

## NicStar Frequently Asked Questions

### General Questions:

#### 1). What's the difference between the IDT 77201 and the IDT 77211?

The IDT 77211 is actually a newer revision of the IDT 77201. It fixed many of the errata items found in the IDT 77201 and is therefore a much cleaner part. The IDT 77211 is pin compatible with the IDT 77201 and is transparent to the software. There isn't really any hardware changes that need to be made aside from eliminating the external circuitry used to correct some of the errata items in the IDT 77201.

#### 2). Are there reference designs available that implements the IDT 77211?

We have several designs that implements the IDT 77211. The IDT 77914, IDT 77915, and the IDT 77916 are all production worthy network interface cards (NICs) that implement the IDT 77211. An evaluation board (IDT 77911) is also provided for testing. These NICs are currently available for purchase and the schematics are also available.

#### 3). Will IDT continue to provide and support the IDT 77201?

No, the 77201 has been obsoleted.

#### 4). What is the IDT 77252 and how is it different from the IDT 77211?

The IDT 77252 is a pin compatible, drop in replacement part of the IDT 77211. Many more features are implemented in the IDT 77252. Few examples of the new features available are hardware ABR implementation, larger receive buffers, and PCI 2.1 support. Please refer to the appendix section in the IDT 77252 User's Manual for a more detailed comparison of the additional features over the IDT 77211.

#### 5). I'm currently using the IDT 77211 and want to eventually make a transition over to the IDT 77252. What design changes will I need to make?

No hardware changes are required since the two parts are pin compatible. However for a full line rate 155Mbps operation, a 80MHz clock is required for the IDT 77252 instead of a 50MHz clock as in the IDT 77211. For a 25Mbps operation, there's no hardware change.

Some software changes are required in making the transition to the IDT 77252. The software will need to incorporate the transmit connection table, rate tables, additional registers, and some bit changes into its code. IDT has contracted a third part software vendor in implementing these changes. Current estimate time to incorporate all the changes varies from one to three man weeks.

**6). What kind of software is available with the IDT 77211?**

IDT has contracted with several third party software vendors to write the drivers for the IDT 77211. These companies and their contact information are provided in the IDT web site under the Network Communication Devices section. IDT also provides several low level drivers for the IDT 77211. These can be downloaded via IDT's web site.

IDT also provides a sample source code for the IDT 77211 driver.

IDT also provides a test software utility called SARWIN. This utility runs in a Windows 3.1 environment and currently is available on the IDT Product CD-ROM.

Finally, additional software is available on the IDT Products CD-ROM and also will soon be added to IDT's web site.

**7). What kind of documents are available for the IDT77211 and how can I get them?**

All the documents for the IDT 77211 are on IDT's web site. These documents include the user's manual, data sheet, and applications notes. The errata list is also available on the IDT web site.

Note: IDT 77211 will be referred to as the NicStar from this point.

**Utopia Questions:**

**1). Is the NicStar Utopia Level 2 compliant?**

No, the NicStar is Utopia Level 1 compliant only.

**2). Can the NicStar operate both as a Utopia master and a slave?**

No, NicStar will operate only as the Utopia master.

**3). I looked at the NicStar data sheet and can't find the signals RxClav or TxClav used for Utopia cell mode. Does the NicStar support both Utopia byte mode and the cell mode?**

Yes, the NicStar supports both Utopia byte mode and cell mode. The NicStar data sheet only shows TxFull# (pin173) and RxEmpty# (pin187). What it should really state for pin 173 is (TxFull# / TxClav) and for pin 187 it should be (RxEmpty# / RxClav). The functionality of these two pins depend upon whether the Utopia bus is operating in a byte or cell mode.

**4). What's the maximum clock input that the NicStar's Utopia interface can handle?**

The maximum clock according to the Utopia specification is 33MHz for a 8-bit data bus and 50MHz for a 16-bit data bus. Since the NicStar's Utopia data bus is 8 bits wide, 33 MHz is the maximum value.

**5). Is an input needed for the PHY\_CLK in order to generated TxClk and RxClk?**

Yes. Both TxClk and RxClk are buffered versions of PHY\_CLK. The skew between the PHY\_CLK and the TxClk and the RxClk is less than 15 ns.

**6). Does the NicStar de-assert TxEnb# during a data transfer cycle in the byte mode and cell mode?**

The NicStar does not de-assert TxEnb# if the Utopia is operating in a cell mode. This is because in a cell mode the PHY layer must be able to accept an entire cell once it asserts TxClav.

If the Utopia is operating in a byte mode, the NicStar will de-assert TxEnb# if it detects TxFull# assertion by the PHY layer. The NicStar will re-assert TxEnb# once TxFull# is de-asserted.

**7). Does the NicStar de-assert RxEnb# during a data transfer cycle in the byte mode and cell mode?**

Yes. The NicStar can de-assert RxEnb# during a data transfer cycle both in the byte mode and the cell mode. In the NicStar, the signal RxEnb# is only dependent upon whether there's room in the RxFifo to accept any more data or not. As long as there's room in this fifo RxEnb# is asserted. When the RxFifo becomes full, the NicStar de-asserts RxEnb# and will re-assert only when additional space becomes available.

**8). When TxFull# is asserted, how many more bytes does the NicStar transmit to the PHY?**

The NicStar samples all input signals on the rising edge of TxClk and therefore will see that TxFull# is asserted on the very next rising edge of TxClk after TxFull# assertion. The NicStar will immediately de-assert TxEnb# at this point. Therefore if a data was already present when TxFull# assertion occurred, then one byte is sent. Otherwise no byte is sent.

**9). When TxFull# is de-asserted, how many more clock cycles does it take to transfer the next byte of data?**

Since the NicStar samples all input signals on the rising edge of TxClk, it won't see that TxFull# is de-asserted until the very next rising edge of TxClk after TxFull# de-assertion. The NicStar will immediately assert TxEnb# at this point. Therefore the maximum clock cycles it takes to transfer the next byte of data after TxFull# de-assertion is one.

**10). Does the NicStar ignore RxData when RxEmpty# is asserted?**

Yes.

**11). Will asserting RxSoc before de-asserting RxEmpty# or before asserting RxClav cause any problems?**

Yes. The Utopia specification doesn't mention this particular event as an invalid case, but it'll cause an incorrect operation in the NicStar.

## VBR Questions:

### 1). I've read the NicStar user's manual and I'm still confused about how the VBR m/n counters work. How does it work?

The m/n counters essentially limits the number of actual cells that are being sent. The m/n values are programmed in the Transmit Buffer Descriptors (TBDs). The "m" value specifies a 3 bit field and represents how many cells will be burst back to back when the n counter rolls over. The "n" value specifies a 7 bit field and represents a divisor of the basic cell line rate. So for every "n" cell time the NicStar will transmit "m" cells.

For example:

If  $m/n = 1/1$ , then the data transmission is at 155.52 Mbps

If  $m/n = 1/2$ , then the data transmission is at 77.76 Mbps since one cell is sent and the other cell time is not used to transmit a data from this channel.

If  $m/n = 2/127$ , then the data transmission is at 2.45 Mbps since two cells are sent and the remaining 125 cell times is not used to transmit a data from this channel.

### 2). Will setting m/n with different values, but keeping the their ratios same (i.e. $m/n = 1/4$ , $m/n = 2/8$ , $m/n = 4/16$ , $m/n = 8/32$ , etc) yield the same result?

Well it depends upon the PDU size since the m/n value is reset every time a new PDU is obtained even though the "n" counter may not have completed its countdown.

For example if the PDU size is 4 cells or 192 bytes, then:

$m/n = 1/4$  would result in 25% of the full line rate.

$m/n = 2/8$  would result in 25% of the full line rate.

$m/n = 4/16$  would result in 25% of the full line rate.

$m/n = 8/32$  would result in full line rate since getting the 5th cell translates to getting a new PDU.

### 3). If there are two VBR channels, let's say one channel transmitting at ( $m/n = 1/1$ ) and the other channel simultaneously transmitting at ( $m/n = 1/6$ ), then once the channel with $m/n = 1/6$ begins transmitting, do cells from the $m/n = 1/1$ channel get inserted in the spaces while the channel with $m/n = 1/6$ countdown is occurring?

Yes. However, two VBR queues cannot simultaneously be transmitting. For example, if the high priority queue has the  $m/n = 1/6$  and the medium priority queue has the  $m/n = 1/1$ , then the data from the medium priority queue isn't transmitted until the cell from the high priority queue is sent (i.e. the high priority queue starts counting down).

### 4). Consider one VBR queue with two PDUs from different VCIs in the queue, both having $m/n = 1/2$ . Do cells from both VCIs get interleaved?

No, the NicStar withholds sending the second PDU until the first PDU is completely sent first.

**5). Consider one VBR queue with two PDUs from the same VCIs in the queue, both having  $m/n = 1/2$ . Does the NicStar withhold sending the second PDU until the first is completely sent?**

Yes.

**6). Combination of the previous two cases. With 3 PDUs in the queue, all with  $m/n = 1/2$  in the following order: VCIx, VCIx, VCIy. Do cells from the first and the third PDU get interleaved while the cells from the second one are deferred until completion of the first one?**

No it doesn't. If many VCIs are in the same queue, then one slow connection will slow down the other connections also. Therefore three VBR queues are provided.

### **General Technical Questions:**

**1). What's the difference between "idle" cells and "null cells"? What is the actual value of these cells?**

There is no difference between these two terms. They all refer to the same thing. The values of these cells are all 0's.

**2). Is there any way to disable sending Null cells?**

No there isn't. However an external logic can be used to prevent "idle" cells from reaching the PHY device.

**3). I'd like to be able to use different VPI/VCI bits than what's specified in bits [19:18] of the NicStar Configuration register. How would I achieve this?**

Unfortunately the user is limited to what's specified in the Configuration register of the NicStar user's manual.

**4). Can the NicStar receive and generate OAM cells?**

Yes. A clear description is provided in the appendix section of the NicStar user's manual.

**5). Can the NicStar begin segmentation from a non-word aligned buffer?**

No, the transmit buffer must point to a word aligned buffer.

**6). Does inserting a TSR in the SCQ occupy a cell time?**

No it does not.

**7). If there is less than 48 bytes remaining in a particular receive buffer, will part of next cell to be placed be stored in the remaining memory space for that particular buffer?**

No, the NicStar will grab another receive buffer and store the cell there.

**8). If the last cell of the PDU is deposited into a receive buffer and that buffer still has many spaces available, will the remaining space of that buffer be used to store the next cells of another PDU?**

No, as soon as the last cell of the PDU is stored into a particular buffer, the rest of the memory space of that particular space is not used and therefore wasted. The worst case scenario would be if a large free buffer size was 16384 bytes and the last cell of a PDU was placed into the starting location of that large free buffer.

**9). What are the power consumption values of the NicStar?**

The typical value is 250mA and the maximum value is 300mA.

**10). Is a low voltage part available?**

No, the NicStar is only a 5V part.

**11). What's the EPROM and EEPROM used for? Are they really needed? Who do I contact to obtain more information?**

The EPROM and the EEPROM are optional devices and are not required. The EPROM can be used to connect a network boot time driver. The EEPROM can be used to store any information that the designer chooses to. One example would be to use it to store MAC addresses.

For the EEPROM, IDT recommends the user use Xicor X25020. Please contact Xicor for more information.

**12). Which clock does the Utility bus use? What about the SRAM bus?**

Both the Utility bus and the SRAM use the SAR\_CLK. For a 155Mbps operation the 50MHz clock input to the NicStar is divided by two internally. This divided by two clock that's used to clock all the internal state machines is also used to clock the Utility bus and the SRAM.

**13). I still have some more questions. Whom do I sent it to?**

You can call the ATM hot line at (408)953-2529 or email to atmhelp@idt.com