Mercury/ISL55169 Quick Start Application Note

Rev A05: 05/20/2013

This document contains information on a product under development. The parametric information contains target parameters that are subject to change.

Document Revision History

Revision	Date	Description
A01	7/05/04	Initial Draft
A02	3/24/05	Added PLL Register configuration to Ring Oscillator; section 1.5
A03	11/10/05	Added PMU FI Example
A04	11/14/05	Changed FI example to say Loop=1 (instead of Tight=1)
A05	In Progress	Fixed Die Address = 0xC07F, section 1.2.1 Added Va/Vb expected voltages, section 1.6.1 Updated to Elevate Semiconductor Format.

Table of Contents

1 Introduction

This document describes the steps to perform an initial system check-out of the Mercury device.

These instructions assume the customer system can set the Mercury supplies, provide a PLL Clock, has a mechanism to read/write registers, has the ability to measure voltages (either using an external DMM or system resource), provide a simple DATA stream, and so on.

Important Note: The steps described below illustrate a cause and effect to demonstrate how to interface with the device. In a normal application; sequencing, calibration and other factors may require the registers to be written in a different order. Please refer to the Planet ATE Software Driver documentation for details.

1.1 Set Mercury Supplies

The first step is to apply the appropriate voltage. After power is applied, it is recommended to toggle the hardware reset (active high) on Mercury device.

Verify the voltage at the Mercury pins or within close proximity to ensure there are no IR drops. In addition, the customer should verify the approximate current flow for each supply.

Note: In most systems, it may not be possible to measure the supply currents.

Table 1: Power-ON Current Values

Supply	Voltage (V)	Approximate Current (mA)	
		PLL_CK = 50MHz	PLL_CK = 100MHz
VCCH	13.0	50	50
VCC	8.0	35	35
VEE	-3.0	-100	-100
VDD	3.3	400	425
VREF	3.0	0	0
VOH	3.0	0	0
VOL	0.0	0	0

Notes:

1) The VTT and VBB must also be properly set based on the application's digital levels. In many systems, the digital signals into the Mercury device will be 0.0V to 3.3V with a 50 Ohm output impedance. For this application, setting VTT = 1.65V and VBB = 1.65V along with setting the Mercury input termination to 50 Ohm will suffice.
2) The VOH and VOL should have minimal current since the COMP pins are in a static state and typically drive a high impedance input.

1.1.1 Measure REXT and CAP_PLL Voltages

The next step is to measure voltage of the REXT and CAP_PLL pins.

REXT voltage should track the VREF voltage within 25mV.

The CAP_PLL voltage is a function of the PLL_CK. In addition, the CAP_PLL voltage is dependant on the Vswing and Fclamp registers. Table 2 lists the CAP_PLL vs PLL_CK assuming the power-ON default Vswing (00b) and Fclamp (00b) settings.

Table 2: Default CAP_PLL Voltages

PLL_CK (MHz)	CAP_PLL (V)
50.0	1.0
75.0	1.3
100.0	1.6

1.2 Register/RAM Access (CPU Port Transactions)

The following steps perform simple register/RAM access (write/read) to verify to CPU port is functioning.

1.2.1 Read Die ID Register

Read the Die ID registers (address = 0xC07F); Table 3 lists the expected return values for different silicon revisions.

Table 3: Die ID Register Values

Rev	Read-back Data (HEX)
3	0x0323
4	0x0324
#	0x032#

1.2.2 Register Access

Follow Table 4 to verify the basic ability to write/read registers. The following bullet items highlight some key aspects of a register access:

- Address bit D15 (Register Bit) must be '1' to access a register
- Address bit D14 (Central Bit) is used to distinguish between Per-Chan and Central Registers
- WE bit must be '1' for the corresponding Data bits to be written
- WE bits (typically) read back '0'
- Writing to Undefined Data bits has no effect
- Undefined Data bits (typically) read back '0'
- A RESET sets all registers to a default value of '0000' (except Read-Only registers)
- Refer to the CPU Protocol Timing diagram for details

Table 4: Write/Read Registers

Description	Address (HEX)	Write Data (HEX)	Read Data (HEX)
Ch#0: Set all Driver Term	0x8000	0xFFFF	0x001B
Ch#0: Clear EN Term. DATA Term should be unchanged.	0x8000	0x0020	0x0003
Ch#6: Set all DIN Switch Control	0x8341	0xFFFF	0x0555
Ch#6: Clear all DIN Switch Control	0x8341	0x0AAA	0x0000
Parallel Write: Set DVL Radj (need to set 'Parallel Load' bit)	0x8002	0x001F	0x000F (read on all chans)
Central Reg: Set IR = 2mA (0x20)	0xC002	0x0120	0x0020
Central Reg: Software RESET	0xC006	<don't care>	0x0000 (all registers)

1.2.3 RAM Access

Follow Table 4 to verify the basic ability to write/read RAM. The following bullet items highlight some key aspects of a RAM access:

- The Read-back Data matches the Written Data
- Address bit D15 (Register Bit) must be '0' to access a register
- Address bit D14 (Central Bit) is used to distinguish between Per-Chan and Central Registers
- A RESET does NOT set the RAM contents to a default value. They are left unchanged
- A RAM read-back requires 2 addition clock cycles
- Refer to the CPU Protocol Timing diagram for details

Table 5: Write/Read Registers

Description	Address (HEX)	Data (HEX)
Ch#0: DVH Value DAC	0x0000	0xFFFF
Ch#0: DVH Offset DAC	0x0020	0xAAAA
Ch#0: DVH Gain DAC	0x0040	0x5555
Ch#4: CVA Value DAC	0x0202	0x1234
Parallel Write: DVL Offset DAC (need to set 'Parallel Load' bit)	0x0021	0xABCD (read on all chans)
Central Reg: VFV Value DAC	0x4000	0x8765
Central Reg: IClamp Low Gain DAC	0x4047	0x1FFF

1.3 Reset State

Whenever a Hardware or Software reset is issued, the device is configured into the following state.

Note: The rest of the examples rely on the default (reset) configuration to demonstrate the minimum register transactions to achieve the desired configuration.

- RAM – Unknown at power ON. Left unchanged if reset issued from a valid configuration
- Registers – all registers are initialized to '0'; which implies:
  - Driver Block
    - Driver is in high-Z mode
    - Data/En sourced from CPU control
    - Deskews disabled
    - All switches open
    - Data/En Input Termination none
    - Voltage Range 0
  - Comparator Block
    - All switches open
    - COMP pins sourced from real-time comparator.
      - Since switches are open, this node is floating internally
    - Deskews disabled
    - Voltage Range 0
  - PMU Block
    - PMU Disabled
    - All switches open
    - Tight loop
    - All IR open
    - Voltage Range 0
    - V-Clamps and I-Clamps are disabled
    - Monitor is in high-Z
  - PLL
    - No input termination
    - Vswing/Fclamp set to '0'.

1.4 Set Driver (DVH/DVL) Level

The following steps are used to output a Driver voltage at DOUT_#. The CPU-Data/En register control bits are used to set the driver state: Set CPU-Data high for DVH and low for DVL.

In addition to measuring the voltage at the DOUT_# pin, the customer should (could) also confirm the voltage at the DVH_# and DVL_# pin (driver compensation capacitors).

Note: The 'Output Voltage' is the approximate voltage since these are un-calibrated settings.

Table 6: Set Driver DVH/DVL Levels

Description	Address (HEX)	Data (HEX)	Output Voltage
Software Reset	0xC006	<don't care>	High-Z
Ch#0: Set DVH = 3.0V (Driver defaults to VR0)	0x0000	0xDFFF	High-Z
Ch#0: Set DVH Offset = 0.0	0x0020	0x7FFF	High-Z
Ch#0: Set DVH Gain = 1.0	0x0040	0x7FFF	High-Z
Ch#0: Set DVL = 0.0V	0x0001	0x1FFF	High-Z
Ch#0: Set DVL Offset = 0.0	0x0021	0x7FFF	High-Z
Ch#0: Set DVL Gain = 1.0	0x0041	0x7FFF	High-Z
Ch#0: Enable Driver (CPU-En = 1) and set Driver High (Data = 1)	0x8001	0x02D0	3.0 (DVH)
Ch#0: Set DVH = 1.5V	0x0000	0x7FFF	1.5
Ch#0: Set DVH = 3.5V	0x0000	0xFFFF	3.5
Ch#0: Set Voltage Range 1 (VR1)	0x8006	0x0005	6.0 (RAM contents don't change)
Ch#0: Set Driver Low (Data = 0)	0x8001	0x0040	0.0 (DVL)
Ch#0: Set DVL = -1.0V Driver is already in VR1	0x0001	0x0000	-1.0
Ch#0: Set Voltage Range 2 (VR2)	0x8006	0x0006	-2.0 (RAM contents don't change)
Ch#0: Set DVL = 0.0V	0x0001	0x1FFF	0.0

Note: Repeat the above steps for other channels.

1.5 Ring Oscillator (Driver/Comparator Outputs)

The following steps use the internal Ring Oscillator to generate an approximate 30MHz pulse. The output waveforms can be monitored at the DOUT_# and COMPA/B_# pins. The Ring Oscillator is a useful debug feature since the customer doesn't need to provide a DATA_# stream to the device.

Note: The Ring Oscillator is used in production testing, demonstrations and for initial troubleshooting. This feature is typically not useful in a customer system application.

Table 7: Ring Oscillator Output Waveforms

Description	Address (HEX)	Data (HEX)	DOUT_# Output	COMPA/B_# Output
Software Reset	0xC006	<don't care>	High-Z	Unknown (internal node floats)
Set Fclamp/Vswing/PLL Term based on PLL_CK	0xC004	See Datasheet	High-Z	Unknown (internal node floats)
Ch#0: Set DVH = 3.0V (Driver defaults to VR0)	0x0000	0xDFFF	High-Z	Unknown (internal node floats)
Ch#0: Set DVH Offset = 0.0	0x0020	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Set DVH Gain = 1.0	0x0040	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Set DVL = 0.0V	0x0001	0x1FFF	High-Z	Unknown (internal node floats)
Ch#0: Set DVL Offset = 0.0	0x0021	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Set DVL Gain = 1.0	0x0041	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Set CVA = 1.5V (Comparator defaults to VR0)	0x0002	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Set CVA Offset = 0.0	0x0022	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Set CVA Gain = 1.0	0x0042	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Set CVB = 1.5V (Comparator defaults to VR0)	0x0003	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Set CVB Offset = 0.0	0x0023	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Set CVB Gain = 1.0	0x0043	0x7FFF	High-Z	Unknown (internal node floats)
Ch#0: Enable Driver (CPU-En=1) Set Driver = Real-Time Data	0x8001	0x02E0	Matches DATA_0 input	Unknown (internal node floats)
Ch#0: Connect DOUT to Comp	0x8041	0x0300	Matches DATA_0 input	Matches DOUT_0 state
Ch#0: Connect and Enable Ring Oscillator to DATA path	0x8045	0x0061	Most likely Low	Matches DOUT_0 state
Ch#0: Trigger Ring Oscillator	0x8045	0x0180	15MHz 3V swing	15MHz VOH/VOL swing

Note: Repeat the above steps for other channels.

1.5.1 Real-Time Data

To demonstrate a real-time data pattern using the DATA_# to DOUT_# path; the customer should skip the last 2 steps (don't enable the Ring Oscillator). The customer needs to provide a valid DATA_# input signal. The Mercury device is configured for CPU-EN = High (Driver enabled).

1.6 Monitor Output

The following steps use the internal Test & Cal diagnostic mux to output a known voltage at the MONITOR pin.

Note: The 'Output Voltage' is the approximate voltage since these are un-calibrated settings.

Table 8: Monitor Output

Description	Address (HEX)	Data (HEX)	MONITOR Output
Software Reset	0xC006	<don't care>	High-Z
Central: Set VFV = 3.5V (PMU defaults to VR0)	0x4000	0xFFFF	High-Z
Central: Set VFV Offset = 0.0V	0x4020	0x7FFF	High-Z
Central: Set VFV Gain = 1.0V	0x4040	0x7FFF	High-Z
Central: Set Sel-Mu-Diag to source from Test & Cal mux and Diag to source from VFV DAC	0xC003	0x00F0	High-Z
Central: Enable Monitor	0xC001	0x0C00	3.5

1.6.1 Measure Junction Temperature

The following steps show how to measure and calculate the junction temperature using the internal temperature sensors.

Tj = (Va – Vb) * 1637 – 221

The measured temperature will be dependant on if there is a heat sink and/or air-flow present.

Table 9: Measure Junction Temperature

Description	Address (HEX)	Data (HEX)	MONITOR Output
Software Reset	0xC006	<don't care>	High-Z
Central: Set Sel-Mu-Diag to source from Test & Cal mux and Diag to source from PMU-Va-Tj	0xC003	0x00FA	High-Z
Central: Enable Monitor	0xC001	0x0C00	Measure Va (expect ~2.2V)
Central: Set Test & Cal to PMU-Vb-Tj	0xC003	0x00FB	Measure Vb (expect ~2.0V)
Apply above formula			

1.7 PMU Force Voltage (FV) Mode

The following steps route the Central PMU to the Channel #0 DOUT.

Note: The 'Output Voltage' is the approximate voltage since these are un-calibrated settings.

Table 10: PMU Force Voltage (FV) Mode

Description	Address (HEX)	Data (HEX)	DOUT_# Output
Software Reset	0xC006	<don't care>	High-Z
Central: Set VFV = 3.0V (PMU defaults to VR0)	0x4000	0xDFFF	High-Z
Central: Set VFV Offset = 0.0	0x4020	0x7FFF	High-Z
Central: Set VFV Gain = 1.0	0x4040	0x7FFF	High-Z
Central: Set IR=2mA	0xC002	0x0120	High-Z
Central: Enable PMU	0xC000	0x3000	High-Z
Ch#0: Connect PMU to DOUT_0	0x8003	0x00A0	3.0V
Central: Set VFV = 1.5V	0x4000	0x7FFF	1.5V

1.8 PMU Force Current (FI) Mode

The following steps route the Central PMU to the Channel #0 DOUT. A load needs to be connected to the DOUT_0. If no load is present then the FI will cause the PMU to output a voltage at one of the supply rails. The PMU is connected to the DOUT_0 prior to switching into FI mode to prevent a large glitch.

Note: if the part is put into FI mode before connecting to the DOUT pin then the internal voltage will go to one of the supply rails, therefore when the PMU is connected it will cause a glitch on DOUT#.

Note: The 'Output Current' is the approximate current since these are un-calibrated settings.

Table 11: PMU Force Current (FI) Mode

Description	Address (HEX)	Data (HEX)	DOUT_# Output
Software Reset	0xC006	<don't care>	High-Z
Central: Set VFI = 1.0V (+Imax)	0x4001	0xFFFF	High-Z
Central: Set VFI Offset = 0.0	0x4021	0x7FFF	High-Z
Central: Set VFI Gain = 1.0	0x4041	0x7FFF	High-Z
Central: Set IR=2mA	0xC002	0x0120	High-Z
Ch#0: Connect PMU to DOUT_0	0x8003	0x00A0	0.0V (FV Mode)
Central: Enable PMU, set FI mode, and set Loop=1 (sense across Rsense instrument amp)	0xC000	0x3033	FI Mode (voltage depends on load)

1.9 Planet ATE Software Driver and Documentation

At this point, the customer has demonstrated the basic ability to configure the device, set some levels, make some measurements, and output a driver waveform. This should imply the Mercury device has been properly designed and assembled in the customer system.

The next step is to integrate the Planet ATE software driver (reference/example code) which allows the customer to:

- Configure the device for different modes
- Perform DC calibration
- Set Levels as a function of voltage rather than HEX codes
- Set Deskews
- Configure the PMU for different modes; including proper sequencing
- And more

