VITESSE

VSC7384
Data Sheet

Stansted™ – 12-Port Layer 2 Gigabit Ethernet Switch

Features

- 12 Gigabit Ethernet ports with nonblocking wire-speed performance
- Tri-speed (10/100/1000 Mbps) RGMII interfaces
- Support for both wire-speed automatic learning, and CPU-based learning
- 208 kB on-chip frame buffer
- Jumbo frame support
- Programmable classifier for QoS (Layer 4/Multimedia) into four classes of service
- 8,192 MAC addresses and 4,096 VLAN support (IEEE802.1Q)
- Per-port shaping, policing, and Broadcast and Multicast Storm Control
- IEEE802.1Q-in-Q nested VLAN support
- Full duplex flow control (IEEE802.3x) and half duplex back pressure
- Flexible link aggregation compliant with IEEE802.3ad
- Spanning Tree Protocol support (IEEE802.1D)
- Multiple Spanning Tree support (IEEE802.1s)
- Port-based Access Control (IEEE802.1X)
- IGMP, GARP, GMRP, and GVRP support
- Cost effective 4-pin serial CPU interface
- Selection between on-chip 8051 CPU, or off-chip 8-bit or 16-bit CPU for SNMP and Web-based management

General Description

The Stansted is a full-featured, 12-port, Gigabit Ethernet switch-on-a-chip featuring several integrated management interfaces and support for both copper and optical PHYs using RGMII interfaces. The device provides nonblocking, wire-speed gigabit performance on all its ports. Web-based and SNMP management is made possible with the optionally enabled on-chip 8051 CPU, or through its performance-optimized 8-bit or 16-bit parallel interface.

In addition, less processor-intensive, managed operation is possible using a simple 4-wire serial interface.

Stansted has been optimized for SMB and desktop market segments, and does not require additional external memory.

Each device port is equipped with a Policer for ingress traffic control and a Shaper for egress traffic rate management.

The Stansted device also supports programmable higher layer classification and prioritization to enable enhanced Quality of Service (QoS) support for real time applications such as Voice over IP (VoIP).

Contents

Figures

Tables

Functional Overview

The VSC7384 Stansted device can operate as either a VLAN aware switch or a VLAN unaware switch. It forwards frames at Layer 2 based on information up to and including Layer 4.

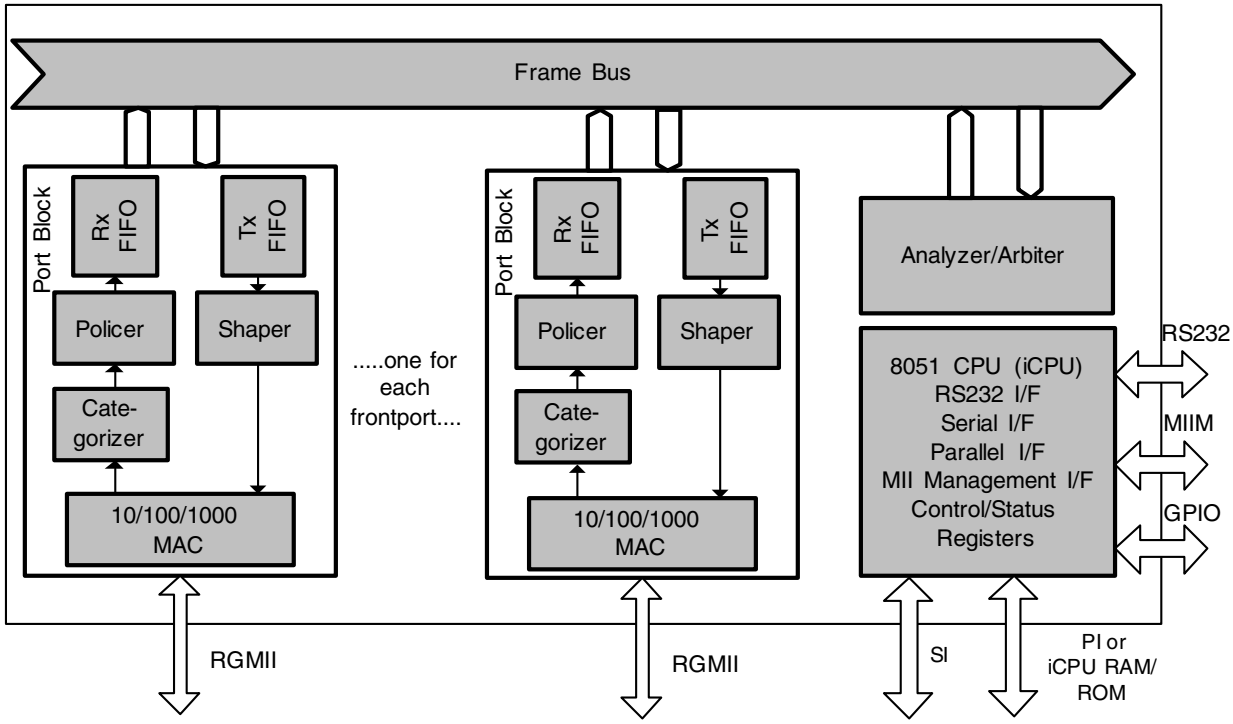


Figure 1. Block Diagram

Tri-Speed Gigabit MACs

The Stansted device implements 12 separate tri-speed Ethernet MACs designed to comply with IEEE 802.3-2002. The tri-speed MACs support full duplex operation at 10/100/1000 Mbps and half duplex operation at 10/100 Mbps.

Register Access

The control block receives commands from either the iCPU or from an off-chip controller/processor. The commands can be divided into four categories:

- MII Management reads and writes for PHY control
- Configuration
- Status or statistics gathering
- Capture and transmission of packets

Link Aggregation

Stansted supports ingress and egress port aggregation. Any number of ports can be aggregated into any number of groups with a frame distribution function based on the Source MAC address, the Destination MAC address, or a combination of both.

VLAN Support

Stansted can be configured as either VLAN unaware (behaving transparently against VLAN tagged frames) or as VLAN aware.

When VLAN is enabled, frames received without a VLAN tag on a given port are tagged with a port specific, configurable VLAN number. Frames that already have a VLAN tag when they are received have no additional tag added.

VLAN awareness (and thus tagging/untagging of frames) can be configured on a per-port basis, which is a very powerful feature for advanced applications.

Stansted can set up and maintain 4096 VLANs.

Furthermore, each port can be configured to a set of ports it can forward to, thereby facilitating port-based VLANs. By default, all ports can forward to all others.

802.1 Q-in-Q Support

Q-in-Q is an efficient method for enabling Subscriber Aggregation. This is very useful in, but is not limited to, Metropolitan Area Network (MAN) applications. Q-in-Q is, in effect, the use of double VLAN tags.

In Stansted, Q-in-Q is supported in three ways:

1. Each port can add or strip a VLAN tag (independently configurable for ingress and egress per port).
2. Frame priority can be determined based on information in the outer VLAN tag.
3. The frame can be forwarded based on information in the outer VLAN tag.

If several Stansteds are used to aggregate a number of subscribers towards the edge of the network, the Q-in-Q support can be used to store prioritization information and the source port number of each frame for use by subsequent Stansteds. In this application, prioritization need only be done once, because the network edge router easily stores the route back to the subscriber from where the frame originated.

The effective maximum frames size will be 1526 bytes throughout the Q-in-Q network, as the standard end-station MTU of 1518 bytes gets appended with two VLAN tags.

Quality of Service

The Stansted device supports four priority levels. On each port, an enhanced classifier/categorizer assigns priorities based on information taken from Layer 2 to Layer 4.

The Categorizer analyzes all received frames. It assigns each frame to one of four priority levels based on:

1. Priority in the IEEE802.1Q (VLAN) tag
2. Differentiated Services Code Point (DSCP) from the IP-header
3. TCP/UDP port
4. DSAP Value
5. EtherType field
6. Priority in a Q-in-Q tag

Based on the priority determined by the Categorizer, higher priority traffic takes precedence over lower priority traffic during forwarding through the switching engine. When congestion occurs, the traffic with the lowest priority is dropped before traffic with higher priority. Latency is minimized for expedited data because the two highest priority levels can overtake the two lowest priority levels.

Flow Control

All ports can be independently configured to use nondropping flow control and to enable the use of Asymmetric Flow Control. The programmer can set up individual high and low thresholds for each FIFO. These thresholds control the start and stop of pause signaling. The internal FIFOs have enough memory to handle flow control on short-haul, full duplex lines without using excessive pause signaling.

The switch generates flow control when necessary to ensure frames are never dropped, or it can be programmed to discard low priority traffic first in case of congestion.

Packet Forwarding Analysis

The advanced filtering and forwarding capabilities of the Stansted device are a result of the work carried out by the Analyzer.

The Analyzer maintains and uses three tables for packet forwarding: a source port table, a VLAN table, and a destination address table. When frame header information is extracted from an incoming packet, the Analyzer uses these tables to look up the following information:

- A 12-bit Source Port Mask (one bit per port)
- A 12-bit VLAN Mask
- A 12-bit Destination Port Mask (if the DMAC address is known, otherwise a programmable mask)

These three masks are combined by a bit-by-bit AND operation, and the results are adjusted to reflect any active link aggregations. The resulting bit mask is forwarded to the Arbiter as the next forwarding decision for the given source port.

By default, the three masks are set up so that:

- Packets received on a given port cannot be forwarded to that same port.
- There is no VLAN present (all ports are members of all VLANs, frame tags are not examined).
- Packets to a given destination address are forwarded only to the port where the destination was learned.

MAC Address Learning

When a packet is received, the source MAC address is looked up in the destination address table (see "Packet Forwarding Analysis" on page 21). If it is not presently registered and it is not a multicast address, a new entry is created. If necessary, an entry is discarded to make room for the new one based on a "least recently used" algorithm.

The Stansted device is capable of looking up or adding all incoming entries to the MAC table at maximum load. This process is referred to as "wire-speed learning".

Addresses can also be locked using a CPU external to Stansted or the iCPU by writing into the table. Locked entries are never discarded.

Shaping and Policing

The Stansted device provides per-port policing as well as per-port shaping. Policing can be used to limit incoming subscriber streams according to a traffic contract, and shaping can be used to protect a receiving device against high bandwidth bursts.

Based on a leaky bucket algorithm, the Shaper as well as the Policer support a bit rate granularity of 244 kbps for 1 G, 48.8 kbps for 100 M, and 4.88 kbps for 10 M. Spanning Tree and IGMP control frames are only forwarded to the CPU capture facility and will not be policed from the stream.

The ingress Policer can initiate flow control messages, enabling dropless policing.

An additional leaky-bucket system handles the Stansted's Broadcast and Multicast Storm Control.

CPU Interfaces

The Stansted device provides three interfaces to core registers of which two are mutually exclusive. The three interfaces are:

- An 8- or 16-bit Parallel Interface (PI)
- On-chip 8051 CPU Interface
- A serial, SPI-style, 4-wire interface (SI)

The serial interface exists in all configurations of the Stansted. It allows interfacing to a vast amount of microcontrollers and is capable of transferring approximately 2.5 MBps.

The Parallel Interface and the on-chip 8051 CPU (iCPU) Interface are mutually exclusive as they use the same external interface. The selection between the two is made with a strapping pin named ICPU_PI_En. Strapping the pin low enables the Parallel Interface, whereas strapping the pin high enables the iCPU.

With the ICPU_PI_En pin strapped low, the Stansted operates as a slave on an external CPU parallel bus. The PI connects to many well-known processors, like the PowerPC® and ARM® CPUs when operated in 16-bit mode. In addition, the PI interface can be configured to run in 8-bit mode, allowing it to work with an 8-bit CPU system such as an external 8051 controller. The PI can handle about 25 MBps.

With the ICPU_PI_En pin strapped high, the iCPU is enabled, and the aforementioned parallel interface is used as a RAM/Flash interface for programs running within the iCPU. The iCPU is a standard 8-bit 8051 microcontroller with a number of add-on modules extending the basic 8051 functionality. It is designed to replace an external CPU by providing monitoring and configuration functions and the transmission of packets for STP and GxRP (GARP, GVRP, and so forth) protocols, in addition to direct access to the output queues.

Besides standard 8051 functionality, the iCPU features 8 kilobytes of on-chip RAM accessible as both program and data memory and 256 bytes of scratchpad RAM (internal RAM). Both RAMs can be accessed from an external CPU connected to the SI interface, enabling external debugging.

The glueless connection to most commercial Flashes and RAMs is made through the shared 16-bit address and 8-bit data buses with the option of extending the address bus with four bits through a paging mechanism, allowing for up to 1 Mbyte of external data RAM and up to 1 megabyte (Mb) of external program memory (Flash or ROM). The external interface's read and write timing can be configured separately for read and write accesses and for code and data accesses.

The iCPU uses only 4 clock cycles per instruction compared to the original 8051 CPU, which uses 12 clock cycles. In addition, the clock frequency can be changed dynamically (7.8125 to 62.5 MHz) to meet the timing of external memories and speed up execution if running in internal memories only.

The iCPU has access to a standard RS232 interface, three timers, a watchdog, and four additional GPIO pins, one of which may be used for externally controlled program single-stepping.

Software synchronization and handshaking between the iCPU and a possible external CPU connected to the device's SI interface is made possible through a mailbox accessible by both CPUs.

The iCPU supports industry-standard compilers and assemblers.

MII Management Interface

Set up and status monitoring of connected PHYs is done through the built-in MII Management controller. It is capable of connecting to up to 32 PHYs.

The controllers are suitable for direct connection to standard RGMII PHYs and are accessed using the CPU interfaces.

General Purpose I/Os

The Stansted features five general purpose I/Os (GPIO). These pins are freely configurable as either inputs or outputs and are accessed through either CPU interface.

Functional Description

This section provides detailed information on aspects of the Stansted device's functionality that were highlighted in the features list and general description sections of this document. It should be noted, however, that many device features and settings not mentioned here. To learn more about any of these functions and settings, refer to the Configuration and Register Description descriptions of this document.

Introduction – A Packet's Life in the Stansted Device

In this section, several functional blocks of the Stansted that are referenced in Figure 1, the "Block Diagram", in Functional Overview section, are explained using a discussion of the "life" of a packet in the device, which is shown in Figure 2.

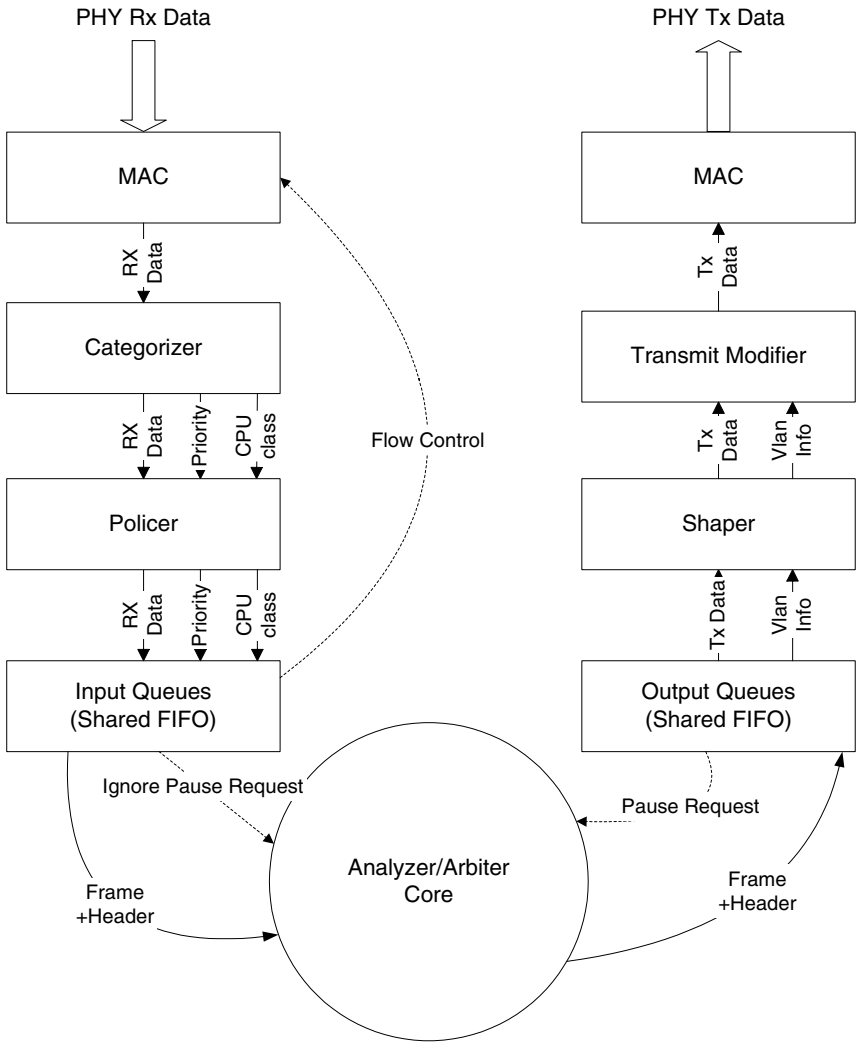


Figure 2. The Life of a Packet

Switch Engine Operation and Congestion Control

Each Stansted port is equipped with a MAC that receives and validates data from the PHY. From the Mac, the data is passed through the Categorizer for frame priority determination, the Policer for subscriber control. The data is then sent into one of the port ingress queues.

The packets are dequeued and forwarded on the internal frame bus as soon as it becomes available. The port or ports the packet is destined for receive the packet into one of the egress queues. As soon as the MAC is ready for transmission, the packet is dequeued and transmitted through the Shaper for subscriber control. The packet then goes through the modifier where a VLAN tag is added to the frame (if VLAN tagging is configured), and finally sent to the PHY.

Each port has a 16 kilobyte (kB) pool of memory that is shared between the ingress and egress queues for storing frames.

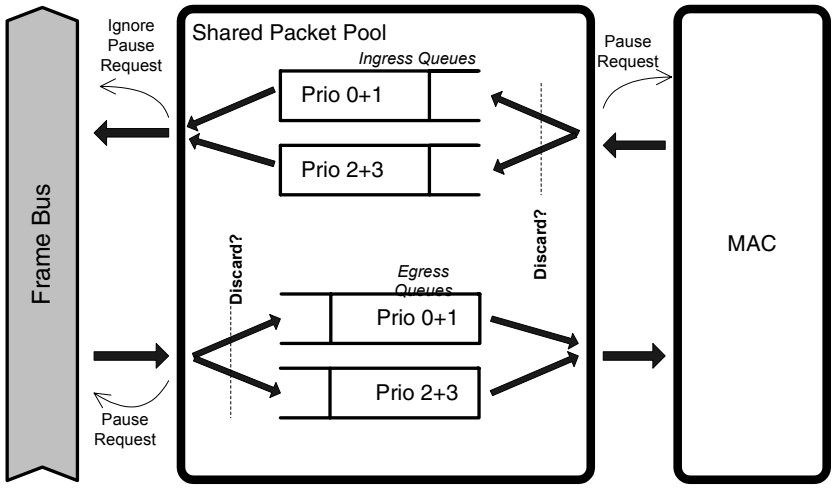


Figure 3. Shared Packet Pool

The priority determined by the Categorizer is used to select the ingress or egress queue during the enqueuing process. The priority ranges from 0 (lowest) to 3 (highest), and the traffic from different priority levels is, to some degree, split into different queues upon receipt. During dequeuing, the 2+3 queues are checked before the 0+1 queue. This causes the packets with higher priorities to overtake the packets with lower priorities.

Each port of Stansted can be programmed to handle congestion, either by discarding frames when specific fill levels are reached, or by requesting flow control from the sources of additional packets. The required type of congestion control for a switch application that uses the Stansted device is network dependant.

In the shared pool, the memory allocation unit is 256 bytes, and each port has its own 64 units of memory, resulting in a total of 16 kB of memory per port. For protection against overflow in the pool, the SharedFIFO register group contains various registers that can be used to select pause requests and discard conditions.

Priority Based Frame Discard

When a frame is added to a queue, the current amount of pending packet data is checked against a threshold. If adding the frame causes the queue to go over the threshold, the frame is discarded. The threshold is determined by the priority of the incoming frame. This allows a portion of the memory to be reserved for higher priority traffic.

The DROPCFG and MISCFIFO registers contain the thresholds, used as described in Table 1.

Table 1. Conditions Triggering a Frame Discard

Frame Priority	Discarded if Condition is Above
0	DROPCFG.LOW - MISCFIFO.SUBDIST
1	DROPCFG.LOW
2	DROPCFG.HIGH - MISCFIFO.SUBDIST
3	DROPCFG.HIGH

In the DROPCFG register, there are two thresholds, HIGH and LOW, for both the ingress and the egress. The thresholds are determined based on the total amount of ingress and egress data. For example, if DROPCFG.LOW=10, DROPCFG.HIGH=20, and MISCCONF.SUBDIST=4, ingress frames with a priority of 2 are dropped if more than 16 slices of ingress data are used; frames with a priority of 1 are dropped if more than 10 are used.

Pausing Streams using Flow Control

Another method of avoiding memory overflow is to request flow control from the data source. This is typically accomplished by asking the MAC to pause the link partner, which it does in cases of ingress overflow by issuing MAC pause frames (or throttle back collisions, when the device is configured for half duplex operation). In cases of egress overflow, flow control is accomplished by telling the Arbiter (see Figure 2) that no more room is available for egress packets.

There is a set of thresholds for each direction. They are located in the POOLCFG register. The set is used for hysteresis purposes. For example, if POOLCFG..INGRESSLOW=10, and POOLCFG.INGRESSHIGH=20 the MAC is asked to pause incoming frames when more than 20 slices of ingress data are used, and to cancel this request when less than 10 are left.

Advanced Drop Mode

When running the port in Drop mode, the possibility of overflow is avoided because frames are dropped depending upon their priority. When in Drop mode, the general performance of the device can be improved by using a limited version of flow control. This is done by enabling the internal flow control from egress back to ingress to some degree. This keeps the data in the ingress queues to avoid egress overflow. This has head-of-line blocking effects, which can be solved by enabling forward pressure on the ports (Ignore Pause in Figure 3). The effects for the two frame directions are discussed later in this document.

Egress

When the amount of egress data exceeds POOLCFG.EGRESSHIGH, the ingress ports are stopped from forwarding more data. In this case, the congestion is fed back to the ingress queues.

Ingress

If an egress port has blocked further transfers, the forwarding stops until the POOLCFG.INGRESSHIGH threshold is reached.

The forwarding pressure option is enabled through the POOLCFG register (Ingress Protection Method). It is enabled by default.

Flow Control Mode

When running a port in the flow control mode, no packet drop is allowed on the port, and memory depletion must be handled by flow controlling the source of additional data. In this mode, the DROPCFG register must be set to not drop any packets at all. This is accomplished by setting all thresholds to 63 because the theoretical maximum consumption from either ingress or egress is 61 memory units. The flow control mechanism is handled through the POOLCFG register.

Asymmetric flow control, where only ingress or egress has flow control enabled, is supported by enabling flow control in the desired direction only. Egress flow control is enabled through the "Obey Flow Control" option in the FCCONF register, and ingress flow control can be enabled through the "Ingress Protection Method" option in the POOLCFG register.

Flow Control Thresholds

Thresholds must be set with care to avoid packet loss in all cases (provided that no pause frames are lost). The amount of data that can be received from the moment the MAC is asked to send a pause frame until the last byte from the remote partner has been received is defined here:

2 x (maximum frame size) +	[media occupied]
2 x (cable length) x 5 bits x (speed / 1000) +	[data on cable]
250 bytes	[various reaction times]

This value varies for different speeds and cable lengths, as shown in Table 2. Standard Ethernet frame sizes are assumed with sizes up to 1526 bytes (see "Jumbo Frames," which begins on page 41).

Table 2. Flow Control Data Receivable After Pause Frame

Speed	Length			
	10 m	100 m	550 m	2000 m
10	3.1 kilobytes	3.1 kilobytes		
100	3.1 kilobytes	3.1 kilobytes		
1000	3.1 kilobytes	3.2 kilobytes	3.8 kilobytes	5.8 kilobytes

When setting the thresholds, it is beneficial to have as large a part of the pool memory as possible allocated for egress data. This can be accomplished by setting the thresholds using the values in Table 2 in place of the "X" in Table 3.

Table 3. Flow Control Thresholds to Avoid Packet Loss

Threshold	Value
Ingress High	4 kilobytes (at least two max size frames)
Ingress Low	3.5 kilobytes (a small hysteresis)
Egress High	15 kilobytes minus (Ingress High + X from Table 2)
Egress Low	15 kilobytes minus (Ingress High + X from Table 2)

The ingress thresholds can be set to other values than the ones suggested. The 15 kilobytes offset used for calculating the egress thresholds is derived from the 16 kilobytes available memory minus the maximum overhead for each queue.

As an example, the configuration for a 1G port on a 550 m media results in thresholds:

- Ingress High=4 kilobytes ~ register value=0x10 (256 bytes slices)
- Ingress Low=3.5 kilobytes ~ register value=0x0e
- Egress High=15 kilobytes – (4 kilobytes + 3.8 kilobytes)=7.2 kilobytes ~ register value=0x1d
- Egress Low=Egress High ~ register value=0x1d

MAC Features

Each of the 12 ports are equipped with a MAC that can operate in various modes. The modes are normally set up in the MACCONF register, where speed, duplex mode, and other options are selected. When changing modes, the port must be reset. This is done by writing the configuration word twice: first with the reset bits set, and then with the reset bits cleared. See Table 38, "CPU Transfer Mode - CPUMODE (Address 00h)", on page 87 for a detailed description on how to reset a port.

The more advanced features of the MAC are configured in the ADVPORTM register where the transmit clock can be inverted and halted. Half duplex settings such as the Frame Gap values are configured in the MACHDXGAP register.

The MAC supports frame lengths up to 12.2 kB. The maximum length accepted by the MAC is configurable in the MAXLEN register.

VLAN Features

The method used by each port to handle frame tagging and untagging is configured by the Port Controller as described here:

For ingress (Rx) frames, the processing of tagged and untagged frames can be configured as follows:

- For determining the priority of ingress frames:
  - If TAG prioritization is in use, the priority field in a tagged frame is used for internal prioritization.
  - If TAG prioritization is in use, all untagged frames get a configurable internal priority.
  - If TAG prioritization is not in use, the internal prioritization is based on the contents following the tag.
- For determining the VLAN membership:
  - The tag can be ignored and the VLAN membership is based solely on the priority determined by the payload. This is used when using Stansted as a VLAN-unaware switch.
  - The tag can be used for VLAN membership allocation. Priority tagged frames (VID=0) in this respect are not handled as VLAN tagged, and are treated as an untagged frame.
  - Untagged frames are allocated to a VLAN determined by the internal prioritization.

Egress (Tx) frame handling can be configured as follows:

- Do not tag frames. This is used when using the Stansted device as a VLAN-unaware switch or when the port is configured with respect to the VLAN as an ACCESS port.
- Tag all frames. This is used when the port is configured as a trunk port with respect to the VLAN.
- Tag all frames except those with a specific VID. This is used when the port is configured as a hybrid port with respect to the VLAN.

These parameters are configured for each individual port in the CATCONF, CATPVID, and TXUPDCFG registers.

Frame Categorization

Each port has a Frame Categorizer that determines the priority of ingress frames. This priority is in the range 0 to 3, and its use is described in the switching engine section. It is determined by information from layers 2, 3, and 4 within the frame. Several registers in the Categorizer Block Registers are used for configuring this.

Frame Priority Determination

Frame priority determination is based on a set of parameters that can be configured in each port. The parameters are mainly targeted at recognizing specific IP frame types, but other specific frames specified by Layer 2 header fields can be recognized also. The set of configurable parameters are listed in Table 4 and are shown in order of execution in Figure 4 (register values shown in the flow chart are not necessarily default values). Each FSx value in the table is actually a value set in the CATPRIO/CATSUBPRIO registers, which can be programmed from 00 (lowest priority) to 11 (highest priority). The priorities are programmed into the categorizer register as most significant bit in CATPRIO and least significant bit in CATSUBPRIO. As an example: FS1 is set to 1 by setting the FS1 field in CATPRIO to 0 and FS1 field in CATSUBPRIO to 1.

When classifying IP frames, only frames of type IPv4 are recognized as of IP type (IP header version field equals 4). A frame is recognized as an 802.2 frame if the Type/Length field is less than 0x600.

Table 4. Frame Priority Parameters

Parameter	Description
Tag Only	Priority is always taken from the IEEE802.1D TCI field. Untagged frames are given the priority FS2. If this option is NOT set, tag information is disregarded, except if CFI in the frame is set. In this case, the priority will be set to FS1.
CATDSAP	IEEE802.2 frames with SNAP encapsulated data will be classified as what is found in the encapsulated data. IEEE802.2 frames with DSAP matching this register, will get priority FS4. All other IEEE802.2 frames will get priority FS5.
CATETHT	Frames with matching Ethernet type (except IP frames) get priority FS6. IP packets are categorized based on the parameters below. All other frame types are assigned priority FS8.
CATPORTx	For TCP/UDP Frames: If their source or destination ports match one of these parameters, their priority is based on the IP DS field found in frame. Nonmatching port numbers result in priority FS7.
CATIPPRT	IP frames (non TCP/UDP) with matching protocol number get their priority based on the IP DS field found in the frame. All other IP frames get priority FS7.

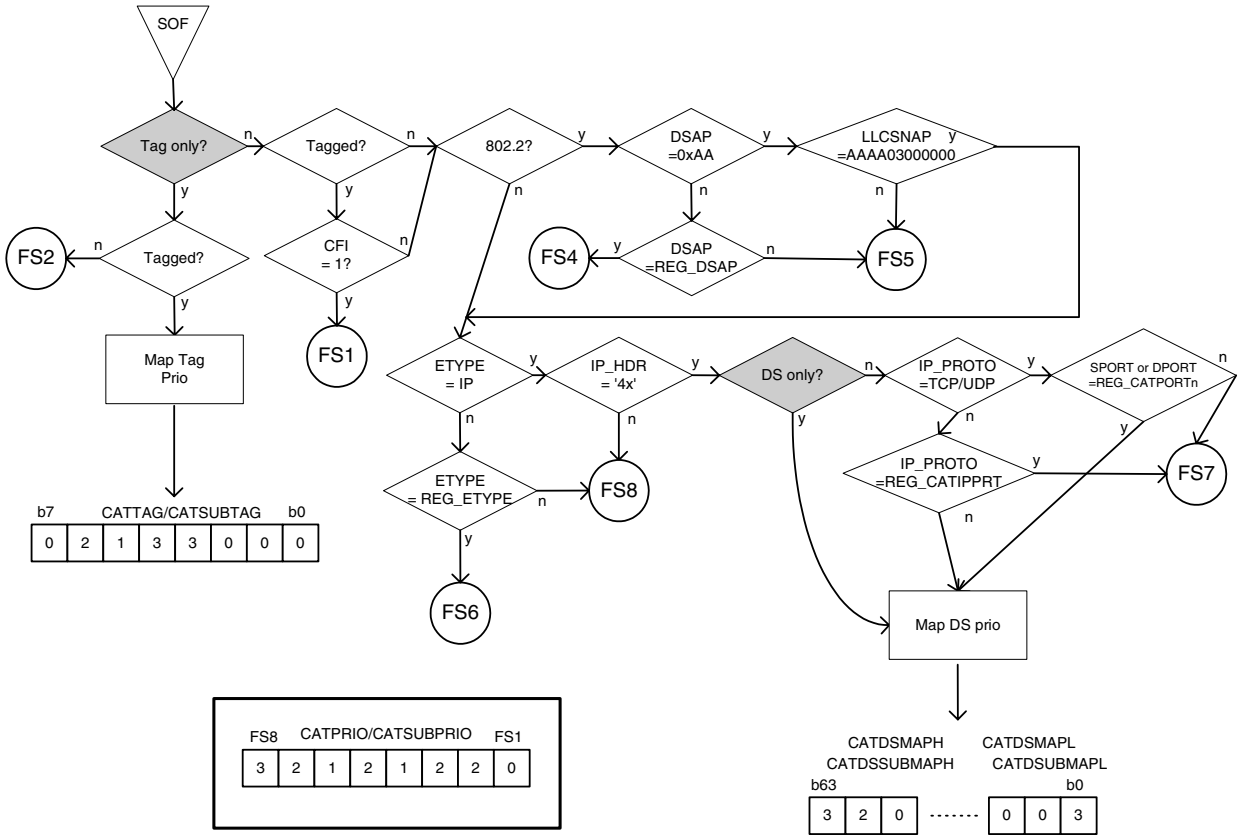


Figure 4. Categorizer Priority Assignment Flow Chart

Example:

All frames with Ethernet type 09A1h are assigned priority 2 and all other frames are assigned priority 0. To do this, set CATETHT=9A1h, CATPRIO.FS6=1, CATSUBPRIO.FS6=0, and all other FSs=0.

When the Tag priority is used, the CATTAG register is used to translate the 8 tag priority values into internal priority. When IP packets' DS fields are used for priority calculation (according to Table 4), the CATDSMAPL/CATDSMAPH + subregisters are used. DS are used either when an IP port or protocol match is found, or when the DS-only flag is set in the CATCONF register. Refer to the register list for further information.

Frame Class Determination

Another task for the Categorizer is to determine a class for each packet. This class is used in the frame analysis process to decide where to send the packet. The classes recognized by the Categorizer and the codes given to the Analyzer are shown in Table 5.

Table 5. Frame Classes

Frame Type	Condition	Class To Analyzer
Reserved Addresses (IEEE802.1D 7.12.6)	DMAC=0180C200000x (BPDUs and various Slow Protocols supporting Spanning Tree, Link aggregation, Port Authentication)	FORCECPUONLY (MAC control frames will be filtered by the MAC)
IP/ARP MAC broadcast	DMAC=BC, Type=IP/ARP	FORCECPU
IGMP	DMAC=01005Exxxxxx, Type=IP, IP type IGMP	FORCECPUONLY
IP Multicast Data	DMAC=01005Exxxxxx, Type=IP, non-IGMP, DIP outside 224.0.0.x	IPMCFLOOD
IP Multicast Ctrl	DMAC=01005Exxxxxx, Type=IP, non-IGMP, DIP inside 224.0.0.x	FORCECPU
Others	Others	NORMAL

For a class to be recognized, it must be enabled in the CATCONF register. The class is transferred to the Analyzer module, which uses it in the process of determining the set of egress ports to which each packet should be forwarded. Refer to Table 9.

Policing

The policing feature provides per-port data rate control at a fine granularity and operates on all frame types. There is also a separate Policer dedicated to broadcast and multicast traffic; this enables Broadcast Storm Control.

General Policer

The POLICECONF register is used to set up the Policer:

- Bit 24:16: “Bucket Threshold”: The bucket threshold, measured in units of 512 bytes
- Bit 11:0: “Data Rate”: The bucket is emptied with this rate. Rate unit is speed dependent:
  - 1G: 244 kbps
  - 100M: 48.8 kbps
  - 10M: 4.88 kbps

Each incoming packet causes the bucket level to be increased by a number equal to the packet size (without preamble and IFG) measured in bytes. The bucket level is decreased at a steady rate corresponding to the value of the Data Rate bit field.

If the bucket level reaches the Policer threshold, flow control frames are generated. Packets are not discarded. Over time, the bucket level will drop below the threshold. Then, the flow control condition is cleared again.

Broadcast/Multicast Policer

The Policer used for Broadcast and Multicast Storm Control is set up with the MCSTORMCONF register. Rate parameters are set up like the general Policer.

Shaping

Shaping allows you to manage the data rate for each port in great detail. To set up the shaper, use the SHAPECONF register as described here:

- Bit 24:16: “Bucket Threshold”: The bucket threshold, measured in units of 512 bytes
- Bit 11:0: “Data Rate”: The bucket is emptied with this rate. Rate unit depends on speed dependent as follows:
  - 1 G:244 kbps
  - 100 M:48.8 kbps
  - 10 M:4.88 kbps

Each packet leaving the switch causes the bucket level to be increased by a number equal to the packet size (without preamble and IFG) measured in bytes. The bucket level is decreased at a steady rate corresponding to the value of the Data Rate bit field.

If the bucket level reaches the Shaper threshold, packets are kept in the egress queue. As no packets are leaving the transmit queues, the bucket level is not increased. Over time, the bucket level will drop below the threshold. Then, packets will be allowed to leave the switch again.

Frame Analysis

Tables in the Analyzer

The central Analyzer module maintains a number of tables and masks. Table 6 lists the most important ones. Refer to the register list for more details.

Table 6. Basic Frame Analysis Data

Table/Register	Description
MAC Table	8192 stations learned by CPU or autolearned; each entry in the table is a MAC record
VLAN Table	4096 masks with allowed egress ports for each VID, and a flag – SourceChk - for checking the ingress port as being a member of the VLAN in which frames are received
Source Port Masks	12 port masks with allowed egress ports for each ingress port
Destination Port Masks	64 port masks with translation of logical port indexes to port masks
Aggregation Port Masks	16 port masks with allowed egress ports for each Aggregation Key
Unicast Flooding Mask	A port mask indicating where to send unicast packets with unknown destination addresses
Multicast Flooding Mask	A port mask indicating where to send multicast packets with unknown destination addresses; this mask also covers the broadcast address

Analysis Overview

The end result of a frame analysis is the Forwarding Mask. It contains the set of egress ports to which the analyzed frame are forwarded.

The decision to forward is made depending on header information from the incoming packets, user configured tables, autolearned information, and the CPU class reported by the Categorizer (see "Frame Class Determination," which begins on page 33).

The general flow of Frame Analysis is shown in Figure 5. Special cases (analysis exceptions) are not shown, but are described in "Exception Flags," which begins on page 40.

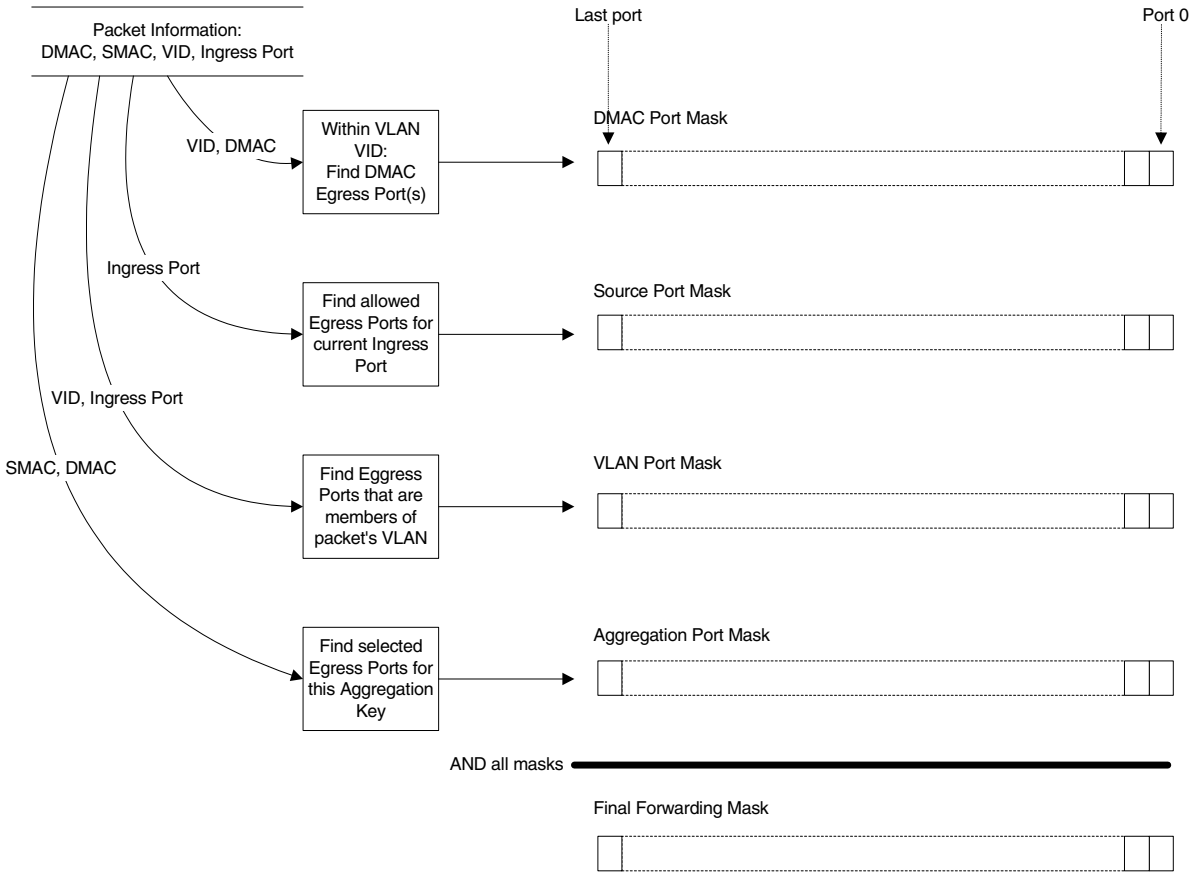


Figure 5. Frame Analysis

The Final Forwarding Mask is a set of egress ports, each of which receives a copy of the packet. This normally includes the ports that pass the four subanalysis steps shown in Figure 5.

The MAC Table

Stansted keeps a track of ports by their Destination MAC address by writing to and reading from an internal MAC address table. This table is used in the DMAC Analysis Block where it is checked if the DMAC address in the packet has been previously used as an SMAC address.

The MAC table consists of 8192 records with the contents described in Table 7.

Table 7. MAC Address Table

MAC Record Field	Description
MAC Address	The 48-bit Mac address (matched at lookup)
VLAN ID	The 12–bit VLAN ID (matched at lookup)
Destination Index	Destination Mask number
Aged Flag	Flag: aging has run since last learn of this address
Locked Flag	Flag: entry is locked. It will not be aged out nor overwritten
Valid	Flag: entry is valid
CPUCopy	Exception Flag: copy frames destined for this DMAC to the CPU buffer
FwdKill	Exception Flag: do not forward packets with this DMAC to any ports
Ignore	VLAN Exception Flag: do not apply VLAN port mask for packets with this DMAC

The table is automatically updated by an autolearning process, a CPU based learning process, or by direct manipulation from the CPU through register accesses (see "Direct MAC Table Access," which begins on page 36). The MAC address and VLAN ID are the identification of the station, the three flags hold the status of the entry, and the three exception flags are for special purposes, as explained later.

For autolearned ports, the Destination Index number is the port number it was learned from. It is used as a pointer into a table of 64 Destination Port Masks, each translating this logical index number into a set of egress ports. In standard setups, only the first 12 Destination Port Masks are used, and by default they are all configured to have only the port bit corresponding to their index set. So, if a station is learned on port 4, its record will have '4' in the destination index, and when packets are sent to the station, Destination Port Mask number 4 will be used—which is a port mask with only the port 4 bit set.

Direct MAC Table Access

As mentioned, apart from the more automated learning processes, it is also possible to directly access the table in order to dump the contents or to write specific data at specific locations. This is accomplished using the register interface.

The MAC table is organized as a 2048 x 4 entry hash table. When directly accessing the table, the MACTINDX is first written with the desired record table position, and a Read Entry or Write Entry command is then issued through the MACACCES register. Each operation autoincrements the index to ease dumping or updating the whole table.

Table 8. MAC Table Layout

Hash Key	0	1	2	3
0	MAC Record	MAC Record	MAC Record	MAC Record
1	MAC Record	MAC Record	(empty)	(empty)
2	MAC Record	(empty)	(empty)	(empty)
3	(empty)	(empty)	(empty)	(empty)
4	MAC Record	MAC Record	MAC Record	MAC Record
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
2047	MAC Record	MAC record	(empty)	(empty)

A MAC entry must be written into the correct hash chain in order to be found by the search engines. Therefore, when accessing the table for specific addresses, the table index must comply with the hash key formula, which is an XOR operation on most station identifications (VLAN, MAC).

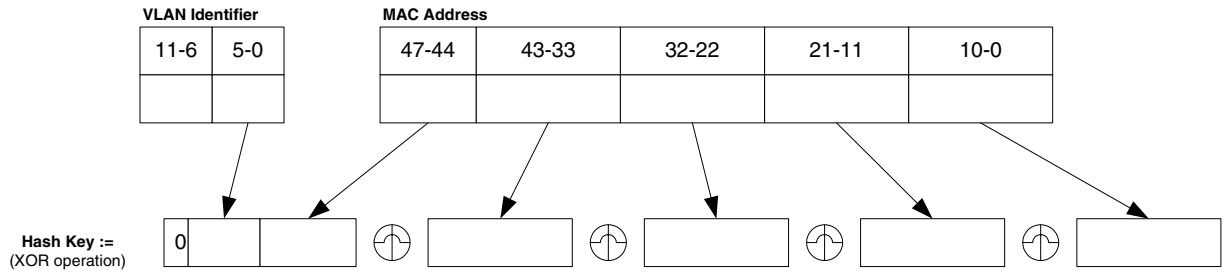


Figure 6. Hash Key Calculation

Example:

MAC address 00-00-12-34-56-78, in VLANID 15 has the hash key=XOR (0F0h, 000h, 048h, 68Ah, 678h)=4Ah=74 decimal.

Every time the search engines look up this address, hash chain 74 is searched.

The frame analysis process reorders each of the chains internally. In order to read out all four entries of a chain while the analysis engine is operating, a chain shadow register can be enabled using the MACTINDX register. When using this, a physical read from the table can only be accomplished by executing a read from bucket 0, and buckets 1–3 are stored in a shadow chain. Refer to the register description for more details.

Automatic Learning from Incoming Packets

Learning of stations is done automatically if learning is enabled for the ingress port. The learning process inserts an entry into the MAC table where:

- MAC = source MAC address from packet
- VID = Source VID from packet (or PVID if untagged)
- Destination Index = ingress port number
- Flags aged = locked = false, valid = true
- Flags Ignore VLAN, CPUCopy, FwdKill as configured in the AGENCNTL register

The learning occurs at wire speed on every packet received, regardless of a station already being present in the table. In this way, a station is learned every time it is seen as a source address.

The learning can be made dependent on the port being a member of the VLAN from which the frame is received by setting the VlanCheck flag in the ADVLEARN register.

Manually Manipulating MAC Table Entries using the CPU

MAC table entries can also be manually manipulated using the CPU Interface. When using this method, the CPU has full control over the flags and Destination Index inserted. This can be used for setting up permanent stations or multicast groups, where the locked flag must be set. In addition, it is used for CPU Based Learning (see "CPU Based Learning," which begins on page 48).

When setting up a permanent station on a port, the CPU can write an entry with the Destination Index pointing to the port where the station is permanently attached and the locked and valid bits are set.

When setting up a multicast group, an unused Destination Port Mask is first allocated. It is used to remember the set of egress ports that belong to the group. After that, an entry is created in the MAC Table with the Destination Port Index pointing to the allocated Destination Port Mask and the locked and valid bits are set.

Unlearning/Aging

By writing an AGE command to the MAC Table Command register (MACACCES), which subsequently causes the Analyzer to run through all entries, aging is performed. In such an aging pass, one of two things happens to each entry: if the aged flag is set, the entry is cleared; if the aged flag is clear, the flag is set. The only exceptions to this procedure are locked entries, which are not modified. In this way, all nonlocked entries that have not been relearned between two aging passes are removed.

Immediate unlearning of single MAC addresses is done by issuing a FORGET to the MACACCES register.

Frame Forwarding Decision

The analysis of where to send a packet is done in four separate steps (see Figure 5). Each of these steps takes different conditions into account, as explained in the following sections.

DMAC Processing

This block finds the possible egress ports based on the DMAC and VID found in the packet. The MAC table is searched for an entry with matching VID and DMAC, and the output mask from this step is the Destination Port Mask pointed to by the Destination Index found in the record. If the entry cannot be found, one of the globally configured Flooding Masks is instead returned causing the packet to be forwarded to all the ports that have their corresponding bit set to 1 in the Flooding Mask.

VLAN Processing

The VLAN processing part looks up the allowed set of egress ports for the reported VID. The mask containing this information is taken from a configured table of records (VLAN Table). By default, this table is set to all 1s in all records, which means that all ports are members of all VLANs. Each VLAN in the table can also be marked with the requirement that the ingress port must be a member of the VLAN if ingress packets are to be forwarded from this port. If this flag (“Source Check”) is set and the ingress port's bit is not set in the given VLAN Table entry, the resulting mask from this step is all 0s; the packet is not be forwarded to any port.

Source Port Processing

A packet must never be forwarded to its ingress port, as this results in network loops. This step ensures that this cannot happen. The ingress port number is used as an index into a table of 12 port masks, each containing the set of egress ports to which a packet from any of the ingress ports may be forwarded. By default, every port bit in every mask of this table is set to 1, except the bit corresponding to each port's number. The source port masks can also be used to facilitate VLANs only defined by the switch port on which the packets are received (port grouping or port based VLANs).

Aggregation Processing

Link aggregation is a method of logically grouping a set of ports so that two network stations can be interconnected using multiple links. All stations learned in a group of ports must be able to receive packets from all connections in the group. Also, the stations learned on any port in the group must be able to correctly forward frames to any other port in the group – like station B in Figure 7. This method is used to create more bandwidth between network nodes.

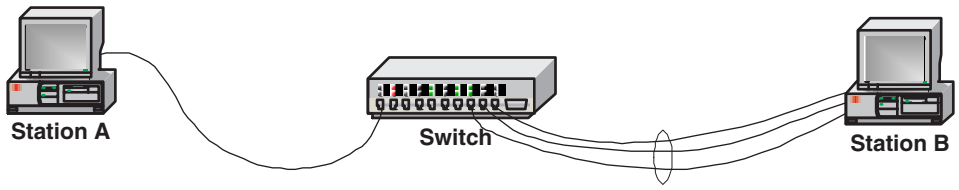


Figure 7. Aggregation Example

Some of the port masks mentioned previously must be set to reflect the groups:

- The Source Port Masks (SRCMASKS): For each port, this mask must be set so that an ingress port in a group cannot forward to any of the other ports in the same group.
- The Destination Port Masks (DSTMASKS): This mask must be set so that a station learned on a port in a group can receive frames from all ports in that group.

When using link aggregation, the Aggregation Port Masks (AGGRMSKS) are used to guarantee that any one frame is transmitted only on one port within a group. To achieve this, Stansted calculates an Aggregation Key with a value between 0 and 15 for every packet received. The value is calculated in one of three ways as determined by the AGGRCNTL register:

- The four least significant bits of the Source MAC Address (SMAC)
- The four least significant bits of Destination MAC Address (DMAC)
- The four least significant bits of (SMAC XOR DMAC)

When calculated, the Aggregation Key is then used as an index into a table of 16 Aggregation Port Masks. In a nonaggregated setup, these masks are set to all 1s (the default value), but when setting up one or more aggregated groups, the masks must be configured such that only one port bit is set in each group.

In short, aggregation works like this:

1. SMAC/DMAC is calculated into an Aggregation Key.
2. The Aggregation Key points to one of sixteen Aggregation Port Masks.
3. The selected Aggregation Port Mask shows which port in the group should get the packet.

Exception Flags

In the preceding, only the most simple analysis cases have been explained. As mentioned in "The MAC Table," which begins on page 36, some exception flags exist. Together with the class reported by the Categorizer on the ingress port, the exception flags can alter the way the masks are ANDed together.

Each entry in the MAC table has 3 exception flags:

- Ignore VLAN flag: The VLAN port mask is not ANDed with the forward mask.
- FwdKill: The Destination Port Mask is set to all 0s.
- CPUCopy: Frames for this address is copied to the CPU capture buffer.

Entries that have been autolearned have their exception flags set to the configured learn flags found in the AGENCNTL register. CPU learned entries have their exception flags set individually at the insert operation. During analysis, the flags found in the MAC record are used if the lookup succeeded. Otherwise, flooding flags (also found in the AGENCNTL register) are used.

Examples of the use of these exceptions are:

- GVRP frames are to be sent to the CPU buffer, and not forwarded elsewhere.
- Packets are to be held back for a specific station.
- Packets destined for a reserved management MAC address are to be sent to the CPU and not elsewhere

In the first case, an entry from the CPU with DMAC = 0180C2000021 (GVRP), valid = locked = 1, FwdKill = 1, CPUCopy = 1, is inserted. An entry must be inserted for each VLAN in which the frame can be received. As GVRP frames are untagged, this means that the possible VLANs are the set of PVIDs on the ports where GVRP frames can be received.

In the second case, a permanent entry with DMAC=the station, VID=the VLAN in which the station was found, valid = locked = 1, FwdKill = 1, CPUCopy = 0, is inserted.

The third case is handled in the same manner as the first case.

Categorizer Class

The final special case of the analysis flow is the Categorizer Class reported by the Categorizer block on the ingress port. As mentioned in "Frame Class Determination," which begins on page 33, four different classes exist. The default class is NORMAL. When this is reported, no change is made to the analysis flow as it has been described above. But if enabled in the CATCONF register, the other classes are reported. The meaning of these is explained in Table 9.

Basically, all the masks are ANDed. As previously mentioned, the flags reported by the DMAC process, the VLAN process, or the received Categorizer Classification can change this.

Table 9. Categorizer Classes

Categorizer Class	Meaning
IPMCFLOOD	If the DMAC for IPMC data frames is not found in the MAC table, the frames are flooded with the IFLODMSK if Enable IPMC Flood Mask is set in CATCONF. Otherwise, the MFLODMSK is used.
FORCECPU	The CPUCopy flag will be forced. This is used for IP/ARP MAC broadcast frames and for capturing ARP broadcasts. This is necessary when operating an IP end station through Stansted.
FORCECPUONLY	Both the CPUCopy and the FwdKill flags are forced. This is used for all BPDU type frames. BPDUs must never be forwarded to any port according to IEEE802.1D.
NORMAL	The CPUCopy and FwdKill flags are as reported from the DMAC process.

Port Mirroring

The frame analysis engine has an option to mirror all packets that either belong to a specific VLAN, or are otherwise received on specific ports.

The target port number is configured in the AGENCNTL register, and the source number (VLAN or port) is set in the VLAN table and source port masks. The CPU capture can also be mirrored in the sense that all frames destined for the CPU can be forwarded to the mirror port also. This is also set in the AGENCNTL register.

Jumbo Frames

Stansted supports jumbo frames up to 12.2 kilobytes. The default configuration of the Stansted device is set to a maximum length of 1518 bytes, but this can be changed through the MAXLEN register. A jumbo frame is any frame larger than the maximum normal-sized frame of 1526 bytes (double-tagged 1518-byte frame).

Jumbo frames are supported at all three speeds (10/100/1000 Mbps).

Because the shared pool for frames in each port is 16 kilobytes, the thresholds must be carefully chosen to allow flows of jumbo frames. Otherwise, the pool control system drops frames before they are fully stored. This is handled through the threshold (watermark) registers, DROPCFG and POOLCFG. In jumbo frame mode, frame loss must be expected in certain traffic scenarios, because the 16 kilobytes cannot hold both a pending 9.6-kilobyte ingress frame and a pending 9.6-kilobyte egress frame, which is needed in worst-case scenarios. For example, in a bi-directional jumbo frame throughput test, the maximum frame size for non-dropping behavior is limited to 8.2 kilobytes.

Jumbo frame performance is significantly affected by a number of factors such as threshold (watermark) settings or test configuration type (including port-to-port or fully meshed, unidirectional or bi-directional traffic, maximum frame length). For more information, contact your Vitesse Field Applications Engineer.

Early Transmission

For ports running 1 Gbps, a feature called Early Transmission is available to improve forwarding of jumbo frames of up to 8.2 kilobytes, especially enabling non-dropping bi-directional throughput. Basically, the feature is similar to a cut-through mode allowing an egress MAC to initiate transmission of a frame, before the transfer of the frame on the internal frame bus from ingress to egress is complete. The advantage is that less buffer is needed for storage in the egress port, because the buffer is being emptied at the same time as it is being filled. Early Transmission is enabled in the DROPCFG register.

If Early Transmission is enabled, the maximum length must be set to a maximum of 8.2 kilobytes and thresholds or watermarks in general must be carefully set. For more information, contact your Vitesse Field Applications Engineer.

QoS and Jumbo Frames

Because the memory for storing frames is limited to 16 kilobytes, it is not possible to have jumbo frames of multiple priorities flowing concurrently. Either all frames must be classified for equal priority, or only frames with the highest priority can be allowed to have jumbo size (by setting the lower priority thresholds much lower than the highest). For smaller sizes of jumbo frames, like 4 kilobytes, some degree of QoS would be possible.

Flow Control and Jumbo Frames

The use of flow control while simultaneously allowing jumbo frames has limitations. This is because the amount of excess data that can arrive between the time a pause frame is required to be transmitted and the time at which the remote partner stops transferring more data can be as much as two maximum sized frames. The sequence is:

1. The MAC starts transmission of a maximum size frame.
2. Shortly after, the thresholds trigger a pause frame transmission.
3. The MAC finalizes the (1) transmission and initiates the (2) pause frame.
4. The remote link partner initiates a maximum size frame transmission.
5. Shortly after, the remote partner receives the (3) pause frame.
6. The remote partner finalizes the (4) maximum size frame.
7. The remote partner suspends transmission due to (5).

The delay from (2) to (7) is more than two maximum sized frames, which, in the case of jumbo frames, is more than the available memory. It is therefore not immediately possible to utilize flow control and bidirectional jumbo frames on the same ports. If, however, the jumbo frames are flowing only in one direction, it is possible to configure the ports for proper flow control.

CPU Packet Transmit and Receive

It is possible to send and receive frames to and from any port via the CPU interfaces. This is useful for implementing traffic control protocols such as Spanning Tree Protocol, GVRP, IGMP snooping, and so forth for letting the CPU act as an end station on the network, or when using CPU based learning. The packet injection/readout is done through the register interface.

CPU Packet Transmission

The transmission of packets is done by injecting packets into the transmit queues of the ports that should transmit them. This is accomplished by writing certain registers in the shared FIFO block of the ports. The sequence is:

1. Write the frame length shifted 16 bits left to the CPUTXDAT register (length is in bytes, including CRC).
2. Write a fixed signature – 0x00000254 to CPUTXDAT.
3. Write the frame data including CRC field to CPUTXDAT.
4. If an odd number of writes occurred, write an extra word to CPUTXDAT.
5. Set the CPU Tx bit of the MISCFIFO register.

A write operation to the CPUTXDAT queue can internally take more time than the delay before the next operation. Therefore, the CPU must check that the queues are ready for additional data, which can be done in two ways:

- By checking the “CPU Tx Data Pending” bit in the MISCSTAT register for the port after each two words written
- By checking the “CPU Tx Data Overflow” bit in the MISCSTAT register for the port after the whole frame is written, but before the “CPU Tx” bit (step 5) is applied. If this bit is set, a rewind operation must be issued to the MISCFIFO register, and the frame must be rewritten to the queue. This can be done by either the “pending” or the “overflow” method as desired.

NOTE: *It is the responsibility of the switch control software to ensure a minimum frame size of 64 bytes, as Stansted does not pad automatically.*

Stansted can replace the CRC value set in the frame by the use of the TXUPDCFG register. This saves the CPU from having to calculate the correct CRC value.

CPU Packet Reception

As explained previously, it is possible to copy or redirect frames to the 16 kilobytes CPU buffer. Through the register interface, these packets can be read out and released from the buffer again so more frames can be stored. If the buffer space is depleted, additional frames targeted for the capture buffer will be dropped.

The frame availability status of the buffer can be read in the CPUCTRL register. Through this register, it is also possible to have an interrupt generated when packets are ready to be read.

By writing to the CAPREADP register, the first frame in the receive buffer is released, and the status bit of the CPUCTRL register is updated. The capture RAM is mapped into the register space as shown in Figure 8.

The address space for the capture buffer is always aligned so that the next nonreleased packet is read from the beginning of the buffer. The most significant byte in this address is the first byte of the frame header. When the packet is released, the address space is realigned to the next frame.

The CPU capture buffer is 16 kilobytes, but only 4 kilobytes can be seen at a time through the register interface. The first 1 kilobyte is accessed in Subblock 0, the second in Subblock 1, the third in Subblock 2, and the fourth in Subblock 3. For a single normal frame (which only have sizes of below 2 kilobytes [1526 bytes, in fact]), only the first two subblocks are needed. For further details about the register space, refer to "Chip Register Access," which begins on page 59. If jumbo frames are forwarded to the capture buffer with sizes larger than 4 kilobytes, they can only be read out by means of some manipulations with the stored headers. If this is needed, refer to the software API provided.

The frame header is shown in Figure 8. For a detailed description of each field, refer to Table 10.

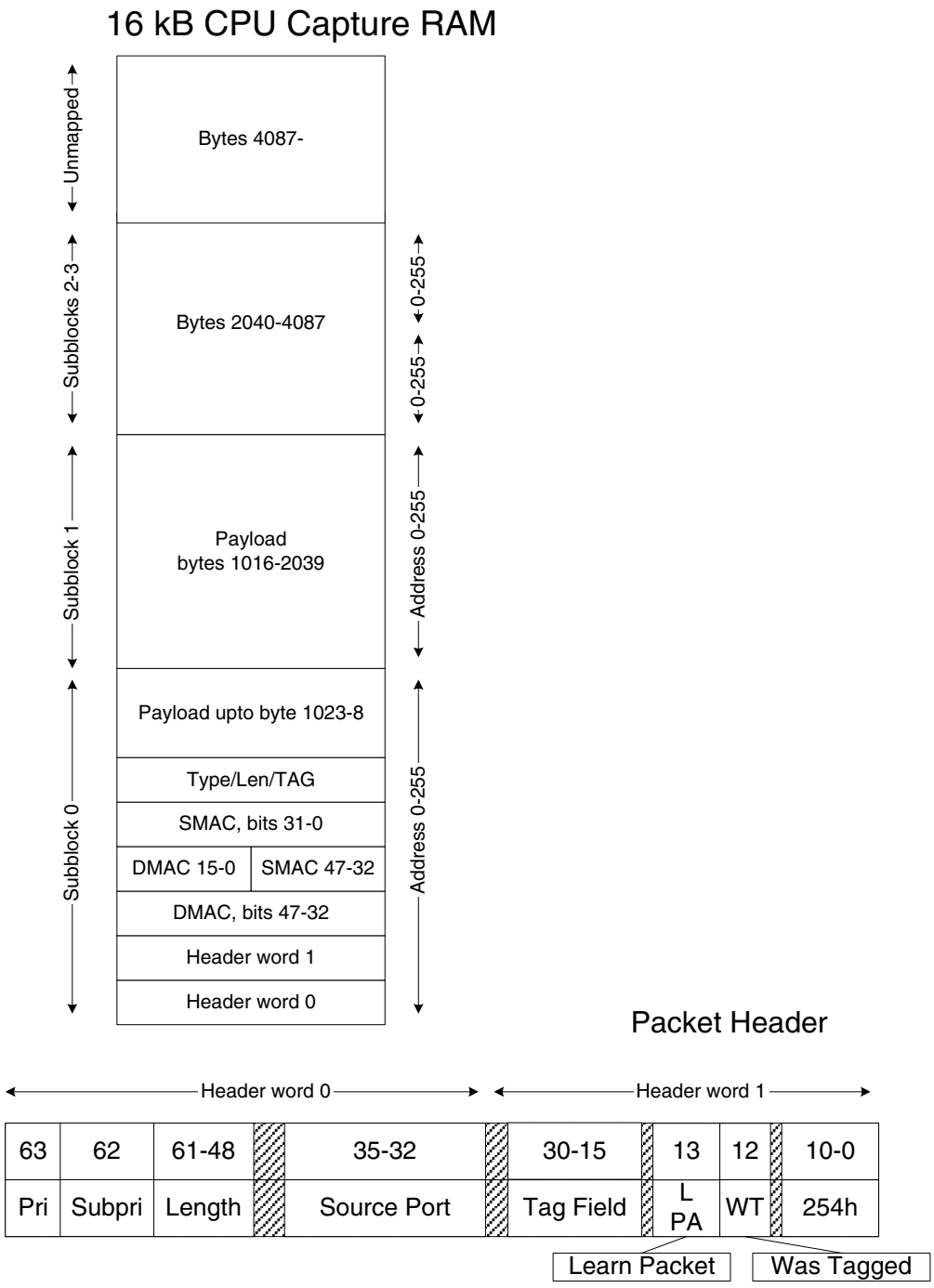


Figure 8. CPU Capture Buffer

Table 10. Packet Header Format

Header Field	Description
Pri	The priority decided by the Categorizer. 1=High
Length	The frame length excluding this header, including CRC
Source Port	The source port of the frame. 0-11
TagField	The tag field found in the frame. If frame was untagged, this field is set to the CATPVID register value
LPA	Learn Packet. Packet forwarded to CPU as a learn request
WT	Was Tagged. Packet was tagged when arriving at the ingress port
254h	This is a fixed signature checked by the internal queue system

Example:

An interrupt routine for reading out frame from the CPU Capture Buffer. (Notation for registers: block. subblock. Address. Further information about register addressing is found in "Chip Register Access," which begins on page 59).

```
Interrupt(
  If read(SYSTEM.0.CPUCTRL).CPU Rx_Frame_Ready = 1 then
     Hmsw = read(CAPTURE.0.0); Hlsw = read(CAPTURE.0.1)
     If NOT ((Hlsw and 0x7ff) = 0x254) then -- Unexpected error
     Pri:=(Hmsw >> 31) AND 0x1
     Len:=(Hmsw >> 16) AND 0x3FFF
     SrcPort:=Hmsw AND 0x1F
     Tag:=(Hlsw >> 15) AND 0xFFFF
     WasTagged:=(Hlsw >> 12) AND 0x1
     Learn:=(Hlsw >> 13) AND 0x1
     For x=0 to (len-1)/4 loop
        ofs=x+2 -- First two words are the buffer header!
        Pktdata(x)= read(CAPTURE.(ofs/256).ofs MOD 256)
     End loop
     If Learn AND (read(SYSTEM.0.CPUCTRL).Learn_Truncate = 1) then
        LenCaptured:=64
     Else
        LenCaptured:=Len
     End If
     Enqueue (pktdata,src,len,...)
     Write(CAPTURE.4.CAPREADP)
  End if
)
```

CPU Based Learning

Stansted autolearns all SMAC addresses into the MAC table by default. When more advanced learning policies are required, the CPU capture function can be used instead to allow CPU based learning (see "Manually Manipulating MAC Table Entries using the CPU," which begins on page 38). The learning method is configured through the ADVLEARN register, where it is possible to disable the autolearning mode and to request that all packets subject to learning or station move to be forwarded to the capture queue. It is also possible to have all learn packets forwarded to specific ports. This can be used for NPU attachment or for debugging purposes.

If packets from unlearned stations are to be discarded for security reasons, the learn drop mode can be enabled (also through the ADVLEARN register).

Through the CPUCTRL register, it is possible to truncate all learn frames forwarded to the CPU so that only the first 64 bytes are stored. The length field in the packet header holds the original packet length (see coding example above).

iCPU

The text in this section applies only when the iCPU is enabled; that is, when the ICPU_PI_En pin is strapped high. This section describes the additional features of the iCPU compared to a standard 8051 CPU. Refer to the data sheet or user's manual of a standard 8051 implementation for a description of the original 8051 characteristics.

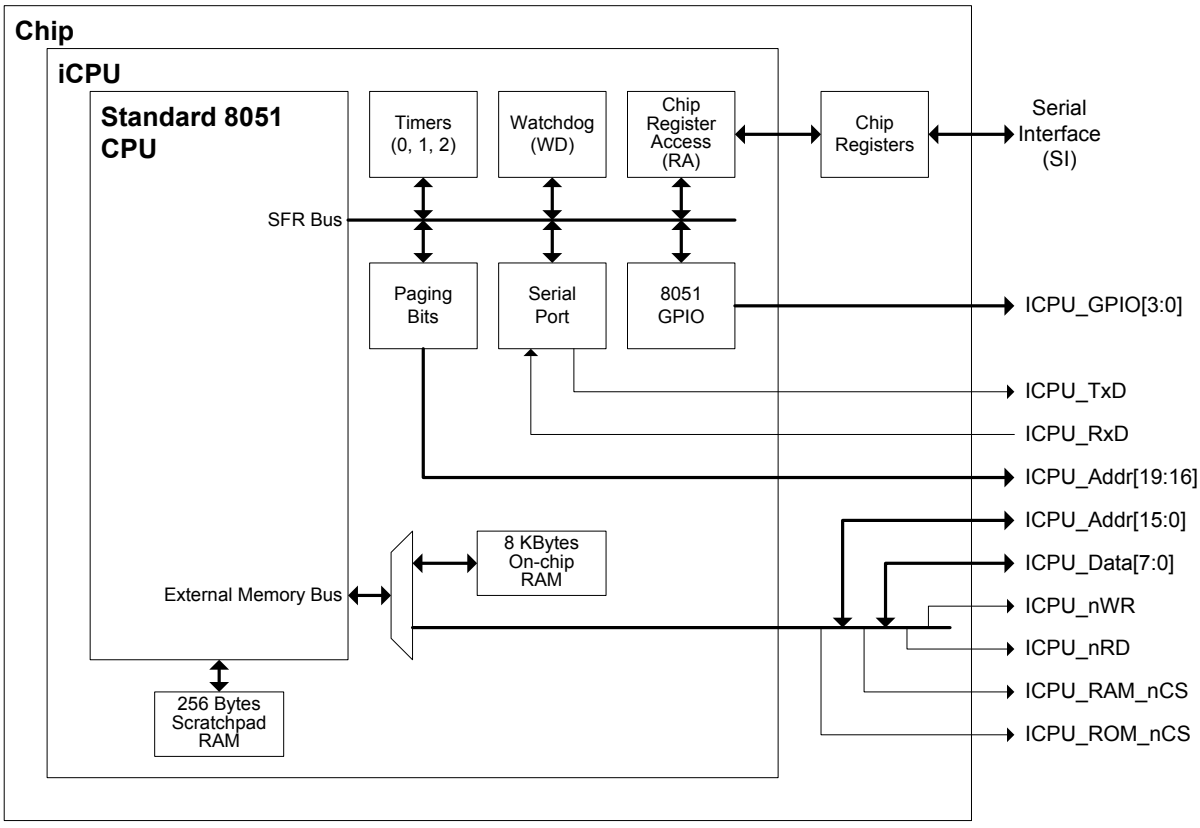


Figure 9. iCPU Block Diagram

Figure 9 provides an overview of the iCPU with its additional features and interfaces to the surroundings. The iCPU is designed to replace the functionality of an external CPU, effectively leaving out the necessity of external computational aids. Only a preprogrammed external Flash memory and—in case of managed software—external RAM is needed for booting up the on-chip CPU.

The shared use of the iCPUs RAM/ROM interface and Stansted's Parallel Interface makes the pin naming a bit awkward. To preserve the overview in this section, the pins are named in accordance with their functionality with respect to the iCPU. A mapping table between the logical pin names as shown on the right side of Figure 9 and the real pin names are provided in Table 13.

The following text refers to register names and fields in a symbolic form rather than in absolute addresses. These registers are described in tabular form under "Configuration," which begins on page 70. The registers are divided into two groups, SFR (Special Function Registers) registers and chip registers. Chip registers may be accessed by both an external CPU connected to the Serial Interface and by the iCPU, whereas SFRs are accessible only by the iCPU using MOV instructions.

To distinguish between the two register varieties, SFR references are prefixed with "SFR::" and chip registers are prefixed with "CHIP::".

Memory Organization

The 8051 CPU has separate address spaces for Program and Data memory. In the iCPU, Program Memory is accessible only through the External Memory Bus (see Figure 9). By default, 8 kilobytes of on-chip RAM overlay the upper 8 kilobytes of the 64 kilobytes address space. The 8051 fetches instructions only from this bus. The same bus is used for external data memory accesses (MOVX instructions). Instruction fetches and external data accesses do not occur simultaneously, which makes it possible to share the address and data buses for such accesses. To distinguish the accesses, the 8051 provides different control signals, as will be explained in the following sections. The same 8 kilobytes of on-chip RAM are by default multiplexed onto the external data space.

Besides the External Memory Bus, the 8051 incorporates an internal memory bus, which is used to access the Scratchpad RAM.

Scratchpad RAM

The Scratchpad RAM[1] is located in the "internal" data space, which is only 256 bytes wide and accessed via MOV instructions. The only difference from the standard 8051 implementation is that the Scratchpad RAM is 256 bytes rather than 128 bytes[2]. Direct accesses to addresses greater than 0x7F end up on the SFR bus. The additional 128 bytes of Scratchpad RAM may only be referred by indirect addressing. Indirect addressing always refers to the Scratchpad RAM, never to an SFR. Thus, to read the value contained in internal RAM address 0x98, the developer would need to code something similar to the following:

```
MOV R0, #0x98 ; Set the indirect address to 0x98
MOV A,  @R0   ; Read the contents of the Scratchpad RAM pointed to by R0.
```

This is very different from the following:

```
MOV A,  0x98  ; Reads the contents of SFR 0x98 (SCON)
```

1. In many standard 8051 data sheets, the "Scratchpad RAM" is known as the "Internal Data Memory". To distinguish this from the 8 kilobytes of on-chip RAM, "Scratchpad RAM" is used for the 256 bytes of RAM accessed through MOV instructions.
2. 8052 programmers are aware of this already, as the 8052 Microcontroller implements 256 bytes of Scratchpad RAM.

Which reads the value of SFR 0x98, which happens to be the SCON register.

The upper 128 bytes of the Scratchpad RAM is not bit accessible, but may be used as stack space.

Paging

Before describing the external memory layout, it is necessary to have a basic understanding of the paging mechanism. Basically, the 8051 is only able to access 64 kilobytes of external data or program memory, because its address bus is only 16 bits wide. The paging mechanism extends the width of the address bus by 4 bits. These 4 bits end up on the ICPU_Addr[19:16] pins. By connecting these pins to the most significant address bits of an external Flash or external RAM, the total memory space is extended up to 1 Mbyte.

Because the 8051 only holds instructions to access 64 kilobytes of memory, the page bits must be set programmatically. The PAGE SFR is intended for this purpose. This SFR contains two groups of four bits each, IFP[3:0] and OP[3:0]. The IFP group holds the four page bits used for instruction fetches and programmatic program memory reads (MOVC instructions) while the OP group holds the four page bits used for all other types of external memory accesses; that is, program memory writes, data memory reads, and data memory writes (and thus not program memory reads). By dividing the page bits into two chunks, the 8051 may, for instance, execute code from different pages of the Flash while reading data from a separately numbered RAM page. The PAGE SFR is located on an address divisible by 8, which makes it bit addressable (read-modify-write).

External Program Memory

Upon a reset, the 8051 begins execution of code from address 0x0000. By default, this location is mapped to external Flash or ROM as shown in Figure 11a.

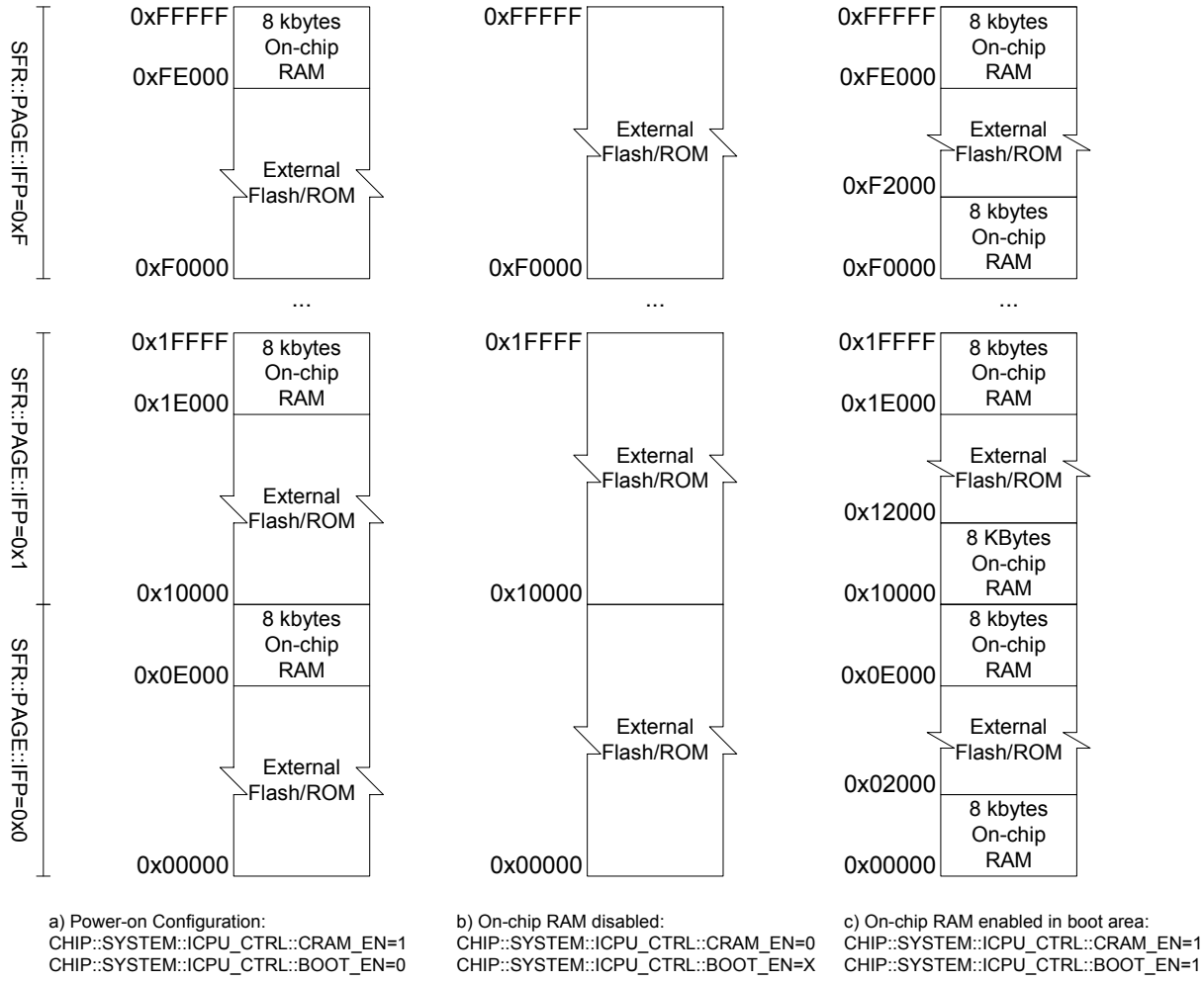


Figure 10. Three Possible Program Memory Layouts for Program Memory Reads

It is important to note that Figure 10 only shows the memory layout for program memory reads (both instruction fetches and programmatic reads using MOVC instructions). All program memory writes are forwarded to the external interface. This has the convenient effect that code can execute in the on-chip RAM while programming the flash-addresses that it overlays.

With the chip in the power-on configuration, the upper 8 kilobytes of each 64 kilobyte segment is overlaid with the on-chip RAM for program memory reads. This means that all instruction fetches in this area are forwarded to the on-chip RAM rather than to the external Flash. The left side of Figure 10 shows how the value of the IFP page bits affects the addresses accessed, and that the on-chip RAM is repeated throughout the 1 Mbyte of address space. The OP page bits take effect when *writing* to program memory.

The CHIP::SYSTEM::ICPU_CTRL register holds two bits for deciding how the on-chip RAM overlays the program memory data space—if at all. The CRAM_EN bit controls whether the on-chip RAM is mapped into the program memory space or not, and the BOOT_EN bit determines if it is also mapped into the lower 8 kilobytes of each page or

just the upper 8 kilobytes. CRAM_EN takes precedence over BOOT_EN, so that the value of BOOT_EN is ignored when CRAM_EN is 0.

Figure 10b shows the situation where CRAM_EN is 0; that is, where the on-chip RAM does not overlay any of the program memory addresses on reads. The value of BOOT_EN is insignificant in this situation.

Figure 10c shows the effect of setting the BOOT_EN bit with the CRAM_EN being 1. The on-chip RAM now occupies the lower 8 kilobytes of each page rather than the upper 8 kilobytes of each page. This is useful in situations where the code in the external Flash is corrupt and another image should be loaded into it. An external CPU attached to the Serial Interface can keep the iCPU reset while loading code into the on-chip RAM (see "External Access to On-Chip RAMs," which begins on page 67). The external CPU then sets the BOOT_EN flag, effectively mapping the on-chip RAM to address 0x0000, and then releases the iCPUs reset signal. The iCPU now executes out of on-chip RAM and the external Flash may now be erased and reprogrammed.

Care should be taken not to toggle the BOOT_EN bit when executing code in the lower 8 kilobytes of the program memory space (unless CRAM_EN is 0).

The standard 8051 instruction set does not include functionality for writing to program memory space. The iCPU does, however. The SPC_FNC SFR contains a bit, WRS, which—when set—causes writes to external memory to happen to Flash rather than to RAM. WRS=0 is the default, making writes compatible with the standard 8051. Thus, a function to write to flash could look like:

```
MOV  DPTR,     #0xABCD ; Load some address into the DPTR SFR
MOV  A,        #0x12   ; Load some value into A SFR
MOV  SPC_FNC, #1       ; Set WRS bit. Future writes are to external Flash
MOVX @DPTR,    A       ; Write value to external Flash
MOV  SPC_FNC, #0       ; Back to normal external RAM accesses.
```

Because all program memory writes are forwarded to the external interface independently of the setting of the CRAM_EN and BOOT_EN bits, it may seem impossible to load code into the on-chip RAM by executing instructions on the 8051 itself. This is not true, however. The on-chip RAM is shared between program and data memory accesses. Hence, by accessing the RAM as data it is possible to copy code into it, as described in the next section.

External Data Memory

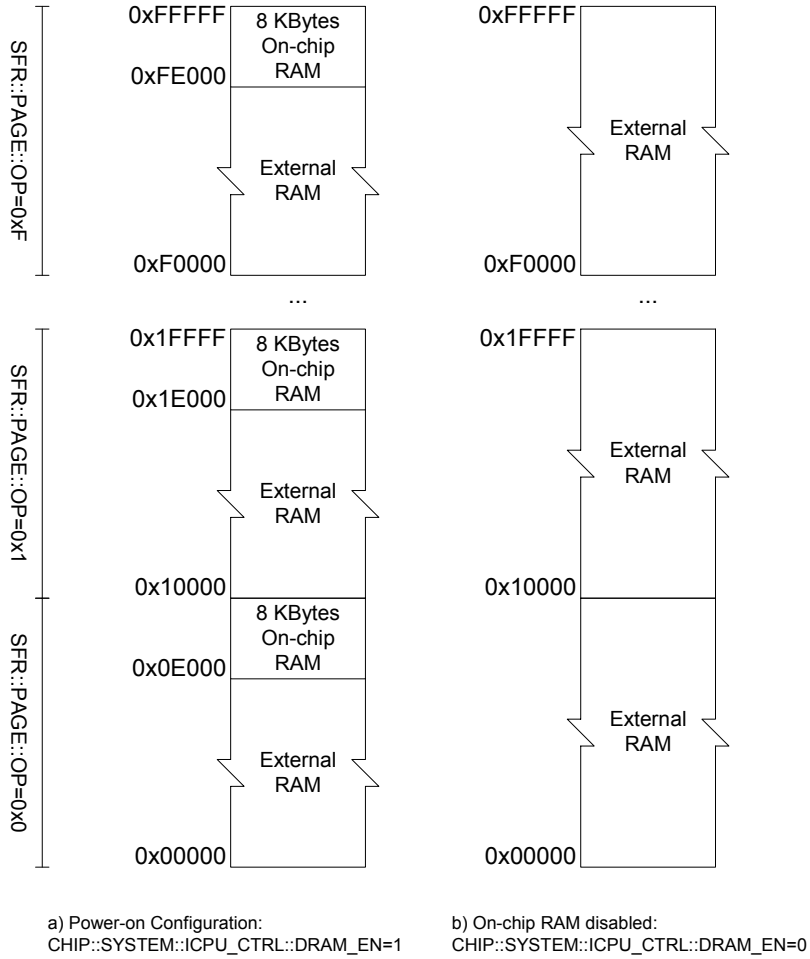


Figure 11. Data Memory Accesses with the On-Chip RAM a) Mapped In and b) Mapped Out

Both read and write accesses to external data memory use the same page bits, namely the OP group in the PAGE SFR. In the power-up configuration, the on-chip RAM overlays the upper 8 kilobytes of each page as shown in Figure 11a. Thus, all reads and writes in this region will be forwarded to the on-chip RAM rather than external RAM.

The DRAM_EN bit in the CHIP::SYSTEM::ICPU_CTRL register controls whether or not the on-chip RAM overlays this region. Figure 11b shows the situation where the on-chip RAM is mapped out of the external data memory space, so that all RAM accesses are forwarded to the external RAM.

Pin Name Mapping

Table 11 shows the mapping from the logical iCPU pin names used in this section to the pin names listed in Table 208, "Signal List by Ball Number", on page 147.

Table 11. Pin Name Mapping between iCPU and Parallel Interface

Pin Name (iCPU)	Pin Name (PI)	Pin Direction (iCPU)	Pin Direction (PI)
ICPU_GPIO[3:0]	PI_Data[15:12]	I/O	I/O
ICPU_Addr[19:16]	PI_Data[11:8]	O	I/O
ICPU_Addr[15:0]	PI_Addr[15:0]	O	I
ICPU_Data[7:0]	PI_Data[7:0]	I/O	I/O
ICPU_nWR	PI_nWR	O	I
ICPU_nRD	PI_nOE	O	I
ICPU_RAM_nCS	PI_nDone	O	O
ICPU_ROM_nCS	PI_nCS	O	I
ICPU_TxD		O	
ICPU_RxD		I	

External Memory Interface and Timing Configuration

The external memory interface is designed to attach most 8-bit RAMs and ROMs/Flashes. Because of the paging mechanism, up to 1 Mbyte of each type of memory may be connected, but the iCPU itself can only address 64 kilobytes.

The address and data buses (ICPU_Addr[19:16] and ICPU_Data[7:0]) are shared between RAM and ROM, which is possible because accesses occur at different time slots. The chip provides four control signals of which ICPU_nWR and ICPU_nRD control whether a read or a write access is in progress. These signals are active low and should be connected to both RAMs and ROMs. Two other control signals, ICPU_RAM_nCS and ICPU_ROM_nCS, determine whether a RAM or ROM access, respectively, is in progress. These signals are also active low and should be connected to RAMs and ROMs.

The interface is highly configurable so that both RAMs/Flashes that latch data on the falling edge or the rising edge of either the write or chip-select signal are supported. Thus, the chip implements configuration registers for both configuring the displacement and the width of the control signals (ICPU_nRD, ICPU_nWR, ICPU_RAM_nCS, and ICPU_ROM_nCS).

Read and write timing diagrams for RAM and Flash accesses are shown in "AC Characteristics," which begins on page 162.

Control Signal Width Configuration

The width of the control signals is determined by two parameters:

1. The iCPUs internal clock frequency, which at power-on is set to 1/16 of the system clock frequency corresponding to 128 ns per clock cycle. The clock period can be controlled via the CLK_DIV field in the CHIP::SYSTEM::ICPU_CTRL register and should never be set lower than the external memories can cope with.
2. The value of the SFR::CKCON::MD[2:0] bits. Table 12 shows the width of the control signals as a function of these three bits.

Table 12. Control Signal Width as a Function of the CKCON::MD2-0 SFR Value

MD[2:0]	Control Signal Width in 8051 Clock Cycles
0	2
1	4 (default)
2	8
3	12
4	16
5	20
6	24
7	28

The value of the MD[2:0] bits rules in all RAM access and Flash writes. Flash reads are not affected by the value of these bits. The 8051 clock frequency, however, still affects the width of Flash reads, which is always 2 8051 clock cycles.

Control Signal Displacement Configuration

The displacement of the control signals is controlled through the use of two chip registers, CHIP::SYSTEM::ICPU_RAM_CFG and CHIP::SYSTEM::ICPU_ROM_CFG. Reads and writes are controlled individually as described in the following sections.

RAM Reads

When the iCPU reads from external RAM, the parameters configured in the ICPU_RAM_CFG register govern. The register contains two fields, CHIP_SEL_READ_DELAY and READ_DELAY, which control how the ICPU_RAM_nCS and ICPU_nRD signals, respectively, are displaced with respect to the address. Each of the two signals can be displaced up to 24 ns in steps of 8 ns.

RAM Writes

Control signal displacement of write accesses to external RAM is configured through the ICPU_RAM_CFG register's CHIP_SEL_WRITE_DELAY and WRITE_DELAY fields. As for reads, the two signals can be displaced up to 24 ns in steps of 8 ns.

In addition, the iCPU can be configured to hold the written data (ICPU_Data) up to 24 ns longer than “normal”. This is controlled via the register’s WRITE_DATA_HOLD field.

ROM/Flash Reads

The displacement of control signals for ROM/Flash reads is configured through the ICPU_ROM_CFG register’s CHIP_SEL_READ_DELAY and READ_DELAY fields. The displacement can be up to 24 ns (the default being 8 ns which allows most 8-bit Flashes to work with Stansted during start-up).

ROM/Flash Writes

Writes to external Flash are controlled via the ICPU_ROM_CFG register’s CHIP_SEL_WRITE_DELAY and WRITE_DELAY fields. The displacement can be up to 24 ns.

In addition, the iCPU can be configured to hold the written data (ICPU_Data) up to 24 ns longer than “normal”. This is controlled via the register’s WRITE_DATA_HOLD field.

Power-On and Reset Timing

After power-on-reset, the 8051 clock frequency is set to 1/16 of the system clock frequency, or 128 ns. The first instruction fetch from the Flash’s address 0x0000 occurs approximately five 8051 clock cycles (640 ns) after release of the chip’s nReset signal.

Clock Frequency Select

The iCPU runs in its own clock domain and can be programmed to run at frequencies ranging from 7.8125 MHz to 62.5 MHz. Because the 8051 clock frequency directly reflects the timing on the external RAM/ROM interface, the slowest possible is chosen after a power-on reset; that is, 7.8125 MHz or 128 ns per clock period.

The frequency is controlled from the CLK_DIV field in the CHIP::SYSTEM::ICPU_CTRL register and is computed from:

$$F_{8051} = \frac{125}{CLK\_DIV + 1} MHz,\ CLK\_DIV \in [1;15]$$

After having selected an external Flash and RAM, the frequency can be trimmed to match the access time of the external memory. This new frequency needs to be programmed by the 8051 code after boot-up.

Reset Options

The device can be programmatically reset (soft-reset) in three different ways:

1. Reset the device including the iCPU
2. Reset the device excluding the iCPU
3. Reset only the iCPU

Reset Chip Including the iCPU

Resetting the whole device corresponds very closely to a power-on reset and causes all registers to return to their default values. This means that the whole device needs to be re-initialized from the very beginning after such a reset by following the guidelines in “Initialization Sequence,” which begins on page 80.

The reset is carried out by first performing a write to the CHIP::SYSTEM::GLORESET register with the STROBE field set and the remaining fields cleared. This is followed by a write to the same register, but this time with the MASTER_RESET field set and the remaining fields cleared.

Reset Chip Excluding iCPU

Leaving the iCPU from a chip-reset is sometimes desirable. The effect of such a reset is that all chip registers return to their default values, while the iCPU register space (SFR) retains its values. Among others, this implies that the programmed RAM/ROM displacement values (CHIP::SYSTEM::ICPU_RAM_CFG and ICPU_ROM_CFG) and the 8051 clock frequency configuration are lost and need to be reprogrammed, because these registers are located in the system domain. Care should be taken not to execute the resetting code out of on-chip RAM when this is mapped into the lower 8 kilobytes of code memory (when both BOOT_EN and CRAM_EN bits are set in the CHIP::SYSTEM::ICPU_CTRL register), since the reset will clear the CHIP::SYSTEM::ICPU_CTRL::BOOT_EN bit and thus switch from executing out of on-chip RAM to executing out of external Flash/ROM.

The reset is carried out by writing to the CHIP::SYSTEM::GLORESET register with both the ICPU_LOCK and STROBE bits set, leaving the remaining fields in the register cleared. This must be followed by a write to the same register, this time with MASTER_RESET field set and the remaining fields cleared. The last chip-register write must be followed by a write to the SFR::RA_DONE with the DONE bit set to clear the hardware state machine (see "Chip Register Writes," which begins on page 59). The device must be re-initialized by following guidelines in "Initialization Sequence," which begins on page 80.

Reset iCPU Only

The last type of reset is the one where only the iCPU is reset; the rest of the device keeps running unaffected. All chip registers thus retain their values, whereas the SFRs are reset to default. Consequently, when the iCPU boots up again, it keeps running at the previously programmed frequency and the external memories access configuration is also the same.

To invoke such a reset, first write to the CHIP::SYSTEM::SYSTEM_GLORESET register with the MEM_LOCK and STROBE bit set and the remaining fields cleared. Then write to the CHIP::SYSTEM::ICPU_CTRL register with the SOFT_RST bit cleared. This causes an immediate reset of the iCPU. The SOFT_RST bit gets auto-set by hardware when the reset is over.

This kind of reset is useful for instance when the on-chip RAM is loaded with boot-code and the user wishes to boot from it. Instead of just setting the SOFT_RST bit in the CHIP::SYSTEM::ICPU_CTRL register, the user would also set BOOT_EN bit. This will cause the on-chip RAM to be mapped to address 0x0000 of the program memory space, effectively causing the boot to be from on-chip RAM.

The iCPU may be kept reset using the CHIP::SYSTEM::ICPU_CTRL register's SOFT_RST_HOLD bit. To enable this feature, set SOFT_RST_HOLD to 1 while clearing the SOFT_RST bit.

Interrupts

Table 13 lists the interrupts available with the iCPU organized in decreasing natural interrupt priority.

Table 13. SFRs Associated with a Particular Interrupt

Interrupt Name	Flag(s)	Enable	Priority Control	Edge/Level Sensitivity	Interrupt Vector	Interrupt Number
External Interrupt #0	TCON::IE0 RA_DONE::DONE	IE::EX0	IP::PX0	TCON::IT0	0x0003	0
Timer 0 Interrupt	TCON::TF0	IE::ET0	IP::PT0	—	0x000B	1
External Interrupt #1	TCON::IE1 GPIO_IN::GI0	IE::EX1	IP::PX1	TCON::IT1	0x0013	2
Timer 1 Interrupt	TCON::TF1	IE::ET1	IP::PT1	—	0x001B	3
Serial Port Tx or Rx Interrupt	SCON::TI SCON::RI	IE::ES	IP::PS	—	0x0023	4
Timer 2 Interrupt	T2CON::TF2	IE::ET2	IP::PT2	—	0x002B	5

The IE and IP SFRs provide interrupt enable and priority control, as with the standard 8051. The IE::EA bit (Enable All) is a global enable for all interrupts. When IE::EA is 1, each interrupt is enabled/masked by its individual enable bit in the IE register.

When an enabled interrupt occurs, the CPU vectors to the address of the interrupt service routine (ISR) associated with that interrupt. The vector address is shown in column 6 of Table 13. The iCPU always completes the instruction in progress before servicing an interrupt. If the instruction in progress is RETI, or a write access to any of the IP or IE SFRs, the iCPU completes one additional instruction before servicing the interrupt.

The interrupt latency depends on the current state of the iCPU. The fastest response time is five instruction cycles, that is, twenty 8051 clock cycles: one to detect the interrupt and four to perform the LCALL to the ISR. The maximum latency occurs when the iCPU is currently executing an RETI instruction followed by a MUL or DIV instruction. The thirteen instruction cycles (52 clock cycles) in this case are: one to detect the interrupt, three to complete the RETI, five to execute the DIV or MUL, and four to execute the LCALL to the ISR.

Each ISR ends with an RETI (return from interrupt) instruction. After executing the RETI, the CPU returns to the next instruction that would have been executed if the interrupt had not occurred. An interrupt can only be interrupted by a higher priority interrupt. The interrupts listed in Table 13 are organized in decreasing natural priority. In addition to the natural priority an interrupt can be assigned a high or low priority level, which takes precedence over the natural priority. This selection is made with the IP SFR. Simultaneous interrupts with the same priority level (for example, both high) are resolved according to their natural priority. Once an interrupt is being serviced, only an interrupt of higher priority level can interrupt the service routine of the interrupt currently being serviced.

The two external interrupts—External Interrupt #0 and #1—are connected to the RA module (Register Access) and to the GPIO module, respectively. In the standard 8051, these two interrupts are active low. In the iCPU implementation they become active when the RA_DONE::DONE bit transitions from a 0 to a 1 (External Interrupt #0) or when the GPIO_IN::GI0 bit transitions from a 0 to a 1 (External Interrupt #1). When active, the IE::IEx ($x \in [0; 1]$) returns a 1. These two interrupts can be programmed through the TCON::IT0 and TCON::IT1 bits to be either edge or level sensitive. For example, when TCON::IT0 is 0, the interrupt is level sensitive and the iCPU sets the TCON::IE0 flag

when the RA_DONE::DONE bit samples high. When TCON::IT0 is 1, the interrupt is edge sensitive and the iCPU sets the TCON::IE0 flag when the RA_DONE::DONE is sampled low then high on consecutive samples.

Chip Register Access

The iCPU uses indirect access in order to read and write Stansted's internal chip registers. The RA module located on the SFR bus interfaces between the iCPU and the chip registers.

Chip registers are 32 bits wide and are addressed through a 3–bit block number, a 4–bit subblock number, and an 8-bit register address. The eight SFRs (RA_xxx) making up the RA module contains registers for holding this information before initiating a read or a write access. Accesses vary in execution time and a flag (SFR::RA_DONE::DONE) tells when an access is complete. The following two sections explain write and read accesses.

Chip Register Writes

Four registers in the RA module are used to hold the 32–bit data value to write to the chip register. These four SFRs are named RA_DA0 through RA_DA3 with RA_DA0 designated to hold the least significant byte and RA_DA3 the most significant byte of the 32–bit value.

The RA_BLK register contains two fields, BLOCK and SUBBLOCK, into which the 3–bit block number and 4–bit subblock number of the register to access must be written.

Finally write the register address within the block and subblock into SFR::RA_AD_WR. This operation will initiate the real write access, effectively sending the data held in RA_DA0-3, RA_BLK, and RA_AD_WR to the chip register.

Only one chip register access can be in progress at a time, and the access time may vary from register to register, so all write operations should enter a loop where it continuously polls the SFR::RA_DONE::DONE bit. As long as this bit is cleared, the write operation is still in progress and another access cannot be initiated. When the bit gets set, the write operation is complete, and software must clear it by writing a 1 to it.

There is one situation where the DONE bit is never set by hardware, however, and that is when the iCPU resets Stansted without resetting itself. In this case, the DONE bit never gets set, but a 1 must still be written to it.

The DONE bit is also connected to the External Interrupt #0 (SFR::IE::EX0) allowing for interrupt driven operations rather than polled operations.

The following code snippet writes 0x89ABCDEF to the chip register with block number 0x1, subblock number 0xA, and address 0x12

```
MOV RA_DA3,   #0x89 ; Bits[31:24] of data to write
MOV RA_DA2,   #0xAB ; Bits[23:16] of data to write
MOV RA_DA1,   #0xCD ; Bits[15: 8] of data to write
MOV RA_DA0,   #0xEF ; Bits[ 7: 0] of data to write
MOV RA_BLK,   #0x2A ; Write to block 0x1, subblock 0xA
MOV RA_AD_WR, #0x12 ; Initiate the write by specifying reg. address (0x12)
poll:JNB RA_DONE,  poll  ; Wait for the write to complete, i.e. DONE bit to get set
MOV RA_DONE,  #1    ; Prepare for next access by clearing the DONE flag
```

The data written to the RA_DAx and RA_BLK registers is persistent. This means that if the same value should be written to several register addresses within the same block/subblock, new write-accesses can be initiated by simply writing a new value to the RA_AD_WR register followed by a poll of the DONE bit.

Chip Register Reads

Programming a read access is very much like programming a write access, except that the RA_DAx registers are read from rather than written to. As for the write access, fill in the RA_BLK with the block number and subblock number of the register to read, and initiate the read by *writing* the register address to RA_AD_RD. Poll the RA_DONE::DONE bit until it gets set, and assemble the returned data by reading the RA_DAx SFRs. Reading the RA_DAx registers also clears the DONE flag. Because the DONE bit is connected to External Interrupt #0, an interrupt driven approach also may be implemented.

The following code snippet shows an example of a read of a chip register with block number 0x7, subblock number 0 and register address 0x10 (this happens to be the CHIP::SYSTEM::ICPU_CTRL register).

```
  MOV RA_BLK,   #0xE0  ; Read from block 0x7, subblock 0x0
MOV RA_AD_RD, #0x10  ; Initiate the read by specifying reg. address (0x10)
poll:JNB RA_DONE,  poll ––; Wait for the write to complete, i.e. DONE bit to get set
MOV R3,       RA_DA3 ; Bits[31:24] of returned data. Also clears DONE bit
MOV R2,       RA_DA2 ; Bits[23:16] of returned data.
MOV R1,       RA_DA1 ; Bits[15: 8] of returned data.
MOV R0,       RA_DA0 ; Bits[ 7: 0] of returned data.
```

The returned data remains in the RA_DAx registers until the next read completes. In fact, data written to the RA_DAx registers is different from the data returned by reading from the registers. Therefore, it is not possible—in the write function—to verify the data just written to them, because a read will return a different set of register values.

Watchdog

The Watchdog SFR module provides a means of preventing 8051 software from running wild. This is achieved by requiring that software periodically writes alternating values to the watchdog, once enabled. If it fails to do so within the watchdog period, which is approximately 1 second, the watchdog resets the iCPU and sets a sticky–bit in a chip register. This bit, CHIP::SYSTEM::ICPU_CTRL::WATCHDOG_RST, can be tested by the 8051 software during boot-up to see the reason for the boot. If it is 1, the watchdog has reset it, if not, it was a normal boot. Write a 1 to the sticky–bit to clear it.

A reset caused by the watchdog only involves the iCPU; the rest of the chip is unaffected. This corresponds very closely to an iCPU-reset as described in "Reset iCPU Only," which begins on page 57.

The Watchdog module implements two SFRs, one for enabling and disabling the watchdog (WDCON) and another for keeping the watchdog alive (WDDA).

To enable the watchdog, write a 1 to SFR::WDCON::WD_EN. Once enabled, software must alternately—and with at most 1 second interval—write the two values 0xBE and 0xEF to the WDDA register, the first being 0xBE. If software writes a value different from the expected, the watchdog issues a reset immediately. After waking up from a reset, the watchdog is always disabled, which prevents subsequent resets from occurring.

The watchdog can be disabled at any time by clearing the SFR::WDCON::WD_EN bit. Disabling the watchdog also resets the watchdog timer, so that a subsequent reactivation leaves software with 1 second before it needs to write to WDDA. It is strongly recommended that the watchdog be disabled while programming the Flash.

Timer 2

The standard 8051 timers (Timer 0 and Timer 1) are 8-bit timers/counters only, whereas Timer 2 offers 16-bit capabilities. This makes Timer 2 suitable as baud-rate generator for the serial port, because it offers much better granularity for the baud-rates (see also "The Serial Port," which begins on page 62).

The modes supported with Timer 2 are:

- 16-bit auto-reload timer/counter
- Baud-rate generator

Timer 2 is an SFR module and is thus configured through the SFR bus.

The default count-rate for all three timers is the same as for the standard 8051 timers, namely 12 clock cycles per increment. This can be changed to 4 clock cycles per increment by setting the SFR::CKCON::TxM field to 1, where 'x' is either 0, 1, or 2.

16-bit Timer/Counter Mode with Auto-Reload

Figure 12 shows how Timer 2 operates in timer/counter mode with auto-reload. The T2CON::TR2 bit enables the counter. When it increments from 0xFFFF the T2CON::TF2 flag gets set and the starting value is reloaded into TL2 and TH2 from RCAP2L and RCAP2H, which are programmed by software. Upon overflow, an interrupt also occurs if the Timer 2 interrupt is enabled (IE::ET2) and interrupts are globally enabled (IE::EA).

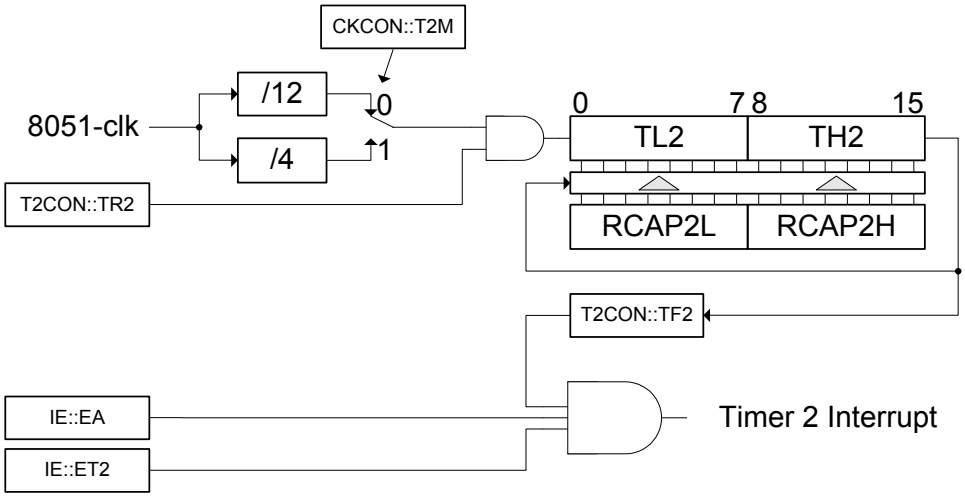


Figure 12. Timer 2 as Timer/Counter with Auto-Reload

Baud-Rate Generator Mode

Setting either T2CON::RCLK or T2CON::TCLK to 1 configures Timer 2 to generate baud rates for the Serial Port in serial mode 1 or 3. In baud-rate generator mode, Timer 2 functions in auto-reload mode. However, instead of setting the TF2 flag, the counter overflow generates a shift clock for the serial port function. As in normal auto-reload mode, the overflow also causes the preloaded start value in the RCAP2L and RCAP2H registers to be reloaded into the TL2 and TH2 registers.

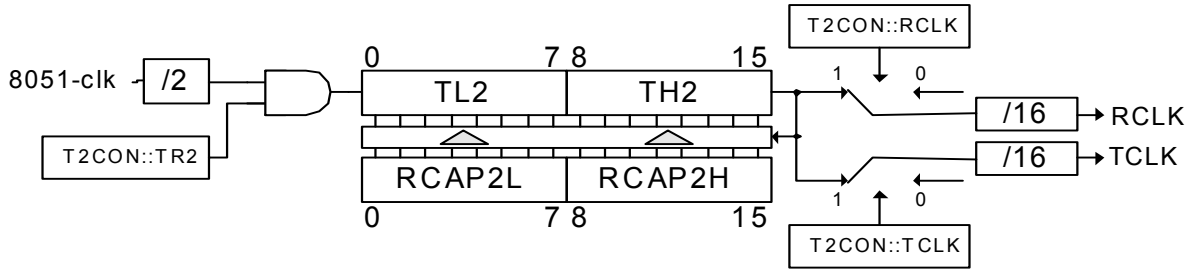


Figure 13. Timer 2 as Baud Rate Generator

The counter time base in baud-rate generator mode is 8051-clk/2.

The Serial Port

The serial port provided in the iCPU is identical in operation to the standard 8051 serial port. This section is provided in order to show how the serial port works with Timer 2, which is not part of the standard 8051 processor. The description will take basis in the serial port's Mode 1. Mode 1 provides standard asynchronous, full-duplex communication using a total of ten bits: one start bit, eight data bits, no parity, and one stop bit (also known as "8N1"). The serial port transmits on the chip's ICPU_TxD pin and receives on the ICPU_RxD pin.

The baud rate is a function of timer overflow. The serial port can use either Timer 1 or Timer 2 as baud-rate generator, but in the following only Timer 2 is described. Given a desired baud rate, *b*, the value to load into RCAP2L and RCAP2H of Timer 2 is computed from:

$$RCAP2H,RCAP2L = Round\left(65536 - \frac{f_{8051}}{32 \cdot b}\right)$$

Where RCAP2H, RCAP2L is the value to write to RCAP2H and RCAP2L takes as a 16-bit unsigned integer, and f_{8051} is the iCPUs current clock frequency. The "32" in the denominator is the result of the clock being divided by 2 and the Timer 2 overflow being divided by 16 (see Figure 13). The actual baud rate given an integer count can thus be computed from:

$$b = \frac{f_{8051}}{32 \cdot (65536 - RCAP2H, RCAP2L)}$$

Because it is the 8051 clock frequency that forms the basis for the values to load into RCAP2x, it may be that not all baud rates are achievable with sufficient accuracy for all frequencies. Table 14 shows the deviation from the desired baud rate as function of 8051 clock frequency for commonly used baud rates. As a rule of thumb, baud rates that cause deviations larger than 3% should be avoided.

Table 14. Deviation Between Desired and Actual Baud Rate for Selected Clock Frequencies and Desired Baud Rates

CLK_DIV [1]	f_{8051} [Hz]	Desired Baud Rate [bps]	RCAP2x	Actual Baud Rate [bps]	Deviation [%]
1	62,500,000	9,600	0xFF35	9,621.3	0.2
		57,600	0xFFDE	57,444.9	0.3
		115,200	0xFFEF	114,889.7	0.3
2	41,666,667	9,600	0xFF78	9,574.1	0.3
		57,600	0xFFE9	56,612.3	1.7
		115,200	0xFFF5	118,371.2	2.8
3	31,250,000	9,600	0xFF9A	9,574.1	0.3
		57,600	0xFFEF	57,444.9	0.3
		115,200	0xFFF8	122,070.3	6.0
4	25,000,000	9,600	0xFFAF	9,645.1	0.5
		57,600	0xFFF2	55,803.6	3.1
		115,200	0xFFF9	111,607.1	3.1
5	20,833,333	9,600	0xFFBC	9,574.1	0.3
		57,600	0xFFF5	59,185.6	2.8
		115,200	0xFFFA	108,506.9	5.8
6	17,857,143	9,600	0xFFC6	9,621.3	0.2
		57,600	0xFFF6	55,803.6	3.1
		115,200	0xFFFB	111,607.1	3.1
7	15,625,000	9,600	0xFFCD	9,574.1	0.3
		57,600	0xFFF8	61,035.2	6.0
		115,200	0xFFFC	122,070.3	6.0
8	13,888,889	9,600	0xFFD3	9,645.1	0.5
		57,600	0xFFF8	54,253.5	5.8
		115,200	0xFFFC	108,506.9	5.8
9	12,500,000	9,600	0xFFD7	9,527.4	0.8
		57,600	0xFFF9	55,803.6	3.1
		115,200	0xFFFD	130,208.3	13.0

Table 14. Deviation Between Desired and Actual Baud Rate for Selected Clock Frequencies and Desired Baud Rates *(continued)*

CLK_DIV [1]	f_{8051} [Hz]	Desired Baud Rate [bps]	RCAP2x	Actual Baud Rate [bps]	Deviation [%]
10	11,363,636	9,600	0xFFDB	9,597.7	0.0
		57,600	0xFFFA	59,185.6	2.8
		115,200	0xFFFD	118,371.2	2.8
11	10,416,667	9,600	0xFFDE	9,574.1	0.3
		57,600	0xFFFA	54,253.5	5.8
		115,200	0xFFFD	108,506.9	5.8
12	9,615,385	9,600	0xFFE1	9,692.9	1.0
		57,600	0xFFFB	60,096.2	4.3
		115,200	0xFFFD	100,160.3	13.1
13	8,928,571	9,600	0xFFE3	9,621.3	0.2
		57,600	0xFFFB	55,803.6	3.1
		115,200	0xFFFE	139,508.9	21.1
14	8,333,333	9,600	0xFFE5	9,645.1	0.5
		57,600	0xFFFB	52,083.3	9.6
		115,200	0xFFFE	130,208.3	13.0
15	7,812,500	9,600	0xFFE7	9,765.6	1.7
		57,600	0xFFFC	61,035.2	6.0
		115,200	0xFFFE	122,070.3	6.0

[1] CLK_DIV refers to the value written into the CHIP::SYSTEM::ICPU_CTRL::CLK_DIV field.

The following code sample exemplifies a serial port initialization with the iCPU running 7.8125 MHz and a desired baud rate of 9600 bps. The serial port is configured to use Timer 2, and looking up the value to write to RCAP2x yields 0xFFE7.

```
  MOV RCAP2H, #0xFF ; MSByte of baud-rate divisor
MOV RCAP2L, #0xE7 ; LSByte of baud-rate divisor
MOV T2CON,  #0x34 ; Use Timer 2 as baud-rate generator by activating both
  ; the RCLK, TCLK, and the Timer 2 itself
MOV SCON,   #0x52 ; Set-up serial port to Mode 1, REN=1 (receive enable)
MOV PS,     #0x1  ; Set high priority for Serial Port interrupts (bit in IP SFR)
MOV ES,     #0x1  ; Enable serial port interrupt (bit in IE SFR)
MOV TH2,    #0xFF ; Force a new value into the counter, so that the baud-rate
MOV TL2,    #0xFF ; generation starts ASAP
```

The example also enables interrupt generation—both transmit and receive interrupts. An interrupt handler must be installed to take care of this.

General Purpose I/O

Four pins on Stansted are assigned to GPIO (ICPU_GPIO[3:0]) for the iCPU. It is important not to confuse these pins with the device's GPIO pins, which are named GPIO[4:0].

The iCPU GPIO pins (in the following simply named "GPIO") are controlled from four SFRs, GPIO_IN, GPIO_OUT, GPIO_OE, and GPIO_STAT. The first three of these registers are located on SFR addresses divisible by 8, which make them bit addressable.

After reset, all pins are configured as inputs. The value currently found on an input pin can be read from the GPIO_IN SFR. The pin directions can be changed individually via the GPIO_OE, Output Enable, register. By setting a bit in this register, the corresponding GPIO pin gets driven by the chip with the value found in the GPIO_OUT register.

The last of the four registers, GPIO_STAT, is used to track state changes in input values. If an input transitions from low to high or from high to low, the corresponding bit will be set in GPIO_STAT. Software must write a 1 to the bit to clear it. When a pin is configured as output, the corresponding bit in GPIO_IN takes the value of the output. Hence, changes in outputs also affect the value of the corresponding bits in GPIO_STAT.

ICPU_GPIO[0] is connected to the iCPUs external interrupt #1. If the input transitions from low to high, external interrupt #1 is generated (in edge-sensitive mode (TCON::IT1=1). This feature can be used in hardware-supported single-stepping (see "Debugging Features," which begins on page 66).

Dual Data Pointers

The iCPU employs dual data pointers (SFRs) to accelerate data memory block moves. The standard 8051 data pointer SFR (DPTR) is a 16-bit value used to address external data RAM or peripherals. The iCPU maintains the standard data pointer as SFR::DPTR0. It is not necessary to modify existing code to use DPTR0.

The iCPU adds a second data pointer (SFR::DPTR1). The SEL bit in the DPTR Select register, DPS::SEL, selects the active pointer. When DPS::SEL is 0, instructions that use the DPTR will use DPL0 and DPH0. When DPS::SEL is 1, instructions that use the DPTR will use DPL1 and DPH1.

All DPTR-related instructions use the currently selected data pointer. To switch the active pointer, toggle the DPS::SEL bit. The fastest way to do this is to use the increment instruction (INC DPS). This requires only one instruction to switch from a source address to a destination address; saving application code from having to save source and destination addresses when doing a block move. It is possible to use the INC instruction without side effects, because the remaining bits of the DPS SFR are unused. Using dual data pointers provides significantly increased efficiency when moving large blocks of data.

Mailbox

Synchronizing tasks between a possible external CPU attached to the device's Serial Interface and the iCPU can be a problematic job without hardware support. The mailbox provides functionality to assist in overcoming these problems. It consists of three 32–bit chip registers that are accessible from both the on-chip 8051 and an external CPU.

The CHIP::SYSTEM::ICPU_MBOX_VAL is a read-only register that returns the current value of the mailbox. Its value is changed by the use of the SYSTEM::ICPU_MBOX_SET and SYSTEM::ICPU_MBOX_CLR event

registers. All bits that are set in the value written to ICPU_MBOX_SET will also become set in the ICPU_MBOX_VAL. Bits that are 0 in the value written to ICPU_MBOX_SET will not affect the bits in ICPU_MBOX_VAL. Likewise for the ICPU_MBOX_CLR register: All bits set in the value written to ICPU_MBOX_CLR register will cause the corresponding bits to become cleared in the ICPU_MBOX_VAL register. Bits that are 0 in the value written to ICPU_MBOX_CLR will not affect the corresponding bits of ICPU_MBOX_VAL.

In order to use the mailbox, it is suggested that the 32 bits be divided into two or more segments, where a segment is logically assigned to each of the CPUs. One CPU can then set bits in one segment, which are conveyed to the other CPU, which in turn clears the bits in the segment when the event is handled, effectively telling the first CPU that the event is handled—and vice versa.

Numbers can also be transferred via the mailbox by first writing the integer as is to the ICPU_MBOX_SET register, then bit-wise negating the integer and writing this value to the ICPU_MBOX_CLR register.

Another chip-register, CHIP::SYSTEM::HWSEM, works as a semaphore with which it is possible to perform mutual exclusions in software tasks. To obtain the semaphore, read the register and monitor the returned value. If the register returns a 1, the reader is now the owner of the semaphore. If it is 0, the semaphore is already taken by another task. To release the semaphore, write a 1 to the register.

Debugging Features

It is often convenient to have some means of debugging the code running within the iCPU. This section describes a solution that uses a combination of software and hardware for debugging the code within the iCPU.

Single-Stepping

The iCPUs interrupt structure allows for single-stepping programs. When exiting an interrupt service routine (ISR) with an RETI instruction, the iCPU will always execute at least one instruction of the task program. Therefore, once an ISR is entered, it cannot be re-entered until at least one program instruction is executed. To perform single-step execution, program the GPIO interrupt (external interrupt #1) to be level sensitive and write an ISR for this interrupt that terminates as follows:

```
wait1:JB  TCON.3, wait1
            ;   Wait for inactive level (high) on external interrupt #1.
            ;   This corresponds to ICPU_GPIO[0]=0.
wait2:JNB TCON.3, wait2
            ;   Wait for active level (low) on external interrupt #1.
            ;   This corresponds to ICPU_GPIO[0]=1.
  IRET      ;   Return from interrupt.
```

To enter the ISR, the ICPU_GPIO[0] pin (must be configured as input) must be set high. The ISR then performs whatever task it needs to (toggles LEDs or the like to indicate to the user that the ISR is entered), and finishes the ISR by first waiting for a 0 then for a 1 on the GPIO input. As soon as the IRET instruction and one more instruction have been executed, the ISR is re-entered, effectively causing just one user-instruction to be executed.

If the ICPU_GPIO[0] pin is connected to a glitch-free switch or button, the user can single-step the program using this switch.

External Access to On-Chip RAMs

The single-stepping procedure outlined in the previous section can be combined with an external CPU reading out the contents of the on-chip RAMs in each step. The external CPU, attached to Stansted's Serial Interface, can gain complete control over both the 8 kilobytes of on-chip RAM and the Scratchpad RAM via two chip-registers, SYSTEM::ICPU_ADDR and SYSTEM::ICPU_DATA.

Continuing the example from "Debugging Features," which begins on page 66, an external CPU may now take over the on-chip RAMs while the ISR is waiting for the ICPU_GPIO[0] to become active again by performing the following steps:

1. First stop the iCPUs clock by writing a 0 to CHIP::SYSTEM::ICPU_CTRL::CLK_EN.
2. Enable external access to the on-chip RAMs by writing a 1 to CHIP::SYSTEM::ICPU_CTRL::EXT_ACC_EN.

The effect of the above is that the on-chip CPU is temporarily halted and the external CPU has both read and write access to both on-chip RAMs.

To read out data from the 8 kilobytes of on-chip RAM, write the start-address (0-based) to SYSTEM::ICPU_ADDR::ADDR while keeping the SYSTEM::ICPU_ADDR::SP_SELECT cleared. Then read the contents from the RAM by successively reading the SYSTEM::ICPU_DATA register. Each read returns 1 byte of data from the RAM and auto-increments the address register by 1. The 256 bytes of on-chip Scratchpad RAM can be read by setting the SYSTEM::ICPU_ADDR::SP_SELECT bit to 1 while specifying the start-address.

Loading new data into the RAMs happens in much the same way as reading data from the RAMs, but instead of reading the SYSTEM::ICPU_DATA register, software must write to it 1 byte at a time. As for reads, writes also cause the address to auto-increment by 1.

When the external CPU has completed the reading/writing of the on-chip RAMs, it will allow the iCPU to continue operation by performing the two writes to ICPU_CTRL mentioned above in reverse order, and this time writing 0 to the EXT_ACC_EN and 1 to the CLK_EN field.

MII Management Bus

An MII Management controller exists for accessing PHY registers (up to 32 PHYs).

The controller is programmed with target PHY address, register index, and write data. A status register indicates when the operation is complete.

The operation is set up in the MIIMCMD register, status is read from the MIIMSTAT register, and the read result can be found in the MIIMDATA register.

Scan Operation

The controller also has the ability to scan a range of PHYs, mainly for automatic polling of link status from the PHYs' status registers. The scan feature is programmed with:

- The range of PHY addresses to be read
- The index of the register to be read from each PHY
- A value mask (PhyRegMask) used to mask out the bits that are to be checked

Stansted continuously reads the specified register bits in the PHYs within the specified PHY range. When the register value is available, a scan result register (MIIMSRES) has the bit position corresponding to the PHY address read updated. This bit value will be set to 1 if all the bits set in the valuemask is also set in the read reply.

To handle multiple changes in the response from a PHY between two CPU reads of the MIIMSRES register, only the first change in each PHYs scan result is saved, until the CPU has read the result register.

Example:

How to continuously update the Link Status for PHY numbers 10-17.

- Set MIIMSCAN to PhyAddrLow=10, PhyAddrHigh=17, PhyRegMask=0x0004.
- Set MIIMCMD to SCAN, READ, PhyAddr=10, PhyReg=1.

Bit 10-17 in the MIIMSRES register now reflects the link status of PHYs 10-17.

In the above example, it is presumed that the link status of the PHYs is found in register 1, bit 2 (common position in most PHYs).

Spanning Tree Protocol

This section describes how the various Spanning Tree Protocol states are supported in Stansted. The states include disabled, blocking, listening, learning, and forwarding. According to the IEEE802.1D standard, the Spanning Tree Protocol states have the properties listed in Table 15.

Table 15. Spanning Tree Protocol Port State Properties.

State	BPDU reception	BPDU generation	Frame forwarding	SMAC learning
Disabled	No	No	No	No
Blocking	Yes	No	No	No
Listening	Yes	Yes	No	No
Learning	Yes	Yes	No	Yes
Forwarding	Yes	Yes	Yes	Yes

These states can all be configured through the CATCONF, SRCMASKS, and LERNMASK registers. Table 16 lists how a port ‘portId’ is configured to the different states.

Table 16. Spanning Tree Protocol Port State Configuration of Stansted

State	’BPDU Capture’ in CATCONF, port ‘portId’	SRCMASKS, mask ‘portId’	LERNMASK, bit ‘portId’
Disabled	0	0	0
Blocking	1	0	0
Listening	1	0	0
Learning	1	0	1
Forwarding	1	1 except for bit ‘portId’	1

In terms of configuration, there is no difference between the Blocking and the Listening state. It is simply up to the CPU to throw away BPDU in the Blocking state.

Multiple Spanning Tree Protocol

Stansted supports the Multiple Spanning Tree Protocol except when a port is in the learning state for a Spanning Tree instance. The switch will not be learning on that port for the given Spanning Tree instance. This, however, has limited impact since when the port is taken to the forwarding state, learning is done at wire-speed and the SMAC learn delay is thus less important. Multiple Spanning Trees are supported for all VLANs.

In order to enable the various port states the switch must be VLAN aware. All VLAN masks written to the VLAN table must have the ‘VLAN Source Check’ bit in VLANCES set Also, the switch must have the ‘VLAN Check’ bit in ADVLEARN set. Furthermore, a port must be configured as described for the Forwarding State in Table 16. Port states per VLAN are hereafter solely configured through the VLAN masks.

Table 17 lists special settings in order to enable the different port states in Table 15 per VLAN.

Table 17. Multiple Spanning Tree Protocol Port State Configuration of Stansted

State per VLAN	Action
Disabled	Port removed from VLAN mask for the particular VLAN
Blocking	Port removed from VLAN mask for the particular VLAN
Listening	Port removed from VLAN mask for the particular VLAN
Learning	Port removed from VLAN mask for the particular VLAN
Forwarding	Port added to VLAN mask for the particular VLAN

Configuration

Command Interfaces

Through the iCPU and/or an external CPU attached to the SI or PI interface, it is possible to configure, monitor, and collect statistics from different parts of the switching engine. This set of chip registers and functionality is independent of the chosen CPU interface. In addition, the iCPU, if enabled, has access to a different set of registers known as Special Function Registers (SFRs).

The PI interface and iCPU are mutually exclusive and are enabled through pin strapping of ICPU_PI_En.

Chip Register Space

The Stansted device consists of several functional blocks. Some of these blocks (the port and the MII Management System) are further divided into subblocks. All subblocks within a given block are clones with equal sets of registers.

Within each subblock an 8-bit address space exists, where each address accesses 32 bits of register data.

The blocks are identified with block IDs as listed in Table 18.

Table 18. Functional Blocks

Block ID	Subblocks	Slow	Functional Block	Short Name
1	0-11	Yes	Ports	PORT
2	0	Yes	Analyzer	ANALYZER
3	0	Yes	MII Management	MIIM
3	1	Yes	Memory Initialization	MEMINIT
4	0-3, 4, 7	No	CPU Capture Buffer	CAPTURE
5	0	Yes	Frame Arbiter	ARBITER
7	0	No	System Registers	SYSTEM

The Slow column is used only when the PI interface is enabled, and refers to a property of the block that will be explained in "Reading Slow Registers," which begins on page 78.

For the ports, the subblock number is the port number (0 through 11). The subblock number also applies to the CPU capture buffer, but not as a selector between submodules, rather as an extension to the 8-bit address space. Refer to "CPU Packet Reception," which begins on page 44.

The entire set of registers is listed in "Register Overview," which begins on page 82.

SFR Register Space

In addition to the chip register space, the iCPU has access to its Special Function Registers register space. Table 19 shows the SFRs implemented in the iCPU. Registers and fields marked with an asterisk (*) are not part of a standard 8051 implementation, and are detailed in subsequent sections. Refer to a standard 8051 data sheet for a description of the remaining registers and fields

Table 19. SFR Register Overview

Register	Description	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default Value
GPIO_IN*	GPIO input	0x80	0	0	0	0	GI3	GI2	GI1	GI0	0x00
SP	Stack pointer	0x81									0x07
DPL0	Data pointer 0 Lo	0x82									0x00
DPH0	Data pointer 0 Hi	0x83									0x00
DPL1*	Data pointer 1 Lo	0x84									0x00
DPH1*	Data pointer 1 Hi	0x85									0x00
DPS*	Data pointer select	0x86	0	0	0	0	0	0	0	SEL	0x00
PCON	Power control	0x87	SMOD0	0	1	1	GF1	GF0	STOP	IDLE	0x30
TCON	Timer 0 & 1 control	0x88	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0x00
TMOD	Timer mode	0x89	GATE	C/T	M1	M0	GATE	C/T	M1	M0	0x00
TL0	Timer Lo 0	0x8A									0x00
TL1	Timer Lo 1	0x8B									0x00
TH0	Timer Hi 0	0x8C									0x00
TH1	Timer Hi 1	0x8D									0x00
CKCON*	Clock control	0x8E	0	0	T2M	T1M	T0M	MD2	MD1	MD0	0x01
SPC_FNC*	Special function	0x8F	0	0	0	0	0	0	0	WRS	0x00
GPIO_OUT*	GPIO output	0x90	0	0	0	0	GO3	GO2	GO1	GO0	0x00
SCON	Serial control	0x98	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	0x00
SBUF	Serial data buffer	0x99									0x00
GPIO_OE*	GPIO output enable	0xA0	0	0	0	0	GOE3	GOE2	GOE1	GOE0	0x00
GPIO_STAT*	GPIO status	0xA1	0	0	0	0	GS3	GS2	GS1	GS0	0x00
WDDA*	Watchdog data	0xA2									0xEF
WDCON*	Watchdog control	0xA3	0	0	0	0	0	0	0	WDEN	0x00
IE	Interrupt enable	0xA8	EA	0	ET2*	ES	ET1	EX1	ET0	EX0	0x00
PAGE*	PagerRegister	0xB0	IFP3	IFP2	IFP1	IFP0	OP3	OP2	OP1	OP0	0x00

Table 19. SFR Register Overview *(continued)*

Register	Description	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default Value
IP	Interrupt priority	0xB8	1	0	PT2*	PS	PT1	PX1	PT0	PX0	0x80
T2CON*	Timer 2 control	0xC8	TF2	0	RCLK	TCLK	0	TR2	0	0	0x00
RCAP2L*	Timer 2 capture Lo	0xCA									0x00
RCAP2H*	Timer 2 capture Hi	0xCB									0x00
TL2*	Timer Lo 2	0xCC									0x00
TH2*	Timer Hi 2	0xCD									0x00
PSW	Programming status word	0xD0	CY	AC	F0	RS1	RS0	OV	F1*	P	0x00
ACC	Accumulator	0xE0									0x00
B	B register	0xF0									0x00
RA_DONE*	Register accumulator done	0xF8	0	0	0	0	0	0	0	DONE	0x00
RA_BLK*	Register accumulator block	0xF9	BL2	BL1	BL0	0	SBL3	SBL2	SBL1	SBL0	0x00
RA_AD_RD*	Register accumulator read address	0xFA									0x00
RA_AD_WR*	Register accumulator write address	0xFB									0x00
RA_DA0*	Register accumulator data 0	0xFC									0x00
RA_DA1*	Register accumulator data 1	0xFD									0x00
RA_DA2*	Register accumulator data 2	0xFE									0x00
RA_DA3*	Register accumulator data 3	0xFF									0x00

* These registers and fields are not part of a standard 8051 implementation, and are detailed in subsequent sections.

The Serial Interface

The Serial Interface (SI) is a simple interface that operates as described in the “Serial Peripheral Interface” or “SPI” specification by Motorola (not to be confused with “SPI4” which is a high speed interface), running in mode CPHA=0 and CPOL=0. It consists of four signal lines: an input clock (SI_Clk), an enable signal (SI_nEn), and a data signal in each direction (SI_DI and SI_DO).

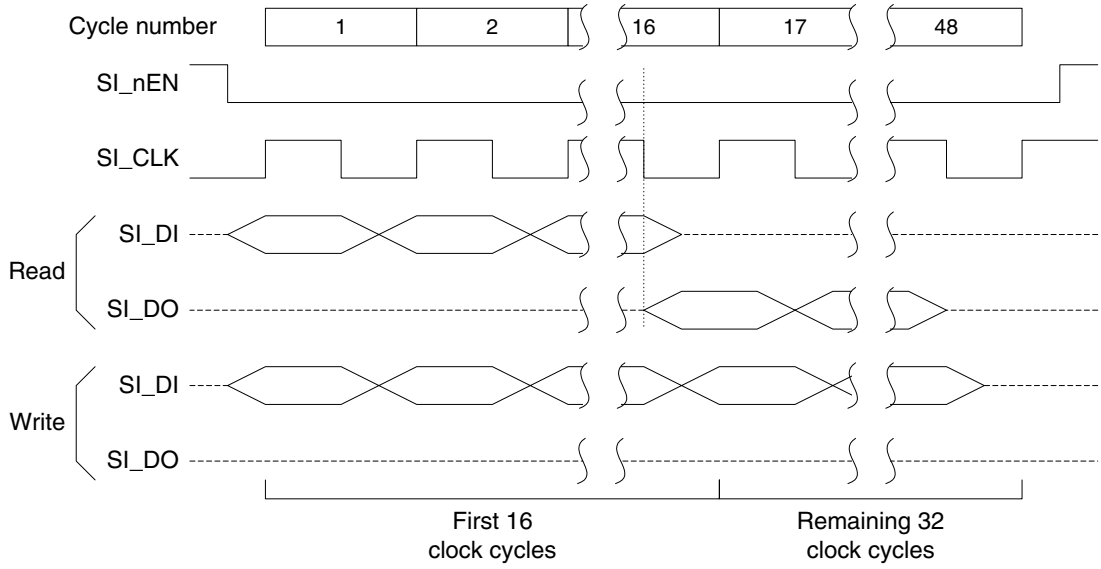


Figure 14. Serial Interface Communication Sequence

An external clock running at a speed between 0 and 25 MHz clocks the interface. SI_nEn low starts an operation consisting of 48 clock cycles: 8 for command/block/subblock, 8 for address, and 32 for data. During a read operation, SI_DO is driven; otherwise it is tri-stated.

Table 20. Block Address Structure

Byte #	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Block ID			Write	Subblock Number			
1	Address							
2	Data							
3	Data							
4	Data							
5	Data							

SI Clock Select

The internal timing of the Stansted requries that the SI user must ensure at least a 400 ns delay from when the last bit of a read address is transmitted to when SI_Clk goes low again. This can be done in three ways:

- Let the SI clock run at about 1.25 MHz or lower (see Figure 15 and Figure 16)
- Let the SI clock pause (at HIGH) after the last address is clocked out (see Figure 17)
- Ask the device to insert idle byte periods before the read data (see Figure 18)

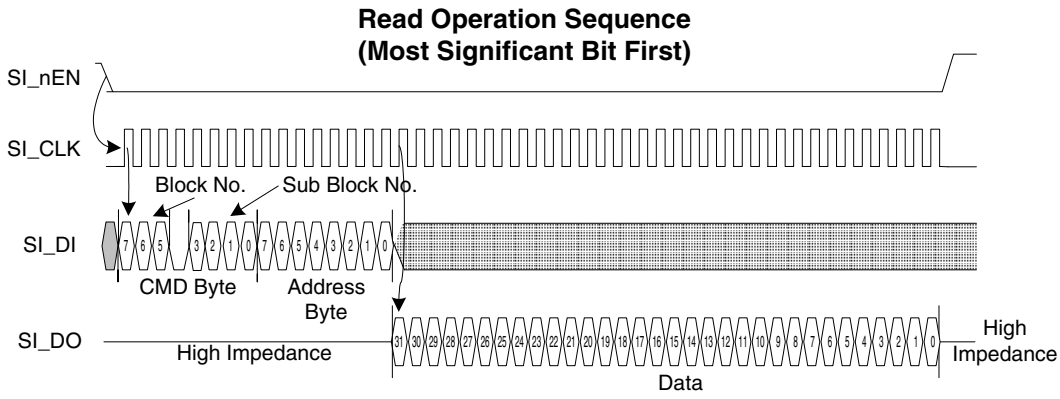


Figure 15. SI Read Operation Sequence (Low Clock)

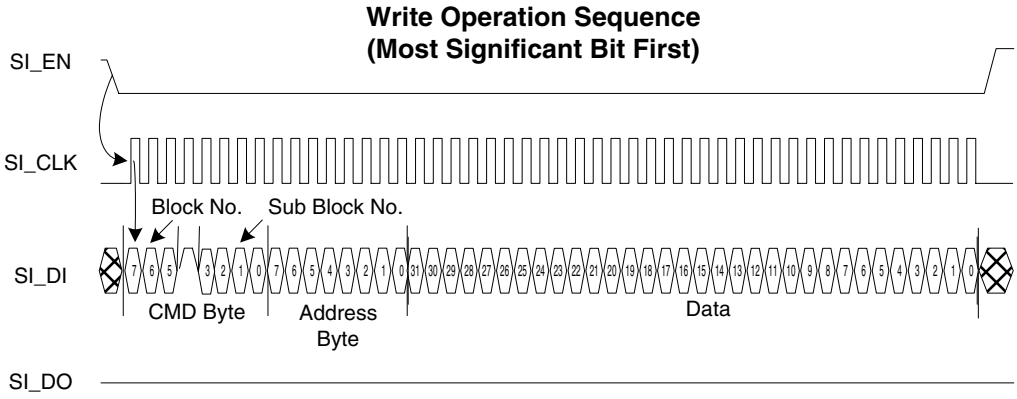


Figure 16. SI Write Operation Sequence (Low Clock)

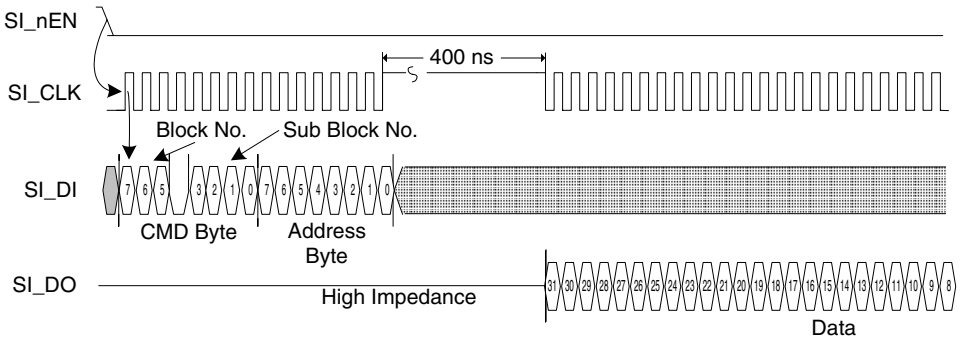


Figure 17. SI Read Operation with Clock Pause

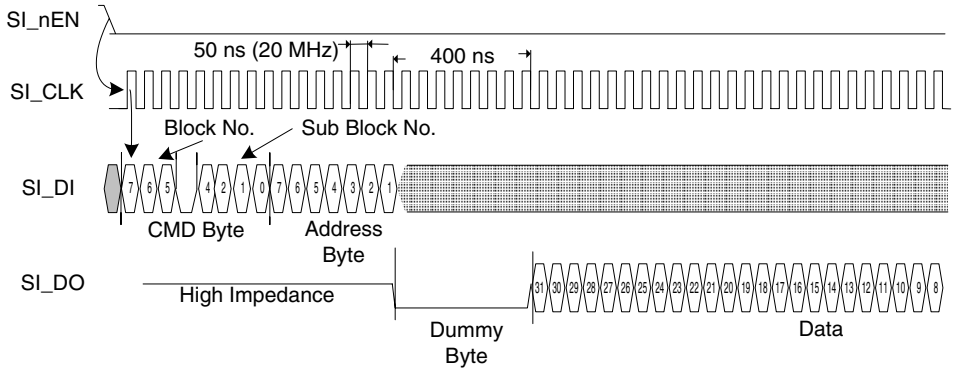


Figure 18. SI Read Operation with Dummy Byte

As an example of the third option, assume the SI clock is running at 20 MHz. This gives a clock cycle of 50 ns, so to ensure a 400 ns delay before data is read out, the Stansted must add padding bytes (PB) so that (8 x PB + 0.5) x ClkCycle ≥ 400 ns. Hence, to run a 20 MHz SI, 1 byte of extra dummy data must be inserted in the read protocol sequence between command/address and data. Table 21 lists the clock limits for each configuration of dummy bytes.

Table 21. Number Of Dummy Bytes and Maximum Serial Interface Clock

Maximum SI Clock	Number of Dummy Bytes
1.25 MHz	0
21.25 MHz	1
25 MHz	2

Write operations do not contain any padding bytes, no matter what padding configuration has been chosen. Thus, no special care need be taken during the initial configuration of the Stansted for pad insertion; this can be done at full speed.

SI Configuration

The SI protocol sequence is programmed using system registers. There are three possible configurations possible.

- Select the byte order for the interface when sending 32 bit data words (endian selection)
- Select whether to transfer the most or the least significant bit first within each byte
- Select insertion of idle bytes before read data is transmitted

Table 22. SI Configuration Register

System Register	Description
CPUMODE	Selects endian mode and bit transfer order
SIPAD	Selects number of padding bytes

The Parallel Interface

When enabled (ICPU_PI_En pin strapped low), the Parallel Interface's access to the registers is superior in terms of speed compared to the Serial Interface. In the default 16-bit mode, it consists of 16 address lines, 16 bidirectional data lines, and 5 control signals. In the 8-bit mode, it consists of 16 address lines, 8 bidirectional data lines, and 5 control signals. The 8-bit mode of operation is described in "8-bit Data Bus Width," which begins on page 79.

Table 23. PI Interface Signals

PI Signal	Direction	Width	Description
PI_Addr	I	16	Selects the block, sub_block and address
PI_Data	I/O	8/16	Data - driven by CPU at write, by the Stansted at read
PI_nWR	I	1	Selects read (1) or write (0)
PI_nOE	I	1	Enables driving of the data bus from the device
PI_nCS	I	1	Start a CPU operation
PI_nDone	OZ	1	Acknowledges an operation
PI_IRQ	O	1	A configurable interrupt signal for various events

The operations of the interface can be seen in Figure 19. To use the interface, do the following:

1. Lower nCS signal.
2. Within 120 ns, set up address, R/W select and data in case of write operation.
3. Wait until at least 140 ns from when nCS was lowered, or wait for nDone to fall.
4. Store data from data bus in case of read operation.
5. Raise nCS signal.

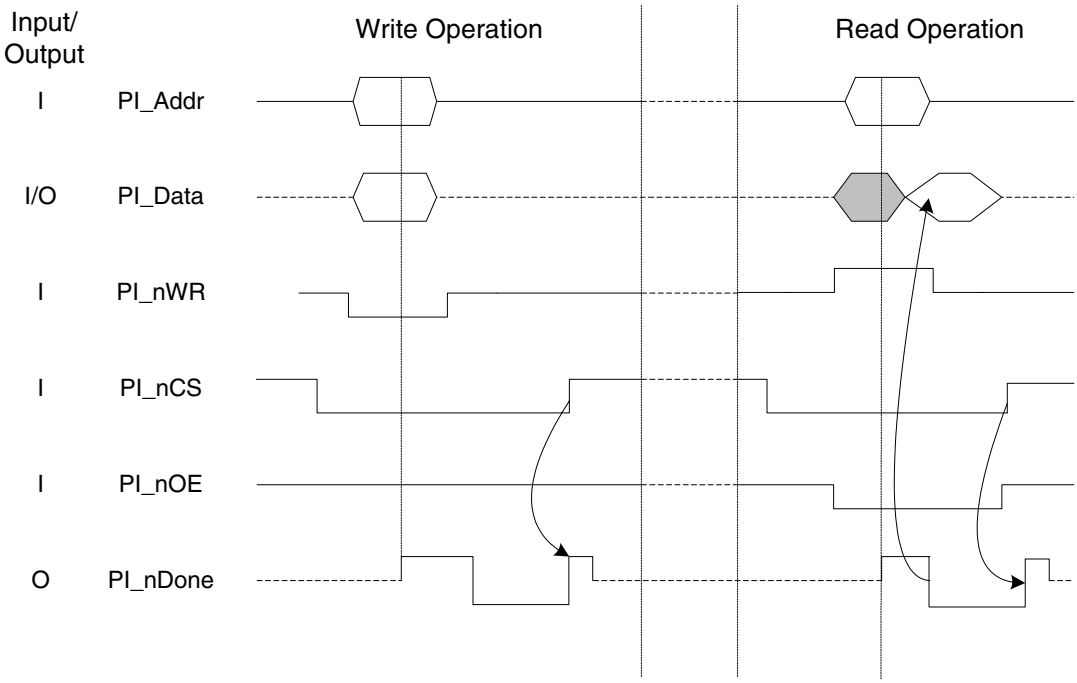


Figure 19. Parallel Interface Communication Signals

All inputs are latched a configurable amount of time after nCS falling. The delay is set in the CPUCTRL register and is configurable from 10 to 130 ns with the default being 130 ns. nDone acknowledges the operation around 10 ns later. For precise timing information, refer to the AC specifications for the PI interface.

The nOE signal along with nCS directly control the output driver of the data bus.

As the registers internally in the Stansted are 32–bit organized, every register access must be performed in two consecutive CPU cycles. In a write operation, the Stansted does not update any target inside the chip until both 16-bit half words have been written, and in a read operation, the device only accesses the internal 32–bit source when the first half word of a register is read.

Reading Slow Registers

Most of the registers in the Stansted device have a longer access time than acceptable for a CPU bus cycle. The access time can be up to 400 ns, and therefore special action must be taken to read these.

Using the SLOWDATA Postponed Result

A way to cope with the slow read result is to make a dummy read of one of the half words from the target register, and then read the result from the SLOWDATA register. This feature must be enabled through the CPUCTRL register.

Example:

1. Read one of the half words of the register.
2. Poll the SlowDone bit of CPUCTRL, or wait for SlowDone interrupt.
3. Read one of the half words of the SLOWDATA register.
4. Read the other half word of the SLOWDATA register.

Extended Bus Cycle

It is also possible to extend the bus cycle, thereby postponing the nDone handshake until slow data is available. All registers are in this mode read like fast registers, but the nDone signal will not be activated until up to 400 ns after falling nCS. The extended bus cycle is default behavior, and can be disabled through the CPUCTRL register. Only the first half word of a slow register will have a long access time.

Interrupt Control

The PI_IRQ pin of the Stansted device can be used to signal when a CPU should take action on two different events: When the SLOWDATA register has been filled, or when a packet is ready in the CPU packet receive buffer. The interrupt signal is of level type and is cleared when the source condition inside the device is removed by reading the SLOWDATA register or by acknowledging a CPU packet.

Register Addressing

The device registers are mapped into the 16-bit address space of the PI as shown in Table 24.

Table 24. Register Mapping in PI Address Space

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Block ID			Subblock Number				Register address								WS

The WS bit selects the most or least significant 16 bits of the 32–bit word. In big endian mode, WS=1 selects the least significant. In little endian mode, WS=1 selects the most significant. For proper use, the first reading/writing of the 32–bit word must have WS=0, and the second reading/writing must have WS=1.

8-bit Data Bus Width

The parallel interface also supports CPUs equipped with only an 8-bit interface. This is configured through the PIWIDTH register, which has the address 0xe002. In this mode, all register accesses are split into four byte accesses. The byte select is not added to the address bus, so in effect a register is read through multiple accesses to the same address, and an 8-bit CPU should thus have its address lines 1..16 attached to the device.

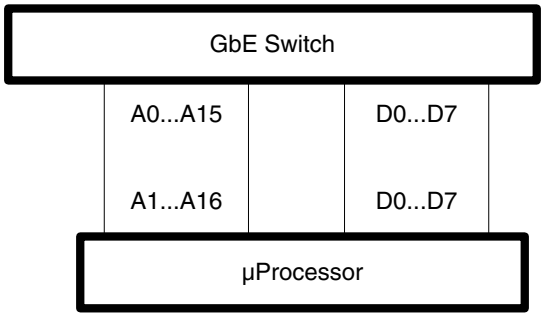


Figure 20. 8-bit Data Bus Width

As an example, system register 0 is read (Big endian mode) as shown in Table 25.

Table 25. PI Bus 8-bit Data Width Example

Access No.	Address	CPU Bus address	Data
0	0xe000	0x1C000	Most significant byte
1	0xe000	0x1C001	Second most significant byte
2	0xe001	0x1C002	Second least significant byte
3	0xe001	0x1C003	Least significant byte

Minimum Software Requirements

The Stansted device is almost fully working with its chip default configuration, and the minimum set of registers to setup for basic operation is listed in Table 26. For using statistics, aggregation, VLAN, flow control, MAC aging, and so forth, refer to "Register Overview," which begins on page 82 and the software API.

Table 26. Minimum Register Set to Set Up

Register Name	Use For
MEMINIT	Initializing memories
MEMRES	Reading result from memory initialization
MACACCESS	Formatting MAC table
VLANACCESS	Formatting VLAN table
MACCONF	Setting the port mode according to PHY status

Table 26. Minimum Register Set to Set Up *(continued)*

Register Name	Use For
RECVMASK	Enabling frame forwarding from ports
MIIMCMD	Performing a MII management operation
MIIMDATA	Reading PHY read results from MII management operation

The mandatory operation for the software is split into an initialization sequence and a PHY polling operation. If the PHYs are running with fixed speed and duplex mode, the poll operation could be ignored, and the device ports set up to the desired mode as a part of the initialization sequence.

The very simple routines below are only for showing the minimum requirements, and cannot be used as the basis for a final software system. The examples are shown here to illustrate the use of the chip registers. The register accesses are shown in a block.subblock.address format.

Initialization Sequence

The following initialization sequence is REQUIRED to ensure proper operation of the switch. The iCPU may use the scratchpad RAM, but not the 8 kB on-chip RAM, while the following code-snippet is executed.

```
/* Initialize memories */
for (memId=0; memId<=19; memId++){
  if (memId!=6) {
     /* Initiate initialization except for the iCPU scratchpad RAM */
     Write(MEMINIT.1.MEMINIT, 0x1010400 + memId);
     Sleep 1 ms;
  }
}
Sleep 30 ms;
/* Read result from memory initialization */
for (memId=0; memId<=19; memId++){
  if (memId!=6) {
     /* Initiate reading except for the iCPU scratchpad RAM */
     Write(MEMINIT.1.MEMINIT, 0x20000 + memId);
     Sleep 1 ms;
     /* Read the result */
     result=Read(MEMINIT.1.MEMRES);
     if ((result&0x3)!=0x3){
        /* Memory initialization failed for Memory memId */
     }
```

```
  }
}
/* Format memories */
Write(ANALYZER.0.MACACCESS, 5); /* CLEAR MAC TABLE command */
Write(ANALYZER.0.VLANACCESS, 3); /* CLEAR VLAN TABLE command */
Sleep 40 ms;
```

Port Mode Procedure

Execute this procedure on a regular basis for updating the port mode according to the PHY state. This example illustrates the basic requirements only; it must be expanded for software control program under normal conditions in most applications. Refer to the API documentation of the device for a more fully detailed description of the procedure.

```
for (portId=0; portId<12; portId++) {
  /*Use MIIMCMD and MIIMDATA to retrieve current speed
     and link status from PHY attached to port <portId>
     Put the status into the <mode>
  */

  /* enable forwarding from a port, only if the port link is up */
  If (mode==DOWN) then
     Write(ANALYZER.0.RECVMASK, Read(ANALYZER.0.RECVMASK) & ~(1<<portId));
  else
     Write(ANALYZER.0.RECVMASK, Read(ANALYZER.0.RECVMASK) | (1<<portId));

  /* Set the port MAC to the current mode */
  Switch (mode) {
  Case DOWN: Write(PORT.portId.MACCONF, 0x20000030); break;
  Case 1GFULL: Write(PORT.portId.MACCONF, 0x10070141); break;
  Case 100MFULL: Write(PORT.portId.MACCONF, 0x10050442); break;
  Case 10MFULL: Write(PORT.portId.MACCONF, 0x10050443); break;
  Case 100MHALF: Write(PORT.portId.MACCONF, 0x90010442); break;
  Case 10MHALF: Write(PORT.portId.MACCONF, 0x90010443); break;
  }
}
```

Register Overview

Table 27. System Block Registers (Block 7)

Address	Register Name	Short Name
00h	CPU Transfer Mode	CPUMODE
01h	SI Padding	SIPAD
02h	PI BusWidth	PIWIDTH
10h	Internal CPU Control	ICPU_CTRL
11h	Internal CPU On-chip RAM address	ICPU_ADDR
12h	Internal CPU On-chip RAM data	ICPU_DATA
13h	Semaphore Register	HWSEM
14h	Global Reset	GLORESET
15h	Mailbox Value	ICPU_MBOX_VAL
16h	Mailbox Set	ICPU_MBOX_SET
17h	Mailbox Clear	ICPU_MBOX_CLR
18h	Chip Identification	CHIPID
19h	External Data Memory Access Control Configuration	ICPU_RAM_CFG
1Ah	External Program Memory Access Control Configuration	ICPU_ROM_CFG
24h	Time Compare Value	TIMECMP
2Ch	SlowData	SLOWDATA
30h	CPU Control	CPUCTRL
34h	General Purpose IO	GPIO

Table 28. MAC Block Registers (Block 1)

Address	Register Name	Short Name
00h	MAC Config	MACCONF
02h	Half Duplex Gaps	MACHDXGAP
04h	Flow Control Setup	FCCONF
08h	Flow Control SMAC High	FCMACHI
0Ch	Flow Control SMAC Low	FCMACLO
10h	Max Length	MAXLEN
11h	Shaper Setup	SHAPECONF
12h	Policer Setup	POLICECONF
13h	Multicast Storm Setup	MCSTORMCONF

Table 28. MAC Block Registers (Block 1) *(continued)*

Address	Register Name	Short Name
19h	Advanced Port Mode Setup	ADVPORTM
24h	Transmit Modify Setup	TXUPDCFG
25h	CFI Drop Counter	CFIDROP

Table 29. Shared FIFO Block Registers (Block 1)

Address	Register Name	Short Name
C0h	CPU Transmit DATA	CPUTXDAT
C4h	MISC Control Register	MISCFIFO
CCh	Pool Control Register	POOLCFG
DCh	Drop Control Register	DROPCFG
C8h	Misc Status	MISCSTAT
D8h	Free RAM Counter	FREEPOOL

Table 30. Categorizer Block Registers (Block 1)

Address	Register Name	Short Name
60h	Categorizer Config	CATCONF
64h	Categorizer Map Tag	CATTAG
65h	Categorizer SUBMap Tag	CATSUBTAG
68h	EtherType Register	CATETHT
6Ch	DSAP Register	CATDSAP
70h	IP Protocol Register	CATIPPRT
74h	Categorizer Priorities	CATPRIO
75h	Categorizer SubPriorities	CATSUBPRIO
78h	DS Mapping Register High	CATDSMAPH
79h	DS Mapping Register Low	CATDSMAPL
7Ah	DS SubMapping Register High	CATDSSUBMAPH
7Bh	DS SubMapping Register Low	CATDSSUBMAPL
7Ch	PVID Register	CATPVID
80h	TCP/UDP Port Register 1	CATPORT1
84h	TCP/UDP Port Register 2	CATPORT2
88h	TCP/UDP Port Register 3	CATPORT3
8Ch	TCP/UDP Port Register 4	CATPORT4
90h	TCP/UDP Port Register 2	CATPORT5

Table 31. Statistics Block Registers (Block 1)

Address	Register Name	Short Name
37h	Rx Total Bad Packets	RXBADPKT
39h	Rx Control Packets	RXCTRL
3Dh	Rx Pause Frames	C_RXPAUSE
3Eh	Tx Pause Frames	C_TXPAUSE
3Fh	Rx Classified Drops	C_RXCATDROP
44h	Tx Total Error Packets	TXERR

Table 32. Detailed Counters Block Registers (Block 1)

Address	Register Name	Short Name
50h	Rx Octets	C_RXOCT
51h	Tx Octets	C_TXOCT
A0h	Rx Drops	C_RXDROP
A1h	Rx Packets	C_RXPKT
A2h	Rx Broadcasts	C_RXBC
A3h	Rx Multicasts	C_RXMC
A4h	Rx CRC/ALIGN	C_RXCRC
A5h	Rx Undersize	C_RXSHT
A6h	Rx Oversize	C_RXLONG
A7h	Rx Fragments	C_RXFRAG
A8h	Rx Jabbers	C_RXJAB
A9h	Rx 64 Bytes	C_RX64
AAh	Rx 65-127 Bytes	C_RX65
ABh	Rx 128-255 Bytes	C_RX128
ACh	Rx 256-511 Bytes	C_RX256
ADh	Rx 512-1023 Bytes	C_RX512
AEh	Rx 1024-long Bytes	C_RX1024
AFh	Tx Drops	C_TXDROP
B0h	Tx Packets	C_TXPKT
B1h	Tx Broadcasts	C_TXBC
B2h	Tx Multicasts	C_TXMC
B3h	Tx Collisions	C_TXCOL
B4h	Tx 64 Bytes	C_TX64
B5h	Tx 65-127 Bytes	C_TX65

Table 32. Detailed Counters Block Registers (Block 1) *(continued)*

Address	Register Name	Short Name
B6h	Tx 128-255 Bytes	C_TX128
B7h	Tx 256-511 Bytes	C_TX256
B8h	Tx 512-1023 Bytes	C_TX512
B9h	Tx 1024-long Bytes	C_TX1024
BAh	Tx FIFO Drops	C_TXOVFL
BBh	Rx High Priority Frames	C_RXHP
BCh	Rx Low Priority Frames	C_RXLP
BDh	Tx High Priority Frames	C_TXHP
BEh	Tx Low Priority Frames	C_TXLP

Table 33. MII Management Bus Block Registers (Block 3)

Address	Register Name	Short Name
00h	MII-M Status	MIIMSTAT
01h	MII-M Command	MIIMCMD
02h	MII-M Return Data	MIIMDATA
03h	MII-M Prescaler	MIIMPRES
04h	MII-M Scan setup	MIIMSCAN
05h	MII-M Scan Results	MIIMSRES

Table 34. Memory Initialization Block Registers (Block 3)

Address	Register Name	Short Name
00h	Initialize	MEMINIT
01h	Read Result	MEMRES

Table 35. Frame Arbiter Block Registers (Block 5)

Address	Register Name	Short Name
0Ch	Arbiter Empty	ARBEMPTY
0Eh	Arbiter Discard	ARBDISC

Table 36. CPU Capture Block Registers (Block 4)

Address	Register Name	Short Name
00h	Read Pointer	CAPREADP
03h	Write Pointer	CAPWRP
FFh	Full Reset	CAPRST

Table 37. Frame Analyzer Block Registers (Block 2)

Address	Register Name	Short Name
03h	Advanced Learning Setup	ADVLEARN
04h	IP Multicast Flood Mask	IFLODMSK
05h	VLAN Source Port Mask	VLANMASK
06h	Mac Address High	MACHDATA
07h	Mac Address Low	MACLDATA
08h	Station Move Logger	ANMOVED
09h	Aging Filter	ANAGEFIL
0Ah	Event Sticky Bits	ANEVENTS
0Bh	Event Sticky Mask	ANCNTMSK
0Ch	Event Sticky Counter	ANCNTVAL
0Dh	Learn Mask	LERNMASK
0Eh	Unicast Flood Mask	UFLODMSK
0Fh	Multicast Flood Mask	MFLODMSK
10h	Receive Mask	RECVMASK
20h	Aggregation Mode	AGGRCNTL
30h - 3Fh	Aggregation Masks	AGGRMSKS
40h - 7Fh	Destination Port Masks	DSTMASKS
80h - 8Bh	Source Port Masks	SRCMASKS
B0h	Mac Table Command	MACACCES
C0h	Mac Table Index	MACTINDX
D0h	VLAN Table Command	VLANACES
E0h	VLAN Table Index	VLANTINDX
F0h	Analyzer Config Register	AGENCNTL

Chip Register Details

Unspecified fields in the registers must be written zero and can be ignored on read. The mode column in the register tables shows the access for the register:

R/WRead and write

R/ORead only

W/OWrite only (read dummy)

C/RClear on read (Sticky status bit)

System Block Registers (Block 7)

Table 38. CPU Transfer Mode - CPUMODE (Address 00h)
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
3	Endian	R/W	Configures the byte order mode on SI and PI (if present) interfaces. 0=SI: Least significant byte first, PI: Least significant halfword first (A0=0). 1=SI: Most significant byte first, PI: Most significant halfword first.	1
0	BitDone	R/W	Configures the SI and PI (if present) interfaces as: 0=SI: Least significant bit first in each byte, PI: PI_nDone active high. 1=SI: Most significant bit first in each byte, PI: PI_nDone active low.	1
If the ICPU_PI_En strapping pin is strapped high, the Parallel Interface is not present. The register controls the data transfer mode for the Serial Interface. For the write to work independently of the current transfer mode, BitDone must be mirrored to bits 7, 8, 15, 16, 23, 24, and 31; and endian must be mirrored to bits 4, 11, 12, 19, 20, 27, and 28. In this respect, the possible values for this register are: 0x00000000=little endian, least significant bit first / PI_nDone active high 0x18181818=big endian, least significant bit first / PI_nDone active high 0x81818181=little endian, most significant bit first / PI_nDone active low 0x99999999=big endian, most significant bit first / PI_nDone active low				

Table 39. SI Padding - SIPAD (Address 01h)
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
2:0	SI padding	R/W	Cycle count	0
Number of byte cycles (0-7) during SI read between command and the first byte read. Used to assure 400 ns delay and must be set according to the SI clock frequency.				

Table 40. PI BusWidth - PIWIDTH (Address 02h)
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
31:0	HalfWidth mode	W/O	Write all-ones to this register to swap to 8-bit bus width.	0
Regardless of the current bus-width, a single write to this register will be effective. With an 8-bit attached CPU, a single 8-bit write to this register with value 0xFF will change to 8-bit mode. Default width is 16. When the ICPU_PI_En pin is strapped high, PI is not present, and the value written to this register is ignored.				

Table 41. Internal CPU Control - ICPU_CTRL (Address 10h)
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
31	WATCHDOG_RST	R/W	Set by hardware if the iCPU was reset due to a failure in servicing the watchdog in due time. Write a ‘1’ to clear it.	0
11:8	CLK_DIV	R/W	Selects the iCPU's clock frequency. The resulting frequency is computed as 125 MHz/(CLK_DIV+1). Upon a system reset, the frequency is 125 MHz / 16 or 7.8125 MHz. The chip ignores writes of zero to this field.	Fh
7	SOFT_RST_HOLD	R/W	Normally the SOFT_RST bit in this register is auto-released. By setting SOFT_RST_HOLD together with clearing SOFT_RST, the iCPU is kept reset until this bit is cleared again. When SOFT_RST is '1', the value of this bit has no influence.	0
6	ICPU_PI_En	R/O	Returns the value of the ICPU_PI_En strapping pin. ‘0’: The Parallel Interface is present. ‘1’: The iCPU is active.	0/1
5	DRAM_EN	R/W	Setting this bit causes the on-chip RAM to be present in the Data RAM address space. Clearing it causes accesses to its address space to be forwarded to external memory.	1
4	CRAM_EN	R/W	Setting this bit causes the on-chip RAM to be present in the Code RAM address space. Clearing it causes accesses to its address space to be forwarded to external memory. Program write accesses are always forwarded to the external memory independent of the setting of this bit.	1

Table 41. Internal CPU Control - ICPU_CTRL (Address 10h) *(continued)*
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
3	BOOT_EN	R/W	Setting this bit causes the on-chip RAM to be mapped into address 0x0 in program memory space if (and only if) CRAM_EN is also 1. By default this bit is cleared, effectively making an external Flash mapped into address 0x0 of the code space. Care should be taken not to execute code currently running in the lower 8 kilobytes of the Flash while setting this bit. A sound way to use this bit is in conjunction with the SOFT_RST bit (see below). Setting both bits simultaneously causes the iCPU to reboot from on-chip RAM.	0
2	EXT_ACC_EN	R/W	This field is intended for debugging purposes only. It enables external access to both the on-chip Scratchpad and 8 kilobytes CRAM/DRAM. The on-chip CPU should be stopped or held reset while an external CPU manipulates the contents of the RAM(s). Two registers, ICPU_ADDR and ICPU_DATA provide read/write access to the RAMs once enabled.	0
1	CLK_EN	R/W	When this bit is 1, the iCPU's clock is running, when 0, it is stopped. When debugging the on-chip scratchpad and/or CRAM/DRAM RAMs, the iCPU must be halted using this control bit or held reset using the SOFT_RST control bit. Note: Be sure to have the iCPU's watchdog disabled when disabling the iCPU's clock, since the watchdog is active even when the iCPU is sleeping.	1
0	SOFT_RST	R/W	Write a 0 to this bit in order to soft-reset the iCPU. Upon reset, the iCPU will start executing instructions from the code memory space's address 0x0000. When the iCPU is running, this bit will always return 1. See also the SYSTEM::GLORESET register for a description of how to reset the chip without resetting the iCPU.	1
This register is only used when the ICPU_PI_En strapping pin is tied high.				

Table 42. Internal CPU On-Chip RAM address - ICPU_ADDR (Address 11h)
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
31	SP_SELECT	R/W	Scratchpad RAM Select. If cleared, the RAM accessed using this and the ICPU_DATA register is the 8-kilobyte RAM. If set, it is the 256-byte Scratchpad RAM.	0
12:0	ADDR	R/W	Sets the access address of the on-chip RAMs. Only the eight least significant bits are used when SP_SELECT is 1.	0
This register is intended for debugging purposes only. It is only valid when the ICPU_PI_En strapping pin is tied high.				

Table 43. Internal CPU On-Chip RAM data - ICPU_DATA (Address 12h)
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
7:0	DATA	R/W	Read or write this field to get or set the data held in the selected RAM's address pointed to by the ICPU_ADDR::ADDR field. The address is auto-incremented after the access. The ICPU_CTRL register's EXT_ACC_EN field must be set to 1 prior to accessing this register. If not, reading or writing this register has no effect other than incrementing the ICPU_ADDR register, which wraps around from 0x1FFF to 0x0000 independent of the SP_SELECT value.	0
This register is intended for debugging purposes only. It is only valid when the ICPU_PI_En strapping pin is tied high.				

Table 44. Semaphore Register - HWSEM (Address 13h)
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
0	SEMAPHORE	R/W	This field behaves like a semaphore. Read this field to take the semaphore. If the read returns 1, the reader has the semaphore. If it returns 0, the reader must keep reading the register. When a semaphore owner wants to release the semaphore, he must write a 1 to this field.	0

Table 45. Global Reset - GLORESET (Address 14h)
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
4	STROBE	W/O	Used to enable access to the xxx_LOCK fields. For example, write 0x14 to lock the memory initialization. Perform a soft-reset in two steps: First write to the lock bits to prevent the internal CPU or the memories from being reset, then write to the MASTER_RESET bit to perform the reset.	0
3	ICPU_LOCK	W/O	Used to prevent the internal CPU from being reset when MASTER_RESET is asserted. The value is ignored unless ICPU_PI_En strapping pin is tied high.	0
2	MEM_LOCK	W/O	Used to prevent the internal memory initialization, specified in the “Minimum Software Requirements” section, from being reset when MASTER_RESET is asserted. If this bit is 0 during a MASTER_RESET the memory initialization must be rerun in order to insure proper operation of the switch.	0

Table 45. Global Reset - GLORESET (Address 14h) *(continued)*
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
0	MASTER_RESET	W/O	Write 1 to this register to execute software reset. The reset acts as an external reset except that the protected registers are untouched if their corresponding lock signal is asserted. The reset should be followed by a delay of 125 µs before any access is made to the chip. Do not write a 1 to this bit while also asserting the STROBE.	0

Table 46. Mailbox Value - ICPU_MBOX_VAL (Address 15h)
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
31:0	MBOX_VAL	R/O	Read this register to obtain the current value of the mailbox used for exchanging data/events between the internal and an external CPU. Its value is changed by the use of the ICPU_MBOX_SET and ICPU_MBOX_CLR registers. It is suggested that the 32 bits are split into two or more segments, so that events can be signaled from the internal to the external CPU through one segment and vice versa. For example, the ICPU_MBOX_SET register is used by the ICPU on segment #1 only, and the ICPU_MBOX_CLR is used by the iCPU on segment #2 only, and vice versa. The semaphore register, HWSEM, may be used to protect shared bits, which could be allocated in a third segment.	0
The register is general purpose and may be used without the iCPU being enabled.				

Table 47. Mailbox Set - ICPU_MBOX_SET (Address 16h)
Block 7 Subblock

Bit	Name	Mode	Description	Default
31:0	MBOX_SET	W/O	Write this register to set the corresponding bits in ICPU_MBOX_VAL. Only bits that are 1 get set in the ICPU_MBOX_VAL register; the remaining are untouched.	0
The register is general purpose and may be used without the iCPU being enabled.				

Table 48. Mailbox Clear - ICPU_MBOX_CLR (Address 17h)
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
31:0	MBOX_CLR	W/O	Write this register to clear the corresponding bits in ICPU_MBOX_VAL. Only bits that are 1 get cleared in the ICPU_MBOX_VAL register; the remaining bits are untouched.	0

Table 48. Mailbox Clear - ICPU_MBOX_CLR (Address 17h) *(continued)*
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
The register is general purpose and may be used without the iCPU being enabled.				

Table 49. Chip Identification - CHIPID (Address 18h)
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
31:28	REVISION_NUMBER	R/O	Revision ID: 0000=First revision. 0001=Second revision. ...	0
27:12	PART_NUMBER	R/O	BCD Encoded Part Number for this device.	7384
11:1	MANUFACTURER_ID	R/O	The unique identifier for the chip vendor.	74h
0	RESERVED	R/O	Always read 1.	1
This register returns the same value as the JTAG Identifier.				

Table 50. External Data Memory Access Control Configuration - ICPU_RAM_CFG (Address 19h)
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
9:8	CHIP_SEL_WRITE_ DELAY	R/W	Controls the number of 8 ns cycles the iCPU will delay the RAM chip select signal (ICPU_RAM_nCS) for data writes (ICPU_nWR active).	0
7:6	CHIP_SEL_READ_DELAY	R/W	Controls the number of 8 ns cycles the iCPU will delay the RAM chip select signal (ICPU_RAM_nCS) for data reads (ICPU_nRD active).	1h
5:4	WRITE_DELAY	R/W	Controls the number of 8 ns cycles the iCPU will delay the write enable signal (ICPU_nWR) for data writes.	0
3:2	READ_DELAY	R/W	Controls the number of 8 ns cycles the iCPU will delay the read enable signal (ICPU_nRD) for data reads.	1h
1:0	WRITE_DATA_HOLD	R/W	Controls the number of 8 ns cycles the iCPU will hold written data (ICPU_Data[7:0]) beyond “normal”.	1h
The value written to this register has no effect unless the iCPU is enabled.				

Table 51. External Program Memory Access Control Configuration - ICPU_ROM_CFG (Address 1Ah)
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
9:8	CHIP_SEL_WRITE_ DELAY	R/W	Controls the number of 8 ns cycles the iCPU will delay the chip select signal (ICPU_ROM_nCS) for instruction writes (ICPU_nWR active).	0
7:6	CHIP_SEL_READ_DELAY	R/W	Controls the number of 8 ns cycles the iCPU will delay the chip select signal (ICPU_ROM_nCS) for instruction reads (ICPU_nRD active).	1h
5:4	WRITE_DELAY	R/W	Controls the number of 8 ns cycles the iCPU will delay the write enable signal (ICPU_nWR) for instruction writes.	0
3:2	READ_DELAY	R/W	Controls the number of 8 ns cycles the iCPU will delay the read enable signal (ICPU_nRD) for instruction reads.	1h
1:0	WRITE_DATA_HOLD	R/W	Controls the number of 8 ns cycles the iCPU will hold written instruction data (ICPU_Data[7:0]) beyond “normal”.	1h
The value written to this register has no effect unless the iCPU is enabled.				

Table 52. Time Compare Value - TIMECMP (Address 24h)
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
25:0	Time Compare Value	R/W	Time value for frame age calculation.	3B9ACA0h
This register sets the time that a frame can be queued in the device before it is regarded as being too old and dropped. Aging occurs after 32 ns times the value of this register. Default value is 2 seconds. Setting TIMECMP to zero effectively disables aging. Requires that ports are reset [for example: setting MAC_CONF bits 29 (Port Reset), 5 (Rx Reset) and 4 (Tx Reset)].				

Table 53. SlowData - SLOWDATA (Address 2Ch)
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
31:0	SlowData	R/O	This is the read data from the slow registers.	0
When read, the SlowDone bit in the SYSTEM::CPUCTRL register is cleared. This register is only present when the Parallel Interface is enabled (ICPU_PI_En is tied low).				

Table 54. CPU Control - CPUCTRL (Address 30h)
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
11:8	PI Wait	R/W	Number of clock cycles to wait after PI_nCS is seen low before other inputs are sampled. The unit is 8 ns, thereby allowing the input signals to be unstable for up to 120 ns more than the data sheet specifies. The field should be lowered to match what the PI master can deliver. The value of this field is only used in case the Parallel Interface is enabled (ICPU_PI_En is tied low).	Fh
7	Learn Truncate	R/W	Captured frames for learning will be truncated to 64 bytes.	0
5	ExtCpu Use Slow	R/W	Use slow mode for slow registers when accessed through PI, splitting a read operation into two phases. The value of this field is only used in case the Parallel Interface is enabled (ICPU_PI_En is tied low).	0
4	SlowDone	R/O	Set when a slow register operation is complete internally. Cleared when the SLOWDATA register is read, or when a new slow data operation is initiated. The value of this field is only valid in case the Parallel Interface is enabled (ICPU_PI_En is tied low).	0
3	CPU Rx Frame Ready	R/O	Set as long as a packet is ready in the CPU capture buffer. This bit is a mirror of the Frame Ready bit in the CAPWRP register	0
2	Inverse INTR polarity	R/W	Set the interrupt signal to be active low. This field is only used in case the Parallel Interface is enabled (ICPU_PI_En is tied low).	0
1	Int Enable Packet	R/W	Enables interrupt when a CPU Rx Frame is ready. This field is only used in case the Parallel Interface is enabled (ICPU_PI_En is tied low).	0
0	Int Enable Slow	R/W	Enables interrupt when a slow register read or write is complete. This field is only used in case the Parallel Interface is enabled (ICPU_PI_En is tied low).	0

Table 55. General Purpose IO - GPIO (Address 34h)
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
9	Output Enable 0	R/W	GPIO0 will be of type output.	0
8	Output Enable 1	R/W	GPIO1 will be of type output.	0
7	Output Enable 2	R/W	GPIO2 will be of type output.	0
6	Output Enable 3	R/W	GPIO3 will be of type output.	0
5	Output Enable 4	R/W	GPIO4 will be of type output.	0
4	Data Value 0	R/W	The value seen on the GPIO0 pins when read, and the value to drive when written.	1

Table 55. General Purpose IO - GPIO (Address 34h) *(continued)*
Block 7 Subblock 0

Bit	Name	Mode	Description	Default
3	Data Value 1	R/W	The value seen on the GPIO1 pins when read, and the value to drive when written.	1
2	Data Value 2	R/W	The value seen on the GPIO2 pins when read, and the value to drive when written.	1
1	Data Value 3	R/W	The value seen on the GPIO3 pins when read, and the value to drive when written.	1
0	Data Value 4	R/W	The value seen on the GPIO4 pins when read, and the value to drive when written.	1
Note: When the GPIO pins are set into output mode, the value read will match the value written.				

MAC Block Registers (Block 1)

Table 56. MAC Config - MACCONF (Address 00h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31	Excessive Collision Back Off Disable	R/W	Determines whether or not the MAC backs off after an excessive collision has occurred. If set, back off is disabled after excessive collisions.	0
29	Port Reset	R/W	The port is held reset while this bit is set. Only the Shared FIFO block registers and Shared FIFO drop counters (C_RXDROP and C_TXOVFL) will be reset. All other port configurations will be kept.	1
28	Tx_en	R/W	Enable frame transmission.	0
27	Seed Load	R/W	Load seed for back off algorithm.	0
26:19	Back Off Seed	R/W	Value used to seed the randomizer used for the back off algorithm. To load a seed into the randomizer a transmit clock must be present, and the transmitter must be idle. The 'Seed Load' must be asserted for at least 1 µs, and the 'Back Off Seed' field must not be changed simultaneously with deassertion of 'Seed Load'.	0
18	Full Duplex	R/W	Enable full duplex mode.	1
17	Giga Mode	R/W	Set MAC to gigabit mode.	0
16	Rx_en	R/W	Enable frame receive.	0
14	VLAN Awareness	R/W	Allow tagged frames to be max_length+4 bytes long.	0
10:6	Tx IPG	R/W	The interframe gap between two consecutive transmitted frames. Recommended: 10/100=17. 1 G=5.	0
5	Rx Reset	R/W	The receive domain is held reset while this bit is set.	1

Table 56. MAC Config - MACCONF (Address 00h) *(continued)*
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
4	Tx Reset	R/W	The transmit domain is held reset while this bit is set.	1
3	Local Rx Clock	R/W	If set - use transmitter clock for rx domain. Used in loopback tests only.	0
2:0	Transmit Clock Select	R/W	000=No transmit clock. 001=125 MHz (RGMII-1000I). 010=25 MHz (RGMII-100). 011=2.5 MHz (RGMII-10). 100=External MII RxClk (Test only).	0
Note: This register configures most MAC functions. When changing the clock or duplex select fields, the register must be written twice. Once with all three reset flags asserted, once without.				

Table 57. Half Duplex Gaps - MACHDXGAP (Address 02h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
19:16	Reserved	R/W	Must be default value.	7
15:12	BACKOFFBIAS	R/W	This value can be used to adjust the back off time with a resolution of 8 bit times. If the register is increased by 1, the back off time will be decreased by 8 bit times. The correct value depends on the delay from the MAC to the I/O pads and should normally be set to its default value.	0
11:8	LCOLPOS	R/W	Late Collision Position. Adjusts the border between a collision and a late collision in steps of 1 byte. According to IEEE 802.3 section 21.3 this border is allowed to be on data byte 56 (counting frame data from 1), that is, a frame experiencing a collision on data byte 55 will always be retransmitted, and on byte 57 it will never be retransmitted. Using LCOLPOS=2, the border will be in this range. For each higher LCOLPOS value, the border is moved 1 byte higher.	3
7:4	IFG2	R/W	Second part of half duplex RX to TX Inter Frame Gap. On RGMII RX-to-TX_IFG=4.5 + (IFG1 + IFG2) / 2 byte. Within IFG2, transitions on CRS are ignored.	8
3:0	IFG1	R/W	First part of half duplex RX to TX Inter Frame Gap. The sum of IFG1 and IFG2 times the RX to TX IFG. Within IFG1 this timing will be restarted if CRS has multiple high-low transitions, that is, is noisy.	6

Table 58. Flow Control Setup - FCCONF (Address 04h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
17	Zero Pause Enable	R/W	If set, a zero delay pause frame is transmitted when pause condition is left.	0
16	Flow Control Obey	R/W	Obey pause control frames.	0
15:0	Pause Value	R/W	This value is inserted into the generated pause frames.	0
Note: This register controls the flow control setup. The frames will be generated by the conditions set up in POOLCONF and/or POLICECONF.				

Table 59. Flow Control SMAC High - FCMACHI (Address 08h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Flow Control SMAC High	R/W	These are the upper 3 bytes inserted in the generated flow control frames.	0

Table 60. Flow Control SMAC Low - FCMACLO (Address 0Ch)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Flow Control SMAC Low	R/W	These are the lower 3 bytes inserted in the generated flow control frames.	0

Table 61. Max Length - MAXLEN (Address 10h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
21:17	Shaping Mode	R/W	The shaper and policer can shape/police traffic in two ways: Either at line-rate or at payload-rate. At line-rate, IFG and preamble is added for each packet resulting in shaping/policing the line bandwidth. At payload-rate, IFG and preamble is not added resulting in shaping/policing the payload bandwidth. Line-rate: Set Rate Control to 2. Payload-rate: Set Rate Control to 0.	0
16	Type/Len Check	R/W	Enable check of valid type/length field. If enabled, ingress frames can be dropped due to inrange/outrange errors.	0
15:0	Max Length	R/W	Frame will be 'long' (dropped and counted in 'longs' counter) if frame length exceeds this limit. The internal limit is 12.2 kilobytes. See VLAN Awareness in the MACCONF register for tagged length check.	5EEh

Table 62. Shaper Setup - SHAPECONF (Address 11h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
30	Enable	R/W	Hold back frame transmission when bucket is filled.	0
29	Reset	R/W	Reset counters in shaper. Counters are kept reset while this bit is set.	0
24:16	Bucket Threshold	R/W	Bucket is filled when the fill level is above this number of 512 byte slices.	0
11:0	Data Rate	R/W	The bucket is emptied with this rate. Rate unit is speed dependent: 1 G = 244 kbps. 100 M=48.8 kbps. 10 M=4.88 kbps.	0

Table 63. Policer Setup - POLICECONF (Address 12h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
30	Enable	R/W	Generate flow control frames when bucket is filled.	0
29	Reset	R/W	Reset counters in policer. Counters are kept reset while this bit is set.	0
24:16	Bucket Threshold	R/W	Bucket is filled when the fill level is above this number of 512 byte slices.	0
11:0	Data Rate	R/W	The bucket is emptied at this rate. Rate unit is as for SHAPECONF.	0
Note: The flow control frames are generated as configured in the flow control configuration registers.				

Table 64. Multicast Storm Setup - MCSTORMCONF (Address 13h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31	Enable	R/W	Drop ingress frames when the bucket is filled.	0
29	Reset	R/W	Reset counters in multicast storm bucket. Counters are kept reset while this bit is set.	0
24:16	Bucket Threshold	R/W	Bucket is filled when the fill level is above this number of 512 byte slices.	0
11:0	Data Rate	R/W	The bucket is emptied at this rate. Rate unit is as for SHAPECONF.	0
Note: The dropped frames are counted by the C_RXDROP counter.				

Table 65. Advanced Port Mode Setup - ADVPORTM (Address 19h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
9	PausePrio	R/W	If set, the queue transmit interface will not start when a pause frame is going to be generated by the MAC. In 1G, PausePrio must be 1. In 10/100M, PausePrio must be 0.	1
8	WaitForDone	R/W	If set, the queue transmit interface will not start the next frame until the MAC is completely done. In 1G, WaitForDone must be 0. In 10/100M, WaitForDone must be 1.	0
7	GtxDelay2	R/W	If set the GTX clock will be delayed a fourth of the GTX clock period related to data. This may help resolving timing problems against RGMII phys.	0
6	Loopback	R/W	Enable internal loopback. All egress frames will be echoed through the ingress path.	0
4	GtxDelay1	R/W	If set the GTX clock will be delayed half of the GTX clock period related to data. This may help resolving timing problems against RGMII PHYs.	0
3	Halt GTX	R/W	Set if GTX is to be kept silent.	0

Table 66. Transmit Modify Setup - TXUPDCFG (Address 24h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
15:4	Untagged vID	R/W	Frames with this vID will be transmitted untagged.	0
3	Untagged vID Enable	R/W	If set, frame in a specific VLAN will not be tagged, even if port is set up for tagging.	0
2	Update CRC Tag	R/W	Recalculate CRC after tag insertion/removal.	1
1	Update CRC CPU	R/W	Recalculate CRC for CPU transmit frames.	0
0	Insert TAG	R/W	If set, frames will be transmitted with VLAN tags.	0

Table 67. CFI Drop Counter - CFIDROP (Address 25h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31:0	CFI Drops Counter	R/O	Counts the number of CFI marked frames dropped before transmit on this port.	0
Note: If a port is set up to transmit untagged frames, CFI marked frames will be dropped.				

Shared FIFO Block Registers (Block 1)

Table 68. CPU Transmit DATA - CPUTXDAT (Address C0h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31:0	CPU Transmit DATA	W/O	Write 4 bytes of Tx Data. An even number of writes must be performed.	0
Note: This register is used by the CPU to transmit frames. The first two words must hold a header for the frame, which is: word1=length shl 16 word2=0x00000254				

Table 69. MISC Control Register - MISCFIFO (Address C4h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
22	Flush Low Enable	R/W	Enables flushing of all low priority ingress frames when “INGRESS DROP LOW” level is reached.	1
21:16	Subprio WM Adjust	R/W	This value specifies the watermarks for low subpriority frames, as an offset to the watermarks given through the DROPCFG register. Frames with low sub priority will be checked against watermarks this level below the levels configured in DROPCFG. That is, “Ingress Drop Low” configured to 10 and this field to 3 means that frames with low main priority and high subpriority will be dropped if more than 10 chunks of ingress data is present, and for low sub priority frames when only 7 chunks are used.	0
1	Rewind CPU Tx	W/O	Cancel the CPU Tx Data that have been written through the Transmit Data register.	0
0	CPU Tx	W/O	Transmit the frame data written through the Transmit Data register.	0

Table 70. Pool Control Register - POOLCFG (Address CCh)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31	Use Total Count	R/W	If set, the total size (ingress + egress) of allocated memory will be used when referencing threshold limits. This feature is only added for special applications, and should in general not be used.	0
30	Track Pool Usage	R/W	When set the freepool register will report the largest memory usage since last read instead of current.	0
29:24	Egress Low	R/W	Stop internal flow control towards ingress ports when below this watermark.	30h

Table 70. Pool Control Register - POOLCFG (Address CCh) *(continued)*
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
21:16	Egress High	R/W	Start internal flow control towards ingress ports when above this watermark.	30h
14	Ingress Protection Method	R/W	Protection for overflow in ingress queues is done either by disobeying egress status (0) or by issuing pause frames to the MAC.	0
13:8	Ingress Low	R/W	Stop pause generation or resume obeying egress status when below this watermark.	2Ch
5:0	Ingress High	R/W	Start pause generation or stop obeying egress status when above this watermark.	2Ch
Note: All values are in 256-bytes slices.				

Table 71. Drop Control Register - DROPCFG (Address DCh)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31:30	Early Transmit Level	R/W	Allow MAC to initiate transmit before the whole packet has been received from the ingress device, if the amount of egress data is less than this level. The unit is 2 kilobytes.	0
29:24	Egress Drop Low	R/W	Drop low priority egress frames when above this watermark.	Ah
21:16	Egress Drop All	R/W	Drop all egress frames when above this watermark.	32h
13:8	Ingress Drop Low	R/W	Drop low priority ingress frames when above this watermark.	Ah
5:0	Ingress Drop All	R/W	Drop all ingress frames from MAC when above this watermark.	32h
Note: All values are in 256-bytes slices.				

Table 72. Misc Status - MISCSTAT (Address C8h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
8	CPU Tx Data Pending	R/O	Indicates that Tx data recently written to CPUTXDAT have not yet been transferred to the Tx queue. No further writes to CPUTXDAT must take place before this bit is clear.	0
7	CPU Tx Data Overflow	R/O	Indicates that additional TX data was written by the CPU before the “data pending” bit above was clear. When this flag is detected, the rewind command must be issued.	0

Table 73. Free RAM Counter - FREEPOOL (Address D8h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
21:16	FreeCount	R/O	Number of free 256-byte slices of RAM in shared FIFO.	0
13:8	Ingress Used	R/O	Number of 256-byte slices used for ingress data.	3
5:0	Egress Used	R/O	Number of 256-byte slices used for egress data.	2
Note: The default values for the use counters are above zero due to a minimum allocation for each queue of 1 slice. None of the three fields in this register can therefore hit more than 63.				

Categorizer Block Registers (Block 1)

The registers in this section primarily control the prioritization of incoming frames. The function of each of the registers cannot be disabled, so a network-unused value must be applied to a field if the function is not wanted.

Table 74. Categorizer Config - CATCONF (Address 60h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31	Tag Only	R/W	If set, the priority of the tagged frames will be calculated from the TCI priority within the frame, and any higher layer information will be disregarded. Untagged frames will get the priority FS2.	0
30	Ignore TCI	R/W	Always report the PvID for incoming frames instead of taking it from the tag field within frame.	1
29	Ignore CTRL	R/W	Forward MAC Control frames. If set and if the destination MAC address is in the BDPU range, frames will be forwarded based on the BPDU Capture flag. This field does not affect pause frames.	0
26	DS Only	R/W	If set, the priority will be based on the DS field for all IP frames. If cleared, only IP frames with protocol and port match will be based on DS. If DS field is not used, the priority will be FS7.	0
25	Keep Tag	R/W	The frame tag will not be removed from the frame when forwarding. This bit must be set when operating as a VLAN unaware switch, so that a tag field will be left untouched.	1
23	Drop Untagged	R/W	If set, all untagged or priority-tagged frames received on this port will be dropped (except BPDUs).	0
22	IGMP Capture	R/W	Recognize IGMP frames and redirect them to CPU.	0
21	BPDU Capture	R/W	Recognize BPDUs and redirect them to CPU. If this flag is not set, BPDUs will be forwarded as any other multicast frame.	1

Table 74. Categorizer Config - CATCONF (Address 60h) *(continued)*
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
20	Enable IPMC Flood Mask	R/W	When set, IP multicast frames outside 224.0.0.x with unknown DMAC will be will be flooded with the IFLODMSK. If not set, the packets will be flooded with the MFLODMSK.	0
19	ARP BC Capture	R/W	Recognize ARP broadcasts and copy them to CPU.	0
18	Drop BadMac	R/W	Drop packets with zero source or destination MAC address or packets with multicast source MAC address.	0
17	IPMC Ctrl snoop	R/W	Recognize IP multicast frames in DIP range 224.0.0.x, and copy them to the CPU capture buffer.	0
15	IP BC Capture	R/W	Recognize IP broadcasts and copy them to CPU.	0
14	Forward PAUSE	R/W	Forward MAC Pause frames. If set and if the destination MAC address is in the BDPU range, frames will be forwarded based on the BPDU Capture flag.	0

Table 75. Categorizer Map Tag - CATTAG (Address 64h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
7	Categorizer Map Tag	R/W	Internal priority for tag priority no. 7	1
6	Categorizer Map Tag	R/W	Internal priority for tag priority no. 6	1
5	Categorizer Map Tag	R/W	Internal priority for tag priority no. 5	1
4	Categorizer Map Tag	R/W	Internal priority for tag priority no. 4	1
3	Categorizer Map Tag	R/W	Internal priority for tag priority no. 3	1
2	Categorizer Map Tag	R/W	Internal priority for tag priority no. 2	1
1	Categorizer Map Tag	R/W	Internal priority for tag priority no. 1	1
0	Categorizer Map Tag	R/W	Internal priority for tag priority no. 0	1
Note: If the priority is to be based on the tag field, this register maps the 8 possible priorities to the internal low/high queues.				

Table 76. Categorizer SUBMap Tag - CATSUBTAG (Address 65h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
7	Categorizer SUBMap Tag	R/W	Internal subpriority for tag priority no. 7	1
6	Categorizer SUBMap Tag	R/W	Internal subpriority for tag priority no. 6	1
5	Categorizer SUBMap Tag	R/W	Internal subpriority for tag priority no. 5	1
4	Categorizer SUBMap Tag	R/W	Internal subpriority for tag priority no. 4	1

Table 76. Categorizer SUBMap Tag - CATSUBTAG (Address 65h) *(continued)*
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
3	Categorizer SUBMap Tag	R/W	Internal subpriority for tag priority no. 3	1
2	Categorizer SUBMap Tag	R/W	Internal subpriority for tag priority no. 2	1
1	Categorizer SUBMap Tag	R/W	Internal subpriority for tag priority no. 1	1
0	Categorizer SUBMap Tag	R/W	Internal subpriority for tag priority no. 0	1
Note: If the subpriority is to be based on the tag field, this register maps the eight possible priorities to the internal subpriority class.				

Table 77. EtherType Register - CATETHT (Address 68h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
15:0	EtherType Register	R/W	EtherType value for prioritization	800h
Note: Refer to "Frame Priority Determination," which begins on page 30.				

Table 78. DSAP Register - CATDSAP (Address 6Ch)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
7:0	DSAP Register	R/W	A DSAP value for prioritization	AAh
Note: Refer to "Frame Priority Determination," which begins on page 30.				

Table 79. IP Protocol Register - CATIPPRT (Address 70h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
7:0	IP Protocol Register	R/W	An IP protocol for prioritization	6
Note: Refer to "Frame Priority Determination," which begins on page 30.				

Table 80. Categorizer Priorities - CATPRIO (Address 74h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
7	FS8	R/W	Non-IP frame priority	1
6	FS7	R/W	TCP/UDP frames with unconfigured port number in the Categorizer	1
5	FS6	R/W	Non-IP frames with configured EtherType	1
4	FS5	R/W	802.2 frames, non-SNAP and non-DSAP match	1

Table 80. Categorizer Priorities - CATPRIO (Address 74h) *(continued)*
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
3	FS4	R/W	802.2 frames, non-SNAP with DSAP match	1
2	---	R/W	Unused	1
1	FS2	R/W	Untagged frames when only using priority field in tag ID	1
0	FS1	R/W	Tagged frames with CFI field =1	1
Note: This register maps all the frame classifications into high or low priority. Refer to Figure 4, on page 32.				

Table 81. Categorizer SubPriorities - CATSUBPRIO (Address 75h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
7	FS8	R/W	Non-IP frame subpriority	1
6	FS7	R/W	TCP/UDP frames with unconfigured port number in the Categorizer	1
5	FS6	R/W	Non-IP frames with configured EtherType	1
4	FS5	R/W	802.2 frames, non-SNAP and non-DSAP match	1
3	FS4	R/W	802.2 frames, non-SNAP with DSAP match	1
2	---	R/W	Unused	1
1	FS2	R/W	Untagged frames when only using priority field in tag ID	1
0	FS1	R/W	Tagged frames with CFI field=1	1
Note: This register maps all the frame classifications into high or low subpriority.				

Table 82. DS Mapping Register High - CATDSMAPH (Address 78h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31:0	DS Mapping Register High	R/W	This mask defines the priority for DSCP 63,62,...,32	FFFFFFFFh

Table 83. DS Mapping Register Low - CATDSMAPL (Address 79h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31:0	DS Mapping Register Low	R/W	This mask defines the priority for DSCP 31,30,...,0	FFFFFFFFh

Table 84. DS SubMapping Register High - CATDSSUBMAPH (Address 7Ah)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31:0	DS SubMapping Register High	R/W	This mask defines the subpriority for DSCP 63,62,...,32	FFFFFFFFh

Table 85. DS SubMapping Register Low - CATDSSUBMAPL (Address 7Bh)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31:0	DS SubMapping Register Low	R/W	This mask defines the subpriority for DSCP 31,30,...,0	FFFFFFFFh

Table 86. PVID Register - CATPVID (Address 7Ch)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31:20	Port vID	R/W	Default vID for untagged frames, classified as low priority and either high or low subpriority.	0
19	Port CFI Field	R/W	Default CFI bit for untagged frames, classified as low priority and either high or low subpriority.	0
18:16	Port VLAN Priority	R/W	Default priority for untagged frames, classified as low priority and either high or low subpriority.	0
15:4	Port vID	R/W	Default vID for untagged frames, classified as high priority and either high or low subpriority.	0
3	Port CFI Field	R/W	Default CFI bit for untagged frames, classified as high priority and either high or low subpriority.	0
2:0	Port VLAN Priority	R/W	Default priority for untagged frames, classified as high priority and either high or low subpriority.	0
Note: If bit 30 of CATCONF is set, all frames will get the PvID VLAN information as forwarding information.				

Table 87. TCP/UDP Port Register 1 - CATPORT1 (Address 80h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31:16	Port 1	R/W	When this port number is matched in an incoming frame, priority can optionally be based on the frame's DS code point.	50h
15:0	Port 2	R/W	When this port number is matched in an incoming frame, priority can optionally be based on the frame's DS code point.	14h

Table 88. TCP/UDP Port Register 2 - CATPORT2 (Address 84h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31:16	Port 3	R/W	When this port number is matched in an incoming frame, priority can optionally be based on the frame's DS code point.	50h
15:0	Port 4	R/W	When this port number is matched in an incoming frame, priority can optionally be based on the frame's DS code point.	14h

Table 89. TCP/UDP Port Register 3 - CATPORT3 (Address 88h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31:16	Port 5	R/W	When this port number is matched in an incoming frame, priority can optionally be based on the frame's DS code point.	50h
15:0	Port 6	R/W	When this port number is matched in an incoming frame, priority can optionally be based on the frame's DS code point.	14h

Table 90. TCP/UDP Port Register 4 - CATPORT4 (Address 8Ch)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31:16	Port 7	R/W	When this port number is matched in an incoming frame, priority can optionally be based on the frame's DS code point.	50h
15:0	Port 8	R/W	When this port number is matched in an incoming frame, priority can optionally be based on the frame's DS code point.	14h

Table 91. TCP/UDP Port Register 2 - CATPORT5 (Address 90h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31:16	Port 9	R/W	When this port number is matched in an incoming frame, priority can optionally be based on the frame's DS code point.	50h
15:0	Port 10	R/W	When this port number is matched in an incoming frame, priority can optionally be based on the frame's DS code point.	14h

Statistics Block Registers (Block 1)

Table 92. Rx Total Bad Packets - RXBADPKT (Address 37h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx Total Bad Packets	R/O	Bad packets only	0

Table 93. Rx Control Packets - RXCTRL (Address 39h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx Control Packets	R/O	Number of MAC control frames received	0

Table 94. Rx Pause Frames - C_RXPAUSE (Address 3Dh)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx Pause Frames	R/O	Number of received pause frames	0

Table 95. Tx Pause Frames - C_TXPAUSE (Address 3Eh)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Tx Pause Frames	R/O	Number of transmitted pause frames	0

Table 96. Rx Classified Drops - C_RXCATDROP (Address 3Fh)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx Classified Drops	R/O	Number of packets dropped due to classifier rules (see the CATCONF register)	0

Table 97. Tx Total Error Packets - TXERR (Address 44h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Tx Total Error Packets	R/O	Number of bad packets transmitted	0

Detailed Counters Block Registers (Block 1)

The counters in this block are for RMON-II counter support. They are cleared by writing the C_RXDROP register with any value. All counters will wrap around at their maximum values. For 1G-operation, 24-bit counters can wrap every 10 seconds whereas 32-bit counters can wrap around every 30 seconds depending on the traffic load. For 100 Mbps and 10 Mbps operations, these values must be scaled by a factor of 10 and 100, respectively.

Table 98. Rx Octets - C_RXOCT (Address 50h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31:0	Rx Octets	R/O	Received octets in good and bad packets	0

Table 99. Tx Octets - C_TXOCT (Address 51h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
31:0	Tx Octets	R/O	Transmitted octets in good and bad packets	0

Table 100. Rx Drops - C_RXDROP (Address A0h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
15:0	Rx Drops	R/W	Frames dropped due to lack of receive buffer	0
Note: Any write to this register will clear all counters.				

Table 101. Rx Packets - C_RXPKT (Address A1h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx Packets	R/O	Number of good and bad packets	0
Note: For counting only good RX packets, the sum of high and low priority packets must be used (C_RXHP+C_RXLP).				

Table 102. Rx Broadcasts - C_RXBC (Address A2h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx Broadcasts	R/O	Number of good broadcasts	0

Table 103. Rx Multicasts - C_RXMC (Address A3h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx Multicasts	R/O	Number of good multicasts	0

Table 104. Rx CRC/ALIGN - C_RXCRC (Address A4h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Number of CRC errors	R/O	Alignment errors and RX_ER events	0

Table 105. Rx Undersize - C_RXSHT (Address A5h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx Undersize	R/O	Number of short frames with valid CRC (<64 Bytes)	0

Table 106. Rx Oversize - C_RXLONG (Address A6h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx Oversize	R/O	Number of long frames with valid CRC (according to max_length register)	0

Table 107. Rx Fragments - C_RXFRAG (Address A7h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx Fragments	R/O	Number of short frames with invalid CRC (<64 bytes)	0

Table 108. Rx Jabbers - C_RXJAB (Address A8h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx Jabbers	R/O	Number of long frames with invalid CRC (according to max_length register)	0

Table 109. Rx 64 Bytes - C_RX64 (Address A9h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx 64 Bytes	R/O	Number of 64-byte frames in good and bad packets	0

Table 110. Rx 65-127 Bytes - C_RX65 (Address AAh)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx 65-127 Bytes	R/O	Number of 65-127-byte frames in good and bad packets	0

Table 111. Rx 128-255 Bytes - C_RX128 (Address ABh)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx 128-255 Bytes	R/O	Number of 128-255-byte frames in good and bad packets	0

Table 112. Rx 256-511 Bytes - C_RX256 (Address ACh)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx 256-511 Bytes	R/O	Number of 256-511-byte frames in good and bad packets	0

Table 113. Rx 512-1023 Bytes - C_RX512 (Address ADh)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx 512-1023 Bytes	R/O	Number of 512-1023-byte frames in good and bad packets	0

Table 114. Rx 1024-long Bytes - C_RX1024 (Address AEh)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx 1024-long Bytes	R/O	Number of 1024-max_length-byte frames in good and bad packets	0

Table 115. Tx Drops - C_TXDROP (Address AFh)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Tx Drops	R/O	Number of frames dropped due to excessive collision, late collision or frame aging	0

Table 116. Tx Packets - C_TXPKT (Address B0h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Tx Packets	R/O	Number of good packets	0

Table 117. Tx Broadcasts - C_TXBC (Address B1h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Tx Broadcasts	R/O	Number of good broadcasts	0

Table 118. Tx Multicasts - C_TXMC (Address B2h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Tx Multicasts	R/O	Number of good multicasts	0

Table 119. Tx Collisions - C_TXCOL (Address B3h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Tx Collisions	R/O	Number of collisions that transmitting frames experience. An excessive number of frame collisions yields 16 counts.	0

Table 120. Tx 64 Bytes - C_TX64 (Address B4h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Tx 64 Bytes	R/O	Number of 64-byte frames in good and bad packets	0

Table 121. Tx 65-127 Bytes - C_TX65 (Address B5h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Tx 65-127 Bytes	R/O	Number of 65-127-byte frames in good and bad packets	0

Table 122. Tx 128-255 Bytes - C_TX128 (Address B6h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Tx 128-255 Bytes	R/O	Number of 128-255-byte frames in good and bad packets	0

Table 123. Tx 256-511 Bytes - C_TX256 (Address B7h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Tx 256-511 Bytes	R/O	Number of 256-511-byte frames in good and bad packets	0

Table 124. Tx 512-1023 Bytes - C_TX512 (Address B8h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Tx 512-1023 Bytes	R/O	Number of 512-1023-byte frames in good and bad packets	0

Table 125. Tx 1024-long Bytes - C_TX1024 (Address B9h)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Tx 1024-long Bytes	R/O	Number of 1024-max_length-byte frames in good and bad packets	0

Table 126. Tx FIFO Drops - C_TXOVFL (Address BAh)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
15:0	Tx FIFO Drops	R/O	Number of frames dropped due to lack of transmit buffer	0

Table 127. Rx High Priority Frames - C_RXHP (Address BBh)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx High Priority Frames	R/O	Number of Rx frames classified as high priority with either high or low subpriority.	0

Table 128. Rx Low Priority Frames - C_RXLP (Address BCh)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Rx Low Priority Frames	R/O	Number of Rx frames classified as low priority with either high or low subpriority.	0

Table 129. Tx High Priority Frames - C_TXHP (Address BDh)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Tx High Priority Frames	R/O	Number of Tx frames classified as high priority with either high or low subpriority.	0

Table 130. Tx Low Priority Frames - C_TXLP (Address BEh)
Block 1 Subblock 0-11

Bit	Name	Mode	Description	Default
23:0	Tx Low Priority Frames	R/O	Number of Rx frames classified as low priority with either high or low subpriority.	0

MII Management Bus Block Registers (Block 3)

Table 131. MII-M Status - MIIMSTAT (Address 00h)
Block 3 Subblock 0

Bit	Name	Mode	Description	Default
3	Busy	R/O	The bus is active	0
1	Reading	R/O	A read operation is in progress	0
0	Writing	R/O	A write operation is in progress	0
Note: This register provides the status of the management bus.				

Table 132. MII-M Command - MIIMCMD (Address 01h)
Block 3 Subblock 0

Bit	Name	Mode	Description	Default
27	Scan	R/W	Enable SCAN mode	0
26	Operation	R/W	Set MII-M operation to read (1) or write (0)	0
25:21	PHY Address	R/W	Address of PHY to operate on	0
20:16	PHY Register	R/W	Register number for operation	0
15:0	Write Data	R/W	Data to write in write operations	0
Note: An operation commences when this register is written.				

Table 133. MII-M Return Data - MIIMDATA (Address 02h)
Block 3 Subblock 0

Bit	Name	Mode	Description	Default
16	Failure	R/O	Operation failed - no PHY read reply	0
15:0	MII-M Return Data	R/O	Read Data	0

Table 134. MII-M Prescaler - MIIMPRES (Address 03h)
Block 3 Subblock 0

Bit	Name	Mode	Description	Default
5:0	Prescale Value	R/W	The divisor for the MIIM clock. A 62.5 MHz clock will be divided by this value. Default clock with default value at approximately. 2 MHz. The register must not be configured to any value less than 5.	20h

Table 135. MII-M Scan setup - MIIMSCAN (Address 04h)
Block 3 Subblock 0

Bit	Name	Mode	Description	Default
25:21	PhyAddress HIGH	R/W	This is the upper limit for the PHY addresses to be scanned	0
20:16	PhyAddress LOW	R/W	This is the low limit for the PHY addresses to be scanned	0
15:0	PhyRegMask	R/W	The MASK with target bits in the PHY replies	0
Note: When the SCAN bit is set in the command register, the operation set up is repeated indefinitely until the bit is reset by a register write. The PHY address will be looping between the lower and upper limits of the address. In case of a read operation, the scan results register bit (phy addr) gets the value 1 if (phy_reply and phy_mask)=phy_mask, otherwise 0. By this mechanism, it is possible to have an updated link state vector for all 32 PHYs attached to the MII-M bus.				

Table 136. MII-M Scan Results - MIIMSRES (Address 05h)
Block 3 Subblock 0

Bit	Name	Mode	Description	Default
31:0	MII-M Scan Results	R/O	The scan results as explained in Table 135.	0

Memory Initialization Block Registers (Block 3)

Table 137. Initialize - MEMINIT (Address 00h)
Block 3 Subblock 1

Bit	Name	Mode	Description	Default
24:8	Memory Operation	R/W	For proper initialization of RAM, this field must be written 10104h. For initiating reading of result of RAM initialization, the field must be written 200h.	0
7:0	Memory Id	R/W	The identifier for the RAM to initialize or to initiate reading of result from. Range is 0-19.	0

Table 138. Read Result - MEMRES (Address 01h)
Block 3 Subblock 1

Bit	Name	Mode	Description	Default
31:2	Reserved	R/O	Contains garbage.	0
1:0	Memory Result	R/O	This field contains the result of the memory initialization. For proper initialization of RAM, this field must read 3h.	0

Frame Arbiter Block Registers (Block 5)

Table 139. Arbiter Empty - ARBEMPTY (Address 0Ch)
Block 5 Subblock 0

Bit	Name	Mode	Description	Default
11:0	Arbiter Empty	R/O	Status per source port. No frame is pending from this source.	FFFh

Note: This register is to be used when resetting a port, to stop interaction with other ports before shutting the port down. A correct reset sequence is:

```
Set DROPCFG to 0x00000000 (drop all frames)
Set POOLCFG to 0x00000000 (ignore back pressure)
Clear RX_EN bit in MACCONF
Wait until arbiter empty bit is set for the port
If time out from above operation:
              Set DROPCFG, POOLCFG and RX_EN=0 on all ports
              Wait until arbiter empty bit is set for all ports
              If time out from above operation:
              Make a full core reset.
              Reconfigure everything according to mode
              else
              Reset the target port
              Reconfigure all ports according to mode
              endif
else
              Reset the target port
              Reconfigure port according to mode
endif
```

The time-out period should be set to 500msec, but is not critical.
For further information on the reset procedure, please refer to the Stansted Switch Software Manual.

Table 140. Arbiter Discard - ARBDISC (Address 0Eh)
Block 5 Subblock 0

Bit	Name	Mode	Description	Default
11:0	Arbiter Discard	R/W	Config bit per port. All frames from a source will be discarded if the source corresponding bit is set in this register.	0

CPU Capture Block Registers (Block 4)

Table 141. Read Pointer - CAPREADP (Address 00h)
Block 4 Subblock 4

Bit	Name	Mode	Description	Default
10:0	Read Pointer	R/W	The physical address of the first frame for reading. The RAM cells are 8 bytes wide. (2048 x 8=16 kilobytes buffer).	0
Note: If anything is written to this register, the read pointer will be advanced to the next frame for readout. This register is only present in Subblock 4. Subblocks 0-3 are for frame readout.				

Table 142. Write Pointer - CAPWRP (Address 03h)
Block 4 Subblock 4

Bit	Name	Mode	Description	Default
17	Frame Ready	R/O	A frame is ready for readout. Buffer address 0 is always pointing to the next frame available.	0
16	Frame Dropped	R/W	A frame for the CPU buffer is dropped due to lack of buffer space.	0
10:0	Write Pointer	R/O	The physical address of the start of the next incoming frame.	0
Note: The Frame Dropped status bit is cleared by writing anything to this register. This register is only present in Subblock 4. Subblocks 0-3 are for frame readout.				

Table 143. Full Reset - CAPRST (Address FFh)
Block 4 Subblock 7

Bit	Name	Mode	Description	Default
31:0	Full Reset	W/O	Writing anything to this register reset the CPU receive block fully.	0
Note: This register is only present in Subblock 7.				

Frame Analyzer Block Registers (Block 2)

Table 144. Advanced Learning Setup - ADVLEARN (Address 03h)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
31	Automode	R/W	Unicast source addresses will automatically be inserted into the MAC table.	1
30	Dropmode	R/W	Frames subject to learning or station port move will not be forwarded based on destination address.	0
29	VlanCheck	R/W	If a frame is discarded because of the VLAN Source Check flag in the VLAN table (see the VLANACES register), the source address will not be learned.	0
24	CpuLearn	R/W	Learn frames will be forwarded to the CPU Capture module.	0
11:0	LearnMirror	R/W	Learn frames will also be forwarded to ports marked in this mask.	0

Table 145. IP Multicast Flood Mask - IFLODMSK (Address 04h)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
11:0	IP Multicast Flood Mask	R/W	Port mask with allowed ports for unknown IP multicast flooding. Will be overruled if 'Flood FwdKill' flag is set. The mask only applies if the 'Enable IPMC Flood Mask' field in CATCONF register is set.	0

Table 146. VLAN Source Port Mask - VLANMASK (Address 05h)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
11:0	VLAN Source Port Mask	R/W	Mask for requiring VLAN Source Check on Ingress port. If bit<port> is set, the <port> must be marked in the ingress frame's VLAN Port Mask, otherwise the frame is dropped.	0

Table 147. Mac Address High - MACHDATA (Address 06h)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
27:16	VID	R/W	VID to be used in MAC CPU write operations. Additionally, VID is returned in read operations.	0
15:0	MAC Address High	R/W	Upper 16 MAC address bits for CPU write operations. Additionally, MAC Address High is returned in read operations.	0

Table 148. Mac Address Low - MACLDATA (Address 07h)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
31:0	Mac Address Low	R/W	Lower 32 MAC address bits for CPU operations	0

Table 149. Station Move Logger - ANMOVED (Address 08h)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
11:0	Station Move Logger	R/W	Sticky bit set when a station has been learned on a port while already learned on another port	0
Note: This register is cleared by writing the bits to be cleared. This mask can be used to detect topology problems in the network, where stations are learned on multiple ports repeatedly. If some bits in this register get asserted repeatedly, the ports can be shut down, or management warnings can be issued.				

Table 150. Aging Filter - ANAGEFIL (Address 09h)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
31	PID Enab	R/W	If set only entries with matching destination are aged	0
19:16	PID Value	R/W	Destination port for aging only the part of table with entries on specific ports	0
15	vID Enab	R/W	If set only entries with matching VLAN membership are aged	0
11:0	vID Value	R/W	VLAN Identifier for aging only the parts of the MAC table that contain entries belonging to specific VLANs	0
Note: This register sets up which entries should be touched by an aging operation. This way it is possible to have different aging periods in each VLAN, and to have quick removal of entries on specific ports.				

Table 151. Event Sticky Bits - ANEVENTS (Address 0Ah)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
21	Learn Drop	R/W	A frame was dropped as it was subject to learning, and the DropMode flag was set in ADVLEARN.	0
20	Aged Entry	R/W	An entry was removed at CPU Learn, or CPU requested an aging process.	0
19	CPU Learn Failed	R/W	An learn failed due to hash table depletion.	0
18	AUTO Learn Failed	R/W	A learn of incoming source MAC address failed due to hash table depletion.	0
17	Learn Remove	R/W	An entry was removed when learning a new source MAC address.	0
16	AUTO Learned	R/W	An entry was learned from an incoming frame.	0

Table 151. Event Sticky Bits - ANEVENTS (Address 0Ah) *(continued)*
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
15	AUTO Moved	R/W	A station was moved to another port.	0
14	Dropped	R/W	A packet was not transmitted to any ports.	0
13	Classified Drop	R/W	A packet was not forwarded due to classification (like BPDUs).	0
12	Classified Copy	R/W	A packet was copied to the CPU due to classification.	0
11	VLAN Discard	R/W	A packet was discarded due to lack of VLAN membership on source port.	0
10	FWD Discard	R/W	A packet was discarded due to missing forwarding state on source port.	0
9	Multicast Flood	R/W	A packet was flooded with multicast flooding mask.	0
8	Unicast Flood	R/W	A packet was flooded with unicast flooding mask.	0
7	Destination Known	R/W	A packet was forwarded with known destination MAC address.	0
6	Bucket3 Match	R/W	A destination was found in hash table bucket 3.	0
5	Bucket2 Match	R/W	A destination was found in hash table bucket 2.	0
4	Bucket1 Match	R/W	A destination was found in hash table bucket 1.	0
3	Bucket0 Match	R/W	A destination was found in hash table bucket 0.	0
2	CPU Operation	R/W	A CPU initiated operation has been processed.	1
1	DMAC Lookup	R/W	A destination address has been looked up in the MAC table.	0
0	SMAC Lookup	R/W	A source address has been looked up in the MAC table.	0
Note: This register contains various debug event logs. A sticky bit is cleared by writing it.				

Table 152. Event Sticky Mask - ANCNTMSK (Address 0Bh)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
21:0	Sticky Mask	R/W	This mask determines which events are to be counted by the ANCNTVAL register.	0
Note: Refer to the flag list in the ANEVENTS register. For proper operation, only one bit should be set in the mask. Otherwise - if two events occur simultaneously, they might be mistaken as only one event.				

Table 153. Event Sticky Counter - ANCNTVAL (Address 0Ch)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
27:0	Event Sticky Counter	R/W	This counter counts the number of events seen as specified in the Event Sticky Mask.	0

Table 154. Learn Mask - LERNMASK (Address 0Dh)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
11:0	Learn Mask	R/W	If bit<port> is set in this mask, incoming frames are subject to autolearning on that port.	FFFh

Table 155. Unicast Flood Mask - UFLODMSK (Address 0Eh)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
11:0	Unicast Flood Mask	R/W	Port mask with allowed ports for unknown unicast flooding.	FFFh

Table 156. Multicast Flood Mask - MFLODMSK (Address 0Fh)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
11:0	Multicast Flood Mask	R/W	Port mask with allowed ports for unknown multicast flooding, and for broadcast flooding	FFFh

Table 157. Receive Mask - RECVMASK (Address 10h)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
11:0	Receive Mask	R/W	If a port is not marked in this mask, incoming frames will be discarded.	0

Table 158. Aggregation Mode - AGGRCNTL (Address 20h)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
1:0	Aggregation Mode	R/W	Mode of aggregation 00=Reserved 01=SMAC 10=DMAC 11=SMAC xor DMAC	1h

Table 159. Aggregation Masks - AGGRMSKS (Address 30h - 3Fh)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
11:0	Mask	R/W	Mask used to select only one port within each aggregation group.	FFFh
Note: These 16 masks select a single port in each aggregated port group. If no aggregation is configured, these masks will be all 1s.				

Table 160. Destination Port Masks - DSTMASKS (Address 40h - 7Fh)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
11:0	Destination Port Mask	R/W	Mask used to translate a logical port number from a destination lookup into a set of ports.	See below
Note: By default, the first 12 port masks have the bit corresponding to their number set only. The remaining are cleared to 0 for multicasts. In normal situations, it does not make sense to change the first 12 masks from the default, except in aggregation setups. Otherwise frames to a station would be transmitted on another port than the port it was autolearned on.				

Table 161. Source Port Masks - SRCMASKS (Address 80h - 8Bh)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
25	CPU Copy	R/W	All frames from this port will be copied to the CPU capture buffer.	0
24	Mirror	R/W	All frames from this port will be mirrored to the port set in the Analyzer Config register.	0
11:0	Ports	R/W	Mask used to disallow frames from being forward from specific source port.	See below
Note: These masks are used to prevent frames from being looped back to the ports on which they were received and must be updated according to the aggregation configuration. A frame received on port n will use register 0x80 + n as a mask to filter out transmit ports to avoid loop back or to facilitate port grouping (port based VLANs). The default values are that all bits are set except for the index number. Example: srcmask 5 has a default value of 111111011111b, or FDFh.				

Table 162. Mac Table Command - MACACCES (Address B0h)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
14	CPUCopy	R/W	Frames for this destination will be copied to the CPU capture buffer.	0
13	FwdKill	R/W	Frames for this destination are dropped.	0
12	Ignore VLAN	R/W	The VLAN mask is ignored for this destination.	0

Table 162. Mac Table Command - MACACCES (Address B0h) *(continued)*
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
11	Aged Flag	R/W	This flag is set on every aging run. Entry is removed if set already. The flag is cleared when the entry is target for a source address lookup.	0
10	Valid	R/W	Entry is valid.	0
9	Locked	R/W	Entry is locked, and will not be removed by aging or autolearn pushout. Port move events are only registered in ANMOVE if a MAC address is learned with the lock flag cleared.	0
8:3	Destination Index	R/W	Index into the destination mask table. For unicasts a number below12 applies.	0
2:0	Mac Table Command	R/W	Mac Table Command.	0

Note: This register is used for updating or reading the MAC table from the CPU. The command selects between different operations and takes the following values:

```
000 Idle          The previous operation is complete.
001 Learn         The MAC data is learned into the table.
010 Forget        The MAC data is removed from the table.
011 Age table     The aging procedure is performed on the table.
100 Flush table   All nonlocked entries are removed from the table.
101 Clear table   Table is completely cleared.
110 Read entry    The entry pointed to by the MAC Table Index register is read.
111 Write entry   The entry pointed to by the MAC Table Index register is written.
```

A table age/clear/flush runs for around 50 µs. The other commands execute immediately.

Table 163. Mac Table Index - MACTINDX (Address C0h)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
13	Shadow	R/W	Enable MAC table shadow register. With this register set, when reading from bucket 0, the remaining three buckets will be latched into a register for future access.	0
12:11	Bucket	R/W	The bucket selects one of the four entries per table line.	0
10:0	Index	R/W	The index selects one of the 2048 MAC table lines.	0

Note: Used in MAC table read and write operations, where this register selects the entry to read or write.

Table 164. VLAN Table Command - VLANACES (Address D0h)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
27	VLAN Mirror	R/W	Set if all frames in this VLAN are to be mirrored onto the port specified in the Analyzer Config register.	0
26	VLAN Source Check	R/W	Set if frames in this VLAN must be marked in the VLAN Port Mask.	0
13:2	VLAN Port Mask	R/W	Frames in this VLAN may only be sent to ports in this mask.	FFFh
1:0	VLAN Table Command	R/W	VLAN Table Command.	0

Note: This register is used for updating and reading the VLAN table from the CPU. The command selects between different operation and takes on the following values:

```
00   Idle           The previous operation is complete.
01   Read entry     The entry set in VLAN Index is read into this register.
10   Write entry    The entry set in VLAN Index is written.
11   Clear table    The VLAN table is initialized to default values.
```

The Command bits must read as Idle before a new command can be issued. The execution times are as for the MACACCES register.

Table 165. VLAN Table Index - VLANTINDX (Address E0h)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
11:0	Index	R/W	The index selects one of the 4096 VLAN table lines.	0

Note: Used in VLAN table read and write operations, where this register selects the entry to read or write.

Table 166. Analyzer Config Register - AGENCNTL (Address F0h)
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
11	Flood CPUCopy	R/W	Flooded frames will be copied to the CPU.	0
10	Flood FwdKill	R/W	Flooded frames will not be sent to any ports.	0
9	Flood Ignore VLAN	R/W	Flooded frames will not be sensitive to the VLAN masks.	0
8	Mirror CPU	R/W	Frames destined for the CPU capture buffer will also be forwarded to the configured mirror port.	0
7	Learn CPUCopy	R/W	Autolearned stations will get the CPUCopy flag set in their entry in the MAC table.	0
6	Learn FwdKill	R/W	Autolearned stations will get the FwdKill flag set in their entry in the MAC table.	0
5	Learn Ignore VLAN	R/W	Autolearned stations will get the Ignore VLAN flag set in their entry in the MAC table.	0

Table 166. Analyzer Config Register - AGENCNTL (Address F0h) *(continued)*
Block 2 Subblock 0

Bit	Name	Mode	Description	Default
4:0	Mirror Port	R/W	Frames mirrored will be sent to this port.	0

SFR Register Overview

The registers in this section are only accessible from the iCPU, and thereby only valid when the iCPU is enabled by strapping the ICPU_PI_En pin high.

General Purpose I/O SFRs

This group of registers provides access to the iCPU GPIO bits.

Table 167. General Purpose I/O Input - GPIO_IN (Address 80h)

Bit	Name	Mode	Description	Default
3	GI3	R/O	The value of this bit reflects the value of the chip's 8051 GPIO pin #3. When the GPIO for this bit is configured as an output bit, the read value becomes the value of the corresponding output.	0
2	GI2	R/O	See GI3 for a description.	0
1	GI1	R/O	See GI3 for a description.	0
0	GI0	R/O	See GI3 for a description. This bit is also connected to external interrupt #1.	0

Table 168. General Purpose I/O Output - GPIO_OUT (Address 90h)

Bit	Name	Mode	Description	Default
3	GO3	R/W	The value of this bit is presented on the chip’s 8051 GPIO pin #3 when the bit in question is configured as output (see GPIO_OE register).	0
2	GO2	R/W	See GO3 for a description.	0
1	GO1	R/W	See GO3 for a description.	0
0	GO0	R/W	See GO3 for a description.	0

Table 169. General Purpose I/O Output Enable - GPIO_OE (Address A0h)

Bit	Name	Mode	Description	Default
3	GOE3	R/W	Setting this bit enables the chip's 8051 GPIO pin #3 for output. The value specified in the GPIO_OUT::GO3 bit is presented on the pin. When this bit is cleared, the pin is configured as an input pin. 0=Configure 8051 GPIO pin as input (as seen from the chip). 1=Configure 8051 GPIO pin as output (as seen from the chip).	0
2	GOE2	R/W	See GOE3 for a description.	0
1	GOE1	R/W	See GOE3 for a description.	0
0	GOE0	R/W	See GOE3 for a description.	0

Table 170. General Purpose I/O Status - GPIO_STAT (Address A1h)

Bit	Name	Mode	Description	Default
3	GS3	R/W	Whenever GPIO_IN::GI3 changes, this bit gets set. The bit is sticky in the sense that it keeps its value once set. The bit is cleared by writing a '1' to it. Note that this field also senses changes on GPIO_OUT::GO3 when the pin is configured as an output. 0=There have been no events on this GPIO pin. 1=There has been at least one event on this GPIO pin.	0
2	GS2	R/W	See GS3 for a description.	0
1	GS1	R/W	See GS3 for a description.	0
0	GS0	R/W	See GS3 for a description.	0

Dual Data Pointer SFRs

The iCPU employs dual data pointers to accelerate data memory block moves. The standard 8051 data pointer (DPTR) is a 16-bit value used to address external data RAM or peripherals. The iCPU maintains the standard data pointer as DPTR0 at SFR locations 0x82 and 0x83. The iCPU adds a second data pointer (DPTR1) at SFR locations 0x84 and 0x85 and a select register at 0x86 as described below.

Table 171. Data Pointer 1 Low - DPL1 (Address 84h)

Bit	Name	Mode	Description	Default
7:0	VALUE	R/W	Access this register to read or write the value of the least significant byte of DPTR1. This register is used in external data accesses when the DPS::SEL bit is 1.	0

Table 172. Data Pointer 1 High - DPH1 (Address 85h)

Bit	Name	Mode	Description	Default
7:0	VALUE	R/W	Access this register to read or write the value of the most significant byte of DPTR1. This register is used in external data accesses when the DPS::SEL bit is 1.	0

Table 173. Data Pointer Select - DPS (Address 86h)

Bit	Name	Mode	Description	Default
0	SEL	R/W	When this bit is 0, instructions that use the DPTR will use DPL0 and DPH0. When this bit is 1, instructions that use the DPTR will use DPL1 and DPH1. Since no other bits in this SFR are used, the fastest way to toggle this bit is to issue an “INC DPS” instruction.	0

Memory Access Control SFRs

This group of SFRs provides control over accessed memory.

Table 174. Special Function Register - SPC_FNC (Address 8Fh)

Bit	Name	Mode	Description	Default
0	WRS	R/W	External ROM is normally read-only, but can be written to for program downloading by setting this bit. This will cause all “MOVX @DPTR”-like instructions to activate the ICPU_ROM_nCS signal rather than the ICPU_RAM_nCS signal on the external memory bus. 0: When writing to external memory, activate ICPU_RAM_nCS signal. '1': When writing to external memory, activate ICPU_ROM_nCS signal.	0

Table 175. Paging Control - PAGE (Address B0h)

Bit	Name	Mode	Description	Default
7:4	IFP	R/W	Instruction Fetch page bits. These four page bits are presented on the address bus (ICPU_Addr[19:16]) when an instruction fetch is in progress, that is, when both ICPU_ROM_nCS and ICPU_nRD are asserted. This is also the case when the code is read programmatically with MOVC instructions.	0
3:0	OP	R/W	“Other” memory accesses page bits. These four page bits are presented on the address bus (ICPU_Addr[19:16]) when the access is **not** an instruction fetch or MOVC instruction (see also PAGE::IFP above).	0
Note: The paging bits are used to extend the 8051's 16-bit address space to a 20–bit address bus, thus allowing for access to up to 1 Mbytes of external data and code. The user programmatically selects the page to hit by changing the value of the bits in this register. These bits directly map to the ICPU_Addr[19:16] bits.				

Watchdog SFRs

This group of registers provides control over the built-in watchdog.

Table 176. Watchdog Data - WDDA (Address A2h)

Bit	Name	Mode	Description	Default
7:0	DATA	R/W	Alternately write the two values 0xBE and 0xEF to this register to reset the watchdog's 1.073 second-interval timer. After (re-)enabling the watchdog (see WDCON register) the first value written to this field must be 0xBE. When read, the register returns the latest written value. Failing to write the correct value results in an 8051 reset. Failing to write the correct value within 1.072 seconds after the previous write also causes an 8051 reset. A reset caused by the watchdog can be determined by reading the chip's ICPU_CTRL::WATCHDOG_RST bit in the SYSTEM registers block, which is not reset with an 8051 reset.	EFh

Table 177. Watchdog Control - WDCON (Address A3h)

Bit	Name	Mode	Description	Default
0	WD_EN	R/W	Write a '1' to this field to enable the watchdog. Doing so also resets the watchdog timer. The next byte to be written to WDDA is 0xBE. Write a '0' to this field to disable the watchdog.	0

Additional Timer Related SFRs

The third timer (Timer 2) is controlled with these SFRs, as are the count incrementers of Timer 0 and Timer 1. The additional timer also affects the layout of the Interrupt Enable (IE) and Interrupt Priority (IP) registers, whose extra bits are described within here.

Table 178. Timer Clock and External Memory Stretch Cycles Control CKKON (Address 8Eh)

Bit	Name	Mode	Description	Default
5	T2M	R/W	Timer 2 Clock Select. Selects whether Timer 2 increments every 4 or every 12 iCPU clock cycles. The value of this bit has no effect when Timer 2 is used for baud-rate generation. 0=Timer 2 increments every 12 iCPU clock cycles. 1=Timer 2 increments every 4 iCPU clock cycles.	0
4	T1M	R/W	Timer 1 Clock Select. Selects whether Timer 1 increments every 4 or 12 iCPU clock cycles. 0=Timer 1 increments every 12 iCPU clock cycles. 1=Timer 1 increments every 4 iCPU clock cycles.	0
3	T0M	R/W	Timer 0 Clock Select. Selects whether Timer 0 increments every 4 or 12 iCPU clock cycles. 0=Timer 0 increments every 12 iCPU clock cycles. 1=Timer 0 increments every 4 iCPU clock cycles.	0
2:0	MD	R/W	Controls memory stretch cycles. Controls the number of cycles to be used for external MOVX instructions. All types of external accesses except instruction fetches and MOVC instructions are affected by this field. The ICPU_nRD, ICPU_nWR, ICPU_RAM_nCS, and ICPU_ROM_nCS signals are affected by the value of this field. 000=Width is 2 iCPU clock cycles. 001=Width is 4 iCPU clock cycles. 010=Width is 8 iCPU clock cycles. ... 111=Width is 28 iCPU clock cycles.	1h

Table 179. Interrupt Enable - IE (Address A8h)

Bit	Name	Mode	Description	Default
7	EA	R/W	Global Interrupt Enable. See standard 8051 documentation.	0
6	Reserved	R/W		0
5	ET2	R/W	Enable Timer 2 Interrupt. 0=Disable Timer 2 Interrupt. 1=Enable Timer 2 Interrupt generated by T2CON::TF2 flag.	0
4	ES	R/W	Enable Serial Port Interrupt. See standard 8051 documentation.	0
3	ET1	R/W	Enable Timer 1 Interrupt. See standard 8051 documentation.	0
2	EX1	R/W	Enable External Interrupt 1. See standard 8051 documentation.	0
1	ET0	R/W	Enable Timer 0 Interrupt. See standard 8051 documentation.	0
0	EX0	R/W	Enable External Interrupt 0. See standard 8051 documentation.	0

Table 180. Interrupt Priority - IP (Address B8h)

Bit	Name	Mode	Description	Default
7	Reserved	R/W	—	1
6	Reserved	R/W	—	0
5	PT2	R/W	Timer 2 Interrupt Priority Control. 0=Sets Timer 2 interrupt (T2CON::TF2) to low priority. 1=Sets Timer 2 interrupt (T2CON::TF2) to high priority.	0
4	PS	R/W	See standard 8051 documentation.	0
3	PT1	R/W	See standard 8051 documentation.	0
2	PX1	R/W	See standard 8051 documentation.	0
1	PT0	R/W	See standard 8051 documentation.	0
0	PX0	R/W	See standard 8051 documentation.	0

Table 181. Timer 2 Control - T2CON (Address C8h)

Bit	Name	Mode	Description	Default
7	TF2	R/W	Timer 2 overflow flag. Hardware will set TF2 when Timer 2 overflows from 0xFFFF. TF2 must be cleared to 0 by the software. TF2 will only be set to a 1 if RCLK and TCLK are both cleared to 0. Writing a 1 to TF2 forces a Timer 2 interrupt if enabled.	0
6	Reserved	R/W	Always write a 0 to this bit. Writing a 1 may cause unpredictable behavior.	0
5	RCLK	R/W	Receive Clock Flag. Determines whether Timer 1 or Timer 2 is used for the serial port timing of received data in serial mode 1 or 3. 0=Selects Timer 1 overflow as the receive clock. 1=Selects Timer 2 overflow as the receive clock.	0
4	TCLK	R/W	Transmit Clock Flag. Determines whether Timer 1 or Timer 2 is used for the serial port timing of transmitted data in serial mode 1 or 3. 0=Selects Timer 1 overflow as the transmit clock. 1=Selects Timer 2 overflow as the transmit clock.	0
3	Reserved	R/W	Always write a 0 to this bit. Writing a 1 may cause unpredictable behavior.	0
2	TR2	R/W	Timer 2 Run Control Flag. 0=Stops Timer 2. 1=Starts Timer 2.	0
1	Reserved	R/W	Always write a 0 to this bit. Writing a 1 may cause unpredictable behavior.	0
0	Reserved	R/W	Always write a 0 to this bit. Writing a 1 may cause unpredictable behavior.	0

Table 182. Timer 2 Capture Low - RCAP2L (Address CAh)

Bit	Name	Mode	Description	Default
7:0	VALUE	R/W	Least significant byte of reload-value when Timer 2 overflows.	0

Table 183. Timer 2 Capture High - RCAP2H (Address CBh)

Bit	Name	Mode	Description	Default
7:0	VALUE	R/W	Most significant byte of reload-value when Timer 2 overflows.	0

Table 184. Timer 2 Low - TL2 (Address CCh)

Bit	Name	Mode	Description	Default
7:0	VALUE	R/W	Least significant byte of the current value of the 16-bit count.	0

Table 185. Timer 2 High - TH2 (Address CDh)

Bit	Name	Mode	Description	Default
7:0	VALUE	R/W	Most significant byte of the current value of the 16-bit count.	0

Chip Register Access SFRs

This group of registers provides access to the chip's internal registers.

Table 186. Register Access Done - RA_DONE (Address F8h)

Bit	Name	Mode	Description	Default
0	DONE	R/W	When this bit is set, a transaction (read or write) to the chip's registers is complete. The bit is cleared when either of the data byte registers (RA_DA0-3 below) is read, a new request is submitted, or whenever a '1' is written to the bit. Note that after a chip-reset it is necessary to clear the hardware state machine by writing a '1' to this bit. This bit is connected to External Interrupt #0 of the 8051.	0

Table 187. Register Access Block and Subblock - RA_BLK (Address F9h)

Bit	Name	Mode	Description	Default
7:5	BLOCK	R/W	The ID of the block of the chip to access.	0
3:0	SUBBLOCK	R/W	The sub-block number of the chip to access.	0

Table 188. Register Access Read Address - RA_AD_RD (Address FAh)

Bit	Name	Mode	Description	Default
7:0	ADDR	R/W	Address within the block/sub-block specified with RA_BLK to read. Writing this field will initiate a read of a register. Keep polling the RA_DONE::DONE field and wait for it to become 1, then read the returned data from the RA_DAx registers.	0

Table 189. Register Access Write Address - RA_AD_WR (Address FBh)

Bit	Name	Mode	Description	Default
7:0	ADDR	R/W	Address within the block/sub-block specified with RA_BLK to write to. Writing this field will initiate a write to the register with address ADDR. The data to write should be set-up in advance by writing to the RA_DAx registers.	0

Table 190. Register Access Data Byte #0 - RA_DA0 (Address FCh)

Bit	Name	Mode	Description	Default
7:0	DATA	R/W	Read this register to obtain bits 7:0 of the 32–bit data returned from a read, or write bits 7:0 of the 32–bit data to write to a particular chip register. Besides returning data, reading this field also clears the RA_DONE::DONE bit.	0

Table 191. Register Access Data Byte #1 - RA_DA1 (Address FDh)

Bit	Name	Mode	Description	Default
7:0	DATA	R/W	Read this register to obtain bits 15:8 of the 32–bit data returned from a read, or write bits 15:8 of the 32–bit data to write to a particular chip register. Besides returning data, reading this field also clears the RA_DONE::DONE bit.	0

Table 192. Register Access Data Byte #2 - RA_DA2 (Address FEh)

Bit	Name	Mode	Description	Default
7:0	DATA	R/W	Read this register to obtain bits 23:16 of the 32–bit data returned from a read, or write bits 23:16 of the 32–bit data to write to a particular chip register. Besides returning data, reading this field also clears the RA_DONE::DONE bit.	0

Table 193. Register Access Data Byte #3 - RA_DA3 (Address FFh)

Bit	Name	Mode	Description	Default
7:0	DATA	R/W	Read this register to obtain bits 31:24 of the 32–bit data returned from a read, or write bits 31:24 of the 32–bit data to write to a particular chip register. Besides returning data, reading this field also clears the RA_DONE::DONE bit.	0

Other SFRs

Table 194. Program Status Word - PSW (Address D0h)

Bit	Name	Mode	Description	Default
7	CY	R/W	See standard 8051 documentation.	0
6	AC	R/W	See standard 8051 documentation.	0
5	F0	R/W	See standard 8051 documentation.	0
4	RS1	R/W	See standard 8051 documentation.	0
3	RS0	R/W	See standard 8051 documentation.	0
2	OV	R/W	See standard 8051 documentation.	0
1	F1	R/W	User Flag 1. Bit-addressable, general purpose flag for software control.	0
0	P	R/O	See standard 8051 documentation.	0

Signal Description

To distinguish the signals from each other and to which interface each belongs, most of the signals have the interface name appended as prefix. Additionally, an active low signal is denoted with a lower case ‘n’ as a prefix on the actual ‘name’, not interface name (for example: PI_nOE).

Signal List By Function

The notation in Table 195 has been used to define the pin types.

Table 195. Pin Type Definitions

Pin Type	Definition
I	Input only
O	Output only
I/O	Bidirectional
OZ	3-state output
A	Analog
Power	Power
GND	Ground
NC	No connection, do not connect
3 V	3.3 V tolerant
5 V	5 V tolerant

Clock Circuits

Table 196. Clk Interface

Signal Name	Type	Description
PLL_Cap0 PLL_Cap1	A	PLL Loop filter capacitor. Connect 100 nF capacitor between the pins.
Clk	I, 3 V	System reference clock, LVTTL input. It can be either 25 MHz or 125 MHz, selectable by the Clk125_En signal.
Clk125_En	I, 3 V	0=25 MHz clock select 1=125 MHz clock select
PLL_En	I, 3 V	Internal test. Pull high to VDD_IO25 or VDD_OUT33

RGMII Interface

Table 197 contains a complete description of the RGMII ports, when used in the different modes.

Table 197. RGMII Ports

Signal Name	Type	Description
RGMII[11:0]_RD0 RGMII[11:0]_RD1 RGMII[11:0]_RD2 RGMII[11:0]_RD3	I	RGMII mode (10/100 Mbit): Receive data input. Contains bit [3:0] on the rising edge of the Rx_Clk. Data is only sampled on the rising edge, not on the falling. RGMII mode (1000 Mbit): Multiplexed receive data input. Contains bits [3:0] on the rising edge of the Rx_Clk and bit, [7:4], on the falling edge.
RGMII[11:0]_Rx_Clk	I	Receive clock. This clock is used to synchronize the receive data and control.
RGMII[11:0]_Rx_Ctrl	I	RGMII mode: Data valid and receive error input. On the rising edge of the Rx_Clk, this input serves as data valid signaling valid data from the PHY and is available on RD[3:0]. On the falling edge of Rx_Clk it contains a logical derivative of data valid and receive error from the PHY (see RGMII standard).
RGMII[11:0]_TD0 RGMII[11:0]_TD1 RGMII[11:0]_TD2 RGMII[11:0]_TD3	O	RGMII mode (10/100 Mbit): Transmit data output. Contains bit [3:0] on the rising edge of the Tx_Clk. No data change occur on the falling edge of Tx_Clk. RGMII mode (1000 Mbit): Multiplexed transmit data output. Contains bit [3:0] on the rising edge of the Tx_Clk and bit [7:4] on the falling edge.
RGMII[11:0]_Tx_Clk	O	Transmit clock. This clock is continuously driven from the MAC, and transmit data and control are synchronized to it.
RGMII[11:0]_Tx_Ctrl	O	RGMII mode: Transmit control. On the rising edge of Tx_Clk it serves as transmit enable indicating valid data on TD[3:0]. On the falling edge it contains a logical derivative based on transmit enable and error from the MAC (see RGMII standard).

Power Supply and Ground Pins

Table 198. Power and Ground

Signal Name	Type	Description
VDD	Power	1.8 V core supply
VDD_IO25	Power	2.5 V I/O supply
VDD_OUT33	Power	3.3 V output supply
VSS	GND	Common ground
VDD_PLL	Power	2.5 V supply to internal PLL
VSS_PLL	GND	Internal PLL ground

JTAG Interface

Table 199. JTAG Interface

Signal Name	Type	Description
JTAG_nTRST	I, 5 V	JTAG Test Reset, active low. For normal operation JTAG_nTRST should be pulled low to VSS.
JTAG_TCK	I, 5 V	JTAG Clock. For normal operation, JTAG_TCK should be pulled high to VDD_OUT33.
JTAG_TDI	I, 5 V	JTAG Test Data In. For normal operation, JTAG_TDI should be pulled high to VDD_OUT33.
JTAG_TDO	OZ, 3 V	JTAG Test Data Out.
JTAG_TMS	I, 5 V	JTAG Test Mode Select. For normal operation, JTAG_TMS should be pulled high to VDD_OUT33.

iCPU External Memory Interface

The information provided in Table 200 only applies if the iCPU enabled, that is, if the ICPU_PI_En pin is strapped high. The iCPU signals are multiplexed on top of the PI signals. See Table 202 for reference.

Table 200. iCPU External Memory Interface

Signal Name	Type	Description
ICPU_Addr0 ICPU_Addr1 ICPU_Addr2 ICPU_Addr3 ICPU_Addr4 ICPU_Addr5 ICPU_Addr6 ICPU_Addr7 ICPU_Addr8 ICPU_Addr9 ICPU_Addr10 ICPU_Addr11 ICPU_Addr12 ICPU_Addr13 ICPU_Addr14 ICPU_Addr15	O	Least significant bit iCPUs address interface to external memories. Most significant bit
ICPU_Data0 ICPU_Data1 ICPU_Data2 ICPU_Data3 ICPU_Data4 ICPU_Data5 ICPU_Data6 ICPU_Data7	I/O, 3 V	Least significant bit iCPUs data interface to external memories. Driven by Stansted on iCPU-writes to external memories, driven by external memories on iCPU-reads. Most significant bit
ICPU_Addr16 ICPU_Addr17 ICPU_Addr18 ICPU_Addr19	O	Least significant bit Upper four bits of address, controlled with paging. Most significant bit
ICPU_nWR	O	Active low signal that selects write of external memory.
ICPU_nRD	O	Active low signal that selects read of external memory.
ICPU_ROM_ nCS	O	Active low signal that selects external Flash/ROM.
ICPU_RAM_ nCS	O	Active low signal that selects external RAM.
ICPU_PI_En	I, 3V	Enable of iCPU or Parallel CPU Interface. Pull high to VDD_OUT33 to enable the iCPU or pull low to VSS to enable the Parallel CPU Interface.

iCPU GPIO and RS232 Interface

The information provided in Table 201 only applies if the iCPU enabled, that is, if the ICPU_PI_En pin is strapped high.

Table 201. iCPU GPIO and RS232 Interface

Signal Name	Type	Description
ICPU_GPIO0 ICPU_GPIO1 ICPU_GPIO2 ICPU_GPIO3	I/O, 3 V	Additional General Purpose I/Os only accessible from the iCPU. Pull high to VDD_IO25 or low to VSS if unused.
ICPU_TxD	O	Serial data out of chip.
ICPU_RxD	I, 3 V	Serial data in to chip.

iCPU Pin Mapping

Table 202 shows how the iCPU pins map on to the PI interface.

Table 202. iCPU Pin Mapping to the PI Interface

iCPU Signal Name	PI Signal Name
ICPU_GPIO3	PI_Data15
ICPU_GPIO2	PI_Data14
ICPU_GPIO1	PI_Data13
ICPU_GPIO0	PI_Data12
ICPU_Addr19	PI_Data11
ICPU_Addr18	PI_Data10
ICPU_Addr17	PI_Data9
ICPU_Addr16	PI_Data8
ICPU_Addr15	PI_Addr15
ICPU_Addr14	PI_Addr14
ICPU_Addr13	PI_Addr13
ICPU_Addr12	PI_Addr12
ICPU_Addr11	PI_Addr11
ICPU_Addr10	PI_Addr10
ICPU_Addr9	PI_Addr9
ICPU_Addr8	PI_Addr8
ICPU_Addr7	PI_Addr7
ICPU_Addr6	PI_Addr6
ICPU_Addr5	PI_Addr5

Table 202. iCPU Pin Mapping to the PI Interface *(continued)*

iCPU Signal Name	PI Signal Name
ICPU_Addr4	PI_Addr4
ICPU_Addr3	PI_Addr3
ICPU_Addr2	PI_Addr2
ICPU_Addr1	PI_Addr1
ICPU_Addr0	PI_Addr0
ICPU_Data7	PI_Data7
ICPU_Data6	PI_Data6
ICPU_Data5	PI_Data5
ICPU_Data4	PI_Data4
ICPU_Data3	PI_Data3
ICPU_Data2	PI_Data2
ICPU_Data1	PI_Data1
ICPU_Data0	PI_Data0
ICPU_nWR	PI_nWR
ICPU_nRD	PI_nOE
ICPU_RAM_nCS	PI_nDone
ICPU_ROM_nCS	PI_nCS

Parallel CPU Interface (PI)

The information provided in Table 203 only applies if the Parallel Interface is enabled, that is, if the ICPU_PI_En pin is strapped low.

Table 203. PI Interface

Signal Name	Type	Description
PI_Addr0 PI_Addr1 PI_Addr2 PI_Addr3 PI_Addr4 PI_Addr5 PI_Addr6 PI_Addr7 PI_Addr8 PI_Addr9 PI_Addr10 PI_Addr11 PI_Addr12 PI_Addr13 PI_Addr14 PI_Addr15	I, 3 V	Least significant bit Parallel CPU interface address bus Selects the block, subblock, and address. Refer to the "Register Addressing" on page 78 for a description of the address space. Most significant bit
PI_Data0 PI_Data1 PI_Data2 PI_Data3 PI_Data4 PI_Data5 PI_Data6 PI_Data7 PI_Data8 PI_Data9 PI_Data10 PI_Data11 PI_Data12 PI_Data13 PI_Data14 PI_Data15	I/O, 3 V	Least significant bit Parallel CPU interface data bus Driven by CPU at write, Stansted at read. Most significant bit
PI_IRQ	O	A configurable interrupt signal for various events. The interrupt signal's polarity can be programmed. The Interrupt pin is not an open drain output and should not be wire-ORed to other pins.
PI_nCS	I, 3 V	Start a CPU operation.
PI_nDone	OZ, 3 V	Acknowledges an operation. Programmable polarity.
PI_nOE	I, 3 V	Enable drive of the data bus from Stansted.
PI_nWR	I, 3 V	Selects read (1) or write (0).

Serial Interface

Table 204. SI Interface

Signal Name	Type	Description
SI_Clk	I, 3 V	SI clock from the master.
SI_DI	I, 3 V	Serial input from the master.
SI_DO	OZ, 3 V	Serial output to the master.
SI_nEn	I, 3 V	0 = enable SI interface. 1 = disable SI interface.

MII Management Interface

Table 205. MII Management Interface

Signal Name	Type	Description
MDIO	I/O, 3 V	Management data input/output. MDIO is a bidirectional signal between the PHY and the Stansted device that transfers control and status information. Control information is driven by the device synchronously with respect to MDC and is sampled synchronously by the PHY. Status information is driven by the PHY synchronously with respect to MDC and is sampled synchronously by Stansted.
MDC	O	Management data clock. MDC is sourced by the Station Management entity to the PHY as the timing reference for transfer of information on the MDIO signal. MDC is an aperiodic signal.

General Purpose I/Os

Table 206. GPIO Signals

Signal Name	Type	Description
GPIO0 GPIO1 GPIO2 GPIO3 GPIO4	I/O, 3 V	General purpose I/O. Pull high to VDD_IO25 or VDD_OUT33 or pull low to VSS if unused.

Miscellaneous

Table 207. Miscellaneous Signals

Signal Name	Type	Description
nReset	I, 3 V	Global chip reset, active low
Test_Enable	I, 3 V	Internal test, pull low to VSS
Reserved_0	NC	Leave floating
Reserved_2	I, 3 V	Internal test, pull high to VDD_IO25 or VDD_OUT33
Reserved_1	I, 3 V	Internal test, pull low to VSS

Signal List by Ball Number

Table 208. Signal List by Ball Number

Ball	Signal Name	Ball	Signal Name	Ball	Signal Name
A1	VSS_0	B12	RGMII1_Rx_Ctrl	D1	RGMII4_TD1
A2	VSS_1	B13	VDD_IO25_2	D2	RGMII4_TD0
A3	RGMII3_RD3	B14	RGMII1_TD3	D3	VSS_14
A4	RGMII3_RD2	B15	RGMII0_RD3	D4	VSS_15
A5	RGMII3_RD0	B16	RGMII0_RD0	D5	RGMII3_Tx_Clk
A6	RGMII3_TD3	B17	VDD_IO25_3	D6	RGMII3_TD2
A7	RGMII3_TD0	B18	RGMII0_TD1	D7	RGMII2_Rx_Clk
A8	RGMII2_RD0	B19	PLL_Cap0	D8	RGMII2_RD2
A9	RGMII2_TD3	B20	VDD_PLL	D9	RGMII2_Tx_Ctrl
A10	RGMII1_Rx_Clk	B21	Reserved_0	D10	RGMII2_TD1
A11	VSS_2	B22	Reserved_1	D11	VSS_16
A12	RGMII1_RD1	C1	RGMII3_Rx_Clk	D12	RGMII1_RD3
A13	RGMII1_RD0	C2	VSS_6	D13	RGMII1_Tx_Clk
A14	RGMII1_TD1	C3	VSS_7	D14	RGMII1_TD2
A15	RGMII1_TD0	C4	VSS_8	D15	RGMII0_Rx_Clk
A16	RGMII0_RD1	C5	RGMII3_Tx_Ctrl	D16	RGMII0_RD2
A17	RGMII0_TD3	C6	VSS_9	D17	RGMII0_Tx_Ctrl
A18	RGMII0_TD2	C7	RGMII2_RD3	D18	VSS_17
A19	VSS_PLL	C8	VSS_10	D19	VSS_18
A20	PLL_Cap1	C9	VDD_IO25_4	D20	Test_Enable
A21	Clk	C10	RGMII2_TD2	D21	nReset
A22	VSS_3	C11	VSS_11	D22	JTAG_TMS
B1	RGMII3_Rx_Ctrl	C12	RGMII1_RD2	E1	RGMII4_Tx_Ctrl
B2	VSS_4	C13	RGMII1_Tx_Ctrl	E2	VDD_IO25_5
B3	VDD_IO25_0	C14	VSS_12	E3	RGMII4_TD2
B4	RGMII3_RD1	C15	RGMII0_Rx_Ctrl	E4	RGMII4_TD3
B5	VDD_IO25_1	C16	VSS_13	E5	VSS_19
B6	RGMII3_TD1	C17	RGMII0_Tx_Clk	E6	VDD_IO25_6
B7	RGMII2_Rx_Ctrl	C18	RGMII0_TD0	E7	VDD_IO25_7
B8	RGMII2_RD1	C19	PLL_En	E8	VDD_IO25_8
B9	RGMII2_Tx_Clk	C20	Clk125_En	E9	VDD_IO25_9
B10	RGMII2_TD0	C21	VDD_OUT33_0	E10	VDD_IO25_10
B11	VSS_5	C22	Reserved_2	E11	VSS_20

Table 208. Signal Name by Ball Number *(continued)*

Ball	Signal Name	Ball	Signal Name	Ball	Signal Name
E12	VSS_21	G2	RGMII4_Rx_Ctrl	J4	RGMII5_Tx_Ctrl
E13	VDD_IO25_11	G3	RGMII4_RD2	J5	VDD_IO25_23
E14	VDD_IO25_12	G4	RGMII4_RD3	J6	VDD_15
E15	VDD_IO25_13	G5	VDD_IO25_18	J8	VDD_16
E16	VDD_IO25_14	G6	VDD_7	J9	VDD_17
E17	VDD_IO25_15	G17	VDD_IO25_19	J10	VSS_34
E18	VSS_22	G18	GPIO0	J11	VSS_35
E19	VSS_23	G19	PI_Addr15	J12	VSS_36
E20	JTAG_nTRST	G20	VSS_29	J13	VSS_37
E21	JTAG_TDI	G21	GPIO1	J14	VDD_18
E22	JTAG_TCK	G22	VDD_OUT33_2	J15	VDD_19
F1	RGMII4_Tx_Clk	H1	RGMII5_TD1	J17	VDD_IO25_24
F2	VSS_24	H2	RGMII5_TD0	J18	VDD_OUT33_3
F3	RGMII4_RD0	H3	VSS_30	J19	PI_Addr8
F4	RGMII4_RD1	H4	RGMII5_TD2	J20	VDD_OUT33_4
F5	VDD_IO25_16	H5	VDD_IO25_20	J21	PI_Addr9
F6	VSS_25	H6	VDD_8	J22	PI_Addr10
F7	VDD_0	H8	VDD_9	K1	RGMII5_RD0
F8	VDD_1	H9	VDD_10	K2	RGMII5_RD1
F9	VDD_2	H10	VDD_11	K3	RGMII5_RD2
F10	VSS_26	H11	VSS_31	K4	RGMII5_RD3
F11	VSS_27	H12	VSS_32	K5	VSS_38
F12	VSS_28	H13	VDD_12	K6	VSS_39
F13	VDD_3	H14	VDD_13	K8	VDD_20
F14	VDD_4	H15	VDD_14	K9	VSS_40
F15	VDD_5	H17	VDD_IO25_21	K10	VSS_41
F16	VDD_6	H18	VSS_33	K11	VSS_42
F17	VDD_IO25_17	H19	PI_Addr12	K12	VSS_43
F18	VDD_OUT33_1	H20	PI_Addr11	K13	VSS_44
F19	GPIO4	H21	PI_Addr13	K14	VSS_45
F20	GPIO3	H22	PI_Addr14	K15	VDD_21
F21	GPIO2	J1	RGMII5_TD3	K17	VDD_IO25_25
F22	JTAG_TDO	J2	VDD_IO25_22	K18	VSS_46
G1	RGMII4_Rx_Clk	J3	RGMII5_Tx_Clk	K19	PI_Addr5

Table 208. Signal Name by Ball Number *(continued)*

Ball	Signal Name	Ball	Signal Name	Ball	Signal Name
K20	PI_Addr4	M13	VSS_68	P6	VDD_26
K21	PI_Addr6	M14	VSS_69	P8	VDD_27
K22	PI_Addr7	M15	VSS_70	P9	VDD_28
L1	VSS_47	M17	VDD_IO25_28	P10	VSS_79
L2	VSS_48	M18	VSS_71	P11	VSS_80
L3	VSS_49	M19	PI_nWR	P12	VSS_81
L4	VSS_50	M20	PI_nCS	P13	VSS_82
L5	VSS_51	M21	VDD_OUT33_5	P14	VDD_29
L6	VSS_52	M22	PI_nOE	P15	VDD_30
L8	VSS_53	N1	RGMII6_TD2	P17	VDD_31
L9	VSS_54	N2	VDD_IO25_29	P18	VSS_83
L10	VSS_55	N3	RGMII6_TD3	P19	PI_Data11
L11	VSS_56	N4	RGMII6_Tx_Ctrl	P20	PI_Data10
L12	VSS_57	N5	VDD_IO25_30	P21	PI_Data12
L13	VSS_58	N6	VDD_22	P22	PI_Data13
L14	VSS_59	N8	VDD_23	R1	VSS_84
L15	VSS_60	N9	VSS_72	R2	RGMII6_RD2
L17	VDD_IO25_26	N10	VSS_73	R3	RGMII6_RD3
L18	VDD_IO25_27	N11	VSS_74	R4	RGMII6_Rx_Ctrl
L19	PI_Addr1	N12	VSS_75	R5	VDD_IO25_32
L20	PI_Addr0	N13	VSS_76	R6	VDD_32
L21	PI_Addr2	N14	VSS_77	R8	VDD_33
L22	PI_Addr3	N15	VDD_24	R9	VDD_34
M1	RGMII5_Rx_Clk	N17	VDD_25	R10	VDD_35
M2	RGMII5_Rx_Ctrl	N18	VDD_OUT33_6	R11	VSS_85
M3	RGMII6_TD0	N19	PI_Data15	R12	VSS_86
M4	RGMII6_TD1	N20	PI_Data14	R13	VDD_36
M5	VSS_61	N21	PI_nDone	R14	VDD_37
M6	VSS_62	N22	PI_IRQ	R15	VDD_38
M8	VSS_63	P1	RGMII6_RD1	R17	VDD_39
M9	VSS_64	P2	RGMII6_RD0	R18	VDD_OUT33_7
M10	VSS_65	P3	RGMII6_Tx_Clk	R19	PI_Data7
M11	VSS_66	P4	VSS_78	R20	VDD_OUT33_8
M12	VSS_67	P5	VDD_IO25_31	R21	PI_Data8

Table 208. Signal Name by Ball Number *(continued)*

Ball	Signal Name	Ball	Signal Name	Ball	Signal Name
R22	PI_Data9	U22	PI_Data3	W12	RGMII9_Rx_Clk
T1	RGMII7_TD1	V1	RGMII7_Tx_Clk	W13	RGMII10_Tx_Ctrl
T2	RGMII6_Rx_Clk	V2	RGMII7_RD0	W14	RGMII10_Tx_Clk
T3	VSS_87	V3	RGMII7_RD2	W15	RGMII10_Rx_Ctrl
T4	RGMII7_TD0	V4	RGMII7_RD1	W16	RGMII11_TD2
T5	VDD_IO25_33	V5	VSS_94	W17	RGMII11_Tx_Ctrl
T6	VDD_40	V6	VDD_IO25_36	W18	VSS_101
T17	VDD_41	V7	VDD_IO25_37	W19	VSS_102
T18	VSS_88	V8	VDD_IO25_38	W20	SI_DI
T19	PI_Data5	V9	VDD_IO25_39	W21	SI_nEn
T20	PI_Data4	V10	VDD_IO25_40	W22	ICPU_PI_En
T21	PI_Data6	V11	VSS_95	Y1	RGMII7_Rx_Clk
T22	VDD_OUT33_9	V12	VSS_96	Y2	VSS_103
U1	VDD_IO25_34	V13	VDD_IO25_41	Y3	VSS_104
U2	RGMII7_TD2	V14	VDD_IO25_42	Y4	VSS_105
U3	RGMII7_TD3	V15	VDD_IO25_43	Y5	RGMII8_Tx_Ctrl
U4	RGMII7_Tx_Ctrl	V16	VDD_IO25_44	Y6	RGMII8_RD0
U5	VDD_IO25_35	V17	VDD_IO25_45	Y7	RGMII8_RD3
U6	VSS_89	V18	VSS_97	Y8	VSS_106
U7	VDD_42	V19	SI_DO	Y9	RGMII9_Tx_Clk
U8	VDD_43	V20	ICPU_RxD	Y10	RGMII9_RD1
U9	VDD_44	V21	VDD_OUT33_11	Y11	VSS_107
U10	VSS_90	V22	ICPU_TxD	Y12	RGMII9_Rx_Ctrl
U11	VSS_91	W1	RGMII7_RD3	Y13	RGMII10_TD3
U12	VSS_92	W2	RGMII7_Rx_Ctrl	Y14	VSS_108
U13	VDD_45	W3	VDD_IO25_46	Y15	RGMII10_Rx_Clk
U14	VDD_46	W4	VSS_98	Y16	VSS_109
U15	VDD_47	W5	VSS_99	Y17	RGMII11_Tx_Clk
U16	VDD_48	W6	RGMII8_Tx_Clk	Y18	RGMII11_RD2
U17	VSS_93	W7	RGMII8_Rx_Ctrl	Y19	VSS_110
U18	VDD_OUT33_10	W8	RGMII9_TD2	Y20	VSS_111
U19	PI_Data1	W9	RGMII9_Tx_Ctrl	Y21	MDIO
U20	PI_Data0	W10	RGMII9_RD0	Y22	SI_Clk
U21	PI_Data2	W11	VSS_100	AA1	VSS_112

Table 208. Signal Name by Ball Number *(continued)*

Ball	Signal Name
AA2	VSS_113
AA3	VSS_114
AA4	RGMII8_TD1
AA5	RGMII8_TD3
AA6	VSS_115
AA7	RGMII8_RD2
AA8	RGMII9_TD1
AA9	RGMII9_TD3
AA10	RGMII9_RD2
AA11	VSS_116
AA12	RGMII10_TD0
AA13	RGMII10_TD2
AA14	RGMII10_RD0
AA15	RGMII10_RD3
AA16	RGMII11_TD1
AA17	RGMII11_TD3
AA18	RGMII11_RD1
AA19	RGMII11_Rx_Ctrl
AA20	VDD_IO25_47
AA21	VSS_117
AA22	MDC
AB1	VSS_118
AB2	VDD_IO25_48
AB3	RGMII8_TD0
AB4	RGMII8_TD2
AB5	VDD_IO25_49
AB6	RGMII8_RD1
AB7	RGMII8_Rx_Clk
AB8	RGMII9_TD0
AB9	VDD_IO25_50
AB10	RGMII9_RD3
AB11	VSS_119
AB12	RGMII10_TD1
AB13	VDD_IO25_51
AB14	RGMII10_RD1
AB15	RGMII10_RD2
AB16	RGMII11_TD0
AB17	VDD_IO25_52
AB18	RGMII11_RD0
AB19	RGMII11_RD3
AB20	RGMII11_Rx_Clk
AB21	VSS_120
AB22	VSS_121

Signal List by Signal Name

Table 209. Signal List by Signal Name

Signal Name	Ball	Signal Name	Ball	Signal Name	Ball
Clk	A21	PI_Addr14	H22	RGMII0_RD3	B15
Clk125_En	C20	PI_Addr15	G19	RGMII0_Rx_Clk	D15
GPIO0	G18	PI_Data0	U20	RGMII0_Rx_Ctrl	C15
GPIO1	G21	PI_Data1	U19	RGMII0_TD0	C18
GPIO2	F21	PI_Data2	U21	RGMII0_TD1	B18
GPIO3	F20	PI_Data3	U22	RGMII0_TD2	A18
GPIO4	F19	PI_Data4	T20	RGMII0_TD3	A17
ICPU_PI_En	W22	PI_Data5	T19	RGMII0_Tx_Clk	C17
ICPU_RxD	V20	PI_Data6	T21	RGMII0_Tx_Ctrl	D17
ICPU_TxD	V22	PI_Data7	R19	RGMII1_RD0	A13
JTAG_nTRST	E20	PI_Data8	R21	RGMII1_RD1	A12
JTAG_TCK	E22	PI_Data9	R22	RGMII1_RD2	C12
JTAG_TDI	E21	PI_Data10	P20	RGMII1_RD3	D12
JTAG_TDO	F22	PI_Data11	P19	RGMII1_Rx_Clk	A10
JTAG_TMS	D22	PI_Data12	P21	RGMII1_Rx_Ctrl	B12
MDC	AA22	PI_Data13	P22	RGMII1_TD0	A15
MDIO	Y21	PI_Data14	N20	RGMII1_TD1	A14
nReset	D21	PI_Data15	N19	RGMII1_TD2	D14
PI_Addr0	L20	PI_IRQ	N22	RGMII1_TD3	B14
PI_Addr1	L19	PI_nCS	M20	RGMII1_Tx_Clk	D13
PI_Addr2	L21	PI_nDone	N21	RGMII1_Tx_Ctrl	C13
PI_Addr3	L22	PI_nOE	M22	RGMII2_RD0	A8
PI_Addr4	K20	PI_nWR	M19	RGMII2_RD1	B8
PI_Addr5	K19	PLL_Cap0	B19	RGMII2_RD2	D8
PI_Addr6	K21	PLL_Cap1	A20	RGMII2_RD3	C7
PI_Addr7	K22	PLL_En	C19	RGMII2_Rx_Clk	D7
PI_Addr8	J19	Reserved_0	B21	RGMII2_Rx_Ctrl	B7
PI_Addr9	J21	Reserved_1	B22	RGMII2_TD0	B10
PI_Addr10	J22	Reserved_2	C22	RGMII2_TD1	D10
PI_Addr11	H20	RGMII0_RD0	B16	RGMII2_TD2	C10
PI_Addr12	H19	RGMII0_RD1	A16	RGMII2_TD3	A9
PI_Addr13	H21	RGMII0_RD2	D16	RGMII2_Tx_Clk	B9

Table 209. Signal List by Signal Name *(continued)*

Signal Name	Ball	Signal Name	Ball	Signal Name	Ball
RGMII2_Tx_Ctrl	D9	RGMII5_TD3	J1	RGMII8_TD1	AA4
RGMII3_RD0	A5	RGMII5_Tx_Clk	J3	RGMII8_TD2	AB4
RGMII3_RD1	B4	RGMII5_Tx_Ctrl	J4	RGMII8_TD3	AA5
RGMII3_RD2	A4	RGMII6_RD0	P2	RGMII8_Tx_Clk	W6
RGMII3_RD3	A3	RGMII6_RD1	P1	RGMII8_Tx_Ctrl	Y5
RGMII3_Rx_Clk	C1	RGMII6_RD2	R2	RGMII9_RD0	W10
RGMII3_Rx_Ctrl	B1	RGMII6_RD3	R3	RGMII9_RD1	Y10
RGMII3_TD0	A7	RGMII6_Rx_Clk	T2	RGMII9_RD2	AA10
RGMII3_TD1	B6	RGMII6_Rx_Ctrl	R4	RGMII9_RD3	AB10
RGMII3_TD2	D6	RGMII6_TD0	M3	RGMII9_Rx_Clk	W12
RGMII3_TD3	A6	RGMII6_TD1	M4	RGMII9_Rx_Ctrl	Y12
RGMII3_Tx_Clk	D5	RGMII6_TD2	N1	RGMII9_TD0	AB8
RGMII3_Tx_Ctrl	C5	RGMII6_TD3	N3	RGMII9_TD1	AA8
RGMII4_RD0	F3	RGMII6_Tx_Clk	P3	RGMII9_TD2	W8
RGMII4_RD1	F4	RGMII6_Tx_Ctrl	N4	RGMII9_TD3	AA9
RGMII4_RD2	G3	RGMII7_RD0	V2	RGMII9_Tx_Clk	Y9
RGMII4_RD3	G4	RGMII7_RD1	V4	RGMII9_Tx_Ctrl	W9
RGMII4_Rx_Clk	G1	RGMII7_RD2	V3	RGMII10_RD0	AA14
RGMII4_Rx_Ctrl	G2	RGMII7_RD3	W1	RGMII10_RD1	AB14
RGMII4_TD0	D2	RGMII7_Rx_Clk	Y1	RGMII10_RD2	AB15
RGMII4_TD1	D1	RGMII7_Rx_Ctrl	W2	RGMII10_RD3	AA15
RGMII4_TD2	E3	RGMII7_TD0	T4	RGMII10_Rx_Clk	Y15
RGMII4_TD3	E4	RGMII7_TD1	T1	RGMII10_Rx_Ctrl	W15
RGMII4_Tx_Clk	F1	RGMII7_TD2	U2	RGMII10_TD0	AA12
RGMII4_Tx_Ctrl	E1	RGMII7_TD3	U3	RGMII10_TD1	AB12
RGMII5_RD0	K1	RGMII7_Tx_Clk	V1	RGMII10_TD2	AA13
RGMII5_RD1	K2	RGMII7_Tx_Ctrl	U4	RGMII10_TD3	Y13
RGMII5_RD2	K3	RGMII8_RD0	Y6	RGMII10_Tx_Clk	W14
RGMII5_RD3	K4	RGMII8_RD1	AB6	RGMII10_Tx_Ctrl	W13
RGMII5_Rx_Clk	M1	RGMII8_RD2	AA7	RGMII11_RD0	AB18
RGMII5_Rx_Ctrl	M2	RGMII8_RD3	Y7	RGMII11_RD1	AA18
RGMII5_TD0	H2	RGMII8_Rx_Clk	AB7	RGMII11_RD2	Y18
RGMII5_TD1	H1	RGMII8_Rx_Ctrl	W7	RGMII11_RD3	AB19
RGMII5_TD2	H4	RGMII8_TD0	AB3	RGMII11_Rx_Clk	AB20

Table 209. Signal List by Signal Name *(continued)*

Signal Name	Ball	Signal Name	Ball	Signal Name	Ball
RGMII11_Rx_Ctrl	AA19	VDD_22	N6	VDD_IO25_7	E7
RGMII11_TD0	AB16	VDD_23	N8	VDD_IO25_8	E8
RGMII11_TD1	AA16	VDD_24	N15	VDD_IO25_9	E9
RGMII11_TD2	W16	VDD_25	N17	VDD_IO25_10	E10
RGMII11_TD3	AA17	VDD_26	P6	VDD_IO25_11	E13
RGMII11_Tx_Clk	Y17	VDD_27	P8	VDD_IO25_12	E14
RGMII11_Tx_Ctrl	W17	VDD_28	P9	VDD_IO25_13	E15
SI_Clk	Y22	VDD_29	P14	VDD_IO25_14	E16
SI_DI	W20	VDD_30	P15	VDD_IO25_15	E17
SI_DO	V19	VDD_31	P17	VDD_IO25_16	F5
SI_nEn	W21	VDD_32	R6	VDD_IO25_17	F17
Test_Enable	D20	VDD_33	R8	VDD_IO25_18	G5
VDD_0	F7	VDD_34	R9	VDD_IO25_19	G17
VDD_1	F8	VDD_35	R10	VDD_IO25_20	H5
VDD_2	F9	VDD_36	R13	VDD_IO25_21	H17
VDD_3	F13	VDD_37	R14	VDD_IO25_22	J2
VDD_4	F14	VDD_38	R15	VDD_IO25_23	J5
VDD_5	F15	VDD_39	R17	VDD_IO25_24	J17
VDD_6	F16	VDD_40	T6	VDD_IO25_25	K17
VDD_7	G6	VDD_41	T17	VDD_IO25_26	L17
VDD_8	H6	VDD_42	U7	VDD_IO25_27	L18
VDD_9	H8	VDD_43	U8	VDD_IO25_28	M17
VDD_10	H9	VDD_44	U9	VDD_IO25_29	N2
VDD_11	H10	VDD_45	U13	VDD_IO25_30	N5
VDD_12	H13	VDD_46	U14	VDD_IO25_31	P5
VDD_13	H14	VDD_47	U15	VDD_IO25_32	R5
VDD_14	H15	VDD_48	U16	VDD_IO25_33	T5
VDD_15	J6	VDD_IO25_0	B3	VDD_IO25_34	U1
VDD_16	J8	VDD_IO25_1	B5	VDD_IO25_35	U5
VDD_17	J9	VDD_IO25_2	B13	VDD_IO25_36	V6
VDD_18	J14	VDD_IO25_3	B17	VDD_IO25_37	V7
VDD_19	J15	VDD_IO25_4	C9	VDD_IO25_38	V8
VDD_20	K8	VDD_IO25_5	E2	VDD_IO25_39	V9
VDD_21	K15	VDD_IO25_6	E6	VDD_IO25_40	V10

Table 209. Signal List by Signal Name *(continued)*

Signal Name	Ball	Signal Name	Ball	Signal Name	Ball
VDD_IO25_41	V13	VSS_9	C6	VSS_43	K12
VDD_IO25_42	V14	VSS_10	C8	VSS_44	K13
VDD_IO25_43	V15	VSS_11	C11	VSS_45	K14
VDD_IO25_44	V16	VSS_12	C14	VSS_46	K18
VDD_IO25_45	V17	VSS_13	C16	VSS_47	L1
VDD_IO25_46	W3	VSS_14	D3	VSS_48	L2
VDD_IO25_47	AA20	VSS_15	D4	VSS_49	L3
VDD_IO25_48	AB2	VSS_16	D11	VSS_50	L4
VDD_IO25_49	AB5	VSS_17	D18	VSS_51	L5
VDD_IO25_50	AB9	VSS_18	D19	VSS_52	L6
VDD_IO25_51	AB13	VSS_19	E5	VSS_53	L8
VDD_IO25_52	AB17	VSS_20	E11	VSS_54	L9
VDD_OUT33_0	C21	VSS_21	E12	VSS_55	L10
VDD_OUT33_1	F18	VSS_22	E18	VSS_56	L11
VDD_OUT33_2	G22	VSS_23	E19	VSS_57	L12
VDD_OUT33_3	J18	VSS_24	F2	VSS_58	L13
VDD_OUT33_4	J20	VSS_25	F6	VSS_59	L14
VDD_OUT33_5	M21	VSS_26	F10	VSS_60	L15
VDD_OUT33_6	N18	VSS_27	F11	VSS_61	M5
VDD_OUT33_7	R18	VSS_28	F12	VSS_62	M6
VDD_OUT33_8	R20	VSS_29	G20	VSS_63	M8
VDD_OUT33_9	T22	VSS_30	H3	VSS_64	M9
VDD_OUT33_10	U18	VSS_31	H11	VSS_65	M10
VDD_OUT33_11	V21	VSS_32	H12	VSS_66	M11
VDD_PLL	B20	VSS_33	H18	VSS_67	M12
VSS_0	A1	VSS_34	J10	VSS_68	M13
VSS_1	A2	VSS_35	J11	VSS_69	M14
VSS_2	A11	VSS_36	J12	VSS_70	M15
VSS_3	A22	VSS_37	J13	VSS_71	M18
VSS_4	B2	VSS_38	K5	VSS_72	N9
VSS_5	B11	VSS_39	K6	VSS_73	N10
VSS_6	C2	VSS_40	K9	VSS_74	N11
VSS_7	C3	VSS_41	K10	VSS_75	N12
VSS_8	C4	VSS_42	K11	VSS_76	N13

Table 209. Signal List by Signal Name *(continued)*

Signal Name	Ball
VSS_77	N14
VSS_78	P4
VSS_79	P10
VSS_80	P11
VSS_81	P12
VSS_82	P13
VSS_83	P18
VSS_84	R1
VSS_85	R11
VSS_86	R12
VSS_87	T3
VSS_88	T18
VSS_89	U6
VSS_90	U10
VSS_91	U11
VSS_92	U12
VSS_93	U17
VSS_94	V5
VSS_95	V11
VSS_96	V12
VSS_97	V18
VSS_98	W4
VSS_99	W5
VSS_100	W11
VSS_101	W18
VSS_102	W19
VSS_103	Y2
VSS_104	Y3
VSS_105	Y4
VSS_106	Y8
VSS_107	Y11
VSS_108	Y14
VSS_109	Y16
VSS_110	Y19
VSS_111	Y20
VSS_112	AA1
VSS_113	AA2
VSS_114	AA3
VSS_115	AA6
VSS_116	AA11
VSS_117	AA21
VSS_118	AB1
VSS_119	AB11
VSS_120	AB21
VSS_121	AB22
VSS_PLL	A19

Electrical Specifications

All specifications are qualified under recommended operating conditions unless otherwise noted. The information presented in this section is guaranteed over process, recommended supply voltage.

Operating Conditions

Table 210. Recommended Operating Conditions

Parameter	Minimum	Typical	Maximum	Unit
Supply voltage:				
V_{DD}	+1.70	+1.80	+1.90	V
V_{DD_OUT33}	+3.13	+3.30	+3.47	V
V_{DD_IO25}	+2.37	+2.50	+2.63	V
V_{DD_PLL}	+2.37	+2.50	+2.63	V
VSC7384 operating temperature[1]	0		+100	°C
VSC7384-03 operating temperature [1]	–40		+100	°C
[1] Lower limit of specification is ambient temperature, and upper limit is case temperature.				

Maximum Ratings

Table 211. Absolute Maximum Ratings

Parameter	Minimum	Maximum	Unit
Storage temperature	–65	+150	°C
Electrostatic discharge (human body model)	–750	+750	V
Note: Stresses listed under Absolute Maximum Ratings may be applied to devices one at a time without causing permanent damage. Functionality at or above the values listed is not implied. Exposure to these values for extended periods may affect device reliability.			

ELECTROSTATIC DISCHARGE

This device can be damaged by ESD. Vitesse recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures may adversely affect reliability of the device.

Power Sequencing

The following must be observed at all times including power up and down.

V_{DD_OUT33} $\leq V_{DD\_IO25} + 1.35$ V

V_{DD_PLL} $\leq V_{DD\_IO25} + 0.5$ V

The nReset input should be held low until all power supply voltages have reached their recommended operating condition values.

DC Characteristics

Clock Signal

The reference clock is a single-ended LVTTL input.

Table 212. DC Specifications for Clock

Symbol	Parameter	Conditions		Minimum	Maximum	Unit
		Supply	Input/Output			
V_{IH}	Input high voltage			2.0	3.6	V
V_{IL}	Input low voltage			–0.3	0.80	V
I_{IH}	Input high current	V_{DD_IO25}=minimum	V_I=3.3 V		280	µA
I_{IL}	Input low current	V_{DD_IO25}=maximum	V_I=0.0 V	–210		µA

RGMII and MII Management

The outputs and inputs of the RGMII and MII Management interfaces meet or exceed the requirements in the JEDEC JESD8-5 2.5V CMOS interface except for the input leakage current in low state. All RGMII and MIIM outputs and inputs comply with the specifications in Table 213.

Table 213. DC Specifications for RGMII and MII Management

Symbol	Parameter	Conditions		Minimum	Maximum	Unit
		Supply	Input/Output			
V_{OH}	Output high voltage	V_{DD_IO25}= minimum	I_{OH}=–1.0 mA	2.0		V
V_{OL}	Output low voltage	V_{DD_IO25}= minimum	I_{OL}=1.0 mA		0.40	V
V_{IH}	Input high voltage			1.70	3.0	V
V_{IL}	Input low voltage			–0.3	0.70	V
I_{IH}	Input current	V_{DD_IO25}=maximum	V_I=V_{DD_IO25}		5	µA
I_{IL}	Input low current	V_{DD_IO25}=maximum	V_I=0.0 V	–75		µA

iCPU, SI, JTAG, and Other Control Signals

The outputs and inputs meet or exceed the requirements of the LVTTL and LVCMOS standard, JEDEC JESD8-B (September 1999) standard, except for the input leakage current in low state. All outputs and inputs comply with the specifications in Table 214.

Table 214. DC Specifications for PI, iCPU, SI, JTAG, and Other Control Signals

Symbol	Parameter	Conditions		Minimum	Maximum	Unit
		Supply	Input/Output			
V_{OH}	Output high voltage	V_{DD_OUT33} = minimum	I_{OH} = –100 µA	V_{DD_OUT33} – 0.2		V
			I_{OH} = –2 mA	2.4		V
V_{OL}	Output low voltage	V_{DD_OUT33} = minimum	I_{OL} = 100 µA		0.20	V
			I_{OL} = 2 mA		0.40	V
V_{IH} [1]	Input high voltage			2.0	3.6 [1]	V
V_{IL}	Input low voltage			–0.3	+0.8	V
I_{IH}	Input high current	V_{DD_IO25} = minimum	V_I = 3.3 V		5	µA
I_{IL}	Input low current	V_{DD_IO25} = maximum	V_I = 0.0 V	–75		µA

[1] V_{IH}(maximum) = 5.5 V for JTAG input signals.

The following I/O signals comply with the specifications in Table 214:

Clk125_En	Test_Enable	PI_nOE	SI_Clk	JTAG_nTRST
GPIOx	PI_Addr[15:0]	PI_nDone	SI_DI	JTAG_TCK
nReset	PI_Data[15:0]	PI_IRQ	SI_DO	JTAG_TDI
PLL_En	PI_nCS	ICPU_RxD	SI_nEn	JTAG_TDO
JTAG_TMS	PI_nWR	ICPU_TxD		

The GPIOx pin current capabilities are further described in "GPIO," which begins on page 183.

Current Consumption

Table 215 shows the absolute maximum operating current for device, which is found at worst case silicon, minimum operating temperature, and maximum power supply.

Table 215. Maximum Operating Current

Symbol	Parameter	Conditions	Maximum	Unit
I_{DD}	Average active operating current for core	@ V_{DD} = 1.9 V	1100	mA
I_{DD_IO25}	Average active operating current for 2.5 V outputs	@ V_{DD_IO25} = 2.63 V	850	mA
I_{DD_OUT33} [1]	Average active operating current for 3.3 V outputs	@ V_{DD_OUT33} = 3.47 V	25	mA
I_{DD_PLL}	Average active operating current for PLL	@ V_{DD_PLL} = 2.63 V	22	mA

[1] Excluding current sourced on GPIO pins.

Note: All power consumption values assume 100% TX activity.

Typical Current Consumption

Table 216 shows typical values for current consumption for the device in various situations. The numbers are measured under typical conditions (process, temperature, and supply voltage).

Table 216. Typical Current Consumption

Supply	Idle [1]	Full [2]	Unit
V_{DD} (1.8 V)	590	880	mA
V_{DD_IO25} (2.5 V)	190	470	mA
V_{DD_OUT33} (3.3 V)	10	10	mA
V_{DD_PLL} (2.5 V)	22	22	mA

[1] Idle after reset.

[2] Full traffic on Gbps ports and the parallel CPU interface.

Note: All power consumption values assume 100% TX activity.

Power Dissipation

Table 217 shows the typical and maximum power dissipation based on the currents and supply voltages listed in Table 215 and Table 216.

Table 217. Power Dissipation

Parameter	Value	Unit
Typical Power Dissipation	2.9	W
Maximum Power Dissipation	4.5	W

AC Characteristics

All AC electrical characteristics are guaranteed over worst-case to best-case silicon, recommended operating temperature, recommended power supply voltages, and under the individual load conditions shown in the following sections.

Clock Timing

The signal applied to the clock input should comply with the requirements in Table 218 at the pin of the device.

Table 218. System Clock AC Specifications

Symbol	Parameter	Conditions	Minimum	Typical[1]	Maximum	Unit
f_{clk}	Nominal clock frequency	25 MHz input clock		25		MHz
		125 MHz input clock		125		MHz
	Clock frequency tolerance [2]		–50		+50	ppm
	Clock duty cycle	@ 0.5 V_{DD_OUT33}	30		70	%
t_r, t_f	Clock rise and fall times				5	ns

[1] Typical values are at 25°C.

[2] The frequency tolerance of the output clocks of the ports is directly connected to the frequency tolerance on the input clock.

The frequency tolerance of the output clocks of the ports is directly connected to the frequency tolerance on the input clock.

The internal PLL uses the positive going edge; hence this edge has to be clean (for example, free of excess jitter). Any jitter on this edge will be passed to the output clocks.

Reset Timing

Figure 21 shows the nReset signal waveform and required measurement points for the timing specification.

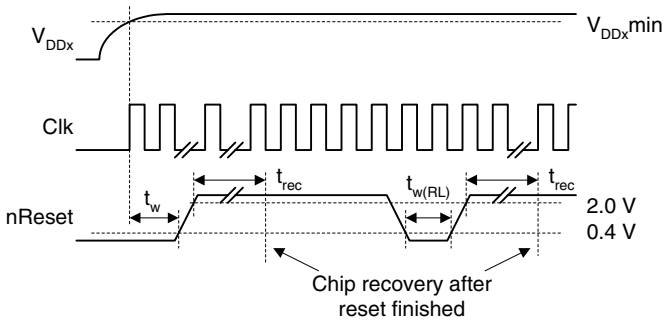


Figure 21. nReset Signal Timing Parameters

The nReset assertion time during power up is measured from when the last of the following happens (all power supplies has reached within valid levels plus Clk signal is stable) to when the nReset signal enters the transition region.

The signal applied to the nReset input complies with the requirements in Table 219 at the pin of the device.

Table 219. nReset AC Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
t_w	nReset assertion time after all power supplies and clock stable		10		ms
t_{rec}	Recovery time from reset inactive to device fully active			60	ns
t_w(RL)	Reset pulse width		20		ns
t_r, t_f	Rise and fall time of nReset	0.8 V to 2.0 V		10	ns
Note: All values are guaranteed by design.					

RGMII (10/100/1000 Mbps)

All AC specifications for the RGMII interfaces meet or exceed the requirements of the HP RGMII draft 1.3. The RGMII timing parameters are defined in Figure 22 and Figure 23.

The skew between the clock and the other signals (both Tx and Rx) is defined as the time period between when the signals and when the clock reach the threshold value of 1.25 V. This is a double data rate interface, so both the rising and falling edges of the clock are used.

Figure 22 and Figure 23 show the RGMII transmit and receive waveforms and required measurement points for the different signals. All AC timing requirements are specified relative to 1.25 V. Note that all values shown in the figures are measured at the “measurement point shown in Figure 24.

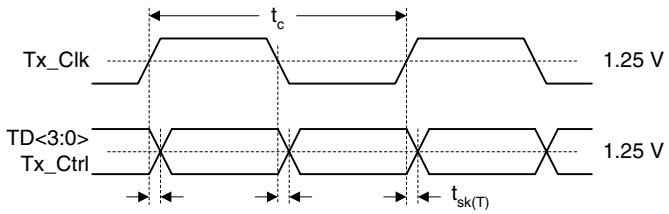


Figure 22. RGMII Transmit Waveforms

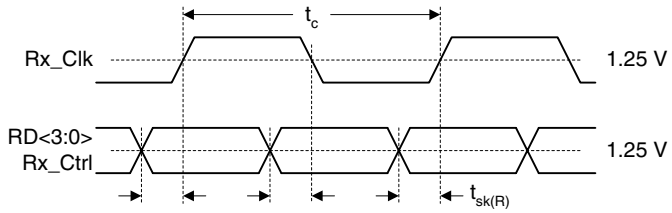


Figure 23. RGMII Receive Waveforms

Additionally, in 10/100 RGMII mode, data is duplicated on the falling edge of the clock (no data transitions on the falling clock).

All RGMII transmit signals comply with the specifications in Table 220 and Table 221 when measured using the test circuit in Figure 24. All RGMII receive signal requirements are requested at the device pins.

Table 220. RGMII 1000 Mbps AC Specifications

Symbol	Parameter	Conditions	Minimum	Typical[1]	Maximum	Unit
f	Clock frequency	1000 Mbps		125		MHz
	Clock frequency stability		–50		+50	ppm
t_c	Clock cycle time	@ 1.25 V	7.2	8	8.8	ns
	Clock duty cycle	@ 1.25 V	45	50	55	%
t_r	RGMII signal rise time	20 to 80%[2]			0.75	ns
t_f	RGMII signal fall time	80 to 20%[2]			0.75	ns
$t_{sk(T)}$	Data to clock output skew	@ 1.25 V	–500	0	+500	ps
$t_{sk(R)}$	Data to clock input skew [3]	@ 1.25 V	1		2.6	ns

[1] Typical values are at 25°C.

[2] Measured relative to worst case DC output levels.

[3] These parameters correspond to a minimum setup time of 1 ns and a minimum hold time of 1 ns.

Table 221. RGMII 10/100 Mbps AC Specifications

Symbol	Parameter	Conditions	Minimum	Typical[1]	Maximum	Unit
f	Clock frequency	100 Mbps		25		MHz
		10 Mbps		2.5		MHz
	Clock frequency stability		–50		+50	ppm
t_c	Clock cycle time	100 Mbps	36	40	44	ns
		10 Mbps	360	400	440	ns
	Clock duty cycle	@ 1.25 V	40	50	60	%
t_r	RGMII signal rise time	20 to 80%[2]			0.75	ns

Table 221. RGMII 10/100 Mbps AC Specifications *(continued)*

Symbol	Parameter	Conditions	Minimum	Typical[1]	Maximum	Unit
t_f	RGMII signal fall time	80 to 20%[2]			0.75	ns
$t_{sk(T)}$	Data to clock output skew	@ 1.25 V	–500	0	500	ps
$t_{sk(R)}$	Data to clock input skew	@ 1.25 V	1			ns

[1] Typical values are at 25°C.

[2] Measured relative to worst case DC output levels.

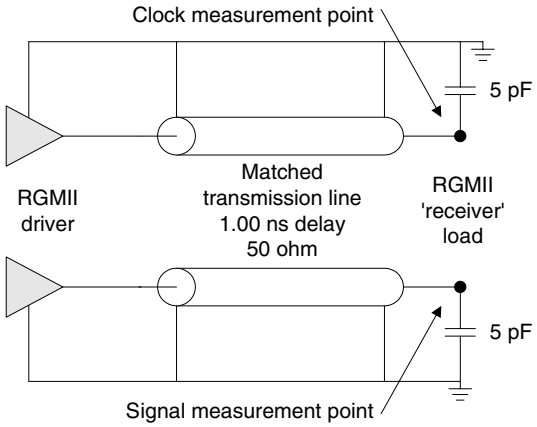


Figure 24. RGMII Test Circuit

MII Management

All AC specifications for the MII Management interface have been designed to meet or exceed the requirements of IEEE 802.3-2002 (clause 22.2-4).

All MII management AC timing requirements are specified relative to the $V_{IL(DC)max}$ and $V_{IH(DC)min}$ threshold values found in the DC specification, that is respectively 0.7 V and 1.7 V.

Additionally, the MDC signal has to be monotonically in the switching region between $V_{IL(DC)max}$ and $V_{IH(DC)min}$.

Figure 25 shows the MII management waveforms and required measurement points for the signals.

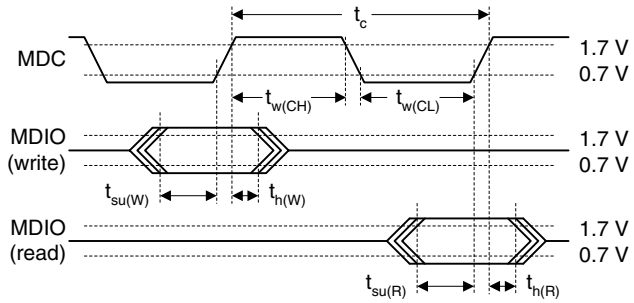


Figure 25. MII Management Waveforms

The setup time of MDIO relative to the rising edge of MDC is defined as the length of time between when the MDIO exits and remains out of the switching region and when MDC enters the switching region. The hold time of MDIO relative to the rising edge of MDC is defined as the length of time between when MDC exits the switching region and when MDIO enters the switching region.

All MII management transmit signals comply with the specifications in Table 222. The MDIO signal requirements are requested at the device pin.

Table 222. MII Management AC Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
f_{clk}	MDC frequency		1	12.5	MHz
t_c	MDC cycle time		80		ns
$t_{w(CH)}$	MDC time high	C_{load} = 50 pF	35		ns
$t_{w(CL)}$	MDC time low	C_{load} = 50 pF	35		ns
$t_{su(W)}$	MDIO setup to MDC on write	C_{load} = 50 pF	10		ns
$t_{h(W)}$	MDIO hold from MDC on write	C_{load} = 50 pF	10		ns
$t_{su(R)}$	MDIO setup to MDC on read	C_{load} = 50 pF on MDC	15		ns
$t_{h(R)}$	MDIO hold from MDC on read		0		ns

Serial CPU Interface

All serial CPU interface (SI) AC timing requirements are specified relative to the input low threshold level of 0.8 V, and the input high threshold level of 2.0 V. The SI timing parameters and required measurement points are defined in Figure 26 and Figure 27.

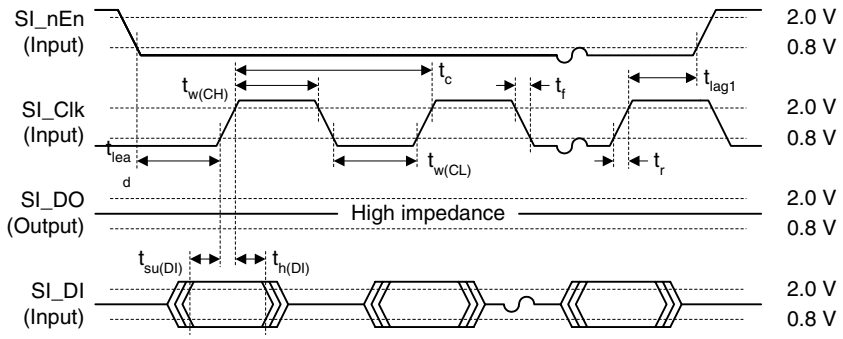


Figure 26. SI Input Data Waveform

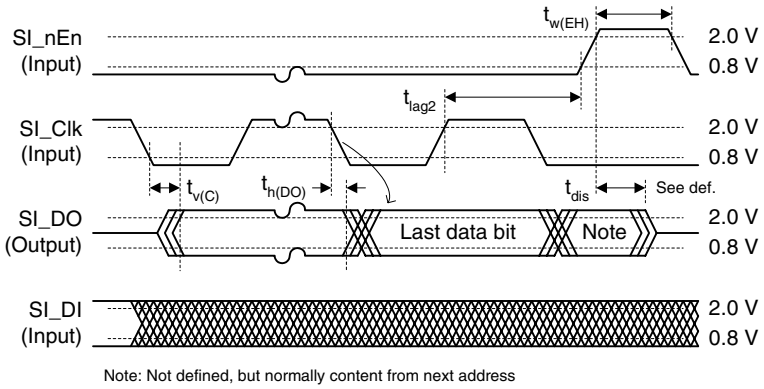


Figure 27. SI Output Data Waveform

All SI signals comply with the specifications in Table 223, and the SI receive signal requirements are requested at the device pins.

Table 223. SI Interface AC Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
f_{clk}	Clock frequency			25 [1]	MHz
t_c	Clock cycle time		40 [1]		ns
$t_{w(CH)}$	Clock time high		16		ns
$t_{w(CL)}$	Clock time low		16		ns
t_r, t_f	Clock rise and fall time	Between V_{IL} and V_{IH}		10	ns
$t_{su(DI)}$	DI setup to clock		4		ns
$t_{h(DI)}$	DI hold from clock		2		ns
t_{lead}	Enable active before first clock		10		ns
t_{lag1} [2]	Enable inactive after clock (input cycle)		32		ns

Table 223. SI Interface AC Specifications *(continued)*

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
t_{lag2}	Enable inactive after clock (output cycle)		See note 3.		ns
$t_{w(EH)}$	Enable inactive width		32		ns
$t_{v(C)}$	DO valid after clock	C_L = 30 pF		12	ns
$t_{h(DO)}$	DO hold from clock	C_L = 0 pF	0		ns
t_{dis} [4]	DO disable time	See Figure 28		15 [5]	ns

[1] The SI clock frequency may be up to 25 MHz, but if it exceeds 1.25 MHz, dummy bytes must be inserted.

[2] t_{lag1} is only defined for write operations to the device, not read.

[3] The last rising edge on the clock is necessary for the master to be able to read in the data. The lag time depends on the necessary hold time on the master data input.

[4] The pin begins to float when a 300 mV change from the loaded V_{OH}/V_{OL} level occurs.

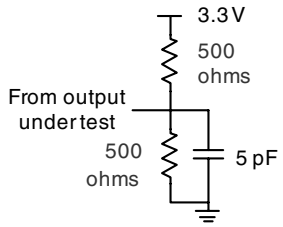
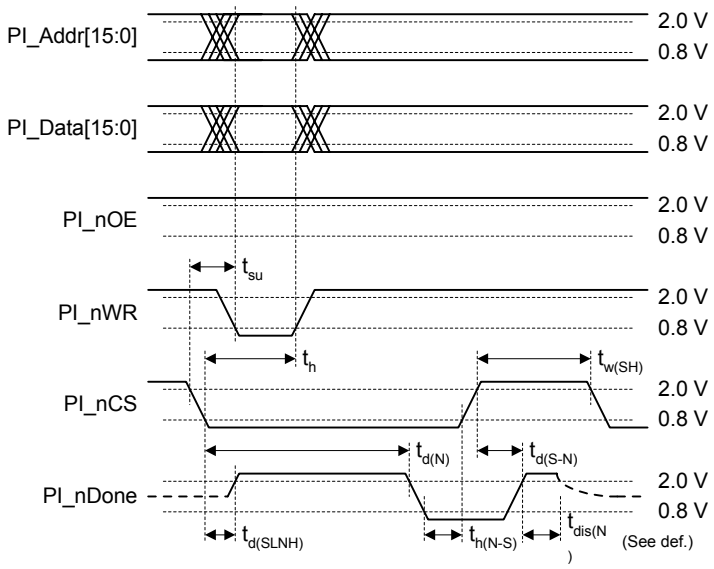


Figure 28. Test Circuit for Signal Disable Test

Parallel CPU Interface

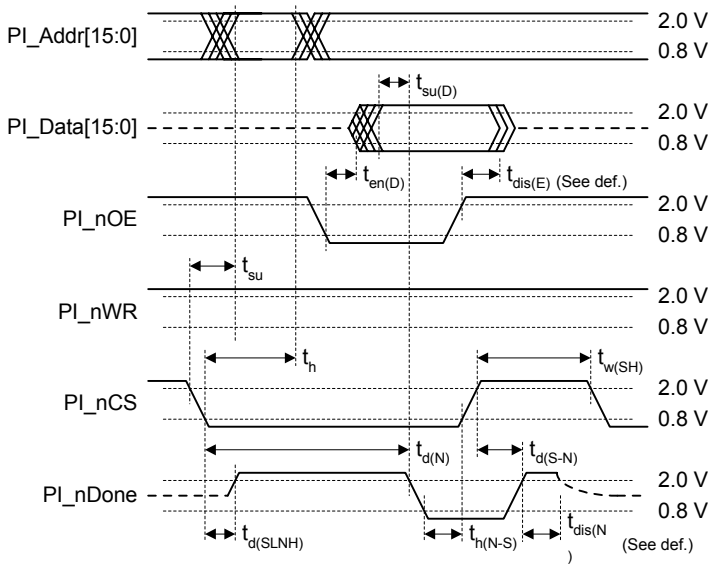
The AC timing provided in this section applies when the parallel CPU interface (PI) is enabled; when the ICPU_PI_En pin is strapped low.

All PI AC timing requirements are specified relative to the input low and high threshold level (0.8 V and 2.0 V), respectively. The PI timing parameters and required measurement points are defined in Figure 29 and Figure 30.



PI_nDone's polarity is programmable, here shown as active low.

Figure 29. PI Write Cycle (Input)



PI_nDone's polarity is programmable, here shown as active low.

Figure 30. PI Read Cycle Waveforms

All PI signals comply with the specifications in Table 224, and the PI receive signal requirements are requested at the device pins.

Table 224. PI Interface AC Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
t_{su}	Addr, Data, nWR setup to nCS falling	Data only on write	–10 [1]		ns
t_h	Addr, Data, nWR hold from nCS low	Data only on write	30 [1]		ns
$t_{d(SLNH)}$	Delay from nCS low to nDone rising[2]	C_L=30pF		10	ns
$t_{d(N)}$	Delay from nCS low to nDone falling[2]	C_L=30pF		52 [1, 2, 3]	ns
$t_{h(N-S)}$	nCS hold from nDone falling[2]		0		ns
$t_{d(S-N)}$	Delay from nCS high to nDone high[2]	C_L=30pF		10	ns
$t_{dis(N)}$ [4]	nDone disable time from nDone pulled inactive[2]	See Figure 31		12	ns
$t_{w(SH)}$	Width of nCS high		15		ns
$t_{en(D)}$ [5]	nOE and nCS low to Data enabled	C_L = 30pF		15	ns
$t_{su(D)}$	Data setup to nDone falling on read[2]	C_L = 30pF	2		ns
$t_{dis(E)}$ [4, 5]	Data disable time from either nCS or nOE high	See Figure 31		12	ns

Notes:
1. An initial delay can be added before input data or conditions are sampled. This delay can be added in steps of 8 ns, and can range from 0 ns to 120 ns; the default is 120 ns. The default ensures that the device will operate with slow CPUs. It is recommended that the value be changed to improve performance. Timing values shown in this table are derived with a 0 ns delay setting.
2. nDone polarity is programmable. Values in this table are derived with nDone set to active low.
3. When using extended bus cycles, the response time can be up to 550 ns.
4. The pin begins to float when a 300 mV change from the loaded V_{OH}/V_{OL} level occurs.
5. Both nOE and nCS must be active to enable the internal data output.

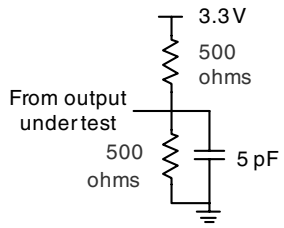


Figure 31. Test Circuit for Signal Disable Test

A reset of the interrupt status signal (PI_IRQ) caused by a read or write will occur simultaneously with the falling edge of PI_nDone, as shown in Figure 32 and Figure 33.

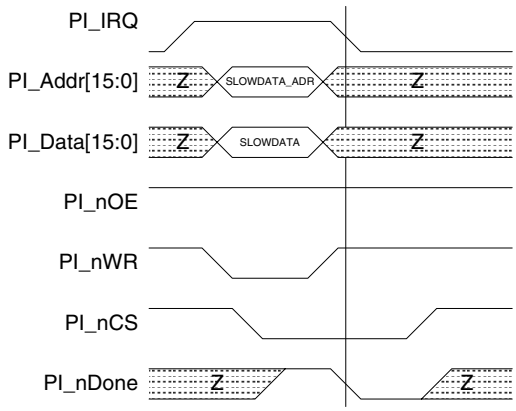


Figure 32. PI Interrupt Write Cycle

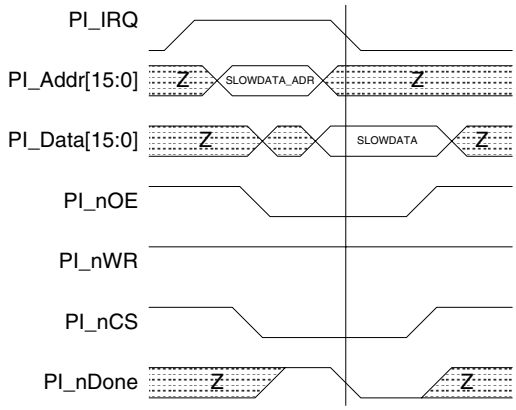


Figure 33. PI Interrupt Read Cycle

iCPU External Memory Interface

The figures and tables in this section refer to the iCPUs external RAM/ROM interface. This only applies when the iCPU is enabled, that is, when the ICPU_PI_En pin is strapped high.

Both the logical and real pin names are presented in the following figures with the logical names in parentheses. See Table 202 on page 142 for information on the pin mapping.

iCPU RAM Read

The iCPU timing parameters and required measurement points for a RAM read access are defined in Figure 34. All iCPU signals comply with the specification in Table 225.

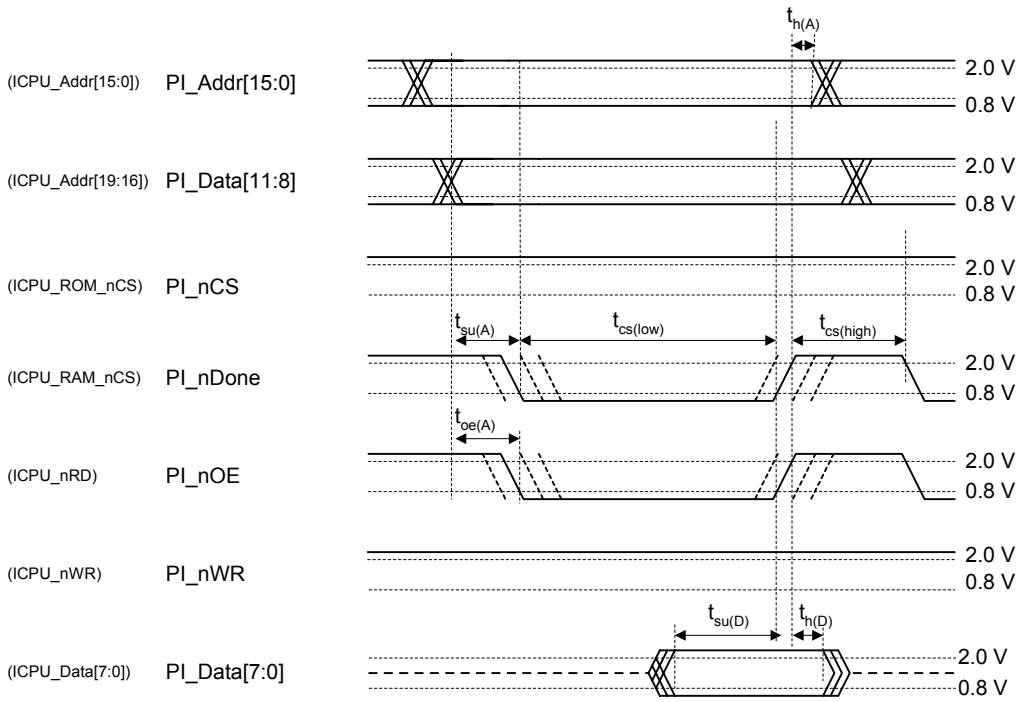


Figure 34. iCPU RAM Read

Table 225. iCPU RAM Read Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
$t_{su(A)}$	Address setup to RAM chip select	C_{load} = 30 pF	4 [1]		ns
$t_{h(A)}$	Address hold from RAM chip select	C_{load} = 30 pF	10 [1]		ns
$t_{cs(low)}$	Chip select low	C_{load} = 30 pF	508 [2]	516 [2]	ns
$t_{cs(high)}$	Chip select high before new access	C_{load} = 30 pF	60		ns
$t_{oe(A)}$	Address setup to Ram Read/OE	C_{load} = 30 pF	4 [3]		ns
$t_{su(D)}$	Data setup to chip select high	C_{load} = 30 pF	35 [4]		ns
$t_{h(D)}$	Data hold from chip select high	C_{load} = 30 pF	0		ns

Notes:

1. RAM chip select (ICPU_RAM_nCS) can be delayed 0, 8, 16, or 24 ns. Default is 8 ns. This is controlled using the CHIP::SYSTEM::ICPU_RAM_CFG register's CHIP_SEL_READ_DELAY field. Address setup to RAM chip select depends on the delay. If the RAM chip select delay is 0 ns, the setup time will be –4 ns maximum. If the RAM chip select delay is 24 ns, the setup time will be 20 ns minimum.
2. The RAM chip select (ICPU_RAM_nCS) low time is decided from two parameters. One is a division of the system clock (125 MHz) to generate the CPU clock. This is controlled using the CHIP::SYSTEM::ICPU_CTRL register's CLK_DIV field. The division can be from 2 to 16 times, giving a CPU clock from 7.8125 to 62.5 MHz. Default is 16 times. The second parameter is the RAM chip select low time, which can be set to 2, 4, 8, 12, …, or 28 8051-clock cycles using the SFR::CKCON::MD register. A clock of 25 MHz and the MD parameter set to 2 clock cycles, gives a RAM chip select low time of 76 ns minimum. If the chip select delay is 8 ns, a 35 ns RAM can be used (Chip Select to Data Out).
3. The Read signal (ICPU_nRD) can also be delayed with 0, 8, 16, or 24 ns. Default is 8 ns. This is controlled with the CHIP::SYSTEM::ICPU_RAM_CFG register's READ_DELAY field. The length of Read low is the same as for RAM chip select (see note 2 above).
4. The setup time for Data to RAM chip select depends on the delay for the RAM chip select. For the default delay of 8 ns, the setup time is 35 ns. If the delay is 0 ns, the setup time is 27 ns; with a delay of 24 ns, the setup time is 51 ns.

iCPU ROM/Flash Read

The iCPU timing parameters and required measurement points for a ROM/Flash read access are defined in Figure 35. All iCPU signals for the ROM/Flash read access comply with the specification in Table 226.

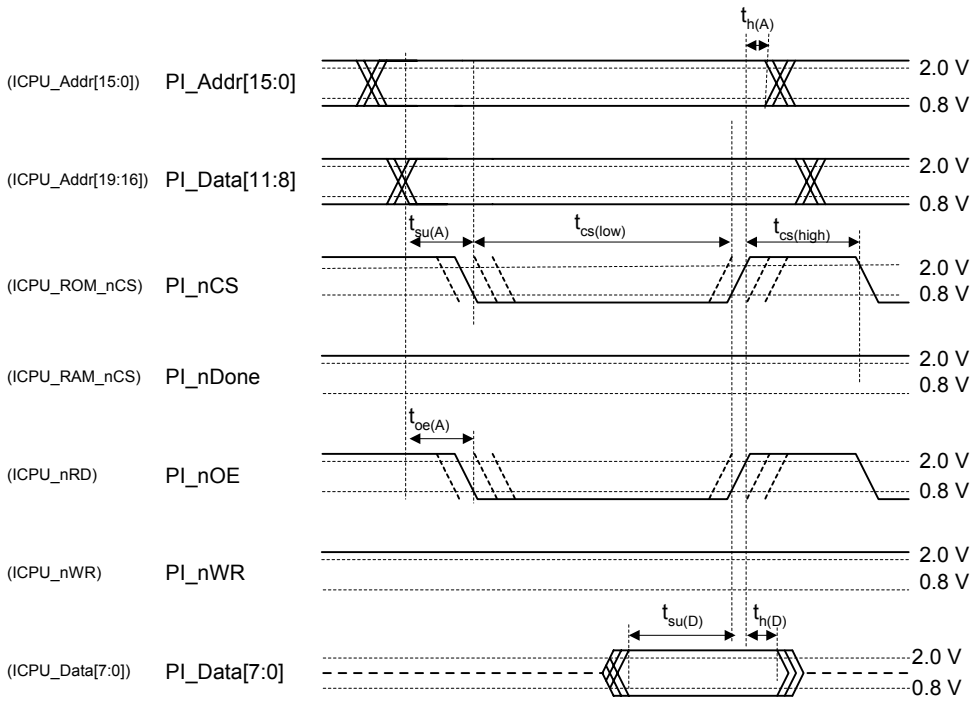


Figure 35. iCPU ROM/Flash Read

Table 226. iCPU ROM/Flash Read Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
$t_{su(A)}$	Address setup to RAM chip select	C_{load} = 30 pF	4 [1]		ns
$t_{h(A)}$	Address hold from RAM chip select	C_{load} = 30 pF	10 [1]		ns
$t_{cs(low)}$	Chip select low	C_{load} = 30 pF	252 [2]	260 [2]	ns
$t_{cs(high)}$	Chip select high before new access	C_{load} = 30 pF	60		ns
$t_{oe(A)}$	Address setup to RAM Read/OE	C_{load} = 30 pF	4 [3]		ns
$t_{su(D)}$	Data setup to chip select high	C_{load} = 30 pF	35 [4]		ns
$t_{h(D)}$	Data hold from chip select high	C_{load} = 30 pF	0		ns

Notes:

1. ROM chip select (ICPU_ROM_nCS) can be delayed 0, 8, 16, or 24 ns. Default is 8 ns. This is controlled with the CHIP::SYSTEM::ICPU_ROM_CFG register's CHIP_SEL_READ_DELAY field. Address setup to ROM chip select depends on the delay. If the ROM chip select delay is 0 ns, the setup time will be –4 ns worst case. If the ROM chip select delay is 24 ns, the setup time will be 20 ns minimum.
2. As opposed to the other types of accesses, the ROM chip select (ICPU_ROM_nCS) low time is decided by only one parameter: the iCPU's current clock frequency, which may be changed using the CHIP::SYSTEM::ICPU_CTRL register's CLK_DIV field. The frequency can range from 7.8125 to 62.5 MHz, with the default being 7.8125 MHz. The value of the SFR::CKCON::MD bits have no effect on the number of clock cycles that ROM chip select is low for reads. It is always 2 8051-clock cycles.
3. The Read signal (ICPU_nRD) can also be delayed with 0, 8, 16, or 24 ns. Default is 8 ns, but can be modified using the CHIP::SYSTEM::ICPU_CTRL_ROM_CFG register's READ_DELAY field. The length of Read low is the same as for ROM chip select. See note 2 above.
4. The setup time for Data-to-ROM chip select depends upon the delay for the ROM chip select. With a delay of 8 ns (default), the setup time is 35 ns. If the delay is 0 ns, the setup time is 27 ns; with a delay of 24 ns, the setup time is 51 ns.

iCPU RAM Write

The iCPU timing parameters and required measurement points for a RAM write access are defined in Figure 36. All iCPU signals for the RAM write access comply with the specification in Table 227.

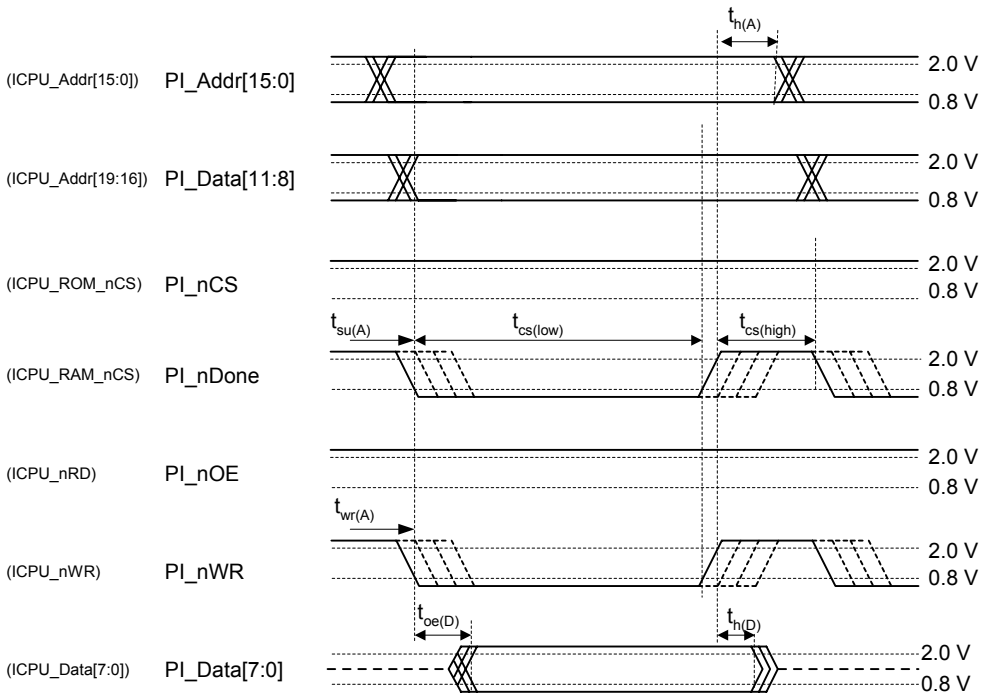


Figure 36. iCPU RAM Write

Table 227. iCPU RAM Write Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
$t_{su(A)}$	Address setup to RAM chip select	C_{load} = 30 pF	–4 [1]		ns
$t_{h(A)}$	Address hold from RAM chip select	C_{load} = 30 pF	0 [1]		ns
$t_{cs(low)}$	Chip select low	C_{load} = 30 pF	508 [2]	516 [2]	ns
$t_{cs(high)}$	Chip select high before new access	C_{load} = 30 pF	60		ns
$t_{wr(A)}$	Address setup to RAM write	C_{load} = 30 pF	–4 [3]		ns
$t_{oe(D)}$	Data valid from chip select	C_{load} = 30 pF		12 [4]	ns
$t_{h(D)}$	Data hold from chip select high	C_{load} = 30 pF	4 [5]		ns

Notes:

1. RAM chip select (ICPU_RAM_nCS) can be delayed 0, 8, 16, or 24 ns. Default is 0 ns in a write access. This is controlled with the CHIP::SYSTEM::ICPU_RAM_CFG register's CHIP_SEL_WRITE_DELAY field. Address setup to RAM chip select depends on the delay. If the RAM chip select delay is 0 ns, the setup time will be –4 ns worst case. If the RAM chip select delay is 24 ns, the setup time will be 20 ns minimum.
2. The RAM chip select (ICPU_RAM_nCS) low time is determined by two parameters. The first is the iCPU's clock frequency, which is controlled using the CHIP::SYSTEM::ICPU_CTRL register's CLK_DIV field. The frequency can range from 7.8125 to 62.5 MHz with default being 7.8125 MHz. The value of the SFR::CKCON::MD bits determines the number of 8051-clock cycles the RAM chip select signal is low. Default is 4 clock cycles, but can be changed to 2, 4, 8, 12, …, or 28 clock cycles.
3. The Write signal (ICPU_nWR) can also be delayed with 0, 8, 16, or 24 ns. Default is 0 ns for write accesses, but can be changed via the CHIP::SYSTEM::ICPU_RAM_CFG register's WRITE_DELAY field. The length of Write low is the same as for RAM chip select (see note 2 above).
4. Data valid from RAM chip select low is 12 ns maximum. The time depends on the RAM chip select delay. If the delay is 8 ns, the valid time is 4 ns; with a delay of 24 ns, the data valid time from RAM chip select is –12 ns.
5. Data can be extended in the end with 0, 8, 16, or 24 ns. Default is 8 ns, but can be modified through the CHIP::SYSTEM::ICPU_RAM_CFG register's WRITE_DATA_HOLD field. The default of 8 ns gives a hold time of 4 ns (RAM chip select delay is 0 ns). If the RAM chip select is delayed 8 ns, the data has to be extended with 16 ns to maintain a hold of 4 ns.

iCPU Flash Write

The iCPU timing parameters and required measurement points for a Flash write access are defined in Figure 37. All iCPU signals for the Flash write access comply with the specification in Table 228.

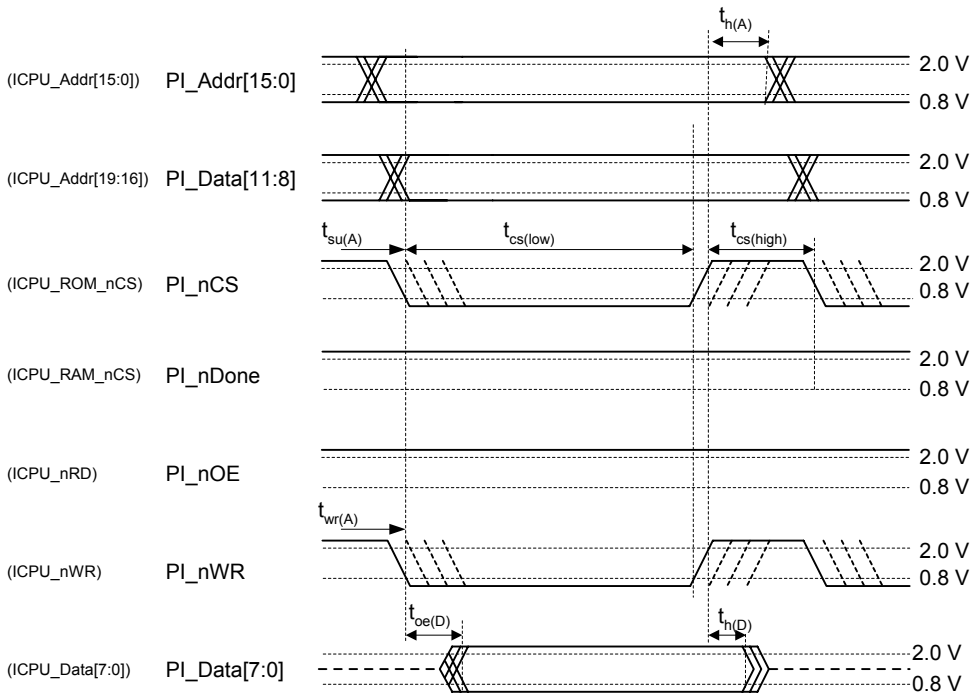


Figure 37. iCPU Flash Write

Table 228. iCPU Flash Write Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
$t_{su(A)}$	Address setup to RAM chip select	C_{load} = 30 pF	–6 [1]		ns
$t_{h(A)}$	Address hold from RAM chip select	C_{load} = 30 pF	0 [1]		ns
$t_{cs(low)}$	Chip select low	C_{load} = 30 pF	508 [2]	516 [2]	ns
$t_{cs(high)}$	Chip select high before new access	C_{load} = 30 pF	60		ns
$t_{wr(A)}$	Address setup to RAM write	C_{load} = 30 pF	–6 [3]		ns
$t_{oe(D)}$	Data valid from chip select	C_{load} = 30 pF		12 [4]	ns
$t_{h(D)}$	Data hold from chip select high	C_{load} = 30 pF	4 [5]		ns

Notes:

1. ROM chip select (ICPU_ROM_nCS) can be delayed 0, 8, 16, or 24 ns. Default is 0 ns for write accesses. This is controlled with the CHIP::SYSTEM::ICPU_ROM_CFG register's CHIP_SEL_WRITE_DELAY field. Address setup to RAM chip select depends on the delay. If the ROM chip select delay is 0 ns, the setup time will be –4 ns worst case. If the ROM chip select delay is 24 ns, the setup time will be 20 ns minimum.
2. The ROM chip select (ICPU_ROM_nCS) low time is decided from two parameters. The first is the iCPU's clock frequency, which is controlled with the CHIP::SYSTEM::ICPU_CTRL register's CLK_DIV field. The frequency can range from 7.8125 to 62.5 MHz with default being 7.8125 MHz. The value of the SFR::CKCON::MD bits determines the number of 8051-clock cycles the ROM chip select signal is low. Default is 4 clock cycles, but can be changed to 2, 4, 8, 12, …, or 28 clock cycles.
3. The Write signal (ICPU_nWR) can also be delayed with 0, 8, 16, or 24 ns. Default is 0 ns for write accesses, but can be modified with the CHIP::SYSTEM::ICPU_ROM_CFG register's WRITE_DELAY field. The length of Write low is the same as for the ROM chip select (see note 2 above).
4. Data valid from ROM chip select low is 12 ns maximum. The time depends on the ROM chip select delay. If the delay is 8 ns, the valid time is 4 ns; with a delay of 24 ns, the data valid time from ROM chip select is –12 ns.
5. Data can be extended in the end with 0, 8, 16, or 24 ns. Default is 8 ns, but can be modified through the CHIP::SYSTEM::ICPU_ROM_CFG register's WRITE_DATA_HOLD field. The default of 8 ns gives a hold time of 4 ns (ROM chip select delay is 0 ns). If the ROM chip select is delayed 8 ns, the data has to be extended with 16 ns to maintain a hold of 4 ns.

JTAG

All AC specifications for the JTAG interface have been designed to meet or exceed the requirements of IEEE 1149.1-2001.

Figure 38 shows the JTAG transmit and receive waveforms and required measurement points for the different signals.

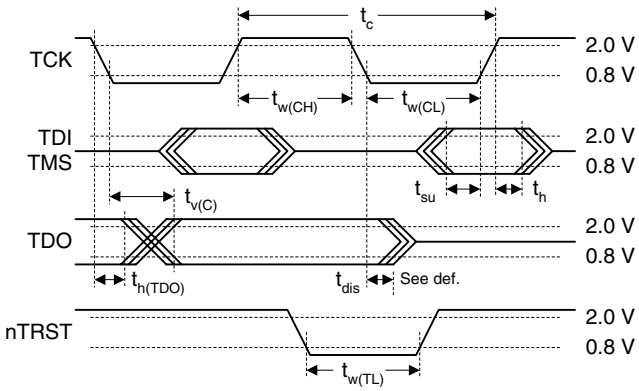


Figure 38. JTAG Interface Timing Definitions

All JTAG signals comply with the specifications in Table 229, and the JTAG receive signal requirements are requested at the device pin.

Table 229. JTAG AC Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
f_{clk}	TCK frequency			10	MHz
t_c	TCK cycle time		100		ns
$t_{w(CH)}$	TCK time high		40		ns
$t_{w(CL)}$	TCK time low		40		ns
t_{su}	Setup to TCK rising		0		ns
t_h	Hold from TCK rising		30		ns
$t_{v(C)}$	TDO valid after TCK falling	C_L = 10 pF		28	ns
$t_{h(DO)}$	TDO hold from TCK falling	C_L = 0 pF	0		ns
t_{dis} [1]	DO disable time	See Figure 39		30 [1]	ns
$t_{w(TL)}$	nTRST time low		30		ns
[1] The pin begins to float when a 300 mV change from the loaded V_{OH}/V_{OL} level occurs.					

The JTAG_nTRST signal is asynchronous to the clock and consequently does not have a setup and hold time requirement.

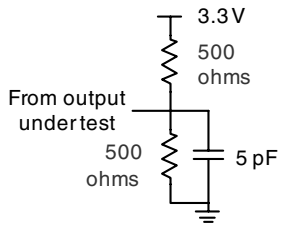


Figure 39. Test Circuit for Signal Disable Test

Design Guide

Power Supplies

Power Supply Decoupling

Core 1.8 V (VDD)

The VDD supply should be made as a plane where all VDD pins are connected.

2.5 V I/O Supply (VDD_IO25)

All VDD_IO25 pins must be connected to the same power supply.

3.3 V Output Supply (VDD_OUT33)

All VDD_OUT33 pins must be connected to the same power supply.

PLL

PLL Supply Filtering

It is recommended that the two PLL power pins (VDD_PLL, VSS_PLL) be connected to a “quiet” supply. This can be done by low-pass filtering the digital VDD_IO25 using a ferrite bead in series with the decoupling capacitor. Remember to connect the VSS_PLL to the filter capacitor first, then connect it to the digital VSS power plane.

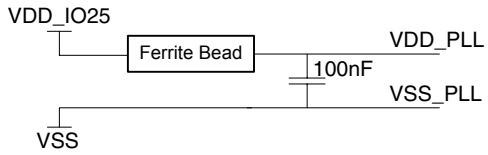


Figure 40. PLL Supply Filtering

PLL Loop Filter Capacitor

The on-chip PLL uses an external 0.1 µF capacitor connected between its Cap0 and Cap1 terminals to control the Loop Filter. This capacitor should be a multilayer ceramic dielectric, or better, with at least a 5 V working voltage rating and a good temperature coefficient. This capacitor is used to minimize the impact of common-mode noise on the Clock Multiplier Unit (especially power supply noise). Higher value capacitors provide better robustness in systems. NPO is preferred but X7R may be acceptable. NPO is preferred over X7R because the power supply noise sensitivity varies with temperature.

For best noise immunity, you may choose to use a 3-capacitor circuit with one differential capacitor between Cap0 and Cap1 (C1), a capacitor from Cap0 to ground (C2), and a capacitor from Cap1 to ground (C3). Larger values are

better but 0.1 µF is adequate. However, if the designer cannot use a 3-capacitor circuit, a single differential capacitor (C1) is adequate. These components should be isolated from noisy traces.

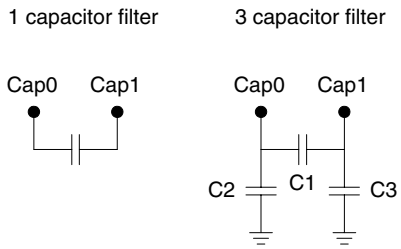


Figure 41. PLL Loop Filter Capacitor

PLL Inputs

The reference clock can be either a 25 MHz or a 125 MHz clock. The function is:

Clk125_En selects 125 MHz clock if high or 25 MHz clock if low.

Clk must be a 25 MHz LVTTL clock when Clk125_En is low, else a 125 MHz LVTTL clock when Clk125_En is high. Refer to "Termination Considerations." which begins on page 186 for layout recommendations on clock layout. The clock can be supplied from a 2.5 V or 3.3 V oscillator/driver.

PLL_En must be high for normal operation.

Logic high can be supplied from 2.5 V to 3.3 V sources, VDD_IO25 or VDD_OUT33.

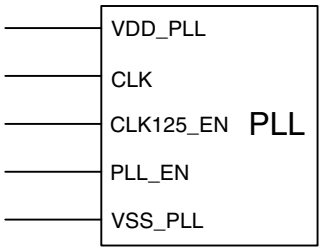


Figure 42. PLL Power and Control Signals

GPIO

The GPIO pins are designed to provide a current sourcing capability of 4 mA and a current sinking capability of 2 mA. GPIOs are typically used to control status LEDs and should thus consider the use of high-efficiency LEDs as well as using the GPIO to source current rather than sink it. If external current requirements are higher, you must consider the inclusion of an external buffer/driver.

External Pull Resistors

All single ended inputs on Stansted that are unused must have a pull resistor to maintain a stable level. The following signals may need a pull resistor:

Clk125_En	JTAG_TMS	PI_nOE	RGMIIx_Rx_Clk	SI_nEn
JTAG_nTRST	nReset	PI_nWR	RGMIIx_Rx_Ctrl	Test_Enable
JTAG_TCK	PI_Addr[15:0]	PLL_En	SI_DI	ICPU_RxD
JTAG_TDI	PI_nCS	RGMIIx_RD[3:0]	SI_Clk	

Remember to pull the reserve pins according to the values in Table 207 on page 146.

Interfaces

MII Management

The Stansted device includes one management bus. It is recommended that layout is done very carefully.

It is recommended that the clock signal (MDC) is routed from the device to PHY 1 then PHY 2 then PHY n and then terminated with 100 Ω to ground and 90 Ω toVDD_IO25. It is also recommended that the PCB trace is routed with as high impedance as possible and at least 68 Ω.

MDIO should have 1.5 k Ω pull-up resistor to 2.5 V, and the trace should follow the MDC trace. See Figure 43.

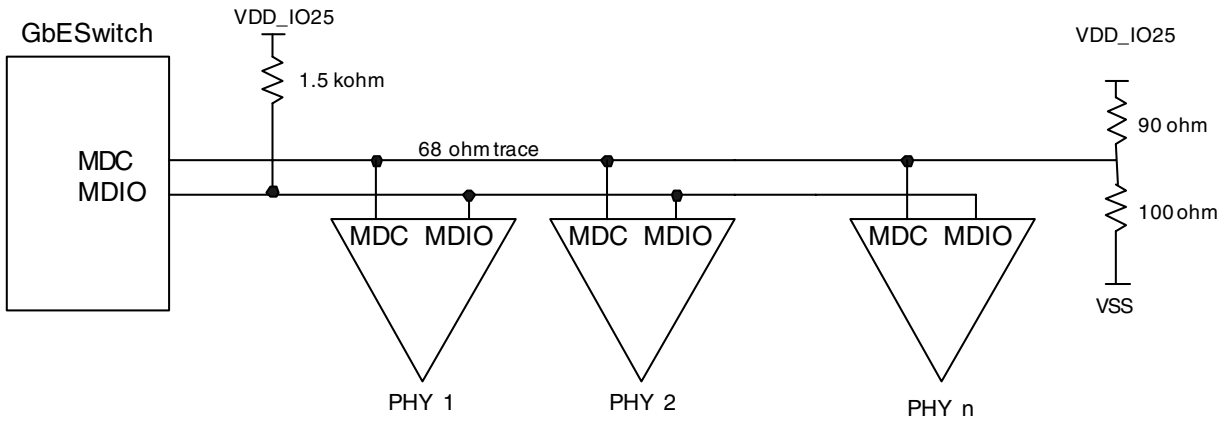


Figure 43. MDC/MDIO Layout Scheme

Care should be taken with respect to the layout guidelines described in "Termination Considerations." which begins on page 186.

If it is desired to route the PCB-trace with 50 Ω, it is recommended that a clock driver be used. This driver should be placed close to the device. It must be verified against the data sheet that this driver is capable of driving 35 Ω traces. This is because the impedance is lowered when the trace is loaded capacitive-wise by the PHYs. Routing guidelines will be the same as above, and the termination should be equal to the loaded trace impedance (typically 35 Ω).

It also is possible to use individual clock drivers for each PHY, routed in a star configuration.

Parallel CPU Interface

This section applies when the ICPU_PI_En strapping pin is pulled low, that is, when the parallel CPU interface (PI) is enabled (even when it is not used).

This interface consists of PI_Addr[15:0], the PI_Data[15:0], PI_nCS, PI_nDone, PI_nOE, PI_nWR, and PI_IRQ. If this interface is not used, all signals except PI_IRQ must be pulled high. The PI_Addr[15:0], the PI_Data[15:0], and the four control signals can be pulled high with a single resistor. When using an 8-bit interface, the unused data bits (PI_data[8:15]) must be individually pulled high or low. Use resistors with values from 5 kΩ to 20 kΩ.

This interface can be configured either as an 8-bit or a 16-bit interface. If configured as an 8-bit interface, PI_Data[7..0] carries data information and PI_Data[15..8] needs to be pulled (high is preferred). For address mapping using an 8-bit interface, see the "8-bit Data Bus Width." which begins on page 79. Note that when using the parallel interface, the time $t_{d(SLNH)}$ listed in Table 224 on page 170 indicates when an issued command is sampled by the device.

If the host CPU is not located close to the device, it is recommended that the address and (at least) the control signals are series terminated close to the CPU to avoid crosstalk or malfunction of the interface. For specific values, refer to data for the CPU output driver.

To ensure that the PI_nDone signal is driven inactive properly, add a 4.7 kΩ pull-up resistor to this signal. Note that during reset, PI_nDone is not inactive, but is driven low during the entire reset cycle.

Serial CPU Interface

The serial CPU iinterface (SI) consists of four signals: SI_DO, SI_DI, SI_Clk, and SI_nEn. SI_DO, SI_DI, and SI_Clk can be shared by multiple devices and there must be one SI_nEn separate for each device on the bus. Care should be taken when routing SI_Clk to avoid reflections or noise from other traces that causes double clocking of one or more devices on the bus (see "Termination Considerations." which begins on page 186 for recommendations).

If this interface not is used, the inputs should be pulled high or low (SI_nEn must be pulled high). For the other three signals, high is recommended.

JTAG Interface

When the JTAG interface is not used, the input pins must be pulled high (inactive) except JTAG_nTRST, which must be pulled to ground. It can either be tied to ground or if a JTAG chain is used, a pull down resistor of 2 kΩ maximum should be used. If other JTAG resets must be pulled low in the JTAG chain, a smaller value resistor should be used. Consult specific devices for their internal pull resistors.

MAC Interface

The MAC interfaces for the device can be configured for RGMII only.

To reduce the crosstalk between signal and clock PCB lines (Tx_Clk and Rx_Clk), make sure that the spacing on each side of the clock lines is larger than twice the track width. If a port is not used, pull RGMII_Rx_Ctrl low and RGMII_RD[3:0] plus RGMII_Rx_Clk high (or low).

Termination Considerations

MAC Interfaces

All signals in the MAC interface and the reference clock input (both on the VSC7384 Stansted and the PHYs) require termination considerations.

All RGMII outputs are designed to drive one load and they have built-in series termination optimized for 55Ω transmission lines. These transmission lines should be manufactured as 55Ω, ±10%.

If any additional termination is added to the RGMII output signals, it is recommended that all signals within each port be terminated equally.

All RGMII traces should be impedance controlled. RGMII traces should be kept shorter than 1.8 ns.

Summary:

- Place series termination resistors, if needed, as close to the output pins as possible.
- Keep output traces, between the device and one PHY, approximately the same length. Notice the special clk track requirement in RGMII mode, if the PHY does not include the clock delay.
- Keep input traces, between one PHY and the device, approximately the same length. Notice the special clk track requirement in RGMII mode, if the PHY does not include the clock delay.
- Keep RGMII traces shorter than 1.8 ns.

Other Outputs

All other outputs do not have a built-in series termination, and they should be terminated appropriately on board if required.

Other Inputs

For recommendations on the termination of input signals to the device, refer to the data sheet of the specific devices.

iCPU Flash Access

When selecting an external flash for the iCPU, the clock frequency of the iCPU must be taken into consideration, the reason being that during a read operation ICPU_ROM_nCS is low for two 8051-clock cycles. Refer to "iCPU ROM/ Flash Read." which begins on page 174 for details on the timing.

Table 230 illustrates the timing restrictions focusing on time left for the flash access for a 15.625 MHz and a 25 MHz iCPU clock frequency.

Table 230. Flash Access Timing

Timing	Clock Frequency 15.625 MHz	Clock Frequency 25 MHz	Unit
ICPU_ROM_nCS active (2 iCPU clock cycles)	128	80	ns
Data setup time before ICPU_ROM_nCS goes inactive with IPU_ROM_CFG::CS_RD_DELAY set to 0 ns	27	27	ns
Board delay (mapping of ICPU_ROM_nCS to the flash _nCS + traces)	7	7	ns
Time left for flash access	128 – 27 – 7= 94	80 – 27 – 7 = 46	ns

Running the iCPU at 15.625 MHz requires a flash with a 90 ns access time and running at 25 MHz requires a flash with a 45 ns access time.

Other Design Information

Signal Reference Plane

In the package, all signals are referenced to the common ground plane (VSS). Therefore, it is recommended that all high-speed signals are referenced to ground or have a low impedance AC return path to ground if referenced to another power supply.

Reference Clock

The reference clock for the VSC7384 Stansted and PHYs can be taken from the same oscillator. The reference clock for PHYs must be a low-jitter type and must come from a non-PLL based clock source (including oscillator, buffer, and so forth). A PLL base source clock will disturb the built-in PLL in the PHYs.

Package Information

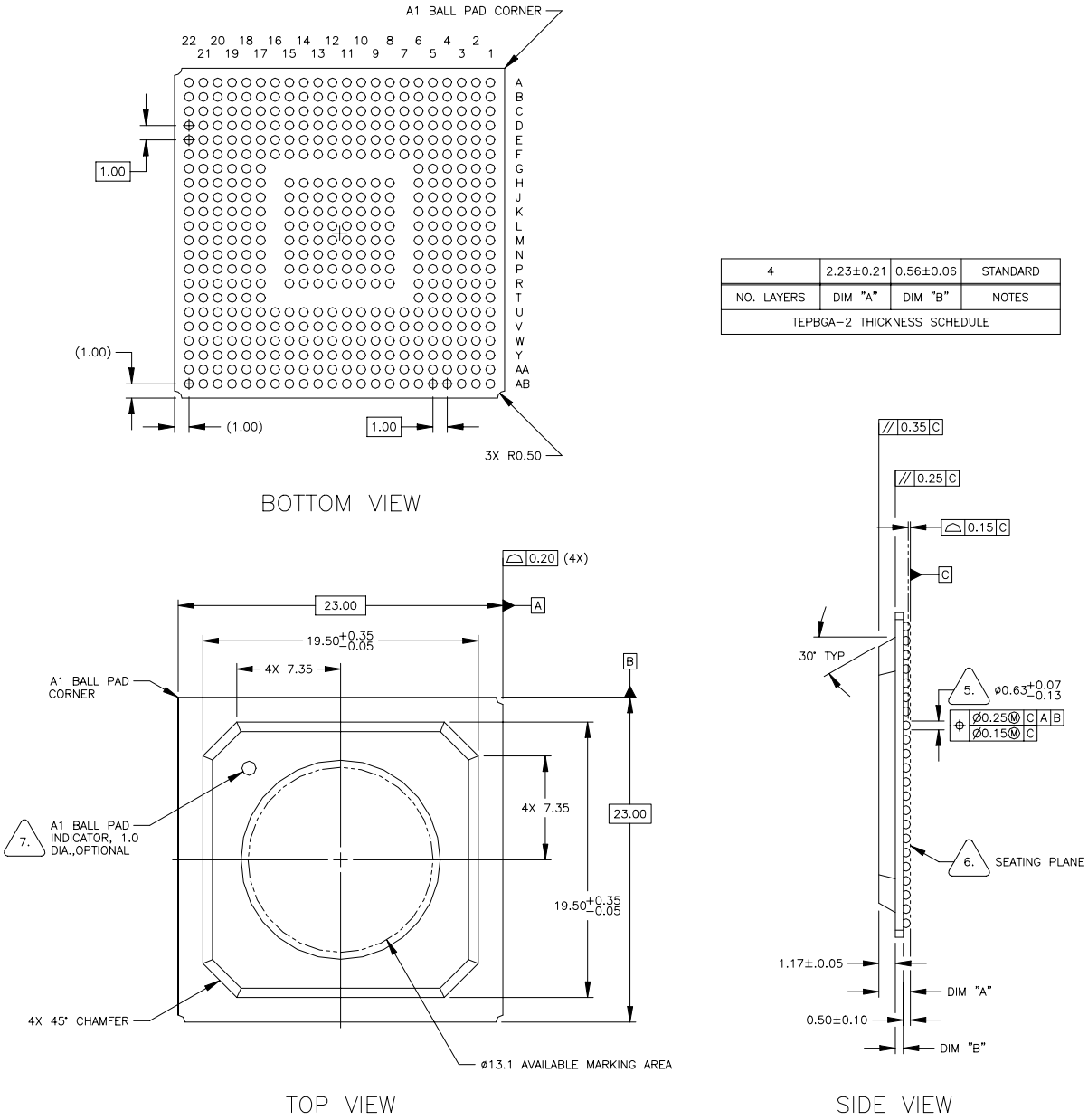
The VSC7384 device is available in the following package types, including lead-free packages:

- VSC7384VU is a 448-pin, thermally enhanced, plastic ball grid array (BGA) with a 23 mm × 23 mm body size, 2.23 mm body thickness, and 1 mm pitch. The operating temperature is 0 °C ambient to 100 °C case.
- VSC7384XVU is a lead-free, 448-pin, thermally enhanced, plastic ball grid array (BGA) with a 23 mm × 23 mm body size, 2.23 mm body thickness, and 1 mm pitch. The operating temperature is 0 °C ambient to 100 °C case.
- VSC7384VU-03 is a 448-pin, thermally enhanced, plastic ball grid array (BGA) with a 23 mm × 23 mm body size, 2.23 mm body thickness, and 1 mm pitch. The operating temperature is –40 °C ambient to 100 °C case.
- VSC7384XVU-03 is a lead-free, 448-pin, thermally enhanced, plastic ball grid array (BGA) with a 23 mm × 23 mm body size, 2.23 mm body thickness, and 1 mm pitch. The operating temperature is –40 °C ambient to 100 °C case.

Lead-free products from Vitesse comply with the temperatures and profiles defined in the joint IPC and JEDEC standard IPC/JEDEC J-STD-020. For more information, see the IPC and JEDEC standard.

Package Drawing

All Measurements are in mm
Body Size: 23.00 x 23.00 TEPBGA-2 Package
No. of Balls: 448
Ball Pitch: 1.00 mm
A1 Ball Corner Location Indication on Top of Package



4	2.23±0.21	0.56±0.06	STANDARD
NO. LAYERS	DIM "A"	DIM "B"	NOTES
TEPBGA–2 THICKNESS SCHEDULE			

Figure 44. Package Drawing

Thermal Specifications

Thermal specifications for this device are based on the JEDEC standard EIA/JESD51-2 and have been modeled using a four-layer test board with two signal layers, a power plane, and a ground plane (2s2p PCB). For more information, see the JEDEC standard.

Table 231. Thermal Resistances

Part Number	θ_{JC}	θ_{JA} (°C/W) vs. Airflow (ft/min)		
		0	100	200
VSC7384VU	2.5	13.6	11.0	9.6
VSC7384XVU	2.5	13.6	11.0	9.6
VSC7384VU-03	2.5	13.6	11.0	9.6
VSC7384XVU-03	2.5	13.6	11.0	9.6

To achieve results similar to the modeled thermal resistance measurements, the guidelines for board design described in the JEDEC standard EIA/JESD51 series must be applied. For information about specific applications, see the following:

EIA/JESD51-5, *Extension of Thermal Test Board Standards for Packages with Direct Thermal Attachment Mechanisms*

EIA/JESD51-7, *High Effective Thermal Conductivity Test Board for Leaded Surface Mount Packages*

EIA/JESD51-9, *Test Boards for Area Array Surface Mount Package Thermal Measurements*

EIA/JESD51-10, *Test Boards for Through-Hole Perimeter Leaded Package Thermal Measurements*

EIA/JESD51-11, *Test Boards for Through-Hole Area Array Leaded Package Thermal Measurements*

Moisture Sensitivity

This device is rated moisture sensitivity level 3 or better as specified in the joint IPC and JEDEC standard IPC/JEDEC J-STD-020. For more information, see the IPC and JEDEC standard.

Design Considerations

Designers should pay particular attention to the layout to ensure that the timing requirements for the RGMII interface are met. Vitesse field application engineers (FAEs) have extensive experience with optimizing board layouts. Contact Vitesse FAEs for layout recommendations. They can also review schematics and gerber files to ensure a robust, manufacturable design.

Incoming Pause Frames are Ignored for Egress-Congested Links

Issue

On a flow control port, incoming pause frames transmitted by a link partner can be ignored if egress is congested.

Implications

This only has implications for flow control ports.

Since pause frames may be ignored, the Stansted device does not intercept and pause the egress traffic flowing toward the link partner signaling flow control. There are no impacts on performance on the Stansted, but the link partner may experience frame drops from buffer overflow due to the Stansted device's not pausing the traffic.

Workaround

For 10/100M ports, setting the WaitForDone bit in register ADVPORTM will remove the issue completely. There are no side effects with this setting.

For Gigabit ports, setting the WaitForDone bit will remove the issue with ignored pause frames but resolve in the minimum interframe gap being at least 14 bytes instead of the usual 12 bytes. This applies for uncongested traffic as well. The larger interfame gap will result in less than 100% throughput rates. For a stream of 64-byte frames, the maximum throughput is 97.7% and for a stream of 1518-byte frames, the maximum throughput is 99.9%.

Status

Presently, there are no plans to make any adjustments to this issue.

BC/MC Storm Control Drops Too Many Frames

Issue

It can be observed that the BC and MC storm control will drop more than the configured rate in the MCSTORMCONF register when the offered load of BC/MC frames exceeds the configured rate.

Implications

The larger the offered load of BC/MC frames is relative to the configured BC/MC storm control rate, the smaller the actual rate will be. This is due to a fraction of the dropped frames being counted as part of the storm control rate. The problem is greatest for small frames and is insignificant for large frames.

Workaround

No workaround currently exists.

Status

Presently, there are no plans to make any adjustments to this issue.

Ordering Information

The VSC7384 device is available in the following package types, including lead-free packages:

- VSC7384VU is a 448-pin, thermally enhanced, plastic ball grid array (BGA) with a 23 mm × 23 mm body size, 2.23 mm body thickness, and 1 mm pitch. The operating temperature is 0 °C ambient to 100 °C case.
- VSC7384XVU is a lead-free, 448-pin, thermally enhanced, plastic ball grid array (BGA) with a 23 mm × 23 mm body size, 2.23 mm body thickness, and 1 mm pitch. The operating temperature is 0 °C ambient to 100 °C case.
- VSC7384VU-03 is a 448-pin, thermally enhanced, plastic ball grid array (BGA) with a 23 mm × 23 mm body size, 2.23 mm body thickness, and 1 mm pitch. The operating temperature is –40 °C ambient to 100 °C case.
- VSC7384XVU-03 is a lead-free, 448-pin, thermally enhanced, plastic ball grid array (BGA) with a 23 mm × 23 mm body size, 2.23 mm body thickness, and 1 mm pitch. The operating temperature is –40 °C ambient to 100 °C case.

Lead-free products from Vitesse comply with the temperatures and profiles defined in the joint IPC and JEDEC standard IPC/JEDEC J-STD-020. For more information, see the IPC and JEDEC standard.

The following table lists the ordering information for the VSC7384 device.

Ordering Information

Part Number	Description
VSC7384VU	Stansted 12 x 1 Gigabit Ethernet switch. 448-pin, thermally enhanced, plastic BGA with a 23 mm × 23 mm body size, 2.23 mm body thickness, and 1 mm pitch. Operating temperature is 0 °C ambient to 100 °C case.
VSC7384XVU	Stansted 12 x 1 Gigabit Ethernet switch. Lead-free, 448-pin, thermally enhanced, plastic BGA with a 23 mm × 23 mm body size, 2.23 mm body thickness, and 1 mm pitch. Operating temperature is 0 °C ambient to 100 °C case.
VSC7384VU-03	Stansted 12 x 1 Gigabit Ethernet switch. 448-pin, thermally enhanced, plastic BGA with a 23 mm × 23 mm body size, 2.23 mm body thickness, and 1 mm pitch. Operating temperature is –40 °C ambient to 100 °C case.
VSC7384XVU-03	Stansted 12 x 1 Gigabit Ethernet switch. Lead-free, 448-pin, thermally enhanced, plastic BGA with a 23 mm × 23 mm body size, 2.23 mm body thickness, and 1 mm pitch. Operating temperature is –40 °C ambient to 100 °C case.

Standard References

Ports

The Ethernet ports have been designed to meet or exceed the requirements found in the following standards:

- IEEE 802.3-2002 Edition for Ethernet, Fast Ethernet, and Gigabit Ethernet
- Hewlett Packard RGMII specification v1.3; Reduced Gigabit Media Independent Interface

Abbreviations

Table 232. Abbreviations

Abbreviation	Explanation
CFI	canonical format indicator
CMD	command
CPU	central processing unit
CRAM	code RAM, referring to iCPUs on-chip RAM when accessed in code space
CRC	cyclic redundancy check
CS	chip select
DS	differentiated services
DSAP	destination service access point
DMAC	destination media access controller address
DRAM	data RAM, referring to iCPUs on-chip RAM when accessed in data space
FIFO	first in first out
GARP	generic/group attributes registration protocol
GbE	Gigabit Ethernet
Gbps	gigabits per second
GMRP	generic/group multicast registration protocol
GPIO	general purpose input/output
GVRP	generic/group VLAN registration protocol
iCPU	internal CPU, referring to on-chip 8051 CPU
Id	identification
IEEE	Institute of Electrical and Electronic Engineers
IGMP	internet group management protocol

Table 232. Abbreviations *(continued)*

Abbreviation	Explanation
IP	internet protocol
ISR	interrupt service routine
LAN	local area network
MAC	media access controller
Mbps	megabit per second
MDIO	management data input output
MII	medium independent interface
MIIM	medium independent interface management
Mpacketsps	million packets per second
PCS	physical coding sublayer
PHY	physical layer device
PI	parallel CPU interface
QoS	quality of service
RA	register access
RAM	random access memory
Rbc	receive bit clock
RD	read
RFC	request for comments
RGMII	reduced gigabit medium independent interface
RMON	remote monitoring
ROM	read only memory
RST	reset
RW	read/write
SFD	start of frame delimiter
SFR	special function register
SMAC	source media access controller
SNMP	simple networking management protocol
SI	serial CPU interface
STP	spanning tree protocol
TCP	transmission control protocol
UDP	user data protocol
VLAN	virtual local area network
VID	virtual local area network identifier
VoIP	voice over internet protocol

Table 232. Abbreviations *(continued)*

Abbreviation	Explanation
WAN	wide area network
WD	watchdog
WR	write

CORPORATE HEADQUARTERS
Vitesse Semiconductor Corporation
741 Calle Plano
Camarillo, CA 93012
Tel: 1-800-VITESSE • FAX:1-(805) 987-5896

For application support, latest technical literature, and locations of sales offices,
please visit our web site at
www.vitesse.com