DS31256DK
256-Channel, High-Throughput HDLC Controller Demonstration Kit

www.maxim-ic.com

GENERAL DESCRIPTION

The DS31256DK is a demonstration and evaluation kit for the DS31256 256-channel, high-throughput HDLC controller. The DS31256DK board is a PCI-based platform that offers quick and easy evaluation of the DS31256 HDLC controller, low-cost prototyping, and rapid software development. The DS31256DK operates with a software suite that runs under Microsoft Windows®95/98/2000/NT. The PC platform must be at least a 200MHz+ Pentium II class CPU with 32MB of RAM.

Windows is a registered trademark of Microsoft Corp.

ORDERING INFORMATION

PART	DESCRIPTION
DS31256DK	Demonstration Kit

FEATURES

- Headers Provide Access to Clock, Sync, and Data for 16 Physical Ports
- Full Source Code
- Windows/Windows NT GUI for Device Configuration and Evaluation
- Programmable PLD for Port Connectivity and Clock Generation
- Local Bus Header for Control and Configuration of External Devices
- Hardware Prototyping Area

DEMO KIT CONTENTS

CD-ROM Includes:

HDLC Application Notes

DS31256DK Data Sheet/Manual

Installation Guide

DS31256 Data Sheet

Executable Evaluation Application

Source Code for Driver and GUI Application

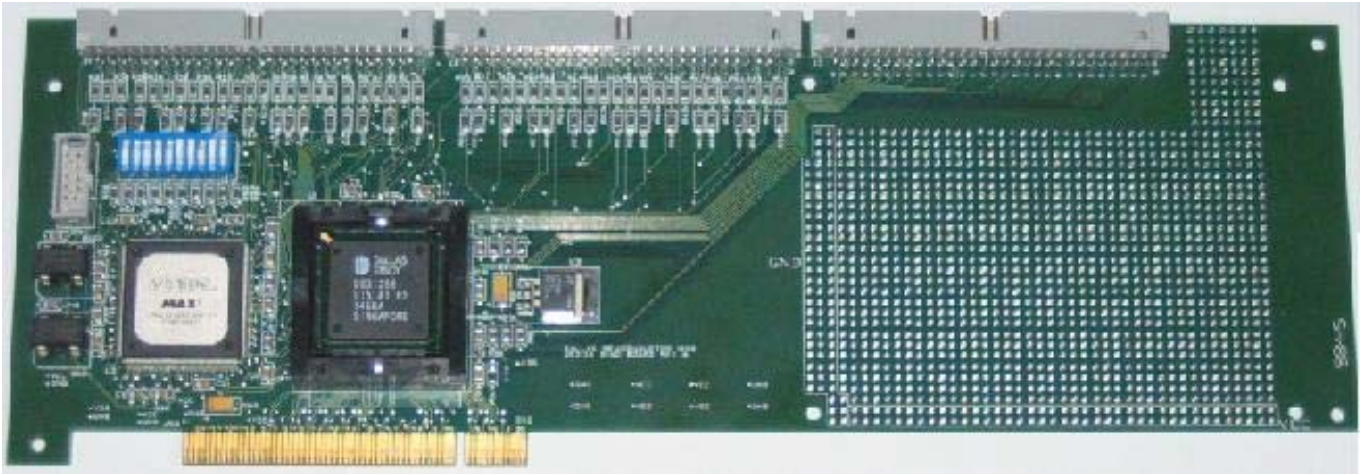


TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. GENERAL OVERVIEW

The DS31256DK is a demonstration and evaluation kit for the DS31256 256-channel, high-throughput HDLC controller. The DS31256DK is intended to be used in a full-size PC platform, complete with PCI. The DS31256DK operates with a software suite that runs under Microsoft Windows 95/98/2000/NT. The PC platform must be at least a 200MHz+ Pentium II class CPU with 32MB of RAM.

The DS31256DK was designed to be as simple as possible but offers the flexibility to be used in a number of different configurations. The DS31256DK has all the DS31256's port and local bus pins, which are easily accessible through headers on top of the card. Figure 1-1 details an outline of the PCI board for the DS31256DK.

A second DS31256DK can also be loaded into the PC in an adjacent PCI slot to add additional functions such as:

- Multiple T1/E1 framers
- T3 line interface
- HSSI interface
- V.35 interfaces

An Altera 9000-series PLD device is connected to all port pins on the DS31256. The PLD can be loaded with various configurations through a programming port (J4) that resides on the DS31256DK. This PLD generates clocks and frame syncs, as well as routes data from one port to another in a daisy-chain fashion to allow testing the device under worst-case loading (Figure 1-2). Two oscillators provide the port timing.

The transmit side of a port is derived from one clock and the receive side from another, so that they can be asynchronous to one another. If the PLD is not needed, it can be tri-stated to remove it (electrically) from the board. Signals can then be sent to the DS31256 by the pin headers.

The board is intended to be a full-size PCI card that can only be plugged into a 5V PCI system environment. There is a 256-pin plastic BGA socket on the board for the DS31256.

Only the DS31256 operates at 3.3V. Since it cannot be guaranteed that a 3.3V supply will exist in a 5V PCI system environment, the DS31256DK has a linear regulator on it (U4: LT1086) to convert from 5V to 3.3V. All of the other logic, including the PLD and oscillators, operate at 5V. If 3.3V exists on the PCI bus, the linear regulator can be removed and a 0Ω jumper can be installed at R97 (Figure 1-1).

The JTAG pins on the DS31256 are not active on the DS31256DK. Therefore, the JTCLK, JTDI, and JTMS signals are wired to 3.3V and $\overline{\text{JTRST}}$ is wired low.

Figure 1-1. PCI Card Configuration

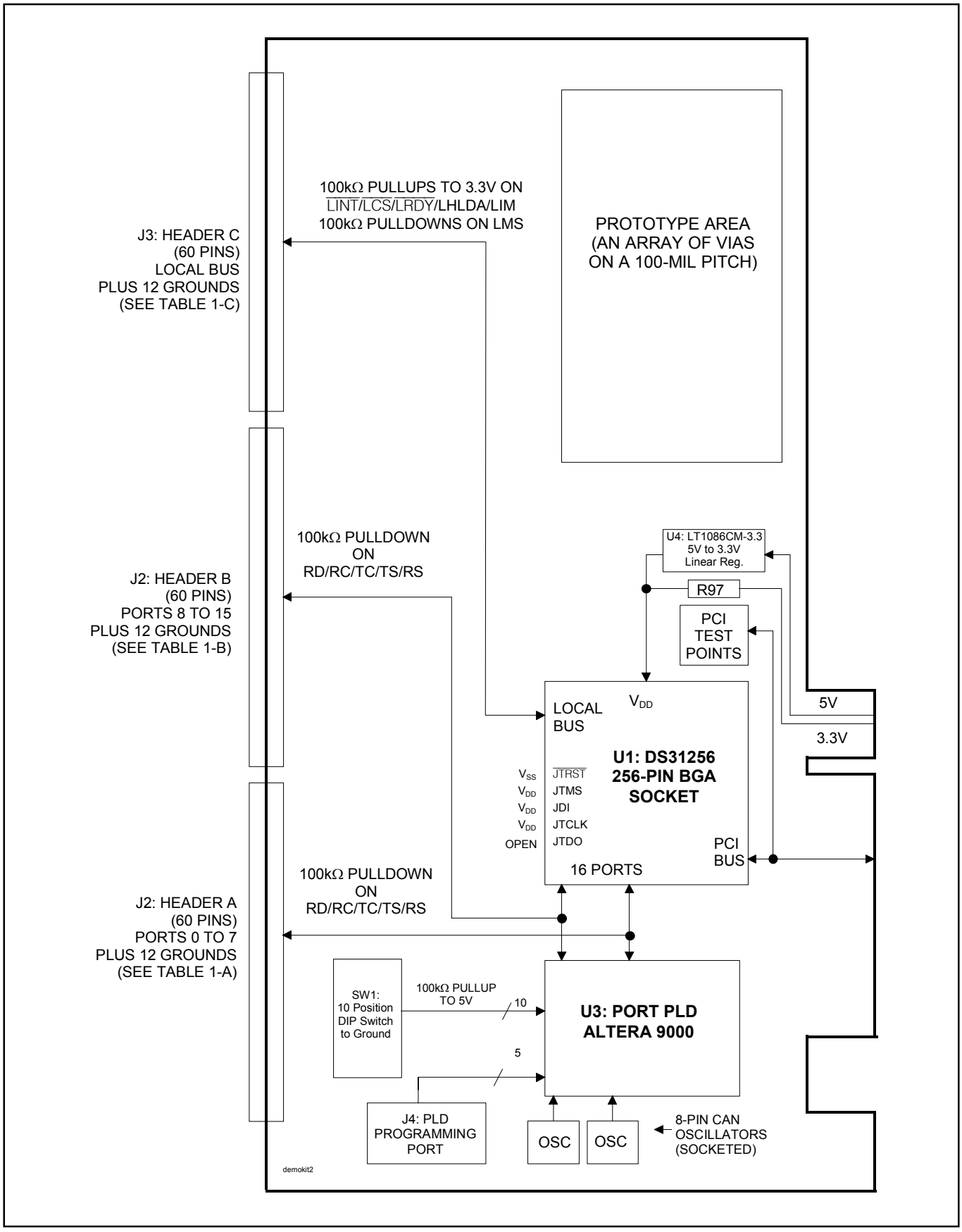
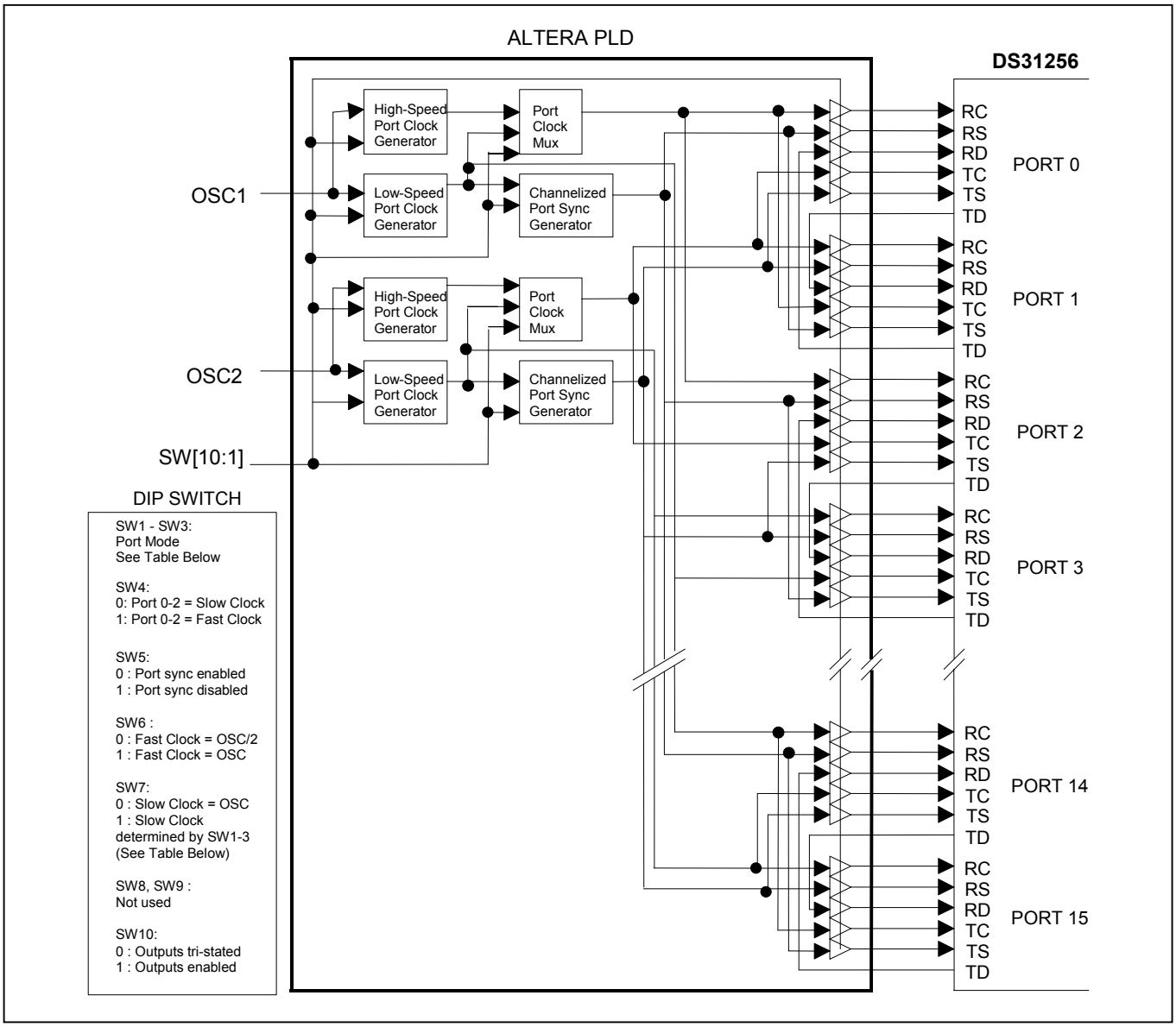


Figure 1-2. Port PLD Schematic



Clock/Sync Definitions

SW7	SW3	SW2	SW1	MODE	SLOW PORT CLOCK
0	0	0	0	Unchannelized (sync = low)	OSC
0	0	0	1	4 T1/E1 (sync active)	OSC
0	0	1	0	2 T1/E1 (sync active)	OSC
0	0	1	1	E1 (sync active)	OSC
0	1	0	0	T1 (sync active)	OSC
1	0	0	0	Unchannelized (sync = low)	66MHz / 6 = 11MHz
1	0	0	1	4 T1/E1 (sync active)	66MHz / 8 = 8.25MHz
1	0	1	0	2 T1/E1 (sync active)	66MHz / 16 = 4.125MHz
1	0	1	1	E1 (sync active)	66MHz / 32 = 2.0625MHz
1	1	0	0	T1 (sync active)	66MHz / 42 = 1.572MHz
X	1	0	1	Clock off (sync = low)	Clock driven low
X	1	1	X	Clock off (sync = low)	Clock driven low

Note 1: Switch Open = Off = High (1).
Note 2: Switch Closed = On = Low (0).
Note 3: OSC1 and OSC2 are recommended to be the same frequency.
Note 4: The frequency of OSC1 and OSC2 supplied with the DK board is 66MHz.

Table 1-A. Header A Definitions

1	RS0	2	TS0
3	RD0	4	TD0
5	RC0	6	TC0
7	GND	8	GND
9	RS1	10	TS1
11	RD1	12	TD1
13	RC1	14	TC1
15	GND	16	GND
17	RS2	18	TS2
19	RD2	20	TD2
21	RC2	22	TC2
23	GND	24	GND
25	RS3	26	TS3
27	RD3	28	TD3
29	RC3	30	TC3
31	RS4	32	TS4
33	RD4	34	TD4
35	RC4	36	TC4
37	GND	38	GND
39	RS5	40	TS5
41	RD5	42	TD5
43	RC5	44	TC5
45	GND	46	GND
47	RS6	48	TS6
49	RD6	50	TD6
51	RC6	52	TC6
53	GND	54	GND
55	RS7	56	TS7
57	RD7	58	TD7
59	RC7	60	TC7

Table 1-B. Header B Definitions

1	RS8	2	TS8
3	RD8	4	TD8
5	RC8	6	TC8
7	GND	8	GND
9	RS9	10	TS9
11	RD9	12	TD9
13	RC9	14	TC9
15	GND	16	GND
17	RS10	18	TS10
19	RD10	20	TD10
21	RC10	22	TC10
23	GND	24	GND
25	RS11	26	TS11
27	RD11	28	TD11
29	RC11	30	TC11
31	RS12	32	TS12
33	RD12	34	TD12
35	RC12	36	TC12
37	GND	38	GND
39	RS13	40	TS13
41	RD13	42	TD13
43	RC13	44	TC13
45	GND	46	GND
47	RS14	48	TS14
49	RD14	50	TD14
51	RC14	52	TC14
53	GND	54	GND
55	RS15	56	TS15
57	RD15	58	TD15
59	RC15	60	TC15

Table 1-C. Header C Definitions

1	LD0	2	LD1
3	LD2	4	LD3
5	LD4	6	LD5
7	GND	8	GND
9	LD6	10	LD7
11	LD8	12	LD9
13	LD10	14	LD11
15	GND	16	GND
17	LD12	18	LD13
19	LD14	20	LD15
21	LIM	22	LMS
23	GND	24	GND
25	LHOLD	26	LHLDA
27	$\overline{\text{LBGACK}}$	28	$\overline{\text{LINT}}$
29	$\overline{\text{LCS}}$	30	$\overline{\text{LRDY}}$
31	LCLK	32	$\overline{\text{LBHE}}$
33	$\overline{\text{LWR}}$	34	$\overline{\text{LRD}}$
35	LA0	36	LA1
37	GND	38	GND
39	LA2	40	LA3
41	LA4	42	LA5
43	LA6	44	LA7
45	GND	46	GND
47	LA8	48	LA9
49	LA10	50	LA11
51	LA12	52	LA13
53	GND	54	GND
55	LA14	56	LA15
57	LA16	58	LA17
59	LA18	60	LA19

2. ARCHITECTURE

2.1 Software

The architecture of the DS31256DK software (Figure 2-1) consists of a graphical user interface (GUI), a DK Driver, and the hardware abstraction layer using WinDriver. See Figure 2-2 for the software application.

- **Application:** The user can configure and access the DS31256 through a different GUI. Each GUI has separate functionality. The user can also monitor PCI and packet information.
- **DK Driver:** The DK Driver is two-way communication between the application and driver. The Application calls the DK Driver functions and sometimes passes in the necessary data. Some functions in the DK Driver return the data to the Application and display it in the GUI.
- **Hardware Abstraction Layer:**
  HDLC Library—The driver interfaces with the Windows operating system to the DS31256DK's PCI hardware through the WinDriver functions. The HDLC Library was generated by WinDriver through its automatic code generation. The functions have the structure required for the WinDriver function calls.

Figure 2-1. Software Architecture

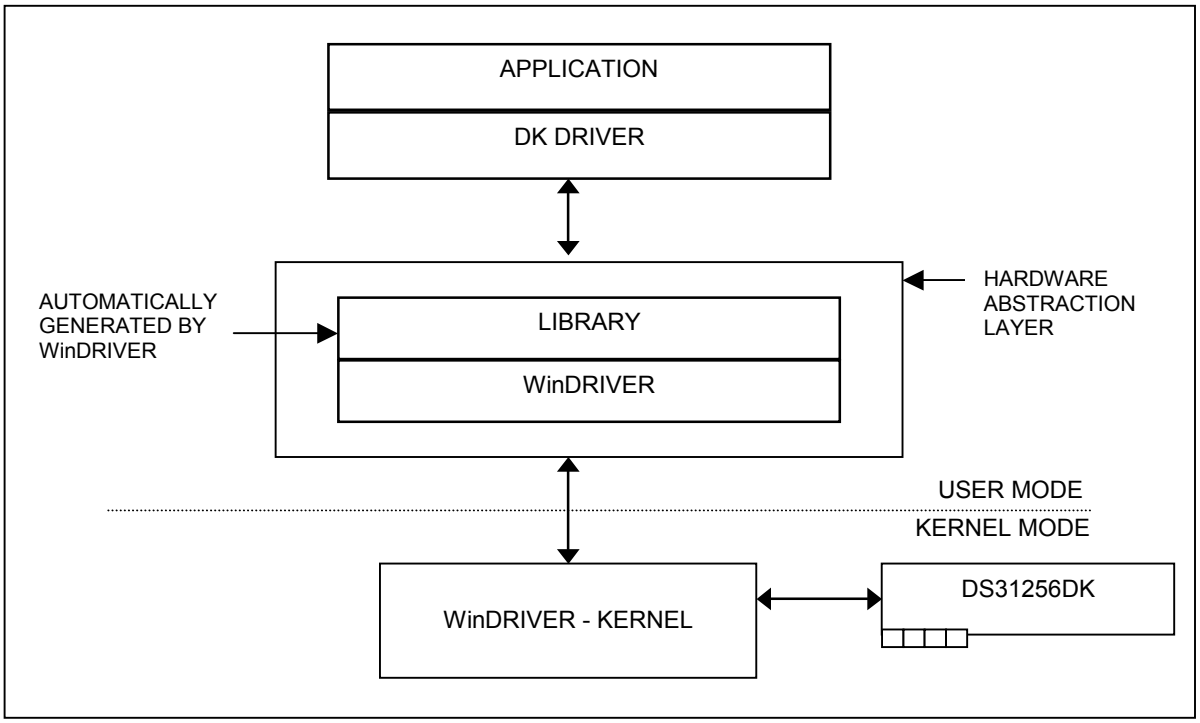
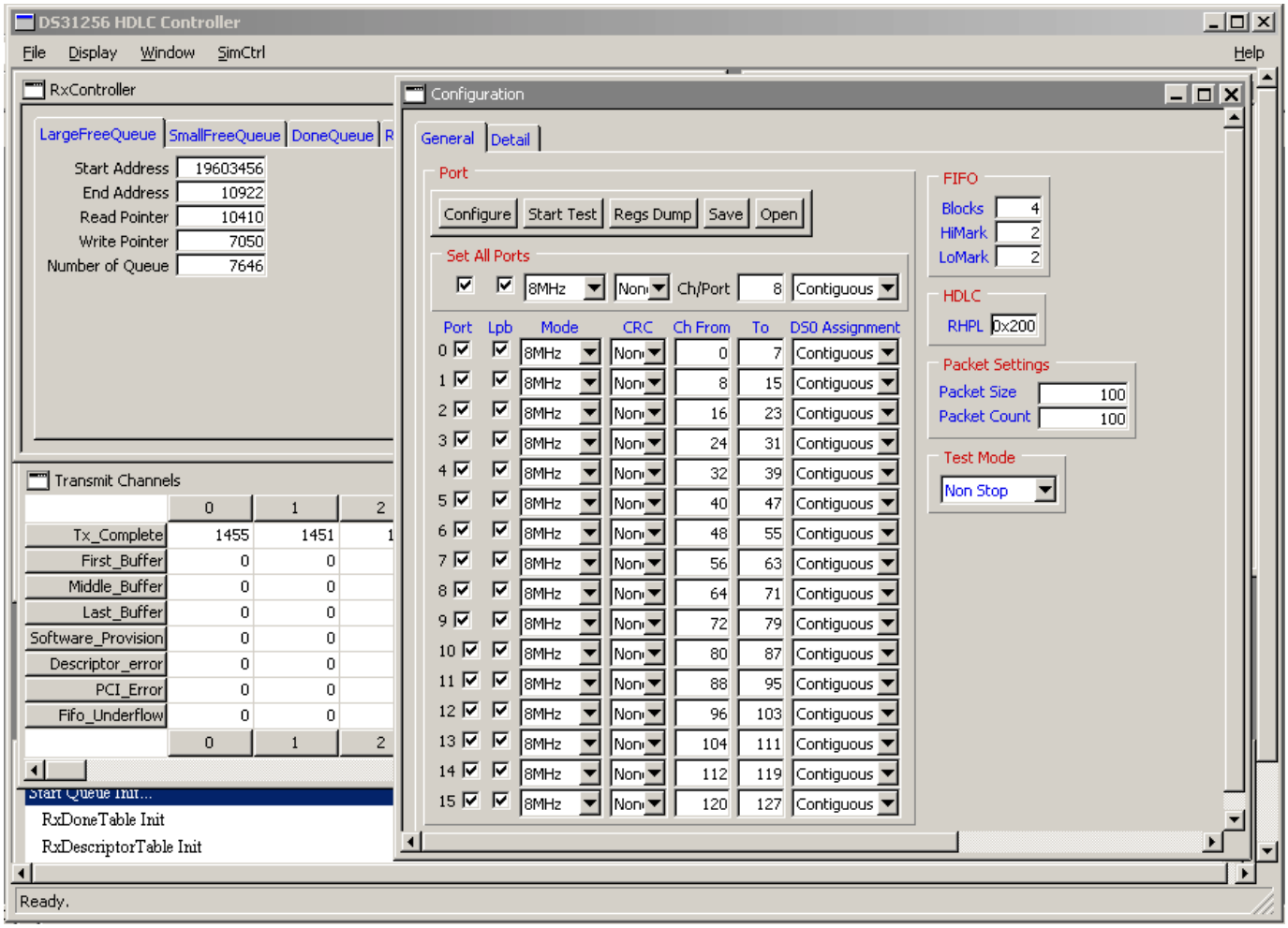


Figure 2-2. DS31256DK Software GUI



2.2 Software Overview

The DS31256DK software is written to run under a PC loaded with a Windows 95/98/NT/2000 operating system using the DS31256DK PCI card. The three components of the software are the Application, the DK Driver, and the Hardware Abstraction Layer.

The first part of the software architecture is the Application. The second part is the DK Driver. The DK Driver is independent and communicates between the Application and WinDriver. The third part is the Hardware Abstraction Layer developed by using WinDriver.

The applications software provides:

- A simple demonstration of the DS31256 with the ability to set the device into a number of different configurations.
- The detail configuration that let the user does the detail configuration bit by bit via the bits/register setting GUI.
- The portable driver code which can be reused with other applications.
- The ability to explore and load new data into the registers.
- The ability to read back and write into the registers.
- A utility to dump the internal registers and DMA into a file.

The user can generally select the desired number of ports, mode, channels, CRC setting, loopback mode, etc., through the general configuration GUI to configure the DS31256, or, configure the DS31256 bit by bit through the detail bits/registers GUI.

The Application passes parameters when it calls the DK Driver functions. The driver will communicate to the device by using the existing functions from WinDriver. The data transmission will start after the configuration processes are completed. The Application sits on the top level will display the most update information that returns from the requested procedures.

When a test starts, the DS31256 transmits data that will be looped back to either the same port (if Local Loopback is used) or to an adjacent port (if the Altera PLD is used to loop the data). The software checks the reception of packets to make sure that they are received without error (i.e., the CRC is correct). For each HDLC channel that is enabled, the software also keeps track of the number of packets sent, the number of packets received, the number of packets received in error, and a variety of other statistics/counts.

The design kit software does not implement all the functions available in the DS31256. However, it does implement 16 ports coupled with 256, independent, bidirectional HDLC channels.

2.3 DK Driver

The independent DK Driver shown in Figure 2-1 handles the necessary functions to transmit and receive data into and out of the hardware. The users can easily reuse the DK Driver with their own application and replace the part of WinDriver with their own layer to talk to the hardware/system.

The DK Driver includes four files.

- BitDef—The definition file of DS31256 registers and bits.
- DkDriver—The interface code to WinDriver, system, and memory management functions.
- Hdlc_lib—The library for accessing the HDLC card. The code accesses the hardware through WinDriver functions.
- RegDef—The definition file of DS31256 registers and addresses.

The DK Driver can be used with any application that uses WinDriver function calls. Since the DK Driver and the hardware abstraction layer is totally independent, the user can replace the hardware abstraction layer if the WinDriver is not used. The user could create their own driver base on the functions in DkDriver file to interface their system or hardware. In any case, this software should be a good example or reference for an implementation.

Appendix A contains reference tables listing all functions in DkDriver and Hdlc_lib files. The registers (RegDef) and bits (BitDef) definition files are not listed in Appendix A. Users can use them for their software development.

Note that some of the data structures are elaborate and are defined in header files that must be include in the project at compile time.

3. INSTALLATION AND GETTING STARTED

3.1 Card Installation

The following are separate instructions for Windows 95/98/2000/NT systems.

3.1.1 Windows 95 Systems

1) Power-down the host computer system and open its case. Be sure to follow ESD precautions while in contact with the card, the DS31256, and system components.
2) If not already seated, install the DS31256 chip into the BGA socket on the DK's PC board. (As shown on cover page)
3) Set the DIP switches on the card to configure the board and operational mode (see Figure 1-2).
4) Plug the DS31256DK card into an empty PCI slot.
5) Reassemble the computer.
6) Boot the computer.
7) Insert the DS31256DK1 CD.
8) Open a DOS window to perform the following commands.
9) Change directory to c:\windows\system\vmm32.
10) Copy the file *windrvr.vxd* from the CD "Install\Win95" directory to c:\windows\system\vmm32.
11) Copy the file *wdreg.exe* from the CD "Install\Win95" directory to c:\windows\system\vmm32.
12) Run *wdreg -vxd install* from the DOS prompt in the current working directory.
13) Close the DOS shell and reboot the machine.

3.1.2 Windows 98 Systems

1) Power-down the host computer system and open its case. Be sure to follow ESD precautions while in contact with the card, the DS31256, and system components.
2) If not already seated, install the DS31256 chip into the BGA socket on the DK's PC board (as shown on cover page).
3) Set the DIP switches on the card to configure the board and operational mode (see Figure 1-2).
4) Plug the DS31256DK card into an empty PCI slot.
5) Reassemble the computer.
6) Boot the computer and do not allow the system to search for or install drivers for the new hardware.
7) Insert the DS31256DK1 CD.
8) Open a DOS window to perform the following commands.
9) Copy the file *windrvr.sys* and *wdpnp.sys* from the CD "Install\Win98" directory to c:\windows\system32\drivers.
10) Copy the file *wdreg.exe* from the CD "Install\Win98" directory to c:\windows\system32\drivers.
11) Run *wdreg install* from the DOS prompt in the current working directory.
12) Close the DOS shell, and reboot the machine.

3.1.3 Windows NT/2000 Systems

1) Power-down the host computer system and open its case. Be sure to follow ESD precautions while in contact with the card, the DS31256, and system components.
2) If not already seated, install the DS31256 chip into the BGA socket on the DK's PC board (as shown on the cover page).
3) Set the DIP switches on the card to configure the board and operational mode (see Figure 1-2).
4) Plug the DS31256DK card into an empty PCI slot.
5) Reassemble the computer.
6) Boot the computer.
7) Insert the DS31256DK1 CD.
8) Open a DOS window to perform the following commands.
9) Change directory to c:\winnt\system32.
10) Copy the file *windrvr.sys* and *wdpnp.sys* from the CD "Install\WinNT" directory to c:\winnt\system32\drivers.
11) Copy the file *wdreg.exe* from the CD "Install\WinNT" directory to c:\winnt\system32\drivers.
12) Run *wdreg install* from the DOS prompt in the current working directory.
13) Close the DOS shell and reboot the machine.

3.2 Software Installation

1) Make a directory on the system.
2) Copy *DKGui.exe* from the CD “Install\<OS_Type>” to the target directory.
3) Create a shortcut to the program, or set up a menu entry for it.

Note: The source code for DKGui and the underlying drivers is in the CD “Source” directory. If desired, the source directory can also be copied off of the CD to the host.

3.3 Operational Test

After performing the card and software installations as described,

1) Make sure that the board's DIP switches are set as follows:

1	2	3	4	5	6	7	8	9	10
On	On	Off	On	On	Off	Off	Off	Off	Off

2) Execute the DkGui.exe program.
3) The first GUI will list all the devices that are sitting in the PC PCI bus.
4) Pick the card you want to use/configure.
5) Complete the configuration processes in the next pop-up GUI.
6) All the GUIs with different functionality will pop up after the configuration processed completed.
7) You will see the detail data from the GUI that has been selected.

4. APPENDIX A

DkDriver.h	The interface code to WinDriver, system, and memory management functions.	
	FUNCTION	**PURPOSE**
	DkAllocateBuffer	Allocate the buffer in the memory
	DkClearMemory	Clear the memory
	DkCloseDevice	Close device
	DkConfigDevice	Initial the buffer in the last step of configuration
	DkCreateDefaultDevice	Default setting of device
	DkDisableDevice	Disable the device by master control register
	DkDmaChanCtrl	Set DMA Channel Configuration RAM
	DkDmaCtrl	Transmit/Receive DMA enable/disable
	DkEnableChannel	Enable HDLC channel in DMA
	DkEnableDevice	Enable the device
	DkFlushLargeBuffer	Set RFQLF bit in TDMAQ register
	DkFlushPendingQueue	Set TPQF bit in TDMAQ register
	DkFlushRxDoneQueue	Set RDQF bit in TDMAQ register
	DkFlushSmallBuffer	Set RFQSF bit in TDMAQ register
	DkFlushTxDoneQueue	Set TDQF bit in TDMAQ register
	DkGetPhysicalAddress	Get physical address
	DkGetUserAddress	Get user address
	DkIndReadWord	Read back from the indirect register
	DkIndVerifyWord	Verify the indirect registers
	DkIndWriteWord	Write into the indirect register
	DkInitBuffer	Initial buffer table
	DkIsWriteWord	Write into the DS31256 indirect registers
	DkOpenDevice	Open device when called
	DkPciRead	Write into the PCI register
	DkPciWrite	Write into the PCI register
	DkReadWord	Read back from the register
	DkSetInterruptState	Set interrupt
	DkSetReceiveDoneQueueWrite	Receive Done Queue write
	DkSetTransmitDoneQueueWrite	Transmit Done Queue write
	DkSoftReset	DS3131 Reset Function

DkDriver.h	The interface code to WinDriver, system, and memory management functions.	
	FUNCTION	**PURPOSE**
	DkVerifyWord	Verify the register
	DkWriteBaseAddress	Write into the base address
	DkWriteWord	Write into the register

hdlc_lib	**Library for accessing the HDLC card, code was generated by Driver Wizard. It accesses the hardware via WinDriver functions. (See WinDriver developer's guide.)**	
	FUNCTION	**PURPOSE**
	HDLC_AllocatePhyMem	Allocate the physical memory for future use
	HDLC_ClearMemory	Clear the memory
	HDLC_Close	Must be called after finished using the driver to close WinDriver device.
	HDLC_CloseDriver	Close the driver after complete use
	HDLC_CountCards	Returns the number of matching PCI cards found
	HDLC_DetectCardElements	Get card information includes the ubterrupts, IO and memory
	HDLC_GetPhyAddr	Get the physical address from the memory via WinDriver function
	HDLC_GetUserAddr	Get the address from the memory via WinDriver function
	HDLC_IntDisable	Disable interrupt processing
	HDLC_IntEnable	Enable interrupt processing
	HDLC_IntIsEnabled	Checks whether interrupts are enabled or not
	HDLC_IsAddrSpaceActive	Checks if the specified address space is enabled
	HDLC_Open	Open a handle to the card
	HDLC_PCIGetWDhandle	Get PCI cards information
	HDLC_ReadByte	Reads a byte from address space on board
	HDLC_ReadDword	Reads a dword from address space on board
	HDLC_ReadPCIReg	Read from the PCI configuration registers
	HDLC_ReadWord	Reads a word from address space on board
	HDLC_ReadWriteBlock	General read/write function
	HDLC_RegisterDriver	WinDriver registration
	HDLC_WriteByte	Writes a byte from address space on board
	HDLC_WriteDword	Writes a dword from address space on board
	HDLC_WritePCIReg	Write to the PCI configuration registers
	HDLC_WriteWord	Writes a word from address space on board

5. ADDITIONAL INFORMATION

This data sheet has shown how to set up and use the DS31256 software in different operating systems. Full free source code is also included in the DS31256DK CD.

If you have further questions about our HDLC controller products, contact the Telecommunication Applications support team via email telecom.support@dalsemi.com, or call 972-371-6555.

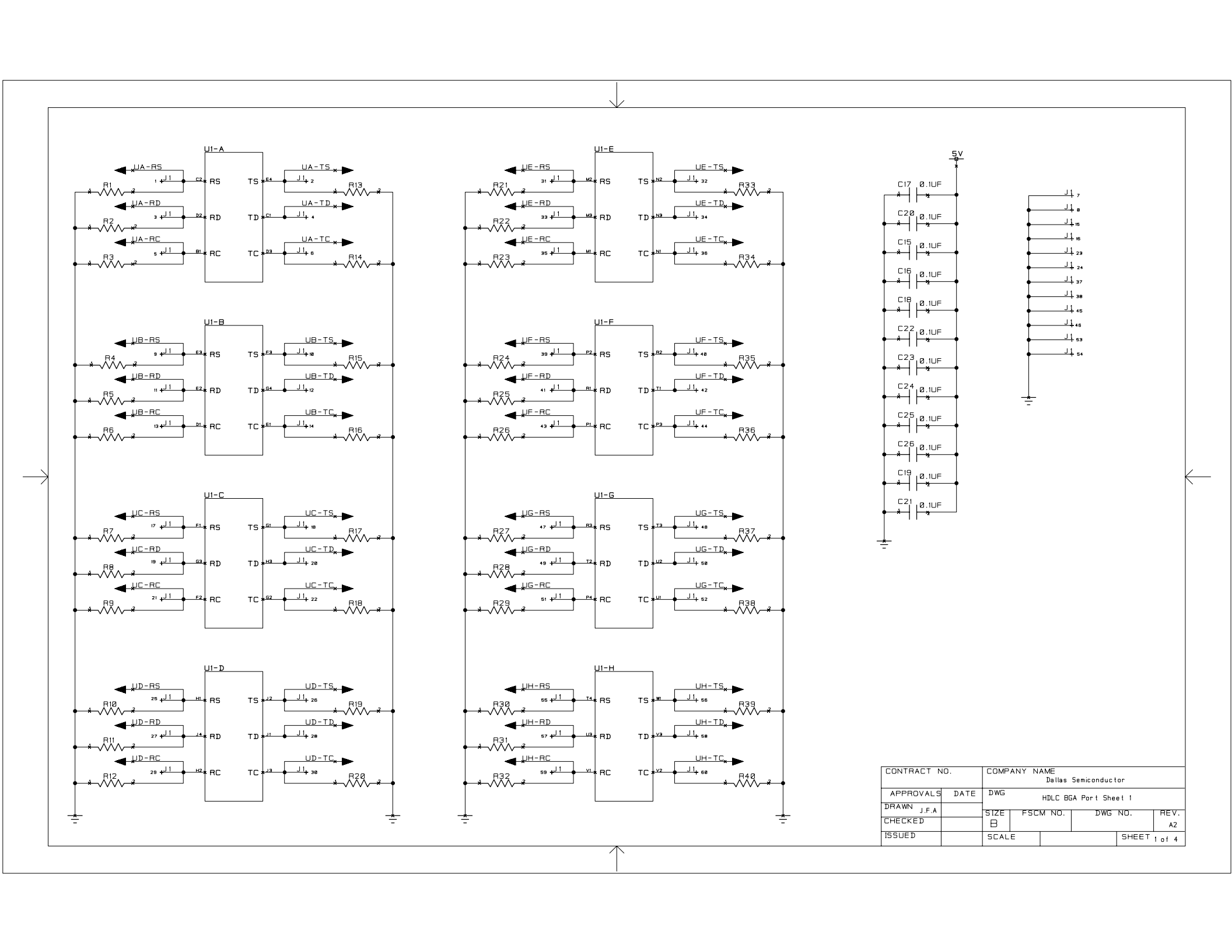
5.1 DS31256 Information

For more information about the DS31256, refer to the DS31256 data sheet available on our website at www.maxim-ic.com/DS31256.

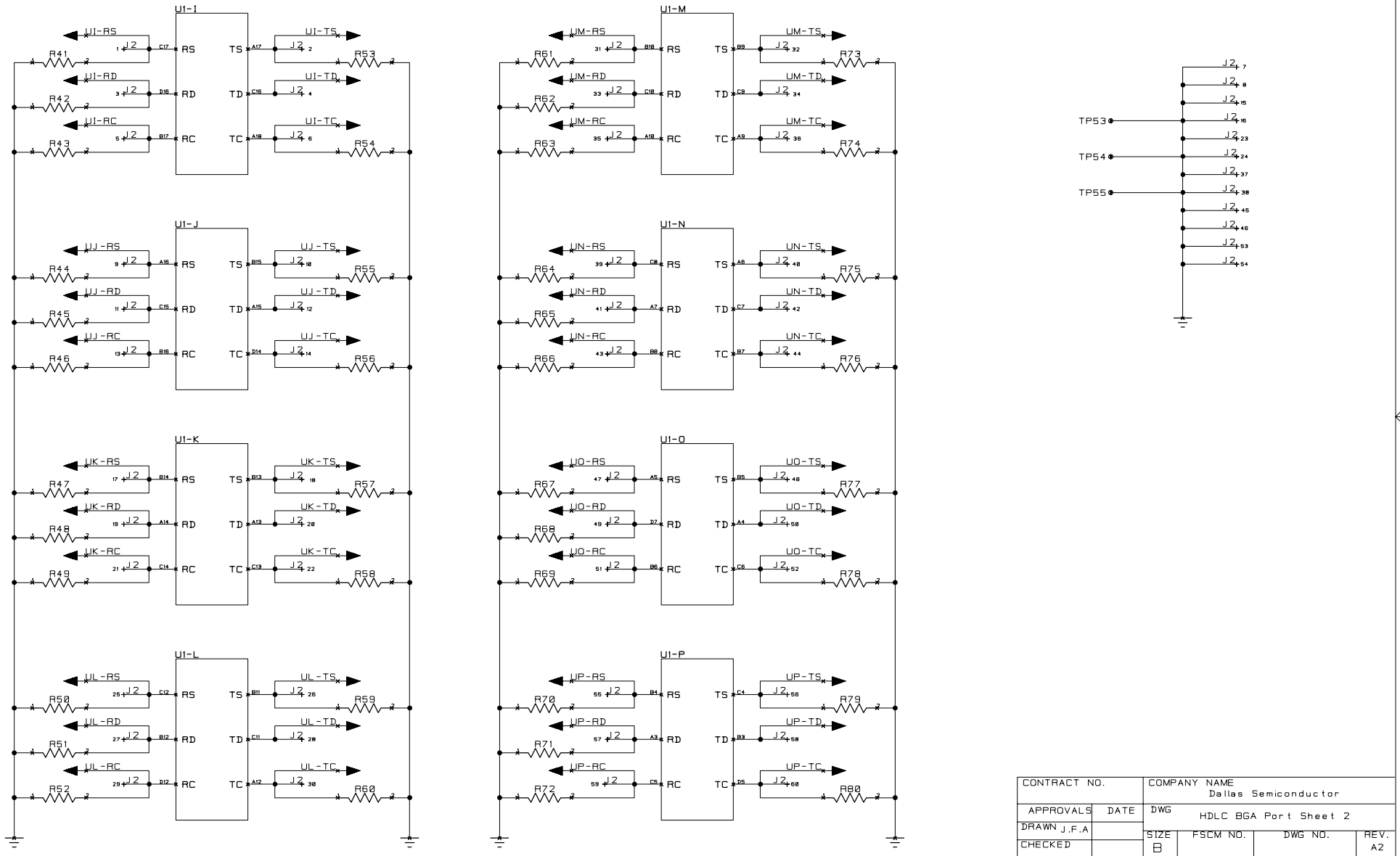
6. SCHEMATICS

The DS31256DK schematic is featured on the following pages.

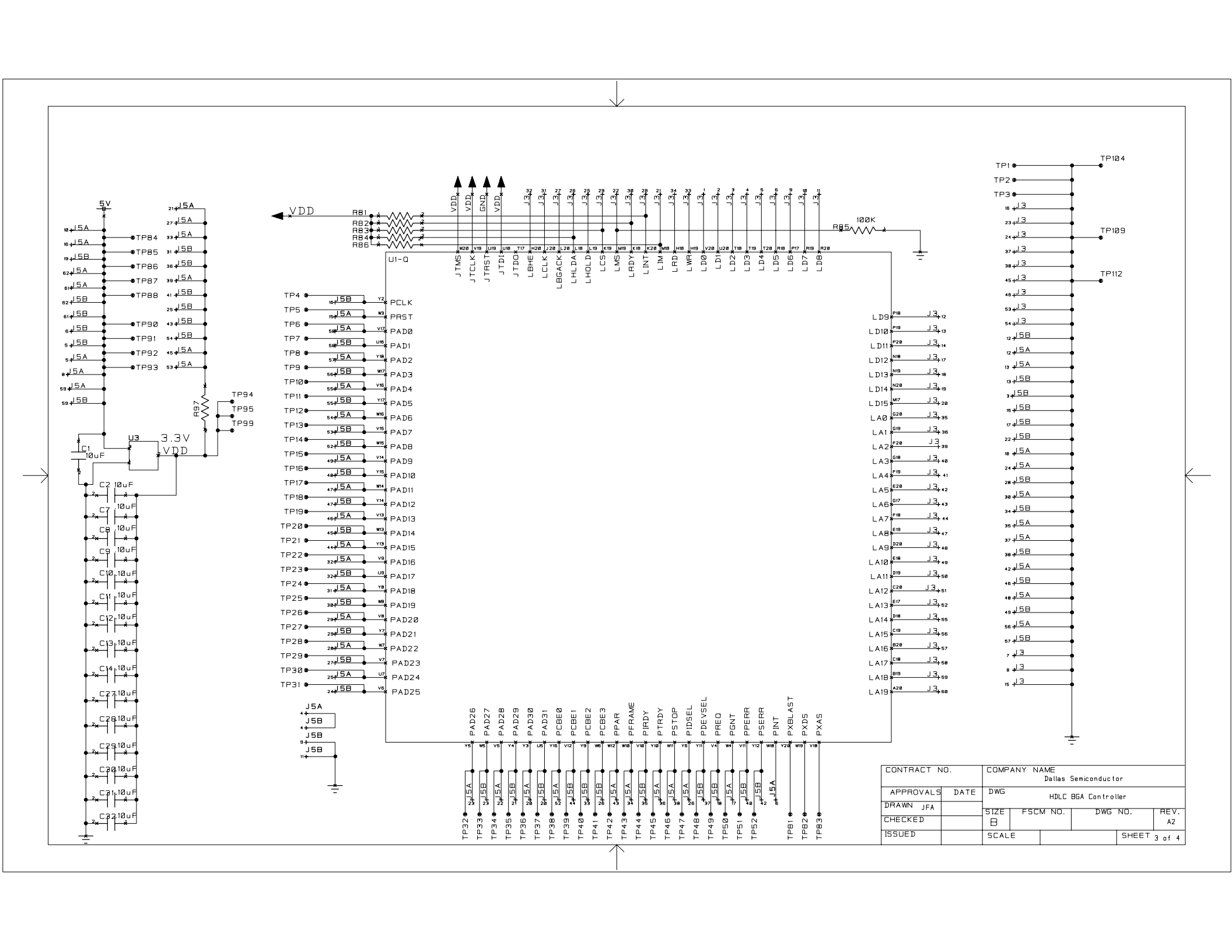
Maxim Integrated Products, 120 San Gabriel Drive, Sunnyvale, CA 94086 408-737-7600

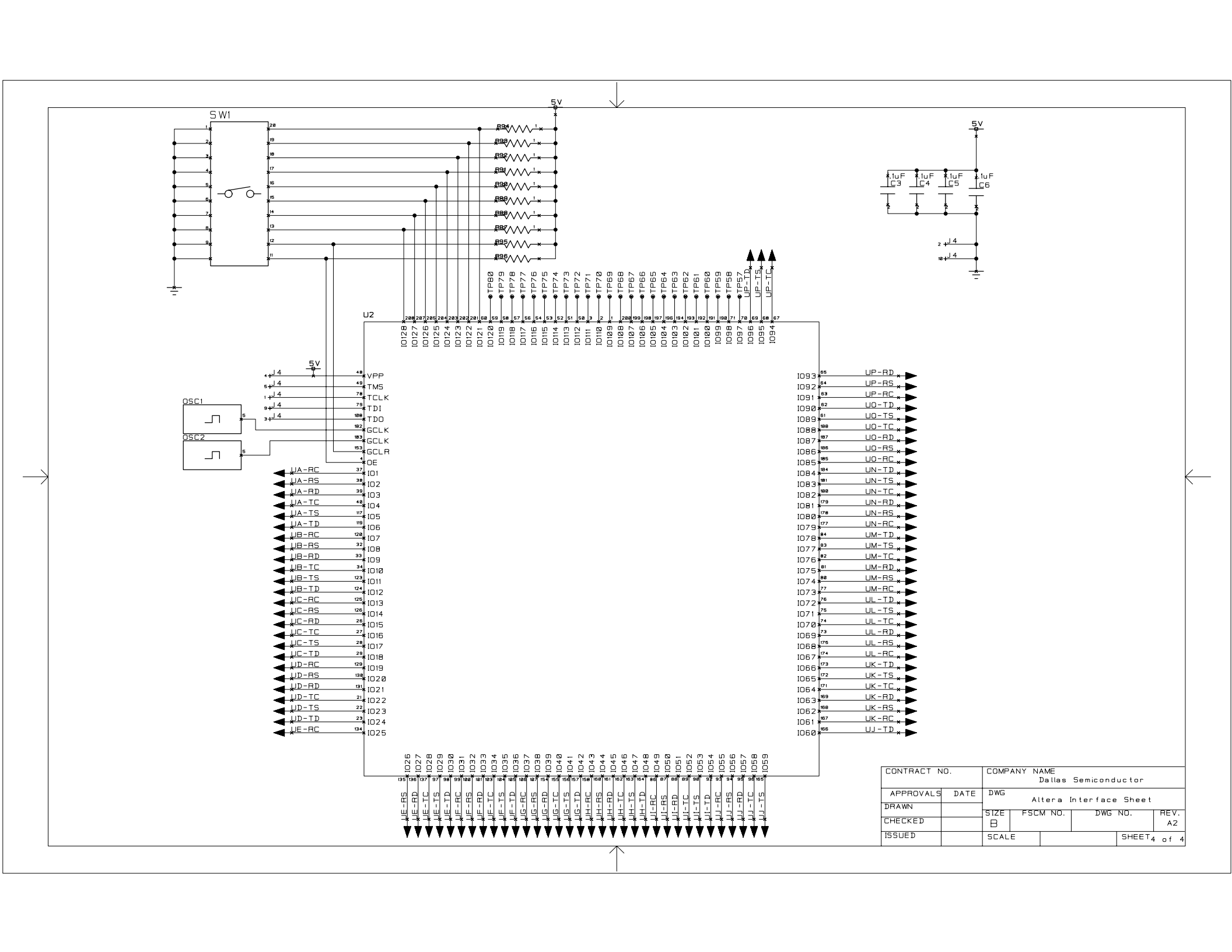


CONTRACT NO.		COMPANY NAME Dallas Semiconductor	
APPROVALS	DATE	DWG HDLC BGA Port Sheet 1	
DRAWN J.F.A		SIZE B	FSCM NO. / DWG NO. / REV. A2
CHECKED			
ISSUED		SCALE	SHEET 1 of 4



CONTRACT NO.		COMPANY NAME Dallas Semiconductor		
APPROVALS	DATE	DWG HDLC BGA Port Sheet 2		
DRAWN J.F.A		SIZE B	FSCM NO.	DWG NO. REV. A2
CHECKED				
ISSUED		SCALE		SHEET 2 of 4


5V
3.3V
VDD
U3
C1
10uF
C2 10uF
C7 10uF
C8 10uF
C9 10uF
C10 10uF
C11 10uF
C12 10uF
C13 10uF
C14 10uF
C27 10uF
C28 10uF
C29 10uF
C30 10uF
C31 10uF
C32 10uF
R97
TP84
TP85
TP86
TP87
TP88
TP90
TP91
TP92
TP93
TP94
TP95
TP99
R81
R82
R83
R84
R86
R85
100K
U1-Q
JTMS
JTCLK
JTRST
JTDI
JTDO
LBHE
LCLK
LBGACK
LHLDA
LHOLD
LCS
LMS
LRDY
LINT
LM
LRD
LWR
LD0
LD1
LD2
LD3
LD4
LD5
LD6
LD7
LD8
LD9
LD10
LD11
LD12
LD13
LD14
LD15
LA0
LA1
LA2
LA3
LA4
LA5
LA6
LA7
LA8
LA9
LA10
LA11
LA12
LA13
LA14
LA15
LA16
LA17
LA18
LA19
PCLK
PRST
PAD0
PAD1
PAD2
PAD3
PAD4
PAD5
PAD6
PAD7
PAD8
PAD9
PAD10
PAD11
PAD12
PAD13
PAD14
PAD15
PAD16
PAD17
PAD18
PAD19
PAD20
PAD21
PAD22
PAD23
PAD24
PAD25
PAD26
PAD27
PAD28
PAD29
PAD30
PAD31
PCBE0
PCBE1
PCBE2
PCBE3
PPAR
PFRAME
PIRDY
PTRDY
PSTOP
PIDSEL
PDEVSEL
PREQ
PGNT
PPERR
PSERR
PINT
PXBLAST
PXDS
PXAS
TP1
TP2
TP3
TP104
TP109
TP112
CONTRACT NO.
COMPANY NAME
Dallas Semiconductor
APPROVALS
DATE
DRAWN JFA
CHECKED
ISSUED
DWG
HDLC BGA Controller
SIZE
B
FSCM NO.
DWG NO.
REV.
A2
SCALE
SHEET 3 of 4



CONTRACT NO.		COMPANY NAME: Dallas Semiconductor	
APPROVALS	DATE	DWG: Altera Interface Sheet	
DRAWN		SIZE: B	FSCM NO. / DWG NO. / REV.: A2
CHECKED		SCALE	SHEET: 4 of 4
ISSUED			

Parts List

Cadstar Design Editor Version 2.2

Design: DS3134DK REVA 100698

Date: Tuesday, October 06, 1998
Time: 3:21 PM

DS3134 DESIGN PARTS LIST

Part Name	Part Number	Description	Qty.	Comps.
.10UF-5%	87-47101-050	.10UF 5% CAP (1206)	30	C3-32
0K-5%-TR4		0K 5% RES (1206)	1	R97
0SC66	SE1128-ND:DGK	66 MHZ OSC. 8P DIP SOCKETED	2	OSC1-2
100K-5%-TR4	P100KETR-ND:DGK	100K 5% RES (1206)	96	R1-96
10UF-5%	87-47303-008	10UF 5% CAP (1206)	2	C1-2
DS3134_BGA	655256427020902:WEL	HDLC CONTROL 256P BGA SOCKETED	1	U1
EPM93020	EPM9320RC-208:ALT	PLD 208P RQFP	1	U2
Edge Conn 10P	MHB10K-ND:DGK	10P EDGE CONN (.100 SPC)	1	J4
Edge Conn 188P		94P PCI CONNECTOR	2	J5A
				J5B
Edge Conn 60P	MHD60K-ND:DGK	60P RT EDGE CONN (.100 SPC)	3	J1-3
LT1086-1.5A-3.3V	LT1086CM-3.3-ND:DGK	1.5A 3P LOW DROP REG (DD)	1	U3
SDS10-014		10 POS DIPSWITCH 20P DIP	1	SW1

End of report

NOTES:
1. DO NOT STUFF R97,C27-C32.
2. U1,OSC1-2 ARE SOCKETED
3. DGK = DIGIKEY
4. ALT = ALTERA
5. WEL = WELLS-CTI
5. U1 SOCKET ALT PART NUMBER IS PLASTRONICS 256BG12B127