Integrated Device Technology, Inc.

155 Mbps ATM SAR CONTROLLER FOR PCI-BASED NETWORKING APPLICATIONS

PRELIMINARY INFORMATION
IDT77201

KEY FEATURES

- Full-duplex Segmentation and Reassembly (SAR) at 155 Mbps "wire-speed" (310 Mbps aggregate speed).
- Performs ATM layer protocol functions.
- Supports AAL5, AAL3/4, "AAL0" and "Raw Cell" formats.
- Supports Constant Bit Rate (CBR), Available Bit Rate (ABR), Variable Bit Rate (VBR) and Unassigned Bit Rate (UBR) service classes.
- Reassembles received CS-PDUs directly into host memory.
- Segments CS-PDUs ready for transmission directly from host memory.
- PCI bus master interface for efficient, low latency DMA transfers with host system.
- Operates with ATM networks up to 155.52 Mbps.
- Up to 16 million open transmit connections.
- Up to 16K simultaneous receive connections.
- Glue-less integration to host system's PCI bus.
- UTOPIA Interface to PHY.
- Utility & Management Interface to PHY.
- Standalone controller: embedded processor not required.
- Supports high-performance, lowest-cost ATM NIC solution.

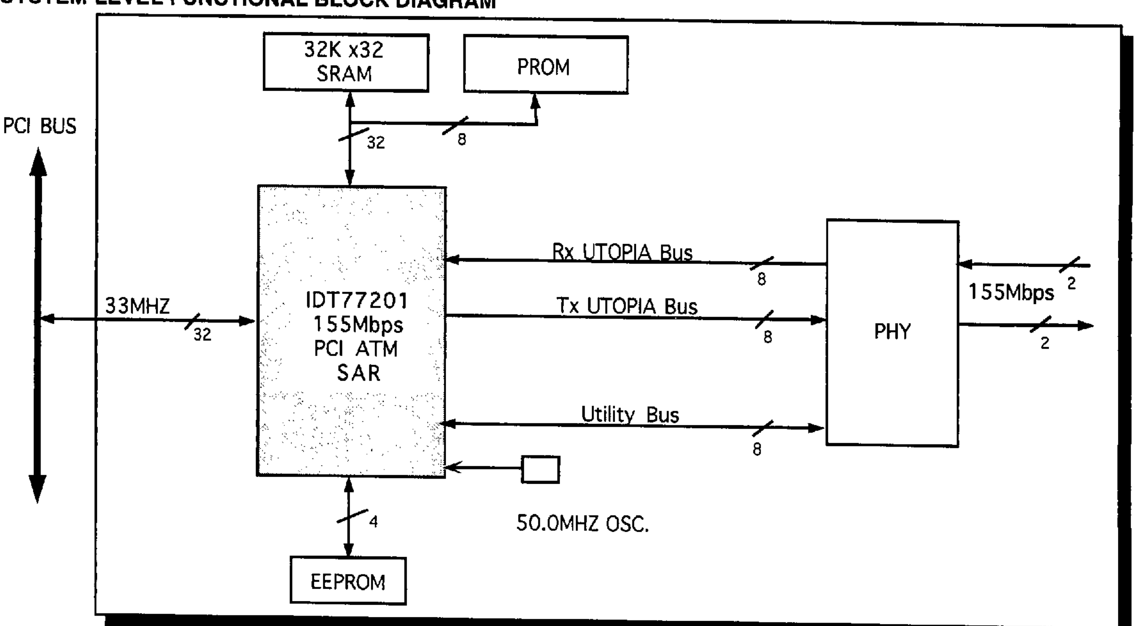
DESCRIPTION

The IDT77201 NICStAR™ is a member of IDT's family of products for Asynchronous Transfer Mode (ATM) networks. The NICStAR performs both the ATM Adaption Layer (AAL) Segmentation and Reassembly (SAR) function and the ATM layer protocol functions.

A Network Interface Card (NIC) or internetworking product based on the NICStAR uses host memory, rather than local memory, to reassemble Convergence Sublayer Protocol Data Units (CS-PDUs) from ATM cell payloads received from the network. When transmitting, as CS-PDUs become ready, they are queued in host memory and segmented by the NICStAR into ATM cell payloads. From this, the NICStAR then creates complete 53-byte ATM cells which are sent through the network. The NICStAR's on-chip PCI bus master interface provides efficient, low latency DMA transfers with the host system, while it's UTOPIA interface provides direct connection to PHY components used in 25.6 Mbps to 155 Mbps ATM networks.

The IDT77201 is fabricated using state-of-the-art CMOS technology, providing the highest levels of integration, performance and reliability, with the low-power consumption characteristics of CMOS.

SYSTEM-LEVEL FUNCTIONAL BLOCK DIAGRAM

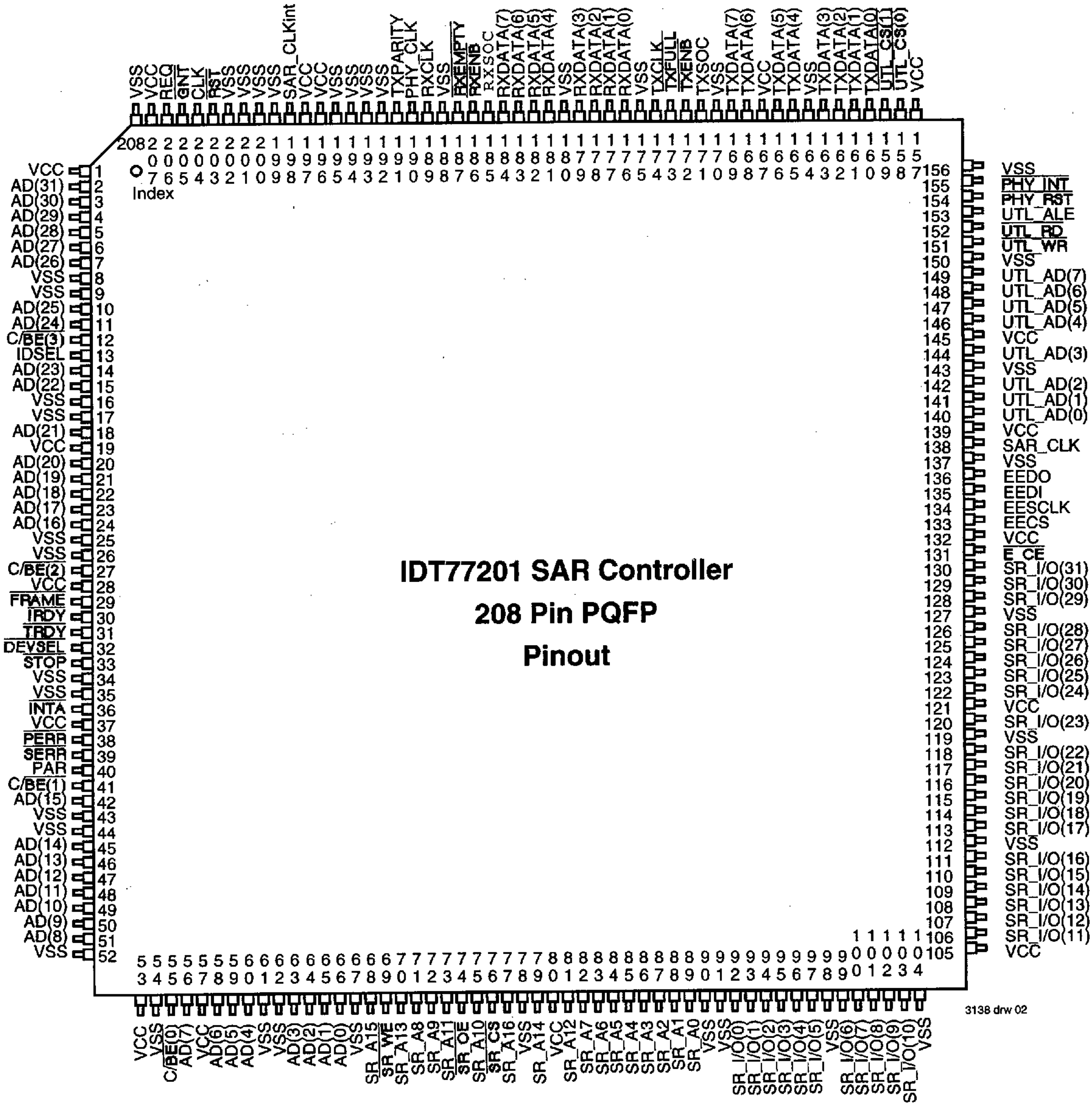


3138 drw 01

NICStAR is a trademark and the IDT logo is a registered trademark of Integrated Device Technology, Inc.

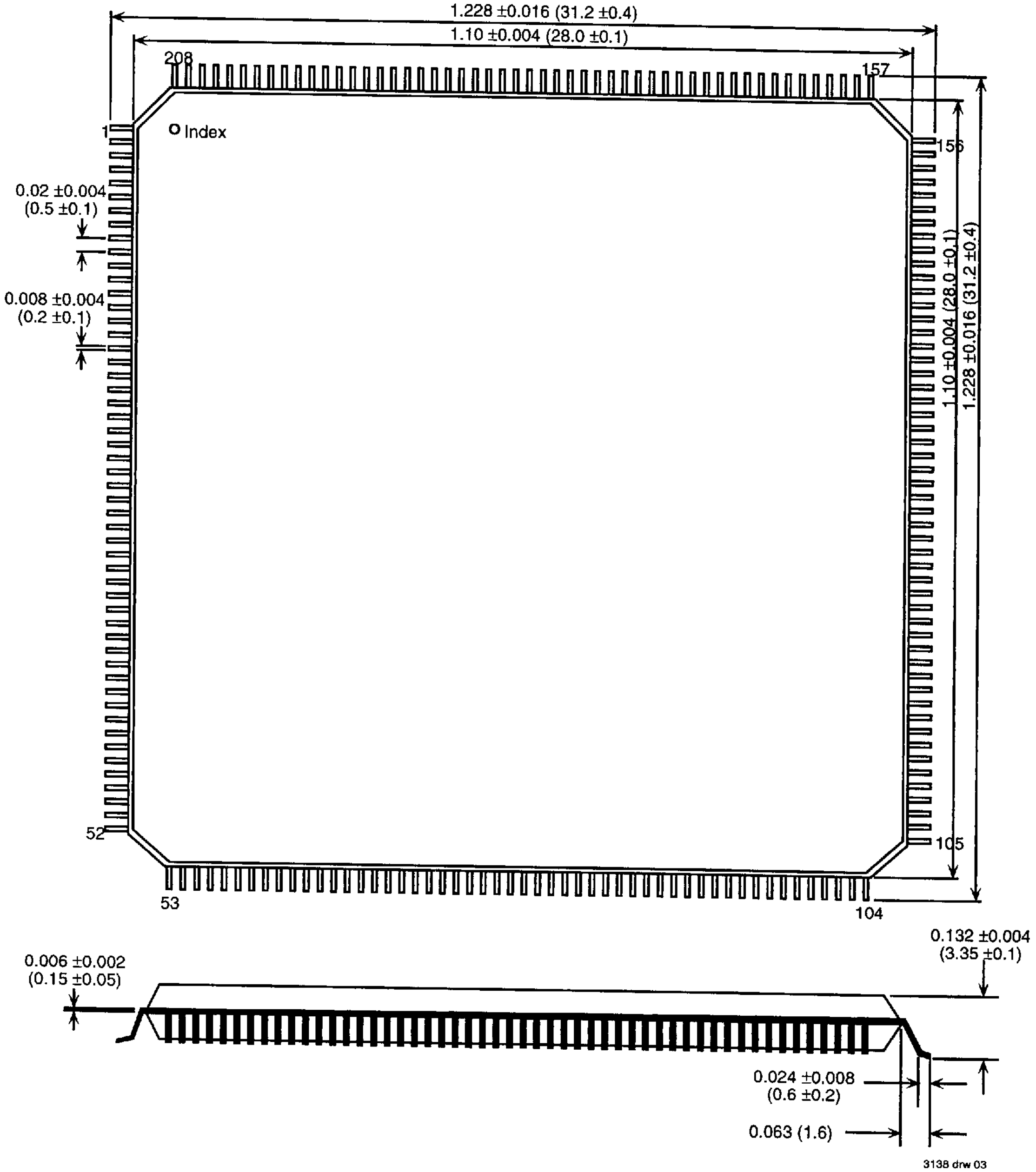
For latest information contact IDT's web site at www.idt.com or fax-on-demand at 408-492-8391.

PACKAGE PINOUT



3138 drw 02

PACKAGE DRAWING



PIN DEFINITIONS

Pin #	Name	I/O	Bus Name	Description
1	Vcc	I	power	
2	AD(31)	I/O	PCI	address/data line
3	AD(30)	I/O	PCI	address/data line
4	AD(29)	I/O	PCI	address/data line
5	AD(28)	I/O	PCI	address/data line
6	AD(27)	I/O	PCI	address/data line
7	AD(26)	I/O	PCI	address/data line
8	Vss	I	power	
9	Vss	I	power	
10	AD(25)	I/O	PCI	address/data line
11	AD(24)	I/O	PCI	address/data line
12	C/BE(3)*	I/O	PCI	bus command
13	IDSEL	I	PCI	
14	AD(23)	I/O	PCI	address/data line
15	AD(22)	I/O	PCI	address/data line
16	Vss	I	power	
17	Vss	I	power	
18	AD(21)	I/O	PCI	address/data line
19	Vcc	I	power	
20	AD(20)	I/O	PCI	address/data line
21	AD(19)	I/O	PCI	address/data line
22	AD(18)	I/O	PCI	address/data line
23	AD(17)	I/O	PCI	address/data line
24	AD(16)	I/O	PCI	address/data line
25	Vss	I	power	
26	Vss	I	power	
27	C/BE(2)*	I/O	PCI	bus command
28	Vcc	I	power	
29	Frame*	I/O	PCI	cycle frame
30	IRDY*	I/O	PCI	initiator ready
31	TRDY*	I/O	PCI	target ready
32	DEVSEL*	I/O	PCI	target indicating address decode
33	STOP*	I/O	PCI	target requesting master to stop
34	Vss	I	power	
35	Vss	I	power	
36	INTA*	O	PCI	"interrupt ""A"" request "
37	Vcc	I	power	
38	PERR*	I/O	PCI	data parity error
39	SERR*	O	PCI	system error
40	PAR	I/O	PCI	parity (for AD[0:31] and C/BE[0:3])
41	C/BE(1)*	I/O	PCI	bus command
42	AD(15)	I/O	PCI	address/data line
43	Vss	I	power	
44	Vss	I	power	
45	AD(14)	I/O	PCI	address/data line
46	AD(13)	I/O	PCI	address/data line
47	AD(12)	I/O	PCI	address/data line
48	AD(11)	I/O	PCI	address/data line

PIN DEFINITIONS (CONT)

Pin #	Name	I/O	Bus Name	Description
49	AD(10)	I/O	PCI	address/data line
50	AD(9)	I/O	PCI	address/data line
51	AD(8)	I/O	PCI	address/data line
52	Vss	I	power	
53	Vcc	I	power	
54	Vss	I	power	
55	C/BE(0)*	I/O	PCI	bus command
56	AD(7)	I/O	PCI	address/data line
57	Vcc	I	power	
58	AD(6)	I/O	PCI	address/data line
59	AD(5)	I/O	PCI	address/data line
60	AD(4)	I/O	PCI	address/data line
61	Vss	I	power	
62	Vss	I	power	
63	AD(3)	I/O	PCI	address/data line
64	AD(2)	I/O	PCI	address/data line
65	AD(1)	I/O	PCI	address/data line
66	AD(0)	I/O	PCI	address/data line
67	Vss	I	power	
68	SR_A15	O	SRAM	Address line
69	SR_WE*	O	SRAM	Write enable
70	SR_A13	O	SRAM	Address line
71	SR_A8	O	SRAM	Address line
72	SR_A9	O	SRAM	Address line
73	SR_A11	O	SRAM	Address line
74	SR_OE*	O	SRAM	Output Enable control
75	SR_A10	O	SRAM	Address line
76	SR_CS*	O	SRAM	Chip Select
77	SR_A16	O	SRAM	Address line
78	Vss	I	power	
79	SR_A14	O	SRAM	Address line
80	Vcc	I	power	
81	SR_A12	O	SRAM	Address line
82	SR_A7	O	SRAM	Address line
83	SR_A6	O	SRAM	Address line
84	SR_A5	O	SRAM	Address line
85	SR_A4	O	SRAM	Address line
86	SR_A3	O	SRAM	Address line
87	SR_A2	O	SRAM	Address line
88	SR_A1	O	SRAM	Address line
89	SR_A0	O	SRAM	Address line
90	Vss	I	power	
91	Vss	I	power	
92	SR_I/O(0)	I/O	SRAM	Data input/output line
93	SR_I/O(1)	I/O	SRAM	Data input/output line
94	SR_I/O(2)	I/O	SRAM	Data input/output line
95	SR_I/O(3)	I/O	SRAM	Data input/output line
96	SR_I/O(4)	I/O	SRAM	Data input/output line

8

PIN DEFINITIONS (CONT)

Pin #	Name	I/O	Bus Name	Description
97	SR_I/O(5)	I/O	SRAM	Data input/output line
98	Vss	I	power	
99	SR_I/O(6)	I/O	SRAM	Data input/output line
100	SR_I/O(7)	I/O	SRAM	Data input/output line
101	SR_I/O(8)	I/O	SRAM	Data input/output line
102	SR_I/O(9)	I/O	SRAM	Data input/output line
103	SR_I/O(10)	I/O	SRAM	Data input/output line
104	Vss	I	power	
105	Vcc	I	power	
106	SR_I/O(11)	I/O	SRAM	Data input/output line
107	SR_I/O(12)	I/O	SRAM	Data input/output line
108	SR_I/O(13)	I/O	SRAM	Data input/output line
109	SR_I/O(14)	I/O	SRAM	Data input/output line
110	SR_I/O(15)	I/O	SRAM	Data input/output line
111	SR_I/O(16)	I/O	SRAM	Data input/output line
112	Vss	I	power	
113	SR_I/O(17)	I/O	SRAM	Data input/output line
114	SR_I/O(18)	I/O	SRAM	Data input/output line
115	SR_I/O(19)	I/O	SRAM	Data input/output line
116	SR_I/O(20)	I/O	SRAM	Data input/output line
117	SR_I/O(21)	I/O	SRAM	Data input/output line
118	SR_I/O(22)	I/O	SRAM	Data input/output line
119	Vss	I	power	
120	SR_I/O(23)	I/O	SRAM	Data input/output line
121	Vcc	I	power	
122	SR_I/O(24)	I/O	SRAM	Data input/output line
123	SR_I/O(25)	I/O	SRAM	Data input/output line
124	SR_I/O(26)	I/O	SRAM	Data input/output line
125	SR_I/O(27)	I/O	SRAM	Data input/output line
126	SR_I/O(28)	I/O	SRAM	Data input/output line
127	Vss	I	power	
128	SR_I/O(29)	I/O	SRAM	Data input/output line
129	SR_I/O(30)	I/O	SRAM	Data input/output line
130	SR(I/O(31)	I/O	SRAM	Data input/output line
131	E_CE*	O	EPROM	EPROM chip select
132	Vcc	I	power	
133	EECS	O	EEPROM	chip select
134	EESCLK	O	EEPROM	clock
135	EEDI	I	EEPROM	Data input
136	EEDO	O	EEPROM	Data output
137	Vss	I	power	
138	SAR_CLK	I		SAR clock input
139	Vcc	I	power	
140	UTL_AD(0)	I/O	Utility	address/data bus
141	UTL_AD(1)	I/O	Utility	address/data bus
142	UTL_AD(2)	I/O	Utility	address/data bus
143	Vss	I	power	
144	UTL_AD(3)	I/O	Utility	address/data bus

PIN DEFINITIONS (CONT)

Pin #	Name	I/O	Bus Name	Description
145	Vcc	I	power	
146	UTL_AD(4)	I/O	Utility	address/data bus
147	UTL_AD(5)	I/O	Utility	address/data bus
148	UTL_AD(6)	I/O	Utility	address/data bus
149	UTL_AD(7)	I/O	Utility	address/data bus
150	Vss	I	power	
151	UTL_WR*	O	Utility	write control
152	UTL_RD*	O	Utility	read control
153	UTL_ALE	O	Utility	address latch enable
154	PHY_RST*	O	PHY	reset control
155	PHY_INT*	I	PHY	interrupt input from PHY
156	Vss	I	power	
157	Vcc	I	power	
158	UTL_CS(0)*	O	Utility	chip select (0)
159	UTL_CS(1)*	O	Utility	chip select (1)
160	TxData(0)	O	UTOPIA	transmit data line
161	TxData(1)	O	UTOPIA	transmit data line
162	TxData(2)	O	UTOPIA	transmit data line
163	TxData(3)	O	UTOPIA	transmit data line
164	Vss	I	power	
165	TxData(4)	O	UTOPIA	transmit data line
166	TxData(5)	O	UTOPIA	transmit data line
167	Vcc	I	power	
168	TxData(6)	O	UTOPIA	transmit data line
169	TxData(7)	O	UTOPIA	transmit data line
170	Vss	I	power	
171	TxSOC	O	UTOPIA	start of cell
172	TxEnb*	O	UTOPIA	transmit enable control
173	TxFull*	I	UTOPIA	transmit buffer full
174	TxCLK	O	UTOPIA	transmit data sync clock
175	Vss	I	power	
176	RxData(0)	I	UTOPIA	receive data line
177	RxData(1)	I	UTOPIA	receive data line
178	Rxdata(2)	I	UTOPIA	receive data line
179	RxData(3)	I	UTOPIA	receive data line
180	Vss	I	power	
181	RxData(4)	I	UTOPIA	receive data line
182	RxData(5)	I	UTOPIA	receive data line
183	RxData(6)	I	UTOPIA	receive data line
184	RxData(7)	I	UTOPIA	receive data line
185	RxSOC	I	UTOPIA	start of cell
186	RxEnb*	O	UTOPIA	receive enable control
187	RxEmpty*	I	UTOPIA	receive buffer empty
188	Vss	I	power	
189	RxClk	O	UTOPIA	receive data sync clock
190	PHY_Clk	I	UTOPIA	Transmit sync clock input
191	TxParity	O	UTOPIA	transmit data parity bit
192	Vss	I	power	

PIN DEFINITIONS (CONT)

Pin #	Name	I/O	Bus Name	Description
193	Vss	I	power	
194	Vss	I	power	
195	Vss	I	power	
196	Vcc	I	power	
197	Vcc	I	power	
198	SAR_CLKint	O		SAR_Clk divided by 2
199	Vss	I	power	
200	Vss	I	power	
201	Vss	I	power	
202	Vss	I	power	
203	RST*	I	PCI	system bus reset
204	CLK	I	PCI	bus clock
205	GNT*	I	PCI	bus grant signal from arbiter
206	REQ*	O	PCI	bus request
207	Vcc	I	power	
208	Vss	I	power	

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Min.	Max.	Unit
VCC	Supply voltage	-0.3	6.5	V
Vin	InputVoltage	VSS-0.3	VCC+0.3	V
Vout	Output Voltage	VSS-0.3	VCC+0.3	V
Tstg	Storage Temperature	0	125	deg. C

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Min.	Max.	Unit
VCC	Supply voltage	4.75	5.25	V
Vi	Input Voltage	0	VCC	V
Ta	Operating temperature	0	70	deg. C
titr	Input TTL rise time	-	2	ns
titf	Input TTL fall time	-	2	ns

CAPACITANCE

Symbol	Parameter	Condition	Min.	Max.	Typical	Unit
Cin	Input Capacitance	except PCI Bus	-	-	4	pF
Cout	Output Capacitance	all outputs	-	-	6	pF
Cbid	Bi-Direactional Capacitance	all bi-directional pins	-	-	10	pF
Cinpci	PCI Bus Input Capacitance	PCI Bus inputs	-	10	-	pF
Cclkpci	PCI Bus Clock Input	-	5	12	-	pF
Cidsel	PCI Bus ID Select Input	-	-	8	-	pF

8

DC OPERATING CONDIONS

Symbol	Parameter	Condition	Min.	Max.	Typical	Unit
Vil	Low-level TTL input voltage	-	-	0.8	-	V
Vih	High-level TTL input voltate	-	2	-	-	V
Vol	Low-level TTL Output voltage	except PCI Bus	-	0 4	-	V
Vol	PCI Bus Low-level TTL output	PCI Bus voltage	-	0.55	-	V
Voh	High-level TTL output voltage	-	2.4	-	-	V
Iol	Low-level TTL output current: SR_A16-0	VSS+0.4V	-	-	12	mA
Ioh	High-level TTL output current: SR_A16-0	Vdd-0.4V	-	-	-4	mA
Iol	Low-level TTL output current: RxEnb#, RxClk, TxSOC, TxData 7-0, TxEnb#, TxParity, TxClk, WE#, OE#, CS#, SR_D31-0	VSS+0.4V	-	-	6	mA
Ioh	High-level TTL output current: RxEnb#, RxClk, TxSoc, TxData7-0, TxEnb#, TxPariety, TxClk, WE#, OE#, CS#, SR_D31-0	Vdd-0.4V	-	-	-2	mA
Iol	Low-level TTL output current: UTL_AD7-0, UTL_RD#, UTL_WR#, UTL_ALE#, UTL_CS1/2#, EESCLK, EECS, EEDO, PHY_RST#	Vss+0.4V	-	-	3	mA
Ioh	High-level TTL output current: UTL_AD7-0, UTL_RD#, UTL_WR#, UTL_ALE#, UTL_CS1/2#, EESCLK, EECS, EEDO, PHY_RST#	Vss+0.4V	-	-	-1	mA
Iil	Input leakage current	-	-1	1	-	uA
Ityp	Dynamic Supply Current	-	-	-	TBD	mA

PCI BUS

Symbol	Parameter	Min.	Max.	Units
tval	CLK to Output Signal Valid Delay: AD31-0, C/BE3-0, PAR, FRAME#, IRDY#, DEVSE;#. TRDU#. STOP#. PERR#. SERR#	-	11	ns
tval(ptp)	CLK to Output Signal Valid Delay: REQ#	-	12	ns
ton	Float to Signal Active Delay: AD31-0, C/BE3-0, PAR, FRAME#, IRDY#, DEVSEL#, TRDY#, STOP#, RERR#, SERR#	2	-	ns
toff	Signal Active to Float Delay: AD31-0, C/BE3-0, PAR, FRAME#, IRDY#, DEVSEL#, TRDY#, STOP#, RERR#, SERR#	-	28	ns
tsu	Input Setup Time to CLK: AD31-0, C/BE3-0, PAR, FRAME#, IRDY#, DEVSEL#, TRDY#, STOP#, RERR#, SERR#, GNT#, IDSEL#	7	-	ns
tsu(ptp)	Input Setup Time to CLK: GNT#	10	-	ns
th	Input Hold Time from CLK: AD31-0,C/BE3-0, PAR, FRAME#, IRDY#, DEVSEL#, TRDY#, STOP#, PERR#, SERR#, GNT#, IDSEL#	0	-	ns
trst-pwr	Reset Active Time After Power Stable	1	-	ms
trst-clk	Reset Active Time After CLK Stable	100	-	ns
trst-off	Reset Active to Output Float Delay:AD31-0, C/B3-0, PAR, FRAME#, IRDY#, DEVSEL#, TRDYU#, STOP#, PERR#, SERR#	-	40	ns

UTOPIA BUS

Symbol	Parameter	Min.	Max.	Units
tl	TxClk, RxClk Delay from PHY_CLK	-	15	ns
t2	TxData7-0, TxSOC, TAxEnb#, TxParity Output Valid from TxClk	-	20	ns
t3	TxFull#/TAxCLAV Setup Time to ExClk	10	-	ns
t4	TxFull#/TxCLAV Hold Time from TxClk	0	-	ns
t5	RxEnb# Output Valid from RxClk	-	20	ns
t6	RxData7-0, RxSOC Setup Time to RxClk	10	-	ns
t7	RxData7-0, RxSOC Hold Time from RxClk	0	-	ns
t8	RxEmpty# Setup Time to RxClk	10	-	ns
t9	RxEmpty# Hold Time from RxClk	0	-	ns

UTILITY BUS WRITE CYCLE

Symbol	Parameter	Min.	Max.	Units
twl	UTL_ALE Pulse Width	25	-	ns
tw2	UTL_CS1/2# Output Valid to UTL_ALE falling edge	25	-	ns
tw3	UTL_WR# Output Valid from UTL_ALE falling edge	-	80	ns
tw4	UTL_CS1/2# Pulse Width	275	-	ns
tw5	UTL_WR# Pulse Width	185	-	ns
tw6	UTL_ALE falling edge to UTL_CS1/2#2,UTL_WR# rising edge	245	-	ns
tw7	UTL_AD7-0 Address Setup Time to UTL_ALE falling edge	30	-	ns
tw8	UTL_AD7-0 Address Hold Time from UTL_ALE falling edge	10	-	ns
tw9	UTL_AD7-0 Data Setup Time to UTL_CS1/2#, UTL_WR# rising edge	185	-	ns
tw10	UTL_AD7-0 Data Hold Time from UTL_CS1/2#, UTL_WR# rising edge	10	-	ns

UTILITY BUS READ CYCLE

Symbol	Parameter	Min.	Max.	Units
trl	UTL_ALE Pulse Width	25	-	ns
tr2	UTL_CS1/2# Output Valid to UTL_ALE falling edge	25	-	ns
tr3	UTL_RD# Output Valid from UTL_ALE falling edge	-	80	ns
tr4	UTL_CS1/2# Pulse Width	275	-	ns
tr5	UTL_RD# Pulse Width	185	-	ns
tr6	UTL_ALE falling edge to UTL_CS1/2#, UTL_RD# rising edge	270	-	ns
tr7	UTL_AD7-0 Address Setup Time to UTL_ALE falling edge	30	-	ns
tr8	UTL_AD7-0 Address Hold Time from UTL_ALE falling edge	10	-	ns
tr9	UTL_AD7-0 Data Setup Time to UTL_CS1/2#, UTL_RD# rising edge	80	-	ns
tr10	UTL_AD7-0 Data Hold Time from UTL_CS1/2#, UTL_RD# rising edge	10	-	ns

SRAM BUS WRITE CYCLE

Symbol	Parameter	Min.	Max.	Units
t1	SR_CS# falling edge to SR_WR# falling edge	0	-	ns
t2	SR_WE# rising edge to SR_CS# rising edge	0	-	ns
t3	SR_A16-0 Setup Time to SR_WE# falling edge	2	-	ns
t4	SR_A16-0 Hold Time from SR_CS# rising edge	0	-	ns
t5	SR_D31-0 Setup Time to SR_CS# rising edge	11	-	ns
t6	SR_D31-0 Setup Time to SR_WR# rising edge	11	-	ns
t7	SR_D31-0 Hold Time from SR_CS# rising edge	0	-	ns
t8	SR_D31-0 Hold Time from SR_WR# rising edge	0	-	ns

SRAM BUS READ CYCLE

Symbol	Parameter	Min.	Max.	Units
t1	SR_CS# falling edge to SR_OE# falling edge	0	-	ns
t2	SR_OE# rising edge to SR_CS# rising edge	0	-	ns
t3	SR_D31-0 Setup Time to SR_)E# rising edge	15	-	ns
t4	SR_D31-0 Setup Time from SR_CS# rising edge	15	-	ns
t5	SR_D31-0 Hold Time to SR_OE# rising edge	10	-	ns
t6	SR_D31-0 Hold Time to SR_SC# rising edge	10	-	ns
t7	SR_CS#0 falling edge to SR_ADR16-0 Valide	0	-	ns
t8	SR_A16-0 to SR_D31-0 Valid	15	-	ns

EPROM

Symbol	Parameter	Min.	Max.	Units
t1	SR_D7-0 Hold Time fromROM_CS# rising edge	0	-	ns
t2	ROM_CS# falling edge to SR_A16-0 Valid	0	-	ns
t3	ROM_CS# rising edge to SR_A16-0 Delay	0	-	ns
t4	ROM_CS# Pulse Width	345	-	ns
t5	SR_A16--0 Change to SR_D7-0 Valid	-	70	ns
t6	SR_A16-0 to SR_A16-0 Change	75	-	ns

EEPROM

Symbol	Parameter	Min.	Max.	Units	Comments
t1	SAR_CLK to Output Signal Valid Delay: EECS, EED0, EECLK	100	-	ns	software controlled
t2	EEDI Input Setup Time to SAR_CLK	10	-	ns	software controlled
t3	EDDI Input Hold Time from SAR_CLK	0	-	ns	software controlled

NICStAR OVERVIEW

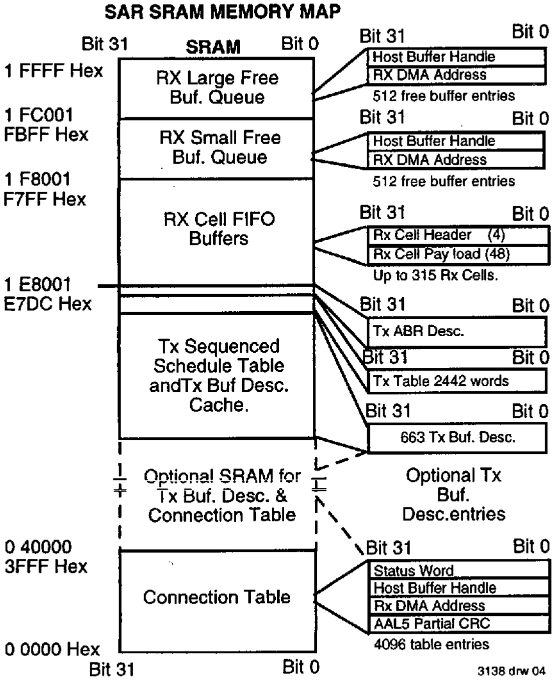
A NIC or internetworking product based on the NICStAR includes:

- IDT77201 NICStAR
- 32K x 32 - 15 ns SRAM (expandable to 128K x 32):
  - Receive Small/Large Free Buffer Queues
  - 315-cell Receive FIFO Buffer
  - Receive Connection Table
  - Transmit Buffer Descriptors
  - Transmit Schedule Table
  - Intermediate AAL5 CS-PDU CRC storage
- 32K x 8 - 100 ns (optional) PROM (expandable to 128Kx 8)
  - Host driver storage (loaded at boot time).
- EEPROM, serial I/O (optional)
  Non-volatile configuration data storage.
- Crystal Clock Oscillators
  - 66.67 MHz for NICStAR clock
  - 25.00 MHz for UTOPIA interface

Local SRAM

A small amount of external SRAM is used by the NICStAR for various key functions, as shown below. As the table at the right illustrates, the size of the local SRAM determines the maximum number of simultaneously open receive and transmit connections; 32K x 32 SRAM should be sufficient for most applications.

SAR SRAM MEMORY MAP



Options for Max. # of Receive VC Connections:

	32K x 32	128K x 32
4K VCs	Yes	Yes
8K VCs	-	Yes
16K VCs	-	Yes

Max. # of Transmit VC Connections:

	32K x 32	128K x 32
CBR VCs*	647	2430*
ABR/VBR/UBR VCs	= Rx VCs	= Rx VCs

*Specifies the # of simultaneously open Tx CBR VCs.
The theoretical maximum # is 2430 with 155.52 Mbps ATM.

PCI Interface

The NICStAR includes a PCI DMA master interface, which requires no glue logic to interface to the host system's PCI bus. This interface provides efficient, low latency transfers to and from the host memory. Further, the DMA master transfer method relieves the host system processor from most of the activities involved in ATM communication. The device driver only needs to write and maintain small descriptors in the host memory and to update pointers in local SRAM for the NICStAR. All ATM cell payload transfers, as well as all key descriptor transfers, are controlled by the NICStAR.

To achieve optimum performance, other devices and interface cards in the host system which have PCI bus master capability should have their Latency Timers set to values < 30 (representing the number of PCI clocks a bus master may use for transfer purposes). This should allow a NICStAR-based device to obtain access to the PCI bus in ~ 1 us, low enough that isochoronous data will not be affected in 155 Mbps ATM networks.

PHY Interface

For connecting to PHY components, the NICStAR provides a UTOPIA (Universal Test and Operations PHY Interface for ATM) interface. UTOPIA is a standard data path handshake protocol which eases PHY and other product integration and interchange.

SAR Function Implementation

The NICStAR implements the Segmentation and Reassembly (SAR) function as described in the ATM User-Network Interface Specification, Version 3.1, and other documents published by the "ATM Forum".

Host Driver Operation

The NICStAR operates under the control of a software device driver running on a host system. In receive, the device driver generates lists of host memory buffer addresses which constitute reassembled CS=PDUs in host memory buffers. Once reassembly is complete, a list of addresses is provided to the application program(s) for conversion of the CS-PDU back to user data.

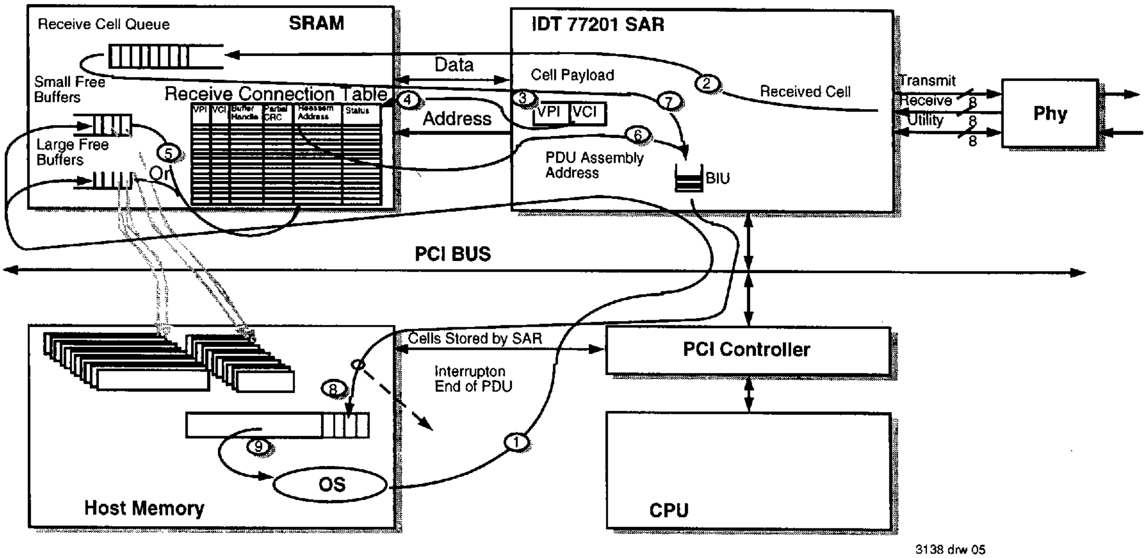
When transmitting, CS-PDUs are queued in host memory as they become ready. The device driver creates descriptors of the host memory buffer addresses which contain the PDU, and then writes these descriptors into a descriptor queue (located in host memory), for processing by the NICStAR. The device driver initiates the transmit process by incrementing a pointer to the descriptor queue (located in local SRAM).

NICStAR Receive Operation

The NICStAR may simultaneously receive AAL5, AAL3/4, OAM, "AAL0" and "Raw Cell" formats. This section provides a description of the overall receive operation, followed by an overview of how each AAL format is supported.

Following the above diagram by the numbers:

1. Before reassembly may begin, the device driver must provide the NICStAR with a supply of host memory locations (buffers) which may be used for reassembly of ATM cell payloads into CS-PDUs. The start address of each buffer allocated for reassembly, called Small Free Buffers and Large Free Buffers, must be programmed into the local SRAM's Small Free Buffer Queue and Large Free Buffer Queue, respectively. The size of both types is programmed at initialization; Small Free Buffers default to 64 bytes (carriage returns, message receipt acknowledgements, etc), while Large Free Buffers default to 2 Kbytes. The NICStAR accomodates up to 512 Small and 512 Large Free Buffers at any one time.
2. A 53-byte ATM cell received from the PHY is immediately written by the NICStAR into the local SRAM's Receive Cell Queue (315 cell FIFO). The NICStAR writes the ATM cell header without the HEC byte, since the HEC byte was calculated and compared within the PHY prior to being received by the NICStAR.
3. The ATM cell header is read by the NICStAR.
4. The NICStAR uses the VPI/VCI field of the ATM cell header to index into the Receive Connection Table, which contains the following information:
   - VPI/VCI (unique for each virtual connection)
   - Buffer Handle (virtual start address of a free buffer)
   - Partial CRC value (for AAL5 PDU)
   - Reassembly Address (from Free Buffer Queues)
   - Status (AAL format, etc.)
5. Assuming this is the first ATM cell received for this CS-PDU, the first free buffer address in the Small Free Buffer Queue is copied into the Receive Connection Table entry for the specified virtual channel (VC). As additional cells are received for this CS-PDU, cell payloads are deposited into host memory at remaining addresses pointed to by this Small Free Buffer. Once the Small Free Buffer memory area is exhausted, subsequent free buffers (as needed) are copied from the Large Free Buffer Queue to finish reassembly of the PDU. The first ATM cell payload of a new CS-PDU is always stored into a memory location addressed by a Small Free Buffer.
6. The NICStAR writes the start address for the Small Free Buffer to it's Bus Interface Unit (BIU).
7. The NICStAR writes the 12 word ATM cell payload to it's BIU.
8. The NICStAR performs a PCI DMA-master transfer of the 48-byte ATM cell payload to the specified Small Free Buffer in host memory. After completely filling any Small or Large Free Buffer in host memory, the NICStAR writes the start address of the buffer to the Receive Status Queue, located in host memory. As additional Large Free Buffers are filled with ATM cell payloads, the NICStAR writes the start addresses of the Large Free Buffers to the Receive Status Queue for the specified VC. After the NICStAR detects an end of PDU, it may



IDT 77201 SAR Controller Receive Data Flow (optionally) generate an interrupt, informing the host system to service the Receive Status Queue.

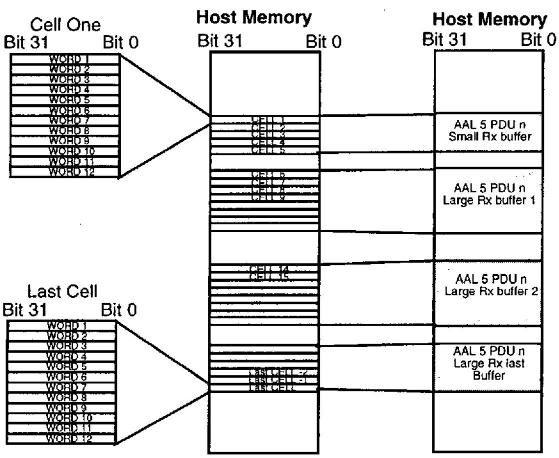
9. After an "end of PDU" is detected, the device driver reads the Receive Status Queue, generates a list of host memory buffer addresses which constitute the received CS-PDU and then provides the list of addresses to the application program(s) for converting back to user data.

• ATM Adaptation Layer (AAL) Support

As a VC connection is being established, the NICStAR assigns it a specific AAL format identifier, which is maintained in the local SRAM's Receive Connection Table. The following are descriptions of how each AAL format is supported:

• AAL5

AAL5 cells are reassembled by the NICStAR and stored directly to the appropriate host memory buffers. As each AAL5 cell contains a 48 byte payload (with the possible exception of the last cell), the cell payload is mapped directly



3138 drw 06

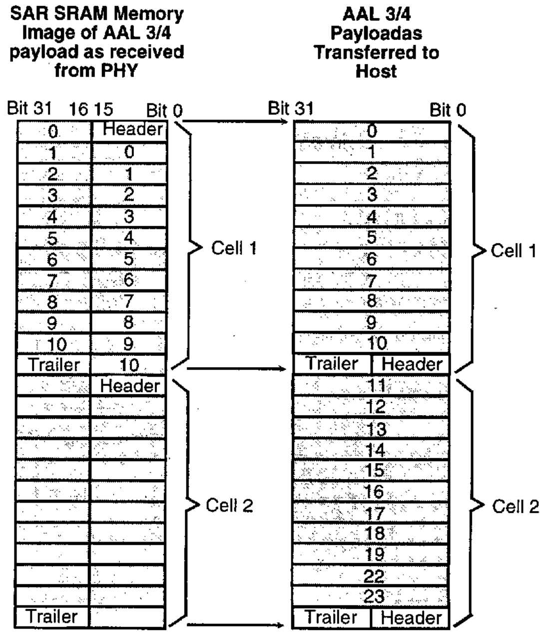
into 12 32-bit words and transferred as shown below.

The above diagram illustrates a Small Free Buffer for storing the first ATM cell payload, followed by successive Large Free Buffers. The NICStAR accumulates a CRC-32 value for all AAL5 cells from a VC, and stores the running total in the Receive Connection Table. When the last AAL5 cell is received from a specific VC, the NICStAR compares it's final calculated CRC-32 value to the CRC-32 value contained within the last AAL5 cell's payload.

• AAL3/4

As the first byte (header) and the last two bytes (trailer) of an AAL3/4 payload contain overhead information, AAL3/4 cells receive special processing.

As illustrated in drawing 5, the NICStAR shifts the header to payload byte positions 47 and 48, and leaves the AAL3/4 trailer in it's original location (payload bytes 45 and 46). In addition, payload data is all shifted to an even word boundary. Transferring the cell payload in this format to the host system



3138 drw 07

supports subsequent data processing effiiciency. On receiving the cell payload, the device driver merely decodes the AAL3/4 header and trailer, followed by a simple word-aligned reassembly into a complete CS-PDU. The NICStAR calculates a payload CRC-10 value and stores it in the trailer. If the NICStAR detects a CRC error, it will set an error bit in the Receive Status Queue for the host memory buffers associated with this CS-PDU.

• OAM Cells

Operations and Management (OAM) cells are identified by several reserved (ATM Forum specification) VPI/VCI addresses, as well as several of the possible states contained in the Payload Type Identifier (PTI) field of the cell header. Since the header of OAM cells contains useful information, the entire cell is transferred to host memory; specifically stored in the Raw Cell Queue (see Raw Cell below). There are three possible OAM cell states:

1. Currently established VPI/VCI connections which may be passing application data; these connections may also pass OAM cells (ie, without application data) by setting certain PTI bits in the cell header. OAM cells of this type are detected by the NICStAR and transferred to the Raw Cell Queue in host memory. The NICStAR may optionally generate an interrupt upon completion of the transfer.

2. 'Special' VPI/VCI connections which may be assigned for OAM cell communication. These are assembled according to their AAL format (created on establishment of connection). Operation continues as 'normal'; the device driver is interrupted as each CS-PDU is reassembled.

3. 'Unidentified' VPI/VCI combinations are those ATM cells which are received, but which do not have a corresponding

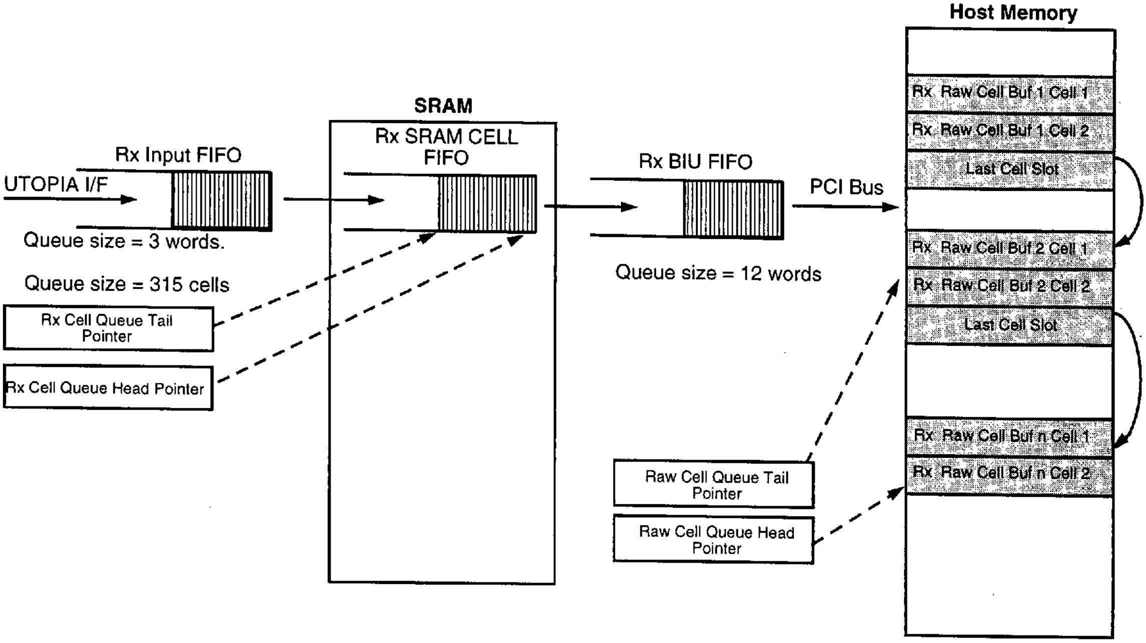
entry in the Receive Connection Table. These cells are passed on to the "Raw Cell Queue" (described in the AAL0 section below) for identification processing.

• "AAL0"

"AAL0" cells are ATM cells which conform to the 5 byte header, 48 byte payload structure of "general" ATM cells, but which do not fit within the requirements of other AAL formats. These "AAL0" cells are treated identical to AAL5 format cells, but without CRC processing and checking.

Using "AAL0", the NICStAR provides a means to support future AAL definitions. The device driver, on receipt of an AAL0 CS-PDU could perform additional payload (or PDU) processing as required by the newly defined AAL.

• "Raw Cells"



3138 drw 08

"Raw Cells" are defined as follows:

1. Identified as "Raw Cell" in the Receive Connection Table, by a particular VC.
2. Unknown VPI/VCI (entry not found in Receive Connection Table). This is selectable via the host driver: "Unknown" traffic may either be discarded, or placed in a Raw Cell Queue.
3. OAM cells (defined either by specific VC or PTI bits).

The diagram below illustrates the path flow of an incoming "Raw Cell" arriving via the UTOPIA interface, and its deposition into a Raw Cell Queue.

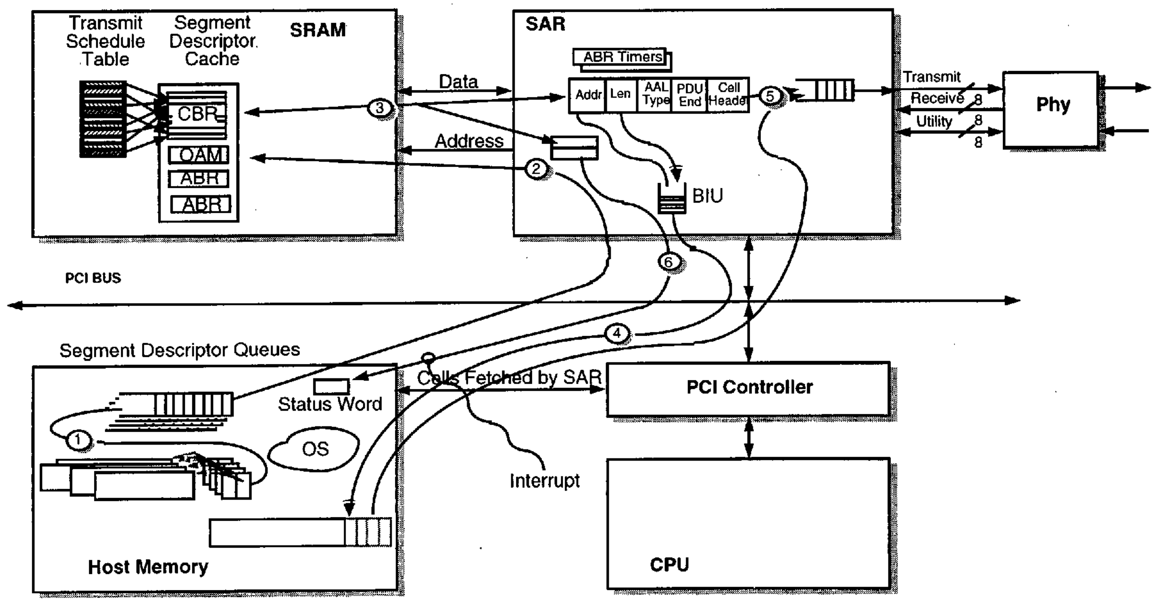
Note that Raw Cells are transferred in their entirety (payload and header) to the Raw Cell Buffer Queue for processing within the host.NICStAR Transmit Operation.

As CS-PDUs are available, the NICStAR continuously segments and transmit ATM cells at the full 155 Mbps "wire speed". It simultaneously accomodates Constant Bit Rate (CBR), Unassigned Bit Rate (UBR), Available Bit Rate (ABR), and Variable Bit Rate (VBR) traffic types. Depending on the amount of external SRAM, the NICStAR supports up to 16K open CBR connections; independent of the size of the SRAM, it always supports the maximum of 16,000,000 VC connections (the full 24 bit VPI/VCI address space).

This section describes the overall transmission portion of the NICStAR. Following sections describe the Transmit Buffer Descriptors (TBDs) and the Transmit Cell Schedule Table (TCST), which manages the overall channel bandwidth and provides CBR connections with "guaranteed" bandwidth allocation.

Following the above diagram by the numbers:

1. As a CS-PDU becomes available for transmit, the device driver creates Transmit Buffer Descriptors (TBDs) for the sequence of buffers in host memory which constitute the CS-PDU, and then writes the TBDs into a TBD queue, located in host memory.
2. The device driver then causes the NICStAR to copy the first one or two TBDs to local SRAM.
3. The NICStAR reads the first TBD. The ATM cell header, also part of this buffer descriptor, is loaded into the output FIFO. During this process, a HEC byte place holder (00h) is added as the fifth byte of the header.
4. The PCI bus is arbitrated using the address and length taken from the TBD.
5. The ATM cell payload is transferred from host memory to the output FIFO via DMA. On completion, the 53-byte ATM cell is transferred out of the NICStAR via the UTOPIA interface.



3138 drw 09

IDT 77201 SAR Controller Transmission Data Flow

6. Status information is returned to the host system to communicate transmission state, error conditions, etc.

• Transmit Buffer Descriptors

A Transmit Buffer Descriptor (TBD) is a four word descriptor which contains information such as the base address of a buffer in host memory, the number of words in the buffer, the AAL format of the information in the buffer (used when segmenting the buffer into ATM cells) and the ATM cell header (all TBDs in the same queue have identical cell headers; that of the first ATM cell of the CS-PDU).

The device driver writes the TBDs into a TBD Queue in host memory, and then increments a pointer to the queue in local SRAM, which causes the NICStAR to copy the first one or two TBDs to local SRAM. The NICStAR then reads the TBD and begins it's transmits process. The information contained in a TBD is dependent upon which traffic type is stored in the corresponding Tx buffer:

CBR Traffic:

- Control Information (e.g. interrupt at end, etc)
- Cell Header
- Buffer Size, Base FIFO Address

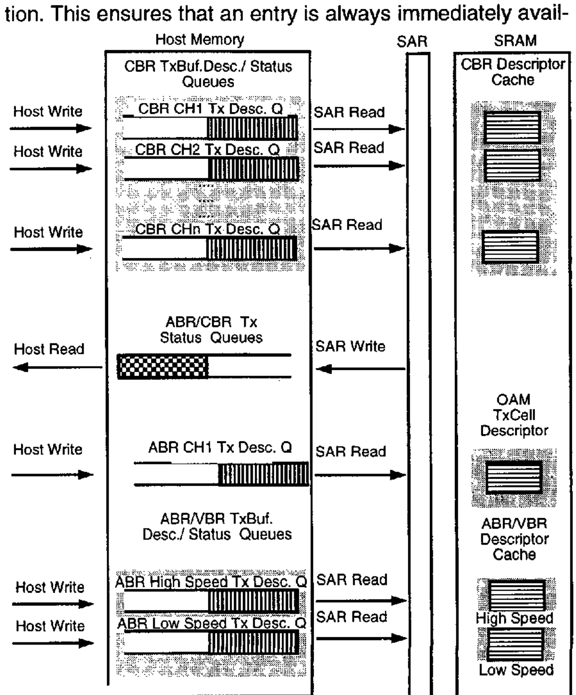
UBR/ABR/VBR Traffic:

- Timer mantissa and exponent
- Interrupt at EOB
- Buffer Address, Size
- Status
- Segment Length
- Cell Header

The NICStAR maintains 3 types of transmit descriptor caches (queues):

1. CBR

This cache holds two entries from each open CBR connection. This ensures that an entry is always immediately avail-



3138 drw 10

able for each connection, under schedule control of the NICStAR's Transmit Cell Schedule Table.

2. OAM

This cache is reserved for OAM cells which are considered higher priority than UBR/VBR traffic, but are to be sent only during time slots not reserved for CBR connections.

3. UBR/ABR/VBR

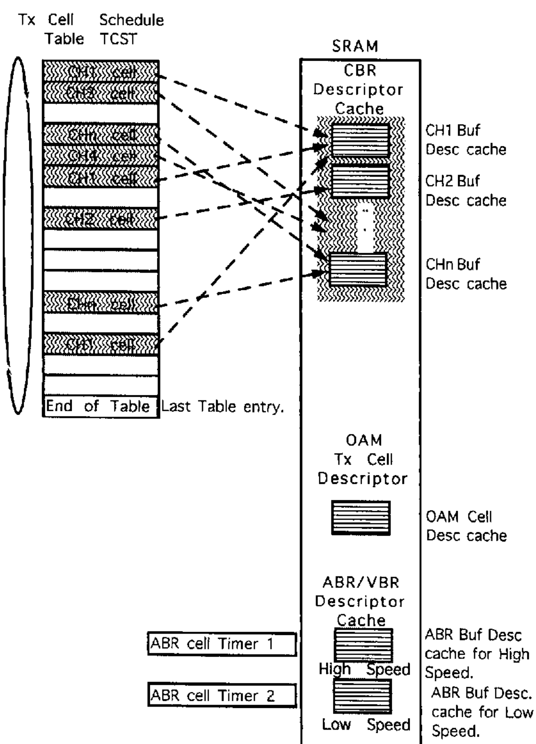
This cache consists of two sections a "high speed" cache and "low speed" cache. This separation provides a 'passing lane' for higher-speed/higher-priority traffic. Descriptors in the "Low Speed" queue are serviced only after the "High Speed" queue is empty, ensuring that higher-speed traffic is shipped at the highest data rate possible without exceeding its negotiated bandwidth. The facility operates under software control such that it can be tailored for specific applications and/or current operating conditions.

• Transmit Schedule Table (TST)

The Transmit Schedule Table is used to guarantee CBR transmission at fixed data rates and specific timing intervals within the system bandwidth. The TST is a circular table, in local SRAM, which the NICStAR continually scans to allocate bandwidth and control which connection is serviced. The number of entries in the table is equivalent to the line speed divided by the desired bandwidth resolution.

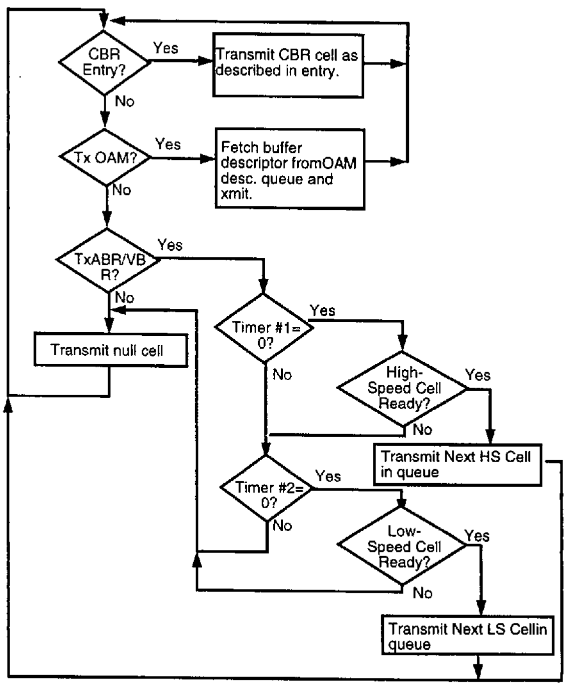
As an example, a 155Mb/s line would support 2430 64Kb/ CBR conneactions. Since the TST is scanned many times each second, any CBR channel may be allocated bandwidth in multiples of 64Kb/s. Each 64Kb/s entry 'contains' one line-speed cell time, which at 155Mb/s equals 2.7.µs. I It contains It contains three entry types:

1. CBR
2. OAM
3. ABR
4. VBR



3138 drw 11

TCST Entry Control Flow Chart



3138 drw 12

CBR entries are VC-specific: it tells the SAR exactly which connection is to be serviced at that time. All other entry types designate available opportunities to transmit these data types.

Each TCST entry is either CBR, OAM, or ABR/VBR. If the entry is not defined, or cells are not available for transmission, a null cell is generated and transmitted. This feature is provided to assist users in integrating the 77201 SAR with PHY transceivers which may not have automatic null cell generation.

Each ABR/VBR entry has associated with it, a timer value which is used to throttle its transmission speed based upon the bandwidth allocated to it when the connection was established. Thus, if the TCST is servicing an ABR/VBR entry, the entry can point to one of two possible states:

1. A new buffer descriptor. In this case, the 'timer' is set to zero, since this connection has not been serviced yet. Once a cell has been transmitted, the timer is set for countdown.
2. A buffer descriptor whose transmission is 'in progress'. Data remains in the buffer. If the bandwidth-timer has timed out, a cell from this buffer is transmitted. Otherwise, flow control is transferred to check the "Low Speed" timer (Timer #2), which operates in the same way for entries in the "Low Speed" buffer descriptor cache.

8

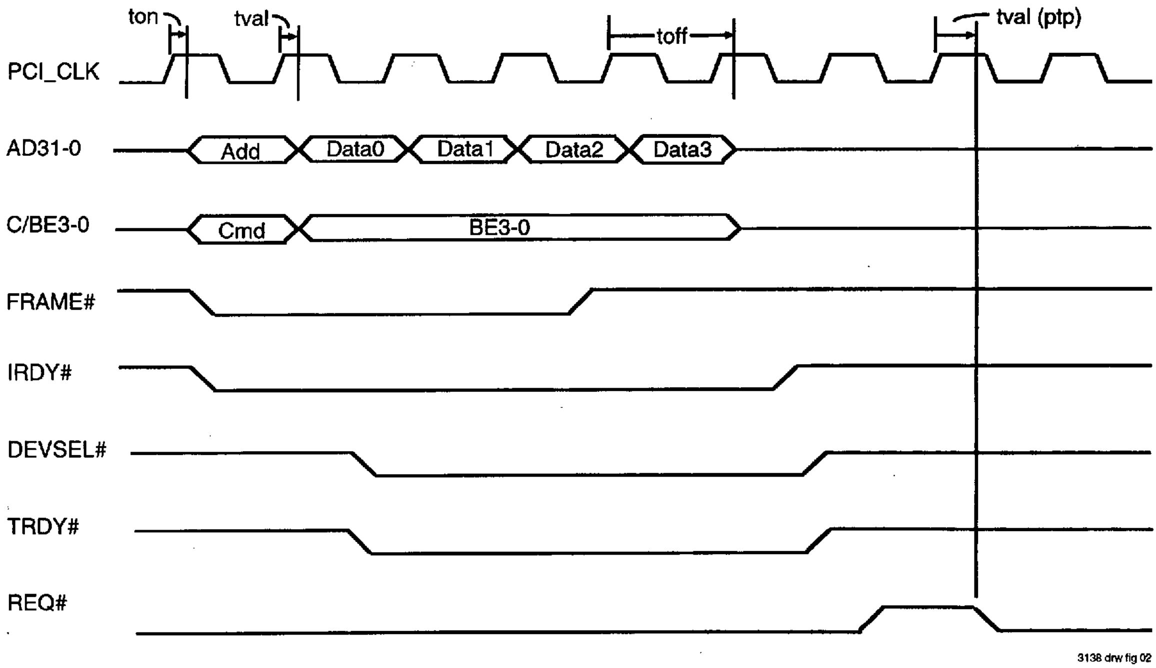


Figure 1. The NICStAR as a PCI master (illustrates a 4-word write by the NICStAR to host memory)

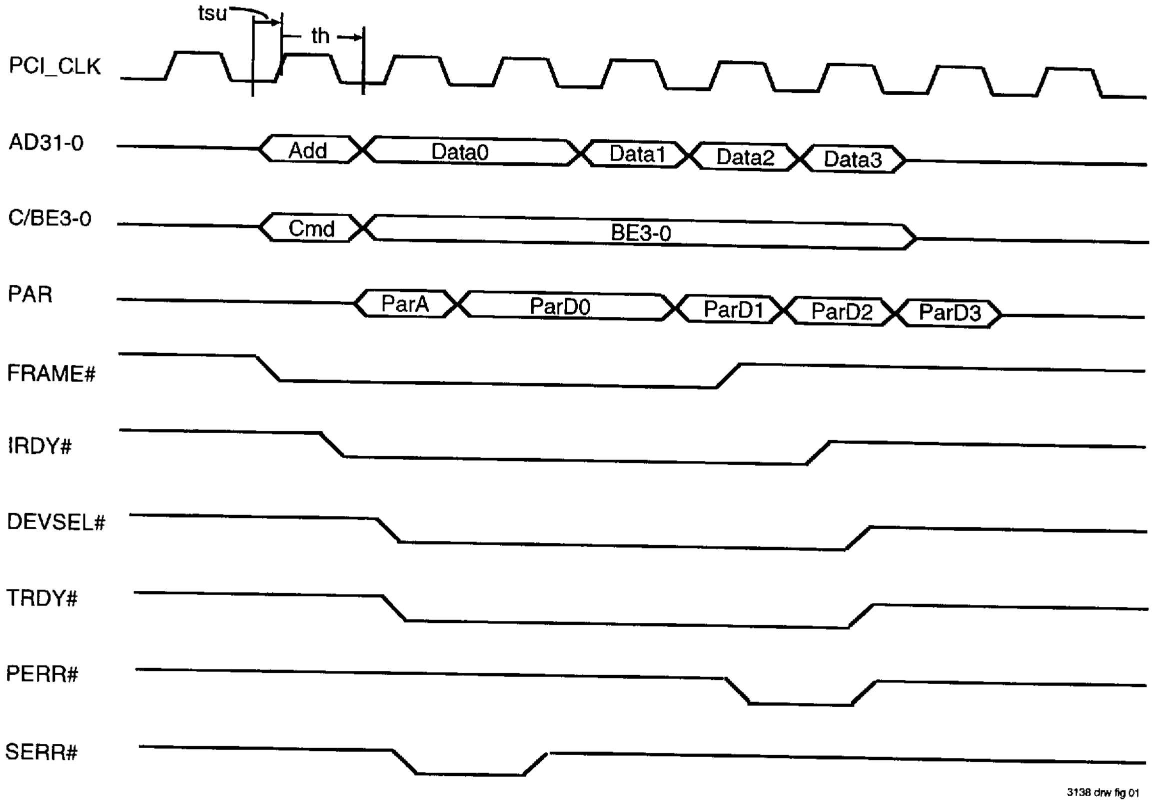


Figure 2. The NICStAR as a PCI target (illustrates a 4-word write operation by the host device driver to the NICStAR)

8

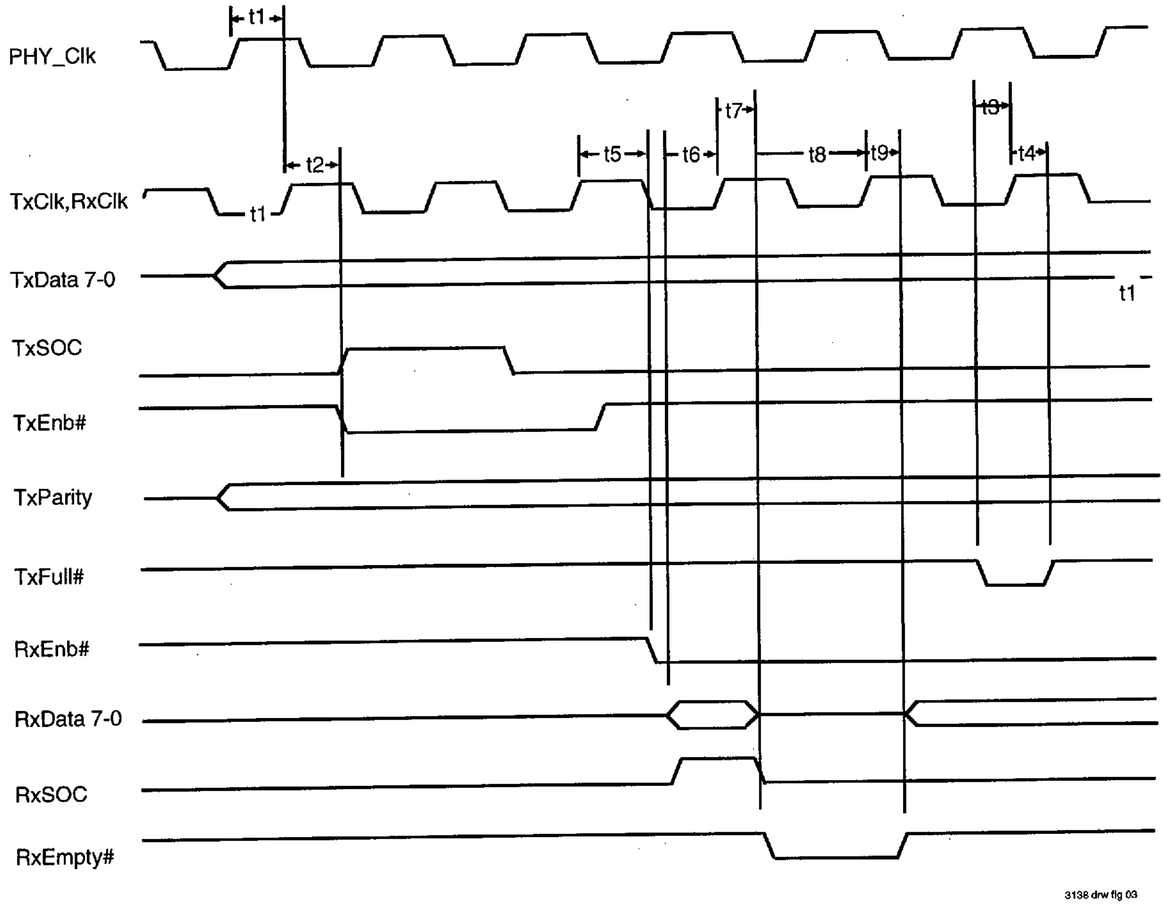


Figure 3. UTOPIA Bus Timing

4825771 0026691 853

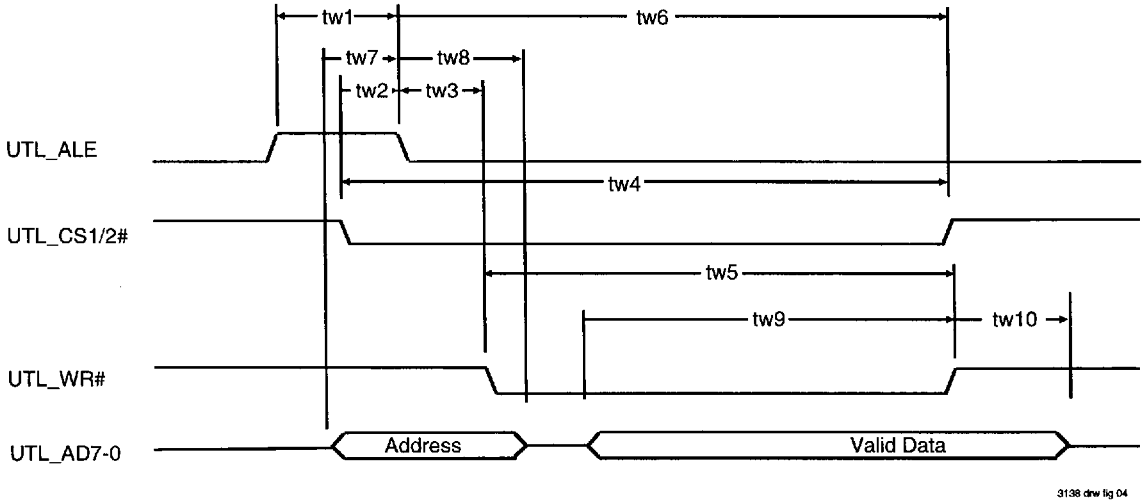


Figure 4. Utility Bus Write Cycle

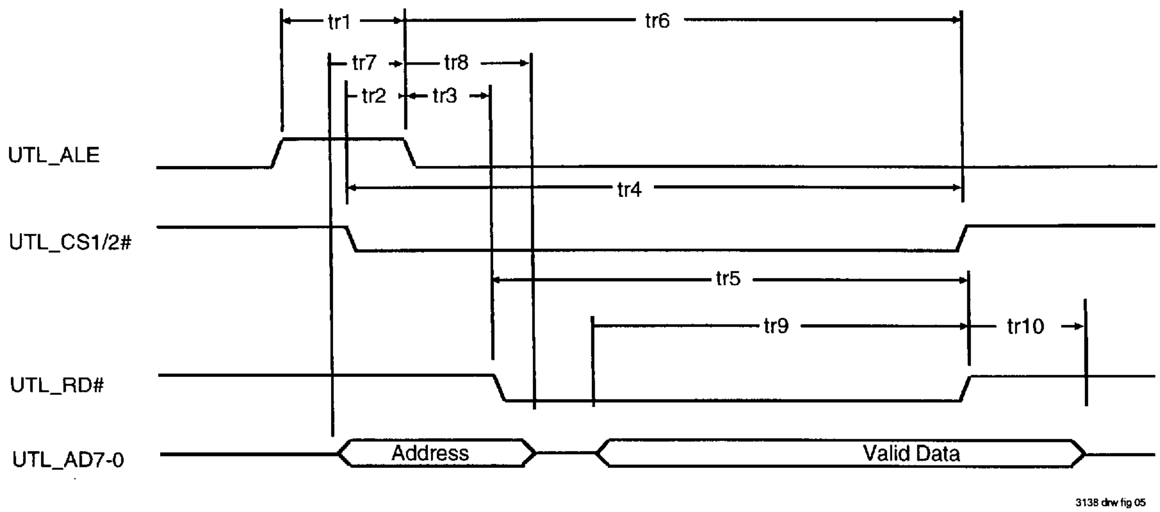


Figure 5. Utility Bus Read Cycle

8

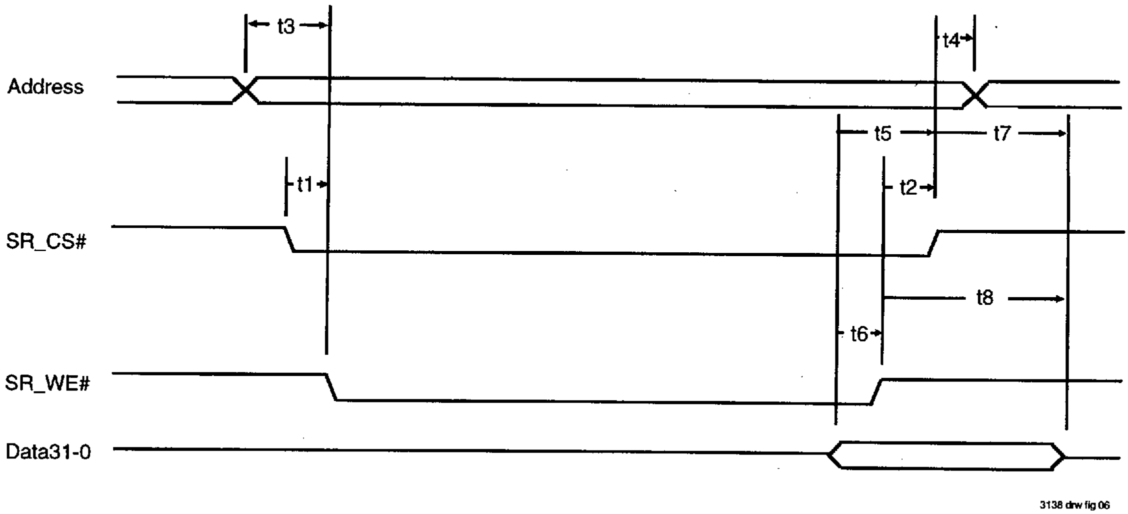


Figure 6. SRAM Bus Write Cycle Timing

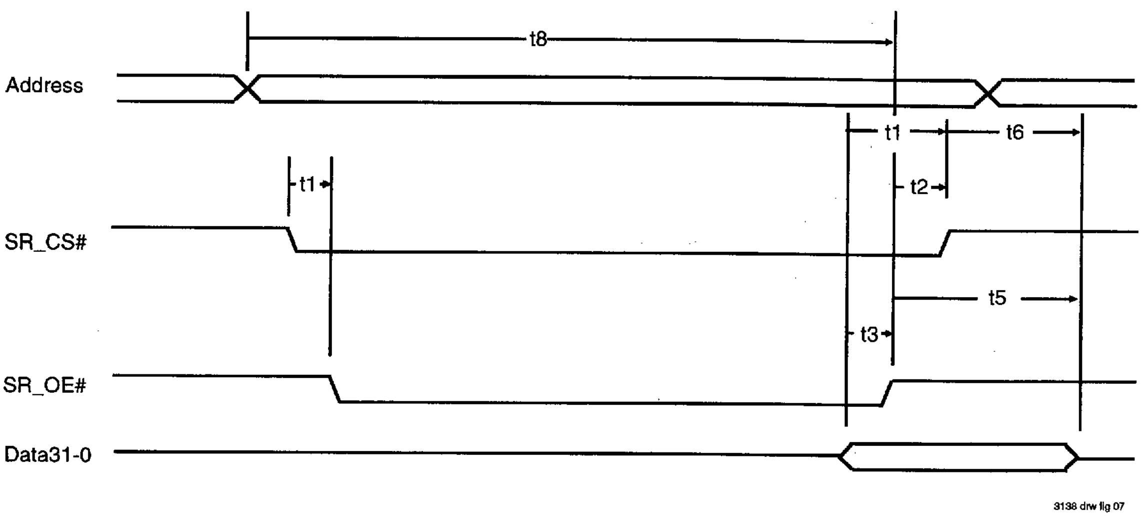


Figure 7. SRAM Bus Read Cycle Timing

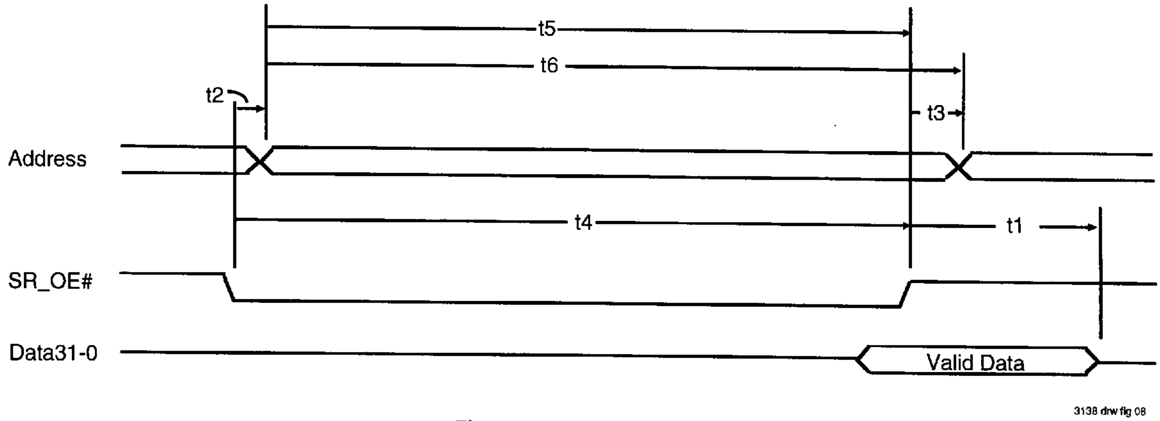


Figure 8. EPROM Timing

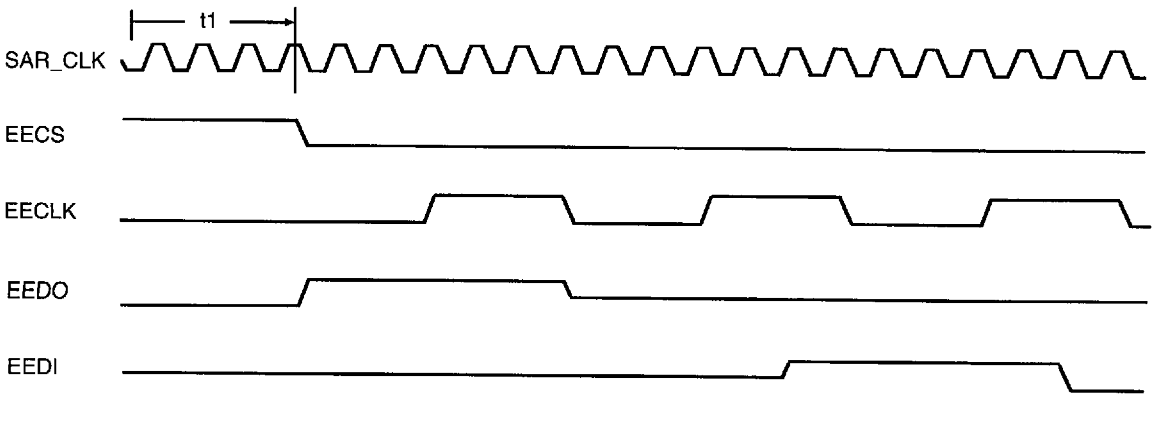


Figure 9. EEPROM Timing

SOFTWARE AND SOFTWARE DRIVERS

Evaluation Software

IDT offers a share-ware program called "SARWIN" that runs on Windows 3.1™. This is a window dirven program that allows access to all registers and memory in the SAR, PHY, NIC memory, and host memory. Registers and memory are both visible and accessible. Data can be formated, cells built, and data sent and received via this program. Please contact IDT or look on IDT's latest CD ROM for this software.

Software Drivers

IDT has contracted with several contractors who have writen software drivers for the IDT 77201. Please contact those listed below for more information:

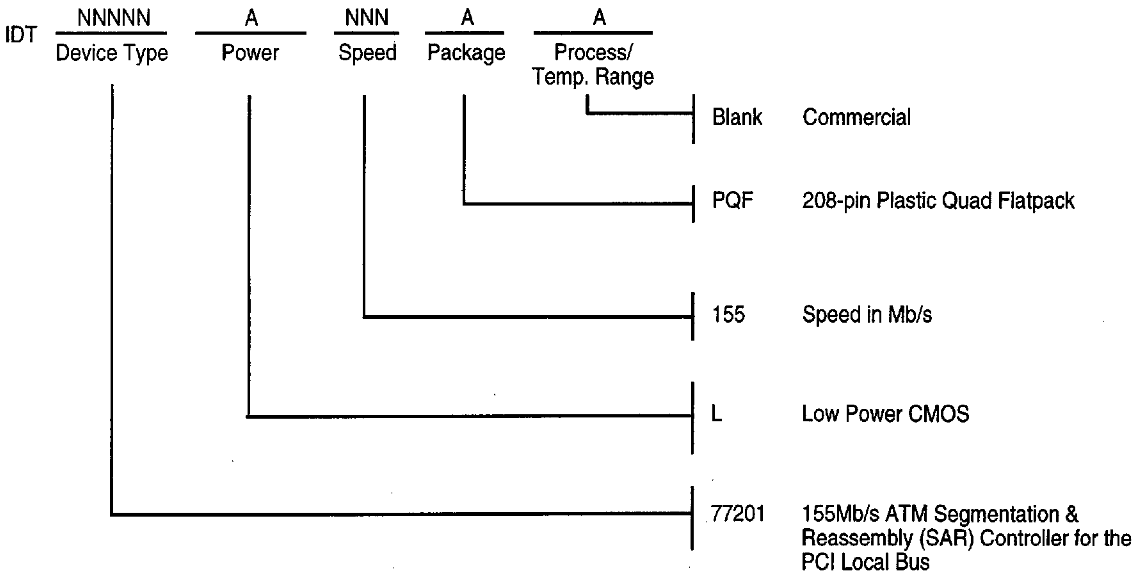
Windows NT™ Drivers and other Windows Drivers

Telogy Networks
William Simmelink, Vice President, Sales
21 Firstfield Road
Gaithersburg, Maryland 20878
(301) 527-2717 (phone) (301) 417-0324 (FAX)
wsimmelink@telogy.com

Novel Netware™ Drivers

Harris & Jefferies
Robert Zockoff, Vice President, Sales
888 Washington Street
Dedham, Massachusetts, 02026-6031
(617) 329-3200 (phone) (617) 329-4148 (FAX)
bobz@hjinc,com

ORDERING INFORMATION



3138 drw 13

PRELIMINARY DATASHEET: DEFINITION

"PRELIMINARY" datasheets contain descriptions for products that are in early release, including features and block diagrams.

Datasheet Document History

8/11/94: Initial Public Release
9/28/94: Pinout and Pin Definitions updated.
12/8/94: Pinout revised to final.
12/21/94: Pin 133 changed from EECS* to EECS with input polarity selectable via command register.
4/3/95: Rise/Fall times for external oscillator relaxed from 2ns to 4ns.
4/47/95: Pin 198 redefined from Vcc to SAR_CLKint, which is an output pin.
6/15/95: SAR Clock frequency reduced from 66MHz to 50MHz.
10/12/95: The 25.0MHz oscillator was removed from the block diagram, page 1.
1/9/96: Modified pin definitions.