Simplifying System Integration™

73M1866B/73M1966B Implementer's Guide

March 26, 2010

Rev. 1.3

UG_1x66_016

Teridian Semiconductor Corp., 6440 Oak Canyon, Suite 100, Irvine, CA 92618
TEL (714) 508-8800, FAX (714) 508-8877, http://www.teridian.com

Table of Contents

Figures

1 Introduction

This guide describes how to use the 73M1866B and 73M1966B MicroDAA® FXO for Voice-over-IP (VoIP) applications. The guide provides application-specific detail that is not found in the *73M1866B/73M1966B Data Sheet*. The guide also includes suggested algorithms that can be followed by users who are developing their own software. In most cases the software implementations described are very similar, if not the same, as those used in the Teridian 73M1x66B Reference Driver.

The 73M1966B and 73M1866B will be collectively referred to as the 73M1x66B in this document.

1.1 Procedure Conventions

The following conventions apply to the procedures in this document:

- Firmware/software variables in the procedures are in italics (e.g. *VAL*) to distinguish them from register bits or fields.
- The register is shown with the address in the leftmost cell of the first row. The first row shows the bit mnemonic and is ordered with the most significant bit to the left and least significant bit to the right.

0x12	**OFH**	**ENDC**	**ENAC**	**ENSHL**	**$\overline{\text{ENLVD}}$**	**ENFEL**	**ENDT**	**ENNOM**
Write	0/1	0	1	X	0	1	0	X

- The second row indicates values written in the indicated bits during the procedure.
- Cells with multiple values indicate that the bit is used more than once during the procedure. For example, in the example above, OFH is first set to 0 and later set to 1.
- Register fields with an "X" in the procedures indicate bits that remain unchanged during write operations or bits that are not relevant during read operations.
- Registers referenced in the procedures are represented as RGnn where nn is the hexadecimal register address from 0x00 to 0x25 (for example, RG12 represents the register at address 0x12).

1.2 Read-Modify-Write Procedure

Register writes that include unchanged bits should use a read-modify-write procedure to preserve the setting of bits that do not need to be modified during a given SPI transaction. The read-modify-write procedure consists of the following steps:

1. Read the register value into a variable.
2. Apply a mask to the variable to retain bits that are not to be changed and clear bits that are to be changed.
3. Apply a mask to the variable to change the desired bits.
4. Write the new value back to the register.

The following example uses the read-modify-write procedure to write 01xx_110x to Register 0xnn:

1. *VAR* = RGnn
2. *VAR* = *VAR* AND 0011_0001
3. *VAR* = *VAR* OR 0100_1100
4. RGnn = *VAR*

2 Hardware Requirements

2.1 Reset

The reset pin of the 73M1x66B is active low. Following the power-up of the device and the reset pin being de-asserted, the 73M1x66B SPI interface is ready for communicating with the host.
Though not explicitly required, it is recommended that the PCM clock and FS be running and stable before reset is de-asserted on the 73M1x66B device.

2.2 SPI Interface

The SPI transactions are described in Section 5 of the *73M1866B/73M1966B Data Sheet*. The write and read transactions are illustrated in Figure 1 and Figure 2, respectively.

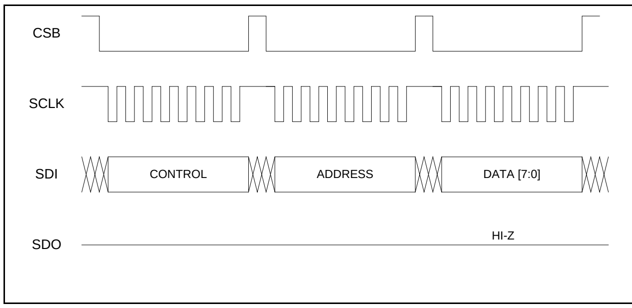


Figure 1: SPI Write Transaction

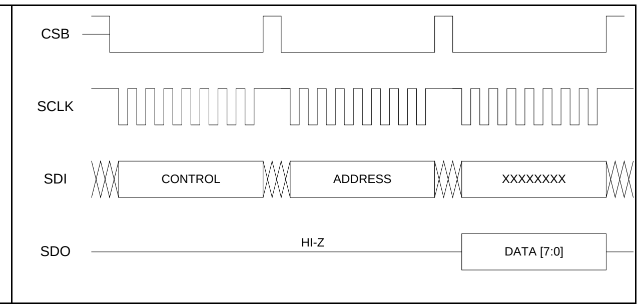


Figure 2: SPI Read Transaction

All SPI transactions that are targeting 73M1916 control registers (from address 0x12 to 0x19 and address 0x1F) must maintain a minimum inter-transaction gap of 500 µs. The inter-transaction gap starts from the de-assertion of CSB after the data byte of the first transaction and terminates at the assertion of CSB before the control byte of the second transaction.

2.3 PCM Interface

The PCM Highway Interface is described in Section 8 of the *73M1866B/73M1966B Data Sheet*. The PCM Highway Clock and Frame Sync signals must be stable and running at legal values for the 73M1x66B device to operate properly.

The settings that control the PCM Highway interface can be set via the SPI bus without a running PCLK and FS.

See Section 3.1.4 PCM Interface Configuration for more information on how to properly configure the PCM Highway interface.

2.4 Interrupts

The 73M1x66B devices provide a hardware interrupt pin (active low – open drain) that goes active upon detection of any of several programmable hardware events within the 73M1x66B. The interrupt functionality is described in Section 7.2 of the *73M1866B/73M1966B Data Sheet*.

The interrupt pin is active and configured for operation upon reset of the 73M1x66B. Because interrupts are enabled by default, the device will generate an interrupt as soon as reset is de-asserted (due to a barrier failure detect). The host application must be ready to service or safely ignore this interrupt before the de-assertion of reset. The recommended way to deal with the first interrupt after reset is to disable the interrupt generation until the system is ready to handle them (see Section 3.1.1 Reset and Disable Interrupts).

3 Device Configuration and Initialization

3.1 Host-Side Device (73M1906B) Configuration

The Host-side device configuration and initialization includes the following steps:

1. Reset and Disable Interrupts
2. PCLK Clock Recovery and PLL
3. Call Progress Monitor Reset
4. PCM Interface Configuration

3.1.1 Reset and Disable Interrupts

The recommended way to deal with the first interrupt after reset is to disable the interrupt generation until the system is ready to handle them.

Despite being in Register 0x05, ENAPOL is not an interrupt masking register and should not be disabled.

The registers used in this procedure are:

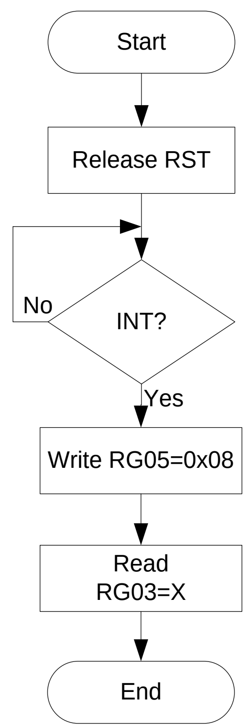
0x03	GPIO7	GPIO6	GPIO5	PCLKDT	RGMON	DET	SYNL	RGDT
Read	X	X	X	X	X	X	X	X

0x05	ENGPIO7	ENGPIO6	ENGPIO5	ENPCLKDT	ENAPOL	ENDET	ENSYNL	ENRGDT
Write	0	0	0	0	1	0	0	0

Begin

1. Release RST.
2. Wait for Interrupt.
3. Set ENGPIO[7:5] = ENPCLKD = ENDET = ENSYNL = ENRGDT = 0, set ENAPOL = 1 (RG05 = 0x08).
4. Read RG03 to clear the register value and de-assert the INT pin.

End



3.1.2 PCLK Clock Recovery and PLL Lock Detection

The 73M1x66B requires that the PLL be locked (to a stable PCLK and FS) to allow access to the line side device. The 73M1x66B does not require the PLL to be locked (nor a stable PCLK and FS) to access the host side device through the SPI interface. It is recommended that the PCLK and FS be stable before releasing the 73M1x66B from reset.

The registers used in this procedure are:

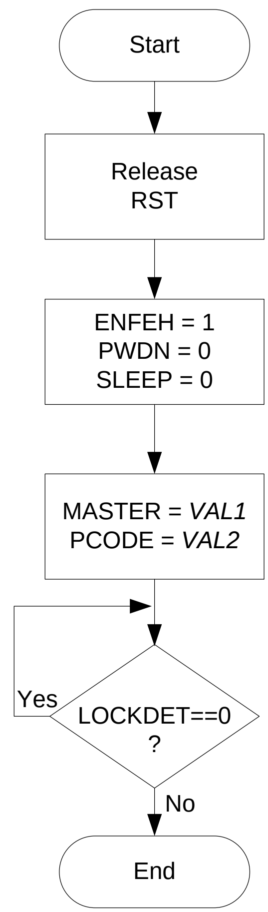
0x0F	ENFEH	PWDN	SLEEP	Res	Res	Res	Res	Res
Write	1	0	0/1	X	X	X	X	X

0x23	PCMEN	MASTER	PCODE[3:0]	LIN	LAW
Write	X	*VAL1*	*VAL2*	X	X

0x0D	LOKDET	SLHS	Res	Res	RSTLSBI	Res	Res	*Res*
Read	?	X	X	X	X	X	X	X

The temporary variables defined in this procedure are:

VAL1 = System appropriate value to write to the MASTER bit.
VAL2 = System appropriate value to write to the PCODE[3:0] bits.



Begin

1. Release RST.
2. Write ENFEH = 1, PWDN = 0, SLEEP = 0.
3. Write MASTER = *VAL1* and PCODE[3:0] = *VAL2*.
4. Read RG0D.
5. If LOKDET == 0 goto 5.

End

Slave Mode

If the 73M1x66B is configured for slave mode and PCODE is set to the default value of 0, if PCLK and FS are present and stable before reset is released the 73M1x66B will automatically lock to the appropriate PCLK frequency.

If this cannot be achieved then the user should reset the PLL after PCLK and FS have become stable before proceeding with further device initialization. This can be done by toggling the SLEEP bit in Register 0x0F[5]. The user can check that the PLL is locked and ready by polling the LOCKDET bit in Register 0x0D[7].

Master Mode

In Master Mode, if PCLKI is present and stable before reset is released the 73M1x66B will automatically lock to PCLKI and provide PCLK and FS as outputs to the PCM Highway.

3.1.3 Call Progress Monitor Reset

If used in 16 kHz mode, the call progress monitor (CPM) circuit must be re-initialized by cycling the SLEEP bit.

The registers used in this procedure are:

0x0F	**ENFEH**	**PWDN**	**SLEEP**	**Res**	**Res**	**Res**	**Res**	**Res**
Write	X	X	1/0	X	X	X	X	X

0x0D	**LOKDET**	**SLHS**	**Res**	**Res**	**RSTLSBI**	**Res**	**Res**	***Res***
Read	?	X	X	X	X	X	X	X

0x10	**Res**	**Res**	**Res**	**CMVSEL**	**CMTXG1**	**CMTXG0**	**CMRXG1**	**CMRXG0**
Write	0	0	0	*VAL1*	*VAL2*		*VAL3*	

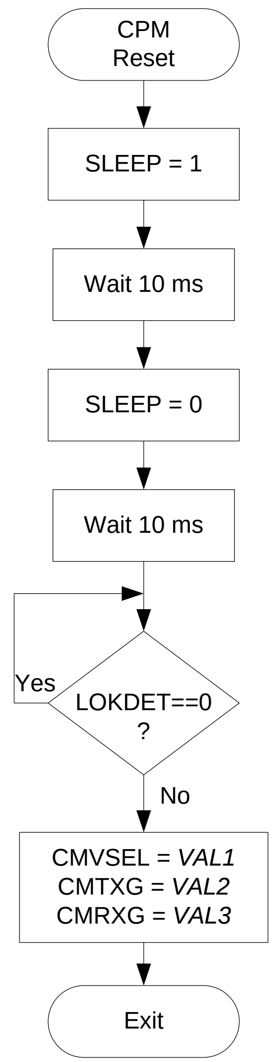
The temporary variables defined in this procedure are:

VAL1 = System appropriate value to write to the CMVSEL bit.
VAL2 = system appropriate value to write to the CMTXF[1:0] bits.
VAL3 = System appropriate value to write to the CMRXG[1:0] bits.

Begin

1. Write SLEEP = 1.
2. Wait 10 ms.
3. Write SLEEP = 0.
4. Wait 10 ms.
5. Read RG0D.
6. If LOKDET == 0 goto 5.
7. Write CMVSEL = *VAL1*, CMTXG[1:0] = *VAL2* and CMRXG[1:0] = *VAL3* to RG10.

End



3.1.4 PCM Interface Configuration

The PCM Highway Interface is described in Section 8 of the *73M1866B/73M1966B Data Sheet*. The PCM Highway Clock and Frame Sync signals must be stable and running at legal values for the 73M1x66B device to operate properly.

After the device has locked to PCLK and FS the user must configure the PCM interface for the specific system it resides on. The PCM configuration settings can change per call or remain static over the power cycle but before setting the PCMEN bit = 1, the user should make sure to set all other PCM register settings.

The settings that control the PCM Highway interface can be set via the SPI bus without a running PCLK and FS. The following registers control the PCM Highway interface.

0x20	**TPOL**	**TTS6**	**TTS5**	**TTS4**	**TTS3**	**TTS2**	**TTS1**	**TTS0**
Write	*VAL1*							

0x21	**RPOL**	**RTS6**	**RTS5**	**RTS4**	**RTS3**	**RTS2**	**RTS1**	**RTS0**
Write	*VAL2*							

0x22	**SR**	**ADJ**	**RCS2**	**RCS1**	**RCS0**	**TCS2**	**TCS1**	**TCS0**
Write	*VAL3*							

0x23	**PCMEN**	**MASTER**	**PCODE3**	**PCODE2**	**PCODE1**	**PCODE0**	**LIN**	**LAW**
Write	0/1	X	X	X	X	X	*VAL4*	

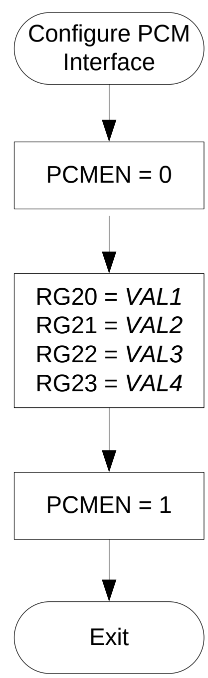
The temporary variables defined in this procedure are:

VAL1 = System appropriate value to write to the TPOL and TTS[6:0] bits.
VAL2 = system appropriate value to write to the RPOL and RTS[6:0] bits.
VAL3 = System appropriate value to write to the SR, ADJ, RCS[2:0] and TCS[2:0] bits.
VAL4 = System appropriate value to write to the LIN and LAW bits.

Begin

1. Write PCMEN = 0.
2. Configure other PCM interface registers: Write *VAL1* to RG20, *VAL2* to RG21, *VAL3* to RG22 and *VAL4* to RG23.
3. Write PCMEN = 1.

End



3.2 Line-Side Device (73M1916) Configuration

The Line-side device setup includes the following procedures:

1. Barrier Synchronization Recovery
2. Receiver DC Offset Calibration
3. Initial Line State Configuration

3.2.1 Barrier Synchronization Recovery

Before the Line-side device can be initialized, the barrier must be in sync and error free. The barrier is designed to power up the line-side device and come into sync automatically upon PLL lock. The user should check that the device indicates barrier sync before proceeding with line-side initialization.

The following registers control the Barrier Sync procedure.

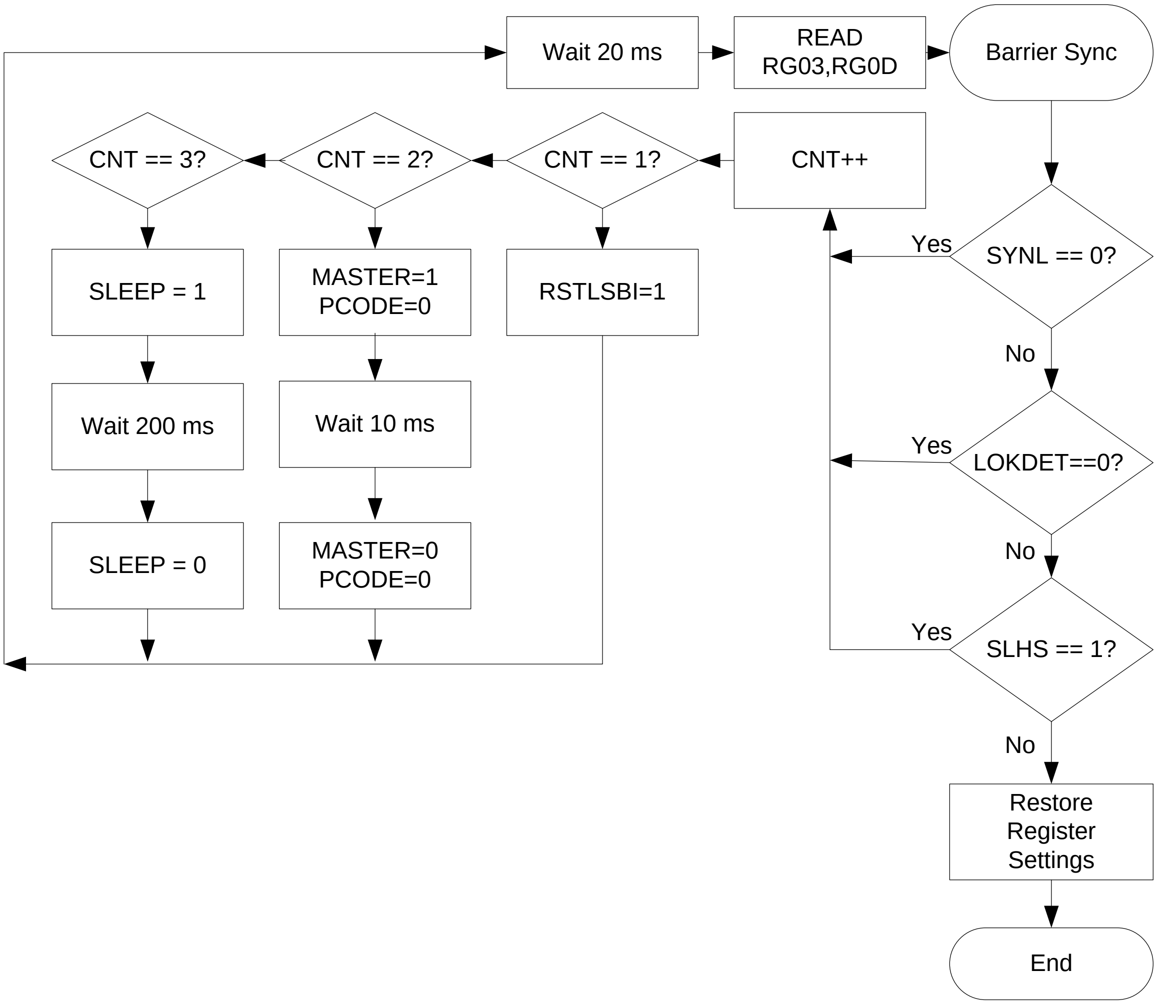
0x03	**GPIO7**	**GPIO6**	**GPIO5**	**PCLKDT**	**RGMON**	**DET**	**SYNL**	**RGDT**
Write	X	X	X	X	X	X	?	X

0x05	**ENGPIO7**	**ENGPIO6**	**ENGPIO5**	**ENPCLKDT**	**ENAPOL**	**ENDET**	**ENSYNL**	**ENRGDT**
Write	X	X	X	X	X	X	1	X

0x0D	**LOKDET**	**SLHS**	**Res**	**Res**	**RSTLSBI**	**Res**	**Res**	**Res**
Read	?	?	X	X	1/0	X	X	X

0x0F	**ENFEH**	**PWDN**	**SLEEP**	**Res**	**Res**	**Res**	**Res**	**Res**
Write	X	X	1/0	X	X	X	X	X

0x23	**PCMEN**	**MASTER**	**PCODE3**	**PCODE2**	**PCODE1**	**PCODE0**	**LIN**	**LAW**
Write	X	1/0	0	0	0	0	X	X



The temporary variables defined in this procedure are:

CNT1 = Resync Counter. Initial value = 0

Begin : BARRIER CHECK

1. Read RG03.
2. If SYNL == 0 goto RESYNC.
3. Read RG0D.
4. If LOKDET == 0 goto RESYNC.
5. If SLHS == 1 goto RESYNC.
6. Restore Previous Register Settings and go On Hook.

End

Begin : RESYNC

1. IF CNT1 !=1 goto 8
2. Write RSTLSBI = 1 (RG0D = xxxx_1xxx).
3. Wait 20 ms.
4. READ RGO3,RG0D
5. Goto BARRIER CHECK
6. IF CNT1 !=2 goto 15
7. Write PCODE=0 MASTER = 1 (RG023 = x100_00xx).
8. Wait 10 ms.
9. Write PCODE=0 MASTER = 0 (RG023 = x000_00xx).
10. Wait 20 ms.
11. READ RGO3,RG0D
12. Goto BARRIER CHECK.
13. IF CNT1 !=3 20
14. Write SLEEP = 1 (RG0F = xx1x_xxxx).
15. Wait 200 ms.
16. Write SLEEP = 0 (RG0F = xx0x_xxxx).
17. Wait 10 ms.
18. READ RGO3,RG0D.
19. Goto BARRIER CHECK
20. Resync Failed. Report to user and reset or power down entire device.

End

3.2.2 Receiver DC Offset Calibration

The effect of residual DC offset caused by the external components to the 73M1x66B can be reduced by a simple procedure in which the magnitude of this DC offset is measured and subtracted from each sample received from the PSTN. This calibration is intended to be performed once after power-on/reset and is done only after the PLL is locked and the barrier is synchronized. The calibration can be performed later if circumstances require intermittent calibration (for example if operating temperatures vary drastically). The procedure can only be performed when the device is On-Hook.

The registers used in this procedure are:

0x12	**OFH**	**ENDC**	**ENAC**	**ENSHL**	**$\overline{\text{ENLVD}}$**	**ENFEL**	**ENDT**	**ENNOM**
Write	0	0	1	X	0	1	0	X

0x15	**ENOLD**	**DISNTR**	**Res**	**CIDM**	**THEN**	**ENUVD**	**ENOVD**	**ENOID**
Write	X	X	X	X	1	0	X	X

0x16	**TXEN**	**RXEN**	**RLPEN**	**ATEN**	**ACZ3**	**ACZ2**	**ACZ1**	**ACZ0**
Write	0	X	X	0	X	X	X	X

0x17	**Res**	**Res**	**RXOCEN**	**Res**	**Res**	**Res**	**Res**	**Res**
Write	X	X	1/0	X	X	X	X	X

0x25	**RXOM7**	**RXOM6**	**RXOM5**	**RXOM4**	**RXOM3**	**RXOM2**	**RXOM1**	**RXOM0**
Read/ Write	*OFFSET*							

The temporary variables defined in this procedure are:

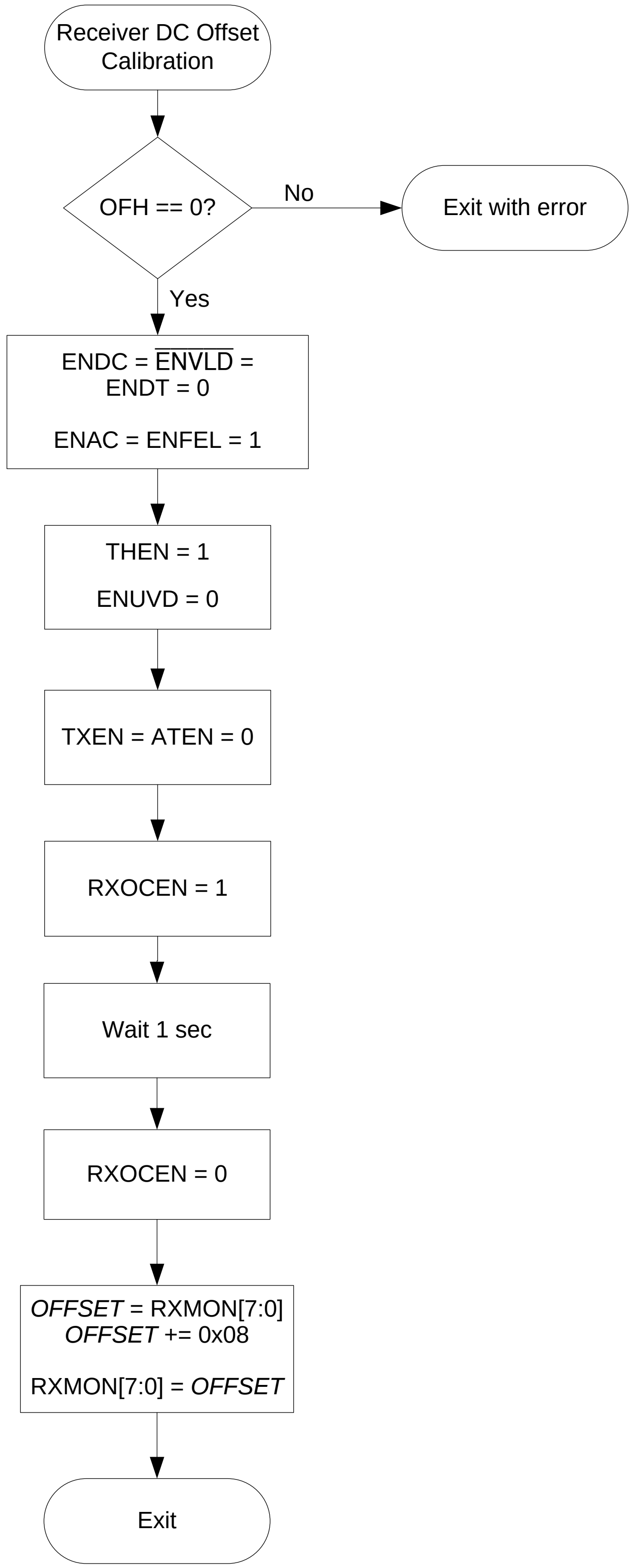
OFFSET = DC Offset value, RXOM[7:0].

Begin

1. Ensure that the device is On Hook (OFH==0). Executing the procedure when Off Hook will degrade performance. If the device is Off Hook, return an error code indicating that the device cannot perform the calibration routine.
2. Set ENDC, $\overline{\text{ENLVD}}$ and ENDT to 0; set ENAC and ENFEL to 1 (write RG12 = x01x_010x2).
3. Set THEN to 1 and ENUVD to 0 (write RG15 = xxxx_10xx).
4. Set TXEN and ATEN to 0 (write RG16 = 0xx0_xxxx).
5. Set RXOCEN to 1 (write RG17 = xx1x_xxx).
6. Allow 1 second for the receive offset measurement to stabilize.
7. Set RXOCEN to 0 (write RG17 = xx0x_xxxx).
8. *OFFSET* = RXOM[7:0] (Read RG25).
9. Add 08h to *OFFSET* and write this result back into RXOM[7:0] (RG25 = *OFFSET*).

End

0x14	**TXBST**	**DAA1**	**DAA0**	**Res**	**RXBST**	**RLPNH**	**RXG1**	**RXG0**
Write	X	X	X	0	X	X	X	X


Receiver DC Offset Calibration
OFH == 0?
No
Exit with error
Yes
ENDC = $\overline{\text{ENVLD}}$ = ENDT = 0
ENAC = ENFEL = 1
THEN = 1
ENUVD = 0
TXEN = ATEN = 0
RXOCEN = 1
Wait 1 sec
RXOCEN = 0
OFFSET = RXMON[7:0]
OFFSET += 0x08
RXMON[7:0] = *OFFSET*
Exit

3.2.3 Initial Line State Configuration

The default condition of the device is to be on hook after reset and startup.

The registers used in this procedure are:

0x05	**ENGPIO7**	**ENGPIO6**	**ENGPIO5**	**ENPCLKDT**	**ENAPOL**	**ENDET**	**ENSYNL**	**ENRGDT**
Write	X	X	X	X	1	X	1	X

0x12	**OFH**	**ENDC**	**ENAC**	**ENSHL**	**$\overline{\text{ENLVD}}$**	**ENFEL**	**ENDT**	**ENNOM**
Write	0	0	0	0	0	1	0	0

0x13	**DCIV1**	**DCIV0**	**ILM**	**RSVD**	**PLDM**	**OVDTH**	**IDISPD**	**Res**
Write	*VAL1[1:0]*		0	0	0	0	0	0

0x14	**TXBST**	**DAA1**	**DAA0**	**Res**	**RXBST**	**RLPNH**	**RXG1**	**RXG0**
Write	X	X	X	0	X	X	X	X

0x15	**ENOLD**	**DISNTR**	**Res**	**CIDM**	**THEN**	**ENUVD**	**ENOVD**	**ENOID**
Write	0	0	0	0	1	0	0	0

0x16	**TXEN**	**RXEN**	**RLPNEN**	**ATEN**	**ACZ3**	**ACZ2**	**ACZ1**	**ACZ0**
Write	1	1	0	1	*VAL2[3:0]*			

0x17	**Res**	**Res**	**RXOCEN**	**Res**	**Res**	**Res**	**Res**	**Res**
Write	0	0	0	0	0	0	0	0

The temporary variables defined in this procedure are:

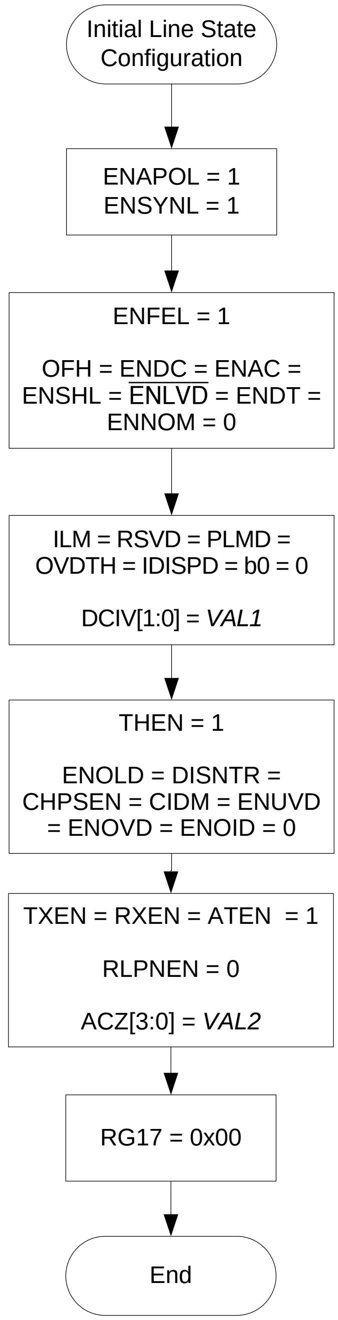
VAL1 = Set the DCIV bits to the desired DC current voltage characteristic as defined in the *73M1866B/73M1966B Data Sheet*.
VAL2 = System appropriate value for ACZ[3:0].

Begin

1. Set ENAPOL =1 and ENSYNL = 1, to allow automatic polling of line side registers (write RG05 = xxxx_1x1x).
2. Set ENFEL = 1, OFH = ENDC = ENAC = ENSHL = $\overline{\text{ENVLD}}$ = ENDT = ENNOM = 0 (Write RG12 = 0x40).
3. Set ILM = RSVD = PLDM = OVDTH = IDISPD = Res = 0, DCIV = *VAL1* (Write RG13 = *Val1[1:0]*00_0000).
4. Write RG14 = xxx0_xxxx.
5. Set THEN = 1, set ENOLD = DISNTR = CHPSEN = CIDM = ENUVD = ENOVD = ENOID = 0 (Write RG15 = 0000_1000).
6. Set TXEN = RXEN = ATEN = 1, RLPNEN = 0 and ACZ[3:0] =*VAL2[3:0]* (Write RG16 = 1101_*VAL2[3:0]).*
7. Set RXOCEN and all Res bits in RG17 = 0 (write RG17 = 0x00).

End

After this point the host may selectively enable other on-hook operations such as ring detection or CID mode.


Initial Line State Configuration
ENAPOL = 1
ENSYNL = 1
ENFEL = 1
OFH = ENDC = ENAC = ENSHL = ENLVD = ENDT = ENNOM = 0
ILM = RSVD = PLMD = OVDTH = IDISPD = b0 = 0
DCIV[1:0] = VAL1
THEN = 1
ENOLD = DISNTR = CHPSEN = CIDM = ENUVD = ENOVD = ENOID = 0
TXEN = RXEN = ATEN = 1
RLPNEN = 0
ACZ[3:0] = VAL2
RG17 = 0x00
End

4 On-Hook Procedures

The on-hook procedures described in this section include:

- CID Mode
- Off-Hook Request
- Ring Detection and Line Voltage Reversal
- Line-in-use and Loss of Battery Feed

4.1 CID Mode

It is possible (while on hook) to allow the RGP/RGN input, which is AC coupled to the line through high voltage capacitors, to be transmitted on the PCM bus. This allows the host to listen for Caller ID information without going off hook.

The registers used in this procedure are:

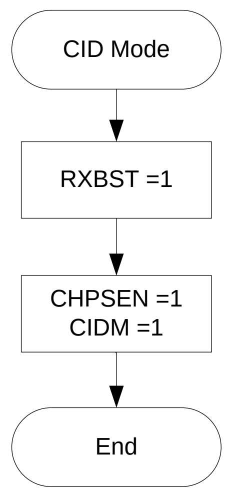
0x14	TXBST	DAA1	DAA0	Res	RXBST	RLPNH	RXG1	RXG0
Write	X	X	X	X	1	X	X	X

0x15	ENOLD	DISNTR	Res	CIDM	THEN	ENUVD	ENOVD	ENOID
Write	X	X	1	1	X	X	X	X

Begin

1. Set RXBST = 1 (Write RG14 = xxxx1xxx).
2. Set CHPSEN = 1, CIDM = 1 (write RG15 = xx11_xxxx).

End



4.2 Off-Hook Request

The registers used in the off-hook request procedure are:

0x08	TXDG -12	TXDG - 6	TXDG +3.5	TXDG +2	TXDG +1	TXDG +0.5	TXDG +0.25	TXDG +0.125
Write	*VAL3*							

0x09	RXDG -12	RXDG - 6	RXDG +3.5	RXDG +2	RXDG +1	RXDG +0.5	RXDG +0.25	RXDG +0.125
Write	*VAL4*							

0x12	OFH	ENDC	ENAC	ENSHL	$\overline{\text{ENLVD}}$	ENFEL	ENDT	ENNOM
Write	X/X/1/X	1/X/X/X	X/X/0 or 1/1	X	X	1/X/X/X	X	X/0/X/1

0x13	DCIV1	DCIV0	ILM	RSVD	PLDM	OVDTH	IDISPD	Res
Write	*VAL1[1:0]*		X	X	X	X	X	X

0x14	TXBST	DAA1	DAA0	Res	RXBST	RLPNH	RXG1	RXG0
Write	X	X	X	X	0	X	X	X

0x15	ENOLD	DISNTR	Res	CIDM	THEN	ENUVD	ENOVD	ENOID
Write	X	X	0/X	0/X	X/1	X	X	X

0x16	TXEN	RXEN	RLPNEN	ATEN	ACZ3	ACZ2	ACZ1	ACZ0
Write	X/1	X/1	X	X/1	*VAL2[3:0]* / X			

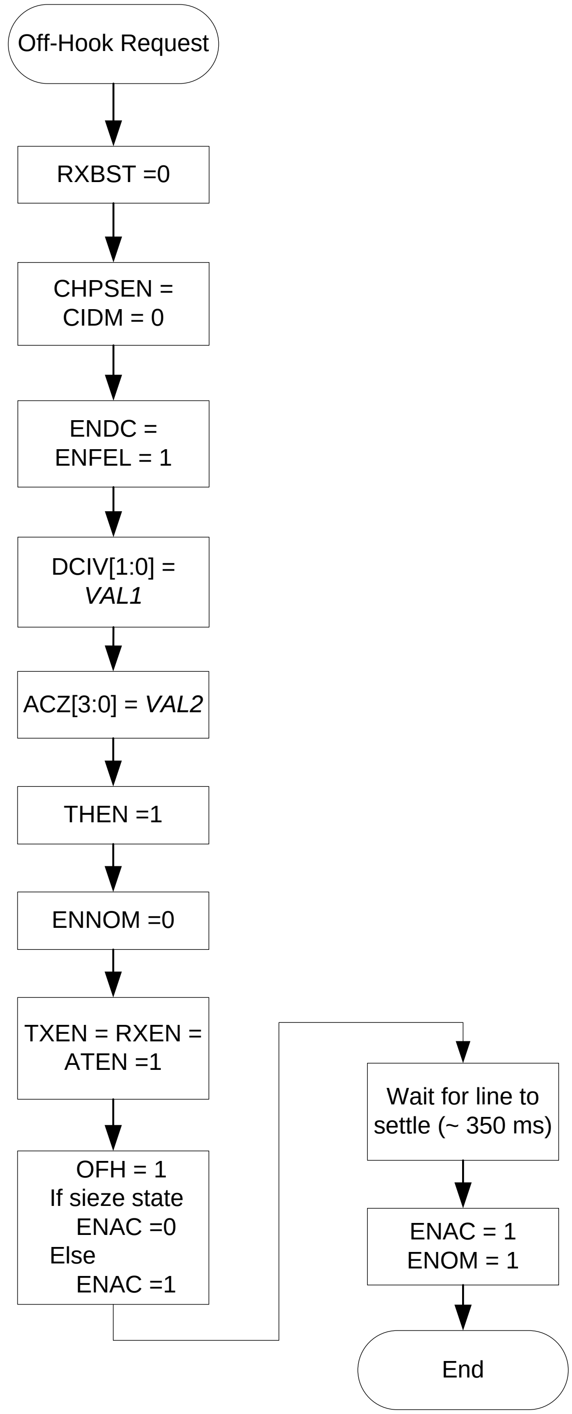
The temporary variables defined in this procedure are:

VAL1 = System appropriate value for DCIV[1:0].
VAL2 = System appropriate value for ACZ[3:0].
VAL3 = System appropriate value for Tx Gain (RG08).
VAL4 = System appropriate value for Rx Gain (RG09).

Begin

1. Set RXBST = 0 (write xxxx_0xxx to RG14).
2. Set CIDM = 0 (write xx00_xxxx to RG15).
3. Set ENDC = ENFEL =1 (write x1xx_x1xx to RG12).
4. Set DCIV[1:0] = *VAL1* (write *VAL1[1:0]*xx_xxxx to RG13).
5. Set ACZ[3:0] = *VAL2* (write xxxx_*VAL2[3:0]* to RG16).
6. Set THEN = 1 (write xxxx_1xxx to RG15).
7. Set ENNOM = 0 (write xxxx_xxx0 to RG12).
8. Set TXEN = RXEN = ATEN = 1 (write 11x1_xxxx to RG16).
9. Set Rx Gain and Tx Gain as needed. Write *VAL3* to RG08, write *VAL4* to RG09.
10. Set OFH = 1. If using sieze state, set ENAC =0 else set ENAC =1. (write 1x0x_xxxx or 1x1x_xxxx to RG12).
11. Wait until line settles (typical maximum is 350 ms).
12. Set ENAC = ENNOM =1 (write xx1x_xxx1 to RG12).

End


Off-Hook Request
RXBST =0
CHPSEN =
CIDM = 0
ENDC =
ENFEL = 1
DCIV[1:0] =
VAL1
ACZ[3:0] = VAL2
THEN =1
ENNOM =0
TXEN = RXEN =
ATEN =1
OFH = 1
If sieze state
ENAC =0
Else
ENAC =1
Wait for line to
settle (~ 350 ms)
ENAC = 1
ENOM = 1
End

4.3 Ring Detection and Line Voltage Reversal

When the 73M1966 is in on-hook mode, and when the ring detect interrupt is enabled (ENRGDT), a ring signal can be detected. Figure 3 shows the possible scenarios that can be encountered on the line.

The threshold voltage for detecting the ring signal is programmable in the 73M1x66B (RGTH[1:0]). When the line voltage exceeds the programmed threshold, the 73M1x66B asserts the hardware interrupt output pin ($\overline{\text{INT}}$) and asserts both RGDT and RGMON in the interrupt register, as illustrated by "1" and "3" in Figure 3. Note that the behavior is identical when the line voltage goes below the programmed threshold, as shown by "2" and "7" in Figure 3.

When the host reads the 73M1x66B interrupt register, RGDT is cleared, while RGMON remains asserted as long as:

1. The line voltage is above (or below) the upper (lower) threshold level shown in Figure 3 and
2. A time period of 25 ms has not elapsed after the line voltage crossed back over the threshold level. The activity of the timer ruling RGMON is indicated by "4" and "5" (as well as "8" and "9") in Figure 3.

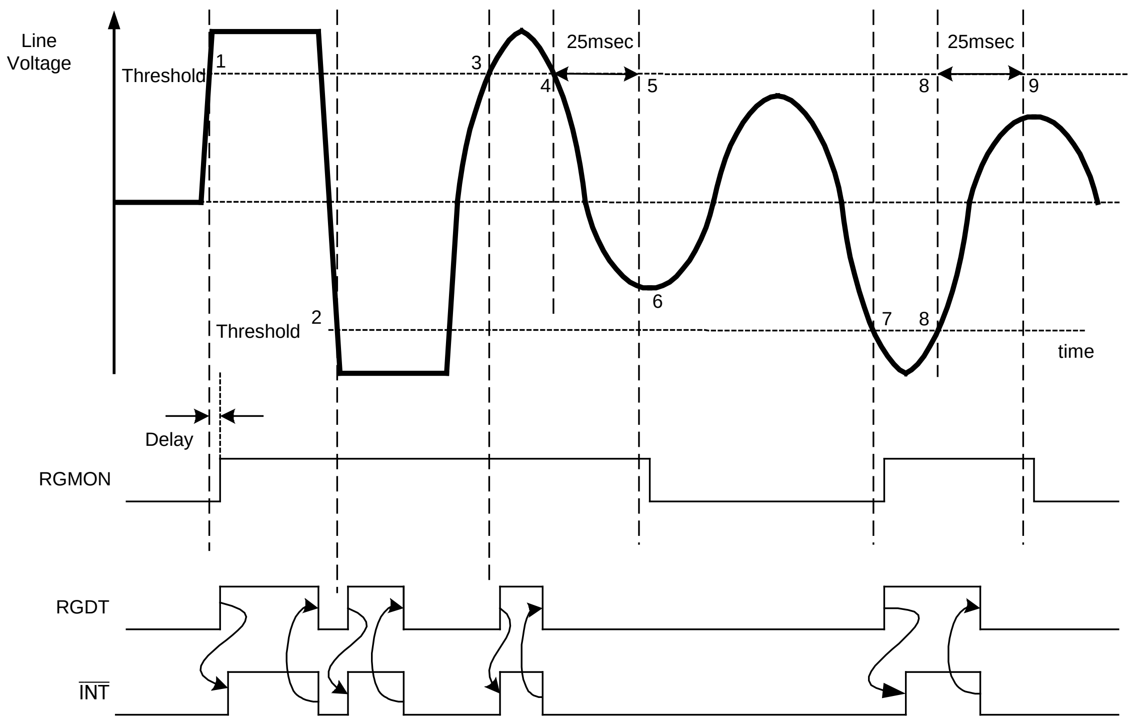


Figure 3: Ring Detection

If the line voltage does not exceed the threshold within the 25 ms period, RGMON becomes de-asserted, as illustrated by "6" and "9" in Figure 3.

Upon the first reception of the first RGDT interrupt, if no other ring interrupt was received during a sufficiently long period then it can be assumed that a Line Polarity Reversal has occurred. To filter out spurious ring events (from a parallel device going off hook or battery loss), the driver should check that the line voltage is the same as previously before determining that the single ring event was, in fact, a line polarity reversal.

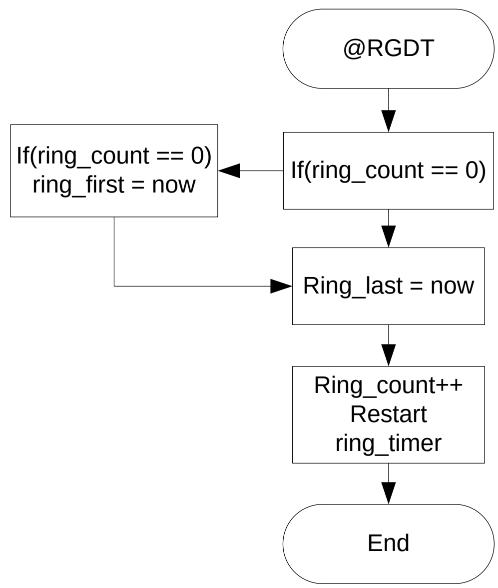
The registers used for Ring Detection and Line Voltage Reversal are:

0x05	ENGPIO7	ENGPIO6	ENGPIO5	ENPCLKDT	ENAPOL	ENDET	ENSYNL	ENRGDT
Write	X	X	X	X	X	X	X	1

0x0E	FRCVCO	Res	Res	Res	Res	Res	RGTH1	RGTH0
Write	X	X	X	X	X	X	*Threshold*	

0x03	GPIO7	GPIO6	GPIO5	PCLKDT	RGMON	DET	SYNL	RGDT
Write	X	X	X	X	X	X	X	?

Set the ring detect threshold voltage[1] (*Threshold*) in the RGTH[1:0] bits in Register 0x0E.
Set ENRGT = 1 in Register 0x05 to enable the RGMON and RGDT interrupts. See Section 6 Interrupt Processing for more information on interrupts.



The system variables defined in this procedure are:

ring_count = initial 0, keeps track of number of ring interrupts
ring_first = time of first ring interrupt
ring_last = time of last ring interrupt
ring_frequency = ring frequency in HZ
ring_duration = ring duration

Begin @ RGDT interrupt (RGDT = 1)

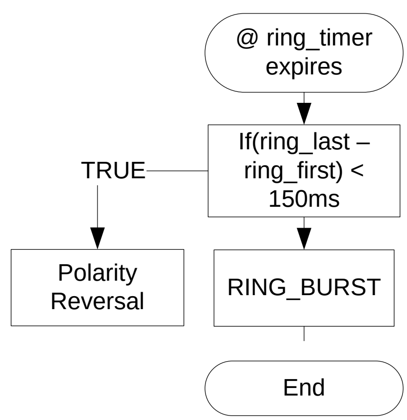
1. If (*ring_count* == 0) *ring_first* = now;
2. *ring_last* = now
3. start/restart ring_timer for approx 150 ms

End

Begin @ ring_timer expire

1. If (*ring_last* - *ring_first*) < 150 ms Polarity Reversal Event
2. Else Ring Burst
   a. *ring_duration* = *ring_last* – *ring_first*
   b. *ring_frequency* – 2* *ring_count* / *ring_duration*)

End



[1] The ring detect threshold is country specific. The recommended values are shown in the *73M1x66 Worldwide Design Guide.*

4.4 Line-in-use and Loss of Battery Feed

When the FXO line is in the on-hook mode, the driver monitors the voltage on the line (typically 48 V in the US) at regular intervals. A loss of battery voltage or a line-in-use event can affect the value of the line voltage.

The loss of battery voltage results from disconnecting the phone line or a central office failure. In this case, the line voltage falls to zero volts after a time constant.

The line-in-use event occurs when another phone connected to that line is going off-hook. The line voltage drops to about 6 V when the line is in use.

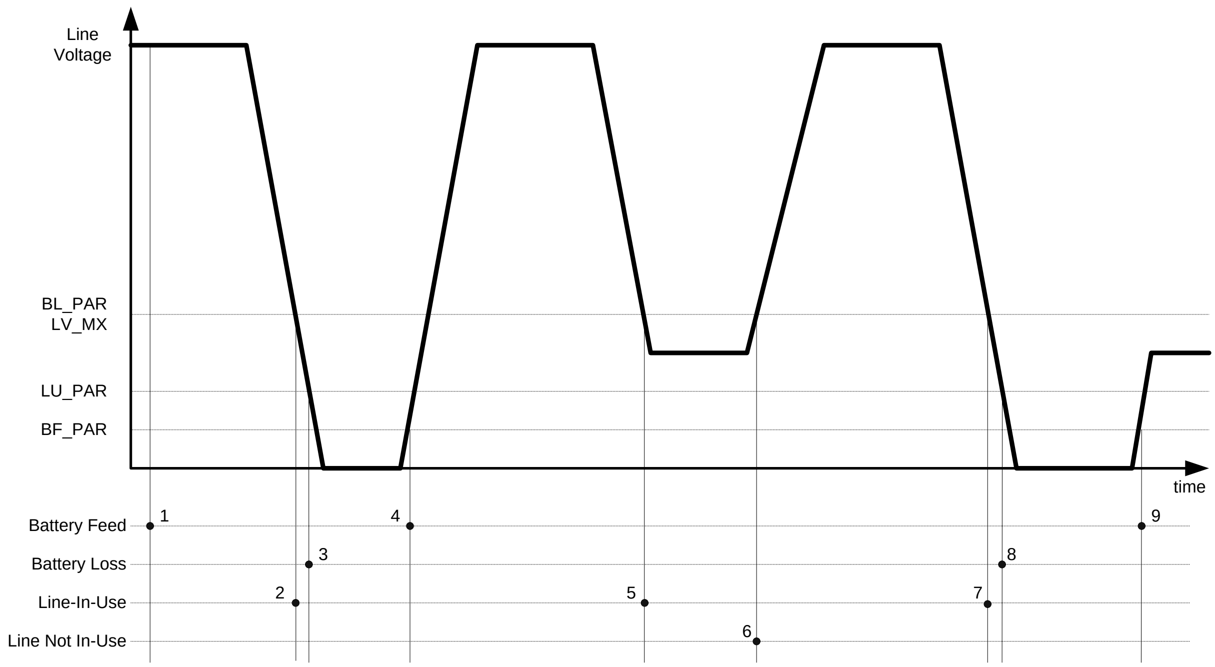


Figure 4: Battery Feed and Line-in-Use Detection

The LV register (0x1B) can be polled to determine the line voltage. In an on-hook case (and using the reference schematics) the line voltage can be calculated as follows:

*Line Voltage = ((LV[7:0] & 0xFE) * 1.13) + 1.4V*

The register which is read to determine the Line voltage is:

0x1B	**LV7**	**LV6**	**LV5**	**LV4**	**LV3**	**LV2**	**LV1**	**Res**
Read	*Line Voltage[7:1]*							

5 Off-Hook Procedures

The off-hook procedures described in this section include:

- Barrier Synch Loss
- On-hook Request
- Parallel Pickup Event Detection

5.1 Barrier Synch Loss

When in an off-hook state the host must be ready to act in case of a barrier failure. The host should attempt to put the device in an on-hook state and then reset the PLL and barrier so they can resync.

The on-hook request procedure is described in Section 5.2. The procedure for barrier synchronization is described in Section 3.2.1.

5.2 On-hook Request

The registers used in the on-hook request procedure are:

0x12	**OFH**	**ENDC**	**ENAC**	**ENSHL**	$\overline{\textbf{ENLVD}}$	**ENFEL**	**ENDT**	**ENNOM**
Write	0	0	0	X	X	X	X	X

Begin

1. Set ENDC = ENAC = OFH = 0 (write RG12 = 000x_xxxx).

End

5.3 Parallel Pickup Event Detection

To be defined.

6 Interrupt Processing

During the course of operation the 73M1x66 can be expected to generate interrupts when errors or other events occur that require immediate action. Each interrupt source corresponds to a bit in register 0x03.

This section will cover each interrupt and the appropriate actions to process it. The registers used in the procedure are:

0x03	GPIO7	GPIO6	GPIO5	PCLKDT	RGMON	DET	SYNL	RGDT
Read	?	?	?	?	?	?	?	?

0x05	ENGPIO7	ENGPIO6	ENGPIO5	ENPCLKDT	ENAPOL	ENDET	ENSYNL	ENRGDT
Write	X	X	X	X	1	X	X	X

The temporary variables defined in the procedure are:

IntSrc = Interrupt sources (read from Register 0x03).
GPIO[7:5] bits indicate a GPIO Interrupt.
PCLKDT bit indicates a PCLK error: PCLKDT Interrupt.
DET bit indicates a line condition error: DET Interrupt.
SYNL bit indicates a barrier failure: SYNL Interrupt.
RGMON and RGDT bits indicate a line signal detection: RGDT and RGMON Interrupts.

Begin

1. Wait for Interrupt.
2. Read Register 0x03 to determine the interrupt source(s), to clear the register value and de-assert the INT pin (*IntSrc* = RG03).
3. Read RG05 (IntMsk = RG05.
4. Set ENAPOL = 1 (write xxxx_1xxx to RG05).
5. Process each interrupt source.
6. Re-enable interrupts set RG05 = IntMsk.
7. goto 1.

End

6.1 GPIO Interrupt

GPIO interrupts are a user dependent function controlled through the POL and ENGPIO registers. Please refer to the *73M1866B/73M1966B Data Sheet* for their individual functionality. The functional requirements of the software to handle a GPIO interrupt are user dependent.

0x05	ENGPIO7	ENGPIO6	ENGPIO5	ENPCLKDT	ENAPOL	ENDET	ENSYNL	ENRGDT
Write	1	1	1	X	1	X	X	X

0x06	POL7	POL6	POL5	Res	Res	Res	Res	Res
Write	0/1	0/1	0/1	X	X	X	X	X

Set the ENGPIO[7:5] bits of RG05 to enable the corresponding GPIO[7:5] interrupt(s).
Set the POL[7:5] bit of RG06 to the desired polarity.

Begin

1. If GPIO7 or GPIO6 or GPIO5 = 1 (*IntSrc*[7:5] ≠ 0) then goto user defined GPIO Interrupt Processing.

End

6.2 PCLKDT Interrupt

The triggering of the PCLKDT interrupt indicates that the device has detected a PCLK error. When found, it can be assumed that there is a problem with the PCLK. If the error is temporary, the error can be cleared by resetting the PLL. If the error is due to an external clock failure this error will not clear.

0x05	ENGPIO7	ENGPIO6	ENGPIO5	ENPCLKDT	ENAPOL	ENDET	ENSYNL	ENRGDT
Write	X	X	X	1	1	X	X	X

Set the ENPCLKDT bit in Register 0x05 to enable the PCLKDT interrupt.

Begin

1. If PCLKDT =1 (*IntSrc* = xxx1_xxxx) then goto PLL Recovery Process (see Section 3.1.2 PCLK Clock Recovery and PLL Lock Detection).

End

6.3 DET Interrupt

The triggering of the DET interrupt indicates that the device has detected one of several line condition errors. When found, it can be assumed that there is a problem with the line status. The general corrective action is for the line side device to go on hook. These interrupts are only valid when the device is in an off-hook condition.

0x05	ENGPIO7	ENGPIO6	ENGPIO5	ENPCLKDT	ENAPOL	ENDET	ENSYNL	ENRGDT
Write	X	X	X	X	1	1	X	X

0x12	OFH	ENDC	ENAC	ENSHL	$\overline{\text{ENLVD}}$	ENFEL	ENDT	ENNOM
Write	1	1	1	x	x	1	1	x

0x13	DCIV1	DCIV0	ILM	RSVD	PLDM	OVDTH	IDISPD	Res
Write	X	X	0	0	0	1/0	0	0

0x14	TXBST	DAA1	DAA0	Res	RXBST	RLPNH	RXG1	RXG0
Write	X	X	X	0	X	X	X	X

0x15	ENOLD	DISNTR	Res	CIDM	THEN	ENUVD	ENOVD	ENOID
Write	1	0	0	0	1	1	1	1

0x1E	ILMON	UVDET	OVDET	OIDET	OLDET	SLLS	Res	Res
Read	X	?	?	?	?	X	X	X

Set the ENDET bit in Register 0x05 to enable the DET interrupt.

Begin

1. If DET = 1 (*IntSrc* = xxxx_x1xx).
2. Read RG1E
3. If OVDET = 1, report condition to user and put device on hook.
4. If OIDET = 1, report condition to user and put device on hook.
5. If OLDET = 1, report condition to user and put device on hook.
6. If UVDET = 1, report condition to user and put device on hook.

End

6.4 SYNL Interrupt

The triggering of the SYNL interrupt indicates that the device has detected a failure in the barrier between the line and host side device. When found, it can be assumed that there is a problem with the barrier. This failure is usually a problem with the PLL operation, therefore, the recommended way to deal with this interrupt is to reset the device PLL.

0x05	ENGPIO7	ENGPIO6	ENGPIO5	ENPCLKDT	ENAPOL	ENDET	ENSYNL	ENRGDT
Write	X	X	X	X	1	X	1	X

Set the RNSYNL bit in Register 0x05 to enable the SYNL interrupt.

Begin

1. If SYNL = 1 (*IntSrc* = xxxx_xx1x), goto PLL Recovery Process (see Section 3.1.2 PCLK Clock Recovery and PLL Lock Detection).

End

6.5 RGDT and RGMON Interrupts

The triggering of an RGDT or RGMON interrupt indicates that the device has detected an AC signal on the line greater than the threshold programmed through the RGTH register.

0x05	ENGPIO7	ENGPIO6	ENGPIO5	ENPCLKDT	ENAPOL	ENDET	ENSYNL	ENRGDT
Write	X	X	X	X	1	X	X	1

0x0E	FRCVCO	Res	Res	Res	Res	Res	RGTH1	RGTH0
Write	X	X	X	X	X	X	*VAL1[1:0]*	

Set the ENRGDT bit in Register 0x05 to enable the RGDT and RGMON interrupts.

Begin

1. If RGDT = 1 (*IntSrc* = xxxx_xxx1) or RGMON = 1 (*IntSrc* = xxxx_1xxx), goto Ring Detection and Line Polarity Reversal (see Section 4.3).

End

7 Register Summary

Address (hex)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02	TMEN	Res	Res	Res	Res	ENLPW	Res	Res
0x03	GPIO7	GPIO6	GPIO5	PCLKDT	RGMON	DET	SYNL	RGDT
0x04	DIR7	DIR6	DIR5	Res	REVHSD3	REVHSD2	REVHSD1	REVHSD0
0x05	ENGPIO7	ENGPIO6	ENGPIO5	ENPCLKDT	ENAPOL	ENDET	ENSYNL	ENRGDT
0x 06	POL7	POL6	POL5	Res	Res	Res	Res	Res
0x07	Res	Res	Res	Res	Res	Res	DTST1	DTST0
0x08	TXDG -12	TXDG -6	TXDG +3.5	TXDG +2	TXDG +1	TXDG +0.5	TXDG +0.25	TXDG +0.125
0x09	RXDG -12	RXDG -6	RXDG +3.5	RXDG +2	RXDG +1	RXDG +0.5	RXDG +0.25	RXDG +0.125
0x0D	LOKDET	SLHS	Res	Res	RSTLSBI	Res	Res	Res
0x0E	FRCVCO	Res	Res	Res	Res	Res	RGTH1	RGTH0
0x0F	ENFEH	PWDN	SLEEP	Res	Res	Res	Res	Res
0x10	Res	Res	Res	CMVSEL	CMTXG1	CMTXG0	CMRXG1	CMRXG0
0x12	OFH	ENDC	ENAC	ENSHL	$\overline{\text{ENLVD}}$	ENFEL	ENDT	ENNOM
0x13	DCIV1	DCIV0	ILM	Res	PLDM	OVDTH	IDISPD	SEL16K
0x14	TXBST	DAA1	DAA0	Res	RXBST	RLPNH	RXG1	RXG0
0x15	ENOLD	DISNTR	Res	CIDM	THEN	ENUVD	ENOVD	ENOID
0x16	TXEN	RXEN	RLPNEN	ATEN	ACZ3	ACZ2	ACZ1	ACZ0
0x17	Res	Res	RXOCEN	Res	Res	Res	Res	Res
0x18	TEST3	TEST2	TEST1	TEST0	Res	Res	Res	Res
0x19	POLL	MATCH	Res	IDL2	INDX3	INDX2	INDX1	INDX0
0x1A	RNG7	RNG6	RNG5	RNG4	RNG3	RNG2	RNG1	RNG0
0x1B	LV7	LV6	LV5	LV4	LV3	LV2	LV1	Res
0x1C	LC6	LC5	LC4	LC3	LC2	LC1	LC0	Res
0x1D	REVLSD3	REVLSD2	REVLSD1	REVLSD0	Res	Res	Res	Res
0x1E	ILMON	UVDET	OVDET	OIDET	OLDET	SLLS	Res	Res
0x1F	POLVAL7	POLVAL6	POLVAL5	POLVAL4	POLVAL3	POLVAL2	POLVAL1	POLVAL0
0x20	TPOL	TTS6	TTS5	TTS4	TTS3	TTS2	TTS1	TTS0
0x21	RPOL	RTS6	RTS5	RTS4	RTS3	RTS2	RTS1	RTS0
0x22	SR	ADJ	RCS2	RCS1	RCS0	TCS2	TCS1	TCS0
0x23	PCMEN	MASTER	PCODE3	PCODE2	PCODE1	PCODE0	LIN	LAW
0x24	Res	Res	Res	Res	Res	Res	Res	LB
0x25	RXOM7	RXOM6	RXOM5	RXOM4	RXOM3	RXOM2	RXOM1	RXOM0

8 Related Documentation

The following 73M1x66B documents are available from Teridian Semiconductor Corporation:

73M1866B/73M1966B Data Sheet
73M1866B/73M1966B Demo Board User Manual
73M1866B/73M1966B GUI User Guide
73M1866B/73M1966B Layout Guidelines
73M1x66 Worldwide Design Guide
73M1866B/73M1966B Programming Guidelines
73M1866B/73M1966B Reference Driver User Guide

9 Contact Information

For more information about Teridian Semiconductor products or to check the availability of the 73M1866B and 73M1966B, contact us at:

6440 Oak Canyon Road
Suite 100
Irvine, CA 92618-5201

Telephone: (714) 508-8800
FAX: (714) 508-8878
Email: modem.support@teridian.com

For a complete list of worldwide sales offices, go to http://www.teridian.com.

Revision History

Revision	Date	Description
1.0	5/21/2009	First publication.
1.1	8/5/2009	Removed the CHPSEN bit. Made RG15 bit 5 "Reserved". Re-wrote the Barrier Synchronization Recovery procedure. Miscellaneous editorial changes.
1.2	8/20/2009	In Section 3.2.1, changed "SYNL == 0" in the flowchart to "SYNL == 1".
1.3	3/26/2010	In Section 3.1.3, deleted "The recommended procedure is to toggle SLEEP once before checking LOKDET for the first time." In Section 3.2.2,