AN12617

LLC Resonant Converter Using MC56F83783

Rev. 0 — 10/2019 Application Note

Contents

1 Introduction

This application note describes the implementation of LLC converter 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 the software design. System description includes system structure, controller features, modulation mode and control loop. Software design includes project file structure, state machine, control timing, peripheral configuration and code runs in RAM. The detailed system operation, hardware design and control loop design is described in *LLC Resonant Converter Design Using MC56F82748* (document DRM172).

2 System description

2.1 System structure

LLC is an isolated buck-boost converter, and the isolation between the primary and secondary side is formed by transformer.

The primary side incorporates the pulse-wave voltage generator, resonant network, isolated drivers, and isolated UART port to communicate with other devices, such as, the front stage PFC converter.

The secondary side incorporates the synchronous rectifier, voltage/current sensing circuitry, output controller, drivers, PM Bus communication, and the DSC controller board.

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

Figure 1 shows the overall system structure.

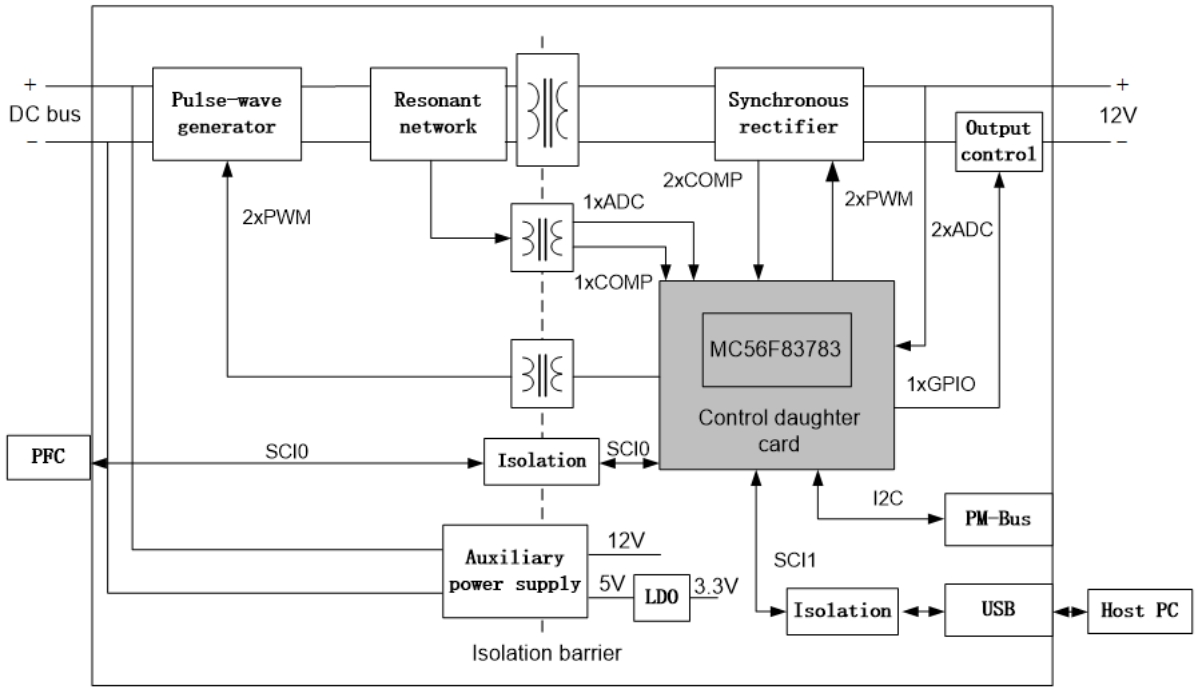


Figure 1. System block diagram

The sensing circuitries are used for sensing DC Bus voltage, resonant current, output voltage, output current and accommodating them to the MCU acceptable voltage level.

The drivers are used for amplification of MCU PWM signals. Isolated drivers for primary side half bridge are implemented by pulse transformer. Non-isolated drivers are used for synchronous rectifier's MOSFET on the secondary side.

The synchronous rectifier rectifies the output voltage to 12 V level and can reduce the conduction losses. The output controller determines the load ON or OFF by software.

The DSC MC56F83783 controller is situated on the control daughter card and connected to the power board via the PCI slot. The control card is powered from the secondary side and it works as the master for the whole application.

The controller is also used to communicate with outside devices. One isolated UART is applied to communicating with the front PFC stage. Another UART to USB conversion is applied to communicating with the host PC for FreeMASTER or firmware updating. One IIC is reserved for the PM Bus network.

2.2 Controller features

The 56F837xx microcontroller is a member of the 32-bit 56800EX core-based 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 Converter (ADC).
- 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 ADCs, 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:
  - 2 x 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 Modulation mode

Most conventional LLC resonant converter applies the Pulse Frequency Modulation (PFM), namely complementary pulse signals with 50 % duty cycle drive the upper and lower arm switch and regulate the output voltage by adjusting the switching frequency. Assuming that the power is transferred mainly by the fundamental component, the First Harmonic Approximation (FHA) method can be used to analyze the LLC resonant converter, the input-to-output voltage gain can be obtained as:

$$\textbf{Eqn. 1}\quad M_{PFM}=\frac{2nV_0}{V_{DC}}=\left|\frac{mf_n^{\,2}}{[(m+1)f_n^{\,2}-1]+jQ_e mf_n(f_n^{\,2}-1)}\right|$$

Where,

- $$Q_e=\frac{1}{R_e}\times\sqrt{\frac{L_r}{C_r}}$$
- $$m=\frac{L_m}{L_r}$$
- $$f_n=\frac{f_{sw}}{f_r}$$
- $$f_r=\frac{1}{2\pi\sqrt{L_rC_r}}$$
- f_{sw} is the operating frequency.

Figure 2 shows the input-to-output gain varies with f_n under certain m and Q_e.

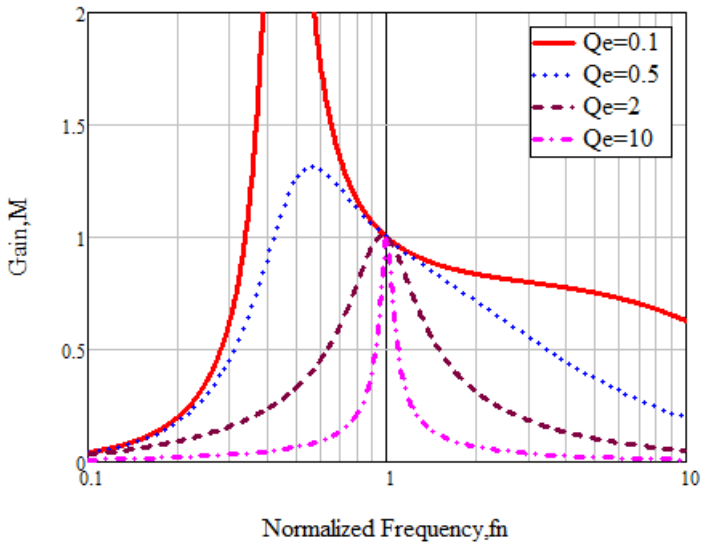


Figure 2. Plots of PFM voltage gain function

However, it is obvious that the variation range of *M* is limited when f_n>1. To meet the requirements of a wide input voltage, improving the switching frequency unlimitedly is not desirable, because of the restriction of component availability, the effect of the transformer parasitic parameters, and so on. Considering the above contradictions, the additional symmetric Pulse Width Modulation (PWM) mode, namely adjusting duty cycle of fixed frequency drive signals to regulate the output voltage, is desired.

When using symmetric PWM, the voltage gain can be obtained as:

$$\textit{Eqn. 2}\quad M_{PWM} = \frac{1-cos(2\pi d)}{\pi} M_{PFM}$$

Assuming a fixed working frequency which is larger than the resonant frequency, the voltage gain function for different values of Q_e under PWM mode is as shown in Figure 3. The voltage gain can be any value between 0 and 1 when the duty cycle varies from 0 to 0.5.

NOTE

In order to ensure ZVS, the duty cycle can't be too small. This reference chooses 0.3 as the minimum duty cycle.

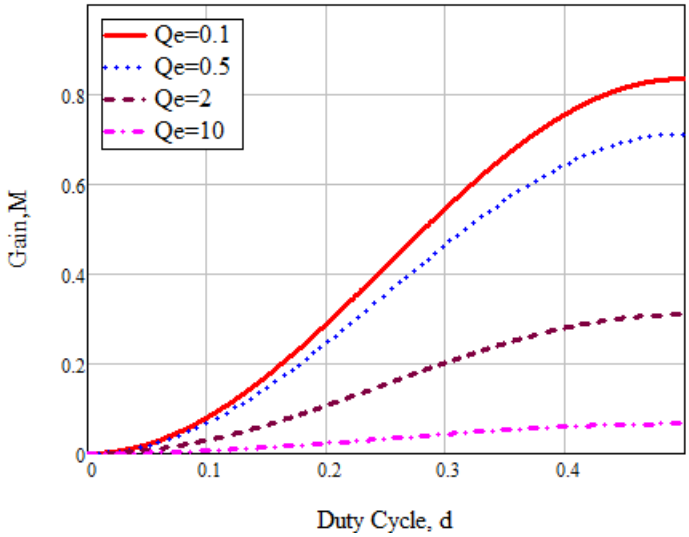


Figure 3. Plots of PFM voltage gain function

When the input voltage is too high or the load is too light, the symmetric PWM mode may still not meet requirements. So the burst mode, namely blocking switching drive signals periodically, is taken. The burst mode control can also improve the light load efficiency of the LLC converter. Figure 4 shows the burst operation processes. When the drive signals are not blocked, the output voltage rises, and conversely declines.

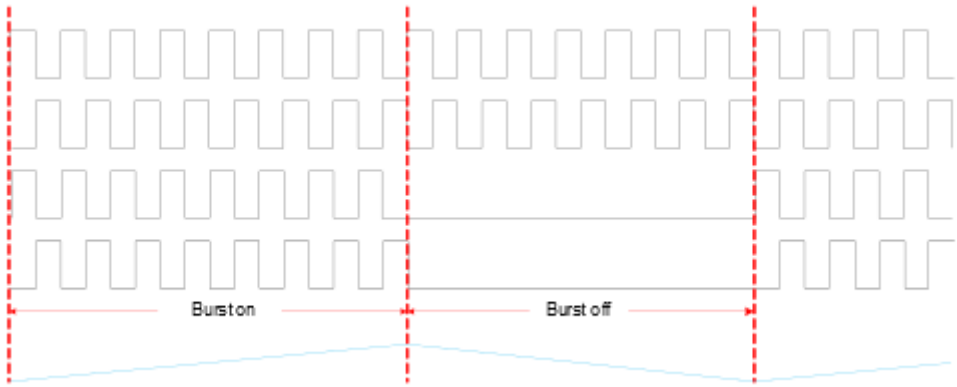


Figure 4. Waveforms of Burst operation

As seen in Figure 2, the voltage gain decreases as the switching frequency increases in the inductive area, while in Figure 3, the voltage gain decreases as the duty cycle decreases. In PFM mode, when the maximum switching frequency can't meet the gain requirement, transfer to PWM mode. If the minimum duty cycle still can't meet requirements, transfer to burst mode. In the closed-loop control, we take the required duty cycle which reflect the current state of the system instead of output voltage as the judgement condition to smoothly switch between different modes. Figure 5 shows the condition of each mode and the transition between them. A hysteresis is taken in burst mode to avoid frequent switching.

As shown in Figure 5,

- T_{max} is the allowed maximum switching period.
- T_{min} is the allowed minimum period.
- D_{off} is the minimum allowed duty cycle.

- D_{on} and D_{off} is the upper and lower limitation of hysteresis.

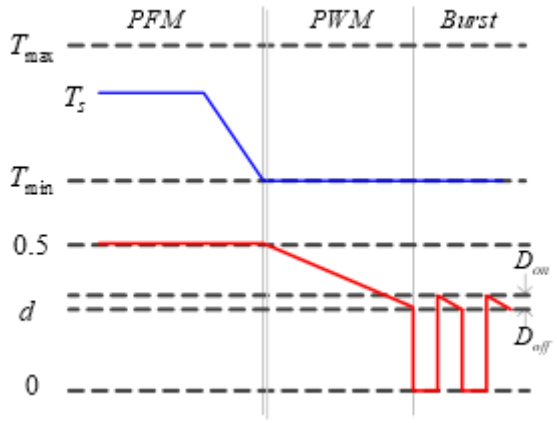


Figure 5. PFM, PWM and burst modes

2.4 Control loop

In the server, the battery charger and many other industrial applications, and the power supplies are required to have both constant voltage and overload current limiting control. So the dual outer loop control scheme is applied. It includes the inner primary current loop and outer output voltage or out current loop determined by the output current. The inner average current control loop is used to eliminate the difference in small-signal characteristics among different static operation, which is equivalent to modify the object model, to adapt to the wide range requirement. Figure 6 shows the control scheme.

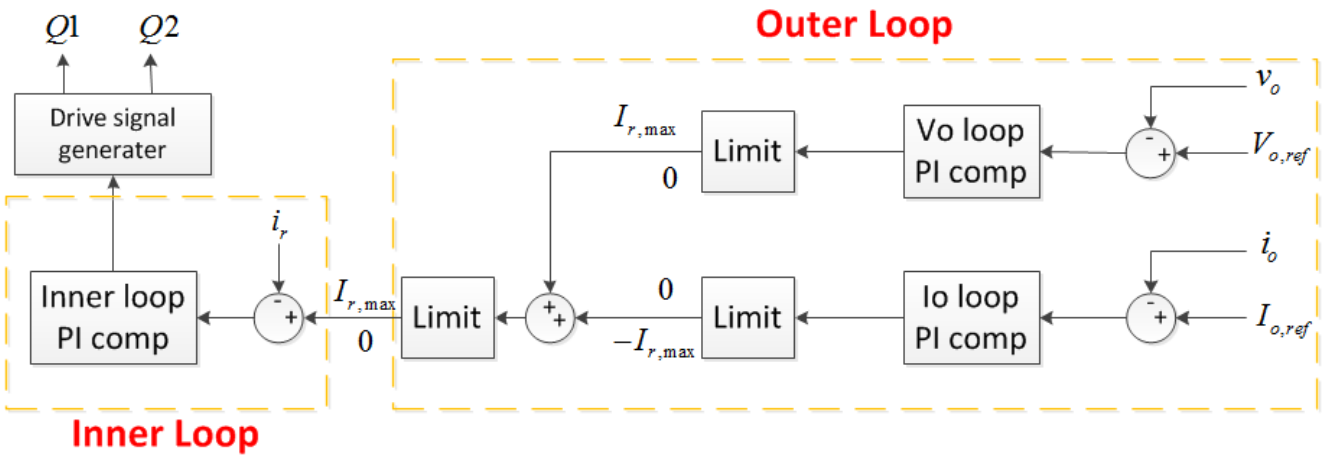


Figure 6. LLC control scheme

With this control scheme, LLC can work under three states:

- S1 constant voltage state: System mainly working under this mode to maintain constant 12V output. Output voltage equals its reference and output current smaller than its reference, negative deviation causes the PI result of output current regulator reaches its upper limit which is zero. So the output current loop is calculated but has no effect on the control, the control signal is only related to output voltage.
- S2 constant current state: When output current reaches the current limitation threshold, output voltage decrease and constant current loop takes effect. Output current equals its reference and output voltage smaller than its reference, continuing positive deviation causes the PI result of the output voltage regulator has been equals its upper limit. So even if the voltage loop is calculated, only the change of output current takes effect.

- S3 transient state: It is the intermediate state when switching between S1 and S2 and both the voltage and current loop take effect in this state. If the load exceeds the rated load in constant voltage mode, system will change from constant voltage state to constant current state. In this process, the PI result of voltage loop increases and the PI result of current loop decreases. If the load reduces to less than ratings, system changes from constant current loop to constant voltage loop. In this process, the PI result of voltage loop decreases and the PI result of current loop increases.

In summary, the concurrent outer loop smooth transition achieves both constant output voltage and overload current limitation. The resonant current inner loop ensures the good performance in all range with the same controller. Figure 7 shows the output I-V curve.

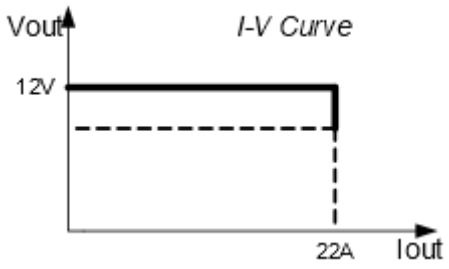


Figure 7. Dual outer loop I-V curve

3 Software design

This chapter describes the software design of LLC resonant converter using 56F83783, including software structure, configuration of the DSC peripherals, control timing and implementation of codes run in ram.

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 CodeWarrior11.1 generated Bareboard project without configuration tool, such as Process Expert and Quick Start. The overall project files structure is as shown in Figure 8.

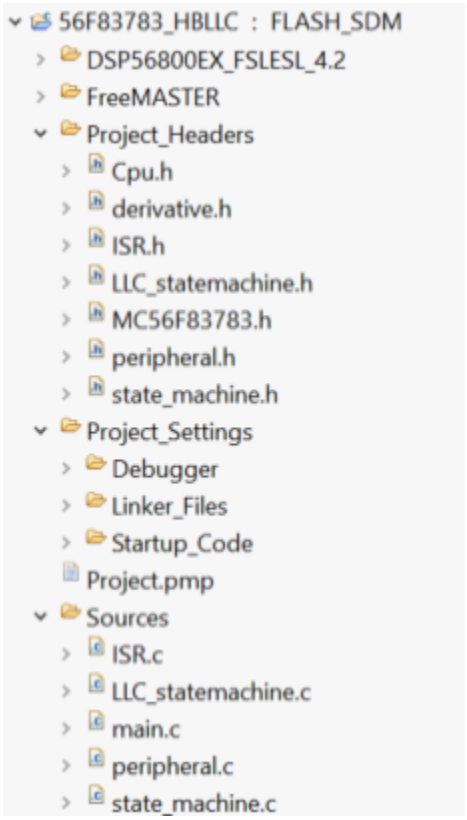


Figure 8. Project file structure

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

1. `..\DSP56800EX_FSLESL_4.2`: FSLESL libraries.
2. `...\FreeMASTER`: FreeMASTER drive and configuration files, to 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 main `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\LLC_statemachine.c`: The LLC state function `C` source file.
9. `..\Project_Headers\ISR.h`: The interrupt vector header file.
10. `..\Project_Headers\peripheral.h`: The application peripheral configuration head file.
11. `..\Project_Headers\statemchine.h`: The common statemachine header file.
12. `..\Project_Headers\LLC_statemachine.h`: The application definition head 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.

$$\textbf{Eqn.3} \qquad \text{Frac value} = \frac{\text{actual value}}{\text{quantization range}}$$

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

Figure 9 shows how this application uses the state machine to control the system flow. After resetting the DSC to configure 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 waiting for the Run command, and the PWM output is disabled.
- **RUN**: The system is running, the run sub-state is called, and the PWM output is enabled and can be stopped with the Stop command.
- **FAULT**: The system faced a fault condition and the PWM output is disabled.

After the parameters initialization, the application state machine continues into the **STOP** state. In the **STOP** state, different control algorithms can be configured by FreeMASTER online:

- Open loop.
- Voltage single loop: The Constant output voltage control with a single loop.
- Voltage/current dual loops: The output voltage control with the voltage outer loop and the primary current inner loop.
- CCM/CVM outer loops: The concurrent output voltage or the current control with the primary current inner loop.

Besides, the software regularly checks the `LLC_run` instruction to decide whether to start up. When the `LLC_run` command is set, the application state machine continues into the **RUN** state. In the **RUN** state, the controller starts to take effect to achieve constant output voltage or limiting output current. Besides, the `LLC_run` instruction is also regularly checked in the **RUN** state. When the `LLC_run` command is cleared, the application state machine goes back to the **STOP** state and waits for the `LLC_run` command to be set again.

The fault detection executes under all states. 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.

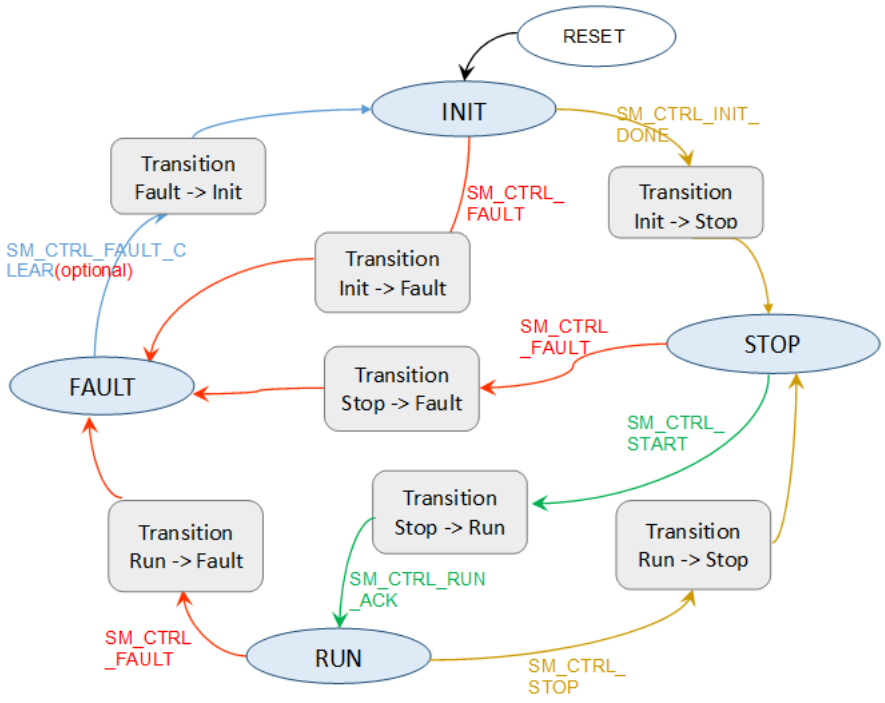


Figure 9. Application main state machine

The **RUN** state is divided into two sub-state according to LLC application, as shown in Figure 10. When application state machine transits from **STOP** to **RUN**, the system first comes into the soft-start sub-state and starts LLC resonant converter to run with maximal switching frequency and changeable duty cycle which is determined by current output voltage. The duty cycle is increased following the ramp until it reaches 50%. After that, the switching frequency is decreased following the ramp until a certain output level is reached. Then the sub-state continues into the normal state and the closed-loop controller works to calculate the desired duty cycle and switching frequency. In this sub-state, the reference output voltage is increased following ramp until it reaches desired value and the load is connected when output voltage reaches its final reference.

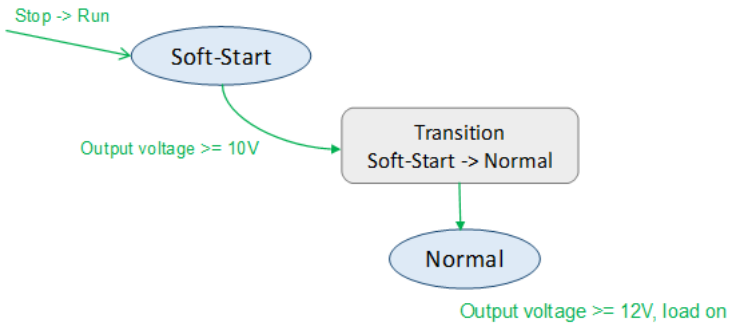


Figure 10. Application run sub-state

The application main state machine is declared in `state_machine.c` and `state_machine.h`. The state machine is called in the background loop. The sub-state functions and corresponding transition functions are defined in `LLC_statemachine.c` and are called when the system is in the **RUN** state periodically in interrupt routine.

3.4 Control timing

The LLC software consists of two periodical interrupts and a background loop. The first periodical interrupt `PWM_Trigger_ISR()` is driven by the VAL0 compare event from eFlexPWM SM0. This routine is configured for higher priority to execute the control loop calculation and get the desired switching signal. The ADC conversion is triggered at the same time, so register push and ADC sampling operate simultaneously which leaves more time for algorithm calculation. Besides, center aligned PWMs ensure this sampling point not affected by the switching operation and can get precise sample result. `PIT0_ISR()` is called every 1ms by the roll-over event of PIT0. This routine is used to create a 1ms time base which can be used as software timer for protection and so on. The background loop runs in a never ending loop. It includes the application state machine and communication with PC (FreeMASTER).

As described above, `PWM_Trigger_ISR()` is a time critical routine. The FSLESL is applied to minimize the total execution time. However, because of the high switching frequency, this routine still can't execute every switching period under each switching frequency. In order to achieve more precise control, the application tries to increase the effective controlling frequency by two ways:

- Run the time critical algorithm in RAM which is described in detail in the following section.
- Change the controlling frequency according to current switching frequency.

So we need different eFlexPWM sub-modules to handle controlling frequency and PWM generating. MC56F83783 eFlexPWMA module has four sub-modules SM0-4. SM0 is used to determine the controlling frequency and synchronize the ADC sample. SM1&2 are used to construct PWM signals need by switch drivers. The synchronization between different sub-modules is achieved by master reload signal forcing initialization. As measurement, the maximum time from PWM event trigger to the end of interrupt is 8.2 us, which includes the outer output current limitation loop, output voltage loop and inner primary voltage loop. So the application takes 10 us as the minimum control period, which means when the switching frequency is lower than 100 kHz the controller is executed every PWM period. So the modulo count value of SM0 and SM2 are equivalent. When the switching frequency is between 100 kHz and 200 kHz, the controller is executed every two PWM periods. So the modulo count value of SM0 is two times of SM2. When the switching frequency is between 200 kHz and 300 kHz, the controller is executed every three PWM periods. So the modulo count value of SM0 is three times of SM2.

As mentioned in last section, LLC **RUN** state is divided into two sub-states. The control timing for **Soft-Start** state is shown in Figure 11. The start-up frequency f_{st} is set to 250 kHz which lets the controller execute every three PWM periods. The control timing for **Normal** state is shown in Figure 12. When switching frequency is lower than 200 kHz, duty cycle is fixed at 50%. LLC converter works under PFM mode. When desired frequency is higher than 200 kHz, switching frequency is fixed at 200 kHz and LLC converter works under symmetric PWM mode.

The controlling frequency change is implemented in `LLC_PWM_UPDATE()` function.

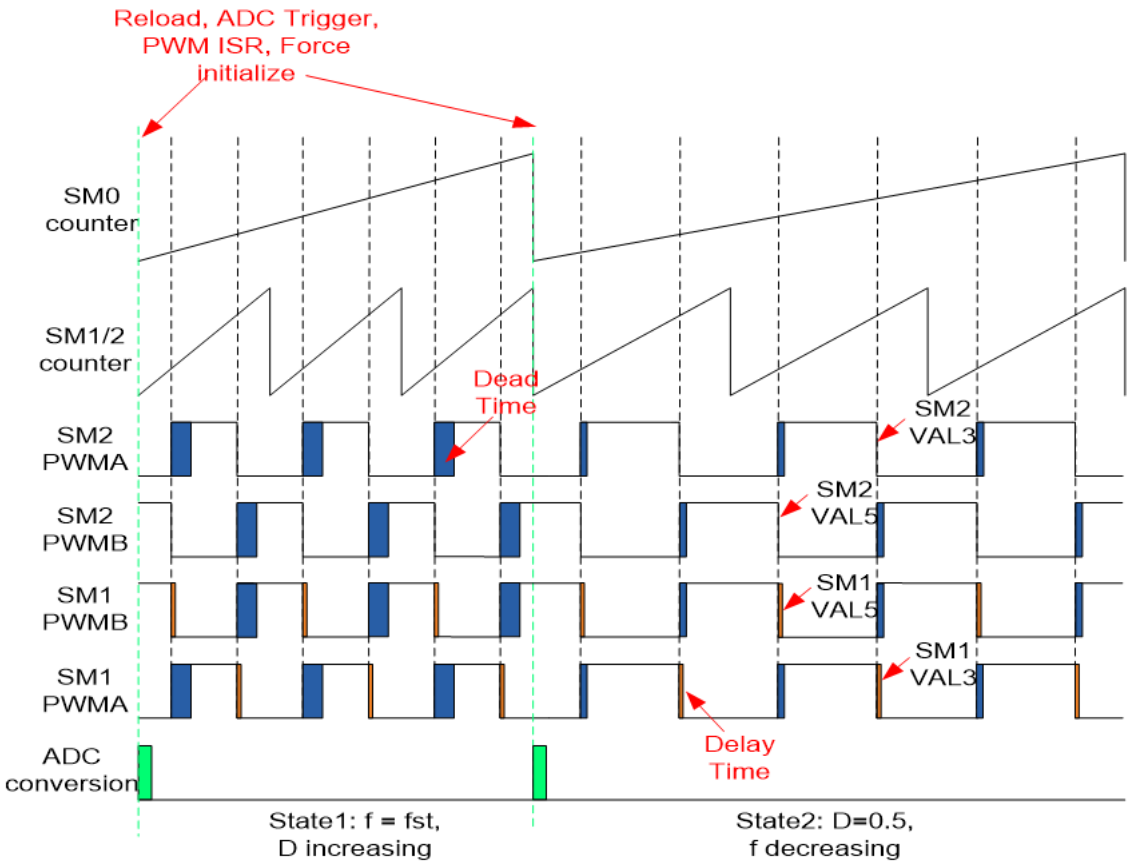


Figure 11. Control timing for Soft-Start state

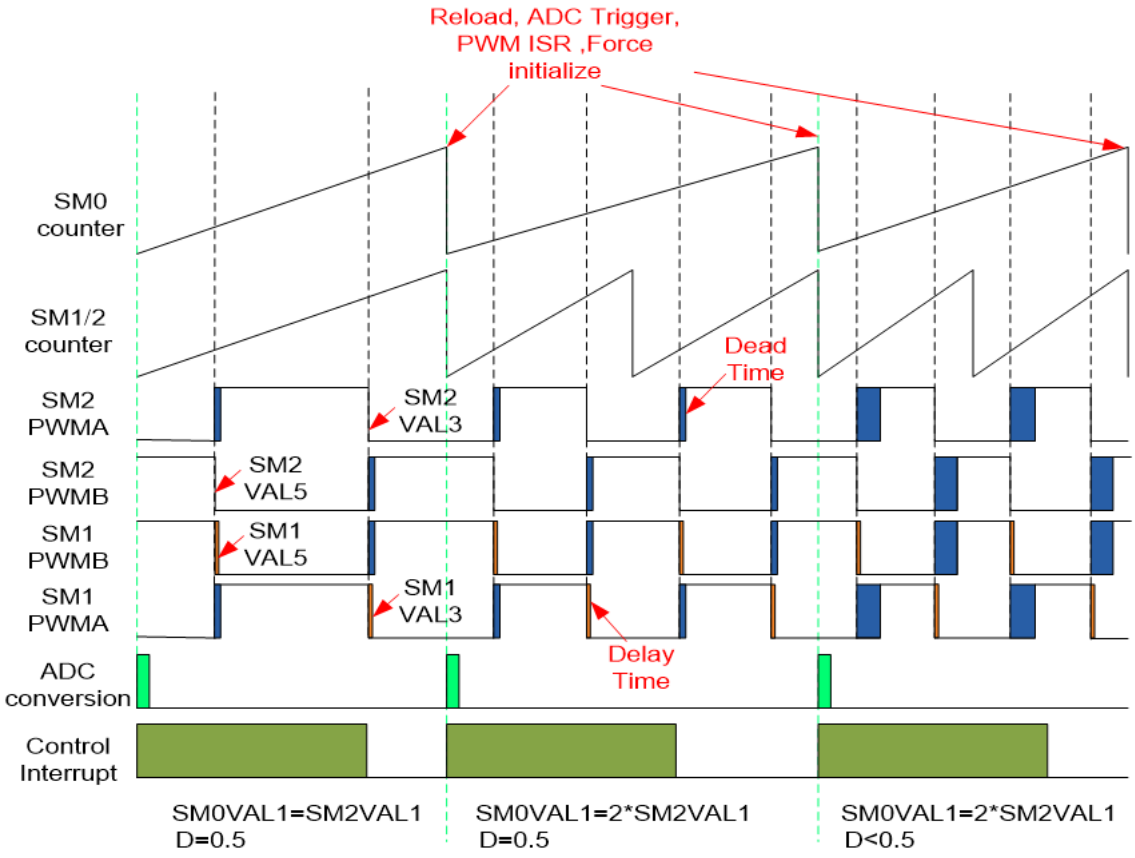


Figure 12. Control timing for Normal state

3.5 Fault protection

The hardware protection is over-current protection of resonant current, which is generated by the build-in high speed comparator A. And the fault threshold is set to 4.2 A.

The software protection includes resonant current over-current protection, output under-voltage and over-voltage protection, output over-current protection. Under the CCM/CVM outer loops mode, the output current is limited so the output under-voltage will take effect if the load is too heavy. Under other conditions, the output current protection takes effect and the overload duration is considered. 50% overload duration is now set to 5 ms and 20% overload duration is 20 ms, which can be easily modified as needed.

3.6 Peripheral configuration

This application uses dedicated peripherals for LLC algorithm implementation and communication. The peripherals used in the application are: ADC, eFlexPWM, XBAR, EVTG, PIT, 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 synchronize with eFlexPWM sub-module 0 via Trigger 0. The trigger signals connection between eFlexPWM and ADC is provided by a cross bar switch module. The ADC A samples resonant current and output current, ADC B samples DC bus voltage and output voltage.

3.6.2 Pulse width modulator eFlexPWMA

- The eFlexPWMA sub-module 0 is used for timing control and no PWM signal is generated from this sub-module. Register VAL0 is used to generate trigger signal for ADC and cause interrupt to handle control algorithm. Modulo count value VAL1 varies according to current switching frequency.
- The eFlexPWMA sub-module 2 generates two complementary PWM signals. The PWM outputs have variable frequency and duty cycle and route to XBARB for the generation of half bridge MOSFET drive signals. The variation of duty cycle is implemented by change of dead time generated by software. Due to NanoEdge placement this module can generate duty cycle and frequencies with higher resolution up to 312.5 ps. The sub-module 2 is synchronized with sub-module 0 by master reload forcing initialization.
- The eFlexPWMA sub-module 1 has the same modulo count value with sub-module 2 and generates two complementary signals. The PWM outputs have the variable frequency as sub-module2 and route to XBARB for the generation of synchronous rectification MOSFET drive signals. During operation above resonant frequency the synchronous rectification MOSFETs can switch off later than half bridge MOSFET. This means the falling edge comes later in comparison with the primary PWM signals.

Therefore, the PWM outputs of sub-module 1 are same as sub-module 2 except the falling edge. During operation below resonant frequency the earlier shutdown is achieved by hardware detection. The sub-module 1 is also synchronized with sub-module 0 by master reload forcing initialization, so it is synchronized with sub-module 2.

3.6.3 High speed comparator HSCMPA

The build-in high speed comparator A is used to detect over current condition of the primary resonant current. One of the comparator's inputs is the measured resonant current, the other is the fault threshold set by built-in 8-bit DACA.

3.6.4 High speed comparator HSCMPB&D

The build-in high speed comparator B&D are used to detect the voltage flip of transformer for synchronous rectification MOSFET ON/OFF control. The transformer voltage thresholds are set by built-in 8-bit DACB&D.

3.6.5 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 12, ADC trigger: from XBARIN28, PWMA0 trigger0
- Channel 29, PWMA fault0: from XBARIN12, CMPA_OUT

3.6.6 Event generator EVTG

Each EVTG includes two AOI module which share the four associated inputs. An XBAR switch is typically used to select the EVTG inputs from available peripheral outputs and GPIO signals.

Drive signals for half bridge MOSFETs can output directly or connect to internal-peripheral crossbar switch (XBAR). Precise synchronous rectification drive signals combine the hardware detection signal and eFlexPWM SM1 PWMs through AOI **AND** function. Detected hardware signal is the voltage of transformer and SM1 PWMs is a little delay to half bridge MOSFET PWMs. The combination and delay ensure the correctness and accuracy of synchronous rectification drive signals which minimize the loss of diode rectification. Fault signal is also needed to close the synchronous rectification MOSFETs.

The application configuration is as follows:

- The EVTG0 input A (`XBAROUT47`) is linked to `CMPD_OUT` (`XBARIN15`)
- The EVTG0 input B (`XBAROUT48`) is linked to PWMA1 trigger1 (`XBARIN23`)
- The EVTG0 input C (`XBAROUT49`) is linked to `CMPA_OUT` (`XBARIN12`)
- The EVTG1 input A (`XBAROUT51`) is linked to `CMPB_OUT` (`XBARIN13`)
- The EVTG1 input B (`XBAROUT52`) is linked to PWMA1 trigger0 (`XBARIN22`)

- The EVTG1 input C (XBAROUT53) is linked to CMPA_OUT (XBARIN12)
- The EVTG2 input A (XBAROUT55) is linked to PWMA2 trigger0 (XBARIN24)
- The EVTG2 input B (XBAROUT56) is linked to PWMA2 trigger1 (XBARIN25)
- The EVTG2 input C (XBAROUT57) is linked to CMPA_OUT (XBARIN12)

Figure 13 shows the drive signal generation logic. The SW PWM output control is used to ON/OFF the PWM outputs flexibly. PWM_1&2 are the half bridge MOSFET drive signals and PWM_SR1&2 are the synchronous rectification drive signals. All combinations are implemented by hardware which ensures the timeliness of drive signal and simplifies the software design.

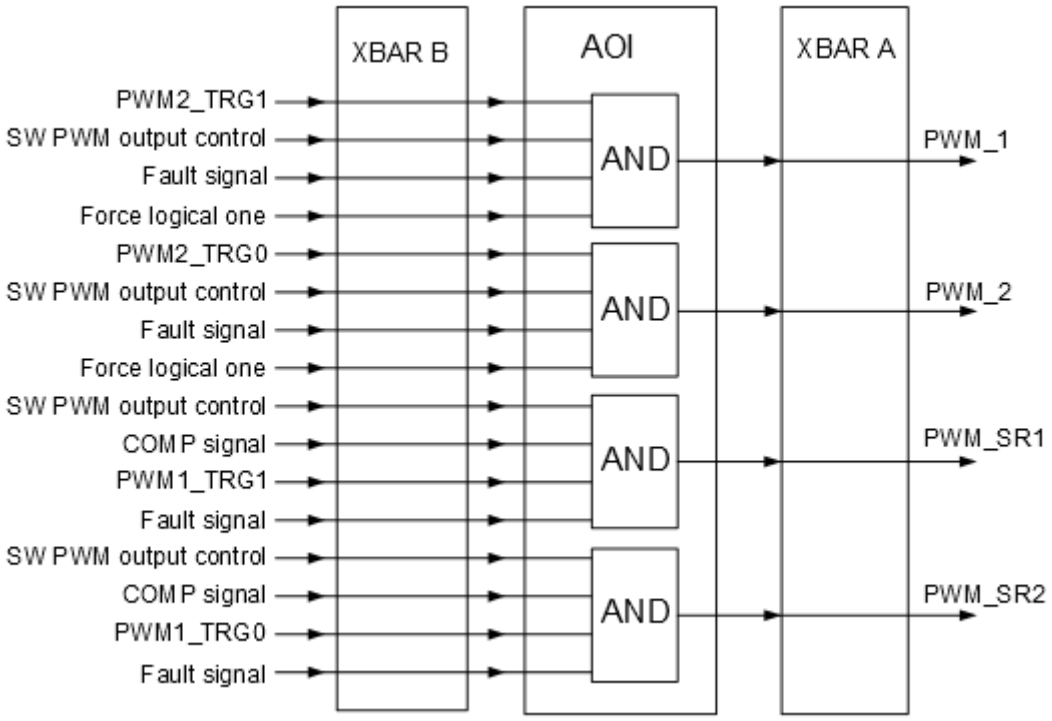


Figure 13. Drive signal generation logic

3.6.7 Periodic interrupt timer PIT0

The periodic interrupt timer 0 is configured to generate a periodical interrupt every 1 ms, which creates a time base for use.

3.6.8 Serial communication interface SCI0 & SCI1

The serial communication interface SCI0 is used for communication with front stage, it can exchange the information with PFC side, 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 an SCI USB converter on the board, so the communication with host PC finally goes via USB interface.

3.7 Code runs in RAM

MC56F83783 core can operate up to 100 MHz when code operates from on-chip RAM AND flash cache buffer. Because of limited cache size, the critical routine PWM_Trigger_ISR() is configured to run 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:

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

```
#pragma define_section CODES_IN_RAM "codesInRam.text"  RX
#pragma section CODES_IN_RAM begin
#pragma interrupt saveall
void PWMA0_trig0_isr()
{ … }
#pragma section CODES_IN_RAM end
```

- Reside the defined code section in corresponding section segment with Linker Command File (LCF). And remember to reconfigure the address to prevent overlap. In this application, `rtlib.text` and library are also included in this section.

```
.ramFunctions : AT(F__pROM_code_start)
{
   . = . + __data_size;
   F__pRAM_code_start = .;

   * (rtlib.text) # suggest to put rtlib function into RAM if ISR performance needs to be improved
   * (codesInRam.text)
   GFLIB_SDM.lib  (.text)
   PCLIB_SDM.lib  (.text)
   MLIB_SDM.lib   (.text)
   GDFLIB_SDM.lib (.text)
   # save address where for the data start in pROM
   . = ALIGN(2);
   F__pRAM_code_end = .;
   __ramfunctions_size = F__pRAM_code_end - F__pRAM_code_start;
} > .p_ramSpace
```

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

```
F_Livt_size     = __ramfunctions_size;
F_Livt_RAM_addr = F__pRAM_code_start;
F_Livt_ROM_addr = F__pROM_code_start + __data_size;

F_xROM_to_xRAM   = 0x0000;
F_pROM_to_xRAM   = 0x0001;  # Enable loading data initial value to corresponding variables before
entering main function
F_pROM_to_pRAM   = 0x0001;
```

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

4 Application setup and control

4.1 Hardware setup

The complete hardware setup is shown in Figure 14.

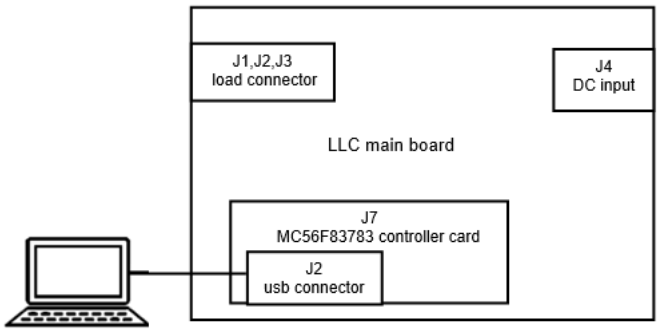


Figure 14. Hardware setup

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.

To control and monitor the working status of the LLC system, you must connect the isolated USB interface on the control board to the PC through a micro USB cable.

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, `Project.pmp`, located in the project folder. After the project opens, press **Stop** in the main menu to start the communication. Different control algorithm can be chosen under stop state and configured in FreeMASTER online:

- **Open loop**: The system operates in constant frequency.
- **Voltage single loop**: Constant output voltage control with single loop.
- **Voltage/current dual loops**: The output voltage control with voltage outer loop and primary current inner loop.
- **CCM/CVM outer loops**: The concurrent output voltage or current control with primary current inner loop.

Then, the LLC operation is started by change `LLC_Run` to **1**.

5 Testing

This section provide the testing results of LLC resonant converter.

5.1 System efficiency

Table 1, Table 2, and Table 3 show the system efficiency at different voltages.

Table 1. System efficiency at VIN=380 V

	Iout (A)	Vout (V)	Pout (W)	Iin (A)	Vin (V)	Pin (W)	Efficiency
5%	1	12	12	0.047	380.1	17.8647	67.17%
10%	2	12	24	0.079	380.1	30.0279	79.93%
20%	4	12	48	0.148	380.1	56.2548	85.33%
30%	6	12	72	0.216	380.1	82.1016	87.70%
40%	8	12	96	0.28	380.1	106.428	90.20%

Table continues on the next page...

Table 1. System efficiency at VIN=380 V (continued)

	Iout (A)	Vout (V)	Pout (W)	Iin (A)	Vin (V)	Pin (W)	Efficiency
50%	10	12	120	0.345	380.1	131.1345	91.51%
60%	12	12	144	0.411	380.1	156.2211	92.18%
70%	14	12	168	0.478	380.1	181.6878	92.47%
80%	16	12	192	0.546	380.1	207.5346	92.51%
90%	18	12	216	0.615	380.1	233.7615	92.40%
100%	20	12	240	0.685	380.1	260.3685	92.18%

Table 2. System efficiency at VIN=330 V

	Iout (A)	Vout (V)	Pout (W)	Iin (A)	Vin (V)	Pin (W)	Efficiency
5%	1	12	12	0.055	330	18.15	66.12%
10%	2	12	24	0.092	330	30.36	79.05%
20%	4	12	48	0.166	330	54.78	87.62%
30%	6	12	72	0.24	330	79.2	90.91%
40%	8	12	96	0.315	330	103.95	92.35%
50%	10	12	120	0.391	330	129.03	93.00%
60%	12	12	144	0.467	330	154.11	93.44%
70%	14	12	168	0.543	330	179.19	93.76%
80%	16	12	192	0.621	330	204.93	93.69%
90%	18	12	216	0.698	330	230.34	93.77%
100%	20	12	240	0.776	330	256.08	93.72%

Table 3. System efficiency at VIN=360 V

	Iout (A)	Vout (V)	Pout (W)	Iin (A)	Vin (V)	Pin (W)	Efficiency
5%	1	12	12	0.051	360	18.36	65.36%
10%	2	12	24	0.1	360	36	66.67%
20%	4	12	48	0.157	360	56.52	84.93%
30%	6	12	72	0.223	360	80.28	89.69%
40%	8	12	96	0.291	360	104.76	91.64%
50%	10	12	120	0.36	360	129.6	92.59%
60%	12	12	144	0.43	360	154.8	93.02%
70%	14	12	168	0.501	360	180.36	93.15%
80%	16	12	192	0.572	360	205.92	93.24%
90%	18	12	216	0.644	360	231.84	93.17%
100%	20	12	240	0.716	360	257.76	93.11%

Figure 15 shows the system efficiency curve.

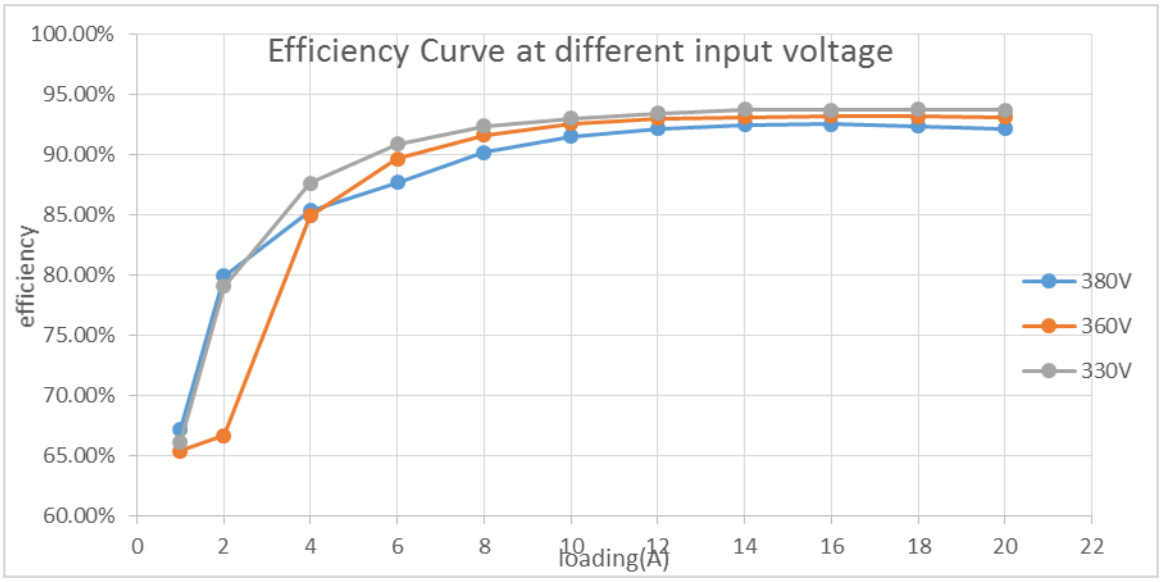


Figure 15. Efficiency curve

Figure 16 shows the switching frequency curve.

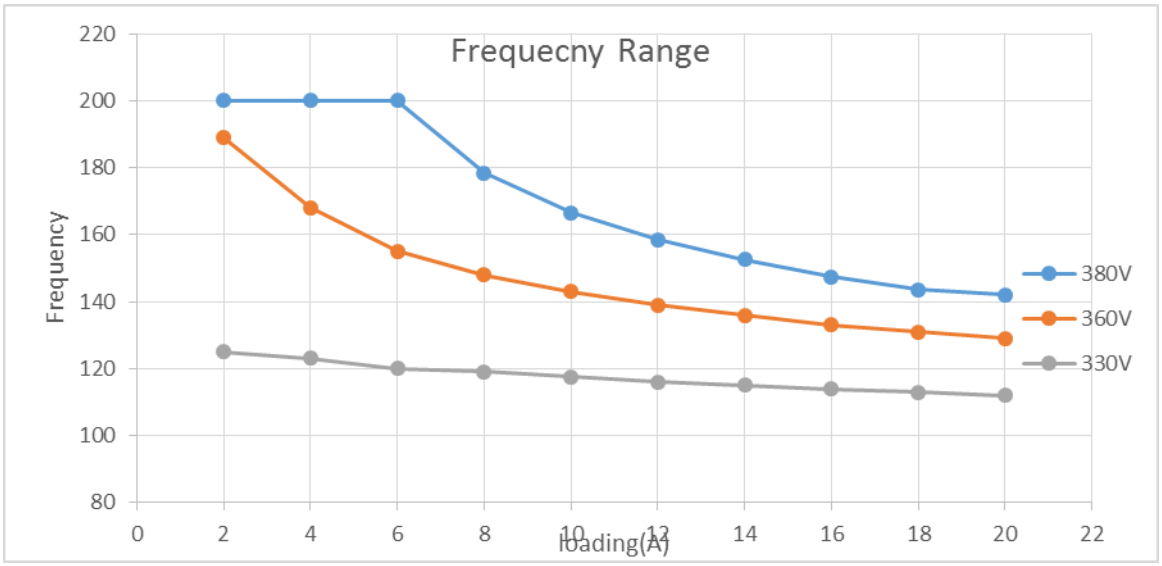


Figure 16. Switching frequency with loading

5.2 Dynamic performance

Figure 17 shows the dynamic performance at 380 V input, load transition from 0% to 65%, from 50% to 100%, and from 0% to 100%.

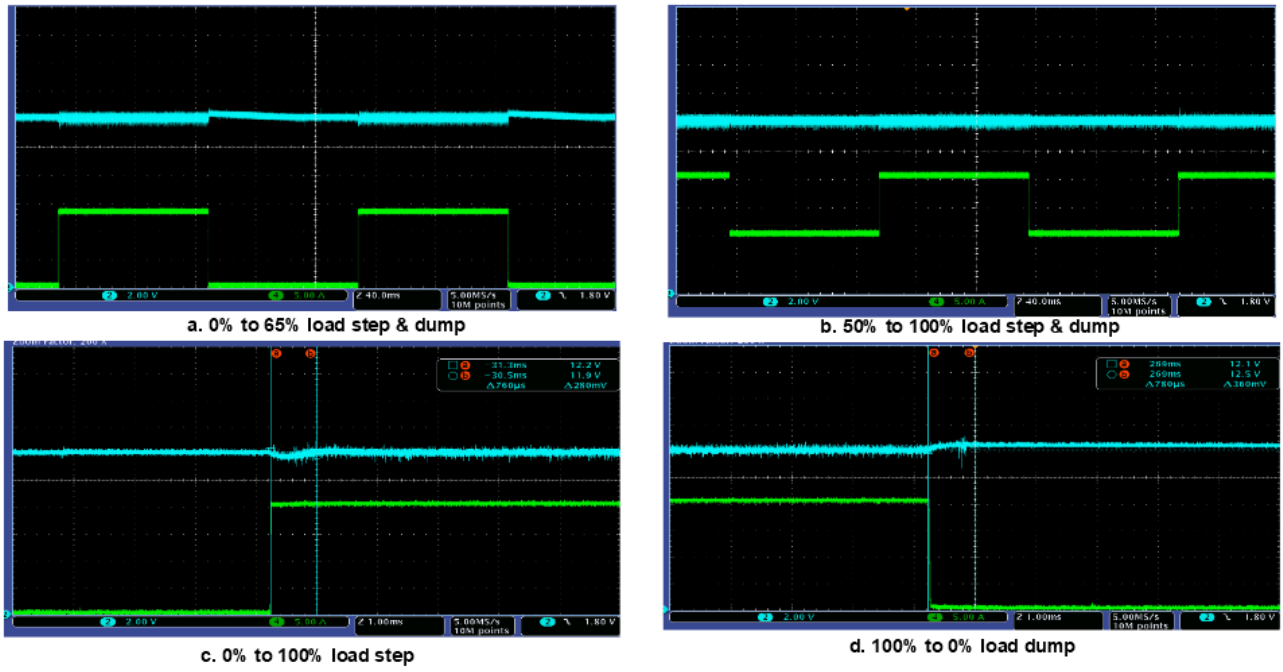


Figure 17. Load transition performance at 380 V input (Ch2: Output Voltage; Ch4: Output Current)

Figure 18 shows the dynamic performance at 330 V input, load transition from 0% to 65%, from 50% to 100%, and from 0% to 100%.

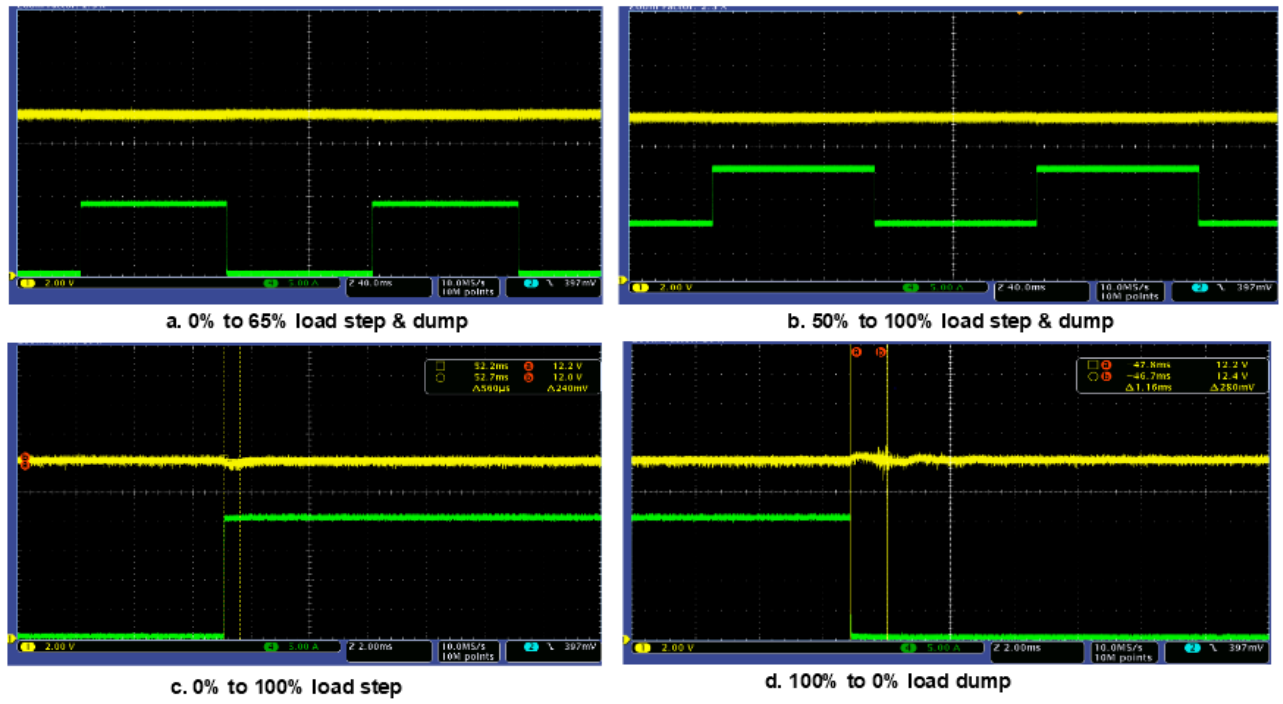
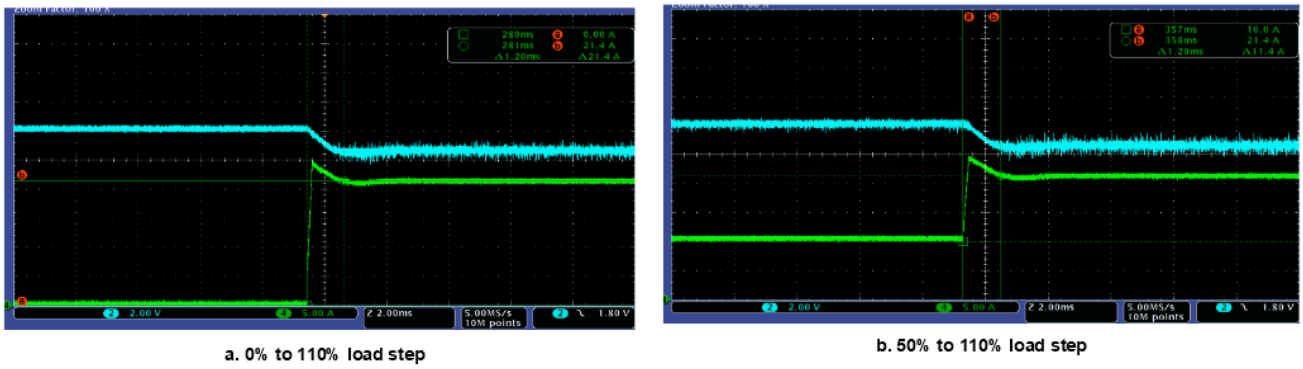


Figure 18. Load transition performance at 330 V input (Ch1: Output Voltage; Ch4: Output Current)

5.3 Current limitation function

Figure 19 shows the performance when enable and trigger current limitation function at 380 V input, current limitation threshold was set at 22A. Fig 19 a shows the loading transition from 0A to 24A and Fig 19 b shows the loading transition from 10A to 24A. Both output current can be limited at about 22A and the output voltage decrease to about 10.5 V.



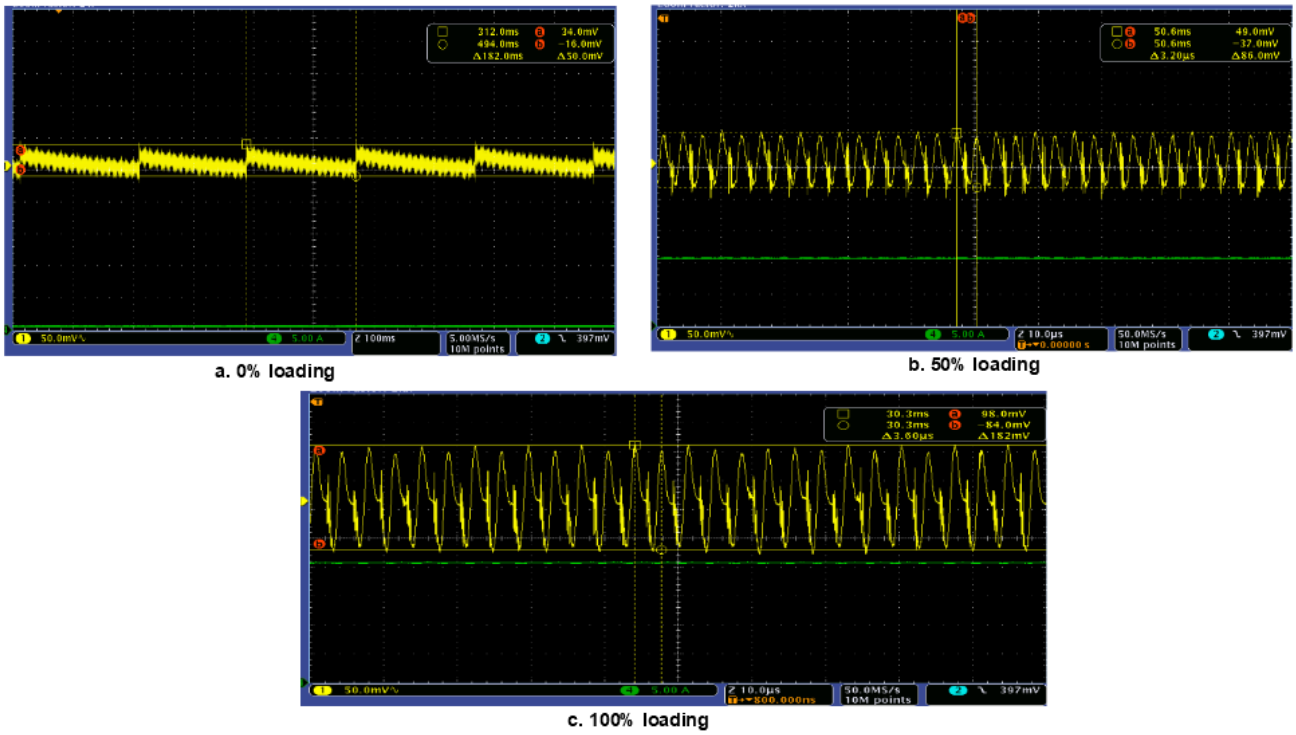
a. 0% to 110% load step

b. 50% to 110% load step

Figure 19. Output current limitation test waveform (Ch2: Output Voltage; Ch4: Output Current)

5.4 Output voltage ripple

Figure 20 shows the performance of 12 V output voltage ripple at 380 V input, system working in burst mode when output current is zero, and in PFM when output current is 10A and 20A.



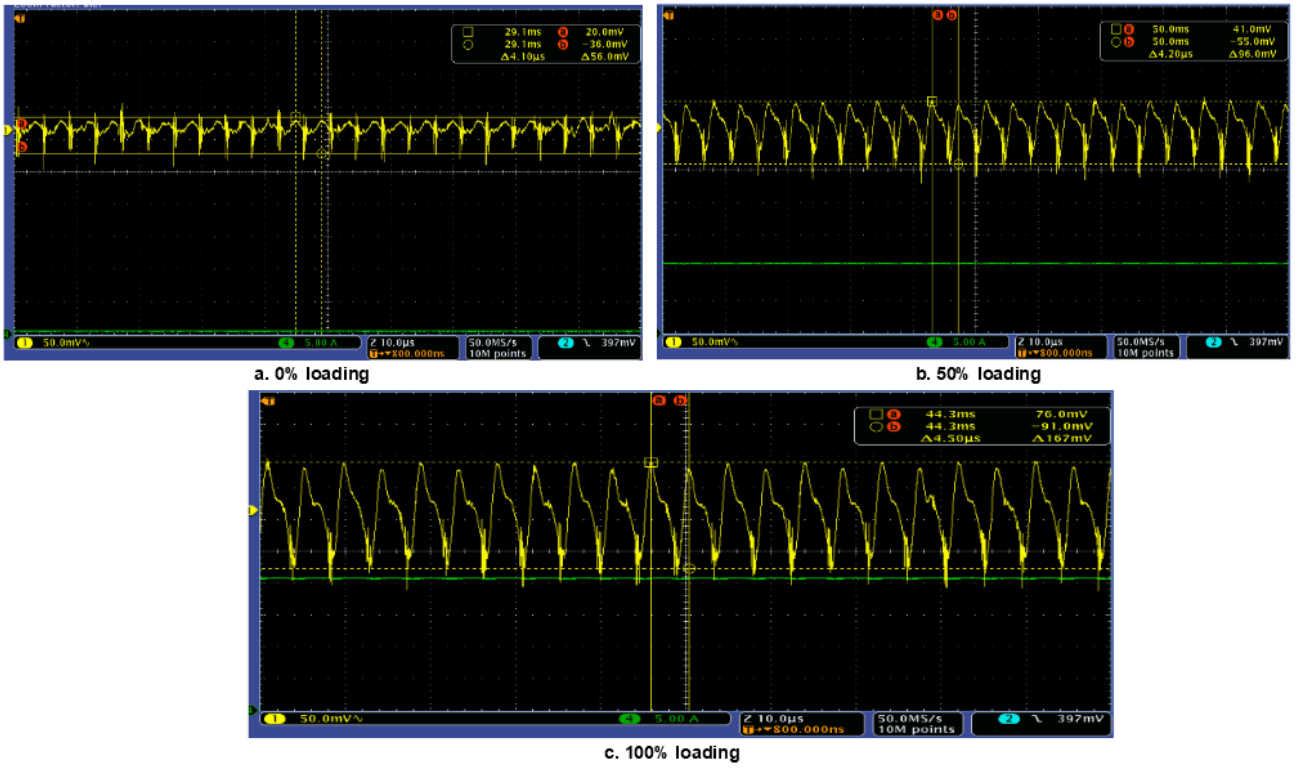
a. 0% loading

b. 50% loading

c. 100% loading

Figure 20. Output voltage ripple test waveform at 380 V input (Ch1: Voltage ripple; Ch4: Output Current)

Figure 21 shows the performance of 12 V output voltage ripple at 330 V input, system working in PFM when output current is 0A, 10A and 20A.



a. 0% loading

b. 50% loading

c. 100% loading

Figure 21. Output voltage ripple test waveform at 330 V input (Ch1: Output Voltage; Ch4: Output Current (AC mode))

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

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: AN12617