Inverting Buck-Boost with XDPP1100

XDPP1100 Firmware

About this document

Scope and purpose

This document describes a custom firmware implementation for XDPP1100 microcontroller to regulate an inverting buck-boost topology.

Buck-boost implemented features are listed below:

- output current (IOUT) correction feature – new formula is applied to recalculate the output current;
- efficiency table feature – input power and input current telemetry correction based on output power and efficiency;
- feed-forward feature – new formula is applied to recalculate the feed-forward transient behavior;
- VIN telemetry trim feature (disable)– changing voltage trim config based on regulation mode;
- dynamic dead-time feature (disable) – applying larger dead-time for startup;
- active current sharing feature (disable) – enable current sharing by controlling IMON resistor connection;
- ZVS feature – power delivery improvements based on the output current.
- Diode emulation at light load – disabling SR PWM if the output load is lower than certain user's threshold.
- Phase shedding at light load - disabling phase 2 if the output load is lower than certain user's threshold.
- Burst mode at light load - PWM pulses skipping once if the load current is lower than certain user's threshold.

Each feature is described in its respective section in this document. Moreover, we show how the user might use them.

Intended audience

Power management engineers who wish to explore Inverting Buck-Boost solution with XDPP1100.

Table of contents

1 Inverting buck-boost firmware

The custom buck-boost firmware is developed and combined into one single XDPP1100 patch. In case you are unfamiliar with XDPP1100 firmware projects, refer to “XDPP1100 Firmware Development Guide” and “XDPP1100 Firmware Examples Code”.

1.1 Patch usage

The buck-boost patch is delivered with a pre-built image, which the user can find */build* folder. This image can be stored to XDPP1100 RAM or OTP via GUI. Upload */src/shasta_pmbus.xlsx* to GUI to use buck-boost features through MFR PMBus commands.

Table 1 Buck-boost MFR PMBus commands

PMbus command name	PMBus address	Read / Write	Description
MFR_ESTIMATE_EFFICIENCY	(0xB1)	Read only	Read feedback of an estimated efficiency value in u0.8 format.
MFR_ZVS_DISABLE_THRESHOLD	(0xB2)	Write / Read	To set desired ZVS threshold in Linear11 format. ZVS threshold hysteresis is 1 A. Can be modified. If “0”, then ZVS feature is disabled.
MFR_LIGHT_LOAD_THRESHOLD	(0xB3)	Write / Read	To set desired LIGHT_LOAD threshold in Linear11 format. ZVS threshold hysteresis is 2 A. Can be modified.
MFR_LIGHT_LOAD_MODE	(0xB4)	Write / Read	To set desired mode during light load: '1' - phase shedding mode is enabled. '2' - diode emulation mode is enabled; ‘3’ – burst mode (not implemented); '0' to use nothing.
MFR_BURST_CONFIG	(0xB5)	Write / Read	To set desired burst configuration: [7:4] desired value ‘2’…’15’, which means “skip every value (1/fsw * BURST_IRQ_RATE)”; [3:0] desired BURST_IRQ_RATE ‘2’…’7’, where ‘2’ means 2 cycles, ‘3’ – 4 cycles, ‘4’– 8 cycles, ‘5’ – 16 cycles, ‘6’ – 32 cycles, ‘7’ – 64 cycles. These settings are applied once MFR_LIGHT_LOAD_MODE is ‘3’.
MFR_EN_BUCK_BOOST_FEED_FORWARD	(0xB6)		Set "1" to enable FF, set "0" to disable it
MFR_IOUT_OFFSET_CORR_SLOPE_FACTOR	(0xB7)		Exponent -8 offset_corr = [MFR_IOUT_OFFSET_CORR_SLOPE_FACTOR

PMbus command name	PMBus address	Read / Write	Description
			x Vin_telem + MFR_IOUT_OFFSET_CORR_FACTOR],
MFR_IOUT_OFFSET_CORR_FACTOR	(0xBC)		Exponent -1 offset_corr = [MFR_IOUT_OFFSET_CORR_SLOPE_FACTOR x Vin_telem + MFR_IOUT_OFFSET_CORR_FACTOR],
MFR_VIN_TRIMMING_ACTIVE	(0xBD)	Write / Read	To set input voltage trim for active mode: [31:16] desired vin_pwl_slope in format u-2.14; [15:0] desired vin_trim in format s.6.4.
MFR_VIN_TRIMMING_STANDBY	(0xBE)	Write / Read	To set input voltage trim for standby mode: [31:16] desired vin_pwl_slope in format u-2.14; [15:0] desired vin_trim in format s.6.4.
MFR_DEADTIME	(0xC0)	Write / Read	To set the larger dead-time for a soft-startup in PWM_DEADTIME format.
MFR_ISHARE_THRESHOLD	(0xDA)	Write / Read	To set current sharing deadzone in linear11 format Amps units.
MFR_ADDED_DROOP_DURING_RAMP	(0xFC)	Write / Read	To set the additional droop resistor applies to loop during startup ramp. LINEAR11 with Binary point at 1 mOhm.

Attention: Perform features' enabling/disabling/config settings via respective PMBus commands only when XDPP1100 regulation is OFF to avoid unexpected artifacts.

1.2 Features

This section provides general overview of implemented buck-boost features. For more details and code modification, refer to the section **2 "Advanced description and custom settings"**.

1.2.1 Output current correction feature

This feature allows to get the output current telemetry from the combination ISEN + BISEN sources and adjusts it based on a dead-time and duty cycle values, according to the following equation:

$$Output\ Current = (1 - Duty\ Cycle\ \pm Deadtime) * Inductor\ Current$$

In addition to the formula above, we apply another output current offset to enhance accuracy of the output current measurement in buck-boost board, which depends on input voltage (see the results on Figure 1 below).

The equation is:

$$Full\ Output\ Current = Output\ Current + Output\ Current\ vs\ Input\ Voltage\ offset$$

$$Output\ Current\ vs\ Input\ Voltage\ offset = -\ Slope\ Factor * Input\ Voltage + Factor$$

Slope Factor is defined by (0xB7) MFR_IOUT_OFFSET_CORR_SLOPE_FACTOR and Factor is defined by (0xBC) MFR_IOUT_OFFSET_CORR_FACTOR. Default value for Factor is 30.5.

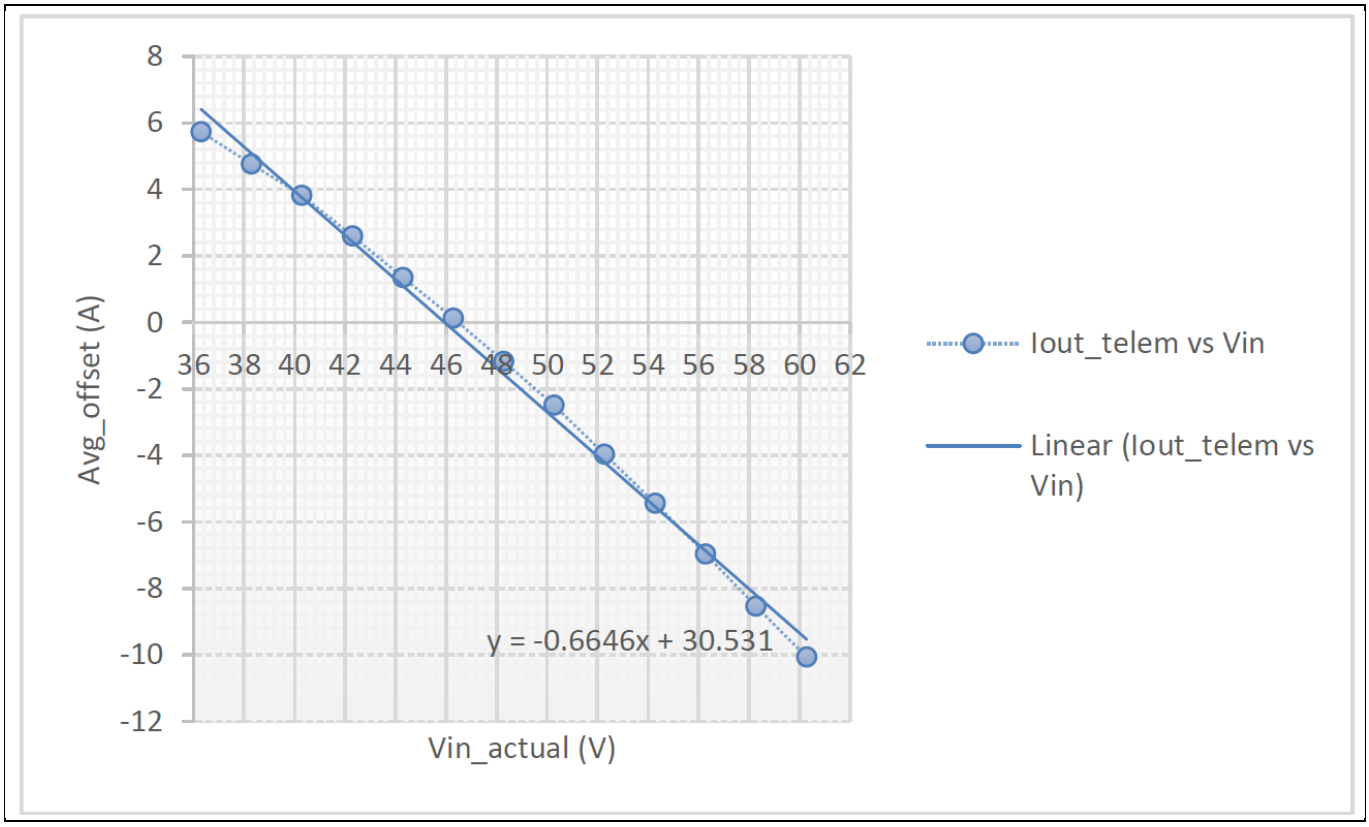

Figure 1 Iout_offset vs Vin attitude results after lab testing

Table 2 IOUT Offset Commands

Command	Value	Comments
MFR_IOUT_OFFSET_CORR_SLOPE_FACTOR	0xAA	Value is set in u0.8 format Example: $\frac{0.6646}{2^{-8}} = 170$ or $'0xAA'$
MFR_IOUT_OFFSET_CORR_FACTOR	0x3D	Value is set in u7.1 format Example: $\frac{31.5}{2^{-1}} = 61$ or $'0x3D'$

1.2.2 Efficiency table feature

This feature allows to improve a measurement accuracy of the input current/power telemetry based on pre-programmed power efficiencies for different ranges.

To reflect more accurate readings on the input current telemetry, the following equation is used:

$$Input\ Current\ Corrected = Input\ Current\ /\ Efficiency\ LUT\ value$$

where *Efficiency LUT value* is selected from the Efficiency Look-Up Table based on the output current and the input voltage current ranges of operation.

In the default buck-boost project, the following Efficiency Look-Up Table is programmed:

Table 3 Efficiency Look-Up Table

Output Current / Input Voltage	1.25 A	2.5 A	6.25 A	12.5 A	15.0A
36 Vdc	85.4 %	91.5 %	95.6 %	96.4 %	96.4%
48 Vdc	82.4 %	89.7 %	94.8 %	96.2 %	96.2%
60 Vdc	79.3 %	88.0 %	94.0 %	95.9 %	95.9%

The input current/power correction happens every frequency iteration.

Efficiency Look-Up Table hysteresis 1 V and 0.75 A for the input voltage for the output current respectively.

Feature example

If the output current is 10 A and the input voltage is 42 V, then Efficiency Look-Up Table values is 97.32 %.

The MFR command MFR_ESTIMATE_EFFICIENCY is a debug command. Read this command returns efficiency code. For examples, read value 0xD3 = 249, indicate the efficiency is 97.3%.

1.2.3 Feed-forward feature

This is a re-engineered feed-forward feature to support inverting buck-boost topology.

This feed-forward feature is executed every 8th regulation frequency switch, or:

$$Feed\ Forward\ Frequency = Regulation\ Frequency/8$$

To enable the buck-boost feed-forward feature, set the patched command "MFR_EN_BUCK_BOOST_FEED_FORWARD" to 1 and "FW_CONFIG_REGULATION" command as show in Figure 2.

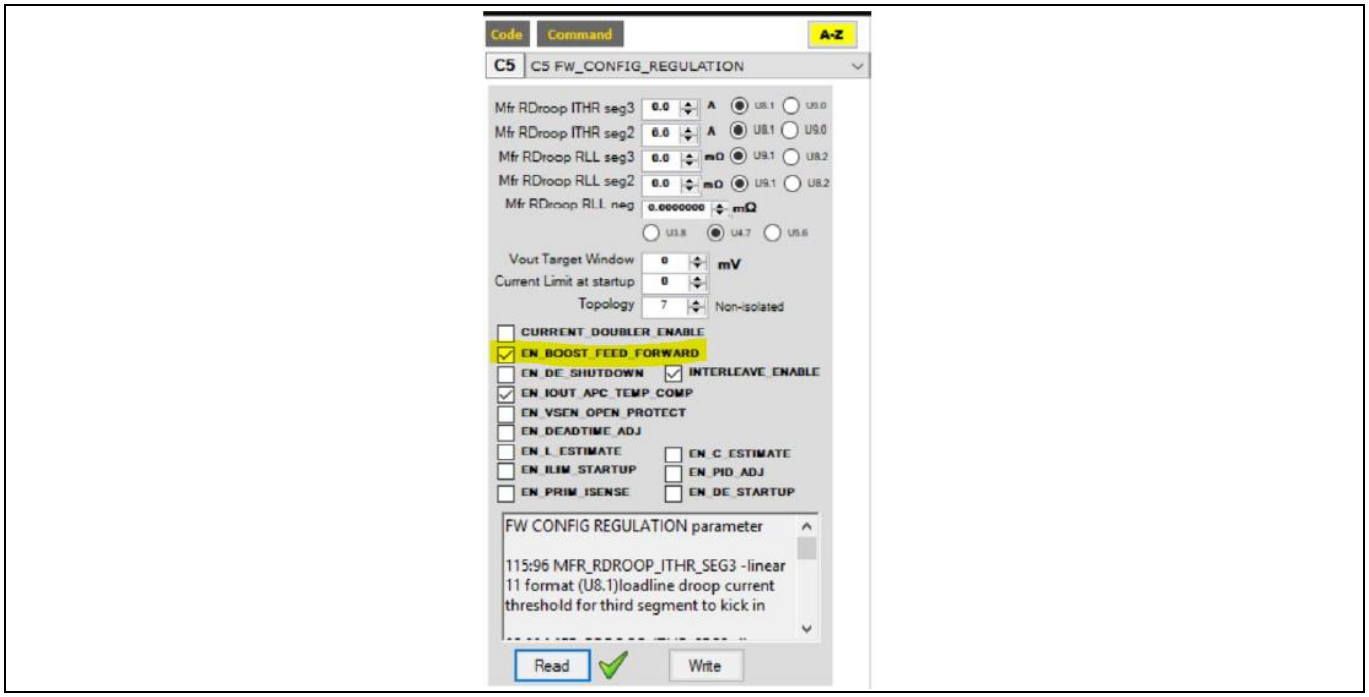

Figure 2 FW_CONFIG_REGULATION config the buck_boost feed-forward

1.2.4 VIN telemetry trim feature (disable)

The input voltage is sensed by the auxiliary power supply through the secondary winding. This voltage will change with operation conditions. The voltage is lower in the OFF state when the gate drivers are not switching, the voltage is higher in regulation mode. So we use two sets of parameters to improve the accuracy of VIN telemetry. The PRISEN registers are shown in Table 4. And the configuration can be done by the patched MFR commands that listed in Table 4.

Table 4 PRISEN configuration

Register	Value	Meaning
tlm0_vin_src_sel	3	TS ADC Vin. Input voltage is sensed at PRISEN pin.
vin_pwl_slope	2462	LSB = 2^-14, u-2.14
vin_trim	0	Vin = ADC * vin_pwl_slope * 2^-14 + vin_trim LSB = 62.5mV, s.6.4

The **vin_pwl_slope** assumes a linear slope of the PRISEN signal, which is proportional to input voltage V_{IN}. The **vin_pwl_slope** can be calculated by the following equation:

$$vin\_pwl\_slope = \frac{1.2 \times 2^5}{PRISEN\_SCALE}$$

The PRISNE resistor divider ratio is set to 1.3 / (82 + 1.3) =0.0156.

$$vin\_pwl\_slope = \frac{1.2 \times 2^5}{0.0156} = 2462$$

The Vin telemetry offset can be configured by **vin_trim** register. Table 5 shows example of how-to config the MFR_VIN_TRIMMING command.

Table 5 MFR_VIN_TRIMMING

Command	Value	Comments
MFR_VIN_TRIMMING_ACTIVE	0xB09F0000	Configure the vin_pwl_slope and vin_trim in active mode. The upper two bytes defines vin_pwl_slope register in Linear 11 format of exponent -10. 0xB09E, Linear11 format, convert to decimal = 0.155, it sets the register of vin_pwl_slope = 0.155*2^14 = 2540 The lower two bytes defines vin_trim register. 0x0000, linear 11 format, it sets the vin_trim = 0
MFR_VIN_TRIMMING_STANDBY	0xB09EE00D	Configure the vin_pwl_slope and vin_trim in standby mode. The upper two bytes defines vin_pwl_slope register in Linear 11 format of exponent -10. 0xB09E, Linear11 format, convert to decimal = 0.154, it sets the vin_pwl_slope = 0.154*2^14 = 2523 The lower two bytes defines vin_trim register in Linear 11 format of exponent -4.

Command	Value	Comments
		0xE00D, linear 11 format, convert to decimal = 0.813, it sets the register of vin_trim = 0.813/0.0625 = 13

1.2.5 Dynamic Dead-time feature (disable)

This feature allows to use a larger dead-time on PWM2 and PWM6 for startup and change it back to the normal set of dead-times when the output voltage is at its target value.

(0xC0) MFR_DEADTIME command was added to set a custom larger dead-time for the startup. It has the same format as the regular PWM_DEADTIME command.

Pwm2_dr = pwm6_dr = MFR_DEADTIME / 1.25ns, for startup.

The MFR_DEADTIME will override the patched pwm2 and pwm6 dead-times.

Table 6 MFR_DEADTIME

Command	Value	Comments
MFR_DEADTIME	0xF0	Set the rise dead-time of PWM2, PWM6 in TON_RISE state. 0xF0 = 240, dead-time = 240 * 1.25ns = 300ns. The default dead-time will restore after the output voltage reaching to the target. The default dead-time is configured by command 0xCF.

1.2.6 Active Current sharing feature (disable)

The active current sharing patch disconnects the R_{ishare} resistor from the IMON bus when the converter is not in operation. PWM11 is used as the on/off switch. The output of PWM11 is in tri-state (HiZ) when the XDPP1100 is not biased or in OFF state, the R_{ishare} is floating and won't affect IMON bus voltage. PWM11 will pull down and connects R_{ishare} to ground when the XDPP1100 is enabled.

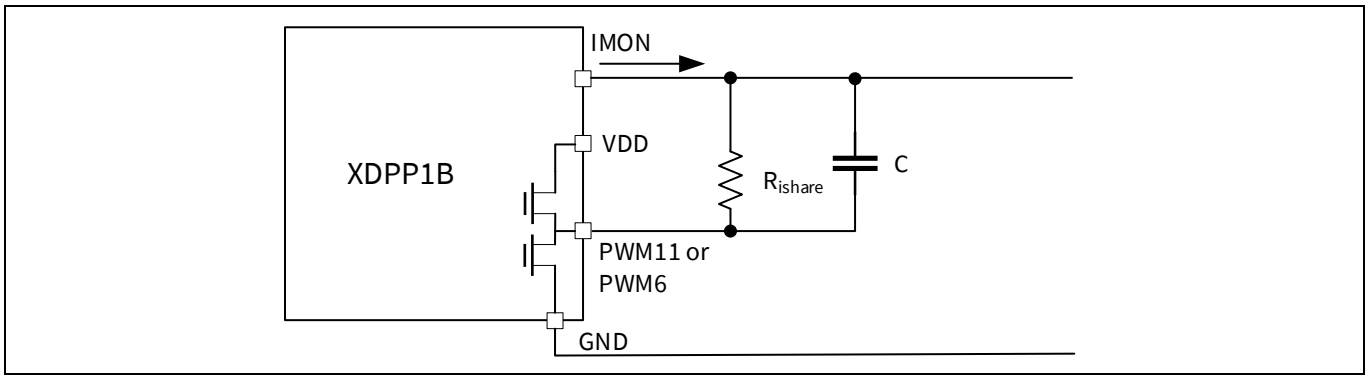

Figure 3 Use PWM11 to disconnect R_{ishare}

Prior to startup, the IMON DAC is disabled (ts_tsidac_imon_sel=0), en_ishare is disabled (en_ishare=0). IMON resistor is disconnected from the circuit with PWM11 output floating. When the converter is enabled, active current sharing will be enabled (en_ishare=1) if the MFR PMBus command 0xDA MFR_ISHARE_THRESHOLD is none zero. The IMON current source is enabled (ts_tsidac_imon_sel=1), and PWM11 will be pulled low to have IMON resistor connected in the circuit. The active current sharing is enabled at the beginning of output ramp.

Current sharing during start-up is always more challenging than in steady-state. The power supplies could have different start-up delays or different ramp times due to device variation. The voltage difference between units could be much more than the set-point error in the steady-state.

An added_droop feature was implemented in the FW patch. The droop is added during start-up ramp. The added_droop helps to reduce the error of current sharing at start-up. It is removed once the power supply reaches regulation, thus the maximum output power won't be sacrificed. It could be configured by the patched PMBus command 0xFC MFR_ ADDED_DROOP_ DURING_RAMP.

Table 7 Current share commands

Command	Value	Comments
MFR_ISHARE_THRESHOLD	0x01	Defines current sharing threshold. When the error between the average current and local current is less than the threshold, active current sharing stopped adjusting Vout. LINEAR11, unit Amps 0x01 set the ishare threshold to 1A Set this parameter to 0 to disable active current sharing.
MFR_ADDED_DROOP_ DURING_RAMP	0x00	Defines added droop during startup ramp. LINEAR11, unit mΩ

1.2.7 ZVS feature

At 5 A and below the efficiency is better if ZVS is disabled (PWM3 and PWM4 are disabled). See Table 8 below.

Table 8 General efficiency with ZVS and without ZVS

Iout (amps)	42Vdc with ZVS	42Vdc without ZVS parts	42Vdc without ZVS	48Vdc with ZVS	48Vdc without ZVS parts	48Vdc without ZVS	57Vdc with ZVS	57Vdc without ZVS parts	57Vdc without ZVS
2.5	84.31%	86.08%	86.06%	82.75%	84.41%	84.58%	80.67%	82.57%	82.51%
5	90.83%	92.30%	91.55%	89.97%	91.61%	91.16%	88.71%	90.66%	90.29%
10	94.60%	93.90%	93.56%	94.07%	93.34%	92.96%	93.29%	92.48%	92.04%

The ZVS patch feature enable ZVS at startup, once reach target Vout it will disables ZVS once the output load is below ZVS disable threshold.

The user can config ZVS threshold level with (0xB2) MFR_ZVS_DISABLE_THRESHOLD PMBus command. Its format is similar to (0x8C) READ_IOUT with its READ_IOUT_EXP exponent value. Default value is 0.

If MFR_ZVS_DISABLE_THRESHOLD is '0', then ZVS will not be disable at low output load.

Take a note, once READ_IOUT_EXP is updated, rewrite MFR_ZVS_DISABLE_THRESHOLD with new value according to new READ_IOUT_EXP.

In addition, a hysteresis 1 A is implemented to avoid ZVS jittering.

Feature example

If READ_IOUT_EXP is '-3' and the user wants to have 5 A ZVS threshold level, set MFR_ZVS_DISABLE_THRESHOLD to '0xE828' (or 40dec).

1.2.8 Modes at light load

Use MFR_LIGHT_LOAD_MODE to set a phase shedding '1', diode emulation '2' or burst '3' once the output load is lower a threshold defined by MFR_LIGHT_LOAD_THRESHOLD in Linear11 format. Set '0' to disable any light load mode. Default value is '0'.

There is a hardcoded 2 A output load hysteresis to avoid jingling, which is applied to the 3 light load features, which are described in 1.2.8.1, 1.2.8.2 and 1.2.8.3.

Light load modes are enabled once XDPP1100 reaches a target output voltage. It might be changed if desired.

Note: ZVS feature can be in operation with one of light load feature in parallel.

1.2.8.1 Diode emulation

To improve efficiency of the system, we disable SR PWMs (defined by 0xC4 FW_CONFIG_PWM, bits [27:16]) every time, when the output load is lower, than a certain threshold. Once the output load is above this threshold, SR PWMs are enabled again.

DE mode at light load:

- When IOUT is lower than FW_CONFIG_DE_THRESH, turn off SR. The SR PWMs are defined by 0xC4 FW_CONFIG_PWM, bits 27:16.
- When IOUT is higher than (FW_CONFIG_DE_THRESH + hysteresis), enable SR.

1.2.8.2 Phase shedding

Drop phase 2 when the load current is lower than the MFR_LIGHT_LOAD_THRESHOLD. Enable phase 2 when output current is higher than MFR_LIGHT_LOAD_THRESHOLD + hysteresis. The same PID compensation is used in single phase mode. Different compensation is also possible if desired.

1.2.8.3 Burst mode

This feature allows to perform a PWM pulses skipping once the load current is lower than the MFR_LIGHT_LOAD_THRESHOLD. When output current is higher than MFR_LIGHT_LOAD_THRESHOLD + hysteresis, XDPP1100 starts normal regulation.

Burst mode be customized via (0xB5) MFR_BURST_CONFIG.

Feature example

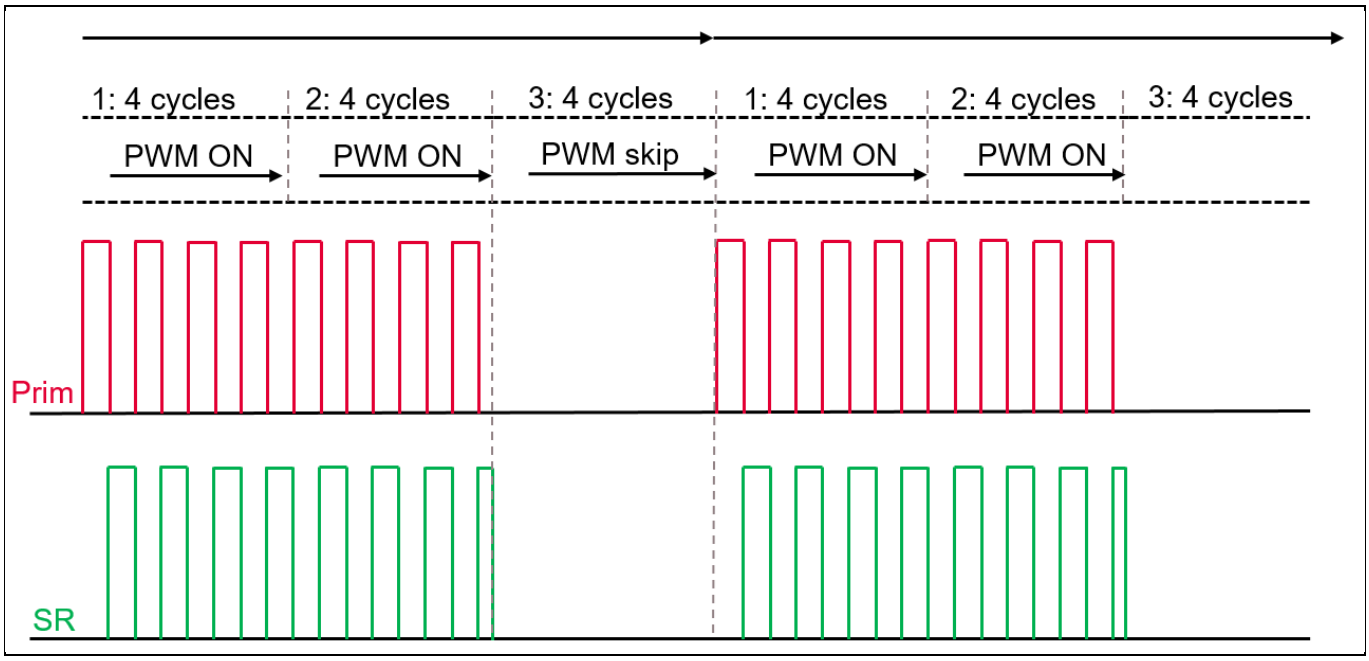

Figure 4 MFR_BURST_CONFIG '33' representation

The picture above is an overview of burst mode if MFR_BURST_CONFIG is set as '33'. First '3' ([7:3] bits) means "every 3rd skip", and second '3' means "at IRQ_RATE 4 cycles. You might imagine from that case that you have 3 iterations, each of them has 4 cycles. PWM skipping happens at the 3rd iteration. Take a note, that 1 cycle is equal to 1/fsw, where fsw is a frequency switch.

If MFR_BURST_CONFIG = '45', then PWM skipping happens every 4th iteration at IRQ_RATE 32 cycles.

Attention: Due to firmware limitation settings IRQ_RATE has to be higher than 2. Otherwise, XDPP1100 is crashed. Recommended settings: '33' and higher.

2 Advanced description and custom settings

This section shows where, what and how the user can modify each described feature.

In the buck-boost firmware project, the list of features is shown in the file */src/user_app/buck_boost.h*

```
0033   #define ishare_feature_en
0034   #define eff_table_feature_en
0036   #define iout_correction_feature_en
0037   #define feed_forward_feature_en
0038   #define deadtime_feature_en
0039   #define vin_trim_feature_en
0040   #define zvs_feature_en
0040   #define fan_en
0040   #define light_load_features_en
```

There is an option to build the patch without certain non-desired features. To do that, simply comment **#define feature_en** out in buck_boost.h and re-compile the project.

The default project patch is build without these features, ishare_feature_en, deadtime_feature_en, vin_trim_feature_en, zvs_feature_en and fan_en.

Explore the code below in its given order to get know each feature better. Relevant comments are presented and they will navigate you through the code.

Table 9 Project Blocks

Filename	Description and Properties
buck_boost.c/h	Buck-boost feature holder
user_app.c/h	User entry point
add_on_features.c/h	Settings initialization and current share feature
efficiency_table.h	Efficiency Look-Up Table storage
regulation_state_machine_callbacks.c/h	State machine
pmbus_mfr_autogen.c/h	PMBus generated commands
pmbus_mfr_specific_handlers.c/h	PMBus commands' handlers
deadtime.c/h	Dead-time feature
vin_trim.c/h	Input voltage trim feature
zvs_feature.c/h	ZVS feature
user_ntc_temperature_lut.h	Temperature Sense Look-Up Table storage

2.1 Features' implementation

Each feature implementation can be found per sections below. Before starting exploring each feature, it's important to get the idea of features' thread, which was created to lower CPU load down.

ZVS feature and light load features were merget under one IRQ interrup handler assuming these features triggering are depended on only the output load level. These features were inserted into a feed-forward IRQ

handler *buck_boost.c/patch_Regulation_Compute_Feed_Forward()*, assuming default 8 cycles IRQ is sufficient to handle switching of light load modes.

In addition, assuming we care only about loop 0, a telemetry update for loop 1 was disabled in *buck_boost.c/patch_Telemetry_Sample()*.

2.2 Output current correction feature

Patch code reference:

```
iout_correction_feature_en
```

To support fast frequency output current telemetry update and avoiding excessive PMBus reads (which are relatively slow), *buck_boost.c/iout_telemetry_get_high_frequence()* was created. Still, *buck_boost.c/ patch_Telemetry_get ()* updates needed output current parameters (like pwm1_deadtime_rise, etc.) every 1ms, which is sufficient.

Table 10 Output Current Correction Additions/Modifications

Filename	Function and/or Variable Name
buck_boost.c/h	patch_Telemetry_get () iout_telemetry_get_high_frequence () iout_telemetry_get_high_frequence_limited ()

2.3 Efficiency table feature

Patch code reference:

```
eff_table_feature_en
```

Refer to */src/user_app/efficiency_table.h* to set a custom Efficiency Look-Up Table in u0.8 format.

Refer to */src/user_app/add_on_features.c* to set custom Efficiency Look-Up Table hysteresis for the input voltage and the output current.

Note: Do not change Efficiency look-Up Table size. Hysteresis function is hardcoded and is only appropriate for the same Efficiency Look-Up Table size as per default.

Table 11 Feature Additions/Modifications

Filename	Function and/or Variable Name
buck_boost.c/h	patch_Telemetry_get ()
efficiency_table.h	efficiency_table[][] vin_table[] iout_table[]
add_on_features.c	add_on_features_init ()

2.4 Feed-forward feature

Patch code reference:

```
feed_forward_feature_en
```

Refer to */src/user_app/buck_boost.c/patch_Regulation_Compute_Feed_Forward ()* to get more details about feed-forward implementation for an inverting buck-boost topology.

Refer to */src/user_app/regulation_state_machine_callbacks.c/TON_RISE_ENABLE ()* under *feed_forward_feature_en* to change a feed-forward IRQ rate execution. Change *fsw_irq_rate_sel_X*, where X – numbers of cycles, and:

$$Feed\ Forward\ Frequency = Regulation\ Frequency / X$$

There are following oprions for the user to set:

Code Listing 1 Examples. Safe rates are in range from 8 to 64 frequency cycles

```
Regulation_setup_fsw_irq(loop, fsw_irq_idx_1, fsw_irq_rate_sel_8);
Regulation_setup_fsw_irq(loop, fsw_irq_idx_1, fsw_irq_rate_sel_16);
Regulation_setup_fsw_irq(loop, fsw_irq_idx_1, fsw_irq_rate_sel_32);
Regulation_setup_fsw_irq(loop, fsw_irq_idx_1, fsw_irq_rate_sel_64);
```

Attention: Don't set feed-forward IRQ rate less than fsw_irq_rate_sel_8. This will cause reset issue.

Attention: Once feed-forward IRQ is changed, ZVS and light load features will have same IRQ rate.

Table 12 Feature Additions/Modifications

Filename	Function and/or Variable Name
buck_boost.c/h	patch_Regulation_Compute_Feed_Forward ()
regulation_state_machine_callbacks.c	TON_RISE_ENABLE () AT_SHUTDOWN ()

2.5 VIN telemetry feature

Patch code reference:

```
vin_trim_feature_en
```

Refer to */src/user_app/vin_trim.c/h* to get more details about the input voltage trim implementation.

Table 13 Feature Additions/Modifications

Filename	Function and/or Variable Name
vin_trim.c/h	Update_Vin_Trim () PMBUS_HANDLE_MFR_FLEX_VIN_TRIMMING_ACTIVE () PMBUS_HANDLE_MFR_FLEX_VIN_TRIMMING_STANDBY ()
regulation_state_machine_callbacks.c/h	regulation_sm_callbacks_init () TON_RISE_ENABLE () AT_SHUTDOWN () AT_TARGET_ENABLE ()

2.6 Dynamic dead-time feature

Patch code reference:

`deadtime_feature_en`

Refer to */src/user_app/deadtime.c/h* to get more details about active current sharing implementation.

Table 14 Feature Additions/Modifications

Filename	Function and/or Variable Name
deadtime.c/h	Set_StartUp_Deadtime () Set_SteadyState_Deadtime () PMBUS_HANDLE_MFR_DEADTIME ()
regulation_state_machine_callbacks.c/h	regulation_sm_callbacks_init () AT_SHUTDOWN () TON_RISE_VID_REACHED () TON_DELAY_ENABLE ()

2.7 Active current sharing feature

Patch code reference:

`ishare_feature_en`

Refer to */src/user_app/add_on_features.c/h* to get more details about active current sharing implementation.

Table 15 Feature Additions/Modifications

Filename	Function and/or Variable Name
add_on_features.c/h	added_droop_disable() added_droop_enable() remove_added_droop_irq_callback() enable_ishare() disable_ishare() patch_Regulation_Shutdown_Sequence()
regulation_state_machine_callbacks.c/h	regulation_sm_callbacks_init () TON_RISE_ENABLE () AT_SHUTDOWN () TON_RISE_VID_REACHED ()

2.8 ZVS feature

Patch code reference:

`zvs_feature_en`

Refer to */src/user_app/add_on_features.c* to set a custom ZVS hystoresis: *user_data.iout_zvs_hysteresis* and *user_data.iout_zvs_hysteresis_exp*.

Refer to */src/user_app/zvs_feature.c/h* to get more details about ZVS feature implementation.

Table 16 Feature Additions/Modifications

Filename	Function and/or Variable Name
add_on_features.c/h	add_on_features_init ()

Filename	Function and/or Variable Name
regulation_state_machine_callbacks.c/h	regulation_sm_callbacks_init () TON_RISE_ENABLE () AT_TARGET_ENABLE () AT_SHUTDOWN ()
zvs_feature.c/h	iout_zvs_threshold_irq_handle_enable () iout_zvs_threshold_irq_handle_disable () iout_zvs_threshold_irq_handle () iout_zvs_enable () zsv_enable() PMBUS_HANDLE_MFR_ZVS_DISABLE_THRESHOLD ()
buck_boost.c/h	patch_Regulation_Compute_Feed_Forward () light_load_features_core_function ()

2.9 Light load feature

Patch code reference:

```
light_load_features_en
```

Refer to */src/user_app/add_on_features.c* to set a custom light load features' hystoresis: *user_data.light_load_features_hysteresis* and *user_data. light_load_features_hysteresis_exp*.

Refer to */src/user_app/light_load_features.c/h* to get more details about light load features implementation. Take a note, that SR PWM for each mode are hardcoded, but the user can changed these settings in *light_load_features_enable()*.

In this section burst mode implementation is worth paying attention. Once the output current is lower than defined light load threshold, a new thread fsw_irq_idx_4 is enabled to perform PWM skipping with pre-defined burst settings from its PMBus command. Refer to */src/user_app/pwm_skip.c/h* for more details.

Table 17 Feature Additions/Modifications

Filename	Function and/or Variable Name
add_on_features.c/h	add_on_features_init ()
regulation_state_machine_callbacks.c/h	regulation_sm_callbacks_init () AR_TARGET_ENABLE () AT_SHUTDOWN ()
light_load_features.c/h	light_load_features_threshold_irq_handle_enable () light_load_features_threshold_irq_handle_disable () light_load_features_threshold_irq_handle () light_load_features_enable () PMBUS_HANDLE_MFR_LIGHT_LOAD_THRESH () PMBUS_HANDLE_MFR_LIGHT_LOAD_MODE ()
buck_boost.c/h	patch_Regulation_Compute_Feed_Forward () light_load_features_core_function ()
pwm_skip.c/h	pwm_skip () pwm_unskip ()

Filename	Function and/or Variable Name
	pwm_skip_irq_handle () pwm_skip_irq_enable () pwm_skip_irq_disable () PMBUS_HANDLE_MFR_BURST_CONFIG ()

References

[1] A Reference. See the code examples at **www.infineon.com**

[2] XDPP1100 Datasheet

[3] XDPP1100 Firmware Development Guide

[4] XDPP1100 Firmware Examples Code

Revision history

Document version	Date of release	Description of changes
V 1.0	12-16-2020	Created. All implemented features were described.
V1.1	2-16-2021	Updated chapter 1.2.8, modes at light load
V1.2	2-17-2021	Updated all chapters. Added modes at light load implementation and usage.
V1.3	5-27-2021	Updated Chapter 1.2.1, added equation to get correct Iout telemetry. Updated Chapter 1.2.3, MFR_EN_BUCK_BOOST_FEED_FORWARD
V1.4	6-7-2021	Updated Chapter 1.2.1, Change in PMBus Commands
V1.5	2-10-2022	Update Chapter 1.2.1 with format for PMBus commands. Rearrange all chapters and align with config manual.
V1.6	05-09-2022	Updated Chapter 1.2.7 and 2.8 on ZVS features. Enable ZVS at TON_RISE. Updated Chapter 1.2.2. add additional column to efficiency look up table

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2022-05-09

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
AppNote Number

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.