NXP Semiconductors
Application Note

Document Number: AN5186
Rev. 2.0, 2/2016

PT2000 Diagnostics

1 Introduction

This application note explains how to use PT2000 diagnostics in a typical six cylinder internal combustion engine (ICE) application. The field of power train is just one example where diagnostics are required at very high speed. The PT2000 diagnostics manage this through six independent microcores.

This application note seeks to address different fault cases and describes how to program the microcode to detect them during idle and actuation mode.

NXP analog ICs are manufactured using the SMARTMOS process, a combinational BiCMOS manufacturing flow that integrates precision analog, power functions, and dense CMOS logic together on a single cost-effective die.

2 Overview

The PT2000 is a 15-channel gate driver IC for automotive engine control applications. The IC consists of seven external MOSFET high- side pre-drivers and eight external MOSFET low side pre-drivers. The PT2000 provides a flexible solution for the MOSFET's gate drive with a versatile control and optimized latency time. Gate drive, diagnosis, and protection are managed through six independent microcores, three Code RAM, and three Data RAM banks.

Contents

3 Application Schematic

The PT2000 typical application controls three injection banks, one DC-DC and a pump bank.

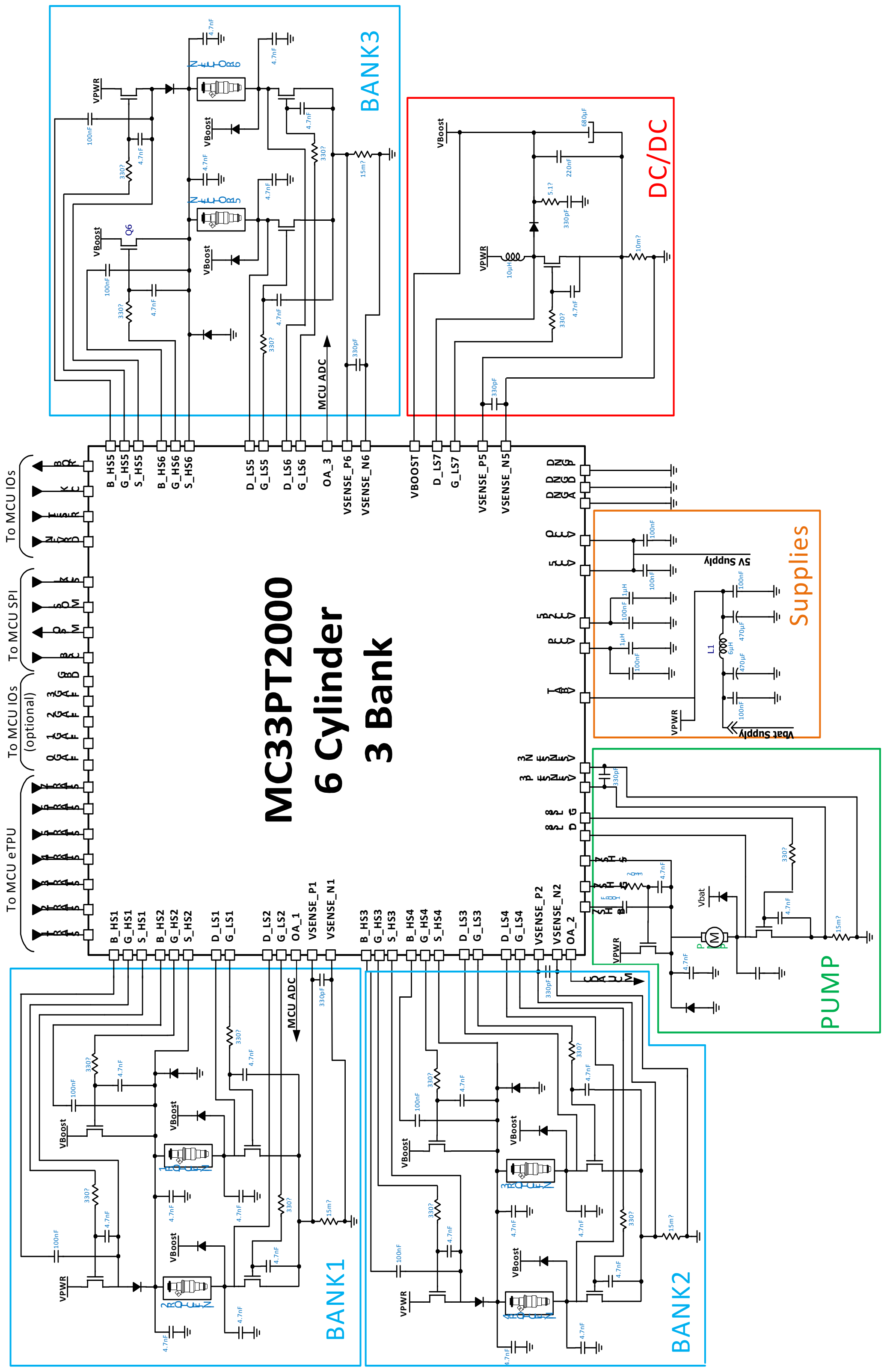

Figure 1. Typical Six Injector Three Bank Application Schematic

4 Application Instructions

This topology can be used on the evaluation board KITPT2000FRDM6C. Register settings and microcode downloads can be achieved by using the KL25Z embedded on the KITPT2000FRDM6C.

Each bank is individually managed by one microcore of the digital channel 1 as described next:

- The bank # 1 is managed by the digital microcore Uc0Ch1 with diagnostics
- The bank # 2 is managed by the digital microcore Uc1Ch1 with diagnostics, but no description is provided in this document since it is a simple copy paste from the previous bank. Microcode is available.
- The bank # 3 is managed by the digital microcore Uc0Ch2 with diagnostics, but no description is provided on this document since it is a simple copy paste from the previous bank. It is programmed on the microcode example available on the web.

The two microcores of the second channel (Channel 3) drive the DC-DC and the fuel pump as described next:

- The VFM (Variable Frequency Modulation) is managed by the digital microcore Uc1Ch3
- The fuel pump is managed by the digital microcore Uc0Ch3

This application note only focuses on BANK1 diagnostics managed by the digital microcore Uc0Ch1. Refer to AN5187 **for register settings and microcode related to injection or DC-DC, unless specified in this document.**

The following is the start-up sequence:

- Apply a battery voltage between 9.0 V and 16 V
- Download the registers Channel Configuration, Main Configuration, IO Configuration, and Diagnostic Configuration
- Download the dedicated microcode in the Logic Channel 1, Logic Channel 2 and Logic Channel 3 Data RAM
- Set ‘1’ in the pre-flash enable bit and en dual seq bit in the Flash_enable register of channel 1 (100h), channel 2 (120h) and channel 3 (140h)

The register configurations and the microcodes are detailed in the following chapters.

Table 1. Example of Injection Current Profile Key Parameters

Parameter Name	Description	Value
I_{BOOST}	Current threshold in Boost Phase	14 A
I_{PEAK}	Current threshold in Peak Phase (Depends on injectors type)	6 A
I_{HOLD}	Current threshold in Hold Phase	3 A
t_{PEAK_OFF}	Fixed time for high-side switch off in Peak Phase	10 μs
t_{PEAK_TOT}	Fixed time for end of Peak Phase	500 μs
t_{BYPASS}	Fixed time for Bypass Phase	20 μs
t_{HOLD_OFF}	Fixed time for high-side switch off in Hold Phase	10 μs
t_{HOLD_TOT}	Fixed time for end of Hold Phase (timeout)	10 ms
$t_{INJMAXBOOST}$	Maximum time allowed to reach I_{BOOST} (Depends on injectors type)	500 μs

Diagnostics interrupts description:

Diagnostics interrupts are handled in two different subroutines: automatic interrupt and software interrupt.

Status_reg_uc0 register (105h) is used to inform MCU on error detected, and then control register is used to unlock the Bank.

In all cases, the IRQB pin is set low to inform the MCU about the error detected in PT2000.

Interrupts can be filtered by their req id (stored in Uc0_irq_status Registers 10Fh):

Pre-Diagnostics checks (req id = 0): interrupt occurs if the high-side V_{BOOST}/ V_{BAT}, V_{DS} or V_{SRC} are low or V_{DS} low-side is low.

If an error occurs, the Status_reg_uc0 register (105h) bit 7 sets high, Bank1 sets OFF, and MCU needs to write a 1 to the control register (101h) bit 7 to unlock the Bank1.

Boost Error (Req id = 1): If I_{BOOST} is not reached before $t_{INJMAXBOOST}$ = 500 μs, this number has to be set according to the injector characteristics.

If an error occurs, Status_reg_uc0 register (105h) bit 5 sets high, Bank1 sets OFF, and MCU needs to write a 1 to the Ctrl_reg_uc0 register (101h) bit 5 to unlock the Bank1.

Hold Error (Req id = 2): If Start signal is still high after t_{HOLD_OFF}.

If an error occurs, the Status_reg_uc0 register (105h) bit 4 sets high, Bank1 sets OFF, and MCU needs to write a 1 to the Ctrl_reg_uc0 register (101h) bit 4 to unlock the Bank1.

An automatic Interrupt occurs during actuation, if comparators feedback is different than the error table (see Diagnostics Configuration Registers).

If an error occurs, the Status_reg_uc0 register (105h) bit 6 sets high, Bank1 sets OFF, and MCU needs to write a 1 to the Ctrl_reg_uc0 register (101h) bit 6 to unlock the Bank1.

Table 2. Status_reg_uc0 Registers (105h) Configuration

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	status_register															
Value	X	X	X	X	X	X	X	X	SW Interrupt Status	Auto Interrupt Status	Boost Error Status	Hold Error Status	X	X	Injector 2 Error Status	Injector 1 Error Status

Reading this register indicates the type of fault.

Table 3. Ctrl_reg_uc0 Registers (101h) Configuration

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	control_register_shared								control_register							
Value	X	X	X	X	X	X	X	X	SW Interrupt Unlock	Auto Interrupt Unlock	Boost Error Unlock	Hold Error Unlock	X	X	X	X

Depending on the status register information, one of the bits must be set to 1 to unlock the BANK.

If an automatic diagnostics error is detected, it is possible to detect which failure occurred by reading the following registers to have an image when the failure occurred of the command, the V_{DS} and V_{SRC} state for each high-side and low-side. It allow detection of a short to ground, an open load, a short circuit to GND, or a short-circuit to V_{BAT}.

Table 4. Err_ucxchy_1 (1EDh to 1FEh)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved	cmd_Hs5	Vsrc_Hs5	Vds_Hs5	cmd_Hs4	Vsrc_Hs4	Vds_Hs4	cmd_Hs3	Vsrc_Hs3	Vds_Hs3	cmd_Hs2	Vsrc_Hs2	Vds_Hs2	cmd_Hs1	Vsrc_Hs1	Vds_Hs1

Table 5. Err_ucxchy_2 (1EDh to 1FEh)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	cksys_missing	Reserved									cmd_Hs7	Vsrc_Hs7	Vds_Hs7	cmd_Hs6	Vsrc_Hs6	Vds_Hs6

Table 6. Err_ucxchy_3 (1EDh to 1FEh)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	cmd_LS8	Vds_Ls8	cmd_LS7	Vds_Ls7	cmd_Ls6	Vds_Ls6	cmd_Ls5	Vds_Ls5	cmd_Ls4	Vds_Ls4	cmd_Ls3	Vds_Ls3	cmd_Ls2	Vds_Ls2	cmd_Ls1	Vds_Ls1

For example:

- IRQB pin pulled low
- MCU reads status register (105h) and detects Injector 1 fails during automatic diagnostics
- MCU then reads Err_uc0_ch1_1 register (EDh) and detects that cmd_HS1 = 1 and Vds_Hs1 = 1, it means that a short to GND occurred on the HS1 (refer to Table 13)
- Depending on strategy, MCU decides to write control register (101h) to unlatch the fault and make another trial.

5 Diagnostic Descriptions

The PT2000 gives the possibility to check faults using two different methods:

- Automatic diagnostics (Actuation phase):
  - Boost Phase (HSBoost ON): automatic diagnostics are used during actuation phase; it performs a coherency check between an output and the related V_{DS} feedback (for all the outputs) and V_{SRC} feedback (for the high-side outputs only).
  - Peak and Hold phase (HSBat ON): automatic diagnostics are used during actuation phase; it performs a coherency check between an output and the related V_{DS} feedback (for all the outputs) and V_{SRC} feedback (for the high-side outputs only).
- Idle Diagnostics (Pre-actuation): Internal voltage biasing V_{BIAS} should be applied to the load to enable diagnostics in this phase.

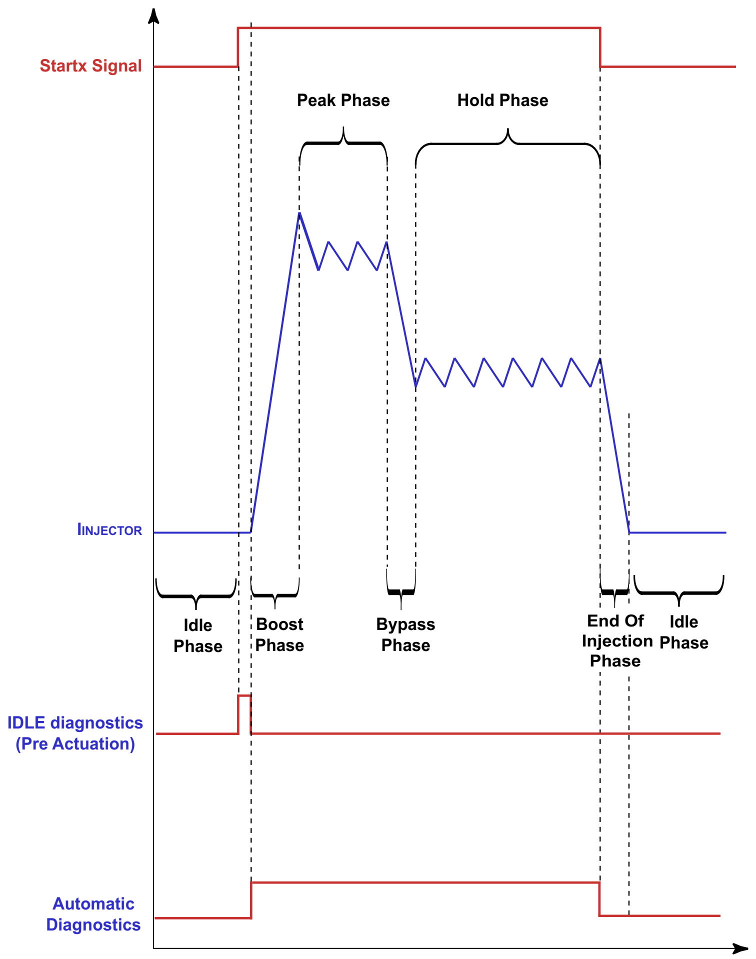

Figure 2. Typical Peak and Hold Current Profile with Diagnostics

Several fault cases could occur in the application, this chapter describes most of them, and explains how the PT2000 is able to detect them.

5.1 Idle Diagnostics (Pre-actuation)

As described in Figure 2, idle diagnostics start after a rising edge on the start 1 or start 2 (Bank 1). A voltage biasing V_{BIAS} should be applied to the load, to enable electrical diagnosis while the external load is not actuating the power stage.

This V_{BIAS} voltage is generated by:

- the activation of the SRC_{PUX} pull-up voltage source connected to each of the S_HSx pins. Each pull-up voltage source is supplied from VCC5
- the activation of each SRC_{PDX} pull-down current source connected to each of the D_LSx pins. Each pull-down voltage source is referenced to ground

When the battery voltage V_{BATT} is in the nominal range or greater, the external load is biased at a minimum voltage of typically 3.8 V. In a low battery voltage condition (V_{BATT} < 8.0 V), the load is biased at half the V_{BATT} voltage, to guarantee symmetrical voltage margins to high-side and low-side VDS comparators.

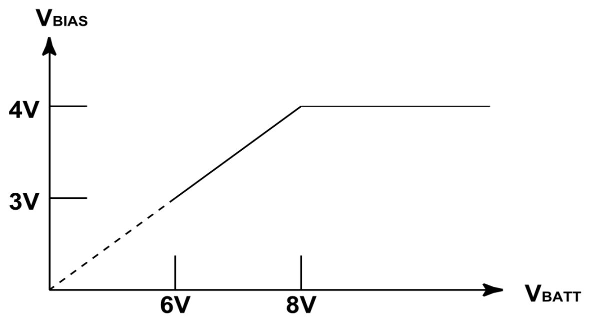

Figure 3. Biasing Voltage vs. V_{BATT}

The bias generators can be kept ON even during actuation, to control the voltage on the source, even if the MOSFET is OFF. This does not impact the application, because of their low strength. If at least one MOSFET is turned ON, it fixes the voltage on the load and does not affect the bias.

These pre-actuation diagnostics are used to ensure the injectors can be turned ON safely. If an error occurs in any of the following cases, the PT2000 keeps Bank 1 OFF until the MCU writes a 1 through the SPI to the Ctrl_reg_uc0 register (101h) bit 7.

5.1.1 Normal Behavior

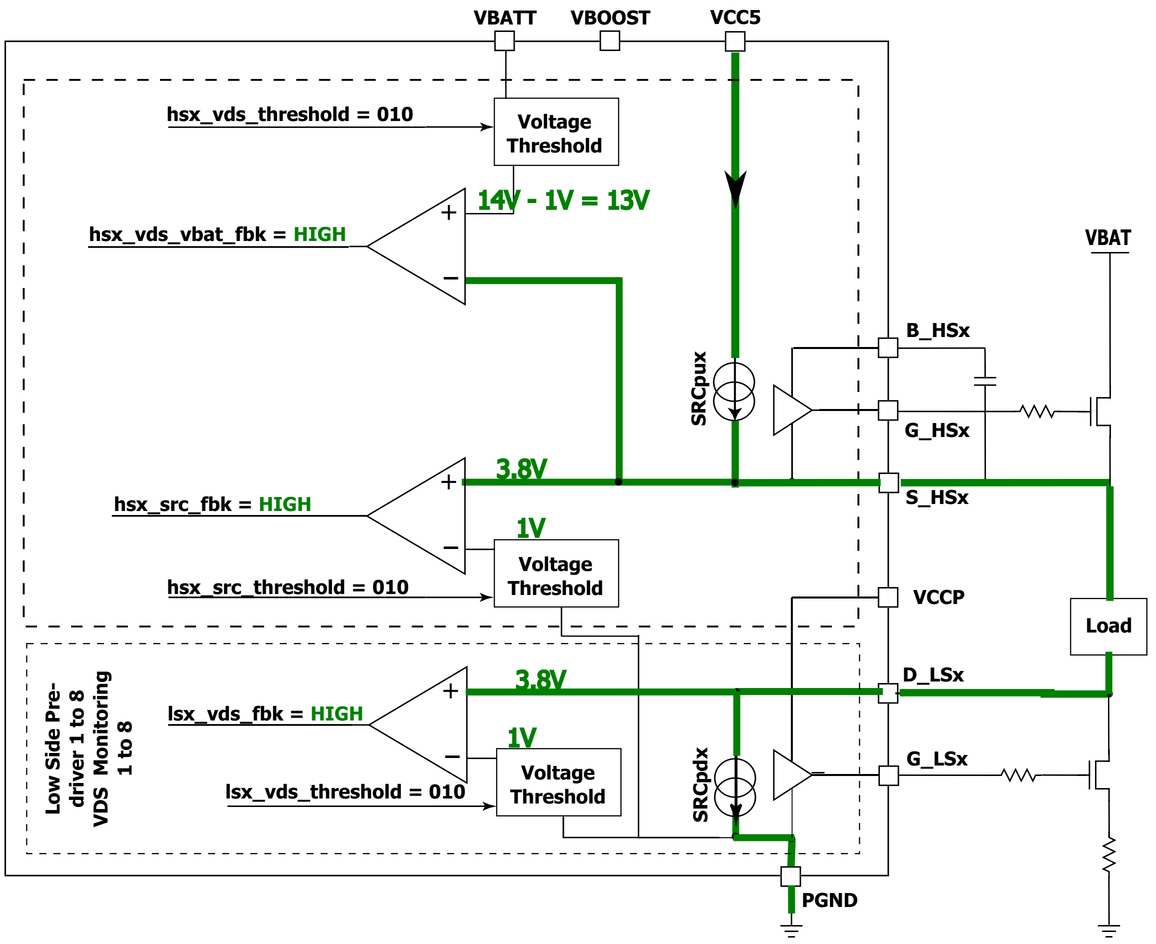

Figure 4. Idle Normal Mode

During normal operation, a current limited pull-up voltage source (SRC_{PUX}) generates a voltage on S_HSx (min. 3.8 V). Its current goes to the load and to a pull-down current source on D_LSx, generating a 3.8 V min. voltage. Drain source voltage on the high-side is not monitored directly, and since there is no pin for the drain, monitoring is directly done from V_{BAT} or V_{BOOST}, and only HS2, HS4, and HS6 can use V_{BOOST} as a reference. Voltage thresholds are selected to be lower than the voltage generated.

Table 7. Normal Mode Truth Table

Error Case	LSx_vds_fbk	HSx_src_fbk	Hsx_vds_Vbat_fbk	S_HSx voltage V_{BAT}	D_LS_x voltage
Normal mode	1	1	1	3.8	3.8

5.1.2 High-side Source or Low-side Drain Shorted to GND

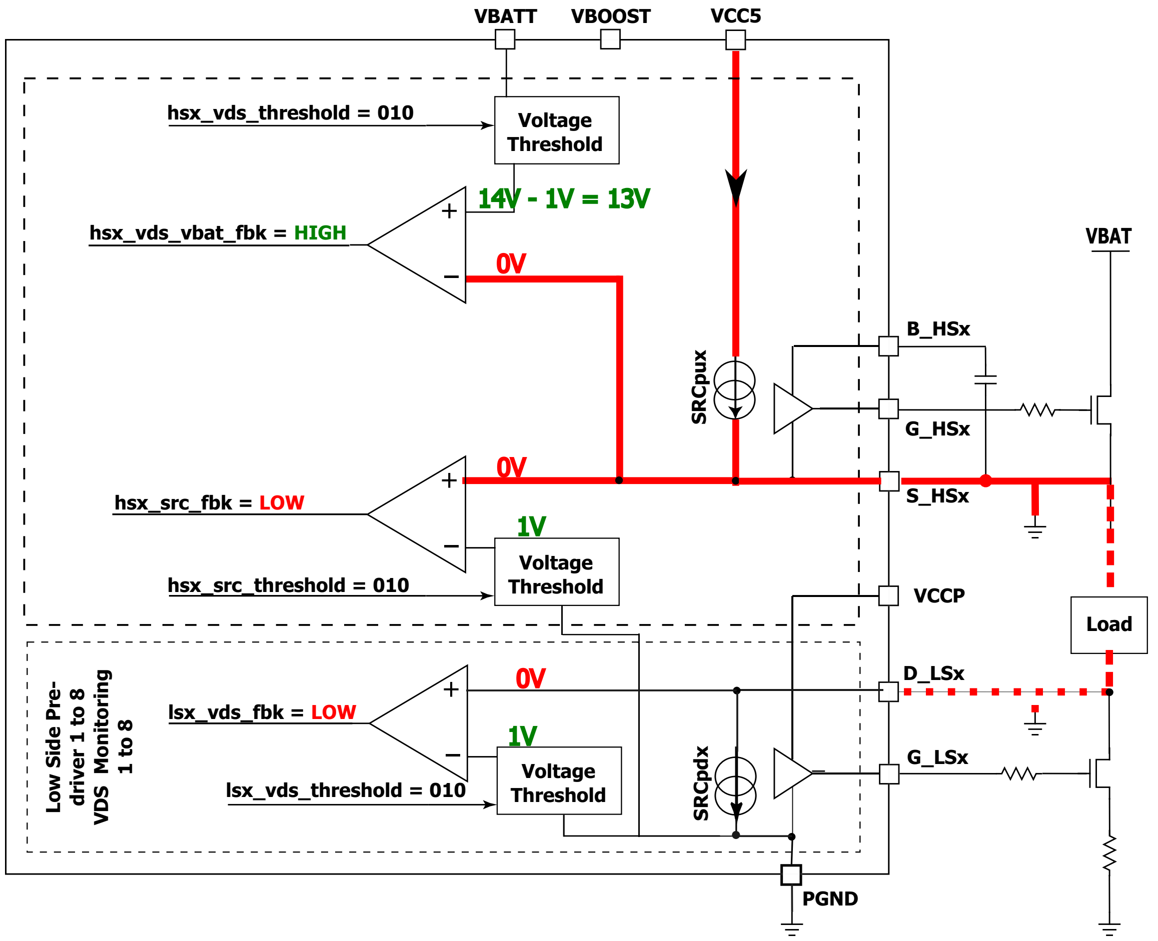

Figure 5. High-side Source or Low-side Drain Shorted

In cases where the High Source (S_HSx) shorts to GND or the Low-side Drain (D_LSx) shorts to GND, the current limited voltage source pulls to ground, and the voltage on S_HSx and D_LSx is 0 V. A diagnostic error is detected, and since the high-side V_{SRC} and low-side V_{DS} feedback are low, the bank does not turn ON.

Table 8. S_HSx or D_LSx Shorted to GND Truth Table

Error Case	LSx_vds_fbk	HSx_src_fbk	Hsx_vds_Vbat_fbk	S_HSx voltage V_{BAT}	D_LS_x voltage
Normal mode	1	1	1	3.8	3.8
D_LS GND short	0	0	1	0	0

5.1.3 Drain Source Low-side Shorted to GND

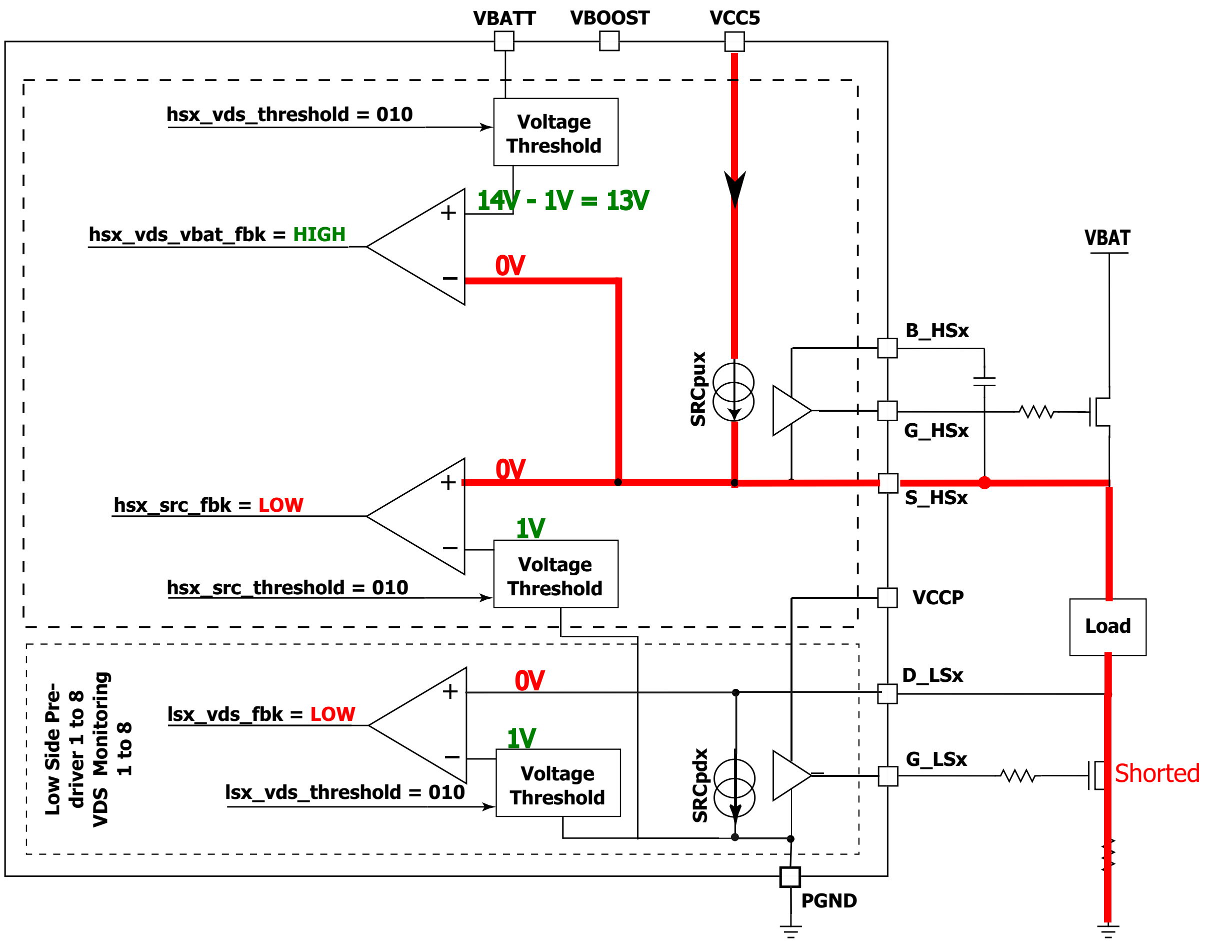

Figure 6. Low-side Drain Source Shorted

In cases where the Low-side Drain Source shorts, D_LSx pulls to 0 V, the current limited voltage source pulls to ground, and the voltage on S_HSx and D_LSx is 0 V. A diagnostic error is detected, since the high-side V_{SRC} and low-side V_{DS} feedback are low.

Table 9. Drain Source Low-side Shorted Truth Table

Error Case	LSx_vds_fbk	HSx_src_fbk	Hsx_vds_Vbat_fbk	S_HSx voltage V_{BAT}	D_LS_x voltage
Normal mode	1	1	1	3.8	3.8
Low-side Drain Source Short	0	0	1	0	0

5.1.4 Drain Source High-side Shorted to VBAT

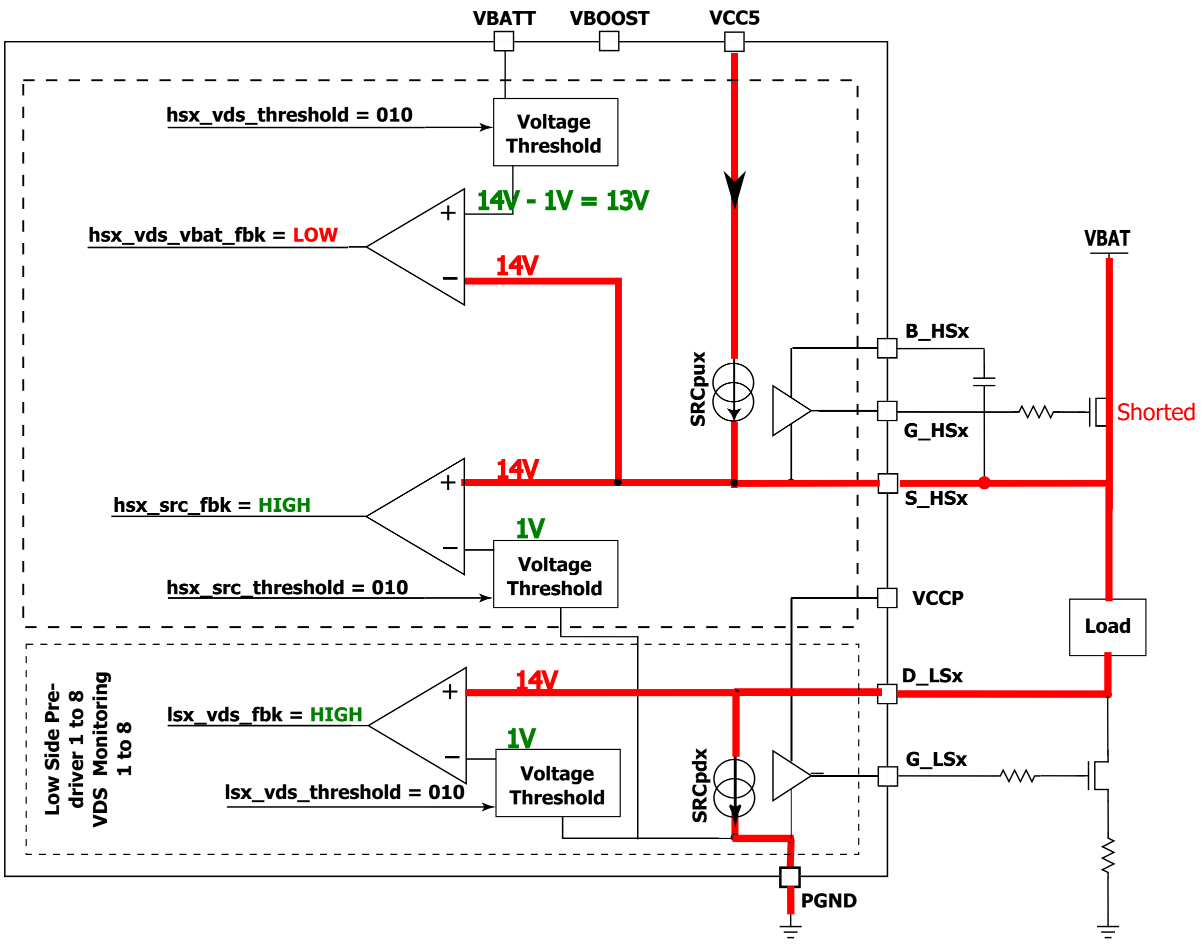

Figure 7. High-side Drain Source Shorted

The diagnostic fails in cases where the high-side drain source shorts. As a consequence, S_HSx and D_LSx pulls up to V_{BAT}, the difference between drain and source on the high-side goes negative, resulting in low feedback on the high-side V_{DS}.

Table 10. Drain Source High-side Shorted Truth Table

Error Case	LSx_vds_fbk	HSx_src_fbk	Hsx_vds_Vbat_fbk	S_HSx voltage V_{BAT}	D_LS_x voltage
Normal mode	1	1	1	3.8	3.8
High-side Drain Source Short	1	1	0	14	14

5.1.5 Open Load

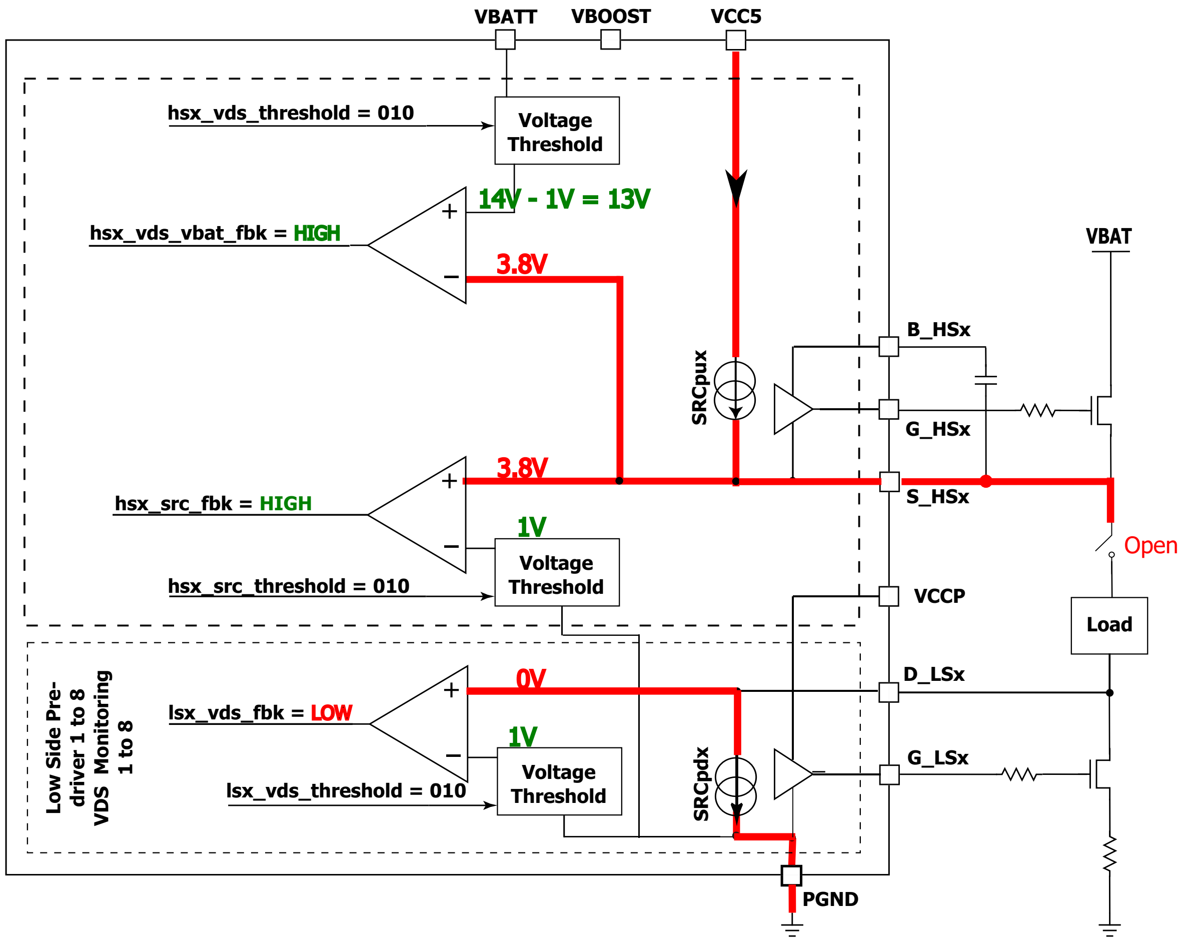

Figure 8. Open Load

If one of the sides of the load is not connected properly, there is no current path between S_HSx and D_LSx. The voltage on D_LSx is forced to ground, because of the SRC_{PDX} current pull-down. The diagnostic fails, because the low-side V_{DS} feedback is low.

Table 11. OpenLoad Truth Table

Error Case	LSx_vds_fbk	HSx_src_fbk	Hsx_vds_Vbat_fbk	S_HSx voltage V_{BAT}	D_LS_x voltage
Normal mode	1	1	1	3.8	3.8
OpenLoad on Low-side	0	1	1	3.8	0
OpenLoad on High-side	0	1	1	3.8	0

5.1.6 Faults Not Detected in Idle Phase

There are different cases that cannot be detected in the idle phase:

- High-side V_{BAT} or V_{BOOST} open: not possible to be detected since both are OFF in idle phase
- LS open: not possible to be detected since it is OFF in idle phase
- Short between load pins: not possible to detect, because it allows the bias current to pass through

All these faults are detected in the actuation mode only.

5.2 Actuation Phase

The bias voltage used for idle diagnostics is kept ON, to predict the voltage on each pin even if the MOSFETs are OFF. In this case, when the MOSFETs are OFF, there is a 3.8 V voltage on the high-side source. In each case, if an error occurs, the PT2000 turns bank 1 OFF, keeps it OFF, and sets the Status_reg_uc0 register (105h) bit 6 high until the MCU writes a 1 to the control register bit 6.

5.2.1 Actuation Diagnostics Peak and Hold phase (HS Boost OFF, HS Bat ON, LS ON)

5.2.1.1 Normal Mode

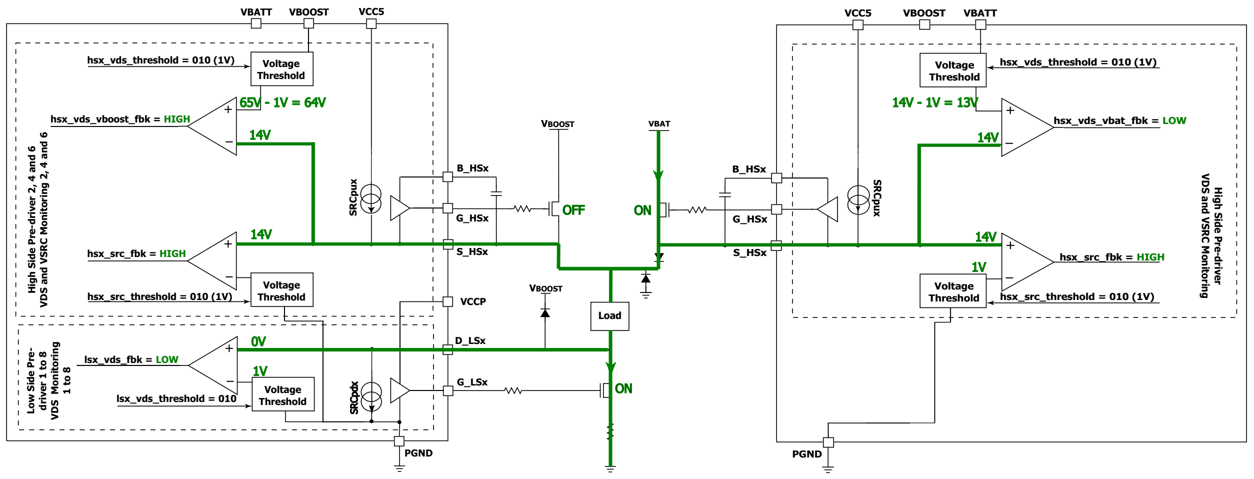

Figure 9. Normal Mode Peak and Hold

During peak and hold phase, the low-side is fully ON and V_{BAT} high-side is controlled in PWM to regulate the current inside the injector.

To have a device as flexible as possible, detection error during automatic diagnostics is configurable for each low-side and high-side. To configure which case leads the device to an error, it is necessary to set the registers "Error_table" for each low-side V_{DS}, high-side V_{DS}, and high-side S_{RC} where diagnostics are needed (refer to Diagnostics Configuration Registers). In Normal mode, the PT2000 comparator outputs should be in the following state:

Table 12. Actuation in Normal Mode Truth Table

Error Case	LSx_vds_fbk	HSx_src_vbat_fbk	Hsx_vds_Vbat_fbk	HSx_vds_Vboost_fbk	HSx_vsrc_Vboost_fbk
Normal mode	0	1	0	1	1

5.2.1.2 High-side (Bat or Boost) Source Shorted to GND

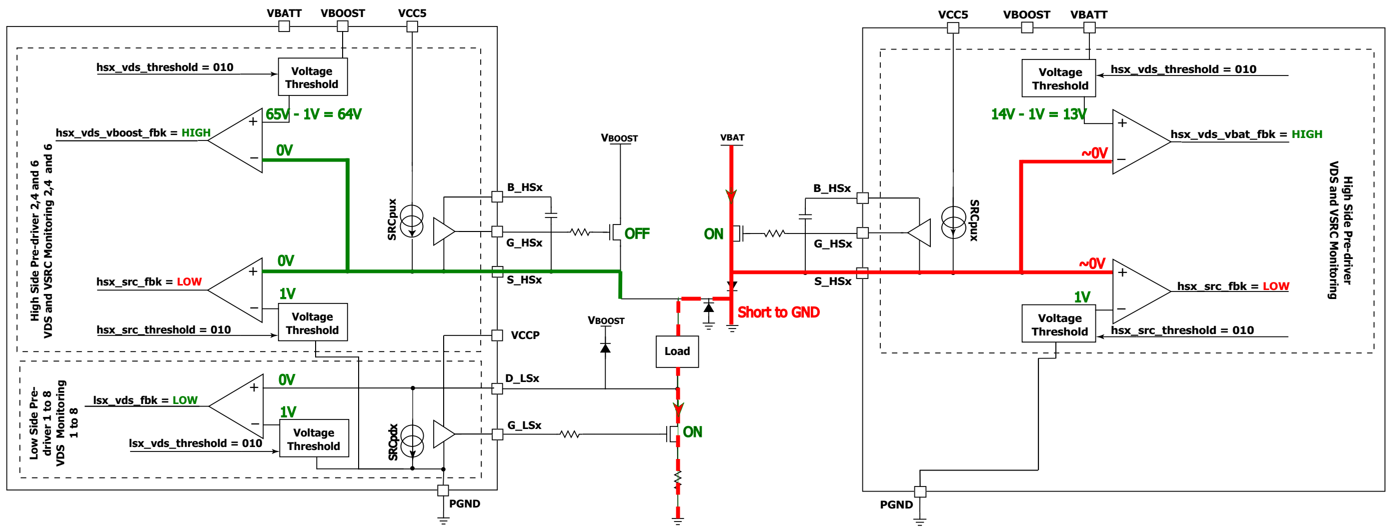

Figure 10. High-side Source Shorted to GND

When S_HSx shorts to GND, the PT2000 detects an overcurrent, due to the V_{DS} monitoring on the V_{BAT} high-side. The high-side shuts down as soon as the current is substantial enough to generate a higher drop across the MOSFET than the threshold. In this case, it's important to set a threshold (1.0 V, in this case) and a filter time to the lowest value allowed by the application, to quickly detect it (refer to Filter time). The automatic diagnostic fails, because high-side V_{DS} feedback is high.

Table 13. High-side Source Shorted to GND Truth Table

Error Case	LSx_vds_fbk	HSx_src_vbat_fbk	Hsx_vds_Vbat_fbk	HSx_vds_Vboost_fbk	HSx_vsrc_Vboost_fbk
Normal mode	0	1	0	1	1
S_HSx V_{BAT} or V_{BOOST} GND Short	0	0	1	1	0

This case is also applicable when there is a short between the two load pins, substantial current flows inside V_{BAT} HS and LS until the difference between drain and source is higher than the threshold.

5.2.1.3 High-side V_{BOOST} Short Drain Source

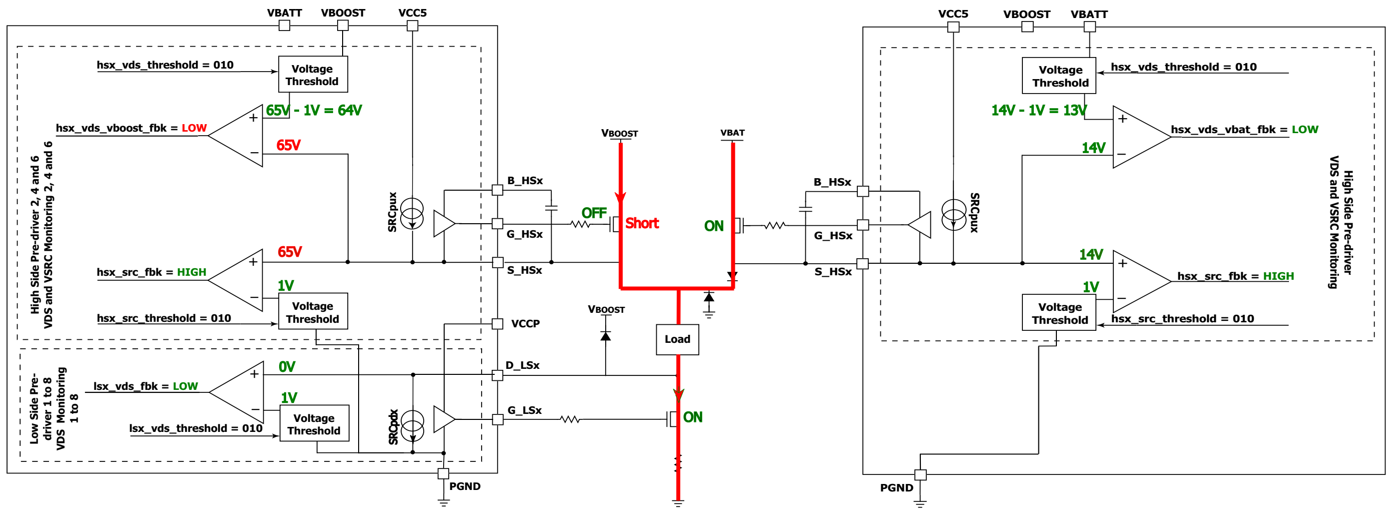

Figure 11. High-side V_{BOOST} Shorted

During a peak and hold phase, V_{BOOST} high-side should be OFF, but if there is a short-circuit between the drain and source, the voltage on the V_{BOOST} high-side source rises to V_{BOOST}. The automatic diagnostic fails, because V_{DS} on the V_{BOOST} high-side is low.

Table 14. High-side V_{BOOST} Drain Source Shorted Truth Table

Error Case	LSx_vds_fbk	HSx_src_vbat_fbk	Hsx_vds_Vbat_fbk	HSx_vds_Vboost_fbk	HSx_vsrc_Vboost_fbk
Normal mode	0	1	0	1	1
HSvboot Drain Source short	0	1	0	0	1

5.2.1.4 High-side V_{BAT} Open

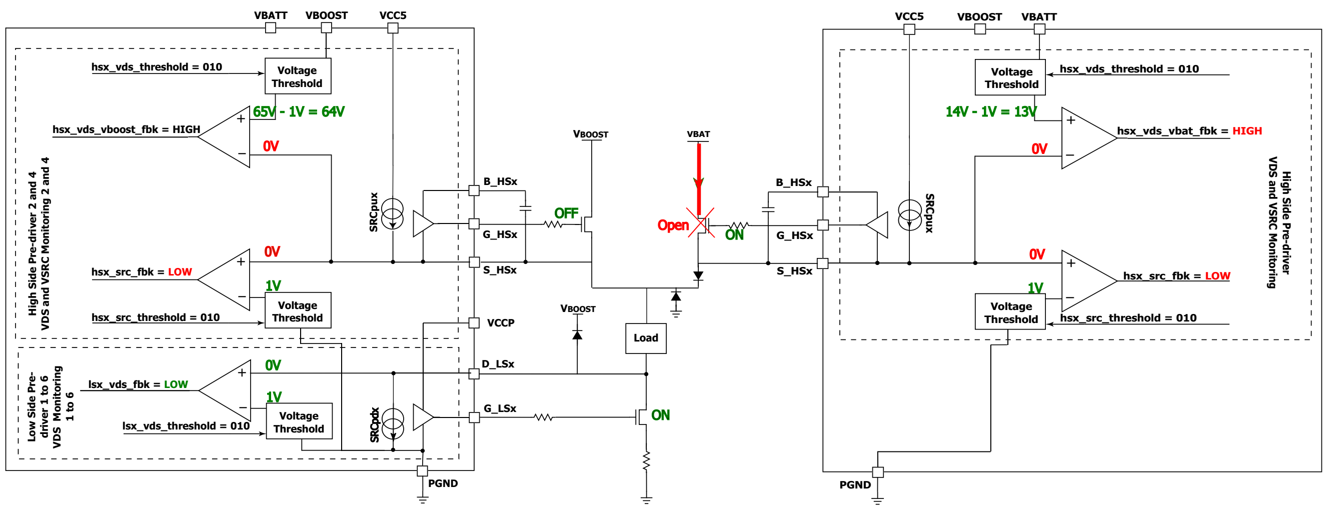

Figure 12. High-side Open

During the peak and hold phase, high-side V_{BAT} is ON. If it’s open or not controlled properly, S_HSx voltage will be lower than expected. The automatic diagnostic fails because on V_{BAT} high-side, the V_{DS} feedback is high and V_{SRC} feedback is low.

Table 15. High-side V_{BAT} Open Truth Table

Error Case	LSx_vds_fbk	HSx_src_vbat_fbk	Hsx_vds_Vbat_fbk	HSx_vds_Vboost_fbk	HSx_vsrc_Vboost_fbk
Normal mode	0	1	0	1	1
HS vbat open	0	0	1	1	0

This case is only detectable in actuation mode.

5.2.1.5 Low-side Open

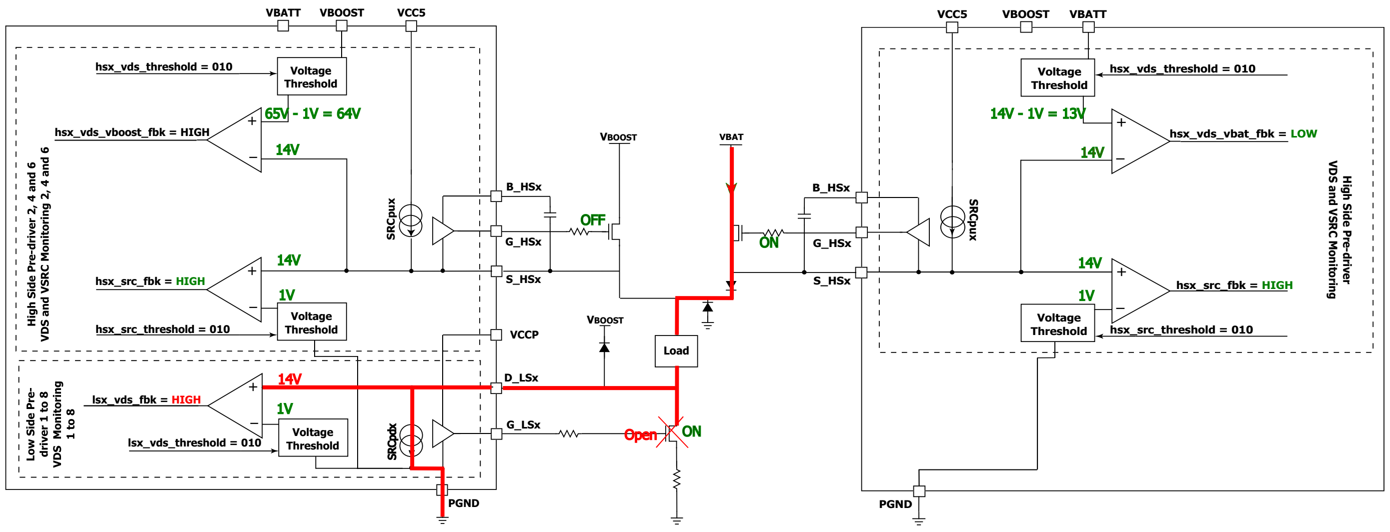

Figure 13. Low-side Open

With the low-side open, current on the D_LSx pin flows through the load to the internal pull-down (SCR_{PDX}) and the voltage rises to V_{BAT}. Automatic diagnostics fail, because the low-side V_{DS} feedback is high.

Table 16. Low-side Open Truth Table

Error Case	LSx_vds_fbk	HSx_src_vbat_fbk	Hsx_vds_Vbat_fbk	HSx_vds_Vboost_fbk	HSx_vsrc_Vboost_fbk
Normal mode	0	1	0	1	1
LS Open	1	1	0	1	1

This is one case only detectable in actuation mode.

5.2.1.6 Drain Low-side Shorted to V_{BAT} or V_{BOOST}

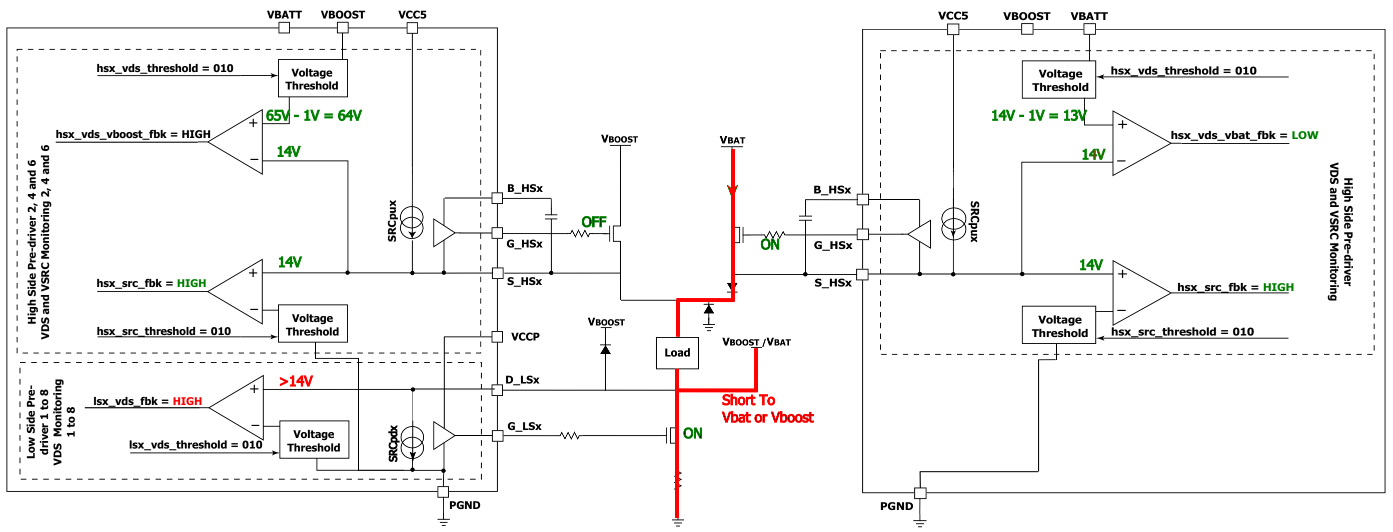

Figure 14. Drain Low-side Shorted

When the low-side drain is shorted to V_{BAT} or V_{BOOST} (low probability case), the voltage on D_LSx rises to V_{BAT}/V_{BOOST}. Voltage thresholds and filter times must be set to the lowest value allowed by the application, to detect the error as fast as possible. The automatic diagnostic fails, because the low-side V_{DS} feedback is high.

Table 17. D_LS Battery Short Truth Table

Error Case	LSx_vds_fbk	HSx_src_vbat_fbk	Hsx_vds_Vbat_fbk	HSx_vds_Vboost_fbk	HSx_vsrc_Vboost_fbk
Normal mode	0	1	0	1	1
D_LS Batt short Depending on external MOS behavior	1	1	0	1	1

5.2.1.7 Cases Undetectable During Peak and Hold Phase

There are different cases that can't be detected during peak and hold phase:

- Drain low-side shorted to GND: not detectable since the low-side is ON, in this case (detectable in idle phase)
- High-side V_{BAT} drain source shorted: not detectable since the high-side is ON, in this case (detectable in idle phase)
- High-side V_{BOOST} open: not detectable since the high-side V_{BOOST} is OFF in this mode (detectable during V_{BOOST} phase)

5.2.2 Actuation Diagnostics Boost Phase (HS Boost ON, HS Bat ON, LS ON)

During Boost phase, boost voltage is used to turn the injector ON as fast as possible, high-side V_{BOOST} and low-side are ON. The high-side V_{BAT} source needs to be turned ON, to avoid errors during diagnostics, which has no impact on the application. Another option would be to disable automatic diagnostics on the high-side V_{BAT} source during boost phase.

In this example, the PT2000 automatic diagnostics are configured using instruction “endiags” (refer to Application Source Code). During actuation phase, automatic diagnostics monitor HS V_{BAT} V_{DS}, HS V_{BAT} V_{SRC}, HS V_{BOOST} V_{DS}, and LS V_{DS} continuously. To simplify the diagnostics code, HS V_{BAT} is kept ON during Boost phase to avoid unwanted errors on the V_{BAT} source.

5.2.2.1 Normal Mode

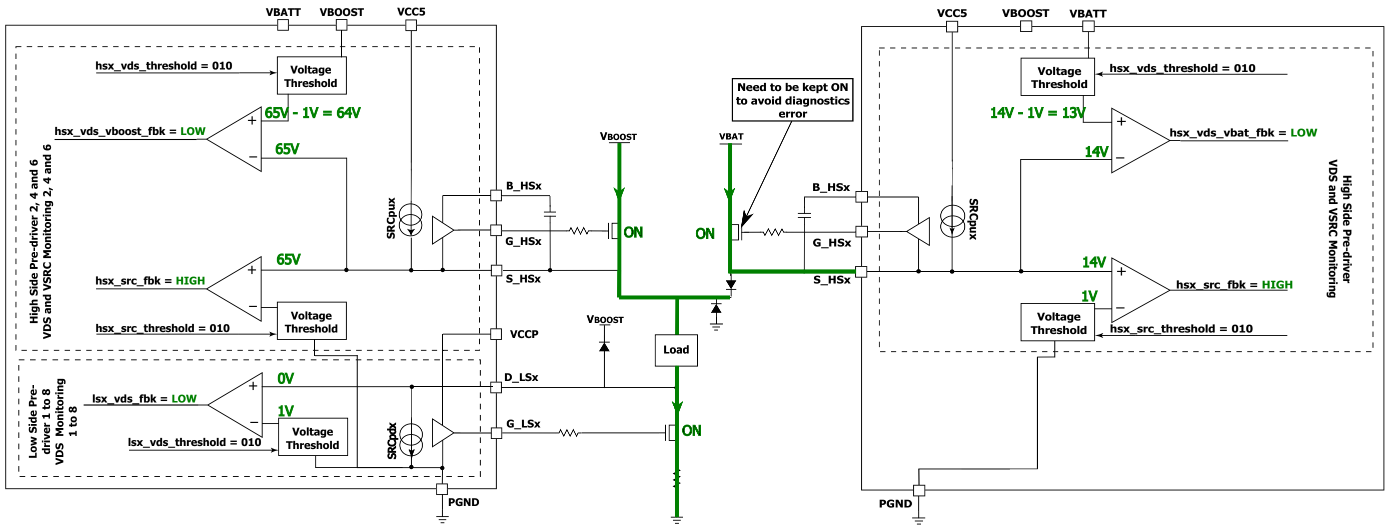

Figure 15. Boost Phase Normal Mode

During boost phase, the high-side boost is fully ON to reach boost current as fast as possible, high-side V_{BAT} is ON (for diagnostic purposes), and the low-side is fully ON. As with the peak and hold phase, the high-side V_{BOOST} error table must be set-up accordingly (see Diagnostics Configuration Registers).

Table 18. Boost Phase Normal Mode Truth Table

Error Case	LSx_vds_fbk	HSx_src_vbat_fbk	Hsx_vds_Vbat_fbk	HSx_vds_Vboost_fbk	HSx_vsrc_Vboost_fbk
Normal mode	0	1	0	0	1

5.2.2.2 High-side Boost Source Shorted to GND

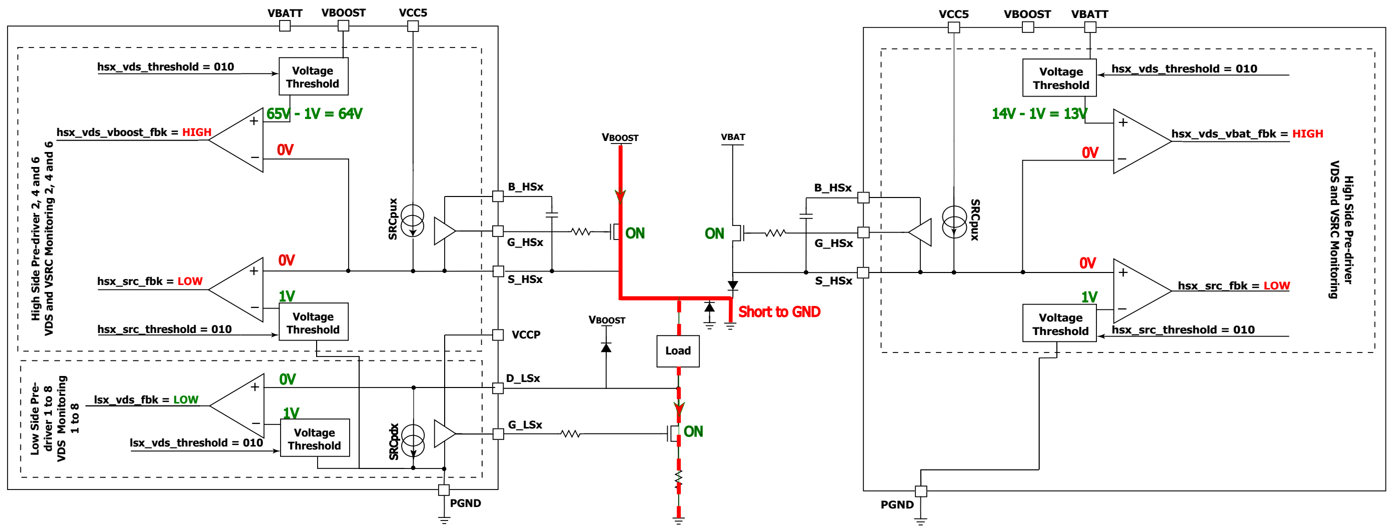

Figure 16. High-side Source Shorted to GND

The same behavior as in the Peak and Hold phase except this time the short is from V_{BOOST} to GND. The comparator threshold must be set as low as possible to detect the overcurrent faster and avoid any damage to the MOS. The automatic diagnostic on high-side V_{BOOST} fails because V_{DS} monitoring is high.

Table 19. High-side Boost Source Shorted to GND Truth Table

Error Case	LSx_vds_fbk	HSx_src_vbat_fbk	Hsx_vds_Vbat_fbk	HSx_vds_Vboost_fbk	HSx_vsrc_Vboost_fbk
Normal mode	0	1	0	0	1
HSvbat Drain Source short	0	0	1	1	0

5.2.2.3 High-side V_{BOOST} Open

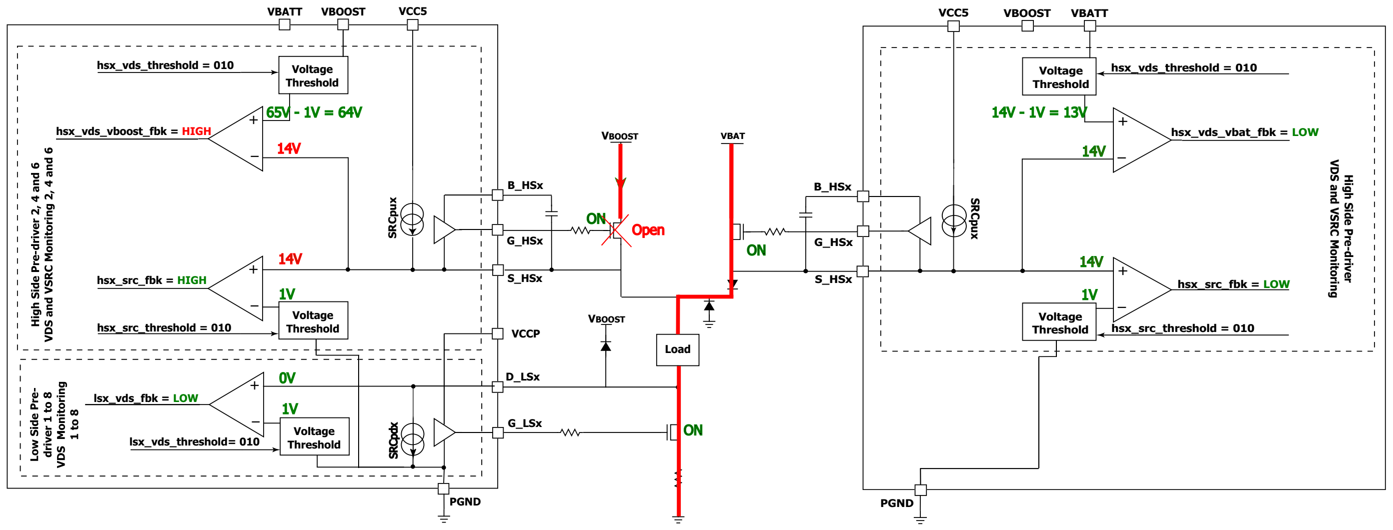

Figure 17. High-side V_{BOOST} Open

When V_{BOOST} high-side is open, the voltage on S_HSx floats and forced to 0 V, due to the parasitic leakage on the S_HSx pin. The automatic diagnostic on high-side V_{BOOST} fails because V_{DS} feedback is high.

Table 20. High-side Boost Open Truth Table

Error Case	LSx_vds_fbk	HSx_src_vbat_fbk	Hsx_vds_Vbat_fbk	HSx_vds_Vboost_fbk	HSx_vsrc_Vboost_fbk
Normal mode	0	1	0	0	1
HS vboot open	0	1	0	1	1

This case is undetectable in idle phase.

5.2.2.4 Low-side Open

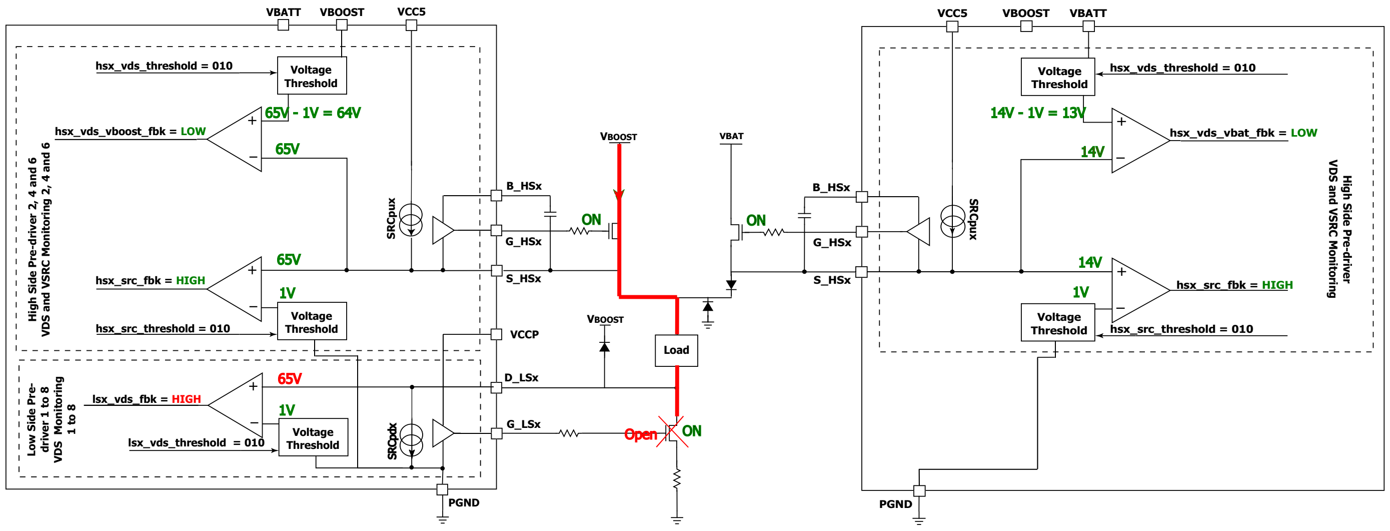

Figure 18. Low-side Open

When the low-side is not connected properly, the voltage on D_LSx is around a V_{BOOST} of 65 V. The automatic diagnostic fails due to V_{DS} feedback on the low-side.

Table 21. Low-side Open Truth Table

Error Case	LSx_vds_fbk	HSx_src_vbat_fbk	Hsx_vds_Vbat_fbk	HSx_vds_Vboost_fbk	HSx_vsrc_Vboost_fbk
Normal mode	0	1	0	0	1
LS open	1	1	0	0	1

This case is undetectable in idle phase.

5.2.2.5 Drain Low-side Shorted to V_{BAT} or V_{BOOST}

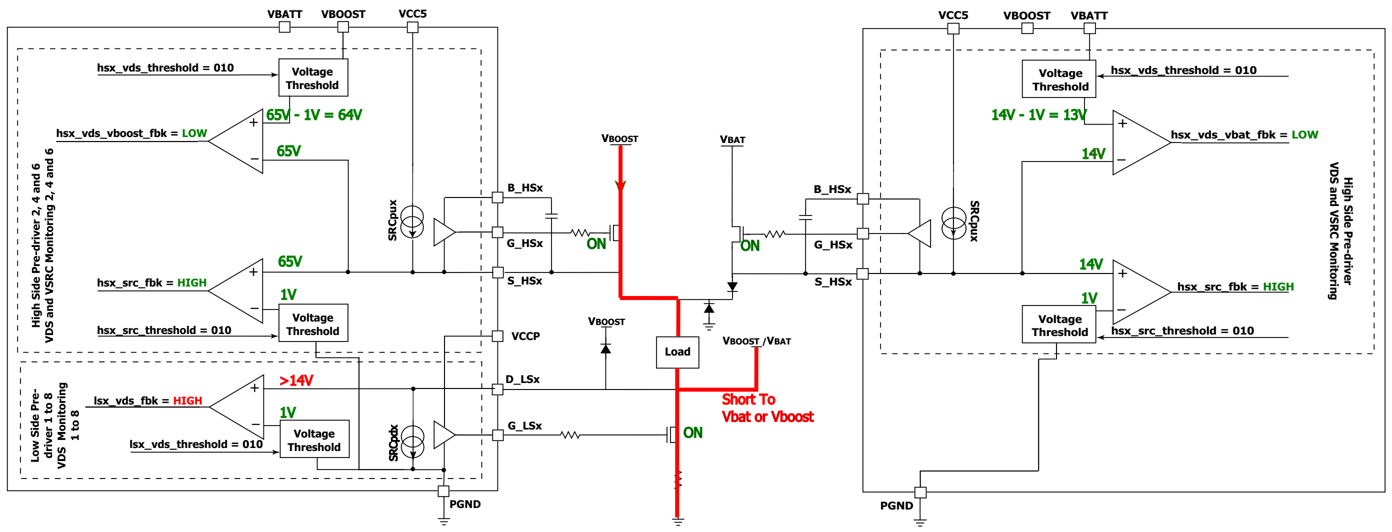

Figure 19. Drain Low-side Shorted to V_{BAT} or V_{BOOST}

This is the same behavior as in the Peak and Hold phase, when the drain low-side is shorted to V_{BAT} or V_{BOOST} (Low probability), with a short to GND on V_{BOOST} or V_{BAT}. The automatic diagnostic fails, because the voltage on D_LSx is higher than the V_{DS} threshold.

Table 22. Drain Low-side Shorted to V_{BAT} or V_{BOOST} Truth Table

Error Case	LSx_vds_fbk	HSx_src_vbat_fbk	Hsx_vds_Vbat_fbk	HSx_vds_Vboost_fbk	HSx_vsrc_Vboost_fbk
Normal mode	0	1	0	0	1
D_LS Boost/Bat short	1	1	0	0	1

5.2.2.6 Cases Undetectable During Boost Mode

There are different cases undetectable in the Boost phase:

- Drain low-side shorted to GND: not detectable, since the low-side is ON in this case (detectable in the idle phase)
- High-side V_{BAT} or V_{BOOST} drain source shorted: not detectable, since in this case, high-side is ON (detectable in the idle phase)

6 Software

6.1 Interrupt State Machine

The following state diagrams describe how the MCU knows, which interrupt occurred, and which fault was detected during both automatic and software interrupts. For injectors actuation and DCDC state diagram, refer to AN5187.

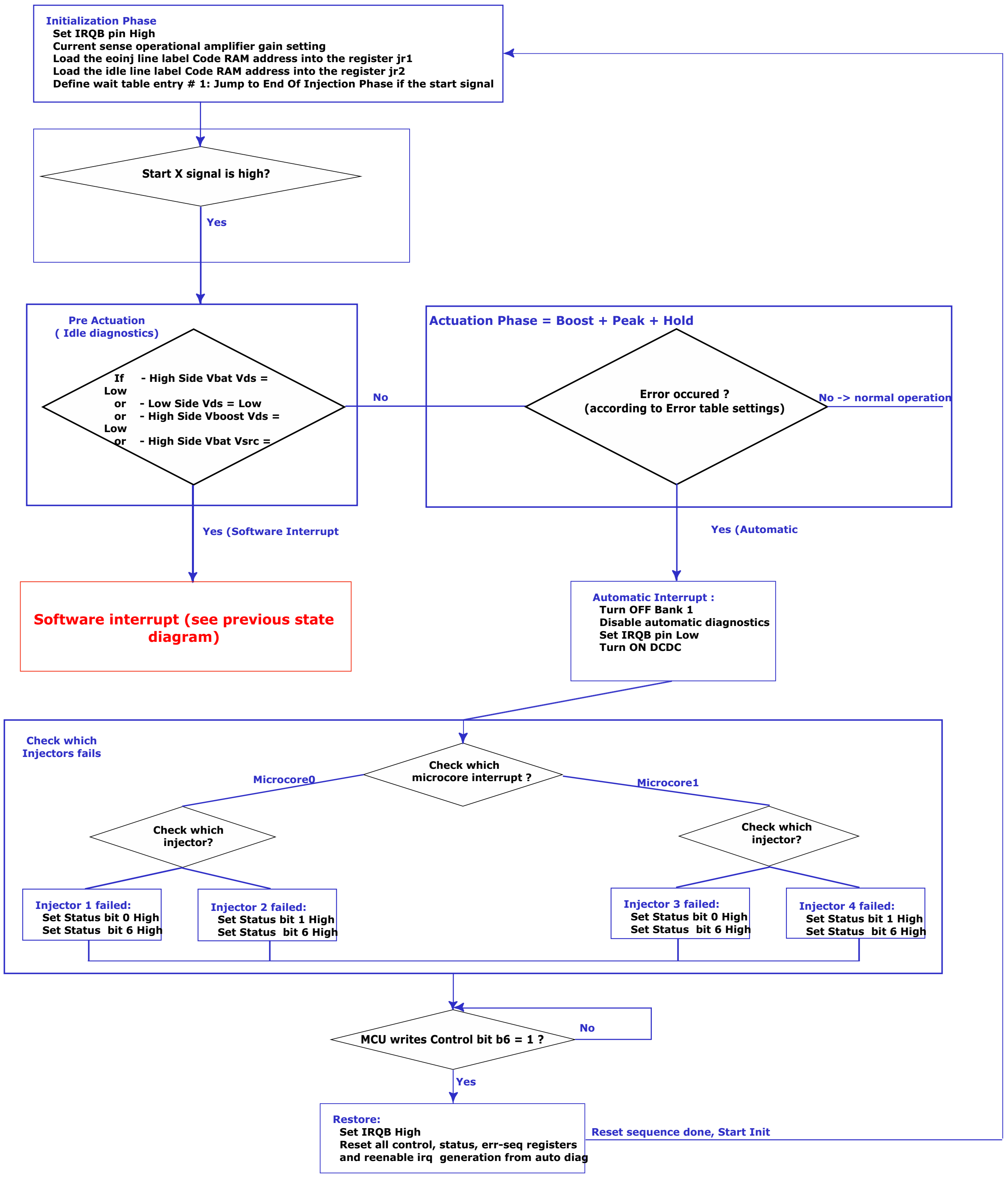
Initialization Phase
Set IRQB pin High
Current sense operational amplifier gain setting
Load the eoinj line label Code RAM address into the register jr1
Load the idle line label Code RAM address into the register jr2
Define wait table entry # 1: Jump to End Of Injection Phase if the start signal
Start X signal is high?
Yes
Pre Actuation (Idle diagnostics)
If - High Side Vbat Vds = Low
or - Low Side Vds = Low
or - High Side Vboost Vds = Low
or - High Side Vbat Vsrc =
No
Actuation Phase = Boost + Peak + Hold
Error occured ? (according to Error table settings)
No -> normal operation
Yes (Software Interrupt
Yes (Automatic
Software interrupt (see previous state diagram)
Automatic Interrupt :
Turn OFF Bank 1
Disable automatic diagnostics
Set IRQB pin Low
Turn ON DCDC
Check which Injectors fails
Check which microcore interrupt ?
Microcore0
Microcore1
Check which injector?
Check which injector?
Injector 1 failed: Set Status bit 0 High Set Status bit 6 High
Injector 2 failed: Set Status bit 1 High Set Status bit 6 High
Injector 3 failed: Set Status bit 0 High Set Status bit 6 High
Injector 4 failed: Set Status bit 1 High Set Status bit 6 High
MCU writes Control bit b6 = 1 ?
No
Yes
Restore:
Set IRQB High
Reset all control, status, err-seq registers
and reenable irq generation from auto diag
Reset sequence done, Start Init

Figure 20. Software Interrupt State Machine

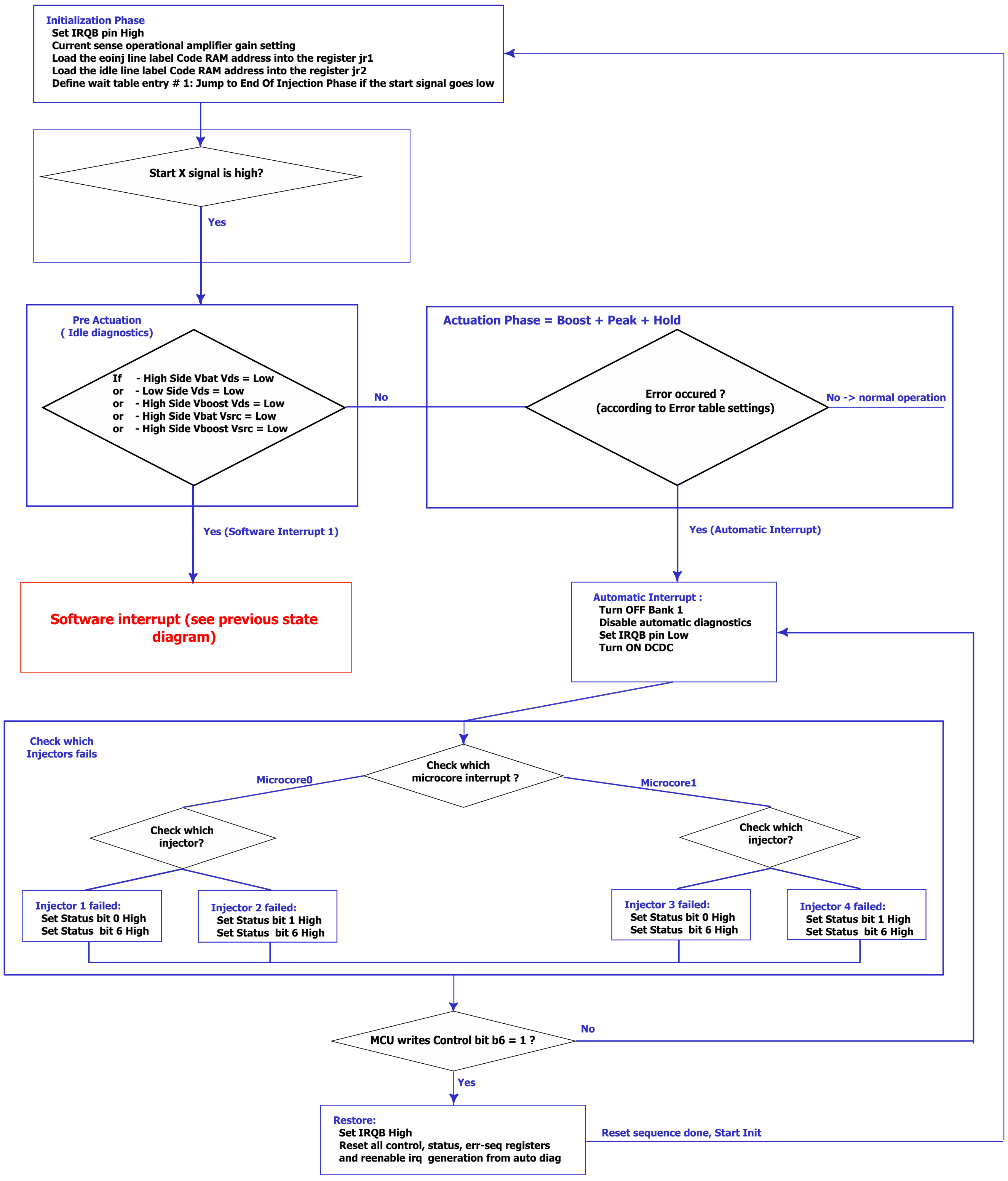
Initialization Phase
Set IRQB pin High
Current sense operational amplifier gain setting
Load the eoinj line label Code RAM address into the register jr1
Load the idle line label Code RAM address into the register jr2
Define wait table entry # 1: Jump to End Of Injection Phase if the start signal goes low
Start X signal is high?
Yes
Pre Actuation
(Idle diagnostics)
If - High Side Vbat Vds = Low
or - Low Side Vds = Low
or - High Side Vboost Vds = Low
or - High Side Vbat Vsrc = Low
or - High Side Vboost Vsrc = Low
No
Actuation Phase = Boost + Peak + Hold
Error occured ?
(according to Error table settings)
No -> normal operation
Yes (Software Interrupt 1)
Yes (Automatic Interrupt)
Software interrupt (see previous state diagram)
Automatic Interrupt :
Turn OFF Bank 1
Disable automatic diagnostics
Set IRQB pin Low
Turn ON DCDC
Check which Injectors fails
Check which microcore interrupt ?
Microcore0
Microcore1
Check which injector?
Check which injector?
Injector 1 failed:
Set Status bit 0 High
Set Status bit 6 High
Injector 2 failed:
Set Status bit 1 High
Set Status bit 6 High
Injector 3 failed:
Set Status bit 0 High
Set Status bit 6 High
Injector 4 failed:
Set Status bit 1 High
Set Status bit 6 High
MCU writes Control bit b6 = 1 ?
No
Yes
Restore:
Set IRQB High
Reset all control, status, err-seq registers
and reenable irq generation from auto diag
Reset sequence done, Start Init

Figure 21. Automatic Interrupt State Machine

6.2 General Registers Setup

Unless specified, use the register settings described in AN5187. Only registers related to diagnostics and interrupts are described in the following chapter.

6.2.1 Main Configuration Registers

Table 23. Driver_config Register (1A5h)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved			Overtemp_irq_en	Drv_en_irq_en	Vboost_irq_en	Vcc5_irq_en	Vccp_irq_en	Iret_en	Irq_uc1_ch3_en	Irq_uc0_ch3_en	Irq_uc1_ch2_en	Irq_uc0_ch2_en	Irq_uc1_ch1_en	Irq_uc0_ch1_en	Irq_uc_en
Value	X	X	X	X	X	X	1	X	1	X	X	X	X	X	X	1

This register doesn’t need to be set for the diagnostic on the external MOSFET, since it is handled in the microcode directly. If an error is detected, it forces IRQB low using the microcode. The return address (iret) is also determined in the microcode.

As an example, set vcc5_irq_en to ‘1’, to force IRQB low in cases of undervoltage on VCC5. When the undervoltage is gone, the IRQB pin is kept low until the user writes a ‘1’ in the uv_vcc5 bit (Driver_status register (1B2h)).

6.2.2 6.1.2 IO Configuration Registers

This register (one for each microcore) selects the feedback by which each microcore is enabled. Setting the bit to ‘1’ generates an interrupt towards UcXChY, in case an error is detected on the HSx or LSx feedback.

Table 24. Fbk_sens_uc0ch1 Register_Part1 (154h)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved	Hs7_Vsrc_sens	Hs6_Vsrc_sens	Hs5_Vsrc_sens	Hs4_Vsrc_sens	Hs3_Vsrc_sens	Hs2_Vsrc_sens	Hs1_Vsrc_sens	Reserved	Hs7_Vds_sens	Hs6_Vds_sens	Hs5_Vds_sens	Hs4_Vds_sens	Hs3_Vds_sens	Hs2_Vds_sens	Hs1_Vds_sens
Value	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1

Table 25. Fbk_sens_uc0ch1 Register_Part2(155h)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved								Ls8_Vds_sens	Ls87_Vds_sens	Ls6_Vds_sens	Ls5_Vds_sens	Ls4_Vds_sens	Ls3_Vds_sens	Ls2_Vds_sens	Ls1_Vds_sens
Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

In this particular application (see the schematics in KITPT2000FRDM6C), microcore 0 Channel 1 controls HS1 as high-side V_{BAT} and HS2 as high-side V_{BOOST}, LS1 and LS2. As show in table Table 24, an interrupt is generated if an error occurs on LS2 V_{DS}, LS1 V_{DS}, HS2 V_{DS}, HS1 V_{DS}, HS2 V_{SRC}, and HS1 V_{SCR}.

6.2.2.1 Threshold settings

Each comparator threshold is set on three bits. The V_{DS} and V_{SRC} thresholds are defined by registers 16Bh and 16Ch, for the high-side pre-drivers, and by the Vds_threshlod_ls_1 (16Fh) for the low-side pre-drivers.

As described during fault description, these threshold must be set according to the external MOSFET and maximum current level used in the application. As with KITPT2000FRDM6C, $R_{DS(ON)}$ MOSFET ~40 mΩ (worst case condition). The maximum current used in this application is 14 A, and overcurrent detection (using V_{DS} monitoring) must be set at around 40% higher than max. current allowed (20 A).

Table 26. V_{DS} and V_{SRC} Monitoring Typical Threshold Selection

Threshold(2:0)	VSRC and VDS (V)
0000	0.00
1001	0.10
1010	0.20
1011	0.30
1100	0.40
0001	0.50
0010	1.0
0011	1.5
0100	2.0
0101	2.5
0110	3.0
0111	3.5

High-side V_{DS} Threshold Calculation

V_{DS} Threshold (HS) = Overcurrent x $R_{DS(on)}$ = 20 A x 0.040 Ω = 0.8 V → **1.0 V** threshold selected. In this case, overcurrent = 1.0 V/0.04 Ω = 25 A.

Table 27. Vds_threshold_hs Register (16Bh)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Vds_thr_Hs4				Vds_thr_Hs3				Vds_thr_Hs2				Vds_thr_Hs1			
Value	0000				0000				0010				0010			

Low-side V_{DS} Threshold Calculation

Low-side the V_{DS} monitoring is done between D_LSx and GND, sense resistance must be included in the calculation.

V_{DS} threshold (LS) = Overcurrent x ($R_{DS(on)}$ + R_{SENSE}) = 20 A x (0.040 Ω + 0.015 Ω) = 1.1 V → **1.0 V** threshold selected, in this case overcurrent = 1.0 V / 0.055 Ω = 18 A.

Table 28. Vds_threshold_ls_1 Register (16Fh)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Vds thr Ls4				Vds thr Ls3				Vds thr Ls2				Vds thr Ls1			
Value	0000				0000				0010				0010			

High-side SRC Threshold

V_{SRC} is used mostly during idle phases to understand the type of fault present. It is better to keep the “detection threshold” far from the polarization condition. During actuation in this application, recirculation is done through a diode, keeping the voltage of the HS source below ground. In this case, any V_{SRC} voltage is ok, to prevent false diagnostics. In order to avoid detecting noise and to be far from the 3.8 V threshold, the PT2000 V_{SRC} threshold is set to 1.0 V

Table 29. Vsrc_threshold_hs Register (18Bh)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Vsrc_thr_Hs4				Vsrc_thr_Hs3				Vsrc_thr_Hs2				Vsrc_thr_Hs1			
Value	0000				0000				0010				0010			

6.2.3 Channel 1 Configuration Registers

Unless specified, use the same settings specified in AN4849.

Table 30. Ctrl_reg_uc0 Control Registers for the Microcore 0 (101h)

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**control_register_shared**								**control_register**							
Value	X	X	X	X	X	X	X	X	SW Interrupt Unlock	Auto Interrupt Unlock	Boost Error Unlock	Hold Error Unlock	X	X	X	X

control_register: Control bits 4,5,6,and 7 are used to control the turn ON of the bank after a fault occurs

- B4: if START pin is still high after t_{HOLD_TOT} is reached (see Application Source Code)
- B5: if I_{BOOST} is not reached before the specified time
- B6: if errors are detected during actuation (automatic diagnostics)
- B7: if errors are detected during pre-actuation phase (idle diagnostics)

Entry point for each microcode and interrupt are specified by the following. It corresponds to the CRAM line where each microcore and interrupt start. With the new IDE software provided, this sets automatically using the **Settings Menu** by using the label as an address.

Table 31. Uc0_entry_point Registers (10Ah)

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved						entry_point_address									
Value	000000						100110000									

With the code provided, uc0 channel 1 starts line 2Ch label “init0”, interrupt code should not be taken in account in the entry code.

Table 32. Uc1_entry_point Registers (10Bh)

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved						entry_point_address									
Value	000000						000001000									

With the code provided, uc1 channel 1 starts line 74h label “init1”.

The automatic and software interrupt is at the same address, label "irq_detected" on the example provided. To differentiate them, the irq source register is used.

Table 33. Diag_routine_addr Registers (10Ch, 12Ch)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved				diagnosis_routine_address_uc1						diagnosis_routine_address_uc0					
Value	0000				000000						000000					

- diagnosis_routine_address_uc0: Automatic diagnostics are located at line 0 (label irq_detected)
- diagnosis_routine_address_uc1:Automatic diagnostics are located at line 0 (label irq_detected)

The same settings on software interrupt are needed to specify the location in the CRAM where SW interrupts are handled.

Table 34. Sw_interrupt_routine_addr Registers (10Eh, 12Eh)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sw_irq_falling_edge_start_uc1	sw_irq_rising_edge_start_uc1	sw_irq_falling_edge start_uc0	sw_irq_rising_edge_start_uc0	software_interrupt_routine_address_uc1						software_interrupt_routine_address_uc0					
Value	x	x	x	x	000000						000000					

- software_interrupt_routine_address_uc0: Line 0 in the CRAM
- software_interrupt_routine_address_uc1: Line 0 in the CRAM
- sw_irq_rising_edge_start_uc0: Not used in this example
- sw_irq_falling_edge start_uc0: Not used in this example.

6.3 Diagnostics Configuration Registers

6.3.1 LS1 and LS2 Output Register

6.3.1.1 Filter time

These registers define the automatic diagnostics filtering. Values depend on noise in the application and MOSFET switching time to get stable for reliable feedback when diagnostics start.

Table 35. Lsx_diag_config1 Registers (1C0h, 1C3h)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved		filter_type	filter_length						disable_window						
Reset	00		0	000101						0001110						

- filter_type: Set to 0, in this case, means any different sample resets the filter counter
- filter_length: The filtering time is: t_{FTN} = t_{CK} x (5.0 + 1.0) = 1/6 MHz x 6.0 = 1.0 μs
- disable_window: this 7-bit parameter configures a time period during which any check on the LSx_Vds_feed signal is disabled after any change on the output_command signal. t_{DTL}= t_{CK} x (14 + 4.0) = 1/6 MHz x 18 = 3.0 μs

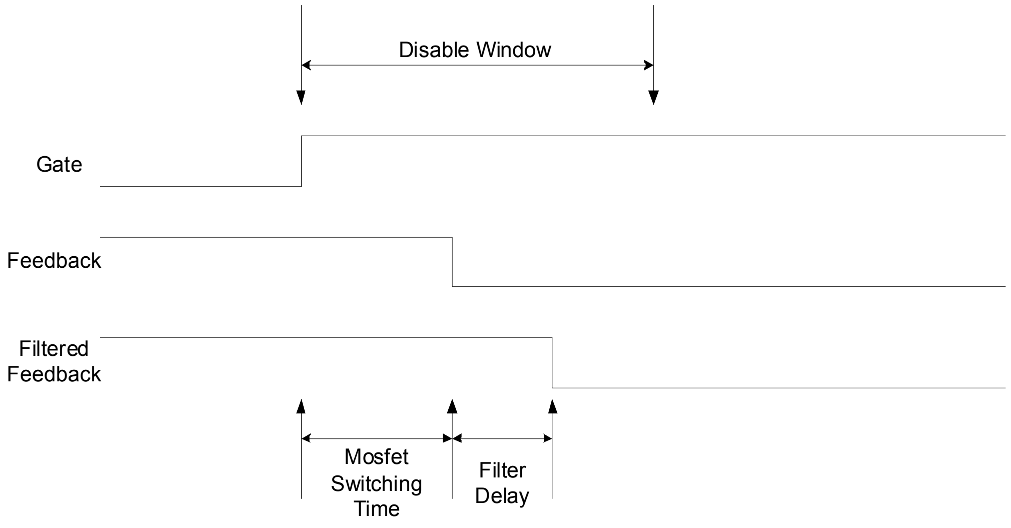

Figure 22. FIlter Time and Disable Windows Diagram

6.3.1.2 Error Table

Using the Diagnostic Descriptions section, error tables can be easily generated,

Table 36. Lsx_diag_config2 Registers (1C1h, 1C4h)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved												error_table			
Reset	000000000000												1001			

- error_table: this 4-bit parameter defines the logical value of an error signal, issued from the output and the related V_{DS} feedback signal. This table defines the output of the coherency check between the driven output and the acquired feedback; a logic one value means there is no coherency in the check, and then an error signal towards the microcore should be generated

Table 37. Error Table for Both Low-sides

	output_command = 0 (Pre-driver switched OFF)	**output_command = 1 (Pre-driver switched ON)**
lsx_vds_fbk = 0 (V_{DS} below threshold)	error_table (0) = 1	error_table (2) = 0 (OK)
lsx_vds_fbk = 1 (V_{DS} above threshold)	error_table (1) = 0 (OK)	error_table (3) = 1

Normal mode in this application:

- Low-side is ON and the V_{DS} comparator should be low
- Low-side is OFF and the V_{DS} comparator should be high

6.3.2 HS1 / HS2 Output Register

6.3.2.1 Filter Time

Use the same filtering as the low-side, since the same MOSFET and slew rates are used for both.

Table 38. Hsx_diag_config_1 Registers (1D8h, 1DBh)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved		Filter_type	filter_length						disable_window						
Reset	00		0	000101						0001110						

- filter_type: Set to 0, in this case, means any different sample resets the filter counter
- filter_length: The filtering time is: $t_{FTN} = t_{CK} \times (5.0 + 1.0) = 1/6\ \text{MHz} \times 6.0 = 1.0\ \mu s$
- disable_window: this 7-bit parameter configures a time period during which any check on the LSx_Vds_feed signal is disabled after any change on the output_command signal. $t_{DTL} = t_{CK} \times (14 + 4.0) = 1/6\ \text{MHz} \times 18 = 3.0\ \mu s$

6.3.2.2 Error Table

HS1 (VBAT) error table

Table 39. Hsx_diag_config_2 Registers (1D9h)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved								error_table_src				error_table_vds			
Reset	00000000								0110				1001			

Table 40. Error Table for High-side V_{DS}

	output_command = 0 (Pre-driver switched OFF)	**output_command = 1 (Pre-driver switched ON)**
hsx_vds_fbk = 0 (V_{DS} below threshold)	error_table_vds (0) = 1	error_table_vds (2) = 0
hsx_vds_fbk= 1 (V_{DS} above threshold)	error_table_vds (1) = 0	error_table_vds (3) = 1

Normal mode in this application:
- High-side is ON and the V_{DS} comparator should be low
- High-side is OFF and the V_{DS} comparator should be high

Table 41. Error Table for High-side V_{SRC}

	output_command = 0 (Pre-driver switched OFF)	**output_command = 1 (Pre-driver switched ON)**
hsx_src_fbk = 0 (V_{SRC} below threshold)	error_table_src (0) = 0	error_table_src (2) = 1
hsx_src_fbk = 1 (V_{SRC} above threshold)	error_table_src (1) = 1	error_table_src (3) = 0

Normal mode in this application:
- High-side is ON and the V_{SRC} comparator should be high
- High-side is OFF and the V_{SRC} comparator should be low

HS2 (VBOOST) error table

Table 42. Hsx_diag_config_2 Registers (1DCh)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved								error_table_src				error_table_vds			
Reset	00000000								0100				1001			

	output_command = 0 (Pre-driver switched OFF)	**output_command = 1 (Pre-driver switched ON)**
hsx_vds_fbk = 0 (V_{DS} below threshold)	error_table_vds (0) = 1	error_table_vds (2) = 0
hsx_vds_fbk= 1 (V_{DS} above threshold)	error_table_vds (1) = 0	error_table_vds (3) = 1

	output_command = 0 (Pre-driver switched OFF)	**output_command = 1 (Pre-driver switched ON)**
hsx_src_fbk = 0 (V_{SRC} below threshold)	error_table_src (0) = 0	error_table_src (2) = 1
hsx_src_fbk = 1 (V_{SRC} above threshold)	error_table_src (1) = 0	error_table_src (3) = 0

V_{BOOST} high-side source detection is different in this application from V_{BAT} since S_HSVbat and S_HS_VBoost are shorted together through a diode. If V_{BAT} high-side is ON, voltage on the V_{BOOST} source high-side is equal to V_{PWR} - diode. Consequently, the PT2000 should not detect an error on high-side V_{BOOST} if command = 0 and source feedback = 1.

7 Application Source Code

The following microcode can be directly downloaded from http://www.nxp.com/files/analog/doc/app_note/AN5186SW.zip (see KITPT2000FRDM6C). Using the IDE and SPIGEN, the microcode can be downloaded to the PT2000.

7.1 Injection Banks Management Source Code

```
* This microcore will control BANK1 will check all diagnostics and detect the EOI using PT2000

#define HSBatB1   hs1;
#define HSBoostB1 hs2;
#define LS1B1     ls1;
#define LS2B1     ls2;

#define HSBatB2   hs3;
#define HSBoostB2 hs4;
#define LS1B2     ls3;
#define LS2B2     ls4;

*cur1 for ucore0
*cur2 for uCore1

#include "dram1.def";


*################### STATUS REGISTER ##############
* This bit must be set to 1 if the Iboost current is never reached during the boost phase
#define BoostErrorBit0 b5;
* This bit must be set to 1 the sequencer is currently executing the Automatic interrupt routine
#define AutoIrqBit0 b6;
* This bit must be set to 1 the sequencer is currently executing the Idle Diag interrupt routine
#define IdleIrqBit0 b7;
* This bit must be set to 1 if start pin stays high longer than 10ms
#define HoldErrorBit0 b8;
* The followings are the status bits of the 2 injectors
* Bit = 1 => last actuation not completed correctly
* Bit = 0 => no problems during last actuation
#define Inj1FaultBit b0;  *uc0
#define Inj2FaultBit b1;  *uc0
#define Inj3FaultBit b0;  *uc1
#define Inj4FaultBit b1;  *uc1




*################### FLAGS ##############
* This flag is sent to the DCDC sequencer. It must be active for the whole period the boost voltage is used
* When the boost voltage is used, the DCDC must be deactivated
* flag = 0 => boost voltage is used, DCDC must be deactivated
* flag = 1 => boost voltage not used, DCDC can be active
#define BstFlag b0;


*################### CONTROL REGISTER ##############
* During the interrupt routine, the sequencer wait for this bit to be set to '1' before resuming execution of
application code
#define AutoDiagResetBit0 b6;
* During the interrupt routine, the sequencer wait for this bit to be set to '1' before resuming execution of
application code
#define IdleDiagResetBit0 b7;
* During the interrupt routine, the sequencer wait for this bit to be set to '1' before resuming execution of
application code
#define BoostResetBit0 b5;
* During the interrupt routine, the sequencer wait for this bit to be set to '1' before resuming execution of
application code
#define HoldResetBit0 b8;


*################### ALU registers  ################
#define IRQ_stat_Reg0 r0;


*****************************************************************************************************
*                                       AUTOMATIC and SOFTWARE INTERRUPT                            *
*****************************************************************************************************
irq_detected:   stos off off off;                       * Disable drivers
                endiaga diagoff;                        * Disable automatic diagnostic
                stirq low;                          * Set low the IRQB pin
                stf high BstFlag;                   * Set flag0 high to release the DC-DC converter idle mode


*######################## Check which injector was interrupted          #################################

check_inj:    joidr check_inj1 seq1;                * Check which ucore(sequencer) detected the error

check_inj0:   joslr MarkInj1 start1;               * the last active injector was number 1
              joslr MarkInj2 start2;               * the last active injector was number 2
              jmpr check_irq;                      * jump to restore

check_inj1:   joslr MarkInj3 start3;               * the last active injector was number 3
              joslr MarkInj4 start4;               * the last active injector was number 4
              jmpr check_irq;                      * jump to restore
*######################## Report fault to status register for the injector interrupted #########################
MarkInj1:     stsrb high Inj1FaultBit;             * mark the injector 1 as fault
              jmpr check_irq;
MarkInj2:     stsrb high Inj2FaultBit;             * mark the injector 2 as fault
              jmpr check_irq;
MarkInj3:     stsrb high Inj3FaultBit;             * mark the injector 3 as fault
              jmpr check_irq;
MarkInj4:     stsrb high Inj4FaultBit;             * mark the injector 4 as fault
              jmpr check_irq;

*######### Check which interrupt occurred BoostErr 0 or Idle Diag Fail 1 or Automatic Diagnostics #################
check_irq:      cp irq IRQ_stat_Reg0;                   * copy the irq status registers to a temp ALU reg
                                                       * This register contains also the sw irq ID

*######### Check if its auto or sw interrupt        ##############################################
               ldirh 10h rst;                        * load MSB in ir reg: 0x1000 in immediate register, to use
as mask for irq status
               and IRQ_stat_Reg0;                    * extract the irq source from irq status register (bits 12)
               jarr auto_waitEnable0 all0;            * if the irq source is 0 => Automatic Diagnostics detected
                                                       * Else => a software interrupt occured go to next line

*######### Check which sw interrupt occured       ##############################################
                cp irq IRQ_stat_Reg0;                   * copy the irq status registers to a temp ALU reg
               ldirh 0Ch rst;                        * load MSB in ir reg: 0x0C00 in immediate register, to use
as mask for irq status
                and IRQ_stat_Reg0;                     * extract the sw id from irq status register (bits 11-10)
             jarr Boost_waitEN0 all0;            * if the sw id is 0 => Iboost never reached => go to Boost_waitEN
                                                       * Else => error detected in idle diag=> go to next line
=> seq stuck until micro write 1 in control register b8


               ldirh 08h rst;                        * load MSB in ir reg: 0x0800 in immediate register, to use
as mask for irq status
                and IRQ_stat_Reg0;                     * extract the sw id from irq status register (bits 11-10)
             jarr Idle_waitEnable0 all0;          * if the sw id is 0 => Means IRQ = b01 => Idle diagnostics fails
                         * Else => Hold error => go to next line => idle diag fail


Hold_waitEN0:   stsrb high HoldErrorBit0;                * Start pin stays higher longer than 10ms
              jcrr Hold_waitEN0 HoldResetBit0 low;      * Wait here until control bit register b4 is write to 1
                jmpr restore0;

Idle_waitEnable0: stsrb high IdleIrqBit0;                 * IDle diag fail we set status b7 high to let user know
which error occured
              jcrr Idle_waitEnable0 IdleDiagResetBit0 low;* Wait here until control bit register b7 is write to 1
                jmpr restore0;

auto_waitEnable0: stsrb high AutoIrqBit0;                  * Set status register bit 6 when automatic diagnosis
interrupt trig
                jcrr auto_waitEnable0 AutoDiagResetBit0 low;  * the sequencer is stuck here until the bit of the
control register is set to '1' b6
                jmpr restore0;                            * Jump to restore

Boost_waitEN0:  stsrb high BoostErrorBit0;               * Iboost never reached, let user know by setting status
register bit b5
                jcrr Boost_waitEN0 BoostResetBit0 low;     * Wait here until control bit register b5 is write to 1



restore0:       stirq high;                         * Set high IRQB pin
                rstreg all;                         * Reset a) control registers
                                                    *       b) status regsiter
                                                 *       c) err_seq register (status of automatic diagnosis
                                                *       d) re-enables irq generation from automatic diagnosis
                iret restart rst;                   * Clear interrupt queue and restart from init phase

*******************************************************************************************
*                                          INIT PHASE
```

```
*******************************************************************************

* ### Initialization phase ###
init0:      stgn gain8.68 sssc;                    * Set the gain of the opamp of the current measure block 1
           ldjr1 eoinj0;                      * Load the eoinj line label Code RAM address into the register jr1
            ldjr2 idle0;                        * Load the idle line label Code RAM address into the register jr2
            stirq high;
            cwef jr1 _start row1;                  * If the start signal goes low, go to eoinj phase

*******************************************************************************
*                                    IDLE PHASE                                *
*******************************************************************************

* ### Idle phase- the uPC loops here until start signal is present ###
idle0:      joslr inj1_start start1;               * Perform an actuation on inj1 if start 1 (only) is active
            joslr inj2_start start2;               * Perform an actuation on inj1 if start 2 (only) is active
offcompON0: stoc on sssc;                           * Enable Offset Compensation
            jmpr idle0;                            * jump back to idle0

*******************************************************************************
*                                SHORTCUT DEFINITION                           *
*******************************************************************************

* ### Shortcuts definition per the injector to be actuated ###
inj1_start: dfsct HSBatB1 LS1B1 HSBoostB1;          * Set the 3 shortcuts: VBAT, VBOOST, LS2
            dfcsct dac1;                            * use current feedback1
            stab 0;                                * Set the base address to 0 for Injector 1
            jmpr idle_diag0;                        * Jump to launch phase
* ### Shortcuts definition per the injector to be actuated ###
inj2_start: dfsct HSBatB1 LS2B1 HSBoostB1;          * Set the 3 shortcuts: VBAT, VBOOST, LS2
            dfcsct dac1;                            * use current feedback1
            stab 10;                               * Set the base address to 10 for Injector 2
            jmpr idle_diag0;                        * Jump to launch phase

*******************************************************************************
*                           PRE-ACTUATION DIAG PHASE                           *
*******************************************************************************
* All Vds should be high due to bias
* All HSs Vsrc should be high as well
idle_diag0:     stoc off sssc;                      * Disable Offset Compensation
               bias all on;                   * Enable all biasing structures, kept ON even during actuation
                 jocr idle_diag_fail0 _sc1v;        * Error detected if Vds of shortcut1 (HS) is low
                 jocr idle_diag_fail0 _sc2v;        * Error detected if Vds of shortcut2 (LS) is low
                 jocr idle_diag_fail0 _sc3v;        * Error detected if Vds of shortcut3 (Boost) is low
                 jocr idle_diag_fail0 _sc1s;        * Error detected if Vsrc of shortcut1 (HS) is low
                 jocr idle_diag_fail0 _sc3s;      * Error detected if Vsrc of shortcut3 (Boost) is low
               jmpr boostT0;                   * Jump to actuation phase if no failure detected in idle phase

idle_diag_fail0:reqi 1;                          * Go to software subroutine is fault detected in idle phase
HSBat error

*******************************************************************************
*                                     BOOST PHASE                              *
*******************************************************************************

* ### Launch phase enable boost ###
boostT0:   load Iboost dac_sssc _ofs;               * Load the boost phase current threshold in the current DAC
            ldcd rst _ofs keep keep injMaxTBoost c3;     * Start Boost Counter in case Iboost never reached
           ldcd rst _ofs keep keep injMinTBoost c2;    * Start Boost Counter in case Iboost is reached too fast
            stf low BstFlag;                             * Set flag0 high to stop the DC-DC converter
            stos on on on;                               * Turn VBAT off, BOOST on, LS on
            endiags on on on on;                         * Enable auto diag

            cwer boostT1 tc2 row5;                       * After injMinBoost go to Boost T2
           cwer boost_err0 ocur row2;                   * Jump to error in case current is reached before the
end of injMinBoost
            wait row125;

boostT1:    cwer peak0 ocur row2;                        * Jump to peak phase when current is over threshold
           cwer boost_err0 tc3 row5;                  * Define Wait Table if actuation longer than injMaxGuard
go to eoinj
            wait row125;


boost_err0: stf high BstFlag;                            * set flag0 high to release DCDC regulation
           reqi 0;                                     * Go to software subroutine is fault detected in Boost
phase, did not reach Iboost on time

*******************************************************************************
*                                     PEAK PHASE                               *
*******************************************************************************
* ### Peak phase continue on Vbat ###
peak0:     store cnt3 OpenInj_record ofs;         * Store opening time in the data in DRAM with offset to separate
```

```
inj1 and 2
            stf high BstFlag;                        * set flag0 high to release the DC-DC converter idle mode
            ldcd rst _ofs keep keep Tpeak_tot c1;    * Load the length of the total peak phase in counter 1
            load Ipeak dac_sssc _ofs;                * Load the peak current threshold in the current DAC

            cwer bypass0 tc1 row2;                   * Jump to bypass phase when tc1 reaches end of count
            cwer peak_on0 tc2 row3;                  * Jump to peak_on when tc2 reaches end of count
            cwer peak_off0 ocur row4;                * Jump to peak_off when current is over threshold

peak_off0:  ldcd rst _ofs keep keep Tpeak_off c2;    * Load in the counter 2 the length of the peak_off phase
            stos off on off;                         * Turn VBAT off, BOOST off, LS on
            wait row123;

peak_on0:   stos on on off;                          * Turn VBAT on, BOOST off, LS on
            wait row124;                             * Wait for one of the previously defined conditions

*****************************************************************************************************
*                                               BYPASS PHASE                                        *
*****************************************************************************************************

* ### Bypass phase ###
bypass0:    ldcd rst _ofs keep keep Tbypass c3;       * Load in the counter 3 the length of the off_phase phase
            stos off off off;                        * Turn VBAT off, BOOST off, LS off
            cwer hold0 tc3 row4;                     * Jump to hold when tc3 reaches end of count
            wait row14;                              * Wait for one of the previously defined conditions

*****************************************************************************************************
*                                               HOLD PHASE                                          *
*****************************************************************************************************

* ### Hold phase on Vbat ###
hold0:      ldcd rst _ofs keep keep Thold_tot c1;    * Load the length of the total hold phase in counter 2
            load Ihold dac_sssc _ofs;                * Load the hold current threshold in the DAC
            cwer hold_error0 tc1 row2;                    * Jump to eoinj phase when tc1 reaches end of count
            cwer hold_on0 tc2 row3;                  * Jump to hold_on when tc2 reaches end of count
            cwer hold_off0 ocur row4;                * Jump to hold_off when current is over threshold

hold_off0:  ldcd rst _ofs keep keep Thold_off c2;    * Load the length of the hold_off phase in counter 2
            stos off on off;                         * Turn VBAT off, BOOST off, LS on
            wait row123;

hold_on0:   stos on on off;                          * Turn VBAT on, BOOST off, LS on
            wait row124;                             * Wait for one of the previously defined conditions

hold_error0: reqi 2;                                 * If Start high is longer than Thold_tot go to sw interrupt



*****************************************************************************************************
*                                    END OF INJECTION PHASE                                         *
*****************************************************************************************************

* ### End of injection phase ###

eoinj0:     stos off off off;                        * Turn VBAT off, BOOST off, LS off
            endiaga diagoff;                         * Disable auto diag
           bias all off;                           * Enable all biasing structures, kept ON even during actuation
            stf high BstFlag;                        * set flag0 to high to release the DC-DC converter idle mode
            jmpf jr2;                                * Go back to idle mode

* ### End of Channel 1 - uCore0 code ###




*############################################ UCORE1
############################################################


* ### Initialization phase ###
init1:      stgn gain8.68 sssc;                      * Set the gain of the opamp of the current measure block 1
          ldjr1 eoinj1;                         * Load the eoinj line label Code RAM address into the register jr1
           ldjr2 idle1;                            * Load the idle line label Code RAM address into the register jr2
            cwef jr1 _start row1;                    * If the start signal goes low, go to eoinj phase

*****************************************************************************************************
*                                         IDLE PHASE                                                *
*****************************************************************************************************

* ### Idle phase- the uPC loops here until start signal is present ###
```

```
idle1:      joslr inj3_start start3;                * Perform an actuation on inj1 if start 1 (only) is active
            joslr inj4_start start4;                * Perform an actuation on inj1 if start 2 (only) is active
offcompON1: stoc on sssc;                               * Enable Offset Compensation
            jmpr idle1;                                 * jump back to idle0
*****************************************************************************************************
*                                        SHORTCUT DEFINITION                                        *
*****************************************************************************************************

* ### Shortcuts definition per the injector to be actuated ###
inj3_start: dfsct HSBatB2 LS1B2 HSBoostB2;          * Set the 3 shortcuts: VBAT, VBOOST, LS2
            dfcsct dac2;                                * use current feedback1
            stab 20;                                    * Set the base address to 0 for Injector 1
            jmpr idle_diag1;                            * Jump to launch phase
* ### Shortcuts definition per the injector to be actuated ###
inj4_start: dfsct HSBatB2 LS2B2 HSBoostB2;          * Set the 3 shortcuts: VBAT, VBOOST, LS2
            dfcsct dac2;                                * use current feedback1
            stab 30;                                * Set the base address to 10 for Injector 2
            jmpr idle_diag1;                            * Jump to launch phase

*****************************************************************************************************
*                                   PRE-ACTUATION DIAG PHASE                                        *
*****************************************************************************************************
* All Vds should be high due to bias
* All HSs Vsrc should be high as well
idle_diag1:     stoc off sssc;                          * Disable Offset Compensation
              bias all on;                         * Enable all biasing structures, kept ON even during actuation
                jocr idle_diag_fail1 _sc1v;             * Error detected if Vds of shortcut1 (HS) is low
                jocr idle_diag_fail1 _sc2v;             * Error detected if Vds of shortcut2 (LS) is low
                jocr idle_diag_fail1 _sc3v;             * Error detected if Vds of shortcut3 (Boost) is low
                jocr idle_diag_fail1 _sc1s;             * Error detected if Vsrc of shortcut1 (HS) is low
                jocr idle_diag_fail1 _sc3s;           * Error detected if Vsrc of shortcut3 (Boost) is low
              jmpr boost1T0;                        * Jump to actuation phase if no failure detected in idle phase

idle_diag_fail1:reqi 1;                               * Go to software subroutine is fault detected in idle phase
HSBat error


*****************************************************************************************************
*                                                BOOST PHASE                                        *
*****************************************************************************************************

* ### Launch phase enable boost ###
boost1T0:   load Iboost dac_sssc _ofs;                 * Load the boost phase current threshold in the current DAC
            ldcd rst _ofs keep keep injMaxTBoost c3;       * Start Boost Counter in case Iboost never reached
           ldcd rst _ofs keep keep injMinTBoost c2;      * Start Boost Counter in case Iboost is reached too fast
            stf low BstFlag;                               * Set flag0 high to stop the DC-DC converter
            stos on on on;                                 * Turn VBAT off, BOOST on, LS on
            endiags on on on on;                           * Enable auto diag

            cwer boost1T1 tc2 row5;                        * After injMinBoost go to Boost T2
            cwer boost_err1 ocur row2;                    * Jump to error in case current is reached before the
end of injMinBoost
            wait row125;

boost1T1:   cwer peak1 ocur row2;                          * Jump to peak phase when current is over threshold
           cwer boost_err1 tc3 row5;                    * Define Wait Table if actuation longer than injMaxGuard
go to eoinj
            wait row125;


boost_err1: stf high BstFlag;                              * set flag0 high to release DCDC regulation
            reqi 0;                                     * Go to software subroutine is fault detected in Boost
phase, did not reach Iboost on time

*****************************************************************************************************
*                                                 PEAK PHASE                                        *
*****************************************************************************************************

* ### Peak phase continue on Vbat ###
peak1:     store cnt3 OpenInj_record ofs;          * Store opening time in the data in DRAM with offset to separate
inj1 and 2
            ldcd rst _ofs keep keep Tpeak_tot c1;   * Load the length of the total peak phase in counter 1
            load Ipeak dac_sssc _ofs;               * Load the peak current threshold in the current DAC

            cwer bypass1 tc1 row2;                  * Jump to bypass phase when tc1 reaches end of count
            cwer peak_on1 tc2 row3;                 * Jump to peak_on when tc2 reaches end of count
            cwer peak_off1 ocur row4;               * Jump to peak_off when current is over threshold

            stf high BstFlag;                       * set flag0 high to release the DC-DC converter idle mode

peak_off1:  ldcd rst _ofs keep keep Tpeak_off c2;   * Load in the counter 2 the length of the peak_off phase
            stos off on off;                        * Turn VBAT off, BOOST off, LS on
            wait row123;
```

```
peak_on1:   stos on on off;                     * Turn VBAT on, BOOST off, LS on
            wait row124;                        * Wait for one of the previously defined conditions

*****************************************************************************************************************
*                                               BYPASS PHASE                                                    *
*****************************************************************************************************************

* ### Bypass phase ###
bypass1:    ldcd rst _ofs keep keep Tbypass c3;  * Load in the counter 3 the length of the off_phase phase
            stos off off off;                   * Turn VBAT off, BOOST off, LS off
            cwer hold1 tc3 row4;                * Jump to hold when tc3 reaches end of count
            wait row14;                         * Wait for one of the previously defined conditions

*****************************************************************************************************************
*                                               HOLD PHASE                                                      *
*****************************************************************************************************************

* ### Hold phase on Vbat ###
hold1:      ldcd rst _ofs keep keep Thold_tot c1;   * Load the length of the total hold phase in counter 2
            load Ihold dac_sssc _ofs;           * Load the hold current threshold in the DAC
            cwer hold_error1 tc1 row2;               * Jump to eoinj phase when tc1 reaches end of count
            cwer hold_on1 tc2 row3;             * Jump to hold_on when tc2 reaches end of count
            cwer hold_off1 ocur row4;           * Jump to hold_off when current is over threshold

hold_off1:  ldcd rst _ofs keep keep Thold_off c2;   * Load the length of the hold_off phase in counter 2
            stos off on off;                    * Turn VBAT off, BOOST off, LS on
            wait row123;

hold_on1:   stos on on off;                     * Turn VBAT on, BOOST off, LS on
            wait row124;                        * Wait for one of the previously defined conditions

hold_error1: reqi 2;                            * If Start high is longer than Thold_tot go to sw interrupt

*****************************************************************************************************************
*                                    END OF INJECTION PHASE                                                     *
*****************************************************************************************************************

* ### End of injection phase ###

eoinj1:     stos off off off;                   * Turn VBAT off, BOOST off, LS off
            endiaga diagoff;                    * Disable auto diag
           bias all off;                      * Enable all biasing structures, kept ON even during actuation
            stf high BstFlag;                   * set flag0 to high to release the DC-DC converter idle mode
            jmpf jr2;                           * Go back to idle mode

* ### End of Channel 1 - uCore1 code ###
```

8 References

Following are URLs where you can obtain information on NXP products and application solutions:

Document Number and Description		URL
PT2000	Data Sheet	www.nxp.com/files/analog/doc/data_sheet/MC33PT2000.pdf
KITPT2000FRDM6C	User Guide	www.nxp.com/files/analog/doc/user_guide/KTPT2000FRDM6CUG.pdf
PT2000SWUG	Programming Guide	www.nxp.com/files/analog/doc/user_guide/PT2000SWUG.pdf
PT2000IDEUG	Developer Studio	www.nxp.com/files/analog/doc/user_guide/PT2000-IDEUG.pdf
NXP.com Support Pages		**URL**
NXP.com		www.nxp.com
PT2000 Product Summary Page		www.nxp.com/PT2000
Analog Home Page		www.nxp.com/analog

9 Revision History

Revision	Date	Description
1.0	9/2015	• Initial release
2.0	2/2016	• Reduced duration of Auto Diagnostics wave in Figure 2 • Updated Filter Time values in Section 6.3.1.1, Filter time, page 27 • Updated Filter Time values in Section 6.3.2.1, Filter Time, page 29

RoHS

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number: AN5186
Rev. 2.0
2/2016