life.augmented

STWBC2-HP

Datasheet

Digital controller for wireless battery charger transmitters

Product status link
STWBC2 -HP

Product label
SUSTAINABLE TECHNOLOGY

Table 1. Ordering information

Order code	Package	Packaging
STWBC2-HP	VFQFPN68	Tape & Reel

Features

- Digital controller for Qi certified wireless power transmitters
  - Compliant with WPC 1.3
  - Power Class 0 BPP (5 W) and EPP (15 W)
  - Power Tx design topologies MP-A2 and MP-A22
  - Proprietary ST Super Charge extension for high power charging
- SiP with uC and front end device:
  - ARM 32-bit Cortex™-M0+ CPU, frequency up to 64 MHz
  - 3x half-bridge drivers
  - Embedded 3.6 V / 5 V DC-DC
  - 3.3 V and 1.8 V LDOs
  - 6 V to 9 V voltage doubler
  - Qi FSK programmable modulator
  - Integrated current, voltage and phase demodulators
  - Integrated current and voltage sensors.
- Support for half- and full-bridge topologies with input DC-DC
  - Single and multi-coil topologies
  - Support for limited power sources such as 5 V 500 mA USB
- VIN operative range: 4.1 V to 24 V
- USB physical interfaces
  - USB power delivery
- Communication interfaces
  - UART
  - SPI (up to 28 Mbit/s)
  - I2C (up to 1 Mbit/s)
  - Up to 8x GPIOs
- Peripherals
  - 8-channel 12-bit 0.5 us ADC
  - Low side differential current sensor
  - Q-Factor driver
- Memory
  - 128 Kbytes of Flash memory with ECC
  - 32 Kbytes of SRAM with HW parity check
- Development support: serial wire debug (SWD)
- Operating temperature: -40°C up to 125°C.
- Package: VFQFPN68L 8x8 mm pitch 0.4
- 96-bit unique ID

Description

The STWBC2 -HP is a digital controller specifically dedicated to design Qi-certified Wireless Power TX applications. The STWBC2-HP is a SiP including an STM32™ microcontroller and an application-specific front-end die.

Specifically, the STM32™ microcontroller embedded into the STWBC2-HP is the STM32G071.

The STWBC2-HP is capable of driving the DC-DC stage and the half- or full-bridge inverter stage of a generic Qi wireless battery charging TX. It generates and controls the relevant PWM signals by means of a PWM machine capable of 1.47 ns resolution. In order to achieve that, the front-end die includes a 40 MHz PLL and a 17-step DLL.

The STWBC2-HP front end is capable of working from any DC voltage in the range of 4.1 V to 24 V and embeds:

- 3x half-bridge drivers capable of driving both the FB inverter or the DC-DC MOSFET
- 2x PWM outputs for external gate drivers
- 2x LED / speaker drivers
- 3.6 V / 5 V monolithic buck DC-DC to supply the analog portion of the die
- 3.3 V LDO to supply the STM32
- 1.8 V LDO to supply the core
- 6 V to 9 V voltage doubler to supply the gate drivers
- Resonant tank current, voltage and phase sense circuitry and Q-factor driver
- USB D+/D- external interface pins for USB PD

The Front End die also includes specific registers and an SPI interface with the STM32, in combination with an output mux for analog signals to the STM32 ADC, and GPIOs for digital signals to the STM32 core.

The STM32 microcontroller embeds an ARM Cortex™-M0+ CPU with up to 128 k Flash memory and up to 32 k of SRAM with parity check, with a clock frequency up to 64 MHz. It also embeds a 12-bit, 0.5 µs ADC and provides UART, I2C, SPI and GPIOs

The STM32G071 embedded in STWBC2-HP also provides two 12-bit DACs and USB-PD controller.

1 Block diagrams

1.1 STWBC2-HP architecture

The following image illustrates the overall system blocks implemented in the STWBC2-HP architecture.

The STWBC2-HP is a system-in-package combining an STM32G071 MCU die with a Wireless Power specific die which integrates several application-specific functions.

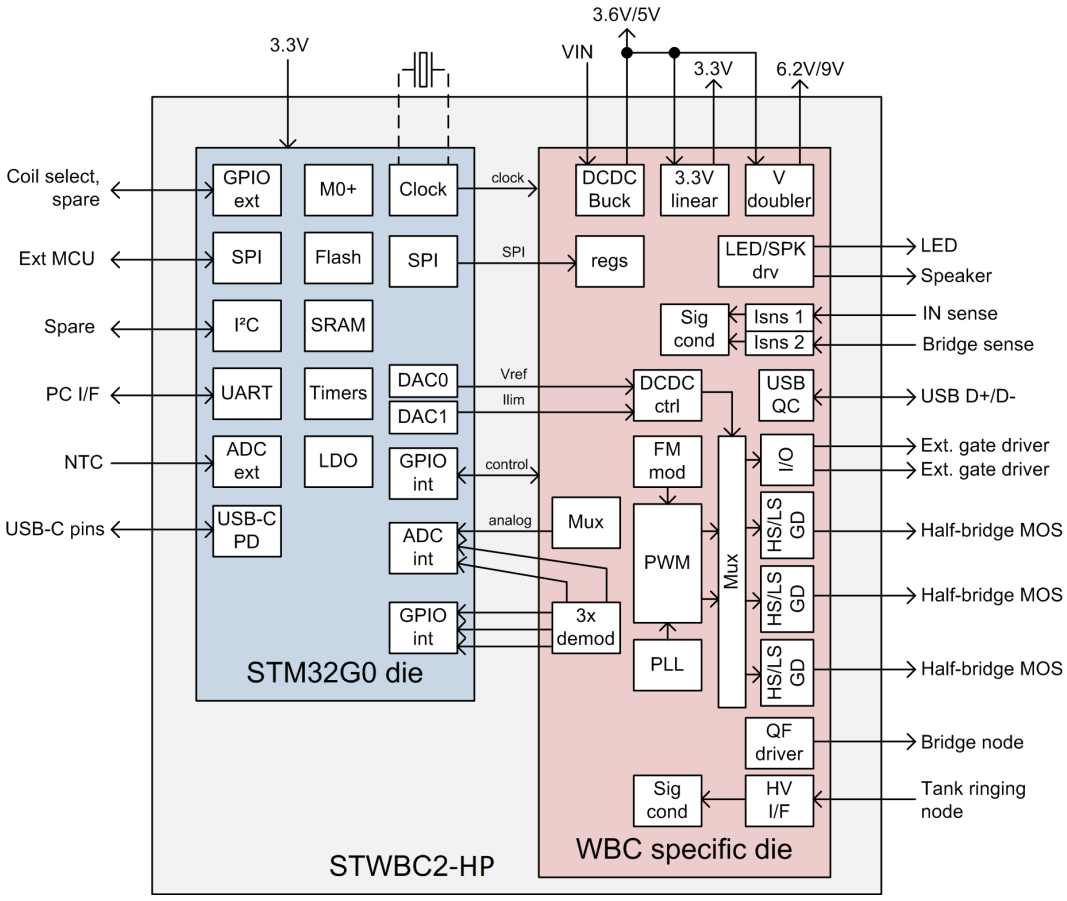
The STWBC2-HP supports both half-bridge and full-bridge coil driver topologies, together with a digital DC-DC controller, which regulates the input voltage of the bridge in order to control the amount of power transmitted to the receiver.

The digital controller also regulates the inverter operating frequency and duty cycle to further adjust the amount of power increasing the overall efficiency at light load.

It implements the WPC Qi protocol, including Foreign Object Detection (FOD) extensions. An accurate Q-factor measurement provides enhanced FOD.

The STWBC2-HP is able to manage both WPC Baseline Power Profile (up to 5 W) and Extended Power Profile (EPP).

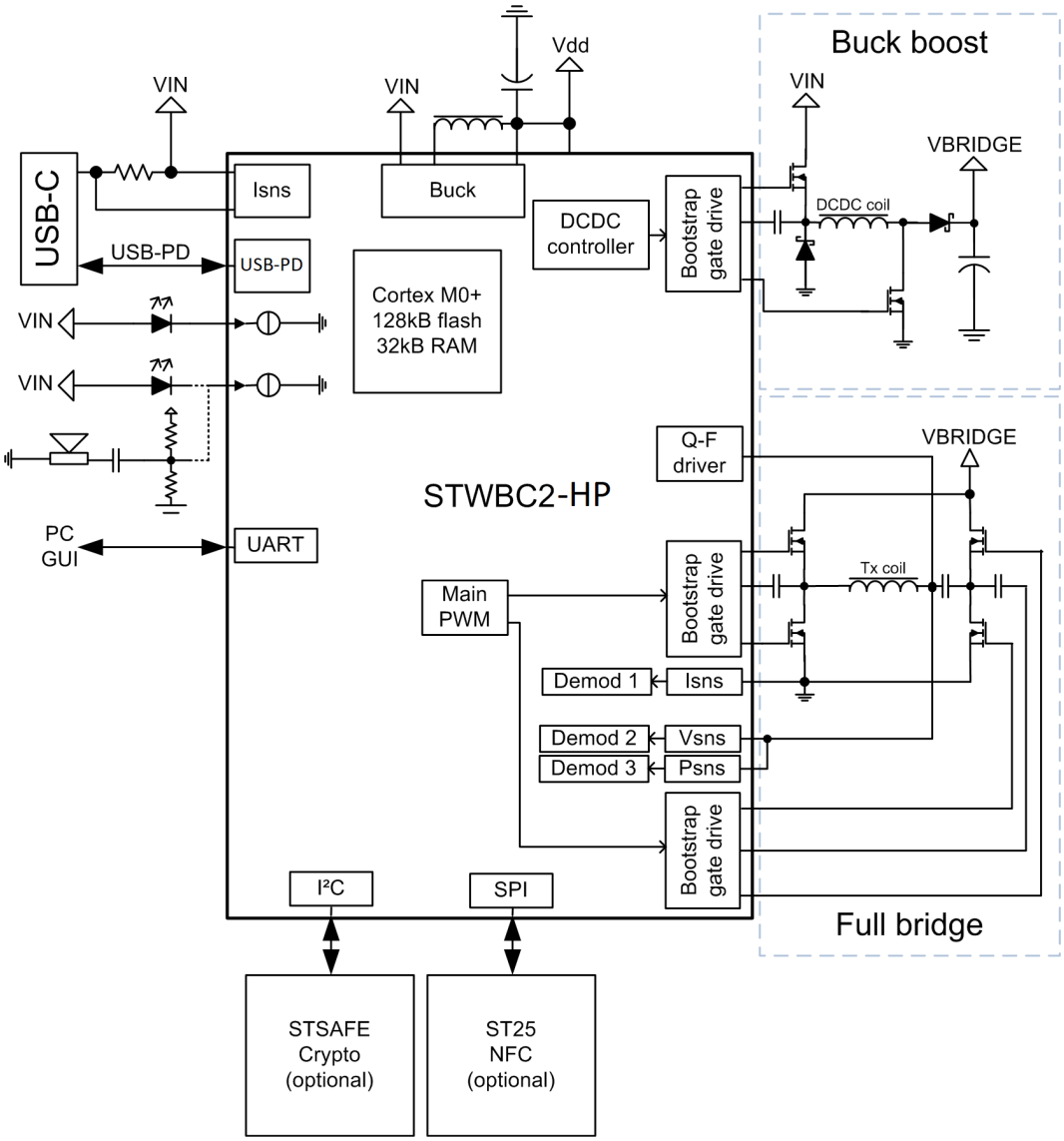
Figure 1. STWBC2-HP device architecture



1.2 Typical application

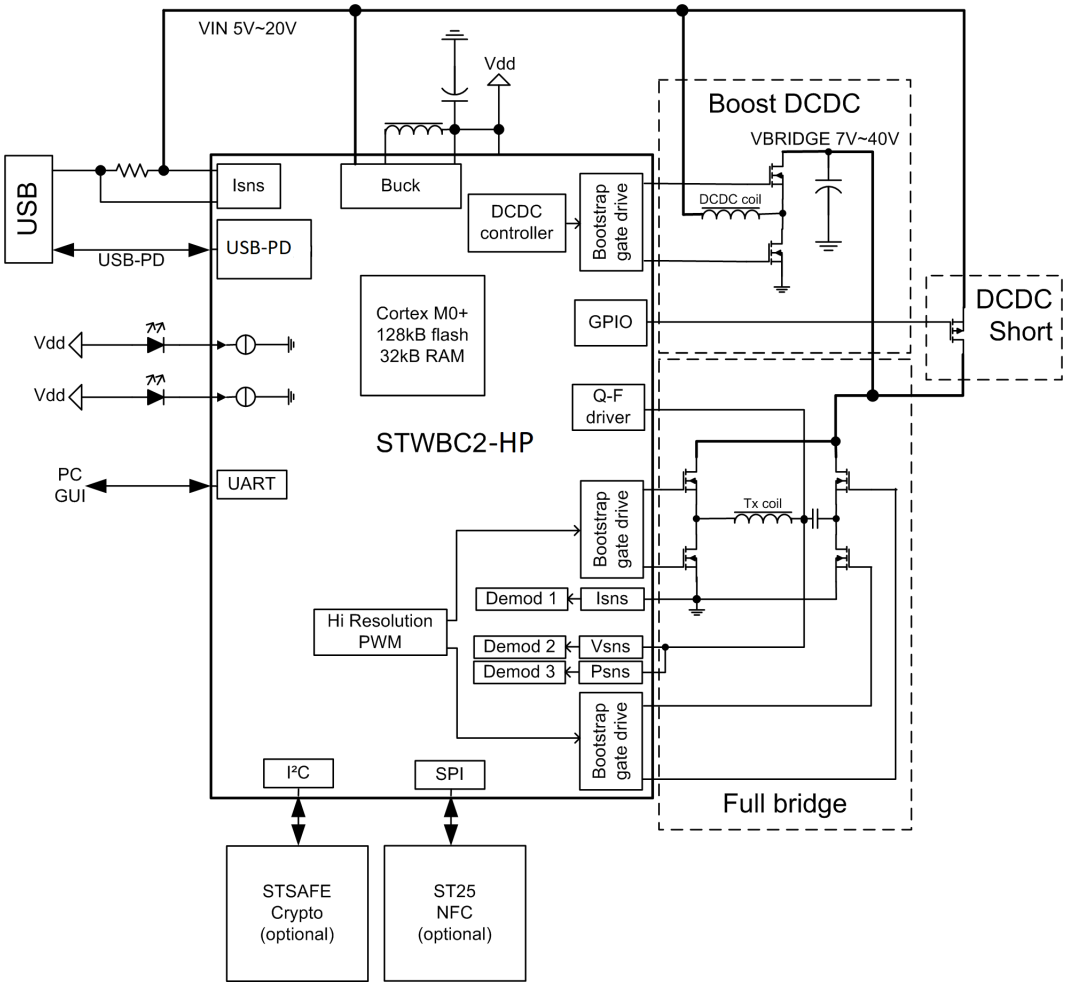
1.2.1 Qi single coil, fixed frequency full-bridge, USB powered

Figure 2. **Qi single coil, fixed frequency full-bridge, USB powered**



1.2.2 Qi single coil, variable frequency full-bridge, USB powered

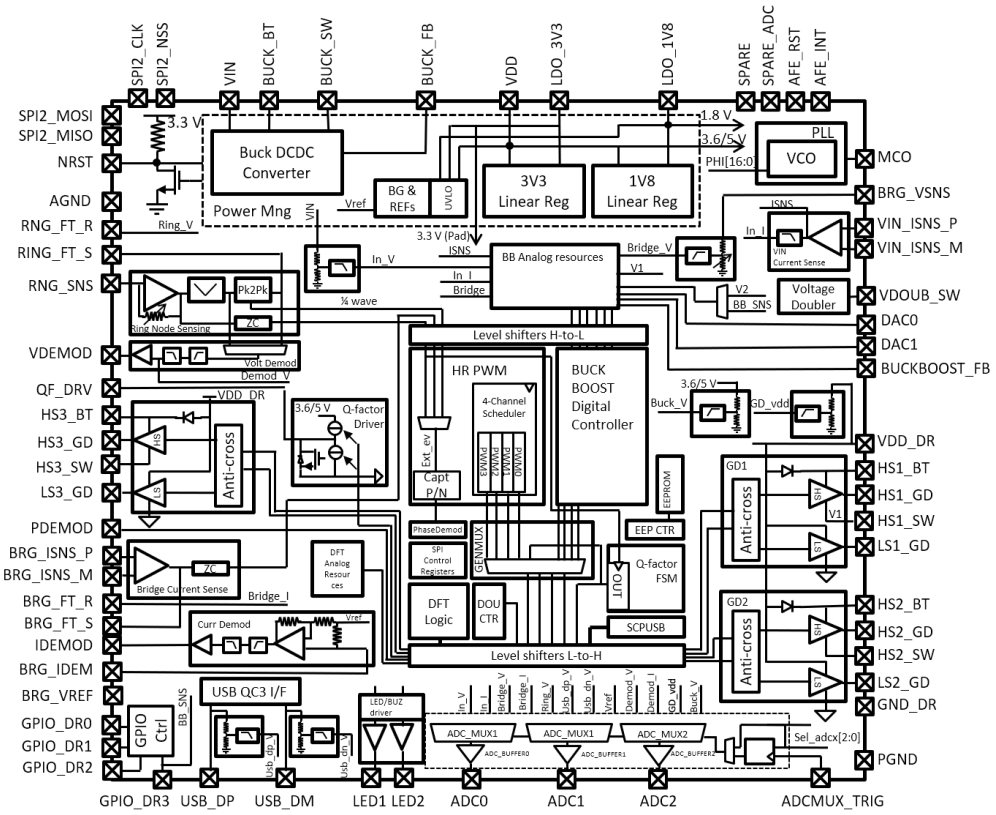
Figure 3. **Qi single coil, variable frequency full-bridge, USB powered**



2 Analog Front-End (AFE)

The following figure shows a simplified block diagram of STWBC2-HP AFE.

Figure 4. STWBC2-HP AFE block diagram



2.1 Analog AFE functions

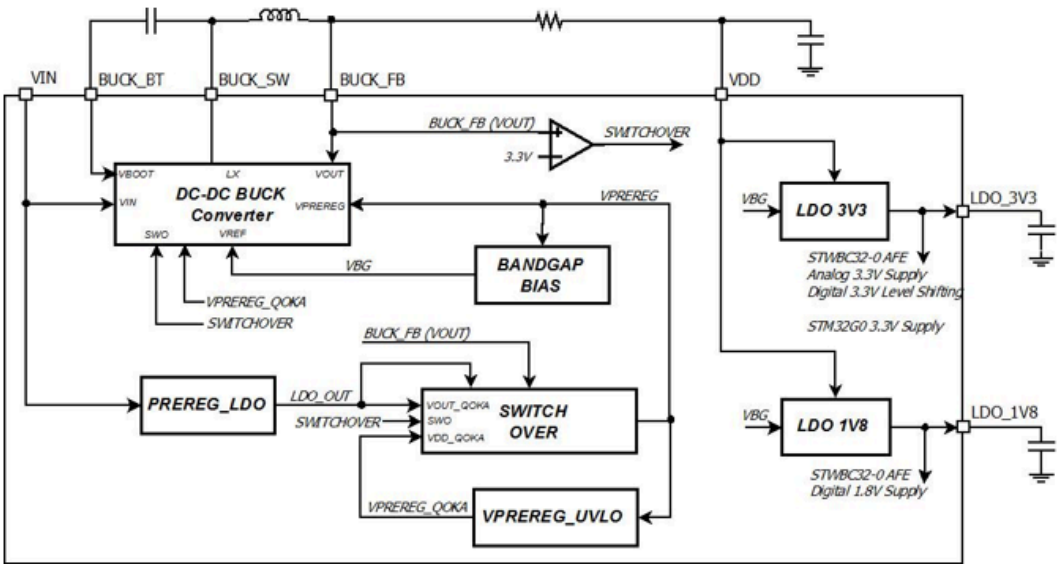
2.1.1 Power management

The figure below shows the block diagram of the STWBC2-HP power management. The power management of the whole system is fully integrated in STWBC2-HP AFE. Power supply for µC (STM32G071) is provided by LDO 3V3.

The power source of the STWBC2-HP application is a DC voltage. In industrial applications it is generally a simple DC source up to VIN=24 V. In consumer applications most power sources are interfaced using µUSB or USB-C connector, therefore the STWBC2-HP is designed to support current-limited supplies operating down to 4.1 V. The STWBC2-HP is able to negotiate power contract with USB-C signaling or using USB Power Delivery protocol.

STWBC2-HP generates its own power supplies from the power source directly and with good efficiency: to obtain this result a DC-DC BUCK converter is integrated into the STWBC2-HP.

Figure 5. **Power management**



The on-board DC-DC BUCK converter provides VDD supply voltage starting from external VIN DC source. VDD voltage is available on the external pin and supplies the internal voltage regulators LDO 3V3 and LDO 1V8. Both outputs of these regulators are available on the external pin as shown in Figure 5.

LDO 3V3 provides supply voltage for STM32G071 (internal bonding connection), for analog low-voltage circuits on STWBC2-HP AFE and finally for the level shift stages (from 1.8 V up to 3.3 V) used to interface digital signal between the STWBC2-HP (1.8 V digital domain) and the STM32G071 (3.3 V digital domain).

LDO 1V8 provides supply voltage for STWBC2-HP AFE digital core and for input stage of the level shifter blocks mentioned before.

2.1.2 Reset strategy

The STWBC2-HP includes a single NRST pin that is star-connected to both AFE and STM32G071; both STM32 and AFE use bi-directional pad for NRST. The following figure shows the simplified schematic of NRST function implementation.

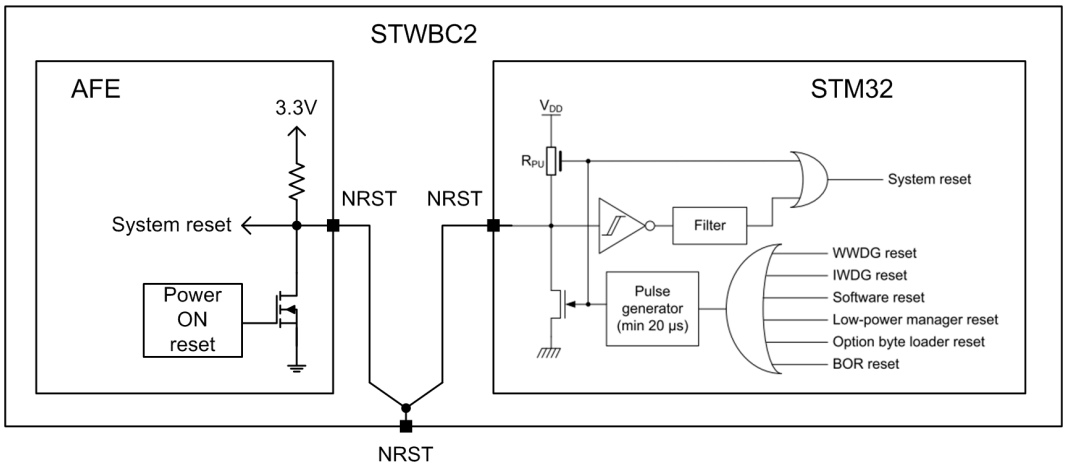
For STM32G071 the NRST pin function is:

- Reset input
- Warm reset output from WDG, SW reset, …

For AFE the NRST pin function is:

- Reset input
- Power ON Reset in link with BUCK DC-DC startup and LDOs startup.

Figure 6. **Reset strategy**



The NRST pin electrical specifications are described in Section 3.5 .
System reset filter is typically 440 ns.

2.1.3 Buck Boost DC-DC analog section

The DC-DC BUCK-BOOST converter provides power to the bridge which drives the wireless charging coil; this operation is done by using external power MOS driven by internal gate drivers. DC-DC BUCK-BOOST (hereinafter named BB-DCDC) is based on digital controller: the digital machine drives the gate drivers and uses low complexity analog circuits. Several different topologies can be implemented as shown by the following figures (LS=Low-Side driver, HS=High-Side driver):

Figure 7. **Boost topologies**

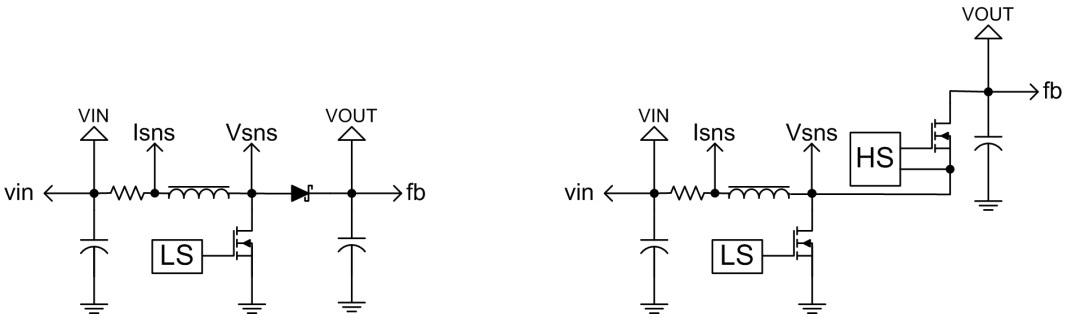


Figure 8. **Buck topologies**

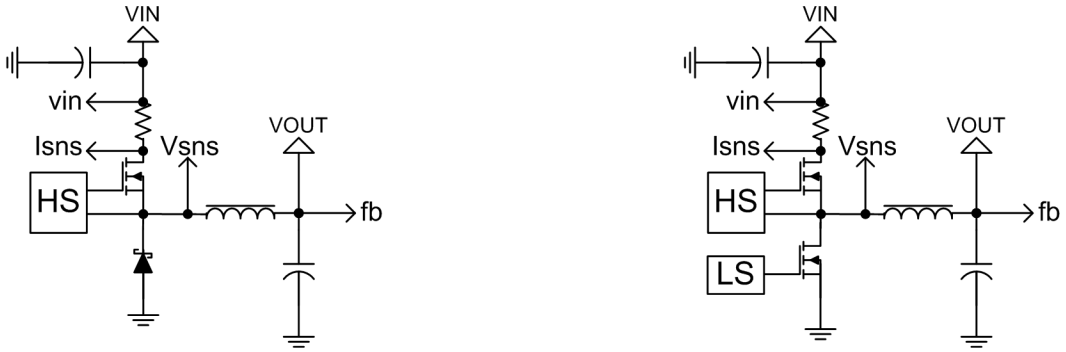
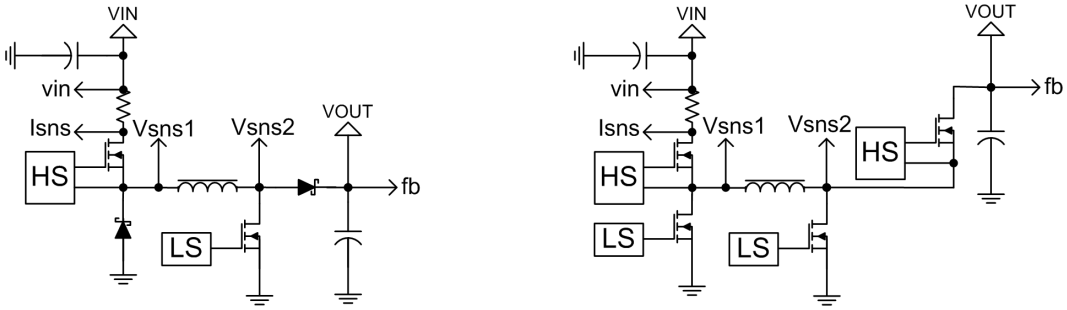
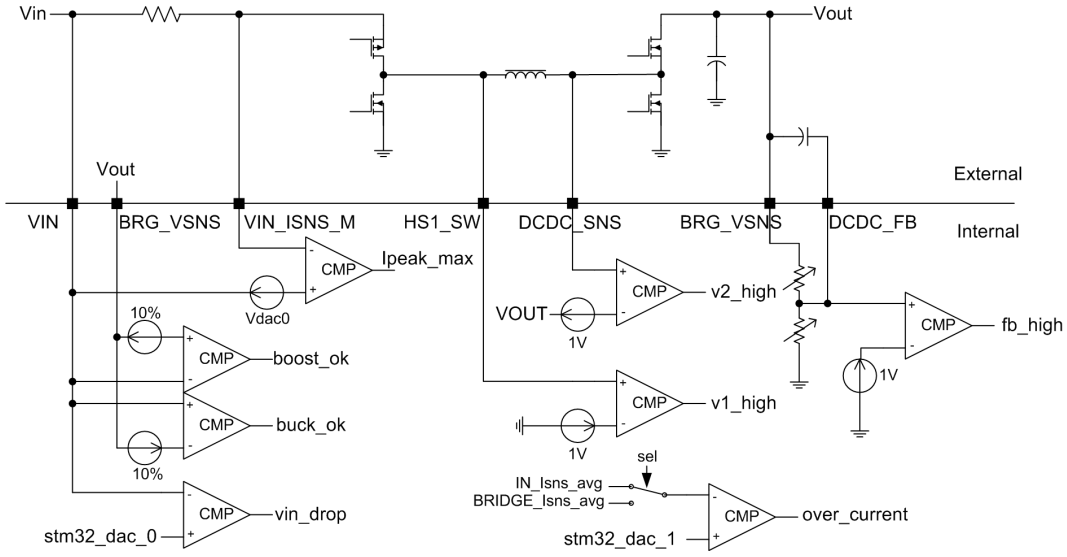


Figure 9. Buck/boost topologies



The BB-DCDC controller (see 2.1.3 Buck Boost DC-DC analog section) requires the following circuit from analog domain:

Figure 10. BB DC-DC analog section



2.1.4 Analog monitors

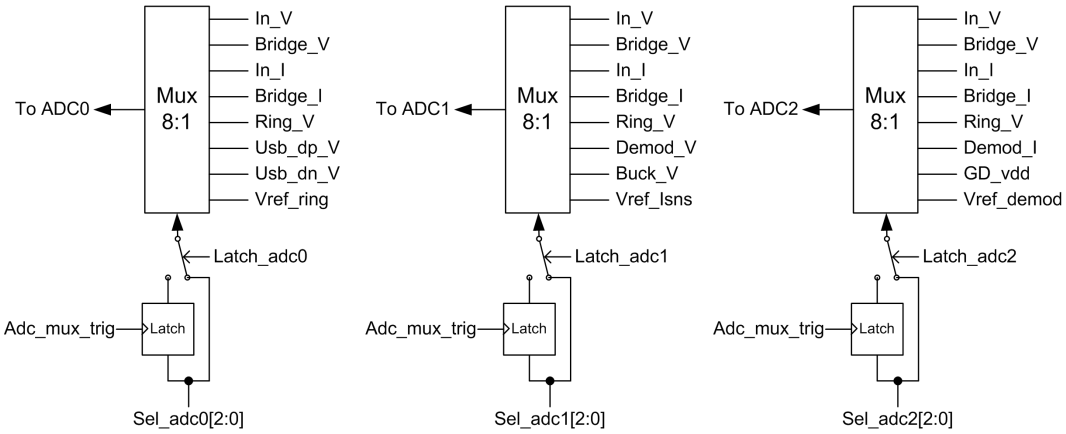
The STWBC2-HP AFE integrates 3 analog mux that are connected to STM32G071 die ADC inputs. Each mux selects a signal provided by an analog monitor cell. The mux switch time is fast enough allowing high speed conversion time from STM32 die.

3 multiplexers are in use:

- ADC0: this is the mux used for general purpose measurements
- ADC1: this mux is either used as general purpose or to provide the voltage demodulation analog form.
- ADC2: this mux is either used as general purpose or to provide the current demodulation analog form.

The channel selection (Sel_adcX[2:0]) is provided using SPI registers; selection is either directly applied to the mux or through a latch. This is configured using SPI register (Latch_adcX). When the channel selection is using latch mode, the Adc_mux_trig signal is used to latch the selection. The Adc_mux_trig signal comes from STM32 die GPIO. When using latch mode, the FW can prepare the next mux selection by accessing SPI registers while an ADC conversion is on-going. This selection is applied only after the Adc_mux_trig GPIO toggling. In this way, the SPI access time can be pipelined with the ADC conversion which improves the overall timing for multiple channel conversion. The internal signals are shipped to the STM32 ADC input with the necessary rescaling and gain to fit the ADC conversion range.

Figure 11. **Analog monitors**



The mux channel assignment is described below:

Table 2. **ADC MUX assignment**

Channel	ADC mux 0	ADC Mux 1	ADC mux 2
0	VIN voltage (In_V)		
1	Bridge voltage (Bridge_V)		
2	VIN current (In_I)		
3	Bridge current (Bridge_I)		
4	Ring node voltage (Ring_V)		
5	USB DP pin voltage (Usb_dp_V)	Voltage demodulator filter output (Demod_V)	Current demodulator filter output (Demod_I)
6	USB DM pin voltage (Usb_dm_V)	VDD voltage (Buck_V)	VDD of gate driver (GD_vdd)
7	Reference voltage of ring node circuit (Vref_ring)	Reference voltage of current sensor (Vref_Isns)	Reference voltage of demodulators (Vref_demod)

Each ADC mux output voltage ranges from 0 V to 2 V and channel selection time is 0.2 µs typ.

2.1.4.1 *VIN voltage monitor (In_V)*

The VIN voltage monitor provides a scaled down low pass filtered signal to the ADC mux.

Figure 12. **VIN voltage monitor**

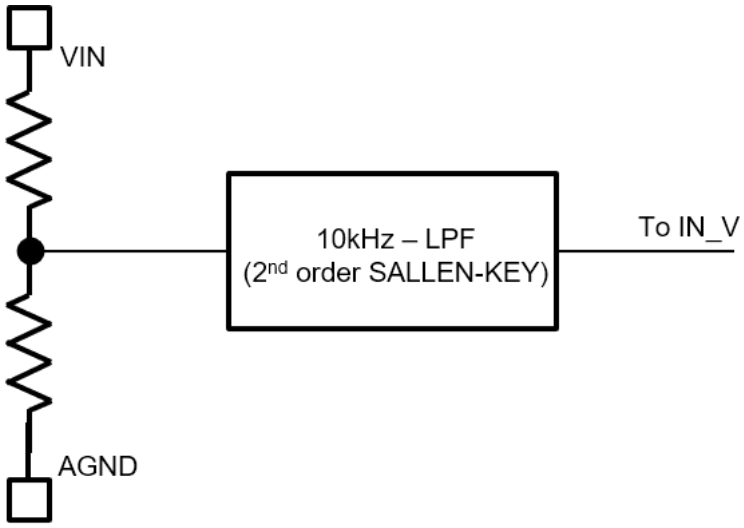


Table 3. **Vin voltage monitor operating parameters**

Parameter	Value
Division factor from VDD_DRIVER to ADC mux	/ 14.2
LPF	10kHz 2nd order

2.1.4.2 *Bridge voltage monitor (Bridge_V)*

The Bridge voltage monitor provides a scaled down low pass filtered signal to the ADC mux. The division factor is adjustable.

Figure 13. **Bridge voltage monitor**

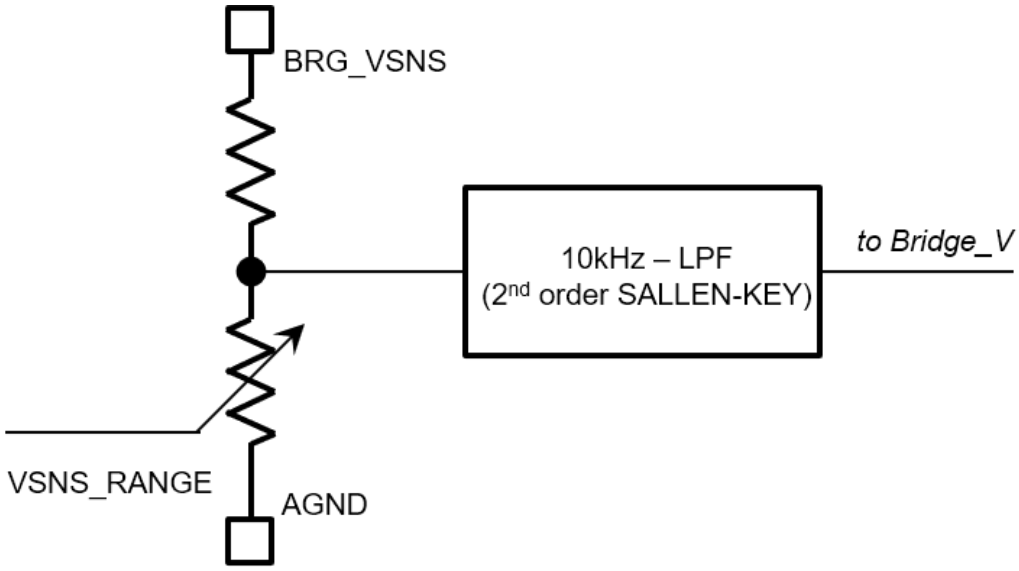


Table 4. **Bridge voltage monitor operating parameters**

Parameter	Condition	Value
Division factor from BRG_VSNS to ADC mux	Range = 0	/ 21.33

Parameter	Condition	Value
Division factor from BRG_VSNS to ADC mux	Range = 1	/ 3.33
LPF		8kHz 2nd order

2.1.4.3 *VDD_DRIVER voltage monitor (GD_vdd)*

The VDD_DRIVER voltage monitor provides a scaled down low pass filtered signal to the ADC mux.

Figure 14. VDD_DRIVER voltage monitor

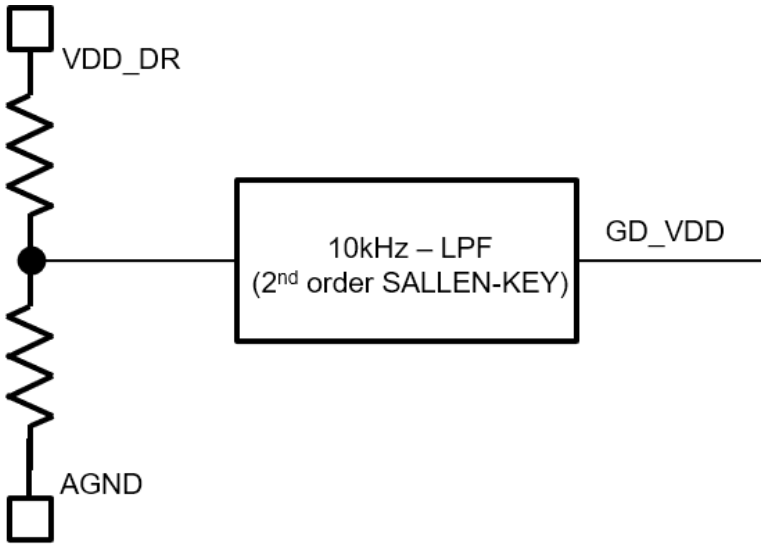


Table 5. VDD_DRIVER voltage monitor operating parameters

Parameter	Value
Division factor from VDD_DRIVER to ADC mux	/ 5.26
LPF	10kHz 2nd order

2.1.4.4 *VIN current monitor (In_I)*

The input current sense provides a filtered value to the ADC mux and BUCK-BOOST DC-DC controller (current limitation). The circuit provides also a direct signal to the BB-DCDC controller in order to monitor the peak current in the BB-DCDC inductor.

Figure 15. VIN current monitor

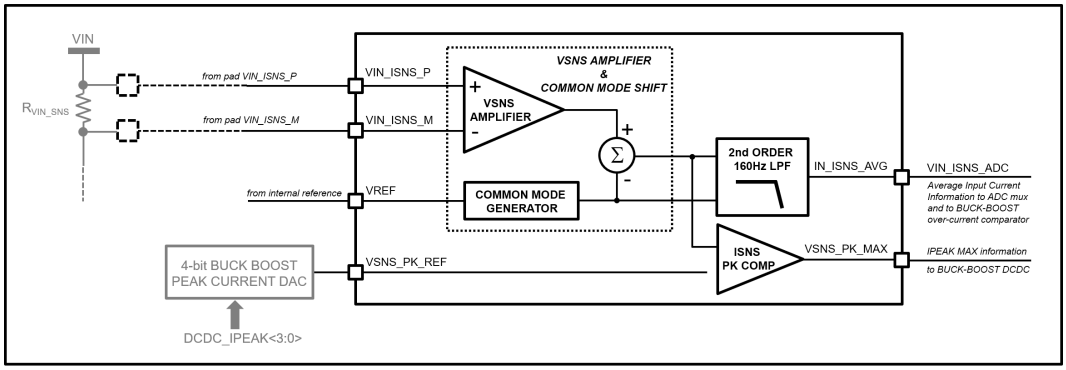


Table 6. VIN current monitor operating parameters

Parameter	Value
Multiplication factor from V(VIN_SNS_P, VIN_SNS_N) to ADC mux	x 4.1
LPF	160Hz 2nd order

2.1.4.5 *Bridge current monitor (Bridge_I)*

The Bridge Current Sense stage provides a voltage proportional to the filtered value of current flowing in RBRG_SNS to the ADC MUX. Since the sense resistor RBRG_SNS is placed at the bridge GND node, a synchronous rectification is indirectly applied on the sensed voltage.

The current flowing in the sense resistor can be positive or negative; the zero cross detector is a simple comparator (ZCD COMP) providing the polarity of the current flowing in RBRG_SNS.

The required 2nd order low pass filter (1 ms LPF) is implemented by means of external passive components as shown in the figure below (requirement is T=1ms ±10%).

Figure 16. Bridge current monitor

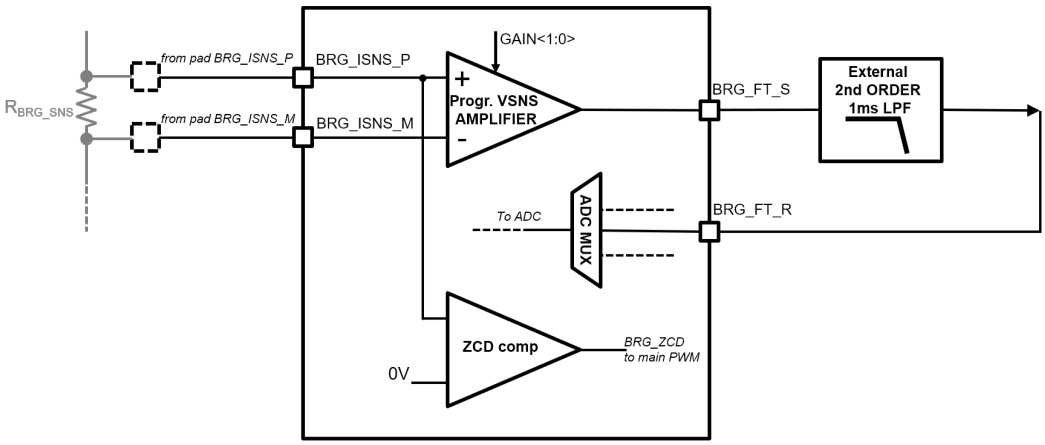


Table 7. Bridge current monitor operating parameters

Parameter	Condition	Value
Multiplication factor from V(BRG_ISNS_P, BRG_ISNS_N) to ADC mux	Gain = 0	x 4
	Gain = 1	x 8
	Gain = 2	x 16
	Gain = 3	x 32

2.1.4.6 *Ring voltage monitor (Ring_V)*

The tank ring voltage is AC coupled with the STWBC2-HP using an RC circuit. As RNG_SNS pin is maintained at a fixed voltage, the ring voltage is converted in current flowing in the RNG_SNS pin thanks to the resistor:

$$I_{RNG\_SNS} = \frac{V_{RING}}{R}$$

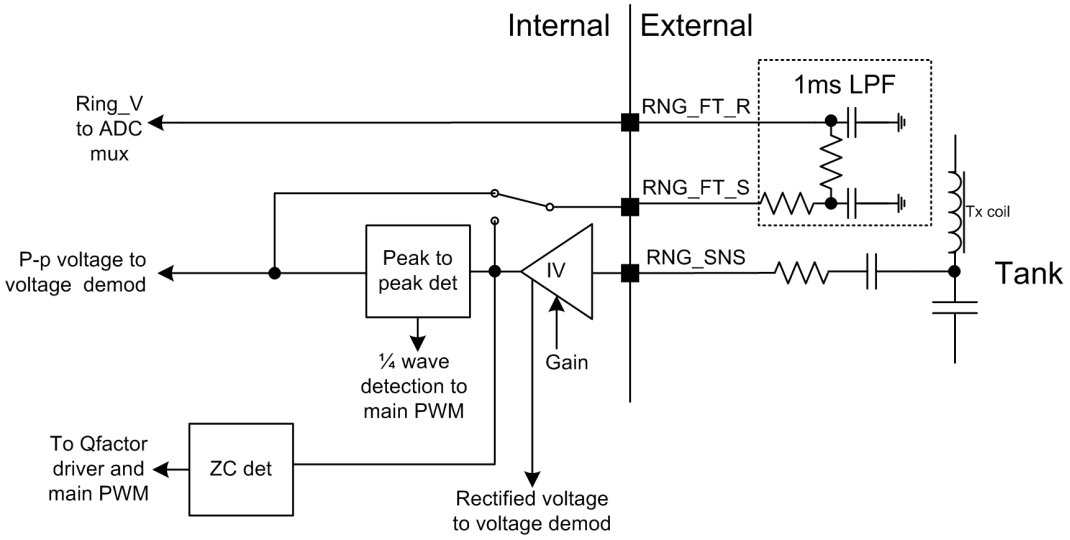
The current to voltage amplifier (IV) converts the input current in voltage. It rectifies also the signal.

The peak-to-peak detector tracks the positive and negative peaks of the ring signal.

The RNG_FT_S pin is driven by the rectified signal from IV converter or from the peak-to-peak detector. An external 1 ms low pass filter is connected to the RNG_FT_S pin.

The output of the external low pass filter is connected to the RNG_FT_R pin. This signal is internally routed to the ADC mux Ring_V input.

Figure 17. **Ring voltage interface**



2.1.4.7 *USB DP and DM monitors (Usb_dx_V)*

The USB DP and USB DM voltage monitors provide scaled down low pass filtered signals to the ADC mux.

Figure 18. **USB DP/DM monitor**

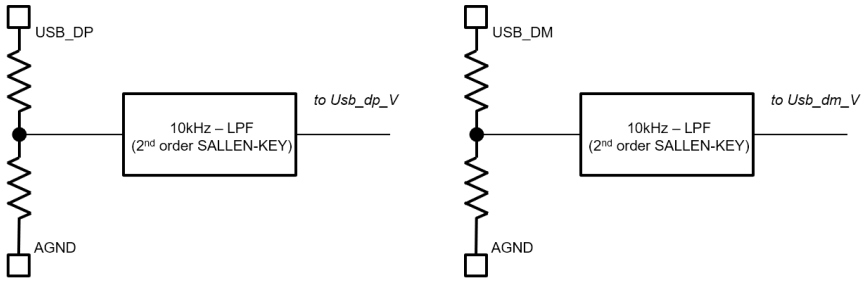


Table 8. **USB DP/DM monitor operating parameters**

Parameter	Value
Division factor from USB_DP or DM to ADC mux	/ 2.5
LPF	10 kHz 2nd order

2.1.4.8 *VDD voltage monitor (Buck_V)*

The DC-DC buck voltage monitor provides a scaled down low pass filtered signal to the ADC mux.

Figure 19. **DC-DC buck monitor**

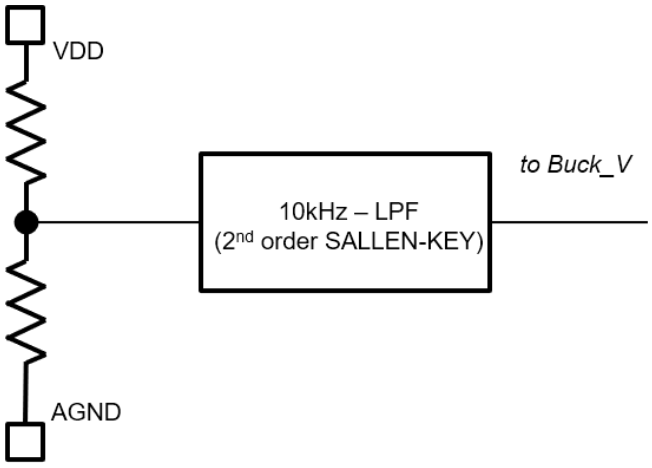


Table 9. **DC-DC buck monitor operating parameters**

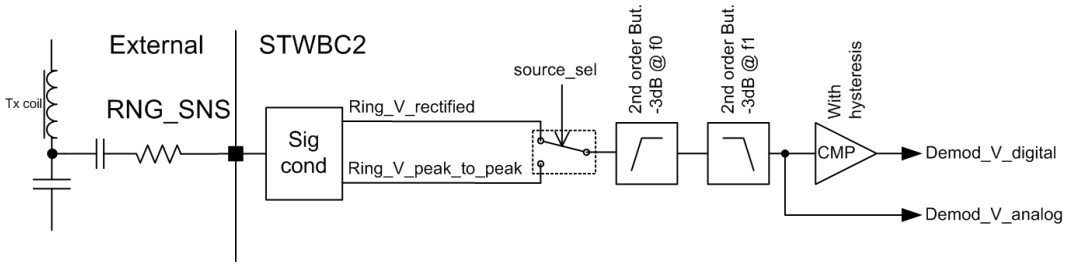
Parameter	Value
Division factor from VDD to ADC mux	2.86
LPF	10k Hz 2nd order

2.1.5 Analog demodulators

2.1.5.1 Voltage demodulator

The voltage demodulator detects Rx modulation from the variation of the ringing node voltage.

Figure 20. **Voltage demodulator**



The ring voltage monitor cell provides the peak-to-peak of the probed voltage (see Ring voltage monitor (Ring_V)). It provides also a rectification of the ringing voltage. From SPI register, the FW can select on-the-fly the peak-to-peak or the rectified signal to be demodulated. The selected signal is directly filtered using a HPF and an LPF to select the band of the useful modulation. This filtered signal is shipped to ADC mux for STM32 acquisition through ADC. The signal is also sent to a comparator in order to provide a direct digital 1bit form of the modulation to an STM32 GPIO.

Table 10. **Voltage demodulator operating parameters**

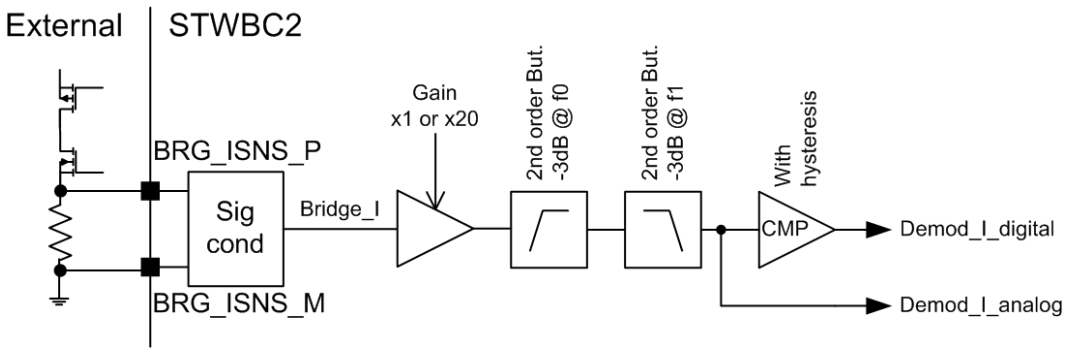
Parameter	Value
Demodulation HPF	2nd order F0 adjustable by SPI registers to 300 Hz, 450 Hz, 600 Hz, 750 Hz Frequency tolerance +/-10%

Parameter	Value
Demodulation LPF	3rd order F1 adjustable by SPI registers to 4k-Hz, 8k-Hz, 12k-Hz, 16k-Hz Frequency tolerance +/-10%
Comparator hysteresis	Adjustable by SPI registers 0, 0.6 mV, 1.2 mV, 2.4 mV

2.1.5.2 *Current demodulator*

The current demodulator detects Rx modulation from the variation of the bridge current.

Figure 21. Current demodulator



The bridge current monitor provides a voltage image Bridge_I. As the current is sensed on the bridge supply, it is naturally rectified by the toggling of the driver. Hence, the image of the current can be directly filtered to retrieve the modulation. This signal is filtered using a HPF and an LPF to select the band of the useful modulation. This filtered signal is shipped to ADC mux for STM32 acquisition through ADC. The signal is also pushed to a comparator in order to ship a direct digital 1bit form of the modulation to an STM32 GPIO. A configurable gain allows adapting the signal level to the filtering chain.

Table 11. Current demodulator operating parameters

Topic	Value
Demodulation HPF	2nd order F0 adjustable by SPI registers to 300 Hz, 450 Hz, 600 Hz, 750 Hz Frequency tolerance +/-10%
Demodulation LPF	3rd order F1 adjustable by SPI registers to 4k Hz, 8k Hz, 12k Hz, 16k Hz Frequency tolerance +/-10%
Comparator hysteresis	Adjustable by SPI registers 0, 0.6 mV, 1.2 mV, 2.4 mV

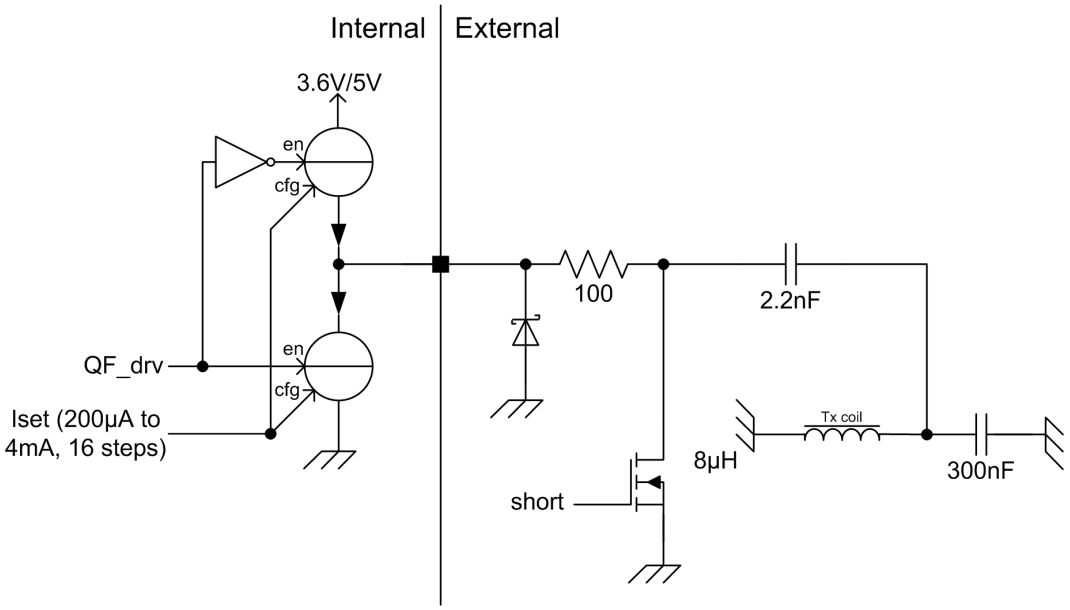
2.1.6 Quality factor driver

The purpose of the circuit is to drive the resonant LC network used in the wireless power transfer in order to transmit a steady oscillation by which to measure its quality factor (Q factor).

When a Qi receiver is coupled, the Q factor of the network would decrease. Moreover, a foreign object inadvertently and dangerously placed in the coupling region would absorb power from the transmitter causing the Q factor to fall by even orders of magnitude.

The measurement is done by evaluating the amplitude and the period of the waveform so obtained on the ringing node. The Q factor is first evaluated on the transmitter in absence of receiver during the final assembly test procedure. The measurement needs not to be absolute but it is recorded in the device NVM as a calibration point. The same measurement is repeated during normal functioning before starting the power transfer process. By comparison with the stored value, the application can reveal the presence of a receiver or detect an undesired foreign object.

Figure 22. QF driver



The analog driver is a square current generator with symmetrical source/sink programmable current strength. During power transfer, the QF_DRV pin must be shorted to GND to avoid damage from the high voltage ringing node. In this case the total internal/external QF_DRV pin resistance to GND has to be as low as to guarantee a maximum voltage on the QF_DRV pin of ±300 mV.

The QF_drv signal is driven by the QF controller digital machine (see Section 2.2.3).

Table 12. QF driver operating parameters

Topic	Value
Current generator	250 µA to 4 mA, 16 steps
QF_DRV swing	20 mVpp min. (low Q) 1Vpp typ. 2Vpp max.
Ringing node voltage	20 mVpp min. (low Q) 1Vpp typ. (high Q)

2.1.7 LED/buzzer driver

LED/buzzer driver is a VDD (3.6 V/5 V) digital IO or current source that can be driven by a pattern generator from the digital section (see Section 2.2.6). The frequency of pattern generator output is in the range of Hz for LED and kHz for buzzer. The pattern generator can be bypassed for a direct driving of the pin (GPIO mode).

The digital driver can be configured in open drain modes. The current source current value can be configured.

Figure 23. LED/buzzer driver

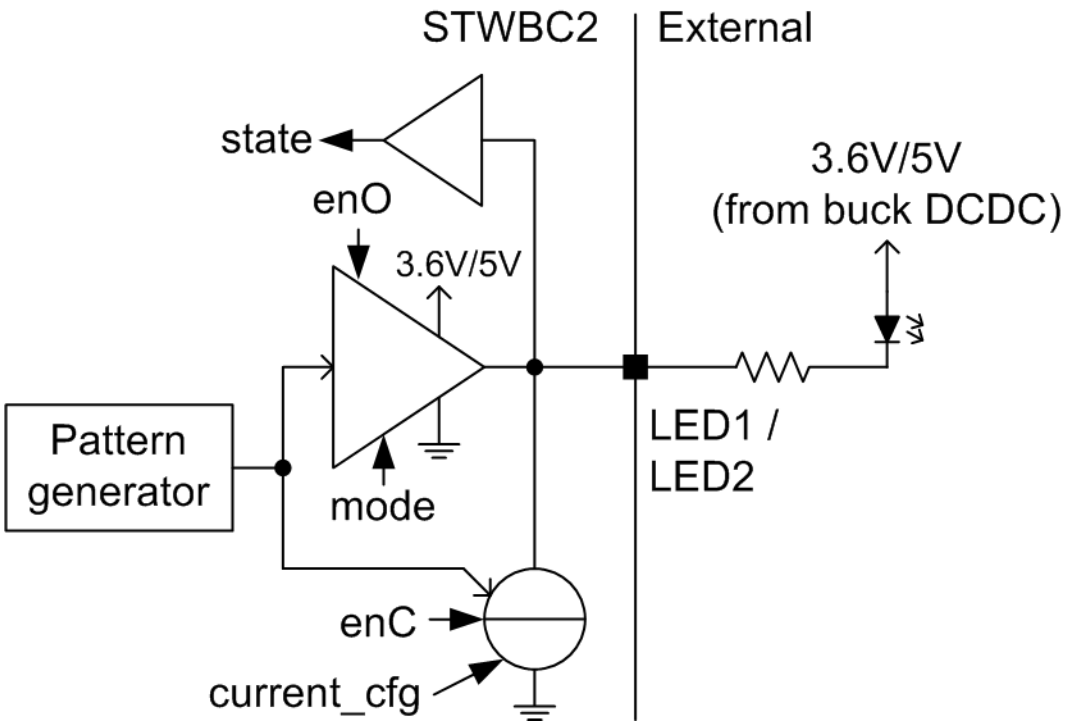


Table 13. LED/buzzer driver operating parameters

Parameter	Value	Comment
Driver mode	HiZ, push-pull, open drain high-side, open drain low-side, current source	
Input mode	Yes, pad state readable in register	
Operating voltage	3.6 V/5 V	
Digital IO drive current	20 mA for LED with 0.5 V driver drop Optionally, 50 mA for buzzer with 0.5 V driver drop	VCC=5 V
Constant current sink	Configurable at 2.5 mA, 5 mA, 7.5 mA, 10 mA	

2.1.8 USB detection

The USB detection circuit allows the biasing and signaling of DP and DM USB pins. Using this circuit in combination with an adequate FW that uses ADC measurement on DP and DM, it is possible to detect the connection of:

- Basic downstream ports (USB simple host)
- Downstream ports with charging capabilities as per USB BC 1.2 specification
- Dedicated chargers as per USB BC 1.2 specification
- Other manufacturer specific chargers

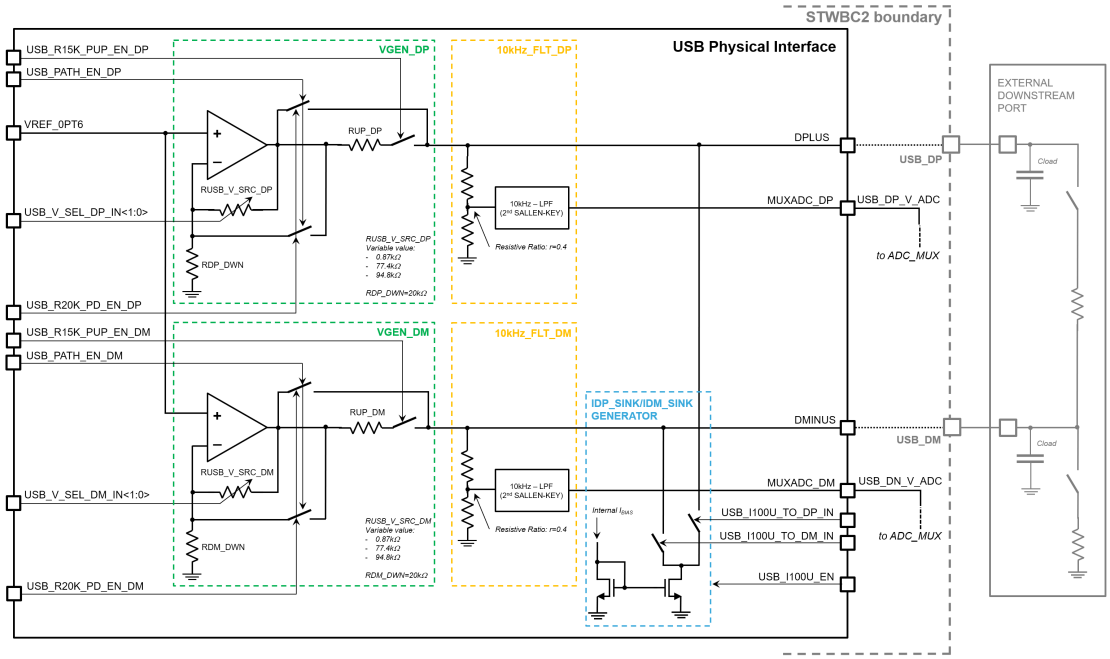
The detection circuit integrates two independent voltage sources driving DP and DM pins with up to 1mA current capability. The voltage can be either 0.6 V, 2.8 V or 3.3 V. Optionally, a 15 kohms resistor can be enabled in series with the voltage source.

The detection circuit integrates two independent 20k pull-down resistor that can be enabled on DP and DM.

The detection circuit integrates a 100 µA current sink that can be applied either on DP or on DM.

Finally, the voltage on DP and DM are scaled down and filtered and then sent to the ADC mux (see Section 2.1.4.7).

Figure 24. USB detection



The detection of USB source type is done by FW. It configures the USB detection circuit and uses ADC measurements to check the response of source.

The first step of detection consists in checking the battery charging capability as per the USB BC 1.2 specification:

1. Primary detection: DP is driven at 0.6 V and DM is biased with 100 µA current sink. DM voltage is measured.
2. Secondary detection: DM is driven at 0.6 V and DP is biased with 100 µA current sink. DP voltage is measured.
3. Based on table below and DM/DP voltage measures, the FW can deduce the BC1.2 compatible type of source.

Table 14. USB source type detection

Mode	USB source type	DP BIAS	DM BIAS	Check
Primary Detection	Dedicated charger	0.6 V	100 µA sink	DM > 0.4 V
	USB host with charging cap	0.6 V	100 µA sink	DM > 0.4 V
	Simple USB host	0.6 V	100 µA sink	DM ~ 0.0 V
Secondary Detection	Dedicated charger	100 µA sink	0.6 V	DP > 0.4 V
	USB host with charging cap	100 µA sink	0.6 V	DP ~ 0.0 V

In case of dedicated charger detected, the FW proceeds in detection of high voltage charger. To do so:

1. Drive DP to 0.6 V and enable 100 µA sink on DM.

2. Wait 1.5 s
3. Check DM voltage.
   a. If DM voltage has felt to 0V, the dedicated charger is a high voltage one.
   b. If DM stays > 0.4 V, the source is a simple dedicated charger.

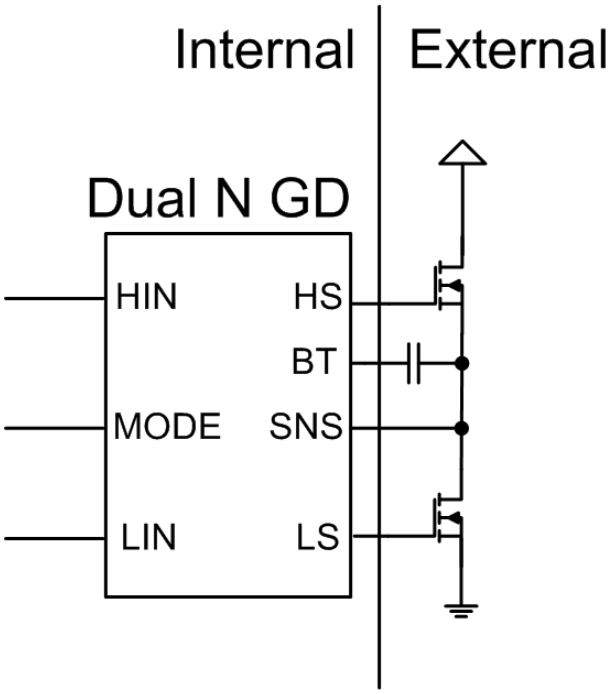
If the source is a USB host or a simple dedicated charger, the DP and DM pins can be turned to a 20k pull-down resistor.

If a high voltage dedicated charger is detected, the DP voltage should be maintained equal or above 0.6 V. The USB detection circuit can be used to configure the charger to produce a desired VBUS voltage.

2.1.9 Gate drivers

The STWBC2-HP embeds 3 gate drivers designed to drive NMOS for low- and high-side using a bootstrap technique.

Figure 25. Gate driver



All gate drivers are powered from VDD_DR and GND_DR pins. VDD_DR supports up to 10 V.

The sense and bootstrap pins are designed to support high voltage switching up to 40 V.

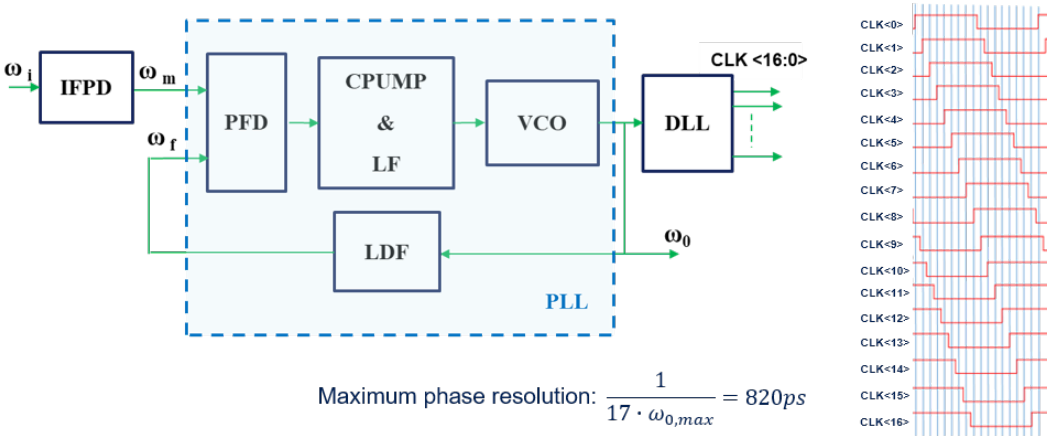
The gate drivers integrate their own anti-cross conduction circuit based on gate current analysis. The anti-cross conduction circuit can be individually disabled on each gate driver. This mode allows using fixed deadtime from PWM source directly. This mode is recommended if the gate driver is fed by the DC-DC digital controller. It can also be needed if NMOS gate resistor is too large. In this case, the embedded anti-cross conduction circuit can be inefficient and explicit fixed deadtime should be programmed in PWM generator.

2.1.10 PLL

The STWBC2-HP AFE embeds Phase Locked Loop IP which main purpose is to generate the accurate 17 phases required for High Resolution PWM implementation (see Section 2.2.1 for details about HR PWM).

The PLL typical operating frequency is 40 MHz.

Figure 26. **PLL**



Maximum phase resolution: $\frac{1}{17 \cdot \omega_{0,max}} = 820ps$

The PLL IP is comprised by the following main blocks:

- Input Frequency Pre-Divider IFPD
- Phase/Frequency Detector PFD
- Charge Pump CPUMP
- Loop Filter LF
- Voltage Controlled Oscillator VCO
- Loop Frequency Divider (in which LDF is the Loop Division Factor)
- Delay Locked Loop DLL

Table 15. **PLL operating parameters**

PARAMETER/FEATURES/TOPIC	VALUE	COMMENT
Input Reference Clock Frequency (ωi)	8MHz to 56MHz	Typical 16MHz from MCU (MCO/XTAL)
Input Frequency Pre-Divider Factor – IFPD (M)	1 to 32	Adjustable
Pre-Divider Output Frequency (ωm)	2.375MHz to 20MHz	
Loop Frequency Divider Factor – LDF (N)	2 to 16	Adjustable
VCO Frequency (ω0)	38MHz to 70MHz	ωo =(M/N)·ωi
Delay Locked Loop Output Frequency	38MHz to 70MHz	Frequency value of the 17 phases (CLK<16:0>)
Maximum Lock Time	30ms + 256 cycles of PFD input frequency	PFD frequency of 8 MHz÷20 MHz
	60ms + 256 cycles of PFD input frequency	PFD frequency of 4 MHz÷8 MHz
	120ms + 256 cycles of PFD input frequency	PFD frequency of 2.375 MHz÷4 MHz

2.1.11 GPIO

The STWBC2-HP AFE includes 4 GPIO (GPIO_DRx) that can be driven from internal PWM sources (see Section 2.2.4). Using the dedicated SPI registers, these GPIO can also be used as general purpose digital input or output.

Table 16. **GPIO operating parameters**

PARAMETER/FEATURES/TOPIC	VALUE	COMMENT
GPIO VDD	3.3 V	**LDO3V3**

PARAMETER/FEATURES/TOPIC	VALUE	COMMENT
Sink and source current	>6 mA	
Output voltage low	0.4 V max. at 6 mA load	
Output voltage high	LDO3V3-0.4 V min. at 6 mA load	
Fall time, Rise time	11 ns max. with 50 pF	10%-90% threshold

2.1.12 Control of analog functions

The analog functions can be independently enabled in order to optimize the power consumption of the STWBC2-HP. This can be done through the SPI registers ANA_PWR_CTRL_x of page 0.

Table 17. ANA_PWR_CTRL_0 mapping

Bit position	Controled cell	Comment
7	Current demodulator LPF	
6	Current demodulator HPF	
5	Current demodulator comparator	
4	Bridge zero cross detector	
3	Bridge voltage monitor	
2	ADC mux 2	
1	ADC mux 1	
0	ADC mux 0	

Table 18. ANA_PWR_CTRL_1 mapping

Bit position	Controled cell	Comment
7	Gate driver 1 low-side	Force input of gate driver to 0 before enabling
6	Gate driver 1 high-side	Force input of gate driver to 0 before enabling
5	Voltage demodulator LPF	
4	Voltage demodulator HPF	
3	Voltage demodulator comparator	
2	VCCD monitor	
1	Ring node peak-to-peak detector	
0	Ring node I to V amplifier	

Table 19. ANA_PWR_CTRL_2 mapping

Bit position	Controled cell	Comment
7	VIN voltage sense monitor	
6	VIN current sense	Used for BB-DCDC
5	VIN peak current comparator	Used for BB-DCDC
4	Bridge current amplifier	
3	Gate driver 3 low-side	Force input of gate driver to 0 before enabling
2	Gate driver 3 high-side	Force input of gate driver to 0 before enabling
1	Gate driver 2 low-side	Force input of gate driver to 0 before enabling

Bit position	Controled cell	Comment
0	Gate driver 2 high-side	Force input of gate driver to 0 before enabling

Table 20. ANA_PWR_CTRL_3 mapping

Bit position	Controled cell	Comment
7	BB-DCDC general enabling	Used for BB-DCDC
6	Reserved, keep at 0	
5	Reserved, keep at 0	
4	Reserved, keep at 0	
3	Voltage demodulator VR	
2	Voltage demodulator comp Int	
1	Current demodulator INA	
0	Current demodulator comp Int	

Table 21. ANA_PWR_CTRL_4 mapping

Bit position	Controled cell	Comment
7	Reserved, keep at 0	
6	BB-DCDC V2 high comparator	Used for BB-DCDC
5	BB-DCDC V1 high comparator	Used for BB-DCDC
4	BB-DCDC Ipeak max. DAC	Used for BB-DCDC
3	BB-DCDC feedback comparator	Used for BB-DCDC
2	BB-DCDC VIN drop detection	Used for BB-DCDC
1	BB-DCDC overcurrent detection	Used for BB-DCDC
0	BB-DCDC buck boost OK comparator	Used for BB-DCDC

Table 22. ANA_PWR_CTRL_5 mapping

Bit position	Controled cell	Comment
7	PLL supply	
6	Reserved, keep at 0	
5	Reserved, keep at 0	
4	Reserved, keep at 0	
3	Reserved, keep at 0	
2	Reserved, keep at 0	
1	Enable digital pads that are in parallel to analog functions	
0	Demodulators pre regulator	

2.2 Digital AFE functions

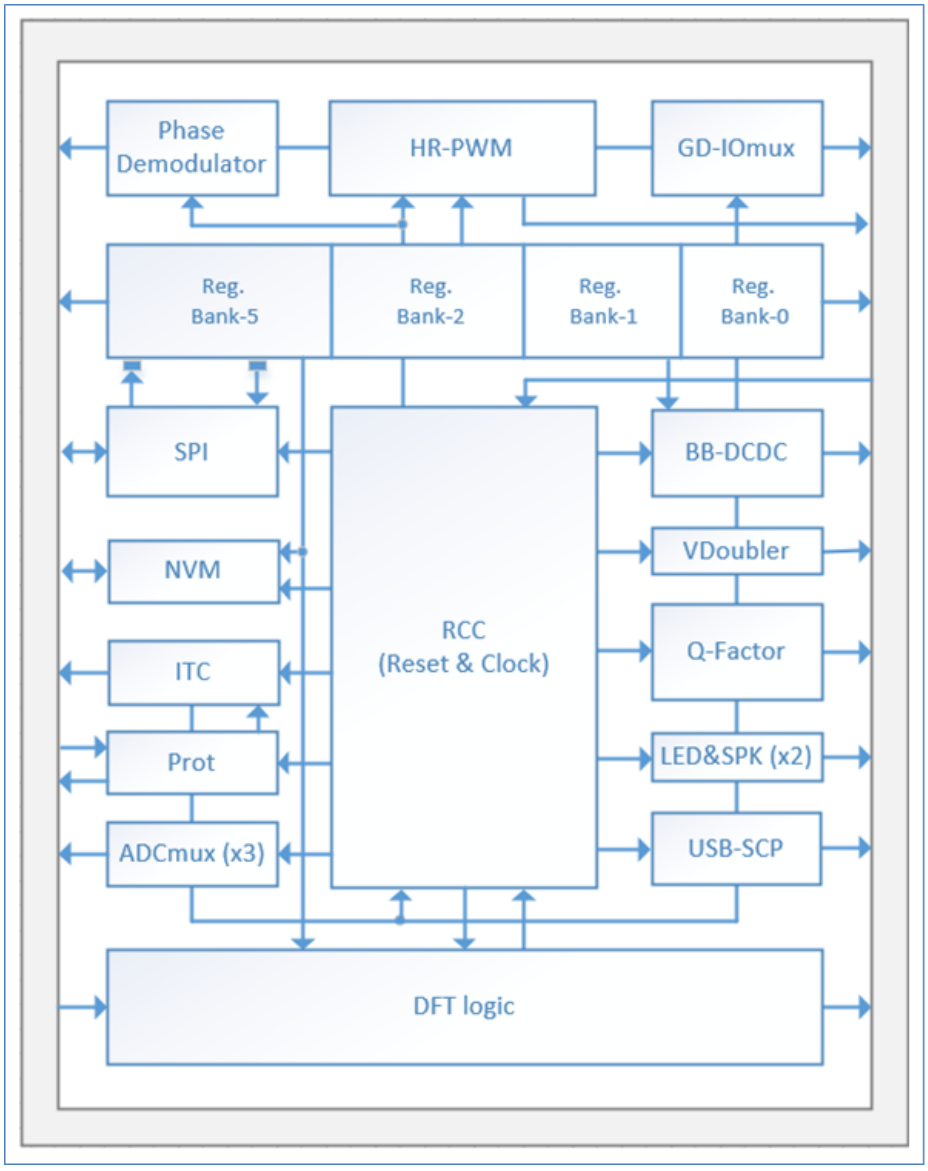
This section describes the digital core logic implemented in the Analog Front-End (AFE).

The digital block integrates an SPI slave interface providing access to registers through a parallel peripheral bus. Registers are arranged by pages. The registers allow controlling both digital IPs and analog macros.

The digital block embedded the following functional IPs:

HR-PWM: High resolution PWM
Phase demodulator: Digital phase demodulator
QF controller: Quality factor measurement digital part
GD modes: Management of integrated gate driver modes
GD mux: Driving signal mux for gate drivers
ADC mux latch controller: Set of mux with latch capabilities
LED pattern generator: Pattern generation for LED blinking or buzzer sound generation
ITC: Interrupt controller
Protection: Fault detection logic
NVM controller: NVM control logic interface
RCC controller: Reset and Clock Controller logic
SPI: SPI interface
BB-DCDC: Back Boost DCDC voltage regulator
VDoubler: Voltage doubler control logic
Reg-banks: Internal bank of register

Figure 27. Digital top level block diagram overview

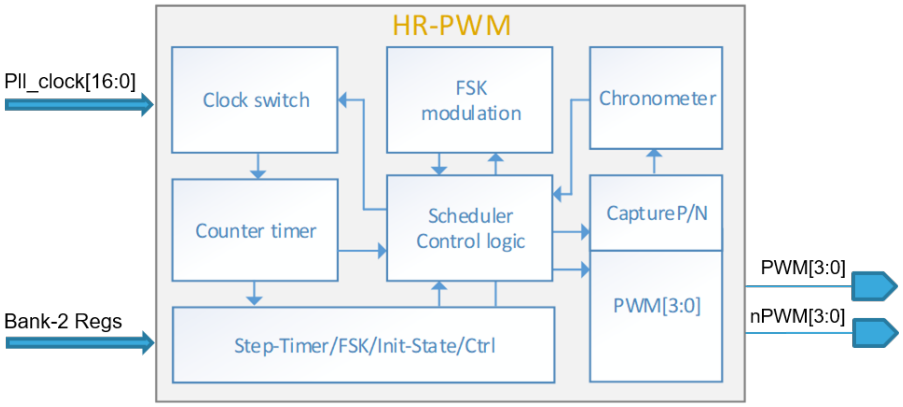


2.2.1 Main PWM

2.2.1.1 *Overview*

This block implements the high resolution PWM (HR-PWM) configurable to synthesize very accurate frequencies and duty cycle to fulfil the wireless charger standard requirements.

Figure 28. HR-PWM block diagram overview



2.2.1.2 *Feature list*

- Timing with high resolution using 17-step DLL
- PWM generation based on a sequence of steps
- Integrated FSK generation using data FIFO
- External event reactivity
- High resolution chronometer
- Chronometer output port for phase demodulation
- Graceful start-up and stop-down
- Synchronous update of sequence
- Pipeline configuration through shadow registers

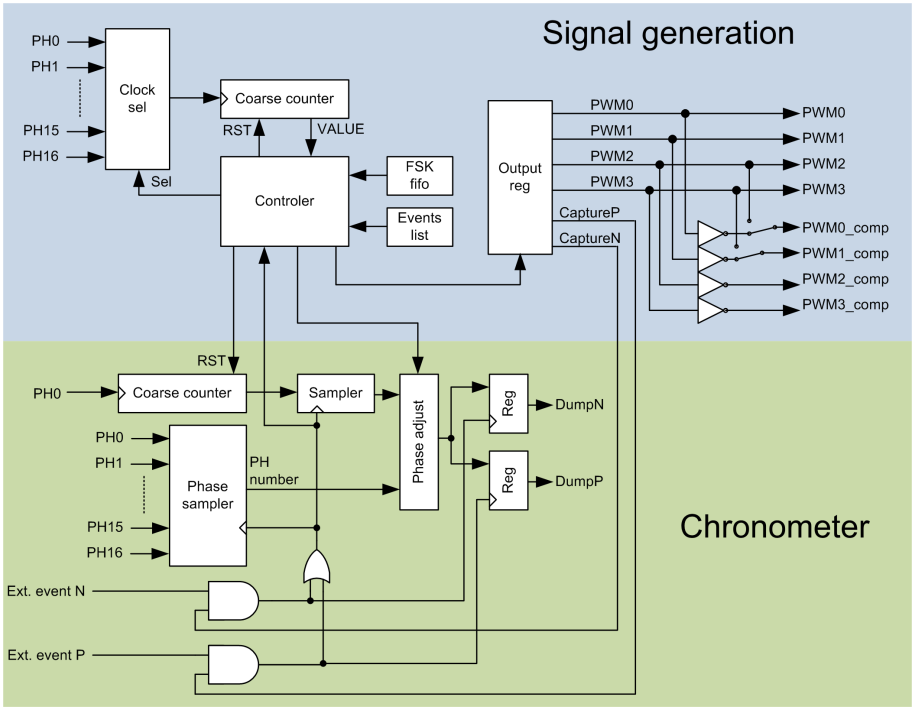
2.2.1.3 *Functional description*

The Main PWM is composed of two essential blocks:

- The signal generation block produces PWM outputs based on a programmable sequence
- The chronometer block captures the timing of external events

Here below is shown the internal block diagram.

Figure 29. **HR-PWM subsystem overview**



The Main PWM runs with high resolution timings. To do so, the clocking of the cell is made through 17 phases of a 40 MHz typical clock. This provides a resolution equivalent to a clock at 680 MHz.

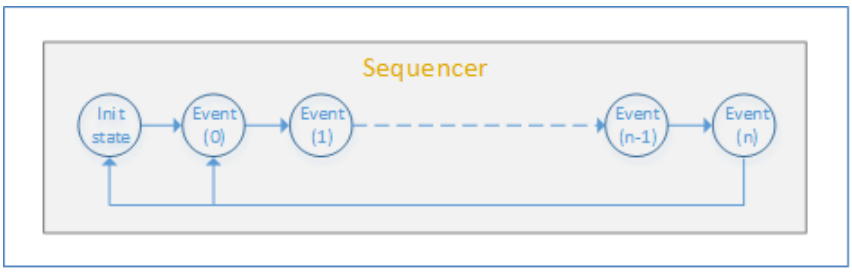
2.2.1.3.1 Sequencer PWM signal generation

The signal generation block is designed around a controller that processes the event list. Each event in the list is informing about a delay before applying a specific toggling of the PWM outputs. The delay is made of a coarse value corresponding to a count of clock cycles and a fine value corresponding to a count of clock phases.

When the delay expires, the controller applies the required toggling on PWM outputs and selects the clock phase in order to achieve the next fine delay. The controller treats then the next active event in the list. When the controller reaches the end of the event list, it restarts processing the list from the beginning.

Figure 30 shows the sequences controlled by the scheduler logic.

Figure 30. **Sequencer circuit overview**

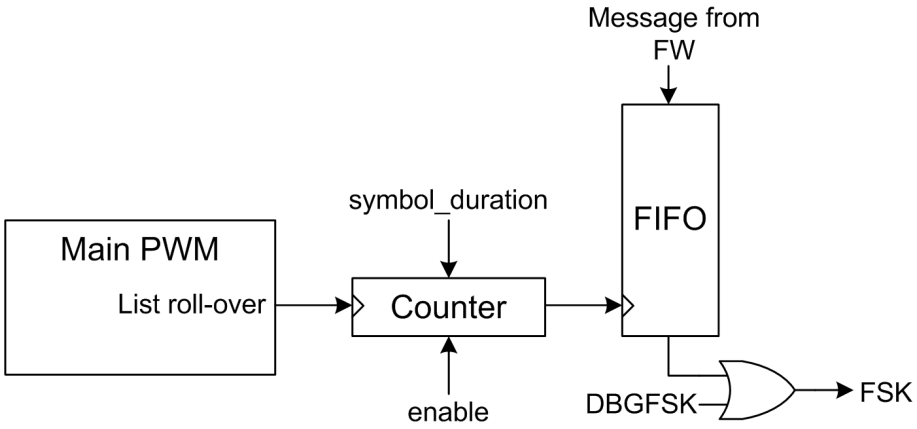


2.2.1.3.2 FSK modulation

Each event in the list can be tagged to be active only with a specific state of FSK bit, either 0 or 1. With such tagging, some events in the list become conditionals so the PWM generation depends on the state of the FSK bit. This feature allows programming the signal generation block to generate frequency shift keying messaging.

The FSK bit generation is using a FIFO:

Figure 31. **FSK modulation circuit overview**



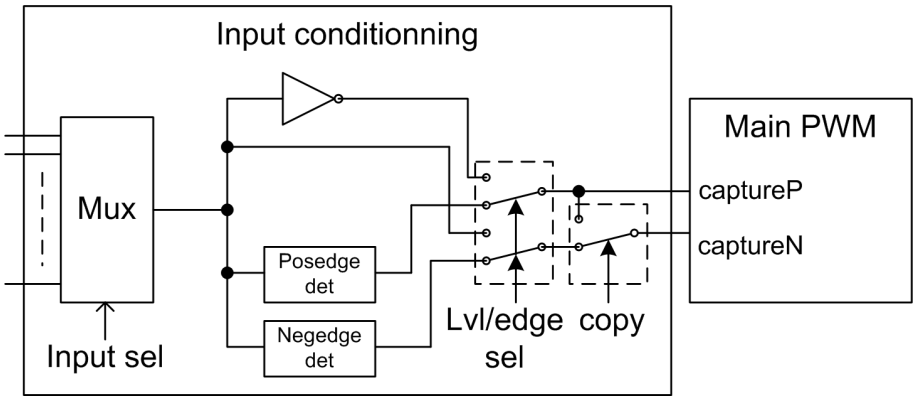
A bit from the FIFO is drained after a programmable count of the Main PWM cycle. A cycle is completed each time the list of events rolls over.

Note: FW has to guarantee the non-overflow of the FSK FIFO data.

2.2.1.3.3 Chronometer

The chronometer block measures the time for which external events occurs. In a full PWM cycle, it is considered that a positive and negative event will occur. The chronometer then stores two event times in registers.

Figure 32. **Chronometer circuit overview**



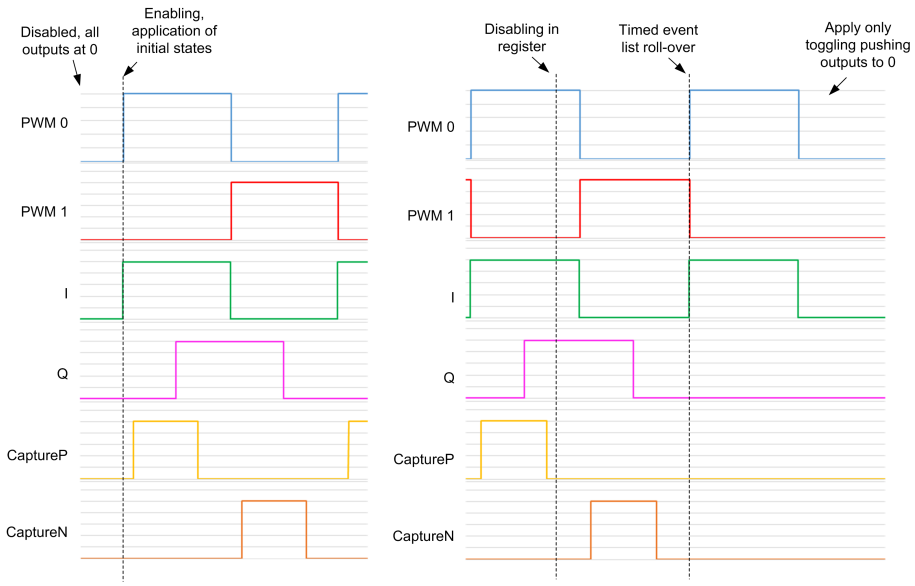
The external events are gated using two outputs of the signal generation block (CaptureP & N). The chronometer tracks the count of cycles that elapsed since the start of PWM cycle. When external event occurs, the on-going count cycles is sampled as well as the phase number at which the event appears. The phase count is readjusted considering the phase selected at the beginning of the signal generation cycle. Then, the adjusted timing is written into the dump registers. In addition to write data in dump registers, the Main PWM provides a chronometer data port to be used externally. This port is connected typically to a phase demodulator. The signal generation block can also be programmed to terminate the current step since an external event occurs. The signal generation block can also fake an event on a defined polarity based on the timing of the other polarity.

The event list of signal generation block is copied locally so that programming of new list can be done during PWM operation. The update of the list can be asynchronous or synchronous with the rollover of the event list processing.

2.2.1.3.4 Enabling/Disabling

The Main PWM supports smart enabling and disabling:

Figure 33. **Sequencer timing diagram overview**



The disabling sequence is starting when list rolls over.

2.2.1.4 *Programming*

This section gives details about the HR-PWM SW configuration.

2.2.1.4.1 Sequencer configuration

The HR-PWM provides PWM0 & 1 primary outputs. PWM0 drives a half-bridge or one side of a full-bridge. PWM1 drives the other side of a full-bridge.

The HR-PWM provides also the complementary output of PWM0 and PWM1. By default, the complementary signals are simply the inverse of the primary outputs. Hence, PWM0 and its complementary drive a half-bridge. PWM1 and its complementary drive the other half-bridge. This default mode relies on the capability of the gate driver to deal with anti-cross conduction.

If the gate drivers do not have anti-cross conduction function, the complementary outputs can be driven by PWM2 & PWM3. In this case, PWM0 & PWM2 drive one half-bridge. PWM1 & PWM3 drive the other half-bridge. To enable this mode, set the bit PWM_2_3_AS_COMPLEMENT in the PWM_CONTROL_3 register.

The HR-PWM executes steps one by one. Each step codes a toggling of PWM outputs, a toggling of time capture windows. Each step can be activated with a particular state of the FSK bit. Each step codes finally a delay before applying the toggling of outputs or capture windows. At the moment the step delay elapses, the HR-PWM applies the step toggling and moves to the next step.

The HR-PWM processes step incrementally from 0 to n (max. step number). The steps associated with an FSK bit polarity different from the running FSK bit are ignored. In the steps that have no output toggling and no capture toggling are ignored. However, it is possible to insert a step that has no output or capture toggling request if FSK and FSKb bits of the step are both set to 1. Such a step can be considered as a simple delay step. When reaching the end of the step list, the HR-PWM rolls over to step 0. By default, the end of the step list corresponds to the first step programmed with no output toggling or capture window having FSK & FSKb at 1. It is possible to force the HR-PWM to scan the entire list of steps (so from 0 to n) before rolling over. To do so, set the bit SCAN_FULL_LIST in the PWM_CONTROL_2 register.

- Sequencer enable:
  - At enabling, the HR-PWM applies an initial state coded in PWM_INITIAL_STATE register to the outputs and capture window. By default, the next output and capture states depend on the toggling coded in the different steps. It is however possible to force the application of PWM_INITIAL_STATE register to the outputs and capture window at each list of step rollover. To do so, set the bit OUTPUT_REINIT in the PWM_CONTROL_2 register.
- Sequencer disable:
  - At disabling, the HR-PWM waits for list of step rollover and then engages a disabling procedure. The steps in the list are processed normally except that only the toggling from 1 to 0 are applied to the outputs and capture window. When all outputs are 0, the HR-PWM is effectively disabled. The state of HR-PWM is monitored by the bit PWM_RUNNING in the PWM_STATUS register.

The enabling/disabling of HR-PWM is done using the bit ENABLE in the PWM_CONTROL_1 register. When HR-PWM is running and ENABLE bit is written to 0, the disabling procedure is engaged. In this case, the disabling time depends on the programming of steps. If an immediate disabling of HR-PWM is required, set the bit FORCE_OFF in the PWM_CONTROL_2 register and wait for the bit PWM_RUNNING in the PWM_STATUS to pass to 0. Then, clear the FORCE_OFF bit.

- Pipeline configuration
  - The HR-PWM runs on a copy of steps registers and the PWM_INITIAL_STATE register. It is then possible to write a new step configuration in registers without disrupting a running sequence. A copy of steps registers and PWM_INITIAL_STATE register is done automatically when the HR-PWM is enabled. A copy can also be triggered while HR-PWM is running to update the sequence. When writing 1 in bit UPDATE_STEPS of the PWM_CONTROL_2 register, a copy is done at next rollover of list of steps. This allows doing a synchronous update of the PWM generation. When writing 1 in bit PUSH_STEPS of the PWM_CONTROL_2 register, the copy of registers is done immediately.

The basic configuration HR-PWM requires the following steps:

1. Configure the complementary mode of PWM0 & 1 (bit PWM_2_3_AS_COMPLEMENT)
2. Write the sequence of steps in the corresponding registers
3. Write the initial state of outputs (PWM_INITIAL_STATE)
4. Set the options of list processing (OUTPUT_REINIT bit and SCAN_FULL_LIST bit)
5. Enable the HR-PWM (ENABLE bit)
6. While running, a new sequence can be written in the registers. When registers write is done, update this new sequence in the HR-PWM sequencer (UPDATE_STEPS bit or PUSH_STEPS bit).
7. The HR-PWM can be gracefully disabled by clearing the ENABLE bit. It will be effectively disabled when PWM_RUNNING bit is 0. Alternatively, the HR-PWM can be immediately disabled using FORCE_OFF bit.

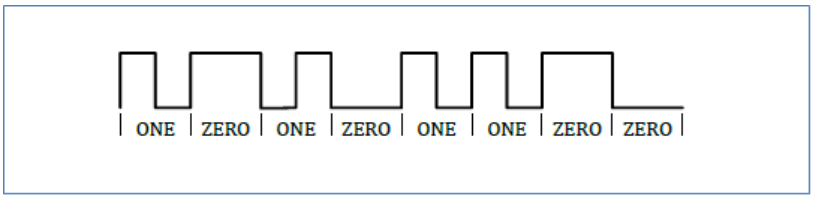
2.2.1.4.2 FSK configuration

The HR-PWM includes an FSK function that allows sending data to a WPC-Qi receiver using Frequency Shift Keying messaging. The variation of PWM frequency is pre-programmed in the list of steps. In fact, the different steps in the list can be associated to a polarity of the FSK signal. When the FSK polarity of a step is not the same as the FSK signal in the HR-PWM, the step is simply ignored. Then, the list can include some steps associated with a particular polarity of FSK. Depending on the running FSK signal polarity, the PWM timings are different as some steps will be active or not. This can lead to different PWM frequencies depending on the state of FSK signal.

For debug or verification, it is possible to force the FSK signal to 1 in a continuous way (by default FSK signal is 0). To do so, set the bit DBG_FSK in the PWM_CONTROL_2 register. This debug mode allows verifying that the FSK enabled sequence is giving the expected result.

The WPC-Qi FSK consists of converting data bits in NRZ symbols. A data bit at 1 gives 2 opposite symbols (01 or 10), a data bit at 0 gives 2 similar symbols (00 or 11). Between each bit, the symbol polarity inverts.

Figure 34. **FSK bit streaming overview**



The WPC-Qi fixes the duration of a symbol to 256 cycles of PWM.

The HR-PWM manages FSK function using a FIFO of 72 symbols. The FIFO is filled with symbols by writing FSK_FIFO_DATA register (8 symbols written per access). When FSK is enabled using the bit FSK_EN of the PWM_CONTROL_1 register and the FSK FIFO is not empty, the HR-PWM gets bits from FIFO and assigns the FSK signal accordingly.

The bit rate of FIFO data sent is configured in the FSK_SYMBOL_DURATION registers. The value corresponds to bit rate counted in number of PWM cycles. A PWM cycle corresponds to a list of steps rollover.

For WPC-Qi compliant FSK, FSK_SYMBOL_DURATION is set to 256. The FIFO can be flushed by setting the bit FLUSH_FSK_FIFO in the PWM_CONTROL_2 register. The PWM_STATUS register gives FIFO status information: FSK_FIFO_EMPTY, FSK_FIFO_FULL and FSK_FIFO_HALF_FULL bits.

The FSK_FIFO_HALF_FULL passes 1 if the quantity of symbols in the FIFO is strictly above 36. This information is useful to reload the FIFO in case of message occupying more than 72 symbols. In fact, since FSK_FIFO_HALF_FULL is 0, it is certain that 36 symbols can be written in the FIFO. Because of 8bits mapping, it is certain that 4 writes to FSK_FIFO_DATA register can be done (so 32 symbols).

The FSK functionality requires the following configuration steps:

1. Configure the duration of a symbol in FSK_SYMBOL_DURATION (256 for WPC-Qi).
2. Upload up to 72 symbols in the FSK FIFO by writing FSK_FIFO_DATA (8 symbols per write).
3. Enable the FSK function by setting 1 the bit FSK_EN in the PWM_CONTROL_1 register.
4. The FSK function drains the FIFO at the configured rate.
5. While some symbols have not been pushed in the FIFO:
   a. Wait for FSK_FIFO_HALF_FULL passing 0.
   b. Write up to 32 symbols in FSK_FIFO_DATA (4 writes).
6. Wait for FSK_FIFO_EMPTY passing 1. That signals that message has been sent.
7. Disable the FSK by writing 0 in FSK_EN bit.

It is possible to let FSK_EN always be 1. In this case, the FSK starts since a write is done in the FSK_FIFO_DATA register. This method imposes that subsequent symbols must be written at a rate higher than the FSK draining one.

2.2.1.4.3 Chronometer configuration

The HR-PWM includes a chronometer function that allows measuring time elapsed between the start of PWM period (so when list of steps rolls over) and a HW event. The HW event can be chosen by programming the field EXT_EVENT_SEL in the PWM_CONTROL_1 register:

1. Quarter wave of LC ringing node
2. Inverse of quarter wave of LC ringing node
3. Zero crossing of bridge current
4. Inverse of zero crossing of bridge current
5. Zero crossing of LC ringing node
6. Inverse of zero crossing of LC ringing node
7. pwm_output[3]
8. Inverse of pwm_output[3]

From the selected HW event, the chronometer captures by default the time for which the event is logic 1 (P polarity) and for which the event is logic 0 (N polarity). These two times are stored in the PWM_TIME_COARSE_P + PWM_TIME_FINE_P registers for P polarity and PWM_TIME_COARSE_N + PWM_TIME_FINE_N registers for N polarity. It is possible to force the N polarity to logic 1 as well. To do so, write 1 to the bit COPY_EVENT of the PWM_CONTROL_2 register.

The time is captured in registers only once. After the capture, the values in the PWM_TIME_x registers are memorized. To capture a new time, the PWM_TIME_x registers must be cleared. This is done by writing 1 to the CLEAR_CAPTURE bit of the PWM_CONTROL_1 register. When a new time is captured, the corresponding bits TIME_N_CAPTURED and TIME_P_CAPTURED in the PWM_STATUS register pass 1.

The time capture is gated by the CAPTURE_P and CAPTURE_N produced by the HR-PWM sequence. The time is effectively captured only if the HW event polarity is reached and the CAPTURE_x corresponding signal is high. The CAPTURE_x signals are useful to mask some glitches present in the HW event signal.

The sensitivity on HW event is by default on level. It can be set to rising edge by writing 1 to the EXT_EVENT_EDGE bit of the PWM_CONTROL_1 register.

Even if the time is captured only once in the PWM_TIME_x registers, the chronometer runs continuously and provides time information to the phase demodulator.

It is possible to choose which time event is provided to the phase demodulator using the bits DEMOD_ON_N and DEMOD_ON_P in the PWM_CONTROL_3 registers. If both DEMOD_ON_N and DEMOD_ON_P are set to 1, the sum of time P and time N is provided to the phase demodulator.

The event mux can select pwm_output[3] as source. Using this, it is possible to qualify the chronometer exactness by programming precise timing of pwm_output[3] toggling.

- Oscillator mode:
  - The HR-PWM can run in oscillator mode. In this mode, the HW event interacts directly with the PWM sequence so that the effective PWM frequency depends on the HW event timing. When this mode is enabled, the HR-PWM sequencer can exit the current running step since the HW event happens. This early exit of step is enabled by the EXIT_ON_EVENT bit of the PWM_CONTROL_1 register.
- During oscillator mode, it is possible to be in a situation where the HW event is valid only for P polarity or N polarity (this is typically the case on the bridge current of a half-bridge circuit). In this case, it could miss a valid event to exit the step.

The HR-PWM is capable of faking the missing event by producing one having the same timing as the valid one. By writing 1 to the bit FAKE_EVENT_N of the PWM_CONTROL_3 register, the HR-PWM generates a fake N event having the same timing than the valid P event. By writing 1 to the bit FAKE_EVENT_0 of the PWM_CONTROL_3 register, the HR-PWM generates a fake P event having the same timing as the valid N event.

The time capture functionality requires the next configuration steps:

1. Be sure that PWM sequence of steps includes the management of CAPTURE_N and CAPTURE_P signals that gate the HW event (CAPTURE_N & P are exclusives).
2. Select the HW event source in the EXT_EVENT_SEL field. If edge sensitivity is needed, set the EXT_EVENT_EDGE bit. If N timing has to be linked to the HW event logic 1, set the COPY_EVENT bit.
3. For each needed time capture:
   a. Set CLEAR_CAPTURE bit to clear the PWM_TIME_x registers.
   b. Wait for TIME_N_CAPTURED and/or TIME_P_CAPTURED bits passing one.
   c. Read the time captured in PWM_TIME_x registers.

Oscillator mode requires the next configuration steps:

1. Configure the chronometer function as detailed above.
2. If needed, fake an invalid event using the FAKE_EVENT_N or FAKE_EVENT_P bits.
3. Set the bit EXIT_ON_EVENT that indirectly configures the HR-PWM in oscillator mode.
4. Enable the HR-PWM.

2.2.1.5 *Operating parameters*

Table 23 summarizes the IP functional target operating parameters.

Table 23. **Main PWM operating parameters**

Parameter	Value
Frequency	40 MHz max.
Phase count	17
Number of signal generation steps	16
FSK FIFO data size	72-bit
Step coarse time value	2 to 4095
Step fine time value	0 to 16
FSK FIFO cycle count	1 to 1023

2.2.1.6 *Configuration register overview*

For the HR-PWM complete configuration register list overview please contact our sales representative.

2.2.2 Phase demodulator

2.2.2.1 *Overview*

The phase demodulator is used to detect the variation of tank phase linked to the Rx modulation. The phase demodulator provides a digital output being the detected modulation.

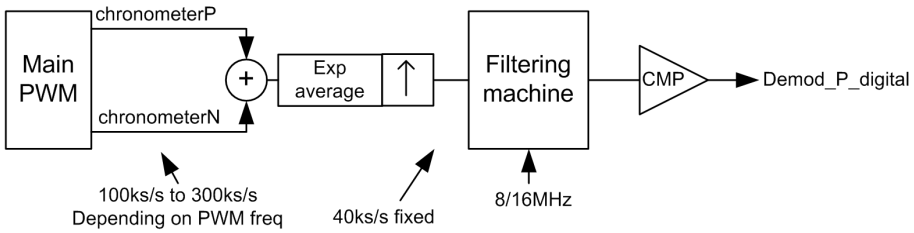
2.2.2.2 *Feature list*

- Variable input frequency decimator
- 3rd order digital LP & HP filter with programmable cut-off frequency
- Comparator with programmable hysteresis

2.2.2.3 *Functional description*

Here below is shown the internal circuit block diagram.

Figure 35. **Phase modulator block diagram overview**



The Main PWM provides chronometer values. These values are sampled in a form coarse + fine.

The phase/frequency demodulator uses chronometer information. The times captured at each half-wave of Main PWM are summed together. The sample rate is a direct function of PWM frequency so variable from 100 ks/s to 300 ks/s. A simple decimation is implemented using an exponential average. The average value is sampled at 40 ks/s fixed sample rate. A filtering machine calculates the required LPF & HPF to select the modulation bandwidth. Finally, a comparator with 0 provides a digital 1bit form of the demodulation.

The exponential average computes the following formulae at each Input sample:

$$Average = Average + Input - \frac{Average}{4}$$

When the 40 ks/s sampling time occurs, the Average/4 value is captured and is used by the filtering machine. After the sampling, the Average value is initialized at the first incoming sample with the following formulae:

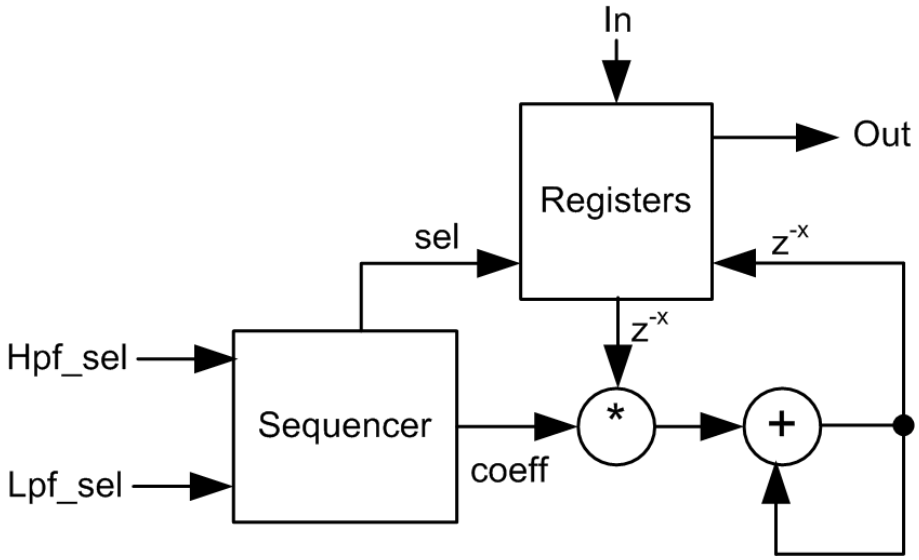
$$Average = 4*Input$$

The filtering machine runs HPF and LPF filters to select the modulation bandwidth. The filters have the following equations:

$$H(z) = \frac{a_0 + a_1.z^{-1} + a_2.z^{-2} + a_3.z^{-3}}{1 + b_1.z^{-1} + b_2.z^{-2} + b_3.z^{-3}}$$

To perform the required operation, the filtering machine has the following structure:

Figure 36. Phase modulator filter machine structures overview



The machine is based on a MAC (multiply-and-accumulate) and a set of registers to implement the samples delay. The sequencer drives the MAC operation, selects the delayed sample to be processed and supplies the coefficient. Filter coefficients are fixed but 4 sets exist for the HPF and LPF.

The corresponding frequency response is shown below:

Figure 37. LPF (Low Pass Filter) demodulator frequency response

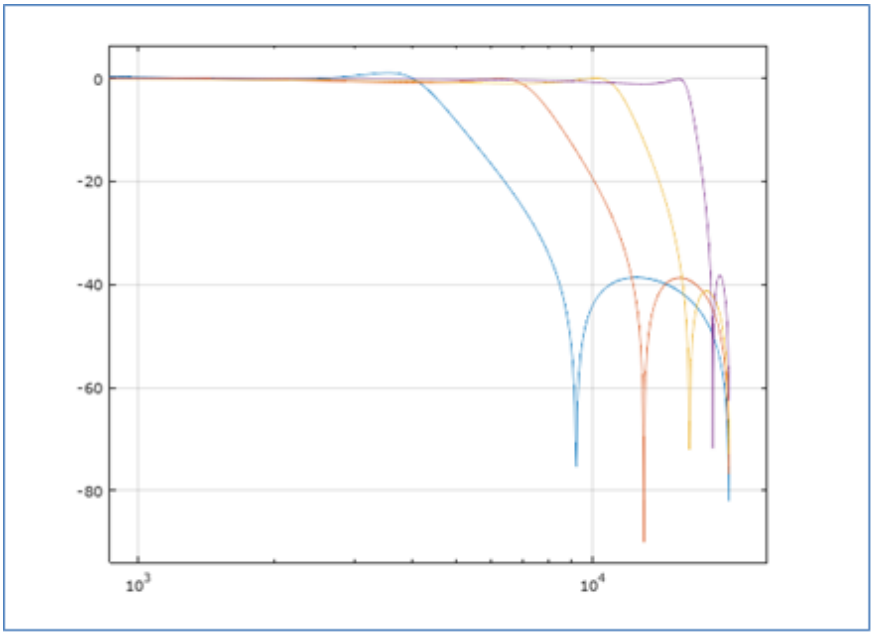
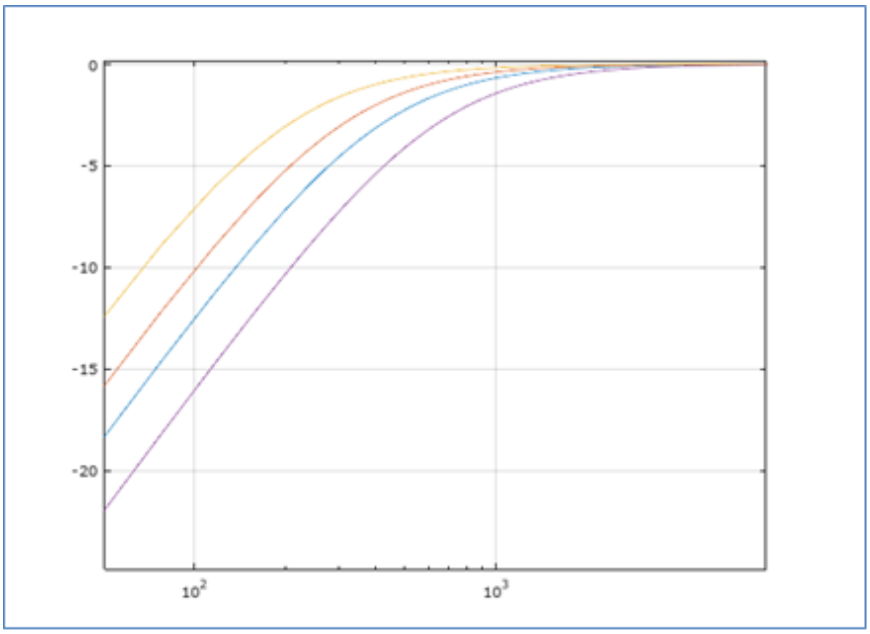


Figure 38. **HPF (High Pass Filter) demodulator frequency response**



The digital comparator compares the output of filter machine with 0. It includes hysteresis. The hysteresis magnitude is configured through the SPI registers.

2.2.2.4 *Programming*

The phase modulator logic requires the following configuration sequences:

2.2.2.4.1 Enable sequence

The phase modulator enabling sequence has the following configuration steps:

1. Configure the P_DEMOD_CLK_L register value
2. Configure the P_DEMOD_CLK_H register value
3. Configure the P_DEMOD_HYSTERESIS register value
4. Program the P_DEMOD register value as follows:
   a. Select the high pass filter frequency coefficient value bit1-0
   b. Select the low pass filter frequency coefficient value bit3-2
   c. Set the Enable bit7

2.2.2.4.2 Reconfigure sequence

When the phase modulator is active the reconfiguration of the filter coefficient value requires the following operations:

1. Disabling the phase modulator by writing 0x00 to the P_DEMOD register
2. Program the P_DEMOD register with the newer value as follows:
   a. Select the high pass filter frequency coefficient value bit1-0
   b. Select the low pass filter frequency coefficient value bit3-2
   c. Set the Enable bit7

2.2.2.4.3 Disable sequence

The phase demodulator disabling sequence requires to clear the P_DEMOD register value.

2.2.2.5 *Operating parameters*

Table 24 summarizes the IP functional target operating parameters.

Table 24. **Phase demodulator operating parameters**

Parameter	Condition	Value
Operating frequency		8 MHz to 16 MHz
Chronometer data rate		100 – 300 ks/s
Target sampling rate		40 ks/s
LPF frequencies	Fs=40 ks/s	4 kHz, 8 kHz, 12 kHz, 16 kHz
HPF frequencies	Fs=40 ks/s	200 Hz, 300 Hz, 500 Hz, 650 Hz
Input data width		17 bits
Output data width		1 bit

2.2.2.6 *Configuration register overview*

Table 25. **Phase demodulator configuration registers overview**

Registers overview					
Name	**Description**	**Type**	**Page**	**Offset**	**Reset val.**
P_DEMOD	Phase demodulation configuration	R/W	0	0x1E	0x01
P_DEMOD_CLK_L	Phase demodulation clock configuration	R/W	0	0x1F	0x00
P_DEMOD_CLK_H	Phase demodulation clock configuration	R/W	0	0x20	0x00
P_DEMOD_HYSTERESIS	Phase demodulation hysteresis configuration	R/W	0	0x21	0x00

2.2.3 QF controller

2.2.3.1 *Overview*

This block is used to control the analog driver which stimulates the LC resonant network keeping its natural oscillation at steady-state. Changes in the Q factor allows the system to detect foreign object presence and to check for an in-range receiver periodically (according to the Qi standard) and with minimum SW intervention.

2.2.3.2 *Feature list*

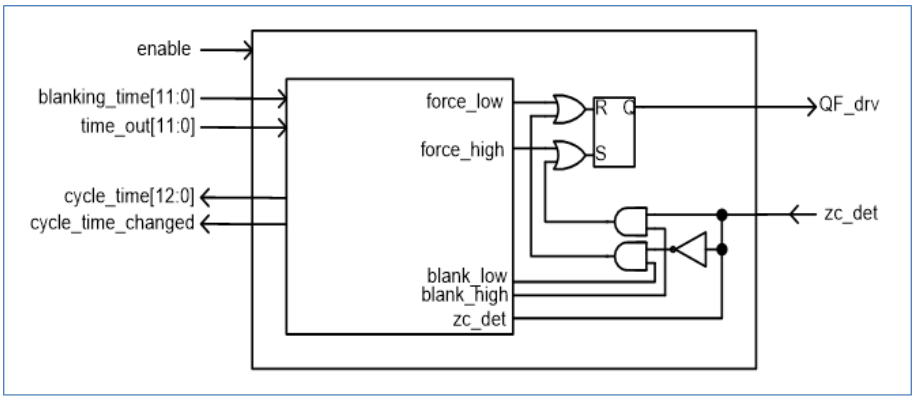
- Blank time to get noise and disturbances immunity (frequency high limit)
- Input capture of waveform zero-cross for oscillation period measurement
- Synchronous FSM controlling an asynchronous loop for optimal capture resolution
- Time-out function to set the frequency low limit

2.2.3.3 *Functional description*

The QF measurement system uses an analog driver providing a symmetrical square wave current of programmable strength. The system uses the IV-converter zero-voltage cross detector attached to the ringing node. The zc_det signal is the output of this detector and is asynchronously fed back to the current generator as source/sink command (QF_drv). A counter in the digital machine is dedicated to capturing the oscillation period and to providing blanking and time-out function. The polarity of the output driver (QF_drv) may be inverted by the POLARITY field of the QF_CTRL control register.

Here below is shown the circuit block diagram.

Figure 39. **QF block diagram overview**



Note that by default the QF circuit is configured with the PLL source clock.

2.2.3.3.1 Digital FSM

The digital machine includes the asynchronous RS path to close the loop with the analog cell.

The controller uses 12bits counters to process the blank time, the time-out and to evaluate the ringing half-period. The cycle_time output is the sum of two consecutive half-period durations so that the full period of ring is available.

The digital machine is clocked with the PLL by default to get enough timing resolution of the ringing period. It can be also clocked by the AFE clock to reduce current consumption accepting a lower timing resolution.

2.2.3.3.2 Startup oscillation

The QF controller drives the current generator synchronously with the tank zero cross. Hence, the QF controller and the LC tank form an oscillator where the LC tank contributes to the loop delay. However, at startup, the LC tank is not yet ringing so that no zero cross event can happen to synchronize the controller. By default, the current generator polarity is reverted when the time-out time is reached. By consequence, at startup, the QF controller drives the tank at an effective period of 2 times the time-out value. This period represents the effective start-up period.

2.2.3.3.3 Spread functionality

If the start-up period is far away from the resonating point of the LC tank or if the tank is very selective, the LC oscillation is small. Considering some offset in the zero-cross comparator, these oscillations can be small enough so that they never trig the comparator. If this happens, the system does not run an oscillator mode and stays at the start-up period.

To avoid this situation, the QF controller integrates a spread function that can be enabled independently. When the QF system is not locked in oscillator mode, the spread function changes the effective time-out time at each loop of the state machine.

The effective time-out is incremented from blanking-time value to the programmed time-out value by 1/8 steps (so each step is (time-out – blanking-time)/8).

This time-out variation allows exciting the LC tank with a variable frequency and a variable duty cycle. By consequence, the LC tank excitation could come to be closer to the resonant frequency. Then, the ring voltage enlarges enough to pass the zero-cross comparator threshold and the system can expect to lock in oscillator mode.

The QF controller reports a locked status information. Locked status bit is set if the controller detects a zero-cross happening before the time-out. If the time-out occurs, the locked bit is cleared.

The blanking time and time-out values can be freely modified when the QF controller is disabled. When QF controller is running, the values are synchronized using an update bit present in the registers.

2.2.3.4 *Programming*

The QF controller builds an oscillating loop through the LC tank of the transmitter. To do so, the controller drives the analog current generator polarity synchronously with a polarity feedback of the LC tank. The polarity feedback is typically made with a voltage zero-cross detector probing the common node of the L and the C. When the QF controller runs a normal case, the resulting oscillating frequency is equal to the self-resonant frequency of the LC tank.

2.2.3.4.1 Oscillation width

The QF controller limits the oscillating period to a minimum and maximum value. This allows the locking of the oscillation.

- **The minimum period** is set in the QF_BLANKING_TIME registers. The blanking value corresponds to the minimum time between each toggling of the current generator. Hence, the programmed value corresponds to the minimum half period of the oscillation.
- **The maximum period** is set in the QF_TIMEOUT registers. The time-out value corresponds to the maximum time between each toggling of the current generator. Hence, the programmed value corresponds to the maximum half period of the oscillation. When the time-out time is reached, the current generator polarity is forced to toggle. If the LC tank polarity comparator toggles between the blanking and the time-out times, the QF controller toggles the current generator polarity.

If the zcd_det signal coming from the zero-cross comparator toggles between the blanking and the time-out times, the QF controller toggles the current generator polarity QF_drv

2.2.3.4.2 Cycle time oscillation

The QF controller tracks the time taken between each toggling of the current generator polarity. It sums the time taken when polarity is low and the time taken when polarity is high. The results are available in the QF_CYCLE_TIME registers. The cycle_time value corresponds directly to the period of oscillation.

The timing of the QF controller is expressed in clock cycles of the selected source operation.

The drive current is tunable in the QF_ISET register.

2.2.3.4.3 Programmable current level

The drive current is programmable through the QF_ISET register. Depending on losses of the tank, the user has to set the correct current level to make the LC network oscillate around 1 V_{pp}.

2.2.3.4.4 Enable sequence

The QF controller requires the following configuration sequences:

1. Selection frequency operation
2. Program the blanking and time-out times. Typically the blanking time is 0.75 times the typical half period of the LC tank and time-out is 1.25 times the typical half period of the LC tank.
3. Set the current generator drive current as needed for the tank.
4. In the QF_CTRL control register, clear the SHORT bit, choose the POLARITY needed and set the ENABLE bit (this can be done in a single access). The POLARITY bit inverts the polarity of the current generator to adapt to an inverted configuration of the tank.
5. Wait for the oscillation to lock (1 ms to 10 ms).
6. Get the cycle time measured by the QF controller from the corresponding register.

2.2.3.4.5 Disable sequence

The disabling of QF controller consists in writing the QF_CTRL control register with the ENABLE bit cleared.

The current generator injects current through a capacitor connected to the LC common node. During power transfer, this LC node rings up to 200 V_{pp}.

To avoid damaging the QF current generator due to this high voltage, the QF driver pin of the STWBC2-HP must be shorted to GND during power transfer phase.

The on-chip NMOS short switch is enabled by setting the bit SHORT in the QF_CTRL control register.

2.2.3.4.6 Change timing on-the-fly

Blanking time and time-out can be changed on-the-fly. To do so, write the new value and set the bit UPDATE of the QF_BLANKING_TIME_H or QF_TIMEOUT_H registers respectively. The update can be done optimally by writing the low byte first and then writing the high part of the value ORed with 0x80. 0x80 corresponds to the UPDATE bit.

2.2.3.5 *Operating parameters*

Table 26 summarizes the IP functional operating parameters:

Table 26. QF controller main parameters

Parameter	Condition	Value
Operating frequency		8 MHz – 40 MHz
Measurable ring frequency	clk = 40 MHz	>4.9 kHz
	clk = 16 MHz	>1.9 kHz
	clk = 8 MHz	>0.9 kHz

2.2.3.6 *Configuration register overview*

Table 27 shows the list of QF configuration registers overview:

Table 27. QF configuration registers overview

Registers overview					
Name	**Description**	**Type**	**Page**	**Offset**	**Reset val.**
QF_CTRL	Control configuration	R/W	0	0x14	0x10
QF_BLANKING_TIME_L	Blanking time low value	R/W	0	0x15	0x00
QF_BLANKING_TIME_H	Blanking time high value	R/W	0	0x16	0x00
QF_TIMEOUT_L	Time-out low value	R/W	0	0x17	0x00
QF_TIMEOUT_H	Time-out high value	R/W	0	0x18	0x00
QF_CYCLE_TIME_L	Cycle time low value	R/W	0	0x19	0x00
QF_CYCLE_TIME_H	Cycle time high value	R/W	0	0x1A	0x00
QF_ISET	Drive current setting	R/W	0	0x1B	0x00

2.2.4 Gate driver mode, IO mode and mux

2.2.4.1 *Overview*

This logic generates the gate driver and the I/O signals.

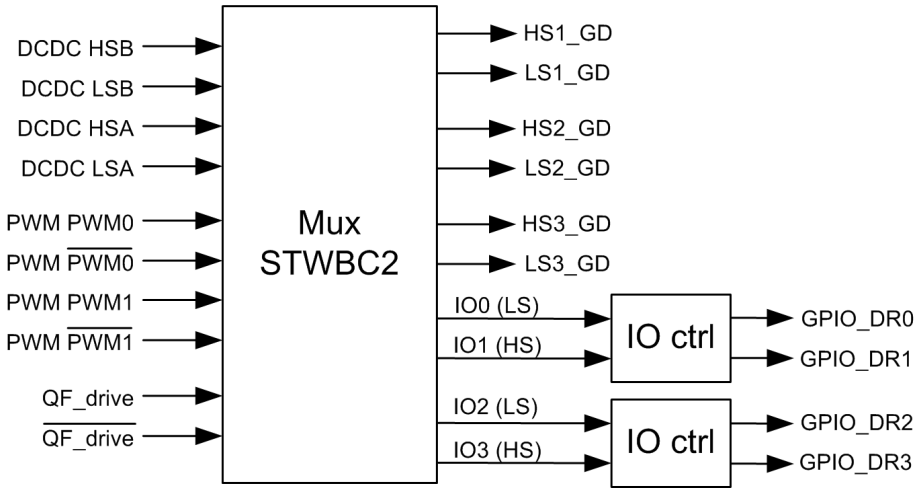
2.2.4.2 *Feature list*

- Control embedded gate driver operations
- Control IO line mode operation
- Control mux signals routed to GD or I/O lines

2.2.4.3 *Functional description*

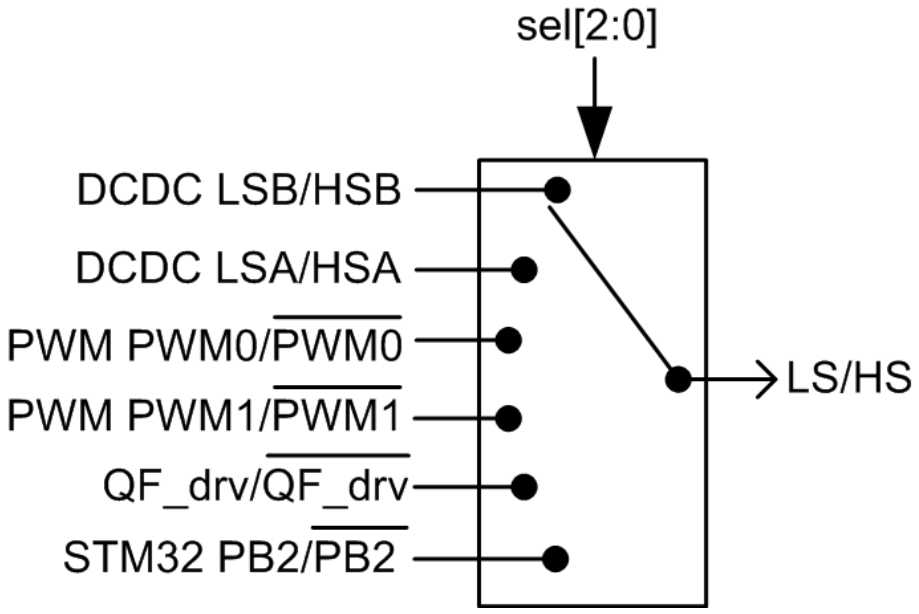
The STWBC2-HP embeds three gate drivers and four configurable external I/O lines (GPIO_DR0 to GPIO_DR3). A mux allows routing the different generators to the gate drivers or IO control cells:

Figure 40. **Gate driver and IO signal overview**



The mux logic is composed of individual mux instance cells as shown below in Figure 41.

Figure 41. **Multiplexer control logic**

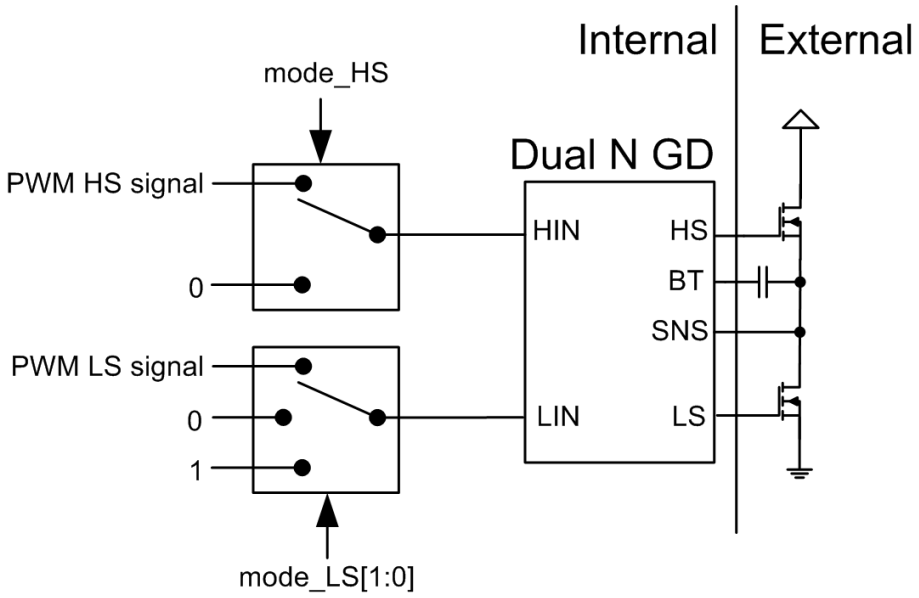


Each cell allows connecting a LS/HS couple source to a LS/HS couple output.

For an asynchronous buck/boost case requiring connecting HSB and LSA for BB-DCDC to a single gate driver, the BB-DCDC integrates the capability to swap LSA and LSB on its output. Hence, the BB-DCDC source is LSA/HSB instead of LSB/HSB.

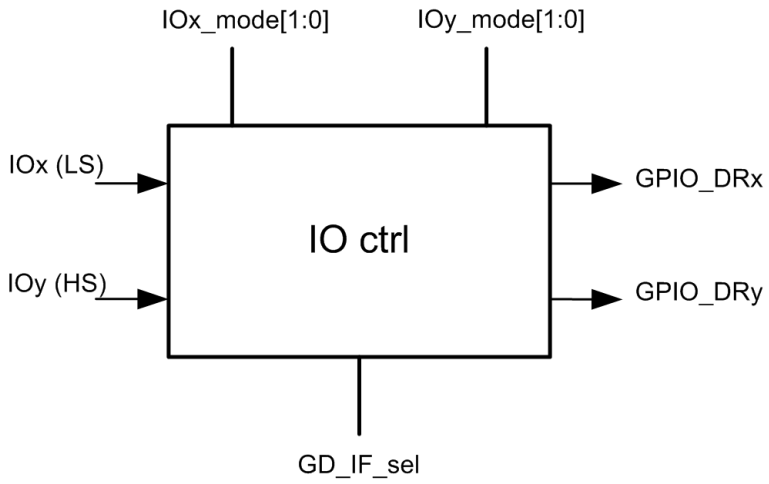
The embedded gate drivers can be configured to ignore any PWM signal and to drive a constant high or low state.

Figure 42. **Gate driver control signal configuration**



The IO mode is controlled per pair GPIO_DR0/1 and GPIO_DR2/3 by the following logic:

Figure 43. **IO mode configuration**



Note that some signals in the above figure may be not aligned with the RTL implementation.

- **IOx_mode[1:0] & IOy_mode[1:0]**: Configure the mode of operation of each I/O pin:
  - **00**: high Z
  - **01**: push-pull output at 0
  - **10**: push-pull output at 1
  - **11**: output driven by HS_in or LS_in
- **GD_IF_sel**: Select the interface with the external gate driver:
  - **0** : Direct interface, HS_in drives IOx, LS_in drives IOy
  - **1**: PWM/EN interface with IOx behaving as EN signal and IOy as PWM signal.

When GD_IF_sel is 1, the state of IOx/IOy follows the following truth in Table 28:

Table 28. IO selection overview

IO selection overview				
	LS_in = 0 HS_in = 0	LS_in = 1 HS_in = 0	LS_in = 0 HS_in = 1	LS_in = 1 HS_in = 1
IOx (EN)	0	1	1	0
IOy (PWM)	0	0	1	1

Table 29 provides the effective routing of mux to the gate driver and IO control cells.

Table 29. Signal routing

Signal routing						
	HSB/LSB selected	HAS/LSA selected	PWM0 selected	PWM1 selected	QF drv selected	PB2 selected
GD LS input	LSB	LSA	~PWM0	~PWM1	~QF_drv	~PB2
GD HS input	HSB	HSA	PWM0	PWM1	QF_drv	PB2
IO LS input	LSB	LSA	~PWM0	~PWM1	~QF_drv	~PB2
IO HS input	HSB	HSA	PWM0	PWM1	QF_drv	PB2

2.2.4.4 *Configuration register overview*

Table 30 shows the list of GD_IOmux configuration registers overview:

Table 30. GD_IOmux configuration registers overview

Registers overview					
Name	**Description**	**Type**	**Page**	**Offset**	**Reset val.**
GATE_DRIVER_1_CFG	Gate Driver 1 configuration	R/W	0	0x2C	0xY0(1)
GATE_DRIVER_2_CFG	Gate Driver 2 configuration	R/W	0	0x2D	0xY0(1)
GATE_DRIVER_3_CFG	Gate Driver 3 configuration	R/W	0	0x2E	0xY0(1)
GATE_DRIVER_IO_0_1_CTRL	GD_IO 0 & 1 control	R/W	0	0x2F	0xY0(1)
GATE_DRIVER_IO_2_3_CTRL	GD_IO 2 & 3 control	R/W	0	0x30	0xY0(1)
GATE_DRIVER_IO_MUX	Mux configuration for GD_IO	R/W	0	0x31	0x00

1. *Refer to register definition.*

2.2.5 ADC_MUX

2.2.5.1 *Overview*

This block generates the ADC selection lines of an eight-way analog multiplexer.

2.2.5.2 *Feature list*

- Programmable multiplexed line selection
- Pipeline ADC channel selection line
- Three independent module instantiated

2.2.5.3 *Functional description*

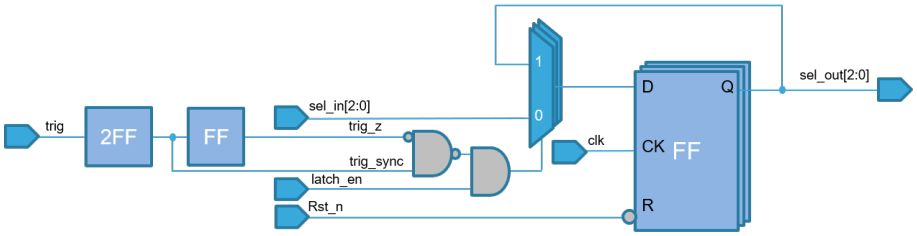
The IP generates the ADC selection lines interconnected to an eight-way analog multiplexer; this IP is instantiated three times (one instance per ADC channel). The sel_in[2:0] lines are provided by the ADC_MUX_<x>_ADC_MUX_<x> internal registers while the latch_en signal is interconnected with the register bit ADC_MUX_<x>_LATCH_ADC<x>_EN. The trig signal is interconnected with the adc_mux_trig signal generated by a GPIO signal; this command allows to pipeline the analog mux channel selection during fast ADC conversion.

When the channel selection is configured in latch mode, the Adc_mux_trig signal latches the mux selection lines. The Adc_mux_trig comes from the STM32 GPIO signal. When using latch mode, the FW can prepare the next mux selection by accessing the SPI registers while an ADC conversion is on-going. This selection is applied only after the Adc_mux_trig GPIO toggling. In this way, the SPI access time can be pipelined in respect to the ADC conversion improving the bandwidth conversion.

Note that "<x>" in the register name means that the bit field is replicated three times to be interconnected with all instance modules.

Here below is shown the internal circuit block diagram:

Figure 44. ADC_MUX block diagram overview



2.2.5.4 *Configuration register overview*

Table 31 shows the list of ADC_MUX configuration registers overview:

Table 31. ADC_MUX configuration registers overview

Registers overview					
Name	**Description**	**Type**	**Page**	**Offset**	**Reset val.**
ADC_MUX_0	Mux0 to ADC0	R/W	0	0x05	0x00
ADC_MUX_1	Mux1 to ADC1	R/W	0	0x06	0x00
ADC_MUX_2	Mux2 to ADC2	R/W	0	0x07	0x00

2.2.6 LED_SPK pattern generator

2.2.6.1 *Overview*

This block generates a configurable pattern output signal in the range of Hz for LED monitor or kHz for the buzzer audio speaker.

2.2.6.2 *Feature list*

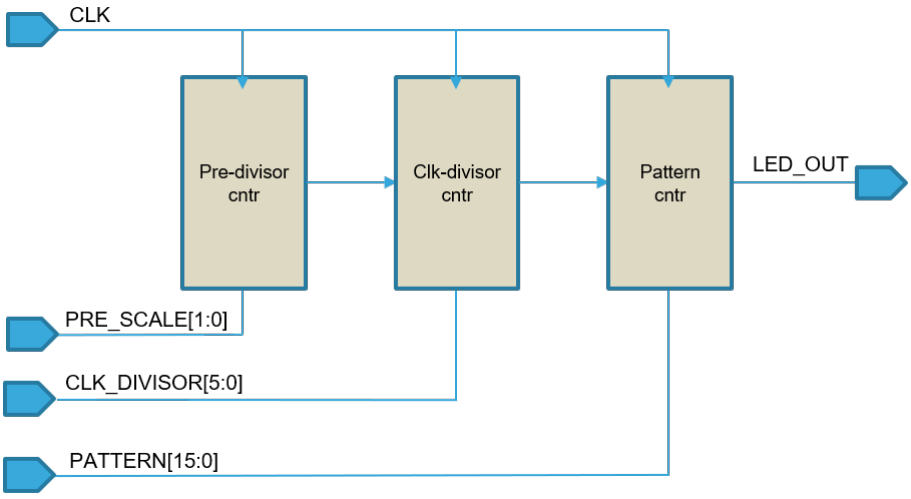
- Programmable pulse pattern generator
- Configurable parameters:
  - Pattern output data
  - Clock pre-scale
  - Clock pre-divider
  - Current sink
- Control enable functionality
- Two independent modules instantiated
- Alternatively, I/O signal configuration

2.2.6.3 *Functional description*

This IP generates a repetitive programmable pulse pattern (usable with LED or Buzzer circuit) accordingly with the register parameter's configuration values. Alternatively, the LED signal may be used as a general I/O signal controlled by SW.

Here below is shown an outline view of the LED pattern generator block diagram:

Figure 45. LED_SPK block diagram overview



2.2.6.4 *Programming*

After power-up the IP is disabled; the usage of this requires the following program sequences:

2.2.6.4.1 Pattern generator

1. Configure the 16-bit repetition data pattern through the registers: LED_BUZZER_<x>_PATTERN_L and LED_BUZZER_<x>_PATTERN_H.
2. Configure the pulse frequency by programming the clock pre-scale (Prescale) through register field LED_BUZZER_<x>_CLK_DIV[1:0] and clock division (Clkdivision) via register field LED_BUZZER_<x>_CLK_DIV[7:2].

   The pulse pattern frequency is generated according to the following equation:

$$Fpulse = \frac{Fafe}{((Clk_{division} + 1) * Pre_{scale})}$$

   Table 32 summarizes the range of pulse frequency configurable with the register parameters assuming Fafe = 16 MHz:

Table 32. Pulse range configuration values

Pulse frequency configuration				
Pre_{scale}	Step division	$Clk_{division}$ = 0	$Clk_{division}$ = 126	Unit
00	256	62500	492.126	Hz
01	2048	7812.5	61.51575	
10	16384	976.5625	7.689469	
11	131072	122.0703	0.961184	

3. Selecting the I/O driver mode operations and current sink configuring respectively the register fields LED_BUZZER_<x>_CFG[2:0] and LED_BUZZER_<x>_CFG[4:3].
4. Enabling the pulse logic generation by clearing the register field LED_BUZZER_<x>_CFG[5] enable bit.

2.2.6.4.2 I/O signal

1. Disabling the pattern generator by setting the LED_BUZZER_<x>_CFG[5] enable bit.
2. Direct controls of the I/O signal through the LED_BUZZER_<x>_IO[1:0] register bits.

2.2.6.5 *Configuration register overview*

Table 33 shows the list of LED configuration registers overview:

Table 33. LED configuration registers overview

Registers overview					
Name	**Description**	**Type**	**Page**	**Offset**	**Reset val.**
LED_BUZZER_1_CFG	Led/Buzzer driver configuration	R/W	0	0x22	0x00
LED_BUZZER_1_IO	Led/Buzzer I/O State	R/W	0	0x23	0x0Y(1)
LED_BUZZER_1_PATTERN_L	Low byte of pattern	R/W	0	0x24	0x00
LED_BUZZER_1_PATTERN_H	High byte of pattern	R/W	0	0x25	0x00
LED_BUZZER_1_CLK_DIV	Clock division	R/W	0	0x26	0x00
LED_BUZZER_2_CFG	Led/Buzzer driver configuration	R/W	0	0x27	0x00
LED_BUZZER_2_IO	Led/Buzzer I/O State	R/W	0	0x28	0x0Y(1)
LED_BUZZER_2_PATTERN_L	Low byte of pattern	R/W	0	0x29	0x00
LED_BUZZER_2_PATTERN_H	High byte of pattern	R/W	0	0x2A	0x00
LED_BUZZER_2_CLK_DIV	Clock division	R/W	0	0x2B	0x00

1. *Refer to register definition.*

2.2.7 BB-DCDC

2.2.7.1 *Overview*

The BB-DCDC is a digital controller for buck, boost, buck-boost DC to DC conversion.

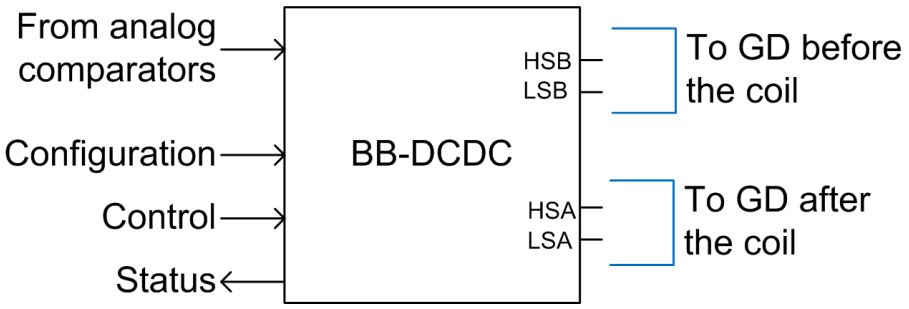
2.2.7.2 *Feature list*

- Synchronous clocked machine
- Uses only comparators as analog resources
- Configurable as boost, buck or buck-boost
- Run burst, DCM, QR and CCM conversion modes
- Detects overload
- Can regulate VIN drop
- Can limit output current

2.2.7.3 *Functional description*

Here below is shown the group of signals handled by the BB-DCDC.

Figure 46. **BB-DCDC block diagram overview**



2.2.7.4 *Charging step*

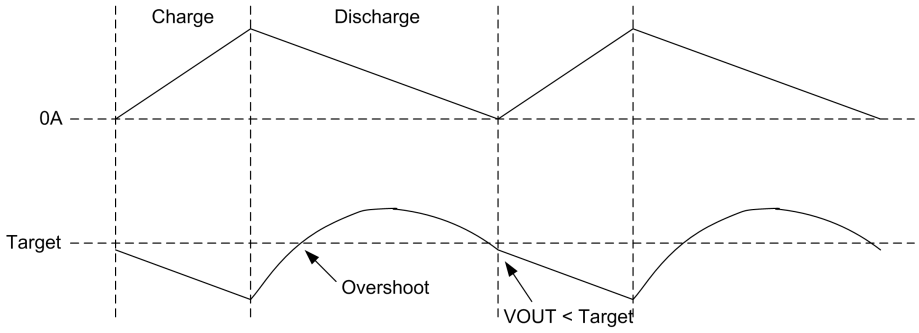
The digital DC-DC controller uses a variable charging time to control the current in the coil:

- The charge time is increased if the DC-DC does not reach the target regulation voltage during the discharge phase. So, the next cycle furnishes more energy. The charge time increase is only applied if the coil current does not reach the max. value.
- The charge time is decreased if the target regulation voltage is overshot at the beginning of the discharge phase. So, the next cycle furnishes less energy.
- The charge time is also decreased if the output voltage is above the target at the end of the discharge step.
- If the output voltage overshoots the target regulation voltage around the middle of the discharge phase, the charge time is not modified.

2.2.7.5 *Boost in QR mode*

The drawing below shows a case of boost in QR mode where the control loop has converged. During the charge phase, the output voltage drops because of the DC-DC load. During the discharge phase, the output voltage passes above the target but not too early. So, the charge time is not changed. Furthermore, as the output voltage is below the target at the end of discharge phase, a new charge cycle is engaged immediately.

Figure 47. **QR mode converged loop operation**

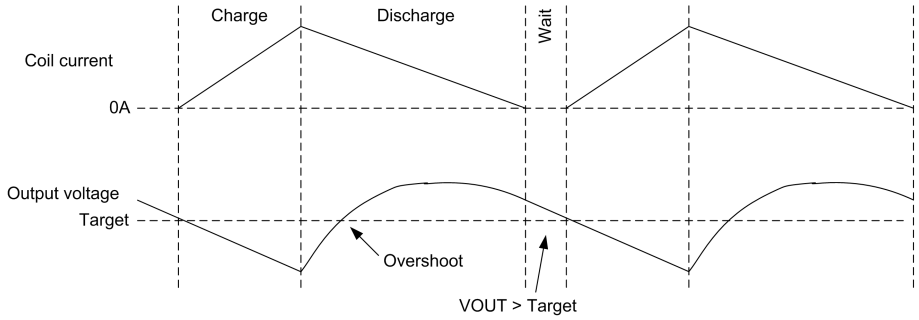


2.2.7.6 *DCM case*

The drawing below shows a DCM case where the output voltage is above the target at the end of the discharge step. In this situation, the charge time is probably decreased because the output voltage overshoots the target too early during the discharge step.

The charge time is also decreased because the output voltage stays above the target at the end of the discharge phase. In addition, after the discharge step, the DC-DC controller waits for output voltage to be below the target to start a new charge cycle. This is the way DCM and burst mode can operate.

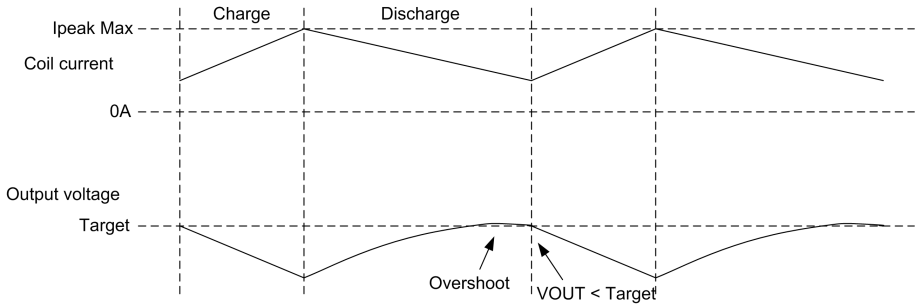
Figure 48. **DCM operation output voltage above target**



2.2.7.7 *CCM case*

The drawing below shows a CCM case which happens because the coil current reached the maximum peak current allowed for the coil. In this case, the charge time is not increased.

Figure 49. **CCM Icoil reach max. peak**



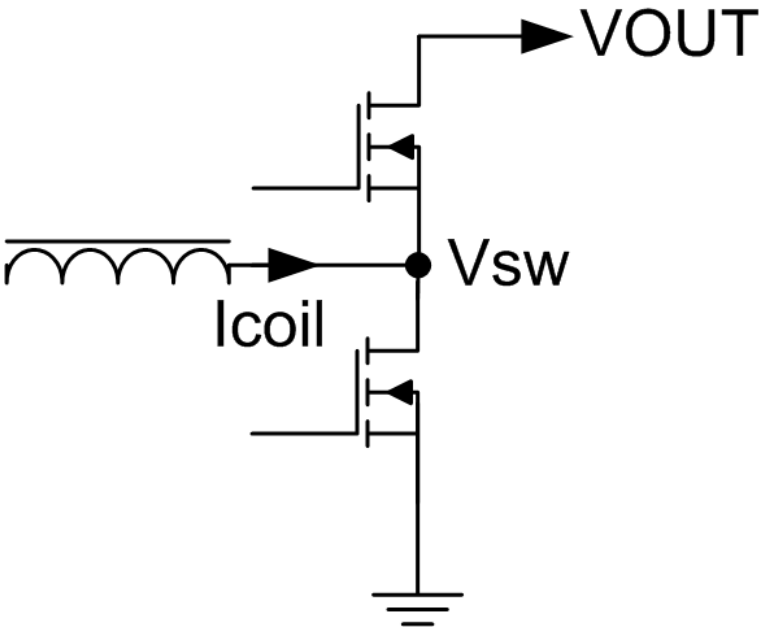
2.2.7.8 *Discharge and Probing*

As for the charge step, the discharge step is controlled by its duration. The discharge step is adjusted to target a QR operation while ensuring that current in the coil is not reverting.

The current reversal detection during the discharge step is done using an additional probing step placed right after the discharge step. During this probe step, the output stage of the converted is placed in high impedance (both high- and low-side MOS are OFF):

- If the current in the coil I_{coil} was still flowing to VOUT before the probe step, the switching node voltage V_{sw} rises up to VOUT+0.7 V through the high-side MOS body diode.
- If the current in the coil I_{coil} has reverted before the probe step, the switching node voltage V_{sw} drops down to -0.7 V through the low-side MOS body diode.

Figure 50. **Vsw switch node voltage**



By consequence, checking the V_{sw} voltage during the probe step is sufficient to conclude that current in the coil has reverted or not.

The current reversal information is combined with others to adjust the discharge time:

- If current is not reverted and max. peak current in the coil is not reached, the discharge time is enlarged. This offers more chance to drain completely the coil current during the discharge phase in order to run QR mode.
- If current has reverted or max. peak current in the coil is reached, the discharge time is decreased. Reducing the discharge time helps avoiding the over-discharging of coil. In case of max. peak current, reducing the discharge time pushes the controller in CCM mode.
- If the output voltage overshoots the target at the end of the discharge step, the discharge time is increased. This pushes the DC-DC duty cycle in the direction of lowering the output voltage.

2.2.7.9 *Check step*

The check step is simply stalling the converter while the output voltage is above the target. A new charge/discharge cycle is engaged only if the output voltage is lower than the target.

2.2.7.10 *Refresh step*

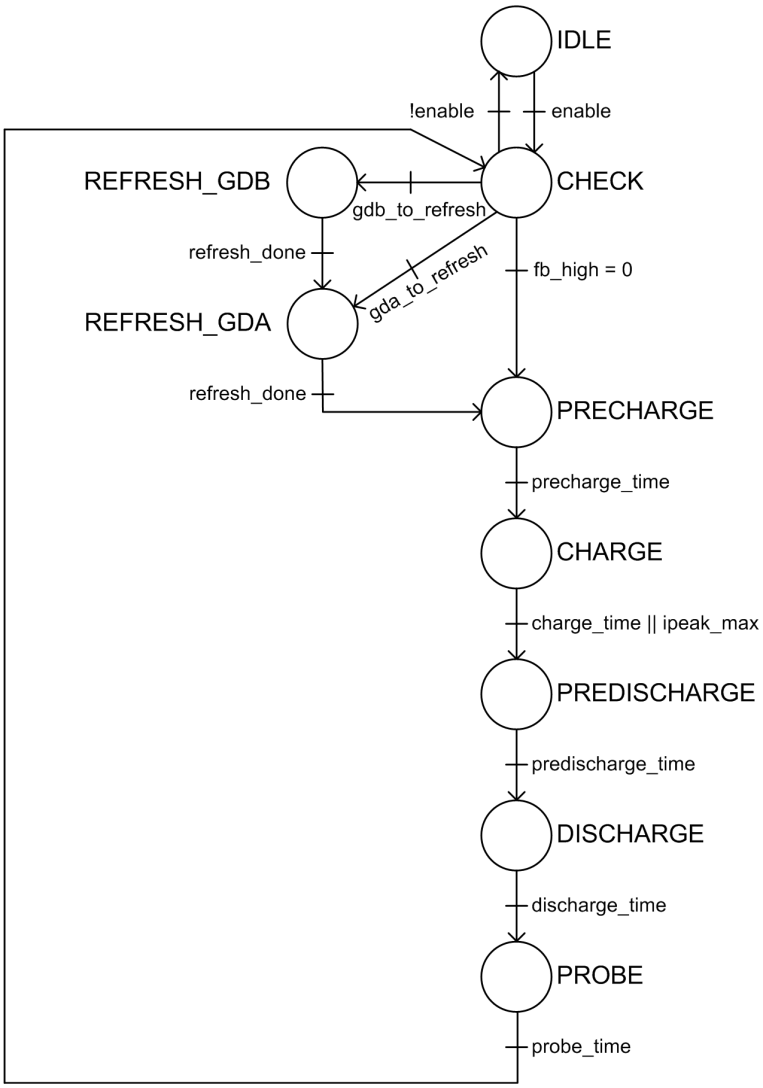
Depending on the DC-DC converter topology and gate driver usage, it is possible that a toggling of a gate driver is required to recharge the bootstrap capacitor. This is the role of these steps.

2.2.7.11 *FSM control logic*

The BB-DCDC main operations are controlled by an embedded finite state machine control logic.

Figure 51 shows the controller FSM state diagram:

Figure 51. **BB-DCDC FSM state diagram overview**



Note that some signals in the flow diagram may be not aligned with the RTL implementation.

2.2.7.11.1 FSM state overview

Here below are listed all the FSM states:

- IDLE: enabling waiting state
- CHECK: check feedback comparator and start a new cycle
- REFRESH_GDB: refresh gate driver B
- REFRESH_GDA: refresh gate driver A
- PRECHARGE: charge the inductor for a fixed amount of time
- CHARGE: charge the inductor for a variable time or up to max. peak current
- PREDISCHARGE: discharge the inductor for fixed amount of time
- DISCHARGE: discharge the inductor for a variable time
- PROBE: probe the switching node to check inductor current reversal

2.2.7.11.2 Overcurrent case

When an overcurrent situation has been detected during the CHECK state, the charge and discharge timings are updated systematically with a specific rule. The goal of this rule is to regulate the current information to stay at the limit of overcurrent.

The algorithm allows a pseudo steady-state operation even if the overcurrent comparator is fed with a signal low pass filtered with 1 ms time constant.

At least at the beginning of an overcurrent condition, the charging time is updated every fixed number of cycles (the rate is fixed by overcurrent_rate parameter). If the comparator states there is an overcurrent, the charge time is decreased, or else the charge time is increased.

At every cycle, the discharge time is adjusted in order to stay in QR operation.

When overcurrent comparator is high and charge time is decreased, the output voltage decreases, which reduces the current. At some point, the current is lower than the overcurrent threshold. However, any LPF in the comparator path adds delay before having the comparator toggling low.

Hence, the charge time is decreased much more than needed.

After the charge time decreasing and the overcurrent comparator toggling low, the charge time is increased again, enlarging the current. At some point, the current is above the threshold but the comparator is delayed because of LPF. So, the charge time still increases before the comparator toggles high again. Hence, the current ends higher than the limit.

Based on this fact, since the DC-DC controller has acquired these min & max charge times, it forces the charge time to (max+min)/2 and freezes charge time variation during a programmable time (overcurrent_lock_time).

This freeze allows the LPF to converge.

When the freeze time ends, the charge time is again increased or decreased depending on the state of the overcurrent comparator. By default, the rate of charge time modification is still fixed by the overcurrent_rate parameter. However, if the min & max captured charge times are close together, the rate is fixed by the overcurrent_lock_time parameter which is slower.

If the converter changes his mode (buck or boost or buck/boost), the min & max captured times are cleared which forces a new “learning” phase for overcurrent mode.

During overcurrent mode, the buck mode is allowed only if it was on-going when the overcurrent happens.

To avoid this current oscillation directly linked with the LPF delay, the DC-DC controller stores the on-going charging time at the time the comparator toggles low and at the time the comparator toggles high. These two data represent the peaks of current, one being below the overcurrent threshold, the other one being above the overcurrent threshold. So, it is very probable that a steady-state operation can happen in the middle of these two charge times.

2.2.7.12 *Short-circuit and overload*

During short-circuit or overload, the DC-DC stays far away from target voltage. The DC-DC controller counts at a slow rate the consecutive time the DC-DC stays far away from the target (the rate is dcdc_clk/8192).

If this count is above a programmable threshold, the DC-DC is stopped in error case. It can be enabled again only if it is first disabled from the control bit.

The counter is not enabled during start-up time.

2.2.7.13 *Startup*

When enabled, the DC-DC controller starts with a progressive ramping of input current. The timings are updated with the same calculation than in normal operation. However, the timings are updated at a slow rate of dcdc_clk/8192.

In addition, a stalling time is added between each DC-DC cycle. This time begins at 241 cycle of dcdc_clk and it is decremented by 2, 4 or 16 depending on a programmable start-up rate. This stalling time is updated at a rate of dcdc_clk/8192.

If the output voltage reaches the target during the start-up phase, this one is cleared and the DC-DC operates normally.

2.2.7.14 *Deadtime*

The DC-DC controller embeds a programmable fixed deadtime circuit.

2.2.7.15 *Voltage ramp*

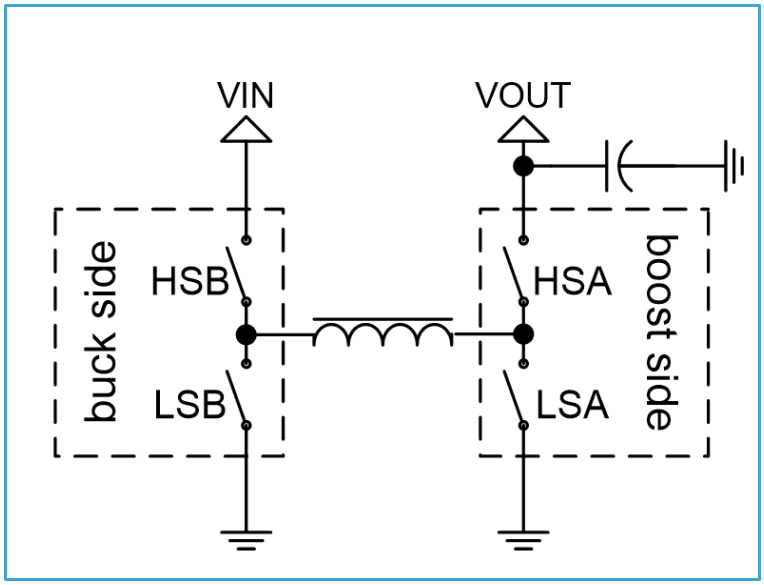
Even if the DC-DC controller embeds a soft-start solution, this does not prevent large input current that can happen when the regulation target voltage rises significantly. In fact, if the converter uses large output capacitor, the converter raises its energy transfer significantly in order to charge the capacitors as fast as possible.

To avoid this large transfer current associated with the rise of target voltage, an automatic voltage ramp can be enabled. When activated, the ramp generator increments the target voltage value at the rate of AFE clock (16 MHz typically) only if the feedback comparator is high and the maximum peak current is not reached. This indirectly produces a rise of target voltage that tracks the progression of output voltage. This avoids having the DC-DC entering in a high energy transfer mode to charge the output capacitor.

2.2.7.16 *Programming*

The BB-DCDC digital controller must be first configured according to the system application topology implemented in the PCB.

Figure 52. BB-DCDC application topology configuration



The conversion type is configured through the DCDC_HW_CONFIG register fields as shown in the following table:

Table 34. BB-DCDC conversion type configuration

Conversion type				
BB-DCDC	**BUCK_ONLY**	**BOOST_ONLY**	**Switch HSB**	**Switch LSA**
Buck	1	0	Active(1)	0
Boost	0	1	0	Active(1)
Buck/boost	0	0	Active(1)	Active(1)

1. *Toggling signal driven from BB-DCDC logic.*

Table 35 shows the Boost HSA switch configuration mode:

Table 35. Boost switch configuration overview

Switch Boost side (after inductor)		
Switch configuration	**HSA_ACTIVE**	**Comment**
High-side switch is diode	0	
High-side switch is MOSFET	1	

Table 36 shows the Buck LSB switch configuration mode:

Table 36. Buck switch configuration overview

Switch Buck side (before inductor)		
Switch configuration	**LSB_ACTIVE**	**Comment**
Low-side switch is diode	0	
Low-side switch is MOSFET	1	

When the BB-DCDC is running a buck/boost hardware with passive switches on low-side buck and high-side boost, a single gate driver can drive the two remaining active switches. This means that HSB+LSA have to drive the gate driver.

However, coherent pairs of BB-DCDC controller signals are routed to the gate driver, so HSB+LSB and HSA+LSA. To obtain HSB+LSA in the gate driver, it is possible to swap LSA and LSB at the output of the BB-DCDC controller. Set the bit SWAP_LSA_LSB of the DCDC_CTRL register to enable the LSA/LSB swapping.

Depending on the application, it is possible to disable some protection functions in the DCDC_CTRL register:

- Set DISABLE_OVER_CURRENT to disable the overcurrent limiter function
- Set DISABLE_VIN_DROP to disable the VIN drop limiter function
- Set DISABLE_OVERLOAD to disable the overload detection function

The STARTUP_RATE field in the DCDC_STARTUP_BLANK_TIME register allows adjusting the smoothing of start-up current. Set this field to the slowest start-up time if possible.

The BB-DCDC supports on-the-fly changes of target voltage. In case of large increase of target voltage, the BB-DCDC can create large input current spikes to reach the required voltage as fast as possible. Automatic voltage ramp can be enabled to reduce such current spikes. To do so, set the bit PROGRESSIVE_V_DAC in the DCDC_CTRL register.

The BB-DCDC must probe the switching node it is driving. BB-DCDC uses V1 comparator to monitor the buck side (if any) and V2 comparator to monitor the boost side (if any). Different pins of the STWBC2-HP can be assigned to V1 and V2 comparator. Use the V1_MUX and V2_MUX fields of the DCDC_COMPARATOR_MUX register to route the V1 and V2 comparators to the right pin (SWx (HB_SW1 or HB_SW2 or HB_SW3) pins of gate drivers or GD_IO3 pin).

The BB-DCDC manages fixed deadtime between high-side and low-side drive. Adjust the DCDC_GD_DEAD_TIME register to set the deadtime required by the gate drivers.

The start of charge step of BB-DCDC can be synchronized with a self-generated clock. Adjust the DCDC_SYNC_PERIOD register to set the period of this clock in counts of BB-DCDC clock period. A value of 0 disables the synchronization. The synchronization clock can be changed during the BB-DCDC operation.

When the new value is written, set the bit UPDATE of the DCDC_CTRL register and the BB-DCDC takes it into account.

The default timing values of the BB-DCDC controller are valid for inductance in the range of 2 µH to 10 µH. It is recommended to keep these default values in the registers

(DCDC_PRECHARGE_TIME_SHORT, DCDC_PRECHARGE_TIME_LONG, DCDC_PRESDISCHARGE_TIME_SHORT, DCDC_PRESDISCHARGE_TIME_LONG, DCDC_PROBE_TIME, DCDC_RELAX_TIME_L, DCDC_RELAX_TIME_H, DCDC_GD_TIME_BEFORE_REFRESH, DCDC_GD_REFRESH_TIME, DCDC_OVERCURRENT_RATE, DCDC_OVERCURRENT_LOCK_TIME, DCDC_OVERLOAD_THR).

The output voltage range of the BB-DCDC has to be configured according to the range of the application. Set the range in the DCDC_DAC_RANGE register.

When the configuration above is done, the DC-DC can be enabled or disabled and the voltage can be changed on-the-fly.

To enable the BB_DCDC set the bit ENABLE in the DCDC_CTRL register. Clear this bit to disable the BB_DCDC.

To change the voltage target, set the corresponding code to the registers DCDC_DAC_VALUE_L & DCDC_DAC_VALUE_H. The written value is taken into account only when the self-clearing bit UPDATE_V_DAC of register DCDC_DAC_VALUE_H is set. It is then recommended to write the L register first and when writing the H register, OR the value with 0x80. This procedure combines the H value write with the update request.

During the DC-DC operation, the DCDC_STATUS register reports the status of the controller. If the BB-DCDC detects a hard error (overload), the DCDC_ERROR bit is set to 1 and the BB-DCDC stops its operation immediately. To restart the BB-DCDC, disable it by clearing the ENABLE bit in the DCDC_CTRL register and enable it again.

2.2.7.17 *Configuration register overview*

For the BB-DCDC complete configuration registers overview please contact our sales representative.

2.2.8 VDoubler

2.2.8.1 *Overview*

The voltage doubler provides approximatively twice the buck DC-DC voltage to power the gate drivers. The digital part is a simple programmable clock source.

2.2.8.2 *Feature list*

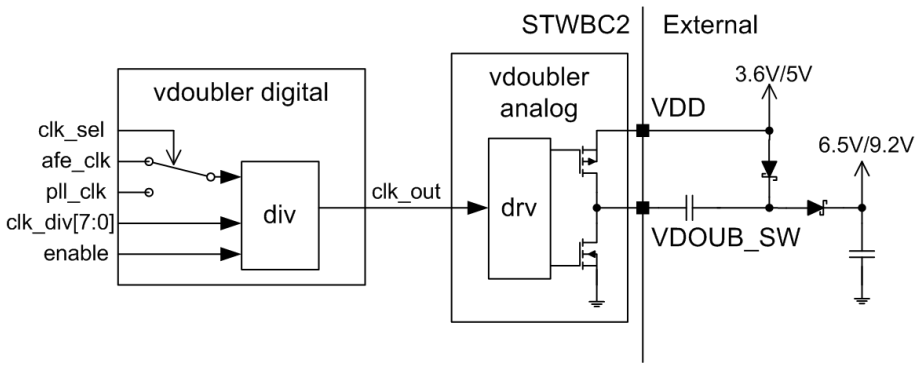
- Provide clock to the analog power buffer
- Select base clock from AFE main clock or PLL

2.2.8.3 *Functional description*

The VDoubler digital module is simply providing a configurable clock to the analog cell. The analog cell buffers this clock with anti-cross-conduction circuit. By using external diodes and capacitor, the buck DC-DC voltage is approximatively doubled.

Figure 53 shows the internal circuit block diagram

Figure 53. VDoubler block diagram overview



2.2.8.4 *Programming*

The VDoubler circuit requires the following program sequences:

- If the Vdoubler circuit is enabled disable this by clearing the V_DOUBLER register.
- Configure the constant division from 2 to 255 in the V_DOUBLER_CLK register.
- Configure the clock source operation through the CLK_SEL bit field of the V_DOUBLER register.
- Enable the Vdoubler circuit by setting the ENABLE bit field of the V_DOUBLER register.

Note that any constant division update or change of frequency requires the iteration of the above procedures.

2.2.8.5 *Operating parameters*

Table 37 summarizes the IP functional operating parameters

Table 37. VDoubler operating parameters

Parameter	Value
Operating frequency	16 MHz or 40 MHz

Parameter	Value
Division factor	2 - 256
Output frequency	100 kHz – 2 MHz

2.2.8.6 *Configuration register overview*

Table 38 shows the list of the VDoubler configuration register overview.

Table 38. VDoubler configuration registers overview

Registers overview					
Name	**Description**	**Type**	**Page**	**Offset**	**Reset val**
V_DOUBLER	Configuration and enabling	R/W	0	0x0B	0x00
V_DOUBLER_CLK	Clock division	R/W	0	0x0C	0x01

2.2.9 ITC (Interrupt Controller)

2.2.9.1 *Overview*

The interrupt controller IP captures signal events and generates a maskable notify interrupt to the MCU companion DIE.

2.2.9.2 *Feature list*

- The IP controls 24 request lines
- Interrupt output request level or pulse
- Configurable parameters:
  - Rising edge interrupt capture request line
  - Maskable interrupt lines
  - Interrupt status interrupt
  - Readable interrupt sources active high

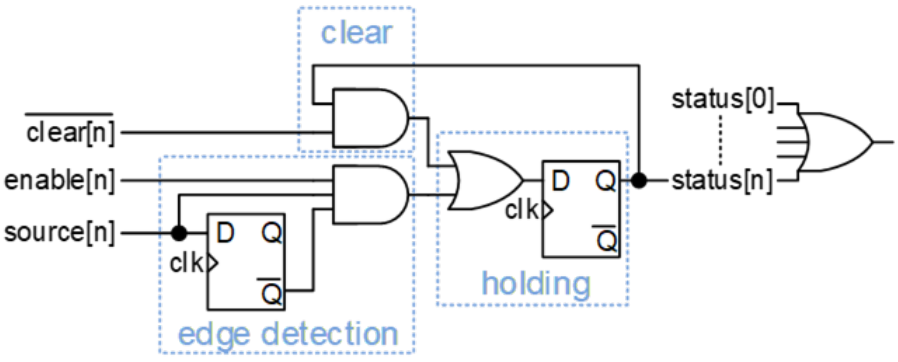
2.2.9.3 *Functional description*

This IP manages all the interrupt request lines within the AFE logic like error events and status events requiring low latency notification and fast recovery actions from the MCU DIE.

The interrupt controller generates an interrupt whenever it detects an interrupt request rising edge transition and the related interrupt line request line is enabled.

Here below is shown an outline view of the ITC block diagram.

Figure 54. ITC block diagram overview



After power-up all the interrupt request lines are masked; these may be enabled by configuring the INT_EN_<x> registers. When the interrupt rising edge is detected the “IRQ” interrupt request signal is asserted.

The interrupt source request is memorized and stored in the readable register INT_STATUS_<x> then the interrupt notify is clearable by setting to “1” the corresponding INT_CLEAR_<x> bit. The interrupt row request lines may be read from the INT_SOURCE_<x> registers.

Note that enabling of a line for which the source is already 1 has no effect.

The status signal of each line is ORed together to generate the IRQ signal going to MCU. The source, enable, clear and status vectors are mapped to SPI registers. Some interrupt source signals are externally synchronized.

2.2.9.3.1 Interrupt level or pulse generation

The interrupt controller can generate a pulse on IRQ output if pulse_duration is not 0. The pulse generation is interesting to use if the microcontroller that receives the IRQ line is edge sensitive only and cannot be configured in level.

When the controller is configured to generate pulse, the IRQ line is forced to 0 for pulse_duration cycle count when any interrupt is cleared. By consequence, if another interrupt was pending, after being forced to 0, the IRQ line returns to 1 and is representing an edge that can be captured by the microcontroller.

2.2.9.3.2 Interrupt table

Table 39 shows the Interrupt request assignment:

Table 39. ITC Interrupt table overview

Interrupt Table		
Irq#	Request line	Description
0	io0_incoherent	IO0 drive incoherence fault
1	io1_incoherent	IO1 drive incoherence fault
2	Io2_incoherent	IO2 drive incoherence fault
3	Io3_incoherent	IO3 drive incoherence fault
4	gdls0_incoherent	gdls0 drive incoherence fault
5	gdls1_incoherent	gdls1 drive incoherence fault
6	gdls2_incoherent	gdls2 drive incoherence fault
7	pll_unlock	PLL lose lock
8	dcdc_error	DCDC error fault
9	dcdc_vin_drop	DCDC voltage drop fault
10	dcdc_over_current	DCDC overcurrent fault
11-17	RFU	Reserved for future use
18	pwm_status_fsk_fifo_half_full	PWM FSK FIFO count is passing below half capacity (time to refill)
19	pwm_status_fsk_fifo_empty	PWM FSK FIFO is empty (end of FSK data)

Note:
1. *All interrupt request signals are active high.*
2. *Register with post-fix “_0” controls the interrupt request lines [7:0].*
3. *Register with post-fix “_1” controls the interrupt request lines [15:8].*
4. *Register with post-fix “_2” controls the interrupt request lines [23:16].*

2.2.9.4 *Programming*

To prevent interrupt missing, the following sequence should be applied during the interrupt controller configuration:

1. Configure INT_PULSE_DURATION to generate pulse on the IRQ line (0 means no pulse).
2. Enable all the interrupt source lines needed to be monitored through the registers INT_EN_<x>.
3. When a source edge happens, the IRQ signal generates an interrupt on the MCU companion DIE.
4. While the IRQ signal is 1, the MCU enters into its interrupt handler. The handler routine (ISR) should perform the following actions:
   a. Get interrupt status by reading INT_STATUS_<x> registers.
   b. Check the status lines at 1 by priority order and decide to treat a first one.
   c. Before treating the chosen, the interrupt, clear it by writing 1 to the corresponding bit in INT_CLEAR_<x> register.
   d. Manage the interrupt and reach the end of interrupt handler (don't loop on the status).
   e. The clearing of the chosen line might release the IRQ line. It is the case if this line was the only one active. In this situation, the interrupt handler exits. If another interrupt were present or if the same interrupt fires again, the IRQ line is 1 at the end of interrupt handler. So, since the handler exits, the MCU re-enters it immediately and the same sequence is processed again (this creates indirectly a loop).

To simplify the SW interrupt handling routine, the fault interrupts are mapped on the lower interrupt request lines.

2.2.9.5 *Configuration register overview*

Table 40 shows the list of ITC configuration registers overview

Table 40. ITC configuration registers overview

Registers overview					
Name	**Description**	**Type**	**Page**	**Offset**	**Reset val.**
INT_EN_0	Interrupt enabling	R/W	0	0x40	0x00
INT_EN_1	Interrupt enabling	R/W	0	0x41	0x00
INT_EN_2	Interrupt enabling	R/W	0	0x42	0x00
INT_CLEAR_0	Interrupt clear	Wsc	0	0x43	0x00
INT_CLEAR_1	Interrupt clear	Wsc	0	0x44	0x00
INT_CLEAR_2	Interrupt clear	Wsc	0	0x45	0x00
INT_STATUS_0	Interrupt status	R	0	0x46	0x00
INT_STATUS_1	Interrupt status	R	0	0x47	0x00
INT_STATUS_2	Interrupt status	R	0	0x48	0x00
INT_SOURCE_0	Interrupt source status	R	0	0x49	0xYY(1)
INT_SOURCE_1	Interrupt source status	R	0	0x4A	0xYY(1)
INT_SOURCE_2	Interrupt source status	R	0	0x4B	0xYY(1)
INT_PULSE	Interrupt pulse configuration	R/W	0	0x4C	0x00

1. *Refer to register definition.*

2.2.10 Protection

2.2.10.1 *Overview*

This block is responsible of the fault error detection inside the Analog Front-End (AFE) logic which is notified to the FW application running on the MCU companion DIE.

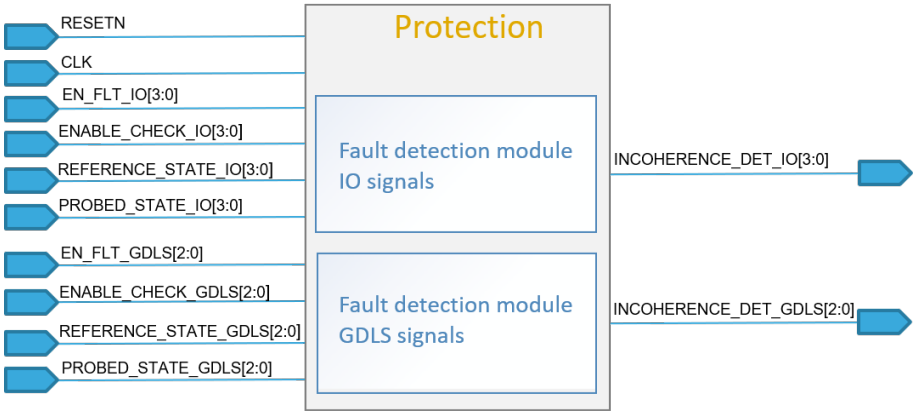
2.2.10.2 *Feature list*

- Fault event capture logic equipped with debounce filter
- Individual enable control line
- IO signals fault detection logic
- GDLS signals fault detection logic
- Notify the fault event to the interrupt controller logic (ITC)

2.2.10.3 *Functional description*

Here below is shown the internal Protection circuit block diagram.

Figure 55. Protection logic block diagram overview



This block detects the fault events raised in the AFE logic during functional operation; most of them are notified directly to the FW for the error management handling sequence, while some of them are controlled by the Protection block.

Once the IP is enabled, by configuring the PROT_CTRL_1 register, the error detection logic is allowed to trigger the fault events. Two class of faults may be detected:

- Hard fault: The highest priority faults due to the undervoltage lockout events; these are detected and managed in the analog logic domain.
- Soft fault: According to the fault type this may be detected by the protection logic and notified to the MCU or may be recognized indirectly by the FW application through the ADC conversion process. In both cases the fault recovery action is submitted to the FW application.

The configuration of the EN_FLTDIS_ANA and EN_FLTDIS_PWM fields of register PROT_CTRL_0 allows to mask the enable functionality respectively of the Analog macros and the PWM signals when one of the following fault events is triggered by the interrupt controller logic:

- IO coherency[3:0]
- GDLS coherency[2:0]
- PLL unlock
- DCDC error

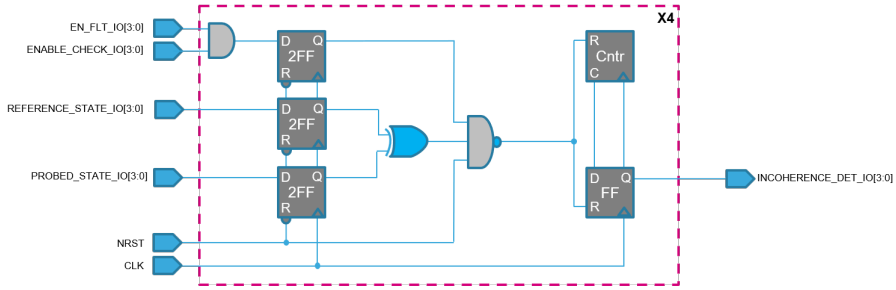
The masking condition is active until the fault error event is recovered by FW.

2.2.10.4 *IO fault detection*

The IO coherence checker verifies that the state of an IO corresponds effectively to the configured value. This allows detecting short-circuit on IO signals.

When enabled (IO data_oe), the cell compares a probed state of IO (data_in) with a reference state (IO data_out).

Figure 56. **IO fault detection circuit overview**



While the probed state is equal to the reference, the counter and the FF used to memorize the fault condition are reset, so incoherence_det_io is 0. When the probed state differs from the reference state, the counter increments. If the counter counts to its maximum, the “RS” is set and incoherence_det_io signal passes 1.

The presence of the counter offers a debounce filter and accommodates for propagation delays. The counter width is set to 3 bits.

The fault signal is a dynamic event in case of toggling IO. It has to be captured by an external logic to be memorized and reported to the MCU. The incoherence_det_io lines are connected to the interrupt controller that latches the request event.

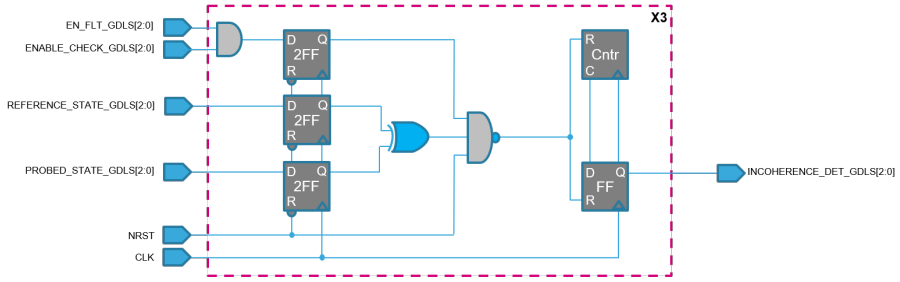
The checkers' outputs are connected to the interrupt controller. This block is instantiating two times.

2.2.10.5 *GDLS fault detection*

The GDLS coherence checker verifies that the state of a GDLS corresponds effectively to the configured value. This allows detecting short-circuit on GDLS signals.

When enabled, the cell compares a probed state of GDLS (data_in) with a reference state (GDLS data_out).

Figure 57. **GDLS fault detection circuit overview**



Note: In the current implementation the “enable_check_gdls[2:0]” vector signal is kept fixed high.

While the probed state is equal to the reference, the counter and the FF used to memorize the fault condition are reset, so incoherence_det_gdls is 0. When the probed state differs from the reference state, the counter increments. If the counter counts to its maximum, the “RS” is set and incoherence_det_gdls signal passes 1.

The presence of the counter offers a debounce filter and accommodates for propagation delays. The counter width is set to 3 bits.

The fault signal is a dynamic event in case of toggling IO. It has to be captured by an external logic to be memorized and reported to the MCU. The incoherence_det_gdls lines are connected to the interrupt controller that latches the request event.

The checkers' outputs are connected to the interrupt controller. This block instantiates twice.

2.2.10.6 *Operating parameters*

Table 41 summarizes the IP functional operating parameters.

Table 41. **Protection operating parameters**

Parameter	Value
Operating frequency	16 MHz
IO check timer	4 clock cycles
GDLS check timer	3 clock cycles

2.2.10.7 *Configuration register overview*

Table 42 shows the list of PROTECTION configuration registers overview.

Table 42. **Protection configuration registers overview**

Registers overview					
Name	**Description**	**Type**	**Page**	**Offset**	**Reset val.**
PROT_CTRL_0	Protection control config.	R/W	0	0x58	0x00
PROT_CTRL_1	Protection control config.	R/W	0	0x59	0x00

2.2.11 NVM control logic

2.2.11.1 *Overview*

This block manages the control access to the embedded NVM memory.

2.2.11.2 *Feature list*

- NVM deferred transaction
- NVM single sector operation
- NVM control enable bit to save consumption
- NVM data order from LSB to MSB
- NVM auto-load trimming data
- NVM soft-program, erase and word program allowed in test mode
- NVM clock operation configurable

2.2.11.3 *Functional description*

This IP manages the i/f with the NVM macro which allows read, erase and program memory command operations.

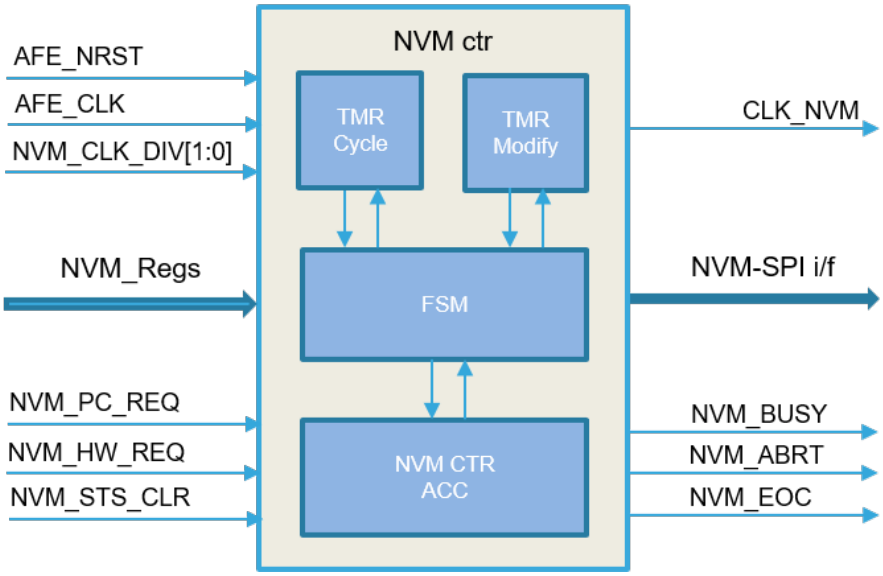
Only a single sector operation (64-bit access) is allowed; multi-sector operation requires the iteration of the single sector command sequences.

In functional mode the IP supports only the memory read operation while in test mode all commands are permitted.

All memory commands are executed in a deferred way by an internal FSM controller leaving the SPI bus i/f free to support other transactions. The memory read data are posted in the NVM_DATA_0-7 registers readable from the SPI i/f.

Here below is shown a high level overview of the internal block diagram.

Figure 58. **NVM block diagram overview**



Note: For readability reasons, the block diagram shows only the most relevant signals.

2.2.11.3.1 NVM details operation

The memory command requests are set through the SPI register signals NVM_HW_REQ and NVM_PC_REQ belonging to register NVM_CTRL_0, then the controller sets the NVM_BUSY bit notifying an access in progress or the NVM_ABRT bit in case of an ongoing transaction triggering the FSM flow execution. As soon as the NVM operation is completed the NVM_EOC status bit is set and the NVM_BUSY flag is reset. All status bits are reset by the NVM_STS_CLR command bit.

The memory command operations are controlled by an internal FSM logic which generates the STANDBY command at the beginning and at the end of any user command sequences to meet the NVM memory access requirements.

The FSM works according to an embedded counter timer count value (the timer is enabled/disabled directly from the FSM logic) an additional long timer counter is used to control the modify timing access used during SOFT-ERASE, ERASE and WORD-PROGAM commands.

2.2.11.3.2 NVM clock operation

The NVM clock frequency is configured through the NVM_CLK_DIV[1:0] bit fields of register NVM_CTRL_2. In order to minimize the dynamic power consumption, the CLK_NVM is enabled only during the memory command execution. The CLK_NVM operating frequency should be configured in a range between 4 MHz to 5 MHz; the clock line is generated by a programmable clock divider logic interconnected to the NVM controller “afe_clk” clock signal.

Table 43 summarizes the NVM legal clock frequency configurations.

Table 43. **NVM clock frequency configuration overview**

AFE_CLK	NVM_CLK_DIV[1:0]	CLK_NVM	Comment
20 MHz	2	5 MHz	ATE_CLK
16 MHz	2	4 MHz	AFE_CLK
10 MHz	1	5 MHz	ATE_CLK
8 MHz	1	4 MHz	AFE_CLK

2.2.11.3.3 NVM commands overview

The NVM controller allows configuring all user command operations on the NVM_OP_REG[3:0] signals with codes comprised between 0x0 up to 0x6; all the other commands are provided in test mode by the ATE tester equipment driving directly the NVM memory exposed pins during the EWS and FT.

Table 44. NVM memory command overview

Operation	OP_REG[3:0]	Description
Standby	0x0	No operation
Read Memory	0x1	Read memory data
Soft-Program	0x2	Soft program selected sector
Erase	0x3	Erase selected sector
Word Program	0x4	Program Word selected sector
Write Data on PL	0x5	Write Pre-Load data
Read PL data	0x6	Read Pre-Load data
FDMA direct	0x8	Read single ended mode – direct cell
FDMA complementary	0x9	Read single ended mode – comp. cell
DMA direct	0xA	Direct Memory Access – direct cell
DMA complementary	0xB	Direct Memory Access – comp cell
Write TOP_REG	0xC	Write Test Operation Register
Read TOP_REG	0xD	Read Test Operation Register
Read complementary PL	0xE	Read Pre-Load complementary data

2.2.11.4 Programming

2.2.11.4.1 SW access

The NVM memory access requires the following program sequences:

- Evaluate that there is no NVM pending command by checking the NVM_BUSY bit of the NVM_CTRL_0 register (the bit must be cleared); wait until NVM_BUSY bit is cleared before moving to the next step.
- Configure the memory operation command and the sector number respectively in the NVM_OP_REG and the NVM_SECT_SEL fields of the NVM_CTRL_1 register.
- Select the NVM memory access request by setting the NVM_PC_REQ wc-bit field of register NVM_CTRL_0.
- Waiting for the command completion notified by the NVM_EOC; if the on-going operation was a read command the return data may read back from the NVM_DATA_0 to 7 registers. In case of abort notified by NVM_ABRT bit field (this event should be avoided thanks to the SW interlock procedures described in the first step) the entire command sequence has to be reiterated.
- Reset the NVM status bit by setting the NVM_STS_CLR wc bit field of register NVM_CTRL_0.
- To deeply reduce the power consumption, the NVM should be disabled (power down) after reading the manufacturing parameters in functional mode.

2.2.11.4.2 HW access

The sector-0 of NVM which stores the trimming data parameters is uploaded automatically from HW either following the reset sequence or by setting the NVM_HW_REQ command bit. During the NVM access the trimming data are written by HW.

Here below follows the related program sequences:

- Evaluate that there is no NVM pending command by checking the NVM_BUSY bit of the NVM_CTRL_0 register (the bit must be cleared); wait until NVM_BUSY bit is cleared before moving to the next step.
- Configure the memory operation command and the sector number respectively in the NVM_OP_REG and the NVM_SECT_SEL fields of the NVM_CTRL_1 register selecting the sector-0 “000” and command memory read “0x1”.

- Select the NVM memory access request by setting the NVM_HW_REQ wc-bit field of register NVM_CTRL_0.
- Wait for the command completion notified by the NVM_EOC. In case of abort notified by NVM_ABRT bit field (this event should be avoided thanks to the SW interlock procedures described in the first step) the entire command sequence has to be reiterated.
- Reset the NVM status bit by setting the NVM_STS_CLR wc bit field of register NVM_CTRL_0.

2.2.11.5 *Operating parameters*

Table 45 summarizes the IP functional operating parameters.

Table 45. NVM operating parameters

Parameter	Value
NVM operating clock	4 MHz to 5 MHz

2.2.11.6 *Configuration register overview*

Table 46 shows the list of NVM configuration registers overview

Table 46. NVM configuration registers overview

Registers overview					
Name	**Description**	**Type**	**Page**	**Offset**	**Reset val.**
NVM_CTRL_0	NVM control configuration 0	R/W	5	0x02	0x00
NVM_CTRL_1	NVM control configuration 1	R/W	5	0x03	0x00
NVM_CTRL_2	NVM control configuration 2	R/W	5	0x04	0xC2
NVM_DATA_0	NVM data register	R/W	5	0x05	0xYY(1)
NVM_DATA_1	NVM data register	R/W	5	0x06	0xYY(1)
NVM_DATA_2	NVM data register	R/W	5	0x07	0xYY(1)
NVM_DATA_3	NVM data register	R/W	5	0x08	0xYY(1)
NVM_DATA_4	NVM data register	R/W	5	0x09	0xYY(1)
NVM_DATA_5	NVM data register	R/W	5	0x0A	0xYY(1)
NVM_DATA_6	NVM data register	R/W	5	0x0B	0xYY(1)
NVM_DATA_7	NVM data register	R/W	5	0x0C	0xYY(1)

1. *Sector-0 byte values in functional mode or 0x00 in test mode.*

2.2.12 RCC (Reset & Clock Controller) circuit

2.2.12.1 *Overview*

This block controls the generation of the Reset and Clock signals after the IC power-up. Whenever the power is applied to the system, this block grants the congruent sequence of internal reset controls and the delivery of the system clock.

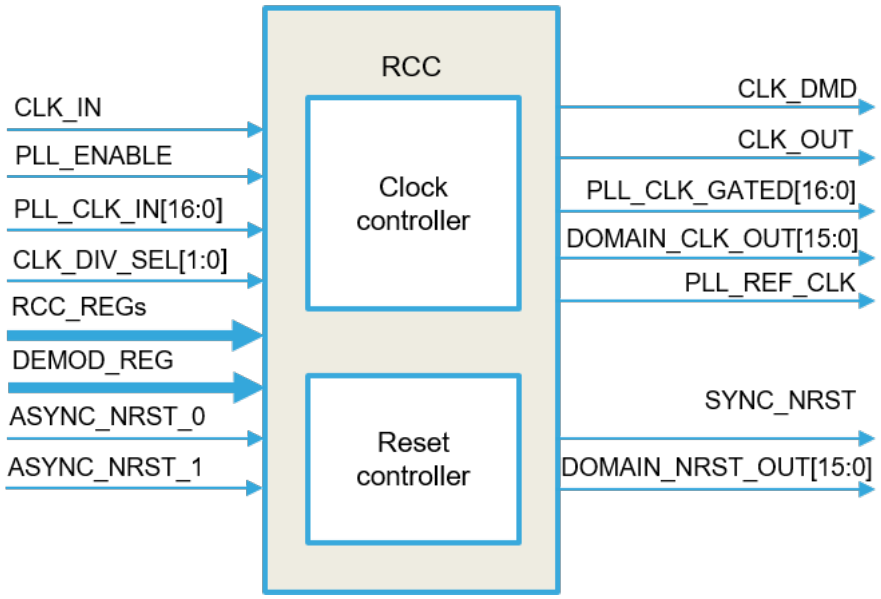
2.2.12.2 *Feature list*

- Two independent clock and reset line domains
- Global reset assertion on HW/SW reset commands
- Selective IP SW reset generation
- PLL clock control circuit
- Control clock gating for every digital IP
- DMD configurable clock line.

2.2.12.3 *Functional description*

Figure 59 shows the RCC block diagram.

Figure 59. RCC block diagram



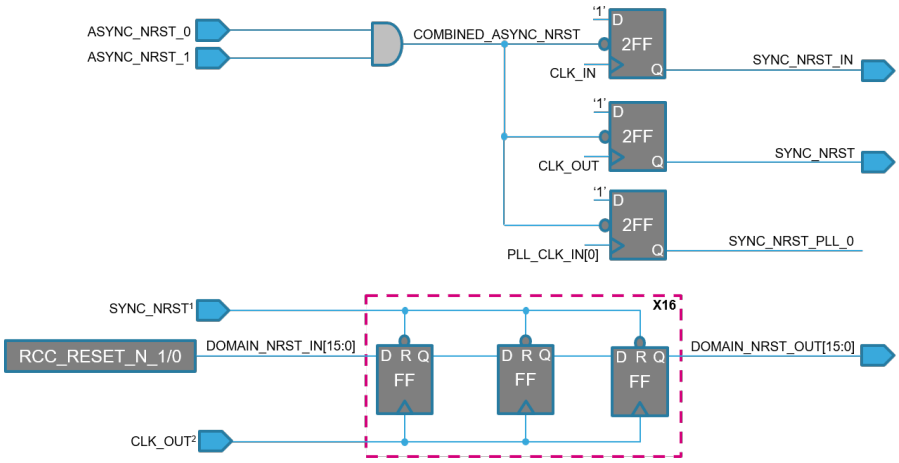
2.2.12.4 *Reset functionality*

The block uses the ASYNC_NRST_0 signal coming from the pad and the ASYNC_NRST_1 provided by a GPIO driven from the FW application. These signals concur to generate the internal main reset signal in the two clock domains.

A selective reset may be generated by SW in order to reset a specific IP without impacting the functionalities of other circuits.

Figure 60 shows an outline view of the AFE reset circuit.

Figure 60. Reset circuit overview



Note: *1. SYNC_NRST_PLL_0 used for DOMAIN_NRST_OUT[13-15*

2. PLL_CLK_GATED[0] used for DOMAIN_NRST_OUT[13-15]

Note that the reset scheme is intended to show the circuit functionality and may be not aligned with the RTL implementation.

2.2.12.5 *Clock functionality*

The AFE companion DIE has two asynchronous clock lines.

- The main clock is originated by the MCU companion device after the initialization. The clock is interconnected to a programmable divisor logic to select the AFE target operating frequency in a range of 8 – 16 MHz. The main clock is supplied to the PLL reference clock line in a range of 8 – 18 MHz.
- The high-speed clock is provided by a PLL phase locked circuit interconnected to a 17 phase DLL which provides 17 clocks up to 40 MHz

Note: After reset the CLK_OUT is equal to the CLK_IN.

Figure 61 shows the clock divisor circuit overview.

Figure 61. Clock divisor circuit overview

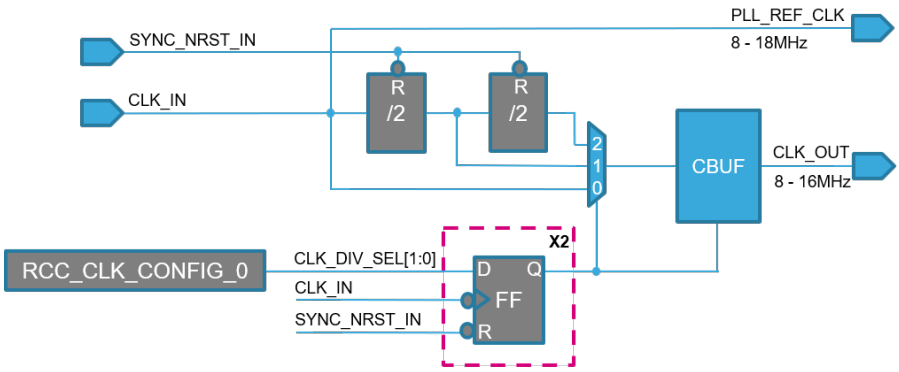
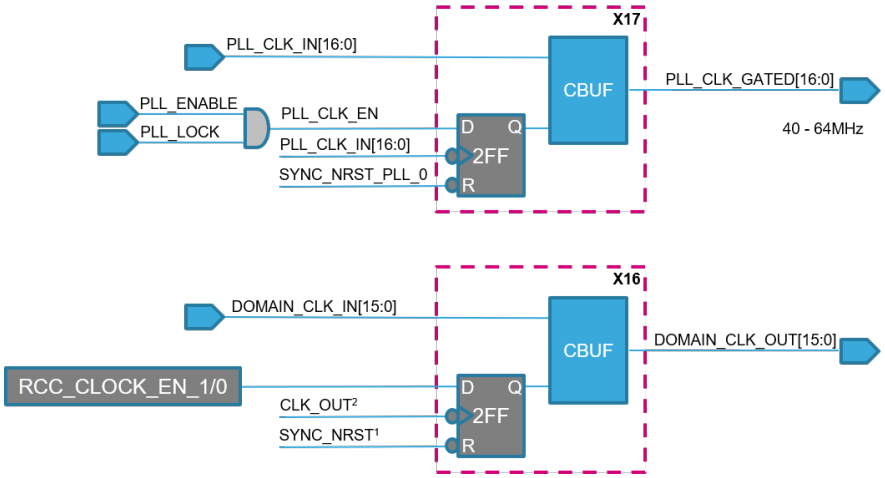


Figure 62 shows the clock enable circuit overview.

Figure 62. Clock enable circuit overview

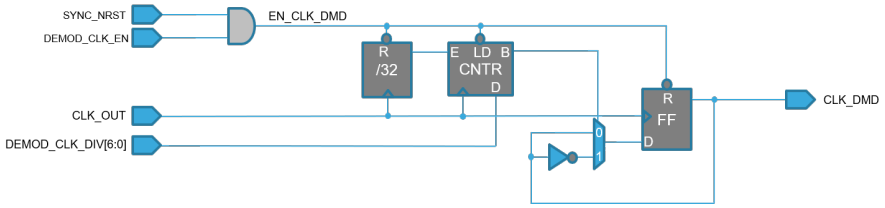


Note:
1. *SYNC_NRST_PLL_0 used for DOMAIN_CLK_OUT[13-15]*
2. *PLL_CLK_GATED[0] used for DOMAIN_CLK_OUT[13-15]*
3. *The clock enable controllability of PLL_CLK_GATED[15] is reserved*
4. *PLL_CLK_GATE[16:0] clock lines are directly interconnected to the HR-PWM IP*

2.2.12.5.1 DMD clock

Figure 63 shows the clock circuit used to generate the demodulation comparator auto-zero offset functionality.

Figure 63. **DMD clock circuit overview**



The clk_dmd is used in both current and voltage demodulator circuits to provide a fixed frequency closed to about 100 kHz (+/-30%). The controllability of the above circuit is done through the DEMOD_CTRL register.

The ENABLE bit controls the clock toggle functionality while the CLK_DIV represents the clock count.

The clk_out is divided by a fixed prescale value of 4-bits and is then counting the value configured in the CLK_DIV register field from 0x01 up to 0x7F.

The CLK_DMD equation is shown in the next equation.

$$Fdmd = \frac{Fafe}{(((Clk_{div} * Pre_{scale}) + 1)*2)}$$

Where:

- Pre_{scale} = 4
- CLK_DIV is comprised between 0x01 up to 0x7F

2.2.12.5.2 PLL clock configuration

The PLL configuration parameters require first to disable the PLL by clearing the PLL_EN bit field of register RCC_CLK_CONFIG_1 then configures both PLL_PRE_DIV and PLL_MULT bit fields belonging to registers RCC_CLK_CONFIG_1 and RCC_CLK_CONFIG_2, finally the PLL should be re-enabled by setting the PLL_EN bit.

Any further register modification parameters require the iteration of the above procedures.

The RCC provides clock and reset to 16 controllable domains (peripherals). The clock source of each domain is assigned to clk_out or pll_clk_gated[0]. Similarly, the async reset source for each domain is assigned to sync_nrst or sync_nrst_pll_0. The configuration is static (list of define).

Table 47 shows the clock and reset signal vector assignment.

Table 47. **RCC SW RST and Clock gating bit control**

SW RST and Clock gating			
bit	Peripheral	source	Comment
0	RFU		Reserved
1	Page 1 registers	clk_out / sync_nrst	BB-DCDC registers
2	Page 2 registers	clk_out / sync_nrst	HR-PWM registers
3	RFU		Reserved
4	RFU		Reserved
5	Page 5 registers(1)	clk_out / sync_nrst	Trimming registers, NVM and DFT registers
6	RFU		Reserved
7	Reserved	clk_out / sync_nrst	Reserved
8	Common IPs	clk_out / sync_nrst	LEDs, ADC_MUXs, Prot, and ITC IPs
9	Phase demod.	clk_out / sync_nrst	Phase demodulator
10	RFU		Reserved
11	RFU		Reserved
12	RFU		Reserved

SW RST and Clock gating			
13	BB-DCDC	pll_clk_gated[0] / sync_nrst_pll_0	BB-DCDC IP
14	QF controller	pll_clk_gated[0] / sync_nrst_pll_0	Qfactor IP
15	HR-PWM(2)	sync_nrst_pll_0	Controls only the reset. The clock of HR-PWM is directly provided by the pll_clk_gated[16:0] lines disabling by PLL_ENABLE signal.

1. *The trimming and DFT registers are reset with the main reset signal line.*
2. *The clock enable controllability PLL_CLK_GATED[15] line is reserved.*

2.2.12.6 *Operating parameters*

Table 48 summarizes the IP functional operating parameters:

Table 48. RCC operating parameters

Parameter	Value	Comment
AFE input clock (CLK_IN)	16 MHz	Basic clock low speed
PLL reference output clock	16 MHz	Synch. with CLK_IN
CLK_OUT output clocks	16 MHz	Synch. with CLK_IN
PLL_CLK input clock	40 MHz	Clock high speed
PLL_CLK_GATED output clocks	40 MHZ	Synch. with PLL_CLK
CLK_DMD output clock	100 kHz	Synch. with CLK_IN

2.2.12.7 *Configuration register overview*

Table 49 shows the list of RCC configuration registers overview:

Table 49. RCC configuration registers overview

Registers overview					
Name	Description	Type	Page	Offset	Reset val.
RCC_CLOCK_EN_0	Clock enabling of digital cells 0	R/W	0	0x38	0xFF
RCC_CLOCK_EN_1	Clock enabling of digital cells 1	R/W	0	0x39	0xFF
RCC_RESET_N_0	Reset of digital cells 0	R/W	0	0x3A	0xFF
RCC_RESET_N_1	Reset of digital cells 1	R/W	0	0x3B	0x1F
RCC_CLK_CONFIG_0	Clock configuration 0	R/W	0	0x3C	0x00
RCC_CLK_CONFIG_1	Clock configuration 1	R/W	0	0x3D	0x00
RCC_CLK_CONFIG_2	Clock configuration 2	R/W	0	0x3E	0x00
RCC_STATUS	Reset and clock status	R	0	0x3F	0x0Y(1)

1. *Refer to register definition.*

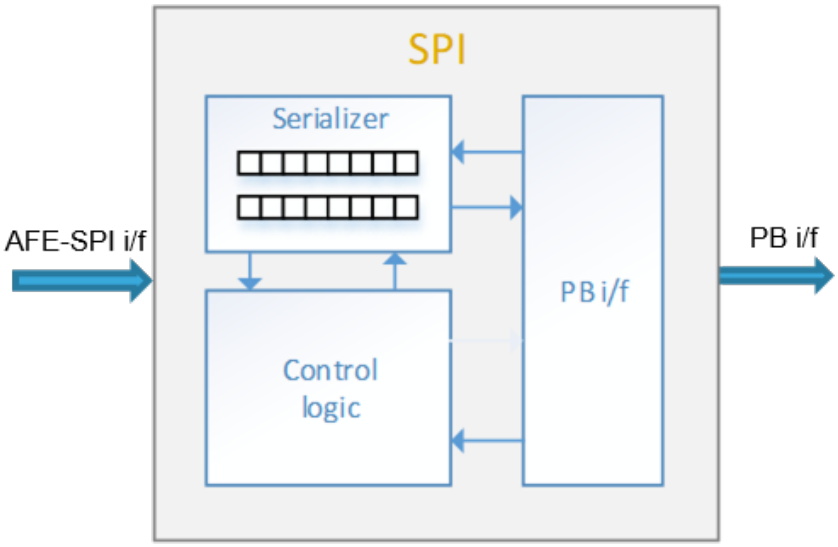
2.2.13 SPI slave interface

2.2.13.1 *Overview*

This block is the AFE communication interface with the MCU companion DIE; the SPI slave interface decodes SPI serial signals and generates a corresponding parallel peripheral bus transaction.

Figure 64 shows the SPI block diagram overview.

Figure 64. **SPI block diagram overview**



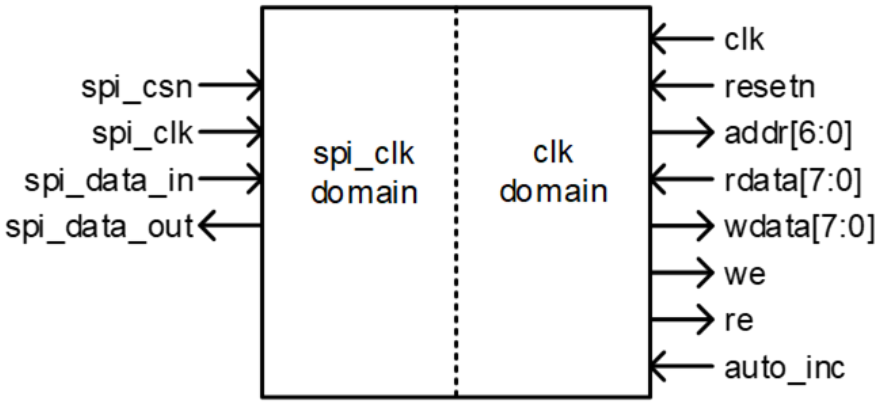
2.2.13.2 *Feature list*

- Slave 16bits serial interface to master parallel peripheral bus
- Complete asynchronous link between the SPI and PB bus
- Clock ratio of spi_clk vs afe_clk is 2
- Supports multi-byte read and write with fixed, auto-increment or explicit address
- Supports read-modify-write atomic sequence
- Serial data order from MSB to LSB
- Disable synchronizer stages in ATE test mode

2.2.13.3 *Functional description*

The cell has 2 separate clock domains:

Figure 65. **SPI clock domain overview**



- The spi_clk domain contains the serial engine. It is reset with resetn and partially also with spi_csn.
- The clk domain generates parallel peripheral bus access corresponding to the SPI transaction.

The spi_clk domain serializer/deserializer is reset with spi_csn at 1. Hence, the machine restarts from scratch at each new SPI access.

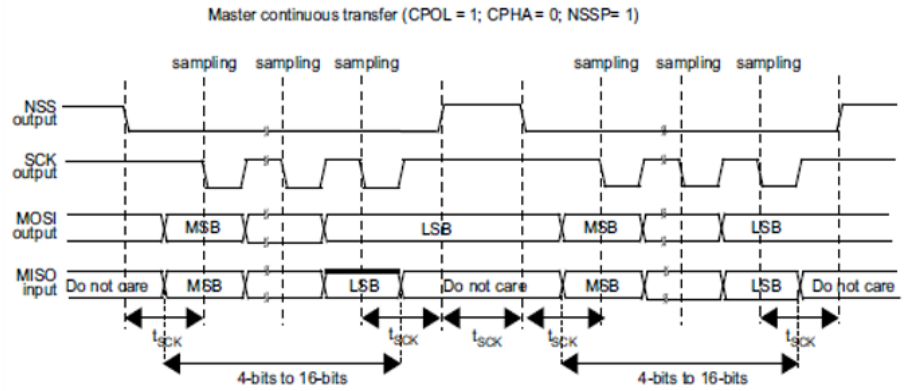
The serial machine is clocked on neg edge of spi_clk. During the SPI transaction, the machine decodes the access direction, the access address and the eventual write data.

The addr vector is asynchronously provided to the peripheral bus based on the spi_clk. The peripheral bus read operation being completely combinational, there is no synchronization between clk and spi_clk domain.

The clk domain receives access request from the spi_clk domain through read and write toggling flags. These flags are synchronized on clk domain with dual edge. Since a flag is seen toggled, the clk domain generates the corresponding re or we pulse. The addr and wdata vectors are guaranteed to be static before and during the re or we pulse generated on clk domain.

Here below is shown the SPI basic timing diagram.

Figure 66. MCU-SPI interface timing diagram overview



The next section details all the supported transaction operations.

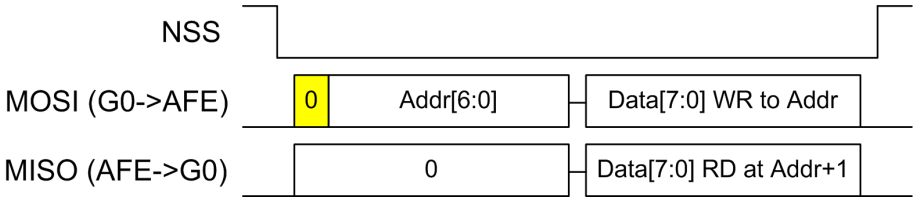
2.2.13.3.1 Single byte write access:

The first byte transmitted by the master after NSS goes low corresponds to the access address. The MSB of this byte is 0, meaning that it is a write access. During the transmission of the first byte, the slave transmits 8bits data at 0.

The second byte sent by the master is the 8bits data to be written at the requested address. While the data byte to be written is sent by the master, the slave transmits the value at address+1. This offers the possibility to combine a write and read on adjacent address.

The write process stops when NSS comes high.

Figure 67. Single byte write access



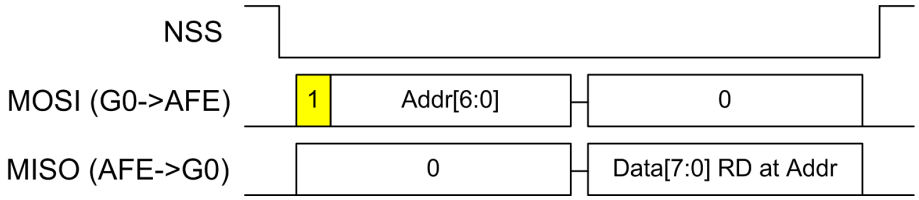
2.2.13.3.2 Single byte read access:

The first byte transmitted by the master after NSS goes low corresponds to the access address. The MSB of this byte is 1 meaning that it is a read access. During the transmission of the first byte, the slave transmits 8bits data at 0.

The second byte sent by the master is not used and must be 0. During the master transmission of second byte, the slave transmits the value at address.

The read process stops when NSS comes high.

Figure 68. **Single byte read access**



2.2.13.3.3 Multiple byte write access with address auto-increment:

The auto-increment is activated with a specific signal of the I/F.

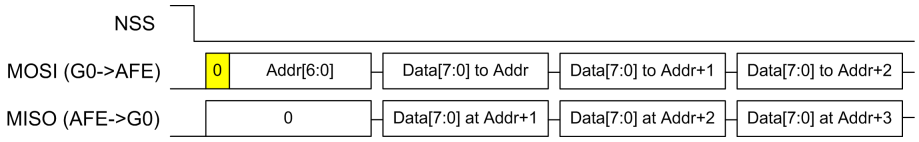
The first byte transmitted by the master after NSS goes low corresponds to the initial access address. The MSB of this byte is 0, meaning that it is a write access. During the transmission of the first byte, the slave transmits 8bits data at 0.

The second byte sent by the master is the 8bits data to be written at the requested address. While the data byte to be written is sent by the master, the slave transmits the value at address+1. This offers the possibility to combine a write and read on adjacent address.

For each following byte, the address of the access is automatically incremented up to 0x7F. When 0x7F is reached, the address is no more incremented.

The write process stops when NSS comes high.

Figure 69. **Multiple byte write with address auto-increment**



2.2.13.3.4 Multiple byte write access at fixed address:

The auto-increment is deactivated with a specific signal of the I/F.

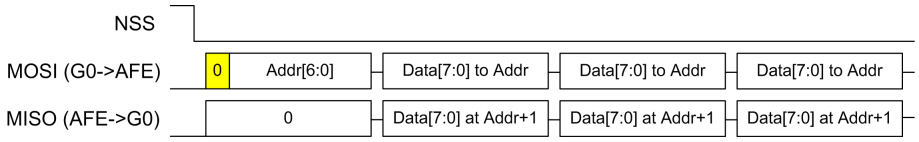
The first byte transmitted by the master after NSS goes low corresponds to the initial access address. The MSB of this byte is 0, meaning that it is a write access. During the transmission of the first byte, the slave transmits 8bits data at 0.

The second byte sent by the master is the 8bits data to be written at the requested address. While the data byte to be written is sent by the master, the slave transmits the value at address+1. This offers the possibility to combine a write and read on adjacent address.

For each following byte, the address of the access is the same. This mode of operation fits perfectly with a FIFO fill operation.

The write process stops when NSS comes high.

Figure 70. **Multiple byte write with fixed address**



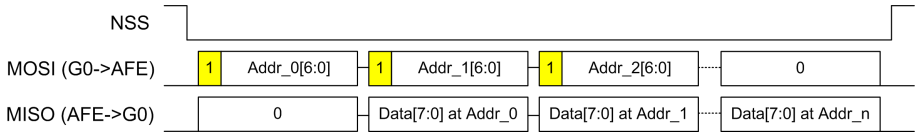
2.2.13.3.5 Multiple byte read access with explicit addresses:

The first byte transmitted by the master after NSS goes low corresponds to the first access address. The MSB of this byte is 1, meaning that it is a read access. During the transmission of the first byte, the slave transmits 8bits data at 0.

Each following byte transmitted by the master corresponds to a new address on which the read has to be done. These bytes must have their MSB bit at 1. During the transmission of the new address, the slave responds to the data read at the previous address request.

The read process stops when NSS comes high.

Figure 71. **Multiple byte read access with explicit address**



2.2.13.3.6 Multiple byte read access with address auto-increment:

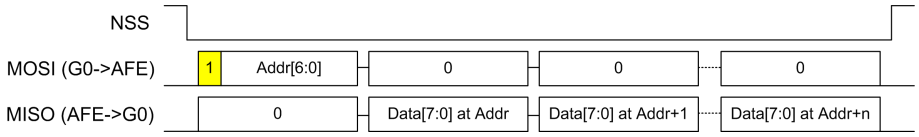
The auto-increment is activated with a specific signal of the I/F.

The first byte transmitted by the master after NSS goes low corresponds to the first access address. The MSB of this byte is 1, meaning that it is a read access. During the transmission of the first byte, the slave transmits 8bits data at 0.

Each following byte must be 0 to engage the auto-increment of the address. The address of the access is automatically incremented up to 0x7F. When 0x7F is reached, the address is no more incremented.

The read process stops when NSS comes high.

Figure 72. **Multiple byte read access with address auto-increment**



2.2.13.3.7 Multiple byte read access at fixed address:

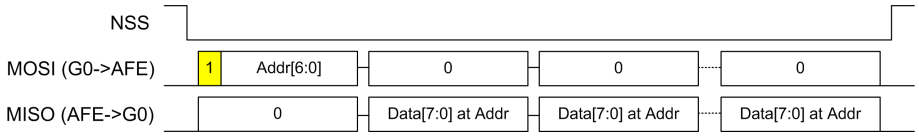
The auto-increment is deactivated with a specific signal of the I/F.

The first byte transmitted by the master after NSS goes low corresponds to the first access address. The MSB of this byte is 1, meaning that it is a read access. During the transmission of the first byte, the slave transmits 8bits data at 0.

Each following byte must be 0 to engage the fixed address access. For each byte, the slave transmits the data read at the same address. This mode is useful to read a FIFO content at a fixed address.

The read process stops when NSS comes high.

Figure 73. **Multiple byte read access at fixed address**



2.2.13.3.8 Read-Modify-Write access:

The Read-Modify-Write (RMW) access allows modifying one bit or two contiguous bits at a given position in a register.

The first byte transmitted by the master after NSS goes low corresponds to the access address. The MSB of this byte is 1, meaning that it is a read access. During the transmission of the first byte, the slave transmits 8bits data at 0.

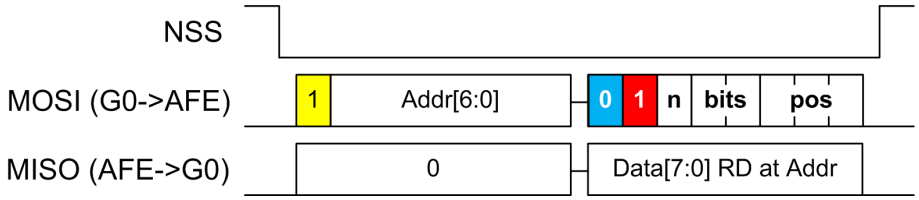
The second byte transmitted must have bit 7 = 0 and bit 6 = 1. This informs the SPI slave that access is an RMW.

Bit 5, called n, informs of the number of bits to modify. If bit 5 = 0, only 1 bit is modified. If bit 5 = 1, 2 bits are modified.

Bit 4 & 3, called bits[1:0], gives the bit value to set in the register. If the RMW is only 1bit, the bits[0] bit is used.

Bit 2, 1 & 0, called pos[2:0], gives the position in the register byte to apply the bit modification.

Figure 74. **Read Modify Write access**



Example 1:

- Register is 8'b00110100
- RMW operation is done on 1 bit, pos[2:0] = 1 and bits[0] = 1.
- Then register becomes 8'b00110**1**0

Example 2:

- Register is 8'b00110100
- RMW operation is done on 2 bit, pos[2:0] = 2 and bits[1:0] = 2'b10.
- Then register becomes 8'b0011**10**00

2.2.13.4 *Operating parameters*

Table 50 summarizes the IP functional operating parameters.

Table 50. **SPI operating parameters**

Parameter	Value
Operating frequency	16 MHz
SPI frequency	16 MHz

Note: The minimum clock ratio of SPI_CLK vs AFE_CLK is 2.

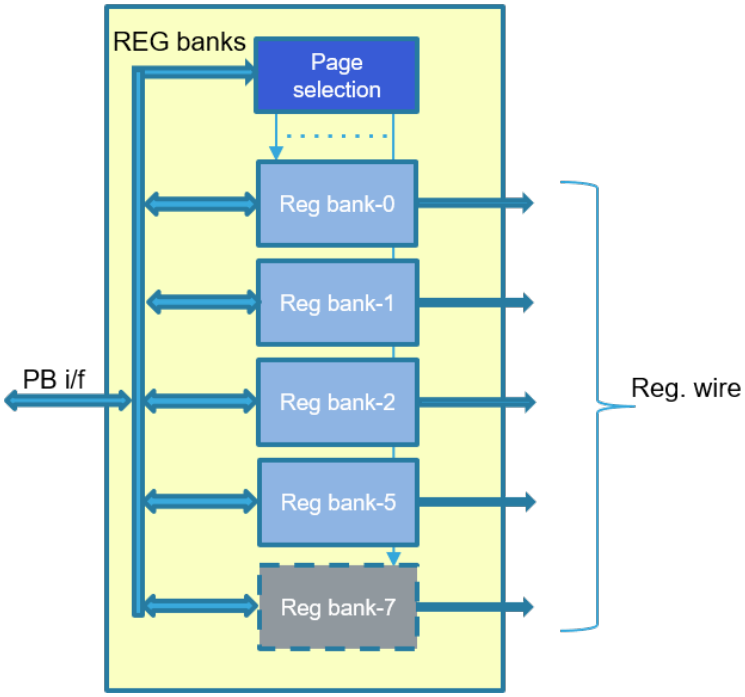
2.2.14 Register sub-system

2.2.14.1 *Overview*

This block is a multifunction IP which includes the page register bank selection and the data register array selected by the page register contents.

Figure 75 shows the page register bank selection block diagram.

Figure 75. **Register subsystem diagram overview**



2.2.14.2 *Feature list*

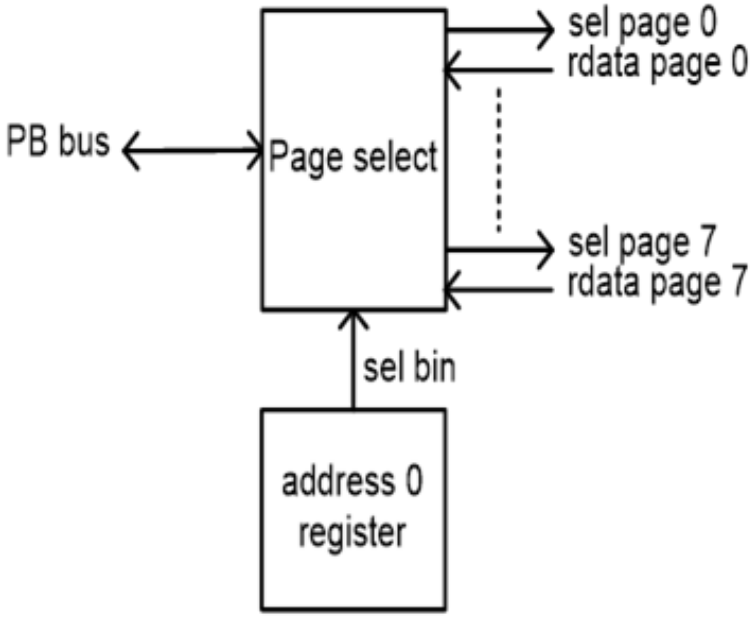
- Provides page selection register
- Performs read data mux
- Data register array (8bits registers with R/W, R only or self-clearing bit-fields)
- Register banks organization:
  - Bank-0: Miscellaneous registers of common peripherals
  - Bank-1: BB-DCDC configuration registers
  - Bank-2: HR-PWM configuration registers
  - Bank-5: Manufacturing trimming parameters, NVM and DFT registers
  - Bank-7: Reserved for FPGA emulator environment

2.2.14.3 *Functional description*

2.2.14.3.1 Page register:

Here below is shown the internal block diagram of the page register selection.

Figure 76. **Page Register sub-block diagram overview**



The register sub-system is arranged around a page select cell and a single register located at address 0. The address 0 register is always accessed since the PB bus address is 0. This register has the following layout:

Offset: 0x00

Default Value: 0x00

PAGE_REGISTER Control Interface

7	6	5	4	3	2	1	0
auto_inc	RFU				page_sel		
R/W	R				R/W		

R/W	R	R/W

- auto_inc bit: this bit is provided to the SPI slave interface to control the auto increment of access address.
- page_sel[2:0]: this vector is provided to the page select cell in order to select a page of registers.

2.2.14.4 *Operating parameters*

Table 51 summarizes the IP functional operating parameters:

Table 51. **Register subsystem operating parameters**

Parameter	Value
Bus data width	8 bits
Bus address width	7 bits

2.2.14.5 *Configuration register overview*

Table 52 shows the list of Register subsystem configuration registers overview:

Table 52. **Reg. subsystem configuration registers overview**

Registers overview					
Name	**Description**	**Type**	**Page**	**Offset**	**Reset val.**
PAGE_SELECTION	Register at address 0 as described in cell description section	R/W	0	0x00	0x00

Note: For data register description please contact our sales representative.

3 Electrical data

3.1 Absolute Maximum Ratings

Stresses above the absolute maximum ratings listed in Table 53 may cause permanent damage to the device. Exposure to maximum rating conditions for extended periods may affect device reliability.

Table 53. Absolute maximum ratings

Symbol	Parameter	Test condition	Min.	Max.	Unit
V_{IN}	Power supply voltage (VIN)	-	-0.3	27	V
V_{DD_DRIVER}	Gate driver supply voltage (VDD_DR)		-0.3	15	V
V_{LSn_GD}	Low-side gate driver voltage (LS1_GD, LS2_GD, LS3_GD)		-0.3	V_{DR} + 0.3	V
V_{HBn_SW}	HB Switching node voltage (HB1_SW, HB2_SW, HB3_SW)		-2	50	V
V_{BRG_VSNS}	BRG_VSNS, DCDC_SNS		-0.3	50	V
V_{HBn_BT}	HB Bootstrap voltage (HB1_BT, HB2_BT, HB3_BT)		Max(V_{HBn_SW} - 0.3 or -0.3)	Min((V_{HBn_SW} + VDR + 0.3) or 60)	V
V_{HSn_GD}	HB High-side gate driver voltage (HS1_GD, HS2_GD, HS3_GD)		V_{HBn_SW} -0.3	V_{HBn_BT}	V
V_{DD}	Input voltage for Analog die (VDD)		-0.3	5.5	V
V_{BUCK_SW}	Buck switching node voltage (BUCK_SW)		-1	27	V
V_{BUCK_BOOT}	Buck bootstrap voltage (BUCK_BOOT)		-0.3	32.0	V
V_{IN_SNS}	Vin sense voltage (VIN_ISNS_P, VIN_ISNS_M, VIN_SNS)		-0.3	Vin	V
V_{LDO_3V3}	Input voltage for MCU die (LDO_3V3, VREF_A, MCU_3V3)		-0.3	4	V
V_{IO_TT}	MCU logic input voltage (I/O TT structure pins – see Internal signal connection Table 38)		-0.3	4	V
V_{IO_FT}	MCU logic input voltage (I/O FT structure pins – see Internal signal connection Table 38)		-0.3	V_{LDO_3V3} + 4	V
V_{IO_DRV}	General purpose driver I/Os (GP_DRV<0:3>)		-0.3	4	V
V_{NRST}	NRST I/O		-0.3	4	V
V_{LDO_1V8}	Supply voltage for digital circuitry (LDO_1V8)		-0.3	2	V

Symbol	Parameter	Test condition	Min.	Max.	Unit
V_{IO_AN}	Analog pin voltage (BUCK_FB, BRG_ISNSP, BRG_ISNSM, BRG_SNSB, BRG_IDEM, BRG_FT_R, BRG_FT_S, RNG_FT_R, RNG_FT_S, RNG_SNS, USB_DP, USB_DM, QF_DRV,LED1,LED2)		-0.3	V_{DD}	V
T_{STG}	Storage temperature		-55	150	°C

3.2 Recommended Operating Conditions

Table 54. Recommended operating conditions

Symbol	Parameter	Test condition	Min.	Typ.	Max.	Unit
V_{IN}	Power supply voltage	-	4.1		24	V
	Power supply voltage slope	V_{IN}=22V			100	V/µs
V_{DD_DRIVER}	Voltage doubler output and gate driver supply voltage		5	7.5	10	V
V_{BUCK_SW}	Buck switching node voltage (BUCK_SW)				24	V
V_{BUCK_BOOT}	Buck bootstrap voltage (BUCK_BOOT)				29	V
V_{HBn_BT}	HB bootstrap voltage (HB1_BT, HB2_BT, HB3_BT)				50	V
V_{HBn_SW}	HB switching node voltage (HB1_SW, HB2_SW, HB3_SW)				40	V
V_{DD}	Buck regulator output and input voltage for Analog die		3.4		5.2	V
L_{BUCK}	Buck regulator inductance			10	22	µH
C_{BUCK}	Buck regulator capacitance			10		µF
E_{SRBUCK}	Buck regulator capacitance ESR		0.1			□
V_{LDO_3V3}	Linear regulator output and input voltage for MCU die		3.0	3.3	3.6	V
C_{LDO_3V3}	Linear regulator output capacitance		4.7		22	µF
V_{LDO_1V8}	Linear regulator output and supply voltage for digital circuitry			1.8	1.95	V
C_{LDO_1V8}	Linear regulator output capacitance		-20%	4.7	+20%	µF
V_{HS}	HB floating supply voltage(2)			V_{DD_DRIVER} (1)	10	V
T_j	Operating junction temperature		-40		125	°C

1. $V_{HS} = V_{BT\_HS} - V_{HB\_SW}$

3.3 ESD Protections

Table 55. ESD protection ratings

Symbol	Parameter	Test condition	Value	Unit
HBM		Conforming to ANSI/ESDA/JEDEC	2	kV

Symbol	Parameter	Test condition	Value	Unit
	Human body model	JS-001-2014		
CDM	Charged device model	Conforming to ANSI/ESDA/JEDEC JS-001-2014	250	kV

3.4 Thermal data

Table 56. Thermal data

Symbol	Parameter	Value	Unit
$R_{th(JA)}$	Thermal resistance junction to ambient	45	°C/W

3.5 Electrical characteristics

Conditions: V_{IN}=12 V, V_{DD_DRIVER}=7.5 V, V_{DD}=5 V, V_{LDO_3V3}=3.3 V, V_{LDO_1V8}=1.8 V, T_J from -40°C to 125°C, C_{LSn_GD}=C_{HSn_GD}=1 nF, C_{BRIDGE_VREF}=10 nF, L_{BUCK}=10 µH, C_{LDO_1V8}=1 µF, C_{LDO_3V3}=4.7 µF, C_{BUCK}=10 µF (ceramic capacitor), C_{EXT}=6.8 nF, R_{EXT}=100 kohm (C_{EXT} with R_{EXT} is AC coupling network on RING_NODE_SNS), C_{QFDRV_VRING}=2.2 nF (coupling cap for Q-factor measurement), $L_{BUCKBOOST}$=3.6 µH, C_{D+}=C_{D-}=100 pF; unless otherwise noted.

Typical values are tested at T_J = 25°C, minimum and maximum values are guaranteed by thermal characterization in the temperature range of -40 to 125°C, unless otherwise specified.

Table 57. Electrical charachteristics

Symbol	Parameter	Test condition	Min.	Typ.	Max.	Unit
POWER SUPPLY						
I_{IN_STBY}	Power supply current consumption in quiescent state	V_{IN}=22 V; V_{DD_DRIVER}=7.5 V, V_{DD}=5 V externally supplied			0.5	mA
		Fault condition, NRST=0 V_{IN}=22 V; V_{DD_DRIVER}=7.5 V, V_{DD}=3.6 V externally supplied			0.5	mA
V_{INon}	V_{IN} UVLO turn-on threshold	V_{IN} rising from 0 V	2.5		3.5	V
I_{DD_DRIVER}	V_{DD_DRIVER} current consumption	V_{DD_DRIVER}=7.5 V externally supplied, no commutation.			0.2	mA
I_{VDD}	V_{DD} current consumption	V_{DD}=5 V f_{SW}=100 kHz (system frequency)		10	15	mA
		QF on V_{IN}=22 V; V_{DD_DRIVER}=7.5 V, V_{DD}=5 V			5	mA
		Standby (NRST=0) V_{DD}=3.6 V T_j = 25°C		0.5		mA
		Standby (NRST=0) V_{DD}=3.6 V T_j = 125°C			1	mA
V_{ON},V_{DD3V6}	V_{DD} UVLO turn-on threshold at startup (no trimming)	V_{DD} turn-on threshold	3.25	3.35	3.45	V

Symbol	Parameter	Test condition	Min.	Typ.	Max.	Unit
V_{OFF},V_{DD3V6}	V_{DD} UVLO turn-off threshold (with 3.6 V selection)	V_{DD} turn-off threshold	3.25	3.35	3.45	V
V_{OFF},V_{DD5V}	V_{DD} UVLO turn-off threshold (with 5 V selection)	V_{DD} turn-off threshold	4.55	4.65	4.75	V
T_{DELAY_VDD}	Comparator propagation delay	dV/dt=100 mV/ms			1	µs
T_{FILTER_VDD}	Comparator output filter		-25%	1	+25%	µs
NRST PIN(1)						
$V_{IL(NRST)}$	NRST input low level voltage				0.3 x V_{LDO_3V3}	V
$V_{IH(NRST)}$	NRST input high level voltage		0.7 x V_{LDO_3V3}			V
$V_{hys(NRST)}$	NRST Schmitt trigger voltage hysteresis			0.6 x V_{LDO_3V3}		mV
R_{PU}	Weak pull-up equivalent resistor(2)	STM32G071 parameter, V_{SS}=V_{LDO_3V3}		75		kΩ
BUCK SWITCHING REGULATOR						
V_{IN}	V_{IN} operative range for VDD=5 V		5.5		24	V
	V_{IN} operative range for VDD=3.6 V		4.1		24	V
V_{BUCK_FB}	Average output voltage	V_{DD}=5 V selection, V_{IN} ≥ 6V	4.75	5.0	5.25	V
		V_{DD}=3.6 V selection, V_{IN} ≥ 4.5V	3.42	3.6	3.78	V
I_{LOAD}	Output load current				450	mA
$D_{VBUCK\_FB\_DO}$	V_{DD} Dropout Voltage	I_{LOAD}=[0÷450] mA DC; V_{DD}=3.6 V/5 V			500	mV
$I_{VALLEYLIMIT\_BUCK}$	Valley Current Limit	L=10 µH	450	600	750	mA
$I_{PEAKLIMIT\_BUCK}$	Peak Current Limit	In Low-Dropout operations only	700	900	1100	mA
f_{SW_BUCK}	Switching frequency	CCM when not in LDO, L=10 µH		1		MHz
R_{DSON_HS}	Buck switching node pull-up resistance	I_{LOAD}=100 mA, T_J = 25°C	300	430	685	mΩ
R_{DSON_LS}	Buck switching node pull-down resistance	I_{LOAD}=100mA, T_J = 25°C	240	340	540	mΩ
LINEAR REGULATORS						
V_{LDO_3V3}	LDO_3V3 regulation voltage	I_{LDO_3V3}=[0÷150] mA; V_{DD}=[3.4÷5.25] V	3.1	3.3	3.5	V
I_{LDO_3V3}	LDO_3V3 DC current capability				150	mA
□$V_{LDO\_3V3\_DO}$	LDO_3V3 dropout voltage	I_{LDO_3V3}=150 mA DC; V_{DD}=3.1 V			350	mV
I_{LIMIT_LDO3V3}	LDO_3V3 output current limit	Trimmed parameter	270	380	520	mA
C_{LDO_3V3}	LDO_3V3 output capacitance allowed range	Ceramic capacitor	4.7		22	µF
V_{LDO_1V8}	LDO_1V8 regulation voltage	I_{LDO_1V8}=[0÷50] mA; V_{DD}=[3.4÷5.25]	1.7	1.8	1.9	V
IL_{DO_1V8}	LDO_1V8 DC current capability				50	mA

Symbol	Parameter	Test condition	Min.	Typ.	Max.	Unit
I_{LIMIT_LDO1V8}	LDO_1V8 output current limit	Trimmed parameter	80		160	mA
C_{LDO_1V8}	LDO_1V8 output capacitance allowed range	Ceramic capacitor	-20%	4.7	+20%	µF
$V_{ON,V3V3}$		V_{LDO_3V3} rising voltage	2.85	2.95	3.05	V
$V_{OFF,V3V3}$		V_{LDO_3V3} falling voltage	2.7	2.8	2.9	V
T_{DELAY_V3V3}	Comparator propagation delay	dV/dt=100 mV/ms		1		µs
T_{FILTER_V3V3}	Comparator output filter		-25%	0.85	+25%	µs
$V_{ON,V1V8}$		V_{LDO_1V8} rising voltage	1.67	1.7	1.73	V
$V_{OFF,V1V8}$		V_{LDO_1V8} falling voltage	1.64	1.67	1.7	V
T_{DELAY_V1V8}	Comparator propagation delay	dV/dt=100 mV/µs		1		µs
T_{FILTER_V1V8}	Comparator output filter		-25%	0.85	+25%	µs
VOLTAGE DOUBLER CONTROL						
$f_{SW\_VOLTDOUBLE\_RNG}$	Switching frequency range (clock source from AFE/ PLL)		0.01		1	MHz
$R_{UP\_VOLTDOUBLE}$	Internal PMOS ON-resistance				10.5	Ω
$R_{DOWN\_VOLTDOUBLE}$	Internal NMOS ON-resistance				3.5	Ω
BUCK BOOST DC DC						
V_{IN}	V_{IN} operating range		4.1		24	V
f_{SW_BB}	Operating frequency				1	MHz
V_{BRIDGE}	Output voltage	Selectable via FW		1		V
				14		V
				18		V
				24		V
				30		V
				40		V
GATE DRIVERS						
$GD1_{RUP}$	Gate driver 1 high-side on resistance	T_J = 25°C			5	Ω
$GD1_{RDWN}$	Gate driver 2 low-side on resistance	T_J = 25°C			2.5	Ω
$GD2_{RUP}$	Gate driver 2 high-side on resistance	T_J = 25°C			5	Ω
$GD2_{RDWN}$	Gate driver 2 low-side on resistance	T_J = 25°C			2.5	Ω
$GD3_{RUP}$	Gate driver 3 high-side on resistance	T_J = 25°C			5	Ω
$GD3_{RDWN}$	Gate driver 3 low-side on resistance				2.5	Ω
LED DRIVER (can be bypassed to 5 V-GPIO with pad operating voltage V_{DD})						
I_{LEDDRV}	Sink current values			2.5		mA
				5		mA
				7.5		mA
				10		mA
I_{LEDDIM}	Dimming current step			2.5		mA

Symbol	Parameter	Test condition	Min.	Typ.	Max.	Unit
I_{LED_FREQ}	Pattern frequency				10	kHz
V_{LED_IL}	GPIO operations(2): low input voltage level				0.8	V
V_{LED_IH}	GPIO operations(2): high input voltage level		2			V
V_{LED_OL}	GPIO operations(2): low output voltage level	I_{LOAD}=6 mA (sourced)			0.4	V
V_{LED_OH}	GPIO operations(2): high output voltage level	I_{LOAD}=-6 mA (sunk)	V_{LDO_3V3}-0.4			V
f_{LED_GPIO}	GPIO operations(2): operating frequency	C=50 pF		2		MHz
Q-FACTOR DRIVER						
$V_{RING\_NODE\_RNG}$	Oscillation amplitude	Steady-state with ZCD lock	0.020	1	2	V_{pp}
EI_{QF_SINK}	Sink current capability error on all steps	V_{DD}=5 V V_{QF_DRV}≤V_{DD}-0.5 V SHORT=0, I_{QF_SOURCE}=0			10	%
EI_{QF_SOURCE}	Source current capability error on all steps	V_{DD}=5 V V_{QF_DRV}≥0.5 V S_{HORT}=0, I_{QF_SINK}=0			10	%
EΔI	Sink/Source mismatch error on all steps	V_{DD}=5 V V_{QF_DRV}=V_{DD}/2 SHORT=0		7		%
I_{SEL}	Current programmable step count	(excluded off state)		16		
I_{QF_STEP}	Current regulation step			250		µA
RINGING NODE VOLTAGE SENSE & DEMODULATION						
I_{RING_NODE}	RING_NODE_SNS input current during power transfer	V-I conversion is made by means of 100 kΩ external resistor.	±0.01		±1	mA
	RING_NODE_SNS input current during Q-factor measurement	V-I conversion is made by means of 100 kΩ external resistor.	±0.1		±10	µA
f_{RING_NODE}	Current & Voltage frequency range		100		300	kHz
$R_{GAIN\_NODE\_SNS}$	Trans-resistance gain values of the internal I-V converter	I_{RING_NODE}=±[0.8÷1] mA		2.2		kW
		I_{RING_NODE}=±[0.6÷0.8] mA		3.0		kW
		I_{RING_NODE}=±[0.44÷0.6] mA		4.0		kW
		I_{RING_NODE}=±[0.32÷0.44] mA		5.4		kW
		I_{RING_NODE}=±[0.24÷0.32] mA		7.4		kW
		I_{RING_NODE}=±[0.18÷0.24] mA		9.9		kW
		I_{RING_NODE}=±[0.135÷0.18] mA		13.2		kW
		I_{RING_NODE}=±[0.060÷0.135] mA		17.7		kW
		I_{RING_NODE}=±[0.025÷0.060] mA		39.4		kW
		I_{RING_NODE}=±[0.001÷0.025] mA		94.8		kW
		I_{RING_NODE}=±[4÷1] µA(3)		224.8		kW
		I_{RING_NODE}=±[1.6÷4] µA(3)		566.2		kW
		I_{RING_NODE}=±[0.1÷1.6] µA(3)		1442		kW
$f_{LPF\_NODESNSDEMOD}$	Low pass filter frequency values		-10%	4	+10%	kHz

Symbol	Parameter	Test condition	Min.	Typ.	Max.	Unit
$f_{LPF\_NODESNSDEMOD}$	Low pass filter frequency values		-10%	8	+10%	kHz
			-10%	12	+10%	kHz
			-10%	16	+10%	kHz
$f_{HPF\_NODESNSDEMOD}$	High pass filter cut-off frequency values		-10%	300	+10%	Hz
			-10%	450	+10%	Hz
			-10%	600	+10%	Hz
			-10%	750	+10%	Hz
BRIDGE CURRENT SENSE & DEMODULATION						
I_{BRIDGE_ZCD}	Zero cross detection offset voltage	Between BRIDGE_ISNS_P and BRIDGE_ISNS_N	-10	0	+10	mV
$V_{BDG\_CURRSNS\_CM}$	Sensing resistor common mode voltage			0		V
$V_{BDG\_CURRSNS\_RNG}$	Sensing resistor differential voltage range	Between BRIDGE_ISNS_P and BRIDGE_ISNS_N	±0.02		±200	mV
$G_{BDG\_CURRSNS}$	Bridge Current Sense Gain Four selectable values: 4, 8, 16, 32.		4		32	V/V
$Z_{IN\_BDGCURRSNS}$	Differential input impedance		5			kΩ
$I_{IN\_BDGCURRSNS}$	Input Current				5	mA
$f_{LPF\_BDGCURRDEMOD}$	Low pass filter cut-off frequency values		-10%	4	+10%	kHz
			-10%	8	+10%	kHz
			-10%	12	+10%	kHz
			-10%	16	+10%	kHz
$f_{HPF\_NODESNSDEMOD}$	High pass filter cut-off frequency values		-10%	300	+10%	Hz
			-10%	450	+10%	Hz
			-10%	600	+10%	Hz
			-10%	750	+10%	Hz
VIN CURRENT SENSE						
$V_{IN\_CURR\_SNS\_CM}$	Common mode voltage on sense resistor		4		24	V
$V_{IN\_CURR\_SNS\_RNG}$	Differential input voltage range (across sense resistor)		±16		±200	mV
$I_{IN\_CURR\_SNS\_CM}$	Common mode input current (through sense nodes)				5	mA
$Z_{IN\_CURR\_SNS\_CM}$	Input differential impedance			50		Ω
EXTERNAL GATE DRIVERS (GPIOs with pad operating voltage V_{LDO_3V3})						
V_{GP_IL}	Input low level voltage				0.8	V
V_{GP_IH}	Input high level voltage		2			V
V_{GP_OL}	Output low level voltage	I_{LOAD}=6 mA (sourced)			0.4	V
V_{GP_OH}	Output high level voltage	I_{LOAD}=- 6 mA (sunk)	V_{LDO_3V3}- 0.4			V
f_{GP}	Output signal frequency	C=50 pF, V_{LDO_3V3}			25	MHz
C_{OUT}	Pin capacitance			5		pF
PLL						
f_{PLL}	Output frequency		40		68	MHz
HIGH RESOLUTION PWM GENERATOR						
PWM_RES	PWM resolution	PLL frequency=40 MHz		1.47		ns
USB INTERFACE						

Symbol	Parameter	Test condition	Min.	Typ.	Max.	Unit
$V_{GEN\_DP(DM)}$	DP & DM generator	0.6 V voltage mode, I_{LOAD}: 0 ÷ 1 mA	0.5	0.6	0.7	V
		3.3 V voltage mode, I_{LOAD}: 0 ÷ 1 mA	3.0	3.3	3.6	V
		2.8 V voltage mode, I_{LOAD}: 0 ÷ 1 mA	2.6	2.8	3.0	V
		Current Mode Boundaries(4)	25		175	µA
$I_{VGEN\_DP(DM)}$	$V_{GEN\_DP(DM)}$ voltage generator current capability		0.5		1	mA
I_{DM_SINK/IDP_SINK}	Sink current from USB_DM or USB_DP pad		See STM32G071	62	100	144 µA
$V_{ISINK\_DP(DM)}$	$I_{DP(DM)\_SINK}$ current generator operative voltage		0.1			V
$R_{LD\_DP(DM)}$	USB_DP & USB_DM pull-down resistance	Pull-down mode - 20 k□ load	16	20	24	kΩ
$T_{SETT\_VGEN\_DM(DP)}$	Configuration settling time			50		µs

1. *See STM32G071 datasheet for details about NRST MCU pin.*
2. *LED driver bypassed, direct pin driving (5 V GPIO).*
3. *To be used during Q-factor measurement only.*
4. *For source current less than I_{DP_SINK} min. (25 µA), the voltage on USB_DP pin is required to be no higher than V_{DAT_SINK} max. (150 mV). For USB_DP voltage lower than V_{LGC} max. (2 V) the current sunk from USB_DP pin shall to not higher than I_{DP_SINK} max. (175 µA). Same requirements apply to USB_DM pin.*

4 STWBC2-HP pinout and pin description

4.1 Pinout

This section shows the pinout used by the STWBC2-HP.

Figure 77. STWBC2-HP pin connection (top view)

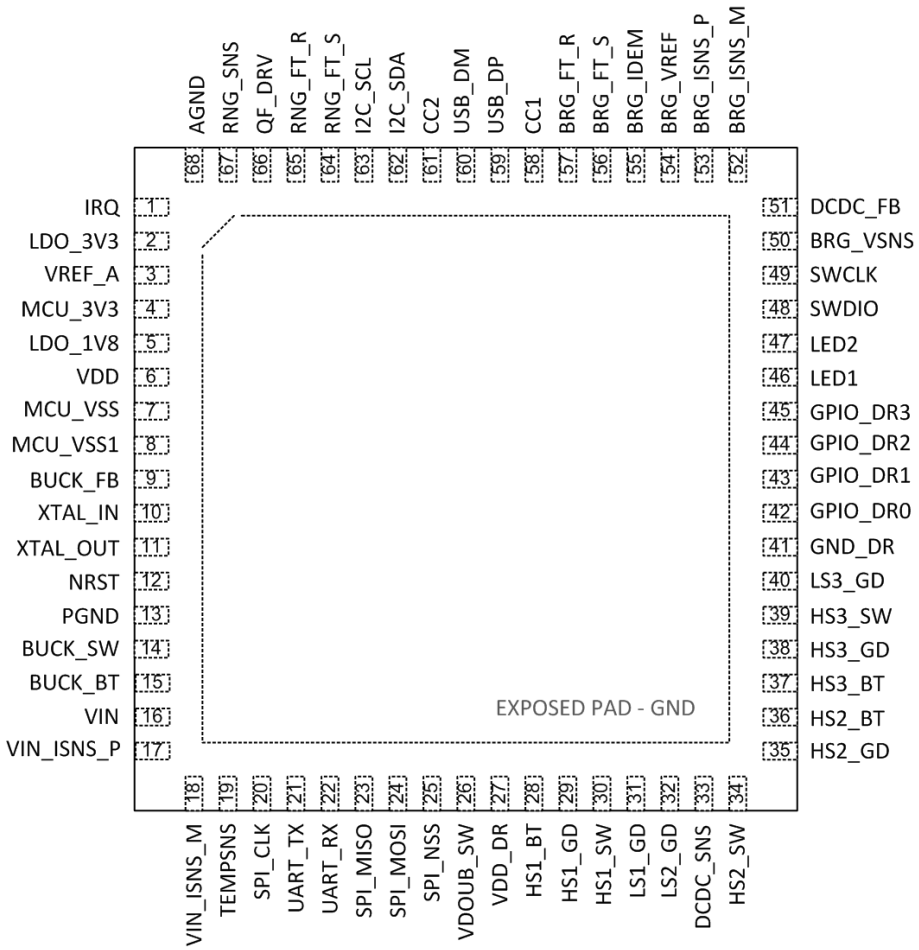


Table 58. Pinout description

Pin Number	Pin Name	Pin Type	Description
EP	GND	Power	AFE substrate GND
1	IRQ	Digital I/O	Interrupt pin or GPIO (PC15-OSC32-OUT)
2	LDO_3V3	Power	LDO 3.3 V output (VBAT)
3	VREF_A	Analog Out	Analog reference voltage (VREF+)
4	MCU_3V3	Power	MCU 3.3V supply input (VDD/VDDA)
5	LDO_1V8	Power	LDO 1.8 V output
6	VDD	Power	AFE 5 V supply input
7	MCU_VSS	Power	MCU GND (VSS/VSSA)

Pin Number	Pin Name	Pin Type	Description
8	MCU_VSS1	Power	MCU GND (VDD_VREFM)
9	BUCK_FB	Analog In	DCDC 5 V output (feedback)
10	XTAL_IN	Digital In	PF0- XTAL in pin (OSC driver in)
11	XTAL_OUT	Digital Out	PF1 -XTAL out pin
12	NRST	Digital I/O	PF2 - nRST
13	PGND	Power	AFE analog Ground
14	BUCK_SW	Analog Out	Embedded DCDC Buck switching node
15	BUCK_BT	Power	Embedded DCDC Buck bootrstrap
16	VIN	Power	Input voltage
17	VIN_ISNS_P	Analog In	VIN current sensor Plus
18	VIN_ISNS_M	Analog In	VIN current sensor Minus
19	TEMPSNS	Analog In	ADC input for temperature sense
20	SPI_CLK	Digital I/O	SPI CLK
21	UART_TX	Digital I/O	USART Tx
22	UART_RX	Digital I/O	USART Rx
23	SPI_MISO	Digital I/O	SPI MISO
24	SPI_MOSI	Digital I/O	SPI MOSI
25	SPI_NSS	Digital I/O	SPI NSS
26	VDOUB_SW	Analog Out	Charge pump
27	VDD_DR	Power	Gate driver supply voltage
28	HS1_BT	Power	High-side gate driver bootstrap cap
29	HS1_GD	Analog Out	High-side gate driver output
30	HS1_SW	Analog In	Half-bridge switching node and voltage sense for DCDC buck stage
31	LS1_GD	Analog Out	Low-side gate driver output
32	LS2_GD	Analog Out	Low-side gate driver output
33	DCDC_SNS	Analog In	Voltage sense for DCDC boost stage
34	HS2_SW	Analog In	Half-bridge switching node
35	HS2_GD	Analog Out	High-side gate driver output
36	HS2_BT	Power	High-side gate driver bootstrap cap
37	HS3_BT	Power	High-side gate driver bootstrap cap
38	HS3_GD	Analog Out	High-side gate driver output
39	HS3_SW	Analog In	Half-bridge switching node
40	LS3_GD	Analog Out	Low-side gate driver output
41	GND_DR	Power	Gate driver ground
42	GPIO_DR0	Analog Out	Logic level pin to drive ext. gate driver
43	GPIO_DR1	Analog Out	Logic level pin to drive ext. gate driver
44	GPIO_DR2	Analog Out	Logic level pin to drive ext. gate driver
45	GPIO_DR3	Analog Out	Logic level pin to drive ext. gate driver and DC voltage sensing function
46	LED1	Analog Out	LED 1
47	LED2	Analog Out	LED 2

Pin Number	Pin Name	Pin Type	Description
48	SWDIO	Digital I/O	Debugger IO
49	SWCLK	Digital I/O	Debugger CLK
50	BRG_VSNS	Analog In	Bridge voltage sense
51	DCDC_FB	Analog In	DCDC feedback pin
52	BRG_ISNS_M	Analog In	Bridge current sense Minus
53	BRG_ISNS_P	Analog In	Bridge current sense Plus
54	BRG_VREF	Analog In/Out	External filtering for internal voltage reference
55	BRG_IDEM	Analog In	Bridge current demodulation input
56	BRG_FT_S	Analog Out	Bridge LP filter send
57	BRG_FT_R	Analog In	Bridge LP filter return
58	CC1	Analog I/O	USB power delivery CC1 pin
59	USB_DP	Analog I/O	USB D+ pin
60	USB_DM	Analog I/O	USB D- pin
61	CC2	Analog I/O	USB power delivery CC2 pin
62	I2C_SDA	Digital I/O	I2C SDA
63	I2C_SCL	Digital I/O	I2C SCL
64	RNG_FT_S	Analog Out	Ringing node LP filter Send
65	RNG_FT_R	Analog In	Ringing node LP filter Return
66	QF_DRV	Analog Out	Q factor driver output
67	RNG_SNS	Analog In	Ringing node sense / PC14-OSC32-IN
68	AGND	Power	AFE analog Ground

4.2 Internal signal connection

Table 59. STWBC2-HP PAD Mapping

STWBC2-HP pin	AFE pad	MCU Pad	MCU Function	Type	I/O structure
EP	SUB<1:2>	-	-	A	-
1	-	PC15-OSC32-OUT	PC15	I/O	FT
2	LDO_3V3 <1:2>	VBAT	VBAT	I/O	-
3	-	VREF+	VREF_OUT	I/O	-
4	-	VDD<1:3>	VDD	S	-
5	LDO_1V8<1:2>	-	-	A	-
6	VDD<1:3>	-	-	A	-
7	-	VSS/VSSA<1:3>	VSS	S	-
8	-	VSS/VSSA<4:5>	VSS	S	-
9	BUCK_FB	-	-	A	-
10	-	PF0-OSC_IN	PF0-OSC_IN	I/O	FT
11	-	PF1-OSC_OUT	PF1-OSC_OUT	I/O	FT
12	NRST	PF2 - NRST	PF2-NRST	I/O	-

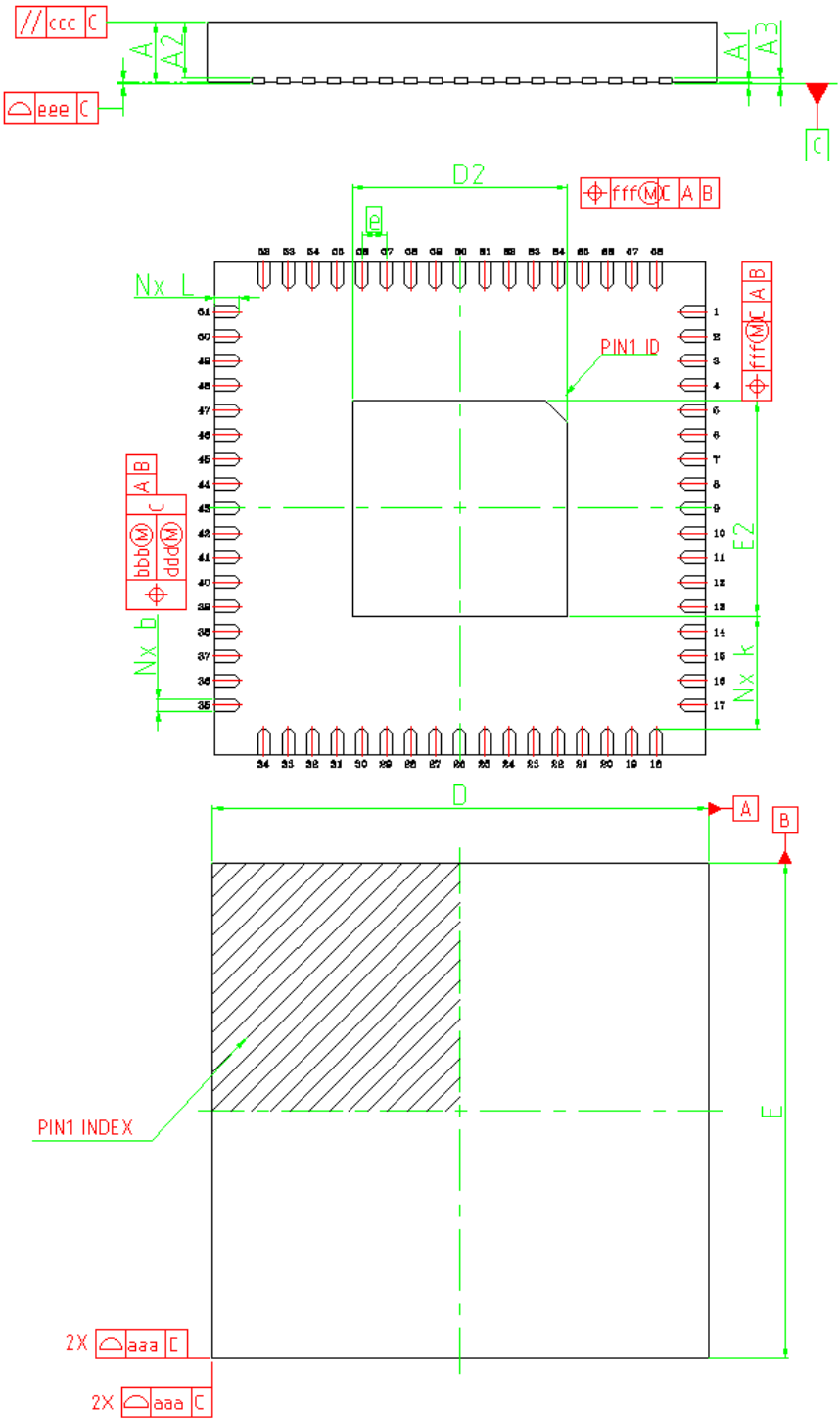
STWBC2-HP pin	AFE pad	MCU Pad	MCU Function	Type	I/O structure
13	PGND<1:2>	-	-	S	-
14	BUCK_SW<1:2>	-	-	A	-
15	BUCK_BT	-	-	A	-
16	VIN<1:2>	-	-	S	-
17	VIN_SNS, VIN_ISNS_P	-	-	A	-
18	VIN_ISNS_N	-	-	A	-
19	ADC0	PA0	ADC_IN0	I/O	FT_a
20	-	PA1	SPI1_SCK	I/O	FT_a
21	-	PA2	USART2_TX	I/O	FT_a
22	-	PA3	USART2_RX	I/O	FT_a
-	DAC0	PA4	DAC_OUT1	I/O	FT_a
-	DAC1	PA5	DAC_OUT2	I/O	FT_a
23	-	PA6	SPI1_MISO	I/O	FT_a
24	-	PA7	SPI1_MOSI	I/O	FT_a
-	ADC0	PC4	ADC_IN17	I/O	FT_a
-	ADC1	PC5	ADC_IN18	I/O	FT_a
25	-	PB0	SPI1_NSS	I/O	FT_da
-	ADC2	PB1	ADC_IN9	I/O	FT_a
-	SPARE_ADC	PB2	ADC_IN10	I/O	FT_a
-	SPARE	PB10	PB10	I/O	FT_fa
-	SPI2_MOSI	PB11	SPI2_MOSI	I/O	FT_fa
-	SPI2_NSS	PB12	SPI2_NSS	I/O	FT_a
-	SPI2_SCK	PB13	SPI2_SCK	I/O	FT_f
-	SPI2_MISO	PB14	SPI2_MISO	I/O	FT_f
-	PDEMOD	PB15	PB15	I/O	FT_c
-	MCO	PA8	MCO	I/O	FT_c
-	ADCMUX_TRIG	PA9	PA9	I/O	FT_fd
-	VDEMOD	PC7	PC7	I/O	FT
-	IDEMOD	PD8	PD8	I/O	FT
-	AFE_INT	PD9	PD9	I/O	FT
-	AFE_RST	PA10	PA10	I/O	FT_fd
26	VDOUB_SW	-	-	A	-
27	VDD_DR<1:3>	-	-	S	-
28	HS1_BT	-	-	A	-
29	HS1_GD	-	-	A	-
30	HS1_SW	-	-	A	-
31	LS1_GD	-	-	A	-
32	LS2_GD	-	-	A	-
33	DCDC_SNS	-	-	A	-
34	HS2_SW	-	-	A	-

STWBC2-HP pin	AFE pad	MCU Pad	MCU Function	Type	I/O structure
35	HS2_GD	-	-	A	-
36	HS2_BT	-	-	A	-
37	HS3_BT	-	-	A	-
38	HS3_GD	-	-	A	-
39	HS3_SW	-	-	A	-
40	LS3_GD	-	-	A	-
41	GND_DR<1:3>	-	-	S	-
42	GPIO_DR0	PA11	PA11	I/O	FT_f
43	GPIO_DR1	PA12	PA12	I/O	FT_f
44	GPIO_DR2	-	-	A	FT
45	GPIO_DR3	-	-	A	FT
46	LED1	-	-	A	-
47	LED2	-	-	A	-
48	-	PA13	SWDIO	I/O	FT
49	-	PA14	SWCLK	I/O	FT
50	BRG_VSNS	-	-	A	-
51	DCDC_FB	-	-	A	-
52	BRG_ISNS_M	-	-	A	-
53	BRG_ISNS_P	-	-	A	-
54	BRG_VREF	-	-	A	-
55	BRG_IDEM	-	-	A	-
56	BRG_FT_S	-	-	A	-
57	BRG_FT_R	-	-	A	-
58	-	PD0	USBPD2_CC1	I/O	FT_c
	-	PD1	UCPD2_DBCC1	I/O	FT_d
59	USB_DP	-	-	A	-
60	USB_DM	-	-	A	-
61	-	PD2	USBPD2_CC2	I/O	FT_c
	-	PD3	USBPD2_DBCC2	I/O	FT_d
62	-	PB7	I2C1_SDA	I/O	FT_fa
63	-	PB8	I2C1_SCL	I/O	FT_f
64	RNG_FT_S	-	-	A	-
65	RNG_FT_R	-	-	A	-
66	QF_DRV	PC_14-OSC32IN	PC14	A	-
67	RNG_SNS	-	-	I/O	FT
68	AGND<1:2>	-	-	S	-

5 Package characteristics

5.1 Package design overview

Figure 78. **Package drawing VFQFPN68L**



5.2 Package mechanical data

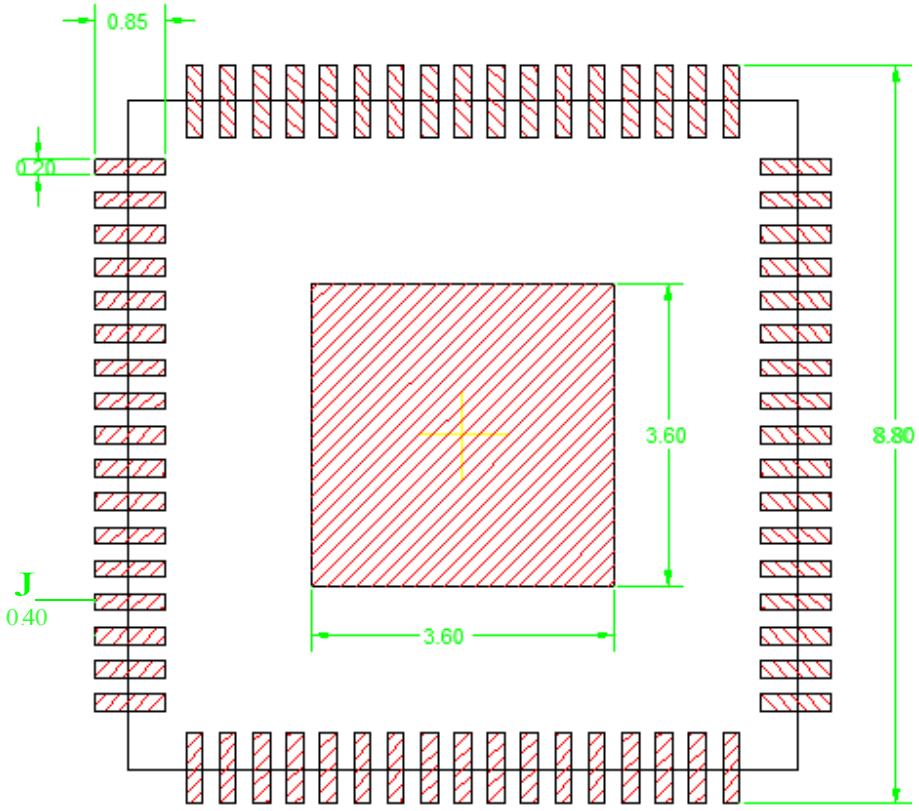
Figure 79. **Package mechanical data VFQFPN68L**

	DIMENSIONS						
	DATABOOK (mm)			DRAWING (mm)			
REF.	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	NOTES
A	0.80	0.90	1.00	0.90	0.95	1.00	
A1		0.02	0.05	0.00	-	0.05	
A3		0.10 REF.			0.10 REF		
b	0.15	0.20	0.25	0.15	0.20	0.25	
D	7.90	8.00	8.10	7.90	8.00	8.10	
D2	SEE EXPOSED PAD VARIATION			3.40	3.50	3.60	
E	7.90	8.00	8.10	7.90	8.00	8.10	
E2	SEE EXPOSED PAD VARIATION			3.40	3.50	3.60	
e		0.40		0.40 BSC.			
L	0.30	0.40	0.50	0.30	0.40	0.50	

SYMBOL	TOLERANCE OF FORM AND POSITION	
	DATABOOK	
aaa	0.15	
bbb	0.10	
ccc	0.10	
ddd	0.05	
eee	0.08	
fff	0.10	
NOTE	1,2	
REF		

5.3 Recommended footprint

Figure 80. **Recommended footprint**



Revision history

Table 60. **Document revision history**

Date	Version	Changes
14-Sep-2021	1	Initial release.

Contents

List of tables

List of figures