AN12618

Totem-Pole Bridgeless PFC Using MC56F83783

Rev. 0 — 10/2019

Application Note

Contents

1 Introduction

This application node describes the implementation of a Totem-Pole Bridgeless Power Factor Correction (PFC) using MC56F83783. MC56F83783 is a member of 32-bit 56800EX core-based Digital signal controllers (DSCs). The processing power of this core and the peripherals available on this MCU are dedicated for easy implementation of high performance power conversion applications.

This document focuses on the system description and software design. System description part includes system structure, controller features, control logic, different working mode, and control loop. Software design part mainly includes the project file structure, state machine, control timing, peripheral configuration and zero crossing PWM generation. The detailed hardware design and control loop design is described in *Totem-Pole Bridgeless PFC Design Using MC56F82748* (document DRM174).

2 System description

2.1 System structure

The application is developed on high voltage bridgeless PFC power stage and the MC56F83783 controller card. The high voltage power stage contains the whole power circuitry, the signal conditioning stage, drive circuit and the auxiliary power supply. 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 amplification of DSC PWM signals. The synchronous rectifiers are used to reduce the conduction losses in replace of conducting diodes.

DSC MC56F83783 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.

One isolated Universal Asynchronous Receiver / Transmitter (UART) port is used to communicate with other devices, such as backward stage LLC resonant converter. 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 1.

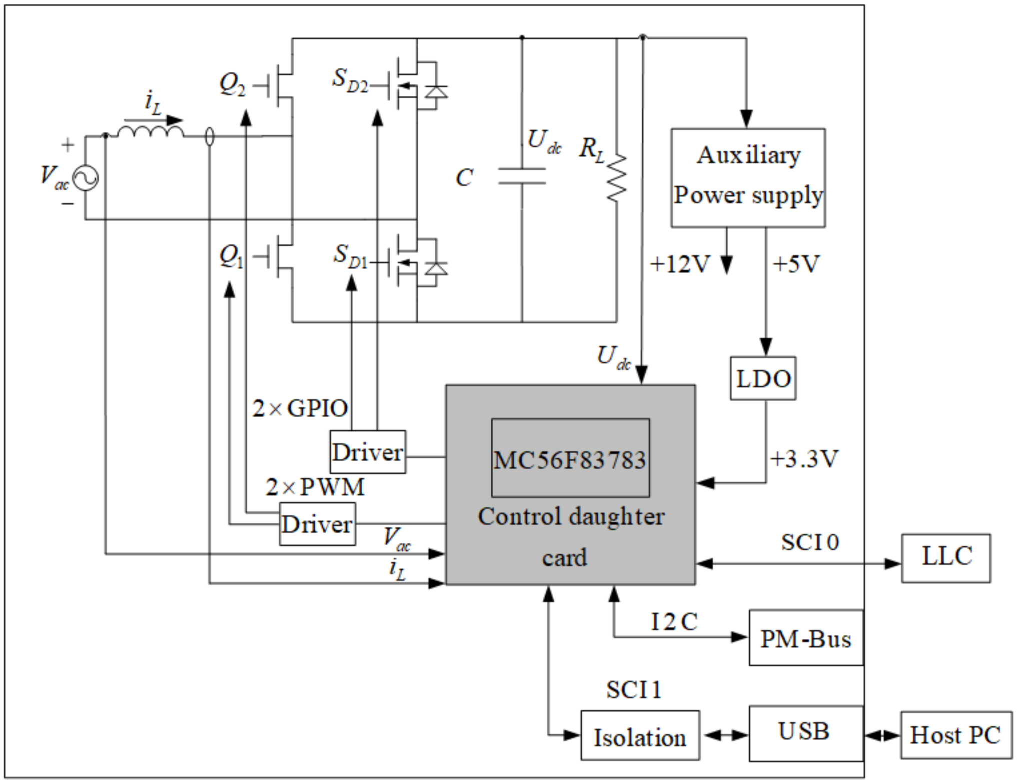


Figure 1. System block diagram

2.2 Controller feature

The 56F837xx 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, 56F837xx is well-suited for many consumer and industrial applications.

With numerous, highly-integrated peripherals and powerful processing capabilities, the 56F837xx is a low-cost family especially useful for industrial control, motion control, home appliances, general-purpose inverters, smart sensors, fire and security systems, switched-mode power supply, power management, wireless charging, UPS, Solar inverter, and medical monitoring applications.

The following list summarizes the superset of features across the entire 56F837xx family.

- 56800EX 32-bit DSC core.
- Up to 100 MHz operation frequency.
- Up to two 128 KB program flash memory with ECC protection and partition swap function, up to 64 KB dual port program/ data RAM, up to 32 KB boot ROM.
- Protects supervisor programs and resources from user programs.
- Two 8-channel eFlexPWM module with NanoEdge^{TM} placement and enhanced capture.
- Two 8-channel 12-bit cyclic Analog-to-Digital Converters (ADCs).
- One 4-channel eDMA.
- One windowed watchdog timer, power Supervisor.
- On-chip 48 MHz/200 kHz relaxation oscillator and 4 MHz to 16 MHz Crystal Oscillator (XOSC).

- Inter-Module Crossbar and Event Generator (EVTG) perform Boolean logic with Flip-Flop being included.
- Programmable Interrupt Controller (INTC).
- Two Quad Timer, two Periodic Interval Timers.
- Two 12-bit DAC modules.
- Four High Speed Comparators with integrated 8-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, inter-module crossbar and EVTG.

This PWM module offers flexibility in its configuration and 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 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, up to 312.5 ps.
- PWM outputs that can operate as complementary pairs or independent channels.
- Six synchronization events can be generated per PWM cycle and output via hardware.
- Both PWM outputs and trigger events can pinout through XBAR.
- Support for synchronization to external hardware or other PWM, half cycle reload capability.
- Trigger frequency can be controlled to be consistent with the reload frequency.
- Fault inputs with programmable filters can be assigned to control multiple PWM outputs.
- Independently programmable PWM output polarity, 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 ADC function consists of two 12-bit resolution separate analog-to-digital converters, each with eight analog inputs and its own sample and hold circuit. A common digital control module configures and controls the functioning of the converters. ADC features include:

- Two independent 12-bit ADCs:
  - Two 8-channel external inputs.
  - Built-in x1, x2, x4 programmable gain pre-amplifier.
  - Maximum ADC clock frequency up to 25 MHz, having period as low as 40-ns.
  - Single conversion time of 10 ADC clock cycles.
  - Additional conversion time of 8 ADC clock cycles.
- Support of analog inputs for single-ended and differential conversions.
- Sequential, parallel, and independent scan mode.
- All 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.
- A scan can pause and await new SYNC input prior to continuing.
- Optional interrupts at end of scan.

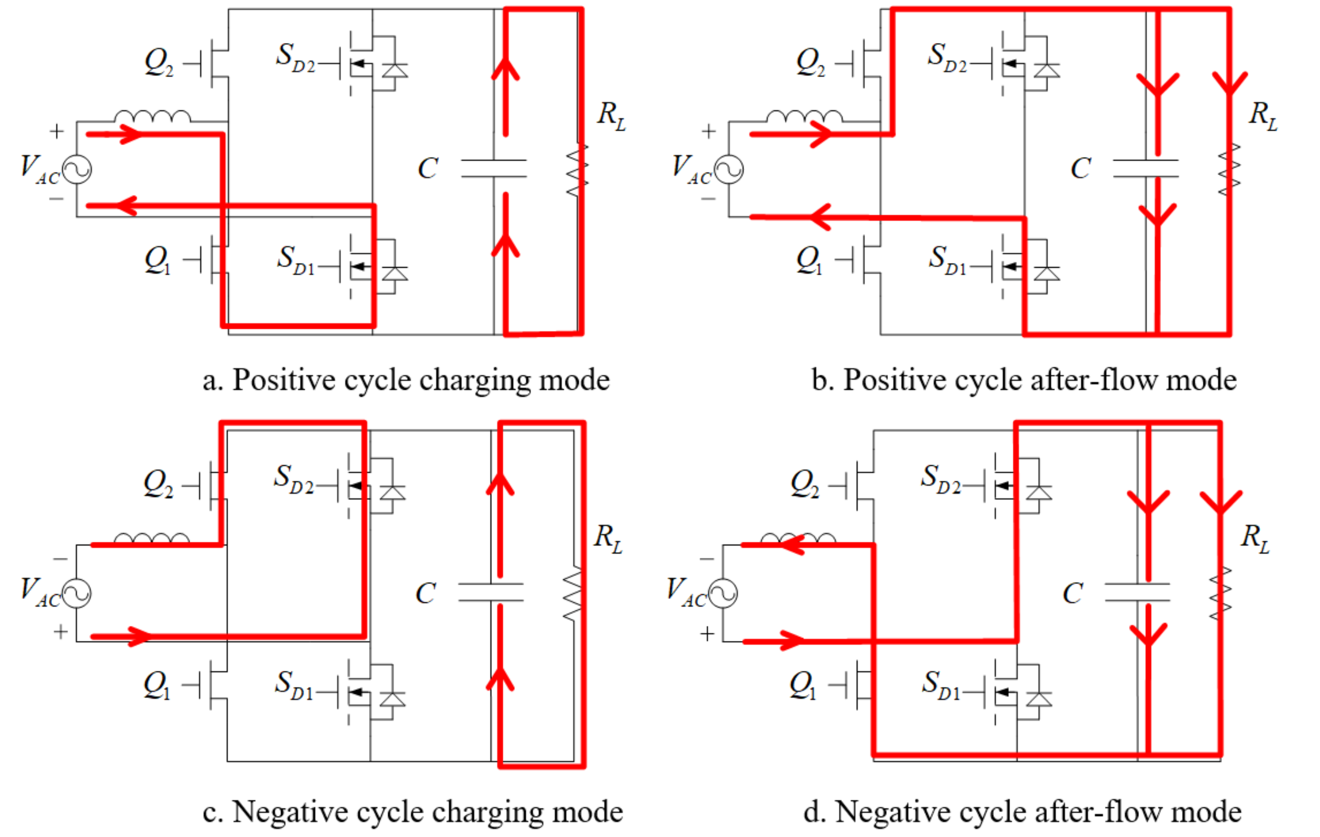
The Inter-Module Crossbar implements an array of M N-combinational muxes. All muxes share the same N inputs but has its own independent select field. XBAR features include:

- 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.
- Register write protections.

The EVTG module includes the following features:

- Highly programmable module for creating combinational boolean events.
- Each EVTG has four inputs and two outputs, two AOI.
- One flexible FF can be configured as RS, D-FF, T-FF, JK-FF, Latch or bypass.
- Programmable filter to remove input glitch.
- All logics are synchronous in bus `clk` domain.

2.3 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 2. 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. Two silicon MOSFET switches (SD1&SD2) operates at the input voltage frequency. And dead time is added to avoid the short of the bridge for GaN and Silicon MOSFET.

2.4 Working 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 Figure 3. PFC is turned off and output voltage decreases. When output voltage reaches `VDC_BURSTOFF`, PFC is turned on again with constant reference current 0.25 A. If the output voltage drops quickly to a lower threshold, PFC will enter back to normal mode again.

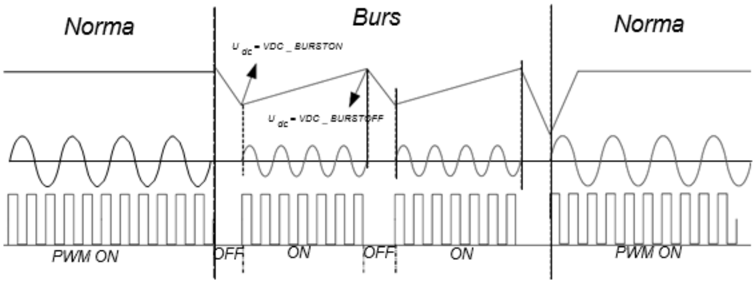


Figure 3. Illustration of normal mode and burst mode

2.5 Control loop

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. Based on the MC56F83783, a digital PFC rectifier is implemented as shown in Figure 4. The PFC control loop's tasks are:

- **Controls inductor current**: To make the current sinusoidal and maintains the same phase as the input voltage.
- **Controls output voltage**: To make the output voltage equaled to target value.

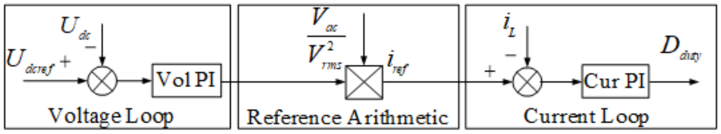


Figure 4. Totem-pole bridgeless PFC control scheme

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

- **Voltage outer loop**: To make sure that the output voltage follows the reference-constant voltage output.
- **Reference arithmetic**: To make sure that the current reference follows the sine reference.
- **Current inner loop**: To make sure that 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, input voltage feedforward is used:

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

Where:

- V_{ac}: The input voltage involved to ensure the current wave follows the input voltage wave.
- V_c: The output of voltage regulator.
- V_{rms}: The RMS value of input voltage.

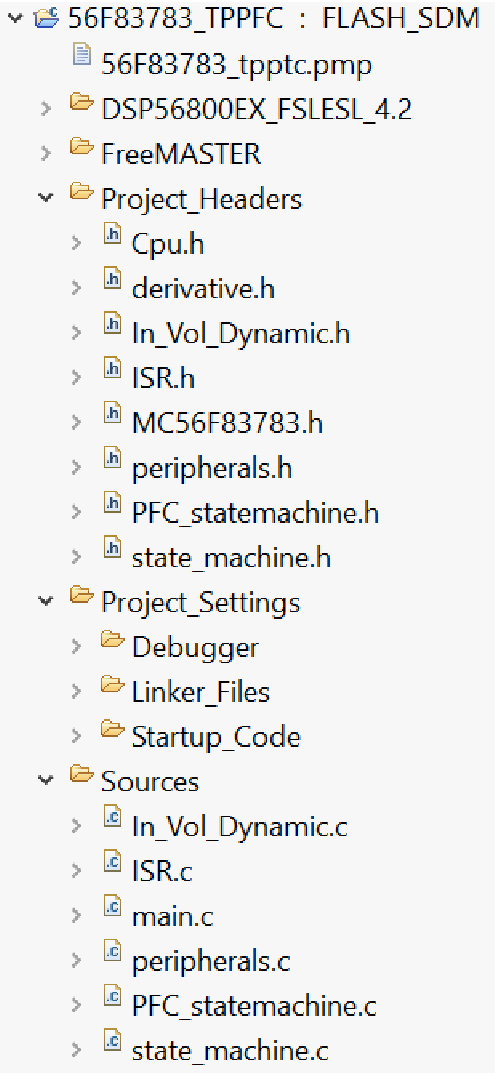
3 Software design

This chapter describes the software design of totem-pole bridgeless PFC using 56F83783, including software structure, state machine, configuration of the DSC peripherals, control timing and PWM generation mechanism .

The software is written in C language using Code Warrior11.1, 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 *Inclusion of DSC Freescale Embedded Software Libraries in Code Warrior 10.2* (document AN4586).

3.1 Project file structure

This project is built based on CodeWarrior 11.1 generated Bareboard project without configuration tool, such as Processor Expert and Quick Start. The overall project files structure is as shown in Figure 5.



```
56F83783_TPPFC : FLASH_SDM
    56F83783_tpptc.pmp
  > DSP56800EX_FSLESL_4.2
  > FreeMASTER
  v Project_Headers
      > Cpu.h
      > derivative.h
      > In_Vol_Dynamic.h
      > ISR.h
      > MC56F83783.h
      > peripherals.h
      > PFC_statemachine.h
      > state_machine.h
  v Project_Settings
      > Debugger
      > Linker_Files
      > Startup_Code
  v Sources
      > In_Vol_Dynamic.c
      > ISR.c
      > main.c
      > peripherals.c
      > PFC_statemachine.c
      > state_machine.c
```

Figure 5. Project file structure

Newly-added and files that need to be modified are listed as follows:

1. `..\DSP56800EX_FSLESL_4.2` : FSLESL libraries.
2. `...\FreeMASTER`: FreeMASTERdrive and configuration files, change FreeMASTER configuration in `freemaster_cfggen.h`.
3. `..\Project_Settings\Linker_Files\MC56F83783_Internal_PFlash_SDM.cmd`: The project linker file.
4. `..\Sources\Main.c`: The mian `C` source file.
5. `..\Sources\ISR.c`: The application interrupt routines `C` source file.
6. `..\Sources\Peripheral.c`: The application peripheral initialization `C` source file.
7. `..\Sources\state_machine.c`: The common statemachine `C` source file.
8. `..\Sources\PFC_statemachine.c`: PFC state function `C` source file.
9. `..\Sources\In_Vol_Dynamic.c`: The specific control functions `C` source file, such as ac drop detection.
10. `..\Project_Headers\ISR.h`: The interrupt vector header file.
11. `..\Project_Headers\peripheral.h`: The application peripheral configuration head file

12. `..\Project_Headers\statemchine.h`: The common statemachine header file.
13. `..\Project_Headers\PFC_statemachine.h`: The application definition head file.
14. `..\Project_Headers\In_Vol_Dynamic.h`: The specific control functions header file.

3.2 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 below.

$$Eqn. 2 \quad Fracvalue = \frac{actualvalue}{quantizationrange}$$

Where:

- *Frac value*: The normalized value of the physical quantity.
- *Actual value*: The actual value of the physical quantity expressed in unit.
- *Quantization range*: The maximum measurable value of this physical quantity.

3.3 State machine

This application uses the state machine to control the system flow, as shown in Figure 6. When the DSC resets and configures all peripherals, the system enters a never-ending loop including the application state machine. The application state machine includes four states:

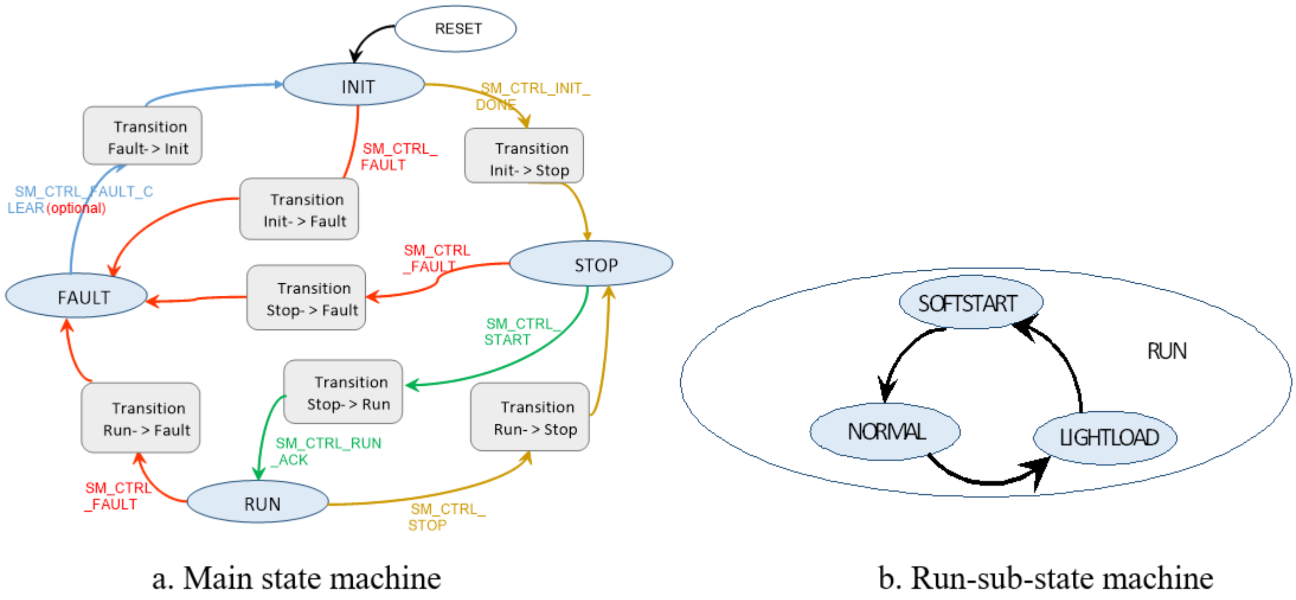
- **INIT**: The variable initialization.
- **STOP**: The system is initialized and checking for the start conditions and PWM output is disabled.
- **RUN**: The system is running, the `run sub-state` is called, and PWM output is enabled or disabled according to the load and can be stopped by the **STOP** command.
- **FAULT**: The system faced a fault condition and 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 `PFC_run` command 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 the **RUN** 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 `PFC_run` command 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 the **FAULT** state. In the **FAULT** state, the 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 the **INIT** state. If not, the software stays in a never-ending loop and output of the converter is disabled. The system needs to power up again for restart.

The **RUN** state is divided into three sub-state according to PFC application, as shown in Figure 6. 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, the system transfers into the **NORMAL** sub-state. Under the **NORMAL** sub-state, when the load is too light, system will come into the **LIGHTLOAD** sub-state. And vice versa, the system will come from **LIGHTLOAD** to **NORMAL** when the load increases and makes the output voltage lower than the burst low limit. 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.



a. Main state machine

b. Run-sub-state machine

Figure 6. Application main state machine and Run-sub-state machine

3.4 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 eFlexPWMA 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 eFlexPWMA 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. MC56F83783 eFlexPWMA 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.

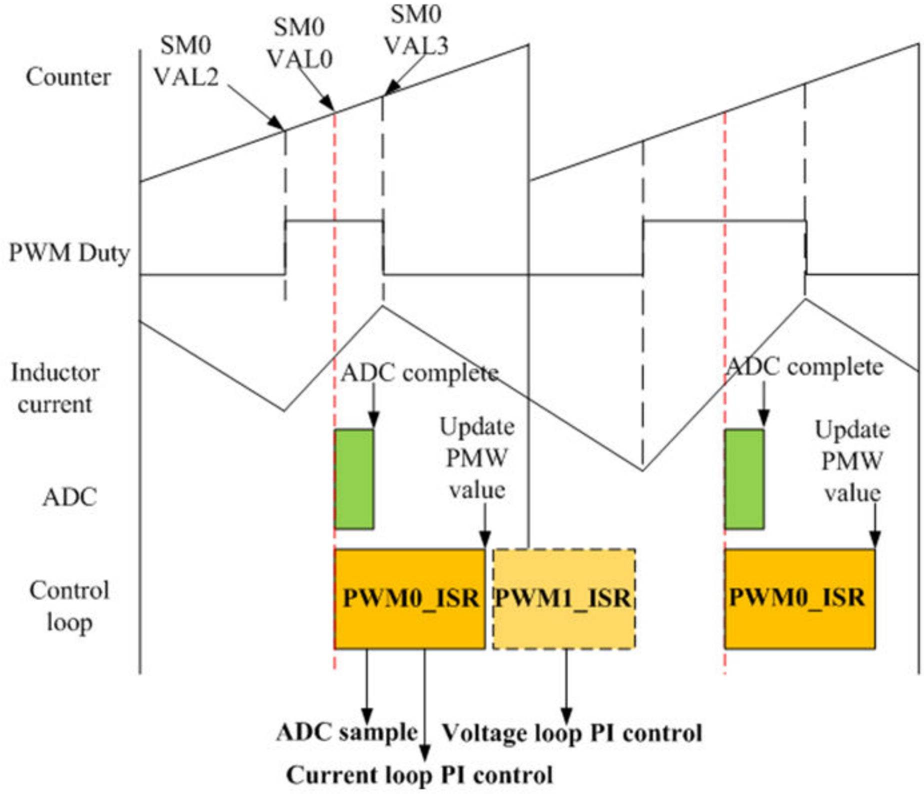


Figure 7. Control timing details

3.5 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.

3.6 Peripheral configuration

This application uses dedicated peripherals for PFC algorithm implementation and communication. The peripherals used in the application are: ADC, eFlexPWM, XBAR, EVTG, CMPs, SCI and GPIOs. The other unused peripherals are disabled and not powered.

3.6.1 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.

3.6.2 Power width modulator eFlexPWM

The eFlexPWMA sub-module 0 generates two complementary PWM signals with constant frequency 80 kHz and variable duty cycle. The generated PWM signals are routed to XBAR. 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 eFlexPWMA 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-module2 is synchronized with sub-module 0 by master reload forcing initialization, the ADC trigger signal is almost synchronized with `PWM0_ISR()` trigger.

3.6.3 High speed comparator HSCMP

- *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 inductor current, the other is the fault threshold set by built-in 8-bit DACB&C.
- *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 built-in 8-bit DACD.

3.6.4 Inter-peripheral crossbar switch XBAR

The XBAR provides flexible connection from any input to any output under user's control. The application configuration is as follows:

- Channel 29, PWMA fault0: from `XBARIN13` and `CMPB_OUT`
- Channel 30, PWMA fault1: from `XBARIN14` and `CMPC_OUT`
- Channel 31, PWMA fault2: from `XBARIN15` and `CMPD_OUT`
- Channel 12, ADC trigger: from `XBARIN24` and `PWMA_trigger0`

3.6.5 Event generator EVTG

Each EVTG includes two AOI module which share the four associated inputs. A XBAR switch is typically used to select the EVTG inputs from available peripheral outputs and GPIO signals. Drive signals can output directly or connect to internal-peripheral crossbar switch (XBAR).

3.6.6 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.

3.7 Zero crossing PWM generation

In bridgeless totem pole PFC, to avoid current pulse in both side at zero crossing point, specific PWM generation mechanism is needed. Figure 8 shows the PWM generation mechanism of this application.

- Active GaN is started with a small duty cycle and increased gradually to the value determined by control loop when zero crossing detected.
- Rectifier FETs delay for some time after the zero crossing before opening.
- Sync GaN is enabled until current increases to a threshold.

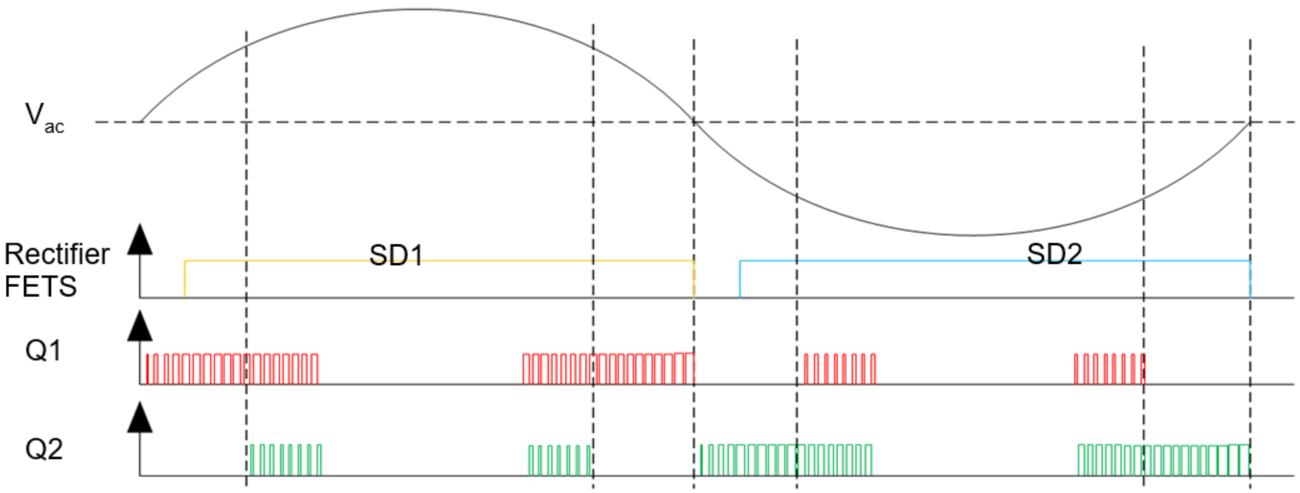


Figure 8. PWM generation mechanism

4 Application setup and control

4.1 Hardware setup

TP-PFC main board is as shown in Figure 9. The control board interface is used to connect the control board, the input interface is used for AC input connection, and the UART interface is used to ommunicate with other boards.

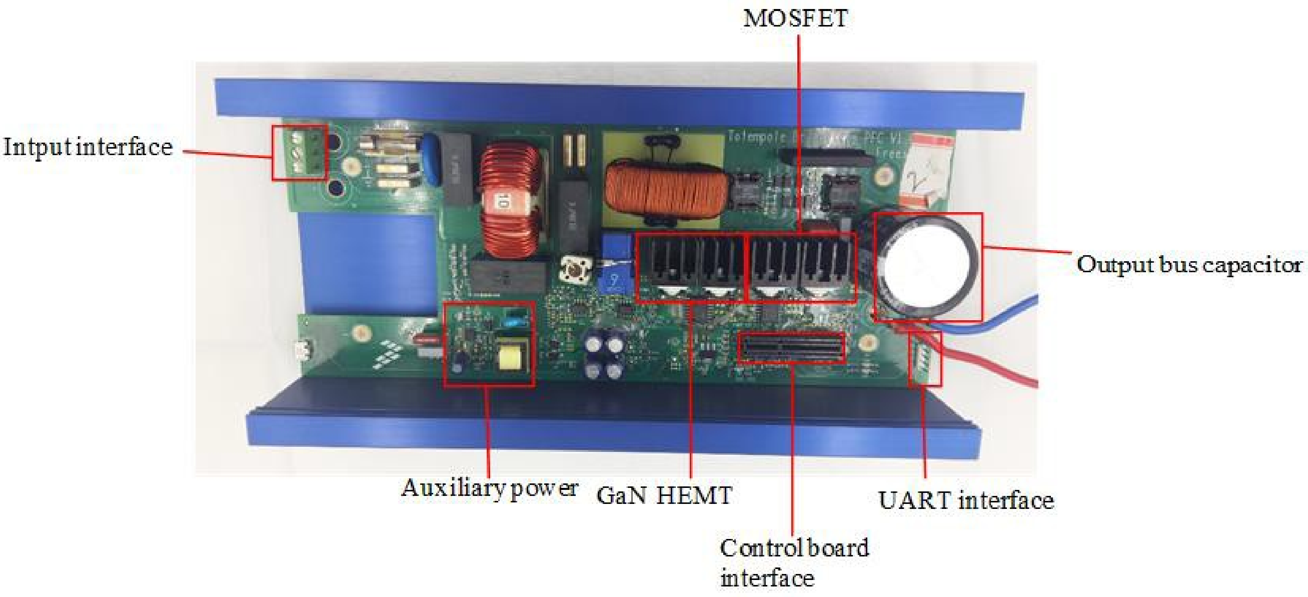


Figure 9. Totem-pole bridgeless PFC main board

To download program to the control chip, you need to connect a debugger (P&E-Multilink) to the SWD port of the control board using a 14-pin cable. The control board can powered from the main board or micro USB interface in the control board. In order to control and monitor the working status of the PFC system, you must connect the isolated USB interface on the control board to the PC through a micro USB cable.

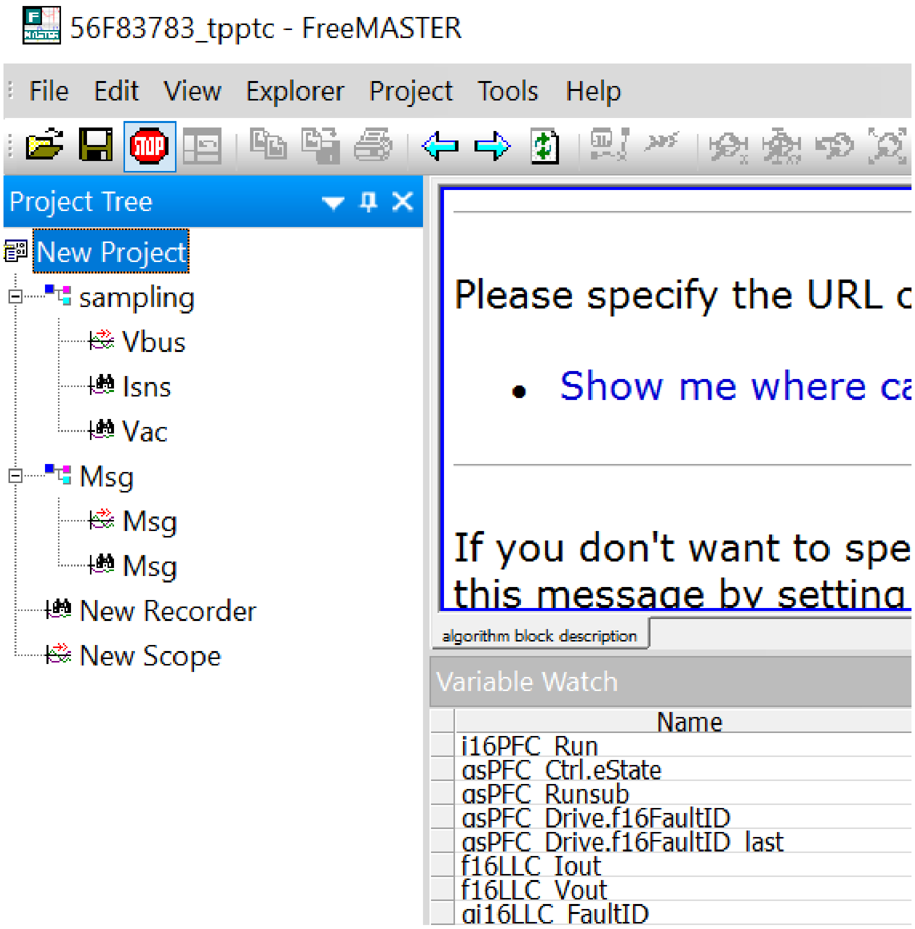
NOTE

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.

4.2 Application control

The application can be controlled remotely from a computer using FreeMASTER. The USB interface for communication on the controller board is galvanically isolated, no additional isolation is required.

Open the the FreeMASTER project file, `56F83783_tppfc.pmp`, located in the project folder. After the project is opened, pressing **Stop** in the main menu to start communication. Control the application ON/OFF by changing `iPFC_Run`.



56F83783_tpptc - FreeMASTER

Please specify the URL c

- Show me where ca

If you don't want to spe
this message by setting

i16PFC_Run
asPFC_Ctrl.eState
asPFC_Runsub
asPFC_Drive.f16FaultID
asPFC_Drive.f16FaultID_last
f16LLC_Iout
f16LLC_Vout
ai16LLC_FaultID

Figure 10. PFC main control panel

5 Testing

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

5.1 System efficiency

Table 1 and Table 2 describe the system efficiency at different voltage.

Table 1. 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 2. 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 11 shows the system efficiency power factor curve.

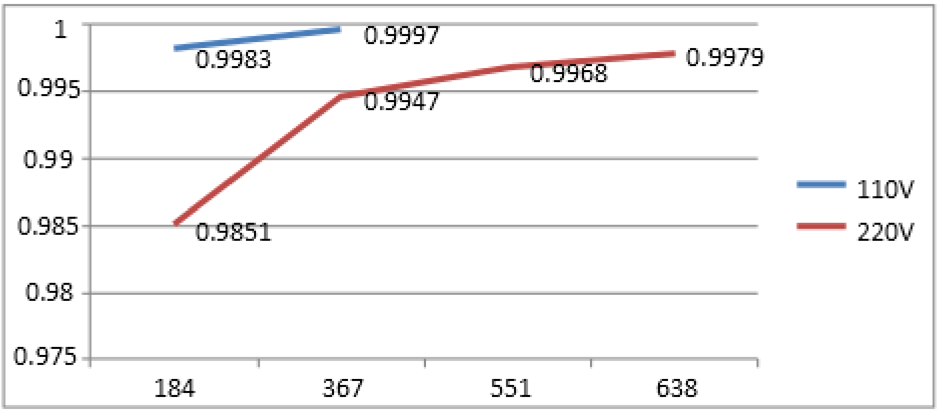


Figure 11. Power factor value curve

Figure 12 shows the Total Harmonic Distortion value curve.

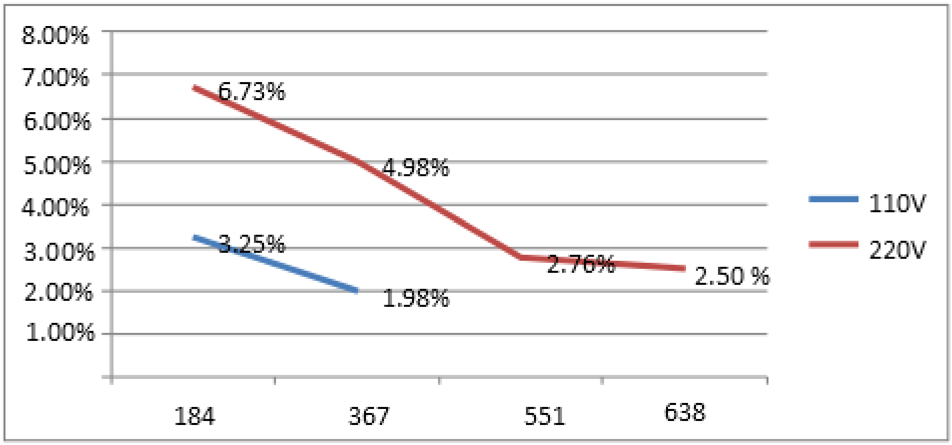


Figure 12. Total Harmonic Distortion value curve of input

Figure 13 shows the efficiency curve.

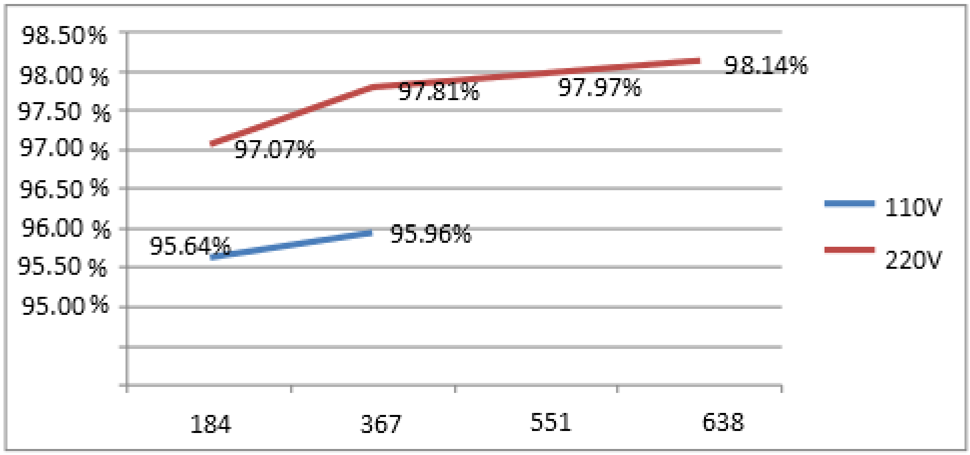
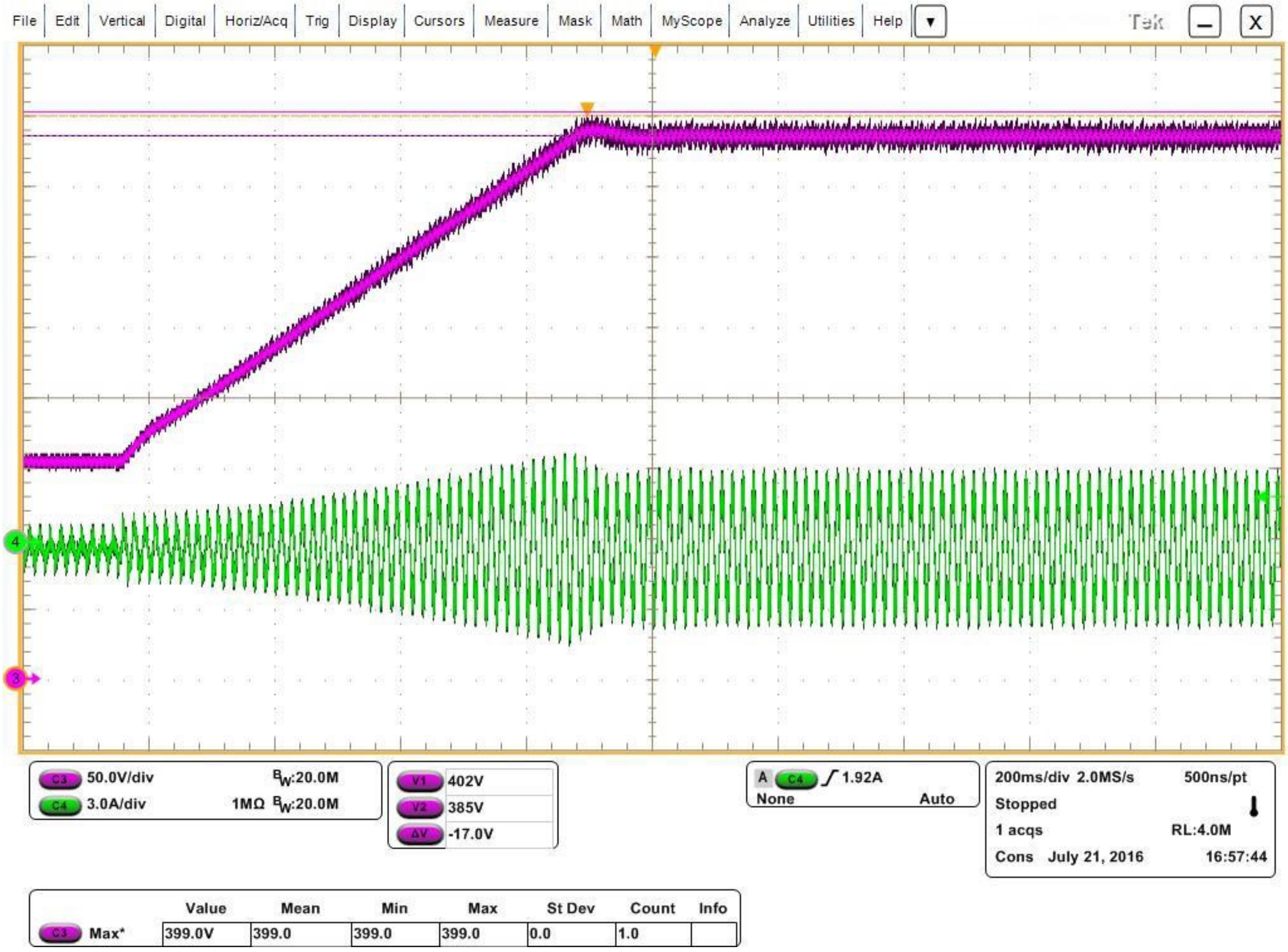


Figure 13. Efficiency curve

5.2 Startup performance



C3 50.0V/div BW:20.0M
C4 3.0A/div 1MΩ BW:20.0M

V1 402V
V2 385V
ΔV -17.0V

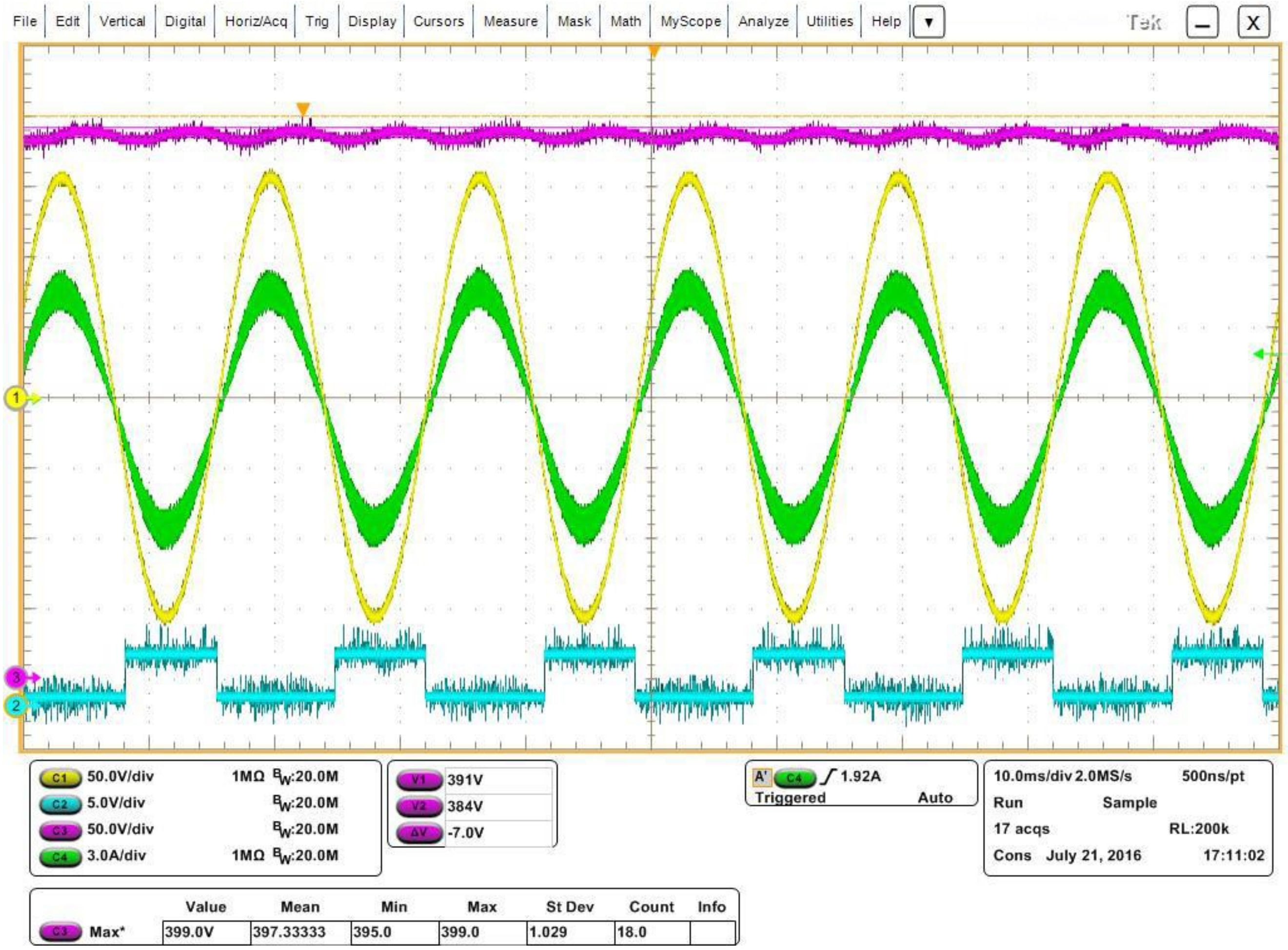
A C4 ∫ 1.92A
None Auto

200ms/div 2.0MS/s 500ns/pt
Stopped
1 acqs RL:4.0M
Cons July 21, 2016 16:57:44

		Value	Mean	Min	Max	St Dev	Count	Info
C3	Max*	399.0V	399.0	399.0	399.0	0.0	1.0	

Figure 14. Start-up of totem-pole bridgeless PFC (CH3:Output voltage;CH4:inductor current)

5.3 Steady state performance



C1 50.0V/div 1MΩ B$_W$:20.0M
C2 5.0V/div B$_W$:20.0M
C3 50.0V/div B$_W$:20.0M
C4 3.0A/div 1MΩ B$_W$:20.0M

V1 391V
V2 384V
ΔV -7.0V

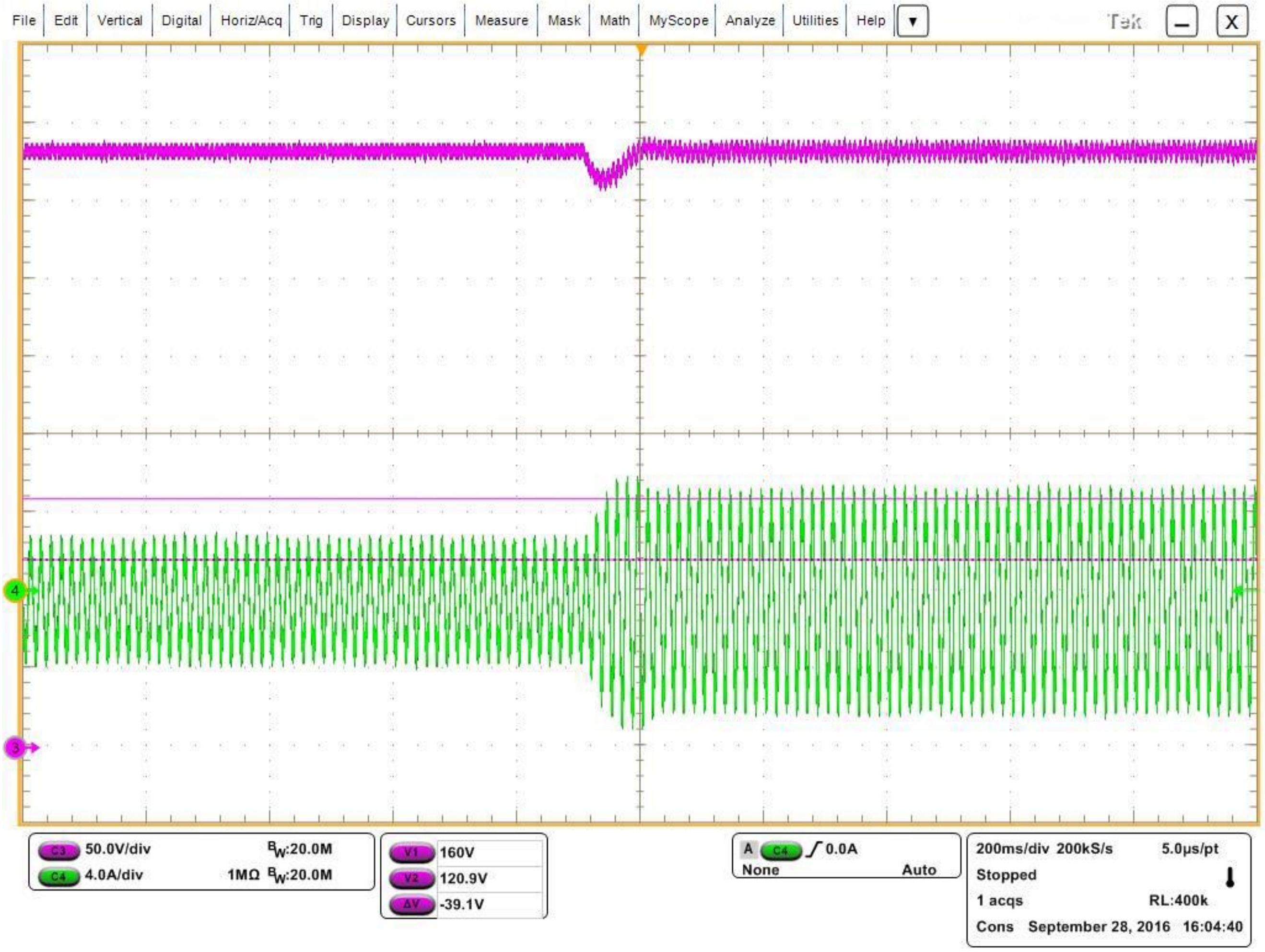
A' C4 ∫ 1.92A
Triggered Auto

10.0ms/div 2.0MS/s 500ns/pt
Run Sample
17 acqs RL:200k
Cons July 21, 2016 17:11:02

	Value	Mean	Min	Max	St Dev	Count	Info
C3 Max*	399.0V	397.33333	395.0	399.0	1.029	18.0	

Figure 15. Steady-state of totem-pole bridgeless PFC (CH1:Input AC voltage; CH2:PWM gate for SD2; CH3:Bus capacitor voltage; CH4:Inductor current)

5.4 Dynamic performance



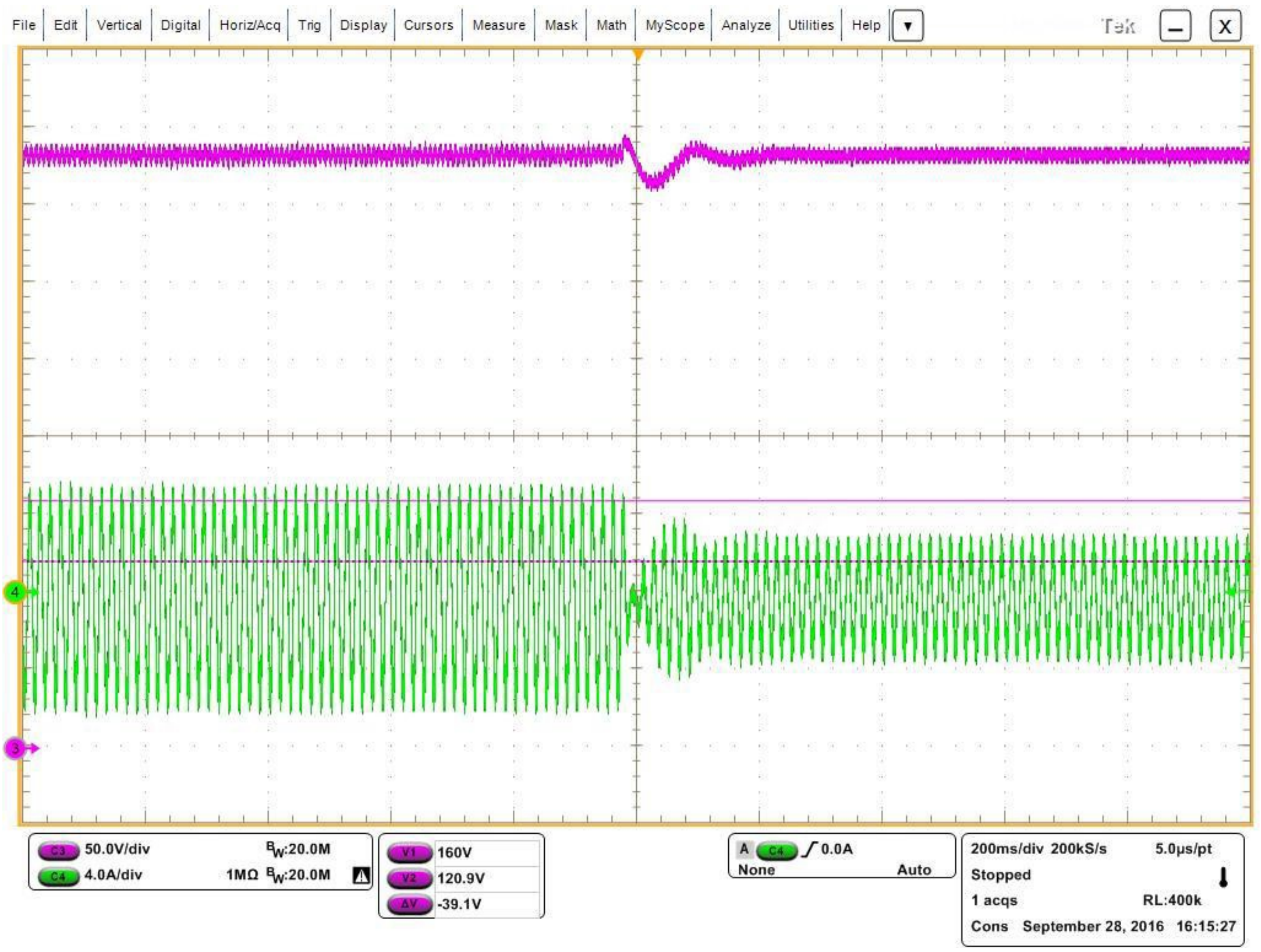
```
C3  50.0V/div            BW:20.0M
C4  4.0A/div     1MΩ     BW:20.0M

V1  160V
V2  120.9V
ΔV  -39.1V

A  C4  ∫ 0.0A
None              Auto

200ms/div  200kS/s      5.0µs/pt
Stopped
1 acqs                  RL:400k
Cons  September 28, 2016  16:04:40
```

Figure 16. Load 50% to 100% transition performance at 110V input; Ch3: Output Voltage; Ch4: Input



C3 50.0V/div Bw:20.0M

C4 4.0A/div 1MΩ Bw:20.0M

V1 160V

V2 120.9V

ΔV -39.1V

A C4 ∫ 0.0A

None Auto

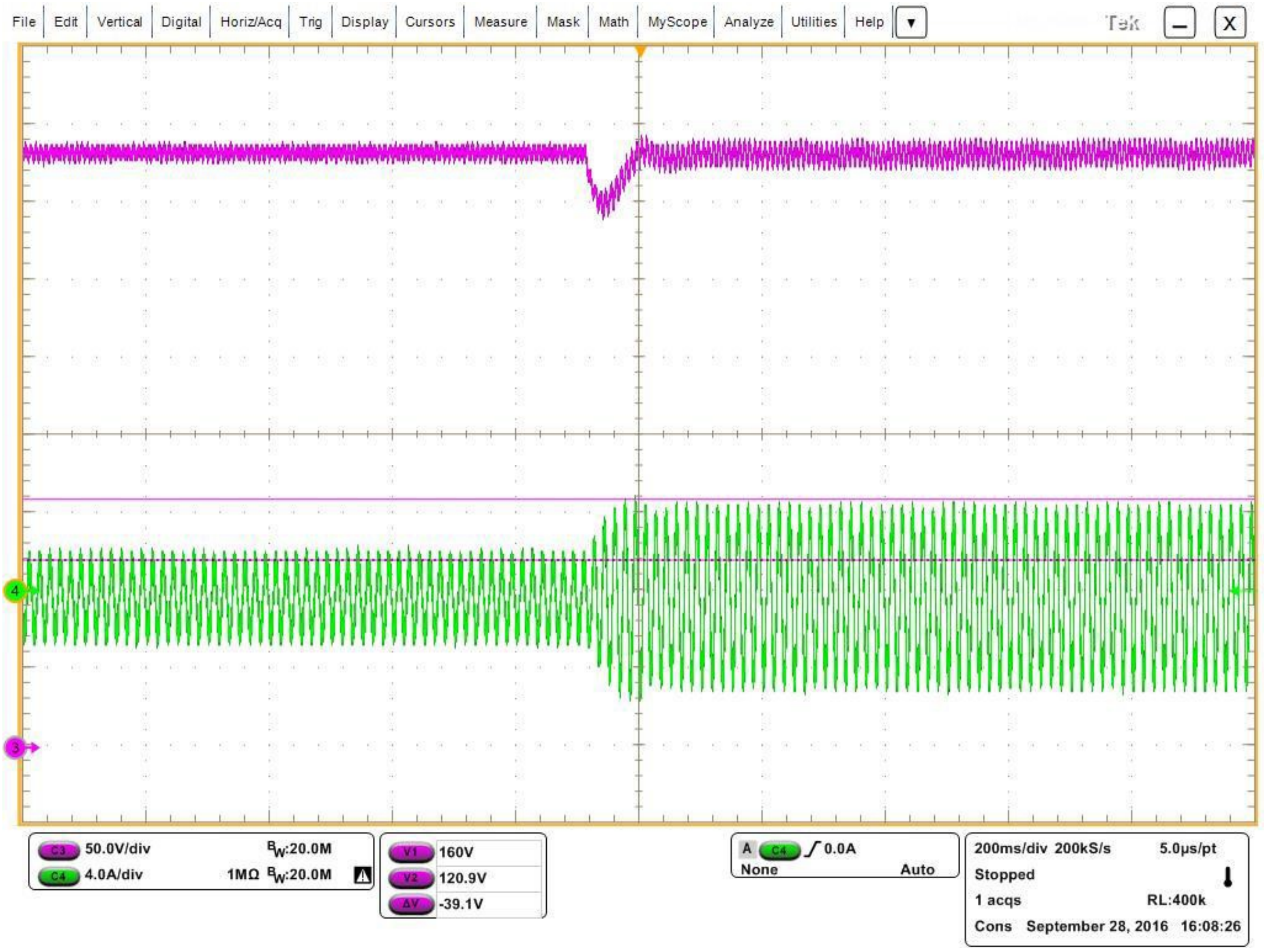
200ms/div 200kS/s 5.0µs/pt

Stopped

1 acqs RL:400k

Cons September 28, 2016 16:15:27

Figure 17. Load 100% to 50% transition performance at 110V input; Ch3:Output Voltage; Ch4 Input current



C3 50.0V/div ᴮw:20.0M

C4 4.0A/div 1MΩ ᴮw:20.0M

V1 160V

V2 120.9V

ΔV -39.1V

A C4 ∫ 0.0A

None Auto

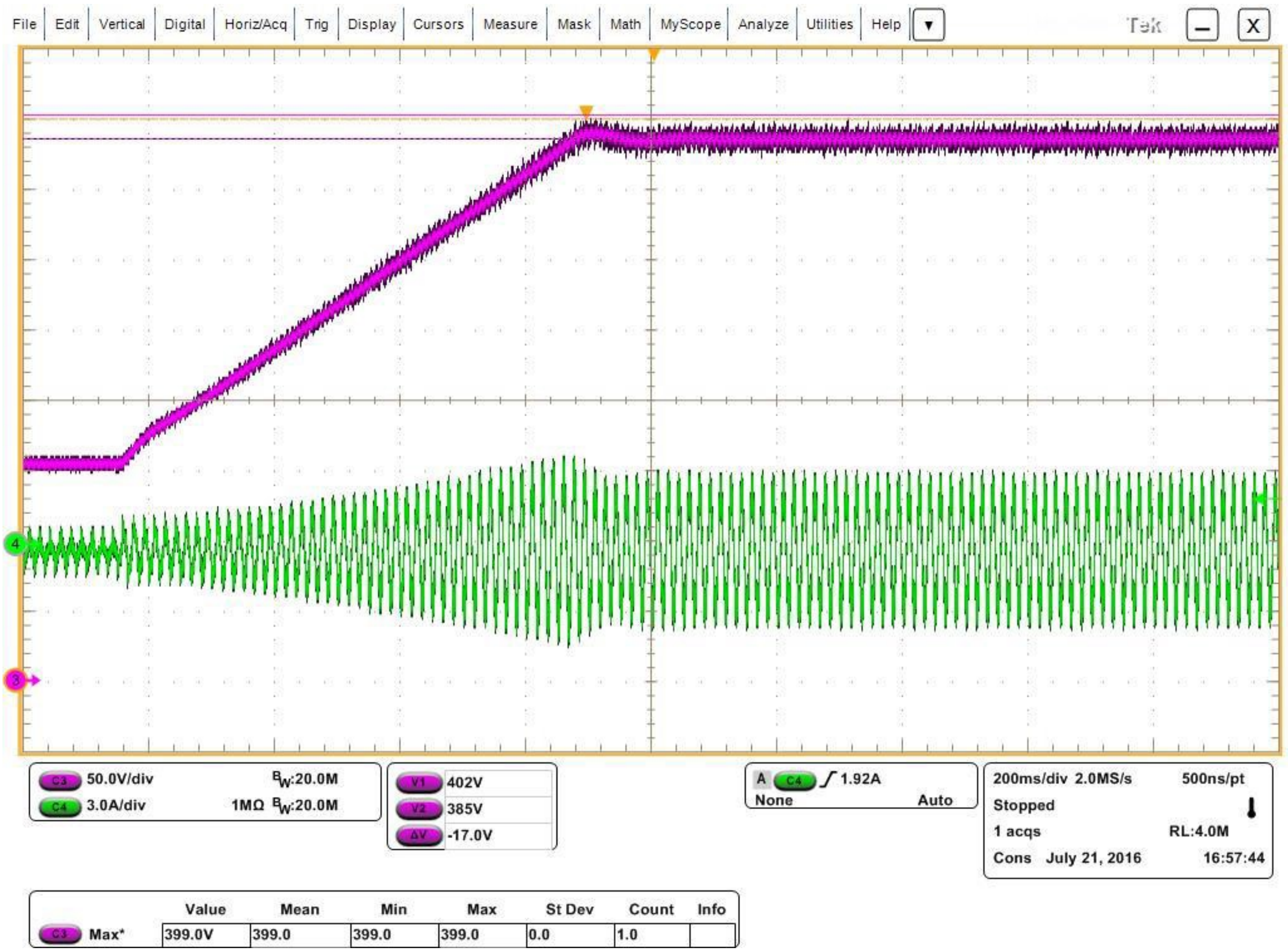
200ms/div 200kS/s 5.0μs/pt

Stopped

1 acqs RL:400k

Cons September 28, 2016 16:08:26

Figure 18. Load 50% to 100% transition performance at 220V input; Ch3: Output voltage; Ch4: Inductor current



C3 50.0V/div BW:20.0M

C4 3.0A/div 1MΩ BW:20.0M

V1 402V

V2 385V

ΔV -17.0V

A C4 ∫ 1.92A

None Auto

200ms/div 2.0MS/s 500ns/pt

Stopped

1 acqs RL:4.0M

Cons July 21, 2016 16:57:44

	Value	Mean	Min	Max	St Dev	Count	Info
C3 Max*	399.0V	399.0	399.0	399.0	0.0	1.0	

Figure 19. Load 100% to 50% dynamic transition performance at 220V input; Ch3: Output Voltage; Ch4: Input Current

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer’s applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

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

Date of release: 10/2019
Document identifier: AN12618