PM5365

TEMAP

VT/TU Mapper and M13 Multiplexer

Datasheet Errata

Proprietary and Confidential

Released

Issue 2: October, 2001

Legal Information

Copyright

PMC-2001480 (R2)

Disclaimer

None of the information contained in this document constitutes an express or implied warranty by PMC-Sierra, Inc. as to the sufficiency, fitness or suitability for a particular purpose of any such information or the fitness, or suitability for a particular purpose, merchantability, performance, compatibility with other parts or systems, of any of the products of PMC-Sierra, Inc., or any portion thereof, referred to in this document. PMC-Sierra, Inc. expressly disclaims all representations and warranties of any kind regarding the contents or use of the information, including, but not limited to, express and implied warranties of accuracy, completeness, merchantability, fitness for a particular use, or non-infringement.

In no event will PMC-Sierra, Inc. be liable for any direct, indirect, special, incidental or consequential damages, including, but not limited to, lost profits, lost business or lost data resulting from any use of or reliance upon the information, whether or not PMC-Sierra, Inc. has been advised of the possibility of such damage.

Trademarks

TEMAP, SBI, and PMC-Sierra are trademarks of PMC-Sierra, Inc.

Patents

The technology discussed is protected by one or more Patents. Relevant patent applications and other patents may also exist.

Contacting PMC-Sierra

PMC-Sierra
8555 Baxter Place Burnaby, BC
Canada V5A 4V7

Tel: +1.604.415.6000
Fax: +1.604.415.6204

Document Information: document@pmc-sierra.com
Corporate Information: info@pmc-sierra.com
Technical Support: apps@pmc-sierra.com
Web Site: http://www.pmc-sierra.com

Revision History

Issue No.	Issue Date	Details of Change
2	October 1, 2001	Updated to reflect new PM5365 TEMAP Datasheet issued August 2001 (PMC-1991148 r3) and the PM5365 TEMAP Register Description issued July 2001 (PMC-1990682 r2).
1	July 12, 2001	This document is a notice of additional information and error corrections to be inserted into the PM5365 TEMAP Datasheet, document number PMC01991148, Issued February 2000 and the PM5365 TEMAP Register Description, document number PMC-1990682, Issued March 2000.

Table of Contents

1 Introduction

In this document:

- Section 2 lists the known functional errata for the Production Released Version of the PM5365 TEMAP.
- Section 3 lists documentation errors found in the PM5365 TEMAP Datasheet issued August 2001 (PMC-1991148 r3) and the PM5365 TEMAP Register Description issued July 2001 (PMC-1990682 r2).

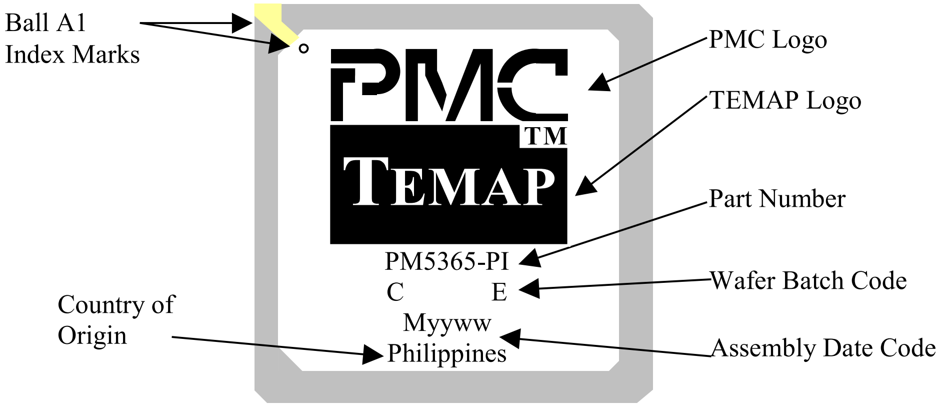
1.1 Device Identification

The information contained in Section 2 relates to the Production Released version of the PM5365 TEMAP device only. This version is identified by the letter E at the end of the wafer batch code, which is the device revision code, and the designation –PI at the end of the part number instead of the designation –PI-P.

Register 0002H: Revision/Global PMON Update identifies the Production Release revision of the PM5365 TEMAP using ID[3:0]=0101.

Note: This errata only applies to issues specific to the production released PM5365-PI TEMAP device. It does not apply to any of the prototype TEMAP devices.

Figure 1 PM5365 Branding Diagram



1.2 References

- Issue 3 of the PM5365 TEMAP Datasheet (PMC-1991148).
- Issue 2 of the PM5365 TEMAP Register Description (PMC-1990682).

2 Device Functional Deficiency List

This section lists the known functional deficiencies for the Production Released version of the PM5365 TEMAP as of the publication date of this document.

Please report any functional deficiencies not covered in this errata to PMC-Sierra.

2.1 Byte Deletion/Insertion in VT/TU Mapping Mode

While rare, it is possible for jitter on XCLK or the selected T1/ E1 transmit clock to cause an overflow or underflow in the transmit mapper (TTMP) FIFO. The T1/E1 transmit timing can be sourced from the two clock master sources, the CTCLK input or the recovered T1/E1 clock, or from the two clock slave sources, an ECLK[x] input or the SBI tributary rate received on the SBI add bus. By design, this will result in the loss of a frame of data. The initialization procedure shown in Appendix A on page 13 will push the FIFO closer to the center such that it can withstand more jitter from these clocks to reduce any impact on the datapath. If the FIFO is close to an underflow or overflow condition and jitter does push it over or under, the FIFO will insert or delete a byte of data, pushing itself away from these states and giving itself at least another 8 UI of margin against future jitter events on XCLK or the selected T1/E1 transmit clock. It is recommended that the initialization sequence shown in Appendix A be used for all revisions of the TEMAP device.

2.2 VT-AIS Tributary Corruption

An issue in the TEMAP demapper results in the VT-AIS causing corruption to the previous tributary of SPE#3 only. To circumvent this issue apply the software workaround outlined in Appendix B: SPE Configuration to Prevent TU-AIS Data Corruption.

2.3 TEMAP Initialization Sequencing in VT/TU Mapping Mode

When using TEMAP in VT/TU mapping mode with any system side option, care should be taken in the initialization of the device to make sure it comes up in the proper state. Please use the recommended initialization sequence below.

1. Initialization of transmit timing options

In TEMAP Register 004H+80H*N: T1/E1 Egress Line Interface Configuration select the transmit timing source by setting PLLREF[1:0] correctly. The T1/E1 transmit timing can be sourced from the two clock master sources, the CTCLK input or the recovered T1/E1 clock, or from the two clock slave sources, an ECLK[x] input or the SBI tributary rate received on the SBI add bus. Do not bypass the TJAT. This will ensure that when the transmit timing source is optionally referenced to the recovered timing for the tributary the transition will be smooth. TJAT is also used in the TTMP centering procedure outlined in section 3 below. Note that in the case where the ECLK[x] input or the SBI add bus is the source of the transmit timing, looptiming via the PLLREF[1:0] bits is invalid.

2. Initialization of RTDM

Program Register 12E0H: RTDM Pointer Justification Rate Control for a T1 rate control of 0.4 seconds (T1RATE[1:0]=10) and an E1 rate control of 0.4 seconds (E1RATE[1:0]=10) depending if T1 or E1 mode is used.

3. Initialization of RJAT and TJAT for all TU11/TU12 T1/E1 framing configurations

In the initial setting of the RJAT (registers 0011+80*N, 0012+80*N and 0013+80*N) perform both PLL reset and FIFO centering with the following sequence. In the RJAT configuration register, set the CENT and LIMIT bits to 0 and the reserved and SYNC bits to 1. Program the RJAT N1 and N2 divisors to 2F and then go back to write a logic 0 to the CENT, LIMIT and SYNC bits and a logic 1 to the reserved bit. This is done for both T1 and E1. The time between writing the N1 and N2 values and clearing the SYNC bit needs to be as short as possible.

Note the following applies only to configurations in which the transmit timing source is equal to line rate (T1 or E1). This is applicable for all transmit timing modes except when the CTCLK input is a multiple of 8 kHz. For that configuration use the TTMP Reprov workaround as outlined in Appendix A: Pseudo Code for Detection of Tributary Errors. The following sequence does not apply to Revision A devices. The TTMP Reprov workaround must be used for Revision A devices.

The TJAT for a specific tributary should be initialized after the transmit tributary mapper's (TTMP) tributary is provisioned and the transmit timing source has been selected. Then for the initial setting of the TJAT (registers 0015+80*N, 0016+80*N and 0017+80*N) perform both PLL reset and FIFO centering with the following sequence. Program the TJAT N1 and N2 divisors to 2F and then write a logic 0 to the CENT, LIMIT and SYNC bits and a logic 1 to the reserved bit. Next the N2 divisor should be programmed to the value A specified in Table 1 depending upon the mode selected (T1 or E1) frequency referenced to TJAT initially. Delay 100ms, then change the N1 value to B specified in Table 1. Delay another 100ms then change the N1 value to C specified in Table 1. Then in the TJAT configuration register, set the CENT and LIMIT bits to 0 and the reserved and SYNC bits to 1. Rewrite the value C into the N1 register and then clear the SYNC bit back to 0. Accurate delays need to be implemented in this procedure. The speed is important in order to create long frequency steps that are required to stabilize the TTMP. In addition, keep the relative time between re-writing the C value to N2 clearing SYNC bit as short as possible.

It is recommended to have the TTMP FIFO check from Appendix A on page 13 in place following the TJAT centering procedure to make sure the centering was successful. If this check is to be performed immediately after the above procedure then an additional 100ms of delay is required before the check. If the check of the TTMP FIFO is performed after all the tributaries have had the above procedure performed the additional delay is not required because considerable time will have passed since the centering procedure was performed on the first tributary. Example software scripts are as follows:

```
// RJAT initialization procedure

write temap 0013+80*N 22 // RJAT Configuration register: CENT=LIMIT=0, RESERVED=SYNC=1

write temap 0011+80*N 2F // RJAT N1
write temap 0012+80*N 2F // RJAT N2
write temap 0013+80*N 20 // RJAT Configuration register: CENT=LIMIT=SYNC=0, RESERVED=1

// TJAT centering procedure
write temap 0017+80*N 20 // TJAT Configuration register: CENT=LIMIT=SYNC=0, RESERVED=1
write temap 0015+80*N 2F // TJAT N1
write temap 0016+80*N 2F // TJAT N2
write temap 0016+80*N “A” // TJAT N2, see Table 1
wait 100ms
write temap 0015+80*N “B” // TJAT N1, see Table 1
wait 100ms
write temap 0015+80*N “C” // TJAT N1, see
write temap 0017+80*N 22 // TJAT Configuration register: CENT=LIMIT=0, RESERVED=SYNC=1
write temap 0015+80*N “C” // TJAT N1, see Table 1
write temap 0017+80*N 20 // TJAT Configuration register: CENT=LIMIT=SYNC=0, RESERVED=1
wait 100ms before performing TTMP FIFO pointer check
```

Table 1: N1 and N2 values for TJAT centering procedure

T1/E1 Transmit Clock (kHz)	T1			E1		
	A	B	C	A	B	C
1544	0xff	0x70	0xff	N/A	N/A	N/A
2048	N/A	N/A	N/A	0xff	0xa0	0xff

Note: This only applies to configurations where the T1/E1 transmit clock = Line Rate.

4. Procedure for Ingress datapath recovery.

Once the datapath has been restored, the RJAT should be centered.

For all cases use the removal of the TU-LOP alarm (located in register 1240H for TU#1 of TUG2#1) and the TU-AIS alarm (located in register 1241H for TU#1 of TUG2#1) as the trigger for centering the RJAT FIFO.

In framed E1 mode with the PMON block in the ingress direction you can optionally use the removal of the E1 RED Alarm (located in register 0067+80*N) as the trigger for centering the RJAT FIFO. The centering procedure will cause a sudden bit slip and thus will cause the E1 framer to have a single change of frame alignment. Once the centering procedure has been completed an interval of 50ms is required before, optionally, switching back to looptimed mode if applicable.

An example software script is as follows:

```
// RJAT centering procedure
write temap 0013+80*N 22 // RJAT Configuration register: CENT=LIMIT=0, RESERVED=SYNC=1
write temap 0011+80*N 2F // RJAT N1
write temap 0013+80*N 20 // RJAT Configuration register: CENT=LIMIT=SYNC=0, RESERVED=1

// Ingress SBI Reset if SBI is used
write temap INSBI(trib) ENBL 1 // re-writes a logic 1 to enable bit for selected tributary causing a reset
```

5. Algorithm for datapath corruption restoration when link is internally timed:

With the T1 or E1 link internally timed, upon detection of datapath corruption the RJAT centering procedure should be used before bringing the link back up as described in section 4 above.

A sample algorithm is as follows:

```
#define data_corruption_event // either framed or unframed as noted above processed by an interrupt
service routine
#define centerRJAT() // Centering procedure as shown above
#define declare_link_up // Routine for bringing up a link
#define declare_link_down // Routine for tearing down a link
#define ingress_SBI_reset  // Routine for resetting a tributary on the ingress SBI bus if SBI is used
if (!data_corruption_event) {
  centerRJAT(tributary)
  wait 50ms
  ingress_SBI_reset(trib)
  declare_link_up(tributary)
  } else {
     declare_link_down(tributary)
     }
```

6. Algorithms for switching from internally timed to looptimed and vice versa

When switching from looptimed to internally timed mode, or vice-versa, the initialization sequence is important. Once the desired mode has been set, the TJAT should be centered using the same procedure as noted above in section 3 and the connected device's SBI blocks should be reset if applicable. The TTMP pointer procedure listed in Appendix A on page 13 should also be followed.

Example pseudocode:

```
looptime(framer) {
write temux 0x04+(0x80*$framer) = 0x62
..centerTJAT(framer)
..check_TTMP_pointers(framer)
..}
internaltime(framer) {
..write temux 0x04+(0x80*$framer) = 0x61
..centerTJAT(framer)
..check_TTMP_pointers(framer)
}
```

7. Algorithm for datapath corruption and restoration when link is looptimed:

With the T1 or E1 link looptimed, upon detection of datapath corruption as defined in section 4 above, the transmit timing reference to TJAT must be changed to the normal transmit timing source. This occurs because the recovered clock for that link is not valid. Thus when the datapath is restored not only must the RJAT centering procedure from section 4 be used before bringing the link back up but the TJAT centering procedure from section 3 should be used after referencing it back to the recovered clock.

A sample algorithm is as follows:

```
#define data_corruption_event // either framed or unframed as noted above processed by an interrupt service routine
#define centerRJAT() // centering procedure as shown above
#define declare_link_up() // Routine for bringing up a link
#define declare_link_down() // Routine for tearing down a link
#define looptime() // write PLLREF[1:0]=10 of register 004H+80H*N
#define internaltime() write PLLREF[1:0]=01 of register 004H+80H*N

if (!data_corruption_event) {
  centerRJAT(tributary)
  wait_50ms
  looptime(tributary)  // includes TJAT center, see above
  declare_link_up(tributary)
    } else {
      wait 15ms
      internaltime(tributary) // includes TJAT center, see above
      declare_link_down(tributary)
      }
```

3 Documentation Deficiency List

This section of the document is a notification of additional information and known documentation deficiencies for Issue 3 of the PM5365 TEMAP Datasheet and Issue 2 of the PM5365 TEMAP Register Description, as of the publication date of this document.

Please report any documentation deficiencies not covered in this errata to PMC-Sierra.

3.1 Duplication of Ground Pin Descriptions

The description of ground pins N3, Y12, L20 and B12 has been duplicated in two rows of the pin description section, pins VSSQ[1:4] and VSS3.3[19:22]. These pins should only be described in the VSS3.3 row. The VSSQ row should be eliminated. In any event these pins should be connected to GND as described.

3.2 Connecting to the Telecom ADD bus via an External MUX

The pin description for LAC1J1V1 may need clarification. When connecting the TEMAP to the TelecomBus via an external MUX (instead of simply tri-stating the bus), the LAC1J1V1 signal should not be muxed. This means that when there are three TEMAPs connected to a Spectra's TelecomBus through the MUX, the LAC1J1V1 signal from only one of the three TEMAPs should be connected to the SPECTRA. The other two LAC1J1V1 signals should remain unconnected. All TEMAPs must have the LOCK0 bit in registers 1202H and 15E5H set the same.

3.3 Telecom ADD Bus Parity Generation

The TEMAP device cannot generate parity on the Telecom ADD bus when LAC1J1V1 is set to participate in the egress parity generation. Parity generation only helps check integrity of the bus connections and has no effect on the data path or with control of the device. If egress parity needs to be generated, LAC1J1V1 should not be used.

3.4 Telecom DROP Bus Parity Detection

The TEMAP device *can* correctly detect parity on the Telecom DROP bus when LDC1J1V1 is set to participate in the ingress parity detection. The Register Description documentation is erroneously missing this optionally selectable feature.

Accordingly, Register 1201H of the Register Description document should be amended as follows:

Register 1201H: SONET/SDH Master Ingress Configuration

Bit	Type	Function	Default
Bit 7	R/W	LDPE	0
Bit 6	R/W	ITMFEN	0
Bit 5	R/W	IVTPPBYP	0
Bit 4	R/W	ITSEN	0
Bit 3	R/W	INCLDPL	0
Bit 2	**R/W**	**INCLDC1J1V1**	**0**
Bit 1	R/W	LDOP	0
Bit 0	R/W	ICONCAT	0

INCLDC1J1V1:

The INCLDC1J1V1 bit controls whether the LDC1J1V1 input signal participates in the incoming parity calculations. When INCLDC1J1V1 is set high, the parity signal set includes the LDC1J1V1 input. When INCLDC1J1V1 is set low, parity is calculated without regard to the state of LDC1J1V1. Selection of odd or even parity is controlled by the LDOP bit.

Appendix A: Pseudo Code for Detection of Tributary Errors

```
# pseudo Code for TEMAP TTMP Pointer detection

checkTributaries {} {
    fifoSize = read (0x1580 + 0x66) + 0x08
    for {set tug3 0} {tug3 <=2} {incr tug3} {
            for {set tug2 0} {tug2 <= 6} {incr tug2}
                    for {set tu 0} {tu <= 3} {incr tu}
                            trib = getTrib{tug3,tug2,tu}
                            if {PROVbitSet(trib)} {
                                    if {checkTributary (trib fifoSize) == BAD_POINTER} {
                                            centerTJAT(trib) OR toggleProv (trib)
    # If T1/E1 Transmit Clock = Line Rate use centerTJAT(trib) else use toggleProv(trib)
                                    }
                            }
                    }
            }
    }
    wait 250msec
}

# Build build trib address from trib information.
getTrib (tug3, tug2, tu) {
    tribAddr = ((tug3 && 0x03) << 5) | ((tu && 0x03) << 3) |(tug2 && 0x07)
}

# Checks if the trib's PROV bit is set
PROVBitSet (Trib){
    tribCfg = read (0x1580 + Trib)
    return (tribCfg & 0x20)
}

# Checks a tributary's pointer values 1000 times
# Pointers difference should not be less than 1
checkTributary (trib fifoSize) {
    for {set iter 0} {iter <= 1000} {incr iter} {
            pointerDifference = checkPointers (trib fifoSize)
            if {pointerDifference <= 1 || pointerDifference >= fifoSize - 2} {
                    return "BAD_POINTER"
            }
    }
    return "POINTERS_OKAY"
}

check_pointers {trib fifoSize}
    write (0x1580 + 0x67) trib; #Set up RAM Capture Address
    write (0x1580 + 0x68) 0xC0; #Enable ctrl to capture RAM Vectors
    wait at least 6microseconds
    write (0x1580 + 0x68) 0x40; #Disable ctrl to stop RAM Vector capture
    readPointer = getReadPointer (trib)
    writePointer = getWritePointer (trib)
    pointerDifference = writePointer - readPointer
    if {pointerDifference < 0} {
            pointerDifference = pointerDifference + fifoSize
    }
    return pointerDifference
}

getWritePointer {fifoSize} {
    # Write pointer overlaps byte boundary into readPointer byte
    readPointer = {read (0x1580 + 0x73)}     # Read portion of RAM Vector that contains Write Ptr
    writePointer = {read (0x1580 + 0x72)}    # Read portion of RAM Vector that contains Write Ptr
    readPointer = (readPointer << 3) & 0x08
    writePointer = (writePointer >> 5) & 0x07
    writePointer = readPointer | writePointer  # Make write pointer from two portions
    writePointer = (writePointer)mod fifoSize
    return writePointer
}
```

```
getReadPointer {} {
    readPointer = {read (0x1580 + 0x73)}     # Read portion of RAM Vector that contains Read Ptr
    readPointer = (readPointer >> 1) & 0x0F # Adjust read pointer data
    return readPointer
}
#If T1/E1 Transmit Clock = Line Rate use centerTJAT(trib)
centerTJAT(trib) {
write TEMAP 0017+80*N 20 // TJAT Configuration register: CENT=LIMIT=SYNC=0, RESERVED=1
write TEMAP 0015+80*N 2F // TJAT N1
write TEMAP 0016+80*N 2F // TJAT N2
write TEMAP 0016+80*N "A" // TJAT N2, see Table 2
wait 100ms
write TEMAP 0015+80*N "B" // TJAT N1, see Table 2
wait 100ms
write TEMAP 0015+80*N "C" // TJAT N1, see Table 2
write TEMAP 0017+80*N 22 // TJAT Configuration register: CENT=LIMIT=0, RESERVED=SYNC=1
write TEMAP 0015+80*N "C" // TJAT N1, see Table 2
write TEMAP 0017+80*N 20 // TJAT Configuration register: CENT=LIMIT=SYNC=0, RESERVED=1
}
```

Table 2: N1 and N2 values for TJAT centering procedure

T1/E1 Transmit Clock (kHz)	T1			E1		
	A	B	C	A	B	C
1544	0xff	0x70	0xff	N/A	N/A	N/A
2048	N/A	N/A	N/A	0xff	0xa0	0xff

```
#If T1/E1 transmit timing clock ≠ Line Rate use toggleProv(trib)

toggleProv(trib)
    tribCfg = read (0x1580 + trib)
    write (0x1580 + trib) (tribCfg &0xDF)
    wait 1 usec
    write (0x1580 + trib) tribCfg
}
```

Appendix B: SPE Configuration to Prevent VT-AIS Data Corruption

When the TEMAP is setup to receive SPE #3(LDROPSEL[1:0] of register 1200H=11) of a given STS3 you must use the RTDM's time switch RAM to look at the copy of SPE#3 that exists on SPE#2. Otherwise you will see data corruption in the tributary adjacent to the tributary receiving TU-AIS. The internal RAM makes a copy of the data in SPE3# in all three TUG3s in this configuration so to prevent this problem from occurring, the RTDM Time Switch should be setup to switch the copy of SPE#3's data in TUG3 #2 to TUG3 #3.

The following is a script to switch TU #1 of TUG2 #1 of TUG3 #2 to TU#1 of TUG2 #1 of TUG3 #3. It should be repeated for all 28 VT1.5/TU11s for T1 mode or all 21 VT2/TU12s for E1 mode.

a) Register 12E2H RTDM Time Switch Page Control

APAGE	Set to 0 when configuring page 1 Set to 1 when configuring page 0

b) Register 12A0H RTDM TU#1 in TUG2 #1 of TUG3#2, Control

PROV	Set to 1

c) Register 12C0H RTDM TU#1 in TUG2 #1 of TUG3#3, Control

PROV	Set to 1

d) Register 12E4H RTDM Indirect Time Switch Internal Link Address

INT_SPE[1:0]	Set to '11'
INT_LINK[4:0]	Set to '00001'

e) Register 12E5H RTDM Indirect Ingress Tributary Data

ING_TUG3[1:0]	Set to '10'
INT_TUG2[2:0]	Set to '001'
ING_TU[2:0]	Set to '001'

f) Register 12E3H RTDM Indirect Time Switch Tributary RAM Status and Control

RWB	Set to 0
PAGE	Set to 0 if APAGE of 12E2H was set to 1 Set to 1 if APAGE of 12E2H was set to 0

g) Repeat a) through f) for the other 27 VT1.5/TU11s or 20 VT2/TU12s.

h) Register 12E2H RTDM Time Switch Page Control

APAGE	Set to 0 if PAGE 0 was just configured Set to 1 if PAGE 1 was just configured

i) Register 1201H SONET/SDH Master Ingress Configuration

ITSEN	Set to 1

Notes