Basic Table Maintenance Operations

Application Note AN-271

Overview

This application note details the proper execution of various table maintenance operations in the IDT75T43100 Internet Protocol Co-Processor (IPC). The maintenance functions that are presented are the operations of adding, deleting and modifying entries in the IPC's Data and Mask arrays plus storing entries in an external SRAM.

Background Information

The IDT75T43100 IPC is a powerful and efficient peripheral that can be used with all of the various Network Processors (NPs), in addition to custom ASIC lookup solutions, for enhancing the NP's or ASIC's packet processing capabilities. The primary application of the IPC is to assist NPs and ASICs in achieving the high-speed data searching requirements found in routers, the high layer packet processing environments of switches and applications involved in the convergence of voice, data, and video. The IDT75T43100 IPC can support a sustained bandwidth of at least 66 million searches per second.

Of fundamental importance to an IPC is proper data table maintenance: loading, deleting, and changing the various array values within and controlled by the IPC. The IPC is comprised of both a data array and a mask array. Additionally, there can be an external SRAM to hold the associated data to that of the data stored within the IPC entries.

The combination of the two arrays, data and mask, gives the IPC the capability of maintaining a ternary data set that allows for fast, flexible and efficient data searches. Ternary data is comprised of the traditional binary data values of '0' and '1' along with a third value of 'X' or don't care. The IPC Data Array contains the '0' and '1' values while the IPC Mask Array contains the care / don't care information. During a lookup operation both arrays along with a Global Mask Register (GMR) are used to find a match to a requested data set.

To properly maintain data values in the IPC the user must control the values in the Data and Mask Arrays and additionally that of an optional external SRAM when used.

<u>Nomenclature</u>

In the examples detailed in this application note the following structure will be used to illustrate the discussion:

CCLK cycle X	the CCLK cycle that the bus activity is taking place (the CCLK is the lookup cycle)
BUS_NAME[X:Y]	an identification of the bus being discussed
• BUS_FIELD =	description of a data field in BUS_NAME[X:Y] and of the action occurring by the specified field value
t_x	the CLK2X cycle that the bus activity is taking place

At the bottom of every page that a bus diagram is presented, a bus diagram legend is also given. The bus diagram legend is as follows:

☐	= Required Data Field	- data field that the discussion is focused on and is required to be specified as per the discussion
▨	= Optional Parameters	- optional parameters whose value may be selected by the user as per the user's requirements
■	= N/A	- data that is not applicable for the discussion or data bits that are reserved; typically these bits will have a value of '0'

For detailed bus information please see the IDT75T43100 data sheet.

MAY 2001

Add Entry

The Add Entry function is used to load values into the Data or Mask Arrays of the IPC. Typically this will be used for initialization of these two arrays prior to any useful searches of IPC data. This function would also be used for the normal maintenance routine of table updates. To add data the user needs to do a single operation: the Write command.

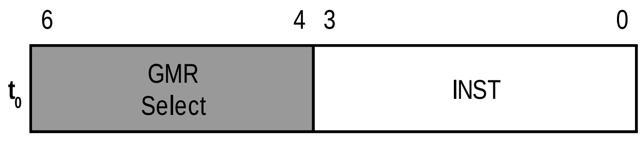
The Write command will be a direct Write in which the address is explicitly presented to the IPC. Additional parameters of the Write command that need to be specified are: 1) Access Type - to determine which array is to be written to (Data Array or Mask Array) and 2) GMR Select - an optional selection as to whether there will be masking of any of the Write bits (no masking will be used for the Add Entry function).

The Write operation is illustrated in the following diagram.

Add Entry - Perform Direct Write:

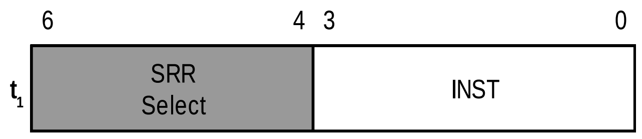
CCLK cycle 1:

INST[6:0]



tbl 01

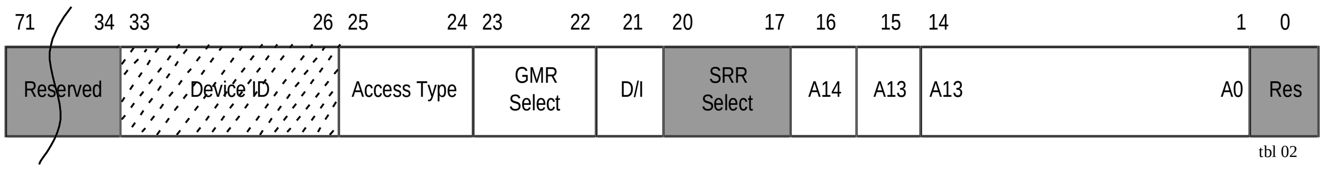
- INST = '0100', Write



tbl 01a

- INST = '0100', Write

REQDATA[71:0]



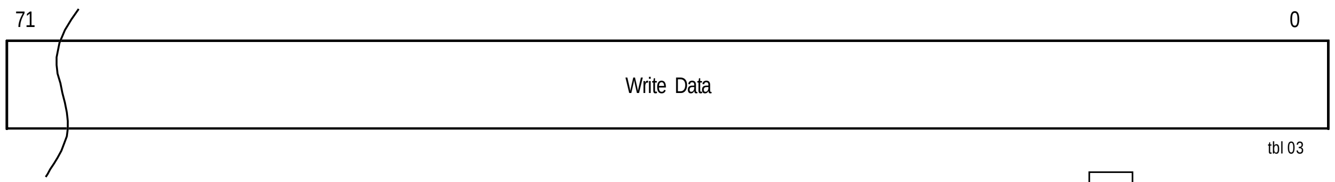
tbl 02

- Access Type = ‘01’, Data Array; or ‘10’, Mask Array
- GMR Select = ‘11’, no mask
- D/I = ‘0’, Direct
- [A14:A0] = <address of entry to be stored>

(Note: A13 must be presented in both REQDATA[15] and REQDATA [14])

CCLK cycle 2:

REQDATA[71:0]



tbl 03

- Write Data = <Data Value to be Stored>

□ = Required Data Field

▨ = Optional Parameters

■ = N/A

Delete Entry

The purpose of a Delete Entry function is to remove a specific entry, that has been previously stored, from the Data Array. If the address of the data is not known, the user must first search the Data Array for the data value needing to be deleted and then once found, remove it from the array. This function requires two operations: a Lookup command followed by a Write command. If the address is known a lookup is not required and a Direct Write command can be simply used.

The Lookup command returns the address of the specified data value stored in the Data Array. The Write command, in an indirect addressing mode, will then take the resulting address from the Lookup, which was stored in a selected Search Result Register (SRR), to point to the entry to be removed. Thus, through use of an Indirect Write the user is able to immediately delete the entry, alleviating the need to read the returned Lookup result prior to proceeding with its deletion.

The Lookup command is issued first and is shown in step 1 of the Delete Entry example. Selection of a GMR specifies which data fields are to be included in the search of the Data Array and is specified during time t_0. During time t_1 the SRR is specified. The x72 Lookup instruction must be present during both times t_0 and t_1. There are seven (7) x72 Lookup GMRs and eight (8) SRRs that can be selected for this operation.

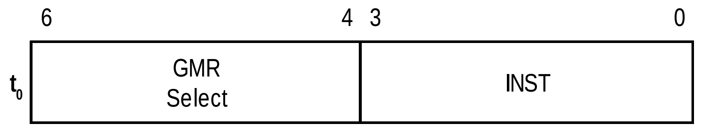
Following the Lookup command is the indirect Write command. The indirect Write is shown in step 2.

An important factor in the Delete Entry function is the timing of the Write command to the Lookup command. For the proper execution of an indirect addressing operation, the operation requires there to be a valid value loaded into the SRR prior to its use. The IDT75T43100 requires an indirect write, using the results of a previous lookup to specify the indirect address, not to be commanded until the 5th CCLK cycle of the Lookup command.

Delete Entry (step 1) – Perform x72 Lookup:

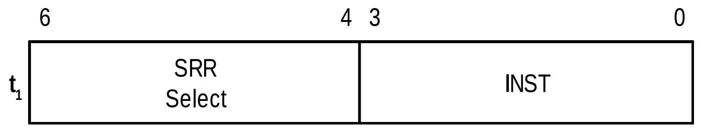
CCLK cycle 1:

INST[6:0]



tbl 04

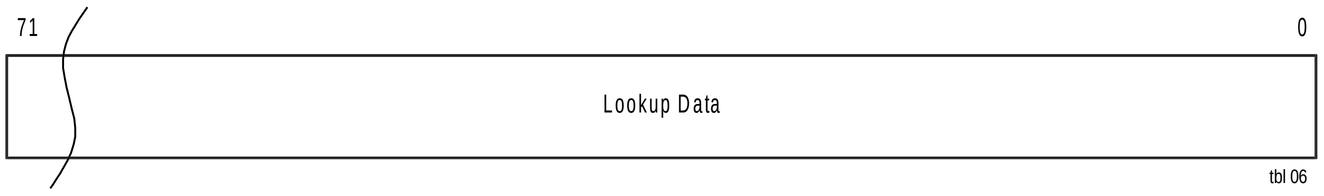
- GMR Select = Select field(s) to parse
- INST = ‘0010’, x72 Lookup



tbl 05

- SRR Select = Select Result Register to store Lookup result
- INST = ‘0010’, x72 Lookup

REQDATA[71:0]



tbl 06

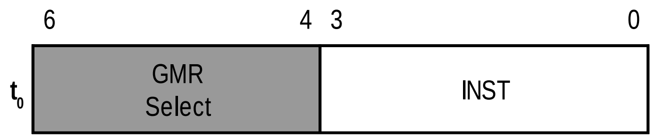
- Lookup Data = Data entry to be deleted

☐	= Required Data Field
▨	= Optional Parameters
■	= N/A

Delete Entry (step 2) - Perform Indirect Write:

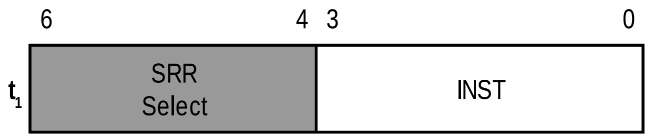
CCLK cycle 5:

INST[6:0]



tbl 07

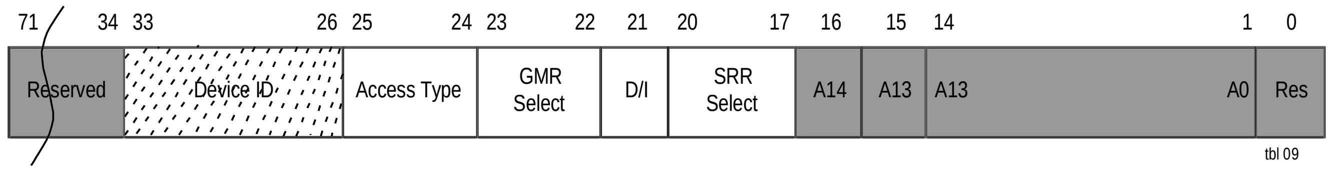
- INST = ‘0100’, Write



tbl 07A

- INST = ‘0100’, Write

REQDATA[71:0]

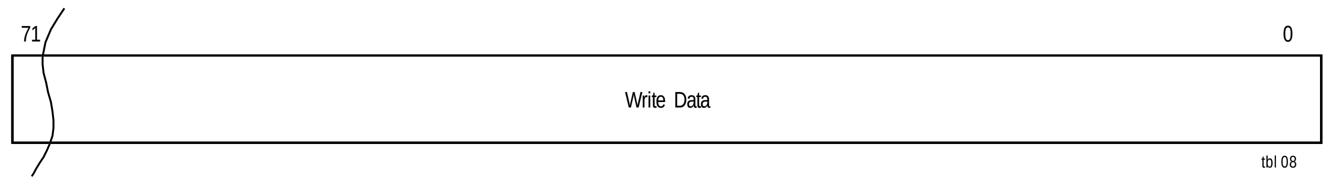


tbl 09

- Access Type = ‘01’, Data Array
- GMR Select = ‘11’, no mask
- D/I = ‘1’, Indirect
- SSR Select = SSR used in Lookup Instruction of step 1

CCLK cycle 6:

REQDATA[71:0]



tbl 08

· Write Data = “000000000000000000”h

☐	= Required Data Field
▨	= Optional Parameters
■	= N/A

Modify Entry

The Modify Entry function follows the same form as the Delete Entry function except, instead of zeroing-out the entire selected data entry, only some of the data is changed. To accomplish this the user will need to specify one of the IPC's write capable GMRs in the indirect Write command. This GMR will provide the mask to determine which bits of the 72-bit data array are to be affected by the Write command.

If the address of the entry to modify is not known, the user must first search the data array for the data value needing to be modified and then once found, modify it. Just as in deleting an entry, this function requires two operations: a Lookup command followed by a Write command. If the address is known a Direct Write can be used to modify the entry in a single step.

The Lookup command is issued first and is shown in step 1. Selection of a Lookup GMR, INST[6:4] during time t_0, specifies which data fields are to be included in the search of the data array. During time t_1 the SRR is specified. The x72 Lookup instruction must be present during both times t_0 and t_1. There are seven (7) x72 Lookup GMRs and eight (8) SRRs that can be selected for use.

Following the Lookup command, the Modify Entry function is completed with an indirect Write command. The indirect Write is shown in step 2. During this step the GMR to use with the Write is specified.

The IPC has a total of 15 GMRs that can be used with its various (x72-bit, x144-bit, x288-bit) Lookup commands to parse which data is to be searched. Only 3 of these can be used to mask a Write command. These are GMRs 10, 11 and 12.

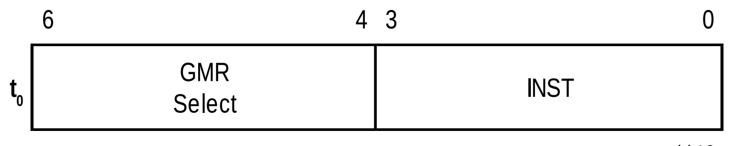
(Note: The use of a GMR is predicated with the fact that the GMR must have been previously loaded with the desired mask prior to its use. Caution must also be exercised when multiple contexts are accessing the IPC's GMR resources so that mask values do not become corrupted.)

Again, the timing of the Lookup command to return a result into the specified SRR to be used in an indirect write must be maintained. The IDT75T43100 requires an indirect write, using the results of a previous lookup to specify the indirect address, not to be commanded until the 5th CCLK cycle of the Lookup command.

Modify Entry (step 1) – Perform x72 Lookup:

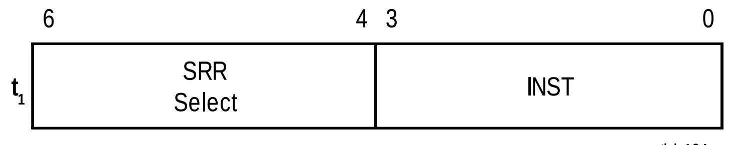
CCLK cycle 1:

INST[6:0]



tbl 10

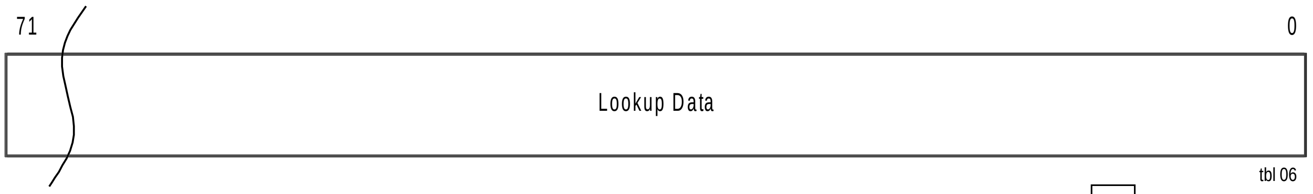
- GMR Select = Select field(s) to parse
- INST = '0010', x72 Lookup



tbl 10A

- SRR Select = Select Result Register to store Lookup result
- INST = '0010', x72 Lookup

REQDATA[71:0]



tbl 06

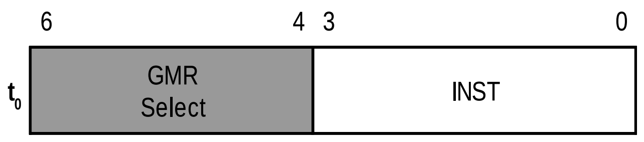
- Lookup Data = Data entry to be modified

☐	= Required Data Field
▨	= Optional Parameters
■	= N/A

Modify Entry (step 2) - Perform Write:

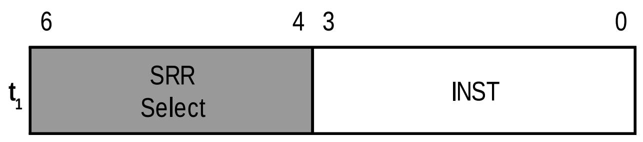
CCLK cycle 5:

INST[6:0]



tbl 11

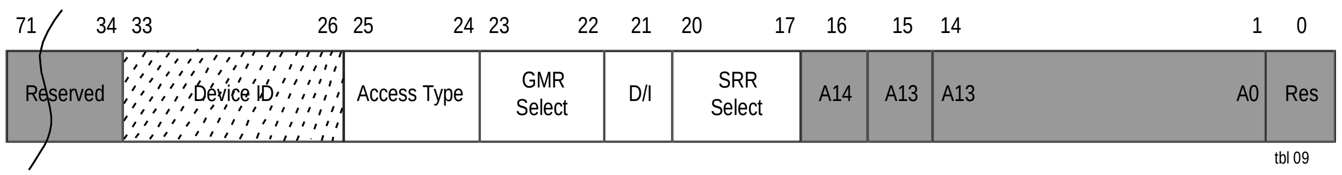
- INST = '0100', Write



tbl 11A

- INST = '0100', Write

REQDATA[71:0]

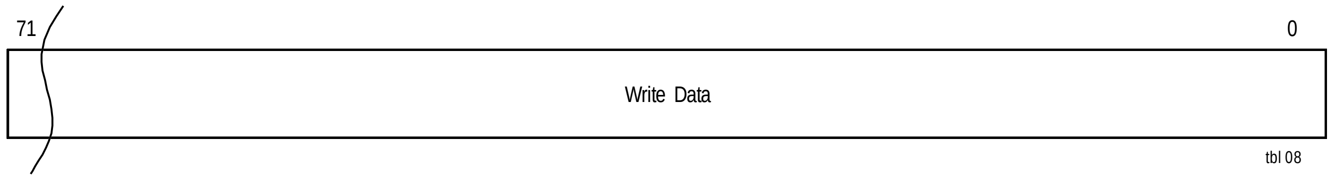


tbl 09

- Access Type = ‘01’, Data Array
- GMR Select = Select specific data field(s) to modify with the Write
- D/I = ‘1’, Indirect
- SSR Select = SSR used in Lookup Instruction of step 1

CCLK cycle 6:

REQDATA[71:0]



tbl 08

- Write Data = <New Data Value>

□ = Required Data Field

▨ = Optional Parameters

■ = N/A

Load External SRAM Entry

The Load External SRAM Entry function is used to load IPC associated data into an external SRAM. This function will be used during initialization of the associated data prior to searches of IPC data as well as at times of normal table maintenance and updates. To load external SRAM data the user needs to do a single operation: the Write command.

The Write command will be an external SRAM Write in which a 20-bit address is explicitly presented to the IPC. The IPC will then take and pass that address to the external SRAM along with the required control signals.

Of key consideration in an external SRAM Write is the coordination of the control and address signals with the data to the external SRAM. Address and control signals are from the IPC. Data is presented by the ASIC/FPGA. When the IPC is presented with the external SRAM Write command, the IPC will transfer the 20-bit address from the Request Data Bus to the Index Bus and generate the necessary SRAM control signals. The data will need to be presented to the SRAM by the ASIC/FPGA when required. (Refer to the IDT75T43100 data sheet for the sequencing and timing requirements of external SRAM accesses.)

Understanding the external SRAM interface:

1) Addresses for external SRAM accesses are formatted differently than for internal IPC accesses. When accessing an external SRAM the IPC is configured in a Bypass Mode in which the IPC passes a 20-bit address from the Request Data Bus to the Index Bus. The format of the REQDATA Bus is different than that of an internal IPC access. (See IDT75T43100 data sheet.)
2) $\overline{\text{CE}}/\overline{\text{OE}}$ and $\overline{\text{WE}}$ are output signals from the IPC that directly drive the appropriate pins of the external SRAM device (ZBT or PBSRAM).
3) Data flow into and out of the external SRAM is not through the IPC but through the ASIC/FPGA. External SRAM accesses can be of whatever data width is chosen for the external SRAM to ASIC/FPGA interface.
4) External SRAM addressing is never specified through indirect addressing.
5) A GMR cannot be used in combination with a Write command to an external SRAM to mask off data bits. (Data written to the external SRAM does not go through the IPC and is not able to be qualified by the IPC.)

Also, please review the IDT75T43100 data sheet sections regarding the System Configuration Register (SCR) especially with respect to the following fields:

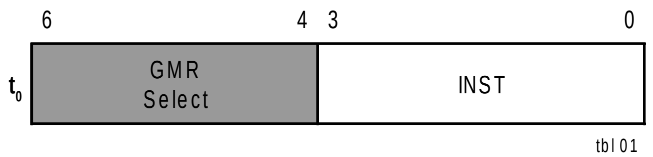
(SR) SRAM Type Bit
(LS) Last SRAM Bit
(PD) Pipeline Delay Bits
(IPC Grp) IPC Group Bits

The external SRAM Write operation is illustrated on the next page in the Add External SRAM Entry example.

Add External SRAM Entry - Perform external SRAM Write:

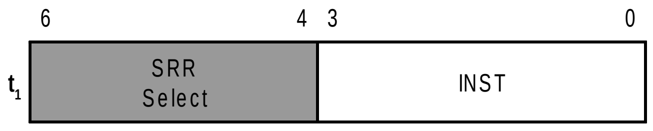
CCLK cycle 1:

INST[6:0]



tbl 01

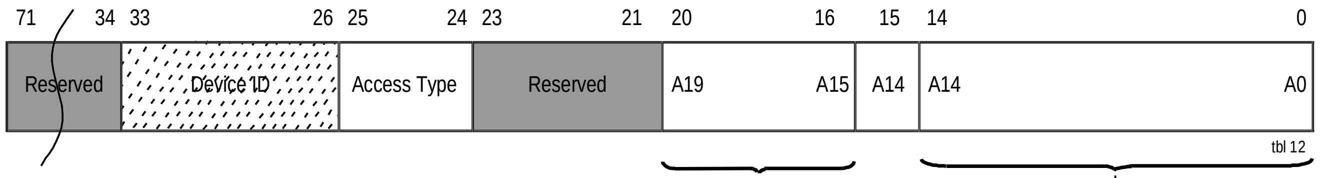
- INST = '0100', Write



tbl 01a

- INST = '0100', Write

REQDATA[71:0]



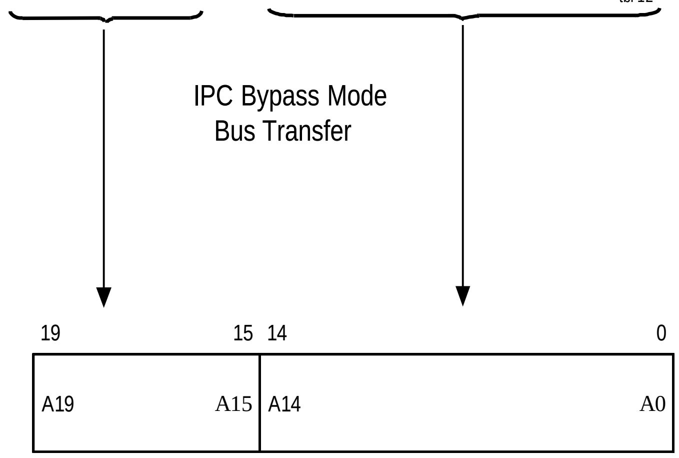
tbl 12

- Access Type = ‘11’, external SRAM
- [A19:A0] = <address of entry to be stored>

(Note: A14 must be presented in both REQDATA[15] and REQDATA [14])

CCLK cycle 6:

INDX[19:0] is driven.



tbl 13

- Address on the REQDATA bus is transferred to the INDX bus.
  REQDATA[20:16],[14:0] $\longrightarrow$ INDX[19:15],[14:0]

SRAM Control Signals $\overline{\text{CE}}$/$\overline{\text{OE}}$ and $\overline{\text{WE}}$ are generated.

CCLK cycle 8:

Write Data is presented to the external SRAM from the ASIC/FPGA on the ZBT Data Bus.

☐	= Required Data Field
▨	= Optional Parameters
■	= N/A

CORPORATE HEADQUARTERS
2975 Stender Way
Santa Clara, CA 95054

for SALES:
831-754-4555
fax: 831-754-4547
www.idt.com

for Tech Support:
ipchelp@idt.com
831-754-4555