PM73122

AAL1GATOR-32

REFERENCE DESIGN

PRELIMINARY INFORMATION

ISSUE 4: OCTOBER 2001

PUBLIC REVISION HISTORY

Issue No.	Issue Date	Details of Change
1	January 2000	Document created.
2	November 2000	AAL1gator SRAMs changed in schematics due to part availability. Pull-downs added to SOC and CA signals of UTOPIA buses. Clarifications made to document.
3	January 2001	Byte Write Enable lines corrected (swapped) on AAL1gator-32 SRAMs in schematics.
4	October 2001	Added appendix D for DS3 Adaptive Clock Recovery VHDL source code, and changed document status from preliminary to released, to match production release of AAL1gator-32 device.

PMC-Sierra, Inc.

CONTENTS

PMC-Sierra, Inc.

LIST OF FIGURES

1 INTRODUCTION

The AAL1gator-32 Circuit Emulation Service (CES) Reference Design provides a SONET/SDH network with T1, E1, E3 or DS3 access to an ATM Adaptation Layer One (AAL1) Constant Bit Rate (CBR) ATM network.

The AAL1gator-32 Reference Design is based on the 6U compact PCI standard. However, this reference design is a paper reference design only and has not been built or tested.

1.1 Purpose

This document provides a detailed hardware specification for the AAL1gator-32 Reference Design. This specification is detailed enough to allow design implementation and verification.

1.2 Scope

The purpose of this reference design is to assist engineers in designing their products using PMC-Sierra's PM73122 AAL1gator-32 and PM8315 TEMUX devices. A block diagram is shown for the design. A description is then given for the functional blocks of the design. A detailed implementation description then follows.

1.3 Applications

- DACS with an ATM interface.
- Any Service, Any Port Application.
- ATM Access Multiplexers.
- Part of a TDM Digital Access Cross-connect Systems (DACS) Replacement.
- High density T1/E1 interfaces for multiplexers, multi-service switches, routers and digital modems.
- SONET/SDH Add Drop Multiplexers.
- SONET/SDH Terminal Multiplexers.

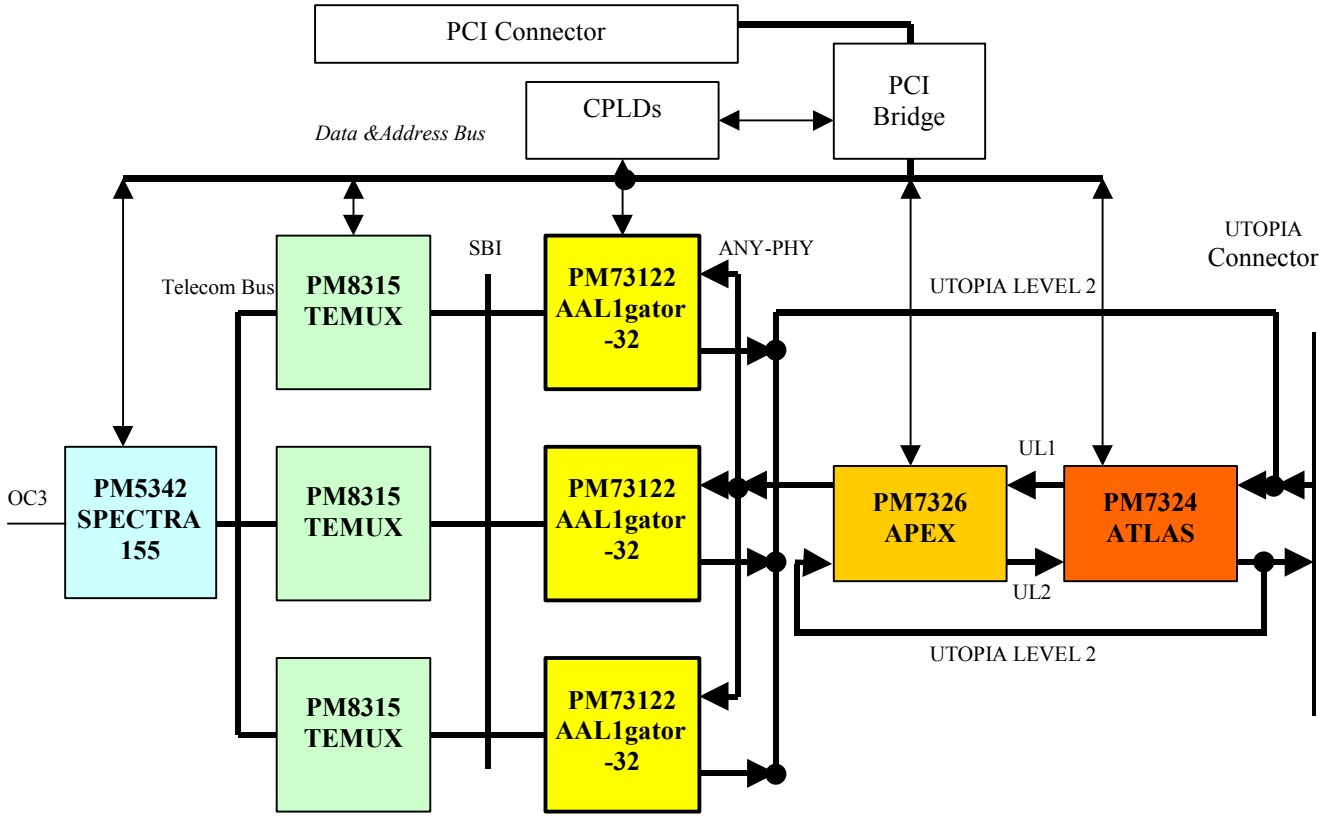
2 FEATURES

- Supports OC-3 capacity, scaleable to OC-12.
- Supports T1, E1, E3, and DS3 rates in channelized and unchannelized mode.
- Supports ATM and CES services.
- Telecom 8 bit Add/Drop TDM bus connects VT/TU channelized SONET/SDH processor (SPECTRA-155) to the high density framer (TEMUX) to support T1/E1 mapped payloads.
- Microprocessor interface utilized by a PCI bridge and a PCI connector.

3 GENERAL DESCRIPTION

A block diagram of the design is shown in Figure 1 consisting of several blocks. The design is a single Compact PCI card using one PM5342 SPECTRA-155, three PM8315 TEMUXes, three PM73122 AAL1gator-32s, one PM7324 ATLAS, one PM7326 APEX, a PLX9054 PCI bridge, two CPLDs and a HFCT-5205 optical transceiver.

Figure 1 AAL1gator-32 Reference Design



A Hewlett Packard HFCT-5205 optical transceiver provides a SONET/SDH compliant link at a serial signal rate of 155.52 Mbit/s. The transceiver performs optical-to-electrical conversion, converting the OC-3 optical signal into an STS-3/STM-1 stream and vice versa. This transceiver communicates with the

PM5342 SPECTRA via a PECL interface. The SPECTRA extracts/aligns the SONET/SDH payload and acts as a Telecom Bus interface.

The TEMUX devices receive/transmit data through the Telecom Bus and the data is then passed through the T1/E1 framers. It is then formatted for transmission to the AAL1gator-32 devices for CBR/VBR servicing. The APEX and ATLAS perform the routing, switching, traffic policing and shaping of the cells. The APEX and the ATLAS are connected to the AAL1gator-32 and the UTOPIA connector such that all cells first pass through the ATLAS so that policing and OAM functions can be performed first.

The microprocessor interface, utilized by the PCI connector and bridge, configures, controls and monitors all the above devices.

The CPLD in the design is used to generate a frame pulse signal for the SPECTRA-155, TEMUX, and AAL1gator-32 devices and to generate the chip select signals for the PMC-Sierra devices on the board.

Power requirements of the board are +5V, +3.3V and 2.5V. The SPECTRA-155 requires 5V while the TEMUX, AAL1gator-32, and APEX devices require a +2.5V power supply. ATLAS requires +3.3V.

SPECTRA-155, TEMUX, AAL1gator-32, and S/UNI-ATLAS devices require de-multiplexed address and data bus for the microprocessor interfaces while S/UNI-APEX requires 32-bit multiplexed microprocessor bus. In order to provide maximum system implementation flexibility, a PCI bridge chip has been used.

The two 80-pin female UTOPIA connectors carry the receive and transmit UTOPIA signals between the S/UNI-ATLAS and any external PHY board or a Parallel Cell Traffic Generator and Analyzer such as HP E1401B UTOPIA Tester.

The design also includes several LED circuits for the device alarms and power indications.

4 BLOCK DESCRIPTIONS

4.1 SPECTRA-155

The SPECTRA-155 is a SONET/SDH payload extractor/aligner for use in STS-1, STS-3 or STS-3c interface applications, operating at serial interface speeds of up to 155 Mb/s.

In the receive direction, the SPECTRA-155 receives SONET/SDH frames via a bit serial interface, recovers clock and data, and terminates the SONET/SDH regenerator section, multiplexer section and path overhead. It performs framing, descrambling, detects alarm conditions, and monitors section and line bit interleaved parity. In addition, the SPECTRA-155 interprets the received payload pointers and extracts the Synchronous Payload Envelope (SPE).

The SPE extracted by SPECTRA-155 is placed on a Telecom DROP bus. The SPECTRA-155 maps the three DS3s from the STS-3 SPE and provides serialized bit streams with derived clocks.

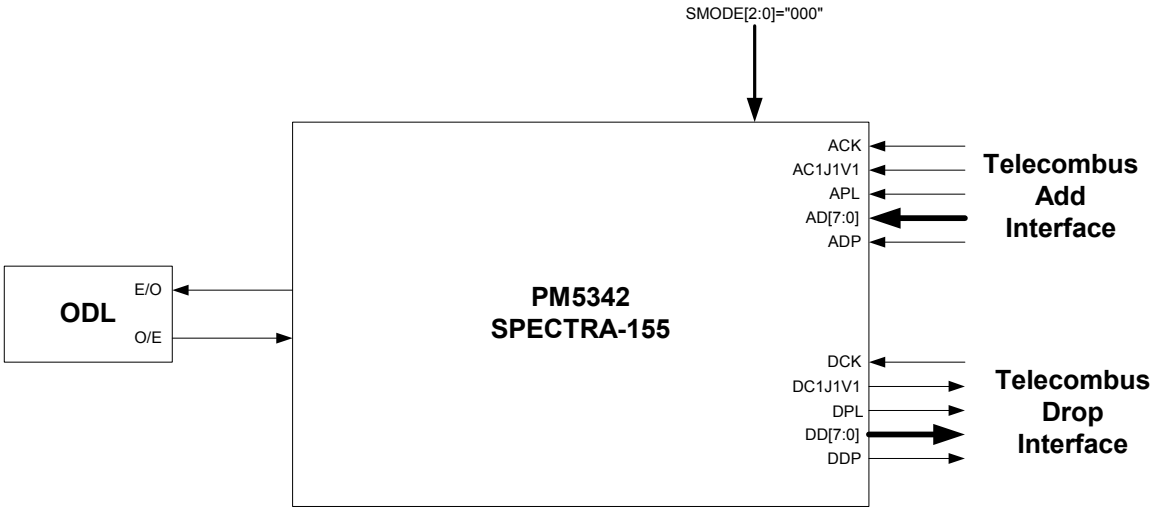
In the transmit direction, the SPECTRA-155 transmits SONET/SDH frames, via a bit serial interface, and formats the section, line and path overheads. It performs framing pattern insertion, scrambling, alarm signal insertion, and creates the section and line bit interleaved parity. In addition, the SPECTRA-155 generates the transmit payload pointers.

The inserted SPE is either sourced from a Telecom ADD bus stream, from DS3 serial streams, or from data streams. For the Telecom bus applications, the SPECTRA-155 maps the SPE from Telecom ADD bus into the transmit stream. Figure 2 shows the direct interface of the SPECTRA-155 to a Telecom bus.

The SPECTRA-155 uses analog power pins QAVD, RAVD, and TAVD, which must be applied after VDD. A simple filtering network is placed on each of the analog pins to delay the voltage rise until the digital pins are at the proper voltage.

For more information about the SPECTRA-155, please refer to [1].

Figure 2 SPECTRA-155 STS-3 Configuration



4.2 TEMUX

TEMUX is an integrated circuit which integrates 28 T1 framers, 21 E1 framers, SONET/SDH VT1.5/V2/TU-11/TU-12 bit asynchronous mapper and full featured M13 multiplexer with DS3 framer. It also contains a SONET/SDH DS3 mapper for terminating DS3 multiplexed T1 streams, SONET/SDH mapped T1 streams or SONET/SDH mapped E1 streams. Virtual Tributary VT1.5 carries enough bandwidth to transport a DS1 signal of 28 DS0s at 64 Kb/s. Analogously, VT2 carries enough bandwidth to transport an E1 (2.048 Mb/s) signal. The M13 multiplexing process involves the combination of 28 DS1 (1.544 Mb/s) signals into a single DS3 signal (44.736 Mb/s). The DS3 is asynchronously mapped into a STS-1 SPE. Three STS-1 signals form an STS-3.

The device supports a byte serial Scalable Bandwidth Interconnect (SBI) bus interface for high density system side device interconnection of up to 84 T1 streams, 63 E1 streams, 3 DS3 streams or 3 E3 streams. On the line side TEMUX supports SONET/SDH Telecom bus and provides an 8-bit microprocessor bus interface for configuration, control, and status monitoring.

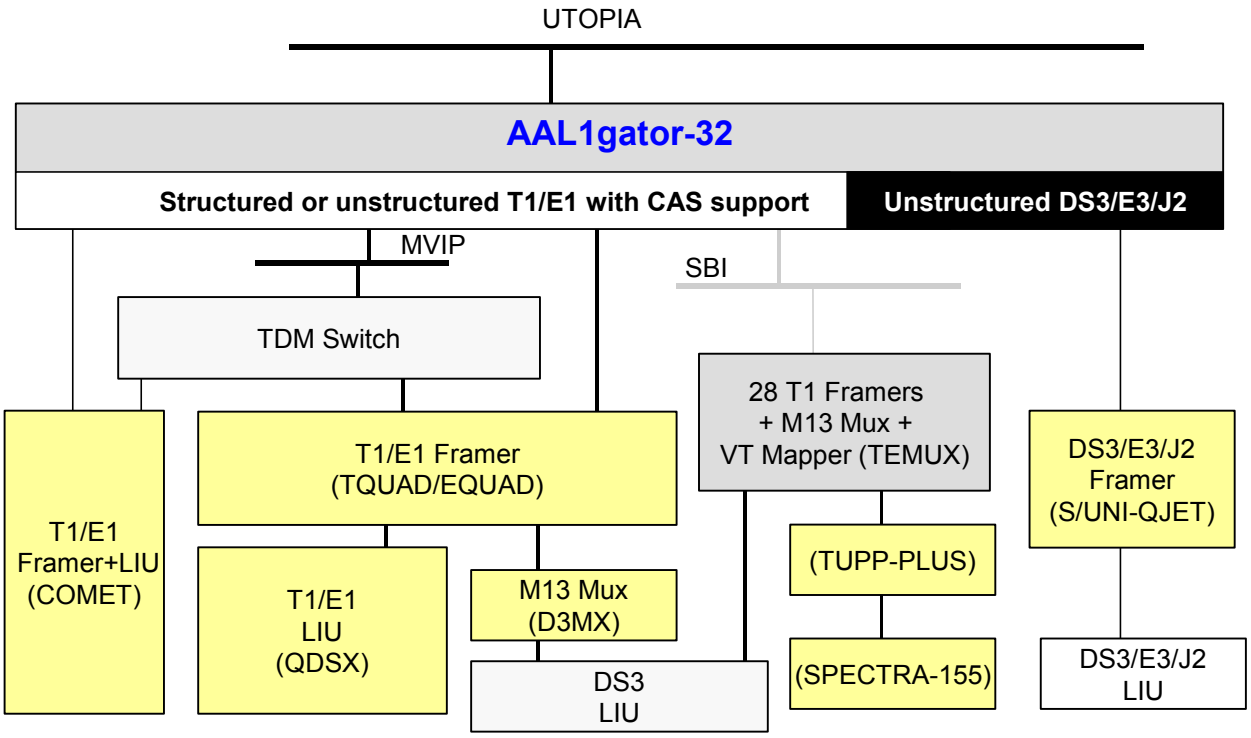
For more information about the TEMUX, please refer to [2] of the references.

4.3 AAL1gator-32

The purpose of the AAL1gator-32 is to provide high density T1/E1, or DS3/E3/J2 line interfaces access to an AAL1 CBR ATM network. The AAL1gator-32 can support 32 links via PMC-Sierra's SBI bus or eight 8Mb/s H-MVIP links. The

AAL1gator-32 is capable of supporting 1024 VCs. Figure 3 indicates the ways in which an AAL1gator-32 can be used to connect to T1/E1 or DS3/E3/J2 line interfaces.

Figure 3 AAL1gator-32 Configurations



In this design each AAL1gator-32 interfaces with a PM8315 TEMUX via the SBI bus to support:

- 28 structured/unstructured T1s if only 1 TEMUX is used, 32 if more than 1 TEMUX is used
- 21 structured/unstructured E1s if only 1 TEMUX is used, 32 if more than 1 TEMUX is used

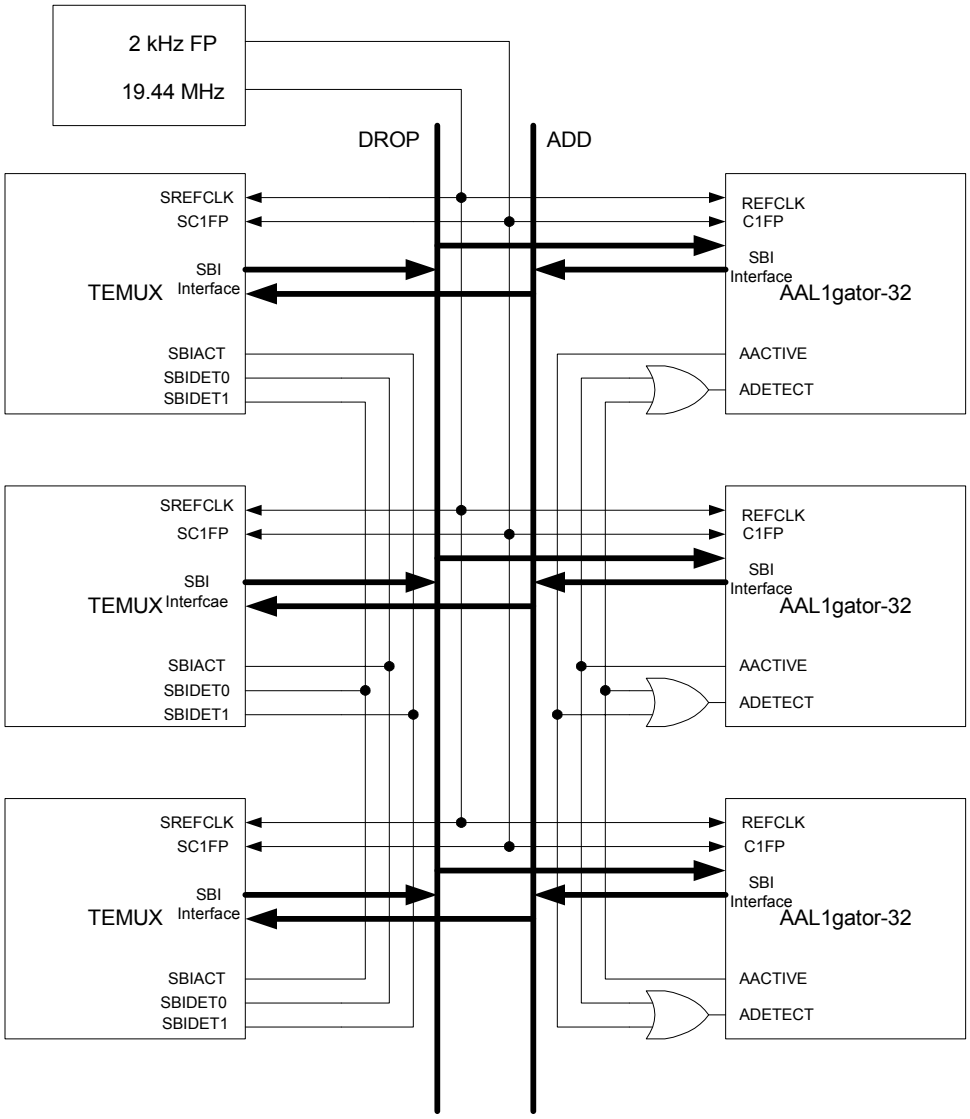
When used with the TEMUX, the AAL1gator-32 can be a part of a multiservice switch application which can provide circuit emulation services on E1 or T1 pipes being carried over a DS3 or OC-3 link. It can also provide circuit emulation services for an unstructured DS3 link. This system is scalable to an OC-12 system, using SPECTRA-622 instead of the SPECTRA-155 and more TEMUX and AAL1gator-32 devices.

The AAL1 Segmentation and Reassembly (SAR) Processor (AAL1gator-32) is a monolithic single chip device that provides DS1, E1, E3, or DS3 line interface access to an ATM Adaptation Layer One (AAL1) Constant Bit Rate (CBR) ATM network. It arbitrates access to an external SRAM for storage of the configuration, the user data, and the statistics. Some of the device's important functionality is as follows:

- Compliant with the ATM Forum's Circuit Emulation Services (CES) specification (AF-VTOA-0078), and the ITU-T I.363.1
- Supports Dynamic Bandwidth Circuit Emulation Services (DBCES). Compliant with the ATM Forum's DBCES specification (AF-VTOA-0085).
- Supports idle channel detection via processor intervention, CAS signaling, or data pattern detection.
- Provides idle channel indication on a per channel basis.
- Provides AAL1 segmentation and reassembly of 16 individual E1 or T1 lines in the direct low speed mode, 8 H-MVIP lines at 8 Mb/s in the H-MVIP mode, or 2 E3 or DS3 lines in the high speed mode.
- Using the Scalable Bandwidth Interconnect (SBI) Interface, provides AAL1 segmentation and reassembly of up to 32 T1, E1, VT1.5, or VT2 links, or 2 DS3 or E3 links.
- Provides a standard UTOPIA level 2 Interface which optionally supports parity and runs up to 52 MHz. The following modes are supported:
    - 16-bit Level 2, Multi-Phy Mode (MPHY)
    - 8-bit Level 2, MPHY
    - 8-bit Level 1, ATM Master
- Supports up to 1024 Virtual Channels (VC).
- Supports n x 64 (consecutive channels) and m x 64 (non-consecutive channels) structured data format.

Each AAL1gator-32 interfaces with a TEMUX through PMC-Sierra's SBI bus. The high level design of this interface is shown in Figure 4.

Figure 4 TEMUX- AAL1gator-32 SBI Interface



This reference design uses zero bus turnaround (ZBT) synchronous SRAM for the AAL1gator-32 so that the highest degree of performance is available. ZBT SSRAM should be used in situations where the AAL1gator-32 is configured for a large number of partial cell VCs or a large number of VCs are being used over all channels. For normal operation, pipelined single-cycle deselect SSRAM (non-ZBT) will suffice. Note that the connections between the AAL1gator-32 and ZBT or non-ZBT SSRAM are slightly different. Refer to the AAL1gator-32 datasheet and the datasheet of the SSRAM being used for details.

PMC-Sierra does not recommend the use of one specific manufacturer's SSRAM. A list of possible solutions is provided below; however, this list should not be considered exhaustive.

Pipelined Single-Cycle Deselect SSRAM (non-ZBT)

Motorola MCM69P819-4 (TQ4)
Galvantech (Cypress) GVT71256G18-6
Micron MT58L25618P-10
GSI Technology GS84018T/B-100

Zero Bus Turnaround (ZBT) SSRAM

Cypress CY71352
Micron MT55L256L18P-10
Motorola MCM63Z818-100 (TQ133)

The AAL1gator-32 is capable of internally synthesizing an E1/T1 clock for each line using both the synchronous residual timestamp (SRTS) and adaptive clock recovery methods in unstructured data format (UDF-ML) mode. The AAL1gator-32 is not able to internally synthesize a DS3/E3 line rate clock; however, the AAL1gator-32 does output both SRTS and adaptive clock recovery information to the Clock Generation and Control (CGC) port to support external DS3/E3 clock synthesis for two DS3/E3 signals.

The CGC port of the AAL1gator-32 is connected to the CPLD so that a few different functions can be implement externally. These possible functions are:

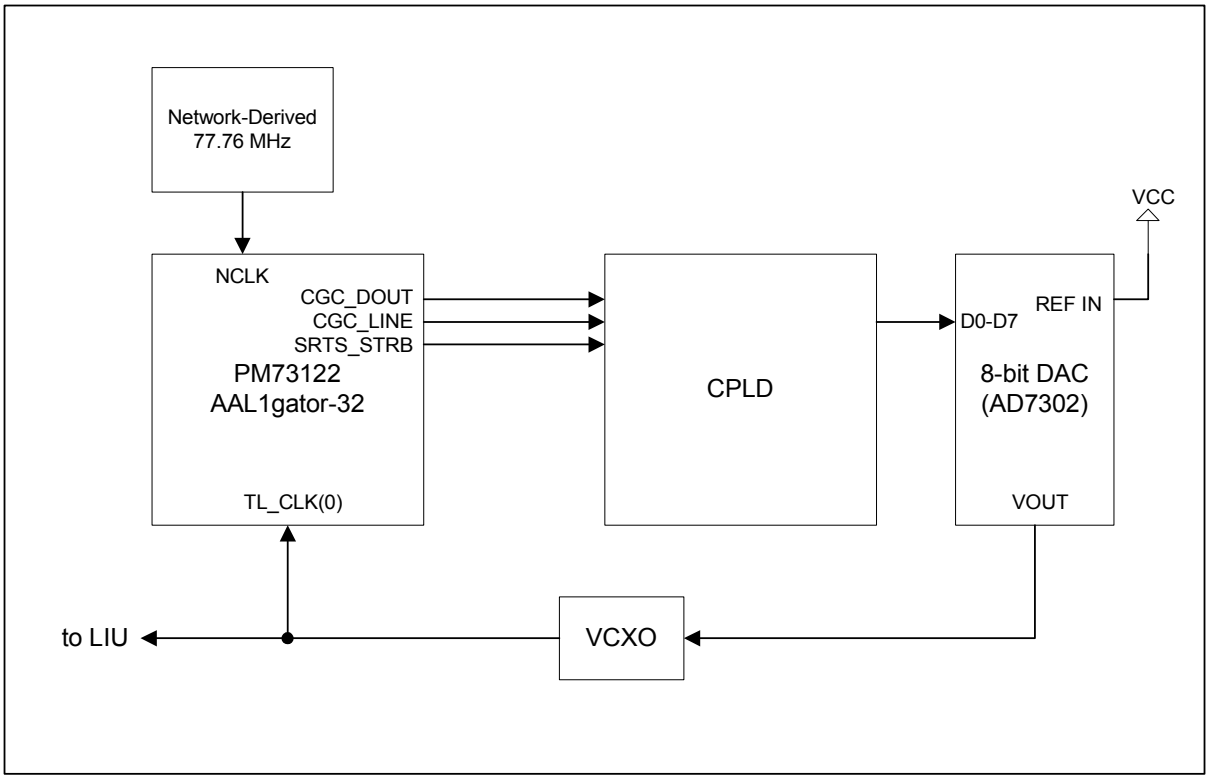
- External control of the AAL1gator-32 internal T1/E1 clock synthesizers to implement a custom SRTS algorithm.
- Generation of 32 external TL_CLK sources per AAL1gator-32 when in SBI mode.
- Read CGC nibbles for implementation of an external adaptive clocking algorithm.
- Read CGC nibbles for implementation of an external DS3/E3 SRTS clock recovery algorithm.
- Read CGC nibbles for implementation of an external DS3/E3 adaptive clock recovery algorithm.

As the last two points suggest, the high speed SRTS information from the AAL1gator-32's SRTS port can be used to synthesize and control a DS3 or E3 clock externally. The circuitry for external DS3/E3 SRTS is shown in Figure 5.

Some external algorithms use digital or analog filtering between certain stages of the circuitry; however, PMC-Sierra has found the circuitry of Figure 5 to synthesize and track the appropriate clock well and therefore PMC-Sierra does not recommend any other filtering.

External SRTS is not implemented within this reference design. However, the Analog Devices AD7302 8-bit DAC and the MMD VCXO MVA3025HACY (DS3: 44.736 MHz, E3: 34.368 MHz) have been used with success. Similar devices are also available from other manufactures. The VHDL code for the CPLD is given in Appendix C.

Figure 5 SRTS-based DS3/E3 Clock Recovery Circuit



This circuit functions as follows: the AAL1gator-32 asserts SRTS_STRB indicating a new SRTS nibble is available on CGC_DOUT for the line indicated by CGC_LINE. A lookup table within the CPLD is then used to convert the CGC_DOUT nibble into an 8-bit code to drive the digital-to-analog converter (DAC). The DAC output voltage then controls the voltage-controlled oscillator VCXO, which has a DS3 or E3 center frequency. The resulting clock is then fed to the AAL1gator-32 TL_CLK(0) input and the LIU. Note that NCLK must be network derived, but SYSCLK can be asynchronous to the network.

The circuit shown in Figure 5 above can also be used for DS3/E3 adaptive clock recovery with the following changes: the network-derived 77.76 MHz clock is not used, and CGC output ADAP_STBH should be connected to the CPLD. A DS3 Adaptive Clock Recovery CPLD implementation is given in Appendix D.

For a more detailed description of the AAL1gator-32, please refer to [3] of the references.

4.4 S/UNI-APEX

The PM7326 S/UNI-APEX is a full duplex ATM traffic management device, providing cell switching, per VC queuing, traffic shaping, congestion

management, and hierarchical scheduling to up to 2048 loop ports and up to 4 WAN ports.

The S/UNI APEX provides per-VC queuing for 64K VCs. A per-VC queue may be allocated to any Class of Service (COS), within any port, in either direction (ingress or egress path). Per-VC queuing enables PCR or SCR per-VC shaping on WAN ports and greater fairness of bandwidth allocation between VCs within a COS.

The APEX supports an 8/16-bit Any-PHY compliant loop side master/slave interface supporting up to 2048 ports. Egress cell transfers across the interface are identified via an inband port identifier prepended to the cell. The slave devices must match the inband port identifier with their own port ID or port ID range in order to accept the cell. Per port egress flow control is effected via a 12-bit address polling bus to which the appropriate slave device responds with out of band per port flow control status. Ingress cell transfers across the interface are effected via a combination of UTOPIA L2 flow control polling and device selection for up to 32 slave devices. The Any-PHY loop side interface may be reconfigured as a standard single port 16-bit Any-PHY or UTOPIA L2 compliant slave interface. 16-bit prepends are optionally supported on both ingress and egress for cell flow identification enabling use with external address resolution devices, switch fabric interfaces, or other layer devices.

The S/UNI APEX provides an 8/16-bit Any-PHY or UTOPIA L2 compliant WAN side master interface supporting up to 4 ports. The 16-bit prepends are optionally supported on both ingress and egress for cell flow identification enabling use with external address resolution devices, switch fabric interfaces, or other layer devices.

The S/UNI APEX provides a 36-bit SSRAM interface for context storage supporting up to 4MB of context for up to 64kVCs and up to 256k cell buffer pointer storage. Context Memory protection is provided via 2 bits of parity over each 34-bit word.

The S/UNI-APEX should be used if functions such as VC-based switching or traffic management are required within a design. Traffic management includes congestion management, class of service aware scheduling, and shaping.

For more information about the S/UNI-APEX, please refer to [4] of the references.

4.5 S/UNI-ATLAS

The PM7324 S/UNI-ATM Layer Solution (S/UNI-ATLAS, or simply ATLAS) is a PMC-Sierra standard product that implements the following ATM Layer functions:

- OAM processing according to ITU-T I.610 1995 and 1998 living list.
- Header Translation on full VPI/VCI address range.
- Prepend/Postpend tagging.
- Cell rate policing according to ITU-T I.371 using the Generic Cell Rate Algorithm.
- GFR Policing according to ATM Forum's Traffic Management 4.0 1998 living list.
- Per-PHY queuing to prevent head-of-line blocking.

The ATLAS performs both ingress and egress functionality. The ingress side has a SCI-PHY level 2 interface at the input, and a SCI-PHY level 1 interface at the output. Cells coming into the ATLAS from a PHY are identified according to the PHY ID, VPI, and VCI. The cells are processed according to the information stored in context RAM for the particular connection. Cells may also be copied to the microprocessor cell interface for external processing. The egress has a SCI-PHY level 2 interface at both the input and output interface. The connection is identified according to the PHY ID, VPI, and VCI, and processed according to the information in external RAM for the particular connection. As with the ingress, cells can be copied to the microprocessor cell interface for external processing. The ATLAS is configured and controlled through a generic 16 bit asynchronous microprocessor bus.

The S/UNI-ATLAS should be used in applications where policing or I.610 OAM compliance are required, or when traffic is destined for an ATM WAN or switch. Policing of ingress traffic is performed to ensure that the actual traffic pattern fits within contract (possibly for DBCES applications). Per VC counters are also available within the S/UNI-ATLAS for monitoring bandwidth.

The S/UNI-ATLAS is utilized such that all ingress cells pass first through the S/UNI-ATLAS, then through the S/UNI-APEX, and then back out through the S/UNI-ATLAS again. This setup allows ingress packets to be policed at the ingress to the network, and for OAM cells to be looped back for I.610 compliance. For more detail on traffic management and switching, please refer to [5] of the references.

For a more detailed description of the S/UNI-ATLAS, please refer to [6] of the references.

4.6 Microprocessor Interface and CPLD

As illustrated in Figure 6, the microprocessor interface to the card is implemented using the CompactPCI standards with a cPCI connector, a PCI 9054 chip and a CPLD. This interface provides an external host CPU connection to perform the following functions on the AAL1gator-32 Reference Design:

- Configuration of the SPECTRA-155, TEMUX, AAL1gator-32, S/UNI-APEX, and S/UNI-ATLAS devices
- Setting up connections in the AAL1gator-32, S/UNI-APEX, and S/UNI-ATLAS context RAMs
- Monitoring of alarms and interrupts in the SPECTRA-155, TEMUX, AAL1gator-32, S/UNI-APEX, and S/UNI-ATLAS devices
- Background Debug mode for the board's feature tests.

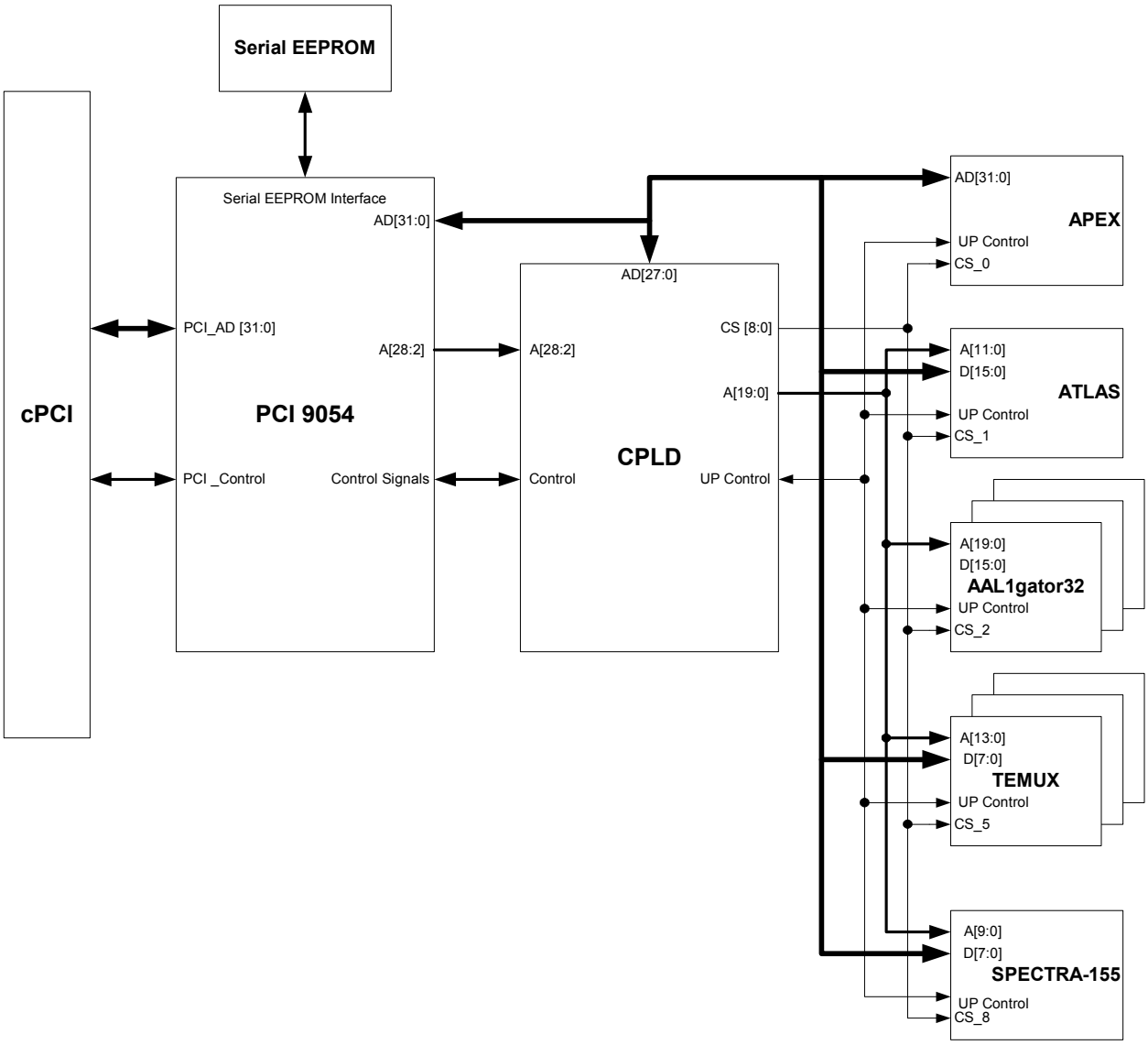
Note that a second CPLD is included in the design for clocking; however, VHDL code is not included for this CPLD for external clocking options because the AAL1gator-32 is capable of performing these functions internally. Only users wishing to implement their own algorithm externally need to implement this CPLD.

The essential characteristic of the PCI concept is the strict decoupling of an external processor's main memory subsystems and the standard expansion bus by means of a bridge. The PLX's PCI 9054 bridge is the PCI v2.2 compliant 32-bit, 33 MHz bus master interface controller which enables PCI burst transfers up to 132 Mbytes per second.

Since the S/UNI-APEX requires a multiplexed address/data bus, the PCI 9054 is configured for the multiplexed bus mode (J mode) on its local bus side. As shown in Figure 6, the 32-bit multiplexed AD[31:0] Address/Data bus from the PCI 9054 is directly connected to the APEX address/data pins. The remaining devices, by means of their Address Latch Enable (ALE) pins, have the microprocessor interface option to work with the multiplexed address/data bus. The ALE pin latches the address signals during the address phase of the bus transaction. Therefore, except for the APEX, the AD[31:0] is connected to the data port of the other devices. The proper resistive termination along the bus line is necessary to help with signal integrity.

The additional A[28:2] address bus is connected to the other devices which need separate address lines through the CPLD. Also, the CPLD is used to generate the chip select and other microprocessor control signals for the PMC devices.

Figure 6 Microprocessor Interface Block



4.7 Power Supply

Power is provided to the AAL1gator-32 Reference Design through the Compact PCI interface. PMC-Sierra's devices require that the +3.3V power rail be higher voltage than the +2.5V power rail at all times. This is achieved by regulating the lower voltage from the higher voltage, as shown on page 29 of the schematics in the appendix. +5.0V and +3.3V are taken from the Compact PCI interface. +2.5V is then regulated using the LT1580 regulator. This regulator is designed to have a low dropout voltage, and therefore requires +5.0V as well. However, most of the current is drawn from the +3.3V rail. A zener diode is added to the +5.0V rail to

dampen any voltage spikes and LEDs are connected across each of the supply rails to indicate the presence of voltage.

5 TESTABILITY

Figure 7 illustrates a simple test bench connection for the AAL1gator-32 Reference Design Card. The required tester/analyzer for testing the reference design card must be capable of embedding ATM cells inside T1 or E1 frames, inside a DS3 or E3 bit stream, and inside a Synchronous Payload Envelope (SPE). The HP 37717C communications performance analyzer equipped with the optical interfaces, the optical adaptor, the SONET/SDH test and interfaces, the PDH/DSn and ATM test and interfaces, and the ATM services layer test options may be used for this purpose. The analyzer sources the STS-3 data in to the receiver and receives the returned data from the transmitter part of the optical data link module of the reference design card.

Figure 7 Simple System Test Bench Block Diagram

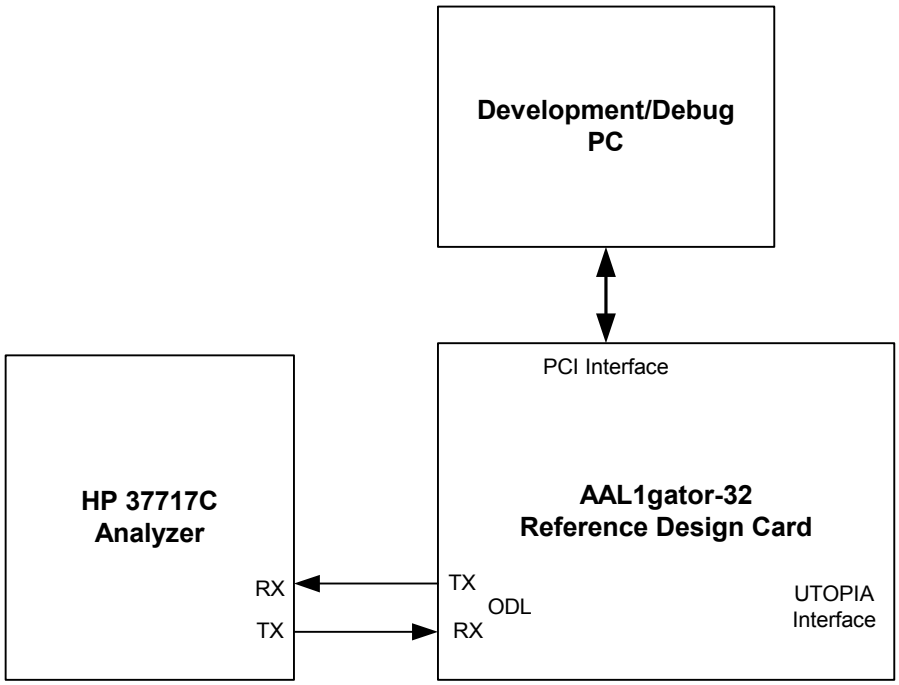
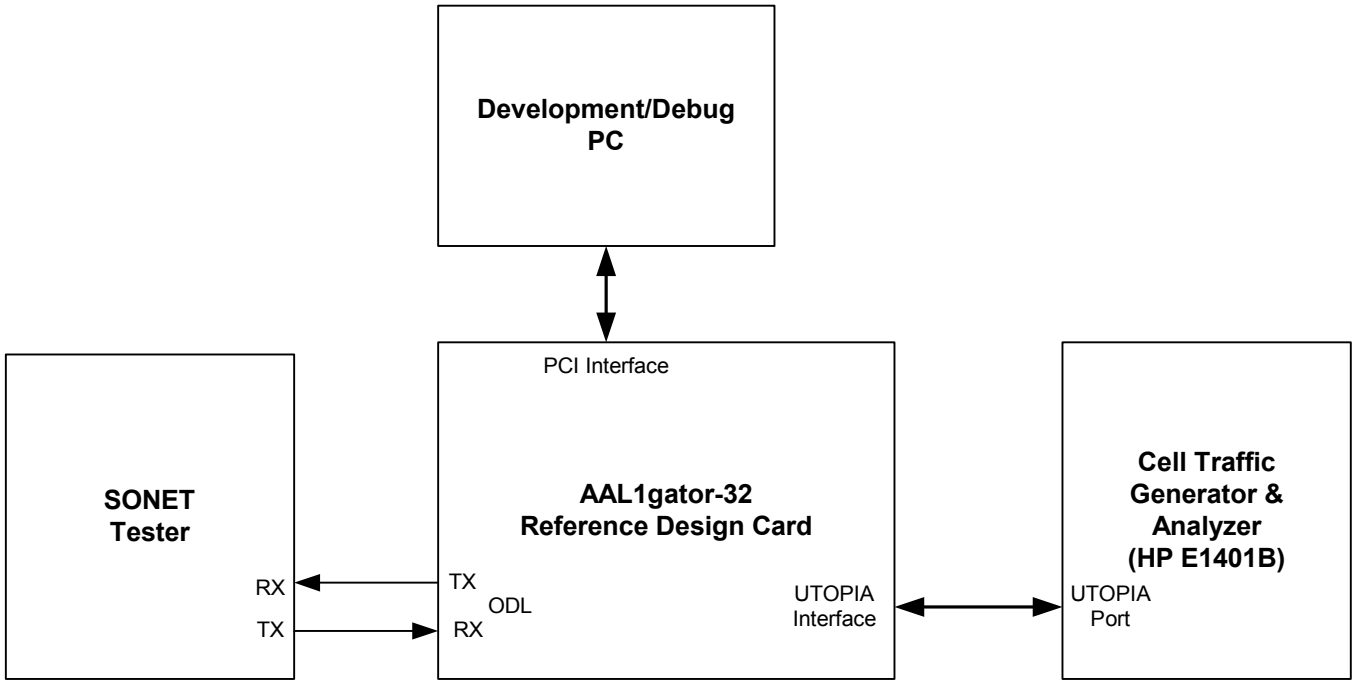


Figure 8 illustrates a complex test bench connection for the AAL1gator-32 Reference Design Card. The optical transceiver is attached to external SONET test equipment which sources OC3 data into the receiver and receives the returned data from the transmitter part of the optical data link module of the card. A Cell Traffic Generator and Analyzer such as HP E14011B receives/transmits and monitors the ATM cell traffic from/to the reference design card through the UTOPIA connector on the board.

Figure 8 Complex System Test Bench Block Diagram



The diagnostic loopback mode of the devices can be used to loopback the transmit data to the receive path.

The cPCI port of the Reference Design Card allows the microprocessor interface with a host CPU (the development/debug PC) to monitor the status of SPECTRA-155, TEMUX, AAL1gator-32, S/UNI-APEX, and S/UNI-ATLAS devices. This port is also used to program registers for initialization, software load and reset, and diagnostics.

6 IMPLEMENTATION DESCRIPTION

The schematic contains 29 pages as follows:

Sheet 1: Root Drawing

This sheet provides a block view of the interface signals between each block of the AAL1gator-32 Reference Design.

Sheet 2: Optical Interface

This sheet shows the connections needed for the HP HFCT5205 optical transceiver.

Sheet 3: SPECTRA-155

This sheet contains the PM5342 SPECTRA-155, alarm LEDs, and other supporting circuitry.

Sheet 4-9: Temuxes Block

These sheets contain the 3 PM8315 TEMUXes and their connections.

Sheet 10-18: AAL1gator-32 Block

These sheets contain the 3 PM73122 AAL1gator-32s including their synchronous SRAMs.

Sheet 19-21: Apex Block

These sheets contain the PM7326 APEX and the requied synchronous DRAM and SRAM.

Sheet 22-23: Atlas Block

These sheets contain the PM7324 ATLAS and the the required synchronous SRAM.

Sheet 24: UTOPIA Connector

This sheet contains the transmit and receive UTOPIA Level 2 connector.

Sheet 25: 25 MHz Oscillator Block

This sheet contains the 25 MHz crystal oscillator for the UTOPIA buses.

Sheet 26-28: CPLD Block

These sheets contain the two CPLDs that are used to generate frame pulses and distribute clock signals.

Sheet 29: Power and Reset Block

This sheet contains the voltage regulators for powering the devices.

7 GLOSSARY

AAL1	ATM Adaptation Layer 1
ASAP	Any Service-Any Port
ATM	Asynchronous Transfer Mode
CBR	Constant Bit Rate
CES	Circuit Emulation Services
COS	Class of Service
PCI	Peripheral Component Interconnect
SAR	Segmentation and Re-assembly
SCI-PHY	PMC-Sierra enhanced UTOPIA bus
SBI	Scalable Bandwidth Interconnect
SRTS	Synchronous Residual Time Stamp
UTOPIA	Universal Test & Operations PHY Interface for ATM
VBR	Variable Bit Rate
VC	Virtual Circuit
VCC	Virtual Channel Connection
VCI	Virtual Circuit Identifier
VP	Virtual Path
VPC	Virtual Path Connection
VPI	Virtual Path Identifier
WAN	Wide Area Network
ZBT	Zero Bus Turnaround

8 REFERENCES

1. PMC-Sierra Inc., PMC-970133, “SONET/SDH Payload Extractor/Aligner for 155 Mbits/s Telecom Standard Product Data Sheet”, August 1998, Issue 4.

2. PMC-Sierra Inc., PMC-981125, “High Density T1/E1 Framer With Integrated VT/TU Mapper and M13 Multiplexer Telecom Standard Product Data Sheet”, June 1998, Issue 4.

3. PMC-Sierra Inc., PMC-981419, “ATM Adaptation Layer 1 Segmentation and Reassembly Processor-32 (AAL1gator-32) Telecom Standard Product Data Sheet”, December 1998, Issue 1.

4. PMC-Sierra Inc., PMC-981224, ”ATM/Packet Traffic Manager and Switch (S/UNI-APEX) Data Sheet”, Issue 3.

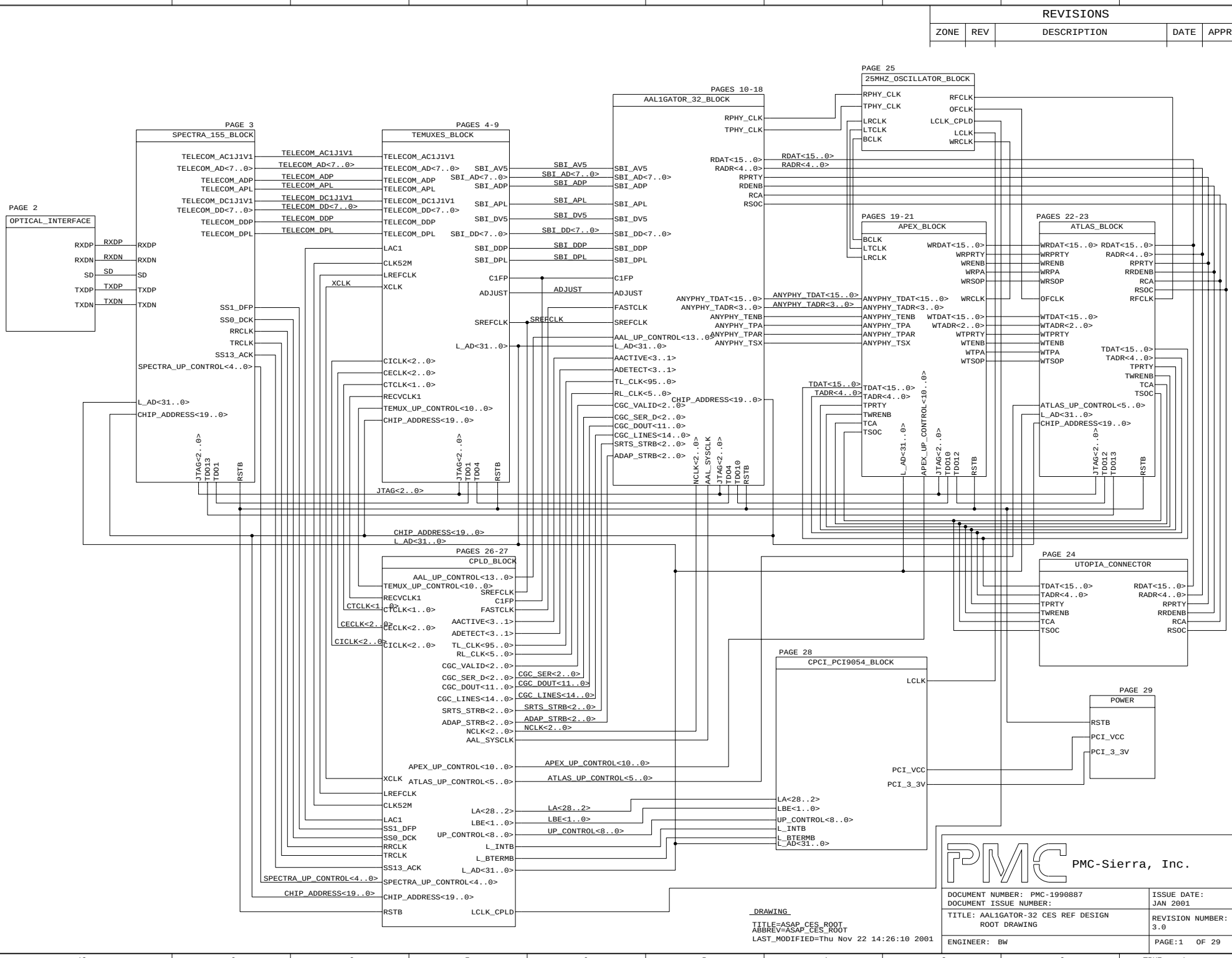
5. PMC-Sierra, Inc., PMC-1981024, “Traffic Management and Switching Using the Vortex Chipset: S/UNI-APEX Technical Overview”, August 30, 1999, Issue 1.0.

6. PMC-Sierra Inc., PMC-971154, ”S/UNI-ATM Layer Solution Data Sheet”, January 1999, Issue 5.

9 DISCLAIMER

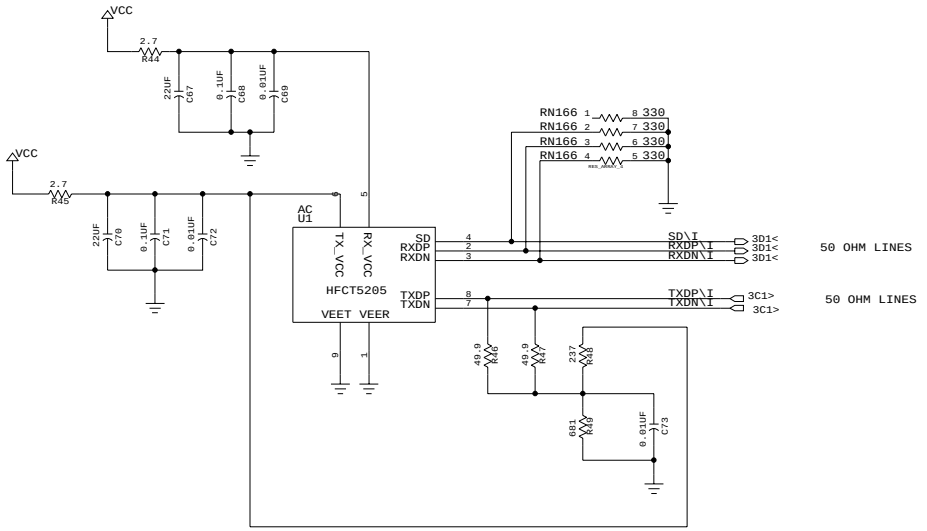
This document is a paper reference design and, as such, has not been built or tested.

10 APPENDIX A: AAL1GATOR-32 REFERENCE DESIGN SCHEMATICS


REVISIONS
ZONE
REV
DESCRIPTION
DATE
APPR
PAGE 2
OPTICAL_INTERFACE
RXDP
RXDN
SD
TXDP
TXDN
PAGE 3
SPECTRA_155_BLOCK
TELECOM_AC1J1V1
TELECOM_AD<7..0>
TELECOM_ADP
TELECOM_APL
TELECOM_DC1J1V1
TELECOM_DD<7..0>
TELECOM_DDP
TELECOM_DPL
LAC1
CLK52M
LREFCLK
XCLK
SS1_DFP
SS0_DCK
RRCLK
TRCLK
SS13_ACK
SPECTRA_UP_CONTROL<4..0>
L_AD<31..0>
CHIP_ADDRESS<19..0>
JTAG<2..0>
TDO13
TDO1
RSTB
PAGES 4-9
TEMUXES_BLOCK
SBI_AV5
SBI_AD<7..0>
SBI_ADP
SBI_APL
SBI_DV5
SBI_DD<7..0>
SBI_DDP
SBI_DPL
C1FP
ADJUST
SREFCLK
CICLK<2..0>
CECLK<2..0>
CTCLK<1..0>
RECVCLK1
TEMUX_UP_CONTROL<10..0>
TDO1
TDO4
PAGES 10-18
AAL1GATOR_32_BLOCK
FASTCLK
AACTIVE<3..1>
ADETECT<3..1>
TL_CLK<95..0>
RL_CLK<5..0>
CHIP_ADDRESS<19..0>
CGC_VALID<2..0>
CGC_SER_D<2..0>
CGC_DOUT<11..0>
CGC_LINES<14..0>
SRTS_STRB<2..0>
ADAP_STRB<2..0>
NCLK<2..0>
AAL_SYSCLK
PAGE 25
25MHZ_OSCILLATOR_BLOCK
RPHY_CLK
TPHY_CLK
LRCLK
LTCLK
BCLK
RFCLK
OFCLK
LCLK_CPLD
LCLK
WRCLK
RDAT<15..0>
RADR<4..0>
RPRTY
RDENB
RCA
RSOC
PAGES 19-21
APEX_BLOCK
ANYPHY_TDAT<15..0>
ANYPHY_TADR<3..0>
ANYPHY_TENB
ANYPHY_TPA
ANYPHY_TPAR
ANYPHY_TSX
WRDAT<15..0>
WRPRTY
WRENB
WRPA
WRSOP
WTDAT<15..0>
WTADR<2..0>
WTPRTY
WTENB
WTPA
WTSOP
PAGES 22-23
ATLAS_BLOCK
WRDAT<15..0>
WRPRTY
WRENB
WRPA
WRSOP
OFCLK
WTDAT<15..0>
WTADR<2..0>
WTPRTY
WTENB
WTPA
WTSOP
RDAT<15..0>
RADR<4..0>
RPRTY
RRDENB
RCA
RSOC
RFCLK
TDAT<15..0>
TADR<4..0>
TPRTY
TWRENB
TCA
TSOC
ATLAS_UP_CONTROL<5..0>
L_AD<31..0>
CHIP_ADDRESS<19..0>
APEX_UP_CONTROL<10..0>
PAGES 26-27
CPLD_BLOCK
AAL_UP_CONTROL<13..0>
TEMUX_UP_CONTROL<10..0>
CGC_SER<2..0>
CGC_SER_D<2..0>
CGC_DOUT<11..0>
CGC_LINES<14..0>
SRTS_STRB<2..0>
ADAP_STRB<2..0>
NCLK<2..0>
AAL_SYSCLK
APEX_UP_CONTROL<10..0>
ATLAS_UP_CONTROL<5..0>
LA<28..2>
LBE<1..0>
UP_CONTROL<8..0>
L_INTB
L_BTERMB
L_AD<31..0>
LCLK_CPLD
PAGE 24
UTOPIA_CONNECTOR
PAGE 28
CPCI_PCI9054_BLOCK
LCLK
PAGE 29
POWER
RSTB
PCI_VCC
PCI_3_3V
PMC-Sierra, Inc.
DOCUMENT NUMBER: PMC-1990887
DOCUMENT ISSUE NUMBER:
ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 CES REF DESIGN ROOT DRAWING
REVISION NUMBER: 3.0
ENGINEER: BW
PAGE:1 OF 29
DRAWING
TITLE=ASAP_CES_ROOT
ABBREV=ASAP_CES_ROOT
LAST_MODIFIED=Thu Nov 22 14:26:10 2001

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR

NOTE: VCC IS +5V DC.

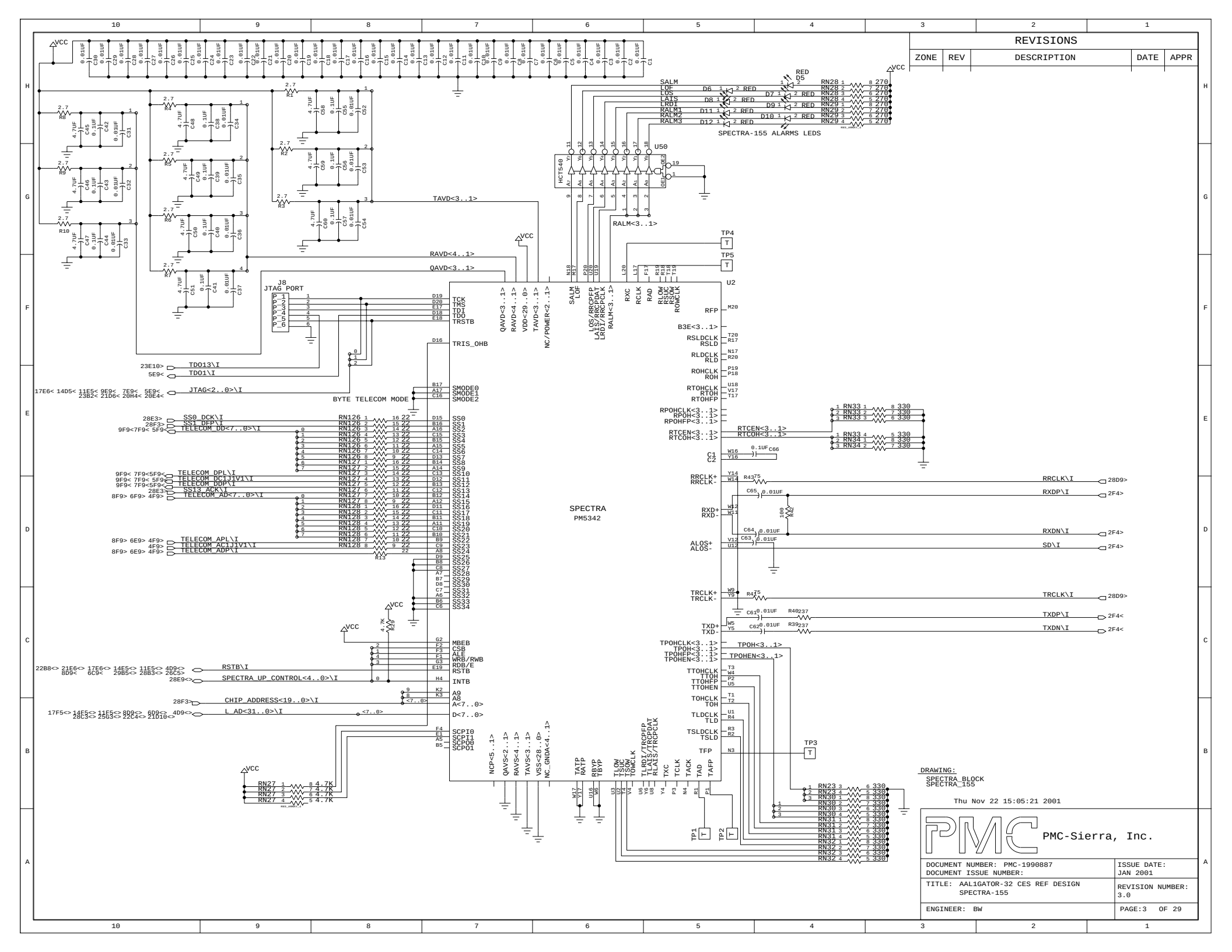


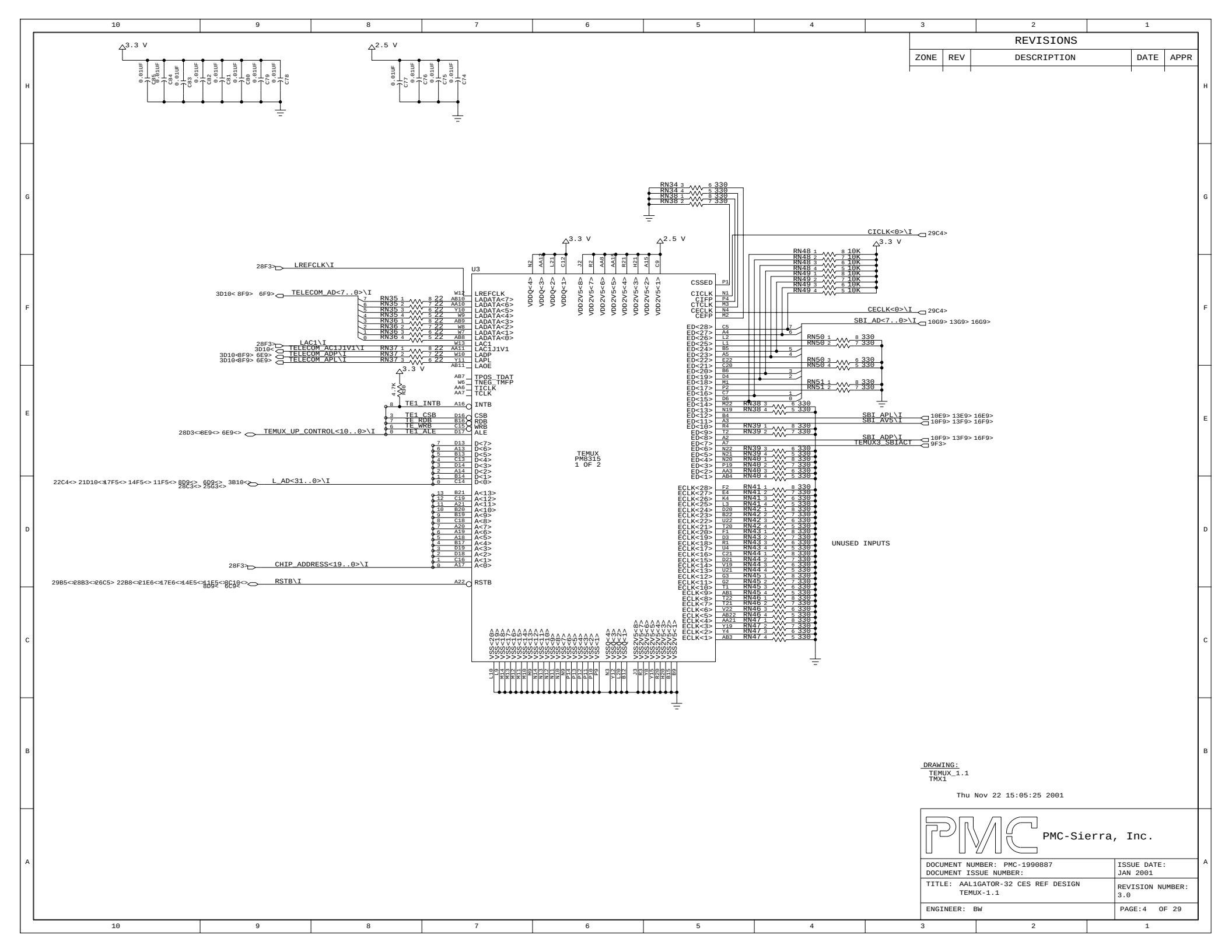
DRAWING:
OPTICAL_BLOCK
OPTICAL_BLOCK

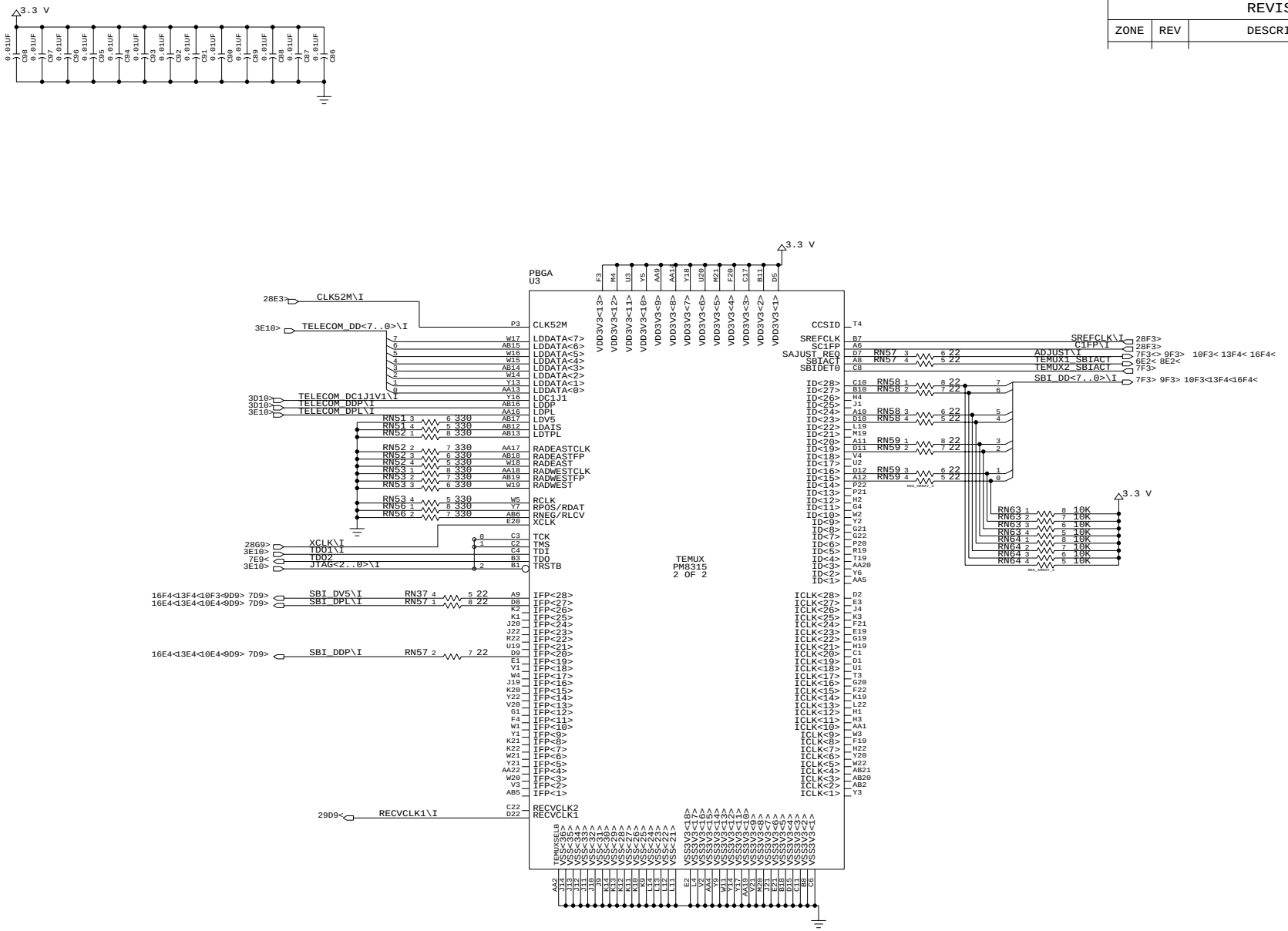
Thu Nov 22 15:06:04 2001

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 REF DESIGN OPTICAL INTERFACE	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:2 OF 29


REVISIONS
ZONE | REV | DESCRIPTION | DATE | APPR
SPECTRA-155 ALARMS LEDS
SALM
LOF
LOS
LAIS
LRDI
RALM1
RALM2
RALM3
HCT540
U50
RALM<3..1>
TAVD<3..1>
RAVD<4..1>
QAVD<3..1>
J8
JTAG PORT
TCK
TMS
TDI
TDO
TRSTB
TRIS_OHB
TDO13\I
TDO1\I
JTAG<2..0>\I
BYTE TELECOM MODE
SMODE0
SMODE1
SMODE2
SS0_DCK\I
SS1_DFP\I
TELECOM_DD<7..0>\I
SS3_ACK\I
TELECOM_AD<7..0>\I
TELECOM_DPL\I
TELECOM_DC1J1V1\I
TELECOM_DDP\I
TELECOM_APL\I
TELECOM_AC1J1V1\I
TELECOM_ADP\I
SPECTRA
PM5342
RSTB\I
SPECTRA_UP_CONTROL<4..0>\I
INTB
CHIP_ADDRESS<19..0>\I
L_AD<31..0>\I
RRCLK\I
RXDP\I
RXDN\I
SD\I
TRCLK\I
TXDP\I
TXDN\I
TPOH<3..1>
TPOHEN<3..1>
TP1
TP2
TP3
TP4
TP5
DRAWING:
SPECTRA_BLOCK
SPECTRA_155
Thu Nov 22 15:05:21 2001
PMC-Sierra, Inc.
DOCUMENT NUMBER: PMC-1990887
DOCUMENT ISSUE NUMBER:
ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 CES REF DESIGN SPECTRA-155
REVISION NUMBER: 3.0
ENGINEER: BW
PAGE:3 OF 29


REVISIONS
ZONE | REV | DESCRIPTION | DATE | APPR
TEMUX
PM8315
1 OF 2
U3
UNUSED INPUTS
DRAWING:
TEMUX_1.1
TMX1
Thu Nov 22 15:05:25 2001
PMC-Sierra, Inc.
DOCUMENT NUMBER: PMC-1990887
DOCUMENT ISSUE NUMBER:
ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 CES REF DESIGN TEMUX-1.1
REVISION NUMBER: 3.0
ENGINEER: BW
PAGE:4 OF 29



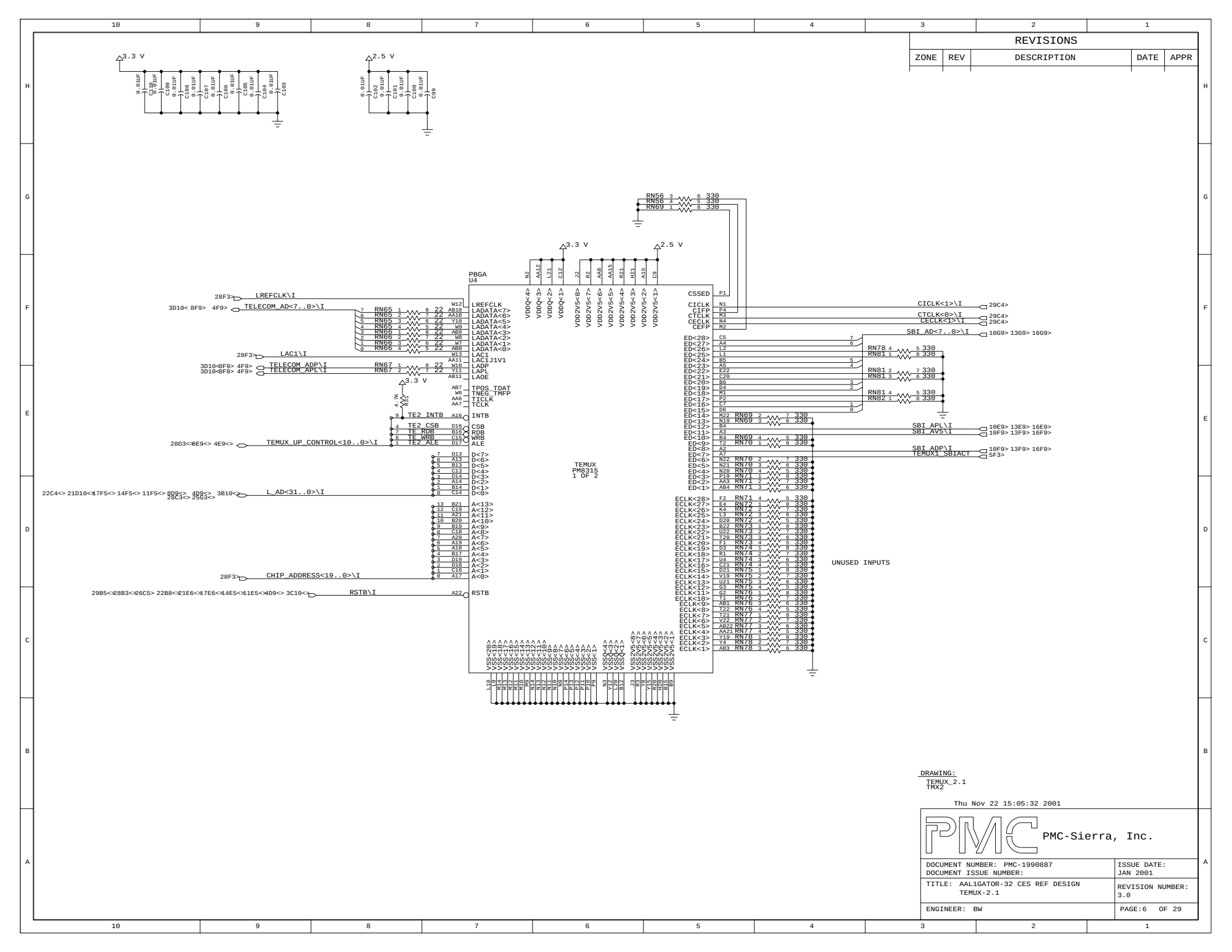
REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR

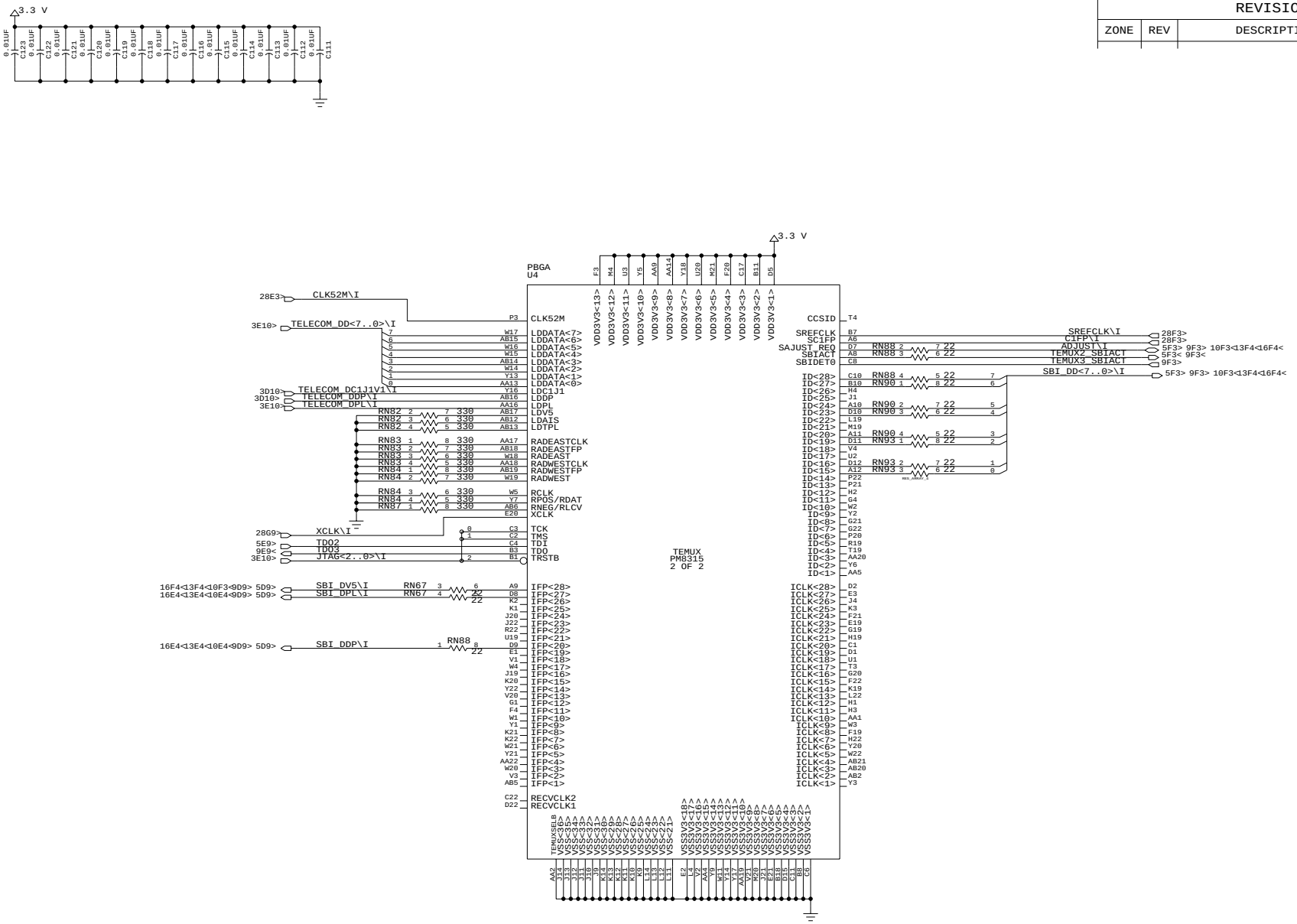
DRAWING:
TEMUX_1.2
TMX1

Thu Nov 22 15:05:29 2001

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 CES REF DESIGN TEMUX-1.2	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:5 OF 29


REVISIONS
ZONE | REV | DESCRIPTION | DATE | APPR
3.3 V
2.5 V
PBGA
U4
TEMUX
PM8315
1 OF 2
UNUSED INPUTS
LREFCLK\I
TELECOM_AD<7..0>\I
LAC1\I
TELECOM_ADP\I
TELECOM_APL\I
TE2_INTB
TE2_CSB
TE_RDB
TE_WRB
TE2_ALE
TEMUX_UP_CONTROL<10..0>\I
L_AD<31..0>\I
CHIP_ADDRESS<19..0>\I
RSTB\I
CICLK<1>\I
CTCLK<0>\I
CECLK<1>\I
SBI_AD<7..0>\I
SBI_APL\I
SBI_AVS\I
SBI_ADP\I
TEMUX1_SBIACT
DRAWING:
TEMUX_2.1
TMX2
Thu Nov 22 15:05:32 2001
PMC-Sierra, Inc.
DOCUMENT NUMBER: PMC-1990887
DOCUMENT ISSUE NUMBER:
ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 CES REF DESIGN TEMUX-2.1
REVISION NUMBER: 3.0
ENGINEER: BW
PAGE:6 OF 29



REVISIONS

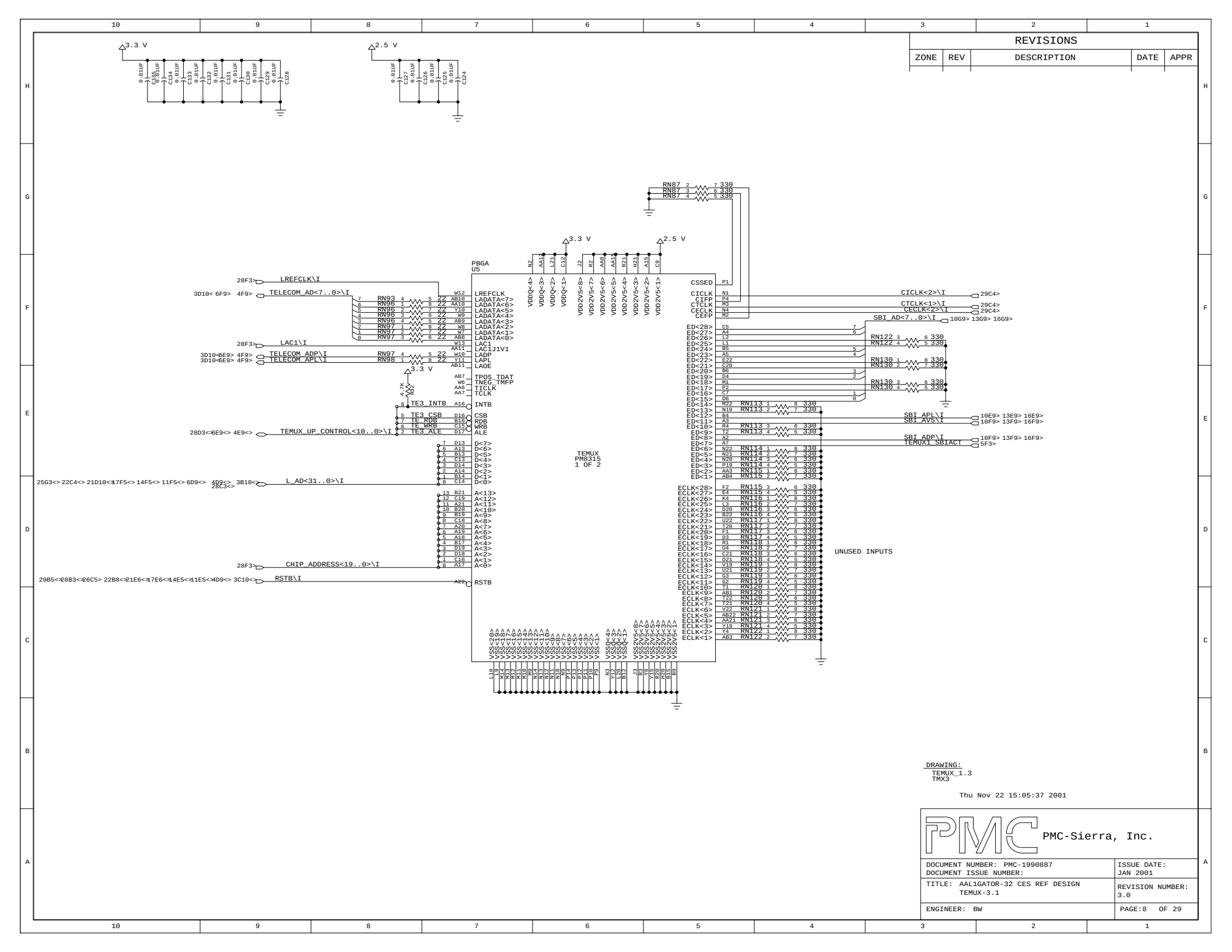
ZONE	REV	DESCRIPTION	DATE	APPR

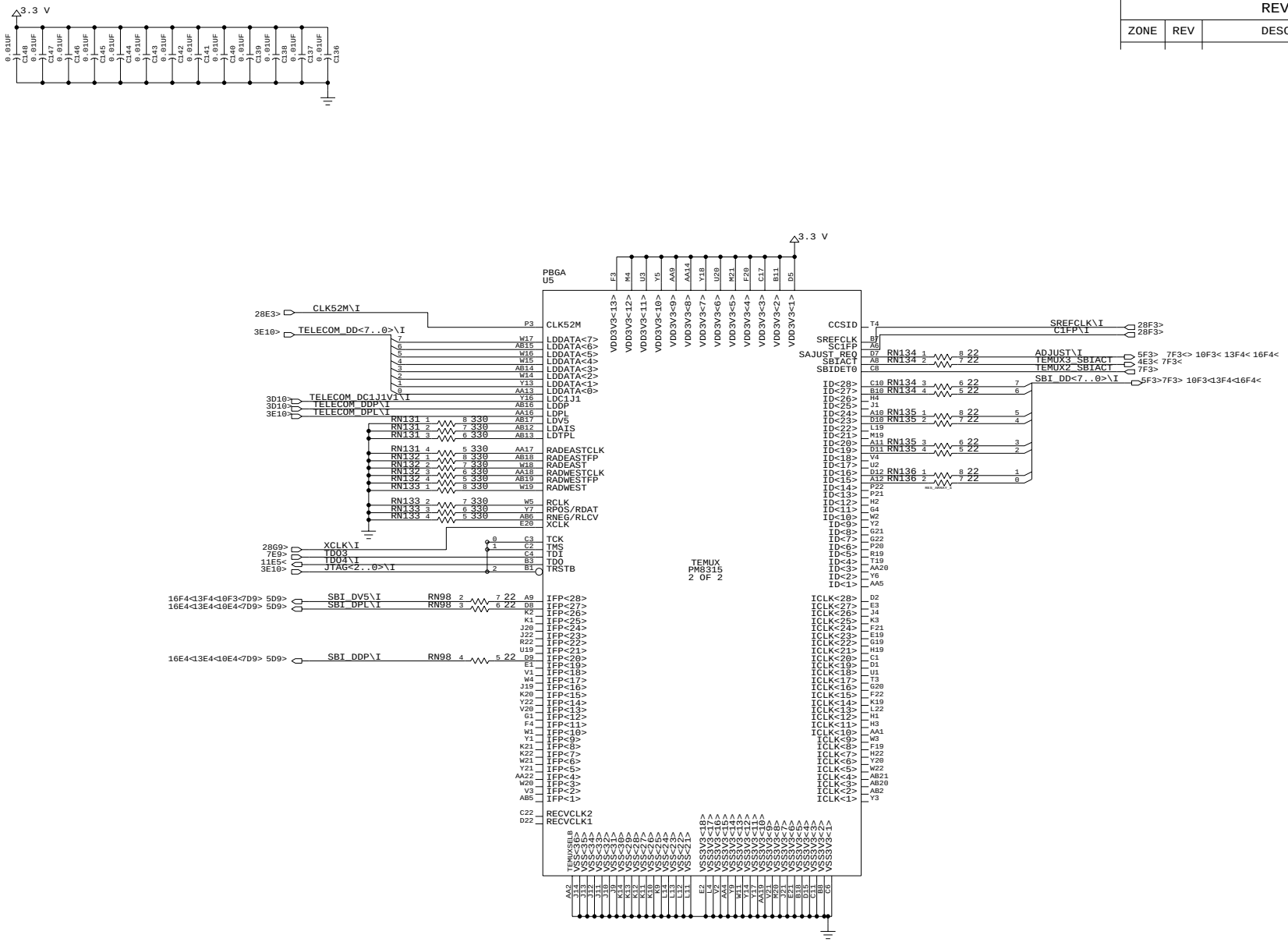
DRAWING:
TEMUX_2.2
TMX2

Thu Nov 22 15:05:34 2001

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 CES REF DESIGN TEMUX-2.2	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:7 OF 29


REVISIONS
ZONE | REV | DESCRIPTION | DATE | APPR
TEMUX
PM8315
1 OF 2
PBGA
U5
UNUSED INPUTS
DRAWING:
TEMUX_1.3
TMX3
Thu Nov 22 15:05:37 2001
PMC-Sierra, Inc.
DOCUMENT NUMBER: PMC-1990887
DOCUMENT ISSUE NUMBER:
ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 CES REF DESIGN TEMUX-3.1
REVISION NUMBER: 3.0
ENGINEER: BW
PAGE:8 OF 29

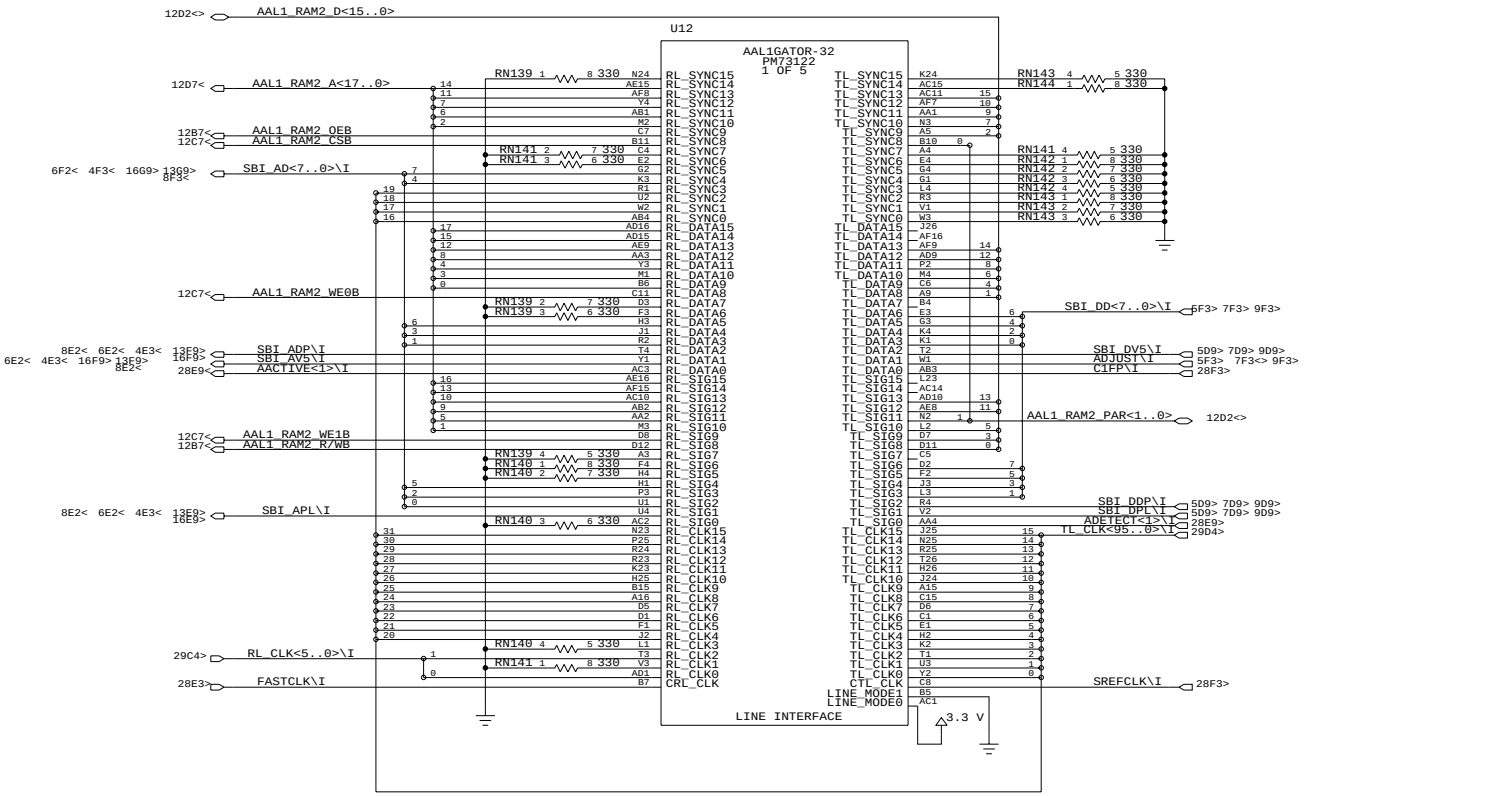


REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 REF DESIGN TEMUX-3.2	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:9 OF 29

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR



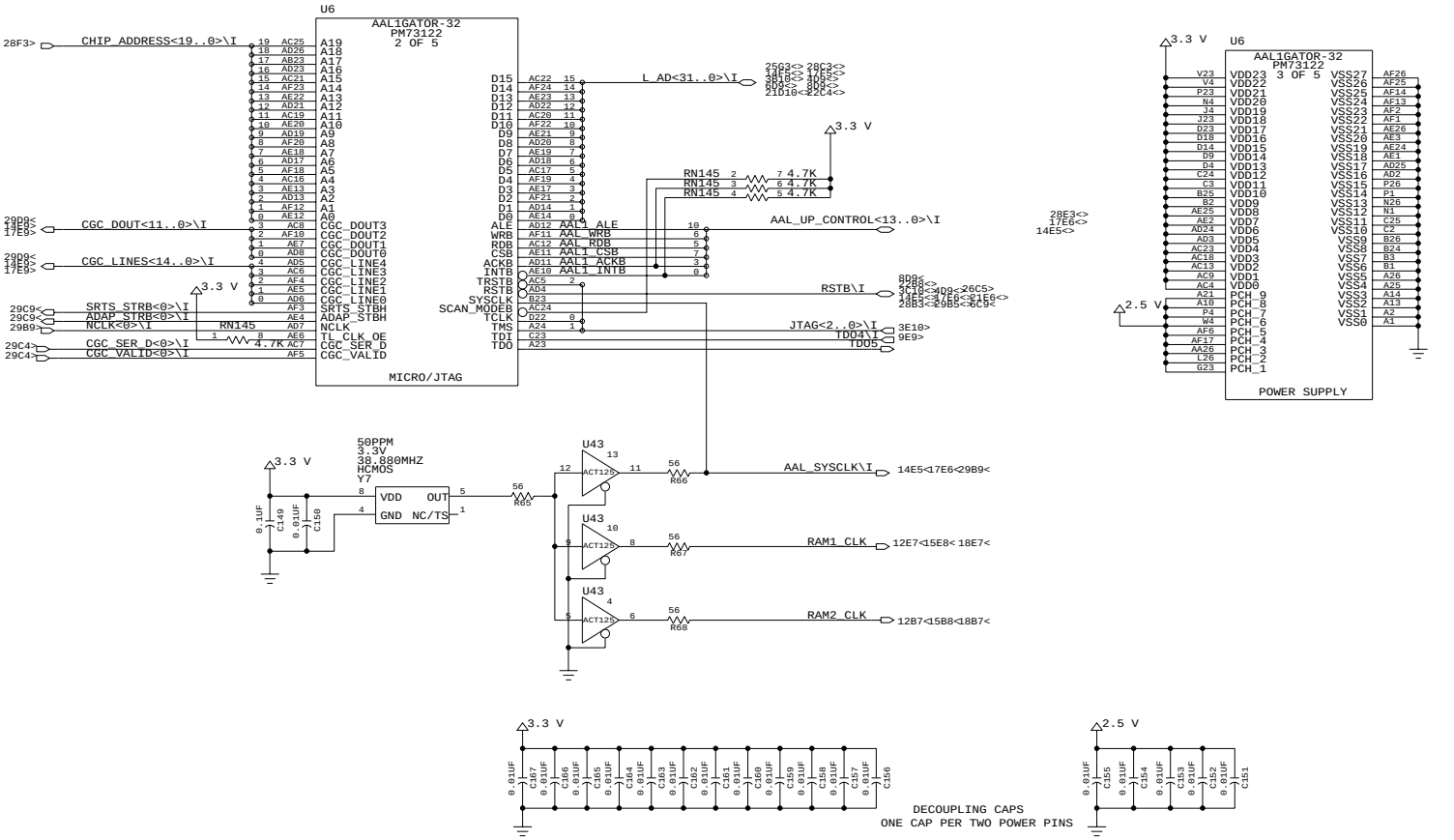
DRAWING:
AAL1GATOR_32_1.1
AAL1

Thu Nov 22 15:05:43 2001

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 CES REF DESIGN AAL1GATOR-32_1 LINE-UTOPIA INTER.	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:10 OF 29

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR

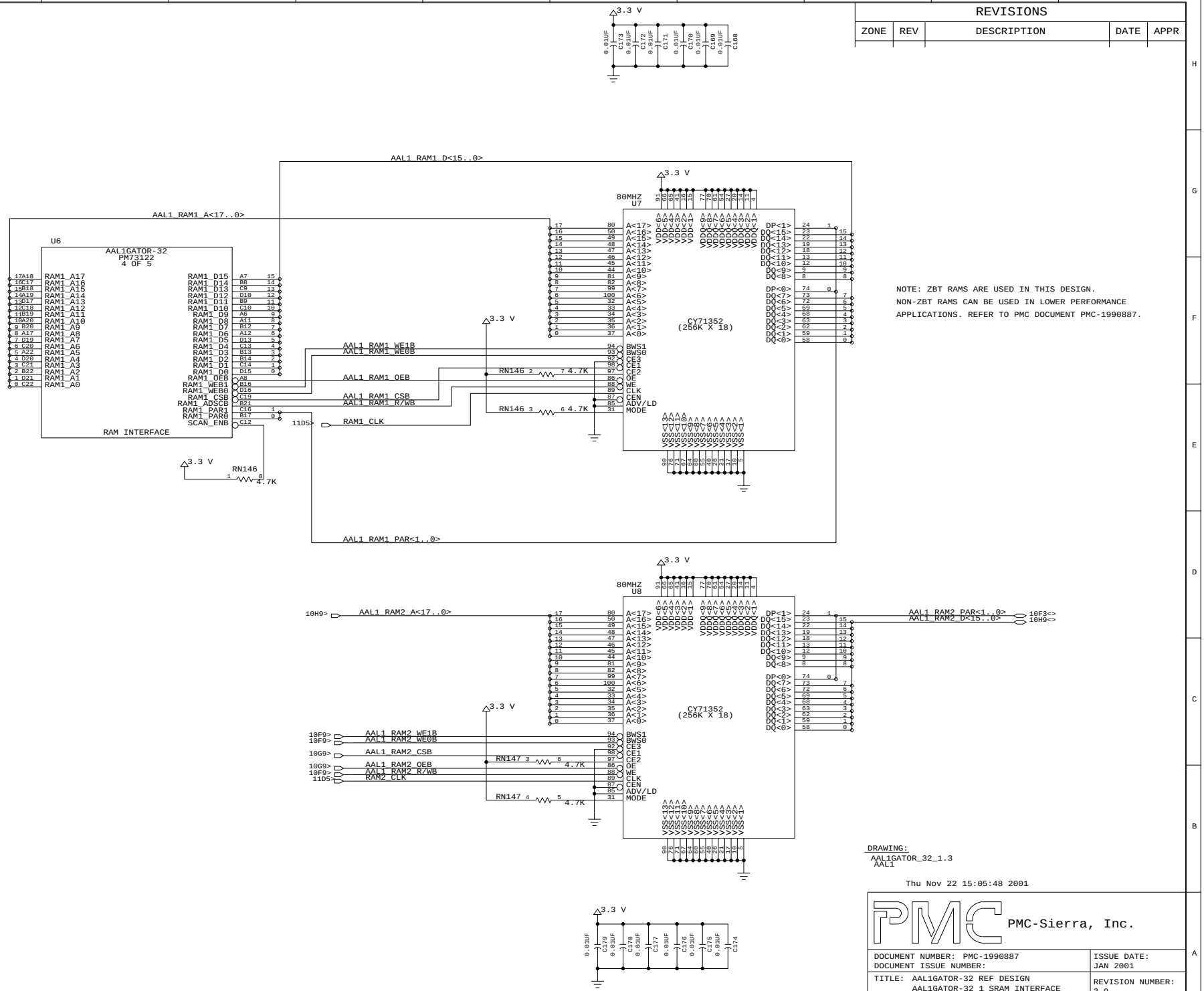


DRAWING:
AAL1GATOR_32_1.2
AAL1

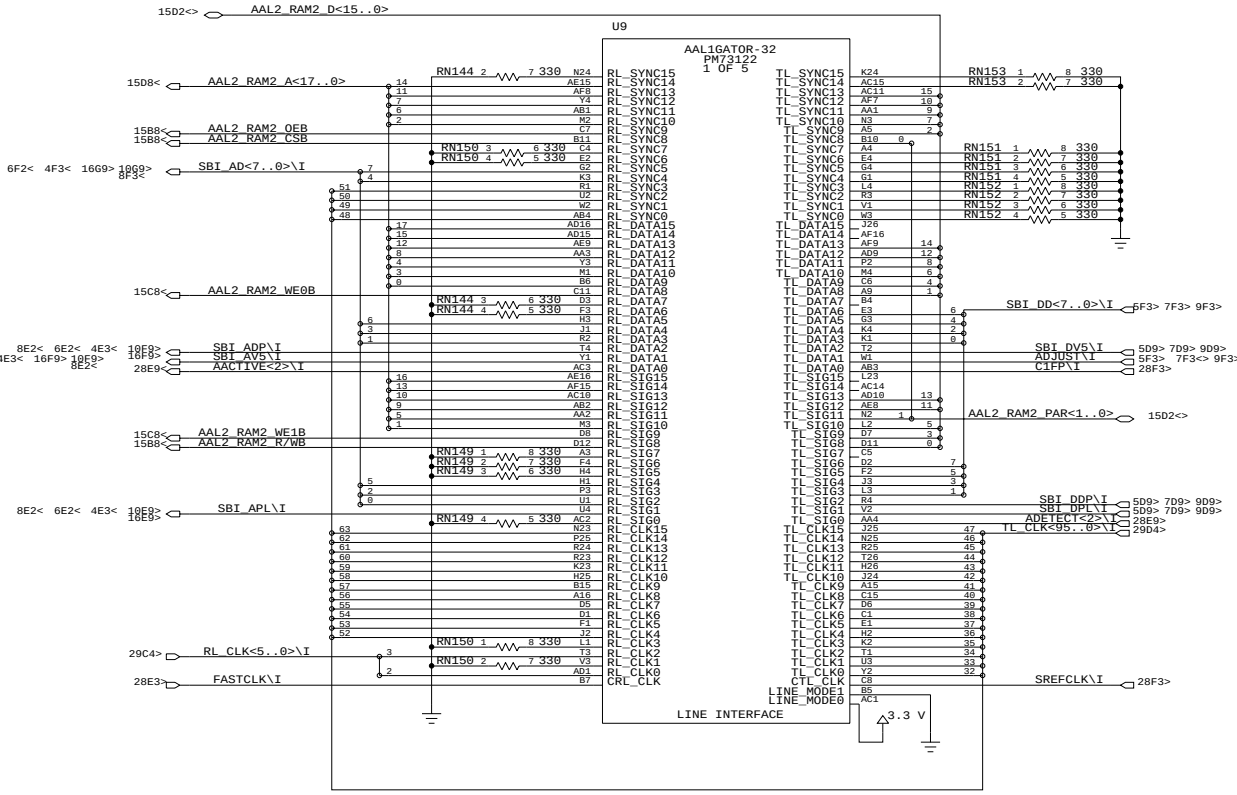
Thu Nov 22 15:05:45 2001

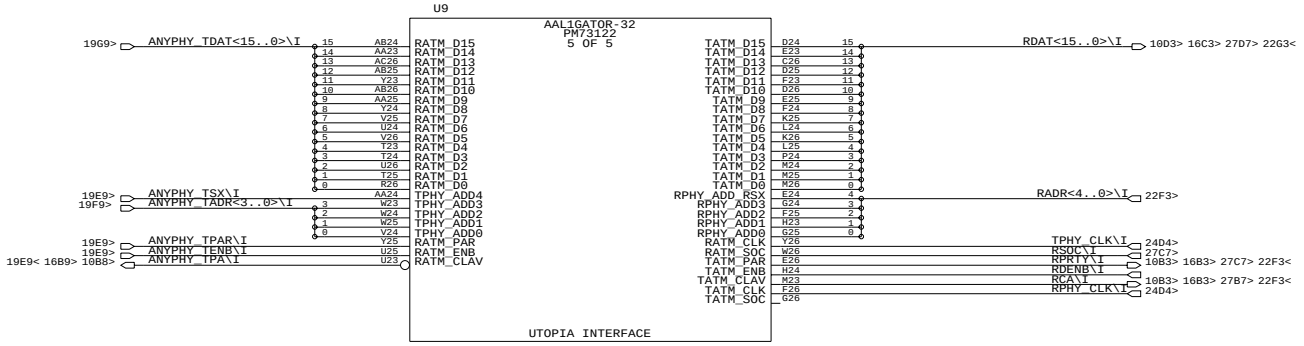
PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 CES REF DESIGN AAL1GATOR-32_1 POWER AND UP	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:11 OF 29


REVISIONS
ZONE | REV | DESCRIPTION | DATE | APPR
3.3 V
AAL1_RAM1_D<15..0>
AAL1_RAM1_A<17..0>
U6
AAL1GATOR-32
PM73122
4 OF 5
RAM_INTERFACE
RAM1_A17
RAM1_A16
RAM1_A15
RAM1_A14
RAM1_A13
RAM1_A12
RAM1_A11
RAM1_A10
RAM1_A9
RAM1_A8
RAM1_A7
RAM1_A6
RAM1_A5
RAM1_A4
RAM1_A3
RAM1_A2
RAM1_A1
RAM1_A0
RAM1_D15
RAM1_D14
RAM1_D13
RAM1_D12
RAM1_D11
RAM1_D10
RAM1_D9
RAM1_D8
RAM1_D7
RAM1_D6
RAM1_D5
RAM1_D4
RAM1_D3
RAM1_D2
RAM1_D1
RAM1_D0
RAM1_OEB
RAM1_WEB1
RAM1_WEB0
RAM1_CSB
RAM1_ADSCB
RAM1_PAR1
RAM1_PAR0
SCAN_ENB
RN146
4.7K
AAL1_RAM1_WE1B
AAL1_RAM1_WE0B
AAL1_RAM1_OEB
AAL1_RAM1_CSB
AAL1_RAM1_R/WB
RAM1_CLK
11D5>
80MHZ
U7
CY71352
(256K X 18)
BWS1
BWS0
CE3
CE1
CE2
OE
WE
CLK
CEN
ADV/LD
MODE
NOTE: ZBT RAMS ARE USED IN THIS DESIGN.
NON-ZBT RAMS CAN BE USED IN LOWER PERFORMANCE
APPLICATIONS. REFER TO PMC DOCUMENT PMC-1990887.
AAL1_RAM1_PAR<1..0>
10H9>
AAL1_RAM2_A<17..0>
U8
AAL1_RAM2_PAR<1..0>
AAL1_RAM2_D<15..0>
10F3<>
10H9<>
10F9>
AAL1_RAM2_WE1B
AAL1_RAM2_WE0B
AAL1_RAM2_CSB
AAL1_RAM2_OEB
AAL1_RAM2_R/WB
RAM2_CLK
RN147
DRAWING:
AAL1GATOR_32_1.3
AAL1
Thu Nov 22 15:05:48 2001
PMC-Sierra, Inc.
DOCUMENT NUMBER: PMC-1990887
DOCUMENT ISSUE NUMBER:
ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 REF DESIGN
AAL1GATOR-32_1 SRAM INTERFACE
REVISION NUMBER: 3.0
ENGINEER: BW
PAGE:12 OF 29

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR





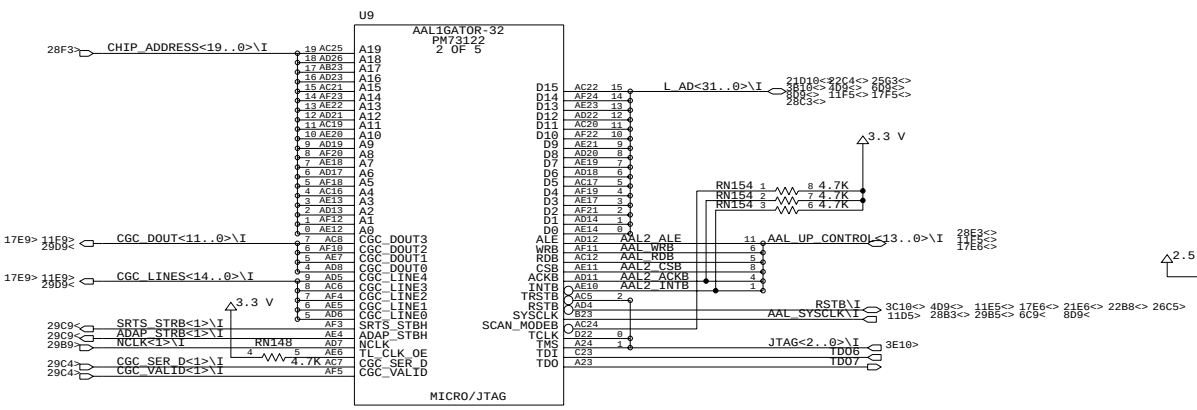
DRAWING:
AAL1GATOR_32_2.1
AAL2

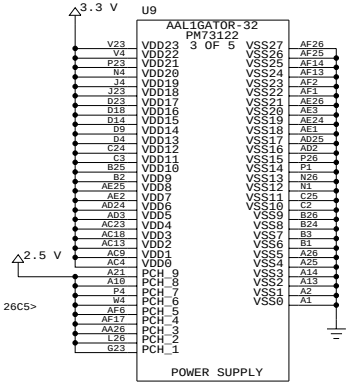
Thu Nov 22 15:05:50 2001

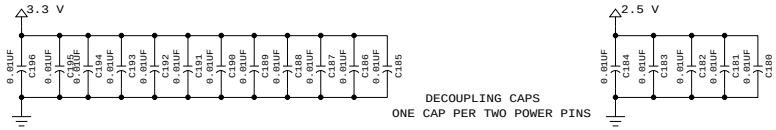
PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 REF DESIGN AAL1GATOR-32_2 LINE-UTOPIA INTER.	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:13 OF 29

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR







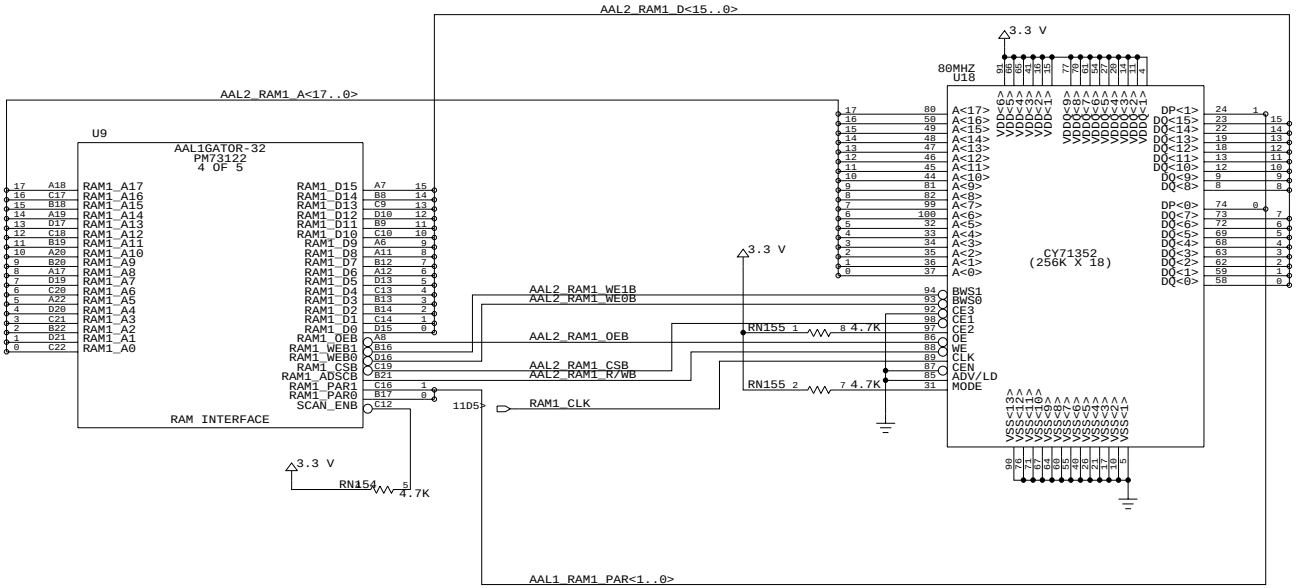
DRAWING:
AAL1GATOR_32_2.2
AAL2

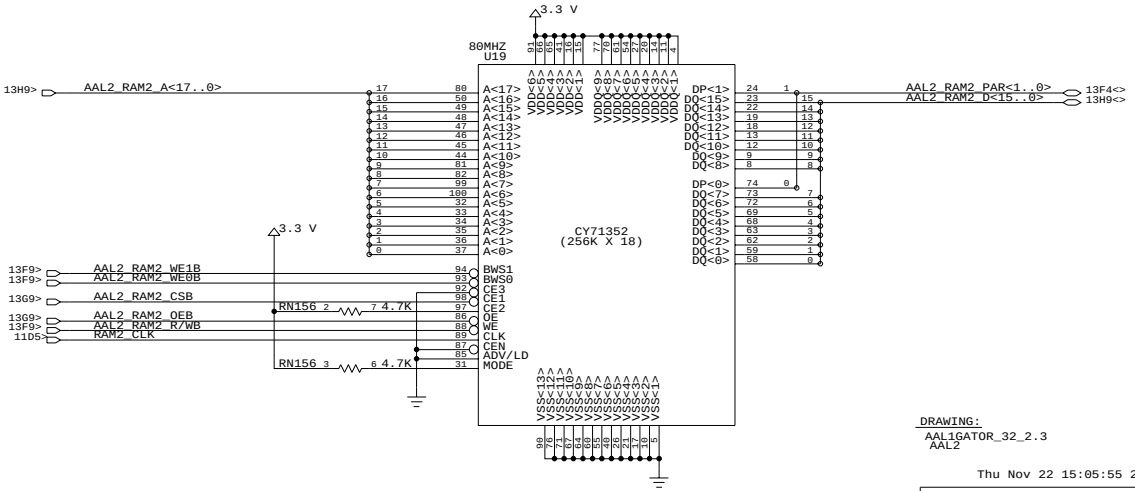
Thu Nov 22 15:05:52 2001

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 REF DESIGN AAL1GATOR-32_2 POWER AND UP	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:14 OF 29

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR





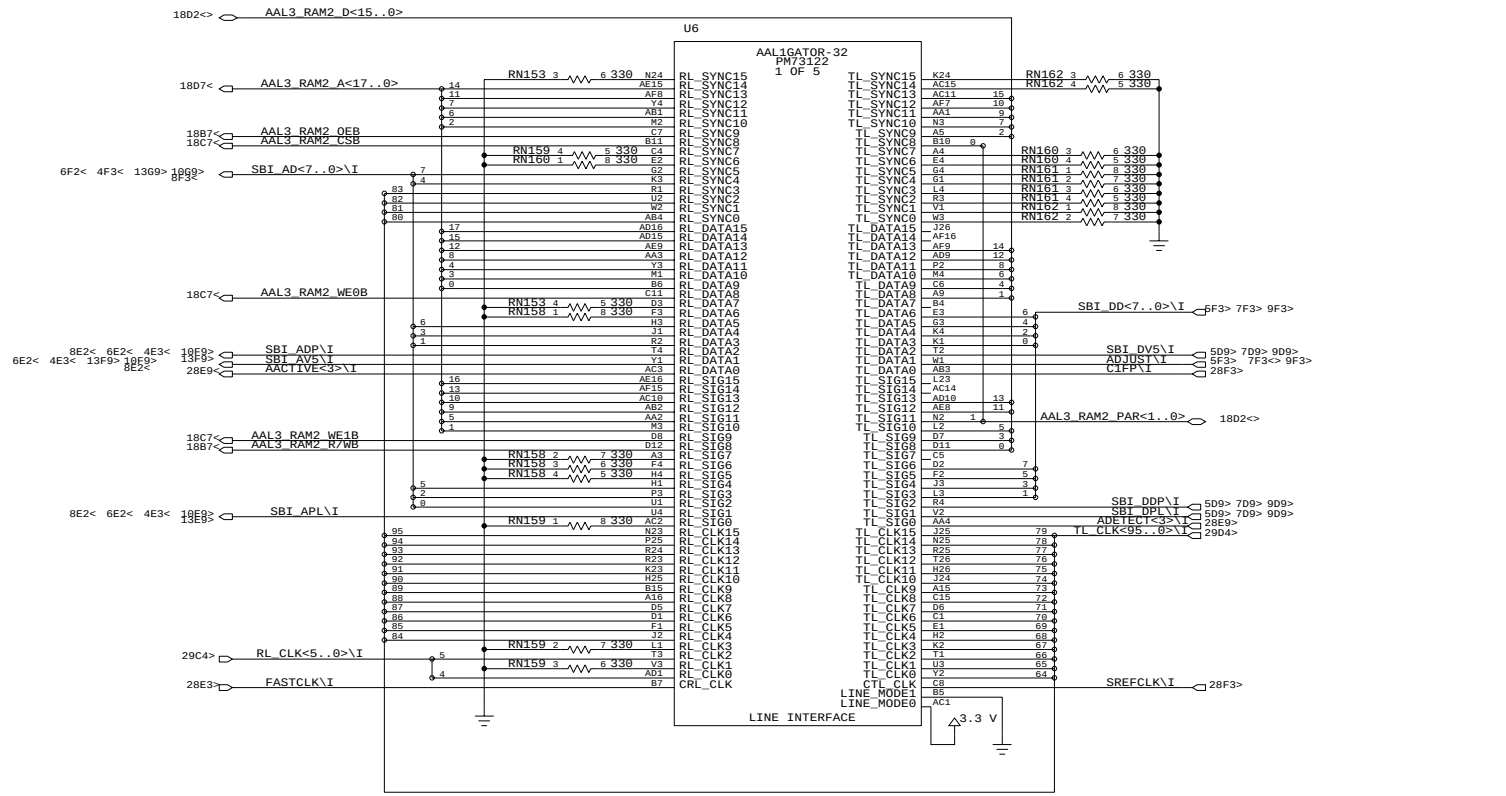
DRAWING:
AAL1GATOR_32_2.3
AAL2

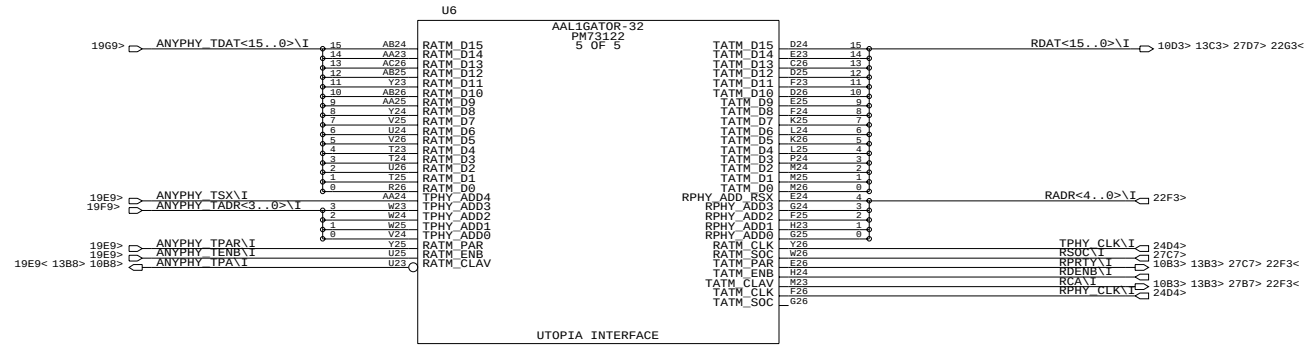
Thu Nov 22 15:05:55 2001

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 CES REF DESIGN AAL1GATOR-32_2 SRAM INTERFACE	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:15 OF 29

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR





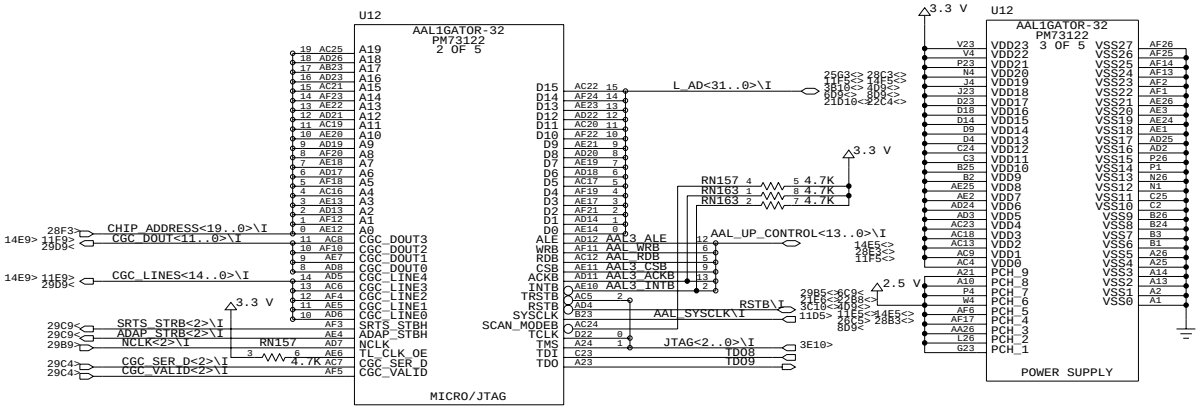
DRAWING:
AAL1GATOR_32_3.1
AAL3

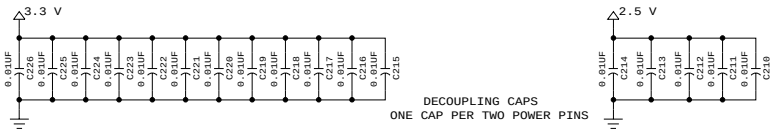
Thu Nov 22 15:05:57 2001

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 REF DESIGN AAL1GATOR-32_3 LINE-UTOPIA INTER.	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:16 OF 29

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR





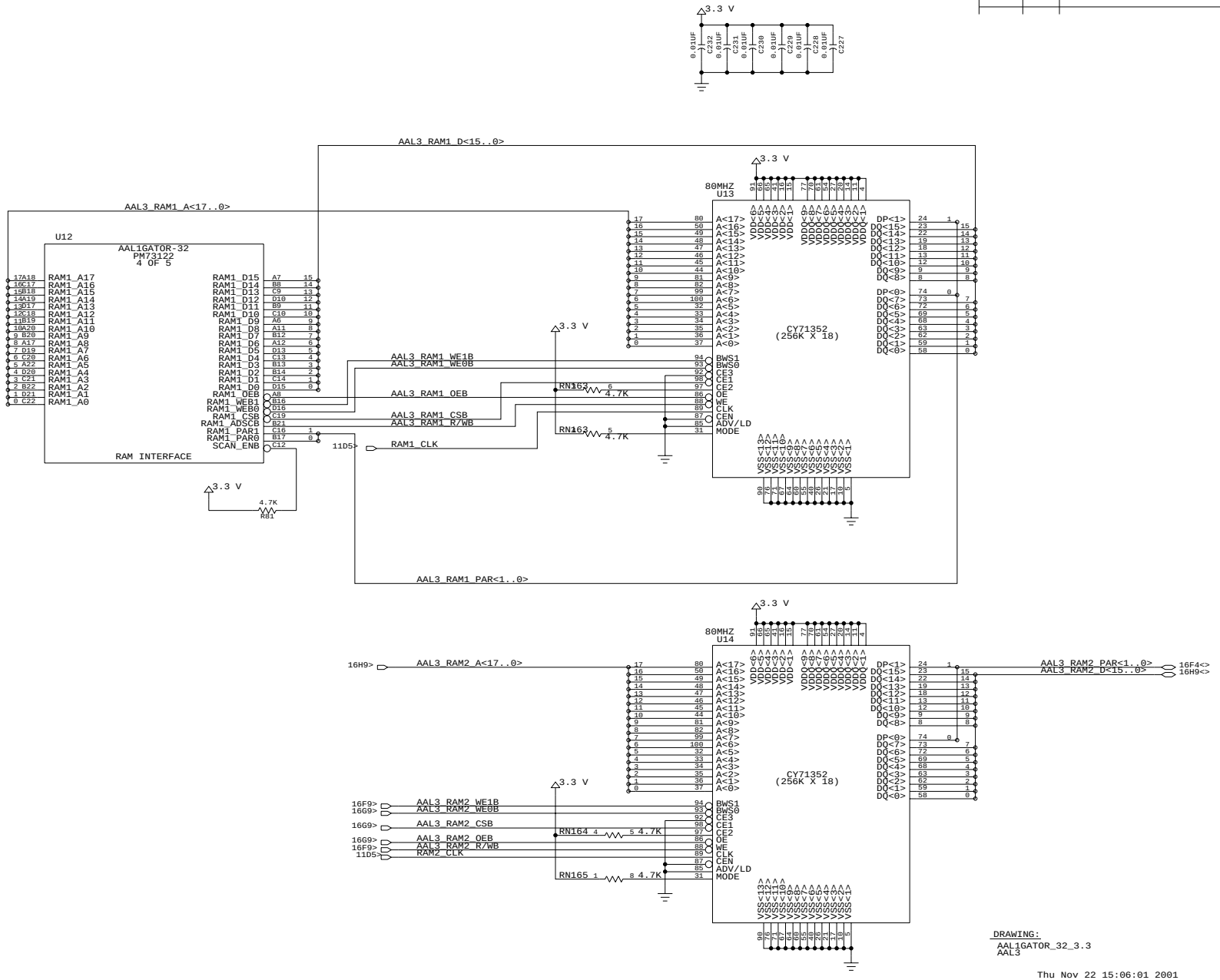
DRAWING:
AAL1GATOR_32_3.2
AAL3

Thu Nov 22 15:05:59 2001

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 REF DESIGN AAL1GATOR-32_3 POWER AND UP	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:17 OF 29

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR



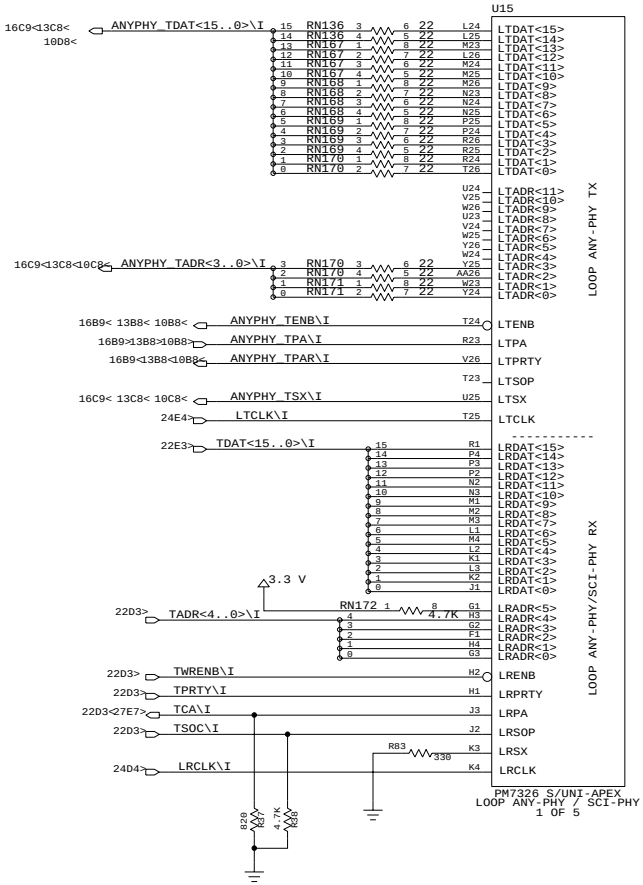
DRAWING:
AAL1GATOR_32_3.3
AAL3

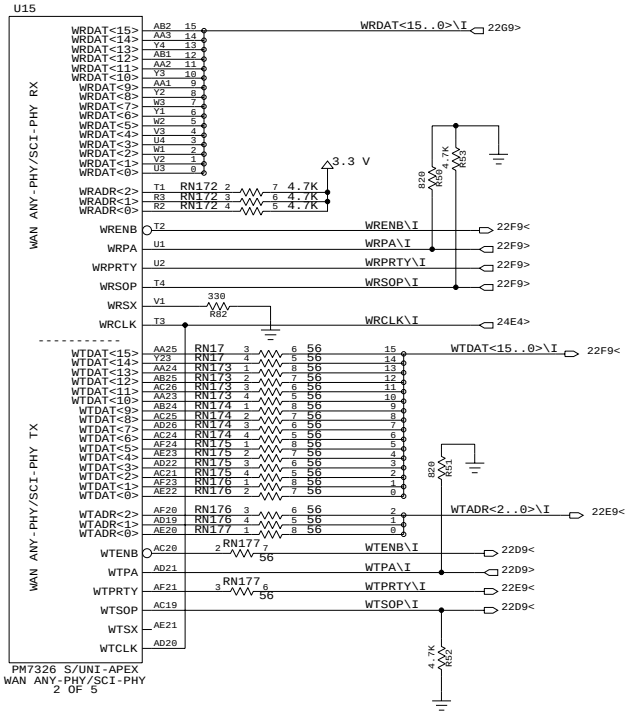
Thu Nov 22 15:06:01 2001

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 CES REF DESIGN AAL1GATOR-32_3 SRAM INTERFACE	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:18 OF 29

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR



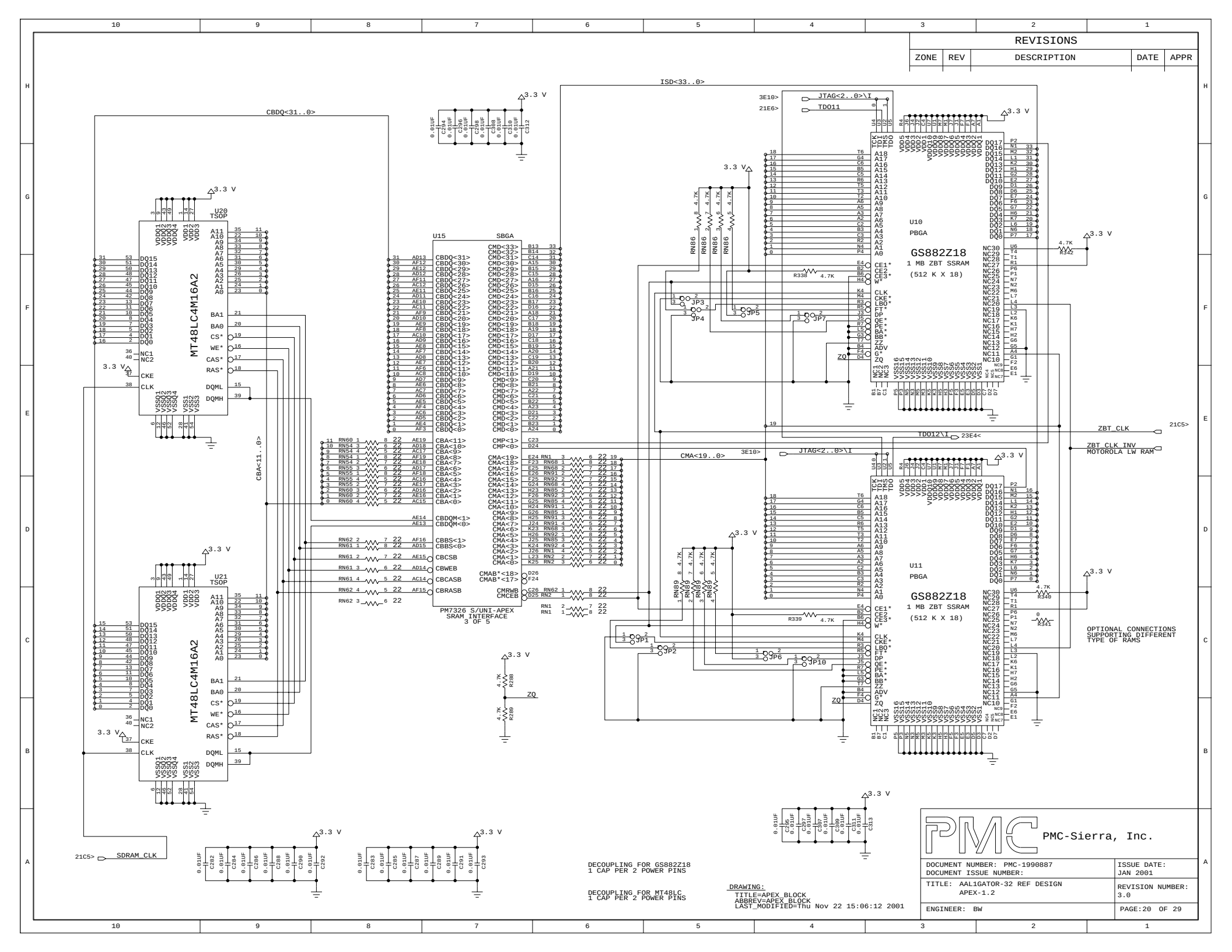


DRAWING:

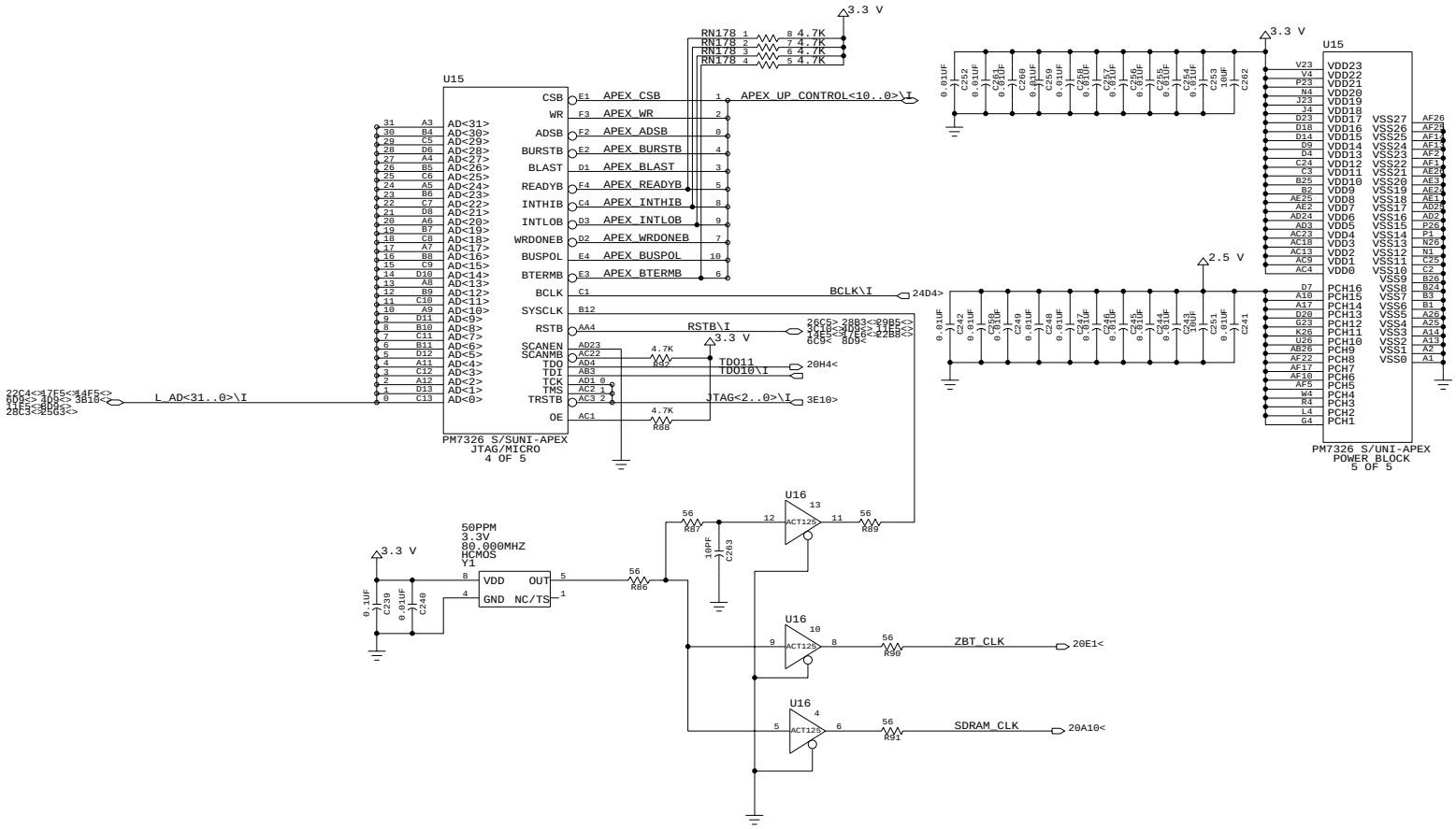
APEX_1.1
APX
Thu Nov 22 15:06:06 2001

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 REF DESIGN APEX-1.1	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:19 OF 29


REVISIONS
ZONE
REV
DESCRIPTION
DATE
APPR
ISD<33..0>
CBDQ<31..0>
CBA<11..0>
CMA<19..0>
U15
SBGA
PM7326 S/UNI-APEX
SRAM_INTERFACE
3 OF 5
U20
TSOP
U21
TSOP
MT48LC4M16A2
U10
PBGA
U11
PBGA
GS882Z18
1 MB ZBT SSRAM
(512 K X 18)
JTAG<2..0>\I
TDO11
TDO12\I
ZBT_CLK
ZBT_CLK_INV
MOTOROLA LW RAM
SDRAM_CLK
3.3 V
ZQ
JP1
JP2
JP3
JP4
JP5
JP6
JP7
JP10
OPTIONAL CONNECTIONS
SUPPORTING DIFFERENT
TYPE OF RAMS
DECOUPLING FOR GS882Z18
1 CAP PER 2 POWER PINS
DECOUPLING FOR MT48LC
1 CAP PER 2 POWER PINS
DRAWING:
TITLE=APEX_BLOCK
ABBREV=APEX_BLOCK
LAST_MODIFIED=Thu Nov 22 15:06:12 2001
PMC-Sierra, Inc.
DOCUMENT NUMBER: PMC-1990887
DOCUMENT ISSUE NUMBER:
ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 REF DESIGN
APEX-1.2
REVISION NUMBER: 3.0
ENGINEER: BW
PAGE:20 OF 29

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR



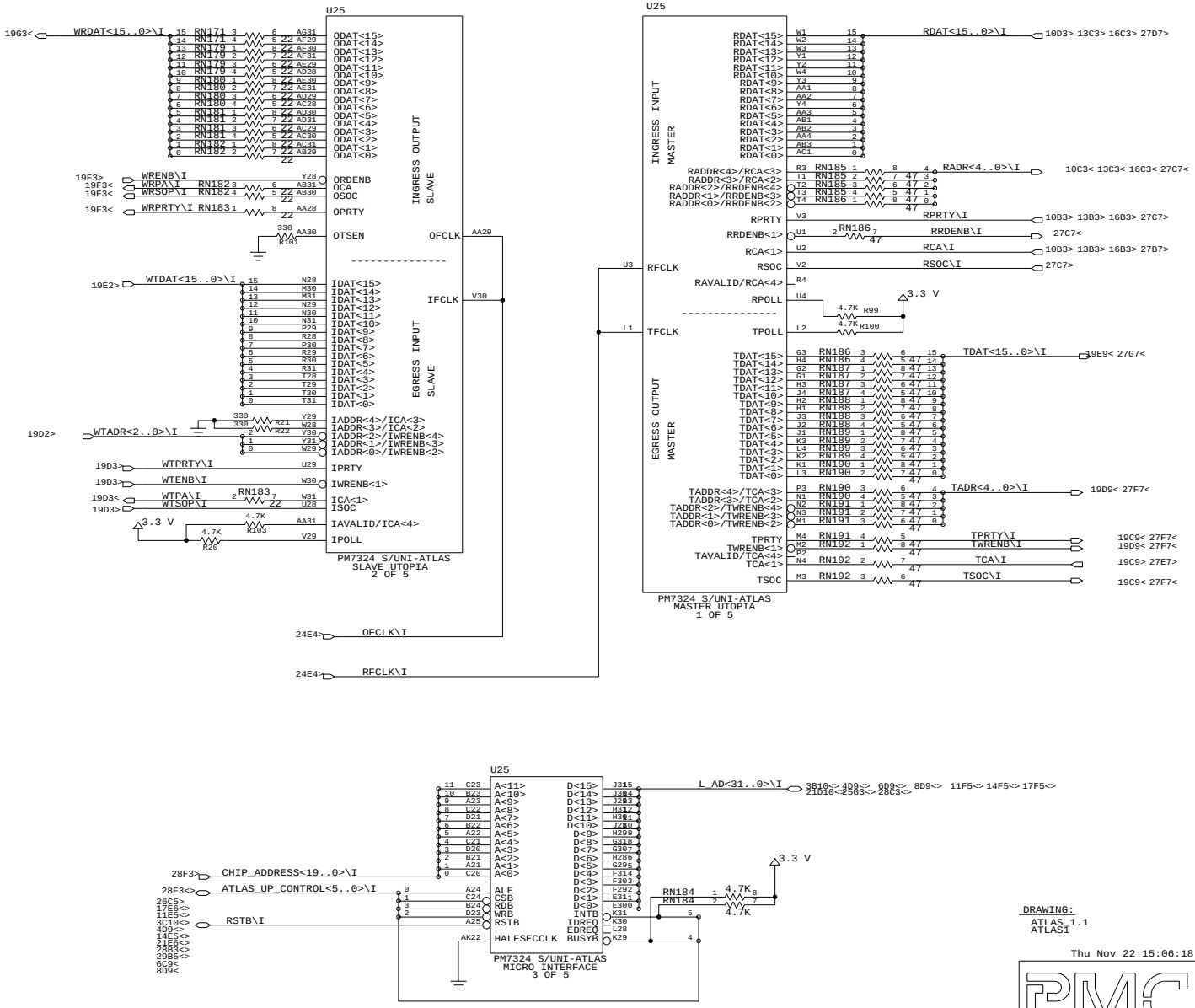
DRAWING:
APEX_1.3
APX

Thu Nov 22 15:06:15 2001

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 REF DESIGN APEX-1.3	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:21 OF 29

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR

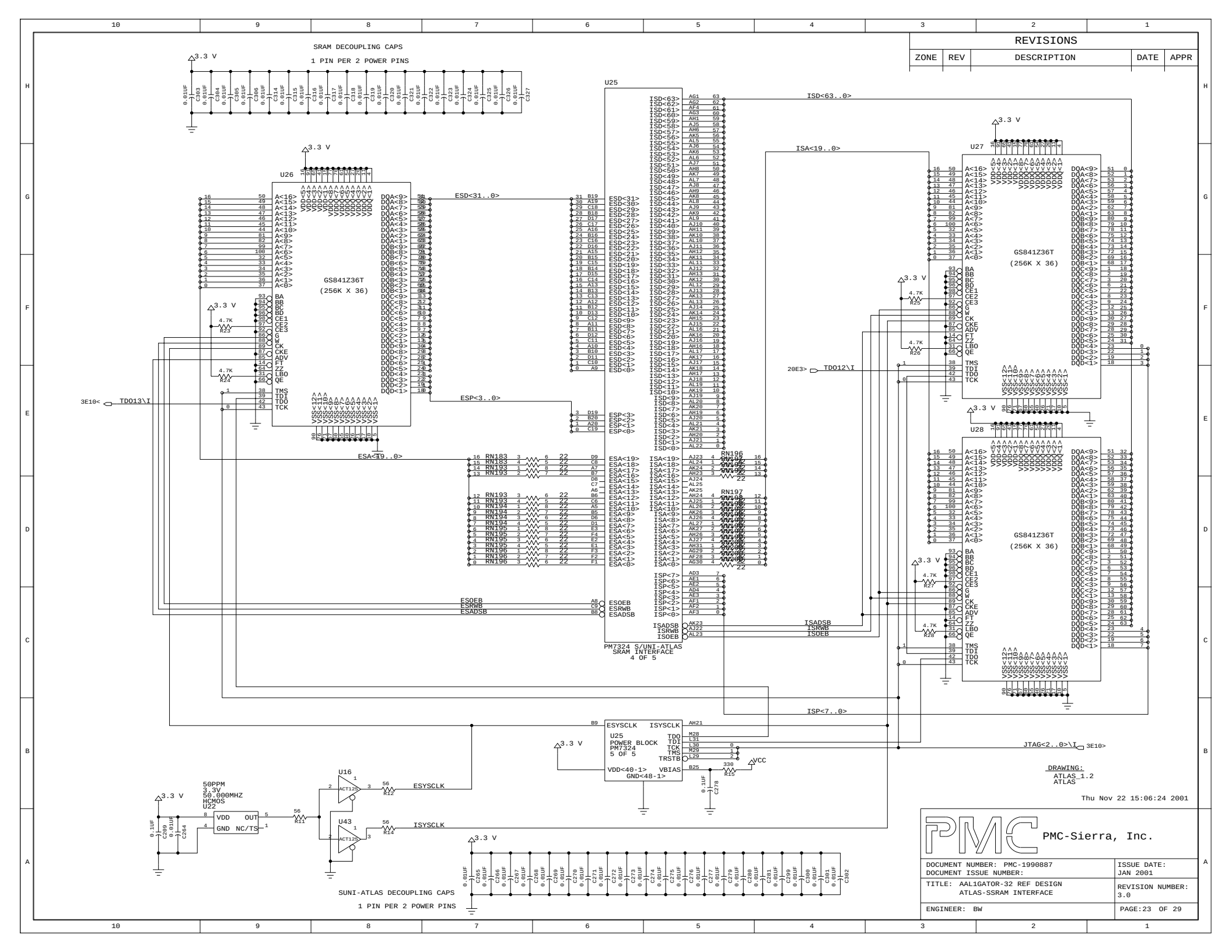


DRAWING:
ATLAS_1.1
ATLAS1

Thu Nov 22 15:06:18 2001

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 REF DESIGN ATLAS INGRESS AND EGRESS INTERFACE	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:22 OF 29

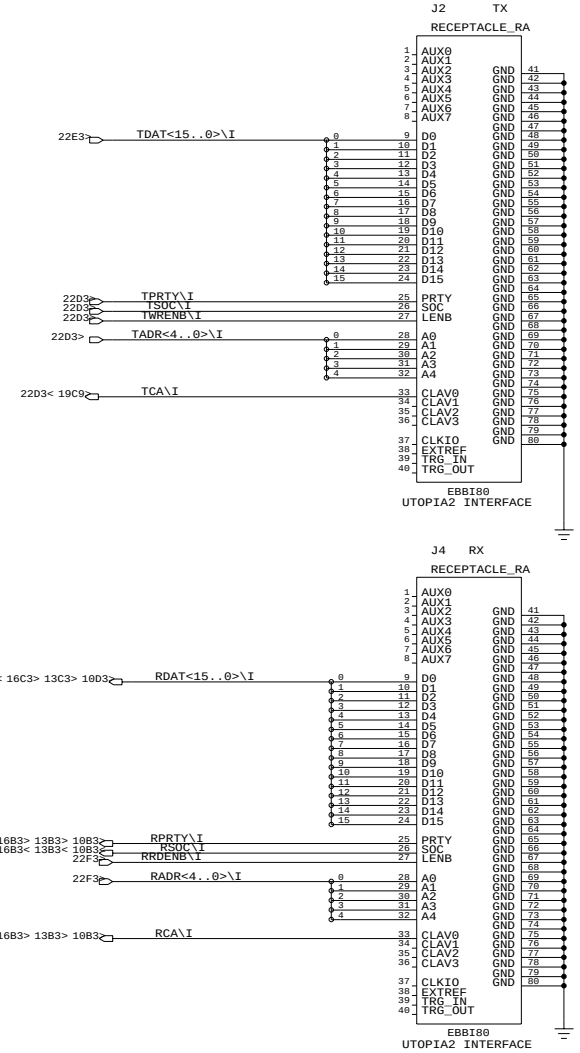


REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 REF DESIGN ATLAS-SSRAM INTERFACE	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:23 OF 29

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR



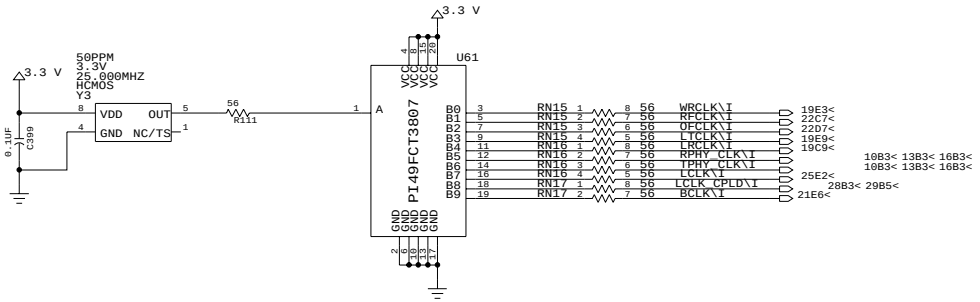
DRAWING:
ATLAS 1.6
ATLAS

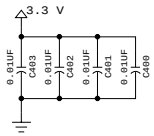
Thu Nov 22 15:05:15 2001

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 REF DESGIN UTOPIA CONNECTOR	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:24 OF 29

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR





DECOUPLING CAPS

DRAWING:
OSCIALLTOE_25M
OSCILLATOR_25M

Thu Nov 22 15:05:08 2001

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 REF DESIGN 25MHZ OSCILLATOR BLOCK	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:25 OF 29

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR

U47
XC95288XL-PQ208
CPLD
1 OF 2

U47
XC95288XL-PQ208
CPLD
2 OF 2
POWER

XCLK\I
50PPM 3.3V 51.840MHZ HCMOS Y4
50PPM 3.3V 19.440MHZ HCMOS Y5
32PPM 3.3V 37.056MHZ HCMOS Y6

ADETECT<3..1>\I
AACTIVE<3..1>\I
APEX_UP_CONTROL<10..0>\I
APEX_BUSPOL
APEX_INTLOB
APEX_INTHIB
APEX_WRDONEB
APEX_BTERMB
APEX_READYB
APEX_BURSTB
APEX_BLAST
APEX_WR
APEX_CSB
APEX_ADSB
SPECTRA_WRB/RWB
SPECTRA_RDB/E
SPECTRA_CSB
SPECTRA_ALE
SPECTRA_INTB
SPECTRA_UP_CONTROL<4..0>\I
TRCLK\I
RRCLK\I
LA<28..2>\I
L_BLASTB
L_LW/RB
L_READYB
L_LSERRB
L_ADSB
L_WAITB
L_DT_RB
L_DENB
L_ALE
UP_CONTROL<8..0>\I
LBE<1..0>\I
LBE0
LBE1
L_INTB\I
L_BTERMB\I
IOBLOCK

CHIP_ADDRESS<19..0>\I
LREFCLK\I
C1FP\I
LAC1\I
SSI_DFP\I
ATLAS_UP_CONTROL<5..0>\I
ATLAS_INTB
ATLAS_BUSYB
ATLAS_RDB
ATLAS_WRB
ATLAS_CSB
ATLAS_ALE
SREFCLK\I
FASTCLK\I
CLK62M\I
SS0_DCK\I
SS13_ACK\I
AAL_UP_CONTROL<13..0>\I
AAL3_INTB
AAL2_INTB
AAL1_INTB
TE3_INTB
TE2_INTB
TE1_INTB
TEMUX_UP_CONTROL<10..0>\I
L_AD<31..0>\I
LCLK_CPLD\I
RSTB\I
TCK
TDI
TDO
TMS
J6
4.7K RN12

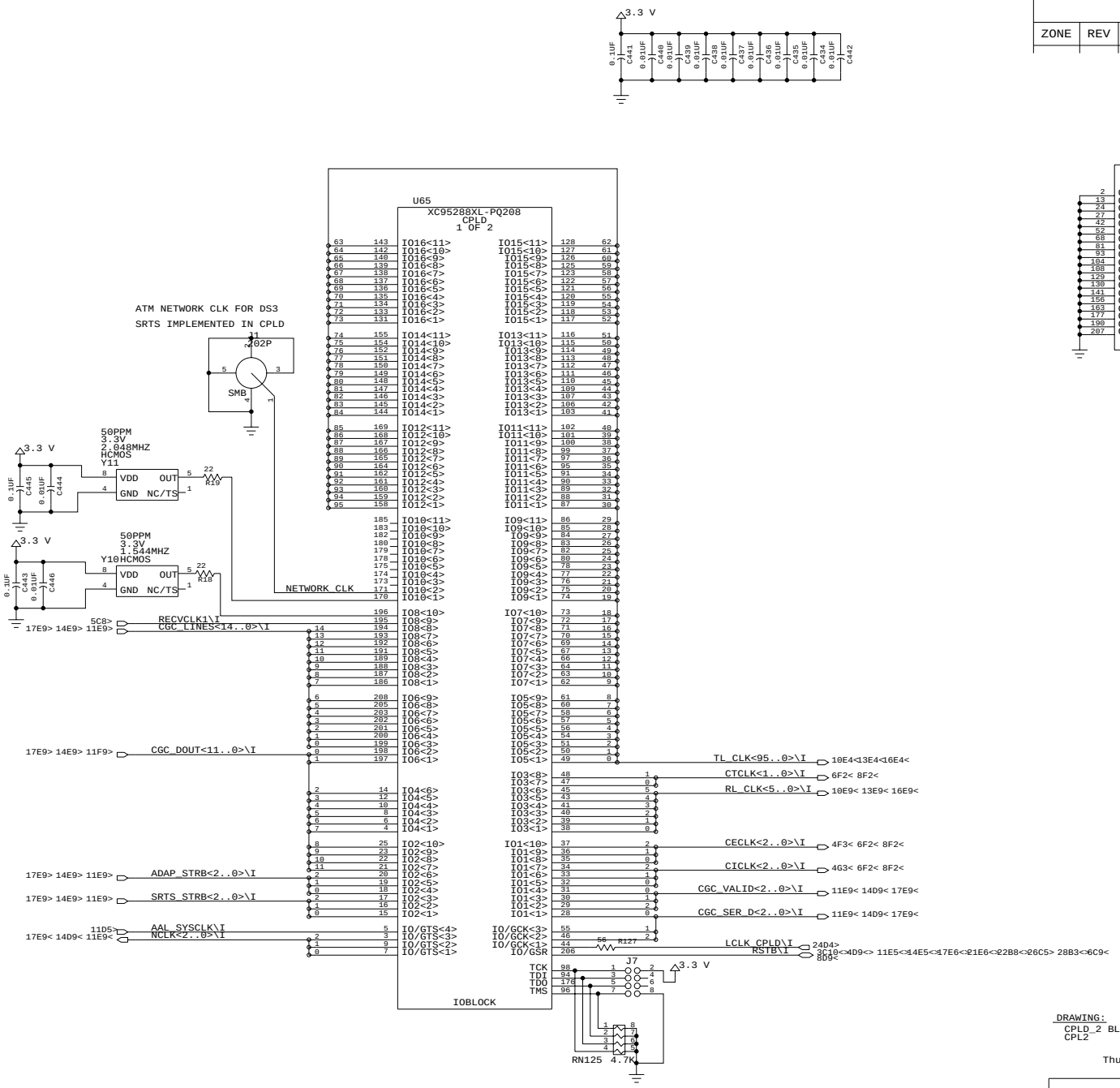
AAL1GATOR_32 INTERRUPT LEDS
TEMUX INTERRUPT LEDS
ATLAS INTERRUPT LED
SPECTRA INTERRUPT LED
YELLOW LEDS
APEX INTERRUPT LEDS
D13 D14 U23 U24 HCT541

DRAWING:
CPLD_1 BLOCK
CPL1 Thu Nov 22 15:05:04 2001

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 REF DESIGN CPLD_1 BLOCK	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:26 OF 29

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR



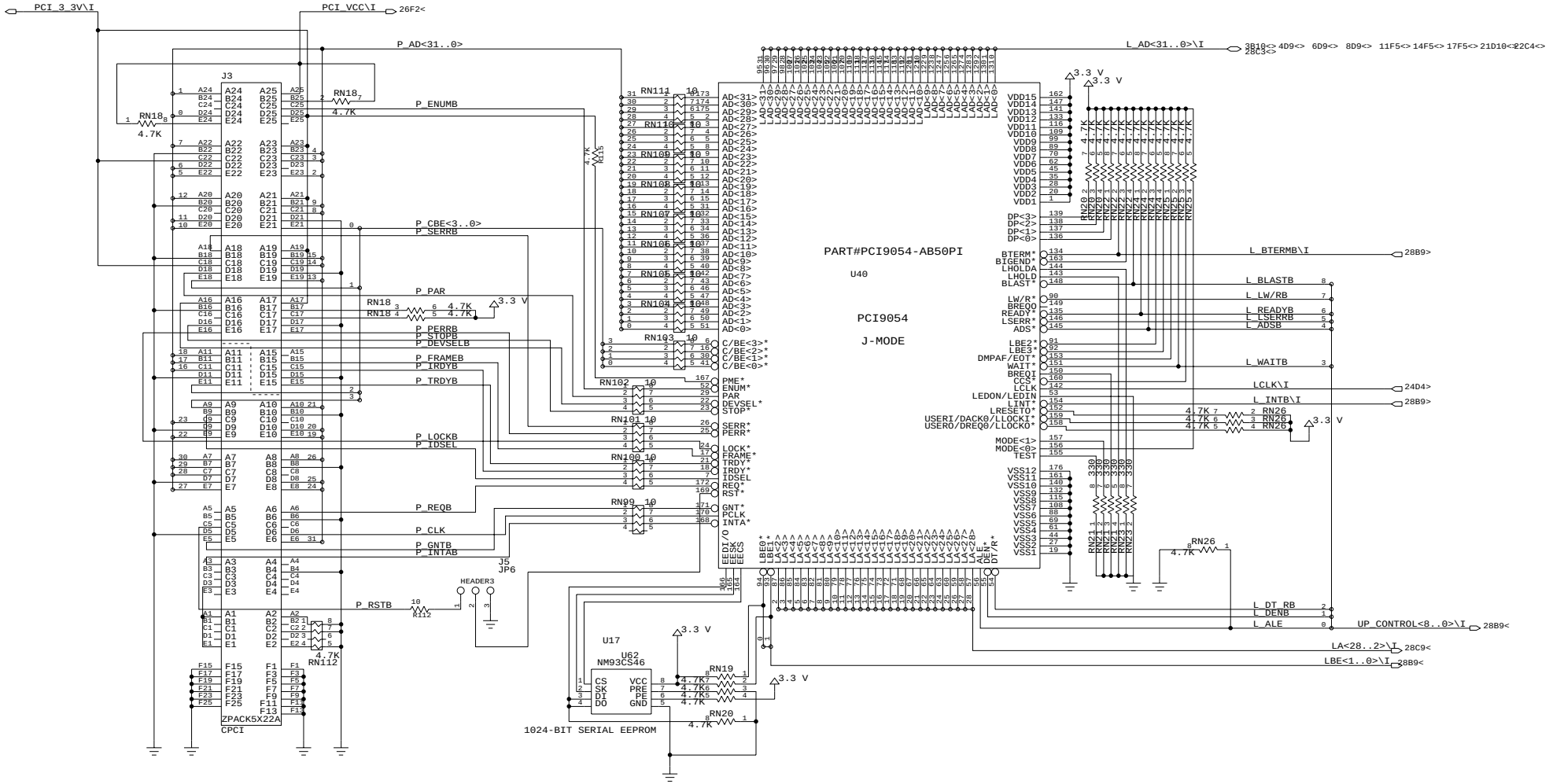
DRAWING:
CPLD_2 BLOCK
CPL2

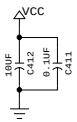
Thu Nov 22 15:05:07 2001

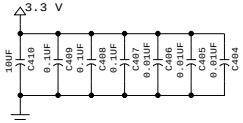
PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 REF DESIGN CPLD_2 BLOCK	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:27 OF 29

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR







PLACE THE DECOUPLING CAPS NEAR THE CHIP.

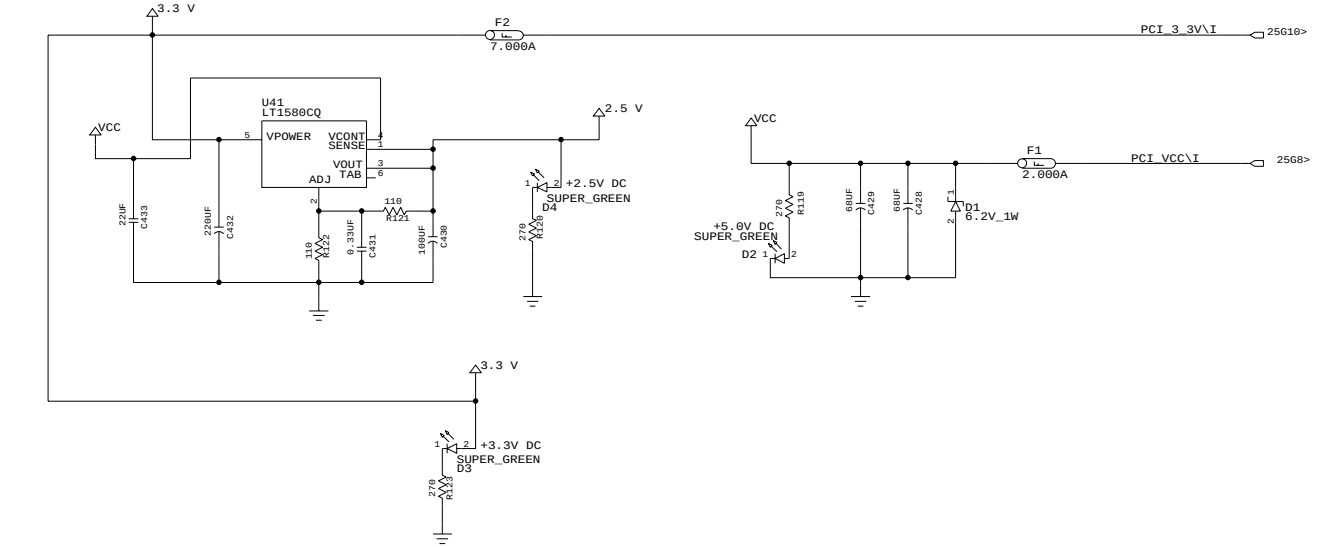
DRAWING:
PCI_BLOCK
PCI_BLOCK

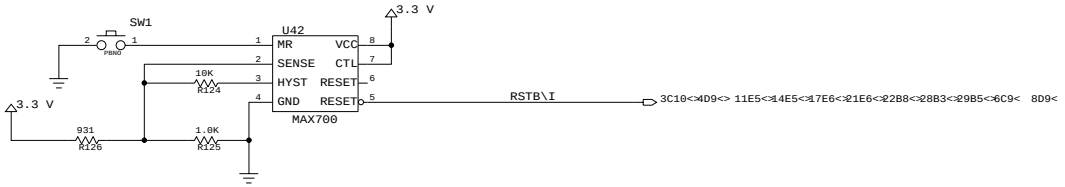
Thu Nov 22 15:05:13 2001

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 REF DESIGN CPCI & PCI9054	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:28 OF 29

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPR





RESET CIRCUITRY

PMC-Sierra, Inc.

DOCUMENT NUMBER: PMC-1990887 DOCUMENT ISSUE NUMBER:	ISSUE DATE: JAN 2001
TITLE: AAL1GATOR-32 REF DESIGN POWER AND RESET BLOCK	REVISION NUMBER: 3.0
ENGINEER: BW	PAGE:29 OF 29

DRAWING:
POWER_AND_RESET
POWER_BLOCK

Thu Nov 22 15:06:03 2001

11 APPENDIX B: U47 MICROPROCESSOR CPLD VHDL CODE

```
-- PMC-Sierra, Inc.
-- PROPRIETARY AND CONFIDENTIAL
--
-- Copyright 2000 PMC-Sierra, Inc.
--
-- All rights reserved. No part of this documentation or computer program may
-- be used, modified, reproduced, or distributed in any form by any means
-- without the prior written permission of PMC-Sierra, Inc.
--
-- This documentation and computer program contains trade secrets,
confidential
-- business information and commercial or financial information (collectively,
-- the "information") of PMC-Sierra, Inc., or unlawful disclosure of any or
all
-- of the information may cause irreparable harm and result in significant
-- commercial and competitive loss to PMC-Sierra, Inc.
-- ------------------------------------------------------------------------
-- ------------------------------------------------------------------------
-- PMC-Sierra, Inc.
-- 105 - 8555 Baxter Place
-- Burnaby, B.C.
-- Canada V5A 4V7
-- Tel: 604-415-6000
-- Fax: 604-415-6206
-- email: apps@pmc-sierra.com
-- ------------------------------------------------------------------------
-- Project     :PMC-990887
-- File Name   :adetect.vhd
-- Path        :
-- Designer    :KM
--
-- Revision History
-- Issue       Date       Initial Descriptions
-- 1           10/12/99   KM
--
-- Function
-- This is a VHDL code for the AAL1gator-32 Reference Design.
-- This code generates ADETECT  signals for AAL1gator-32 devices.
-- ------------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;

entity ADETECTgenerator is
  port
  (
-- input AACTIVE signals to the CPLD:
   AACTIVE  : in std_logic_vector (3 downto 1);
-- output ADETECT signals from the CPLD:
   ADETECT  : out std_logic_vector (3 downto 1)
  );
end ADETECTgenerator;

architecture  ADETECTgenerator_arch of ADETECTgenerator is
```

```
begin
--    process(AACTIVE)
--    begin
      ADETECT(1) <= AACTIVE(2) or AACTIVE(3);
      ADETECT(2) <= AACTIVE(1) or AACTIVE(3);
      ADETECT(3) <= AACTIVE(1) or AACTIVE(2);
--    end process;
end ADETECTgenerator_arch;
```

```
-- PMC-Sierra, Inc.
-- PROPRIETARY AND CONFIDENTIAL
--
-- Copyright 2000 PMC-Sierra, Inc.
--
-- All rights reserved. No part of this documentation or computer program may
-- be used, modified, reproduced, or distributed in any form by any means
-- without the prior written permission of PMC-Sierra, Inc.
--
-- This documentation and computer program contains trade secrets,
confidential
-- business information and commercial or financial information (collectively,
-- the "information") of PMC-Sierra, Inc., or unlawful disclosure of any or
all
-- of the information may cause irreparable harm and result in significant
-- commercial and competitive loss to PMC-Sierra, Inc.
-- ------------------------------------------------------------------------
-- ------------------------------------------------------------------------
-- PMC-Sierra, Inc.
-- 105 - 8555 Baxter Place
-- Burnaby, B.C.
-- Canada V5A 4V7
-- Tel: 604-415-6000
-- Fax: 604-415-6206
-- email: apps@pmc-sierra.com
-- ------------------------------------------------------------------------
-- Project     :PMC-990887
-- File Name   :chipaddress.vhd
-- Path        :
-- Designer    :KM
--
-- Revision History
-- Issue       Date       Initial Descriptions
-- 1           10/13/99   KM        Preliminary
--
-- Function
-- This is a VHDL code for the ASAP-CES Reference Design.
-- This code generates the CHIP_ADDRESS<19..0> signals      to SPECTRA-155,
-- TEMUX , APEX, ATLAS and AAL1gator-32 devices.
-- ------------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity CHIPADgeneration is
  port
  (
-- input signals to the CPLD
   L_ADSB  : in std_logic;
   L_ALE   : in std_logic;
   L_DENB  : in std_logic;
   L_DT_RB : in std_logic;
   L_READYB  : in std_logic;
   L_BLASTB  : in std_logic;
   L_LW_RB : in std_logic;
   L_AD : inout std_logic_vector(31 downto 0);
--  reset signal
```

```
   RSTB   : inout std_logic;
-- CPLD clock:
   LCLK_CPLD : in std_logic;
-- output signals:
   CHIP_ADDRESS : out std_logic_vector(19 downto 0) := (others => 'Z')
  );
end CHIPADgeneration;
architecture  CHIPADgeneration of CHIPADgeneration is
signal RDB : std_logic;
signal WRB : std_logic;
begin
    process(LCLK_CPLD, L_ADSB, L_ALE, L_DT_RB, L_DENB)
    begin
    if (LCLK_CPLD'event and LCLK_CPLD = '1') then
      if (L_ADSB = '0' and L_ALE = '1' and L_BLASTB = '1' and L_DENB = '1')
then
        CHIP_ADDRESS(19 downto 0) <= L_AD(19 downto 0);
--      starting a single read cycle:
        if (L_LW_RB = '0' and L_DT_RB = '0' and L_READYB = '1') then
          RDB <= '0';
--      starting a single write cycle:
        elsif (L_LW_RB = '1' and L_DT_RB = '1' and L_READYB = '1') then
          WRB <= '0';
        end if;
      else
        CHIP_ADDRESS(19 downto 0) <= (others => 'X');
      end if;
    end if;
    end process;
end CHIPADgeneration;
```

```
-- PMC-Sierra, Inc.
-- PROPRIETARY AND CONFIDENTIAL
--
-- Copyright 2000 PMC-Sierra, Inc.
--
-- All rights reserved. No part of this documentation or computer program may
-- be used, modified, reproduced, or distributed in any form by any means
-- without the prior written permission of PMC-Sierra, Inc.
--
-- This documentation and computer program contains trade secrets,
confidential
-- business information and commercial or financial information (collectively,
-- the "information") of PMC-Sierra, Inc., or unlawful disclosure of any or
all
-- of the information may cause irreparable harm and result in significant
-- commercial and competitive loss to PMC-Sierra, Inc.
-- ------------------------------------------------------------------------
-- ------------------------------------------------------------------------
-- PMC-Sierra, Inc.
-- 105 - 8555 Baxter Place
-- Burnaby, B.C.
-- Canada V5A 4V7
-- Tel: 604-415-6000
-- Fax: 604-415-6206
-- email: apps@pmc-sierra.com
-- ------------------------------------------------------------------------
-- Project     :PMC-990887
-- File Name   :chipselect.vhd
-- Path        :
-- Designer    :KM
--
-- Revision History
-- Issue       Date       Initial Descriptions
-- 1           10/12/99   KM        Preliminary
--
-- Function
-- This is a VHDL code for the ASAP-CES Reference Design.
-- This code generates chip select signals for SPECTRA-155, TEMUX,
-- APEX, ATLAS and AAL1gator-32 devices.
-- ------------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity CSBgeneration is
  port
  (
-- input address lines to the CPLD
   LA  : in std_logic_vector (28 downto 20);
-- reset signal
   RSTB   : inout std_logic;
-- Chip select signals:
   APEX_CSB    : out std_logic; -- chip select to APEX
   ATLAS_CSB   : out std_logic; -- chip select to ATLAS
   AAL1_CSB    : out std_logic; -- chip select to AAL1gator 1
   AAL2_CSB    : out std_logic; -- chip select to AAL1gator 2
   AAL3_CSB    : out std_logic; -- chip select to AAL1gator 3
```

```
   TE1_CSB     : out std_logic; -- chip select to TEMUX 1
   TE2_CSB     : out std_logic; -- chip select to TEMUX 2
   TE3_CSB     : out std_logic; -- chip select to TEMUX 3
   SPECTRA_CSB : out std_logic  -- chip select to SPECTRA-155
  );
end CSBgeneration;
architecture  CSBgeneration of CSBgeneration is
begin
    process(LA, RSTB)
    begin
      if RSTB = '0'  then
           APEX_CSB <= '1';
           ATLAS_CSB <= '1';
           AAL1_CSB <= '1';
           AAL2_CSB <= '1';
           AAL3_CSB <= '1';
           TE1_CSB <= '1';
           TE2_CSB <= '1';
           TE3_CSB <= '1';
           SPECTRA_CSB <= '1';
      else
          case LA is
                  when "000000001" => APEX_CSB <= '0';
                  when "000000010" => ATLAS_CSB <= '0';
                  when "000000100" => AAL1_CSB <= '0';
                  when "000001000" => AAL2_CSB <= '0';
                  when "000010000" => AAL3_CSB <= '0';
                  when "000100000" => TE1_CSB <= '0';
                  when "001000000" => TE2_CSB <= '0';
                  when "010000000" => TE3_CSB <= '0';
                  when "100000000" => SPECTRA_CSB <= '0';
              when OTHERS      => SPECTRA_CSB <= '1';  -- do nothing
          end case;
       end if;
    end process;
end CSBgeneration;
```

```
-- PMC-Sierra, Inc.
-- PROPRIETARY AND CONFIDENTIAL
--
-- Copyright 2000 PMC-Sierra, Inc.
--
-- All rights reserved. No part of this documentation or computer program may
-- be used, modified, reproduced, or distributed in any form by any means
-- without the prior written permission of PMC-Sierra, Inc.
--
-- This documentation and computer program contains trade secrets,
confidential
-- business information and commercial or financial information (collectively,
-- the "information") of PMC-Sierra, Inc., or unlawful disclosure of any or
all
-- of the information may cause irreparable harm and result in significant
-- commercial and competitive loss to PMC-Sierra, Inc.
-- ------------------------------------------------------------------------
-- ------------------------------------------------------------------------
-- PMC-Sierra, Inc.
-- 105 - 8555 Baxter Place
-- Burnaby, B.C.
-- Canada V5A 4V7
-- Tel: 604-415-6000
-- Fax: 604-415-6206
-- email: apps@pmc-sierra.com
-- ------------------------------------------------------------------------
-- Project     :PMC-990887
-- File Name   :clocking.vhd
-- Path        :
-- Designer    :KM
--
-- Revision History
-- Issue       Date       Initial Descriptions
-- 1           08/27/99   KM        Preliminary
--
-- Function
-- This is a VHDL code for the ASAP-CES Reference Design.
-- This code generates clock signals for SPECTRA-155, TEMUXes,and AAL1gator-
32s
-- ------------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity CLKgeneration is
  port
  (
-- input clocks to the CPLD: 51.84 MHz, 37.056 MHz, and 19.44 MHz
   clk52  : in std_logic;
   clk37  : in std_logic;
   clk19  : in std_logic;
-- CPLD outputs:
   CLK_RRCLK  : out std_logic; -- to SPECTRA-155
   CLK_TRCLK  : out std_logic; -- to SPECTRA-155
   CLK_DCK    : out std_logic; -- to SPECTRA-155
   CLK_ACK    : out std_logic; -- to SPECTRA-155
   CLK_LREFCLK : out std_logic; -- to TEMUXes
```

```
    CLK_SREFCLK : out std_logic; -- to TEMUXes and AAL1gator-32s
    CLK_CLK52M : out std_logic; -- to TEMUXes
    CLK_FASTCLK : out std_logic; -- to AAL1gator-32s
    CLK_XCLK   : out std_logic  -- to TEMUXes
  );
end CLKgeneration;
architecture  CLKgeneration of CLKgeneration is
begin
     CLK_RRCLK <= clk19;
     CLK_TRCLK <= clk19;
     CLK_DCK <= clk19;
     CLK_ACK <= clk19;
     CLK_LREFCLK <= clk19;
     CLK_SREFCLK <= clk19;
     CLK_CLK52M <= clk52;
     CLK_FASTCLK <= clk52;
     CLK_XCLK <= clk37;
end CLKgeneration;
```

```
-- PMC-Sierra, Inc.
-- PROPRIETARY AND CONFIDENTIAL
--
-- Copyright 2000 PMC-Sierra, Inc.
--
-- All rights reserved. No part of this documentation or computer program may
-- be used, modified, reproduced, or distributed in any form by any means
-- without the prior written permission of PMC-Sierra, Inc.
--
-- This documentation and computer program contains trade secrets,
confidential
-- business information and commercial or financial information (collectively,
-- the "information") of PMC-Sierra, Inc., or unlawful disclosure of any or
all
-- of the information may cause irreparable harm and result in significant
-- commercial and competitive loss to PMC-Sierra, Inc.
-- ------------------------------------------------------------------------
-- ------------------------------------------------------------------------
-- PMC-Sierra, Inc.
-- 105 - 8555 Baxter Place
-- Burnaby, B.C.
-- Canada V5A 4V7
-- Tel: 604-415-6000
-- Fax: 604-415-6206
-- email: apps@pmc-sierra.com
-- ------------------------------------------------------------------------
-- Project     :PMC-990887
-- File Name   :framepulse.vhd
-- Path        :
-- Designer    :KM
--
-- Revision History
-- Issue       Date       Initial Descriptions
-- 1           08/27/99   KM        Preliminary
--
-- Function
-- This is a VHDL code for the ASAP-CES Reference Design.
-- This code generates frame pulses for SPECTRA-155, TEMUX,
-- and AAL1gator-32 devices for the Telecombus and SBI interfaces.
-- The generated frame pulses are 2 kHz.
-- ------------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity FPgenerator is
  port
  (
   clk19  : in std_logic; -- input clock to the CPLD: 19.44 MHz
   RESET  : in std_logic;
   count  : out std_logic_vector (13 downto 0);

-- CPLD outputs
   frame_DFP    : out std_logic; -- frame pulse to SPECTRA-155
   frame_C1FP   : out std_logic; -- frame pulse to TEMUXes and AAL1gator-32s
```

```
   frame_LAC1   : out std_logic  -- frame pulse to TEMUXes
  );
end FPgenerator;

architecture  FPgenerator_arch of FPgenerator is
begin
    process(clk19)
      variable count_int : std_logic_vector (13 downto 0);
    begin
-- 2 kHz frame pulses are sampled at rising edge of the 19.44 MHz clock and
-- happens every 9720 pulses.
      if clk19 = '1' and clk19'event then
        if RESET = '1' then
           count_int := "00000000000000";
           frame_C1FP <= '0';
           frame_LAC1 <= '0';
           frame_DFP <= '0';
        else
          if count_int = 9719 then
            frame_C1FP <= '1';
            frame_LAC1 <= '1';
            frame_DFP <= '1';
            count_int := (others => '0');
            count <= count_int;
          else
            count_int := count_int + 1;
            frame_C1FP <= '0';
            frame_LAC1 <= '0';
            frame_DFP <= '0';
            count <= count_int;
          end if;
        end if;
      end if;
    end process;
end FPgenerator_arch;
```

```
-- PMC-Sierra, Inc.
-- PROPRIETARY AND CONFIDENTIAL
--
-- Copyright 2000 PMC-Sierra, Inc.
--
-- All rights reserved. No part of this documentation or computer program may
-- be used, modified, reproduced, or distributed in any form by any means
-- without the prior written permission of PMC-Sierra, Inc.
--
-- This documentation and computer program contains trade secrets,
confidential
-- business information and commercial or financial information (collectively,
-- the "information") of PMC-Sierra, Inc., or unlawful disclosure of any or
all
-- of the information may cause irreparable harm and result in significant
-- commercial and competitive loss to PMC-Sierra, Inc.
-- ------------------------------------------------------------------------
-- ------------------------------------------------------------------------
-- PMC-Sierra, Inc.
-- 105 - 8555 Baxter Place
-- Burnaby, B.C.
-- Canada V5A 4V7
-- Tel: 604-415-6000
-- Fax: 604-415-6206
-- email: apps@pmc-sierra.com
-- ------------------------------------------------------------------------
-- Project     :PMC-990887
-- File Name   :interrupt.vhd
-- Path        :
-- Designer    :KM
--
-- Revision History
-- Issue       Date       Initial Descriptions
-- 1           10/13/99   KM        Preliminary
--
-- Function
-- This is a VHDL code for the ASAP-CES Reference Design.
-- This code generates the L_INTB signal from the INTB signals of SPECTRA-155,
-- TEMUX , APEX, ATLAS and AAL1gator-32 devices.
-- ------------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity INTBgeneration is
  port
  (
-- input interrput signals to the CPLD
   SPECTRA_INTB  : in std_logic;
   TE1_INTB  : in std_logic;
   TE2_INTB  : in std_logic;
   TE3_INTB  : in std_logic;
   AAL1_INTB  : in std_logic;
   AAL2_INTB  : in std_logic;
   AAL3_INTB  : in std_logic;
   APEX_INTLOB  : in std_logic;
   APEX_INTHIB  : in std_logic;
```

```
   ATLAS_INTB  : in std_logic;
--  reset signal
   RSTB   : inout std_logic;
-- output  signal:
   L_INTB    : out std_logic
  );
end INTBgeneration;
architecture  INTBgeneration of INTBgeneration is
begin
    process(RSTB, SPECTRA_INTB, TE1_INTB, TE2_INTB, TE3_INTB, AAL1_INTB,
AAL2_INTB, AAL3_INTB,
            APEX_INTLOB, APEX_INTHIB, ATLAS_INTB)
    begin
      if RSTB = '0'  then
        L_INTB <= '1';
      else
        L_INTB <= (SPECTRA_INTB and TE1_INTB and TE2_INTB and TE3_INTB and
AAL1_INTB and AAL2_INTB
                   and AAL3_INTB and APEX_INTLOB and APEX_INTHIB and
ATLAS_INTB);
      end if;
    end process;
end INTBgeneration;
```

```
-- PMC-Sierra, Inc.
-- PROPRIETARY AND CONFIDENTIAL
--
-- Copyright 2000 PMC-Sierra, Inc.
--
-- All rights reserved. No part of this documentation or computer program may
-- be used, modified, reproduced, or distributed in any form by any means
-- without the prior written permission of PMC-Sierra, Inc.
--
-- This documentation and computer program contains trade secrets,
confidential
-- business information and commercial or financial information (collectively,
-- the "information") of PMC-Sierra, Inc., or unlawful disclosure of any or
all
-- of the information may cause irreparable harm and result in significant
-- commercial and competitive loss to PMC-Sierra, Inc.
-- ------------------------------------------------------------------------
-- ------------------------------------------------------------------------
-- PMC-Sierra, Inc.
-- 105 - 8555 Baxter Place
-- Burnaby, B.C.
-- Canada V5A 4V7
-- Tel: 604-415-6000
-- Fax: 604-415-6206
-- email: apps@pmc-sierra.com
-- ------------------------------------------------------------------------
-- Project     :PMC-990887
-- File Name   :networkclock.vhd
-- Path        :
-- Designer    :KM
--
-- Revision History
-- Issue       Date       Initial Descriptions
-- 1           10/08/99   KM
--


-- Function:
-- This is a VHDL code for the AAL1gator-32 Reference Design.
-- This code generates NCLK clock signal for the AAL1gator-32 devices from
-- AAL_SYSCLK (38.88 MHz).
-- This code divides the AAL_SYSCLK by 16 to generate the NCLK clock.
-- The generated clock is  2.43 MHz.
-- ------------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;
entity NetClkgeneration is
  port
  (
-- CPLD inputs:
-- input clock to the CPLD: 38.88 MHz
   aal_sysclk  : in std_logic;
-- input reset signal
   RSTB  : in std_logic;
```

```
-- CPLD outputs:
   nclk  : out  std_logic -- to AAL1gator-32s
  );
end NetClkgeneration;
architecture  NetClkgeneration  of  NetClkgeneration is
  constant max_count: integer := 8;
  signal divided_clk : std_logic := '0';
begin

    process(aal_sysclk, RSTB)
    variable count: integer:= 0;
    begin
      if (RSTB = '0') then
        divided_clk <= '0';
        count := 0;
      elsif (aal_sysclk'event and aal_sysclk = '1') then
        count := count + 1;
        if (count = max_count) then
          divided_clk <= not divided_clk;
          count := 0;
        end if;
      end if;
    end process;
    nclk <= divided_clk;
end  NetClkgeneration;
```

12 APPENDIX C: EXTERNAL DS3 SRTS VHDL CODE

```
--
--     Copyright (C) 1995-1998. PMC-Sierra, Inc. All Rights Reserved.
--
--
-- DESIGN TITLE: AAL1GATOR32 DS3 SRTS
-- FILE NAME: a32-v1.vhd
--
-- DETAILED REVISION HISTORY
--
-- ISSUE    DATE(YY/MM/DD)   NAME        DESCRIPTION
-- 00       00/06/12                          creation
--
--------------------------------------------------------------------------------
-
--
-- FUNCTION:  AAL1GATOR-32 External Implementation for DS3 SRTS
--            Features Include:
--                - Interaface to the A-32 SRTS strobe and data signals
--                - Mapping the 4 bits SRTS code to 8 bits D/A converter data
--                - Generates DAC write pulse
--------------------------------------------------------------------------------
---
--
--Synthesis Options:
--    FSM Synthesis: Binary
--    FSM synthesis: Interpretation of VHDL 'when others' : Fasterst &
smallest
--    Export schematics to Vhdl
--    Input XNF Bus Style: %s<%d>
--    Default Frequency: 16
--
-- Synthesis/implementation
--    Top Level: AAL1GATOR
--    Family XC4000XLA
--    Device XC4044XLA-09HQ304C
--    Speed Grade -09
--    Target Clock Frequency: 16
--          optimize for: Speed
--    Effort: High
--    Insert I/O pads
--
-- Implementation Options
--    User Constraints: c:\
```

```
--    Implemntation: Optimize Speed
--          Simulation: Generic Vhdl
--    Produce Configuration Data
--
--  XC4000 Implemation Options:
--  Basic:
--  Y  Use Global Clocks
--  Y  Use Global Output enables
--  Y  Use Global Set/Reset
--  Y  Use Timing constraints
--  Y  Use Design Location Constraints
--
--    Macrocell Power setting: Std
--    Output Slew Rate: Fast
--  Advanced:
--  N  Use Timing Optimization
--  Y  Use Multi-level Logic Optimization
--  Y  Use Advanced fitting
--  Y  Enable D <--> T Type Transform Optimization
--
--     Collapsing Pterm Limit: 20
--     Collapsing Input Limit (XC9500XL only): 36
--
--  Programming:
--    Signature / User  Code: use design name
--------------------------------------------------------------------------------
-
-- design files
--
--  a32-v1.vhd:   top level (this file)
--  a32-v1.ucf:   user contraint file (timing and pin out)
--  tbd.vhd:            test bench
--
--
-- LIBRARIES
--
lIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;
USE IEEE.STD_LOGIC_arith.all;
USE IEEE.STD_LOGIC_unsigned.all;


-- pragma translate_off
LIBRARY unisim;
USE unisim.Vcomponents.all;
-- pragma translate_on
```

```
--EJECT_
--------------------------------------------------------------------------------
-
--
-- Entity               : ds3srts
--
-- Author(s)            :
-- Organization         : PMC-Sierra
--
-- Project              : AAL1GATOR-32 DS3 SRTS FPGA
--
-- Creation Date        : 000612
--
-- Description          :
--
--------------------------------------------------------------------------------
-
ENTITY ds3srts IS

  PORT
  (

   sys_clk:  IN   STD_LOGIC;
   brd_rstb:     IN     STD_LOGIC;
   rstb:         IN     STD_LOGIC;
   rstb_out:     OUT    STD_LOGIC;


   cgc_dout:       IN   STD_LOGIC_VECTOR(3 downto 0);
   cgc_line:       IN   STD_LOGIC_VECTOR(4 downto 0);
   srts_stbh:      IN   STD_LOGIC;

   dac_csb:  OUT  STD_LOGIC;
   dac_wrb:  OUT  STD_LOGIC;
   dac_abb:  OUT  STD_LOGIC;
   dac_dout:       OUT  STD_LOGIC_VECTOR(7 downto 0);

-- test stuff
   test_in:      IN     STD_LOGIC;

   test_rstb:    OUT    STD_LOGIC;
   test_cgcstbh: OUT    STD_LOGIC;
   test_cgcdout: OUT    STD_LOGIC_VECTOR(3 downto 0);
   test_cgcline: OUT    STD_LOGIC_VECTOR(4 downto 0);
```

```
   test_csb:       OUT  STD_LOGIC;
   test_wrb:       OUT  STD_LOGIC;
   test_abb:       OUT  STD_LOGIC;
-- changed to input
   test_dout:      OUT  STD_LOGIC_VECTOR(7 downto 0)

);

END ds3srts;

ARCHITECTURE rtl OF ds3srts IS
--This device is used on the DS3 SRTS FTP.
--It is derived from the AAL1GATOR-2 DS3 board design.
--It implements the a/d bus controls, as well as translating
--the 4 bit serial SRTS code to an 8 bit code for the D/A.

--  6/12/00 Initial construction
--
type statetype is (IDLE, S1, S2, S3, S4, S5);


  --
  -- CONSTANTS/SIGNAL DECLARATIONS
  --
  -- registered input
    SIGNAL cgc_dout_reg  : STD_LOGIC_VECTOR(3 downto 0);
    SIGNAL cgc_line_reg  : STD_LOGIC_VECTOR(4 downto 0);
    SIGNAL cgc_dout_reg2 : STD_LOGIC_VECTOR(3 downto 0);
    SIGNAL cgc_line_reg2 : STD_LOGIC_VECTOR(4 downto 0);
    SIGNAL srts_stbh_reg : STD_LOGIC;

    SIGNAL dac_d         : STD_LOGIC_VECTOR(7 downto 0);
    SIGNAL nextstate     : statetype;
    SIGNAL test_dout_reg : STD_LOGIC_VECTOR(7 downto 0);
    SIGNAL clk           : STD_LOGIC;
--   SIGNAL rst           : STD_LOGIC;


COMPONENT BUFG
    PORT (i: IN STD_LOGIC; o : OUT STD_LOGIC);
  END COMPONENT;

--component STARTUP
--    port (GSR: in std_logic);
--  end component;

component INV
```

```
         port (I: in  STD_LOGIC;
               O: out STD_LOGIC);
end component;


  BEGIN
  rstb_out <= rstb;

--  dac_dout <= test_dout;

--clk_buf: BUFG
--  PORT MAP(i => sys_clk, o => clk);

--U1: STARTUP  port map (GSR=>rst);

--U2: INV port map(I => brd_rstb, O => rst);

   inp_data_latch: PROCESS(sys_clk, brd_rstb, cgc_dout, cgc_line, srts_stbh)
    BEGIN
      IF (brd_rstb = '0') THEN
         cgc_dout_reg <= "0000";
         cgc_line_reg <= "00000";
         cgc_dout_reg2 <= "0000";
         cgc_line_reg2 <= "00000";
         srts_stbh_reg <= '0';
  test_rstb <= '0';
  test_cgcstbh <= '1';
  test_cgcline <= "00000";
  test_cgcdout <= "0000";
  test_dout_reg <= "10000001";
      ELSIF (sys_clk = '1') and (sys_clk'EVENT) THEN
  test_rstb <= brd_rstb;
  test_cgcstbh <= srts_stbh;
-- changed to get a more stable cgc_line test output
  test_cgcline <= cgc_line_reg;
  test_cgcdout <= cgc_dout;
--  test_dout_reg <= test_dout;
--  test_cgcline <= "00000";

         srts_stbh_reg <= srts_stbh;
         cgc_dout_reg <= cgc_dout;
         cgc_line_reg <= cgc_line;
--             cgc_line_reg <= "00000";

         IF srts_stbh_reg = '1' THEN
             cgc_dout_reg2 <= cgc_dout_reg;
                 cgc_line_reg2 <= cgc_line_reg;
```

```
         END IF;
       END IF;
    END PROCESS inp_data_latch;

   statemc: PROCESS(sys_clk, brd_rstb, nextstate, srts_stbh_reg,
cgc_line_reg2)
    BEGIN
      IF (brd_rstb = '0') THEN
         nextstate <= IDLE;
      ELSIF (sys_clk = '1') and (sys_clk'EVENT) THEN
           CASE nextstate IS
              WHEN IDLE =>
                IF srts_stbh_reg = '0' THEN
                  nextstate <= IDLE;
                ELSE
                  nextstate <= S1;
                END IF;
              WHEN S1 =>
                nextstate <= S2;
              WHEN S2 =>
              -- changed to uncondition S3, 'cause I don't know the line
number
--                IF cgc_line_reg2 = "00000" OR cgc_line_reg2 = "00010" THEN
                 nextstate <= S3;
--                ELSE
--                 nextstate <= IDLE;
--                END IF;
              WHEN S3 =>
                nextstate <= S4;
              WHEN S4 =>
                nextstate <= S5;
              WHEN S5 =>
                nextstate <= IDLE;
              WHEN OTHERS =>
                nextstate <= IDLE;
            END CASE;
      END IF;
    END PROCESS statemc;

   convert_proc: PROCESS(sys_clk, brd_rstb, cgc_dout_reg2)
    BEGIN
      IF (brd_rstb = '0') THEN
         dac_d <= "00000000";
      ELSIF (sys_clk = '1') and (sys_clk'EVENT) THEN
         CASE cgc_dout_reg2 IS
--            WHEN "0000" => dac_d <= "10000010";    -- 80
--            WHEN "0001" => dac_d <= "10001101";    -- 8D
```

```
--            WHEN "0010" => dac_d <= "10011010";    -- 9A
--            WHEN "0011" => dac_d <= "10100110";    -- A6
--            WHEN "0100" => dac_d <= "10110011";    -- B3
--            WHEN "0101" => dac_d <= "11000000";    -- C0
--            WHEN "0110" => dac_d <= "11001101";    -- CD
--            WHEN "0111" => dac_d <= "11100111";    -- E7
--            WHEN "1000" => dac_d <= "00011001";    -- 19
--            WHEN "1001" => dac_d <= "00100110";    -- 26
--            WHEN "1010" => dac_d <= "00110011";    -- 33
--            WHEN "1011" => dac_d <= "01000000";    -- 40
--            WHEN "1100" => dac_d <= "01001101";    -- 4D
--            WHEN "1101" => dac_d <= "01011010";    -- 5A
--            WHEN "1110" => dac_d <= "01100110";    -- 66
--            WHEN "1111" => dac_d <= "01110011";    -- 73
--            WHEN others => dac_d <= "10000100";    -- 80
            WHEN "1000" => dac_d <= "00011010";    -- 19 -> 1A
            WHEN "1001" => dac_d <= "00100111";    -- 26 -> 27
            WHEN "1010" => dac_d <= "00110100";    -- 33 -> 34
            WHEN "1011" => dac_d <= "01000000";    -- 40 -> 40
            WHEN "1100" => dac_d <= "01001101";    -- 4D -> 4D
            WHEN "1101" => dac_d <= "01011010";    -- 5A -> 5A
            WHEN "1110" => dac_d <= "01100110";    -- 66 -> 66
            WHEN "1111" => dac_d <= "01110011";    -- 73 -> 73
            WHEN "0000" => dac_d <= "10000000";    -- 80 -> 80
            WHEN "0001" => dac_d <= "10001001";    -- 8D -> 89
            WHEN "0010" => dac_d <= "10010011";    -- 9A -> 93
            WHEN "0011" => dac_d <= "10011100";    -- A6 -> 9C
            WHEN "0100" => dac_d <= "10100110";    -- B3 -> A6
            WHEN "0101" => dac_d <= "10101111";    -- C0 -> AF
            WHEN "0110" => dac_d <= "10111001";    -- CD -> B9
            WHEN "0111" => dac_d <= "11000010";    -- E7 -> C2
            WHEN others => dac_d <= "10000000";    -- 80

        END CASE;
      END IF;
    END PROCESS convert_proc;

   oup_data_latch: PROCESS(sys_clk, brd_rstb, dac_d, test_in, test_dout_reg)
    BEGIN
      IF (brd_rstb = '0') THEN
         test_dout <= "10000000";
           dac_dout  <= "10000000";
      ELSIF (sys_clk = '1') and (sys_clk'EVENT) THEN
         test_dout <= dac_d;
         dac_dout  <= dac_d;
--           IF (test_in = '1') THEN
--            dac_dout <= test_dout_reg;
```

```
--            ELSE
--             dac_dout  <= dac_d;
--            END IF;
         END IF;
     END PROCESS oup_data_latch;

    oup_ctl: PROCESS(sys_clk, brd_rstb, dac_d, cgc_line_reg2, nextstate)
     BEGIN
       IF (brd_rstb = '0') THEN
          dac_csb <= '1';
          dac_wrb <= '1';
          dac_abb <= '0';
          test_csb <= '1';
          test_wrb <= '1';
          test_abb <= '0';
       ELSIF (sys_clk = '1') and (sys_clk'EVENT) THEN
          IF (nextstate = S3 or nextstate = S5) THEN
            dac_csb <= '0';
              test_csb <= '0';
              dac_wrb <= '1';
              test_wrb <= '1';
            ELSIF nextstate = S4 and cgc_line_reg2 = "00000" THEN
              dac_csb <= '0';
              dac_wrb <= '0';
              test_csb <= '0';
              test_wrb <= '0';
              dac_abb <= '0';
              test_abb <= '0';
            ELSIF nextstate = S4 and cgc_line_reg2 /= "00000" THEN
                dac_csb <= '0';
                dac_wrb <= '0';
                test_csb <= '0';
                test_wrb <= '0';
                dac_abb <= '0';
                test_abb <= '0';
                dac_abb <= '1';
                test_abb <= '1';
            ELSE
                dac_csb <= '1';
                dac_wrb <= '1';
                dac_abb <= '0';
                test_csb <= '1';
                test_wrb <= '1';
                test_abb <= '0';
            END IF;
        END IF;
     END PROCESS oup_ctl;
```

```
END;
```

13 APPENDIX D: EXTERNAL DS3 ADAPTIVE CLOCK RECOVERY VHDL CODE

```
--------------------------------------------------------------------------------
--     Copyright (C) 1995-1998. PMC-Sierra, Inc. All Rights Reserved.
--
-- DESIGN TITLE: AAL1GATOR-32 DS3 Adaptive Clock Recovery
--
-- REVISION HISTORY
-- ISSUE    DATE(YY/MM/DD)   DESCRIPTION
-- 00       2000/06/12       creation
-- 01       2001/10/02       changed nominal DS3 buffer depth from
--                           00000001000000 (64 bytes/R_CDVT=2/11 us)
--                           to 01010111000000 (5568 bytes/R_CDVT=174/996 us)
--                           because 11 us is too small, and 996 us seems
--                           more realistic.
--------------------------------------------------------------------------------
-- FUNCTION:  AAL1GATOR-32 External FPGA Implementation for DS3 Adaptive Clock
--            Recovery.
--            Features:
--                - Interfaces to the AAL1gator-32 CGC port.
--                - Captures DS3 Buffer Underrun status and buffer depth for
--                  two DS3 signals.
--                - Converts DS3 Buffer Underrun status and depth to an 8-bit
--                  DAC code (Analog Devices AD7302 Dual Voltage Output 8-bit
--                  DAC).
--                - Generates DAC write access to either DAC A or DAC B of the
--                  AD7302.
--
-- SUMMARY OF PROCESSES:
--   inp_data_latch: latch CGC port outputs CGC_DOUT, CGC_LINE, and ADAP_STBH.
--   statemc: state machine follows CGC Channel Status(S1-S4) and CGC Buffer
--            Depth (D1-D6) playout states, and generates states (R1-R4) to
--            enable generation of DAC write accesses.
--   line_dout: captures DS3 line number (line_code) and DS3 Buffer depth
--              (buf_depth) from the CGC port. If the DS3 Buffer is in
--             underrun, then the buf_depth is set to nominal.
--   convert_proc: converts the raw DS3 Buffer depth to an AD7302 DAC code.
--   oup_data_latch: latches DAC code.
--   oup_ctl: generates the DAC write access control signals, based on
--            statemc states.
--------------------------------------------------------------------------------
---
--
--Synthesis Options:
--    FSM Synthesis: Binary
```

```
--    FSM synthesis: Interpretation of VHDL 'when others' : Fasterst &
smallest
--    Export schematics to Vhdl
--    Input XNF Bus Style: %s<%d>
--    Default Frequency: 16
--
-- Synthesis/implementation
--    Top Level: AAL1GATOR
--    Family XC4000XLA
--    Device XC4044XLA-09HQ304C (note: this device contained other logic in
--           addition to this DS3 Adaptive Clock Recovery logic.)
--    Speed Grade -09
--    Target Clock Frequency: 16
--          optimize for: Speed
--    Effort: High
--    Insert I/O pads
--
-- Implementation Options
--    User Constraints: c:\
--    Implemntation: Optimize Speed
--          Simulation: Generic Vhdl
--    Produce Configuration Data
--
--  XC4000 Implementation Options:
--  Basic:
--  Y  Use Global Clocks
--  Y  Use Global Output enables
--  Y  Use Global Set/Reset
--  Y  Use Timing constraints
--  Y  Use Design Location Constraints
--
--    Macrocell Power setting: Std
--    Output Slew Rate: Fast
--  Advanced:
--  N  Use Timing Optimization
--  Y  Use Multi-level Logic Optimization
--  Y  Use Advanced fitting
--  Y  Enable D <--> T Type Transform Optimization
--
--     Collapsing Pterm Limit: 20
--     Collapsing Input Limit (XC9500XL only): 36
--
--  Programming:
--    Signature / User  Code: use design name
----------------------------------------------------------------------------
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;
```

```
USE IEEE.STD_LOGIC_arith.all;
USE IEEE.STD_LOGIC_unsigned.all;

-- pragma translate_off
LIBRARY unisim;
USE unisim.Vcomponents.all;
-- pragma translate_on
----------------------------------------------------------------------------
-- Entity               : ds3adap
----------------------------------------------------------------------------
ENTITY ds3adap IS
PORT(
   sys_clk:  IN   STD_LOGIC;
   brd_rstb:     IN     STD_LOGIC;
   rstb:         IN     STD_LOGIC;

   cgc_dout:       IN   STD_LOGIC_VECTOR(3 downto 0);
   cgc_line:       IN   STD_LOGIC_VECTOR(4 downto 0);
   adap_stbh:      IN   STD_LOGIC;

   dac_csb:  OUT  STD_LOGIC;
   dac_wrb:  OUT  STD_LOGIC;
   dac_abb:  OUT  STD_LOGIC;
   dac_dout:       OUT  STD_LOGIC_VECTOR(7 downto 0);

   rstb_out:     OUT    STD_LOGIC;
-- test stuff
   test_in:      IN     STD_LOGIC;

   test_rstb:    OUT    STD_LOGIC;
   test_cgcstbh: OUT    STD_LOGIC;
   test_cgcdout: OUT    STD_LOGIC_VECTOR(3 downto 0);
   test_cgcline: OUT    STD_LOGIC_VECTOR(4 downto 0);


   test_csb:       OUT  STD_LOGIC;
   test_wrb:       OUT  STD_LOGIC;
   test_abb:       OUT  STD_LOGIC;
-- changed to input
   test_depth2_0 : OUT  STD_LOGIC_VECTOR(2 downto 0);
   test_dout:      OUT  STD_LOGIC_VECTOR(7 downto 0)

);

END ds3adap;

ARCHITECTURE rtl OF ds3adap IS
```

```
type statetype is (IDLE, D1, D2, D3, D4, D5, D6, S1, S2, S3, S4, R1, R2, R3,
R4);
  --
  -- CONSTANTS/SIGNAL DECLARATIONS
  --
  -- registered input
    SIGNAL cgc_dout_reg  : STD_LOGIC_VECTOR(3 downto 0);
    SIGNAL cgc_line_reg  : STD_LOGIC_VECTOR(4 downto 0);
    SIGNAL cgc_dout_reg2 : STD_LOGIC_VECTOR(3 downto 0);
--    SIGNAL cgc_line_reg2 : STD_LOGIC_VECTOR(4 downto 0);
    SIGNAL adap_stbh_reg : STD_LOGIC;

    SIGNAL dac_d         : STD_LOGIC_VECTOR(7 downto 0);
    SIGNAL nextstate     : statetype;
    SIGNAL test_dout_reg : STD_LOGIC_VECTOR(7 downto 0);
    SIGNAL clk           : STD_LOGIC;
    SIGNAL buf_depth     : STD_LOGIC_VECTOR(13 downto 0);
    SIGNAL line_code     : STD_LOGIC_VECTOR(4 downto 0);

--   SIGNAL rst           : STD_LOGIC;


COMPONENT BUFG
    PORT (i: IN STD_LOGIC; o : OUT STD_LOGIC);
  END COMPONENT;

--component STARTUP
--    port (GSR: in std_logic);
--  end component;

component INV
         port (I: in  STD_LOGIC;
               O: out STD_LOGIC);
end component;


  BEGIN


--  dac_dout <= test_dout;

--clk_buf: BUFG
--  PORT MAP(i => sys_clk, o => clk);

--U1: STARTUP  port map (GSR=>rst);
```

```
--U2: INV port map(I => brd_rstb, O => rst);

    rstb_out <= rstb;

   inp_data_latch: PROCESS(sys_clk, brd_rstb, cgc_dout, cgc_line, adap_stbh)
    BEGIN
      IF (brd_rstb = '0') THEN
         cgc_dout_reg <= "0000";
         cgc_line_reg <= "00000";
         cgc_dout_reg2 <= "0000";
--             cgc_line_reg2 <= "00000";
         adap_stbh_reg <= '0';
  test_rstb <= '0';
  test_cgcstbh <= '1';
  test_cgcline <= "00000";
  test_cgcdout <= "0000";
  test_dout_reg <= "10000001";
      ELSIF (sys_clk = '1') and (sys_clk'EVENT) THEN
  test_rstb <= brd_rstb;
  test_cgcstbh <= adap_stbh;
  test_cgcline <= cgc_line;
  test_cgcdout <= cgc_dout;
--  test_dout_reg <= test_dout;
  test_cgcline <= "00000";

         adap_stbh_reg <= adap_stbh;
         cgc_dout_reg <= cgc_dout;
         cgc_line_reg <= cgc_line;

         IF adap_stbh_reg = '1' THEN
             cgc_dout_reg2 <= cgc_dout_reg;
--               cgc_line_reg2 <= cgc_line_reg;
         END IF;
      END IF;
    END PROCESS inp_data_latch;


   statemc: PROCESS(sys_clk, brd_rstb, nextstate, adap_stbh_reg, cgc_line_reg)
    BEGIN
      IF (brd_rstb = '0') THEN
         nextstate <= IDLE;
      ELSIF (sys_clk = '1') and (sys_clk'EVENT) THEN
           CASE nextstate IS
              WHEN IDLE =>
                IF adap_stbh_reg = '0' THEN
                  nextstate <= IDLE;
                ELSIF cgc_line_reg = "00101" THEN
```

```
        nextstate <= D1;
      ELSIF cgc_line_reg = "01111" THEN
        nextstate <= S1;
      ELSE
        nextstate <= IDLE;
      END IF;
    WHEN D1 =>
      IF adap_stbh_reg = '0' THEN
        nextstate <= IDLE;
      ELSIF cgc_line_reg = "00100" THEN
        nextstate <= D2;
      ELSE
        nextstate <= IDLE;
      END IF;
    WHEN D2 =>
      IF adap_stbh_reg = '0' THEN
        nextstate <= IDLE;
      ELSIF cgc_line_reg = "00011" THEN
        nextstate <= D3;
      ELSE
        nextstate <= IDLE;
      END IF;
    WHEN D3 =>
      IF adap_stbh_reg = '0' THEN
        nextstate <= IDLE;
      ELSIF cgc_line_reg = "00010" THEN
        nextstate <= D4;
      ELSE
        nextstate <= IDLE;
      END IF;
    WHEN D4 =>
      IF adap_stbh_reg = '0' THEN
        nextstate <= IDLE;
      ELSIF cgc_line_reg = "00001" THEN
        nextstate <= D5;
      ELSE
        nextstate <= IDLE;
      END IF;
    WHEN D5 =>
      IF adap_stbh_reg = '0' THEN
        nextstate <= IDLE;
      ELSIF cgc_line_reg = "00000" THEN
        nextstate <= D6;
      ELSE
        nextstate <= IDLE;
      END IF;
    WHEN D6 =>
```

```
                nextstate <= R1;

            WHEN S1 =>
              IF adap_stbh_reg = '0' THEN
                nextstate <= IDLE;
              ELSIF cgc_line_reg = "01110" THEN
                nextstate <= S2;
              ELSE
                nextstate <= IDLE;
              END IF;
            WHEN S2 =>
              IF adap_stbh_reg = '0' THEN
                nextstate <= IDLE;
              ELSIF cgc_line_reg = "01101" THEN
                nextstate <= S3;
              ELSE
                nextstate <= IDLE;
              END IF;
            WHEN S3 =>
              IF adap_stbh_reg = '0' THEN
                nextstate <= IDLE;
              ELSIF cgc_line_reg = "01100" THEN
                nextstate <= S4;
              ELSE
                nextstate <= IDLE;
              END IF;
            WHEN S4 =>
                nextstate <= R1;

            WHEN R1 =>
                nextstate <= R2;
            WHEN R2 =>
                nextstate <= R3;
            WHEN R3 =>
                nextstate <= R4;
            WHEN R4 =>
                nextstate <= IDLE;
            WHEN OTHERS =>
                nextstate <= IDLE;
          END CASE;
    END IF;
  END PROCESS statemc;

 line_dout : PROCESS(sys_clk, brd_rstb, cgc_dout_reg2)
  BEGIN
      IF (brd_rstb = '0') THEN
        buf_depth <= "00000000000000";
```

```
            line_code <= "00000";
      ELSIF (sys_clk = '1') and (sys_clk'EVENT) THEN
         CASE nextstate IS
              WHEN D1 => line_code(2 downto 0) <= cgc_dout_reg2(3 downto 1);
              WHEN D3 => line_code(4 downto 3) <= cgc_dout_reg2(3 downto 2);
                         buf_depth(13 downto 12) <= cgc_dout_reg2(1 downto 0);
              WHEN D4 => buf_depth(11 downto 8) <= cgc_dout_reg2(3 downto 0);
              WHEN D5 => buf_depth(7 downto 4) <= cgc_dout_reg2(3 downto 0);
              WHEN D6 => buf_depth(3 downto 0) <= cgc_dout_reg2(3 downto 0);
              WHEN S1 => line_code(2 downto 0) <= cgc_dout_reg2(2 downto 0);
              WHEN S3 => IF cgc_dout_reg2 /= "0000" THEN
                           -- (original) buf_depth <= "00000001000000";
                           -- set nominal buffer depth, during underrun,
                           -- to 5568 bytes, corresponding to
                           -- R_CDVT=174=0x00AE (996 us)
                           -- note: nominal buf_depth is chosen such that
                           --   nominal buf_depth(6:3)=”1000” to simplify
                           --   logic, and tolerance to CDV is approximately
                           --   one millisecond.
                           buf_depth <= "01010111000000"; -- 0x15C0
                         END IF;
              WHEN S4 => line_code(4 downto 3) <= cgc_dout_reg2(1 downto 0);
              WHEN OTHERS => NULL;
         END CASE;
      END IF;
   END PROCESS line_dout;

  convert_proc: PROCESS(sys_clk, brd_rstb, buf_depth)
   BEGIN
      IF (brd_rstb = '0') THEN
         dac_d <= "00000000";
      ELSIF (sys_clk = '1') and (sys_clk'EVENT) THEN
         -- (original) IF buf_depth(13 downto 7) /= "0000000" THEN
         -- (original)   dac_d <= "11010101";                    --  D5
         IF buf_depth(13 downto 7) > "0101011" THEN
            -- buffer too full, speed up DS3 clock.
            dac_d <= "11010101";                    --  D5
         ELSIF buf_depth(13 downto 7) < "0101011" THEN
            -- buffer too empty, slow down DS3 clock.
            dac_d <= "00001101";                    --  0D
         ELSE
           -- buffer depth is in proper range of nominal.
           CASE buf_depth(6 downto 3) IS
              WHEN "1111" => dac_d <= "11001100";  --  CC
              WHEN "1110" => dac_d <= "11000010";  --  C2
              WHEN "1101" => dac_d <= "10111001";  --  B9
              WHEN "1100" => dac_d <= "10101111";  --  AF
```

```
            WHEN "1011" => dac_d <= "10100110";  --  A6
            WHEN "1010" => dac_d <= "10011100";  --  9C
            WHEN "1001" => dac_d <= "10010011";  --  93
            WHEN "1000" => dac_d <= "10000000";  --  80
            WHEN "0111" => dac_d <= "01110011";  --  73
            WHEN "0110" => dac_d <= "01100110";  --  66
            WHEN "0101" => dac_d <= "01011010";  --  5A
            WHEN "0100" => dac_d <= "01001101";  --  4D
            WHEN "0011" => dac_d <= "01000000";  --  40
            WHEN "0010" => dac_d <= "00110100";  --  34
            WHEN "0001" => dac_d <= "00100111";  --  27
            WHEN "0000" => dac_d <= "00011010";  --  1A
            WHEN others => dac_d <= "10000000";  -- 80
       END CASE;
    END IF;
    END IF;
 END PROCESS convert_proc;

oup_data_latch: PROCESS(sys_clk, brd_rstb, dac_d, buf_depth)
 BEGIN
    IF (brd_rstb = '0') THEN
       test_dout <= "10000000";
         dac_dout  <= "10000000";
         test_depth2_0 <= "000";
    ELSIF (sys_clk = '1') and (sys_clk'EVENT) THEN
       test_dout <= dac_d;
         dac_dout  <= dac_d;
         test_depth2_0 <= buf_depth(2 downto 0);
      END IF;
 END PROCESS oup_data_latch;

oup_ctl: PROCESS(sys_clk, brd_rstb, nextstate)
 BEGIN
    IF (brd_rstb = '0') THEN
       dac_csb <= '1';
       dac_wrb <= '1';
       dac_abb <= '0';
       test_csb <= '1';
       test_wrb <= '1';
       test_abb <= '0';
    ELSIF (sys_clk = '1') and (sys_clk'EVENT) THEN
       IF (nextstate = R1 or nextstate = R3) THEN
         dac_csb <= '0';
           test_csb <= '0';
           dac_wrb <= '1';
           test_wrb <= '1';
           IF line_code = "00000" THEN
```

```
                dac_abb <= '0';
                test_abb <= '0';
              ELSE
                dac_abb <= '1';
                test_abb <= '1';
              END IF;
            ELSIF (line_code = "00000" and (nextstate = R2 )) THEN
              dac_csb <= '0';
              dac_wrb <= '0';
              test_csb <= '0';
              test_wrb <= '0';
              dac_abb <= '0';
              test_abb <= '0';
            ELSIF (line_code /= "00000" and (nextstate = R2 )) THEN
                dac_csb <= '0';
                dac_wrb <= '0';
                test_csb <= '0';
                test_wrb <= '0';
                dac_abb <= '1';
                test_abb <= '1';
            ELSE
                dac_csb <= '1';
                dac_wrb <= '1';
                dac_abb <= '0';
                test_csb <= '1';
                test_wrb <= '1';
                test_abb <= '0';
            END IF;
        END IF;
    END PROCESS oup_ctl;
END;
```

NOTES

PMC-Sierra, Inc.

CONTACTING PMC-SIERRA, INC.

PMC-Sierra, Inc.
105-8555 Baxter Place Burnaby, BC
Canada V5A 4V7

Tel: (604) 415-6000

Fax: (604) 415-6200

Document Information: document@pmc-sierra.com
Corporate Information: info@pmc-sierra.com
Application Information: apps@pmc-sierra.com
(604) 415-4533
Web Site: http://www.pmc-sierra.com

PMC-1990887 (R4) Issue date: October 2001