase 48
Fig 35. MAC-Host link with SGMII..............................50
Fig 36. Cascaded switches with SGMII.......................51
Fig 37. SGMII connected to modern SFP-Module for copper..51
Fig 38. Config_Reg SGMII...53
Fig 39. LOOPBACK configuration options for the SGMII port...54
Fig 40. SGMII Eye Diagram – Recommended Settings 56
Fig 41. L2 Lookup Table operation.............................59
Fig 42. Static L2 Entry...60
Fig 43. Output a L2 address lookup table entry 62
Fig 44. Example for generating an Ingress Timestamp at the HOST port 66
Fig 45. Example CBS Operation 67
Fig 46. Port / priority to policer mapping......................69
Fig 47. Ingress to Egress PCP resolution and queue selection 71
Fig 48. Transmission selection in 802.1Qbv using TAS 72
Fig 49. Gate states over time in the example configuration..73
Fig 50. TAS Schedule Computation from 802.1Qbv Gate List..77
Fig 51. Definition of BAG and JITTER for stream rate policing 78
Fig 52. Blacklisting a MAC address............................82
Fig 53. Enforce Port Example 83
Fig 54. Ingress mirroring of frames received on port 2 to port 3 without tagging..................................85
Fig 55. Ingress mirroring of frames received on port 2 to port 3 with additional tagging of mirrored frames 85
Fig 56. Egress mirroring of frames received on port 2 to port 3 with additional tagging of mirrored frames 85
Fig 57. VLAN entry configuration example 1...............86
Fig 58. VLAN entry configuration example 2...............86
Fig 59. Ingress policing rule example 1.......................87
Fig 60. Ingress policing rule with broadcast policing example 2..87
Fig 61. L2 Forwarding table configuration with no remapping 88
Fig 62. L2 Forwarding table configuration with reverse mapping of PCP values.................................88
Fig 63. General Parameter Table...............................88
Fig 64. Example Stream..89
Fig 65. Example configuration settings for stream based policing 89
Fig 66. Dissected SYNC frame on the HOST_PORT..91

© NXP Semiconductors N.V. 2020. All rights reserved.

Fig 67. Dissected FOLLOW_UP frame on the HOST_PORT 91
Fig 68. PTP_CLK toggle output 92
Fig 69. CASSYNC control 93
Fig 70. Cascading example 94
Fig 71. Code snippet for temperature measurement... 96
Fig 72. Latency measurement point definitions....... 97
Fig 73. Dual switch setup for SGMII latency measurements 98
Fig 74. Setup for explaining the effect of clock tolerances 99
Fig 75. Definitions of rates 99
Fig 76. Throughput measurement example 100
Fig 77. Packet-to-packet distance histogram of an iperf generated UDP stream 101
Fig 78. Recommended iperf call parameters for TCP 102
Fig 79. Recommended iperf call parameters for UDP 102

14. List of tables

Table 1. Switch variants 3
Table 2. Voltage requirement overview 4
Table 3. Clock Pins and purpose 8
Table 4. Recommended Crystal Parameters 9
Table 5. Recommended external clock parameters 10
Table 6. Recommended external clock parameters 12
Table 7. Reset pin properties 12
Table 8. Pin behavior during reset 13
Table 9. RGMII Pin behavior under power-fail 14
Table 10. SGMII Pin behavior under power-fail 14
Table 11. SPI pins 17
Table 12. Reset control register (addr 100440h) 21
Table 13. CGU register overview 24
Table 14. PLL Operating Modes 26
Table 15. Clock sink registers 1 to 30 (addr. 100013h to 100030h) 28
Table 16. Source to sink mapping 30
Table 17. Register Settings for 100 Mbps MII-MAC 30
Table 18. Source to sink mapping 31
Table 19. Register Settings for 100 Mbps MII-PHY 32
Table 20. Source to sink mapping 33
Table 21. Register Settings for 10 Mbps MII-PHY 33
Table 22. Source to sink mapping 36
Table 23. Register Settings for 100 Mbps RMII-MAC 36
Table 24. Source to sink mapping 39
Table 25. Register Settings for 100 Mbps RMII-PHY 39
Table 26. Source to sink mapping 42
Table 27. Register Settings for 1Gbps RGMII 42
Table 28. Source to sink mapping 43
Table 29. Register Settings for 100Mbps RGMII 43
Table 30. Source to sink mapping 44
Table 31. Register Settings for 10 Mbps RGMII 44
Table 32. Supported Slew Rates 49
Table 33. Recommended Slew Rates 49
Table 34. Advertised Capabilities 53
Table 35. SGMII attenuation coding 54
Table 36. SGMII Level Settings 55
Table 37. Disable switch’s output clock to support Sleeping Peer 57
Table 38. Fields to be configured in an entry of the Schedule Table for TAS operation 73
Table 39. Example configuration of Schedule Table 73
Table 40. Fields to be configured in the Schedule Entry Points Parameters for TAS operation 74
Table 41. Fields to be configured in the Schedule Parameters for TAS operation 74
Table 42. Fields to be configured in an entry of the Schedule Entry Points Table for TAS operation 74
Table 43. Frame learning and frame dropping behavior . 81
Table 44. Static configuration of 1-step and 2-step SYNC frame handling 89
Table 45. L2 management table example for cascaded switches 94
Table 46. Static config items for cascaded switches 94
Table 47. Frame latency in a single switch setup 97
Table 48. Frame latency in a dual switch setup 98

15. Contents

1. Introduction.. 3
1.1 SJA1105PQRS switch family 3
1.2 Other documents 3
2. Schematic and Layout....................................... 4
2.1 Power Supply 4
2.1.1 Voltages 4
2.2 Filtering... 5
2.2.1 Core and Auxiliary Supply 5
2.2.2 MII/RMII/RGMII/SGMII Supply.......................... 6
2.2.3 Unused Ports.. 7
2.2.4 HOST and CLK_OUT Supply 7
2.2.5 Layout Recommendations................................ 7
2.2.6 Power consumption 8
2.3 Clocking.. 8
2.3.1 Reference Clock Design Requirements............ 8
2.3.1.1 Crystal Mode 8
2.3.1.2 Crystal Layout... 9
2.3.2 Clock Mode 10
2.3.3 Cascaded clocking scheme.............................11
2.3.4 Clock output (CLK_OUT).................................11
2.4 Reset and Power-on Sequence Design............12
2.4.1 Reset 12
2.4.2 Pin behavior during Reset 13
2.4.3 Pin behavior when supply is missing 13
2.4.3.1 RGMII pins 14
2.4.3.2 SGMII pins...14
2.4.4 Reset and Power-On Sequence.......................14
2.5 Interface Design 15
2.5.1 Ethernet Interface 15
2.5.1.1 MII/RMII/RGMII layout guideline......................15
2.5.1.2 SGMII layout guideline 15
2.5.1.3 Unused xMII Interface 16
2.5.1.4 RMII and MII-lite 17
2.5.2 SPI Interface..17
2.5.3 JTAG Interface 18
3. Recommended System Solutions 18
3.1 Interoperability with TJA1101 and TJA1102(S) 18
3.1.1 MII Mode with XTAL 18
3.1.2 MII Mode with digital clock (CLK_OUT)............19
3.1.3 RMII Mode with digital clock (REF_CLK)..........20
3.2 PHY-less connection to a microcontroller’s MAC20
3.3 RMII Interoperability Remarks 21
4. Infrastructure Modules.....................................21
4.1 RCU...21
5. SPI communication protocol 22
5.1 SPI waveform 22
5.2 SPI write operation length 23
6. Interface Configuration 23
6.1 MII/RMII/RGMII Configuration 23
6.1.1 Selectable Clock Sources.................................25
6.1.2 PLL Frequency Selection 26
6.1.3 Configurable Clock Sinks 28
6.1.4 MII-MAC Mode 100 / 10 Mbps..........................29
6.1.4.1 Sources 30

6.1.4.2 Sinks..30
6.1.5 MII-PHY Mode 100 Mbps 31
6.1.5.1 Sources 31
6.1.5.2 Sinks..31
6.1.6 MII-PHY Mode 10 Mbps 33
6.1.6.1 Sources 33
6.1.6.2 Sinks..33
6.1.7 RMII-MAC Mode 100 / 10 Mbps 34
6.1.7.1 Sources 36
6.1.7.2 Sinks..36
6.1.7.3 Configuration example......................................37
6.1.8 RMII PHY Mode 100 / 10 Mbps........................37
6.1.8.1 Sources 39
6.1.8.2 Sinks..39
6.1.9 RMII-PHY Mode with a MAC without REF_CLK39
6.1.9.1 RMII-PHY Mode and external oscillator............40
6.1.9.2 SJA1105PQRS used as a PHY and providing REF_CLK 40
6.1.10 GMII Mode 1 Gbps 41
6.1.10.1 Sources 42
6.1.10.2 Sinks..42
6.1.11 RGMII Mode 100 Mbps 42
6.1.11.1 Sources 42
6.1.11.2 Sinks..43
6.1.12 RGMII Mode 10 Mbps 43
6.1.12.1 Sources 43
6.1.12.2 Sinks..44
6.1.13 RGMII clock delay 44
6.1.13.1 RGMII clock delay concept...............................44
6.1.13.2 RGMII TDL Configuration 46
6.1.13.3 Frequency Transitions 48
6.1.13.4 Further hints 48
6.1.14 Slew Rate Settings 48
6.2 SGMII 49
6.2.1 Initialization...50
6.2.2 Disabled Auto-Negotiation................................50
6.2.3 Auto-Negotiation...50
6.2.3.1 SGMII Auto-Negotiation....................................50
6.2.3.2 SFP Module..51
6.2.3.3 PHY-MAC Roles...52
6.2.3.4 1000Base-X Clause 37 Auto-Negotiation 52
6.2.3.5 Capabilities...52
6.2.4 Loopback mode..53
6.2.5 SGMII Level Setting..54
6.2.5.1 SGMII output level calculation 54
6.2.5.2 Maximum Output amplitude..............................55
6.2.5.3 Minimum Output Amplitude 55
6.2.6 Recommended Values 55
6.3 Low power sleep operation...............................56
6.3.1 Sleeping microcontroller 56
6.3.2 Sleeping switch...58
7. Switch Core Configuration 58
7.1 Static L2 Configuration Entries (TCAM)............58
7.1.1 IVL and SVL 60
7.1.2 Mirroring 60
7.1.3 Retagging 60

7.1.4 Trapping 61
7.2 Dynamic Reconfiguration 61
7.2.1 Reading a L2 Lookup Table Entry 61
7.2.2 Writing a L2 Lookup Table Entry 62
7.3 Timestamping and AVB..........62
7.3.1 Frame Trapping..........62
7.3.1.1 Disabling trapping..........63
7.3.2 Sending management frames 63
7.3.3 1-step and 2-step SYNC frames..........64
7.3.4 Reconstruction for META frames..........64
7.3.5 Ingress Timestamp towards the Host Interface 65
7.3.6 Credit Based Shaper Parameters..........66
7.3.7 Policing..........68
7.3.8 Priority Regeneration and Priority to Queue Mapping 69
7.3.9 Timestamping in Cascaded Switch designs 71
7.4 TSN – Time Sensitive Networking..........71
7.4.1 Time Aware Shaper (TAS) – 802.1Qbv..........72
7.4.1.1 Introduction TAS..........72
7.4.1.2 Static Configuration of TAS Schedule Table 72
7.4.1.3 Static Configuration of TAS Parameters..........74
7.4.1.4 Rate Ratio Correction and Synchronization......75
7.4.1.5 Starting/Stopping TAS Scheduling 75
7.4.1.6 802.1Qbv Pre-Standard Conformance 76
7.4.2 Per Stream Policing..........77
7.5 Security 79
7.5.1 Port Reachability 79
7.5.2 VLAN Configuration..........79
7.5.2.1 Single tag and general VLAN behavior..........79
7.5.2.2 Double tag detection and blocking..........79
7.5.3 Quality of Service (QoS)..........80
7.5.3.1 Priority Handling and Regeneration..........80
7.5.3.2 Policing and Broadcast Storm Prevention 80
7.5.3.3 Egress Shaping (Credit Based Shaper)..........80
7.5.4 Address Learning 80
7.5.4.1 Dynamic Address Learning 80
7.5.4.2 Dynamic Address Learning Limitation 81
7.5.4.3 Single Shot Learning 82
7.5.5 MAC Address Blacklisting 82
7.5.6 MAC Address Whitelisting..........82
7.5.7 802.1X Port-Based Authentication..........83
7.6 Configuration Examples 84
7.6.1 Mirroring 84
7.6.2 VLANs 85
7.6.3 Ingress Policing 86
7.6.4 Priority Remapping..........87
7.6.5 Per-Stream Policing..........88
7.6.6 PTP 89
7.6.6.1 1-Step setup..........90
7.6.6.2 2-Step setup..........90
7.6.6.3 PTP output clock 91
7.6.7 Cascading Switches 92
7.6.7.1 Ethernet links for cascading 92
7.6.7.2 Clocks and Timestamp counter synchronization92
7.6.7.3 Trapping in a cascading setup..........93
7.6.7.4 Sending management frames 94
7.6.7.5 Static config stream..........94

7.6.8 Temperature sensor 95
8. Safety..96
9. Measurements...97
9.1 Frame Latency...97
9.2 Packet drop caused by clock tolerance 98
9.3 Throughput measurements 100
9.3.1 Speed step configuration................................100
9.3.2 Bursts 101
9.3.3 iperf measurement with TCP and UDP...........102
10. Issues and Workarounds 103
10.1 RUNT Frames 103
10.2 JTAG Boundary Scan....................................103
11. Migration from SJA1105(T) 103
11.1 Assumptions..103
11.2 Static Configuration 104
11.3 Programming Interface 104
12. Legal information..105
12.1 Definitions..105
12.2 Disclaimers..105
12.3 Licenses 105
12.4 Patents 105
12.5 Trademarks 105
13. List of figures 106
14. List of tables 108
15. Contents 109

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

© NXP Semiconductors N.V. 2020. All rights reserved.

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

Date of release: 23 July 2020

Document identifier: AH1704