*Application Note*

# *UCD3138A: Implement Inverse BUCK-BOOST control and solve common problem using PCM*

*Peng Zhang* *FAE AA2*

**ABSTRACT**

*This document was translated from a simplified Chinese source.* (ZHCAG54)

PCM mode of UCD3138A is often used to control power topologies such as HSFB, PSFB, and Inverse BUCK-BOOST (IBB). For the IBB topology, because the shunt inductor current can generate negative voltages, an additional bias voltage is needed in the actual application to raise the voltage input to UCD3138A. This can cause runaway of the loop. This article will explain how to set PCM loop of UCD3138A for IBB control with two phases interleaved. In practical applications, under conditions where a bias voltage exists, the IBB system implements two functions: the shortest pulse output for pre-charging and normal loop startup.

## Table of Contents



## List of Figures

# 1 IBB basic control loop

UCD3138A, a widely used digital controller in TI power management systems, features a high-speed and high-precision hardware loop. Its high flexibility makes the UCD3138 suitable for multiple power supply topologies. UCD3138 is a fully programmable solution that can be customized to meet the needs of customers in different applications. PCM mode of UCD3138 is often applied to the control of the IBB topology. The system in an IBB application is designed as follows:

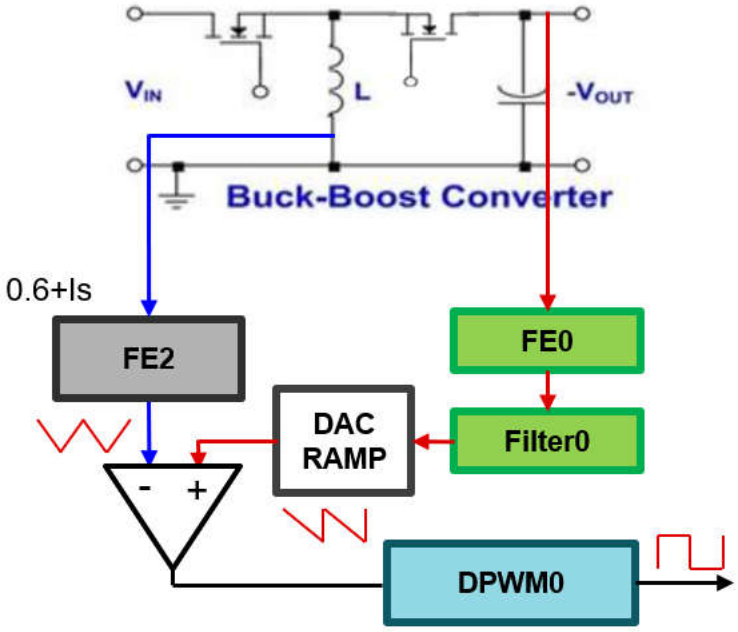



**Figure 1-1. UCD3138A is applied to IBB control**

In actual use, due to customer needs such as pre-charge and input bias voltage etc. PCM control failure may occur. Therefore a proper understanding of the configuration and calculation process of PCM loop and critical registers of UCD3138A is crucial for solving such problems.

## 2 Basic configuration of UCD3138A PCM loop

The most core feature of UCD3138A is its programmable high-speed hardware loop. In practical testing of UCD3138A, this loop achieves a response time from sampling to completed regulation within 400ns. The latest generation UCD3138R achieves loop response speeds within 156ns. Control PCM with UCD3138A. From a hardware perspective, it should be understood as Figure 4: FE0 sampled output voltage is compared to DAC reference value in FE0 to obtain an error value. Enter Filter 0 to obtain the voltage loop calculation results by PID operation (gain programmable), which is the output Filter_Duty of Fliter0. Since only FE2 module in UCD3138A supports PCM mode, the output of Fliter0 should be looped back to FE2 as the input for the Ramp module. And the sampling of FE2 acts as a current loop. When the current loop output sampled by FE2 exceeds the voltage loop output result (DAC current value) after Ramp compensation, DPWM0A sends high level. Instead, DPWM0B sends high level. Implement PCM control.

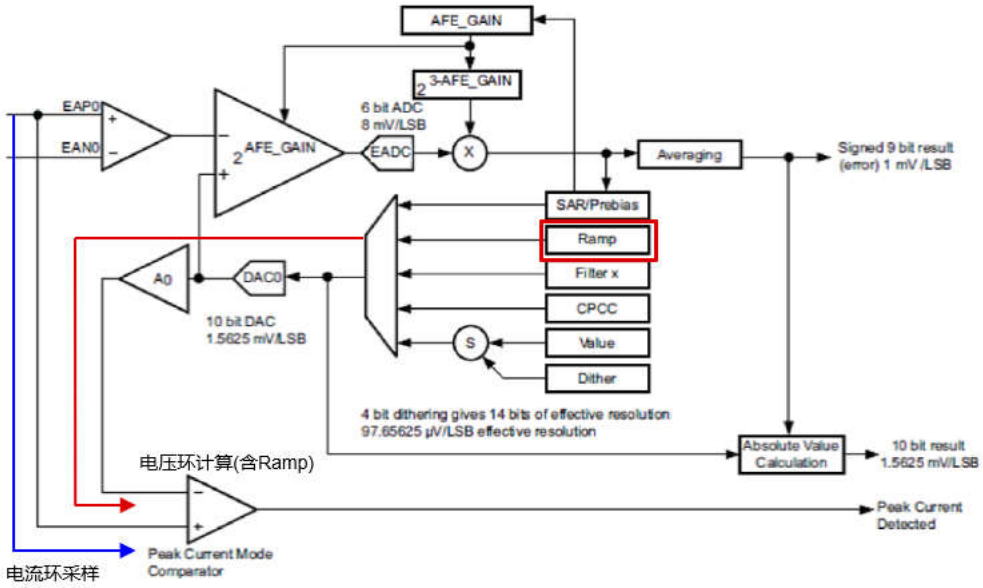



**Figure 2-1. PCM hardware loop**

The key register is configured as follows:

```
///

void init_loop_mux(void)
{
    LoopMuxRegs.DPWMMUX.bit.DPWM0_FILTER_SEL = 0;       // use filter 0 for DPWM 0
    LoopMuxRegs.SAMPTRIGCTRL.bit.FE0_TRIG_DPWM0_EN = 1; // use DPWM0 for filter0 sample trigger
//  LoopMuxRegs.SAMPTRIGCTRL.bit.FE1_TRIG_DPWM0_EN = 1; // use DPWM0 for filter1 sample trigger

    LoopMuxRegs.FECTRL2MUX.bit.DPWM0_FRAME_SYNC_EN = 1; //also need a trigger to make it work better.
    LoopMuxRegs.FECTRL0MUX.bit.DPWM0_FRAME_SYNC_EN = 1; //also need a trigger to make it work better.


    LoopMuxRegs.APCMCTRL.bit.PCM_FE_SEL    = 2;   // use FE2 for PCM
    LoopMuxRegs.APCMCTRL.bit.PCM_LATCH_EN = 1;   ///


    LoopMuxRegs.APCMCTRL.bit.PCM_EN        = 1;   // enable PCM

    LoopMuxRegs.PCMCTRL.bit.PCM_FILTER_SEL = 0; //select filter0
}
```

```
//----------------------------------------
Dpwm0Regs.DPWMCTRL0.bit.PWM_A_INTRA_MUX  = 0;   // Send PWM-C out the A output

Dpwm0Regs.DPWMCTRL0.bit.CBC_PWM_AB_EN = 1; //使能CBC
Dpwm0Regs.DPWMCTRL0.bit.CBC_ADV_CNT_EN = 1;

Dpwm0Regs.DPWMCTRL0.bit.BLANK_A_EN = 1;        //使能DPWM0A Blank time
Dpwm0Regs.DPWMBLKABEG.bit.BLANK_A_BEGIN = 0;   //Blank 起始位置
Dpwm0Regs.DPWMBLKAEND.bit.BLANK_A_END = 25; //25 x 4ns =100ns

Dpwm0Regs.DPWMCTRL0.bit.BLANK_B_EN = 1;            //使能DPWM0B Blank Time
Dpwm0Regs.DPWMBLKBBEG.bit.BLANK_B_BEGIN = EVENT3;   // Blank B 起始位置
Dpwm0Regs.DPWMBLKBEND.bit.BLANK_B_END = EVENT3 + 25; //25 x 4ns =100ns

Dpwm0Regs.DPWMCTRL0.bit.PWM_EN = 1;     // enable DPWM0 locally

FaultMuxRegs.DPWM0CLIM.bit.ANALOG_PCM_EN   = 1; //DPWM0 Enable PCM

//----------------------------------------
Dpwm0Regs.DPWMCTRL0.bit.PWM_A_INTRA_MUX  = 0;   // Send PWM-C out the A output

Dpwm0Regs.DPWMCTRL0.bit.CBC_PWM_AB_EN = 1; //使能CBC
Dpwm0Regs.DPWMCTRL0.bit.CBC_ADV_CNT_EN = 1;

Dpwm0Regs.DPWMCTRL0.bit.BLANK_A_EN = 1;        //使能DPWM0A Blank time
Dpwm0Regs.DPWMBLKABEG.bit.BLANK_A_BEGIN = 0;   //Blank 起始位置
Dpwm0Regs.DPWMBLKAEND.bit.BLANK_A_END = 25; //25 x 4ns =100ns

Dpwm0Regs.DPWMCTRL0.bit.BLANK_B_EN = 1;            //使能DPWM0B Blank Time
Dpwm0Regs.DPWMBLKBBEG.bit.BLANK_B_BEGIN = EVENT3;   // Blank B 起始位置
Dpwm0Regs.DPWMBLKBEND.bit.BLANK_B_END = EVENT3 + 25; //25 x 4ns =100ns

Dpwm0Regs.DPWMCTRL0.bit.PWM_EN = 1;     // enable DPWM0 locally

FaultMuxRegs.DPWM0CLIM.bit.ANALOG_PCM_EN   = 1; //DPWM0 Enable PCM
```

```
void init_front_end0(void)
{
    FeCtrl0Regs.EADCDAC.bit.DAC_VALUE = 160;
    FeCtrl0Regs.EADCCTRL.bit.AFE_GAIN = 2;

//    FeCtrl2Regs.RAMPCYCLE.bit.SWITCH_CYC_PER_STEP = 0;   //32ns/cycle
//
    FeCtrl2Regs.DACSTEP.bit.DAC_STEP              = 1000; //
    FeCtrl2Regs.RAMPCTRL.bit.PCM_START_SEL        = 1;    // 0 - Use DAC value from DAC; 1 - use filter out
    FeCtrl2Regs.RAMPDACEND.bit.RAMP_DAC_VALUE = 500; //may not be perfect, keep above 0 to avoid potential problems

    FeCtrl2Regs.EADCDAC.bit.DAC_VALUE  =320 * 16;   //Set DAC at 0.97V

    FeCtrl2Regs.EADCCTRL.bit.AFE_GAIN  = 2;

    FeCtrl2Regs.EADCCTRL.bit.EADC_MODE = 5;       // Peak current mode
    FeCtrl2Regs.RAMPCTRL.bit.RAMP_EN   = 1;       //use internal slope--1182012
    FeCtrl2Regs.EADCCTRL.bit.D2S_COMP_EN = 1; //vital to enable comparator - this enables APCM comparator.
}

void connect_filter_0_to_front_end2_DAC(void)
{
    //LoopMuxRegs.SAMPTRIGCTRL.bit.FE0_TRIG_DPWM0_EN = 1; //use DPWM0 for filter0 sample trigger
   // LoopMuxRegs.FILTERKCOMPA.bit.KCOMP0 = 0x3fff; //0x3FFF is full scale

    LoopMuxRegs.EXTDACCTRL.bit.DAC2_SEL = 4; // 3 for CPCC. 4 for filter 0;
    LoopMuxRegs.EXTDACCTRL.bit.EXT_DAC2_EN = 0; //enable DAC 2 to take input from selected spot.1 is driven by filter
}
```

The combination of a waveform graph provides a better understanding of the role of registers in real control. DAC_Step register in FE2 controls the compensation slope for Ramp and the DAC_Value controls the end point of Ramp compensation. It should be noted here that during Ramp compensation, the current value of DAC is always close to the end point set by DAC_Value. If the output of Filter is greater than DAC_Value, the current result of DAC is increasing continuously, otherwise, it decreases.

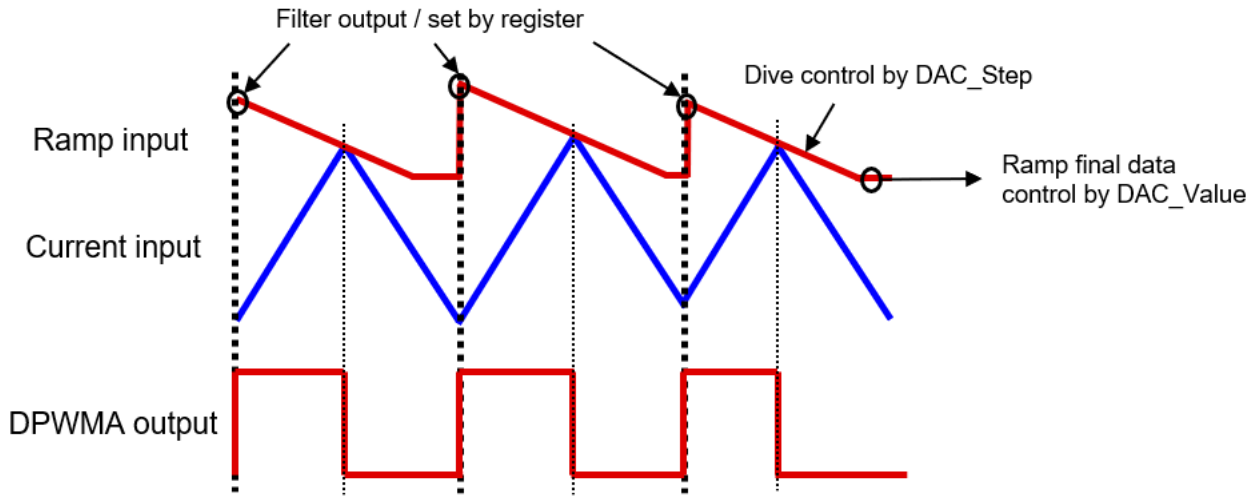



**Figure 2-2. Operating principle of UCD3138A PCM**

The significance of Blank time lies in the discontinuity of the inductor current when MOSFET turns on. At this point, significant ringing may occur at the switch node, as a result, the output voltage is sampled incorrectly and then erroneous shutdown of the DPWM. To address this issue, UCD3138 can mask DPWM rising edge for a period by enabling Blank register, preventing DPWM's rising edge from erroneously triggering PCM. The register to enable DPWM Blank function and Blank time is described as follows:

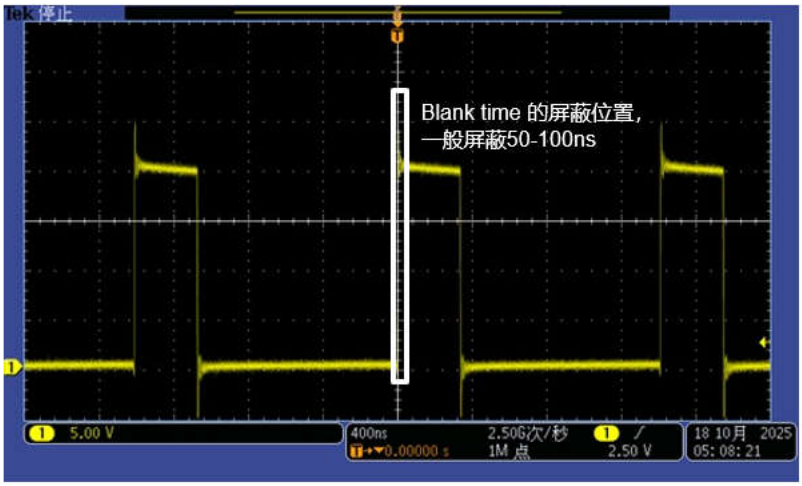



**Figure 2-3. Ringing and false trigger protection of switch node**

```
//----------------------------------------
Dpwm0Regs.DPWMCTRL0.bit.PWM_A_INTRA_MUX  = 0;   // Send PWM-C out the A output

Dpwm0Regs.DPWMCTRL0.bit.CBC_PWM_AB_EN = 1; //使能CBC
Dpwm0Regs.DPWMCTRL0.bit.CBC_ADV_CNT_EN = 1;

Dpwm0Regs.DPWMCTRL0.bit.BLANK_A_EN = 1;        //使能DPWM0A Blank time
Dpwm0Regs.DPWMBLKABEG.bit.BLANK_A_BEGIN = 0;   //Blank 起始位置
Dpwm0Regs.DPWMBLKAEND.bit.BLANK_A_END = 25; //25 x 4ns =100ns

Dpwm0Regs.DPWMCTRL0.bit.BLANK_B_EN = 1;            //使能DPWM0B Blank Time
Dpwm0Regs.DPWMBLKBBEG.bit.BLANK_B_BEGIN = EVENT3;   // Blank B 起始位置
Dpwm0Regs.DPWMBLKBEND.bit.BLANK_B_END = EVENT3 + 25; //25 x 4ns =100ns

Dpwm0Regs.DPWMCTRL0.bit.PWM_EN = 1;     // enable DPWM0 locally

FaultMuxRegs.DPWM0CLIM.bit.ANALOG_PCM_EN   = 1; //DPWM0 Enable PCM
```

**Figure 2-4. Enable UCD3138A Blank time and set time**

# 3 PCM failure analysis after introduction of bias voltage

In practical operation, UCD3138 performs two adjustments during IBB control, which are additional functions required based on theoretical control.

1. The loop is pre-charged with the UCD3138 Blank function (a total of around 100ms) before starting the loop normally.
2. An additional bias voltage is required for FE sampling to ensure the input voltage to FE remains positive.

For synchronous Inverse BUCK-BOOST topology power control, both voltage loop and current loop require sampling. The voltage loop generally samples the output voltage and its polarity is not reversed, so it is secure for UCD3138 Front End (FE) interface. However, in light load operation in FPWM mode, IBB, like other power supplies, will reverse the inductor current (e.g. Figure 3-1 [3]).

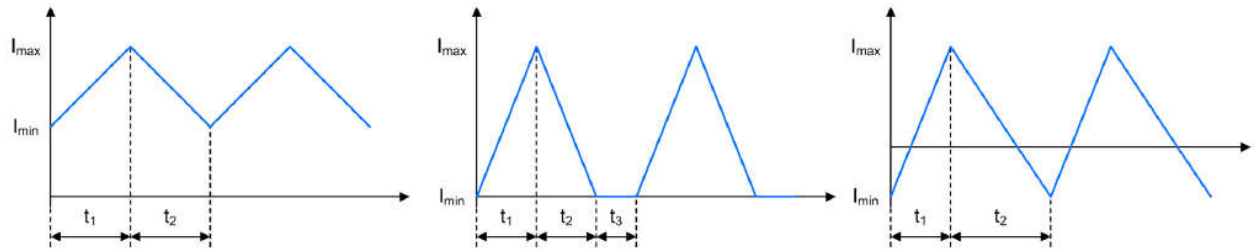



**Figure 3-1. Inductor current reverses at light load**

Therefore, when the current loop is sampled, the voltage output to FE interface may be negative, which is dangerous for UCD3138 and may damage to UCD3138.

## 7.1 Absolute Maximum Ratings(1)

over operating free-air temperature range (unless otherwise noted)

| | | VALUE | | UNIT |
|---|---|---|---|---|
| | | MIN | MAX | |
| Voltage | V33D to DGND | –0.3 | 3.8 | V |
| | V33DIO to DGND | –0.3 | 3.8 | V |
| | V33A to AGND | –0.3 | 3.8 | V |
| | BP18 to DGND | -0.3 | 2.5 | V |
| | Ground difference, \|DGND – AGND\| | | 0.3 | V |
| | Voltage applied to any pin, excluding AGND(2) | –0.3 | 3.8 | V |

**Figure 3-2. UCD3138A voltage input range [2]**

Therefore, we usually need to provide a positive bias voltage to the input of FE to solve this problem, as shown in Figure 3-3, where a 0.6V positive bias voltage is applied to UCD3138 input to ensure the FE input remains positive.

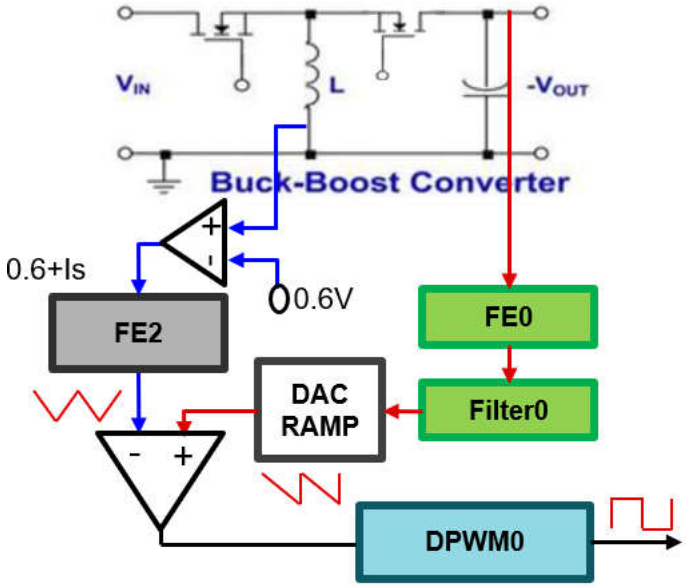



**Figure 3-3. Provide 0.6V forward bias voltage to FE2**

Figure 3-4provides a more intuitive illustration of the differing calculation expectations for voltage loops and current loops during the pre-charge phase versus the normal operating phase in practical operation.

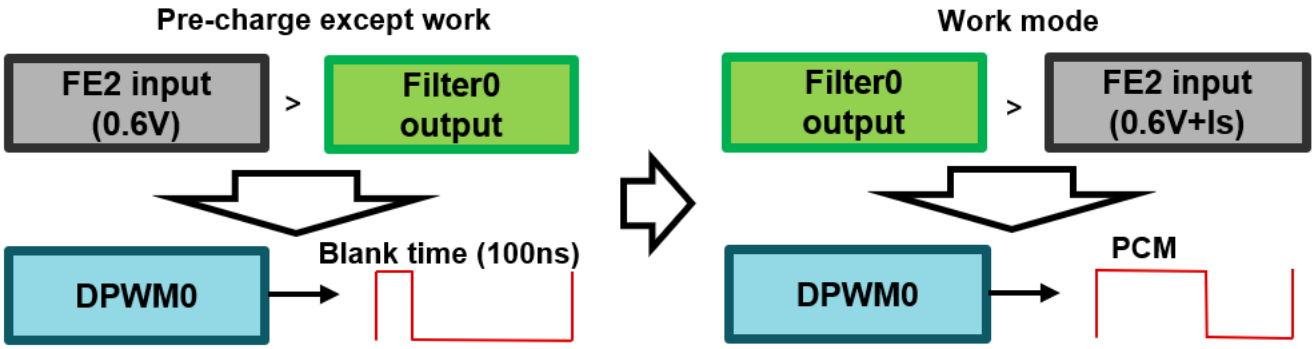



**Figure 3-4. Loop expectations for pre-charge phase versus normal operating mode**

But raising this bias voltage will cause a problem. The current loop sampling value of the FE2 input will be 0.6V+*Isample*, so that at start-up, the sample of FE2 will have 0.6V even if there is no current. And the output voltage is 0 at start-up, so the loop will perpetually calculate an output voltage that is too low. Consequently, the pre-charge function utilizing blank time in the first stage loses control. It can be seen from the waveform that DPWM0A remains continuously on until reaching maximum duty cycle or triggering overcurrent protection.

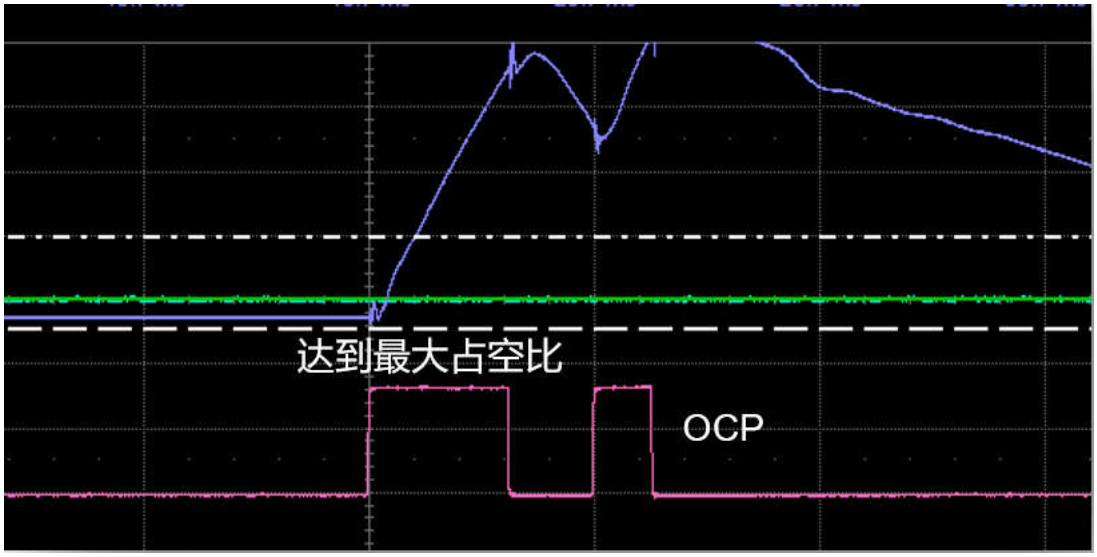



**Figure 3-5. PCM failure**

## 4 Loop correction and experimental verification

Analysis of the issue reveals that the primary cause of instability during the pre-charge phase is the filter calculation results failing to adapt to the presence of bias voltage. The Filter calculation results is controlled by the error input of FE0 in conjunction with the loop gain of Filter. For general loop gain deficiency or reference setting problems, the gain can be adjusted by adjusting DAC_Value adjustment threshold of FE0 or KI coefficient of Filter. However, DAC_Value cannot be set to a negative value for the IBB startup pre-charge failure and the error output is constant positive when the voltage output is 0. The loop gain factor will continue to be used in subsequent normal operating modes, which can be cumbersome to modify. For this problem, we recommend using the Clamp register in Filter to clamp the output of Filter.

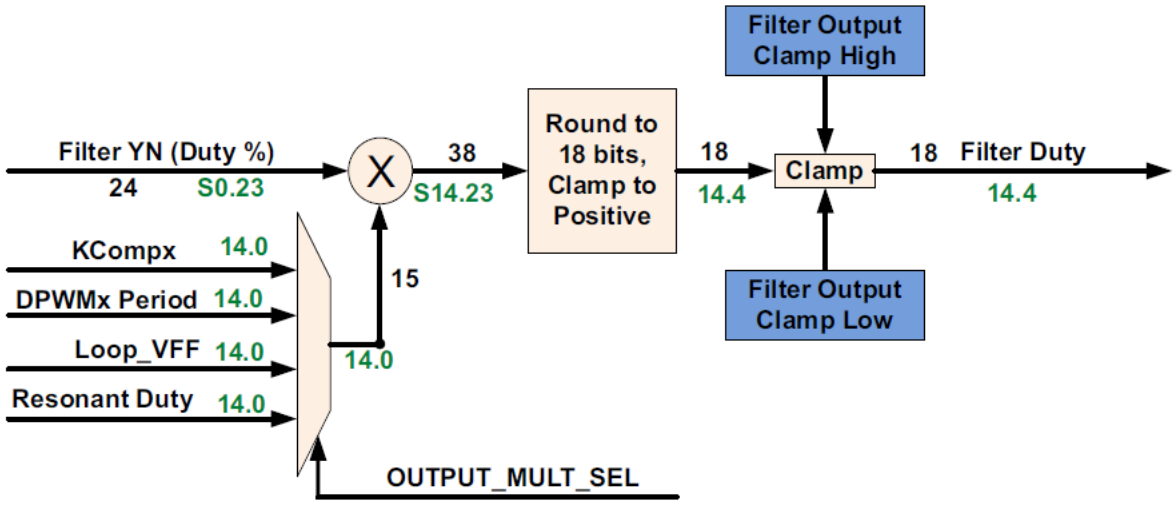



**Figure 4-1. Filter output Clamp register** [1]

For two links of IBB. First we should clamp the Filter Output Clamp Low to a lower value (e.g. 0X10). Make sure the output of the voltage loop is always below the 0.6V sampled value of the current loop in the first link. After enabling 100ns Blank time, the short pulse output for pre-charging will be implemented normally.


**Figure 4-2. Pre-charging pulse output**

After a 100ms delay, the clamp will return to the normal range. Set DAC_Value (FB Ref) in FE0 to normal operating mode, and the loop will start normally.

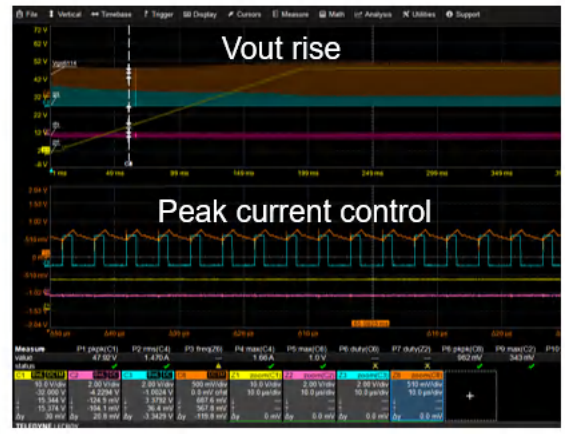


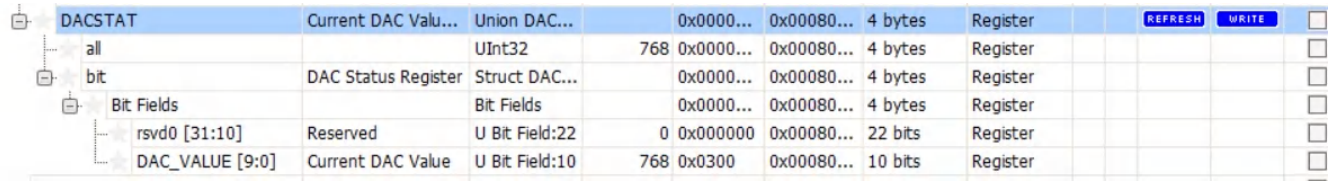

**Figure 4-3. IBB operates and outputs pulses normally**

Thus, we have implemented a simple method that enables pre-charging and normal loop startup in applications involving sampled inductor current under biased conditions.

# 5 Other debugging experience and frequently asked questions

It should be noted that the Filter involves multi-level clamping. In practical applications, we should focus on application requirements. If it is desired to detect overflow or excessively low values during calculation, clamping may be applied to the first two stages. However, in UCD31XX technical documentation, the described Filter-like calculation result gives loop to a module. Unless otherwise specified, all results will uniformly refer to those of Filter_Duty. If bit truncation is required in subsequent operations, the lower bits shall be truncated uniformly, and the upper bits shall be retained.

Take PCM control as an example. If we set Filter Output Clamp Low & High to: 0x300FF, but DAC of the Front End module only supports 10bit, then the actual DAC will be assigned to be 0x300. This is a question common to many developers. After turning on PCM mode, we can set DAC_Step to 0 and Filter Output Clamp Low & High to the same value, and then read the current value of DAC through the memory debugger. Compare this with the calculation result to verify whether your understanding of the calculation process is correct.



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| DACSTAT | Current DAC Valu... | Union DAC... | | 0x0000... | 0x00080... | 4 bytes | Register | REFRESH | WRITE |
| all | | UInt32 | 768 | 0x0000... | 0x00080... | 4 bytes | Register | | |
| bit | DAC Status Register | Struct DAC... | | 0x0000... | 0x00080... | 4 bytes | Register | | |
| Bit Fields | | Bit Fields | | 0x0000... | 0x00080... | 4 bytes | Register | | |
| rsvd0 [31:10] | Reserved | U Bit Field:22 | 0 | 0x000000 | 0x00080... | 22 bits | Register | | |
| DAC_VALUE [9:0] | Current DAC Value | U Bit Field:10 | 768 | 0x0300 | 0x00080... | 10 bits | Register | | |

**Figure 5-1. Set the Clamp register and read the value of DAC_Value with Memory debugger**

# 6 Summary

The UCD3138 family of digital power controllers deliver high-speed, high-precision control and flexible topology options. There are also a wealth of application notes and documentation to help developers at all levels develop different application topologies. The next generation UCD3138R will feature a more powerful main core (100MHz M33), higher DPWM accuracy, and faster loop response. In summary, the UCD3138 family of products will provide an excellent solution for achieving high-speed and high-precision power control.

# 7 References:

1. UCD3138A datasheet
2. Technical Manual of UCD31xx Digital Power Supply Controller
3. Power Stage Topology Handbook

# IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.



Last updated 10/2025