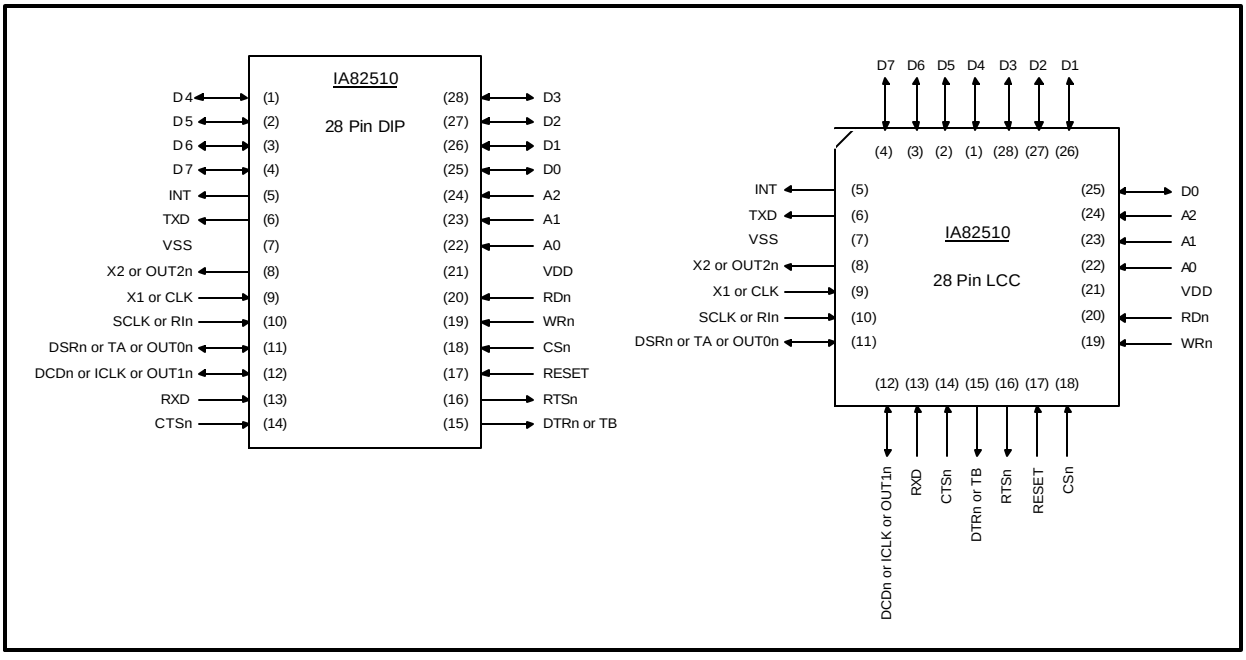
FEATURES

- **Form, Fit, and Function Compatible with the Intel® 82510**
- **Packaging options available: 28 Pin Plastic or Ceramic DIP, 28 Pin Plastic Leaded Chip Carrier, 28 Pin Ceramic Leadless Chip Carrier**
- **Asynchronous Serial Channel Operation**
- **Separate Transmit and Receive FIFOs with Programmable Threshold**
- **Programmable Baud Rate Generators up to 288K Baud**
- **Special Protocol Features**
  - **Control Character Recognition**
  - **Auto Echo and Loopback Modes**
  - **9-Bit Protocol Support**
  - **5 to 9 Bit Character Format**

The IA82510 is a "plug-and-play" drop-in replacement for the original IC. innovASIC produces replacement ICs using its MILES™, or Managed IC Lifetime Extension System, cloning technology. This technology produces replacement ICs far more complex than "emulation" while ensuring they are compatible with the original IC. MILES™ captures the design of a clone so it can be produced even as silicon technology advances. MILES™ also verifies the clone against the original IC so that even the "undocumented features" are duplicated. This data sheet documents all necessary engineering information about the IA82510 including functional and I/O descriptions, electrical characteristics, and applicable timing.

Package Pinout

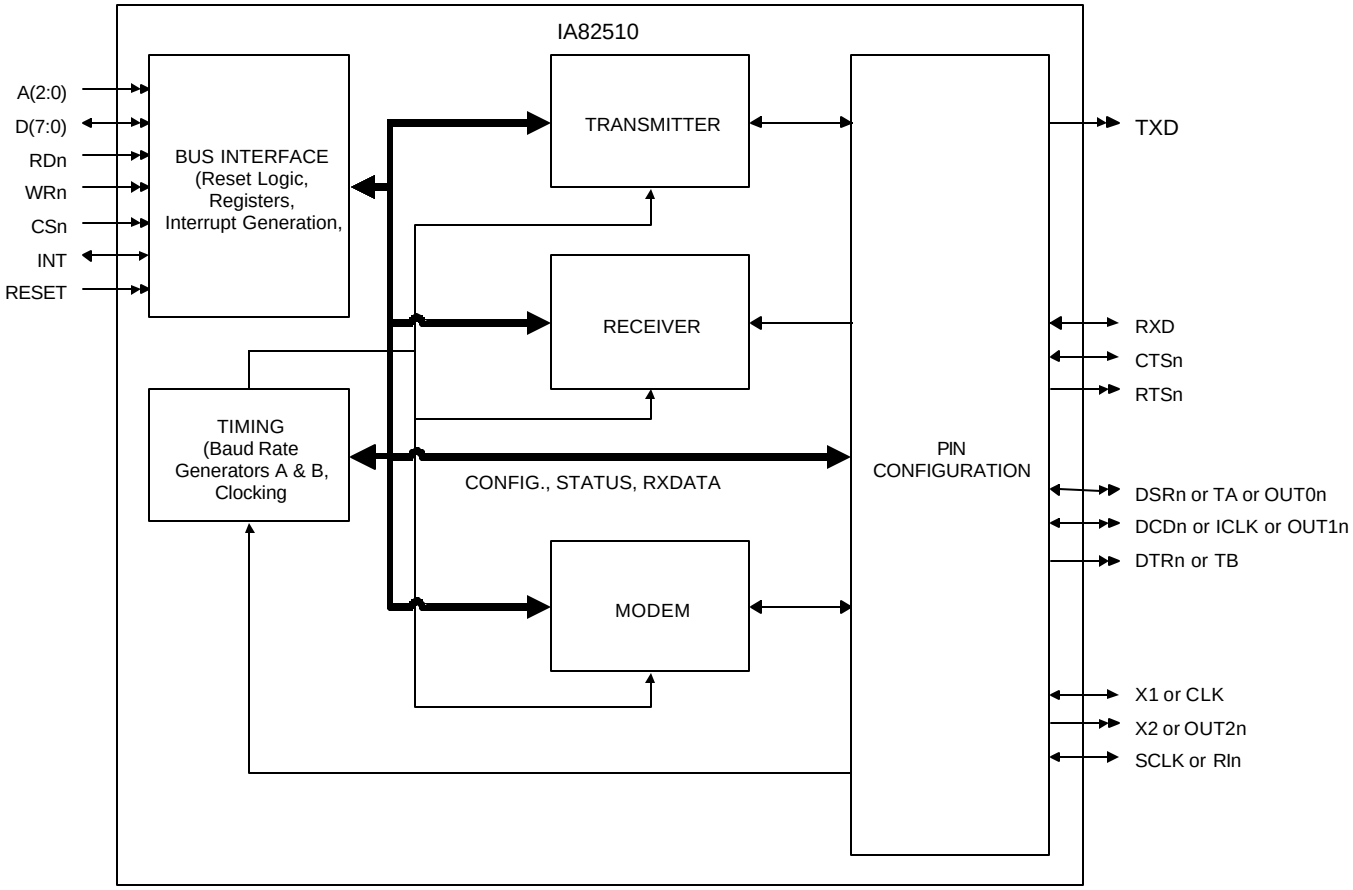


www.innovasic.com
Customer Support:
1-888-824-4184

DESCRIPTION

The IA82510 is an asynchronous serial controller that provides a CPU interface to one transmit and one receive channel. It is Form, Fit, and Function compatible with the Intel 82510. Configuration registers are used to control the serial channel, interrupts, and modes of operation. The CPU controls this device via address and data lines with read/write control. The CPU also uses this interface to read and write data to receive and transmit data through the serial channel. FIFOs and various serial modes can be used to help off-load the CPU from transmitting and receiving data. An interrupt line provides an indication to the CPU that the device requires servicing. The device can be configured for 8250A/16450 compatibility.

Functional Block Diagram



Functional Overview

Transmitter

The Transmit function consists of a 4×11 bit FIFO, and a Transmit Engine. The 4×11 FIFO is configurable as any depth between one and four words inclusive. The transmit engine is responsible for reading the data out of the FIFO and placing it in the proper order on the TXD pin. The transmit engine is highly configurable to be compatible with numerous formats, including 16450 and 8250 modes of communication. Transmit Communication parameters that can be programmed include:

- Parity modes
- Stop Bits
- Character Length
- FIFO Depth
- Clocking Options
- RTS and CTS modes

See the Register Description for more details.

Receiver

The Receiver function consists of a 4×11 configurable FIFO and a Receive Engine. The receive engine is responsible for sampling the data on the RXD input pin, formatting the data, and placing the data in the FIFO. The receive engine is highly configurable with parameters that include:

- Parity modes
- Stop Bits
- Character Length
- FIFO Depth
- Clocking Options
- Address Matching Options
- Control Character Detection
- RTS and CTS modes

See the Register Description for more details.

Bus Interface

The Bus Interface is a simple interface that allows a micro-processor or micro-controller to read and write the IA82510 Registers. It consists of the following I/O lines:

- A0, A1, A2 : 3 Bit Address
- D0-D7 : 8 Bit Data
- RDn: Active Low Read Enable
- WRn: Active Low Write Enable
- CSn: Active Low Chip Select
- INT: Interrupt Output
- RESET: Chip Reset

Register Description

Table 1 – IA82510 Register Summary

Register	ADDR	Bank	DLAB	Mode	Default
ACR0	111	00	X	R/W	00000000
ACR1	101	10	X	R/W	00000000
BACF	001	11	0	R/W	00000100
BAH	001	00	1	R/W	00000000
BAL	000	00	1	R/W	00000010
BANK	010	X	X	W	00000000
BBCF	011	11	X	R/W	10000100
BBH	001	11	1	R/W	00000000
BBL	000	11	1	R/W	00000101
CLCF	000	11	0	R/W	00000000
FLR	100	01	X	R	00000000
FMD	001	10	X	R/W	00000000
GER	001	00	0	R/W	00000000
GIR_BANK	010	X	X	R	00000001
GSR	111	01	X	R	00010010
ICM	111	01	X	W	N/A
IMD	100	10	X	R/W	00001100
LCR	011	00	X	R/W	00000000
LSR	101	00	X	R/W	01100000
MCR	100 100	00 01	X X	R/W W	00000000
MIE	101	11	X	R/W	00001111
MSR	110 110	00 01	X X	R/W R	00000000
PMD	100	11	X	R/W	11111100
RCM	101	01	X	W	N/A
RIE	110	10	X	R/W	00011110
RMD	111	10	X	R/W	00000000
RST	101	01	X	R	00000000
RXDATA	000	00 01	0 X	R	Unknown
RXF	001	01	X	R	Unknown
TCM	110	01	X	W	N/A
TMCR	011	01	X	W	N/A
TMD	011	10	X	R/W	00000000
TMIE	110	11	X	R/W	00000000
TMST	011	01	X	R	00110000
TXDATA	000	00 01	0 X	W	N/A
TXF	001	01	X	W	N/A

AC/DC Parameters

Absolute maximum ratings:

Supply Voltage, V_{DD}……..…-0.3V to +6.0V
Input Voltage, V_{IN}…….-0.3V to V_{DD} +0.3V
Input Pin Current, IIN…….±10 mA, 25° C
Operating Temperature Range……..-40° C to +85°C
Ambient temperature under bias.....……..-40°C to +85°C *
Storage temperature.....….....….- 55°C to +150°C
Lead Temperature…….+300°C, 10 sec.
Power dissipation.....155 mW, 125°C, 25MHz, 15% Toggle

Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. Operating the device beyond the conditions indicated in the "recommended operating conditions" section is not recommended. Operation at the "absolute maximum ratings" may adversely affect device reliability.

* The input and output parametric values in section VII-B, parts 1, 2, and 3, are directly related to ambient temperature and DC supply voltage. A temperature or supply voltage range other than those specified in the Operating Conditions above will affect these values and part performance is not guaranteed by innovASIC.

DC Characteristics

Symbol	Parameter	Notes	Min	Max	Unit
V_{IL}	Input Low Voltage	(1)	-0.5	0.7	V
V_{IH1}	Input High Voltage-Cerdip	(1)	2.1	V_{DD}+.07	V
V_{IH2}	Input High Voltage-LCC	(2)	2.1	V_{DD}+.07	V
V_{OL}	Output Low Voltage	(2),(8)		0.4	V
V_{OH}	Output High Voltage	(3),(8)	2.4		V
I_{LI}	Input Leakage Current	(4)		±1	µA
I_{LO}	3-State Leakage Current	(5)		±1	µA
I_{CC}	Power Supply Current	(6)		1.12	mA/MHz
I_{PU}	Strapping Pullup Resistor	(12)	-283	-137	µA
I_{STBY}	Standby Supply Current	(9)		100	µA
I_{OHR}	RTSn, DTRn Strapping Current	(10)		1.92	mA
I_{OLR}	RTSn, DTRn Strapping Current	(11)	N/A		mA
C_{IN}	Input Capacitance	(7)		5	pF
C_{IO}	I/O Capacitance	(7)		6	pF
C_{XTAL}	X1, X2 Load			6	pF

Notes:

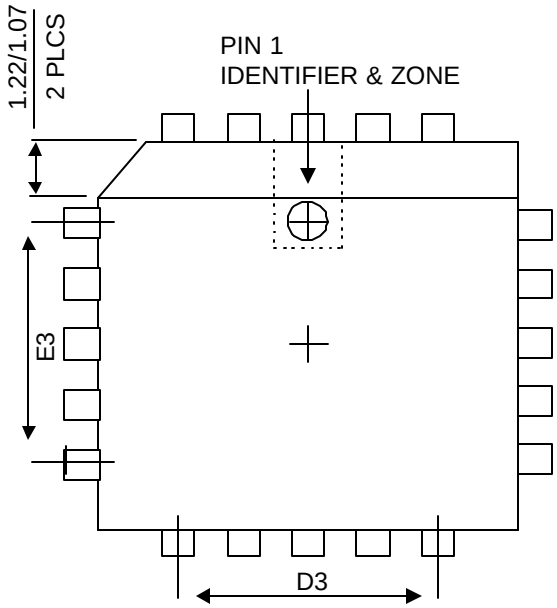
1. Does not apply to CLK/X1 pin, when configured as crystal oscillator input (X1).
2. @I_{OL} = 1.92 mA
3. @I_{OH} = 1.92 mA
4. $0 < V_{IN} < V_{CC}$
5. $0.4V < V_{OUT} < V_{CC} - 0.4V$
6. V_{DD} = 5.5V, V_{IL} = 0.7V (max), $V_{IH} = V_{DD} - 0.7V$ (min), Typ. Val = 1.12 mA/MHz (Not Tested), Ext. 1X CLK, $I_{OL} = I_{OH} = 0$
7. Freq. = 1 MHz
8. Does not apply to OUT2/X2 pin, when configured as crystal oscillator output (X2).
9. Freq. = 1 MHz, but input clock not running. Static IDD current is exclusive of input/output drive requirements and is measured with the clocks stopped and all inputs tied to VDD or VSS, configured to draw minimum current.
10. Applies only during hardware reset for clock configuration options. Strapping current for logic HIGH.
11. Applies only during hardware reset for clock configuration options. Strapping current for logic LOW.
12. Inputs (RTSn, DTRn, TB) with Pullups tested @ V_{in} = 0.0V, V_{DD} = 5.5V

AC Characteristics

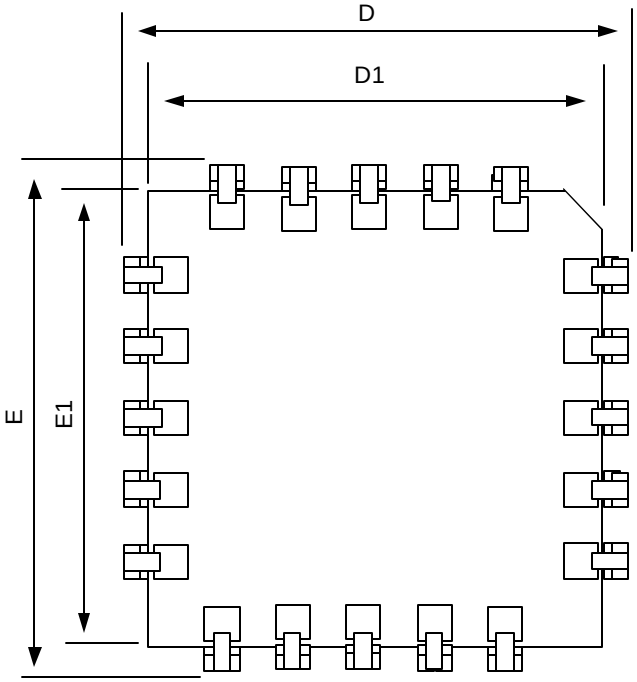
Parameter	Min	Max	Notes
CLK period	54 ns	250 ns	Divide by Two
CLK period	54 ns	108 ns	No Divide by
CLK Low Time	25 ns		
CLK High Time	25 ns		
CLK Rise Time		10 ns	Divide by Two Measured between 0.3 * VDD and 0.7 * VDD
CLK Fall Time		10 ns	Divide by Two Measured between 0.3 * VDD and 0.7 * VDD
CLK Rise Time		15 ns	No Divide by
CLK Fall Time		15 ns	No Divide by
Crystal Frequency	1 Mhz	20 Mhz	
Reset Width	8 * Clock Period		
RTS/DTR Low Setup to Reset inactive	6 * Clock Period		
RTS/DTR Low Hold after Reset inactive		Clock Period – 20 ns	
RDn Active Width	2* clock period + 65 ns		
Address/CSn Setup Time to RDn Active	7 ns		
Address/CSn Hold after RDn Inactive	0 ns		
RDn or WRn Inactive to Active Delay	Clock Period + 15 ns		
Data Out Float Delay after RDn Inactive		40 ns	
WRn Active Width	2 * Clock Period + 15 ns		
Address CSn Setup Time to WRn Active	7 ns		
Address and CSn hold Time after WRn	0 ns		
Data in Setup Time to WRn Inactive	90 ns		
Data In Hold Time after WRn Inactive	12 ns		
SCLK Period	216 ns		16x Clocking Mode
SCLK Period	3500 ns		1x Clocking Mode
RXD Setup Time to SCLK High	250 ns		
RXD Hold Time after SCLK High	250 ns		
TXD Valid after SCLK Low		170 ns	
TXD Delay after RXD		170 ns	Remote Loopback

Packaging Information

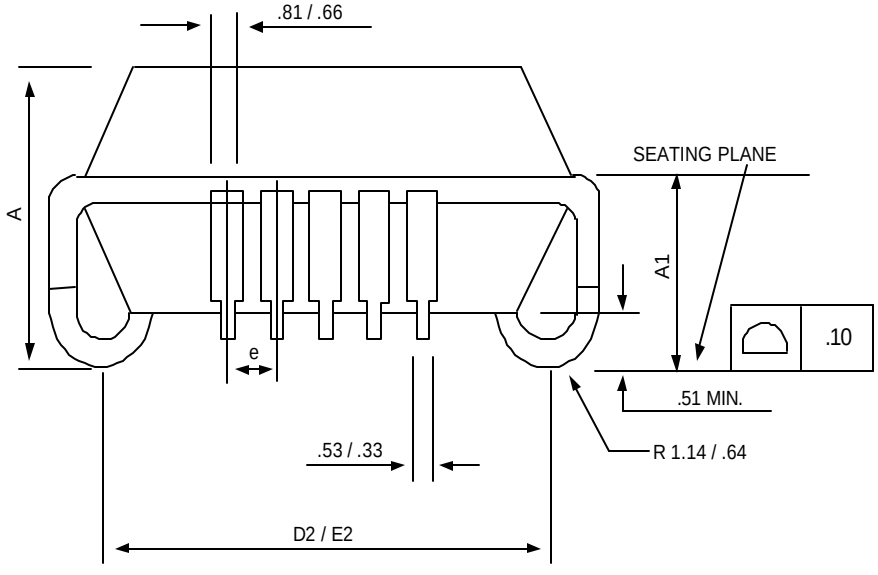
PLCC Package



TOP VIEW



BOTTOM VIEW

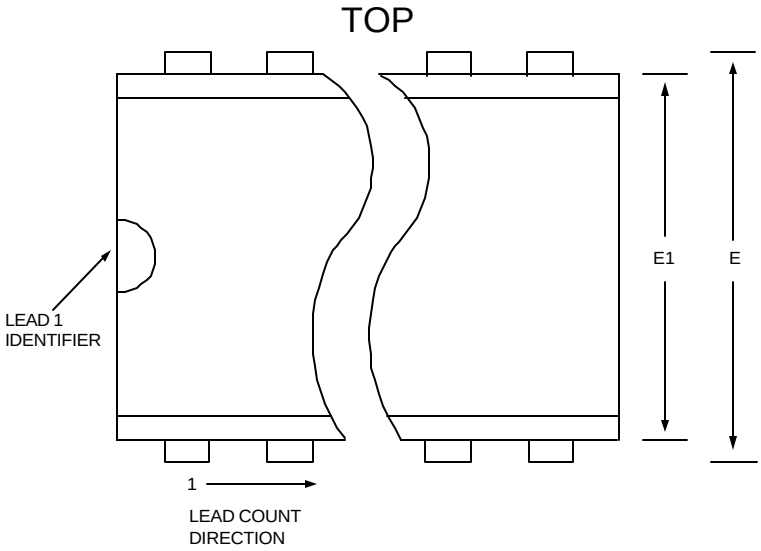


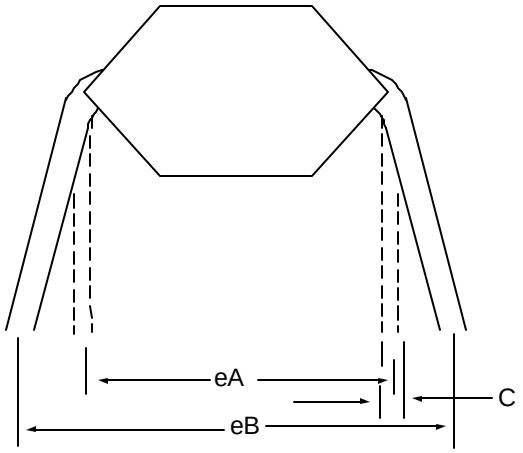
SIDE VIEW

LEAD COUNT

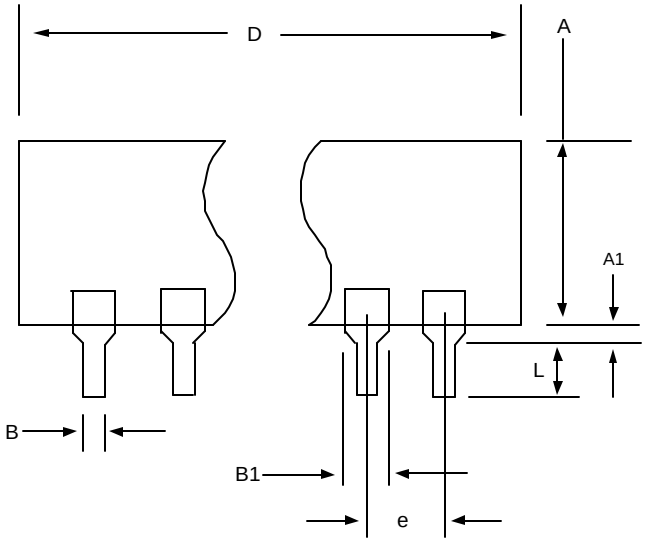
Symbol	28 (in Millimeters)	
	MIN	MAX
A	4.20	4.57
A1	2.29	3.04
D1	11.43	11.58
D2	9.91	10.92
D3	7.62 BSC	
E1	11.43	11.58
E2	9.91	10.92
E3	7.62 BSC	
e	1.27 BSC	
D	12.32	12.57
E	12.32	12.57

PDIP Package





SIDE VIEW (WIDTH)



SIDE VIEW (LENGTH)

Lead Count

Symbol	28 (in Inches)	
	MIN	MAX
A	-	.200
A1	.015	-
B	.015	.020
B1	.050	.070
C	.008	.012
D	1.380	1.470
E	.580	.610
E1	.520	.560
e	.100 TYP	
eA	.580	-
eB	-	.686
L	.100 MIN	
B2	-	-
S	-	-

Ordering Information

The IA82510 may be ordered per the tables below.

Production Version -01

Order Number	Environment	Package Type
IA82510-PDW28I-01	Industrial	28 Lead Plastic DIP, 600 mil wide
IA82510-PLC28I-01	Industrial	28 Lead Plastic Leaded Chip Carrier

OEM Part Number Cross-Reference

The following table identifies which OEM Part Number is compatible with the corresponding InnovASIC Part Number

innovASIC Part Number	Intel Part Number
IA82510-PLC28I	❑ N82510
	❑ TN82510
IA82510-PDW28I	
	❑ P82510
	❑ TP82510

ERRATA

PLEASE NOTE:

- **When using the -01 version of the IA82510, please refer to the errata section, "Production Version -01, Errata".**
- **When using the -00 version of the IA82510, please refer to the errata section, "Production Version -00, Errata".**

Production Version -01, Errata

The following errata are known problems with the -01 version of the IA82510. This is inclusive of all package types and environment grades. A workaround to the identified problem has been provided where possible. **ALL ERRATA LISTED IN PRODUCTION VERION -00 HAVE BEEN FIXED IN THIS VERSION OF THE DEVICE UNLESS OTHERWISE NOTED.**

-00 Errata not fixed in this Production Version:

1. Problem: Device does not operate at 8 MHz in divide-by-one mode

   Analysis: System testing revealed this operational deficiency.

   Workaround: Switch to divide-by-two mode using 2X clock input

New Errata for Production Version -01:

2. Problem: RX FIFO locks up unexpectedly just after configuration and before starting reception.

   Analysis: An RCM command is executed with data of xB8. This is an “enable RX”, “flush RX machine”, “flush RX FIFO”, and “lock RX FIFO” command done in a single instruction. The “flush RX machine” should unlock the RX FIFO, creating a conflict with the simultaneous “lock RX FIFO” command. The original Intel device apparently ignores or gives the “lock RX FIFO” command lower priority in this case. The IA82510 has this priority reversed. Apparently, the application software in this case expected the “lock RX FIFO” command to fail.

   Workaround: Do not execute a “flush RX FIFO” and “lock RX FIFO” command simultaneously. Break up into separate RCM commands.

3. Problem: Unreliable transmits in AUTO TX mode.

   Analysis: Many systems use the RTS output to activate the line transceiver. When the Transmit Mode field in the TMD register is set to semi-auto or automatic mode, RTS is controlled by the TX state machine. On the first character, RTS asserts at the same time as the start bit on the TXD output, whereas the original Intel device asserts RTS a full bit time before assertion of the start bit on TXD. At full temperature range, the width of the start bit

can be altered to the point of confusing the downstream receiver.

Workaround: Change firmware to Manual TX mode to control RTS vs. start of character.

Production Version -00, Errata

The following errata are known problems with the -00 version of the IA82510. This is inclusive of all package types and environment grades. A workaround to the identified problem has been provided where possible.

1. Problem: Scrambled data during boot code shuts down UART, however device works for application code

   Analysis: The RX FIFO is locked, configuration of all registers is done, then the RX FIFO is unlocked just before entering loopback mode in both boot and application code before normal operations begin. Boot code additionally does a blind block read of all registers before normal operations including two reads from the unwritten RX Data FIFO. RX unlock command is inadvertently incrementing the write pointer. For boot code, the two reads of RX data cause the read/write pointers to be permanently out of sync. For application code, the pointers end up synched to the same location, only because the code waits for four characters before reading. This ends up causing an RX overrun, but to our favor because the pointers are now synched.

   Workaround: Execute a "Flush RX FIFO" command (via RCM register) after configuration and block read is complete.

2. Problem: Device does not operate at 8 MHz in divide-by-one mode

   Analysis: System testing revealed this operational deficiency.

   Workaround: Switch to divide-by-two mode using 2X clock input

3. Problem: Setting CLCF to x30, which effectively generates the TX clock from the incoming SCLK signal, kills all transmits.

   Analysis: Configuration of PMD inadvertently set so RI function is selected instead of SCLK function. Original Intel device allows SCLK through anyway, IA82510 suppresses it.

   Workaround: Set correct configuration for PMD allows TX clock generation

4. Problem: Receiving streamed data has many framing errors and corrupt data when connected to some modems.

   Analysis: Shortened stop bit followed immediately by next start bit does not correctly detect that start bit.

   Workaround: Configure external modem to transmit two stop bits

5. Problem: Transmission of streamed data does not return interrupt.
   Analysis: Stray read of GIR sets TX FIFO interrupt hold logic, but this logic does not reset when GER[1] is de-asserted..

   Workaround: Reset logic with write to TX data or avoid stray reads of GIR

6. Problem: Receiving streamed data has many framing errors at fast baud rates (divisor=6) through bad modem lines.
   Analysis: DPLL is not robust for RXD signal with more than 1/16 bit time of variation.
   Workaround: None

7. Problem: Difficulty starting oscillator with crystal.
   Analysis: No internal feedback resistor between X1 and X2.
   Workaround: Install external 1-10Mohm resistor

8. Problem: Intermittent and temperature sensitive crystal oscillator operation when cycling power.
   Analysis: Strapping state elements apparently transparent latches instead of flip flops. If flip flop powers up to wrong state, crystal oscillator is disabled while reset is active. OK after first reset following power-up.
   Workaround: None

9. Problem: Auto-acknowledge of interrupts via writing of LSR does not work.
   Analysis: Writing LSR directly sets/resets bits 4 through 0. Also writing 0 to LSR(0) – RX FIFO – clears the RX FIFO level as seen by FLR. Writing zero to any other LSR bits clears the corresponding LSR/RST flag, but also corrupts the FIFO location the write pointer is set to, then increments both the write and read pointers.
   Workaround: Use other means to service interrupts, such as read of RST or RXD

10. Problem: ICM Status Clear command does not clear LSR/RST overrun error
    Analysis: ICM Status Clear command should clear everything in RST/LSR, MSR, and TMST except RST/LSR(0). Overrun error was missed.
    Workaround: Use other means to service interrupts

11. Problem: In semi-automatic/uLAN mode, the RX FIFO is only opened when an address character matches the ACR1 or ACR0 registers (like full auto mode).
    Analysis: In semi-auto mode, the RX FIFO should open on any address character.
    Workaround: None

12. Problem: Device fails to reset interrupt signal in auto acknowledge mode when character is read from RX FIFO.
    Analysis: RD strobe is outside the CS enable, which is outside of the Intel datasheet, but apparently still works in the Intel device. Such a bus cycle allows the read data out, but fails to generate the necessary internal strobe to change pointers. The same problem is found on write accesses.
    Workaround: Force bus interface to bracket RD strobe inside the CS enable