NXP Semiconductors
Design Reference Manual

Document Number: DRM174
Rev. 0 , 11/2016

Totem-Pole Bridgeless PFC Design Using MC56F82748

1. Introduction

1.1. Application outline

This reference design manual describes a Totem-Pole Bridgeless Power Factor Correction (PFC) design using MC56F82748 DSC.

This document focuses on the key parts design of Totem-Pole bridgeless PFC. System description part includes the system structure, different modulation mode, and control loop. Hardware design part includes PFC choke design, power circuits and drive circuits, sensing circuits and PCB layout consideration. Software design part mainly includes the system state machine, control timing, drive signal generation logic and fault protection.

Contents

1.2. Totem-Pole bridgeless PFC topology and features

Power Factor Correction control scheme is widely adopted for power conversion of AC-DC application.

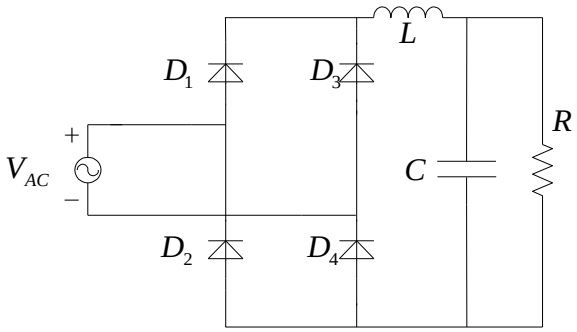


Figure 1. Typical passive power factor correction circuit

The earliest topology applied in AC-DC power conversion is Passive Power Factor Correction (PPFC) technique. The scheme put a filter inductor between the rectifier and bus capacitor. The topology is as *Figure 1*

In practical application, inductor is put between AC source and rectifier bridge. As shown in *Figure* 2, the improved PPFC topology will not have DC component in the inductor, which can prevent the inductor from saturation. The advantages of PPFC are that it is simple, reliable, without need of real-time control and low cost. In addition, this topology will limit the current total harmonic distortion under 30% by suppressing odd harmonics. In this application, inductor L is the “core” of design. When inductance is larger, Total Harmonic Distortion (THD) is smaller and current wave is more sinusoidal. However, the phase difference will become bigger.

The disadvantages of the topology is that the passive component is often heavy with low power factor. This makes the power loss severe and produce a lot of heat. The noise of power frequency vibration is also a problem. The application is adopted under 300W, especially applied in occasion that has no much limit for space, weight but sensitive with cost.

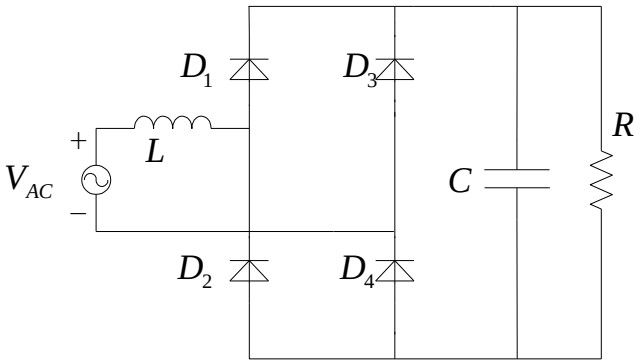


Figure 2. Improved passive power factor correction circuit

The most common topology for AC-DC application is active power factor correction (APFC), it is the most effective way to improve power factor. *Figure 3* shows the principle block diagram of APFC. The basic method is: rectify the alternating voltage to direct value, and then conduct DC-DC transformation by rectified voltage. There are voltage and current loops. Outer loop ensures that the bus voltage keep up with the set value. Inner loop ensures that the current wave can real-time track the input voltage wave.

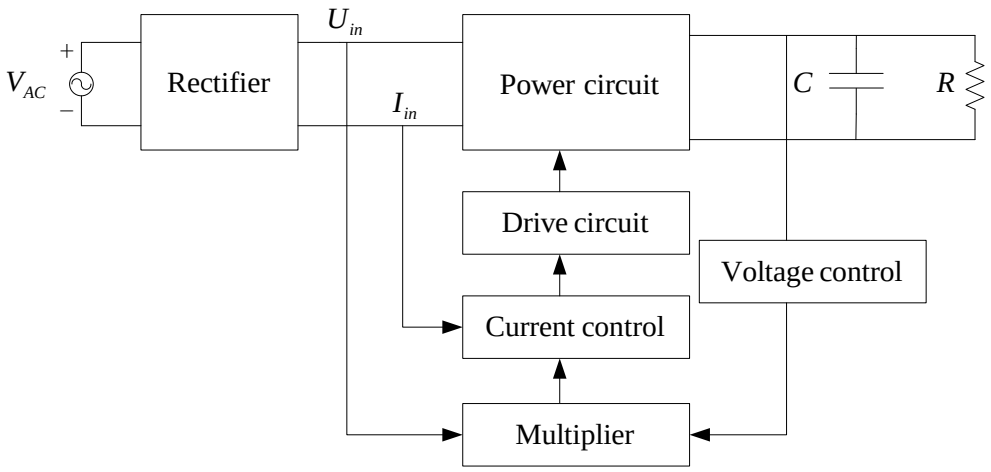


Figure 3. Principle block diagram of APFC

APFC uses full-controlled semiconductor device. *Figure* 4 shows the topology of conventional PFC. The control scheme will help reduce the THD of the current below 5%, and make the PF value over 0.99 high. Besides, the input voltage can range from 85~265V with high stability, low vibration and noise. APFC techniques can effectively reduce harmonic content, improve power factor to satisfy current standard.

The disadvantages of the topology is the high cost and complexity of controlling circuits. Besides, with higher switch frequency, the switched loss becomes significant. Thus, it is applied in low power occasion.

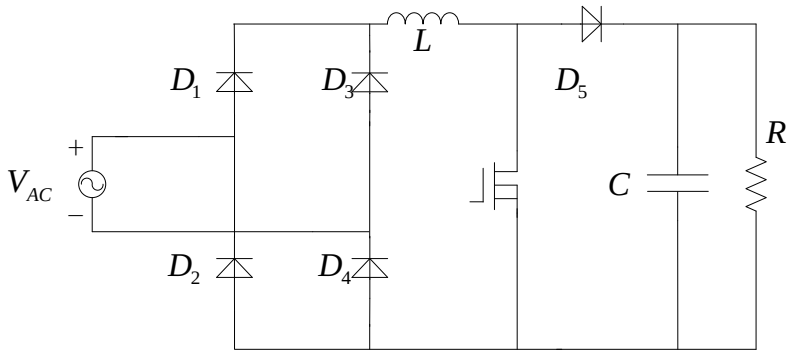


Figure 4. Conventional Active Power Factor Correction topology

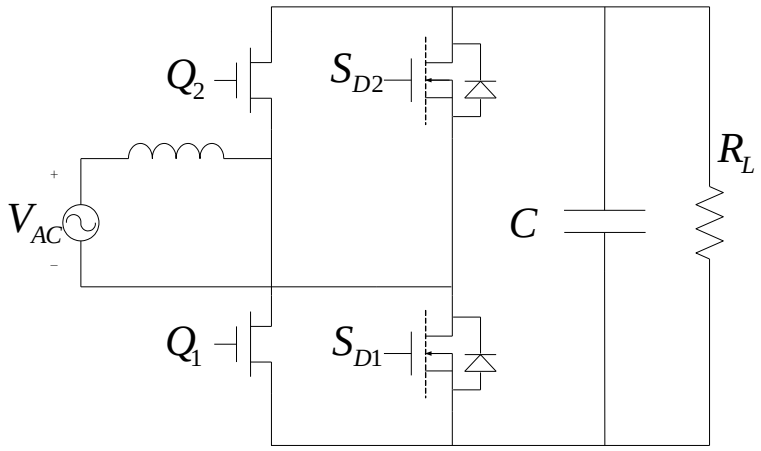


Figure 5. Totem-pole bridgeless Power Factor Correction topology

In conventional PFC topology, there are two bridge diode voltage drops and one drop at the boost stage, which set a limit on system efficiency, especially for low voltage and large current occasion. In this case, bridgeless PFC converter was put forward by eliminating the diode bridge. Totem-Pole bridgeless is one of them. *Figure 5* shows the topology of the basic totem-pole bridgeless PFC boost rectifier. It is shown that two GaN HEMT are used as boost converters. And two low resistance silicon MOSFETs are placed to eliminate diode drop to improve the efficiency.

The significantly smaller reverse recovery charge of GaN HEMT compared with silicon MOSFETs makes the totem-pole bridgeless PFC practical. It meets the demand for increasing power density of switched-mode power supplies by adopting high switching frequency as well as not to increase switching losses at pulse width modulated (PWM) converters. We choose the 1st generation GaN as the power device, whose reverse recovery charge is 20 times lower than Si counterpart .It's suitable for hard-switched bridge.

The Totem-Pole bridgeless PFC focus on minimizing power loss in the conduction path. As *Figure 5* shows, it consists of two fast GaN HEMT(Q1&Q2) operating at a pulse-width-modulation frequency(We use 80kHz)and a pair of slow but low resistance MOSFETs(S_{D1}&S_{D2}) operating at power frequency(45~60Hz). Thus the conduction path includes one fast switch and one slow switch without diode drop. The switch S_{D1}& S_{D2} are the synchronized rectifier. During positive AC cycle, S_{D1} is on while S_{D2} is off and vice versa.

Table 1 shows the difference between conventional PFC and Totem-pole bridgeless PFC. Two switches of Totem-pole bridgeless PFC plays the same role with fast recovery diode as conventional PFC.

Table 1. **Device comparison of conventional PFC and Totem-pole bridgeless PFC**

	Low speed diode	High speed diode	Switch device	Conduction path on/(off)
\Conventional PFC	4	1	1	2 low speed diode+ 1switch/(2 low speed diode+1 high speed diode)
Totem-Pole Bridgeless PFC	2(Si MOSFET)	0	2(GaN)	1 Si MOSFET+1 switch(GaN)

1.3. MC56F827xx controller advantages and features

The 56F827xx microcontroller is a member of the 32-bit 56800EX core-based Digital Signal Controllers (DSCs). Each device in the family combines, on a single chip, the processing power of a 32-bit DSP and the functionality of a microcontroller with a flexible set of peripherals. Due to its cost-effectiveness, configuration flexibility, and compact program code, 56F827xx is well-suited for many consumers and industrial applications.

The following list summarizes the superset of features across the entire 56F827xx family.

- 56800EX 32-bit DSC core
- Up to 50 MHz operation frequency in normal mode and 100 MHz in fast mode, up to 64 KB (32 KW) program flash memory, up to 8 KB (4 KW) dual port program/data RAM
- Protects supervisor programs and resources from user programs
- One 8-channel eFlexPWM module with NanoEdge™ placement and enhanced capture
- 2 separate 8-channel 12-bit cyclic ADC
- Single conversion time of 10 ADC clock cycles (10 × 100 ns = 1μs)
- Additional conversion time of 8 ADC clock cycles (8 × 100 ns = 800ns)
- One windowed watchdog timer, power Supervisor
- On-chip 8 MHz/400 kHz relaxation oscillator, 200 kHz Relaxation Oscillator and 4MHz to 16 MHz external Crystal Oscillator (XOSC)
- Inter-Module Crossbar with AND-OR-INVERT function
- Programmable Interrupt Controller (INTC) with 4 level priority
- One Quad Timer, two Periodic Interval Timers
- Two 12-bit DAC modules
- Four High Speed Comparators with integrated 6-bit DAC references
- 5 V tolerant I/O (except for RESETB pin which is a 3.3 V pin)

The switched-mode power supply applications benefit greatly from the flexible eFlexPWM module, fast ADC module, on-chip analog comparator module, and inter-module crossbar with AOI function.

This Pulse Width Modulation (PWM) module can generate various switching patterns, including highly sophisticated waveforms. It can be used to control all known motor types and is ideal for controlling different Switched Mode Power Supplies (SMPS) topologies as well, it has the following features:

- 16 bits of resolution for center, edge aligned, and asymmetrical PWMs
- Fractional delay for enhanced resolution of the PWM period and edge placement
- PWM outputs that can operate as complementary pairs or independent channels
- Support for synchronization to external hardware or other PWM, half cycle reload capability
- Multiple output trigger events can be generated per PWM cycle via hardware
- Fault inputs can be assigned to control multiple PWM outputs, programmable filters for fault inputs
- Independently programmable PWM output polarity, independent top and bottom dead time insertion

- Each complementary pair can operate with its own PWM frequency and dead time values
- All outputs can be programmed to change simultaneously via a FORCE_OUT event

The eFlexPWM offers flexibility in its configuration, enabling efficient control of any SMPS topology. The eFlexPWM module is capable of free control of rising and falling edges for each PWM output and includes automatic complementary signal generation and dead time insertion. Due to NanoEdge placement the eFlexPWM can generate duty cycles and frequencies with high a resolution of up to

390 s.

The eFlexPWM module can also generate up to 6 synchronization events per sub-module to provide synchronization with other modules (ADC, Quad-timer).

This Totem-pole bridgeless PFC application uses the eFlexPWM module for generating two PWM signals for primary side power MOSFETs and two PWM signals for secondary side synchronous rectifier, and provide the trigger signal for ADC sample.

ADC features include:

- Two independent 12-bit analog-to-digital converters (ADCs):
- 2 x 8-channel external inputs
- Built-in x1, x2, x4 programmable gain pre-amplifier
- Maximum ADC clock frequency up to 10 MHz, having period as low as 100 ns
- Single conversion time of 10 ADC clock cycles (10 × 100 ns = 1 μs)
- Additional conversion time of 8 ADC clock cycles (8 × 100 ns = 800 ns)
- Support of analog inputs for single-ended and differential, including unipolar differential, conversions
- Sequential, parallel, and independent scan mode
- All 16 samples have offset, limit and zero-crossing calculation supported
- ADC conversions can be synchronized by any module connected to the internal crossbar module, such as PWM, timer, GPIO, and comparator modules.
- Support for simultaneous triggering and software-triggering conversions
- Support for a multi-triggering mode with a programmable number of conversions on each trigger
- Each ADC has ability to scan and store up to 8 conversion results.

This Totem- pole PFC application uses the ADC module in triggered parallel mode, ADC samples are triggered at the center of PWM signal. ADCA sampled the input voltage and output bus voltage, ADCB sampled input current. The Inter-Module Crossbar and AND-OR-INVERT logic features:

- Provides generalized connections between and among on-chip peripherals: ADCs, 12-bit DAC, comparators, quad-timers, eFlexPWMs, EWM, and select I/O pins
- User-defined input/output pins for all modules connected to the crossbar
- DMA request and interrupt generation from the crossbar
- Write-once protection for all registers
- AND-OR-INVERT function provides a universal Boolean function generator that uses a four-term sum-of-products expression, with each product term containing true or complement values of the four selected inputs (A, B, C, D).

This Totem-pole bridgeless PFC converter application uses the Inter-Module Crossbar and AND-OR-INVERT logic to provide interconnection between the eFlexPWM module and ADC module, interconnection between fault signal and on-chip comparator, generate the synchronization rectifier PWM by multiple signals from eFlexPWM and on-chip comparator.

The application also uses other peripherals like on-chip comparator for hardware protection and 2xSCI module for communication with the primary side and remote control via PC and several GPIOs for LED indication.

2. System description

2.1. Structure

The application system incorporates power stage, control board and isolated Universal Asynchronous Receiver／Transmitter (UART) port to communicate with other devices, such as backward stage LLC resonant converter.

The control module involves voltage/current sensing circuitry, drivers, PM Bus communication and the DSC controller board.

The auxiliary power supply takes power directly from the DC Bus, then generates the desired voltages with Flyback converter.

The sensing circuitries are used for sensing input voltage, input current, DC bus voltage and accommodates them to the DSC acceptable voltage level.

The drivers are used for both amplification of DSC PWM signals and synchronous rectifier's MOSFET.

The synchronous rectifiers are used to reduce the conduction losses in replace of conducting diodes.

DSC MC56F82748 controller is placed on the control daughter card and connected to power board via PCI slot. The control card is powered from power board side and it works as the master for the whole application.

The controller is also used to communicate with other equipment and the boards. One UART is applied to communicate with backward LLC stage, another UART to USB conversion is applied to communicate with host PC for FreeMASTER or firmware updating, one IIC reserved for PM Bus network.

The overall system structure is shown in *Figure 6*

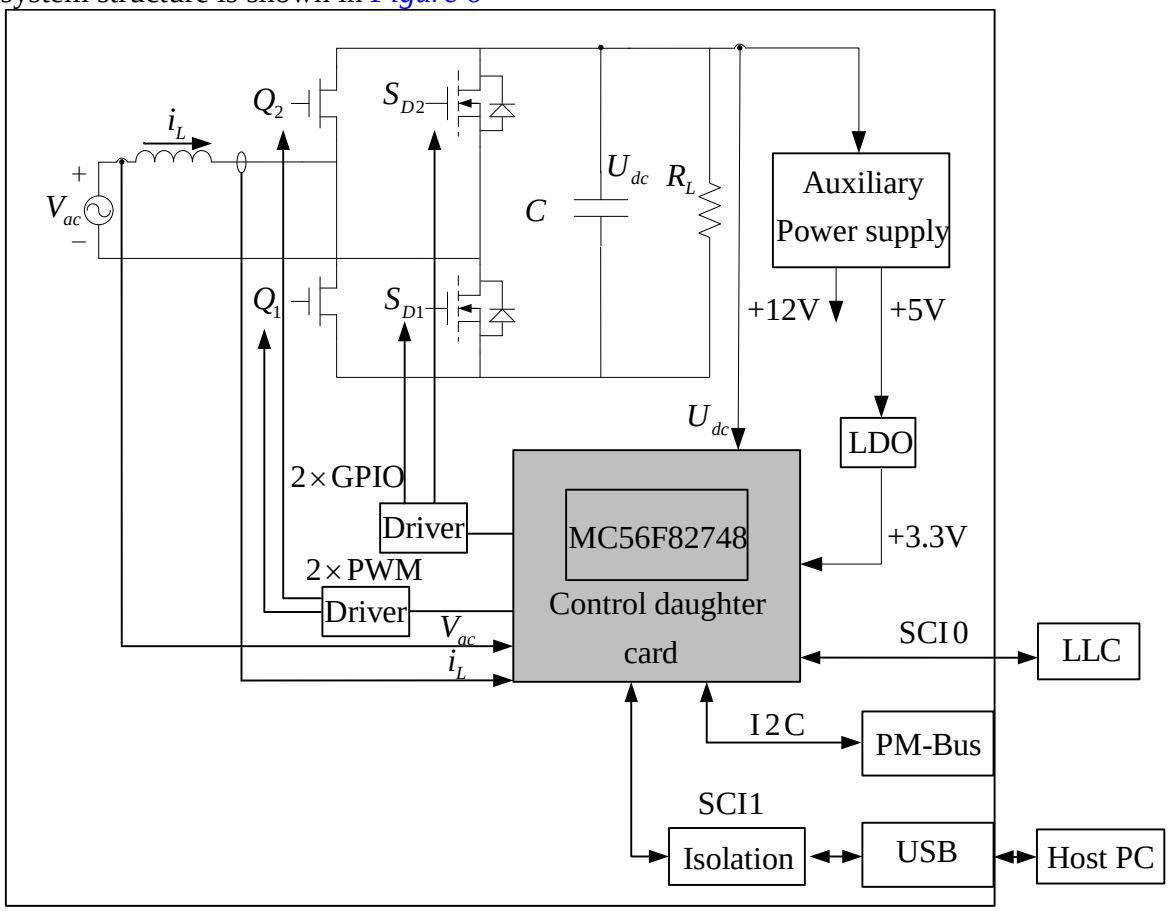
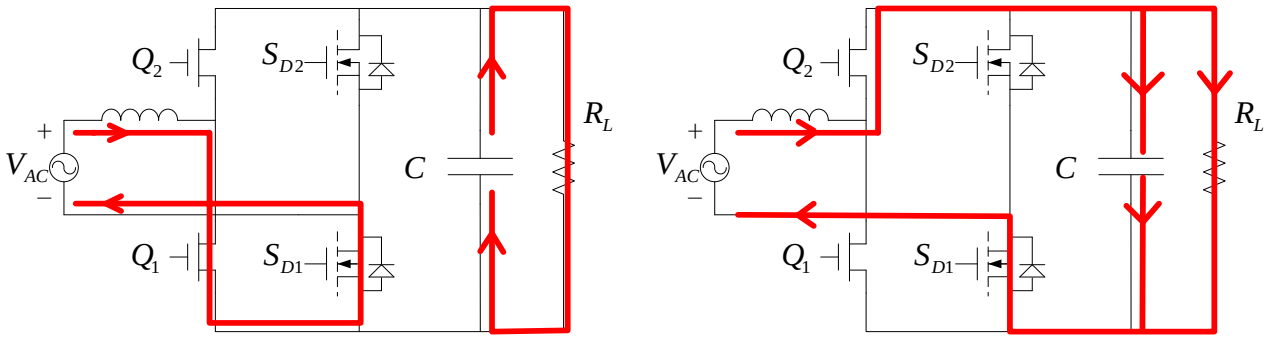


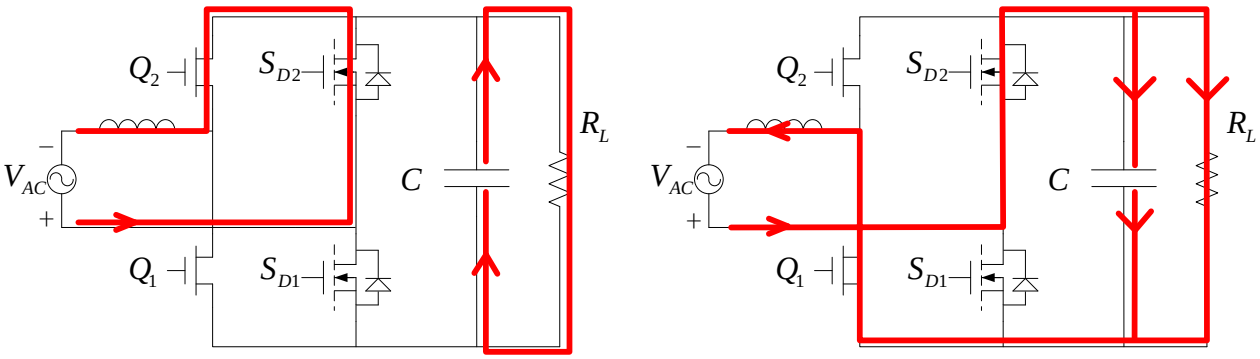
Figure 6. System block diagram

2.2. Control logic



a) Positive cycle charging mode

b) Positive cycle after-flow mode



c) Negative cycle charging mode

d) Negative cycle after-flow mode

Figure 7. Totem-pole bridgeless PFC simplified illustration

The totem-pole bridgeless PFC consists of a pair of GaN HEMT switches(Q1&Q2) that operates at PWM frequency. They work as a pair of boost converters within a half cycle. In either cycle, one GaN HEMT will act as a master switch to increase the boost inductor L's current and the other transistor will act as a slave switch to force the current flow to the DC output meanwhile releasing energy. The roles of the pair interchange when AC polarity changes. And dead time is added to avoid the short of the bridge for GaN and Silicon MOSFET.

2.3. Working mode

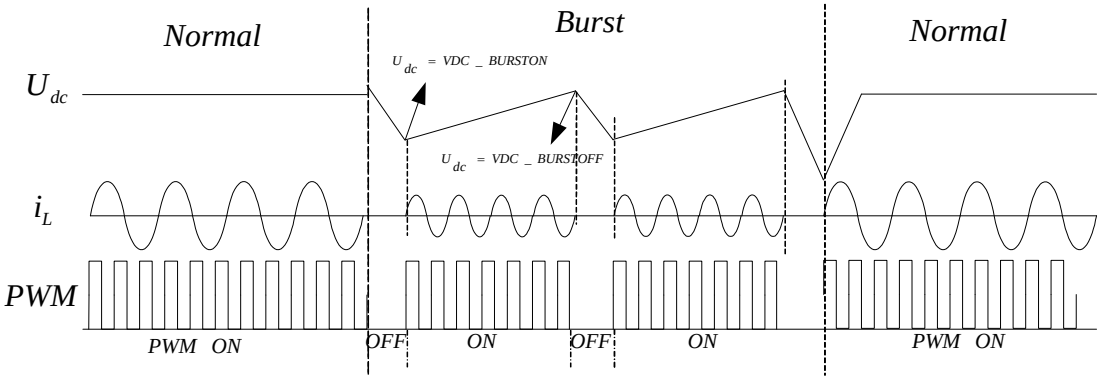


Figure 8. Illustration of normal mode and burst mode

The constant frequency controller is hard to ensure good input current shape in DCM at light load due to the nonlinear characteristics of the converter especially for GaN synchronous rectification. The poor input current shape in DCM will cause high current distortion and large RMS current drawn, which results in poor power factor. Besides, at light load, the load independent constant losses, which consists of gate charge loss, parasitic capacitance loss in MOSFET and core loss in inductor, become dominant and will cause very poor efficiency.

In this application, the control will ensure high efficiency by tuning on and off the PFC to regulate the output voltage within a hysteresis bandwidth.

The PFC is turned off when the load is very light. In this application, when the input current reference is equaled to 0.25 A for a certain time which is the low limit of current reference, PFC will enter burst

mode as shown in Fig.8. PFC is turned off and output voltage decreases. When output voltage reaches VDC_BURSTOFF, PFC is turned on again with constant reference current 0.25A. If the output voltage drops quickly to a lower threshold, PFC will enter back to normal mode again.

2.4. Control loop introduction

Based on the MC56F82748, a digital PFC rectifier is implemented as shown in *Figure 9.*

Power Factor (PF) is defined as the ratio between AC input's real power and apparent power. Assuming input voltage is a perfect sine wave, PF can be defined as the product of current distortion and phase shift. The PFC control loop's tasks are:

• **Controls inductor current**, which makes the current sinusoidal and maintains the same phase as the input voltage

• **Controls output voltage,** which makes the output voltage equaled to target value

2.4.1. Arithmetic of Current Reference

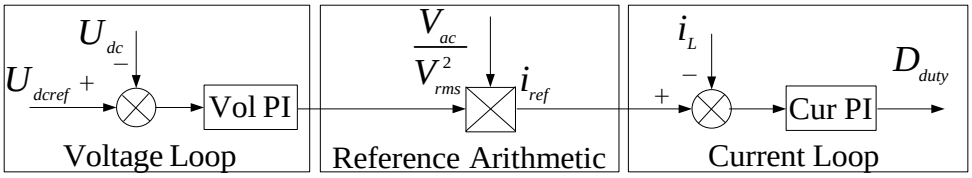


Figure 9. Totem-pole bridgeless PFC control scheme

According to PFC theory, PFC arithmetic can be divided into three parts.

- Voltage outer loop, which insures the output voltage follows the reference-constant voltage output
- Reference arithmetic, which insures that the current reference follows the sine reference
- Current inner loop, which insures the input current follows the given current reference

In analog arithmetic, the input voltage sample is introduced as the input current's reference so the ripple voltage is introduced to current control at the same time. In this application, we use

$$i_{ref} = \frac{V_{ac} \cdot V_c}{V_{rms}^2} \qquad Eqn.1$$

Where:

V_{ac} is the input voltage involved to ensure the current wave follows the input voltage wave

V_c is the output of voltage regulator

V_{rms} is the RMS value of input voltage

2.4.2. Voltage and Current Loop Design

PI loop control is widely used and classical in industry control. In this application, the voltage and current loops adopt PI regulator arithmetic. Figure 10 represents the small signal model block diagram of boost converter. The definition of variables will be represented blow.

Where:

R_m is the equivalent resistor of the equivalent multiplier/divider output current.

g_c is the gain between equivalent multiplier/divider output current and the output of voltage regulator.

The subscript 'p' denotes the small-signal perturbations.

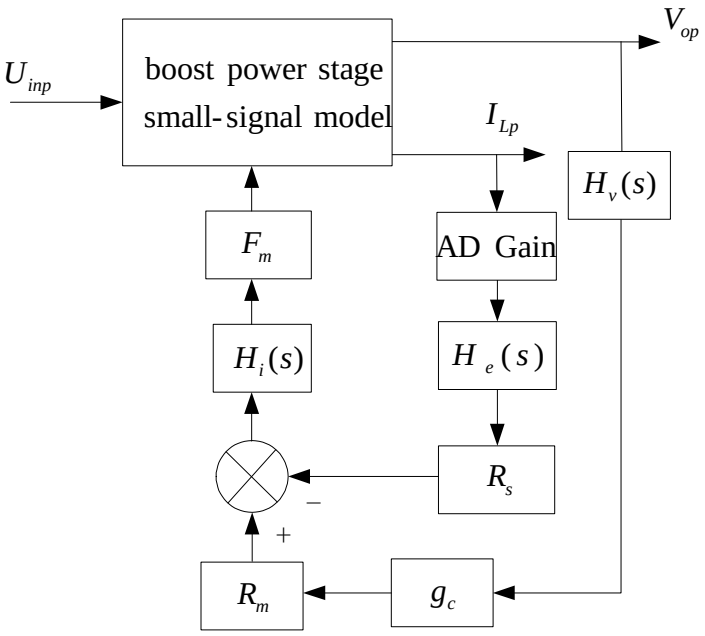


Figure 10. Small-signal model block diagram of the boost converter

2.4.2.1. Current loop control design

Figure 11 represents the control block diagram of current loop.

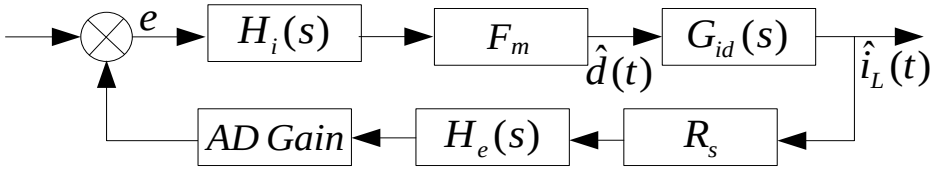


Figure 11. Control block diagram of current loop

In this control diagram, H_i(s) is the current loop controller, which is designed to maintain the stability of current control loop; G_{id}(s) is the duty-to-current transfer function; F_m is the pulse-width ratio. In this

application, F_m = 1/625; R_s is the equivalent sample resistor for unit current; H_e(s) is a second-order filer for current sample; *AD Gain* = 2^{15}/3.3;

The transfer function of open current loop excluding H_i(s) is:

$$T_i(s) = F_m \times G_{id}(s) \times R_s \times H_e(s) \qquad Eqn.2$$

Figure 12 shows the bode plot of current loop excluding H_i(s). From the bode plot, the bandwidth is only 34Hz.

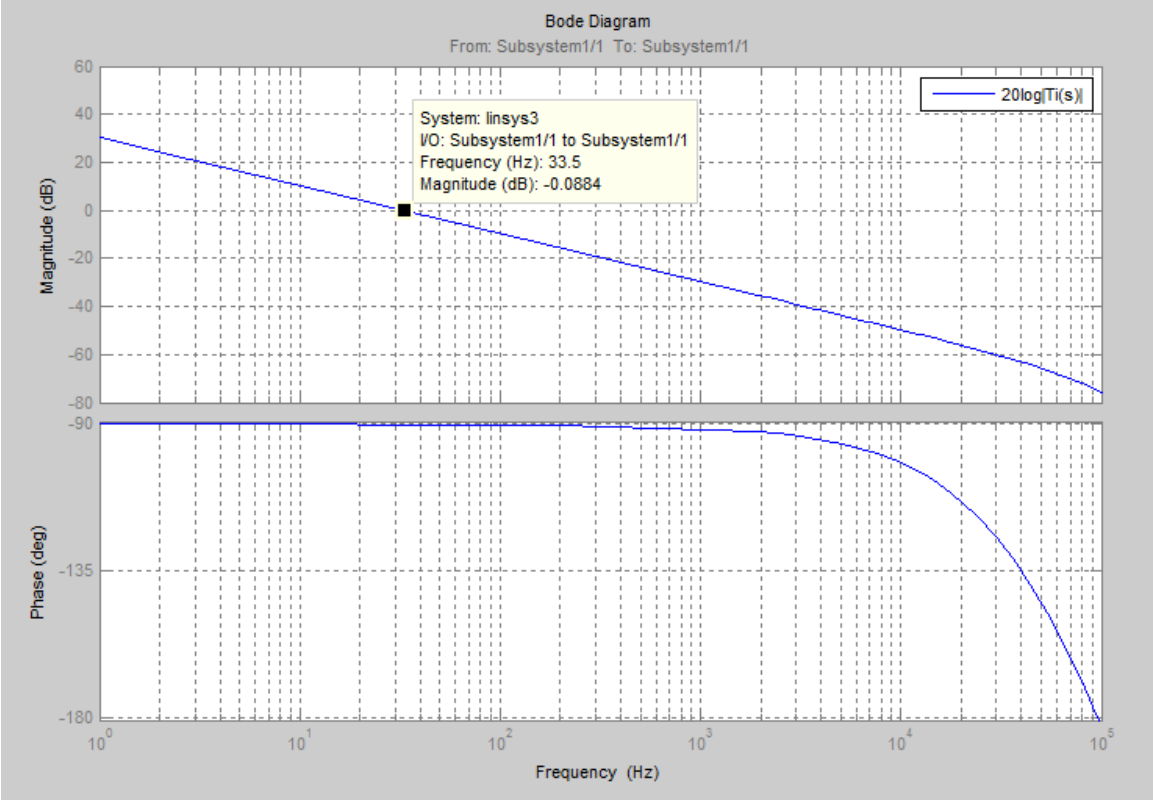


Figure 12. Bode plot of current loop excluding controller H_i(s) with AD gain

To have good current loop performance, the bandwidth needs to be within 3~10 kHz, and phase margin needs to be over 40 degrees. In this application, we set a zero in 3.5 kHz and set the gain to be 0.022.

$$H_i(s) = K_p + \frac{K_i}{s} = \frac{K_p(s + 2 \cdot \pi \cdot 3500)}{s} \qquad Eqn.3$$

Therefore, we have:

$$\begin{cases} Kp = 0.022 \\ Ki = 480 \end{cases} \qquad Eqn.4$$

The PI algorithm in the continuous time domain is expressed as:

$$y(t) = K_p \cdot e(t) + \int_0^t K_i e(t)dt = K_p \cdot e(t) + y_I(t) \qquad Eqn.5$$

The integral term above can be rewritten into discrete time domain by Backward Euler transformation:

$$y_I(n) = y_I(n-1) + K_i \cdot T_s \cdot e(n) \qquad Eqn.6$$

The discrete time domain representation of PI algorithms is:

$$y(n) = K_p \cdot e(n) + y_I(n-1) + K_i \cdot T_s \cdot e(n) \qquad Eqn.7$$

Therefore, we can get:

$$\begin{cases} K_{pz} = K_p = 0.022 \\ K_{iz} = K_i \cdot T_s = 0.006 \end{cases} \qquad Eqn.8$$

Figure 13 shows the bode plot of current loop including H_i(s). From the bode plot, the bandwidth is around 8 kHz. And the phase margin is over 55 degrees. Both of them meet our requirements.

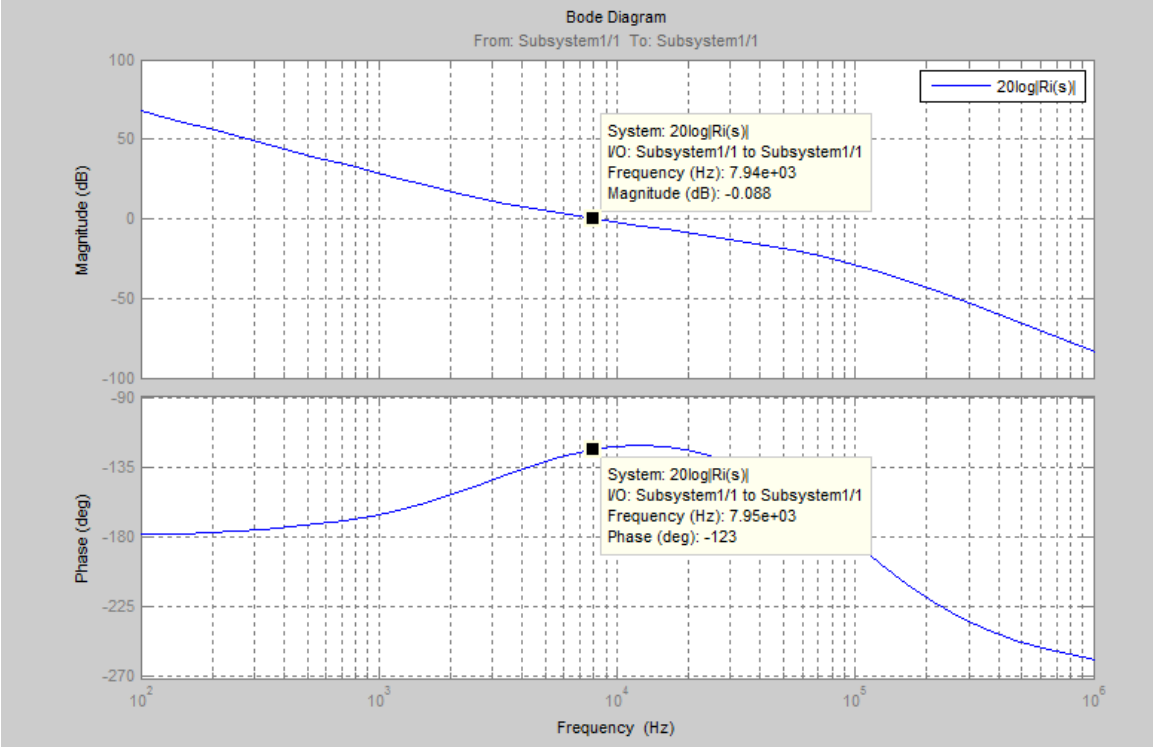


Figure 13. Bode plot of current loop including controller H_i(s)

2.4.2.2. Voltage loop control design

The control-to-output voltage transfer function is given through Figure 10:

$$\frac{v_{op}}{v_{cp}} = \frac{F_m g_c R_m H_i(s) G_{vd}}{1 + T_i} \qquad Eqn.9$$

Where

G_{vd} is the duty-to-output transfer function; In this application, $g_c = 1/V_{rms}$;

For resistive load, the following approximations are valid:

$$\frac{v_{op}}{v_{cp}} = g_c \frac{R_m V_{rms} R/2}{R_s V_o sC} \qquad \textit{Eqn.}10$$

To have good voltage loop performance, the bandwidth needs to be within 2~10 Hz in order to reduce the effect of input voltage ripple towards inductor current regulation. In this application, we set a zero in 12 Hz. Then, the whole open loop can be defined as:

$$R_v(s) = g_c \frac{R_m V_{rms} R/2}{R_s V_o sC} H_v(s) \qquad \textit{Eqn.}11$$

Where H_v(s) is the voltage loop controller.

Figure 14 shows the bode plot of voltage loop. It can be seen that voltage loop bandwidth is about 9Hz.

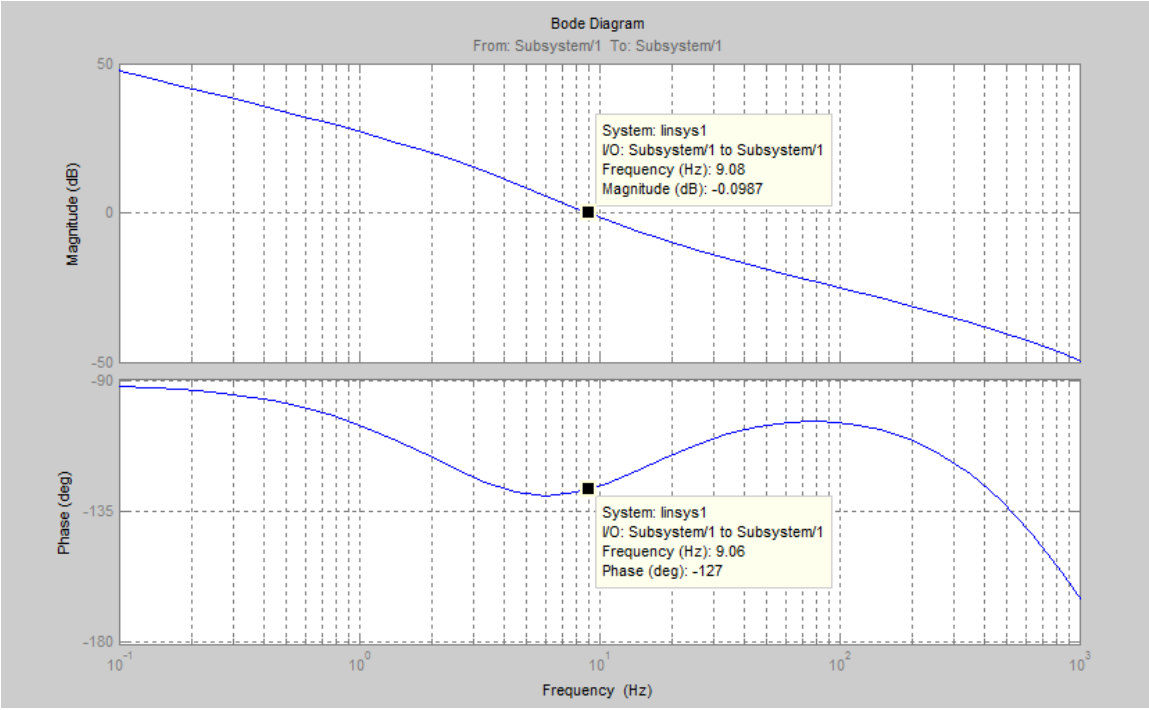


Figure 14. Bode plot of voltage loop including controller H_v(s)

3. Hardware design

This chapter describes the totem-pole bridgeless PFC hardware design and provides the layout consideration.

3.1. Specification

Table 2. **Totem-pole bridgeless PFC specification**

Parameter	Minimum	110Vac input	220Vac input	Maximum
Input voltage(RMS)	85V			265V
Input current(RMS)	—	2.7A	2.7A	—
Output voltage	—	380V	380V	—
Output current	—	0.789A	1.579A	—
Output power	—	300W	600W	—
iTHD	—	<5%	<5%	5%
Power Factor	0.95	>0.95	>0.95	—
Efficiency	—	95%	98%	—

3.2. PFC choke design

Calculate the PFC choke according to the choke current ripple, and we set ΔIlmax=35%Ilpk, and below is the calculation process.

Set the Vinmin 85Vac with 350W load, and will calculate the choke inductance under this condition.

$$I_{Lrms} = \frac{P_{in}}{V_{inrms}} = \frac{P_{out}}{\eta V_{inrms}} = 4.334A \qquad Eqn.12$$

$$I_{Lpk} = \sqrt{2} \times I_{Lrms} = 6.130A \qquad Eqn.13$$

The operation frequency is 80 kHz， t=12.5us；

Sendust core: CS400060 core is selected to make the choke.

$$L_{\min} = \frac{V_{out}}{4 \times f_{SW} \times \Delta I_{L\max}} = \frac{380}{4 \times 80k \times 6.130 \times 0.35} = 550uH \qquad Eqn.14$$

$$N = \sqrt{\frac{L_{\min}}{A_l}} = 80 \qquad Eqn.15$$

So the below is the detail information for the PFC choke:

1. Core: Type-Toroid

Size:2×39.90OD×24.10ID×14.5H
Material: 60
Magnetics, Sendust core: **CS400060**
KDM, **KS157-060A**

2. Construction: 80 Turns(reference), D1.02mm×2 Cu wire

3. Inductance: 600uH±10% [100 kHz 0.3V]

4. Schematic: Rated Current: 10A

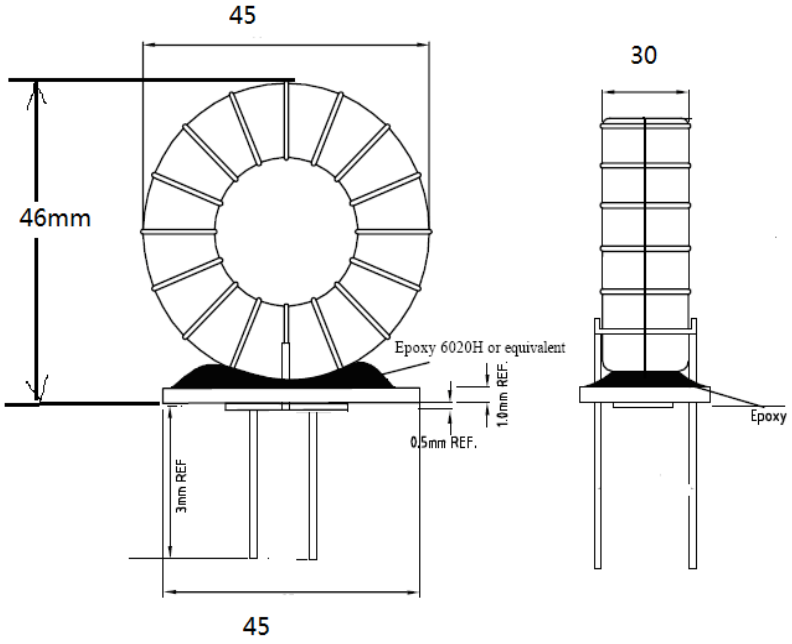


Figure 15. Dimension of inductance coil

3.3. Power circuits and drives

Figure 16 shows the power circuits for the EMI circuits, inrush limitation circuits, GaN MOSFET bridge and Si MOSEFT bridge. C31A, C31, C32 is used to absorb the voltage spike of GaN. And these ceramic capacitors should be placed as close as possible to GaN.

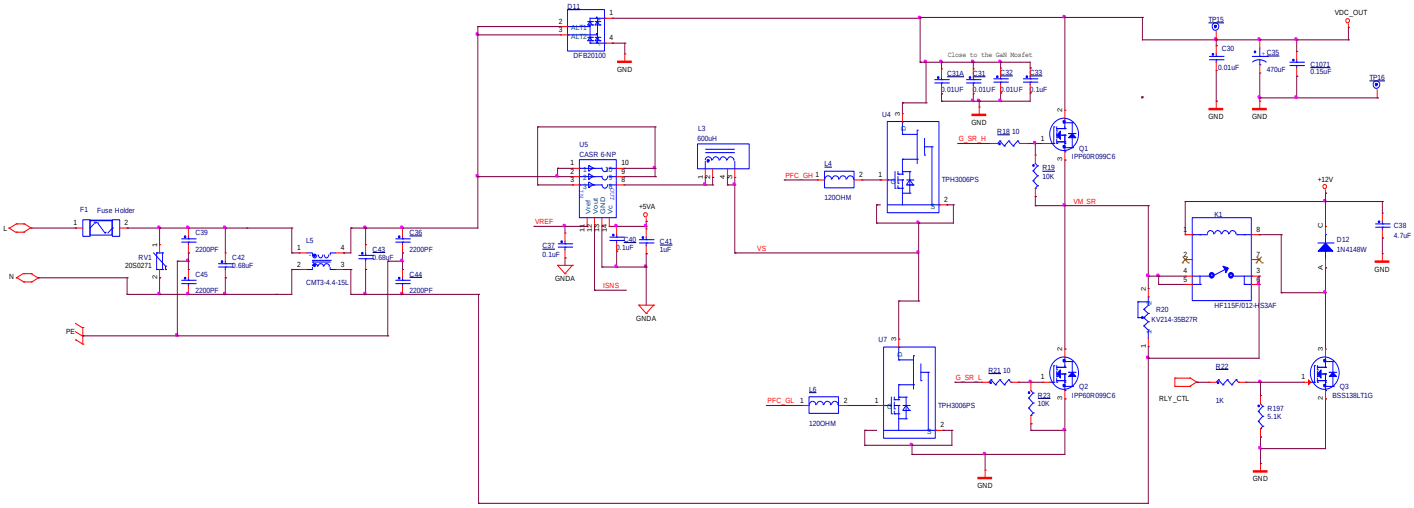


Figure 16. . Main power circuits

R20 and K1 are for the inrush limitation circuits; 27R power resistor is selected to prevent the high pulse current when the system is powered on.

GaN MOSFET TPH3006PS is chosen as main switch component. For the high recovery speed(30 ns) of body diode, it is suitable to act as the BOOST PFC free wheel diode, and brings high efficiency.

The common Si MOSFET IPP60R099C6(Q1, Q2) is selected to realize the grid power frequency synchronization function, which will help to improve the system efficiency more, and will replace the low speed rectifier diode.

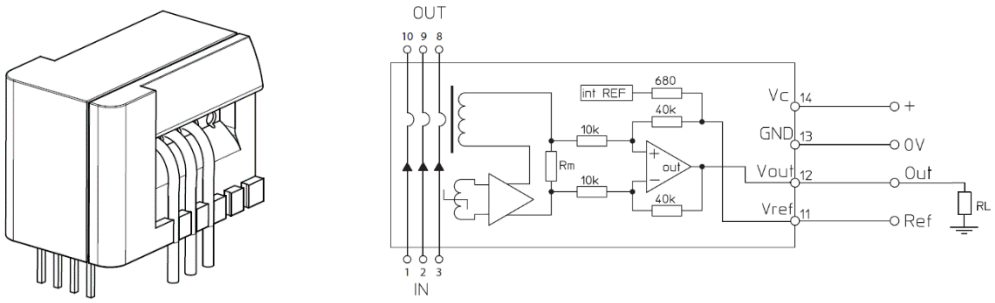
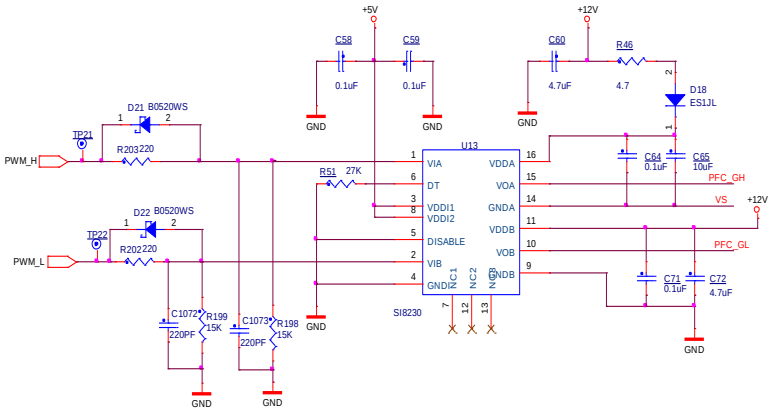
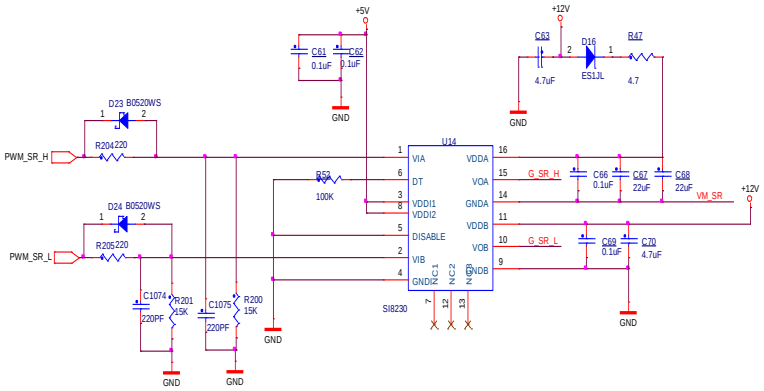


Figure 17. Application illustrations

CASR 6-NP current transducer is for the choke current sensing, and we set the primary side power circuits in series connection mode to increase the gain between the primary to secondary, which will enhance the choke current signal's amplitude.



a) GaN MOSFET drive circuit



b) SR MOSFET drive circuit

Figure 18. . MOSFETs' drive circuits

Because of the critical driver requirements for the GaN MOSFETs, the isolated and individual driver IC SI8230 is selected to drive the GaN MOSFET and Si MOSFET. A small RC filter is placed between the DSC and driver IC to decrease the noise, otherwise the disturbance signal may cause the mistake pulse on the driver IC, and one diode is paralleled with the resistor to let the PWM change from high to low level in time. And the bridge high side and low side MOSFETs are driven with the individual GND, this will help to decrease the disturbance between them. The IC's drive current is 0.5A, and this can meet the requirement.

3.4. Analog signal sensing circuits and Auxiliary power

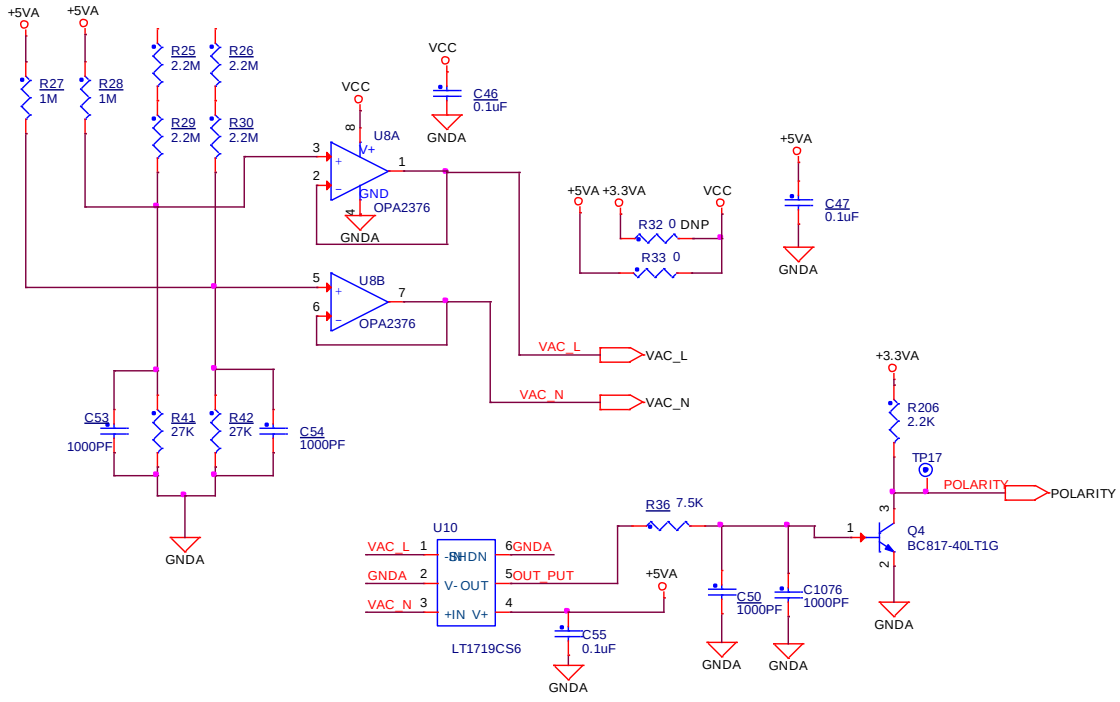


Figure 19. Vac and Polarity detection circuits

Figure 19 shows the Vac divider circuits and the Vac polarity circuits. The Vac_L and Vac_N is connected to the DSC ADC pin, and this signal will be used to convert the Vac amplitude, and will be used for the system control loop, such as the current reference, and Vrms calculation.

The comparator LT1719 is used for the Vac polarity detection, and one RC filter is added for the noise decrease, and one transistor of Q4 is used to make the signal more clean . The Q4 should be placed closely to DSC.

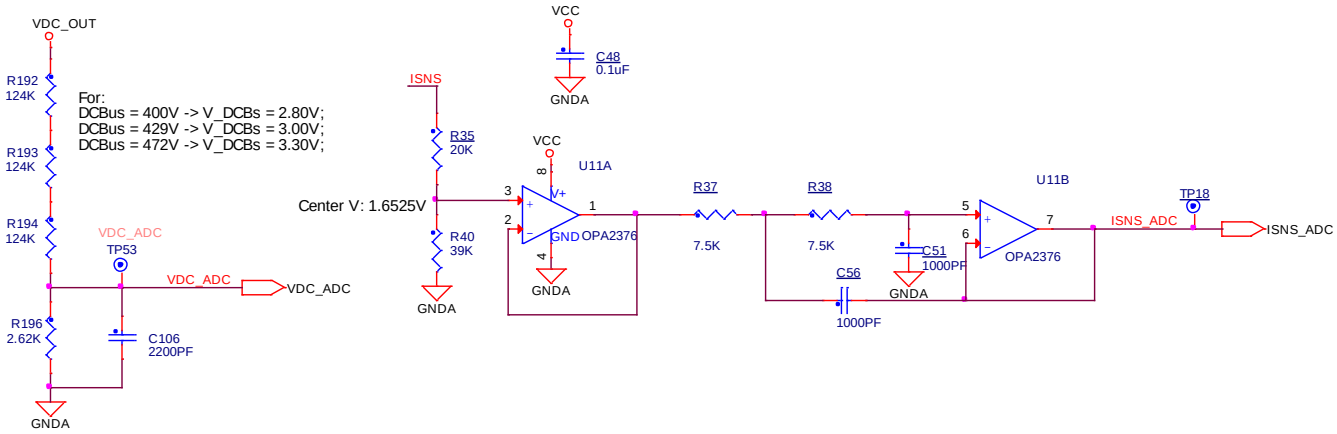


Figure 20. Bus voltage and choke current filter circuits

We use the common resistor divider circuits to detect the high voltage signal Vbus. And as the DSC ADC input port is with high resistance, amplifier is not needed on this voltage circuits. However the R192, R193, R194 should be power resistors. Package can be 1206 or similar.

For the choke current analog circuits, one low pass filter is used to deal with the noise and ripple. The filter bandwidth is 13.5 kHz, and this will affect the system PF value and THDi, so when adjust the PFC control parameter, please pay attention to this filter's parameter too, and find the best performance.

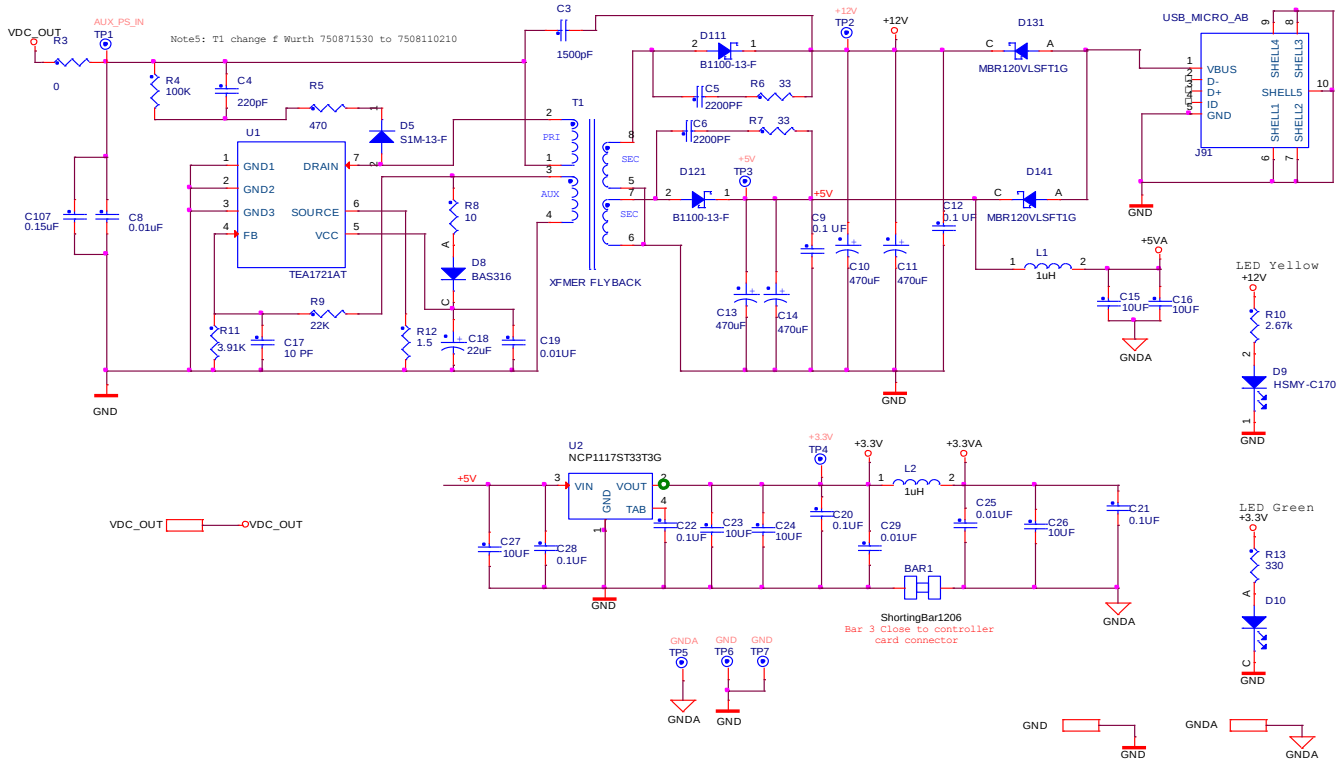


Figure 21. Auxiliary power circuits

One Flyback converter generates the auxiliary power, include the driver voltage 12V, driver chip power 5V, and the digital Vcc 3.3V. NXP TEA1721 chip is selected to control the converter, and this design can operation normally under 85Vac to 265Vac.

3.5. Communication with backward LLC

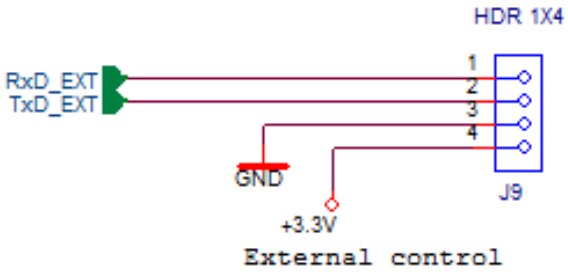


Figure 22. Main board placement strategy

Figure 22 shows the SCI connection for totem-pole PFC.

Pin 1 is the SCI RxD pin.

Pin 2 is the SCI TxD pin.

Pin 3 connects to ground.

Pin 4 connects to 3.3V.

3.6. PCB layout considerations

In totem-pole bridgeless PFC reference design, the input voltage and Vdc output are both high voltage, and system is working in high frequency mode 80 kHz with high dv/dt and di/dt. Thus the PCB layout need to be well considered, especially for following points:

- The input and output current loop routing;
- Power ground copper pouring to avoid influence by high frequency and high current signal;
- Heat dissipation for power components;
- The absorb circuits placement;
- Distance consideration for high voltage circuits;
- Routing method for the Sensitive signals

Figure 23 shows the placement for the totem-pole bridgeless PFC. The power loop should be as small as possible. Make sure the output current flow through the filter circuits first before it arrived at the output port. And the control circuits should be isolated from the power components, otherwise those signal will be disturbed by the high frequency switching power signal.

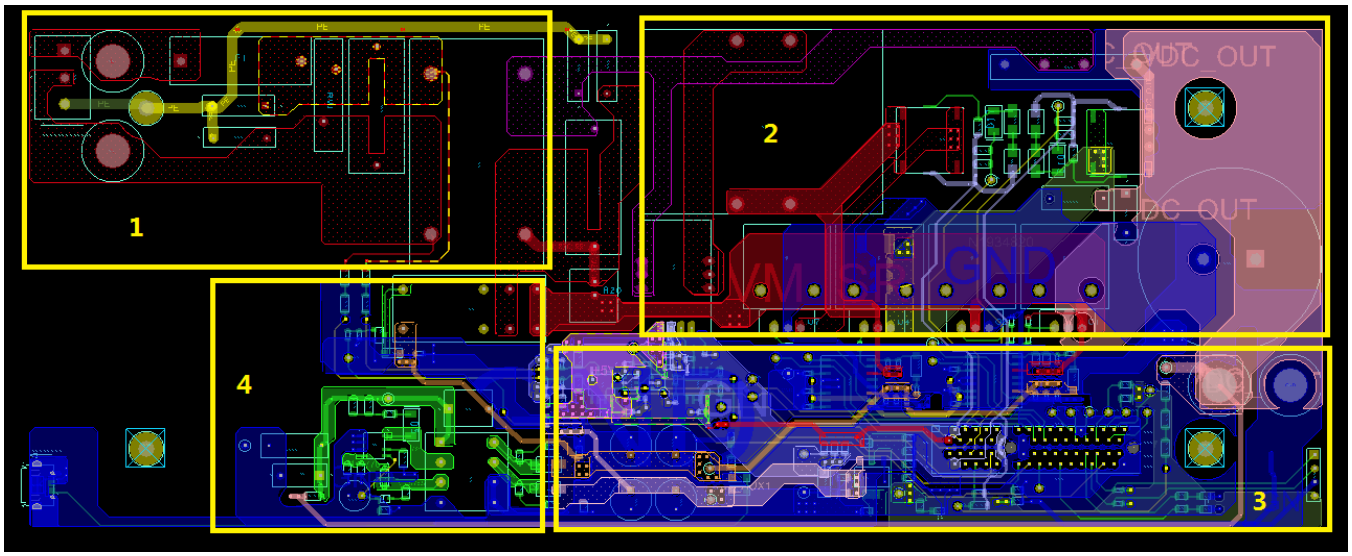


Figure 23. . Main board placement strategy

1 is the EMC filter circuits;

2 is the power circuits;

3 is the signal and DSC control board circuits;

4 is the auxiliary power circuits;

Except the placement, some other sensitive circuits should be taken into consideration.

1, GaN MOSFET driver circuit trace, the driver chip should be placed as closely as possible to the GaN MOSFET, and the drive signal to G pin and the low side S pin loop should be as small as possible, figure 24 shows the routing method.

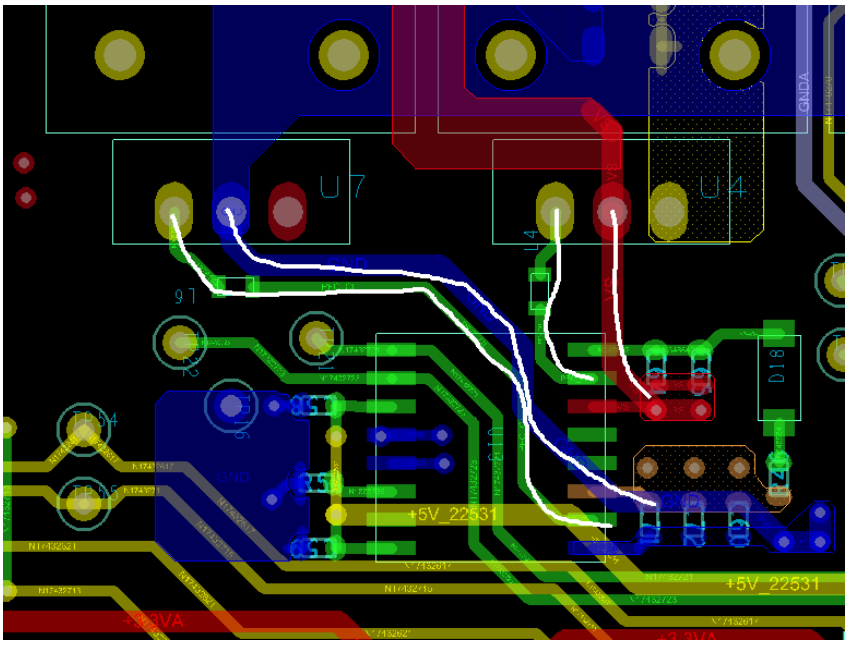


Figure 24. GaN MOSFET driver circuit trace

2, High voltage ceramic capacitor absorb circuits for the high speed GaN MOSFET bridge, C30,C31, C32,C33 should be placed very closely to the high side GaN MOSFET U4, and the capacitor GND should be as close as possible to the U7 S pin.

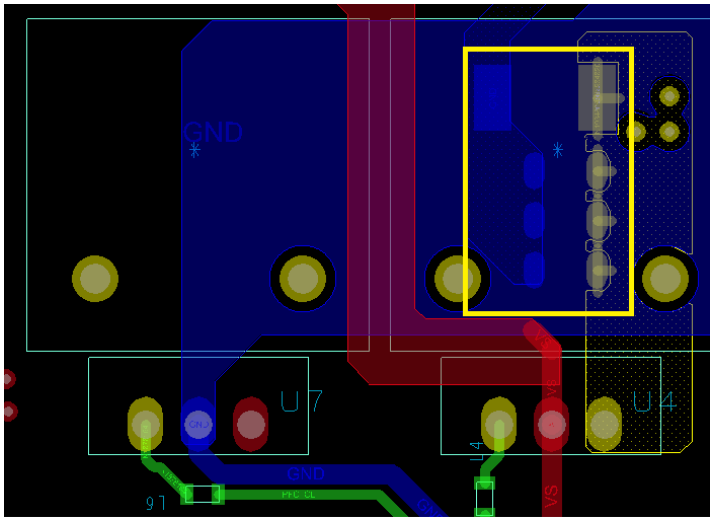


Figure 25. GaN MOSFET absorb circuit

3, Because this is the power system, the GND strategy is very important. We should separate the power GND, control GND and the analog AGND, the power GND and control GND should be connected together at the Vbus E-cap pin. Figure 26 shows the GND routing method.

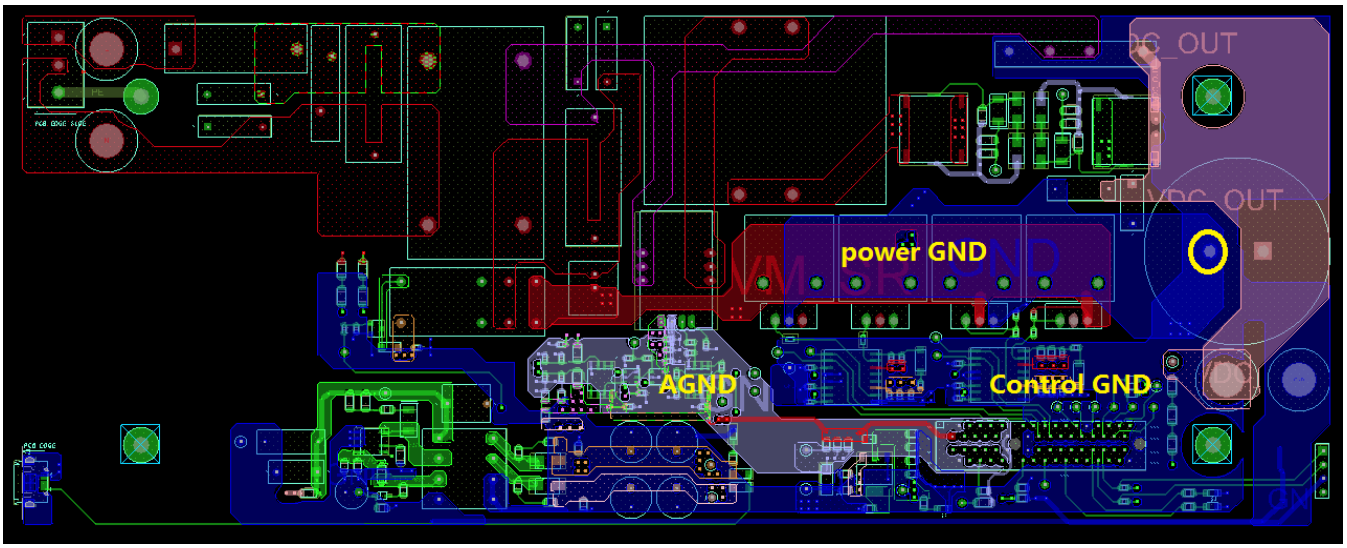


Figure 26. Totem-pole bridgeless PFC GND routing strategy

4. Software design

This application core is the MC56F82748 digital signal controller. This low-cost DSC has enough peripherals and features suitable for implementation of full digital control of totem-pole bridgeless PFC.

The software is written in C language using Code Warrior10.6, and calls the embedded software library (FSLESL) for time saving. For more information about how to use these libraries in the Code Warrior project, refer to the documentation Inclusion of DSC Freescale Embedded Software Libraries in Code Warrior 10.2 (AN4586).

This section describes the design of the software blocks, including software structure, configuration of the DSC peripherals, control timing, implementation of codes run in ram and boot loader.

4.1. Parameter normalization

In order to make full use of DSC resources, the application related physical quantities are all normalized to fixed-point decimal format, namely Q data format.

The relationship between actual value of the physical quantity and its normalized value is shown as following:

$$Frac\ value = \frac{actual\ value}{quantization\ range} \qquad Eqn.16$$

Where

- Frac value = the normalized value of the physical quantity
- Actual value = the actual value of the physical quantity expressed in units
- Quantization range = the maximum measurable value of this physical quantity

4.2. State machine

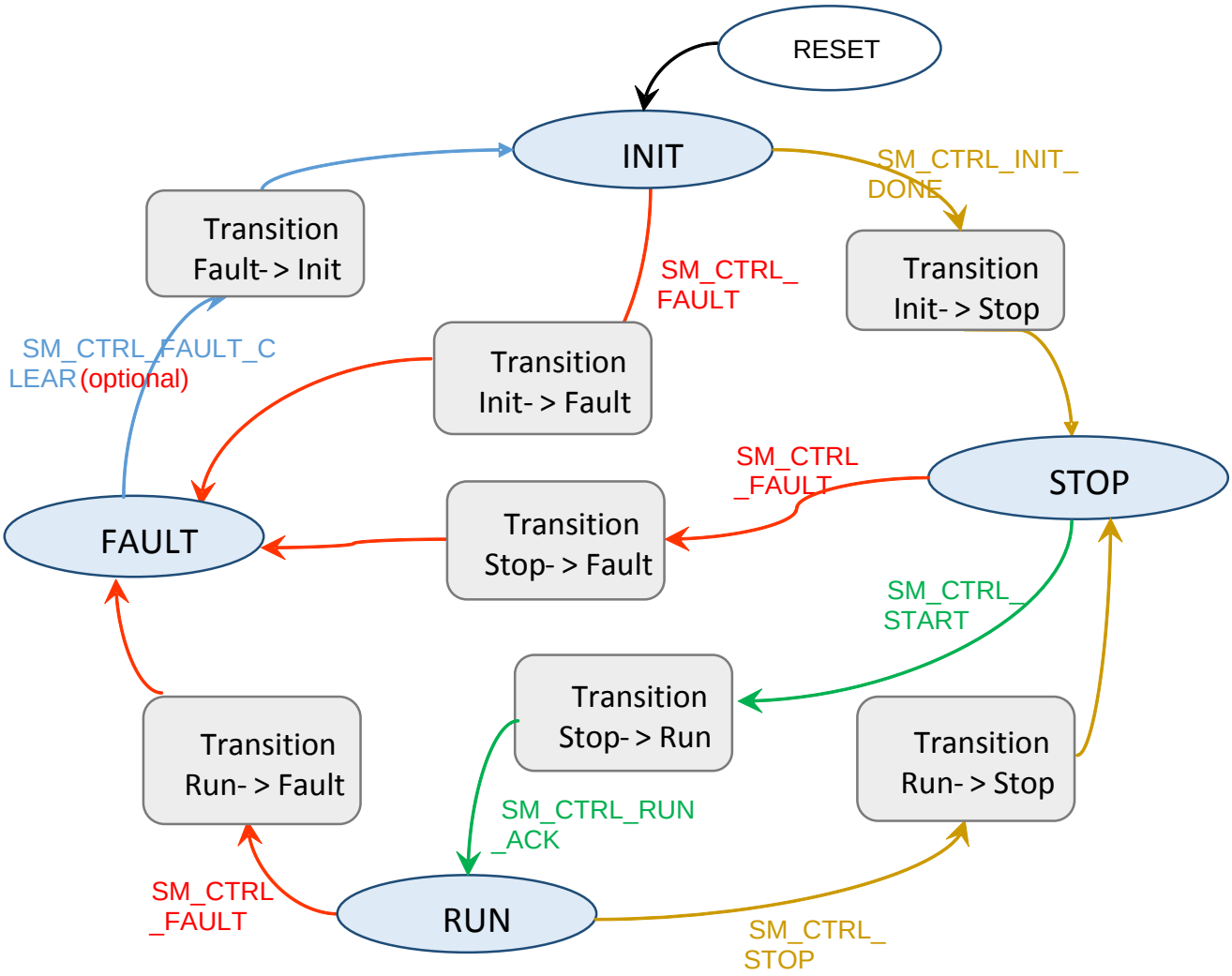
This application uses the state machine to control the system flow, seen as *Figure* 27. After reset the DSC configures all peripherals, the system enters a never ending loop including the application state machine. The application state machine includes four states:

- INIT: variable initialization.
- STOP: system is initialized and checking for the start conditions; PWM output is disabled.
- RUN: system is running; run sub-state is called; PWM output is enabled or disabled according to the load and can be stopped by Stop command.
- FAULT: system faced a fault condition, PWM output is disabled.

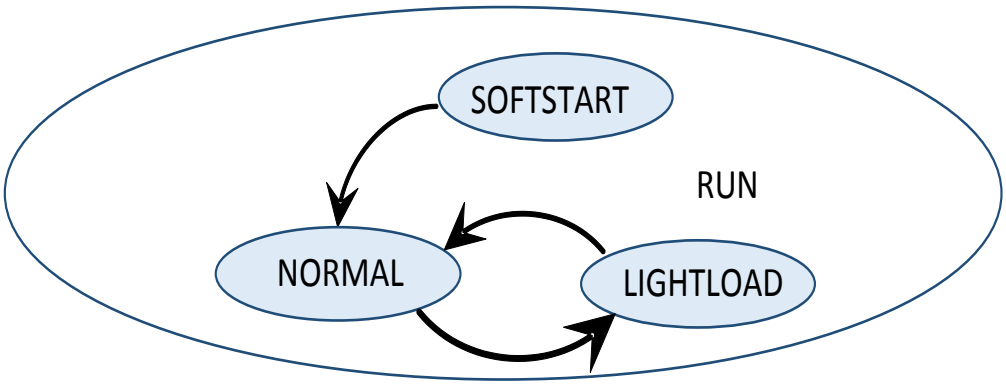
After the parameters initialization, the application state machine continues into the STOP state.

The software regularly checks the “PFC_run” instruction to decide whether to start up. When the command “PFC_run” is set, the input voltage RMS is detected to see whether it satisfies the requirements. If it is, after the output voltage reaching stable value with relay closed, the application state machine continues into the RUN state. In this state, the controller starts to take effect to achieve given output voltage and current according to load. Besides, the “PFC_run” instruction is also regularly checked in RUN state. When the command “PFC_run” is cleared, the application state machine goes back to STOP state and waits for the “PFC_run” command to be set again.

Under each state, if any fault is detected, the application state machine enters FAULT state. In this state, fault detection is still executed and whether to restart the application is optional. If allowed, when the fault condition is certainly cleared, the application state machine restarts from INIT state. If not, the software stays in a never ending loop and output of the converter is disabled. System need to power up again for restart.



a) Main state machine



b) Run-sub-state machine

Figure 27. Application main state machine and Run-sub-state machine

The RUN state is divided into three sub-state according to PFC application, as shown in *Figure* 27.b) When application state machine transits from STOP to RUN, system first comes into the soft-start sub-state. The bus voltage reference will follow a ramp to make the bus voltage rise smoothly. After the output voltage or its reference reaches the target, system transfers into the NORMAL sub-state. Under NORMAL sub-state, when the load is too light, system will come into the LIGHTLOAD sub-state. And vice versa, system will come from LIGHTLOAD to NORMAL when the load increases and makes the output voltage lower than the burst low limit. The application main state machine is declared in state_machine.c and state_machine.h. It is declared as:

```
/* State machine control structure */
typedef struct
{
__pmem SM_APP_STATE_FCN_T const* psState; /* State functions */
__pmem SM_APP_TRANS_FCN_T const* psTrans; /* Transition functions */
SM_APP_CTRL uiCtrl; /* Control flags */
SM_APP_STATE_T eState; /* State */
} SM_APP_CTRL_T;
```

It consists of four parts:

- psState: pointer to the user state functions.
- psTrans: pointer to the user transition functions. When the system transfers between different states, corresponding transition function is called.
- uiCtrl: the system flow control flag.
- eState: the actual state of the state machine.

The state machine is periodically called in the software by inline function which calls the user defined state function. The corresponding user end state functions and transition functions are defined in PFC_statemachine.c.

```
/* pointer to function with a pointer to state machine control structure */
typedef void (*PFCN_VOID_PSM)(SM_APP_CTRL_T *sAppCtrl);


/* State machine functions field (in pmem) */
__pmem const PFCN_VOID_PSM gSM_STATE_TABLE[4] = {SM_StateFault, SM_StateInit, SM_StateStop,
SM_StateRun};


/* State machine function */
extern inline void SM_StateMachine(SM_APP_CTRL_T *sAppCtrl)
{
gSM_STATE_TABLE[sAppCtrl -> eState](sAppCtrl);
}
```

The sub-state functions and corresponding transition functions are defined in PFC_statemachine.c and are called when the system is in the RUN state as the same way as the main state.

4.3. Control timing

The PFC software consists of two periodical interrupts and a background loop. The first periodical interrupt PWM0_ISR() is driven by the VAL0 compare event from eFlexPWM SM0. This routine is configured for higher priority to execute the inner current loop calculation at 80 kHz and get the desired switching signal. The other periodical interrupt PWM1_ISR() is driven by the VAL0 compare event from eFlexPWM SM1.This routine is used as a lower priority to execute outer voltage loop calculation at 10 kHz and the input voltage and current RMS are calculated here when a zero-cross is detected. The background loop runs in an endless loop. It includes the application state machine and communication with PC (FreeMASTER) and the backward LLC.

To summarize, PWM0_ISR() is a time critical routine. The FSLESL is applied to minimize the total execution time.

In this PFC application, PWM output signal is re-routed onto the output trigger ports so that it can be sent to the chip routing logic through XBAR and AOI block to generate the final drive signal. Since the output trigger port is connected with PWM output signal, we need additional sub-module to trigger ADC A&B sample. MC56F82748 eFlexPWM module has four sub-modules SM0~4. SM0 is used to generate higher priority interrupt service request PWM0_ISR() which updates PWM output duty and synchronize the ADC sample. SM1 is used to generate PWM1_ISR() which executes voltage loop calculation. SM2 is used to provide output trigger to ADC A&B sample. The ADC trigger is a little bit ahead of the switch midpoint, so that center aligned PWMs ensures ADC sampling point at the center of PWM and not affected by the PWM switch operation and can get the middle value of inductor current when current rises. Besides, the timing of SM0 VAL0 event and ADC trigger makes register push, algorithm that doesn’t need real time sample results and ADC sampling operate simultaneously which leaves more time for other algorithm calculation.

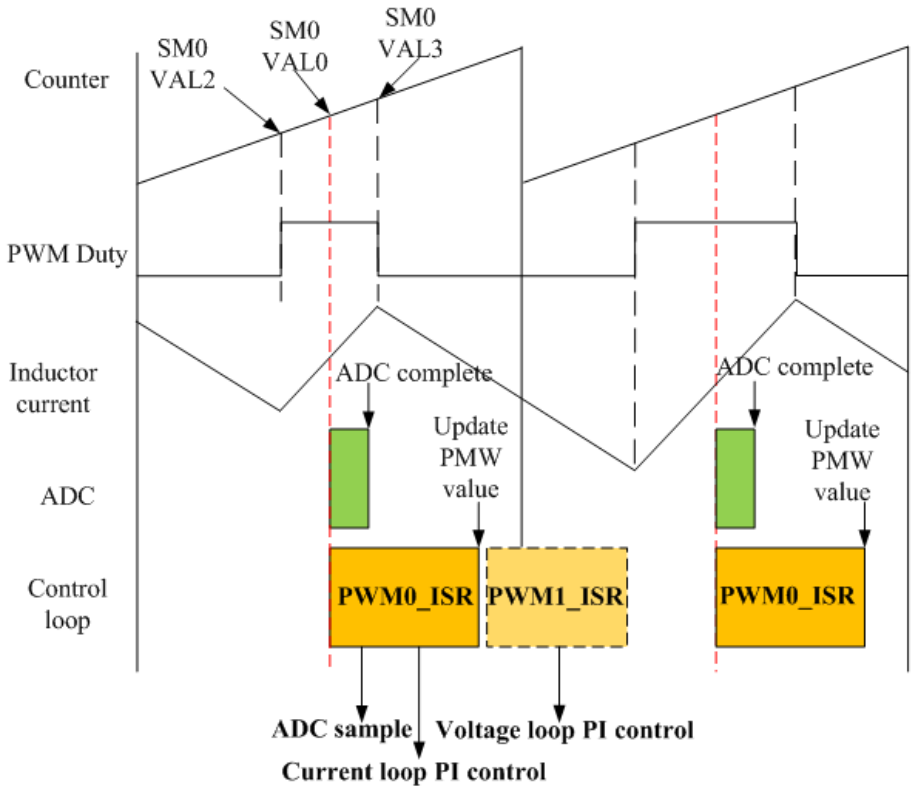


Figure 28. Control timing details

When counter matches SM2 VAL0, PWM_OUT_TRIG0 of SM2 is generated to trigger ADC A&B sample. The PWM0_ISR started at nearly the same time (actually PWM0_ISR is ahead a little), which leaves more time for algorithm calculation. Because the fast loop frequency is eight times of the slower loop, it means PWM0_ISR() is executed eight times before PWM1_ISR() is executed once. The figure shows that between updating PWM value and the next ADC trigger, there is sufficient time for PWM1_ISR() to run. And the center aligned PWM will ensure average current sample for the control scheme.

4.4. Drive signal generation logic

Drive signals for GaN HEMTs are generated by eFlexPWM SM0. The generated PWMs can output directly or connect to internal-peripheral crossbar switch (XBAR).

Drive signal generation logic is shown in *Figure* 29. The SW PWM output control is used to turn on/off the PWM outputs. PWM_L& PWM_H are GaN HEMT drive signals and synchronous rectification drive signal for MOSFETs are generated by GPIO output. All combinations are implemented by hardware, which ensures the timeliness of drive signal and simplifies the software design.

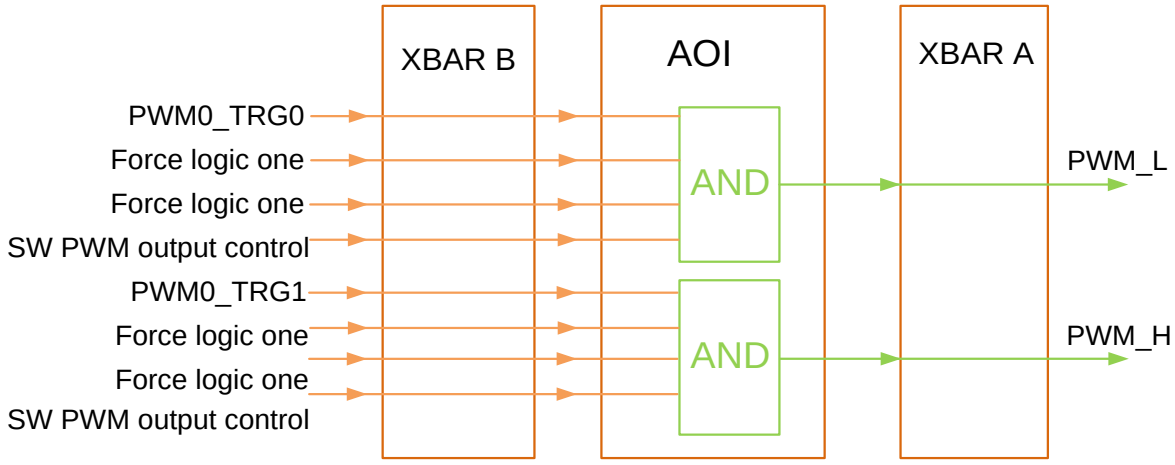


Figure 29. Drive signal generation logic

4.5. Fault protection

The hardware protection is over-current protection of inductor current, output voltage over-voltage protection. Over-current protection is generated by the build-in high speed comparator CMPB and CMPC for both positive and negative side. Over-voltage protection is generated by comparator D.

The software protection includes inductor over-current protection, output under-voltage and over-voltage protection, as well as input voltage under-voltage protection and over-voltage protection.

4.6. Peripherals settings

4.6.1. ADC converter

The ADC A&B converters are set to run simultaneously and triggered by VAL0 compare event of eFlexPWM sub-module 2. The trigger signals connection between eFlexPWM and ADC is provided by a cross bar switch module. The ADC A samples input voltage and output DC bus voltage, ADC B samples input current.

4.6.2. Pulse width modulator eFlexPWM

The eFlexPWM sub-module 0 generates two complementary PWM signals with constant frequency 80 kHz and variable duty cycle. The generated PWM signals are routed to XBARB for the generation of GaN drive signals. VAL0 compare event is used to generate interrupt routine PWM0_ISR() to handle inner current loop algorithm. In order to maximum the algorithm execution time, VAL0 is set to 35 bus cycles ahead of the switch midpoint.

The eFlexPWM sub-module 1&2 are used for timing control and no PWM signal is generated. Sub-module 1 is used to generated the 10kHz interrupt routine PWM1_ISR() to handle outer voltage loop algorithm. Sub-module 2 is used to generate the trigger signal for ADC. In order to obtain the accurate input average current, ADC trigger signal is set to 20 bus cycles ahead of the switch midpoint. Because

sub-module 2 is synchronized with sub-module 0 by master reload forcing initialization, the ADC trigger signal is almost synchronized with PWM0_ISR() trigger.

4.6.3. High speed comparator HSCMPD

The build-in high-speed comparator D is used to detect over voltage condition of the output voltage. One of the comparator's inputs is the measured output voltage, the other is the fault threshold set by VREF D.

4.6.4. High speed comparator HSCMPB&C

The build-in high-speed comparator B&C are used to detect both side over current condition of the input current. One of the comparator's inputs is the single side measured output voltage, the other is the fault threshold set by VREF B&C.

4.6.5. Inter-peripheral crossbar switch XBARA&B and AOI

Crossbar switch A provides the connection for trigger signals between eFlexPWM and ADC. It also works together with crossbar switch B and AOI module to generate the drive signals for GaN.

4.6.6. Serial communication interface SCI0 & SCI1

The serial communication interface SCI0 is used for communication with backward LLC, and it can exchange the information, accept the control instructions and execute.

The serial communication interface SCI1 is used for communication with the host PC for FreeMASTER or firmware updating. There is also a SCI -> USB converter on the board, so the communication with host PC finally goes via USB interface.

4.7. Code runs in RAM

There are two speed modes for DSC56F82748. In normal mode, codes run at a maximum frequency of 50MHz in both Flash and RAM. In fast mode, codes run at 50MHz in flash, but 100MHz in RAM. In this application, the time critical routine PWM0_ISR() is configured to runs in RAM for time saving.

Because RAM is volatile, the relocated code is stored in flash and copied into RAM during

microcontroller start-up. In order to realize this goal, take the following steps:

Disable the auto generated LCF by processor Export option, as shown in Figure 30.

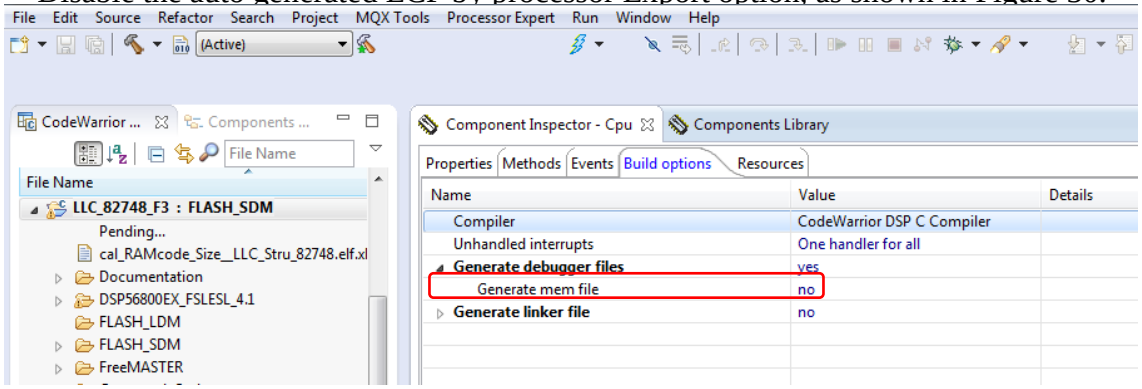


Figure 30. Disable LCF auto generation

- Define code sections with pragma directive. Both the PWM0_ISR() routine and the functions called in it are included in this section.

```
#pragma define_section ramCodes_main "ramFunc_main.text" RX
        #pragma section ramCodes_main begin
        #pragma interrupt saveall
        void PWM0_ISR(void)
{    …    }
        #pragma interrupt off
        #pragma section ramCodes_main end

pragma define_section ramCodes_pfc "ramFunc_pfc.text" RX
#pragma section ramCodes_pfc begin
static void PFC_StateRun_Normal(void)
{    …    }
        #pragma section ramCodes_pfc end
        #pragma section ramCodes_pfc begin
        static void PFC_StateRun_Lightload(void)
{    …    }
#pragma section ramCodes_pfc end
```

- Shrunken the .p_Code memory segment length and assign a size fixed flash area for relocated code with linker command file (LCF).

```
MEMORY {
        .x_Peripherals  (RW)  : ORIGIN = 0xC000, LENGTH = 0

        # List of all sections specified in the "Build options" tab
        .p_Interrupts   (RWX) : ORIGIN = 0x00000000, LENGTH = 0x000000DE
        .p_Code         (RWX) : ORIGIN = 0x00000208, LENGTH = 0x00006DF5
        .x_Data         (RW)  : ORIGIN = 0x00000000, LENGTH = 0x00001000
        .p_reserved_FCF (RWX) : ORIGIN = 0x00000200, LENGTH = 0x00000008
        .x_internal_ROM (RW)  : ORIGIN = 0x000040DE, LENGTH = 0x00000122
        .p_ramFuncCode  (RWX) : ORIGIN = 0x00007000, LENGTH = 0x00001000
        .p_ramFuncSpace (RWX) : ORIGIN = 0x0000F000, LENGTH = 0x00001000

        .p_flash_ROM_data  (RX) : ORIGIN = 0x00000000, LENGTH = 0x00001000
}
```

- Reside the defined code section in corresponding section segment with linker command file (LCF). And remember to reconfigure the address of each section segments to prevent overlap.

```
SECTIONS {
        .at_of_p_ramFuncSpace :
             {
               WRITEW(0X2); # dummy insertion, prevent warning
               __pROM_data_start = .;
             } > .p_ramFuncCode

              .data_in_p_flash_ROM : AT(__pROM_data_start)
             {
                 …
              __ramFunc_code_start = __pROM_data_start + __data_size;
             }

       .codes_onRAM : AT(__ramFunc_code_start)
             {
        . = . + __data_size; #prevent overlap with p_falsh_ROM_data
              F_RAMcode_start_addr = .;
              * (rtlib.text)
        #-------------------------------------------------------
        #-- relocate functions to section .codes_onRAM  ---------
        #-------------------------------------------------------
                     * (ramFunc_main.text)
                     * (ramFunc_pfc.text)
        #-------------------------------------------------------
        #-- manually relocate called library into this section --
        #-------------------------------------------------------
                     GFLIB_SDM.lib  (.text)
                     PCLIB_SDM.lib  (.text)
                     . = ALIGN(2);
              F_RAMcode_end_addr = .;
              __ramFunctions_size = F_RAMcode_end_addr - F_RAMcode_start_addr;
             __ramFunctions_LdAddr = __ramFunc_code_start + __data_size;
             } >.p_ramFuncSpace
…
}
```

- Enable pROM-to-pRAM and pROM-to-xRAM copy utility and set the pass code sizes, resident address and runtime address in LCF.

```
F_Livt_size      = __ramFunctions_size;
F_Livt_RAM_addr = F_RAMcode_start_addr;
F_Livt_ROM_addr = __ramFunctions_LdAddr;


F_xROM_to_xRAM    = 0x0000;
F_pROM_to_xRAM    = 0x0001;
F_pROM_to_pRAM    = 0x0001;
```

For more details on how to relocate codes into RAM, refer to AN5143: Relocate subroutines to PRAM for MC56F827xx DSC.

5. Getting started

TP-PFC main board is shown in *Figure 31*. The control circuit is powered by auxiliary power. One isolated UART port is reserved for more usage.

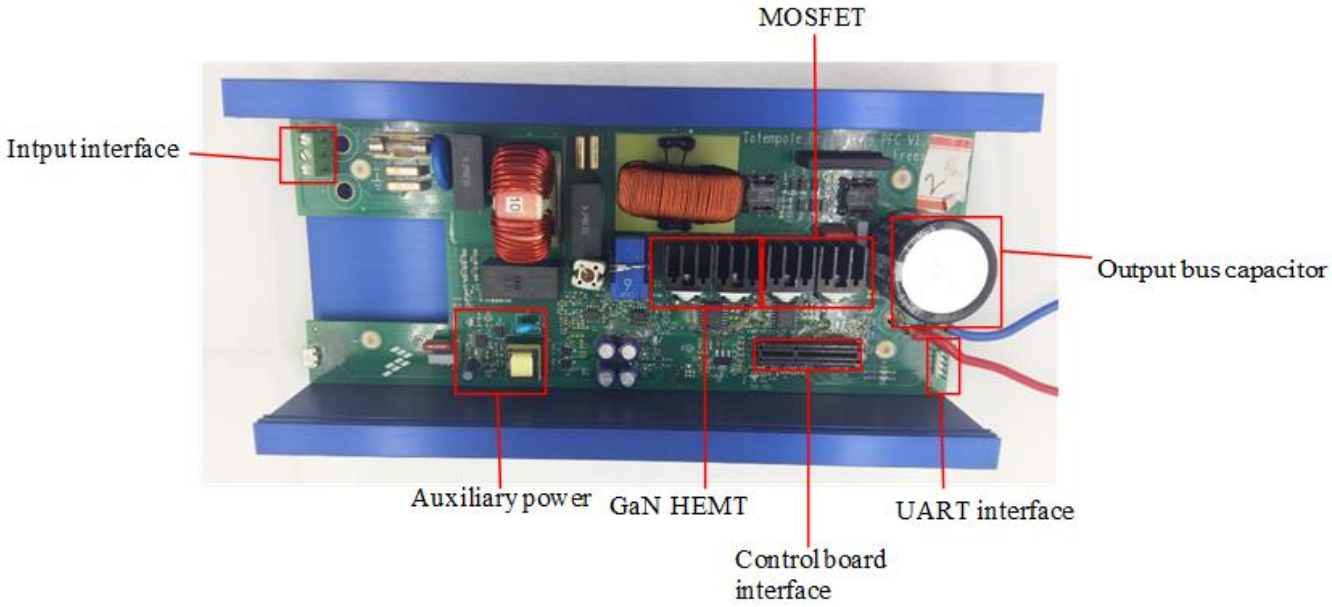


Figure 31. Totem-pole bridgeless PFC main board

5.1. System development environment

The TP-PFC supports debugging with CodeWarrior and FreeMASTER. The set-up of the debug connection is shown in *Figure* 32.

To download program to the control chip, you need to connect a debugger (USB TAP or P&E-Multilink) to SWD port of the control board using a 14-pin cable. The control board can be powered from the main board or micro USB interface in the control board. In order to control and monitor the

working status of the TP-PFC system, you must connect the micro interface on the control board to the PC through a micro USB cable.

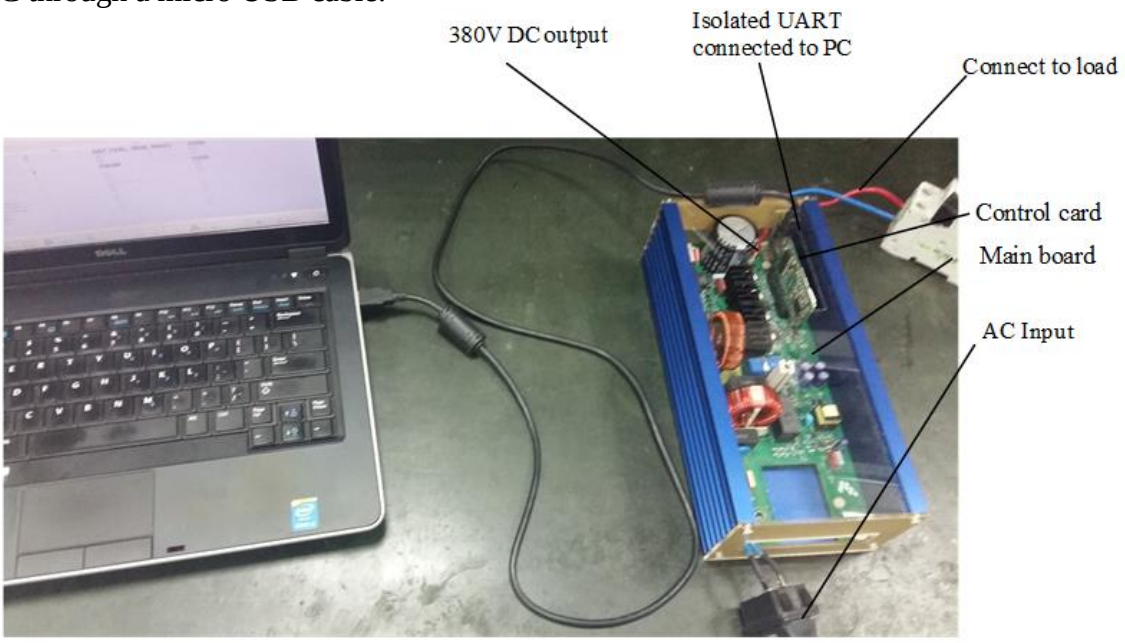


Figure 32. Development environment

5.2. Debugging TP-PFC board with FreeMASTER

FreeMASTER is the tool for controlling and monitoring. PFC.pmp is needed.

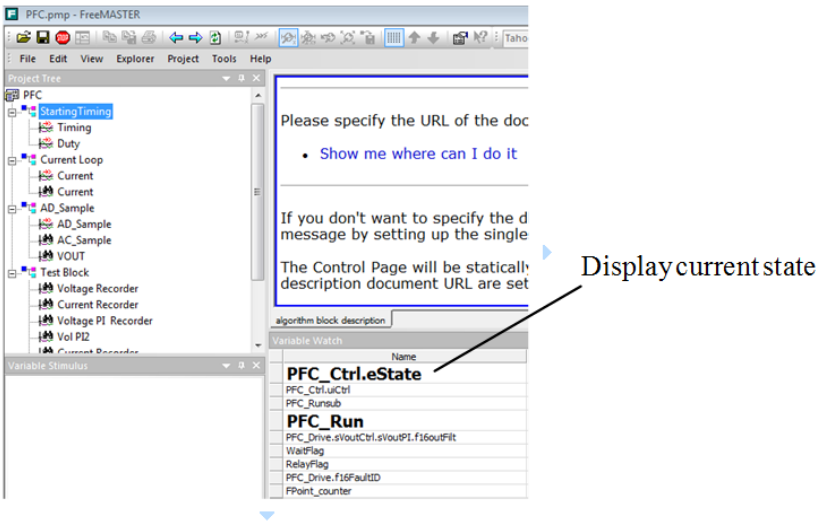


Figure 33. PFC main control pane

In order to connect the FreeMASTER to the target board, perform the following steps:

1. Select a symbol file for your project

Select the symbol file in FreeMASTER by navigating to Project –> Options –> MAP Files, as shown in *Figure* 34.

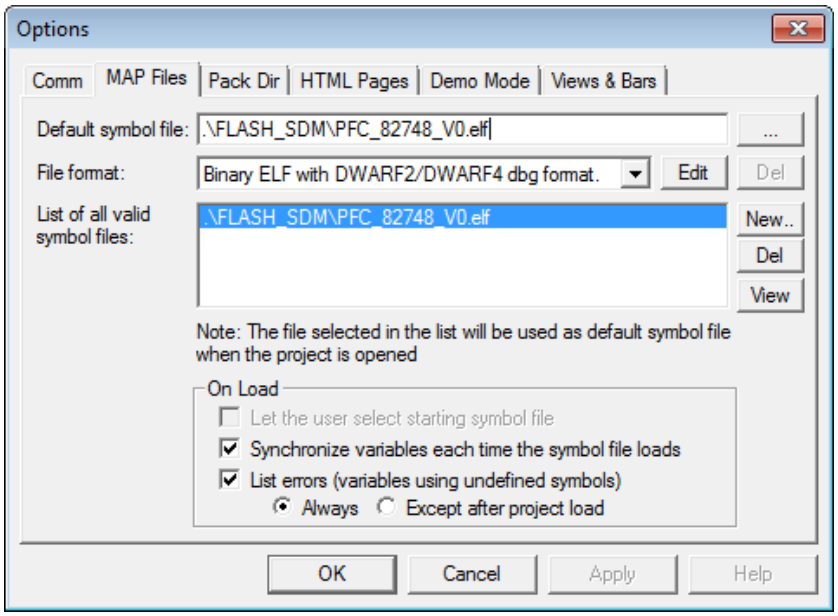


Figure 34. Selecting a symbol file

2. Adjust the settings for using FreeMASTER

Select “Direct RS232” in FreeMASTER by navigating to Project –> Options –> Comm, and set the speed as 19200 as shown in *Figure* 35.

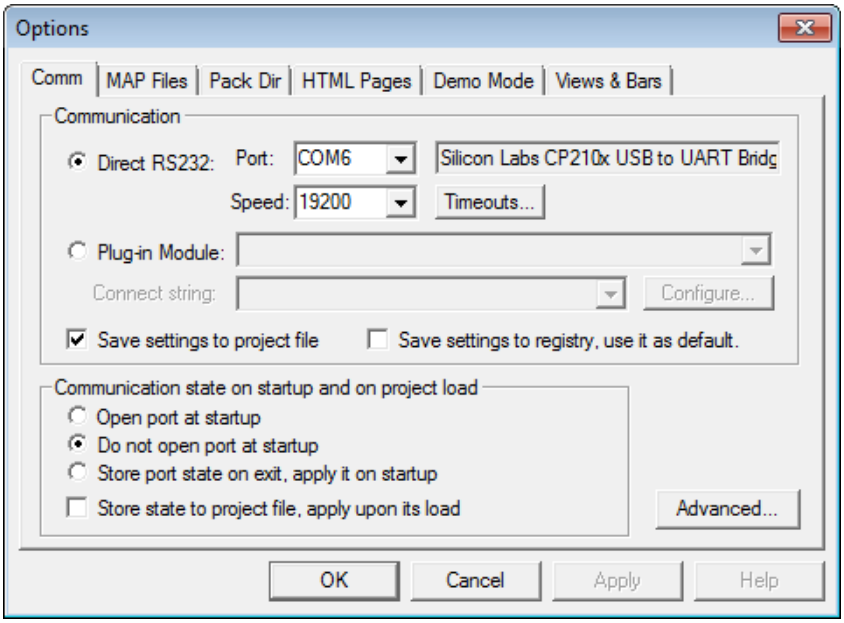


Figure 35. FreeMASTER options dialog box

Note: The micro interface in the control board must be connected to PC for FreeMASTER control. Totem-pole bridgeless PFC **must not** start up with load to prevent from damage of start-up resistor.

6. Testing

This section provide the testing results of totem-pole bridgeless PFC.

6.1. System efficiency

Table 3~ *Table* 4 shows the system efficiency at different voltage.

Table 3. **System efficiency at VIN=110V RMS**

Vin(V)	Iin(A)	Pin(W)	PF	THDi	Vout(V)	Iout(A)	Pout(W)	Efficiency
109.55	1.7658	193.12	0.9983	3.25%	381.6	0.484	184.6944	95.64%
109.08	3.5062	382.35	0.9997	1.98%	381.8	0.961	366.9098	95.96%

Table 4. **System efficiency at VIN=220V RMS**

Vin(V)	Iin(A)	Pin(W)	PF	THDi	Vout(V)	Iout(A)	Pout(W)	Efficiency
220.04	0.8775	190.23	0.9851	6.73%	381.5	0.484	184.646	97.07%
219.77	1.7149	374.92	0.9947	4.98%	381.6	0.961	366.7176	97.81%
219.51	2.5574	559.62	0.9968	2.76%	381.8	1.436	548.2648	97.97%
219.37	2.9737	651.05	0.9979	2.50%	381	1.677	638.937	98.14%

Figure 36. Shows the system efficiency power factor curve. Figure 37. Shows the Total Harmonic Distortion value curve. Figure 38. shows the efficiency curve.

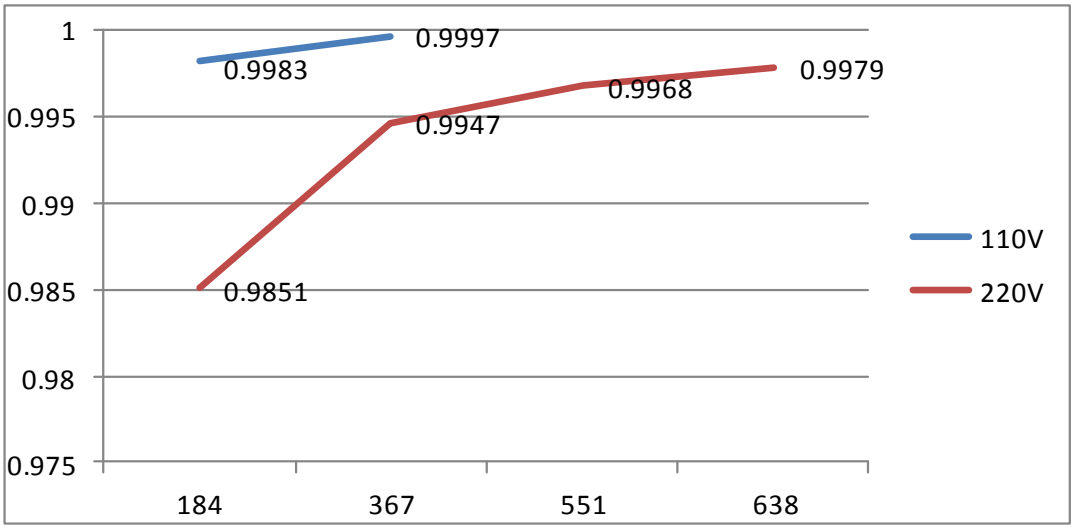


Figure 36. Power Factor value curve

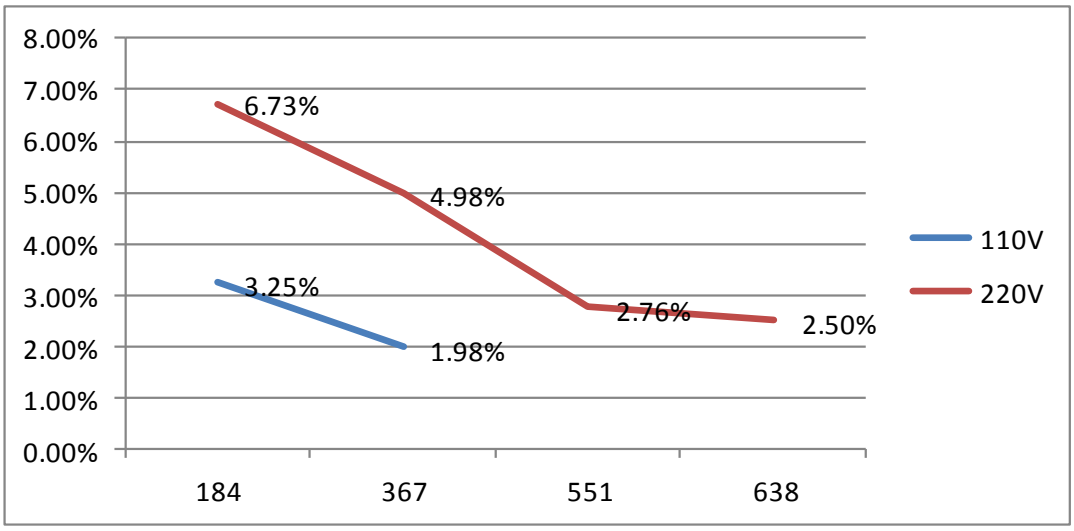


Figure 37. Total Harmonic Distortion value curve of input current

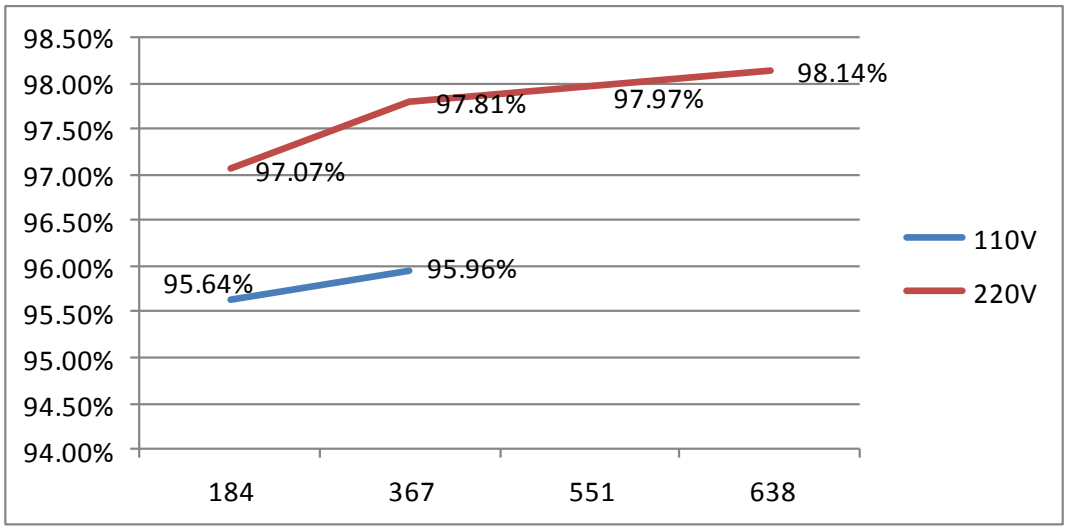


Figure 38. Efficiency curve

6.2. Startup performance

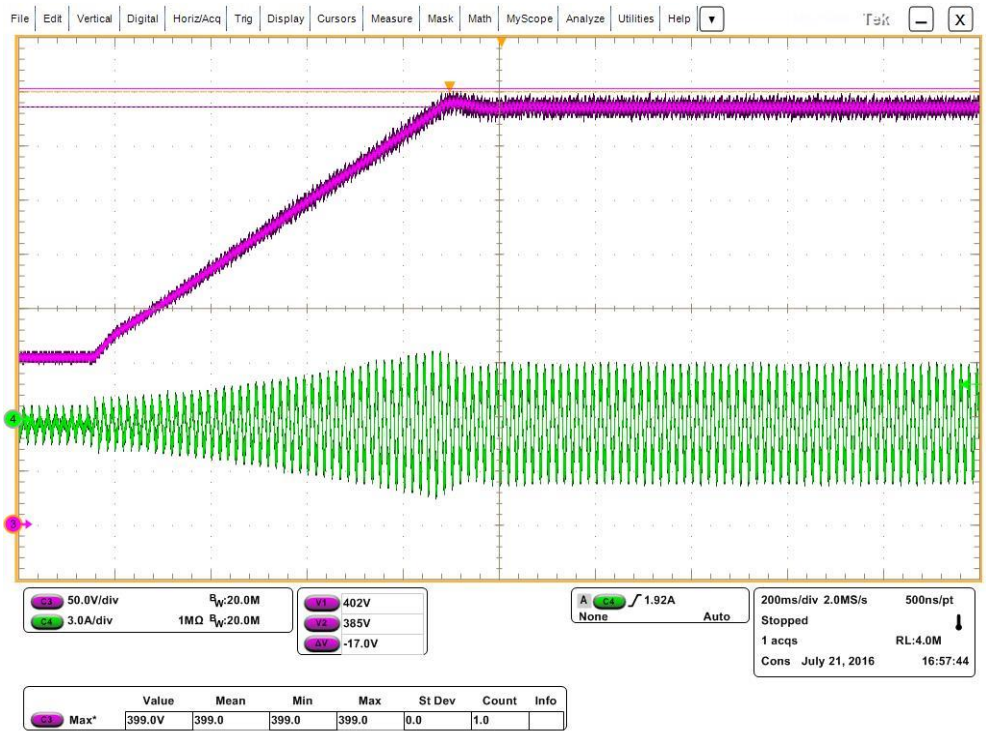


Figure 39. Start-up of totem-pole bridgeless PFC(CH3:Output voltage;CH4:inductor current)

6.3. Steady state performance

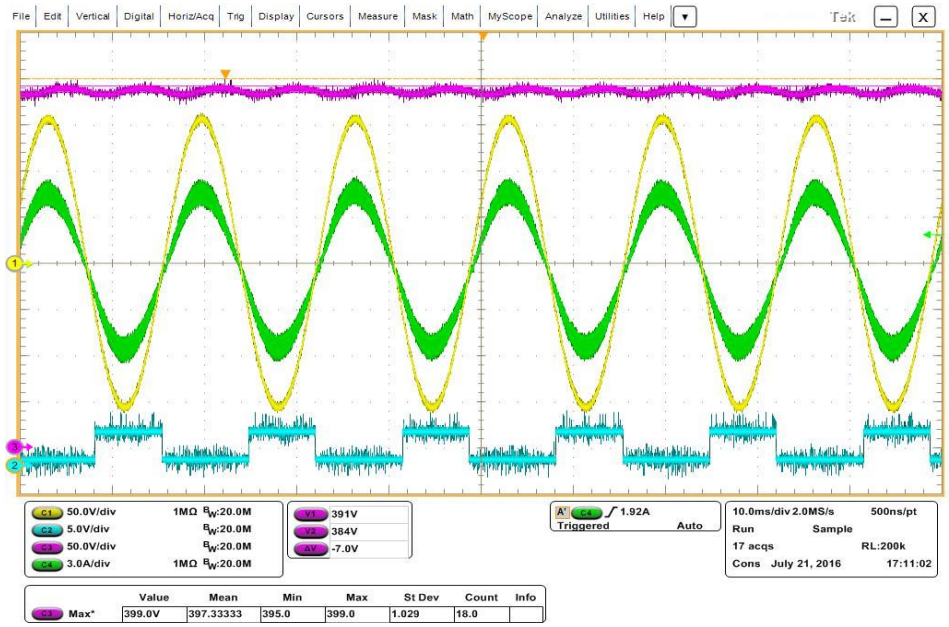


Figure 40. Steady-state of totem-pole bridgeless PFC(CH1:Input AC voltage;CH2:PWM gate for S_{D2};CH3:Bus capacitor voltage;CH4:Inductor current)

6.4. Dynamic performance

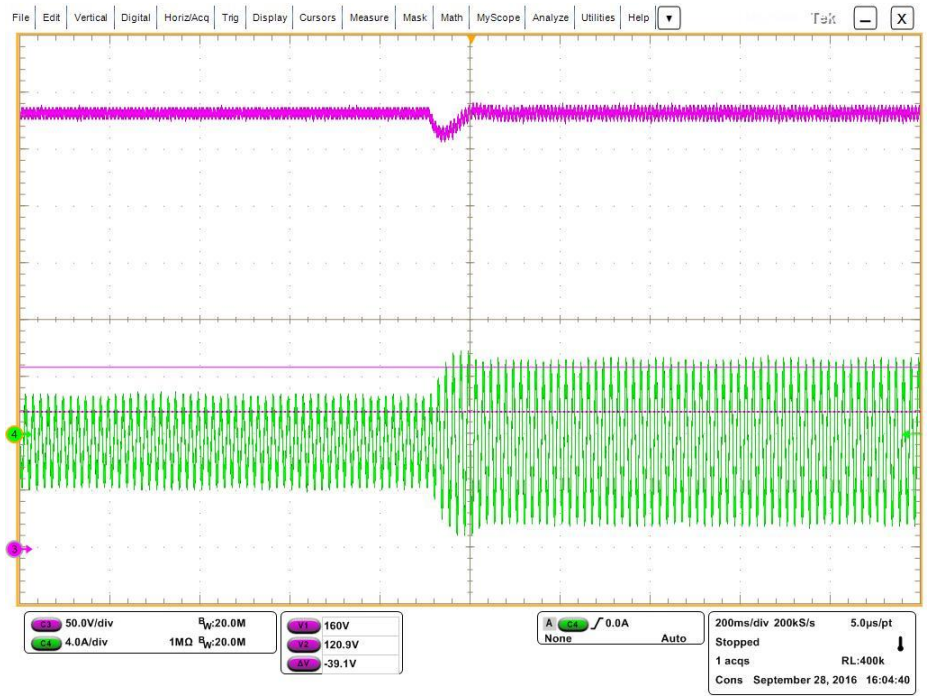


Figure 41. Load 50% to 100% transition performance at 110V input; Ch3: Output Voltage; Ch4: Input Current

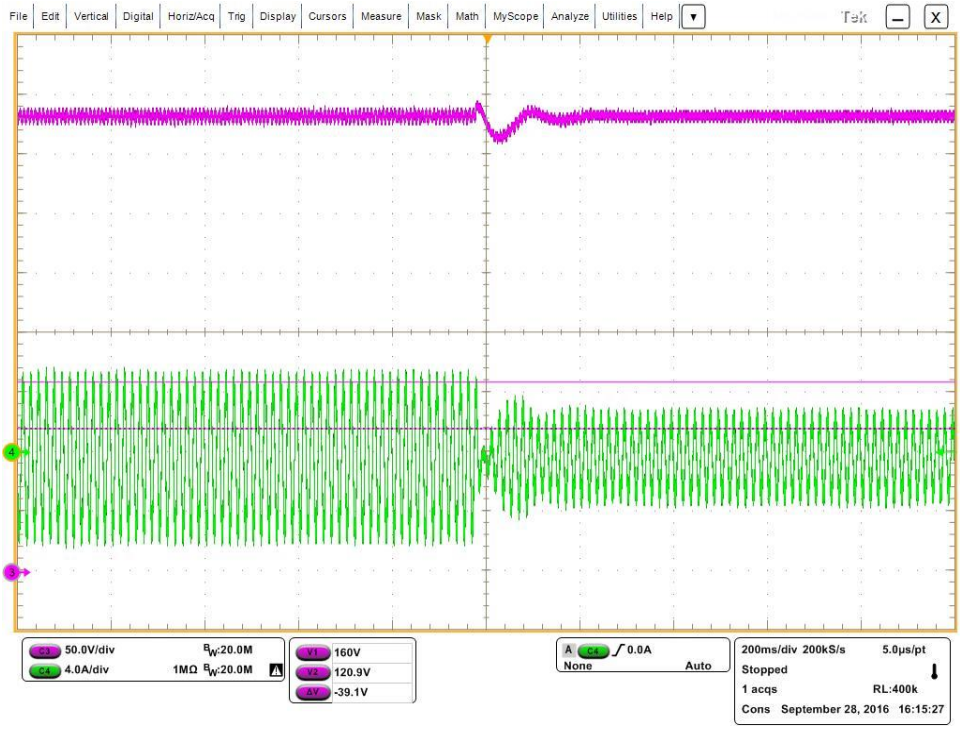


Figure 42. Load 100% to 50% transition performance at 110V input;Ch3:Output Voltage;Ch4 Input Current

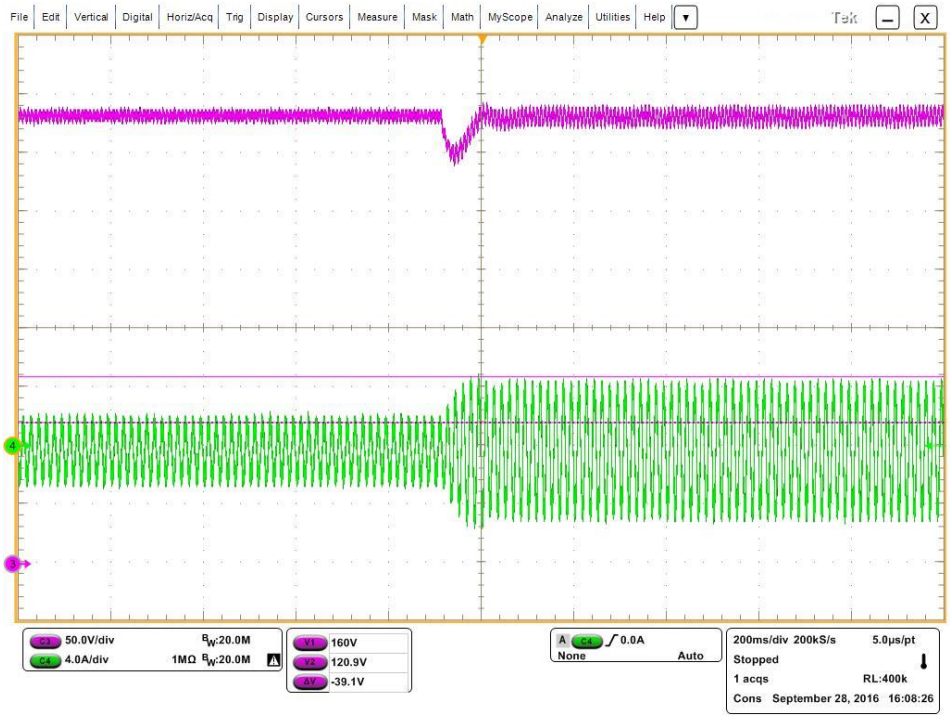


Figure 43. Load 50% to 100% transition performance at 220V input; Ch3: Output voltage; Ch4: Inductor current

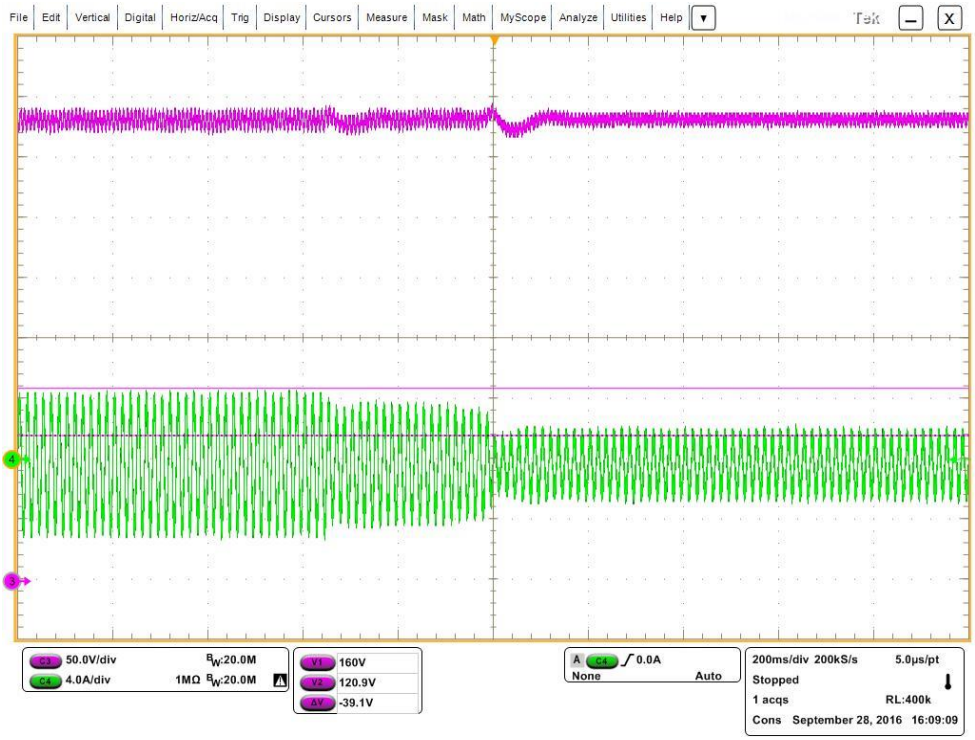


Figure 44. Load 100% to 50% dynamic transition performance at 220V input; Ch3: Output Voltage; Ch4: Input Current

7. Revision history

Revision number	Date	Substantive changes
0	11/2016	Initial release

How to Reach Us:

Home Page:
www.nxp.com

Web Support:
www.nxp.com/support

Document Number: DRM174
Rev. 0
11/2016